A Spatial Artificial Chemistry Implementation of a Gene Regulatory Network Aimed at Generating Protein Concentration Dynamics

Iliya Miralavy
Michigan State University
Department of Computer Science and Engineering
BEACON Center for the Study of Evolution in Action

Wolfgang Banzhaf*
Michigan State University
Department of Computer Science and Engineering
BEACON Center for the Study of Evolution in Action
banzhafw@msu.edu

Abstract Gene regulatory networks are networks of interactions in organisms responsible for determining the production levels of proteins and peptides. Mathematical and computational models of gene regulatory networks have been proposed, some of them rather abstract and called artificial regulatory networks. In this contribution, a spatial model for gene regulatory networks is proposed that is biologically more realistic and incorporates an artificial chemistry to realize the interaction between regulatory proteins called the transcription factors and the regulatory sites of simulated genes. The result is a system that is quite robust while able to produce complex dynamics similar to what can be observed in nature. Here an analysis of the impact of the initial states of the system on the produced dynamics is performed, showing that such models are evolvable and can be directed toward producing desired protein dynamics.

Keywords
Gene regulatory network, artificial chemistry, artificial regulatory network, modeling, complex network, evolutionary computation

1 Introduction

A biological system is a network of interactions between natural entities that operate with a specific purpose (Muggianu et al., 2018). For example, the lungs, trachea, nose, and related muscles work together to form the respiratory system in humans. But biological systems are not limited to the compounds of different organs. An ant colony is another example of a complex regulated system, in this case, made of a population of organisms serving the purpose of survival of the colony. Among the means of their organization is the spatial distribution of individuals in the ecosystem (Theraulaz et al., 2002). Most biological systems are adaptable and robust and exhibit complex dynamics. They work in a spatially organized world, partially reflecting the hierarchical order of numerous spatial and temporal scales on which cause–effect relationships play out.

1.1 30 Years of the Journal *Artificial Life*

1.1.1 Simulation

In the last 30 years, the journal *Artificial Life* has dedicated itself to the study of life and its origin, evolution, and diverse expression. According to the original idea of Christopher Langton (1986), the field of Artificial Life was "to study Life as it could be," beyond existing life on Earth, which has its

* Corresponding author.

own discipline of biology. The argument was to try to construct living systems, either in the medium of computation or in matter, as a way to understand the phenomenon of life. Continuing this pursuit, Artificial Life has developed ways to synthesize lifelike behaviors in computational models, thereby teasing out the universal aspects of life. This synthetic approach not only augments our grasp of biological principles but also broadens the scope of inquiry to include the vast possibilities of life-as-it-could-be, revealing life's potential variations and capabilities (Langton, 1989/2019).

1.1.2 Modeling

There has been substantial progress in modeling biological and physical systems throughout the literature (Deakin, 1990; Villaverde et al., 2019; Wilkinson, 2009). In general, it is possible to classify biological models into three types: *static*, *comparative static*, and *dynamic* (Hannon & Ruth, 1997). A *static* model is a snapshot of a process or event, for example, a map of pandemic intensity in different regions of the world at a certain time. A *comparative static* model comprises several snapshots of a process or event at different times that can be used to compare and retrieve meaningful data without modeling the process itself. Finally, a *dynamic* model aims to model a sequence of processes and events by representing the changes in the state of a system over time. For example, a differential equation showing and predicting the spread of a pandemic over time is a dynamic model (Mohamadou et al., 2020). However, dynamic models are not limited to mathematical modeling approaches. Depending on the system and its potential for discretization, methods such as discrete-event modeling (Wainer, 2017) and agent-based modeling (Holcombe et al., 2012) have become important. This followed the realization that mathematical modeling can only go so far without the execution of these models in algorithms. Although mathematical models can be easily formulated, as soon as real-world spatial and temporal aspects come into play, there are limits to what is solvable without simulation.

1.1.3 Cellular Automata

Cellular automata (CAs), the subject of Langton's examination, have a venerable history. Long before the field was named, they were used for studying self-replicating systems. When Johann von Neumann became interested in self-reproducing systems, he hit on them as the vehicle for his examination of these phenomena. His proof that self-reproducing automata are possible, published posthumously by his collaborator Arthur Burks, is to this day a classic in the literature of Artificial Life (von Neumann, 1966). But CAs are not only a means to study self-reproduction; they have been used for studies of emergent phenomena (see, e.g., Bilotta & Pantano, 2005) and also have become a tool to model partial differential equation dynamics and other dynamical systems (Toffoli, 1984). They served even as an inspiration for computer hardware, as is testified by the success of the series of famous connection machines from thinking machines (Tucker & Robertson, 1988). One of the most useful aspects of the CA paradigm is its natural ability to model discretized spatial phenomena. This will be a critical component of the study undertaken here, where we examine spatial interactions in an agent-based system.

1.1.4 Artificial Life

In a seminal article by Bedau et al., published in 2000, 14 important problems were listed for researchers in Artificial Life to tackle. They were grouped into three categories worth mentioning again:

1. How does life arise from the nonliving?
2. What are the potentials and limits of living systems?
3. How is life related to mind, machines, and culture?

To examine these questions, all kinds of techniques have been used, some of which are applied here.

1.1.5 Artificial Chemistries

Concomitant with the open problems for Artificial Life question, a discussion ensued in the pages of this journal about the "right stuff" (Nehaniv & Wagner, 2000). In the words of the editors of this special issue of *Artificial Life*, we are

> looking for "the right stuff," that is, appropriate mathematical and computational tools and models for describing, studying, building, or understanding fundamental aspects of natural living systems or living systems as they could be (whether carbon-based, digital, or otherwise) as opposed to inanimate systems. (p. 1)

This question inspired one of the authors of this article and his then students to argue for artificial chemistries (ACs) as the appropriate tool for this undertaking. Our manuscript came too late for the special issue in 2000 but was later published as stand-alone article (Dittrich et al., 2001). This is the second aspect of the current work: an agent-based system of extreme simplicity. The agents under discussion are intended to represent molecules, with multiple agents/molecules of the same type interacting with other agents/molecules of the same or different types. The agent interaction is based on simple rules that resemble chemical reaction rules. These types of systems have been coined *artificial chemistries* and have turned out to offer a useful perspective on many complex systems (Banzhaf & Yamamoto, 2015).

To recap quickly, an AC can be denoted as a triple (S, R, A), in which S is a set of available molecules, R is a set of all possible interaction rules, and A is an algorithm that describes the system and how the molecules or objects interact with each other (Dittrich et al., 2001). In case molecules can move, an AC generally allows for rich and more complex interactions to emerge in the system (Hutton, 2002).

The AC of this contribution is set to act in a discrete spatial grid that can be described as a lattice network of cells that can hold molecules. Molecules can move randomly within the spatial grid and interact with each other, forming bonds that control the system. The AC formalism is used to define these rules of interaction.

1.1.6 Gene Regulatory Networks

The third aspect of this work, and the actual target of the modeling and simulation effort proposed here, comprises gene regulatory networks (GRNs). After the spectacular realization that higher-level organisms are not as different genetically as their morphology and behavior would suggest, it did not take a long time to realize that the main differences between different species must lie in the timing and intensity of expression of their genes rather than in their expressed sequence differences (King & Wilson, 1975).

Gene expression involves various regions, including promoters, enhancers, and inhibitors that influence protein production. For example, the promoter region determines the starting point of a gene to be transcribed. In transcription, RNA polymerase binds to the promoter region of a gene, separates the two DNA strands, and transcribes one of them to create RNA molecules. In translation, ribosomes use RNA codons as templates to create sequences of amino acids. A complete sequence of amino acids forms a protein (Calladine et al., 2004; Watson & Berry, 2009). Enhancer and inhibitor regions are other regulatory sites located upstream or downstream of the promoter region of a gene (Pennacchio et al., 2013). A special class of regulatory proteins called transcription factors (TFs) can bind to the enhancer or inhibitor region of a gene, increasing (or decreasing) the likelihood of its transcription. Besides the protein end products, intermediate products of RNA molecules also serve as regulators in this system, creating feedback loops and forming a network of interactions between genes. This complex network of interactions that control the cell production is called a GRN (Levine & Davidson, 2005).

GRNs help to differentiate cells to form varied biological tissues, control cell metabolism, influence cell signal transduction, and determine the body shapes and behaviors of complex

organisms. Unraveling the complexities of these networks is important for better understanding how DNA governs life and can have applications such as identifying and curing genetic disorders (Gnanakkumaar et al., 2019). Modeling GRN dynamics in abstract frameworks has been used to study these networks and also was applied to solve various computational problems. However, these models are often mathematical abstractions of their biological counterparts, not accounting for the stochastic nature of their building blocks (Arias et al., 2014), which can be a shortcoming of many such models (Hannon & Ruth, 1997).

1.1.7 Artificial Regulatory Networks

This article focuses on modeling GRNs: complex networks of interactions between genes in a cell responsible for regulating, among other things, the metabolic flux of matter. Our study builds on the artificial regulatory network (ARN) proposed by Banzhaf (2003). Specifically, we investigate a GRN model in which the deterministic mathematical structure of ARNs previously used for formulating gene interactions is replaced with a stochastic spatial approach. We aim to verify our hypothesis that introducing stochastic elements to the system through the movement of artificial entities in a 2D grid will result in protein dynamics that are robust to changes in most of the initial states, but also malleable to the forces of evolution. We investigate the robustness of the proposed system and explore potential approaches to utilize this model as an evolvable problem-solving tool. It turns out that spatial properties play a crucial role in generating its produced dynamics.

The ARN model introduced by Banzhaf (2003) is inspired by natural gene regulation, utilizing interactions in genomes represented by bit strings to form GRN networks. In our study, genes and regulatory sites are also represented by bit strings. Furthermore, we incorporate an AC in a spatial representation to build an agent-based model with spatial aspects. Refining the rules of interaction for the system with spatial properties makes it less abstract and allows us to account for the stochasticity caused by random movements of simulated molecules in a 2D space. Inclusion of space in bio-inspired computational models has previously proved effective in many instances, showing how such models represent biological systems more accurately. For instance, Hickinbotham et al. (2021) shows that modeling a Stringmol automata chemistry with spatial positioning enhances the model by allowing for the formation of spatial patterns that prevent extinction by parasitism. In nonspatial models, parasites can easily overtake the system, leading to extinction. However, in spatial models, the formation of chaotic wave fronts allows replicators to invade new spaces and outcompete parasites at the back of these waves, thus preserving the replicating population and enabling the evolution of complex behaviors and ecosystems. Allen et al. (2015) examine how asymmetric spatial population structures can influence the rate of neutral genetic mutations, affecting the molecular clock used to time evolutionary events. Additionally, incorporating a 2D space facilitates the visualization of the model, allowing us to observe the spatial patterns that emerge in such systems.

The rest of the article is organized as follows. The next section explores the previous literature for modeling GRNs. Then, the methods and algorithms used to define the proposed model are explained. Next, we show the results for the dynamics produced by the proposed system based on different initial conditions. Finally, we discuss our results and point out possible future directions of the current research.

2 Related Literature

In essence, this work introduces a model that more closely resembles biological systems for representing ARNs by incorporating an AC in a 2D space. We predominantly study the dynamics produced by such a representation and explore its application. In this section, we review previous models of GRNs found in the literature and provide details for each study.

Logical and discrete models are the most straightforward approaches to modeling GRNs (Karlebach & Shamir, 2008). In these methods, GRNs are considered to have discrete states and time steps. States define the conditions or the configuration of a system entity at a given time step.

In each time step, the system updates according to regulatory functions, which might result in a change of state.

Boolean networks (BNs) (Glass & Kauffman, 1973) and probabilistic Boolean networks (PBNs) (Shmulevich et al., 2002) are the most common logical techniques to model GRNs. In BNs, each gene has only two possible states, *expressed* and *not expressed*. The state of each gene in the current time step is determined by the state of other genes in the previous time steps and the regulatory functions of the PBN comprise a subset of BNs that accounts for the stochasticity in dynamic systems and gives insights into the biological GRNs. A substantial increase in the number of states in BNs and PBNs makes analyzing such systems difficult.

A Petri net (PN) is a nondeterministic mathematical modeling approach that has been used to represent GRNs (Chaouiya et al., 2011). PNs are made of *transitions*, *places*, and *arcs*. In each *place*, there can be zero or more *tokens*. An *arc* is an entity that connects a transition to a place, or vice versa, and has a weight. In a PN, a transition is enabled if there are sufficient *tokens*, that is, a number equal to or greater than the *arc* weight. A *transition* of a PN may fire if it is enabled, in which case, it consumes the *tokens* and creates *tokens* in the output. In biological modeling of GRNs using PNs, *places* represent molecules, *transitions* represent reaction rules, and *tokens* represent concentration levels (Cussat-Blanc et al., 2019). Stochasticity of PNs occurs when multiple transitions are enabled to the same place. In such cases, transitions may fire in any order. This uncertainty makes PNs similar to PBNs.

Fractal GRNs are models that use *fractal proteins* and pattern matching interaction rules to represent GRNs. In Bentley (2003), *fractal proteins* are defined as a finite square subset of the Mandelbrot set that can exist in an environment or an artificial cell. Apart from *cell fractal proteins*, a cell contains cytoplasm, a genome, and some behaviors. A receptor gene in the cell works like a mask that allows for specific protein patterns to enter the cell area. Proteins interact through their fractal shapes and the genetic markers of the genome's regulatory sites to form a network of interaction.

An example of an AC has been previously used by Astor and Adami (2000) to model a regulatory network for the evolution of artificial neural networks (ANNs). These authors utilized a hexagonal grid in which each cell could have a concentration of substrates produced by neurons. These substrates can be different types of proteins or neurotransmitters. In their system, proteins diffuse based on differential equations, and genes are expressed if there are enough chemicals of certain types in the cell's cytoplasm. The hexagonal grid they incorporated for their work could be characterized as a CA. CAs have provided an excellent framework within which to model GRN algorithms in other works as well. For example, Chavoya and Duthen (2008) use a genetic algorithm (GA) to evolve an ARN to solve the French Flag problem on a cellular automaton grid.

Some work focuses on the dynamic analysis of GRNs. Cussat-Blanc and Pollack (2012) analyze the complex patterns generated by the dynamics of ARNs by generating pictures and videos from the changes in the concentrations of proteins. They evolve ARNs to produce patterns by asking human users to rate the fitness of the produced images. Bentley (2004), Bongard and Pfeifer (2003), and Joachimczak and Wróbel (2009) used GRN models to perform morphogenesis. An interesting characteristic of using GRN models for this purpose is the emergence of repetitive patterns, rather than chaotic ones, in the evolved shapes.

GRN models have also been used in applications like agent or robot control, showing comparable performance with other artificial intelligence methods (Asr & Majd, 2013; Sanchez & Cussat-Blanc, 2014). Finally, indirect encoding has been a topic of interest for applying GRN models (Wróbel et al., 2012; Wróbel & Joachimczak, 2014). The compact and evolvable representation of GRNs can produce massive networks of interactions of entities, which makes them good candidates for indirectly encoding other systems, such as ANNs.

3 Methodology

The extended model proposed here accounts for the protein–gene interactions in a single artificial cell to produce protein concentration dynamics. We construct GRNs from linear DNA sequences

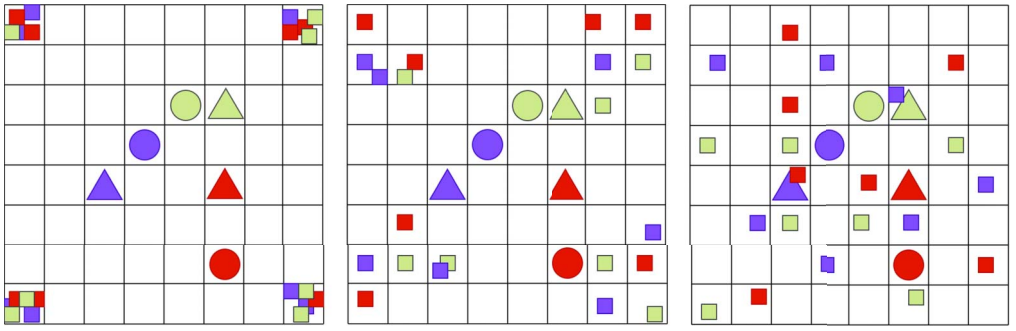


Figure 1. A snapshot of the 2D grid of the model showing four different genes and their transcription factors (TFs) in three different simulation stages. Different colors code for different genes and TFs. Triangles represent the enhancer regions, circles represent the inhibitor sites, and small squares illustrate the different TFs that move around the grid and can bind to the regulatory sites of genes other than their producing genes.

represented by string sequences of bases. A DNA sequence can have a number of genes that are identified by promoter and terminator regions. Each gene codes for a specific type of protein, which then functions as a regulatory agent (TF) controlling the transcription rates of other genes by binding to their regulatory sites. These regulatory sites are enhancer and inhibitor regions that in nature are located downstream or upstream of a promoter sequence. Binding to the enhancer region of a gene increases the protein production rate of that gene, whereas binding to the inhibitor region reduces this rate. Biologists determine the locations of these regulatory sites by genome-wide location analysis (Jin et al., 2011). We simplify this step by determining their location here as being right after the promoter sequence.

An individual is built from a single DNA sequence. First, the genes and their regulatory sites are identified and placed randomly close to the center of a 2D grid. To initialize the dynamics, an equal number of TFs for each gene[1] are positioned in a corner of the cell grid. In each regulatory time step of the system, these TFs can randomly move on the grid. In nature, TF binding happens with various patterns. TFs might locate the target site and directly bind to it in 3D space. It is also possible for TFs to slide on a DNA sequence in a 1D manner or to hop from site to site until a target regulatory site is found (de Jonge et al., 2022). The 1D search for regulatory sites is usually faster than a 3D approach because it reduces the dimensions of the problem search space. However, modeling all of the binding patterns significantly increases the complexity of the computational systems and therefore we consider only a 2D binding in this work. That means that if the location of a TF is within a threshold distance of a regulatory site of a gene, it may bind to that site. TFs stay bound for a certain number of regulatory cycles, depending on the binding strength, and are subsequently removed from the cell once they detach. When a TF protein is removed, it is replaced by another TF from the gene with the highest protein concentration in that cycle. Because all TFs are proteins in nature, replacing detached TFs with those from genes with the highest protein concentration ensures that the correlation between the protein concentrations and the number of TFs is maintained within the system. TFs are the unique product of genes in the system and are represented in a discrete form. The continuous concentration value for each gene is directly proportional to the number of TFs associated with it at any stage of the regulatory system. In the system studied here, an additional restriction is enforced: A TF cannot bind to the regulatory sites of its producing gene; otherwise, the dynamics stabilize to constant values too quickly due to self-coupling. Figure 1 comprises three snapshots of this system in three stages of the simulation.

1 Each TF corresponds to a unique gene, identified by its source of creation as its producing gene.

The model's AC is configured by applying the frequently used techniques explained by Banzhaf and Yamamoto (2015), such as defining entities and rules of the AC, measuring time, pattern matching, and a spatial topology.

2D space. Spatial properties play an important role in the biological factories of a cell. In our study, we incorporate a 2D discrete space in the form of a grid to add spatial dynamics to our system. The 2D grid represents an artificial biological cell. Utilizing a 2D space enables us to introduce a spatial topology, to measure distances between entities,[2] and allows entities to move around the grid while not being too computationally expensive. Entities might overlap in the same grid cell, and grid borders are wrapped around, meaning that if an entity moves out of one side, it will return to the grid on the side of the opposing border, continuing the move in the same direction.

Time measurement. We use the notion of a *cycle* to determine regulatory time steps in the system. In each cycle, the ARN goes through a movement phase in which TFs perform a random walk with a limited number of steps on the 2D grid. This process is followed by a regulation phase in which the outcome of the movement phase enables nearby entities to interact with one another and provides the basis for change in the system dynamics.

Pattern matching. We use pattern matching as the interaction rule between different entities in the system. This technique is similar to its biological counterpart and is explained in detail when we discuss the system's rules of interaction.

3.1 Set of System Entities (*S*)

The studied system models entities positioned on a single DNA molecule. The artificial entities in *S* are defined as follows.

3.1.1 DNA

A DNA molecule is modeled as a linear sequence of bases (A, G, C, and T). DNA is randomly initialized at the very start, determines the structure of the network of interactions, and consists of a number of genes and nonexpressed code segments. In biology, DNA is made of two complementary strands; however, here we simplify to model only one strand. This sequence is not modeled spatially on the 2D grid and serves only as the genome representation of individual GRNs. Once genes are identified from the DNA, their regulatory sites will have random immutable positions in the 2D space to enable TFs to interact with them.

3.1.2 Gene

A gene is a subset of DNA that starts and ends with a unique pattern of bases. The four-base patterns AGCT and TCGA are chosen to determine the start and the end of all genes, respectively, which play the role of the promoter and terminator regions in biological genes. The probability of finding an arbitrary four-base pattern in a sequence is 0.39%, whereas that of a three-base pattern is 1.5%. In the experimental setup here, a four-base pattern was chosen that reduced the number of genes identified in a sequence. This approach aimed to control the number of genes and their lengths in the artificial DNA molecule. However, employing three-base patterns, while having longer DNA sequences, would yield similar outcomes. The proposed system allows for specifying various start and stop "codons" with different patterns and lengths. Gene identification happens in the system after the DNA is initialized and results in identifying genes of different lengths. Genes code for proteins and have two regulatory sites of *enhancer* and *inhibitor* regions that regulate the protein production of the artificial cell. They are arbitrarily positioned in order (first enhancer, then inhibitor)

2 Distance refers to the spatial distance between two entities on the 2D grid. The distance between two base pairs in a sequence is referred to as the base distance.

Figure 2. An identified gene in a DNA sequence. The promoter sequence (green) determines the gene's starting point, and the terminator sequence (purple) specifies the end point. Genes have enhancer and inhibitor regulatory sites that are upstream of the promoter sequence (brown and yellow).

immediately following the promoter region of the DNA. In our computational case, aimed at generating protein concentration dynamics, no explicit functions could correlate with these regulatory site sizes. Thus an arbitrary decision was made to embed this size in the DNA sequences of each model. The length of these regions depends on the gene's length. Genes with longer sequences have larger regulatory sites, and vice versa. The sequence of bases between the inhibitor and terminator patterns of a gene determines the genetic marker of the TF class produced by that gene (Figure 2).

The size of the regulatory sites, including the inhibitor, enhancer, and protein, is the same and is calculated using the following equation:

$$\text{size} = \lfloor \sqrt{L} \rfloor$$

where L is the length of the sequence between the promoter and the terminator sequence. Note that unlike enhancer or inhibitor regions, proteins are not directly coded into a gene. Instead, a protein sequence is computed from the protein coding region of the gene, which usually is longer than the regulatory sites.

3.1.3 Protein

Proteins are the end products of genes. Genes with higher transcription rates have higher produced protein concentrations. Each gene codes for a specific protein sequence. TF proteins are modeled in the proposed system and are responsible for regulatory actions. The presence of proteins in the system is indicated by both a discrete and a continuous value. The discrete value indicates the count of TFs spread in the 2D grid at each time step (cycle), and the continuous value indicates the protein concentration level produced by a gene. These two values are correlated, meaning that the number of available TFs on the 2D grid created by each gene is proportional to the protein production level of that gene. In each cycle, TFs do random walks in a cell and bind to the regulatory sites of genes when their distance is below a certain threshold value. As the system updates, the concentration levels of proteins will vary based on the network of interactions between genes. This causes the interesting dynamics we observe in the system.

To maintain simplicity and avoid generating excessively long protein sequences, protein lengths are adjusted to match those of other regulatory sites (enhancer and inhibitor regions). Similar to patterns in nature, the length of the resulting proteins correlates with the length of the gene exons. The genetic marker of proteins is determined using a majority rule based on the protein coding region located between the inhibitor and the terminator regions. Figure 3 illustrates how the protein sequence is determined. In our simulations, the length of the region between the promoter and the terminator sequences is $L = 33$, and therefore the size of the regulatory regions is $S = \lfloor \sqrt{33} \rfloor = 5$. The protein coding region (surrounded by a dashed rectangle in Figure 3) has a size of 23. First, this region is divided into S chunks with the size of $\geq N$ using the following formula:

$$N = \lfloor \frac{(L - 2 \times S)}{S} \rfloor$$

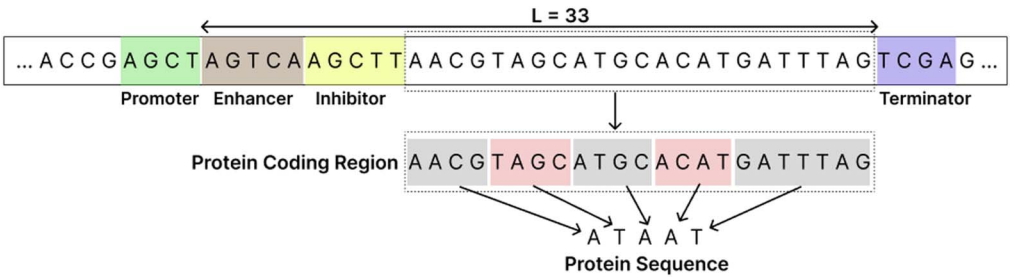


Figure 3. An example for calculating the protein sequence of a gene. The last chunk can be larger, if necessary.

where N is the length of each chunk, except for the last one; S is the size of the regulatory sites; and L is the total length of the gene sequence between the terminator and the inhibitor regions. The majority rule applies to each chunk in such a way that in each case, the nucleotide with the highest frequency of occurrence gets selected as a base in the protein sequence. In the case of a tie, the base with the highest frequency that occurs first in the chunk gets selected. Following the preceding formula, each chunk will have a size equal to N, except for the last chunk, which can have all the remaining bases of a protein coding region. However, the size of the last chunk will always be less than $2 \times N$.

3.2 Set of System Rules of Interaction (R)

Entirely modeling the transcription and translation process computationally seems unnecessary. So in each cycle of the regulation, first, the transcription rates of genes update with regard to the number of TF proteins bound to that gene's regulatory sites. The next step is the moving phase, during which all TFs can move around in the artificial cell by a random walk. Next, TFs that are within the binding range of regulatory sites can bind to those sites. The binding strength is calculated by counting the number of base–base bindings of the regulatory site's sequence and the TF's sequence. Here base A binds only to T and base G binds only to C. To simplify the measurement of binding strengths, we exclude weak bindings. If the two sequences are not of the same length, any extra bases in the longer sequence are ignored. If the binding strength is zero, meaning that no AT, TA, GC, or CG base–base binding could be found, binding simply does not happen. The binding strength indicates for how many cycles the bound TF alters the transcription rate of the gene until the binding expires. Figure 4 illustrates the AC binding method used in the proposed system. Also, TFs cannot bind to their producing gene. If no binding occurs, each artificial gene produces proteins at a minimal rate. However, in the case of binding, the transcription rate or protein concentration produced by that gene might vary during that cycle, depending on the site (enhancer or inhibitor) to which the TF is binding. Multiple TFs can bind to the same regulatory site.

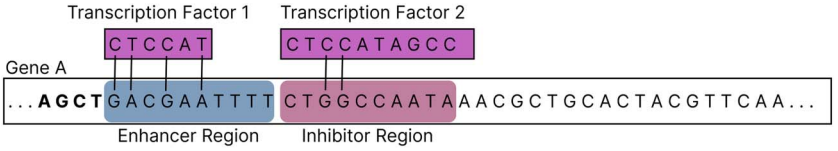


Figure 4. Binding between TF and regulatory sites of two genes. These artificial bindings occur similarly to DNA nucleotide hydrogen bindings. The number of base–base bindings determines the binding strength. On the left side, TF 1 is connected to the enhancer region of gene A with a binding strength of 4 that lasts for four cycles. TF 2 is connected to the inhibitor region of this gene with a binding strength of 2 that lasts for two cycles. The ending of gene A is not depicted in the figure.

The impact of the *TF-enhancer* and *TF-inhibitor* bindings on the translation rate of the respective gene is calculated using the following formulas:

$$R_{i,t+1} = R_{i,t} + \frac{1}{N}\sum_{j=1}^{N} e^{\beta \times (S_{i,j} - S_{\text{total}} - 1)}$$

(TF-enhancer)

$$R_{i,t+1} = R_{i,t} - \frac{1}{N}\sum_{j=1}^{N} e^{\beta \times (S_{i,j} - S_{\text{total}} - 1)}$$

(TF-inhibitor)

where $R_{i,t}$ refers to the transcription rate of gene i at cycle t, N is the total number of bindings to gene i, β is an arbitrary parameter, and $S_{i,j}$ and S_{total} is the binding strength between the regulatory site of gene i and TF j and the strongest binding strength witnessed in the cycle, respectively.

At the end of the regulation cycle, protein concentrations update with the following formula:

$$C_{i,t+1} = C_{i,t} + \delta \times C_{i,t} \times R_{i,t}$$

where $C_{i,t}$ denotes concentration of protein i at cycle t, δ is an arbitrary parameter, and $R_{i,t}$ is the transcription rate of gene i at time t. After calculating the new protein concentrations, these values for each gene are normalized by dividing them by the total concentration of all proteins to keep the sum of the concentration levels equal to 1 at all times. The normalization step aims to simulate the cell's limited resources, inducing competition among concentration levels and giving rise to intriguing system dynamics.

3.3 The Algorithm (A)

Before the regulatory cycles start, the grid is initialized, and the positions of all entities are determined. In each cycle, if a TF is bound (with binding strength > 0), the transcription rates of the corresponding gene are updated based on the regulatory site to which the TF is connected. If a TF is not bound, it randomly moves around the 2D grid during the movement phase. Next, the distances between each TF and the regulatory sites of each gene are measured. If this distance is smaller than a specified binding threshold, a binding between the two entities occurs. Finally, the protein concentration of each gene is determined and normalized, and the count of TFs is updated. Algorithm 1 summarizes these steps.

4 Results

A series of experiments are conducted using the studied GRN model to show the varying protein dynamics produced by such systems as well as to study how the initial states of the system impact the produced dynamics and how these dynamics can evolve. Different sets of initial parameters and configurations are used to run these experiments and are described for each experiment separately.

Algorithm 1. Algorithm of the proposed GRN model.

```
Data: TF_list, gene_list, binding_threshold
Result: Artificial Gene Regulatory Network
initialize_grid( )
cycle ← 0
while cycle < max cycle do
    for TF in TF_list do
        # Update Transcription Rates
        if TF.binding_strength > 0 then
            update_rate(TF.bound_gene)
            TF.binding_strength ← TF.binding_strength - 1
            if TF.binding_strength == 0 then
                remove_tf(TF)
                create_tf()
            end
            continue;
        end
        # Movement Phase
        random_walk(TF)
        # Binding Phase
        for gene in gene_list do
            if distance(TF, gene) < binding_threshold then
                bind(TF, gene)
            end
        end
    end
    # Production/Translation Phase
    for gene in gene_list do
        update_concentration(gene)
    end
    cycle ← cycle + 1;
end
```

4.1 Varying Protein Dynamics

It is difficult to systematically quantify and classify the produced dynamics regarding their complexity, similarity, or stability. Nonetheless, an attempt has been made to handpick and introduce a few of the outstanding patterns observed during our experiments. These dynamics are handpicked from a large pool of randomly generated GRNs and represent single replicates of an experiment. Table 1 summarizes the parameters used for the experiments conducted in this section. The effects of changing most of these parameters are studied in this article. However, the initial conditions are selected experimentally in a manner that facilitates easier observation of the dynamics in 1,000-cycle experiments.

Figure 5 illustrates generated protein concentration dynamics belonging to four classes of patterns. In some cases, protein concentrations develop over time following simple patterns (Figure 5(a)). A shared characteristic of such dynamics is the smooth development of protein concentrations to reach a stable state where the concentrations do not vary rigorously over time. Smooth and oscillatory dynamics are the most predominant types of produced protein dynamics in the studied system. In Figure 5(a) a competition between production levels can be observed, with Protein 2's concentration level slowly increasing over time, while this value decreases for Proteins 1 and 3.

Table 1. Experimental parameters used for generating protein dynamics.

Parameter	Value	Description
Cycles	1,000	Regulation time cycles
Grid size	10	Size of the 2D grid, a size of 10 results in a 10 × 10 grid space
Initial TF count	25	Indicates how many TFs for each gene; initially put on the 2D grid to start the dynamics
Starting concentration	1 / number of genes	Initial protein concentration value value for each gene
Step size	5	Step size of TFs used for randomly walking in the 2D grid
Beta	1	Arbitrary parameter to control production rates
Delta	1	Arbitrary parameter to control production rates
DNA length	3,000	Number of bases in the initial DNA sequence used for identifying genes

Even though competing dynamics is a common behavior of the studied system, sometimes this competition stabilizes in a way that no concentration level changes over time anymore.

Figure 5(b) shows an oscillatory dynamic in which one or more proteins produce a repeated pattern of altering concentration levels. This is often accomplished by two TF types competing to achieve higher production levels. The most apparent competing interaction to form the oscillatory behavior of this figure is perhaps between Protein 8 and all the other proteins. Increasing levels for Protein 8 often cause other protein levels to decrease. However, this competing interaction is not the sole reason for the dynamics of Figure 5(b) to appear. For example, at approximately 200 time cycles, the concentration level of Protein 7 (pink) increases, while all other protein concentration levels (including for Protein 8) decrease. Unlike in the complicated case of Figure 5(b), there are times at which oscillation can be explained with obvious competing concentration dynamics. Figure 6 shows a case in which a somewhat chaotic oscillatory behavior is caused by only two competing proteins.

Figure 5(c) illustrates a hybrid behavior in which both oscillation (concentration levels of Proteins 4 and 6) and simple development can be observed. For approximately 170 cycles, a smooth development is apparent in the system up until the concentration levels for Proteins 4 and 6 reach an equal level. Crossing this junction triggers an oscillatory dynamic between the two proteins, defining a hybrid class of regulatory dynamics.

Finally, Figure 5(d) shows a more chaotic dynamic behavior. Until around Cycle 230, the network produces oscillatory dynamics between Proteins 3, 2, and 7 that seem to be stabilizing; however, the dynamics change to a different type of oscillation after this cycle with ostensibly unique intervals. Unlike Figure 5(c), the reason for the sudden change in the dynamic illustrated in Figure 5(d) is not obvious.

Figure 7 gives an example of a simple development dynamic in which, after a certain number of regulatory cycles, the concentration levels reach a steady state and do not change over time. In this figure, protein levels compete for the first 100 time cycles, then remain constant after Protein 4 disappears.

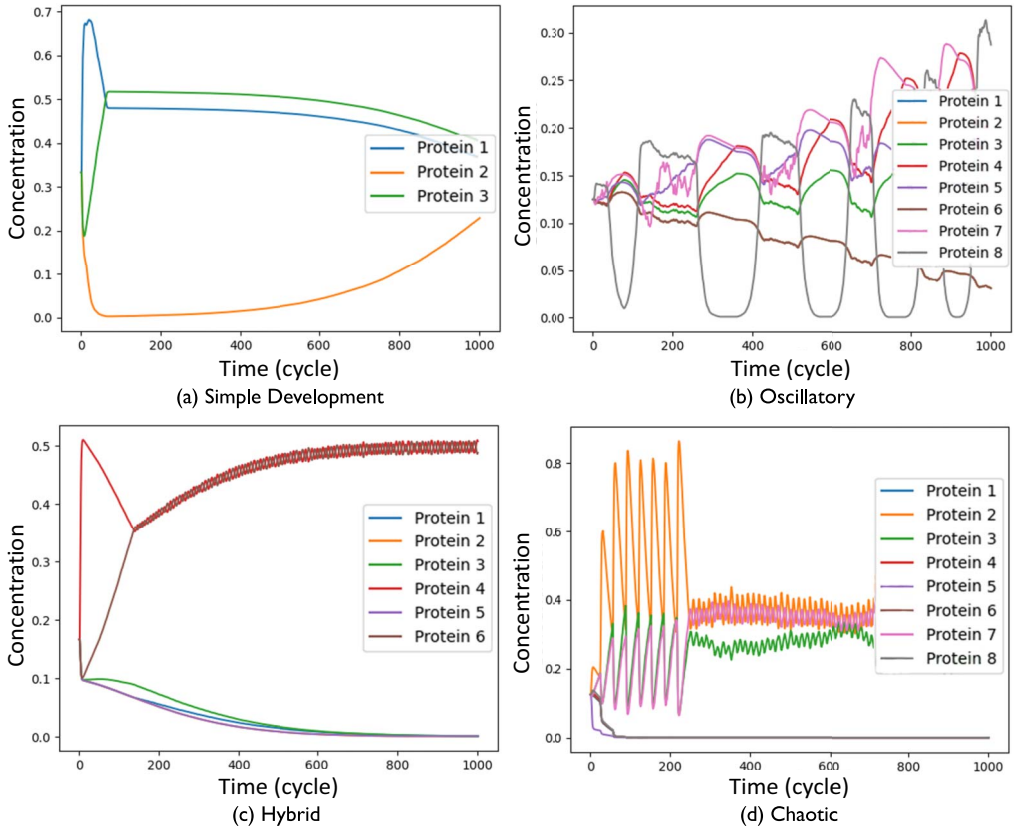


Figure 5. Different types of observed dynamics produced by the proposed gene regulatory network (GRN).

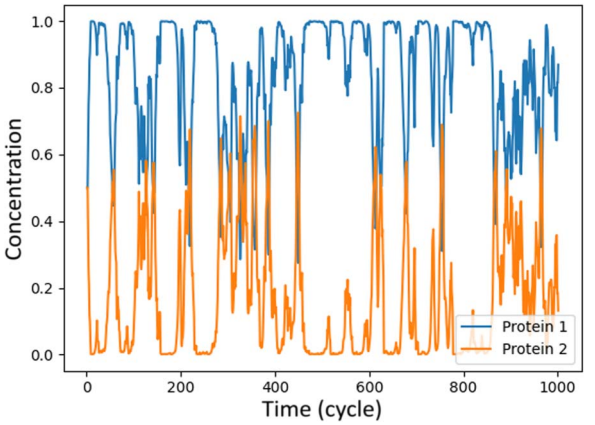


Figure 6. An oscillatory dynamic caused by competition between only two proteins.

Table 2 shows the distribution of the dynamics for each type defined in Figure 5(c) out of 50 randomly generated dynamics. Oscillatory dynamics are the dominant class, comprising 58% of the total generated plots, followed by simple (20%), hybrid (12%), and then chaotic (10%). Only 6 out of the 10 simple dynamics reach a steady state after 500 regulatory time cycles.

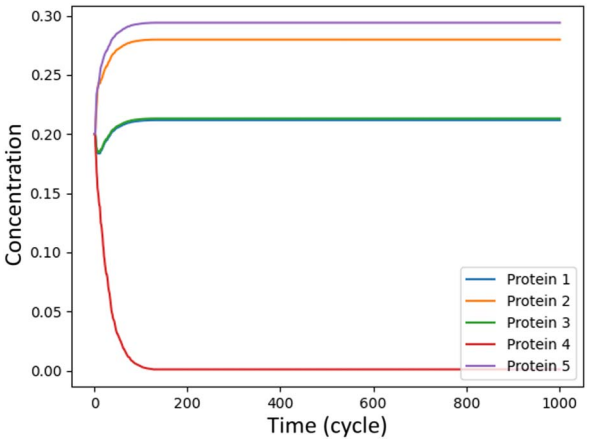
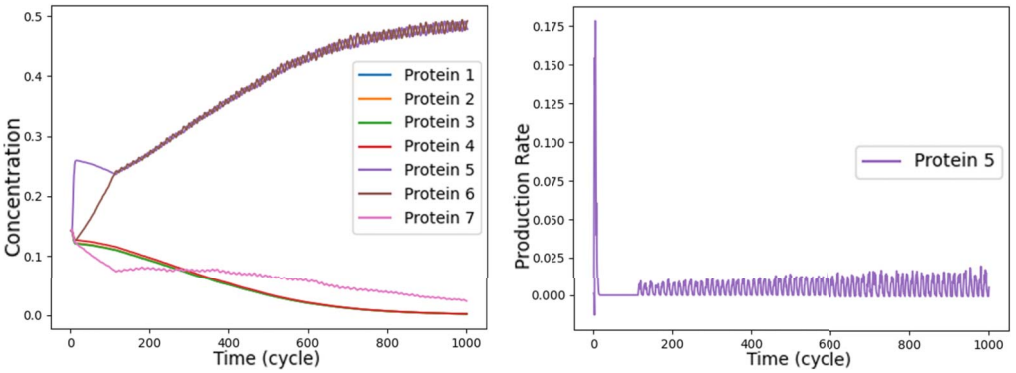


Figure 7. A simple development dynamic in which all the production levels stabilize after a certain number of regulatory cycles.

Table 2. Distribution of each dynamic type in 50 randomly generated dynamics.

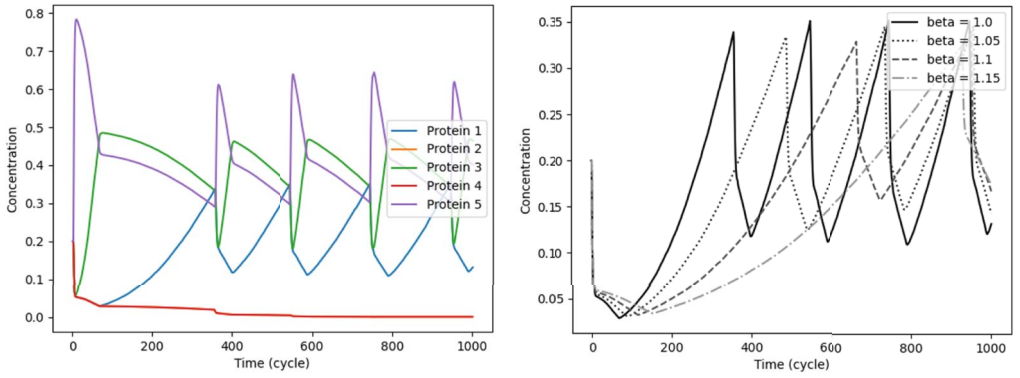
Dynamic	Count	Percentage
Simple	10	20
Oscillatory	29	58
Hybrid	6	12
Chaotic	5	10



(a) Protein Dynamic (b) Production Signals

Figure 8. Dynamics versus transcription rate. (a) A hybrid-produced protein dynamic. (b) Transcription rates produced for Protein 5 over time.

Figure 8 illustrates the protein dynamics (Figure 8(a)) and the production rates/signals over time responsible for regulating these dynamics of Protein 5 (Figure 8(b)). The production rate increases if more and stronger TF binding happens in the enhancer region of the gene compared to the inhibitor region. If no binding happens, the production rate will be close to zero. The first few bonds result in a more intense increase/decrease in production rates because other TFs need also to move and spread in the cell to start stabilizing the network of interactions (see the increase in the production level of Protein 5 in Figure 8(b) during the first 10 cycles). In Figure 8(b), this is followed by a steady



(a) Base Dynamic (b) Protein 1 Changes

Figure 9. The impact of changing β on the produced dynamics. (a) Base dynamics. (b) Changes to Protein 1 for different values of β.

no-production state for 100, cycles in which a drop in protein concentrations can be noticed in the protein dynamics. No production signals or rates less than 0 can be considered equivalent to natural genes not being expressed or turned off by repressors. After Cycle 150, a sudden switch occurs to another regime: an oscillatory behavior for Protein 5 that correlates with the oscillatory patterns of the production rates.

4.2 Impact of Initial States on the System Dynamics

In this section, the impact of the initial states and parameters of the system on the outcome of the protein dynamics is studied. In each replicate of this experiment, a genome is randomly initialized with the parameters described in Table 1, and its genes are identified. Using the same genome, multiple GRNs are constructed in which a single initial state parameter is altered. We compare and differentiate the protein dynamics produced by these GRNs. It is important to note that the random movement of TFs plays a critical role in generating system dynamics. To keep the comparisons fair, the random state for the movement of TFs is preserved in all cases. This experiment is conducted for 50 replicates. Because it is not feasible to include the figures for all the replicates, we depict only one GRN system for each case. In the event that we observe irregularities for any of the other replicates, those irregularities are also discussed.[3]

4.2.1 Parameter β

Parameter β is a strength parameter that can be used to control the intensity of the inhibitory and enhancing signals. The value of β is initially set to 1 for most experiments. Figure 9(a) shows a protein dynamic selected as a base dynamic to compare the impact of changing this parameter. Figure 9(b) illustrates how different β values change the dynamics for Protein 1. With an increase in the β value, a time shift in the generated patterns can be seen in such a way that the same patterns happen later in the regulatory cycles. In other words, increasing β expands the produced dynamics.

4.2.2 Parameter δ

This parameter controls the intensity of protein production and is multiplied by the production signal in each development cycle. Similarly to β, the initial value of δ is usually a default of 1. We use the same base dynamic depicted in Figure 9(a) and the result of comparing different δ values and their impact on Protein 1 is depicted in Figure 10. As the value of δ decreases, a shift in time for the generated patterns can be seen. In other words, lower δ values expand the produced dynamics,

3 Graphs associated with the rest of the replicates, along with code to replicate the results, additional design information, and experimental parameters, can be found at https://github.com/elemenohpi/AC-ARN-ArtificialLife.

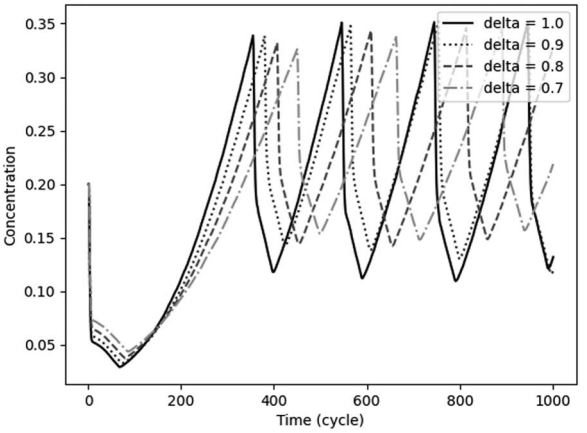


Figure 10. Changes to Protein 1 for different values of δ.

whereas higher δ values shrink it. This impact is the opposite of the impact of β on the dynamics. Unlike for β, the concentration levels do not seem to change as much, and the scale of the dynamics remains closer to the original.

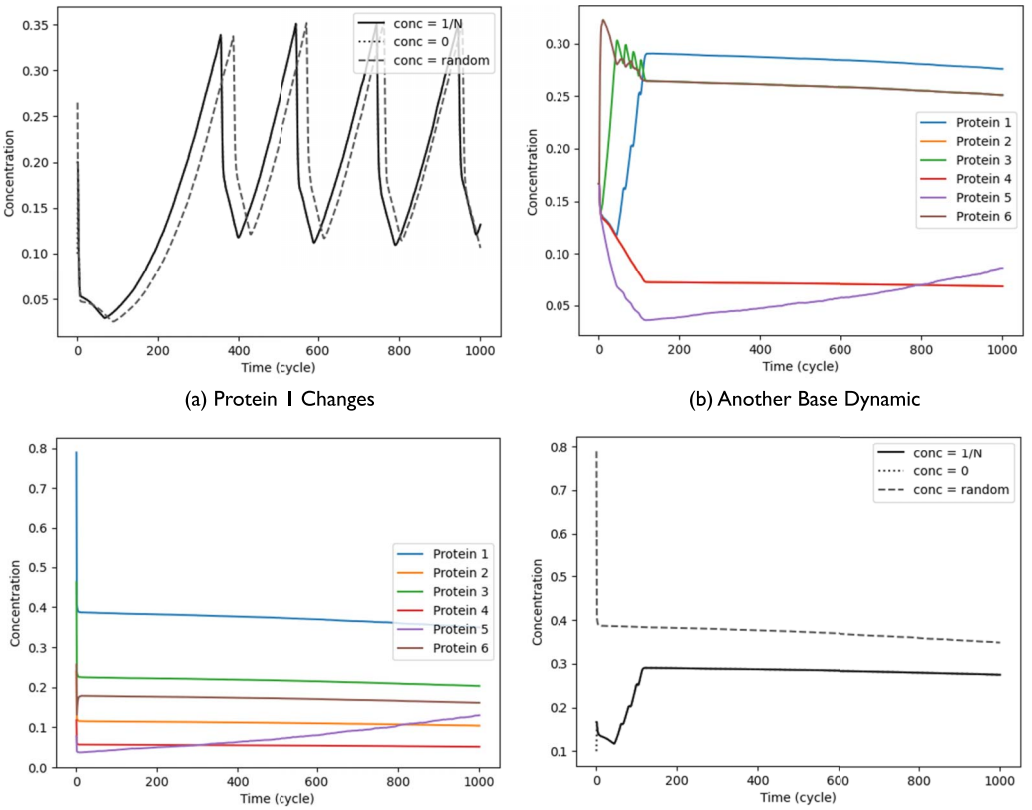
4.2.3 Initial Protein Concentration Levels

In the natural context, small, spontaneous expressions frequently initiate regulatory dynamics. In this study, we initiated these dynamics by configuring the proteins' initial concentration levels to $1/N$ (where N represents the gene count) for the majority of our experiments. Here an experiment was conducted to see the impact of these initial conditions by trying 0 and random initial levels. For the cases in which initial concentrations are set to 0 and $1/N$, plot lines overlap and no changes to the produced dynamics can be observed (Figure 11(a) and (d)). When setting initial concentration levels to random values, usually, small transformations can be observed. However, that was not the case for all the replicates. Figure 11(b) shows another base dynamic in which the initial concentration levels for proteins are set to $1/N$. Figure 11(c) depicts the same GRN but with randomly initialized concentration levels. It is apparent that the oscillatory dynamic of Proteins 3 and 6 in the period between 0 and 150 time cycles is gone and that the order of highest to lowest protein levels has altered, portraying a different dynamic. Figure 11(d) shows how the change in concentration levels impacts Protein 1 over time.

4.2.4 Initial TF Count

Another influential factor is the initial number of TFs per gene, impacting the resulting dynamics. To explore this, we conducted two experiments. Initially, we assigned different numbers of starting TFs to each gene (with an equal TF count for each gene). As visible in Figure 12, changing the TF counts leads to shifts in both time and scale of the resulting dynamics. Importantly, unlike in the β and δ cases, the time shift seems somewhat random, occurring in both directions.

Subsequently, we examined the effect of introducing a single TF molecule to the overall protein dynamics of the system. Figure 13(a) illustrates the protein dynamics generated by a GRN where each gene has only a single TF molecule in the spatial grid. Conversely, Figure 13(b) depicts a distinct dynamic generated by the same GRN, with an additional TF molecule produced by Protein 1 available in the system at any given time. The resulting dynamic differs notably from the original, with altered sequences of protein concentration levels (Protein 3 now behaving completely differently). In Figure 13(b), the concentration levels of Proteins 1 and 4 are overlaid. Repeating this experiment 50 times revealed that sometimes the change in dynamics is as subtle as a small time or scale shift. However, when the extra TF is assigned to a gene with a low count in the system, the impact is more pronounced.



(a) Protein 1 Changes

(b) Another Base Dynamic

(c) Dynamic for Randomly Initialized Concentrations

(d) Protein 1 Changes

Figure 11. The impact of changing initial protein concentration levels on the produced dynamics. (a) Changes to Protein 1 for different concentration levels for the same protein dynamic as Figure 9(b). (b) Dynamics produced by another GRN model with concentration levels equal to $1/N$. (c) Dynamic of the GRN system of (b) when concentration levels are randomly initialized. (d) Changes to Protein 1 for different concentration levels for the GRN of (b).

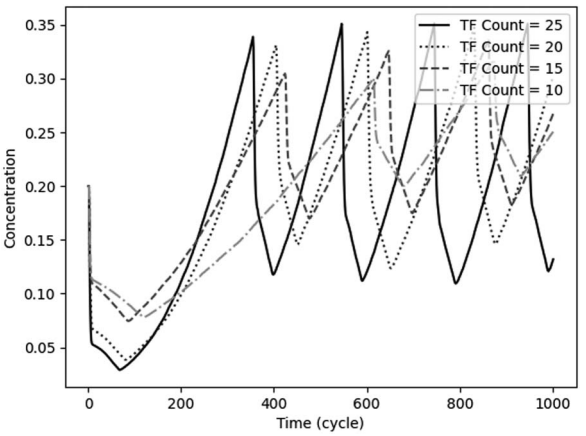
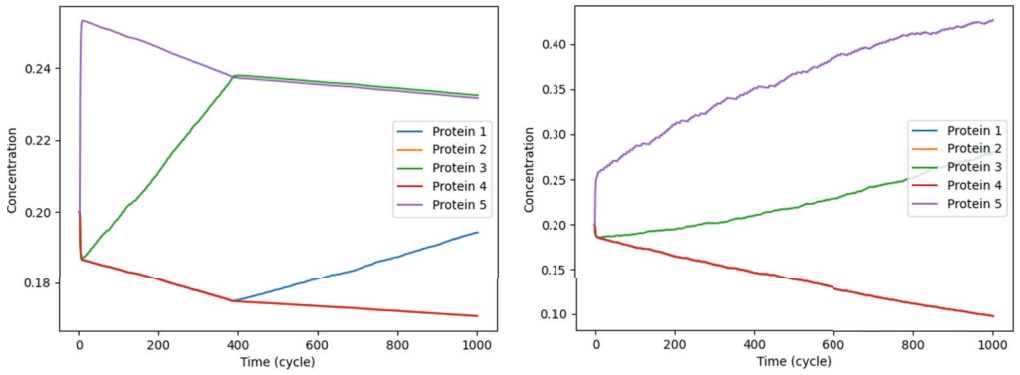


Figure 12. Changes to Protein 1 for different TF counts.



(a) A GRN Dynamic (b) Dynamic After Adding an Extra TF

Figure 13. (a) A GRN dynamic whereby every gene only has only a single TF molecule in the spatial grid. (b) Impact of adding an extra TF produced by Gene 1 to the system.

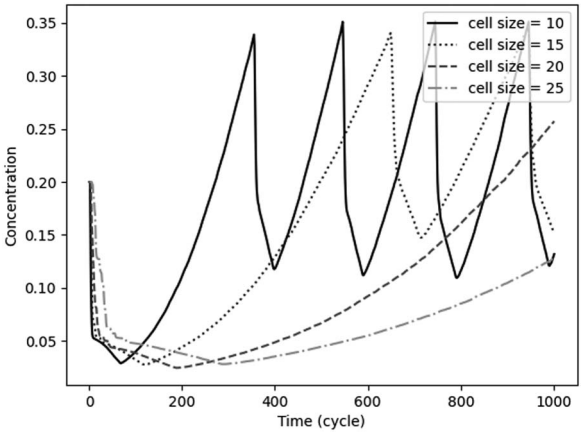


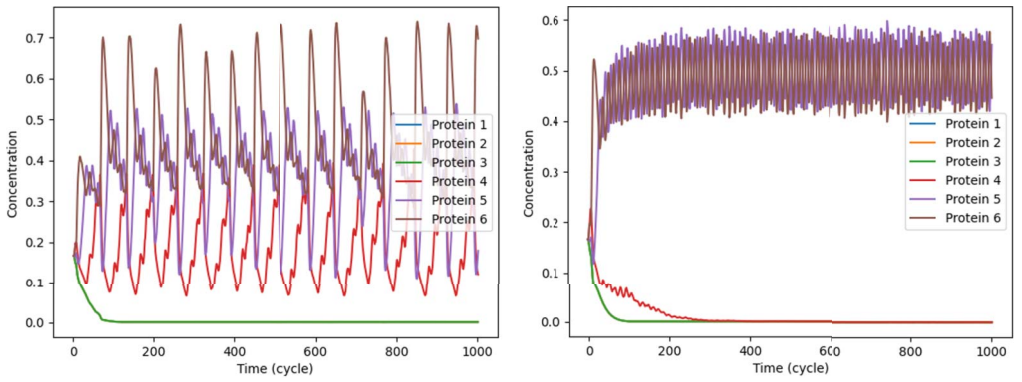
Figure 14. Changes to Protein 1 for different sizes of the grid.

4.2.5 Cell/Grid Size

Similarly to the case for different TF values, a shift in time and scale can be seen for different grid sizes (Figure 14). The randomness in the scale and time shift is due to the random movements of the TFs. The larger the cell is, the longer it takes for the TFs to spread in the grid and help the network to stabilize.

4.2.6 Changing Spatial Position of the Regulatory Sites

The proposed system displays robustness. Altering the initial states mostly maintains consistent generated patterns, noticeable as shifts in time or scale. However, minor adjustments to the positions of regulatory sites within the 2D grid can exert a noticeable influence on the resulting dynamics. This suggests that the system's robustness may be somewhat reduced when altering the spatial positions of regulatory sites. Figure 15(a) shows the dynamic produced by a network with six genes. In Figure 15(b), the position of the enhancer region of Gene 1 is slightly changed, resulting in a different pattern of dynamic. In most cases, a small change in the spatial position of the regulatory sites of a gene results in significant changes in the produced dynamics. However, instances arise in which dynamics shift only slightly or remain unchanged. This occurs primarily because, in such cases, the pattern of available TFs in the system does not match with the regulatory site that has



(a) A GRN Dynamic (b) Dynamic After Spatial Mutation

Figure 15. (a) A GRN dynamic. (b) Impact of performing a spatial mutation on the enhancer region of Gene 1.

been spatially repositioned. As previously mentioned, the system's random state is preserved even when the spatial positions of the regulatory sites change. In other words, the TFs maintain the same random movement routes in every scenario. When a regulatory site is relocated, it is no longer in proximity to the same number of TFs, thus altering the total number of protein bindings at that site and resulting in significant changes to the dynamics produced.

4.3 Evolution of Dynamics

So far, the nature of the proposed system has been explained, and the different dynamics produced from random genomes generated from random seeds have been studied. However, to apply the proposed ARN in other applications, it is essential for this system to be evolvable to achieve desired dynamics. In this section, two experiments are conducted to evolve regulatory networks that meet a specific dynamic criterion. To evolve these networks, a simple GA was used that alters the initial DNA genotype of each individual. The utilized GA consists of a population of genomes, point mutations, and a one-point crossover, with a tournament as selection mechanism. For both evolutionary experiments, a population size of 20, a mutation rate of 0.15, and a tournament size of 3 are configured. Each experiment was run for 25 generations.

In Problem 1, the goal is for Protein 1 to reach a 0.085 concentration level at Cycle 100. Figure 16(a) shows the evolutionary results for this problem. The x axis is time, and the y axis is the deviation from the goal concentration in the form of absolute error. Therefore lower values indicate a better individual. The depicted line represents the median fitness for the experiment's 10 parallel runs, and the shaded areas represent the 75 and 25 quantiles. Figure 16(b) illustrates one of the evolved solutions for solving this problem. Several proteins share the same concentration level.

In Problem 2, the goal is for Proteins 1 and 2 to alternate in concentration level every 50 cycles so that in the starting period, if Protein 1 has more concentration than Protein 2, the individual will be rewarded with 1 point. The individual receives another reward if, in the next period, Protein 2 has more concentration than Protein 1. The same level alteration process should continue for 10 cycle periods to achieve the maximum reward of 10. This is a more challenging task than Problem 1, and the considered fitness function based on discrete rewards does not provide significant pressure toward solving the problem. Figure 16(c) shows the evolutionary results for solving Problem 2. Although the median of individuals does not solve the problem, some cases fully solve it in 25 generations. Figure 16(d) shows a perfect solution to the problem, achieved during evolution by one of the runs with fitness equal to 10.

The next step is to look more closely at the solutions that can evolve in the system to address the two respective problems. Figure 17(a) illustrates another solution for Problem 1 that consists of fewer proteins. Unlike the previous solution, which comprised mostly simple developments of

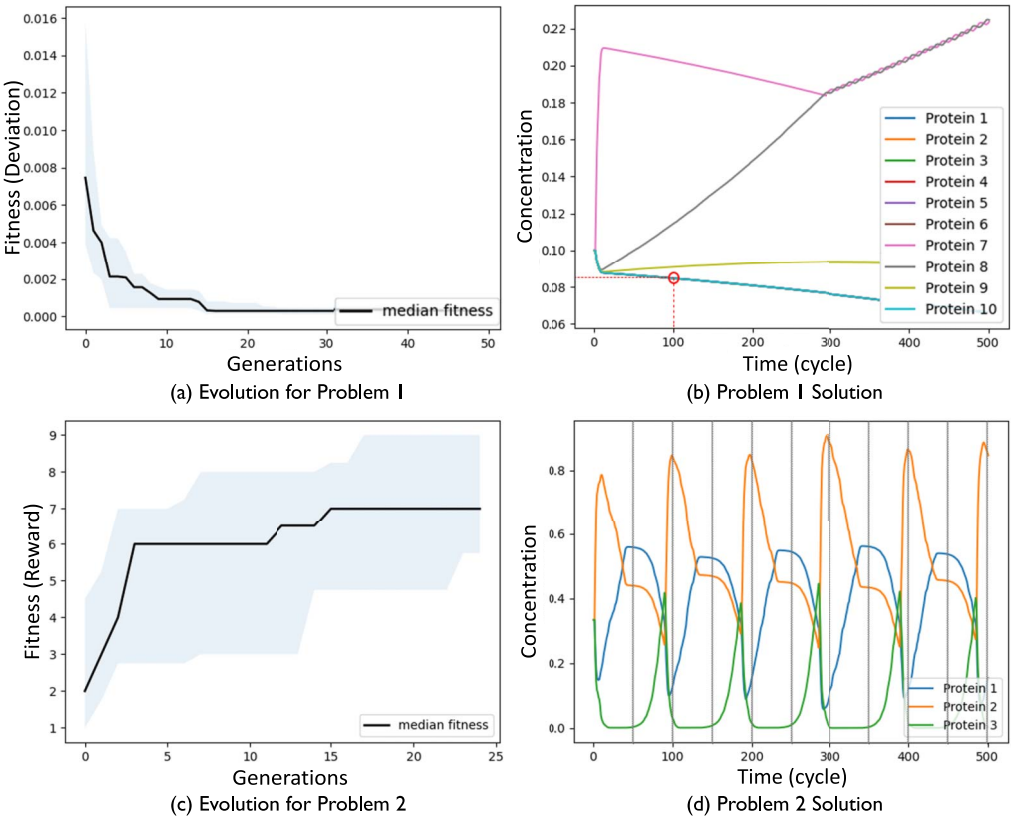
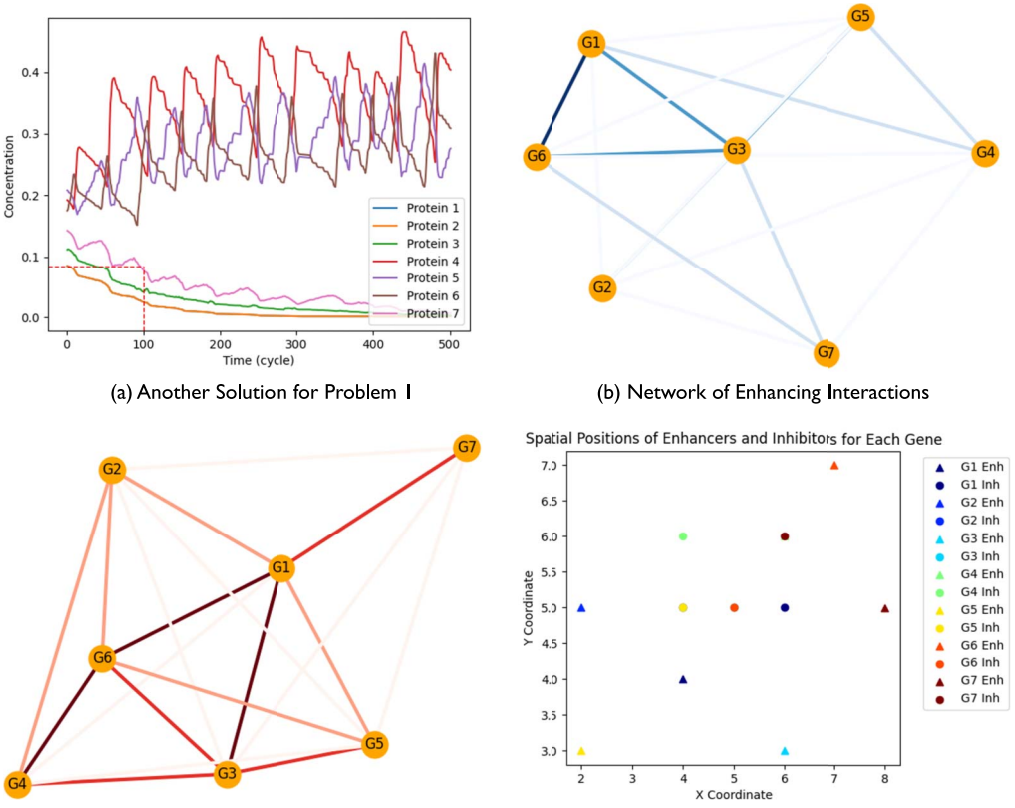


Figure 16. Evolution of dynamics. (a) Fitness (deviation) over generations for solving Problem 1. (b) A solution evolved to solve Problem 1. (c) Fitness (reward) over generations for solving Problem 2. (d) A solution evolved to solve Problem 2.

protein dynamics with subtle fluctuations, the dynamics of the solution in Figure 17 fluctuate more rapidly, forming an oscillatory dynamic. Figure 17(b) and (c) show the enhancing and inhibitory networks of interactions built from the solution's genome, respectively. Each node represents a gene, and the intensity of the color of each edge represents the inhibiting or enhancing impact of the two connecting genes on one another. Specifically, the presence of Genes 1, 3, and 6 seems to influence a higher enhancing impact on each other, while many genes show high inhibitory intensity toward each other. Figure 17(d) shows the spatial positions of the enhancer and inhibitor regions of every gene in the solution genome. Note that (5, 5) is the center of the 2D space. Each gene is distinguished by a unique color. Triangles represent enhancing regions, whereas circles represent inhibiting regions. The enhancer regions of Genes 4, 5, and 6 (with Genes 4 and 5 having overlapping enhancer regions) are notably closer to the edge of the 2D space representing the cell, where the TFs are introduced into the system. It is evident that these three genes also exhibit higher protein production compared to the rest. Such behavior is typical for the initial regulatory steps of protein dynamics with more than a few genes in most of the produced networks. However, unlike what is observed in Figure 17(a), the proteins that start with higher production values do not always remain the most highly produced ones.

Figure 18 comprises four charts showing the dynamics, the enhancing and inhibiting networks of interaction, and the spatial organization of another solution for Problem 2. This solution achieves a fitness of 9. Figure 18(a) shows the dynamics produced by this solution's genome. This GRN



(a) Another Solution for Problem 1

(b) Network of Enhancing Interactions

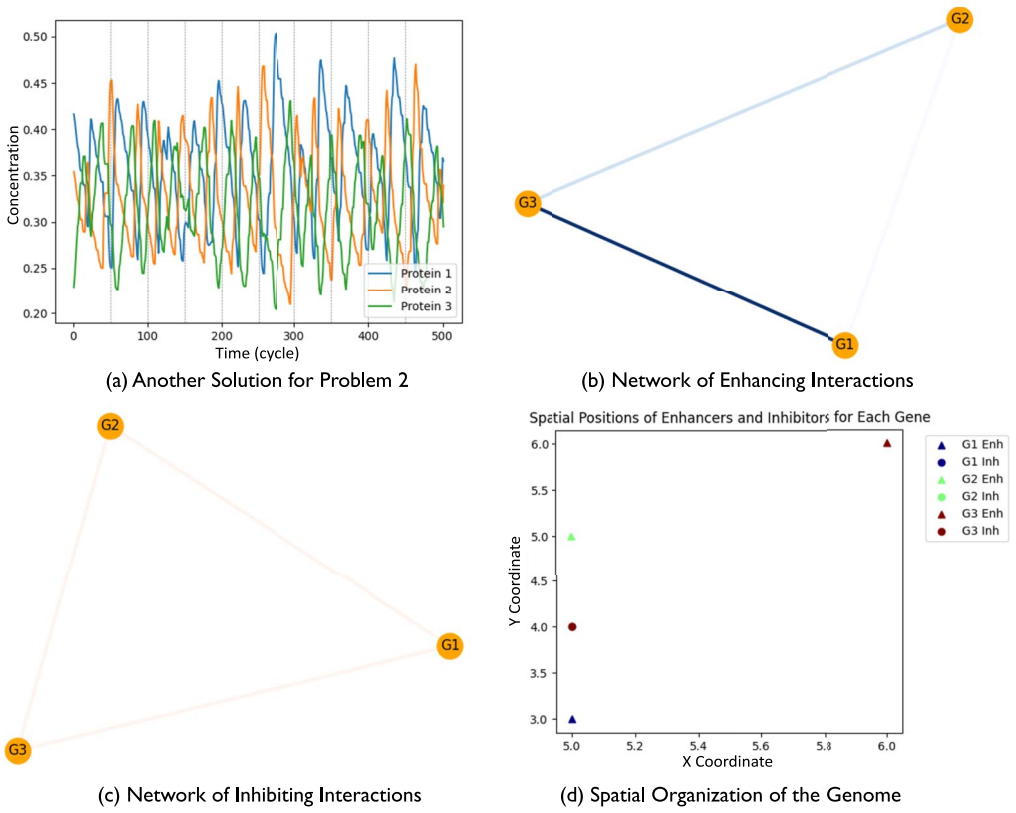
(c) Network of Inhibiting Interactions

(d) Spatial Organization of the Genome

Figure 17. A different solution for Problem 1.

dynamic consists of only three proteins, which fluctuate nonstop during the 500 regulatory cycles. Figures 18(b) and (c) illustrate the enhancing and inhibiting networks of interaction for this genome. Although there seems to be a somewhat similar intensity of inhibition between any pair of genes, Genes 1 and 2 appear to enhance each other's production more. Figure 18(d) shows the spatial organization of this genome. Notably, the three inhibiting regions overlap in the same spatial location. Similar to earlier (5, 5) represents the center of the 2D space; therefore all of the regions are located relatively close to the center of the 2D space. A solution for Problem 2 is expected to be oscillatory and therefore the two solutions for this problem are of the same type. However, the intensity of oscillation in the second solution, caused by the interaction of the three genes, is more evident.

Last, we conduct an experiment in which an attempt is made to depict the impact of point mutations on the outcome of the dynamics. Mutations occurring on different sites result in different behaviors, and their impact highly depends on the rest of the expressed genome. A single point mutation on the protein-coding site can sometimes completely change the system's dynamics, whereas at other times, it serves as a neutral mutation. In general, during the experiments, three significant outcomes from mutations could be observed in the system: (a) a neutral mutation, (b) a complete change in the dynamics, and (c) a shifts in the scale and time of the dynamics. Figure 19 shows the concentration levels of Protein 1 of the GRN dynamic depicted in Figure 9(a) that is undergoing zero to four point mutations on the regulatory sites of its expressed genes. In the case of only one mutation, the concentration dynamic completely changes. The oscillatory dynamic of Protein



(a) Another Solution for Problem 2

(b) Network of Enhancing Interactions

(c) Network of Inhibiting Interactions

(d) Spatial Organization of the Genome

Figure 18. A different solution for Problem 2.

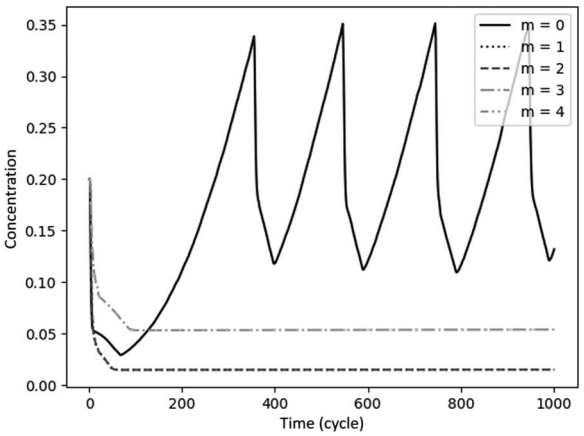


Figure 19. A comparison between the impact of different numbers of mutations on expressed genes. Mutations 1 and 2 and Mutations 3 and 4 have overlapping dynamics.

1 turns into a simple dynamic. In the case of two mutations, no changes can be observed in the protein dynamic, and the lines overlap. For the third mutation, a shift in concentration levels can be observed. Finally, the fourth mutation is a neutral mutation. Needless to say, mutations on the genome outside regulatory sites might change the total number of identified genes for a genome.

5 Discussion and Conclusion

In this article, a spatial and biologically close model of GRNs was introduced based on the work of Banzhaf (2003). A 2D grid was utilized to introduce spatial properties to the system. The rules of interactions between proteins and regulatory regions were defined by an AC. The AC adds molecular aspects to the system. Our results show that the protein dynamics produced are close to their biological counterpart, and a classification of these dynamics was performed. The impact of initial states on the produced dynamics and how they can help control the outcome were explored. An interesting take on these experiments is the controllable heterochrony in the proposed system created by the spatial implementation. Changing the number of TFs or the grid size causes a time shift in the resulting dynamics because it impacts the probability of TFs binding to regulatory sites. The results indicate that the proposed system is highly robust: Changing most of the initial states of the system does not change the dynamics produced. However, a slight change in the spatial position of the regulatory sites on the 2D grid or addition of a single TF molecule to the system can drastically change these dynamics, which could be used as a means for providing inputs to the system. We employed a standard evolutionary algorithm to solve two simple problems of state specification at specific cycle periods. Our findings indicate that different types of solutions can be found for the same problem. Finally, the impact of the mutation on the produced dynamics was studied, which showed high evolvability of such a system. In the future, techniques like dynamic time warping and compression-based dissimilarity measures will be used to analyze and differentiate the produced dynamics systematically.

ARN representations were previously used as direct and indirect representations for genetic programming. It would be worthwhile to try the representation studied here for genetic programming to solve more sophisticated computational problems. Previously, special gene types were introduced for inputs and outputs as the dominant approach to using such a system as a problem solver. We believe that the spatial positioning of entities in this AC-ARN could serve as a novel method to introduce I/O to the system. Although this work focuses primarily on producing and comparing the protein dynamics of the system, performing a quantitative analysis on the dynamics' evolution using the proposed system presents an intriguing opportunity. Another possible future direction for this research is to study the impact of DNA size and the number of genes on the complexity of the dynamics. It would be interesting to test the following hypothesis: Regulatory networks with more genes produce more complex dynamics. In the future, we aim to investigate in greater detail the evolution of dynamics produced by the proposed representation. Analyzing such individuals could yield valuable insights into using this representation for problem-solving applications, such as in the form of genetic programming. Our results for a different genetic programming system (Miralavy & Banzhaf, 2023), in which individuals are spatial and represented by a collection of computer programs distributed in a 2D space, show that the dimension of space can lead to higher structural diversity, can cause program nodes to form spatially localized clusters, and does not impair performance. Another instance in which space has proved to be beneficial in genetic programming is presented by Dick and Whigham (2013), who showed that introducing a spatial population reduces bloat in the system. It would be interesting to see how a spatial, biologically closer representation would compare for this purpose.

References

Allen, B., Sample, C., Dementieva, Y., Medeiros, R. C., Paoletti, C., & Nowak, M. A. (2015). The molecular clock of neutral evolution can be accelerated or slowed by asymmetric spatial structure. *PLoS Computational Biology*, *11*(2), e1004108. https://doi.org/10.1371/journal.pcbi.1004108, PubMed: 25719560

Arias, C. F., Catalán, P., Manrubia, S., & Cuesta, J. A. (2014). toyLIFE: A computational framework to study the multi-level organisation of the genotype-phenotype map. *Scientific Reports*, *4*(1), Article 7549. https://doi.org/10.1038/srep07549, PubMed: 25520296

Asr, N. R., & Majd, V. J. (2013). A new artificial genetic regulatory network model and its application in two dimensional robot control. *International Journal of Information and Electronics Engineering*, *3*(5), 461–465. https://doi.org/10.7763/IJIEE.2013.V3.358

Astor, J. C., & Adami, C. (2000). A developmental model for the evolution of artificial neural networks. *Artificial Life*, *6*(3), 189–218. https://doi.org/10.1162/106454600568834, PubMed: 11224915

Banzhaf, W. (2003). On the dynamics of an artificial regulatory network. In W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, & J. T. Kim (Eds.), *Advances in Artificial Life* (pp. 217–227). Springer. https://doi.org/10.1007/978-3-540-39432-7_24

Banzhaf, W., & Yamamoto, L. (2015). *Artificial chemistries*. MIT Press. https://doi.org/10.7551/mitpress/9780262029438.001.0001

Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., & Ray, T. S. (2000). Open problems in Artificial Life. *Artificial Life*, *6*(4), 363–376. https://doi.org/10.1162/106454600300103683, PubMed: 11348587

Bentley, P. J. (2003). Evolving fractal proteins. In A. M. Tyrrell, P. C. Haddow, & J. Torresen (Eds.), *International conference on evolvable systems* (pp. 81–92). Springer. https://doi.org/10.1007/3-540-36553-2_8

Bentley, P. J. (2004). Evolving beyond perfection: An investigation of the effects of long-term evolution on fractal gene regulatory networks. *Biosystems*, *76*(1–3), 291–301. https://doi.org/10.1016/j.biosystems.2004.05.019, PubMed: 15351151

Bilotta, E., & Pantano, P. (2005). Emergent patterning phenomena in 2D cellular automata. *Artificial Life*, *11*(3), 339–362. https://doi.org/10.1162/1064546054407167, PubMed: 16053574

Bongard, J. C., & Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. In *Morpho-functional machines: The new species* (pp. 237–258). Springer. https://doi.org/10.1007/978-4-431-67869-4_12

Calladine, C. R., Drew, H., Luisi, B., & Travers, A. (2004). *Understanding DNA: The molecule and how it works* (3rd ed.). Academic Press.

Chaouiya, C., Naldi, A., & Thieffry, D. (2011). Logical modelling of gene regulatory networks with GINsim. In J. Helden, A. Toussaint, & D. Thieffry (Eds.), *Bacterial molecular networks: Methods and protocols* (pp. 463–479). Springer. https://doi.org/10.1007/978-1-61779-361-5_23, PubMed: 22144167

Chavoya, A., & Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, *94*(1–2), 95–101. https://doi.org/10.1016/j.biosystems.2008.05.015, PubMed: 18616980

Cussat-Blanc, S., Harrington, K., & Banzhaf, W. (2019). Artificial gene regulatory networks—a review. *Artificial Life*, *24*(4), 296–328. https://doi.org/10.1162/artl_a_00267, PubMed: 30681915

Cussat-Blanc, S., & Pollack, J. (2012). Using pictures to visualize the complexity of gene regulatory networks. In *ALIFE 2012: The thirteenth international conference on the Synthesis and Simulation of Living Systems* (pp. 491–498). MIT Press. https://doi.org/10.1162/978-0-262-31050-5-ch064

de Jonge, W. J., Patel, H. P., Meeussen, J. V., & Lenstra, T. L. (2022). Following the tracks: How transcription factor binding dynamics control transcription. *Biophysical Journal*, *121*(9), 1583–1592. https://doi.org/10.1016/j.bpj.2022.03.026, PubMed: 35337845

Deakin, M. A. (1990). Modelling biological systems. In T. L. Vincent, A. I. Mees, & L. S. Jennings (Eds.), *Dynamics of complex interconnected biological systems* (pp. 2–16). Springer. https://doi.org/10.1007/978-1-4684-6784-0_1

Dick, G., & Whigham, P. A. (2013). Controlling bloat through parsimonious elitist replacement and spatial structure. In K. Krawiec, A. Moraglio, T. Hu, A. Şima Etaner-Uyar, & B. Hu (Eds.), *Genetic programming* (pp. 13–24). Springer. https://doi.org/10.1007/978-3-642-37207-0_2

Dittrich, P., Ziegler, J., & Banzhaf, W. (2001). Artificial chemistries—a review. *Artificial Life*, *7*(3), 225–275. https://doi.org/10.1162/106454601753238636, PubMed: 11712956

Glass, L., & Kauffman, S. A. (1973). The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, *39*(1), 103–129. https://doi.org/10.1016/0022-5193(73)90208-7, PubMed: 4741704

Gnanakkumaar, P., Murugesan, R., & Ahmed, S. S. S. J. (2019). Gene regulatory networks in peripheral mononuclear cells reveals critical regulatory modules and regulators of multiple sclerosis. *Scientific Reports*, *9*(1), 12732. https://doi.org/10.1038/s41598-019-49124-x, PubMed: 31484947

Hannon, B., & Ruth, M. (1997). Modeling dynamic biological systems. In B. Hannon & M. Ruth (Eds.), *Modeling dynamic biological systems* (pp. 3–27). Springer. https://doi.org/10.1007/978-1-4612-0651-4_1

Hickinbotham, S. J., Stepney, S., & Hogeweg, P. (2021). Nothing in evolution makes sense except in the light of parasitism: Evolution of complex replication strategies. *Royal Society Open Science*, *8*(8), 210441. https://doi.org/10.1098/rsos.210441, PubMed: 34386257

Holcombe, M., Adra, S., Bicak, M., Chin, S., Coakley, S., Graham, A. I., Green, J., Greenough, C., Jackson, D., Kiran, M., MacNeil, S., Maleki-Dizaji, A., McMinn, P., Pogson, M., Poole, R., Qwarnstrom, E., Ratnieks, F., Rolfe, M. D., Smallwood, R., … Worth, D. (2012). Modelling complex biological systems using an agent-based approach. *Integrative Biology*, *4*(1), 53–64. https://doi.org/10.1039/C1IB00042J, PubMed: 22052476

Hutton, T. J. (2002). Evolvable self-replicating molecules in an artificial chemistry. *Artificial Life*, *8*(4), 341–356. https://doi.org/10.1162/106454602321202417, PubMed: 12650644

Jin, F., Li, Y., Ren, B., & Natarajan, R. (2011). Enhancers: Multi-dimensional signal integrators. *Transcription*, *2*(5), 226–230. https://doi.org/10.4161/trns.2.5.17712, PubMed: 22231119

Joachimczak, M., & Wróbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: Scaling the tricolour problem to the third dimension. In G. Kampis, I. Karsai, & E. Szathmáry (Eds.), *Advances in Artificial Life* (pp. 35–43). Springer. https://doi.org/10.1007/978-3-642-21283-3_5

Karlebach, G., & Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, *9*(10), 770–780. https://doi.org/10.1038/nrm2503, PubMed: 18797474

King, M. C., & Wilson, A. C. (1975). Evolution at two levels in humans and chimpanzees: Their macromolecules are so alike that regulatory mutations may account for their biological differences. *Science*, *188*(4184), 107–116. https://doi.org/10.1126/science.1090005, PubMed: 1090005

Langton, C. G. (1986). Studying Artificial Life with cellular automata. *Physica D: Nonlinear Phenomena*, *22*(1–3), 120–149. https://doi.org/10.1016/0167-2789(86)90237-X

Langton, C. (2019). *Artificial Life: Proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems*. Routledge. (Original work published 1989)

Levine, M., & Davidson, E. H. (2005). Gene regulatory networks for development. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(14), 4936–4942. https://doi.org/10.1073/pnas.0408031102, PubMed: 15788537

Miralavy, I., & Banzhaf, W. (2023). *Analyzing the impact of space in evolving spatially represented genetic programming models*. Unpublished manuscript.

Mohamadou, Y., Halidou, A., & Kapen, P. T. (2020). A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. *Applied Intelligence*, *50*(11), 3913–3925. https://doi.org/10.1007/s10489-020-01770-9, PubMed: 34764546

Muggianu, F., Benso, A., Bardini, R., Hu, E., Politano, G., & Di Carlo, S. (2018). Modeling biological complexity using biology system description language (BISDL). In *2018 IEEE international conference on Bioinformatics and Biomedicine (BIBM)* (pp. 713–717). IEEE. https://doi.org/10.1109/BIBM.2018.8621533

Nehaniv, C. L., & Wagner, G. P. (2000). The right stuff: Appropriate mathematics for evolutionary and developmental biology. *Artificial Life*, *6*(1), 1–2. https://doi.org/10.1162/106454600568285

Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., & Bejerano, G. (2013). Enhancers: Five essential questions. *Nature Reviews Genetics*, *14*(4), 288–295. https://doi.org/10.1038/nrg3458, PubMed: 23503198

Sanchez, S., & Cussat-Blanc, S. (2014). Gene regulated car driving: Using a gene regulatory network to drive a virtual car. *Genetic Programming and Evolvable Machines*, *15*(4), 477–511. https://doi.org/10.1007/s10710-014-9228-y

Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, *18*(2), 261–274. https://doi.org/10.1093/bioinformatics/18.2.261, PubMed: 11847074

Theraulaz, G., Bonabeau, E., Nicolis, S. C., Solé, R. V., Fourcassié, V., Blanco, S., Fournier, R., Joly, J. L., Fernández, P., Grimal, A., Dalle, P., & Deneubourg, J.-L. (2002). Spatial patterns in ant colonies. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(15), 9645–9649. https://doi.org/10.1073/pnas.152302199, PubMed: 12114538

Toffoli, T. (1984). Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D: Nonlinear Phenomena*, *10*(1–2), 117–127. https://doi.org/10.1016/0167-2789(84)90254-9

Tucker, L. W., & Robertson, G. G. (1988). Architecture and applications of the connection machine. *IEEE Computer*, *21*(8), 26–38. https://doi.org/10.1109/2.74

Villaverde, A. F., Cosentino, C., Gábor, A., & Szederkényi, G. (2019). Computational methods for identification and modelling of complex biological systems. *Complexity*, *2019*, Article 4951650. https://doi.org/10.1155/2019/4951650

von Neumann, J. (1966). *Theory of self-reproducing automata*. (A. W. Burks, Ed.). University of Illinois Press.

Wainer, G. A. (2017). *Discrete-event modeling and simulation: A practitioner's approach*. CRC Press. https://doi.org/10.1201/9781420053371

Watson, J. D., & Berry, A. (2009). *DNA: The secret of life*. Knopf.

Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, *10*(2), 122–133. https://doi.org/10.1038/nrg2509, PubMed: 19139763

Wróbel, B., Abdelmotaleb, A., & Joachimczak, M. (2012). Evolving spiking neural networks in the GReaNs (gene regulatory evolving artificial networks) platform. In *EvoNet2012: Evolving Networks, from Systems/Synthetic Biology to Computational Neuroscience Workshop at Artificial Life XIII* (pp. 19–22).

Wróbel, B., & Joachimczak, M. (2014). Using the genetic regulatory evolving artificial networks (GReaNs) platform for signal processing, animat control, and artificial multicellular development. In T. Kowaliw, N. Bredeche, & R. Doursat (Eds.), *Growing adaptive machines* (pp. 187–200). Springer. https://doi.org/10.1007/978-3-642-55337-0_6