Biophysical Reports

ARTICLE

Nonlinear classifiers for wet-neuromorphic computing using gene regulatory neural network

Adrian Ratwatte,[1,*] Samitha Somathilaka,[1,2] Sasitharan Balasubramaniam,[1] and Assaf A. Gilad[3,4]
[1]School of Computing, University of Nebraska-Lincoln, 104 Schorr Center, Lincoln, Nebraska, USA; [2]VistaMilk Research Centre, Walton Institute for Information and Communication Systems Science, South East Technological University, Waterford, Ireland; [3]Department of Chemical Engineering and Materials Science, Michigan State University, East Lansing, Michigan, USA; and [4]Department of Radiology, Michigan State University, East Lansing, Michigan, USA

ABSTRACT The gene regulatory network (GRN) of biological cells governs a number of key functionalities that enable them to adapt and survive through different environmental conditions. Close observation of the GRN shows that the structure and operational principles resemble an artificial neural network (ANN), which can pave the way for the development of wet-neuromorphic computing systems. Genes are integrated into gene-perceptrons with transcription factors (TFs) as input, where the TF concentration relative to half-maximal RNA concentration and gene product copy number influences transcription and translation via weighted multiplication before undergoing a nonlinear activation function. This process yields protein concentration as the output, effectively turning the entire GRN into a gene regulatory neural network (GRNN). In this paper, we establish nonlinear classifiers for molecular machine learning using the inherent sigmoidal nonlinear behavior of gene expression. The eigenvalue-based stability analysis, tailored to system parameters, confirms maximum-stable concentration levels, minimizing concentration fluctuations and computational errors. Given the significance of the stabilization phase in GRNN computing and the dynamic nature of the GRN, alongside potential changes in system parameters, we utilize the Lyapunov stability theorem for temporal stability analysis. Based on this GRN-to-GRNN mapping and stability analysis, three classifiers are developed utilizing two generic multilayer sub-GRNNs and a sub-GRNN extracted from the *Escherichia coli* GRN. Our findings also reveal the adaptability of different sub-GRNNs to suit different application requirements.

WHY IT MATTERS In recent years the significance of artificial intelligence has been steadily rising, driven by the development of numerous algorithms that are applicable across various domains. As we envision a future of "*AI everywhere*," we are faced with the prospects of applying AI into media that is beyond silicon technology, such as wet biological environments. In this study our objective is to propose a paradigm of biological AI that is built from the gene regulatory process. Realizing a vision of wet-neuromorphic computing systems can result in novel theranostic applications for disease detection and treatment as well as new bio-hybrid computing systems that integrate biological cells with silicon technology.

INTRODUCTION

In recent years, the field of artificial intelligence (AI) has developed rapidly resulting in sophisticated learning algorithms that have benefited a plethora of applications (e.g., manufacturing, economics, computer vision, robotics, etc.) (1,2). Inspired by the functions of neurons, the ultimate vision of AI is to create human-like intelligence that one day will have a working capacity close to the brain. Based on the system applications, AI can be categorized into software or hardware based. Software-based AI includes various forms of algorithms that depend on their structure as well as training process (e.g., convolutional neural networks (3), recurrent neural networks (4), where a novel application is large language models such as generative pre-trained transformer (5).

Neuromorphic computing is a hardware-based AI platform that architecturally consists of neurons and synapses constructed from memristor devices that communicate based on encoded neural spikes (6). Presently, the vast majority of AI machines are constructed using instruction-encoded circuits and

Submitted December 22, 2023, and accepted for publication May 31, 2024.

*Correspondence: aratwatte2@huskers.unl.edu

Editor: Yoav Shechtman.

https://doi.org/10.1016/j.bpr.2024.100158

silicon-based semiconductors and nanotechnology (7–9). While this enables more efficient computer systems that have capabilities of learning and computing, it also results in significant challenges such as deployments in mediums that support silicon technologies (e.g., biological mediums), as well as utilizing large amounts of energy (10). Building upon prior research in neuromorphic hardware, our focus extends to explore the wet-neuromorphic computing properties using the nonlinear dynamics of gene expression and the molecular communication processes observed within bacterial cells.

Current research has aimed to address these challenges and one direction taken is through biological AI, where computing is performed through living biological cells (11,12). A recent example is the *DishBrain*, where the system is composed of living neurons that can be trained to play the game of "*Pong*" on a computer (13). In other works, artificial neural networks (ANNs) have been programmed into bacterial cells (14,15). Similarly, molecular circuits programmed to behave like ANN have also been proposed, and one example is the biomolecular neural network (16). The underlying basis for all these approaches is the communication of molecules (17,18) that operate as part of the chemical reactions to enable computing operations.

From the perspective of gene regulatory networks (GRN), there has been a connection between its structure and the operation of an ANN. In our recent work (19), we developed a model that transforms the gene-gene interaction within the GRN using weights, forming a gene regulatory neural network (GRNN) while also exploring the impact of structural changes on the computing capacity. Fig. 1 illustrates the mapping from ANN to GRNN. In a conventional ANN, a perceptron takes multiple inputs (x_1 and x_2) and computes their weighted summation $(\sum)$ that goes through an activation function $(z(x))$ (20). In the context of the GRNN, the weights are embodied by transcription factor (TF) concentration, which corresponds to the half-maximal RNA concentration (K_A), and gene product copy number (C_N). These factors individually contribute to the RNA and protein concentrations, reflecting a linear combination in the logarithmic domain, which is equivalent to the multiplication of weighted inputs (21). Input genes (g_{X_1} and g_{X_2}) possess TFs that bind to the promoter region of gene-perceptron $g_{1,i}$, which subsequently transcribes RNA (R_i) at a rate of k_1 and degrades at a rate of d_1. The subsequent step involves translation into protein at a rate of k_2 and degradation at a rate of d_2 through a nonlinear activation function, resulting in the output of maximum-stable protein concentration $[P]^*$ at equilibrium.

In this study, we mathematically model chemical reactions of the transcription and translation process of a gene-perceptron, which we term as the dual-layered transcription-translation reaction model (from here on we simply term this as dual-layered chemical reaction

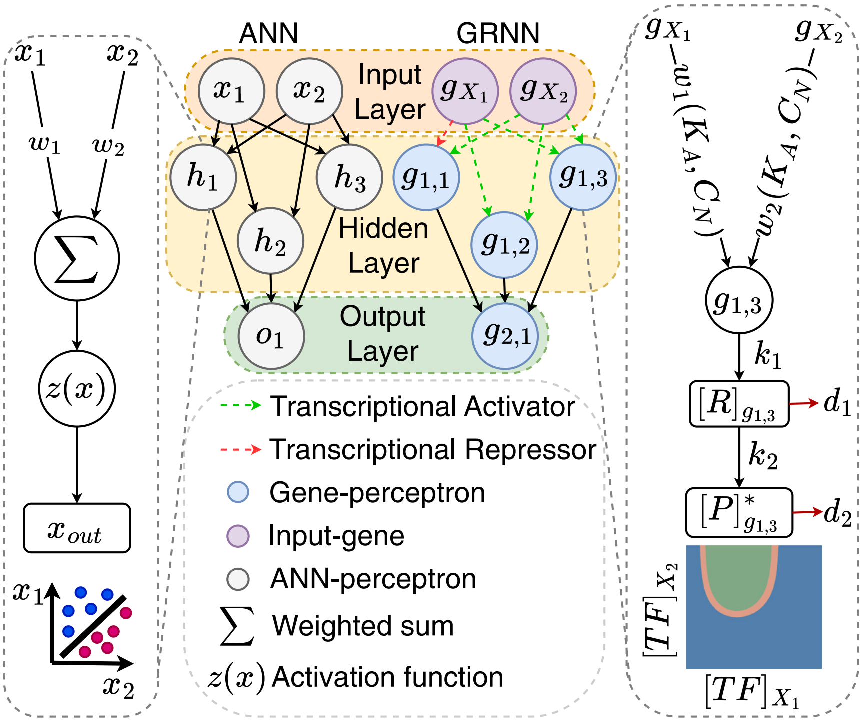

FIGURE 1 Illustration of mapping between components of ANN to GRNN. In this depiction, w_i and $w_i(K_A, C_N)$ represent the weights of a perceptron in ANN and GRNN, respectively, while activation function $z(x)$ is equivalent to a combination of the transcription process of RNA concentration $[R]_i$ as well as translation of maximum-stable protein concentration $[P]^*_i$. The chemical reactions are governed by the transcriptions rate k_1, translation rate k_2, degradation rate of RNA d_1, and degradation rate of protein d_2.

model). The dual-layered chemical reaction model can be integrated with a trans-omic data model (transcriptome and proteome) and the cellular GRN in order for us to identify active genes for the specific environments, which will be the basis for us to create the GRNN.

We investigate the behavior of fully connected sub-GRNNs derived from a larger GRN, aiming to showcase how genes function as perceptrons which we term as gene-perceptron, ensuring reliable computing in a stable state. Additionally, we use extracted sub-GRNNs to explore different nonlinear classifiers that can be applicable to diverse applications. This investigation primarily focuses on the gene expression-level stability of the translation and transcription process to ensure reliable computing operation. Once the gene-perceptron reaches stability, its output can be represented by the maximum-stable protein concentration $([P]^*)$. The stability of the gene-perceptron is characterized as the point where RNA and protein concentrations peak and stabilize over time in a sigmoidal manner. The eigenvalue-based method and Lyapunov stability theorem are established tools in systems biology (22) and for assessing overall system energy (23). However, we choose to utilize the Lyapunov stability for the size of the network given that previous studies have used it for small NNs (24), and we see this appropriate for our sub-GRNN analysis. The eigenvalue-based stability analysis verifies the existence of upper bounds of transcription and translation guaranteeing minimal concentration fluctuations that can reduce potential computing errors. Given the dynamic nature of the GRN and the variability of temporal system parameters, the eigenvalue-based method lacks critical information on the time interval during which the gene-perceptron gradually gets close to equilibrium (25–27), which is crucial for GRNN computing. Therefore, we opt to utilize the Lyapunov stability theorem, specifically Lyapunov's second method for stability, in our analysis (28).

Once we prove the stability of the gene-perceptron, as an application we focus on a nonlinear classifier relying on the maximum-stable protein concentration for different concentrations of TFs that act as inputs. We concentrate on nonlinear classification by utilizing the inherent shifted sigmoidal behavior (19,21) as an activation function, observed in the relationship between the input and output gene concentrations. This behavior, reminiscent of the Hill function and influenced by the nonnegative nature of expression values, prompts our investigation into the potential of nonlinear classification using GRNNs. Additionally, certain real-world problems exceed the limitations of linear methods confined to 2D decision planes. To evaluate the model's performance, we analyze two generic multilayer sub-GRNNs and an *Escherichia coli* sub-GRNN. By manipulating parameters within the Hill function, such as TF concentration corresponding to half-maximal RNA concentration and Hill coefficient, we showcase the potential to shift the classification area. Previous research indicates that adjusting these parameters can effectively alter the nonlinearity of the sigmoid activation function (21,29,30), providing a valuable tool for engineering the GRN through synthetic biology approaches and employing various sub-GRNNs for diverse nonlinear classifiers tailored to specific application needs.

The contributions of this study can be outlined as follows.

- *Developing GRNNs inspired from ANN structures using dual-layer chemical reaction models.* Using the dual-layered chemical reaction model, we show that gene transcription and RNA translation processes exhibit sigmoidal-like molecular concentration dynamics at their stable points. This behavior is governed by the weights, which is a function of gene product copy number and TF concentration corresponding to the half-maximal RNA concentration.
- *Stability analysis of GRNN.* We developed a full mathematical model derived from the chemical reactions and apply Lyapunov's stability theorem (28) for the gene-perceptron to determine temporal stability that will facilitate reliable GRNN computing.
- *GRNN application for nonlinear classifiers*. Using the mapping of GRN-to-GRNN and the stability analysis, we are able to determine the decision boundaries of the derived sub-GRNNs to classify data within regions of protein concentration output. By varying parameters of the chemical reactions, we demonstrate how the classification area can be shifted, which can serve as a tool for engineering the GRN and using different sub-GRNNs for several nonlinear classifiers based on the application's requirements.

System modeling

This section describes the mathematical models for the gene transcription and translation within gene-perceptrons, employing a dual-layered chemical reaction model (Fig. 2) that breaks down the steps of the translation and transcription process. The production of RNAs depends on RNA polymerase, TFs, and σ factors that binds to the promoter $(Prom)$ (31), as well as the dissociation constant (k_A). Once the TF binds to the promoters *Prom*, the transcription begins at the rate of k_1. This is followed by the RNA degradation at the rate of d_1 based on their half-life value (32) and RNA binding proteins (33), as well as the degradosome components that includes *RNase E*, *RNA helicase*,

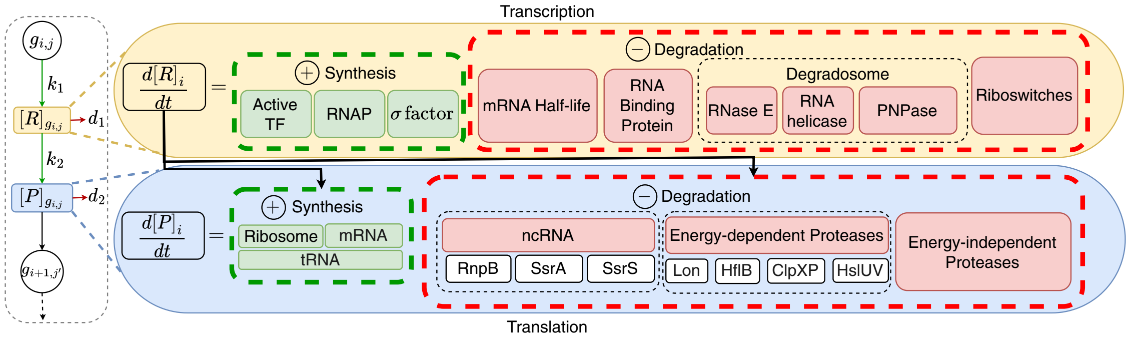

FIGURE 2 Illustration of dual-layered transcription-translation chemical reaction model of the gene-perceptron. Each component corresponds to the synthesis and degradation of RNA and protein for the j^{th} gene-perceptron in the i^{th} layer ($g_{i,j}$) of the GRNN. Here, *RnpB*, *SsrA*, and *SsrS* are examples for noncoding RNA (ncRNA). Examples of energy-dependent proteases include *Lon*, *HflB*, *ClpXP*, and *HslUV*. Active TF, RNAP, PNPase, RNase E, and tRNA correspond to active TFs, RNA polymerase, polyribonucleotide phosphorylase, ribonuclease E, and transfer RNA, respectively.

and *PNPase* (34). Following the transcription of the RNAs is the translation into protein, which occurs at the rate of k_2 facilitated by ribosome and transfer RNA (tRNA) (35). Once the RNA is translated, the protein molecules start to degrade gradually at the rate of d_2. Significant factors that affect the degradation of protein are noncoding RNA, as well as energy-dependent and energy-independent proteases. Overall, to maintain the concentration stability in the cell, RNA and protein production are balanced by the degradation process.

By taking the dual-layered chemical reactions model into account, we model the concentration changes at the transcriptome and proteome using mathematical models. These models enable us to assess the concentration stability of the gene-perceptron through the eigenvalue method and determine the stabilization time using the Lyapunov stability theorem. After determining if a particular gene-perceptron expression is stable, we determine the stability of the entire sub-GRNN. Then, based on the application study, the classification ranges for each gene-perceptron in a sub-GRNN is determined at the equilibrium maximum-stable protein concentration state. Based on the sigmoidal input-output behavior and adjustable threshold, we deduce that gene-perceptrons in the GRNN consist of conventional NN properties. For the overview of the algorithm mentioned above, please refer to Fig. 3.

Modeling transcription of a gene

In this section, we discuss transcription and the corresponding RNA concentration model. During the transcription process, the RNA polymerase and TFs bind to the promoter region and then the σ factor attaches to the promoter region and unwinds the DNA (36). This is followed by σ factor release from the polymerase, allowing for the elongation of the RNA chain. Based on (37), the concentration change over time t of RNA for a particular gene-perceptron i can be expressed as follows (chemical species are represented using uppercase letters [e.g., X], and their corresponding concentration is enclosed within brackets [e.g., $[X]$])

$$\frac{d[R]_i}{dt} = k_{1_i} C_{N_i} \frac{[TF]^n}{K_{A_i}^n + [TF]^n} - d_{1_i}[R]_i. \qquad \text{(Equation 1)}$$

The gene-perceptron is activated by the TF, where $[R]_i$, k_{1_i}, $[TF]$, d_{1_i}, n, C_{N_i}, and K_{A_i} are the RNA concentration, transcription rate, concentration of TFs, degradation rate of RNA, Hill coefficient, gene product copy number, and TF concentration when the production of RNA is at the half-maximal point for gene-perceptron i, respectively. The gene product copy number denotes the average count of protein molecules generated by an mRNA throughout its lifespan (22).

Given the initial RNA concentration transcribed by a gene-perceptron is $[R]_i(0)$ (i.e., $[R]_i(t = 0) = [R]_i(0)$), the solution of Eq. 1 is derived as follows

$$[R]_i = \frac{k_{1_i} C_{N_i}}{d_{1_i}} \left(\frac{[TF]^n}{[TF]^n + K_{A_i}^n} \right) \left(1 - e^{-d_{1_i} t} \right) + [R]_i(0) e^{-d_{1_i} t}. \qquad \text{(Equation 2)}$$

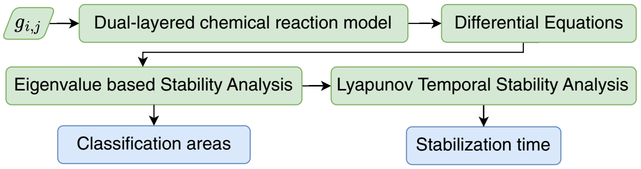
$g_{i,j}$
Dual-layered chemical reaction model
Differential Equations
Eigenvalue based Stability Analysis
Lyapunov Temporal Stability Analysis
Classification areas
Stabilization time

FIGURE 3 Flow chart for the calculation of classification areas as well as stability based on the dual-layered transcription-translation chemical reaction model of each gene-perceptron.

In contrast, in the event that the gene-perceptron is repressed by the TF, the RNA concentration changes over time t (37) is represented as follows,

$$\frac{d[R]_i}{dt} = k_{1_i} C_{N_i} \frac{K_{A_i}^n}{K_{A_i}^n + [TF]^n} - d_{1_i} [R]_i. \qquad \text{(Equation 3)}$$

Equations 1 and 3 are expressed as a mass balance differential equation with the difference between the RNA synthesis, which is modeled using the Hill function integrated with the degradation process of the RNA (22,38–40). The Hill coefficient n represents the number of TF molecules that bind simultaneously to the promoter $Prom$ with K_A dissociation constant when the gene-perceptron is transcribing RNA (37) and is represented as $Prom + n\,TF \overset{K_A}{\rightleftharpoons} Prom_{n.TF}$. The Hill coefficient is critical for the sigmoidal input-output characteristics of the gene-perceptron, as depicted in Fig. 4. According to the plot, we can see that, when we increase the Hill coefficient, the sigmoidicity increase for the maximum-stable protein concentration ($[P]^*$) over the input-gene concentration ($[TF]$). Thus, when a gene-perceptron possesses a higher Hill coefficient, it exhibits more sigmoidal-like behavior. For our analytical model we consider $n = 1$.

Modeling translation of a RNA

In this section, we describe RNA-to-protein translation and associated models. Initially, the ribosome and tRNAs form a complex that draws the amino acids in the polypeptide chain to attach to the first codon position of the RNA (41). This is followed by the tRNAs adding amino acids one by one to form a polypeptide chain while moving along the RNA (42). Once the stop codon is detected, the polypeptide chain is released, dissociating the ribosome complex from the RNA and forming the protein (43). This process can be summarized through the protein concentration change over time (37), and is modeled as follows for a particular gene-perceptron i:

$$\frac{d[P]_i}{dt} = k_{2_i}[R]_i - d_{2_i}[P]_i, \qquad \text{(Equation 4)}$$

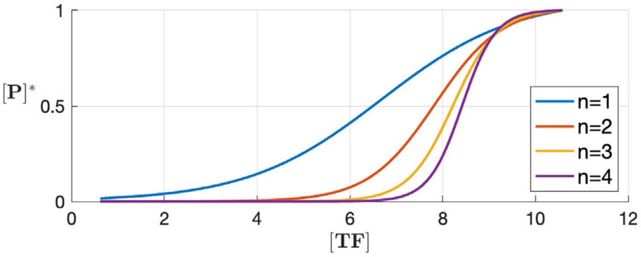

FIGURE 4 Sigmoidicity fluctuations for different Hill coefficients.

where $[P]_i$, k_{2_i}, and d_{2_i} are the protein concentration, translation rate, and degradation rate of protein for gene-perceptron i. Moreover, $[R]_i$ is the concentration of RNA from Eq. 1, and the TF activates the gene-perceptron i based on Eq. 3 if the TF represses the gene-perceptron. Similar to Eqs. 1 and 3, Eq. 4 is modeled based on mass-balance differential equation taking the difference between the RNA produced at the transcriptome level, which is translated into protein at the rate of k_{2_i} and the amount of protein that is degraded at the rate of d_{2_i} due to the factors presented in Fig. 2. Provided that the initial protein concentration translated by a RNA for gene-perceptron i is $[P]_i(0)$ (i.e., $[P]_i(t = 0) = [P]_i(0)$), the solution of Eq. 4 is given by

$$\begin{aligned}
[P]_i = {} & \frac{k_{1_i} k_{2_i} C_{N_i}}{d_{1_i}} \left(\frac{[TF]^n}{[TF]^n + K_{A_i}^n} \right) \left(\frac{1}{d_{2_i}} - \frac{e^{-d_{1_i} t}}{d_{2_i} - d_{1_i}} \right) \\
& + [R]_i(0) k_{2_i} \left(\frac{e^{-d_{1_i} t}}{d_{2_i} - d_{1_i}} \right) + e^{-d_{2_i} t} [P]_i(0) - e^{-d_{2_i} t} \\
& \times [R]_i(0) k_{2_i} \frac{1}{(d_{2_i} - d_{1_i})} - e^{-d_{2_i} t} \frac{k_{1_i} k_{2_i} C_{N_i}}{d_{1_i}} \\
& \left(\frac{[TF]^n}{[TF]^n + K_{A_i}^n} \right) \times \left(\frac{1}{d_{2_i}} - \frac{1}{(d_{2_i} - d_{1_i})} \right).
\end{aligned} \qquad \text{(Equation 5)}$$

METHODS

This section introduces the mathematical models for the stability analysis and RNA/protein concentration changes over time, and subsequently demonstrates how to apply these mathematical models in the GRNNs.

Gene expression stability analysis

In this section, we discuss the approach toward analyzing the expression-level stability of the gene-perceptron. Our view of the stability of the gene-perceptron is when the RNA transcription as well as the protein translation concentrations reach maximum over time and remain stable at that level exhibiting a sigmoidal behavior. To confirm the existence of transcription and translation upper bounds, we use eigenvalue-based stability analysis. This, in turn, ensures a stable classification region of the GRNN due to a protein concentration with minimum fluctuations that can result in minimized computing errors. Moreover, considering the dynamic characteristics of the GRN and its parameters, we explore the time necessary for GRNN stability utilizing the Lyapunov function, which is a crucial aspect of analyzing the reliability duration of the GRNN computing.

Stability of gene-perceptron based on eigenvalues

The algorithm for analyzing the stability of gene-perceptrons through the eigenvalue method is outlined in Algorithm 1 in the appendix, which initiates with the input of RNA and protein concentration changes $f([R]_i, [P]i)$, $g([R]i, [P]i)$ as expressed in Eqs. 1 and 4, alongside protein degradation rate d_{2_i}, RNA degradation rate d_{1_i}, and translation rate k_{2_i} for the i^{th} gene-perceptron. Although this study has only considered the case of gene transcription in Eq. 1, our approach is also applicable for the repression process defined in Eq. 3. Since we are analyzing the stability of the gene-perceptron at the equilibrium point, we can represent the maximum-stable RNA $[R]_i^*$ and protein $[P]_i^*$ concentration as follows:

$$[R]_i^* = \frac{k_{1_i} C_{N_i}}{d_{1_i}} \left(\frac{[TF]^n}{[TF]^n + K_{A_i}^n} \right), \qquad \text{(Equation 6)}$$

$$[P]_i^* = \frac{k_{1_i} k_{2_i} C_{N_i}}{d_{1_i} d_{2_i}} \left(\frac{[TF]^n}{[TF]^n + K_{A_i}^n} \right). \qquad \text{(Equation 7)}$$

The maximum-stable RNA and protein concentrations are determined for different TF concentrations.

To determine the eigenvalues of Eqs. 1 and 4 at the equilibrium points of Eqs. 6 and 7, we use the Jacobian matrix given in Eq. 24 (please see Algorithm 1 in the appendix). Hence, the eigenvalues are $\lambda_1 = -d_{1_i}$ and $\lambda_2 = -d_{2_i}$. Stability is confirmed when both eigenvalues are negative ($d_{1_i}, d_{2_i} > 0$), indicating that deviations from equilibrium will diminish over time, leading the system back to its stable state. Conversely, if any eigenvalue is nonnegative, the gene-perceptron is deemed unstable, signifying that deviations from equilibrium may escalate, impeding the system's return to its initial state. Thus, the stability of the gene-perceptron hinges on system parameters, specifically the RNA and protein degradation rates (d_{1_i} and d_{2_i}) within the GRN.

Stability of a gene-perceptron using the Lyapunov function

The Lyapunov function algorithm for temporal stability analysis is detailed in Algorithm 2 in the appendix. Initially, we define the Lyapunov function ($V([R]_i, [P]_i)$) as given by Eq. 25, which satisfies the necessary conditions: $V([R]_i, [P]_i) = 0$ when $[R]_i = [R]_i^*$ and $[P]_i = [P]_i^*$, where $[R]_i^*$ and $[P]_i^*$ are RNA and protein concentration at the equilibrium. In addition, $V([R]_i, [P]_i) > 0$ due to the quadratic nature of all terms. Finally, we consider the first derivative of Eq. 25 ($\frac{dV}{dt}$) as given by Eqs. 26 and 27, as the last condition to be satisfied for the stability of the gene-perceptron. We use Eq. 27 to determine the time during which the gene-perceptron gets closer to equilibrium, where we set this threshold as $\dot{V}_{th}$. In all our simulations, this threshold marks the time point $t = \bar{t}$ when $|\frac{dV}{dt}|$ is significantly minimized, beyond which $\frac{dV}{dt}$ continues to converge toward zero with minimal change. The gene-perceptron moves closer to the equilibrium rapidly when $\frac{dV}{dt} < \dot{V}_{th}$ and at a slower pace when $\frac{dV}{dt} \geq \dot{V}_{th}$.

We illustrate the temporal fluctuation of Eq. 27 in all simulations, as shown in Fig. 5, providing insights into the dynamic stability behavior of the gene-perceptron and delineating the time frame during which the gene-perceptron gets closer to the equilibrium. The gene-perceptron accelerates toward equilibrium within the time window $t \leq 5$, demonstrated by the negative amplitude of $\frac{dV}{dt}$, and subsequently slows this tendency after $t \approx 5$ while continuing to progress gradually toward equilibrium. The orange dashed line in the figure marks the time step ($\bar{t} \approx 18$), where $\frac{dV}{dt} = \dot{V}_{th}$, indicating that the gene-perceptron gets closer to equilibrium within the temporal window ($t \leq 18$).

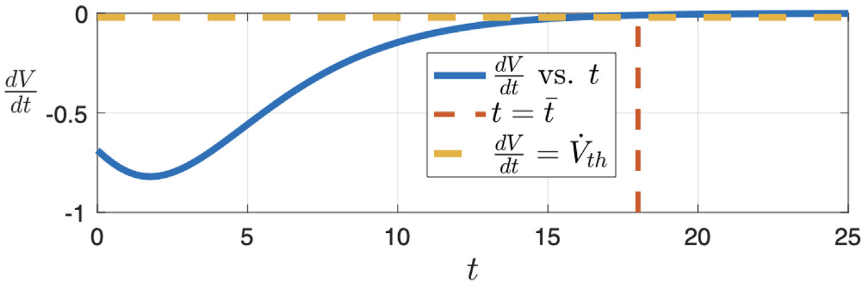

FIGURE 5 Temporal stability of a gene-perceptron based on the derivative of the Lyapunov function with respect to time. A threshold $\dot{V}_{th}$ is defined where $|\frac{dV}{dt}|$ is significantly minimized, as indicated by the yellow horizontal line ($\frac{dV}{dt} = \dot{V}_{th}$). The orange vertical dashed line signifies the corresponding time point $\bar{t}$ when $\frac{dV}{dt}$ reaches $\dot{V}_{th}$. Beyond the time point, $\frac{dV}{dt}$ continues to converge toward zero with minimal change.

GRNN analysis

While the previous section presents the stability analysis of each individual gene-perceptron, they need to be integrated into a sub-GRNN to perform the classification operation. In this study, our emphasis is on developing a mathematical framework to elucidate the weights representing gene-gene influence within the GRNN and focus on the classification application. This required us to search for a sub-GRNN that has structural and operational parallels shared with an ANN for classification, where we focused on multilayer and randomly structured sub-GRNNs that form the basis for larger sub-GRNN architectures.

Multilayer sub-GRNN

This sub-GRNN, which is illustrated in Fig. 6, consists of three hidden-layer gene-perceptrons ($g_{1,1}, g_{1,2}, g_{1,3}$) and one output-layer gene-perceptron ($g_{2,1}$) ($g_{i,j}$ represents the j^{th} gene-perceptron in the i^{th} layer in the sub-GRNN). The concentrations that are output from layer 1 to layer 2 are $[TF]_{1,1}, [TF]_{1,2}$, and $[TF]_{1,3}$, and $[P]$ is the output from gene-perceptron $g_{2,1}$. The two input genes (g_{x_1} and g_{x_2}) are TFs with corresponding concentrations, $[TF]_{x_1}$ and $[TF]_{x_2}$, respectively. The RNA concentration changes over time t for the hidden-layer gene-perceptrons, based on Eq. 1, can be expressed as,

$$\frac{d[R]_i}{dt} = k_{1_i} C_{N_i} \left(\frac{[TF]_{x_1}^n}{K_{A_i}^n + [TF]_{x_1}^n} \right) \cdot \left(\frac{[TF]_{x_2}^n}{K_{A_i}^n + [TF]_{x_2}^n} \right) - d_{1_i} [R]_i, \qquad \text{(Equation 8)}$$

for the activators, $i = g_{1,1}, g_{1,2}$. Since gene-perceptron $g_{1,3}$ has a repression from gene-perceptron g_{x_2}, the changes in the RNA production based on Eq. 3, is given by

$$\frac{d[R]_{g_{1,3}}}{dt} = k_{1_{g_{1,3}}} C_{N_{g_{1,3}}} \left(\frac{[TF]_{x_1}^n}{K_{A_{g_{1,3}}}^n + [TF]_{x_1}^n} \cdot \frac{K_{A_{g_{1,3}}}}{K_{A_{g_{1,3}}} + [TF]_{x_2}^n} \right) - d_{1_{g_{1,3}}} [R]_{g_{1,3}}. \qquad \text{(Equation 9)}$$

The RNA concentration changes of the output gene-perceptron $g_{2,1}$, which consists of TFs from gene-perceptrons $g_{1,1}$, $g_{1,2}$, and $g_{1,3}$ with the output protein concentration that contribute as TF

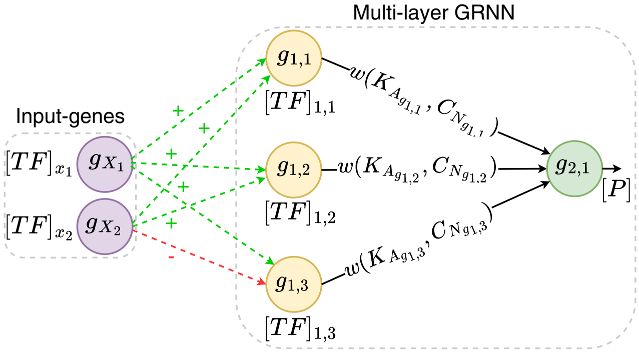

FIGURE 6 Multilayer sub-GRNN with two input-layer nodes, three hidden-layer gene-perceptrons $(g_{1,1}, g_{1,2}, g_{1,3})$, and one output-layer gene-perceptron $(g_{2,1})$; and their corresponding output concentrations are transcription factors $[TF]_{1,1}, [TF]_{1,2}, [TF]_{1,3}$, and protein concentration $[P]$, respectively. There are two input genes $(g_{x_1},\ g_{x_2})$ considered as two TFs with concentration of $[TF]_{x_1}$ and $[TF]_{x_2}$, respectively. In this context, $g_{i,j}$ represents the j^{th} gene-perceptron in i^{th} layer in the sub-GRNN. Input-gene activators and input-gene repressors are denoted by $(+)$ and $(-)$ edges, respectively. The weights (w) of this sub-GRNN is a function of the TF concentration corresponding to the half-maximal RNA concentration (K_{A_i}) and gene product copy number (C_{N_i}) for gene-perceptron i represented as $w(K_{A_i}, C_{N_i})$.

concentration $([TF]_{1,1} = [P]_{g_{1,1}}, [TF]_{1,2} = [P]_{g_{1,2}}$, and $[TF]_{1,3} = [P]_{g_{1,3}})$ to accumulate to invoke the expression is given by,

$$\frac{d[R]_{g_{2,1}}}{dt} = k_{1_{g_{2,1}}} C_{N_{g_{2,1}}} \left(\frac{[TF]^n_{1,1}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,1}} \right) \cdot \left(\frac{[TF]^n_{1,2}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,2}} \right) \cdot \left(\frac{[TF]^n_{1,3}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,3}} \right) - d_{1_{g_{2,1}}} [R]_{g_{2,1}}. \tag{Equation 10}$$

Each of the gene-perceptrons also undergoes a translation process. Therefore, the protein concentration change for each gene-perceptron can be modeled using Eq. 4 for $i = g_{1,1}, g_{1,2}, g_{1,3}$, and $g_{2,1}$. The maximum-stable protein concentration can be derived by setting Eqs. 8, 9, and 10 to zero to find $[R]^*_i$, which is then plugged into Eq. 4 and set to zero for $i = g_{1,1}, g_{1,2}, g_{1,3}$, and $g_{2,1}$, respectively.

$$i = g_{1,1}, g_{1,2} \Rightarrow [P]^*_i = \frac{k_{1_i} k_{2_i} C_{N_i}}{d_{1_i} d_{2_i}} \left(\frac{[TF]^n_{x_1}}{K^n_{A_i} + [TF]^n_{x_1}} \right) \times \left(\frac{[TF]^n_{x_2}}{K^n_{A_i} + [TF]^n_{x_2}} \right), \tag{Equation 11}$$

$$[P]^*_{g_{1,3}} = \frac{k_{1_{g_{1,3}}} k_{2_{g_{1,3}}} C_{N_{g_{1,3}}}}{d_{1_{g_{1,3}}} d_{2_{g_{1,3}}}} \left(\frac{[TF]^n_{x_1}}{K^n_{A_{g_{1,3}}} + [TF]^n_{x_1}} \right) \times \left(\frac{K_{A_{g_{1,3}}}}{K_{A_{g_{1,3}}} + [TF]^n_{x_2}} \right), \tag{Equation 12}$$

$$[P]^*_{g_{2,1}} = \frac{k_{1_{g_{2,1}}} k_{2_{g_{2,1}}} C_{N_{g_{2,1}}}}{d_{1_{g_{2,1}}} d_{2_{g_{2,1}}}} \left(\frac{[TF]^n_{1,1}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,1}} \right) \times \left(\frac{[TF]^n_{1,2}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,2}} \right) \left(\frac{[TF]^n_{1,3}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,3}} \right). \tag{Equation 13}$$

Equations 11, 12, and 13, which are the stable concentration quantity of proteins produced, are used to compute the classification areas for each gene-perceptron based on the value of concentration, which is further elaborated in the results as we present a case study. Subsequently, we apply the approach from the methods to show the stability of the gene-perceptron in this sub-GRNN. The overall stability of the GRNN based on the derived Lyapunov function of Eq. 27 (please see the appendix), which can be further expressed for I number of TFs connected to a gene-perceptron (i), is represented as follows

$$\begin{aligned} \frac{dV}{dt} = & - \prod_{j=1}^{I} \frac{C^2_{N_i} \cdot [TF]^{2n}_j \cdot k^2_{1_i} \cdot e^{\left(-2t\left(d_{1_i} + d_{2_i}\right)\right)}}{d_{1_i} d_{2_i} \left([TF]^n_j + K^n_{A_j} \right)^2 \left(d_{1_i} - d_{2_i} \right)^2} \\ & \times \left(d^3_{2_i} \cdot e^{\left(2d_{2_i} t\right)} - 2d_{1_i} d^2_{2_i} \cdot e^{\left(2d_{2_i} t\right)} \right. \\ & \left. + d^2_{1_i} d_{2_i} \cdot e^{\left(2d_{2_i} t\right)} \right) + \left(d_{1_i} k^2_{2_i} \cdot e^{\left(2d_{1_i} t\right)} \right. \\ & \left. + d_{2_i} k^2_{2_i} \cdot e^{\left(2d_{2_i} t\right)} \right) - - \left(d_{1_i} k^2_{2_i} \cdot e^{\left(t\left(d_{1_i} + d_{2_i}\right)\right)} \right. \\ & \left. + d_{2_i} k^2_{2_i} \cdot e^{\left(t\left(d_{1_i} + d_{2_i}\right)\right)} \right), \end{aligned} \tag{Equation 14}$$

where $[TF]_j$ and K_{A_j} are concentration of j^{th} TF and corresponding half-maximal RNA concentration for gene-perceptron i, respectively.

Random structured sub-GRNN

As described earlier, the relationship of gene-perceptrons within a GRN that have common TFs may have intermediate gene-perceptrons within the path of connections. We analyze how this impacts on the overall stability of the sub-GRNN, where the network for this case is presented in Fig. 7. In this form of networks, it is necessary to consider the RNA concentration change from the intermediate gene-perceptron $(g_{2,1})$ and its impact on the output-layer gene-perceptron $(g_{3,1})$. The expressions for each gene-perceptron, and their relative TFs from their immediate predecessor, are represented as follows:

$$\frac{d[R]_{g_{2,1}}}{dt} = k_{1_{g_{2,1}}} C_{N_{g_{2,1}}} \left(\frac{[TF]^n_{1,1}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,1}} \right) - d_{1_{g_{2,1}}} [R]_{g_{2,1}}, \tag{Equation 15}$$

$$\frac{d[R]_{g_{3,1}}}{dt} = k_{1_{g_{3,1}}} C_{N_{g_{3,1}}} \left(\frac{[TF]^n_{2,1}}{K^n_{A_{g_{3,1}}} + [TF]^n_{2,1}} \right) \cdot \left(\frac{[TF]^n_{1,2}}{K^n_{A_{g_{3,1}}} + [TF]^n_{1,2}} \right) \times \left(\frac{[TF]^n_{1,3}}{K^n_{A_{g_{3,1}}} + [TF]^n_{1,3}} \right) - d_{1_{g_{3,1}}} [R]_{g_{3,1}}. \tag{Equation 16}$$

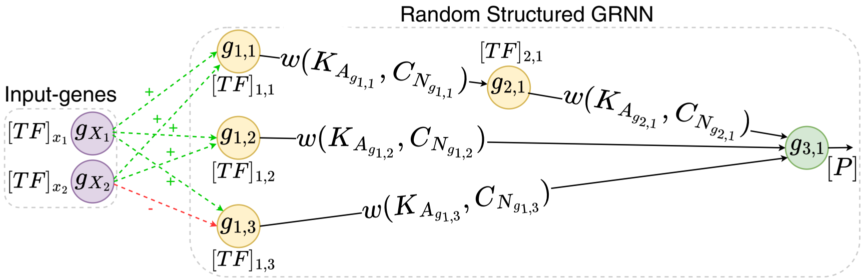

FIGURE 7 Random structured sub-GRNN with three input-layer gene-perceptrons ($g_{1,1}$, $g_{1,2}$, $g_{1,3}$), one intermediate gene-perceptron ($g_{2,1}$), and one output-layer gene-perceptron ($g_{3,1}$). This structure is an extension from the sub-GRNN in Fig. 6.

Here, the protein concentration from Eq. 5 can be derived from Eq. 15 (i.e., $[TF]_{1,1} = [P]_{1,1}$), since gene-perceptron $g_{2,1}$ is activated by gene-perceptron $g_{1,1}$. The RNA concentration models behave similarly to the case without the intermediate gene-perceptron for gene-perceptrons $g_{1,1}, g_{1,2},$ and $g_{1,3}$ and can be derived directly from Eqs. 8 and 9. Using Eq. 4 we can determine the protein concentration change for each gene-perceptron Fig. 7.

Using the maximum-stable protein concentration derived from Eqs. 15 and 16, we can determine $[R]_i^*$, which is then applied to Eq. 4 and used to determine the maximum-stable value for $i = g_{2,1}$ and $g_{3,1}$. This will result in the following maximum-stable protein production that is represented as follows

$$[P]^*_{g_{2,1}} = \frac{k_{1_{g_{2,1}}} k_{2_{g_{2,1}}} C_{N_{g_{2,1}}}}{d_{1_{g_{2,1}}} d_{2_{g_{2,1}}}} \left(\frac{[TF]^n_{1,1}}{K^n_{A_{g_{2,1}}} + [TF]^n_{1,1}} \right), \tag{Equation 17}$$

$$[P]^*_{g_{3,1}} = \frac{k_{1_{g_{3,1}}} k_{2_{g_{3,1}}} C_{N_{g_{3,1}}}}{d_{1_{g_{3,1}}} d_{2_{g_{3,1}}}} \left(\frac{[TF]^n_{2,1}}{K^n_{A_{g_{3,1}}} + [TF]^n_{2,1}} \right) \cdot \left(\frac{[TF]^n_{1,2}}{K^n_{A_{g_{3,1}}} + [TF]^n_{1,2}} \right) \left(\frac{[TF]^n_{1,3}}{K^n_{A_{g_{3,1}}} + [TF]^n_{1,3}} \right). \tag{Equation 18}$$

We use Eq. 11 to determine $[P]_i^*$ for $i = g_{1,1}$ and $g_{1,2}$, while for $i = g_{1,3}$ we use Eq. 12. For the stability analysis, Eq. 14 is used with $I = 2$ for $g_{1,1}, g_{1,2}$, and $g_{1,3}$, $I = 1$ for $g_{2,1}$, and $I = 3$ for $g_{3,1}$, corresponding to the number of TFs for each gene-perceptron.

RESULTS

In this section, we perform temporal stability analysis and obtain the classification areas for the two multilayer sub-GRNN network topologies (Figs. 6 and 7) as well as the sub-GRNN derived from *E. coli* GRN.

Multilayer sub-GRNN

The temporal stability for each gene-perceptron within the generic multilayer sub-GRNN is illustrated in Fig. 8. This simulation employed the model from Eq. 14 with parameter set 1 (Table 1). Gene-perceptrons $g_{1,1}$ and $g_{1,2}$ accelerate toward equilibrium within the time window $t \leq 5$, as demonstrated by the negative amplitude of the derivative of the Lyapunov function. Here, $\dot{V}_{th}$ is set to -1×10^{-11} for gene-perceptrons $g_{1,1}$ and $g_{1,2}$. While these two gene-perceptrons show a positive trend after $t \approx 5$, at $\bar{t} \approx 30$ and $\bar{t} \approx 35$ the derivative of their Lyapunov function, $\frac{dV}{dt}$, approaches $\dot{V}_{th} = -1 \times 10^{-11}$, indicating a slower pace toward equilibrium. Furthermore, $\dot{V}_{th}$ for gene-perceptron $g_{1,3}$ is set to -1×10^{-8}. In contrast to $g_{1,1}$ and $g_{1,2}$, $g_{1,3}$ exhibits a positive trend from the beginning and $\frac{dV}{dt}$ approaches $\dot{V}_{th}$ at $\bar{t} \approx 15$ due to its distinct repression from the input gene g_{X_1}, signifying a gradual approach toward equilibrium. The output-layer gene-perceptron ($g_{2,1}$) displays a similar pattern to gene-perceptrons $g_{1,1}$ and $g_{1,2}$ owing to its direct predecessors being activators. At $\bar{t} \approx 35$, $\frac{dV}{dt}$ of $g_{2,1}$ approaches $\dot{V}_{th} = -1 \times 10^{-8}$, indicating a slower approach toward equilibrium. Thus, within the time window ($t \leq 35$), the multilayer sub-GRNN is considered computationally accurate, as all gene-perceptrons within the sub-GRNN have approximately reached equilibrium.

Given the gene-perceptron's stability at equilibrium (Fig. 8), we can use Eqs. 11, 12, and 13, to calculate output protein $[P]_i^*$ for different input concetrations ($[TF]_{x_1}$ and $[TF]_{x_2}$). In our simulations, we set the classifier threshold to 0.5, a commonly used default threshold in classification tasks in previous works

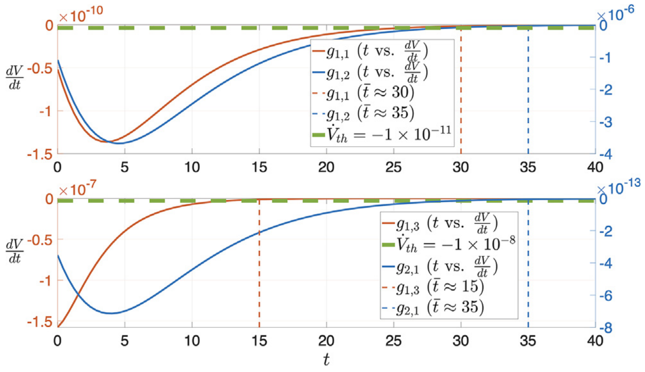

FIGURE 8 Temporal stability analysis of gene-perceptrons within the multilayer sub-GRNN, where vertical dashed lines indicate the time points when $\frac{dV}{dt} = \dot{V}_{th}$ (here $\dot{V}_{th} = -1 \times 10^{-11}$ for $g_{1,1}$ and $g_{1,2}$, and $\dot{V}_{th} = -1 \times 10^{-8}$ for $g_{1,3}$ and $g_{2,1}$).

TABLE 1 **Parameter configuration for the generic multilayer sub-GRNN**

	$C_{N_{g_{1,1}}}$	$C_{N_{g_{1,2}}}$	$C_{N_{g_{1,3}}}$	$C_{N_{g_{2,1}}}$	$k_{1_{g_{1,1}}}$	$k_{1_{g_{1,2}}}$	$k_{1_{g_{1,3}}}$	$k_{1_{g_{2,1}}}$	$k_{2_{g_{1,1}}}$	$k_{2_{g_{1,2}}}$	$k_{2_{g_{1,3}}}$	$k_{2_{g_{2,1}}}$
Parameter	$d_{1_{g_{1,1}}}$	$d_{1_{g_{1,2}}}$	$d_{1_{g_{1,3}}}$	$d_{1_{g_{2,1}}}$	$d_{2_{g_{1,1}}}$	$d_{2_{g_{1,2}}}$	$d_{2_{g_{1,3}}}$	$d_{2_{g_{2,1}}}$	$K_{A_{g_{1,1}}}(\times 10^{-7})$	$K_{A_{g_{1,2}}}(\times 10^{-7})$	$K_{A_{g_{1,3}}}(\times 10^{-7})$	$K_{A_{g_{2,1}}}(\times 10^{-7})$
Parameter set 1	10	70	100	90	0.1	0.2	0.4	0.5	0.1	0.2	0.4	0.5
	0.3	0.2	0.5	0.6	0.3	0.2	0.5	0.6	500	100	1000	50
Parameter set 2	10	15	25	45	0.1	0.2	0.4	0.5	0.1	0.2	0.4	0.5
	0.3	0.2	0.5	0.6	0.3	0.2	0.5	0.6	100*	20*	10*	50*

The values marked with an asterisk (*) are the parameters that are modified. Units of $C_{N_i}, k_{1_i}, k_{2_i}, d_{1_i}, d_{2_i}$, and K_{A_i} are molecules, s^{-1}, s^{-1}, min^{-1}, h^{-1}, and molecules, respectively).

(44,45), chosen also for clarity in our simulation figures illustrating classification areas. The calculated output protein $[P]_i^*$ is illustrated over varying input concentrations, highlighting the values above and below the threshold ($[P]^* = 0.5$). Decision boundaries reflect how the classification areas change based on the edge (activation or repression) connected to the target gene-perceptron and corresponding parameters in Eqs. 11, 12, and 13. The inputs ($[TF]_{x_1}$ and $[TF]_{x_2}$) vary, while parameters such as gene product copy number (C_{N_i}), transcription rate (k_{1_i}), translation rate (k_{2_i}), RNA degradation rate (d_{1_i}), protein degradation rate (d_{2_i}), and TF concentration, corresponding to the half-maximal RNA concentration (K_{A_i}), are kept constant. We consider two parameters sets to determine the different classification regions, which are presented in Table 1.

For parameter set 1, we obtain the classification areas shown in Fig. 9 *a*. The decision boundary and their top view for each gene-perceptron are shown in the first and second rows, respectively. The gene-perceptron $g_{1,2}$ has the largest classification area above the threshold due to its lower TF concentration corresponding to half-maximal RNA concentration K_{A_i}, compared with gene-perceptrons $g_{1,1}$ and $g_{1,3}$. Moreover, the decision boundaries for gene-perceptrons $g_{1,1}$ and $g_{1,2}$ exhibit a similar shape, classifying the majority of the values above the threshold. In contrast, gene-perceptron $g_{1,3}$ covers a larger area for the values below the threshold since it is repressed by

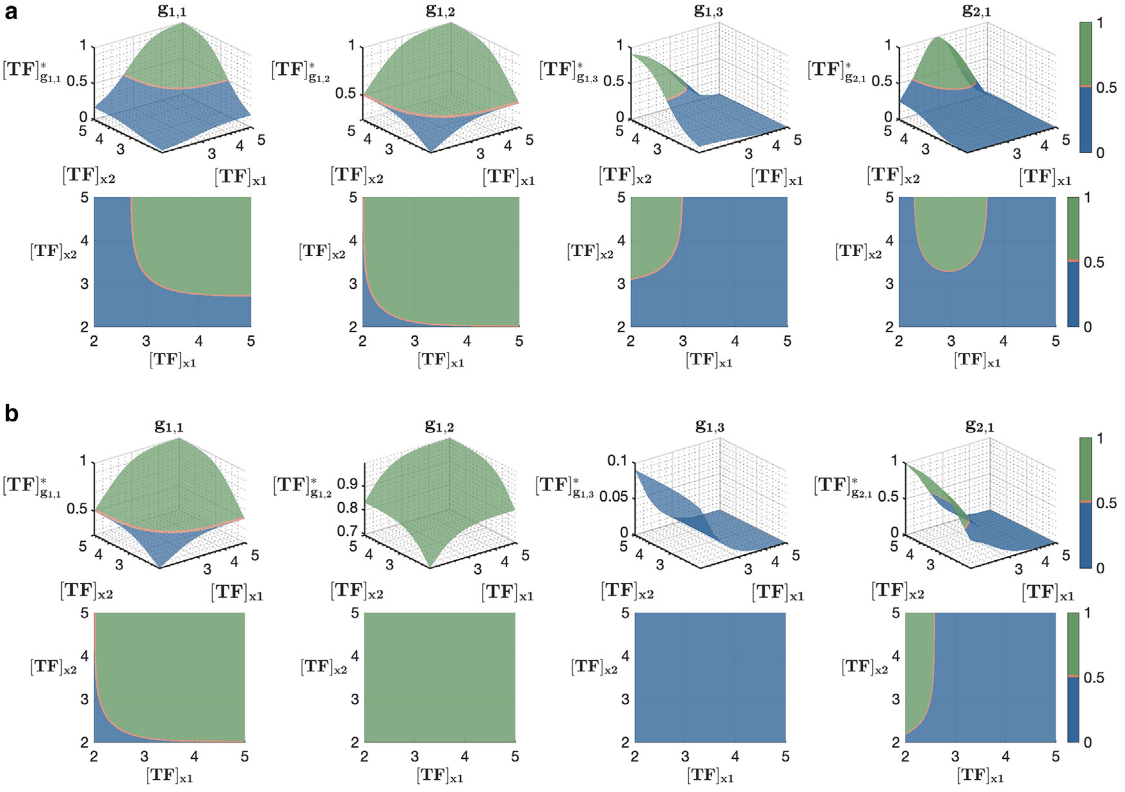

FIGURE 9 Parameter configurations for the multilayer sub-GRNN depicted in Fig. 6. Each graph depicts the classification area of each gene-perceptron and for (*a*) parameter set 1, as well as (*b*) parameter set 2 ($g_{2,1}$ is the output gene-perceptron that combines all classification areas of gene-perceptrons from the previous layer).

the input gene g_{x_2}. The intersection of classification areas corresponding to hidden-layer gene-perceptrons is represented by the output-layer gene-perceptron $g_{2,1}$, where the classification area above the threshold is approximately bounded by input concentrations, $2.5 \leq [TF]_{x_1} \leq 3.5$ and $3.4 \leq [TF]_{x_2}$. Due to the significant contribution from gene-perceptrons $g_{1,1}$ and $g_{1,2}$ beyond the threshold, the output-layer gene-perceptron $g_{2,1}$ exhibits a rightward shift.

For parameter set 2 (Table 1), the lower K_{A_i} values have shifted the classification area above the threshold compared with parameter set 1. This shift is evident in Fig. 9 *b*, particularly for gene-perceptron $g_{1,2}$, which results in classifying the majority of the values above the threshold. Conversely, for gene-perceptron $g_{1,3}$, the classification area shifts below the threshold due to the repression from the input when reducing the half-maximal RNA concentration K_{A_i}. The classification range for gene-perceptron $g_{1,1}$ expands compared with parameter set 1, approximately bounded by $2.3 \leq [TF]_{x,1}$ and $2.1 \leq [TF]_{x,2}$. Considering all gene-perceptrons, the output-layer gene-perceptron $g_{2,1}$ shows a leftward shift in the decision boundary, becoming slightly more linear. Overall, modifying the half-maximal RNA concentration K_{A_i} can significantly expand the classification area.

Random structured sub-GRNN

This sub-GRNN consists of three hidden-layer gene-perceptrons, one intermediate gene-perceptron, and one output-layer gene-perceptron, as illustrated in Fig. 7. The temporal stability analysis for this sub-GRNN is presented in Fig. 10 and utilizes Eq. 14 and parameter set 1 from Table 2. In this simulation, we set $\dot{V}_{th} = -1 \times 10^{-11}$ for gene-perceptrons $g_{1,1}$ and $g_{1,2}$, -1×10^{-8} for gene-perceptrons $g_{1,3}$ and $g_{2,1}$, and -1×10^{-5} for gene-perceptron $g_{3,1}$. Fig. 10 illustrates that gene perceptrons $g_{1,1}$, $g_{1,2}$, $g_{3,1}$, and $g_{2,1}$ display fluctuations in the Lyapunov function derivative $\frac{dV}{dt}$ similar to those observed in Fig. 8. This similarity can be attributed to their immediate predecessors acting as activators, leading to the derivative of the Lyapunov function, $\frac{dV}{dt}$, converging to $\dot{V}_{th} = -1 \times 10^{-11}$ at $\bar{t} \approx 30$ and 35 for $g_{1,1}$ and $g_{1,2}$, respectively, -1×10^{-8} at $\bar{t} \approx 35$ for $g_{2,1}$, and -1×10^{-5} at $\bar{t} \approx$ 35 for $g_{3,1}$. For gene-perceptron $g_{1,3}$, $\frac{dV}{dt}$ gradually approaches $\dot{V}_{th} = -1 \times 10^{-8}$ within the time interval $t \leq 15$, exhibiting stability fluctuations akin to those observed in the network lacking the intermediate gene-perceptron. This similarity arises from both gene-perceptrons being influenced by their repressive predecessors. Consequently, within the temporal window ($t \leq 35$), the randomly structured sub-GRNN is considered computationally reliable, with all gene-perceptrons within the sub-GRNN converging toward equilibrium.

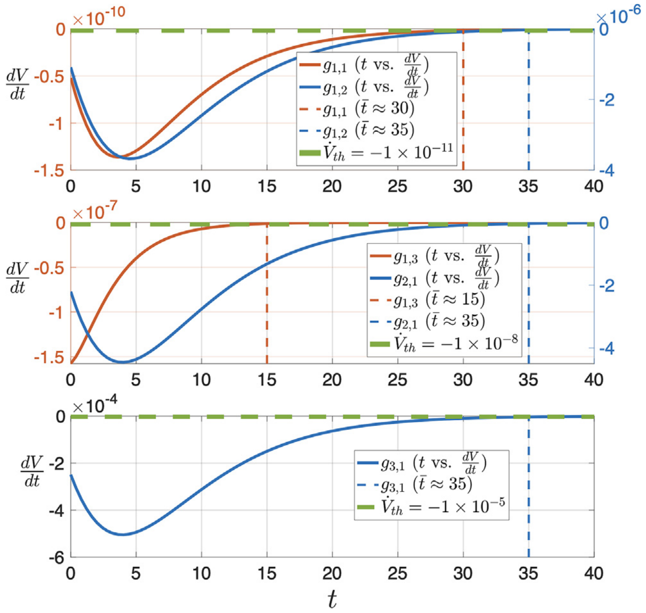

FIGURE 10 Temporal stability analysis of gene-perceptrons within the random structured sub-GRNN, with vertical dashed lines marking the time points when $\frac{dV}{dt} = \dot{V}_{th}$, where $\dot{V}_{th} = -1 \times 10^{-11}$ for $g_{1,1}$ and $g_{1,2}$, $\dot{V}_{th} = -1 \times 10^{-8}$ for $g_{1,3}$ and $g_{2,1}$, and $\dot{V}_{th} = -1 \times 10^{-5}$ for $g_{3,1}$.

Following the temporal stability analysis, we apply Eqs. 11 and 12 to determine the maximum-stable protein concentration ($[P]_i^*$) for gene-perceptrons $g_{1,1}$,$g_{1,2}$, and $g_{1,3}$. However, unlike the sub-GRNN in Fig. 6, Eq. 13 is not used to determine the classification area for the output-layer gene-perceptron. Instead, for the computation of $[P]_i^*$ for gene-perceptrons $g_{2,1}$ and $g_{3,1}$, both Eqs. 17 and 18 are employed due to the addition of the intermediate gene-perceptron compared with the multilayer sub-GRNN in Fig. 6. The calculated protein concentration output $[P]_i^*$ values for different input concentrations used to determine the classification area for each gene-perceptron are presented in Fig. 11. We also used two different sets of parameters from Table 2 to analyze different classification areas.

Parameter set 1 results in the classification areas shown in Fig. 11 *a*. As gene-perceptron $g_{2,1}$ serves as the intermediate gene-perceptron of $g_{1,1}$, we observe similar classification areas and decision boundaries. In addition, repression from the input gene g_{x_1} to gene-perceptron $g_{1,3}$ results in a distinctive decision boundary, approximately within the range of $3 \leq [TF]_{x_2}$ and $3 \geq [TF]_{x_1}$. Overall, gene-perceptron $g_{3,1}$ represents the intersection of the hidden-layer gene-perceptrons, with the classification area beyond the threshold bounded by $2.5 \leq [TF]_{x_2} \leq 3.5$ and $3 \geq [TF]_{x_1}$.

TABLE 2 Parameter configuration for the random structured sub-GRNN

	$k_{1_{g_{1,1}}}$	$k_{1_{g_{1,2}}}$	$k_{1_{g_{1,3}}}$	$k_{1_{g_{2,1}}}$	$k_{1_{g_{3,1}}}$	$k_{2_{g_{1,1}}}$	$k_{2_{g_{1,2}}}$	$k_{2_{g_{1,3}}}$	$k_{2_{g_{2,1}}}$	$k_{2_{g_{3,1}}}$
	$d_{1_{g_{1,1}}}$	$d_{1_{g_{1,2}}}$	$d_{1_{g_{1,3}}}$	$d_{1_{g_{2,1}}}$	$d_{1_{g_{3,1}}}$	$d_{2_{g_{1,1}}}$	$d_{2_{g_{1,2}}}$	$d_{2_{g_{1,3}}}$	$d_{2_{g_{2,1}}}$	$d_{2_{g_{3,1}}}$
Parameter	$C_{N_{g_{1,1}}}$	$C_{N_{g_{1,2}}}$	$C_{N_{g_{1,3}}}$	$C_{N_{g_{2,1}}}$	$C_{N_{g_{3,1}}}$	$K_{A_{g_{1,1}}} (\times 10^{-7})$	$K_{A_{g_{1,2}}} (\times 10^{-7})$	$K_{A_{g_{1,3}}} (\times 10^{-7})$	$K_{A_{g_{2,1}}} (\times 10^{-7})$	$K_{A_{g_{3,1}}} (\times 10^{-7})$
Parameter set 1	0.1	0.2	0.4	0.8	0.5	0.1	0.2	0.4	0.7	0.5
	0.3	0.2	0.5	0.7	0.6	0.3	0.2	0.5	0.9	0.6
	10	15	25	45	6	500	100	1000	50	50
Parameter set 2	0.1	0.2	0.4	0.8	0.5	0.1	0.2	0.4	0.7	0.5
	0.3	0.2	0.5	0.7	0.6	0.3	0.2	0.5	0.9	0.6
	10	15	25	45	30	50*	100*	1000*	10*	50*

The values marked with an asterisk (*) are the parameters that are modified. Units of $C_{N_i}, k_{1_i}, k_{2_i}, d_{1_i}, d_{2_i}$, and K_{A_i} are molecules, s^{-1}, s^{-1}, min^{-1}, h^{-1}, and molecules, respectively).

In contrast, reducing the TF concentration at the half-maximal RNA concentration (K_{A_i}) for a gene-perceptron as shown in parameter set 2, alters the classification areas for both gene-perceptron $g_{1,1}$ and its immediate intermediate gene-perceptron $g_{2,1}$, as illustrated in Fig. 11 *b*. The classification area significantly expands above the threshold, while dropping below it when lowering the TF concentration corresponding to the half-maximal RNA concentration K_{A_i}, as it is inversely proportional to the maximum protein concentration $[P]_i^*$ based on Eqs. 8 and 17. Alterations made to gene-perceptron $g_{1,1}$ notably impact $g_{2,1}$, the predecessor gene-perceptron in the GRNN. Other hidden-layer gene-perceptrons $g_{1,2}$ and $g_{1,3}$ remain unaffected between parameter sets 1 and 2. Parameter set 2 results in a leftward shift in the classification area of the output-layer gene-perceptron $g_{3,1}$ compared with set 1. In summary, parameter adjustments lead to shifts in the decision boundary of the output-layer gene-perceptrons; with decreased K_{A_i} causing a leftward shift in the the classification area.

E. coli sub-GRNN classification analysis

This section demonstrates the classification areas for the *E. coli* sub-GRNN illustrated in Fig. 13 *a*, which is

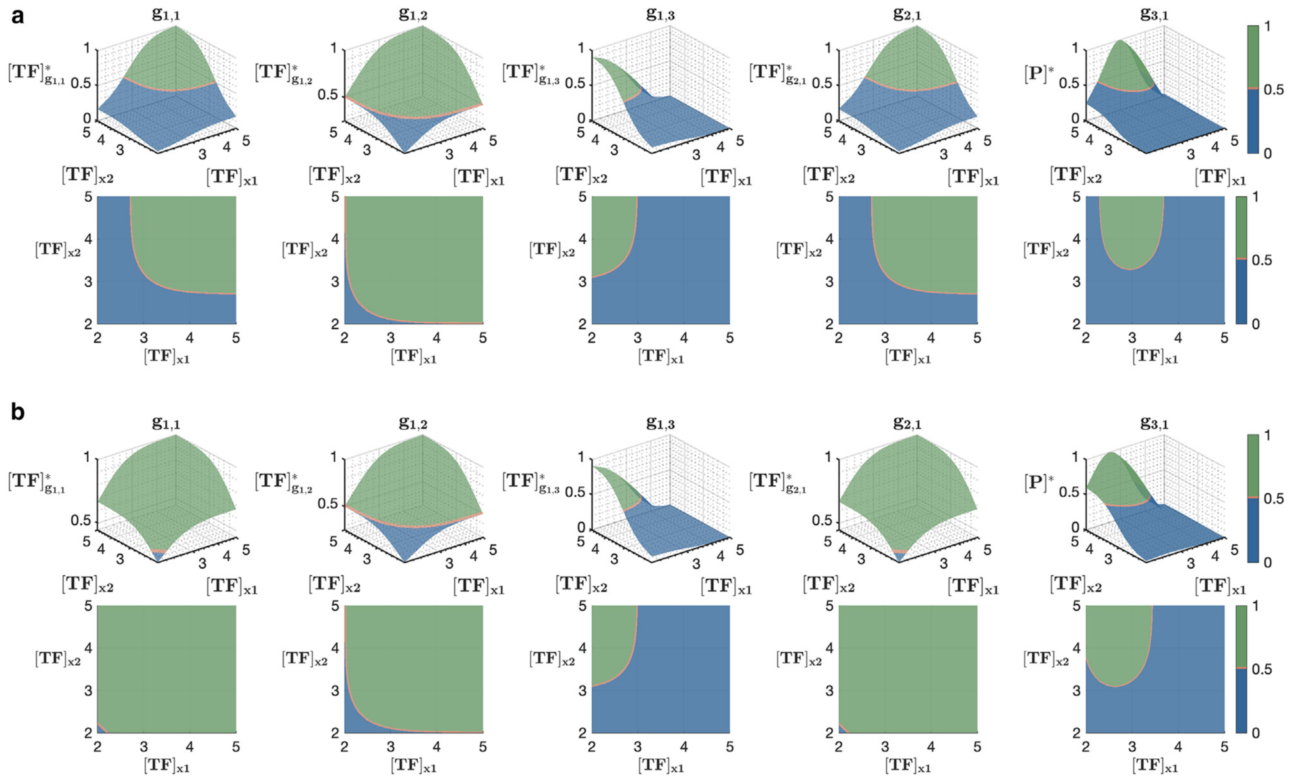

FIGURE 11 Parameter configurations for the random structured sub-GRNN in Fig. 6. Each graph depicts the classification area of each gene-perceptron and for (*a*) parameter set 1 and (*b*) parameter set 2 ($g_{3,1}$ is the output gene-perceptron that combines all classification areas of gene-perceptrons from the previous layer).

TABLE 3 Parameter values used for the *E. coli* sub-GRNN

	Value			
Parameter	$b1891$	$b1892$	$b1071$	Ref.
$k_{1_i}(\mathrm{s}^{-1})$	0.05	0.05	0.05	Milo et al. (47)
$k_{2_i}(\mathrm{s}^{-1})$	0.05	0.05	0.05	Gong et al. (48); Zhu and Dai (49)
C_{N_i}(molecules)	45	45	45	Milo et al. (47); Schaechter and The View From Here Group (50), Glauner et al. (51)
$d_{1_i}(\mathrm{min}^{-1})$	0.2	0.2	0.2	Milo et al. (47)
$d_{2_i}(\mathrm{h}^{-1})$	3.5%	3.5%	3.5%	Milo et al. (47)
$K_{A_i}(\times 10^{-7})$	75.30 ($b3025$) 4261.64 ($b3357$)	71.10 ($b3025$) 2061.56 ($b3357$)	306 ($b1891$) 377 ($b1892$)	GSE65244

K_{A_i} of the corresponding TF is given in the table.

extracted from the trans-omic data of *E. coli* GRN (46). The network consists of two input genes ($b3025$, $b3357$), two hidden-layer gene-perceptrons ($b1891$ and $b1892$), and one output-layer gene-perceptron ($b1071$), with their corresponding TF concentrations $[TF]_i$ for $i = b3025, b3357, b1891$, and $b1892$, and protein concentration $[P]_{b1071}$. In this specific GRNN, all TFs are considered activators. For the output-layer gene-perceptron ($i = b1071$), we employ Eq. 8, Eqs. 4 and 11 with TFs $x_1 = b1891$ and $x_2 = b1892$ to calculate RNA, protein concentration change, and maximum protein concentration ($[P]_i^*$), respectively, using the parameter values in Table 3.

Similar to the previous sub-GRNNs, we based the stability analysis for this sub-GRNN on (14). For the two input-layer gene-perceptrons ($i = b1891$ and $b1892$), we consider TFs $j = b3025$ and $b3357$, while for the output-layer gene-perceptron $i = b1071$, we evaluate stability with the TFs $j = b1891$ and $b1891$. In Figs. 8 and 10, we observe that gene-perceptrons with an immediate activator exhibit analogous stability fluctuations as the derivative of the Lyapunov function $\frac{dV}{dt}$ approaches $\dot{V}_{th} = -1 \times 10^{-12}$. This behavior agrees also with the *E. coli* sub-GRNN, which is shown in Fig. 12, which shows the temporal stability for gene-perceptrons ($g_{1,1}, g_{1,2}$, and $g_{2,1}$), which is influenced by the immediate activator predecessors displaying uniform stability. Here, $\dot{V}_{th}$ is set to -1×10^{-12} for each gene-perceptron. According to the figure, the derivative of the Lyapunov function, $\frac{dV}{dt}$ of gene-perceptrons $b1891$, $b1892$, and $b1071$ approach $\dot{V}_{th} = -1 \times 10^{-12}$ at times $\bar{t} \approx 32$, 30, and 35, respectively. Overall, the analysis indicates that, within the temporal window ($t \leq 35$), all gene-perceptrons in the sub-GRNN reach equilibrium, ensuring network-wide stability and computational reliability.

Once proving the stability of the sub-GRNN, we ascertain the maximum-stable protein concentration to obtain the classification ranges. To compute maximum-stable protein concentration ($[P]_i^*$) for gene-perceptrons $i = b1891$ and 1892, we use Eq. 11 with the replacement of x_1 and x_2 by $b3025$ and $b3357$ as input genes. Furthermore, for the computation of output concentrations $[P]_i^*$, concerning gene-perceptron $i = b1071$, Eq. 11 is used with TFs as $x_1 = b1891$ and $x_2 = b1892$ with the assumption that the Hill coefficient n is equal to 1 in all simulations. Since K_{A_i} is the TF concentration corresponding to the half-maximal RNA concentration, there are two K_{A_i} values for each gene-perceptron

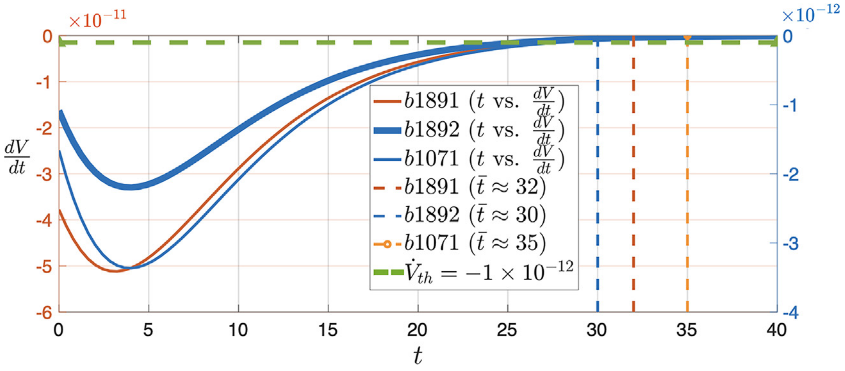

FIGURE 12 Temporal stability analysis for each gene-perceptron within the *E. coli* sub-GRNN indicated by vertical dashed lines marking the time points when $\frac{dV}{dt} = \dot{V}_{th}$, where $\dot{V}_{th} = -1 \times 10^{-12}$ for each gene-perceptron.

because each has two TFs, as shown in Fig. 13 *a*. The time-series data of gene expression levels for *E. coli* was used by first identifying the gene's half-maximal expression level K_{A_i} and then finding the expression level of its TF at that corresponding time point. For the remaining parameters that were obtained from the literature as shown in Table 3, the average value was used.

The classification area from our analysis is shown in Fig. 13 *b*. The classification area of gene-perceptron *b*1892 has expanded toward the left when compared with *b*1891, and this is because the expression level of the half-maximal RNA concentration K_{A_i} of both TFs (*b*3025 and *b*3357) corresponding to *b*1891 exceed the value of K_{A_i} for *b*1892. The classification area above the threshold of *b*1892 is defined within the limits of $[TF]_{b3025} \geq 2.7$ and $[TF]_{b3357} \geq 2.7$, in contrast to *b*1891, which is approximately bounded by $[TF]_{b3025} \geq 3.5$ and $[TF]_{b3357} \geq 3.8$. Consistent with the decision boundary simulations performed on the two generic multilayer sub-GRNNs (Figs. 9 and 11), the output-layer gene-perceptron (*b*1071) of this sub-GRNN also exhibited a intersection of classification areas driven by the input-layer gene-perceptrons. In line with this, as gene-perceptron *b*1891 had the majority of its classification area below the threshold and gene-perceptron *b*1892 had the majority above the threshold, the decision boundary of gene-perceptron *b*1071 is approximately bounded by $[TF]_{b3025} \geq 2.9$ and $[TF]_{b3357} \geq 2.9$. Overall, gene-perceptrons within the sub-GRNN derived from *E. coli* GRN exhibit tunable decision boundaries by selecting subnetworks from the GRN at steady state and collectively they function as multilayer sub-GRNNs showcasing aspects of biological AI.

CONCLUSION

Biological cells naturally perform a number of key functionalities through their GRN, showcasing similarities to ANN computing. Controlling chemical inputs to the GRN and engineering of the genetic circuit can inspire the development of wet-neuromorphic computing systems. In this study, we considered a sub-GRNN derived from a larger GRN, mathematically modeling the transcription and translation process to showcase the nonlinear sigmoidal behavior of gene expressions resulting in the transformation of genes into gene-perceptrons. Due to this sigmoidal behavior of gene expressions, we established nonlinear classifiers using different sub-GRNNs. To ensure reliable computing with minimal concentration fluctuations, stability analysis was conducted for the sub-GRNN using the eigenvalue method and Lyapunov's stability theorem. The latter determining the time at which the stability is achieved.

Three nonlinear classifiers were developed using two multilayer sub-GRNNs and a sub-GRNN extracted from the *E. coli* GRN. From the simulation for different parameter settings for the two multilayer sub-GRNNs revealed that the TF concentration at the half-maximal gene expression level K_{A_i}, has a significant impact on the shifting of the classification boundary. Based on the outcomes of the stability analysis and simulations, we can conclude that the GRN exhibits NN properties as the gene-perceptron demonstrated sigmoidal-like behavior for multiple inputs and tunable decision boundaries. Further, by engineering living cells it is possible to obtain desired nonlinear classifiers based on our application. Our model has the potential to transform GRNs into GRNNs when the suitable

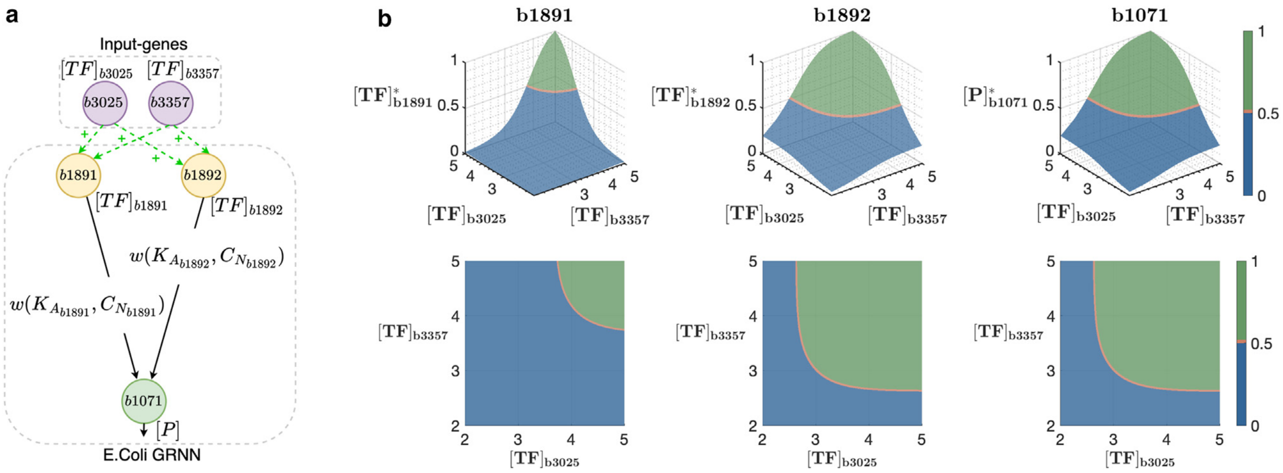

FIGURE 13 *E. coli* sub-GRNN classification analysis. (*a*) Fully connected sub-GRNN derived from the *E. coli* GRN. This network consists of two input genes (*b*3025 and *b*3357), two hidden-layer gene-perceptrons (*b*1891 and *b*1892), and one output-layer gene-perceptron (*b*1071). (*b*) Classification regions of each gene perceptron within the *E. coli* sub-GRNN, with gene-perceptron *b*1071 as the output.

parameters are established for the dual-layered chemical reaction model.

APPENDIX

RNA and protein concentration model

To model the RNA and protein concentration change, mass-balance differential equations were used based on Hill function (37). Transcription of a gene-perceptron begins with TF and RNA polymerase binding to the promoter, which is modeled by

$$[Prom.TF] = C_{N_i}\frac{[TF]^n}{[TF]^n + K_{A_i}^n}, \qquad \text{(Equation 19)}$$

where $[TF], n, K_{A_i}, [Prom.TF]$, and C_{N_i} are concentration of TFs, Hill coefficient, TF concentration corresponding to half-maximal RNA concentration, complex produced after TFs bind to promoter and gene product copy number, respectively. The complex, *Prom.TF* transcribes into RNA at the rate of k_{1_i} and subsequently RNA degrades at the rate of d_{1_i}, which can be modeled as

$$\frac{d[R]_i}{dt} = k_{1_i}[Prom.TF] - d_{1_i}[R]_i. \qquad \text{(Equation 20)}$$

By plugging Eqs. 19 in 20 we can obtain Eq. 1. In contrast, if a gene-perceptron is repressed by a TF, Eq. 19 can be expressed as

$$[Prom.TF] = C_{N_i}\frac{K_{A_i}^n}{K_{A_i}^n + [TF]^n}. \qquad \text{(Equation 21)}$$

Since the initial RNA concentration transcribed by a gene-perceptron is $[R]_i(0)$ (i.e., $[R]_i(t = 0) = [R]_i(0)$), the solution of Eq. 1 as given by Eq. 2 can be derived using the integrating factor, $IF = e^{\int d_{1_i} \ dt} = e^{d_{1_i} t}$, where t and d_{1_i} are time and RNA degradation rate, respectively. Transcribed RNA is then translated into protein at the proteome level. To solve the differential equation of protein concentration change for Eq. 4 we can follow two steps. Step 1: replacing RNA concentration ($[R]_i$) in Eq. 4 with the solution obtained for the differential equation of RNA concentration change from Eq. 2. Step 2: using the integrating factor ($IF = e^{\int d_{2_i} dt} = e^{d_{2_i} t}$) and initial RNA concentration ($[R]_i(0)$), as well as initial protein concentration $[P]_i(0)$ (i.e., $[P]_i(t = 0) = [P]_i(0)$), we can obtain the equation for the protein concentration in Eq. 5. By setting $\frac{d\ [R]_i}{dt} = 0$, we can obtain maximum-stable RNA concentration at the steady state ($[R]_i^*$) expressed by Eq. 6. In addition, protein concentration at the steady state ($[P]_i^*$) can be represented by Eq. 7, which is derived by plugging $\frac{d\ [P]_i}{dt} = 0$ in Eq. 4.

Determining gene-perceptron stability

In this section, we derive the stability of a gene-perceptron using eigenvalues (Algorithm 1) of differential equations for RNA and protein concentration change Eqs. 1 and 4 and using Lypunov's stability theorem. The first step encompasses the *DefineDynamics* function, which formalizes alterations in RNA and protein concentrations over time, encapsulating the system's concentration dynamical Eqs. 1 and 4. While our study primarily addresses gene transcription as detailed in Eq. 1, the same approach is applicable to the repression process described in Eq. 3. Subsequently, the algorithm transitions to the *CalculateJacobian* function, where the Jacobian matrix, J_i, is computed Eq. 24. Derived from partial derivatives of the concentration dynamical equations concerning RNA and protein concentrations Eqs. 1 and 4 , this matrix sheds light on the system's local behavior around equilibrium points Eqs. 6 and 7. Following this, the *ComputeEigenvalues* function is invoked to ascertain the eigenvalues, λ_1 and λ_2, of the Jacobian matrix. Finally, the *CheckStability* function evaluates the stability of the gene-perceptron based on the calculated eigenvalues ($\lambda_1 = -d_{1_i}, \lambda_2 = -d_{2_i}$). Stability is confirmed if both eigenvalues are negative (i.e., $d_{1_i}, d_{2_i} > 0$), indicating that deviations from equilibrium will decay over time, eventually leading the system to return to its equilibrium state.

ALGORITHM 1 Gene-perceptron stability analysis using eigenvalue method

```
Input: RNA and protein concentration changes ..., $g([R]_i, [P]_i)$, protein degradation rate $d_{2_i}$, RNA degradation rate $d_{1_i}$, translation rate $k_{2_i}$, $i^{th}$ gene-perceptron.
Output: eigenvalues $\lambda_1$, $\lambda_2$ and stability condition.
$f, g \leftarrow$ DefineDynamics($f, g, d_{1_i}, d_{2_i}, k_{2_i}$);
$J_i \leftarrow$ CalculateJacobian($f, g, d_{1_i}, d_{2_i}$);
$\lambda_1, \lambda_2 \leftarrow$ ComputeEigenvalues($J_i$);
CheckStability($\lambda_1, \lambda_2$);
Function DefineDynamics($\frac{d[R]_i}{dt}, \frac{d[P]_i}{dt}$):
    Define concentration dynamics from Eqs. 1 and 4:
        $(1) \Rightarrow \frac{d[R]_i}{dt} = f([R]_i, [P]_i),$    (Equation 22)
        $(4) \Rightarrow \frac{d[P]_i}{dt} = g([R]_i, [P]_i).$    (Equation 23)
    return $f, g$;
Function CalculateJacobian($f, g, d_{1_i}, d_{2_i}, k_{2_i}$):
    Calculate Jacobian $J_i$:
        $J_i = \begin{bmatrix} \frac{\partial f}{\partial [R]_i} & \frac{\partial f}{\partial [P]_i} \\ \frac{\partial g}{\partial [R]_i} & \frac{\partial g}{\partial [P]_i} \end{bmatrix} = \begin{bmatrix} -d_{1_i} & 0 \\ k_{2_i} & -d_{2_i} \end{bmatrix}.$
                                                   (Equation 24)
    return $J_i$;
Function ComputeEigenvalues($J_i$):
    Compute the eigenvalues $\lambda_1, \lambda_2$ of $J_i$ using $|J_i - \lambda I| = 0$;
    return $\lambda_1, \lambda_2$;
Function CheckStability($\lambda_1, \lambda_2$):
    if $\lambda_1 < 0$ and $\lambda_2 < 0$ then
        The gene-perceptron is stable;
    else
        The gene-perceptron is unstable;
    end
```

ALGORITHM 2 Analyzing temporal stability using Lyapunov function

```
Input: RNA and protein concentration changes $f([R]_i, [P]_i)$, $g([R]_i, [P]_i)$, protein degradation rate $d_{2_i}$, RNA degradation rate $d_{1_i}$, translation rate $k_{2_i}$, transcription rate $k_{1_i}$, gene product copy number $C_N$, TF concentration for half-maximal RNA concentration $K_{A_i}$, a threshold, $V_{th_i}$ at which $\frac{dV}{dt}$ is minimized, Hill coefficient $n$ for $i^{th}$ gene-perceptron.
Output: Time point $(\bar{t})$ at which the derivative of the Lyapunov function $\frac{dV}{dt} = \dot{V}_{th}$.
Function DefineLyapunovFunction($[R]_i, [P]_i, [R]_i^*, [P]_i^*$):
    $V([R]_i, [P]_i) \leftarrow ([R]_i - [R]_i^*)^2 + ([P]_i - [P]_i^*)^2$ as given by Eq. 25;
    return $V$;
Function CalculateDotV($V, [R]_i, [P]_i, C_{N_i}, k_{1_i}, d_{1_i}, d_{2_i}, k_{2_i}, [TF], K_{A_i}, n, t$):
    Calculate $\dot{V}([R]_i, [P]_i)$ as given by Eqs. 26 and 27;
    return $\dot{V}$;
Function FindMinT($\dot{V}, \dot{V}_{th}$):
    if $\dot{V} \le \dot{V}_{th}$ then
        $\bar{t} \leftarrow \text{argmin}_t |\dot{V} - \dot{V}_{th}|$;
        return $\bar{t}$;
    end
    else
        return "Lyapunov stability conditions are not satisfied";
    end
Function Main():
    $V \leftarrow$ DefineLyapunovFunction($[R]_i$, $[P]_i$, $[R]_i^*$, $[P]_i^*$);
    if $V > 0$ and $V = 0$ when $[R]_i = [R]_i^*$ and $[P]_i = [P]_i^*$ then
        $\dot{V} \leftarrow$ CalculateDotV($[R]_i, [P]_i, C_{N_i}, k_{1_i}, d_{1_i}, d_{2_i}, k_{2_i}, TF, K_{A_i}, n, t$);
        $\bar{t} \leftarrow$ FindMinT($\dot{V}$);
        return $\bar{t}$;
    end
    else
        return "Lyapunov stability conditions are not satisfied";
    end
```

Conversely, if any eigenvalue is nonnegative, the gene-perceptron is labeled as unstable (52), suggesting that deviations from equilibrium may amplify, hindering the system from reverting to its original state.

Algorithm 2 commences by defining the Lyapunov function (53) (*DefineLyapunovFunction*), utilizing inputs such as RNA concentration $([R]_i)$, protein concentration $([P]_i)$, RNA equilibrium concentration $([R]_i^*)$, and protein equilibrium concentration $([P]_i^*)$, to compute the Lyapunov function $(V([R]_i, [P]_i))$ as follows:

$$V([R]_i, [P]_i) = ([R]_i - [R]_i^*)^2 + ([P]_i - [P]_i^*)^2. \qquad \text{(Equation 25)}$$

This computed V is then processed by the *CalculateDotV* function, which takes additional parameters including C_{N_i}, k_{1_i}, k_{2_i}, d_{1_i}, d_{2_i}, TF concentration $([TF])$, K_{A_i}, n, and time (t). It calculates the derivative of V $\left(\frac{dV}{dt}\right)$ as follows:

$$\dot{V}([R]_i, [P]_i) = \frac{dV}{dt} = \frac{\partial V}{\partial [R]_i} \cdot \frac{d[R]_i}{dt} + \frac{\partial V}{\partial [P]_i} \cdot \frac{d[P]_i}{dt}. \qquad \text{(Equation 26)}$$

By plugging $\frac{d[R]_i}{dt}$ and $\frac{d[P]_i}{dt}$ from Eqs. 1 and 4, differentiating Eq. 25 with respect to $[R]_i$ and $[P]_i$ to obtain $\frac{\partial V}{\partial [R]_i}$ and $\frac{\partial V}{\partial [P]_i}$ and finally replacing $[R]_i^*, [P]_i^*, [R]_i$ and $[P]_i$, with Eqs. 6, 7, 2, and 5 we get Eq. 26, which is represented as follows:

$$\begin{aligned}(26) \Rightarrow \frac{dV}{dt} = & -\frac{C_{N_i}^2 \cdot [TF]^{2n} \cdot k_{1_i}^2 \cdot e^{\left(-2t\left(d_{1_i}+d_{2_i}\right)\right)}}{d_{1_i} d_{2_i} \left([TF]^n + K_{A_i}^n\right)^2 \left(d_{1_i} - d_{2_i}\right)^2} \cdot \\ & \left(d_{2_i}^3 \cdot e^{\left(2d_{2_i} t\right)} - 2d_{1_i} d_{2_i}^2 \cdot e^{\left(2d_{2_i} t\right)} \right. \\ & \left. + d_{1_i}^2 d_{2_i} \cdot e^{\left(2d_{2_i} t\right)}\right) + \left(d_{1_i} k_{2_i}^2 \cdot e^{\left(2d_{1_i} t\right)} \right. \\ & \left. + d_{2_i} k_{2_i}^2 \cdot e^{\left(2d_{2_i} t\right)}\right) - \left(d_{1_i} k_{2_i}^2 \cdot e^{\left(t\left(d_{1_i}+d_{2_i}\right)\right)} \right. \\ & \left. + d_{2_i} k_{2_i}^2 \cdot e^{\left(t\left(d_{1_i}+d_{2_i}\right)\right)}\right),\end{aligned} \qquad \text{(Equation 27)}$$

where we assume initial RNA concentration of zero $([R]_i(0) = 0)$ and initial protein concentration of zero $([P]_i(0) = 0)$. Then, this calculated $\frac{dV}{dt}$ is forwarded to *FindMinT* function. When $\frac{dV}{dt} \le \dot{V}_{th}$, the *FindMinT* function calculates the time point $\bar{t}$, where the absolute difference between $\dot{V}$ and $\dot{V}_{th}$ is minimized, denoted as $\bar{t} \leftarrow \text{argmin}_t |\dot{V} - \dot{V}_{th}|$, signifying the moment at which the derivative of the Lyapunov function $\frac{dV}{dt}$ approaches $\dot{V}_{th}$. Otherwise, it indicates that the conditions of the Lyapunov stability theorem are not satisfied. Within the *main()* function, we systematically invoke the specified functions: initially, the *DefineLyapunovFunction* is called to establish V, followed by a verification of Lyapunov function criteria, specifically $V > 0$ and $V = 0$ under conditions $[R]_i = [R]_i^*$ and $[P]_i = [P]_i^*$. Upon satisfying these conditions, the *CalculateDotV* function is utilized to determine $\frac{dV}{dt}$, which in turn informs the calculation of $\bar{t}$ via *FindMinT*. Conversely, if the Lyapunov function conditions are not met, it indicates that the Lyapunov stability conditions are not satisfied.

AUTHOR CONTRIBUTIONS

A.R., S.S., and S.B. designed the theoretical framework of the study. The implementation of the analysis was done by A.R. while A.A.G. provided 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 National Science Foundation (NSF) under grant no. 2316960.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Chowdhury, M., and A. W. Sadek. 2012. Advantages and limitations of artificial intelligence. *In* Artificial intelligence applications to critical transportation issues, *6*, pp. 360–375.
2. Imboden, S., X. Liu, …, N. Y. C. Lin. 2023. Trustworthy in silico cell labeling via ensemble-based image translation. *Biophys. Rep.* 3, 100133.
3. Li, Z., F. Liu, …, J. Zhou. 2022. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transact. Neural Networks Learn. Syst.* 33:6999–7019.
4. Medsker, L., and L. Jain. 1999. Recurrent Neural Networks: Design and Applications. *In* International Series on Computational Intelligence CRC Press.
5. Kasneci, E., K. Seßler, …, G. Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Indiv. Differ.* 103, 102274.
6. Schuman, C. D., S. R. Kulkarni, …, B. Kay. 2022. Opportunities for neuromorphic computing algorithms and applications. *Nat. Comput. Sci.* 2:10–19.
7. Nesbeth, D. N., A. Zaikin, …, T. Laptyeva. 2016. Synthetic biology routes to bio-artificial intelligence. *Essays Biochem.* 60:381–391.
8. Akan, O. B., H. Ramezani, …, M. Kuscu. 2017. Fundamentals of molecular information and communication science. *Proc. IEEE.* 105:306–318.
9. Akan, O. B., E. Dinc, …, B. A. Bilgin. 2023. Internet of Everything (IoE) - From Molecules to the Universe. *IEEE Commun. Mag.* 61:122–128.
10. Schwenk, H., and J.-L. Gauvain. 2005. Training neural network language models on very large corpora. *In* Proceedings of human language technology conference and conference on empirical methods in natural language processing, pp. 201–208.
11. Balasubramaniam, S., S. Somathilaka, …, M. Pierobon. 2023. Realizing Molecular Machine Learning Through Communications for Biological AI. *IEEE Nanotechnol. Mag.* 17:10–20.
12. Bi, D., A. Almpanis, …, R. Schober. 2021. A Survey of Molecular Communication in Cell Biology: Establishing a New Hierarchy for Interdisciplinary Applications. *IEEE Commun. Surv. Tutorials.* 23:1494–1545.
13. Kagan, B. J., A. C. Kitchen, …, K. J. Friston. 2022. In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron.* 110:3952–3969.e8.
14. 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.
15. Li, X., L. Rizik, …, R. Daniel. 2021. Synthetic neural-like computing in microbial consortia for pattern recognition. *Nat. Commun.* 12:3139.
16. Samaniego, C. C., A. Moorman, …, E. Franco. 2021. Signaling-based neural networks for cellular computation. *In* 2021 American Control Conference (ACC) IEEE, pp. 1883–1890.
17. Söldner, C. A., E. Socher, …, H. Sticht. 2020. A Survey of Biological Building Blocks for Synthetic Molecular Communication Systems. *IEEE Commun. Surv. Tutor.* 22:2765–2800.
18. Bicen, A. O., I. F. Akyildiz, …, Y. Koucheryavy. 2016. Linear Channel Modeling and Error Analysis for Intra/Inter-Cellular Ca2+ Molecular Communication. *IEEE Trans. Nanobiosci.* 15:488–498.
19. Somathilaka, S. S., S. Balasubramaniam, …, X. Li. 2023. Revealing gene regulation-based neural network computing in bacteria. *Biophys. Rep.* 3, 100118.
20. Wu, Y.-c., and J.-w. Feng. 2018. Development and application of artificial neural network. *Wireless Pers. Commun.* 102:1645–1656.
21. Rizik, L., L. Danial, …, R. Daniel. 2022. Synthetic neuromorphic computing in living cells. *Nat. Commun.* 13:5602.
22. Alon, U. 2019. An Introduction to Systems Biology: Design Principles of Biological Circuits. *In* Chapman & Hall/CRC mathematical and computational biology series CRC Press https://books.google.com/books?id=sBHUwQEACAAJ.
23. Sabzi, S., M. Asadi, and H. Moghbeli. 2019. Regenerative energy management of electric drive based on Lyapunov stability theorem. *J. Mod. Power Syst. Clean Energy.* 7:321–328.
24. Yerramalla, S., E. Fuller, …, B. Cukic. 2003. Lyapunov Analysis of Neural Network Stability in an Adaptive Flight Control System. *In* Self-Stabilizing Systems. S.-T. Huang and T. Herman, eds Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 77–92.
25. Rau, V., and G. Durga Prasad. 1993. Dynamic stability assessment of wind turbine generators using the Lyapunov function approach. *Elec. Power Syst. Res.* 27:61–72.
26. Lim, K. H., K. P. Seng, …, S. W. Chin. 2009. Lyapunov Theory-Based Multilayered Neural Network. *IEEE Trans. Circuits Syst. II.* 56:305–309.
27. Dai, H., B. Landry, …, R. Tedrake. 2021. Lyapunov-stable neural-network control. Preprint at ArXiv. https://api.semanticscholar.org/CorpusID:235657467.
28. Li, H., J. Wang, and J. Meng. 2021. Chapter 4 - Nonlinear control. *In* Learning Control. D. Zhang and B. Wei, eds Elsevier, pp. 93–102. https://www.sciencedirect.com/science/article/pii/B9780128223147000092.
29. Segall-Shapiro, T. H., A. J. Meyer, …, C. A. Voigt. 2014. A `resource allocator'for transcription based on a highly fragmented T7 RNA polymerase. *Mol. Syst. Biol.* 10:742.
30. Landry, B. P., R. Palanki, …, J. J. Tabor. 2018. Phosphatase activity tunes two-component system sensor detection threshold. *Nat. Commun.* 9:1433.
31. Wang, J., Y. Dong, …, H. Xu. 2021. Direct Quantification of Gene Regulation by Transcription-Factor Binding at an Endogenous Gene Locus. *Biophys. J.* 120:260a.
32. Bernstein, J. A., A. B. Khodursky, …, S. N. Cohen. 2002. Global analysis of mRNA decay and abundance in Escherichia coli at single-gene resolution using two-color fluorescent DNA microarrays. *Proc. Natl. Acad. Sci. USA.* 99:9697–9702.
33. Holmqvist, E., and J. Vogel. 2018. RNA-binding proteins in bacteria. *Nat. Rev. Microbiol.* 16:601–615. https://doi.org/10.1038/s41579-018-0049-5.
34. Tejada-Arranz, A., V. de Crécy-Lagard, and H. de Reuse. 2020. Bacterial RNA degradosomes: molecular machines under tight control. *Trends Biochem. Sci.* 45:42–57.
35. Irshad, I. U., and A. K. Sharma. 2023. Decoding stoichiometric protein synthesis in E. coli through translation rate parameters. *Biophys. Rep.* 3, 100131.
36. Cao, Z., T. Filatova, …, R. Grima. 2020. A stochastic model of gene expression with polymerase recruitment and pause release. *Biophys. J.* 119:1002–1014.
37. Santillán, M. 2008. On the use of the Hill functions in mathematical models of gene regulatory networks. *Math. Model Nat. Phenom.* 3:85–97.
38. Yugi, K., S. Ohno, …, S. Kuroda. 2019. Rate-oriented trans-omics: integration of multiple omic data on the basis of reaction kinetics. *Curr. Opin. Struct. Biol.* 15:109–120.
39. Thompson, B. M., I. Marinelli, …, L. S. Satin. 2020. Multiple Feedback Mechanisms Underlying Beta Cell Secretory Oscillations. *Biophys. J.* 118, 562a.
40. Lavrinenko, I. A., G. A. Vashanov, …, Y. D. Nechipurenko. 2022. A New Model of Hemoglobin Oxygenation. *Entropy.* 24:1214.

41. Saito, K., R. Green, and A. R. Buskirk. 2020. Translational initiation in E. coli occurs at the correct sites genome-wide in the absence of mRNA-rRNA base-pairing. *Elife.* 9, e55002.

42. Xu, B., L. Liu, and G. Song. 2021. Functions and regulation of translation elongation factors. *Front. Mol. Biosci.* 8, 816398.

43. Marintchev, A. 2012. Preface. *In* Fidelity and Quality Control in Gene Expression, Academic Press, volume 86 of *Advances in Protein Chemistry and Structural Biology*. A. Marintchev, ed, p. ix. https://www.sciencedirect.com/science/article/pii/B978012386497000013X.

44. Chen, J. J., C.-A. Tsai, …, C.-H. Chen. 2006. Decision threshold adjustment in class prediction. *SAR QSAR Environ. Res.* 17:337–352.

45. Zou, Q., S. Xie, …, Y. Ju. 2016. Finding the best classification threshold in imbalanced classification. *Big Data Res.* 5:2–8.

46. Tierrafría, V. H., C. Rioualen…, 2022. RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in Escherichia coli K-12. *Microb. Genom.* 8:mgen000833.

47. Milo, R., P. Jorgensen, …, M. Springer. 2010. BioNumbers—the database of key numbers in molecular and cell biology. *Nucleic Acids Res.* 38:D750–D753.

48. Gong, X., S. Fan, …, S. Tao. 2008. Comparative analysis of essential genes and nonessential genes in Escherichia coli K12. *Mol. Genet. Genom.* 279:87–94.

49. Zhu, M., and X. Dai. 2019. Maintenance of translational elongation rate underlies the survival of Escherichia coli during oxidative stress. *Nucleic Acids Res.* 47:7592–7604.

50. Schaechter, M.; The View From Here Group. 2001. Escherichia coli and Salmonella 2000: the view from here. *Microbiol. Mol. Biol. Rev.* 65:119–130.

51. Glauner, B., J. V. Höltje, and U. Schwarz. 1988. The composition of the murein of Escherichia coli. *J. Biol. Chem.* 263:10088–10095. https://www.sciencedirect.com/science/article/pii/S0021925819814813.

52. Woolf, P. 2009. Chemical Process Dynamics and Controls. *In* Open textbook library University of Michigan Engineering Controls Group https://books.google.com/books?id=Op87vwEACAAJ.

53. Khalil, H. K. 2009. Lyapunov stability. *In* Control systems, robotics and automation, *12*, p. 115.