**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

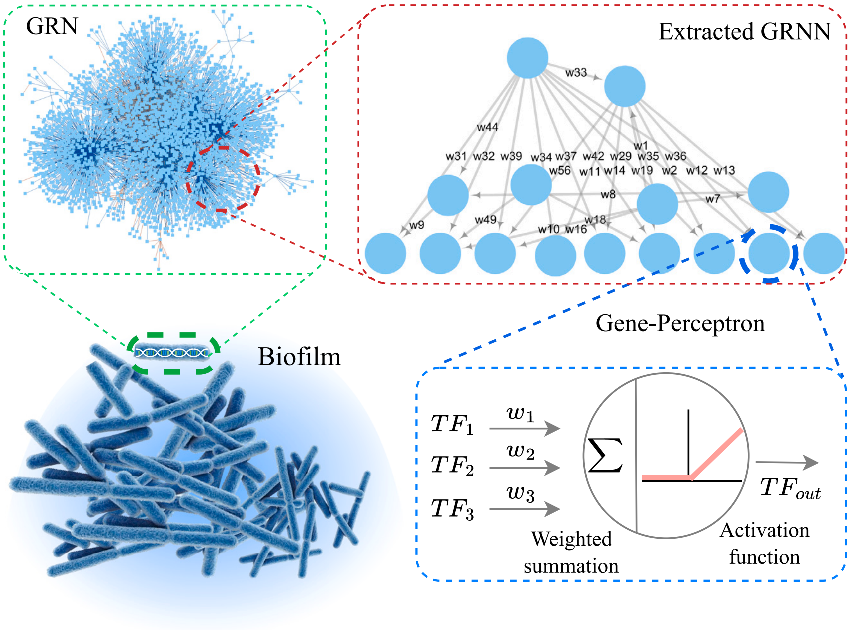


FIGURE 1 Illustration of a biofilm and the extraction of GRNNs from within the bacterial cells.

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

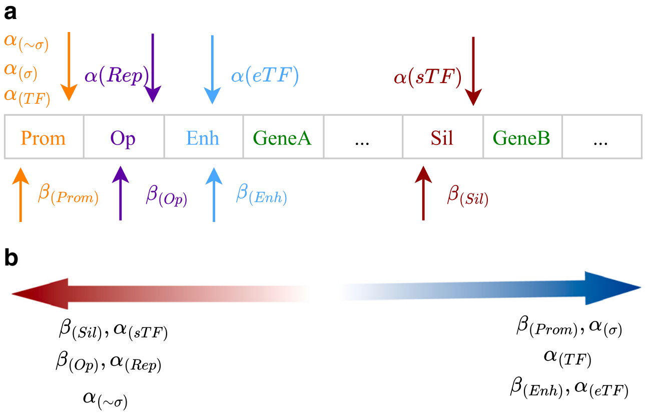


FIGURE 2 Illustration of gene expression regulators that are considered the weight influencers of the edges of GRNNs. Here, (*a*) shows the $\alpha_{(\sigma)}$, $\alpha_{(\sim\sigma)}$, $\alpha_{(TF)}$, $\alpha_{(Rep)}$, $\alpha_{(eTF)}$, and $\alpha_{(sTF)}$ are relative concentrations of sigma factors, 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

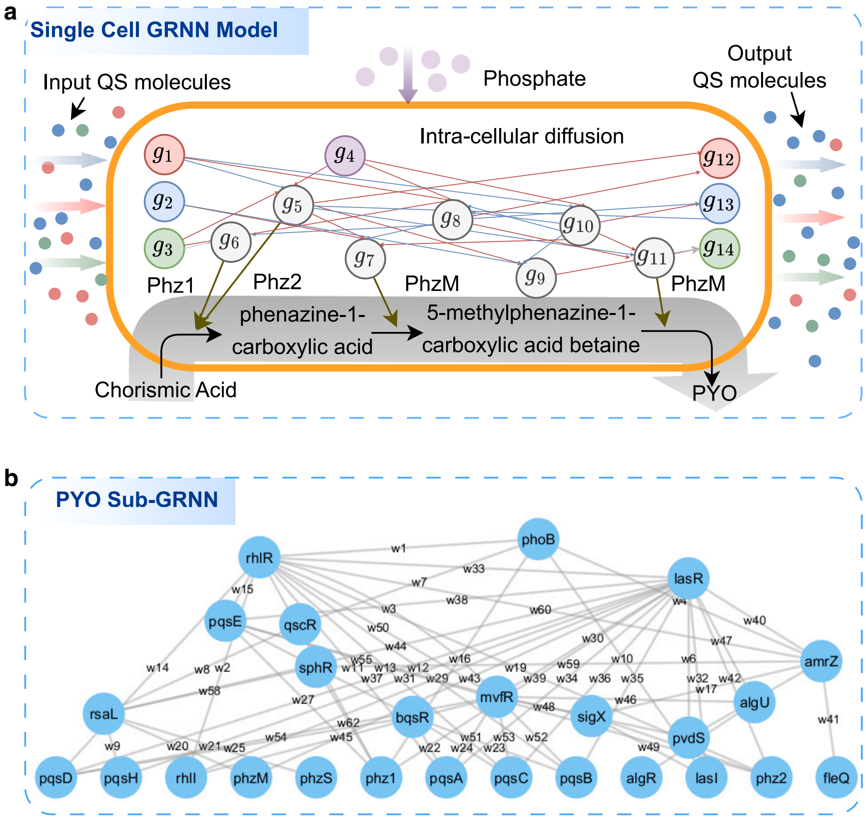


FIGURE 3 Illustration of cell-computing model of PYO production network where (*a*) depicts the computing process of the incoming cell-cell communication molecules and the conversion of chorismic acid to PYO that is driven by the GRNN outputs in response to phosphate input, whereas (*b*) shows the PYO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

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

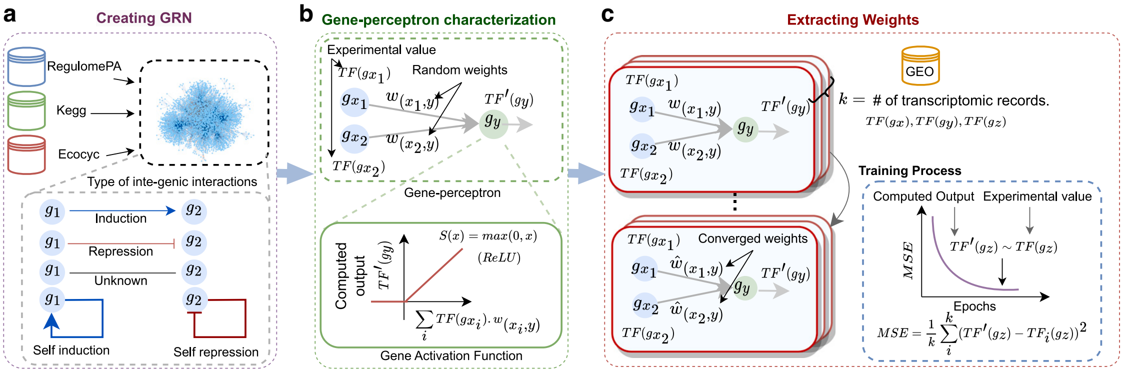


FIGURE 4 Illustration of GRNN extract steps where (*a*) is the creation of GRN structure with various interaction types between genes that is extracted from databases, (*b*) shows disassembling of the GRN into gene-perceptions with ReLU as the activation function, and (*c*) shows the weight extraction 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,

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

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

$$\mathrm{D} = \{D_{m_1}, D_{m_2}, \ldots, D_{m_Q}\}, \qquad (3)$$

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,

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

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,

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

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,

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

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,

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

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 $\mathrm{R}^{(t+1)}_P$ 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,

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

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,

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

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,

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

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,

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

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,

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

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,

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

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

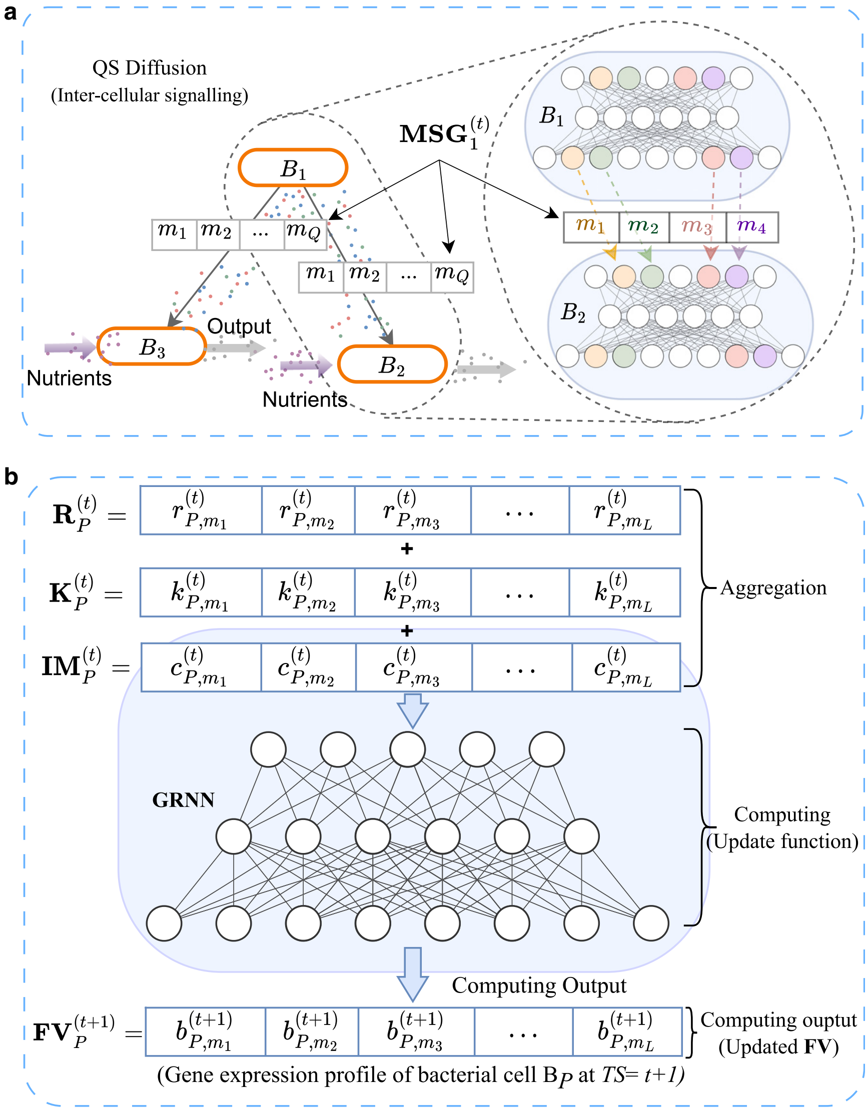


FIGURE 5 The process of one GRNN outputs reaching another GRNN as molecular messages where (*a*) illustrates how the GRNN output of one cell influences the others and (*b*) explains how the graph neural networks are utilized to model the computing of bacterial cells.

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

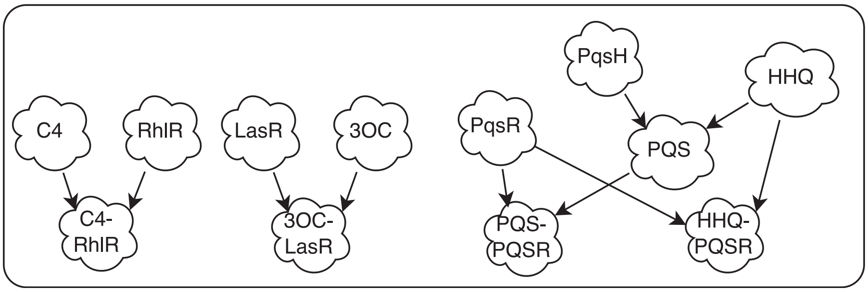


FIGURE 6 Illustrations of intra-cellular metabolite interaction where the QS molecules form complexes with response regulators.

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

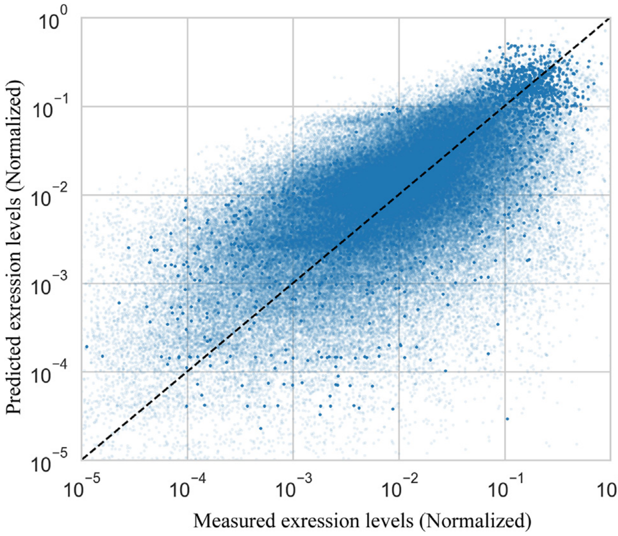


FIGURE 7 Comparison between measured expression levels of 2851 genes for 217 transcription records and gene expression values computed by the extracted full GRNN.

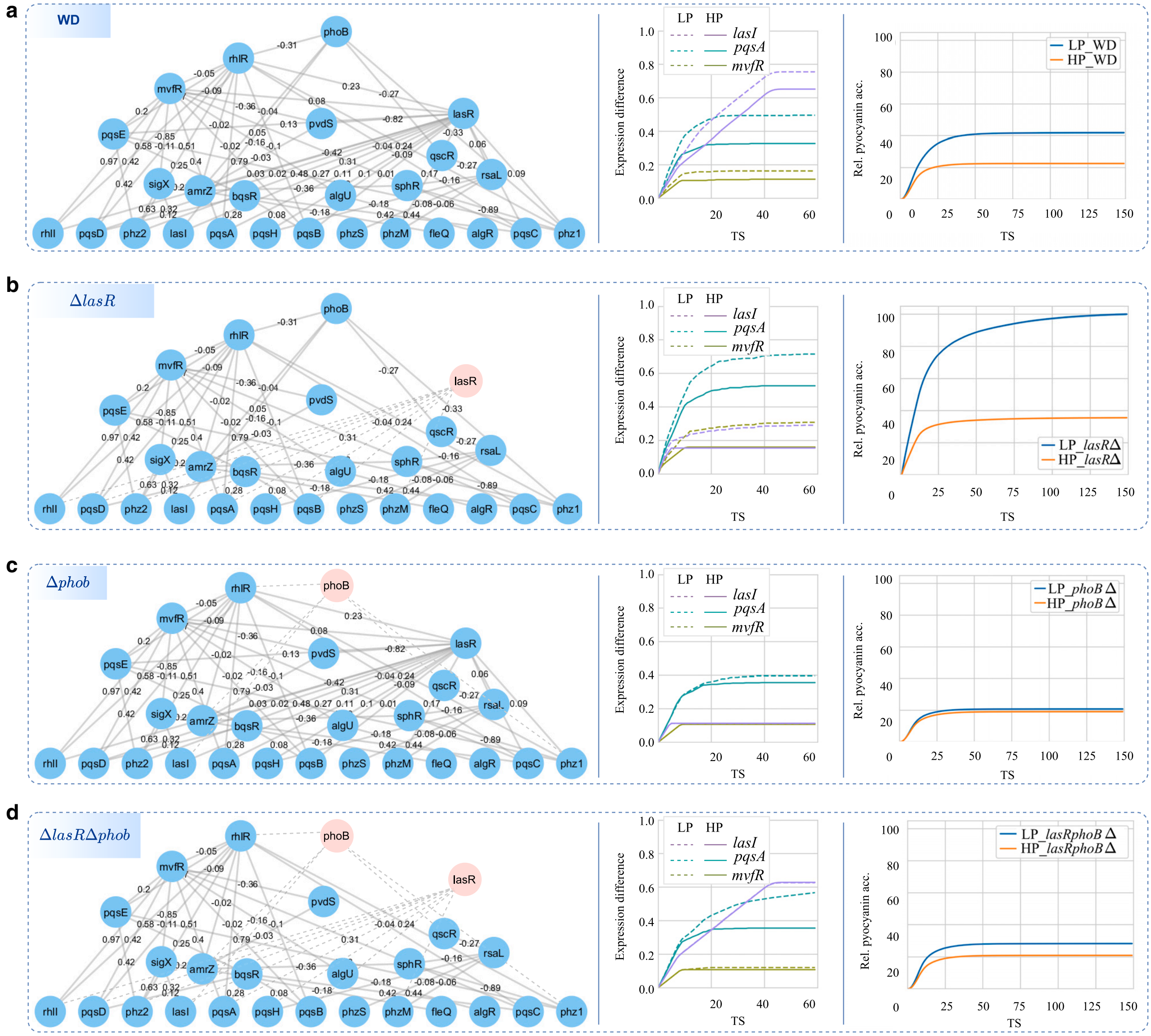


FIGURE 8 Mutagenesis analysis to investigate the impact of GRNN structural deviation on the PYO production. The gene expression variations and resulting PYO production are shown in LP and HP for four GRNN structures of (*a*) WD, (*b*) Δ*lasR*, (*c*) Δ*phoB*, and (*d*) Δ*lasR*Δ*phoB*. The genes highlighted 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 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.

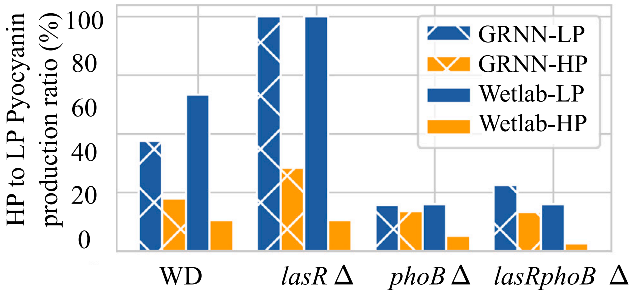


FIGURE 9 Evaluation of the model accuracy by comparing HP to LP PYO production ratio with wet-lab data from (73).

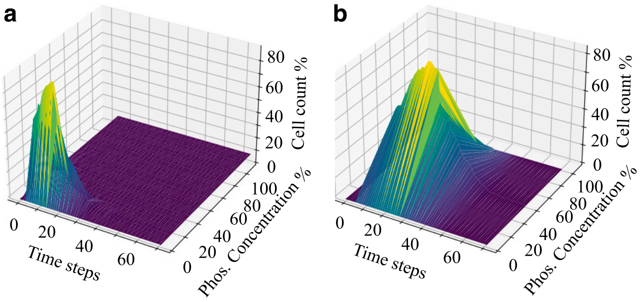


FIGURE 10 The nutrient accessibility variations of cells is expressed in two different environment conditions: (*a*) LP and (*b*) HP concentrations.

To 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 $MI$;
$MI = \mathrm{MutualInformation}(f_{XY}XY, f_X X, f_Y)$;
return $MI$;
```

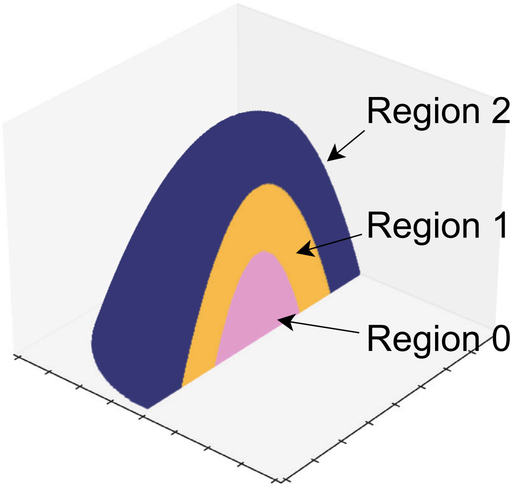


FIGURE 11 Illustration of three layers considered in the biofilm to investigate the computing reliability and the solution space, where region 2 is the outer layer, which has the most access to nutrients. Region 1 is the middle layer, and region 0 is the core with the least nutrient access.

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

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.

the lowest MI values as the information flow of GRNN 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 $S(x)$,

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

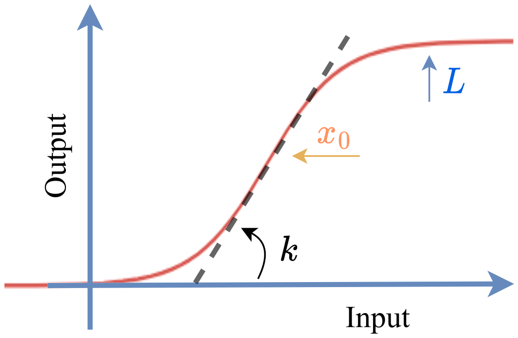


FIGURE 13 Illustration of $L$, $k$, and $x_0$ parameters that determine the height, steepness, and horizontal shift of the sigmoid curve, respectively.

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 nonlinearity 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
      $\hat{Y} = \mathrm{MSigmoid}(X, \mathbf{P})$;
    Update the parameter values
      $\mathbf{P} = \mathrm{LeastSquares}(X, Y, \hat{Y})$;
  until convergence;
  return $\mathbf{P}$;
Function MSigmoid $(X, P)$:
  Extract parameter values $L, k, x_0$ from $\mathbf{P}$;
  Compute the predicted values $\hat{Y}$ using the sigmoid
    function: $\hat{Y} = \frac{L}{1+e^{-(kX-x_0)}}$;
  return $\hat{Y}$;
Function LeastSquares $(X, Y, \hat{Y})$:
  Compute the residual vector $R = Y - \hat{Y}$;
  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 $\mathbf{P}$;
$\mathbf{P} = \mathrm{CurveFitting}\ (X, Y)$;
```

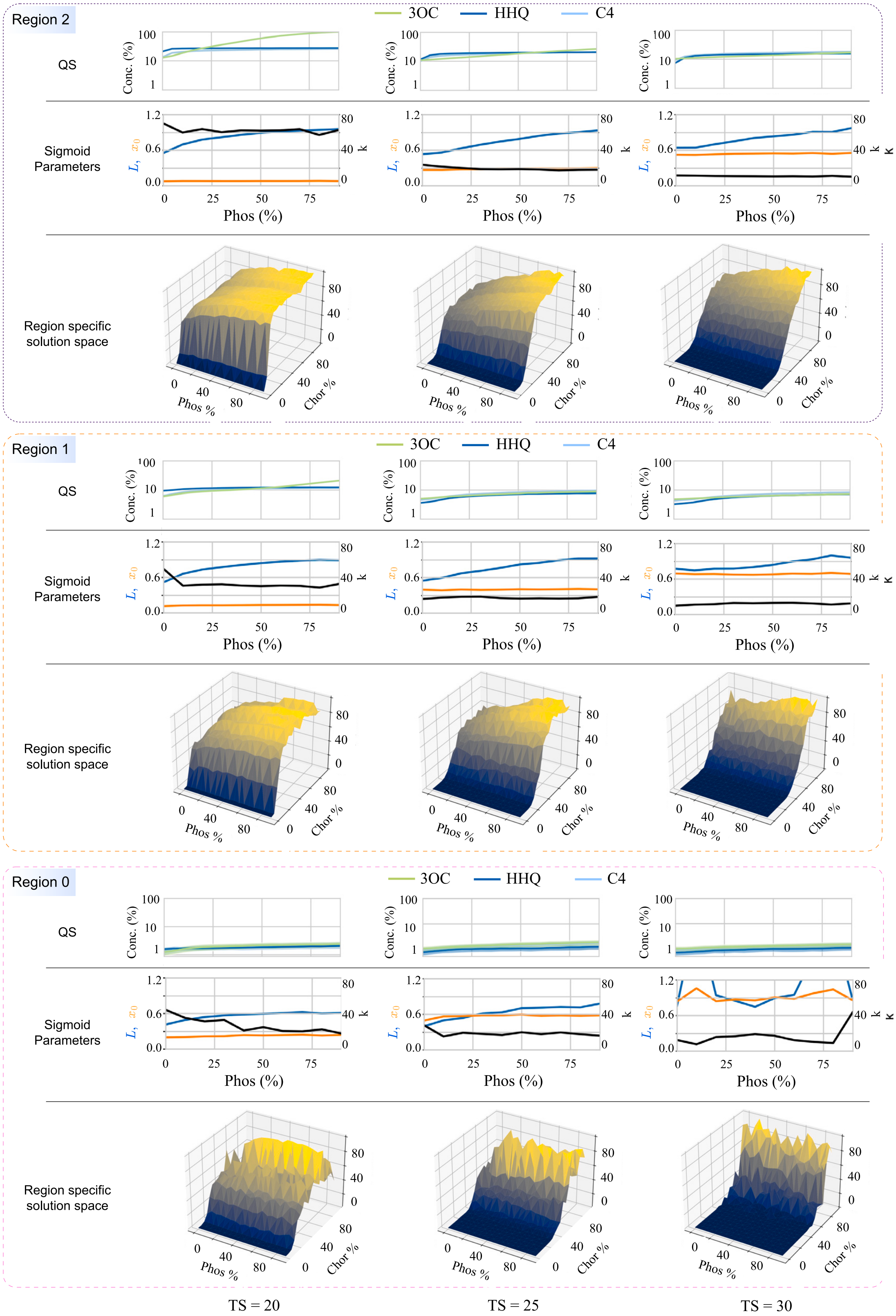
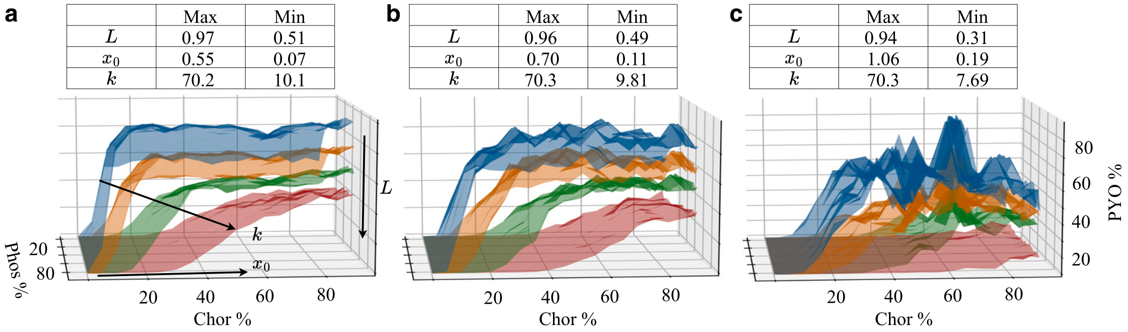


FIGURE 14 Illustration of a sigmoid function-based solution space found in biofilms, where variations in non-linear behavior are shown with respect to location in each column and time in each row. Each layer of the diagram consists of QS, sigmoid parameter, and sigmoid curve variation plots corresponding 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.



a

| | Max | Min |
|---|---|---|
| $L$ | 0.97 | 0.51 |
| $x_0$ | 0.55 | 0.07 |
| $k$ | 70.2 | 10.1 |

b

| | Max | Min |
|---|---|---|
| $L$ | 0.96 | 0.49 |
| $x_0$ | 0.70 | 0.11 |
| $k$ | 70.3 | 9.81 |

c

| | Max | Min |
|---|---|---|
| $L$ | 0.94 | 0.31 |
| $x_0$ | 1.06 | 0.19 |
| $k$ | 70.3 | 7.69 |

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.