Information Visualization (2003) 2, 201–217
© 2003 Palgrave Macmillan Ltd. All rights reserved 1473-8716 $25.00
www.palgrave-journals.com/ivs

GeneVis: simulation and visualization of genetic networks

Charles Baker[1]
Sheelagh Carpendale[2]
Przemyslaw Prusinkiewicz[2]
Michael Surette[3]

[1]Southern Alberta Mass Spectrometry Centre, Health Sciences Centre, University of Calgary, Calgary, Alberta, Canada; [2]Department of Computer Science, University of Calgary, Calgary, Alberta, Canada; [3]Department of Microbiology and Infectious Disease, University of Calgary, Calgary, Alberta, Canada

Correspondence:
Sheelagh Carpendale, Department of Computer Science, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4. Tel: +(403) 220 6015; Fax: +(403) 284 4707;
E-mail: sheelagh@cpsc.ucalgary.ca

Received: 6 August 2003
Revised: 15 September 2003
Accepted: 17 September 2003

Abstract

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: an individual protein representation, a protein concentration representation, and a network structure representation. The individual protein 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 individual protein representation to the protein concentration representation and from the individual protein representation to the network structure representation. Three types of lenses are used to provide different views within a representation: 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.

Information Visualization (2003) **2,** 201–217. doi:10.1057/palgrave.ivs.9500055

Keywords: Bio-visualization; visual representation; integrated views; computational biology; dynamic visualization; simulation; genetic networks

Introduction

Rapid advances in genetic research are creating a need for computational tools to support exploration of massive amounts of genetic data.[1] New technology is making it increasingly possible to obtain data about how activity levels of genes vary as a function of time,[2] yet this is still a difficult, expensive and exacting task that can be error prone. In addition, using experimental data to develop an understanding of how a particular genetic network functions is extremely complex. Despite the advances in measuring gene expression and in genome sequence analysis, understanding the behaviour of even simple genetic regulatory networks can be challenging. These systems often exhibit patterns of expression and responses that are not obvious from simple network connectivity. One approach is to develop models of the observed genetic activity. From these models, simulations and visualizations can be developed to support the exploration of the ideas on which the models are based.

In this paper, we present GeneVis, a software system for simulating and visualizing genetic regulation networks. The goal was to develop an exploratory simulation system that supports the possibility to input, adjust

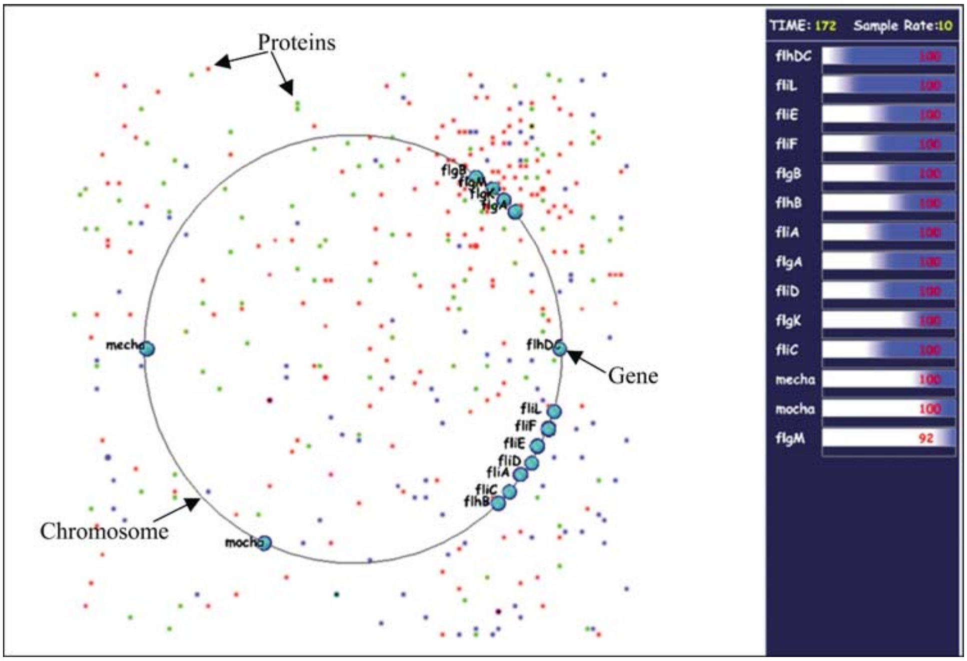


Figure 1 GeneVis simulation of flagella system of the *Escherichia coli (E. coli)* organism in progress. On the left, the individual protein visualization. The circle represents the organism's chromosome with genes represented as filled discs located around the chromosome according to their base pair position. Proteins are the small particles located throughout the environment. On the right, the gene expression history visualization.[3]

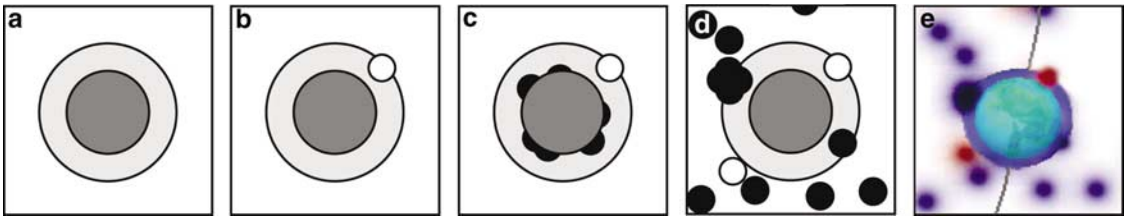


Figure 2 Diagram of a gene: (a) an inactive gene, (b) a gene with a bound regulatory protein, (c) a gene beginning to express proteins, (d) a gene continuing to express, and (e) an screen shot of the gene expressing in GeneVis.

and manipulate genetic regulatory networks in an interactive visual environment. GeneVis offers: a real-time simulation, the ability to edit and modify the data that the simulation uses, three visualizations of the simulation in progress, a visualization of the genetic network that results from the simulation, and several visual integration and interaction tools. 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. The spatial organization of the simulated entities is taken into consideration 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 processes. Figure 1 shows GeneVis while simulating the flagella system of *E. coli*. In this view, one can see two dynamic visualizations. On the left is the individual protein visualization. This visualization explicitly represents the activity of the proteins as they move in the simulation. The Gene Expression History [3] visualization presents the changes in expression activity for each gene (Figure 1, right).

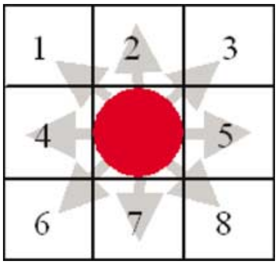


Figure 3 The eight possible directions a protein can move.

When presenting multiple visualizations of dynamic processes, it is important to coordinate the access to different views and the transitions between views. In GeneVis, we have paid particular attention to this, since with other successful visual representations of genetic network structure, such as Random Boolean Networks,[4] comprehension breaks down in transitions. Consequently, it can be difficult to assess how a mutated network relates to the original network. Sometimes a single mutation can create considerable change. To address this, we include a representational transition between two of the dynamic visualizations and animated transition from the dynamic visualization to the network structure visualization.

GeneVis has been developed in an iterative process of participatory design including a microbiologist (Surette) and computer scientists (Baker, Carpendale, Prusinkiewicz). This paper expands on[5] by including more details about the simulation, using the visualizations to explain the simulation, describing the iterative nature of the development process, and explaining how the multiple views are integrated. The following section outlines genetic regulation background. Then there is a section that describes the conceptual model and its simulation. We use illustrations from the individual protein representation as part of this description. The section on visualization presents the four visual representation; individual protein, protein concentration, gene expression history, and network structure. Then transitions between these visual representations and the methods of interacting with them are described. The penultimate section presents the related work and the last section concludes the paper.

Genetic regulation background

This section is included to provide a brief and simplified explanation of genetic regulatory networks.[6,7] Genetic networks can be thought of as sets of genes that are regulated by sets of proteins encoded by the same or other genes. GeneVis focuses on the domain of prokaryotes because their genetic interactions are less complex than eukaryotes, but in principle can be extended to any cell type.

The genetic information within each cell is encoded in deoxyribonucleic acid (DNA). A *gene* is a functional subsequence of the nucleic acid strand, which encodes the information for the production of a protein. There are several processes involved in a gene's production of a protein. During the first process, *transcription*, the RNA polymerase copies the *coding region* in the gene to ribonucleic acid (RNA). The RNA is then further processed through *translation,* where RNA message is converted to a sequence of amino acids (or protein). The process of transcription and translation is referred to as *expression*.

In addition to the coding region, each gene has a region of DNA that controls the expression of that gene. This includes a *promoter,* which directs the RNA polymerase to transcribe the gene. The activity of the promoter may be influenced by one or more *operators*. An operator is a site for binding of transcriptional regulators that can have positive (activator) or negative (inhibitor) effect on promoter activity. Each operator site is specific for particular transcriptional regulator(s) and binding is reversible. How this binding occurs and which proteins bind to which operator sites is based on biochemical laws of interaction between molecules.[8]

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* carry out other metabolic processes and make up the physical structure of the organism. The regulation of a gene's expression – either to increase or decrease – is through the gene's operator sites and is accomplished by regulatory proteins binding to the operator site. After transcriptional regulatory protein has been produced by one gene, it can move throughout the cell and interact with the operator sites of the same or other genes, altering their expression. 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 diffusive.[9] If no proteins are bound to any of the gene's operator sites, polymerase can still bind independently and transcribe the gene at some level (*basal activity*).

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. These dependencies can continue for many levels resulting in complex genetic regulatory networks. Since these networks control both the development and behaviour of organisms, they are of crucial importance in investigating the question of how organisms function.[8]

The simulation

In GeneVis, a genetic network is considered to consist of a set of genes that are related through a collection of regulatory proteins. A gene receives input through the binding of regulatory protein(s) to one or more of its operator sites. The genes and their characteristics (affinity, operator site(s), proteins expressed, etc.) can be used to create rule sets. These rule sets can be modified at any time, even during a simulation. According to the rule sets, only specific proteins are able to bind to particular sites on particular genes. 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 there is that it will come in contact with a gene that requires it. In addition, proteins decay at different rates, which also affect the cellular dynamics.

GeneVis models five factors that affect gene–protein interaction: the protein's direction of movement, the protein's speed of movement, the protein's lifespan, the protein's operator site binding, and reversibility of the protein's binding (or unbinding). In this simulation environment, all the proteins and genes are 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 rewrite 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.

The simulation is composed of a set of proteins and a set of genes that are situated in an environment. The environment is passive in that it takes no actions itself, the genes are active in that their expression behaviour changes over time and is affected by the proteins. The proteins are also active in that they move throughout the environment and interact with genes. Each gene and each type of protein is given individual rules that define its behaviour and its 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.

The simulation environment

Spatial organization The spatial organization of the simulation environment was developed to avoid restricting the effect of the input parameters on the simulation results. The initial spatial organization was found to be a serious constraint. Originally, the system was designed with a tree-like structure to both represent the structure of the genetic network and simulate its behaviour. Coupling these two goals hindered the simulation behaviour. The simulation was based on multiple particle systems. The particles were used to represent the proteins and were projected towards a set of receiving genes. Genes were activated when the particles intersected with their outer surface, causing an increase in expression. This enforced the structure of the layout on the genetic network, did not allow proteins to come to concentration levels within a general space, did not provide the possibility of proteins probabilistically affecting gene expression, and made the simulation results deterministic. 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.

To avoid the simulation environment imposing structure on the genetic network by its organization, instead of using the assumed network structure, genes are located according to their base pair positions around a circle representing the prokaryotic chromosome (Figure 1, left). These organisms typically have a single looped chromosome. The entire chromosome in the simulation represents a range of base pairs, which is specified in the input file (see the section Dynamic mutations–genetic network specification). The base pair range starts and ends on the far right side of the circle. This chromosome is centered in a 2D grid. Within this 2D grid, proteins move randomly in any of the eight possible directions (Figure 3). The edges of the grid are wrapped in both the x and y directions 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 them. While the reorganization did remove some spatial effects, it created 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 further reduced the effects of spatial location.

The simulation environment is a 2D grid representing a symbolic view of the cell. Each grid cell can be occupied by protein(s) or gene(s) 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.

Genes Inside the 2D environment genes are located around the chromosome according to their base-pair positions (Figure 1, left). Genes have two functions within this simulation: they can have proteins bind to their operator sites, which in turn regulate the gene's 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 general rule-set for each 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 (due to the Activator Factor).
If *the gene's operator site is bound with the right inhibitor protein* ***then***
the gene will express a decreased amount of protein (due to the Inhibitor Factor), which can be none.

The individual rule–sets vary according to specified basal activity rates, required proteins, and the manner in which a required protein affects the gene's activities. The genes produce proteins as calculated based on their rule-sets. 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:

$$\text{Number of Expressed Proteins} = \text{BA}(1 + (\text{AF}\times\text{AB}) + (\text{IF}\times\text{B})), \quad (1)$$

where BA is the Basal Activity (non-negative floating pointing number), F the Activator Factor (positive floating point number), AB the Activator Bound (0 if activator is not bound, 1 if it is bound) IF the Inhibitor Factor (negative floating point number), and IB is the Inhibitor Bound (0 if inhibitor is not bound, 1 if it 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. The activator factor is a positive number and the inhibitor factor is negative, causing the corresponding increases and decreases in gene expression depending on the bound protein. Eq. (1) is calculated for each gene at every time step of the simulation calculating the number of proteins that a gene will produce at the current time step. As genes change state between activated and inhibited, their protein production will change. Figure 2 is a diagram of GeneVis's visual representation of a gene. A gene is displayed as two concentric circles (Figure 2a). 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 2(b) shows a regulatory protein bound to the operator site. Using full circumference of the gene's outer circle for operator sites prevents the probability of intersections from being too low. When the appropriate activator or inhibitor protein is bound, the gene adjusts its expression (Figure 2c), creating proteins that are displayed emerging from within the inner circle (Figure 2(c)–(e)).

Proteins Proteins in GeneVis are the only elements free to move with the environment. The movement of proteins is responsible for creating the possibility of interaction between genes and proteins. The movement of proteins is typically diffusive although it does not need to be so.[9] In GeneVis, protein movement is approximated by randomly moving each protein in the simulation in any of the eight directions indicated by the grid – up, down, left, right, and the four diagonals (see Figure 3). Each protein also has a maximum distance it 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 maximum can be defined in the input file and can be modified during the simulation. In each simulation step the protein is moved using the randomly selected step size in the randomly selected direction. The larger the step size, the faster the proteins will disperse. Each protein has a lifespan. The protein's life is decreased each step in the simulation until it reaches zero, at which point the protein is removed from the simulation environment.

The binding of proteins to genes is accomplished in each grid-cell that is covered by a gene (the number of grid cells a gene covers depends on the resolution of grid). Thus, the use of the grid makes it possible to test intersections between proteins and genes in a linear time with respect to the resolution of the grid irrespective number of proteins and genes (Figure 4). In GeneVis, protein binding can occur anywhere within the grid squares occupied by the gene and not just the outer edge (Figure 4) of the gene. This increases the number of possible binding events. If a protein is intersecting a gene, then the operator site's binding affinity probability 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 reversible binding, which is tested for at every time step according to a user specified probability. The second circumstance is when protein decay causes it to unbind from the operator site. This effect is simulated by a protein decay factor: the protein unbinds from the gene when its decay factor reaches zero.

Figure 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 then disperse 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

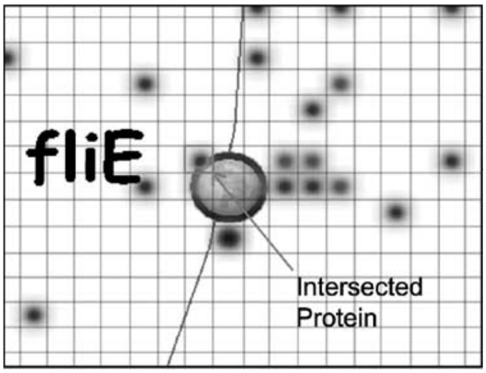


Figure 4 A gene covering a number of squares within the 2D grid. An intersection test between the gene and the proteins has revealed the marked protein as an 'intersected protein' about to be bound.

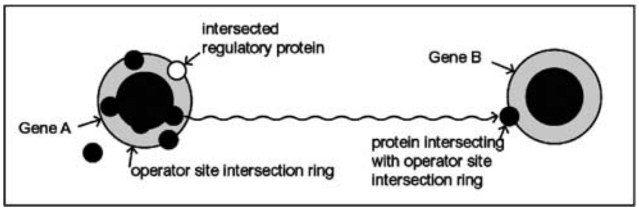

intersected regulatory protein
Gene B
Gene A
protein intersecting with operator site intersection ring
operator site intersection ring

Figure 5 A single protein bound to gene A promotes its expression. A resulting protein moves randomly through the simulation environment until intersecting with gene B that requires it.

protein. Figure 6 illustrates this process with screen shots from GeneVis.

Dynamic mutations – genetic network specification

There are two methods of specifying a genetic network within GeneVis: using an input file or with a dialog box. In order to describe different prokaryotic genetic networks, the input to GeneVis is specified in a flexible format. The intention is to support evaluative networks and interactive adjustments.

Input The schema of the input file has general fields that are applicable to most prokaryotic genetic networks. This simple format is intended to allow people to explore genetic network behaviour without the prerequisite of significant computer or molecular biology knowledge.

The simulation algorithm

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

```
1. For each gene calculate the number of proteins that have been produced at the current time step.
2. Update the released protein positions by moving them a random number of grid squares in a random direction (one of the eight possible
   directions).
3. Subtract one unit from the lifespan of each active protein. This models protein decay.
4. Test each protein for intersecting a gene:
       If the grid position of the protein overlaps a gene, then
           if the protein is a required activator or inhibitor then
               that protein may bind based on the gene's affinity.
5. If a protein's life has expired, then:
       Delete the protein from the environment,
       If the protein was bound to a gene, then
           free that operator site.
```

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. A small genetic network may only occupy a small part of the looped chromosome. The BasePairRange can be specified as the range that contains the genes in the current network. This can then be used as the range for the circular chromosome in the simulation. The input file then has a specification for each gene, which includes:

(1) Unique Gene Identifier.
(2) Name of the gene.
(3) Base pair position of the gene.
(4) A flag to indicate if the gene is an initiator.
(5) An identifier for the protein produced by this gene.
(6) The basal activity rate of the gene.
(7) The expression rate of the gene.
(8) The expressed protein's decay rate.

A gene can have several operator sites. All operator sites are listed in the input file consecutively. Each operator site has the following specification:

(1) The identifier of the protein that promotes the expression of the gene.
(2) The degree to which the promoter protein activates the gene's expresson.
(3) The identifier of the protein that inhibits the expression of the gene.
(4) The degree to which the inhibitor protein suppresses the gene's expression.
(5) The affinity for the binding of a protein to an operator site.

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 it is useful for debugging network dynamics, which in turn is useful of developing conceptual models about the genetic regulatory behaviour: from the blank state it is much easier to see the interaction of proteins and genes as they start to fill the cell. The cell becomes populated from the blank state within a relatively short period of time. For example, it takes <3 min. for the simulation environment to become populated when using the network from Figure 1 on an AMD Athlon 1.3 GHz computer with 512 MB 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.

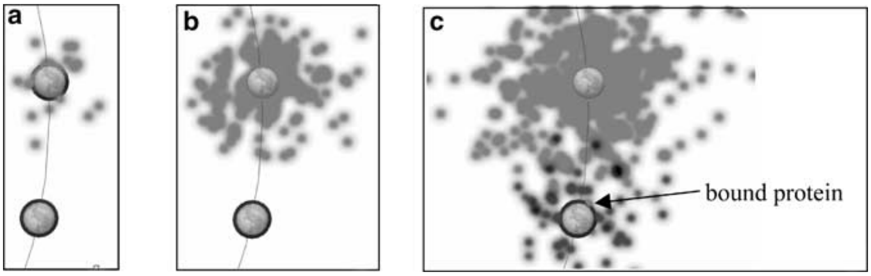


Figure 6 (a) Gene A begins to express, (b) the resulting proteins begin to spread, (c) a protein binding to gene B triggers its expression.

This feature also makes it possible 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 that 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.

Adjustments All of the parameters in the input file are used to specify the genetic network. The input files make it possible to store conveniently genetic network topologies and to initialize them. It is also possible to modify interactively 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. Changing basal expression rates and activator/repressor affinities in simple gene networks has resulted in predictable changes in the magnitude and dynamics of gene expression and as well as in the observed stochasticity.

A user can change the gene's name, base pair position, produced protein, decay rate, and protein colour used in this visualization. Also the gene's basal activity, expression rate, and the properties of each of its operator sites can be edited, including 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 on the simulation. These adjustments can be used to build genetic networks and change parameters that affect the network dynamics. Any changes can be saved in the input file for reuse in later simulations.

The visualizations

This section describes the four visual representations; individual protein representation, protein concentration representation, gene expression history, and genetic network structure. The individual protein representation is covered very briefly since it was used to illustrate the description of the simulation in the previous section.

Individual protein representation

The individual protein visualization is the one that was used to illustrate the description of the simulation. All the images in the paper thus far are of the individual protein visualization. The looped chromosome of the prokaryote is represented as a large circle placed centrally in the simulation environment (Figures 1 and 7). Each gene is depicted as two filled concentric circles (Figures 5 and 6), which are placed on the chromosome loop according to their base-pair position (Figure 7). The small coloured particles that are spread throughout represent the proteins. The protein representation is created as a texture mapped square in which the colour is saturated in the centre and attenuated towards the edges. This attenuation creates proteins as discs with a fuzzy circular appearance. This attenuation clarifies the separation between two proteins in adjacent grid squares and makes an increase in the number of proteins present in a given grid square apparent through an increasing brightness. Each individual protein is drawn at each time step during the simulation. Visualizing the actions of the individual proteins as they move according to the simulation makes the random motion, decay of the individual proteins and the change in activity rate for each gene explicit.

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 lifespans are updated. The genetic network dynamics are visualized as the simulation proceeds. 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 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.

Protein concentration representation

The genetic network dynamics can also be visualized in a more macroscopic manner, by showing protein

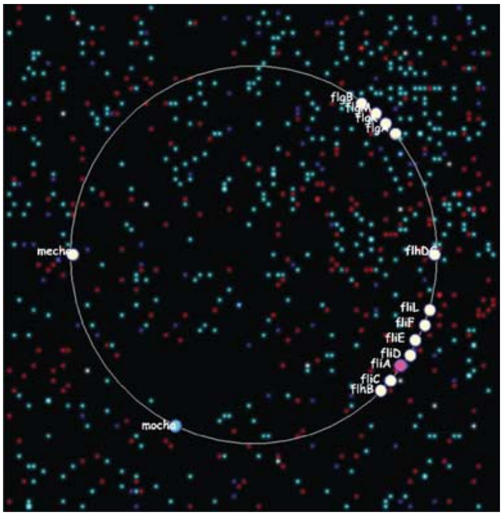


Figure 7 Individual protein representation.

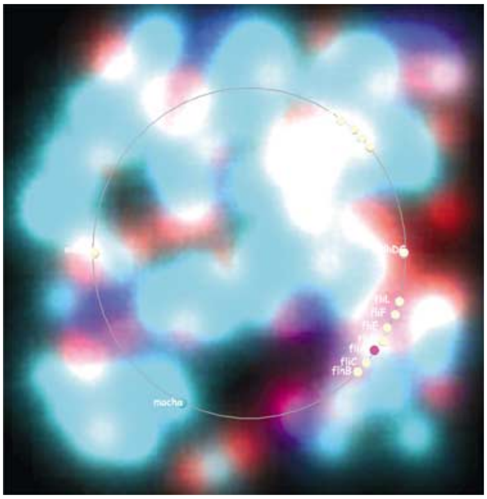


Figure 8 Protein concentration representation.

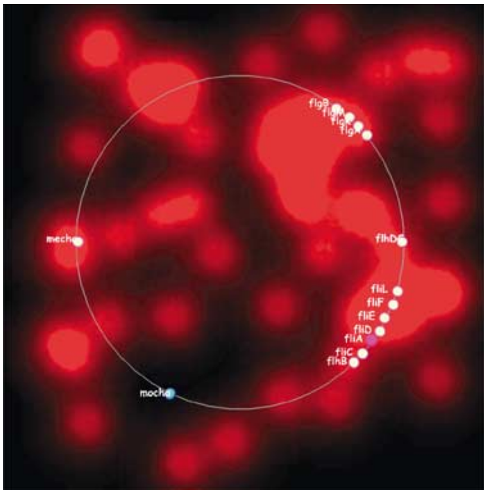
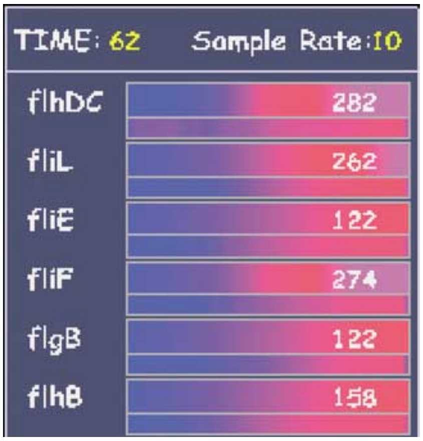


Figure 9 Protein concentration representation for one protein type.



TIME: 62 Sample Rate:10

flhDC 282

fliL 262

fliE 122

fliF 274

flgB 122

flhB 158

Figure 10 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 static rather than dynamic mode.

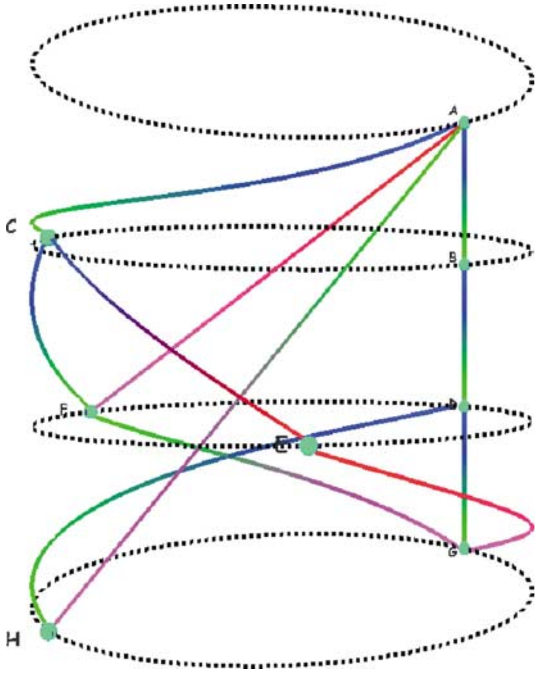


Figure 12 An example of the genetic network structure visualization: Each ring represents a level in the gene hierarchy. The genes (spheres) are related by lines representing regulatory proteins. Forward and backward lines are drawn blue and magenta, respectively, at the producing end. At the receiving end, all promoting connections fade to green and all inhibiting connections fade to red.

concentrations rather than the position of individual molecules. The probability of a gene's expression being affected increases and decreases with the 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 identify visually when the protein concentrations have increased. In GeneVis, the proteins can be viewed as individual molecules, 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 individual protein 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 is proportional to the number of proteins it represents. This attenuated disc is centered at the location of one of the proteins it represents.

Figure 8 shows the protein concentration representation at the same point in a simulation as the individual protein representation in Figure 7. Notice how it is hard to tell if the proteins are uniformly distributed in the individual protein representation. This information is more readily apparent in the protein concentration representation. In Figure 8 one can see that that the 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 9.

Gene expression history visualization

Expression analysis is a technique for measuring and visually representing the expression of genes. The new methods available to experimentalists allow the simultaneous measurement of large number of genes. Consequently, expression analysis has become an invaluable asset to biologists in the study of genetic networks.[3]

In GeneVis, gene expression histories show the number of active protein molecules (*produced proteins–decayed proteins*) currently present in system at each step of the simulation. 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[3] where one colour (blue) is used to indicate *no expression* and a second colour (red) to indicate *expression* (Figure 10). These colour values are plotted vertically in a rectangle with intermediate levels of expression shown by interpolating between the two colours. All gene expression histories are updated in dynamically.

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 a previous live or simulated experiment, and used for a comparison or a reference. These expression histories in the bottom rectangle are imported from a file. Each rectangle indicates the expression history of the gene from the left side at time 0 to the right side, which is the current time in the simulation. As the simulation proceeds, the expression history is compressed toward the left to include the current expression data. The number in the right side of the rectangle shows the gene's current number of active proteins. When displaying results from a file (either laboratory or simulated) they can be shown either in full, showing all the data from time 0 to the end, or from time 0 to the current time step in the simulation. All expression bars are positioned on the right side of the screen, and are labelled according to the gene they represent. They are visible concurrently with the dynamic visualization that has been selected. To facilitate the comparison of simulation runs, it is possible to output the simulation's expression histories to a file for numerical comparison, plotting, or statistical analysis.

Genetic network structure visualization

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 or previous simulation results. However, biologists are also concerned with the structure of a genetic network that emerges from interactions between genes and proteins. This type of structure is not apparent in either the individual protein 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.

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 analysing gene–protein interaction during the simulation. Every gene is checked for the earliest time at which a regulatory protein binds with it and affects its activity level. This is used to place that gene appropriately within the network.

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,[10] and one method of illustrating these interactions is shown in Figure 11.

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. Similarly, the genes in the third row require

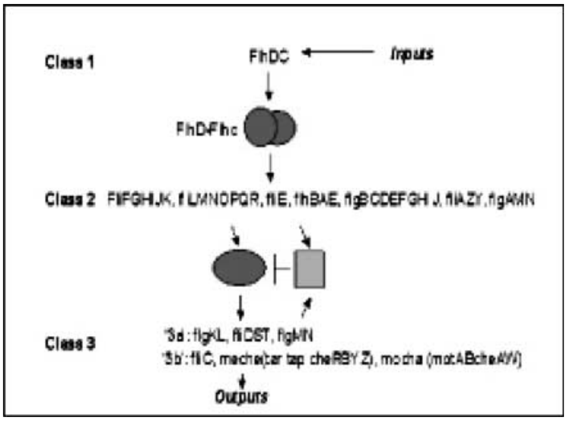

Class 1
FlhDC
Inputs
FlhD-Flhc
Class 2 FliFGHIJK, fliLMNOPQR, fliE, flhBAE, flgBCDEFGHIJ, fliAZY, flgAMN
Class 3
'3a': flgKL, fliDST, flgMN
'3b': fliC, meche(tar tap cheRBYZ), mocha (motABcheAW)
Outputs

Figure 11 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 to the genes. There are three levels of genes in this network (adapted from Kalir *et al*[10]).

regulatory proteins from the previous row to express. Thus, these levels in the hierarchy 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.[10] Given the significance of these levels, one goal in creating the network structure visualization was to make them explicit.

In the structural visualization, the regulatory relationships to be represented include: forward promoting and inhibiting relationships, backward promoting and inhibiting relationships, and within-level relationships including 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 at 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.[11]

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 10 to a 3D ring, and the genes within that level are distributed evenly around the ring (Figure 12). 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 regulation 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.

Integrating the visualizations

Many different 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.[12] Similarly, when we create visual representations it is our intention to reveal particular aspects of the data. Providing more than one visualization offers support for different ways of thinking about the data. However, making the relationships between these methods visually explicit can be even more useful.

To support comparisons the Gene Expression History can be viewed concurrently with either of the other two dynamic visualizations (Individual Protein or Protein Concentration). It can also be viewed concurrently with a previous Gene Expression history. This previous history can be from either an earlier simulation or wet-lab data.

To provide support for visual transition from one representation to another, a visual transition between the Individual Protein and the Protein Concentration representations has been developed (see the next section) and the transition between either of the dynamic representation and the static Network Structure representation is animated (see the section Animating between dynamic and static visualization). Also, partial and combined representation views are possible through the use of Fuzzy Lenses (see the section Lenses)

Representational transition

In the visualization section, two of the visual representations of genetic regulatory dynamics presented are the individual protein representation and protein concentration representation. Here, we describe how a 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 13a shows the *individual protein view*, in which individual proteins are displayed. This view is equivalent to the representation in Figures 1 and 7. Figure 13d shows the *protein concentration view*. Figures 13(b and c) present in-between views that correspond to the use of increasingly large discs that represent increasingly large numbers of proteins.

In reverse order, changing from the concentration representation to the individual representation, the attenuated discs that represent groups of proteins become smaller until they represent individual proteins. This reverse direction can be seen in the progression from Figure 13d to Figure 13a. Representational transformation is created by changing the size of attenuated discs, the number of proteins a disc represents, and the number of discs shown. In each case the attenuated disc covers the same area that the proteins it represents would cover.

Table 1 shows how the representational transition between individual protein 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 1.

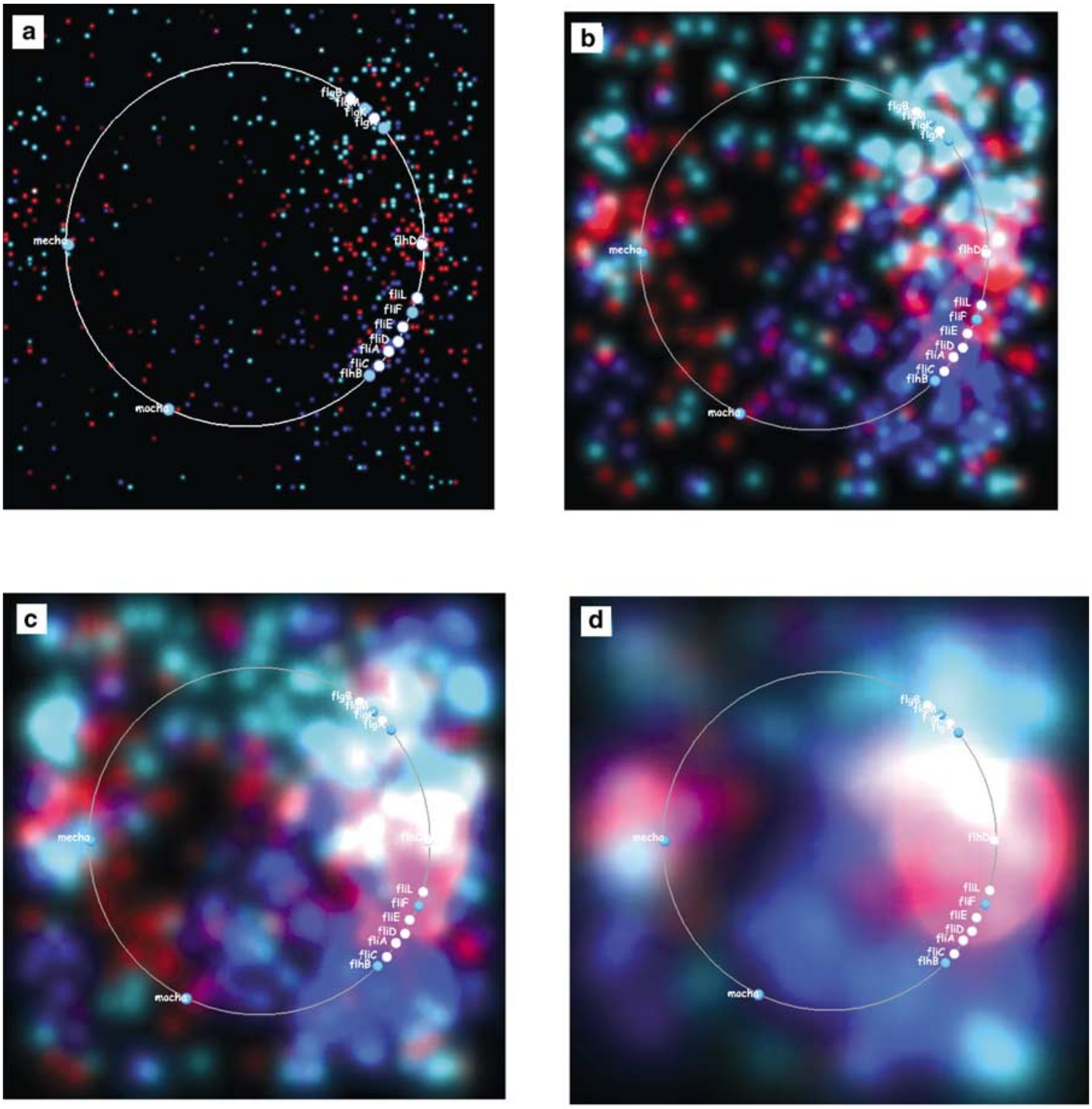


Figure 13 Inter representational transition. (a) Protein view: 100% displayed, with individual proteins viewable; (b) Transition view: 65% displayed with small concentrations discs viewable; (c) Transition view: 35% displayed with larger concentration discs viewable; (d) Concentration view: 1.56% displayed, with concentrations viewable.

Table 1 Representational transition: this table shows how the number of displayed discs relates to the number of proteins represented by the attenuated disc and the size of that disc

Displayed %	Proteins represented by a 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

For an individual protein 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 continues 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 centre. 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 centred 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.

Animating between dynamic and static visualizations

To visually integrate the simulation and the network structure, the transition between the two visualizations can be animated. This animation can be viewed as a continuous motion or stepped through in either direction. Figure 14 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 added to the circular chromosome of the simulation visualization (Figure 14a). Next, each level is drawn inward, one by one, until the network is partitioned into levels. Figure 14b shows the display after the first level has been drawn inward. At this point, the network is represented as a series of concentric rings in a 2D plane. The next stage of the transition (Figure 14c) moves the viewpoint, to give a side view. Then each ring is translated upwards, showing each level and its connections (Figure 14d). At the end of the animation in Figure 14e, all the rings are enlarged to the same diameter and the forward connections are changed to curves. Each transition takes place gradually to allow the user to track individual genes from one step to the next.

Lenses

Fuzzy lenses Fuzzy lenses have been implemented in GeneVis to provide access to alternate representations in different areas of the visualization. Lenses[13,14] 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 15a), a *protein lens*, which provides the individual protein view of the simulation (Figure 15b), and a *dual lens*, which shows both the concentration view and the individual proteins (Figure 15c). 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. The lenses are *fuzzy* in 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 15). With the exception of their fuzzy edges, these lenses relate directly to the concepts presented as Magic Lenses[13] in that an alternate representation or a combined representation is shown within the lens.

Base pair lens GeneVis simulates genetic networks for prokaryotic organisms. As discussed before (section on Genetic regulation background), 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.[6] Within the chromosome, genes with related functions may be grouped closely together.[6] When genes with close base pair positioning are visualized within GeneVis, their representations may overlap due to limited resolution (Figure 16a). 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*

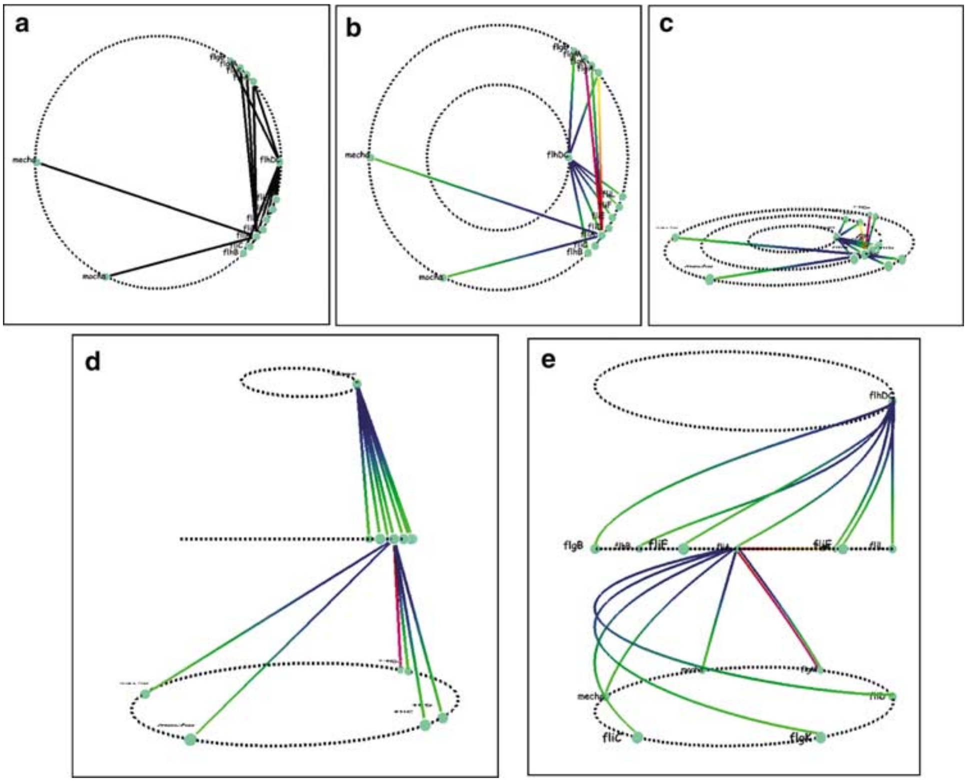


Figure 14 Visual integration that moves the user from the simulation visualization (a) to the network structure visualization (d and e).

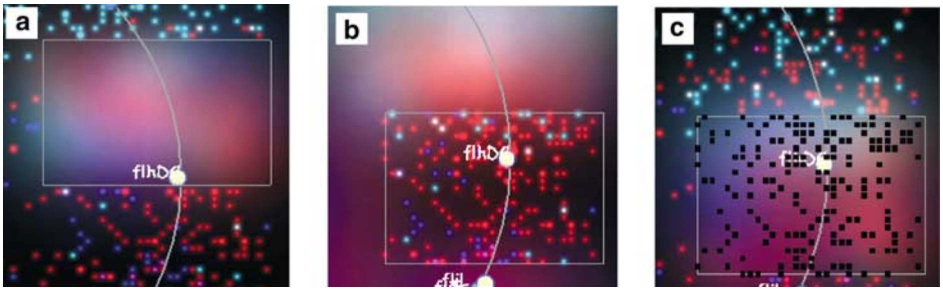


Figure 15 Fuzzy lenses: (a) concentration lens, (b) protein lens, and (c) dual lens.

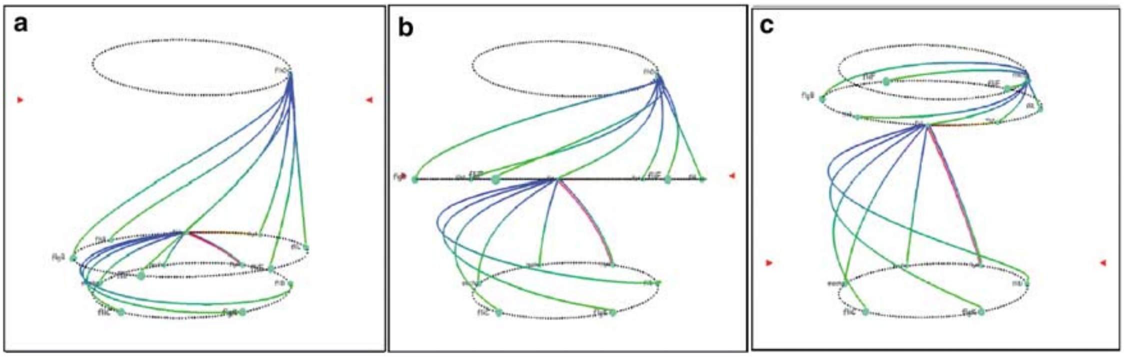


Figure 19 Ring lens view transformation: (a) ring lens cursor positioned at the top of the network, (b) ring lens cursor positioned at the middle of the network, and (c) ring lens cursor positioned at the bottom of the network.

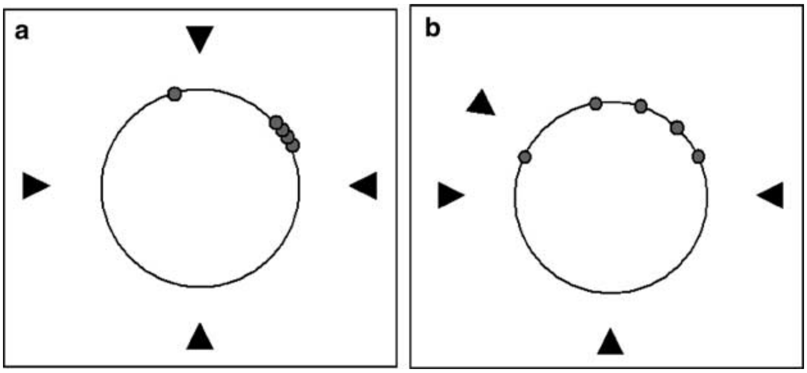


Figure 16 Diagram of the base pair lens: (a) genes clustered on the left side of the chromosome, (b) genes distributed more evenly.

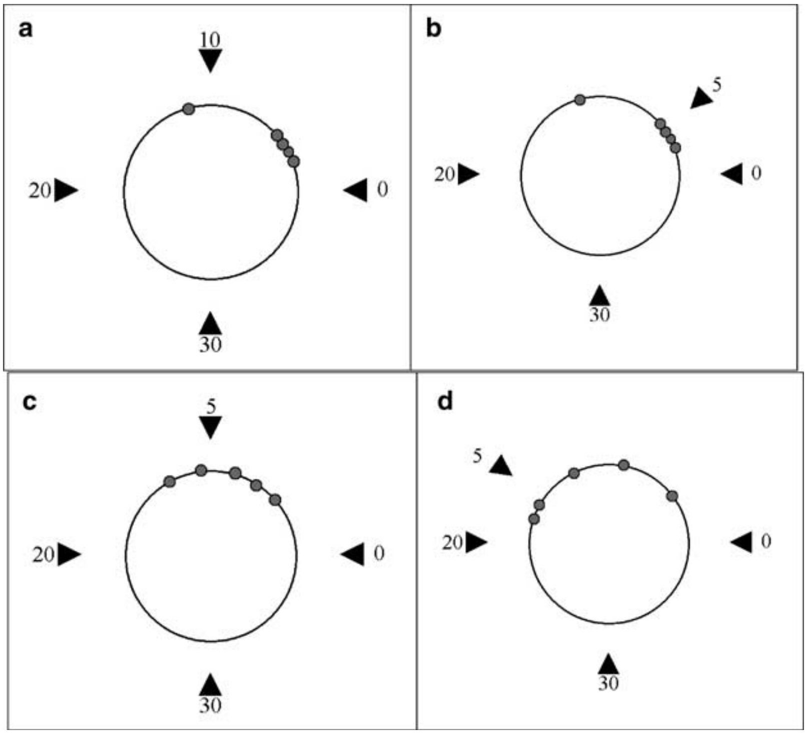


Figure 17 Diagram showing the interaction of the base pair lens: (a) the handles are evenly distributed across the base pair range which is 40 in this diagram, (b) the top handle is adjusted with the right mouse button to narrow the affected base pair range to 5, (c) the top handle is moved to the left using the left mouse button spreading the genes across a greater circumference, (d) the top handle moved farther left, further spreading the genes.

that allows the user to interactively separate the genes and then proceed with the simulation.

The base pair lens consists of four sliders. Figure 16 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 Figure 16a 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 quarter of the circle and compress the top left-hand quarter. The right image of Figure 16b 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 16 shows a diagram of the interaction with the handles when using the right mouse button. In Figure 17, the

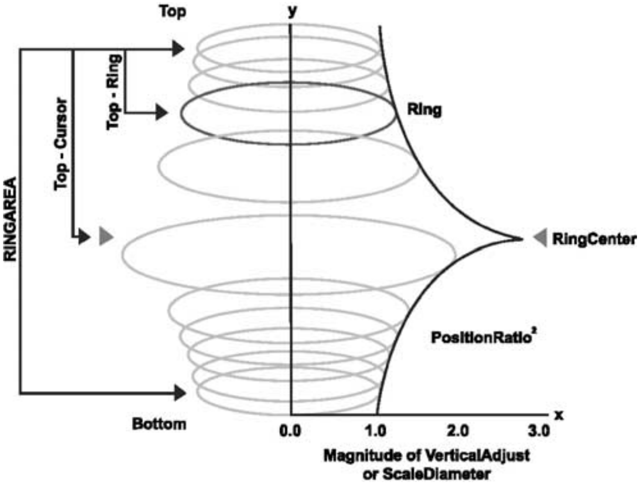


Figure 18 Diagram outlines the different calculations and variables that make up the distortion function for the *ring lens.*

right mouse button has been used to change the acting base-pair range from 0 through 10 (Figure 17a) to 0 through 5 (Figure 17b). Then the left mouse button can be used again to stretch out the 0 through 5 section over a greater circumference (Figure 17c, d).

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* (Figure 18) 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:

$$\begin{aligned}&\mathit{If}\,(\mathit{Ring} > \mathit{LensCenter})\\&\quad \mathit{PositionRatio} = (\mathit{Top} - \mathit{Ring})/(\mathit{Top} - \mathit{LensCenter})\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathit{else}\\&\quad \mathit{PositionRatio} = (\mathit{Ring} - \mathit{Bottom})/(\mathit{LensCenter} - \mathit{Bottom}).\end{aligned} \tag{3}$$

Here *Ring* is the vertical position of the ring to be adjusted, *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:

$$\mathit{scaleDiameter} = (\mathit{MaxMag} * \mathit{PositionRatio}\,{}^\wedge 2) + 1.0. \tag{4}$$

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:

$$\mathit{VerticalAdjust} = (\mathit{PositionRatio}{}^\wedge 2(\mathit{Top} - \mathit{Bottom}))/\mathit{VerticalScaleFactor}. \tag{5}$$

VerticalAdjust is subtracted from *Top* if the ring is above the lens centre, and added to *Bottom* if it is below the lens centre. The vertical position of the ring lens is controlled by the mouse. Figure 19 shows screen shots of the ring lens in different positions. Figure 19a shows the ring lens shifted towards the bottom of the view. This makes the connections between the upper two rings more visible by increasing the amount of space between them. Figure 19b shows the ring lens placed at the second level ring, causing it to be enlarged in diameter. Figure 19c shows the lens near the top of the view, opening up the space between the lower 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.

Related work

GeneVis is a software system for exploring genetic regulatory networks. Its features include: a real-time simulation, three visualizations of the simulation as it progresses, and a visualization of the genetic network that results from the simulation. It also provides the ability to edit and modify the data that the simulation uses and several visual integration and interaction tools. Our work is related to previous research in the domains of genetic simulation models, visualizations of the genetic network structure, and genetic database access interfaces.

Each previous simulation model takes a different approach to simulating genetic networks. Random Boolean Networks[4] are based on a simple model in which an individual gene's expression is either on or off. Simulations are run as a cellular automaton. A given gene's state is based on the state of the genes from which it receives input. The results are displayed as a directed graph. It is possible to modify a given gene(s), however, the resulting graph maybe so different from the previous that understanding the changes is difficult. Network[15] uses the similar logic as Random Boolean Networks and includes some interface methods for searching for a given gene and for comparisons.

Circuit Simulation[16] presents the resulting graph in a similar manner to circuit diagrams including such elements as AND and NOT gates. The simulation is based on the logic in the diagram. StochSim,[17] although not specifically designed for genetic networks, can be used to simulation them. This simulation is based on the probabilities that a given pair of molecules will interact. At each time step every pair is checked. BioSim[18] will approximate information that is missing and returns a set of possible behaviour trees. Genetic Network Analyzer[8] uses differential equations to simulate protein diffusion and network behaviour. The simulation obtains data from files containing the differential equations. The protein concentrations resulting from the simulation can be displayed as a function of time. The results from these simulations have been presented as charts, in which the simulated gene activity levels are plotted as functions of time or as directed graphs. 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.

There are several tools, such as GeneGraph,[19] WebGenNet,[20] GeneNet Viewer,[21] and G.Net[22] that visualize the genetic network structure by creating directed graph representations of there data. The nodes of the graph represent genes and the directed edges represent the proteins that are produced by one gene and required by another gene. These tools are usually linked to a particular data source.

Software tools being developed to aid in the access and exploration of the growing genome database include Ensemble,[23] and GenDB.[24] However, their focus is on data access rather than visualization. Chi *et al*[25] developed a visualization for multivariate sequence data that supports comparisons for more than one data access. Virtual environments have been used to surround a user with clusters from different gene expression datasets.[26] Adams *et al*[27] provide visual data access software in which one can zoom from a view of a chromosome to a view of the individual genes and their associated proteins.

Conclusions

GeneVis evolved from a deterministic system to a 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 and the inclusion of an initial approximate populated state, have been identified for future research.

The GeneVis simulation model for genetic network behaviour is based on gene–protein interaction. Through having each component act as an individual, an emergent complexity can occur as each individual component interacts with others in the environment. The component architecture allows for additional mechanism to be added adjusted within the visualization. Users can easily alter genetic network parameters when modelling its behaviour within GeneVis. 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: protein direction of movement, distance of protein movement, protein life span, operator site binding, and reversible Binding.

GeneVis reads in data files, provides interactive editing of these files and the changes are reflected immediately in the simulation and visualization. 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. 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. The following are the methods and tools provided for the exploration of the simulation.

Individual Protein view: This 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.

Protein concentration view: This 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.

Gene expression histories: During the simulation, a plot of each gene's produced protein over time is recorded and displayed with the simulation. The data are recorded and visualized as gene expression histories 3 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.

Network structure visualization: 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. The network structure reflects any alteration made to the parameters of the simulation.

View transformation tools: A number of view transformation tools provide different viewing methods for both the simulation and the structure model. A *representational transformation* provides a method of gradually switching between the individual protein view and the protein concentration view. Also, *fuzzy lenses* 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 the *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.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Canadian Institutes of Health Research and Intel Inc.

References

1 Stein L. *Creating a bioinformatics nation. a web services model will allow biological data to be fully exploited*. Nature 2002; **417**: 119–120.

2 Knudsen S. *A Biologist's Guide to Analysis of DNA Microarray Data*. Wiley: New York; 2002.

3 Eisen M, Spellman P, Brown P, Botstein D. *Cluster analysis and display of genome-wide expression patterns*. National Academic Science 1998; **95**: 14863–14868.

4 Wuensche A. *Genomic regulation modeled as a network with basins of attraction.* 1998 Pacific Symposium on Biocomputing (Maui, Hawaii, U.S.A., 1998), 89–102.

5 Baker CAH, Carpendale MST, Surette M, Prusinkiewicz P. *GeneVis: visualization tools for genetic regulatory network dynamics.* IEEE Conference on Visualization, Vis'02 (Boston, U.S.A., 2002), IEEE Computer Society Press: Silver Spring, MD; 243–250.

6 Griffiths AJ. *An Introduction to Genetic Analysis*. W.H. Freeman: New York; 1996.

7 Hartwell LH. *Genetics: From Genes to Genomes*. McGraw-Hill: New York; 2000.

8 de Jong H, Page M, Hernandez C, Geiselmann J. *Qualitative simulation of genetic regulatory networks: method and application.* 17th International Joint Conference on Artificial Intelligence (San Mateo, U.S.A., 2001) Morgan Kauffman: Los Altos, CA; 67–73.

9 Elowitz MB, Surette MG, Wolf PE, Stock JB, Leibler S. *Protein mobility in the cytoplasm of* Escherichia coli. Journal Bacteriol 1999; **181**: 197–203.

10 Kalir S, McClure J, Pabbaraju K, Southward C, Ronen M, Leibler S, Surette MG, Alon U. *Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria*. Science 2001; **292:** 2080–2083.

11 Purchase H, Cohen R, James M. *Validating graph drawing aesthetics.* Symposium on Graph Drawing, Lecture Notes in Computer Science, Vol. 1027 (Passau, Germany, 1995), Springer-Verlag: Berlin: 435–446.

12 Marr D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman and Company: New York: 1982.

13 Bier E, Stone M, Fishkin K, Buxton W, Baudel T. *A Taxonomy of see-through tools.* ACM Conference on Human Factors in Computing Systems CHI'94 (Boston, MA, U.S.A., 1994), ACM Press: New York; 358–364.

14 Carpendale MST, Jirasek C. *A framework for unifying presentation space.* ACM Symposium on User Interface Software and Technology UIST'00 (Orlando, U.S.A., 2001), ACM Press: New York; 82–92.

15 Samsonova MG, Serov VN. *NetWork: An interactive interface to the tools for analysis of genetic networks structure and dynamics.* Pacific Symposium on Biocomputing (Hilo, Hawaii, U.S.A., 1999), 102–111.

16 McAdams H, Shapiro L. *Circuit Simulation of Genetic Networks*. Science 1995; **269**: 650–656.

17 LeNovère N, Shimizu T. *StochSim: modelling of stochastic biomolecular processes*. Bioinformatics 2001; **17**: 575–576.

18 Heiftke K, Schulze-Kremer S. *BioSim – a new qualitative simulation environment for molecular biology.* Sixth International Conference on Intelligent Systems for Molecular Biology 1998 (Montreal, Canada, 1998), AIAA Press: New York; 85–94.

19 Samsonova M, Serov V, Trushkina A. *Tools for visualization of genetic network structure and dynamics*. 'Computation in Cells' Conference (University of Hertfordshire, U.K., 2000), 17–18.

20 Yoshida M, Shimano H, Shibagaki Y, Fukagawa H, Mizuno T. *WebGen-Net: a workbench system for support of genetic network construction. International Conference on Intelligent Systems for Molecular Biology* (Copenhagen, Denmark, 2001), Poster Abstract.

21 Kolpakov FA, Ananko EA, Kolesov GB, Kolchanov NA. *GeneNet: a database for gene networks and its automated visualization*. Bioinformatics 1998; **14**: 529–537.

22 Aoshima K, Ikawa M, Tanaka S. *A visualization tool for gene network discovery – G.NET*. Genome Informatics 2002; **13**: 445–446.

23 Clamp M, Andrews D, Barker D, Bevan P, Cameron1 G, Chen Y, Clark L, Cox T, Cuff J, Curwen V, Down T, Durbin R, Eyras E, Gilbert J, Hammond M, Hubbard T, Kasprzyk A, Keefe D, Lehvaslaiho H, Iyer V, Melsopp C, Mongin E, Pettett R, Potter S, Rust A, Schmidt E, Searle S, Slater G, Smith J, Spooner W, Stabenau A, Stalker J, Stupka E, Ureta-Vidal A, Vastrik I, Birney E. *Ensembl 2002: accommodating comparative genomics*. Nucleic Acids Research 2003; **31**: 38–42.

24 Bioinformatics group of the Center for Genome Research, Bielefeld University. *The GENDB system, an open source genome annotation system*. http://gendb.genetik.uni-bielefeld.de/ (accessed 3 August 2003). http://www.Genetik.Uni-Bielefeld.DE/ZfG/. http://www.genetik.uni-bielefeld.de/ZfG/Bioinformatics/projects.html. August 3, 2003.

25 Chi EH, Barry P, Shoop E, Carlis JV, Retzel E, Riedl J. *Visualization of biological sequence similarity search results.* IEEE Conference on Visualization Vis'95 (Atlanta, U.S.A., 1995) IEEE Computer Society Press: Silver Spring, MD; 44–51.

26 Kano M, Tsutsumi S, Nishimura K. *Visualization for genome function analysis using immersive projection technology.* IEEE Virtual Reality Conference 2002 (Orlando, U.S.A., 2002), IEEE Computer Society Press: Silver Spring, MD; 224–231.

27 Adams RM, Stancampiano B, McKenna M, Small D. *Case study: a virtual environment for genomic data visualization*. IEEE Conference on Visualization Vis'02 (Boston, U.S.A., 2002), IEEE Computer Society Press: Silver Spring, MD; 513–516.