THE UNIVERSITY OF CALGARY

GeneVis:

Simulating and Visualizing Genetic Regulatory Networks

by

Charles Andrew Hugh Baker

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

June, 2002

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "GeneVis: Simulating and Visualizing Genetic Regulatory Networks" submitted by Charles Andrew Hugh Baker in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.

Dr. Sheelagh Carpendale
Department of Computer Science
University of Calgary

Dr. Michael G. Surette
Department of Microbiology and Infectious Diseases
University of Calgary

Dr. Przemyslaw Prusinkiewicz
Department of Computer Science
University of Calgary

Dr. John Dill
School of Engineering Science
Simon Fraser University

Date

Abstract

This thesis presents GeneVis, a program that simulates and visualizes a conceptual model of genetic network interaction. In this model, genetic networks are considered to be sets of genes that are regulated by sets of proteins. When genes in the network express, they trigger the production of proteins, which in turn can regulate the expression of other genes, thus creating a network of dependence. GeneVis simulates genetic networks and visualizes the process of this simulation interactively, providing a visual environment for exploring the dynamics of genetic regulatory networks. The visualization environment supports several representational modes, which include: a protein interaction representation, a protein concentration representation, and a network structure representation. The protein interaction representation shows the activities of the individual proteins. The protein concentration representation illustrates the relative spread and concentrations of the different proteins in the simulation. The network structure representation depicts the genetic network dependencies that are present in the simulation. GeneVis includes several interactive viewing tools. These include animated transitions from the protein interaction representation to the protein concentration representation, and from the protein interaction representation to the network structure representation. There are also three types of lenses: fuzzy lenses, base pair lenses and the network structure ring lens. With a fuzzy lens an alternate representation can be viewed in a selected region. The base pair lenses allow users to reposition genes for better viewing or to minimize interference during the simulation. The ring lens provides detail-in-context viewing of individual levels in the genetic network structure representation.

Acknowledgements

I would like to take this chance to say to some small extent how much I appreciate all the help I have received during the last few years.

To my supervisor, Dr. Sheelagh Carpendale, thank you for our many long discussions and giving me the freedom to explore my ideas. Thank you for the many hours of editing of this thesis and our papers.

To my collaborator, Dr. Michael Surette, thank you for helping at every stage of the development of GeneVis, and the assistance in all areas of genetics.

To the professors who inspired new and different thoughts throughout this Masters program, Dr. Przemyslaw Prusinkiewicz, Dr. Christian Jacob, and Dr. Kees van Overveld.

To Kari Lyn Basaraba, thank you for being a true friend and for your numerous revisions to this thesis.

To Gemma Lindsay Collins, Christopher Paul Marriott and Stacey Scott, thank you for editing this thesis and providing invaluable feedback and comments.

I would also like to thank all the students who have helped me along the way, Kaye Mason, Peter MacMurchy, Mark Matthews, Brendan Lane, Vincenzo Marra, and Brenda Lynn Eshpeter.

And finally, to my parents, Chuck Baker and Faye Baker, I could not have done this without you. I am incredibly fortunate to have two wonderful parents and thank you for your love and undying belief in me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since the mapping of the human genome, research interests in biology have shifted towards the issue of discovering what the genetic code actually does. This includes such questions as: For what proteins do *genes* code? How does this affect the development and functioning of the organism? How do genes communicate appropriate information to each other? How do *genetic networks* function? And what are their dynamics?

With the advent of new technology, such as DNA micro-arrays [19] it is now possible for biologists to measure, in parallel, the activity levels of genes as a function of time. Biologists may use these temporal measurements to infer which genes interact with each other and to determine the patterns of these interactions. However, this is a non-trivial exercise. The data is expensive and difficult to obtain, and can contain errors. Furthermore, even relatively small genetic networks may have complex dynamics due to positive and negative *feedback loops*. To assist in the process of inference, models of the observed genetic activity are being developed. These models can be used to create simulations and visualizations, helping us form mental constructs of the behaviour of genetic networks and thus further our understanding.

These models are becoming more complex and more sophisticated which makes their analysis more difficult. Consequently, new research areas such as Computational Biology and Biological Visualization have emerged. Computational Biology refers to the use of computers to study biological processes. Biological Visualization

refers to the use of computer graphics to develop visual representations of biological data. Research in these fields is contributing to the development of software that can be used in conjunction with laboratory technology to provide: effective data storage and access, *computational models* for studying the behaviour of this data, and *visual representations* to aid in the interpretation and comprehension of this data.

GeneVis, as described in this thesis, is research directed at exploring the possibility of creating simulation and visualization software tools that aid comprehension of genetic network processes. The focus is on creating a simulation based on a conceptual model of *genetic regulation* and on developing dynamic visualizations of the simulation as it progresses. In GeneVis, spatial organization of the simulated entities is used and adjusted interactively in order to help illustrate and support the exploration of mental concepts. Moreover, the intention is that the integration of different visualization techniques may assist in understanding different aspects of the same data set.

The next section provides an overview of the biological terms and concepts that are used in this thesis. Section 1.2 states the research objectives of this thesis followed by the research methodologies for achieving the stated objectives. Section 1.4 presents the conceptual model on which GeneVis is based and then this chapter concludes with an explanation of the organization of this thesis.

1.1 Biological Background

For this thesis the specific area of interest is genetic regulatory networks. The following is a brief and simplified explanation of background information [15, 16] about

the interactions present within genetic networks. All words in italics can also be found defined in Appendix A.

Genes produce proteins and the rate of protein production of a gene can be regulated by other proteins. Genetic networks can be thought of as sets of genes that are regulated by sets of *proteins*. These networks exist in both *prokaryotes* and *eukaryotes*. All cellular organisms are either prokaryotes or eukaryotes. Prokaryotes are organisms that have no nuclear membrane. Eukaryotes are organisms that have a nucleus and multiple linear chromosomes within a single cell. This thesis is limited to the domain of prokaryotes because their genetic interactions are less complex than eukaryotes.

Within each prokaryotic cell there is deoxyribonucleic acid (DNA). DNA consists of a strand containing a sequence of four possible nucleic acids: guanine (G), cytosine (C), thymine (T) and adenine (A). This DNA strand defines the genetic code, which creates and maintains the organism. DNA is subdivided into subsequences of nucleic acids called chromosomes. Each chromosome is divided into genes. A gene is a functional subsequence of the nucleic acid strand, which can produce proteins. Each gene can have an *operator site* region as well as coding regions. The operator site region can contain a number of operator sites; each site can promote or inhibit the production of its gene's coded protein. The coding region is the subsequence of the gene that codes for the protein that the gene can produce.

There are several processes involved in a gene's production of a protein. Transcription is the first process that is required for a gene to be able to create its coded protein. Transcription begins with RNA polymerase binding to the coding region in the gene and uncoiling the double helix so it can be copied to form a copy strand

called ribonucleic acid (RNA). The RNA is then further processed through translation where nucleic acids are paired with amino acids (additional copy strands can also occur). This forms an amino acid strand with a peptide backbone. The sequence of amino acids then coils into a minimum energy three-dimensional structure. This is what is referred to as a protein. Through the process of transcription, a gene's coding region can be translated into a protein. When a gene's activities result in the creation of a protein this is referred to as *expression*.

The operator sites can increase or decrease a gene's production of protein. The regulation of a gene through its operator site is accomplished by proteins binding to the operator site. When a protein has been produced by another gene it can move throughout the cell and possibly interact with other genes. A protein's movement within a cell can be approximated as diffusive, however a protein may interact with cellular components and its behaviour may not be truly diffuse [12]. If and when a protein binds to an operator site it can either facilitate or hinder the binding of RNA polymerases. This assistance or hindrance changes the production or expression rate of the gene. Each operator site can only be affected by particular proteins that are capable of binding to it. An operator site may have one or more promoter proteins and may have one or more inhibitor proteins. A gene may have one or more operator sites. If no proteins are bound to any of the gene's operator sites, polymerase can still bind independently and transcribe new proteins at a base rate. This base rate is called the *basal activity* of the gene. Any of the proteins that are capable of binding to any of the gene's operator sites are referred to as that *gene's required proteins*. When speaking in terms of a protein, the genes it can bind to are referred to as *requiring genes* and sometimes specifically as *requiring operator sites*. Proteins also

unbind from an operator sites, this is called *reversible binding*.

Since genes produce proteins and are regulated by proteins, networks of regulation can form. This occurs when the produced protein of one gene regulates the production of another gene, through that gene's operator site. These dependencies can continue for many levels resulting in complex networks or genetic networks.

Since these networks control both the development and function of organisms, they are of crucial importance in investigating the question of how organisms function [10].

1.2 Research Objectives

My intention was to create a visual simulation of genetic regulatory network dynamics. To achieve this involves developing:

- a simulation of genetic regulatory network dynamics,
- an interactive visual representation of these dynamics,
- a visual representation of the network structure,
- visual integration tools that link the network dynamic visualization and the network structure visualization,
- providing manipulation techniques that allow exploration of the visualization, and
- linking the visualization with established data formats.

Previous genetic regulation visualization research focused on the representation of the genetic network structure. This structure has been commonly represented using directed graphs as in GeneGraph [16], WebGen-Net [19], and GeneNet Viewer [8]. There have also been different simulation models created, such as Random Boolean Networks [18], NetWork [17], Circuit Simulation [12], GeneNet Modeller [13], Genetic Network Analyzer [4], StochSim [14], and BioSim [6]. The results from these simulations have been presented as charts, in which the simulated gene activity levels have been plotted as functions of time. While these programs do consider static and dynamic data, their visualizations are not dynamic. They display either a static network structure or a static representation of the simulated dynamics.

1.3 Research Methodology

This problem was investigated using a variation of the information visualization pipeline. Figure 1.1 [8] shows the steps of development in finding a visual approach to the problem statement. As a first step, Dr. Michael G. Surette, from the Department of Microbiology and Infectious Disease at the University of Calgary, was contacted to obtain biological data. Dr. Surette provided data for the flagella system of the *Escherichia coli* (*E. coli*) organism. From this point, development of GeneVis was a process of iterative, participatory design with Dr. Surette and members of his research lab.

The data that was obtained outlined the genetic regulatory network for the *E. coli* flagella system. Two aspects of data were selected to be visualized: the network's behaviour and the network's structure. Emphasis was placed on the network

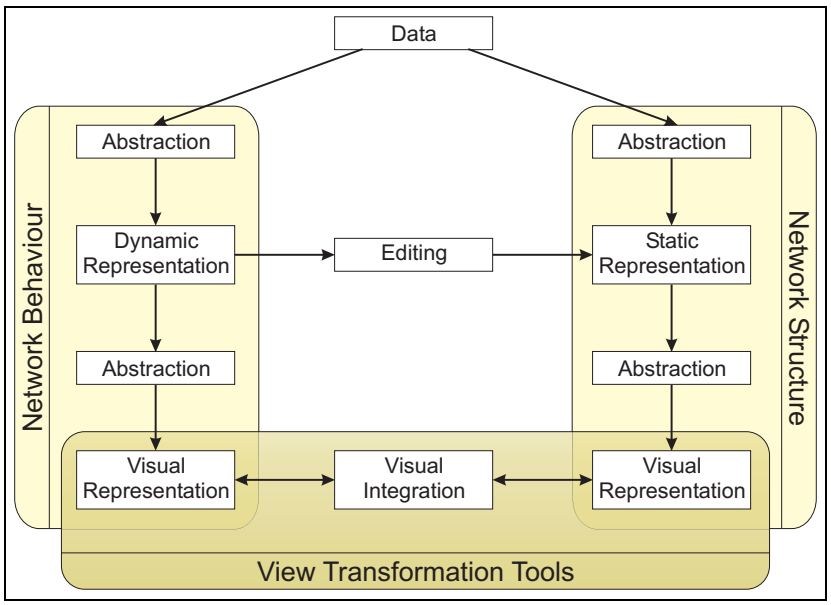

Figure 1.1: Information visualization pipeline: Within this pipeline there are two paths of visualization. One visualizes dynamic data (network behaviour) while the other visualizes static data (network structure). Between each there are transformations tools, which allow the viewing of different aspects of the data.

behaviour, as dynamic visualizations of this process had not been created before.

Figure 1.1 shows: how the dynamic and static visualization are based on the same data; how the editing of the simulation parameters affect both of the visualizations and how the resulting visualizations were integrated through animated transitions.

1.4 GeneVis' Conceptual Model

The conceptual model on which GeneVis is based is a simplified version of the process just described. In GeneVis a genetic network is considered to consist of a set of genes that are related through a collection of regulatory proteins. Each gene may require an input and may produce an output. A gene's output consists of the production of either regulatory or constructive proteins. *Regulatory proteins* act as inputs for the other genes and affect their expression, while *constructive proteins* make up the physical structure of the organism.

A gene receives input through the binding of regulatory protein(s) to one or more of its operator sites. The bound regulatory protein then promotes (or inhibits) the gene's transcription and subsequent expression. Each regulatory protein binds to specific operator site(s). In reality, how this binding occurs and which proteins bind to which operator sites is based on biochemical laws of interaction between molecules [10]. Variations in *binding affinity* are based on the DNA sequences of operator sites. The genes and their characteristics (affinity, operator site(s), proteins expressed, etc.) can be used to create a rule set, on which simulations of the genetic network dynamics are based. In GeneVis, these conditions are declared as rule sets and are based on current understandings of the particular genetic regulatory network.

These rule sets can be reset even during a simulation. According to the rule sets, only specific proteins are able to bind to particular sites on particular genes.

Genetic network behaviour is determined by many hard to predict and dynamic variables. The fluctuating numbers and positions of proteins determine the likelihood that a requiring gene will express. The higher the concentration of a protein, the greater the chance that it will come in contact with a gene that requires it. In addition, proteins decay at different rates, which also affects the cellular dynamics.

1.5 Thesis Organization

This section explains the organization of this thesis. Chapter 1 introduces the problem of viewing and simulating genetic networks, including the research methodology used to approach this problem. Chapter 2 surveys the literature that has previously approached this problem both in the areas of simulation and visualization. In chapter 3 the simulation of genetic network behaviour is discussed. Here, the simulation using GeneVis's genetic network behaviour model is explained with the accompanying visual representation. Presented in Chapter 4 are the visualization tools, first discussing the dynamic visualizations, protein interaction representation, protein concentration representation and view transformation tools for the simulation, and finally describing the static representation of the network structure visualization. Chapter 5 explains how to interact with GeneVis and how these interactions can affect the simulation's dynamics. To conclude, chapter 6 discusses the contributions of this thesis and the future directions of this research.

Chapter 2

Literature Survey

This thesis focuses on the use of computational visualization and simulation as applied to genetic networks. With this in mind, two main bodies of literature are reviewed and presented as the two main sections of this literature survey. The first section is about computational visualization and simulation tools for genetic networks. It is further subdivided into research focusing on the visualization of genetic network structure and research that focuses on simulation of genetic network behaviour. This is followed by a survey of information visualization techniques that are potentially applicable to the visualization of genetic networks.

2.1 Computational Genetic Network Tools

Computational genetic network tools have been applied to the genetic network's structure and the genetic network's behaviour. Only the genetic network structure has been visualized. The genetic network behaviour is simulated and then the results of the simulation are visualized. The genetic network structure tools use static data that provides the information necessary to calculate the network structure model of a genetic network. The genetic network behaviour tools use dynamic data that provides the information needed to create simulations of the network behaviour model.

In previous research, network structures were commonly visualized as directed graphs. Directed graphs show the topological connections between any number of

nodes by identifying the relationships between each element. These directed graphs show the genes as nodes and use the directed edges to indicate the connection between a gene that expresses a particular protein and those genes that require this protein (see Figure 2.1). The tools that focus on visualizing network structure are discussed in Section 2.2.

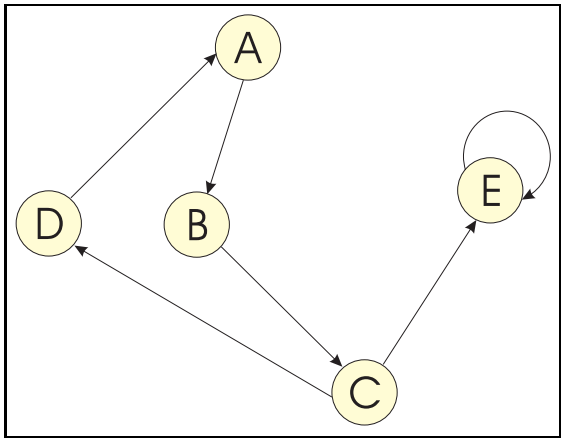

Figure 2.1: Example of a directed graph with 5 nodes connected with arrows indicating the direction of the connection between the nodes.

Section 2.3 describes the research that explores the simulation of network behaviour from dynamic data. Thus far there have been four approaches to simulation: differential equations, Boolean networks, stochastic simulation, and qualitative simulation. Each mimics protein movement in some form to simulate the execution of genetic networks. The simulation results are then typically visualized with line graphs (Figure 2.2) which are plotted depicting concentrations of proteins expressed over time. Figure 2.2 is a line graph of a set of genes showing their expression over time.

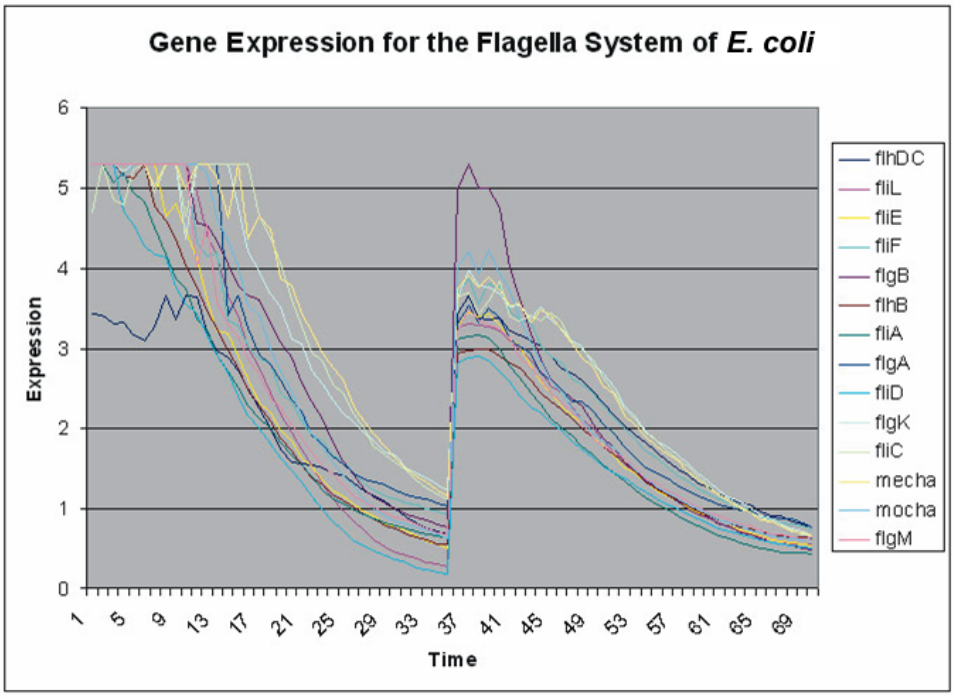

Figure 2.2: Sample line graph of expression analysis results from *E. coli* K12. This data was obtained from Dr. Surette [18].

2.2 Network Structure

This section discusses the research that visualizes genetic network structures. The structure of a genetic network consists of a number of genes related through a number of regulatory proteins [16]. These proteins diffuse through the cell forming connections (regulatory bindings) between genes and causing subsequent gene expression. To identify the network structure, the static data is used to find existing regulatory connections between genes.

2.2.1 GeneGraph

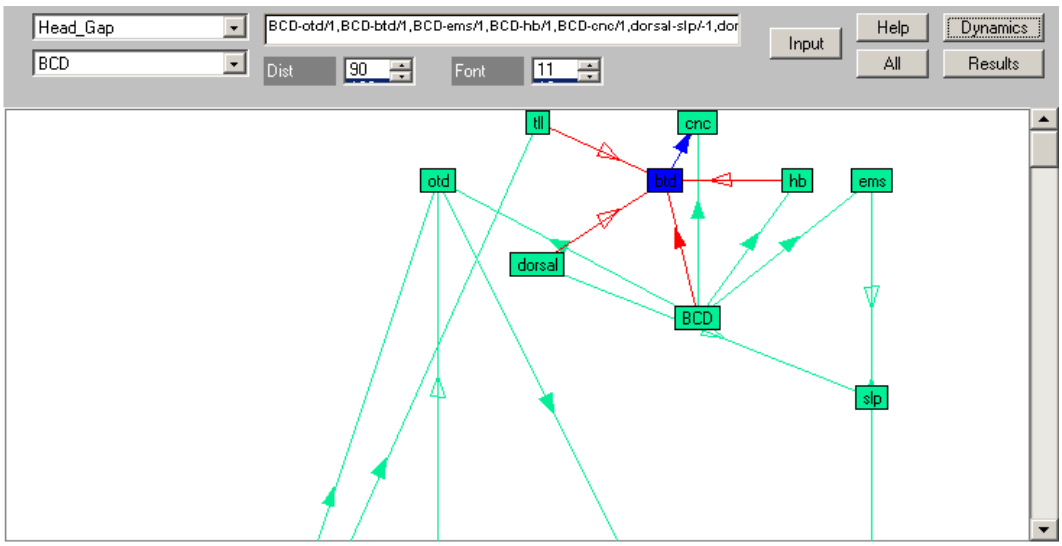

Figure 2.3: GeneGraph: directed graph visualization of a genetic network. The square nodes represent each gene and the connections are displayed as lines with arrows indicating the direction of regulation [41].

GeneGraph [38] is a Java program that is publicly available on the Internet which provides a directed graph visualization of the genetic network structure. The

visualization consists of a square node representing each gene and connections are shown as edges between the nodes (Figure 2.3). Arrows indicate the direction of regulation, meaning that the arrow travels from the producing gene to the gene that requires that produced protein. The nodes are initially positioned according to x, y coordinates on the screen. A set number of filter types are provided to reduce clutter within the graph representation. These filters allow different types of genes to be selectively shown and hidden. This applet provides a directed graph visualization of the network structure and a minimal amount of additional biological information about each element is accessible through the visualization. User interaction allows manipulation of node placement as one method of alleviating edge crossings.

2.2.2 WebGen-Net

WebGen-Net [46] is a Windows program that allows the user to visually construct a model of the genetic network structure in the form of a directed graph. Each constructed genetic network can be saved and loaded. The graph layout uses grid-based positioning with the mouse allowing the user to position a gene on any one of the grid lines. Once a network has been constructed it can be algorithmically laid out in the form of a circle or a fan shape. WebGen-Net provides more information about the genes and proteins than GeneGraph. Genes are visualized as spheres and protein-protein interaction is depicted through the use of different styles of edges to represent different biological mechanisms (coloured, solid or dotted, and thickness) connecting two genes. Each edge connects two genes with an arrow to identify the producing gene and the requiring gene (Figure 2.4 (Top)). The user can interactively edit the directed graph by adding, deleting, and moving edges and genes. A properties dialog

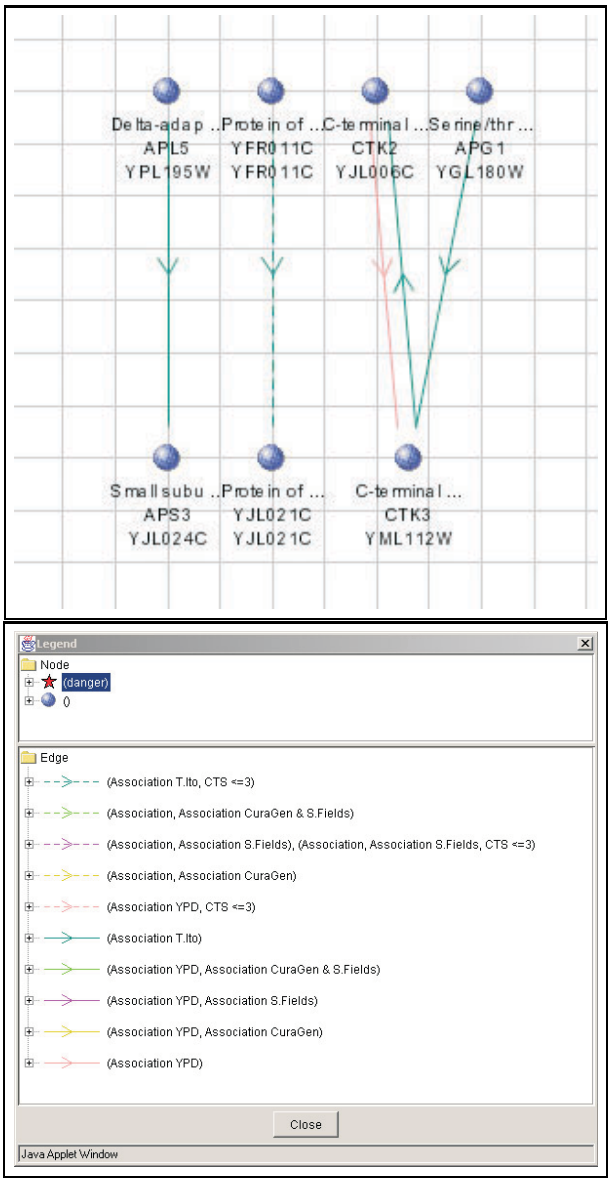

Figure 2.4: WebGen-Net: (Top) Directed graph visualization using spheres to indicate genes and lines with arrows to represent regulatory information, (Bottom) A listing of the regulatory connection information in the form of a tree.[46]

box provides many details about the proteins and genes, which can be examined and changed through a system of menus (Figure 2.4 (Bottom)). Filtering is also provided, allowing selective viewing of different genes within the genetic network display.

2.2.3 GeneNet Viewer

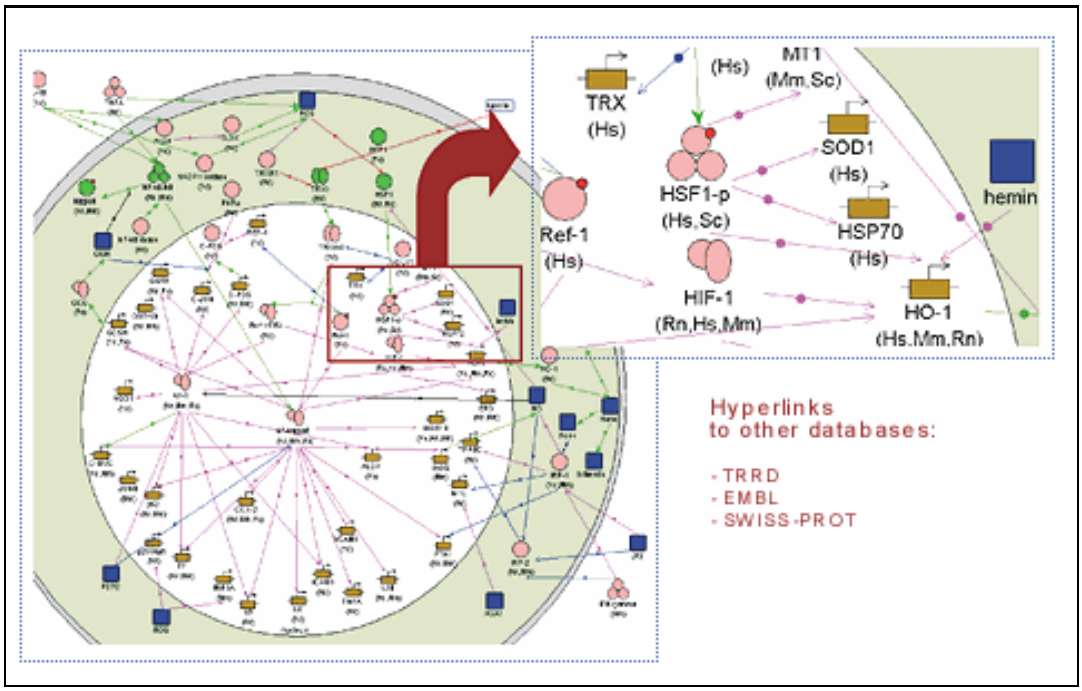

Figure 2.5: GeneNet Viewer: Displays large directed graph visualizations of genetic networks. Genes, proteins, pathways, and groups are represented symbolically. Genetic networks can be encapsulated into circular groups with genes existing inside [20].

GeneNet Viewer [20] is a Java program that visualizes genetic network structure. This program acquires data from the GeneNet database and is hyper-linked within the viewer to the GeneNet database. The GeneNet database is object-oriented, providing information on genes, proteins, and relations that are pertinent to genetic

network structures. Currently, GeneNet supports the viewing of 23 different genetic networks, ranging from simple to complex plants as well as animals. GeneNet refers to gene networks in terms of genes as entities and proteins as relations between entities. The visualization uses symbols to represent genes, proteins, pathways, and classifies groups of genes and proteins (Figure 2.5). Each element within the visualization can be selected to display a list of additional information pertaining to that particular type of element. A zoom feature is included, with scroll bars allowing for navigation of the entire network. Filters also provide layers of information and reduce occlusion among graph elements, by displaying only the elements selected. Furthermore, sub-networks can be encapsulated into categorical groups and are displayed as coloured boxes surrounding the genes within the sub-network.

2.3 Network Behaviour

Network behaviour is the process of the genetic network functioning through gene expression and regulation. In order to simulate execution, protein diffusion must be modelled. While each method presented takes a different approach towards simulating the execution of genetic networks. Simulations are then evolved according to their specific model and the results are visualized using line graphs. This section discusses the approaches that simulate genetic networks and then visualize the simulation results.

2.3.1 Random Boolean Networks

A Random Boolean Network [44] or Random Boolean Cellular Automaton was one of the first approaches used in the simulation of genetic network models. Random Boolean Networks are based on a Boolean rule set with two possible states: on (chemically expressed) or off (chemically repressed). Each gene within this model is dependent on other genes, thus forming a network. The expression of one gene is dependent on the previous outputting genes' values (ON or OFF). These values are used within the Boolean equation to calculate the state of the current gene. Within the network structure connections can form loops. These loops can cause genes to regulate themselves directly or indirectly with the protein they produce. A *direct regulation* means the protein produced by a gene is the regulator for that same gene. *Indirect regulation*, on the other hand, occurs when the protein produced by a gene causes a chain of other genes to be regulated that eventually produces the protein that is needed to regulate the original gene. These regulation loops are called *attractor cycles* [44]. These attractor cycles are the main cycles of regulation and have the effect of controlling the operation of the whole network. Such cycles are central to all other sub-networks, creating a dependency on the attractor cycle for these sub-networks continued expression.

In order for the system to begin simulation, an initial state for each gene is needed. The initial state is set by assigning random values within specified bounds to all elements in the network. These values include random Boolean functions, random number of genes and random initial values. Each function takes other gene outputs as their inputs creating connections between genes. Then the simulator

is turned on and calculates whether or not each gene is to be expressed. This is continued until either attractor cycles are identified or the system is found to be unstable having no attractor cycle. These networks can be viewed and altered in a program called Discrete Dynamics LAB (DDLab) [44], which produces network layouts consisting of nodes and wires (Figure 2.6). *Nodes* represent genes within the network while *wires* represent directional connections between the nodes.

Attractor cycles appear in the center of the screen as a circle. Subtrees or transient trees are upstream genes to the attractor. These subtrees surround the outside of the attractor (Figure 2.6). Genes are represented as circles and regulatory connections are represented as edges between genes. Any gene that does not have an inputting gene for regulation is considered to be in a "garden-of-Eden" state since it is the beginning of the network's process. The transient trees and subtrees show how large numbers of genes are the input for one genes' regulation creating a huge dependence on the genes that are located upstream. This visualization dramatically changes when a perturbation or modification of one gene is made. The program recalculates possible network configurations based on the perturbed gene which results in new basins of attraction. The visualization has no animation when a perturbation is made, and consists only of still images of the networks. There is no interactive editing or manipulation within this program; all input is done through command line prompts.

DDLab creates visualizations which show at a high-level the varying complexities of genetic networks dependent on the number of nodes and connections. This software is not explicitly designed to simulate and visualize specific networks where, in this context, labelling of genes would be essential. Without labelled genes, it is hard

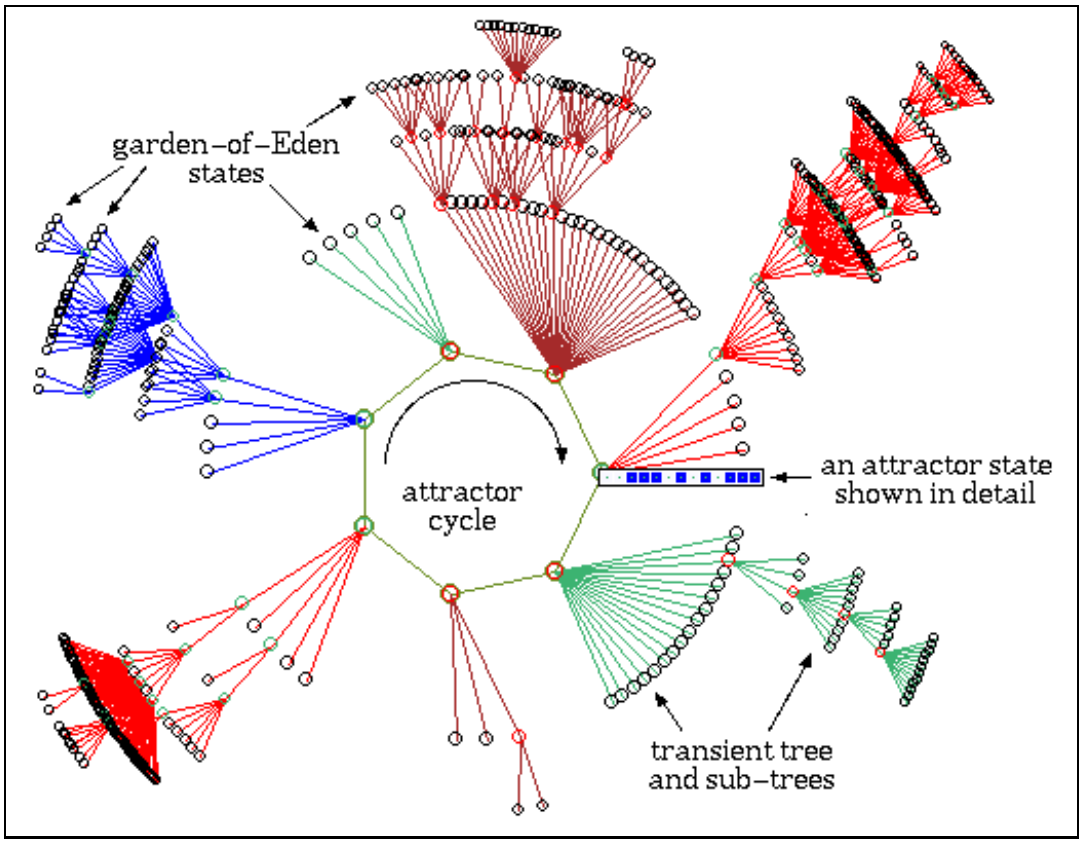

Figure 2.6: Random Boolean Networks: visualization of a genetic network with an emphasis on a central basin of attraction (attractor cycle). Genes are displayed as circles and regulatory connections are represented as lines between genes. The genes that do not have any inputting gene determining their regulation are called "garden-of-Eden" states. The trees of nodes that regulate a single gene are called transient trees and sub-trees. These trees of regulation form a central cycle called an attractor cycle that regulates the entire network [44].

to distinguish changes in network structure when a perturbation is made. Since DDLab is used to view complexity with random networks of particular size and depth, this narrows its application space to a smaller subset of genetic networks.

2.3.2 NetWork

The NetWork[39] Java applet allows users to view and simulate genetic network models. NetWork visualizes static structure data using the same visual representation as its predecessor GeneGraph. It allows very large genetic networks to be simulated and visualized. To perform the simulation the applet uses Boolean logic. As discussed in Section 2.3.1, this results in the simulation finding an attractor cycle. The computational results of the simulation are shown within the network by highlighting genes that have become active due to the network dynamics. The program depicts genes as rectangles and edges are shown as different coloured lines. Edges have four possible types: downstream gene, upstream gene, activation, or repression. An upstream gene is an input for the current gene, while a downstream gene is an output for the current gene. Activation is represented with a filled arrow and repression with a hollow one. This simulation is not interactive and is based on the Boolean simulation model.

The user interface provides a gene search feature, which is used to find genes within large networks that may not be on the screen. When the network is larger than the screen space, pan and scroll techniques are available to allow the user to view the entire network. Alternatively, the network can be displayed in a smaller amount of space by altering the label's font size and the distance between genes. This provides a type of zooming, which varies the amount of network detail. Network

layouts can be interactively edited allowing the user to add or delete genes through a text box. This facilitates both the construction of networks and the mutation of particular elements. Upon mutation, the dynamics can then be retested to reveal any differences in expression.

There is an additional sister program called NetGem. This applet is an expansion of NetWork which uses an annealing algorithm to evenly space out nodes in the network structure layout.

2.3.3 Visual Object Net++

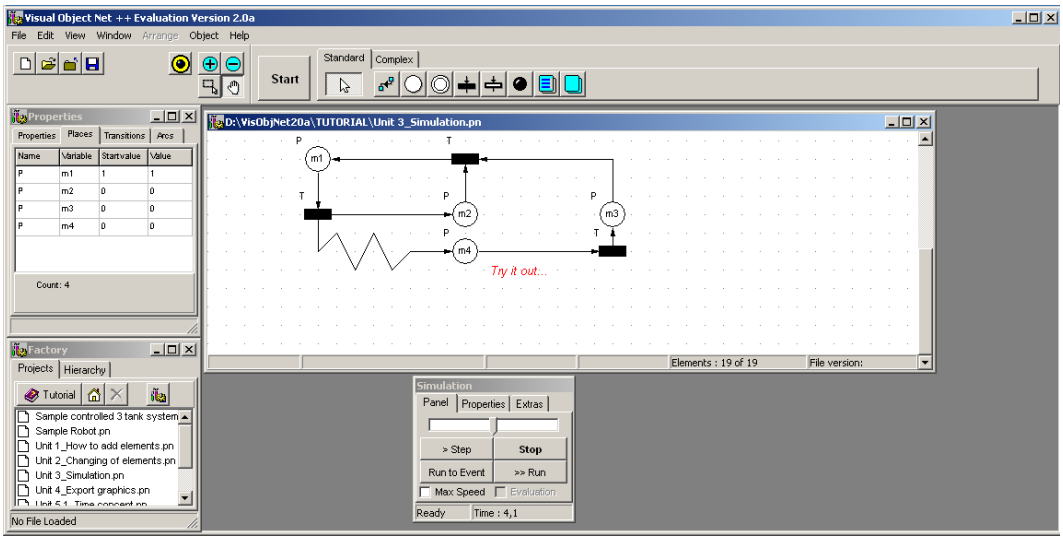

Figure 2.7: Visual Object Net++: directed graph visualization that is based on circuit design methodologies. Here each element is displayed symbolically and the interactions are structured similarly to an electric circuit. A Boolean simulation environment accompanies this visualization which calculates the resulting execution of the network based on the symbols within the circuit [31].

The Visual Object Net++ program allows the construction and simulation of any

type of network. It is not specifically designed for genetic network models, however it can be used with genetic networks. The user can visually construct network components (ie. Genes) and relate them through the use of edges (ie. Regulatory proteins). The layout is very similar to that of circuit design with different symbols representing genes and edges (Figure 2.7). Once construction of the network is complete, a simulation can be run. The simulation is based on Boolean logic and uses Boolean functions for the rules of each state. The execution of the network is visualized by an animated indicator, which moves from one element to the next as the network executes. This animation shows how an electrical circuit would execute; however, this is not identical to the behaviour of genetic networks. The state logic does not allow complex rule sets to be formed which limits the possible networks that can be constructed within this system. The simulation is also limited by being highly discrete and by having only two concentration levels: none or all. The program uses circuit design methodologies to simulate networks. These methodologies are thought to apply to genetic networks [28]; however, this is still in question and the methodologies do limit the number of networks portrayable.

2.3.4 Circuit Simulation

McAdam's [29] Circuit Simulation of genetic network behaviour model is a tool for predicting when activation occurs between connected elements. This tool visualizes the overall organization of genetic networks by using a representation similar to electrical circuit diagrams. The circuit simulation uses different symbols to represent different genetic mechanisms and connects them together in a similar manner to circuit diagrams. Symbols can represent such elements as different types of genes

and proteins. Each gene is connected to other genes using *wires*. Each gene can have Boolean logic gates to represent repression and activation of the genes expression. Such operators as AND and NOT can be used. Sections of a genetic network can be grouped into functional units similar to how chips represent multiple interactions with electrical circuit diagrams.

The simulation's execution is calculated by determining protein concentrations from equations that calculate the balance between protein production and degradation. The execution of the genetic network is depicted through the use of timing diagrams common to circuit design. These diagrams show the varying signal or concentration of each element within the network. Multiple genes are shown at once, over a single time period allowing biologists to analyze and reveal a temporal ordering of gene expression. This circuit simulation is customized for genetic networks whereas Visual Object Net++ is not specific to genetic networks.

2.3.5 Genetic Network Analyzer

The Genetic Network Analyzer (GNA)[10] uses differential equations to simulate diffusion and thus network behaviour. Data is obtained through files containing the differential equations with threshold and equilibrium inequalities describing the behaviour of the genetic network. For each gene there are two threshold concentrations which are used to determine whether the gene is active or repressed depending on the current concentration of required protein. The algorithm calculates volumes of chemicals and reports the results in the form of a state transition graph showing the connections between genes. The piece-wise differential equations lead to the state transitions, which in turn can become attractor cycles present within the network.

Each connection has a '+' (activated), or '-' (inhibited). These line graphs plot concentration over time for each element showing a temporal history of each. The rate of expression is determined using a sigmoid function [45]. The *Bacillus subtilis* organism was used to test the system's simulation and it was found that with significant variation to the network parameters the simulation results were consistent with experimental results. However, the significant alteration to the parameters was troublesome and as a result the simulation network may be incomplete.

2.3.6 StochSim

StochSim [33] is a stochastic simulator that does not use diffusion to simulate molecular interaction. While this program is not specifically designed for genetic network models, it does simulate biochemical interactions similar to that of genetic networks. Each molecule can be used to represent either a protein or a gene. A molecule is represented as an individual software object and reacts according to concentrations and molecular rate constants. Each object can interact with another object depending on its type. There is a special type of object called multistate molecules, which has a set of binary flags associated with it. These binary flags represent the state that the molecule is currently in (ON or OFF). When all the flags are combined for a particular molecule its overall state can be determined. The simulation method is as follows: two objects are randomly selected out of the group of all the objects. Any two objects selected will interact based on a probability of interaction. A random number is generated for the interaction and is compared in a previously computed table that contains the probabilities of all possible interactions between all objects. If the probability is less than the random number, the particles react; otherwise they

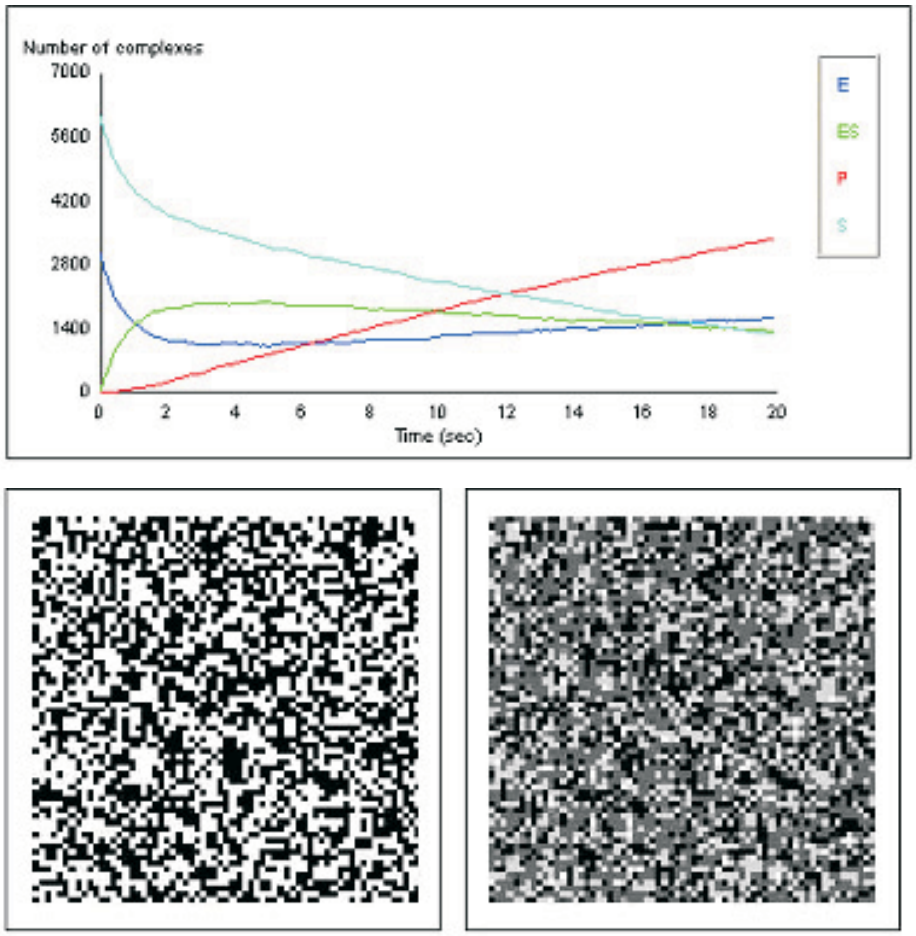

Figure 2.8: StochSim: (Top) A dynamic concentration line graph with number of complexes plotted over time in seconds, (Bottom) Two spatial grid visualizations of the simulation where on the left is an instantaneous snapshot and on the right is a time-lapse snapshot (ie. an average over an internval). [33].

do not. This process of selecting two objects is continued at each time slice. The simulation can be paused at any time and parameters may be changed to affect the outcome of the simulation. This simulation can be visualized in both interactively and after the simulation is completed. The interactive visualization consists of a concentration line graph, which is dynamically updated as the simulation proceeds (Figure 2.8 (Top)). The graph displays the objects (proteins and enzymes) involved in the reaction and outputs their concentration in the form of a coloured line graph. Once the simulation has been completed, a set of integer results can be viewed statically or as a line graph. As well, at this point the user can select which objects are to be shown. A zoom feature is also provided for both axes of the graph to allow for viewing of localized regions of data.

Biochemical interaction can also be simulated using nearest neighbours with a spatial grid (Figure 2.8 (Bottom)). The grid is a large array in which molecules are only able to interact with their surrounding neighbours. It allows for the spatial layout of the chemical system to be observed. The resulting data from this type of simulation is displayed in the grid with white cells representing highlighted states and black representing all other states. There are two types of snapshots the system can create: instantaneous and time-lapse. The instantaneous snapshot presents the square grid with the current state values within each cell at that specific time point. The time-lapse displays the average state of each cell over time. The average is displayed as a grey scale value within each cell depending on its value. While StochSim is not specifically designed for genetic networks, it could be used for them. StochSim presents a dynamic visualization of the simulation results and allows for interactive editing of object parameters during execution to change the outcome of

the reactions.

2.3.7 BioSim

BioSim [17] is a program that takes a different approach from most other simulators. This approach approximates missing information since often a large amount of information is unavailable. This is done by using qualitative data rather than quantitative data. BioSim has integrated qualitative reasoning [13, 21, 22] with a biological knowledge base to create a qualitative simulation environment. Qualitative values differ from numerical in that there is only positive, negative, and zero, which biologically can be considered as active, repressed, and stable. Qualitative processes are then used to simulate biological elements. The user can also initiate new processes by marking threshold values called landmarks. BioSim returns sets of possible behaviours in the form of behaviour trees. These trees show the user many different possible outcomes of the genetic network. The resultant trees are displayed in the form of a graph over a series of time points. At each time point either a circle, an up arrow, or a down arrow is present. The circle indicates steady expression, the up arrow shows an increase in expression, and the down arrow shows a decrease in expression. This is used to show the variations in concentration over time. This type of display leaves the user to look at the data and to ascertain the temporal ordering since no network diagram is produced. This approach is good when a model has a large amount of missing information however the simulation does not produce any quantitative data.

2.4 Information Visualization

The information visualization techniques discussed here have not previously been applied to the visualization of genetic networks. However, these techniques may prove to be useful across different application domains [7]. In particular these are viewing techniques and spatial organization techniques.

2.4.1 Viewing Techniques

Viewing techniques are visualization methods that change what one sees without changing the representation. Very simple examples of viewing techniques include rotations and zooming. The viewing techniques that relate to this research are lenses. These lenses allow the viewing parameters for a particular area of a visual presentation to be changed.

Detail-in-Context Lenses

Detail-in-context lenses expand a chosen region allowing details of the data to be viewed while still set in their context. These lenses usually achieve this effect through some use of distortion. While there are many variations of these lenses in the literature that provide detail-in-context views, they all have different appearances and have been given many different names, including Bifocal Display [2], Generalized Fisheyes [14], Graphical Fisheyes [40], Perspective Wall [26], and Document Lens [35]. Leung's taxonomy [24] discusses the distinction between these distortion-oriented presentation techniques. Carpendale [8] provides a framework that supports the distinctions between these viewing techniques while algorithmically uniting them. This framework allows for the inclusion of more than one type of detail-in-context lens

within a given application.

Magic Lenses

While detail-in-context lenses magnify a region of the representation, Magic Lenses [6] can reveal some alternate aspect of the representation for the specific region they cover. For example, placing a Magic Lens over a region of a modelled hand might reveal the bone structure or the muscle structure. This concept has been incorporated into sets of tools [6] and filters [42].

2.4.2 Spatial Organization Techniques

These techniques address the positioning of information or data on a computer screen. The spatial organization technique chosen depends on the type of data being presented. For this research, graph layouts are particularly interesting since they pertain to the viewing of genetic networks.

Graph Layout

Graph layout is useful for data that can be represented by a number of nodes with connections existing between these nodes. The connections or relations create the structure of the graph or a subset of graphs or trees. There are many differing techniques discussed in literature for displaying these relationships. Some examples of these relationships being displayed as trees are the Hyperbolic Browser [23] and Cone Trees [36]. Noik has done research into emphasis techniques for visualizing graphs [32], and Battista et al. [5] has provided a taxonomy which overviews a large number of algorithms for graph drawing.

2.5 Summary

The network structure visualizations provide viewing environments for the structure of genetic networks. Directed graphs are most commonly used to depict network structure, since these graphs are capable of showing the relationships between genes. The advantage of this method is that directed graphs are a well established and a common method of displaying network structures. Furthermore, they are well understood as a way of describing connection information. The disadvantage is the common problem of edge-congestion [32, 5] produced within directed graphs. This problem escalates as the network size increases, causing a considerable amount of visual clutter and obstruction of connection information. While the viewing tools such as interactive node placement, zooming, and filters may alleviate some of the problems of edge-congestion, they do not provide a complete solution.

Network behaviour is modelled to output simulated expression results. Boolean network models are among the simplest and least computationally taxing method of simulation that is available. Boolean network models can model the explicit structure of genetic networks but do not include any probabilistic effects. Differential equations and stochastic simulation offer more computationally expensive solutions but gain a higher degree of simulation accuracy than Boolean network models. Qualitative simulation allows one to work with speculative data.

As part of their output all of these simulations produce line graphs that show the expression of the genes in the simulation as they vary over time. These gene expression line graphs can be used to compare expression data from multiple genes in order to find a temporal ordering of gene expression. While these graphs visualize

the results of these simulations, there is no visualization of the interaction between genes and proteins that occurs as the simulation is in progress.

The nature of genetic networks, being a number of genes interconnected through regulatory proteins, can be generalized to a graph layout problem. There has been considerable research in information visualization and graph layout that may prove applicable to this problem. GeneVis applies ideas from distortion-oriented viewing techniques [8] and the concept of magic lenses [6] to the problems associated with the visualization of genetic networks.

Chapter 3

GeneVis: Simulation of Genetic Networks

GeneVis is a software tool that models occurrences of gene-protein interaction to simulate prokaryotic genetic network behaviour. The dynamic simulation is based on probabilistic gene-protein interaction. The main visualization is a symbolic 2D representation of the cell with genes and proteins. Additionally, supporting visualizations of the histories of each gene's expression are provided to allow for comparison between simulations and biological laboratory results.

The first section of this chapter is an introduction to the GeneVis software tool. Section 3.2 presents the simulation model used to model gene-protein interaction. Within this section, the simulation environment, gene modelling, and protein modelling are discussed. The simulation algorithm is then discussed in Section 3.3, followed by an explanation of the gene expression history visualization in Section 3.4.

3.1 Introduction

Genetic networks depend on the interaction of proteins and genes. Genes can express proteins, which in turn can regulate the expression of other genes, creating a network of dependence. These interactions constitute the network's structure, and produce both the development and functioning of organisms. Even small genetic networks are complex and unravelling these complexities is a daunting but an important task. Computational simulation and visualization may be used to provide

aides in these endeavours. To this end we have developed a program, which simulates genetic regulatory interactions and visualizes these interactions dynamically. What makes our approach different from previous work is a focus on simulating the interactions between genes and proteins. Additionally, past approaches often visualized the results of the simulation and not the simulation dynamics. GeneVis visualizes both the simulation dynamics and the results of the simulation. The visualization of simulation dynamics provides visual explanations of network behaviour and is useful for constructing or editing a network. Within the visual simulation environment it is possible not only to watch the simulation progress but also to pause it and adjust the parameters for individual genes or proteins. On resuming the simulation the adjusted parameters take effect immediately.

3.2 Simulation Model

Our simulation models gene-protein interactions to mimic genetic network behaviour. To model genetic interaction, factors effecting interaction need to be identified and incorporated into the simulation. GeneVis models five factors that affect gene-protein interaction: the protein's direction of movement, the protein's distance of movement, the protein's life span, the protein's operator site binding, and reversible binding (or unbinding). The value of each factor is based on statistical data for the specific protein. By randomizing these factors separately it allows for probabilistic interaction between genes and proteins. A random number generator is used and each of these elements is independently seeded with the system time. By seeding each individually a more natural probabilistic randomness occurs.

The development of GeneVis as a software tool was done with the specific intention of extending is capabilities. GeneVis was designed with a flexible program architecture that can be easily modified to incorporate additional gene-protein interaction factors as required. During the creation of the simulation environment, a framework was created, where all the proteins and genes were made as components. Each component has its own behaviour and can interact with other components in the simulation. This underlying component structure provides the freedom to adjust or add particular factors of interaction through these components without having to re-write the entire simulation. Also, each instantiation of a component can behave independently. For instance, each protein in the simulation operates independently from every other protein.

This approach considers the synthesis of an emergent simulation from the interactions of the individual aspects. In GeneVis the dynamics of genetic regulation are represented independently of their physical manifestation.

The simulation is composed of a set of proteins and a set of genes that are situated in an environment. The environment is passive as it takes no actions itself, but the gene's expression behaviour changes over time since proteins are free to move throughout the environment and interact with genes. Each gene and each type of protein is given individual rules that define their behaviour and their interactions with other genes and proteins within the environment. They react to the local state of their environment, which can include genes and proteins, without reference to - or knowledge of - any global goals. This creates the simulation in which the complexity of the system emerges from the accumulation of these local interactions.

3.2.1 Simulation Environment

The simulation environment is a 2D grid representing a symbolic view of the cell. Each grid cell can be occupied by a protein or a gene and is used to track the position of these elements. The grid size can vary between an array of 100 by 100 to an array of 1000 by 1000 cells, which allows the user to control the resolution of the simulation. Figure 3.1 shows this environment.

Within this environment there is a circle which represents the chromosome of the prokaryotic organism. These organisms typically have a single looped chromosome. Genes within this chromosome are located around it according to their base pair position. The entire chromosome represents a range of base pairs which is specified in the input file (see Chapter 5). The base pair range starts and ends on the far right hand side of the circle. The circle representation is used to provide a visual cue for relative base pair positioning. The circle is located centrally within the grid and, to maximize freedom of protein movement, all edges of the grid are "wrapped".

3.2.2 Genes

Inside the 2D environment genes are located around the chromosome and remain stationary for the entire simulation. Genes have two purposes within this simulation: genes can have proteins bind to their operator sites which regulates their expression, and they can produce/express their own proteins.

To allow genes to have their expression regulated by protein binding, each gene has an associated rule set. There is one rule-set per operator site and each gene can have any number of operator sites. The rule-set takes required regulatory proteins as input and possibly produces another protein as its output. The rule-set for each

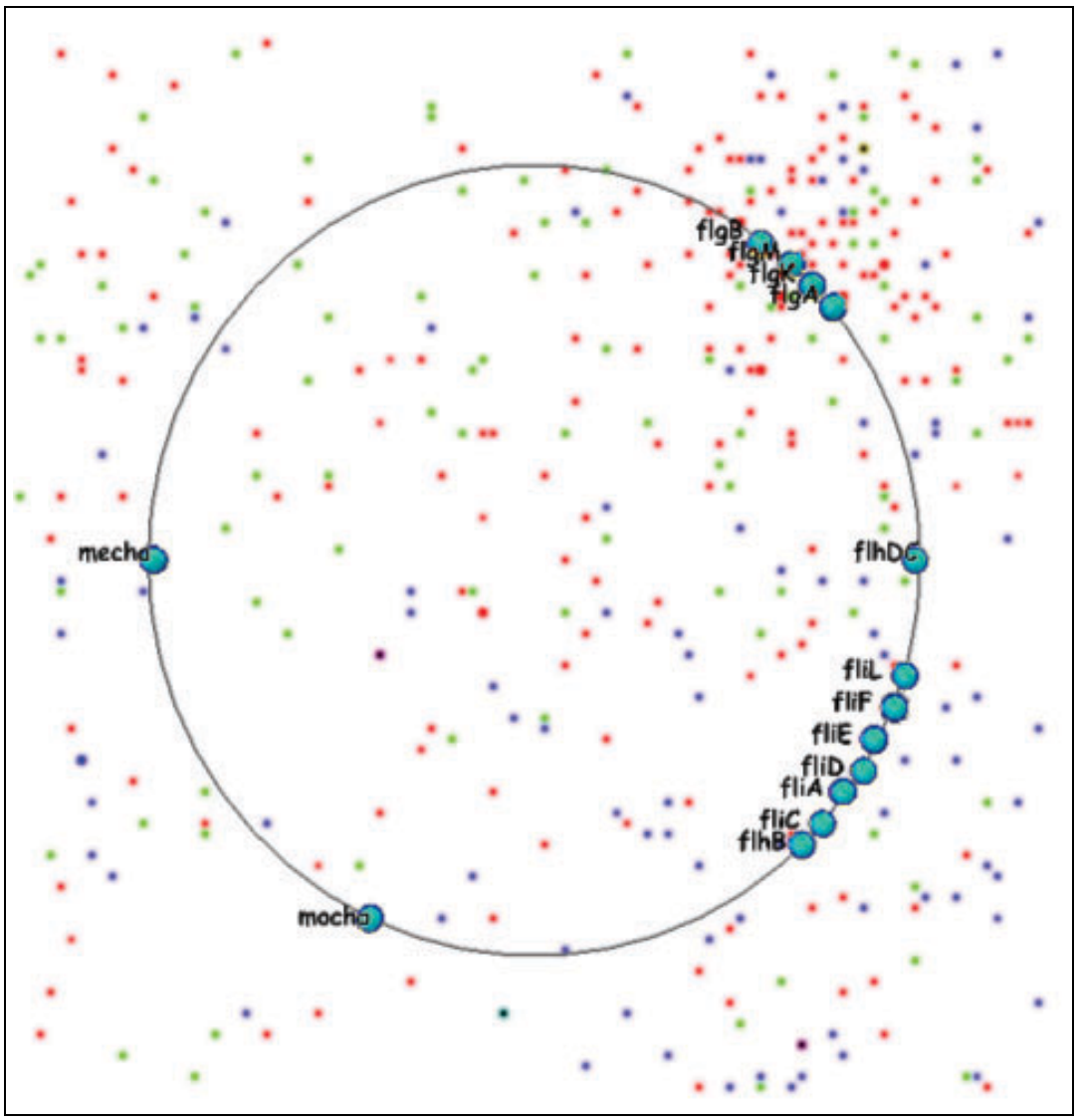

Figure 3.1: GeneVis: The circle represents the organism's chromosome with genes represented as spheres which are located around the chromosome according to their base pair position. Proteins are located throughout the environment.

gene's operator site is shown below.

If the gene's operator site is vacant *then*
the gene produces at its basal activity rate
If the gene's operator site is bound with the right activator protein *then*
the gene will express an increased amount of protein (Activator Factor).
If the gene's operator site is bound with an inhibitor protein *then*
the gene will express a decreased amount of protein which is often none (Inhibitor Factor).

The genes produce proteins as calculated based on their rule-set. The calculation considers whether an activator or inhibitor is bound to an operator site and the degree to which regulatory proteins impact or change the gene's expression. The number of proteins expressed from a gene is calculated as follows:

Number of Expressed Proteins = BA + (BA * (AF * AB) + BA * (IF * IB))

BA = Basal Activity (floating pointing number >= 0)

AF = Activator Factor (positive floating point number)

AB = Activator Bound (0 if not bound, 1 if activator is bound)

IF = Inhibitor Factor (negative floating point number)

IB = Inhibitor Bound (0 if not bound, 1 if inhibitor is bound)

This equation shows that if either an activator or inhibitor is bound to the operator site the expression of the gene will be increased or decreased according to the product of the Basal Activity and the Activator/Inhibitor Factor. This equation is calculated for each gene on every time step and the number of expressed proteins represents the number of proteins that gene will produce on the current time step. As genes change state between activated and inhibited (binding and unbinding of proteins) their protein production will subsequently change. The activator factor is a positive number while the inhibitor factor is negative, causing the corresponding increases and decreases in gene expression depending on the bound protein.

Figure 3.2 is a diagram of GeneVis's visual representation of a gene. A gene is displayed as two concentric circles (Figure 3.2(a)). The outer circle represents the gene's operator site(s). Proteins can attach anywhere on the outer circle and are then considered bound to an operator site. Figure 3.2(b) shows a regulatory protein bound to the operator site. If just one point on the gene's circumference is used for an operator site, then the probability of an intersection is too low. When the appropriate activator or inhibitor protein is bound, the gene adjusts its expression

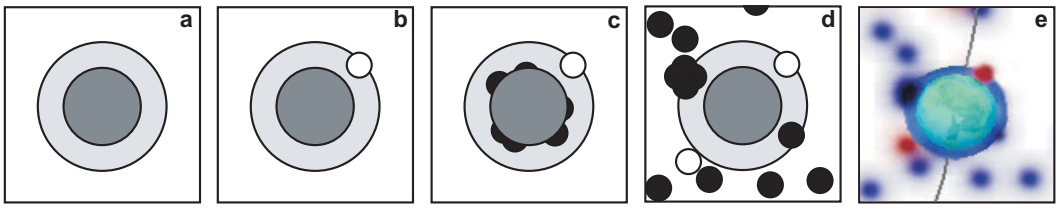

Figure 3.2: (a) an inactive gene, (b) a gene with a bound regulatory protein, (c) a gene beginning to express proteins, (d) a gene continuing to express, and (e) an actual screen shot of the gene expressing in GeneVis.

of its own proteins (Figure 3.2(c)), which are displayed as they emerge from within the inner circle (Figure 3.2 (c), (d), and (e)). It is these protein-gene interactions that make the network function.

3.2.3 Protein

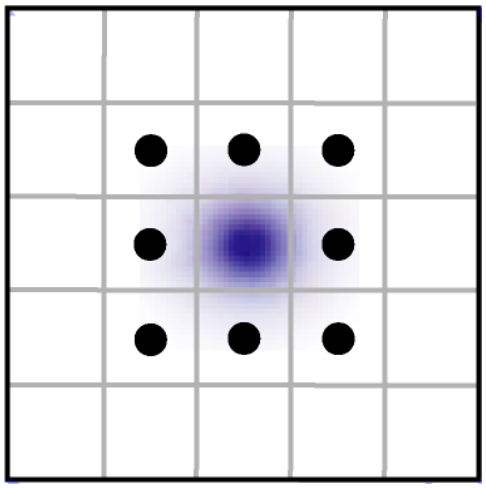

Figure 3.3: The eight possible directions a protein can move on a 2D grid.

Proteins in GeneVis are unique because they are the only elements free to move

with the environment. Both the 2D grid and all of the genes remain stationary during simulation. The movement of proteins is responsible for creating the possibility of interaction between genes and protein. While the movement of proteins is typically diffusive it may not be truly diffusive [12]. A very simple approximate model has been chosen to model gene-protein interaction. In GeneVis protein movement is modelled by randomly moving each protein in the simulation. The grid imposes a directional restriction (Figure 3.3) on the proteins' movements to one of eight directions (up, down, left, right, and the four diagonals).

Each protein also has a maximum distance the protein can move in one time step. This maximum step size controls the rate of protein dispersion. The step size is randomly selected between 0 and the maximum. The protein is then moved the selected step size in the randomly selected direction. This ensures proteins do not repeatedly step over entire genes when the step size is large. The larger the step-size the faster the proteins will disperse.

Each protein has a life span. The protein's life is decreased each time step in the simulation until its life is zero, at which point the protein is fully decayed and is removed from the simulation environment.

The binding of proteins to genes is accomplished by having each grid-cell covered by a gene (the number of grid cells a gene covers depends on the resolution of grid) assigned the gene's *ID number*. This allows intersection testing with only one comparison between the grid value and the location of a protein within the grid (Figure 3.4). In GeneVis protein binding can occur anywhere within the cells occupied by the gene and not just the outer edge 3.2 of the gene. This increases the number of possible binding events. If a protein is intersecting a gene then the

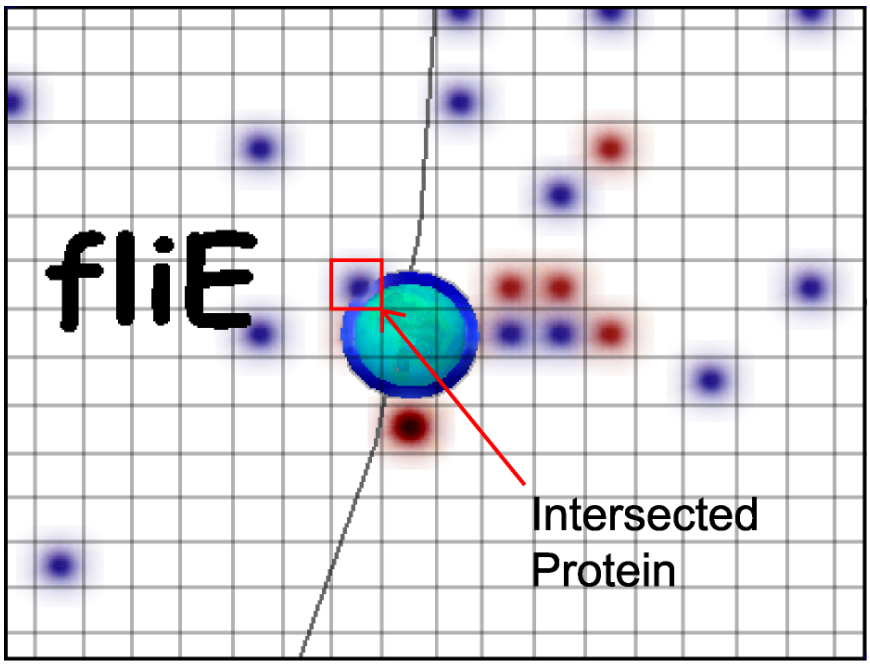

Figure 3.4: A gene covering a number of cells within the 2D grid. An intersection test between the gene and all other proteins has caused the protein marked as an "intersected protein" and about to become bound.

operator site's binding affinity percentage is used to determine whether the protein binds to the gene. Once a protein is bound there are two circumstances in which it can unbind from a gene. The first is when the protein becomes dislodged from the operator site. This effect is simulated by having a reversible binding factor which is tested every time step to determine if the protein unbinds from the gene. This factor is the percentage chance of reversible binding for that gene. The second circumstance is when the protein decay causes it to unbind from the operator site. This effect is simulated by having a protein decay factor which can cause the protein to unbind from the gene when it reaches zero.

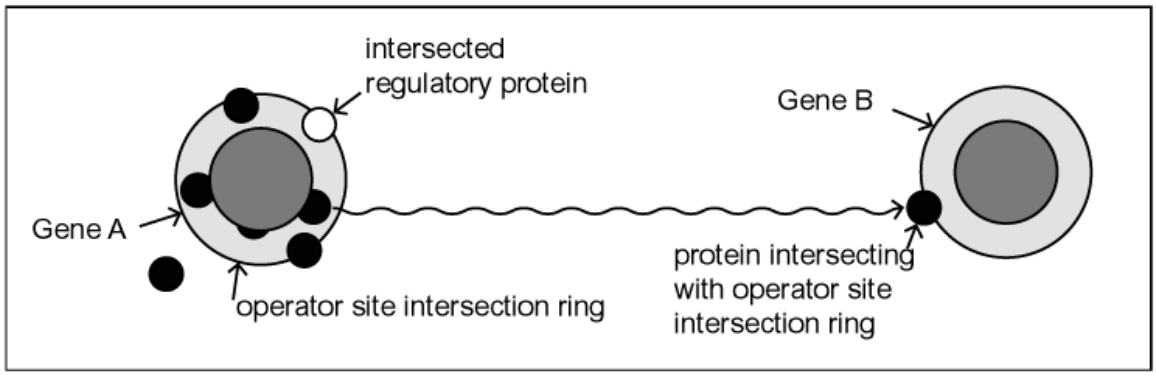

Figure 3.5: A single protein bound to the gene promotes expression. A protein moves randomly through the environment until intersecting with a gene that requires it.

Figure 3.5 shows how proteins spread throughout the environment and then interact with other genes, causing their subsequent expression. On the left a single protein is bound to Gene A. This bound protein is causing Gene A to express proteins which are dispersing throughout the cell. Over time, proteins will disperse and eventually a protein may intersect with Gene B. When the protein intersects with

Gene B the affinity of binding for Gene B's operator site determines whether the protein binds. If the protein binds, Gene B will subsequently start expressing its produced protein at a new rate based on this bound protein.

3.3 Simulation Algorithm

The simulation algorithm runs for every protein and every gene at each time step. Following is the pseudo-code for the simulation algorithm:

1. For each gene a calculation occurs (see the calculation in Section 3.2.2) to determine the number of proteins that have been produced for this time step. This determines the number of proteins expressed by a gene.

2. The released protein positions are updated by moving them a random number of cells in a random direction (one of the eight possible directions).

3. Each active protein has one unit subtracted from its life span. This subtraction models protein decay.

4. Test each protein for intersecting a gene:

   if the grid position of the protein overlaps a gene, *then*

   > *see if* the protein is a required activator or inhibitor *then*
   >
   > > *if* so, that protein may bind based on the gene's affinity.

5. If a protein's life has expired meaning the protein is fully decayed or dead (<0) then:

   Delete the protein from the environment.

   If the protein was bound to any gene, free that operator site.

3.4 Gene Expression Visualization

Expression analysis has become an invaluable asset to biologists in the study of genetic networks [1, 11, 30]. Expression analysis is a technique which measures the expression of genes. The new methods available to experimentalists allow the simultaneous measurement of large number of genes.

In GeneVis, gene expression histories show the number of alive proteins (*produced proteins − decayed proteins*) in the simulation environment at each time point over an interval. These gene expression histories are similar to expression analysis results. This allows for a direct comparison between laboratory and simulated results. GeneVis represents each gene's expression history, using an existing visualization method [11] where one colour is used to indicate *no expression* and a second colour to indicate *expression*. These colour values are plotted vertically in a coloured rectangle with intermediate levels of expression shown as a gradient between the two colours. In the simulation, gene properties can be changed and the virtual experiments can be re-simulated to identify differences and similarities in expression. All gene expression histories are updated in dynamically (Figure 3.6).

The gene expression history visualization is as follows. With each gene expression history the top rectangle depicts the current gene's history of expression and the bottom rectangle visualizes expression history results retrieved from either live or simulated experiment. These expression histories are imported from a file. Each rectangle indicates the expression history of the gene where the left side is time step 0 and the right side is the current time step in the simulation. As the simulation proceeds, the expression history is compressed toward the left to show the new

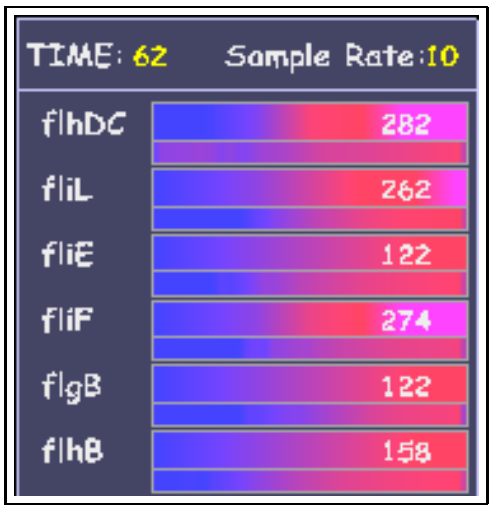

Figure 3.6: Gene expression history: this image shows six genes that are being simulated with a loaded expression history in the bar below each gene's simulated expression history. The loaded expression history is being shown in full rather than dynamic mode.

current expression data. The number in the right side of the rectangle shows the gene's current number of active proteins (the number of expressed proteins minus the number of decayed proteins). The gene's expression level is shown using the colour gradient of the two colours where blue indicates no expression and red indicates full expression. When displaying laboratory results they can be shown either in full, showing all the data from zero to the end, or from zero to the current time step in the simulation. All expression bars are positioned on the right side of the screen for comparison between individual genes. (Figure 3.6).

To facilitate the comparison of simulation runs, it is possible to output the simulation's expression histories to a file for numerical comparison, plotting and/or more sophisticated statistical analysis. This allows for smaller variations to be identified.

3.5 Discussion

The GeneVis simulation model for genetic network behaviour uses gene-protein interaction and five mechanisms of interaction. Through having each component act as an individual, an emergent complexity occurs as each individual component interacts with others in the environment. The component architecture allows for additional mechanism to be added and each can be adjusted within the visualization through the property dialog box (see Chapter 5). This allows biologists to easily alter their genetic network parameters to model its behaviour within GeneVis.

Chapter 4

GeneVis: Visualization Tools for the Observation of Genetic Networks

Models of genetic networks can be used to create simulations and visualizations, helping us form mental constructs of their behaviour and thus further our understanding of them. In order to aid this comprehension, we have made GeneVis, a visual environment for exploring the behaviour and structure of genetic networks. In this environment the simulated entities are spatially organized and can be adjusted interactively in order to help illustrate and support the exploration of mental concepts. Moreover, different visualization techniques can assist in understanding different aspects of the same data set.

This chapter discusses the visualization environment which contains several visual representations including: a gene expression history representation, a *protein interaction representation*, a *protein concentration representation*, and a *network structure representation*. The protein interaction representation shows the activities of the individual proteins. The protein concentration representation illustrates the relative spread and concentrations of the different proteins in the simulation. The network structure representation depicts the genetic network dependencies that are present in the simulation. Figure 3.1 shows a protein interaction representation in GeneVis. Of the visualizations available in GeneVis, gene expression histories are commonly used when visualizing with this type of data [11, 43]. However, the other three are

new visualizations created especially for visualizing genetic regulatory networks by applying, adjusting, and modifying several visualization techniques [8, 5, 42, 6, 23].

GeneVis incorporates several interactive viewing tools. These include animated transitions from the protein interaction representation to the protein concentration representation, and from the protein interaction representation to the network structure representation. There are also three types of lenses: *fuzzy lenses*, *base pair lenses* and the network structure *ring lens*. With a fuzzy lens an alternate representation can be viewed in a selected region. The base pair lenses allow users to reposition genes for either better viewing or to minimize interference during the simulation. The ring lens provides for detail-in-context viewing of individual levels within the genetic network structure representation.

This chapter is organized in two sections. The first section describes the visualization environment for the simulation. This includes the two different visual representations as well as the fuzzy lenses and the base pair lenses. The second section discusses the network structure visualization of the gene-protein interactions. The ring lens is also described in this section.

4.1 Visualizing the Simulation

GeneVis uses random protein movement to simulate the interactions in genetic networks. By this random movement, proteins disperse throughout the simulation environment. The cell is modelled as a grid that wraps around on all four sides so that proteins can circulate through the environment continuously (see Figure 3.1). How each protein moves is randomized in the choice of distance and which of the eight

possible directions to take.

In GeneVis the simulation starts in the initial state in which no proteins are present and the genes are operating at their basal activity level. This means that each gene's expression is neither promoted nor inhibited by a bound protein [25]. This basal level activity results in the production of some proteins, which spread throughout the environment and start interacting with the genes, promoting or inhibiting their expression.

4.1.1 Protein Interaction Representation

In visualizing the interactions between the individual proteins and genes, we:

- use the base pair positioning [15] to depict the actual locations of the genes on a circular chromosome (Figure 3.1),
- stylize the visual representations of individual genes to make the operator sites visible (Figure 3.2),
- make the random motion and decay of the individual proteins explicit, and
- show the change in activity rate for each gene as a result of the network dynamics.

Figure 3.1 shows a screenshot of the protein interaction visual representation of a GeneVis simulation. The large circle in the middle of the screen represents the chromosome. The filled circles located on the chromosome are the genes, and the small coloured particles, which are spread throughout the grid, represent the proteins. This protein representation is created as a texture mapped square in which the

colour is saturated in the center and attenuated towards the edges. This attenuation gives the proteins a fuzzy circular appearance. We will refer to these as discs. The attenuation keeps the proteins visually distinct as they move around the environment. The colour of the disc signifies which protein type it represents, and can be set by the user. For each time step of the simulation, all protein positions and life spans are updated. Furthermore, genes are activated or deactivated depending on regulatory proteins that are in their proximity, and according to their binding rules (see Chapter 3).

A strength of this dynamic visualization is that interactions between genes and proteins can be seen as they occur in the simulation. For instance, a protein bound to a gene's operator site is visible, as is the change in activity that results when a required promoter protein binds to a gene. One can see the burst of genetic activity and the resulting release of new individual proteins into the environment.

The genetic network dynamics are visualized as the simulation proceeds. The simulation and visualization can be paused or restarted at any time. The coupling of the simulation and visualization allows for interactive network construction and debugging of network dynamics (for additional information see [4]).

4.1.2 Protein Concentration Representation

The genetic network dynamics can also be visualized in a more macroscopic manner, by showing protein concentrations, rather than the position of individual molecules. The probability of a gene's expression being affected increases and decreases with the chemical concentration of the required proteins.

In terms of genetic dynamics, the simulation becomes much more interesting

once proteins have increased sufficiently in number and have spread throughout the environment. When viewing the individual proteins, it can be difficult to gauge whether the proteins have dispersed throughout the entire system. The concentration visualization of the simulation can be used to more readily visually identify when the protein concentrations have increased. In GeneVis, the proteins can be viewed as individuals, as concentrations, or at varying representation levels that exist in-between. Concentrations show the spread of the proteins present, thus providing a more general view of the system dynamics.

In the protein interaction representation each protein molecule is represented as an attenuated disc. Conceptually, the protein concentration representation is created by using a larger single attenuated disc to represent several protein molecules. The size of this disc visually covers the same area as the proteins it represents. This attenuated disc is centered at the location of one of the proteins it represents. The rest are not drawn.

Figure 4.1, top image, shows the protein interaction representation, Figure 4.1, middle image, shows the protein concentration representation at the same point in a simulation. Notice how it is hard to tell if the proteins are uniformly distributed in the protein interaction representation. This information is more readily apparent in the protein concentration representation. In Figure 4.1, middle image, one can see that that proteins are coming close to having spread throughout the whole environment. Additionally, the protein colours can be adjusted so that only one protein type is displayed. This allows one to see when specific protein types are dispersed, as illustrated in Figure 4.1, bottom image.

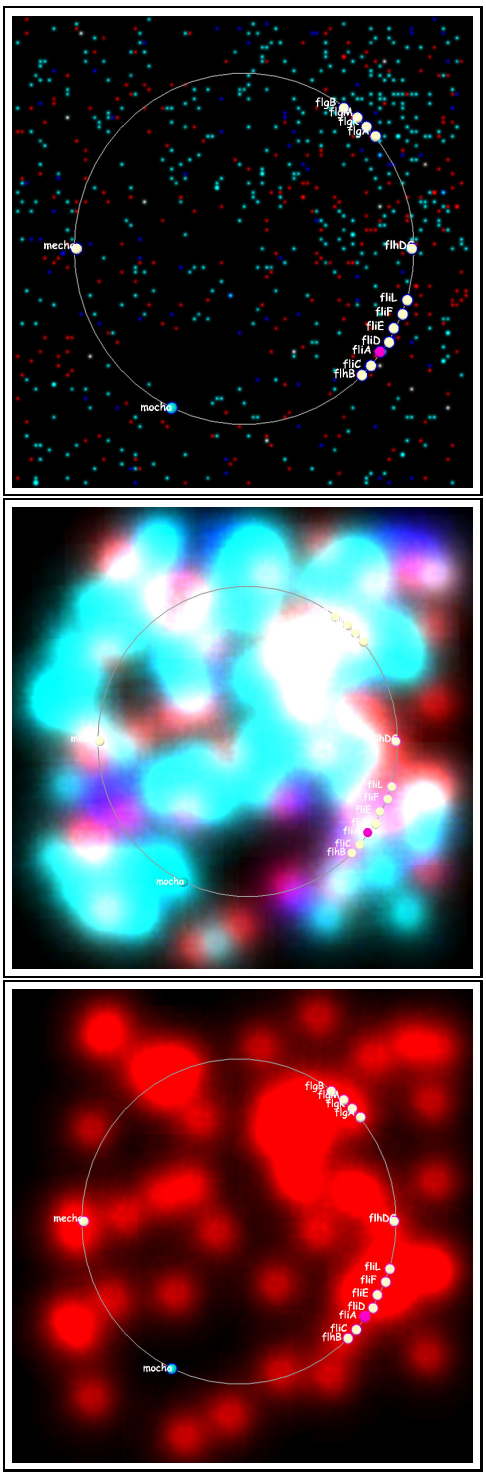

Figure 4.1: Top: protein interaction representation. Middle: protein concentration representation. Bottom: protein concentration representation for one protein type.

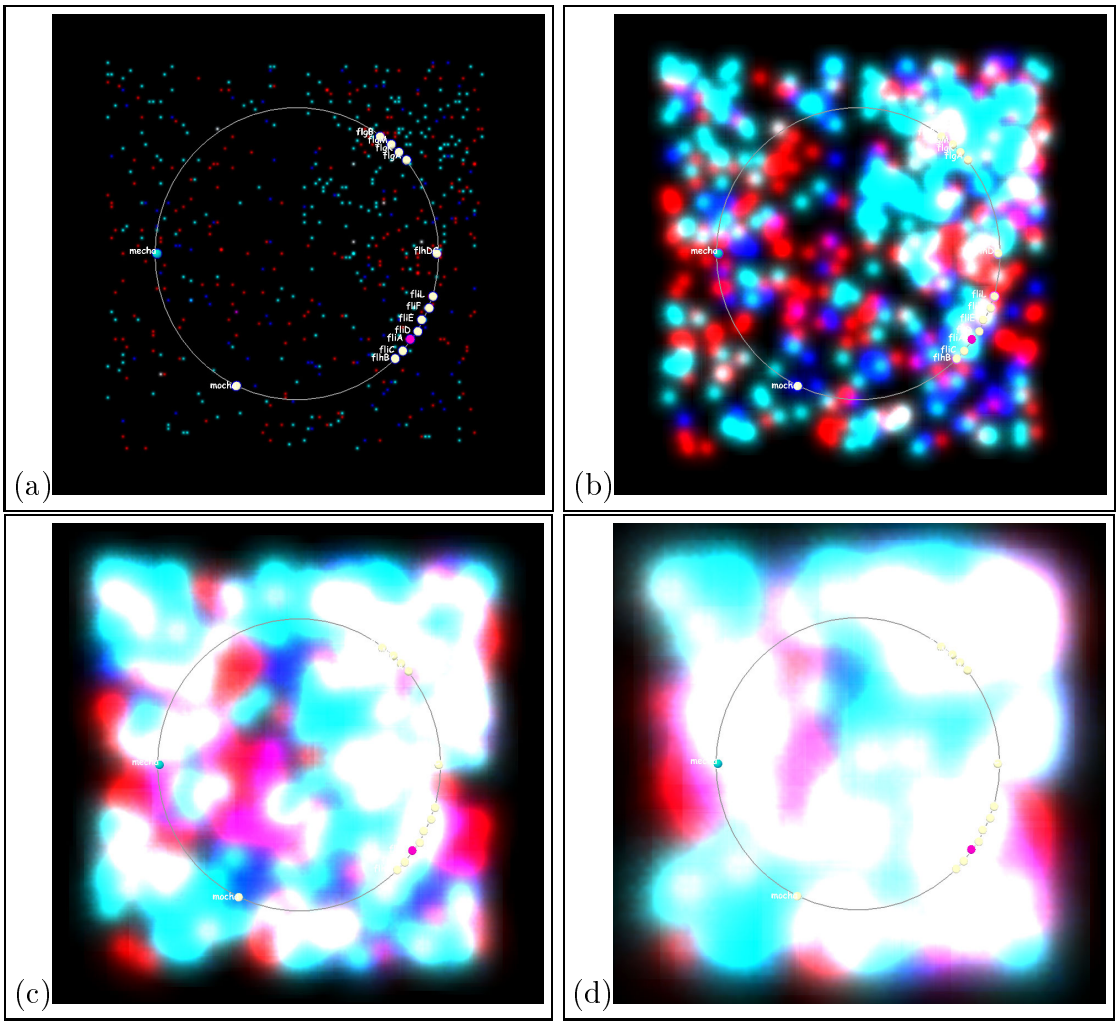

Figure 4.2: Representational Transformation: (a) *Protein View*: 100% displayed, with individual proteins viewable, (b) *Transition View*: 65% displayed, with small concentration discs viewable, (c) *Transition View*: 35% displayed, with larger concentrations discs viewable, (d) *Concentration View*: 1.56% displayed, with concentrations viewable.

4.1.3 Representational Transition

Many visual representations of complex data or concepts are possible. For instance, the concept of a number can be represented in many forms such as binary or decimal. Both of these representations are valid and useful, however the decimal representation makes information about powers of ten more accessible, while the binary representation makes information about powers of two easier to find [27]. Similarly, when we create visual representations as part of our visualization process it is our intention to reveal particular aspects of the data. In the previous sections we presented two visual representations of genetic regulatory dynamics: protein interaction and protein concentration.

Representational transition provides varying degrees of detail within the simulation visualization. The detail is varied from individual proteins, in which 100% of proteins are drawn individually, each as its own disc, to general concentrations, in which each disc represents many proteins. Figure 4.2 (a), shows the *protein interaction view*, in which individual proteins are displayed. This view is equivalent to the representation in Figure 3.1. Figure 4.2 (d), shows the *protein concentration view*. In reverse order, changing from the concentration representation to the interaction representation, the attenuated discs that represent groups of proteins become smaller until they represent individual proteins. This reverse direction can be seen from Figure 4.2 (d) to Figure 4.2 (a). Representational transformation is created by changing the size of attenuated discs, the number of proteins a disc represents and the number of discs shown. The attenuated disc's size covers the same area that the proteins it represents would cover.

Displayed	Proteins represented by single disc	Relative disc size
100%	1	1.0
50%	2	2.0
25%	4	4.0
12.5%	8	8.0
6.25%	16	16.0
3.125%	32	32.0
1.56%	64	64.0

Table 4.1: Representational transition: this table shows how the percentage transformed relates to the number of proteins represented by an attenuated disc and the size of that disc.

Table 4.1 shows how the representational transition between protein interaction representation and protein concentration representation is calculated. As the number of proteins represented by each disc increases, the number of discs decreases. When a disc represents more than one protein it represents that number of proteins by its increase in size and the number of proteins a particular disc represents is estimated as shown in Table 4.1. For a protein interaction representation, one protein is represented using one disc with a relative size of 1.0. At 50% displayed, two proteins are represented using one disc with a relative size of 2.0. At 25% displayed, four proteins are represented using one disc with a relative size of 4.0. This is continued on until reaching a cap of 1.56%, where 64 proteins are represented using one disc with a relative size of 64.0. This cap is used to prevent representations that contain too few large discs for lower concentrations.

Regardless of whether a disc represents a single protein or a group of proteins it

is positioned according to its center. A disc representing a single protein is placed according to that protein's position in the simulation. The location of discs that represent multiple proteins is resolved as follows. Each disc is centered at the location of one of the proteins it represents. This location is chosen from the locations of the proteins that have been alive in the simulation for the longest. The longest-living protein's positions have been most often randomized, making this position the most representative of the protein spread in the environment. Since we are taking a subset of location coordinates from a randomly distributed set of coordinates, the subset will also be randomly distributed throughout the area to which the proteins have dispersed in the simulation. Since these larger discs are located randomly, they can overlap. This overlapping causes RGBA (red, green, blue, and alpha) disc colours to add. If the added values exceed the maximum they are clamped to the maximum.

4.1.4 Fuzzy lenses

Fuzzy lenses have been implemented in GeneVis to provide access to alternate representations in different areas of the visualization. Lenses [6, 9, 24] are variable sized regions that can be moved over the visualization to reveal different information.

There are three Fuzzy lenses available: a *concentration lens*, which provides a concentration view of the simulation (Figure 4.3, top), a *protein lens*, which provides the individual protein view of the simulation (Figure 4.3, middle), and a *dual lens*, which shows both the concentration view and the individual proteins (Figure 4.3, bottom). Each lens is defined over a viewable region in which the lens's representation type is enforced. The regions are movable and resizable, so that any area of the visualization can be viewed within the lens (Figure 4.3). The lenses are *fuzzy* in

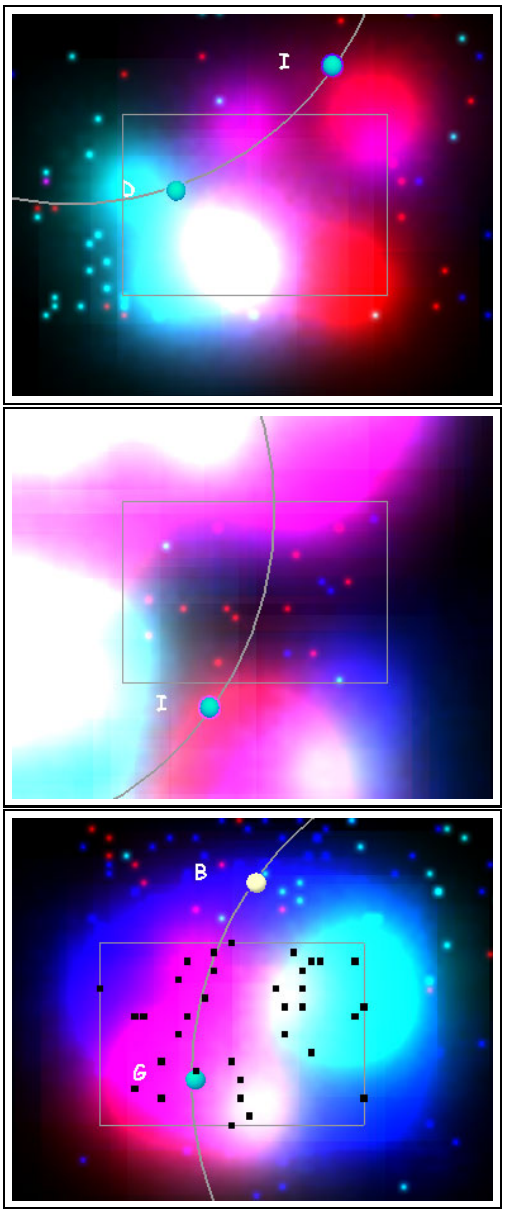

Figure 4.3: Fuzzy lenses: (Top) *Concentration Lens*, (Middle) *Protein Lens*, (Bottom) *Dual Lens*.

that the discs that represent the proteins are allowed to overlap the lens' borders. If discs that happened to be near the edge of a lens were cropped, the resulting visual impression of concentration would be affected. Drawing the discs fully, according to their central location resolves this. Since the discs are semi-transparent, the alternate representation on the other side of the lens boundary is also visible (Figure 4.3). With the exception of their fuzzy edges these lenses relate directly to the concepts presented as Magic Lenses [6] in that an alternate representation or a combined representation is shown within the lens.

4.1.5 Base Pair Lens

GeneVis simulates genetic networks for prokaryotic organisms. In these organisms a chromosome is typically a flexible loop. In GeneVis, this is represented as a circle. The genes in the network are located on this circle according to their base pair coordinates [15]. Within the chromosome, genes with related functions may be grouped closely together [15]. When genes with close base pair positioning are visualized within GeneVis, their representations may overlap due to limited resolution (Figure 4.4, left image). In addition to the visual crowding, the overlapping of operator sites can adversely affect the simulation. To rectify this problem, GeneVis includes the *base pair lens* that allows the user to interactively separate the genes and then proceed with the simulation.

The base pair lens consists of four sliders. Figure 4.4 shows a diagram of the interaction with the base pair lens when the left mouse button is used. Note that on the right side of the left image there is an area of the chromosome where genes are closely clustered. Moving the top handle to the left will expand the top right-hand

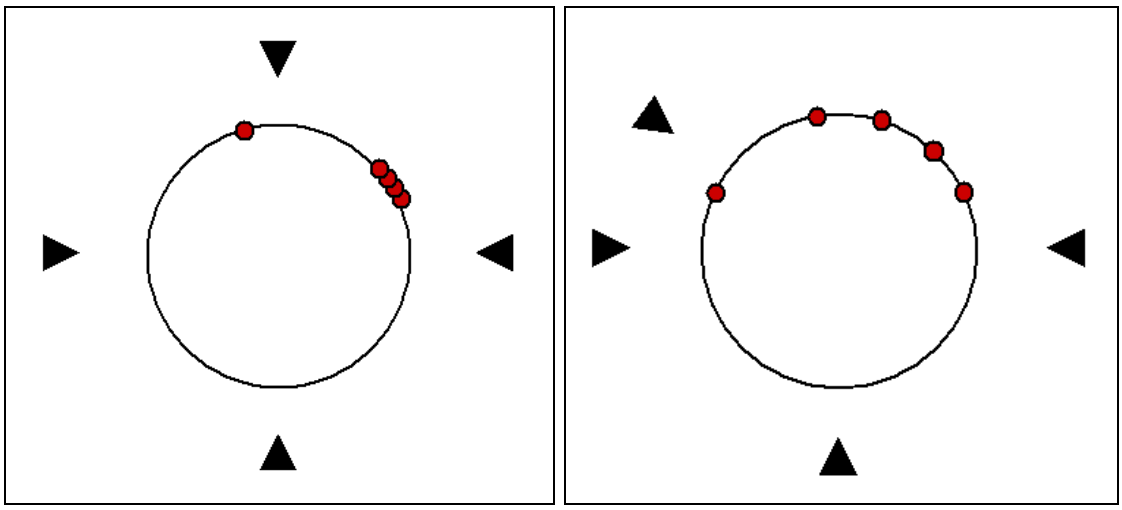

Figure 4.4: This is a diagram of the base pair lens: (Left) genes clustered on the left side of the chromosome, (Right) genes distributed more evenly.

quarter of the circle and compress the top left-hand quarter. The right image of Figure 4.4 shows how the black handles reflect the genes' new positions, which have now been distributed more evenly.

Alternatively, moving a handle with the right mouse button can alter the base pair range affected. Figure 4.5 shows a diagram of the interaction with the handles when using the right mouse button. In Figure 4.5 the right mouse button has been used to change the acting base-pair range from 0 through 10 (Figure 4.5(a)) to 0 through 5 (Figure 4.5(b)). Then the left mouse button can be used again to stretch out the 0 through 5 section over a greater circumference (Figure4.5(c,d)).

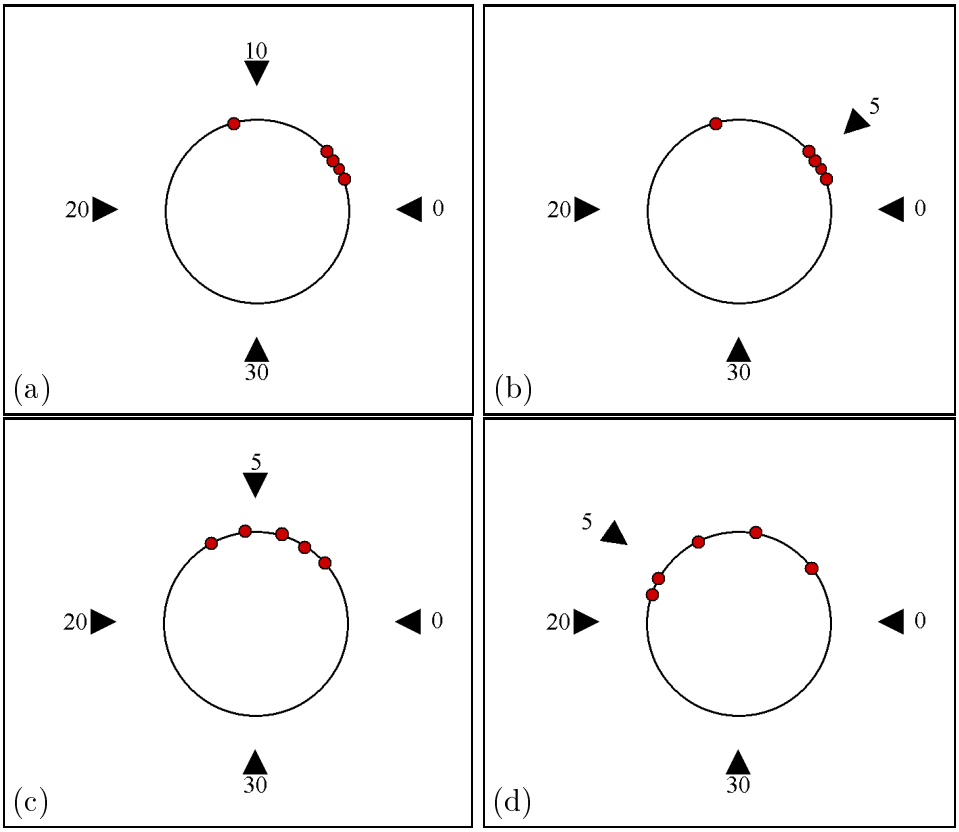

Figure 4.5: This is a diagram showing the interaction of the base pair lens: (a) the handles are evenly distributed across the base pair range which is 40 in this diagram, (b) the top handle is adjusted with the right mouse button to narrow the affected base pair range to 5, (c) the top handle is moved to the left using the left mouse button spreading the genes across a greater circumference, (d) the top handle moved farther left, further spreading the genes.

4.2 Visualizing the Genetic Network Structure

Visualizing the simulation in progress allows the user of GeneVis to examine the genetic network dynamics and compare the simulation results to actual wet-lab experiments. However, biologists are also concerned with the way interactions between genes and proteins form the structure of a genetic network. This type of information is not apparent in either the protein interaction or the protein concentration views. Consequently, a visualization has been specifically designed that displays the genetic network structure by showing regulatory connections between genes through directed graph layouts. This section describes this structural visualization.

The network structure displayed always reflects the structure of the network that is currently simulated. The behaviour of the genes and proteins can be interactively adjusted, thus the network organization is calculated by analyzing gene-protein interaction during the simulation. Every gene is checked for the earliest time in which a regulatory protein binds with it and affects its activity level. This is used to place that gene within its appropriate level.

When viewing the dynamics of the network, sometimes a hierarchy can be seen in the early stages of the simulation. In this hierarchy, genes are grouped according to the proteins that regulate them. For example, gene-protein interactions of the flagella system of *E. coli* have been identified, and one method of illustrating these interactions is shown in Figure 4.6 [18]. The spatial organization of this diagram is based on the hierarchy of gene expression. Each row holds the genes that have common regulators. 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 to express. These

levels can define significant points in the operation of the genetic network, and often have a specific purpose within the organism, for example building a particular section of the organism [18]. Given the significance of these levels, one goal in creating the network structure visualization was to make them explicit.

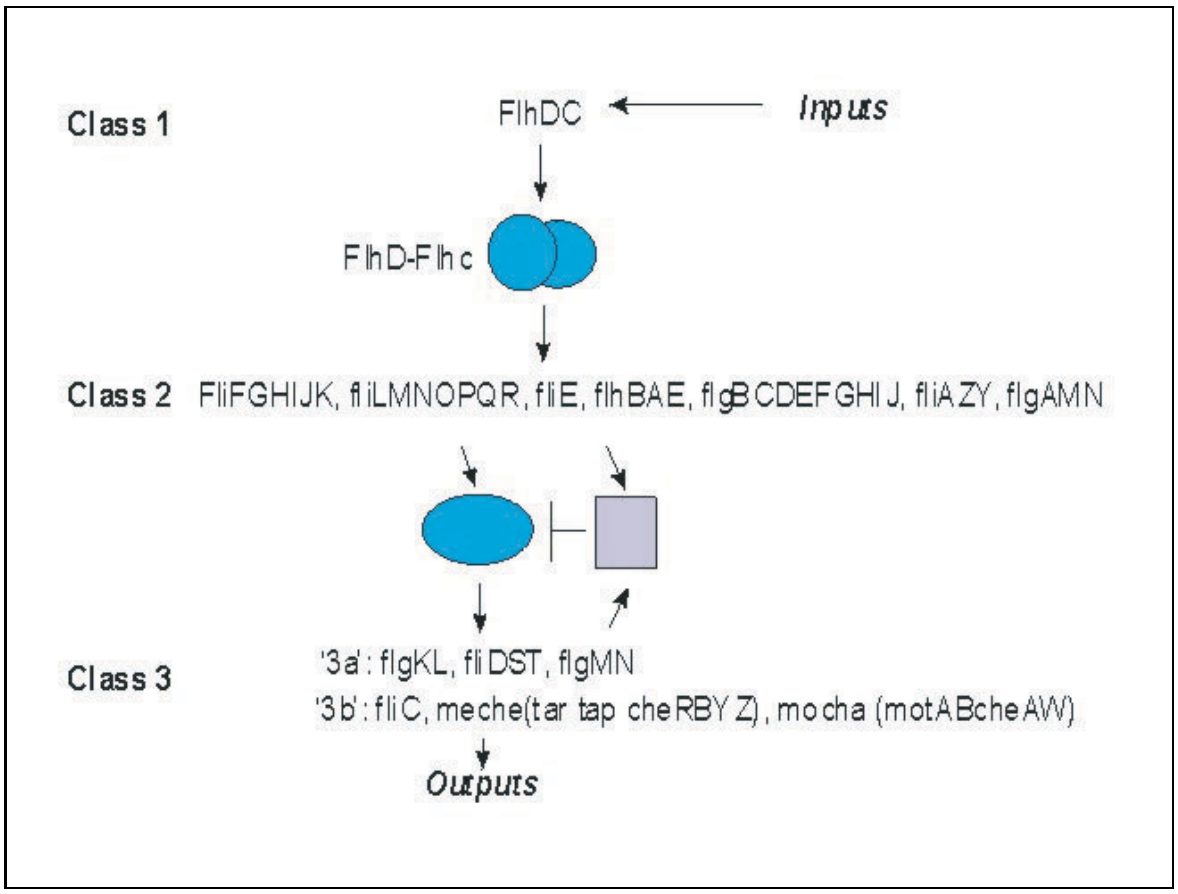

Figure 4.6: Gene network hierarchy of the flagella operons in *E. coli*. Genes are represented as character strings (e.g. flhDC), with lines in between representing the proteins that relate the genes. There are three levels of genes in this network (adapted from [18]).

In the structural visualization, the regulatory relationships to be represented include: forward promoting and inhibiting relationships, backward promoting and in-

hibiting relationships, and within-level and self-loops both promoting and inhibiting. The forward relationships are those in accord with the level structure of the network. The backward relationships or feedbacks occur when a gene's activity results in the production of a protein that regulates a gene located at an earlier level. The within-level relationships are those in which a gene's activity affects other genes in the same level. Self-loops are those relationships in which a gene produces a protein that regulates the expression of that gene. These different types of regulatory relationships frequently make the network non-planar, and their presence often interferes with the ease of displaying genetic networks using 2D graph layouts. Graph layouts can very quickly become hard to read when they include multiple edge-crossings [34].

To address the difficulties of displaying feedbacks, GeneVis presents the genetic network structure in 3D. The network is drawn with the nodes representing genes and the edges representing the relationships between genes. Each level of the hierarchy is transformed from a 2D row of Figure 4.6 to a 3D ring, and the genes within that level are distributed evenly around the ring (Figure 4.7). The rings are indicated by dashed lines to keep them visually distinct from the network connections.

Forward protein regulation connections are displayed as curved lines. Feedbacks are shown as straight lines. Within-level relationships are drawn around the ring. Self-loops are small loops starting and ending at the same gene. Colours are also used to indicate the direction and type of the relationship. The forward regulation line is blue at the producing end, the backward regulation line is magenta at the producing end, and the within-level line is yellow at the producing end. All lines with promoting connections fade to green at the receiving end, and to red if they inhibit the expression of the genes they control. Making the different types of regu-

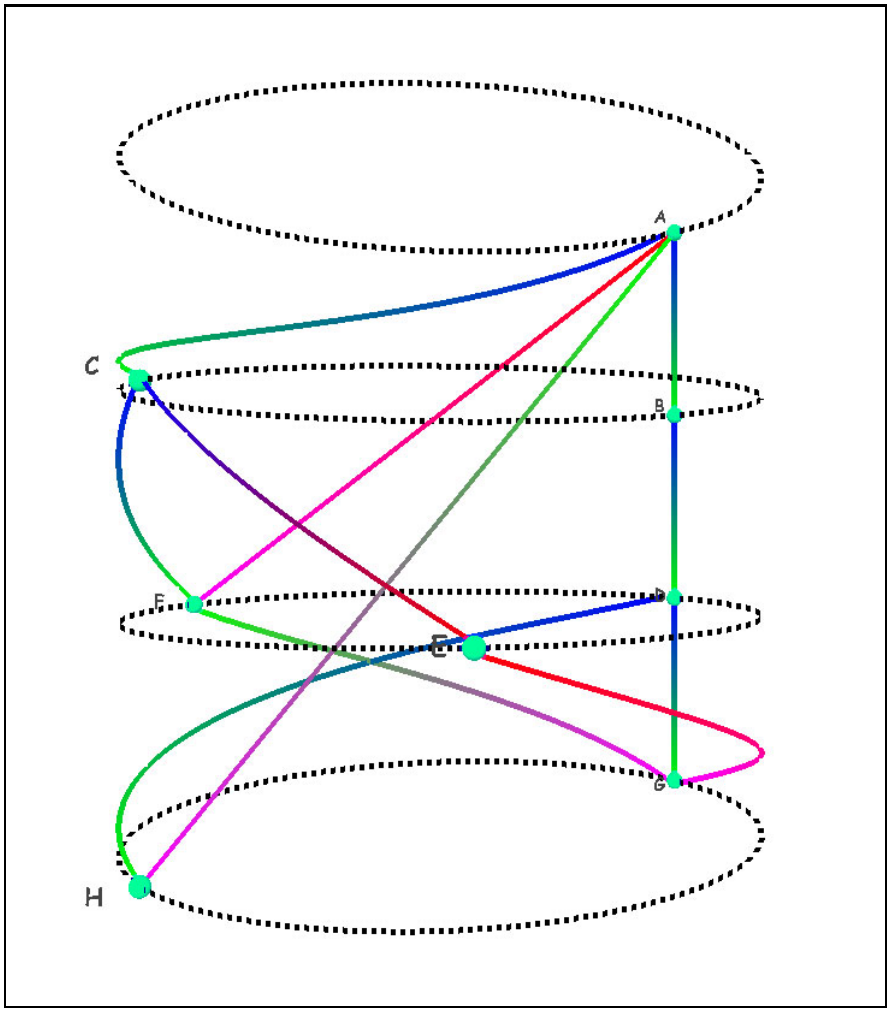

Figure 4.7: An example of the genetic network structure visualization: Each ring represents a level in the gene hierarchy. The genes (spheres) are related by lines representing regulatory proteins. Forward, backward, and within-level lines are drawn blue, magenta, and yellow respectively at the producing end. At the receiving end all promoting connections fade to green and all inhibiting fade to red.

lation visually distinct in both colour and shape alleviates some of the edge-crossing problems common to graph layouts. To take advantage of the 3D layout, the entire network can be rotated, giving the user different views of the network architecture.

4.2.1 Visual Integration of Network Structure

To visually integrate the simulation and the network structure, the transition between the two visualizations can be animated. This animation can be viewed at once or stepped through in either direction.

Figure 4.8 shows steps of this animation, moving from the simulation to the network structure visualization. The purpose of this animation is to allow a user to track a gene from its location in the simulation to its location in the network structure visualization. In the first step of the transition, the lines that represent the regulatory connections are drawn on the circular chromosome of the simulation visualization (Figure 4.8, first image). Next, each level is drawn inward, one by one, until the network is partitioned into levels (Figure 4.8, second image). At this point, the network is represented as a series of concentric rings in a 2D plane. The next stage of the transition (Figure 4.8, third image) moves the viewpoint, to give a side view. Then each ring is translated upwards, showing each level and its connections (Figure 4.8 fourth image). At the end of the animation, Figure 4.8, last image, shows all the rings enlarged to the same diameter and the forward connections changed to curves. Each transition takes place gradually to allow the user to track individual genes from one step to the next.

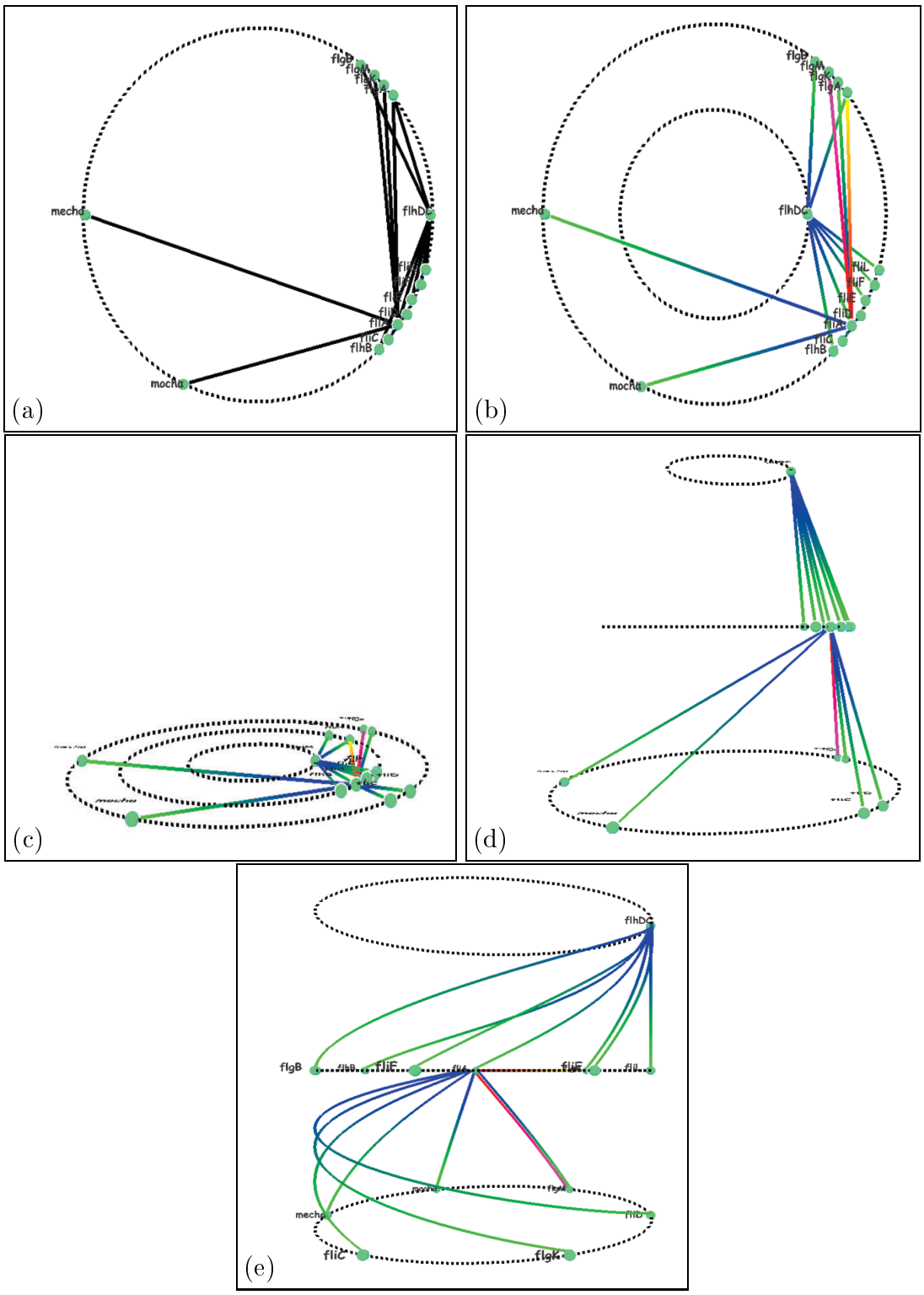

Figure 4.8: Visual integration that moves the user from the simulation visualization (a) to the network structure visualization (e).

4.2.2 Ring Lens

As the network size increases, the level rings become closely packed together. This congestion can make connections between the genes difficult to discern. The *Ring Lens* addresses this problem. It is a type of detail-in-context lens, which increases the space for the viewing of details in the selected region while maintaining the surrounding context. To this end, the Ring Lens enlarges the diameter of the selected rings and spreads them vertically. The new position and diameter of the ring is calculated as follows. First, a parameter called $PositionRatio$ is calculated for each ring according to this formula:

If $(Ring > LensCenter)$

$$PositionRatio = \frac{(Top - Ring)}{(Top - LensCenter)} \tag{4.1}$$

else

$$PositionRatio = \frac{(Ring - Bottom)}{(LensCenter - Bottom)} \tag{4.2}$$

Here $Ring$ is the vertical position of the ring to be adjusted and $LensCenter$ is the vertical position of the Ring Lens, Top is the position of the topmost ring and $Bottom$ is the position of the lowest ring. $PositionRatio$ is used to calculate the change in ring diameter:

$$scaleDiameter = (PositionRatio^2 * (MaxMag)) + 1.0 \tag{4.3}$$

Squaring the $PositionRatio$ makes the amount of magnification drop off more quickly. Adding 1.0 ensures that the ring's diameter does not diminish. $PositionRatio$ is also used to calculate the new vertical location of the ring. To this end, the parameter

$VerticalAdjust$ is calculated with the formula:

$$VerticalAdjust = \frac{PositionRatio^2 * (Top - Bottom)}{VerticalScaleFactor} \tag{4.4}$$

$VerticalAdjust$ is subtracted from Top if the ring is above the lens center, and added to $Bottom$ if it is below the lens center. Figure 4.9 is a diagram that shows how the ring lens works.

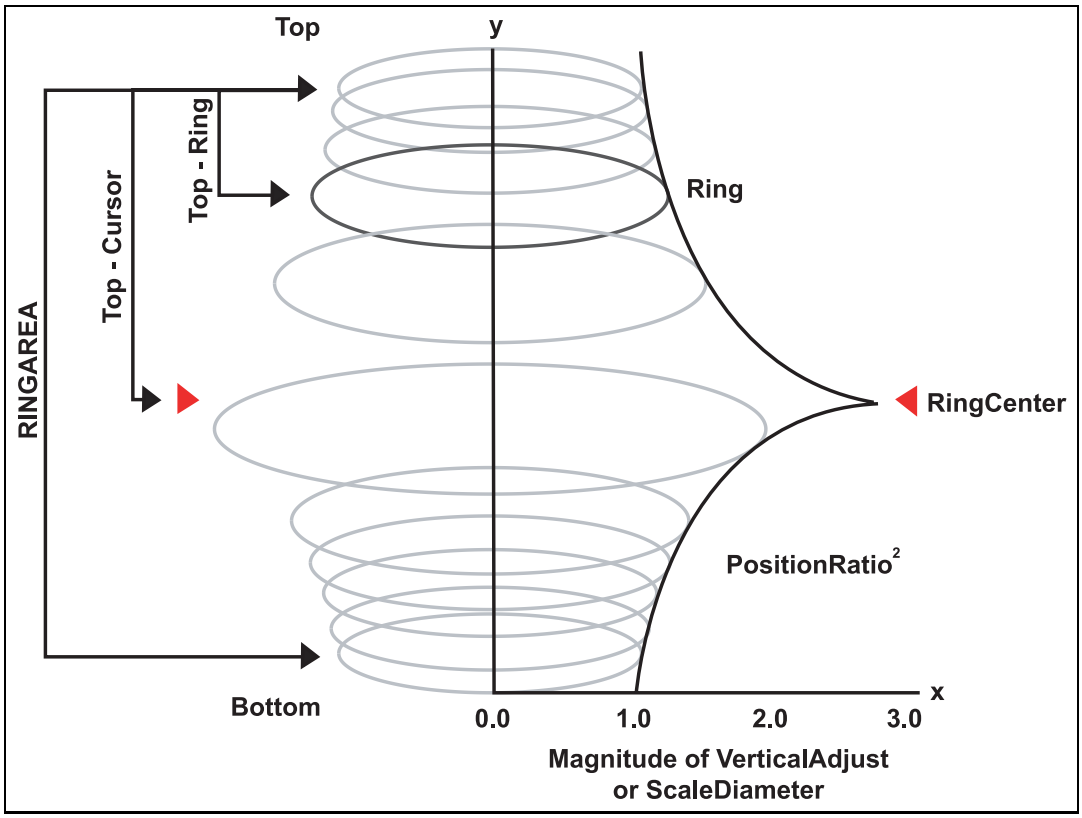

Figure 4.9: This diagram outlines the different calculations and variables that make up the distortion function for the *ring lens*.

The vertical position of the Ring Lens is controlled by the mouse. Figure 4.10 shows screen shots of the Ring Lens in different positions. The bottom image shows

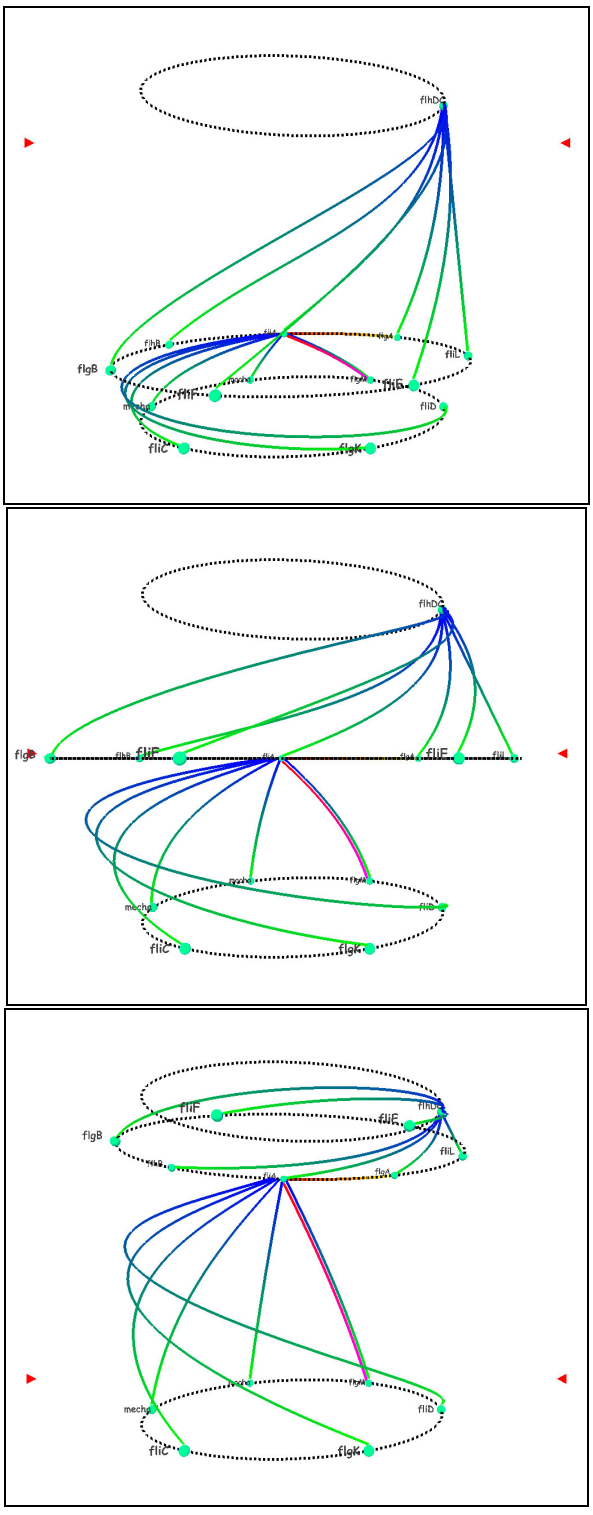

Figure 4.10: Ring lens view transformation. (Top) Ring lens cursor positioned at the top of the network, (Middle) Ring lens cursor positioned at the middle of the network, (Bottom) Ring lens cursor positioned at the bottom of the network.

the lens shifted towards the bottom of the view. This makes the connections between the lower two level rings more visible by increasing the amount of space between them. The middle image shows the Ring Lens placed at the second level ring, causing it to be enlarged in diameter. The top image shows the lens near the top of the view, opening up the space between the first two levels. The Ring Lens allows the user to interactively view the selected levels within the genetic network structure while maintaining the context of all the other rings.

4.3 Conclusions

GeneVis provides dynamic visualizations of simulated genetic network behaviour and a visualization of the network structure. This chapter describes three visual representations. The protein interaction representation shows the dynamics of the simulated network behaviour through the motion of individual proteins. The protein concentration representation depicts the concentrations of proteins during the simulation. The network structure representation shows the dependency structure of the genetic network using a 3D graph layout. This representation shows several types of regulatory relationships, including forward, backward, and self regulation. All of these can have either promoting or inhibiting effects.

GeneVis also provides several specialized viewing tools and techniques. These include:

- The continuous representational transformation between the protein interaction representation and the concentration representation.
- The three Fuzzy lenses, which allow one to view selected regions of the simu-

lation dynamics with the representation of choice.

- The base pair lens, which allows one to reposition the genes, thus separating and more evenly distributing closely clustered genes.
- The animated transition between the dynamic visualizations and the network structure visualization.
- The Ring Lens, which provides detail-in-context viewing for the network structure.

With these representations and tools, genetic regulation networks can be viewed and explored.

Chapter 5

Interacting with GeneVis

There are two methods of specifying a genetic network within GeneVis: by input file and by dialog box. In order to describe as many different prokaryotic genetic networks as possible, the input to GeneVis has been designed in a general format. Similarly, the simulation environment has a generalized design and care was taken to limit the degree to which the spatial organization of the simulation restricted the simulation results. The intention was to support flexible input and interactive adjustment. Both the simulation and the input methods were developed through iterative design in collabration with Dr. Michael G. Surette from the Department of Microbiology and Infectious Diseases.

This chapter describes the available ways of interacting with GeneVis. Section 5.1 outlines how to use an input file to specify a genetic network for GeneVis. An explanation of the development of spatial organization in GeneVis is contained in Section 5.2. Section 5.3 discusses how the simulation behaviour is affected by different input parameters. Section 5.4 contains a brief summation.

5.1 Genetic Network Specification

5.1.1 Input

The schema of the input file that is used in GeneVis is designed to allow for many different prokaryotic genetic networks to be described. It has general fields that are

applicable to most prokaryotic genetic networks. The input file was structured in a simple format developed in collaboration with Dr. Michael Surette's laboratory in the Department of Microbiology and Infectious Diseases at the University of Calgary. This simple format allows anyone to explore genetic network behaviour without the prerequisite for significant computer or molecular biology knowledge. Figure 5.1 shows the contents of a sample input file.

```
EnvironmentSize
100

BasePairRange
1000000 2100000

Genes
ID      Name BasePairPosition   Initator ProducedProteinID      BasalActivity   ExpressionRate
  1     flhDC           1000500 1               1               1               10
  2     fliL            2046264 0               0               1               10
  3     fliE            2015586 0               0               1               10
  4     fliF            2030530 0               0               1               10
  5     flgB            1162691 0               0               1               10
  6     flhB            1960331 0               0               1               10
  7     fliA            1987757 0               2               1               10
  8     flgA            1118691 0               3               1               10
  9     fliD            2001530 0               0               1               10
  10    flgK            1134865 0               0               1               10
  11    fliC            1974397 0               0               1               10
  12    mecha           1550000 0               0               1               10
  13    mocha           1750000 0               0               1               10
  14    flgM            1148322 0               3               1               10
```

Continued below

Continued from above

```
        (Can have multiple Operator sites)
Decay   ActivatorID     ActivatorFactor InhibitorID     InhibitorFactor AffinityFunc
2               0       12.0000         0               -2.5000         0.00
2               1       12.0000         0               -2.5000         0.85
2               1       12.0000         0               -2.5000         1.00
2               1       12.0000         0               -2.5000         0.92
2               1       12.0000         0               -2.5000         0.85
2               1       12.0000         0               -2.5000         0.62
2               1       12.0000         3               -12.5000        0.62
2               1       12.0000         0               -2.5000         0.85
2               2       12.0000         0               -2.5000         0.85
2               2       12.0000         0               -2.5000         0.95
2               2       12.0000         0               -2.5000         0.85
2               2       12.0000         0               -2.5000         0.85
2               2       12.0000         0               -2.5000         0.85
2               2       12.0000         0               -2.5000         0.77
```

Figure 5.1: This is the input file, which specifies the data for a genetic network in GeneVis. The EnvironmentSize specifies the size of the grid to be used, the BasePairRange specifies the range to be used around the circular chromosome representation. Under the heading of Genes each row specifies one gene that will be used in this network and its parameters.

The first item in the input file is the specification of the EnvironmentSize which is

used to set the resolution of the grid. This grid is used to control protein movement and its size affects the speed of the simulation (see Chapter 3 section 3.2.3). Next, the BasePairRange specifies the range of base pairs. This is used around the circular chromosome. The input file then has a specification for each gene which includes:

1. Unique Gene Identifier (ID)
2. Name of the gene (Name)
3. Base pair position of the gene (BasePairPosition)
4. A flag to indicate if the gene is an initiator (Initiator)
5. An identifier for the protein produced from transcription (ProdProteinID)
6. The basal activity rate of the gene (BasalActivity)
7. The expression rate of the gene (ExpressionRate)
8. The expressed protein's decay rate (Decay)

A gene can have several operator sites. Figure 5.1 shows a genetic regulatory network file in which all genes have one operator site. All operator sites are listed in the input file on the same row consecutively. Each operator site has the following specification:

1. The identifier of the protein that promotes the expression of the gene (ActivatorID)
2. The degree to which the promoter protein activates the gene's expression (ActivatorFactor)

3. The identifier of the protein that inhibits the expression of the gene (InhibitorID)

4. The degree to which the inhibitor protein suppresses the gene's expression (InhibitorFactor)

5. The affinity for the binding of a protein to an operator site (AffinityFunc)

Each one of these parameters is set for every gene. If a parameter is not applicable to the gene then that field is marked with a 0 (for example Gene 2 has no inhibitor protein as indicated by the 0 in that field). From these input parameters for each gene a simulation can be run.

GeneVis begins its simulations from a blank state. That is, there are no proteins in the simulation environment at the initial time point. This is not the case in an actual cell. A cell is always populated with proteins and other chemicals at any point in time. GeneVis starts in this blank state because this state is useful for debugging network dynamics. From a blank state it is much easier to see the interaction of proteins and genes as they start to fill the cell. However, the cell becomes populated from a blank state after a relatively short period of time. For example, it takes less than three minutes for the simulation environment to become populated when using the network from Figure 5.1 on an AMD Athlon 1.3GHz computer with 512MB RAM. To assist in moving GeneVis to a populated state there is an option of specifying initiator genes that will act as if a promoter protein is bound at onset of the simulation. This was done to simulate an external input source such as another genetic network that affects the current genetic network. The initiator represents the protein from another network that activates the gene in the

current network. Initiator genes are usually genes which are central to the cascading sequence of gene-protein interaction in the network. The effect of an initiator can also be created by setting a high basal activity for a gene. If the network being studied does not have external inputs from other genes then the basal activities of every gene are relied upon to bring the genetic network to a populated state.

A future goal is to adapt GeneVis so that it will begin in an approximate populated state.

5.1.2 Adjustments

All of the parameters in the input file are used to specify the genetic network. While the input files are good for the initialization and storage of genetic network topologies it is also possible to interactively modify these parameters within GeneVis. A dialog box is used to allow editing of the parameters. These changes are applied immediately, affecting how the genetic network functions during simulation. Any changes during the simulation can be considered as gene or protein mutations (see Appendix A).

On the top half of the dialog box in Figure 5.2 the user can change the gene's name, base pair position, produced protein, decay, and protein color. In the bottom half of the dialog box the gene's basal activity, expression rate, and the properties of each operator site can be edited. Each gene can have any number of operator sites. Each operator site has the properties of affinity, required activator protein, required inhibitor protein, activator factor, and inhibitor factor. Any change made to a property has an immediate effect in the simulation. These adjustments can be used to build genetic networks and change parameters that affect the network

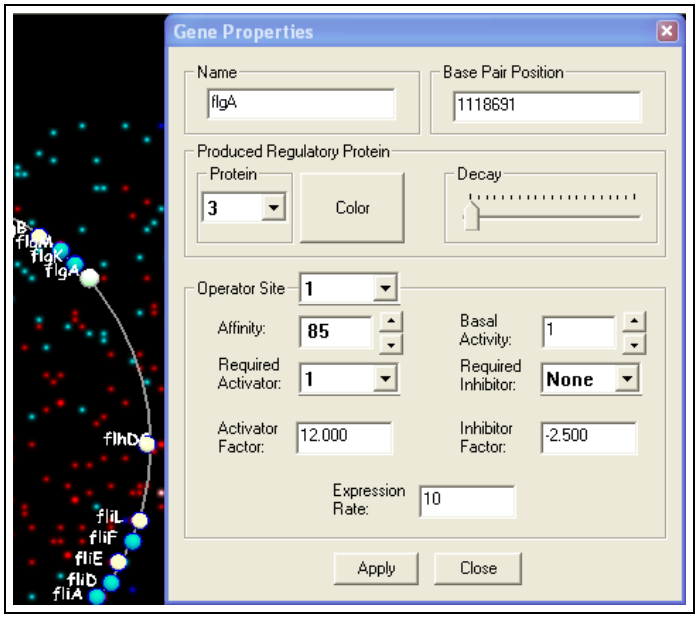

Figure 5.2: The gene properties dialog box. In the top part of the dialog, the gene's name, base pair position, produced protein, decay, and protein color can be set. In the bottom part of the dialog box, the gene's basal activity, expression rate, and the properties of each operator site can be edited. The gene can have N operator sites each of which has the properties of affinity, required activator protein, required inhibitor protein, activator factor, and inhibitor factor.

dynamics. Any changes can be saved in the input file for reuse in later simulations.

5.2 Spatial Organization

Through specifying network parameters, either by the input file or the dialog box, the simulation behaviour can be affected. As discussed in Section 5.1, the input files are generalized to make it possible to simulate many networks. Similarily, the simulation environment was developed to avoid restricting the effect of the input parameters on the simulation results.

The initial spatial organization of the simulation environment was found to be a serious constraint. Originally the system was designed in a tree like structure to both represent the structure of the genetic network and simulate its behaviour at the same time. In this first version it was found that the coupling of these two goals hindered the simulation behaviour making it deterministic.

Figure 5.3 shows the first version of GeneVis where both the structure and simulation are integrated. This simulation is based on multiple particle systems that project the particles that represented proteins towards a set of receiving genes. Genes would be activated by particles intersecting with their outer surface, which would subsequently cause the next gene to express. This enforced the structure of the layout on the genetic network. This structure made the simulation results deterministic and did not allow proteins to come to concentration levels within a general space where they could then probabilistically affect gene expression. Such restrictions imposed by the structure are not present within actual cells and therefore were seen as a disadvantage. The effects of spatial organization needed to be minimized.

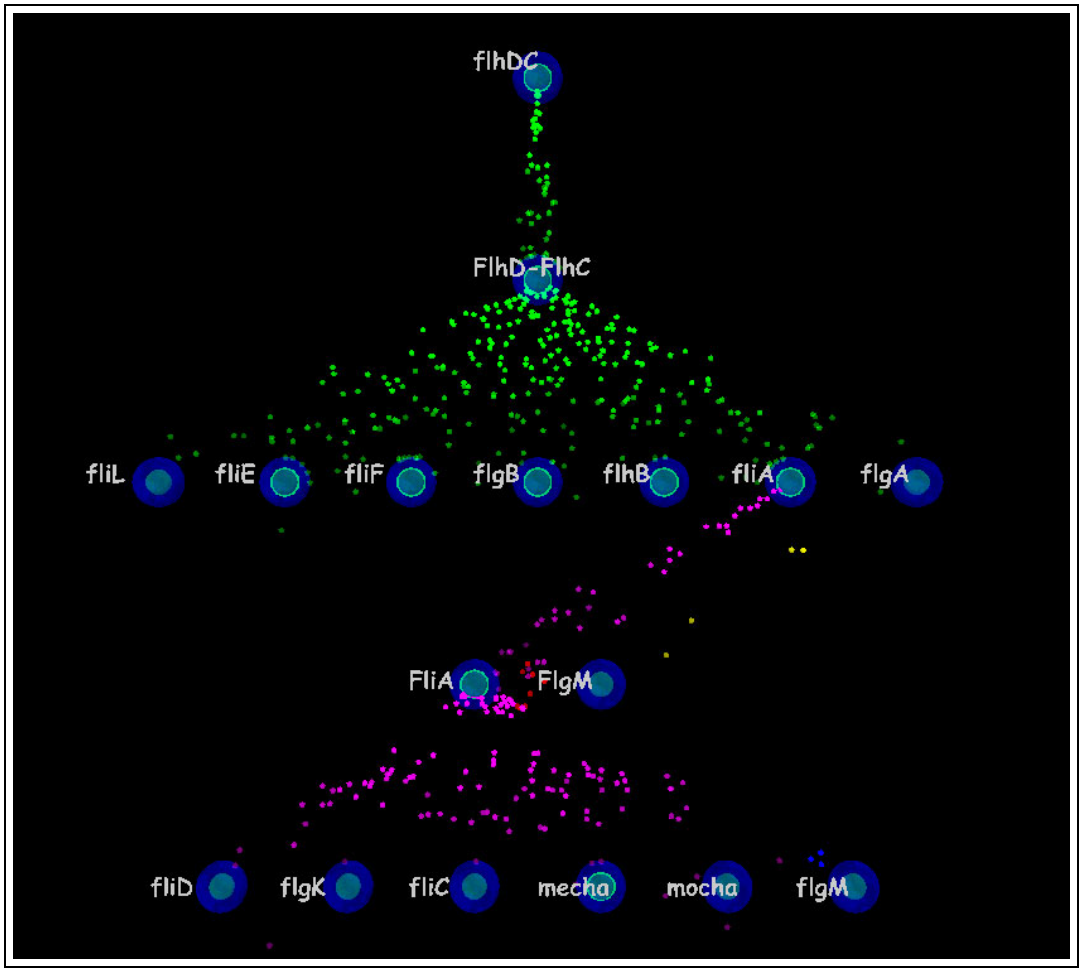

Figure 5.3: This is a screenshot of one of the first versions of GeneVis where the simulation and the structure of the genetic network are coupled. This produced an artificial spatial restriction causing the simulation to be deterministic.

5.2.1 Spatial Reorganization

It was decided that the simulation environment should not impose the structure of the genetic network by its organization. Instead of using the assumed network structure, genes are located according to their base pair position around a circle representing the prokaryotic chromosome. This chromosome is centered in a 2D grid. Within this 2D grid proteins move randomly in any of the eight possible directions and at randomized distances (see Chapter 3). All edges of the grid are wrapped to further loosen restrictions on movement. This spatial organization has reduced the restriction its predecessor had imposed. One observation was the disappearance of strict localized effects. That is, genes did not all turn on at the same time but rather were promoted at variable rates as proteins came in contact with the genes. While the reorganization did remove some spatial effects, it did create some additional ones, in that the order of activation was usually dependent on the distance of the receiving gene from the producing gene. To address this the step length was randomized, which reduced the effects of spatial location. Additional research into the current organization may reveal further possibilities for reducing spatial effects.

5.3 Simulation Behaviour

This section discusses the behaviour resulting from GeneVis simulations that use different input parameters. In terms of comparing outputs with an independent data set, the following is not a validation of GeneVis [37]. While validation is a future aspiration, preliminary work in this direction has shown that additions are needed to make GeneVis more accurately model genetic regulation.

The following section shows that the model works as it was designed. The goal was to take a first step towards visualizing the processes in a manner that could be presented interactively. There are three specific behavioural results that were observed:

1. Gene-Protein Interaction: proteins interact with genes according to the rule sets. For instance, when a protein binds to a gene based on its rule set that gene's expression should change in a manner that reflects that gene's rule set.

2. Steady States: since each gene has an expression rate declared in its rule set and its produced protein has a declared decay rate, produced protein concentration should be able to be maintained at different concentration levels depending on the given gene expression rate and protein decay rate. Steady states are observed as the relatively constant level of protein existing within the cell over time which is caused by the protein production rate and protein decay rate being approximately equal.

3. Stochastic Variability: the proteins exhibit random variability in their concentration but within a range. For example, for a given steady state of one type of protein, there is a degree of variation over time caused by the five separate randomized genetic interaction mechanisms.

All of the graphs used to illustrate the discussion that follows depict expression results created by the GeneVis simulation model. GeneVis allows the output of simulated expression histories to a file in a format that can be imported into Excel for plotting. The following graphs were created using Excel. Each graph has a

legend on the right hand side which shows a series of letters that each represent a gene. These letters correspond to the name that was declared for this gene in the input file. In each graph the y axis is protein level and the x axis is time. The protein level is the number of proteins in the environment at a given time point. A time point in these graphs represents one step of the simulation algorithm.

5.3.1 Gene-Protein Interaction

The sequence of gene-protein interaction specified within the input is expected to be reflected in the simulation's behaviour. Given an initial blank state these interactions will show up as a specific cascading sequence. A cascading sequence of gene expression appears as genes increase in expression one after the other. This occurs because each gene is dependent on the one before it in the sequence.

Figure 5.4 shows the resulting gene expression (bottom) from a GeneVis simulation that was run using the sample network (top). As the simulation begins every gene starts expressing at its basal activity level. This causes proteins to be introduced into the environment. As the proteins disperse throughout the environment they come in contact with other requiring genes. If one protein binds to a requiring gene that gene will subsequently begin expressing proteins at its promoter rate. These proteins can then interact with other requiring genes and cause their subsequent expression. In Figure 5.4 the graph shows a cascading sequence of expression where each gene is activated at later time points.

Cascading sequences can have cycles within their network topology. Cycles can cause much more complex expression sequences to occur. These cycles are commonly known as feedback loops. Feedback loops typically create reoccurring patterns in

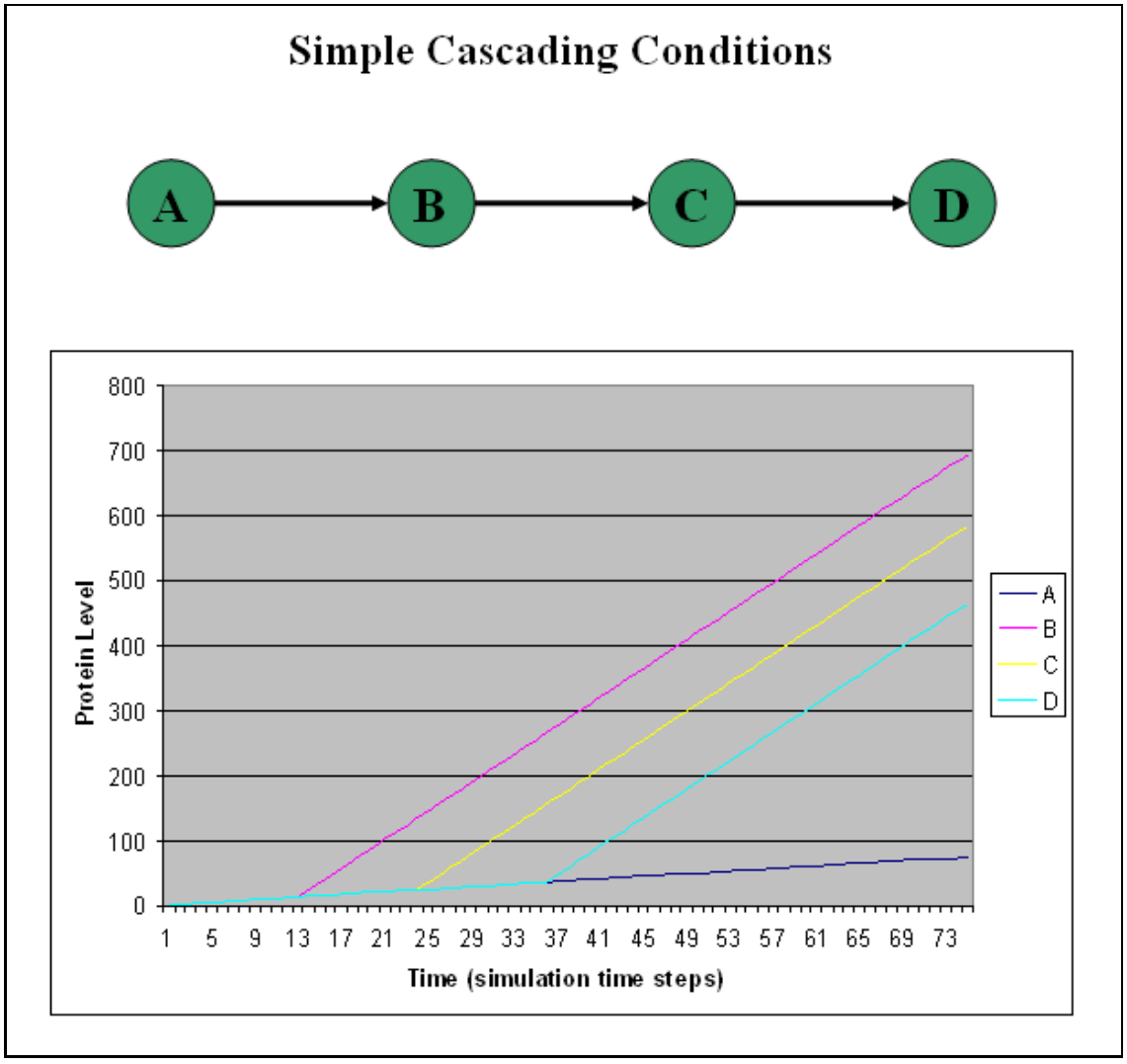

Figure 5.4: This is the gene expression resulting from a GeneVis simulation using the sample network shown at the top of this diagram. Every gene has the same basal activity and as a result expresses equal numbers of proteins in the beginning of the simulation. Then at time steps 13, 25, and 37 there are spikes in the protein level for each gene. These spikes represent the time points when the gene has been promoted.

expression.

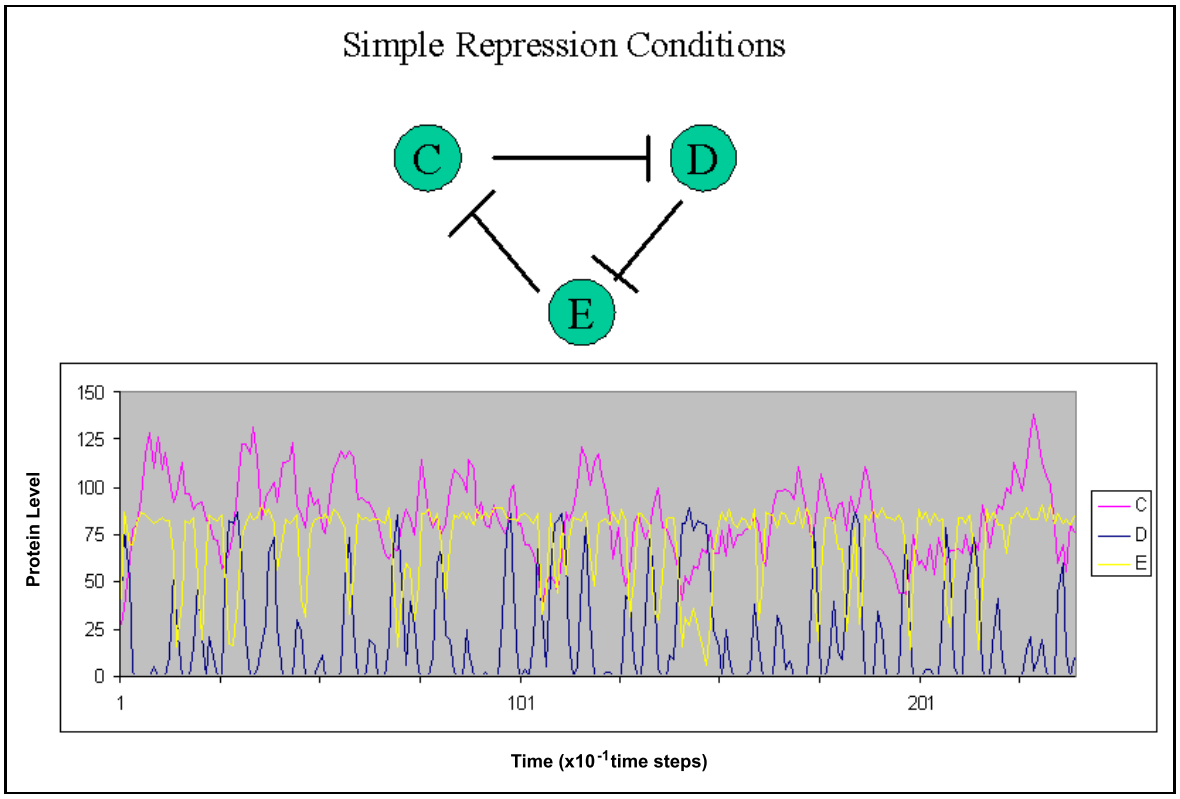

Figure 5.5: The diagram in the top shows the topology of this repression network. Inhibition is indicated through a line with a perpendicular cross at the end of it. The network above produced the results in the graph below. For each gene there is an oscillating pattern in its expression profile. This is caused by each gene being dependent on the previous gene.

Figure 5.5 (top section) shows a simple repression network where gene C inhibits gene D which inhibits E which in turn inhibits C. The loop which exists in this network (the network cycles through C, D, and E) should cause oscillating peaks in expression. The graph shows large variable jumps in gene expression since each gene's expression is dependent on the other.

5.3.2 Steady States

Steady states are levels of gene expression that remain relatively constant over time. These levels are maintained by a gene producing its protein at one rate and this protein decaying at another rate.

Figure 5.6 depicts gene expression results from GeneVis showing ten genes which are all maintaining different steady states. As expected, it was found that changing only the basal activity and leaving decay rate fixed, yielded different steady state levels of expression.

Figure 5.7 shows expression results from GeneVis that display similar steady state levels as in the previous Figure 5.6, but the initial protein concentrations now vary widely. This occurs by varying not just the basal activity but also the gene expression rate and protein decay. By varying basal activity, gene expression rate and protein decay, the initial protein concentration levels are more variable but still stabilize after about 50 time steps.

5.3.3 Stochastic Variability

The randomization of five separate factors (protein decay, direction of protein travel, distance of protein travel, operator site affinity, reversible binding) should cause random fluctuation in the number of proteins in the system within a range. At any steady state the protein concentration should not remain constant but fluctuate to some degree.

Figure 5.8 shows a bar graph with the variance in protein level produced from a GeneVis simulation. Each bar represents the average protein level. The maximum expression and minimum expression levels are also shown. This graph shows that

Varying Basal Promoter Activity at Fixed Expression and Decay Rates

Gene	Basal Activity	Expression Rate	Decay
A	1	8	5
B	2	8	5
C	3	8	5
D	4	8	5
E	5	8	5
F	6	8	5
G	8	8	5
H	10	8	5
I	15	8	5
J	20	8	5

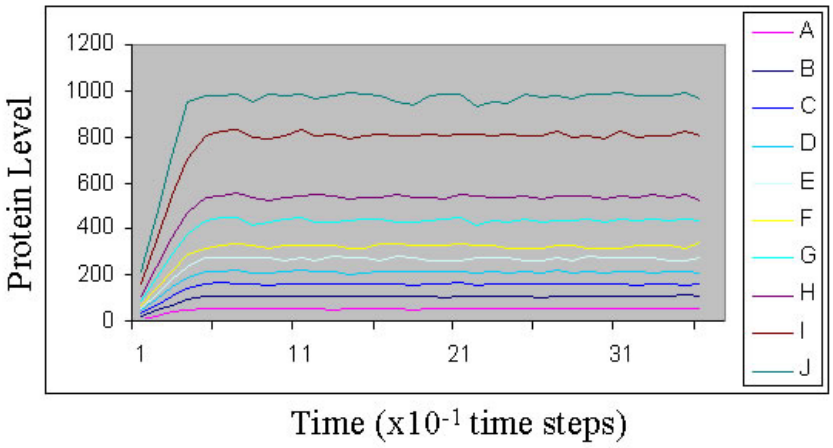

Figure 5.6: The table at the top shows the parameters used within GeneVis to produce the resulting graph on the bottom. The graph shows the effects of varying basal activity with fixed expression and decay rates (see Appendix A). Notice each gene (A through J) levels off at a different protein level based on the difference in basal activity.

Similar Steady Levels Using Different Expression Parameters Have Different Initial Expression Results

Gene	Basal Activity	Expression Rate	Decay
A	4	8	6
B	8	8	3
C	8	22	8
D	2	4	6
E	5	8	5
F	8	8	3
G	8	18	8
H	3	3	3
I	3	2	2
J	4	6	4

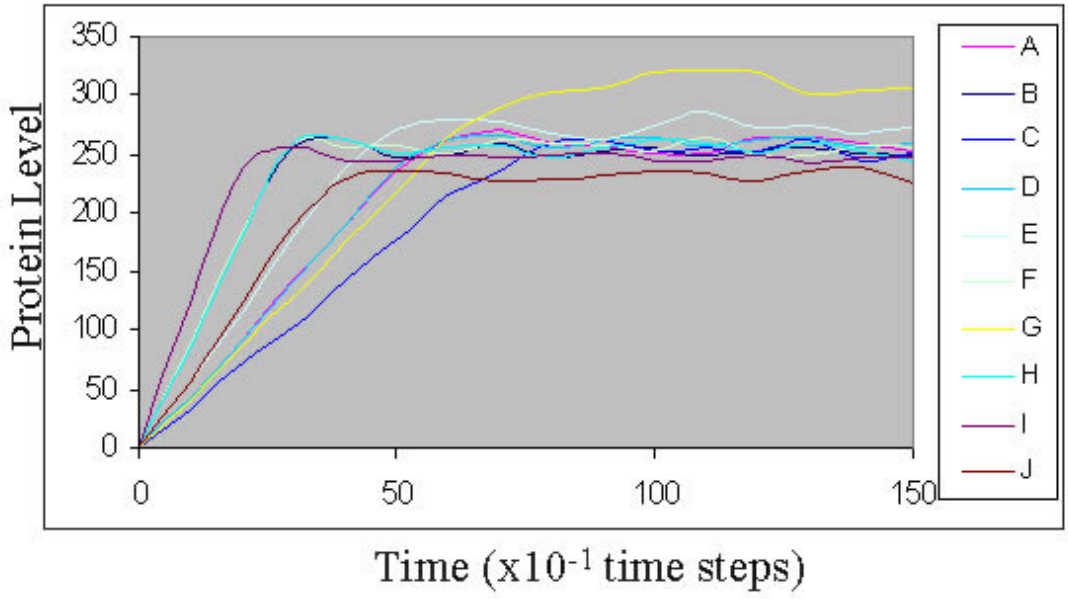

Figure 5.7: The table of the top shows the parameters used within GeneVis to produce the resulting graph on the bottom. The graph shows the effects of varying basal activity, expression rate and decay rates. The initial period of protein productions (time 0 through 50) vary widely on slope and stability.

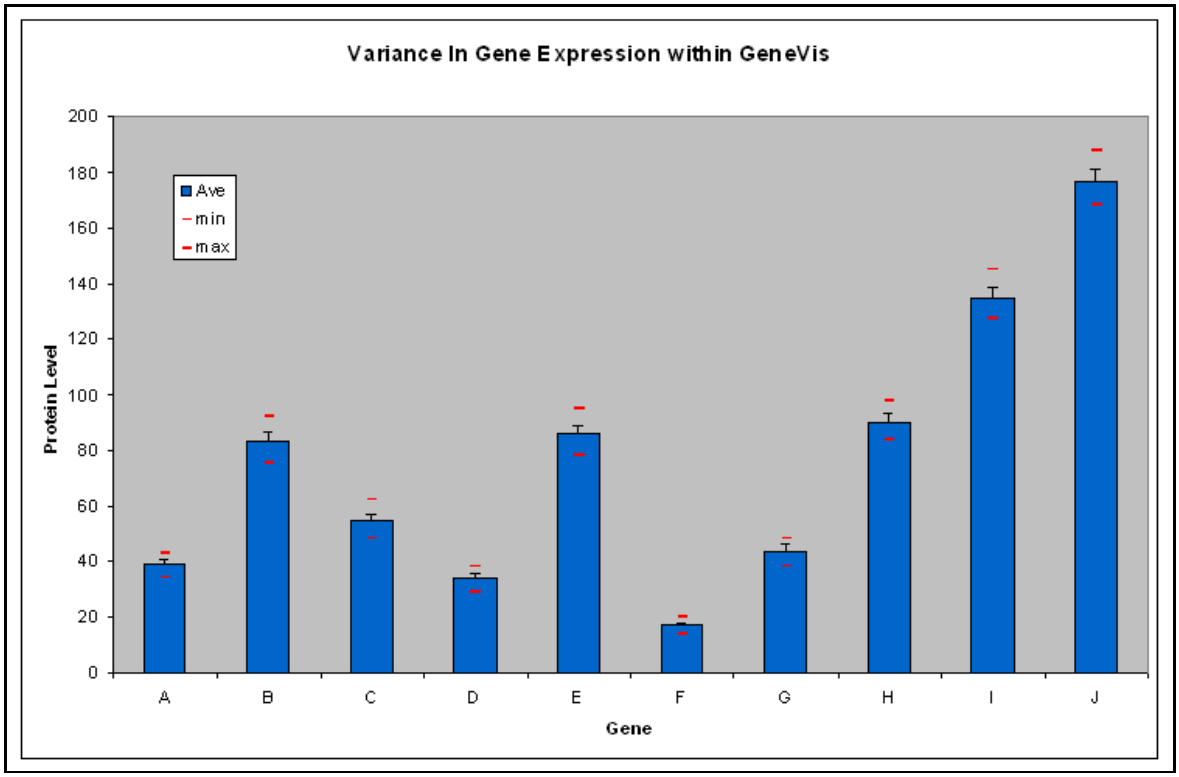

Figure 5.8: The simulation results have been plotted to show the protein level on the *y-axis* with each gene on the *x-axis*. The bar graph indicates the average protein level and the error bars show the minimum, and the maximum protein levels after steady state has been reached.

genes operate within a range of expression values even when this range is not explicitly set within the program, but rather the range of expression values are a result of the combination of the parameters and the simulation dynamics.

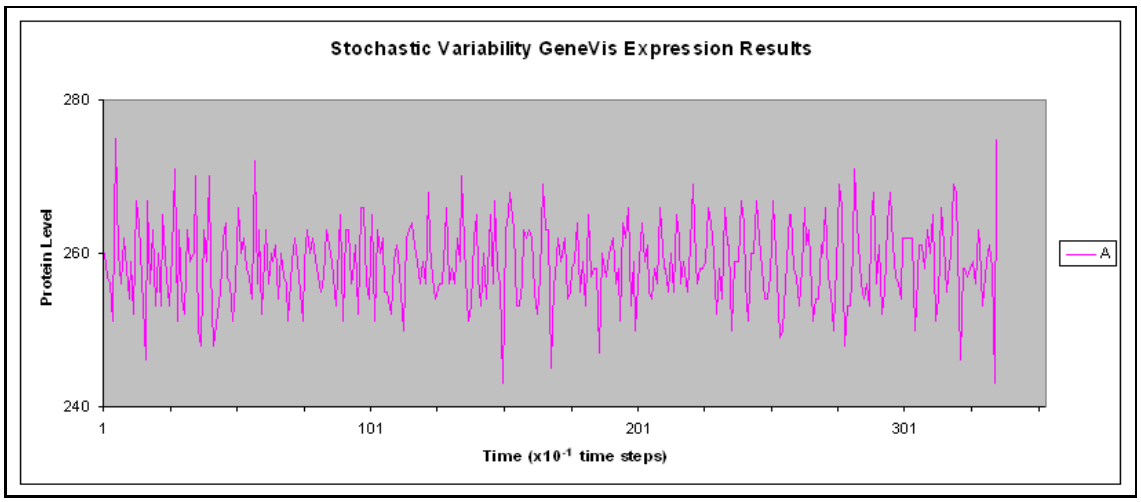

Figure 5.9: This graph shows the significant variation in the steady state levels over time with the range on the *y-axis* of 240 to 280 proteins. As time proceeds in the simulation stochastic variability within the gene's expression is noticeable.

The actual variance can be seen in the magnified portion of a single gene's expression results shown in Figure 5.9. Here the variability in protein level can be seen as ranging from 240 to 280 proteins, with a prevalent steady state of 260 proteins.

Figure 5.10 shows the amount of noise in varying steady state levels created by the GeneVis simulation. This noise measurement shows an inverse relationship between the amount of noise and the steady state level. As the steady state level decreases the noise in the expression increases. Noise is computed as standard deviation divided by average steady state where steady state levels are taken as averages. This is significant as the inverse relationship between noise and steady state level is common

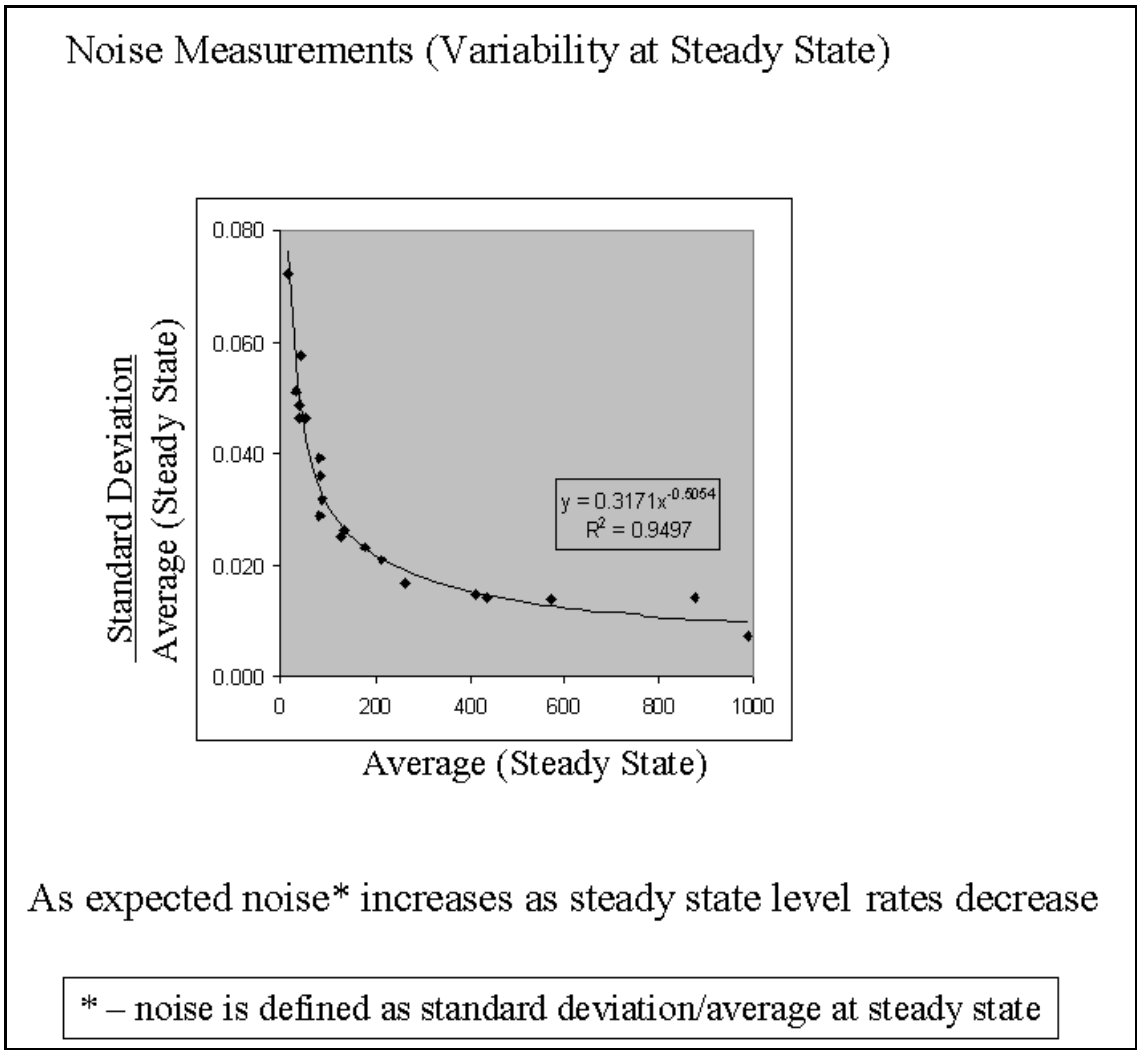

Figure 5.10: This graph shows the noise of expression within the GeneVis simulation by relating standard deviation divided by the average at steady state. As expected, the noise increases as the steady state level decreases.

in experimental results.

5.4 Discussion

In collaboration with Dr. Surette, GeneVis moved from a deterministic system to a much more generalized system for describing genetic networks. Parameters that affect the simulation can be specified in a general input file and adjusted during the simulation. By altering those parameters in a systematic manner, expression results can be changed in a predictable way. For example the following behaviours can occur within GenVis: probabilistic gene-protein interactions, steady states and stochastic variability in protein concentrations. However, through experimenting with GeneVis, additional genetic mechanisms, such as cooperativity between binding sites, have been identified as needed.

Chapter 6

Conclusions

This thesis presents GeneVis, which implements a conceptual model of genetic network interaction. In this model, genetic networks are considered to be sets of genes that are regulated by sets of proteins. When genes in the network express, they trigger the production of proteins, which in turn can regulate the expression of other genes, thus creating a network of dependence. This thesis is limited to the domain of prokaryotes, which are simple organisms with only one looped chromosome. GeneVis implements this conceptual model as an adjustable simulation with dynamic visual representations and a static representation of the genetic network structure. The simulation is visualized with a number of different techniques and integrated with the genetic network structure visualization. The simulation, with its dynamic visualizations, and network structure visualization are interactive and allow changes to be made to genetic network parameters.

6.1 Contributions

This thesis makes the following contributions in the area of simulating and visualizing genetic network interaction:

6.1.1 Simulation Model

The simulation model in GeneVis is based on probabilistic interactions in which five individual parts of the simulation are randomized to create a probabilistic simulation. The five individual parts are:

1. *Protein direction of movement* - each protein can move in one of eight directions within the cell and this direction is randomly chosen at each time step. This movement allows for proteins to move through the environment and interact with requiring genes.

2. *Distance of protein movement* - each protein can move a number of grid cells within the environment and this distance is chosen randomly between a distance of one grid cell to the user-specified maximum distance. The distance parameter allows for the user to adjust the rate at which proteins spread throughout the environment.

3. *Protein life span* - each protein has a life span, which is randomized in a user specified range and the life span is reduced at each time step to simulate the decay of the protein.

4. *Operator site binding* - each operator site has an affinity associated with it. The affinity is a percentage chance which describes on average how often a protein will bind to a requiring operator site.

5. *Reversible Binding* - each protein can unbind from an operator site once it has been bound. Proteins have a user assigned percentage chance of unbinding during each time step.

6.1.2 Data

GeneVis reads in data files, which have been designed and formatted to be consistent with the input and output of tools used in Dr. Surette's laboratory. Dialog boxes provide interactive editing of these files and the changes are reflected immediately in the simulation and visualization. This interactive editing provides the ability to make changes at any time during the simulation. The user can export simulation results to data files that can then be plotted in programs like Microsoft Excel to create line graphs and bar graphs of the simulation data.

6.1.3 Dynamic Visualizations

The program provides multiple visualizations of simulated genetic network behaviour. The simulation is visualized in a 2D environment showing both the genes and the proteins involved in the simulation. The simulation is dynamically visualized, showing the proteins as they spread throughout the cell and interact with genes subsequently affecting their expression. There are two different methods of viewing the simulation:

1. *Protein Interaction View*: The *protein interaction* view shows each protein as a single entity randomly moving throughout the cell and binding or not to operator sites. This view can be used for editing purposes, as the user can see exactly where proteins are spreading and to which genes they are bound.

2. *Protein Concentration View*: The *protein concentration* view shows proteins as concentrations rather than as individual proteins. This view can be used to identify when proteins have spread throughout the entire simulation environment.

During the simulation, a plot of each gene's produced protein over time is recorded and displayed with the simulation. The data is recorded and visualized as gene expression histories [1, 11, 30] in a format commonly used by biologists. Laboratory expression results can be loaded into GeneVis and visually contrasted to the simulation's expression results, allowing for direct comparison between simulated and laboratory data.

6.1.4 Static Visualization

Each gene is regulated by a protein that is expressed by another gene. This dependence forms a network of interactions between genes. This is referred to as the *network structure* and is depicted within GeneVis to reveal the existing interactions between genes. This data is static as it only contains the relationships between genes and does not simulate protein expression data over time. The network structure is depicted in 3D and makes different forms of gene regulation visually explicit. The resulting visualization can be viewed interactively to reveal different aspects of the network's structure. GeneVis reflects any alteration made to the parameters of the simulation in the structure visualization.

6.1.5 View Transformation Tools

A number of *view transformation tools* provide different viewing methods for both the simulation and the structure model. A tool called *representational transformation* provides a method of gradually switching between the individual protein view and the protein concentration view. Also, tools called *fuzzy lens* provide the same type of viewing but in specified regions of the simulation visualization. The visual

representation of the network structure can be manipulated with a *ring lens*, which supports detail-in-context viewing. The representations (dynamic visualizations and structure visualization) are visually integrated by providing a coherent step-through animation that allows the user to switch back and forth between the different visual representations.

6.2 Future Directions

There are many possible future directions for this research. Some possibilities include:

6.2.1 Simulation

The simulation could be improved to model a greater scope of genetic interactions by:

1. Adding more genetic interaction mechanisms that may allow for more realistic simulations.

2. Including complex binding rule sets that may provide ways of making cooperativity between binding sites. This is needed for some specific organisms (i.e. Phage Lambda).

3. Providing for protein complex formation that may also allow for more organisms to be modelled (e.g. Lac Operon)

4. Starting the simulation from a populated state rather than having no proteins in the environment initially (blank state).

5. Linking the affinity parameter to each protein that binds to a gene's operator site. Currently the affinity applies only to operator sites rather than to the proteins that bind to that operator site.

6. Separating what is currently a single step gene expression into the more realistic steps of transcription. It may be possible to simulate each step of transcription to make the simulation more accurate.

6.2.2 Network Structure

In the network structure visualization color is used to indicate both regulation type and direction of regulation. It is possible that this could be improved by indicating the direction of regulation with arrows instead of color.

The network structure visualization still suffers from edge-congestion problems when the size and complexity of the loaded networks increase. A detail-in-context approach could be taken further. For instance, transparency might be used to fade surrounding items and brighten selected details. Filtering is another option for reducing the visual clutter of edge-crossings.

Another possible addition is to develop a viewing paradigm for displaying extremely large networks. This was not addressed with the network structure visualization and will need to be addressed as the network size increases to hundreds and even thousands of genes.

6.2.3 Transformational Methodologies

I would like to investigate generalizing the idea of representational transformations. This would involve investigating possible frameworks, which may exist, to create a

transformational methodology. Specifically I am interested in looking at ways of visually transforming between different mathematical representation models of the same phenomenon.

6.2.4 User Study

A user study conducted in conjunction with a laboratory is another future direction of interest. During collaboration with Dr. Surette's laboratory I received such comments as, GeneVis could be used as an educational tool as it shows the conceptual workings of genetic networks. A user study could provide valuable feedback both about the simulation and the visualizations produced by GeneVis that could help indicate directions for further development.

Bibliography

[1] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. *Ninth Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 695–702, 1998.

[2] M.D. Apperley, I. Tzavaras, and R. Spence. A bifocal display technique for data presentation. In *European Computer Graphics Conference and Exhibition*, pages 27–43. North-Holland, 1982.

[3] C.A.H. Baker, M.S.T. Carpendale, P. Prusinkiewicz, and M.G. Surette. Genevis: Visualization tools for genetic regulatory network dynamics. In *IEEE Visualization 2002*, pages 243 – 250. IEEE Press, 2002.

[4] C.A.H. Baker, M.S.T. Carpendale, and M.G. Surette. Simulation and visualization of genetic regulatory networks. Technical Report 2001-690-13, University of Calgary, 2001.

[5] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry: Theory and Applications*, 4:235–282, 1994.

[6] E.A. Bier, M.C. Stone, K. Fishkin, W.A.S. Buxton, and T. Baudel. A taxonomy of see-through tools. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, pages 358–364. ACM Press, 1994.

[7] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, 1999.

[8] M.S.T. Carpendale. *A Framework For Elastic Presentation Space*. PhD thesis, Simon Fraser University, March 1999.

[9] M.S.T. Carpendale and C. Jirasek. A framework for unifying presentation space. In *The 14th Annual ACM Symposium on User Interface Software and Technology*, pages 82–92. ACM Press, 2001.

[10] H. de Jong, M. Page, C. Hernandez, and J. Geiselmann. Qualitative simulation of genetic regulatory networks: Method and application. In *Proceeding of the 17th International Joint Conference on Artificial Intelligence*, pages 67–73. Morgan Kaufmann Publishers, Inc., 2001.

[11] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *National Academic Science*, 95(25):14863–14868, 1998.

[12] M.B. Elowitz, M.G. Surette, P.E. Wolf, J.B Stock, and S. Leibler. Protein mobility in the cytoplasm of escherichia coli. *J Bacteriol*, 181(1):197–203, 1999.

[13] K.D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[14] G.W. Furnas. Generalized fisheye views. In *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, pages 16–23. ACM Press, 1986.

[15] A.J. Griffiths. *An Introduction to genetic analysis*. W.H. Freeman, 1996.

[16] L.H. Hartwell. *Genetics: from genes to genomes*. McGraw-Hill, 2000.

[17] K.R. Heidtke and S. Schulze-Kremer. BioSim: A new qualitative simulation environment for molecular biology. In *Proceedings 6th Intl. Conf. on Intelligent Systems for Molecular Biology*, pages 85–94, 1998.

[18] 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.

[19] S. Knudsen. *A biologist's guide to analysis of DNA Microarray Data.* Wiley, 2002.

[20] F.A. Kolpakov, E.A. Ananko, G.B. Kolesov, and N.A. Kolchanov. Genenet: a database for gene networks and its automated visualization. *Bioinformatics*, 14(6):529–537, 1998.

[21] B.J. Kuipers. Qualitative simulation. *Artificial Intelligience*, 29:289–338, 1986.

[22] B.J. Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. In *Qualitative Reasoning*, page 414. MIT Press, 1994.

[23] J. Lamping, R. Roa, and P. Pirolli. A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, pages 401–408. ACM Press, 1995.

[24] Y.K. Leung and M.D. Apperley. A review and taxonomy of distortion-oriented presentation technique. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, June 1994.

[25] B. Lwein. *Genes VII.* Oxford University Press, 2001.

[26] J. Mackinlay, G.G. Robertson, and S. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 173–179. ACM Press, 1991.

[27] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information.* W. H. Freeman and Company, San Francisco, CA, 1982.

[28] H.H. McAdams and A. Arkin. Towards a circuit engineering discipline. *Current Biology 2000*, 10:R318–R320, 2000.

[29] H.H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, August 1995.

[30] G.S. Michaels, D.B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi. Cluster analysis and data visualization of large-scale gene expression data. *Pacific Symposium on Biocomputing*, 3:42–53, 1998.

[31] Visual Object Net++.
http://www.systemtechnik.tu-ilmenau.de/ drath/visual.htm. Website.

[32] E.G. Noik. A space of presentation emphasis techniques for visualizing graphs. *Proceedings of Graphics Interface '94*, pages 225–234, 1994.

[33] N. Le Novre and T.S. Shimizu. Stochsim: modelling of stochastic biomolecular processes. In *Bioinformatics*, volume 17, pages 575–576, 2001.

[34] H.C. Purchase, R.F. Cohen, and M. James. Validating graph drawing aesthetics. *Lecture Notes in Computer Science*, 1027:435–446, 1995.

[35] G.G. Robertson and J.D. Mackinlay. The document lens. In *Proceedings of the 6th Annual Symposium on User Interface Software and Technology*, pages 101–108. ACM Press, 1993.

[36] G.G. Robertson, J.D. Mackinlay, and S.K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 189–194. ACM Press, 1991.

[37] E.J. Jr. Rykiel. Testing ecological models: the meaning of validation. In *Ecological Modelling*, volume 90, pages 229–244, 1996.

[38] M. Samsonova, V. Serov, and A. Trushkina. Tools for visualization of genetic network structure and dynamics. In *Proceedings of "Computation in Cells" conference*, pages 17–18, April 2000.

[39] M.G. Samsonova and V.N. Serov. Network: An interactive interface to the tools for analysis of genetic networks structure and dynamics. *Pacific Symposium on Biocomputing*, 4:102–111, 1999.

[40] M. Sarkar, S.S. Snibbe, O.J. Tversky, and S. Reiss. Stretching the rubber sheet: A metaphor for viewing large layouts on small screen. In *Proceedings of the 6th Annual Symposium on User Interface Software and Technology*, pages 81–92. ACM Press, 1993.

[41] V.N. Serov, A.V. Spirov, and M.G. Samsonova. Graphical interface to the genetic network database genet. *Bioinformatics*, 14:546–547, 1998.

[42] M. Stone, K. Fishkin, and E. Bier. The moveable filter as a user interface. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2, pages 306–312. ACM Press, 1994.

[43] D. Thieffry, A.M. Huerta, E. Perez-Rueda, and J. Collado-Vides. From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in escherichia coli. *BioEssays*, 20:433–440, 1998.

[44] A. Wuensche. Genomic regulation modeled as a network with basins of attraction. *Proceedings of the 1998 Pacific Symposium on Biocomputing*, pages 89–102, 1998.

[45] G. Yagil and E. Yagil. On the relation between effector concentration and the rate of induced enzyme synthesis. *Biochemistry*, 11:11–17, 1971.

[46] M. Yoshida, H. Shimano, Y. Shibagaki, H. Fukagawa, and T. Mizuno. Webgennet: a workbench system for support of genetic network construction. *Poster Abstract in the International Conference on Intelligent Systems for Molecular Biology*, 2001.

Appendix A

Biological Background

Basal Activity - The base level of gene expression activity when no activator or inhibitor protein is bound to any of the gene's operator sites.

Binding Affinity - The strength of interaction between a protein and a gene. Here it is determined by a percentage that defines the likelihood of a binding event occurring.

Constructive Proteins - Constructive proteins make-up the physical structure of the cell and the organism.

Direct Regulation - Direct regulation means the protein produced by a gene is the regulator for that same gene.

Eukaryotes - An organism having cells containing a nucleus [15].

Expression - Expression occurs when a gene's operator sites are either unbound or activated by an activator protein. This is also referred to as chemically expressed.

Feedback Loops - A feedback loop occurs when one gene produces a protein that regulates the expression of another gene in a previous level of the genetic network. These feedback loops are cycles which can regulate the expression of entire sets of genes. These are also referred to as regulation loops.

Genes - Genes are segments of DNA which code to produce polypeptide chains. Each gene has a number of operator sites and coding regions. The operator sites are used to promote or inhibit transcription of the gene's coding regions.

Genetic Network - Genetic networks consist of a set of genes that are related through a set of regulatory proteins.

Genetic Network Structure - This refers to the structure of the regulatory relationships between genes. When looking at genetic networks in terms of graphs, each gene represents a node and proteins represent connection that can exist between nodes (genes). These connections can form a network of gene interaction, which is referred to as the genetic network's structure.

Gene's Required Protein - A protein required by a gene to inhibit or activate its expression.

Genetic Regulation - See Regulation.

Indirect Regulation - Indirect regulation occurs when the protein produced by a gene causes a chain of other genes to be regulated that eventually produces the protein that is needed to regulate the original gene.

Operator Sites - Operator sites are a region of DNA which requires a particular regulatory element (ie. regulatory protein). Each region either assists or hinders the binding of RNA polymerase that subsequently causes expression of the gene.

Producing Gene - A producing gene is a gene currently expressing proteins.

Prokaryotes - An organism having cells that contain no nuclear membrane and hence no separate nucleus [15].

Protein-Protein Interaction - Protein-protein interaction occurs when two proteins bind to form a protein complex.

Proteins - Proteins are large 3-Dimensional structures which are the products of a genes expression. Proteins can be either constructive or regulatory in nature. Constructive proteins constitute the physical makeup of the organism. Regulatory proteins bind to particular operator sites causing the gene's subsequent activation or inhibition.

Regulation - Regulation is the control of a gene's expression through some means, in particular, through the gene's operator site.

Regulatory Proteins - Regulatory proteins control the expression of particular genes depending on their operator site requirements. These proteins can activate (positive regulation) the expression of genes by assisting RNA polymerase in binding or they can inhibit (negative regulation) the binding of RNA polymerase preventing transcription and the impending expression of the gene.

Repression - Repression occurs when an inhibitor protein is bound to the gene's operator site causing a decrease in expression. This is also referred to as chemically repressed.

Requiring Gene - A gene that requires a certain protein to be inhibited or promoted.

Requiring Operator Site - A operator site that requires a certain protein to inhibit or promote the gene.

Reversible Binding - The unbinding of a protein from a gene's operator site.

Transcription - Transcription is the process of RNA polymerases binding to DNA and creating an RNA copy of the DNA template.

Appendix B

Inclusion of Work Previously Published

B.1 Technical Report

C.A.H. Baker, M.S.T. Carpendale and M.G. Surette, SIMULATION AND VISUALIZATION OF GENETIC REGULATORY NETWORKS. Technical report 2001-690-13, University of Calgary, December 18, 2001.

This technical report was submitted to the Department of Computer Science. Small sections were used within Chapter 3.

B.2 IEEE Visulization

C.A.H. Baker, M.S.T. Carpendale, P. Prusinkiewicz and M.G. Surette, GeneVis: Visualization Tools for Genetic Regulatory Network Dynamics. IEEE Visualization 2002, pages 243-250, 2002.

This paper [3] was published in the IEEE Visualization 2002 conference. Chapter 4 contains the content of this paper excluding the introduction and conclusion. Chapter 1 contains portions of the introduction of this paper.