

Computational, Integrative and Comparative Methods for the Elucidation of Genetic Co-Expression Networks

Nicole E. Baldwin¹, Elissa J. Chesler², Stefan Kirov³, Michael A. Langston^{1*}, Jay R. Snoddy³, Robert W. Williams² and Bing Zhang³

¹ Dept of Computer Science, University of Tennessee, Knoxville, TN 37996, USA

² Dept of Anatomy and Neurobiology, University of Tennessee, Memphis, TN 38163, USA

³ Life Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

Abstract. Gene expression microarray data can be used for the assembly of genetic co-expression network graphs. Using mRNA samples obtained from recombinant inbred *Mus musculus* strains, it is possible to integrate allelic variation with molecular and higher-order phenotypes. The depth of quantitative genetic analysis of microarray data can be vastly enhanced utilizing this mouse resource in combination with powerful computational algorithms, platforms and data repositories. The resulting network graphs transect many levels of biological scale. This approach is illustrated with the extraction of cliques of putatively co-regulated genes and their annotation using gene ontology analysis and *cis*-regulatory element discovery. The causal basis for co-regulation is detected through the use of quantitative trait locus mapping.

1 Introduction

The purpose of this paper is to describe novel research combining

- emergent computational algorithms,
- high performance platforms and implementations,
- complex trait analysis and genetic mapping, and
- integrative tools for data repository and exploration.

In this effort we employ huge datasets extracted from a panel of recombinant inbred (RI) strains that were produced by crossing two fully-sequenced strains of C57BL/6J and DBA/2J mice [45]. The essential feature of these isogenic RI strains is that they are a genetic mapping panel. They can therefore be used to convert associative networks into causal networks. This is done by finding those polymorphic genes that actually produce natural endogenous variation in gene networks [4]. In this regard, RI strains differ fundamentally from standard inbred strains, knockout strains, transgenic lines and mutants. This approach, termed quantitative trait locus (QTL) mapping, is usually limited to a single continuously distributed trait such as brain weight or neuron number [34], or a behavioral trait such as open-field activity [26]. In this paper, however, we map regulators of entire networks, clusters, and cliques [20].

* Communicating author: langston@cs.utk.edu

We employ combinatorial algorithms and graph theory to reduce the high dimensionality of this megavariable data. Advances in clique finding algorithms generate highly-distilled gene sets, which we interpret using novel, integrative bioinformatics resources. See Figure 1. Tools of choice include GeneKeyDB [31], WebQTL [41] and GOTM [46].

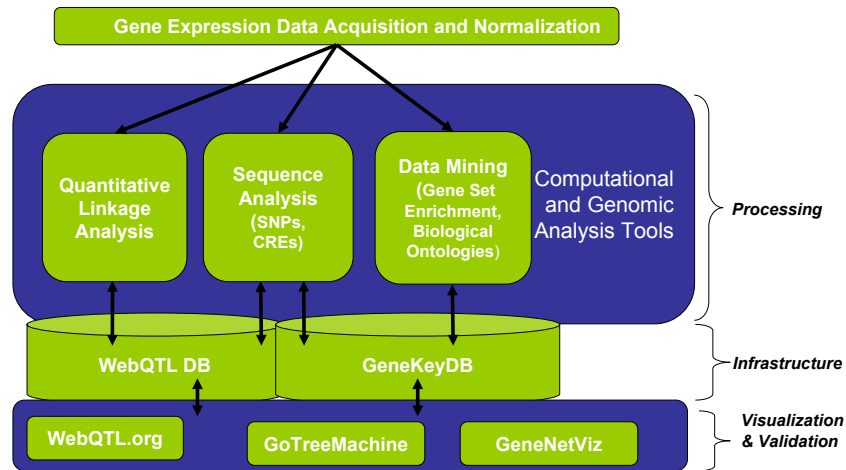


Figure 1. A process overview.

2 QTL Mapping

Experimental Design. Microarrays provide an extraordinary efficient tool with which to obtain very large numbers of quantitative assays from tissue samples. For example, using the Affymetrix M430 arrays one can obtain approximately 45,000 measurements of relative mRNA abundance from a whole tissue such as the brain or from a single cell population, such as hematopoietic stem cells. In much of our recent work we have used the Affymetrix U74Av2 array to estimate the abundance of 12,422 transcripts from the mouse brain. The design of our experiments is quite simple. We extract mRNA from three litter-mate mice of the same strain and sex, pool the mRNA, and hybridize the sample to the microarray. We do this three times for each strain and sex (independent biological replicates). There is no intentional experimental manipulation of the animals or strains of mice. The essential feature to note in our experimental design is that the isogenic strains of mice that we study are all related yet genetically unique from one another.

RI Strains. These related strains of mice collectively form what is called a “mapping panel.” The strain set that we use is called the BXD mapping panel because all of the 32 strains originate from the same two original progenitor strains: C57BL/6J (the B strain) and DBA/2J (the D strain). The 32 derivative BXD strains are genetic mosaics of the two parental strains. If one were to pick one of the 32 strains at random and

examine a piece of one particular chromosome, there would be an approximately 0.5 probability of that piece having descended down through the generations from the B or the D parental strain. If one looked at the same part of a particular chromosome in all 32 strains one would end up with a vector of genotypes. For example, the tip of chromosome 1 of BXD strains number 11 through 16 might read BDBBDD. Thus there are 2^{32} or 4.29×10^9 possible combinations of these vectors of genotypes. The chromosomes of individual BXD strains actually consist of very long stretches of B-type or D-type chromosomes. The average stretch is almost 50 million base pairs long. The entire set of 32 BXD strains incorporate sufficient recombinations between the parental chromosomes to encode a total of about 2^{11} locations across the mouse genome. This means that in a best case one can only specify locations to about 1.27×10^6 bp. 1.27 million bp will typically contain 17 genes. (Of course, the locations of these recombinations is a close to a random Poisson process.)

Unlike other recombinant cross progeny used for QTL analysis, all of the BXD strains are fully inbred. To make these recombinant inbred (RI) strains, full siblings were mated successively for 20 generations to produce each of the 32 strains. This has been an expensive process which has led to several strain sets available to the research community by commercial suppliers (The Jackson Laboratory, www.jax.org) or the originating laboratory. Making fully RI strains from a breeding cross between the C57BL/6J and DBA/2J parental strains has taken about eight years. These strains have been used for over 20 years for the detection of QTLs in a wide range of phenotypes [39]. An additional 45 strains have been generated recently [35].

Finding The Genetic Regulatory Locus. There are numerous genetic polymorphisms (allelic variants) that exist between the two parental strains. As an illustrative example, consider two alleles of a gene coding for a product that is absolutely required to deposit pigmentation in the hair and eye. Further let us assume that these two alleles act as a digital switch: the B allele inherited from C57BL/6J is the active form and the D allele inherited from DBA/2J is the inactive form.

In the D-state, the mice are albinos; in the B-state, they are normally pigmented. The vector of this phenotype across the strains might look like this PWWWWP (P, pigmented; W, white) for strains 11 through 16. A simple comparison of this vector of phenotypes to the vector of genotypes on the tip of chromosome 1 (BDBBDD) clearly rules out this location since the vectors do not match particularly well. A vector of genotypes on chromosome 7 at 77 million based pairs (Mb), however, is a perfect fit: BDDDDDB. Depending on the coding convention that we use, this will give either a correlation of 1 or -1. This is the central concept of mapping simple one-gene (monogenic) traits to discover one or more genotype vectors (markers) that have tight quantitative associations with the phenotype vectors across a large mapping panel. Recall however that our particular 32 strain genotype vector only provides enough resolution to get us down to a genetic neighborhood containing about 17 genes. We call this a *genetic locus* (or sometimes a gene locus), although we have to remember that we cannot yet assert which gene in this locus is actually the pigmentation switch.

Up to this point we have considered a trait that can be easily dichotomized. The vast majority of traits in which we are interested, however, are spread continuously over a

broad range of values that often approximates a Gaussian distribution. These traits are frequently controlled by more than a single genetic locus. Furthermore, environmental factors typically introduce a complementary non-genetic source of variance to a trait measured across a genetically diverse group of individuals. Consider, for example, body weight. This is a classic example of a complex highly variable population trait that is due to a multifactorial admixture of genetic factors, environmental factors, and interactions between genes and environment. Even a trait such as the amount of mRNA expressed in the brains of mice and measured using microarrays is a very complex trait. We refer the interested reader to our previous work [20,42] for more information on this subject. The abundance of mRNA is influenced by rates of transcription, rates of splicing and degradation, stages of the circadian cycle and a variety of other environmental factors. Many of these influences on transcript abundance exert their effects via the actions of other genes. QTL mapping of mRNA abundance allows one to detect these genetic sources of variation in gene expression [15, 20, 37, 42].

3 Computational Methods

A Clique-Centric Approach. Current high-throughput molecular assays generate immense numbers of phenotypic values. Billions of individual hypotheses can be tested from a single BXD RI transcriptome profiling experiment. QTL mapping, however, tends to be highly focused on small sets of traits and genes. Many public users of our data resources approach the data with specific questions of particular gene-gene and/or gene-phenotype relationships [19]. These high-dimensional data sets are best understood when the correlated phenotypes are determined and analyzed simultaneously. Data reduction via automated extraction of co-regulated gene sets from transcriptome QTL data is a challenge. Given the need to analyze efficiently tens of thousands of genes and traits, it is essential to develop tools to extract and characterize large aggregates of genes, QTLs, and highly variable traits.

There are advantages to placing our work in a graph-theoretic framework. This representation is known to be appropriate for probing and determining the structure of biological networks including the extraction of evolutionarily conserved modules of co-expressed genes. See, for example, [5, 8, 33]. A major computational bottleneck in our efforts to identify sets of putatively co-regulated genes is the search for cliques, a classic graph-theoretic problem. Here a gene is denoted by a vertex, and a co-expression value is represented by the weight placed on an edge joining a pair of vertices. Clique is widely known for its application in a variety of combinatorial settings, a great number of which are relevant to computational molecular biology. See, for example, [38]. A considerable amount of effort has been devoted to solving clique efficiently. An excellent survey can be found in [14].

In the context of microarray analysis, our approach can be viewed as a form of clustering. A wealth of clustering approaches has been proposed. See [11–13, 28, 29] to list just a few. Here the usual goal is to partition vertices into disjoint subsets, so that the genes that correspond to the vertices within each subset display some measure of homogeneity. An advantage clique holds over most traditional clustering methods is that cliques need not be disjoint. A vertex can reside in more than one (maximum or maxi-

mal) clique, just as a gene product can be involved in more than one regulatory network. There are recent clustering techniques, for example those employing factor analysis [6], that do not require exclusive cluster membership for single genes. Unfortunately, these tend to produce biologically uninterpretable factors without the incorporation of prior biological information [27]. Clique makes no such demand. Another advantage of clique is the purity of the categories it generates. There is considerable interest in solving the dense k -subgraph problem [23]. Here the focus is on a cluster's edge density, also referred to as clustering coefficient, curvature, and even cliquishness [36, 44]. In this respect, clique is the "gold standard." A cluster's edge density is maximized with clique by definition.

The inputs to clique are an undirected graph G with n vertices, and a parameter $k \leq n$. The question asked is whether G contains a clique of size k , that is, a subgraph isomorphic to K_k , the complete graph on k vertices. The importance of K_k lies in the fact that each and every pair of its vertices is joined by an edge. Subgraph isomorphism, clique in particular, is \mathcal{NP} -complete. From this it follows that there is no known algorithm for deciding clique that runs in time polynomial in the size of the input. One could of course solve clique by generating and checking all $\binom{n}{k}$ candidate solutions. But this brute force approach requires $O(n^k)$ time, and is thus prohibitively slow, even for problem instances of only modest size.

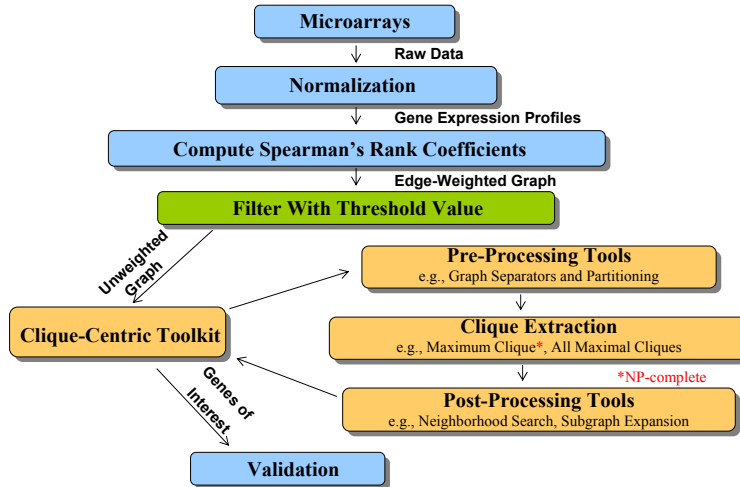


Figure 2. The clique-centric toolkit, and its use in microarray analysis.

Our methods are employed as illustrated in Figure 2. We shall concentrate our discussion on the classic maximum clique problem. Of course we also must handle the related problem of generating all maximal cliques once a suitable threshold has been chosen, which is itself often a function of maximum clique size. There are a variety of other issues dealing with pre- and post-processing. Although we do not explicitly deal with them in the present paper, they are for the most part quite easily handled and

are dwarfed by the computational complexity of the fundamental clique problem at the heart of our method.

Fixed-Parameter Tractability. The origins of *fixed parameter tractability* can be traced at least as far back as the work done to show, via the Graph Minor Theorem, that a variety of parameterized problems are tractable when the relevant input parameter is fixed. See, for example, [24, 25]. Formally, a problem is FPT if it has an algorithm that runs in $O(f(k)n^c)$, where n is the problem size, k is the input parameter, and c is a constant independent of both n and k [21]. Unfortunately, clique is not FPT unless the \mathcal{W} hierarchy collapses. (The \mathcal{W} hierarchy, whose lowest level is FPT, can be viewed as a fixed-parameter analog of the polynomial hierarchy, whose lowest level is \mathcal{P} .) Thus we focus instead on clique’s complementary dual, the *vertex cover* problem. Consider \overline{G} , the complement of G . (\overline{G} has the same vertex set as G , but edges present in G are absent in \overline{G} and vice versa.) As with clique, the inputs to vertex cover are an undirected graph G with n vertices, and a parameter $k \leq n$. The question now asked is whether G contains a set C of k vertices that covers every edge in G , where an edge is said to be covered if either or both of its endpoints are in C . Like clique, vertex cover is \mathcal{NP} -complete. Unlike clique, however, vertex cover is also FPT. The crucial observation here is this: a vertex cover of size k in \overline{G} turns out to be exactly the complement of a clique of size $n - k$ in G . Thus, we search for a minimum vertex cover in \overline{G} , thereby finding the desired maximum clique in G . Currently, the fastest known vertex cover algorithm runs in $O(1.2852^k + kn)$ time [18]. Contrast this with $O(n^k)$. The requisite exponential growth (assuming $\mathcal{P} \neq \mathcal{NP}$) is therefore reduced to a mere *additive* term.

Kernelization, Branching, Parallelization and Load Balancing. The initial goal is to reduce an arbitrary input instance down to a relatively small computational kernel, then decomposing it so that an efficient, systematic search can be conducted. Attaining a kernel whose size is quadratic in k is relatively straightforward [16]. Ensuring a kernel whose size is linear in k has until recently required much more powerful and considerably slower methods that rely on linear programming relaxation [30, 32].

In [1], we introduced and analyzed a new technique, termed *crown reduction*. A *crown* is an ordered pair (I, H) of subsets of vertices from G that satisfies the following criteria: (1) $I \neq \emptyset$ is an independent set of G , (2) $H = N(I)$, and (3) there exists a matching M on the edges connecting I and H such that all elements of H are matched. H is called the *head* of the crown. The *width* of the crown is $|H|$. This notion is depicted in Figure 3.

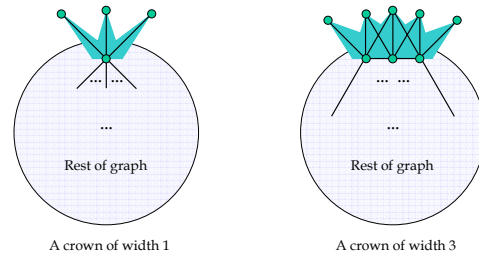


Figure 3. Sample crown decompositions.

Theorem[1] Any graph G can be decomposed into a crown (I, H) for which H contains a minimum-size vertex cover of G and so that $|H| \leq 3k$. Moreover, the decomposition can be accomplished in $O(n^{\frac{3}{2}})$ time.

The problem now becomes one of exploring the kernel efficiently. A branching process is carried out using a binary search tree. Internal nodes represent choices; leaves denote candidate solutions. Subtrees spawned off at each level can be explored in parallel. The best results have generally been obtained with minimal intervention, in the extreme case launching secure shells (SSHs) [2]. To maintain scalability as datasets grow in size and as more machines are brought on line, some form of dynamic load balancing is generally required. We have implemented such a scheme using sockets and process-independent forking. Results on 32-64 processors in the context of motif discovery are reported in [7]. Large-scale testing using immense genomic and proteomic datasets are reported in [3].

4 Sample Computational Results

We are now able to solve real, non-synthetic instances of clique on graphs whose vertices number in the thousands. (Just imagine a straightforward $O(n^k)$ algorithm on problems of that size!)

To illustrate, we recently solved a problem on *Mus musculus* neurogenetic microarray data with 12,422 vertices (probe-sets). With expression values normalized to $[0,1]$ and the threshold set at 0.5, the clique we returned (via vertex cover) denoted a set of 369 genes that appear experimentally to be co-regulated. This took a few days to solve even with our best current methods. Yet solving it at all was probably unthinkable just a short time ago. After iterating across several threshold choices, a value of .85 was selected for detailed study. For this graph, G , the maximum clique size is 17. Because it is difficult to visualize G , we

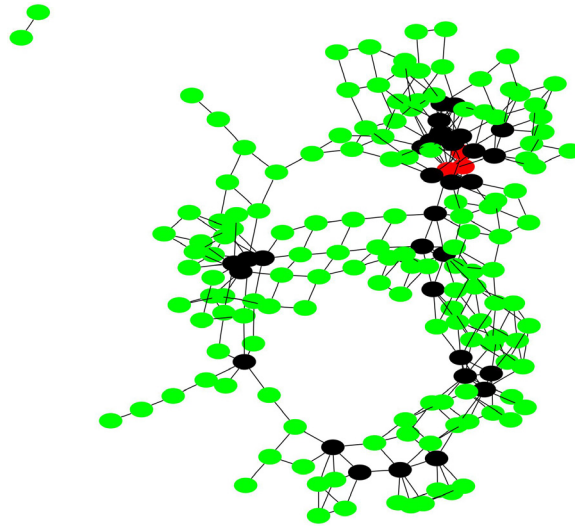


Figure 4. A clique intersection graph for a large microarray dataset.

employ a clique intersection graph, C_G , as follows. Each maximal clique of size 15 or more in G is represented by a vertex in C_G . An edge connects a pair of vertices in C_G if and only if the intersection of the corresponding cliques in G contains at least 13 members. C_G is depicted in Figure 4, with vertices representing cliques of size 15 (in green), cliques of size 16 (in black) and cliques of size 17 (in red). One rather surprising result is that the gene found most often across large maximal cliques is *Veli3* (aka *Lin7c*). This

appears not to be due to some so-called “housekeeping” function, but instead because the relatively unstudied *Veli3* is in fact central to neurological function [10, 17].

5 Cliques of Highly Correlated Transcripts and Behavioral Phenotypes

We can infer that co-expression of genes in mice of common genetic background is due to a shared regulatory mechanism, because the correlation is between trait means from different lines of mice, rather than from within an experimental group. Clique membership alone does not tell us anything about the basis of common genetic regulation. By combining clique data with QTL analysis, the regulatory loci underlying the shared genetic mediation of gene expression can be identified. This allows us to determine the impact of genetic variability in gene expression on other biological processes. Using the afore-mentioned stringent correlation threshold of 0.85, the most highly connected transcript identified was that of *Veli3*. One maximal clique of seventeen highly-associated transcript abundances includes several nuclear proteins. A single principal component of these transcripts accounts for 95% of the total genetic sample variance.

No single QTL can be found for the members of this clique, but a multiple QTL mapping analysis reveals an interacting pair of loci on chromosomes 12 and X, at markers *D12Mit46* (29.163 Mb) and *DXMit117* (110.670 Mb). See Figure 5, which shows the results from a search for pairs of genetic loci that modulate expression of a clique. Chromosomes 12, 19 and X are shown. Likelihood ratio statistics for multiple QTL models are plotted on the pseudo-color scale. The upper left triangle shows fit results for an interaction model, and the lower right triangle shows fit results for a model containing both additive and interaction effects of the two loci. The joint model including markers on 12 and X is significant ($p < .05$) by permutation analysis. A ‘D’ allele at both loci results in low levels of the phenotype and a ‘B’ allele at both loci results in a high level of the phenotype. The chromosome 12 locus is the physical location of two clique members: B Cell Receptor associated protein *Bcap29*, and myelin tran-

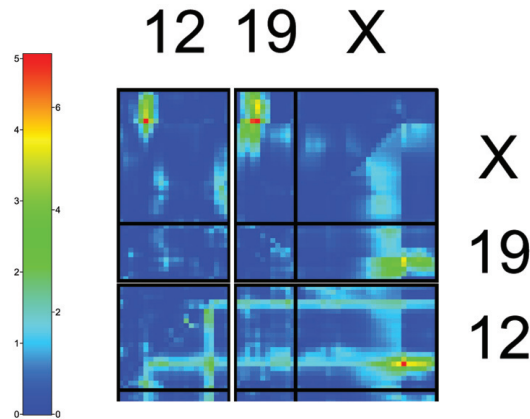


Figure 5. Multiple QTL mapping analysis. In the upper left triangle, a pseudo-color plot shows the likelihood ratio statistic for each two-locus interaction. In the lower right triangle, a likelihood ratio statistic is depicted for the full two-locus model, which fits additive effects for each pair of loci and their interaction. Significance was assessed by genome-wide permutation analysis.

scription factor 1-like protein, *Myt1l*. An interesting functional and positional candidate at the Chr X locus is Integral membrane protein 1, *Itm1*. While this is not a member of the clique we are analyzing here, it does frequently co-occur along with *Veli3* in many maximal cliques.

In addition to tens of thousands of transcript traits, we have assembled a database of over 600 organismic phenotypes, including many morphometric traits. An understanding of the genetic control of these phenotypes can help explain their evolution. In the present example, we have found the previously mentioned clique to associate with behavioral and metabolic phenotypes. This clique correlates with both midbrain iron levels, and locomotor behavior. Interestingly, one of the clique members, *Gs2na* (GS2 nuclear autoantigen), that at 46.048 Mb on chromosome 12 is a little too far afield to be a positional QTL candidate gene, is a striatin family member and the negative correlation we observe between clique expression and locomotor behavior is consistent with literature reports of locomotor impairment associated with decreased levels of striatin [9]. At this point research becomes hypothesis driven; indeed, the result of this collaborative analysis is a simple testable hypothesis, extracted from many billions of data relations. We are now in the process of evaluating the hypothesis that genetic variation in iron metabolism influences expression of the *Veli3* clique members in the brain and consequently affects locomotor activity.

6 Integrative Genomic Data Mining

GeneKeyDB. High-throughput, high-volume data like these gene expression data from genetically-variant mice should be examined in a biological context. The subsets of interesting genes must be analyzed, in part, by using existing information that describe the role these genes play in biological processes. When computing and navigating these data in terms of graphs and networks, we need to have a way to manipulate various kinds of metadata about sets of genes and gene products.

We have developed several such tools for genes and gene products that are discovered from the clique and QTL data analysis. Most gene-centered data resources that are generally available for retrieving metadata about genes are displayed and manipulated in a one-gene-at-a-time format (e.g., Entrez Gene). We have developed a lightweight data mining environment that allows the automated integration of various types of data about sets of genes. This environment is called *GeneKeyDB*. This system includes metadata from GenBank, Entrez Gene, Ensembl, and several other well-established biological databases. GeneKeyDB uses a relational database backend to facilitate interactions between tools and data. Among other functions, GeneKeyDB automatically converts the different database identifiers from these different databases. It can, for example, start with GenBank cDNA identifiers, locate the “sequence feature” information from genome sequence data entries, and assist in retrieving sequences for detailed analyses. GeneKeyDB can also obtain various kinds of homologs, functional annotation, or other attributes of genes and gene products. Furthermore, it serves as a repository for results that are created by our computational tools.

We have devised two types of computational analyses that are supported by the underlying GeneKeyDB system.

GoTreeMachine. Gene Ontology (GO) produces structured, precisely defined, common, controlled vocabulary for describing the roles of genes and gene products [22]. GO has been used frequently in the functional profiling of high-throughput data. We have developed a web-based tool, GOTM, for the analysis and visualization of sets of genes using Gene Ontology hierarchies [46]. Besides being a stand-alone functional profiling tool, GOTM can work with other computational tools for gene set centered integrated analysis. GOTM has been employed in various ways in this effort. This includes WeBQTL’s use of GOTM to narrow down candidate gene lists and generate functional profiles for genes in a relevance network or genes correlated to complex phenotypes.

We use ontology analysis to evaluate the functional significance of the cliques found by our graph algorithms, and prioritize the cliques for further study. Figure 6 depicts a clique of size eight that was detected within the gene co-expression network constructed using the microarray data from the RI mouse lines. The five green vertices denote genes that belong to the GO functional category of “DNA binding.” The red vertices denote genes that either have no annotation or are annotated as function “unknown.” If we randomly pick five genes from all annotated genes on the microarray, the expected frequency of genes in the category “DNA binding” is only 0.9. The chance of finding all five genes in the category “DNA binding” is $p = 0.00051$ as calculated by the hypergeometric test implemented in GOTM. Out of the 5227 maximal cliques we generated, ontology analysis has detected a total of 342 of them that are significantly enriched in one or more GO categories ($p < 0.01$). The clique shown in Figure 6 has a p value less than 0.001, and is one of several we are studying. Note the presence of the gene *Veli3*.

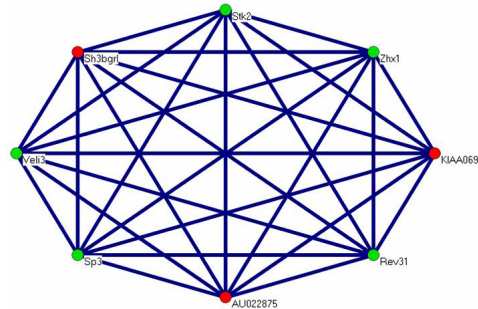


Figure 6. A relevant clique containing *Veli3*.

Batch Sequence Analysis. We are also deploying integrative methods that attempt to predict *cis*-Regulatory Elements (CREs) in the upstream regions from sets of genes that are putatively co-regulated. These CREs are thought to be the DNA sites to which protein regulatory transcription factors preferentially bind in promoters or enhancers and exert regulatory control of gene expression. We are combining a number of analyses to look at sets of CREs that are found in a subset of genes that seems to show strong co-regulation and are consistent with the clique and QTL data.

We have assembled a pipeline, BSA, that can retrieve the sequence data for the target genes and their orthologous counterparts in other chordate organisms. This pipeline carries out a number of processes that enable us to use both co-regulated gene sets and phylogenetic footprinting in an integrated pipeline to identify putative CREs. An important advantage of the pipeline is its ability to define the evolutionarily conserved non-coding sequences, which are thought to contain most of the CREs [43]. This should

substantially reduce the noise levels. BSA can be carried out in a high throughput, automated process because of the underlying GeneKeyDB infrastructure. BSA is routinely using both MEME and MAST as part of the sequence analysis, but other motif finding and searching methods are under development. A set of CRE motifs can be found in cliques or other interesting subsets of genes with motif searches (like MEME). We can then use MAST or similar searching tools to take those sets of putative CREs to do a global search for all possible targets in a database that contains promoter sequences from all human, mouse, and rat genes. The latter step could help define new genes that are targets of a gene regulatory network that were not initially identified. The BSA pipeline stores its results in the GeneKeyDB relational database.

7 Summary and Directions for Future Research

Our current work demonstrates the use of clique to extract signal from large genetic correlation matrices. We also employ genome-scale tools to interpret the shared molecular function, biological process, cellular localization and sequence motifs of clique members. Despite what has been accomplished in the BXD lines, the size of existing RI strain sets limits the power and resolution of this technique. The Complex Trait Consortium plans to expand this set with the development of a 1024 RI strain panel [40]. The creation of this resource will greatly increase the depth of our analysis. The breadth of the analysis can be expanded almost indefinitely. Although the work we have described here has been restricted to the analysis of gene expression microarray phenotypes, any attribute of these strains that can be measured can in principle be incorporated into the genetic correlation matrix. We already have a wealth of data on micro- and macro-scale biological phenotypes ranging from cellular responses to behavior. Novel high-throughput molecular phenotypes will greatly expand this collection. To accommodate such vast increases in data dimensionality, we are currently in the process of porting our codes to supercomputers at Oak Ridge National Laboratory (ORNL). These are difficult tasks indeed, given the many novel features of our algorithms. Great care is required to manage processor and memory resources. Load balancing can be especially problematic [7]. Initial targets include a 256-node SGI Altix and a 256-node Cray X1. In the longer term, we aim to employ the tremendously more powerful machines now under construction and awarded to ORNL in the recent competition to build the Nation's next leadership class computing facility for science. We believe that with our algorithms and these platforms we can solve problem instances previously considered hopelessly out of reach.

Acknowledgments

This research has been supported in part by the National Science Foundation under grants CCR-0075792 and CCR-0311500; by the National Institute of Mental Health, National Institute on Drug Abuse, and the National Science Foundation under award P20-MH-62009; by the National Institute on Alcohol Abuse and Alcoholism under INIA grants P01-Da015027, U01-AA013512-02, U01-AA13499 and U24-AA13513; by the Office of Naval Research under grant N00014-01-1-0608; and by the Department of Energy under contracts DE-AC05-00OR33735 and DE-AC05-4000029264.

We wish to thank Drs. Lu Lu and Kenneth Manly for helping develop data sets, analytic tools and the WebQTL website www.WebQTL.org. We also wish to express our appreciation to Drs. Mike Fellows and Henry Suters for helping develop fast kernelization alternatives and to Dr. Faisal Abu-Khzam for greatly improved branching implementations.

References

1. F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *Proceedings, Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2004.
2. F. N. Abu-Khzam, M. A. Langston, and P. Shanbhag. Scalable parallel algorithms for difficult combinatorial problems: A case study in optimization. In *Proceedings, International Conference on Parallel and Distributed Computing and Systems (PDCS)*, pages 563–568, 2003.
3. F. N. Abu-Khzam, M. A. Langston, P. Shanbhag, and C. T. Symons. Scalable parallel algorithms for FPT problems. Technical Report UT-CS-04-524, Department of Computer Science, University of Tennessee, 2004.
4. D. W. Airey, S. Shou, L. Lu, , and R. W. Williams. Genetics sources of individual differences in cerebellum. *The Cerebellum*, 1:233–240, 2002.
5. U. Alon. Biological networks: the tinkerer as an engineer. *Science*, 301:1866–1867, 2003.
6. O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97:10101–10106, 2000.
7. N. E. Baldwin, R. L. Collins, M. A. Langston, M. R. Leuze, C. T. Symons, and B. H. Voy. High performance computational tools for motif discovery. In *Proceedings, IEEE International Workshop on High Performance Computational Biology (HiCOMB)*, 2004.
8. A-L Barabási and Z. N. Oltvai. Network biology: Understanding the cell’s functional organization. *Nature Reviews Genetics*, 5:101–113, 2004.
9. M. Bartoli, J. P. Ternaux, C. Forni, P. Portalier, P. Salin, M. Amalric, and A. Monneron. Down-regulation of striatin, a neuronal calmodulin-binding protein, impairs rat locomotor activity. *Journal of Neurobiology*, 40(2):234–243, 1999.
10. C. Becamel, G. Alonso, N. Galeotti, E. Demey, P. Jouin, C. Ullmer, A. Dumuis, J. Bock-aert, and P. Marin. Synaptic multiprotein complexes associated with 5-HT(2C) receptors: a proteomic approach. *EMBO Journal*, 21:2332–2342, 2002.
11. A. Bellaachia, D. Portnoy, Y. Chen, and A. G. Elkahoul. E-CAST: A data mining algorithm for gene expression data. In *Workshop on Data Mining in Bioinformatics*, pages 49–54, 2002.
12. A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, pages 54–64, 2000.
13. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.
14. I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 4. Kluwer Academic Publishers, 1999.
15. R. B. Brem, G. Yvert, R. Clinton, and L. Kruglyak. Genetic dissection of transcriptional regulation in budding yeast. *Science*, 296:752–755, 2002.

16. J. F. Buss and J. Goldsmith. Nondeterminism within \mathcal{P} . *SIAM Journal on Computing*, 22:560–572, 1993.
17. S. Butz, M. Okamoto, and T. C. Sudhof. A tripartite protein complex with the potential to couple synaptic vesicle exocytosis to cell adhesion in brain. *Cell*, 94:773–782, 1998.
18. J. Chen, I. Kanj, and W. Jia. Vertex cover: further observations and further improvements. *Journal of Algorithms*, 41:280–301, 2001.
19. E. J. Chesler, L. Lu, J. Wang, R. W. Williams, and K. F. Manly. WebQTL: rapid exploratory analysis of gene expression and genetic networks for brain and behavior. *Nature Neuroscience*, 7:485–486, 2004.
20. 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*, 1:343–357, 2003.
21. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
22. M. Ashburner *et al.* Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
23. U. Feige, D. Peleg, and G. Kortsarz. The dense k -subgraph problem. *Algorithmica*, 29:410–421, 2001.
24. M. R. Fellows and M. A. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the ACM*, 35:727–739, 1988.
25. M. R. Fellows and M. A. Langston. On search, decision and the efficiency of polynomial-time algorithms. *Journal of Computer and Systems Sciences*, 49:769–779, 1994.
26. J. Flint. Analysis of quantitative trait loci that influence animal behavior. *Journal of Neurobiology*, 54:46–77, 2003.
27. M. Girolami and R. Breitling. Biologically valid linear factor models of gene expression. *Bioinformatics*, 2004. to appear.
28. P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, 1997.
29. E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrachs, and R. Shamir. An algorithm for clustering cDNAs for gene expression analysis. In *RECOMB*, pages 188–197, 1999.
30. S. Khuller. The vertex cover problem. *ACM SIGACT News*, 33:31–33, June 2002.
31. S. Kirov, X. Peng, E. Baker, D. Schmoyer, B. Zhang, and J. R. Snoddy. *GeneKeyDB: A lightweight, gene-centric, relational database to support data mining environments for sets of genes and gene products*. See <http://genereg.ornl.gov/gkdb>.
32. G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
33. Z. N. Oltvai and A.-L. Barabási. Systems biology. Life’s complexity pyramid. *Science*, 298:763–764, 2002.
34. J. Peirce, E. J. Chesler, R. W. Williams, and L. Lu. Genetic architecture of the mouse hippocampus: Identification of gene loci with regional effects. *Genes, Brain and Behavior*, 2:238–252, 2003.
35. J. L. Peirce, L. Lu, J. Gu, L. M. Silver, and R. W. Williams. A new set of BXD recombinant inbred lines from advanced intercross populations in mice. *BMC Genetics*, 29, 2004.
36. J. Rougemont and P. Hingamp. DNA microarray data and contextual analysis of correlation graphs. *BMC Bioinformatics*, 4, 2003.
37. E. E. Schadt, S. A. Monks, T. A. Drake, A. J. Luskis, N. Che, V. Colinayo, T. G. Ruff, S. B. Milligan, J. R. Lamb, G. Cavet, P. S. Linsley, M. Mao, R. B. Stoughton, and S. H. Friend. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422:297–302, 2003.
38. J. C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, 1997.

39. B. A. Taylor, C. Wnek, B. S. Kotlus, N. Roemer, T. MacTaggart, and S. J. Phillips. Genotyping new bxd recombinant inbred mouse strains and comparison of bxd and consensus maps. *Mammalian Genome*, 10:335–348, 1999.
40. G. Vogel. Genetics. scientists dream of 1001 complex mice. *Science*, 301:456–457, 2003.
41. J. Wang, R. W. Williams, and K. F. Manly. *WebQTL*. See www.webqtl.org.
42. J. Wang, R. W. Williams, and K. F. Manly. Webqtl: Web-based complex trait analysis. *Neuroinformatics*, 1:299–308, 2003.
43. W. W. Wasserman, M. Palumbo, W. Thompson, J. W. Fickett, and C. E. Lawrence. Human-mouse genome comparisons to locate regulatory sites. *Nature Genetics*, 26(2):225–228, 2000.
44. D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
45. R. W. Williams, J. Gu, S. Qi, and L. Lu. The genetic structure of recombinant inbred mice: High-resolution consensus maps for complex trait analysis. *Genome Biology*, 2:0046.1–0046.18, 2001.
46. B. Zhang, D. Schmoyer, S. Kirov, and J. Snoddy. GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using gene ontology hierarchies. *BMC Bioinformatics*, 5, 2004.