
# Exploring Molecular Communication and Gene Regulation-Based Biocomputing of Bacteria

**Samitha Sulakshana Somathilaka**

**Supervisors:** Dr. Sasitharan Balasubramaniam and Dr. Daniel Perez Martins

School of Science and Computing
South East Technological University
July, 2024

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

# Declaration

I hereby declare that this material, which I now submit for assessment on the program of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Samitha S. Somathilaka

Submitted to South East Technological University,

July 2024

# Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Sasitharan Balasubramaniam and Dr. Daniel P. Martins, for their unwavering guidance, support, encouragement, and patience throughout my studies. I will always be grateful for their help and motivation, which made my PhD journey both productive and stimulating. Their example as outstanding researchers has been truly inspiring.

I extend my heartfelt thanks to the entire team and staff at the Walton Institute, the broader community at South East Technological University and the University of Nebraska-Lincoln for their assistance during my time as a student. I am also highly appreciative of the support provided in part by Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland under Grant 16/RC/3835.

I am profoundly grateful to my mother for her endless love, support, and encouragement. Morover, to everyone around me who has constantly supported me morally and emotionally, your encouragement and faith have given me the strength to pursue my dreams. A special thanks must go to Dr. Kaushalya Dissanayake and Dr. Dixon Vimalajeewa for their unwavering support. Finally, I am grateful to all my friends from Sri Lanka and Ireland for their constant motivation, help, and the wonderful memories we have shared over the years.

"*We live in illusion and the appearance of things. There is a reality. We are that reality. When you understand this, you see that you are nothing, and being nothing, you are everything. That is all.*"

- Kalu Rinpoche

"*I would rather have questions that can't be answered than answers that can't be questioned.*"

- Richard Feynman

# Abstract

Artificial Intelligence (AI) has become a cornerstone of modern technological advancements, deeply intertwined with neuroscience and transforming into an essential part of daily life. AI has reshaped various industries, enhancing problem-solving capabilities and impacting societal norms. Originally inspired by the brain's functions, such as neurons and synapses, AI has continually integrated neuroscience findings to improve systems' sophistication and efficiency. This includes understanding brain plasticity and neuronal communication. Moreover, as AI has progressed, the focus has expanded from conventional neural networks to exploring neuromorphic architectures, including both silicon-based and biological systems, to enhance hardware-based AI solutions.

Although the integration of AI with silicon-based computing has significantly transformed society by enhancing efficiency and automating tasks across various sectors with minimal human input, this combination faces challenges such as high energy demands, complexity, adaptability, and biocompatibility. Therefore, this thesis explores the potential of bacteria as a living biocomputing platform. It begins with a macroscopic examination of bacterial communities' collective behaviors and computational dynamics at the biome and population levels, which provides insights into their information processing, decision-making, and communication strategies. The focus then shifts to the single-cell level, specifically on the gene regulatory network (GRN) that drives bacterial computation. This investigation into the GRN reveals the cellular logic behind bacterial computing and paves the way for evaluating the

reliability, energy efficiency, and practicality of bacterial systems for computational tasks like regression and classification. Highlighting the ultra-low energy dynamics of bacterial metabolism offers a solution to the energy limitations of silicon-based systems. Furthermore, the scalability, adaptability, and biocompatibility of bacterial populations address challenges in generalizing biological systems. The thesis aims to integrate these biological computing properties into conventional computing challenges, envisioning a transformative approach to AI and neuromorphic engineering through bacteria-based wet-neuromorphic systems, which could blend biological and computational intelligence.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The inception of Artificial Intelligence (AI) was significantly inspired by the intrinsic mechanisms of the brain, including neurons, synapses, and neural circuits [1]. This inspiration stems from a desire to emulate the brain's remarkable capabilities in problem-solving, learning, and memory. As AI has evolved, the field has continued to draw from neuroscience, incorporating insights on brain plasticity, the complex structure of dendrites, and the ways neurons communicate through electrical and chemical signals. The interplay between AI and neuroscience not only enhances our understanding of human cognition but also propels AI towards more sophisticated and versatile systems, capable of tackling tasks with better efficiency.

The significance of AI in society with its deep-rooted connection to neuroscience has formed one of the foundational pillars of the modern era's technological advancements. This has led the transition of AI from a fascinating concept to a crucial component of our daily lives, reshaping industries, enhancing problem-solving capabilities, and significantly impacting societal norms [2]. This, in turn, led to groundbreaking applications that are vital for numerous sectors including, manufacturing, automobile, finance and healthcare [3, 4].

With the expansion of conventional Neural Networks (NNs) as the most crucial subset of AI, researchers have started exploring the potential of hardware-based sys-

tems, specifically neuromorphic architectures. However, this thesis extends beyond merely silicon-based solutions, encompassing biological entities as well, including brain organoids and in particular cells like bacteria.

## 1.1 Background and Motivation

This section lays the groundwork and rationale for the research conducted in this thesis. It begins by delving into the evolution of AI technologies in Section 1.1.1. Next, Section 1.1.2, highlights the importance of innovative approaches that integrate contemporary AI with attributes such as energy efficiency, biocompatibility, and generalizability setting the stage for the thesis's central premise. Finally, Section 1.1.3 focuses on the paradigm shift towards neuromorphic computing.

### 1.1.1 Evolution of Neural Approaches to AI

The genesis of NNs and AI can be traced back to the mid-20th century, driven by the ambition to emulate the information-processing capabilities of biological systems. The seminal work during this period started with the McCullock-Pitts neuron model in 1943 [5], introducing a basic computational view of neurons, followed by the establishment of Hebbian learning as the first rule for updating NNs in 1949 [6], and notably, the introduction of the perceptron by Frank Rosenblatt in 1958 [7, 8]. These milestones were pivotal in laying the groundwork for the evolution of AI.

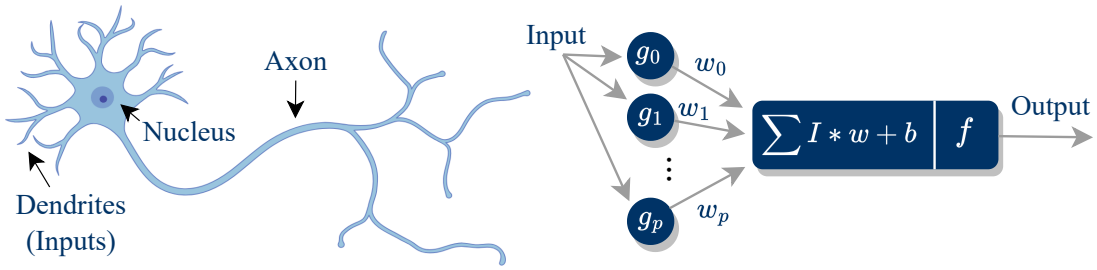



Figure 1.1: Illustration of the fundamental idea of the perceptron.

The artificial perceptron operates on the principle of simulating the basic func-

tionality of a biological neuron, by aggregating input signals, applying weights to signify the strength of each input, and then passing the sum through an activation function to generate a binary output, analogous to the neuron's firing mechanism as shown in Fig. 1.1.

As the discipline advanced, NN models encompass diverse network structures, activation functions, learning strategies, and information propagation techniques. Additionally, this growth began to emphasize on aspects such as energy efficiency and biocompatibility and that led to drastic improvements of both software and hardware, showcasing the evolutionary journey of AI from conceptual foundations to the sophisticated architectures of today.

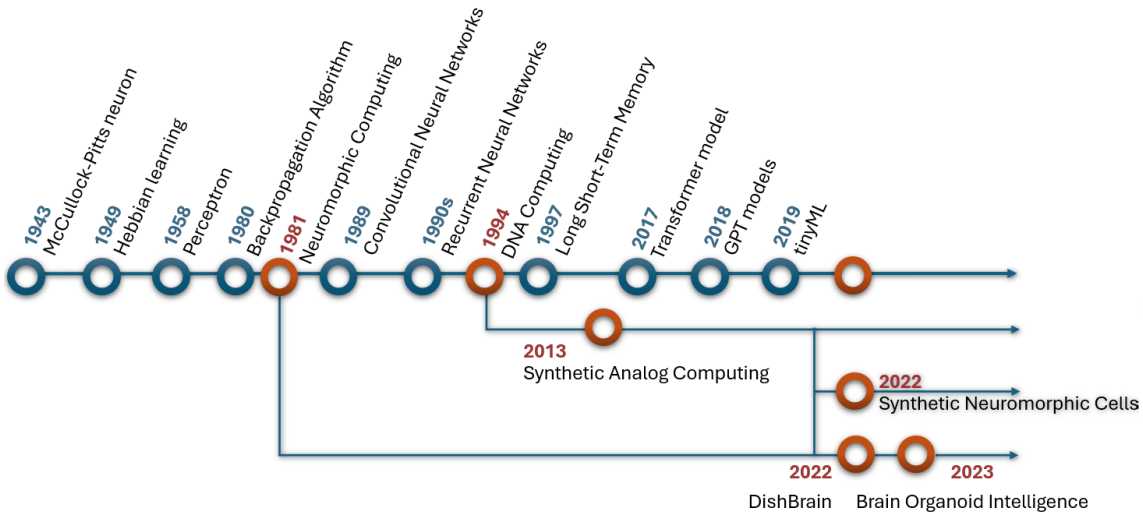



Figure 1.2: Illustration of the evolution of AI technologies.

A notable advancement in the field is the increasing complexity of NN architectures. In contrast to simpler models, a deeper NN architecture comprising multiple layers of neurons stacked together was designed and introduced [9], where each layer progressively reduces the feature set while transforming the data into a more abstract representation, facilitating the extraction of complex patterns and features essential for various deep-learning applications. In comparison to Frank Rosenblatt's simple single-layer perceptron model, the architectural complexity of NNs has dramatically evolved to this day, culminating in models such as GPT-4, which boasts up to 1.7

trillion parameters. This evolution is depicted in Fig.1.2.

The following list explains each step of the timeline briefly.

- **1943**: The introduction of **McCullock-Pitts** model of neuron [5].
- **1949**: **Hebbian learning** as the first rule for updating NNs [6].
- **1958**: The idea of the **perceptron** was established [7, 8].
- **1980s**: Advent of the **Backpropagation algorithm**, enabling the training of multi-layer perceptrons and deepening NN architectures [10].
- **1981**: Neuromorphic computing began with Caltech's creation of analog silicon retina and cochlea devices, inspired by neural paradigms.
- **1989**: Birth of **Convolutional Neural Networks (CNNs)** with the LeNet architecture, designed for image recognition tasks [11].
- **Late 1990s**: Introduction of **Recurrent Neural Networks (RNNs)**, enhancing the processing of sequential data [12].
- **Late 1994**: DNA computing was first proposed by Leonard Adleman, through his Science article "Molecular Computations of Solutions to Combinatorial Problems" published in November 1994 [13].
- **1997**: Emergence of **Long Short-Term Memory (LSTM) networks**, a variant of RNNs, tackling the challenge of learning long-term dependencies [14].
- **2013**: Introduced engineered synthetic analog gene circuits can perform complex computational functions within living cells [15].
- **2017**: Introduction of the **Transformer model**, revolutionizing natural language processing with attention mechanisms [16].

- **2018**: Introduction of **GPT-1** and subsequent iterations, marking significant advancements in NN capabilities and applications [17].

- **2019**: The concept of tinyML, introduced in 2019, revolutionized the field of machine learning by enabling the deployment of advanced models on highly resource-constrained devices [18].

- **2022**: **DishBrain** employs human and mouse brain cells, utilizing a micro-electrode array as the interface, to learn playing the game Pong [19].

- **2022**: Development of synthetic neuromorphic system using *Escherichia coli* cell [20].

- **2023**: An AI hardware development leveraging the adaptive reservoir computation capabilities of biological neural networks within a brain organoid [21].

Each of these milestones not only showcases the evolving complexity and capabilities of NNs but also reflects the expanding diversity of applications they enable.

However, the determination of the number of layers, their types (e.g., convolutional, recurrent), connections between nodes and other parameters is a crucial task that requires extensive expertise. The Network Architecture Search (NAS) is a cutting-edge field in machine learning focuses on automating the discovery of architectures that maximize performance omitting the traditional trial-and-error approach. The development of NAS techniques demonstrated that reinforcement learning could identify high-performing architectures [22, 23], while Real et al. [24] showed that evolutionary strategies also offer similar potential. By leveraging these techniques, NAS has the potential to uncover innovative NN designs that outperform manually engineered models, significantly advancing the field of AI which further influence the methodology of this thesis.

However, three significant deviations from this base timeline can be observed with the introduction of neuromorphic computing, DNA computing, and organoid

computing. Neuromorphic computing refers to a hardware-based platform that mimics the neural networks of the human brain. DNA computing views DNA as a fundamental unit of computation, leveraging its biological processes for computing tasks. Organoid computing involves the use of biological organoids, such as brain organoids and proteonoids, for computational purposes. The convergence of neuromorphic and DNA computing represents a transformative step in synthetic biology, leading to the exploitation of cells that operate on neuromorphic principles. Neuromorphic computing, emphasizes efficient, parallel processing with low power consumption, while DNA computing utilizes the molecular properties of DNA for data storage and complex computations. This integration enables the development of living cells that can perform intricate computations and adapt in real-time, with applications ranging from medical diagnostics to environmental monitoring. Synthetic biology further leverages living cells to perform human-defined computations, often using the "genetic circuit" metaphor similar to silicon-based computers [25]. The immense potential of these advancements to revolutionize biological and computational problem-solving is one motivation for this thesis.

### 1.1.2 Drive Towards Energy-Efficient and Sustainable AI

The drive towards energy-efficient and sustainable AI is motivated by the need to minimize the substantial energy demands of current AI systems' training and operational phases. For example, training a single expansive language model, such as ChatGPT-3, can utilize as much as 10 gigawatt-hours (GWh) of power, while processing hundreds of millions of daily queries may incur operational energy costs of approximately 1 GWh each day [26]. In contrast, the human brain consumes only about 20W to perform computations at the rate of 1 exaFLOPS (estimated), in stark contrast to a supercomputer, which requires approximately 21 MW to achieve the same computational performance [27]. The following compares the energy use or production of various computational entities, ranging from biological brains to

complex simulations, providing a comprehensive overview of the differences in energy requirements among natural systems and computational models of varying scales.

| **Description** | **Energy Consumption (Watt)** |
|---|---|
| Mouse brain | $1 \times 10^{-3}$ |
| Human brain | $2 \times 10^{1}$ |
| Laptop | $1 \times 10^{2}$ |
| Mouse cortex section simulation | $1 \times 10^{5}$ |
| Mouse brain simulation, scaled | $1 \times 10^{7}$ |
| Human brain simulation, scaled | $1 \times 10^{9}$ |
| Chat GPT model training | $1 \times 10^{9}$ |
| Mouse brain simulation, scaled & time corrected | $1 \times 10^{11}$ |
| Human brain simulation, scaled & time corrected | $1 \times 10^{15}$ |
| 8 million human brains, scaled & time corrected | $1 \times 10^{20}$ |

Table 1.1: Energy consumption comparison [26, 27, 28].

TinyML is an advancement towards energy-efficient AI, focusing on deploying machine learning algorithms on low-power microcontrollers. This approach enables several key capabilities, such as supporting energy-harvesting edge devices to run learning models efficiently, using battery-operated embedded edge devices, and offering scalability to accommodate numerous sensors in cost-effective embedded devices. TinyML models are also compact enough to be stored within a few kilobytes of on-device RAM.

TinyML works by optimizing machine learning algorithms to operate within the hardware constraints of microcontrollers, which typically have limited processing power, memory, and energy availability. Techniques such as model quantization, pruning, and efficient data handling help adapt models to run effectively in such environments. The use of specialized frameworks like TensorFlow Lite for Microcontrollers allows developers to compress models without significant loss in performance, enabling the execution of tasks like sensor data analysis, speech recognition, and anomaly detection on low-power devices[29, 30].

Further, in this pursuit of energy-efficient and sustainable AI, neuromorphic systems emerge as a promising solution, drawing inspiration from the efficiency of the human brain [31, 32]. These systems process through specialized hardware that sim-

ulates neurons and synapses [33], thereby significantly reducing power consumption compared to traditional computing architectures.

Techniques beyond tinyML and neuromorphic computing include model pruning, quantization, knowledge distillation, and low-rank factorization. Model pruning removes redundant weights or neurons, reducing model size and computational load while maintaining accuracy. Quantization reduces parameter precision, typically converting 32-bit to 8-bit, which decreases memory use and speeds up computation with minimal performance loss. Knowledge distillation trains a smaller model (student) to mimic a larger model (teacher), retaining much of the performance while reducing complexity. Low-rank factorization approximates weight matrices using lower-rank versions, reducing parameters and computational requirements, particularly in CNNs.

However, challenges including maintaining accuracy during extreme compression, improving training efficiency, and minimizing model size and energy consumption remain. Addressing these is crucial for scalable, efficient AI deployment.

### 1.1.3 Paradigm Shift Towards Neuromorphic Computing

As AI continues to evolve, a paradigm shift is necessitated by the current challenges in AI, such as the high energy consumption of training large NNs and the need for more intuitive, context-aware processing. Neuromorphic computing, with its potential for low-power operation and real-time learning and adaptation, could lead to the development of AI that is not only more powerful but also more integrated with the natural environment. Key characteristics of neuromorphic computing can be elaborated as follows,

- **High parallelizability**: Neuromorphic computers operate on an inherently parallel architecture, allowing neurons and synapses to function simultaneously, yet the computations they perform are relatively simple compared to those in parallelized von Neumann systems.

- **In memory computing**: Neuromorphic hardware eliminates the traditional separation between processing and memory, blending these functions within neurons and synapses that both process information and store values. This integration mitigates the von Neumann bottleneck, which slows down throughput due to processor-memory separation, and reduces energy consumption by avoiding frequent data accesses from main memory typical in conventional computing systems.

- **Event-driven computating**: Neuromorphic computers utilize event-driven computation, where neurons and synapses are activated only in the presence of data, specifically spikes, and rely on temporally sparse activity. This approach enables highly efficient computation as work is performed only when necessary, and spikes, the primary data events in these networks, occur infrequently.

Examples of this transformative approach include IBM's TrueNorth and Intel's Loihi, which represent significant steps toward realizing brain-like processing capabilities, demonstrating the ability to perform complex computations more efficiently.

Conversely, researchers have developed biocomputing approaches including brain organoids and proteinoids as alternatives to silicon-based neuromorphic systems,

- **Brain organoids**

  Brain organoids present a pioneering approach in the development of neuromorphic systems using biological neural networks, simulating the brain's structural and functional complexities on a microscale. These 3D cellular models emulate specific aspects of the human brain's architecture, offering a dynamic platform for studying NNs and computational strategies inherent to brain function [34, 35, 36]. The application of brain organoids in neuromorphic computing involves leveraging their biological fidelity to understand and replicate neural processing mechanisms, potentially leading to advancements

in AI that are closer to human cognitive capabilities.

- **Proteinoids** Proteinoids, synthesized polymers resembling natural proteins, represent an innovative frontier in neuromorphic systems, venturing beyond traditional silicon-based approaches. These polymers offer a unique platform for simulating neural functions, leveraging the inherent properties of proteins for information processing and storage. The advantage of proteinoids lies in their biomimetic properties, allowing for the creation of systems that more closely replicate the biological processes of neural tissue, potentially leading to advances in computing power and efficiency [37, 38].

However, integrating proteinoids and brain organoids into functional computing systems introduces complexities, including the challenge of interfacing biological and electronic components effectively. Moreover, the stability and scalability of these systems under varying environmental conditions remain concerns.

Focusing on these challenges, the next section explores bacterial cells as promising candidates for biocomputing.

### 1.1.4 Bacteria for Biocomputing

Bacterial cells exhibit astonishing sensitivity to their environment, processing information through gene expression, engaging in complex communication with neighboring cells, and interacting with other extracellular entities. This intricate network of sensing, information processing, and actuation bears a resemblance to the workings of a neuromorphic computing system.

#### Sensing

Bacteria possess remarkable abilities to detect external stimuli and initiate a diverse array of responses [39, 40]. They interpret external signals, which encompass molecules communicated by other microbes, as well as fluctuations in environmental

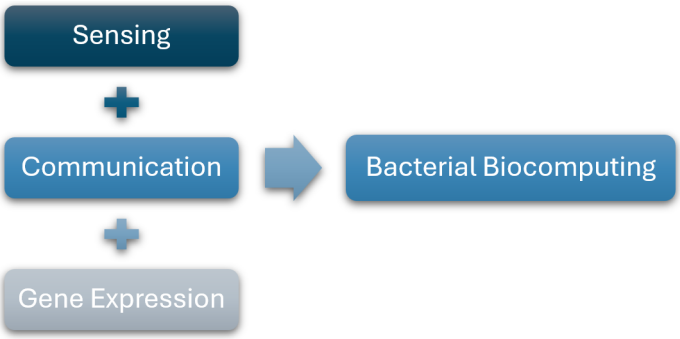



Figure 1.3: Abstraction of the bacterial biocomputing components.

conditions such as temperature or pH levels [41]. By continuously monitoring these extracellular cues, bacterial cells adjust their gene expression in response, leading to the regulated production of proteins [42].

Bacterial cells possess the capability to detect a variety of environmental signals, such as,

- **Nutrients**: Bacteria can detect sugars, amino acids, and other essential nutrients, which helps them to find food sources for growth and survival. Sensing nutrients allows bacteria to move towards favorable environments through processes like chemotaxis [43].

- **Toxic Compounds**: The ability to sense harmful chemicals or antibiotics enables bacteria to avoid or neutralize threats, contributing to their resilience in hostile environments [44].

- **Quorum Sensing Molecules (Autoinducers)**: These molecules are used by bacteria to communicate with each other, coordinating activities such as biofilm formation, virulence, and resistance mechanisms. Quorum sensing is critical for bacterial communities to function as a collective, rather than as individual cells [45].

- **pH Levels**: Sensing acidity or alkalinity allows bacteria to maintain internal pH homeostasis and adapt to or colonize different environments, which is vital for their metabolic processes [46].

- **Temperature**: Bacteria can sense changes in temperature, enabling them to seek out optimal conditions for growth or to induce stress responses that may protect them from heat or cold shock [47].

- **Metal Ions**: The ability to sense concentrations of various metal ions, such as iron, is crucial for bacterial nutrient acquisition and regulation of metal homeostasis, impacting their growth and pathogenicity [48].

- **Light (Phototaxis)**: Some bacteria can sense light, enabling them to move towards or away from light sources, which can be important for photosynthetic bacteria or those using light as a cue for environmental changes [49].

### Communication

Bacteria inhabit virtually all environments, existing within complex ecosystems where they interact with one another. From the standpoint of MC, these bacterial communities form intricate networks of communication, with individual bacteria functioning as transmitters, receivers, or both. For instance, *P. aeruginosa* features three distinct Quorum Sensing (QS) systems—Las, Rhl, and PQS (Pseudomonas Quinolone Signal)—dedicated to communication. The signaling molecules utilized include 3-oxo-C12-HSL in the Las system, C4-HSL in the Rhl system, and 2-heptyl-3-hydroxy-4(1H)-quinolone (HHQ) in the PQS system [50, 51, 52]. Analogously, the functional roles and behaviors of bacterial cells in these networks can be likened to various electronic components utilized in silicon-based technology networks, underscoring the sophisticated nature of bacterial communication. These sensing and communication capabilities are crucial for bacteria to understand and adapt to their environment, a key aspect explored in this thesis.

### Gene Expression

Bacterial cells meticulously monitor extracellular conditions to regulate gene expression and, consequently, protein synthesis. This regulatory mechanism is highly complex and consists of a multitude of elements such as mRNA, activators, repressors, the genetic information encoded within DNA, RNA polymerase, and protein-binding sites [53, 54, 55]. This sophisticated process steers cellular behavior to enhance survival, often likened to a form of decision-making. Moreover, it can be viewed as a chemical-based computational activity, processed through the intricate pathways of the GRN. Within this network, numerous molecular signals are transduced, leading to gene expressions in both parallel and sequential fashions. These operations are orchestrated by genetic circuits, which can comprise anywhere from about 100 to over 11,000 genes, depending on the organism. For instance, Sorangium cellulosum strain So0157-2 is noted for possessing the largest known genome in this context [56] [57].

To summarize, the inherent computing capabilities of bacterial cells, such as their sophisticated sensing, intricate communication networks, and regulated gene expression, make them viable candidates for biocomputing. These cells exhibit a remarkable ability to process information and adapt to their environments through biochemical interactions, resembling the principles of neuromorphic computing systems. The exploration of these capabilities underscores the potential of bacteria to revolutionize biocomputing, offering a sustainable and efficient alternative to traditional silicon-based technologies.

## 1.2 Research Scope of the Thesis

This section outlines the scope of the doctoral research detailed in this thesis. It begins with Section 1.2.1, which highlights the challenges associated with contemporary silicon-based computing systems, setting the stage for the main research focus.

Following this, Section 1.2.2 delineates the research objectives, providing a clear roadmap for the investigation undertaken.

### 1.2.1 Challenges

The intertwining of AI with silicon-based computing has marked a significant societal shift, revolutionizing industries and embedding efficiencies into our daily lives, propelling advancements across numerous sectors, automating tasks and enhancing decision-making with minimal human input. Despite these strides, the union of AI and silicon technology faces challenges, including energy demands and biocompatibility. In addition, existing biocomputing platforms exhibit better energy efficiency and biocompatibility, but the generalizability is limited. Additionally, the dependence on silicon highlights the need for alternative innovations toward global sustainability goals.

Therefore, this section is dedicated to examining some of the crucial challenges associated with existing AI methodologies, as outlined below:

- **C1: Energy efficiency and physical scale -** One of the paramount challenges facing the field of AI systems today concerns their energy efficiency and physical scalability. Traditional AI systems, particularly those based on silicon computing architectures, consume significant amounts of power, especially as they scale up to handle complex tasks and larger datasets. This not only poses sustainability concerns but also limits the practical deployment of AI solutions in energy-sensitive environments. On the other hand, neuromorphic systems, designed to emulate the brain's energy efficiency and computational prowess, offer a promising avenue for reducing energy consumption. However, the physical realization of such systems at a scale that matches or exceeds the capabilities of current AI technologies remains a significant technical hurdle. The challenge lies in designing neuromorphic hardware that can operate at the low power levels characteristic of the human brain while still delivering

the computational speed and capacity required for advanced AI applications. Overcoming these limitations in energy efficiency and physical scalability is crucial for the next leap forward in AI and neuromorphic computing, paving the way for more sustainable, powerful, and widely applicable cognitive computing solutions.

- **C2: Biocompatibility -** The biocompatibility of silicon technologies presents a notable challenge, particularly as the integration of electronic devices with biological systems becomes increasingly desirable for medical and research applications. Silicon, the cornerstone of contemporary electronics, poses several biocompatibility issues that complicate the deployment of silicon-based devices in long-term implants or sensors intended for monitoring or therapeutic purposes within the human or animal body. While silicon's electrical properties and manufacturability have made it a material of choice in the electronics industry, its integration into biologically interactive applications necessitates careful consideration. This is especially the case if we consider interactions with living tissues. Addressing these biocompatibility challenges is essential for advancing silicon technologies in biomedical fields, requiring innovative approaches to materials science and device engineering to ensure both the functionality and safety of silicon-based bioelectronic systems.

- **C3: Generalizable Biocomputing -** The generalizability of biocomputing approaches stands as a formidable challenge within the realm of computational biology and bioinformatics. Biocomputing integrates biological principles with computational techniques to solve general computing, inherently faces the hurdle of transferring findings and methodologies across different biological systems and scales. Biocomputing addresses a range of problems, including computationally hard problems using biological substrates. Examples include temporal computing with brain organoids, where electrical sig-

nals serve as inputs and outputs; bacterial computing, where metabolites are inputs and phenotypes are outputs; and using slime mold to find optimum routes. Inputs for biocomputing can include biological data like DNA sequences or metabolic profiles, while outputs are predictions, simulations, or optimized solutions. These models can be trained using both conventional and unconventional methods. For example, brain organoids adapt to stimuli using their plasticity, while bacterial computing involves offline model training followed by engineering genetic circuits. However, these approaches face many limitations when it comes to generalizability due to immense diversity of biological organisms and the complexity of biological processes, which can vary significantly even within the same species. The ability to create models and algorithms that are universally applicable or at least broadly generalizable across various computing tasks is a critical yet difficult objective. This difficulty is compounded by the necessity to accurately model the intricate, nonlinear interactions within biological systems, which often involve a level of detail and specificity that resists straightforward generalization. Overcoming these challenges is essential for the advancement of biocomputing, as it seeks not only to provide insights into specific biological phenomena but also to develop tools and approaches that have wide applicability.

### 1.2.2 Research Objectives

In response to the multifaceted challenges in silicon-based technologies including contemporary AI and neuromorphic systems, and other biocomputing systems emphasized in Section 1.2.1, this thesis investigates the potential of bacteria as a wet-neuromorphic solution.

The concept of leveraging bacteria as computing systems presents a fascinating yet challenging frontier in the realm of biocomputing. Despite the intriguing computational capabilities demonstrated by bacterial communities, such as infor-

mation processing, decision-making, and network communication, there remains a significant gap in our understanding of the underlying mechanisms that enable these biological processes. This lack of comprehension presents considerable challenges in harnessing bacteria for practical computing applications. The inherent limitations in predictability, reliability, and control over bacterial computing processes further compound these obstacles. Unlike traditional silicon-based systems, where behavior can be precisely defined and outcomes reliably predicted, bacterial systems operate with a level of stochasticity and environmental sensitivity that can be difficult to model and harness. Moreover, the unique properties of bacterial natural computing, such as the ability to adapt and evolve over time, while advantageous in biological contexts, introduce variables that challenge the consistency and repeatability required of conventional computing systems. Consequently, while the potential for using bacteria as a basis for computing systems opens up exciting possibilities for bio-inspired computing architectures, the path forward is hindered by our incomplete understanding of their internal natural computing capabilities. Addressing these knowledge gaps and developing methods to reliably integrate and control bacterial computing activities are crucial steps toward realizing the full potential of bacterial systems in computing applications.

This thesis seeks to harness the inherent computing capabilities of bacterial systems, viewing them not just as biological entities but as components in a living, bio-computational network. Initially, understanding the collective behaviors and computational dynamics at the biome and population levels provides insights into how bacterial communities process information, make decisions, and communicate. This macroscopic perspective sets the stage for a deeper investigation into the computational properties at the single-cell level, focusing on the GRN as the fundamental mechanism driving bacterial computation.

By examining the GRN, we delve into the cellular logic that underpins bacterial decision-making and information processing, revealing a complex, yet potentially

harnessable system for computing tasks. This thesis paves the way for assessing the reliability, energy efficiency, and practical applicability of bacterial systems for general computing tasks, such as regression and classification. The unique energy dynamics of bacterial metabolism offer a model for ultra-low-energy computing, addressing one of the critical limitations of silicon-based systems. Moreover, the inherent scalability and adaptability of bacterial populations, coupled with their natural biocompatibility, present a solution to the challenges of genaralizability with biological systems. By translating these biological computing properties into a framework that can be applied to conventional computing challenges, the vision of bacteria-based wet-neuromorphic computing systems holds the promise of revolutionizing our approach to AI and neuromorphic engineering, offering a symbiotic blend of biological and computational intelligence.

Therefore, this thesis is structured around three core research questions designed to thoroughly investigate the inherent computing capabilities of bacterial cells as explained below. These questions specifically focus on natural communication processes, gene regulation-based cellular functions, and their potential applications from a computing perspective, exploring how these biological mechanisms can be harnessed for advanced biocomputing solutions.

- **RQ 1: How can communication of bacterial multi-species computing be used to understand population network structures?** The research question investigates the intricate communication mechanisms within bacterial ecosystems and how these interactions shape the overall network dynamics. Bacteria naturally thrive in diverse ecosystems, engaging in complex communication through molecular signaling, and cross-feeding. These signaling processes allow bacteria to detect and respond to environmental cues and signals from other bacterial cells or different cell types. This MC facilitates a coordinated response, enabling bacterial communities to adapt their metabolic activities to optimize survival and function within their environment. By

studying these interactions, researchers aim to understand how bacterial populations self-organize into network structures, maintaining ecological balance and resilience.

When bacteria alter their metabolic activities in response to environmental signals, it leads to compositional changes within the ecosystem, thereby influencing the community's collective behavior. This dynamic adaptation not only optimizes resource utilization but also enhances the community's ability to withstand environmental stresses. The study of multi-species bacterial communication can reveal how these alterations in metabolic activities drive the formation of complex population network structures. Understanding these principles can also provide valuable applications in biotechnology, medicine, and environmental management by harnessing the natural adaptability and resilience of bacterial ecosystems.

- **RQ 2: Can gene regulation networks be used to discover artificial neural networks for biocomputing?** Observations of bacterial metabolic activities and adaptations at the network structure level suggest a complex computational behavior originating at the single-cell level. To explore this phenomenon further, the second research question is formulated.

  Bacteria exhibit natural computing abilities through their gene expression processes, where GRNs function as intricate, complex graph networks. These networks manage gene activity in response to various stimuli, effectively processing information in a manner akin to computational systems. This thesis aims to harness the inherent computing capabilities of bacteria by using their GRNs to infer and operate artificial NN for biocomputing purposes.

  This approach involves using bacterial cells themselves as the hardware for biocomputing, capitalizing on their natural regulatory mechanisms to perform computational tasks. The goal of this RQ is to create a biocomputing frame-

work where bacterial cells act as non-silicon processors, potentially leading to revolutionary advances in computing efficiency, biocompatibility and generalizability.

- **RQ 3: How can the bacterial computing diversity be expanded by exploiting cellular plasticity?** This thesis focuses on harnessing the inherent computing abilities of bacteria without genetically engineering the cells. To fully exploit these capabilities, it is essential to evaluate the generalizability of bacterial computing. As previously discussed, specific GRN sub-networks can perform designated computational tasks. However, to achieve a broader range of computing functions, it is necessary to expand the search space for these GRN sub-networks. This approach involves identifying and utilizing the natural variability and adaptability of bacterial cells, known as cellular plasticity, to increase the diversity of computational tasks they can perform.

  Cellular plasticity refers to the ability of bacterial cells to adapt and reconfigure their gene expression profiles in response to different environmental conditions and stimuli. By studying and exploiting this plasticity, the RQ aims to identify a wider array of GRN sub-networks capable of performing diverse computing tasks. This approach not only enhances the computing diversity of bacterial cells but also leverages their inherent flexibility, making it possible to develop robust biocomputing systems that can dynamically adjust to different computational needs without the need for genetic modifications.

- **RQ 4: Can mathematical and pattern recognition applications be realized through bacterial neural networks?** Bacterial gene expression responds intricately to environmental conditions, creating a rich dataset of biological responses that can be analyzed and modeled mathematically. This RQ aims to identify specific mathematical functions from these gene expression patterns that can be harnessed to perform generic regression computing. By

decoding how bacterial GRNs respond to varying stimuli, it is possible to construct mathematical models that predict outcomes based on input conditions, effectively utilizing bacteria's natural computational processes for regression tasks.

Additionally, this research question investigates the bacterial cell's potential to recognize and classify complex patterns within data, akin to how artificial neural networks operate. By studying the gene expression responses and regulatory mechanisms, the RQ seeks to develop biocomputing framwork that can perform pattern recognition tasks.

Ultimately, this research aims to demonstrate that bacterial neural networks can be effectively utilized for both mathematical regression and pattern recognition, providing a novel and bio-inspired approach to computational problem-solving.

- **RQ 5: What search algorithms can be developed to discover natural GRNN for biocomputing applications?** Bacterial GRNs exhibit event-driven computing properties due to the specificity of gene regulation in response to environmental stimuli. The question lies with identifying relevant sub-networks within the larger GRN that can effectively perform desired computational tasks.

  To address this challenge, this RQ focuses on developing search algorithms capable of extracting functional sub-networks from the full GRN. These sub-networks are critical as they pinpoint the chemical inputs and outputs necessary for executing bacterial computing processes.

## 1.3 Summary

The contemporary computing faces observable limitations, particularly in areas where energy efficiency (**C 1**) and biocompatibility (**C 2**) are crucial. While biocom-

puting approaches show promise in addressing these challenges, effectively tackling both energy conservation and compatibility, they still grapple with the issue of generalizability (**C 3**). This remains a significant hurdle, as achieving broad applicability across various contexts without compromising performance.

In summary, thesis explores the potential of bacterial cells for biocomputing by addressing several key research questions. Firstly, it investigates how communication within multi-species bacterial communities can elucidate population network structures and optimize computing responses to environmental changes (**RQ 1**). Secondly, it examines whether bacterial gene regulation networks can be used to develop a novel bacterial biocomputing concept, leveraging their natural computational capabilities (**RQ 2**). Thirdly, it explores how cellular plasticity can be harnessed to expand the diversity of bacterial computing tasks, enhancing the generalizability and adaptability of biocomputing systems without genetic modifications (**RQ 3**). Fourthly, it seeks to determine if mathematical regression and pattern recognition tasks can be realized through bacterial neural networks by analyzing their complex gene expression patterns (**RQ 4**). Lastly, the research aims to design algorithms for extracting specific sub-networks from bacterial gene regulatory networks to identify key chemical inputs and outputs for effective bacterial computing (**RQ 5**). Together, these questions form a comprehensive investigation into the capabilities and applications of bacterial cells as natural computing units.

## 1.4 Organization

The rest of the thesis is organized as follows. Chapter 2 introduces the biological background and the state-of-the-art laying the foundation for biocomputing approaches. Next, Chapter 3 discusses methodologies used for the studies in this report starting from the designing of simulation tools/computational models to analytical models. Chapters 5 to 10 present publications associated with this thesis.

# Chapter 2

# State-of-the-art

As this thesis focuses on introducing bacteria as a novel wet-neuromorphic computing platform, this chapter discusses state-of-the-art biocomputing solutions. Initially, Section 2.1 presents an overview of biocomputing, mainly focusing on DNA Computing (Section 2.1.1) and organoid intelligence(Section 2.2.1). Next, Section 2.3 is dedicated to one of the key approaches of biocomputing, which is **Bacterial computing** as it is the focal point of this thesis. This section discusses state-of-the-art methods that use consortia (Section 2.3.1) of cells and whole-cell computing (Section 2.3.2). Finally, section 2.4 summarizes the key insights of biocomputing and current challenges that need further attention.

## 2.1 Overview of Biocomputing

Biological components exhibit an extraordinary ratio of computing power to physical scale and energy efficiency. This was exemplified in 2013 when it took the world's fourth-largest supercomputer 40 minutes to simulate merely 1 second of 1% of human brain activity [58]. Additionally, the brain's storage capacity is estimated at about 2,500 terabytes owing to its 86–100 billion neurons forming over a quadrillion (more than $10^{15}$) synaptic connections, showcasing the immense computational and storage capabilities inherent in biological systems [27]. Consequently, researchers are

blending biology with computer science to harness biological mechanisms for computational tasks, leading to the burgeoning field of biocomputing. Prominent among these approaches are those that leverage DNA, proteins, and cells for generic computing tasks. These methods provide significant benefits over conventional silicon-based computing, particularly in terms of energy efficiency, biocompatibility and parallel processing capabilities.

### 2.1.1 DNA Computing

DNA computing has been a revolutionary research field under the domain of biocomputing for decades [59, 13, 60, 61]. Similarly to binary data encoded using zeros and ones, DNA strands are encoded with four nucleotides (A, T, C, G), offering a unique method of data storage. These nucleotides are positioned every 0.35 nm along the DNA, enabling an extraordinary data density of one bit per cubic nanometer, allowing for the storage of approximately 455 billion GB of data per gram [62]. DNA's base pair complementarity introduces two essential computing elements: (1) a processing unit comprising enzymes that can manipulate DNA through actions like cutting, copying, and pasting, and (2) a storage unit within the sequences of the DNA strands themselves. This setup allows DNA computing to execute operations in parallel across multiple DNA strands, significantly enhancing its computational power. Notably, DNA replication in bacteria can occur at a rate of approximately 500 base pairs per second, far surpassing the replication speed in human cells and effectively translating to a data processing rate of about 1000 bits per second. With multiple replication enzymes working concurrently, this rate can exponentially increase, reaching up to 1 terabits per second after 30 replication cycles, showcasing the immense potential for memory capacity and parallel processing in DNA computing [63].

Utilizing the well-established programmability of DNA fueled by base sequence designing capabilities, one innovative DNA system termed as DNA droplets was de-

veloped in 2023. This approach combines the dynamic, fluid-like properties with the computing capabilities of DNA creating a hybrid approach that facilitates in powerful designing tool for intelligent and dynamic cell-like machinery [64]. Another highlighted achievement is by Elowitz and Liebler, who engineered an *E. coli* strain with a synthetic network that causes it to oscillate, producing green fluorescent protein in periodic cycles [65]. Later, Gardner and colleagues constructed a genetic toggle switch in bacteria, capable of flipping between stable states in response to specific chemical or thermal stimuli [66], Furthermore, using DNA's computing principles, researchers have designed automata [67], logic circuits [68, 69], neural networks [70] and DNA-based programmable gate arrays [71], showcasing their capability in molecular information processing and the creation of synthetic intelligent devices.

It is important to note that DNA computing approaches significantly depend on genetic engineering techniques resulting in a range of limitations. The scalability and integration are key issues, where adding more genetic circuits increases metabolic load and can cause unforeseen interactions, leading to system inefficiencies. Stability and robustness are also concerns, as genetic mutations and environmental factors can alter circuit functions, affecting consistency and reliability. Containment and biosafety risks arise from the potential environmental release of genetically modified organisms, necessitating robust containment measures and regulatory compliance. Furthermore, the complexity of designing predictable genetic circuits is compounded by biological noise and the intricate interactions within cellular systems, making the design process both complex and resource-intensive.

## 2.2 Wet-Neuromorphic Systems

### 2.2.1 Organoid Intelligence

Organoid Intelligence (OI) is a pioneering approach in biocomputing, using brain organoids derived from human stem cells to mimic learning and memory functions. This field seeks to blend bioengineering with scientific advancements within an ethical framework, potentially surpassing silicon-based computing in efficiency and processing power [27].

One interesting example is the "DishBrain", a revolutionary system that connects biological neural networks (BNNs) with silicon technology, using neurons' natural electrical communication. It grows cortical cells from rodent embryos on microelectrode arrays in a nutrient-dense setup for extended periods. DishBrain operates in a closed-loop feedback system, dynamically interacting with neural cultures by "reading" and "writing" sensory data, allowing neural actions to influence sensory inputs in real-time. This setup is designed to study learning effects in BNNs within a virtual environment. An early experiment with DishBrain successfully simulated the arcade game "Pong" using inputs across eight electrodes, demonstrating its foundational capabilities [19].

Another fascinating study introduces "Brainoware", an AI hardware utilizing 3D biological neural networks from mature human brain organoids as a dynamic, living network for computing. It processes information through spatiotemporal electrical stimulation on a multielectrode array, demonstrating learning abilities and handling complex tasks like solving non-linear equations. Employing human brain organoids for dynamic, unsupervised learning and feature engineering, it effectively turns temporal inputs into computational solutions with its unique physical reservoir traits, such as nonlinear dynamics and spatial processing [35].

On the contrary, there are a few key challenges associated with organoid computing that can be identified. Keeping the organoid alive is a critical challenge that

requires developing advanced artificial blood vessels that can deliver the required nutrients. Further, the natural brain comprises various types of cells, that are vital for enhanced learning and memory capacities. Adopting this diversity with cell types like oligodendrocytes and astrocytes, and optimizing culture conditions to promote learning and memory are still considered limitations. Additionally, the introduction of next-gen 3D microelectrode arrays and innovative input devices is critical for the real-time control and complex recording methods needed in organoid intelligence research [27].

### 2.2.2 Proteinoid Computing

Proteinoids are created through thermal polycondensation of amino acids at 160–200°C under inert conditions. In an aqueous medium, these hollow microspheres can swell and exhibit dynamic electrical activity, including spontaneous electrical potential bursts, known as flip-flops, and smaller potential fluctuations during their inactive phases, suggesting a complex behavior similar to biological neurons [72, 73]. These proteinoids are durable and resistant, withstand extreme environments and catalyze reactions, demonstrating the ability to self-assemble into more complex structures, reflecting a significant step towards understanding early cellular life forms and synthetic biology applications [37, 74].

Proteinoids, also called proto-neurons due to the intriguing characteristic of maintaining a membrane potential of 20 to 70 mV without external stimulation and displaying oscillatory electrical potentials. These oscillations, lasting days or weeks, underline their potential as neuromorphic computing devices. This study explored leveraging proteinoid microspheres' unique electrical behaviors for unconventional computing applications, suggesting a novel approach to biomimetic technology development [75, 76, 38].

Based on the unique voltage patterns that align with various logical outputs of proteinoids, the researchers have identified four types of logical gates: AND, OR,

XOR and NAND [74]. Similarly, another research group was able to utilize reservoir computing properties of proteinoids with colloidal mixtures of ZnO, the possibility to conduct logical operations [77]. However, the computing potential of the proteinoids spans further exhibiting high-level characteristics including learning, memory and forgetting by modifying current response influenced by the previous signals. [38].

Further, proteinoid microspheres, as the foundational elements of networks, enable rudimentary communication for computational capabilities. With network expansion, these microspheres form complex structures, leveraging molecular connectivity for enhanced functionalities, setting them apart from traditional computing that depends on external connections. This reveals a proteinoid microsphere nervous system, providing a model for understanding their interaction and organizational structure [78].

While proteinoid computing demonstrates impressive computational abilities, exploring proteinoid communication remains significantly unrevealed [79]. Additionally, the synthesis process of proteinoids consists of five-step synthesis stages heating, water treatment, centrifugation, dialysis, and lyophilization [80]. This complexity poses hurdles for practical applications and scalability.

## 2.3 Bacterial Computing

Bacterial computing is one of the emerging fields under the domain of biocomputing that has gained attention in recent years. This field explores engineering computing circuits on cells and populations. [81].

### 2.3.1 Consortium Computing

In bacterial consortium computing, the principles from distributed computing have been innovatively applied within multicellular systems through the alteration of cell-cell communication pathways [82, 83].

A study by Sarkar et. al developed cellular devices acting as artificial neuro-synapses in bacteria that process input chemical signals through a combination of linear and non-linear functions to generate fluorescent outputs. The creation of these devices involved establishing a set of rules that link truth tables, ANN equations, and the design of cellular devices. This approach, which integrates design directly from mathematical models without needing traditional circuit diagrams, marks a departure from conventional cellular computing methods [84].

Inspired by the similarities between artificial neural networks and cellular networks, another team developed a system for pattern recognition using bacterial consortia. This system uses quorum sensing for communication between receiver and sender bacteria, where chemical inducers create input patterns that prompt senders to emit signaling molecules at programmed levels, serving as adjustable weights. A gradient descent algorithm was also created for optimizing these weights, and tested on recognizing 3x3-bit patterns. This approach highlights the potential for sophisticated computing within microbial communities [85].

### 2.3.2 Whole-Cell BioComputing

In contrast with multi-cellular computing systems, whole-cell computing approaches are being suggested, focusing mainly on their remarkable survival skills in harsh environments by making complex decisions based on information processing that involves memory, sensing, feedback, and communication within a single bacterial cell. An *E. coli* bacterium as a whole cell consists of around 4.6 million base-pairs [86], and possesses a memory capacity equivalent to 9.2 megabits, enabling it to code for up to 4300 different polypeptides controlled by hundreds of promoters. Subsequently, synthetic systems leveraging whole-cell interactions present promising avenues for broadening the scope of computational tasks achievable with living systems [25].

Researchers have explored the potential of genetic engineering to replicate the functionality of silicon semiconductors, termed 'silicon mimicry,' that aims to har-

ness the sophisticated processing powers of microorganisms to emulate the operational principles of silicon-based devices [87].

Another key study proposed "Ribocomputers", *de-novo* devices that transform key circuit functions into RNA-RNA interactions, detectable through toehold switch mechanisms, facilitating RNA-based computational processes within cells. They operate at the RNA level, centralizing all components for sensing, computation, and output into one complex platform, enhancing efficiency and reliability. As demonstrated in *E. coli*, these devices can effectively execute complex logic functions, such as two-input logic with significant dynamic ranges and scalable to more sophisticated configurations like four-input AND, six-input OR, and 12-input expressions [88].

Another pivotal study showcased the development of a genetic toggle switch in *E. coli*, forming a bistable gene-regulatory network. This design, based on two interlocking repressible promoters, can switch between stable states with precise chemical or thermal triggers [66]. Further, Green et. al introduced a novel synthetic network called the repressilator, utilizing three transcriptional repressor systems not found in natural biological clocks to create oscillations in *E. coli* [65].

In contrast to the bacterial logic gate operations, in 2022, a study designed a neural network inside a single *E. coli* cell. First, they trained a neural network offline to adjust the connections between neurons for specific input-output responses. This trained network was then simulated in silico using Gro language to program bacterial cells, observing their behavior at various times. Then, this neural network was translated into a genetic network inside a plasmid utilizing the Cello platform and Verilog language [40].

The introduction of "perceptgene" can be identified as one of the most promising approaches in recent discoveries of bacterial computing. This perceptron operates in the logarithmic domain, facilitating the creation of devices that can calculate minimum, maximum, and average values from analog inputs. Innovations include

Table 2.1: Evaluation of challenges and limitations of state-of-the-art approaches where, C1:Energy efficiency and scale, C2: Biocompatibility, and C3: Generalizability.

| | SOTA | C1 | C2 | C3 |
|---|---|---|---|---|
| Non-bacterial | Organoid Intelligence/ Proteinoid Computing Intelligence | Energy efficient and smaller in size. | End-to-end system is not biocompatible. | Capable of general computing. |
| Bacteria | Consortium Computing | Less energy efficient and larger in size compared to whole-cell computing. | As most approaches use engineered cultures, the biocompatibility has limitations. | Can only perform limited computing tasks. |
| | Whole-cell computing | Energy efficient and smaller in size. | As most approaches use engineered cells, the biocompatibility has limitations. | Can only perform limited computing tasks. |

multi-layer circuits capable of executing soft majority functions, analog-to-digital conversion, and ternary switching. Additionally, a programmable perceptgene circuit has been engineered to switch between OR and AND logic functions through small molecule induction. This approach also opens avenues for optimizing circuits using artificial intelligence algorithms, marking a significant leap in bacterial computing capabilities [20].

## 2.4 Summary: Challenges and Limitations

This section outlines the challenges and limitations of cutting-edge research, primarily focusing on the biocomputing domain. Although organoid-based computing

approaches are energy-efficient and compact, they significantly struggle with viability and scalability in enhancing computational power. Additionally, they depend on external inputs, leading to biocompatibility concerns. Similarly, genetic engineering-based computing methods are also energy-efficient and small-scale, yet their computational abilities are severely restricted by the limitations of genetic engineering techniques. Moreover, bacteria-based computing, which employs fixed-engineered genetic circuits, lacks adaptability and generalizability. Consequently, this thesis aims to explore the inherent computing capabilities of bacteria to address these limitations.

# Chapter 3

# Research Summary

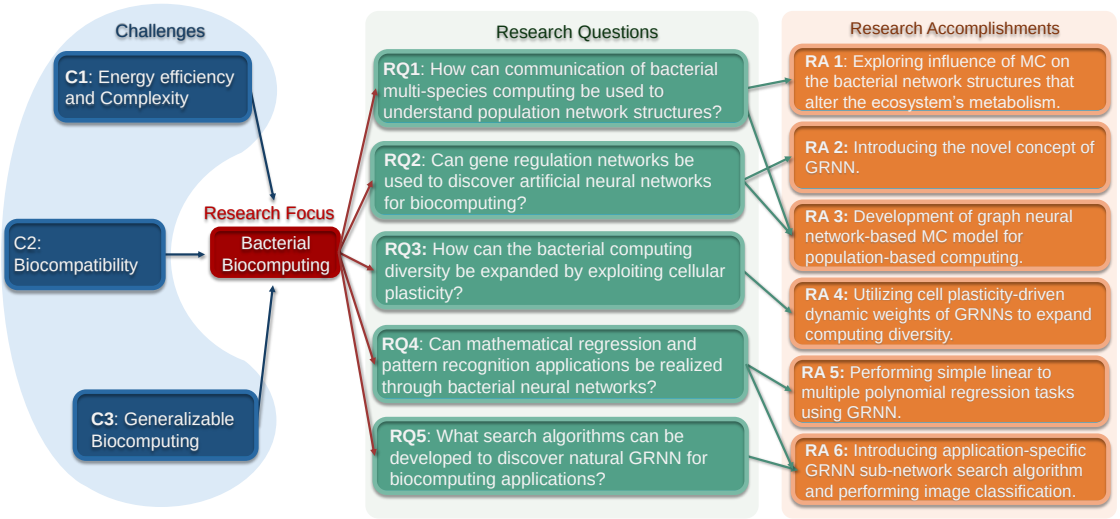



Figure 3.1: An overview of the research plan and the mappings between the challenges, research questions and research accomplishments.

This chapter outlines the Research Accomplishments (RAs) aligning with the Research Questions (RQs) as shown in Fig. 3.1. As depicted in this figure, this thesis highlights three key challenges in the field of computing. The first two challenges, energy efficiency (**C1**) and biocompatibility (**C2**), have been investigated extensively. Biocomputing approaches are suggested in the literature as potential solutions for (**C1**) and (**C2**). However, the generalizability of these biocomputing solutions (**C3**) is still insufficient. Therefore, this thesis focuses on "Bacterial

Biocomputing" as a solution for the three identified challenges. First, Section 3.1 analyzes bacterial MC under **RA 1** in order to understand its role in population-level computing (**RQ 1**). Subsequently, Section 3.2 dives deep into the intrinsic computing behaviors governed by the GRN under **RA 2** and introduces the concept of Gene Regulatory Neural Networks (GRNNs) as solutions for **RQ 2**. Further, purely focusing on GRNN computing, Section 3.3 evaluates the structural and algorithmic complexities and energy consumption under **RA 2** to find more solutions for **RQ 2**. The influence of the cell-cell communications on computing dynamics of the GRNN is investigated as research questions in **RQ 1** and **RQ 2**, which is explored in Section 3.4 as **RA 3**. Subsequently, **RA 4** in Section 3.5 and 3.6 analyzes how cell plasticity influences GRNN computing targetting **RQ 3**. Next, **RA 5** seeks to determine if GRNN can perform mathematical regression by analyzing complex gene expression patterns to answer **RQ 4** in 3.6. Finally, in the same section, **RA 6** aims to design algorithms for extracting application-specific sub-networks from GRNN and perform classification tasks aligning with **RQ 5**.

## 3.1 Bacteriome MC Analysis

Bacteria engage in sophisticated communication and interaction mechanisms across various contexts, including interactions between bacteria themselves, different bacterial populations, non-bacterial cells (e.g., epithelial cells), and viruses. Although the literature highlights that bacteria use electrical pulses for communication, MC is the primary mode of interaction among bacteria. These MC-based interactions are viewed from two main perspectives: Quorum Sensing (QS), which governs collective behavior based on population density, and cross-feeding interactions, highlighting the metabolic exchanges and influences between different bacterial species and other organisms. This multifaceted communication network underscores the versatility and sophistication of bacterial interactions within their ecosystems. These sophis-

ticated communication and interaction systems are crucial for the decision-making processes of bacteria, which can be interpreted as a form of population-based computing mechanism that influences their survivability, and ecological equilibrium.

This thesis initially concentrates on the MC mechanism to explore its impact on population dynamics aligning with **RQ1**. As the use-case for this research task, the analysis focuses on the human Gut Bacteriome (GB).

The human GB is an extensive bacterial ecosystem within the human gut, often regarded as a virtual organ that plays crucial roles in the host's metabolic functions through molecular interactions. It hosts approximately 100 trillion microorganisms that form intricate networks by exchanging metabolites with the host and other bacterial populations [89] performing essential tasks like nutrient extraction and metabolite absorption, including amino acids, vitamins, bile acids, and short-chain fatty acids (SCFAs). Distinct metabolic pathways in bacterial species contribute to their varied roles within the human GB. These pathways, documented in databases like Metacyc [90] and KEGG [91, 92, 93], alongside literature on the metabolic activities of prevalent genera under different conditions [94], facilitate the development of a molecular interaction model in this thesis. The human GB is envisioned as a collection of numerous sub-networks that undertake vital functions such as SCFA production, culminating in a complex molecular interaction network with numerous nodes and interactions across diverse molecular species.

Gut microbiome databases like MicrobiomeDB [95] offer insights into bacterial compositions, in which the majority of the microbiome belongs to several phyla such as *Firmicutes*, *Bacteroidetes*, and *Actinobacteria*. Further, the human GB's composition is influenced by genetics, dietary habits, and age, while external factors like toxins, drugs, antibiotics, and certain diseases can lead to dysbiosis, disrupting metabolite production and impacting health [96]. Conditions linked to dysbiosis include inflammatory bowel disease, type-2 diabetes, obesity, and cancers [97, 98]. Resources like the Disbiome database [99] provide data on imbalances related to

various health conditions. Subsequently, the next sections present a theoretical framework under **RA 1** to analyze the dynamics of human GB from the perspectives of communications and computing aligning with **RQ1**.

### 3.1.1 Theoretical Framework

There are many studies that aim to identify the underlying causes of microbial behavioral changes and their health implications [100, 101]. Motivated by these studies, this work introduces a communication model (termed Virtual Gut Bacteriome - VGB) to better understand the interactions within the human GB's populations and resulting computing behaviors. The VGB model represents the human GB through a two-tiered framework focusing on metabolite-based MC between bacteria in order to understand their interactions and behaviors as shown in Fig. 3.2.

The upper layer - *"Bacterial Population Graph Layer"* simplifies the GB into a graph where bacterial populations are nodes linked by edges representing metabolite exchanges, encapsulating the metabolic functions of bacterial cells within each population. This includes considering the molecular interactions between the host and bacterial populations, as well as within the bacterial communities themselves, through the intake and emission of molecules. These interactions transform this layer into a directed multi-graph network that governs the metabolism.

The bottom layer - *"MC layer"* where nodes act as receivers or transmitters depending on their role in the MC network and the edges represent MC channels facilitating molecular signal transport between nodes (bacterial populations) via diffusion.

The model explores how molecular signals' changes impact the bacterial population graph layer, emphasizing the interconnectedness of the two layers. Factors influencing network node performance include molecule size, ligand-receptor attraction, binding noise, and detection thresholds. After receiving signals, nodes process them internally, potentially resulting in new signal production, modeled through

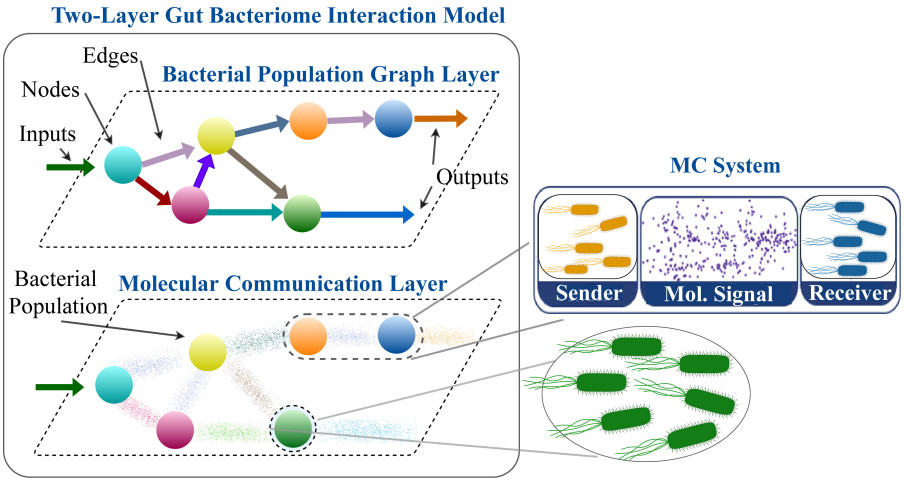



Figure 3.2: Illustration of a two-layer system model designed to explore molecular interactions within a simulated virtual GB.

Signal Processing Performance (SPP). This process is detailed down to a single bacterial cell level, considering metabolite reception, encoding/decoding processes, and secretion, thereby illustrating the complex interplay within the GB.

### 3.1.2 VGB Simulator

Investigating bacterial MC behaviors with high-dimensional, longitudinal datasets is crucial. Various computational methods exist for realistically modeling bacterial interactions within ecosystems, each with specific strengths and limitations. Overcoming existing challenges and catering to specific data extraction needs of this research task, a new simulation tool is developed in this research thesis. This tool can interpret ecosystem dynamics through MCs, design metabolic pathways, process data in parallel, and generate high-dimensional data. This simulator utilizes C++ and the CUDA platform to enhance simulation performance through parallel processing, reflecting the concurrent activities of bacterial populations. Each bacterial cell is represented by a GPU block, with threads within the block handling the cell's intracellular functions as illustrated in Fig. 3.3.

To model bacterial interactions accurately, the simulator employs metabolic flux to depict molecule exchange in a diffusive medium. It features a 3D environment con-

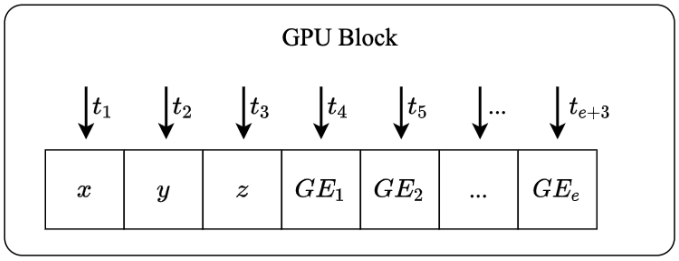



Figure 3.3: Illustration of GPU block and thread utilization where $t_e$ is the thread assigned for the gene $GE_e$. The memory array assigned for each GPU block contains placeholders for the specific location of the 3D environment in $x$, $y$ and $z$ coordinates.

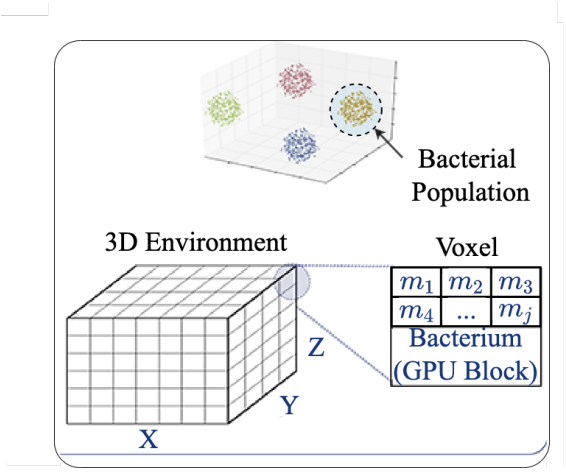



Figure 3.4: Illustration of the voxel architecture of virtual GB.

structed with voxel architecture, enabling detailed data extraction for each metabolite and bacterial cell as shown in Fig. 3.4. The voxel architecture is designed to store the concentration of each molecular species (denoted as $m_1, m_2, \ldots, m_j$) required for a simulation. This capability is crucial for accurately simulating the diffusion of molecules enhancing the simulation's ability to mimic complex biological environments. Further, it enables simulating the consumption or secretion of metabolites by bacteria which is vital for investigating computing capabilities. Subsequently, this voxel architecture further helps in dividing the 3D space into layers corresponding to different molecular types as shown in Fig 3.5, in order to investigate the role of each molecular type in isolation.

This simulator additionally offers the flexibility to introduce new bacterial species by simulating their unique metabolic pathways and physiological traits, including

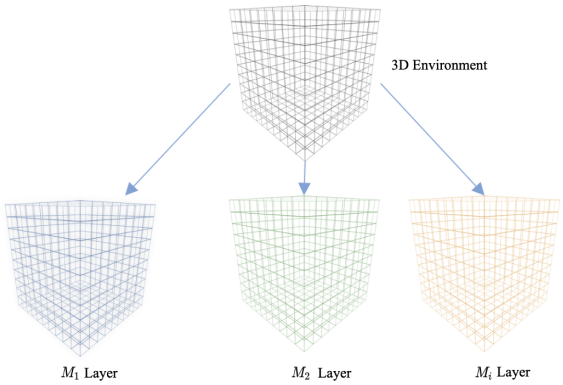



Figure 3.5: Depiction of the simulation environment, highlighting that molecular interactions for each species are organized into distinct layers

motility, shape, and size. This adaptability makes the simulator ideal for exploring a variety of scenarios, from studying different microbial ecosystems and metabolic functions to focusing on specific behaviors such as QS across diverse habitats. Additionally, it is equipped to track and log data related to metabolite consumption, production, accumulation, and bacterial growth, providing a comprehensive tool for detailed analysis of microbial interactions and behaviors.

### 3.1.3 *In-silico* Experiment and Results

In the initial study using this simulator, we focused on a specific segment pertinent to the production of SCFAs within the human GB. To configure the simulator accurately, we utilized data concerning the composition and metabolic activities of the human GB. The abundance data critical for determining the average human GB composition was sourced from data in microbiomeDB [102]. The calculated relative abundances are shown in Fig 3.6.

Subsequent to gathering metagenomic and metabolomic data from various sources and databases, including Oliphant et al. [94], KEGG [92, 91, 93], NJS16 [103], and MetaCyc [90], has allowed us to create species level network as shown in Fig 3.7. Further, the SCFA network is extracted and scaled up to the genera level which is elucidated in Fig. 3.8. The results indicate that *Bacteroides* dominate

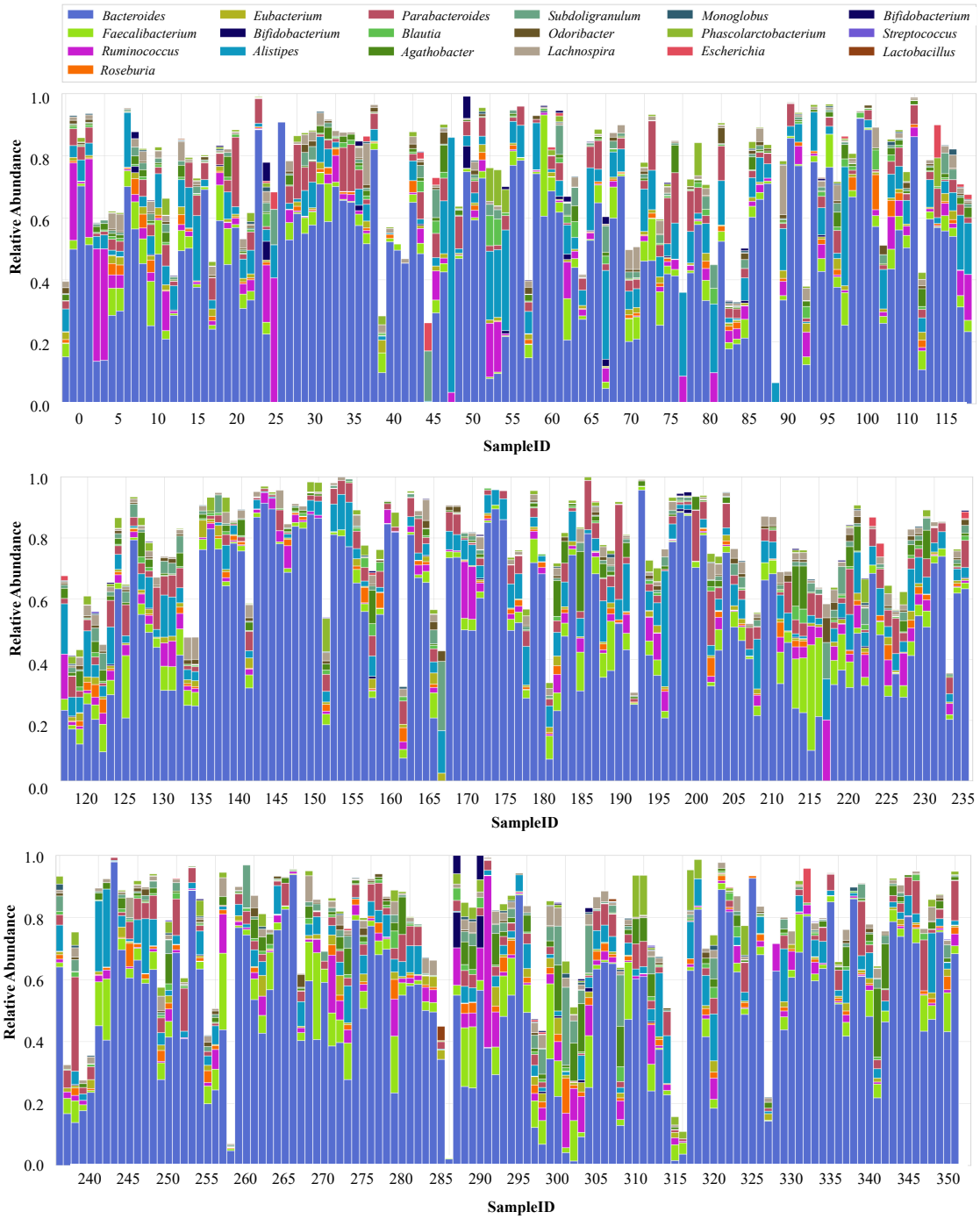



Figure 3.6: Illustration of relative abundances of 352 gut bacteriome samples used in the case study for SCFA production in the human GB. This data was collected from the MicrobiomeDB[102]. Please note that these figures show the collective species RA for that particular genus.

the composition of the gut bacteriome compared to all other genera. Additionally, *Faecalibacterium*, *Alistipes*, and *Parabacteroides* are observed as the next most abundant genera. Although numerous genera are present in the human gut bacteriome,

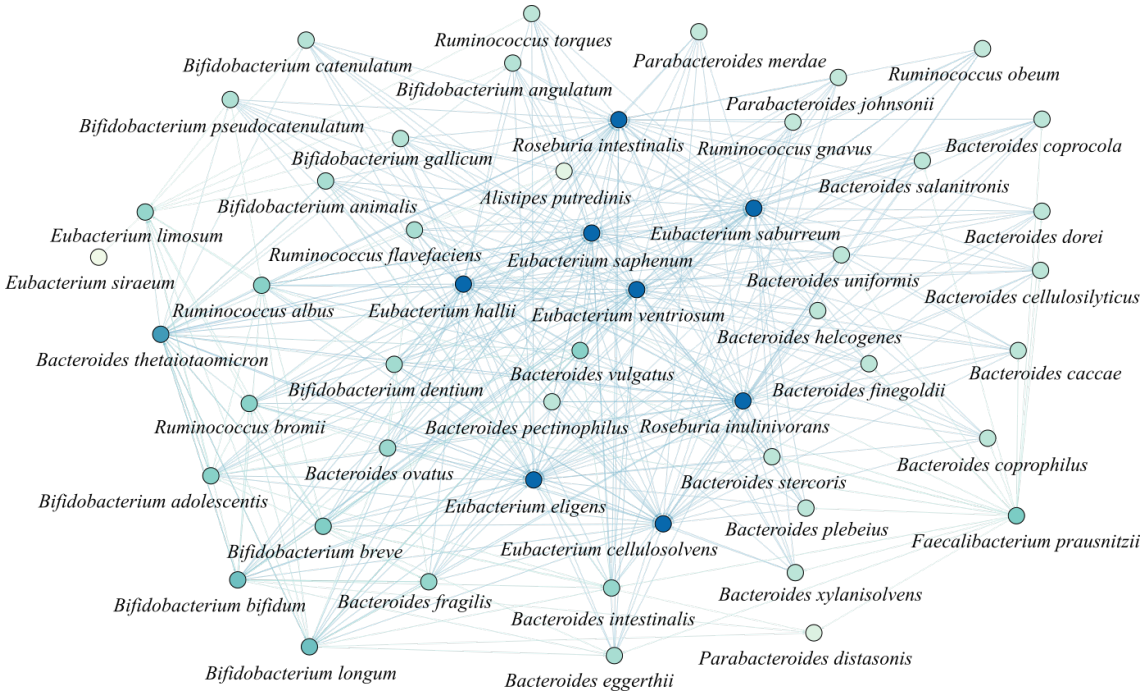



Figure 3.7: Illustrates a subgraph of the human GB only considering species of nine genera related to SCFA. The nodes are color-coded according the degree ranking, where the darker color indicates higher number of inward and outward interactions while nodes with lesser number of interactions are with lighter color.

their relative abundance is comparatively low.

Subsequently, two primary sets of experiments are conducted, as illustrated in Fig. 3.9. The first experiment examines how the system's inputs affect the connectivity structure of the VGB, while the second set alters the VGB's composition to study changes in metabolite production within our MC network.

### Analysis 1 - Input Impact on Human GB Structure

In order to investigate how the input affects the human GB structure, the first *in-silico* experiment is designed by varying the inputs to the VGB and observing the outputs. Here, the study focuses on the effect of glucose on three bacterial populations within the SCFA-producing subset of the virtual Gut Bacteriome. Fig. 3.10a visualize the sub-network associated with glucose intake and subsequent figures maintaining the same color coding. Figures 3.10b and 3.10c display changes in edge weight and population sizes, relative to an average human GB network, in

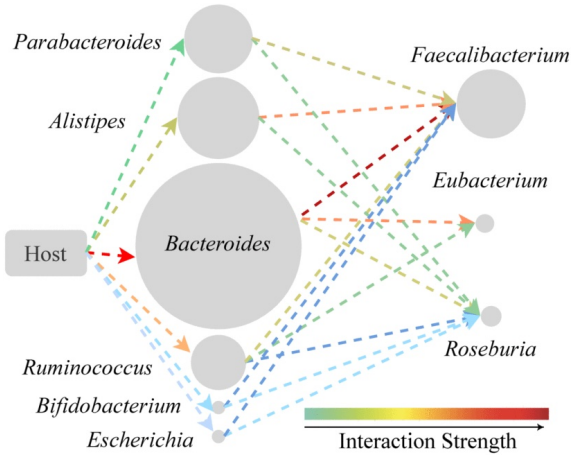



Figure 3.8: Representation of the phylum-level subgraph within the human GB associated with SCFA production.

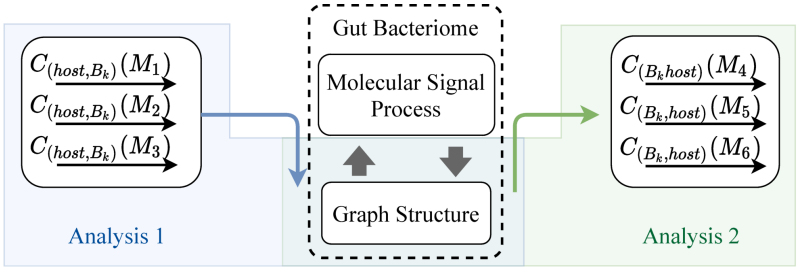



Figure 3.9: Illustration of the study's analytical framework, where Analysis 1 examines the impact of inputs on the graph structure, while Analysis 2 investigates the response of graph output to structural deviations.

response to variations in glucose input rates. These changes impact the interactions involving acetate and lactate, which are essential for the growth of *Faecalibacterium* and *Eubacterium*, respectively. Fig. 3.10b further delves into how these input rate variations indirectly affect the growth dynamics of these bacterial populations. It notes a steady increase in *Eubacterium* growth with glucose inputs up to double the standard level, while the other two populations stabilize. This pattern is attributed to the metabolic conversion stoichiometry, with acetate and lactate production from glucose impacting the growth of *Escherichia* and *Faecalibacterium* directly.

This, in turn, alters the overall structure of the network as shown in Fig. 3.11, where the variation graph structure is relative to the average human GB. When glucose is lower than the standard (1.0) level, the graph shows a significant deviation. Conversely, when glucose levels exceed the standard, the graph still deviates but

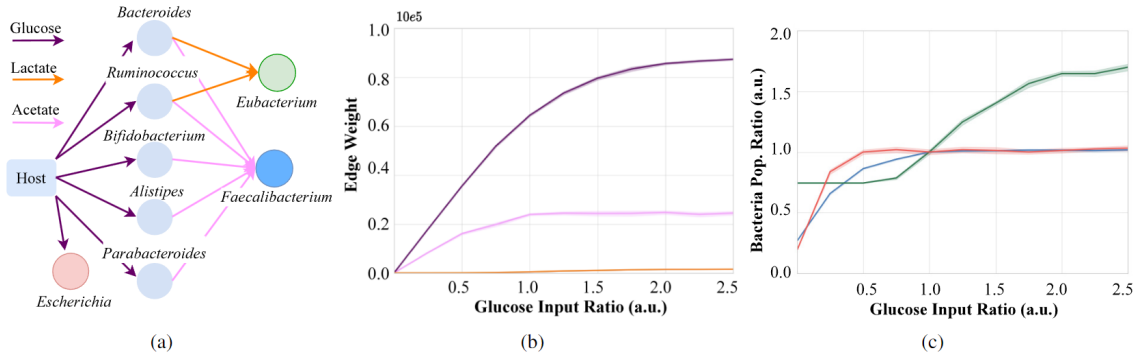



Figure 3.10: Variations in population sizes of *Faecalibacterium*, *Eubacterium*, and *Escherichia* from baseline levels in response to differing glucose concentrations, where a) subgraph depicting glucose consumption, b) edge weight dynamics of intermediate interactions, and c) patterns of population growth.

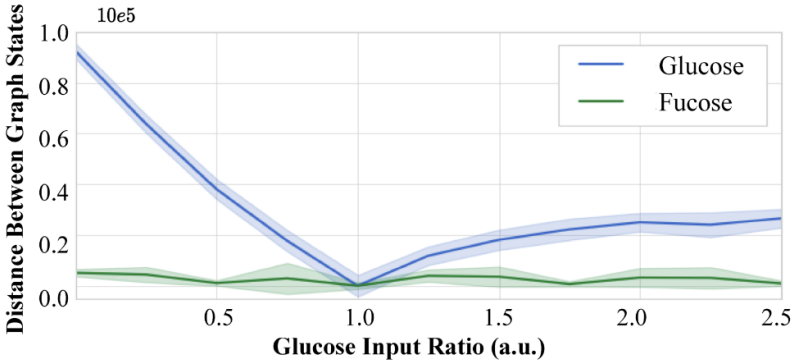



Figure 3.11: Dynamics of overall graph weights in response to variations in input types and their concentrations.

less dramatically than with lower levels. This indicates that the human GB is more sensitive to low glucose concentrations.

Further, these structural changes influence metabolite intake, intermediate metabolism, and resulting molecular outputs, which can be interpreted as modifications to the computational architecture of population-based bacterial computing systems. This relationship is investigated under *RQ 1*, leading to *RA 1*. By examining how these metabolic adjustments impact the overall computing structure, we can gain insights into optimizing bacterial populations for enhanced computational efficiency and adaptability in response to environmental changes. This investigation bridges the gap between biological processes and computational frameworks, highlighting the potential of bacterial systems in advanced biocomputing applications

across various fields, including therapeutics.

### Analysis 2 - Impact of Human GB Structure on Molecular Output

This analysis involves manually adjusting the sizes of bacterial populations within the virtual Gut Bacteriome and monitoring the corresponding changes in metabolite production.

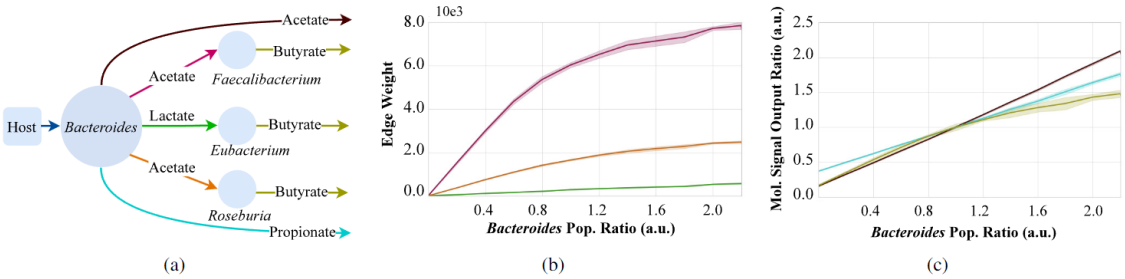



Figure 3.12: Responses of SCFA production to different *Bacteroides* population sizes: (a) subgraph illustrating *Bacteroides* population interactions, (b) dynamics of edge weights, and (c) SCFA production levels.

Fig. 3.12 elucidates how varying the *Bacteroides* population size affects SCFA production as an output in the human GB. Fig. 3.12a shows the associated SCFA subnetwork, while Fig. 3.12b and Fig. 3.12c follows the same consistency in color coding as Fig. 3.12a. While keeping other inputs and population sizes constant, the *Bacteroides* population is adjusted from none to 2.2 times its standard size. Fig. 3.12b illustrates the effect of these changes on connections leading from *Bacteroides* to other populations, *Faecalibacterium*, *Eubacterium* and *Roseburia*, mediated by acetate and lactate. Fig. 3.12c highlights the direct correlation between the *Bacteroides* size and SCFA production levels, with acetate and propionate production showing linear increases alongside *Bacteroides* growth. Furthermore, the analysis indicates a plateau in butyrate production as *Bacteroides* exceed 80% of their standard population size, suggesting a limit to the benefit of increasing *Bacteroides* numbers on butyrate output. This analysis under **RA 1** reveals that the structure of the human GB significantly affects its output underscoring the dynamics of the computing

process of a bacterial ecosystem. These results show the role of communication of bacterial computing (**RQ 1**).

This experiment is repeated for all the populations and extracted the relationship between the human GB composition and the output molecular signal production.

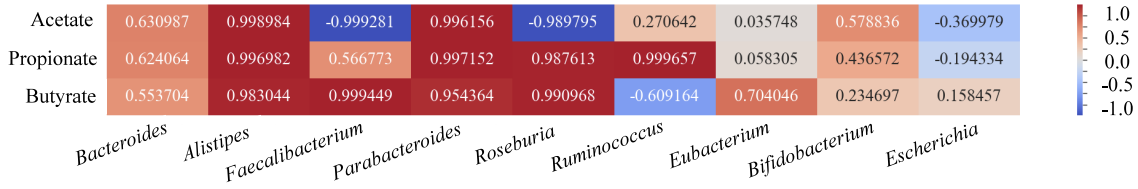



Figure 3.13: Pearson correlation heatmap showing the impact of nine bacterial populations on the production of three output molecular signals.

Figure 3.13 illustrates the correlation between each bacterial population and SCFA abundance in the gut. While *Bacteroides* are the major producers of SCFAs, they show a weaker correlation with SCFA levels compared to other producers like *Alistipes* and *Parabacteroides*. This suggests that decreased glucose consumption by *Bacteroides* may increase other bacterial populations, thereby enhancing SCFA production. However, despite this boost from other bacteria, overall SCFA production remains low in the absence of *Bacteroides*. The heatmap also reveals a strong negative correlation between *Faecalibacterium* and *Roseburia* with acetate, as they consume this SCFA. Interestingly, it shows *Escherichia* switching from producing to consuming high concentrations of acetate. Similarly, *Ruminococcus* switches from consuming fucose to glucose when fucose is scarce, leading to decreased production of intermediate metabolites and reduced butyrate production. Further information on the above analysis can be found in Chapter 5.

From a computing perspective, these findings highlight the intricate and dynamic nature of bacterial metabolic networks, emphasizing the potential for leveraging such biological systems for computational tasks. Mapping of biological data into computational frameworks can drive advancements in synthetic biology and bioinformatics, offering innovative solutions to complex computational problems.

### 3.1.4 Bacterial Communication Network Reliability

In order to gain deeper insights into the MC of bacteria within-population dynamics resulting from internal computing mechanisms, it is crucial to investigate bacterial communication networks. Bacterial ecosystems consist of complex Bacterial Molecular Communication Networks (BMCN) with redundant pathways that transmit signals through a shared medium, leading to the accumulation of diverse molecules. BMCN are characterized by their cascading networks with parallel paths, where multiple network segments perform similar functions. This redundancy can enhance the ecosystem's resilience, as one segment's failure can be compensated by others, maintaining network performance.

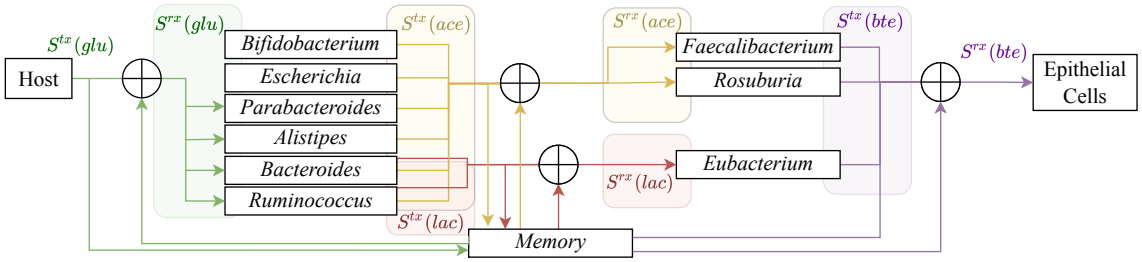



Figure 3.14: The bacterial cascading system for SCFA production that includes an environmental memory component. Transmitted signals for glucose, acetate, and lactate are denoted as $S^{tx}(glu)$, $S^{tx}(ace)$, and $S^{tx}(lac)$, respectively. Correspondingly, received signals for glucose, acetate, lactate, and butyrate are represented as $S^{rx}(glu)$, $S^{rx}(ace)$, $S^{rx}(lac)$, and $S^{rx}(bte)$. These signals are influenced by noise originating from the memory component.

This analysis employs information and MC theories to explore the influence of Cooperative Amplification (CA) on InterSymbol Interference (ISI) within BMCN. Additionally, this analysis examines how information traverses these networks, focusing on a cascading, parallel structure where various molecules serve as signals. Here, butyrate is considered the end product within a cascading BMCN featuring nine bacterial genera associated with the SCFA network as shown in Fig. 3.14. These genera are chosen based on their metabolic functionalities and ancestral origins. The butyrate production pathway begins with glucose entering the human GB, and is structured into four layers: 1) host cells acting as glucose transmitters,

2) acetate/lactate-producing bacteria that consume glucose, 3) butyrate-producing bacteria that utilize acetate/lactate, and 4) epithelial cells that receive butyrate. *Bacteroides*, *Alistipes*, *Parabacteroides*, *Bifidobacterium*, *Ruminococcus*, and *Escherichia* genera are responsible for converting glucose into acetate, while *Faecalibacterium* and *Roseburia* genera convert acetate into butyrate, and *Eubacterium* utilizes lactate for butyrate production. Epithelial cells serve as the final receivers of butyrate. In this analysis Mutual Information (MI) is employed to gauge the flow of information through the network, providing insights into how CA influences communication efficiency and reliability within the system.

Employing the same simulation introduced previously, the experiment is initialized with the relative abundance of each genus, derived from species-level RA data from MicrobiomeDB [102] and Disbiome [99] databases. This approach was also applied to Disbiome database samples to outline the average compositions associated with autism and Parkinson's disease. The simulation introduces twenty distinct single-pulse glucose inputs, varying from 0.259 µmol/$m^3s$ to 2.594 µmol/$m^3s$, across three human GB compositions: control, autism, and Parkinson's. These input amplitudes are chosen to match the bacterial cell counts in the simulator, ensuring observable information flow changes, whereas the inputs outside this range did not yield significant results. The simulation duration was set to 500 minutes, accommodating the dynamics of the strongest (glucose) and weakest (lactate) signals, ensuring all vital signal changes occurred within this timeframe. To enhance result accuracy against system stochasticity, each scenario was repeated 50 times, with molecular consumption and production data for each bacterial population collected at every time step.

In response to the glucose pulse-like inputs of varying concentrations into the average network, subsequent SCFA signal levels are measured. The resulting cascading signal flow is depicted in Fig. 3.15, demonstrating how these inputs propagate through the system.

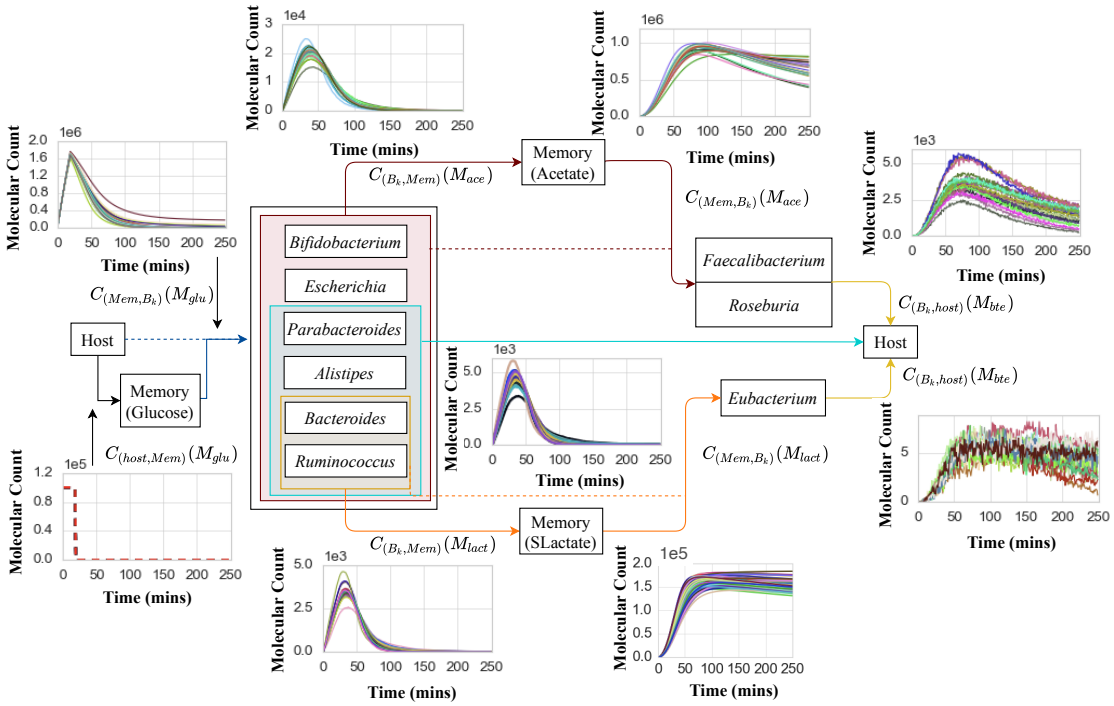



Figure 3.15: Representation of the simplified MC network with plots illustrating link behaviors for different compositional changes. Please note that *glu*, *ace*, *lact* and *bte* stands for glucose, acetate, lactate and butyrate respectively

Initially, the input signal entropy and conditional entropies of the cascading BMCN for each scenario are calculated. The analysis of conditional entropy is performed in three stages for the nine bacterial genera shown in Fig. 3.14. The first stage involves calculating the conditional entropy for *Bacteroides*, *Alistipes*, *Parabacteroides*, *Bifidobacterium*, *Ruminococcus*, and *Escherichia*, focusing on their response to glucose inputs. The second stage assesses the conditional entropy for *Faecalibacterium* and *Roseburia* based on acetate reception, and the third stage evaluates *Eubacterium*'s response to lactate and epithelial cells' butyrate reception. These steps are crucial for assessing the MI through the full system. These MI results are presented in Fig. 3.16.

The findings, detailed across Figures 3.16a, 3.16b, and 3.16c, reveal variations in information flow through the network corresponding to compositional changes. The estimated MIs show significant differences in the first layer across the three compositions, with lesser variations in the second layer, and epithelial cells' MIs

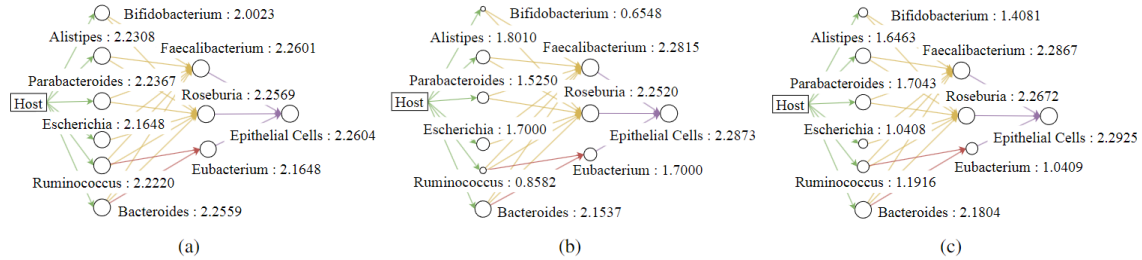



Figure 3.16: Estimated MI values for each bacterial population and epithelial cells across (a) control human GB, (b) Parkinson's GB, and (c) autistic GB. The sizes of the nodes reflect the MI values, measured in bits.

displaying similar values in all scenarios. Notably, *Bacteroides* exhibits the highest MI in all setups, indicating its dominance. The study highlights that, in systems with redundant paths and cooperative transceivers, an increase in information flow can be observed.

Subsequently, these results underscore the significance of CA in ensuring reliable information flow through BMCN by exploiting another aspect of **RQ 1**. Further description and additional results can be found in Chapter 6.

## 3.2 Introducing Gene Regulatory Neural Networks

The population-wide behavior of bacteria stems from the decision-making processes of individual cells. Therefore, this section explores the internal computing mechanisms within each cell that drive their cellular functions and contribute to the overall dynamics of the bacterial population. By understanding these mechanisms, we can better comprehend how individual cellular decisions aggregate to influence population-level behaviors and enhance our ability to utilize bacterial systems for biocomputing applications.

Despite lacking neural structures for computation, bacteria's gene regulatory network (GRN) empowers them to strategize and adapt across varying conditions. This adaptability not only results in the production of molecules that influence

other cells, fostering complex social interactions and motility towards more favorable environments, but also in significant physiological state changes. These complex behaviors provide sufficient evidence for an existence of natural computing power inside bacterial GRN. Investigating the inherent computing capabilities of bacteria under **RQ 2**, enhances our understanding of their behavior, paving the way for innovative approaches in programming cells for novel treatments and laying the groundwork for future bio-computing systems.

### 3.2.1 From Gene Regulatory Networks to Gene Regulatory Neural Networks

The literature explains the GRN guiding bacterial decision-making encompasses a structure reminiscent of a hidden neural network [104, 105]. Typically, the GRN only offers information about the presence of interactions and their types (activation or repression). However, investigation of transcriptomic data reveals a 'weight' behaviours that determines the influence magnitude of one gene on another. This behaviour emerges from the binding affinity of transcription factors (TFs), and elements like thermoregulators and enhancers/silencers [106, 107]. The presence of activator TFs and sigma factors leads to enhanced gene expression [108, 109], akin to higher positive weights in neural networks. Conversely, repressors and anti-sigma factors, which dampen gene expression, are likened to larger negative weights. The non-linear aspect of GRNs is similar to the rectified linear unit (ReLU) activation function in NNs, especially since gene expression cannot fall below zero despite potentially negative weighted sums. This research task introduces a weight extraction mechanism that quantifies the gene-gene interaction in the form of weights. This weight extraction transforms the GRN into a pre-trained neural network. Hence, this novel concept is introduced as **Gene Regulatory Neural Networks** (GRNNs) and addresses **RA 2**, opening avenues to explore genetic regulation through the lens of neural network theory.

The weight extraction process starts by constructing the GRN as a graph depicting gene-gene interactions, where the expression of each gene is predominantly influenced by TF signals from adjacent genes or occasionally the same gene.

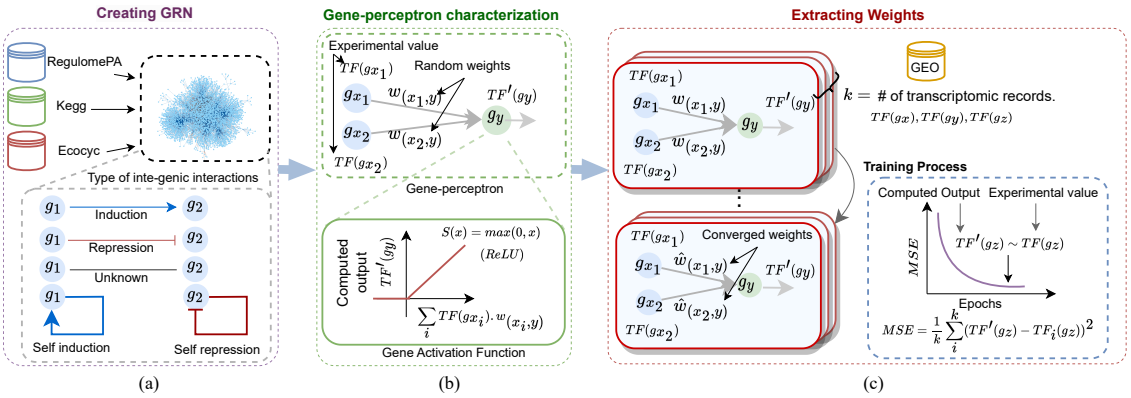



Figure 3.17: Depiction of the GRNN extraction process, where a) involves constructing a GRN structure showcasing diverse gene interactions sourced from databases, b) dissects the GRN into gene-perceptrons utilizing ReLU activation functions, and c) describes the weight extraction for gene-perceptrons, fine-tuning edge weights to minimize the Mean Squared Error (MSE) between calculated ($TF'(g_z)$) and experimental ($TF(g_z)$) gene expression levels.

Elaborating on this, Fig. 3.17a outlines the construction of the GRN graph, incorporating five types of regulatory influences based on data from RegulonDB [110] (specific to *P. aeruginosa*), KEGG [111, 112, 113], and Ecocyc [114] databases. Subsequently, this network is broken down into sub-graphs, each centered around a target gene and its regulatory genes, mirroring a single-layer perceptron's structure with ReLU activation, as demonstrated in Fig. 3.17b, thereby designating the target gene as the "gene perceptron."

Although the exact biophysical definition of weights for these gene perceptrons has remained uncharted, the proposed approach extracts the weights similar to a single-layer perceptron's training mechanism. This involves adjusting initially random weights based on the MSE between calculated and observed gene expression data to minimize the discrepancy. The optimal weights, indicative of the regulatory impact on the target gene, are identified at the point of least MSE, detailed in Fig. 3.17c. This process essentially converts the GRN into a pre-trained random

structured neural network, hence a GRNN.

The extracted weight matrix is denoted as,

$$\mathbf{W} = \begin{array}{c} \\ g_1 \\ g_2 \\ \vdots \\ g_P \end{array} \begin{array}{c} \begin{array}{cccc} g_1 & g_2 & \cdots & g_P \end{array} \\ \begin{pmatrix} w_{(1,1)} & w_{(1,2)} & \cdots & w_{(1,P)} \\ w_{(2,1)} & w_{(2,2)} & \cdots & w_{(2,P)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(P,1)} & w_{(P,2)} & \cdots & w_{(P,P)} \end{pmatrix} \end{array}, \tag{3.1}$$

where $w_{(i,j)}$ is the weight of the interaction between $i^{th}$ and $j^{th}$ gene with $i : j = \{1, 2, .., P\}$. The $w_{(i,j)}$ is the weight in the case of self-regulation.

Next, the computational output is modeled as, $\mathbf{O}^{(t+1)}$ at $t + 1$ using weight $\mathbf{W}$ as,

$$\mathbf{O}^{(t+1)} = max(\mathbf{W} \cdot (\mathbf{I}^{(t)} + \widetilde{\mathbf{N}}) + \mathbf{B}), \tag{3.2}$$

where $\mathbf{I}^t$ is the input matrix, while $\mathbf{B}$ is the bias matrix and $\widetilde{\mathbf{N}}$ is the added Gaussian noise ($\widetilde{\mathbf{N}} = N(0, 0.1)$) extracted based on the iterative experiments [115] (GEO accession number GSE215300). For the next time step, the input matrix $\mathbf{I}^{t+1} = \mathbf{O}^{t+1}$ and $\mathbf{O}^{t+2}$ is computed as,

$$\mathbf{O}^{(t+2)} = max(\mathbf{W} \cdot (\mathbf{I}^{(t+1)} + \widetilde{\mathbf{N}}) + \mathbf{B}). \tag{3.3}$$

### 3.2.2 *Pseudomonas aeruginosa* GRNN

First, the weights extraction method explained in Section 3.2.1 is applied to the *P. aeruginosa* GRN which encompasses 2851 genes and 4903 interaction links. The transcriptomic data utilized for this weight extraction process is obtained from the GEO database [116]. Following extensive preprocessing of this data, 80% is allocated for weight extraction of the gene perceptrons, with the remaining portion reserved for validation purposes. The extraction process is started by initializing the *P. aerug-*

*inosa* single-layer gene perceptrons with random weights. For the training process, the learning rate and the number of epochs are set to $10^{-6}$ and $10^{9}$, respectively.

The accuracy of the extracted GRNN weights is then evaluated by using the remaining 20% of the dataset. The results are depicted in Fig. 3.18, revealing that most data points closely align with the 45-degree line, suggesting a high level of prediction accuracy. These results were published as presented in Chapter 7.

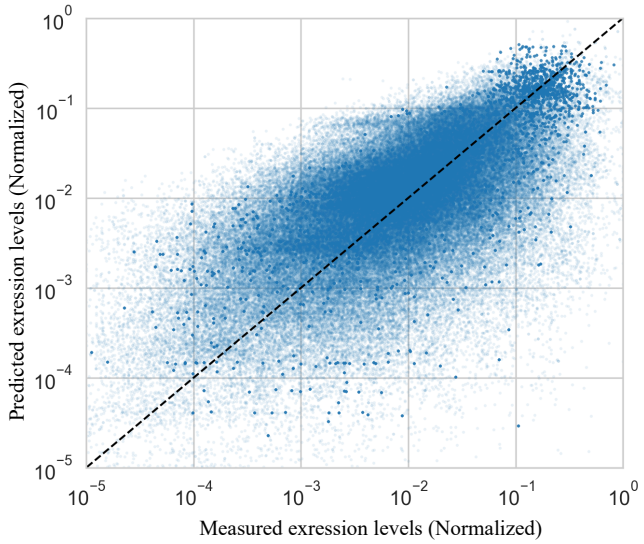



Figure 3.18: Comparison of measured expression levels for 2,851 genes across 217 transcription records against gene expression values calculated by the fully extracted GRNN.

### 3.2.3 *E. Coli* GRNN

Utilizing the same methodology, the GRNN of *E. coli* K-12 strain CSH50 is constructed. This process begins with the acquisition of the GRN dataset in [117], which is categorized into various interaction types, including TF to gene, TF to operon, TF to Transcription Units (TU), TF to TF, sigma factor to gene, SF to TU, and small RNA to gene. By amalgamating these interactions, a comprehensive GRN for *E. coli* is constructed as a directed graph, comprising 3,175 gene nodes and 9,678 interaction edges.

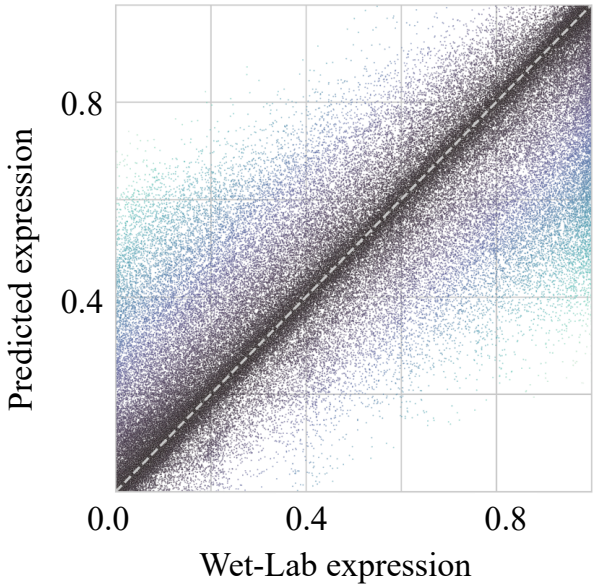



Figure 3.19: Comparison of measured expression levels for 3175 genes across 43 transcription records against gene expression values calculated by the fully extracted *E.coli* GRNN.

The next phase involves the extraction of the weights and biases, utilizing temporal transcriptomic data from [115] (GEO accession number GSE65244) and integration back into the GRN, effectuating its transformation into GRNN.

The accuracy of the *E. coli* GRNN is assessed in Fig. 3.19 by comparing predicted gene expression levels against those measured in wet-lab experiments. In Fig. 3.19, the dashed line inclined at 45° serves as a benchmark, indicating perfect alignment between predicted and experimental values. The close proximity of most data points to this line suggests a high level of agreement between the model's outputs and empirical observations. These results were published as presented in Chapter 7 and 9.

Three critical insights emerge from these observations:

1. The results affirm the feasibility of quantifying complex gene-gene interactions through computational means, specifically in the form of weights and biases within the model.

2. The cross-genome applicability of the introduced weight extraction model as proven by the accuracies of extracted *P. aeruginosa* and *E. coli* GRNNs.

3. The performance of the GRNN model in mimicking biological gene regulatory mechanisms is substantiated, underscoring the model's potential as a valuable tool for biological research and its compatibility across different bacterial genomes.

## 3.3 GRNN Computing

This section, aligning with **RA 2**, explores GRNN computing with a focus on its capabilities for generalized computing targeting **RQ 2**. It begins by assessing the structural and algorithmic complexities of GRNN, which is crucial for determining its suitability for general computing applications. Further, this section evaluates the energy consumption of GRNN.

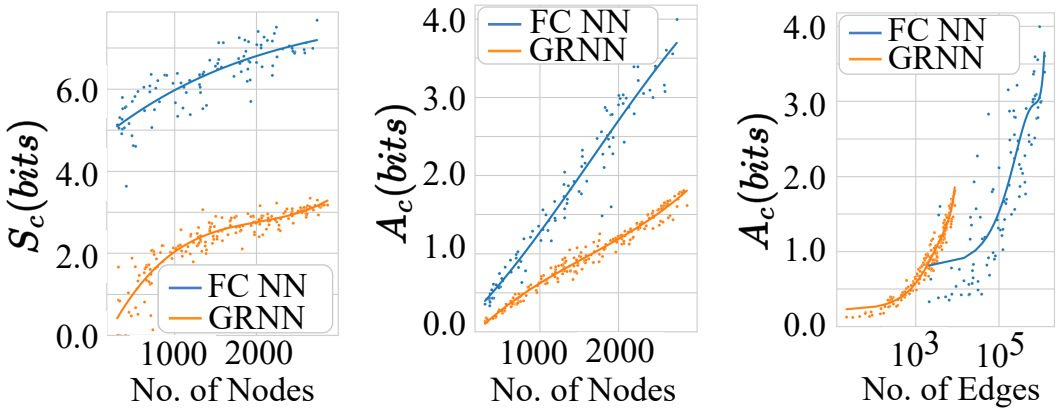



Figure 3.20: A comparison of structural ($S_c$) and algorithmic ($A_c$) complexity between Fully Connected Neural Networks (FCNNs) and GRNNs is presented. a) and b) examine how $S_c$ and $A_c$ change with the number of nodes in both network types, whereas c) explores $A_c$ in relation to the number of edges.

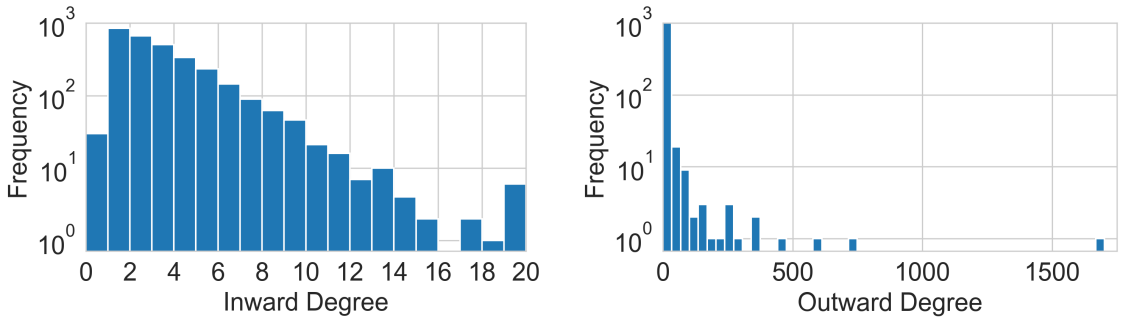
### 3.3.1 Structural and Algorithmic Complexities

Structural complexity relates to the network's architecture, including the number of nodes/edges and the overall topology, which significantly influences the network's computational power. Algorithmic complexity, on the other hand, refers to the computational demands of the training and inference processes in NNs. Thus, this study undertakes a comparative analysis of the structural and algorithmic complexities of GRNNs against conventional fully connected NNs. The calculation of the structural and algorithmic complexities are detailed in Chapter 9.

Fig. 3.20a demonstrates a comparison between the structural complexities as a function of the node count in fully connected NN and GRNNs. The analysis reveals that GRNNs, characterized by their random structure, exhibit lower structural complexity due to the presence of power-law properties, in contrast to fully connected NNs. Furthermore, an investigation into algorithmic complexity, as depicted in Figure 3.20b, indicates that GRNNs display reduced complexity relative to fully connected NNs. This reduction is attributed to a decreased number of edges for a comparable node count. However, within the edge range of 2000 to 10000, GRNNs exhibit a higher algorithmic complexity than their fully connected counterparts, as illustrated in Figure 3.20c. This observation suggests that specific configurations of GRNNs can achieve complex computational tasks while simultaneously enhancing interpretability over fully connected NNs.

Further, inward and outward degree distributions of GRNNs, exhibit a few crucial characteristics portraying them as suitable candidates for general computing tasks. This argument is supported by using the *E. coli* GRNN as a use case. This GRNN contains approximately 68.45% of gene-perceptrons that receive input from more than one inward edge, facilitating the processing of multiple inputs simultaneously (Figure 3.21a).

Furthermore, the outward edge distribution illustrated in Figure 3.21b demonstrates the presence of hub gene-perceptrons capable of influencing up to 92.12%




(a) (b)

Figure 3.21: Degree distribution of the *E. coli* GRNN is illustrated, with a) displaying the frequency of inward degrees and b) showcasing the frequency of outward degrees.

of terminal nodes. This structure enables the base-GRNN to effectively handle a variety of computational problems. For example, the gene b3067 is connected by 1703 outward edges, with 91% leading to terminal nodes. Activation of this gene-perceptron triggers a broad spectrum of expression levels in these terminal nodes, thus showcasing the system's computational diversity. Consequently, this power-law distribution exemplifies the base-GRNN's potential as a comprehensive source of diverse, pre-trained sub-GRNNs.

Further, this research task focuses on exploring the diversity of GRNN sub-networks by analyzing the configurations of input, intermediate hidden layers, and output gene-perceptrons. Starting with sets of 100 input gene-perceptrons and progressively increasing to 500, the research tracks the connections and depth (up to 10 layers) of these subnetworks, iterating the process 100 times for each set size to average the number of gene-perceptrons per layer. This methodical exploration reveals that with an initial setup of 100 input nodes, the network can expand to approximately 500 output nodes by the sixth layer, offering a vast array of combinations (up to $8.9 \times 10^{26}$) for configuring output nodes tailored to specific applications as shown in Fig. 3.22. This grows exponentially with more input candidates, reaching up to $5.9 \times 10^{297}$ combinations, illustrating the GRNN's capability to adapt and provide highly customizable solutions for diverse applications. Further expanding the input

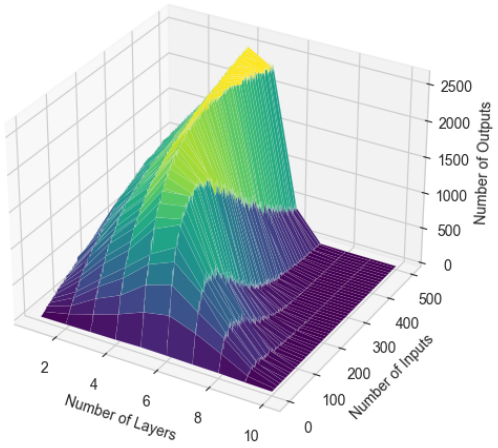



Figure 3.22: Illustrations of the number of output node variations given the number of input nodes and the depth of the GRNN subnetwork.

layer to 500 nodes increases the output layer to about 2,500 nodes, thereby broadening the adaptation possibilities (up to $9.3 \times 10^{33}$ configurations for 10 outputs) and underscoring the significant impact of input layer size on the network's versatility and applicability across various domains.

### 3.3.2 Energy Profiling of GRNN

Aligning with **RA 2**, this section focuses on the energy consumption aspect of the GRNNs. This comparative analysis of energy consumption across various computing platforms strictly focuses only on the energy expended for computational activities, deliberately excluding the energy requirements for auxiliary or 'housekeeping' functions.

In this investigation, the energy efficiency of GRNN is evaluated against four other computing processors, across 200 model sizes, each differentiated by its algorithmic complexity. This analysis involves adjusting the number of nodes for each model size on both traditional von Neumann architectures and neuromorphic computing platforms. The outcomes of this comparative study are presented in Fig. 3.23.

A striking observation from this analysis is the GRNN's remarkably low power

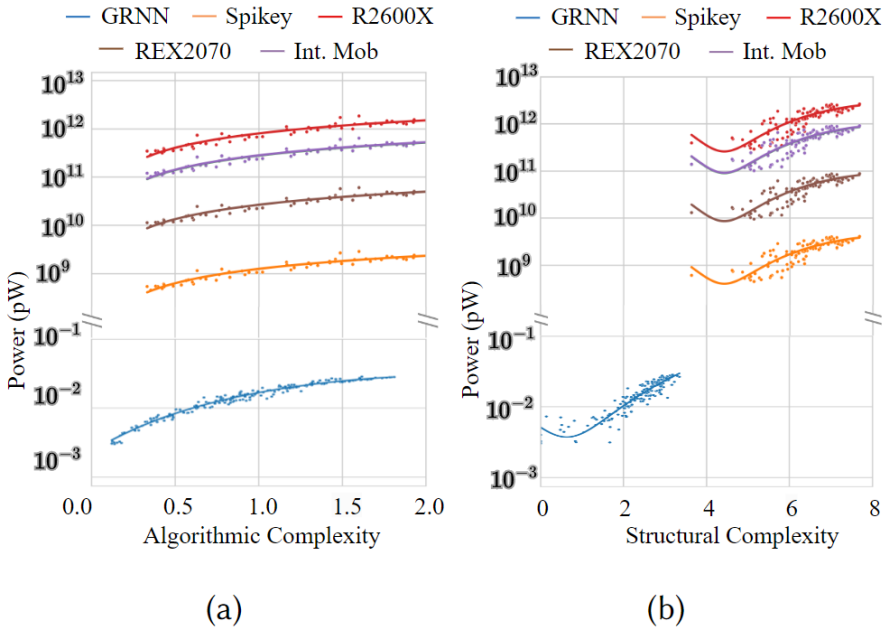



Figure 3.23: Comparison of power consumption between GRNN, von Neumann, and neuromorphic computing systems, focusing on a) algorithmic complexity and b) structural complexity.

consumption. As depicted in Fig. 3.23a, the peak energy usage of GRNN does not exceed 0.05 picowatts (pW), even when subjected to the highest algorithmic complexity challenges. This is in stark contrast to the energy consumption observed in other computing platforms, where power requirements span from $10^9$ pW to $10^{12}$ pW for models of comparable neuron counts. This contrast underscores the GRNN's superior energy efficiency, particularly in high-complexity computational tasks.

Recognizing the intricate computing architecture akin to a wet-neuromorphic system within bacterial cells, which will be further explored in the subsequent section, this study views them as natural computing powerhouses. Here, the GRN functions as the core computing mechanism.

These results exhibit that the GRNN-based computing can cater generalizability while maintaining high energy efficiency. Therefore, bacterial cells with GRNN-base computing capabilities can be placed as a novel wet-neuromorphic computing system due to its physical architecture, computing diversity and energy efficiency.

## 3.4 MC Model for Computing

Previous chapters investigated the MC of bacterial population behaviors and their internal computing mechanisms separately. However, this chapter aims to understand the interplay between MC and gene regulation-based computing at the single-cell level. This integrated approach will provide a more comprehensive understanding of how these processes work together to drive bacterial behavior and functionality.

### 3.4.1 MC and GRNN Integration

Bacteria's remarkable ability to sense molecular signals from other microbes and environmental shifts, such as temperature or pH changes, enables dynamic regulation of their gene expression and protein production prolonging their survivability. This intricate process not only enhances bacterial survivability by driving adaptive behaviors but also represents a form of chemical-based computing as abstracted in Fig. 3.24.

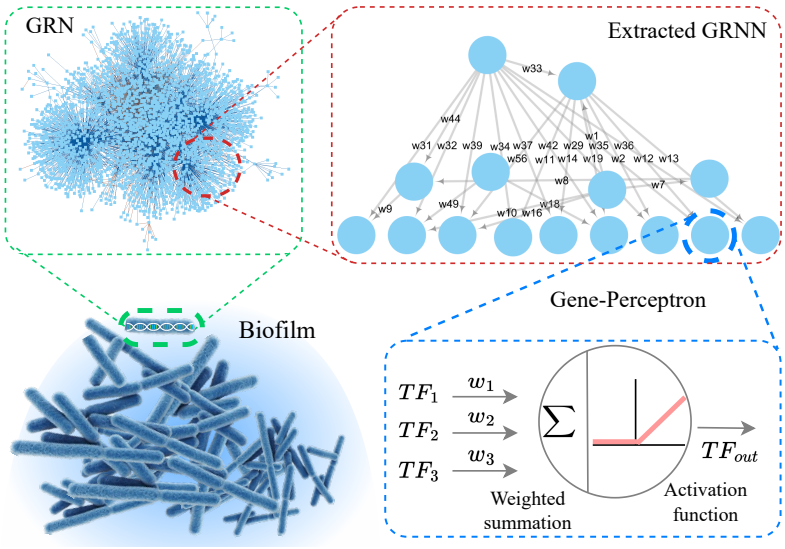
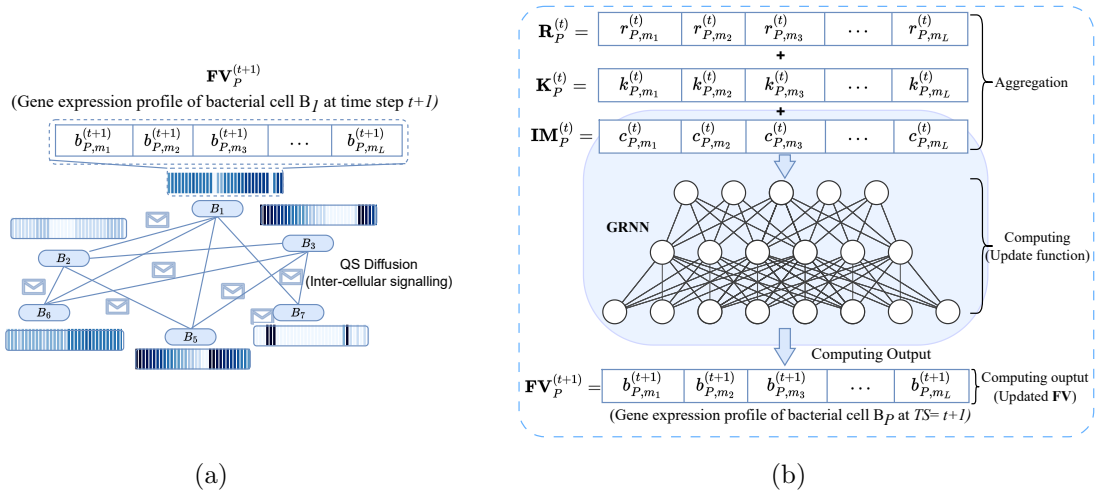



Figure 3.24: Depiction of a biofilm and the process of extracting GRNN from bacterial cells within it.

In Section 3.1 aligned with **RA 1**, the thesis showed the influence of intercellular

communication on population-level computing dynamics. In contrast, this section explores the influence of intercellular communication on the dynamics of GRNN aligning with **RA 3** by employing graph neural networks to mimic cell-cell communication, capturing elements like the random spatial distribution of cells, molecular diffusion dynamics from cell to cell, and the modulation of cellular responses to molecular signals.




(a) (b)

Figure 3.25: Illustration of a) the graph neural network model of the MC in bacterial population and b) the mechanism of the outputs from one GRNN is conveyed to another GRNN as molecular messages.

As shown in Fig. 3.25a, in the construction of the bacterial ecosystem as a graph network, each node represents a cell $(B_1, B_2 ... B_7)$. The corresponding feature vector $(\mathbf{FV}_P^{(t)})$ contains the GRNN gene expression profile of the cell $B_P$ at time $t$ as shown in Fig. 3.25b. The edges in this figure symbolize diffusion-based communication of various molecular species $(m_1, m_2, ... m_Q)$ between cells are treated according to a message-passing protocol of the graph neural networks. Further in Fig. 3.25b, the incoming signal vector, nutrient concentrations at the location of the cell and the accumulated intra-cellular molecular concentrations are denoted by $\mathbf{R}_P^{(t)}$, $\mathbf{K}_P^{(t)}$ and $\mathbf{IM}_P^{(t)}$ respectively.

A cell's receipt of molecular signals is represented through an aggregation func-

tion, while the GRNN embedded in each node acts as the update function, processing aggregated signals to alter gene expression patterns and, consequently, the cell's feature vector as depicted in Fig. 3.25b.

### 3.4.2 Use Case model

In this research task, *P. aeruginosa* single species biofilm is utilized as a use case to explore the impact of cell-cell communication on GRNN computing based on the methodologies explained in Section 3.2 and 3.4. This focus is primarily due to *P. aeruginosa*'s extensive research background, driven by its association with significant health issues such as pneumonia, blood infections, and infected wounds. Notably, *P. aeruginosa* produces pyocyanin (PYO), a toxin that impairs human cell functions, further underscoring its relevance in medical research and public health.

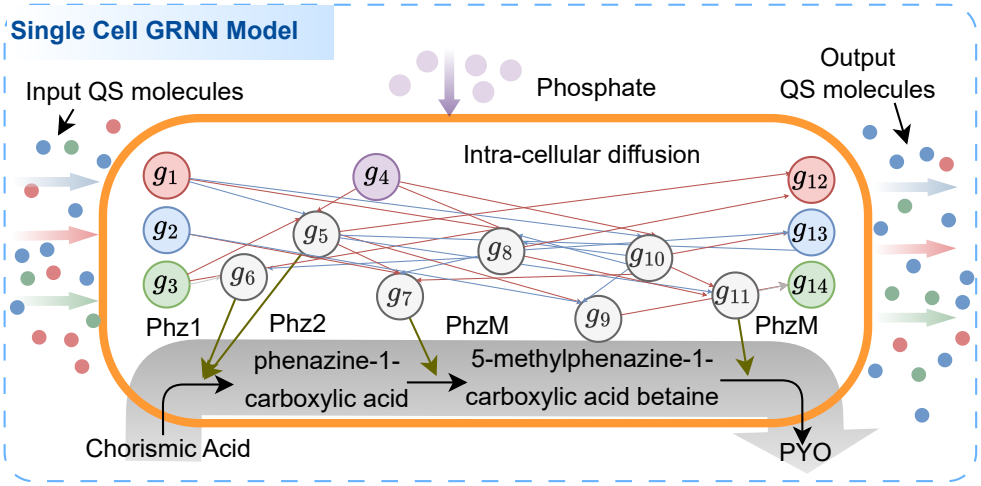



Figure 3.26: Depiction of the computational process for incoming cell-cell communication molecules and the transformation of chorismic acid into PYO, mediated by GRNN outputs in reaction to phosphate input.

Fig. 3.26 illustrates the impact of extracellular nutrient signals and QS signals exert a wide range of regulatory effects on bacterial gene expression associated with PYO production.

Therefore, as the next step, the sub-GRNN is extracted from the full *P. aeruginosa* GRNN using shortest path analysis, focusing on genes associated with PYO

production, QS-related genes and the two-component system (TCS) PhoR-PhoB, which regulates genes activated by phosphate intake (phZ1, phz2, phzS, and phzM) that are crucial for enzyme production essential for PYO synthesis.

This model is embedded with another metabolic interaction layer which is essential for molecular uptake but can be kept as a separate layer as shown in Fig. 3.27.

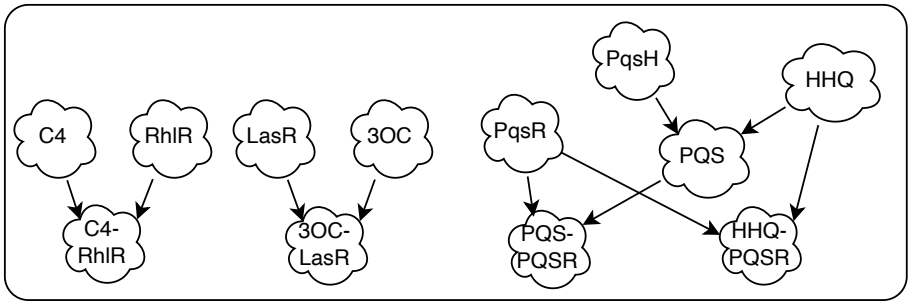



Figure 3.27: Depictions of intracellular metabolite interactions, highlighting how QS molecules interact with response regulators to form complexes.

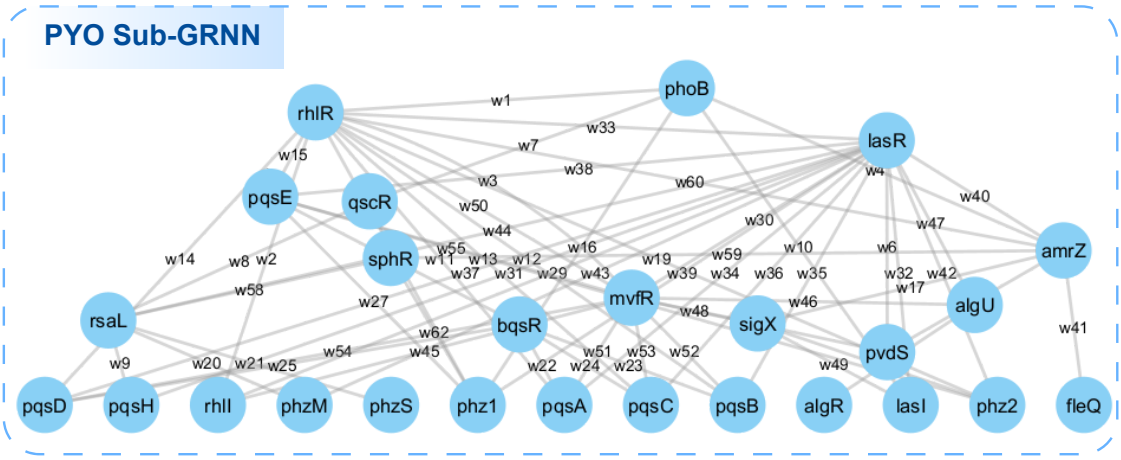



Figure 3.28: Illustration of the PYO production sub-GRNN before the weight extraction process.

In this context, RhlR, a key transcriptional regulator in *P. aeruginosa*, binds to its cognate inducer C4-HSL, serving as an input to the GRNN. Similarly, the LasR and PqsR transcriptional regulators, when bound to 3-oxo$-C_{12}-$HSL (3OC), PQS, and HHQ, respectively, also provide inputs to the GRNN. Concurrently, environmental chorismic acid ($C_{10}H_{10}O_6$) is transformed by *P. aeruginosa* through a

series of reactions mediated by the GRNN products phz1, phz2, phzS, and phzM. This process begins with the conversion of $C_{10}H_{10}O_6$ into phenazine−1-carboxylic acid via enzymes from Phz1 and Phz2 genes, followed by its conversion into 5-methylphenazine-1-carboxylate, and ultimately, PYO, through the actions of PhzM and PhzS. Thus, GRNN computing plays a crucial role in converting $C_{10}H_{10}O_6$ into PYO, as depicted in Fig. 3.28.

### 3.4.3 Mutagenesis Analysis to Reveal the MC Impact on GRNN Computing

Next, the mutagenesis analysis is conducted by altering the GRNN structure to observe the resulting changes in gene expression and PYO production. The analysis is carried out under two phosphate conditions: high phosphate (HP) and low phosphate (LP), to assess the impact of phosphate levels as well as network alterations due to mutations on the GRNN's computing behavior. Therefore, this subsection focuses on **RA 3** aligning with **RQ 1** and **RQ 2**.

Eight simulation experiments are designed as follows: 1) wild-type bacteria without mutations (WD) under LP, 2) *lasR* mutant ($\Delta lasR$) under LP, 3) *phoB* mutant ($\Delta phoB$) under LP, 4) *lasR* and *PhoB* double mutant ($\Delta lasR\Delta phoB$) under LP, 5) WD under HP, 6) $\Delta lasR$ under HP, 7) $\Delta phoB$ under HP, and 8) $\Delta lasR$ $\Delta phoB$ under HP. In this setup, the WD condition utilizes the entire PYO sub-GRNN. The $\Delta lasR$ mutation involves removing the *lasR* node, the $\Delta phoB$ mutation excludes the *PhoB* node, and the double mutant ($\Delta lasR\Delta phoB$) removes both the *lasR* and *phoB* genes respectively, as depicted in the GRNNs of Fig. 3.29. These mutations induce structural changes in the GRNN, affecting computational outputs as evidenced by variations in gene expression and PYO production levels.

In the mutagenesis studies described, we compare GRNN-computed PYO production values against wet-lab data from [118], observing the molecular output behaviors of different GRNN structures in *P. aeruginosa* biofilms. The findings reveal

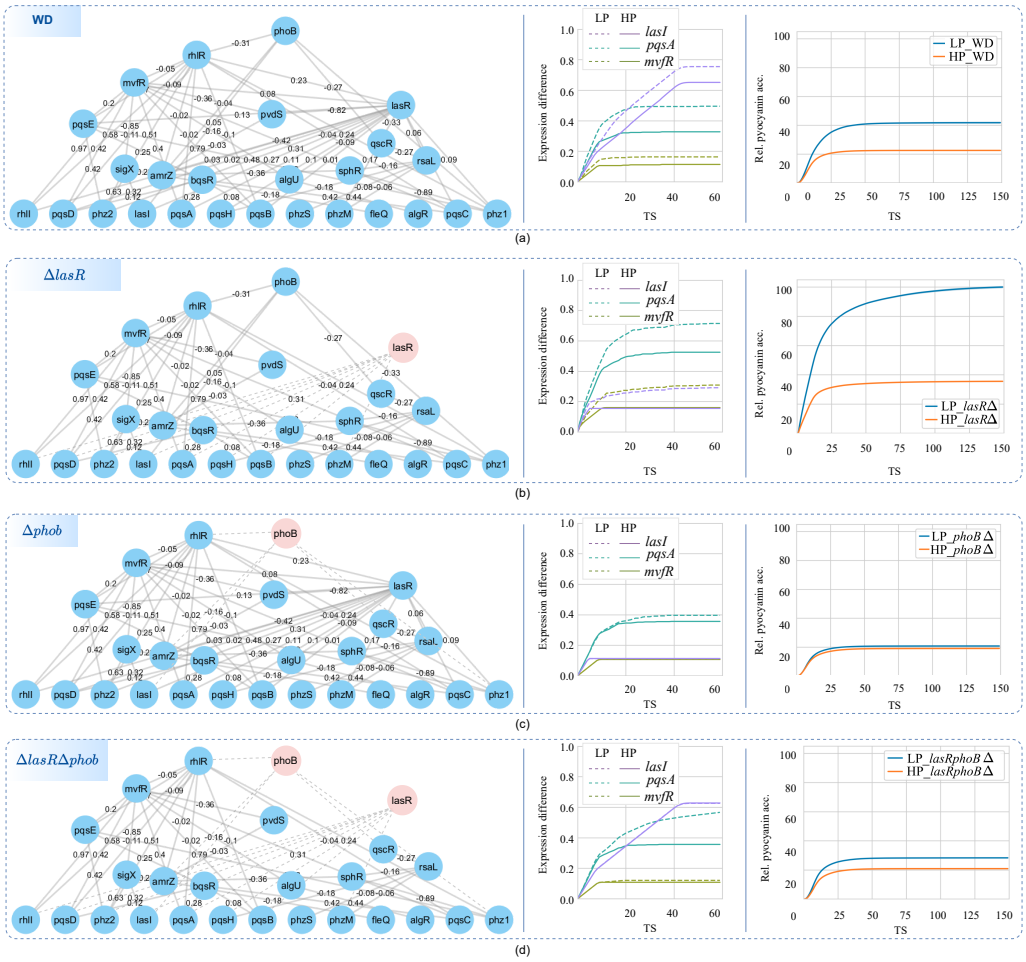



Figure 3.29: Analysis of mutagenesis to explore the effects of structural variations in GRNN on PYO production. This study illustrates gene expression changes and subsequent PYO outputs under low phosphate (LP) and high phosphate (HP) conditions for four different GRNN structures: (a) wild type (WD), (b) $\Delta lasR$, (c) $\Delta phoB$, and (d) $\Delta lasR\Delta phoB$. Genes within red circles are omitted from the GRNN across the various structural models to highlight the network's structural alterations, thereby demonstrating the computational shifts in PYO production.

higher PYO levels in low phosphate (LP) conditions across all cases, with the $\Delta lasR$ mutation showing the most significant increase, and the $\Delta phoB$ mutation the least. This pattern primarily results from the repressive effect of the *phoB* gene on key genes, where higher phosphate levels lead to increased *phoB* expression, consequently repressing other genes crucial for PYO production.

The *lasR* mutation amplifies this effect, with the *mvfR* gene identified as critical in this context. The *mvfR* gene, which enhances the expression of seven other genes, results in increased PYO production, especially in LP conditions due to di-

minished *rhlR* expression. This scenario highlights the GRNN's enhanced sensitivity to phosphate levels through the *lasR* mutation.

Conversely, the smallest difference in PYO production between LP and HP conditions in the $\Delta phoB$ case is attributed to the GRNN's reduced phosphate sensitivity due to the absence of the *phoB* gene. The combined $\Delta lasR\Delta phoB$ mutation shows a greater difference between LP and HP PYO production compared to $\Delta phoB$ alone, illustrating the compound effects of both mutations on PYO production. This study underscores the variability in GRNN's response to environmental conditions and the potential for designing application-specific GRNNs by manipulating gene expression and interactions.

Furthermore, these results elucidate the GRNN computing variability in response to the network structure changes. Additionally, more information and results regarding this analysis were published as presented in Chapter 7.

### 3.4.4 Inferring Cluster-scale Collective Perceptrons

Interestingly, the output patterns of individual cells' GRNNs within the biofilm collectively demonstrate that bacteria generate a series of non-linear output functions over space and time through cell-cell communication. This process is explored more using the same graph neural network-based *P. aeruginosa* PYO model. This section investigate the non-linear output properties of various biofilm regions and time points, resulting from clusters of cells each equipped with a GRNN. The investigation focuses on understanding these dynamics through the lens of a sigmoid activation function $S(x)$,

$$S(x) = \frac{L}{1 + e^{-(kx - x_0)}}. \tag{3.4}$$

where the parameters $L$, $k$, and $x_0$ in the model govern the maximum value, steepness, and horizontal shift of the curve, respectively, as illustrated in Fig. 3.30.

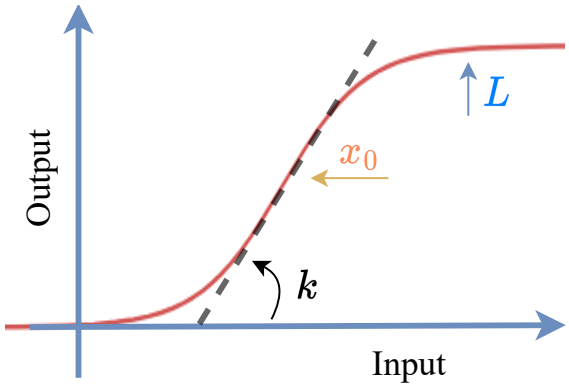



Figure 3.30: Depiction of the parameters $L$, $k$, and $x_0$, which respectively determine the height, steepness, and horizontal shift of the sigmoid curve.

The PYO production within the biofilm is examined across three designated regions (depicted in Fig. 3.31) over three time intervals (TS = 20, TS = 25, and TS = 30), from which we derive a solution space comprising various sigmoid activation function variants.

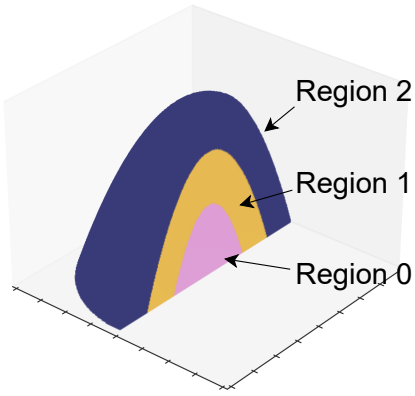



Figure 3.31: Illustration of the three layers within a biofilm analyzed for computing reliability and solution space exploration, where "Region 2" represents the outer layer with the greatest nutrient access. "Region 1" serves as the intermediate layer, and "Region 0" constitutes the core layer, characterized by the least access to nutrients.

This analysis includes evaluating their dynamics in relation to QS influences. Research indicates that employing modified activation functions such as scaled sigmoid, penalized Tanh, and bounded ReLu can enhance performance for specific computational tasks. It has been demonstrated that these improved versions of standard non-linear activation functions outperform in the context of application-

specific problems [119, 120, 121]. Consequently, a biological system featuring a diverse array of non-linear functions offers significant benefits for computational applications, including adaptive classification or analog to digital conversion, by providing increased specificity and adaptability.

In region 1, QS levels are lower due to reduced nutrient accessibility, as these bacterial cells reside deeper beneath the surface. At $TS = 20$, QS concentrations in region 1 are similar to those in region 2 at $TS = 25$, leading to comparable sigmoid parameters. However, region 1 at $TS = 20$ exhibits more noise in its 3D sigmoid plot, indicative of higher computing uncertainty—mirroring the lower MI values observed compared to region 2. This uncertainty escalates in region 0, where the sigmoid function plot shows significant distortion and noise.

Despite these challenges, regions 2 and 1 present a series of sigmoid curves that form a dependable solution space. Additionally, environmental phosphate levels act as a fine-tuning element, with each 3D sigmoid plot exhibiting slight variations in shape related to phosphate concentration changes. These results further elucidate the importance of MC for computing as under **RA 3** that provide a better understanding of the **RQ1**.

## 3.5 GRNN Plasticity

This section discusses **RA 4**, which focuses on **RQ 3** and extends the GRNN computing diversity by delving into their remarkable cellular plasticity from the ANN perspective as shown in Fig. 3.33. It is important to emphasize that this study introduces two types of plasticites; **input-dependent plasticity**, which refers to the cell's adaptive response to varying environmental stimuli, and **temporal plasticity**, denoting the cell's ability to modify its behavior over time in response to sustained changes in its surroundings.

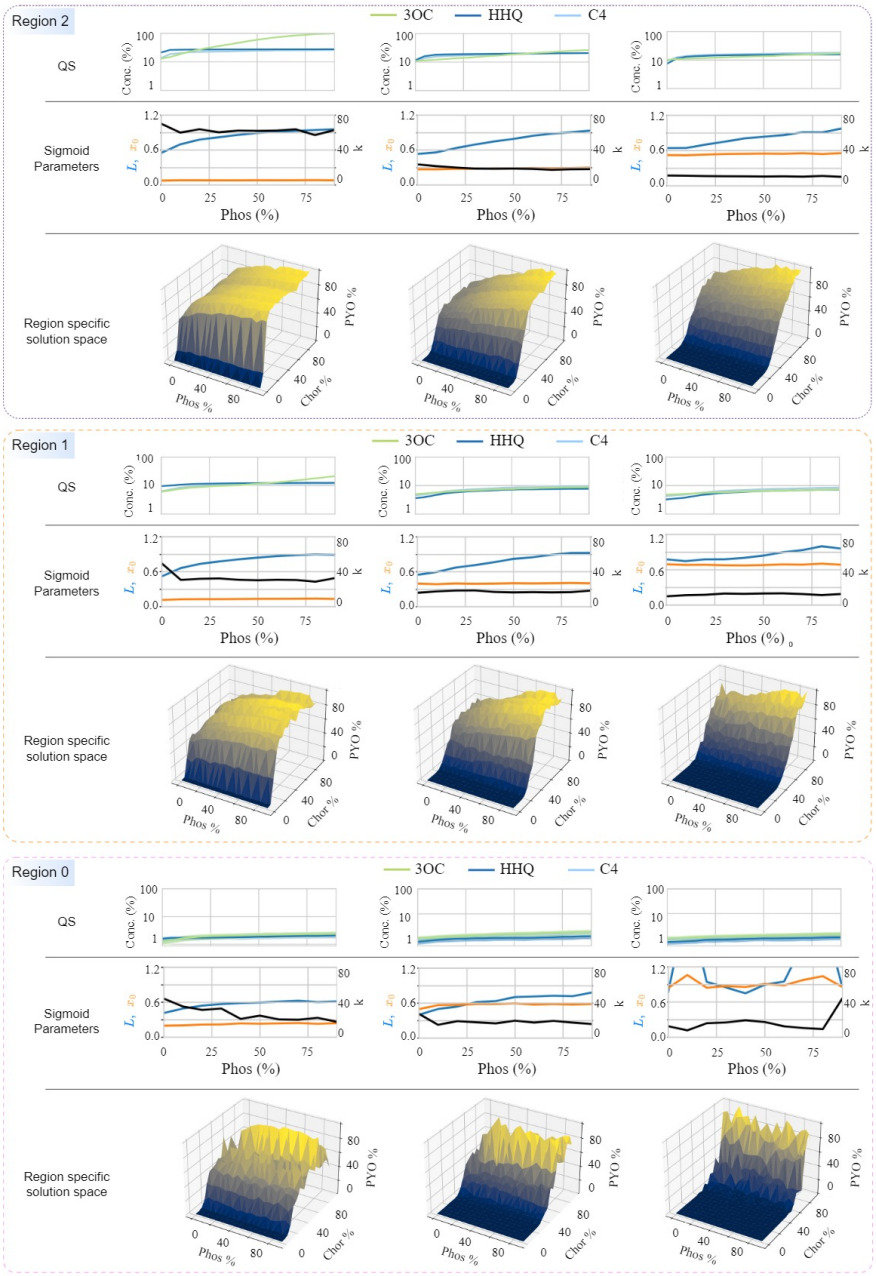



Figure 3.32: Depiction of a biofilm sigmoid function diversity, showcasing non-linear behavior variations across different locations (columns) and over time (rows). The diagram is structured into layers, each illustrating QS signal variations, sigmoid parameters, and curve changes specific to biofilm regions. QS plots highlight percentage differences in 3OC, HHQ, and C4 signal concentrations, while plots of sigmoid parameters detail adjustments in the curve's height ($L$), steepness ($k$), and horizontal shift ($x_0$).

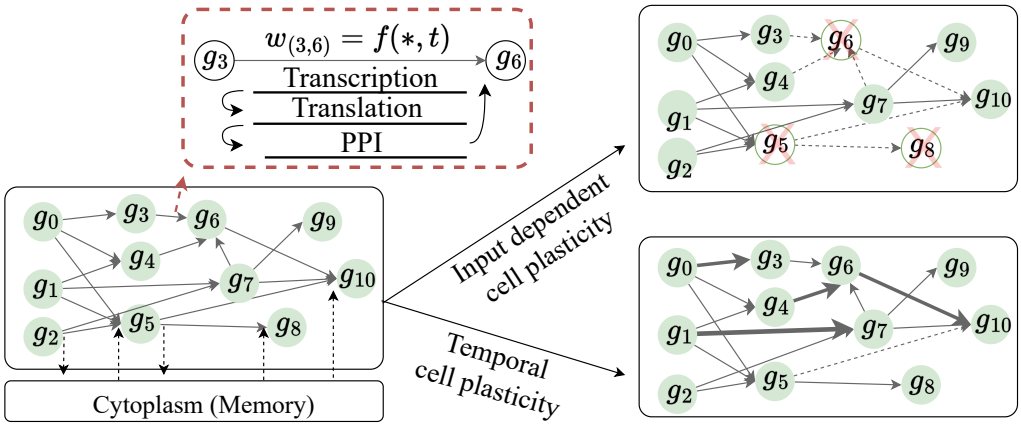



Figure 3.33: Within the GRNN framework, gene-perceptrons function akin to perceptrons in ANNs, processing inputs through weights shaped by multi-omic layer interactions. Bacterial cells display input-dependent plasticity through distinctive gene expression pathways that vary with diverse inputs. Furthermore, they exhibit temporal plasticity by adjusting the interaction weights of GRNN subnetworks over time.

### 3.5.1 Input-dependent plasticity

This phenomenon emerges from the genes' selective reaction to particular input chemicals, as detailed by Zhang et al. in [122]. Similar to the input layer in ANNs, genes situated at the GRNN's periphery display heightened sensitivity to specific chemical effectors. This sensitivity orchestrates the activation of particular gene subsets in response to the presence of these chemicals. Consequently, the GRNN adeptly modulates information flow, engaging only pertinent expression pathways and leaving other genes in a state of quiescence. Such selective activation facilitates the employment of distinct GRNN subnetworks tailored to the chemical inputs, a process depicted in Fig. 3.33, thereby bolstering the cellular gene regulatory mechanism's energy efficiency.

### 3.5.2 Temporal plasticity

Conversely, *temporal plasticity* pertains to the dynamic adjustments in gene interactions over time, as explored by Rivera et al. in [123]. This adaptation mirrors the concept of weight plasticity within neural networks, adjusting the strength of gene

influences to cultivate an optimal response to the environment under unchanged input conditions. This aspect of plasticity underscores the cell's capability to refine its behavior through internal regulatory modifications, ensuring survival and efficiency in fluctuating environmental contexts.

Section 3.6.2 elucidates experimental results on employing GRNN plasticity to increase the computing diversity.

## 3.6 GRNN Computing Applications

This section represents **RA 5** and **RA 6**, exploring the practicality of employing GNNs for general computing tasks exploiting **RQ 4** and **RQ 5** from another perspective. It begins by assessing the potential of GRNNs in performing regression tasks and examining their effectiveness and adaptability. Following this, the focus shifts to exploring how GRNNs can be applied to classification tasks.

### 3.6.1 GRNN Application in Regression

Mathematical regression techniques have long served as a cornerstone in data mining applications, as evidenced by numerous studies [124, 125]. Subsequently, this section is dedicated to analyzing the GRNN for a spectrum of regression problem types as shown in Fig. 3.34 utilizing previously extracted *E. coli* GRNN.

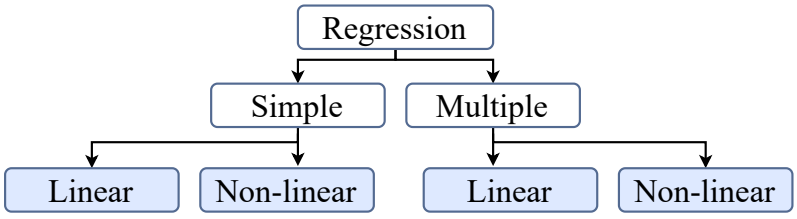



Figure 3.34: Illustration of sub-categories of regression problems.

While this approach provides the freedom of choosing any gene-perceptron as the input, this research task begins with linear regression analysis using *E. coli* gene-perceptron *b3067* as the input, due to its capability of influencing 1703 other gene-

perceptrons. Gene-perceptron *b3067* is stimulated with 25 varying concentration levels, while other gene-perceptrons start at minimal expression values based on existing datasets. This setup is repeated 10 times for each concentration level to ensure accuracy.

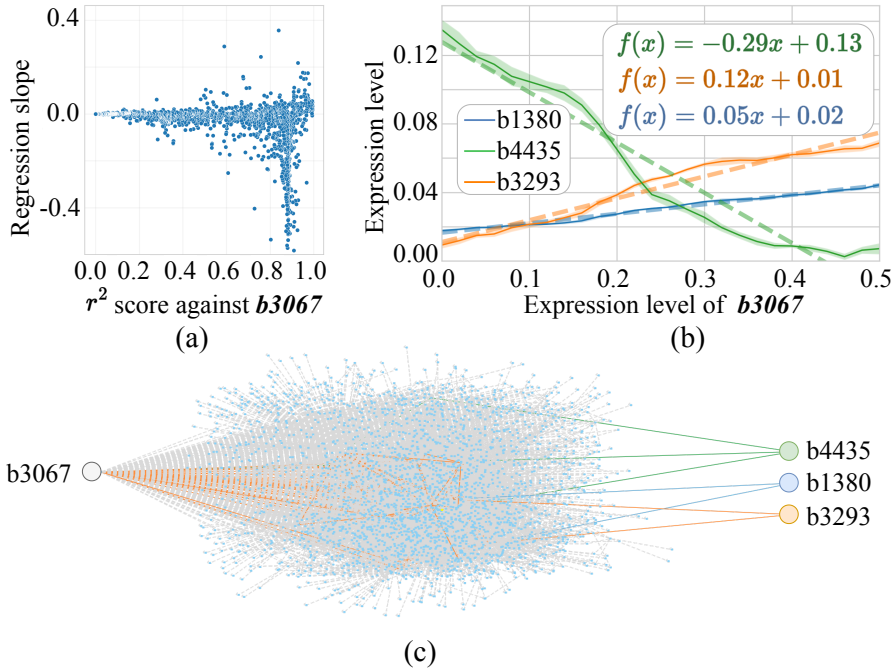



Figure 3.35: Depiction of simple linear regression utilizing *E. coli* GRNN: a) presents the distribution of regression slopes for all genes against their corresponding $r^2$ scores, b) demonstrates three regression lines corresponding to three output gene-perceptrons, and c) showcases the sub-GRNN designed for these linear regressions.

Figure 3.35a reveals the diversity in linear regression slopes for *E. coli* GRNN, with the highest slopes identified as 0.36 (positive) and -0.50 (negative) when *b3067* is the input gene-perceptron. Figure 3.35b exemplifies three gene-perceptrons with varying slopes, and Figure 3.35c displays their corresponding sub-GRNNs, highlighting the network's parallel computing ability. This illustrates the GRNN's versatile linear regression capability, enabling tailored mapping of gene-perceptrons for specific applications.

The GRNN's outputs, as previously discussed, are further analyzed for quadratic polynomial regressions. Figure 3.36a displays the quadratic and linear coefficients of each gene perceptron, color-coded by the *RSS* (the residual sum of squares)

value, indicating the fit quality. The highest $RSS$ values correspond to quadratic coefficients ranging from -2 to 2 and linear coefficients from -1 to 0.5.

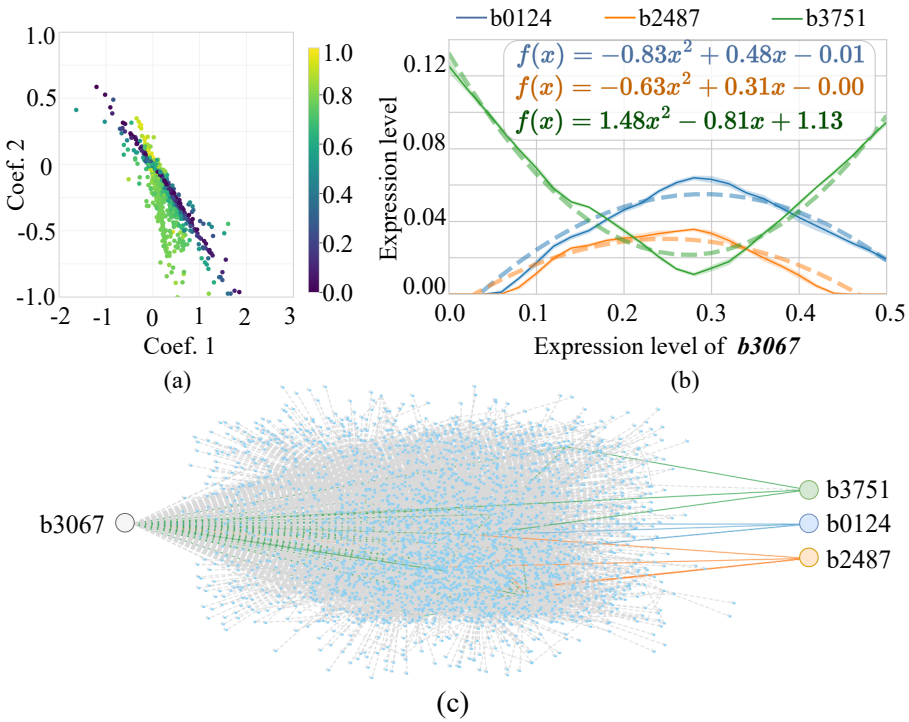



Figure 3.36: Visualization of non-linear quadratic regression via *E. coli* GRNN: a) depicts the distribution of quadratic and linear coefficients for all genes, with color coding based on the $RSS$ value, b) presents three sample regression curves, and c) features the sub-GRNN linked to these sample regression curves.

Figure 3.36b highlights the diversity in quadratic regressions with examples from genes *b0124*, *b2487*, and *b3751*, showcasing varying quadratic coefficients. Corresponding sub-GRNNs for these genes are shown in Figure 3.36c, indicating that different output gene-perceptron combinations can transition from linear to quadratic regressions, all with *b3067* as the input gene-perceptrons.

Similarly, the analysis is extended to cubic polynomial regressions for *E. coli* GRNN. Figure 3.37a presents the cubic coefficients, with Figure 3.37b showcasing example curves, and Figure 3.37c displaying the corresponding sub-GRNNs. Cubic coefficients with $RSS > 0.7$ span from 0 to 13, indicating limited variation in higher-degree polynomial regressions and suggesting a potential limitation in exploring complex functions. However, this analysis, centered around the input gene-

perceptron *b3067*, hints at a broader solution space achievable through diverse input configurations.

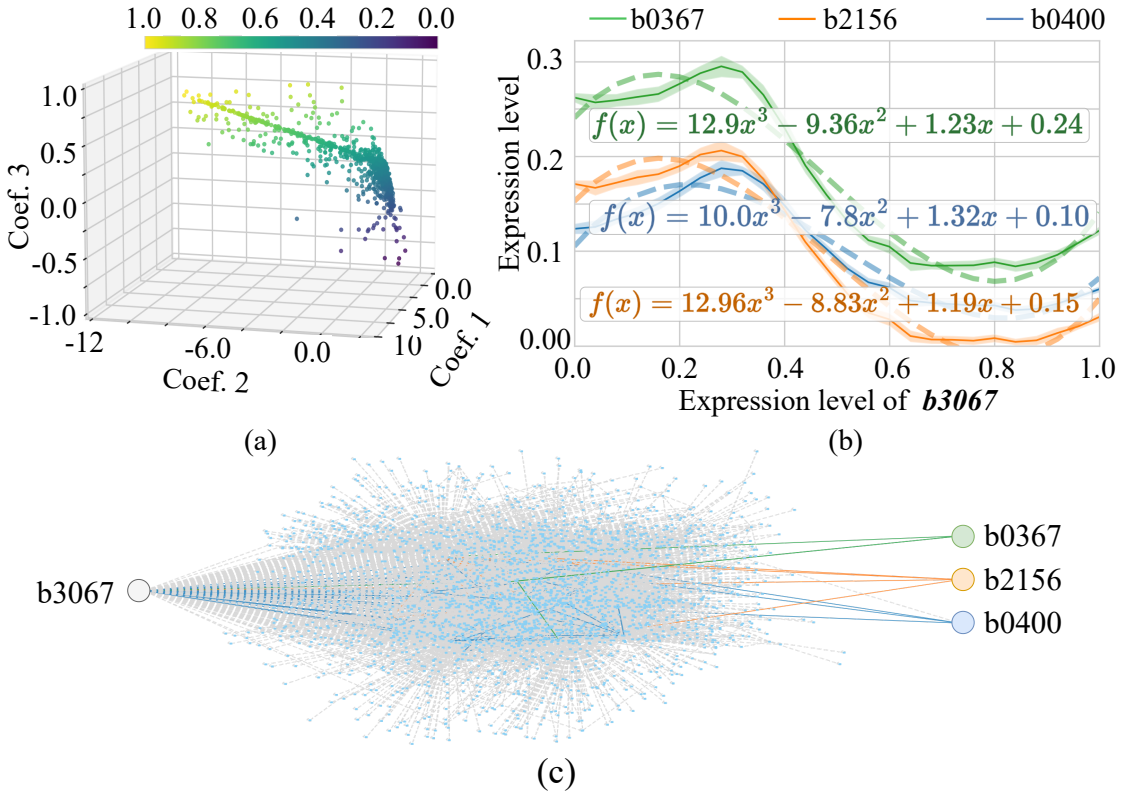



Figure 3.37: Depiction of non-linear cubic regression with *E. coli* GRNN: a) portrays the distribution of cubic, quadratic, and linear coefficients across all genes, with a color-coding scheme based on the *RSS* value, b) provides three examples of cubic regression curves, and c) visualizes the corresponding sub-GRNNs for these three cubic regression instances.

The GRNN-based regression tasks are further analyzed by exploring multiple regression in *E. coli* GRNN with input gene-perceptrons *b3067* and *b3357*. The study varies expression levels from 0 to 0.5 to create 625 configurations, using *RSS* to gauge model variance.

Coefficient variations are detailed in Figure 3.38a, with significant outcomes for output *b3090* shown in Figure 3.38b. The corresponding sub-GRNN configuration is visualized in Figure 3.38c. This approach highlights the capacity to generate diverse modeling solutions by varying input and output gene-perceptron pairs.

Multiple polynomial regressions, such as those estimating "Affective States" in

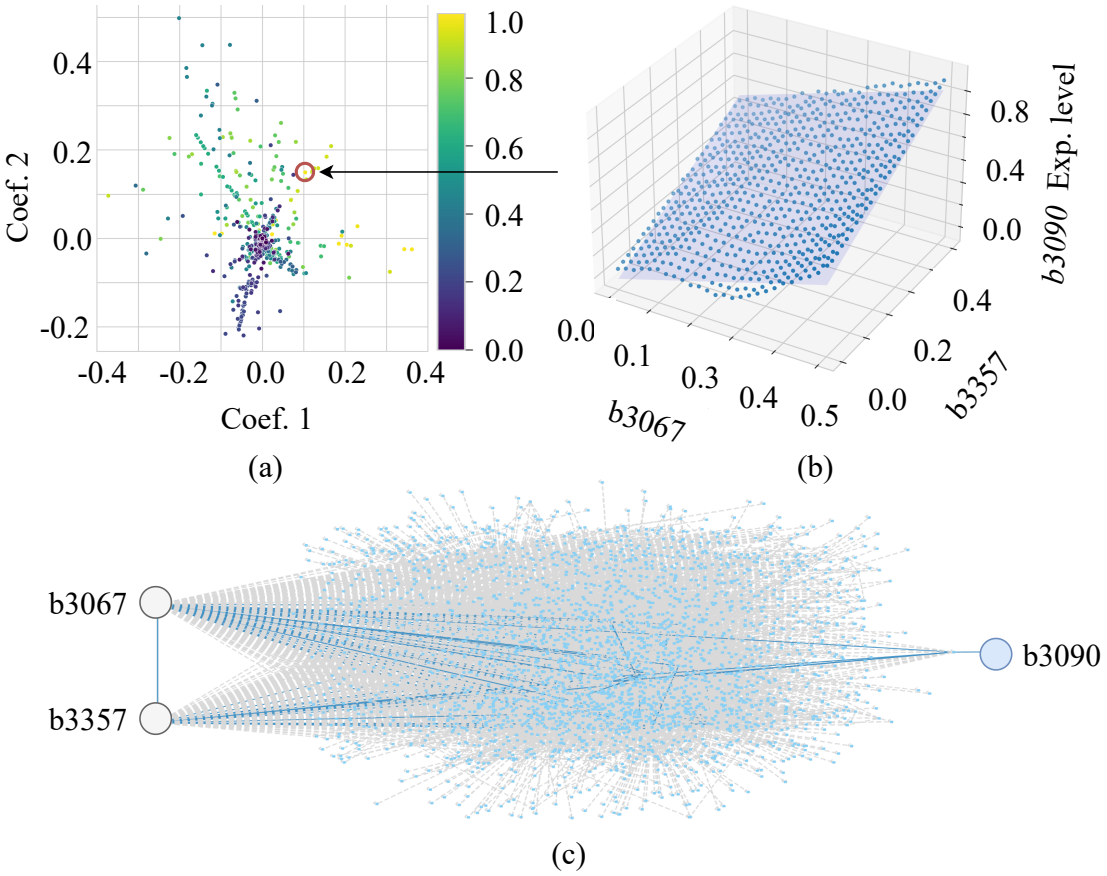



Figure 3.38: Illustration of multiple-linear regression through *E. coli* GRNN with gene-perceptrons *b3067* and *b3357* as inputs: a) illustrates the distribution of the first and second coefficients for all genes, color-coded according to the *RSS* value, while b) and c) display the example plane for the output gene-perceptron *b3090* and the associated sub-GRNN, respectively.

humans, are next analyzed using gene-perceptrons *b3067* and *b3357* as inputs. For this analysis a generic regression model is defined as follows,

$$f(x_1, x_2) = d_1 x_1^2 + d_2 x_2^2 + d_3 x_1 x_2 + d_4 x_1 + d_5 x_2 + d_6, \tag{3.5}$$

where coefficients $d_1$ to $d_6$ correspond to the inputs and interactions of *b3067* and *b3357*.

Results showcased in Fig. 3.39 illustrate the capability to derive complex multi-variable polynomial models. Coefficients $d_1$ and $d_2$ indicate the model's quadratic nature, affecting its curvature based on input concentrations, with distributions ranging notably in skewness. The cross-term $d_3$ reflects the combined input ef-

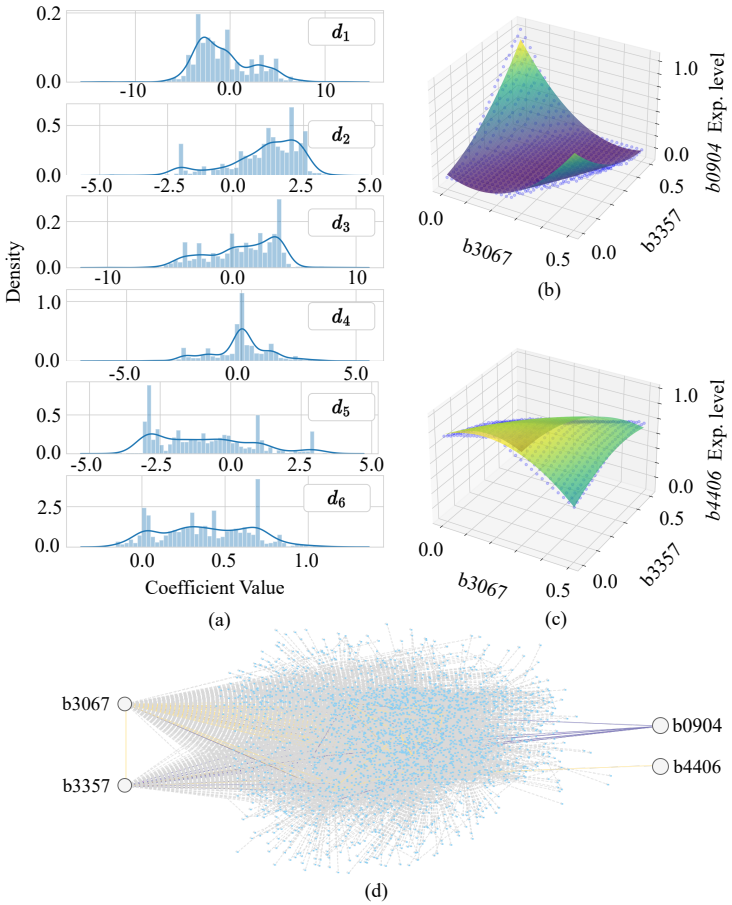



Figure 3.39: Illustration of multiple non-linear regression employing *E. coli* GRNN with gene-perceptrons *b3067* and *b3357* as inputs: a) displays the coefficient distributions related to the equation (3.5), b) and c) illustrate example curves characterized by positive and negative coefficients for coef. 1, respectively. Following this, d) reveals the sub-GRNNs corresponding to the regression examples depicted in b) and c).

fect, whereas $d_4$ and $d_5$ influence the curve's vertical displacement. The constant $d_6$ determines the curve's baseline position. Figures 3.39b and 3.39c present distinct examples of multi-variable polynomial regressions, highlighting the diversity of outcomes possible with this modeling approach.

In summary, this comprehensive analysis underscores the GRNN's ability to model a range of regression tasks, from linear to cubic polynomial regressions, through strategic stimulation of specific input gene-perceptrons. The findings reveal the network's adaptability and potential for application in diverse computational biology scenarios, offering a foundation for future explorations of gene-regulatory

networks in *E. coli* and beyond. Further information and results associated with this analysis was published as presented in Chapter 9.

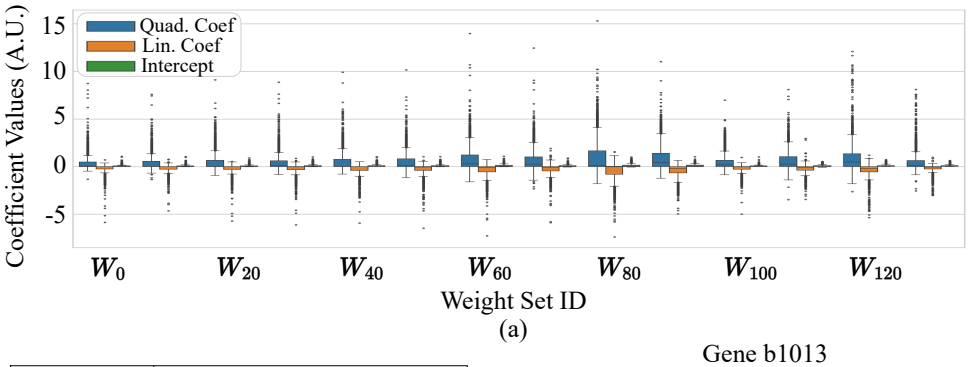
### 3.6.2 GRNN Applications in Regression with Plasticity

This section presents a use case of regression to elucidate how cell plasticity theoretically broadens computing diversity focusing on the *E. coli* GRNN as a use case model.

Figure 3.40 demonstrates the variance in regression outcomes for a GRNN, attributing to temporal cell plasticity. It shows box plots for the coefficients of 2,875 gene-perceptrons across different weight configurations ($w_i$), highlighting substantial diversity in quadratic coefficients and, by extension, the curvatures of regression models. This variation underscores the expansive solution space available for specific applications, with linear regressions emerging when quadratic coefficients equal zero. Linear coefficients generally show negative values, whereas intercepts fall within a more confined range, particularly between 0 and 2.

Figures 3.40b and 3.40c showcase regression curves for the gene-perceptron *b1013* across various timesteps, with five different quadratic coefficients indicating changes from linear to more curved regressions under different weights. This illustrates how temporal plasticity enriches the solution space beyond static weight configurations, enabling multiple regression models for each output gene-perceptron, as opposed to a single model per gene-perceptron in static scenarios.

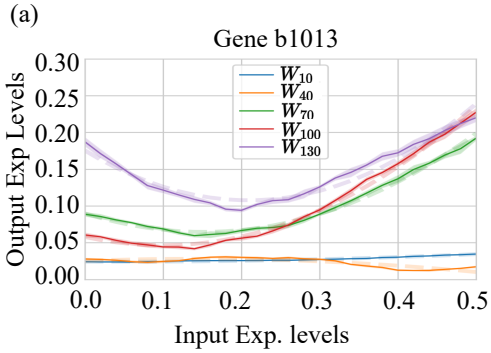
In conclusion, this section highlights how cell plasticity enhances the computing diversity of the *E. coli* GRNN through regression modeling. The analysis demonstrates significant variance in gene-perceptron coefficients across different weight configurations, expanding the solution space for various applications. Temporal plasticity further enriches this diversity, allowing multiple regression models per gene-perceptron, thereby enhancing the GRNN's computational flexibility and application potential in computational biology.




(a)

| Weightset ID | Regression |
|---|---|
| $W_{10}$ | $0.04x + 0.02$ |
| $W_{40}$ | $-0.13x^2 + 0.03x + 0.02$ |
| $W_{70}$ | $1.10x^2 - 0.33x + 0.08$ |
| $W_{100}$ | $1.17x^2 - 0.23x + 0.05$ |
| $W_{130}$ | $1.55x^2 - 0.64x + 0.17$ |

(b)




(c)

Figure 3.40: The regression analysis, utilizing *b3067* as the exclusive input gene-perceptron, encompasses a) examination of quadratic and linear coefficients, along with intercepts, across various weight configurations to delineate the solution space; b) regression coefficients for *b1013*; and c) the corresponding regression curves.

These results were published as presented in Chapter 8.

### 3.6.3 GRNN Application in Classification

Contrary to the regression capabilities of GRNN explored previously, this study delves into the feasibility of using GRNN for classification tasks, with a specific focus on image recognition applications.

Acknowledging that GRNNs are essentially randomly structured pre-trained NNs, solving problems utilizing GRNNs necessitates the identification and extraction of the suitable sub-GRNN. Consequently, this thesis proposes a tailored algorithm aimed at facilitating the search for application-specific sub-GRNNs in the next section.

**Application-specific sub-GRNN Search Algorithm for Classification**

The search algorithm employs a random permutation method to identify the optimal sub-GRNN for specific problems, aiming for precise classification.

The algorithm first identifies suitable candidates for the input layer focusing on each gene's inward and outward degrees, as shown in Fig. 3.41 (Step 1). Gene-perceptrons with an inward degree close to zero, minimize the influence of unnecessary incoming signals aside from the problem-specific inputs. Moreover, the input gene-perceptrons with a higher outward degree can pass information to mere sections of the GRNN allowing complex computing capabilities. This filtration process employs graph theoretical degree distribution to create a set of gene-perceptrons for the input layer, denoted as $G(Trimmed)$. The $G(Trimmed)$ set contains $P'$ genes, where $P'$ is less than the total number of genes $P$ in the GRNN.

Since $G(Trimmed)$ comprises $P'$ gene-perceptrons and the problem involves $K$ features, the number of possible input layers that can be generated is $P'PK = \frac{P'!}{(P'-K)!}$. Given the immense number of potential sub-GRNNs, a heuristic search algorithm might be more efficient. However, because exploring such algorithms is beyond the scope of this study, we employ a random permutation-based algorithm instead.

Further in Step 1, the algorithm randomly selects $G(In_J)$, a set of $K$ inputs for the $J^{th}$ permutation, where $J = 0, 1, 2, ..., P'PK$ and $K$ represents the number of input features for the problem. To mimic the base behavior of the cell at $t = 0$ (the initial time of the computing process), a base-TF array is created using the expression levels at the zero timestep from the transcriptomic data used for weight extraction [115] (GEO accession number GSE65244) before encoding the search dataset into expression levels (Fig. 3.41 - Step 2). This step is essential to ensure that the cell's functions align with the environmental conditions at the start of the process.

Next, in order to create the input matrix $\mathbf{I}^{(t=0)}$, the base-TF array is then mod-

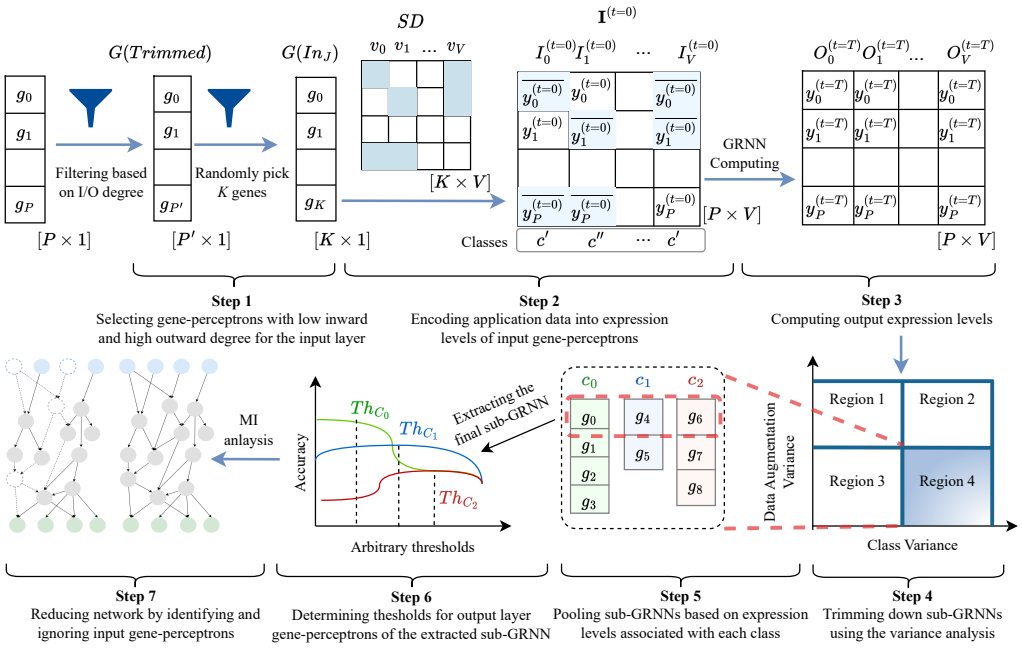



Figure 3.41: Illustration of the proposed application-specific sub-GRNN search algorithm for One-vs-All classification includes several key steps. Step 1 selects input gene-perceptrons ($G(Trimmed)$) based on degree distributions and chooses a subset ($G(In_j)$) for the application's input features ($K$). The search dataset is then converted into an expression-level input matrix ($I^{(t=0)}$), and the corresponding output matrix ($O^{(t=T)}$) is calculated using the base-GRNN model in Step 2 and 3. A set of gene-perceptrons demonstrating significant expression variance between classes and minimal within-class variance is identified for class pooling based on expression levels in Step 4. In Step 5 and 6, the algorithm optimizes expression thresholds for each class to enhance accuracy and Step 7 performs a MI analysis to prune insignificant input gene-perceptrons, streamlining the network.

ified to encode the inputs of search dataset. This matrix contains input TF arrays $I_v^{(t=0)}$, where $v = 0, 1, 2, ..., V$. For a digital search dataset, state "1" represents the highest expression level of the corresponding gene, while state "0" represents the lowest value. In the case of an analog search dataset, the values are normalized and mapped to concentrations based on the highest and lowest expression levels of the relevant gene. Following the decoding of all input records with the expression levels in Step 2, the output expression levels are computed in Step 3 using the mathematical model described in (3.2) and (3.3). This subseqently produces an expression matrix, $O^{(t=T)}$, with output arrays $O_v^{(t=T)}$, where $v = 0, 1, 2, ..., V$ corresponds to each class.

Step 4 performs a variance analysis to identify gene-perceptrons suitable for representing each class at the output layer of the sub-GRNN. A gene-perceptron is considered a good candidate to represent a class $c_i$ if it expresses at a higher level for the corresponding input $I^{(t=0)}$ of that class, while maintaining low variance within the same class and higher variance between different classes. Therefore, we search for gene-perceptrons for all classes in "Region 4" (as shown in Fig. 3.41 - Step 4), where the variance between classes is high and the variance within records of the same class is low. This results in a set of output gene-perceptrons.

To assign a gene-perceptron $g_i$ to class $c_l$ in Step 5, it must meet the condition $\overline{y}(g_i, c_l) > \overline{y}(g_i, c_m) \forall m < |c|, m \neq l$, where $\overline{y}(g_i, c_l)$ is the mean expression level for class $c_l$. This process is carried out for all gene-perceptrons in "Region 4". Finally, the gene-perceptron with the highest mean expression level and the largest gap from the others is chosen to represent each class.

Step 6 focuses on identifying the threshold for each gene-perceptron using an accuracy-maximizing approach. We first determine the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each class using an arbitrary threshold value $Th = a$. The accuracy for class $c_l$ is then calculated as follows:

$$Acc(c_l, Th = a) = \frac{TP + TN}{TP + TN + FN + FP}.$$

In order to determine the optimal threshold for class $c_l$, this calculation is repeated for various thresholds $a$, ranging from zero to one in 0.05 increments. The threshold is identified by:

$$Th = \arg\max_{a} Acc(c_l, Th = a).$$

This process is iteratively performed for all classes involved in the problem. Next, this process is repeated multiple times ($< {}_{P'}P_K$), ranking the sub-GRNNs based on

accuracy to select the best candidate.

Following the selection of a suitable application-specific sub-GRNN, a perturbation-based MI analysis is conducted in Step 7 to optimize the network. During this step, all inputs of the extracted sub-GRNN are subjected to signals fluctuating from zero to one, and the outputs are recorded from the gene-perceptrons at the output layer. The MI between the input and output nodes is calculated as:

$$I(g_x; g_y) = \int f(x, y) \cdot \log \left( \frac{f(x, y)}{f(x) \cdot f(y)} \right) dxdy$$

where $g_x$ and $g_y$ are the input and output nodes, respectively, and $f(x, y)$ is the joint probability density function of $g_x$ and $g_y$ expressions. The amount of information flow from input nodes to output nodes is indicated by these MI values. Input nodes with lower MI values can be disregarded, leading to a reduced and more efficient network.

### Digit Classification Use case

This section outlines the experimental setup for digit classification using a proposed sub-GRNN search algorithm and evaluates the performance of GRNN computing.

To simplify the analysis, the study employs $4 \times 4$ images, totaling 16 pixels, and a search dataset, comprising five digit classes ("0", "1", "2", "6", and "7") with 10 distinct pattern augmentations as shown in Fig. 3.42, resulting in a dataset size of $50 \times 16$ alongside a $50 \times 1$ label matrix.

Utilizing the search algorithm explained previously, the GRNN sub-network depicted in Fig. 3.43 is extracted. Next, the suggested perturbation-based MI analysis is performed on the identified application-specific sub-GRNN. This process involves applying fluctuating input signals, ranging from zero to one, to all inputs of the sub-GRNN. The responses from five output gene-perceptrons, specifically *b2436, b0613, b0675, b2417*, and *b3902*, are then recorded.

Based on the MI values, an optimal network is derived, that only contains *b0080*,

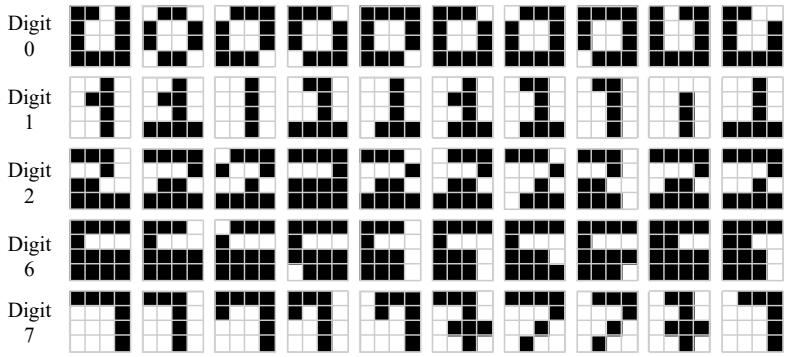



Figure 3.42: $4 \times 4$ pixel images representing five digit classes and their corresponding augmentations.

*b4401*, *b0889*, *b2217* and *b3905* as the input layer gene-perceptrons.

The study subsequently evaluates the computing accuracy of the sub-GRNN, comparing its performance before and after network reduction. The aim is to assess the impact of reducing input quantity on network effectiveness, with results presented in Fig. 3.44.

Findings indicate that the optimized sub-GRNN maintains accurate computing capabilities comparable to its prior configuration, despite a simplified structure with fewer input nodes. This simplification potentially leads to lower ATP energy requirements for computational processes and reduces noise leading to enhanced reliability.

## 3.7 Summary

This thesis explores the computational capabilities of bacterial systems by integrating MC and GRNNs, while aiming to understand the underlying mechanisms of bacterial behavior and functionality, providing insights into the potential applications of bacterial biocomputing.

The initial phase of the thesis investigates the MC mechanisms within bacterial populations, specifically focusing on the human GB. The human GB is a complex ecosystem of bacteria that interact through metabolite exchange, significantly influ-

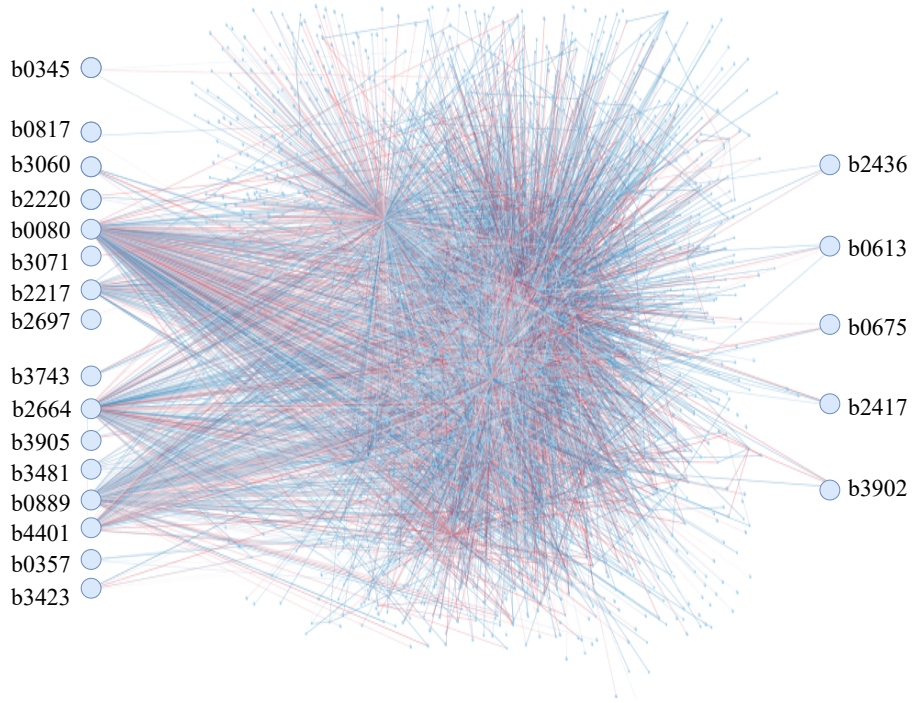



Figure 3.43: Illustration of perturbation-based MI analysis conducted on the input and output layers of the extracted sub-GRNN.

encing host metabolic functions. This intricate network of interactions, documented in databases like Metacyc and KEGG, plays a crucial role in nutrient extraction, metabolite absorption, and overall health. Dysbiosis in the GB, caused by various factors such as diet, genetics, and external agents, can lead to health issues including inflammatory bowel disease, diabetes, and cancers. To model these interactions, a VGB simulator was developed, employing a two-tiered framework to represent the GB's metabolic and communication processes. The upper layer simplifies the GB into a graph of bacterial populations exchanging metabolites, while the lower MC layer models the molecular signals facilitating these exchanges.

In-silico experiments using the VGB simulator further validate the theoretical models. These experiments focus on SCFA production within the GB, exploring how changes in bacterial population sizes and metabolite inputs affect the overall system dynamics. The findings highlight the intricate relationships between bacterial composition, metabolite production, and computational efficiency, offering insights

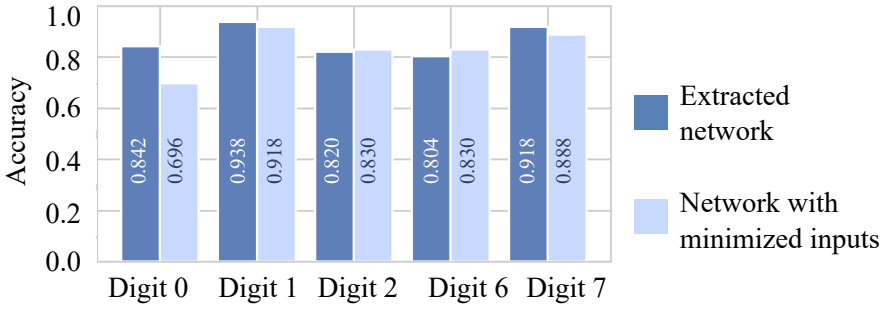



Figure 3.44: Comparison of the extracted sub-GRNN's accuracy before and after reducing the number of inputs, with darker columns indicating class accuracies prior to reduction and lighter columns showing accuracies post-input minimization.

into optimizing bacterial systems for enhanced biocomputing applications.

Following the exploration of MC, the research delves into GRNNs to understand their computational capabilities at the cellular level. GRNNs mimic the biological gene regulatory networks, allowing the quantification of complex gene-gene interactions through computational models. The study constructs GRNNs for different bacterial species, including *E. coli* and *P. aeruginosa*, validating their accuracy against experimental data. The results demonstrate the feasibility of using GRNNs to replicate biological processes, highlighting their potential for cross-genome applications and the possibility of performing biocomputing.

The integration of MC with GRNNs provides a comprehensive understanding of natural bacterial computing. By simulating bacterial ecosystems as graph networks, the research captures the dynamic interactions between cells and their molecular environment. This approach reveals how intercellular communication influences the dynamics of GRNNs, enhancing our understanding of bacterial adaptive behaviors and survivability. This is an essential analysis in order to place bacteria as a viable biocomputing solution.

The study also evaluates the energy efficiency and computational complexity of GRNNs compared to traditional computing platforms. The results indicate that GRNNs exhibit remarkably low power consumption, particularly in high-complexity tasks, making them highly efficient for biological computing applications. This en-

ergy efficiency, combined with the GRNN's structural complexity, underscores their suitability for general computing tasks and their potential for functioning as a neuromorphic computing platform.

The dissertation also delves into the development of an application-specific sub-GRNN extraction algorithm, which is pivotal for identifying GRNNs to specific computational tasks. This algorithm extracts suitable sub-networks from the larger GRNN, optimizing them for tasks such as regression and classification.

The feasibility analyses of these tasks demonstrate the practical applicability of GRNNs in solving real-world problems. For regression, the study evaluates various types, including linear, quadratic, and cubic regression models, using the extracted sub-GRNNs. These models effectively predict gene expression levels based on input variables, showcasing the accuracy and versatility of GRNNs in handling complex, non-linear relationships.

Similarly, the classification feasibility analysis underscores the potential of GRNNs in distinguishing between different classes based on gene expression profiles. By applying the sub-GRNN extraction algorithm, the research identifies the most significant gene-perceptrons that contribute to accurate classification. This approach enhances the precision and efficiency of GRNNs in categorizing data, which is crucial for applications in medical diagnostics, environmental monitoring, and synthetic biology. The successful analysis of regression and classification tasks highlights the robustness and adaptability of GRNNs.

Overall, this dissertation advances our understanding of bacterial biocomputing, demonstrating the potential of integrating MC and GRNNs to model and harness the computational power of bacterial systems. The research provides a foundation for future studies in energy-efficient, biocompatible and generalizable bacterial computing platforms.

# Chapter 4

# Conclusion and Future Work

This chapter presents the conclusion in Section 4.1, which addresses the three research questions. This is followed by the future work outlined in Section 4.2.

## 4.1 Conclusion

AI has become a cornerstone of modern technological advancements, transforming from an intriguing idea into an essential part of everyday life. It has revolutionized industries, improved problem-solving abilities, and significantly influenced societal norms, leading to groundbreaking applications across various sectors, including manufacturing, automotive, finance, and healthcare.

In contrast to AI's capabilities, a key challenge is the high energy demand required for training and functional phases, especially in deep learning and large language models. Numerous efforts to address this issue focus on hardware optimization, with neuromorphic systems emerging as a promising solution due to their better energy efficiency. However, neuromorphic computing faces certain challenges when it comes to the general-purpose applications. While biocomputing platforms offer energy efficiency and biocompatibility, their limited generalizability presents significant obstacles for widespread adoption.

Therefore, this thesis aims to explore the inherent computing capabilities of bac-

teria to address these limitations and introduce them as a novel, energy-efficient, biocompatible and generalizable computing platform. Initially, the thesis formulated first RQ; **"How can communication of bacterial multi-species computing be used to understand population network structures?"** that explores how bacterial communication through molecular signaling and cross-feeding shapes population network structures. By studying these interactions, the goal is to understand how bacterial communities adapt and self-organize, enhancing their resilience and ecological balance. In order to model these interactions, focusing on the human GB, a VGB simulator is developed as one of the contributions of this thesis. In addition, a two-tiered framework is introduced with the upper layer representing bacterial populations exchanging metabolites and the lower MC layer modeling molecular signals. In-silico experiments using the VGB simulator, particularly on SCFA production, validate the theoretical models and provide insights into the involvement of MC and compositional changes in altering bacterial behaviors.

This thesis next targets the RQ 2: **"Can gene regulation networks be used to discover artificial neural networks for biocomputing?"** that aims to leverage bacterial GRN to derive ANN-like sructures, positioning bacterial cells as biocomputing hardware for efficient, biocompatible, and generalizable computational tasks. Subsequently, the thesis introduces the novel concept of GRNNs that resembles a random-structured, pre-trained NN by examining the gene expression-based computational capabilities at the cellular level. GRNNs are extracted by quantifying gene-gene interactions using transcriptomic data, and their accuracy is validated against experimental data for bacterial species such as *E. coli* and *P. aeruginosa*. Additionally, combining MC with GRNNs provides a comprehensive understanding of natural bacterial computing by modeling bacterial ecosystems as graph networks. This reveals how intercellular communication affects GRNN dynamics and improves our knowledge of bacterial adaptive behaviors and survivability. Further, this thesis found that, using GRNN analysis, it is possible to extract secreted molecular

species from bacterial populations, such as biofilms, that can act as mathematical functions, particularly modified sigmoid functions. The features such as height, steepness, and shift of biofilm-based sigmoid functions can be fine-tuned by regulating environmental conditions. The ability to harness and modify these biofilm-based functions extends the versatility of bacterial population-based computing, allowing for sophisticated control over computational processes within a living AI system.

Moreover, the bacterial adaptability explored in this thesis focusing on RQ 3: **"How can the bacterial computing diversity be expanded by exploiting cellular plasticity?"** introduces a new form of plasticity to the AI world in terms of dynamic weights. These cells can dynamically adjust their internal states in response to external stimuli, allowing for real-time, context-aware modifications to their computational processes. This, in turn, leads in a more flexible and resilient form of biocomputing paradigm. Moreover, the incorporation of dynamic weights expands the computing diversity massively, enabling a broader range of applications and more robust performance across various tasks. It allows for the development of living AI systems that are capable of learning and adapting on the fly. This new approach opens up possibilities for more sophisticated, adaptive AI solutions that can handle real-world variability with greater ease and precision, pushing the boundaries of what current AI technologies can achieve. Further, posibility of regulating weights can be exploited in the future.

Further, this thesis assesses the energy efficiency and computational complexity of GRNNs compared to traditional computing platforms. The findings elucidate that GRNNs have remarkably low power consumption, especially in high-complexity tasks, making them highly efficient for biological computing applications. This efficiency, combined with their structural complexity and analog computing capabilities highlights their potential as a wet-neuromorphic computing platform. Additionally, targeting the RQ 4: **"Can mathematical and pattern recognition applications be realized through bacterial neural networks?"** a feasibility analysis

on regression revealed the potential for bacterial computing systems to perform tasks ranging from simple linear to complex multiple polynomial regression, demonstrating high generalizability. This capability could be applied *in-situ* in medical diagnostics for environmental monitoring for predicting pollutant dispersion and accurately modeling disease progression. These practical applications highlight the versatility of bacterial computing in handling diverse regression tasks across various fields.

Finally, RQ 5 **"What search algorithms can be developed to discover natural GRNN for biocomputing applications?"** results in an application-specific sub-GRNN extraction algorithm that optimizes sub-networks for classification tasks, enabling bacterial computing systems to perform molecular pattern recognition. This innovation allows bacterial cells to be used in future practical applications such as disease identification, specific environmental condition identification, or even generic image classifications.

In addition to positioning bacterial-based computing as an energy-efficient and generalizable novel computing approach, it is embeddes with the unique property biocompatibility. The biocompatibility is particularly advantageous in contexts where implantable devices are essential, such as health monitoring, drug delivery, smart diagnostics and even in environmental monitoring. Silicon-based devices, while advanced, often face significant biocompatibility challenges, leading to issues such as immune rejection, inflammation, and long-term stability problems within the body. In contrast, bacterial cells used in the proposed computing approach can integrate seamlessly with biological tissues, minimizing adverse reactions and enhancing compatibility with the human body.

Furthermore, by combining bacterial computing with MC, it is possible to a develop biological devices that can be controlled using external signals or retrieve information from their environment. This integration enables the creation of responsive living AI machines capable of precise control over therapeutic interventions,

real-time monitoring of physiological conditions, and advanced diagnostics. These machines will not only passively monitor or release molecules to environment, but also perform complex context aware computing tailored to specific applications. This capability positions them as a new class of AI-based living implantables. These advanced systems can dynamically interpret and respond to biological signals, making real-time decisionsFurther, these biological devices could be engineered to respond to specific molecular cues or environmental changes, providing a highly targeted and efficient approach to medical treatment and diagnostics.

## 4.2 Future Work

The new bacteria-based biocomputing avenue explored in this PhD thesis opens up several paths for future studies. This section lists three main potential research directions that can be taken forward into the future.

- **Heuristic application-specific GRNN sub-network search algorithms.** This thesis introduced a random permutation application-specific GRNN sub-network search algorithm, demonstrating the feasibility of extracting suitable GRNN sub-networks as a proof of concept. The algorithm successfully identified potential sub-networks, but the vast number of variables and their associated ranges make random searches impractical for large-scale applications. As a result, future research focused on heuristic search methods is essential to improve the efficiency and effectiveness of GRNN sub-network search. This advancement will significantly enhance the field of bacterial-based biocomputing, enabling more practical and scalable applications.

- **Distributed computing architectures with bacterial populations consist of multiple species.** This thesis began by exploring the role of MC in the behavior of bacterial ecosystems. This initial phase focused on how MC methods could model and predict interactions within bacterial communities,

providing insights into their complex dynamics. In subsequent phases, the thesis examined the impact of MC on the computing properties of GRNNs at the single-cell level, revealing how these networks operate within individual bacterial cells. The thesis then highlighted the computational diversity present within a single *E. coli* cell.

By combining insights from these analyses, the research suggests the potential for creating distributed computing systems. Such systems could not only replicate the same GRNN within a single species bacterial population but also utilize GRNNs from multiple species GRNNs working together with the help of MC. This approach would leverage the inherent variability and adaptability of bacterial systems, leading to more flexible and efficient biocomputing solutions. Further, this advancement represents a significant improvement in the field, opening new possibilities for the development of sophisticated, distributed biocomputing networks that can tackle complex computational tasks with greater efficiency and resilience.

- **Bio-hybrid approaches** The bacterial biocomputing approach explored in this thesis focused on considering the cell or the population as a complete computing platform. This method demonstrated the potential of bacterial systems to perform complex computational tasks by leveraging their natural processes and interactions. However, there is significant room for improvement by integrating silicon components and designing bio-hybrid systems.

  Embedding silicon into these bacterial systems could enhance their computational capabilities, making them more generalized and compatible with existing technologies. This integration would allow for a seamless interface between biological and traditional electronic computing systems, combining the strengths of both. Silicon components could provide precise control, faster processing speeds, and better scalability, while bacterial systems offer unique advantages

in adaptability and energy efficiency.

These hybrid systems can introduce hybrid intelligent biocomputers with energy efficiency and real-time adaptability, develop dynamic agricultural sensors for optimized resource use, and create wearable health monitors for continuous and precise health tracking. Overall, incorporating silicon into bacterial biocomputing systems represents a promising direction for future research, aiming to create advanced, integrated computing platforms that leverage the best of both biological and electronic worlds.

# Chapter 5

# Journal Paper: A Graph-based Molecular Communications Model Analysis of the Human Gut Bacteriome

| | |
|---|---|
| **Journal Title:** | IEEE Journal of Biomedical and Health Informatics |
| **Article Type:** | Regular Paper |
| **Complete Author List:** | Samitha S. Somathilaka, Daniel Perez Martins, Wiley Barton, Orla O'Sullivan, Paul Cotter, Sasitharan Balasubramaniam |
| **Keywords:** | Biological network systems, Graph analysis, Molecular communications, Human gut bacteriome, Metabolic interactions |
| **Status:** | Published: February 2022. doi: 10.1109/JBHI.2022.3148672 |

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



# A Graph-based Molecular Communications Model Analysis of the Human Gut Bacteriome

Samitha Somathilaka, *Student Member, IEEE,* Daniel P. Martins, *Member, IEEE,* Wiley Barton, Orla O'Sullivan, Paul D. Cotter, Sasitharan Balasubramaniam, *Senior Member, IEEE*

***Abstract*—Alterations in the human Gut Bacteriome (GB) can be associated with human health issues, such as type-2 diabetes and obesity. Both external and internal factors can drive changes in the composition and in interactions of the human GB, impacting negatively on the host cells. This paper focuses on the human GB metabolism and proposes a two-layer network system to investigate its dynamics. Furthermore, we develop an *in-silico* simulation model (virtual GB), allowing us to study the impact of the metabolite exchange through molecular communications in the human GB network system. Our results show that regulation of molecular inputs strongly affects bacterial population growth and creates an unbalanced network, as shown by shifts in the node weights based on the produced molecular signals. Additionally, we show that the metabolite molecular communication production is greatly affected when directly manipulating the composition of the human GB network in the virtual GB. These results indicate that our human GB interaction model can help to identify hidden behaviours of the human GB depending on molecular signal interactions. Moreover, the virtual GB can support the research and development of novel medical treatments based on the accurate control of bacterial population growth and exchange of metabolites.**

***Index Terms*—Biological network systems, graph analysis, molecular communications, human gut bacteriome, metabolic interactions.**

## I. Introduction

THE Gut Bacteriome (GB) is an ecosystem of a massive number of bacterial cells which play a vital role in maintaining the stability of the host's metabolism [1]. The bacterial populations of the GB build complex interaction networks by exchanging metabolites with the host and/or other bacterial populations [2], resulting in the production of new metabolites, such as Short Chain Fatty Acids (SCFAs), proteins, and other molecules [3].

External factors such as the availability of nutrients, antibiotics, and pathogens can affect this interaction network [4]. These factors mainly alter the compositional balance of the human GB, subsequently disrupting the metabolite production [5]. In humans, these GB changes have a significant impact on the host's health and may lead to many diseases, including inflammatory bowel disease, type-2 diabetes, obesity and cancers [6], [7]. Although studying complex causal metabolic networks is challenging [8], several studies have been undertaken to precisely identify the causes for microbial behavioural alterations and their consequent health effects in humans and animals [9], [10]. For example, Yang et al. [11] performed a cross-sectional whole-genome shotgun metagenomics analysis of the microbiome and proposed a combinatorial marker panel to demarcate microbiome-related major depressive disorders from a healthy microbiome. From a different perspective, Kim et al. introduced a split graph model to analyse the microbial compositions of healthy or Crohn's disease microbiome compositions [12]. Inspired by these works, we propose a novel tool to further characterise the interactions among the bacterial populations often found in the human GB.

In this paper, we propose a two-layer interaction model supported by the exchange of molecular signals, i.e. metabolites, to model the human GB. Here, we identify the interactions between bacterial cells as **Molecular Communications** (MC) systems and their collective behaviour as a MC network. MC aims to model the communication between biological components [13] using molecules as information [14], [15] and it is fundamental to characterise the exchange of metabolites in our two-layer interaction model.

In the graph network, bacterial populations act as nodes while the edges represent the interactions between them. This interpretation allows quantifying the behaviours of the human GB using graph theoretical incorporating MC analysis to understand impacts from distances between different graph states and variations of node/edge weights. Moreover, conducting *in-vivo* or *in-vitro* experiments on the human GB to extract data related to each interaction of the network often requires a significant number of resources and time. On the other hand, calculating them theoretically using Flux Balance Analysis (FBA) is extensively complex due to the large number of variables that prompt the same number of equations to be solved (see in Section IV for further details). On top of that, FBA is known as a static approach that fails to capture the stochastic nature of biological networks. Hence, we designed an agent-based simulator (henceforth named virtual GB) to simulate the human GB, which produces the same set of data that we expect by conducting *in-vivo* or *in-vitro* experiments or FBA calculations. The virtual GB performs the behaviours of the human GB considering natural characteristics. Hence, the generated data represents bacterial behaviours that are influenced by the aforementioned stochastic parameters.

S. Somathilaka, Daniel P. Martins are with VistaMilk Research Centre and the Walton Institute for Information and Communication Systems Science, Waterford Institute of Technology, Waterford, X91 P20H, Ireland. E-mail: samitha.somathilaka,daniel.martins@waltoninstitute.ie.

S. Balasubramaniam is with the Department of Computer , University of Nebraska-Lincoln, 104 Schorr Center, 1100 T Street, Lincoln, NE, 68588-0150, USA. E-mail:sasitharanb@gmail.com

Wiley Barton, Orla O'Sullivan and Paul D. Cotter are with VistaMilk Research Centre and the Teagasc, Food Research Centre, Moorepark, Ireland, P61 C996. E-mail: paul.cotter@teagasc.ie

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



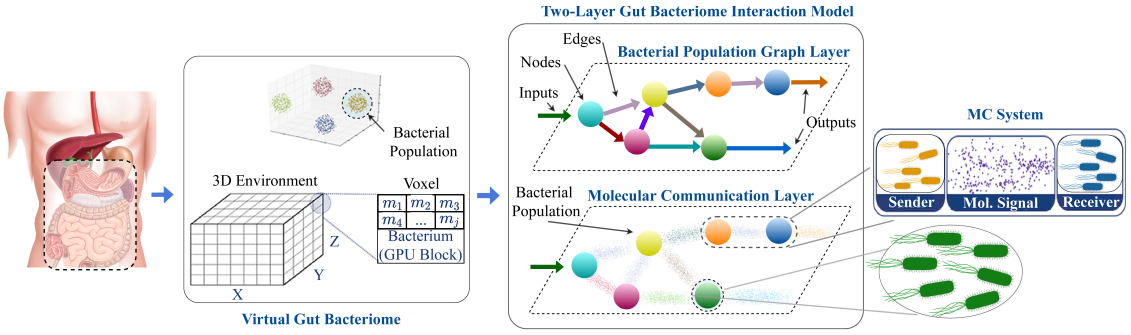



Fig. 1: Illustration of the system model. (a) We recreated the human GB functionalities on virtual GB using voxel architecture and parallel processing dedicating one GPU block for each bacterial cell to produce quantitative data on the MC layer, and (b) we propose a two-layer system model to investigate the molecular interactions simulated in the virtual GB.

Our main contributions are as follows:

- **Design of a two-layer interaction model of the human GB**: The collective gut bacteria metabolism forms a complex interaction network among the different bacterial populations. Hence, in this study, we design a layered interaction model to investigate the dynamics of the human GB based on the exchange of metabolites.
- **Analysing molecular communication impact on the human GB graph structure:** Deviations of bacterial populations' metabolism cause alterations in molecular interaction within the human GB, which may impact the graph layer structure. We analyse this relationship between the MC measures and the graph structure of the human GB in terms of graph nodes and edges behaviours.
- **Development of a human GB simulator to perform *in-silico* experiments:** We design and utilise an *in-silico simulation model* of the human GB to investigate the direct and hidden interactions among bacterial populations based on the exchange of metabolites.

In the next sections, we detail our approach to model the human GB and assess its network performance. In the section II, we describe the basics of the human GB and highlight the existent gaps that this research aims to address. Our proposed model is detailed in Section III. Then, the metrics considered in this paper are introduced in Section IV, and our analysis results are presented in Section V. Further, in Section V-A, we introduce the simulation environment built to utilise metagenomics data and perform *in-silico* experiments with the human GB. Finally, our conclusions are shown in Section VI.

## II. Background on the Human GB Model

The human GB is the bacterial ecosystem residing inside the human digestive system, comprising approximately 1000 species interacting with each other and carrying out crucial functions such as nutrient metabolism and immunomodulation of the host [16]. These bacteria utilise products of host metabolism, metabolites produced by other bacteria or dietary components from the gastrointestinal tract to convert into various products essential for the host through different metabolic pathways [17]. Bacteria in the human GB manifest their cellular functions by exhibiting various social behaviours such as commensalism [18], and competition by interacting with other populations mainly using molecules (e.g., proteins, metabolites and *quorum* sensing) rather than individual entities [19]. We identify these interactions as MC system and assert that the communication process in the GB is quite similar to routing and relaying information in a conventional network system which has inspired different network models (including ours) of the human GB interactions. For example, Naqvi et al. used a network-based approach to characterise the human gut microbiome composition and analysed healthy vs diseased states using network statistics [20]. Another study focuses on the use of Boolean dynamic models that combines genome-scale metabolic networks to determine the metabolic deviations between community members, which was used to characterise their metabolic roles of interactions [21].

The composition of the human GB is a crucial driver for the processing of metabolites (i.e., small molecules produced and used in metabolic reactions) in the lower intestine, which significantly impacts the health of the host [22]. Human GB composition differs among individuals, and it depends on various factors, including dietary patterns, gut diseases, exercise regimes, antibiotic usage, age, and genetic profiles [23].

## III. Two-Layer Human GB Interaction Model

In this paper, we represent the metabolic interactions of select representative bacterial genera of the human GB as a two-layer interaction model, as shown in Figure 1. First, the compositional and behavioural data on the human GB is extracted from the databases and literature and implemented the virtual GB (Figure 1a), see Section V-A for further details. The virtual GB then simulates the human GB functionalities according to various experimental setups (later explained in section V), producing data on bacterial, molecular and gut environmental behaviours. The produced data is analysed

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



according to the introduced two-layer interaction model as shown in Figure 1b.

The upper layer of this model, which is the bacterial population graph layer, defines the interconnections and overall structure of the human GB, where we model the bacterial populations and host as nodes and interactions between them as edges. To minimize the complexity of the model, this graph layer considers bacterial genera as nodes, as species in the same genus share a common ancestral origin and the data availability. Further, the edges of the network represent the direct connections between the nodes that produce a particular metabolite and the nodes that consume the corresponding metabolite. In this layer, we can investigate the network topology of the human GB.

The bottom layer consists of the cascading molecular communications systems created by the bacterial populations to establish their exchange of metabolites and support their network structure. Here, each node is viewed as a molecular transceiver, and the edges are the communications channels interconnecting the nodes. Furthermore, this model extends to the molecular signals that reach the human GB from the environment, as well as, the ones that are output from the human GB and return to the environment. The interactions represented in this layer are dynamic and will depend on several environmental conditions, such as media characteristics and human GB composition. Please note that this is the layer where we initially observe the impacts of any alterations on the human GB composition (we further model and analyse this effect in Section IV-A). The upper layer and the bottom layer are further described in the following sections.

### A. Upper Layer - Bacterial Population Graph Layer

Bacteria display a wide variety of social behaviours, and this can lead to processes such as the metabolism of molecules or coordinated biofilm formation [24]. The bacteria's ability to consume and produce multiple metabolites results in dense interaction patterns that can lead to challenges in the analysis.

Our human GB interaction model aims to provide a better global view of the functionality of the human GB, leading to the understanding of the causes and effects of its imbalance and to propose precise alterations to fix such issues. Therefore, we model the human GB as follows. We first consider that all bacterial cells $b^{B_k}$ of a bacterial population $B_k$ (where $k$ is the bacterial population identifier) perform the same series of metabolic functions to process the metabolite inputs in the human gut. Each node of the proposed graph layer is a bacterial population, and each edge is an interaction between two bacterial populations through metabolite exchange. The nodes of this layer comprise the collective metabolic functions of all cells within the corresponding population. Let $\Omega$ be the set of all agents in this study, i.e. host cells and bacterial populations, $\Omega = \{host, B_k\}$. In this case, the molecular intake of particular bacterial population $B_{k'}$ from $\Omega$, $C_{(\Omega,B_{k'})}$ is considered $C_{(\Omega,B_{k'})} \simeq \sum c_{(\Omega,b^{B_{k'}})}$ where $C$ represents population interactions, $c$ represents the intercellular interactions and $c_{(\Omega,b^{B_{k'}})}$ is the molecular reception of bacterial cell $b^{B_{k'}}$ (a cell from the bacterial population $B_{k'}$) from a $\Omega$ source. In the same way, molecular emission of the population is considered the combined molecular emission of all bacterial cells of the particular population, $C_{(B_{k'},\Omega)} \simeq \sum c_{(b^{B_{k'}},\Omega)}$, where $C_{(B_{k'},\Omega)}$ is the molecular emission from population $B_{k'}$ to any receiver (host or other bacterial populations), and $c_{(b^{B_{k'}},\Omega)}$ is the molecular emission of a single bacterial cell of the population $B_{k'}$ to any receiver. Additionally, the metabolite consumed by the bacterial cell $b^{B_{k'}}$, $M_{Con}(b^{B_{k'}})$ is obtained as $M_{Con}(b^{B_{k'}}) = c_{(\Omega,b^{B'_k})} - c_{(b^{B'_k},\Omega)}$. Hence the metabolite consumption of a bacterial population is defined as $M_{Con}(B_{k'}) \simeq \sum M_{Con}(b^{B_{k'}})$.

Next, we map the interactions between bacterial populations to a directed multi-graph network, $\Gamma = (B, C, B^s, B^d, M)$, where $B$ is the set of all bacterial populations, $C$ is the set of all interaction in the human GB, $B^s \in B$ is the bacterial population interaction sources, $B^d \in B$ is the bacterial population interaction destinations, and $M$ is the set of metabolites. In this work, we consider SCFAs production as the use case for our model on the bacterial population interactions.

### B. Bottom Layer - MC System

As detailed in the previous section, the metabolism of nutrients by the human GB involves the reception, processing, production of metabolites. These activities are fundamental for the maintenance of the human GB, and this is modelled as the MC layer shown in Figure 1b. Our aim of having the two-layer model is to determine how the changes due to molecular signals of the metabolites will affect the relationship of the bacterial population graph layer. Therefore, any changes in the bottom layer directly affect the upper layer and vice-versa.

Here, we define the metabolites as the molecular signals that are exchanged by the nodes, which can assume different functions depending on the MC network structure. For example, when the node receives molecular signals, we model it as a receiver, and when processing and secreting molecular signals, we define them as transmitters based on the MC paradigm. The edges of the proposed MC network are represented as the MC channels to model the physical transport of molecular signals between the nodes by diffusion. Figure 1b shows a visual representation of the proposed bottom layer and its relationship with the upper layer.

The diffused molecular signal is received by the nodes which have the membrane receptors that will allow the metabolites to bind. The performance of this network node function (i.e., molecular reception) relies on many factors such as molecule size [25], ligand-receptor maximum attraction length and bond equilibrium [26], binding noise due to the Brownian motion of molecules near the receptors [27], and the minimum required concentration to be detected [28]. After receiving the molecular signals, the node will process them internally, which may result in the production of a new molecular signal to be transmitted to the next node (focus of this paper).

Received molecular signals are processed through signalling pathways and produce different metabolites that will be transmitted to the next node [29]. Even though we only focus on the genus level, the signal processing occurs in each

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



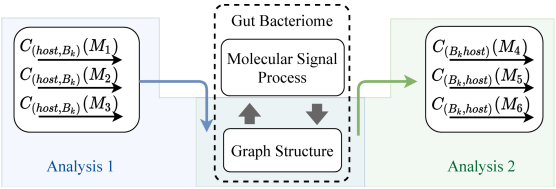



Fig. 2: Illustration of the analysis structure of this study. Analysis 1: Influence of inputs on the graph structure and Analysis 2: Behaviours of graph output against structural deviations.

bacterial cell. Accordingly, we present the signalling process performance of a bacterial cell $b^{B_k}$ related to any metabolite $M_j$ as $SPP_{b^{B_k}}(M_j)$. Let's assume that the cell $b^{B_k}$ produces $M_j$ by consuming another metabolite $M_{j'}$. Then the signal process performance $SPP_{b^{B_k}}(M_j)$ can be modelled by considering the metabolite $M_{j'}$ reception process (defined as $R_{b^{B_k}}(M_{j'})$), the encoding/decoding process from metabolite $M_{j'}$ to $M_j$ (defined as $E_{b^{B_k}}(M_{j'}, M_j)$), and $M_j$ metabolite secretion process by the cells in the bacterial population $b^{B_k}$ (defined as $L_{b^{B_k}}(M_j)$). Hence, we represent the signal process performance as follows,

$$SPP_{b^{B_k}}(M_j) = f(R_{b^{B_k}}(M_{j'}), E_{b^{B_k}}(M_{j'}, M_j), L_{b^{B_k}}(M_j)). \tag{1}$$

Therefore, the SPP of the populations $B_k$ can be modeled as follows,

$$SPP_{B_k}(M_j) = \sum SPP_{b^{B_k}}(M_j). \tag{2}$$

Since the output of the molecular signal processing is the emission of a particular molecular signal, it is fair to say,

$$SPP_{B_k}(M_j) = C^r_{(B_k,\Omega)}(M_j) \tag{3}$$

where $C^r_{(B_k,\Omega)}(M_j)$ is the rate of molecule $M_j$ production by the bacterial population $B_k$ to any node (either other bacterial populations or host cells).

## IV. System Dynamics

We investigate the system dynamics of the human GB through a series of simulations using the virtual GB and propose a two-layer human GB model. First, we recreate the digital form of the human GB on the simulator, which is explained in depth later in Section V-A. Then we perform two main sets of experiments, as depicted in Figure 2. In the first set, we analyse the impact of the system's inputs on the connectivity structure of the virtual GB, and in the second set, we manipulate the composition of our virtual GB to investigate the impact on the metabolite production of our MC network. Through this second set of experiments, we aim to identify the nodes that can play a pivoting role in the GB imbalances.

In our analyses, first we define a standard graph state $S_0$, which represents the functionality of an average healthy human GB with the intention of quantifying structural changes and behavioural deviations relative to the standard structures. The average composition, interactions, and metabolite production dynamics were mainly considered in defining the $S_0$. The average composition and the interactions of $S_0$ for the case study of this paper is presented in Section V

### A. Molecular input impact on the human GB structure

Due to the variety of bacterial behaviours induced by the exchange of molecules, some of the molecular input signals have a significant impact on the structure of the human GB (our focus), while others are directly converted into output metabolites. In this section, we detail how the molecular input signals impact the structure of our MC network. As the structural deviations of the graph is a crucial measurement in understanding the deviation of the human GB behaviour, the structural deviation is evaluated in terms of edges and nodes weight using the rates of the interaction of the nodes. Hence, we explain how the interaction rates can be calculated theoretically using FBA and are represented as follows,

$$F_{[k\times q]} \cdot \vec{C} = \vec{M}_{Con}(B_k) \tag{4}$$

where $F_{[k\times q]}$ is the stoichiometric matrix of $k$ number of bacterial populations and $q$ number of interactions based on the flux of metabolites between the nodes in the MC network. Here $\vec{C} = [C^r_1, C^r_2, ..., C^r_q]_{1\times q}$ and $C^r_q$ is the rate of interactions for $C_q$. We can solve (4) as follows,

$$\begin{matrix} B_1 \\ B_2 \\ \vdots \\ B_k \end{matrix} \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,q} \\ a_{2,1} & a_{2,2} & \dots & a_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1} & a_{k,2} & \dots & a_{k,q} \end{pmatrix} \cdot \begin{pmatrix} C^r_1 \\ C^r_2 \\ \vdots \\ C^r_q \end{pmatrix} = \begin{pmatrix} \frac{\mathrm{d}M_{Con}(B_1)}{\mathrm{d}t} \\ \frac{\mathrm{d}M_{Con}(B_2)}{\mathrm{d}t} \\ \vdots \\ \frac{\mathrm{d}M_{Con}(B_k)}{\mathrm{d}t} \end{pmatrix} \tag{5}$$

where, $a_{k,q}$ is the stoichiometry of the interaction $C^r_q$ for bacterial population $B_k$.

Based on (5), we can extract the relationship between rates of interactions starting from the node $B_k$ using Mass Balance Equation (MBE), which is based on the following relationship

$$\frac{\mathrm{d}M_{Con}(B_k)}{\mathrm{d}t} = \sum_q a_{(k,q)} C^r_q. \tag{6}$$

On the other hand, the rate of molecular consumption can be modeled as follows [30],

$$\frac{\mathrm{d}M_{Con}(B_k)}{\mathrm{d}t} = -U_1 \left( \mu_k \frac{M_{Con}(B_k)}{M_{Con}(B_k) + K_{S1}} \right) N_{B_k} \tag{7}$$

where $N_{B_k}$ is the bacterial concentration, $\mu_k$ is maximum growth rate, $K_{S1}$ is the half-saturation constant of the bacteria, and $U_1$ is an utility parameter. Hence,

$$-U_1 \left( \mu_k \frac{M_{Con}(B_k)}{M_{Con}(B_k) + K_{S1}} \right) N_{B_k} = \sum_q a_{(k,q)} C^r_q. \tag{8}$$

By solving the series of MBEs, all the interaction rates can be calculated. This is a highly complex calculation due to the massive number of nodes, edges of the network, and a large number of parameters associated with the structural connections. The introduced virtual GB produces data on the rates of interactions avoiding complex FBA calculations.

The extracted rates of interactions are then used to quantify the graph structural changes in two ways. First, we investigate

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



the graph structural changes considering the behaviours of the node weights. Here, the statistical distances between the weights of the same node in different graph states are measured. The node weight, $B_k^w(S_g)$ of the bacterial population $B_k$ in the graph state $S_g$ is considered as the collective $SPP$ and can be evaluated as follows,

$$B_k^w(S_g) = \sum_j SPP_{B_k}(M_j). \tag{9}$$

Alternatively, using (3) we compute the node weight as follows,

$$B_k^w(S_g) = \sum_j C^r_{(B_k,\Omega)}(M_j). \tag{10}$$

Based on this, $d(B_k^w : S_g, S_0)$ represents the distance of node $B_k$ between the two graph states $S_0$ and $S_g$ is evaluated as follows,

$$d(B_k^w : S_g, S_0) = B_k^w(S_g) - B_k^w(S_0). \tag{11}$$

Next, we quantify the structural deviation of the graph using the interaction changes. In this study, we consider static snapshots of different graph states that can enable the use of the *Hamming Distance* to evaluate graphical distances for two states [31] among other techniques. The Hamming distance $d_h(S_0, S_g)$ between the graph states $S_g$ and the standard state $S_0$ is defined as the difference of two adjacent matrices corresponding to the two graph states. First, we define the adjacency matrix of the graph state $S_g$ as follows,

$$\begin{array}{c} \\ B_1 \\ B_2 \\ \vdots \\ B_k \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & \dots & B_k \end{array} \\ \begin{pmatrix} C^w_{(B_1,B_1)} & C^w_{(B_1,B_2)} & \dots & C^w_{(B_1,B_k)} \\ C^w_{(B_2,B_1)} & C^w_{(B_2,B_2)} & \dots & C^w_{(B_2,B_k)} \\ \vdots & \vdots & \ddots & \vdots \\ C^w_{(B_k,B_1)} & C^w_{(B_k,B_2)} & \dots & C^w_{(B_k,B_k)} \end{pmatrix} \end{array} \tag{12}$$

where $C^w_{(*,*)}$ is the weight of the interaction $C_{(*,*)}$. Note that the weights of interactions in the main diagonal of the above matrix represents the interactions that take place within the same bacterial population, which is a type of interaction that cannot be observed in the metabolic network we considered in this study. Further, we define the weight of the interaction $C_{(B_k,B_{k'})}(M_j)$ between any bacterial population $B_k$ and $B'_k$ through metabolite $M_j$ as follows,

$$C^w_{(B_k,B_{k'})}(M_j) = \frac{C^r_{(B_k,\Omega)}(M_j)}{\sum_k C^r_{(B_k,\Omega)}(M_j)} \cdot \frac{C^r_{(\Omega,B_{k'})}(M_j)}{\sum_{k'} C^r_{(\Omega,B_{k'})}(M_j)}. \tag{13}$$

Moreover, from released molecules by a bacterial population, only a fraction is consumed directly by the other populations and the rest will get accumulated in the environment. This means the most significant portion of molecular consumption by the bacterial populations is from the environment. We define this process with the help of a memory component concept as depicted in Figure 3. Since the metabolites are accumulated in the environment, we consider it a memory, then model the metabolite accumulation as an interaction starting from a bacterial population that releases the metabolites and ending with the memory, $C_{(B_k,Mem)}$. In the same way, the metabolite consumption from the environment is modelled as an interaction starting from the memory and ending with a bacterial population that consumes the particular metabolite, $C_{(Mem,B_k)}$. Hence, we modify (13) by applying the memory, which is represented as follows,

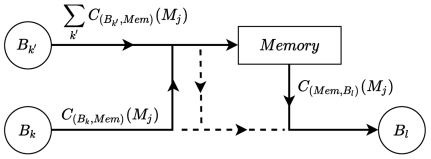



Fig. 3: Illustration of the environment working as a memory of molecules.

$$C^w_{(B_k,B_{k'})}(M_j) = \frac{C^r_{(B_k,Mem)}(M_j) C^r_{(Mem,B'_k)}(M_j)}{\sum_k C^r_{(B_k,Mem)}(M_j) \sum_{k'} C^r_{(Mem,B'_k)}(M_j)}. \tag{14}$$

Then, the Hamming distance, $d_h(S_0, S_g)$ can be represented as,

$$d_h(S_0, S_g) = \sum_{k,k'} |C^w_{(B_k,B_{k'})}(S_g) - C^w_{(B_k,B_{k'})}(S_0)| \tag{15}$$

where, $C^w_{(B_k,B_{k'})}(S_g)$ and $C^w_{(B_k,B_{k'})}(S_0)$ are the weights of interaction $C_{(B_k,B_{k'})}$ in graph states $S_g$ and $S_0$ respectively.

### *B. Human GB structure impact on the molecular output*

This analysis explores the impact of interaction variations of the human GB on the output. Here, we keep the inputs at an optimal level and manually alter the graph structure by changing the population sizes, which leads to variations in the $SPP$ of the nodes. Then the output of the system is measured in different graph states and the weights of the edges are calculated using (13) to determine the molecular output of the MC layer using graph theory.

The ratio between the three SCFAs can be identified as a critical measurement to evaluate the metabolite production accuracy of the bacteriome. We adopt the signal to noise ratio (SNR) to evaluate the consistency of the output signal ratios. In this analysis, we calculate SNR of any signal $SNR(M_j)$, considering the other output signals, $M_{j'}$ as noise. This SNR value directly indicates the ratio between the molecular signal $M_j$ and other metabolite signals $M_{j'}$. Then $SNR(M_j)$ is calculated as follows,

$$SNR(M_j) = \sum_k \frac{C_{(B_k,host)}(M_j)}{\sum_{j'} C_{(B_k,host)}(M_{j'})}. \tag{16}$$

Moreover, some bacterial populations do not produce specific SCFAs, but have an indirect influence on them. For example, *Bacteroides* cells do not produce butyrate, but the acetate produced by the *Bacteroides* cells is a substrate for the butyrate production by *Faecalibacterium* and *Roseburia* cells. Hence, the *Bacteroides* population indirectly influences the butyrate production. Considering the above mentioned effect,

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



a correlation matrix is generated for variation of node weights vs the collective SCFA output of the human GB to analyse the impact of various bacterial populations in SCFA production. Here, we denote the rate of SCFA $M_j$ output by all the bacterial populations as $O^r(M_j)$ where

$$O^r(M_j) = \sum_k C^r_{(B_k,host)}(M_j). \tag{17}$$

Then, the correlation coefficient $r(B_k)$ of node weight $B^w_k$ versus the collective output of $M_j$ is calculated as follows.

$$r(B_k) = \sum_g \frac{\overline{B^w_k} - B^w_k(S_g)(\overline{O^r}(M_j) - O^r(M_j))}{\sqrt{(\overline{B^w_k} - B^w_k(S_g))^2}\sqrt{(\overline{O^r}(M_j) - O^r(M_j))^2}} \tag{18}$$

where, $\overline{O^r}(M_j)$ is the standard collective output rate for $M_j$ by all the bacterial populations and $\overline{B^w_k}$ is the weight of the node $B_k$ in the standard state $S_0$.

## V. Analytical Results

In this section, we describe the development of the virtual GB and the results from our analysis that is based on the models presented in Section IV.

### A. Virtual GB Design

We developed the virtual GB using metagenomic data to characterise the bacterial populations signalling interactions and their impact on the network relationships. The virtual GB is inspired by the BSim agent-based cell simulator [32]. The virtual GB is written in C++ with CUDA platform for parallel processing to increase the simulation performance and most importantly, mimic the parallel processing typically executed by the bacterial populations. We dedicate one GPU block for each bacterial cell, and the threads of that block to intracellular functions of the corresponding cell. To simulate the bacterial interactions, we model the exchange of molecules using metabolic flux in a diffusive media. The simulator has a 3D environment with voxel architecture (Figure 1a), which provides the ability of extracting data on each metabolite and bacterial cell separately. Moreover, we can introduce any new cell type by creating their internal metabolic pathways and other physiological characteristics such as motility, shape, size, etc. Therefore, the simulator can be used for a range of setups including other metabolic functions, microbial ecosystems in different habitats or targeting specific bacterial behaviour like quorum sensing. Further, the simulator can log data on the metabolite consumption/production/accumulation and bacterial proliferation. In this study, we setup the virtual GB to simulate the SCFA production using metagenomic and metabolomic data obtained in [29], KEGG [33]–[35], NJS16 [36], and MetaCyc databases [37].

Here, we present a series of analyses conducted on SCFA production within the human GB using the two-layer model. First, we defined the average composition of the human GB using the average relative abundance (RA) (see Table I) calculated based on data extracted from 352 samples of the MicrobiomeDB [38].

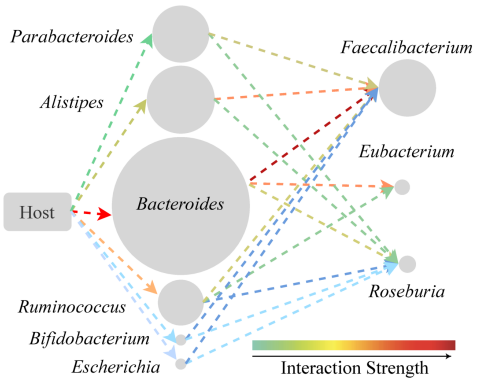



Fig. 4: Representation of the subgraph, $\Gamma_{SCFA}$ considered in the case study which contains the nodes and edges related to SCFA production.

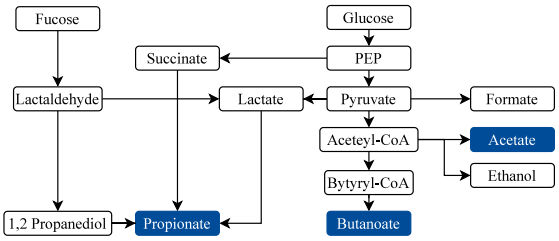



Fig. 5: Combined and simplified SCFA production pathway of converting fucose and glucose, into SCFAs.

TABLE I: Average RAs of bacterial populations

| Genus | Average RA |
|---|---|
| *Bacteroides* | 0.4899173 |
| *Alistipes* | 0.05960802 |
| *Faecalibacterium* | 0.04329791 |
| *Parabacteroides* | 0.04096428 |
| *Ruminococcus* | 0.03320183 |
| *Roseburia* | 0.01039938 |
| *Eubacterium* | 0.0093219 |
| *Bifidobacterium* | 0.00179366 |
| *Escherichia* | 0.00185639 |

Using these RA data along with the extracted interaction data from the databases mentioned earlier, we created a graph network for SCFA production, $\Gamma_{SCFA}$ following the definitions presented in Section III-A. We only considered nine bacterial populations based on their RA, their metabolic activities, and data availability. We include *Bacteroides*, *Alistipes*, *Faecalibacterium*, *Parabacteroides*, and *Ruminococcus* in the model as they are the most abundant bacterial genera. To add further metabolic diversity to the network, we include other bacterial genera used in this study as they perform different metabolic functions compared to the most abundant bacterial genera. Figure 4 illustrates the $\Gamma_{SCFA}$ where node sizes indicate the RAs of the respective bacterial genera shown in Table I. Furthermore, the edges are colour-coded to highlight the strengths of the interactions which are quantified using

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



TABLE II: Parameters utilised in Section V-B and V-C.

| Parameter | Value | Description |
|---|---|---|
| **Standard setup** | | |
| *Bacteroides* cell count | $7.3\times10^5$ | |
| *Alistipes* cell count | $8.9\times10^4$ | |
| *Faecalibacterium* cell count | $6.4\times10^4$ | |
| *Parabacteroides* cell count | $6.1\times10^4$ | Calculated based on the RA (see Table I) and to keep the total cell count |
| *Ruminococcus* cell count | $4.9\times10^4$ | lesser than $1.5 \times 10^6$ (maximum number of cells was limited by the memory |
| *Roseburia* cell count | $1.5\times10^4$ | availability of our GPU server). |
| *Eubacterium* cell count | $1.3\times10^4$ | |
| *Bifidobacterium* cell count | $1.5\times10^3$ | |
| *Escherichia* cell count | $1.5\times10^3$ | |
| Maximum GPU blocks utilised | $1.7\times10^6$ | Calculated based on number of voxels. |
| Maximum threads per utilised per block | 24 | Calculated based on number of metabolites used in the simulations. |
| **Analysis 1: Molecular input impact on the human GB structure** | | |
| Glucose input rate (min-max) | 0.000-16.605 $\mu$mol/m$^3$s | Calculated the using the number of cells and the stoichiometry of metabolic pathways [37], [39]. |
| All bacterial population cell count | Fixed at standard values | Values are same as the standard setup. |
| **Analysis 2: Human GB structure effect on the graph outputs** | | |
| Glucose input rate | 6.642 $\mu$mol/m$^3$s (Fixed) | Obtained from simulations to match SCFA production of average human GB. |
| ***Bacteroides* setup** | | |
| *Bacteroides* cell count (min-max) | 0-$1.6\times10^6$ | Calculated using the stoichiometry of metabolic pathways of *Bacteroides* and the number of cells to obtain results range with significant changes [37]. |
| Other bacterial population counts | Fixed at standard values | Values are same as the standard setup. |
| ***Faecalibacterium* setup** | | |
| *Faecalibacterium* cell count (min-max) | 0-$1.4\times10^5$ | Calculated using the stoichiometry of metabolic pathways of *Faecalibacterium* and the number of cells to obtain results range with significant changes [37]. |
| Other bacterial population counts | Fixed at standard values | Values are same as the standard setup. |

(13).

For illustration purposes, we combine the metabolic processes executed on different bacterial cells and simplify the SCFA pathway to focus on the most important steps that leads to the production of the three most abundant SCFAs in the human GB, namely acetate, butyrate and propionate (see Figure 5) [40]. The parameters utilised in V-B and V-C are presented in Table II. As we explained earlier, the bacterial cell counts for the standard setup are calculated based on the calculated RA and to keep the total cell count less than $1.5\times10^6$. The number of GPU blocks equals to the number of voxels in the system and the maximum number of threads per block calculated based on the number of metabolites in the environment. Further, the glucose input rate is extracted by an array of iterative experiments to match the ratio of SCFA abundance of an average human GB. Please note that in a typical human GB, SCFA abundance ratios range from 3:1:1 to 10:2:1 [41]. The maximum glucose input rate of the Analysis 1, and the maximum *Bacteroides* and *Faecalibacterium* cell counts of the Analysis 2 are fixed at certain values to obtain results with significant behaviours. Beyond those maximum values, the results only continue the trends without significant changes.

### *B. Analysis 1: Molecular input effects on the graph structure*

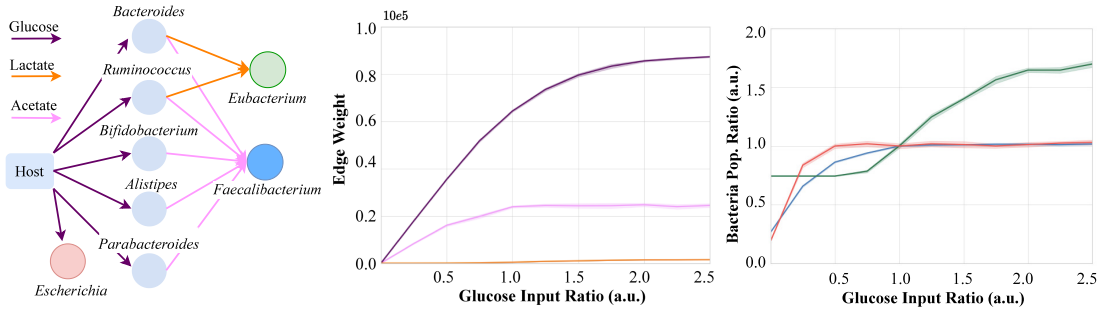
Here, we present the results for the analyses mentioned in Section IV-A. The analyses are conducted by regulating the input glucose rate $C^r_{(host,Mem)}(M_{glu})$ and fucose rate $C^r_{(host,Mem)}(M_{fse})$ from the host cells to the system that contains the memory of existing metabolites and evaluating the human GB compositional changes. The simulation for these experiments only contains growth dynamics of *Faecalibacterium*, *Eubacterium* and *Escherichia* bacteria as their growths are supported by the metabolites involved in the same SCFA production. Further, with the data availability, the model can be extended to analyse the growth dynamics of other bacterial genera as well.

Figure 6 illustrates the impact of glucose on the three bacterial populations based on $\Gamma_{glu}$ ($\Gamma_{glu} \subseteq \Gamma_{SCFA}$), shown in Figure 6a. The colours used in Figures 6b and 6c follow the same colour scheme as in Figure 6a. Figures 6b and 6c shows the behaviours of edge weight and variation of population sizes as a fraction of that in $S_0$ due to the changes in $C^r_{(host,Mem)}(M_{glu})$ respectively. The variations of the input rate $C^r_{(host,Mem)}(M_{glu})$ alters the intermediate interaction from any bacterial population $B_k$ to other population $B_{k'}$ through acetate, $C_{(B_k,B_{k'})}(M_{ace})$ and lactate $C_{(B_k,B_{k'})}(M_{lact})$, which are required for the growth of *Faecalibacterium* and *Eubacterium*, respectively. Figure 6b explains the graph theoretical behaviour of indirect influence on the growth dynamics of the respective bacterial populations. The growth of *Eubacterium* keeps increasing steadily until the $C^r_{(host,Mem)}(M_{glu})$ is twice the standard level, while the growths of the other two bacterial populations converge to the standard static level. This is due to the stoichiometry of the metabolite conversion, where an acetate molecule is produced by one glucose molecule while a lactate molecule requires two. The growth of *Escherichia* and *Faecalibacterium* are directly altered by the variations of glucose inputs and the behaviours. We calculated the maximum Mean Standard Error (MSE) as 0.03374 for any metabolite by iterating the experiment 20 times.

Deviations of a bacterial population concentration refer to deviations in node weights according to the (2) and (9). Figure 7 represents the node weight deviation compared to standard graph state $S_0$ due to the variability in inputs. This

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672






(a) (b) (c)

Fig. 6: Deviation of population sizes of *Faecalibacterium*, *Eubacterium* and Escherichia from the standard levels due to different input concentrations of glucose: (a) subgraph for the glucose consumption, (b) edge weight behaviours of the intermediate interactions, and (c) population growth behaviours.

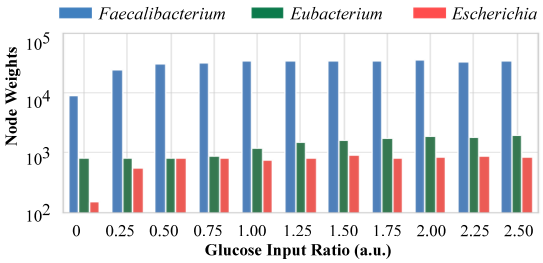



Fig. 7: Changes of node weights due to the variations in molecular signal inputs.

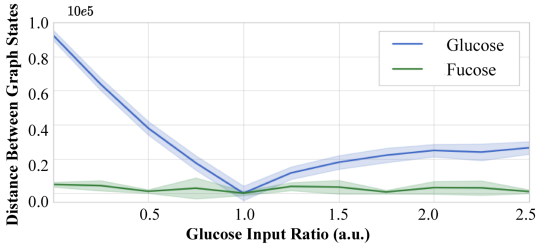



Fig. 8: Behaviours of overall graph weights against the changes in inputs and their concentrations.

analysis reveals the impact of different input conditions on the molecular signal performance $SPP$ of bacterial populations.

While Figure 7 explains the node weight variations, Figure 8 focuses on the overall interaction weight behaviours compared to $S_0$. This graph provides an insight into how the structure is being modified by the input variability. When the $C^r_{(host,Mem)}(M_{glu})$ is low compared to the standard level, the graph deviates significantly from the standard level, and when the $C^r_{(host,Mem)}(M_{glu})$ exceeds the standard level, the graph starts to deviate again from the standard structure, but with a lesser magnitude compared to a weaker signal (the standard level is 1.0). This reveals that the human GB is more sensitive to low glucose concentrations. The experiment is repeated for the fucose input rates $C^r_{(host,Mem)}(M_{fse})$ as well, but the impact is minimal compared to $C^r_{(host,Mem)}(M_{glu})$.

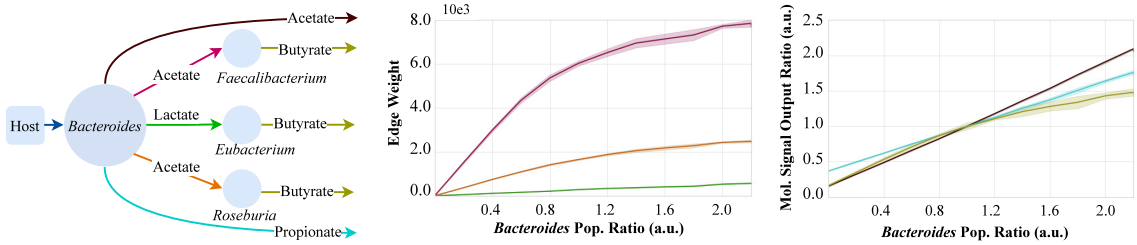
### *C. Analysis 2: Human GB structure effect on the graph outputs*

In this section, we analyse the direct and indirect impacts of the human GB compositional changes on the network behaviours. The analyses are conducted by altering the bacterial population sizes manually on the virtual GB and extracting the metabolite production data with respect to each alteration. The resulting behaviours of the MC layer are explained using the graph analyses. Although we conduct similar experiments for all the nine populations, we only show results on *Bacteroides* (Figure 9) and *Faecalibacterium* (Figure 10) populations as they provide a better understanding of the metabolite production dynamics of the human GB.

Figure 9 shows the impact of *Bacteroides* population size variation on the human GB SCFA production. In this experiment, we focus on the graph $\Gamma_{Bct}$ ($\Gamma_{Bct} \subseteq \Gamma_{SCFA}$) considering only the interactions that are related to the *Bacteroides* population, as shown in Figure 9a. The colour scheme used in Figures 9b and 9c follow the same colour scheme as in Figure 9a. The metabolite inputs to the graph and the population sizes are maintained fixed at the standard level except for the *Bacteroides* population size. We modify the population size of *Bacteroides* ($|B_{Bct}|$) from zero cells to 2.2 times the standard population size. Figure 9b explains the behaviours of the intermediate links from *Bacteroides* to *Faecalibacterium* node through acetate, *Bacteroides* to *Eubacterium* populations through lactate, and *Bacteroides* to *Roseburia* populations through acetate, while Figure 9c shows SCFA production behaviours in the MC layer due to changes in the population size. From Figure 9c, it is evident that all the SCFAs have strong positive relationships with the population size of *Bacteroides*. Acetate and propionate are direct products of *Bacteroides* cells. As a result of that, acetate and propionate outputs show steady trends against the increase of *Bacteroides* population sizes. Moreover, the edge weight

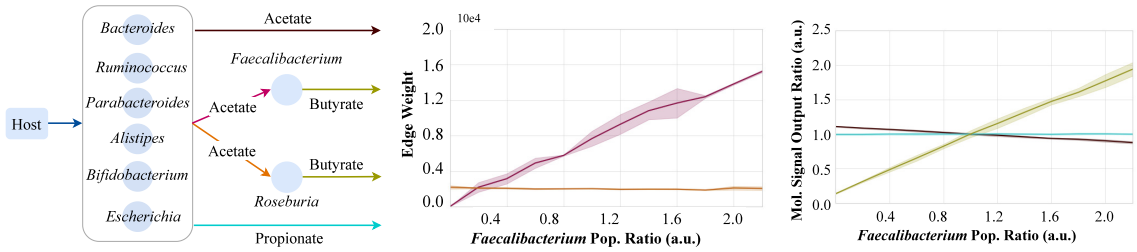
This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672





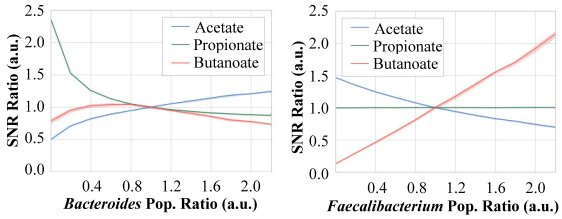

(a) (b) (c)

Fig. 9: Behaviours of SCFA production for various in *Bacteroides* population sizes: (a) subgraph of Bacteroides population interactions, (b) edge weight behaviours, and (c) SCFA output.




(a) (b) (c)

Fig. 10: Behaviours of SCFA production for various in *Faecalibacterium* population sizes: (a) subgraph related to the interactions of *Faecalibacterium* population, (b) edge weight behaviours, and (c) SCFA output.




(a) (b)

Fig. 11: Simulation results for SNR of three output signals with the changes in population sizes of *Bacteroides* and Feacalibacterium. (a) SNR results for *Bacteroides* population and (b) SNR results for Feacalibacterium population.

variations are shown in Figure 9b justify the butyrate signal behaviour in the MC layer shown in Figure 9c. To be more precise, the butyrate output curve starts to become flat when the *Bacteroides* population size $|B_{Bct}|$ is greater than 0.8 times the standard value. The graph theoretical quantification of links also shows the same trend in Figure 9b, emphasizing that the graph theoretical measures can be used to explain the metabolite production behaviours.

In the same way, Figure 10 illustrates the results for a similar experiment on *Faecalibacterium* population. Figures 10a, 10b and 10c represent the subgraph $\Gamma_{Fae}$ ($\Gamma_{Fae} \subseteq \Gamma_{SCFA}$), edge and the MC layer behaviours, respectively. Similarly to the previous analysis, we modify the population size of *Faecalibacterium* $|B_{Fae}|$ ranging from zero cells to 2.2 times the standard population size. As the *Faecalibacterium* cells consume acetate and produce butyrate, the rate of acetate consumption from the environment increases when the $|B_{Fae}|$ is increased. Hence, the weight of interaction between environment and *Faecalibacterium* population increases, which can be observed in Figure 10b, and the resulting reduction in acetate output is visible in Figure 10c. Moreover, since *Faecalibacterium* population is one of the key butyrate producers, there is a clear positive relationship evident between $|B_{Fae}|$ and butyrate. Due to the smaller population size of the *Roseburia* population, the influence on the metabolite production is relatively low, which can be observed from Figure 10b. For all the graphs, the maximum MSEs are calculated below 0.03087.

The MC layer results presented for the two analyses on *Bacteroides* and *Faecalibacterium* populations (Figures 9c and 10c) are then interpreted in terms of SNR in Figure 11. In the plots of this figure, SNR values are shown as ratios of the SNR value at the standard state of the human GB, and the bacterial population sizes are increased similar to the previous analyses. Here, we show the three SNRs of acetate, propionate, and butyrate of two bacterial populations: *Bacteroides* and *Faecalibacterium*. Figure 11a shows the SNR behaviours of the three SCFAs against the $|B_{Bct}|$. It is clearly evident that the acetate production is higher compared to the other two SCFAs when the $|B_{Bct}|$ is increased. This means, when the composition of human GB is changed as the $|B_{Bct}|$

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672



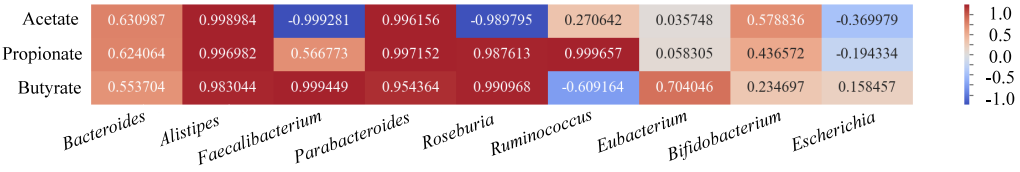



Fig. 12: Pearson correlation heat map of the impact on the three output signals by nine bacterial populations.

increases, the output of the GB also loses balance and tends to produce more acetate compared to the other two SCFAs. On the contrary, the propionate production rate reduces when the $|B_{Bct}|$ increases. When the population size of *Bacteroides* $|B_{Bct}|$ is smaller than the standard level, the system tends to produce molecular signal with higher deviated ratios, but when $|B_{Bct}|$ is greater than the standard level, the deviation is relatively low. Figure 11b shows the SNR behaviours of the three SCFAs against the $|B_{Fae}|$. Since *Faecalibacterium* is the main butyrate producer of this network, the butyrate SNR increases with the $|B_{Fae}|$ increment. Hence, compositional imbalance related to *Faecalibacterium* causes a significant imbalance in output molecular signal ratios. Furthermore, due to the acetate consumption of *Faecalibacterium*, the acetate signal becomes weaker, resulting in the acetate SNR deviating from the standard level.

Figure 12 explains the correlation between each bacterial population and the SCFA abundance in the gut environment. Although *Bacteroides* are the biggest producer of all the SCFAs, it has a weak correlation with SCFAs compared to other producers such as *Alistipes* and *Parabacteroides*. This reveals that the reduction of glucose consumption by *Bacteroides* increases the other bacterial population, resulting in boosted SCFA production. Note that, even the SCFA production of the other bacterial population is boosted in the absence of *Bacteroides*, the overall production is low. Since the *Faecalibacterium* and *Roseburia* consume acetate, the heat map shows a strong negative correlation with acetate. Interestingly, this heat map indicates metabolic switching by *Escherichia*, from a SCFA producer to a high acetate concentration consumer. This is the same for the *Ruminococcus* when the fucose concentration is not sufficient for the increased population, it switches from fucose to glucose consumption reducing the intermediate metabolite production, which causes a reduction in butyrate production.

## VI. Conclusion

The gut bacteriome has been largely investigated due to its importance to the human health. We contribute to this research topic by introducing a two-layer GB interaction model to investigate the impacts of bacterial population compositional changes on the overall structure of the human GB utilising data collected from MicrobiomeDB and NJS16 databases. Our proposed human GB interaction model combines a bacterial population graph layer, which models the structure typically found in the human GB (i.e. bacterial populations genus and sizes), with a molecular communications layer, which models the exchange of metabolites by the bacterial populations in this structure. Supported by these models, we also developed a virtual GB to simulate the metabolic interactions that typically occurs in the human GB. These simulations allowed us to study the impacts caused by the metabolite exchanges on the human GB structure (i.e. nodes weight and hamming distance). Through our analyses, we found that the molecular inputs affect the bacterial populations in the human GB differently by modifying the nodes and edges weights of our GB interaction model. Our results also show that modifications in the human GB structure, specifically changing the sizes of *Bacteroides* and *Faecalibacterium* populations can lead to improvement/reduction in the production of SCFA, which may result in metabolic diseases in humans. Based on our results, we also infer that there is an intrinsic relationship between the investigated bacterial populations sizes, the increase/decrease of specific metabolites (SCFAs), and the overall balance of the human GB. These results can support the development of novel strategies to treat unbalanced human GB, and can provide insights on the role of other metabolites and molecules on the maintenance of a healthy gut bacteriome.

## Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland under Grant Number [16/RC/3835].

## References

[1] L. K. Ursell, J. L. Metcalf, L. W. Parfrey, and R. Knight, "Defining the human microbiome," *Nutrition reviews*, vol. 70, no. suppl_1, pp. S38–S44, 2012.

[2] S. Balasubramaniam, N. Lyamin, D. Kleyko, M. Skurnik, A. Vinel, and Y. Koucheryavy, "Exploiting bacterial properties for multi-hop nanonetworks," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 184–191, 2014.

[3] S. Sanna, N. R. van Zuydam, A. Mahajan, A. Kurilshikov, A. V. Vila, U. Võsa, Z. Mujagic, A. A. Masclee, D. M. Jonkers, M. Oosting, *et al.*, "Causal relationships among the gut microbiome, short-chain fatty acids and metabolic diseases," *Nature genetics*, vol. 51, no. 4, pp. 600–605, 2019.

[4] L. Wen and A. Duffy, "Factors influencing the gut microbiota, inflammation, and type 2 diabetes," *The Journal of nutrition*, vol. 147, no. 7, pp. 1468S–1475S, 2017.

[5] A. Iljazovic, U. Roy, E. J. Gálvez, T. R. Lesker, B. Zhao, A. Gronow, L. Amend, S. E. Will, J. D. Hofmann, M. C. Pils, *et al.*, "Perturbation of the gut microbiome by prevotella spp. enhances host susceptibility to mucosal inflammation," *Mucosal Immunology*, pp. 1–12, 2020.

[6] M. A. Henson and P. Phalak, "Microbiota dysbiosis in inflammatory bowel diseases: in silico investigation of the oxygen hypothesis," *BMC systems biology*, vol. 11, no. 1, p. 145, 2017.

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3148672




[7] J. T. Bjerrum, Y. Wang, F. Hao, M. Coskun, C. Ludwig, U. Günther, and O. H. Nielsen, "Metabonomics of human fecal extracts characterize ulcerative colitis, crohn's disease and healthy individuals," *Metabolomics*, vol. 11, no. 1, pp. 122–133, 2015.

[8] S. M. Hill, L. M. Heiser, T. Cokelaer, M. Unger, N. K. Nesser, D. E. Carlin, Y. Zhang, A. Sokolov, E. O. Paull, C. K. Wong, *et al.*, "Inferring causal molecular networks: empirical assessment through a community-based effort," *Nature methods*, vol. 13, no. 4, pp. 310–318, 2016.

[9] V. K. Gupta, M. Kim, U. Bakshi, K. Y. Cunningham, J. M. Davis, K. N. Lazaridis, H. Nelson, N. Chia, and J. Sung, "A predictive index for health status using species-level gut microbiome profiling," *Nature communications*, vol. 11, no. 1, pp. 1–16, 2020.

[10] S. V. Lynch, S. C. Ng, F. Shanahan, and H. Tilg, "Translating the gut microbiome: ready for the clinic?," *Nature Reviews Gastroenterology & Hepatology*, vol. 16, no. 11, pp. 656–661, 2019.

[11] J. Yang, P. Zheng, Y. Li, J. Wu, X. Tan, J. Zhou, Z. Sun, X. Chen, G. Zhang, H. Zhang, *et al.*, "Landscapes of bacterial and metabolic signatures and their interaction in major depressive disorders," *Science advances*, vol. 6, no. 49, p. eaba8555, 2020.

[12] S. Kim, I. Thapa, L. Zhang, and H. Ali, "A novel graph theoretical approach for modeling microbiomes and inferring microbial ecological relationships," *BMC genomics*, vol. 20, no. 11, pp. 1–13, 2019.

[13] V. Petrov, S. Balasubramaniam, R. Lale, D. Moltchanov, Y. Koucheryavy, *et al.*, "Forward and reverse coding for chromosome transfer in bacterial nanonetworks," *Nano Communication Networks*, vol. 5, no. 1-2, pp. 15–24, 2014.

[14] I. F. Akyildiz, M. Pierobon, and S. Balasubramaniam, "An information theoretic framework to analyze molecular communication systems based on statistical mechanics," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1230–1255, 2019.

[15] I. F. Akyildiz, M. Pierobon, and S. Balasubramaniam, "Moving forward with molecular communication: From theory to human health applications [point of view]," *Proceedings of the IEEE*, vol. 107, no. 5, pp. 858–865, 2019.

[16] S. M. Jandhyala, R. Talukdar, C. Subramanyam, H. Vuyyuru, M. Sasikala, and D. N. Reddy, "Role of the normal gut microbiota," *World journal of gastroenterology: WJG*, vol. 21, no. 29, p. 8787, 2015.

[17] A. Visconti, C. I. Le Roy, F. Rosa, N. Rossi, T. C. Martin, R. P. Mohney, W. Li, E. de Rinaldis, J. T. Bell, J. C. Venter, *et al.*, "Interplay between the human gut microbiome and host metabolism," *Nature communications*, vol. 10, no. 1, pp. 1–10, 2019.

[18] L. V. Hooper and J. I. Gordon, "Commensal host-bacterial relationships in the gut," *Science*, vol. 292, no. 5519, pp. 1115–1118, 2001.

[19] M. Hasan, E. Hossain, S. Balasubramaniam, and Y. Koucheryavy, "Social behavior in bacterial nanonetworks: Challenges and opportunities," *IEEE Network*, vol. 29, no. 1, pp. 26–34, 2015.

[20] A. Naqvi, H. Rangwala, A. Keshavarzian, and P. Gillevet, "Network-based modeling of the human gut microbiome," *Chemistry & biodiversity*, vol. 7, no. 5, pp. 1040–1050, 2010.

[21] S. N. Steinway, M. B. Biggs, T. P. Loughran Jr, J. A. Papin, and R. Albert, "Inference of network dynamics and metabolic interactions in the gut microbiome," *PLoS computational biology*, vol. 11, no. 6, p. e1004338, 2015.

[22] R. K. Singh, H.-W. Chang, D. Yan, K. M. Lee, D. Ucmak, K. Wong, M. Abrouk, B. Farahnik, M. Nakamura, T. H. Zhu, *et al.*, "Influence of diet on the gut microbiome and implications for human health," *Journal of translational medicine*, vol. 15, no. 1, pp. 1–17, 2017.

[23] M. S. R. Rajoka, J. Shi, H. M. Mehwish, J. Zhu, Q. Li, D. Shao, Q. Huang, and H. Yang, "Interaction between diet composition and gut microbiota and its impact on gastrointestinal tract health," *Food Science and Human Wellness*, vol. 6, no. 3, pp. 121–130, 2017.

[24] B. D. Unluturk, S. Balasubramaniam, and I. F. Akyildiz, "The impact of social behavior on the attenuation and delay of bacterial nanonetworks," *IEEE transactions on nanobioscience*, vol. 15, no. 8, pp. 959–969, 2016.

[25] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, "A comprehensive survey of recent advancements in molecular communication," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1887–1919, 2016.

[26] Y. Chahibi and I. F. Akyildiz, "Molecular communication noise and capacity analysis for particulate drug delivery systems," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3891–3903, 2014.

[27] M. Kuscu, E. Dinc, B. A. Bilgin, H. Ramezani, and O. B. Akan, "Transmitter and receiver architectures for molecular communications: A survey on physical design with modulation, coding, and detection techniques," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1302–1341, 2019.

[28] I. Llatser, A. Cabellos-Aparicio, M. Pierobon, and E. Alarcón, "Detection techniques for diffusion-based molecular communication," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 726–734, 2013.

[29] K. Oliphant and E. Allen-Vercoe, "Macronutrient metabolism by the human gut microbiome: major fermentation by-products and their impact on host health," *Microbiome*, vol. 7, no. 1, p. 91, 2019.

[30] D. P. Martins, M. T. Barros, and S. Balasubramaniam, "Using competing bacterial communication to disassemble biofilms," in *Proceedings of the 3rd ACM International Conference on Nanoscale Computing and Communication*, pp. 1–6, 2016.

[31] M. M. Deza and E. Deza, "Encyclopedia of distances," in *Encyclopedia of distances*, pp. 1–583, Springer, 2009.

[32] T. E. Gorochowski, A. Matyjaszkiewicz, T. Todd, N. Oak, K. Kowalska, S. Reid, K. T. Tsaneva-Atanasova, N. J. Savery, C. S. Grierson, and M. di Bernardo, "BSim: An agent-based tool for modeling bacterial populations in systems and synthetic biology," *PLoS ONE*, vol. 7, p. e42790, Aug. 2012.

[33] M. Kanehisa, S. Goto, and K.E.G.G., "Kyoto encyclopedia of genes and genomes," *Nucleic Acids Res*, vol. 28, p. 27–30, 2000. pubmed] [doi.

[34] M. Kanehisa, "Toward understanding the origin and evolution of cellular organisms," *Protein Sci*, vol. 28, p. 1947–1951, 2019. pubmed] [doi.

[35] M. Kanehisa, M. Furumichi, Y. Sato, M. Ishiguro-Watanabe, and M. Tanabe, "Kegg: integrating viruses and cellular organisms," *Nucleic Acids Res*, vol. 49, p. 545– 551, 2021. pubmed] [doi.

[36] J. Sung, S. Kim, J. J. T. Cabatbat, S. Jang, Y.-S. Jin, G. Y. Jung, N. Chia, and P.-J. Kim, "Global metabolic interaction network of the human gut microbiota for context-specific community-scale analysis," *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.

[37] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Keseler, A. Kothari, A. Kubo, *et al.*, "The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases," *Nucleic acids research*, vol. 42, no. D1, pp. D459–D471, 2014.

[38] C. Huttenhower, D. Gevers, R. Knight, S. Abubucker, J. H. Badger, A. T. Chinwalla, H. H. Creasy, A. M. Earl, M. G. FitzGerald, R. S. Fulton, *et al.*, "Hmp phase i (v3-v5)," 2012.

[39] A. O'Callaghan and D. van Sinderen, "Bifidobacteria and their role as members of the human gut microbiota," *Frontiers in microbiology*, vol. 7, p. 925, 2016.

[40] I. Rowland, G. Gibson, A. Heinken, K. Scott, J. Swann, I. Thiele, and K. Tuohy, "Gut microbiota functions: metabolism of nutrients and other food components," *European journal of nutrition*, vol. 57, no. 1, pp. 1–24, 2018.

[41] G. Den Besten, K. Van Eunen, A. K. Groen, K. Venema, D.-J. Reijngoud, and B. M. Bakker, "The role of short-chain fatty acids in the interplay between diet, gut microbiota, and host energy metabolism," *Journal of lipid research*, vol. 54, no. 9, pp. 2325–2340, 2013.

# Chapter 6

# Conference: Information Flow of Cascading Bacterial Molecular Communication Systems with Cooperative Amplification

| | |
|---|---|
| **Conference Title:** | IEEE International Conference on Computer Communications |
| **Article Type:** | Regular Paper |
| **Complete Author List:** | Samitha S. Somathilaka, Daniel Perez Martins, Sasitharan Balasubramaniam |
| **Keywords:** | Molecular communication, Bacterial networks, Mutual information, Residual noise, Parallel communications channels, Metabolic pathways, Cooperative amplification. |
| **Status:** | Published. August 2022. doi:10.1109/ICC45855.2022.9839035 |

# Information Flow of Cascading Bacterial Molecular Communication Systems with Cooperative Amplification

Samitha S. Somathilaka
*Walton Institute*
*Waterford Institute of Technology*
Waterford, X91 P20H, Ireland
samitha.somathilaka@waltoninstitute.ie

Daniel P. Martins
*Walton Institute*
*Waterford Institute of Technology*
Waterford, X91 P20H, Ireland
daniel.martins@waltoninstitute.ie

Sasitharan Balasubramaniam
*School of Computing*
*University of Nebraska-Lincoln*
Lincoln, NE, USA
sasi@unl.edu

***Abstract*—Bacterial ecosystems are integrated with cascading molecular communications networks that contain redundant paths transmitting molecular signals through a shared medium, resulting in accumulation of diverse molecules. Due to a range of factors, including residual noise and channel attenuation, the information flow between bacterial populations can be affected. Although the cooperative transceiver bacterial populations in parallel paths of the network amplify molecular signals to overcome channel attenuation, it further minimises the residual noise by absorbing higher signal molecules resulting in reliable information flow through the network. In this study, using information and molecular communications theory, we investigate the impact of Cooperative Amplification (CA) on InterSymbol Interference (ISI) in Bacterial Molecular Communication Networks (BMCN) with redundant paths. Moreover, we analyse the information flow through a cascading and parallel molecular communications system that uses different molecules as signals. We first show the effect of CA on the ISI and then the reliability of bacterial molecular networks using a vital metabolic functionality of the Human Gut Bacteriome (HGB), which is Short Chain Fatty Acids (SCFA) production. The analysis on the CA shows that the performance of the network can be enhanced up to a certain level by increasing the number of cooperate transceivers. Finally, the estimated Mutual Information (MI) for each bacterial population for three different networks using the data generated from simulations, indicates that the molecular communication network with redundant paths can support reliable information flow despite significant molecular residual noise.**

***Index Terms*—Molecular communication, Bacterial networks, Mutual information, Residual noise, Parallel communications channels, Metabolic pathways, Cooperative amplification.**

## I. Introduction

The Human Gut Bacteriome (HGB) is the bacterial ecosystem residing in the human gut and considered as a virtual organ [1] due to its involvement in critical metabolic tasks of the host through molecular interactions [2]. This results in formation of a complex molecular interaction network containing a large number of nodes and interactions through various molecular species, as illustrated in Figure 1. Hence, we identify this network as a Molecular Communications (MC) network that contains the interactions of a massive amount of channels with numerous molecular species. MC is an emerging field with a plethora of novel applications, especially in the biological field, such as biosensing and biocomputing [3], [4]. Some of the characteristics of HGB environment contrast the behaviors of MC from the conventional communication system, such as the production of residual noise. This process occurs due to the incomplete utilization of molecular species found in HGB environments [5]. Therefore, the residual noise has been modelled in MC systems as ISI, where a molecular signal emitted in a specific time slot will interfere with the decodification of the next [6]. Moreover, this effect has an enlarged impact in cascading MC networks due to the number of parallel transmitters sending molecular signals towards a single receptor.

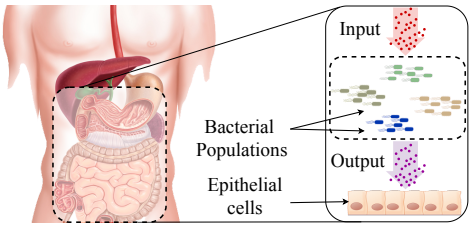



Fig. 1: The figure illustrates how the HGB receive molecular inputs and convert them into output signals. Finally, the output signal is being received by the host's epithelial cells.

In addition to the residual noise, the exchange of molecules in the HGB can also suffer from Brownian noise due to molecular displacement [7]. Therefore, it is crucial to investigate the dynamics of the communications that support the bacterial networks in HGB when subjected to such types of noise. To this day, many studies have investigated the effects of noise in diffusive communications channels. For instance, an analysis of diffusion-based noise source that affects end-to-end MC systems was developed in [8]. Further, Moore et al. [9] modelled residual noise and introduced two approaches for noise reduction in order to achieve higher information rates. The effect of molecular noise on the channel capacity has been analysed by Nakano et al. [10], where they consid-

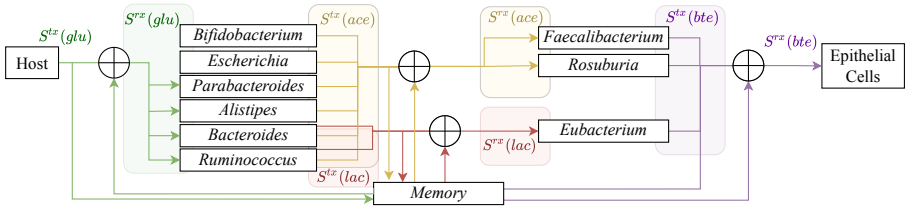



Fig. 2: Bacterial cascading system of SCFA production with the environment memory component. $S^{tx}(glu)$, $S^{tx}(ace)$, $S^{tx}(lac)$ are the transmitted and $S^{rx}(glu)$, $S^{rx}(ace)$, $S^{rx}(lac)$ and $S^{rx}(bte)$ are the received glucose, acetate, lactate and butyrate signals respectively that are affected by the noise from the memory.

ered a nanomachine transmitter placed in a one-dimensional molecular communication channel (they assumed that the molecular propagation via Brownian motion would degrade over time) [10]. In a different direction, molecular noise analysis targeting specific application such as drug delivery systems have also been proposed. For example, Chahibi and Akyildiz [11] conducted a noise analysis of the nano-particle propagation based on a long-term drug distribution throughout the body. Here, we extend these works to investigate the effects of both residual and Brownian noises on a Bacterial Molecular Communication Network.

This paper investigates the reliability of BMCN considering parallel molecular signal transceivers as cooperative amplifiers, which are identified as multiple nodes coordinating and transmitting the same signal and amplifying it to improve the range of transmission [12]. Although CA is a solution to overcome the channel attenuation [13], it may increase the residual noise and ISI, reducing the performance of the BMCN. To investigate this phenomenon, we use a sub-network of HGB, where cascading communications are required for the production of Short Fatty Chain Acids - SCFA (SCFA is one of the most important metabolic functions of the HGB [14]) as shown in Figure 2. The bacterial populations represented in Figure 2 were selected after a careful investigation on real data obtained from MicrobiomeDB [15]. Furthermore, we investigate the performance of our proposed cascading MC system through the evaluation of the MI (using a method inspired by [16]) for each one of the communications links involved in this sub-network. Our model also takes into account the accumulation effect that occurs in an MC channel when multiple sources produce a large number of molecules, approximating our investigation to the real production of SCFA by the bacterial populations in HGB.

The rest of the paper is organized as follows. Section II explains the system dynamics of the considered cascading sub-network of the HGB, and Section III focuses on the analysis conducted on the mentioned system. Next, in Section IV we discuss the final results and conclude the paper in Section V.

## II. Cascading Molecular Communication System

The MC interaction network of the HGB is a collection of numerous metabolic paths. Each path consists of multiple bacterial populations (i.e., molecular transceivers or nodes) that can receive molecular signals from a node, process them and release another type of molecular signal to the next node. The molecular signal processing can be described as follows, $M_1 + M_2 \underset{k_r}{\overset{k_f}{\rightleftharpoons}} M_3$ where $M_1$, $M_2$ and $M_3$ are the molecules, $k_f$ and $k_r$ are the forward and reverse reaction rates. If the reaction is non-reversible, $k_r = 0$.

The molecular propagation between two nodes in the HGB environment is affected by various factors such as molecular noise [8]. In cascading molecular communication systems, the magnitude of noise impact is expanded with the progression through cascading layers as each layer has its own noise component. If the cascading channels use the same type of molecules, the issue becomes more significant as the diffusion medium facilitates increased accumulation. Nevertheless, this study focuses only on a BMCN that uses different types of molecules for cascading layers, as shown in Figure 2.

In diffusion based MC channels, only a certain portion of emitted molecules reach the particular receiver within a respective time frame. The rest of them accumulate in the environment and act as residual noise (Figure 3) causing ISI.

## III. Information Flow Analysis

The analysis on the bacterial MC network of this study is structured to investigate the impact of CA on residual memory and the reliability of the network in terms of information flow. First, we analyse how the CA changes the residual memory resulting in alterations of ISI. Later, MI is calculated to understand the information flow through the network.

### A. Intersymbol interference analysis

As shown in Figure 2, the considered system contains parallel paths that transmit the same signal through the system. This phenomenon can be identified as CA [13], and it can affect the ISI levels experienced by the investigated system. To investigate this relationship, we first use a generic setup that contains multiple bacterial nodes cooperating to amplify the molecular signals. Then, we apply the obtained results to the evaluation of the SCFA production. Here, we measure the ISI by calculating the interference signal to total signal strength ratio [6], as follows.

$$ISI = \frac{\sum_{k=0}^{K} \sum_{a_0=1}^{A_0} q^{rx}_{(B_k,a_0)}}{\sum_{k=0}^{K} \sum_{a_0=0}^{A_0} q^{rx}_{(B_k,a_0)} + \sum_{k=0}^{K} \sum_{a_1=0}^{A_1} q^{rx}_{(B_k,a_1)}} \quad (1)$$

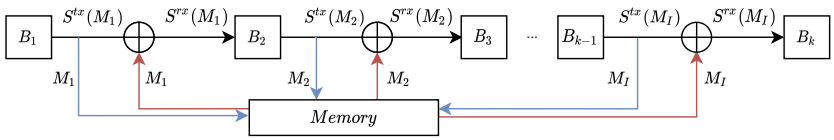



Fig. 3: Cascading communication system with channel memories causing residual noise

where $q^{rx}_{(B_k,a_0)}$ and $q^{rx}_{(B_k,a_1)}$ are the number of molecules received by the bacterial population $B_k$ $(k = 0, 1, ..., K)$ for $a_0^{th}$ bit $(a_0 = 0, 1, ..., A_0)$ and $a_1^{th}$ bit $(a_1 = 0, 1, ..., A_1)$ for "0" and "1" symbols, respectively.

### *B. Mutual information analysis*

In addition to investigating the relationship between CA and ISI, this study explores the information flow capabilities of BMCN in the HGB. Here, we estimate the MI of the cascading BMCN for the butyrate production for three different scenarios (control, Autism, and Parkinson's disease average compositions). Butyrate is one of the most important SCFA, and it can be considered as the primary energy source for the colon epithelial cells. Further, it has the properties for immunomodulation and anti-inflammation [17].

We model the butyrate as the final product of a cascading BMCN containing nine bacterial genera, and this selection of bacterial populations takes into account their similarities between metabolic functionalities and ancestral origins. The butyrate production starts from the glucose input into the HGB, as illustrated in Figure 2, and we can identify four layers in this system: 1) host - glucose transmitters, 2) acetate/lactate producers - consume glucose and produce acetate/lactate, 3) butyrate producers - consume acetate/lactate and produce butyrate, and 4) epithelial cells - butyrate receivers. For such configuration, the *Bacteroides, Alistipes, Parabacteroides, Bifidobacterium, Ruminococcus* and *Escherichia* genera receive glucose and transmit acetate. Moreover, the *Faecalibacterium* and *Roseburia* genera receive acetate, and *Eubacterium* receives lactate to produce butyrate. Finally, the epithelial cells receive butyrate signals produced by the bacterial populations. This bacterial ecosystem is implemented (as shown in Figure 2) on our simulations by combining the literature curated data on metabolism and compositional data from MicrobiomeDB.

We can break down the nine bacterial genera (i.e., nodes) into a combination of series and parallel communications links and investigate the interactions between pairs of nodes. This combination is our cascading BMCN model, which can be seen in Figure 3 (representation of one of the communications links required for the butyrate production). Considering one channel between two bacterial genera, i.e. pair $p$, in the cascading system that uses a $M_i$ molecule as the signalling molecule, we calculate the MI of this interaction $I_p$ as follows,

$$I_p = H(S_h^{tx}) - H(S_h^{tx}|\{S^{rx}_{B_k}(t),\ t_0 \le t \le t_{max}\}). \quad (2)$$

where $H(S_h^{tx})$ is the estimated entropy of the input signal, $S^{tx}$ from the host and $H(S_h^{tx}|S^{rx}_{B_k})$ is the estimated conditional entropy of input signal $S_h^{tx}$ given the received signal $S^{rx}_{B_k}$ by the bacterial population $B_k$, where $tx$ and $rx$ are the identifiers for the transmitted and received signals, respectively.

We generate data for different input concentrations $c_j, j = 1, ..., J$ of the input signal $S^{tx}$. For each concentration, the simulator iterated $R$ times with $T$ number of time steps. We utilise the data obtained from these iterations to compute the entropy of the glucose input signal $H(S_h^{tx})$ using a histogram approach [18], [19] as follows,

$$H(S_h^{tx}) = -\sum_{j=1}^{J} p_{S_h^{tx}}(c_j) \log_2 \left( \frac{p_{S_h^{tx}}(c_j)}{w_{S_h^{tx}}} \right) \quad (3)$$

where $p_{S_h^{tx}}(c_j) = 1/J$ is the probability of each input concentration $c_j$, $J$ is number of input concentrations, and

$$w_{S_h^{tx}} = \frac{\max(c_j) - \min(c_j)}{J}, \quad (4)$$

is the sample interval for the input signal. Using (3)-(4), we compute the conditional entropy as follows

$$\begin{aligned} &H(S_h^{tx}|\{S^{rx}_{B_k}(t),\ t_0 \le t \le t_{max}\}) = \\ &-\sum_{N_{B_k,t_0}} \sum_{N_{B_k,t_1}} \cdots \sum_{N_{B_k,t_{max}}} p_{S^{rx}_{B_k}}(\alpha) \\ &\cdot \sum_{s=1}^{S_\alpha} p_{S_h^{tx}|\alpha}(c_j) \log_2 \left( \frac{p_{S_h^{tx}|\alpha}(c_j)}{w_{S_h^{tx}\alpha}} \right) \end{aligned} \quad (5)$$

where $\alpha = \{c^{rx}_{B_k,t}\}_{t=0}^{t_{max}}$ is the set of concentrations of the received signal $S^{rx}$ through out the considered simulation time steps, $N_{B_k,t_n}$ is the number of bins of the received signal $S^{rx}_{B_k}$ used to generate the multi-dimensional histogram $p_{S^{rx}_{B_k}}(\alpha)$, $V_\alpha$ is the number of histogram bins used to generate the multi-dimensional histogram $p_{S_h^{tx}|\alpha}(c_j)$ while $w_{S_h^{tx},\alpha}$ which is calculated as,

$$w_{S_h^{tx},\alpha} = \frac{\max(c_j) - \min(c_j)}{V_\alpha}. \quad (6)$$

Moreover, $N_{B_k,t_n}$ is calculated as follows to the nearest integer [19],

$$N_{B_k,t_n} = 1 + \log_2(U) + \log_2 \left( 1 + \frac{g_\beta}{\sigma_{g_\beta}} \right) \quad (7)$$

where $\beta = c^{rx}_{B_k}(t_n)$ is the concentration of the received signal by $B_k$ at time $t_n$ of , $U = J \cdot R$ is the number of simulation runs, and $g_\beta$ is estimated skewness of the distribution of $P_{S^{rx}_{B_k}}$ while $\sigma_{g_\beta}$ is calculated as,

$$\sigma_{g_\beta} = \sqrt{\frac{6(U-2)}{(U+1)(U+3)}}. \quad (8)$$

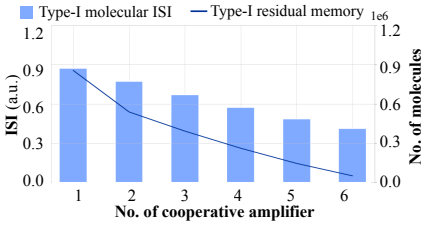



Fig. 4: Impact of the CA on the residual memory and ISI

Using the same approach to calculate the number of bins of the received signal in (7), we calculate the number of histogram bins for the input signal $V_\alpha$ to the nearest integer, which is represented as,

$$V_\alpha = 1 + \log_2(I) + \log_2\left(1 + \frac{g_{c_j}}{\sigma_{g_{c_j}}}\right), \tag{9}$$

where $g_{c_j}$ is estimated skewness of the distribution of $c_j$ and,

$$\sigma_{g_{c_j}} = \sqrt{\frac{6(I-2)}{(I+1)(I+3)}}. \tag{10}$$

## IV. Results

In this section, we show the analytical results for ISI against CA and results for the MI of the investigated BMCN. We generated data for these analyses using a bacterial ecosystem simulator that we developed based on real data collected from several microbiome databases available online, such as MicrobiomeDB and NJS16 [5]. The simulator is used to conduct *in silico* experiments and generate data on bacterial interactions. It has a 3D environment (dimensions: $150 \times 150 \times 150\,\mu$m) with a static medium where molecules propagate via diffusion. Mainly, this diffusion induces the stochastic nature of the system. The simulator utilises a voxel architecture to discretise the environment for precise data generation on the transmission and reception for each molecular species. Further, the simulator was fined tuned by using an array of iterative experiments to match the average production behaviours of SCFA in human GB (for further details on the extraction of the average behaviour, please refer [20]). We utilise this configuration to approximate our simulation to *in vivo* or *in vitro* experiments on SCFA production/consumption in the HGB.

### A. Intersymbol interference vs cooperative amplification

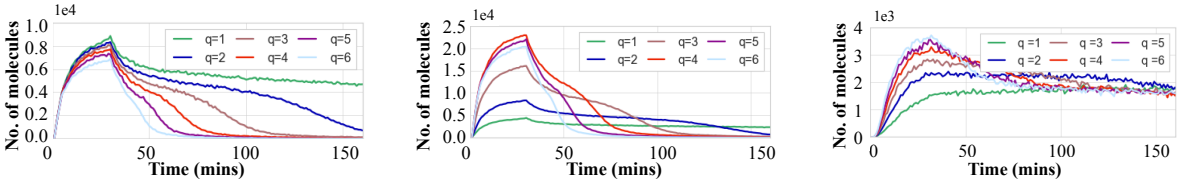
In order to investigate the impact of multiple transceivers on the signal amplification and how it affects the residual memory indirectly, we use a generic setup with six experiments. The first setup only contains one transceiver and developed the other five setups by increasing the number of cooperate transceivers up until six. We used the same input signal with four "1" bits (where bit "1" represented by the transmission of molecules and bit "0" by not transmitting any molecule) of molecule type-I from $TX$ with a fixed amplitude for each setup. The transceivers receive type-I molecules and transmit type-II signal to the receiver $RX$. Figure 4 shows the results of the signal behaviours due to CA. When the number of cooperative transceivers increases in the environment, they absorb more molecules resulting minimised probability of molecular accumulation. Reduction of accumulation reflects in ISI. When there is only one transceiver in the environment, Figure 4 shows the accumulation of Type-I molecules is high, and similarly, ISI is also high. With the increment of the cooperative transceivers, it is clearly evident that the amount of Type-I residual memory decreases and the ISI of Type-I also reduces. Therefore, it can be concluded that in MC, CA can be considered as a solution for ISI.

Further, we extended the analysis to evaluate the signal reception behaviour by an individual transceiver, signal strength of cooperative transmission and the received signal strength at the end of the system. As shown in Figure 5, here we only focus on a single pulse to clearly observe the variations. In Figure 5a it is evident that when the number of transceivers increases in the environment the possibility of signal reception of each node degrades. Nevertheless, the transmitted signal from the transceivers collectively amplifies until a certain concentration of transceivers in the environment (in this setup, the maximum strength of the transmitted signal is in the setup with four transceivers) as shown in the Figure 5b. Hence, Figure 5c explains there is a maximum strength that can be expected by increasing the number of cooperative transceivers.

### B. Analytical results for information flow

BMCNs in the HGB suffers from the same noise impact we discussed in previous sections, but they are equipped with redundant paths through parallel transceiver bacterial populations that can be considered as cooperative amplifiers. The composition of the HGB differs for each individual, and the scale of CA relies on the compositions as well. Therefore, we estimated the information flow for the control, autistic and Parkinson's HGBs average compositions.

We initiate the simulator with the Relative Abundance (RA) of each genus, which is calculated based on the species level RA found in the data analysed from MicrobiomeDB and Disbiome [21] databases. We followed the same procedure on the sample data from the Disbiome database to define the average compositions related to two diseases, autism and Parkinson's disease. Next, we introduce twenty single pulse glucose inputs with different concentration ranging from $0.259\,\mu\text{mol}/\text{m}^3\text{s}$ to $2.594\,\mu\text{mol}/\text{m}^3\text{s}$ to start the production of molecules in the second layer of our cascading MC model for all three setups (control, autism and Parkinson's). The amplitudes of the molecular inputs were selected based on the number of bacterial cells we implemented on our simulator to get results with clear changes in the information flow. Above and below that range, results does not indicate any significant changes. The simulation duration was set to 500 minutes by observing the strongest (glucose input) and the weakest (lactate input) signal behaviours in order to have all the significant changes of signals within the mentioned time frame. Further, each setup




(a) (b) (c)

Fig. 5: Illustration of CA by simultaneous transceivers where (a) Number of type-I molecules received per transceiver, (b) total number of type-II molecules produced of $q$ number of transceivers and (c) number of type-II molecules received by the end receiver.

iterated for 50 arbitrary times to increase the accuracy of the results over the system stochasticity and collected the data of molecular consumption and production of each bacterial population for all time steps.

Figure 6 depicts the behaviours of all the four input signals. Figure 6a shows the ten out of twenty single pulse glucose inputs to the system, while Figure 6b exhibits the combination of the acetate transmission by *Bacteroides*, *Alistipes*, *Parabacteroides*, *Bifidobacterium*, *Ruminococcus* and *Escherichia*, which is several times weaker than the glucose signal. The strength of the signal transmitted by these bacteria genera relies on the number of acetate-producing bacterial cells and the quality of signal reception by the next layer of this cascading MC system. Compared to the acetate signal, lactate and butyrate signals are significantly weaker, but with a preserved waveform as shown by Figures 6c and 6d. These signals are used to compute the performance of the information flow on this cascading BMCN.

First, we compute the input signal entropy and conditional entropy of the cascading BMCN using (3) and (5) respectively for each setup. Then, the conditional entropy for all nine bacteria genera were estimated in three steps for each setup. First, the conditional entropy for the *Bacteroides*, *Alistipes*, *Parabacteroides*, *Bifidobacterium*, *Ruminococcus* and *Escherichia* genera were computed considering glucose input signals. Then, the conditional entropy for *Faecalibacterium* and *Roseburia* genera were evaluated based on the data generated for their acetate reception. Finally, conditional entropy for lactate channel of *Eubacterium* genera and butyrate reception by epithelial cells were also computed. This step is required for the evaluation of the MI for these bacterial interactions. MI for each bacterial interaction is calculated using (2) and the results for each bacterial population is shown in Figure 7.

The results in Figure 7a, 7b and 7c, elucidate that, with the compositional changes, information flow differ through the network. The set of glucose receivers is considered the first layer of cooperative amplifiers and the acetate and lactate receivers as the second layer of cooperative amplifiers in this cascading system. The estimated MIs of first layers of three compositions differ significantly. The estimated MIs in the second layer show a lesser variation compared to the first layer, and at the end, MI of epithelial cells have close values for all

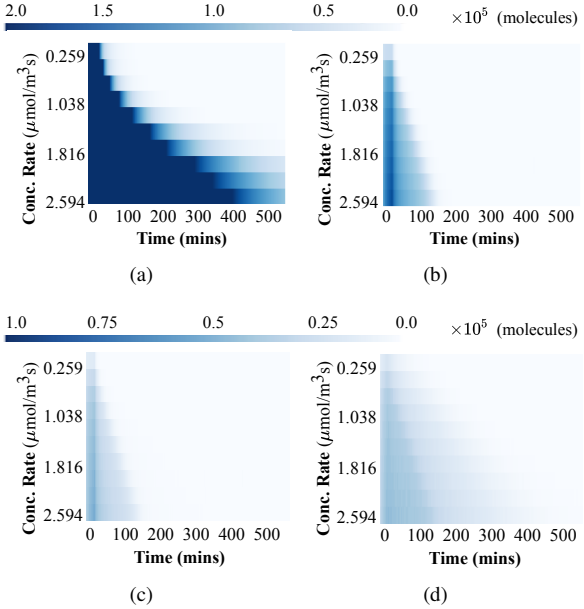



Fig. 6: Input signal behaviours for the considered time frame of (a) glucose, (b) acetate, (c) lactate and (d) butyrate signals for all the glucose input concentrations. Note that (c) and (b) are with a lower scale compared to (a) and (b).

three setups. The maximum value of MI is from *Bacteroides* in all setups, as it dominates the ecosystems RA-wise. Interestingly, transceivers' MIs of a layer tend to have higher values compared to those of the previous layer. This highlights, in systems with redundant paths through cooperative transceivers, higher information flow can be expected.

These results show that the use of CA is an important mechanism to support a reliable information flow through BMCN. This opens up possibilities to utilise concepts and techniques currently applied to multi-path wireless communications to improve the performance of cascading MC systems.

## V. Conclusion

Similarly to conventional communications systems, the performance of MC systems can suffer from ISI. In a BMCN,

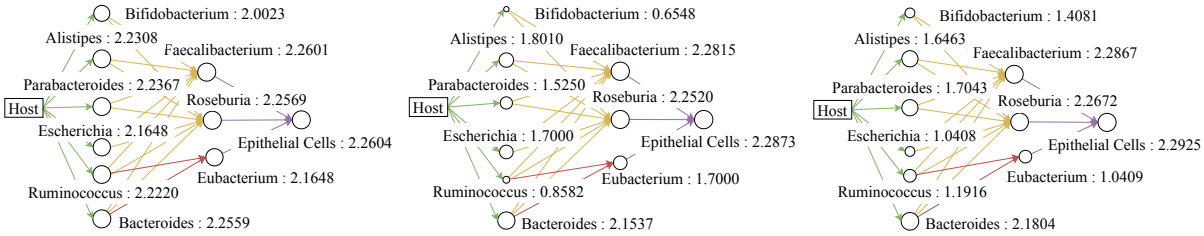



Fig. 7: Estimated MI values for each bacterial population and epithelial cells for (a) control HGB, (b) parkinson's HGB and (c) autistic HGB. Sizes of nodes represent the corresponding MI values in bits.

such as the one created in the HGB to metabolise SCFA, this is translated into multiple cascading molecular noises. Hence, in this study, we have considered this effect to investigate the information flow through a cascading MC system required for butyrate production. Here, we show that the problem of residual noise can be addressed up to a certain content using CA. At the same time, we also could observe, through the MI results, that the natural architecture created in HGB to produce butyrate makes this cascading BMCN more robust against such molecular noises. Furthermore, the MI results highlight that multipath communication with CA is an important mechanism to maintain the performance of a cascading BMCN in the HGB, which opens the possibility of implementing wireless techniques associated with indoor propagation to improve such MC systems. The obtained results contribute to the future design of a more reliable cascading BMCN that can send and receive information from the body. Reliable information transmission from/to the body is essential in smart drug delivery and smart diagnostics systems.

## Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland under Grant Number [16/RC/3835].

## References

[1] J. M. Evans, L. S. Morris, and J. R. Marchesi, "The gut microbiome: the role of a virtual organ in the endocrinology of the host," *J Endocrinol*, vol. 218, no. 3, pp. R37–47, 2013.

[2] S. Sanna, N. R. van Zuydam, A. Mahajan, A. Kurilshikov, A. V. Vila, U. Võsa, Z. Mujagic, A. A. Masclee, D. M. Jonkers, M. Oosting, *et al.*, "Causal relationships among the gut microbiome, short-chain fatty acids and metabolic diseases," *Nature genetics*, vol. 51, no. 4, pp. 600–605, 2019.

[3] T. Nakano, M. J. Moore, F. Wei, A. V. Vasilakos, and J. Shuai, "Molecular communication and networking: Opportunities and challenges," *IEEE Transactions on NanoBioscience*, vol. 11, no. 2, pp. 135–148, 2012.

[4] D. Bi, A. Almpanis, A. Noel, Y. Deng, and R. Schober, "A survey of molecular communication in cell biology: Establishing a new hierarchy for interdisciplinary applications," *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, pp. 1494–1545, 2021.

[5] J. Sung, S. Kim, J. J. T. Cabatbat, S. Jang, Y.-S. Jin, G. Y. Jung, N. Chia, and P.-J. Kim, "Global metabolic interaction network of the human gut microbiota for context-specific community-scale analysis," *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.

[6] M. U. Mahfuz, D. Makrakis, and H. T. Mouftah, "Characterization of intersymbol interference in concentration-encoded unicast molecular communication," in *2011 24th Canadian conference on electrical and computer engineering (CCECE)*, pp. 000164–000168, IEEE, 2011.

[7] S. K. Tiwari and P. K. Upadhyay, "Maximum likelihood estimation of snr for diffusion-based molecular communication," *IEEE Wireless Communications Letters*, vol. 5, no. 3, pp. 320–323, 2016.

[8] M. Pierobon and I. F. Akyildiz, "Capacity of a diffusion-based molecular communication system with channel memory and molecular noise," *IEEE Transactions on Information Theory*, vol. 59, no. 2, pp. 942–954, 2012.

[9] M. J. Moore, T. Suda, and K. Oiwa, "Molecular communication: Modeling noise effects on information rate," *IEEE Transactions on NanoBioscience*, vol. 8, no. 2, pp. 169–180, 2009.

[10] T. Nakano, Y. Okaie, and J.-Q. Liu, "Channel model and capacity analysis of molecular communication with brownian motion," *IEEE Communications Letters*, vol. 16, no. 6, pp. 797–800, 2012.

[11] Y. Chahibi and I. F. Akyildiz, "Molecular communication noise and capacity analysis for particulate drug delivery systems," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3891–3903, 2014.

[12] S. Abadal, I. Llatser, E. Alarcón, and A. Cabellos-Aparicio, "Cooperative signal amplification for molecular communication in nanonetworks," *Wireless networks*, vol. 20, no. 6, pp. 1611–1626, 2014.

[13] I. Llatser, A. Cabellos-Aparicio, and E. Alarcon, "Networking challenges and principles in diffusion-based molecular communication," *IEEE Wireless Communications*, vol. 19, no. 5, pp. 36–41, 2012.

[14] Y. P. Silva, A. Bernardi, and R. L. Frozza, "The role of short-chain fatty acids from gut microbiota in gut-brain communication," *Frontiers in endocrinology*, vol. 11, p. 25, 2020.

[15] C. Huttenhower, D. Gevers, R. Knight, S. Abubucker, J. H. Badger, A. T. Chinwalla, H. H. Creasy, A. M. Earl, M. G. FitzGerald, R. S. Fulton, *et al.*, "Hmp phase i (v3-v5)," 2012.

[16] A. Gohari, M. Mirmohseni, and M. Nasiri-Kenari, "Information theory of molecular communication: directions and challenges," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 2, pp. 120–142, 2016.

[17] A. Rivière, M. Selak, D. Lantin, F. Leroy, and L. De Vuyst, "Bifidobacteria and butyrate-producing colon bacteria: importance and strategies for their stimulation in the human gut," *Frontiers in microbiology*, vol. 7, p. 979, 2016.

[18] A. Papoulis, *Probability, random variables, and stochastic processes*. Boston: McGraw-Hill, 2002.

[19] Z. Sakkaff, A. Immaneni, and M. Pierobon, "Estimating the molecular information through cell signal transduction pathways," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2018.

[20] S. S. Somathilaka, D. P. Martins, W. Barton, O. O'Sullivan, P. Cotter, and S. Balasubramaniam, "A graph-based molecular communications model analysis of the human gut bacteriome," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2022.

[21] Y. Janssens, J. Nielandt, A. Bronselaer, N. Debunne, F. Verbeke, E. Wynendaele, F. V. Immerseel, Y.-P. Vandewynckel, G. D. Tré, and B. D. Spiegeleer, "Disbiome database: linking the microbiome to disease," vol. 18, June 2018.

# Chapter 7

# Journal: Revealing gene regulation-based neural network computing in bacteria

| | |
|---|---|
| **Journal Title:** | Biophysical Reports |
| **Article Type:** | Regular Paper |
| **Complete Author List:** | Samitha S. Somathilaka, Sasitharan Balasubramaniam, Daniel P. Martins, and Xu Li |
| **Status:** | Published. August 2022. https://doi.org/10.1016/j.bpr.2023.100118 |

Biophysical Reports

ARTICLE

# Revealing gene regulation-based neural network computing in bacteria

Samitha S. Somathilaka,[1,2,*] Sasitharan Balasubramaniam,[2] Daniel P. Martins,[1] and Xu Li[3]

[1]VistaMilk Research Centre, Walton Institute for Information and Communication Systems Science, South East Technological University, Waterford, Ireland; [2]School of Computing, University of Nebraska-Lincoln, Lincoln, Nebraska; and [3]Department of Civil and Environmental Engineering, University of Nebraska-Lincoln, Lincoln, Nebraska

ABSTRACT Bacteria are known to interpret a range of external molecular signals that are crucial for sensing environmental conditions and adapting their behaviors accordingly. These external signals are processed through a multitude of signaling transduction networks that include the gene regulatory network (GRN). From close observation, the GRN resembles and exhibits structural and functional properties that are similar to artificial neural networks. An in-depth analysis of gene expression dynamics further provides a new viewpoint of characterizing the inherited computing properties underlying the GRN of bacteria despite being non-neuronal organisms. In this study, we introduce a model to quantify the gene-to-gene interaction dynamics that can be embedded in the GRN as weights, converting a GRN to gene regulatory neural network (GRNN). Focusing on *Pseudomonas aeruginosa*, we extracted the GRNN associated with a well-known virulence factor, pyocyanin production, using an introduced weight extraction technique based on transcriptomic data and proving its computing accuracy using wet-lab experimental data. As part of our analysis, we evaluated the structural changes in the GRNN based on mutagenesis to determine its varying computing behavior. Furthermore, we model the ecosystem-wide cell-cell communications to analyze its impact on computing based on environmental as well as population signals, where we determine the impact on the computing reliability. Subsequently, we establish that the individual GRNNs can be clustered to collectively form computing units with similar behaviors to single-layer perceptrons with varying sigmoidal activation functions spatio-temporally within an ecosystem. We believe that this will lay the groundwork toward molecular machine learning systems that can see artificial intelligence move toward non-silicon devices, or living artificial intelligence, as well as giving us new insights into bacterial natural computing.

WHY IT MATTERS The increasing importance of artificial intelligence (AI) is weaving into numerous disciplines, providing us with new discoveries and levels of knowledge that were previously inaccessible. However, the increased development in AI technology is also providing us with a new opportunity of using it as a concept in understanding natural phenomena. This is the aim of this study, where we intend to use AI as a tool in characterizing the computational capabilities of bacterial cells. Besides improving our understanding through an AI model that is mapped onto the gene regulatory network, our study can also lead to new opportunities for future bio-computing.

## INTRODUCTION

Bacteria are well known for their capabilities to sense external stimuli and adapt into a wide range of responses (1,2). The interpretation of external signals includes molecules communicated from other microbes as well as changes in environmental conditions (e.g., changes in temperature or pH levels) (3). Bacterial cells continuously monitor the extracellular cues to regulate gene expression accordingly and, subsequently, protein production. The regulation mechanism is impressively complex and contains a massive number of components, including mRNA, activators, repressors, information stored in genes, RNA polymerase, and protein-binding regions (4,5). This process drives the cell's behavior to prolong its survivability and this is often identified as a decision-making process. However, this can also be identified as a chemical-based computing process that is computed as it traverses through the branches of the gene regulatory network (GRN) (6), where a large number of molecular transduction signals result in parallel and sequential gene expressions. This

*Submitted June 4, 2023, and accepted for publication July 26, 2023.*

*Correspondence: samitha.somathilaka@waltoninstitute.ie

Editor: Yoav Shechtman.

https://doi.org/10.1016/j.bpr.2023.100118



is governed by genetic circuits that contain approximately 100 to more than 11,000 genes. For example, the largest genome identified belongs to *Sorangium cellulosum* strain So0157-2 (7,8).

Despite the absence of neural structures for computing, the GRN allows the bacteria to strategize and adapt through varying conditions, and this results in molecular production to influence other cells, which leads to complex social interactions, motility to favorable environmental conditions, or physiological state changes. Exploring the natural computing properties of bacteria can lead to a better understanding of their behavior and open new opportunities for programming sensing and actuation functionalities into the cells for novel treatments (9), as well as creating new opportunities for future bio-computing systems (10).

The world of artificial intelligence (AI) is now trickling into our lives, increasing our reliance on numerous facets that play a role in our daily activities. Inspired by the workings of the brain, the core of artificial neural networks (ANN) is a graph structure that abstracts communication and computing in biological neuronal networks (11). This has allowed ANN algorithms to be programmed into various devices to support numerous applications, such as image recognition (12) and autonomous systems (13) However, neuromorphic computing is also inspired by the naturally evolved biological neural circuits differentiating it from conventional computing architectures (14). SpiNNaker (15), Neurogrid (16), Loihi (17), and TrueNorth (18) are examples of well-known neuromorphic processors that have been implemented. These approaches allowed for mitigating the bottlenecks of the von Neumann by having physical memristors in the system (19,20). In addition, neuromorphic architectures are capable of performing massively parallel computing that can outperform conventional computing in terms of efficiency (21). In contrast to designing bio-inspired silicon architectures, the programming of AI into computing devices has now extended to non-silicon machines, for example biological cells, and this has resulted in molecular machine learning systems (22,23). However, from a natural biological system perspective, it has been suggested that the computational process through the GRN that drives the bacterial cell's decision-making comprises a hidden neural network-like architecture (24,25). This indicates that, even though bacterial cells are categorized as non-neural organisms, we can use the neural network architecture representation of the GRN to characterize their computing capabilities. Through this representation, several ANN components can be identified in GRNs, where genes may be regarded as computing units or activation nodes and transcription factors (TF) as incoming signals to the computing unit and their degree of influence as weights/biases. Owing to a large number of genes and weighted relationships in a GRN, it is possible to infer sub-networks with neural network behaviors, which we term gene regulatory neural networks (GRNNs).

Although a number of studies have optioned toward engineering cells to create molecular machine learning (26), in this study we focus on the discovery and extraction of GRNN sub-networks from GRNs. Even though the GRN incorporates the influence of intergenic interactions, it lacks a layer of information on the magnitudes, which, from an neural network paradigm, represents the weights. In a typical NN, a perceptron is the fundamental computing element that can take multiple inputs, multiply them by a corresponding weight, and combine them through a summation process. This weighted summation is then passed through an activation function as shown in Fig. 1. The non-linearity observed between the gene expression patterns and the weighted summation of incoming TF signals of a gene can be mapped as the property of a perceptron. This non-linearity can be better represented by an activation function. By applying an activation function for each node in the GRN and using the single-layer perceptron model, we extract the weights of each edge of the GRN to recreate a GRNN. Further, using graph theoretical path analysis, we extract a sub-network from *Pseudomonas aeruginosa* GRN for pyocyanin (PYO) production to analyze the GRNN's computing behavior.

The contributions of this study are discussed here.

- **Extracting a GRNN**: As discussed earlier, past research has manifested evidence of neural-like behaviors inherited in isolated components, but no research has studied a complete GRNN of a bacterial cell to date. A gene perceptron with multiple inputs can be considered a single-layer perceptron, and transcriptional data provide input and output expression rates for each gene perceptron. Based on this, we developed an algorithm to extract weights for each edge of the GRN in a single-layer perception model. The accuracy of the extracted GRNN is then proved in the transcriptomic layer using a comparison of gene expression dynamics between wet-lab data and the model predictions. Subsequently, we employ this algorithm to extract the GRNN of the model species *P. aeruginosa* to investigate the cell's computing properties.
- **Impact of cell-cell communication on the GRNN computing**: The GRNN only represents single-cell activities, whereas bacteria usually live in complex ecosystems (27) where cell-cell communication heavily influences their behaviors (28). Hence, we focus on a biofilm use case to understand the intricate cell-cell communication that influences varying spatio-temporal behaviors. We establish this by

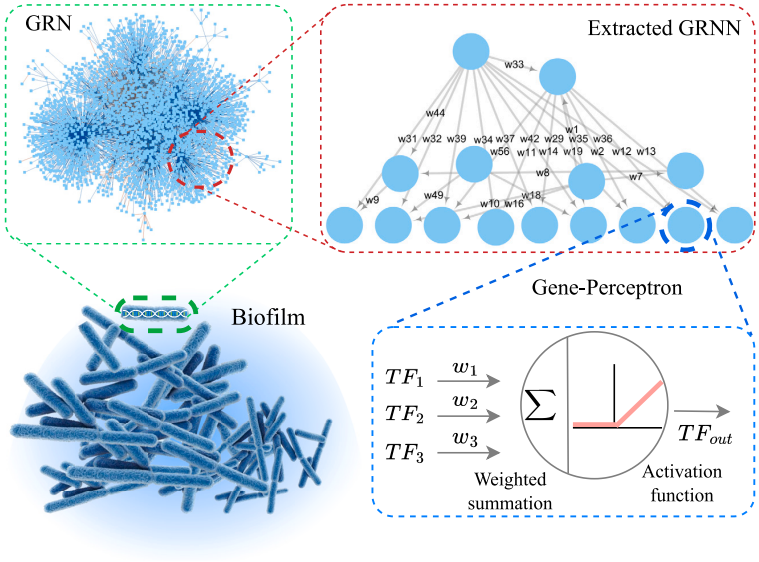



FIGURE 1 Illustration of a biofilm and the extraction of GRNNs from within the bacterial cells.

creating a graph neural network of the bacterial population with each cell embedded with the GRNN. The diffusion-based communication between the bacterial cells in the biofilm is then encoded as the message-passing protocol of the graph neural network. This complete model is then used to further prove the accuracy of the extracted GRNN model utilizing a mutagenesis analysis of various GRNN structures that compare the GRNN-driven PYO production with experimental data. We also explore inherited bacterial computing properties further in terms of diversity and reliability.

- **Bacterial clusters as collective single perceptrons for bio-computing**: The individual GRNNs, facilitated by the cell-cell communication, form collective behaviors essential for the population's survivability. Factors such as molecular diffusion dynamics within the environment of a bacterial ecosystem lead to variations in this collective behavior. Owing to that, we discover diversity in computing properties with respect to bacterial clusters of an ecosystem where cells in each cluster collectively resemble a single-layer perception with a non-linear activation. Consequently, we analyze the properties of these collective perceptrons spatio-temporally to extract a solution space demonstrating the inherited computation diversity.

## BACKGROUND

Past studies have explored natural bacterial computing from a number of approaches, such as using probabilistic Boolean networks (29) and logic circuits (30). All of these models mainly infer that the bacterial cells do computing not just based on the single input-output combinations but they can integrate several incoming signals to produce outputs. Moreover, recent research has demonstrated promising cell engineering approaches (31–35), especially application-specific synthetic biological circuits with neural network properties (36–38). However, the state of the art has pointed out that the process of genetic circuit designing and implementation with the possibility of performing specific tasks is a relatively complex and costly process due to the requirement of developing tools, expertise, and use of specialized materials and equipment (39,40) compared to an approach that harnesses existing circuits within the GRN. In contrast, our alternative view focuses on revealing the natural neural network structure that exists within the GRN stemming from a series of multi-stage biochemical reactions within the gene expression sequence. In this section, we first explore the NN-like properties of GRN, and this is followed by determining how the GRN can be influenced by cell-cell communications.

### Neural network properties of GRN

There are key properties commonly associated with NNs, such as interconnected nodes with a non-linear activation function that processes a weighted summation of input signals. A parallel relationship can also be found in the GRN, where its weight emerges from the

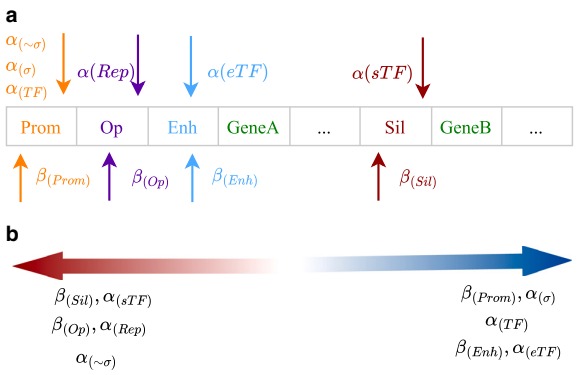



FIGURE 2 Illustration of gene expression regulators that are considered the weight influencers of the edges of GRNNs. Here, (*a*) shows the $\alpha_{(\sigma)}$, $\alpha_{(\sim\sigma)}$, $\alpha_{(TF)}$, $\alpha_{(Rep)}$, $\alpha_{(eTF)}$, and $\alpha_{(sTF)}$ are relative concentrations of sigma factors, anti-sigma factors, TFs, repressors, enhancer-binding TFs, and silencer-binding TFs, respectively. Moreover, $\beta_{(Prom)}$, $\beta_{(Op)}$, $\beta_{(Enh)}$, and $\beta_{(Sil)}$ are the binding affinities of the promoter, operator, enhancer, and silencers regions, respectively, whereas (*b*) elaborates the impact of these influencers for negative (red arrow) and positive (blue arrow) weights of the suggested model.

properties of the TFs that induce gene expressions, as well as the affinity of the TF-binding site and machineries such as thermoregulators and enhancers/silencers (41,42). The weighted summation for GRNs is dependent on multiple TFs (positive or negative weights) combined to regulate gene expression based on an activation concentration. This is illustrated in Fig. 2 *a*, which depicts a set of factors such as the relative concentration of sigma factors, anti-sigma factors, TFs, repressors, enhancer-binding TFs, and silencer-binding TFs. As shown in Fig. 2 *b*, higher binding affinities of the promoter region and enhancers induce the expression of the gene, whereas that in the operator region and silencers do the opposite. We identify this in terms of the weight of positive and negative regulations. Further, the concentrations of activator and enhancer TFs and sigma factors contribute to increased gene expression (43,44), which can be represented with higher positive weights. Conversely, the repressors and anti-sigma factors reduce the gene expression, which can be identified as larger negative weights. The non-linearity arises from the upper and lower bounds of gene expression levels, where the expression itself cannot be negative, despite the possibility that a weighted summation may result in a negative value. This non-linear relationship between the incoming weighted summation of TFs and the output gene expression resembles the rectified linear unit (ReLU) activation function. Therefore, with these characteristics, we highlight the possibility of identifying the GRN as a pre-trained NN.

Since we are deriving the GRNN from the GRN, the sub-network structure is random, with nodes that contain heterogeneous inward and outward degrees. This results in networks as well as computing diversity and this can be regulated by a single or multiple TFs. For instance, a simple graph analysis reveals genes such as *PA3477* can be regulated by up to 15 TFs, whereas *PA0576* can involve regulation of 749 genes. This heterogeneity increases the probability of mining a large number of pre-trained GRNN sub-networks.

### Influence of cell-cell communication on GRNN computing

The concentrations of molecular-input signals from the extracellular environment influences the bacterial activities at the cellular as well as the ecosystem levels (45). Apart from the extracellular signals from nutrients, it has been found that the quorum sensing (QS) input signals have a diverse set of regulative influences on bacterial gene expressions (46,47) as they are highly versatile and can respond to external bio-stress cues, providing the cell with flexibility in controlling the expression of virulence genes (48). Although the QS signals are produced by the bacteria itself, they work as inputs to the GRN similar to other external molecules, as shown in Fig. 3 *a*. Furthermore, Fig. 3 *b* shows the genes associated with the QS systems are connected to many other genes, indicating that the role of QS in GRNN computing is bidirectional, where the QS systems are influenced by various cellular activities and also regulate a range of cellular activities simultaneously. Moreover, past studies have identified that these QS gene expression pathways are interconnected and they can mutually regulate the activities of each QS system. For example, *P. aeruginosa* is known to have four QS systems, namely Las, Rhl, Pqs, and Iqs, where many mutual interactions can be observed. Among them, regulation of the Rhl, Pqs, and Iqs QS systems by the Las system and bidirectional regulation between Rhl and Pqs systems can be highlighted (49). Therefore, the computing GRNN of an isolated cell or a cell lacking QS-related genes is different from a cell within a population, and this is evident through a series of mutagenesis in silico experiments discussed in subsequent sections.

## METHODS

In this section, we first explain the GRNN extraction mechanism using a single-layer perceptron model. Our aim is first to show how we can extract a computing view by mapping the GRN to GRNN. Since bacteria live in complex ecosystems, we want to explain the dynamics of computing behavior and

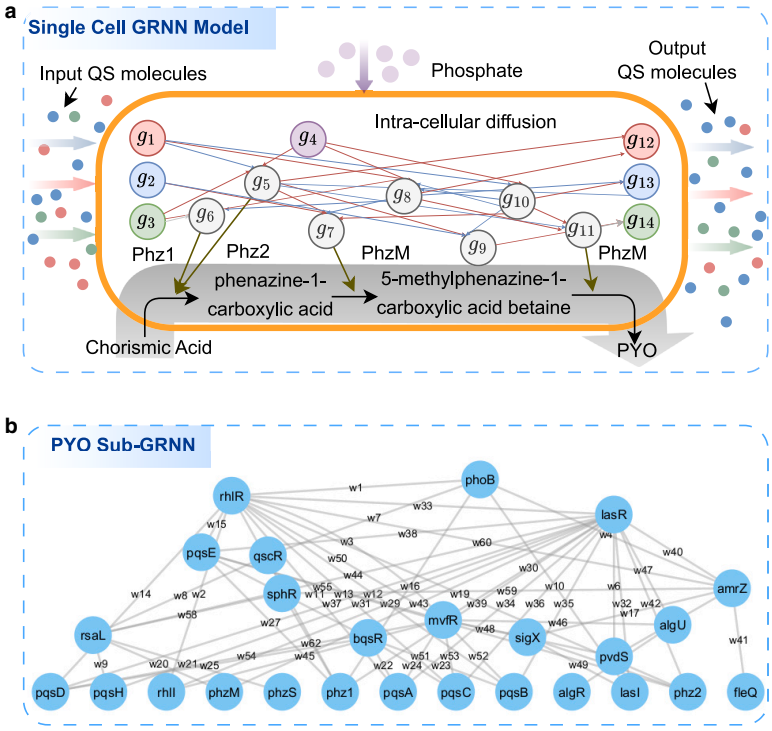



FIGURE 3 Illustration of cell-computing model of PYO production network where (*a*) depicts the computing process of the incoming cell-cell communication molecules and the conversion of chorismic acid to PYO that is driven by the GRNN outputs in response to phosphate input, whereas (*b*) shows the PYO production sub-GRNN before the weight extraction process.

how these change as the cells interact with each other and also with respect to environmental conditions. An appropriate complex ecosystem is biofilms, where cell-cell communication varies at different locations as they communicate differently. This communication affects GRNN-based computing heavily in cells, even drawing similarities to multicellular organisms (50,51). Therefore, we dedicate the second part of this section to exploring the influence of cell-cell communications within bacterial ecosystems on GRNN-based computing.

### Extracting GRNN from GRN

As we consider the GRN as a pre-trained network, the key idea of this approach is to quantify the gene-gene interaction dynamics of the GRN and interpret them as weights. Here, we first construct the GRN as a graph network of gene-gene interactions. Expression of an individual gene is mainly driven by the incoming TF signals (52) from neighboring genes and, in some cases, from the same gene. Expanding on this notion, Fig 4 *a* explains the creation of the graph network of GRN that contains five regulatory influence types using data from sources such as RegulonDB (GRN database specific to the *P. aeruginosa*) (53), Kyoto Encyclopedia of Genes and Genomes (54–56), and Ecocyc (57). Next, we disassemble this graph network into sub-graphs that consist of a target gene with its set of regulatory source genes. This sub-network is analogous to the structure of a single-layer perception with the activation function of ReLU as shown in Fig. 4 *b*, hence we call the target gene the gene perceptron. However, to date, there have not been any known methods to define weights in the gene perceptron. Therefore, in our proposed weight extraction technique of this study, we create in silico perceptrons with the same number of inputs for all target genes (gene perceptrons) in the GRN. Similar to a training process of a conventional single-layer perceptron, gene perception of this model also aims to modify the randomly assigned weights at the zeroth epoch of the gene perception based on the mean squared error (MSE) between the computed and experimental gene expression data, minimizing $TF'(g_y)$ and $TF(g_y)$. In this training process, the point with the least MSE gives the weights that best represent the quantification of the influence of each interaction on the target gene, as illustrated in Fig. 4 *c*. We then compute the perception output as follows,

$$TF'(g_y) = max\left(0, \sum_{i}^{I} TF\left(g_{x_i}\right).w_{(x_i,y)}\right), \qquad (1)$$

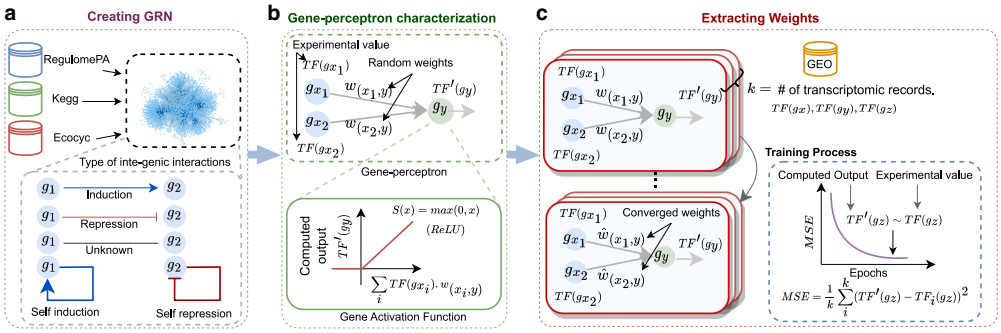



FIGURE 4 Illustration of GRNN extract steps where (*a*) is the creation of GRN structure with various interaction types between genes that is extracted from databases, (*b*) shows disassembling of the GRN into gene-perceptions with ReLU as the activation function, and (*c*) shows the weight extraction process of gene perceptrons where weights of each edge are fine-tuned by minimizing the MSE between computed $(TF'(g_z))$ and experimental $(TF(g_z))$ gene expression levels.

where $TF(g_{x_i})$ is the experimental expression values of the gene $g_{x_i}$. Moreover, $TF'(g_y)$ is the computed perception output based on the given input expression values from transcriptomic data. This model aims to train the perception until the MSE between the computed and experimental gene expression data, $TF'(g_y)$ and $TF(g_y)$, is minimized. In this training process, the point with the least MSE gives the weights that best represent the quantification of the influence of each interaction on the target gene, as illustrated in Fig. 4 *c*.

To extract the weights associated with all gene perceptrons of *P. aeruginosa*, we use transcriptomic data from the GEO database (58). After multiple stages of preprocessing of the collected transcriptomic data, 80% of it is used to extract the weights of all the gene perceptrons, whereas the remaining data are used for validation. We initialize the single-layer gene perceptrons of *P. aeruginosa* with random weight values and train them, as explained earlier. In the training process, the learning rate and the maximum number of epochs are set at $10^{-6}$ and $10^{9}$. With the extraction of the gene-gene interaction weights, the GRN is converted to a GRNN.

## Graph neural network modeling of cell-cell communication influence on GRNN computing

### *Background on graph neural networks*

Graph neural networks have emerged as a prominent approach for analyzing systems with underlying graph structures and require permutation-invariant information processing. Literature shows that graph neural networks are being used in many applications, including the prediction of protein functions (59) and identifying potential drug-target interactions (60). Another application lies in genomics, where graph neural networks have been utilized to predict gene expression patterns and identify regulatory relationships among genes (61).

Graph neural networks contain four main components: feature vectors, message-passing protocol, and aggregate as well as update functions. A feature vector is a representation of the attributes or properties of individual nodes in a graph, whereas the message-passing protocol, on the other hand, determines information flow specific to the application. For instance, diffusion properties can be embedded in the message-passing protocol of a molecule-based communication system. Next, to combine incoming information from neighboring nodes as messages, graph neural networks employ aggregation functions, which can be another application-specific function ranging from a simple summation to a complex function. The aggregated information is then processed through the update functions, which process and combine the aggregated information with the existing node attributes. These functions collectively contribute to the effective modeling and analysis of complex systems.

### *Graph neural network model for GRNN computing through cell-cell communications*

Since our aim is to extract a neural network property through the cell's GRN, we also want to determine a computing solution space of the cells' activation functions within a population. To model and understand the dynamics of this solution space, we need a framework that will enable us to model and simulate the communication between the cells and how this is computed by the GRNN, subsequently producing output molecules that influence neighboring cells. To achieve this, we model cell-cell communication using graph neural networks that have structural and functional similarities such as the random spatial

distribution of cells, cell-to-cell molecular diffusion dynamics, and modulation of cellular activities in response to incoming molecular signals.

The bacterial ecosystem is first created as a graph network where each node is a representation of a cell and the corresponding feature vector holds the computational output, which is the gene expression profile of the GRNN at a given time. The edges between the two nodes represent diffusion-based cell-cell communication (62,63) and are modeled as a message-passing protocol of the graph neural network. The summation of incoming molecular signals received by a cell is then modeled as the aggregation function. Finally, the GRNNs are embedded in each node as the update function, where the aggregated incoming molecular signals are computed, varying the gene expression patterns. This, in turn, updates the feature vector of the corresponding cell. With the embedded properties, the bacterial population represented as a graph neural network can allow an understanding of the complex interplay between cells, as well as the exchange of signaling molecules that influence cellular behavior. Ultimately, this model offers a powerful approach to unraveling the impact of cell-cell communication on bacterial behavior and uncovering underlying inherited computing properties.

As cell-cell communication plays a vital role in the computing diversity within the ecosystem depending on the cellular spatial distribution, we first define the matrix *ED* that reflects the Euclidean distances between the bacterial cells in the population as follows,

$$ED = \begin{array}{c} \\ B_1 \\ B_2 \\ \vdots \\ B_p \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & \cdots & B_P \end{array} \\ \begin{pmatrix} d(1,1) & d(1,2) & \cdots & d(1,p) \\ d(2,1) & d(2,2) & \cdots & d(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ d(P,1) & d(P,2) & \cdots & d(P,P) \end{pmatrix} \end{array}. \quad (2)$$

Here, $d_{(i,j)}$ is the Euclidean distance between the $i^{th}$ and $j^{th}$ cells where $i,j = \{1,2,...,P\}$ and $d_{(i,j)} = 0$ when $i = j$.

Moreover, the diffusion properties of the molecules also influence the cell-cell communication dynamics resulting in variation of input signals to the GRNN. For simplicity, we define a static diffusion coefficients vector $\mathbf{D}$ as,

$$\mathrm{D} = \left\{D_{m_1}, D_{m_2}, ..., D_{m_Q}\right\}, \quad (3)$$

where $D_{m_Q}$ is diffusion coefficient of molecular type $m_Q$.

If we consider a cell $B_i$ as a transmitter, we then express the molecular concentration received by cell $B_P$ that is located at $d_{(P,i)}$ after time $T$ using the Green's function as,

$$g\left(D_{m_Q}, d_{(P,i)}, T\right) = \frac{1}{\left(4\pi D_{m_Q} T\right)^{\frac{3}{2}}} \exp\left(-\frac{d^2_{(P,i)}}{4D_{m_Q} T}\right). \quad (4)$$

To calculate the incoming signals from all the cells in the network at $B_P$, we define a matrix $\mathbf{Y}_i$ that is represented as,

$$\mathrm{Y}_P = \overleftrightarrow{1}_{[Q\times 1]} \times \mathrm{ED}_P, \quad (5)$$

where $\mathrm{ED}_P$ is the $P^{th}$ row of the matrix *ED*, which represents the distance between $B_P$ and other cells. We use a $\overleftrightarrow{1}_{[Q\times 1]}$ to adjust the dimension of $\mathbf{Y}_i$ for the next iteration.

We then create the matrix $\widehat{\mathrm{g}}_P(\mathrm{D}^\top, \mathrm{Y}, t)$, which contains molecular diffusion between the $B_P$ and the rest of the cells using Eqs. 4 and 5, which is represented as follows,

$$\widehat{\mathbf{g}}_P(\mathbf{D}^\top, \mathbf{Y}, t) = \begin{bmatrix} g\left(D_{m_1}, d_{(P,1)}, T\right) & g\left(D_{m_1}, d_{(P,2)}, T\right) & \ldots & g\left(D_{m_1}, d_{(P,P)}, T\right) \\ g\left(D_{m_2}, d_{(P,1)}, T\right) & g\left(D_{m_2}, d_{(P,2)}, T\right) & \ldots & g\left(D_{m_2}, d_{(P,P)}, T\right) \\ \vdots & \vdots & \ddots & \vdots \\ g\left(D_{m_Q}, d_{(P,1)}, T\right) & g\left(D_{m_Q}, d_{(P,2)}, T\right) & \ldots & g\left(D_{m_Q}, d_{(P,P)}, T\right) \end{bmatrix}. \quad (6)$$

Since the cells secrete heterogeneous molecule types, we also need to consider the messaging signals between the cells as illustrated in Fig. 5 *a*. We represent the messaging matrix $\mathbf{MSG}^{(t)}$ for the molecular secretion of the cells at time $t$ as,

$$MSG^{(t)} = \begin{array}{c} \\ B_1 \\ B_2 \\ \vdots \\ B_p \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & \cdots & m_Q \end{array} \\ \begin{pmatrix} msg^{(t)}_{(1,m_1)} & msg^{(t)}_{(1,m_2)} & \cdots & msg^{(t)}_{(1,m_Q)} \\ msg^{(t)}_{(2,m_1)} & msg^{(t)}_{(2,m_2)} & \cdots & msg^{(t)}_{(2,m_Q)} \\ \vdots & \vdots & \ddots & \vdots \\ msg^{(t)}_{(P,m_1)} & msg^{(t)}_{(P,m_2)} & \cdots & msg^{(t)}_{(P,m_Q)} \end{pmatrix} \end{array}, \quad (7)$$

where $msg^{(t)}_{(P,m_Q)}$ is the secreted concentration of molecule type $m_Q$ produced by cell $B_P$.

Now, we define the incoming signal vector $\mathrm{R}_P^{(t+1)}$ for the bacterial cell $B_P$, which contains the concentrations of all the molecule types at time $TS = t+1$, using Eqs. 6 and 7 as follows,

$$\mathbf{R}_P^{(t+1)} = diag\Big(\widehat{\mathbf{g}}(\mathbf{D}^\top, \mathbf{Y}, t) \times \mathbf{MSG}^{(t)}\Big) = \Big[r_{(P,m_Q)}^{(t+1)}\Big]_{1\times Q}, \tag{8}$$

where $r_{(P,m_Q)}^{(t+1)}$ is the received $m_Q$ signal by the cell $B_P$.

Despite the GRNN of $B_P$ receiving molecular signals from the peer cells, the nutrient in the extracellular environment and accumulated molecules in the cytoplasm also act as inputs of the same GRNN. Therefore, we further define the accumulated intra-cellular molecular concentrations $\mathrm{IM}$ at time $t$ as,

$$\mathbf{IM}^{(t)} = \begin{array}{c} \\ B_1 \\ B_2 \\ \vdots \\ B_P \end{array} \begin{array}{c} im_1 im_2 \ldots im_Q \\ \begin{pmatrix} C_{(1,im_1)}^{(t)} & C_{(1,im_2)}^{(t)} & \ldots & C_{(1,im_Q)}^{(t)} \\ C_{(2,im_1)}^{(t)} & C_{(2,im_2)}^{(t)} & \ldots & C_{(2,im_Q)}^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ C_{(P,im_1)}^{(t)} & C_{(P,im_2)}^{(t)} & \ldots & C_{(P,im_Q)}^{(t)} \end{pmatrix} \end{array}, \tag{9}$$

where $C_{(P,im_Q)}^{(t)}$ is accumulated concentration of molecule type $m_Q$ in the cytoplasm of cell $P$.

Further, we integrate our single-cell GRNN and cell-cell communication model into a 3D environment to incorporate the external molecular inputs. The environment of the simulation is designed as a 3D grid of voxels that can store precise information on external nutrients (based on our previous model in (64)). The diffusion of nutrient molecules through the medium is modeled as a random-walk process (27).

In the environment, the nutrient concentrations at the location of cell $B_P$ at time $TS = t$, $\mathrm{K}_P^{(t)}$, is denoted as,

$$\mathbf{K}_P^{(t)} = \Big\{K_{P:m_1}^{(t)}, K_{P:m_2}^{(t)}, \ldots, K_{P:m_Q}^{(t)}\Big\} == \Big[K_{P:m_Q}^{(t)}\Big]_{1\times Q}, \tag{10}$$

where $K_{(P:m_Q)}^{(t)}$ is concentration of the molecular type $m_Q$ in a specific location.

As the next step, we define the aggregation array $S_P^{(t)}$ that contains the summation of all the molecules received by cell $P$ as,

$$\begin{aligned} \mathbf{S}_P^{(t)} &= \mathbf{R}_P^{(t)} + \mathbf{K}_P^{(t)} + IM_P^{(t)} \\ &= \Big[r_{(P,m_Q)}^{(t+1)} + K_{(P:m_Q)}^{(t)} + C_{(P,im_Q)}^{(t)}\Big]_{1\times Q}, \end{aligned} \tag{11}$$

which is further illustrated in Fig. 5 *b*.

The aggregated signals are then computed through the GRNN of each node, where the output is observed in the updated gene expression profile or the feature vector of the corresponding node. This is expressed as follows,

$$\mathrm{FV}_P^{(t+1)} = GRNN\Big(\mathrm{S}_P^{(t)}\Big), \tag{12}$$

where $FV_P^{(t+1)}$ is the cell $B_P$'s array of computed gene expression levels by the GRNN upon the reception of $\mathrm{S}_P^{(t)}$. This mathematical expression explains a bacterial cell's behavior of adaptive gene expression dynamics with respect to the incoming molecular signals.

Finally, the matrix $\mathrm{FV}^{(t)}$ that is obtained by Eq. 12, which denotes the population's computational output at $TS = t$ in terms of gene expression levels, is represented as,

$$\mathbf{FV}^{(t)} = \begin{array}{c} \\ B_1 \\ B_2 \\ \vdots \\ B_P \end{array} \begin{array}{c} g_1 g_2 \ldots g_L \\ \begin{pmatrix} b_{(1,g_1)}^{(t)} & b_{(1,g_2)}^{(t)} & \ldots & b_{(1,g_L)}^{(t)} \\ b_{(2,g_1)}^{(t)} & b_{(2,g_2)}^{(t)} & \ldots & b_{(2,g_L)}^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(P,g_1)}^{(t)} & b_{(P,g_2)}^{(t)} & \ldots & b_{(P,g_L)}^{(t)} \end{pmatrix} \end{array}, \tag{13}$$

where $b_{(P,g_L)}^{(t)}$ is the expression level of the gene $g_L$ of the cell $B_P$ at the same time slot.

The significance of this modeling approach is to allow us to characterize dynamics of GRNN computing of bacterial cells in the ecosystem and to understand how they compute in parallel within the community. As a population, this results in a large parallel processing framework. To reflect this, we use the python-cuda platform to mimic our model as close to the parallel processing architecture of the biofilm, where we dedicate a graphics processing unit block for each bacterial cell and the threads of each block for the matrix multiplication of the GRNN computation. Additionally, due to the high number of iterative components in the model, the computational power demand faces significant challenges with serial programming, making parallelization the best match for the model.

## RESULTS

In this section, we conduct simulations of the bacterial ecosystem within a biofilm for the extracted GRNN computing to determine its accuracy and its dynamics based on the framework described in the previous section. We first describe the simulation setup of the GRNN as well as the biofilm model. This is followed by analyzing the accuracy of the extracted GRNN and its computing behavior by comparing to wet-lab experimental data. This includes analyzing the network structure of the GRNN based on mutagenesis, followed by the reliability of individual GRNN and bio-computing models of the biofilm as bacterial

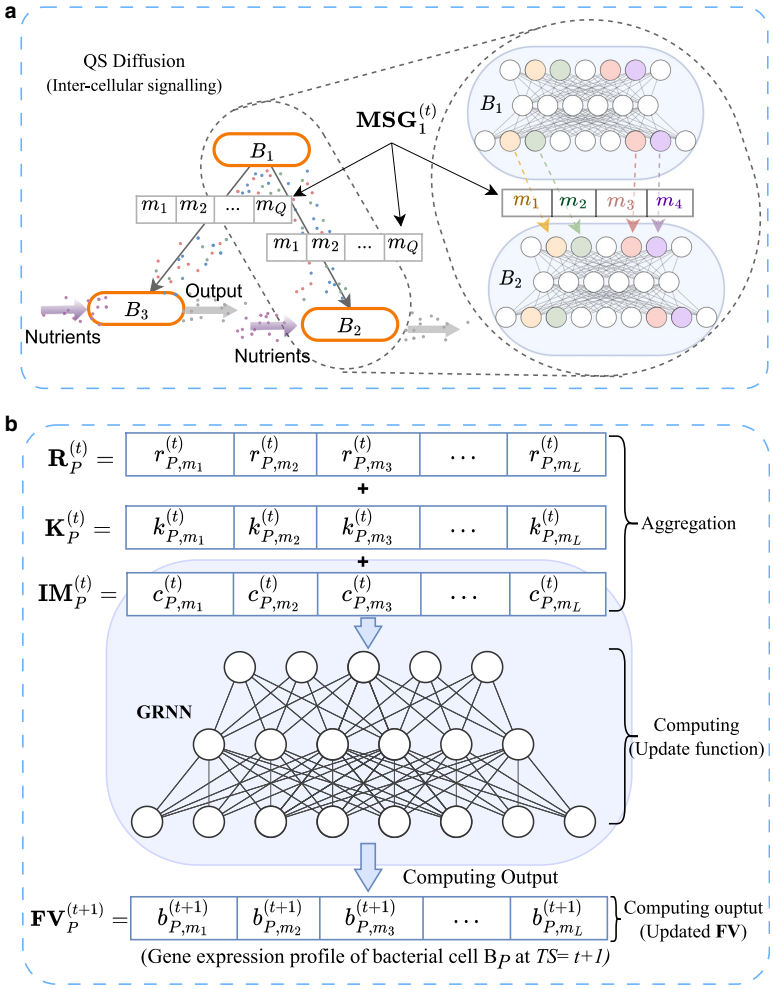



FIGURE 5 The process of one GRNN outputs reaching another GRNN as molecular messages where (*a*) illustrates how the GRNN output of one cell influences the others and (*b*) explains how the graph neural networks are utilized to model the computing of bacterial cells.

cluster-based perceptrons and the dynamics of their sigmoidal activation based on temporal and spatial position.

### Simulation setup

The species *P. aeruginosa* has been extensively studied as it is known for posing serious health problems such as pneumonia, blood infections, infected wounds, and especially the production of PYO, which is a toxin that affects human cell functions. Thus, our study focuses on the GRNN computing that results in PYO production.

*Genetic model*

The literature provides information on the genetic network that incorporates PYO production, including the genes that are affected by phosphate intake and QS signaling (65). First, we extract the PYO sub-GRNN using the shortest path analysis that defines a network for the interactions of QS-related genes and the two components system (TCS) *PhoR-PhoB* that governs genes expressed from the phosphate intake (*phZ1*, *phz2*, *phzS*, and *phzM*), which are responsible for the production of enzymes that are essential for PYO production.

In this computational model, we identify another layer of metabolic interaction that plays a role in PYO production, which is shown in Fig. 6. Since our primary goal is to explore the neural network behaviors of GRNs, we model these inter-cellular metabolic interactions as a separate layer from the GRNN. Here, RhlR is a transcriptional regulator of *P. aeruginosa* and forms a complex by getting attached to its cognate inducer

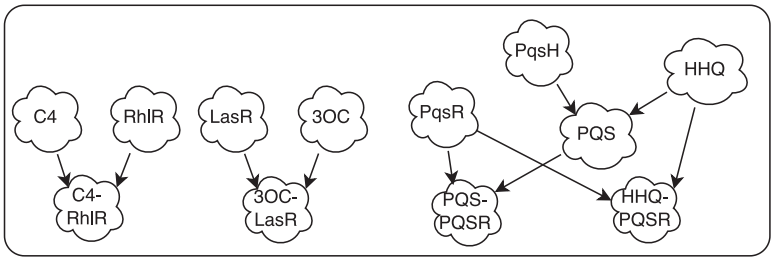



FIGURE 6 Illustrations of intra-cellular metabolite interaction where the QS molecules form complexes with response regulators.

C4-HSL, which in turn works as an input to the GRNN (66). Similarly, LasR transcriptional regulator protein and 3-oxo-C12-HSL (3OC), and PqsR with PQS and HHQ, form complexes that also act as inputs to the GRNN (67,68). In parallel to this process, chorismic acid, $C_{10}H_{10}O_6$ in the environment, are converted by the *P. aeruginosa* cells through multiple steps using the *phz1*, *phz2*, *phzS*, and *phzM* products of the GRNN. First, the $C_{10}H_{10}O_6$ is converted into phenazine-1-carboxylic using the enzymes produced from *Phz1* and *Phz2* genes that are outputs of the GRNN. In the next step, the phenazine-1-carboxylic gets converted into 5-methylphenazine-1-carboxylate, and finally, 5-methylphenazine-1-carboxylate into PYO by the GRNN outputs PhzM and PhzS, respectively (69). Therefore, the GRNN computing will in turn modulate and convert the $C_{10}H_{10}O_6$ into PYO, as illustrated in Fig. 3 *a*.

*Biofilm model*

One of the key differences between cell-cell communication within a biofilm and a collection of planktonic cells is caused by the diffusivity of extra polymeric substances (EPSs). This, in turn, creates interesting cell-cell communication patterns that lead to computing variations in a biofilm, which we show through the mutual information showing the uncertainty in information flow during computing changes. The internal availability and consumption of the nutrients transform the bacterial communication process, and this, in turn, changes the computing behavior. Therefore, in this study, we consider a *P. aeruginosa* single-species biofilm as an ecosystem to investigate the role of inter-cellular communication in GRNN computing. We model a completely formed biofilm and disregard the forming, maturation, and dispersion stages, which are out of the scope of this study. In our model, we consider the biofilm as a static 3D structure of bacterial cells. We first place bacterial cells randomly in a paraboloid-shaped structure using the equation, $z < \frac{x^2}{5} + \frac{y^2}{5} + 20$, where $x$, $y$, and $z$ are the components of 3D Cartesian coordinates. This paraboloid shape is chosen to make the spatial arrangement of the cells close to a real biofilm and keep the cell placement process simple. Within this 3D biofilm structure, we model the diffusivity based on $D_B/D_{aq} = 0.4$, which is the mean relative diffusion (70), where $D_B$ and $D_{aq}$ are the average molecular diffusion coefficients of the biofilm and pure water, respectively. To start the simulation at a stage where the biofilm is fully formed with established communication between the cells, we filled the graph neural network internal memory vector of each cell with the average molecular level at the initial time slot. Each bacterial cell will use the initial signals from the internal memory and apply it to the GRNN to compute and update the feature vector for the next time slot. Table 1 presents the parameter descriptions and values used for the simulation. As shown in Table 1, the model runs for 150 time slots, generating data for a range of functions in the system. These functions can produce data on the graph neural network feature vector of each cell, communication between cells, molecular consumption of the cells, secretion of molecular output to the environment, and nutrient accessibility of cells for each time slot.

### Validating the GRNN accuracy for PYO production

The accuracy of the extracted weights is validated by predicting output expression levels for each gene perceptron using the transcriptomic data from publicly available experimental data in the GEO database (58).

The first analysis is to determine the accuracy of the full GRNN, which contains 2851 genes and 4903 interaction links. In calculating the weights, we used 217 transcriptomic data records (58). Using the weights that are allocated to each link of the gene expression relationship, we predict the output expression levels of each gene using Eq. 1 and compare them to the measured values. The results of this analysis shown in Fig. 7 show that the majority of the data points lie close to the 45° line, indicating an accurate prediction. Note that the deviated points may reflect the variability of the weights, which is not investigated in this study.

Moreover, we further investigate the accuracy of the extracted GRNN through a mutagenesis analysis by

**TABLE 1 Parameters utilized in the system development**

| Parameter | Value | Description |
|---|---|---|
| No. of cells | 2000 | the number of cells is limited due to the memory availability of the server |
| No. of genes | 26 | the network only consists of the gene are directly associated with QS, *PhoR-PhoB* TCS and PYO production |
| No. internal memory molecules | 16 | molecules involved in QS and *PhoR-PhoB* TCS and PYO production |
| No. messenger molecules | 4 | number of molecules that were exchanged between cells in the sub-network |
| Dimensions of the environment | $20 \times 20 \times 20\ \mu m$ | the dimensions were fixed considering the average sizes of *P. aeruginosa* biofilms and computational demand of the model |
| Duration | 150 TSs | the number of TSs can be modified to explore the cellular and ecosystem level activities. For this experiment, we fixed a TS to represent 30 mins |
| No. iterations per setup | 10 | considering the stochasticity ranging from the gene expression (71,72) to ecosystem-wide communications, the experiments were iterated 10 times |

doing modifications to the GRNN structure and observing the corresponding gene expression and PYO production outputs. This simulation experiment is performed for two levels of phosphates (high phosphate (HP) and low phosphate (LP)). Besides the different levels of phosphate, we also aim to analyze how changes in the network structure due to mutations can affect the computing behavior. We conducted eight simulation experiments with the following setup: 1) wild-type bacteria with no mutations (WD) in LP, 2) *lasR* mutant (*lasR* Δ) in LP, 3) *phoB* mutant (*phoB* Δ) in LP, 4) *lasR* and PhoB double mutant (*LasR* Δ *PhoB* Δ) in LP, 5) WD in HP, 6) *lasR* Δ in HP, 7) *PhoB* Δ in HP, and 8) *LasR* Δ *PhoB* Δ in HP. Although the WD uses the complete PYO sub-GRNN, *lasR* Δ results in removal of the node *lasR*, the GRNN of *phoB* Δ is modified by removing the *PhoB*, and the double mutant (*LasR* Δ *PhoB* Δ) is modified by removing both *lasR* and *PhoB* genes, as shown in GRNNs of Fig. 8 *a–d*, respectively. This mutation results in structural changes of the GRNN that alter the computational outputs, which can be observed through the gene expression and PYO production levels.

For the mutagenesis cases described above, we will compare the computed values through the GRNN with wet-lab experimental data from (73) for the PYO production and show the molecular output behavior of the corresponding GRNN structures. The computational output of PYO level in the *P. aeruginosa* biofilm is high in LP compared to HP for all the four cases (Fig. 8 *a–d*), where the highest difference is in *lasR* Δ case and the lowest in *phoB* Δ, as shown in Fig. 8 *b* and *c* respectively. This is mainly due to the negative impact of the gene *phoB* on the other genes, where an increased expression of the gene *phoB* due to higher phosphate condition represses the expression of genes, including *rhlR* (with the weight, $w_{(phoB,rhlT)} \approx -0.31$) and *phZ1* (with the weight, $w_{(phoB,phz1)} \approx -0.33$), which in turn reduce the overall gene expression levels of the GRNN.

This effect is magnified by *lasR* mutation, as shown in Fig. 8 *b*, and we identify that the gene *mvfR* plays a crucial role in PYO production output in this case. The gene *mvfR* positively expresses seven other genes in this network and, in turn, a higher expression of *mvfR* results in increased PYO production. In this particular case, expression of the gene *mvfR* is increased in LP

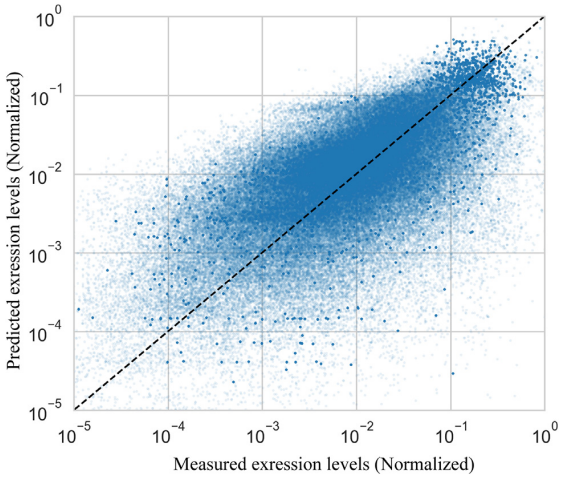



FIGURE 7 Comparison between measured expression levels of 2851 genes for 217 transcription records and gene expression values computed by the extracted full GRNN.

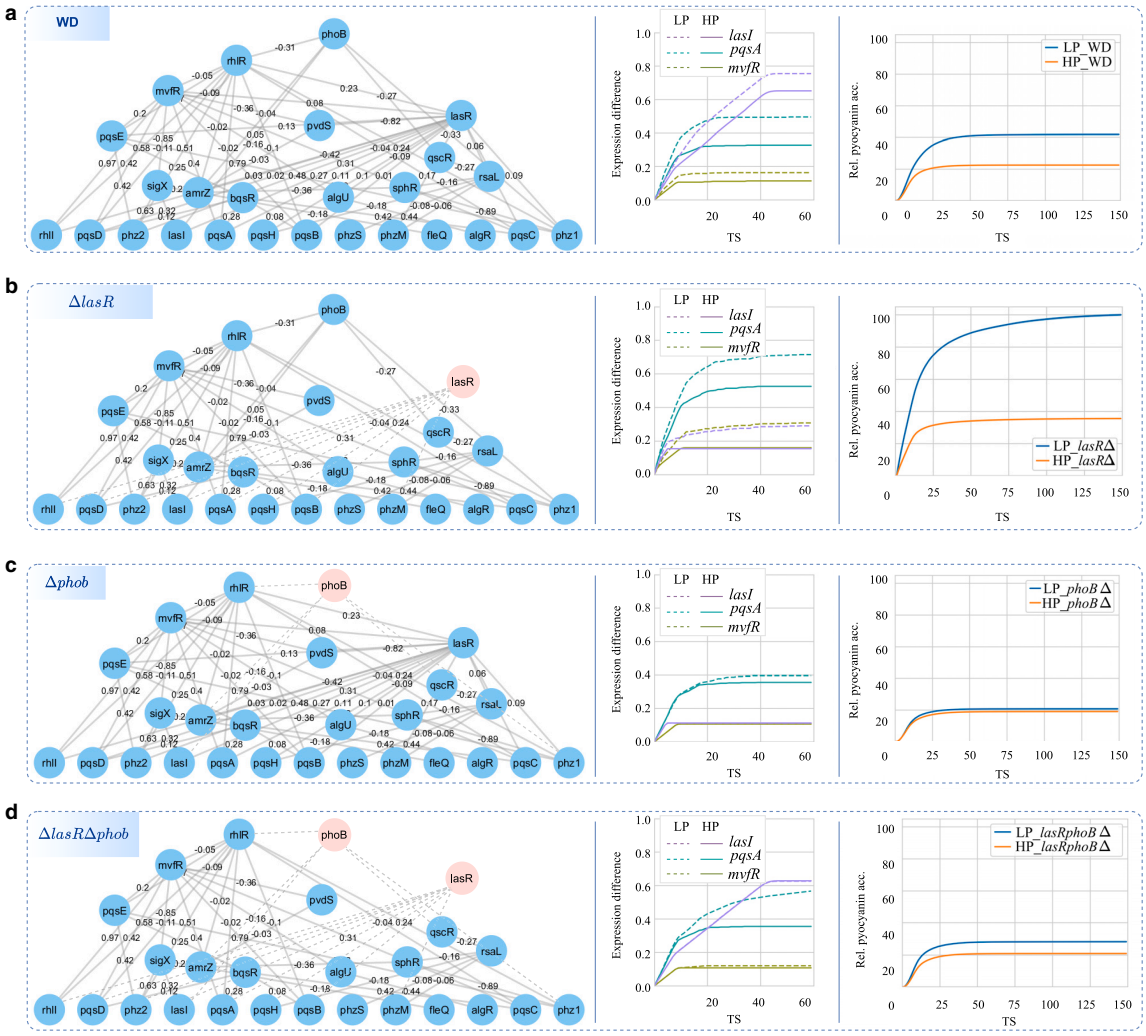



FIGURE 8 Mutagenesis analysis to investigate the impact of GRNN structural deviation on the PYO production. The gene expression variations and resulting PYO production are shown in LP and HP for four GRNN structures of (*a*) WD, (*b*) Δ*lasR*, (*c*) Δ*phoB*, and (*d*) Δ*lasR*Δ*phoB*. The genes highlighted in the red circles are removed from the GRNN for the different structure types we consider to show the structural changes in the network, which in turn shows how computing changes in the PYO production.

as evident in the gene expression plot of Fig. 8 *b* due to the reduced level of *rhlR* expression (which represses the *mvfR* with the weight $w_{(rhlR,mvfR)} \approx -0.05$) as a result of lacking *lasR* (which induces the gene *rhlR* with the weight $w_{(lasR,rhlR)} \approx 0.23$). This mutation can be considered an improvement of GRNN's sensitivity to environmental phosphate concentrations.

In contrast, the lowest difference in LP and HP PYO production of case *phoB* Δ occurs as a result of GRNN's direct insensitivity to phosphate variations formed by the removal of the gene *phoB*, which is the two-component response regulator associated with phosphate intake. This insensitivity contributes to maintaining a fixed gene expression level in the GRNN despite the environmental phosphate variations. Moreover, the difference between the PYO production in LP and HP of the *LasR* Δ *PhoB* Δ case is high compared to *PhoB* Δ, as shown in Fig. 8 *d*, which can be explained as a combined impact of *lasR* Δ and *PhoB* Δ cases on PYO productions where lacking gene *phoB* reduces the phosphate sensitivity of GRNN and lacking gene *lasR* does the opposite. These results emphasize the heterogeneity in the weight distribution of edges and the importance of nodes that is advantageous in extracting application-specific GRNNs in the future.

Our comparison to the wet-lab experimental data to analyze the GRNN computing behavior (73) is based on the ratios of HP to LP for PYO production, as shown in Fig. 9. The differences between the PYO production predicted from GRNN in HP and LP conditions for all four setups in Fig. 9 are relatively close to the wet-lab experimental data. Even though the absolute percentages are not the same between wet-lab and simulated values, the ratios between them are significantly close. We believe deviation of the absolute values is caused by the lack of interactions with gene expression outside our sub-network having an influence on the GRNN. Although the overall comparison in Fig. 9 is considered accurate, we believe this can be further improved with increased association to genes that neighbor the sub-network and increasing the accuracy of the weight calculations using increased transcriptomic data. The modified GRNN structures based on gene mutants for accuracy testing also lay the foundation for us to modify network structures through genetic engineering to create an neural network that fits our target problem for bio-computing applications.

### GRNN computing reliability

Biological systems such as bacterial ecosystems are influenced by many stochastic factors, such as molecular diffusion generated from diverse molecules (74). As a population of cells within the biofilm, this affects the computational paths along the GRNN of each cell. The nutrient molecular diffusion within the biofilm is the first observed using a 3D environment layer. The EPS is known to provide extra protection to the biofilm and it has been proved that it also resists nutrient penetration toward the core of the structure (75). This results in a gradient that reflects the nutrient accessibility variations in the biofilm, which is illustrated in Fig. 10. The cells in the core of the biofilm have lower nutrient accessibility compared to the cells at the periphery, and this is due to variations in diffusion between the environment and the EPS. Fig. 10 *a* compares the flow of nutrients with respect

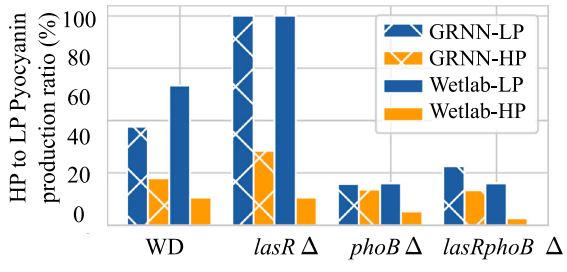



FIGURE 9 Evaluation of the model accuracy by comparing HP to LP PYO production ratio with wet-lab data from (73).

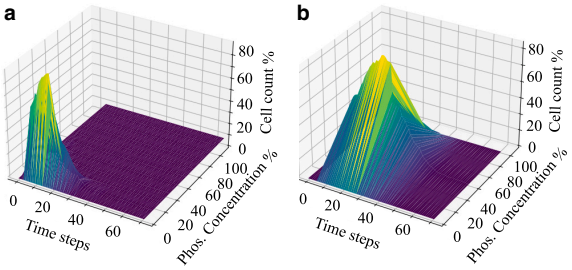



FIGURE 10 The nutrient accessibility variations of cells is expressed in two different environment conditions: (*a*) LP and (*b*) HP concentrations.

to time and concentration for low input concentration, whereas Fig. 10 *b* is for high input concentrations. As shown in the figures, when an LP concentration is introduced, access to the nutrients is significantly limited. This variability in nutrient accessibility plays a significant role in the reliability of the GRNNs and the diverse computing behavior in the biofilm. This, in turn, enables us to take control over the computing dynamics up to a certain extent, which is beneficial in tailoring GRNNs to specific applications.

To analyze the reliability and functionality of the GRNNs, we use mutual information (MI) to statistically measure the dependency between the input and output gene expressions in the GRNN, which reflects the network's computing reliability. As the expression levels of the input and output genes are continuous variables that show the GRNN's analog computing behavior, we use the Gaussian kernel density-based MI estimation with the well-known Silverman's rule for kernel bandwidths selection that is presented in Algorithm 1. Although the diffusion dynamics within the EPS form a continuous nutrient gradient toward the biofilm core, we discretize it into three regions (region 0, the core; region 1, the middle layer; and region 2, the outer layer). An increased number of regions can result in extracting more layers of variations, but, at the same time, it will be challenging to measure or use for computing tasks in practice due to the physical scale of the system. In contrast, the minimized number of regions can provide improved differentiation between the behaviors of each layer. To measure the variations in computing, we estimate the MI of the GRNN in three layers of the biofilm as shown in Fig. 11 and for three time slots ($TS = 20$, $TS = 25$, and $TS = 30$).

Fig. 12 presents the MI results with respect to three regions of the biofilm and time slots. Here, we only focus on the MI behavior of the following five GRNN edges, which includes *phob-rhlR*, *rhlR-phz1*, *phoB-phz1*, *phoB-lasI*, and *bqsR-pqsC*. The significance of these five edges is the information flows through

Algorithm 1 Estimating MI using Gaussian kernel density estimation

**Input** :Two continuous variables $X$ and $Y$, and a set of $N$ samples $(x_i, y_i)$, $i = 1, \ldots, N$
**Output** The mutual information $I(X;Y)$
:
**Function** SilvermanBandwidthSelection $X$:
  $n = |X|$, number of data points in;
  $\sigma$ = standard deviation of $X$;
  $h = \frac{1.06 \cdot \sigma}{n^{1/5}}$;
  **return** $h$;
**Function** GaussianKDE $(X, h)$:
  $kde$ = empty array;
  **for** $x \in X$ **do**
    $kde(x) = \frac{1}{n \cdot h} \sum_{x_i \in X} \exp\left(-\frac{(x - x_i)^2}{2h^2}\right)$;
  **end**
  **return** $kde$;
**Function** GaussianJointKDE $(X, Y, h_X, h_Y)$:
  $kde$ = empty array;
  $\mathbf{K}(\mathbf{u}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$;
  **for** $x \in X$ **do**
    $kde(x_i, y_i) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h_X h_Y} K\left(\frac{x_i - x_j}{h_X}\right) K\left(\frac{y_i - y_j}{h_Y}\right)$;
  **end**
  **return** $kde$;
$h_X$ = SilvermanBandwidthSelection$(X)$;
$h_Y$ = SilvermanBandwidthSelection$(Y)$;
$f_{XY}$ = GaussianJointKDE$(X, Y, h_X, h_Y)$;
$f_X$ = GaussianKDE$(X, h_X)$;
$f_Y$ = GaussianKDE$(Y, h_Y)$;
**Function** MutualInformation $(f_{XY}, f_X, f_Y)$:
  $MI = 0$;
  **for** $x_i \in X, y_i \in Y$ **do**;
    $MI = MI + f_{XY}(x_i, y_i) \cdot \log_2\left(\frac{f_{XY}(x_i, y_i)}{f_X(x_1) \cdot f_Y(y_i)}\right)$;
  **end**
  **return** $MI$;

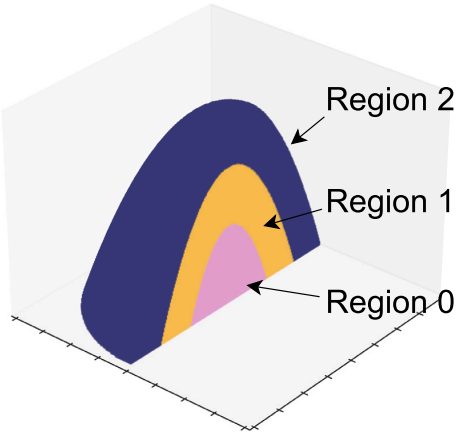



FIGURE 11 Illustration of three layers considered in the biofilm to investigate the computing reliability and the solution space, where region 2 is the outer layer, which has the most access to nutrients. Region 1 is the middle layer, and region 0 is the core with the least nutrient access.

the expressions of these gene pairs, which have the highest variance and the largest MI values. Further, these edges are highlighted with different colors where the thickness corresponds to their MI value. In the considered GRNN, the involvement of the QS signals is clearly visible, as all four edges with the highest MI values except for the edge between *bqsR-pqsC* genes are associated with the QS systems. It is evident that region 2, which is the outer layer of the biofilm, has the GRNNs with better information flow indicated by significantly higher MI values than the other regions. This reflects the high reliability of the GRNN computing since the outputs have a strong dependency on the inputs. However, over time, the MI diminishes gradually with the nutrient reduction of the environment due to the consumption by the bacteria. This is evident in region 1, where the diminished nutrient accessibility results in lower MI values, showing higher uncertainty in the computing process. The uncertainty in the computing process is further amplified in region 0, which has the lowest MI values as the information flow of GRNN


FIGURE 12 Illustration of MI of the solution space in three regions and time steps. The colors of the MI bar plots are mapped to the colors of the corresponding edges. The widths and the arrows of the edges represent the MI values and direction of information flow, respectively. The nodes in the yellow color are considered the outputs as they produce enzymes related PYO production, whereas the pink nodes represent the input nodes associated with phosphate and QS molecule intake.

is more dependent on the stochasticity of gene expressions over incoming nutrient signals in this region. In region 2, at $TS = 20$, the MI value between expressions of genes *phoB* and *rhlR* is the same as that between genes *rhlR* and *phz1*. However, as we consider different time points and regions, it becomes evident that the MI between genes *phoB* and *rhlR* is reduced more compared to MI between gene *rhlR* and *phz1*, indicating that the impact of phosphate on the *RHL* QS system is stronger. This shows that the MI analysis can be used in the future to identify reliable sub-networks for bio-computing applications.

The analysis indicates a high reliability of GRNN computing near the surface of the biofilm, where the output response exhibits a strong dependence on the input signals compared to other regions. This suggests that the computing outputs of the cells closer to the biofilm core are highly stochastic and fluctuate at a higher rate during decision making. The impact of the network's reliability on the outputs can also be observed in the analysis in the next section, when we investigate the cells cooperating to form collective perceptrons.

### Cluster-scale collective perceptrons

The output patterns of the GRNNs of individual cells revealed that, in the biofilm, bacteria collectively form a set of non-linear output functions spatio-temporally with the help of cell-cell communication, which we modeled through the graph neural network. Hence, in this section, properties of the output non-linearity of regions and time points of the biofilm that resulted from the cluster of cells with GRNN in each is investigated in terms of a sigmoid activation function $S(x)$,

$$S(x) = \frac{L}{1 + e^{-(kx - x_0)}}, \tag{14}$$

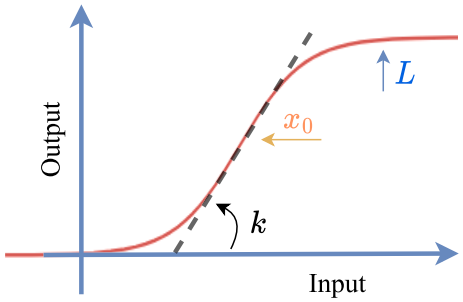



FIGURE 13 Illustration of $L$, $k$, and $x_0$ parameters that determine the height, steepness, and horizontal shift of the sigmoid curve, respectively.

where the parameters $L$, $k$, and $x_0$ control the maximum, steepness, and horizontal shift, respectively, as shown in Fig. 13.

The Hill function is also used in the literature to represent biological activities such as gene expressions and chemical reactions. However, the Hill function is almost indistinguishable from the sigmoid function when the Hill coefficient is greater than 1 for the proper choice of parameters (76). Nevertheless, from the neural network computing perspective, sigmoid functions are widely used, especially in artificial perceptrons. Therefore, we explore the output non-linearity of the collective perceptron with sigmoid functions.

The PYO production of the biofilm is analyzed in the same three regions (shown in Fig. 11) for three time slots ($TS = 20$, $TS = 25$, and $TS = 30$) and extract a solution space with a set of sigmoid activation function variations. Further, we analyze their dynamics based on the role of QS. Literature shows the attempts to use modified versions of activation functions such as scaled sigmoid, penalized Tanh, and bounded ReLu can be tailored for specific computational tasks (77). Subsequently, it has been proved that the improved versions of the standard non-linear activation function comparatively perform well with respect to the application problems (78–80). Therefore, a biological entity that contains a diverse set of non-linear functions can be advantageous in computing applications such as adaptive classifications or analog to digital conversions with more specificity and adaptability.

We used the curve-fitting approach presented in Algorithm 2 to determine the parameters of Eq. 14 and how it, as well as the QS molecules, dynamically changes with respect to variations in the phosphate input. We analyze these values for each region, which is presented in Fig. 14. The top row of Fig. 14 shows the collective non-linear properties in region 2 of the biofilm outer layer, where the nutrient accessibility is relatively high. The higher nutrient availability is positively reflected in the high QS levels in the region compared to region 1 and region 0. This, in turn, results in a higher $k$ value of $S(x)$, which governs the steepness of the sigmoid function. Over time, the QS concentration gets significantly reduced, and the steepness $k$ subsequently increases. In contrast, the parameter $x_0$, which governs the horizontal shift of

Algorithm 2 Curve fitting with least squares method

**Input:** Data points $(x_i, y_i)$ for $i = 1, 2, \ldots, n$
**Output:**Optimized parameter values
**Function** CurveFitting $(X, Y)$:
  Initialize the set of parameter values $\mathbf{P} = (L, k, x_0)$;
  **repeat**
    Compute the predicted values $\widehat{Y} = \text{MSigmoid}(X, \mathbf{P})$;
    Update the parameter values $\mathbf{P} = \text{LeastSquares}(X, Y, \widehat{Y})$;
  **until** *convergence*;
  **return** $\mathbf{P}$;
**Function** MSigmoid $(X, P)$:
  Extract parameter values $L, k, x_0$ from $\mathbf{P}$;
  Compute the predicted values $\widehat{Y}$ using the sigmoid function: $\widehat{Y} = \frac{L}{1+e^{-(kX-x_0)}}$;
  **return** $\widehat{Y}$;
**Function** LeastSquares $(X, Y, \widehat{Y})$:
  Compute the residual vector $R = Y - \widehat{Y}$:
  Compute the Jacobian matrix $\mathbf{J}$ with partial derivatives of the logistic function w.r.t. the parameters;
  Compute the parameter updates $\Delta\mathbf{P}$ using the least squares formula: $\Delta\mathbf{P} = (\mathbf{J}^\top\mathbf{J})^{-1}\mathbf{J}^\top R$;
  $\mathbf{P} = \mathbf{P} + \Delta\mathbf{P}$;
  **return** $\mathbf{P}$;
$\mathbf{P} =$ CurveFitting $(X, Y)$;

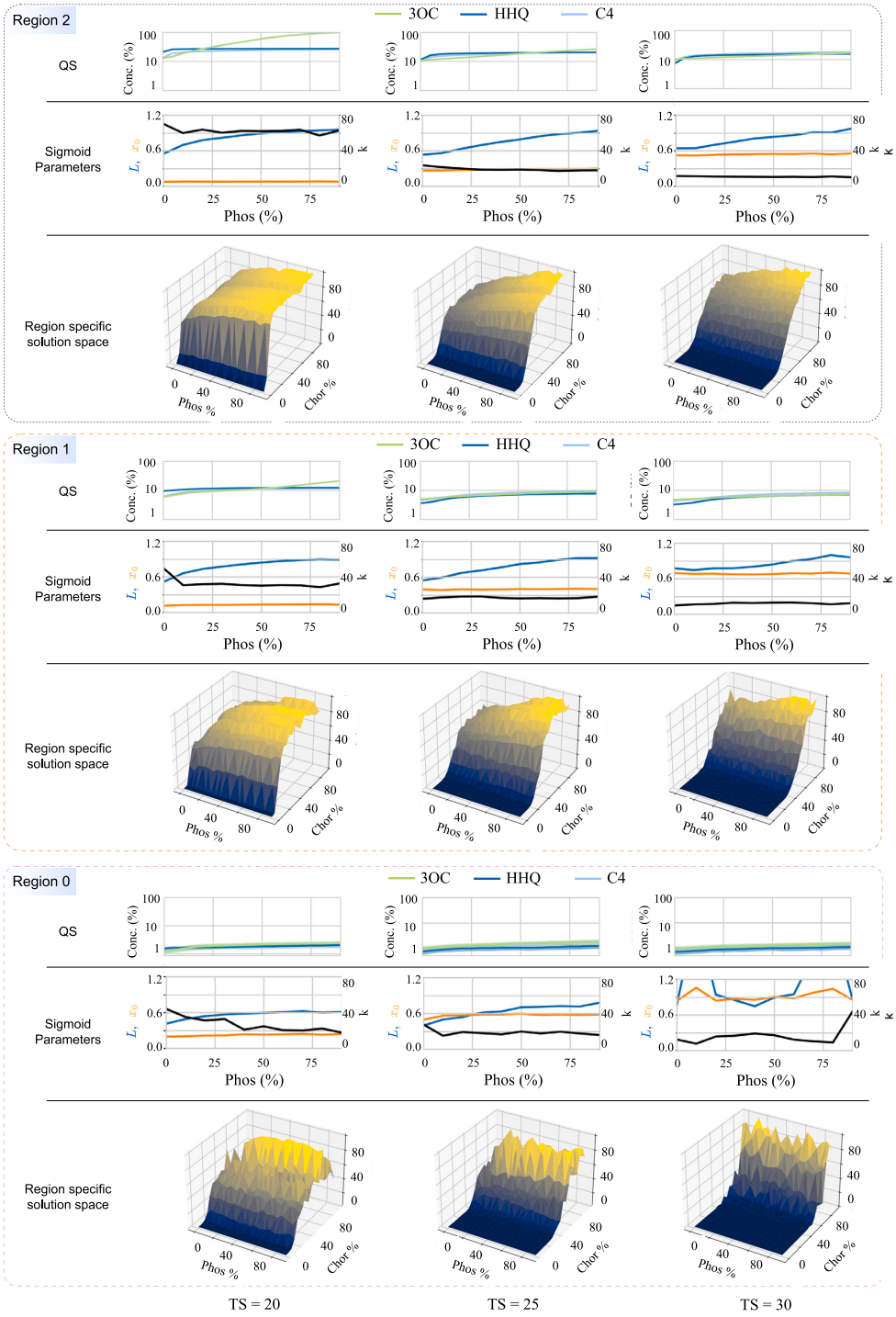



FIGURE 14 Illustration of a sigmoid function-based solution space found in biofilms, where variations in non-linear behavior are shown with respect to location in each column and time in each row. Each layer of the diagram consists of QS, sigmoid parameter, and sigmoid curve variation plots corresponding to the regions of the biofilm. The QS plots show the percentage differences of 3OC, HHQ, and C4 QS signal concentrations and sigmoid parameters plots show the changes in the height ($L$), steepness ($k$), and horizontal shift ($x_0$) of the function.

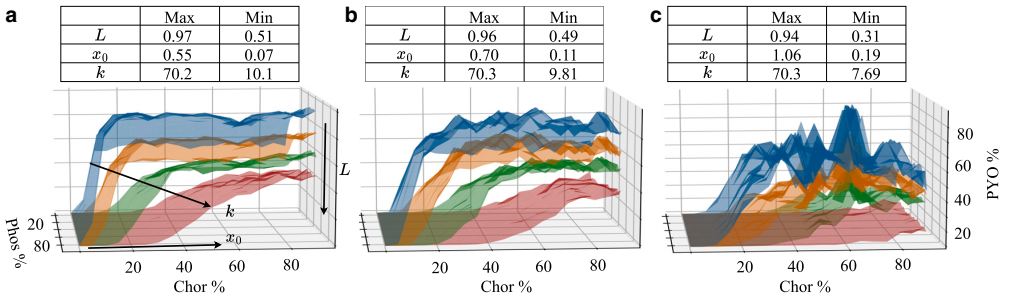



FIGURE 15 Variations of the sigmoid activation function in (*a*) region 2, (*b*) region 1, and (*c*) region 0 based on inputs of phosphate and chorismic acid. Each plot contains four time slots: $TS = 15$ in blue, $TS = 20$ in orange, $TS = 25$ in green, and $TS = 30$ in red. The table of each plot gives the ranges of each parameter for the corresponding region.

the curve, has a negative relationship with the QS concentration. The highest $k$ and the lowest $x_0$ can be observed in region 2 at $TS = 20$ as a result of the highest concentration of QS. It is important to notice that, with the increment of the phosphate concentrations, there is only a slight increase in QS concentrations. Hence, it is clear that the QS concentrations have the most significant impact on the sigmoid shape.

Fig. 14 shows that, in region 1, the QS concentrations are comparatively low compared to region 2, as the bacterial cells are located significantly below the surface, minimizing nutrient accessibility. At $TS = 20$, the QS concentrations in this region are fairly close to that in region 2 at TS = 25. Hence, the sigmoid parameters are also close in these two periods. However, careful observation reveals that region 1 at $TS = 20$ has more noise in the 3D surface plot of the sigmoid series. This effect was demonstrated in the previous section, where the lower MI values were witnessed in region 1 compared to region 2, increasing the uncertainty of the computing process. This is more evident in region 0, where the sigmoid function plot is severely distorted, exhibiting a noisy output. Despite the noise, region 2 and region 1 contain a series of sigmoid curves creating a reliable solution space. Further, the influence of the phosphate concentration in the environment can be considered a fine-tuning factor, as each 3D sigmoid series plot shows slight shape variation with respect to the phosphate concentration.

Fig. 15 compares the sigmoid function variations with respect to the location and time slot, showing the diversity of the solution space. Fig. 15 *a* shows four different sigmoid function variations in region 2. The sigmoid function of this region at $TS = 15$ has the largest $L$ value of 0.97, creating the highest upper bound, whereas the lowest is at $TS = 30$. This trend of the upper bound is repeated in the other locations as well. Although the sigmoid behavior in terms of $L$ and $k$ values of region 1 is relatively close to region 2, the parameter $x_0$ varies from 0.70 to 0.11, which is different from the range of 0.55–0.07 in region 2. This, in turn, leads to a slight decrease in the steepness of the sigmoid functions in region 1 compared to region 2, which results in a sharper decision boundary. These sigmoid fluctuations further elucidate the computational diversity of the collective bacterial layers as perceptrons. In contrast, as we observed earlier, region 0 contains a set of noisy sigmoid curves that do not have a distinct decision boundary to produce a reliable output.

Therefore, the spatio-temporal variation drives the nutrient availability for cells in the biofilm, which in turn regulates cell-cell communication leading to computational diversity. The consequent variations of bacterial GRNN-based computing collectively form a diverse solution space of sigmoid function variations.

## CONCLUSIONS

In this study, we introduce a method to quantify the gene-to-gene interactions that converts the GRN into a GRNN to facilitate in-depth analysis of inherited computing properties at an individual cell as well as biofilm ecosystem levels. We specifically focused on *P. aeruginosa* as the model species and extracted a GRNN using GRN structural and transcriptomic dynamic data. Further, we utilized a graph neural network structure to model the cell-cell communication in a bacterial ecosystem, which heavily influences the computing properties. Using mutagenesis effects that result in GRNN with modified network structures, we prove the accuracy of the extracted weights by comparing the simulated and wet-lab experimental data. In addition, the graph neural network model with embedded GRNN as computing components further reveals the neural network properties of an individual cell's GRNN and spatio-temporal computing variations within a biofilm ecosystem. Another set of

analyses is conducted on the collective computing diversity of cell clusters in terms of sigmoid activation function, which have varying decision boundaries with respect to the location in the biofilm ecosystem as well as time. This proves the possibility of extracting sigmoid activation function solution space that is driven by the nutrient flow of the environment in combination with cell-cell communication. The varying shapes of the sigmoid activation function that is spatially and temporarily placed in the biofilm can lead us to a parallel computing process using a contained bacterial population. Further, we explore the reliability of GRNN computing through MI analysis, which reveals that cell-cell communication and nutrients flow heavily influence the input-to-output computing reliability. This elucidates that higher nutrient accessibility positively reflects on cell-cell communication leading to more reliable computing, whereas limited communication between cells increases higher uncertainty of information flows between gene expressions within the network. The findings from this study contribute to new viewpoints on bacterial decision making and also lay the groundwork for AI-based bio-computing using bacteria.

## AUTHOR CONTRIBUTIONS

S.S., S.B., and D.P.M. designed the theoretical framework of the study. The implementation of the analysis was done by S.S. X.L. provided the knowledge for the biological aspect of this study. All the authors wrote and reviewed the final manuscript.

## ACKNOWLEDGMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland under grant number 16/RC/3835.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

## REFERENCES

1. Alm, E., K. Huang, and A. Arkin. 2006. The Evolution of Two-Component Systems in Bacteria Reveals Different Strategies for Niche Adaptation. *PLoS Comput. Biol.* 2:e143. https://doi.org/10.1371/journal.pcbi.0020143.
2. Becerra, A. G., M. Gutiérrez, and R. Lahoz-Beltra. 2022. Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network. *Biosystems.* 213, 104608.
3. Blair, D. F. 1995. HOW BACTERIA SENSE AND SWIM. *Annu. Rev. Microbiol.* 49:489–522. https://doi.org/10.1146/annurev.mi.49.100195.002421.
4. Riethoven, J.-J. M. 2010. Regulatory regions in DNA: promoters, enhancers, silencers, and insulators. *Methods Mol. Biol.* 674:33–42.
5. Ishihama, A. 2010. Prokaryotic genome regulation: multifactor promoters, multitarget regulators and hierarchic networks. *FEMS Microbiol. Rev.* 34:628–645.
6. Alon, U. 2019. An introduction to systems biology: design principles of biological circuits. CRC press.
7. Land, M., L. Hauser, …, D. W. Ussery. 2015. Insights from 20 years of bacterial genome sequencing. *Funct. Integr. Genomics.* 15:141–161. https://doi.org/10.1007/s10142-015-0433-4.
8. Wang, C., S. Xu, and Z.-P. Liu. 2022. Evaluating Gene Regulatory Network Activity From Dynamic Expression Data by Regularized Constraint Programming. *IEEE J. Biomed. Health Inform.* 26:5738–5749.
9. Ravì, D., C. Wong, …, G.-Z. Yang. 2017. Deep learning for health informatics. *IEEE J. Biomed. Health Inform.* 21:4–21.
10. Lahoz-Beltra, R., J. Navarro, and P. C. Marijuán. 2014. Bacterial computing: a form of natural computing and its applications. *Front. Microbiol.* 5:101.
11. Dressler, F., and O. Akan. 2010. Bio-inspired networking: from theory to practice. *IEEE Commun. Mag.* 48:176–183.
12. Egmont-Petersen, M., D. de Ridder, and H. Handels. 2002. Image processing with neural networks—a review. *Pattern Recogn.* 35:2279–2301.
13. Sun, X., H. Khedr, and Y. Shoukry. 2019. Formal verification of neural network controlled autonomous systems. *In* Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, pp. 147–156.
14. Marković, D., A. Mizrahi, …, J. Grollier. 2020. Physics for neuromorphic computing. *Nat. Rev. Phys.* 2:499–510.
15. Furber, S. B., F. Galluppi, …, L. A. Plana. 2014. The SpiNNaker Project. *Proc. IEEE.* 102:652–665.
16. Benjamin, B. V., P. Gao, …, K. Boahen. 2014. Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proc. IEEE.* 102:699–716.
17. Davies, M., N. Srinivasa, …, H. Wang. 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro.* 38:82–99.
18. DeBole, M. V., B. Taba, …, D. S. Modha. 2019. TrueNorth: Accelerating From Zero to 64 Million Neurons in 10 Years. *Computer.* 52:20–29.
19. Liu, X., F. Wang, …, S. Ramakrishna. 2022. Bio-Inspired 3D Artificial Neuromorphic Circuits. *Adv. Funct. Mater.* 32, 2113050.
20. Mehonic, A., A. Sebastian, …, A. J. Kenyon. 2020. Memristors—From in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing. *Adv. Intell. Syst.* 2, 2000085.
21. Fuller, E. J., S. T. Keene, …, A. A. Talin. 2019. Parallel programming of an ionic floating-gate memory array for scalable neuromorphic computing. *Science.* 364:570–574.
22. Balasubramaniam, S., S. Somathilaka, …, M. Pierobon. 2023. Realizing Molecular Machine Learning Through Communications for Biological AI. IEEE Nanotechnology Magazine.
23. Smirnova, L., B. S. Caffo, …, T. Hartung. 2023. Organoid intelligence (OI): the new frontier in biocomputing and intelligence-in-a-dish. *Front. Sci.* 1
24. Vohradský, J. 2001. Neural network model of gene expression. *Faseb. J.* 15:846–854.
25. Weaver, D. C., C. T. Workman, and G. D. Stormo. 1999. Modeling regulatory networks with weight matrices. . Biocomputing'99. World Scientific, pp. 112–123.
26. Wu, Z., B. Ramsundar, …, V. Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* 9:513–530.

27. Ślęzak, J., and S. Burov. 2021. From diffusion in compartmentalized media to non-Gaussian random walks. *Sci. Rep.* 11:5101.
28. Silva, K. P., P. Chellamuthu, and J. Q. Boedicker. 2017. Signal destruction tunes the zone of activation in spatially distributed signaling networks. *Biophys. J.* 112:1037–1044.
29. Shmulevich, I., E. Dougherty, and W. Zhang. 2002. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE.* 90:1778–1792.
30. Wang, D., K.-K. Yan, …, M. B. Gerstein. 2015. Loregic: a method to characterize the cooperative logic of regulatory factors. *PLoS Comput. Biol.* 11, e1004132.
31. Adir, O., M. R. Albalak, …, A. Schroeder. 2022. Synthetic cells with self-activating optogenetic proteins communicate with natural cells. *Nat. Commun.* 13:2328.
32. Gargantilla Becerra, Á., M. Gutiérrez, and R. Lahoz-Beltra. 2021. A synthetic biology approach for the design of genetic algorithms with bacterial agents. *Int. J. Parallel, Emergent Distributed Syst.* 36:275–292.
33. Ortiz, Y., J. Carrión, …, M. Gutiérrez. 2021. A framework for implementing metaheuristic algorithms using intercellular communication. *Front. Bioeng. Biotechnol.* 9, 660148.
34. Berkovic, G., V. Krongauz, and V. Weiss. 2000. Spiropyrans and spirooxazines for memories and switches. *Chem. Rev.* 100:1741–1754.
35. Andrianantoandro, E., S. Basu, …, R. Weiss. 2006. Synthetic biology: new engineering rules for an emerging discipline. *Mol. Syst. Biol.* 2:2006.0028.
36. Rizik, L., L. Danial, …, R. Daniel. 2022. Synthetic neuromorphic computing in living cells. *Nat. Commun.* 13:5602.
37. Pandi, A., M. Koch, …, J.-L. Faulon. 2019. Metabolic perceptrons for neural computing in biological systems. *Nat. Commun.* 10:3880.
38. Li, X., L. Rizik, …, R. Daniel. 2021. Synthetic neural-like computing in microbial consortia for pattern recognition. *Nat. Commun.* 12:3139.
39. Crowther, M., A. Wipat, and Á. Goñi-Moreno. 2022. A network approach to genetic circuit designs. *ACS Synth. Biol.* 11:3058–3066.
40. Carbonell-Ballestero, M., E. Garcia-Ramallo, …, J. Macía. 2016. Dealing with the genetic load in bacterial synthetic biology circuits: convergences with the Ohm's law. *Nucleic Acids Res.* 44:496–507.
41. Grosso-Becerra, M. V., G. Croda-García, …, G. Soberón-Chávez. 2014. Regulation of Pseudomonas aeruginosa virulence factors by two novel RNA thermometers. *Proc. Natl. Acad. Sci. USA.* 111:15562–15567.
42. Ishihama, A. 2012. Prokaryotic genome regulation: a revolutionary paradigm. *Proc. Jpn. Acad. Ser. B Phys. Biol. Sci.* 88:485–508.
43. Spitz, F., and E. E. M. Furlong. 2012. Transcription factors: from enhancer binding to developmental control. *Nat. Rev. Genet.* 13:613–626.
44. Davis, M. C., C. A. Kesthely, …, S. R. MacLellan. 2017. The essential activities of the bacterial sigma factor. *Can. J. Microbiol.* 63:89–99.
45. Unluturk, B. D., S. Balasubramaniam, and I. F. Akyildiz. 2016. The impact of social behavior on the attenuation and delay of bacterial nanonetworks. *IEEE Trans. NanoBioscience.* 15:959–969.
46. de Kievit, T. R. 2009. Quorum sensing in Pseudomonas aeruginosa biofilms. *Environ. Microbiol.* 11:279–288.
47. Rumbaugh, K. P., J. A. Griswold, and A. N. Hamood. 2000. The role of quorum sensing in the in vivo virulence of Pseudomonas aeruginosa. *Microb. Infect.* 2:1721–1731.
48. Lee, J., and L. Zhang. 2015. The hierarchy quorum sensing network in Pseudomonas aeruginosa. *Protein Cell.* 6:26–41.
49. Yan, S., and G. Wu. 2019. Can biofilm be reversed through quorum sensing in Pseudomonas aeruginosa? *Front. Microbiol.* 10:1582.
50. Abisado, R. G., S. Benomar, …, J. R. Chandler. 2018. Bacterial quorum sensing and microbial community interactions. *mBio.* 9:e02331-17.
51. Penesyan, A., I. T. Paulsen, …, M. R. Gillings. 2021. Three faces of biofilms: a microbial lifestyle, a nascent multicellular organism, and an incubator for diversity. *NPJ Biofilms Microbiomes.* 7:80.
52. Seshasayee, A. S. N., K. Sivaraman, and N. M. Luscombe. 2011. An overview of prokaryotic transcription factors: a summary of function and occurrence in bacterial genomes. *Subcell. Biochem.* 52:7–23.
53. Galán-Vásquez, E., B. C. Luna-Olivera, …, A. Martínez-Antonio. 2020. RegulomePA:a database of transcriptional regulatory interactions in Pseudomonas aeruginosa PAO1. *Database.* https://doi.org/10.1093/database/baaa106.
54. Kanehisa, M., and S. Goto. 2000. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 28:27–30.
55. Kanehisa, M. 2019. Toward understanding the origin and evolution of cellular organisms. *Protein Sci.* 28:1947–1951.
56. Kanehisa, M., M. Furumichi, …, M. Ishiguro-Watanabe. 2023. KEGG for taxonomy-based analysis of pathways and genomes. *Nucleic Acids Res.* 51:D587–D592.
57. Keseler, I. M., J. Collado-Vides, …, P. D. Karp. 2011. EcoCyc: a comprehensive database of Escherichia coli biology. *Nucleic Acids Res.* 39:D583–D590.
58. Barrett, T., S. E. Wilhite, …, A. Soboleva. 2013. NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Res.* 41:D991–D995. https://doi.org/10.1093/nar/gks1193.
59. Ioannidis, V. N., A. G. Marques, and G. B. Giannakis. 2019. Graph neural networks for predicting protein functions. *In* IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) IEEE, pp. 221–225.
60. Zhang, Z., L. Chen, …, X. Li. 2022. Graph neural network approaches for drug-target interactions. *Curr. Opin. Struct. Biol.* 73, 102327.
61. Zhang, X.-M., L. Liang, …, M.-J. Tang. 2021. Graph neural networks and their current applications in bioinformatics. *Front. Genet.* 12, 690049.
62. Castorina, G., L. Galluccio, and S. Palazzo. 2016. On Modeling Information Spreading in Bacterial Nano-Networks Based on Plasmid Conjugation. *IEEE Trans. NanoBioscience.* 15:567–575.
63. Noel, A., K. Cheung, and R. Schober. 2014. Improving diffusion-based molecular communication with unanchored enzymes. *In* Bio-Inspired Models of Network, Information, and Computing Systems: 7th International ICST Conference, BIONETICS 2012, Lugano, Switzerland, December 10–11, 2012, Revised Selected Papers 7 Springer, pp. 184–198.
64. Somathilaka, S. S., D. P. Martins, …, S. Balasubramaniam. 2022. A Graph-Based Molecular Communications Model Analysis of the Human Gut Bacteriome. *IEEE J. Biomed. Health Inform.* 26:3567–3577.
65. Sultan, M., R. Arya, and K. K. Kim. 2021. Roles of two-component systems in Pseudomonas aeruginosa virulence. *Int. J. Mol. Sci.* 22, 12152.
66. Lamb, J. R., H. Patel, …, B. H. Iglewski. 2003. FunctionalDomains of the RhlR Transcriptional Regulator of Pseudomonas aeruginosa. *J. Bacteriol.* 185:7129–7139.
67. Pearson, J. P., E. C. Pesci, and B. H. Iglewski. 1997. Roles of Pseudomonas aeruginosa las and rhl quorum-sensing systems in control of elastase and rhamnolipid biosynthesis genes. *J. Bacteriol.* 179:5756–5767.

68. Wade, D. S., M. W. Calfee, …, E. C. Pesci. 2005. Regulation of Pseudomonas quinolone signal synthesis in Pseudomonas aeruginosa. *J. Bacteriol.* 187:4372–4380.

69. Nadal Jimenez, P., G. Koch, …, W. J. Quax. 2012. The multiple signaling systems regulating virulence in Pseudomonas aeruginosa. *Microbiol. Mol. Biol. Rev.* 76:46–65.

70. Stewart, P. S. 2003. Diffusion in Biofilms. *J. Bacteriol.* 185:1485–1491.

71. Canela-Xandri, O., F. Sagués, and J. Buceta. 2010. Interplay between intrinsic noise and the stochasticity of the cell cycle in bacterial colonies. *Biophys. J.* 98:2459–2468.

72. Heinlein, B., L. Brand, …, S. Lotter. 2023. Stochastic Chemical Reaction Networks for MAP Detection in Cellular Receivers. Preprint at arXiv. https://doi.org/10.48550/arXiv:2305.06006.

73. Meng, X., S. D. Ahator, and L.-H. Zhang. 2020. Molecular mechanisms of phosphate stress activation of Pseudomonas aeruginosa quorum sensing systems. *mSphere.* 5:e00119-20.

74. Granik, N., L. E. Weiss, …, Y. Shechtman. 2019. Single-particle diffusion characterization by deep learning. *Biophys. J.* 117:185–192.

75. Li, Y., P. Xiao, …, Y. Hao. 2020. Mechanisms and control measures of mature biofilm resistance to antimicrobial agents in the clinical context. *ACS Omega.* 5:22684–22690.

76. Liebert, W., and H. Schuster. 1989. Proper choice of the time delay for the analysis of chaotic time series. *Phys. Lett.* 142:107–111.

77. Dubey, S. R., S. K. Singh, and B. B. Chaudhuri. 2022. Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing.

78. Qin, Y., X. Wang, and J. Zou. 2019. The Optimized Deep Belief Networks With Improved Logistic Sigmoid Units and Their Application in Fault Diagnosis for Planetary Gearboxes of Wind Turbines. *IEEE Trans. Ind. Electron.* 66:3814–3824.

79. Alkhouly, A. A., A. Mohammed, and H. A. Hefny. 2021. Improving the Performance of Deep Neural Networks Using Two Proposed Activation Functions. *IEEE Access.* 9:82249–82271.

80. Farzad, A., H. Mashayekhi, and H. Hassanpour. 2019. A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Comput. Appl.* 31:2507–2521.

# Chapter 8

# Symposium: Wet TinyML: Chemical Neural Network Using Gene Regulation and Cell Plasticity

| | |
|---|---|
| **Symposium Title:** | tinyML Research Symposium'24 |
| **Article Type:** | Regular Paper |
| **Complete Author List:** | Samitha S. Somathilaka, Adrian Ratwatte, Sasitharan Balasubramaniam, Mehmet Can Vuran, Witawas Srisa-an, and Pietro Liò |
| **Status:** | Published. April 2024. doi.org/10.48550/arXiv.2403.08549 |

# Wet TinyML: Chemical Neural Network Using Gene Regulation and Cell Plasticity

Samitha Somathilaka
samitha.somathilaka@waltoninstitute.ie
Walton Institute, SETU, Ireland
University of Nebraska-Lincoln, USA

Adrian Ratwatte
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
aratwatte2@huskers.unl.edu

Sasitharan Balasubramaniam
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
sasi@unl.edu

Mehmet Can Vuran
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
mcv@unl.edu

Witawas Srisa-an
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
witty@cse.unl.edu

Pietro Liò
University of Cambridge
Cambridge, UK
Pietro.Lio@cl.cam.ac.uk



## ABSTRACT

In our earlier work, we introduced the concept of Gene Regulatory Neural Network (GRNN), which utilizes natural neural network-like structures inherent in biological cells to perform computing tasks using chemical inputs. We define this form of chemical-based neural network as Wet TinyML. The GRNN structures are based on the gene regulatory network and have weights associated with each link based on the estimated interactions between the genes. The GRNNs can be used for conventional computing by employing an application-based search process similar to the Network Architecture Search. This study advances this concept by incorporating cell plasticity, to further exploit natural cell's adaptability, in order to diversify the GRNN search that can match larger spectrum as well as dynamic computing tasks. As an example application, we show that through the directed cell plasticity, we can extract the mathematical regression evolution enabling it to match to dynamic system applications. We also conduct energy analysis by comparing the chemical energy of the GRNN to its silicon counterpart, where this analysis includes both artificial neural network algorithms executed on von Neumann architecture as well as neuromorphic processors. The concept of Wet TinyML can pave the way for the new emergence of chemical-based, energy-efficient and miniature Biological AI.

## KEYWORDS

Biological AI, Cell Plasticity, Biocomputing, Neuromorphic.

**ACM Reference Format:**
Samitha Somathilaka, Adrian Ratwatte, Sasitharan Balasubramaniam, Mehmet Can Vuran, Witawas Srisa-an, and Pietro Liò. 2024. Wet TinyML: Chemical Neural Network Using Gene Regulation and Cell Plasticity. In *Proceedings of tinyML Research Symposium (tinyML Research Symposium'24).* ACM, San Francisco, USA, 7 pages.


*tinyML Research Symposium'24, April 2024, San Francisco, CA*


## 1 INTRODUCTION

TinyML aims to execute machine learning algorithms with minimum size to conserve energy consumption and deployment into devices with limited computational capacity [25]. Achievements in TinyML include implementation of code sizes down to 1 KB [11] and energy consumption of the algorithm as low as 25 mW [1], enabling deployment in miniature devices such as in-body implantables[4]. However, embedding TinyML codes in environments that cannot accomodate silicon-based devices poses a challenge, necessitating a new design paradigm that adheres to TinyML's goals of low-energy consumption and compact coding. This paper tackles this issue by shifting the focus from silicon technologies to exploring natural processes that mimics Artificial Neural Network (ANN) functions in biological cells, where we can use this to perform conventional computing tasks.

Our prior research delved into the computational aspect of biological cells, demonstrating that Gene Regulatory Networks (GRNs) serve as fundamental computational entities within cells, aiding in decision-making processes in response to environmental cues [19]. This entails the reception of extracellular molecules, their processing via GRNs, and the subsequent production of output molecules such as proteins. Nonetheless, our examination was confined to the static behaviour of cells, which does not reflect their natural adaptability.

However, further investigation on natural cell adaptability along with their learning capabilities, have prompted the question: *"how can non-neuronal organisms that display traits of intelligence through plasticity be used to develop non-silicon-based neural networks?"*. In turn, this study extends our previous work towards a new concept of *Wet TinyML* that is constructed from the gene regulation process and explore the impact of cell plasticity on GRN based computing. We refer to this component of Wet TinyML as **Gene Regulatory Neural Networks** (GRNNs), where we transform a GRN with established weights based on the relative gene expression. In the GRNN, genes can receive molecules known as Transcription Factors (TF) from multiple other neighboring genes [9], akin to how perceptrons receive external inputs. The influence of one gene on another acts as the 'weight' in this analogy. The cumulative impact of molecular signals from neighboring genes and their intensities collectively regulate the expression of a target gene, analogous to the weighted summation in ANN processing. Furthermore, a gene is expressed

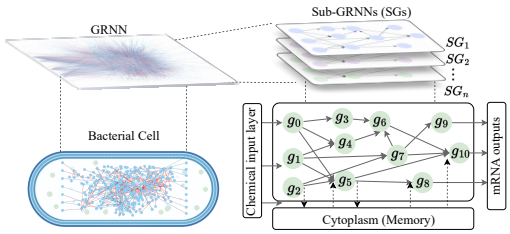



**Figure 1: Wet TinyML based on GRNNs extracted and searched in a bacterium to perform computing. Input chemicals trigger selective activation of relevant GRNN-subnetworks, rendering the GRNN a composite of many sub-networks. Gene products, diffusing into the cytoplasm forming cellular memory system that contributes to cell plasticity.**

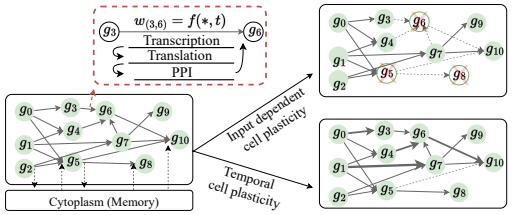



**Figure 2: In GRNN framework, gene-perceptron operate similarly to ANN perceptrons, processing inputs with weights influenced by multi-omic layer interactions (*), and time(t). Bacterial cells exhibit input-dependent plasticity by unique gene expression pathways varying with different inputs. Additionally, cells demonstrate temporal plasticity by altering GRNN subnetwork interaction weights over time.**

only if the cumulative influence of the transcription factors exceeds a certain threshold [20], mirroring the role of activation functions in ANNs. In a number of ways, the GRNN can be associated to a chemical-hardware version of a neuromorphic computing system.

Research indicates that gene expression is selectively influenced by input chemicals, suggesting that the GRNN can be viewed as a large collection of pre-trained GRNN sub-networks and each is triggered depending on its chemical input such as nutrient molecules (Fig. 1). Each GRNN subnetwork comprises an input layer, intermediate nodes that are akin to the hidden layer, and an output layer. The products of the transcription and translation processes resulting from natural computing are diffused into the cytoplasm as depicted in Fig. 1 and interact with other biological components such as ribosomes [17]. Further, as Fig. 2 elucidates, the accumulated molecules in the cytoplasm can act as a memory module that induces feedforward and feedback signals in optimizing GRNN subnetwork switching and adjusting weights over time.

The GRNN, as a pre-trained network, allows bypassing the conventional ANN training phase by directly selecting an appropriate GRNN sub-network for specific tasks[18], similar to the way Network Architecture Search (NAS)[2] using supervised data. Subsequently, this research examines expanding this GRNN sub-network search space by accounting cellular plasticity and assesses energy consumption compared to existing neuromorphic systems. Finally, the study utilizes GRNN-subnetworks to derive various mathematical regression models.

This paper is organized as follows: Section 2 introduces 'Wet tinyML', explaining the inherent computing power of natural biological cells driven by GRNNs. Section 3 compares GRNN's energy efficiency with traditional von Neumann and neuromorphic systems, focusing on algorithmic and structural complexities, and discusses how natural cell plasticity can enhance computing diversity. The applications of GRNN are detailed in Section 4, and Section 5 concludes the study.

## 2 BACKGROUND OF GRNN AND CELL PLASTICITY

This section introduces the elements of GRNN for Wet TinyML, building upon our earlier research [18, 19].

### 2.1 Gene-perceptrons and Weights

As discussed in Section 1, several characteristics of genes in their complex regulatory process exhibit similarity to an artificial neuron. In ANNs, a perceptron processes multiple inputs by applying weights and summing them, followed by an activation function. This process finds parallels to genetic circuit operations, where a gene receives TF molecules that leads to a combinatorial regulation of a gene's expression.

In ANNs, the output from the weighted summation is modulated by activation functions, such as *sigmoid*, *tanh*, or *Rectified Linear Unit* (*ReLU*), which introduces non-linearity to the computing. This concept is mirrored in gene regulation, where a gene can be in an 'on' or 'off' state, depending on the regulatory impact of the TFs. The combined influence of TFs acts like the weighted summation, translating into binary gene expression states ('on' or 'off'), very similar to a sigmoid function's output.

However, when considering the time dynamics of gene expressions, we showed in our previous work [18] the ReLU activation function is more compatible than the sigmoid function, as it accommodates the linear relationship between the TF influence and gene expressions over time. Further observation reveals that prokaryotic genes exhibit 'ground states', where RNA polymerase can bind to promoters in the absence of activators or repressors, suggesting an addition of a bias term to the weighted summation of each gene. Given these gene properties that behave as an ANN perceptron, we introduced the term **gene-perceptron** in the GRNN [18, 19].

### 2.2 GRN-to-GRNN Conversion

The GRNN is considered a pre-trained graph-strucured neural network that is inferred by assigning weights to gene-gene interactions of a GRN, based on relative gene expression levels, which we detail in this section. The weight extraction for each gene-perceptron is conducted iteratively using network modules comprising the target gene-perceptron and its associated source gene perceptrons, structurally akin to single-layer perceptrons.

The weight extraction for each gene-perceptron module involves a process similar to training a single-layer perceptron and involves the use of transcriptomic data as shown in Fig. 3a. In this figure, $y_i^{(t)}$ is the expression level of gene $g_i$ at timestep $t$, $w_{i,p}^{(t,T+1)}$ is the

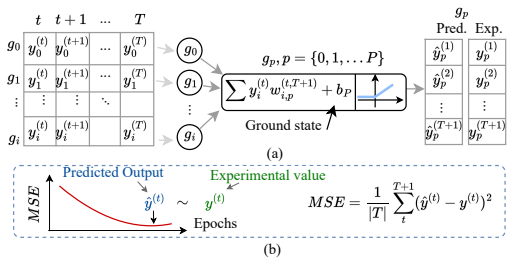



**Figure 3: Illustration of the weight extraction process where a) elucidates the utilization of transcriptomic data considering gene $g_P$ as a target single-layer gene-preceptron and b) depicts the training process of minimizing the MSE between predicted and experiment expression levels.**

weight of the interaction between gene $g_i$ and $g_p$ in the time interval $t$ to $T+1$, and $b_P$ is the bias (the ground state) of the gene $g_p$. Initially, random weights are introduced with the experimental transcription data at the levels of source gene-perceptrons. The target gene-perceptron's predicted transcription level is then calculated by passing the weighted summation of experimental source gene-percepton levels and weights through a ReLU function. The weights are then refined based on adjusting the differences between the predicted and experimental transcription levels of the target gene-perceptron by using learning rate as 0.001 with $10^5$ epochs. This iterative adjustment continues across all experimental transcriptomic data until the error is minimized as shown in Fig. 3b. Repeating this procedure estimates weights for all gene-perceptron modules, which are then applied to the GRN, transforming into a GRNN. More details on this weight extraction can be found in [18].

## 2.3 Input-dependent and Temporal Plasticity

Bacteria are renowned for their remarkable adaptability in various environments, a trait crucial for their survival [24]. In observing bacterial cells as natural computational entities, we analyze this cellular plasticity through the lens of ANNs, identifying two primary features: input-dependent plasticity and temporal plasticity.

*Input-dependent plasticity* is driven by the selective responsiveness of genes to specific input chemicals [24]. Genes at the periphery of the GRNN, akin to the input layer in an ANN, are particularly sensitive to certain chemical effectors. In nature, this sensitivity leads to the expression of a specific subset of genes in response to the abundance of these chemical molecules. The GRNN, thus, selectively channels information flow, activating only the relevant expression pathways, while keeping other genes largely idle. This results in the utilization of specific GRNN subnetworks based on the chemical input, as illustrated in Fig. 2. This in turn enhances energy efficiency during the gene regulatory process of the cells.

On the other hand, *temporal plasticity* involves alterations in the influence of one gene on another over time [16] to achieve the optimal behavior for an given environment, which resembles weight plasticity within the same input conditions.

In previous works [18, 19], we focused on a static GRNN to search application-specific GRNN subnetworks. However, this study focuses on harnessing these natural plasticities to expand the diversity of searching for the optimal applications specifc GRNN subnetwork.

# 3 ENERGY AND COMPUTING DYNAMICS OF GRNN

## 3.1 Energy vs Computing Complexity

The scale and energy efficiency of ANNs are critical factors in the miniaturization of algorithms and neuromorphic hardware designs. Biological systems, such as cells and neurons, which operate at the micrometer scale, demonstrate exceptional energy efficiency in natural computing processes. For example, the human brain, despite its immense computational capacity, has a remarkably low energy consumption of only 20W [7]. Similarly, the expression of a single gene-perceptron, a complex task in itself, consumes a mere 0.01 fW [12]. This observation leads to the consideration of the energy profile of GRNNs as a potential full-scale computing platform.

*3.1.1 GRNN Structural and Algorithmic Complexity.* Taking into account that the GRNN is a randomly structured network with power-law degree distributions, we will analyze the estimated algorithmic and structural complexities. The algorithmic complexity reflects the information diffusion, failure propagation, and resilience through the network [13]. It is defined by the Kolmogorov complexity, which is approximated using the Coding Theorem Method [22]. Understanding the algorithmic complexity, which reflects on the complex nature of the network structure, can provide avenues for interpretability of the model [5]. The structural complexity, on the other hand, is determined by the betweenness centrality and relative degree of gene-perceptrons in the network [8]. Although our previous work analyzed the structural and algorithmic complexity in GRNNs [18], in this paper we use these two measures to determine the relationship to the energy consumption of the GRNN. Further information regarding the use of the structural and algorithmic complexiy in GRNNs can be found in [18].

*3.1.2 GRNN Energy Analysis.* We will now determine how the algorithm and structural complexities of the GRNN and comparison to ANN play a role on the energy consumption. The total energy consumption for the $i^{th}$ GRNN, denoted as $P_{total}(i)$, is computed by summing the energy used in the transcription and translation processes, given by $P_{ex}(i) + P_{tra}(i)$. Here, $P_{ex}(i)$ represents the peak power for gene expression in the $i^{th}$ GRNN, and $P_{tra}(i)$ is the power required for its translation process. The transcription power is calculated as $P_{ex}(i) = |GRNN_i| \cdot \hat{p}_{ex}$, where $\hat{p}_{ex}$ is the per gene-perceptron transcription power and $|GRNN_i|$ is the number of gene-perceptrons in the $i^{th}$ GRNN. As mentioned earlier, $\hat{p}_{ex} = 0.01$ fW for a medium-sized prokaryotic cell such as *Escherichia coli* [12]. However, studies show that approximately 75% of a cell's energy dissipation is attributed to translation processes, while a mere 2% is utilized for transcription [10]. Subsequently, using this 2:75 power ratio between $P_{tra}$ and $P_{ex}$, the total power $P_{total}(i)$ is calculated.

In parallel, the energy consumption of silicon-based computing units is calculated as $P_{total} = N \cdot \hat{p}$, where $N$ is the number of neurons in the system and $p$ is the unit power. According to literature, the unit power consumption of a neuron for different processors, $p$, are listed as follows: Spikey at $1.49\text{x}10^{-06}$, R2600X at $9.62\text{x}10^{-04}$, Intel mobile at $3.37\text{x}10^{-04}$, and RTX2070 at $3.18\text{x}10^{-05}$ [15].

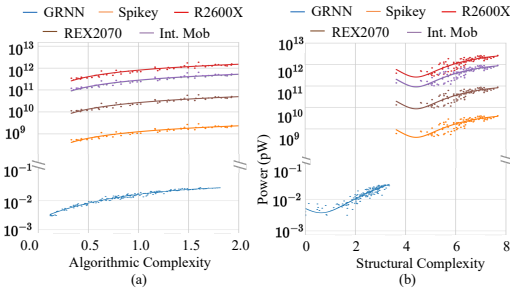



**Figure 4: Power comparison between GRNN vs von Neumann and neuromorphic computing systems with respect to a) algorithmic complexity and b) structural complexity.**

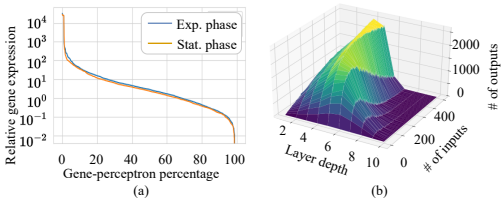



**Figure 5: Illustrations of a) the sparsity of gene expression and b) number of output node variations given the number of input nodes and the depth of the GRNN subnetwork.**

We emphasize that the energy consumption across various computing platform focuses solely on the energy used for computing, excluding housekeeping energy requirements. We first compare the energy consumption of GRNN with four other processors mentioned above with respect to the algorithmic complexity of 200 different model sizes. This evaluation involved varying the number of nodes within each model across four Von Neumann and neuromorphic platforms. The results of this analysis are presented in Fig. 4. Notably, the GRNN's maximum power consumption does not surpass 0.05 pW, even at the highest level of algorithmic complexity, as shown in Fig. 4a. In contrast, the other platforms register energy usage ranging from $10^9$ pW to $10^{12}$ pW for models with an equivalent number of neurons. The less sparse connectivity and low diameter in the GRNN typically contribute towards minimized algorithmic complexity. While the heterogeneity in the weights can increase the algorithmic complexity, this increase still results in low energy consumption due to the chemical energy used by the gene-perceptrons [23].

Furthermore, a similar comparison is conducted to examine energy dissipation with respect to the structural complexity, depicted in Fig. 4b. The results uncover patterns of energy consumption similar to those observed in the energy dissipation with respect to the algorithmic complexity. It is important to highlight that the small-world network structure of the GRNN exhibits notably lower structural complexity compared to others. Small-world networks, known for their high clustering and short path lengths, have a more orderly structure, which lowers their structural entropy [6, 14] and this can be observed in Fig. 4b.

The additional housekeeping energy of GRNN computing depends on a range of factors including administration of chemical inputs and extraction method of outputs, which will be explored in our future research.

## 3.2 Non-Plasticity Search Space of GRNNs

This section first assesses the sparsity of GRNN by analyzing gene expression levels during two key growth stages: exponential growth and the stationary phase, using data sourced from [21]. As illustrated in Fig. 5a, on average, a cell utilizes only about 10% of its genes at the considered two growth phases of bacteria, which are the idling and rapid growing phases, respectively. This observation indicates that the cell possesses a broader range of options for transitioning between GRNN subnetworks to adapt to diverse environmental conditions. The inherent sparsity in the GRNNs further contributes significantly to massively parallelized computing. This explains how bacteria in nature process signals in a parallel manner, regulating multiple metabolic pathways simultaneously. Such natural parallels further underscore the capacity of GRNNs for parallel computing.

We investigate the diversity of GRNN subnetworks by focusing on the size of input, intermediate hidden layer, and output gene-perceptrons. By selecting 100 random input gene-perceptrons and tracking their connections through the network, we analyzed the structure up to 10 layer depths. This process is repeated 100 times with varying input sets aiming to average the count of gene-perceptrons per layer. The experiment is conducted with input sets increasing by 100, up to 500 gene-perceptrons, to ensure a comprehensive assessment.

It is evident from Fig. 5b that with a relatively small sized network with the input layer comprising 100 nodes, which is capable of processing inputs for 100 features, the maximum output node count reaches approximately 500 when the network depth is close to 6. This GRNN subnetwork diversity allows the selection of a certain number of output nodes, up to 500, tailored for a given application. For instance, in an application requiring 10 output nodes, the GRNN offers approximately $8.9\times10^{26}$ combinations of output node choices. It is important to note that this abundance of options results from the initial choice of 100 nodes in the input layer. Considering only 1000 suitable candidates for the input layer, the permutations for 100 input nodes increases to $5.9 \times 10^{297}$. This astronomically large number of combinations underscores the GRNN's possibility of facilitating the identification of a suitable GRNN subnetwork for specific applications exhibiting generalizability.

Furthermore, Fig. 5b reveals that increasing the number of nodes in the input layer to 500 results in the expansion of the output layer to approximately 2,500 nodes. This provides $9.3\times10^{33}$ options for applications requiring 10 outputs, when the network depth is around six. This demonstrates the significant impact of input layer size on the diversity and adaptability for the network's output.

## 3.3 Search Space Expansion with Cell Plasticity

In this section, we examine input-dependent and temporal cell plasticity using the weight extraction algorithm from our previous study [19], and data sourced from the GEO database [3] to evaluate their impact on gene expression within the GRNN.

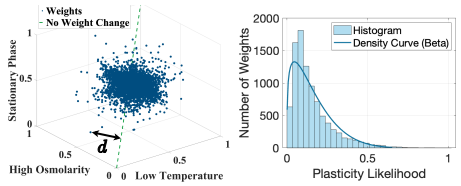



**Figure 6: Environmental Conditions and Cellular Plasticity: (a) Weight changes under conditions of low temperature, stationary phase, and high osmolarity, with $d$ indicating the distance from the no weight change line. (b) Plasticity likelihood related to weight adjustments across various inputs.**

*3.3.1 Input-dependent Cell Plasticity.* We study input-dependent cell plasticity using transcriptomic time series data from three environmental settings: low temperature, high osmolarity, and stationary phase. Weight changes across inputs are calculated by determining the geometric shortest distance between extracted weights for each condition and a line connecting (0,0,0) to (1,1,1). This line indicates no changes in the gene expression for the three conditions. Fig. 6a illustrates that while some weights align with the no change line, the majority exhibit varying degrees of change. Fig. 6b depicts the probability of weight plasticity across all three conditions, showing a left-skewed Beta density curve indicating that most weights have probabilities less than 0.5 for undergoing changes. Furthermore, our analysis in Table 1 explores how individual condition changes influence weight alterations across all conditions, revealing that approximately $2-5\%$ of the total weights have changed between distinct input conditions. This pattern emerges from selective gene activation by environmental factors, rather than affecting all genes universally. This analysis showcases the GRNN's capacity to adapt weights selectively in response to new input conditions, without affecting all weights uniformly.

*3.3.2 Temporal Cell Plasticity.* In this section, we analyze the temporal weights change within the GRNN. To analyze this, we utilize the transcriptomic data (GSE65244), which encompasses gene expression levels recorded at 10-minute intervals within the range of 0 to 420 minutes. We segment this dataset with respect to time into equal-sized partitions, each containing 30 expression levels collected at 30 different time points. The weights extracted for each of these time periods are represented as $W_0$, $W_{10}$, ..., $W_{130}$, as depicted in Fig. 7(a). We compute the correlation coefficient between the weights extracted for the initial time period (0-290 minutes), serving as the reference ($W_0$), and the weights from each subsequent time period. Fig. 7(b) illustrates the deviation of this computed correlation from the ideal positive correlation of 1. The deviation gradually rises to 0.1 between the time period 10-300 and 60-350 minutes at a gradual pace, before increasing rapidly. This suggests that applying identical input conditions for an extended duration prompts the GRNN to update its weights, contributing to the cell's survival as part of its plasticity process. As a result of this deviation from the positive correlation, we attain GRNNs with different weights over time, thereby expanding the search space for the GRNN sub-networks to suit an application. Since weight changes occur primarily in specific parts of the global GRNN, we can extract sub-networks featuring both dynamic and static weights. Tailoring input conditions over specific time periods allows us to derive the desired set of weights suited for the application.

**Table 1: Frequency analysis of altered weights across different experimental conditions.**

| Exp. replicate | # of Weights Showing Higher Probability of Plasticity | Ratio (%) |
|---|---|---|
| All Conditions | 466 | 8.08 |
| Low temperature | 372 | 5.06 |
| High Osmolarity | 130 | 2.42 |
| Stationary phase | 241 | 4.63 |

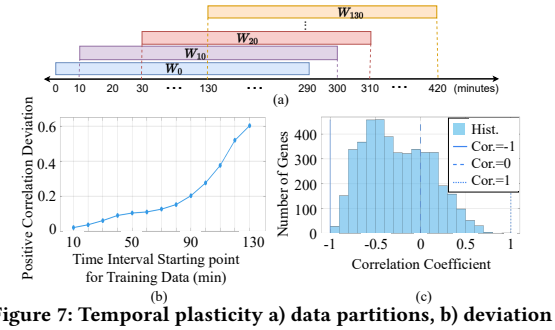


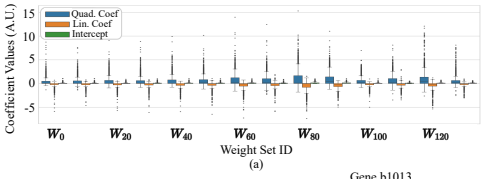

**Figure 7: Temporal plasticity a) data partitions, b) deviation from positive correlation and c) correlation of gene expression between 0-290 and 130-420 minutes time intervals.**

*3.3.3 Correlation Analysis of Gene Expression Temporal Dynamics in the GRNN.* The analysis of temporal gene expression dynamics aims to assess the influence of the temporal weight changes, previously addressed, on the gene expression within the GRNN. We compute correlation coefficients for gene expression levels between the time intervals of 0-290 and 130-420 minutes for each gene. Fig. 7c presents this results and shows that roughly 80% of genes within the GRNN display negative correlation, with correlation coefficients predominantly distributed between 0 and -1, while around 20% of the total genes exhibit positive correlation. This correlates with the results in Fig. 7c, where majority of weights undergo significant changes over time and this results in variations in the expression of most genes within the GRNN.

## 4 GRNN APPLICATION FOR REGRESSION

Unlike traditional ANNs, bacterial GRNNs was introduced with a specialized algorithm to search relevant subnetwork omitting the conventional ANN training, identifying input/output genes and the optimal time window based on supervised application data. This section elucidate a use case of regression to show the contribution of cell plasticity in expanding the search space theoretically. Although, the weight extraction method discussed in Section 2.2 and [18] can be used for any temporal transcriptomic and GRN data, this study uses *E. coli* as the model species.

This analysis utilizes seven distinct time-based weight configurations ($\mathbf{W}_i$, where $i = \{0, 10, 20, \ldots, 130\}$) as shown in Fig. 7a to assess the diversity in regression functions. To sharpen our focus, we select a single gene-perceptron as the input and focus only on quadratic regressions. Gene *b3067* is chosen as the input layer gene-perceptron for its impact on 1702 gene-perceptrons to get a larger solution space. This gene-perceptron is stimulated with five




| Weightset ID | Regression |
|---|---|
| $W_{10}$ | $0.04x + 0.02$ |
| $W_{40}$ | $-0.13x^2 + 0.03x + 0.02$ |
| $W_{70}$ | $1.10x^2 - 0.33x + 0.08$ |
| $W_{100}$ | $1.17x^2 - 0.23x + 0.05$ |
| $W_{130}$ | $1.55x^2 - 0.64x + 0.17$ |

(b)

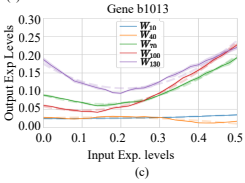



**Figure 8: The regression analysis, using *b3067* as the sole input gene-perceptron, includes: a) analysis of quadratic, linear coefficients, and intercepts across weight configurations outlining the solution space; b) the regression coefficients of *b1013*; and c) the associated regression curves.**

input concentrations (0.1 to 0.5). Based on ten iterations per setup to capture the stochastic behaviour, the output gene-perceptrons' expression levels are averaged. Expression levels are recorded at each timestep in different weight setups $W_i$ and using curve fitting for quadratic functions, quadratic and linear coefficients, and intercepts of gene-perceptrons are determined.

Fig. 8 illustrates the regression diversity of the GRNN with respect to the temporal cell plasticity. For each weight configuration $w_i$, Fig. 8a displays the variations in quadratic and linear coefficients, as well as intercepts. Each box plot in the figure represents the coefficients for 2,875 gene-perceptrons. Notably, there is a substantial variation in quadratic coefficients across all weight configurations, resulting in a range of regressions with varying curvatures. In turn, each box plot emphasizes the diversity in the solution space for a given application. Moreover, the distribution of these quadratic coefficients highlights the potential for deriving linear regressions in cases where the quadratic coefficient equals zero. However, the linear coefficients of curves linked to the chosen input gene-perceptrons (*b3067*) tend to exhibit predominantly negative values. Conversely, the intercepts are confined to a narrower range, in particular between 0 and 2.

Fig.8a and Fig.8b illustrate five regression curves and their plots for a single output gene-perceptron, *b1013*, at various timesteps. Fig.8a presents five quadratic coefficients for the gene-perceptron *b1013* under different weight configurations: 0 for $W_{10}$, -0.13 for $W_{40}$, 1.10 for $W_{70}$, 1.17 for $W_{100}$, and 1.17 for $W_{130}$. Notably, under the $W_{10}$ configuration, gene-perceptron *b1013* exhibits a linear function, while the subsequent configurations result in regression lines with increasing curvatures. These results show how temporal plasticity can expand the solution space. Our previous study [18], presented regressions for output gene-perceptrons based on a static weight configuration, allowing only a single regression function per output gene-perceptron. However, with the introduction of temporal plasticity weights, numerous regression curves can be derived for output gene-perceptrons.

It is important to note that the evolution of the regression curve influenced by weight plasticity in GRNN computing is gradual. This means that systems requiring static weights can only operate in a certain period to guarantee stable computing results. Additionally, an analysis of read count change showed that cells can achieve a maximum value of -2,649.76 as shown in Fig. 9a, ensuring significantly fast computing for a biological entity. We further conduct a principal component analysis of expression levels from temporal experiment replicates as shown in Fig. 9b, where each cluster of replicates, consistent across different conditions, confirms this reliability. Additionally, the concept of temporal plasticity offers a promising approach for addressing dynamic systems, a focus for our future research.

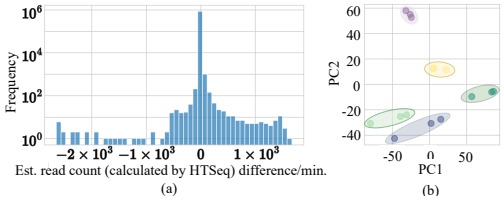



**Figure 9: Illustrations of a) estimated per-minute gene expression differences, and b) PCA clustering of experimental replicates, showing that similar environmental setups have close expression patterns, reinforcing reliable computing.**

## 5 CONCLUSION

Consistent with the vision of TinyML to establish miniature machine learning algorithms that can fit into low-powered devices, we extend our previously introduced GRNN concept towards a new paradigm with chemical-based ML algorithms found in biological cells. This new paradigm called Wet TinyML will transform a gene regulatory process into a GRNN that can compute similarly to a conventional ANN. Using the concept of bacterial cell plasticity, we show that the weights of the GRNN can be modified, opening new opportunities to map to diverse applications. Simultaneously, we estimate the energy consumption of GRNN subnetworks and find that they use less energy compared to traditional Von Neumann and Neuromorphic platforms. Our future research will evaluate the impact of noise in the GRN, cell reusability based on plasticity, computing speed analyzed from wet-lab experiments. The wet lab experiments will explore input genes that can easily be stimulated and engineering reporter genes in the output layer to determine the correct GRNN computing in the subnetwork. The concept of Wet TinyML can expand the paradigm of miniature machine learning algorithms based on using its natural chemical reactions, which will take Biocomputing to a new level. This can result in new healthcare implantables that embed engineered cells or Bio-hybrid computing systems, where computation is performed in synergy between the biological cells and silicon technologies.

## 6 ACKNOWLEDGMENTS

This work was supported by Science Foundation Ireland (SFI) and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland (Grant No. 16/RC/3835), and the National Science Foundation (NSF) (Grant No. 2316960). Authors also like to thank Clare Lyle (Google DeepMind) for valuable input.

## REFERENCES

[1] Youssef Abadade, Anas Temouden, Hatim Bamoumen, Nabil Benamar, Yousra Chtouki, and Abdelhakim Senhaji Hafid. 2023. A Comprehensive Survey on

TinyML. *IEEE Access* 11 (2023), 96892–96922. https://doi.org/10.1109/ACCESS.2023.3294111

[2] George Adam and Jonathan Lorraine. 2019. Understanding Neural Architecture Search Techniques. (2019). https://doi.org/10.48550/ARXIV.1904.00438

[3] Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, et al. 2012. NCBI GEO: archive for functional genomics data sets—update. *Nucleic acids research* 41, D1 (2012), D991–D995.

[4] Toygun Basaklar, Yigit Tuncel, and Umit Y Ogras. 2022. TinyMAN: Lightweight Energy manager using reinforcement learning for energy harvesting wearable IoT devices. *arXiv preprint arXiv:2202.09297* (2022).

[5] Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. 2021. On Interpretability of Artificial Neural Networks: A Survey. *IEEE Transactions on Radiation and Plasma Medical Sciences* 5, 6 (2021), 741–760. https://doi.org/10.1109/TRPMS.2021.3066428

[6] Cristopher GS Freitas, Andre LL Aquino, Heitor S Ramos, Alejandro C Frery, and Osvaldo A Rosso. 2019. A detailed characterization of complex networks using Information Theory. *Scientific reports* 9, 1 (2019), 16689.

[7] Peter J Gebicke-Haerter. 2023. The computational power of the human brain. *Frontiers in Cellular Neuroscience* 17 (2023).

[8] Babak Ghanbari, David Hartman, Vít Jelínek, Aneta Pokorná, Robert Šámal, and Pavel Valtr. 2023. Structure of betweenness uniform graphs with low values of betweenness centrality. *arXiv preprint arXiv:2401.00347* (2023).

[9] Nan Hao and Erin K O'Shea. 2011. Signal-dependent dynamics of transcription factor translocation controls gene expression. *Nature Structural and Molecular Biology* 19, 1 (Dec. 2011), 31–39. https://doi.org/10.1038/nsmb.2192

[10] Franklin M Harold. 1995. *Vital Force Study of Bioenergetics Pprsk.* W.H. Freeman, New York, NY.

[11] Aditya Kusupati, Manish Singh, Kush Bhatia, Ashish Jith Sreejith Kumar, Prateek Jain, and Manik Varma. 2018. FastGRNN: A Fast, Accurate, Stable and Tiny Kilobyte Sized Gated Recurrent Neural Network. *ArXiv* abs/1901.02358 (2018). https://api.semanticscholar.org/CorpusID:53681688

[12] Nick Lane and William Martin. 2010. The energetics of genome complexity. *Nature* 467, 7318 (2010), 929–934.

[13] Mikołaj Morzy, Tomasz Kajdanowicz, Przemysław Kazienko, et al. 2017. On measuring the complexity of networks: Kolmogorov complexity versus entropy. *Complexity* 2017 (2017).

[14] Yamila M Omar and Peter Plapper. 2020. A survey of information entropy metrics for complex networks. *Entropy* 22, 12 (2020), 1417.

[15] Christoph Ostrau, Christian Klarhorst, Michael Thies, and Ulrich Rückert. 2022. Benchmarking neuromorphic hardware and its energy expenditure. *Frontiers in neuroscience* 16 (2022), 873935.

[16] Hanny E Rivera, Hannah E Aichelman, James E Fifer, Nicola G Kriefall, Daniel M Wuitchik, Sara J Smith, and Sarah W Davies. 2021. A framework for understanding gene expression plasticity and its influence on stress tolerance. *Molecular Ecology* 30, 6 (2021), 1381–1397.

[17] Marina V. Rodnina. 2018. Translation in Prokaryotes. *Cold Spring Harbor Perspectives in Biology* 10, 9 (April 2018), a032664. https://doi.org/10.1101/cshperspect.a032664

[18] Samitha Somathilaka, Sasitharan Balasubramaniam, and Daniel P Martins. 2023. Analyzing Wet-Neuromorphic Computing Using Bacterial Gene Regulatory Neural Networks. (Dec. 2023). https://doi.org/10.36227/techrxiv.170327579.97407418/v1

[19] Samitha S. Somathilaka, Sasitharan Balasubramaniam, Daniel P. Martins, and Xu Li. 2023. Revealing gene regulation-based neural network computing in bacteria. *Biophysical Reports* 3, 3 (2023), 100118. https://doi.org/10.1016/j.bpr.2023.100118

[20] François Spitz and Eileen E. M. Furlong. 2012. Transcription factors: from enhancer binding to developmental control. *Nature Reviews Genetics* 13, 9 (Aug. 2012), 613–626. https://doi.org/10.1038/nrg3207

[21] Dmitry Sutormin, Alina Galivondzhyan, Olga Musharova, Dmitrii Travin, Anastasiia Rusanova, Kseniya Obraztsova, Sergei Borukhov, and Konstantin Severinov. 2022. Interaction between transcribing RNA polymerase and topoisomerase I prevents R-loop formation in E. coli. *Nature communications* 13, 1 (2022), 4524.

[22] Hector Zenil, Santiago Hernández-Orozco, Narsis A Kiani, Fernando Soler-Toscano, Antonio Rueda-Toicen, and Jesper Tegnér. 2018. A decomposition method for global evaluation of Shannon entropy and local estimations of algorithmic complexity. *Entropy* 20, 8 (2018), 605.

[23] Hector Zenil, Narsis Kiani, and Jesper Tegnér. 2018. A Review of Graph and Network Complexity from an Algorithmic Information Perspective. *Entropy* 20, 8 (July 2018), 551. https://doi.org/10.3390/e20080551

[24] Bai-Qing Zhang, Zong-Qin Chen, Yu-Qi Dong, Di You, Ying Zhou, and Bang-Ce Ye. 2022. Selective recruitment of stress-responsive mRNAs to ribosomes for translation by acetylated protein S1 during nutrient stress in Escherichia coli. *Communications Biology* 5, 1 (2022), 892.

[25] Shaojie Zhuo, Hongyu Chen, Ramchalam Kinattinkara Ramakrishnan, Tommy Chen, Chen Feng, Yicheng Lin, Parker Zhang, and Liang Shen. 2022. An Empirical Study of Low Precision Quantization for TinyML. https://doi.org/10.48550/ARXIV.2203.05492

# Chapter 9

# Journal: Analyzing Wet-Neuromorphic Computing Using Bacterial Gene Regulatory Neural Networks

| | |
|---|---|
| **Journal Title:** | IEEE Transactions on Emerging Topics in Computing |
| **Article Type:** | Regular Paper |
| **Complete Author List:** | Samitha S. Somathilaka, Sasitharan Balasubramaniam and Daniel P. Martins |
| **Status:** | Submitted. April 2024. doi.org/10.36227/techrxiv.170327579.97407418/v1 |

# Analyzing Wet-Neuromorphic Computing Using Bacterial Gene Regulatory Neural Networks

Samitha Somathilaka, *Student Member, IEEE,* Sasitharan Balasubramaniam, *Senior Member, IEEE*
Daniel P. Martins, *Member, IEEE,*

***Abstract*—The vision of biocomputing is to develop computing paradigms using biological systems, ranging from micron-level components to collections of cells, such as organoids. This paradigm shift exploits hidden natural computing properties, developing miniaturized wet computing devices deployable in harsh environments, and exploring designing novel energy-efficient systems. Parallelly, we witness the emergence of AI hardware including neuromorphic processors aiming to improve computational capacity. This study brings together the concept of bio-computing and neuromorphic systems by focusing on the Bacterial gene regulatory networks and their transformation into Gene Regulatory Neural Networks (GRNNs) that can be used for biocomputing. We explore the intrinsic properties of gene regulations, map this to a gene-perceptron function, and propose an application-specific sub-GRNN search algorithm that maps the network structure to match a problem. Focusing on the model organism Escherichia coli (E. coli), the base-GRNN is initially extracted and validated for accuracy. Subsequently, a comprehensive feasibility analysis of the derived GRNN confirms its computational prowess in classification and regression tasks. Furthermore, we discuss the possibility of performing a well-known digit classification task as a use case. Our analysis and simulation experiments show promising results in offloading computation to GRNN in bacterial cells, advancing wet-neuromorphic computing using natural cells.**

***Index Terms*—Biocomputing, Neuromorphic Computing, Bacteria, Gene Regulatory Network.**

## I. Introduction

Bacterial computing is an emerging field within the broader discipline of biocomputing [1]. The inherent computing properties of bacteria enable them in particular to sense their environment, make decisions, and adapt to changing conditions [2], with remarkable efficiency [3]. These characteristics, along with bio-compatibility, parallelism, self-sustainability [1], communication capabilities [4], [5], as well as storage of data [6] tend to provide bacterial computing an edge over conventional silicon-based computing architectures. Furthermore, the concept of neuromorphic computing is gaining traction, inspired from the workings of the neurons, showing promise compared to Von Neuman computing architectures [7]. By integrating natural biocomputing into neuromorphic systems, researchers are now exploring wet-neuromorphic computing, where living cells are used in tandem with silicon technology. This has resulted in new paradigms such as organoid intelligence [8], one of which is the brain organoid. The "Dishbrain" is one example of many approaches to such systems, which is a brain organoid that harnesses the inherent adaptive computation of brain neurons within a structured environment that is capable of learning and performing complex tasks such as playing a game of Pong [9].

Past research has extensively analyzed the functional components of natural bacterial computing, revealing a complex interplay of molecular processes that results in their decision-making and adaptive behaviors [10]. Signal transduction mechanisms facilitate adept extracellular information reception, followed by the complex orchestration of transcription and translation processes, resulting in a sophisticated computational architecture within bacterial cells using their Gene Regulatory Networks (GRNs). The synergy between these processes emphasizes the profound complexity of bacterial computing, underscoring its potential as a model for advanced bio-inspired computing paradigms within the confines of a single bacterial cell.

Synthetic biology has enabled researchers to use conventional computing theories that is engineered into cells by altering and modifying biological components precisely [11], [12]. While the feasibility of using biological substrates for computing has been established since Adleman's seminal work in 1994 [13], a number of proposals has been made in using bacterial cells for comptuing. One example is Levskaya et al., who proposed bacterial cell as a programmable computational device [14]. Expanding on this concept, Baumgardner et al. successfully programmed *Escherichia coli* (*E. coli*) with a genetic circuit using DNA segments [15]. This breakthrough has resulted in bacteria solving classical problem in artificial intelligence —the Hamiltonian problem. Theoretical models, such as the application of bacteria to solve the "*burnt pancake problem*" [16] have also reinforced the notion that bacteria possess computing capabilities that extend beyond traditional electronic systems. Furthermore, it is possible to witness biocomputing in multiple dimensions such as at the gene, metabolic and population levels. Exemplifying gene-level computation, researchers created encoding devices at the genetic level by altering the parameters of genes [7]. In addition, the computing capabilities in metabolic circuits are proved in whole-cell and cell-free environments in [17] that signify the metabolic-layer biocomputing, while [18] uses

Samitha Somathilaka is with VistaMilk Research Centre, Walton Institute for Information and Communication Systems Science, South East Technological University, Waterford, X91 P20H, Ireland and School of Computing, University of Nebraska-Lincoln, 104 Schorr Center, 1100 T Street, Lincoln, NE, 68588-0150, USA. E-mail: samitha.somathilaka@waltoninstitute.ie.

S. Balasubramaniam is with School of Computing, University of Nebraska-Lincoln, 104 Schorr Center, 1100 T Street, Lincoln, NE, 68588-0150, USA. E-mail:sasi@unl.edu

Daniel P. Martins is with School of Computer Science and Electronic Engineering (CSEE), University of Essex Wivenhoe Park Colchester CO4 3SQ, UK. E-mail: daniel.martins@essex.ac.uk.

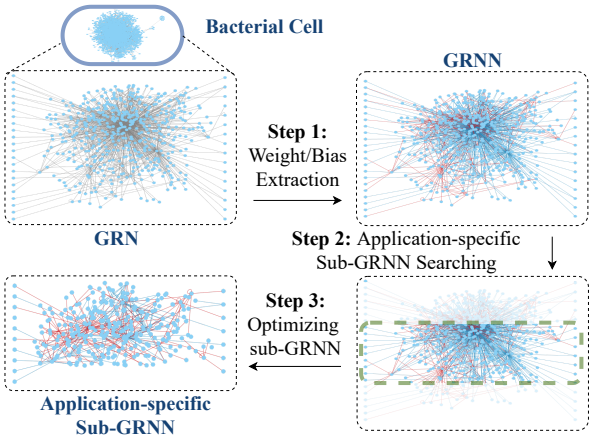



Fig. 1. Illustration of the discovery and development of Gene Regulatory Neural Network (GRNNs). Step 1 is the conversion of gene regulatory network to the GRNN by extracting weights and biases of the gene-gene interaction. Step 2 is the search process of the biocomputing application-specific sub-GRNN and Step 3 is the optimizing stage of the sub-GRNN.

bacteria consortia for pattern recognition giving an example for population-level biocomputing. All these approaches are based on engineering the cells. However, the challenges associated with engineered bacteria limit their practical use for computing. To this day, it is burdensome to design large-scale genetic circuits without stessing the cell, negatively affecting the circuit dynamics and overall reliability [19]. Moreover, the cross-talks between signaling expression pathways also narrows down the possibility of complex genetic circuit design [20]. This is also impacted by the gene regulatory mechanism that competes for expression resources, which further creates unintended cross-talks [21]. Ensuring the long-term stability of engineered cells is also challenging under the altered gene expression pathways as the modifications may affect cell viability and growth rates [19]. This motivates us to explore the possibility of using the inherent computing functions within the cells through external chemical control, without modifying the cells' internal genetic system.

Acknowledging the complex computing architecture similar to a wet-neuromorphic system inherent in bacterial cells (that is further explored in the next section), this study considers them as natural computing powerhouses where the GRN serves as the central computing mechanism as shown in Fig. 1. Owing to the remarkable computing diversity embedded in the GRN, we claim that it is possible to search and extract computing layer suitable for various application problems. In order to test this, we first use an improved version of a previously introduced framework for quantifying the gene-gene interactions [22] that returns a weighted directional network with nodes as genes and edges as expression interaction with the capability of processing the incoming regulatory factors. This network is analogous to a random structured Neural Network (NN), hence, we termed this **Gene Regulatory Neural Network** (**GRNN**) [22]. We then propose an algorithm to extract application-specific sub-networks (sub-GRNNs) from the GRNN as depicted in the second and third steps of Fig. 1 to do application specific computing. The computing application that we will focus in this paper is performing mathematical regression as well as classification tasks using *E. coli* GRNNs.

The contributions of this article are as follows:

- **Modeling the *E. coli* GRNN for *in-silico* experiments:** Our previous studies proved the existence of GRNN [22] using experimental gene expression data. We extend this model in this study, by focusing on the GRNN of *E. coli* and proves its accuracy with the intention of exploring their natural computing capabilities to solve computer science problems.
- **Introducing application-specific sub-GRNN search algorithm:** As we discussed earlier, GRNNs are considered pre-trained NNs leading to significant differences in the application pipelines. Therefore, we introduce a search algorithm tailored to GRNNs that includes extractions of sub-GRNNs by mapping to an application problem. In this approach, we only use a random permutation-based approach as our main goal is to check the GRNN computing feasibility.
- **Feasibility analysis on performing computing on regression and classifications:** We evaluate the feasibility of performing conventional computing tasks such as regression (including linear, multiple variable linear, $2^{nd}$ and $3^{rd}$ degree polynomial regressions) and classification (including binary and multi-class) using the extracted *E. Coli* GRNN. Further, we solve a digit classification problem and evaluate its accuracy as a case study.

The rest of the manuscript is structured as follows. Section II explores the background of GRN computing properties using the literature, subsequently, the existence of GRNN. The next section creates the *E. coli* GRNN using experimental transcriptomic data and proves its accuracy. Further, the same section conducts a structural analysis and introduces an algorithm for application-specific sub-GRNN. Section IV and V conduct feasibility analyses on performing regression and classification problems including a digit classification use-case problem. This is followed by a discussion and conclusions in Section VI.

## II. Background

We identified that the natural computing of a bacterial cell is manifested within the domain of gene regulation. The bacterial cell's genome serves as a repository of encoded information and its dynamic regulation is through the transcription, translation, and post-translational modifications, which constitutes a complex computing network.

### A. Gene as a Computing Unit

The expression process of a gene can be considered intra/extra cellular information computing that results in functional gene products, such as proteins or non-coding RNAs [23]. Prokaryotic genes are often organized into operons, clusters of genes transcribed as a single mRNA molecule with

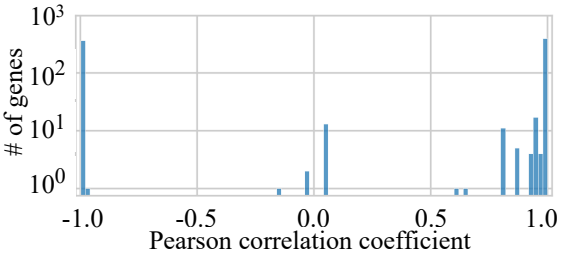



Fig. 2. Gene expression Pearson correlation coefficients between the source and target genes with only one inward and no outward edges. These results indicate the expression levels of the source-target gene pairs have linear relationships.

a shared promoter. Transcription is regulated by specific DNA-binding proteins near the promoter, influencing RNA polymerase activity. The transcribed mRNA, complementary to the DNA template, undergoes translation in the ribosomes, which is composed of rRNA and proteins. This tightly regulated and coordinated process is vital for the cell to execute specific functions and respond to environmental cues.

Many characteristics of genes in this complex regulatory process exhibit similarity to an artificial neuron. The inherent computing capabilities of genes have been investigated in past decades gradually paving the path toward using biological entities for computing. Since the pioneering work of DNA computing in 1994 [13], the field has been relentlessly advancing as an alternative to conventional computing methods. Among many diverse approaches, utilizing genes as perceptrons [24] holds a special position due to the crucial role Artificial Neural Networks (ANNs) play in today's computing world. An artificial neuron can accept multiple inputs, which are then weighted and summed, before passing through an activation function. Similarly, a gene can receive multiple Transcriptional Factors (TFs) that lead to the combinatorial regulation of the target gene [25]. Further, properties of the TFs including the affinity of the TF-binding site and mechanisms such as thermoregulators/enhancers/silencers [26], [27] and lifetime of a DNA-TF complex [28] can determine the magnitude of the source TFs' impact on the target gene resulting in expression level regulation.

In an ANN, the weighted summation of an artificial neuron is then passed through an activation function such as sigmoid, tanh, ReLU, etc., that introduces non-linearity to the computing. Similar behavior can also be witnessed in genes from two perspectives. First, a gene is considered a switch that is either in the "on" state or "off" state that is determined by the regulatory influence of the TFs. Here, the combined influence of the TFs as the weighted summation is converted into the two-state output as the "on" or "off" state of the transcription [29], exhibiting sigmoid-like gene expression dynamics.

However, in order to observe the gene expression behaviors beyond the sigmoidal properties, this study conducted a simple correlation analysis on the temporal expression dynamics utilizing an *E. coli* dataset (accession number GSE65244) from the GEO database [30]. Here, we used 827 target genes with single inward and no outward edges to observe the correlation clearly. Based on our analysis, 95.40% of the considered target genes have a correlation coefficient greater than 0.9, while 4.11% of genes have a coefficient less than -0.9, demonstrating that there are strong linear relationships between the expressions of the source and the target genes. Further, only 0.49% of source-target gene pair expressions have correlation coefficients within the range of -0.9 and +0.9. This analysis reveals that most of the source and target genes have linear relationships as shown in Fig. 2. This emphasizes that the relationships between the source and target genes can be converted to a single value (a.k.a weight) and the suitability of the ReLU activation function over sigmoid to represent a gene's expression behavior in the time domain. Nevertheless, there is a biophysical boundary for the maximum gene expression rate, which emphasizes the requirement of using Bounded Rectify Linear (BReLU) activation function. This argument is proved in Section III-A using the accuracy of the extracted GRNN, where the gene expression behavior is considered to follow BReLU.

Moreover, a further investigation into the gene expression properties explains that the prokaryotic genes have "ground states" as the RNA polymerase can access almost any promoter without the presence of activators or repressors [31]. Considering this property, we improve our previous weight extraction mode by accompanying each gene-perceptron with a bias representing the ground state as shown in Fig. 3

### *B. Existence of GRNN*

Interactions between genes driven by the functional proteins produced by the transcription/translation process form a complex network that is part of the GRN. This collectively involves operons, modules, and motifs, that together achieves coordinated and integrated gene expression [32]. An operon is a set of co-regulated and expressed genes that share a common promoter and produce a single mRNA molecule. A module is a group of operons that are regulated by the same TF or signal. A motif is a recurring pattern of interactions among regulators and genes that has a specific function, such as feedback loops and feedforward loops [33]. A GRN can process and transmit information through these structures, forming a cascade of signals that modulate the activity of downstream genes. Transcription patterns along with the topologies of the GRNs that drive the decision-making process of bacterial cells [34] hint at the existence of complex computing properties.

With the aim of revealing these computing properties inherent in bacterial cells, we introduced a framework for quantifying the dynamics of gene-gene interactions [22]. This framework was designed by perceiving the GRN as a random structured graph network where genes act as nodes and gene-gene interaction as edges. Using the relative transcriptomic data, we quantified the influence of a source gene on a target gene. The GRN with weighted interactions can be considered a gene regulation-based random structured NN, which is when we transform it into a GRNN. This, further reflects our perception of the genetic regulatory processes in

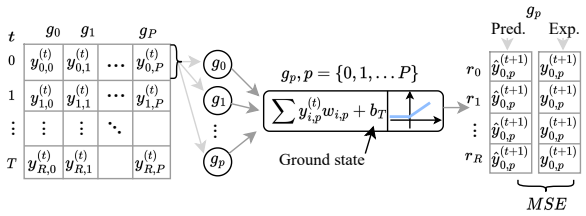



Fig. 3. Illustration of the gene-perceptron model of gene $g_p$, where weighted summation of source gene ($g_0$, $g_1$,...,$g_P$) expression levels at timestep $t$ passes through the activation function BReLU and produces an expression level at timestep $t+1$ corresponding to the input. This figure also elucidates the weight extraction mechanism, where we use temporal transcriptomic records to refine the weights between the predicted and expressed genes, as well as the biases.

bacterial cells, acknowledging their complexity and suggesting a parallel with the computational principles observed in ANNs.

## III. Base-GRNN creation and Application-specific sub-GRNN Extraction

In this section, the improved version of the GRN to GRNN conversion framework from [22] is first explained. Subsequently, we delve into the essential components of GRNNs, encompassing network structures as well as their analog and parallel computing capabilities. Furthermore, as highlighted previously, the conventional NN learning stage has been substituted by the network architectural search in the context of GRNN-based computing, which we will present in this section.

### A. GRN-to-GRNN Conversion

The GRN represents a distinctive gene-gene interaction network specific to each species. Publicly accessible GRN databases covering various species or strains such as E. coli can be found in [35]. Typically, these GRNs only contain data on the existence and type of interactions between static features such as genes, operons, TFs (including Sigma Factors - SFs). The absence of quantitative properties including the magnitude of the impact of one element on another hinders the extraction of accurate natural computing capabilities of biological cells. To overcome such obstacles, we previously introduced a GRN-to-GRNN conversion method [22] to quantify the influence of TFs on the regulation of a target gene.

The previously proposed GRN-to-GRNN conversion method consists of multiple stages that include GRN modeling as a graph network, dividing of the GRN to gene-perceptron, pre-processing of the transcriptomic data, and weight extraction [22]. Initially, the GRN is reconstructed as a directed graph network in which genes are modeled as nodes and their interactions as edges. This GRN is then divided into sub-networks associated with each gene that has at least one inward edge. These sub-networks are similar in structure to the single-layer perceptron since they contain a target gene (the gene with at least one edge) and a set of source genes that regulate its expression. Subsequently, these sub-networks are termed single-layer gene-perceptrons. The transcriptomic data, on the other hand, consists of the expression levels of both the source and target gene(s) in each gene-perceptron, which can be leveraged to assess the strengths of the interactions between the source and target gene(s). There is evidence suggesting that gene-perceptron expressions in the temporal domain exhibit BReLU properties, as previously described in Section II.

Here, we employ a mechanism akin to the training process of the single-layer perceptron to quantify the interactions between source and target genes as shown in Fig. 3. As explained earlier, prokaryotic genes tend to have a ground state, which is identified as the bias of the gene-perceptron model. Therefore, the current study improves the GRN-to-GRNN conversion mechanism by embedding a bias for gene-perceptrons with the intention of extracting the dynamics of gene-gene interaction. The gene-perceptron of this study is depicted in Fig. 3 and its functions represented as follows

$$\hat{y}_p^{t+1} = max\left(\sum_{i=1}^{P} y_{(i,p)}^{t} w_{(i,p)} + b_p\right), \tag{1}$$

where $\hat{y}_p^{t+1}$ is the computed output of the target gene $g_p$ at the time step $t$, $y_{(i,p)}^{t}$ is the output of the gene $g_i$ at time step $t$, $w_{(i,p)}$ is the weight of the interaction between gene $g_i$ and the target gene, and $b_p$ is the bias or the ground stage. We also consider the impact of source gene expression levels of time-step $t$ on the expression level of the target gene at the time-step $t+1$. As the initial step, the weights of gene-perceptrons are initialized with random weights and are subsequently adjusted with an iterative process of minimizing the Mean Squared Error (MSE) between the predicted (from the gene-perceptron) $\hat{y}_p^{t+1}$ and measured (from the transcriptomic data) $y_p^{t+1}$ expression, which is represented as follows

$$MSE(g_p) = \frac{1}{T}\sum_{t=0}^{T}(y_p^t - \hat{y}_p^t)^2 \tag{2}$$

where $T$ is the last time step. This process is iterated for all the gene-perceptrons to extract the weights of all the interactions within the GRN.

### B. E. Coli Base-GRNN

The GRN-to-GRNN conversion method explained in Section III-A is now applied to *E. coli* k-12 strain CSH50 to extract its base-GRNN that is used for all the analysis in this study. In the first stage of the conversion, the GRN data is obtained from [35] under multiple categories including TF - gene, TF - operon, TF - Transcription Units (TU), TF - TF, SF - Gene, SF - TU and sRNA - gene. Merging the interactions in each category, the complete GRN of *E.coli* is created as a directed graph network that consists of 3175 genes as nodes and 9678 interactions as edges.

As the next step, the GRN is divided into single-layered gene-perceptrons that contain corresponding source genes and initialized with random weights. For the weight/bias extraction phase, temporal transcriptomics data [30] (GEO accession number GSE65244) that contains interpolated expression records for 43 time-steps is used after normalizing. However,

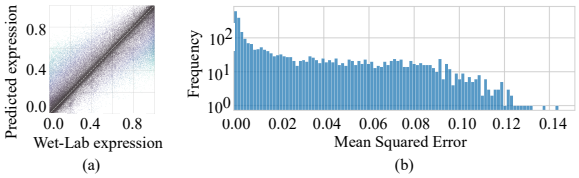



Fig. 4. Illustration of the accuracy of the extracted base-GRNN, where a) compares the predicted $\hat{y}$ and wet-lab experiment $y$ expression levels for all the timesteps of all the gene-perceptrons and b) showes the MSE of each gene for all the timesteps.

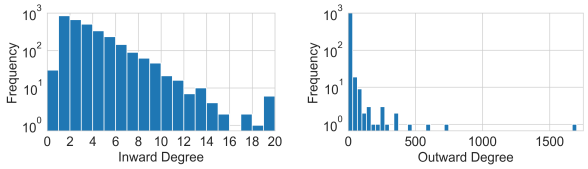



Fig. 5. Network degree distribution of the *E. coli* GRNN, where a) and b) show inward and outward degree frequency, respectively.

we only use 34 expression records which is around 80% of the total records for the weight/bias extraction, while the rest of the transcription records are used to evaluate the accuracy of the extracted weights and biases. The learning rate and the number of epochs are then set to $10^{-5}$ and $10^9$, respectively, and the weights and biases for all the gene-perceptrons are extracted iteratively. Finally, the extracted weights and biases are incorporated with the GRN converting it into the base-GRNN.

The extracted weight matrix is denoted as,

$$\mathbf{W} = \begin{array}{c} \\ g_1 \\ g_2 \\ \vdots \\ g_P \end{array} \begin{array}{c} \begin{array}{cccc} g_1 & g_2 & \cdots & g_P \end{array} \\ \begin{pmatrix} w_{(1,1)} & w_{(1,2)} & \cdots & w_{(1,P)} \\ w_{(2,1)} & w_{(2,2)} & \cdots & w_{(2,P)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(P,1)} & w_{(P,2)} & \cdots & w_{(P,P)} \end{pmatrix} \end{array}, \tag{3}$$

where $w_{(i,j)}$ is the weight of the interaction between $i^{th}$ and $j^{th}$ gene where $i : j = \{1, 2, .., P\}$. The $w_{(i,j)}$ is the weight of self-regulation interaction when $i = j$.

Next, we model the output of the GRNN, $\mathbf{O}^{(t+1)}$ at $t + 1$ using weight $\mathbf{W}$ as,

$$\mathbf{O}^{(t+1)} = max(\mathbf{W} \cdot (\mathbf{I}^{(t)} + \widetilde{\mathbf{N}}) + \mathbf{B}), \tag{4}$$

where $\mathbf{I}^t$ is the input matrix $\mathbf{B}$ is the bias matrix and $\widetilde{\mathbf{N}}$ is the Gaussian noise $\widetilde{\mathbf{N}} = N(0, 0.1)$ extracted based on the iterative experiments [30] (GEO accession number GSE215300). For the next time step, the input matrix $\mathbf{I}^{t+1} = \mathbf{O}^{t+1}$ and $\mathbf{O}^{t+2}$ is computed as,

$$\mathbf{O}^{(t+2)} = max(\mathbf{W} \cdot (\mathbf{I}^{(t+1)} + \widetilde{\mathbf{N}}) + \mathbf{B}). \tag{5}$$

The accuracy of the extracted *E. coli* base-GRNN is evaluated by comparing the predicted gene expression with the real values from the data set [30] (GEO accession number GSE65244). The predicted gene expression rates are plotted against the wet-lab values in Fig. 4a, where the $45^0$ degree dashed line represents the predicted and wet-lab gene expression values are the same. According to the plot, most of the predicted and wet-lab expression values lie close to the $45^0$ line. Further, a histogram of the Mean Squared Error (MSE) between the predicted and wet-lab expression levels is shown in Fig. 4b, statistically elucidating that more than 90% of the gene-perceptrons have less than 0.1 MSE. This evidently establishes three key points. 1) the possibility of converging multi-stage gene-gene interactions to a quantitative value (a.k.a weights and biases), 2) the proposed single-layered gene-perceptron-based [22] weight extraction model is cross-genome compatible (as this method is applied on *P. aeruginosa* in [22], while this study employs it on *E. coli*), and lastly 3) the extracted base-GRNN of *E. coli* perform similarly to the gene regulation mechanisms of a biological cell.

The structural and algorithmic complexity behaviors of the *E. coli* base-GRNN is analyzed in the next section.

### C. GRNN Structural and Algorithmic Complexity

The base-GRNN consists of a graph topology with a power-law distribution that contains a few hub nodes and a significant number of terminal nodes as evident in Fig. 5. Withing the *E. coli* GRNN, 68.45% of the gene-perceptrons have more than one inward edge (Fig. 5a) with the ability of computing multiple inputs together. Moreover, the distribution of the outward edges as shown in Fig. 5b proves the existence of hub gene-perceptrons that can influence around 92.12% of terminal nodes. This feature contributes to making the base-GRNN suitable for a range of problems. For instance, the gene *b3067* has a total of 1703 outward edges, of which 91% are terminal nodes. Activation of this particular gene-perceptron in turn results in a wide range of expression values in the terminal nodes. Therefore, this power law distribution positively reflects the computing diversity. This allows the base-GRNN to be recognized as an extensive repository of diverse pre-trained sub-GRNNs.

An important factor of NN is their structural and algorithmic complexities. We will also need to analyze GRNN for their structural and algorithmic complexities and compare this to the conventional NNs to determine its performance. Structural complexity pertains to the network's architecture, encompassing factors such as the number of neurons/edges, and topology. This complexity has a direct impact on the network's computational capabilities, while the algorithmic complexity corresponds to the computational requirements of the training and inference processes in NNs. Therefore, in this study, we conduct an analysis of the structural and algorithmic complexities of GRNNs, comparing them with those of conventional fully connected NNs. The calculation of the structural entropy $S_c$ begins by computing the betweenness centrality of all the gene-perceptrons, where we denote the betweenness of the $i^{th}$ gene-perceptron $l(i)$ as follows

$$l(i) = \sum_{1 \leqslant i \leqslant N, s \neq i \neq t} \frac{\sigma_{s,p}(i)}{\sigma_{s,p}} \tag{6}$$

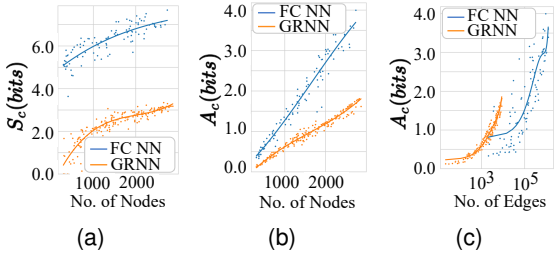



Fig. 6. A comparison of structural ($S_c$) and algorithmic ($A_c$) complexity behaviors between Fully Connected Neural Networks (FCNNs) and GRNNs. a) and b) compare the $S_c$ and $A_c$ variations against the number of nodes in the two types of NNs, while c) focuses on $A_c$ against the number of edges.

where $\sigma_{s,p}$ is the total number of shortest paths between the source $g_s$ and target $g_p$ genes, while $\sigma_{s,p}(i)$ is the number of shortest paths through gene $g_i$ between the source gene $g_s$ and target gene $g_p$.

Further, we define the relative degree $p_i$ of the $i^{th}$ node as,

$$p_i = \frac{Degree(i)}{\sum_{i=1}^{N} Degree(i)}, \tag{7}$$

and $q_i$ as the nonextensive parameter which is defined based on the betweenness $l(i)$ as follows,

$$q_i = 1 + (l(max) - l(i)), \tag{8}$$

where $l(max) = max[l(i), (i = 1, 2, 3, ..., N)]$.

Finally, structural complexity $S_c$ is calculated as follows

$$S_c = \sum_{i=1}^{N} \left( \frac{p_i^{q_i}}{\sum_{i=1}^{n} p_i^{q_i}} \right) log \left( \frac{p_i^{q_i}}{\sum_{i=1}^{n} p_i^{q_i}} \right). \tag{9}$$

This study uses the Kolmogorov complexity (K-complexity) approximated by the Coding Theorem Method ($CTM$) to determine the algorithmic complexity $A_c$, which is considered the basis for the network complexity [36]. $CTM$ is calculated based on the Laplacian matrix $L$, and due to the large dimensions, we also employ the Block Decomposition Method ($BDM$) as follows

$$A_c \approx BDM(L) = \sum_{i=1}^{P} CTM(b_i) + log_2|b_i|, \tag{10}$$

where $b_i$ is the $i^{th}$ row of $L$ and more information on estimating the $CTM$ of $b_i$ can be found in [37].

Fig. 6a compares the behavior of the structural complexity with respect to the number of nodes in a fully connected NN versus GRNN. It is evident that the power-law properties in the random structured GRNNs compared to fully connected NN result in lower structural complexity. We also found lower algorithmic complexity in GRNN structures which is shown in Fig. 6b due to a minimized number of edges compared to a fully connected NN with a similar number of nodes. Moreover, GRNNs comprising edges ranging from 2000 to 10000 exhibit greater algorithmic complexity compared to fully connected NNs as depicted in Fig. 6c. This reveals certain GRNN structures are capable of complex computing, while maintaining an improved interpretability compared to fully connected GRNNs.

### *D. Application-specific sub-GRNN Search Algorithm for Classification*

Owing to the fact that GRNNs are considered pre-trained random structured NNs, problem-solving using GRNNs requires searching and extraction of the precise sub-GRNN. Therefore, as one of the main objectives of this study, we propose an application-specific sub-GRNN search algorithm for classification which is illustrated in Fig. 7. This search algorithm uses a random permutation-based method to find the most suitable sub-GRNN to match a problem, aiming to compute it with high accuracy.

In the initial step, suitable candidates for the input layer are first filtered using the characteristics of the genes, including inward/outward degree as shown in Fig. 7 (Step 1). These characteristics include (i) Gene-perceptrons with an inward degree closer to zero, which are not significantly influenced by unnecessary incoming signals except for the problem-specific inputs and (ii) input gene-perceptrons with a higher outward degree so they have variety of output combinations that can support complex computing capabilities. This stage uses graph theoretical degree distribution to create a set of gene-perceptrons for the input layer, $G(Trimmed)$. Here, the $G(Trimmed)$ contains $P'$ number of genes, where the selected genes, $P'$, is less than the total number of genes, $P$, of the GRNN, respectively. Given the $G(Trimmed)$ contains $P'$ number of gene-perceptrons and the problem has $K$ number of features, there are ${}_{P'}P_K (= \frac{P'!}{(P'-K)!})$ number of different input layers that can be extracted. Due to the massive number of sub-GRNNs, a heuristic search algorithm may be more efficient. However, exploring such algorithms are out of this study's scope and we only use this random permutation-based algorithm.

Subsequently, in the same step (Fig. 7-Step 1), the algorithm randomly picks a set of $K$ number of inputs denoted as $G(In_J)$, for the $J^{th}$ permutation, where $J = \{0, 1, 2, ..., {}_{P'}P_K\}$, where $K$ is number of input features of the problem. This search algorithm requires a dataset (termed as the search dataset $SD_{K\times V}$), which is similar to the training data in conventional NN training to evaluate the fitness of each sub-GRNN to the problem, where $V$ is the number records with corresponding class labels. Before the $SD$ is encoded into expression levels in Fig. 7 (Step 2), a base-TF array is created using the expression levels at the zero timestep of the transcriptomic data used for the weight extraction [30] (GEO accession number GSE65244). This step is crucial to mimic the base behavior of the cell at $t = 0$ (computing process starting time), where the cell functions with respect to the environmental conditions. The base-TF array is then altered adequately to encode the inputs of $SD$ to create the input matrix $\mathbf{I}^{(t=0)}$ that contains input TF input arrays $I_v^{(t=0)}$, where $v = \{0, 1, 2, ..., V\}$. In this stage, if the $SD$ is considered digital, then state "1" represents the highest expression level of the corresponding gene, while state "0" the

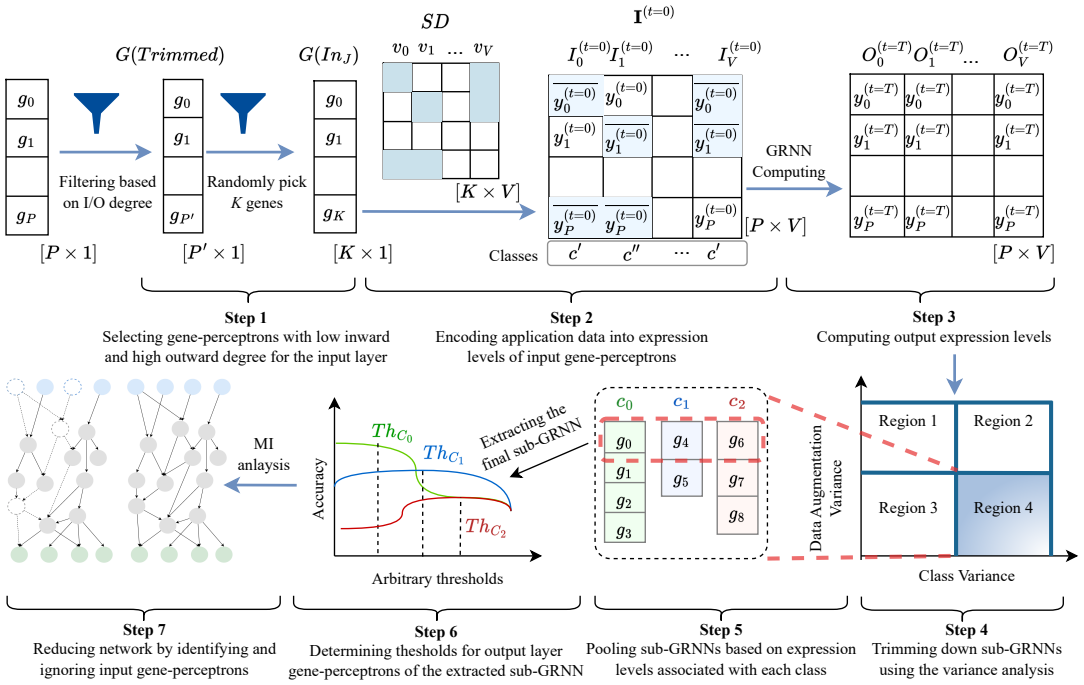



Fig. 7. Illustration of the proposed application-specific sub-GRNN search algorithm for One-vs-All classification. Step 1 focuses on selecting a set of input gene-perceptrons, $G(Trimmed)$ based on their inward/outward degree distributions, and a subset, $G(In_j)$ with $K$ ($K$ is the number of input features of the application) number of gene-perceptrons from $G(Trimmed)$ is selected. In Step 2, the searching dataset $SD$ is encoded into the expression level-based input matrix $I^{(t=0)}$. The output matrix, $O^{(t=T)}$, corresponding to $I^{(t=0)}$, is then calculated using the *in-silico* base-GRNN model as explained in Section III-B. In Step 4, a set of gene-perceptrons is identified, exhibiting higher expression variance between classes and lower expression variance within the same class. This set of gene-perceptrons is then pooled under each class in Step 5 based on their expression levels. Step 6 searches for the optimal expression thresholds for each class by maximizing accuracy. Finally, Step 7 conducts an MI analysis to identify the insignificant input gene-perceptrons and removes them from the input layer to reduce the size of the network.

lowest value. However, if the $SD$ is in analog form, the values are normalized and mapped with the concentrations based on the highest and lowest expression levels of the relevant gene. After decoding all the input records with the expression levels in Step 2, using the mathematical model explained in (4) and (5), the output expression levels are computed in Step 3. This step produces an expression matrix, $O^{(t=T)}$, with output arrays, $O_v^{(t=T)}$, where $v = \{0, 1, 2, ..., V\}$ corresponding to each class.

In Step 4, we conduct a variance analysis with the intention of identifying genes-perceptron that can be used to represent each class at the output layer of the sub-GRNN. Suppose a gene-perceptron can express in a higher level for the corresponding input, $I^{(t=0)}$ of a particular class, $c_i$, while maintaining low variance between augmentations in the same class and higher variance between different classes. In this case, that gene-perceptron is a good candidate to represent $c_i$. Hence, we search for gene-perceptrons for all the classes in "Region 4" (as shown in Fig. 7(Step 4)), where the variance between classes is high and the variance between records of the same class is low, which we will then form a set of output gene-perceptrons.

In Step 5, if a gene-perceptron $g_i$ from the above set fulfills the condition, $\overline{y}(g_i, c_l) > \overline{y}(g_i, c_m) : m\forall; m < |c|, m \neq l$, where $\overline{y}(g_i, c_l)$ is the mean expression level for class $c_l$, then $g_i$ is pooled under the class $c_l$. This process is repeated for all the gene-perceptrons in "Region 4". Following this, the gene-perceptrons with the highest mean expression level that has the largest gap with the rest of the gene-perceptrons is selected to represent each class.

The Step 6 of the algorithm is dedicated to identifying the threshold for each gene-perceptron using an accuracy-maximizing approach. First, we get the *true-positives* (TP), *true-negatives* (TN), *false-positives* (FP) and *false-nagatives* (FN) for each class using an arbitrary threshold value, $Th = a$. The accuracy of the classification is then calculated for class $c_l$, which is denoted as $ACC(c_l, Th = a)$ and represented as follows

$$Acc(c_l, Th = a) = \frac{TP + TN}{TP + TN + FN + FP}. \quad (11)$$

This calculation is repeated with various thresholds $a$ ranging from zero to one with 0.05 increments and the threshold calculated for the class $c_l$ is as follows

$$Th = \arg\max_a Acc(c_l, Th = a). \quad (12)$$

Similarly, the thresholds for all the classes associated with the problem are extracted iteratively.

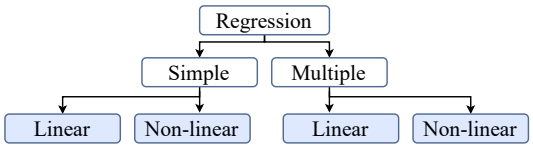



Fig. 8. Illustration of sub-categories of regression problems.

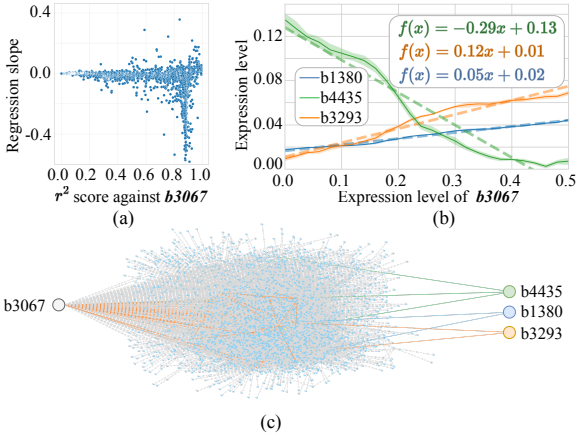



Fig. 9. Illustration of simple linear regression using *E.coli* GRNN, where a) shows the regression slope distribution of all the genes against the respective $r^2$ score, b) exemplifies three regression lines based on three output gene-perceptrons and c) is the sub-GRNN for the linear regressions.

Finally, this process is repeated a number of times ($< {}_{P'}P_K$), where this will result in ranking of the sub-GRNNs based on accuracy that will result in the selection of the best candidate.

## IV. GRNN Application in Regression

Mathematical regression has been widely used in data mining applications [38], [39] for many years. Therefore, one key contribution of this study is the GRNN's expression behaviour that matches regression problems, which will be analyzed in this section. Fig. 8 illustrate the types of regression problems we consider under the regression feasibility analysis.

### A. Linear Regression Analysis

First, we identified *b3067* as the *E. coli* input gene-perceptron that has the highest outward degree that can regulate 1703 other connected gene-perceptrons. This is important for this analysis, as the stimulation of the *b3067* cascades through a significant portion of the GRNN, leading to diverse regression outputs.

Next, the input gene-perceptron is stimulated with 25 concentration input values ranging from 0 to 0.5 normalized concentration units. The initial expression values of the rest of the gene-perceptrons are kept at the minimum level based on the expression profiles in [30] (GEO accession number GSE65244). Each step of this experiment is also iterated 10 times to observe more accurate behaviors.

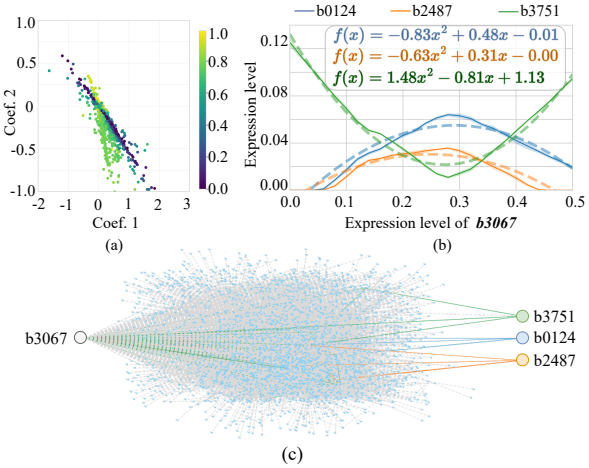



Fig. 10. Illustration of non-linear quadratic regression using *E.coli* GRNN, where a) shows the quadratic and linear coefficient distribution of all the genes that are color-coded to the $RSS$ value, b) shows three example regression curves and c) is the sub-GRNN associated with the three example regression curves.

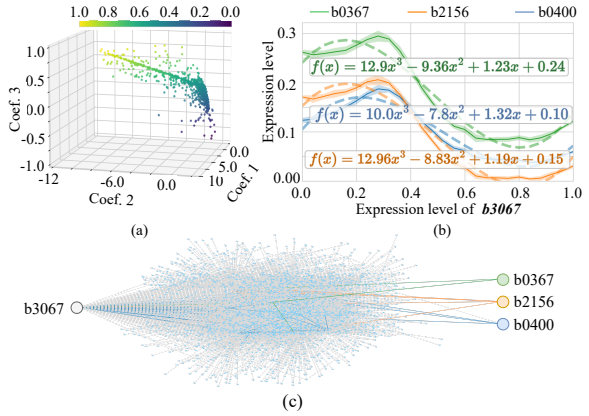



Fig. 11. Illustration of non-linear cubic regression using *E.coli* GRNN, where a) shows the cubic, quadratic and linear coefficient distribution of all the genes that are color-coded to the $RSS$ value, b) shows three example cubic regression curves and c) illustrates the extracted sub-GRNNs of the three cubic regression curves.

Our aim is to determine linear regression functions that match the gene-perceptron expression profiles and to also determine the flexibility of finding other functions with different coefficients. We first extracted the expression levels of all the gene-perceptrons and calculated the $r^2$ score ($r^2 = RSS/TSS$, where $RSS$ is the residual sum of squares and $TSS$ is the total sum of squares) to measure the goodness of fitness of the output gene-perceptron expression with a linear approximation. Fig. 9a shows the variety of linear regression slopes with respect to the $r^2$ fitness, where the gene-perceptrons with the highest $r^2$ values tend to have a variety of coefficients for different slopes. Further, this plot reveals that the highest positive slope is estimated as 0.36 while the -0.50 is the largest negative slope for *E. coli* GRNN when the

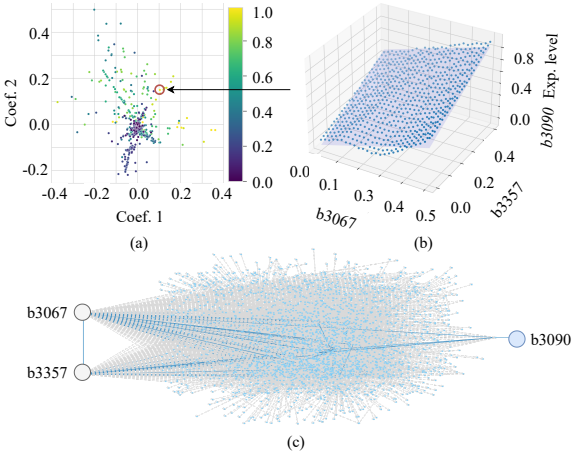



Fig. 12. Illustration of multiple-linear regression using *E.coli* GRNN by using gene-perceptrons *b3067* and *b3357* as two inputs, where a) shows the first and second coefficients distribution of all the genes that are color-coded to the $RSS$ value, b) and c) shows the example plane of the output gene-perceptron *b1411* and the corresponding sub-GRNN, respectively.

input is *b3067*. Fig. 9b presents three output gene-perceptrons for different regression lines, where *b1380* and *b3293* have positive slopes of 0.29 and 0.12, respectively, while *b4435* has a negative regression slope of -0.29. Fig. 9c illustrates the sub-GRNNs associated with the three output gene-perceptrons, *b4435, b1380* and *b3293* with corresponding color codes to Fig. 9b. According to Fig. 9c, it is essential to highlight that all the three linear regressions are done parallelly, proving the parallel computing properties of the GRNN. This analysis evidently elucidates the availability of a diverse linear regression solution space, where an algorithm can search and map gene-perceptrons to applications.

### B. Quadratic Polynomial Regression

The GRNN output generated in the previous section is utilized also for matching to the quadratic polynomial regressions. Fig. 10a represents the behavior of the quadratic (Coef. 1) and linear (Coef. 2) coefficients of each gene perceptron that is color-coded according to the $RSS$ value, where the lighter color (yellow) indicates the higher goodness of fit. The quadratic coefficients of the curves with the highest $RSS$ values range from -2 to 2, while the linear coefficient ranges -1 to 0.5. Fig. 10b shows three example curves to emphasize the diversity of the available quadratic regression within the *E. coli* GRNN given the input gene-perceptron as *b3067*. Three quadratic curves shown in Fig. 10b are for the gene *b0124*, *b2487* and *b3751* where the quadratic coefficients are -0.83 - 0.63 and 1.48, respectively. Figure 10c depicts the sub-GRNNs corresponding to the three output gene-perceptrons, namely *b0124*, *b2487*, and *b3751*, each represented with distinctive color codes as aligned with Fig. 10b. This shows that with the same input gene-pereceptron *b3067*, we can switch from the linear regression to quadratic polynomial regression by finding a different output gene-perceptron combination.

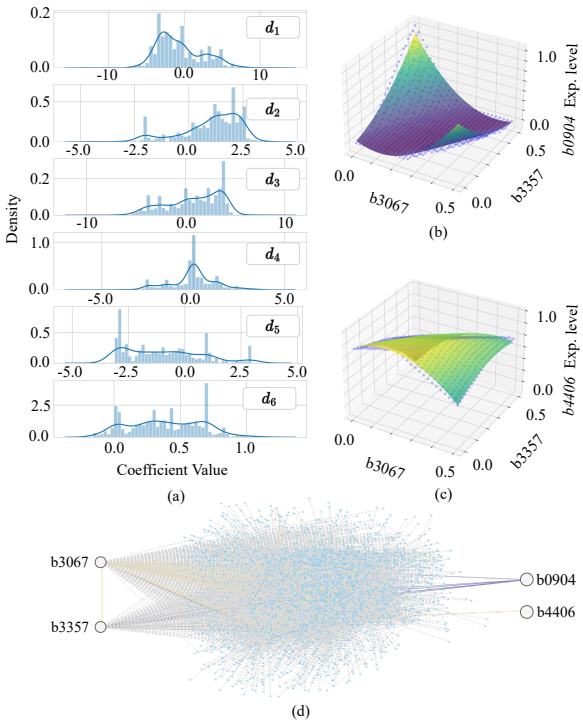



Fig. 13. Illustration of multiple non-linear regression using *E.coli* GRNN by using gene-perceptrons *b3067* and *b3357* as two inputs, where a) shows the distributions of each coefficient associated with (13) while b) and c) exemplify two curves with positive and negative coef. 1, respectively. Subsequently, d) shows the sub-GRNNs for the two examples regressions shown in b) and c).

### C. Cubic Polynomial Regression

The same data set used in Section IV-A and IV-B is employed to discover and match to the cubic polynomial regression of *E. coli* GRNN. Fig. 11a shows the coefficients of the cubic polynomials and Fig. 11b provides three example curves, while Fig. 11c illustrates the corresponding three sub-GRNNs. The cubic coefficient with $RSS > 0.7$ ranges approximately from 0 to 13, while the quadratic and linear coefficients have ranges of -11 to 2 and -0.75 to 0.75, respectively. The ranges of the data points for the curves in Fig. 11 are not spread out, which means there is a minimized variation in the higher-degree polynomial regressions. This can be perceived as a limitation in discovering higher degree functions. Nevertheless, it is essential to mention that these solution spaces are extracted only using input gene-perceptron *b3067* as an example. Therefore, the solution space can be immensely expanded by using different input-gene perceptrons.

### D. Multiple Linear Regression

We further investigate the feasibility of multiple regression of *E.coli* GRNN by using two input gene-perceptrons, *b3067* and *b3357* with outward degrees of 1703 and 596, respectively. Similar to the previous setup, the two inputs are stimulated with expression levels from 0 to 0.5 with 0.02 increments

(a total of 625 input setups.). In this analysis, the $RSS$ is considered the measure of variance of the regression model. Fig. 12a is the coefficient variation, where Coef. 1 and Coef. 2 are associated with the two input gene-perceptrons respectively. Planes with $RSS > 0.7$ have Coef. 1 ranging from -0.2 to +0.4 and coef. 2 ranging from -1 to 1. Fig. 12b exemplifies the plane for the output *b3090* with the first and second coefficients of 0.10 and 0.14, respectively. Fig. 12c depicts the sub-GRNN where the input layer consists of *b3067* and *b3357* and output layer with *b3090* gene-perceptrons. Note that this example only considers the gene-perceptrons *b3067* and *b3357* as the inputs, and it is possible to explore diverse efficient spaces by selecting different inputs and output gene-perceptron combinations.

### *E. Multiple Polynomial Regression*

Multiple polynomial regressions are used for a number of applications, and one example is the estimation of the "Affective States" in humans [40]. Therefore, we evaluate the multiple polynomial regression dynamics using the same two inputs gene-perceptrons *b3067* and *b3357*. However, in this case, the goodness of the curve is fitted to the following equation,

$$f(x_1, x_2) = d_1x_1^2 + d_2x_2^2 + d_3x_1x_2 + d_4x_1 + d_5x_2 + d_6, \quad (13)$$

where $d_1$ to $d_6$ are coefficients that would be extracted for each output gene-perceptron and $x_1$ and $x_2$ are the inputs associated with the two genes *b3067* and *b3357*. Results in Fig. 13 depict the possibility of extracting complex higher-order multivariable polynomial regression models that matches to our problem. The $d_1$ and $d_2$ of Fig. 13a are quadratic coefficients associated with the two inputs that govern the curvature of the model. The positive values of $d_1$ and $d_2$ result in curvature along the *b3067* and *b3357* concentration axes, respectively. It is evident that $d_1$ and $d_2$ distributions exhibit distinct trends, characterized by positive and negative skewness, within approximate ranges of -10 to +10 and -3 to +3, respectively. Furthermore, $d_3$ is the cross-term coefficient that determines how the two inputs, *b3067* and *b3357* combine to affect the shape of the curve. Given that both inputs are consistently positive, the negative skewness of the $d_3$ distribution emphasizes that the majority of resulting curves are shifted upwards due to the combined influence of *b3067* and *b3357*. The linear terms, $d_4$ and $d_5$, have an impact on the curve's vertical position based on individual inputs. The analysis of the plots in Fig. 13a reveals that *b3067* exerts a balanced effect on shifting the curve, while *b3357* primarily tends to shift the curve downwards. Lastly, $d_6$ represents the y-intercept or the offset of the curve, determining the vertical positioning of the curve. Notably, the distribution of $d_6$ is fairly symmetrical around zero. Fig. 13b and Fig. 13c elucidate two significantly different multi-variable polynomial regression examples, using gene-perceptrons *b0904* and *b4406* and this is based on the sub-GRNN presented in Fig. 13c.

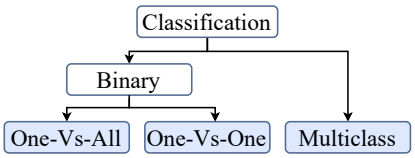



Fig. 14. Illustration of sub-categories of classification problems.

## V. GRNN APPLICATION IN CLASSIFICATION

Besides, regression application, NNs are also well-known for classification tasks [41], which is what we will analyze in this section.

### *A. BReLU as the activation function for GRNN-based classification*

As mentioned earlier, the gene-perceptrons exhibit BReLU activation function behaviors, and that can be beneficial in GRNN-based classification. The sigmoid function produces positive outputs for negative inputs and this can lead to noisy computing in NNs.However, this noisy behavior cannot be observed in the NNs when ReLU (BReLU) activation functions are utilized for classification tasks [42] as evident in Fig. 15a. ReLU including (BReLU) further encourages activation sparsity in computing by zeroing out negative values [43] and it further contributes to pruned networks for better computing efficiency, which will be discussed using Mutual Information later in this section.

Another advantage of BReLU is the increased sensitivity in classification [44]. As shown in Fig. 15b where the upper bound of the BReLU is equal to that in the sigmoid function, BReLU is more sensitive within the input range of zero to one compared to the sigmoid activation. While the variation of the sigmoid activation function is limited to 0.23 within the non-negative region ([0, +1]), the BReLU exhibits a variation of one.

The increased sensitivity in BReLU paves the path towards multi-class classification using single output gene-perceptron. This property is presented in Fig. 15b, where different values of thresholds ($Th_1, Th_2$ and $Th_3$ set to the 0.2, 0.5 and 0.8 values) are spaced to enable higher sensitivity. As the gene expression values are modeled by the BReLU activation function, the thresholds represent varying expression levels corresponding to the input. In the multi-class classification application, the number of thresholds is determined by the number of classes and can be expressed as $|Th| = |c| - 1$, where $|Th|$ and $|c|$ are the number of thresholds and classes, respectively. Due to this continuous output of the gene-perceptrons, two types of thresholds can further be applied, 1) $Th > 0$ for one class versus all the other classes (One-vs-All) and one class versus another class (One-vs-One) and 2) $Th = 0$ for One-Vs-All. However, the determination of the threshold in the case of $Th > 0$ requires additional processes as discussed in Section III-D and elucidated in Step 6 of Fig. 7.

Fig. 15c shows multiple expression levels corresponding to different classes ($C_0$, $C_1$, $C_2$ and $C_3$) where the expected

TABLE I
PARAMETERS UTILIZED FOR THE *in-silico* FEASIBILITY ANALYSIS

| Parameter | Value |
|---|---|
| Input gene | *b2664*, *b0080*, *b3060*, *b2697*, *b2220*, *b3423*, *b3743*, *b0345*, *b3481*, *b4401*, *b0357*, *b0889*, *b0817*, *b2217*, *b3905*, *b3071* |
| Input range | 0 to 0.5(normalized concentration units) |
| Iterations | 10 per each input |

expression level for the class $c_0$, $\overline{E}_{(c_0)}$ is always higher compared to other classes. Therefore, such an expression pattern of a gene-perceptron deems it suitable for One-vs-All classification output. However, determining the appropriate threshold ($Th_{(c_0)}$) necessitates additional steps, as discussed earlier. In addition, if a gene-perceptron expresses in two distinguish levels, $\overline{E}_{(c_0)}$ and $\overline{E}_{(c_1)}$ as depicted in Fig. 15c, it can be utilized for One-vs-One classification output with the threshold $Th_{(c_0,c_1)}$. Similarly, a gene-perceptron capable of expressing in three fixed levels, $\overline{E}_{(c_0)}$, $\overline{E}_{(c_1)}$ and $\overline{E}_{(c_2)}$ as shown in Fig. 15e in response to various inputs is suitable to represent the output layer node for multi-class classification applications. It is essential to emphasize that, in such situations, two thresholds ($Th_{(c_0,c_1)}$, $Th_{(c_1,c_2)}$) should be determined.

The feasibility of binary and multi-class classification is proven with *in-silico* experimental results in the next subsection.

### *B. Binary and Multi-Class Classification*

We discuss the feasibility of gene-perceptron performing a binary classification under two sub-categories, One-Vs-One (Fig. 15c) and One-Vs-All (Fig. 15d). Initially, we established a transcriptomic-level experimental setup using the *E. coli* GRNN, where we use 16 inputs that are represented by 16 randomly selected gene-perceptrons and more details on the selected input layer genes, the input range, and the number of simulation iterations are given in Table. I

For this specific network, the input layer of the GRNN is introduced with five different TF arrays (created as described in Section III-D) associated with five classes $c_i, i = \{0, 1, 2, 3, 4\}$. To capture the stochastic behavior within a cell, each simulation setup is iterated 10 times. Next, the expression levels of each gene-perceptron are recorded and filtered using the search algorithm proposed in Section III-D to identify the suitable gene-perceptron perform three sub-categories of classification, one-vs-all, one-vs-one and multi-class.

For the one-vs-all sub-category of binary classifications that employs the threshold as $Th = 0$, the algorithm seeks genes-perceptrons that express for one class with a minimized variance, while the expression levels for the other classes remain equal to zero. Fig. 16a shows results for one-vs-all classification for class $c_0$ where gene-perceptrons in the x-axis are an example set of suitable candidates that can be considered output nodes. Selecting one of the gene-perceptrons in the x-axis as the output node, it is possible to classify inputs into class $c_0$ proving the possibility of using the GRNN for one-vs-all classification tasks. Moreover, the one-vs-all method can be used for multi-class classification by selecting gene-perceprons that are suitable for other classes.

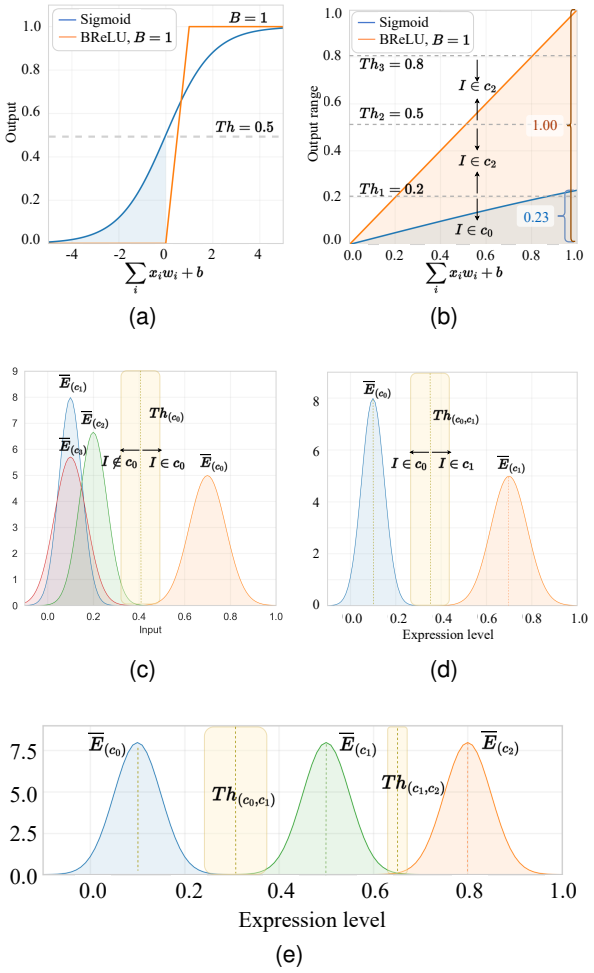



Fig. 15. Comparison of the sigmoid and inherent BReLU properties in the GRNN, where, a) highlights the shaded area in light blue that represents the region, where the sigmoid function outputs a positive value for negative weighted summation of the perceptron, b) compares the Sigmoid Vs BReLU output variation within the weighted summation range [0, 1] and the possibility of having multiple thresholds ($Th_1$, $Th_2$ and $Th_3$) leading to multi-class classifications ($c_0$, $c_1$, and $c_2$), c) example gene expression distribution for One-vs-All classification where $Th > 0$, d) gene expression distribution for One-Vs-One binary classification, and e) gene expression distribution with multi-class classification possibilities.

Fig. 16b shows the feasibility of using GRNN for one-vs-one classification, in which the objective is to search genes that have low variance in expression levels within the same class while showing higher variance between classes. Hence, these genes have different fixed expression levels for each class. This plot provides evidence for the existence of the gene-perceptrons that can be expressed in two distinct levels by representing two classes. The x-axis is sorted based on the mean expression distance between the two classes. The gene-perceptrons *b3403, b4063, b3250, b3251,* and *b4150* clearly differentiate between classes $c_0$ and $c_4$ with expression difference of approximately 0.1. Further, gene-perceptrons

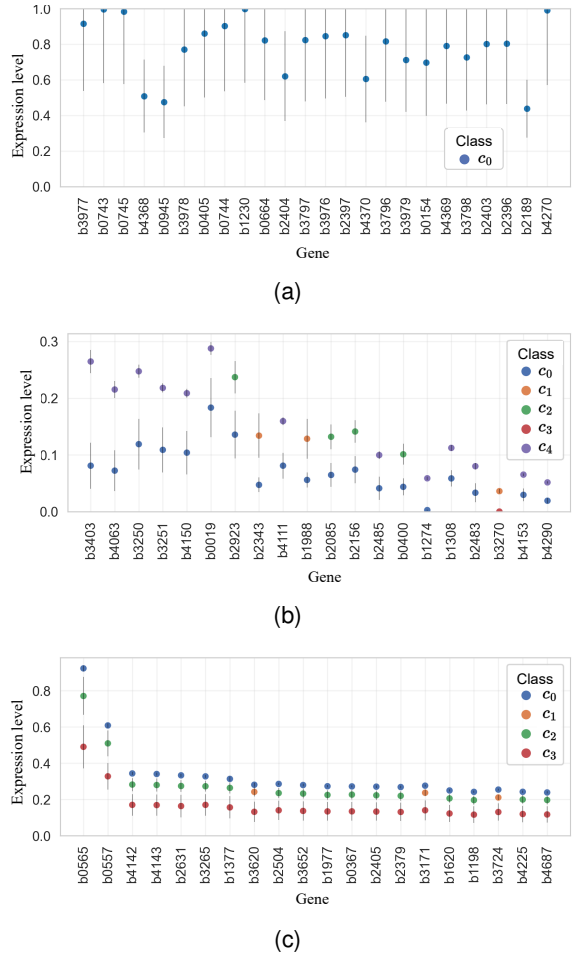



Fig. 16. Classification using the GRNN under three methods, where a) shows the One-vs-All, b) shows the One-vs-One, where the aim is to show different expression levels can be achieved for each gene indicating it can be used to classify more than one class and c) illustrates the multi-class classification, where we can see certain genes have separations that are high to support classification of up to three classes.

*b2923* and *b2343* can express in various levels to differentiate between classes $c_0$ and $c_2$, and $c_0$ and $c_1$, respectively

Finally, the possibility of using gene-perceptrons for multi-class classification is discussed here. As mentioned earlier, the extended output range due to BReLU allows multiple thresholds enabling multi-class classification. Fig. 16c presents a series of genes that can be used to classify three classes. The gene *b0565* has the largest distribution of mean expression levels associated with each class. Note that, except for the genes-perceptrons *b0565* and *b0557*, the deviation between multiple classes is low, making the differentiation between classes less feasible. However, it is important to mention that these results are extracted using one set of random inputs, and various input gene-perceptrons enable finding more output gene-perceptrons that will have sparse expression levels for multiple classes.

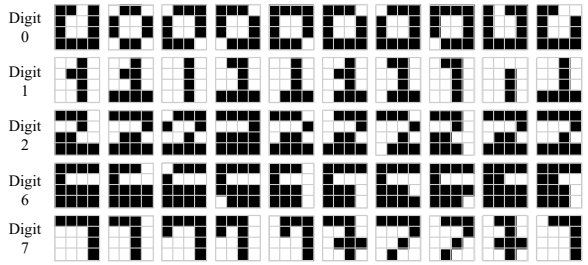



Fig. 17. $4 \times 4$ images under the five classes of digits and the associated augmentations.

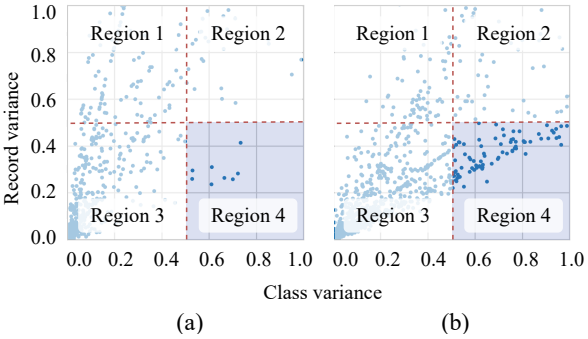



Fig. 18. Illustration of variance analysis results of two input permutations $J = 23$ and $J = 29$ in a) and b) respectively, where the x-axis is the expression variance between classes and the y-axis is the variance for different augmentations within the same class.

### C. Digit Classification Use-case

This study focuses on a classic use case of digit classification task using *E. coli* GRNN that are discovered through our search algorithm. The primary goal of the use case is to systematically analyzing each step of the application-specific sub-GRNN search algorithm and the accuracy of the computing. This section first discusses the experimental setup, the utilization of the proposed sub-GRNN search algorithm and finally the performance of the GRNN computing for digit classification.

The complexity of the problem is kept low in order to make the analysis more explainable. We only use $4 \times 4$ images with 16 pixels. We use a search dataset $SD$ with five classes of digits ("0", "1", "2", "6", and "7") and 10 augmentations with significant pattern differences as shown in Fig. 17. Consequently, this $SD$ has the dimensions of $50 \times 16$ with an accompanying $50 \times 1$ label matrix.

Following the proposed search algorithm, first, a pool of 128 ($P' = 128$) gene-perceptrons as suitable candidates for the input layer is selected based on the inward/outward edges degree distributions. As the images contain 16 ($K = 16$) pixels, the total number of input layer permutations is equal to ${}_{128}P_{16} \approx 1.95 \times 10^{33}$ (Fig. 7, Step 1). Next, the $SD$ is encoded into $\mathbf{I}^{(t=0)}$ as explained in Section III-D (Step 2) taking the binary properties of pixels into consideration.

We only use 150 ($J < 150$) permutations to find the most suitable sub-GRNN for our digit classification scenario.

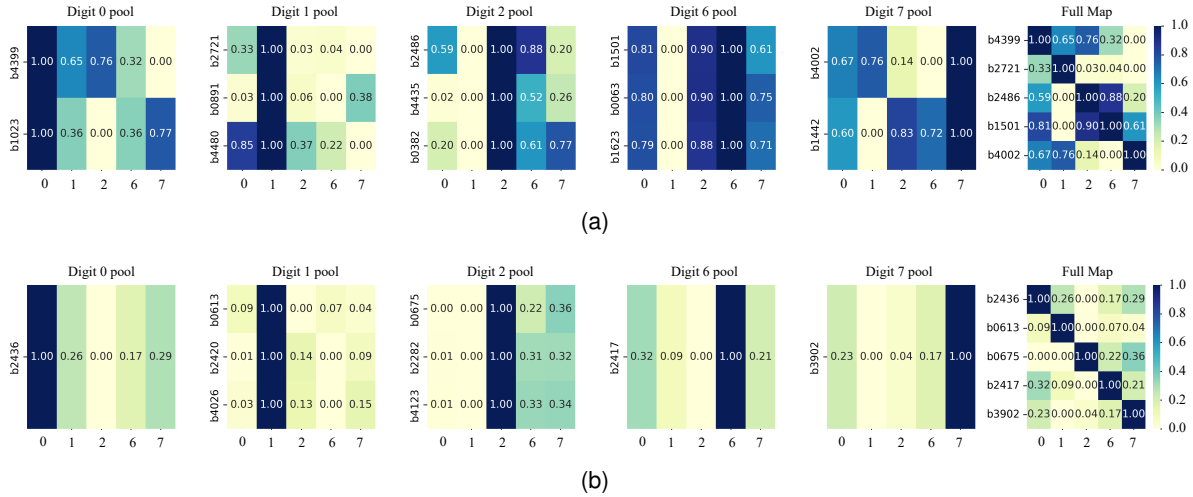



Fig. 19. Heatmaps of normalized gene-perceptron pools for each digit class, where a) shows the expression behaviors for the permutation, $J = 23$ , that is associated with Fig. 18a and b) represents the results for permutation, $J = 29$ that is associated with Fig. 18b.

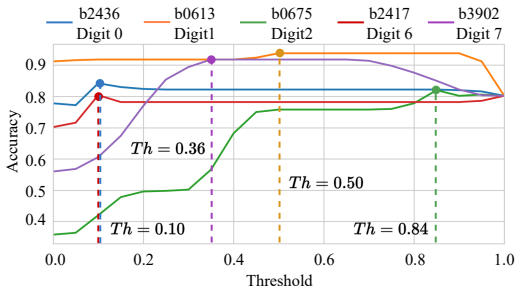



Fig. 20. Determining the thresholds for each class by maximizing the accuracy of each output gene-perceptron. These thresholds are then used to classify the expression levels of the corresponding output gene-perceptrons.

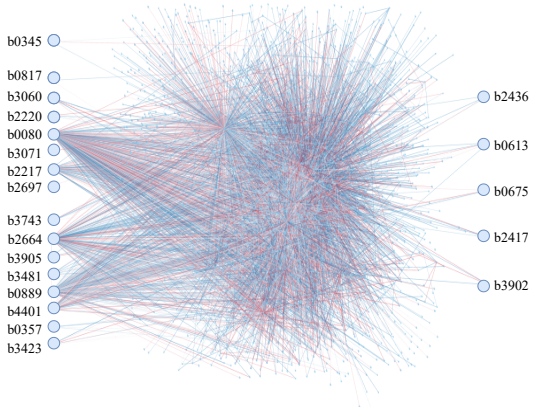



Fig. 21. Illustration of perturbation-based MI analysis on the input and output layer of the extracted sub-GRNN.

Using the GRNN computing model explained in Section III-B (Step 3), the output expression levels for all the genes for each of these input permutations are recorded after 43 times steps (where $T = 42$). As the next stage of the search algorithm, the variance analysis is conducted. Here we extract a pool of gene-perceptrons for the output layer (Fig. III-D, Step 4), with higher expression variation between classes and low expression variations for different augmentation within the same class that results in steady expression rates for the corresponding class. Variance analysis results for two input permutations with significantly high and low gene quantities in "Region 4" (higher class variance and lower record variance) are shown in Fig. 18 Fig. 18a shows the variance behavior of a random permutation $J = 23$ of inputs where a minimized number of candidates for output genes-perceptrons can be observed in "Region 4". In contrast, Fig. 18b associated with the permutation $J = 29$ has a significant number of candidates in "Region 4", maximizing the possibility of selecting the most accurate output gene-perceptrons.

After ranking the gene-perceptrons by the difference between the first and second highest mean expression values, according to Step 5 in Fig. 7, the output gene-perceptron are categorised under each digit as shown in Fig. 19. This categorization will allow us to determine which output gene corresponds to which digit. Evidently, Fig. 19a depicts a low variation between the expression levels for each digit class, that is resulted from the low number of gene-perceptron candidates in Fig. 18a for $J = 23$ permutation. However, contrasting results are observed for the expression patterns in Fig. 19b due to the evidently increased number of gene-perceptron candidates in "Region 4" from Fig. 18b for the $J = 29$ permutation. Hence, we select the input permutation $J = 29$ as the most suitable input layer for this particular problem as shown in Fig. 21.

The search algorithm can be employed to extract sub-GRNNs for one-vs-one, one-vs-all and multi-class classification. However, in this use case, we only show the method for one-vs-all classification where each class is assigned with

corresponding output gene-perceptrons. Based on the statistical distance relative gene expression levels corresponding to each digit class, the algorithm then selects five output gene-perceptrons (*b2436, b0613, b0675, b2417* and *b3902*) from each digit pool as shown in Fig. 19b. Subsequently, the algorithm searches for the appropriate expression thresholds (Step 6) which are selected based on the maximum classification accuracy. The classification accuracies for the selected five output gene-perceptron are calculated using (11) for a range of threshold (from zero to one with 0.05 increments) and consequent accuracy variation is shown in Fig. 20. As depicted in this figure, the accuracy maximization method determines 0.10, 0.50, 0.85, 0.1 and 0.35 as the thresholds that in turn result in accuracies of 0.842, 0.938, 0.820, 0.804 and 0.918 for digit classes 0, 1, 2, 6 and 7, respectively.

After successfully extracting an application-specific sub-GRNN, a perturbation-based MI analysis is performed with the objective of optimizing the network. Here, all the inputs of the extracted sub-GRNN are given input signals with fluctuation ranging from zero to one and the outputs are recorded from the five genes-perceptrons *b2436, b0613, b0675, b2417* and *b3902*. Since the inputs and the outputs for this network are continuous variables, the Mutual Information (MI) between the input and output nodes is denoted as,

$$I(g_x; g_y) = f(x,y) \cdot \log\left(\frac{f(x,y)}{f(x)\cdot f(y)}\right) dxdy, \qquad (14)$$

where $g_x$ and $g_y$ are the input and output nodes, respectively. Further $f(x,y)$ is the joint probability density function of $g_x$ and $g_y$ expressions.

The results of this MI analysis are presented in Fig. 22 distinctively proving that only the input gene-perceptrons, *b0080*, *b4401*, *b0889*, *b2217* and *b3905* contribute to the decision in the output layer gene-perceptrons. It is understood that the shutting down of gene expression pathways as a result of BReLU causes information flow disconnection between the input and output layer nodes. This disconnection is evident in Fig. 22, where mutual information (MI) becomes zero. We then extract an optimized network based on these results by reducing the input layer to only have five gene-perceptrons that are mentioned above.

Finally, we compare the accuracy of decision-making of the extracted sub-GRNN, before and after condensing the network based on minimizing the number of inputs, where the results are shown in Fig. 23. These results suggest that the optimized sub-GRNN can make decisions close to the previous version of the network, despite the reduced structural complexity due to a low number of input nodes that are stimulated. This complexity reduction can further result in lowering ATP energy to fuel the gene-perceptrons for computing [45], lower the amount of noise to maximize reliability, as well as improve explainability and reproducibility.

## VI. Discussion

GRNNs introduced in [22], represent a distinctive form of neural networks naturally embedded within GRNs. These networks can be conceptualized as extensive repositories of

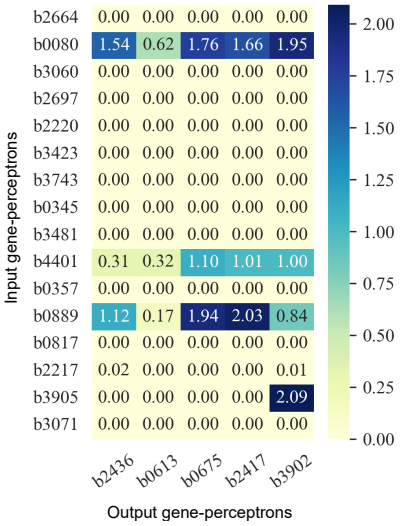



Fig. 22. Illustration of perturbation-based MI analysis on the input and output layer of the extracted sub-GRNN.

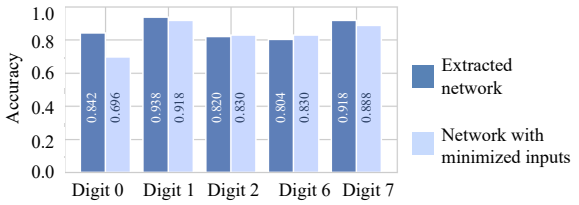



Fig. 23. A comparison of the accuracy of the extracted sub-GRNN before and after minimizing the number of inputs, where the darker columns represent the accuracies of each class before and the lighter columns represent the accuracies of the network after minimizing the number of inputs.

pre-trained NNs at the biological hardware-layer capable of executing intricate and diverse computing tasks. Hence, GRNNs can be regarded as a wet-neuromorphic systems. However, harnessing the potential of GRNNs for computing demands a specialized set of mechanisms, including GRNN extraction and an algorithm for searching application-specific sub-GRNNs. Therefore, this study improved the GRNN extraction method in [22] and introduced a random permutation-based application-specific sub-GRNN search algorithm. Considering *E. coli* as the model species, we extracted the base-GRNN and proved its accuracy indicating the reliability of the proposed extraction method, innate computing, and the possibility of converging the multi-dimensional gene-gene interaction to a single weight. Subsequently, a feasibility analysis on the extracted *E. coli* GRNN proved its computing capability in classification and regression problems.

The feasibility analyses exhibit the computing power embedded in a single cell and the possibility of mapping sub-GRNNs for wide range of applications. The classification and regression results highlight two noteworthy attributes of GRNN-based computing: analog and parallel computing

capabilities. Both analyses used continuous inputs, for one-vs-all and one-vs-one classifications revealing the potential for analog to digital computing. Furthermore, in the multi-class classification, it is evident that the GRNN can be utilized for analog to multi-level computing. In addition, we conducted all the analyses on three datasets generated for classification, simple regression, and multiple regression. All three sub-analyses on classification, three analyses on simple regression and two analyses on multiple regression are done parallelly on the corresponding dataset. This emphasizes the possibility of using GRNNs for parallel computing which can be significantly efficient.

Here, we like to highlight prospective research areas associated with the concept of GRNN as it is still in its infancy. Integrating reporter genes as the output layer of application-specific sub-GRNN for conveniently observable outputs can be one of the promising research. Similarly, we believe that exploring the possibility of utilizing synthetic proteins as inputs can also be one of the avenue for further research. Moreover, embedding the metabolomic layer in the GRNN can lead to incorporating molecular inputs, enhancing both convenience and practicality in this domain.

This study indicates that in the future, GRNN-based bio-computing can be an alternative to silicon-based computing. Moreover, this study proposes an application-specific sub-GRNN search algorithm that will find the most suitable candidate for the targeted problem.

## REFERENCES

[1] J. L. Poet, A. M. Campbell, T. T. Eckdahl, and L. J. Heyer, "Bacterial computing," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 1, pp. 10–15, 2010.

[2] R. Lahoz-Beltra, J. Navarro, and P. C. Marijuán, "Bacterial computing: a form of natural computing and its applications," *Frontiers in Microbiology*, vol. 5, p. 101, 2014.

[3] M. D. Carroll R, "Twenty years of amos research," *Currents in Biblical Research*, vol. 18, no. 1, pp. 32–58, 2019.

[4] M. Kuscu, E. Dinc, B. A. Bilgin, H. Ramezani, and O. B. Akan, "Transmitter and receiver architectures for molecular communications: A survey on physical design with modulation, coding, and detection techniques," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1302–1341, 2019.

[5] A. Lombardo, G. Morabito, C. Panarello, and F. Pappalardo, "Modeling biological receivers: the case of extracellular vesicle fusion to the plasma membrane of the target cell," in *Proceedings of the 9th ACM International Conference on Nanoscale Computing and Communication*, pp. 1–6, 2022.

[6] L. Ceze, J. Nivala, and K. Strauss, "Molecular digital data storage using dna," *Nature Reviews Genetics*, vol. 20, no. 8, pp. 456–466, 2019.

[7] L. Rizik, L. Danial, M. Habib, R. Weiss, and R. Daniel, "Synthetic neuromorphic computing in living cells," *Nature communications*, vol. 13, no. 1, p. 5602, 2022.

[8] L. Smirnova, B. S. Caffo, D. H. Gracias, Q. Huang, I. E. Morales Pantoja, B. Tang, D. J. Zack, C. A. Berlinicke, J. L. Boyd, T. D. Harris, *et al.*, "Organoid intelligence (oi): the new frontier in biocomputing and intelligence-in-a-dish," *Frontiers in Science*, p. 0, 2023.

[9] B. J. Kagan, A. C. Kitchen, N. T. Tran, F. Habibollahi, M. Khajehnejad, B. J. Parker, A. Bhat, B. Rollo, A. Razi, and K. J. Friston, "In vitro neurons learn and exhibit sentience when embodied in a simulated game-world," *Neuron*, vol. 110, no. 23, pp. 3952–3969, 2022.

[10] S. Balasubramaniam, S. Somathilaka, S. Sun, A. Ratwatte, and M. Pierobon, "Realizing molecular machine learning through communications for biological ai," *IEEE Nanotechnology Magazine*, vol. 17, no. 3, pp. 10–20, 2023.

[11] S. Angerbauer, F. Enzenhofer, T. Pankratz, M. Hamidović, A. Springer, and W. Haselmayr, "Novel nano-scale computing unit for the iobnt: Concept and practical considerations," *TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.24167163.v1*, Sept. 2023.

[12] O. B. Akan and S. Balasubramaniam, "Body area nanonetworks with molecular communications in nanomedicine," *IEEE Communications Magazine*, vol. 50, no. 1, pp. 28–34, 2012.

[13] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[14] A. Levskaya, A. A. Chevalier, J. J. Tabor, Z. B. Simpson, L. A. Lavery, M. Levy, E. A. Davidson, A. Scouras, A. D. Ellington, E. M. Marcotte, *et al.*, "Engineering escherichia coli to see light," *Nature*, vol. 438, no. 7067, pp. 441–442, 2005.

[15] J. Baumgardner, K. Acker, O. Adefuye, S. T. Crowley, W. DeLoache, J. O. Dickson, L. Heard, A. T. Martens, N. Morton, M. Ritter, *et al.*, "Solving a hamiltonian path problem with a bacterial computer," *Journal of biological engineering*, vol. 3, pp. 1–11, 2009.

[16] L. J. Heyer, J. L. Poet, M. L. Broderick, P. E. Compeau, J. O. Dickson, and W. L. Harden, "Bacterial computing: using e. coli to solve the burnt pancake problem," *Math Horizons*, vol. 17, no. 3, pp. 5–10, 2010.

[17] A. Pandi, M. Koch, P. L. Voyvodic, P. Soudier, J. Bonnet, M. Kushwaha, and J.-L. Faulon, "Metabolic perceptrons for neural computing in biological systems," *Nature communications*, vol. 10, no. 1, p. 3880, 2019.

[18] X. Li, L. Rizik, V. Kravchik, M. Khoury, N. Korin, and R. Daniel, "Synthetic neural-like computing in microbial consortia for pattern recognition," *Nature communications*, vol. 12, no. 1, p. 3139, 2021.

[19] K. E. Duncker, Z. A. Holmes, and L. You, "Engineered microbial consortia: strategies and applications," *Microbial Cell Factories*, vol. 20, no. 1, pp. 1–13, 2021.

[20] W. Jiang, X. He, Y. Luo, Y. Mu, F. Gu, Q. Liang, and Q. Qi, "Two completely orthogonal quorum sensing systems with self-produced autoinducers enable automatic delayed cascade control," *ACS Synthetic Biology*, vol. 9, no. 9, pp. 2588–2599, 2020.

[21] R. Zhang, H. Goetz, J. Melendez-Alvarez, J. Li, T. Ding, X. Wang, and X.-J. Tian, "Winner-takes-all resource competition redirects cascading cell fate transitions," *Nature communications*, vol. 12, no. 1, p. 853, 2021.

[22] S. S. Somathilaka, S. Balasubramaniam, D. P. Martins, and X. Li, "Revealing gene regulation-based neural network computing in bacteria," *Biophysical Reports*, vol. 3, no. 3, 2023.

[23] S. Cussat-Blanc, K. Harrington, and W. Banzhaf, "Artificial gene regulatory networks—a review," *Artificial life*, vol. 24, no. 4, pp. 296–328, 2019.

[24] J. VOHRADSKY, "Neural network model of gene expression," *the FASEB journal*, vol. 15, no. 3, pp. 846–854, 2001.

[25] C. Scholes, A. H. DePace, and Á. Sánchez, "Combinatorial gene regulation through kinetic control of the transcription cycle," *Cell systems*, vol. 4, no. 1, pp. 97–108, 2017.

[26] A. Ishihama, "Prokaryotic genome regulation: a revolutionary paradigm," *Proceedings of the Japan Academy, Series B*, vol. 88, no. 9, pp. 485–508, 2012.

[27] M. V. Grosso-Becerra, G. Croda-García, E. Merino, L. Servín-González, R. Mojica-Espinosa, and G. Soberón-Chávez, "Regulation of pseudomonas aeruginosa virulence factors by two novel rna thermometers," *Proceedings of the National Academy of Sciences*, vol. 111, no. 43, pp. 15562–15567, 2014.

[28] Z. Koşar and A. Erbaş, "Can the concentration of a transcription factor affect gene expression?," *Frontiers in Soft Matter*, vol. 2, p. 914494, 2022.

[29] A. Ishihama, "Prokaryotic genome regulation: multifactor promoters, multitarget regulators and hierarchic networks," *FEMS microbiology reviews*, vol. 34, no. 5, pp. 628–645, 2010.

[30] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko, *et al.*, "Ncbi geo: archive for functional genomics data sets—update," *Nucleic acids research*, vol. 41, no. D1, pp. D991–D995, 2012.

[31] K. Struhl, "Fundamentally different logic of gene regulation in eukaryotes and prokaryotes," *Cell*, vol. 98, no. 1, pp. 1–4, 1999.

[32] X. Huang, C. Song, G. Zhang, Y. Li, Y. Zhao, Q. Zhang, Y. Zhang, S. Fan, J. Zhao, L. Xie, *et al.*, "scgrn: a comprehensive single-cell gene regulatory network platform of human and mouse," *Nucleic Acids Research*, p. gkad885, 2023.

[33] F. Fioravanti, M. Helmer-Citterich, and E. Nardelli, "Modeling gene regulatory network motifs using statecharts," *BMC bioinformatics*, vol. 13, pp. 1–12, 2012.

[34] G. Xue, X. Zhang, W. Li, L. Zhang, Z. Zhang, X. Zhou, D. Zhang, L. Zhang, and Z. Li, "A logic-incorporated gene regulatory network deciphers principles in cell fate decisions," *bioRxiv*, pp. 2023–04, 2023.
[35] V. H. Tierrafría, C. Rioualen, H. Salgado, P. Lara, S. Gama-Castro, P. Lally, L. Gómez-Romero, P. Peña-Loredo, A. G. López-Almazo, G. Alarcón-Carranza, *et al.*, "Regulondb 11.0: Comprehensive high-throughput datasets on transcriptional regulation in escherichia coli k-12," *Microbial Genomics*, vol. 8, no. 5, 2022.
[36] M. Morzy, T. Kajdanowicz, P. Kazienko, *et al.*, "On measuring the complexity of networks: Kolmogorov complexity versus entropy," *Complexity*, vol. 2017, 2017.
[37] H. Zenil, S. Hernández-Orozco, N. A. Kiani, F. Soler-Toscano, A. Rueda-Toicen, and J. Tegnér, "A decomposition method for global evaluation of shannon entropy and local estimations of algorithmic complexity," *Entropy*, vol. 20, no. 8, p. 605, 2018.
[38] C.-E. Yin and G. Qu, "Obtaining statistically random information from silicon physical unclonable functions," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 96–106, 2014.
[39] J. Rojo, L. G. de Pinho, C. Fonseca, M. J. Lopes, S. Helal, J. Hernández, J. Garcia-Alonso, and J. M. Murillo, "Analyzing the performance of feature selection on regression problems: A case study on older adults' functional profile," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 1, pp. 137–152, 2023.
[40] J. Wei, T. Chen, G. Liu, and J. Yang, "Higher-order multivariable polynomial regression to estimate human affective states," *Scientific reports*, vol. 6, no. 1, p. 23384, 2016.
[41] T. Wei, W.-L. Liu, J. Zhong, and Y.-J. Gong, "Multiclass classification on high dimension and low sample size data using genetic programming," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 704–718, 2022.
[42] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.
[43] X. Zhou, Z. Du, S. Zhang, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen, "Addressing sparsity in deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1858–1871, 2019.
[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[45] Q. Qi, Y. Lu, J. Li, J. Wang, H. Sun, and J. Liao, "Learning low resource consumption cnn through pruning and quantization," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 886–903, 2022.

**Samitha Somathilaka** (Student Member, IEEE) received the Bachelor's degree in science in 2013 from University of Ruhuna, Sri Lanka and the MSc specialized in Cyber security in 2017 from Sri Lanka Institute of Information Technology, Sri Lanka. He is currently working toward the PhD degree with the Department of Computing and Mathematics, Walton Institute, South East Technical University, Waterford, Ireland. He is currently a visiting research scholar with the School of Computing, University of Nebraska-Lincoln, Lincoln, NE, USA. His research interests include biocomputing, bacterial computing, molecular communications, gene regulatory neural networks, and computational modeling.

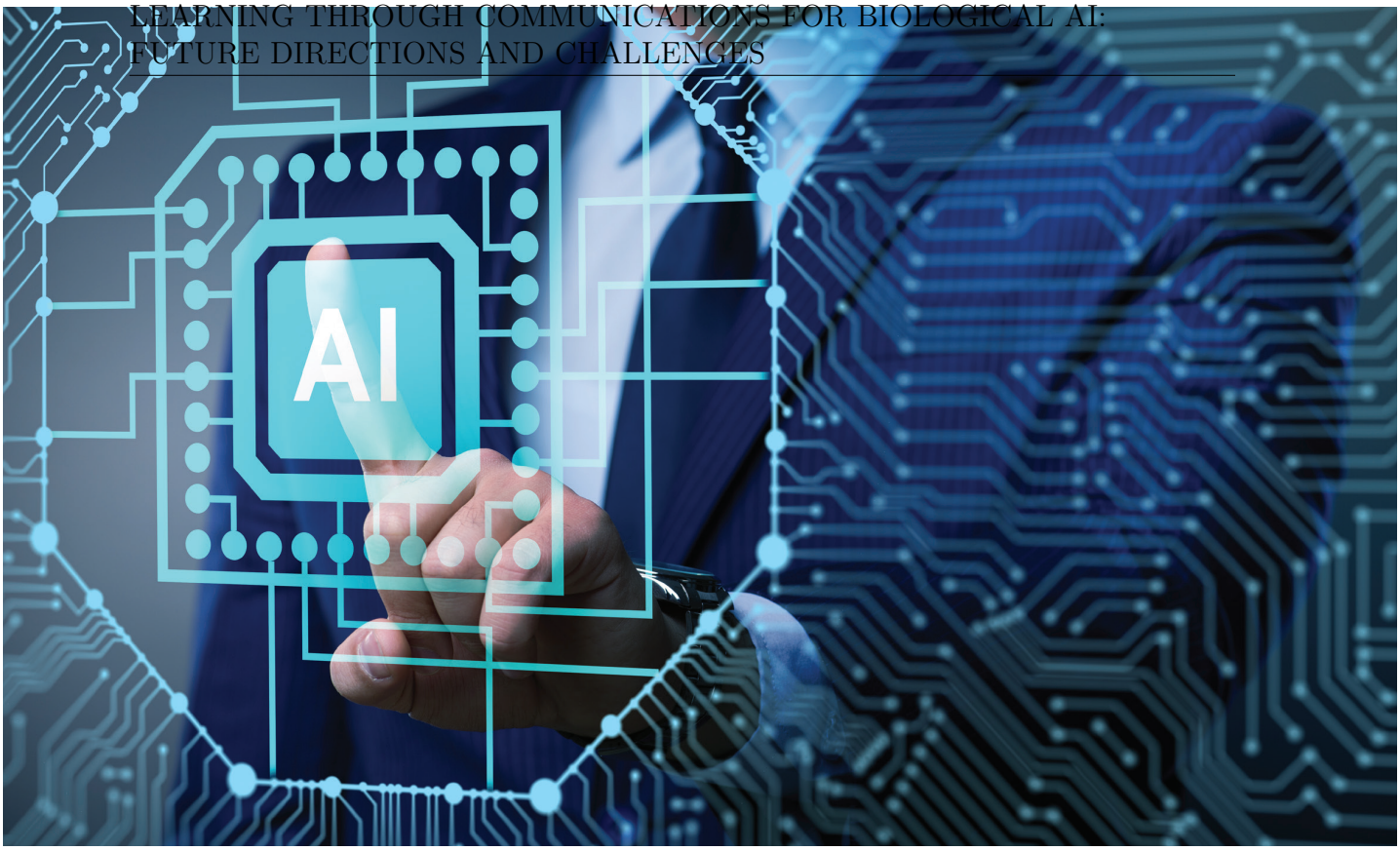
**Sasitharan Balasubramaniam** (Senior Member, IEEE) received the Bachelor's degree in engineering and Ph.D. degree from University of Queensland, Brisbane, Australia, in 1998 and 2005, respectively, and the Masters of engineering science from Queensland University of Technology,Brisbane, in 1999. He was a recipient of Science Foundation Ireland Starter Investigator Research Grant. He was also a recipient of the Academy of Finland Research Fellow with Tampere University, Finland. He was the Director of Research at the Walton Institute, South East Technological University, Ireland. He is currently an Associate Professor with the School of Computing, University of Nebraska-Lincoln, Lincoln, NE, USA. His research interests include molecular/nano communications, Internet of Bio-Nano Things, and 5G/6G networks. He is currently the Editor-in-Chief of IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL AND MULTI-SCALE COMMUNICATIONS and an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING. He was an IEEE Distinguished Lecturer for the IEEE Nanotechnology Council in 2018.

**Daniel P. Martins** (Member, IEEE) received his PhD from Waterford Institute of Technology (2019), his MSc in Electrical Engineering, from Federal University of Campina Grande, Brazil (2014), and his BSc in Telecommunications Engineering, from UNIJORGE, Brazil (2008). He is an IEEE Member and volunteer, since 2005, contributing with the development of student and professional activities in South America, Ireland, and UK. He was a Postdoctoral Researcher at Walton Institute, South East Technological University, Ireland, working on the development of bacteria-based communications and computing systems for the Irish dairy industry, as part of VistaMilk (a SFI research center). Currently, Daniel is a Lecturer with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK. His research interest includes Bacteria-based Computing and Communications Systems, Cyberbiosecurity, Signal Processing and Molecular Communications.

# Chapter 10

# Journal: Realizing Molecular Machine Learning through Communications for Biological AI: Future Directions and Challenges

| | |
|---|---|
| **Journal Title:** | IEEE Nanotechnology Magazine |
| **Article Type:** | Magazine Paper |
| **Complete Author List:** | Sasitharan Balasubramaniam, Samitha S. Somathilaka, Sehee Sun, Adrian Ratwatte and Massimiliano Pierobon |
| **Status:** | Published. April 2023. 10.1109/MNANO.2023.3262099 |






# Realizing Molecular Machine Learning Through Communications for Biological AI

## Future Directions and Challenges

ARTIFICIAL INTELLIGENCE (AI) and Machine Learning (ML) are weaving their way into the fabric of society, where they are playing a crucial role in numerous facets of our lives. As we witness the increased deployment of AI and ML in various types of devices, we benefit from their use into energy-efficient algorithms for low powered devices. In this paper, we investigate a scale and medium that is far smaller than conventional devices as we move towards molecular systems that can be utilized to perform machine learning functions, i.e., Molecular Machine Learning (MML). Fundamental to the operation of MML is the transport, processing, and interpretation of information

*Digital Object Identifier 10.1109/MNANO.2023.3262099*
*Date of current version: 31 May 2023*

SASITHARAN BALASUBRAMANIAM, SAMITHA SOMATHILAKA, SEHEE SUN, ADRIAN RATWATTE, AND MASSIMILIANO PIEROBON




1932-4510/12/23©2023IEEE

propagated by molecules through chemical reactions. We begin by reviewing the current approaches that have been developed for MML, before we move towards potential new directions that rely on gene regulatory networks inside biological organisms, as well as their population interactions to create neural networks. We then investigate mechanisms for training machine learning structures in biological cells based on calcium signaling and demonstrate their application to build an Analog to Digital Converter (ADC). Lastly, we look at potential future directions, as well as challenges that this area could solve.

## INTRODUCTION

In resent years, we have started to witness the widespread development of systems to apply Artificial Intelligence (AI) and Machine Learning (ML) to very diverse application scenarios [1]. This has resulted in software-based systems for AI, such as Artificial Neural Networks (ANN) [2], as well as hardware based systems like neuromorphic hardware [3]. In particular, within the area of ANN, various algorithms have been developed, that includes Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), amongst others, where each has its own properties and behaviour derived from specific functions of neuronal networks of the brain. While developments have been made in AI for both hardware and software, there is still a number of challenges that exists. These challenges include the ability to mimic the behavior and realism of neurons and their internal functionalities, as well as matching their energy requirements. The former challenge is still today a major issue that continues to motivate research to ensure that new algorithms or hardware designs will resemble the properties of internal neuronal signaling (e.g., ion transfer, action potential generation and propagation). However, the more realistic we design AI algorithms to closely resemble neuronal cells, the higher the energy consumption since we are mimicking the chemical and molecular reactions that occurs internally. When making this comparison, the brain consumes approximately 20 W for 100 billion neurons and 1,000 trillion synapses compared to a neuromorphic processor such as the Neurogrid with 65 thousand neurons and 500 M synapses, which consumes 3.1W [4]. In order to minimize energy consumptions, alternative materials have also been proposed for artificial neural systems and one example is the use of spintronics [5].

A number of alternative solutions have also been proposed to mimic natural neuron functions, where biological neuronal cells have been used to perform AI computing to replace conventional computing systems, i.e., biological AI. Examples of this include living neurons that can play pong [6], robots integrated with neuronal cells to control their operation [7], control of a robotic arm [8], and Organoid Intelligence Bio-computing [9]. This approach has also shown that the neurons can also be taught and trained to adapt to specific applications. Besides neurons, other forms for biological systems have also been considered to perform computing functions. Examples include the use of *Physarum* to solve networking problems at the Tokyo railway network [10], and, most recently, the use of fungii to perform molecular computing [11]. Using these approaches can possibly result in new solutions where biological cells work in tandem with silicon technologies, i.e., bio-hybrid AI. While this may address the aforementioned challenges of including more realistic biological properties, protocols and technologies to maintain biological cell lines and keeping them alive for a long period may also invalidate the quest for higher efficiency of these systems.

Fundamental to all biological AI solutions and models that have been proposed is the exchange of molecules between cells to realize computing functions. This communication based on molecules occurs as both an intra, as well as inter-cellular signaling. However, the training and computing processes within these systems can be further enhanced through modeling, optimization, and engineering of these same processes, with the help of molecular communication theory. As this field is slowly maturing, models and systems have been developed to study and engineer information encoding into molecules to be exchanged between different biological or bio-hybrid entities, also called bio-nanomachines, such as the aforementioned AI-enabling cells. Examples include characterizations of channels within biological environments [15], [16], [17], [18] and molecular modulation techniques (e.g., MoSK [19]). These new communication models have been applied to characterize and engineer numerous types of molecular communication systems such as neuronal interconnections [20], multi-hop diffusion-based networks [21], and large scale systems with 3D geometry [22]. Test beds and proofs-of-concept have also been developed, including table top molecular communication systems [23], as well as molecular modulators that transmit digital information between computers [24]. The engineering of molecular communication systems in biological or bio-hybrid AI systems can enable new design, as well as efficiency and robustness. This may include the design of engineered molecules to propagate information during gene expression leading to intra-cellular signaling, as well as inter-cellular signaling that can support ANN functionalities between populations of cells. This can be achieve through the combination of molecular communication theory and the tools provided by synthetic biology, where genetic circuits are engineered to produce molecular signals communicated between cells.

In this article, we will analyze a number of different biological AI and the types of communication that is inherent in the models, i.e., Molecular Machine Learning (MML). MML in here intended as machine learning realized with molecules and chemical reactions as building blocks, rather than computer programs to inform synthetic chemistry, as in [25]. This includes engineered cells to create perceptrons found in ANN or interconnecting engineered cells to behave as neural networks. We will then follow with alternative future directions for developing ANN using the concepts of molecular communication theory through the natural Gene Regulatory Networks (GRN), molecular communication between multi-species population

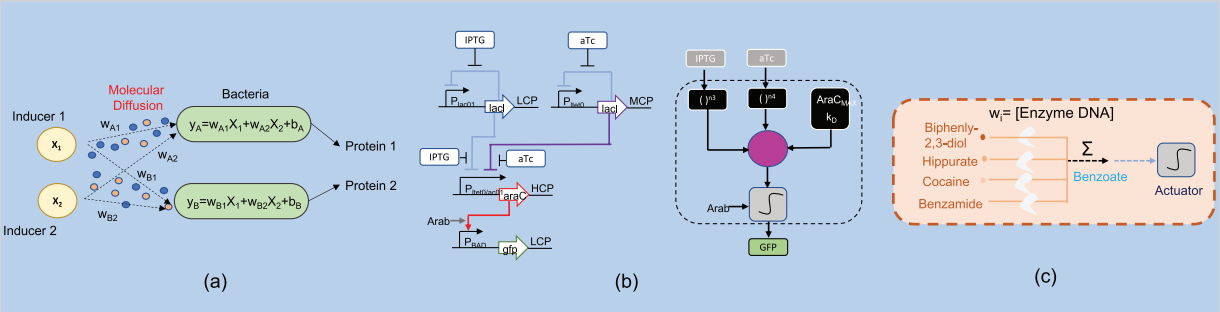



**FIGURE 1** Proposed solutions to develop neural networks from engineering cells. (a) Bactoneuron [12], (b) Perceptgene [13], and (c) metabolic perceptron [14].

of cells, as well as engineering of $Ca^{2+}$ signaling based molecular communications to create an Analog-to-Digital Converter (ADC). Lastly we will focus on future challenges for MML.

This article is organized as follows. Section 2 discusses current background on engineered cells as well as metabolic reaction models to realize ANN. In Section 3, we propose a new direction whereby natural GRNs and their embedded intracellular molecular communication for AI. In Section 4, we introduce an idea for utilizing a multi-species cellular consortia to perform AI using inter-cellular molecular communication. In Section 5, we move towards engineering calcium ($Ca^{2+}$) signaling in cells to achieve perceptron like behavior. In Section 6, we discuss future directions and challenges, while in Section 7, we conclude the paper.

## CURRENT BACKGROUND ON BIOLOGICAL AI

Numerous research has indicated natural intelligence that occurs within cells. From the perspective of molecular communications, this deals with initially sensing molecular signals from the environment, followed by internal signal transduction that leads to gene expressions, as well as corresponding metabolic pathways. This process is largely programmed into the cell's genome [26]. In certain cases, this intelligence and memory management can be performed with organisms that lack a brain, or non-neuronal systems as pointed out in [27]. In the case of bacteria, claims have been made the microbes contain 'minimal cognition' [28].

In [12], a single layer ANN was developed using engineered *E.Coli*, known as Bactoneuron (Figure 1(a)). The developed model is able to achieve both reversible as well as irreversible computing. Each cell is engineered to receive inter-cellular diffusing molecules, and as a response, execute a log-sigmoid activation function to produce Green Fluorescent Protein (GFP) output. This execution is established through a transcriptional regulation which is undertaken by an engineered genetic circuit (also referred to as *cellular device*). The solution proposed uses established set of general rules to map the complete ANN architecture and to derive unit bactoneurons directly from the functional truth table of a complex computing function. The study produced both simulations, as well as experimental validation. Example applications included a 2-to-4 decoder, a 4-to-2-priority encoder, a majority function, a 1-to-2 de-multiplexer, and a 2-to-1 multiplexer and reversible logic mapping through Feynman and Fredkin gates. Rizik et al. [13] developed the Perceptgene (Figure 1(b)), which is a perceptron model of an ANN. This was achieved through the genetic circuit engineering in *E. Coli* bacteria. The perceptron behavior is established through a logarithmic input-output relationship that fits to the non-linear biochemical reactions that occur in the genetic circuits. The implementation is based on engineered genetic circuits whose input-output behavior includes both the power-law, as well as a multiplication function. The power-law function encodes the weighted chemical inputs, while the multiplication function aggregates all the inputs that will determine the activation. The weight of each input is determined by the Hill coefficient. The two inputs used are *isopropyl Beta-D-1-thiogalactopyranoside* (*IPTG*) and *anhydrotetracycline* (*aTc*) molecular signals and results in a repression process that in turn regulates their own production using an auto-negative feedback loop. Similar to the perceptrons of an ANN, the perceptgene also contains a bias component for the sigmoidal activation function. The bias input is set by the ratio of the maximum transcription process to the binding affinities of the protein-protein/protein-DNA reactions. The applications of the perceptgene include weighted multi-input functions, classification, as well as an offline gradient descent learning algorithms. In [28], an offline trained perceptron neural network is used to program a population of bacteria, and it is simulated *in silico*. Through the diffusion of inter-cellular molecular communication within a population, the cells were able to have social interactions and form complex communities. The programmed perceptron was also used to solve an optimization problem. The work was based on an in-silico model, where the plasmid encoded perceptron was designed using *Cello*, while the simulation of the bacterial communication was developed through the *Gro* simulation tool. A particular aspect of the study is the use of programmed ANN into the genetic circuit to control signaling between cells in the population to perform functions. The input are natural molecules (e.g., galactose), which in turn control a downstream behavior. This includes (i) emitting molecular

signals proportional to the concentration of oxygen that is used for metabolic purposes, (ii) inducing chemotaxis for cell movement, (iii) commensalism, where the cells emit a signal that degrades the waste products from other bacteria in the population, and (iv) controlling of cell growth when the environment is harsh.

In [29], a consortia-based bacterial ANN was developed and proved experimentally. An interesting feedback process is developed between the receiver and the sender, which are the perceptron nodes for decision making and this is achieved using quorum sensing. The sender bacteria are able to emit varying molecular signals (*OHC14 - acyl-homoserine lactone 3OHC14:1-HSL*), which represent the weights. These molecular signals are induced by an external signal (*OC6* (*acyl-homoserine lactone 3OC6-HSL*)). The application was specific to 4-bit pattern recognition, where varying levels of the *OC6* inducers are applied to sender bacterial populations, and once the molecular signals diffuse to the receiver, they will activate a genetic circuit to produce an output signal. A novel gradient descent algorithm was also developed to optimize the weights of molecular signals to suit the pattern recognition application.

A cell-free perceptron model was proposed in [14] using the metabolic circuit illustrated in Figure 1(c). The latter was designed with a focus on biochemical retrosynthesis to predict the pathways, which was achieved using the *Retropath* and *Sensipath* computational design tools. The circuit was then embedded into a cell-free system in order to create the Metabolic Perceptron. The metabolic perceptron was able to perform binary classification based on metabolite molecular signals that leads to a classification process. The example application was here a four-input binary classifier.

## GENETIC REGULATORY AI

While the previous section focused on the genetic engineering of living cells to create machine learning systems, in this section, we will look at an alternative approach that is based on computing structures naturally present in biological cells, i.e., GRNs. This approach is based on essential similarities between a GRN and its structure to an ANN. While a number of different works have investigated neural-like properties in GRNs, our investigation focuses on how molecular communication properties can be exploited to perform computing functions as well as training by externally manipulating the weight connections between gene relationships.

### BACKGROUND ON GENE REGULATORY NETWORKS

A GRN is a highly complex network of multi-layered interactions between genes. Each individual cell carries a GRN specific to its species and strain, giving an unique behavioral pattern, as well as functionalities. A cell can sense a range of external stimuli using membrane receptors, perform computing through the GRN and express genes accordingly, thus resembling an input-process-output sequence found in conventional computing. A typical process of gene expression starts with the transcription process of converting the genes into mRNA, and this, depending on the gene, can be followed by the translation process that coverts the information contained in the mRNA into proteins. However, during gene expression within the GRN, molecular communication patterns can be identified in gene-gene interactions, which are complex processes that occur at multiple layers. For example, while these interactions in prokaryotes contribute to the regulation of the aforementioned transcription process, for eukaryotes, they can be post-transcriptional, i.e., contributing to, among other things, mRNA (or other transcript) and/or protein functionalities.

Moreover, the regulation in the post-transcription layer contributes to specific dynamics in the behavior of GNRs. In this context, proteins plays a crucial role complementing the regulation mechanism by integrating sensing, transfer, storage, and processing of information. As an example, proteins can perform computational tasks such as amplification, Boolean logic functions, and information storage through mechanisms of allosteric regulation [30]. In addition, the inter-conversions between phosphorylated and non-phosphorylated states of proteins act as switches enabling them to exhibit sigmoidal behaviours over a limited concentration range.

In the following, we show how these complex molecular signaling processes that involve multiple layers of chemical reactions, as well as components during gene expressions, combined with the network structure of genome relationships, can allow us to identify and exploit natural ANN within GRNs, i.e., Genetic Regulatory AI (GRAI).

### ANN LEARNING AND TRAINING MODELS IN A SIMPLE GENE REGULATORY NETWORK

The transcription of a particular gene in a GRN is combinatorial action of products of other genes, as well as its own. Subsequently, the state of the cell is an action based on a combination of diverse translated gene products. When we observe these properties, we see a resemblance to the dynamics of an ANN, specifically a Recurrent Neural Network (RNN), where the current state depends on the previous. This means that there is a potential to create MML from manipulating the gene expression patterns.

To describe our concept, we will focus on a simple communication pattern found in the GRN of a bacterial cell. Bacteria uses signal transduction pathways to sense the environment by processing input signals. *Two-Component Systems* (*TCS*) are among the most widespread signal transduction mechanisms, which contain a *Sensor Histidine Kinase* (*SHK*) that receives external signals and a response regulator that accordingly initiates the expression of a set of genes. On average, a bacterial cell contains 30 TCSs that are essential for their virulence, growth, and survival. Approximately 87% of the known response regulators of TCS involve gene expression regulation at the transcription layer. Based on this, 96% of SHKs are capable of sensing small-molecule-binding from the extracellular space. Hence, the combination of TCSs can be considered a viable example of a natural GRN pattern that can be modeled and characterized as an ANN, where the input layer is represented by the SHKs, and multiple hidden layers as

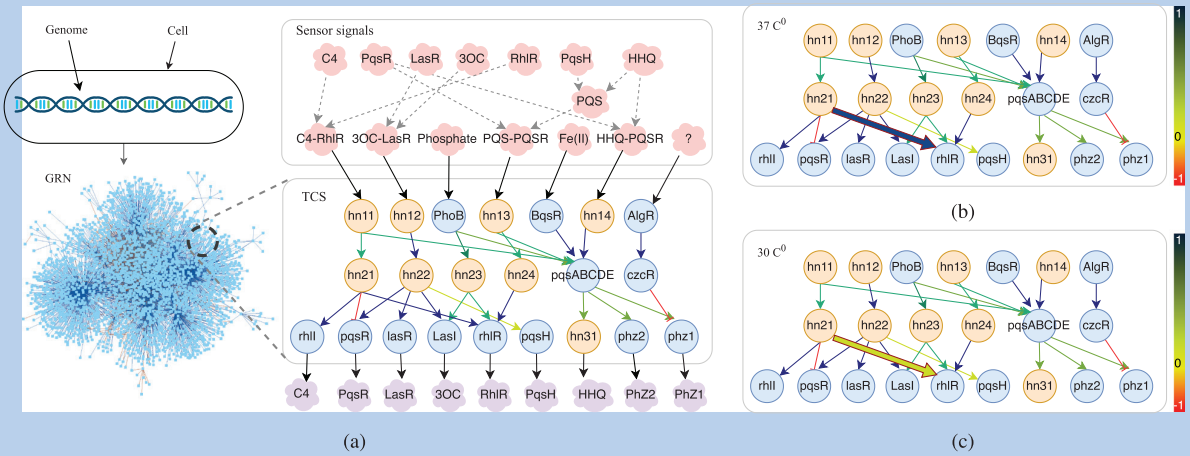



**FIGURE 2** Illustration of inherited GRAI where (a) shows the extraction of a subnetwork that resembles an ANN with relative weights, (b) set of relative weights in one environment condition (temperature at 37°*C*), and (c) modified weight in a different environment condition (temperature at 30°*C*).

well as an output layer consist of genes and their mutual interactions. There are several advantages in using the TCS sub-network of the GRN as an ANN for MML. This includes availability of experimental data that offer validation and quantification of the relationships between gene expressions for both input and output layers. In a number of cases, the direct mapping of a GRN sub-network to an ANN is not feasible. The reason is because sometimes the number of gene interactions (network hops) from the input layer to the output layer can vary for different gene expression paths, resulting in the corresponding ANN to be asymmetric, which leads to less computational efficiency. There are well-known approaches to address this problem, such as introducing phantom nodes that do not alter the overall behavior or treat the network as asymmetric ANN structure. Another alternative is to introduce missing gene interactions through engineered genetic circuits, which can further align the sub-network closer to a typical ANN structure.

Figure 2 illustrates how we recognize an ANN structure from a TCS sub-network of a GRN. As shown in the figure, the cell is able to combine multiple input signals and accordingly express downstream genes through the network. Gene expression products from one gene reach the non-coding region of another via intra-cellular diffusion [31]. The relationship of genes to be expressed in the network can be associated to a set of weights. The values of the weights are a result of several factors that include the transcription factors, affinity of the transcription factor binding site, thermoregulation, enhancers [32], as well as the noise due to the diffusive motion of regulatory molecules [33], [34]. Here, we focus mainly on two TCSs: *PhoB-PhoR* and *BqsR-BqsS* systems, which are associated with phosphate and iron uptake of the *P. aeruginosa* species. Further, we target the inter-cellular molecular communications by considering three QS systems, namely, *Las*, *Rhl*, and *PQS* genes where *Las* uses *3O-C12-HSL* and *Rhl* uses *C4-HSL*, while the *PQS* relies on *2-heptyl-3-hydroxy-4(1H)-quinolone*. To identify the corresponding ANN structure, we first modeled the GRNs as graphs using the interaction structural data from publicly available database [35]. This is followed by extracting the TCS sub-network related to the phosphate intakes iron along with the quorum sensing process. The obtained ANN model contains various numbers of hops from the input layer to the output layer, which require the introduction of phantom nodes that do not have an impact on the interaction dynamics of the network. The weights of the ANN represented by the TCS are estimated relatively using the interaction dynamics, as well as transcriptomic data [36], [37]. The performance accuracy of this model is then evaluated based the pyocyanin production and gene expression levels in low and high phosphate conditions with the data from wet-lab experiments in similar setups [38].

A typical ANN will require modification of weights as it is being trained to serve for a specific purpose. Here, we investigated how the weights of the ANN related to the TCS can be changed with a specific focus on changes that can be operated externally to the biological cell from the environment. Previous research has demonstrated how the temperature can impact the cellular functions of *P. aeruginosa*. This usually results in the modulation of one specific gene expression interaction of the *Rhl* QS system [32]. As highlighted in Figure 2(b), with the reception of *C4-RhlR* at 37°C temperature, the weight of *hn21 - rhlR* is significantly higher compared to the same at 30°C, as shown in Figure 2(c). This corresponds to a higher expression rate of *RhlR* at 37°C. This demonstrates that updating and training of GRAIs is possible through changes in the environmental conditions such as temperature.

## MINING ANN IN GRNS

Our previous section has shown that certain sub-networks of the GRN exhibit natural neural networks. In this section,

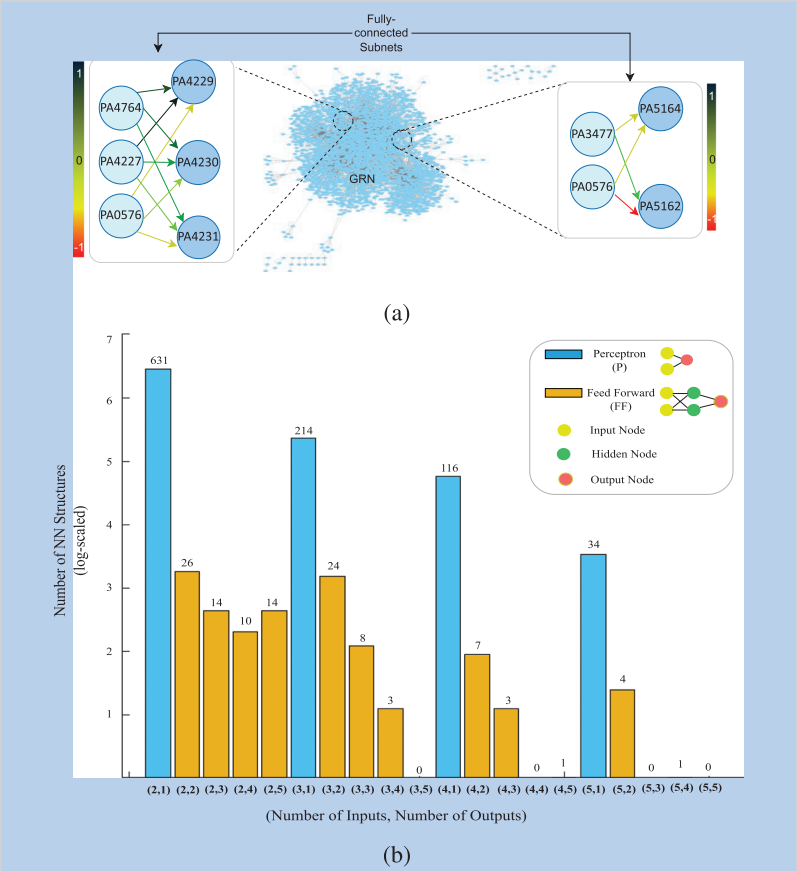



**FIGURE 3** Two fully-connected ANN sub-networks extracted from the full GRN is shown in (a) and number of different sub-network structures that can extracted from the GRN is illustrated in (b).

we want to investigate if other sub-networks that exhibit ANN structures can be extracted from the GRN. We perform this through a search algorithm that mines the GRN for specific types of structures. During the search process, if we need a structure with $i$ number of input nodes and $j$ number of output nodes, the algorithm first mines $j$ number of nodes that have a common predecessor. The $j$ number of nodes will have a number of different predecessors and will be put together into the same group. Within the same group, the nodes will be put together to create different combination, where the combinations must have $i$ number of input nodes that re the predecessor, as well as $j$ output nodes. These combination will reflect the different number of sub-network for nodes input nodes $i$ and output nodes $j$.

Figure 3(a) illustrates examples of a Feed-Forward neural network with different structures of fully connected ANN sub-networks extracted from the GRN. Figure 3(b) shows the number of perceptron and Feed-Forward neural network structures we obtained from the GRN using our mining algorithm. We are able to discover a significant number of perceptron structures with the highest recorded for one output node and two input nodes. As we increase the number of inputs, the number of fully connected Feed-Forward networks becomes harder to discover. In particular, Feed-Forward networks with five output nodes and higher than three input nodes are very rare.

Since these Feed-Forward neural networks are pre-trained with defined weights, the question now rises as to how we can use this for applications. One approach towards using the ANN found in the GRN is to match it to an application's requirement. This will require a mining algorithm that matches the problems that require an ANN with the same structure as well as weight combination. While this can create challenges in terms of finding the right problem to suit the ANN found in a GRN, there is an opportunity to engineer the circuit with addition of genes that will increase the diversity of the network, as well as integrate hidden layers.

## BACTERIAL MULTI-SPECIES DIFFUSION-BASED NEURAL NETWORK

In this section, we look at an alternative model for MML, where we investigate how multiple species of bacteria with symbiotic relationships, such as those found in a bacteriome, i.e., bacteria living in endosymbiosis with a host organisms, can be modeled and exploited as an ANN. In general, bacteria of the same species receive specific types of molecular signals from other populations and process them to produce a set of molecules that can influence other species or host cells. These multi-species bacterial populations can be considered the nodes of a network, where the molecular signals that diffuse between population are the link/edges, based on diffusion-based molecular communications. As the molecular signal cascades through the network from layer to layer, this resembles a feed forward neural network (layer in this instance are bacterial species that receive the same type of signals). The relationship structure of the bacteria and signaling weights depend on factors such as the diversity of the species, population sizes, cross-feeding/intercellular communications and molecular signal diffusion dynamics. The population sizes determine the rate of molecular signal reception and production, and this reflects the weight of the edges of the corresponding ANN model. If a larger population produces a signal and another population that has higher relative abundance consumes that signal, the weight corresponding to the link between these larger populations will be modeled with an ANN edge with a larger weight. On the other hand, if the population sizes of the two different species are smaller, the interaction between them is comparatively weaker and will result in a smaller weight value of the corresponding edge.

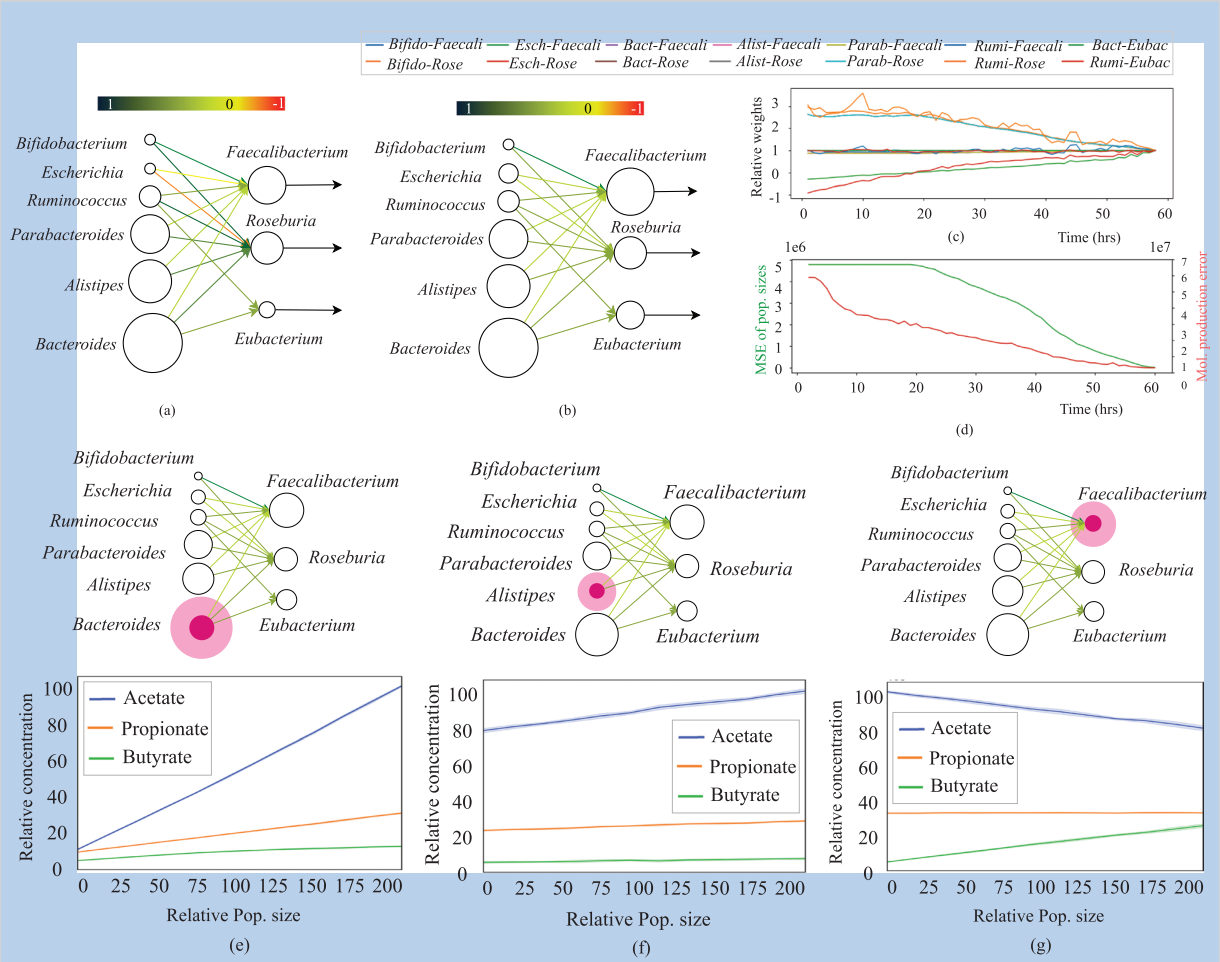



**FIGURE 4** Illustration of population-based ANN weight alteration and its impact on the network outputs is shown here, where (a) is the initial ANN setup, (b) is the ANN with the preferred network weights, (c) is the convergence of weights of all the edges relative to the preferred ANN over the transformation period, and (d) is the MSE behaviors of molecular production relative to the preferred ANN weights. Further, the output signal behaviors due to variations in weights caused by network structural changes are shown in (e), (f), and (g) by changing the population sizes of Bacteroides, Alistipes, and Faecalibacterium, respectively.

One of the well-studied bacterial ecosystems is the Human Gut Bacteriome (HGB), which constitutes up to 1000 species [39], and it suggests a relevant use case for the aforementioned concept. The reliability of the molecular signal flow between the different species is vital in modeling and exploiting the ecosystem as an ANN. In our previous study, the structural derivation of a network of multi-bacterial species using graph theory was analyzed, where input of glucose is received by certain species to produce various Short Chain Fatty Acid (SCFA) communicated between the cells [40]. The study revealed that the weights of the edges, which are the lactate and acetate signals exchanged between the populations, can be modified and adapted based on external inputs (e.g., glucose). Using this concept, we believe we could design a Bacterial Multi-species ANN from the SCFA molecular communication network within the HGB. Figure 4(a) illustrates an example of multi-species bacteria population that are organized into an ANN structure. The arrangement of the structure is based on the input-output relationship of molecular production. For example, when input glucose is consumed, it produces lactate and two SCFA (acetate and propionate) by six species to produce butyrate for other species, then the six species will be the first layer of a corresponding ANN of our NN, and the species that produce butyrate will be the ANN's second layer. Figure 4(a) shows the ANN with the relative weights of each edge shown with different color shades. Our aim is to train the ANN in Figure 4(a) into an ANN with a specific functionality, shown in Figure 4(b). Our training is based on the external input of glucose, where we can see in Figure 4(c) that as the species are consuming and producing molecules, their weight is slowly being modulated by changing the population sizes see Figure 4(d) (as the Mean Squared Error (MSE) of the population

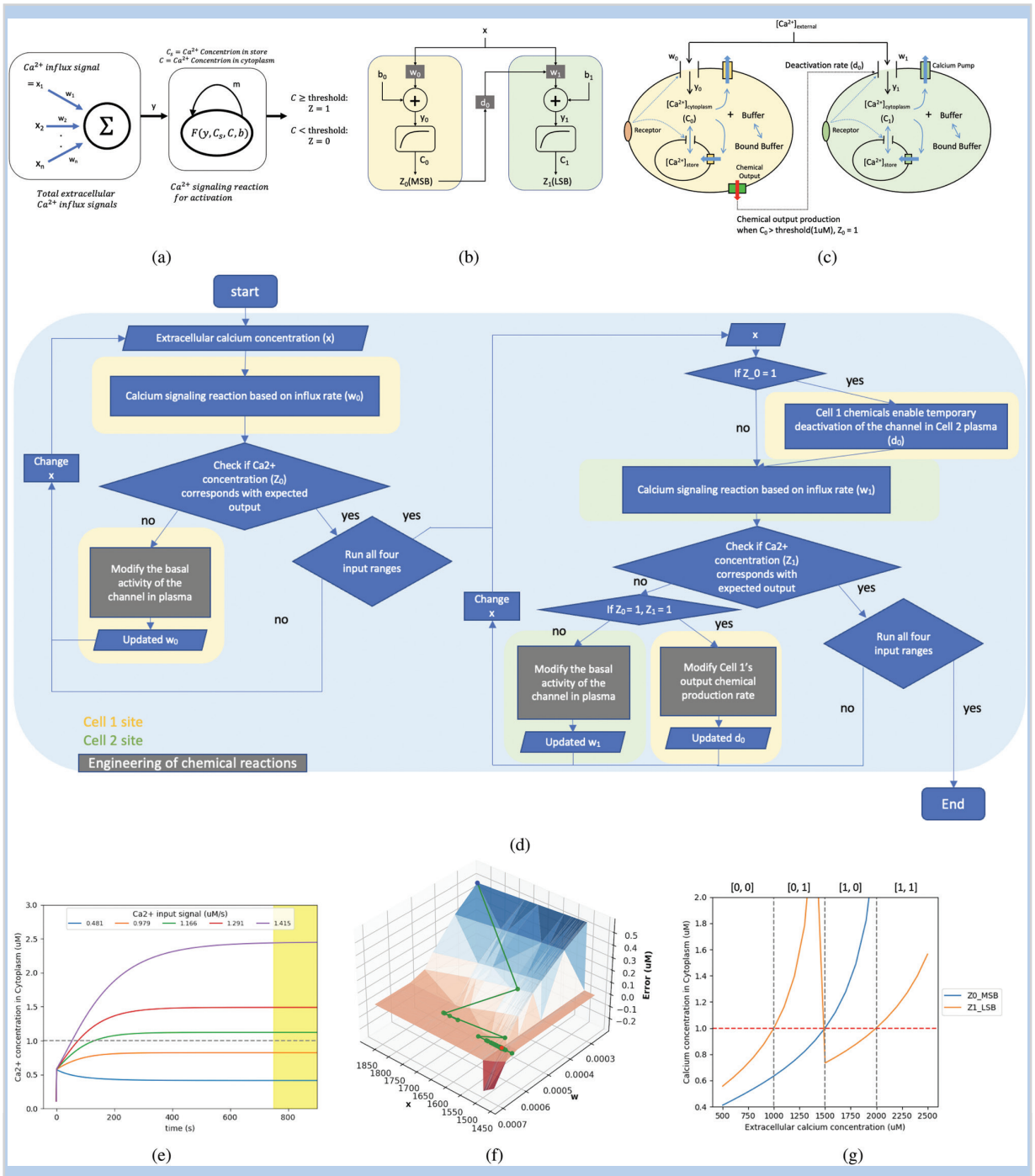



**FIGURE 5** Transforming $Ca^{2+}$ ions molecular communication into a perceptron. (a) A conventional perceptron model, (b) a two-bit ADC architecture, (c) engineering $Ca^{2+}$ signaling into an ADC between two cells, (d) $Ca^{2+}$ signaling training process to modify the basal functional activity and communication channel flowchart, (e) trained $Ca^{2+}$ ions transients in the cytoplasm, (f) dynamics of Cell 1 weight $w_0$ through the training process with respect to the input extracellular $Ca^{2+}$ input ($x$), and (g) variations in output $Ca^{2+}$ ions for the two cells to represent the ADC digital bits.

converges, similarly the molecular production error). Further, we show how significant the impact of the population size variation is on the overall gut metabolic performance by altering the abundance of each species relative to a healthy HGB composition. Figure 4(e) shows the network outputs in terms of acetate, propionate, and butyrate when the abundance of *Bacteroides* is changed from zero cells in the environment to a population size of 200% as in the

healthy HGB. Figure 4(f) and (g) present the behaviors of the same outputs when altering the population sizes of *Alistipes* and *Faecalibacterium*, respectively. These results indicate the possibility of altering weights of Bacterial Multi-species ANN to modify the network outputs significantly, which can be used in applications such as personalized treatment of metabolic disorders.

## $Ca^{2+}$ SIGNALING PERCEPTRON BASED ON MOLECULAR COMMUNICATIONS

In this section, we discuss a perceptron that can be trained by controlling the ion flow as well as the basal reactions of $Ca^{2+}$ Signaling between biological cells. As an example, we demonstrate the design of a multi-cell ADC realized by modulating the cell's $Ca^{2+}$ influx, as well as through the engineering of genetic circuits.

### CALCIUM SIGNALING

Communication through $Ca^{2+}$ ions is one of the essential signaling processes at the basis of numerous cell functions. While a few mathematical models for $Ca^{2+}$ signaling have been proposed, the model by Korngren et al. for $Ca^{2+}$ ion transients in electrically non-excitable cells is one of the most recognized and is at the basis of the concepts we present in the following [41]. According to this well-regarded model, this communication process is based on $Ca^{2+}$ ion influx into the cytoplasm from the extracellular medium, where ion-conducting channels are established through the membrane and controlled by receptors. The receptor in the model is designed in terms of a linear activation instead of complicated non-linear agonist binding curve [41]. As the influx of ions increases the $Ca^{2+}$ signaling reaction is activated, where the $Ca^{2+}$ ion pumps allow the outflow of ions from the cytoplasm to the external medium, as well as its store. Eventually, the $Ca^{2+}$ ions concentration in the cytoplasm reaches a saturated level. Based on this sequence of events, numerous $Ca^{2+}$ signaling based molecular communications systems, models, and their characterization have been investigated and proposed over the years [20], [42], [43], [44].

### OBTAINING A PERCEPTRON FROM $Ca^{2+}$ SIGNALING

We adapt the Korngreen et al. model to exploit a $Ca^{2+}$ signaling system as a perceptron. As illustrated in Figure 5(a), the input ( $x$ ) will be the $Ca^{2+}$ ion concentration in the extracellular medium and the weight ( $w$ ) is the $Ca^{2+}$ ions influx rate through the plasma membrane channels. Therefore, $x * w$ represents the amount of $Ca^{2+}$ ion influx ( $y$ ) into the cytoplasm, representing its transient. As described earlier, the $Ca^{2+}$ ion transients are multi-stage signaling processes that involve the transition of ions within the cytoplasm, store, buffer, as well as the extracellular medium, and regulate the concentration in the cytoplasm. In order to train the $Ca^{2+}$ signaling process into a perception, the cell needs to be the incorporation of an engineered genetic circuit to modify its basal fractional activity to trigger the $Ca^{2+}$ signaling reaction or to modulate the influx channel. In the case of a multiple-cell system to realize an ANN multi-perceptron network, the engineered genetic circuits are required to enable dynamic activation and deactivation of the $Ca^{2+}$ channel.

### TWO-BIT ANALOG TO DIGITAL CONVERTER ARCHITECTURE

We adapted the $Ca^{2+}$ ion signaling model to create interacting perceptrons in multiple cells that altogether realize a two-bit ADC through a simulation model. The architecture of a conventional ADC is illustrated in Figure 5(b). The equivalent model based on $Ca^{2+}$ signaling, where made clear the essential role of ion flow between two cells (the blue arrows in the Figure 5(c) indicate $Ca^{2+}$ ions reactions to facilitate this). The input $x$ is the incoming extracellular $Ca^{2+}$ concentration into the two cells, where the range of input considered in the simulation is set between $500\mu$ M to $2500\mu$ M and sampled according to an interval of $500\mu$ M. By dividing this range into four intervals, each interval will produce different $Ca^{2+}$ signals from two cells, i.e., *Cell* 1 and *Cell* 2 , which map to different digital bits. Based on this, the *Cell* 1 and *Cell* 2 produce the Most Significant Bit (MSB) and the least significant bit (LSB), respectively. $Ca^{2+}$ ions in the extracellular medium( $x$ ) flow into the cytoplasm through the $Ca^{2+}$ channel with an influx rate $w_0$ and $w_1$ for *Cell* 1 and *Cell* 2 , respectively. A bias to the $Ca^{2+}$ ions influx for each of the two cells ( $y_0$ , $y_1$ ) is randomly selected and applied (in this example this is $b_0 = 0.169255\mu$ M and $b_1 = 0.287264\mu$ M, respectively). Through the $Ca^{2+}$ transients, the ion concentrations in the cytoplasm that are set to $C_0$ and $C_1$ , respectively. By setting a threshold, in our case, $1\mu$ M, the $Ca^{2+}$ concentration in the cytoplasm, can be converted into digital bit ( $Z_0$ , $Z_1$ ), which are the MSB and LSB. In order to make an ADC, *Cell* 1 is genetically engineered to produce molecules when enough $Ca^{2+}$ ions ( $1\mu$ M) are present in the cytoplasm. The output molecules temporally deactivate the calcium channel in *Cell* 2 plasma. This deactivation rate is indicated as $d_0$ .

### TRAINING PROCESS

The flow chart for training the $Ca^{2+}$ signaling perceptron is presented in Figure 5(d). The two cells have to be trained to obtain optimal $Ca^{2+}$ influx rates ($w_0$, $w_1$) as well as the correct *Cell* 1's calcium channel deactivation rate for *Cell* 2 ($d_0$) so that *Cell* 1 and *Cell* 2 can produce the aforementioned MSB and LSB, respectively. *Cell* 1 is trained first to find an optimal $w_0$, and then *Cell* 2 to obtain $w_1$ and $d_0$. With initial $w_0$, $Ca^{2+}$ flows into *Cell* 1 and is regulated in the cytoplasm ($C_0$) for a certain period. Based on the amount of input from the extracellular medium ( $x$ ), the concentration at saturation will represent an MSB digital bit ($Z_0$). When $Z_0$ is bit 0, but the expected output is bit 1: an activation chemical from the engineered circuit is injected to elevate the basal activity of the calcium channel in *Cell* 1 plasma. Due to the increased activity of the channel, an increased amount of $Ca^{2+}$ ions will flow into *Cell* 1, which means the influx rate ($w_0$) is also increased. For the opposite case, when $Z_0$ is bit 1 and the expected value is bit 0, a different deactivation chemical signal is expressed by the engineered genetic circuit to reduce the basal activity of the

$Ca^{2+}$ channel. Then, $w_0$ is updated to a lower value. Based on this sequential training process, the optimal $w_0$ will be found. The same training process is performed on *Cell* 2, except for one case. This exception case is when $Z_0$ and $Z_1$ are bits 1, but the expected $Z_1$ is bit 0, which will require manual intervention to modify the rate of *Cell* 1 output chemical production instead of injecting chemicals. Figure 5(e) shows how the perceptron behaves for different levels of $Ca^{2+}$ within the cytoplasm based on varying extracellular influx. Figure 5(f) illustrates an example of convergence of weight $w_0$ during training with respect to the error for varying levels of extracellular input ($x$). Finally, Figure 5(g) shows the variations of output from the two cells that represent digital bits from *Cell* 1 and *Cell* 2. For example, an input between $1000\mu$ M and $1500\mu$ M results in '01,' where the 0 b is from *Cell* 1 and 1 b is from *Cell* 2.

## CHALLENGES

While we have identified solutions that enable non-neural cells to develop perceptron properties, or the exploitation of gene regulations to obtain ANN functionalities, there are still a number of challenges that need to be addressed to move towards practical applications in the future, and some important ones are discussed next.

### CONTROLLING MOLECULAR COMMUNICATIONS IN MOLECULAR MACHINE LEARNING

The MML that we have discussed so far are based on training and computing operations that stem from communications of molecules and chemical reactions. To develop MML systems processes matching the computational capabilities of silicon-based technologies, we will eventually need to consider multilayer perceptron architectures. While the genetic engineering will possibly be the main enabling technology, specific challenges are as follows. First, since the training of the edge weights of molecular signals, which in our case is based on population control, a mechanism is required to ensure that parallel changes in the bacteriome can be performed to modify the relative population of different species/strains in the system. This becomes more challenging when we consider $Ca^{2+}$ signaling between cells and in particular controlling the flow of ions through the gap junction of cells. Second, while GRAI might be inherently including multi-layer perceptrons, the question is how do we determine appropriate chemical inputs to express genes of the input nodes and, at the same time, detect expressions on specific output nodes. From a multi-bacterial species perspective, this will require engineering of cells with different receptors to detect diverse molecular signals from the previous layers. The cells will, therefore, need to have the ability to detect signals efficiently and operate in noisy environments. The other challenge is the ability to synchronize all transmissions as signals propagate between different layers. The latter challenge can have an immense impact on the reliability of the resulting ANN. Since we have shown that multiple ANNs are embedded in a GRN through a subnetwork, the question is whether multiple parallel processing can be achieved through different gene expression paths.

### BIO-HYBRID AI

The paradigm of the Internet of Bio-Nano Things [45] includes the need to interconnect molecular communication systems to connect to the cyber-Internet by propagating information between the molecular and the electrical domains. This can be realized through an electro-chemical based Bio-cyber interfaces. While this can allow to detect chemical outputs from the MML, an issue arises when we want to actively interact and reconfigure the MML system from the electrical domain. In particular, the challenge lies in the mechanism to reconfigure the weights.

### RESPONSIBLE AI IN MOLECULAR MACHINE LEARNING

As AI continues to spread and weave into our everyday lives, besides developing sophisticated hardware and software, we are facing new and emerging ethical concerns has risen, which altogether call for the notion of responsible AI. Responsible AI aims to address the ethical and legal issues in regards to deployment, as well as utilization of AI. This is already a major challenge in conventional AI, which is necessary to address to provide trust for the public in using the technology. This challenge will deepen further when AI is extended in living machines. This is particularly true when we consider the potential applications of learning-based living machines for treating diseases, where they can potentially be deployed into the body or the environment. Another challenge is also the security aspect, in the similar manner that this is a challenge in conventional AI.

## CONCLUSION

As our society embraces AI to play a part in our everyday lives, we are starting to witness various forms and algorithms that are embedded into devices with different computational capabilites. In this article, we investigate MML for Biological AI, where AI occurs in living systems and is based on information propagation through chemical reaction and molecule transport, i.e., molecular communications. We reviewed the current background in Biological AI. This is followed by our proposed directions of MML through the GRN, bacterial multi-species communication, as well as $Ca^{2+}$ signaling. We then discuss future possible directions for the molecular communications research.

## ACKNOWLEDGMENTS

This work was supported by the National Institute of Health under Award No. P20 GM104320.

## ABOUT THE AUTHORS

***Sasitharan Balasubramaniam*** (corresponding author: (sasi@unl.edu)) is with the School of Computing, University of Nebraska-Lincoln, Lincoln, NE, 68588, USA.

***Samitha Somathilaka*** (ssomathilaka2@unl.edu) is with the School of Computing University of Nebraska-Lincoln, Lincoln, NE, 68588, USA, and also with Walton Institute South East Technological University, Carlow, Ireland.

***Sehee Sun*** (ssun12@unl.edu) is with the School of Computing, University of Nebraska-Lincoln, Lincoln, NE, 68588, USA.

# CHAPTER 10. JOURNAL: REALIZING MOLECULAR MACHINE LEARNING THROUGH COMMUNICATIONS FOR BIOLOGICAL AI: FUTURE DIRECTIONS AND CHALLENGES

***Adrian Ratwatte*** (aratwatte2@unl.edu) is with the School of Computing University of Nebraska-Lincoln, Lincoln, NE, 68588, USA.

***Massimiliano Pierobon*** (maxp@unl.edu) is with the School of Computing University of Nebraska-Lincoln, Lincoln, NE, 68588, USA.

## REFERENCES

[1] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[3] T. Bohnstingl, F. Scherr, C. Pehle, K. Meier, and W. Maass, "Neuromorphic hardware learns to learn," *Front. Neurosci.*, vol. 13, 2019, Art. no. 483.

[4] D. Liu, H. Yu, and Y. Chai, "Low-power computing with neuromorphic engineering," *Adv. Intell. Syst.*, vol. 3, no. 2, 2021, Art. no. 2000150.

[5] A. Hirohata et al., "Review on spintronics: Principles and device applications," *J. Magnetism Magn. Mater.*, vol. 509, 2020, Art. no. 166711.

[6] B. J. Kagan et al., "In vitro neurons learn and exhibit sentience when embodied in a simulated game-world," *Neuron*, vol. 110, no. 23, pp. 3952–3969, 2022.

[7] K. Warwick, S. J. Nasuto, V. M. Becerra, and B. J. Whalley, "Experiments with an in-vitro robot brain," in *Computing With Instinct*, Berlin, Germany:Springer, 2011, pp. 1–15.

[8] D. J. Bakkum et al., "Embodying cultured networks with a robotic drawing arm," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2007, pp. 2996–2999.

[9] L. Smirnova et al., "Organoid intelligence (OI): The new frontier in biocomputing and intelligence-in-a-dish," *Front. Sci.*, vol. 1, 2023, Art. no. 1017235.

[10] A. Tero et al., "Rules for biologically inspired adaptive network design," *Science*, vol. 327, no. 5964, pp. 439–442, 2010.

[11] N. Roberts and A. Adamatzky, "Mining logical circuits in fungi," *Sci. Rep.*, vol. 12, no. 1, 2022, Art. no. 15930.

[12] K. Sarkar, D. Bonnerjee, R. Srivastava, and S. Bagh, "A single layer artificial neural network type architecture with molecular engineered bacteria for reversible and irreversible computing," *Chem. Sci.*, vol. 12, no. 48, pp. 15821–15832, 2021.

[13] L. Rizik, L. Danial, M. Habib, R. Weiss, and R. Daniel, "Synthetic neuromorphic computing in living cells," *Nature Commun.*, vol. 13, no. 1, pp. 1–17, 2022.

[14] A. Pandi et al., "Metabolic perceptrons for neural computing in biological systems," *Nature Commun.*, vol. 10, no. 1, pp. 1–13, 2019.

[15] M. Pierobon and I. F. Akyildiz, "A physical end-to-end model for molecular communication in nanonetworks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 4, pp. 602–611, May 2010.

[16] V. Jamali, A. Ahmadzadeh, C. Jardin, H. Sticht, and R. Schober, "Channel estimation for diffusive molecular communications," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4238–4252, Oct. 2016.

[17] W. Guo et al., "Molecular communications: Channel model and physical layer techniques," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 120–127, Aug. 2016.

[18] C. Wu, L. Lin, W. Guo, and H. Yan, "Signal detection for molecular MIMO communications with asymmetrical topology," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 6, no. 1, pp. 60–70, Jul. 2020.

[19] X. Chen, Y. Huang, L.-L. Yang, and M. Wen, "Generalized molecular-shift keying (GMOSK): Principles and performance analysis," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 6, no. 3, pp. 168–183, Dec. 2020.

[20] A. O. Bicen, I. F. Akyildiz, S. Balasubramaniam, and Y. Koucheryavy, "Linear channel modeling and error analysis for intra/inter-cellular $Ca^{2+}$ molecular communication," *IEEE Trans. Nanobiosci.*, vol. 15, no. 5, pp. 488–498, Jul. 2016.

[21] A. Ahmadzadeh, A. Noel, and R. Schober, "Analysis and design of multi-hop diffusion-based molecular communication networks," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 1, no. 2, pp. 144–157, 2015.

[22] Y. Deng, A. Noel, W. Guo, A. Nallanathan, and M. Elkashlan, "Analyzing large-scale multiuser molecular communication via 3-d stochastic geometry," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 3, no. 2, pp. 118–133, Jul. 2017.

[23] N. Farsad, D. Pan, and A. Goldsmith, "A novel experimental platform for in-vessel multi-chemical molecular communications," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–6.

[24] L. Grebenstein et al., "Biological optical-to-chemical signal conversion interface: A small-scale modulator for molecular communications," in *Proc. 5th ACM Int. Conf. Nanoscale Comput. Commun.*, 2018, pp. 1–6.

[25] P. M. Pflüger and F. Glorius, "Molecular machine learning: The future of synthetic chemistry?," *Angewandte Chemie Int. Ed.*, vol. 59, no. 43, pp. 18860–18865, 2020.

[26] E. A. Liberman and S. V. Minina, "Cell molecular computers and biological information as the foundation of nature's laws," *BioSystems*, vol. 38, no. 2-3, pp. 173–177, 1996.

[27] C.-Y. Yang et al., "Encoding membrane-potential-based memory within a microbial community," *Cell Syst.*, vol. 10, no. 5, pp. 417–423, 2020.

[28] A. G. Becerra, M. Gutiérrez, and R. Lahoz-Beltra, "Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network," *Biosystems*, vol. 213, 2022, Art. no. 104608.

[29] X. Li, L. Rizik, V. Kravchik, M. Khoury, N. Korin, and R. Daniel, "Synthetic neural-like computing in microbial consortia for pattern recognition," *Nature Commun.*, vol. 12, no. 1, pp. 1–12, 2021.

[30] D. Bray, "Protein molecules as computational elements in living cells," *Nature*, vol. 376, no. 6538, pp. 307–312, 1995.

[31] P. E. Schavemaker, A. J. Boersma, and B. Poolman, "How important is protein diffusion in prokaryotes?," *Front. Mol. Biosci.*, vol. 5, 2018, Art. no. 93.

[32] M. V. Grosso-Becerra, G. Croda-García, E. Merino, L. Servín-González, R. Mojica-Espinosa, and G. Soberón-Chávez, "Regulation of pseudomonas aeruginosa virulence factors by two novel RNA thermometers," *Proc. Nat. Acad. Sci.*, vol. 111, no. 43, pp. 15562–15567, 2014.

[33] J. S. van Zon, M. J. Morelli, S. Tănase-Nicola, and P. R. ten Wolde, "Diffusion of transcription factors can drastically enhance the noise in gene expression," *Biophys. J.*, vol. 91, no. 12, pp. 4350–4367, 2006.

[34] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, "A comprehensive survey of recent advancements in molecular communication," *IEEE Commun. Surv. Tut.*, vol. 18, no. 3, pp. 1887–1919, thirdquarter 2016.

[35] E. Galán-Vásquez, B. C. Luna-Olivera, M. Ramírez-Ibáñez, and A. Martínez-Antonio, "RegulomePA: A database of transcriptional regulatory interactions in pseudomonas aeruginosa PAO1," *Database*, vol. 2020, 2020, Art. no. baaa106.

[36] V. Venturi, "Regulation of quorum sensing in pseudomonas," *FEMS Microbiol. Rev.*, vol. 30, no. 2, pp. 274–291, 2006.

[37] V. I. Francis, E. C. Stevenson, and S. L. Porter, "Two-component systems required for virulence in Pseudomonas aeruginosa," *FEMS Microbiol. Lett.*, vol. 364, no. 11, 2017, Art. no. fnx104.

[38] X. Meng, S. D. Ahator, and L.-H. Zhang, "Molecular mechanisms of phosphate stress activation of Pseudomonas aeruginosa quorum sensing systems," *MSphere*, vol. 5, no. 2, 2020, Art. no. e00119–20.

[39] S. S. Somathilaka, D. P. Martins, and S. Balasubramaniam, "Information flow of cascading bacterial molecular communication systems with cooperative amplification," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 1728–1733.

[40] S. S. Somathilaka, D. P. Martins, W. Barton, O. O'Sullivan, P. D. Cotter, and S. Balasubramaniam, "A graph-based molecular communications model analysis of the human gut bacteriome," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 7, pp. 3567–3577, Jul. 2022.

[41] A. Korngreen, V. Gold'shtein, and Z. Priel, "A realistic model of biphasic calcium transients in electrically nonexcitable cells," *Biophys. J.*, vol. 73, no. 2, pp. 659–673, 1997.

[42] P. He, T. Nakano, D. Wu, B. Yang, H. Liu, and X. Han, "Calcium signaling in mobile molecular communication networks," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.

[43] C. Allan, R. J. Morris, and C.-N. Meisrimler, "Encoding, transmission, decoding, and specificity of calcium signals in plants," *J. Exp. Botany*, vol. 73, no. 11, pp. 3372–3385, 2022.

[44] M. T. Barros, S. Balasubramaniam, B. Jennings, and Y. Koucheryavy, "Transmission protocols for calcium-signaling-based molecular communications in deformable cellular tissue," *IEEE Trans. Nanotechnol.*, vol. 13, no. 4, pp. 779–788, Jul. 2014.

[45] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy, "The internet of Bio-Nano Things," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 32–40, Mar. 2015.

N

# Bibliography

[1] J. Fan, L. Fang, J. Wu, Y. Guo, and Q. Dai, "From brain science to artificial intelligence," *Engineering*, vol. 6, no. 3, pp. 248–252, 2020.

[2] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug *et al.*, "Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International Journal of Information Management*, vol. 57, p. 101994, 2021.

[3] R. Li, *Artificial intelligence revolution: How AI will change our society, economy, and culture*. Simon and Schuster, 2020.

[4] I. H. Sarker, "Ai-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems," *SN Computer Science*, vol. 3, no. 2, p. 158, 2022.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[6] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[7] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] F. Rosenblatt and S. Papert, *Perceptron*. April, 2021, vol. 9.

[9] H. Wang and B. Raj, "On the origin of deep learning," *arXiv preprint arXiv:1702.07800*, 2017.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[11] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural computation*, vol. 1, no. 1, pp. 39–46, 1989.

[12] R. Hecht-Nielsen, *Neurocomputing*. Upper Saddle River, NJ: Pearson, Jan. 1990.

[13] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] R. Daniel, J. R. Rubens, R. Sarpeshkar, and T. K. Lu, "Synthetic analog computation in living cells," *Nature*, vol. 497, no. 7451, pp. 619–623, 2013.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] S. K. Singh, S. Kumar, and P. S. Mehra, "Chat gpt & google bard ai: A review," in *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*. IEEE, 2023, pp. 1–6.

[18] P. Warden and D. Situnayake, *Tinyml*. O'Reilly Media, Dec. 2019.

[19] B. J. Kagan, A. C. Kitchen, N. T. Tran, F. Habibollahi, M. Khajehnejad, B. J. Parker, A. Bhat, B. Rollo, A. Razi, and K. J. Friston, "In vitro neurons learn and exhibit sentience when embodied in a simulated game-world," *Neuron*, vol. 110, no. 23, pp. 3952–3969, 2022.

[20] L. Rizik, L. Danial, M. Habib, R. Weiss, and R. Daniel, "Synthetic neuromorphic computing in living cells," *Nature communications*, vol. 13, no. 1, p. 5602, 2022.

[21] I. E. Morales Pantoja, L. Smirnova, A. R. Muotri, K. J. Wahlin, J. Kahn, J. L. Boyd, D. H. Gracias, T. D. Harris, T. Cohen-Karni, B. S. Caffo *et al.*, "First organoid intelligence (oi) workshop to form an oi community," *Frontiers in Artificial Intelligence*, vol. 6, p. 1116870, 2023.

[22] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 459–468.

[23] B. Baker, O. Gupta, R. Raskar, and N. Naik, "Accelerating neural architecture search using performance prediction," *arXiv preprint arXiv:1705.10823*, 2017.

[24] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *International conference on machine learning*. PMLR, 2017, pp. 2902–2911.

[25] L. Grozinger, M. Amos, T. E. Gorochowski, P. Carbonell, D. A. Oyarzún, R. Stoof, H. Fellermann, P. Zuliani, H. Tas, and A. Goñi-Moreno, "Pathways to cellular supremacy in biocomputing," *Nature communications*, vol. 10, no. 1, p. 5250, 2019.

[26] "Q&A: UW researcher discusses just how much energy ChatGPT uses — washington.edu," https://www.washington.edu/news/2023/07/27/how-much-energy-does-chatgpt-use/, [Accessed 23-03-2024].

[27] L. Smirnova, B. S. Caffo, D. H. Gracias, Q. Huang, I. E. Morales Pantoja, B. Tang, D. J. Zack, C. A. Berlinicke, J. L. Boyd, T. D. Harris *et al.*, "Organoid intelligence (oi): the new frontier in biocomputing and intelligence-in-a-dish," *Frontiers in Science*, vol. 1, p. 1017235, 2023.

[28] K. M. Stiefel and J. S. Coggan, "The energy challenges of artificial superintelligence," *Frontiers in Artificial Intelligence*, vol. 6, p. 1240653, 2023.

[29] P. Warden and D. Situnayake, *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media, 2019.

[30] P. P. Ray, "A review on tinyml: State-of-the-art and prospects," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2022.

[31] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing," *Advances in neural information processing systems*, vol. 28, 2015.

[32] J. D. Smith, A. J. Hill, L. E. Reeder, B. C. Franke, R. B. Lehoucq, O. Parekh, W. Severa, and J. B. Aimone, "Neuromorphic scaling advantages for energy-efficient random walk computations," *Nature Electronics*, vol. 5, no. 2, pp. 102–112, 2022.

[33] D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, "Physics for neuromorphic computing," *Nature Reviews Physics*, vol. 2, no. 9, pp. 499–510, 2020.

[34] X. Qian, H. Song, and G.-l. Ming, "Brain organoids: advances, applications and challenges," *Development*, vol. 146, no. 8, p. dev166074, 2019.

[35] H. Cai, Z. Ao, C. Tian, Z. Wu, H. Liu, J. Tchieu, M. Gu, K. Mackie, and F. Guo, "Brain organoid reservoir computing for artificial intelligence," *Nature Electronics*, vol. 6, no. 12, pp. 1032–1039, 2023.

[36] J. Badai, Q. Bu, and L. Zhang, "Review of artificial intelligence applications and algorithms for brain organoid research," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 12, no. 4, pp. 383–394, 2020.

[37] A. Adamatzky, "Towards proteinoid computers. hypothesis paper," *Biosystems*, vol. 208, p. 104480, 2021.

[38] P. Mougkogiannis and A. Adamatzky, "Learning in ensembles of proteinoid microspheres," *Royal Society Open Science*, vol. 10, no. 10, p. 230936, 2023.

[39] E. Alm, K. Huang, and A. Arkin, "The evolution of two-component systems in bacteria reveals different strategies for niche adaptation," *PLoS Computational Biology*, vol. 2, no. 11, p. e143, 2006. [Online]. Available: https://doi.org/10.1371/journal.pcbi.0020143

[40] A. G. Becerra, M. Gutiérrez, and R. Lahoz-Beltra, "Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network," *BioSystems*, vol. 213, p. 104608, 2022.

[41] D. F. Blair, "HOW BACTERIA SENSE AND SWIM," *Annual Review of Microbiology*, vol. 49, no. 1, pp. 489–520, Oct. 1995. [Online]. Available: https://doi.org/10.1146/annurev.mi.49.100195.002421

[42] R. Lahoz-Beltra, J. Navarro, and P. C. Marijuán, "Bacterial computing: a form of natural computing and its applications," *Frontiers in Microbiology*, vol. 5, p. 101, 2014.

[43] J. P. Armitage, "Bacterial tactic responses," *Advances in microbial physiology*, vol. 41, pp. 229–289, 1999.

[44] S. J. Sørensen, M. Burmølle, and L. H. Hansen, "Making bio-sense of toxicity: new developments in whole-cell biosensors," *Current opinion in biotechnology*, vol. 17, no. 1, pp. 11–16, 2006.

[45] C. D. Sifri, "Quorum sensing: bacteria talk sense," *Clinical infectious diseases*, vol. 47, no. 8, pp. 1070–1076, 2008.

[46] M. N. Levit and J. B. Stock, "ph sensing in bacterial chemotaxis," in *Novartis Foundation Symposium 221-Bacterial Responses to pH: Bacterial Responses to pH: Novartis Foundation Symposium 221.* Wiley Online Library, 2007, pp. 38–54.

[47] S. Shivaji and J. S. Prakash, "How do bacteria sense and respond to low temperature?" *Archives of microbiology*, vol. 192, pp. 85–95, 2010.

[48] S. Brantl, "Bacterial gene regulation: metal ion sensing by proteins or rna," *Trends in biotechnology*, vol. 24, no. 9, pp. 383–386, 2006.

[49] M. Gomelsky and W. D. Hoff, "Light helps bacteria make important lifestyle decisions," *Trends in microbiology*, vol. 19, no. 9, pp. 441–448, 2011.

[50] J. R. Lamb, H. Patel, T. Montminy, V. E. Wagner, and B. H. Iglewski, "Functionaldomains of the rhlr transcriptional regulator of pseudomonas aeruginosa," *Journal of bacteriology*, vol. 185, no. 24, 2003.

[51] J. P. Pearson, E. C. Pesci, and B. H. Iglewski, "Roles of pseudomonas aeruginosa las and rhl quorum-sensing systems in control of elastase and rhamnolipid biosynthesis genes," *Journal of bacteriology*, vol. 179, no. 18, pp. 5756–5767, 1997.

[52] D. S. Wade, M. W. Calfee, E. R. Rocha, E. A. Ling, E. Engstrom, J. P. Coleman, and E. C. Pesci, "Regulation of pseudomonas quinolone signal synthesis

in pseudomonas aeruginosa," *Journal of bacteriology*, vol. 187, no. 13, pp. 4372–4380, 2005.

[53] S. Baumberg, *Prokaryotic gene expression*. OUP Oxford, 1999, vol. 21.

[54] J.-J. M. Riethoven, "Regulatory regions in dna: promoters, enhancers, silencers, and insulators," *Computational biology of transcription factor binding*, pp. 33–42, 2010.

[55] A. Ishihama, "Prokaryotic genome regulation: multifactor promoters, multitarget regulators and hierarchic networks," *FEMS microbiology reviews*, vol. 34, no. 5, pp. 628–645, 2010.

[56] M. Land, L. Hauser, S.-R. Jun, I. Nookaew, M. R. Leuze, T.-H. Ahn, T. Karpinets, O. Lund, G. Kora, T. Wassenaar, S. Poudel, and D. W. Ussery, "Insights from 20 years of bacterial genome sequencing," *Functional & Integrative Genomics*, vol. 15, no. 2, pp. 141–161, Feb. 2015. [Online]. Available: https://doi.org/10.1007/s10142-015-0433-4

[57] C. Wang, S. Xu, and Z.-P. Liu, "Evaluating gene regulatory network activity from dynamic expression data by regularized constraint programming," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 11, pp. 5738–5749, 2022.

[58] "Fujitsu supercomputer simulates 1 second of brain activity — cnet.com," https://www.cnet.com/culture/fujitsu-supercomputer-simulates-1-second-of-brain-activity/, [Accessed 10-04-2024].

[59] L. M. Adleman, "Computing with dna," *Scientific american*, vol. 279, no. 2, pp. 54–61, 1998.

[60] M. Garzon, P. Neathery, R. Deaton, R. C. Murphy, D. R. Franceschetti, and S. Stevens Jr, "A new metric for dna computing," in *Proceedings of the 2nd genetic programming conference*, vol. 32, no. 1. Morgan Kaufman, 1997, pp. 636–638.

[61] Y. Benenson, R. Adar, T. Paz-Elizur, Z. Livneh, and E. Shapiro, "Dna molecule provides a computing machine with both data and fuel," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2191–2196, 2003.

[62] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A dna-based archival storage system," in *Proceedings of the twenty-first international conference on architectural support for programming languages and operating systems*, 2016, pp. 637–649.

[63] Z. Ezziane, "Dna computing: applications and challenges," *Nanotechnology*, vol. 17, no. 2, p. R27, 2005.

[64] H. Udono, J. Gong, Y. Sato, and M. Takinoue, "Dna droplets: intelligent, dynamic fluid," *Advanced Biology*, vol. 7, no. 3, p. 2200180, 2023.

[65] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–338, 2000.

[66] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, no. 6767, pp. 339–342, 2000.

[67] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro, "Programmable and autonomous computing machine made of biomolecules," *Nature*, vol. 414, no. 6862, pp. 430–434, 2001.

[68] F. Wang, H. Lv, Q. Li, J. Li, X. Zhang, J. Shi, L. Wang, and C. Fan, "Implementing digital computing with dna-based switching circuits," *Nature communications*, vol. 11, no. 1, p. 121, 2020.

[69] L. Qian and E. Winfree, "Scaling up digital circuit computation with dna strand displacement cascades," *science*, vol. 332, no. 6034, pp. 1196–1201, 2011.

[70] K. M. Cherry and L. Qian, "Scaling up molecular pattern recognition with dna-based winner-take-all neural networks," *Nature*, vol. 559, no. 7714, pp. 370–376, 2018.

[71] H. Lv, N. Xie, M. Li, M. Dong, C. Sun, Q. Zhang, L. Zhao, J. Li, X. Zuo, H. Chen *et al.*, "Dna-based programmable gate arrays for general-purpose dna computing," *Nature*, vol. 622, no. 7982, pp. 292–300, 2023.

[72] Y. Ishima, A. T. Przybylski, and S. W. Fox, "Electrical membrane phenomena in spherules from proteinoid and lecithin," *BioSystems*, vol. 13, no. 4, pp. 243–251, 1981.

[73] P. Mougkogiannis and A. Adamatzky, "On interaction of proteinoids with simulated neural networks," *BioSystems*, vol. 237, p. 105175, 2024.

[74] ——, "Logical gates in ensembles of proteinoid microspheres," *Plos one*, vol. 18, no. 9, p. e0289433, 2023.

[75] S. W. Fox, "Thermal proteins in the first life and in the "mind-body" problem," in *Evolution of Information Processing Systems: An Interdisciplinary Approach for a New Understanding of Nature and Society*. Springer, 1992, pp. 203–228.

[76] A. T. Przybylski, "Excitable cell made of thermal proteinoids," *BioSystems*, vol. 17, no. 4, pp. 281–288, 1985.

[77] R. Fortulan, N. R. Kheirabadi, P. Mougkogiannis, A. Chiolerio, and A. Adamatzky, "Reservoir computing with colloidal mixtures of zno and proteinoids," *arXiv preprint arXiv:2312.08130*, 2023.

[78] P. Mougkogiannis and A. Adamatzky, "Proteinoid microspheres as protoneural networks," *ACS omega*, vol. 8, no. 38, pp. 35 417–35 426, 2023.

[79] ——, "Proto-neurons from abiotic polypeptides," *Encyclopedia*, vol. 4, no. 1, pp. 512–543, 2024.

[80] S. Sharma, P. Mougoyannis, G. Tarabella, and A. Adamatzky, "A review on the protocols for the synthesis of proteinoids," *arXiv preprint arXiv:2212.02261*, 2022.

[81] J. L. Poet, A. M. Campbell, T. T. Eckdahl, and L. J. Heyer, "Bacterial computing," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 1, pp. 10–15, 2010.

[82] A. Goni-Moreno, M. Redondo-Nieto, F. Arroyo, and J. Castellanos, "Biocircuit design through engineering bacterial logic gates," *Natural Computing*, vol. 10, pp. 119–127, 2011.

[83] S. Regot, J. Macia, N. Conde, K. Furukawa, J. Kjellén, T. Peeters, S. Hohmann, E. De Nadal, F. Posas, and R. Solé, "Distributed biological computation with multicellular engineered networks," *Nature*, vol. 469, no. 7329, pp. 207–211, 2011.

[84] K. Sarkar, D. Bonnerjee, R. Srivastava, and S. Bagh, "A single layer artificial neural network type architecture with molecular engineered bacteria for reversible and irreversible computing," *Chemical science*, vol. 12, no. 48, pp. 15 821–15 832, 2021.

[85] X. Li, L. Rizik, V. Kravchik, M. Khoury, N. Korin, and R. Daniel, "Synthetic neural-like computing in microbial consortia for pattern recognition," *Nature communications*, vol. 12, no. 1, p. 3139, 2021.

[86] W. C. Hacker, S. Li, and A. H. Elcock, "Features of genomic organization in a nucleotide-resolution molecular model of the escherichia coli chromosome," *Nucleic acids research*, vol. 45, no. 13, pp. 7541–7554, 2017.

[87] M. L. Simpson, G. S. Sayler, J. T. Fleming, and B. Applegate, "Whole-cell biocomputing," *Trends in biotechnology*, vol. 19, no. 8, pp. 317–323, 2001.

[88] A. A. Green, J. Kim, D. Ma, P. A. Silver, J. J. Collins, and P. Yin, "Complex cellular logic computation using ribocomputing devices," *Nature*, vol. 548, no. 7665, pp. 117–121, 2017.

[89] S. Balasubramaniam, N. Lyamin, D. Kleyko, M. Skurnik, A. Vinel, and Y. Koucheryavy, "Exploiting bacterial properties for multi-hop nanonetworks," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 184–191, 2014.

[90] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Keseler, A. Kothari, A. Kubo *et al.*, "The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases," *Nucleic acids research*, vol. 42, no. D1, pp. D459–D471, 2014.

[91] M. Kanehisa, "Toward understanding the origin and evolution of cellular organisms," *Protein Sci*, vol. 28, p. 1947–1951, 2019, pubmed] [doi.

[92] M. Kanehisa, S. Goto, and K.E.G.G., "Kyoto encyclopedia of genes and genomes," *Nucleic Acids Res*, vol. 28, p. 27–30, 2000, pubmed] [doi.

[93] M. Kanehisa, M. Furumichi, Y. Sato, M. Ishiguro-Watanabe, and M. Tanabe, "Kegg: integrating viruses and cellular organisms," *Nucleic Acids Res*, vol. 49, p. 545– 551, 2021, pubmed] [doi.

[94] K. Oliphant and E. Allen-Vercoe, "Macronutrient metabolism by the human gut microbiome: major fermentation by-products and their impact on host health," *Microbiome*, vol. 7, no. 1, p. 91, 2019.

[95] F. S. Oliveira, J. Brestelli, S. Cade, J. Zheng, J. Iodice, S. Fischer, C. Aurrecoechea, J. C. Kissinger, B. P. Brunk, C. J. Stoeckert Jr *et al.*, "Microbiomedb: a systems biology platform for integrating, mining and analyzing microbiome experiments," *Nucleic acids research*, vol. 46, no. D1, pp. D684–D691, 2018.

[96] L. Wen and A. Duffy, "Factors influencing the gut microbiota, inflammation, and type 2 diabetes," *The Journal of nutrition*, vol. 147, no. 7, pp. 1468S–1475S, 2017.

[97] M. A. Henson and P. Phalak, "Microbiota dysbiosis in inflammatory bowel diseases: in silico investigation of the oxygen hypothesis," *BMC systems biology*, vol. 11, no. 1, p. 145, 2017.

[98] J. T. Bjerrum, Y. Wang, F. Hao, M. Coskun, C. Ludwig, U. Günther, and O. H. Nielsen, "Metabonomics of human fecal extracts characterize ulcerative colitis, crohn's disease and healthy individuals," *Metabolomics*, vol. 11, no. 1, pp. 122–133, 2015.

[99] Y. Janssens, J. Nielandt, A. Bronselaer, N. Debunne, F. Verbeke, E. Wynendaele, F. Van Immerseel, Y.-P. Vandewynckel, G. De Tré, and B. De Spiegeleer, "Disbiome database: linking the microbiome to disease," *BMC microbiology*, vol. 18, no. 1, pp. 1–6, 2018.

[100] J. Yang, P. Zheng, Y. Li, J. Wu, X. Tan, J. Zhou, Z. Sun, X. Chen, G. Zhang, H. Zhang *et al.*, "Landscapes of bacterial and metabolic signatures and their interaction in major depressive disorders," *Science advances*, vol. 6, no. 49, p. eaba8555, 2020.

[101] S. Kim, I. Thapa, L. Zhang, and H. Ali, "A novel graph theoretical approach for modeling microbiomes and inferring microbial ecological relationships," *BMC genomics*, vol. 20, no. 11, pp. 1–13, 2019.

[102] C. Huttenhower, D. Gevers, R. Knight, S. Abubucker, J. H. Badger, A. T. Chinwalla, H. H. Creasy, A. M. Earl, M. G. FitzGerald, R. S. Fulton *et al.*, "Hmp phase i (v3-v5)," 2012. [Online]. Available: https://microbiomedb.org/mbio/app/record/dataset/DS_3137502420

[103] J. Sung, S. Kim, J. J. T. Cabatbat, S. Jang, Y.-S. Jin, G. Y. Jung, N. Chia, and P.-J. Kim, "Global metabolic interaction network of the human gut microbiota for context-specific community-scale analysis," *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.

[104] J. VOHRADSKY, "Neural network model of gene expression," *the FASEB journal*, vol. 15, no. 3, pp. 846–854, 2001.

[105] D. C. Weaver, C. T. Workman, and G. D. Stormo, "Modeling regulatory networks with weight matrices," in *Biocomputing'99*. World Scientific, 1999, pp. 112–123.

[106] M. V. Grosso-Becerra, G. Croda-García, E. Merino, L. Servín-González, R. Mojica-Espinosa, and G. Soberón-Chávez, "Regulation of pseudomonas aeruginosa virulence factors by two novel rna thermometers," *Proceedings of the National Academy of Sciences*, vol. 111, no. 43, pp. 15 562–15 567, 2014.

[107] A. Ishihama, "Prokaryotic genome regulation: a revolutionary paradigm," *Proceedings of the Japan Academy, Series B*, vol. 88, no. 9, pp. 485–508, 2012.

[108] F. Spitz and E. E. Furlong, "Transcription factors: from enhancer binding to developmental control," *Nature reviews genetics*, vol. 13, no. 9, pp. 613–626, 2012.

[109] M. C. Davis, C. A. Kesthely, E. A. Franklin, and S. R. MacLellan, "The essential activities of the bacterial sigma factor," *Canadian journal of microbiology*, vol. 63, no. 2, pp. 89–99, 2017.

[110] E. Galán-Vásquez, B. C. Luna-Olivera, M. Ramírez-Ibáñez, and A. Martínez-Antonio, "Regulomepa: a database of transcriptional regulatory interactions in pseudomonas aeruginosa pao1," *Database*, vol. 2020, p. baaa106, 2020.

[111] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[112] M. Kanehisa, "Toward understanding the origin and evolution of cellular organisms," *Protein Science*, vol. 28, no. 11, pp. 1947–1951, 2019.

[113] M. Kanehisa, M. Furumichi, Y. Sato, M. Kawashima, and M. Ishiguro-Watanabe, "Kegg for taxonomy-based analysis of pathways and genomes," *Nucleic acids research*, vol. 51, no. D1, pp. D587–D592, 2023.

[114] I. M. Keseler, J. Collado-Vides, A. Santos-Zavaleta, M. Peralta-Gil, S. Gama-Castro, L. Muñiz-Rascado, C. Bonavides-Martinez, S. Paley, M. Krummenacker, T. Altman *et al.*, "Ecocyc: a comprehensive database of escherichia coli biology," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D583–D590, 2010.

[115] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko *et al.*, "Ncbi geo: archive for functional genomics data sets—update," *Nucleic acids research*, vol. 41, no. D1, pp. D991–D995, 2012.

[116] ——, "Ncbi geo: archive for functional genomics data sets—update," *Nucleic acids research*, vol. 41, no. D1, pp. D991–D995, 2012.

[117] V. H. Tierrafría, C. Rioualen, H. Salgado, P. Lara, S. Gama-Castro, P. Lally, L. Gómez-Romero, P. Peña-Loredo, A. G. López-Almazo, G. Alarcón-Carranza *et al.*, "Regulondb 11.0: Comprehensive high-throughput datasets on transcriptional regulation in escherichia coli k-12," *Microbial Genomics*, vol. 8, no. 5, 2022.

[118] X. Meng, S. D. Ahator, and L.-H. Zhang, "Molecular mechanisms of phosphate stress activation of pseudomonas aeruginosa quorum sensing systems," *MSphere*, vol. 5, no. 2, pp. 10–1128, 2020.

[119] A. Farzad, H. Mashayekhi, and H. Hassanpour, "A comparative performance analysis of different activation functions in lstm networks for classification," *Neural Computing and Applications*, vol. 31, pp. 2507–2521, 2019.

[120] A. A. Alkhouly, A. Mohammed, and H. A. Hefny, "Improving the performance of deep neural networks using two proposed activation functions," *IEEE Access*, vol. 9, pp. 82 249–82 271, 2021.

[121] Y. Qin, X. Wang, and J. Zou, "The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3814–3824, 2018.

[122] B.-Q. Zhang, Z.-Q. Chen, Y.-Q. Dong, D. You, Y. Zhou, and B.-C. Ye, "Selective recruitment of stress-responsive mrnas to ribosomes for translation by acetylated protein s1 during nutrient stress in escherichia coli," *Communications Biology*, vol. 5, no. 1, p. 892, 2022.

[123] H. E. Rivera, H. E. Aichelman, J. E. Fifer, N. G. Kriefall, D. M. Wuitchik, S. J. Smith, and S. W. Davies, "A framework for understanding gene expression

plasticity and its influence on stress tolerance," *Molecular Ecology*, vol. 30, no. 6, pp. 1381–1397, 2021.

[124] C.-E. Yin and G. Qu, "Obtaining statistically random information from silicon physical unclonable functions," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 96–106, 2014.

[125] J. Rojo, L. G. de Pinho, C. Fonseca, M. J. Lopes, S. Helal, J. Hernández, J. Garcia-Alonso, and J. M. Murillo, "Analyzing the performance of feature selection on regression problems: A case study on older adults' functional profile," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 1, pp. 137–152, 2023.