ORIGINAL ARTICLE

Dong Hyun Jeong
Alireza Darvish
Kayvan Najarian
Jing Yang
William Ribarsky

Interactive visual analysis of time-series microarray data

D.H. Jeong (✉) · J. Yang · W. Ribarsky
Charlotte Visualization Center,
The University of North Carolina
at Charlotte, USA
{dhjeong, jyang13, wribarsky}@uncc.edu

A. Darvish
Bioinformatics & Advanced Signal
Processing Lab.,
Dept. of Computer Science,
The University of North Carolina
at Charlotte, USA
adarvish@uncc.edu

K. Najarian
Biomedical Signal and Image
Processing Lab.,
Dept. of Computer Science,
School of Engineering,
Virginia Commonwealth University, USA
knajarian@vcu.edu

Abstract Estimating dynamic regulatory pathways using DNA microarray time-series can provide invaluable information about the dynamic interactions among genes and result in new methods of rational drug design. Even though several purely computational methods have been introduced for DNA pathway analysis, most of these techniques do not provide a fully interactive method to explore and analyze these dynamic interactions in detail, which is necessary to obtain a full understanding. In this paper, we present a unified modeling and visual approach focusing on visual analysis of gene regulatory pathways over time. As a preliminary step in analyzing the gene interactions, the method applies two different techniques, a clustering algorithm and an auto regressive (AR) model. This approach provides a successful prediction of the dynamic pathways involved in the biological process under study. At this level, these pure computational techniques lack the transparency required for analysis and understanding of the gene interactions. To overcome the limitations, we have designed a visual analysis method that applies several visualization techniques, including pixel-based gene representation, animation, and multi-dimensional scaling (MDS), in a new way. This visual analysis framework allows the user to quickly and thoroughly search for and find the dynamic interactions among genes, highlight interesting gene information, show the detailed annotations of the selected genes, compare regulatory behaviors for different genes, and support gene sequence analysis for the interesting genes. In order to enhance these analysis capabilities, several methods are enabled, providing a simple graph display, a pixel-based gene visualization technique, and a relation-displaying technique among gene expressions and gene regulatory pathways.

Keywords Visual analysis · Information visualization · Microarray anaysis · Bioinformatics

1 Introduction

Improving medicine and human health is the primary goal of life scientists. To significantly improve medicine and human health, a more detailed understanding of the vast networks of molecules and interactions among these molecules as well as their interactions with different types of drugs is required. A more quantitative knowledge of dynamic gene regulatory pathways, i.e., the gene interactions through time, can provide an understanding of the time-dependent enhancement and suppression of gene activities and drug effectiveness. Dynamic modeling is a process

through which the future of a set of variables is predicted based on the present and past values of the same variables as well as other variables. Such a modeling process not only allows the investigation of time-based interactions among physical/biological phenomena but also provides a framework to explore properties such as stability and dynamic network connectivity. Using such a model one can predict, for instance, how a particular drug can "turn on or off" a certain gene or group of genes and what combinations of drugs may be more effective over a certain period of time. However, since dynamic biological pathways involve highly complex interactions among the genes, visual representations are necessary to facilitate the exploratory analysis and understanding of these interactions.

A clustering approach is the most popular method for discovering the global relationships among gene expressions or gene interactions [14, 21]. Even though it has the advantage of having relatively low computational cost, it cannot be used to investigate and predict the time-dependent gene networks and interactions [26]. While many different methods for DNA data analysis have been proposed (see details in Sect. 2), including several for microarray data, there is little work that has been done on detailed analysis of time-series DNA microarray data [49].

In this paper we present a new visualization-based framework with emphasis on temporal feature display and visual analysis on time-series microarray data. In particular, the framework applies a gene regulatory pathway (e.g., gene regulatory network) prediction method to predict the gene expressions over time. Although similar in nature to the combined models in [24, 25], our method applies a clustering method to group the gene pool into

a number of biologically-meaningful clusters. Then an auto-regressive (AR) model is used to relate the expression levels of each prototype in time t to the expression levels of other prototypes in previous times. The AR model is one of the broadly used linear prediction methods that predict the output of a system based on its previous values [12, 13].

Based on our combined model (a mixture of clustering and AR model), the visual analysis framework has been designed to support finding known and unknown gene interactions as well as understanding individual genes in detail (The overall layout of the visual analysis system is shown in Fig. 1). In the framework design, we define two steps: *visual representation* and *visual analysis*. Although both are broadly used terms in the visualization domain, we define them as follows: *visual representation* serves the major role of displaying visual entities in a visual space (2D or 3D) and *visual analysis* supports the user's interactive understanding of visualized entities and their relationships (see Sect. 5 for detail).

In the following sections, several design properties and techniques used in the framework will be described further:

– Biological pathways and prediction techniques used in our model,
– Representation methods on gene interactions and clustered pathway information over time,
– A novel overview and detail-view approach with which the user can employ interactive analysis to create new understandings of dynamic pathways, gene properties, and relationships,
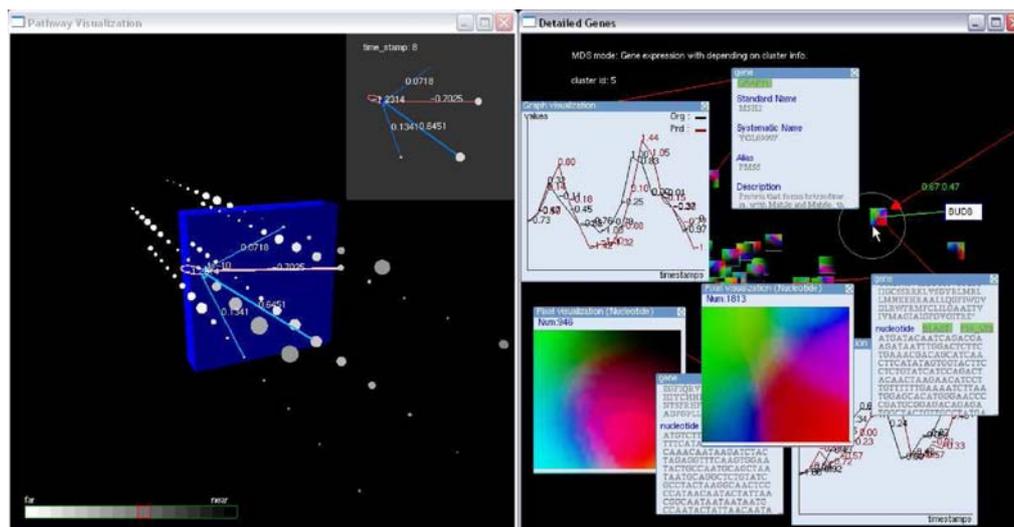

Fig. 1. The overall layout of the visual analysis system, which consists of two main windows: one (*left*) is for showing regulatory pathway information depending on time, and the other (*right*) is for providing interactive analysis of gene information using pixel-based and line-graph visualizations and textual annotations

– Design schemes used in the detail view to continuously reveal the patterns and relations using simple graph drawing and pixel-based representation techniques,
– An integrated analysis tool that can be launched and whose results can then be compared.

That this framework is both efficient and effective in the analysis of time-series microarray data as a complex multi-step process, is supported by the insights described in Sect. 8.

The rest of the paper is organized as follows. In Sect. 2, general knowledge about biological pathways is introduced. Section 3 provides a detailed description of existing microarray analysis methods as well as visualization techniques. In Sect. 4, a detail explanation on applied dynamic modeling of time-series microarray data is given. Section 5 gives an overview of our designed model. In Sects. 6 and 7, several interactive visual analyzing techniques are briefly described. Section 8 provides insights gained on some microarray data while using our visual analysis framework. Finally, conclusions and future work are offered.

2 Modeling biological pathways

In many biological systems and regulatory networks, gene expression levels measured across different time points during a given biological process provide insights into the underlying system. As such, more and more DNA microarray time-series are being used to determine a potential regulatory relationship among genes. Experimentations using DNA microarray chips provide large amounts of information on gene expressions data through mRNA transcripts. One of the challenges in microarray analysis is to discover the dynamic relationships and interactions among expressions. Estimating a gene regulatory network by processing DNA microarray time-series is a process in which microarray time-series data are examined to identify the transcriptional regulation relationships among various genes, especially considering the temporal patterns of gene expressions. Several statistical methods have been designed to address this problem. A common approach when analyzing the gene expressions is to combine the gene expression profiles into a single row vector such as $x_i = [x_{i1}, \ldots, x_{ij}, \ldots, x_{in}]$ where x_{ij} is the gene expression level at time j for gene i, and n is the total number of experiments in time. In gene-regulatory networks, there are two types of regulatory events among genes, activation and inhibition. Activation means that the expression level of the output gene is increased in the presence of the activator; while inhibition indicates that the expression level of the influenced gene is reduced. Even though it is important to understand gene-regulatory events in microarray analysis, it is difficult to find the regulatory events using existing methods. One promising direction for gaining such understanding in microarray analysis is to use visual analytical tools as described in Sect. 3.

3 Previous work

There are a few studies focusing on the temporal regulatory properties of microarray data. Some such approaches include: correlation analysis [14], Boolean networks [39], Bayesian networks [18], dynamic Bayesian networks [29], etc. These methods have proved to be useful in finding the regulatory network for certain applications. However, considering the small number of time steps in typical microarray time series and the large number of genes involved in many biological processes, the parameters calculated in these methods may not be reliable [49]. Instead of using computationally complicated approaches, several researchers addressed the efficiency of using signal processing models in understanding time-series microarray data [12, 39].

Although statistical methods and signal processing models are useful to understanding gene interactions and regulatory events, it is still difficult to utilize all features in microarray analysis. To address this limitation, several pathway visualization applications supporting the understanding of the functions of genes have been designed. The most broadly used applications are KnowledgeEditor [43], GenMAPP [11], GeneSpring [19], PathwayStudio (formerly known as PathwayAssist) [27], etc. KnowledgeEditor is useful to model and analyze biological pathways directly creating biomolecular network graphs in order to find molecular interactions, but it has no prediction capability. GenMAPP is freely downloadable and widely distributed. It visualizes gene expression data on maps representing biological pathways and groupings of genes. Commercial applications such as GeneSpring and PathwayStudio are efficient at permitting the user to create her own gene pathways and find interconnections between genes. Genespring in particular is designed to produce scatter plots as well as correlation values. Although these tools are useful for certain microarray analysis, they do not fully support understanding of gene interconnections in highly interconnected pathways or analyzing time-series microarray data. Because of this limitation, several researchers tried to design visualization applications supporting the analysis of time-series microarray data.

Craig et al. [9] introduced a technique which displays all continuous temporal features of microarray data. They designed a tool (called time-series explorer [10]), with which users easily can find similar and important patterns on time-series data. Peterson [30] addressed the difficulty of analyzing time-series microarray data by proposing a technique which changes the coordinate system using principal component analysis (PCA) in order to help

in observing dynamic behaviors of the data. Symeonidis et al. [41] designed a technique visualizing clusters of genes as a crossing-free graph. Even though their work is good in understanding and finding the important patterns on time-series microarray data, they have not addressed the needs of predicting the pathway values and analyzing gene sequences.

Due to the lack of suitable logical connections between existing analytical tools and the visualization tools, the use of these visualization tools has not proved as insightful as it could be. However, evaluation has been applied to provide the basic requirements in considering human computer interaction (HCI) methods for pathway visualization systems [37]. Unfortunately, many of such requirements have not been followed in existing visualization applications. Eisenstein [15] described several technical features that should be in microarray analysis applications and pointed out that "the major objective of microarray experiments is not to generate endless spreadsheets and scatterplots, but to produce data that can be used to formulate an understanding of biological events." In designing our visualization system, we have kept in mind all these requirements and guidelines in order to maximize the user's understanding of biological interactions.

As mentioned previously, we use a signal processing model by combining an auto regressive (AR) model [44] and a clustering method. In contrast to other methods, our model is useful in analyzing and predicting the dynamic behaviors of time-series microarray data. Even though the method is designed to predict the signal patterns and dynamic interactions, understanding and analyzing the information produced by the model may not be straightforward. In particular, concepts such as cluster prototypes, time steps, and excitatory/inhibitory interactions need to be effectively presented to users [49]. To address these needs, we use visualization techniques to find important patterns, preserve temporal features of time-series microarray data, and analyze gene expressions interactively.

4 A dynamic model

To develop a dynamic model of gene regulatory interactions, we have created a combined technique consisting of auto regressive (AR) modeling [44] and a clustering method. Since almost all biological pathways are composed of a large number of genes, the approach of applying the AR model directly to the individual genes is not computationally feasible. That is, the number of genes in a biological process is often so large that it is impossible to develop reliable AR models for the typically short time-series microarray data while directly incorporating all genes as individual AR variables. To address this issue, we exploit the fact that many genes behave very similarly in the biological sense and therefore can be clustered together before dynamic modelling. As a pre-processing

step, K-means clustering is used to group the gene pool into a number of biologically meaningful clusters. Each of these gene clusters is represented by a prototype that reflects the overall time trends of the cluster. After the preprocessing step, the AR model is applied on each cluster as opposed to individual genes. Since the number of clusters is small enough, the AR model can be reliably developed for the prototypes. The model relates the future expression level of the prototype clusters to the values of the prototypes in past time step (s). The model also considers uncertainty inherent to the model by considering a noise factor (e). In its most general form, the model is a linear system of difference equations:

$$
\begin{aligned}
y_i(t) = &-a_{i11}y_1(t-1) - \cdots - a_{i1n_{1i}}y_1(t-n_{1i}) \\
&- a_{i21}y_2(t-1) - \cdots - a_{i2n_{2i}}y_1(t-n_{2i}) \\
&\cdots \\
&- a_{ip1}y_p(t-1) - \cdots - a_{ipn_{pi}}y_1(t-n_{pi}) + e_i(t),
\end{aligned}
$$

where $y_i(t)$ is the expression level of prototype i at time t, n_{ji} is the maximum time span of the interactions between prototype of cluster i and prototype of cluster j, coefficients a_{ikj}'s are the parameters of the model, and $e_i(t)$ is the noise factor.

In this paper, we use a dataset containing the time-series expression values of approximately 200 genes involved in the cell cycle of the budding yeast *S. Cerevisiae* [7]. The gene expression values were collected in 17 time points. It is known that there are five major phases in cell cycle development: early G1 phase, late G1 phase, S phase, G2 phase and M phase. In each phase only genes whose biological functions correspond to the changes occurring in that phase are active. Based on the known biologically distinctive functions of these five phases, it is reasonable to cluster the genes into five clusters. The results show that the model can very accurately predict the expression values of almost all genes in the future steps (see [12] for detail).

The main application of this dynamic modeling method is twofold. First a dynamic regulatory network governing the quantitative interactions among the prototypes of the main biological trends can be obtained, i.e., the model discovers the effects of each gene group on itself and on other groups in time. Secondly, by using the resulting dynamic network, the expression level of each gene at time t can be predicted based on its expression level and expression level of other genes.

5 Designing a visualization model

We define several major steps to be followed and use them to form our system (Fig. 2).

In microarray data analysis, one must find the biological meanings of the discovered genes. To support find-
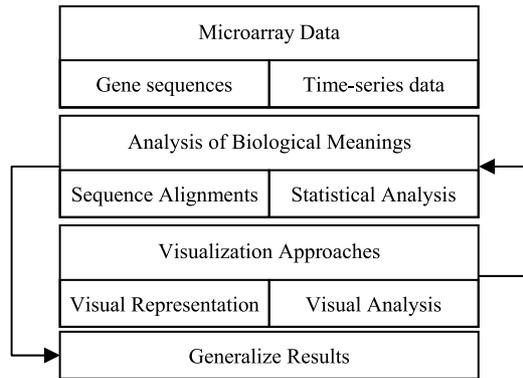
Microarray Data	
Gene sequences	Time-series data

Analysis of Biological Meanings	
Sequence Alignments	Statistical Analysis

Visualization Approaches	
Visual Representation	Visual Analysis

Generalize Results

Fig. 2. Major steps in visual analysis of time-series microarray data

ing the biological meanings, two different approaches, gene sequence analysis and statistical analysis, are typically used. Sequence analysis is performed on the gene sequences to find similar sequential patterns or regulatory sequences, some of which have been previously studied and annotated. Statistical analysis is done using microarray data to identify the regulatory networks or cyclic genes. While statistical analysis can be performed alone without understanding the gene sequences, it is still necessary to consider the gene sequences because sequence analysis can assist in finding the most related genes (regulatory sequences) to the biological system and provide the molecular basis for understanding the detailed gene interactions. Therefore, gene sequence analysis and statistical analysis on microarray data often have to be performed together to understand gene function in detail [40]. This process is usually followed iteratively until relevant knowledge is found. The knowledge must then be generalized to help other research scientists understand fully the meaning of the genes; identified cyclic genes, regulatory events of activation or inhibition, etc.

In developing the visual analysis framework, we design methods supporting each analysis in Fig. 2 and the overall iterative process. In particular, we define two steps, *visual representation* and *visual analysis* in our system.

Visual representation is a preliminary step which can provide the user a general understanding of biological pathways in display space. Since biological pathways are highly complicated, we use both *2D and 3D representations*. These representations are designed for interactive analysis. In the 3D view, the analyzed data (clustered pathway information) are directly mapped to the visual attributes, while interactive analysis of the genes is shown in the 2D view. Temporal attributes in time-series microarray data are carefully managed by preserving clustered gene expressions in the 3D view as well as each gene's individual expressions in the 2D view. The major advantage of using a 3D representation is to increase the possibility of integrating additional visual relations and information into the representation [42]. *Animation* is added to

maximize efficiency and create a better understanding of data [33] because it creates an additional display dimension [48]. However, 2D representation has an advantage of understanding a gene's detailed biological meanings. *Pixel-based gene representation* and *positioning* are applied to support discriminating a gene's properties and understanding its relationships respectively.

Visual analysis is necessary to support exploration of interactions among clusters, to increase the capabilities of both exploratory and focused analyses of gene interactions, and to overcome the limitation of cluttered displays. Four different *visual interactions* are used to facilitate analysis: *labeling*, *detailing*, *navigation*, and *focusing & linking*. *Labeling* displays a simple description in areas of focus. It helps in discriminating the information in highly cluttered display. *Detailing* interactively provides the user with enriched knowledge when requested. *Navigation* gives the user the ability to move through the visual layouts at different scales or extents. *Focusing & linking* allows comparison between genomic data to find similar patterns [6].

Based on the above considerations, the visual interface is designed as shown in Fig. 1. It has two windows. The user starts with exploration of the cluster time series in the 3D view. As the user focuses on time steps, clusters, or individual genes, the 2D view is dynamically updated. The user moves back and forth between these windows as she explores and analyzes. Detailed descriptions of visualization approaches as well as this highly interactive, iterative process are provided in following sections.

6 Visual representation

6.1 Visualization of clustered gene expressions

Most microarray visualization applications use a visual form (e.g., glyphs) to represent abstract information. But there are limitations to what can be accomplished due to limited display space and the scale and richness of detail of the data [37]. Several information visualization techniques are relevant to this problem, such as Overview+detail [42], Focus+context [31], and Pad++ [3]. Multi-scale visualization techniques are appropriate to manage large and complex microarray data [37]. Following this path, our framework uses multi-scale visualization [20] to effectively and efficiently manage and increase understanding of the large-amount of microarray data. In this section, we will explain in detail how we use this framework for displaying microarray data.

A 3D representation method is used to show the time-series microarray data. However, since the 3D display can produce clutter, the selected gene expression values are laid out simultaneously in 2D (Fig. 3). The 3D representation is efficient in showing the gene expressions at the successive time step where the current expression level of
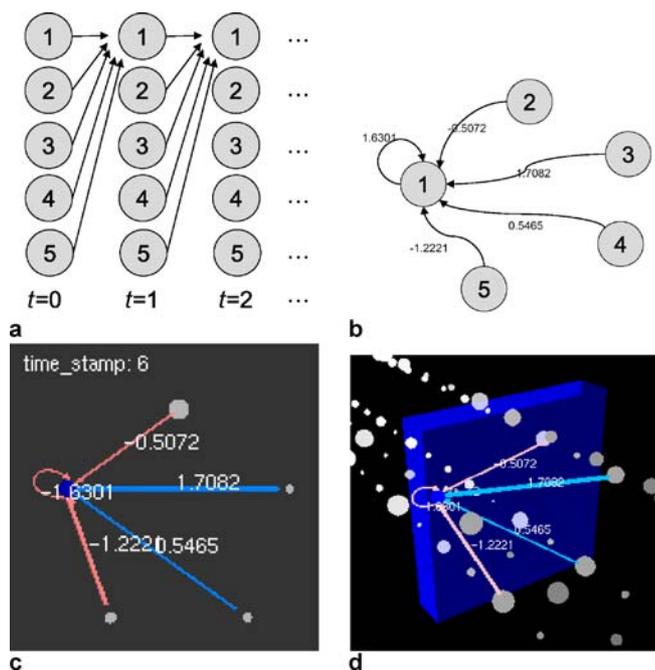
Fig. 3. a Sketch of time-series gene networks for analyzing cluster number 1; **b** static graph prototype at 6th temporal point; **c** direct mapping onto 2D; **d** 3D mapping in context of all time steps

prototype i at time t has a regulatory relation (activation or inhibition) with the prototype j at time $t-1$. Each prototype is regarded as a cluster and represented as a 2D circle or 3D sphere respectively in 2D or 3D view. The perspective view in 3D representation permits the user to maintain an overall context in time while exploring dynamic interactions in detail.

The radius of the cluster represents the normalized mean value of the clustered signal pattern. Thus a bigger sphere or circle represents stronger overall contributions of the cluster to the regulatory network at that time step. Also each cluster makes use of a color coding [38] to enhance visualization of cluster values. When selected, a cluster changes color to blue. Transitional arrows colored blue indicate positive transitional pathways (*activation*) between cluster spheres, and arrows colored red indicate negative transitions (*inhibition*). That is, the transitional arrows show controls of one cluster over another with blue indicating enhancement of gene expression and red indicating suppression. The self-transitional loop indicates a control of the cluster on itself. With this simple palette, users can quickly discern important pathway differences and the state of the regulatory network over time. The user can also select individual arrows for quantitative information.

As shown in Fig. 1 (left), the layouts are accompanied by a temporal slider bar at the bottom left that supports animation corresponding to time-series steps, in particular dynamic animation of cluster pathways over time. The slider bar ranges over all time steps in the dataset. In the bar, the time series is colored along a gray scale from white (17th time step) to dark gray (1st time step). As Wright [48] points out, animation can be an additional dimension to support displaying and analyzing information datasets in 3D view. Also, it provides a natural mapping between time and saturation that allows the user to identify positions in the temporal space while permitting color to be used for other information display. When the user makes a selection on the bar, a thin blue semi-transparent box appears at the selected time step. If the user drags the mouse along the slider bar, the box moves accordingly along the time steps. Alternatively, the user can grab and drag the box directly. In either case an animation showing changes over time is produced. In addition, selections can be made within the box (e.g., a cluster is selected and its transitional arrows displayed). These selections are then maintained at later or earlier time steps as the box is moved around. By employing this animation capability, the user can quickly analyze and compare behavior in time with minimal effort [5, 33].

6.2 Visualization of individual gene expressions within clusters

As mentioned earlier, finding biological meanings is important in microarray data analysis. Even if the visual layout described above (Sect. 6.1) gives the user a quick overview of the dynamics of the gene regulatory network, significantly more details (gene sequences, interactions, regulatory behaviors, etc.) are needed to understand each gene. In general, gene (or protein) sequences range about a few thousand units in length. To manage data for a single protein sequence and provide contextual pattern information in a useful way, a pixel-based gene representation method is used. To support a better understanding on gene interactions and regulatory behaviors, MDS [1] is a useful technique to effectively represent high dimensional data in lower dimensional space. In display space, each gene is positioned depending on its interactions and regulatory behaviors. All these representations are laid out in their own sub-windows in the right window of Fig. 1. The user can select and bring forward the particular sub-windows desired for current exploration. Additionally, several interactive analyses are provided to increase the understanding on time-series microarray data and gene interactions. In this section, techniques used for designing this framework are described in detail.

6.2.1 Pixel-based gene representation

The pixel-based gene representation is designed to reveal the features of gene sequences. To map from gene sequences to pixels, space-filling methods [22] are quite useful because they produce spatial patterns that have consistent locality, even for long gene sequences. Several
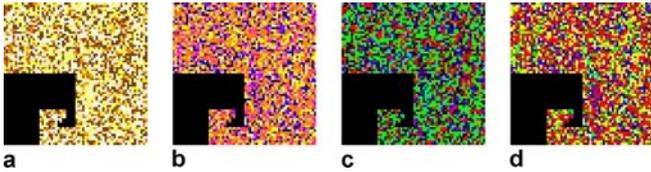
Fig. 4a–d. Pixel-based gene (*SWItching deficient (SWI4)*) representation with several different color codings; **a** A (*white*), C (*yellow*), G (*orange*), T (*dark brown*) [47], **b** A (*orange*), C (*blue*), G (*purple*), T (*yellow*) [36], **c** A (*green*), C (*blue*), G (*black*), T (*red*) [34], and **d** A (*red*), C (*blue*), G (*green*), T (*yellow*) [32]. *Black space* located in *left bottom* of each image represents the empty part of the space-filling curve

methods of this type have been designed. Here we use a Hilbert curve ordering method [4] to arrange sequence information mapping with color information, since it has the advantage of providing continuous curves while maintaining good locality of sequence information.

For mapping with gene sequences, we set the Hilbert curve order to 12 which covers $2^{12} \times 2^{12}$ sizes of gene sequences. Color coding is then used to represent the sequence information. DNA is a linear polymer made up of sequences of four nucleotide bases: adenine, guanine, cytosine, and thymine – designated A, G, C, and T. Gene regulatory pathways can involve hundreds or more genes from which different proteins can be expressed. Hence, two different color mapping approaches have been made, one for the gene sequence and the other for the expressed protein. Originally, pixel-based visualizations on large scale of data were pioneered by Keim et al. [22, 23]. And a pixel-based gene representation method has been suggested by Wong et al. [47]. But they did not concern themselves with finding an efficient color coding for the gene sequences.

For determining the correct color codes, four commonly used color maps used by other researchers were tested on the gene transcription complex *SWItching deficient (SWI4)*[1], a protein involved in the budding yeast *S. Cerevisiae*, in order to find the best-mapped color codes. All images in Fig. 4 are generated using Hilbert curve ordering.

Figure 4 shows mapped images with different color codes. But it is difficult to understand clearly the represented patterns and features in the sequence. Therefore, a pixel enhancement technique (initially proposed by Wong et al. [47]) is applied in order to find the best color mapping method. The pixel enhancement technique consists of three steps. First, a Gaussian filter is used to smooth the high-frequency values. And then histogram equalization is applied to modify the dynamic range and the contrast of an image depending on color channels (for example, R, G, and B channels). Finally, saturation values

are increased using extrapolation as a saturation adjustment technique.

Even though all pixel-enhanced images have similar results (Fig. 5) with respect to the patterns in terms of R, G, B color channels, the pixel representations in the 3rd column of Fig. 5, using color codes from (c) in Fig. 4 (A (green), C (blue), G (black), T (red)), show consistent patterns in all 3 channels. Furthermore, merging all channels and applying saturation adjustment (Fig. 5e and f) show in clearest detail which area of the image has denser information for adenine, guanine, cytosine, or thymine, and what its shape is. We thus apply the result of Fig. 5f in the following. To arrange the gene images in 2D view, we must next apply MDS, as discussed next.

It is important to note that this is a simple way to construct gene images. More sophisticated methods, which might take into account sequence structure and meaning,
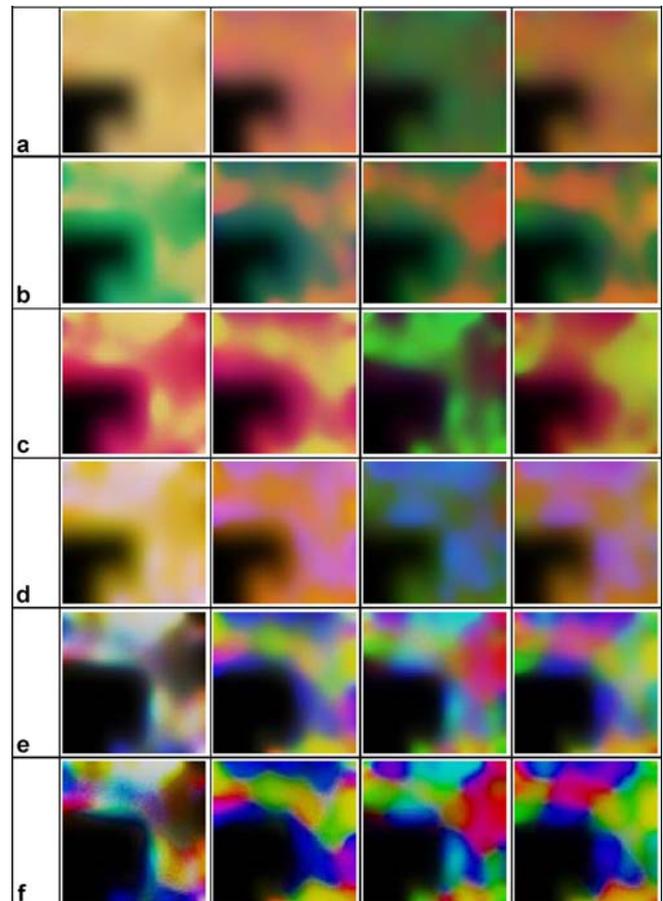

Fig. 5a–f. Rows represent several image processing filters and a pixel enhancement method are applied to pixel maps shown in Fig. 4 (represented by columns). **a** Gaussian filtering, **b** histogram equalization on red channel, **c** histogram equalization on green channel, **d** histogram equalization on blue channel, **e** merged all color channels, **f** saturation adjustment is applied with the saturation value of 2.5

---

[1] SWI4 acts as a transcriptional activator to regulate late G1-specific transcripts in budding yeast required for DNA synthesis and repair.

could be applied. However, the pixel mapping would be the same.

6.2.2 Data positioning in 2D View

In our framework, we use nonmetric multidimensional scaling [45]. Based on this method, the dimensions of time-series data (17 dimensions of timestamps) and gene sequences (4 dimensions of nucleotide bases) are changed to two dimensions.

In each case, the first step is to find a monotonic relationship between the dissimilarities. The relationship is typically found using isotonic regression. In our application, we use Kruskal's nonmetric phase [45] as isotonic regression. For the case of gene sequences, we opted for a simple approach that references the numbers of nucleotides when finding the relationship. First, gene sequences are counted in terms of adenine, guanine, cytosine, and thymine. Based on the counts, the relationship is determined. Finally, all gene sequences are mapped onto 4 dimensional distance matrixes and MDS is applied to map pixel-represented genes (glyphs) to a 2D display space. Even though more detailed measures of relationship on gene sequences could be applied, this approach is quick and gives a rough idea of contextual similarity.

Two different MDS representations are considered with respect to the time-series data. One representation shows local differences among genes in the same cluster group (Fig. 6a), while the other shows global differences among all genes in all the clusters (Fig. 6b). Selections in Fig. 6a are then highlighted in Fig. 6b. In this way the user always has a view available that emphasizes local and global changes over time. The user can then switch attention to a more detailed view for a selected time step (Fig. 7), where the circles are replaced by glyphs.

In Fig. 7, two different glyph forms are used, the pixel-based representation of gene sequences and a simple line graph representing the time-series microarray information. As described above, our visual analysis method is closely tied to the dynamic pathway prediction model. The line graph display for each gene, where the gene expres-
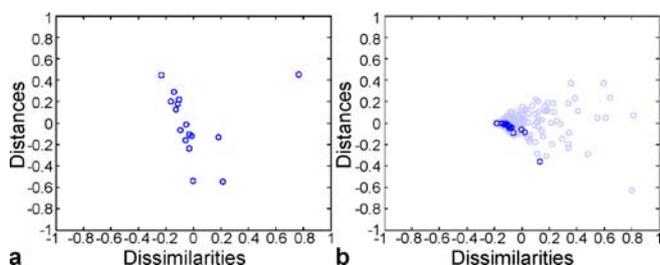

Fig. 7a–d. Glyph representations of **a**, **b** pixel-based sequences and **c**, **d** graph-based DNA microarray time-series information. Both local analysis (**a**, **c**) and global analysis (**b**, **d**) are given. *Highlighted glyphs* in **b**, **d** indicate the genes located in the selected cluster with respect to other genes

sion level is plotted along the vertical axis and the time steps along the horizontal axis, provides an option that emphasizes each gene's time history. Distances between genes are then measured according to the similarity of their line graphs. Hence, there are 4 possible ways to construct gene expression patterns, as indicated in Fig. 7.

From Fig. 7a and b, it is seen that spatial distributions of the local and global gene representations are not much different. But the spatial distribution of the graph-based representations (Fig. 7c and d), show major differences between local and global analysis, with the latter case producing highly clumped results. It is thus clear that clustered data should be analyzed for both local and global patterns, considering both gene sequence (pixel-based glyphs) and time series (line graph representations), since these bring out different aspects that can be used to bring out differences and similarities in gene structure, function, and regulatory behavior. This point is illuminated further in Sect. 7. Visual analysis methods discussed next enrich these capabilities.

Fig. 6a,b. Local similarities of time-series data within a cluster group (**a**) and global similarities of all time-series data (**b**). Each *circle* indicates time-series data of each gene and the *highlighted circles* in **b** indicate the data having the same clustered group in **a**
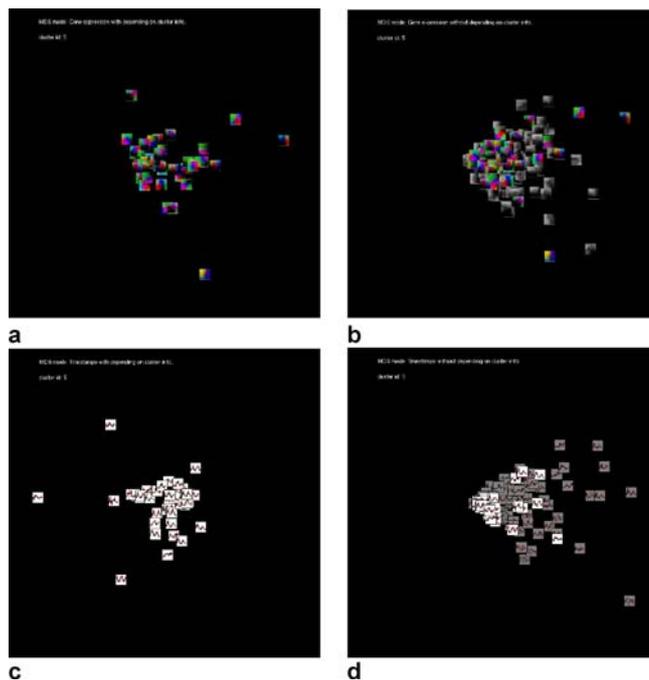
7 Visual analysis

Interaction is a very useful exploration and discovery tool for large scale or complex data. In our visual analysis framework, we apply several interaction methods such as

labeling, *navigation* (panning, zooming, and scaling), *detailing*, and *focusing & linking*. In this section, the interaction methods are described.

Labeling. To heighten the knowledge content of the glyph representations in Fig. 7, we need to have a way of showing brief annotations for specific genes such as scientific name, gene expression values, regulatory events, etc. The method of excentric labeling has been explored for textual labeling [16]. It uses the technique of directly attaching a focus region to the cursor, and annotations are shown whenever the cursor is passed over glyphs. Although several different excentric labeling techniques have been proposed [16], there are still some drawbacks with respect to *cluttering* due to annotations. Also, the method is not able to show the gene regulatory events, and context may be lost since the annotations appear and disappear as the cursor is moved. Therefore, we have designed a modified labeling technique which adopts the position-shifting operation [8] to show textual information without *cluttering*. Manual position shifting is a broadly used technique because it is especially useful in controlling labels in dense information visualizations. But it requires user efforts to manually shift the annotations. Instead of using the manual operation, our approach uses an automatic shifting method. The size of the overlapped region between labels is measured, then, using a strength based on this size, each label has a repulsive force against any overlapping label, so that the labels are pushed apart. This reduces the need for user attention and effort and permits the high value annotation knowledge to flow unimpeded to the user.

The labels can contain a variety of information. Here the labels contain gene names and, in the case of the line graph representation, arrows indicating the point in the line graph corresponding to the current time step (Fig. 8), which supports analyzing the microarray time-series details and relations.

Additionally, the labeling technique supports changing the size of the focus region centered on the cursor, with which it is useful for showing labels when the scale of the glyphs is modified or when the user is interested in a larger or smaller region. The effective rearrangements and smooth transitions caused by the force fields help the
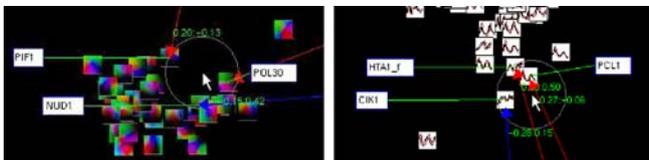
user maintain context and keep track of new annotations even under continuous cursor movement. All these capabilities are incorporated in the navigable view (Fig. 9).

Navigation. Zoom and pan navigation within the detail view is designed based on the "Pad" [28] metaphor and its extension, Pad++ [3, 17]. In this metaphor, the visual space is considered as an infinite 2D plane (called Pad), which can be stretched by orders of magnitude at any point to investigate details. In our previous design of a genomic visualization system, GVis [20], we found that the technique provides an important capability for finding details and relations at all scales within a context of thousands of genomes. By using the Pad++ metaphor, a user can easily compare objects' patterns and find their differences over the whole object space and at multiple scales through successive pans and zooms. Figure 9 shows zoomed-out and zoomed-in navigation states. When the glyphs are in the zoomed-out state, we can find the overall arrangement of the gene expressions with respect to each other. Upon zooming in, we can take a close look at the various glyphs to find complete details in the gene expressions.

Scaling is included in the navigation interactions and is quite useful since it shows the glyphs at larger or smaller
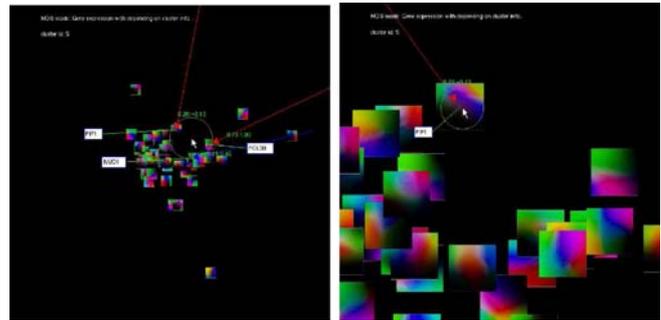

Fig. 9. Navigation in the zoomable space provides the capability for smoothly finding and examining interesting data objects in overview (*left*) or close-up (*right*)
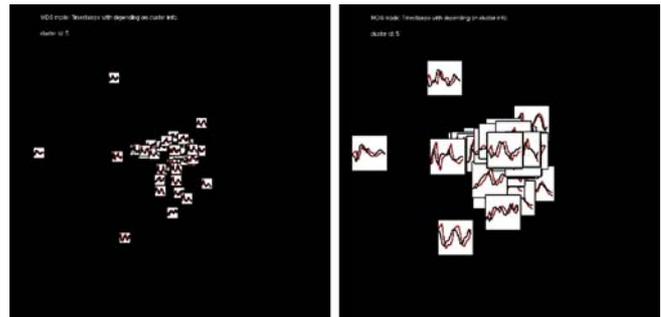

Fig. 8. A modified excentric labeling provides labels in a focus region. The focus region dynamically changes its location while the cursor moves over the display. Labels of the objects located in the focus region are updated smoothly and dynamically depending on the focus region

Fig. 10. Scaling makes it possible to see the detail of glyphs without disrupting their location information

scale without changing their positions or the viewpoint and retaining the overall pattern. Scaling can avoid or reduce overlaps, or increase glyph sizes to show details, as shown in Fig. 10. Even though navigation and scaling work well in avoiding overlaps in the display space, there are still cluttering problems where elements are highly correlated and thus close to one another. To minimize this problem, we permit the reordering of overlapped glyphs in display space. Selected glyphs can be moved to the front or back with respect to the user's viewing perspective. In this way a clear view of the details of a glyph can always be obtained.

Detailing. Since genomic data may contain large amounts of knowledge on sequence, active sites, expressed proteins, etc., it is necessary to present this information as needed. For best focus and efficiency, we provide these annotations in detailing windows within the display and attached to the relevant gene glyphs. These pop up in a "details on demand" mode, upon selection (see Fig. 11). Each window has a standard format for the text record of the item, including the standard name, the original DNA reference name, the sequence start and end, the gene's function, the proteins it creates, and so on (Fig. 11). By panning or scrolling within the pop-up window, text of any length is accessible to the user. The windows align themselves dynamically in the 2D space, near the glyphs they represent, so that overlap is minimized as several boxes are opened. Each window can be positioned into another location by dragging it. It also can be scaled up or down to show the text information more clearly or to avoid overlaps.

Focusing & linking. To support gene analysis, the pop-up window has buttons for pixel-based visualization, line-graph visualization, and BLAST. The BLAST button launches the sequence analyzing tool called BLAST (the Basic Local Alignment Search Tool), as described further below.

Figure 12 shows that this provides a fast and comprehensive view of the available gene sequence information permitting, for example, comparison between pixel-based and line graph views for any gene or gene collection. This close comparison can extend to two or more gene sequences.

As mentioned above, our application incorporates a widely used sequence matching tool, BLAST, which detects relationships among sequences that share only isolated regions of similarity [2]. The National Center for Biotechnology Information (NCBI) provides BLAST utilities such as network-based BLAST, stand-alone BLAST, etc. When the BLAST button is clicked, network-based BLAST is launched and searches for similar alignments in the NCBI database. Depending on the option selected, results are provided either in terms of alignments with the selected nucleotide sequence or its expressed proteins, which are then displayed in the pop-up window (Fig. 13).
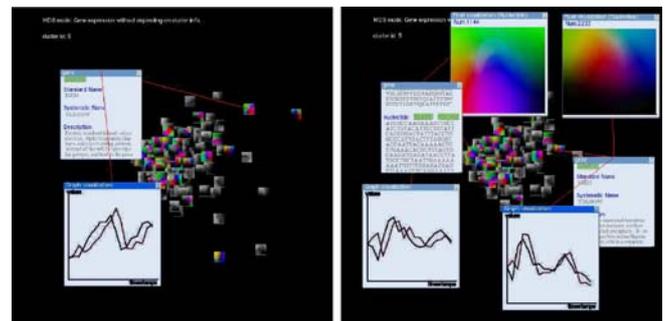

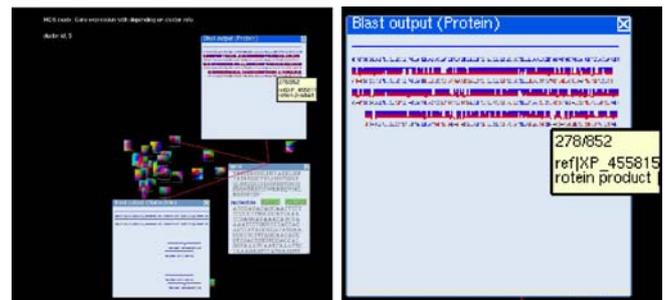**Fig. 12.** Pixel-based gene representations and line-graph visualization are used at the same time when analyzing genes

Fig. 13. The BLAST outputs are shown with protein and nucleotide sequences (*left*); a close-up of BLAST output (*right*) shows similar sequences aligned for comparison. The window has a capability of scaling the output to show in detail

Fig. 11. Additional information will be given for further understanding of the selected genes

The outputs are aligned in a graphical representation that permits the user to get overviews or detailed comparisons by zooming in or out. This is especially useful when finding regulatory elements among genes that were indicated in the time series line graphs.

To complete the interactive analysis framework, we have incorporated the GVis system [20]. GVis supports in-depth study of the structure and function of the genomes, genes, and their expressed proteins involved in the dynamic regulatory pathways. This study provides the user with a natural, more detailed follow-on to the fast exploratory analysis carried out with the above interactive tools. Thus GVis automatically takes genes selected from the above analysis and provides detailed comparison of their fully annotated genome environments with any other genome environment (which may be selected by iterative applications of BLAST or other comparative analysis tools). GVis is capable of permitting interactive exploration of tens of thousands (or more) of genomes from overviews down to the level of the annotated nucleotide sequences.

8 Biological insights

In this section, we explore what kinds of biological insights can be gained when using our framework by presenting some results from visual exploration of the microarray analysis of gene regulation in budding yeast *S. Cerevisiae*.

The user starts by exploring the time series data using the left side of the interface in Fig. 1. When she focuses on particular genes, the predicted regulatory pathways are displayed on the right side of the interface in Fig. 1 in order to provide a detailed view of dynamic gene regulation. As the user zooms in, the pathway values are automatically shown when the scale of the window is large enough to display them on screen. Figure 14 shows
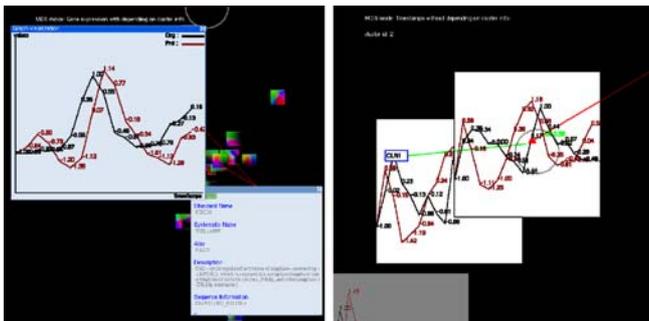

Fig. 14. Two different ways of representing methods: gene expressions of CDC20 (*cell division cycle*) in cluster 5 (*left*) and CLN2 (*CycLiN*) in cluster 1 (*right*). *Red* indicates the predicted values and *black* represents the original data
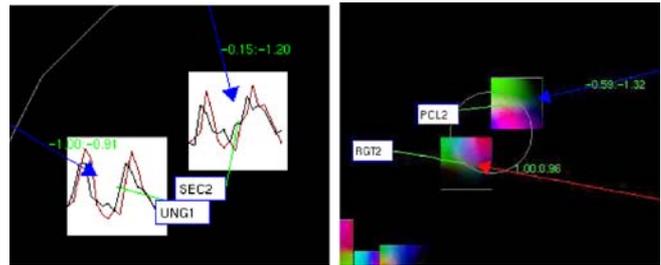

Fig. 15. Similar gene expressions (*left*) and sequence information (*right*) are positioned close to each other in visual space

successfully predicted regulatory pathways alongside the actual observations. This demonstrates that the predicted pathways can be used even in the absence of the observations. In the predicted pathways, it is important to note that there is a one-step delay in computing the effect of all prototypes on time-series microarray data (i.e., the prototypes determine the expression value of the cluster prototype 1 in the next time point). Thus the predicted curve is shifted with respect to the observations.

In the framework, we provided two different distance-based visualization approaches, direct mapping with cluster information and data positioning with multidimensional scaling. As she explores further, the user brings up these distance-based visualizations, which provide useful insights as described next.

Figure 15 shows two different visual layouts of gene expressions. Even though the genes are nominally different from each other, the visualization indicates that the genes in question have similar pathways or patterns. This is important knowledge for directing further investigations. In the left-hand case, we find there is a close-relation between the genes SEC2 (*SECretory*) and UNG1 (*Uracil DNA N-Glycosylase*). It is known that SEC2 and UNG1 have the same upstream[2] of the ATG start codon [46]. On the other hand, in the right-hand case the genes PCL2 (*PHO85 CycLin*) and RGT2 (*restores glucose transport*) have different molecular functions even though they have similar sequences [35].

This example shows how interactive visualization supports rapid and efficient exploration of the data followed by zooming in on subtle but important similarities and differences. The user can then build on these insights by comparing with other regulatory behavior, looking closely at the annotated gene sequencing or protein expressions, as in Figs. 11–13, or launching BLAST to bring up other annotated genes or proteins for comparative analysis.

Finally, we did some preliminary evaluations with bioinformaticists that found some strengths and weakness about the application. Strengths are (1) the capability of

---

[2] Each strand of DNA or RNA has a 5' end and a 3' end. Upstream is the region towards the 5' end of the strand relative to the position on the strand.

using both sequence analysis and pathway analysis in the microarray data, (2) the ability of discriminating the gene interactions in terms of local similarity and global similarity, and (3) predicting the pathway values in future time-stamps. Especially it has been noticed that sequence analysis while doing pathway analysis might usefully lead to further analysis on the gene itself. But further study has to be performed in determining the full use of sequence analysis in pathway analysis. Even though predicting pathway values is quite useful to drug discovery, fully understanding the concept of predictions in 3D and 2D visual layouts is difficult. Also 3D representation is useful to understand the concept of pathway analysis, but it has some limitation in the manipulating of the 3D visual layout with a 2D mouse.

9 Conclusions and future works

In this paper, we designed an interactive analysis framework, with which the user can develop understanding of the dynamic regulatory pathways among genes by using visual analysis coupled with a prediction method. The framework implements a powerful integrated analysis that supports both understanding of the gene interactions over time and the understanding of gene function and structure (through comparative analysis). To support the analyzing procedures, we developed visual analyses in terms of three design steps, *visual representation* and *visual analysis*. In the framework, several interactive analysis features are provided: time-based cluster visualization, pixel-based visualization, simple line-graph layouts, multi-layered navigation tools, and sequence analyzing features. With these features, a user can quickly and effectively move among different perspectives to build an understanding of the time series structure, the gene interactions, their annotations, and their functional meanings. We have given a brief example of how these biological insights can be obtained for a particular gene regulation analysis. This integrated approach is quite important because it supports what the analyst must do anyway and, up to now, has had to do laboriously without an integrated set of tools.

Our future work is to extend the application to display correlative analyses of genomic data in a more comprehensive way. When analyzing time-series microarray data, referencing existing literatures is always necessary to unveil and fully understand gene interactions. Therefore the visualization application should have a feature that provides an overview of relevant literature [37]. For this reason, we plan to automate an exploratory literature reviewing process in our application. Finally, although our team includes bioinformaticists and the visual analysis framework has been developed, tested, and used with their help, we will offer the framework to a wider group of bioinformaticists for their use. As part of this, we are planning thorough evaluations and comparative testing.

References

1. Agrafiotis, D.K., Rassokhin, D.N., Lobanov, V.S.: Multidimensional scaling and visualization of large molecular similarity tables. J. Comput. Chem. **22**(5), 488–500 (2001)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman D.J.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
3. Bederson, B.B., Hollan, J.D.: Pad++: A zooming graphical interface for exploring alternate interface physics. In: UIST '94, pp. 17–26. ACM, Marina del Rey, CA (1994)
4. Breinholt, G., Schierz, C.: Algorithm 781: Generating Hilbert's space-filling curve by recursion. ACM Trans. Math. Software **24**(2), 184–189 (1998)
5. Brown, J., Mcgregor, A., Braun, H.W.: Network performance visualization: insight through animation. In: PAM2000 Passive and Active Measurement Workshop, Apr, pp. 33–41. Hamilton (2000)
6. Buja, A., McDonald, J.A., Michalak, J., Stuetzle, W.: Interactive data visualization using focusing and linking. In: IEEE Conference on Visualization '91, pp. 156–163. IEEE Computer Society, San Diego, CA (1991)
7. Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., Davis, R.: A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell **2**(1), 65–73 (1998)
8. Chuah, M.C., Roth, S.F., Mattis, J., Kolojejchick, J.: SDM: malleable information graphics. In: Information Visualization (INFOVIS '95), Oct, pp. 36–42. IEEE Computer Society, Atlanta, GA (1995)
9. Craig, P., Kennedy, J., Cumming, A.: Towards visualising temporal features in large scale microarray time-series data. In: Sixth International Conference on Information Visualisation, pp. 427–433. IEEE Computer Society, London (2002)
10. Craig, P., Kennedy, J., Cumming, A.: Coordinated parallel views for the exploratory analysis of microarray time-course data. In: Third International Conference on Coordinated and Multiple Views in Exploratory Visualization, pp. 3–14. IEEE Computer Society, London (2005)
11. Dahlquist, K.D., Salomonis, N., Vranizan, K., Lawlor, S.C., Conklin, B.R.: Genmapp, a new tool for viewing and analyzing microarray data on biological pathways. Nature Genetics **31**(1), 19–20 (2002)
12. Darvish, A., Hakimzadeh, R., Najarian, K.: Discovering dynamic regulatory pathway by applying an auto regressive model to time series DNA microarray data. In: 26th Annual International Conference of the IEEE/EMBS (2004), pp. 2941–2944. IEEE Computer Society, San Francisco (2004)
13. Darvish, A., Najarian, K., Jeong, D.H., Ribarsky, W.: System identification and nonlinear factor analysis for discovery and visualization of dynamic gene regulatory pathways. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 76–81. IEEE Computer Society, San Diego, CA (2005)
14. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of

genome-wide expression. Proc. Natl. Acad. Sci. USA **95**(25), 14 863–14 868 (1998)

15. Eisenstein, M.: Microarrays: Quality control. Nature **442**, 1067–1070 (2006)

16. Fekete, J.D., Plaisant, C.: Excentric labeling: Dynamic neighborhood labeling for data visualization. In: Human Factors in Computing Systems (CHI'99), pp. 512–519. ACM, Pittsburgh, PA (1999)

17. Furnas, G., Bederson, B.B.: Scale space diagrams: Understanding multiscale interfaces. In: Human Factors in Computing Systems (CHI'95), pp. 234–241. ACM, Denver, CO (1995)

18. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian Network to Analyze Expression Data. J. Comput. Biol. **7**, 601–620 (2000)

19. GeneSpringTM. Silicon Genetics. http://www.silicongenetics.com

20. Hong, J., Jeong, D.H., Shaw, C.D., Ribarsky, W., Borodovsky, M., Song, C.: Gvis: A scalable visualization framework for genomic data. In: Eurographics/IEEE VGTC Symposium on Visualization (EuroVis 2005), pp. 191–198. Eurographics, Leeds (2005)

21. de Hoon, M.J.L., Imoto, S., Miyano, S.: Statistical analysis of a small set of time-ordered gene expression data using linear splines. Bioinformatics **18**(11), 1477–1485 (2002)

22. Keim, D.A.: Designing pixel-oriented visualization techniques: theory and applications. IEEE Trans. Vis. Comput. Graph. **6**(1), 59–78 (2000)

23. Keim, D.A., Kriegel, H.-P., Seidl, T.: Visual feedback in querying large databases. In: IEEE Conference on Visualization '93, pp. 158–165. IEEE, San Jose, CA (1993)

24. Liu, X., Minin, V., Huang, Y., Seligson, D.B., Horvath, S.: Statistical methods for analyzing tissue microarray data. J. Biopharmaceutical Stat. **14**(3), 671–85 (2004)

25. Moser, R.J., Reverter, A., Kerr, C.A., Beh, K.J., Lehnert, S.A.: A mixed-model approach for the analysis of cDNA microarray gene expression data from extreme-performing pigs after infection with Actinobacillus pleuropneumoniae. J. Animal Sci. **82**, 1261–1271 (2004)

26. Nakahara, H., Nishimura, S., Inoue, M., Hori, G., Amari, S.: Gene interaction in DNA microarray data is decomposed by information geometric measure. Bioinformatics **19**(9), 1124–1131 (2003)

27. PathwayStudioTM. Ariadne Genomics. http://www.ariadnegenomics.com

28. Perlin, K., Fox, D.: Pad: An alternative approach to the computer interface. In: ACM SIGGRAPH '93, Aug, pp. 57–64. ACM, Anaheim, CA (1993)

29. Perrin, B.E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., D'Alche-Buc, F.: Gene networks inference using dynamic Bayesian networks. Bioinformatics **19**(Suppl.2), II138–II148 (2003)

30. Peterson, B.: Dynamic Visualization of Microarray Time Series. In: white paper, http://www.terastat.com (2005)

31. Pirolli, P.L., Card, S.K., Van Der Wege, M.: Visual information foraging in a focus+context visualization. In: Human Factors in Computing Systems (CHI 2001), pp. 506–513. ACM, Seattle, WA (2001)

32. Reichert, J., Jabs, A., Slickers P., Suhnel J.: The IMB jena image library of biological macromolecules. Nucl. Acids Res. **28**(1), 246–249 (2000)

33. Robertson, G.G., Card, S.K., Mackinlay, J.D.: Information visualization using 3D interactive animation. Commun. ACM **36**(4), 57–71 (1993)

34. Rouchka, E.C., Mazzarella, R., States, D.J.: Computational detection of cpg islands in DNA. In: Technical Report, Washington University, Department of Computer Science, WUCS-97-39 (1997)

35. Saccharomyces Genome Database, http://www.yeastgenome.org/

36. Sales-Pardo, M., Guimera, R., Mopeiraa, A., Widom, J., Amaral, L.A.N.: Mesoscopic modeling for nucleic acid chain dynamics. Phys. Rev. E **71**, 051 902 (2005)

37. Saraiya, P., North, C., Duca, K.: Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. J. Inform. Vis. **4**(3), 191–205 (2005)

38. Schulze-Wollgast, P., Tominski, C., Schumann, H.: Enhancing visual exploration by appropriate color coding. In: International Conference in Central Europe on Computer Graphics. Visualization and Computer Vision (WSCG'05), pp. 203–210. WSCG, Plzen-Bory (2005)

39. Silvescu, A., Honavar, V.: Temporal boolean network models of genetic networks and their inference from gene expression time series. Complex Systems **13**(1), 54–70 (2001)

40. Speed, T.: Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall/CRC Press, Barcelona, Catalonia (2003)

41. Symeonidis, A., Tollis, I. G.: Visualization of Biological Information with Circular Drawings. In: Biological and Medical Data Analysis (ISBMDA'04), pp. 468–478 (2004)

42. Tominski, C., Schulze-Wollgast, P., Schumann, H.: 3D Information Visualization for Time Dependent Data on Maps. In: the Ninth International Conference on Information Visualisation (IV'05), pp. 175–181. IEEE, London (2005)

43. Toyoda, T., Konagaya, A.: Knowledgeeditor: a new tool for interactive modeling and analyzing biological pathways based on microarray data. Bioinformatics **19**(3), 433–434 (2003)

44. de Waele, S., Broersen, P.M.T.: Order Selection for Vector Autoregressive models. IEEE Trans. Signal Process. **51**(2), 427–433 (2003)

45. van Wezel, M.C., Kosters, W.A.: Nonmetric multidimensional scaling: Neural networks versus traditional techniques. Intell. Data Anal. **8**(6), 601–613 (2004)

46. Wolfsberg, T.G., Gabrielian, A.E., Campbell, M.J., Cho, R.J., Spouge, J.L., Landsman, D.: Candidate regulatory sequence elements for cell cycle-dependent transcription in saccharomyces cerevisiae. Genome Res. **9**(8), 775–792 (1999)

47. Wong, P.C., Wong, K.K., Foote, H., Thomas, J.: Global visualization and alignments of whole bacterial genomes. IEEE Trans. Vis. Comput. Graph. **9**(3), 361–377 (2003)

48. Wright, W.: Information Animation Applications in the Capital Markets. In: IEEE Symposium on Information Visualization, pp. 19–25. IEEE, Atlanta, GA (1995)

49. Yeung, L.K., Yan, H., Liew, A.W.-C., Szeto, L.K., Yang, M., Kong, R.: Measuring correlation between microarray time-series data using dominant spectral component. Aust. Comput. J. **29**, 309–314 (2004)

DONG HYUN JEONG is a Ph.D. student in the Department of Computer Science at University of North Carolina at Charlotte. Prior to that, he was a research scholar (visiting researcher) at University of North Carolina at Charlotte. He received the B.S. and M.S. degrees from Hallym University in 1999 and 2001, all in computer engineering. His research interests include human computer interaction, virtual reality, and visualization.

ALIREZA DARVISH was born in Noshhar, Iran. He received B.Sc. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Iran in 2000 and Tarbiat Modarres University, Iran, in 2003, respectively. Currently he is a Ph.D. student in the Computer Science Department of University of North Carolina at Charlotte. His research interests include computational biology and machine learning.

KAYVAN NAJARIAN earned his Ph.D. in electrical and computer engineering from University of British Columbia, Vancouver, Canada in 2000 and is currently an assistant professor at the Computer Science Department of University of North Carolina at Charlotte. Dr. Najarians' research interests include biomedical signal and image processing. Dr. Najarian has more than 90 publications including a textbook in biomedical signal and image processing and six chapters in Wiley's Encyclopedia of Biomedical Engineering. Dr. Najarian has received several research grants, awards, and patents pertinent to biomedical applications of signal and image processing.

JING YANG is an assistant professor in the Computer Science Department at UNC Charlotte. She received her Ph.D. in computer science from Worcester Polytechnic Institute in 2005. She has been conducting research in the fields of information visualization and visual analytics, focused on large-scale multivariate visualization. Her work has been published extensively in refereed journals and conferences. Dr. Yang has been on the program committee for the IEEE InfoVis Symposium since 2005.

WILLIAM RIBARSKY is the Bank of America Endowed Chair in Information Technology at UNC Charlotte and the founding director of the Charlotte Visualization Center. He is principal investigator for the new DHS SouthEast Regional Visualization and Analytics Center. He received a Ph.D. in physics from the University of Cincinnati. His research interests include visual analytics; 3D multimodal interaction; bioinformatics visualization; virtual environments; visual reasoning; and interactive visualization of large-scale information spaces. Dr. Ribarsky is the former chair and a current director of the IEEE Visualization and Graphics Technical Committee. He also a member of the steering committees for the IEEE Visualization Conference and the IEEE Virtual Reality Conference, the leading international conferences in their fields. He was an associate editor of IEEE Transactions on Visualization and Computer Graphics and is currently an editorial board member for IEEE Computer Graphics & Applications. Dr. Ribarsky co-founded the Eurographics/IEEE visualization conference series (now called EG/IEEE EuroVis) and led the effort to establish the current Virtual Reality Conference series. In 2007, he will be general co-chair of the IEEE Visual Analytics Science and Technology (VAST) Symposium.

Dr. Ribarsky has published over 100 scholarly papers, book chapters, and books. He has received competitive research grants and contracts from NSF, ARL, ARO, DHS, ONR, EPA, AFOSR, DARPA, NASA, NIMA, and several companies.