SIMULATION AND VISUALIZATION OF GENETIC REGULATORY NETWORKS

C.A.H. BAKER, M.S.T. CARPENDALE
Department of Computer Science, University of Calgary, Calgary, Alberta, Canada

M.G. SURETTE
Department of Microbiology and Infectious Diseases, University of Calgary, Calgary, Alberta, Canada

Gene regulation networks are a significant biological research area. Simulations and visualizations of genetic processes are being created as biologists grapple with the vast amounts of new genetic information. We present a genetic network simulation environment that visualizes protein-gene interactions and concentrations as they occur during the simulation. In addition, the layout of this genetic simulation can be changed into a visualization of a conceptual model of the simulated genetic network. The visual simulation and network visualization are integrated by animating the change between the simulation view and the visualization view. The flagella system of Escherichia coli has been used to verify the results of this tool and to provide a working model. This particle-based real-time visual simulations of genetic networks allows for virtual experimentation using similar methodologies of live experiments.

1 Introduction

Currently genetic research is generating an abundance of gene expression data and developing intricate models of genetic networks. As the study of these gene regulatory networks proceeds, further tools are needed to examine their dynamics. To facilitate this research, a visual simulation enivronment has been designed to provide biologists with a virtual tool, supporting the observation of the movement and general concentration of constituent elements of a genetic network. Any region currently being viewed displays a real-time simulation of the proteins and genes present. In order to visually clarify the structure of the genetic network currently being simulated a corresponding visualization of the conceptual genetic network is provided. The visualization of the simulation and the conceptual model of the network structure are integrated to increase the potential for understanding the complexities of genetic networks.

The results obtained from both the simulation and visualizations are verified against biological data on Escherichia coli K12's flagella system. This tool has been developed to provide a type of virtual laboratory where experiments can be run in a similar methodology to live experiment.

2 Related Work

The simulation and visualization of genetic networks is an active research area. McAdams has identified many important areas of genetic networks and their importance to simulation [1]. Furthermore, modular design has been observed within biology; suggesting the practical use of the conceptual models of genetic networks [2]. Previous work has focused in three main areas: the simulation, visualization and identification of genetic networks. Simulation of such networks has been achieved through the use of differential equations [3], logic [4], and relaxed (stochastic) rule sets [5,6]. While the simulations are not visual, the results of these simulations are displayed visually. The structures of the circuits are presented but no real-time simulation information is shown. Other research has focused on the strict visualization of conceptual genetic networks. These works use large gene-protein databases [7] as input to create visual models of this information. GeneNet pulls from an object-oriented database to visualize a graph layout the represents the interactions [8]. Michaels displays graphs of concentration levels [9]. Random Boolean Networks have also been developed for this use [10]. The identification of genetic networks [11,12,13,14] uses these same concentration graphs to display the identified networks.

3 Simulating and Visualizing Genetic Regulation

To simulate a genetic network requires a conceptualization of the different constituent parts that comprise such networks. Genetic networks consist of a set of genes that are related through a set of regulatory proteins. Each gene requires some input and produces some output. The gene's output (expression) results in the production of constructive or regulatory proteins. A constructive protein comprises the structural makeup of the organism. A gene receives input through binding of regulatory protein(s) to one or more operator sites (DNA segments). This binding stimulates or inhibits the gene's expression. Each regulatory protein binds to specific operator(s) in a DNA sequence dependent manner based on biochemical laws of interaction. Thus only specific proteins are able to bind to certain genes. Variations in binding affinity are based on DNA sequence present within operator site. This conceptual rule set can be exploited to develop simulation environments.

Figure 1 is a diagram of the visual representation of a gene. A gene is visualized as two concentric spheres (Figure 1(a)). The outer circle holds the genes operator sites. Figure 1(b)) shows a regulatory protein bound to an operator site. When the appropriate promoter is bound the gene starts to express its own regulatory proteins. The regulatory proteins being expressed

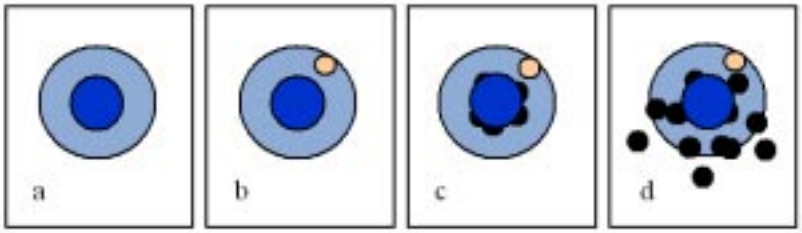

Figure 1: (a) an inactive gene, (b) a gene with a bound regulatory protein, (c) a gene beginning to express proteins (d) a gene continuing to express proteins

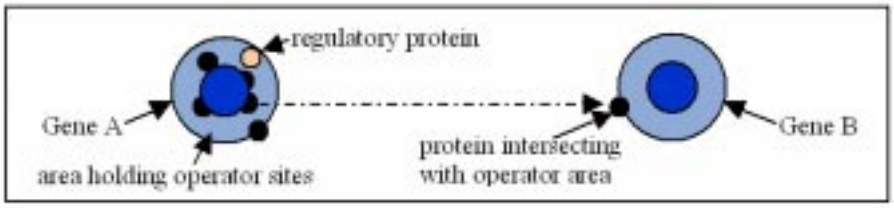

Figure 2: There is a single protein bound to the topside of the gene, which promotes its expression. The proteins are being expressed by the gene and are distributing through the environment

are visualized as appearing from under the inner circle (Figure 1)(c) and (d)).

3.1 The Simulation Environment

Large rule sets define genetic networks, which dictate the resulting course of execution. These networks can be conceptually thought of as two or more genes interrelated through expression. Even with small genetic networks, an emergent pattern in execution or expression becomes apparent. This execution is dynamic, however the overall result of the network remains the same. In biology, protein concentrations create probabilistic environments. The higher the concentration of a protein, the greater the likelihood it will come in contact with a gene which requires it. The lower the concentration the less likelihood of a protein binding. Random protein movement is used to simulate such probabilistic distributions of protein over time.

3.2 The Gene Interaction

Within the simulation, genes require particular proteins to be activated. Each gene has its own rule set, which defines when a gene is activated. During

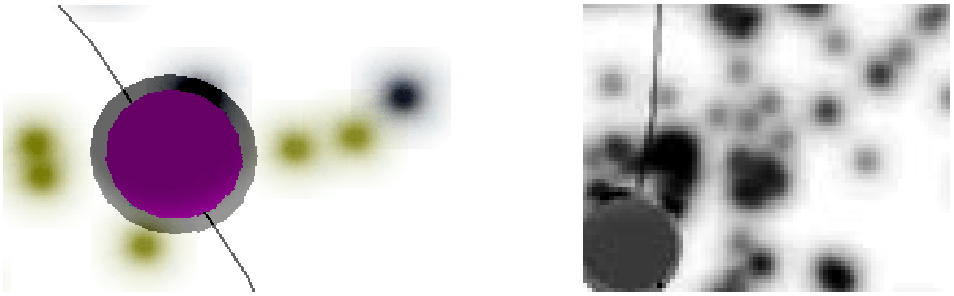

Figure 3: Left: Single gene bound and expressing, Right: Concentration formed from closely packed proteins

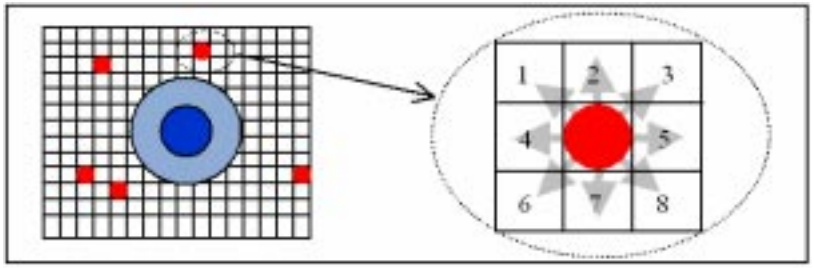

Figure 4: Sample section of environmental grid with random moving proteins and a single gene

simulation the gene's rule set is defined as follows:

If the gene's operator site is vacant then
The gene proceeds with its normal activity.
If the gene's operator site is bound with the right promoter protein then
The gene will express.
If the gene's operator site is bound with an inhibitor protein then
The gene will not express.

In order for a regulatory protein to have a chance to bind to the gene it must intersect the any point on the outer sphere (see Figure 2). This allows a greater chance for operator site binding instead of only a single intersection point. As part of the rule set each gene has an affinity, which is usually expressed as a percentage. This is the percentage chance of the protein binding to the gene during an intersection. The rate by which a gene expresses is controlled by a concentration function. For this simulation the concentration function is obtained from experimental data.

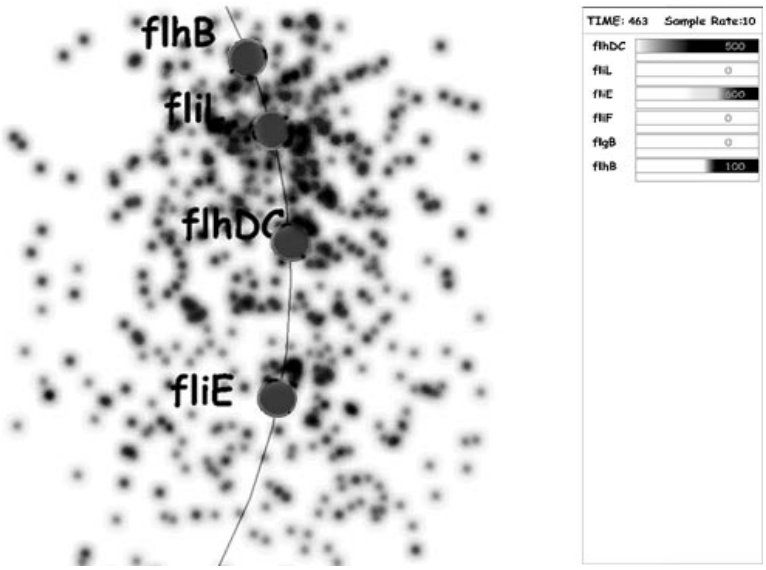

Figure 5: Visualization of genetic network simulation

Multiple operator sites have also been included, allowing genes to bind with one or more proteins. Each operator site has its own rule set including affinity, required promoter protein and required inhibitor protein. Most activator/repressor occurrences can be described by a logical *AND*. Other Boolean operators exist, but currently only *AND* has been implemented. This result decides whether or not the gene expresses.

3.3 Proteins

Regulatory proteins create the interconnections between genes and this subsequently forms the genetic network. It is these proteins that need to move in a manner to correctly simulate a probabilistic environment. This is accomplished by having each protein randomly move within the spatial grid. The grid imposes a directional restriction on the proteins to eight directions (Figure 4). This grid is also used to identify the protein-gene intersections, eliminating computationally expensive intersection tests between genes and proteins (Figure 3). Before the simulation starts, each grid-cell covered by a gene is assigned the gene's ID (Figure 4). This allows intersection testing by only one comparison between the grid value and the location of a protein.

At every point in time during the simulation, each protein is moved randomly in one of the eight directions. Initially, the proteins are emitted from the centre of the gene and are free to roam randomly throughout the spatial grid. Through time the proteins distribute themselves throughout the system. Each protein is represented using a diffused circle, which can blend with other overlapping proteins (Figure 3). When multiple types of protein are present, blending still occurs and gives multiple coloured sections where certain protein concentrations are stronger than others (Figure 5).

All gene parameters can be altered with the use of a dialog box. Any changes made to these parameters result immediately within the dynamic simulation. The genes name, base pair position, produced protein, decay, protein colour, and concentration function can be changed within the top half of the dialog box. The bottom half allows alterations to each operator site, which contains affinity, required promoter protein, and required inhibitor protein. Once Apply is pressed, the simulation is updated with these changes.

3.4 Chromosome

All genes are positioned within a circle representing the gene set's chromosome. The actual physical geometry of a chromosome is not essential to the simulation because proteins rapidly diffuse after transcription. Therefore a circle was choosen to ensures that each protein's travelling distance is the same to each side of the chromosome. This is accomplished by positioning the circle equidistant within the spatial grid with the edges of the grid wrapping around. The position of each gene on the circle is calculated using its real base pair position. A base pair range for the gene set is originally provided and each gene is positioned around the circle within that base pair range.

3.5 Gene Expression Analysis

In recent years cluster analysis has become an invaluable asset to biologists in studying gene networks. It allows the simultaneous analysis of multiple gene expressions. Furthermore, similar gene expressions can be clustered together giving a temporal ordering of expression. The information about how the gene is expressing is visualized with the use of colour. Often using one colour to indicate no expression and a second colour to indicate full expression. The transitional states are shown as a gradient between these two colours. In the simulation environment parameters can be changed and virtual experiments rerun to identify differences and similarities in expression. In our simulation, each gene has it own real-time expression analysis, which is visualized using the two-colour scheme just described.

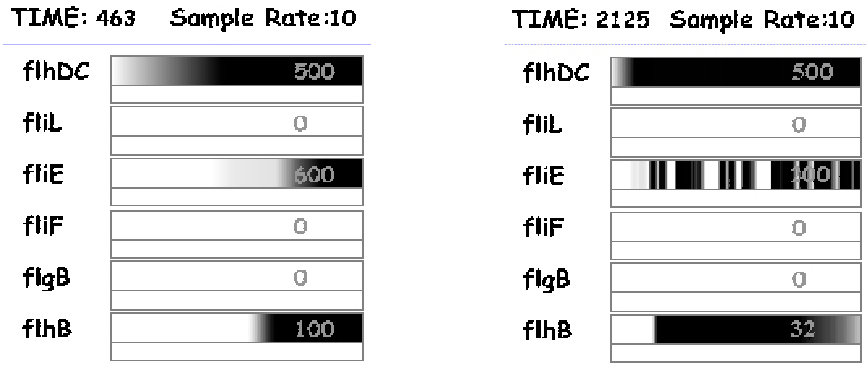

Figure 6: Gene expression analysis: These images show two different simulation runs

The top rectangle depicts the network dynamics of the simulation. The bottom rectangle shows actual cluster analysis results from experiments, which can be inputted through a file. This allows for the direct comparison between the experimental and the simulated. Each one has a time scale from zero to the current time, and updated continuously as the network progresses. Each bar's colour is determined on a scale from no expression (white) to full expression (black). The number in the right side of the bar shows that specific gene's current number of active proteins (the number of expressed minus the number of decayed). All expression bars are aligned to the right side of the screen for comparison between individual genes (Figure 6).

It is also possible to output the simulations expression analysis results to a file for numerical comparison to the actual cluster analysis. This allows for smaller variations to be seen, however, the resolution of this comparison is limited since results from the simulation and biology are never exactly the same.

3.6 Discussion

This simulation uses grid-based random protein movement in a wrap-around torodial environment. In this environment genes are placed in a circle according to their position in the chromosome. These three factors together help minimize the effect of the model's spatial layout on the probabilistic development of the genetic network. However, while these factors work well for modelling the behaviour of the genetic regulation network, this simulation is

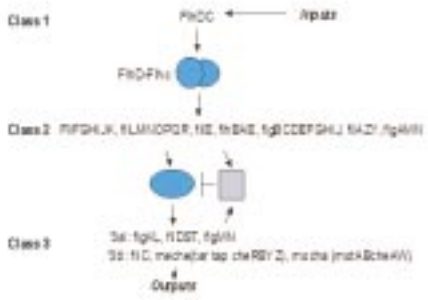

Figure 7: Gene network hierarchy of the flagella operons in E. coli[15]

visually complex and specific structures of genetic networks are hard to discern from simply watching the simulation. Therefore, we have also created a visualization of the resulting network and to make the connection between the simulation and the network layout explicit, we must animate the transition from the visual simulation to the network layout.

4 Structuring the Layout of the Genetic Network

The simulation structures the spatial layout of the genes through the use of their position in the chromosome. In this section we explore using the interactions between the genes to reveal a network organization that results in a different spatial organization.

4.1 Determining Layout Structure

Consider the genetic regulation network that creates the flagella of Escherichia coli. The protein-gene interactions within this network have been identified[15]. One method of visualizing these interactions is shown in Figure 7.

Here spatial organization is based on temporal ordering of gene expression. Each row holds the genes for a temporal class. The topmost gene is the first to express. The genes in the second row require a regulatory protein from a gene in the previous row. This organization can be calculated recursively by analyzing protein-gene interaction for the entire network. This organization can be calculated recursively by analyzing protein-gene interaction for the entire network. An algorithm has been developed that can discover this organizational structure from the simulation (Figure 7). The algorithm checks every

gene for its' earliest expression time point and places that gene within the correct temporal class. This results in a graph structure for the network. This structure can be displayed using a graph layout in which the nodes represent the genes, the rows represent the temporal classes, and the lines represent the fact that a regulatory protein from one gene is used to control the expression of the other gene.

Unfortunately, the directed graph previously described does not completely represent the interactions of genetic networks. Genetic networks usually also contain feedback loops. The presence of these feedback loops often interferes with the ease of displaying genetic networks with basic two dimensional graph layouts, since these feedback loops frequently make the network non-planar. However, feedback loops are prominent and important mechanisms in gene regulation. These loops provide for self-regulation within networks. This self-regulation can exist between one or more genes. When only one gene is present, the gene produces the very protein that regulates itself. When multiple genes are involved, the self-regulation controls multiple productions with the network. There are also positive and negative feedbacks. A positive feedback promotes the continued expression of the network's architecture while a negative feedback inhibits the expression of that network. A negative feedback enables the gene network to terminate its own operation once a particular point of execution is reached. Feedbacks are intrinsically difficult to model and visualize with clarity. Directional graphs can very quickly become hard to read when they include multiple edge-crossings due to the inclusion of the connections that represent feedbacks[16].

To alleviate this problem, our approach brings the graph layout into three dimensions. Moving to three dimensions allows us to use the extra dimension to visualize feedback loops without creating edge-crossings. Each temporal class is taken from a 2D line to a 3D ring. Each gene within that temporal class is placed evenly around the ring. The rings are indicated by dashed lines. This keeps them visually distinct from the network connections. Forward or promoting protein regulation connections are visualized with curved lines that proceed from the previous the ring to connect regulated genes (Figure 8).

Since feedbacks can cause crossings to occur, they are visualized using straight lines. These lines travel through the centre of the rings, visually separating forward regulation from feedbacks and alleviating the edge-crossings caused by the feedback connections. To assist in viewing the rings they can be rotated to view from any angle, giving the user a better 'feel' for the network architecture.

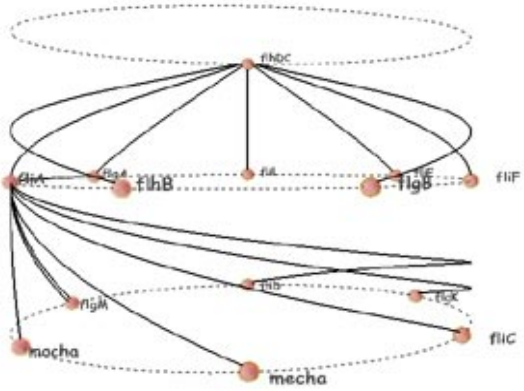

Figure 8: Conceptual genetic network ring visualization

4.2 Morphing

With the separation of the simulation and the genetic network visualization there needs to be some method of cognitively connecting the two. Both visualizations are dramatically different in appearance, however, both contain the same information shown in different ways. Morphing is used to animate the connection between these two visualizations. The morphing can be done back and forth between both visualizations and can be interrupted at any point during the movement. This is accomplished by translating the genes' current position to its new position. The animation allows for the user to cognitively ascertain the relation between the two visualizations.

5 Results and Validation

The validation of the results of our simulation and visualization have been built into the system. This was done so direct comparisons can be done to verify the results against empirical results from biological experiments. The simulation is verified through the use of a gene expression analysis. At any point the results of the simulation can be compared with the real results shown directly below. Furthermore, text output of the simulation results is possible and can be compared numerically with the actual biological results present within the system. To verify the conceptual visualization of genetic networks, a comparison of the biologically formulated networks with the system produced visualization

can be done. This comparison is only valid for well-known networks where the network architecture is certain. More skeptical networks do not provide any form of validation towards the visualization since they could be incorrect. The validity of any such experiment depends on the correctness of the data one is comparing too. Subsequently, in both the simulation and visualization the highly studied E. coli K12 has been used to validate our results. Cluster Analysis has been obtained for the flagella system of E. coli and was used as a comparison to the dynamic cluster analysis of the simulation. Furthermore, the well-known structure of this network has been identified which provides a comparison for the gene network visualization.

6 Conclusions

As biology continues to explore the molecular level, conceptual models become a necessity to understand these complex interactions. Visualization of such interactions may not only assist understanding, but also help biologists to further enhance their conceptual models. By constructing a visualization and simulation of genetic networks in a generic way may provide a possible framework for other genetic networks under research. The flagella system has been used to test and verify the network, revealing similar self-regulation cycles being present in the simulation as observed in biology. The simulation has been created as close to biological systems as possible, while the network visualization depicts the networks structure. Morphing has been used to cognitively integrate the two visualizations together to further understanding. This tool demonstrates visually both the protein-gene interaction as well as the underlying network structure to provide biologists with a new tool to enhance their knowledge of this topic area.

References

1. Harley H. McAdams and Adam Arkin. Simulation of prokaryotic genetic circuits. In *Annu. Rev. Biophys. Struct.*, volume 27, pages 199–224, 1998.
2. Leland H. Hartwell, John J. Hofield, Stanislas Leibler, and Andrew W. Murray. From molecular to modular cell biology. *Nature*, December 1999.
3. Harley H. McAdams and Lucy Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, August 1995.
4. V.N. Serov M.G. Samsonova. Network: An interactive interface to the tools for analysis of genetic networks structure and dynamics. *Pacific*

Symposium on Biocomputing, 4:102–111, 1999.

5. Karsten R. Heiftke and Steffen Schulze-Kremer. Biosim - a new qualitative simulation environment for molecular biology. *6th International Conference on Intelligent Systems for Molecular Biology*, pages 85–94, 1998.
6. Hiroshi Matsuno, Atsushi Doi, Masao Nagasaki, and Satoru Miyano. Hygrid petri net representation of gene regulatory networks. *Pacific Symposium on Biocomputing*, pages 3338–349, 2000.
7. F.A. Kolpakov and V.N. Babenko. Mgl: An object-oriented computer system for molecular genetic data management, analysis, and visualization. *Bioinformatics*, 14(6):56, 1998.
8. Fedor A. Kolpakov, Elena A. Ananko, Grifory B. Kolesov, and Nikolay A. Kolchanov. Genenet: a gene network database and its automated visualization. *Bioinformatics*, 14(6):529–537, 1998.
9. George S. Michaels, Daniels B. Carr, Manor Askenazi, Stefanie Fuhrman, Xiling Wen, and Roland Somogyi. Cluster analysis and data visualization of large-scale gene expression data. *Pacific Symposium on Biocomputing*, 3:42–53, 1998.
10. A. Wuensche. Genomic regulation modeled as a network with basins of attraction. *http://www.santafe.org/ wuensche/ddlab/*.
11. D. Thieffry and R. Thomas. Qualitative analysis of gene networks. *Pacific Symposium on Biocomputing*, 3:77–88, 1998.
12. Denis Thieffry, Araceli M. Huerta, Ernesto Perez-Rueda, and Julio Collado-Vides. From specific gene regulation to genomic networks: a global anlysis of transcriptional regulation in escherichia coli. *BioEssays*, 20:433–440, 1998.
13. Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, and Satoru Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. *Ninth Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 695–702, Janurary 1998.
14. Shin Ando and Hitoshi Iba. Identifying the gene regulatory network by real-coded, variable-length, and multiple-stage ga.
15. S. Kalir, J McClure, K. Pabbaraju, C. Southward, M. Ronen, S. Leibler, M.G. Surette, and U. Alon. Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria. *Science*, 292:2080–2083, June 2001.
16. Helen C. Purchase, Robert F. Cohen, and Murray James. Validating graph drawing aesthetics. In *Symposium on Graph Drawing, GD'95*, pages 435–446, 1995.