Dynamic Visualization of Co-expression in Systems Genetics Data

Joshua New, Wesley Kendall, Jian Huang and Elissa Chesler

Abstract—Biologists hope to address grand scientific challenges by exploring the abundance of data made available through microarray analysis and other high-throughput techniques. However, the impact of this large volume of data is limited unless researchers can effectively assimilate the entirety of this complex information and integrate it into their daily research; interactive visualization tools are called for to support the effort. Specifically, typical studies of gene co-expression can make use of novel visualization tools that enable the dynamic formulation and fine-tuning of hypotheses to aid the process of evaluating sensitivity of key parameters and achieving data reduction. These tools should allow biologists to develop an intuitive understanding of the structure of biological networks and discover genes which reside in critical positions in networks and pathways. By using a graph as a universal data representation of correlation in gene expression data, our novel visualization tool employs several techniques that when used in an integrated manner provide innovative analytical capabilities. Our tool for interacting with gene co-expression data integrates techniques such as: graph layout, qualitative subgraph extraction through a novel 2D user interface, quantitative subgraph extraction using graph-theoretic algorithms or by querying an optimized b-tree, dynamic level-of-detail graph abstraction, and template-based fuzzy classification using neural networks. We demonstrate our system using a real-world workflow from a large-scale, systems genetics study of mammalian gene co-expression.

Index Terms—Visualization in physical sciences, life sciences and engineering, graph and network visualization, bioinformatics visualization, focus+context techniques

1 Introduction

Recent systems genetics research offers near term hopes in addressing scientific questions long-deemed unapproachable due to their complexity. Current research is uncovering how the genetic makeup of an organism is associated with the organism's traits on both molecular levels, such as gene expression or protein abundance, as well as physical levels including body height or tendency toward alcohol addiction. The systems genetics approach is a method to integrate data across all levels of biological scale to uncover molecular and physiological networks from DNA to function. Novel computational tools are called for to support the effort.

The central dogma [9] for genetic studies is that strings of information known as genes are stored in the DNA sequence (genome) of an organism, each gene can be transcribed into messenger RNA (i.e. a transcript), and ultimately into proteins which affect the behavior or morphology of the organism. This multi-step process by which a gene's sequence of nucleic acids (ATCG) is converted into mRNA transcripts is known as gene expression. Gene expression directs the process of cellular differentiation, in which specialized cells are generated for the different tissue types. The regulation of gene expression (i.e. gene regulation) controls the amount and timing of changes to the gene product. This is the basic mechanism for modifying cell function and thereby the versatility and adaptability of an organism. Therefore, gene expression and regulation function as a bridge between genetic makeup and expression of observable traits.

Despite its vital importance, determining the precise roles of given transcripts remains a fundamental challenge. This is due in large part to the complex machinery employed for gene expression in which some gene(s) may regulate the simultaneous transcription levels of other genes. This regulation leads to statistically correlated, or co-expressed, genes in which one gene is expressed at high levels only when the other is as well. While collecting gene expression data already requires great technical sophistication and resources, the limited functionality of current computational tools to discover structural patterns of co-expressed genes from the collected data presents another grand challenge. Without an ability to efficiently and comprehensively explore the problem space, full genome-scale gene expression data are still of limited value for today's hypothesis-driven research.

Genes act alone or in groups during the process of gene expression and regulation. Biological pathways are defined by the connectivity of upstream and downstream effects of genes and gene products, including their action in the regulation of the expression of other genes. The search for genes co-expressed in a common group, which likely affect observable traits as a functional unit, often starts with using microarray technology to profile transcript abundance. A microarray is a device containing microscopic DNA probes and is capable of measuring the expression levels from thousands of genes for a given sample [11]. From microarray data, biologists can statistically construct massive correlation matrices that describe pair-wise gene co-expression. The key challenge then is the representation, decomposition, and interpretation of this genetic correlation matrix.

By treating the correlation matrices as adjacency matrices, it is natural to consider correlations of gene expression in the setting of a graph, where vertices represent genes and edges represent the strength of correlation between pairs of genes. In this analogy, a group of genes that co-express would necessarily form a network or subgraph consisting of "highly correlated" genes.

Although it seems straightforward to directly apply classic graph algorithms to discover those highly correlated subgraphs, this approach by itself is not sufficient. Many common graph problems such as clique finding are NP-complete. Even for moderately sized problems, the computation time is still often overwhelming. Hence, to ensure that problems are computationally tractable, the current practice is to apply data filtering steps to dramatically reduce the density of the graph and dimensionality of the data. The value of key parameters, such as correlation threshold, is often decided by an educated guess based on the data size, algorithmic complexity, and the hardware available. Unfortunately, picking a slightly different threshold often eliminates many of the subtly important correlations and thereby drastically changes the solutions of graph algorithms.

In addition, scientists in many cases cannot exactly define the term "highly" with rigor as it is a qualitative criterion with uncertainty. The uncertainty aspect is further exacerbated by the noise present in current microarray data, inaccuracies introduced during data collection, and residual errors in subsequent statistical analyses. To date, it has been hard for scientists to evaluate the true value of gene co-expression as well as the sensitivity of an "optimal" result computed using expensive

- *Authors New, Kendall and Huang are with the Department of Computer Science of the University of Tennessee, Knoxville. Email: {new, kendall, huangj}@cs.utk.edu.*
- *Author Chesler is the Systems Genetics Group Leader at the Oak Ridge National Laboratory. Email: {cheslerej}@ornl.gov.*

Manuscript received 30 July 2007.
Under review.

graph algorithms.

In this work, we designed a visualization system to provide domain scientists with tools to evaluate the validity and sensitivity of key parameters in their research hypotheses. By allowing realtime feedback of connectivity, determination of biological relevance is facilitated by allowing more thorough analyses of their empirical data. Our main effort focuses on providing interactivity from a number of features beyond fast rendering rates. A user can interactively explore and filter the data to create meaningful subgraphs by leveraging four complementary methods: (i) semi-automatic segmentation of highly correlated subgraphs with a 2D focus+context graphical user interface segmented using block tridiagonalization, (ii) quantitative database queries on data of interest using traditional compound boolean range queries, (iii) qualitative queries on points of interest using neural networks, and (iv) dynamic extraction of subgraphs using several fast graph theoretic algorithms. In addition, these meaningful subgraphs may be used as templates to perform template-based searches through the entire dataset. The whole process of template creation, template-based search, extraction of graph metrics, and displaying statistical and visualization results is interactive.

Modern microarray data is noisy and complex; visualization alone is not the answer. By providing interactive visual analytics tools such as graph algorithms, neural network analysis and level-of-detail control, we bring a human expert into the loop to negotiate the tradeoff between data size and algorithmic complexity by intuitively tuning key parameters with realtime feedback for addressing the scientific question at hand. We demonstrate our system using datasets from a real-world, large-scale, genetical genomics study of mammalian gene co-expression. In this study, the influence of genetic differences among individuals is considered as a source of expression covariation.

In the remainder of this paper, we first describe the background of this research in Section 2. In Section 3, we present our design of the user interface and the overall system, followed by implementation details in Section 4. Finally, our results and discussion are provided in Section 5, and then concluded in Section 6.

2 Background

2.1 The Driving Application

2.1.1 Quantitative Trait Locus

Let us consider an analogy familiar to the field of computer science: a variable stored at a location in the main memory of a computer. In genomics, one can consider the entire memory space roughly corresponding to the genome, a location-specific variable as a gene, and the value stored in each variable as the genotype at that location. The value of a genotype is transmitted by each parent. The fact that each location can take on different genotypes is termed polymorphism, since the same genome location for different individuals may hold (parts of) different genes or non-gene DNA sequences.

The entire set of genotypes across the genome defines the genetic makeup of an organism, while a phenotype defines the actual physical properties, or traits, of the organism. Although genetic makeup is not the sole factor influencing an organism's phenotype, it is often a strong causative predictor of the trait. Consider common traits relating to physical appearances as an example. Having exactly the same genotypes, identical twins have strikingly similar appearances (phenotypes), yet due to environmental influences they may not look exactly the same.

It is of great interest to unravel the inner workings of how genotypes influence molecular networks to affect a phenotype such as agility, seizures, and even drug addiction, to name a few. Geneticists have already achieved great success in associating a genotype and phenotype for a trait determined by one gene (i.e. monogenic traits), but much present attention is now focused on traits that are determined by many genes (i.e. complex traits). These traits are continuously distributed random variables and thus referred to as quantitative traits. Linear modeling is used to identify genotypes that predict phenotype values. The location of these genotypes are quantitative trait loci (QTLs) [3]. Detected via statistical methods [9], QTLs are stretches of DNA highly

Gene

	Probe Set Id	Symbol	Megabase	Description	...
1	100381_at	Acta1	125.68943	Alpha skeletal muscle actin	
2	100959_at	S100a13	93.208839	Calcium-binding protein A13	
3	101086_f_at	Cnbp	89.268018	Zinc finger protein 273	

QTL

	Probe Set Id	Locus	QTL	Chrom	Gene Megabase	Heritability	...
1	100381_at	D12Mit234	7	12	1671.671	0.262125693	
2	100959_at	S01Gnf003.450	51	1	6.592	0.734209247	
3	101086_f_at	DXMit105	9	X	2499.020	0.280626696	

Paraclique

	Vertex	Paraclique	Percent connected to Paraclique 0, 1, 2, 3, ... , 36					...
1	99574_at	4	17.6%	68.9%	12.1%	7%	100%	
2	103752_r_a	31	63.1%	19.6%	12.1%	1.6%	36.4%	
3	100060_i_a	0	100%	15.5%	94.7%	79.5%	63.6%	

Fig. 1. In addition to the gene-gene correlation matrix, our system also handles data supplied in relational tables such as these containing gene annotations, QTL membership from QTL Reaper [41], and paraclique connectivity [23].

associated with a specific phenotype, analogous to genetic landmarks which roughly indicate the position of the active gene. QTLs are not defined at very fine granularity; they usually correspond to areas large enough to hold several genes. The genetic polymorphism (genotypes) in neighboring areas of a set of loci, as a group, influence structure and function on both molecular and organismic scales.

For decades, scientists have systematically randomized and then stabilized genetic variation in groups of mice to effectively create a population of clones. These mice, called "recombinant inbred" (RI) strains, function as a reference population which is used by groups worldwide in order to allow a basis of comparison and integration across different experiments [8]. This is very important from a statistical standpoint as it implies that the potential size of the combined datasets is theoretically unbounded, resulting in extremely high dimensional data. Sufficient confidence is currently allowing integration of diverse biological data across levels of scale in an approach related to systems biology, "systems genetics." This integrative approach for multiscale and multiorgan phenotypic datasets has only become feasible in recent years and relies heavily on statistical techniques, complex algorithms, high-performance computing and visualization.

2.1.2 Gene Expression Data

The statistical approach of QTL mapping associates phenotypes to genotypes in order to identify plausible genome locations that control those traits. The QTL region is large due to the imprecision of phenotypic estimation, the low density of genotypic recombinations (transitions in the distribution of the genotype string across the genome), and an often insufficient number of genotype parameters in mapping models. Identifying the specific region or interacting regions, and homing in on the precise polymorphic regions or other DNA features that regulate trait variability requires tremendous information integration.

Gene expression data have been used to refine QTLs to the granularity of genes and further reveal the underpinnings of how complex traits are controlled. To understand gene expression, we need to identify genetic regulators of gene expression, particularly those in the form of a network of genes that are "regulated" together. In a way, the study of gene expression identifies modules (gene networks), while QTL studies allow one to determine the cause of variation of those modules in relation to complex traits. Gene expression data is collected using microarrays [11]. During the process of gene expression, mRNA transcripts are produced based on the "instructions" contained in the gene's DNA sequence. We will refer to a gene as the DNA sequence or causative polymorphism and transcript as the gene product.

The magnitude of co-expression relations among all pairs of transcripts are computed from the microarray data. The levels of co-expression (i.e. correlation), are then stored in an $n \times n$ matrix, with n being the total number of genes. Treating the resulting symmetric correlation matrix as an adjacency matrix, we then have an undirected

weighted graph. A positive weight means the two transcripts connected by the edge are co-expressed (i.e. if one is active the other is as well). Likewise, a negative edge weight means the two transcripts are under opposing patterns of genetic regulation. In this context, a network of highly related (either co/up- or oppositely/down-regulated) transcripts would take the form of a dense subgraph.

To the visualization community, the main research challenge here is to allow scientists to efficiently and effectively explore gene expression datasets to discover gene networks and to suggest controlling mechanisms of complex traits in a credible manner. To validate causality, scientists can then employ in vivo experiments to perturb (e.g. "knock out" a set of "master switch" genes) gene products and observe whether the organism expresses the expected phenotype.

2.2 Related Work

A graph is a universal concept used to represent many different problems. While more restrictive layouts, such as trees, should be used when possible, this work will address the general case of graph interaction. In relation to this work, we categorize methods to comprehend graph properties as: (i) those solely depending on algorithms, i.e. the algorithmic approach, and (ii) those incorporating human input as an integral component, i.e. the interactive approach. Let us review both approaches in turn.

2.2.1 The Algorithmic Approach

Algorithmic research to automatically compute graph properties of various kinds has been extensively studied. Well known examples include clique, strongly connected components, induced subgraph, shortest paths, and k-connected subgraph. Let us use clique analysis as a representative example. By filtering out edges with weights below a certain threshold, a gene network with high co-regulation should appear as a complete subgraph, or a clique. Hence, it is natural to consider clique analysis in gene expression data analysis.

However, clique analysis is an NP-complete problem. Even though more efficient fixed-parameter methods [23] are currently being used, it is still a very time consuming procedure to compute. It is also hard to treat edges with negative weights in the context of clique analysis, so common approaches typically preprocess the graph to convert all edge weights to absolute values. The impact of information loss due to thresholding is hard to evaluate and is further complicated by the presence of noise. While partially resolved by paraclique [23] methods in which a few missing edges are acceptable, additional problems are introduced such as the meaning of paraclique overlap which may be handled differently depending on the working hypothesis.

Such shortcomings apply to different graph algorithms in varying degrees, but are generally inherent with graph theoretic analysis. However, this should in no way prevent graph algorithms from being used for suitable problems. From this perspective, it would be greatly advantageous to develop a visual, effective and efficient feedback framework. In this framework, a human expert is enabled to quickly identify imperfect portions and details of the data, and not only remove irregularities but also to significantly reduce the dataset's complexity by interactively constructing various levels of abstraction. The resulting problem space would be more appropriate for graph theoretic analysis to be applied. In fact, some undertakings in visualization research have already adopted similar approaches [29].

Here we note that our goal is neither to accelerate all computation in a scientist's workflow nor replace computation solely with visualization. We hope to develop a visualization framework which allows navigation through gene expression data and segmentation of the appropriate data for further study. In this way, s/he can flexibly choose and apply the right computational tool on the right kind of problem.

2.2.2 The Interactive Approach

Much related work in visualization follows the Information Seeking Mantra proposed by Shneiderman [36]. That is: overview first, zoom and filter, and then details on demand. At each of the three stages, there are a number of alternative approaches, many of which are highly optimized for a specific application. A key driving application in this area has been visualization of social networks [28].

To provide an overview, the graph can be rendered in the traditional node-link setting or adjacency matrix [1], and more recently as a self-organizing map [21]. When using the common node-link model, it is pivotal to develop a sufficient hierarchy of abstraction to deal with even moderately sized graphs. Solely relying on force directed methods (i.e. spring embedding [26]) for graph layout cannot resolve visual clutter and may still significantly hamper visual comprehension.

Structural abstraction can be computed either bottom-up or top-down. In bottom-up approaches, one can cluster strongly connected components [22], or by distance among nodes in the layout produced by a spring embedder [40]. Top-down approaches are often used for small scale or largely sparse graphs in which hierarchical clusters are created by recursively dropping the weakest link [4]. More comprehensive systems employ clustering algorithms that consider a number of different node-edge properties [2].

Semantic-based abstraction is a more powerful mechanism for providing an overview, zooming, or giving details. This approach is tied to its intended application since it requires specific domain knowledge of the semantic information [37]. When combined, structural and semantic abstraction can prove to be very effective [34]. Also in [34], it is shown that overview and level-of-detail (LoD) enabled browsing can be based on induced subgraphs of different sizes.

There are many well-known packages that have evolved over time to specifically address visualization of gene correlation data using node-link diagrams such as Cytoscape [32] and VisANT [17]. These tools are built to be web accessible and thus render node-link diagrams using 2D layouts. While 2D layouts are accepted by the community, such packages neglect modern 3D acceleration hardware, rarely scale well beyond hundreds of nodes, and do not leverage 3D operations that have proven to be the preferred representation and navigation techniques for our users. Due to the common 2d framework, and in contrast to Shneiderman's principle, biologists are typically forced into a workflow in which filtering must be first applied and a global overview of the entire dataset simply isn't possible. Our software leverages both OpenGL and efficient C compilation to facilitate interaction with tens of thousands of nodes while maintaining interactive performance with complex visual analytics tools not currently available in these packages. Current work involves integration with a lightweight API [33] to allow web-based interaction and data-sharing so our software may be used synergistically with such well-developed packages.

In contrast to the node-link model, an adjacency matrix is a clutter free interface. While an adjacency matrix interface for large data is limited by the resolution of the display, it is still ideal for a bird's eye view [1]. Some patterns such as clique and bipartite subgraphs could be very distinctive when examined in an adjacency matrix. However, a proper order of vertices is critical. The community has studied this problem at length. In [16], a comprehensive survey on automatic vertex order is included. In general, binary, undirected graphs are the most straightforward. While weighted graphs needed more complicated algorithms, graphs with negative weights are less studied. Based on adjacency matrices, LoD type of browsing is often supported as well [1].

Due to the complexity involved in computing a high quality overview of a graph, researchers have also attempted to use self-organizing maps [21]. Self-organizing maps are a dimension-reduction technique which adjusts weights in a manner similar to neural networks to discretize the input space in a way that preserves its topology. The end result is (usually) a 2D field that can be conveniently rendered as a terrain.

By creating a spatial layout for a graph, it can be interactively visualized while preserving the data's underlying topological relationships. Typical interaction methods include focus+context methods (i.e. zoom and filter), graph queries using language-based methods [35], and filtering databases of graphs using graph similarity metrics, typically based on non-trivial graph theoretic algorithms [29].

Social networks are currently a primary driving application of interactive methods for graph visualization. This has resulted in non-

binary, non-positive definite weights not being as thoroughly studied. Also, tools for extracting highly connected subgraphs from this data in a way that addresses the inherent uncertainty appear to be lacking. Whereas neural networks have already been used for volume segmentation [39], similar approaches have rarely been attempted in graph visualization. In this work, we propose several tools that allow traditional quantitative drill-down as well as qualitative selection and filtering techniques to aid domain experts with their analysis.

2.3 Beyond The Size of the Dataset

Over the years the cutting edge of large data has advanced in strides. At all times, however, the criteria of being large can only be defined in a domain specific manner. For instance, while a terabyte (TB) scale multivariate time-varying simulation is large by current standard, a dataset sized at 10 gigabytes (GB) is already large for medical visualization requiring real time frame rates.

In systems biology, datasets are large in a different way. Specifically for gene co-expression data, 100 megabytes (MB) already qualifies as large. This is due to the fact that genes act more often in combination than alone. The research is to discover how (overlapping) combinations of genes act to regulate various phenotypes. Hence the real problem space to deal with is exponentially larger. Further exacerbating the problem is the uncertainty caused by data noise and those critical data filtering thresholds that are usually chosen according to previous experiences.

While thresholding of edge weights is a useful filter mechanism, it may seriously affect subsequent statistical analyses when decomposing a correlation matrix. Discovery of dense subgraphs and the genes that connect them allow a higher level of abstraction and provide specific targets for gene perturbation tests. Because the genes must be highly connected with a sub-network, thresholding of edge weights is often used. However, this often excludes genes that would receive higher consideration based on domain specific knowledge. Further, this strict thresholding specifically excludes genes that fall outside of the dense subgraph, but which are highly connected to it and potentially other networks; such genes may occupy important positions within the network of expressed genes and QTL locations. The simple capability of dynamically adjusting the threshold, in combination with immediate visual feedback and dynamic statistics, allows users to develop a much better intuitive understanding of their data and thus better define the network properties of genes of interest for a given task. By making this analysis an interactive visual experience, network characteristics can be identified and used to identify similar structures without a deep mathematical understanding of the correlation matrix.

As methods to measure quality, usefulness and uncertainty in data are becoming central issues of visualization research, our research is an application driven undertaking that builds upon the knowledge and expertise of domain scientists in systems biology. Our primary focus is to comprehensively study the exponential problem space embedded in a real world gene co-expression dataset through visualization.

Much previous research and applications have been developed which utilize visualization to facilitate biological insight. For instance, a method to navigate proteomics datasets obtained using liquid-chromatography/mass-spectrometry was reported in [24]. A comprehensive survey of five existing popular gene expression microarray data visualization tools was provided in [31] with empirical evaluations. Methods used include heat map, clustering and graph renderings, parallel coordinates, scatterplots, bar charts, and histograms. While users preferred to have more analysis capabilities such as clustering integrated with the interactive visualizations [31], those tools focused on providing high quality visual representations and interaction with the data. Throughout the literature, we are not aware of any previous works that aimed at supporting visualization and analysis of a large combinatorial space with a tolerance of uncertainty.

3 Approach

Our goal is to develop a visualization system in which a human expert discerns uncertainties in the data and guides the system to segment a large graph using a set of automated tools through an interactive interface. The key components of the system include the 2D interactive interface, modules to select subgraphs both qualitatively and quantitatively, and the neural network based classifier that uses selections as a template. We start the discussion by describing the exact set of input data to our system.

3.1 The Data

The only required data is a matrix containing gene-gene correlation values. While all of our testing data use Pearson's correlation, different metrics of correlation are treated no differently in our system. In addition, we handle a database of information corresponding to each gene as can be seen using three relational tables (Figure 1). Specific information about the object of interest is stored in the Gene table while information relating to computed gene networks is stored in the Paraclique table. Since our driving application is to identify the genes that cause variation in complex traits, it is necessary to show the relationship or distance between genes and QTLs. For that, we need an additional relational table describing the exact location of QTLs in the unit of megabases.

Graph theoretic algorithms provide valuable information that is otherwise hard to discern about the data. However, many such algorithms incur long compute times and are far from being interactive. For those algorithms, it is then necessary to pre-compute and store their results for visualization at run-time. In this work, for example, we precompute and store each gene's membership in any of the paracliques. The resulting data can easily be stored in a relational table.

We treat all data in the relational tables as attributes of individual vertices, and the correlation values as an attribute specific to each edge. This is a very generic model that is applicable to a variety of application domains and is a boon to scientists typically involved in spreadsheet science. Based on these data, it is then the job of the visualization system to facilitate interactive, hypothesis-driven study by the user.

3.2 A Clutter-Free Interface for Graph Abstraction

A major difficulty with graph visualization is the visual clutter caused by the sheer complexity of the data. As discussed in Section 2.2.2, an adjacency matrix provides a concise interface for overviewing the data in a way that is free from visual clutter. However, the 3D space is still a natural domain for user cognition. The added dimension can be used to convey additional data, allowing navigation through node-link rendering and full appreciation of structural cues in the data. For our application, we have designed a framework (illustrated in Figure 2) where an overview is provided through a 2D adjacency matrix. Users can arbitrarily select subsets of interesting vertices and create abstractions for further interactions in 3D.

Unlike popular datasets studied in most previous works, the range of edge weights in our data is `[-1.0,1.0]`. In addition, to let users decide about uncertainty issues, we would like to avoid taking a threshold at this stage of processing due to possible information loss. This makes it hard to directly leverage existing vertex reordering algorithms like those surveyed by Mueller [25] or Henry et al. [16].

Block tridiagonalization (BTD) is a mature numerical algorithm that permutes row and column elements of a matrix in such a way as to cluster nonzero elements in blocks along the diagonal [5]. This algorithm always preserves the eigenvalues of the original matrix within a specified error tolerance. It iterates until the following criteria are met: (1) the final matrix has small bandwidth relative to the size of the matrix, and (2) any off-diagonal blocks in the final matrix have either low dimension or are close to a low-rank matrix.

The BTD algorithm was developed to improve both performance and storage efficiency for solving eigen problems. The smaller a block is in a matrix, the lower the corresponding rank in most cases. Thus, the optimization goal of BTD is to minimize block sizes on the diagonal, and correspondingly reduce block sizes off-diagonal as well.

The result of BTD is often characterized as minimization of bandwidth, because non-zero entries are clustered around the diagonal. It is very significant to our research. In our application, the minimization

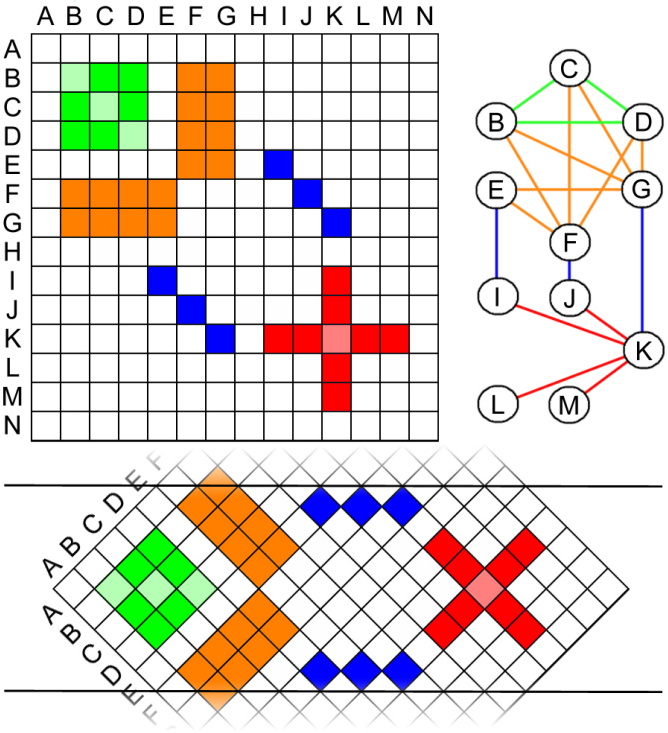

Fig. 2. Illustration of a permuted adjacency matrix with common graph patterns (top), and extraction of the BTD belt for qualitative selection (bottom).

of diagonal block sizes through global optimization provides a reliable means to abstract a large graph into a set of minimal basic "building blocks," each of which represents a densely correlated subgraph. The vertices in these subgraphs appear in contiguous segments along the diagonal. The off-diagonal blocks determine how these "building block" subgraphs are interconnected. In this way, we can conveniently reassemble the original graph using the minimized diagonal blocks, and show more appreciable structures with significantly less clutter.

Let us consider the illustration in Figure 2 from a biological perspective. In this example, we show four graph patterns that are often of interest to geneticists. Since every data entry in an adjacency matrix represents an edge, selections made in adjacency matrices are on edges and only indirectly on vertices. The green subgraph is a clique of highly correlated genes that potentially operate as a unit. The orange subgraph is a bipartite graph, used in gene-phenotype mapping, in which the trait is correlated with a number of related genes which would be of experimental interest. The blue subgraph is a perfect matching graph which functions as bridges between subgraphs. The red subgraph is a star containing a "hub" gene which could be targeted for knock-out and affect expression in many structures.

For our real world datasets, BTD has been able to consistently generate permutations that compress the majority of non-zero to the diagonal. This enabled us to crop a stripe along the diagonal and rotate that stripe to a horizontal position, as shown in Figure 2, bottom. We refer to the horizontal stripe as the BTD belt.

BTD belt is a more efficient use of precious screen resources. Our datasets typically contain several thousand genes so the adjacency matrix is typically very large. Although it is possible to downsample the matrix for on-screen viewing, the essential high frequency details in the matrix could be hard to distinguish.

We show an example of the BTD belt computed from a gene co-expression study of brain development in Figure 3. There are 7,443 genes in this dataset. In an adjacency matrix, one can make selections with square shaped bounding boxes. In BTD belt, these square bounding boxes become diamond shaped. Ten sample diamond shaped selections have been specified in Figure 3 and magnified to show details.

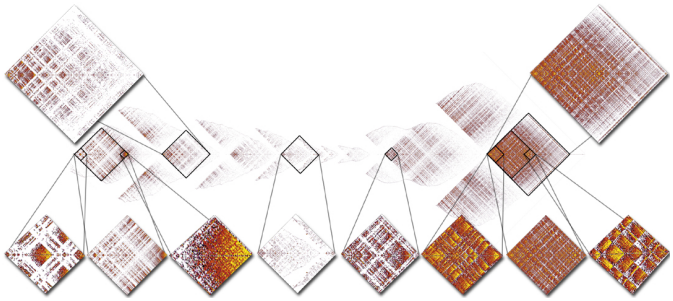

Fig. 3. A BTD belt, with magnified views, from a real-world mammalian gene co-expression study of brain development involving 7,443 genes.

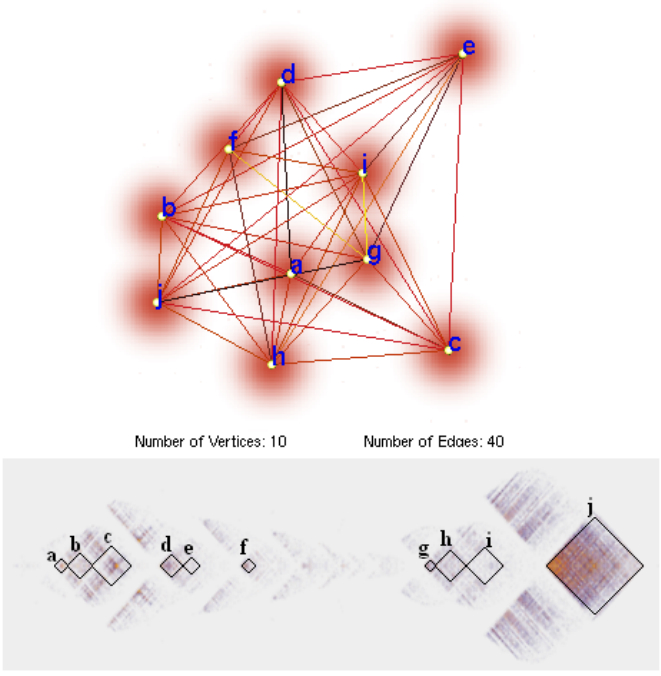

Fig. 4. A 2D level-of-detail graph created from brushed BTD belt selections to show correlations among BTD structures.

From the BTD belt interface, a user can select diagonal blocks that are perceived to be "highly correlated". Letting a human expert decide what can be considered as "highly correlated" is our way of handling data uncertainty. We note that the functionality of detecting high correlation in a general setting is a hard problem, particularly when the acceptable tolerances of error can only be qualitatively determined in a subjective manner.

In this regard, BTD can be considered as a computational tool for creating data abstraction. Figure 4 shows a simple example in which ten subgraphs have been selected using diamond shaped bounding boxes. Each subgraph is abstracted as a supernode in the LoD graph. From the much simplified graph, the interconnections among the graph nodes are clearly discernable. As expected from the BTD representation, *a*-*c* and *g*-*i* form cliques with high edge weights. Interestingly, node *a* has a strong negative correlation (as indicated by the dark line color) with subgraphs *d*, *g*, and *j*. LoD visualization of BTD selections complements BTD in the sense that while these multiple separated clusters may not share many edges (and thus may not be readily visible in the BTD belt), the edges that do exist have strong negative correlations. The LoD representation implies that subgraph *a* is a potential down-regulator for these three major networks.

Since the BTD-belt is a permuted and rotated adjacency matrix,

much information is visible along the diagonals which correspond to an individual vertex. Therefore, the BTD belt immediately shows the major graph structures in which each vertex participates. These properties can be used to quickly determine the role of specific genes in varying numbers and types of networks.

At a higher level of abstraction, several large structures share many edges where their diagonals cross as seen in Figure 4 where subgraphs *g-i* form a bipartite-like structure with *j*. To be specific, the bipartite structure visually represents the edges induced by the intersection of each graph's vertex set. The striping pattern of the bigraph structure implies it is typically the case that a single vertex in *g-i* is connected to many vertices in *j* and not vice versa. This allows biologists to get a view of how multiple structures may interact, or be regulated, through genes that they have in common.

To allow further data abstraction, users are able to dynamically generate level-of-detail (LoD) representations that facilitate simultaneous operation across multiple levels of scale. At any point in our system, the current working graph can be saved and is rendered, along with all other saved subgraphs, in the background of the current working graph. The LoD graph can be generated by taking all user-defined subgraphs and treating them as supernodes. By default, the edge connecting two supernodes has a weight defined as the average of all edge weights between vertices within the two corresponding subgraphs. Optionally, feature vectors of topological metrics can be calculated on each subgraph for querying and graph pattern matching, using methods described in Sections 3.3 through 3.5, to find similar graph structures rather than similar data items. This LoD graph can subsequently be treated as a normal graph and all provided analytics tools in our system can be used to perform increasingly complex analysis at higher levels of abstraction.

3.3 Quantitative Queries

Information contained in relational tables (Figure 1) needs to be studied in an integral fashion with gene networks. A common comprehensive tool for accessing information in those formats is compound boolean range queries. Unfortunately, it is a difficult process to efficiently integrate a commercial database management system with realtime visualization systems. For this application, we have extended the functionality of a recent visualization data server to provide essential data management functionalities. The core of that data server is a simplified B-tree that is optimized assuming no run-time insertion or deletion in the B-tree [12]. Using this B-tree, the database in our system, with compound boolean range queries over 8 features, can be queried in O($log\ n$) time at a rate of 10 million vertices per second on a 2.2Ghz Athlon64. This results in interactive, sub-millisecond response to sets of dynamic queries even for large graphs.

The effect of querying is data filtering and thereby complexity reduction. One of the most common data filtering operations for biologists is thresholding. With visualization, this threshold choice can be applied to a graph interactively and result in both visual and statistical testing for proper threshold selection. Besides thresholding, we also provide more power by also allowing multiple, dynamic, quantitative queries over any computable attributes of a vertex or edge. Sample queries include all genes within a 100-megabases distance to a target QTL, or all genes in the current subgraph that are significantly related to a paraclique of interest.

The queried genes also form a subgraph, no different from those qualitatively selected via the BTD belt interface. Our querying system attains realtime performance, making it feasible to visually evaluate the effects or sensitivity of key parameters, such as what threshold value to use.

3.4 Dynamic Fuzzy Classification

When interacting with complex data, it is often useful to provide an automated arbitrator between the user and the data so that repetitive tasks are off-loaded from the human user. One such important task is to exhaustively search for genes that match a certain pattern and classify the data accordingly while handling the innate uncertainty. A key motivation is to alleviate users from the need to manually browse through the entire dataset and thus specifying the feature they think they are seeing in an overly rigorous manner. It is desirable to have a proper level of fuzziness into this iterative feedback loop. While elaborate agents can take a long time to develop and are too complex to train in realtime, we have implemented a simpler AI system which users can train at run-time.

In our system, we use a feedforward, multilayer, back-propagation neural network capable of quickly learning items of interest and displaying similar items to the user. From this perspective, qualitative selection allows users to visually perceive uncertainties and decide how to best guide the computational process, while quantitative queries provide an exact means to request a subset of data. With all datasets, the neural network classifier can be trained with subsecond efficiency.

3.5 Graph Properties

While our neural network treats input attributes indifferently, the choice of which properties to feed into the classifier is quite important. Besides domain-specific database variables, one may also employ several integrated graph properties for graph similarity classification. Those include: degree of vertex, transitive closure, connected components, edge expansion, and shortest path.

Each of these properties has a unique biological interpretation. Degree is one of the simplest useful metrics that can be calculated and allows biologists to determine statistical correlation between genes and gene networks as there are often many loosely connected genes and few highly-related genes. Transitive closure minimizes the longest distance to all other nodes and can be used to find the core of a genetic structure. Connected components allow biologists to visualize only related data within a given subgraph. Edge expansion shows relationships within a given subgraph without regard to threshold. Shortest path allows biologists to see how specific genes most directly regulate one another. It could also be used to further investigate a favorite location in the graph and gather only the local correlates view of the specific gene(s) under study.

These properties may be used as extra variables in the feature vector used for training a neural network. By adding these to the relational data that are queried, we provide more information to be leveraged for fuzzy classification.

4 System Implementation

In this section, we describe details of our system that may be of interest to readers who would like to implement a similar system. Several performance numbers for our system may be found in Table 1. The number of vertices, edges, and range of absolute values of edge weights are shown, respectively, along with performance metrics referenced in the sections below. The columns of "2D" and "3D" show the running times (in seconds) of Fruchterman and Reingold's [10] method operating in 2D and 3D, respectively. The column of "BTD" shows the timing results of the BTD permutation process in seconds. The time to complete neural network training from a few examples and counterexamples along with classification of the entire dataset is recorded in number of milliseconds (ms) in the column "NN".

Since this discussion is not restricted to one of only gene expression applications, we use four test datasets detailed in Table 1 to indicate the scalability of the system. The datasets include genotype correlation datasets used to study human lung cancer, medical bibliographic references, mouse behavior, and web architecture made available by the developers of GeNetViz [42].

The primary dataset, which we use throughout the rest of this paper, for our driving application involves regulation of 7,443 genes for research of mammalian brain and behavior [7]. Visualization results concerning complex traits in this dataset are shown in Section 5.

4.1 Graph Layout

Our system attempts to load any pre-processed data available or subsequently generates the files if they are not available. Upon startup with a valid weighted-edge graph, the system inspects the graph to determine specific properties which would necessitate a special layout, as in the case of a bipartite graph. Otherwise, it generates 2D and

Table 1. Datasets and Timing Results (in Seconds)

|V|	|E|	|Weight|	2D	3D	BTD	NN(ms)
254	401	[0.57,0.95]	0.203	0.282	0.1	16
2,150	6,171	[0.97,1.00]	16.20	17.19	6.1	31
7,443	695,122	[0.85,1.00]	336.0	318.3	50.9	141
12,343	28,338	[1,74]	883.3	912.7	234.7	172

3D layouts using either Kamada-Kawai [20] energy minimization or Fruchterman and Reingold's [10] force-directed placement. Both layout algorithms are the standard implementations which use simulated annealing to slowly freeze the layout in place. The layout is said to converge when it reaches a maximum number of iterations, reaches a small percentage of the initial system temperature, or does not change significantly between iterations.

Like most regular layout algorithms, in our implementation the vertices are initially placed randomly and then iterated through four main phases until convergence: random impulse, impulse away from all other vertices to keep them from overlapping, impulse toward the center to keep the system from continuously expanding, and per-vertex impulse toward or away from its connected neighbors in proportion to the edge weight to preserve graph topology. The entire process is $O(M|V|^2)$ where $|V|$ is the number of vertices and M is the number of iterations ($M \sim |V|$ but varies significantly). Through experiments we found that the performance differences between 2D and 3D graph layout algorithms are quite minor as shown in Table 1.

The software has been designed to easily incorporate other graph layout algorithms, such as the fast multipole multilevel method (FM^3) [14] or LinLog [27]. Such layout algorithms have different strengths and weaknesses in conveying specific properties in the resulting topology. Custom algorithms can be easily incorporated in our application either procedurally or by simply loading a graph layout file. By default, a single layout is computed for a graph and its appearance is modified at run-time but the user may also switch interactively between multiple layouts. Additionally, the user may dynamically swap between the customary 2D view and the 3D view which tends to convey more relationship information.

It is also noteworthy that clustering is another term often used by biologists in their research. Although in our work "cluster" and "dense subgraph" have similar meanings, the goal of our approach is quite different from the basic goal of popular clustering algorithms. Our goal is to adequately handle uncertainty in the pursuit of co-regulated (putatively cooperating) genes forming a network, and the nature of the interconnections among those networks. We use BTD belt, queries and neural network to computationally assist the discovery and real-time fine-tuning of those dense subgraphs of interest. We do not solely rely on graph layout algorithms to reveal dense clusters.

4.2 Rendering

Vertices are rendered after the edges without the depth buffer to prevent edges from occluding data points. We support the option of rendering vertices as splats (also known as billboards or impostors) or quadrics which can take arbitrary shape (usually spherical). The splatting implementation is the typical geometry-based primitive scheme using pre-classification and thus results in slightly blurry vertices but has the advantage that it is typically fast. While both options render in realtime on most single computers, framerate tests were conducted for the 7,443-node graph at several resolutions on a powerwall using Chromium. This resulted in 16 fps quads vs. 246 fps splats on an 800x600 viewport, and 5 fps quads vs 17 fps splats at 3840x3072 (9 monitors).

There are many interactive rendering options such as color table generation and weight-mapping mechanisms for coloring edges. A default set of these has been provided to express differentiation between solely positive and negative edges (up and down regulation) or to enhance contrast between edges with similar weights. The rendering mechanism is adaptive and can optionally adjust properties such as the number of subdivisions in the quadrics based upon the current frame rate. Semi-transparent vertex halos [38] are rendered using splatting for enhanced depth perception. Intuitive click-and-drag interaction and continuous rotation during manipulation circumvents problems with 3D occlusion and aids perceptual reconstruction of the topology through motion parallax. The system also has dozens of minor utilities including the ability to change the color table used by all elements of the program, take a screenshot, create video, print statistical information for the current graph, and output gene lists.

4.3 Neural Network

We use a multilayer, feedforward, back-propagation, online neural network for realtime classification which is able to learn while in use by employing stochastic gradient descent. Our implementation closely follows the description in [30]. Results are given for a neural network with the number of input nodes corresponding to the number of provided attributes, 30 hidden nodes, 2 output nodes, a learning rate of 0.4, a sigmoid threshold function, and a hard max used to simply select the most likely output. The unusually large number of hidden nodes provides sufficient degrees of freedom for any problem domain and could be reduced if training speed or overfitting become issues. Each of the two output nodes corresponds to the likelihood that the user does or does not want to see the object.

Neural network interaction involves only a few easy steps. The user left-clicks on an arbitrary number of vertices to select them as examples. Similarly, right clicking on a vertex adds the vertex as a counter-example. The user may use any filtering or processing techniques previously mentioned to aid the process of defining the training set. Once all examples and counter-examples have been selected, the entire dataset is processed to only show items like the examples or to segment the data using color. Training the neural network and classifying all other data items is in the order of dozens of milliseconds, shown in Table 1, and is transparent to the user.

Deciding the proper training set is critical when attempting to achieve accurate classification for multivariate data due to the high-dimensional decision space. In our system, we provide two alternative approaches for the user definition of large training sets. First, we allow selection of the training set using supernodes in the LoD graph. Each supernode represents a network of genes, typically segmented through BTD selections or database queries, such that a few example supernodes can correspond to hundreds of individual data points. In this way, users can visually interact with the data while quickly selecting entire groups of vertices as examples or counterexamples. Second, the application allows the import/export of vertex data for the current working graph using ASCII files. This can be used to analyze the corresponding data with much more sophisticated statistical packages such as Statistical Analysis Software (SAS) or statistics programming languages such as R. These analytics tools or their batch programs can be called during run-time to either fully segment or partially classify the data. The result can be stored in a vertex list file and then utilized as a training set.

5 Results and Discussions

5.1 Overview: Data and Workflow

While early microarray studies emphasized differential expression and comparative analyses, modern applications [18] emphasize correlation of gene expression across large sets of conditions, including environments, time points and tissues. Increasingly, this data is being collected in a context of natural genetic variation [7], where it can be integrated with multiple data sources such as genome sequencing, QTL analysis, and disease relevant phenotypic data. For this application we focus on gene expression analysis conducted with a particular emphasis on those traits related to brain and behavior in the laboratory mouse.

A primary source of covariation in gene expression are single nucleotide polymorphisms (SNPs). Studies in genetical genomics [18] attribute variation and covariation in gene expression to the influence

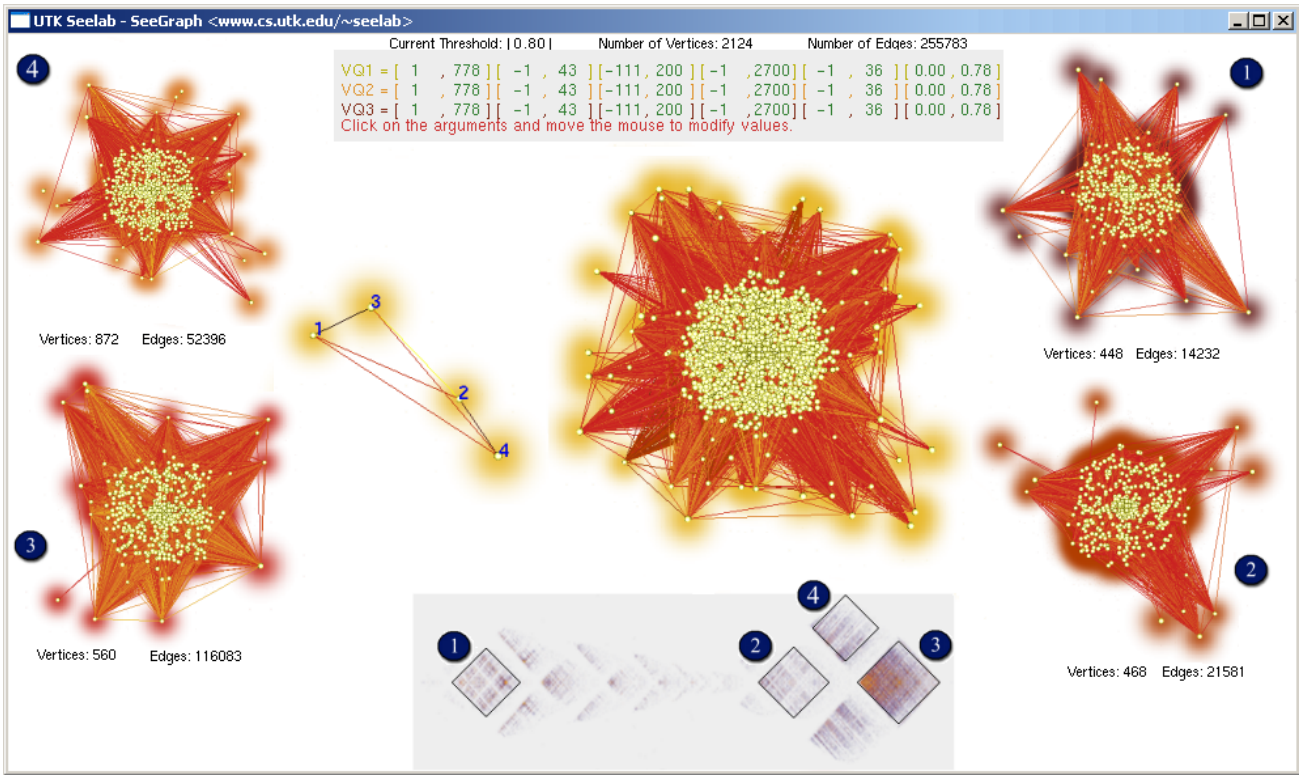

Fig. 5. BTD selections (bottom) qualitatively extract gene networks (sides), are rendered using dynamic level-of-detail (center left), and used for template-based classification of entire subgraphs in the original data (center right) for other regulatory mechanisms.

of these differences in DNA sequence. The use of recombinant inbred strains allows biologists to study replicate populations of mice with identical genomes. These populations allow indefinite aggregation of data across studies as new technologies for characterization of mice become available. When traits are assessed across genetically identical individuals, the correlations among traits are assumed to be due to common genetic regulation. By finding and analyzing statistical correlations between genotypes and phenotypes, geneticists hope to discover and interpret the network of causal genotype-phenotype relationships that determine a trait of interest.

Systems genetics research often follows a workflow of finding a gene network, finding regulators of that network, and then performing a focused gene perturbation experiment to determine the role of the associated network on gene expression or function. To begin, a "large" gene correlation graph must be sifted through, to find a highly connected subgraph which corresponds biologically to a gene network in which genes are expressed together, presumably to regulate or subserve a common function. They must then find a small set of causative genes, highly correlated with the subgraph and likely to regulate coexpression, to be used as targets of focused investigation. By manipulating the expression of these genes, the function of the gene network can be determined through observation of expressed phenotypes. Proof of causality occurs when the gene manipulations recapitulate network relations. It should be noted that while standards of "large" are highly application dependent, even graphs with less than 10k vertices exhibit a combinatorial space that is overwhelming and, indeed, presents a rather large and unique problem unlike dealing with volume datasets.

In this section, we showcase results for publicly available biological data which has been the subject of several previous studies. Whole brain mRNA gene expression data was obtained using the Affymetrix U74Av2 microarray for each of the strains in the BXD mouse population and subsequently processed using Robust Multi-Array (RMA) normalization [7]. Throughout the paper, we use Pearson's correlation over 7,443 genes of this dataset as our driving application. The associated database in our system is used for querying, interactive neural network training, and constructing dynamic level-of-detail (LoD) graph features; it contains information relating to typical systems genetic analyses for each gene such as: the chromosome, position (in megabases), paraclique membership and connectivity, broad-sense heritability indices, and QTL mapping [41] locations with p-values from QTL Reaper (sourceforge.net/projects/qtlreaper).

5.2 Discovery of Novel Networks

Typical biologists bring a large amount of domain-specific knowledge to their investigative process, for which many tools exist but are usually challenged by purely data driven investigation of networks. One approach to discovery of novel systems genetics networks is the use of computational tools which allow extraction of highly connected subgraphs in a qualitative fashion. By providing block tridiagonalization in which clusters around the diagonal constitute highly related genes, biologists can easily select potentially novel gene networks. Indeed, this $O(|V|^2)$ algorithm quickly extracts dense subgraphs and can be treated as a rough approximation to the NP-complete problem of paraclique enumeration in this context.

In Figure 5, the user has selected four BTD regions and dynamically generated a level-of-detail graph. As is expected, the selection 1 is most unrelated to the rightmost selections and therefore placed far away from the other selections with a negative correlation to selection 3. By selecting LoD vertices 2-4 as examples and 1 as a counterexample, neural network training on entire subgraphs is used to perform template-based search for similar genes in the original data. The resulting classification from the database information is the graph in the right center which has been extracted through the application of domain-specific knowledge in combination with several computational tools (BTD selection, LoD graphs, and NN template matching). This highly-connected subgraph contains genes which are similar to cliques 2 and 3 and bipartite structure 4 selected from the BTD and gives biologists a potentially novel network of two highly-related dense subgraphs to inspect for related function(s). This is currently be-

ing applied to create a comprehensive bipartite graph of gene networks (represented by LoD vertices) on one side and all network regulators (network interface genes) on the other. The ability to interactively and qualitatively search across multiple levels of detail has given biologists several tools for which they can not only solve current problems but also find new ways to address more difficult problems.

The central motivation of our system is to enable more in-depth and flexible expert-driven analysis by providing a diverse set of computational tools. However, there are other more established algorithmic tools, such as graph analysis, that are of value to scientific research. Those tools can be leveraged from within our system through our internal B-tree based data structure which allows queries from algorithmic solutions at a rate that facilitates realtime rendering and interaction. In the section below, we present a significant use case that demonstrates parts of our system to discover network interface genes.

5.3 Use Case: Discovery of Network Interface Genes

We now demonstrate our application with a biologically significant use case. Once gene networks have been extracted, it is of primary interest to determine the identity of the gene products that regulate these networks. Using either qualitative BTD selection or algorithmic network extraction, the total decomposition of a genetic correlation matrix into disjoint subgraphs can be achieved. With each disjoint subgraph treated as a structure, finding mRNA transcripts with strong correlations to multiple structures would lead to the discovery of "interface genes". These mRNA transcripts regulate expression of genes in those structures, and thereby couple multiple networks and biological processes. The detection of these transcripts and the analysis of their gene's regulatory polymorphisms could lead to the discovery of major genetic modifiers of large biological networks.

Our domain experts have found paraclique extraction to be the most useful and general algorithmic technique. Although choosing the proper threshold is a hard problem in general, by way of repetitive experimentation, statistical and combinatorial analysis, 0.85 is a preferred threshold for extracting paraclique in the dataset at hand [6]. Paracliques with more than ten vertices were extracted, resulting in thirty-seven dense subgraphs, and stored in the systems database. We show the largest and third largest paracliques corresponding to gene networks ready for further study in Figure 6.

In this use case, the challenge is to identify candidate genes that may be the common regulators of expression for a large number of other genes, and to determine which functional biological characteristics or disease related traits the network may be involved with. Some facts are known about this situation: 1) each gene's physical location within the genome is near the location of one of the genotype markers associated with the gene expression level; 2) the interface gene may be a member of one of the dense subgraphs, but it must be highly connected to members of both dense subgraphs; and 3) the biological regulator is likely to be in the same pathway as the genes it regulates.

By creating a level-of-detail graph in which the edge weight for supernodes is defined as the percent of connectivity to the adjacent gene or supernode, we can use the hot-cold coloring scheme to visually elucidate the correlation between multiple structures. This allows for an intuitive representation of network distance which can allow biologists to identify functional modules and their relationship to each other in forming the pathways underlying specific biological processes. The data is then filtered via threshold to retain only the strongest correlations between individual transcripts and entire networks. Once such a network is constructed, additional queries may be posed that relate to additional candidate genes and their association to the genetic polymorphisms that regulate the network.

In this use case, the technique described above was applied to visually identify a specific gene of interest whose transcript is connected to over 99% of both the largest and third largest paracliques as shown in Figure 6. The gene of interest, Pr1 or Tmem37 (Affymetric probe set ID 95464_at), has thus been implicated as part of the regulatory pathway that co-regulates the two large networks. This gene encodes a voltage-dependent calcium channel gamma subunit-like protein which modulates electrical properties of neuronal cells. It can be hypothesized that the paracliques are related to neuronal activity. Further testing was then accomplished through data export and communication capabilities to specialized bioinformatic pathway analysis tools. By further analysis of expression for the gene of interest, using GeneNetwork.org, it was revealed that its transcript abundance is correlated with stereotyped locomotor behavior induced in BXD mice by drugs such as cocaine [19] or methamphetamine [13].

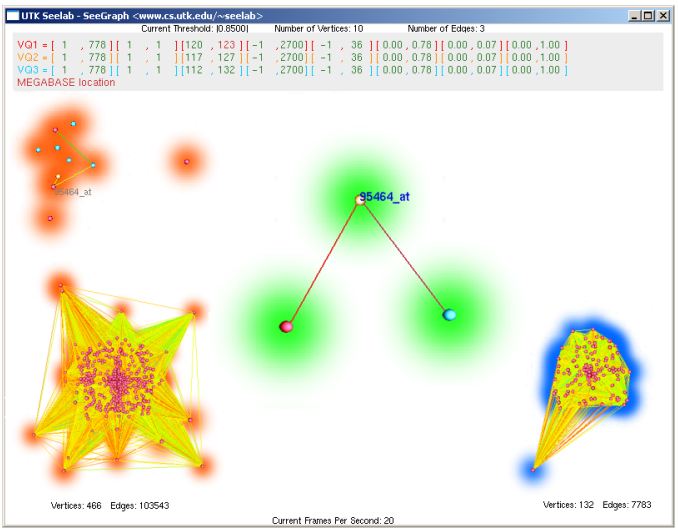

Fig. 6. In this screenshot, two gene networks (bottom left and right) have been discovered with a single putatively co-regulating gene as a potential target of knock-out study (center) with proximity information for other potential regulatory genes (top left) undergoing further study. This illustrates the discovery of candidate genes which can affect expression of several genes throughout the genome that play a role in the locomotor response of mice exposed to methamphetamine and cocaine.

The next stage in the workflow is to enumerate candidate genes which reside at chromosomal locations nearby the gene of interest's regulatory QTLs. To do this, we must first run an analysis of genotype associations to the expression of interface gene Pr1 in order to identify QTLs that regulate its expression. This analysis, which we have performed using GeneNetwork.org and linear modeling in SAS v9.1, reveals putative regulatory loci at specific locations on chromosomes 1, 2, 6 and 14. By using our tool's integrated data query capability, we can then highlight co-expressed genes that reside in regions provided by the other tools. The candidate genes may help regulate the networks since they are located in close physical proximity to one another, their transcripts are highly correlated to many paraclique members, and the QTL that regulates them also regulates many members of both paracliques. This leads to the identification of a limited number of candidate regulators including Pax3, Lama5, Mkrn2, and Dhrs4.

We have now derived testable hypotheses regarding the mechanisms by which the networks are co-regulated and can validate this new knowledge with in vivo experimentation. It follows from the analysis of co-expression that these two paracliques are controlled by a common genetic regulator. A high genetic correlation implies that the biomolecules represented by the connected vertices have values that are determined by the same genotype. Expression of the interface gene Pr1 can also be correlated with biological function and traced back to a regulatory QTL. Pr1 and the two dense subgraphs have been associated with specific traits measured in BXD mice. A small number of candidate regulators from positions near the regulatory QTL has also been identified. The resulting visualization reveals networks which go from locomotor responses to specific drugs down to the connected molecular pathways which underly them.

While Figure 6 represents only a few steps from a typical workflow, the result of this finding captures overwhelming complexity. As early forays into systems genetic analysis have demonstrated, biological processes such as mammalian behavior involve a complex inter-

play of expression from many hundreds of genes across multiple chromosomes and biological systems. For this reason, we expect similar linked views of multiple tools to be the norm for systems genetic visualization. The tool presented herein allows users to retain networks and their relationships as well as rapidly isolate genes and subgraphs based on their connectivity. The tools demonstrated represent a flexible and dynamic approach which allows users to scale up from single genes or traits of interest found in web based tools to results of global analyses such as clustering and high-performance combinatorial analyses.

6 Conclusion and Future Work

In conclusion, it has been shown that the integrated computational tools not only provide research scientists and analysts a way to visualize their data, but it also allows complex querying and filtering for drill-down, graphical analysis, and statistical output. Each of these are facilitated by combinations of a 3D spring-embedded layout, efficient B-tree processing, neural networks, matrix operations, and graph algorithms.

While our visualization tool enables significant biological discoveries, its full potential can only be leveraged when used in combination with mature applications and data management systems for genetical genomics datasets. For this purpose, our future work includes integration with the Gaggle API [33], a web-enabled, platform-independent, multi-application, data-sharing framework for widespread use among systems biologists available at gaggle.systemsbiology.org. In addition to the linked viewports framework, we are also considering domain-specific hybrid visualizations such as [15]. This tool is also currently being integrated as a correlation visualization tool to complement the genome-phenome data integration tools in the Ontological Discovery Environment (ODE) suite at ontologicaldiscovery.org. Ongoing work is currently being conducted with biologists to implement new functionality relating to domain-specific requests for handling linkage disequilibrium, QTL analysis, integration of genetic information across multiple scales, multiple time points, and different graph types.

Acknowledgments

The authors gratefully acknowledge R. Williams and colleagues for making their BXD brain gene expression data public and Shawn Ericson of GeNetViz for the three additional datasets used in Table 1. The authors wish to thank Dr. Michael Langston et al. for the paraclique algorithm used for processing of the biological data, Yihua Bai for her collaboration with the block tridiagonalization code, Stephen Pitts for co-expresion network analysis, and Arne Frick for the GEM3D source. The work is funded in part by NSF CNS-0437508, and through DOE SciDAC Institute of Ultra-Scale Visualization under DOE DE-FC02-06ER25778. Elissa J. Chesler is supported by NIH/NIAAA Integrative Neuroscience Initiative on Alcoholism (U01AA13499, U24AA13513, U01AA016662), NIH/NIDA R01DA020677, and NICHD R01HD052472-01.

References

[1] J. Abello and J. Korn. Mgv: A system for visualizing massive multidigraphs. *IEEE Trans. Visualization and Computer Graphics*, 8(1):21–38, Jan.-Mar. 2002.

[2] J. Abello, F. van Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Trans. Visualization and Computer Graphics*, 12(5):669–677, Sep.-Oct. 2006.

[3] O. Abiola, J. M. Angel, P. Avner, A. A. Bachmanov, and J. K. Belknap et al. The nature and identification of quantitative trait loci: a community's view. *Nature Reviews Genetics*, 4(11):911–916, 2003.

[4] D. Auber, Y. Chiricota, F. Jourdan, and G. Melancon. Multiscale visualization of small world networks. In *IEEE Syposium on Information Visualization*, pages 75–81, 2003.

[5] Y. Bai, W. N. Gansterer, and R. C. Ward. Block tridiagonalization of effectively sparse symmetric matrices. *ACM Trans. Math. Softw.*, 30(3):326–352, 2004.

[6] E. J. Chesler and M. A. Langston. Combinatorial genetic regulatory network analysis tools for high throughput transcriptomic data. In *RECOMB Satellite Workshop on Systems Biology and Regulatory Genomics*, pages 150–165, 2005.

[7] E. J. Chesler, L. Lu, S. Shou, Y. Qu, J. Gu, J. Wang, H. C. Hsu, J. D. Mountz, N. E. Baldwin, M. A. Langston, J. B. Hogenesch, D. W. Threadgill, K. F. Manly, and R. W. Williams. Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, 37(3):233–242, 2005.

[8] E. J. Chesler, J. Wang, L. Lu, Y. Qu, K. F. Manly, and R. W. Williams. Genetic correlates of gene expression in recombinant inbred strains: a relational model system to explore neurobehavioral phenotypes. *Neuroinformatics*, 1(4):343–357, 2003.

[9] R. W. Doerge. Mapping and analysis of quantitative trait loci in experimental populations. *Nature Reviews Genetics*, 3(1):43–52, 2002.

[10] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[11] D. H. Geschwind. Mice, microarrays, and the genetic diversity of the brain. *Proc. National Academy of Sciences*, 97(20):10676–10678, 2000.

[12] M. Glatter, C. Mollenhour, J. Huang, and J. Gao. Scalable data servers for large multivariate volume visualization. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1291–1298, 2006.

[13] J. E. Grisel, J. K. Belknap, L. A. O'Toole, and M. L. Helms et al. Quantitative trait loci affecting methamphetamine responses in bxd recombinant inbred mouse strains. *Journal of Neuroscience*, 17(2):745–754, 1997.

[14] S. Hachul and M. Junger. The fast multipole multilevel method. *GD '04: Proceedings of the Symposium on Graph Drawing*, pages 286–293, 2004.

[15] N. Henry, J. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.

[16] N. Henry and J. D. Fekete. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Trans. Visualization and Computer Graphics*, 12(5):677–685, Sep.-Oct. 2006.

[17] Z. Hu, J. Mellor, J. Wu, and C. DeLisi. Visant: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, pages 5–17, 2004.

[18] R. C. Jansen and J. P. Nap. Genetical genomics: the added value from segregation. *Trends in Genetics*, 17(7):388–391, 2001.

[19] B. C. Jones, L. M. Tarantino, L. A. Rodriguez, and C. L. Reed et al. Quantitative-trait loci analysis of cocaine-related behaviours and neurochemistry. *Pharmacogenetics*, 9(5):607–617, 1999.

[20] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989.

[21] M. Kreuseler and H. Schumann. A flexible approach for visual data mining. *IEEE Trans. Visualization and Computer Graphics*, 8(1):39–51, Jan.-Mar. 2002.

[22] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling emerging cyber-communities automatically. In *Proc. 8th Intl World Wide Web Conf.*, 1999.

[23] M. A. Langston, A. D. Perkins, A. M. Saxton, J. A. Scharff, and B. H. Voy. Innovative computational methods for transcriptomic data analysis. In *SAC'06: Proceedings of the 2006 ACM Symposium on Applied Computing*, pages 190–194, New York, NY, USA, 2006. ACM Press.

[24] L. Linsen, J. Locherbach, M. Berth, Jorg Bernhardt, and Dorte Becher. Differential protein expression analysis via liquid-chromatography/mass-spectrometry data visualization. In *IEEE Conference Visualization*, 2005.

[25] C. Mueller, B. Martin, and A. Lumsdaine. A comparison of vertex ordering algorithms for large graph visualization. *International Asia-Pacific Symposium on Visualization*, pages 141–148, 2007.

[26] P. Mutton and P. Rodgers. Spring embedder preprocessing for www visualization. *IEEE Syposium on Information Visualization*, 00:744–749, 2002.

[27] A. Noack. An energy model for visual graph clustering. *GD '04: Proceedings of the Symposium on Graph Drawing*, pages 425–436, 2004.

[28] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):693–700, 2006.

[29] J. Raymond, E. Gardiner, and P. Willett. Rascal: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644, 2002.

[30] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd Ed)*. Prentice Hall, 2002.

[31] P. Saraiya, Chris North, and Karen Duca. An evaluation of microarray visualization tools for biological insight. In *IEEE Symposium on Infor-*

mation Visualization, pages 1–8, 2004.

[32] P. T. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 11:2498–504, 2003.

[33] P. T. Shannon, D. J. Reiss, R. Bonneau, and N. S. Baliga. The gaggle: an open-source software system for integrating bioinformatics software and data sources. *BMC Bioinformatics*, 7:176, 2006.

[34] Z. Shen, K. L. Ma, and T. Eliassi-Rad. Visual analysis of large heterogeneous social networks by semantic and structure. *IEEE Trans. on Visualization and Computer Graphics*, 12(6):1427–1439, November/December 2006.

[35] L. Sheng, Z. M. Ozsoyoglu, and G. Ozsoyoglu. A graph query language and its query processing. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Austrialia*, pages 572–581. IEEE Computer Society, 1999.

[36] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages*, number UMCP-CSD CS-TR-3665, pages 336–343, College Park, Maryland 20742, U.S.A., 1996.

[37] B. Shneiderman. Network visualization by semantic substrates. *IEEE Trans. Visualization and Computer Graphics*, 12(5):733–741, Sep.-Oct. 2006.

[38] M. Tarini, P. Cignoni, and C. Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244, 2006.

[39] F. Y. Tzeng, E. B. Lum, and K. L. Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of IEEE Visualization '03*, pages 505–512, 2003.

[40] F. van Ham and J. van Wijk. Interactive visualization of small world graphs. In *IEEE Syposium on Information Visualization*, pages 199–206, 2004.

[41] J. Wang, R. W. Williams, and K. F. Manly. Webqtl: web-based complex trait analysis. *Neuroinformatics*, 1(4):299–308, 2003.

[42] B. Zhang, S. Kirov, J. Snoddy, and S. Ericson. Genetviz: A gene network visualization system. In *UT-ORNL-KBRIN Bioinformatics Summit*, 2005.