Modeling and Controlling of Multi-scale Biological Networks with Applications to Precision Medicine

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Chieh Lo
B.S., Electrical Engineering, National Taiwan University

Carnegie Mellon University
Pittsburgh, PA
December, 2018

Acknowledgments

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family. I would like to express my deepest gratitude to my adviser, Dr. Radu Marculescu. I really enjoy working with him for his excellent guidance, patience, and providing me with the opportunity to do research. I always remember his teaching and his excellent presentation skill (which helps me a lot). The “angle" of doing research, think “outside of the box”, and explore new ideas without boundaries are the most important stuffs that he taught me. There aren’t enough words to express my gratitude and I hope all the best for him.

I would like to thank my parents and my family for their support. They are always on my side no matter what had happened.

I would also like to thank my committee members Dr. Diana Marculescu, Dr. N. Luisa Hiller, and Dr. Jie Cheng for providing me various resources, guiding my research and helping me to develop my background in engineering as well as biological science for the past several years.

Many thanks to Kartikeya Bhardwaj, Dr. Ryan Kim, Dr. Filipe Condessa, Connor Walsh, Irina Cazan, William Ehrett, Dr. Gonzalo Carvajal, Dr. Ananth Kalyanaraman and other friends for their hard work in many collaborative projects. My research would not have been possible without their help.

I would also like to thank my best friends, Chen-Hsuan Lin, Wei-Chiu Ma, and YiuChang Lin for their support in CMU.

Additionally, my two internships at AbbVie and Philips where I learn and develop algorithms and models give me great help. Without them, I would not be able to complete my research.

Finally, I would like to thank US National Science Foundation (Grant CCF-1514206, CPS-1135850) for financially supporting my research throughout the years.

Abstract

Biological networks can provide new insights in understanding basic mechanisms controlling normal cellular processes and disease pathologies. In this thesis, we develop a computational framework for bacteria and microbiome dynamics by modeling their interactions and evolution through multi-scale biological networks; this newly developed framework is a step towards making precision medicine a reality.

The first part of this thesis focuses on engineering bacteria at population level. To this end, we first develop a cell-level full pathway model to describe bacteria movement (*i.e.*, chemotaxis) and their communication mechanism (*i.e.*, quorum sensing). Based on this model, we then propose an autonomous and adaptive bacteria-based drug delivery system that integrates bacterial chemotaxis and quorum sensing to deliver drugs efficiently and precisely at various locations in the human body. Further, we address the problem of antibiotic resistance. As new drug-resistant bacteria continue to appear, substantial research efforts have shifted focus toward innovative therapies, such as quorum sensing inhibition (QSI) which aims at disabling bacteria molecular signaling channels. However, the excessive use of QSI may induce the selective pressure among bacteria and make them resistant to QSI. To address this issue, we propose an autonomous biological controller that can adaptively generate QSIs and control the nutrient availability in the environment.

In the second part of this thesis, we consider multiple species of bacteria (microbiome) and investigate the formation and dynamics of microbiome interaction networks. Despite the role the microbiome plays in human health, models and algorithms that can qualitatively and quantitatively describe the interactions among various microbes (*i.e.*, how microbes interact with each other) and identify the microbiome community structure (*i.e.*, how microbes form different functional groups) have not yet been established. Once a general network model becomes available, we can not only develop strategies to cure the diseases caused by changes in the microbiome status, but also provide valuable prophylactic information and analyze the impact of drugs or probiotics on human health; this is precisely what motivates our proposed research on modeling and controlling the microbiome dynamics.

Finally, we propose a computational model that can correctly identify human microbiome related disease. We further explore the possibility of using our network analysis model to identify various microbial functional groups that can be viewed as potential therapeutic targets. The expected results would suggest the customized use of probiotics or drugs in clinical settings according to patients' specific condition in order to maximize drugs efficacy. Consequently, together with the inferred microbial interaction network, the proposed framework can be used to develop a personalized medical system with treatment generation capabilities.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the background knowledge on bacteria and microbiome and the corresponding microbiome datasets. More specifically, we first introduce bacteria basics; this includes bacterial chemotaxis, quorum sensing, and virulence measures. Next, we discuss the important role of human microbiome in microbiome related diseases. Finally, we summarize the metagenomic datasets used in the subsequent chapters.

1.1 Bacteria basics

Bacteria constitute a large domain of prokaryotic microorganisms. They are typically a few micro-metres long, and have various shapes, ranging from spheres to rods and spirals (see Fig.1.1). Despite their size, bacteria hold an overwhelmingly significant influence over earth's biota [5], playing both beneficial [6] and pernicious [7, 8] roles in human health. For instance, the human body contains many trillions of bacteria belonging to several hundred different species. In fact, bacteria cells is on the same order of human cells inside our body [9]. Most of these bacteria live on our skin, teeth, in the mucous membranes, intestine, and gut.

Generally speaking, anaerobic bacteria do not cause diseases, and many of them have useful functions, such as helping with food digestion. However, these bacteria can also cause diseases if they enter tissues that are usually off-limits and have no defense mechanisms against their invasion. For example, bacteria can cause serious infections in many parts of our body, such as sinuses, middle ear, lungs, brain, abdomen, pelvis, and skin, even enter the bloodstream and spread across the body.

1.2 Bacteria quorum sensing (QS) and pathogenicity

Quorum sensing (QS) is a form of molecular signaling which plays a vital role in the coordination and synchronization of bacteria collective behaviors, as shown in Fig.1.2. QS inhibition critically relies on the upstream control that the QS system holds over collective and cooperative behaviors, such as biofilm formation, the production of secondary metabolites, and the expression of disease-causing virulence factors [10, 11, 12, 13]. For

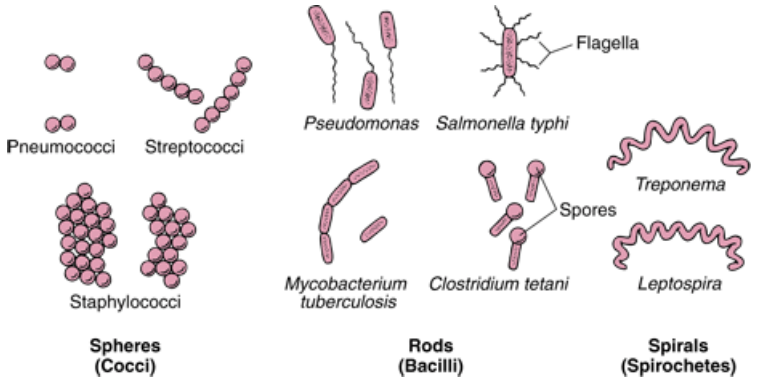

Figure 1.1: All bacteria may be classified as one of three basic shapes: spheres (cocci), rods (bacilli), and spirals or helixes (spirochetes). Many bacterial species are motile by means of flagella; flagella can be located at various sites of the cell body, leading to different swimming modes.

example, in the opportunistic pathogen *Pseudomonas aeruginosa* (PA), roughly 5-10% of cellular metabolic processes during infection are regulated by the QS system [14]. In fact, initial experiments suggest that quorum sensing inhibitors (QSIs) can swiftly and dramatically reduce virulence while increasing pathogen susceptibility to conventional antibiotics and immune system responses [15, 16, 17, 18].

QS can be roughly divided into 4 phases [19]: (1) constitutive production of small autoinducer signaling molecules; (2) release of the autoinducers (generally through passive diffusion of small molecules) into surrounding environment; and (3) sensing of autoinducers by specific receptors, leading to (4) activation of QS-regulated products (see Fig.1.2). Note that the QS activation also leads to increased synthesis of the proteins involved in signal molecule and signal receptor production, creating a positive feedback loop, which is why the signaling molecules are referred to as autoinducers.

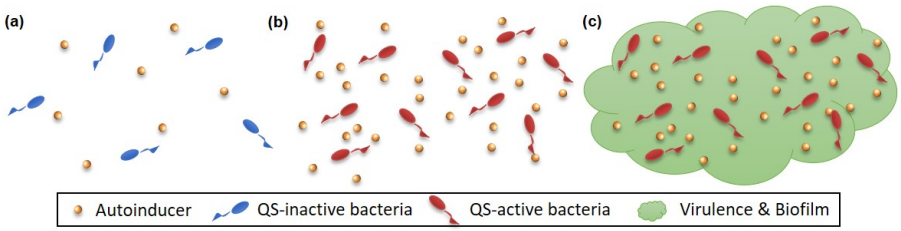

Figure 1.2: Quorum sensing as a distributed decision-making process, as long as individual cells have: (a) a means of assessing the number of peers they interact with and (b) a standard response once a threshold number of cells is detected. Figure from Wikipedia.

1.3 Bacteria chemotaxis

Bacteria chemotaxis is the movement of bacteria in response to chemical stimuli, as bacteria can direct their movements according to certain chemicals in their environment. This is important for bacteria to find food (e.g., glucose) by swimming toward the highest concentration of food molecules, or flee from poisons (e.g., phenol).

The overall movement of a bacterium is the result of alternating tumble and swim phases. Bacteria such as *E. coli* are unable to choose the direction in which they swim, and are unable to swim in a straight line for more than a few seconds due to rotational diffusion. In the presence of a chemical gradient, bacteria will direct their motion based on the concentration gradient. If the bacterium senses that it is moving in the correct direction (toward attractant/away from repellent), it will keep swimming in a straight line for a longer time before tumbling; however, if it is moving in the wrong direction, it will tumble sooner and try a new direction at random. In other words, bacteria like *E. coli* use temporal sensing to decide whether their situation is improving or not, and in this way, find the location with the highest concentration of attractant (usually the source) quite precisely.

1.4 Bacteria antibiotic resistance and microbiome related disease

Bacteria develop resistance to drugs because (1) they acquire genes from other bacteria that have become resistant (horizontal gene transfer); or (2) because of random mutations in their own genes. For example, soon after the drug penicillin was introduced in the 1940s, some *Staphylococcus aureus (S. aureus)* bacteria acquired genes that are resistant to penicillin. The strains that possess these resistant genes have a survival advantage when penicillin is used; this so-called selective pressure makes the drug-resistant strain become dominant over time in the pathogen populations.

Recently, as new drug-resistant bacterial strains, such as carbapenem-resistant Enterobacteriaceae (CRE), continue to appear, health officials are raising concern over the future efficacy of traditional antibiotics [20]. In response, substantial research efforts have shifted focus toward innovative targeted drug development strategies including antivirulence therapy targeting cellular functions essential for pathogenesis within the human host rather than cellular vitality [21]. Additionally, recent research has found that undesirable changes in microbiome compositions can cause microbiome related disease such as inflammatory bowel disease (IBD). Consequently, finding new ways to analyze the human microbiome may help us better understand and cure diseases without the need to use antibiotics or drugs.

1.5 Human metagenomic datasets

Due to the recent advances in modern metagenomics sequencing methods, it becomes possible to directly analyze the microbial communities within the human body. High-

throughput comparative metagenomics data is obtained from the next-generation sequencing (NGS) platforms. More specifically, two types of gene sequencing data are considered: 16S rRNA and shotgun data. Shotgun data analyses are accomplished by unrestricted sequencing of the genome of all microorganisms present in a sample; on the contrary, the domain of 16S rRNA is restricted to bacteria and archaea. Spatial (cross-sectional) and temporal (time-series) metagenomic data have been widely studied in recent years. Cross-sectional data mainly consists of samples collected from different body sites such as gut, mouth and skin. On the other hand, time-series data includes microbial dynamics of individuals usually in the span of years of observation.

1.6 Research questions

In this dissertation, we address the following fundamental research questions:

- How to develop a new cellular model of bacteria that can mathematically capture the communication mechanism among bacteria (QS) and engineer synthetic cells to control their virulence.
- How to develop a computational model for bacterial chemotaxis for drug delivery tasks that can efficiently and precisely deliver drugs at the target locations.
- How to utilize data-driven methods to reveal the underlying patterns of microbiome interactions among each other and the human body.
- How to use machine learning methods to accurately identify the disease states of human using metagenomic information.

By answering these questions, we are able to have a clear view of various mathematical models for describing bacteria QS, chemotaxis, metabolism, and motility. Based on these models, we can further explore how to exploit bacteria chemotaxis and QS for medical applications. Next, by studying the microbial metagenomic, we can understand how microbes interact and find out the “key players" that have huge impact on our human health. Also, we can identify the disease states based on the metagenomic information; this serves as the first step towards precision medicine.

1.7 Thesis overview

To answer above research questions, we propose to model biological networks at three different granularities: First, cell-level models (*i.e.*, single cell): quorum sensing, chemotaxis, growth, and inhibition models. Second, population-level models (*i.e.*, one to three species each with thousands of cells): networks of cell-cell interaction and population control through engineered cells. Third, microbiome-level models (*i.e.*, hundreds of species): microbiome association and interaction networks.

More specifically, we divided the dissertation into seven chapters. In Chapter 2, we tackle one of the most pressing matter nowadays, namely, antibiotic resistance, by modeling bacterial intercellular communication (quorum sensing), and evaluating the long-term effects of an alternative therapy, *i.e.*, QSI. We develop an autonomous biological controller to control the virulence through environment regulations; this strategy shows that we can effectively reduce threat of antibiotic resistance [22, 23]. In Chapter 3, a full chemotaxis pathway model is proposed including chemoreceptors such as Tar, Tsr, phosphorylation pathway, and flagella motor. We then utilize bacteria as a drug delivery agent to effectively and precisely deliver drugs at the target locations. Our results show that the proposed system can achieve outstanding performance in terms of delivery time and accuracy [24, 25]. In Chapter 4, we consider multiple species of bacteria (microbiome) and investigate the formation and dynamics of microbiome interaction networks based on the cross-sectional metagenomic datasets. By incorporating existing information from other sources as prior knowledge, we are able to better infer the microbial association compared to other existing methods. Our newly found microbial associations from the real datasets can be a credible direction for experimentalists validation [26]. In Chapter 5, we change our focus to time-series metagenomic datasets which can provide casual relationships among microbes. We solve the microbial interaction network by formulating it as a regularized linear regression problem. We can also incorporate prior knowledge obtained from cross-sectional data; this can greatly leverage the high dimensionality problem due to the nature of the metagenomic data [27]. In Chapter 6, we propose to use microbial profile as features to diagnose microbiome-related diseases. We propose a novel framework, classification of host phenotypes from Metagenomic data using Neural Networks (MetaNN), which integrates a data augmentation method and neural network models that can effectively solve the problem of data high dimensionality and model over-fitting. We show that our proposed framework outperforms other existing machine learning methods on several metagenomic datasets. Finally, in Chapter 7, we highlight the key findings and conclusions of the work presented in this thesis and suggest a few possible follow-up directions.

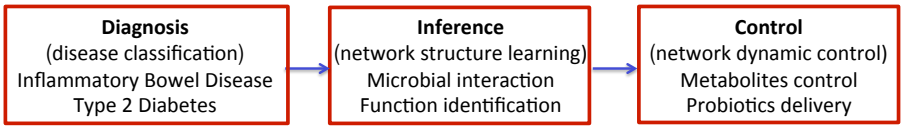

Figure 1.3: Our contributions toward precision medicine are three folds: disease diagnosis, network inference (key microbes identification), and network control.

1.8 Thesis contributions

As shown in Fig. 1.3, our contributions for the development of precision medicine are three folds. First, our proposed diagnostic framework, MetaNN, can accurately identify microbiome related disease such as inflammatory bowel disease. Second, our network inference algorithms, Microbial Prior Lasso (MPLasso) and Microbial Time-series Prior

Lasso (MTPLasso) can first infer the microbial networks and then identify "key microbes" that may correspond to our host phenotypes. Third, our proposed simulation framework and the biological control system (cell-based method) can be used to dynamically and autonomously control metabolites expression and probiotics delivery. Overall, our proposed framework and model can benefit both patient and physician towards better personalized treatments.

Chapter 2

Autonomous and Adaptive Control of Populations of Bacteria Through Environment Regulation

To show the possibility of controlling populations of bacteria, this chapter presents our proposed biological controller that can autonomously and adaptively generate quorum sensing inhibitors and control the iron availability in the environment. As the main theoretical contribution, we provide a detailed analysis of our proposed controller that includes inhibitor effectiveness and controller design. We first focus on the mathematical modeling of the QS regulation system of the opportunistic human pathogen *Pseudomonas aeruginosa* (PA), bacteria growth, and QS inhibition (QSI) model. Next, we analyze the QS system response to environment stimuli and bacteria growth model via simulation. Finally, we formulate a constrained optimization problem for designing the biological controller and provide an design example based on the proposed design guidelines.

2.1 Introduction and motivation

The fight against bacterial virulence represents one of the big challenges of modern medicine. Indeed, due to the large-scale proliferation and inappropriate use of antibiotics, new strains of antibiotic-resistant bacteria begin to emerge. These new, stronger bacteria pose a significant threat to humans health and welfare. To fight antibiotic-resistant bacteria, we propose to engineer synthetic cells, insert them in a population of bacteria, and then control the dynamics and virulence of the entire population [28]. We note that while previous work [29][30] proposed to engineer cells to kill the antibiotic-resistant bacteria, this kind of approaches may actually select strains that can survive under such treatments. In contrast, in this chapter, we design an autonomous controller that can not only regulate the cell-cell communication, but also manipulate the environment signals in order to reduce bacterial virulence and prevent selective pressure among antibiotic-resistant strains.

Getting now into details, bacteria can form biofilms, express virulence, and become resistant to antibiotics after reaching a quorum through cell-cell communication. Quorum sensing (QS) is a fundamental cell-cell communication that is used by bacteria to obtain cell density information and hence, alter their genes expression [31]. In particular, the QS system used by Gram-negative bacteria is mediated by diffusible signaling molecules, termed "autoinducers"[1] [31]. For instance, PA possess a complex QS system that regulates genes and operons which constitute over 6% of its genome. These genes coordinate the biofilm formation and produce large amounts of virulence factors, such as elastase, rhamnolipids, and pyocyanin [32].

QS regulation can be strongly affected by various environmental factors [33]; for example, for PA, the nutrient availability has been shown to affect the expression of QS genes [34]. Several other studies have demonstrated that high iron concentrations favor the formation of biofilms and higher growth rates, but restrict the expression of QS signals [35][36]. On the other hand, QS also regulates bacteria access to nutrients and environmental niches that favor their growth and defense.

The intertwined regulation between QS and environmental signals enable bacteria to thrive in a stringent environment [37][38]; indeed, under such conditions, bacteria must coordinate the expression of related genes in order to successfully form and maintain biofilms [39]. For example, a shortage of iron availability in the environment leads to the increased expression of iron acquisition system [40][41] and decreased activity of pathways that rely on relatively large amounts of iron [32]. However, a rigorous mathematical model that can precisely capture the complex relationship between the QS system and bacteria growth has not yet been explored. Additionally, most studies published so far focus on observing the qualitative behaviors of bacteria and lack the ability to predict long term evolution dynamics under different environmental conditions [42].

We argue that having a quantitative model of QS behavior available can not only capture the important dynamics of bacteria growth, but also give credible predictions for the long term behaviors of bacteria virulence. Based on the QS model in [3][43], we further extend the model to account the effect of environment stimuli; this analytical QS model is the first major contribution of this work. We also raise another important question: Given such an analytical (*i.e.*, quantitative) model, what are the strategies to control bacteria virulence and growth rate, while lowering the chances of developing drug resistance or inducing selective pressure among bacteria wild type and mutants? To address this second question, we propose an autonomous biological controller that can dynamically generate different types of inhibitors; this controller is based on genetic parts used to design genetic circuits [44].

To shed light on the complex relationship between QS and environment signals, we use the opportunistic pathogen PA as a canonical example (Fig. 2.1(a)). PA requires an abundance of iron to produce and sustain infections. Hence, iron depletion prevents bacterial growth and affects their metabolism [45]. By expressing siderophores, PA can sequester iron from environment and regain the ability to form biofilm [46]. Two major genetic components of QS, namely, the las and pqs QS systems have been identified in PA [40]. As shown in Fig. 2.1(a), the las QS system sits at the upstream of pqs QS

[1] Denoted as AI in this thesis.

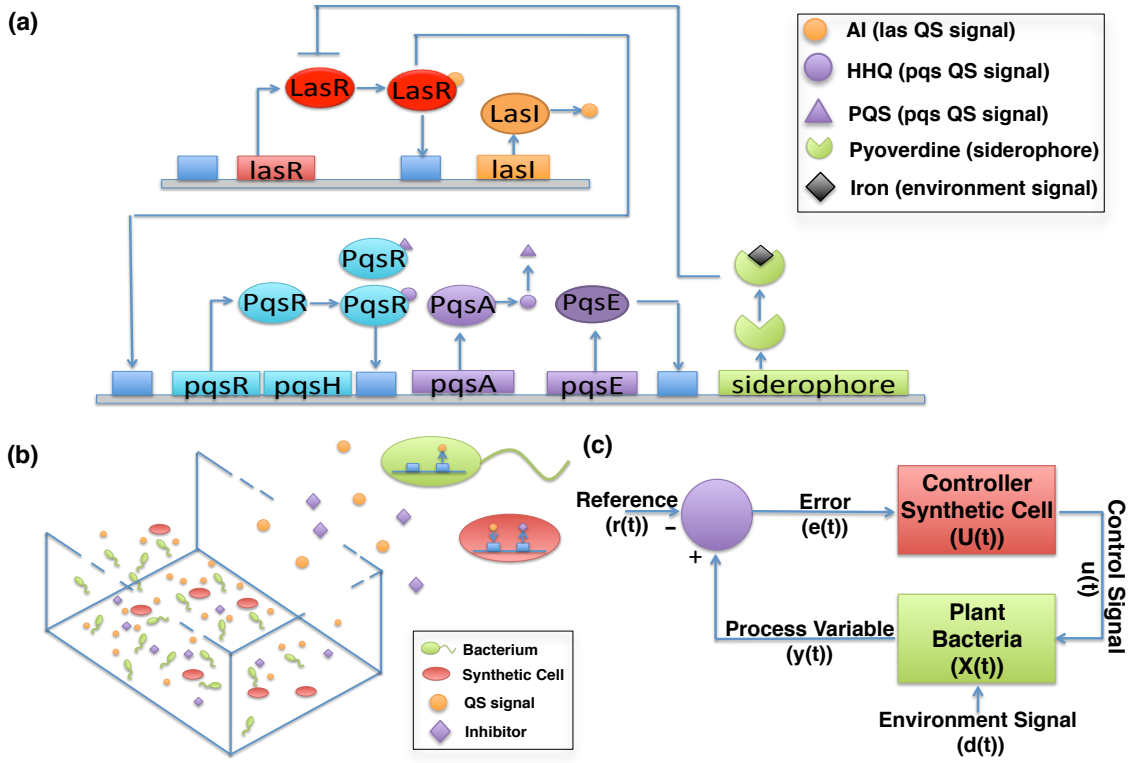

Figure 2.1: (a) Interconnection of the las and pqs QS systems of PA. The small orange circles represent autoinducer molecules (AIs) that move freely through the cell membrane; the purple circles and triangles represent the QS signals of pqs QS system; the green particles represent the pyoverdine molecules. The red, orange, blue, and purple ovals represent LasR, LasI, PqsR, PqsA, and PqsE, respectively. (b) The simulation environment, where bacteria (green cells) flow in space and release the QS signals (orange circles). By placing the synthetic (red) cells in the environment, they first react to the QS signals and then express the inhibitors (purple diamonds) that can quench the communication among bacteria. (c) The diagram of the proposed control system. The environment signal (d) (e.g., nutrient availability) can be viewed as the input to the intracellular bacterial regulations system. The control variables (u) are the QSI inhibitors which can control the dynamics of bacteria. The process variable (y) can be detected by the synthetic controller.

system and positively regulates the operons[2] of the pqs QS system [37]. The pqs QS system produces molecules that mediate the expression of the siderophores [40][47]. To enhance the expression of the siderophores, the upstream las QS system needs to highly express proteins in order to induce the downstream pqs QS system. Hence, as the iron concentration is relatively low, the las QS system is highly expressed and vice versa. However, bacteria can become more virulent when the las QS system strongly expresses proteins as this can regulate the virulence genes.

In summary, in an iron depletion environment, the growth of bacteria can be delayed but the virulence can actually increase [36]. To control both the virulence and the growth

[2]Operon is a functioning unit of DNA containing a cluster of genes under the control of a single promoter.

rate of bacteria simultaneously, we use two different kinds of inhibitors that target the las QS system and the iron availability in the environment. Different types of inhibitors can have different effects on virulence and growth rate, hence, multi-inhibitor schemes can be more effective. To synthesize inhibitors and control the iron concentration, a few simple genetic circuits can serve as the basic control units which automatically detect and react to environment changes. For example, we can construct the genetic circuits by cloning the genes in the plasmid, such as the $aiiA$ gene which expresses the enzyme that hydrolyzes the AI [48] (Fig. 2.1(b)). However, synthesizing excessive amounts of inhibitors in the environment can have toxic effects on the host.

Therefore, in this chapter, we propose a dynamic optimization problem that incorporates bacteria QS, growth, and control dynamics. Solving this optimization problem allows us to choose the biological parameters that can be further used to design controllers that can generate the optimal amount of inhibitors adaptively. By placing the biological controller into the bacterial environment, it becomes possible to detect the concentration of the signaling molecules in the environment and then generate the right amount of inhibitors in real-time. Consequently, the proposed system aims at a paradigm shift from manual to autonomous control of bacteria population dynamics (Fig. 2.1(c)).

2.2 Cell-level mathematical modeling

In this section, we model the dynamics of bacteria QS, growth, and inhibition systems based on ordinary differential equations (ODEs). To uncover the complex interaction between the QS and environment signals (*i.e.*, iron), we first model the las and the pqs QS systems [40] and the growth of PA explicitly. Next, we calibrate our models with reported experimental data [36]; that is, at different iron concentration levels, we calibrate the relative concentration change of the LasR protein (main receptor in las QS system); this provides the basis for examining the QS system response and subsequently designing the biological controller.

2.2.1 QS model of *Pseudomonas aeruginosa*

The QS regulatory network of PA consists of two main systems: las and pqs. The las and the pqs QS systems are linked by (1) LasR-AI complex which directly *up* regulates the expression of the PqsR and the PqsH proteins and (2) iron-chelated complex which *down* regulates the expression of the LasR protein and then reduces the expression of the siderophore (a negative feedback loop). The entire QS system is modeled as follows:

las QS model

The regulatory network of the las QS system has two feedback loops. As shown in Fig. 2.1(a), the LasR-AI complex up regulates the expression of both $lasR$ and the $lasI$ genes. Based on the ODE models proposed in [3][43], we have the following equations

Symbol	Parameter	Source
K	Half saturation concentration	[3][4]
U	Utilization coefficient	introduce in this thesis
V	Maximum production rate	[3][4]
b	Molecule decay rate	[3][4]
c	Basal production rate	[3][4]
d	Membrane diffusion rate	[4]
α	Binding rate	[3][4]
β	Enzyme production rate	[3][4]
δ	Unbinding rate	[3][4]
ρ	Cell density	[4]
P	Promoter strength	[49]
r	Basal production rate	[49]

Table 2.1: Table with model parameters for cellular-level models

for the las QS system:

$$\frac{d[A]}{dt} = c_A + \frac{V_A[C]}{K_A + [C]} - \alpha_{RA}[R][A] + \delta_{RA}[RA] - b_A[A] - \frac{d_A}{\rho}([A_{EX}] - [A]) \tag{2.1}$$

$$\frac{d[A_{EX}]}{dt} = -b_{A_{EX}}[A_{EX}] - \frac{d_A}{1-\rho}([A_{EX}] - [A]) \tag{2.2}$$

$$\frac{d[R]}{dt} = c_R + \frac{V_R[C]}{K_R + [C]} - \alpha_{RA}[R][A] + \delta_{RA}[RA] - b_R[R] \tag{2.3}$$

$$\frac{d[RA]}{dt} = \alpha_{RA}[R][A] - 2\alpha_{RA^2}[RA]^2 - \delta_{RA}[RA] + 2\delta_{RA^2}[C] \tag{2.4}$$

$$\frac{d[C]}{dt} = \alpha_{RA^2}[RA]^2 - \delta_{RA^2}[C] \tag{2.5}$$

where $[X]$ denotes the concentration of a particular molecular species X. In our formulation, A stands for AI, A_{EX} is the extracellular AI, R is LasR, RA is the LasR-AI complex and C is the dimerized complex. The meaning of biological constants are listed in Table 2.1 while their numerical values are listed in Table 8.1 and Table 8.2 in the **Appendix**.

pqs QS model

The pqs QS system [37] consists of two kinds of signaling molecules, PQS (2-heptyl-3,4-dihydroxyquinoline) and HHQ (4-hydroxy-2-heptylquinoline); in addition; we have one receptor regulator PqsR. The PqsR protein can bind to the HHQ and the PQS molecules can up regulate the $pqsABCDE$ operon; this forms a positive feedback since the PqsA protein directly up regulates the synthesis of the HHQ molecules. Another signaling molecule, PQS, is converted from HHQ via PqsH protein. Therefore, pqs QS system forms a second positive feedback loop. By explicitly capturing regulations among proteins and molecules based on molecular transcription and translation, we propose the following new ODEs to describe the pqs QS system:

$$\frac{d[pR]}{dt} = c_{pR} + \frac{V_{pR}[C]}{K_{pR}+[C]} - \alpha_{pR}([pR][A_1]+[pR][A_2]) + \delta_{pR}([C_1]+[C_2]) - b_{pR}[pR] \tag{2.6}$$

$$\frac{d[pH]}{dt} = c_{pH} + \frac{V_{pH}[C]}{K_{pH}+[C]} - b_{pH}[pH] \tag{2.7}$$

$$\frac{d[pA]}{dt} = c_{pA} + \frac{V_{pA,1}[C_1]}{K_{pA,1}+[C_1]} \frac{V_{pA,2}[C_2]}{K_{pA,2}+[C_2]} - b_{pA}[pA] \tag{2.8}$$

$$\frac{d[pE]}{dt} = c_{pE} + \frac{V_{pE,1}[C_1]}{K_{pE,1}+[C_1]} \frac{V_{pE,1}[C_2]}{K_{pE,1}+[C_2]} - b_{pE}[pE] \tag{2.9}$$

$$\frac{d[C_1]}{dt} = \alpha_{pR}[pR][A_1] - \delta_{pR}[C_1] \tag{2.10}$$

$$\frac{d[C_2]}{dt} = \alpha_{pR}[R_2][A_2] - \delta_{pR}[C_2] \tag{2.11}$$

$$\frac{d[A_1]}{dt} = \beta_{pA}[pA]\frac{K_{A_1}}{K_{A_1}+[pE]} - \alpha_{pR}[pR][A_1] + \delta_{pR}[C_1] - b_{A_1}[A_1] + \frac{d_{A_1}}{\rho}([A_{1EX}]-[A_1]) \tag{2.12}$$

$$\frac{d[A_{1EX}]}{dt} = -b_{A_1}[A_{1EX}] - \frac{d_{A_1}}{1-\rho}([A_{1EX}-[A_1]) \tag{2.13}$$

$$\frac{d[A_2]}{dt} = \beta_{pH}[pH][A_1] - \alpha_{pR}[pR][A_2] + \delta_{pR}[C_2] - b_{A_2}[A_2] + \frac{d_{A_2}}{\rho}([A_{2EX}]-[A_2]) \tag{2.14}$$

$$\frac{d[A_{2EX}]}{dt} = -b_{A_2}[A_{2EX}] - \frac{d_{A_2}}{1-\rho}([A_{2EX}]-[A_2]) \tag{2.15}$$

$$\frac{d[Pyo]}{dt} = c_{Pyo} + \frac{V_{Pyo}[pE]}{K_{Pyo}+[pE]} - b_{Pyo}[Pyo] + \frac{d_{Pyo}}{\rho}([Pyo_{EX}]-[Pyo]) \tag{2.16}$$

$$\frac{d[Pyo_{EX}]}{dt} = -\alpha_I[Pyo_{EX}][I] - b_{Pyo_{EX}}[Pyo_{EX}] + \frac{d_{Pyo}}{1-\rho}([Pyo_{EX}]-[Pyo]) \tag{2.17}$$

$$\frac{d[Q]}{dt} = -b_Q[Q] + \frac{d_Q}{\rho}([Q_{EX}]-[Q]) \tag{2.18}$$

$$\frac{d[Q_{EX}]}{dt} = \alpha_I[Pyo_{EX}][I] + \frac{d_Q}{1-\rho}([Q_{EX}]-[Q]) \tag{2.19}$$

where $[X]$ denotes the concentration of a particular molecular species X. In our formulation, pA, pE, pH, and pR stand for PqsA, PqsE, PqsH and PqsR, respectively. A_1, A_2, C_1 and C_2 represent HHQ, PQS, PqsR-HHQ and PqsR-PQS, respectively. Pyo and I represent pyoverdine and iron, respectively. Q is the iron-chelated complex.

Given the pqs QS system, we modify the expression of the LasR protein in (2.3) as follows:

$$\frac{d[R]}{dt} = c_R + \frac{V_R[C]}{K_R+[C]} \frac{V_Q K_Q}{K_Q+[Q]} - \alpha_{RA}[R][A] + \delta_{RA}[RA] - b_R[R] \tag{2.20}$$

where we add a new term (*i.e.*, $\frac{V_Q K_Q}{K_Q+[Q]}$) to account for the effect of iron-chelated complex Q. In this equation, the parameters V_Q and K_Q represent the maximum production rate and Michaelis-Menten constant, respectively.

2.2.2 Bacteria growth model and virulence measures

To describe bacteria growth, Monod introduced the concept of single nutrient controlled kinetics [50], which relates the specific growth rate (μ_X) of a bacterium cell mass (X) to the substrate concentration (S). The kinetic parameters, *i.e.*, maximum specific growth rate (k_X) and substrate affinity (K_S), are assumed to be constant and dependent on strain, medium, and growth conditions (*e.g.*, temperature, pH). In our model, however, we need to consider a second nutrient source and add a new term Q to describe it. However, when cells are metabolically active, but not growing or dividing, they may still take up substrate.

To address bacteria size reduction, a maintenance rate (m) is generally used; consequently, we improve Monod's model as follows:

$$\mu_X = k_X \cdot \frac{S+Q}{S+Q+K_g} \tag{2.21}$$

$$\frac{dX}{dt} = (\mu_X - m) \cdot X \tag{2.22}$$

We also define the virulence (Vir) as the concentration of LasR-AI complex as it controls the downstream virulence expressions; therefore, the total virulence (TV) of the bacteria population is defined as the product of the virulence and the number of bacteria (N)[3]:

$$TV = Vir \times N \tag{2.23}$$

We note that, as discussed later in Section. 2.4, both Vir and N are variables that depend on time (t) and the set of biological parameters ($\mathbf{p}$).

2.2.3 Inhibition model

We target the bacterial iron acquisition as a strategy to control the virulence of bacteria. From our previous discussion, the QS signaling pathways are the primary target. More precisely, to control the iron uptake rate of PA, we propose two strategies that can either modulate the iron uptake or inhibit the upstream las QS system.

AI inhibitors

The AI inhibitor hydrolyzes the extracellular AI molecules which can be viewed as a degradation source and assumed to follow the Michaelis-Menten kinetics. Accordingly, (2.2) should be modified as:

$$\frac{d[A_{EX}]}{dt} = -b_{A_{EX}}[A_{EX}] - \frac{d_A}{1-\rho}([A_{EX}] - [A]) - \frac{V_E[A_{I_{EX}}][A_{EX}]}{K_{A_{EX}} + [A_{EX}]} \tag{2.24}$$

where $A_{I_{EX}}$ denotes the extracellular AI inhibitor.

[3] Since the number of bacteria is proportional to the biomass, we use biomass and the number of bacteria to account for the total virulence interchangeably.

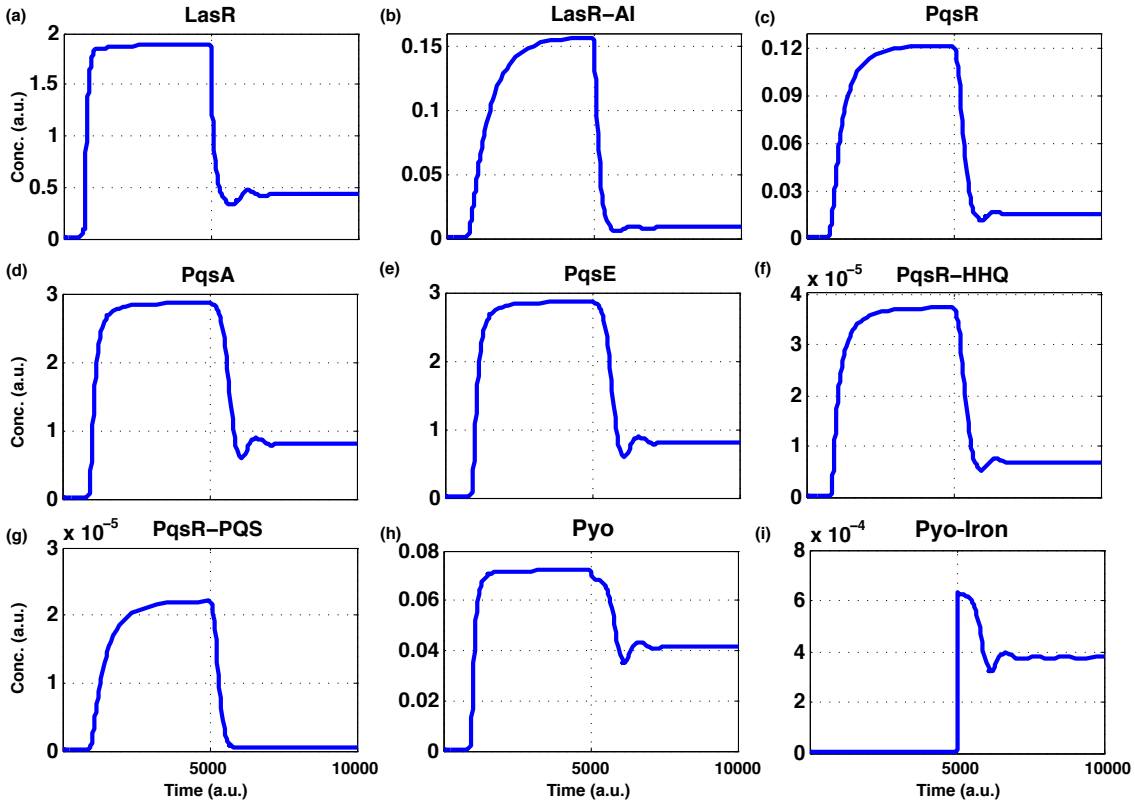

Figure 2.2: Simulation results of the PA QS system responses to different iron concentrations. At first, the iron concentration is 0.01 (a.u.); later at time t = 5000 (a.u.), it is changed to 1 (a.u.). (a) As iron concentration increases, the expression of LasR proteins is repressed due to the negative feedback of iron-chelated complexes (see Fig 2.1(a)). (b) The concentration of the LasR-AI complex decreases accordingly. The downstream proteins (*i.e.*, (c) PqsR, (d) PqsA, and (e) PqsE) are all positively regulated by LasR. Hence, they change in accordance with LasR protein. The (f) PqsR-HHQ and (g) PqsR-PQS concentrations also decrease due to the decrease of PqsR. (h) Pyoverdine (Pyo) concentration shows similar profile since it is positively regulated by PqsE. (i) The concentration of iron-chelated (Pyo-Iron) complex increases due to the high affinity of pyoverdine and iron.

Iron inhibitors

Different species of bacteria can produce different kinds of siderophores to trap the iron from environment, *e.g.*, $Enterobactin$ produced by *E. coli* cannot be up-taken by PA. If the amount of iron is limited, bacteria compete with each other in order to retain the essential resources. Therefore, we consider the siderophores produced by other bacteria as iron inhibitors that can limit the availability of iron in the environment. The dynamics of the available iron in the environment can be simply modeled as:

$$[I_{ava}] = [I](\frac{[Pyo_{EX}]}{[I_I] + [Pyo_{EX}]}) \tag{2.25}$$

where I_{ava} denotes the available iron in the environment and I_I stands for the iron inhibitor. By replacing I with I_{ava} in (2.17) and (2.19), we can incorporate the iron inhibitor dynamics to the QS model.

2.3 QS system analysis

In this section, we first examine the QS system responses to different chemical substances. Next, we examine the effects of substrate utilization constant on bacteria growth. Finally, we examine the effectiveness of AI and iron inhibitors.

2.3.1 QS system responses

We first examine the responses of the las and pqs QS systems by varying the concentration of available iron in the environment. Fig. 2.2 shows the QS system responses to several chemical substances. At first, the concentration of iron is 0.01 (arbitrary units (a.u.)); at $t = 5000$ (a.u.), the concentration of iron is changed to 1.00 (a.u.) (*i.e.*, one hundred fold increase). We observe that the LasR protein concentration decreases due to the increase of the iron concentration; this is discussed in [36] and illustrated with the negative feedback (see also Fig. 2.1(a)). The other chemical substances show similar patterns except the iron-chelated complex which directly increases the growth rate. This way, the system responses confirm that our model can precisely describe the changes of chemical substance concentrations when the concentration of iron changes; this confirms the experiments in [36].

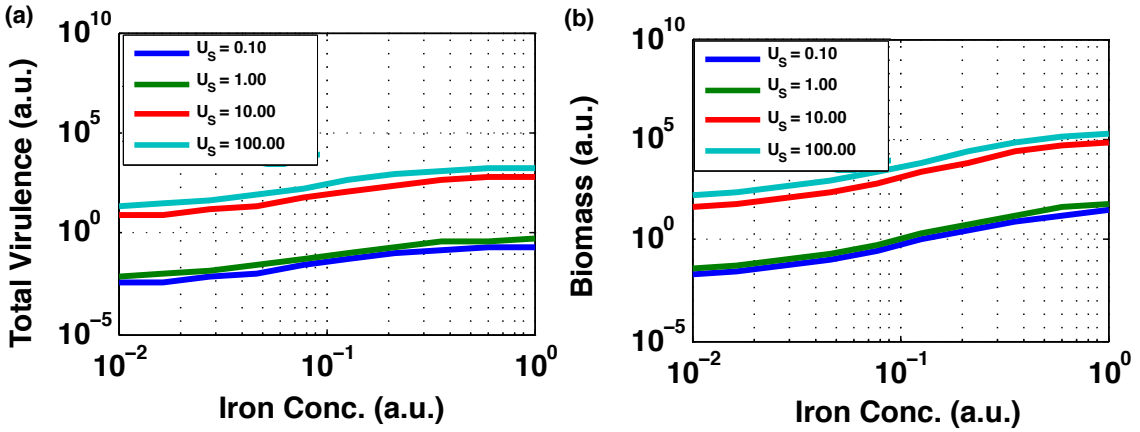

Figure 2.3: Substrate utilization constant (U_s) selection. (a) The effect of substrate utilization constant on total virulence under different iron concentrations. (b) The effect of substrate utilization constant on biomass under different iron concentrations.

2.3.2 Growth model: Utilization constant

In Monod's bacterial growth model, bacteria consume the substrate for their growth. We assume the utilization of substrate (U_S) is constant under different iron concentrations. However, the exact values of the utilization constant are hard to measure and estimate experimentally. To determine the U_S value, we examine the changes of total virulence and biomass under different iron concentrations.

As shown in Fig. 2.3(a), once a certain concentration of iron is reached, the larger the U_S, the greater the total virulence; this is because a low consumption rate of substrate results in a nutrient abundant environment that favors bacteria growth. We can observe that the biomass and the total virulence are almost identical if U_S is greater than 10. Hence, in the following analysis, we set U_S to 10.

2.4 Proposed biological controller

To dynamically and autonomously generate either AI or iron inhibitors, we propose to synthesize some specified genetic circuitry. To obtain variable combinations of the inhibitors with optimal expression levels, we build two circuits separately. More precisely, to generate AI inhibitor, we can assemble the $aiiA$ genes with the lux promoter to sense the concentration of LasR-AI (C). Similarly, the iron inhibitor circuit is built with genes that can express the competing siderophores and sense the concentration of iron-chelated complexes (Q). Based on the genetic circuitry in [49], we model the dynamics of the new biological controller with the following ODEs:

$$\frac{d[A_I]}{dt} = P_{A_I}\left(\frac{1}{r_{A_I}} + \frac{[C]^2}{K_{A_I}^2 + [C]^2}\right) - b_{A_I}[A_I] + \frac{d_{A_I}}{\rho_s}([A_{I_{EX}}] - [A_I]) \tag{2.26}$$

$$\frac{d[A_{I_{EX}}]}{dt} = -\frac{d_{A_I}}{1-\rho_s}([A_{I_{EX}}] - [A_I]) - b_{A_{I_{EX}}}[A_{I_{EX}}] \tag{2.27}$$

$$\frac{d[I_I]}{dt} = P_{I_I}\left(\frac{1}{r_{I_I}} + \frac{[Q]^2}{K_{I_I}^2 + [Q]^2}\right) - b_{I_I}[I_I] + \frac{d_{I_I}}{\rho_s}([I_{I_{EX}}] - [I_I]) \tag{2.28}$$

$$\frac{d[I_{I_{EX}}]}{dt} = -\frac{d_{I_I}}{1-\rho_s}([I_{I_{EX}}] - [I_I]) - b_{I_{I_{EX}}}[I_{I_{EX}}] \tag{2.29}$$

where A_I (I_I) and $A_{I_{EX}}$ ($I_{I_{EX}}$) denote the intracellular and extracellular concentration of the AI (iron) inhibitors, respectively. The inhibitors production rate (second term) in (2.26) and (2.28) can be characterized by the binding of the LasR-AI and iron-chelated complex, respectively. The product of the promoter strength (P_{A_I} and P_{I_I}) and the basal production rate (r_{A_1} and r_{I_I}) characterize the minimal expression rate when there is no LasR-AI and iron-chelated complex present, respectively[4].

[4]We discuss a design example for the biological parameters in subsequent sections.

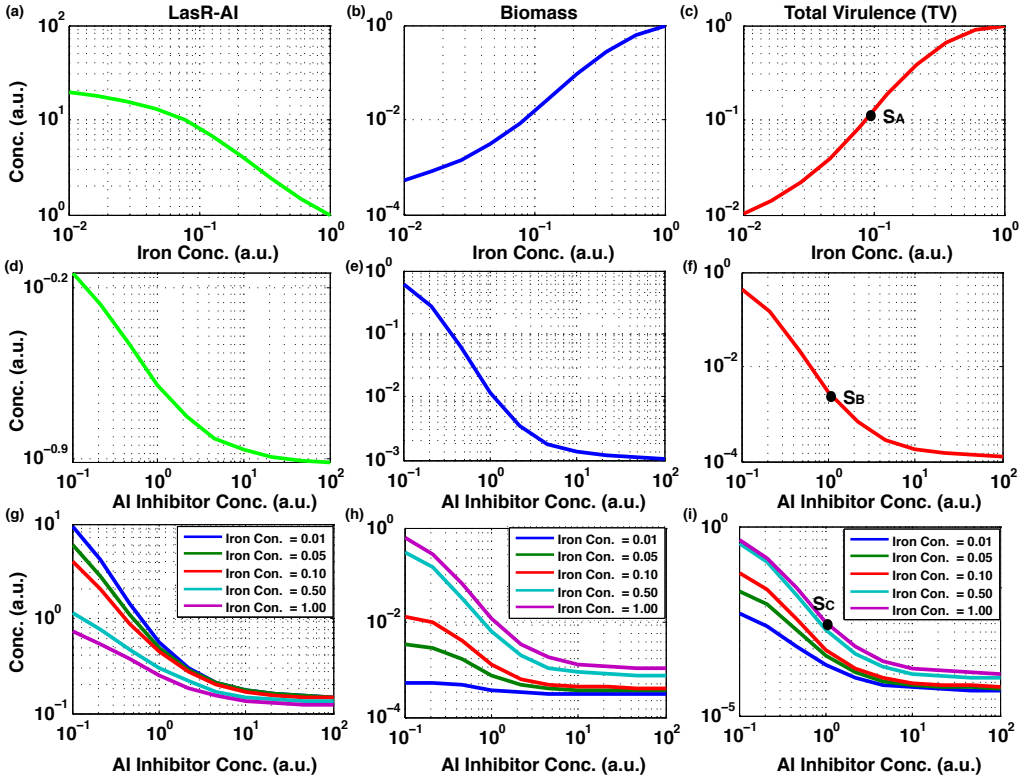

Figure 2.4: The concentration changes of LasR-AI, biomass, and the TV due to the effect of different inhibitors. (a), (b), and (c) show the effect of iron concentration alone. (d), (e), and (f) show the effect of AI inhibitors alone. (g), (h), and (i) show the combined effect of iron and AI inhibitors. The set points in (c), (f), and (i) are denoted as S_A, S_B, and S_C, respectively; they are used to derive results in Fig. 2.5

Based on the general constrained dynamic optimization formulation and control dynamics (see **Appendix**), we can formulate our problem as follows:

$$\begin{aligned} \min_{\mathbf{p}} \quad & TV_t = V(x_t, \mathbf{p}) \times N(x_t, \mathbf{p}) \\ \text{subject to} \quad & \dot{x}_t = f(x_t, u_t, d_t, \mathbf{p})\ \dot{u}_t = h(u_t, e_t, \mathbf{p}) \\ & y_t = g(x_t, \mathbf{p}) \qquad e_t = y_t - r_t \qquad \forall t \in [t_0, t_{FL}] \\ & x_{t_0}(\mathbf{p}) = x_0(\mathbf{p}) \\ & \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U \end{aligned} \tag{2.30}$$

where $t \in \mathbb{R}$ represents time, t_0, t_{FL} are the initial and final time, respectively, $t_i \in [t_0, t_{FL}]$, $\mathbf{x}$ and $\dot{\mathbf{x}} \in \mathbb{R}^n$ are the state variables and their time derivatives, respectively, and $\mathbf{p} \in \mathbb{R}^r$ capture the time-invariant biological parameters that can vary within $[\mathbf{p}^L, \mathbf{p}^U]$. The functions f and h are QS and QSI models, respectively. The function g selects the process variables (y) (*i.e.*, LasR-AI and iron-chelated complexes in our case). The state variable $\mathbf{x}$ represents the set of concentrations of chemical substances described by (2.1)-(2.19); the environment input (d) describes the environment conditions such as

the nutrient availability. The control variables (u) are the inhibitors which target the AI and iron availability.

The genetic circuit can be thought of as an integral controller which reacts to the concentration of LasR-AI and iron-chelated complexes, respectively. The error signal (e) is computed as the difference between the process variable and the reference signal (r); this then feeds back to the controller, which forms a closed loop (see Fig. 2.1(c)).

2.5 Population-level simulations

2.5.1 Inhibitors effectiveness

As shown in Fig. 2.4(a), when the iron concentration is high, the expression of the LasR is repressed. On the other hand, the biomass increases due to the higher growth rate (Fig. 2.4(b)). By using (2.23), the TV increases as the concentration of iron increases as shown in Fig. 2.4(c).

Figs. 2.4(d)-(f) show the effect of adding the AI inhibitor into the environment, both LasR-AI complex and biomass decrease (Fig. 2.4(d)(e)). Hence, the TV decreases as the amount of inhibitors increases.

Our most important observation shows that if we vary both the iron concentration and AI inhibitors, we may decrease the TV. Indeed, Fig. 2.4(i) shows that TV decreases as we increase the concentration of AI inhibitors and decrease the iron concentration. The AI inhibitor and iron concentration have opposite effects on the LasR-AI complex and the biomass. More precisely, lower concentrations of iron result in higher concentrations of the LasR-AI complex (Fig. 2.4(g)), but a decrease in the biomass production (Fig. 2.4(h)).

The autonomous biological controller we propose can automatically detect signals, react to environment, and adaptively release chemical substances for intended objectives. To control the TV, the objective is to find a set of biological parameters $\mathbf{p}$ that minimize (2.23). However, this objective function is subject to various biological constraints including the bacteria QS, growth and QSI, as well as control dynamics. Given the mathematical model in Section 2.2, we formulate a constrained dynamic optimization problem and solve it through numerical methods.

2.5.2 Biological parameters design

To design the biological parameters for our controller, we can numerically solve the above optimization problem by sampling biological parameters within the given constraints. From our analyses in Section 2.5.1, we notice that TV is a monotonically decreasing function (Fig. 2.4(i)). Consequently, by setting (2.23) to a desired value, we can solve (2.30) for biological parameters to fulfill the design specifications.

We now provide a design example for the control circuitry that can effectively achieve the setting objective value. As shown in Fig. 2.4(c)(f)(i), we first choose the setting points S_A, S_B, and S_C for three different strategies that can achieve desired TVs (0.1, 0.001 and 0.001 in this design examples). The biological parameters we choose to engineer are the promoter strength (P) and the basal production rate (r) in (2.26)-(2.29) since we

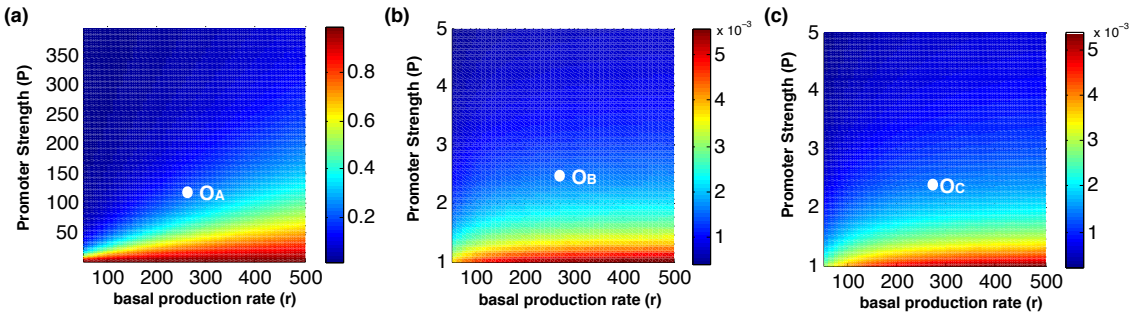

Figure 2.5: Operation points for different types of inhibitors. Different color indicates the TV of a certain combination of biological parameters (promoter strength (P) and basal production rate (r) described in (2.26)-(2.29)). (a) and (b) show the operation points O_A and O_B for iron and AI inhibitor alone; they can achieve TV around 0.1 and 0.001 where we choose the setting points S_A and S_B in Fig. 2.4(c) and (f), respectively. Based on the operation points we choose, we can solve the optimization problem and obtain the corresponding biological parameters where $P = 100, r = 250$ and $P = 2, r = 250$, respectively. (c) The operation point O_C for multi-inhibitors. In this case, $P = 100$ and $r = 250$.

can tune their values through the evolution method [44]. Next, by solving (2.30) through varying the value of a set of biological parameters within the given constraints ($[\mathbf{p}^{\mathbf{L}}, \mathbf{p}^{\mathbf{U}}]$), we can obtain the most suitable combination of biological parameters that express the minimal amount of inhibitors. Fig. 2.5 shows the operation points O_A, O_B and O_C for three strategies that can achieve the setting values (S_A, S_B, and S_C), respectively. Based on the operation points, we obtain the set of desired biological parameters.

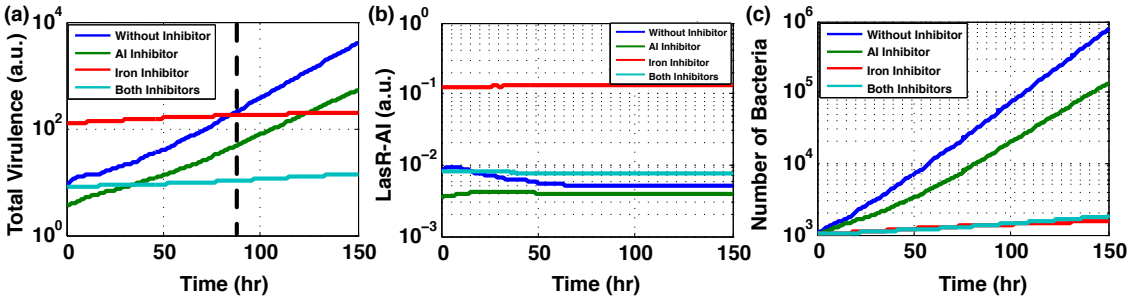

Figure 2.6: The simulation results for (a) TV (b) concentration of LasR-AI complex (c) number of bacteria for four different scenarios. Note that TV is the product of the concentration of LasR-AI complex and the number of bacteria as shown in (2.23). We observe that the multi-inhibition strategy is the most effective in reducing TV.

2.5.3 Simulation results

In this section, we validate the proposed control system by using a 3D microfluidic environment agent-based simulator [51]. First, we explicitly apply the cellular-level model to each agent (bacterium). Next, we consider several physical and stochastic effects (phys-

ical interactions between bacteria, variation in the QS systems, growth model, *etc.*) and examine the growth and virulence of populations of bacteria.

The environmental configurations used in these simulations are presented in the **Appendix**. As shown in Fig. 2.6(a), for the case without inhibitors, the values of TV surpass the other strategies after 70 hrs of cultivation (*i.e.*, the time needed to grow bacteria in the wet-lab); this is because the bacteria growth rate (μ_X in (2.21)) without inhibitors is larger compared to inhibitor schemes (Fig. 2.6(c)). If we use AI inhibitors alone, the concentration of LasR-AI complex is reduced, but this can not repress the growth of bacteria. On the contrary, the iron inhibitor alone can inhibit the bacteria growth but the LasR-AI concentration increases (Fig. 2.6(b)). The multi-inhibitor strategy shows the best results; indeed it can lower the concentration of LasR-AI and bacteria growth simultaneously.

2.6 Conclusion

In this chapter, we have proposed an autonomous optimal controller that incorporates the bacteria QS regulation and growth models and operates within a synthetic cell. By analyzing the system characteristics through numerical methods and simulations, we have shown that such synthetic cells can control the expression level of QS signals and cells growth.

We have also formulated a dynamic optimization problem to design the biological parameters of the proposed controller; this provides general guidelines to synthesize such optimal controllers *in vitro*. The proposed autonomous controlled system represents a first step towards a paradigm change in controlling the dynamics of communicating bacteria. Notice that our target for controlling the cell growth and virulence is not restricted to iron availability only. Our model can be extended to other substances if they are found to be negatively related to the QS system.

Chapter 3

Towards Cell-based Therapeutics: A Bio-inspired Autonomous Drug Delivery System

This chapter presents the autonomous and adaptive bacteria-based drug delivery system that integrates bacterial chemotaxis and quorum sensing in order to deliver drugs efficiently and precisely at various location in the human body (Fig. 3.1). More specifically, we first model bacterial chemotaxis and design a synthetic AND gate that enables bacteria to detect molecules produced by tumors and release the appropriate drugs in a coordinated manner; the system can also dynamically adjust the amount of drugs released based on tumor size and activity level. Next, we simulate and show the proposed system can be effectively used for cell-based therapeutics while preventing drug overuse and multi-drug resistance.

3.1 Introduction and motivation

Traditional drug delivery systems like injecting drugs into blood vessels are often inefficient as they rely on diffusion processes to deliver drugs. Indeed, diffusion makes these systems imprecise spatially (*i.e.*, with respect to the target location) and in terms of dosage [52]; this can lead to drug overuse and multi-drug resistance [53][54]. To address both problems, targeted cell-based therapies have gained attention recently [55]. This type of treatment interferes directly with specific cell molecules required for tumor growth, rather than indiscriminately targeting malignant and nonmalignant cells as is the case in traditional chemotherapy [56]. For instance, Alexander-Bryant *et al.* in [57] introduce strategies for designing targeted cancer therapies. They propose to deliver a high dose of anticancer drugs directly to the cancer tumor, while minimizing the drug uptake by nonmalignant cells. However, the problem of how to efficiently deliver the drugs to the target location remains unsolved.

In recent years, targeted drug delivery has been actively studied and a number of mathematical models have been already proposed. For instance, in [58], the authors

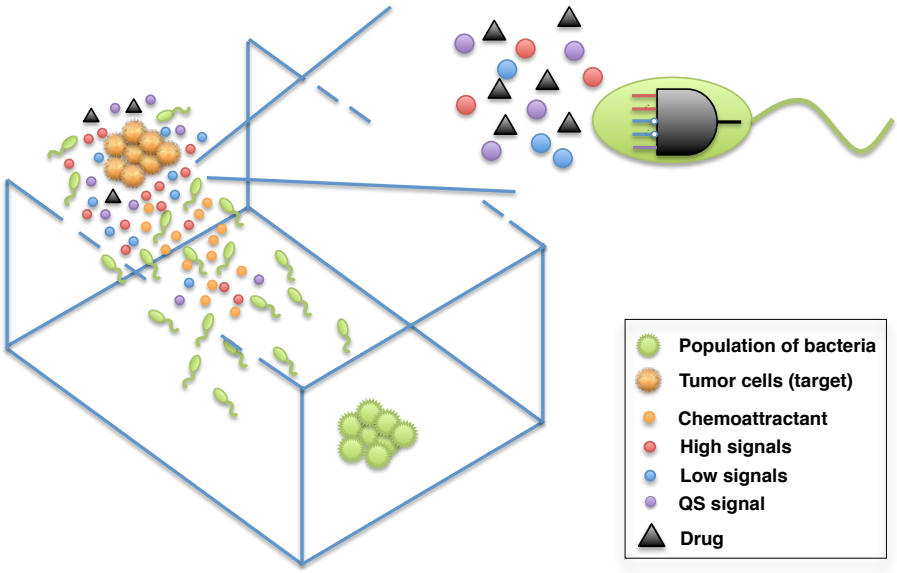

Figure 3.1: Proposed drug delivery system: The green circles represent clusters of bacteria released at an arbitrary (initial) location. The big orange circles represent the tumor cells that release chemoattractants (shown with small orange circles) into the environment. Bacteria (shown with green) sense the the concentration change of chemoattractants and start accumulating around the tumor. Once bacteria reach a quorum and match the High and Low signals released by the tumor (red and blue particles), the QS signal (purple particles) turns on the drug release circuit (embedded inside the bacteria) which stimulates the production of drugs (black triangles).

propose to design micro-robots with a rotating helical tail in order to prevent invasive drug delivery when swimming in a viscous fluid. However, a more bio-compatible approach is to use bacteria as bio-robots designed to perform pre-engineered tasks. A model of using bacterial network to move bacteria toward the target locations has been proposed in [59]. The authors also provide some statistical analyses to quantify the performance of the drug delivery process. As shown, such a bacteria-based drug delivery process can be a perfect candidate because bacteria can be engineered to navigate to the target location. However, to the best of our knowledge, a mechanism by which bacteria can be engineered to collectively release a precise amount of drugs at the target location, in an adaptive manner has not been developed yet. Consequently, as one of our main contributions, we propose to design the genetic circuitry that can sense the tumor related signals and then release drugs precisely to the target in an adaptive and coordinated fashion.

Getting now into more details, one of the advantages of using bacteria as our drug delivery vehicle is that bacteria can sense multiple types of signals and make decisions through complex regulatory pathways. One such instance is the regulation of bacterial movement known as chemotaxis. In a heterogeneous environment, bacteria can sense the concentration of the nutrient (*i.e.*, chemoattractants) and control the rotation of the flagella which decide the direction of movement [60]. Additionally, it was discovered that tumor cells can release certain types of nutrients that can be recognized by bacteria [59].

From this perspective, we propose to utilize bacteria as *vehicles* to deliver drugs at the target locations where various tumors may be located. The specific receptors for bacterial chemotaxis (anchored on the cell membrane) can bind to chemoattractants; the newly formed complexes can then directly regulate the chemotaxis pathway and drive the movement of bacteria [61]. Therefore, by detecting the concentration gradient of chemoattractants [62], it becomes possible to move and accumulate bacteria at any target location [63]. To effectively transport the drug, we consider a dense network of interacting bacteria that can utilize chemotaxis locally and move in formation. As demonstrated in [59], bacteria can aggregate to form clusters; this can be exploited for precise drug delivery.

In addition to moving bacteria toward a target location, another important issue is the timing of releasing the drugs. A naive approach is to make bacteria release the drugs as soon as they reach the target. However, since not all bacteria can reach the target at the same time, drugs may be released at different times; consequently, the proper drug dosage may actually not be delivered to the target. Hence, it is extremely important for bacteria to collectively release the drugs only after they all reach the target location. To handle this requirement, we propose to utilize a well-known cell-cell communication mechanism [64] [65] known as quorum sensing (QS) [31] in order to coordinate the collective behavior of bacteria. Indeed, by sensing the concentration of specific types of the molecules, bacteria can make decisions in a coordinated fashion, after reaching a certain cell density threshold [66]. Hence, we propose to integrate the QS signal as a control input to the drug delivery circuitry; Fig. 3.1 shows our proposed drug delivery system.

Our proposed drug delivery circuitry adaptively controls the amount of drugs being released [67] [68]. As shown in Fig. 3.3, by recognizing a specific molecular pattern of an active tumor, our proposed genetic circuitry can produce drugs as long as that molecular pattern is matched [69]. Indeed, once the molecular pattern is matched, bacteria start releasing precise amount of drugs based on the concentration of tumor-related signals. As drugs accumulate and start working, the tumor size starts to decrease, and thus concentration of these molecules decreases[1]. In order to prevent drug resistance, our proposed genetic circuit is able to detect such molecular changes in the environment and adjust the timing and the amount of drugs released dynamically.

In a real scenario, however, such as one involving obstruction due to blood vessels, different kinds of chemical substances can block the movement of bacteria. For instance, the extracellular matrix is abundant around tumors [70][71] and has a higher viscosity compared to the fluid in the environment. Therefore, we need to model such high viscosity regions as obstacles. Once bacteria enter such a region, their speed should significantly decrease. To evaluate the performance of the drug delivery system, we consider the time when bacteria reach the target location under various conditions including a varying number of obstacles, obstacles with movements, and different initial distributions of bacteria. The overall performance can be used as a guideline to engineer the drug delivery system.

In terms of design methodology, we first model the chemotaxis and QS regulatory pathway inside the bacteria. Next, we design the drug delivery genetic circuitry that

[1]We assume that the concentrations of tumor-related signals are proportional to the tumor size.

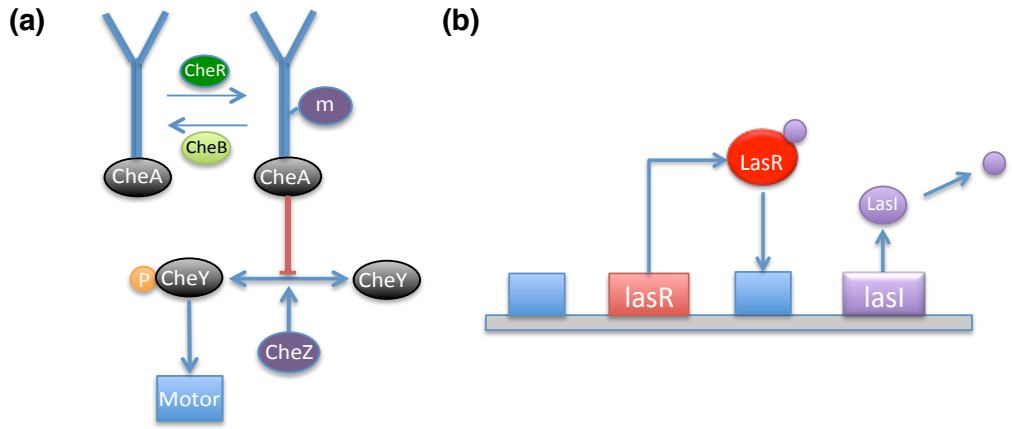

Figure 3.2: Regulatory pathways of chemotaxis and QS. (a) Bacterial chemotaxis: the blue fork is the signal receptor of the chemoattractant; it can phosphorylate the $CheY$ into $CheY_p$ and decrease the clockwise rotation in order to run. The $CheR$ and $CheB$ control the activity of receptor methylation (purple circle m). The $CheA$ and $CheZ$ function as regulators in order to balance the level of $CheY$ and $CheY_p$ (b) QS: the blue box represents the promoter binding regions. QS signaling molecules are expressed by $LasI$; once the bacterial cell density reaches a certain threshold, QS signaling molecules further combine with $LasR$ receptors that form an positive feedback loop.

takes its inputs from the environment (*i.e.*, tumor-released molecular pattern and QS signal). Lastly, we integrate bacterial chemotaxis and drug delivery circuitry to construct an *autonomous* drug delivery system and demonstrate its effectiveness via detailed simulations [51][72].

3.2 Mathematical modeling of a drug delivery system

3.2.1 Bacteria chemotaxis model of *E. coli*

The bacterial chemotaxis regulatory pathway contains several interactions among chemical substances (Fig. 3.2(a)). The chemical substance that determines the movement of bacteria is the phosphorylated $CheY$ [2]. Once the concentration of $CheY_p$ is repressed, bacteria start to run with slight tumbling. Therefore, as long as bacterial receptors (located on the cell membrane) bind to the chemoattractants, they will induce the production of $CheA$ that can repress $CheY_p$ and bacteria can move towards the target. Following

[2]Denoted as $CheY_p$ in this thesis.

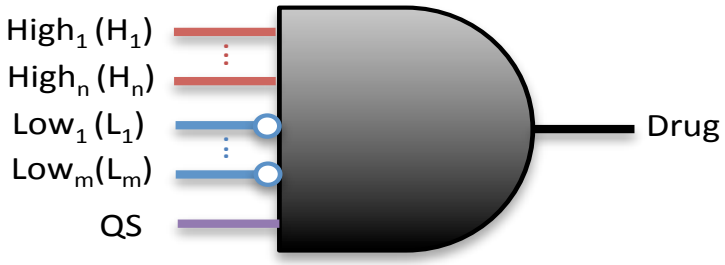

Figure 3.3: The newly proposed drug release circuitry embedded in the green bacteria in Fig. 3.1: The multi-input AND gate can sense n High signals, m Low signals, and the QS signal. When all the molecular pattern is met, the drug release circuitry starts releasing drug (medicine that targets the tumor cells).

work in [73], this process can be modeled as follows:

$$F = \epsilon_0 + log(\frac{1 + [C]/K_{on}}{1 + [C]/K_{off}}) \tag{3.1}$$

$$A = \frac{1}{1 + e^F} \tag{3.2}$$

$$Y_p = CheY_{tot} \times A \tag{3.3}$$

where A represents the average activity of the receptors in the cluster, and F is the sum of the energy difference between the receptor on and off states. [C] denotes the concentration of the nutrient, and $CheY_{tot}$ is the total available $CheY$ inside the bacteria. ϵ_0, K_{on} and K_{off} are constants. Here, we use the definition of $G_0(CheY_p)$, $k-$ and $k+$ from reference [74] to determine whether bacteria should run or tumble:

$$G_0(CheY_p) = \frac{g_0}{4} - \frac{g_1}{2}(\frac{CheY_p}{K_D + CheY_p}) \tag{3.4}$$

$$k^+ = w_0 exp(G_0(CheY_p))$$

$$k^- = w_0 exp(-G_0(CheY_p)) \tag{3.5}$$

where k^+ and k^- denote the transition rates from clockwise (CW) to counter-clockwise (CCW), and CCW to CW, respectively; parameters w_0, g_0, g_1 are all constants. When flagella turn CCW, bacteria swim straight (this is a run), whereas when they turn CW, bacteria tumble [67]. Using chemotaxis, bacteria can swim and locate the tumor.

3.2.2 Drug release circuitry

Bacteria can reach a quorum once the cell density reaches the activation threshold; this way once bacteria reach the target location and reach a quorum, the drug delivery circuitry starts releasing the drugs based on the concentration of the tumor related signal. Tumor cells can release a particular molecular pattern that distinguishes one tumor from another. Therefore, we integrate QS signal and this specific pattern as inputs to the drug delivery circuitry.

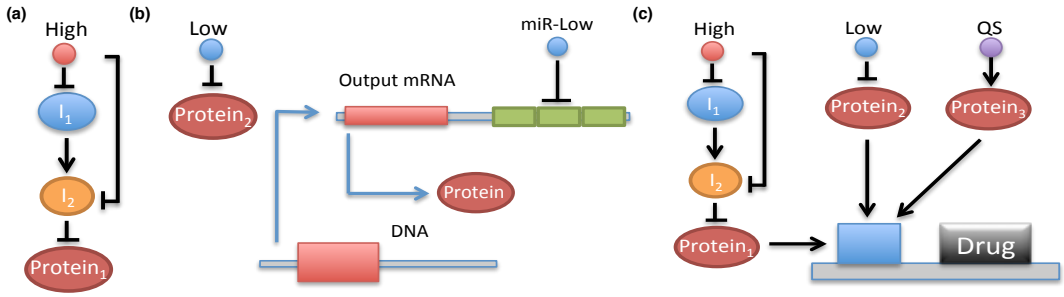

Figure 3.4: Schematic of the proposed genetic circuitry. (a) The double inversion circuit takes High signals as its input and first represses the intermediate product l_1; l_2 represses the output protein. (b) The inversion circuit takes Low signals as its input and directly represses the protein (Adapted from [69]). (c) The resulting drug delivery circuitry (Fig. 3.3) by integrating (a), (b) and QS signal. When the output proteins are all highly expressed, the AND gate starts producing drugs.

As shown in Fig. 3.3, we assume that the tumor cells show a molecular profile that contains n different High ($H_{1:n}$) concentration molecules, m Low ($L_{1:m}$) concentration molecules, and a QS signal. For example, the molecular pattern in [69] targets the Hela cell [3] by recognizing its particular molecular pattern (Hela-High and Hela-Low signals).

We integrate these three different types of chemical signals into a synthetic AND gate since drugs need to be released only when all the molecular patterns are met. We use a double inversion circuit to take in $H_{1:n}$ as inputs and produce intermediate signals that induce the production of the drug (Fig. 3.4(a)). The double inversion circuit is needed because the concentrations of $H_{1:n}$ are too low to directly activate the drug circuitry. Hence, the intermediate step of the double inversion circuit serves as a buffer meant to amplify the input concentration. For the low concentration signal $L_{1:m}$, we use a simple inversion circuit (Fig. 3.4(b)) to invert the input and produce the intermediate signal.

A general inversion genetic circuit can be described by the following equations [49]:

$$\frac{d[mRNA]}{dt} = r_0 - \delta[mRNA] - [Y]\frac{V[mRNA]}{K + [mRNA]} \tag{3.6}$$

$$\frac{d[Protein]}{dt} = P[mRNA] - \delta[Protein] \tag{3.7}$$

where $[X]$ denotes the concentration of a particular molecular species, Y represents the molecules released by the tumor which serves as the input to the genetic circuit ($H_{1:n}$ or $L_{1:m}$ in Fig. 3.4). $Protein$ represents the concentration of a certain kind of protein that can trigger the production of drugs. P and r_0 represent the promoter strength and basal production rate, respectively. Finally, δ is the degradation constant.

Eq. (3.6) describes the transcription of mRNA which is directly regulated by the concentration of the input molecules through the Michaelis-Menten kinetics. Eq. (3.7) describes the translation of the mRNA into protein with the direct regulation of the mRNA.

[3] One type of cancer cells.

Representation	Parameters	Value
Basal production rate	c_A, c_R	1×10^{-4}
Maximum production rate	k_A, k_R	2×10^{-3}
Michaelis-Menten constant	K_A, K_R	1×10^{-6}
Association (dissociation) rate	k_0, k_3	1×10^{-2}
Association (dissociation) rate	k_1, k_2, k_4, k_5	1×10^{-1}
Basal production rate	r_0	1×10^{-3}
Degradation rate	δ	2.5×10^{-1}
Promoter strength	P	1×10^{2}
Michaelis-Menten constant	K_H, K_{QS}	1×10^{-2}

Table 3.1: Model parameters for QS and drug delivery circuitry [3].

The QS regulation system produces the QS molecules needed to set the AND gate output to 1 (see Fig. 3.4(c)). By combining these three types of signals into a synthetic AND gate, we propose the following equation:

$$\frac{d[D]}{dt} = P\left(\frac{K_H}{K_H + [H]}\right)^{n_H} [L]^{n_L} \frac{[QS]}{[QS] + K_{QS}} - \delta[D] \tag{3.8}$$

where D represents the concentration of the drug, L is the concentration of the Low signal, H is the concentration of High signal and QS is the concentration of the QS signal. K is the disassociation constant and the subscript (H and L) corresponds to each type of species. n denotes number of types of molecules for both the categories (H and L). In our case, the molecular release profile of the target tumor can be recognized as one repressor and two activators. Hence, n_L equals one and n_H is two.

3.3 Simulation without obstacles

In this section, we first describe the simulation environment and parameters for the drug delivery system. Next, we describe the simulation results with respect to both chemotaxis and drug delivery.

3.3.1 Simulation setup and parameters

We consider a 3D lattice with length 200 μm, width 200 μm, and height 200 μm. All bacteria (with their initial swimming directions randomly generated) are released from the transmitter group located at (50, 50, 50) μm of the 3D space. Of note, we assume bacteria can release chemoattractant locally and utilize this local attraction force to move in formation.

In the following simulations, the local group size of bacteria is set to 50. Under these conditions, bacteria swim towards the target location (an aggregate of tumor cells) located in the center of the 3D space, *i.e.*, (100, 100, 100) μm. The tumor cells start emitting chemoattractants in the environment when bacteria are released. Bacteria swim with a constant speed of 10$\mu m/s$ [59]. The parameters for QS and drug release circuit are shown in Table 3.1.

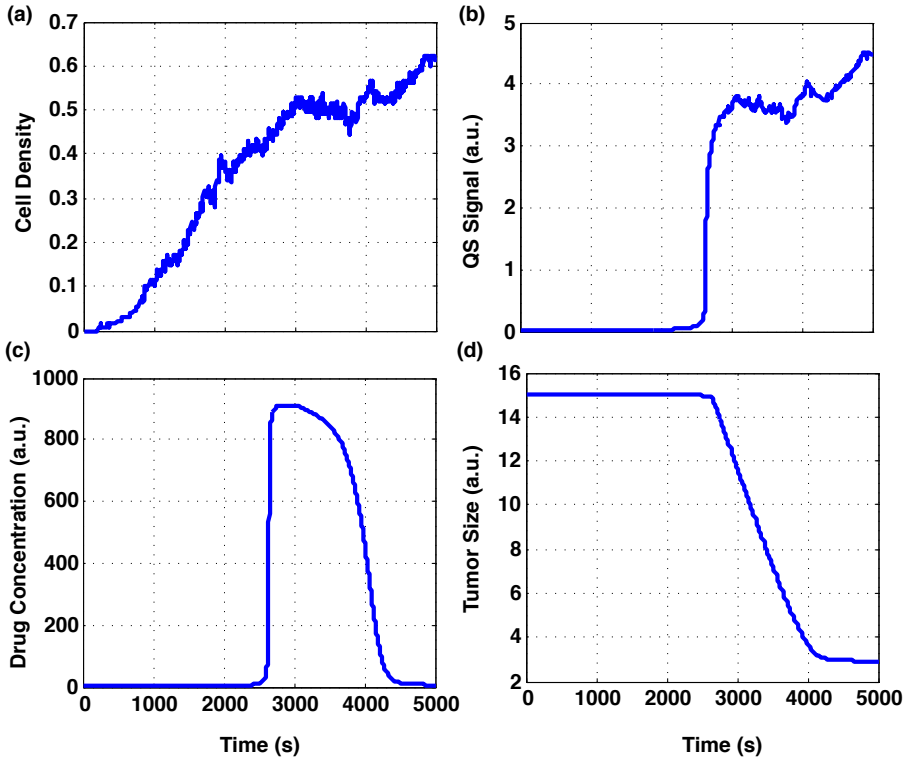

Figure 3.5: Drug delivery system responses. (a) Bacteria cell density variation: At time of 2500s, bacteria reach a cell density value around 0.4. (b) QS signal: Bacteria start to release QS signal once the cell density reaches 0.4 (around 2500s). (c) and (d) Drug concentration and tumor size variation. QS signal activates the drug delivery circuitry. As tumor size decreases, the concentration of drugs released by the circuitry also decrease in order to precisely control the amount of drugs. This prevents drug overuse and longer term drug resistance.

3.3.2 Drug delivery system

We implemented a four-input AND gate including two High signals (H_1 and H_2), one Low signal (L_1), and one QS signal. At time $t = 0$s, we place 5000 bacteria (100 groups each with 50 bacteria) at the origin (*i.e.*, (0,0,0) in 3D coordinates) while the tumor is located at location (100, 100, 100); bacteria then use chemotaxis to reach the target site.

We measure the cell density as the ratio of the total volumes occupied by the bacteria divided by the total volume around the tumor[4]. Fig. 3.5(a) shows the cell density in the vicinity of the tumor as a function of time. We design the activation threshold around 0.4 to trigger the drug release circuitry. For our experiments, the cell density reaches the threshold of 0.4 at time $t = 2500$s.

Fig. 3.5(b) shows the QS signal switching *on* once the cell density goes above the threshold; this change can be attributed to change in concentration of extracellular QS signaling molecules. Further, because the tumor starts at its full size initially, we assume

[4]The range of cell density is within [0 1].

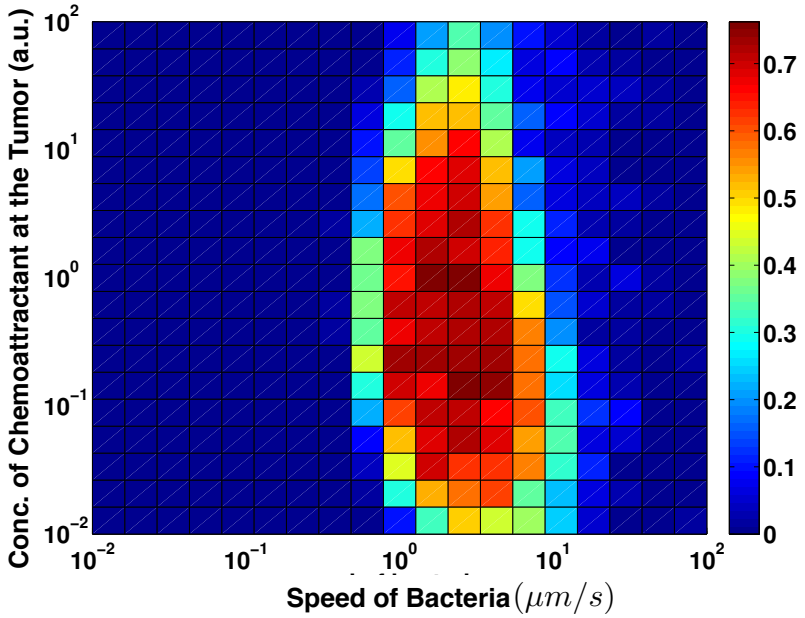

Figure 3.6: System robustness evaluation using bacteria speed and chemical gradient as variables. The color bar represents the cell density. Green and red colors indicate feasible regions where cell density exceeds the activation threshold of the drug release circuitry (we design the activation threshold to be 0.4).

that the initial concentration of H_1 and H_2 equals to 1 and L_1 is 0. The output of the AND gate will change from low to high at time $t = 2500$s. This is illustrated in Fig. 3.5(c).

We observe that bacteria do not release drugs as soon as they reach the target site, rather they wait until they reach a quorum and then collectively release the drug until the tumor size is reduced. This coordinated behavior of bacteria makes the drug release protocol more effective. This also shows how the circuit automatically connects the chemotaxis module with the drug release protocol: it waits until the majority of bacteria reach the tumor using chemotaxis and only then generates a response.

Finally, Fig. 3.5(d) shows the reduction in tumor size due to the drug. As the tumor size decreases, we assume the concentration of molecules released by the tumor also decreases which serves as a feedback to the drug delivery circuitry (Fig. 3.3); in turn, this reduces the amount of drug being released. This effect can be easily observed from Fig. 3.5(d). The drug concentration gradually reduces to zero as the tumor size becomes negligible. Intuitively, this means that our drug delivery system releases precise quantities of the drug and hence prevents any kind of drug resistance or drug overuse.

3.3.3 System robustness

Characterizing the robustness of the system is crucial since the uncertainty in biological parameters and the stochastic behavior of bacteria can cause undesirable system responses. Hence, we need to determine the range of biological parameters for which the

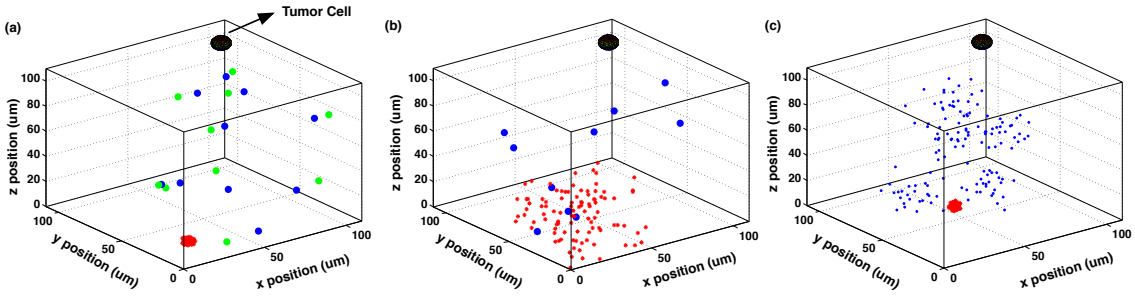

Figure 3.7: Different simulation configurations. (a) Bacteria clustered at the origin and large size obstacles randomly distributed in space. Blue circles represent (initial) spatial distribution of obstacles, while green circles represent the moving obstacles after certain timestamps. (b) Bacteria randomly distributed. (c) Small size obstacles distributed based on a Gaussian distribution.

drug delivery system can function well. Note that the drug delivery system can not deliver the drug if the bacteria do not reach a certain cell density needed to activate the QS system. Therefore, the bacterial chemotaxis plays a critical role and should be examined carefully.

We first focus on robustness analysis of the chemotaxis system. We assume that the amount of bacteria released at the origin is large enough and, thus, the mean-trajectories of bacteria can represent the movement of the entire population. There are two major factors that can affect the bacterial chemotaxis. One is the intrinsic speed of bacteria movement; the other is the concentration gradient of the chemoattractant. Hence, we vary the value of these two parameters and observe different combinations that can affect the system robustness.

As shown in Fig. 3.6, the cell density can reach values above the activation threshold of the QS when these two variables are within a certain range (the red region of Fig. 3.6). It is worth noticing that the speed of bacteria plays a key role compared to the concentration gradient. When the speed of bacteria is too high, bacteria first follow the concentration gradient. However, after several steps, bacteria can not sense the concentration difference of chemoattractants due to their high speed; this results in random trajectories. On the contrary, if the speed of bacteria is too low, it may take a longer time for bacteria to reach the target. Hence, this is not efficient from the drug delivery standpoint.

3.4 Simulation with obstacles

3.4.1 Simulation setup and parameters

In this section, we examine the effect of adding multiple obstacles to the simulation environment. This is a realistic scenario to consider since obstacles such as tissues and extracellular matrix are common in blood vessels. Extracellular matrix, for instance, can trap bacteria due to its high viscosity. Therefore, we assume that if bacteria enter in a region with obstacles, their moving speed is significantly reduced.

Scenario	Average of hitting times (s)	Variance of hitting times	Skewness of hitting times
(0, Not Applicable, O)	2.130×10^3	3.257×10^4	0.269
(5, F, O)	2.795×10^3	1.416×10^6	2.897
(10, F, O)	4.199×10^3	1.100×10^7	2.636
(15, F, O)	6.280×10^3	2.939×10^7	2.003
(135, F, O)	2.697×10^3	1.076×10^6	3.249
(270, F, O)	3.437×10^3	2.833×10^6	2.648
(405, F, O)	4.582×10^3	8.784×10^6	1.820
(810, F, O)	7.437×10^3	2.053×10^7	1.469

Table 3.2: Statistical moments of hitting times for different obstacle sizes under the same obstacle distributions. The triplet for each scenario corresponds to the number of obstacles, moving or fixed obstacles, and the spatial distribution of bacteria. "F" means fixed obstacles and "O" represents bacteria clustered at the origin.

We consider two different sizes of obstacles, namely, obstacles as large as the tumor (*i.e.*, $30\mu m \times 30\mu m \times 30\mu m$) (Fig. 3.7(a) and (b)) and small obstacles of size $10\mu m \times 10\mu m \times 10\mu m$ (Fig. 3.7(c)). We assume that within the region with obstacle, the bacteria speed decreases ten times compared to their usual speed (10 μm/s). Additionally, we consider the the obstacles either follow an uniform or a Gaussian distribution in the environment and then quantify the time when a sufficient number of bacteria reach the target location; this represents the hitting time (or first passage time) and can be regarded as a performance metric for a targeted drug delivery.

3.4.2 Statistical measurements

We measure the hitting times for different configurations along with the baseline scenario *i.e.*, without obstacles. We vary: (1) the number of obstacles and their spatial distributions, (2) the initial spatial distribution of bacteria, and (3) the movement of obstacles. More specifically, when the obstacles are of large size, we randomly distribute them based on an uniform distribution; we need a relatively small number of obstacles (namely, 5, 10, 15), in order to cover the entire simulation space. On the other hand, obstacles of small size follow a Gaussian distribution; they need to be more abundant in order to have similar space coverage.

For the initial (spatial) distribution of bacteria, we consider two scenarios: (1) uniform random distribution and (2) clustered (at the origin) distribution. For the dynamics of obstacles, we consider that they are either static (at the initial position) or perform a random walk as a consequence of drifting in the fluid. For each configuration, we estimate the probability that the hitting time exceeds a certain threshold which can be related to the degree of success of the drug delivery system. Fig. 3.7 shows different simulation configurations.

For completeness, we report in Tables 3.2, 3.3 and 3.4 the average, variance, skewness, and kurtosis of hitting times[5]. From Tables 3.2, 3.3 and 3.4, one can observe that the hitting times exhibit a positive skewness which has two implications: First, the tail of the distribution of hitting time is longer towards the right direction (corresponding to larger

[5] The skewness and kurtosis reflect the asymmetry in the hitting times distribution.

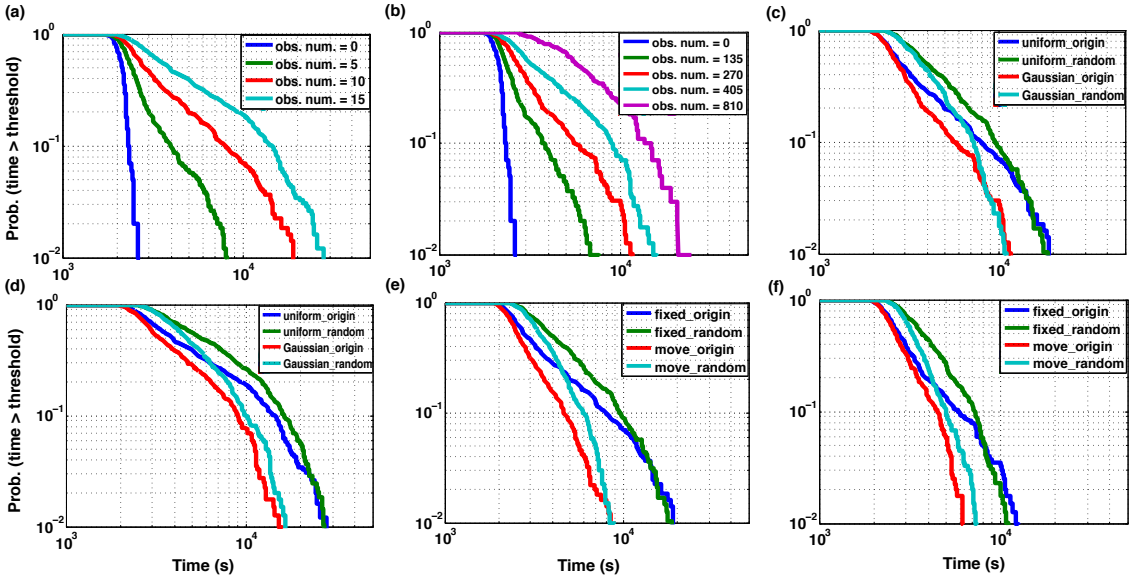

Figure 3.8: The probability distribution of hitting time to exceed a certain threshold as a function of different configurations. We consider the impact of obstacles number, spatial distribution of obstacles and bacteria, and the movement of bacteria. (a) Obstacle numbers equal to 5, 10 and 15 and follow uniform distribution. (b) Obstacle numbers equal to 135, 270, 405 and 810 and follow a Gaussian distribution. For both (a) and (b), bacteria are initially located at the origin. (c) and (d) Bacteria with two different initial spatial distributions (located at the origin or randomly distributed). Obstacles follow two different spatial distributions (uniform and Gaussian) under same spatial coverage. (e) and (f) Obstacle number equals to 10 (under uniform distribution) and 270 (under Gaussian distribution), respectively. For both (e) and (f), bacteria are either located at the origin or randomly distributed and obstacles either move or remain fixed.

hitting times), while towards the left hand side the tail is shorter *i.e.*, shorter hitting times. Second, since the mass of the distribution is concentrated towards the right hand side, one cannot ignore the chances of seeing much larger hitting times.

3.4.3 Impact of number of obstacles

We now quantify the effect of obstacles number. From Table 3.2, it can be observed that the average hitting times increase when the number of obstacles increases (from 20% to 160%), if all other conditions are fixed. We notice that with a greater number of obstacles, the probability of having very large hitting times gets larger (see Fig. 3.8(a) and (b)). However, the distribution of hitting times is less skewed; This is because the distribution does not have apparent peaks and is more similar to a Gaussian distribution. On the contrary, for the scenario of less obstacles, the distribution of hitting times is more skewed and has a longer tail; therefore, the probability of having larger hitting times is lower.

We also consider a scenario in which obstacles number is huge and cover the space up to 40%. As shown in Fig. 3.8(b), the purple line (obstacles number equal to 810) has

Scenario	Average of hitting times (s)	Variance of hitting times	Skewness of hitting times
(10, F, O)	4.199×10^3	1.100×10^7	2.636
(10, F, R)	5.240×10^3	1.104×10^7	1.829
(270, F, O)	3.437×10^3	2.833×10^6	2.648
(270, F, R)	4.341×10^3	3.571×10^6	1.593
(15, F, O)	6.280×10^3	2.939×10^7	2.003
(15, F, R)	7.903×10^3	3.237×10^7	1.470
(405, F, O)	4.582×10^3	8.784×10^6	1.820
(405, F, R)	5.754×10^3	9.831×10^6	1.616

Table 3.3: Statistical moments of hitting times for various spatial distributions of bacteria and obstacles. The triplet for each scenario corresponds to the number of obstacles, moving or fixed obstacles, and the spatial distribution of bacteria. “F” means fixed obstacles. “O” and “R” represent bacteria clustered at the origin and randomly distributed, respectively.

a less skewed hitting times distribution but the average hitting times is about two times longer when space coverage reduces to 20%.

3.4.4 Impact of spatial distribution of obstacles

As discussed above, small size obstacles are assumed to follow a Gaussian distribution. The mean of the Gaussian distribution is randomly generated and the standard deviation is chosen to match the size of larger obstacles in order to have same space occupations. As shown in Table 3.3 and Fig. 3.8(c) and (d), uniformly distributed obstacles (*i.e.*, larger sizes) have similar average and skewness of hitting times when other conditions are fixed. This shows that the spatial distribution of obstacles has little effect on the distribution of hitting times which results from similar spatial coverages of obstacles even if they follow different spatial distributions.

3.4.5 Impact of spatial distribution of bacteria

In this section, we investigate the impact of spatial distribution of groups of bacteria in the simulation space. A group of around 50 bacteria is formed by the local forces resulting from the chemoattractants expressed by other bacteria. To uniformly distribute bacteria in space, one can utilize repulsion forces to separate groups of bacteria; this can be achieved by engineering the bacteria regulatory pathway of chemotaxis to respond to a certain kind of chemorepllent. However, the chemo-repulsion forces among clusters of bacteria should be weaker than the chemo-attraction force within the group because we need to ensure the separation of groups of bacteria while still maintaining the local clustering. After arriving at the target location, the chemo-repulsion forces should be stopped by switching off the regulatory pathway of the chemotaxis.

As shown in Table 3.3 and Fig. 3.8(c) and (d), in general, different spatial distributions of bacteria have a similar performance in terms of average of hitting times. The main difference is in the skewness of hitting times. The randomly distributed case has a lower skewness because the probability of bumping into obstacles is lower than clustered (at the origin) distribution; this implies that the probability of successfully delivering drugs is higher when bacteria are randomly distributed in the space.

Scenario	Average of hitting times (s)	Variance of hitting times	Skewness of hitting times
(10, F, O)	4.199×10^3	1.100×10^7	2.636
(10, M, O)	3.045×10^3	1.722×10^6	2.660
(10, F, R)	5.240×10^3	1.104×10^7	1.829
(10, M, R)	3.920×10^3	2.089×10^6	1.909
(270, F, O)	3.437×10^3	2.833×10^6	2.648
(270, M, O)	2.954×10^3	9.449×10^5	1.650
(270, F, R)	4.341×10^3	3.571×10^6	1.593
(270, M, R)	3.574×10^3	1.184×10^6	1.718

Table 3.4: Statistical moments of hitting times for the effects of obstacle movements. The triplet for each scenario corresponds to the number of obstacles, moving or fixed obstacles, and the spatial distribution of bacteria. “F” represents fixed obstacles at the initial location, while “M” indicates obstacles perform random walks. “O” and “R” represent bacteria clustered at the origin and randomly distributed, respectively.

Additionally, when bacteria initially clustered at the origin of the 3D space, the distribution of hitting times exhibits heavy tails. However, the baseline scenario shows that bacteria start from the origin have a better performance than the randomly distributed one; this implies that obstacles degrade the performance of drug delivery system heavily.

3.4.6 Impact of moving obstacles

We assume now that the movement of obstacles follows a random walk *i.e.*, obstacles move back and forth around the initial locations; this property results in shorter average values of hitting times since the probability of trapping bacteria within the obstacle location decreases.

As can be seen in Table 3.4 and Fig. 3.8(e) and (f), when obstacles move freely in the simulation space (other conditions being fixed), the average of hitting times decreases compared to the scenario of fixed obstacle locations; this is because the obstacles can no longer trap bacteria and bacteria regain the normal speed to move toward the target location. However, the movement of obstacles has little effect in terms of the skewness of hitting times. Hence, as shown in Fig. 3.8(e) and (f), the shapes of the hitting time distributions for each scenario are similar.

3.4.7 Summary of results derived in the presence of obstacles

As a summary of these investigations, the presence of obstacles increases the hitting times in the range of 20% to 160%; this greatly affects the overall drug delivery performance. In general, when the number of obstacles increases, the average of hitting times increases too because of the probability of bumping into obstacles also gets higher.

We also find that the spatial distribution of obstacles has indistinguishable effects on the distribution of hitting times. On the contrary, randomly distributed bacteria can have better performance in terms of the skewness of hitting times. Finally, moving obstacles can decrease the average of hitting times since the probability of trapping bacteria decreases.

Overall, the presence of obstacles degrades the performance of our proposed drug delivery system. Our complete analyses above can provide strategies to enhance the performance of the drug delivery system. For example, since randomly distributed bacteria have better performance in terms of hitting times, we can engineer bacterial chemotaxis to achieve this desired goal.

3.5 Conclusion

In this chapter, we have proposed a cell-based therapeutic approach that utilizes bacteria to deliver drugs autonomously and adaptively. To this end, we have designed a new drug delivery system such that it automatically integrates the chemotaxis and QS signals. We have further demonstrated the functionality of the drug delivery system that can work in an adaptive manner and reduce the tumor size dynamically.

Finally, we have shown that our proposed system satisfies two of the most important characteristics any drug delivery system should have, namely (i) locate the target precisely, and (ii) deliver precise quantities of drugs that decrease as the size of tumor reduces. This is not the case in the conventional drug delivery systems that use diffusion to transport drugs to any given location. Therefore, the proposed drug delivery system prevents unnecessary drug overuse and multi-drug resistance. Our simulation framework can help the synthetic biologists design such bacteria-based drug delivery systems.

Chapter 4

MPLasso: Inferring Microbial Association Networks Using Prior Microbial Knowledge

Microbial communities exhibit rich dynamics including the way they adapt, develop, and interact with the human body and the surrounding environment. The associations among microbes can provide a solid foundation to model the interplay between the (host) human body and the microbial populations. However, due to the unique properties of compositional and high-dimensional nature of microbial data, standard statistical methods are likely to produce spurious results. Although several existing methods can estimate the associations among microbes under the sparsity assumption, they still have major difficulties to infer the associations among microbes given such high-dimensional data. To enhance the model accuracy on inferring microbial associations, we propose to integrate multiple levels of biological information by mining the co-occurrence patterns and interactions directly from large amount of scientific literature. We first show that our proposed method can outperform existing methods in synthetic experiments. Next, we obtain credible inference results from Human Microbiome Project datasets when compared against laboratory data. By creating a more accurate microbial association network, scientists in this field will be able to better focus their efforts when experimentally verifying microbial associations by eliminating the need to perform exhaustive searches on all possible pairs of associations.

4.1 Introduction and motivation

Microbes play an important role both in environment and human life. However, the way microbes affect the human health remains largely unknown. Knowledge of the microbial interactions can provide a solid foundation to model the interplay between the (host) human body and the microbial populations; this can serve as a key step towards precision medicine [75]. Unfortunately, understanding microbes interactions is difficult, as most microbes cannot be easily cultivated in standard laboratory settings. However, the

recent increase of quality and reduced costs of sequencing technologies (*e.g.*, shotgun or PCR directed sequencing [76]) enable researchers to collect information from the entire genome of all microbes under different environment conditions. As a result, various datasets ranging from earth ecosystem to human microbiome have been made publicly available under the Human Microbiome Project [77] or the Earth Microbiome Project [78].

In this chapter, we aim at analyzing the networks of associations (putative interactions) among the microbes of human microbiome in order to understand how microbes can affect the human health. To this end, there exist several challenges: First, the amount of sequenced data that corresponds to human microbiome available from public websites is scarce. To date, one of the largest metagenomics datasets of human niches is the NIH Human Microbiome Project (HMP) [77] which only provides a few hundreds of healthy individual samples (n) of various body sites, while the number of measured microbes (p) usually ranges from hundreds to thousands. As a consequence, the number of associations ($p(p-1)/2$) is much greater than the number of samples (*i.e.*, high-dimensional data). Another big challenge stems from the nature of the data itself. Sequencing data only provides the *relative* abundance of various species; this is because the sequencing results are a function of sequencing depth and the biological sample size [79]. Therefore, from a statistical standpoint, the *relative* taxon abundance falls into the class of compositional data [80]; this causes statistical methods such as Pearson or Spearman correlations (which work with absolute values) to generate spurious results.

4.2 Prior work

To infer microbe associations for both compositional and high-dimensional data, several algorithms have been developed. A pioneering method called SparCC [81] applies log-ratio transform on compositional data and directly approximates the correlation among microbes based on sparsity assumption of microbial associations. However, SparCC does not consider the influence of errors in compositional data; this may reduce the correlation estimation accuracy. More precisely, SparCC approximates the basis variance (*i.e.*, the variance of compositional data) under the assumption that average correlations are small. Second, the iterative procedure used to estimate the magnitude of correlations can exceed value 1; this may cause poor approximations if one tries to remedy the problem by setting up the threshold value to 1 or -1 for the estimated correlations; these series of approximations may reduce the correlation estimation accuracy quite significantly. SPIEC-EASI [82] calculates the covariance of the log-ratio transformed data to approximate the covariance of the absolute abundance of microbes; then, it uses either neighborhood selection (mb) [83] or graphical Lasso (gl) [84] to estimate the conditional dependencies among microbes. CCLasso [85] is similar to SPIEC-EASI which applies log-ratio transform on compositional data and imposes a L_1 penalty on the inverse covariance matrix of the microbes and then solves it to obtain a sparse covariance matrix. However, it is not clear whether or not CCLasso can obtain a consistent estimator on the inferred microbial covariance without showing consistency analysis. (run time comparisons of existing methods are summarized in **Appendix** 8.5 and Fig. 8.3).

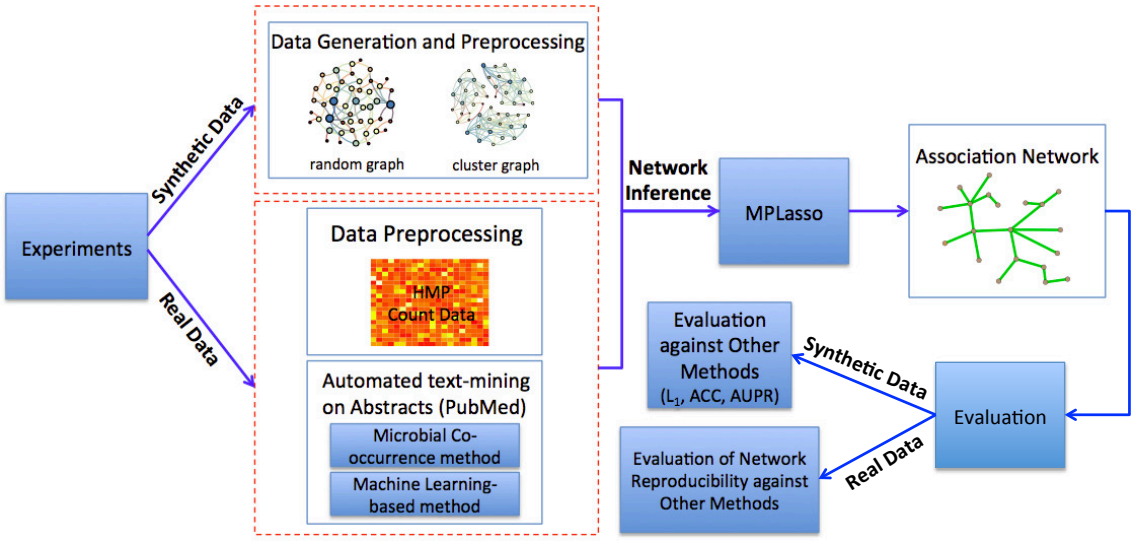

Figure 4.1: Our proposed framework of inferring microbial association network. We conduct two different sets of experiments, namely, synthetic and real data. For the synthetic experiment, we generate data based on different graph structures and evaluate the performance of our proposed algorithm by using three performance metrics (*i.e.*, L_1, ACC, and AUPR (see section 4.6.2.)). For the real data experiments, the prior information is obtained through automated text-mining. Since there is no "gold standard" network to evaluate performance, we evaluate the reproducibility of inferred networks instead.

We note that although the above methods can estimate the covariance among microbes under the sparsity assumption, they still have major difficulties to infer the associations among microbes given such high-dimensional data. To solve the problem caused by high-dimensional data, we propose to integrate multiple levels of biological information to enhance the model accuracy on inferring microbial associations. Indeed, an increasing amount of scientific literature provides a large amount of data which can be mined not only for the co-occurrence of microbes, but also to predict microbes associations directly. For instance, pioneering work [86] considers automated analysis of the co-occurrence of bacterial species through statistical testing approaches (*e.g.*, Fisher's exact test). Recently, Lim et al. [87] incorporated machine learning techniques to automatically identify and extract microbial associations directly from the abstracts of scientific papers. Finally, Wang et al. [88] and Li et al. [89] use prior biological knowledge to reconstruct genes interaction networks.

To the best of knowledge, we are the first to consider experimentally verified biological knowledge as *a priori* information to derive microbial association networks. To this end, we transform the original problem of microbial associations estimation into a graph structure learning problem where nodes represent microbes and edges represent (pairwise) associations among microbes. With this new problem formulation, the graphical Lasso algorithm becomes suitable to infer the microbial association network. We also integrate the text mining results from the scientific literature as prior knowledge for inferring the microbes graph structure; the proposed algorithm *Microbial Prior Lasso (MPLasso)* turns

out to be more accurate than other existing methods on inferring the microbial associations. The proposed MPLasso pipeline is shown in Fig 4.1.

4.3 Acquisition and transformation of microbial count data

Cross-sectional data obtained from the human microbiome project (HMP) have a curated collection of sequence of microorganisms associated with the human body from both shotgun and 16S sequencing technologies. For the 16S rRNA data, we consider the high-quality sequencing reads in 16S variable regions 3-5 (V35) of HMP healthy individuals from Phase one production study (May 1, 2010). The taxonomy classification of the 16S rRNA are performed using either mothur (HMMCP) [90] or QIIME (HMQCP) [91] pipelines. The resulting table for operational taxonomic units (OTUs) at each body site of the human samples can be obtained from http://hmpdacc.org/HMMCP/ and http://hmpdacc.org/HMQCP/. For the shotgun data (HMASM), we obtain data from http://hmpdacc. org/HMASM/ and use the trimmed sequences as inputs to the metaphlan2 [92] pipeline which can generate the OTU abundance for each sample.

For both 16S and shotgun data, the obtained operational taxonomic unit (OTU) table can be represented by a matrix $\mathbf{D} \in \mathbb{N}^{n \times p}$ where $\mathbb{N}$ represents the set of natural numbers. $d^i = [d_1^i, d_2^i, \ldots, d_p^i]$ denotes the p-dimensional row vector of OTU counts from the i_{th} sample ($i = 1, \ldots, n$). To account for different sequencing depths for each sample, the raw count data (d^i) are typically transformed into *relative* abundances (x) by using log-ratio transform [80]. Statistical inference on the log-ratio transform of the compositional data (x) can be shown to be equivalent to the log-ratio transform on the unobserved absolute abundance (d) as: $\log(\frac{x_i}{x_j}) = \log(\frac{d_i/m}{d_j/m}) = \log(\frac{d_i}{d_j})$. Here, we apply the centered log-ratio (clr) transform as follows:

$$c = \mathsf{clr}(x) = [\log(\frac{x_1}{m(x)}), \log(\frac{x_2}{m(x)}), \ldots, \log(\frac{x_p}{m(x)})] \tag{4.1}$$

where $m(x) = (\prod_{i=1}^{p} x_i)^{\frac{1}{p}}$ is the geometric mean of the composition vector x. The resulting vector c is constrained to be a zero sum vector.

The covariance matrix of the clr transform $\mathbf{C} = \mathsf{Cov}[\mathsf{clr}(c)]$ can be related to the covariance matrix of the log-transformed absolute abundances $\mathbf{\Gamma} = \mathsf{Cov}[\log \mathbf{D}]$ via the relationship [80, 82] $\mathbf{C} = \mathbf{U\Gamma U}$, where $\mathbf{U} = \mathbf{I_p} - \frac{1}{p}\mathbf{J}$, where $\mathbf{I_p}$ is the p-dimensional identity matrix, and $\mathbf{J}$ is the p-dimensional all-ones vector. For the case where $p >> 0$, the finite sample estimator ($\hat{\mathbf{C}}$) serves as a good approximation of $\hat{\mathbf{\Gamma}}$; therefore, the finite sample estimator ($\hat{\mathbf{C}}$) serves as the basis on inferring the correlations among microbes. To account for the zero counts in samples, we add pseudo count to the original count data to avoid numerical issues when using the clr transform.

4.4 Proposed algorithm: Microbial Prior Lasso (MPLasso)

To infer the pairwise associations among microbes, we can transform the original inferring problem into a graph learning problem where each node represents an OTU (*e.g.*, taxon)

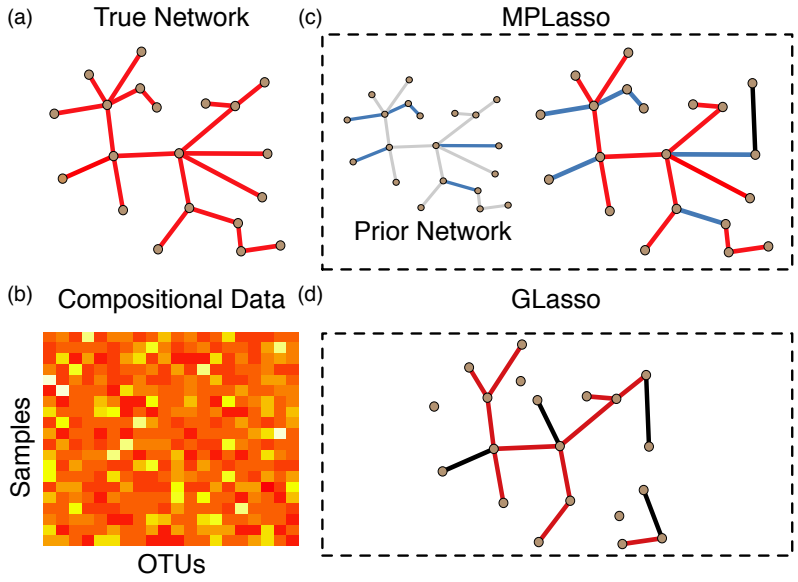

Figure 4.2: Comparison of our proposed MPLasso and graphical Lasso (GLasso) on inferring the same compositional data in a small example. (a) The edges of the true network are shown with red lines. (b) The entities of the compositional data matrix shown with denser colors represent higher values (c) Given the prior network where blue and black edges are correct and wrong information, respectively, the MPLasso can still accurately estimate the graph structure with one missing edge and only one wrongly estimated edge (black edge). (d) GLasso wrongly estimates several edges along with missing edges.

and each edge represents a pairwise association between microbes; the resulting graph is an undirected graph $\mathcal{G} = (V, E)$, where V and E represent the node and edge sets, respectively.

Suppose the observed data (d) are drawn from a multivariate normal distribution $N(d|\mu, \mathbf{\Sigma})$ with mean μ and covariance $\mathbf{\Sigma}$. The inverse covariance matrix (precision matrix) $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ encodes the conditional independence among nodes. More specifically, if the entry (i, j) of the precision matrix $\mathbf{\Omega}_{i,j} = 0$, then node i and node j are conditionally independent (given the other nodes) and there is no edge among them (*i.e.*, $E_{i,j} = 0$).

However, microbial data usually come with a finite amount of samples (n) but with high dimensionality (p); this makes the graph inferring problem intractable since the number of variables ($\frac{p(p-1)}{2}$) is greater than n. To solve this problem, an important assumption that needs to be made is to assume that the underlying (true) graph is reasonably sparse. One suitable algorithm to select the precision matrix under sparsity assumption is to utilize the graphical Lasso proposed previously [84, 82].

As shown in Fig 4.2, we propose to utilize the information obtained from the scientific literature in order to construct the prior matrix $\mathbf{P} \in \mathbb{R}^{p \times p}$, where each entry $\mathbf{P}_{i,j} \in [0, 1]$ represents the prior probability of associations between taxon i and taxon j. We can impose different amounts of penalties on the precision matrix; this is different from the standard formulation where the penalty (ρ) imposed on the precision matrix is the same. Therefore, by incorporating the prior information into the penalty matrix ($\mathbf{P}$), the proposed

MPLasso can be formulated as follows:

$$\hat{\mathbf{\Omega}} = \arg\max_{\mathbf{\Omega}} \{\log \det(\mathbf{\Omega}) - \text{tr}(\mathbf{\Omega}\hat{\mathbf{C}}) - \rho|\mathbf{P} \otimes \mathbf{\Omega}|_1\} \tag{4.2}$$

where $\hat{\mathbf{C}}$ is the empirical covariance of the microbial data, and $\mathbf{\Omega}$ is the precision matrix of the estimated associations among microbes. Here det and tr denote the determinant and the trace of a matrix, respectively. $|\mathbf{\Omega}|_1$ is the L_1 norm, *i.e.*, the sum of the absolute values of the elements of $\mathbf{\Omega}$ and $\otimes$ represents the component-wise multiplication. When the value of $\mathbf{P}_{i,j}$ is large, this directly puts a heavy penalty and represents a weaker association between taxa and vice versa. This way, by imposing the prior information, we can accurately infer the associations among microbes.

4.5 Automated text-mining of microbial associations

We extract two types of data to be used as priors for our model. One type of data is from the microbial co-occurrence in literature that examines the number of abstracts where two taxa appear together and compares this to random chance. The second type of data is from the machine learning-based method that extracts the full details of the interaction, including the sign and direction of the interaction.

To acquire the prior knowledge ($\mathbf{P}$) of microbial associations from reported experiments and published papers, we utilize the PubMed database (https://www.ncbi.nlm.nih.gov/pubmed/) that contains a massive amount of papers with abstracts. For the 16S rRNA data where the taxonomy level can only be achieved at the genus level, we adopt the statistical testing method (*i.e.*, Fisher's exact) [86] to identify the pairwise associations derived from the microbial co-occurrence in literature. On the other hand, for the shotgun data where the taxonomy level can be up to species level, we adopt both the microbial co-occurrence in literature and the machine-learning-based methods [87] to obtain such associations.

We modify the code available on https://github.com/CSB5/atminter that utilizes the Entrez search system to query all the possible combinations of taxon-taxon pairs from the data. More specifically, the query "taxon i AND taxon j" for genus (species) level are performed on PubMed database in order to obtain the *number* of papers that corresponds to this query term. Acquisitions of abstract's *content* follow a similar way where the query term follows the format "species i AND species j" for each pair of species. Note that, all text-mining procedures are completely automated; that is, users only need to specify the species pairs and the tool will extract the information automatically (and comprehensively) from the PubMed database.

4.5.1 Microbial co-occurrence in scientific literature

We use Fisher's exact test, which only requires the *number* of abstracts, to examine the microbial co-occurrence in scientific literature. For example, the query "taxon i AND taxon j" returns four numbers: (1) n_i: the number of abstracts that contains only taxon i, (2) n_j: the number of abstracts that contains only taxon j, (3) $n_{i,j}$: the number of abstracts

that contain both taxa i and j, and (4) M: the number of abstracts that contain neither taxa i and j. Next, by creating a 2-by-2 contingency table using the above four numbers, Fisher's exact test can be used to examine the probability that the number of abstracts where two taxa co-appear occurs at a higher rate than chance expectation. Note that we use the Bonferroni correction [93] to correct the p-value in order to deal with large amounts of candidate associations from the Fisher's exact test.

If taxa pair $\langle i,\, j\rangle$ is rejected by the alternative hypothesis with high statistical significance (*i.e.*, calculated p-value < 0.001), we put a larger penalty on entry (i, j) of the prior matrix $\mathbf{P}$. This way, we narrow down the solution space for candidate association pairs; MPLasso can effectively select the associations from these candidate pairs within this restricted space. In this respect, prior information will not dominate the results, but rather improve the algorithm's accuracy and robustness.

4.5.2 Machine learning-based approach for knowledge extraction

In [87], the authors train the support vector machine [94] based on the manually curated abstracts and classify interactions into three categories: positive, negative, and no interaction. We use the pre-trained model provided by [87] to classify the abstracts of the species pairs obtained from the PubMed database. For example, for the $\langle$*Streptococcus mitis*, *Actinomyces naeslundii*$\rangle$ query, we obtain 65 abstracts that contain both taxa names. By concatenating these abstracts into a single file, the pre-trained classifier is able to classify this pair as either interacting or non-interacting. More specifically, if species pair $\langle i,\, j\rangle$ is classified as interacting, then we put a smaller penalty on entry (i, j) of the prior matrix $\mathbf{P}$. In this respect, the species pair $\langle i,\, j\rangle$ is more likely to be selected by MPLasso. Note that these experimentally validated interactions take precedence over (and we effectively ignore) the prior information obtained from the Fisher's exact test.

4.6 Synthetic data experiments

To show the effectiveness of our proposed model, we first compare our model against several state-of-the-art models: CCREPE, SparCC, REBACCA, CCLasso, SPIEC (mb) and SPIEC (gl). All these codes have been implemented using the R language. We set up p-value at 0.05 for CCREPE and the threshold of correlation for SparCC at 0.1.

For MPLasso in real datasets, the true underlying network is only partially known and contains spurious information. To assess our algorithm performance with imperfect prior information, we consider prior information with different precision levels, where the precision level is defined as the number of true edges over the total number of edges in the prior information. The total number of edges in the prior network is set to be equal to the number of edges in the true underlying network. Therefore, a precision level of 0.1 indicates that 10% of the edges in the prior network are true edges, whereas the other 90% are spurious ones (see **Appendix** 8.10 for details of introducing priors). We report the results we obtained for 0.5 precision level in the synthetic experiments.

Method	L_1	ACC	AUPR	L_1	ACC	AUPR	L_1	ACC	AUPR
Cluster Graph									
MPLasso	0.059 (0.005)	**0.911 (0.010)**	**0.682 (0.024)**	0.052 (0.004)	**0.926 (0.009)**	**0.748 (0.029)**	0.028 (0.002)	**0.959 (0.004)**	**0.692 (0.023)**
CCLasso	0.080 (0.008)	0.893 (0.008)	0.526 (0.029)	0.068 (0.004)	0.903 (0.008)	0.614 (0.026)	0.053 (0.005)	0.950 (0.003)	0.562 (0.027)
SparCC	0.083 (0.004)	0.892 (0.009)	0.507 (0.028)	0.069 (0.003)	0.899 (0.010)	0.590 (0.030)	0.053 (0.002)	0.949 (0.003)	0.533 (0.027)
REBACCA	**0.055 (0.005)**	0.896 (0.010)	0.572 (0.027)	**0.042 (0.003)**	0.905 (0.010)	0.629 (0.031)	**0.025 (0.001)**	0.950 (0.004)	0.583 (0.027)
SPIEC (mb)	-	0.893 (0.010)	0.591 (0.030)	-	0.901 (0.012)	0.615 (0.030)	-	0.952 (0.004)	0.581 (0.026)
SPIEC (gl)	0.064 (0.006)	0.894 (0.010)	0.607 (0.024)	0.063 (0.006)	0.900 (0.011)	0.630 (0.024)	0.030 (0.003)	0.952 (0.004)	0.615 (0.026)
CCREPE	0.123 (0.011)	0.887 (0.009)	0.471 (0.022)	0.123 (0.011)	0.892 (0.009)	0.567 (0.025)	0.060 (0.005)	0.943 (0.003)	0.436 (0.022)
Band Graph									
MPLasso	0.093 (0.002)	**0.867 (0.007)**	**0.654 (0.018)**	0.087 (0.005)	**0.887 (0.007)**	**0.694 (0.019)**	0.048 (0.001)	**0.939 (0.002)**	**0.654 (0.013)**
CCLasso	0.092 (0.006)	0.853 (0.003)	0.468 (0.018)	**0.074 (0.004)**	0.863 (0.005)	0.551 (0.024)	0.062 (0.003)	0.929 (0.002)	0.506 (0.015)
SparCC	**0.087 (0.003)**	0.852 (0.003)	0.452 (0.020)	0.077 (0.003)	0.858 (0.004)	0.523 (0.019)	0.058 (0.001)	0.927 (0.001)	0.476 (0.015)
REBACCA	0.093 (0.002)	0.854 (0.004)	0.520 (0.027)	0.080 (0.002)	0.865 (0.005)	0.576 (0.024)	**0.044 (0.001)**	0.930 (0.002)	0.537 (0.016)
SPIEC (mb)	-	0.851 (0.004)	0.597 (0.039)	-	0.858 (0.007)	0.619 (0.025)	-	0.929 (0.002)	0.571 (0.020)
SPIEC (gl)	0.096 (0.000)	0.850 (0.004)	0.617 (0.027)	0.096 (0.000)	0.856 (0.007)	0.629 (0.016)	0.050 (0.000)	0.928 (0.002)	0.588 (0.013)
CCREPE	0.167 (0.004)	0.848 (0.000)	0.427 (0.017)	0.170 (0.003)	0.851 (0.004)	0.504 (0.019)	0.089 (0.001)	0.922 (0.000)	0.391 (0.012)
Scale-free Graph									
MPLasso	**0.066 (0.008)**	**0.970 (0.003)**	**0.750 (0.027)**	0.065 (0.008)	**0.976 (0.004)**	**0.817 (0.039)**	**0.033 (0.004)**	**0.985 (0.001)**	**0.758 (0.024)**
CCLasso	0.077 (0.008)	0.964 (0.002)	0.620 (0.046)	0.071 (0.010)	0.969 (0.004)	0.740 (0.063)	0.046 (0.005)	0.983 (0.001)	0.641 (0.041)
SparCC	0.078 (0.006)	0.963 (0.001)	0.594 (0.038)	0.067 (0.006)	0.967 (0.003)	0.697 (0.046)	0.050 (0.002)	0.982 (0.001)	0.610 (0.040)
REBACCA	0.069 (0.008)	0.966 (0.003)	0.668 (0.046)	**0.064 (0.010)**	0.973 (0.004)	0.758 (0.046)	0.034 (0.004)	0.984 (0.001)	0.673 (0.030)
SPIEC (mb)	-	0.962 (0.003)	0.646 (0.049)	-	0.969 (0.005)	0.710 (0.055)	-	0.982 (0.002)	0.630 (0.063)
SPIEC (gl)	0.068 (0.007)	0.963 (0.003)	0.695 (0.024)	0.069 (0.008)	0.969 (0.004)	0.747 (0.034)	**0.033 (0.004)**	0.983 (0.001)	0.712 (0.026)
CCREPE	0.072 (0.004)	0.961 (0.000)	0.549 (0.030)	0.070 (0.004)	0.962 (0.001)	0.660 (0.040)	0.035 (0.002)	0.980 (0.000)	0.515 (0.028)

Table 4.1: Performance comparison of different methods for additive log normal model. We consider three different graph structures and three sets of parameters, namely, $(p = 50, n = 50)$, $(p = 50, n = 100)$, and $(p = 100, n = 100)$. For each experiment, we average over 100 simulation runs with standard deviations in round brackets. We use three metrics (L_1, ACC, AUPR) to quantify the performance as defined in 4.6.2. Bold numbers show best results.

4.6.1 Synthetic data generation

We simulate the compositional data from from the additive log normal distribution with a given mean and covariance matrix $\ln d \sim N(\mu, \mathbf{\Sigma}), x_i = \frac{d_i}{\sum_{i=1}^{p} d_i}$, where μ and $\mathbf{\Sigma}$ represent the mean and covariance, respectively; d is the sample generated from a multivariate logarithm normal distribution, and x is a compositional vector. To evaluate the performance of our model to recover different network structures, we report three representative network structures: cluster, band(4), and scale-free graph in Table 4.1. Different sparsities on graph structure can strongly affect network recovery, and thus the network topologies we reported span a range of sparsity where band(4) is the least sparse followed by cluster and scale-free graphs.

We use the package in [95] to generate the precision matrix ($\mathbf{\Theta}$) and the positive definite covariance matrix $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$ for each graph (see **Appendix** 8.4 and Fig. 8.1). The covariance matrix is then computed to generate multivariate normal samples (d). Since the number of samples can be around the same order as the number of OTU in real datasets, we generate a small number of samples to evaluate the performances of MPLasso and other methods. More specifically, we evaluate 6 different combinations, namely, (p = 50, n = (50, 100, 200)) and (p=100, n = (100, 200, 400)). For each combination, we simulate 100 runs and calculate the mean value and standard deviation for all performance metrics.

4.6.2 Performance evaluation metrics

We consider four different metrics as follows:

- Area Under the Precision-Recall Curve (AUPR): We compute the AUPR and ignore the sign of inferred edges. Precision is defined as the number of true positives, divided by the sum of true and false positives, while Recall is defined as the number of true positives, divided by the sum of true positive and false negatives.
- Accuracy (ACC): We estimate ACC as the number of true positives plus the true negatives, divided by total number of pairwise correlations.
- L_1 distance: The L_1 distance is defined as the difference between estimated and true values. More specifically, $L_1 = |\mathbf{R} - \hat{\mathbf{R}}|$, where $\mathbf{R}$ is the true correlation matrix and $\hat{\mathbf{R}}$ is the estimated correlation matrix.

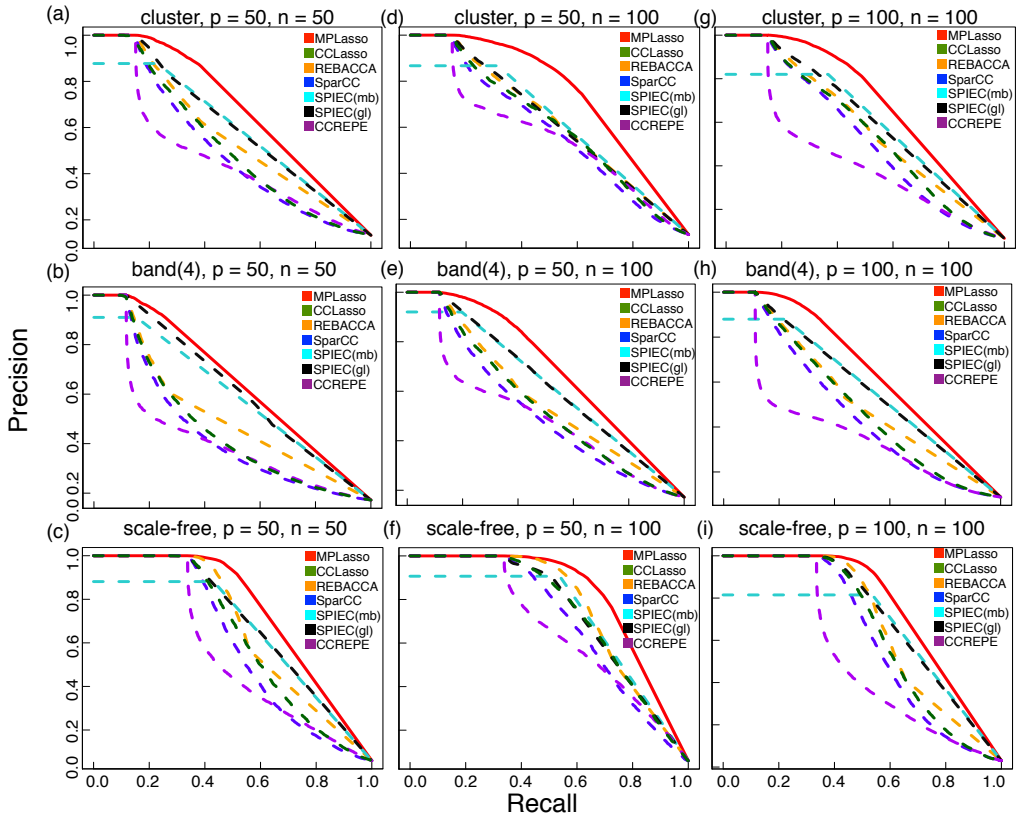

Figure 4.3: AUPR curves of different methods for additive log normal model. Each set of experiment are averaged over 100 simulations. We compare three different sets of sample size (n) and OTU numbers (p) for three different graph structures. For (p = 50, n = 50), (a) cluster, (b) band(4), and (c) scale-free. For (p = 50, n = 100), (d) cluster, (e) band(4), and (f) scale-free. For (p = 100, n = 100), (g) cluster, (h) band(4), and (i) scale-free. As can be seen, the MPLasso (red curve) performs better than all other methods.

4.6.3 Performance comparisons against existing algorithms

We report (p = 50, n = 50), (p = 50, n = 100), and (p=100, n = 100) in Table 4.1. For L_1 distance, all the methods are evaluated on the correlation matrix. For ACC and AUPR, in order to have a fair comparison among different methods, the microbial associations for correlation (covariance) based method (*i.e.*, SparCC, CCREPE, REBACCA, and CCLasso) are obtained from the inferred microbial correlation (covariance), while for precision based methods (*i.e.*, SPIEC (mb), SPIEC (gl), and MPLasso) is obtained from the inferred precision matrix. As it can be seen in both Fig. 4.3 and Table 4.1, our proposed method (MPLasso) achieves the best AUPR on all the cases; this confirms that MPLasso can accurately identify associations among microbes. However, the L_1 distance is greater than REBACCA due to the fact that MPLasso directly estimates the precision matrix of microbial associations, not on the correlation matrix.

As we increase the OTU numbers and fix the sample size, the performance for all methods degrades. For the case where (p =100, n = 100), MPLasso still outperforms all other methods in terms of ACC and AUPR. On the other hand, as we vary the sample size from 50 to 100 and fix the number of OTU to 50, the performance of all the metrics for MPLasso increases, as expected. When sample size equals 100, which is often the case in practice (*e.g.*, HMP dataset), MPLasso can achieve outstanding performance in terms of both *average* ACC and AUPR (0.93 and 0.75, respectively). Also, when sample size equals 200 and 400, MPLasso can near-perfectly recover the network (*i.e.*, AUPR $\approx$ 1).

As shown in Fig 4.3, we can see that most of the algorithms can achieve high precisions under low recalls, which means that they can accurately estimate the true edges. However, as the number of recalls increases, only MPLasso can still achieve high precision when comparing with other methods; this shows that MPLasso can recover edges with very low errors. Additionally, all methods show dependence on different graph structures; this is due to different sparsity of a particular type of graph encodes. Since scale-free graph is less sparse than band(4) and cluster graph, all methods achieve better performance in inferring edges. Additionally, even when precision level is as low as 0.1 (*i.e.*, only 10% of the edges in prior information are true edges, whereas the other 90% are spurious ones), MPLasso can still achieve up to an *average* 0.65 in AUPR for the case where (p = 50, n = 50) (see **Appendix** 8.6 and Fig. 8.2).

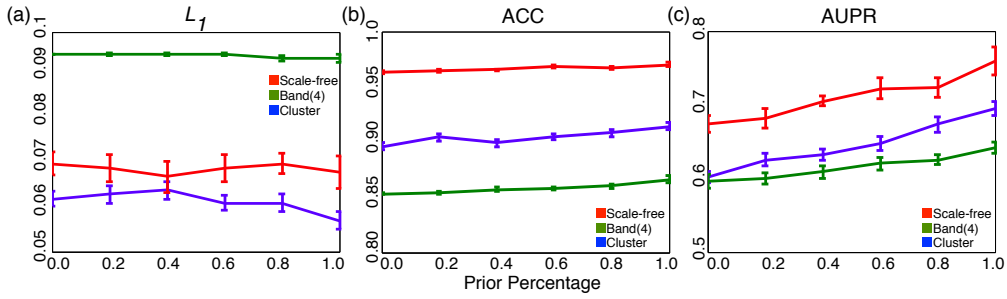

Figure 4.4: The performance of different amount of prior information on three different graph structures. (a) L_1 distance (b) ACC (c) AUPR.

In addition to examining the impact of different precision levels, we also quantify the effect of different amount of prior information being used in the synthetic experiments for three graph structures (cluster, band(4), and scale-free graphs). As shown in Fig 4.4, if the amount of prior information increases, then the performance in terms of AUPR increases too, as expected. For L_1, which is evaluated based on correlations, different amounts of prior information have little effect due to the fact that MPLasso directly estimates the precision matrix. For ACC, since MPLasso has already achieved high performance, increasing the amount of prior information only brings a small increment of improvement in performance. When prior percentage = 100%, AUPR achieves around 20% improvements over the case without using any prior information. For cases without using any prior information, MPLasso can still achieve comparable results with other existing methods presented in Fig 4.3 and Table 4.1.

For the zero-inflated distribution (discussed in **Appendix** 8.7 and Fig. 8.3), as it can be seen in Figs. 8.4-8.5, Table 8.4-8.5, the performance of our proposed method outperforms all the other methods except a few cases involving hub graphs; this is similar to the results for the additive log normal model. In summary, our results show that MPLasso works well with many different distributions and graph structures even in the cases with low precision levels and less prior information.

4.7 Human microbiome project data experiments

Emboldened by the success of our proposed algorithm on synthetic data, we have applied MPLasso to infer the associations among microbes for HMP data. Acquisitions and preprocessing for both 16S rRNA and shotgun sequencing data are described in section 4.3. We report the same three body sites (*i.e.*, buccal mucosa, supragingival plague, and tongue dorsum) of each pipeline and filter out OTUs that appear in less than 10% of total samples — two more body sites (i.e., stool and anterior names) are reported in **Appendix** 8.8 and Fig. 8.6. The total number of samples and OTUs are summarized in Table 4.2 and Table 8.6.

We use the clr transformation in (4.1) and add pseudo count 0.1 to all the samples, then normalize the counts to get compositional data. However, there is no true correlation network of taxon-taxon associations in real data as opposed to synthetic data. To assess and compare the performance among different methods in real data experiments, we measure the reproducibility of the resulting networks. More specifically, we define the “gold standard" network as the one that uses the full dataset. The reproducibility is defined as the number of edges that had been correctly estimated when using only *half* of the samples in the full dataset compared to the "gold standard" network. We randomly select half of the samples in the full dataset of each body site and then average over 20 independent simulations. We compare the reproducibility of the MPLasso against SPIEC (gl) which has a better performance than other existing algorithms on synthetic datasets as well as CCLasso which has a better performance than other correlation based methods in [85].

The reproducibility results are summarized in Table 4.2. MPLasso has a better reproducibility over SPIEC (gl) and CCLasso; this implies that MPLasso is not only more

Body Site	(n, p)	MPLasso	SPIEC (gl)	CCLasso
HMASM				
BucMuc	(113, 73)	**0.963 (0.003)**	0.904 (0.013)	0.915 (0.005)
SupPla	(124, 129)	**0.942 (0.005)**	0.877 (0.009)	0.919 (0.005)
TonDor	(130, 103)	**0.948 (0.004)**	0.754 (0.030)	0.913 (0.015)
HMMCP				
BucMuc	(406, 74)	**0.923 (0.005)**	0.756 (0.033)	0.820 (0.014)
SupPla	(423, 84)	**0.923 (0.004)**	0.862 (0.007)	0.837 (0.012)
TonDor	(410, 77)	**0.934 (0.003)**	0.820 (0.014)	0.850 (0.012)
HMQCP				
BucMuc	(312, 75)	**0.876 (0.006)**	0.777 (0.023)	0.818 (0.011)
SupPla	(313, 51)	0.883 (0.010)	0.796 (0.015)	**0.896 (0.007)**
TonDor	(316, 45)	**0.860 (0.009)**	0.735 (0.020)	0.841 (0.024)

Table 4.2: Reproducibility for MPLasso, SPIEC (gl), and CCLasso at different body sites of different types of HMP datasets. For each experiment, we average over 20 simulation runs with standard deviations in round brackets. Bold number shows best result. n and p represent sample size and taxa number, respectively.

robust, but also more accurate at inferring edges. We also consider reproducibility on different percentages of highly connected nodes in Table 8.7. Only when we consider as little as only 25% of high degree nodes, CCLasso has a better performance (but even so for 2% only, on average).

To compare the estimated association networks at each body site for different pipelines (i.e, HMASM, HMMCP and HMQCP), we select the "top players" (*i.e.*, high degree nodes) and arrange them using a counterclockwise layout as shown in Fig 4.5. For the genus level data, since we only utilize the Fisher's exact test (that only requires the information of the *number* of abstracts), we can use *contents* of published scientific literature to validate the inferred associations. In contrast, for the species level data, the machine learning-based approach has already used the contents of abstract to obtain the prior information; therefore, it is inappropriate to use any papers that appear in the PubMed search results to validate the inferred associations. To circumvent the potential circular validation, we only use the scientific literature that has *not* yet been used to create the prior information.

For the buccal mucosa (BucMuc), the association pair $\langle$*Streptococcus mitis*, *Actinomyces naeslundii*$\rangle$, which was found in HMASM (Fig 4.5(a)), has been shown to have associations [96]. Additionally, the associations are also detected at genus level data as shown in Fig 4.5(b). Note that the top degree nodes in HMMCP and HMQCP has 70% in common (*i.e.*, belongs to same genus) which implies that the microbial composition of BucMuc is relatively robust.

For the supragingival plague (SupPla), the "top players" in species level data (Fig 4.5(d)) mainly come from two genera: *Actinomyces* and *Prevotella* which can be widely found in SupPla and also correspond well with the HMMCP dataset (Fig 4.5(e)). Similarly, the species level associations in tongue dorsum (TonDor) is dominated by *Actinomyces* as

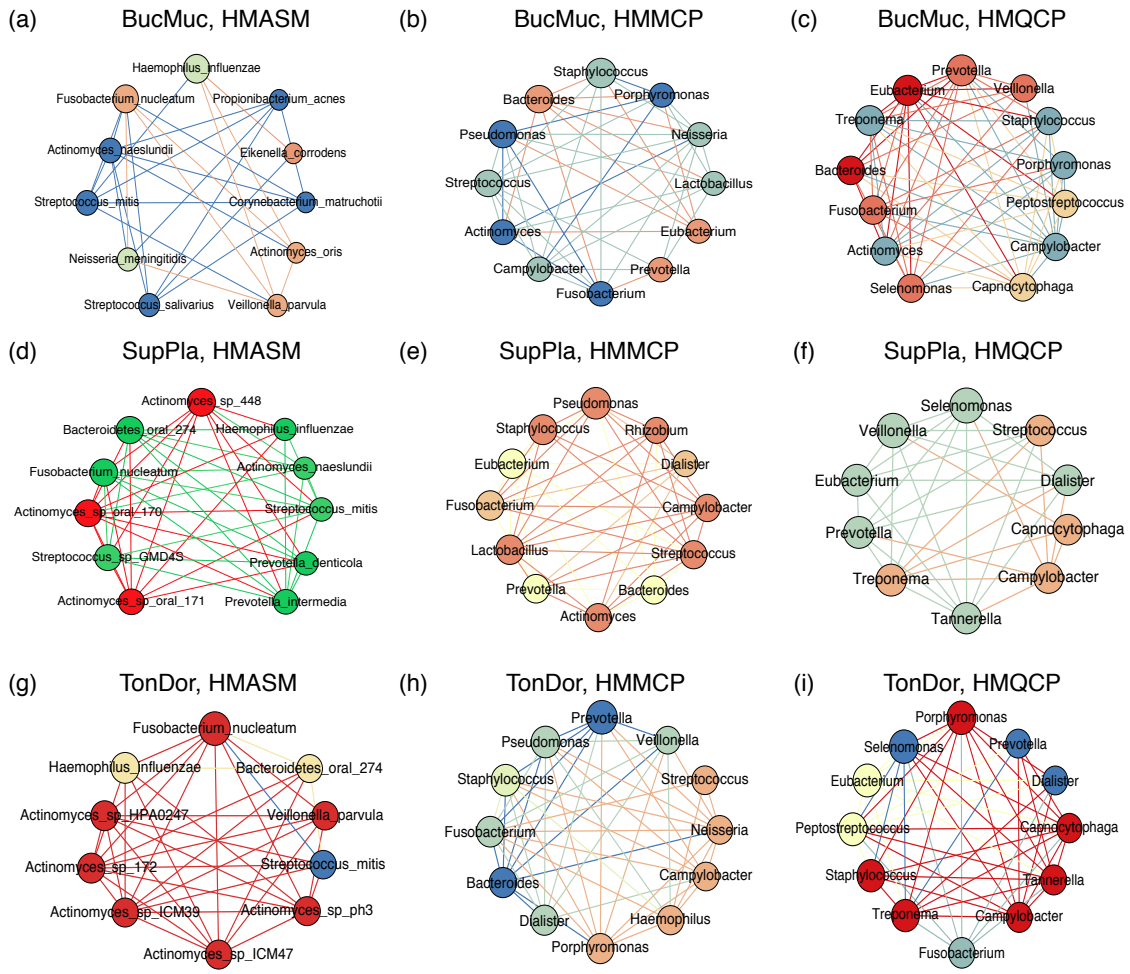

Figure 4.5: Association network visualization of top degree nodes at different human body sites for different data types. The same node colors represent the communities nodes belong to. For BucMuc: (a) HMASM, (b) HMMCP, and (c) HMQCP. For SupPla: (d) HMASM, (e) HMMCP, and (f) HMQCP. For TonDor: (g) HMASM, (h) HMMCP, and (i) HMQCP. As can be seen from species level data (HMASM), phylogenetically related OTUs fall in the same community. Node size represents the relative node degree within the association network with counterclockwise layout. The color of the edges is the same as the node color and does not have any special meaning. Abbreviations: BucMuc: Buccal mucosa, SupPla: Supragingival plague, TonDor: Tongue dorsum.

shown in Fig 4.5(g); this is because *Actinomyces* possess 10 different strains out of the total 103 taxa, yet this does not imply that *all* members of a particular genus group should be associated. Although not seen in Fig 4.5(h) and (i), genus *Actinomyces* is also a high degree node in the association network of the genus data.

One noticeable observation in the species level dataset (HMASM) is that the same genus belongs to the same community which means that edges are mostly found within OTUs of the same taxonomic group. This phenomenon is called assortativity and it has been widely observed in other microbial network studies [97]. However, this does not

imply that all members of the same taxon should be ecologically associated. To quantify the similarity of high degree nodes that are found both in HMMCP and HMQCP datasets, we compute the correlation between node degrees at different body sites by utilizing the Spearman correlation method (see **Appendix** 8.9). We found that TonDor has lower correlations ($\sim$0.5) than other body sites ($\sim$0.7); this can be directly observed from Fig 4.5(h)-(i) that have a few high degree genera in common.

4.8 Conclusion

Inferring associations (putative interactions) among microbes and understanding their influence on the human body is an important step towards precision medicine. Advancements of high-throughput sequencing techniques enable us to gather metagenomic sequence data from different environment and human niches. The available high-throughput experimental data, however, are compositional and high-dimensional in nature.

Existing microbial network inferring methods focus on inferring the compositional data and use the graph sparsity assumption to overcome problems caused by high-dimensional data. However, all of these approaches do not consider the information that can be obtained from the scientific literature to directly describe the associations among microbes or their co-occurrence. By integrating multiple levels of biological information into the statistical models, we have shown that one can dramatically increase the model accuracy and edges recovery rate. To the best of knowledge, this is the first work to propose this automated pipeline to infer the associations on microbial data, show its feasibility, and measure performance metrics on both synthetic and real datasets.

We have also shown that our proposed algorithm *Microbial Prior Lasso (MPLasso)* outperforms all other existing methods when using synthetic data with different graph structures which simulate different levels of sparsity. We have evaluated several combinations of sample sizes and number of taxa to demonstrate the applicability of our approach under different conditions and suggest rough guidelines for requisite sample size for the real data for the given assumption of the underlying graph structures.

Additionally, the use of prior information does not dominate the inferred results. Indeed, the prior information obtained by the microbial co-occurrence in literature is only used to restrict the search space in order to infer associations that are more plausible (*i.e.*, more likely to be associated) than other candidate pairs of associations. More specifically, we first calculate the probability of association among taxa. Next, if two taxa are not associated, we penalize the associations among these two taxa when solving MPLasso. Consequently, MPLasso can effectively select taxa that are highly associated with high statistical confidence. In this respect, prior information will not dominate the results, but rather improve the algorithm's accuracy and robustness.

Our analyses on different levels of real HMP data show that MPLasso achieves better reproducibility than SPIEC (gl) and CCLasso; we have also found the assortativity at the species level data (HMASM) at different body sites. In other words, OTUs are more likely to interact with other phylogenetically related OTUs. Additionally, the detected genera at genus level (HMMCP and HMQCP) datasets show high correlations based on their node degrees (*i.e.*, number of edges a node has to other nodes). Those high degree

nodes (*i.e.*, “top players") have been found experimentally as being ubiquitous at each body site; this confirms that MPLasso can accurately detect the “top players" and even correctly infer the associations among them. The resulting microbial association network can suggest credible directions for experimentalists to validate the results without exhausting search for all possible associations.

Recent studies report that people affected by microbiome related diseases show different microbiome profiles when compared to healthy individuals. For example, results show that individuals affected by the inflammatory bowel disease (IBD) have (30-50)% percent less biodiversity of commensal bacteria (*e.g.*, *Firmicutes* and *Bacteroidetes*) when compared to healthy individuals. Another example shows that individuals with Type 2 diabetes (T2D) exhibit significant changes in 190 microbial OTUs, with particularly high abundance of *Enterobacteriaceae* compared to healthy individuals [98]. Therefore, by creating a more accurate microbial association network, scientists working in this field will be able to accurately identify the relationship between microbiome related diseases (such as T2D) and groups of taxa based on the inferred network. This way, scientists can develop new drugs or use probiotics to directly target identified groups of taxa.

Finally, the estimated microbial association networks of the real datasets can be used to understand why and how various eco-systems evolve over time. Recent studies use association networks to fit dynamic models, *e.g.*, differential equation-based model of gut microbiome evolution of mice [99]. These microbe associations represent the putative microbial interactions that provide partial information about the true interaction network. Therefore, by incorporating the association network as additional information, we may be able to infer the microbial interaction networks more accurately [100]. Overall, MPLasso shows promising results and outperforms state-of-the-art methods. In the present framework, our proposed MPLasso creates the inferred association network to provide additional partial information; this can be useful to reveal the underlying dynamics (*i.e.*, interactions) of microbial communities. However, MPLasso was not tested on a dynamic model of microbial communities. Inferring the dynamics or interactions among microbial communities would require a new algorithm which is presented in Chapter 5.

Chapter 5

Inferring Microbial Interactions from Metagenomic Time-series

In previous chapters, we have mainly focused on how to engineer bacteria to perform certain pre-defined tasks such as virulence control and drug delivery. However, our human body usually contains trillions of bacteria belonging to several hundred different species. In order to understand how microbiome interacts with our human body, in this chapter, we present a novel framework called Microbial Time-series Prior Lasso (MTPLasso) to infer the interactions among different microbes directly from sequencing data; these interactions can be used to understand how microbial communities adapt, develop, and interact over time with the human body and the surrounding environment. We first formulate our proposed framework which integrates sparse linear regression with microbial co-occurrences and associations obtained from scientific literature and cross-sectional metagenomic data. Next, we show that MTPLasso outperforms existing models in terms of precision and recall rates, as well as the accuracy in inferring the interaction types. Finally, the interaction networks we infer from human gut data demonstrate credible results when compared against real data.

5.1 Introduction and motivation

Microbiome plays an important role in determining the human health and well-being [101, 102]. For example, the inflammatory bowel disease (IBD) currently affects approximately 1.8 million Americans. Results show that IBD-affected individuals have 30-50% reduced biodiversity of commensal bacteria, such as significant decrease in *Firmicutes* and *Bacteroidetes* [103]. Consequently, finding new ways to analyze the human microbiome may help better understand, prevent, or cure diseases.

Microbial communities exhibit rich dynamics including the way they adapt, develop, and interact with the human body and the surrounding environment. However, the specific way of how human associated microbes affect the human health remains largely unknown. To this end, studies on the microbial interactions can provide a solid foundation to model the interplay between the (host) human body and the microbial populations; this

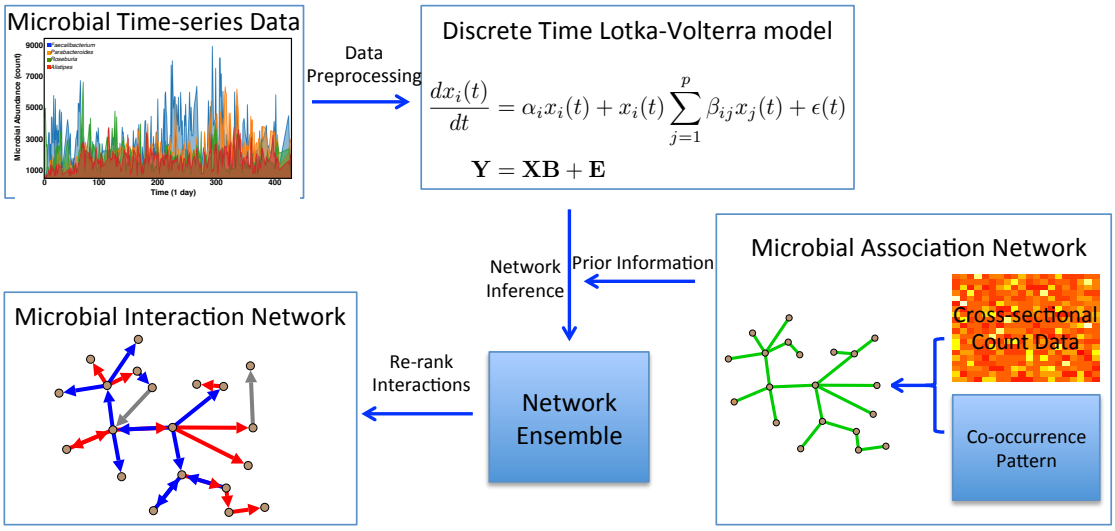

Figure 5.1: The proposed MTPLasso pipeline. We first preprocess the microbial time-series data and then model it using discrete time Lotka-Volterra model. By incorporating the associations from cross-sectional data and microbial co-occurrences, our proposed algorithms can robustly infer interaction among microbes. Network bagging and interaction re-ranking can further improve robustness and provide credible results. The associations among microbes are represented by green color. The competitive and beneficial interactions are represented by red and blue colors, respectively, while the grey color represents wrongly estimated edges.

can serve as a key step towards precision medicine [75]. Unfortunately, understanding microbial interactions is difficult, as most microbes cannot be easily cultivated in standard laboratory settings.

To date, various datasets that focus on human niches exist; they can be classified into two different types: cross-sectional and time-series datasets. Cross-sectional data (*e.g.*, Human Microbiome Project [77]) provides microbial co-occurrences collected from different body sites (*e.g.*, gut, mouth) each with multiple individual samples. Several algorithms [81, 82] have been proposed to utilize such data to infer associations (*i.e.*, putative interactions) among microbes. In contrast, time-series data [2, 104, 99] focus on the dynamics of microbial communities over time, typically, for a few individuals. By utilizing dynamical modeling (*e.g.*, ordinary differential equations), specialized algorithms [99, 105, 106] can infer the interactions between microbe species; this is completely different from the associations inferred from the cross-sectional data.

In this chapter, we aim at analyzing the *structure* and *dynamics* of networks of interaction among microbes of human microbiome in order to understand how microbes can affect the human health. To this end, there exist several challenges: First, the amount of (time-series) sequencing data that corresponds to human microbiome available from public websites is scarce. For instance, to date, one of the largest published datasets [2] only provides a few hundred time points for one individual, while the number of microbes (p) usually ranges from hundreds to thousands. As a consequence, the number of possible interactions (p^2) is much greater than the number of samples that can be obtained

from real datasets (*i.e.*, high-dimensional data). Another big challenge stems from the nature of the data itself. Sequencing data only provides the *relative* abundance of various species; this is because the sequencing results are a function of sequencing depth and the biological sample size [79]. Therefore, from a statistical standpoint, the relative microbial abundance falls into the class of compositional data [80]; this causes statical methods such as Pearson or Spearman correlations (which work with absolute values) to generate spurious results.

To solve the problem caused by high-dimensional data, we propose to integrate multiple levels of biological information to enhance the model accuracy on inferring microbial interactions. To date, an increasing amount of scientific literatures provides data which can be mined not only for the co-occurrence of microbes, but also to predict microbial associations directly. For instance, the pioneering work in [86] considers automated analysis of the co-occurrence of bacterial species through statistical testing approaches (*e.g.*, Fisher's exact test). In [107], the authors show that gut and mouth microbiomes display pronounced universal dynamics (*i.e.*, host independent). As such, by incorporating the microbial co-occurrences with various cross-sectional data into the existing methods [82], the resulting microbial associations can serve as prior information to infer the interactions for microbial time-series; this can truly mitigate the problem of high-dimensionality.

Although several approaches have successfully applied prior biological information to reconstruct gene-interaction networks [88, 89, 108], to the best of knowledge, we are the first to consider multiple levels of biological knowledge as *a priori* information to derive microbial interaction networks. Indeed, as shown in Fig. 6.1, we first transform the microbial time-series count data into relative abundance data (*i.e.*, compositional data) and then model it by using a discrete time Lotka-Volterra model. Thus, we can easily incorporate the prior information provided by microbial associations to a Lasso regression algorithm and solve for a sparse interaction network; this sparse algorithm can help increase the model performance in terms of precision and recall rates.

To boost the performance and robustness, we further conduct model ensemble and selection through bootstrap aggregating [109] and re-ranking [108], respectively for the inferred interactions; the resulting interactions are robust to different choice of parameters. Overall, the proposed algorithm, the *Microbial Time-series Prior Lasso (MTPLasso)*, turns out to be more accurate than the other existing methods on inferring the microbial interactions [104, 105].

To assess the performance of MTPLasso, we first evaluate and compare it against other previously proposed methods, namely, ridge regression [104] and sparse regression [105], under different graph structures that encode various levels of sparsity. We show that our proposed MTPLasso outperforms all these methods in terms of two metrices, namely, area under the precision-recall curve (AUPR) and interaction type classification accuracy (IACC) of network interactions prediction. Next, we evaluate the dataset available in [2] and compare the inferred results against laboratory settings.

Symbols	Representations
p, T	Number of microbes and time points
$\mathbf{Y}$, $\mathbf{X}$, $\mathbf{B}$, $\mathbf{E}$	Response, data, parameter, and error matrices.
S	Mean microbial total abundance
$\boldsymbol{\alpha}$, $\hat{\boldsymbol{\alpha}}$	True and estimated intrinsic growth rates matrices
$\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}$	True and estimated interaction matrices
λ, ρ	Penalizing parameters for interaction and association
$\mathbf{P}$, $\mathbf{Q}$	Prior matrix for interaction and association
γ, $\mathbf{R}$	Error of true and predicted model and interaction rank matrix
μ, $\boldsymbol{\Sigma}$, $\boldsymbol{\Omega}$	Mean, covariance, and precision matrix of normal distribution
$\hat{\mathbf{C}}$	Empirical covariance of cross-sectional abundance data
θ, b	Weight parameter and bagging size

Table 5.1: Model and experimental parameters.

5.2 Mathematical modeling for microbial time-series

In this section, we first describe the mathematical modeling for time-series data based on Lotka-Volterra model. Next, we describe the modeling of our proposed Microbial Time-series Prior Lasso (MTPLasso). We outline the synthetic data generation and real data preprocessing in **Appendix** 8.11 and section 1.5, respectively. Model and experimental parameters are summarized in Table 5.1.

5.2.1 Discrete time Lotka-Volterra (LV) model

Metagenomics data provides the relative abundance for microbes in discrete time intervals (*e.g.*, daily measurements); this makes it possible to infer the interactions among microbes directly from such data. To model the dynamics and interactions among microbes over time, a sensible way is to utilize the discrete time LV model for population dynamics [110].

For *absolute* abundance data (x) that contains p microbes over T time points, the behavior of microbe i at time $t + dt$ is related to other microbes at time t. The LV model explicitly captures the interactions among microbes via the interaction coefficient (β). More precisely, there exist three possible interactions between microbes i and j: beneficial ($\beta_{ij} > 0$), competitive ($\beta_{ij} < 0$), and non-interacting ($\beta_{ij} = 0$). Beneficial interaction means that microbe i can facilitate the growth of microbe j, while competitive interaction inhibits the growth of microbe j.

The change of microbial abundance can also be affected by environment and other noise sources such as measurement errors. To account for such effects, we can directly add a stochastic noise ϵ into the LV model. Specifically, the LV model is as follows:

$$\frac{dx_i(t)}{dt} = \alpha_i x_i(t) + x_i(t) \sum_{j=1}^{p} \beta_{ij} x_j(t) + \epsilon_i(t) \tag{5.1}$$

where α_i represents the intrinsic growth rate for microbe i.

As indicated by [105] and shown in Fig. 8.7, the microbiome abundance profiles stay close to the equilibrium values (with some fluctuations). Therefore, we can assume that the LV system of equations have a unique steady-state solution; by setting $dt = 1$ (*i.e.*, unit time measurement), and without loss of generality, we can rewrite Eq. (5.1) near steady-state as:

$$\frac{dx_i(t)}{x_i(t)} = [\alpha_i + \sum_{j=1}^{p} \beta_{ij}(x_j(t) - \overline{x_j}) + \epsilon_i(t)]dt \tag{5.2}$$

where $\overline{x_j}$ is the equilibrium abundance of microbe j. To fit the dynamics described by Eq. (5.2), we can approximate the left hand side by using the relationship: $\frac{dx_i(t)}{x_i(t)} = d\ln x_i(t) \approx \ln x_i(t+1) - \ln x_i(t)$. Additionally, we can combine the intrinsic growth rates ($\boldsymbol{\alpha} \in \mathbb{R}^p$) and interactions ($\boldsymbol{\beta} \in \mathbb{R}^{p\times p}$) matrices to form the parameter matrix $\mathbf{B} = [\boldsymbol{\alpha}; \boldsymbol{\beta}] \in \mathbb{R}^{(p+1)\times p}$. By using the above notations, we can reformulate Eq. (5.2) into a compact representation:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E} \tag{5.3}$$

where the entry Y_{ti} of the response matrix $\mathbf{Y} \in \mathbb{R}^{T\times p}$ is defined as $Y_{ti} \triangleq \ln x_i(t+1) - \ln x_i(t)$. The entry E_{ti} of the error matrix $\mathbf{E}$ is equal to $\epsilon_i(t)$. The data matrix $\mathbf{X} \in \mathbb{R}^{T\times(p+1)}$ is defined as:

$$\mathbf{X} = \begin{pmatrix} 1 & x_1(1) & x_2(1) & \cdots & x_p(1) \\ 1 & x_1(2) & x_2(2) & \cdots & x_p(2) \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_1(T) & x_2(T) & \cdots & x_p(T) \end{pmatrix}$$

Eq. (5.3) can be considered as a standard linear regression problem if the entry in error matrix ($\mathbf{E}$) is assumed to follow normal distribution. In the following analyses, without loss of generality, error terms are considered to follow a normal distribution for both synthetic and real datasets.

To account for relative abundance, we follow the same assumption as in [105], *i.e.*, given time t, the fluctuations of the total abundance ($S(t) = \sum_i^p x_i(t)$) around the mean value $\overline{S} = \sum_i^p \overline{x_i}$ are small. As a result, the dynamics of the relative abundance is described by:

$$\mathbf{Y} \approx \tilde{\mathbf{X}}\tilde{\mathbf{B}} + \mathbf{E} \tag{5.4}$$

where $\widetilde{X_{ti}} = X_{ti}/\overline{S}$ and $\widetilde{B_{ti}} = \overline{S}B_{ti}$. We can solve Eq. (5.4) to infer unscaled interactions and then relate them to the true interactions through the mean total abundance $\overline{S}$.

5.2.2 Microbial time-series prior lasso (MTPLasso)

The microbial time-series data can be formulated into a linear regression problem as explained in Section 5.2.1. To infer the pairwise interactions among microbes, we can use

sparse regression methods and select the interactions that minimize the mean square error (*i.e.*, $||\mathbf{Y} - \mathbf{XB}||_2^2$). More specifically, by applying the L_1 norm on the interaction matrix, the standard Lasso regression can be formulated as follows:

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta} \in \mathbb{R}^{p \times p}} \left\{ \|\mathbf{Y} - \mathbf{XB}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\} \tag{5.5}$$

where λ is a penalizing parameter and $||\boldsymbol{\beta}||_1$ is the L_1 norm, *i.e.*, the sum of the absolute values of the elements of $\boldsymbol{\beta}$. Note that the optimization problem defined in Eq. (5.5) does not penalize the intrinsic growth rates $\boldsymbol{\alpha}$ since they are viewed as intercept terms in Lasso regression.

However, microbial data usually come with a limited amount of time points (T) but with high dimensionality (p); this makes the regression problem intractable since the number of interactions (p^2) is far greater than time points (T). To make the problem tractable, we propose to utilize the information obtained from the scientific literature and cross-sectional datasets in order to construct the prior matrix $\mathbf{P} \in \mathbb{R}^{p \times p}$, where each entry $P_{i,j} \in [0, 1]$ represents the prior probability of interactions between microbe i and microbe j. We can impose different penalties on the precision matrix; this is different from the standard formulation where the penalty (λ) imposed on the precision matrix is the same. Therefore, by incorporating the prior information into the prior matrix ($\mathbf{P}$), the proposed MTPLasso can be formulated as follows:

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta} \in \mathbb{R}^{p \times p}} \left\{ \|\mathbf{Y} - \mathbf{XB}\|_2^2 + \lambda \|P \otimes \boldsymbol{\beta}\|_1 \right\} \tag{5.6}$$

where $\otimes$ represents component-wise multiplication. When the value of $P_{i,j}$ is large, it directly puts a heavy penalty and represents a weaker interaction between microbes and vice versa. This way, by imposing the prior information, we can accurately infer the interactions among microbes. To select λ, we use n-fold cross validation. We follow standard usage by setting $n = 5$.

5.2.3 Bootstrap aggregating (Bagging) MTPLasso

Microbial time-series datasets often come with a limited amount of time points. Even with the help of prior information, MTPLasso still operates in a high-dimensional setting; to enhance and stabilize the model performance, we can infer the interaction matrix multiple times and average over the inferred parameters; this is know as bootstrap aggregating or bagging [109]. This ensemble meta-algorithm can not only improve the stability and accuracy of the regression results but also reduce variance. To bag the regression results, we uniformly draw the same amount of samples with replacement; this procedure is repeated b times which results in different estimates for the interaction matrix. The proposed Bagging MTPLasso is summarized in Algorithm 1.

5.2.4 Re-rank interactions

The output of the Bagging MTPLasso is the parameter matrix $\hat{\mathbf{B}}$ which contains the estimated intrinsic growth rate ($\hat{\boldsymbol{\alpha}}$) and interaction ($\hat{\boldsymbol{\beta}}$) matrices. We can calculate the confidence score of the inferred interactions and then choose interactions with high confidence

Algorithm 1 Bagging MTPLasso

procedure BAGGING MTPLASSO($\mathbf{Y}, \mathbf{X}, \mathbf{P}, b$)
  $\mathbf{Y} \in \mathbb{R}^{T \times p}, \mathbf{X} \in \mathbb{R}^{T \times (p+1)}, \mathbf{P} \in \mathbb{R}^{p \times p}$
  for $i = 1 : b$ **do** ▷ b: number of bagging
    Draw $\mathbf{Y}'$ from $\mathbf{Y}$ uniformly with replacement
    Draw $\mathbf{X}'$ from $\mathbf{X}$ uniformly with replacement
    Compute $\mathbf{B_i}$ using Eq. (5.6) including model selection
  end for
  Compute median $\hat{\mathbf{B}}$ from $(\mathbf{B_1}, \mathbf{B_2}, ..., \mathbf{B_b})$
  return $\hat{\mathbf{B}}$ ▷ The estimated parameter matrix
end procedure

scores. However, by simply using the magnitude (i.e., $|\beta_{i,j}|$) of the inferred parameters as ranking criteria is not the best scheme since it does not account for model performance as pointed out in [108]. We hence modify the scheme proposed in [108] to account for model performance as follows:

$$r_{ij} = 1 - \frac{\gamma_j}{\gamma_{ij}} \tag{5.7}$$

where r_{ij} represents the rank for interaction β_{ij}. γ_j is defined as: $\frac{||Y_j - XB_i||^2}{Var(Y_j)}$ where Y_j is the j_{th} column vector of $\mathbf{Y}$. γ_{ij} is the model response *without* interaction β_{ij}. The resulting interaction rank matrix $\mathbf{R} = (r_{ij}) \in \mathbb{R}^{p \times p}$ can be used to select interactions that truly explain the microbial time-series.

5.2.5 Microbial prior knowledge acquisition

To acquire the prior knowledge of microbial interactions from reported experiments and published papers, we utilize the PubMed database[1] that contains a massive amount of papers with abstracts. For the 16S rRNA data where the taxonomy level can be achieved at the genus-level, we adopt the co-occurrence method [86] that uses statistical testing to identify the pairwise associations among genera.

We modify the code[2] that utilizes the Entrez search system to query all the possible combinations of genus-genus pairs from the data. More specifically, queries for the query term "genus i AND genus j" for genus-level are performed on PubMed database in order to compute the number of papers that contain each genus name, and contains both genera name.

The co-occurrence approach is based on the statistical significance on the number of abstracts obtained from the PubMed database. More specifically, we can create a 2-by-2 contingency table where entries contain: abstracts containing solely genus i or j, abstracts containing both genera i and j, and abstracts containing neither genera i nor

[1] https://www.ncbi.nlm.nih.gov/pubmed/
[2] https://github.com/CSB5/atminter

j. We use Fisher's exact test and then correct the resulting statistical significance (*i.e.*, p-value) in order to select potentially interacting genus pairs [93].

5.2.6 Cross-sectional data

Next, we incorporate the prior knowledge obtained from section 5.2.5 into the cross-sectional datasets (*i.e.*, HMP dataset) in order to correctly estimate associations among genera. We briefly summarize the approaches we use in this chapter.

Suppose the cross-sectional count data are drawn from a multivariate normal distribution with mean μ and covariance $\mathbf{\Sigma}$. The inverse covariance matrix (precision matrix) $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ encodes the conditional independence among nodes. More specifically, if the entry (i, j) of the precision matrix $\Omega_{i,j} = 0$, then node i and node j are conditionally independent (given the other nodes) and there is no edge among them.

One suitable algorithm to select the precision matrix under sparsity assumption is to utilize the graphical Lasso proposed previously in [84, 82]. By incorporating the prior information (*i.e.*, microbial co-occurrence) into the prior matrix ($\mathbf{Q}$), we can formulate the graph estimation problem as follows:

$$\hat{\mathbf{\Omega}} = \arg\max_{\mathbf{\Omega}} \{\log \det(\mathbf{\Omega}) - \text{tr}(\mathbf{\Omega}\hat{\mathbf{C}}) - \rho|\mathbf{Q} \otimes \mathbf{\Omega}|_1\} \tag{5.8}$$

where $\hat{\mathbf{C}}$ is the empirical covariance of the microbial data, and $\mathbf{\Omega}$ is the precision matrix of the estimated associations among genera. ρ is the penalizing parameter. Here det and tr denote the determinant and the trace of a matrix, respectively. When the value of $Q_{i,j}$ is large, this directly puts a heavy penalty and represents a weaker association between genera.

For our proposed MTPLasso, the entry of the prior matrix P_{ij} is set to θ if genus i has an association with genus j (*i.e.*, $\Omega_{ij} = 1$), where $\theta \in [0, 1]$ is the weight parameter. We will examine the effect of the weight parameter of the prior information in section 5.3.2.

5.3 Experiments with synthetic data

To show the effectiveness of our proposed model, we first compare our model against two regression-based and two baseline methods: ridge regression [104], LIMITS [105], Lasso regression (without prior information), and least square solution on absolute abundances (LSA). We use the package in [95] to generate different graph structures as described in **Appendix** 8.4 and shown in Fig. 8.1. Given a particular graph (*i.e.*, the vertex and edge sets), we follow the procedure as described in **Appendix** 8.4. For each set of time-series, we consider M initial conditions each with T time points. Additionally, we consider different levels of additive noise ϵ when generating time-series data.

For MTPLasso working on real datasets, the true underlying network is only partially known and contains spurious information. To assess our algorithm performance with imperfect prior information, we consider prior information with different precision levels. The total number of edges in the prior network is set equal to the number of edges in the true underlying network. Therefore, a precision level of 0.1 indicates that 10% of the

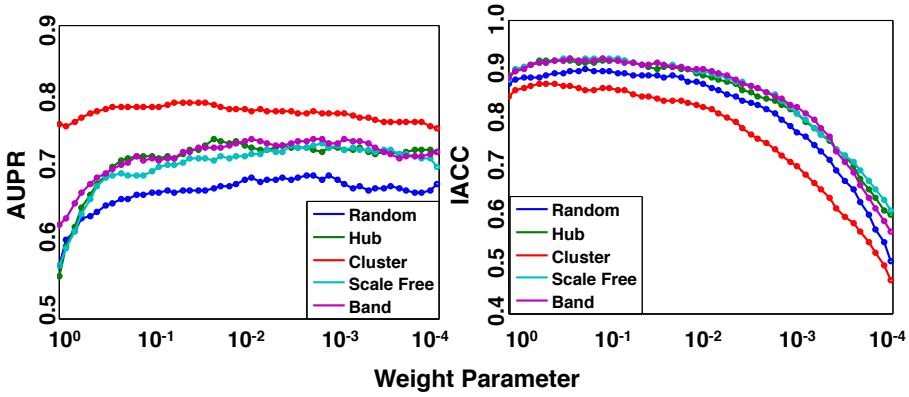

Figure 5.2: The effect of different weight parameters (θ) on AUPR and IACC for five different graphs. For AUPR, MTPLasso is relative robust to different weight parameters. In contrast, a small weight parameter (θ) can dramatically degrade IACC.

edges in prior are true edges, whereas the other 90% are spurious ones. We examine different precision levels in section 5.3.3 and report the 0.5 precision level in the following synthetic experiments.

In the following synthetic experimental analyses, the default parameter setting (without explicit indication) is: $M = 2$, $T = 30$, $\epsilon \sim N(0, 0.1)$, prior percentage = 50%, and precision level = 100%. For each experiment, we average over 20 simulation runs and show standard deviations as error bars in Figs. 5.3-5.4[3].

5.3.1 Performance evaluation metrics

We consider two different metrics as follows:

- Area Under the Precision-Recall Curve (AUPR): We compute the AUPR and ignore the sign of inferred edges. Precision is defined as the number of true positives, divided by the sum of true and false positives, while recall is defined as the number of true positives, divided by the sum of true positive and false negatives.
- Interaction Type Classification Accuracy (IACC): Since interactions can be classified into three types: beneficial ($\beta > 0$), competitive ($\beta < 0$), and non-interacting ($\beta = 0$), we estimate IACC as the fraction of correctly estimated interacting interactions (*i.e.*, $\beta \neq 0$).

5.3.2 Effects of weight parameter

To assess the sensitivity of how weight parameter (θ) can affect the performance, we vary the weight parameter within the range $[10^{-4}, 10^{0}]$.[4] We consider the case where we use 50% true interactions (*i.e.*, prior percentage = 50%) and the same amount of false

[3]For plotting purpose, the standard deviation on the figure is one-fifth of the true value.
[4]This is a typical range for the penalizing parameter using in standard Lasso regression.

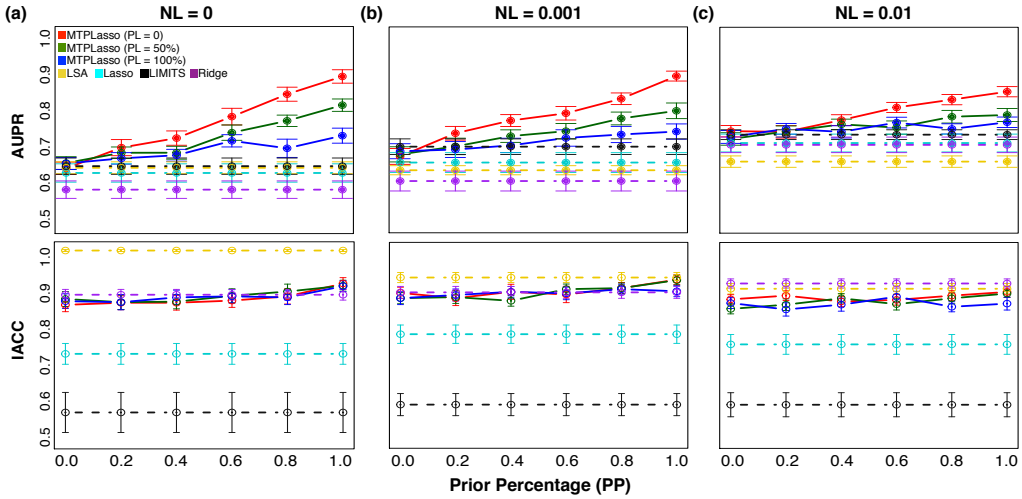

Figure 5.3: Performance evaluation on random graphs under different combinations of parameters. As prior percentage (PP) increases, both AUPR and IACC increase, as expected. MTPLasso outperforms all other methods, even with high precision level (PL) and high noise level (NL). (a) NL = 0, (b) NL = 0.001, (c) NL = 0.01.

interactions (*i.e.*, precision level = 100%). As can be seen in Fig. 5.2, when the weight parameter decreases, AUPR increases and achieves similar values after $\theta = 0.1$; this shows that AUPR is quite robust against different choices of the weight parameter. On the contrary, IACC gradually decreases as the weight parameter decreases; this is because MTPLasso prevents solving wrong edges by sacrificing the accuracy. To obtain both high AUPR and IACC, we choose $\theta = 0.01$ that are robust for different graph structures for the following analyses. Additionally, the performance of our proposed method on cluster graph achieves the best results, while the least on random graph in terms of AUPR. Therefore, in the following analyses, we consider the random graph and scale-free graph which is commonly seen in many real-networks including microbial and gene networks.

5.3.3 Effects of prior percentage and precision level

To quantify the effects of different percentages of prior information and levels of precision, we vary the percentage of prior information that is included into the prior matrix ($\mathbf{P}$) and consider three precision levels: 0%, 50%, and 100%. Additionally, we consider one scenario with noiseless data (*i.e.*, $\epsilon(t) = 0, \forall t$) and two scenarios with noisy data where the noise levels are set to 0.001 and 0.01, respectively. We report two graph structures: random and scale-free as discussed in section 5.3.2.

As shown in Figs. 5.3 and 5.4, when the amount of prior information increases, both AUPR and IACC increase as expected. When the noise level increases, although with some degradation in performances (both in AUPR and IACC), our proposed method can still outperform other existing methods in terms of AUPR. For the IACC, ridge regression and LSA perform relatively well. However, they achieve poor results on AUPR since most of the inferred interactions are wrongly estimated. Another observation is that MTPLasso

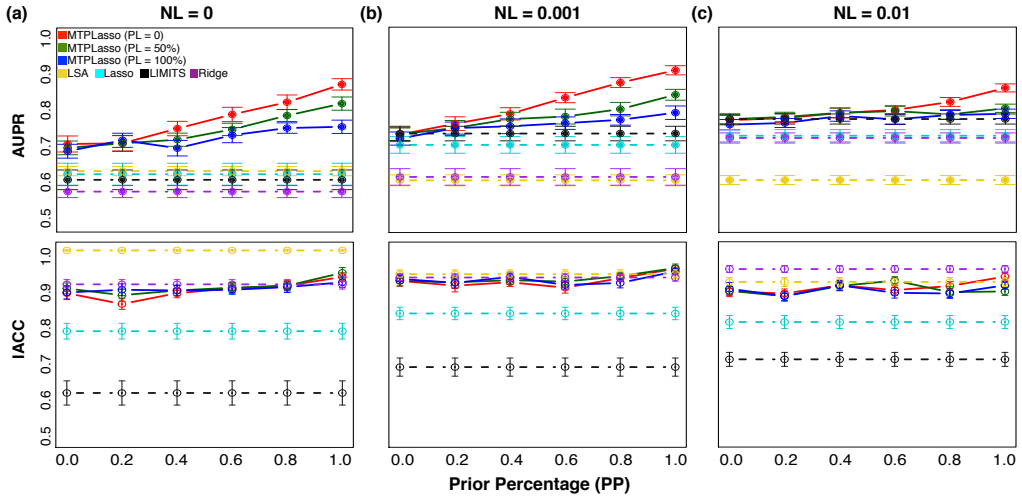

Figure 5.4: Performance evaluation on scale-free graphs under different combinations of parameters. Similar to the results on random graphs, both AUPR and IACC increase when PP increases. Compared to random graphs, scale-free graphs have better performances for all the methods. As it can be seen, MTPLasso outperforms the other methods even with high precision level (PL) and high noise level (NL). (a) NL = 0, (b) NL = 0.001, (c) NL = 0.01.

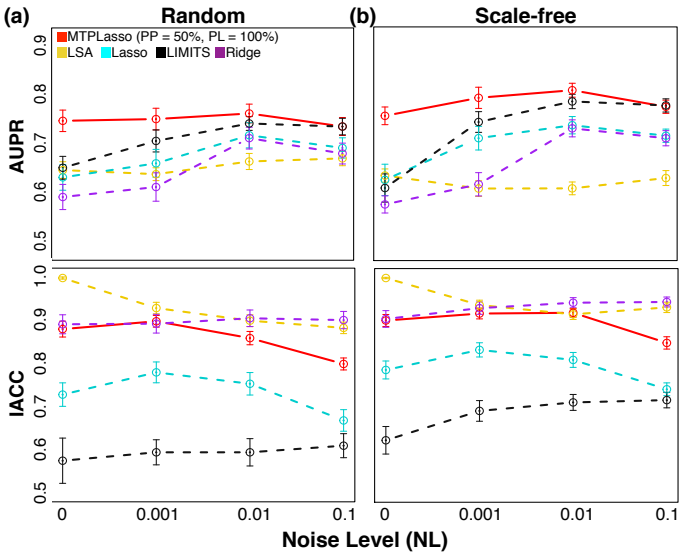

Figure 5.5: The effect of noise levels on (a) random and (b) scale-free graphs. As noise level increases, our proposed method degrades slightly ($\sim$0.05). The performance on scale-free graphs is better than than random graphs.

is relatively robust to different precision levels; this can be found with small differences in IACC.

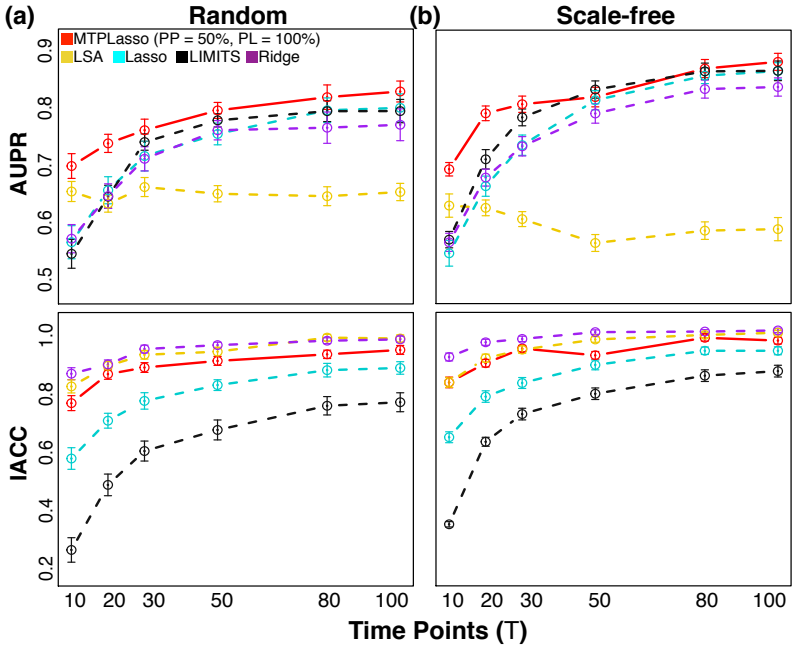

Figure 5.6: Performance evaluation on different lengths of time for (a) random and (b) scale-free graphs. When the length of time increases, all methods achieve better results for both AUPR and IACC, as expected. We observe that, however, at low length values, MTPLasso can still achieve AUPR = 0.7 and IACC = 0.8.

5.3.4 Effects of noise level

In the real-world settings, the metagenomic data can contain different types of noise. One main source of noise comes from the measurement errors. For example, genus i may be misclassified as genus j. To quantify such measurement errors, the additive noise included in Eq. (5.2) is set to different noise levels. More specifically, we consider error ranges from 0.001 to 0.1.[5] As shown in Fig. 5.5, our proposed method is robust against the measurement errors and outperforms other methods with an average AUPR = 0.75. In terms of IACC, the performance slightly degrades as the noise level increases; this is because noise makes it harder to infer the types of interaction (while we can still infer interacting genera). Overall, our proposed method is quite robust to different levels of noise.

5.3.5 Effects of number of time points

For a microbial network that contains p genera, the interaction matrix has up to p^2 free parameters. Thus, in general, we would need at least p^2 time points to infer the interactions from p genera. However, in microbial datasets, samples (T) are often less than number of genera (p). To examine the effect of different lengths of time, we vary the sample size from as small as $T = 10$, all the way up to $T = 100$. As we can see in

[5]For larger noise levels, the LV dynamical system becomes unstable.

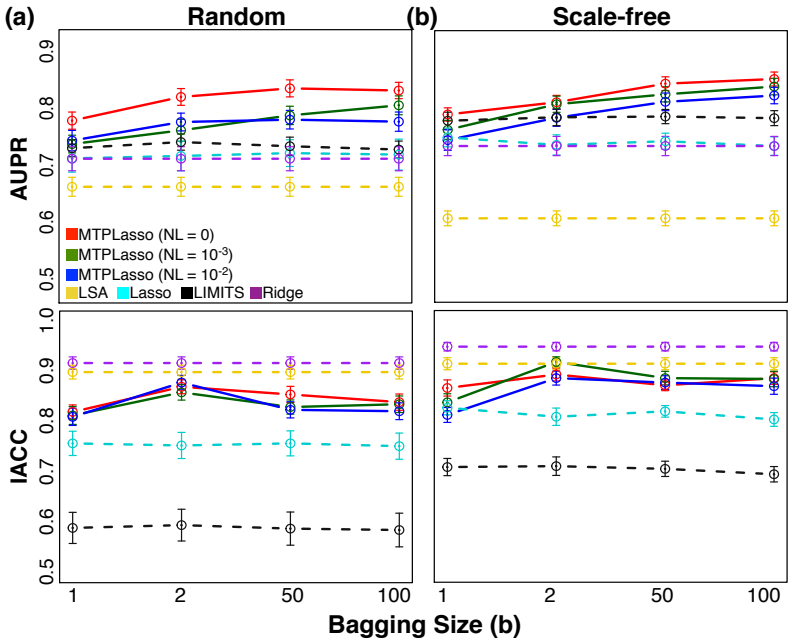

Figure 5.7: Performance evaluation on different bagging sizes for (a) random and (b) scale-free graphs. As it can be seen for both graphs, bagging can increase the performance and stabilize the inference results.

Fig. 5.6, our proposed method still outperforms the other existing methods in terms of AUPR even at small sample sizes. As the sample size increases, MTPLasso can achieve up to AUPR = 0.8. Although ridge regression can more accurately predict the interaction types, the AUPR is much lower than MTPLasso; over all, MTPLasso can achieve high performances both in random and scale-free graphs.

5.3.6 Effects of bagging size

To evaluate the effect of bagging, we vary the bagging size (b), where b = 1 represents no bagging at all. As it can be seen in Fig. 5.7, for both random and scale-free graphs, bagging can increase AUPR with a jump from $b = 1$ to $b = 2$. However, for IACC, there is not much difference when we increase the bagging size; this shows that MTPLasso is quite robust in inferring the type of interaction.

5.4 Experiments with real data

To explore the applicability of our proposed framework, we utilize data from [2] to examine the interactions among genera. Acquisitions and pre-processing for genus-level abundance data are described in **Appendix** 8.12. We utilize the microbial associations found by the HMP cross-sectional datasets as our prior information and set up a precision level = 50% (since it may still contain spurious information). In the end, we report the

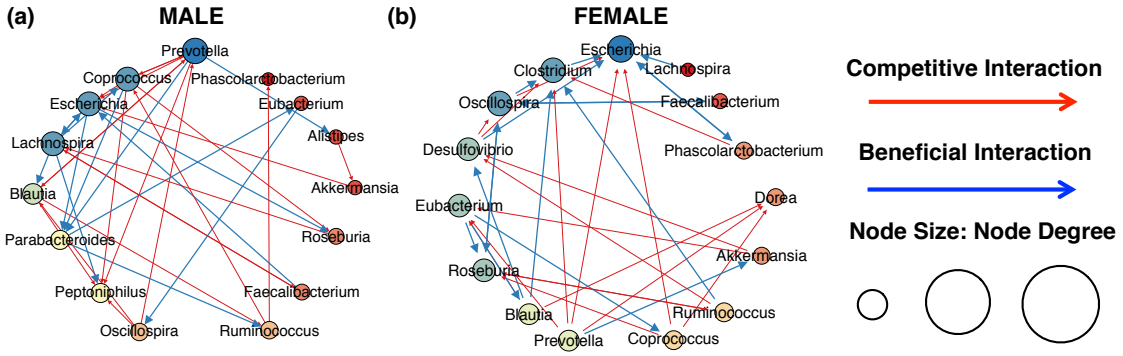

Figure 5.8: Interaction network visualization for highest degree nodes in the human gut of two individuals: (a) MALE and (b) FEMALE. Node size represents the relative node degree within the association network with counterclockwise layout. The red and blue edges represent competitive and beneficial interactions, respectively. Arrows represent the direction of the interaction.

inferred interactions in the gut microbiome of two individuals (*i.e.*, MALE and FEMALE); the resulting *directed* interaction networks are shown in Fig. 5.8.

Based on the node degree, the genus-level interaction networks for individual MALE and FEMALE are dominated by *Escherichia* and *Prevotella*, respectively; these two genera are prevalent in human gut. Notably, *Prevotella* was proposed as one of the three main genera representing the human gut microbiome, namely enterotypes [111]. The number of beneficial and competitive interactions is found to be similar. All interactions found in both graphs in Fig. 5.8 (*e.g.*, $<$*Escherichia*, *Prevotella*$>$) are inferred to be the same type (*i.e.*, competitive interactions); this shows that MTPLasso is robust at inferring interaction types.

If we take a deeper look, we note that the interaction pair $<$*Prevotella*, *Alistipes*$>$ found in individual MALE is classified as beneficial. Additionally, both *Prevotella* and *Alistipes* both have been found abundant in healthy subjects [112]; this may suggest that one of them may be beneficial to another. Another example lies in the competitive interaction pair $<$*Coprococcus*, *Roseburia*$>$ found in individual FEMALE. A recent study [113] has shown that *Coprococcus* and *Roseburia* are important promoters for production of certain chemicals. Therefore, there may exist competition among these two genera in order to gain growth advantage over another.

Additionally, we notice that the dominant genera in both individuals are not the most abundant ones. For example, *Prevotella* has the highest node degree in individual MALE, which is only the fifth abundant species. On the contrary, the node degree of the most abundant genus *Faecalibacterium* is only two in the inferred interaction network; this may suggest that abundant genera do not necessary have more interaction with other genera. The key is the trend of interacting time-series that can determine the type and the strength of interactions among genera. Further experimental results are still needed to validate the interactions inferred by our proposed algorithm.

5.5 Conclusion

Advances of high-throughput sequencing techniques enable us to gather metagenomic sequencing data from different environment and human niches. The available high-throughput experimental data, however, are compositional and high-dimensional in nature. Inferring interactions among microbes and understanding their influence on the human body is an important step towards precision medicine.

Existing microbial interaction network inferring methods either require the data of biomass or use greedy methods to obtain sparse estimations to overcome problems caused by high-dimensional data. However, all these prior approaches do *not* consider the information that can be obtained from the cross-sectional data and scientific literature to directly describe the associations among microbes and their co-occurrences, respectively.

By integrating multiple levels of biological information into the statistical models, we have shown that one can dramatically increase model's precision and recall rates, as well as interaction type classification accuracy. To the best of knowledge, this is the first work to propose an automated pipeline to infer the interactions on microbial data, show its feasibility, and measure performance metrics on both synthetic and real datasets.

We have also shown that our proposed algorithm *Microbial Time-series Prior Lasso (MTPLasso)* outperforms other existing methods (considered in this chapter) in terms of AUPR when using synthetic data with different graph structures which simulate different levels of sparsity. We have also evaluated several different factors including different noise levels, sample sizes, and bagging sizes to demonstrate the applicability of our approach under different conditions; we have also suggested rough guidelines for requisite sample size for the real data for the given assumption of the underlying graph structures.

Embolden by the success of our proposed algorithm on synthetic data, we have applied MTPLasso to infer the genus-level interactions in the human gut of two individuals. Our analyses on real time-series data show that interactions found in both graphs are classified to be the same interaction type. Additionally, we found that *Prevotella* (which represents one of the enterotypes in human gut) is a high degree node of the interaction network. Although some of the inferred interactions have not yet been directly found in laboratory settings, our predictions may still suggest credible directions for possible interactions without exhaustively searching all interaction pairs.

Finally, the estimated microbial interaction networks on real datasets can be used to understand why and how various eco-systems evolve over time. By plugging in the inferred interactions into the dynamic models (such as Lotka-Volterra model), we can even quantify the effects of different types of perturbations from other microbes such as probiotics. Overall, MTPLasso shows promising results. We envision that our model can reveal the underlying dynamics of the microbial compositional data in often high-dimensional settings.

Chapter 6

MetaNN: Accurate Classification of Host Phenotypes From Metagenomic Data Using Neural Networks

In our previous studies, we have developed algorithms to infer microbial interactions and predict their future dynamics based on metagenomic data. However, healthy individuals and subjects with disease may exhibit completely different microbiome community structures and dynamics. To fulfill the goal of personalized treatments, the first step is to identify the disease states of an individual based on their microbial profile, and then conduct further analysis such as target treatment. However, the high-dimensional nature of metagenomic data poses a significant challenge to existing machine learning models. Consequently, to enable personalized treatments, an efficient framework that can accurately and robustly differentiate between healthy and sick microbiome profiles is needed. In this chapter, we propose MetaNN (*i.e.*, classification of host phenotypes from *Metagenomic* data using *Neural Networks*), a neural network framework which utilizes a new data augmentation technique to mitigate the effects of data over-fitting. We show that MetaNN outperforms existing state-of-the-art models in terms of classification accuracy for both synthetic and real metagenomic data. These results pave the way towards developing personalized treatments for microbiome related diseases.

6.1 Introduction and motivation

Due to recent advances in modern metagenomic sequencing methods, several studies have characterized and identified different microbiome profiles in healthy and sick individuals for a variety of microbiome related diseases. For example, for the inflammatory bowel disease (IBD) which affects approximately 1.8 million Americans, it has been shown that individuals have about (30-50)% less biodiversity of commensal bacteria (*e.g.*, *Firmicutes* and *Bacteroidetes*) compared to healthy individuals [114]. Another example is

the Type 2 diabetes (T2D) which affects approximately 29.1 million Americans and costs the healthcare system about 245 billion dollars annually. T2D patients show significant changes in the 190 operational taxonomic units (OTUs), particularly a high abundance of *Enterobacteriaceae* compared to a healthy control group [115]. As a consequence, such differences in the microbiome profiles can be used as a diagnostic tool to differentiate the disease states of an individual. Being able to accurately differentiate the disease states for an individual can ultimately pave the way towards precision medicine for many microbiome related diseases.

A common and widely used approach to characterize the human microbiome profile relies on using the 16S rRNA gene as the taxonomic maker. Indeed, based on this profiling technique, previous studies have used unsupervised learning techniques such as clustering and principal coordinates analysis (PCoA) to perform classical hypothesis testing in order to classify microbial samples [116]. However, these methods are limited in their ability to classify unlabeled data or extract salient features from highly complex or sparse data; consequently, many supervised learning methods have been designed specifically for such classification purposes. For instance, several studies have shown that one can successfully identify differences in the microbiome profile or function of different host phenotypes such as body site, subject, and age [117, 118].

In terms of classification methods, machine learning (ML) models are powerful tools for identifying patterns in highly complex data, including human metagenomic data. In particular, supervised learning methods have been widely used for classification tasks in different areas such as image, text, and bioinformatics analyses [118]. For a typical supervised classification task, each training data point (sample) consists of a set of input features (*e.g.*, relative abundance of taxa) and a qualitative dependent variable giving the correct classification for that data point. For example, microbial samples from human body sites may be labeled as gut, mouth, or skin [119]. The goal of supervised learning is then to develop predictive models (or functions) from training data that can be used to assign the correct class (or category) labels to new samples.

Challenges of host phenotypes classification stem from the very nature of the high dimensionality of the metagenomic data. For instance, a typical dataset may contain few hundred samples, but thousands of OTUs (*i.e.*, features); this large number of features can greatly challenge the classification accuracy of any method and compound the problem of choosing the important features to focus on. Although several ML-based supervised classification algorithms, such as random forest [120], have been successful at classifying microbial samples [118], their classification accuracy remains poor, at least for some datasets [117]. As a consequence, new ML models are needed to improve the classification accuracy.

Recent advances in deep learning have shown significant improvements on several supervised learning tasks such as image classification and object detection [121]. Neural networks (NNs) consist of multiple (non-linear) hidden layers which make them expressive models that can learn complicated relationships between the system inputs and outputs. However, NNs usually require a large amount of training instances to obtain a reasonable classification accuracy and prevent over-fitting of training data. For instance, we need at least tens of thousands of images for a typical image classification task like

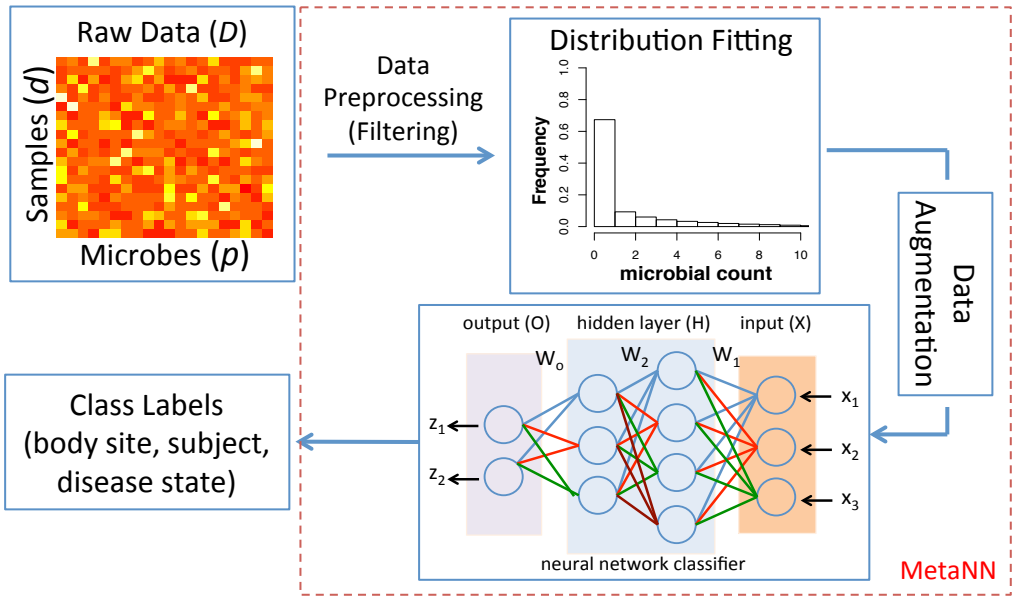

Figure 6.1: Our proposed MetaNN framework for the classification of metagenomic data. Given the raw metagenomic count data, we first filter out microbes that appear in less than 10% of total samples for each dataset. Next, we use negative binomial (NB) distribution to fit the training data, and then sample the fitted distribution to generate microbial samples to augment the training set. The augmented samples along with the training set are used to train a neural network classifier. In this example, the neural network takes counts of three microbes (x_1, x_2, x_3) as input features and outputs the probability of two class labels (z_1, z_2). The intermediate layers are hidden layers each with four and three hidden units, respectively. The input for each layer is calculated by the output of the previous layer and multiplied by the weights (W_1, W_2, W_o) on the connected lines. Finally, we evaluate our proposed neural network classifier on synthetic and real datasets based on different metrics and compare outputs against several existing machine learning models (see Section 6.2).

ImageNet [121]. To the best of our knowledge, we are the first to propose NN models that can be used to classify metagenomic data with small (*e.g.,* in the order of hundreds) microbial sample datasets; this is a challenging problem as the low count of samples can cause data over-fitting, hence degradation of the classification accuracy.

To overcome the problem of data over-fitting, we first consider two different NN models, namely, a multilayer perceptron (MLP) and a convolutional neural network (CNN), with design restrictions on the number of hidden layers and hidden units. Second, we propose to model the microbiome profiles with a negative binomial (NB) distribution and then sample the fitted NB distribution to generate an augmented dataset of training samples. Additionally, we adopt the dropout technique to randomly drop units along with their connections from NNs during training [122]. Data augmentation and dropout can effectively mitigate data over-fitting as we demonstrate in our experiments and analyses.

Finally, to assess the performance of different ML models, we propose a new simulation method that can generate synthetic microbial samples based on NB distributions which are commonly used to model the microbial count data [123]. As a result, the gener-

Dataset	# of samples	# of features	# of classes	Classification task
Classification of body sites				
Costello *et al.* (2009) Body Habitat (CBH)	552	1454	6	Classify body habitats: skin (357), oral cavity (46), External Auditory Canal (44), Hair (14), Nostril (46), Feces (45)
Costello *et al.* (2009) Skin Sites (CSS)	357	600	12	Classify skin sites: external nose (14), forehead (32), glans penis (8), labia minora (6), axilla (28), pinna (27), palm (64), palmar index finger (28), plantar foot (64), popliteal fossa (46), velar forearm (28), umbilicus (12)
Human Microbiome Project (HMP)	1025	323	5	Classify 5 major body sites: anterior nares (269), buccal mucosa (312), stool (319), supragingival plaque (313), tongue dorsum (316)
Classification of subjects				
Costello *et al.* (2009) Subject (CS)	140	464	7	Classify 7 subjects: (20, 20, 20, 20, 20, 20, 20)
Fierer *et al.* (2010) Subject (FS)	104	294	3	Classify 3 subjects: (40, 33, 31)
Fierer *et al.* (2010) Subject x Hand (FSH)	98	294	6	Classify by subject and left/right hand: (20, 18, 17, 14, 16, 13)
Classification of disease states				
Inflammatory Bowel Disease (IBD)	1025	1025	2	Classify disease states: normal (500), IBD (500)
Pei *et al.* (2013) Diagnosis (PDX)	200	5955	4	Classify disease states: normal (28), reflux esophagitis (36), Barrett's esophagus (84), esophageal adenocarcinoma (52)

Table 6.1: Real metagenomic data used in this chapter. We consider three different categories of classification aims: body sites, subjects, and disease states. Number of samples for a particular class is included between the round brackets. The number of features equals the number of different OTUs (*i.e.,* microbes).

ated samples consist of distinct microbiome profiles and particular class labels associated with them. To account for the noise in real microbial data, we consider several sources of measurement errors; this can be used to compare the performance of different ML models and identify scenarios that may degrade the classification accuracy significantly.

We test our framework on eight real datasets, *i.e.,* five benchmarks proposed in [118], one example from HMP [119], and two diseases, *i.e.,* inflammatory bowel disease [124] and esophagus [125]. We show that by augmenting the metagenomic data and using the dropout technique during training, the classification performance for the MLP classifier gets significantly better compared to all other existing methods for seven (out of eight) real datasets for two performance metrics commonly used to evaluate classification models: Area under the receiver operating characteristics (ROC) curve (AUC), and F1 score of class label predictions [126].

6.2 Review of machine learning methods

We compare and contrast different (multicategory) ML classification models: Support vector machines (SVM) [94], regularized logistic regression (LR) [127], gradient boosting (GB) [128], random forest (RF) [120], multinomial Naïve Bayes (MNB) [129] because of their wide and successful application to many datasets from other genomic applications[1].

Since most of these classifiers are designed for binary classification (*i.e.*, have only two output classes), we adopt a *one-versus-rest* type of approach where we train separate binary classifiers for each class against the rest of data and then classify the new samples by taking a vote of the binary classifiers and choosing the class with the 'strongest' vote. The *one-versus-rest* type of approach for classification is known to be among the best performing methods for multicategory classification [117].

[1] All the above methods are implemented with scikit-learn (http://scikit-learn.org/stable/) in Python.

6.2.1 Support vector machines (SVMs)

SVMs perform classification by separating different classes in the data using a maximal margin hyperplane [130]. To learn non-linear decision boundaries, SVMs implicitly map data to a higher dimensional space by means of a kernel function, where a separating hyperplane is then sought. The superior empirical performance of SVMs in many types of high-throughput biomedical data can be explained by several theoretical reasons: SVMs are robust to high variable-sample ratios and large number of features; they can efficiently learn complex classification functions and employ powerful regularization principles to avoid data over-fitting [131].

6.2.2 Regularized logistic regression (LR)

LR is a learning method from the class of general linear models that learns a set of weights that can be used to predict the probability that a sample belongs to a given class [18]. Typically, we can add either a L_1 or L_2 penalty to the LR to regularize and select important features. The weights are learned by minimizing a log-likelihood loss function. An L_2 penalty favors solutions with relatively small coefficients, but does not discard any features. An L_1 penalty shrinks the weights more uniformly and can set weights to zero, effectively performing embedded feature selection. We consider both regularizations in our subsequent experiments.

6.2.3 Gradient boosting (GB)

GB is a machine learning technique for regression and classification problems which produces a prediction model as an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and then generalizes them by allowing optimization of an arbitrary differentiable loss function; this is achieved by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction.

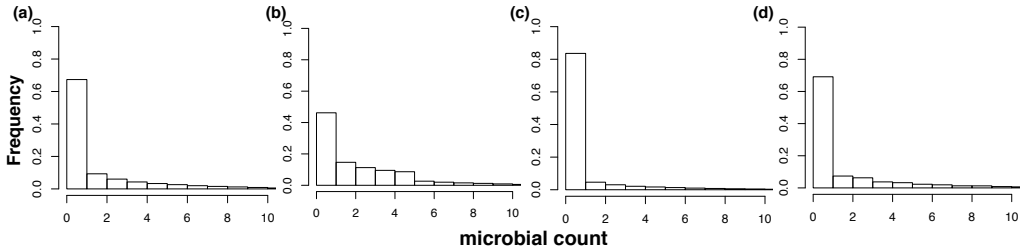

Figure 6.2: Synthetic microbial frequency count distribution generated using NB distribution based on microbiome profiles. (a) The underlying true distribution which is highly zero inflated (*i.e.,* no presence of certain microbe). (b) Type 1 error that adds non-zero noise to the zero count entries in order to change the distribution. (c) Type 2 error that changes the underlying non-zero entries to zeros. (d) Type 3 error changes the distribution of non-zeros counts. Note that all different types of errors are added with probability of 0.5.

6.2.4 Random forests (RF)

RF is a classification algorithm that uses an ensemble of unpruned decision trees, each built on a bootstrap sample of the training data using a randomly selected subset of features [120]. The RF algorithm possesses a number of appealing properties making it well-suited for classification of metagenomic data: (i) it is applicable when there are more predictors (features) than observations; (ii) it performs embedded feature selection and it is relatively insensitive to the large number of irrelevant features; (iii) it incorporates interactions between predictors: (iv) it is based on the theory of ensemble learning that allows the algorithm to learn accurately both simple and complex classification functions; (v) it is applicable for both binary and multicategory classification tasks; and (vi) according to its inventors, it does not require much fine tuning of hyperparameters and the default parameterization often leads to excellent classification accuracy.

6.2.5 Multinomial Naïve Bayes (MNB)

MNB classifier is suitable for classification with discrete features (*e.g.*, word counts for text classification). Hence, MNB is usually used to classify topics (*i.e.*, class labels) among sentences. For microbial data, a class can contain a mixture of OTUs that is shared among samples. Therefore, we can learn the microbiome mixture conditioned on the class labels.

6.3 Data filtering and generation

6.3.1 Acquisition and preprocessing of metagenomic data

In this section, we utilize the high-quality sequencing reads in 16S rRNA variable regions. The taxonomy (OTU) identification of the 16S rRNA is performed using different pipelines for eight different datasets as summarized in Table 6.1. The datasets CBH, CS, CSS, FS, FSH are obtained from the study of [118] and originate from the work of [132] and [133]. The HMP dataset is obtained from the high-quality sequencing reads in 16S variable regions 3-5 (V35) of HMP healthy individuals with taxonomy identification done by the QIIME [91] pipeline. The PDX dataset is obtained from [117] and originate from the work of [125].

The resulting OTU table can be represented by a matrix $D \in \mathbb{N}^{n \times p}$ where $\mathbb{N}$ is the set of natural numbers; n and p represent number of samples and number of microbes, respectively. $d^i = [d^i_1, d^i_2, \ldots, d^i_p]$ denote the p-dimensional row vector of OTU counts from the i^{th} sample ($i = 1, \ldots, n$). The total cumulative count for the i^{th} sample can be expressed as $s^i = \sum_{k=1}^{p} d^i_k$. To account for the different sequencing depth of each sample, the raw count data (d^i) are typically normalized by the cumulative count (s^i) which results in *relative* abundances (or profiles) vector $x^i = [\frac{d^i_1}{s^i}, \frac{d^i_2}{s^i}, \ldots, \frac{d^i_p}{s^i}]$ for any sample i. These relative taxonomy abundances are further rescaled in the range [0, 1] and serve as input features for the ML models. Note that the OTU abundance table is constructed

without any knowledge of the classification labels and thus data preprocessing does *not* influence the performance of ML models.

6.3.2 Modeling the microbiome profile

For biological samples, there exist multiple sources (*e.g.*, biological replication and library preparation) that can cause variability of features [123]. In oder to account for such effects, recent work suggests to use the mixture model to account for the added uncertainty [134]. Taking a hierarchical model approach with the Gamma-Poisson distribution has provided a satisfactory fit to RNA sequencing data [135]. A Gamma mixture of Poisson variables gives a negative binomial (NB) distribution [136] which is more appropriate for handling data overdispersion (*e.g.*, microbial count data is highly zero inflated). As a result, we can simulate and generate augmented samples which consists of unnormalized microbial counts. We then use the same preprocessing procedure (described in Section 6.3.1) to normalize the augmented samples before training our classifiers.

To generate a NB sample, we first assume the mean of the Poisson distribution (λ)[2] to be a Gamma-distributed random variable $\Gamma(r, \theta)$ with shape parameter r and scale $\theta = p/(1-p)$. Note that by construction, the values of r and θ are greater than zero. Next, we sample the Poisson mean λ from this Gamma distribution. Finally, we sample the NB random variable from Pois($u; \lambda$). The compact form of the mass distribution of a discrete NB random variable (v) then reads as:

$$\mathrm{NB}(v; r, p) = \frac{\Gamma(r+v)}{v!\Gamma(r)} p^v (1-p)^r \tag{6.1}$$

where Γ is the gamma function and the data overdispersion is controlled by the parameter r[3]. Note that, samples of a given class are assumed to be independent and identically distributed (from one NB distribution). Therefore, we fit a NB distribution for each class. More specifically, we fit the model parameters r and θ based on the OTU count vector (*i.e.,* d) and the covariance of OTUs (*i.e.,* features) using the maximum likelihood estimation [123]. As a consequence, we are able to capture the dependence of OTUs for different classes.

6.3.3 Synthetic data generation

In order to quantitatively evaluate different ML models for classifying microbial samples, we first generate synthetic microbial data that consider multiple sources of measurement errors. More specifically, we first determine the number of classes of interest and then randomly generate the microbiome profile for each class. Next, we sample the microbial count data for each class independently based on the NB distribution and the previously generated microbiome profile. To account for the variability in the real data, we consider three types of errors in measuring the 16S rRNA sequencing data:

[2]The probability mass function of Poisson distribution is given by: Pois$(u; \lambda) = \frac{\lambda^u e^{-\lambda}}{u!}$, where u is a random variable and λ determines both the mean and variance.

[3]The NB model reduces to the standard Poisson model for $r \to \infty$.

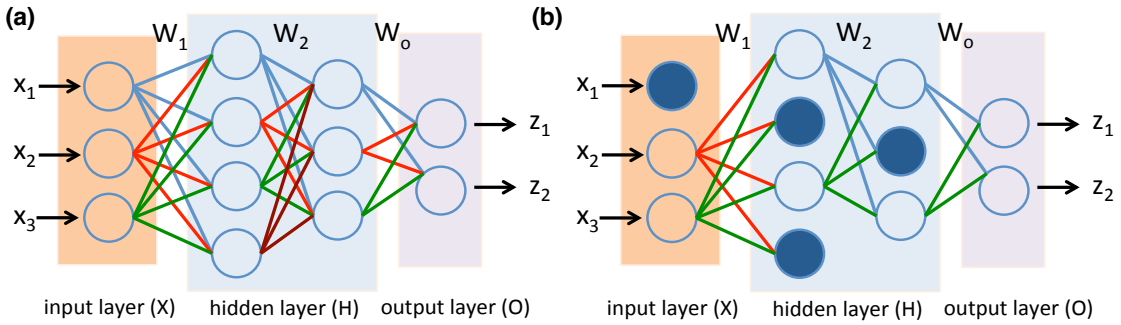

Figure 6.3: Illustration of random dropout where dropout units are shown as blue filled circles. (a) No dropout. (b) With dropout. As it can be seen, connections to the dropout units are also disabled. Since we randomly choose dropout units in NNs, this means we effectively combine exponentially many different NN architectures to prevent data over-fitting.

	Synthetic	CBH	CSS	HMP	CS	FS	FSH	IBD	PDX
MLP	(256, 256)	(1024, 512)	(512, 256)	(512, 256)	(512, 512)	(512, 512)	(512, 256)	(512, 256, 128)	(512, 256, 128)
CNN	Conv1D(8, 3) → Dropout → ReLu → MaxPool1D(2) → Conv1D(8, 3) → ReLu → MaxPool1D(2) → FC								

Table 6.2: Model configurations for MLP and CNN. Number in the round bracket represents the number of hidden units. Conv1D is the one-dimensional convolution layer. ReLu is the non-linear rectifier layer. MaxPool1D represents the one-dimensional max pooling layer. Dropout and FC represent dropout and fully connected layers, respectively. Details of each dataset are described in Table 6.1.

- Type 1 error (e_1): the underlying true count is zero ($d = 0$) but the measurement count is non-zero ($\hat{d} \neq 0$).
- Type 2 error (e_2): the underlying true count is non-zero ($d \neq 0$) but the measurement count is zero ($\hat{d} = 0$).
- Type 3 error (e_3): the underlying true count is non-zero ($d \neq 0$) but with a deviation/fluctuation from the true count ($\hat{d} = d +$ noise).

We generate synthetic data with random combinations of error probabilities $[e_1, e_2, e_3]$. For example, if $e_1 = 0.5, e_2 = 0.3, e_3 = 0.2$, we have a probability of 0.5 to add microbial counts to the zero count entries of the underlying true microbial count data. Similarly, for Type 2 and 3 errors, we set the non-zero count to zero with probability of 0.3 and add deviation or fluctuation counts to the non-zero count data with probability of 0.2, respectively.

As shown in Fig. 6.2, we can see that three different error types can dramatically change the underlying true count distribution. We evaluate the effects of different combinations of error types on the performance of ML models, as well as multilayer perceptron (MLP) and convolutional neural network (CNN); results are presented later in Section 6.5.

6.4 MetaNN framework

As shown in Fig. 6.1, our proposed framework, MetaNN, consists of two important components: First, a new model based on neural networks that is well-suited for classifying metagenomic data. Second, our proposed data augmentation for the microbial count data and adopted dropout training technique that can effectively mitigate the problem of data over-fitting.

6.4.1 Multilayer perceptron (MLP)

We consider MLP [137] models with design restrictions on the number of hidden layer and hidden unit in order to prevent over-fitting of the microbial data. To this end, we consider two or three hidden layers where each hidden unit is a neuron that uses a nonlinear activation function; this distinguishes MLP from a linear perceptron. Therefore, it is possible to distinguish data that is not linearly separable.

More specifically, MLP uses a supervised learning algorithm that learns a function $f(\cdot) : R^m \rightarrow R^o$ by training on a dataset, where m is the number of input dimensions and o is the number of output dimension. Given a set of features $X = (x_1, x_2, ..., x_m)$ and a target $Z = (z_1, z_2, ..., z_o)$, MLP can learn a non-linear function approximator for either classification or regression; this is different from logistic regression, in that between the input and the output layers, there can exist one or more non-linear layers (hidden layers).

As shown in Fig. 6.3(a), the leftmost layer, known as the input layer, consists of a set of neurons $X = (x_1, x_2, x_3)$ representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $H_1 = W_1 X$, followed by a non-linear activation function $g(\cdot) : R \rightarrow R$ - like the Rectifier function (*i.e.*, $g(x) = max(0, x)$). The output layer receives the values from the last hidden layer (H_2) and multiplies them with the output weights (W_o) hence the output values as $Z = (z_1, z_2) = W_o H_2$.

To train the MLP if there exist more than two classes, the output layer is the softmax function which is written as:

$$\hat{z}_k = \text{softmax}(z_k) = \frac{\exp(z_k)}{\sum_{l=1}^{k} \exp(z_l)} \tag{6.2}$$

where $\hat{z}_k$ represents the estimated probability of having class k. Consequently, the predicted label $\hat{y} = \max_k \hat{z}_k$ is the class with the highest probability. The training objective (loss function) is a cross entropy loss [138] which is represented by:

$$J = -\sum_{i}^{N} \sum_{k}^{K} y^{(i)} \log \hat{z}_k^{(i)} \tag{6.3}$$

where N is the number of training samples and K is the total number of classes. $y^{(i)}$ is the true class label for sample i. $z_k^{(i)}$ is the probability of having class k for sample i.

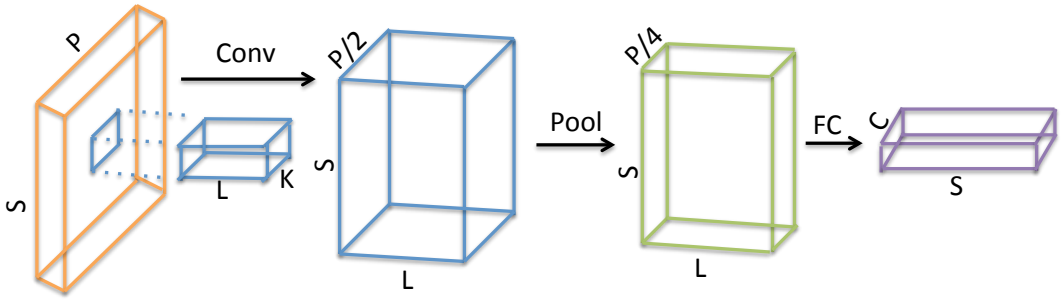

Figure 6.4: A regular convolutional neural network (CNN). The input consists of S samples and P features. The 1D filter with kernel size of K and L channels is used for convolving data with the input. By pooling (downsampling) with kernel size of 2, the resulting tensor now becomes approximately of size $S \times P/4 \times L$. The fully connected layer considers all the features in every channels and output the probability of class labels (C) for each sample.

6.4.2 Convolutional neural network (CNN)

The rationale of using CNN to extract local patterns of microbes is that prior studies have found that phylogenetically related microbes interact with each other and form functional groups [97]. Therefore, we arrange the bacterial species based on their taxonomic annotation, ordered alphabetically, by concatenating the strings of their taxonomy (*i.e.*, phylum, class, order, family, and genus). As a consequence, CNN is able to extract the evolutionary relationship based on the phylogenetic-sorting.

The hidden layers of a CNN typically consist of a set of convolutional layers (Conv), pooling layers (Pool), and fully connected layers (FC) [138]. As shown in Fig. 6.4, convolutional layer computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume (phylogenetic-sorted). The pooling layer performs a downsampling operation along the spatial dimensions. The fully connected layer computes the class scores which is the same as the output layer of MLP. In our implementation, we consider 1D convolutional and 1D pooling layers since each microbial sample is one dimensional. The training objective is the same as (6.3).

6.4.3 Data augmentation

Data augmentation has been widely used in computer vision communities [121]. For example, in image classification, images are cropped or rotated in order to augment the training set. Data augmentation is useful because it directly augments the input data to the model in data space; this idea can be traced back to augmentation performed on the MNIST set in [139].

Existing metagenomic datasets have fewer samples than the number of observed taxa (features); this makes it difficult to model complex interactions between taxa and differentiate the microbiome profiles. In order to deal with such problems, we propose to augment the microbial data with new samples generated from a known distribution. More specifically, we first use the NB distribution defined in Section 6.3.2 to fit the model

parameters of the microbiome profile of each class. Next, we use the fitted NB distribution to generate augmented samples for each class. The samples generated by the NB distribution can be viewed as variations in the data space that effectively mitigate the problem of data over-fitting. Note that we only fit the NB distribution to the training set of each split, and then feed both augmented and training datasets to our newly proposed NN classifiers.

6.4.4 Dropout

Dropout is a technique proposed to addresses data over-fitting [122], and provides a way of approximately combining exponentially many different neural network architectures efficiently. The term “dropout" refers to temporary dropping out units (hidden and visible) in the NNs, along with all its incoming and outgoing connections, as shown in Fig. 6.3(b).

The choice of which units to drop is random. In the simplest case, each unit is retained with a fixed probability q independent of all other units, where q can be simply set at 0.5. In our experimental settings, we use dropout at the input layer for both MLP and CNN with a dropout probability of 0.5, which is commonly used and close to optimal for a wide range of networks and tasks [122].

6.5 Experiments with synthetic data

To show the applicability of MLP and CNN models, we compare our model against several supervised classification ML models (as described in Section 6.2). This set of experiments serves as a proof of concept of quantifying the performance of each model by simulating synthetic data that account for different levels of measurement error in the real data.

6.5.1 Experimental setup

Hyperparameter configurations for MLP and CNN are described in Table 6.2. To train the model, we use softmax function (eq. (6.2)) as the output layer and the cross entropy loss (eq. (6.3)) for both MLP and CNN. We implement our MLP and CNN models in Pytorch (http://pytorch.org/) and use Adam [140] as our gradient optimizer with a default learning rate of 0.001 in the subsequent experiments. We fix the training epoch[4] to 100 and 200 for MLP and CNN to avoid data over-fitting, respectively. Note that for the synthetic experiments, we do *not* apply any training techniques (*i.e.*, data augmentation and dropout) during model training. The hyperparameters for MLP and CNN are reported in Table 6.2.

For SVM (Section 6.2.1), we first select either a linear and radial basis function (RBF, also known as Gaussian kernel) and then select the best regularization parameter and width parameter in the range of $[10^{-2}, ..., 10^{2}, 10^{3}]$ and $[10^{-5}, ..., 10^{1}]$, respectively, using a 3-fold cross-validation approach. For GB (Section 6.2.3), we set up a higher maximum

[4]One forward and one backward pass over all training instances.

F1-micro								
(e_1, e_2, e_3)	SVM	GB	RF	MNB	LR1	LR2	MLP	CNN
(0.5, 0.1, 0.4)	0.96	0.79	**0.98**	**0.98**	0.30	**0.98**	**0.98**	0.75
(0.5, 0.4, 0.1)	0.99	0.82	**1.00**	**1.00**	0.43	**1.00**	**1.00**	0.81
(0.3, 0.1, 0.4)	0.98	0.87	0.98	**0.99**	0.54	**0.99**	**0.99**	0.74
(0.0, 0.7, 0.2)	0.99	0.83	**1.00**	**1.00**	0.66	**1.00**	**1.00**	0.86
(0.0, 0.2, 0.7)	0.89	0.58	0.81	**0.91**	0.51	0.87	**0.91**	0.59

Table 6.3: Performance comparison of different ML and NN models for different types of error (e_1, e_2, e_3). We consider several existing supervised ML methods, as well as NN models (*i.e.*, MLP and CNN). For each experiment, we use 10-fold cross-validation. We use F1-micro to quantify the performance as defined in Section 6.5.2. Bold values represent the best results.

depth equal to 10; minimum samples split equal to 5 as a compromise between over-fitting and under-fitting the training set. For RF (Section 6.2.4), we set up the number of estimators equal to 200 (default is 10) to have a better estimation and then select the depth, sample splits, and number of leaves using 3-fold cross-validation. For MNB (Section 6.2.5), we fit a prior distribution to the number of OTUs in each class; this acts as a smoothing constant. For other ML methods and hyperparameters, we use the default values implemented in *scikit-learn*.

6.5.2 Classification performance metrics

We consider a few metrics as follows:

- Area under the Curve (AUC): We compute the area under ROC curve [5] where a larger area means a better classification model.
- F1-micro: We estimate F1-micro as the true positives plus the true negatives divided by the total number of samples; this is same definition of classification accuracy as widely used in binary classification problems.
- F1-macro: We estimate F1-macro by calculating the F1-micro for each class and then find their unweighted mean; this does not take label imbalance into account.
- Performance Gain: We calculate the performance gain as the F1 score of the best NN model minus the F1 score of the best ML models divided by the F1 score of the best ML models.

[5] Receiver operating characteristic (ROC) is determined by true and false positives. A true positive is the sample that is accurately classified. On the other hand, a false positive is the sample that is wrongly classified.

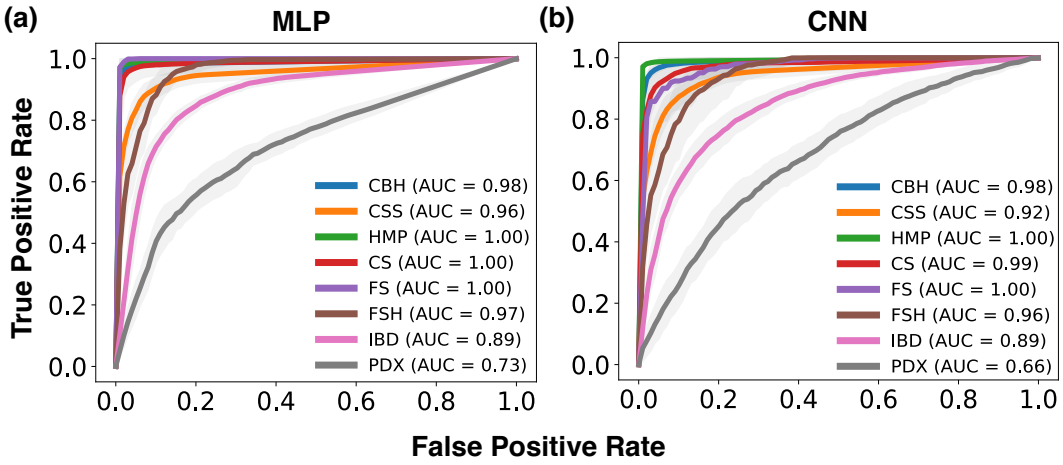

Figure 6.5: ROC curves and AUCs for (a) multilayer perceptron (MLP) and (b) convolutional neural network (CNN). True positive rates are averaged over 10-fold cross-validation each with 5 independent random runs. We show the ROC curves and AUCs for the real datasets considered in this chapter.

6.5.3 Classification performance comparisons

We consider eight classes each with different microbiome profiles[6]. For example, consider the case when the number of microbes is $p = 100$ for each class. For a particular microbiome profile (*e.g.,* $m = (30, 40, 30)$ microbes), we sample three different overdispersion parameters (*e.g.*, $r = (0.1, 1, 10)$) for the NB distribution, respectively. Next, we use r and sample the microbial counts based on eq. (6.1) and then alter the counts by adding different sources of errors with specific probabilities.

We report the results for eight classes where each class has $d = 100$ samples and $p = 100$ microbes. As shown in Table 6.3, when we fix the probability of Type 1 errors (e_1) to 0.5 and 0.0 and vary the probability of Type 2 (e_2) and Types 3 (e_3) errors, we find that the Type 3 errors are more severe than the Type 2 errors; this is because the Type 3 errors can dramatically change the microbial count distribution as shown in Fig. 6.2. We also find that the Type 1 errors have a moderate impact on the performance of each classifier.

We find that MLP and MNB achieve the best (and comparable) performance in all scenarios we considered; this is due to the fact that MLP is able to better deal with the sparse features since NNs can extract higher level features by utilizing hidden units in hidden layers. MNB fits the prior distribution for the microbiome profile of each class; this can largely improve performance since each class is generated based on the NB distribution which complies with the underlying assumptions of MNB. Overall, MLP is suitable to deal with different sources of errors. On the contrary, CNN is not able to deal with sparse features since the convolution layer considers spatial relationships among features; this results in its poor performance for the synthetic datasets.

[6]The generation process of synthetic data is discussed in Section 6.3.3.

6.6 Experiments on real data

We utilize several datasets (see Section 6.3.1) to examine the performance of different ML models in real scenarios. Datasets can be classified into three categories based on their properties: (1) Classification of body sites, (2) classification of subjects, and (3) classification of disease states. The total number of samples and features (*i.e.,* OTUs) are summarized in Table 6.1. We also list the model hyperparameters for MLP and CNN in Table 6.2. In our experimental settings, the number of augmented samples is set equal to the number of training samples, the dropout rate (q) is set to 0.5. We use the same set of hyperparameters for the other ML methods, as described in Section 6.5.1.

The performance of all the ML methods introduced in Section 6.2 is summarized in **Appendix** 8.13 and Table 8.8. Since SVM and RF have better performance over other ML methods, we choose these two methods to compare with our NN models in Table 6.4.

We first show the classification performance of MLP and CNN on different datasets using ROC curves. As shown in Fig. 6.5, MLP shows better performance than CNN; this implies that MLP is a better model since the activation function at the output layer is able to learn a better decision boundary. Additionally, we find that disease datasets (*i.e.*, IBD and PDX) are more difficult to classify. In the following sections, we present the experiment results for datasets in different categories.

					F1-macro				
Dataset	SVM	SVM+A	RF	RF+A	MLP+D	CNN+D	MLP+D+A	CNN+D+A	Gain (%)
CBH	0.78 (0.03)	0.82 (0.03)	0.73 (0.03)	0.75 (0.03)	0.85 (0.03)	0.77 (0.04)	**0.86 (0.03)**	0.82 (0.03)	5
CSS	0.63 (0.07)	0.65 (0.06)	0.58 (0.08)	0.61 (0.06)	0.66 (0.06)	0.59 (0.06)	**0.67 (0.06)**	0.62 (0.06)	3
HMP	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0
CS	0.88 (0.05)	0.88 (0.05)	0.87 (0.05)	0.87 (0.05)	0.92 (0.05)	0.87 (0.06)	**0.93 (0.05)**	0.88 (0.05)	6
FS	0.94 (0.03)	0.95 (0.02)	**1.00 (0.01)**	**1.00 (0.01)**	0.97 (0.03)	0.90 (0.15)	0.98 (0.02)	0.97 (0.02)	-2
FSH	0.68 (0.08)	0.70 (0.08)	0.63 (0.08)	0.68 (0.08)	**0.74 (0.06)**	0.66 (0.07)	**0.74 (0.05)**	0.72 (0.07)	6
IBD	0.68 (0.04)	0.72 (0.02)	0.57 (0.02)	0.60 (0.02)	**0.75 (0.02)**	0.67 (0.03)	**0.78 (0.02)**	0.70 (0.02)	8
PDX	0.29 (0.13)	0.43 (0.02)	0.28 (0.09)	0.34 (0.07)	0.51 (0.00)	0.44 (0.05)	**0.56 (0.03)**	0.45 (0.08)	30
					F1-micro				
Dataset	SVM	SVM+A	RF	RF+A	MLP+D	CNN+D	MLP+D+A	CNN+D+A	Gain (%)
CBH	0.93 (0.02)	0.93 (0.01)	0.91 (0.02)	0.92 (0.02)	**0.94 (0.01)**	0.89 (0.02)	**0.94 (0.01)**	0.92 (0.02)	1
CSS	0.71 (0.03)	0.72 (0.04)	0.67 (0.03)	0.68 (0.03)	0.72 (0.03)	0.67 (0.04)	**0.74 (0.03)**	0.68 (0.04)	3
HMP	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.96 (0.01)	0.97 (0.01)	0.97 (0.01)	0
CS	0.88 (0.06)	0.89 (0.05)	0.88 (0.04)	0.88 (0.05)	0.92 (0.04)	0.87 (0.06)	**0.94 (0.04)**	0.89 (0.05)	6
FS	0.94 (0.03)	0.95 (0.02)	**1.00 (0.01)**	**1.00 (0.01)**	0.97 (0.03)	0.91 (0.12)	0.98 (0.02)	0.97 (0.02)	-2
FSH	0.70 (0.08)	0.71 (0.07)	0.69 (0.05)	0.72 (0.06)	0.75 (0.05)	0.68 (0.06)	**0.76 (0.05)**	0.75 (0.07)	6
IBD	0.79 (0.02)	0.79 (0.02)	0.78 (0.02)	0.79 (0.02)	0.82 (0.01)	0.77 (0.02)	**0.84 (0.01)**	0.78 (0.02)	6
PDX	0.44 (0.07)	0.48 (0.03)	0.43 (0.07)	0.44 (0.06)	0.53 (0.01)	0.49 (0.05)	**0.56 (0.03)**	0.50 (0.06)	17

Table 6.4: Performance comparison of SVM, RF and NN models on eight real datasets described in Table 6.1. +D and +A means dropout and data augmentation, respectively. For each experiment, we consider 10-fold cross-validation and use F1-macro and F1-micro scores to quantify performance as defined in Section 6.5.2. For each fold, we perform five simulation runs with standard deviations shown between round brackets. Performance gains are shown for the best NN and the best ML models. Bold values show the best results.

6.6.1 Classification of body sites

In this set of experiments, we consider a total of three datasets: two came from [132] and one from HMP (see Table 6.1). As discussed in [118] and shown in Table 6.4 and Fig. 6.5, CSS is the most difficult dataset since the microbiome profiles are generally non-differentiable between different skin sites. For the other two datasets (*i.e.*, CBH and HMP), the microbiome profiles tend to be highly differentiated between different body sites; therefore, ML models do obtain a better classification performance. In practice, classification of body sites would not require the use of a predictive model for classification since we would most likely know the site of sampling. However, it is still valuable to use this category to evaluate the performance of different ML methods.

6.6.2 Classification of subjects

In this set of experiments, we consider three benchmark datasets where two come from [133] and one from [132]. As shown in Table 6.4 and Fig. 6.5, this category is more challenging than classifying body sites since the samples of certain subject may be collected at different time points. For the CS dataset, authors in [132] observed significant variations of microbiome profile for individuals over time and most ML models cannot achieve a high accuracy. On the contrary, for the FS dataset, individuals have clear differences since samples are collected at approximately the same time point. FSH dataset is more challenging compared to FS since we need to additionally classify the right and left hand for each individual.

6.6.3 Classification of disease states

In this set of experiments, we consider IBD and PDX datasets from [124] and [125], respectively. As shown in Table 6.1 and Table 6.4, PDX is a challenging dataset, since it contains four classes and the microbiome profiles are similar among these classes. Indeed, existing ML models can only achieve up to 40% accuracy (F1-micro score) of the PDX set.

6.6.4 Classification performance comparisons

As shown in Table 6.4, MLP with dropout and data augmentation (MLP+D+A) achieves the best performance in terms of F1-macro and F1-micro scores among all other ML methods, except the FS dataset. CNN with dropout and data augmentation (CNN+D+A) also provides comparable performance with other ML models. Note that without using data augmentation, MLP (MLP+D) still achieves the best performance against other ML models; this is because MLP can extract higher level features and automatically select the important features.

Other than MLP and CNN, SVM and RF also show better performance; this is because SVM and RF are able to distinguish features even in high dimensional settings while being robust to random features. However, MLP can still have significant average

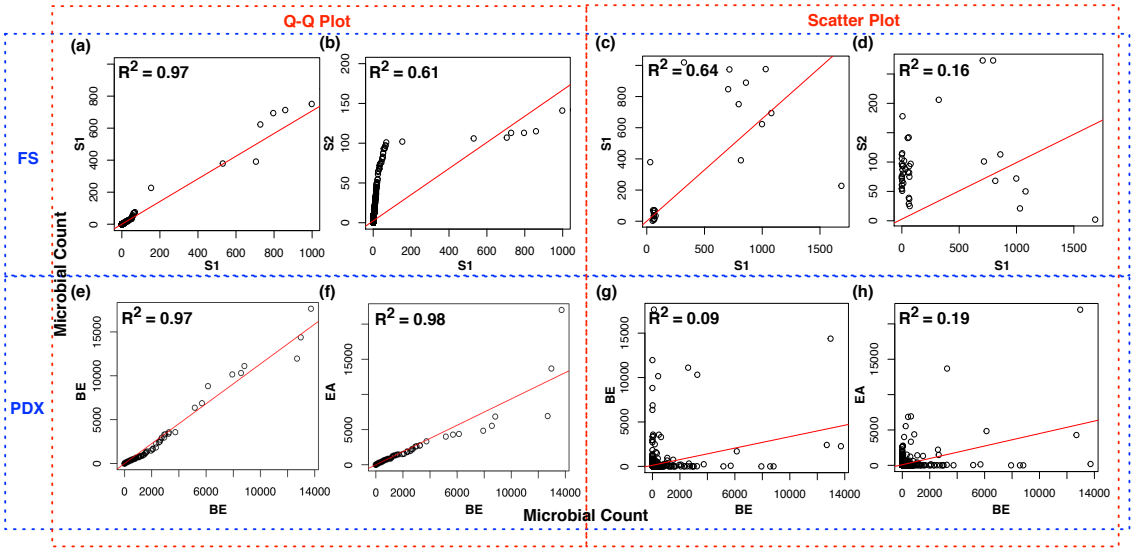

Figure 6.6: (a-b and e-f) Q-Q plots and (c-d and g-h) scatter plots for FS and PDX datasets, respectively. The red line is the linear fitted line with adjusted R square reported at the top-left corner. S1, S2 represent samples from subject 1 and subject 2, respectively. BE, EA represent samples from Barrett's esophagus (BE) and esophageal adenocarcinoma (EA) patients, respectively.

gains of 7% and 5% against the best ML method in terms of F1-macro and F1-micro, respectively. If we take a closer look at the disease datasets, we can see that the MLP+D+A has a dramatic increase in terms of F1-macro scores (8% and 30% gains) compared to other ML methods for both IBD and PDX datasets; this indicates that MetaNN can accurately differentiate and better classify various disease states.

As shown in Table 6.4, data augmentation can improve the classification performance not only for NN models but also for ML models. More specifically, we can have an average of 2-3% improvement compared to the one without using data augmentation; this shows that data augmentation in the training sets can truly leverage the high dimensionality of metagenomic data.

In terms of classification performance of ML methods listed in Table 6.4, we can see that ML methods can achieve up to 80-100% F1 scores for most of the datasets. For example, both MLP and RF can achieve up to 98% classification accuracy for the FS dataset. However, other challenging datasets, such as PDX and CSS have non-differentiable microbiome profiles. To support this claim, we utilize the (1) Q-Q (quantile-quantile) plot to quantify two distributions against each other, and (2) scatter plot to show the consistency of microbiome profiles between different classes.

Q-Q plot is generated based on the quantiles of two distributions, where quantile can be obtained by sorting the microbial counts. For example, Fig. 6.6(b) shows the quantile distributions of subject 1 (S1) against subject 2 (S2). On the contrary, the scatter plot is generated based on the (unsorted) microbiome profile. For example, a point on Fig. 6.6(d) represents a certain microbe (*e.g., E. coli*) found in both S1 and S2 samples

but with different counts.

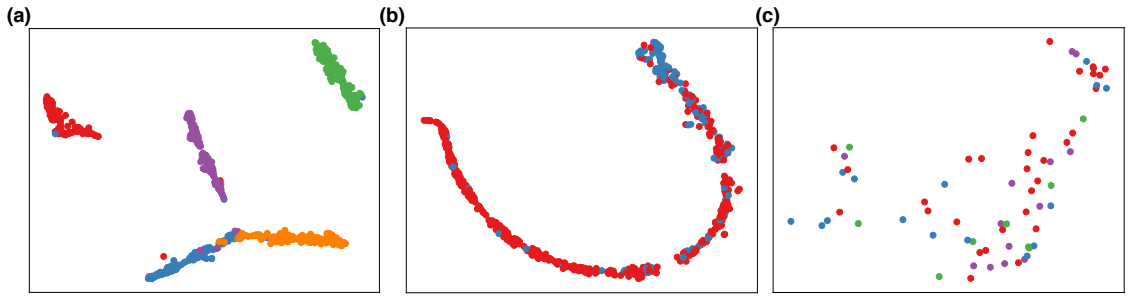

Figure 6.7: Visualization of (a) HMP, (b) IBD, and (c) PDX datasets using t-SNE projection [1]. We project the activation function of the last hidden layer of the test data onto a 2D space, where different colors represent different classes. For instance, the red and green colors represent samples collected from anterior nares and stools, respectively. As it can be seen, HMP and IBD samples show a clear separation between classes, while PDX samples are hard to be distinguished.

For the FS dataset, we first notice that subject 1 (S1) within-class distribution and profile are similar (Fig. 6.6(a, c)) as opposed to between-class case (Fig. 6.6(b, d)); these distinct differences make the FS dataset easy to classify. However, for the PDX dataset, we can see that the distribution and profiles of PDX dataset show completely different behaviors compared to the FS dataset. Microbiome distributions and profiles for Barrett's esophagus (BE) and esophageal adenocarcinoma (EA) patients are shown to be very similar (adjusted R squares up to 0.97). Additionally, the scatter plots (profiles) also show that BE and EA profiles (Fig. 6.6(g, h)) are more similar than samples from BE (Fig. 6.6(e, g)). As a consequence, ML models are unable to distinguish these two classes which results in their poor performance.

6.6.5 Neural network visualization

Visualization of the last hidden layer of the test data can further show that neural network can learn meaningful feature representations. By projecting the activation function of the last hidden layer using t-SNE [1] on a two-dimensional space, we can observe there are obvious distinctions among different classes for HMP and IBD datasets (see Fig. 6.7(a, b)); this shows that neural network provides a non-linear transformation of data that can identify different body sites and subjects diagnosed with IBD. However, for the PDX dataset, there is no clear distinction between different classes which results in poor performance for every ML-based classifiers.

6.7 Discussion and conclusion

Advances of high-throughput sequencing techniques enable researchers to gather metagenomic data from different environment and human niches. The available high-throughput experimental data, however, are high-dimensional in nature; this makes it challenging for

researchers to identify and disentangle the underlying microbiome profiles that relate to different human phenotypes such as body sites and disease states.

Although several existing ML models have been proposed for classifying metagenomic data, their performance is mostly unsatisfactory. To boost the classification accuracy, we have proposed a new neural network based pipeline that is suitable for classifying metagenomic datasets. However, the high-dimensional nature and limited number of microbial samples can make such models easily over-fit the training set and thus result in poor classification of new samples. To remedy the data over-fitting problem, we have proposed data augmentation and dropout during training.

Our analysis on real datasets has revealed that ML methods can achieve high classification accuracy when datasets have distinct distributions among different classes. On the contrary, challenging datasets like PDX show similar distributions for different classes; therefore, the existing ML classifiers are unable to distinguish in such situations, while our proposed MetaNN has significant improvements on the classification accuracy. Ultimately, an ideal classifier needs good feature selection mechanisms to select a subset of features that is the most representative for a particular class. In this respect, NNs are well-suited for automatic feature selection and engineering; this makes NNs better than other ML models for classifying metagenomic data.

Experimental results show that the new data augmentation can effectively improve the classification performance for both NN models and ML models. More importantly, when using the augmented training set, the classification results are as good as or better than that of the best non-augmented model; this shows that data augmentation can truly leverage the high dimensionality of metagenomic data and effectively improve the classification accuracy.

Finally, we have shown that our proposed MetaNN outperforms all other existing methods for both synthetic and real data. For the synthetic experiments, we have evaluated several combinations of measurement errors to demonstrate the applicability of MetaNN to different conditions. For real datasets, our MetaNN has average gains of 7% and 5% in terms of F1-macro and F1-micro scores, respectively. Overall, MetaNN has shown very promising results and better performance compared to existing ML methods.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this dissertation, we aimed at studying biological networks and their applications in real world scenarios. Toward this end, we have modeled biological networks at three different granularities:

- First, cellular-level models (*i.e.*, single cell): quorum sensing, chemotaxis, growth, and inhibition models.
- Second, population-level models (*i.e.*, one to three species each with thousands of cells): networks of cell-cell interaction and population control through engineered cells.
- Third, microbiome-level models (*i.e.*, hundreds of species): microbiome association and interaction networks. We have also developed a diagnostic framework based on microbiome profiles.

The main outcome of this dissertation consists of new algorithms and software that can be used directly by both physicians and patients to understand and monitor the dynamics of the microbiome. In summary, our proposed framework is able to:

- Diagnose patients health states based on microbiome profile. Our proposed method can detect the subtle differences of microbiome profiles.
- Identify the “key" bacterial species that interact and affect human health; this way, we can, for example, alter the composition of microbiome by targeted treatments that can add beneficial bacteria (*e.g.*, probiotics).
- Help physicians and patients monitor and visualize the dynamics of microbiome through computer and personalized devices. Based on this, physicians can adjust the treatment to improve the health of the microbiome.
- Provide guidelines to engineer bacteria cells that can effectively and precisely deliver drugs or probiotics to the target locations.

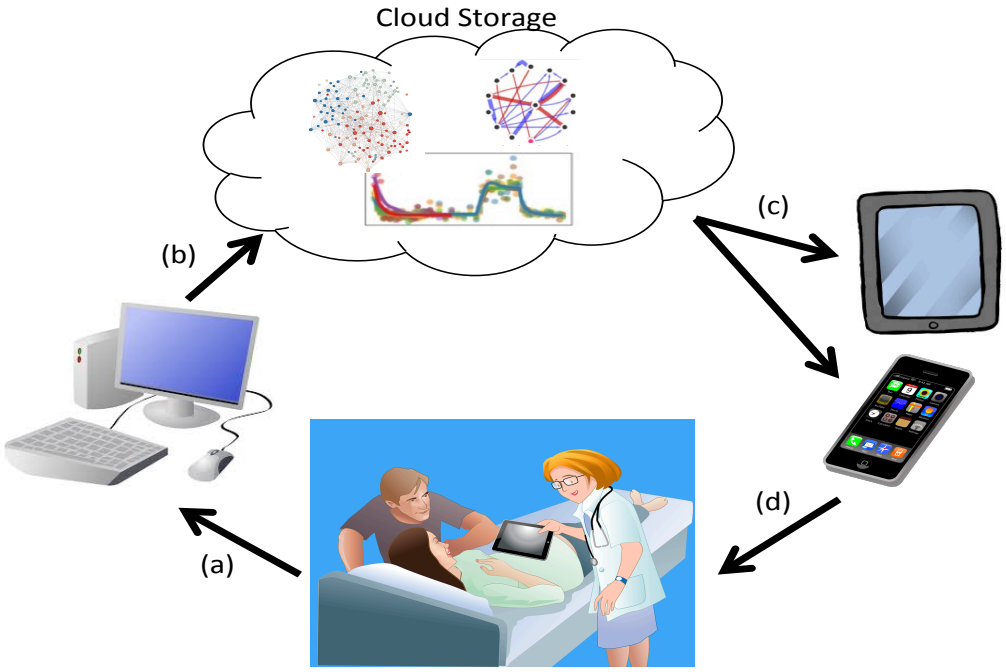

Figure 7.1: Commercial application of our research. We compute and store patients' data in the cloud and then retrieve it through personal devices; this way, physicians and patients can timely track and analyze the microbiome state, analyze changes, etc.

As shown in Fig. 7.1, our proposed framework can be adapted to handle various clinical conditions that depend on the microbiome dynamics. Our software package is able to take in patients sequencing data (a) do "real-time" analysis on the composition of microbiome to infer the interactions among microbes (b) and visualize the dynamics of the microbiome (c). Additionally, the proposed framework can utilize the cloud computing power to directly monitor patients' personal statistics with their mobile devices and then provide relevant data (d).

Taken together, our proposed algorithms and software offer the following unique benefits:

- Generality: Any condition that depends on microbiome state can be analyzed.
- Flexibility: Dynamics of the microbiome can be directly monitored and visualized at patient bed or at home using existing infrastructure and mobile devices.
- Detailed analysis and fast response time: Our algorithms use the existing literature as an input resource, in addition to sample data taken directly from patients, and provide results of microbiome analysis in minutes.
- Easy to use: Using mobile apps with graphical interfaces, both physicians and patients can easily use our software without special training.

7.2 Future Work

Based on our results, we suggest two directions for future work. First, we can utilize a network-based community detection algorithm to cluster microbes into functional groups where microbes in each group are highly correlated. We can construct “gold standard" networks based on the microbe-metabolic relationships which are extracted directly from the metagenomic data; these relationships can be viewed as a bipartite network where node and edge sets (V and E) correspond to microbes and metabolic pathways, respectively. An edge is connected between two nodes if there exists a metabolic pathway in the microbes. By performing clustering on the bipartite network via maximizing the modularity, the resulting community membership for each microbe can be viewed as the *true* label when evaluating different methods.

However, there are several challenges: Disease metagenomic datasets are hard to obtain; therefore, the discovered microbial communities may not be representative enough when applying to microbiome related disease. For example, inflammatory bowel disease (IBD) and type 2 diabetes (T2D) may show completely different microbial community structures. As a consequence, further research is needed on disease related datasets. Second, our inferred microbial association may need further experiments to verify or require other automated pipeline to extract knowledge from existing published literature.

Second, from a sociomicrobiologist’s point of view, it is interesting to consider the evolution of a heterogeneous microbiome consisting of bacteria species competing and coorporating with each other, such as the human gut microbiome. Indeed, many studies have recently shown that changes in the human gut microbiome can be associated with altered human metabolism, immune system, and health. Nowadays, with the rapid advances of sequencing technologies, the individual bacterial species which constitute the microbiome can be elucidated using omics technologies; cues of the metabolic interaction network of the diverse bacteria communities can be also obtained. Based on our newly proposed algorithms that can accurately infer the microbial interactions, we can further establish system-level models of microbe to microbe interactions, as well as microbe-host interactions. Finally, we can analyze this complex hierarchical interaction network using sociomicrobiology theory and complex network theory.

Chapter 8

Appendix

8.1 General form of a dynamic constrained optimization problem

A general dynamic optimization problem can be formulated as follows:

$$\begin{aligned} \min_{p} \quad & J(x_t, p) \\ \text{subject to} \quad & \dot{x}_t = f(x_t, p)\ \forall t \in [t_0, t_{FL}] \\ & x_{t_0}(p) = x_0(p) \\ & p^L \leq p \leq p^U \end{aligned}$$

where $t \in R$ is time, t_0, t_f are the initial and final time, respectively, $t_i \in [t_0, t_{FL}]$, x and $\dot{x} \in R^n$ are the state variables and their time derivatives, respectively, and $p \in R^r$ are the time-invariant parameters and is subjected to the lower constraints p^L and upper constraints p^U. The function J is the objective that we want to minimize. f describes the system dynamics. x_0 is the initial conditions of the state variables.

8.2 Control problem formulation

Consider a general control system which consists of a plant and a controller (see Fig. 2.1(c)). The plant (process) takes in the input variable ($d(t)$) and control variable (CV) ($u(t)$) generating the process variable (PV) ($y(t)$). The controller calculates an error ($e(t)$) signal as the difference between a measured process variable and a desired setpoint (SP) ($r(t)$). The controller aims at minimizing the error by adjusting the process through the control variable ($u(t)$). The control system can be characterized by the following equations:

$$\dot{x}(t) = f(x(t), u(t), d(t)),\ \dot{u}(t) = h(e(t), u(t)) \tag{8.1}$$

$$y(t) = g(x(t)), \qquad e(t) = y(t) - r(t) \tag{8.2}$$

where f, g and h are arbitrary functions. The controller, in this case, can be viewed as an integral controller since the control signal is proportional to the integral of the error signal.

8.3 3D microfluidic environment simulation configuration

We model bacterial growth in a 3D microfluidic environment ($100\mu m$ x $100\mu m$ x $100\mu m$) that is initialized and inoculated with 1000 wild-type cells, all of which are non-overlapping and randomly attached to the substrate. We set up the simulation time up to 150 hrs in order to observe the evolution dynamics of bacteria growth.

Symbol	Value
c_A, c_R	$1e^{-4}/s$
$\alpha_{RA}, \alpha_{RA^2}$	$1e^{-1}$
$\delta_{RA}, \delta_{RA^2}$	$1e^{-1}$
b_A, b_R	$1e^{-2}$
d_A	$1e^{-1}$
V_A, V_R	$2e^{-3}$
K_A	$1e^{-6}$
K_R	$1e^{-5}$

Table 8.1: Table with numerical values of model parameters from [3][4]

Symbol	Value
$c_{pR}, c_{pH}, c_{pA}, c_{pE}, c_{Pyo}$	$1e^{-7}/s$
α_{pR}	$1e^{-1}$
α_I	$1e^{-2}$
δ_{pR}	$1e^{-1}$
$b_{pR}, b_{pH}, b_{pA}, b_{pE}, b_{A_1}, b_{A_2}$	$1e^{-2}$
$b_{A_{1_{EX}}}, b_{A_{2_{EX}}}, b_{Pyo}, b_{Pyo_{EX}}, b_Q$	$1e^{-1}$
$d_{A_1}, d_{A_2}, d_{Pyo}, d_Q$	$1e^{-1}$
$V_{pR}, V_{pH}, V_{pA,1}, V_{pA,2}, V_{pE,1}, V_{pE,2}V_{Pyo}$	$2e^{-3}$
$K_{pA,1}, K_{pA,2}, K_{pE,1}, K_{pE,2}$	$1e^{-6}$
K_{A_1}	$1e^{-3}$
K_{pR}, K_{pH}	$1e^{-1}$
K_{Pyo}	1
β_{pA}	$1e^{-2}$
β_{pH}	$1e^{-1}$

Table 8.2: Table with numerical values of model parameters calibrated in this paper as explained below.

8.4 Synthetic microbial association network

Microbial association network consists of p nodes and e edges where each node represents a microbe and each edge represents a pairwise associations between microbes.

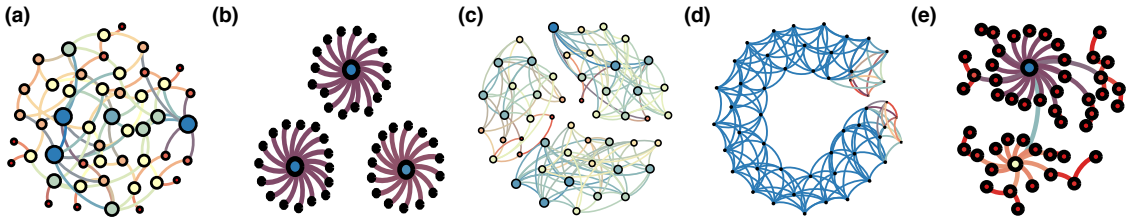

Figure 8.1: Different types of graphs we consider to generate synthetic data: (a) random (b) hub (c) cluster (d) band and (e) scale-free graphs. Node and edges are represented by round circles and curve lines, respectively. See Appendix 8.4 for details.

To evaluate the performance of our model to recover different network structures, we consider five different graph structures (Fig. 8.1) as discussed in [95] with different precision matrices $\mathbf{\Omega}$ defined as follows:

- Random Graph: Each pair of off-diagonal elements are set to non-zero with probability $\frac{3}{p}$ which results in about $e = \frac{3}{2}(p-1)$ edges in the graph.
- Hub Graph: Nodes are randomly partitioned into g groups and within each groups, nodes are connected to the center node which result in $e = p-g$ edges in the graph.
- Cluster Graph: Nodes are randomly partitioned into g groups and within each group, nodes are connected with probability Pr which result in $e = p(\frac{p}{g-1})(\frac{Pr}{2})$ edges in the graph.
- Band Graph: Each adjacent pair of off-diagonal elements (i.e., node i and j) are connected if $1 \leq |i-j| \leq b$ (b is the bandwidth) which result in $e = (2p-1-b)\frac{b}{2}$ edges in the graph. We use $b = 1$ to construct the band graph.
- Scale-free Graph: We generate the graph by using Barabasi-Albert algorithm [141]. The initial graph has two connected nodes and each new node is connected to only one node in the existing graph with the probability proportional to the degree of the each node in the existing graph. It results in $e = p$ edges in the graph.

8.5 Algorithms summaries, simulation settings and run time comparisons

We first summarize the existing methods on inferring correlations or association on microbial data (synthetic data in Table 8.3). More specifically, we consider the following algorithms: CCREPE, CCLasso, SparCC, REBACCA, SPIEC-EASI, and our proposed MPLasso.

For the synthetic and HMP experiments, the settings for each method is as follows: (1) CCLasso: we use the code[1] with the default settings: the number of bootstraps is

[1] https://github.com/huayingfang/CCLasso

Methods	Measures	Assumption	Run-time [secs]
CCREPE	Pearson/Spearman correlation	None	38.284
CCLasso	Pearson correlation of log-ratio transform data	Edge density is no greater than $(\frac{1}{2} - \frac{1}{p-1})$	9.220
SparCC	Pearson correlation of log-ratio transform data	Average correlation of a taxon and others is zero	2.984
REBACCA	Pearson correlation of log-ratio transform data	Each taxon interacts with less than quarter of total taxa	124.840
SPIEC(gl)	Partial correlation of log-ratio transform data	Number of interactions scales linearly with the number of taxa	24.884
SPIEC(mb)	Partial correlation of log-ratio transform data	Number of interactions scales linearly with the number of taxa	20.820
MPLasso	Partial correlation of log-ratio transform data	Number of interactions scales linearly with the number of taxa	**1.098**

Table 8.3: A comparison of correlation based methods.

set to 20. (2) SparCC: we use the implementation from the SPIEC-EASI package with the default settings: the number of iterations for the inner and outer loop is set to 10 and 20, respectively. (3) REBACCA: we use the code[2] with the default settings: the number of bootstraps is set to 40. (4) SPIEC (mb) and SPIEC (gl): we use the code[3] with the default settings: the number of different regularization parameter values is set to 15 and the number of repetitions for StARS is set to 20. (5) CCREPE: we use the code[4] with the default settings: the number of iterations is set to 1000.

The run time is computed by using a synthetic "random" network with 200 samples and 100 OTUs on Intel(R) Core(TM) i5-2400 CPU, 16 GB MEM. We find that MPLasso is much faster than any of the given methods since MPLasso utilize the R package glasso routine which efficiently solve the Lasso problem. SparCC also runs in a much faster fashion since other methods such as CCLasso consists of several optimization procedures from the cross validation.

8.6 Impact of precision levels on prior matrix and synthetic experiments

Given different precision levels, the ratios between maximum (M) and minimum (m) values in prior matrix (P) are adjusted. The minimum and maximum value used in the prior matrix is based on the BIC model selection criteria. More specifically, the maximum value M is set to $m/(1-p)$ where m is the minimum value selected from BIC and p is the precision level.

We examine how different precision level can affect the performance in terms of AUPR on different graph structures. As shown in Fig. 8.2, we notice that the performance of MPLasso increases as the precision level increases; this confirms that accurate microbial prior information can help the graph estimation algorithm (i.e., it becomes more accurate on inferring the graph structures). MPLasso cannot estimate well on band(4) graph due to the special structure. However, it can still achieve up to 0.6 AUPR when precision level is 0.1. For real data experiments, we set up a precision level equal to 0.5 and use BIC to choose m and then calculate M to form the prior matrix.

[2]http://faculty.wcas.northwestern.edu/∽hji403/REBACCA.html

[3]https://github.com/zdk123/SpiecEasi

[4]http://www.bioconductor.org/packages/devel/bioc/html/ccrepe.html

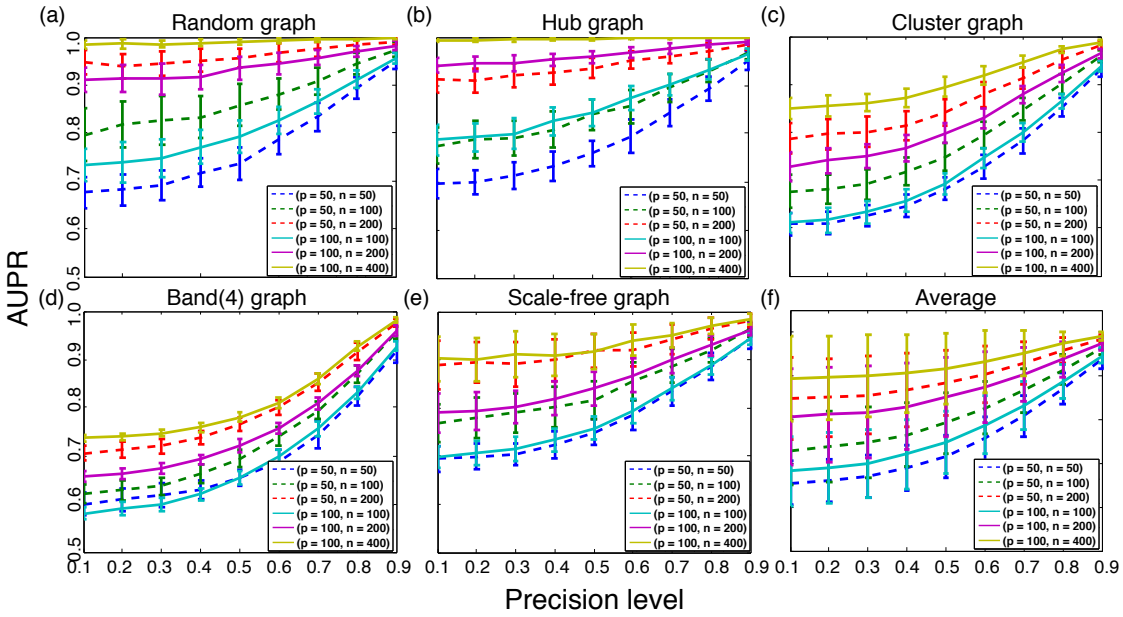

Figure 8.2: Performance of AUPR of different precision levels. Each point is averaged over 100 simulations. We compare 6 different sets of sample size and OTU numbers ((p = 50, n = (50, 100, 200)) and (p = 100, n = (100, 200, 400)).

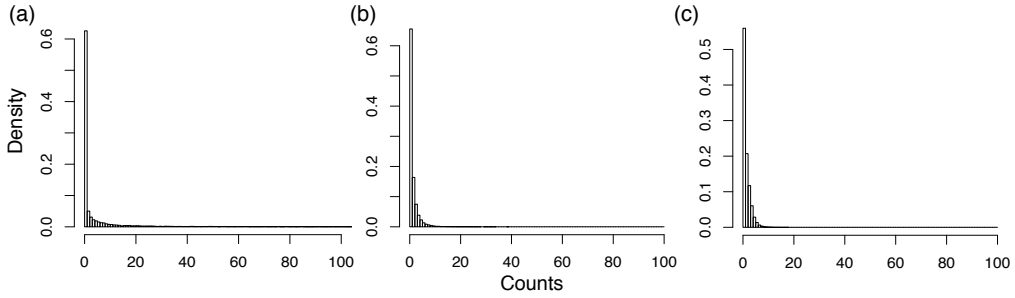

Figure 8.3: The probability density distribution. (a) real data (HMMCP, stool samples), (b) additive log-normal distribution, (c) negative binomial distribution.

8.7 Experiments with synthetic data generated from negative binomial distributions

The metagenomics data is often highly zero-inflated and can have large counts. To show that our log-normal distribution assumption for count data is valid, we look into the density distribution of 16S rRNA experimental data and the simulated data of both log-normal and negative binomial distributions. As shown in FIg. 8.3(a), we first examine the density distribution of the stool samples from the HMMCP dataset; it is obvious that the real data is highly zero-inflated and has a very low density at larger counts (a similar distribution can be found at different body sites).

As shown Fig. 8.3(b), the density for the log-normal distribution is also a zero-inflated distribution, similar to the real data density distribution. We also consider another zero-inflated distribution, namely, the negative binomial distribution with the density distribution shown in Fig. 8.3(c); the results are similar but with a higher density at the locations of smaller counts.

For completeness, we include a new set of experiments to show that our proposed algorithm is able to deal with zero-inflated distributions. More precisely, we choose the negative binomial distribution that is also suitable to model the microbial count data [123].

We consider the same experimental setting of parameters (i.e., different number of taxa, sample sizes, and graph structures) in the main manuscript. As shown in Figs 8.4-8.5 and Tables 8.4-8.5, the performance of our proposed method outperforms all the other methods except a few cases involving hub graphs; this is similar to the results for the additive log-normal model. In summary, our results show that MPLasso works well with many different distributions and graph structures.

Method	L_1	ACC	AUPR	L_1	ACC	AUPR	L_1	ACC	AUPR
Random Graph									
MPLasso	0.071 (0.009)	**0.956 (0.006)**	**0.737 (0.027)**	0.064 (0.009)	**0.971 (0.008)**	**0.855 (0.047)**	0.052 (0.008)	**0.987 (0.006)**	**0.957 (0.024)**
CCLasso	0.077 (0.008)	0.943 (0.007)	0.556 (0.037)	0.071 (0.004)	0.949 (0.007)	0.681 (0.052)	0.058 (0.004)	0.961 (0.009)	0.804 (0.051)
SparCC	0.083 (0.004)	0.943 (0.007)	0.539 (0.032)	0.071 (0.004)	0.948 (0.008)	0.656 (0.050)	0.063 (0.006)	0.959 (0.010)	0.783 (0.051)
REBACCA	**0.070 (0.008)**	0.949 (0.007)	0.658 (0.039)	**0.061 (0.007)**	0.959 (0.008)	0.761 (0.049)	**0.050 (0.008)**	0.969 (0.009)	0.858 (0.047)
SPIEC (mb)	-	0.943 (0.007)	0.627 (0.035)	-	0.950 (0.008)	0.684 (0.034)	-	0.966 (0.013)	0.783 (0.056)
SPIEC (gl)	0.074 (0.009)	0.943 (0.007)	0.654 (0.024)	0.074 (0.008)	0.950 (0.007)	0.700 (0.030)	0.072 (0.011)	0.960 (0.013)	0.768 (0.063)
CCREPE	0.092 (0.005)	0.942 (0.007)	0.561 (0.036)	0.093 (0.007)	0.951 (0.007)	0.703 (0.047)	0.094 (0.008)	0.965 (0.009)	0.839 (0.046)
Hub Graph									
MPLasso	**0.088 (0.001)**	**0.971 (0.003)**	**0.756 (0.027)**	0.086 (0.001)	**0.978 (0.003)**	**0.830 (0.035)**	0.083 (0.001)	0.988 (0.003)	0.940 (0.019)
CCLasso	0.102 (0.010)	0.963 (0.001)	0.537 (0.046)	0.100 (0.003)	0.963 (0.001)	0.603 (0.050)	0.085 (0.003)	0.968 (0.004)	0.753 (0.048)
SparCC	0.112 (0.002)	0.963 (0.001)	0.514 (0.030)	0.101 (0.002)	0.963 (0.001)	0.577 (0.036)	0.057 (0.000)	**0.995 (0.001)**	**0.967 (0.010)**
REBACCA	0.089 (0.002)	0.965 (0.003)	0.629 (0.047)	**0.077 (0.004)**	0.975 (0.006)	0.779 (0.077)	**0.061 (0.004)**	0.991 (0.004)	0.956 (0.029)
SPIEC (mb)	-	0.961 (0.003)	0.650 (0.077)	-	0.962 (0.003)	0.643 (0.067)	-	0.968 (0.004)	0.708 (0.035)
SPIEC (gl)	0.089 (0.000)	0.963 (0.001)	0.682 (0.027)	0.089 (0.000)	0.963 (0.001)	0.683 (0.022)	0.089 (0.000)	0.968 (0.004)	0.731 (0.034)
CCREPE	0.094 (0.006)	0.963 (0.001)	0.518 (0.026)	0.096 (0.006)	0.963 (0.001)	0.613 (0.038)	0.103 (0.005)	0.969 (0.003)	0.758 (0.031)
Cluster Graph									
MPLasso	0.060 (0.005)	**0.911 (0.011)**	**0.686 (0.029)**	0.051 (0.003)	**0.927 (0.010)**	**0.750 (0.029)**	0.043 (0.003)	**0.946 (0.009)**	**0.844 (0.028)**
CCLasso	0.080 (0.009)	0.890 (0.008)	0.476 (0.028)	0.071 (0.004)	0.896 (0.010)	0.560 (0.031)	0.059 (0.003)	0.905 (0.009)	0.642 (0.024)
SparCC	0.088 (0.004)	0.890 (0.009)	0.466 (0.029)	0.074 (0.003)	0.895 (0.010)	0.544 (0.033)	0.066 (0.004)	0.903 (0.008)	0.623 (0.020)
REBACCA	**0.056 (0.004)**	0.897 (0.010)	0.578 (0.029)	**0.042 (0.003)**	0.904 (0.009)	0.627 (0.027)	**0.031 (0.002)**	0.912 (0.009)	0.683 (0.025)
SPIEC (mb)	-	0.891 (0.010)	0.589 (0.034)	-	0.896 (0.010)	0.607 (0.024)	-	0.902 (0.012)	0.640 (0.030)
SPIEC (gl)	0.065 (0.006)	0.891 (0.009)	0.614 (0.021)	0.064 (0.005)	0.896 (0.009)	0.620 (0.019)	0.065 (0.006)	0.902 (0.011)	0.648 (0.024)
CCREPE	0.125 (0.012)	0.889 (0.009)	0.483 (0.028)	0.127 (0.011)	0.898 (0.009)	0.583 (0.026)	0.130 (0.011)	0.911 (0.009)	0.673 (0.023)
Band(4) Graph									
MPLasso	**0.093 (0.002)**	**0.870 (0.008)**	**0.656 (0.021)**	0.087 (0.005)	**0.886 (0.007)**	**0.692 (0.019)**	0.067 (0.005)	**0.909 (0.005)**	**0.762 (0.012)**
CCLasso	0.093 (0.006)	0.850 (0.002)	0.426 (0.037)	0.079 (0.003)	0.855 (0.004)	0.499 (0.020)	0.064 (0.003)	0.866 (0.006)	0.577 (0.019)
SparCC	0.094 (0.003)	0.850 (0.002)	0.416 (0.023)	0.082 (0.002)	0.855 (0.003)	0.485 (0.021)	0.076 (0.002)	0.863 (0.005)	0.557 (0.019)
REBACCA	**0.093 (0.001)**	0.854 (0.003)	0.523 (0.021)	**0.079 (0.001)**	0.864 (0.005)	0.572 (0.025)	**0.063 (0.002)**	0.880 (0.005)	0.643 (0.017)
SPIEC (mb)	-	0.849 (0.002)	0.608 (0.040)	-	0.855 (0.005)	0.612 (0.024)	-	0.858 (0.007)	0.633 (0.021)
SPIEC (gl)	0.096 (0.000)	0.849 (0.002)	0.619 (0.027)	0.096 (0.000)	0.855 (0.005)	0.623 (0.018)	0.096 (0.000)	0.855 (0.007)	0.638 (0.012)
CCREPE	0.167 (0.004)	0.849 (0.001)	0.431 (0.021)	0.171 (0.004)	0.857 (0.004)	0.519 (0.020)	0.177 (0.003)	0.875 (0.006)	0.615 (0.018)
Scale-free Graph									
MPLasso	**0.067 (0.008)**	**0.970 (0.003)**	**0.751 (0.034)**	0.065 (0.009)	**0.976 (0.004)**	**0.818 (0.036)**	0.057 (0.008)	**0.985 (0.004)**	**0.915 (0.034)**
CCLasso	0.074 (0.009)	0.961 (0.001)	0.574 (0.053)	0.070 (0.007)	0.964 (0.002)	0.649 (0.049)	0.060 (0.007)	0.971 (0.004)	0.769 (0.047)
SparCC	0.082 (0.006)	0.961 (0.001)	0.539 (0.039)	0.070 (0.006)	0.964 (0.002)	0.624 (0.044)	0.061 (0.005)	0.969 (0.004)	0.748 (0.052)
REBACCA	0.070 (0.009)	0.966 (0.003)	0.662 (0.046)	**0.063 (0.010)**	0.972 (0.003)	0.749 (0.041)	**0.054 (0.009)**	0.977 (0.005)	0.826 (0.048)
SPIEC (mb)	-	0.960 (0.002)	0.646 (0.055)	-	0.965 (0.003)	0.681 (0.047)	-	0.973 (0.005)	0.761 (0.041)
SPIEC (gl)	0.069 (0.008)	0.962 (0.002)	0.678 (0.023)	0.069 (0.008)	0.966 (0.003)	0.718 (0.031)	0.068 (0.008)	0.972 (0.004)	0.775 (0.037)
CCREPE	0.073 (0.004)	0.961 (0.001)	0.555 (0.040)	0.072 (0.004)	0.964 (0.002)	0.668 (0.047)	0.075 (0.004)	0.973 (0.004)	0.803 (0.045)

Table 8.4: Performance comparison of different methods on negative binomial model. We consider five different graph structures and three sets of parameters, namely, (p = 50, n = 50), (p = 50, n = 100), and (p=50, n=200). For each experiment, we average over 100 simulation runs with standard deviations in round brackets. Bold number shows best result.

Method	L_1	ACC	AUPR	L_1	ACC	AUPR	L_1	ACC	AUPR
Random Graph									
MPLasso	**0.036 (0.004)**	**0.982 (0.003)**	**0.783 (0.033)**	0.033 (0.003)	**0.991 (0.003)**	**0.927 (0.028)**	0.028 (0.003)	**0.997 (0.001)**	**0.990 (0.008)**
CCLasso	0.048 (0.005)	0.973 (0.003)	0.609 (0.041)	0.045 (0.002)	0.979 (0.003)	0.777 (0.045)	0.036 (0.002)	0.986 (0.003)	0.893 (0.033)
SparCC	0.054 (0.002)	0.973 (0.003)	0.580 (0.036)	0.045 (0.002)	0.978 (0.003)	0.753 (0.045)	0.038 (0.002)	0.986 (0.003)	0.881 (0.032)
REBACCA	**0.036 (0.004)**	0.977 (0.003)	0.715 (0.038)	**0.032 (0.003)**	0.984 (0.004)	0.855 (0.045)	**0.027 (0.003)**	0.990 (0.003)	0.932 (0.028)
SPIEC (mb)	-	0.974 (0.003)	0.620 (0.044)	-	0.984 (0.003)	0.766 (0.036)	-	0.990 (0.004)	0.866 (0.046)
SPIEC (gl)	0.038 (0.005)	0.974 (0.003)	0.671 (0.023)	0.039 (0.004)	0.983 (0.003)	0.797 (0.041)	0.037 (0.004)	0.989 (0.004)	0.869 (0.050)
CCREPE	0.048 (0.003)	0.972 (0.003)	0.600 (0.041)	0.048 (0.003)	0.980 (0.003)	0.791 (0.042)	0.047 (0.003)	0.988 (0.003)	0.911 (0.029)
Hub Graph									
MPLasso	0.050 (0.001)	**0.990 (0.001)**	0.846 (0.027)	0.047 (0.001)	**0.995 (0.001)**	0.958 (0.015)	0.044 (0.001)	**0.999 (0.001)**	0.996 (0.003)
CCLasso	0.072 (0.003)	0.982 (0.001)	0.648 (0.038)	0.062 (0.002)	0.985 (0.002)	0.796 (0.041)	0.049 (0.001)	0.990 (0.002)	0.904 (0.030)
SparCC	0.074 (0.001)	0.982 (0.000)	0.621 (0.029)	0.064 (0.001)	0.984 (0.001)	0.762 (0.031)	0.057 (0.000)	0.995 (0.001)	0.967 (0.010)
REBACCA	**0.043 (0.001)**	0.989 (0.002)	**0.847 (0.037)**	**0.031 (0.002)**	**0.995 (0.002)**	**0.966 (0.015)**	**0.017 (0.002)**	**0.999 (0.001)**	**0.997 (0.003)**
SPIEC (mb)	-	0.982 (0.001)	0.620 (0.049)	-	0.987 (0.002)	0.746 (0.039)	-	0.968 (0.004)	0.708 (0.035)
SPIEC (gl)	0.052 (0.000)	0.983 (0.001)	0.701 (0.020)	0.052 (0.000)	0.987 (0.001)	0.779 (0.027)	0.089 (0.000)	0.968 (0.004)	0.731 (0.034)
CCREPE	0.060 (0.003)	0.981 (0.000)	0.620 (0.030)	0.062 (0.002)	0.984 (0.001)	0.780 (0.031)	0.103 (0.005)	0.969 (0.003)	0.758 (0.031)
Cluster Graph									
MPLasso	0.029 (0.002)	**0.959 (0.004)**	**0.694 (0.023)**	0.024 (0.002)	**0.970 (0.003)**	**0.802 (0.024)**	0.020 (0.001)	**0.979 (0.003)**	**0.891 (0.023)**
CCLasso	0.051 (0.004)	0.946 (0.003)	0.489 (0.025)	0.044 (0.003)	0.952 (0.003)	0.608 (0.029)	0.036 (0.001)	0.956 (0.004)	0.682 (0.025)
SparCC	0.056 (0.002)	0.946 (0.003)	0.474 (0.024)	0.046 (0.002)	0.951 (0.003)	0.589 (0.028)	0.038 (0.002)	0.955 (0.004)	0.663 (0.026)
REBACCA	**0.025 (0.002)**	0.950 (0.004)	0.582 (0.025)	**0.019 (0.001)**	0.956 (0.004)	0.674 (0.027)	**0.014 (0.001)**	0.959 (0.005)	0.719 (0.026)
SPIEC (mb)	-	0.948 (0.003)	0.546 (0.034)	-	0.955 (0.004)	0.623 (0.026)	-	0.957 (0.004)	0.649 (0.028)
SPIEC (gl)	0.031 (0.002)	0.948 (0.003)	0.584 (0.016)	0.031 (0.003)	0.954 (0.004)	0.641 (0.029)	0.030 (0.002)	0.955 (0.004)	0.650 (0.027)
CCREPE	0.063 (0.005)	0.945 (0.003)	0.482 (0.024)	0.063 (0.005)	0.953 (0.003)	0.620 (0.028)	0.062 (0.005)	0.957 (0.004)	0.695 (0.026)
Band(4) Graph									
MPLasso	0.048 (0.001)	**0.939 (0.003)**	**0.656 (0.015)**	0.041 (0.001)	**0.950 (0.001)**	**0.723 (0.011)**	0.033 (0.001)	**0.959 (0.001)**	**0.779 (0.009)**
CCLasso	0.059 (0.003)	0.925 (0.001)	0.435 (0.014)	0.052 (0.001)	0.932 (0.002)	0.543 (0.013)	0.041 (0.001)	0.939 (0.002)	0.626 (0.010)
SparCC	0.062 (0.001)	0.925 (0.001)	0.419 (0.015)	0.052 (0.001)	0.931 (0.001)	0.523 (0.012)	0.046 (0.001)	0.938 (0.002)	0.611 (0.010)
REBACCA	**0.044 (0.001)**	0.930 (0.002)	0.539 (0.015)	**0.034 (0.001**)	0.939 (0.001)	0.629 (0.011)	**0.027 (0.001)**	0.945 (0.002)	0.684 (0.009)
SPIEC (mb)	-	0.926 (0.001)	0.556 (0.027)	-	0.930 (0.003)	0.602 (0.014)	-	0.934 (0.003)	0.644 (0.016)
SPIEC (gl)	0.050 (0.000)	0.926 (0.001)	0.574 (0.017)	0.050 (0.000)	0.930 (0.002)	0.611 (0.013)	0.050 (0.000)	0.935 (0.003)	0.649 (0.014)
CCREPE	0.091 (0.002)	0.923 (0.001)	0.428 (0.014)	0.093 (0.001)	0.933 (0.002)	0.554 (0.012)	0.095 (0.001)	0.942 (0.002)	0.643 (0.009)
Scale-free Graph									
MPLasso	**0.032 (0.005)**	**0.986 (0.001)**	**0.761 (0.024)**	0.031 (0.004)	**0.990 (0.002)**	**0.846 (0.038)**	0.028 (0.005)	**0.993 (0.002)**	**0.923 (0.036)**
CCLasso	0.042 (0.005)	0.981 (0.001)	0.565 (0.031)	0.041 (0.003)	0.984 (0.001)	0.696 (0.050)	0.036 (0.005)	0.985 (0.001)	0.771 (0.043))
SparCC	0.052 (0.002)	0.981 (0.000)	0.543 (0.028)	0.042 (0.002)	0.983 (0.001)	0.667 (0.045)	0.036 (0.004)	0.985 (0.001)	0.757 (0.044)
REBACCA	0.034 (0.005)	0.984 (0.001)	0.675 (0.027)	**0.030 (0.005)**	0.986 (0.002)	0.756 (0.051)	**0.025 (0.006)**	0.986 (0.002)	0.786 (0.055)
SPIEC (mb)	-	0.981 (0.002)	0.625 (0.060)	-	0.985 (0.002)	0.707 (0.047)	-	0.987 (0.002)	0.772 (0.035)
SPIEC (gl)	0.033 (0.005)	0.982 (0.001)	0.680 (0.025)	0.034 (0.005)	0.985 (0.002)	0.752 (0.034)	0.034 (0.005)	0.987 (0.002)	0.785 (0.034)
CCREPE	0.038 (0.002)	0.981 (0.000)	0.563 (0.034)	0.037 (0.002)	0.983 (0.001)	0.703 (0.049)	0.037 (0.002)	0.986 (0.002)	0.802 (0.043)

Table 8.5: Performance comparison of different methods on negative binomial model. We consider five different graph structures and three sets of parameters, namely, (p = 100, n = 100), (p = 100, n = 200), and (p=100, n=400). For each experiment, we average over 100 simulation runs with standard deviations in round brackets. Bold number shows best result.

Body Site	(n, p)	MPLasso	SPIEC (gl)	CCLasso
HMASM				
AntNar	(91, 14)	**0.917 (0.023)**	0.679 (0.363)	0.902 (0.022)
Stool	(143, 87)	**0.951 (0.005)**	0.912 (0.019)	0.914 (0.006)
HMMCP				
AntNar	(445, 57)	**0.901 (0.006)**	0.734 (0.035)	0.840 (0.013)
Stool	(437, 135)	**0.956 (0.002)**	0.908 (0.004)	0.827 (0.009)
HMQCP				
AntNar	(269, 116)	**0.937 (0.005)**	0.899 (0.005)	0.921 (0.004)
Stool	(319, 64)	**0.894 (0.005)**	0.689 (0.030)	0.806 (0.016)

Table 8.6: Reproducibility for MPLasso, SPIEC (gl), and CCLasso at different body sites of different types of HMP datasets. For each experiment, we average over 20 simulation runs with standard deviations in round brackets. Bold number shows best result. n and p represent sample size and taxa number, respectively. Abbreviations: AntNar: Anterior nares.

8.8 Experiments with HMP datasets for two more body sites

We report two additional body sites in Table 8.6 and Fig. 8.6. The reproducibility results shows that MPLasso has a better reproducibility over SPIEC (gl) and CCLasso which is the same as in the main manuscript.

For the anterior nares (AntNar), the HMASM dataset (Fig. 8.6(a)) only contains 14 taxa since the trimmed sequences are too short for metaphlan2 to extract effective amounts of taxa. On the other hand, HMMCP and HMQCP (Fig. 8.6(b) and (c)) detect similar genera, for example, *Prevotella*, *Bacteroides*, and *Porphyromonas* which are common in AntNar. The association pairs found in both HMMCP and HMQCP, for example, ⟨*Prevotella*, *Bacteroides*⟩ has been found to have inverse correlation [142].

For the stool HMASM samples (Fig. 8.6(d)), MPLasso suggests an association between the ⟨*Faecalibacterium prausnitzii*, *Escherichia coli*⟩ pair. Although not yet been validated in laboratory settings, researchers have observed the co-abundance of these two species in the human gut [143]. For the genus level data shown in Fig. 8.6(e) and (f), only two genera (*Bacteroides* and *Prevotella*) have been found in common in HMMCP and HMQCP datasets; this may be due to the variations of samples and the number of taxa detected using different pipelines (HMMCP detects 135 taxa while HMQCP obtains 64).

We also consider reproducibility on different percentages of highly connected nodes in Table 8.7. Only when we consider as little as only 25% of high degree nodes, CCLasso has a better performance (but even so for 2% only, on average). While MPLasso outperforms SPIEC (gl) in all the cases reported in Table 8.7.

Body Site	MPLasso	SPIEC (gl)	CCLasso	MPLasso	SPIEC (gl)	CCLasso	MPLasso	SPIEC (gl)	CCLasso
HMASM	top 25%			top 50%			top 75%		
AntNar	0.857 (0.057)	0.633 (0.423)	**0.891 (0.025)**	**0.885 (0.035)**	0.661 (0.379)	0.884 (0.021)	0.897 (0.030)	0.676 (0.363)	**0.901 (0.021)**
BucMuc	**0.929 (0.008)**	0.847 (0.019)	0.703 (0.009)	**0.942 (0.006)**	0.859 (0.020)	0.730 (0.006)	**0.954 (0.004)**	0.880 (0.016)	0.732 (0.005)
Stool	**0.894 (0.011)**	0.820 (0.032)	0.917 (0.008)	**0.919 (0.008)**	0.857 (0.026)	0.913 (0.007)	**0.937 (0.006)**	0.887 (0.022)	0.914 (0.007)
SupPla	0.887 (0.008)	0.794 (0.016)	**0.932 (0.005)**	**0.907 (0.007)**	0.826 (0.012)	0.927 (0.005)	**0.925 (0.006)**	0.852 (0.009)	0.922 (0.005)
TonDor	0.887 (0.010)	0.657 (0.029)	**0.928 (0.015)**	**0.916 (0.008)**	0.692 (0.031)	0.921 (0.014)	**0.935 (0.006)**	0.720 (0.031)	0.918 (0.016)
HMMCP	top 25%			top 50%			top 75%		
AntNar	0.855 (0.012)	0.689 (0.044)	**0.851 (0.017)**	**0.867 (0.010)**	0.706 (0.039)	0.840 (0.015)	**0.880 (0.008)**	0.716 (0.037)	0.835 (0.013)
BucMuc	**0.884 (0.010)**	0.722 (0.037)	0.851 (0.018)	**0.891 (0.008)**	0.725 (0.036)	0.834 (0.015)	**0.905 (0.007)**	0.740 (0.037)	0.824 (0.014)
Stool	**0.911 (0.006)**	0.852 (0.007)	0.858 (0.011)	**0.931 (0.003)**	0.876 (0.006)	0.840 (0.009)	**0.944 (0.002)**	0.892 (0.005)	0.831 (0.010)
SupPla	**0.880 (0.008)**	0.814 (0.014)	0.866 (0.014)	**0.891 (0.005)**	0.829 (0.011)	0.854 (0.013)	**0.905 (0.005)**	0.845 (0.009)	0.845 (0.013)
TonDor	**0.890 (0.007)**	0.786 (0.019)	0.881 (0.011)	**0.901 (0.004)**	0.801 (0.017)	0.872 (0.011)	**0.917 (0.003)**	0.807 (0.016)	0.862 (0.012)
HMQCP	top 25%			top 50%			top 75%		
AntNar	0.873 (0.011)	0.807 (0.009)	**0.909 (0.006)**	**0.905 (0.008)**	0.851 (0.007)	0.910 (0.006)	**0.922 (0.007)**	0.880 (0.006)	0.917 (0.004)
BucMuc	0.813 (0.018)	0.710 (0.033)	**0.860 (0.010)**	**0.825 (0.008)**	0.733 (0.028)	0.844 (0.011)	**0.849 (0.007)**	0.759 (0.026)	0.829 (0.013)
Stool	**0.843 (0.009)**	0.602 (0.049)	0.842 (0.025)	**0.858 (0.007)**	0.622 (0.040)	0.821 (0.022)	**0.874 (0.006)**	0.655 (0.035)	0.804 (0.020)
SupPla	0.844 (0.019)	0.754 (0.023)	**0.915 (0.007)**	0.847 (0.017)	0.765 (0.019)	**0.896 (0.008)**	0.860 (0.013)	0.787 (0.016)	**0.895 (0.007)**
TonDor	0.840 (0.015)	0.716 (0.029)	**0.863 (0.027)**	0.846 (0.012)	0.729 (0.023)	**0.860 (0.022)**	**0.848 (0.011)**	0.726 (0.022)	0.847 (0.024)

Table 8.7: Different percentages of top degree nodes to calculate reproducibility for MPLasso, SPIEC (gl) and CCLasso at different body sites of different types of HMP datasets. For each experiment, we average over 20 simulation runs with standard deviations in round brackets. Bold number shows best result. Abbreviations: AntNar: Anterior nares, BucMuc: Buccal mucosa, SupPla: Supragingival plague, TonDor: Tongue dorsum.

8.9 Methods for calculating Spearman correlation of node degrees

The correlation between node degrees at different body sites is calculated by utilizing the Spearman correlation method. More specifically, we first rank a node (i.e., microbe) based on its node degree. Next, we compute the Spearman correlation based on the rank list. For example, for Stool, we first obtain rank list r_1 and r_2 for HMMCP and HMQCP pipelines, respectively. Next, we compute the Spearman correlation among r_1 and r_2. The purpose of calculating the correlation is to compare and show the differences between the two pipelines and how consistent our proposed algorithms is when inferring the node degrees.

8.10 Prior knowledge introduction in synthetic experiment

The procedure of introducing prior knowledge into the algorithm is as follows: For synthetic data, the prior information is obtained by sampling the true network structure (i.e., the adjacency matrix) and adding noise to simulate realistic conditions.

First, we choose a percentage (based on the prior percentage parameter) of random edges from the true network structure to be used as prior information (i.e., the percentage of "perfect" prior information being used). For example, if the total number of true edges is 100, then a prior percentage = 50% will randomly choose 50 true edges as our prior information.

Second, in order to account for imprecise information (e.g., wrongly annotated associations in the scientific literature) that may appear in the real datasets, we consider the precision of the prior information. If the precision level parameter is set at 50%, then we randomly replace 50% of the correct prior information (25 true edges following our example) with false edges to simulate imprecise information.

8.11 Microbiome time-series generation

Given the association network, we randomly generate the interaction matrix $(\boldsymbol{\beta})$. To ensure the dynamical system that has stable internal equilibriums, we set up the intrinsic growth rate $(\boldsymbol{\alpha})$ to be within [0.1, 0.3]. For the self-interacting parameters (i.e., β_{ii}), we draw from a uniform distribution over the interval [-0.1, -0.3]. The off-diagonal entries β_{ij} are also draw from uniform distribution over the interval [-0.2, 0.2] except 0. Note that the generated interaction matrix is asymmetric and the generated microbiome abundance at any given time should be greater than zero.

8.12 Time-series metagenomic datasets

The time-series dataset is obtained from [2] via preprocessing as relative abundances instead of raw sequencing data; this dataset which consists of approximately half and

one year of daily samples (i.e., daily resolution) for individual MALE and FEMALE, respectively (see Fig. 8.7). The total number of different genera found for both MALE and FEMALE is around 200.

8.13 Performance of machine learning models on real data

We summarize the performance of the ML methods on eight real datasets in Table 8.8. As it can be seen, SVM and RF have better performance compared to other remaining methods in terms of F1-score.

F1-macro						
Dataset	SVM	RF	GB	MNB	LR1	LR2
CBH	0.78(0.03)	0.73(0.03)	0.74(0.04)	0.66(0.03)	0.41(0.04)	0.17(0.01)
CSS	0.63(0.07)	0.58(0.08)	0.48(0.05)	0.49(0.03)	0.26(0.03)	0.24(0.02)
HMP	0.97(0.01)	0.97(0.01)	0.95(0.01)	0.95(0.01)	0.94(0.01)	0.93(0.01)
CS	0.88(0.05)	0.87(0.05)	0.74(0.06)	0.76(0.04)	0.16(0.04)	0.19(0.06)
FS	0.94(0.03)	1.00(0.01)	0.91(0.06)	0.98(0.01)	0.60(0.05)	0.58(0.04)
FSH	0.68(0.04)	0.63(0.08)	0.55(0.06)	0.50(0.04)	0.17(0.01)	0.17(0.00)
IBD	0.68(0.04)	0.57(0.02)	0.65(0.02)	0.43(0.01)	0.47(0.02)	0.43(0.01)
PDX	0.29(0.13)	0.28(0.09)	0.35(0.05)	0.18(0.03)	0.15(0.01)	0.15(0.01)
F1-micro						
Dataset	SVM	RF	GB	MNB	LR1	LR2
CBH	0.93(0.02)	0.91(0.02)	0.89(0.02)	0.88(0.02)	0.76(0.02)	0.68(0.00)
CSS	0.71(0.03)	0.67(0.03)	0.57(0.04)	0.58(0.03)	0.48(0.03)	0.48(0.03)
HMP	0.97(0.01)	0.97(0.01)	0.95(0.01)	0.95(0.01)	0.94(0.01)	0.93(0.01)
CS	0.88(0.06)	0.88(0.04)	0.75(0.05)	0.75(0.05)	0.23(0.05)	0.28(0.07)
FS	0.94(0.03)	1.00(0.01)	0.91(0.06)	0.98(0.01)	0.68(0.03)	0.67(0.03)
FSH	0.70(0.08)	0.69(0.05)	0.58(0.06)	0.62(0.03)	0.33(0.01)	0.33(0.01)
IBD	0.79(0.02)	0.78(0.02)	0.77(0.02)	0.76(0.02)	0.76(0.02)	0.76(0.02)
PDX	0.44(0.07)	0.43(0.07)	0.40(0.05)	0.42(0.04)	0.42(0.04)	0.42(0.04)

Table 8.8: Performance comparison of ML models on eight real datasets described in Table 6.1. We consider several existing supervised ML methods. For each experiment, we consider 10-fold cross-validation and use F1-macro and F1-micro scores to quantify performance as defined in Section 6.5.2. For each fold, we perform five simulation runs with standard deviations shown between round brackets.

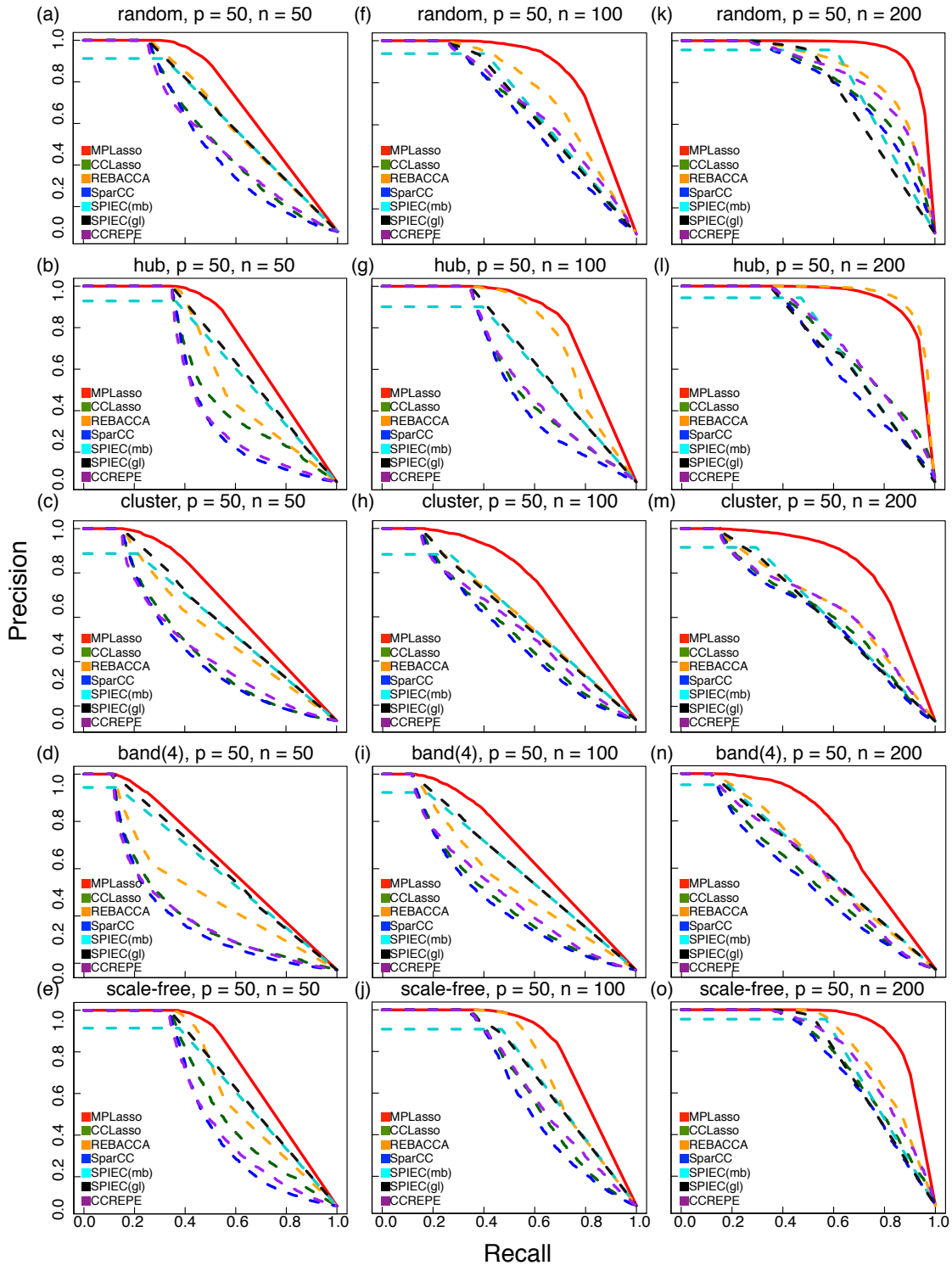

Figure 8.4: AUPR curves of different methods on negative binomial model. Each set of experiment are averaged over 100 simulations. We compare three different sets of sample size and OTU numbers (i.e., (p = 50, n = 50), (p = 50, n = 100), and (p = 50, n = 200)). As can be seen, the MPLasso (red curve) performs better than all other methods.

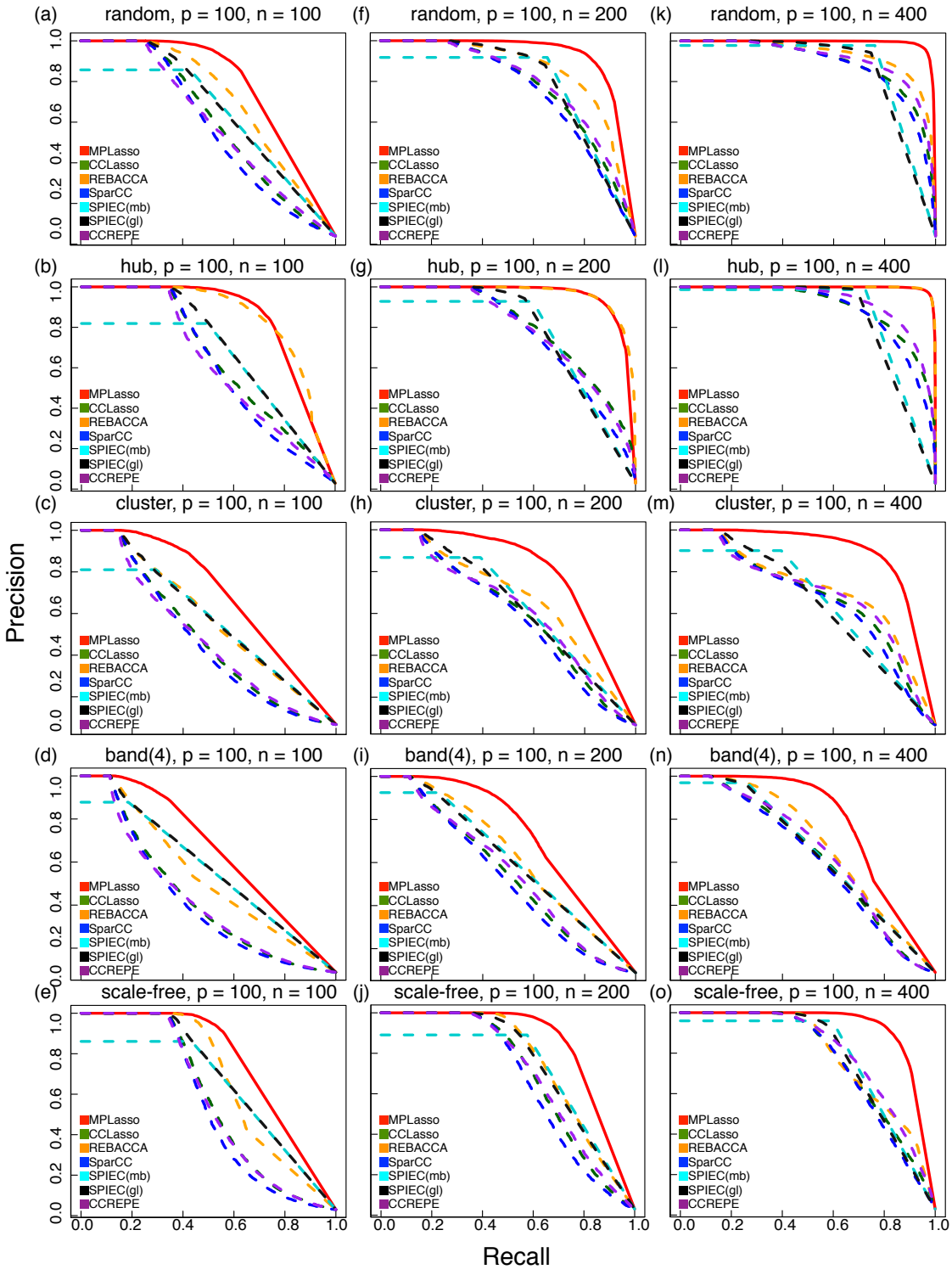

Figure 8.5: AUPR curves of different methods on negative binomial model. Each set of experiment are averaged over 100 simulations. We compare three different sets of sample size and OTU numbers (i.e., (p = 100, n = 100), (p = 100, n = 200), and (p = 100, n = 400)). As can be seen, the MPLasso (red curve) performs better than all other methods.

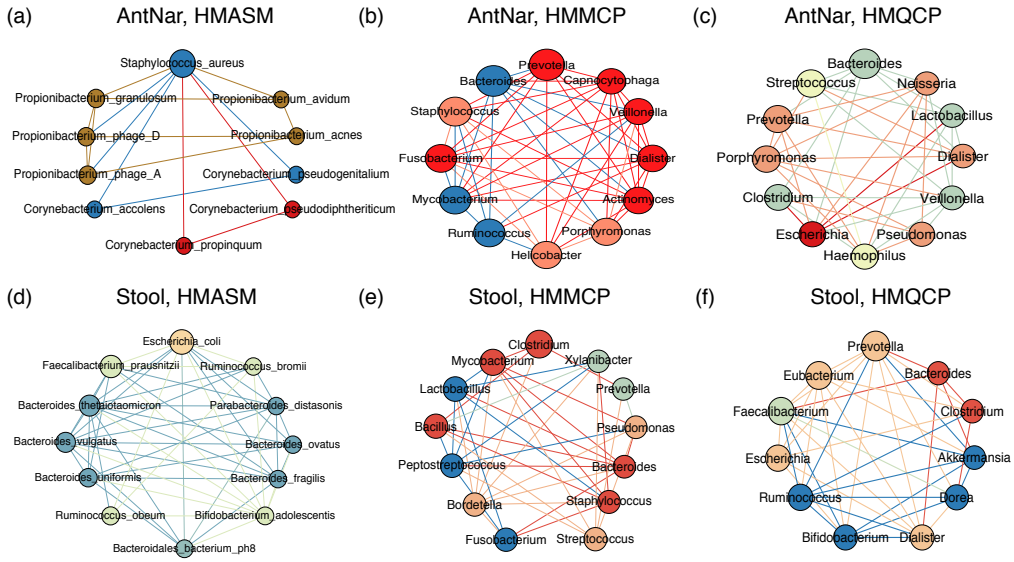

Figure 8.6: Association network visualization of top degree nodes at different human body sites for different data types. The same node colors represent the communities nodes belong to. As can be seen from species level data (HMASM), phylogenetically related OTUs fall in the same community. Node size represents the relative node degree within the association network with counterclockwise layout. Abbreviations: AntNar: Anterior nares.

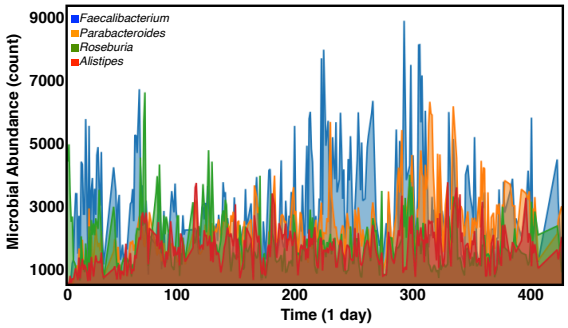

Figure 8.7: Microbial time-series visualization for individual MALE in daily resolution [2]. The abundance of four most abundant genera in human gut (abundance is represented by different colors).

Bibliography

[1] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[2] J.G. Caporaso, C.L. Lauber, and E.K. Costello et al. Moving pictures of the human microbiome. *Genome Biology*, 12(5):R50, 2011.

[3] Pontus Melke, Patrik Sahlin, Andre Levchenko, and Henrik Jönsson. A cell-based model for quorum sensing in heterogeneous bacterial colonies. *PLoS computational biology*, 6(6):e1000819, June 2010.

[4] Magnus G Fagerlind et al. Modeling the effect of acylated homoserine lactone antagonists in Pseudomonas aeruginosa. *Bio Systems*, 80(2):201–13, May 2005.

[5] W. B. Whitman, D. C. Coleman, and W. J. Wiebe. Prokaryotes: the unseen majority. *Proceedings of the National Academy of Sciences of the United States of America*, 95(12):6578–6583, June 1998.

[6] Human Microbiome Project Consortium et al. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–214, 2012.

[7] Luanne Hall-Stoodley, J William Costerton, and Paul Stoodley. Bacterial biofilms: from the natural environment to infectious diseases. *Nature Reviews Microbiology*, 2(2):95–108, 2004.

[8] JW Costerton, Philip S Stewart, and EP Greenberg. Bacterial biofilms: a common cause of persistent infections. *Science*, 284(5418):1318–1322, 1999.

[9] Ron Sender, Shai Fuchs, and Ron Milo. Revised estimates for the number of human and bacteria cells in the body. *PLOS Biology*, 14(8):1–14, 08 2016.

[10] David A Rasko and Vanessa Sperandio. Anti-virulence strategies to combat bacteria-mediated disease. *Nature Reviews Drug Discovery*, 9(2):117–128, 2010.

[11] Luciano Passador, James M Cook, Michael J Gambello, Lynn Rust, and Barbara H Iglewski. Expression of pseudomonas aeruginosa virulence genes requires cell-to-cell communication. *Science*, 260(5111):1127–1130, 1993.

[12] Kendra P Rumbaugh, John A Griswold, and Abdul N Hamood. The role of quorum sensing in the in vivo virulence of pseudomonas aeruginosa. *Microbes and infection*, 2(14):1721–1731, 2000.

[13] Roger S Smith and Barbara H Iglewski. P. aeruginosa quorum-sensing systems and virulence. *Current opinion in microbiology*, 6(1):56–60, 2003.

[14] Sudha Chugani, Byoung Sik Kim, Somsak Phattarasukol, Mitchell J Brittnacher, Sang Ho Choi, Caroline S Harwood, and E Peter Greenberg. Strain-dependent diversity in the pseudomonas aeruginosa quorum-sensing regulon. *Proceedings of the National Academy of Sciences*, 109(41):E2823–E2831, 2012.

[15] Breah LaSarre and Michael J Federle. Exploiting quorum sensing to confuse bacterial pathogens. *Microbiology and molecular biology reviews*, 77(1):73–111, 2013.

[16] Tim Holm Jakobsen, Thomas Bjarnsholt, Peter Østrup Jensen, Michael Givskov, and Niels Høiby. Targeting quorum sensing in pseudomonas aeruginosa biofilms: current and emerging inhibitors. *Future microbiology*, 8(7):901–921, 2013.

[17] Vipin Chandra Kalia and Hemant J Purohit. Quenching the quorum sensing system: potential antibacterial drug targets. *Critical reviews in microbiology*, 37(2):121–140, 2011.

[18] Morten Hentzer, Hong Wu, Jens Bo Andersen, Kathrin Riedel, Thomas B Rasmussen, Niels Bagge, Naresh Kumar, Mark A Schembri, Zhijun Song, Peter Kristoffersen, et al. Attenuation of pseudomonas aeruginosa virulence by quorum sensing inhibitors. *The EMBO Journal*, 22(15):3803–3815, 2003.

[19] Costi D Sifri. Quorum sensing: bacteria talk sense. *Clinical infectious diseases*, 47(8):1070–1076, 2008.

[20] Maryn McKenna. Antibiotic resistance: the last resort. *Nature*, 499(7459):394, 2013.

[21] Anne E Clatworthy, Emily Pierson, and Deborah T Hung. Targeting virulence: a new paradigm for antimicrobial therapy. *Nature chemical biology*, 3(9):541–548, 2007.

[22] C. Lo and R. Marculescu. Towards autonomous control of molecular communication in populations of bacteria. *Proceedings of the Second ACM International Conference on Nanoscale Computing and Communication*, 2015.

[23] C. Lo and R. Marculescu. Autonomous and adaptive control of populations of bacteria through environment regulation. *14th International Conference on Computational Methods in Systems Biology*, 2016.

[24] C. Lo et al. An autonomous and adaptive bacteria-based drug delivery system. *Proceedings of the 3rd ACM International Conference on Nanoscale Computing and Communication*, 2016.

[25] C. Lo et al. Towards cell-based therapeutics: A bio-inspired autonomous drug delivery system. *Nano Communication Networks*, 12:25–33, 2017.

[26] C. Lo and R. Marculescu. Mplasso: Inferring microbial association networks using prior microbial knowledge. *PLoS computational biology*, 13, 2017.

[27] C. Lo and R. Marculescu. Inferring microbial interactions from metagenomic time-series using prior biological knowledge. *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2017.

[28] Michael A Fischbach et al. Cell-based therapeutics: the next pillar of medicine. *Science translational medicine*, 5(179):179ps7, April 2013.

[29] Allen A Cheng et al. Enhanced killing of antibiotic-resistant bacteria enabled by massively parallel combinatorial genetics. *Proceedings of the National Academy of Sciences of the United States of America*, 111(34):12462–7, August 2014.

[30] Ido Yosef, Miriam Manor, Ruth Kiro, and Udi Qimron. Temperate and lytic bacteriophages programmed to sensitize and kill antibiotic-resistant bacteria. *Proceedings of the National Academy of Sciences of the United States of America*, 112(23):7267–72, June 2015.

[31] Bonnie L Bassler and Richard Losick. Bacterially speaking. *Cell*, 125(2):237–46, April 2006.

[32] Susanne Häussler and Tanja Becker. The pseudomonas quinolone signal (PQS) balances life and death in Pseudomonas aeruginosa populations. *PLoS pathogens*, 4(9):e1000166, January 2008.

[33] Ronen Hazan, Jianxin He, Gaoping Xiao, Valérie Dekimpe, Yiorgos Apidianakis, Biliana Lesic, Christos Astrakas, Eric Déziel, François Lépine, and Laurence G Rahme. Homeostatic interplay between bacterial cell-cell signaling and iron in virulence. *PLoS pathogens*, 6(3):e1000810, March 2010.

[34] H Withers, S Swift, and P Williams. Quorum sensing as an integral component of gene regulatory networks in Gram-negative bacteria. *Current opinion in microbiology*, pages 186–193, 2001.

[35] Marvin Whiteley. Identification of genes controlled by quorum sensing in Pseudomonas aeruginosa. *Proc Natl Acad Sci*, 1999(Track II), 1999.

[36] Eun-Jin Kim, Wei Wang, Wolf-Dieter Deckwer, and An-Ping Zeng. Expression of the quorum-sensing regulatory protein LasR is strongly affected by iron and oxygen concentrations in cultures of Pseudomonas aeruginosa irrespective of cell density. *Microbiology (Reading, England)*, 151(Pt 4):1127–38, April 2005.

[37] Deepak Balasubramanian et al. A dynamic and intricate regulatory network determines Pseudomonas aeruginosa virulence. *Nucleic acids research*, 41(1):1–20, January 2013.

[38] Amanda G Oglesby, John M Farrow, Joon-Hee Lee, Andrew P Tomaras, E P Greenberg, Everett C Pesci, and Michael L Vasil. The influence of iron on Pseudomonas aeruginosa physiology: a regulatory link between iron and quorum sensing. *The Journal of biological chemistry*, 283(23):15558–67, June 2008.

[39] Amanda G Oglesby-Sherrouse, Louise Djapgne, Angela T Nguyen, Adriana I Vasil, and Michael L Vasil. The complex interplay of iron, biofilm formation, and mucoidy affecting antimicrobial resistance of Pseudomonas aeruginosa. *Pathogens and disease*, 70(3):307–20, April 2014.

[40] Florian Bredenbruch et al. The Pseudomonas aeruginosa quinolone signal (PQS) has an iron-chelating activity. *Environmental microbiology*, 8(8):1318–29, August 2006.

[41] Ehud Banin et al. Iron and Pseudomonas aeruginosa biofilm formation. *Proceedings of the National Academy of Sciences of the United States of America*, 102(31):11076–81, August 2005.

[42] Nadine S Schaadt, Anke Steinbach, Rolf W Hartmann, and Volkhard Helms. Rule-based regulatory and metabolic model for Quorum sensing in P. aeruginosa. *BMC systems biology*, 7:81, January 2013.

[43] Joshua W Williams, Xiaohui Cui, Andre Levchenko, and Ann M Stevens. Robust and sensitive control of a quorum-sensing circuit by two interlocked feedback loops. *Molecular systems biology*, 4(234):234, January 2008.

[44] James Arpino et al. Tuning the dials of Synthetic Biology. *Microbiology*, 159(Pt 7):1236–53, July 2013.

[45] ML Vasil and UA Ochsner. The response of Pseudomonas aeruginosa to iron: genetics, biochemistry and virulence. *Molecular microbiology*, 34:399–413, 1999.

[46] Liang Yang, Kim B Barken, Mette E Skindersoe, Allan B Christensen, Michael Givskov, and Tim Tolker-Nielsen. Effects of iron on DNA release and biofilm development by Pseudomonas aeruginosa. *Microbiology (Reading, England)*, 153(Pt 5):1318–28, May 2007.

[47] Stephen P Diggle et al. The Pseudomonas aeruginosa 4-quinolone signal molecules HHQ and PQS play multifunctional roles in quorum sensing and iron entrapment. *Chemistry & biology*, 14(1):87–96, January 2007.

[48] SJ Park et al. The Role of AiiA, a Quorum-Quenching Enzyme from Bacillus thuringiensis on the Rhizosphere Competence. *J. Microbiol.*, 2008.

[49] Christopher A Voigt. Genetic parts to program bacteria. *Current opinion in biotechnology*, 17(5):548–57, October 2006.

[50] J. Monod. The growth of bacterial cultures. *Annual Review of Microbiology*, 3:371–394, 1949.

[51] Guopeng Wei et al. Efficient Modeling and Simulation of bacteria-based Nanonetworks with BNSim. *IEEE Journal on Selected Areas in Communications*, 31(12):868–878, 2013.

[52] Seong Hoon Jang, M. Guillaume Wientjes, Dan Lu, and Jessie L.-S. Au. Drug delivery and transport to solid tumors. *Pharmaceutical Research*, 20(9):1337–1350, 2003.

[53] Olivier Trédan, Carlos M Galmarini, Krupa Patel, and Ian F Tannock. Drug resistance and the solid tumor microenvironment. *Journal of the National Cancer Institute*, 99(19):1441–54, October 2007.

[54] IF Tannock, CM Lee, and JK Tunggal. Limited penetration of anticancer drugs through tumor tissue a potential cause of resistance of solid tumors to chemotherapy. *Clinical cancer*, 8(March):878–884, 2002.

[55] NS Forbes. Engineering the perfect (bacterial) cancer therapy. *Nature Reviews Cancer*, 10(11):785–794, 2010.

[56] Won Duk Joo, Irene Visintin, and Gil Mor. Targeted cancer therapy–are the days of systemic chemotherapy numbered? *Maturitas*, 76(4):308–14, December 2013.

[57] Angela A Alexander-Bryant, Wendy S Vanden Berg-Foels, and Xuejun Wen. *Bioengineering strategies for designing targeted cancer therapies.*, volume 118. Elsevier Inc., 1 edition, January 2013.

[58] F. Z. Temel, A. G. Erman, and S. Yesilyurt. Characterization and modeling of biomimetic untethered robots swimming in viscous fluids inside circular channels. *IEEE/ASME Transactions on Mechatronics*, 19(5):1562–1573, Oct 2014.

[59] Guopeng Wei, Paul Bogdan, and Radu Marculescu. Bumpy rides: Modeling the dynamics of chemotactic interacting bacteria. *IEEE Journal on Selected Areas in Communications*, 31(12):879–890, 2013.

[60] Lucien E Weiss, Jonathan P Badalamenti, Lane J Weaver, Anthony R Tascone, Paul S Weiss, Tom L Richard, and Patrick C Cirino. Engineering motility as a phenotypic response to LuxI/R-dependent quorum sensing in Escherichia coli. *Biotechnology and bioengineering*, 100(6):1251–5, August 2008.

[61] Rachel W Kasinskas and Neil S Forbes. Salmonella typhimurium lacking ribose chemoreceptors localize in tumor quiescence and induce apoptosis. *Cancer research*, 67(7):3201–9, April 2007.

[62] Hsuan-Chen Wu, Chen-Yu Tsao, and et al. Autonomous bacterial localization and gene expression based on nearby cell receptor density. *Molecular systems biology*, 9(636):636, January 2013.

[63] Bhushan J Toley and Neil S Forbes. Motility is critical for effective distribution and accumulation of bacteria in tumor tissue. *Integrative biology : quantitative biosciences from nano to macro*, 4(2):165–76, February 2012.

[64] Luis C. Cobo and Ian F. Akyildiz. Bacteria-based communication in nanonetworks. *Nano Communication Networks*, 1(4):244–256, December 2010.

[65] T Nakano et al. Molecular Communication. *Cambridge University Press*, 2013.

[66] Arul Jayaraman and TK Wood. Bacterial quorum sensing: signals, circuits, and implications for biofilms and disease. *Annu. Rev. Biomed. Eng.*, 10:145–167, January 2008.

[67] Mark A.J. Roberts, Antonis Papachristodoulou, and Judith P Armitage. Adaptation and control circuits in bacterial chemotaxis. *Biochemical Society transactions*, 38(5):1265–9, October 2010.

[68] Shana Topp and JP Gallivan. Guiding bacteria with small molecules and RNA. *Journal of the American Chemical Society*, 30322(15):6807–6811, 2007.

[69] Z Xie, L Wroblewska, and L Prochazka. Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science*, 333:1307–1312, 2011.

[70] P Hinow and P Gerlee. A spatial model of tumor-host interaction application of chemotherapy. *Mathematical biosciences and engineering*, 6(3):521–546, 2009.

[71] N. Rady Raz, M. R. Akbarzadeh-T., and M. Tafaghodi. Bioinspired nanonetworks for targeted cancer drug delivery. *IEEE Transactions on NanoBioscience*, 14(8):894–906, Dec 2015.

[72] Chieh Lo, Guopeng Wei, and Radu Marculescu. Towards Autonomous Control of Molecular Communication in Populations of Bacteria. *Proceedings of the 2nd ACM International Conference on Nanoscale Computing and Communication - NANOCOM'15*, pages 1–6, 2015.

[73] Lili Jiang, Qi Ouyang, and Yuhai Tu. Quantitative modeling of Escherichia coli chemotactic motion in environments varying in space and time. *PLoS computational biology*, 6(4):e1000735, April 2010.

[74] Michael W Sneddon, James R Faeder, and Thierry Emonet. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nature methods*, 8(2):177–83, February 2011.

[75] Y.F. Lu and D.B. Goldstein. Personalized medicine and human genetic diversity. *Cold Spring Harbor Perspectives in Medicine*, pages 1–12, 2014.

[76] Juan Jovel et al. Characterization of the Gut Microbiome Using 16S or Shotgun Metagenomics. *Frontiers in microbiology*, 7(April):459, January 2016.

[77] P.J. Turnbaugh, R.E. Ley, and M. Hamady et al. The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature*, 449(7164):804–810, 2007.

[78] JA Gilbert and Folker Meyer. The Earth Microbiome Project: meeting report of the "1 st EMP meeting on sample selection and acquisition" at Argonne National Laboratory October 6 th. *Standards in Genomic Sciences*, pages 249–253, 2010.

[79] D. Sims, I. Sudbery, and N.E. Ilott. Sequencing depth and coverage: key considerations in genomic analyses. *Nature reviews. Genetics*, 15(2):121–32, February 2014.

[80] J. Aitchison. *The Statistical Analysis of Compositional Data*. Chapman & Hall, Ltd., London, UK, UK, 1986.

[81] J. Friedman and E.J. Alm. Inferring correlation networks from genomic survey data. *PLoS computational biology*, 8(9):e1002687, January 2012.

[82] Z.D. Kurtz, C.L. Müller, and E.R. Miraldi et al. Sparse and compositionally robust inference of microbial ecological networks. *PLoS Computational Biology*, pages 1–25, 2015.

[83] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, June 2006.

[84] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, pages 1–14, 2008.

[85] Huaying Fang et al. CCLasso: correlation inference for compositional data through Lasso. *Bioinformatics*, (2009):1–8, 2015.

[86] S. Freilich, A. Kreimer, and I. Meilijson et al. The large-scale organization of the bacterial network of ecological co-occurrence interactions. *Nucleic acids research*, 38(12):3857–68, July 2010.

[87] KMK Lim et al. @ MInter: automated text-mining of microbial interactions. *Bioinformatics*, 32(19):2981–7, October 2016.

[88] Z. Wang, W. Xu, and F.A.S. Lucas et al. Incorporating prior knowledge into Gene Network Study. *Bioinformatics*, 29(20):2633–2640, 2013.

[89] Y. Li and S.A. Jackson. Gene network reconstruction by integration of prior biological knowledge. *G3: Genes| Genomes| Genetics*, 5(6):1075–9, March 2015.

[90] P.D. Schloss, S.L. Westcott, and T. Ryabin. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and environmental microbiology*, 75(23):7537–41, December 2009.

[91] J. Kuczynski and J. Stombaugh. Using QIIME to analyze 16S rRNA gene sequences from Microbial Communities. *Current Protocols in Bioinformatics*, pages 1–28, 2012.

[92] DT Truong et al. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, 12(10):902–3, October 2015.

[93] Y. Hochberg. A sharper bonferroni procedure for multiple tests of significance, 1988.

[94] CC Chang and CJ Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pages 1–39, 2011.

[95] T. Zhao, H. Liu, and K. Roeder. The huge package for high-dimensional undirected graph estimation in R. *Journal of Machine Learning Research*, 13:1059–1062, 2012.

[96] Jens Kreth et al. Bacterial and Host Interactions of Oral Streptococci. *DNA and Cell Biology*, 28(8):397–403, August 2009.

[97] Karoline Faust and JF Sathirapongsasuti. Microbial co-occurrence relationships in the human microbiome. *PLoS Computational Biology*, 8(7):e1002606, January 2012.

[98] Y. Zhang and H. Zhang. Microbiota associated with type 2 diabetes and its related complications. *Food Science and Human Wellness*, 2(3-4):167–172, 2013.

[99] S. Marino, N.T. Baxter, and G.B. Huffnagle et al. Mathematical modeling of primary succession of murine intestinal microbiota. *Proceedings of the National Academy of Sciences of the United States of America*, 111(1):439–44, January 2014.

[100] Chieh Lo and Radu Marculescu. Inferring microbial interactions from metagenomic time-series using prior biological knowledge. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ACM-BCB '17, pages 168–177, New York, NY, USA, 2017. ACM.

[101] I. Cho and M.J. Blaser. The human microbiome: at the interface of health and disease. 13(4):260–270, 2012.

[102] K.J. Pflughoeft and J. Versalovic. Human microbiome in health and disease. *Annual review of pathology*, 7(December 2011):99–122, 2012.

[103] O.C. Aroniadis and L.J. Brandt. Fecal microbiota transplantation: past, present and future. *Current opinion in gastroenterology*, 29(1):79–84, January 2013.

[104] R.R. Stein, V. Bucci, and N.C. Toussaint et al. Ecological Modeling from Time-Series Inference: Insight into Dynamics and Stability of Intestinal Microbiota. *PLoS Computational Biology*, 9(12):31–36, 2013.

[105] C.K. Fisher and P. Mehta. Identifying keystone species in the human gut microbiome from metagenomic timeseries using sparse linear regression. *PLoS ONE*, 9(7):1–10, 2014.

[106] V. Bucci, B. Tzen, and N. Li et al. MDSINE: Microbial Dynamical Systems INference Engine for microbiome time-series analyses. *Genome Biology*, 17(1):121, 2016.

[107] A. Bashan, T.E. Gibson, and J. Friedman et al. Universality of human microbial dynamics. *Nature*, 534(7606):259–62, 2016.

[108] A. Greenfield, C. Hafemeister, and R. Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067, 2013.

[109] L. Breiman. Bagging Predictors. *Machine Learning*, 24(421):123–140, 1996.

[110] J. Hofbauer, V. Hutson, and W. Jansen. Coexistence for systems governed by difference equations of Lotka-Volterra type. *Journal of Mathematical Biology*, 25(5):553–570, 1987.

[111] M. Arumugam, J. Raes, and E. Pelletier et al. Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–180, 2013.

[112] W. Jiang, N. Wu, and X. Wang et al. Dysbiosis gut microbiota associated with inflammation and impaired mucosal immune function in intestine of humans with non-alcoholic fatty liver disease. *Scientific reports*, 5:8096, 2015.

[113] S.H. Duncan, A. Belenguer, and G. Holtrop et al. Reduced dietary intake of carbohydrates by obese subjects results in decreased concentrations of butyrate and butyrate-producing bacteria in feces. *Applied and Environmental Microbiology*, 73(4):1073–1078, 2007.

[114] Jonas Halfvarson et al. Dynamics of the human gut microbiome in inflammatory bowel disease. *Nature Microbiology*, 2, 02 2017.

[115] Yong Zhang and Heping Zhang. Microbiota associated with type 2 diabetes and its related complications. *Food Science and Human Wellness*, 2(3):167 – 172, 2013.

[116] Marti J. Anderson and Trevor J. Willis. Canonical analysis of principal coordinates: A useful method of constrained ordination for ecology. *Ecology*, 84(2):511–525, 2003.

[117] Alexander Statnikov et al. A comprehensive evaluation of multicategory classification methods for microbiomic data. *Microbiome*, 1(1):11, Apr 2013.

[118] Dan Knights et al. Supervised classification of human microbiota. *FEMS Microbiology Reviews*, 35(2):343–359, 2011.

[119] The Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. *Nature*, 486:207–214, 2012.

[120] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[121] Alex Krizhevsky et al. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[122] Nitish Srivastava et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[123] Paul J. McMurdie and Susan Holmes. Waste not, want not: Why rarefying microbiome data is inadmissible. *PLOS Computational Biology*, 10(4):1–12, 04 2014.

[124] Dirk Gevers et al. The treatment-naive microbiome in new-onset crohn's disease. *Cell Host & Microbe*, 15(3):382–392, 2011.

[125] Liying Yang et al. *Foregut Microbiome, Development of Esophageal Adenocarcinoma, Project*, pages 1–5. Springer New York, New York, NY, 2013.

[126] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[127] Rong-En Fan et al. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, June 2008.

[128] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[129] Christopher D. Manning et al. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[130] Terrence S. Furey et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[131] Trevor Hastie et al. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[132] Elizabeth K. Costello et al. Bacterial community variation in human body habitats across space and time. *Science*, 326(5960):1694–7, 2009.

[133] Noah Fierer et al. Forensic identification using skin bacterial communities. *Proceedings of the National Academy of Sciences*, 107(14):6477–6481, 2010.

[134] Jun Lu et al. Identifying differential expression in multiple sage libraries: an overdispersed log-linear model approach. *BMC Bioinformatics*, 6(1):165, Jun 2005.

[135] Mark D. Robinson and Gordon K. Smyth. Small-sample estimation of negative binomial dispersion, with applications to sage data. *Biostatistics*, 9(2):321–332, 2008.

[136] Mingyuan Zhou et al. Beta-negative binomial process and poisson factor analysis. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, 22:1462–1471, 21–23 Apr 2012.

[137] G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234, September 1989.

[138] Ian Goodfellow et al. *Deep Learning*. MIT Press, 2016.

[139] Henry S. Baird. *Structured Document Image Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.

[140] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[141] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(January), 2002.

[142] Anastassia Gorvitovskaia et al. Interpreting Prevotella and Bacteroides as biomarkers of diet and lifestyle. *Microbiome*, 4:15, April 2016.

[143] Mireia Lopez-Siles et al. Mucosa-associated Faecalibacterium prausnitzii and Escherichia coli co-abundance can distinguish Irritable Bowel Syndrome and Inflammatory Bowel Disease phenotypes. *International journal of medical microbiology*, 304(3-4):464–75, May 2014.