Using Visualization and Automation to Accelerate Genetics Discovery

Ross Eugene Curtis

December 2011
CMU-CB-11-103

Lane Center for Computational Biology
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213

Thesis Committee:
Gregory Cooper, Chair
Kathryn Roeder
Daniel Weeks
Sally Wenzel
Eric P Xing, Advisor

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

This research was sponsored by the National Science Foundation under contract no. DBI-0640543 and IIS-0713379 and the National Institutes of Health under contract no. 1R01GM087694, 1R01GM093156 and 0015699 and the Defense Advanced Research Projects Agency under contract no. Z931302. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official polices, either expressed or implied, or any sponsoring institution, the U.S. government, or any other entity.

Keywords: Visualization, automation, genetic association mapping, dynamic network analysis, gene expression analysis, human genetics, expression quantitative locus mapping, genome-wide association study, structured association mapping, visual analytics

This dissertation is dedicated to my beautiful wife, Anna,
and to my three children,
whose support and patience have carried me through.

Abstract

The last ten years since the completion of the human genomic sequencing project have seen huge advances in the understanding of the genetic basis of human disease. Understanding the genes involved in disease and the causal genomic polymorphisms involved holds the promise of better treatment and prevention of disease. Much of the recent progress has been made through the use of the popular genome-wide association study (GWAS). However, despite the success of GWAS, its findings often fail to explain the full heritability of a disease, or the findings include SNPs that affect a disease through some unknown biological mechanism.

The incorporation of gene expression or clinical trait data into GWAS is one approach that can further elucidate the mechanisms behind SNP-disease associations. These so-called intermediate phenotypes have inherent structures, such as correlations and interactions, which can be leveraged to facilitate discovery. The promise of these data has motivated a new generation of GWAS algorithms, termed *structured association mapping*, which use cutting-edge machine learning techniques to fully leverage structures in the data to uncover associations between the genome, transcriptome, and phenome.

However, the increasing amounts of data used in GWAS, and the complexity of the methods used to analyze the data, demand a new integrative approach to genetics discovery. To fully capture the potential available in today's genetic data, we must rely on the strengths of machines and people. With this in mind, I have developed a visual analytics software system called GenAMap. GenAMap has been built to automate the execution of structured association mapping algorithms, making them available to genetics analysts. Through GenAMap, I introduce new visualizations that are built to enable analysts to explore the structure of genetic data while considering genomic associations. Through the integration of the strengths in the machine learning, visualization, and genetics fields, I show that GenAMap has the potential to facilitate and advance the progress of genetics discovery through the analysis of human asthma, yeast, and mouse datasets.

In this work I also demonstrate the integration of visualization and machine learning to another domain in genetics research: the study of dynamic genetic networks. I present TVNViewer, an online visualization tool for exploring these networks, and use a yeast and breast cancer dataset to show how the visualizations in TVNViewer enable the analysis and exploration of the networks as they change across time and space.

In the genetics world where the amount of available data continues to grow, the integration of visualization and machine learning techniques has the potential to accelerate advancement in genetics discovery.

Acknowledgments

I am indebted to many who have helped me along the path to this dissertation. I would like to thank my advisor, Dr. Eric Xing, whose enthusiasm for my work has carried it forward from a few ideas to what it is today. Eric has been a fantastic support and has allowed me to exploit my own strengths in research and to explore new fields and ideas. I have appreciated Eric's knowledge of the field and his own excitement for research. I would also like to thank Dr. Seyoung Kim, who introduced me to the research world and started me on the path towards my research goals. I would like to thank all of those that I have collaborated with along the way: John Woolford, Peter Kinnaird, Sally Wenzel, Jing Xiang, Ankur Parikh, Kriti Puniyani, Seunghak Lee, Anuj Goyal, and Junming Yin. Each of these collaborators comes from a different field with different areas of expertise. Through each collaboration, I have been introduced to and learned exciting new perspectives about research. My education would not be complete without the perspective I have learned from the machine learning, computational biology, molecular biology, human genetics, and information visualization fields.

The support of my family has carried me through all of the hard times on the long road towards the dissertation. Thanks go to my parents who have been positive and supportive while I've kept their grandchildren so far away! I am thankful to my three children, who have been patient while Daddy was at school and have made every hour at home full of excitement. Finally, I would like to thank my wife Anna for her constant love, support, and patience.

I have had great friends who have listened to my ideas and helped me to think through different problems. In particular, I would like to thank Josh Kangas, who endured the rigors of the PhD program with me and who always had a great perspective on the challenges we faced.

I would like to thank all of the students who have worked with me as part-time software coders. It was a wonderful experience to learn how to manage software projects, and fun to work with students who were excited to jump in and make a difference in a short period of time.

Finally, I would like to recognize God's hand in my life and career. The guidance I have received from answered prayers has led me here today.

Contents

1. Background 1

1.1 Genome-Wide Association Studies 2

1.1.1 Structured Association Mapping 6

1.1.2 Structured association mapping leveraging genome structure 7

1.1.3 Structured association analysis using transcriptome and phenome structure 8

1.1.4 Structured association analysis for joint three-way association studies 9

1.2 Visual Analytics and Information Visualization 10

1.3 Current visualization strategies for association mapping 12

1.3.1 Introduction 13

1.3.2 Traditional association mapping software 14

1.3.3 GWAS visualization software 15

1.3.4 Advances in eQTL study visualization 23

1.3.5 Visualization examples from current biology studies 29

1.3.6 Conclusions 37

1.4 The promise of visualization in dynamic gene network analysis 38

1.5 Current visualization strategies in genetic network analysis 42

2. Structured Association Mapping 47

2.1 Problem formulation 48

2.2 GFlasso 49

2.3 Investigating structured association mapping through simulation 51

2.3.1 Simulating the traits 52

2.3.2 Simulating the SNP matrix 54

2.3.3 Methods 55

2.3.4 Simulation Results 58

2.3.5 Conclusions from the simulation study 60

2.4 gGFlasso, a new algorithm for three-way association analysis 61

2.4.1 gGFlasso implementaiton notes 63

2.5 Validation of gGFlasso through simulation 64

3. GenAMap 67

3.1 System Overview 70

3.1.1 Overview of the software 72

3.2 Auto-SAM 74
3.2.1 Data management 75
3.2.2 Algorithm automation and parallelization 76
3.2.3 Running time analysis 79
3.2.4 Auto-SAM case study: Integrating GFlasso into Auto-SAM 80
3.3 Visualization 83
3.3.1 Exploring gene networks and eQTLs using yeast data 83
3.3.2 Exploring the gene network 84
3.3.3 Finding eQTLs using GenAMap 87
3.3.4 Exploring association results through the association tree view 93
3.3.5 A case study on a mouse GWA dataset 94
3.3.6 The population structure view 95
3.3.7 The population association view 96
3.3.8 Conclusions 96
3.4 Three-way Visualization 97
3.4.1 Overview first: an introduction to the visualization 98
3.4.2 Zoom and filter: identifying interesting signals in the data 100
3.4.3 Details on demand: resources for further exploration 101
4. Using GenAMap in Association Analysis 103
4.1 GFlasso analysis reveals interacting hotspots in yeast data 103
4.1.1 The discovery of interacting eQTL hotspots 106
4.1.2 Discussion of yeast results 127
4.1.3 Methods used in yeast study 130
4.2 GenAMap analysis on SARP and CGSA asthma datauncovers new asthma gene 134
4.2.1 Assessing population structure 135
4.2.2 Exploring association by population 136
4.2.3 Using structured association mapping to analyze the asthma data 138
4.3 Structured association analysis of NIH heterogenous stock mice dataset 140
4.3.1 Genetic network analysis 141
4.3.2 eQTL analysis using GFlasso 145
4.3.1 Joint three-way analysis using GFlasso-gGFlasso 149
5. GenAMap Evaluation 155

5.1 User study results 156
5.1.1 Survey results 157
5.1.2 Think-aloud comments and results 158
5.2 The popularity of GenAMap 159
6. TVNViewer 161
6.1 TVNViewer Overview 162
6.2 Implementation and Design 163
6.3 Case Studies: Using TVNViewer for Dynamic Network Analysis 166
6.3.1 One-level network circle view 167
6.3.2 Two-level network with GO annotations 168
6.3.3 Force View 171
6.3.4 Directed Graphs 172
6.3.5 Stack view 173
6.4.6 Timeline View 176
6.4.7 Degree Distribution 177
6.4 Biological Analysis Using TVNViewer 178
6.4.1 Analysis of temporally dependent gene-gene interactions across the yeast cell cycle 179
6.4.2 Exploring the progression and reversal of breast cancer 183
6.4.3 Treegl results on breast cancer data set 186
6.4.4 Analysis of hubs in T4 network 187
6.5 Use and popularity of TVNViewer 189
7. Conclusions 191
7.1 Lessons learned from the user study 194
7.2 Lessons learned from the tool development 195
7.2.1 Problems Solved 196
7.2.2 Development strategies 197
7.3 Future Work 199
7.3 Final Word 201
Contributing Work 202
Bibliography 207

1. Background

My intention in this background chapter is to give the reader the historical context that has motivated my dissertation work. My work has focused primarily in two areas of genetics research – genetic association mapping and dynamic gene-network analysis. As I will show in this dissertation, the development of software visualization tools that combine the strengths of machines and people has contributed significantly to genetics analysis in each of the two areas.

The background chapter is structured as follows. In Section 1.1, I introduce genetic association mapping and discuss the development and potential of *structured association mapping*. In Section 1.2, I discuss the potential that visualization has in association mapping. In Section 1.3, I review the work that has been done for visualization in association mapping studies. It is through Sections 1.2 and 1.3 that I preset the historical context motivating the development of GenAMap, a visual analytics tool for structured association mapping. GenAMap is presented in Chapters 3, 4, and 5 of the dissertation.

The background chapter concludes with two sections that motivate the development of TVNViewer, an online software tool for exploring networks that change over time or space. In Section 1.4, I discuss the promise of visualization in dynamic genetic network analysis, and in

Section 1.5 I provide a short review of network visualization tools that are currently available. TVNViewer is presented in Chapter 6.

1.1 Genome-wide association studies

As genome sequencing technology has improved over the last decade, it has led to an increased understanding of how mutations in the genome lead to complex phenotypes and heritable human diseases. Understanding the specific interactions between DNA, genes, and traits holds the promise of innovative treatments for many diseases. A popular strategy to discover how sequence variation affects the inheritance of complex traits is the genome-wide association study (GWAS) (Manolio et al. 2009; Hindorff et al. 2011). GWAS has led to the successful identification of many so-called *disease genes* and *susceptibility loci* for a variety of diseases such as cancer (Yeager et al. 2009), diabetes (Yaguchi et al. 2005), Alzheimer's disease (Waring and Rosenberg 2008), Crohn's disease (Lettre and Rioux 2008), asthma (Postma and Koppelman 2009), and heart disease (McPherson et al. 2007). In a traditional GWAS, individuals are sequenced for genetic polymorphisms across the genome. The individuals are divided into case and control groups, and machine learning or statistical techniques identify mutations that are associated with the disease of interest.

The motivation for GWAS comes from the central dogma of biology, which is that certain parts of the DNA (genes) code for mRNA, which is translated into the proteins that run the cell and the organism. Thus, mutations in the genome at the DNA level can directly affect the entire organism by altering the creation or function of proteins in the cell. Although most of the human genetic sequence is identical across individuals, there are places in the genome where the sequence has been mutated, causing a genetic polymorphism between individuals. If we think of DNA as a string made up of four nucleotides (characters), then a genetic polymorphism is a difference in the sequence between two individuals. The most common type of genetic polymorphism is a single-nucleotide polymorphism (SNP), an instance where one nucleotide is different between individuals. For example, some individuals may inherit a G at a particular location instead of the A that is common in the population. Although many SNPs make little or no difference to gene expression levels and the normal function of a cell, some SNPs can have a much larger effect. SNPs that turn off important genes, or change the coding sequence of regulator genes, can interact with other expressed genes and lead to a diseased phenotype or greater susceptibility to disease.

The goal of GWAS, then, is to identify genetic polymorphisms associated with disease, which ideally lead to greater insight about disease prevention, acquisition, and progression. However, the success of many GWAS in explaining SNP-phenotype associations and leading to clinical treatments has been limited (Schadt 2009), in part, because traditional GWAS only considers the association from the SNPs to the phenotype and ignores the underlying biological system. Thus for many studied diseases, discovered SNPs only explain a fraction of the disease heritability (Manolio et al. 2009) or identify SNPs that do not affect protein sequence and thus have an unknown regulatory role in the cell (Schadt et al. 2008). Hence, we are far from a complete understanding of how discovered SNPs actually regulate cellular pathways leading to disease (McCarthy and Hirschorn 2008). Unfortunately, without adequate understanding of how SNPs operate in a biological context, it is often impossible for a study’s results to lead to clinical applications (Schadt 2009). Overcoming these barriers is a significant problem the human genetics community faces today.

Recent developments in genetics use a variety of strategies to improve discovery in association studies. One type of study that has the potential to help uncover the biological mechanisms is the expression quantitative locus (eQTL) study. eQTL analysis has been widely used to understand how genetic variations in the genome perturb biological systems by altering cellular mRNA levels (Gilad et al. 2008). A typical eQTL study involves genotype data collected for genetic markers, such as SNPs, along with microarray data for a population of individuals. These studies aim to identify genes whose expression level varies according to genetic variation. As both the genotype and gene expression data are collected at a genome-wide scale, eQTL analysis gives opportunities to probe the complex interplay between the genome and phenome at a systems level rather than at the level of individual loci and genes. For example, genetic variation can perturb the expression of a gene, which then can affect the activity of other genes downstream in the same pathway. A mutation in a regulator, such as a transcription factor, can influence the expression of all of the regulator's target genes, leading to a pleiotropic effect. On the other hand, genetic variants at multiple different loci may jointly influence the expression of some genes. Analyzing a genome-wide eQTL dataset allows us to discover the complex patterns of how genetic variants give rise to variation in expression level. At the same time, examining multiple genes or multiple traits jointly in a genome-wide analysis can give insight into the functional roles that genetic variants play in a biological system and can potentially lead to the discovery of new regulators in a region of associated SNPs. Many eQTL datasets have been collected for various organisms, including yeast (Brem et al. 2002), mouse (Chen et al. 2008), human (Stranger et al. 2007), and Arabidopsis (West et al. 2007), as well as for different diseases such as diabetes (Yaguchi et al. 2005).

The limitations of GWAS may be overcome, in part, through the combination of additional data types (such as eQTL data) to the traditional GWAS data (SNP genotyping and trait measurements) (Schadt 2009; Califano et al. 2011). For example, current research showing that SNPs associated with complex traits are likely to be eQTLs (Nicolae et al. 2010) has motivated studies to use eQTL information to select SNPs in GWAS, or to perform association studies using gene expression data in place of SNP data. Other strategies have added data into GWAS analysis from other "-omes," creating a meta-dimensional analysis with multiple data types. By incorporating established knowledge about the biological system, discovery in association studies is enhanced. Gene expression data are now commonly used to integrate transcriptome information into association studies (Schadt et al. 2008; McCarthy and Hirschorn 2008; Gilad et al. 2008). Successful integration of eQTL analysis into GWAS has led to the identification of new disease genes in humans and mice (Cookson et al. 2009; Hsu et al. 2010; Silveira et al. 2010; Chen et al. 2008).

The subject of this dissertation concerns the growing complexity of GWAS when gene expression or other data are added. While the additional data can be overwhelming, it can also be leveraged *algorithmically* and *visually* to improve and enhance discovery in GWAS. In this dissertation, I propose how this can be done, demonstrating this new paradigm of thinking about GWAS and eQTL studies through a visual analytics software platform called GenAMap. However, in order to understand GenAMap, the reader must be familiar with a new generation of GWAS machine learning technology, termed *structured association mapping*.

1.1.1 Structured association mapping: a new generation of GWAS machine learning technology

Algorithmically, traditional, simple methods that look for pairwise associations between SNPs and genes or between SNPs and phenotypic traits do not take advantage of all the information available in the data. Structure inherent in the genome, transcriptome, and phenome provide information about the underlying biological system and thus can guide analytic methods to discover associations that are hidden to simpler methods. The recent development of a new generation of GWAS algorithms, termed *structured association mapping* algorithms, utilizes structural and other information inherent to GWAS and eQTL data to discover genome-transcriptome-phenome associations and to eliminate false positives.

In this subsection I briefly review structured association mapping, an emerging algorithmic paradigm for GWAS. Structured association mapping is advantageous over simple, pairwise methods because it takes advantage of structure in the genome, transcriptome, and phenome in the discovery of association signals. I consider each of these “-omes” in turn to describe the available structural information that is available and how structured association mapping leverages that structure in a systematic framework.

Structured association mapping is built using sparse-regression techniques, built off of the lasso (Tibshirani 1996). The lasso is advantageous in association mapping as it selects the most informative predictors (SNPs) for each response (genes or traits) and eliminates false positives. As we incorporate further information from the data into this statistical paradigm, we not only select the most informative SNPs, but we also leverage the structure through the addition of further optimization penalties to enhance discovery.

I can define the problem of association mapping as follows. Let $\boldsymbol{X}$ be an $N \times P$ genotype matrix for N individuals and P SNPs and let $\boldsymbol{Y}$ be an $N \times J$ gene expression matrix where expression levels of J genes are measured for the same individuals. Finally, let $\boldsymbol{Z}$ be an $N \times K$ phenotype matrix where each row records K phenotypic traits of an individual. Then, using lasso regression to find associations between $\boldsymbol{X}$ and $\boldsymbol{Z}$, I would optimize the following equation:

$$\boldsymbol{B_3} = \underset{\boldsymbol{B} \in \mathbb{R}^{P \times J}}{argmin} \|\boldsymbol{Z} - \boldsymbol{XB}\|_F^2 + \lambda \, |\boldsymbol{B}| \tag{1.1}$$

$\|.\|_F$ is the Frobenius norm of the matrix. The first term in the equation penalizes based on prediction error, and the second term is the L_1 lasso penalty, which has the property of shrinking the strengths of irrelevant SNPs towards zero. In this scenario, $\boldsymbol{B_3}$ is a $P \times K$ matrix representing associations between SNPs and phenotypes. Structured association mapping differs from classical association mapping approaches in that it uses sparse-regression results instead of *p*-values. Thus, a nonzero value in the association matrix is considered to be a significant association, and tests to account for multiple testing issues are not used.

1.1.2 Structured association mapping levering genome structure

There is structure and information from the genome that can provide insight in association analysis. For example, consider population structure. While many SNPs may be population-specific, some SNPs may have similar effects across populations. The multi-population group lasso (MPGL) is a sparse-regression method that allows associations to be discovered in different populations independently, while incorporating information across all populations (Puniyani et al. 2010). This is done by building on the multi-task learning research in machine learning. More specifically, MPGL does this through the introduction of a L_1/L_2 penalty instead of the lasso penalty, as shown in Eq. 1.2:

$$\boldsymbol{B_3} = \underset{\boldsymbol{B} \in \mathbb{R}^{P \times J}}{argmin} \|\boldsymbol{Z} - \boldsymbol{XB}\|_F^2 + \lambda \|\boldsymbol{B}\|_{L_1/L_2} \text{ where } \|\boldsymbol{B}\|_{L_1/L_2} = \sum_{j=1}^{p} \left\|B_j\right\|_F^2 \tag{1.2}$$

In this case, $\boldsymbol{B_3}$ is calculated trait-by-trait. If we assume C populations, then $\boldsymbol{B_3}$ is a $P \times C$ matrix, with one column for each population. Thus, rows represent SNPs and columns represent associations from SNPs to populations for one trait.

Alternatively, we can consider known features about the SNPs in our dataset. For example, certain SNPs are conserved across species, in genetic promoter regions, or in non-coding DNA. Using this information, we can use the Adaptive Multi-task lasso (AMTL) to find genome-transcriptome or genome-phenome associations (Lee et al. 2010). AMTL uses multi-task learning to optimize a lasso-type equation as well as select weights on the SNP features. The result is a $P \times K$ $\boldsymbol{B_3}$ matrix representing genome-phenome associations.

1.1.3 Structured association analysis using transcriptome and phenome structure

Now I consider leveraging structural information in the transcriptome in association analysis to detect eQTLs. The methods that I consider here can also be used for genome-phenome association analysis, when one is looking for associations to multiple-correlated traits. I review two structured association mapping algorithms that leverage information in the traits or genes. These methods assume that related traits or genes tend to be influenced by a common, small subset of SNPs. Biologically, this might be the case where a genetic regulator affects the expression levels of multiple genes in a common pathway.

The first method, Graph-guided fused lasso (GFlasso) (Kim and Xing 2009), extends the lasso such that a network structure is used to guide the discovery of associations. This network can be constructed using simple techniques based on correlation, or it can reflect known gene-gene or protein-protein interactions. I define $G_G = (V_G, E_G)$ as a relevance graph where each node represents a gene in $\boldsymbol{Y}$ and each edge represents a weighted relationship between two nodes in the network graph. GFlasso can be described by the following optimization problem:

$$\boldsymbol{B_1} = \underset{\boldsymbol{B} \in \mathbb{R}^{P \times J}}{argmin} \|\boldsymbol{Y} - \boldsymbol{XB}\|_F^2 + \lambda \sum_j \sum_p |B_{pj}| + \gamma \sum_{\{u,v\} \in E_G} \sum_p \left|B_{pu} - sign(\rho_{uv}) B_{pv}\right| \tag{1.3}$$

In Eq. 1.3, $\boldsymbol{B_1}$ is a $P \times J$ matrix representing genome-transcriptome associations. Similarly, I can create a network graph $G_T = (V_T, E_T)$ for the traits and substitute $\boldsymbol{Z}$ for $\boldsymbol{Y}$ and G_T for G_G to find a $P \times K$ matrix representing genome-phenome associations. GFlasso thus leverages the network structure of the genes or traits to find associations to SNPs that affect networks or pathways of genes. GFlasso is discussed further in Section 2.2.

A similar approach to GFlasso is TreeLasso (Kim and Xing 2010). TreeLasso builds a hierarchical clustering tree from the genetic network and uses the tree as information about the relationships between genes or traits to guide the association discovery.

1.1.4 Structured association analysis for joint three-way association studies

Finally, I consider a structured association mapping approach that uses combined genome, transcriptome, and phenome data to perform a joint-three way association analysis, GFlasso-gGFlasso (Curtis et al. 2012). This is done through a two-stage process. First, we find genome-transcriptome associations using GFlasso as just described. Next, we find transcriptome-phenome associations using the graph-Graph-guided fused lasso (gGFlasso):

$$\boldsymbol{B_2} = \underset{\boldsymbol{B} \in \mathbb{R}^{J \times K}}{argmin} \|\boldsymbol{Z} - \boldsymbol{YB}\|_F^2 + \lambda \sum_j \sum_k |B_{jk}| + \gamma_1 \sum_{\{u,v\} \in E_G} \sum_k |B_{uk} - sign(\rho_{uv}) B_{vk}| + \gamma_2 \sum_{\{m,l\} \in E_T} \sum_j |B_{jm} - sign(\rho_{ml}) B_{jl}|. \tag{1.4}$$

In gGFlasso, a second fusion penalty is added to the GFlasso framework to encourage genes related in the gene relationship network to influence the same related traits in the trait network. This model assumes that the effects of genes in the same pathways might be similar on multiple-related traits. gGFlasso is discussed further in Section 2.4.

Each of the five structured association mapping algorithms that I have reviewed here (MPGL, AMTL, GFlasso, TreeLasso, and gGFlasso) is available online and is automated in GenAMap.

1.2 Visual Analytics and Information Visualization are the next steps in structured association mapping analysis

In association mapping analysis, millions of SNPs, tens of thousands of genes, and hundreds of phenotypic traits are analyzed. I have already shown the potential of new machine learning technology in these analyses. Although machine learning has carried association mapping well thus far, the next steps strongly indicate that visualization, combined with machine learning, will advance the field still further by taking advantage of the analytic capabilities of *both* machines and people. This type of system is termed *visual analytics* software. Visual analytics is a rapidly emerging field that describes the combination of advanced information visualization techniques with statistics and machine learning. Visual analytics tools combine the power of automated information extraction with the intuition and cognitive strengths of human decision making (Keim et al. 2008). Visual analytics are best employed when analysts need to explore their data, understand the overall structure of the data, discover weak patterns most easily recognized by humans, and retain the ability to perform detailed analysis (Card et al. 1999), which are all characteristics of structured association analysis. However, in order to create this system, new visualization strategies must be designed and built. The vast amount of input and output to structured association mapping algorithms, and the sparseness of useful output, classically suggests that a visualization strategy will aid analysts in the exploration of this data to identify the links between DNA, genes, and traits to eventually produce new treatments for disease.

Indeed, analyzing structured association mapping results is a near perfect fit for visualization for many reasons. For example, once an analyst has run structured association mapping algorithms, the focus of the investigation becomes more exploratory than query driven (Fekete et al. 2008). Information visualization, "the use of computer-supported, interactive visual representations of data to amplify cognition," as a field, touts its strengths as generating exploration-based insights, explanatory and persuasive interaction, and aesthetic representations (Card et al. 1998). Visualization techniques, therefore, excel when providing an explanation of the overall structure of the data or enabling analysts to find weak or unexpected patterns most easily recognized by humans (Card et al. 1999), critical requirements for structured association analysis.

Indeed, the success of visualization has emerged already in many areas of biology. For example, Cytoscape (Shannon et al. 2003) has become an extremely popular application for visualizing biological networks and exploring relationships between genes. In other domains, the recent development of ABySS-Explorer (Birol et al. 2011; Nielsen et al. 2009) has shown that visualization can enhance the analysis of complex biological tasks like genome assembly through a visual representation of the contigs. Another recent approach to visualization in biology, MulteeSum, demonstrated the potential for visualization to aid in the identification of spatial and temporal patterns in gene expression data (Meyer et al. 2010). For simple GWAS with one trait, excellent visualization tools have been built to explore linkage disequilibrium (LD), strength of association, and surrounding genes in the association results (Pruim et al. 2010; Ge et al. 2008).

The potential of visualization and visual analytics in biology has motivated the development of GenAMap, visual analytics software for structured association mapping. In Chapters 3, 4, and 5, I will present GenAMap and the new visualization strategies that I have developed to enable genetics analysts to better explore and analyze their data in structured association mapping studies.

1.3 Current visualization strategies for association mapping

I have now established the popularity and potential of genetic association mapping studies and discussed current advances in machine learning that advance these studies further. I have also suggested that the next steps in the field include information visualization and visual analytics. In this section I discuss the visualizations currently available for association mapping. For the purposes of this section, I divide association mapping studies into two camps – GWAS and eQTL studies. By GWAS, I mean studies that look for genome-wide associations to one phenotype, usually a disease state; by eQTL studies, I refer to studies that look for genome-wide associations to gene expression data.

In either scenario, GWAS or eQTL studies, the genetics analyst must interpret and explore large amounts of data. The results from these types of studies can be complex, and the results of the study need to be presented succinctly to other geneticists. Unfortunately, despite the advances in machine learning technology to enhance and enable genetics discovery in association mapping analysis, the software tools and visualization strategies currently employed are insufficient to support the data and machine learning algorithms available today. In this section I review the tools and visualization strategies currently available for GWAS and eQTL studies. While there are many excellent tools available for the visualization of GWAS results, the development of tools for the exploration of structured association mapping results is especially needed.

This section thus presents the foundation that motivates the development of new visualizations in GenAMap, which visualizations fill in this gap and provide the tools that enable GWAS and eQTL studies to continue to move forward in the structured association mapping paradigm.

1.3.1 Introduction

GWAS and eQTL studies require expertise at many levels, from implementing the biological experiments to the statistics and machine learning strategies that analyze the data. The complexity of the data, and the differences between the genetics and machine learning fields, often makes the interpretation of the results quite difficult. One way to overcome both of these challenges is through the incorporation of visualization strategies into the association analysis. For example, visualization techniques help to overcome the complexity of the data by allowing researchers to emphasize important dimensions or to make key elements stand out (Fekete et al. 2008). Additionally, visualization tools have the potential to bridge the vastly different fields of biology and statistics by presenting the most powerful statistical results in a biologically intuitive manner (Moore et al. 2010). Despite the potential of visualization, in a recent review it was noted that “designing visualization approaches for GWAS data has lagged far behind other areas of GWAS software development” (Buckingham 2008).

As I will discuss in the following subsections, there are excellent visualization tools available for visualizing GWAS results. However, as the complexity of the methods and data used in an analysis increases (as is the case in structured association mapping analysis), the problem of adequately exploring association mapping data increases. To truly take advantage of the power of structured association mapping, adequate visualization tools must be available. However, methods to visualize the structure of the data while exploring the associations between the genome and the phenome are still in primitive form.

I begin this review by discussing the traditional methods used for the visualization of association mapping results. I then review the software platforms that are currently available to explore GWAS and eQTL data. Finally, I sample current literature to demonstrate the common strategies currently used to present the results from GWAS and eQTL analysis. While visualization has tremendous potential, we are only scratching the surface in creating the software that will bridge the gap between the machine learning algorithms and the genetic analyses that can take place.

1.3.2 Traditional association mapping software

To date, most of GWAS and eQTL analysis is done through command line scripts and command line software programs. The reliance on command line software limits the number of analysts that can perform these kinds of studies, creating an "exclusive club" (Buckingham 2008). Much of the work done in GWAS is performed using the popular command line program called PLINK (Purcell et al. 2007). PLINK is command line software that implements multiple association mapping algorithms from haplotype to family-based analyses. Although PLINK is a powerful toolkit and has many utilities for data management, association tests, and data analysis (such as Hardy-Weinberg testing), all output is in tabular format. Therefore, it requires a certain level of expertise to visualize and interpret the results. The workflow is additionally complicated due to the necessity of using multiple software packages to visualize the data once the analysis is complete.

One common technique for exploring results from command line programs like PLINK is to use a general-purpose tool such as CRAN-R (R Develoment Core Team 2010). Although R is not specifically built for visualization, its use is wide-spread. The use of R for exploring association results requires a significant amount of training and expertise; there are many online resources and courses to learn how to perform such an analysis. One example of the process scientists use to combine analyze PLINK data in R and PLINK is available from the *Getting Genetics Done* website (Turner and Bush 2009).

Many association mapping algorithms are distributed as command line tools (Stegle et al. 2010; Kim and Xing 2009). These distributions require expertise to format the data and perform the calculations using the software, and then the results must be interpreted using another tool. Some software packages even create output designed to follow a specific processing pipeline in R (Jayawardena et al. 2010).

As the amount of data and complexity of analysis increases, it will become an increasingly difficult task for the genetics analyst to parse through their association results and come to meaningful conclusions (Buckingham 2008).

1.3.3 GWAS visualization software

In this subsection I review the tools that are currently available to visually explore GWAS results. Interestingly, most of these tools read in a GWAS analysis results file, usually from PLINK, and then provide a way for analysts to explore the results. As I show, the strategies employed in these tools are creative and allow geneticists to analyze results through the incorporation of data from multiple sources. I suggest that the latest GWAS visualization tools, for exploring the results from an analysis with *one* phenotype, are quite good.

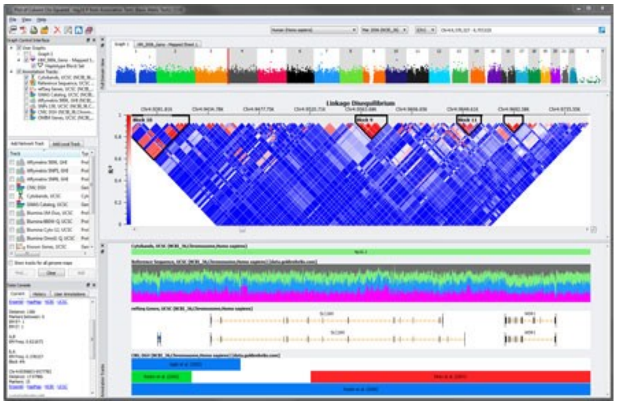
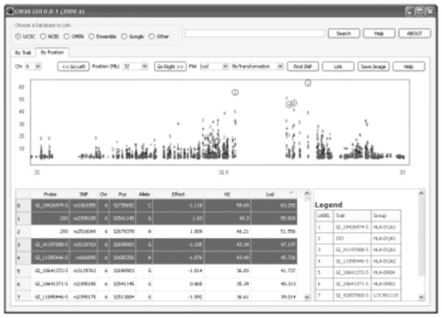


Figure 1.1: An example of the output from Golden Helix's visualization of GWAS data. The image was retrieved from the Golden Helix website, used with permission from SNP & Variation Suite, Golden Helix, Inc.

In GWAS, results are often visualized as a Manhattan Plot of the *p*-values on the –log10 scale. These plots are pervasive throughout genetics journals and conferences, despite the limited information that they provide. Fortunately, with increasing frequency, these plots also include other data such as LD and gene maps across the chromosome. A popular software interface for these types of visualization is produced by the company Golden Helix (2011). Although Golden Helix offers a wide variety of quality control and other software, its tools for association analysis are quite simple. A colored-by-chromosome Manhattan Plot produced by Golden Helix software is shown in Figure 1.1. This plot also includes analysis tools that visualize LD across the genome. Manhattan Plots, like the one shown in Figure 1.1, are a popular visualization to show GWAS results.

I will now review nine other tools available for the visualization of GWAS results. I will first start out with simple tools that demonstrate some of the important features to include in a good GWAS visualization tool, and I will conclude with what I deem the most complete two visualization tools available today – LocusZoom (Pruim et al. 2010) and WGAViewer (Ge et al. 2008).

An important piece of almost all GWAS results-viewing tools is the integration of data from outside sources. One clever tool that integrates data from multiple sources is Path (Zamar et al. 2009). Path's visualization scheme is simple, using a simplified Manhattan Plot to visualize association results. However, Path provides links to visualize pathways, is able to search using different data types, and integrates data from over nine data sources.



	Probe	SNP	Chr	Pos	Allele	Effect	H2	Lod
0	GI_10426974-S	rs1063355	6	32739692	C	-1.118	59.69	63.292
1	293	rs2395185	6	32541145	G	1.03	49.3	55.509
2	293	rs2516049	6	32678378	A	1.009	46.21	51.556
3	GI_41197088-S	rs3129763	6	32698903	G	-1.105	43.34	47.197
4	GI_11095446-S	rs660895	6	32685358	A	-1.076	43.49	45.726
5	GI_10641372-S	rs3129763	6	32698903	G	-1.014	36.83	41.737
6	GI_10641371-S	rs2395185	6	32541145	G	0.865	35.39	40.313
7	GI_11095446-S	rs2395175	6	32513004	G	-1.092	36.61	39.014

LABEL	Trait	Group
1	GI_10426974-S	HLA-DQA1
2	293	HLA-DQA1
3	GI_41197088-S	HLA-DQA1
4	GI_11095446-S	HLA-DQA2
5	GI_10641372-S	HLA-DRB4
6	GI_10641371-S	HLA-DRB3
7	GI_42657068-S	LOC401118

Figure 1.2: The figure from the GWAS GUI paper shows the Manhattan plot and list interface. Used with permission.

GWAS GUI is a front-end interface that allows analysts to explore GWAS results using the most popular visualization strategies – tables and Manhattan plots (Chen et al. 2009). GWAS GUI provides a list of SNPs that can be sorted by the strength of association to the phenotype. While GWAS GUI can store many phenotypes, the analyst cannot explore the associations to different phenotypes simultaneously.

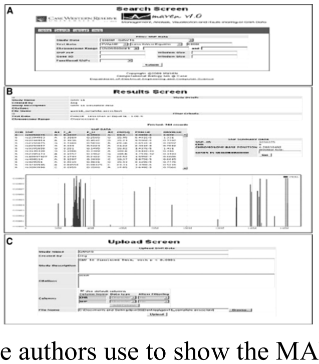
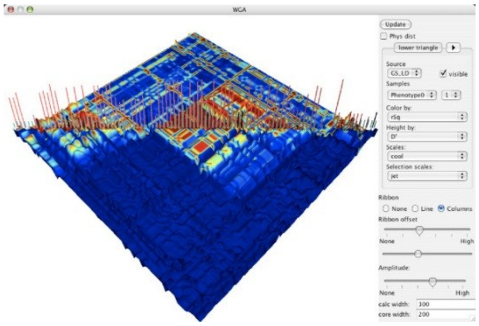


Figure 1.3: The figure the authors use to show the MAVEN interface. Used with permission.

MAVEN is built for the storage and exploration of GWAS data, similar to GWAS GUI in its approach (Narayanan and Li 2010). MAVEN allows analysts to view results using a graphical format of *p*-values as a line chart or as a tabular view that can be sorted and explored. These synchronized views are shown in Figure 1.3. While similar to GWAS GUI in its visualization strategy, MAVEN differs by allowing analysts to query for information about SNPs directly from the online application, which accesses its own copy dbSNP (Sherry et al. 2001). Through the development of interactive tables and lists, GWAS GUI and MAVEN are among the first tools for GWAS visualization.



□ Phys dist

Source: GS_LD ☑ visible

Samples: Phenotype0 1

Color by: rSq

Height by: D'

Scales: cool

Selection scales: jet

Ribbon: ○ None ○ Line ● Columns

calc width: 300

core width: 200

Figure 1.4: This figure from the Goldsurfer2 paper shows the 3D LD visualization technique. Used with permission.

Goldsurfer2 is another software pioneer in GWAS visualization (Pettersson et al. 2008). Goldsurfer2 is similar to GoldenHelix software in its approach; Goldsurfer2 is largely focused on analyzing the SNPs in a GWAS for quality control and sampling accuracy. One unique visualization that was developed for GoldSurfer2 is the 3D exploratory visualization to analyze the LD around discovered SNPs (Figure 1.4). Goldsurfer2 also integrates tables and the classic genome explorer-type view to intuitively explore the structure around discovered SNPs.

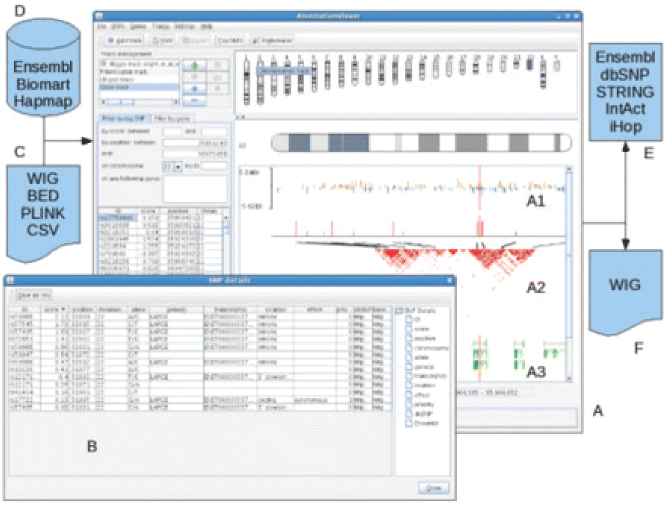


Figure 1.5: The figure from the AssociationView paper demonstrating their interface of tables and charts to explore GWAS results. Used with permission.

AssociationViewer is similar to the GWAS visualization tools discussed thus far, although it provides additional links to external data sources (Martin et al. 2009). Given the results from an association study, AssociationViewer provides the analyst with the typical options to explore LD in the genomic regions, to browse for SNPs through the chromosomes, and to see the top hits list in tabular format. AssociationViewer took a step forward by integrating much more tightly with the internet. In Figure 1.5, I show a figure from the original paper highlighting the tables listing the top hits, the Manhattan plot across the chromosome, and the LD plots that the tool provides. In this figure, the authors make careful note of the many external data sources that the tool utilizes.

Synthesis-View is a tool with a similar visualization strategy, but it is built for the visualization of data with multiple population components (Pendergrass et al. 2010). Given results from association tests, the tool uses color to represent different cohorts or populations and then plots the values using the strategy shown in Figure 1.6. This natural view of the data is easy to interpret and compare results for datasets and handful of SNPs, or by visualizing only the top ranking SNPs in a study.

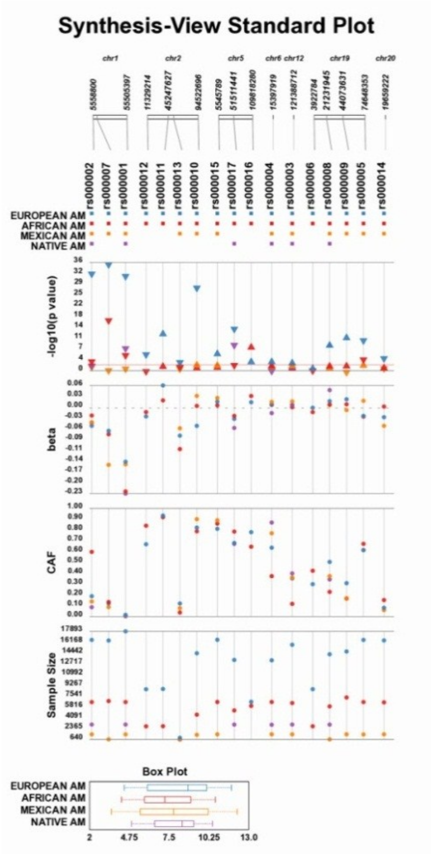


Figure 1.6: This figure from the Synthesis-View paper shows multi-population visualization with color. Used with permission.

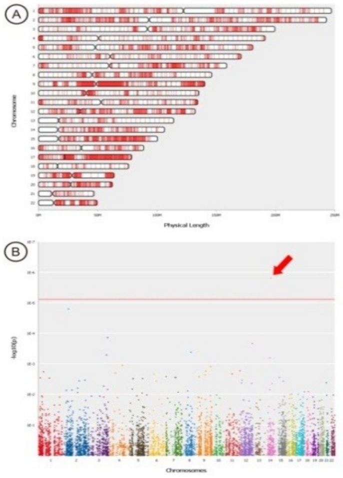


Figure 1.7: A figure taken from the CONAN paper. Used with permission.

CONAN is built for the analysis of GWAS studies using copy-number variations (Forer et al. 2010). Although built with a slightly different focus, it is largely similar to other GWAS tools – providing the Manhattan Plot and density plot (of CNVs) across chromosomes (Figure

1.7). Additionally, links out to information on each CNV region are provided, including links to previous GWAS the UCSC genome browser (Rosenbloom et al. 2009).

To date, the two most complete tools for GWAS analysis are WGAViewer and LocusZoom. WGAViewer is a tool built to allow analysts to explore data resulting from GWAS analyses (Ge et al. 2008). Analysts can switch between many views, which allow them to see Manhattan plots of *p*-values across the genome, lists of top hits, LD structure, gene maps, and recent selection evidence. WGAViewer provides links out to database information online, and also brings information into the tool as well, such as allowing the analyst to consider SNPs that have not been genotyped. While most of the previous tools such as GWAS GUI, MAVEN, or AssociationViewer all contribute to GWAS visualization in a unique way, WGAViewer incorporates all the contributions in a complete visualization tool (Figure 1.8).

LocusZoom is an innovative tool developed to enhance the visualization and exploration of association results from a GWAS study (Pruim et al. 2010). LocusZoom features the coloring of nodes by LD, annotates SNPs with shapes based on potential effect on protein sequence, and

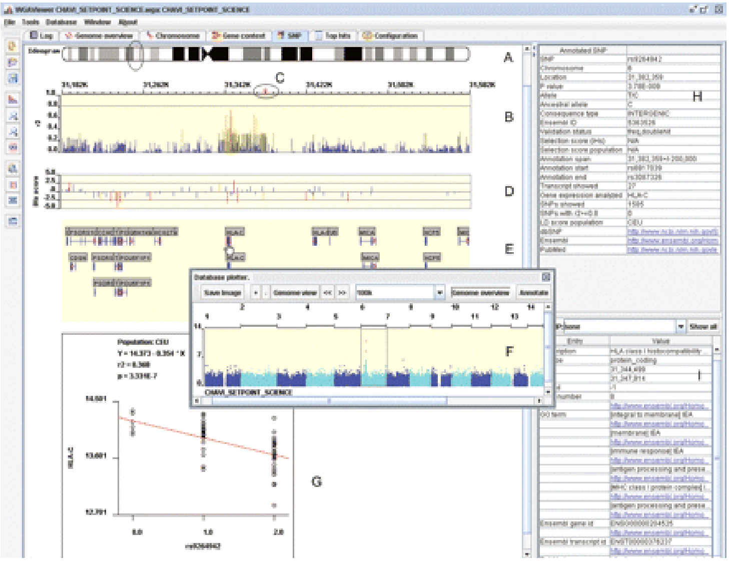


Figure 1.8: The figure from the WGAViewer paper showing their visualization interface. Used with permission.

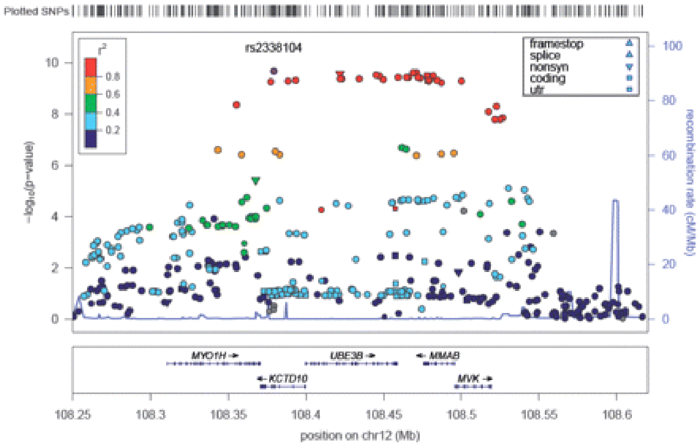


Figure 1.9: The figure from the LocusZoom paper showing the visualization interface. Used with permission.

plots the LD recombination information along the bottom axis. It also includes a summary of genes encoded in the genomic region along the axis. This is a great tool that incorporates data from multiple sources into a visually intuitive format. LocusZoom is especially powerful because of its simple layout. The use of color and shapes eliminates many of the charts that encumber previous visualizations. Although comparable to WGAViewer in functionality, LocusZoom's interface seems to be more compact and complete. See Figure 1.9 for an overview image of the LocusZoom layout.

In conclusion, many different tools have tackled the problem of visualizing the exploration of results from GWAS studies. Of these many tools, most have one feature that they particularly try to tackle – incorporation of data (Path), memory management of large datasets (AssociationViewer), or integration of multiple data types with exploratory tools (MAVEN). The development of each of these tools has ultimately led to complete and functional software, WGAViewer and LocusZoom, that incorporates all of these strategies to create excellent visualizations and great exploratory tools. LocusZoom is especially able to pick out the most important information in a GWAS analysis and display it in a succinct, uncluttered format.

1.3.4 Advances in eQTL study visualization

While many groups have worked on the visualization of eQTL study results, the software that has been developed thus far is still in its infancy and has not progressed to the same level as the tools available for GWAS analysis. Unfortunately, the presentation and visualization of eQTL results is still largely done through customized software or by using simple methods. For example, in a large-scale eQTL study, Zhu *et al.* used a series of long tables to represent results from eQTL analysis (2008). While the tables do convey some information, these tables are complex and do not highlight important information in a visually intuitive way. Another common method used to represent results from eQTL studies is through heat chart visualization (Zhao and Zhang 2010), Figure 1.10. The *y* axis in this visualization represents the SNPs involved in the eQTL study. On the *x* axis, the bins contain traits or genes measured in the study. The heat map is colored black if an eQTL is detected between the SNP (*y*-axis) and the trait (*x*-axis). This visualization is useful to give an overview of both *cis* and *trans* regulation genome-wide, although it hides small-scale regulatory information.

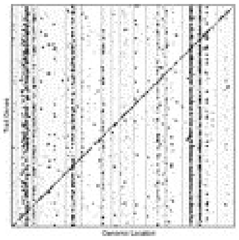
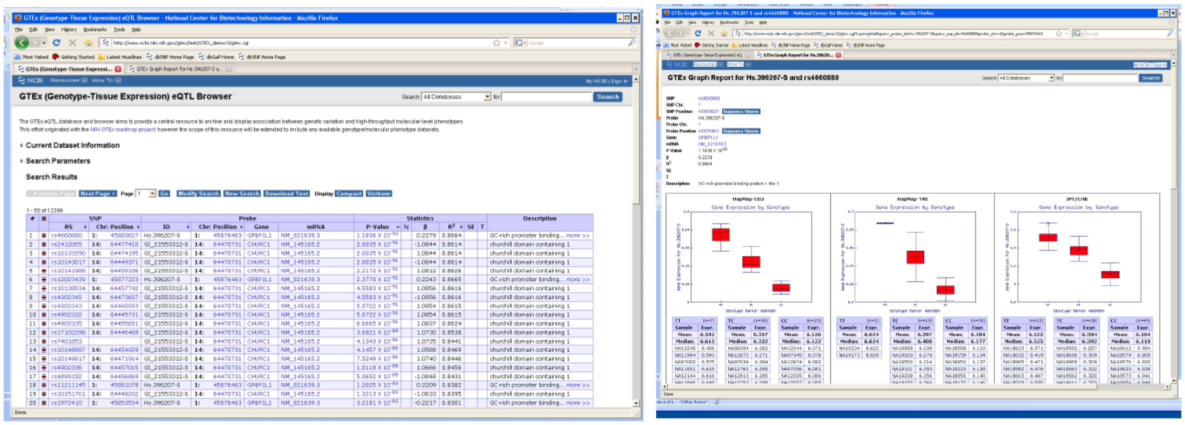


Figure 1.10: An example of a common visualization for eQTL studies taken from Zhang and Zhao (2010). Used with permission.

Aside from tables and customized scripts, there are a handful of eQTL visualization software platforms available. I begin by talking about two tools that are available to explore specific datasets, and then discuss some other visualization strategies that have surfaced in the literature. Finally, I review three general-purpose tools that are available. Although each tool has tackled some element of what is needed to explore eQTL results, there remains a great need for interactive tools that encompass the entire eQTL analysis pipeline on any dataset, from the overall *cis* vs *trans* big picture down to the individual SNP-gene regulatory networks.



GTEx (Genotype-Tissue Expression) eQTL Browser - National Center for Biotechnology Information - Mozilla Firefox

http://www.ncbi.nlm.nih.gov/gtex/test/GTEX_demo33/gtex.cgi

GTEx (Genotype-Tissue Expression) eQTL Browser

The GTEx eQTL database and browser aims to provide a central resource to archive and display association between genetic variation and high-throughput molecular-level phenotypes. This effort originated with the NIH GTEx roadmap project however the scope of this resource will be extended to include any available genotype/molecular phenotype datasets.

- Current Dataset Information
- Search Parameters

Search Results

1 - 50 of 12399

#		SNP		Probe				Statistics						Description
		RS	Chr: Position	ID	Chr: Position	Gene	mRNA	P-Value	N	β	R²	SE	T	
1	⊞	rs4660880	1: 45883627	Hs.396207-S	1: 45876463	GPBP1L1	NM_021639.3	1.1836×10^{-92}		0.2279	0.8664			GC-rich promoter binding... more >>
2	⊞	rs2412065	14: 64477410	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	2.0035×10^{-91}		-1.0844	0.8614			churchill domain containing 1
3	⊞	rs10133290	14: 64474195	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	2.0035×10^{-91}		1.0844	0.8614			churchill domain containing 1
4	⊞	rs10143017	14: 64449371	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	2.0035×10^{-91}		-1.0844	0.8614			churchill domain containing 1
5	⊞	rs10141986	14: 64459156	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	2.2172×10^{-91}		1.0812	0.8626			churchill domain containing 1
6	⊞	rs12023439	1: 45877223	Hs.396207-S	1: 45876463	GPBP1L1	NM_021639.3	2.3779×10^{-91}		0.2243	0.8665			GC-rich promoter binding... more >>
7	⊞	rs10138534	14: 64457742	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	4.5583×10^{-91}		1.0856	0.8616			churchill domain containing 1
8	⊞	rs4902345	14: 64473657	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	4.5583×10^{-91}		-1.0856	0.8616			churchill domain containing 1
9	⊞	rs4902343	14: 64465933	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	5.0722×10^{-91}		1.0854	0.8615			churchill domain containing 1
10	⊞	rs4902332	14: 64445731	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	5.0722×10^{-91}		1.0854	0.8615			churchill domain containing 1
11	⊞	rs4902335	14: 64455651	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	6.6665×10^{-91}		1.0837	0.8624			churchill domain containing 1
12	⊞	rs17102298	14: 64446489	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	3.6821×10^{-89}		1.0730	0.8538			churchill domain containing 1
13	⊞	rs7401853		GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	4.1343×10^{-86}		1.0735	0.8441			churchill domain containing 1
14	⊞	rs10148687	14: 64454029	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	4.1457×10^{-86}		1.0588	0.8469			churchill domain containing 1
15	⊞	rs10149617	14: 64471914	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	7.5249×10^{-86}		1.0740	0.8446			churchill domain containing 1
16	⊞	rs4902336	14: 64457015	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	1.0118×10^{-85}		1.0666	0.8456			churchill domain containing 1
17	⊞	rs4899152	14: 64456869	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	5.0652×10^{-85}		-1.0848	0.8431			churchill domain containing 1
18	⊞	rs11211145	1: 45881078	Hs.396207-S	1: 45876463	GPBP1L1	NM_021639.3	1.2025×10^{-83}		0.2209	0.8382			GC-rich promoter binding... more >>
19	⊞	rs10151701	14: 64448202	GI_21553312-S	14: 64470731	CHURC1	NM_145165.2	1.3213×10^{-83}		-1.0633	0.8395			churchill domain containing 1
20	⊞	rs1972410	1: 45853534	Hs.396207-S	1: 45876463	GPBP1L1	NM_021639.3	3.2181×10^{-82}		-0.2217	0.8301			GC-rich promoter binding... more >>

GTEx Graph Report for Hs.396207-S and rs4660880 - National Center for Biotechnology Information - Mozilla Firefox

GTEx Graph Report for Hs.396207-S and rs4660880

SNP	rs4660880
SNP Chr.	1
SNP Position	45883627
Probe	Hs.396207-S
Probe Chr.	1
Probe Position	45876463
Gene	GPBP1L1
mRNA	NM_021639.3
P-Value	1.1836×10^{-92}
β	0.2279
R²	0.8664
SE	
T	
Description	GC-rich promoter binding protein 1-like 1

Figure 1.11: Two snapshots from GTEx's website showing a tabular view and then box plots of the expression levels.

GTEx is a project currently under development. It will eventually handle different types of eQTL data, although at present it can only be used to analyze the one dataset it was built for. GTEx is available from NCBI and provides an interface to explore eQTL data (2010). It displays sorted results in a table format, and allows the analyst to see gene expression profiles by genotype (Figure 1.11). All data link out to information available on the web. This interface allows analysts to consider small-scale interactions in a dataset, one at a time, accessed through a list view.

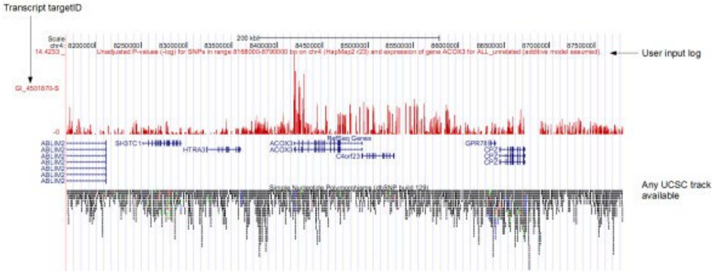


Figure 1.12: The figure from the SNPexp paper. Used with permission.

Another limited, but interesting visualization tool is called SNPexp. SNPexp is a project that incorporates data from the UCSC genome browser (Holm et al. 2010). SNPexp is only available to analyze two publically available datasets. However, the source code is available to be retrieved and used to apply to other datasets. In Figure 1.12, I show the figure from the original paper where they use a general line-plot of associations across a chromosome region with the data from UCSC below.

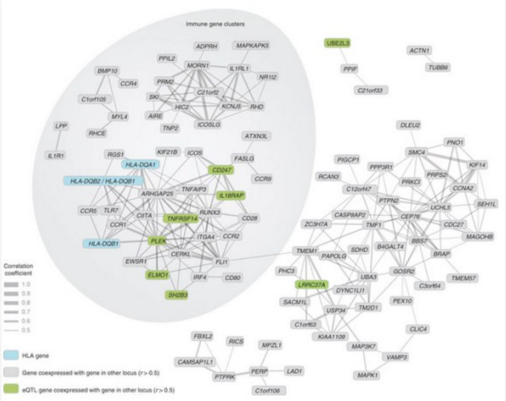


Figure 1.13: A figure from the Dubois et al. 2010 paper. Used with permission.

Of course, both of these visualizations look at one gene at a time and rely on search to find associations. In eQTL data, with hundreds of thousands of genes, this approach is limited. Indeed, Dr. Trey Ideker has talked about the need to consider association studies in terms of pathways, and there has been some talk about integrating association information with tools such as Cytoscape (Buckingham 2008). Steps in this direction have surfaced in the literature. For example, to summarize the results from an eQTL study, Dubois et al. color nodes in a gene network visualization (2010), Figure 1.13. In a similar way, other studies summarize their results using a colored network (Zhu et al. 2008), Figure 1.14. Aside from figures like these, however, there has not been work done to develop a tool that maps gene networks with association results by color. The ability to consider gene network structure simultaneously with association strength

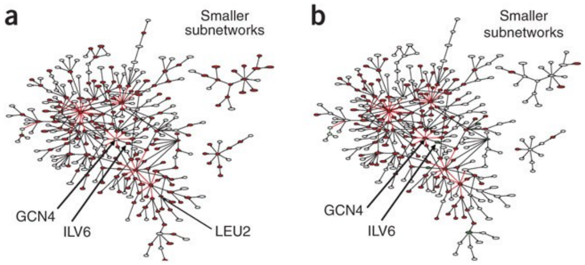


Figure 1.14: A figure from the Zhu et al. 2008 paper. Used with permission.

is especially important in structured association mapping where the structure guides the discovery of the associations.

To my knowledge, there have been three tools built specifically for general-purpose eQTL visualization. The first of these three tools is Genevar (Yang et al. 2010). Genevar is built for analyzing association on a small scale where a single gene is considered. In Figure 1.15, I show two visualizations from the original paper. The first visualization is the standard Manhattan plot combined with a table of SNPs. The analyst can use the tree view at the left to cycle through

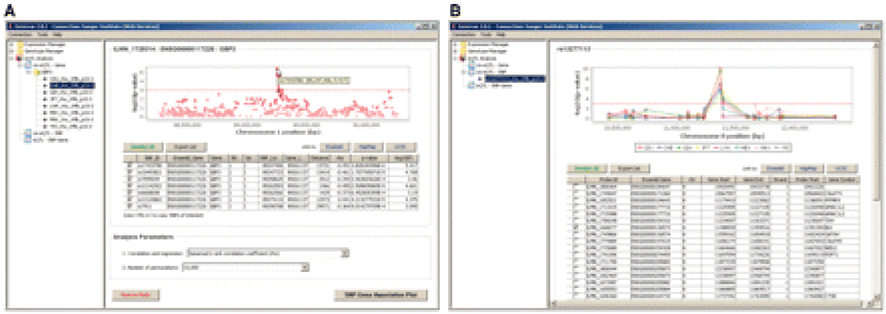


Figure 1.15: A figure from the Genevar paper. Used with permission.

the different gene expression probes one at a time. The second visualization uses multiple colors across populations. In this chart, SNPs surrounding the SNP in question are plotted according to their eQTL values. The analyst can choose the genes to include in the plot. Genevar largely attempts to use standard GWAS visualizations to display eQTL results, while considering populations in the data. While this is effective when only a handful of gene probes are considered, it quickly breaks down with large, complex gene association sets.

On the other hand, eQTL Explorer deviates from classic GWAS visualization techniques. eQTL Explorer is also built for the exploration of datasets with a handful of eQTLs (Mueller et al. 2005). eQTL Explorer presents the eQTLs in a genome representation, shown in Figure 1.16. External data can be imported manually to annotate the nodes. The decision to represent eQTLs as arrows is quite creative, but not scalable to large eQTL datasets. However, for researchers looking for a few particular points of interest, eQTL Explorer can be valuable.

Finally, the eQTL Viewer takes the common heat chart view and makes it interactive through the integration of external data (Zou et al. 2007). To list a few features of the eQTL Viewer, I present Figure 1.17, taken from the original paper. It is readily apparent that the visualization scheme here is built off of the common heat chart visualization used in eQTL analysis. However, a few notable features are emphasized in this figure. The authors have represented eQTLs by small dashes in the plot. By hovering over any eQTL, a list of genes encoded on the corresponding locus appear in the window to the right. Additionally, the analyst

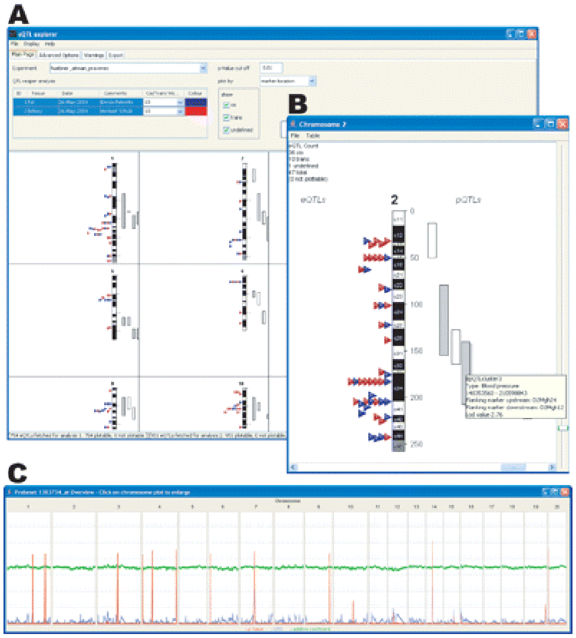


Figure 1.16: A figure from the eQTL Explorer paper. Used with permission.

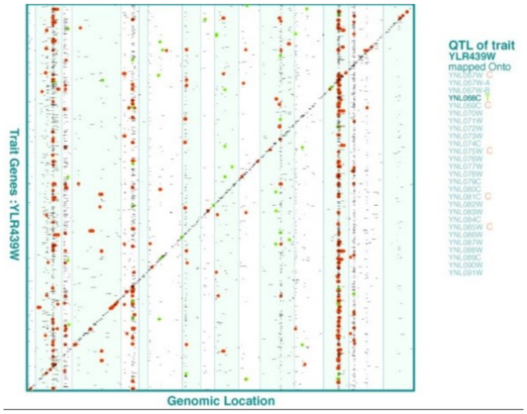


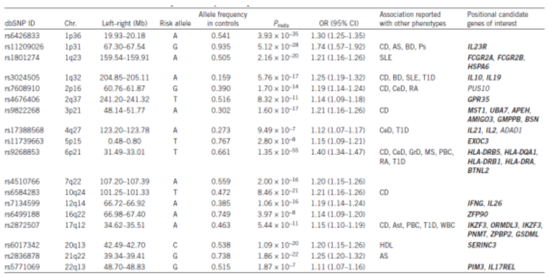
Figure 1.17: A figure from the eQTL Viewer paper. Used with permission.

can zoom in and out or sort the list of eQTLs by some category variable to get a big picture. Finally, the plot in Figure 1.17 has been colored so that all transcription factors known to regulate a certain pathway are colored in green, and all genes known to interact with the gene in question are colored red. These annotations also are displayed in the list to the right. eQTL Viewer takes a simple visualization and increases its usefulness by adding the information that analysts want to see. It does not consider relationships between genes, but it does allow for annotation and exploration.

In summary, the development of software tools for eQTL analysis is in its infancy. Most eQTL visualization is still done manually, through the customization of different general-purpose software. However, the development of eQTL Viewer has shown that by making common visualization techniques interactive and integrating them with external information, there is the potential for effective tools for eQTL analysis.

1.3.5 Visualization examples from current biology studies

The literature reporting results from GWAS and eQTL studies is constantly growing, and these papers almost always use some kind of visualization to represent results. The goal of this subsection is to provide an overview of current practices in representing results from GWAS and eQTL studies. I provide a review of how eight recent studies in GWAS and eQTL mapping did or did not use visualization in the presentation of results. First, I will discuss result reporting from recent GWAS.



dbSNP ID	Chr.	Left–right (Mb)	Risk allele	Allele frequency in controls	P_{meta}	OR (95% CI)	Association reported with other phenotypes	Positional candidate genes of interest
rs6426833	1p36	19.93–20.18	A	0.541	3.93×10^{-35}	1.30 (1.25–1.35)		
rs11209026	1p31	67.30–67.54	G	0.935	5.12×10^{-28}	1.74 (1.57–1.92)	CD, AS, BD, Ps	***IL23R***
rs1801274	1q23	159.54–159.91	A	0.505	2.16×10^{-20}	1.21 (1.16–1.26)	SLE	***FCGR2A, FCGR2B, HSPA6***
rs3024505	1q32	204.85–205.11	A	0.159	5.76×10^{-17}	1.25 (1.19–1.32)	CD, BD, SLE, T1D	***IL10, IL19***
rs7608910	2p16	60.76–61.87	G	0.390	1.70×10^{-14}	1.19 (1.14–1.24)	CD, CeD, RA	*PUS10*
rs4676406	2q37	241.20–241.32	T	0.516	8.32×10^{-11}	1.14 (1.09–1.18)		***GPR35***
rs9822268	3p21	48.14–51.77	A	0.302	1.60×10^{-17}	1.21 (1.16–1.26)	CD	***MST1, UBA7, APEH, AMIGO3, GMPPB, BSN***
rs17388568	4q27	123.20–123.78	A	0.273	9.49×10^{-7}	1.12 (1.07–1.17)	CeD, T1D	***IL21, IL2,*** *ADAD1*
rs11739663	5p15	0.48–0.80	T	0.767	2.80×10^{-8}	1.15 (1.09–1.21)		***EXOC3***
rs9268853	6p21	31.49–33.01	T	0.661	1.35×10^{-55}	1.40 (1.34–1.47)	CD, CeD, GrD, MS, PBC, RA, T1D	***HLA-DRB5, HLA-DQA1, HLA-DRB1, HLA-DRA, BTNL2***
rs4510766	7q22	107.20–107.39	A	0.559	2.00×10^{-16}	1.20 (1.15–1.26)		
rs6584283	10q24	101.25–101.33	T	0.472	8.46×10^{-21}	1.21 (1.16–1.26)	CD	
rs7134599	12q14	66.72–66.92	A	0.385	1.06×10^{-16}	1.19 (1.14–1.24)		***IFNG, IL26***
rs6499188	16q22	66.98–67.40	A	0.749	3.97×10^{-8}	1.14 (1.09–1.20)		***ZFP90***
rs2872507	17q12	34.62–35.51	A	0.463	5.44×10^{-11}	1.15 (1.10–1.19)	CD, Ast, PBC, T1D, WBC	***IKZF3, ORMDL3, IKZF3, PNMT, ZPBP2, GSDML***
rs6017342	20q13	42.49–42.70	C	0.538	1.09×10^{-20}	1.20 (1.15–1.26)	HDL	***SERINC3***
rs2836878	21q22	39.34–39.41	G	0.738	1.86×10^{-22}	1.25 (1.20–1.32)	AS	
rs5771069	22q13	48.70–48.83	G	0.515	1.87×10^{-7}	1.11 (1.07–1.16)		***PIM3, IL17REL***

Figure 1.18: A typical table from a GWAS study listing the top hits. This table was taken from the Anderson et al. 2011 paper and is used with permission.

Despite the use of visualization in many GWAS papers, it is also common to report the findings from a study in a table. This was done in a study that uncovered 29 susceptibility loci for ulcerative colitis, Figure 1.18 (Anderson et al. 2011). These tables show a "top hit" list of SNPs that were found in the study. These tables list mutations in candidate genes for further study. However, they provide little information about the biological mechanism behind the signal, limiting the information available to the reader. Despite their limitations, tables are used because they succinctly summarize the results from a study.

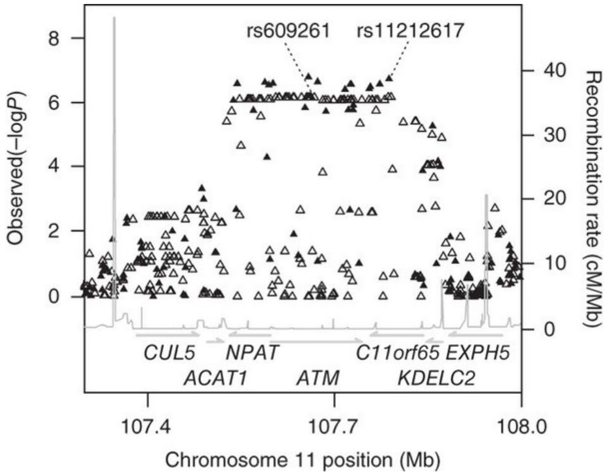


Figure 1.19: A typical plot to summarize significant SNP findings as shown in the study done by GoDARTs, UKPDS Diabetes Pharmacogenetics, & WTCCC 2011. Used with permission.

Manhattan plots are often used in GWAS as an alternative to tables. As an example of an effective use of a Manhattan plot, in Figure 1.19 I present the plot used in a GWAS looking for glycemic response to a drug in individuals with Type 2 diabetes (The GoDARTS and UKPDS Diabetes Pharmacogenetics Study Group & The Wellcome Trust Case Control Consortium 2010). This is a Manhattan plot similar to that seen in many studies, and it was probably generated using R. Note the names of the genes listed across the bottom and the SNPs studied labeled on top. Additionally, the recombination rate is displayed to show which SNPs are in LD. This plot is an improvement over the table shown in Figure 1.18 because the visualization includes more biologically relevant information about the significant SNPs, suggesting the effect they might have on genes in the cell. This visualization does not give us a genome-wide picture like the table did, but it allows us to see what might be going on around the SNPs that were discovered.

Similarly, in Figure 1.20 I show the results from a GWAS study looking for SNPs associated with psoriasis in humans. The authors demonstrate the significance of the susceptibility loci they found using a Manhattan Plot (Stuart et al. 2010). In this plot, the recombination rate is not shown, but rather the LD across the chromosome is shown. The authors

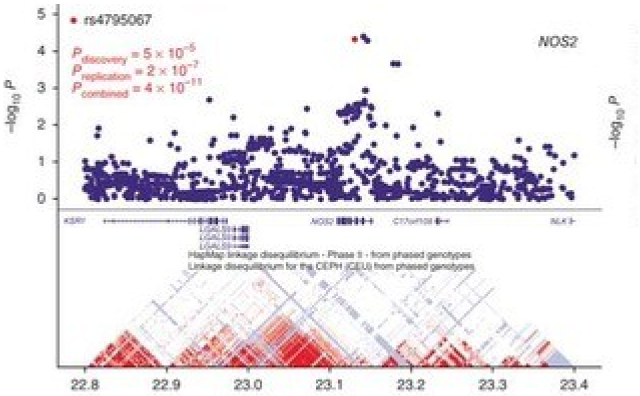


Figure 1.20: The figure from the Stuart et al. 2010 paper shows LD between the SNPs, as well as gene location. Used with permission.

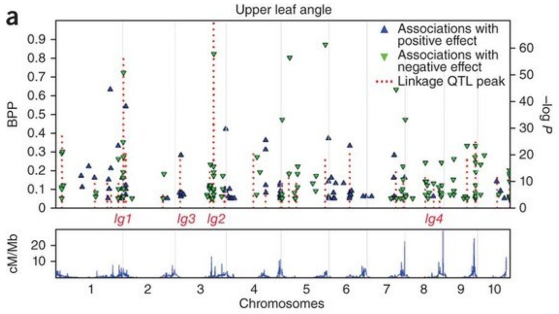


Figure 1.21: A genome-wide Manhattan plot from the Tian et al study (2011). Used with permission.

use color to point the reader's attention to the most important SNP that they have uncovered in this study. This plot is similar to the one presented in the Type 2 diabetes study.

Manhattan plots are not only used to display information for specific chromosomal regions, but they are often used to show patterns across the entire genome. For example, to summarize the results from a study on maize leaf architecture, Tian et al. (2011) use a Manhattan plot to show which alleles have a negative vs. a positive association to leaf shape. The figure from the original paper is shown in Figure 1.21. The lower plot shows the recombination rate across the genome. This plot shows the limitations of using a Manhattan plot for a genome-wide summary. The SNPs are necessarily sparse across the genome, and it is difficult to see where the mutations actually are in relationship to the genes (aside from the labels the authors provide). It seems that this information could have been communicated just as clearly using a table. This

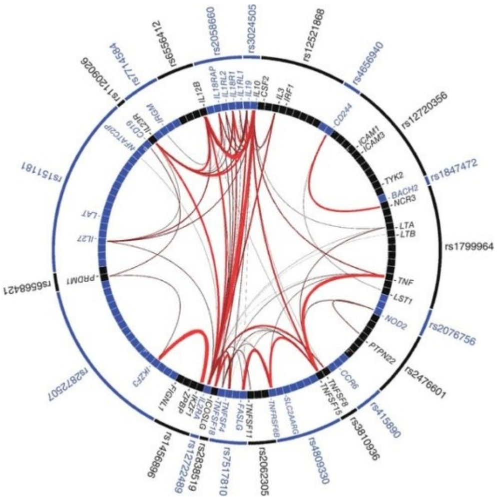


Figure 1.22: The figure from the Franke et al. 2010 paper. In this view, they show links between genes associated with susceptibility SNPs. Used with permission.

example shows how the Manhattan plot visualization, although useful for small-scale visualizations, is limited when expanded to the genome scale.

Manhattan plots, although common, are not the only visualization strategy used to explore the results from a GWAS study. To understand the links between different susceptibility genes in Crone's disease, a circle plot is shown in Figure 1.22 (Franke et al. 2010). Each of the significant SNPs is plotted around the edge, corresponding to a number of genes. Links between the genes are determined algorithmically, and these weighted connections are represented by lines. This visualization is useful to demonstrate evidence-based connectivity between genes in the 71 confirmed Crone's disease susceptibility loci. This type of visualization could be applied in future studies, especially in eQTL studies that also consider relationships between genes. Note that this visualization has a different purpose than the other GWAS visualizations I've considered and that is a good fit for what the authors are trying to convey. As researchers try to uncover the complex picture of gene regulation, there will be cases that unique visualizations will be needed to solve unique problems.

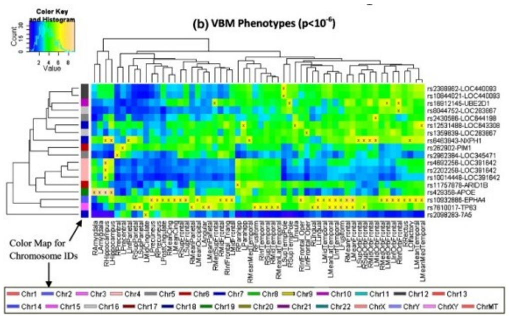


Figure 1.23: A heat map visualization from eQTL data from the Shen et al. study (2010). Used with permission.

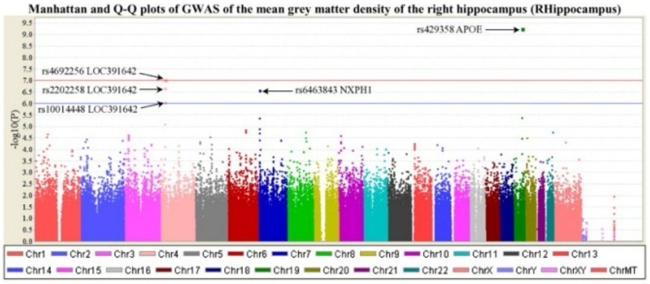


Figure 1.24: A genome-wide Manhattan plot used to summarize the results in the Shen et al. study (2010). Used with permission.

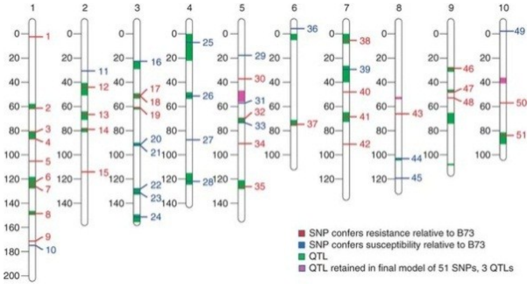

SNP confers resistance relative to B73
SNP confers susceptibility relative to B73
QTL
QTL retained in final model of 51 SNPs, 3 QTLs

Figure 1.25: A chromosome visualization points out the location of eQTLs for maize southern blight. Figure from original paper and is used with permission.

Moving on towards eQTL visualization in the literature, I first consider the Alzheimer's Disease Neuroimaging Initiative study that collected data from imaging neurological scans and genotyping of over 1500 patients (Shen et al. 2010). To report the significant associations they found from the genome to traits associated with the Alzheimer's disease, these researchers used a heat map strategy as shown in Figure 1.23. To summarize their GWAS results, the authors used a general-purpose Manhattan Plot, shown in Figure 1.24. These kinds of plots can be generated using the Golden Helix software and are common in GWAS studies, despite the limited information that they offer. The plot gives a general picture of association signals across the genome and how significant those signals are, similar to a table. The heat map plot shown in Figure 1.23 is more interesting. Given a number of genes, the strength of the eQTL is shown between several selected SNPs and the gene. This plot’s usefulness is enhanced due to the selection of the SNPs, allowing us to consider only the SNPs with the strongest associations. In Figure 1.23, the authors show what SNPs affect genes related through a tree structure shown on the horizontal axis.

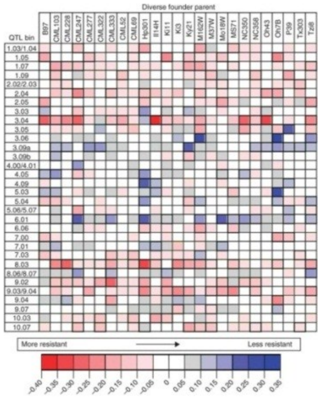


Figure 1.26: A heat map view compares families of maize for southern blight resistance. Figure from original paper and used with permission.

A similar approach was used in a study looking for a genetic cause to resistance to southern leaf blight in maize; the authors employ several different visualizations to summarize qualitative trait locus (QTL) data (Kump et al. 2011). The location of the QTLs is shown using

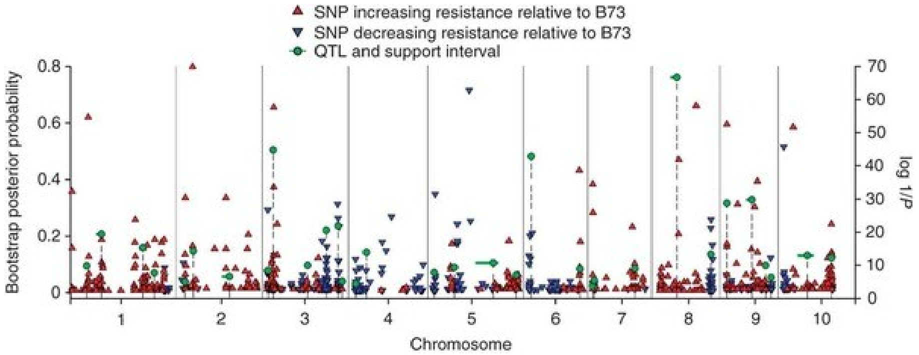
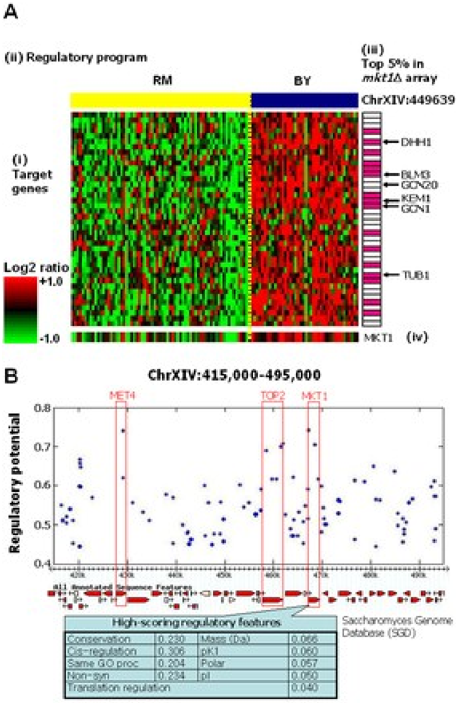


Figure 1.27: A Manhattan plot showing the associations to Southern blight across the maize genome. This figure is from the original paper and is used with permission.

color on a chromosome view (Figure 1.25). They also use a heat map to look at association strengths across different families in Figure 1.26. Finally, they use a Manhattan plot to visualize the significant SNPs and the effect on blight resistance in Figure 1.27. This study used a variety of visualizations to help the reader get a general feel for where the QTL associations lie across the maize genome and to see how these QTLs differ across families. The adaption of these visualizations is effective.



High-scoring regulatory features			
Conservation	0.230	Mass (Da)	0.066
Cis-regulation	0.306	pK1	0.060
Same GO proc	0.204	Polar	0.057
Non-syn	0.234	pI	0.050
Translation regulation			0.040

Figure 1.28: Lee et al. (2009) use a variety of visualizations to move from the overall big picture into the small picture of gene regulation. Used with permission.

Finally, I present the visualization used to summarize the results from a structured association mapping analysis (Lee et al. 2009). In this analysis, the authors use tables to describe the overall results. To get a small-scale picture, the authors zoom in and use charts like the one in Figure 1.28. In Figure 1.28b, the authors use a Manhattan Plot to show the SNPs with the highest "regulatory potential," and simultaneously show the genes encoded in the genomic region of interest. This chart appears to have been generated using the combinations of different tools, such as the SGD (*Saccharomyces* Genome Database 2011) and R. The authors also show expression levels for the genes using a heat map. This heat map is informative, although it is not connected to the association values. I present this visualization as another example where the authors combined several visualization techniques to show the overall picture of regulation and to zoom down into the specific biological picture.

In this section, I have discussed several figures from recent papers that show the current state of the art methods in visualizing results from GWAS and eQTL studies. A few themes stand out: 1) Manhattan plots are often useful visualizations and are familiar to geneticists, although they provide limited information when used on a genome-wide scale, 2) heat maps are a common visualization used to describe associations across genes or families and can be effective at showing an overall picture of association patterns, 3) authors often have to be creative and combine information and/or visualization techniques from several sources in order to create informative figures, and 4) as demonstrated by the Lee et al. (2009) analysis, moving from the big picture to the small picture is also important, and proper tools can assist in this process.

In regards to this section, I would also like to note that while many of these visualization techniques are adept and conveying information, they are not always as good at leading the analyst to the interesting information. In an analysis where only a handful of associated SNPs are identified, this is not a problem. However, in a structured association mapping situation where there may be hundreds or thousands of associations, visualization strategies must also be adept at guiding the analyst to interesting signals in the data. Thus, the visualizations presented here must be considered in both how they lead the analyst to discovery and how they convey information.

1.3.6 Conclusions

In this review, I have discussed association mapping tools available for both GWAS and eQTL studies. I have also reviewed current papers to point out strategies currently employed to visualize results from these types of studies. The visualization of these types of data is important to facilitate biological discovery and to convey results.

While excellent tools now exist for the summary and exploration of GWAS results, the development of tools for eQTL analysis has lagged behind. This becomes especially pertinent when structured association mapping comes into play. By including hundreds or thousands of response traits, the analysis quickly overwhelms current tools and strategies.

Although I have focused on tools specifically built for GWAS and eQTL analysis thus far, there are other software systems that allow analysts to explore structure in the data. Significant work has gone into the development of visualization for the genome and for visualizing gene-gene relationships. A popular tool for gene network visualization, Cytoscape, is one example of tools used in genetic network visualization (Shannon et al. 2003). Tools like Cytoscape could potentially be expanded to incorporate association analysis (Buckingham 2008).

Just as with trait networks, significant effort has gone into the visualization and exploration of genomes. One popular interface is the UCSC Genome Browser. Projects such as ENCODE are ongoing to incorporate more information into the genomic visualization (Rosenbloom et al. 2009). An example visualization from the UCSC Genome Browser is shown in Figure 1.29. The location of genes, recombination rates, SNP frequencies and locations are all important pieces to the visualization.

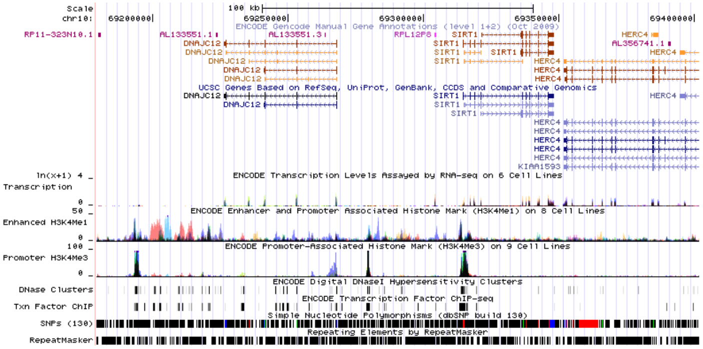

Scale
100 kb
chr10:
69200000
69250000
69300000
69350000
69400000
ENCODE Gencode Manual Gene Annotations (level 1+2) (Oct 2009)
RP11-323N10.1
AL133551.1
AL133551.3
RPL12P8
SIRT1
HERC4
AL356741.1
DNAJC12
UCSC Genes Based on RefSeq, UniProt, GenBank, CCDS and Comparative Genomics
KIAA1593
ln(x+1) 4
Transcription
0
ENCODE Transcription Levels Assayed by RNA-seq on 6 Cell Lines
50
Enhanced H3K4Me1
0
ENCODE Enhancer and Promoter Associated Histone Mark (H3K4Me1) on 8 Cell Lines
100
Promoter H3K4Me3
0
ENCODE Promoter-Associated Histone Mark (H3K4Me3) on 9 Cell Lines
DNase Clusters
ENCODE Digital DNaseI Hypersensitivity Clusters
Txn Factor ChIP
ENCODE Transcription Factor ChIP-seq
SNPs (130)
Simple Nucleotide Polymorphisms (dbSNP build 130)
RepeatMasker
Repeating Elements by RepeatMasker

Figure 1.29: An example genomic visualization by the UCSC Genome Browser. This image is a screen shot taken using the website.

Thus, although the current visualization strategies that have been developed are insufficient when it comes to structured association mapping, a historical foundation has been built. It is off of this foundation that I build GenAMap and design new visualizations that 1) convey the results of structured association mapping in a succinct, clear manner, and 2) facilitate the exploration and analysis of large-scale structured association mapping studies. See Sections 3-5 for more detail.

1.4 The promise of visualization in dynamic gene network analysis

In this section and the following subsections I will consider another area of genetics research, the study of gene-gene interactions represented by networks. This research has implications in the study of diseases like cancer, as well as in understanding biological processes, such as the cell cycle. The promise of visualization in network analysis, especially dynamic network analysis, has motivated the development of TVNViewer, which is presented in Section 6 of the dissertation.

Today's rapid development of high-throughput technology and increasing amounts of biological data promises greater insight into the complex interactions that govern cellular function. In particular, high-throughput technology that measure gene expression levels has already led to a greater understanding of genetic interactions and the complex regulatory circuitry in the cell. Gene expression measurements can be used to infer network relationships between genes in a cell, potentially uncovering important interactions that perturb the cellular state (Li and Chan 2004; Basso et al. 2005; Segal et al. 2003; Schafer and Strimmer 2005). Understanding these network relationships between genes can lead to greater insight into common cellular processes, such as the cell cycle or disease progression (Schadt 2009).

Traditionally, genetic networks have been analyzed as static entities. However, many biological processes, such as developmental processes and disease progression, are context-dependent and temporally specific. As a result, relationships between molecular constituents evolve over time and react to changing environments. Representing these dynamic interactions with a single static network limits the insight available in the analysis of the network. Early

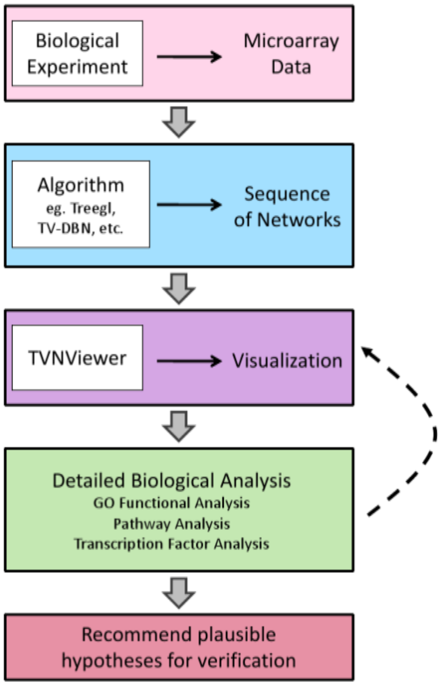


Figure 1.30: Dynamic network analysis pipeline A dynamic network analysis consists of several stages beginning with data collection and leading to generating new hypotheses for further study. First, gene expression data (usually microarray data) is collected across several time points or tissue types. Once the data is preprocessed, machine learning techniques are employed to determine relationships between genes. As a result, a series of networks are created that can then be explored using TVNViewer. Detailed biological analyses are then carried out leading to specific hypotheses that can then be validated experimentally.

studies of dynamic networks indicated that changes in network architecture resulting from processes such as the cell cycle or responses to external stimuli are quite significant (Luscombe et al. 2004). Over the last five years, by studying dynamic gene-gene relationships, biologists have attained a deeper knowledge of the functional and regulatory underpinnings of complex biological processes (Calvano et al. 2005; Dupuy et al. 2007; Jiao et al. 2009; Keller et al. 2008). As techniques in dynamic network analysis continue to advance, we will better understand the systematic rewiring of the transcriptional regulatory circuitry that controls cell behavior.

A dynamic network analysis begins with data collection and the creation of a series of gene-gene interactions (networks) from the data (see Figure 1.30). Dynamic gene expression data are generally available as microarray samples that are collected over a time course or under multiple conditions. Many cutting-edge machine learning techniques are available to fully leverage the information stored within the data to create a series of related, evolving gene networks. Here, I list a few of these strategies. TESLA and KELLER build off sparse regression techniques (Ahmed and Xing 2009; Song et al. 2009) and TV-DBN (Song et al. 2009) estimates a chain of evolving networks using time-varying dynamic Bayesian networks. In addition, Robinson and Hartemink suggest learning a non-stationary dynamic Bayesian network using Markov Chain Monte Carlo sampling (2010) and Lozano et al. propose using the notion of Granger causality to model causal relationships among variables over time (2009). In contrast to linear time-varying networks, Treegl is a method for analyzing networks that evolve over tree-shaped genealogies (such as stem cell differentiation) (Parikh et al. 2011). Each of these strategies can be used to recover a series of networks from dynamic network data for further analysis.

Once a series of networks is available for analysis, these networks must be explored to find the subtle (and obvious) changes in network topology. Analysts must also consider the roles that genes play in the network rewiring to find key regulators that drive the network evolution. At this point, the focus of the investigation becomes more exploratory than query driven. Information visualization, then, is again a powerful tool that be applied to this context (Card et al. 1998). Information visualization researchers have developed techniques for dynamic network visualization which often span fields, but require customization for maximizing analytic impact in a given implementation (Smith et al. 2009; Bertini et al. 2008). Visualization techniques excel when providing an explanation of the overall structure of the data or finding weak or unexpected patterns most easily recognized by humans (Card et al. 1999), critical requirements for dynamic gene network analysis.

Because visualization naturally enables gene network analysis, many visualization tools have been developed to explore biological networks including Cytoscape (Shannon et al. 2003), Osprey (Breitkreutz et al. 2003), VisANT (Hu et al. 2009), and Graphle (Huttenhower et al. 2009). Although there are many of these tools, the state-of-the-art tools in biology network analysis do not support the exploration of dynamic rewiring network data (Pavlopoulos et al. 2008; Suderman and Hallett 2007). Fortunately, information visualization researchers have developed and evaluated techniques for dynamic network analyses of numerous kinds in other contexts, including social networks (Bonsignore et al. 2009; Heer and Boyd 2005), internet traffic networks (Cox and Patterson 2011), and even literature networks (van Ham et al. 2009). Additionally, in my own experience working with others who explore dynamic gene networks, I have found that multiple networks need to be visualized simultaneously and in real-time. To explore these networks, analysts must have access to load and view a large number of networks and must rapidly switch between networks to compare the different topologies. Thus, given my experience, the experience of collaborators, and current visualization research, I have found that the visualization tools available for gene network analysis, such as Cytoscape, are insufficient to support the analysis of a large number of rewiring networks.

To create a visualization tool that enables exploratory dynamic network analysis, I led the development of TVNViewer (Section 6), an online visualization tool specifically designed to support the discovery of spatial or temporal changes in network topology via exploration (Curtis et al. 2011; Curtis et al. 2012). In addition to facilitating exploratory analysis, TVNViewer also allows analysts to create the intuitive visualizations that they need to present their discoveries.

1.5 Current visualization strategies in genetic network analysis

Over the last 20 years, microarray and other developing technologies have allowed for high-throughput measurement of gene expression levels. Thus, gene expression studies have become common due to their potential to uncover the complex interplay of genetic polymorphisms, genes, and transcription factors in biological systems (Montgomery and Dermitzakis 2009). Due to the vast amount of data available from these studies, and the complex relationships that exist in the data, it is still a challenge in systems biology to present the information in a meaningful way where the information can be naturally explored and understood (Pavlopoulos et al. 2008). In network analysis, geneticists need to understand the overall structure of the data and to find weak patterns in the data, naturally suggesting a visualization approach (Card et al. 1999). Indeed there is a plethora of visualization tools available to explore gene-gene interaction data (Pavlopoulos et al. 2008; Suderman and Hallett 2007). In this section, I briefly report on some of the tools and strategies available to analyze gene expression data and the relationships between genes in that data.

One popular method to present and interpret the results from gene expression studies is the 2D heat chart created using red and green colors; the genes are clustered by hierarchical clustering, showing the levels of gene expression for the different genes and clustering genes with similar patterns together. This approach shows overall patterns in large datasets, and can be especially effective with small datasets. The 2D heat charts do provide initial information about the gene expression data, however, these charts are limited in the amount of information that they can provide. Often, additional analysis is used to convert the raw data into more complex data structures, such as networks, for analysis.

There are many simple and sophisticated methods available to create gene-gene networks from gene expression data; examples include the glasso and the soft-thresholding method (Friedman et al. 2007; Zhang and Horvath 2005). Creating a network from gene expression data can elucidate the relationships between genes and help identify patterns in the data. Once created, these networks are usually represented visually as graphs, where vertices represent genes and edges between vertices represent some kind of connection between the genes (Pavlopoulos et al. 2008). In the case of regulatory networks, the edges are directed from regulator to regulated, and the graphs can also be undirected. Gene expression studies can include a handful of genes, or up to thousands or tens of thousands of genes.

Because of the heterogeneity of gene expression studies, there are many tools currently available that support the visualization of gene network data. Two excellent reviews of these tools (Pavlopoulos et al. 2008; Suderman and Hallett 2007) cover up to 50 different network analysis tools that are available. Here I will briefly cover Cytoscape, Osprey, VisANT, and

Graphle. My purpose is to convey the strategies used in gene network visualization, which I take advantage of in the development of GenAMap and TVNViewer. Additionally, as both reviews point out, current tools do not support dynamic network analysis, which is apparent through the review of these four tools.

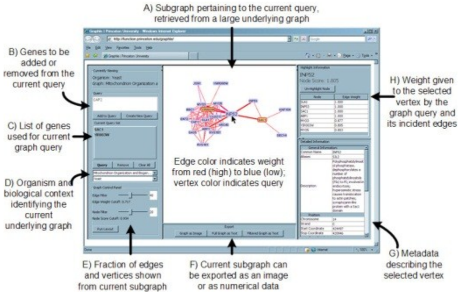

A) Subgraph pertaining to the current query, retrieved from a large underlying graph
B) Genes to be added or removed from the current query
C) List of genes used for current graph query
D) Organism and biological context identifying the current underlying graph
Edge color indicates weight from red (high) to blue (low); vertex color indicates query
H) Weight given to the selected vertex by the graph query and its incident edges
E) Fraction of edges and vertices shown from current subgraph
F) Current subgraph can be exported as an image or as numerical data
G) Metadata describing the selected vertex

Figure 1.31: Graphle. Graphle is a query tool that allows analysts to store large gene networks on a server and then access specific sub-networks through query. Figure is from the original paper and is used with permission.

Graphle (Huttenhower et al. 2009) is an online interface for visualizing networks with "tens of thousands of vertices (e.g. genes) and hundreds of millions of edges" (e.g. interactions). Graphle allows the analyst to query an online system to select specific portions of large network graphs. I show the figure from the paper in Figure 1.31. The interface is designed around the network query structure, and the graph is simply displayed as a node-link graph with edges connecting genes in the network.

Osprey (Breitkreutz et al. 2003) is visualization software built for the exploration of

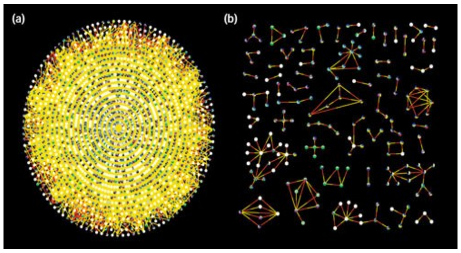


Figure 1.32: Osprey Osprey allows analysts to query for sub-networks in large network graphs. Coloring is used to color nodes by GO annotation and edges by experimental validation. Figure is from original paper and is used with permission.

complex interaction genetic networks. Osprey utilizes color to represent gene function and the experimental validation of relationships between genes (Figure 1.32). Similar to Graphle, Osprey allows users to perform text-search queries to look for genes and sub-networks in large graphs.

VisANT (Hu et al. 2009) is an integrative platform for biological interaction data. Similar to Osprey, VisANT takes advantage of GO enrichments and annotations. However, VisANT provides an expanded toolkit to find over-representation enrichments in user-specified sub-networks. VisANT also integrates the KEGG and Predictome databases to support exploratory pathway analysis (Hu et al. 2007). The integration of multiple data types in VisANT provides genetic network analysts with powerful tools to explore the relationships between genes.

Perhaps the most popular network visualization tool is Cytoscape (Shannon et al. 2003). Cytoscape uses node-link representations of networks with a variety of formats and visualization techniques. Cytoscape has a flexible architecture that allows others to develop plug-ins to add functionality, analysis tools, and external data directly into the system. For example, a Cytoscape

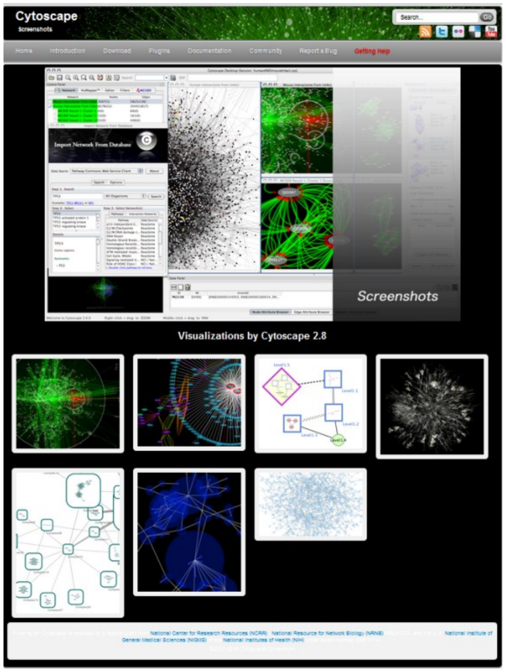

Cytoscape
Screenshots
Search...
Go
Home
Introduction
Download
Plugins
Documentation
Community
Report a Bug
Getting Help
Import Network From Database
Screenshots
Visualizations by Cytoscape 2.8
National Center for Research Resources (NCRR)
National Resource for Network Biology (NRNB)
National Institute of General Medical Sciences (NIGMS)
National Institutes of Health (NIH)

Figure 1.33: Cytoscape. A screen-shot from the Cytoscape website that demonstrates many of the visualization techniques that Cytoscape uses to visualize genetic networks.

plug-in that I customized for GenAMap is the BiNGO (Maere et al. 2005) plugin. This plug-in allows analysts to perform GO enrichment analyses directly from Cytoscape (and now GenAMap).

Indeed, there has been a lot of research in the field of genetic network visualization. The representation of genes as nodes, and edges between nodes representing a relationship between the genes, is common across almost all network visualization tools. In some papers, large networks are also represented by a heat chart (Zhu et al. 2008), where dark pixels represent an association between the genes, which are usually clustered along the axes. This strategy is also used to display large networks in a more general purpose visual analytics tool called Orion (Heer and Perer 2011). In GenAMap I have taken advantage of the research that has been done in genetic network visualization to build a network visualization tool. As I will show, GenAMap development has integrated several of these strategies including the 2D heat chart, the node-link representation, and the integration of external data. GenAMap presents a coherent visualization that is extended to association analysis, which uses multiple-coordinated views to integrate the structure visualization with the association data. I also have used these strategies as a springboard into the development of TVNViewer, which is the first such genetic visualization tool to allow geneticists to explore the dynamic relationships between genes as they change across time and space.

2. Structured Association Mapping

In the previous chapter I introduced structured association mapping and considered several algorithms that have been developed. Chapter 2 is devoted to a more in-depth discussion about structured association mapping.

In Section 2.2, I discuss GFlasso in greater detail (Kim and Xing 2009). GFlasso was my first introduction to structured association mapping. I arrived at CMU just when GFlasso was being finalized, and I got to be a small part of the PLoS Genetics paper. I did the math to solve the update equations and implemented the first GFlasso implementation that was distributed. I also ran the running time tests that were reported in the paper. The project ended up being the starting point for my work, as we first discussed creating "GFlasso software", which evolved into GenAMap. In this chapter, I introduce the basic intuition behind GFlasso and why it works, as it will show up again and again throughout the dissertation in the biological analysis chapters and in GenAMap.

In Section 2.3, I present results from a simulation study that investigated the behavior of GFlasso when compared with other association mapping algorithms. If the assumptions of

GFlasso are correct, GFlasso outperforms other methods for finding associations in terms of true positive rate (TPR) and false-discovery rate (FDR). In Section 2.4, I extend GFlasso to create gGFlasso, a new structured association mapping approach. gGFlasso is used in a two-step process (GFlasso-gGFlasso) to find three-way associations between genome-transcriptome-phenome data. Validation of gGFlasso through simulation is presented in Section 2.5.

Chapter 2 shows the potential of structured association mapping via simulation study, and also presents to the reader that basic intuition of what goes into the design and implementation of structured association mapping algorithms.

2.1 Problem Formulation

Let $\boldsymbol{X}$ be an $N \times P$ genotype matrix for N individuals where each row represents the allele states of an individual at P loci; let $\boldsymbol{Y}$ be an $N \times J$ gene expression matrix where expression levels of J genes are measured for the same set of individuals; finally, let $\boldsymbol{Z}$ be an $N \times K$ phenotype matrix where each row records K phenotypic traits of an individual. We assume that there are two *undirected weighted* relevance graphs $G_G = (V_G, E_G)$ and $G_T = (V_T, E_T)$ available for J genes and K traits, respectively. Each node in G_G represents a gene and each node in G_T represents a trait, hence $|V_G| = J$, $|V_T| = K$. There is a vast literature on network construction algorithms, and any strategy can be applied to create a network where a connection between nodes u and v represents a relationship between nodes. Alternatively, gene-gene relationships can be curated from the literature to create such a network. Each edge $\{u, v\} \in E_G$ or E_T is associated with a weighted connection between nodes.

2.2 GFlasso is a structured association mapping method that maps SNPs to multiple correlated response traits

Since the advent of eQTL analysis, different statistical approaches have been developed that go beyond traditional single-trait analysis to unravel the complex patterns of association between the genetic variants and expression levels. In particular, Dr. Seyoung Kim (joint work with Dr. Eric Xing) has developed a statistical method called GFlasso, which leverages the full gene-expression network to guide a search for genotypes that influence genes whose expression levels are highly correlated (Kim and Xing 2009). To my knowledge, this method is the first that systematically exploits the full gene correlation network in eQTL analysis. This method finds associations between SNPs and sub-networks of correlated genes within the full network, avoiding many of the fundamental limitations of previous methods. For example, performing association analysis using the PCA-based method on transformed traits sacrifices the interpretability of the results. *Lirnet* uses gene expression levels averaged over genes within each cluster and then maps these averages to genetic loci (Lee et al. 2009). In this case, however, the averaging operation can lead to the loss of information on the activity of individual genes, especially genes whose expression levels are negatively correlated. Although Zhu et al. (2008) and other work by the same group of researchers takes advantage of the gene network in eQTL analyses, they do so only as a post-processing step after finding eQTLs for each gene separately, rather than directly using the network during the search for eQTLs.

GFlasso is a sparse multivariate regression method that was developed for finding a correlated genome association for multiple related traits given a trait network and genotype/phenotype dataset (Kim and Xing 2009). GFlasso extends the standard lasso (Tibshirani 1996) that uses an L_1 penalization to shrink the regression coefficients (or parameters for association strengths) towards zero and obtain a sparse estimate with many zero-valued coefficients for SNPs with no associations. In this case, SNPs are treated as predictors and gene expression levels are treated as responses. GFlasso assumes that a relatively small number of markers are associated with each gene and that highly correlated genes tend to be influenced by a common subset of SNPs. This assumption is explicitly expressed as two regularization terms in a linear regression model (Eq. 2.1). Input to GFlasso includes ***X***, ***Y***, and G_G; the output is a regression coefficient matrix $\boldsymbol{B_1}$, where B_{pv} denotes the strength of *p*th SNP associated with *v*th gene. Given an *N* x *P* genotype data matrix ***X***, where *N* is the number of strains and *P* is the number of SNPs, and an *N* x *J* gene-expression data matrix ***Y***, where *J* is the number of genes, GFlasso estimates the regression coefficients $\boldsymbol{B_1}$, a *P* x *J* matrix, for association strengths by solving the following optimization problem:

$$\boldsymbol{B_1} = \underset{\boldsymbol{B} \in \mathbb{R}^{P \times J}}{argmin} \|\boldsymbol{Y} - \boldsymbol{XB}\|_F^2 + \lambda \sum_j \sum_p |B_{pj}| + \gamma \sum_{\{u,v\} \in E_G} \sum_p |B_{pu} - sign(\rho_{uv}) B_{pv}| \qquad (2.1)$$

where $\|.\|_F$ is the Frobenius norm of the matrix. The second term is the L_1 lasso penalty, which has the property of shrinking the strengths of irrelevant SNPs towards zero; the last term is the *graph-guided fused lasso* penalty, which encourages highly correlated genes (connected by an edge in E_G) to be associated with the same SNPs. sign(ρ_{uv}) controls the pattern of fusion applied to the association strengths: if two genes are negatively correlated, their corresponding association strengths are encouraged to have different signs. For each gene, the set of associated SNPs are those with non-zero association strengths in the estimated coefficient matrix $\boldsymbol{B_1}$. The λ and γ in the above equation are the regularization parameters that control the amount of penalization. In order to find associations using GFlasso, GenAMap finds values for λ and γ through a linear search strategy based on cross-ten-validation. Once GenAMap obtains the optimal estimate of association strengths $\boldsymbol{B_1}$, all SNP/gene pairs corresponding to non-zero entries in $\boldsymbol{B_1}$ are considered to be significant SNP-gene associations. GenAMap uses a fast proximal gradient method to optimize GFlasso given the regularization parameters (Chen et al. 2010).

2.3 Investigating structured association mapping through simulation

Researchers employ a variety of methods to find genome-phenome associations. These methods include non-parametric methods like the Wilcoxon Sum-Rank test (Zhu et al. 2008), model-selection regression methods such as the forward selection and lasso (Tibshirani 1996; Wu et al. 2009), and parametric methods such as maximum likelihood and the Wald test implemented in software packages such as PLINK (Purcell et al. 2007). Because the ground truth of the SNP-gene expression associations is not known, it is difficult to directly compare the results of different methods. Simulation study is one method that can be used to access the performance of different methods, given specific assumptions about what is happening in the underlying biological system.

Although simple simulation studies have previously been performed for structured association mapping algorithms such as TreeLasso and GFlasso (Kim and Xing 2009; Kim and Xing 2010), I wanted to investigate how structured association mapping algorithms performed under a more complex simulated model. I compared the structured association mapping method GFlasso to five other association mapping approaches: the Wilcoxon Sum-Rank test with FDR correction (Zhu et al. 2008), PLINK's implementation of the Wald test, forward regression, lasso regression, and the Screen and Clean method (Wu et al. 2010). Because the ground truth is known in simulation study, I can compare the false discovery and true positive rates (FDR and TPR) of each method.

2.3.1 Simulating the traits

For this study, I created four different datasets. I will describe each dataset in detail, and the underlying assumptions behind the datasets. In this subsection I define a *cis* association as a SNP that affects only one gene. I define a *trans* association as a SNP that affects more than one gene.

Dataset 1: The simple dataset.

The first dataset created is a small, simple dataset. I created a SNP matrix with N=100 individuals. Each of these individuals had J =1000 markers, simulated as described later in this subsection. The resolution (described later on) for this simulated-cross was set to R=50000 to allow for some LD even though the marker selection was sparse. There were K=500 traits.

A $\boldsymbol{B}$ matrix representing SNP-trait associations was constructed with a *cis* signal for 400 traits. Each *cis*-signal was distributed B_{jk} ~ UNIF(0.4,0.8), to allow for both weak and strong signals. Additionally, I added four *trans* (single-SNP to multi-trait) signals to **B**. The first signal affected 100 traits with a signal B_{jk} ~ UNIF(0.35, 0.85). The second signal affected 50 traits from the first set with a signal B_{jk} ~ UNIF(0.5, 1). The third and fourth sets affected 45 and 55 traits with signals B_{jk} ~ UNIF(0.4, 0.9) and B_{jk} ~ (0.3, 0.8), respectively. Once the $\boldsymbol{B}$ matrix was defined, the trait dataset was created as $Y = X\mathbf{B} + e \sim N(0, .25)$. This gives us a simple problem where the correlation in the dataset is only due to the *trans*-acting SNPs.

Dataset 2: Network structure due to hidden variables and trans-acting factors

The second dataset that was generated was a much larger dataset. Again, I used N=100 individuals. However, this time I set J=5000 markers and K=5000 traits. I simulated the SNP markers as before, using R=20000.

In the ***B*** matrix, I set a *cis* association for 1290 traits. This signal was strong: $B_{jk} \sim$ UNIF(0.5,1). I then defined 19 *trans* acting signals to modules of size 9 up to 300 traits. These signals were distributed $B_{jk} \sim$ UNIF(0.7, 1.2). Some of the SNPs were associated with more than one trait, while most did not. The 16 signals for the *trans*-acting SNPs were distributed $B_{jk} \sim$ UNIF(0, 10). The 19 trans-acting SNPs created 16 correlated groups (three groups had overlapping causal SNPs).

As an additional influence to the correlation structure of these groups, I added 16 hidden variables to generate correlation structure that was not created by the *trans*-acting SNPs. Each individual was affected by the hidden variable with a probability $P \sim$ UNIF(0.75, 1) (each individual was affected slightly differently by each hidden variable). Hidden variables affected each of the groups of traits that had been created by the *trans* SNPs. Additionally, some traits that were not in any of the groups from the transacting SNPs were selected to be influenced by these hidden variables.

The traits for this dataset were then generated according to the model: $Y = X_{\mathrm{h}}\beta_{\mathrm{h}} + X\mathbf{B} + e \sim$ N(0, 0.25), where β_{h} represents the hidden variables.

In Figure 2.1, a K x K plot of the correlation values between these traits is shown. This plot helps to illustrate the structure in the traits from this simulation. For example, the first 300 traits are strongly correlated as they are all affected by the same SNP. The first 500 traits are all affected by the same hidden variable. Thus, correlation between traits is caused both by the genomics as well as unknown hidden factors.

Dataset #3: Network structure from transacting SNPs only

For dataset #3, I used the same ***B*** matrix as in dataset 2, but generated the traits from the model $Y = X\mathbf{B}+$ e ~ N(0, 0.25), without the hidden variables.

Dataset #4: Network structure from hidden variables only

For dataset #4 I used the same β_h as in dataset 2, but changed **B** to remove all trans-acting signals, leaving only the *cis* signals.

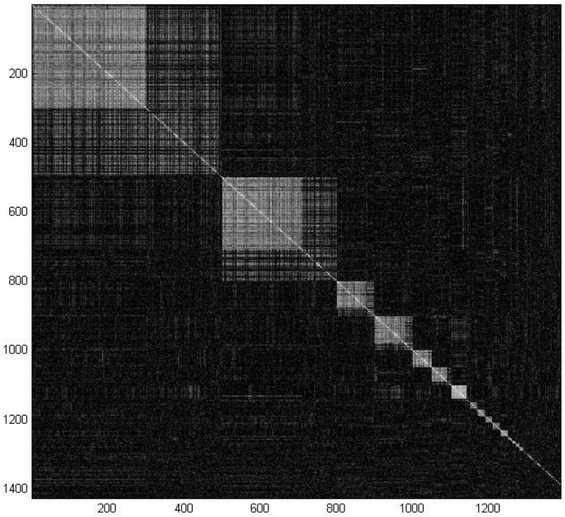


Figure 2.1: The correlation structure of the traits in dataset #2. The correlation is caused by both trans-acting factors as well as by hidden variables.

2.3.2 Simulating the chromosome SNP matrix

Modeled after yeast, the simulation model had 16 chromosomes, all about the same size as their yeast chromosomal counterpart (chromosome 16 is mtDNA). It was assumed that the *N* individuals in the simulation come from a genetic cross from two yeast parent strains. This cross would be similar to the data generated in a previously reported cross between a laboratory and a vineyard strain of yeast (Brem et al. 2002).

To generate the data, I first sample *J* SNPs from the entire genome. This is done through random sampling with each base in the genome being equally likely to be selected. Once *J* SNPs were selected, I went through each chromosome for each individual. The first SNP on each chromosome for each individual was assigned a parent with a 50% probability. I continued across the chromosome to consider each SNP in turn. I assume that a cross-over event (the parent for the new SNP may be different than the parent for the proceeding SNP) happened between each SNP with a Probability P ~ EXP(R), where the R is the resolution number, which was predefined in Subsection 2.3.1. If a cross-over event was found to have occurred, I select between the two parents for the SNP with an equal probability, keeping the same parent as the previous SNP otherwise. Thus, SNPs close together will have a high probability of originating from the same parent, while those farther apart will often come from different parents. This will create a change in parent about once every R bases. This simulation allows for some linkage disequilibrium in the data, while creating a mosaic of parent source across each chromosome.

2.3.3 Methods

Lasso

To run the lasso to find associations for this dataset, I used the glmnet package for R (Friedman et al. 2010). For a broad range of λ, I ran glmnet with all SNPs trait by trait. I combined the nonzero values from this step to recreate the full **B** matrix for each value of λ. Each nonzero value in the matrix was reassigned by calculating the least squares solution trait-by-trait using only the nonzero β_{jk} values. After this first search, I narrowed down my search for λ by selecting the best value according to cross-validation (where 10% were held out from the beginning for testing) and performed a similar search on a finer scale to find an optimal λ.

Plink (Wald test)

PLINK uses the Wald test to determine whether a proposed value of β is significantly different than 0. It compares the Wald statistic (βhat divided by the estimated variance of βhat) to a *t*-distribution. If the *p*-value is significant then we conclude that the predictor variable should be included in the model. I ran PLINK with the default settings for quantitative trait mapping. All SNP/trait pairs that had a p-value of less than .05 were considered to be an association.

Wilcoxon-Sum Rank Test

The Wilcoxon-Sum Rank Test, also called the Mann-Whitney Wilcoxon, Mann-Whitney U or Wilcoxon-Mann-Whitney test, is a non-parametric test. To perform this test, I iterate through all the SNP-trait pairs. For each SNP-trait pair, the traits are ranked and divided into two partitions based on the parent at the SNP in question. The rankings of the two groups are then summed. A U statistic is calculated: $U_1 = R_1 - n_1(n_1+1)/2$, where R_1 is the sum of all ranks in the first group and n_1 is the number of individuals in group 1. For large samples, U will be approximately normally distributed. If this is true, we can find $\mu = n_1n_2/2$ and $\sigma^2 = n_1n_2(n_1+n_2+1)/12$ to calculate a z-statistic to find a *p*-value. If the value is significant, we conclude that the trait has a significantly different distribution based on the group partition.

I followed the model of Zhu et al. (2008) to perform the Wilcoxon-Sum Rank test (WSR) with false-discovery correction to determine what *p*-values would be considered significant. I calculated the Sum Rank test for each SNP/trait pair. Next, I permuted the trait values 10 times for each trait to create a null distribution for the *p*-values across the entire *J* x *K* association matrix. From this null distribution, I define a cutoff that sets the expected value of the FDR to be less than .05 and use this cutoff to define significance. For example, in dataset 1, all *p*-values less than 1e-4 were considered to be significant.

Running the Forward-selection method

The forward-selection method is a greedy regression method for model selection. For each trait, the algorithm looks for the SNP that best explains the response variable and adds it to the additive regression model one by one. I use cross validation to find the "optimal" model among the first 20 models found by forward selection trait-by-trait.

Screen and Clean method

To summarize the Screen and Clean method, the data is divided into two partitions in some way. The Lasso is run on the first partition, and a statistical test is run on the second half of the data using only the SNPs selected for the in the Lasso. The Lasso, which is known to be over-inclusive, therefore will screen for SNPs, and then the Wald test will clean the data to remove over-inclusive predictors (Wu et al. 2010).

I split up the data into 3 equal partitions. I used 2 parts for the screen method and the last part for the clean method. I used glmnet for the screen step and PLINK (Wald test) for the clean step. I used the same basic procedure for the screen step as I used for the glmnet procedure, choosing the values based on cross validation.

Running the GFlasso method

To create a network for the GFlasso, I followed the procedure outlined by Zhang and Horvath to create a weighted network (Zhang and Horvath 2005). This procedure is outlined in detail in Chapter 4. I also ran GFlasso as it is currently automated in Auto-SAM, as described in Chapter 3.

Determining LD blocks

Because the simulation model incorporated linkage disequilibrium (LD), these methods found associations to SNPs in LD with the true, causal SNP. In an attempt to combat this, I assigned each SNP to an LD block. I used a form of SNP-tagging in order to do this.

The first step in the LD analysis is to calculate Δ^2 for each SNP pair. Δ^2 is a measure of correlation between the two SNPs. $\Delta^2 = D^2_{AB} / (p_A(1-p_A)p_B(1-p_B)$ where $D_{AB} = p_{AB} - p_Ap_{B.}$ and A and B represent one of the parents at the two loci in question. I set all values in the Δ^2 matrix to zero that are less than 0.8 (a standard cut-off). I then find the highest value in the matrix and start an LD block with those two loci. All neighboring loci connected to either of these two founders are assigned to that block. I continue until all loci have been assigned to a block. This gives me a way of assigning each SNP to a block with other correlated SNPs.

2.3.4 Simulation Results

After running each of the methods on the data as described in the previous section, I calculated the True Positive Rate (TPR), the FDR, the "*cis*-TPR" and the "*trans*-TPR" for each method.

Results Interpretation strategy

Each method, when run as described, returns a binary J x K matrix representing associations between the traits and SNPs.

My strategy for calculating the TPR and FDR is as follows. I consider each discovered SNP-trait association. If any of the other SNPs in the true association matrix within the SNP's LD block are associated with the trait, it is considered to be a true positive. However, if a SNP is associated with the trait in the predicted matrix, but is not in the LD block of any true SNP-trait associations, it is a false positive. The TPR over the entire matrix is defined as the number of true positives found divided by the number of ground-truth positives. The FDR is defined as the number of false positives found divided by the total number of associations found in the prediction matrix. I also consider the *cis*-TPR, which is the TPR of all SNP-trait associations in *cis*, and the *trans*-TPR, which is the TPR of all SNP-trait associations in *trans*.

Result Tables

For each simulated dataset, I report the TPR, FDR, *cis*-TPR, and *trans*-TPR. These results are reported in Tables 2.1-2.4.

Table 2.1: Results for Trait Set #1: The Simple Dataset

	TPR	*FDR*	*Cis TPR*	*Trans TPR*
GFlasso	0.8869681	0.01801802	0.9596977	0.8056338
PLINK	0.8204787	0.02912281	1.0000000	0.6197183
Wilcoxon-Sum Rank	0.8058511	0.03399725	1.0000000	0.5887324
Screen & Clean	0.712766	0.02489627	0.9899244	0.5667606
Lasso	0.9255319	0.04878049	1.0000000	0.8422535
Forward Selection	0.9512770	0.4851172	0.9974811	0.9014085

Table 2.2: Results for Trait Set #2: Network structure due to hidden variables and trans-factors

	TPR	*FDR*	*Cis TPR*	*Trans TPR*
GFlasso	0.5352176	0.009141846	0.2371901	0.8891070
PLINK	0.6868551	0.01395925	0.5892562	0.8027478
Wilcoxon-Sum Rank	0.6195603	0.007924977	0.5115702	0.7477920
Screen & Clean	0.6581427	0.1530516	0.5553719	0.7801766
Lasso	0.9820547	0.8864157	0.9743802	0.9911678
Forward Selection	0.8896366	0.8672514	0.8545455	0.9313052

Table 2.3: Results for Trait Set #3: Network structure due to transfactors only

	TPR	*FDR*	*Cis TPR*	*Trans TPR*
GFlasso	0.8999551	0.03201787	0.8330579	0.9793916
PLINK	0.8721400	0.02480355	0.8049587	0.9519136
Wilcoxon-Sum Rank	0.7860027	0.01616926	0.6925620	0.8969578
Screen & Clean	0.6536563	0.008408408	0.5074380	0.8272816
Lasso	0.9587259	0.09168185	0.92399669	1.0000000
Forward Selection	0.9681472	0.2522301	0.9479339	0.9921492

Table 2.4: Results for Trait Set #4: Network structure from hidden variables

	TPR	*FDR*	*Cis TPR*	*Trans TPR*
GFlasso	0.5868313	0.1159875	0.5868313	NaN
PLINK	0.7802469	0.01390984	0.7802469	NaN
Wilcoxon-Sum Rank	0.7209877	0.007888805	0.7209877	NaN
Screen & Clean	0.6658436	0.2531153	0.6658436	NaN
Lasso	0.9802469	0.9439060	0.9802469	NaN
Forward Selection	0.8724280	0.8211518	0.8724280	NaN

Observations

There are a few points worth mentioning to aid the reader when perusing these results. First, the results reported for Tables 2.2-2.4 include the first 1290 traits only (excluding the rest of the 5000). This decision was made because only the first 1290 traits have network structure. The GFlasso behaves in the same way as the Lasso when analyzing the remaining traits, and so to provide an accurate comparison, only the traits with network structure are included.

For each method, the reader should notice that there is a trade-off between the TPR and the FDR. The PLINK and Sum-Rank methods both identify many true associations with a small false-discovery rate. The Lasso and forward selection methods are able to identify almost all true associations, but this comes at the cost of (usually) high false-discovery rates. For all of these datasets, the GFlasso has a false-discovery rate comparable to, if not lower, than the PLINK and Sum-Rank methods. At the same time, the GFlasso has a *trans*-TPR statistic often comparable to the other regression methods, and always higher than the PLINK and Sum-Rank methods. However, the GFlasso does poorly in its *cis*-TPR statistic for the datasets with the hidden variables. Meanwhile, Screen and Clean does seem to control the Type I error rate as expected, but does so at a loss of some power.

GFlasso does well on Datasets 1, 2, and 3, but does not do well when the underlying assumptions of the data are not correct. When there are no *trans*-associations, the other methods outperform GFlasso in terms of TPR, although GFlasso maintains a comparable FDR.

2.3.5 Conclusions from the simulation study

In a model simulated with the assumption that mutations drive genetic co-expression, GFlasso outperforms other commonly-used methods in terms of TPR and FDR. In terms of recovering associations from one SNP to multiple-correlated traits, GFlasso had a higher TPR compared to the Wald and Wilcoxon-Sum Rank test, while maintaining a low FDR. While the model-selection methods like the lasso and forward selection have a nearly perfect TPR, this was obtained through the over-inclusion of many false signals (resulting in a high FDR), consistent with current analytic and theoretical knowledge (Wasserman and Roeder 2009; Devlin et al. 2003).

The results from this analysis lead to three natural conclusions. First, given that the model assumptions are true for biological systems, we can expect that structured association mapping methods will increase the TPR of association studies, leading to the discovery of signals that would not be discovered otherwise. Secondly, by taking into account the structure of the data in our analysis, structured association mapping algorithms are able to limit the number of false discoveries. Finally, I note that because the GFlasso penalizes based on the network structure, it is not as adept at finding the single-SNP single-gene associations; thus GFlasso's strength is in finding the associations from one SNP to multiple-correlated phenotypes.

2.4 gGFlasso, a new algorithm for three-way association analysis

In the post-GWAS era, it has been suggested that the GWAS data could be more insightful when integrated with other data types. Given the potential of integrating transcriptome data into GWAS, I worked with Dr. Junming Yin to consider the problem of combining GWAS and gene expression data to find genome-transcriptome-phenome associations in a structured association mapping framework. We proposed a novel two-step strategy that employs structured association mapping to find associations from genome to phenome by leveraging gene expression data and its network structure. Our two-step framework involves first finding genome-transcriptome associations using GFlasso. In the second step, we find transcriptome-phenome associations using a newly development method (gGFlasso), presented in this section.

To date, the general strategy has been to integrate gene expression data into a GWAS study *after* the main analysis and from individuals *unrelated* to the original study (Schadt et al. 2008; McCarthy and Hirschorn 2008). However, in the increasingly common scenario where expression, phenotype, and genomic data are available from the same cohort, expression data can be incorporated directly into the primary analysis. In this case, all the data guide the discovery of genome-transcriptome-phenome associations. This unified three-way association framework has the potential to reveal the functional relationships between associated genomic variations and physical phenotypes, via intermediate phenotypes. This has a direct impact on personalized medicine as different patients may have different regulatory mechanisms by which disease arises, but traditional SNP-trait association studies are insufficient to uncover the hidden mechanisms. Despite the promise of unified three-way association analysis, work done on this problem is limited (Chen et al. 2008; Emilsson et al. 2008).

To find associations between genes and phenotypic traits, we developed a new method by extending GFlasso, called graph-graph-guided fused lasso (gGFlasso), which incorporates the correlation structure of genes G_G as well as the dependency graph of traits G_T. In this setting, expression levels are treated as predictors and traits are regarded as responses. The GFlasso framework only uses the dependency structure on responses and assumes that variations of correlated responses (traits) are likely to be explained by a common set of predictors (genes). However, predictors can also be correlated, such as co-expressed genes, and it seems natural to also exploit the predictor's dependency graph. Given G_G, gGFlasso assumes that correlated (connected) genes tend to influence the same subsets of traits. This assumption is encoded as an additional fusion penalty in the linear regression model:

$$\boldsymbol{B_2} = \underset{\boldsymbol{B} \in \mathbb{R}^{J \times K}}{argmin} \|\boldsymbol{Z} - \boldsymbol{YB}\|_F^2 + \lambda \sum_j \sum_k |B_{jk}| + \gamma_1 \sum_{\{u,v\} \in E_G} \sum_k |B_{uk} - sign(\rho_{uv}) B_{vk}|$$
$$+ \gamma_2 \sum_{\{m,l\} \in E_T} \sum_j |B_{jm} - sign(\rho_{ml}) B_{jl}|. \quad (2.2)$$

As in Eq. 2.1, λ, γ_1, and γ_2 are regularization parameters and B_{jk} is the association strength of jth gene with kth trait. Note that there are two graph-guided fused lasso penalty terms in Eq. 2.2: if two genes u and v are connected in G_G, the fusion penalty encourages their influence on each trait to be similar; if two traits m and l are connected by an edge in E_T, the association strength B_{jm} and B_{jl} for each gene j is encouraged to have the same absolute value. This joint framework that accounts for the information in the correlation structure of predictors (gene expressions) and responses (traits) has the potential to increase the sensitivity and specificity of association studies. Combined with the L_1 lasso penalty, the estimated coefficient matrix $\boldsymbol{B_2}$ has a large fraction of zero entries.

2.4.1 gGFlasso implementation notes

I implemented gGFlasso and automated the GFlasso-gGFlasso strategy in GenAMap. I use a coordinate descent approach and the "η-trick" to optimize Eq. 2.2 (Rakotomamonjy et al. 2008) By introducing the auxiliary variables the above problem is equivalent to the following optimization problem:

$$\min_{\boldsymbol{B_2}, \eta_{jk}, \eta_{kuv}, \eta_{jml}} \|\boldsymbol{Y} - \boldsymbol{XB_2}\|_F^2 + \lambda \sum_j \sum_k \frac{\beta_{jk}^2}{\eta_{jk}} + \gamma_1 \sum_{\{u,v\} \in E_G} f(r_{uv})^2 \sum_k \frac{(\beta_{uk} - \mathrm{sign}(r_{uv}) \beta_{vk})^2}{\eta_{kuv}}$$
$$+ \gamma_2 \sum_{\{m,l\} \in E_T} f(r_{ml})^2 \sum_j \frac{(\beta_{jm} - \mathrm{sign}(r_{ml}) \beta_{jl})^2}{\eta_{jml}}$$
$$\mathrm{s.t.} \sum_j \sum_k \eta_{jk} = 1, \sum_{\{u,v\} \in E_G} \sum_k \eta_{kuv} = 1, \sum_{\{m,l\} \in E_T} \sum_j \eta_{jml} = 1 \quad (2.3)$$

To solve the optimization, I apply the coordinate descent algorithm as follows:

$$\beta_{jk} = \frac{\sum_n x_{nj}(y_{nk} - \sum_{j' \neq j} x_{nj'}\beta_{j'k}) + \gamma_1 \sum_{u:\{u,j\}\in E_G} \frac{f(r_{uj})^2 \operatorname{sign}(r_{uj})\beta_{uk}}{\eta_{kuj}} + \gamma_2 \sum_{m:\{m,k\}\in E_T} \frac{f(r_{mk})^2 \operatorname{sign}(r_{mk})\beta_{jm}}{\eta_{jmk}}}{\sum_n x_{nj}^2 + \frac{\lambda}{\eta_{jk}} + \gamma_1 \sum_{u:\{u,j\}\in E_G} \frac{f(r_{uj})^2}{\eta_{kuj}} + \gamma_2 \sum_{m:\{m,k\}\in E_T} \frac{f(r_{mk})^2}{\eta_{jmk}}}$$

$$\eta_{jk} = \frac{|\beta_{jk}|}{\sum_{j'k'} |\beta_{j'k'}|}$$

$$\eta_{kuv} = \frac{f(r_{uv})|\beta_{uk} - \operatorname{sign}(r_{uv})\beta_{vk}|}{\sum_{\{u',v'\}\in E_G} \sum_{k'} f(r_{u'v'})|\beta_{u'k'} - \operatorname{sign}(r_{u'v'})\beta_{v'k'}|}$$

$$\eta_{jml} = \frac{f(r_{ml})|\beta_{jm} - \operatorname{sign}(r_{ml})\beta_{jl}|}{\sum_{\{m',l'\}\in E_T} \sum_{j'} f(r_{m'l'})|\beta_{j'm'} - \operatorname{sign}(r_{m'l'})\beta_{j'l'}|} \qquad (2.4)$$

I automated gGFlasso in GenAMap with a linear-search strategy to find values for the regularization parameters λ, γ_1, and γ_2. My C++ implementation of gGFlasso is also available from sailing.cs.cmu.edu/genamap/threeway.html. For simulated data with 100 SNPs, 500 genes, and 20 traits, the full GFlasso-gGFlasso runs in less than one day. Using GenAMap to run the GFlasso-gGFlasso for 2500 SNPs with associations to 2000 genes and 200 traits would return results in less than one week.

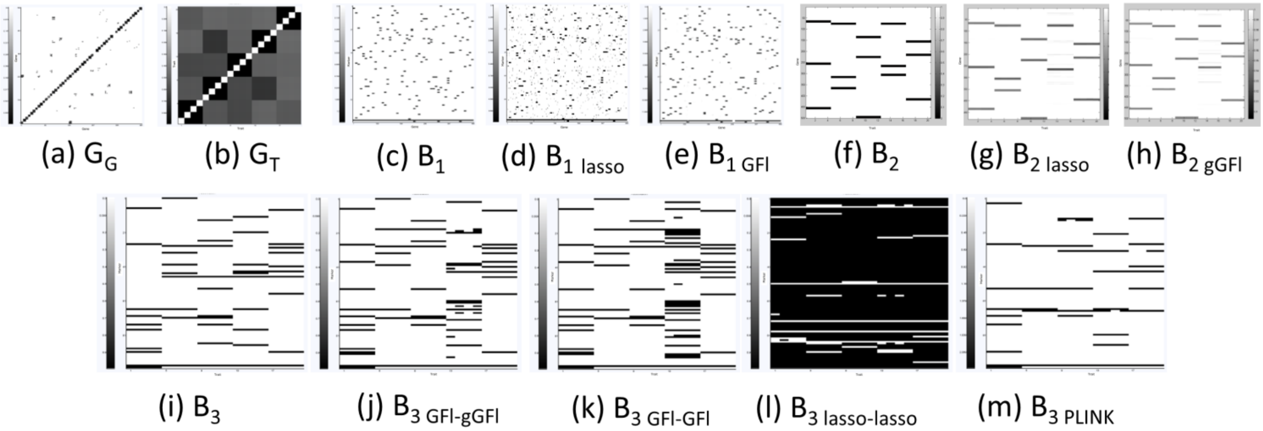
2.5 Validation of gGFlasso through simulation

In collaboration with Dr. Junming Yin, I performed a simulation experiment to validate our GFlasso-gGFlasso strategy for three-way association analysis, comparing the results with a GFlasso-GFlasso strategy, a lasso-lasso strategy, and PLINK (Purcell et al. 2007). We randomly selected N = 250 individuals and P = 100 SNPs from a mice dataset (Johannesson et al. 2009) to create a genotype matrix $\boldsymbol{X}$. The number of genes J was set to 500; we assumed 50 correlated gene groups of equal size. The true $P \times J$ association matrix $\boldsymbol{B_1}$ (representing SNP-gene associations) was created as follows: for each group, we randomly selected three causal SNPs; one (random) housekeeping SNP was assumed to be associated with all genes. Strengths of association for selected SNP-gene pairs were set to 1 (Figure 2.2c). Given $\boldsymbol{X}$ and the association matrix $\boldsymbol{B_1}$, the gene expression matrix $\boldsymbol{Y}$ was simulated as $\boldsymbol{Y} = \boldsymbol{XB_1} + \boldsymbol{\varepsilon_1}$, where $\boldsymbol{\varepsilon_1}$ is an $N \times J$ matrix with entries independently generated from $N(0, \sigma_1^2)$. The gene correlation matrix, simulated with noise level $\sigma_1^2 = 1$ and thresholded at 0.5, reveals the inherent structure of the genes (Figure 2.2a). We set $K = 20$ traits. To generate the trait matrix $\boldsymbol{Z}$, we first created a $J \times K$ association matrix $\boldsymbol{B_2}$ (representing gene-trait associations) similar to the procedure for obtaining $\boldsymbol{B_1}$: we assumed five groups of correlated traits of equal size each with three causal gene groups (Figure 2.2f). Then, $\boldsymbol{Z} = \boldsymbol{YB_2} + \boldsymbol{\varepsilon_2}$, where entries of $\boldsymbol{\varepsilon_2}$ were independently generated from $N(0, \sigma_2^2)$. We also calculated the true SNP-trait association matrix, $\boldsymbol{B_3}$ (representing SNP-trait associations, Figure 2.2i), by assuming that if a SNP is associated with a gene and that gene influences a trait, then there is an association between the SNP and the trait.

In Table 2.5, I present a summary of performance results evaluated by the true positive rate (TPR) and false positive rate (FPR). As the table and Figure 2.2e show, in a low noise setting, GFlasso is able to recover SNP-gene pairs with true association with a tiny FPR. We compared the performance of GFlasso and gGFLasso in recovering $\boldsymbol{B_2}$. Although both methods successfully recover true gene-trait pairs, GFlasso tends to produce a number of false positive pairs with weak associations (Figure 2.2h) and hence has a higher FPR. Both GFlasso-GFlasso and GFlasso-gGFlasso have a much lower FPR than the lasso-lasso strategy in terms of recovering $\boldsymbol{B_3}$ (Figure 2.2l). We were interested in whether the methods that do not incorporate the gene expression data would similarly recover $\boldsymbol{B_3}$, and so we ran the default association algorithm in PLINK to find SNP-trait associations (Figure 2.2m). We found that although there is a low FPR, the power of detecting true associations decreases significantly.

Table 2.5. True positive rates (TPR) and false positive rates (FPR) of different methods in recovering the association matrices at different noise levels. Compare to Figure 2.2.

	$\sigma_1^2 = \sigma_2^2 = 1/4$		$\sigma_1^2 = \sigma_2^2 = 1$		$\sigma_1^2 = 4, \sigma_2^2 = 16$	
	TPR	FPR	TPR	FPR	TPR	FPR
$\boldsymbol{B}_1$ by GFLasso (Fig. 2.2e)	0.9454	0.0060	0.8965	0.0057	0.7884	0.0092
$\boldsymbol{B}_1$ by Lasso (Fig. 2.2d)	0.9758	0.7632	0.9535	0.0763	0.9081	0.7528
$\boldsymbol{B}_2$ by gGFLasso (Fig. 2.2g)	1.0000	0.0000	0.9333	0.0016	0.7067	0.0205
$\boldsymbol{B}_2$ by GFLasso (Fig. 2.2h)	0.9800	0.0004	0.9233	0.0039	0.7000	0.0207
$\boldsymbol{B}_3$ by GFlasso-gGFlasso (Fig. 2.2j)	0.9200	0.0266	0.8600	0.0305	0.8400	0.2450
$\boldsymbol{B}_3$ by GFlasso-GFlasso (Fig. 2.2k)	0.9200	0.0266	0.8600	0.0615	0.8400	0.2500
$\boldsymbol{B}_3$ by Lasso-Lasso (Fig. 2.2l)	1.0000	0.8803	1.0000	0.9030	1.0000	0.8559
$\boldsymbol{B}_3$ by PLINK (Fig. 2.2m)	0.6300	0.0150	0.5357	0.0234	0.5000	0.0294



(a) G_G (b) G_T (c) B_1 (d) $B_{1\ \text{lasso}}$ (e) $B_{1\ \text{GFl}}$ (f) B_2 (g) $B_{2\ \text{lasso}}$ (h) $B_{2\ \text{gGFl}}$

(i) B_3 (j) $B_{3\ \text{GFl-gGFl}}$ (k) $B_{3\ \text{GFl-GFl}}$ (l) $B_{3\ \text{lasso-lasso}}$ (m) $B_{3\ \text{PLINK}}$

Figure 2.2: Results of a simulation study on a dataset simulated with $\sigma_1^2 = \sigma_2^2 = 1$. Dark pixels indicate large values. (a) Correlation matrix of genes thresholded at 0.5 (b) Correlation matrix of traits thresholded at 0.5 (c) True $\boldsymbol{B}_1$ matrix (d) Recovered $\boldsymbol{B}_1$ matrix by lasso (e) Recovered $\boldsymbol{B}_1$ matrix by GFlasso. (f) True $\boldsymbol{B}_2$ matrix (g) Recovered $\boldsymbol{B}_2$ matrix by gGFlasso. (h) Recovered $\boldsymbol{B}_2$ matrix by GFlasso (i) True SNP-trait association matrix $\boldsymbol{B}_3$ (j) Recovered $\boldsymbol{B}_3$ by GFlasso-gGFlasso strategy. (k) Recovered $\boldsymbol{B}_3$ by GFlasso-GFlasso strategy (l) Recovered $\boldsymbol{B}_3$ by lasso-lasso (m) Recovered $\boldsymbol{B}_3$ by plink, values plotted are –log10(p-value).

3. GenAMap

The advances in structured association mapping hold significant potential to facilitate discovery in association studies. However, while initial studies have suggested that structured association mapping leads to increased insight and greater statistical power in association study (Kim and Xing 2009; Puniyani et al. 2010), two key obstacles hinder these advances from being widely accepted in practice. The first obstacle is the expertise required to run structured association mapping algorithms. Structured association mapping algorithms are generally made available as crude, command-line implementations (if they are made available at all). Thus, in order for a geneticist to use a structured association mapping algorithm in a GWAS analysis, they must download a rough MATLAB implementation of the algorithm and customize the code to fit the specific study. Furthermore, many of these algorithms have been scaled and tested only on small, simulated datasets, and while they can be potentially extended to larger datasets, the way to do so is not obvious to those not specialized in machine learning or familiar with the mathematical details. Due to the amount of specialization required to run the algorithms, the algorithms stay on the shelf, despite their potential to aid in genetics analysis.

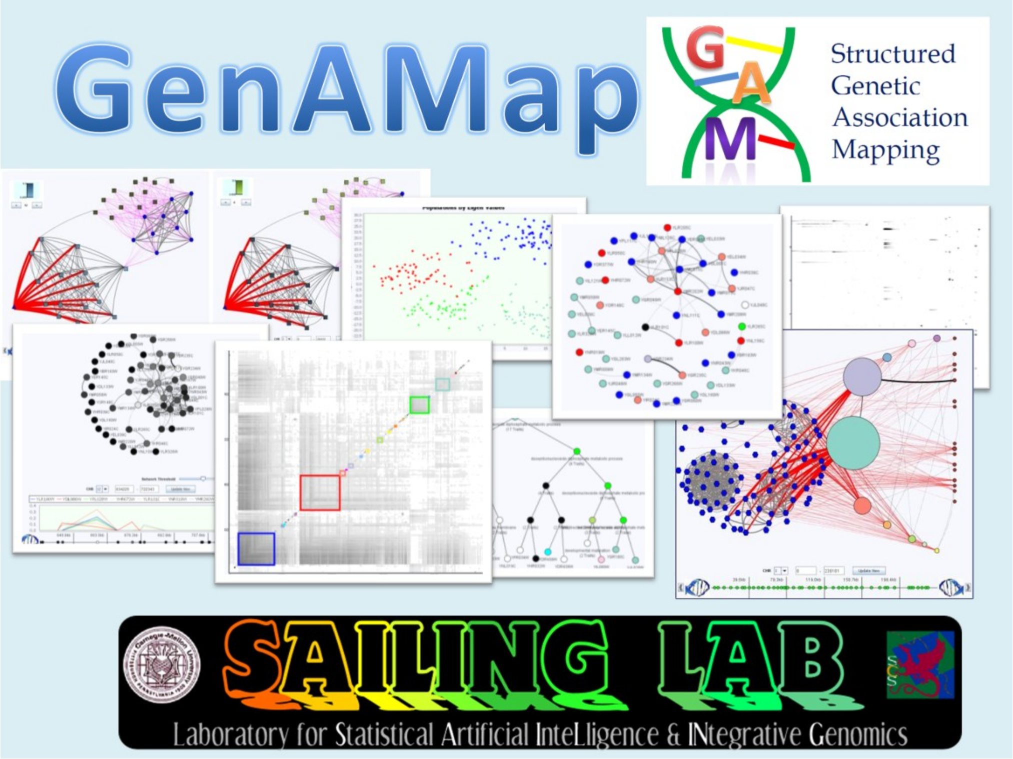

GenAMap
G
A
M
Structured
Genetic
Association
Mapping
SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

Figure 3.1: GenAMap. GenAMap is a visual analytics system for structured association mapping. It incorporates visualizations in a novel way to lead biologists to relevant SNPs and their associated traits. GenAMap enables biologists to explore the structure of the genome and trait data while exploring association strengths.

The second obstacle is the exploration of the results after the algorithms complete. In a structured association analysis, the analyst is no longer considering a small list of SNP-trait associations, but rather a sea of data with complex structure. Current visualization strategies for exploring structure are not customizable to association studies (Shannon et al. 2003), and association visualization strategies for GWAS are limited in the types of analyses they can perform; most can only look at one trait or gene expression level at a time (Pruim et al. 2010; Ge et al. 2008; Mueller et al. 2005). Due to the data complexity, results from these algorithms become a sea of data that can be challenging to explore. These results are generally output as a raw matrix of values representing associations between the genome and the traits. For a simple organism like yeast, a dataset will generally include a few thousand SNPs and up to six thousand gene measurements. For a human dataset, there might be hundreds of thousands of SNPs and over ten thousand gene expression measurements. Therefore, despite the improvements that we have seen in performing structured association mapping studies with thousands of SNPs and thousands of traits, in order to adequately explore the data, researchers still have to rely on in-house scripts, command line tools, and customized software (Buckingham 2008). The reliance on these types of tools limits the number of individuals who can perform these analyses to only the very specialized, and limits the insight that can be gleaned from exploring the structure of the data itself.

As one of the main contributions of this dissertation, I have developed a new visual analytics system called GenAMap. My work with GenAMap has been motivated by the promise of structured association mapping strategies and the need for visualizations to explore the results from these machine learning algorithms. By combining the strength of cutting-edge machine learning technology for structured association mapping with novel visualizations built for the exploration of structured association data, GenAMap is a comprehensive system for structured association mapping to 1) automate the execution of structured association mapping algorithms and 2) provide new visualizations specifically designed to aid in the exploration of association mapping results.

GenAMap is specifically designed to give geneticists an overview of the results then lead them to specific genome-gene interactions. In my experience, in a structured association study, genetics analysts need to get an overall picture of the patterns of association in the data, and then they need to focus their attention on specific, important signals in the data. This immediately suggests a visualization strategy following Shneiderman's well-known mantra: overview first, zoom and filter, details on demand (Shneiderman 1996). As I will show, this mantra has provided an excellent strategy for the development of the visualizations for using data structures to guide discovery in association studies. Once a geneticist has used GenAMap to identify a significant interaction, he can explore the interaction in the tool and link directly to external links to biological databases such as UniProt (The UniProt Consortium 2011) and dbSNP (Sherry et al. 2001). GenAMap specifically aids in the exploration of association mapping results through the integration of multiple views so that an analyst can explore the structures of the genome, transcriptome, and phenome simultaneously when considering associations.

In this chapter, I will demonstrate the GenAMap visual analytics system. I will start by presenting the system overview, move to a discussion of the algorithmic implementation, and finally demonstrate some of the visualizations available in GenAMap.

3.1 System overview

In this section, I will begin by giving an overview of GenAMap by discussing a typical structured association mapping workflow using GenAMap as shown in Figure 3.2. This workflow includes several machine learning and visualization steps; GenAMap is intended to integrate all analytic and visualization tools necessary to complete a structured association analysis. In an association study, at least two types of data must be collected from the individuals in the study. Genotype data are collected as SNPs for hundreds of thousands to millions of SNPs, and trait data are collected as gene expression data from a microarray and/or as clinical trait data collected in the office.

A typical association study might go something like this: after quality control and preprocessing steps, the data are formatted in order to run machine learning algorithms on the data to generate structure. Investigators explore the structure using tools like MATLAB or R (R Development Core Team 2011) to determine what structured association analyses were appropriate for the data and to initially identify interesting patterns. They then reformat the data to run machine learning algorithms to find the structured association results. Next, they write their own scripts or use MATLAB and R to explore the data, observe interesting patterns, and find specific, interesting associations. The analysts investigate the details of these associations using online databases and a handful of other tools in order to understand the interactions and associations. This analysis would lead to conclusions about the data and to other hypotheses to be tested in future studies. GenAMap's integrated system is designed to guide researchers through many of these steps (C through I as shown in Figure 3.2).

When an analyst uses GenAMap, he/she avoids the awkward integration of multiple tools to generate structure and then analyze the associations by structure. GenAMap has algorithms that automatically run structure-generating algorithms, and GenAMap provides visualization to help the analyst explore the data to determine which structured association algorithms will be appropriate. For example, if the analyst notices strong population stratification in the data, he/she

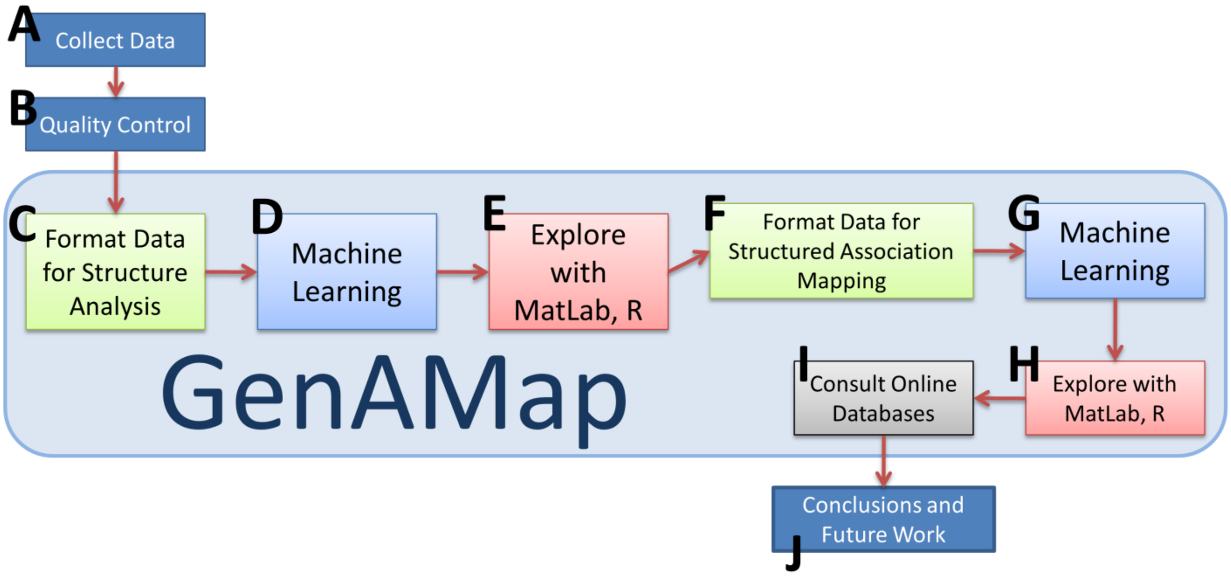


Figure 3.2: GenAMap is integrated into the workflow of a structured association mapping analysis.

will want to perform a population analysis to find associations. If the analyst notices that the trait network consists of many highly connected clusters, he/she will want to use an algorithm such as GFlasso to take advantage of this information. Once the analyst has explored the data and determined the appropriate structured analysis for the dataset, he/she is ready to run the actual association algorithm.

GenAMap automatically runs all structured association mapping algorithms on an external cluster using parallelization. Once the results from the algorithms are available, the analyst has a number of tools to explore the associations in the data.

To provide tools to explore the results of association analysis, GenAMap supports Shneiderman's mantra: *overview first*, *zoom and filter*, and *details on demand* (Shneiderman, 1996). First, GenAMap allows the analyst to get an overview of the results through tools such as a heat map showing patterns of SNP-gene association. GenAMap provides zoom and filter options to drill down to the interesting genes and SNPs in structure or association results. GenAMap also offers details on demand through links to online resources and information so analysts can find the details of associations and substructures in the data.

3.1.1 Overview of the software

In Figure 3.3, I present a high-level view of the system architecture for GenAMap. GenAMap is a GUI that runs on the analyst's desktop machine. GenAMap communicates with back-end servers to run structure-generating and structured association mapping algorithms through a software system called Auto-SAM. There are two databases in Auto-SAM. The *data database* holds the data: SNP data, gene expression data, association mapping results, etc. The *jobs database* stores information about each job request from an analyst. Once an analyst has uploaded data into the data database, he can request to run a job through the jobs database. A

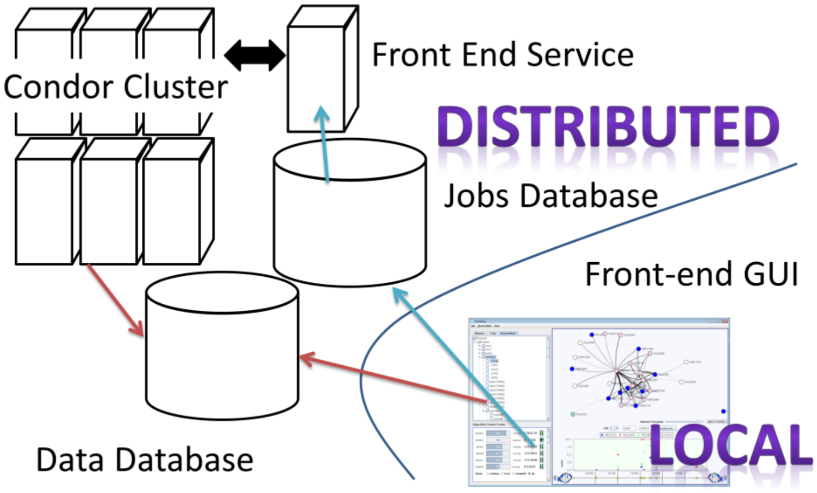


Figure 3.3: GenAMap system architecture. The GenAMap visual analytics system is controlled locally using a desktop application front end. The back-end system, Auto-SAM, runs on a remote server that has access to a condor cluster. This cluster is used to automate the parallel execution of structured association mapping algorithms.

service running on the distributed machine watches the database, and when it finds a new job, it spawns jobs on a cluster. The service monitors these jobs, and updates the jobs database so the front-end GUI can notify the analyst of each job's progress.

GenAMap itself is implemented in Java SE. To facilitate the rapid development of high-quality visualizations, I have integrated open-source visualization Java toolkits into GenAMap, including JUNG (Madahain et al. 2005), JHeatChart (Castle, 2009), and JFreeChart (Gilbert, 2010). GenAMap communicates with Auto-SAM through an Apache web-interface. Data storage is done via MySQL, and algorithms are run on an automatic processing system implemented in Java SE, C++, R (R Development Core Team 2011), and MATLAB. Algorithms are parallelized and run using Condor (Thain et al. 2005).

3.2 Auto-SAM

In this section, I present Auto-SAM as a new strategy and a user-oriented fully automated software system for the deployment of new machine learning algorithms with the purpose to make them accessible to geneticists. In contrast to the general strategy of a raw implementation on the web, I systematically develop each algorithm so it will automatically run in a distributed parallel-computing environment. Thus, little technical specialization is required for a genetics analyst to run the algorithms. I argue that the wide-spread acceptance of such an approach could potentially accelerate biological discovery by facilitating the incorporation of cutting-edge machine learning techniques. I will first describe the database implementation of Auto-SAM, followed by a discussion of how I have automated the execution of the algorithms. Finally, I will discuss the running times for the algorithms using two example datasets and present an example scheme that I use to automate the execution of a structured association mapping algorithm: GFlasso.

It might be argued that one common approach of deploying new algorithms via CRAN-R libraries (R Development Core Team, 2011) is similar to the strategy I propose here. (For examples see glasso (Friedman et al. 2007) or the bioconductor package (Gentleman et al. 2004). However, my approach differs in three significant ways: 1) by running algorithms on a distributed system with access to a cluster-computing system, Auto-SAM is able to handle much larger datasets and run algorithms in parallel; 2) through the use of a database, analysis results are made available to entire teams of analysts; 3) the integration of Auto-SAM with GenAMap provides state-of-the-art visual analytic tools that enable the analyst to explore and analyze the data and results, including links to external databases and integration with gene-ontology resources.

GenAMap runs a variety of algorithms through Auto-SAM including structure-generating algorithms, association algorithms, and structured association mapping algorithms, listed in Table 3.1. To generate structure, Auto-SAM provides algorithms to build networks and find population structure. GenAMap runs baseline association methods through Auto-SAM including PLINK's chi-square and Wald tests (Purcell et al. 2007). Most notably, GenAMap automates five structured association mapping algorithms: GFlasso, TreeLasso, AMLT, MPGL, and gGFlasso. Analysts can also load in their own structures and results into GenAMap, bypassing Auto-SAM and using GenAMap's visualizations to analyze different results.

3.2.1 Data management

All data that are used and generated by Auto-SAM are stored in the data database. I implemented the data database using MySQL. An analyst can use a front-end GUI like GenAMap to upload data into the database. Auto-SAM organizes all data by team, where each team is made up of one or more analysts that all have access to the same data. Each team's data are organized by project, and each project is made up of marker, trait, structure, and association data.

Storing the data in a remote database has several advantages. First, the data can be accessed from any computer with an internet connection, thus analysts only need a login and password to access their data. Second, the Auto-SAM data storage strategy allows for easy data sharing between all analysts on a team. Finally, because the data are stored in the database, it can be accessed by the analyst through a GUI and simultaneously accessed by the cluster as algorithms run.

Auto-SAM's data database allows for the by-need passing of algorithm results to the front-end GUI. For example, if an analyst runs an algorithm to generate network structure from gene expression data, the automated system will split the data into different runs, run it on the cluster, and generate results. Rather than pass these results back to the GUI to be stored on the local machine, the results are inserted into the database and accessed from the GUI by-need. As the GUI monitors the job, it will automatically update once the results are available to the analyst.

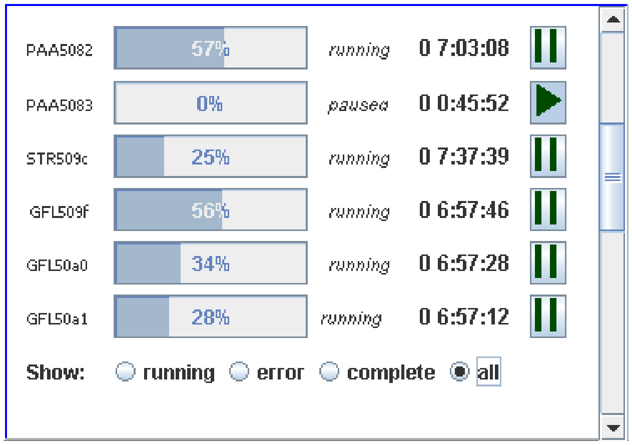
Auto-SAM's reliance on MySQL does have some disadvantages: as the tables fill up the speed of the database slows down and the deletion of datasets can be very slow. However, Auto-SAM can store datasets with up to 20,000 SNPs or traits, which is generally sufficient for most structured association mapping analyses. Additionally, Auto-SAM takes advantage of the fact that many results (such as gene-gene networks or SNP-gene associations) are sparse, and thus saves considerable space by only storing non-zero values. Auto-SAM is built so that it can easily be integrated with new, faster technologies, if needed.

3.2.2 Algorithm automation and parallelization

The Auto-SAM service monitors and updates the jobs database, which keeps track of each job as it goes through a series of steps to run the algorithm. Each available algorithm is defined in the jobs database, with a pointer to the executables needed to run each step in the algorithm. In order to run an algorithm, the front-end GUI inserts a row into the job table that specifies what type of job should be run and what datasets it should use. The service notices the new entry in the table and starts the job from step 0. The service has only two functions: 1) monitor and update the jobs database, and 2) run and monitor jobs on the cluster. The Auto-SAM service is implemented in Java, and the jobs database is implemented in MySQL. Auto-SAM uses a 240 node computing-cluster that is running condor (Thain et al. 2005)

Each algorithm that Auto-SAM runs has a series of steps, each with a specific executable and number of parallel runs, which are completed sequentially. All algorithms have a front-end step and a back-end step. The front-end step downloads the data from the database into files and creates the parameter files for the subsequent jobs. The service uses the parameter files at each step to determine the number of jobs to run and what parameters to pass to each job. Parameter files can be updated dynamically as the job progresses through the pipeline. Once all jobs for a given step have completed, the back-end step reads the results of the algorithm and then inserts them into the database. Thus, each processing pipeline only touches the database at the first and last step. This allows for the quick and easy integration of new algorithms into Auto-SAM, if the algorithms themselves are well-developed for general use.

At the completion of all jobs in a step, the service checks to make sure that there were no errors – all error reporting is done through the standard error output. If there were no errors in completing the step, the job is moved on to the next step. If an error did occur, the job is stopped and the analyst must choose to restart it or kill it altogether. At any time an analyst can choose to pause or kill a job. This is done by inserting a record into the pause table in the jobs database. If



PAA5082	57%	running	0 7:03:08
PAA5083	0%	paused	0 0:45:52
STR509c	25%	running	0 7:37:39
GFL509f	56%	running	0 6:57:46
GFL50a0	34%	running	0 6:57:28
GFL50a1	28%	running	0 6:57:12

Show: ○ running ○ error ○ complete ● all

Figure 3.4: Monitoring jobs in Auto-SAM. I implemented a job-monitoring system that regularly checks the progress of each job in the database. Using this monitor, the analyst can follow each job's progress, request error information, and pause and kill jobs. This job monitor is integrated into GenAMap.

the command is to kill the job, all condor processes are killed and the job is removed from the queue. However, if the command is to pause the job, the job is only stopped once all condor processes for its current step have completed. At this point, the analyst has the option to restart the job or to kill it. I have implemented a front-end control panel in GenAMap that monitors the progress of each job, allowing the analyst to pause, restart, and remove jobs (Figure 3.4).

The integration of an algorithm into Auto-SAM can be done by simply adding a front and back end, or through the complex compilation of several processing steps. I have used both of these strategies in designing and incorporating algorithms into Auto-SAM. For example, I have automated PLINK's Wald test (Purcell et al. 2007) for association between the genome and quantitative traits into Auto-SAM. In this case, the front-end step formats the data, the second step runs PLINK, and the final backend step inserts the results into the database. I used a similar strategy to incorporate Structure (Pritchard et al. 2000), a popular algorithm to generate population structure, into Auto-SAM.

On the other hand, I have taken advantage of the cluster in my integration of other algorithms into GenAMap. For example, when running a lasso job using glmnet (Friedman et al. 2010), Auto-SAM splits the dataset up into jobs the size of 250 traits and runs these each ten times separately (using different data splits for a defined vector of the regularization parameter, λ). Upon completion of this step, the next step reads in the results from each of the runs to choose the best λ based on cross-validation error over all traits. Similarly, when Auto-SAM calculates a correlation network, it splits up the network into sections of 1000 traits and calculates the values for each section in parallel, allowing the job to run much faster. While parallelization using Auto-SAM allows jobs to run much faster than they would run otherwise, the running time of each job is also affected by the number of jobs in the cluster and the number of threads accessing the database.

3.2.3 Running time analysis

Auto-SAM automates five structured association algorithms (GFlasso, TreeLasso, AMTL, MPGL, and gGFlasso), four structured association mapping algorithms (the Wald test, the Wilcoxon Sum-Rank test, the lasso, and a simple population analysis (Curtis et al. 2011)), five structure-generating algorithms (structure, correlation, hierarchical clustering, scale-free network construction (Zhang and Horvath 2005), and glasso). To demonstrate Auto-SAM, I run each algorithm on two publically available datasets. The first dataset that I use is a yeast expression dataset (Brem and Kruglyak 2005) consisting of 1260 SNPs and 5637 gene expression measurements for 114 individual yeast strains. The second dataset is the NIH heterogeneous stock mouse dataset (Johannesson et al. 2009). I use the phenotypic traits and preprocessed gene

Algorithm	Type	Input	Output	Mouse run time (traits) D HH:MM:SS	Mouse run time (genes) D HH:MM:SS	Yeast run time D HH:MM:SS	Automated steps
GFlasso (Kim & Xing 2009)	SAM	G, P, Ep	G-P association	0 05:05:50	1 16:17:45	2 06:43:56	17
MPGL (Puniyani et al., 2010)	SAM	G, P, Pop	G-P association	2 09:07:47	-	3 11:09:46	3
TreeLasso (Kim and Xing, 2010)	SAM	G, P, Ep	G-P association	0 01:12:03	0 12:53:52	0 05:08:42	15
AMTL (Lee et al., 2010)	SAM	G, P, Fg	G-P association	-	-	1 20:35:47	8
gGFlasso (Curtis et al., 2010)	SAM	T, Et, P, Ep, G/T assoc	T-P association	N/A	0 01:54:04	N/A	19
Wald Test (Purcell et al. 2007)	AM	G, P	G-P association	0 00:23:29	0 09:51:23	0 00:54:04	5
Wilcoxon Sum-rank test (Zhu et al. 2008)	AM	G, P	G-P association	0 00:10:21	0 01:05:51	0 00:14:32	4
Lasso (Friedman et al. 2010)	AM	G, P	G-P association	0 00:59:21	0 08:22:42	0 04:07:53	6
association by population (Curtis et al. 2011)	AM	G, P, Pop	G-P association	0 00:57:24	2 19:02:46	1 03:00:44	5
Correlation	network	P	Ep	0 00:01:50	0 00:06:05	0 00:09:29	3
Glasso (Friedman et al. 2007)	network	P	Ep	0 00:07:31	0 01:19:37	0 01:51:24	10
Scale-free network (Zhang and Horvath 2005)	network	P	Ep	0 00:03:11	0 00:41:23	0 00:41:08	6
Hierarchical clustering	tree	P	tree	0 00:43:32	0 01:27:32	0 01:05:04	3
Structure (Pritchard et al. 2000)	population	G	Pop	0 13:30:32	0 13:30:32	0 00:41:54	4
Gene module discovery (Zhu et al. 2008)	network analysis	Ep	phenotype clusters	N/A	0 12:29:24	0 01:03:22	4

Table 3.1: Algorithms available in Auto-SAM. I present a list of all algorithms available to run through the Auto-SAM system and GenAMap. I group the algorithms by type: SAM (structured association mapping), AM (association mapping), network generation, tree generation, population determination, and network analysis. The input for each algorithm can be G (genotype), P (phenotype), T (gene expression data), Ep (edges for the phenotype), Fg (features of the genotype), Et (edges for the gene expression values), and G/T (genome/transcriptome) associations. Times are formatted D HH:MM:SS where D = day, H = hours, M = minutes, and S= seconds.

expression measurements from the liver in my analysis of the stock mice. Thus, I have 12545 SNPs for 259 individual mice, which match up with the phenotype trait set of 158 clinical measurements and an expression dataset with 5965 gene expression measurements.

In Table 3.1, I list the running time from each algorithm, averaged over three independent runs, on the three publically available datasets. I also list the number of automated steps Auto-SAM runs to complete each algorithm. Each algorithm was run in parallel in Auto-SAM simultaneously with up to ten other algorithms. I monitored the cluster to limit the amount of time when all compute nodes were busy, and I limited the number of threads hitting the database to eight at any one time. I believe that the times I report are representative of what an analyst might expect with a heavy load in Auto-SAM. I suggest that these running times are orders of magnitude faster than downloading the implementations, editing data into the correct formats, and piecing the structure and association algorithms together to form a processing pipeline.

3.2.4 Auto-SAM Case Study: Integrating GFlasso into Auto-SAM

Here, I present one example, the integration of GFlasso with Auto-SAM, to demonstrate the complexity of the automated analysis that I undertake to scale these algorithms to large datasets. Auto-SAM's parallel processing environment can be leveraged to increase the scalability and decrease the running time of GWA and structured association mapping algorithms. To demonstrate how this can be done, I use my integration of GFlasso into Auto-SAM as a case study. I show how I have increased the scalability of GFlasso by adding processing steps and by using the cluster in Auto-SAM. I give an overview of this processing pipeline in Figure 3.5. In order for analysts to run GFlasso on their data without Auto-SAM, they would have to develop a similar process, writing their own code for each step.

GFlasso is a structured association mapping algorithm that finds associations between the genome and multi-correlated traits by leveraging the network structure between traits (Kim and Xing 2009). Auto-SAM uses the proximal-gradient optimization method for GFlasso (Chen et al. 2010). This optimization is written in MATLAB, and is therefore bound by memory constraints. The goal I had when I incorporated GFlasso into Auto-SAM was to develop a process that would run GFlasso for as many SNPs and as many traits as possible, while maintaining the integrity of the algorithm. After extensive testing, I found that using my test datasets (mouse and yeast), I could run GFlasso on datasets with 4000 SNPs and 250 traits without running out of memory. Auto-SAM supports loading in datasets much larger than this, and so I introduced a series of preprocessing steps to handle larger datasets. These preprocessing steps are performed automatically in Auto-SAM, without the laborious and time-consuming work to determine what needs to be done and writing the code for the processing of the data.

As shown in Figure 3.5, the first preprocessing steps are run to generate association results by running the lasso. These results are used to select 4000 SNPs to be used to run GFlasso. Because GFlasso looks for SNPs that are associated with multiple-correlated traits, Auto-SAM selects the SNPs that are associated with the most traits in the lasso results matrix. The lasso is run on the data for each trait, in parallel, for a given λ vector. Once all lasso runs complete, the results are combined and the λ with the smallest validation error is used to define a new vector for a fine-tuned search. These two steps are repeated to find the lasso results, which are used to select the markers for GFlasso.

The next two steps in the pipeline preprocess the traits. Unlike the SNPs, the traits can be split up into smaller sub-networks and run in parallel, assuming no edges between sub-networks and the same values for the regularization parameters. Auto-SAM finds these sub-networks first by finding all connected components. For all connected components greater than 250 traits, Auto-SAM runs spectral clustering (Chen et al. 2010) to break the sub-network down further. The trait processing step combines the connected components into sub-networks of 250 traits; GFlasso runs are run in parallel for each sub-network.

Three two-step processes run the GFlasso optimization. Auto-SAM spawns ten GFlasso runs for each sub-network of traits, each with a different division of the data. After all ten runs have finished for all sub-networks, an error calculation step calculates the cross-ten-validation error to select the best regularization parameters for that step. Following a linear search pattern, these two steps are first run for a vector of λ given γ, then for a vector of γ given λ, and finally for λ given γ. The backend copies the results into the database, which can then be accessed by the analyst through GenAMap.

Thus, a sixteen-step process is completely automated in Auto-SAM. This pipeline preprocesses the data, runs optimization runs in parallel, and finds a solution optimal according to cross-ten validation. Whereas the available MATLAB code can only handle datasets with 250

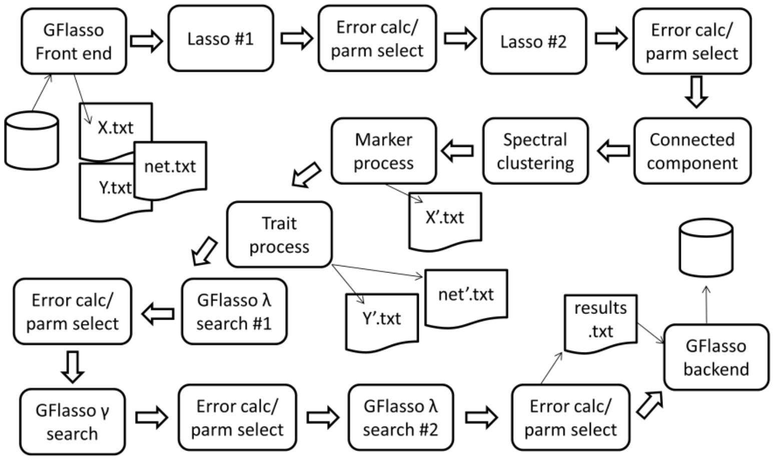


Figure 3.5: GFlasso integration into Auto-SAM. I show an overview of the steps followed in Auto-SAM in order to run GFlasso. These steps represent the combination of many different command-line programs, each automatically tied together in order to preprocess the data, find regularization parameters, and return the results to the analyst.

traits, Auto-SAM can run datasets with up to 20,000 traits (which is the maximum that the data database can hold). I follow a similar process to automate many other structure-generating and structured association mapping algorithms.

3.3 Visualization

To demonstrate the visualization tools in GenAMap, I will present a series of use cases using real data. Through these case studies, I highlight the structured association mapping methods available to run in GenAMap, and I also describe the visualizations available to explore the results from these analyses. I also note that analysts can upload results from network or association analysis and explore these results using GenAMap without uploading their raw results into Auto-SAM.

3.3.1 Exploring gene networks and eQTLs using yeast data

I first introduce GenAMap by analyzing a dataset from budding yeast, *Saccharomyces cerevisiae* (Brem and Kruglyak 2005). Because this dataset has been extensively studied (Zhu et al. 2008; Brem and Kruglyak 2005; Yvert et al. 2003; Lee et al. 2009), it serves as an excellent dataset to highlight the capabilities of structured association mapping and GenAMap in a scenario where much is already known about the associations in the data. This dataset was generated by crossing a laboratory strain (BY4716) of yeast with a wild-type vineyard strain (RM11-1a) to create 112 progeny yeast strains. Each of the 114 strains were sequenced for 1260 unique SNP markers. Gene expression data were also collected from each strain for over 6000 traits. After preprocessing the gene expression data, I used 5637 gene expression measurements for each yeast strain. The data collection and preprocessing steps were completed independently outside of the GenAMap software system.

3.3.2 Exploring the gene network

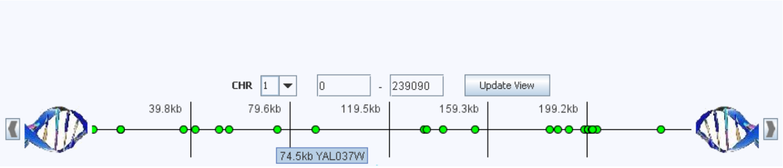


Figure 3.6: GenAMap's genome browser. GenAMap provides a simple genome browser that allows analysts to explore the mutation marker data that they load into GenAMap. SNPs are represented by green circles across the genome. Analysts can link to external databases such as SGD or dbSNP. SNP labels are displayed as the analyst hovers over SNPs.

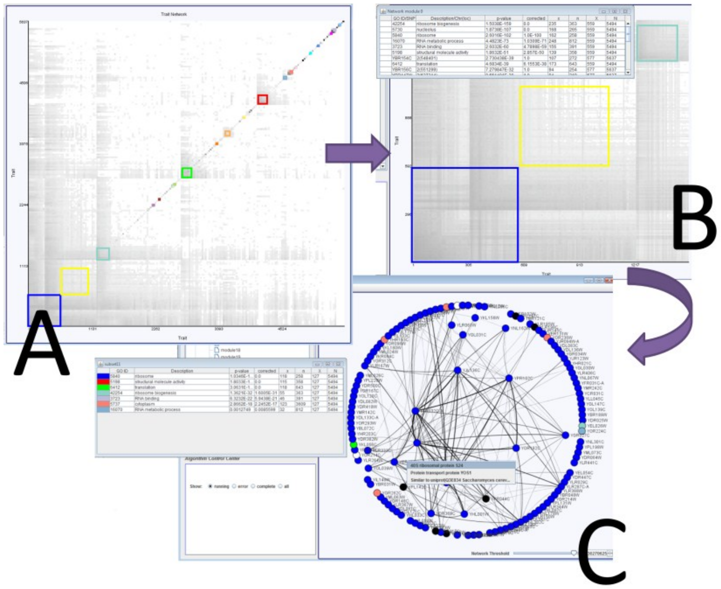
Once the data have been preprocessed, it is ready to import into GenAMap to begin association analysis. I import the SNP data as a tab-delimited file into GenAMap using the import wizard. When the import finishes, I can explore the data using GenAMap's genome browser (Figure 3.6). GenAMap's genome browser is a simple chromosome-by-chromosome browser that displays each SNP as a green circle. I use the genome browser to check the distribution of SNPs on each chromosome and to directly link to the *Saccharomyces* Genome Database (SGD 2011) for more information about the SNPs. For future analyses, I download and standardize twelve features from the SGD for each SNP and add these features to the dataset in GenAMap. These features include eleven discrete variables describing the location of the SNP (intron region, binding site, exon, etc.) and one continuous feature (conservation score) (Lee et al. 2010). As I browse through the SNPs, I can request to see these features by right-clicking on a selected SNP (or many SNPs) in the genome browser.

Similarly, I load the gene expression data into GenAMap using the import wizard. Once the gene expression data have loaded into GenAMap, I use GenAMap to automatically build a gene-gene network using the soft-thresholding method to create a scale-free topological overlap matrix (Zhang and Horvath 2005). GenAMap can also create this network from calculating the pairwise correlation coefficient between genes, or by using glasso (Friedman et al. 2007). Once the network is created, I want to get an overall picture of the gene interactions to understand the network structure. GenAMap supports this type of analysis through the discovery of gene modules within the network. To find gene modules, I first use GenAMap to automatically run a simple pairwise association test, the Wilcoxon-Sum Rank test with false discovery rate (FDR) correction (Zhu et al. 2008), to find SNP-gene associations. Simultaneously, I use GenAMap to run hierarchical clustering to cluster highly connected genes in the network. Once these two algorithms finish running, I run a job in GenAMap to discover the top twenty connected gene modules in the clustered network using a dynamic programming method (Zhu et al 2008). GenAMap identifies these modules automatically on the parallel computing cluster and also calculates eQTL enrichment (using the pairwise associations) and gene ontology (GO) enrichment (using BiNGO (Maere et al. 2005)) for each module.

At this point, I have used GenAMap to prepare a gene expression data network; the next step is to use GenAMap's visualization tools to explore the gene network to find interesting interactions. First, I explore the patterns in the entire network to get an overview. In Figure 3.7 I show a screen-shot of GenAMap's overview visualization of gene networks. This overview is presented as a heat map, where darker pixels represent a weighted relationship between genes. The genes in the heat map have been clustered, and 20 identified modules are outlined in color. As I select different gene modules in the network, GenAMap displays the module's eQTL and GO enrichments. I find that the modules are significantly enriched for specific GO categories and eQTL associations, consistent with previous reports (Zhu et al. 2008).

As I referenced in the design section, all of GenAMap's visualization tools are developed to give an overview first, provide tools to zoom and filter, and then present links to details on demand. The network view follows this pattern. Once I have a big picture of the gene network, I can use GenAMap to zoom into the data to explore interesting sub-networks. I provide one simple example of this drill-down exploration.

From the network overview, I observe that the largest of all the modules in the network is the blue module, made up of 788 genes. This module is enriched for many GO categories including *ribosome biogenesis* (*p*-value = 4.0604e-169) and has eQTL enrichment to many SNPs including a SNP on chromosome II:548401 (*p*-value = 2.769e-48). To explore this sub-network further, I manually zoom into this region of the heat chart display. GenAMap displays gene and trait networks at a series of resolutions, and so as I zoom into this region of the network I see the finer detail of the gene-gene relationships (Figure 3.7C). I select the most highly connected part of the network and switch to the *node-link view*, which displays sub-networks of up to 200 traits/genes in a node-link representation. I summarize the process I follow to zoom into this region in Figure 3.7.



Network module 8

	GO ID/SNP	Description/Chr(loc)	p-value	corrected	x	n	X	N
	42254	ribosome biogenesis	1.5038E-159	0.0	235	363	559	5494
	5730	nucleolus	1.8738E-107	0.0	168	265	559	5494
	5840	ribosome	2.6016E-102	1.0E-100	162	258	559	5494
	16070	RNA metabolic process	4.4823E-73	1.0389E-71	248	812	559	5494
	3723	RNA binding	2.6032E-60	4.7898E-59	155	381	559	5494
	5198	structural molecule activity	1.8632E-51	2.857E-50	139	358	559	5494
	YBR154C	2(548401)	2.730438E-38	1.0	107	272	577	5637
	6412	translation	4.6834E-39	6.1553E-38	173	643	559	5494
	YBR156C	2(551289)	7.278847E-32	1.0	94	254	577	5637

subset11

	GO ID	Description	p-value	corrected	x	n	X	N
🟦	5840	ribosome	1.0346E-1...	0.0	118	258	127	5494
🟥	5198	structural molecule activity	1.8033E-1...	0.0	115	358	127	5494
🟩	6412	translation	3.0631E-1...	0.0	118	643	127	5494
■	42254	ribosome biogenesis	1.3621E-32	1.6805E-31	55	363	127	5494
■	3723	RNA binding	8.3232E-22	5.8438E-21	46	381	127	5494
■	5737	cytoplasm	2.8662E-18	2.2452E-17	123	3609	127	5494
■	16070	RNA metabolic process	0.0012749	0.0085599	32	812	127	5494

Show: ● running ○ error ○ complete ○ all

Figure 3.7: Exploring a gene network in GenAMap.(A) An overview of the entire network, with gene modules identified. (B) Zoomed in regions of the network with GO functional enrichment. (C) Node-link representation of specific regions in the network colored by GO category.

In the node-link view, the genes are now represented as circle nodes, and relationships between genes in the network are represented as weighted lines. Thicker lines imply a strong weight of connection between genes. There are several different layouts available in this view, including a simple circle layout and the KK-layout (Kamada and Kawai 1989) I can also use dynamic query techniques (Shneiderman 1994) to adjust a threshold that controls which edges are displayed. Because this is a highly connected part of the network, I filter out edges with low weights to reveal the highest connected genes in the sub-network. Now that I have zoomed into this region, I can use GenAMap to get specific details about these genes. I perform a GO enrichment analysis, which finds that the genes I have found are enriched for the GO category *ribosome* (*p*-value = 4.89e-169). I adjust the edge threshold manually to remove edges with lower weights; this allows me to find the most highly connected genes in the network. Because the top-connected genes in this network may be important players in the sub-network, I right-click on these gene's labels to link directly to Google search and to the gene's UniProt webpage (The UniProt Consortium 2011). These details on demand potentially help me to understand what genes are particularly active in this sub-network, for example *RPS24A*, a ribosomal protein from chromosome V is one of the genes with the most connections in the network.

3.3.3 Finding eQTLs using GenAMap

Given the high modularity of the gene network, I decide to run TreeLasso (Kim and Xing 2010) to find SNPs associated with genes. TreeLasso uses the network structure to cluster the genes into a tree structure, which guides the algorithm to find associations from SNPs to related genes. I can explore the results from running TreeLasso using the *network association view*. The network association view is similar to the network view, integrated with the genome view. As with the network view, the analyst can explore the overview of the data, zoom in and filter, and then get details on demand. The network association view incorporates tightly coupled coordinated views (North and Shneiderman 2000), allowing the analyst to interactively correlate between SNPs and the network. The analyst will use this view to get a feel for the data and find specific SNP-gene associations for further investigation.

In Figure 3.8, I present an overview of the results from running TreeLasso automatically in GenAMap. I first consider GenAMap's overview of the association results. This heat map shows a matrix of the association values. SNPs are shown along the *y*-axis, and the genes are shown along the *x*-axis; the traits have been clustered by hierarchical clustering. In this case, the associations are represented by a 1260x5637 matrix. Black represents a strong association, and white represents no association. For both this heat map view and the network heat map, the

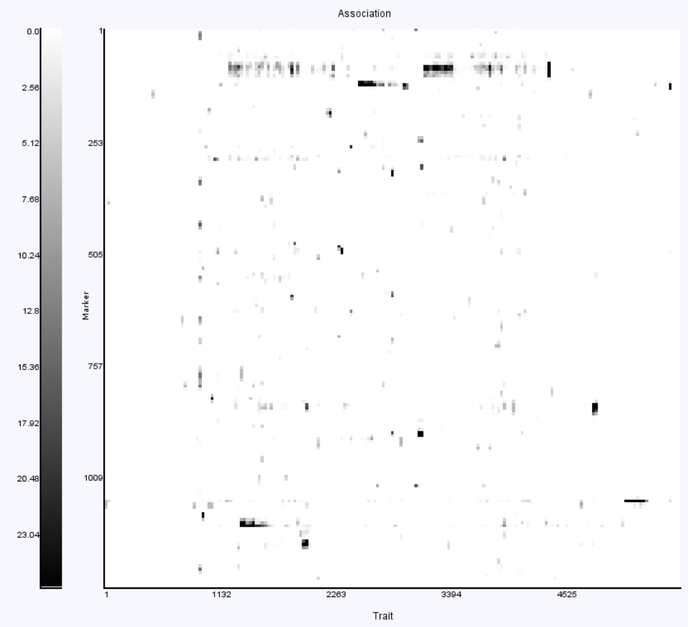


Figure 3.8: GenAMap overview of association results. GenAMap provides a heat chart visualization to explore the results from an eQTL association analysis. SNPs are plotted along the *y* axis and genes are clustered along the *x* axis. This view allows the analyst to explore the overview of the results. For example, in these results from running TreeLasso on the yeast data, many SNPs are associated with all the genes in a gene module, and some gene modules are associated with many different SNPs in different genomic locations.

analyst can zoom in and out of the matrix through a series of resolutions. Each resolution is a 200 pixel by 200 pixel matrix; the association results initially displayed to the analyst in Figure 3.8 is at a resolution where each pixel represents six SNPs and 30 genes. Because the data are inherently sparse, GenAMap colors the pixel by the maximum association value between all SNPs and traits represented. This ensures that the analyst can focus on the signals in the data as the signals are preserved even at lower resolutions.

From the results shown in Figure 3.8, I can get insight into the yeast regulation patterns present in these results. For example, I notice the series of long (and short) horizontal black lines in the matrix. These lines represent associations between a SNP and a cluster of genes. The presence of such patterns indicates to the analyst that gene clusters in the yeast network are associated with a common SNP. Because these lines overlap, I can conclude that some gene clusters are associated with multiple SNPs, representing a case where multiple mutations affect the same set of genes. This view has made it visually obvious which of the gene clusters are associated with multiple genetic locations and approximately where in the genome these associations lie. I can now use my knowledge of the gene network I obtained from the network view to zoom into clusters of traits that are associated with different SNPs. Because I have previously identified the ribosome genes, I zoom into the part of the heat map representing these genes. I notice that there are ten SNPs in the same genomic region that are associated with these genes. To explore these associations, I select 131 genes in the cluster with strong associations and switch to the node-link view (Figure 3.9). From the heat map view, I know that these genes are associated with SNPs on different chromosomes and I want to explore these associations. In the node-link representation of the network, the analyst can explore the gene structure of the network while identifying associations. The view is integrated with a simple genome browser

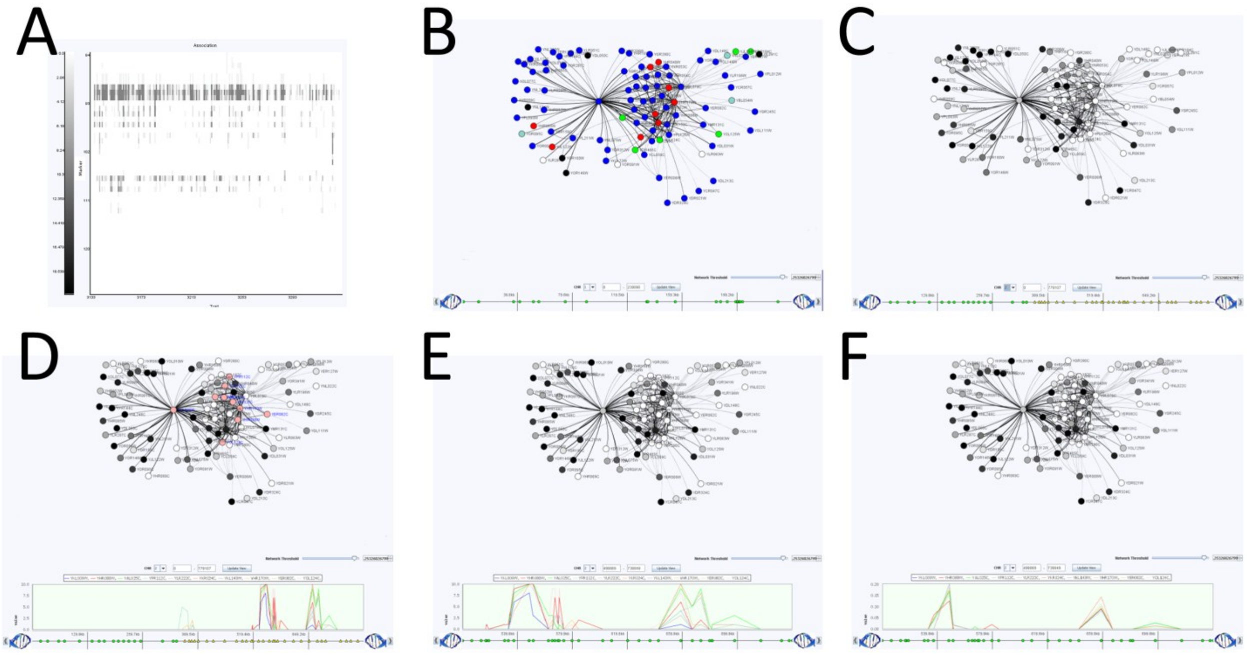


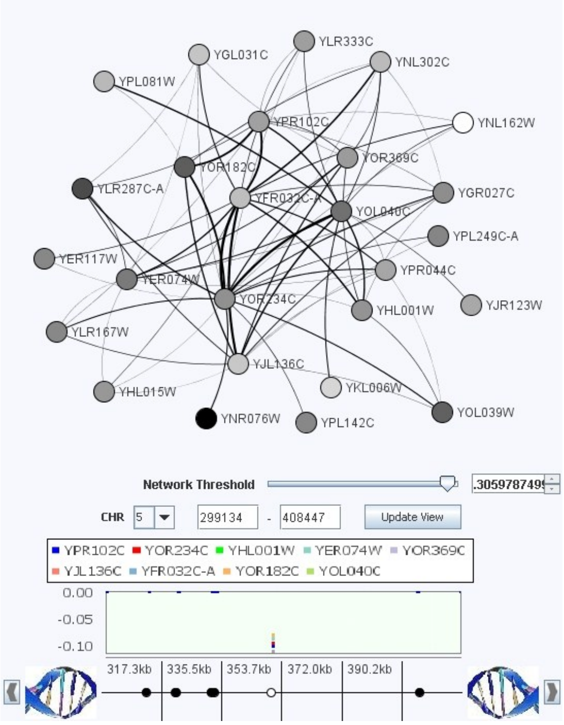
Figure 3.9: Using GenAMap to find eQTLs. GenAMap provides many tools for analysts to explore association results while using the structure of the data to guide the discovery of associations. I demonstrate some of these tools. A) The analyst can zoom into certain regions to see finer detail of the SNP-trait associations. This panel is a zoomed-in region from Figure 3.8. B) The analyst switches to the JUNG view to explore the genes associated with the region and perform a GO enrichment test. C) The analyst colors the genes by strength of association to the genomic region. D) The analyst selects up to ten interesting genes (salmon colored) and views the Manhattan plot of associations from these genes across the genome. E) The analyst zooms into interesting regions in the genome view. F) The analyst can switch between association tests for further insight into the associations.

where nodes represent SNPs (Martin and Ward, 1995) (bottom of Figure 3.10). I can use this genome browser to switch between chromosomes and zoom into certain chromosomal regions. By using these coordinated views, the analyst can explore structure in the genome and the traits, while simultaneously querying to understand the associations between the two.

Once in the node-link view, I perform a GO enrichment test to see if my selected genes have a common function. Indeed, the genes are enriched for the GO annotations *nucleolus* (*p*-value = 2.091e-107), *ribosome biogenesis* (*p*-value = 2.623e-99) and *RNA metabolic process* (*p*-value = 7.081e-66). In Figure 3.9B, I show these genes colored by GO category. All genes with the GO annotation for “nucleus” are shown in blue. Genes without this annotation, but annotated as “ribosome biogenesis” are shown in red. Genes not annotated as either, but annotated as

"RNA metabolic process" are shown in green. I can change this coloring based on the GO categories I am interested in. Based on this analysis, I conclude that this group of genes appears to be a functionally cohesive group of genes involved in ribosome biogenesis that co-locate to the nucleolus.

From the overview of the data, I already know that these functionally coherent genes have strong associations to at least ten SNPs on chromosome II. I select half of chromosome II and color the genes by association to the selected SNPs (Figure 3.9C). Genes with a strong association to these SNPs are shown in white, genes shown in black are not associated, and gray genes represent varying levels of association. In this view, the SNPs being considered are shown as yellow triangles for my reference. I discover that most of the genes in this module are associated to this region on chromosome II. To further explore this association, I select ten genes



Network Threshold .305978749

CHR 5 299134 - 408447

■ YPR102C ■ YOR234C ■ YHL001W ■ YER074W ■ YOR369C
■ YJL136C ■ YFR032C-A ■ YOR182C ■ YOL040C
0.00
-0.05
-0.10
317.3kb
335.5kb
353.7kb
372.0kb
390.2kb

Figure 3.10: In the network association view, GenAMap shows interaction between genes, integrated with the association strengths of the genes to SNPs in the genome.

in the module (highlighted in salmon in Figure 3.9D) and view the Manhattan plot of the associations strengths of these genes across chromosome II. I zoom into the region with the strongest associations (Figure 3.9E) and note that there are many SNPs with associations to these genes as observed from the overview.

The number of SNPs associated to these genes complicates the association analysis. I want to find the specific SNPs that are the most likely to be associated with the genes in this module. Because I have already added feature data to the SNPs, I can run AMTL to find associations in the yeast dataset (Lee et al., 2010). AMTL, unlike TreeLasso, takes into account SNP features instead of genetic structure. Thus, AMTL selects gene-SNP associations based on features that predict the likelihood of the SNP to cause a change in gene expression. Once the AMTL analysis is complete, GenAMap allows me to switch between the TreeLasso results and the AMTL results easily in the same view of the data (Figure 3.9E and 3.9F). Indeed, the AMTL results find associations to far fewer SNPs on chromosome II for this set of genes. I inspect the two SNPs on chromosome II that have associations in the AMTL results to these genes. I use GenAMap to link to the SGD for more information about these SNPs and find that one SNP is in *RPB5*, a component of RNA polymerase, and the other SNP is in *PYC2*, near *SDS24*.

In this demonstration using the yeast data, I have shown how GenAMap enables an analyst to survey a gene expression network to find modules and then to drill down for further detail about these modules. I have also demonstrated how GenAMap enables the exploration of association results to discover gene modules under the regulation of eQTL hotspots. Finally, I have shown how GenAMap allows analysts to compare the results from different structured association mapping tests to better understand association signals.

3.3.4 Exploring association results through the association tree view

In the previous example, I was interested in associations to gene clusters. Often geneticists will want to explore the associations of a particular SNP or SNP region to find out if the genes associated with a SNP are actually in a gene cluster, or to find the strongest associations from a SNP to genes. GenAMap's *association tree view* allows for these types of explorations (Figure 3.11).

In the *tree view*, the leaves of the tree represent genes, and other nodes represent the aggregation of genes descending from them in the given tree. Each non-leaf node is labeled by the number of aggregated genes below the node and by a GO enrichment annotation (if the genes have a significant functional enrichment). By default, the nodes are colored by this GO annotation. The tree view only shows three to eight levels of the tree at a time and allows the analyst to browse through the tree. In the association tree view, the tree view is integrated with

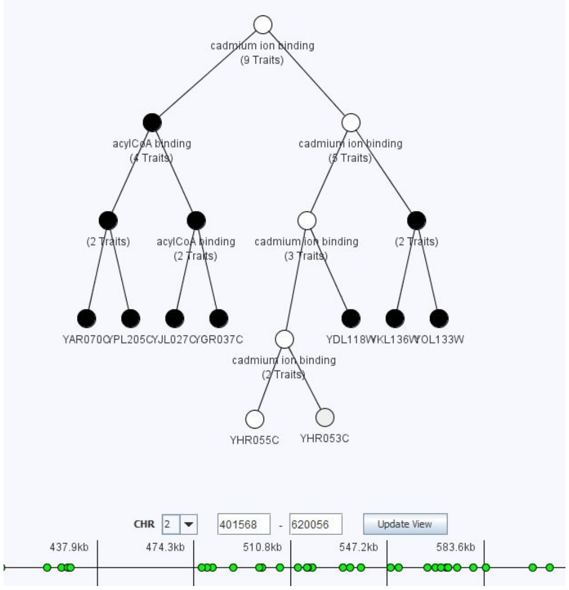


Figure 3.11: In the association tree view, the analyst explores genes structured as a tree in order to identify functionally relevant branches of the tree that are associated with a genomic region.

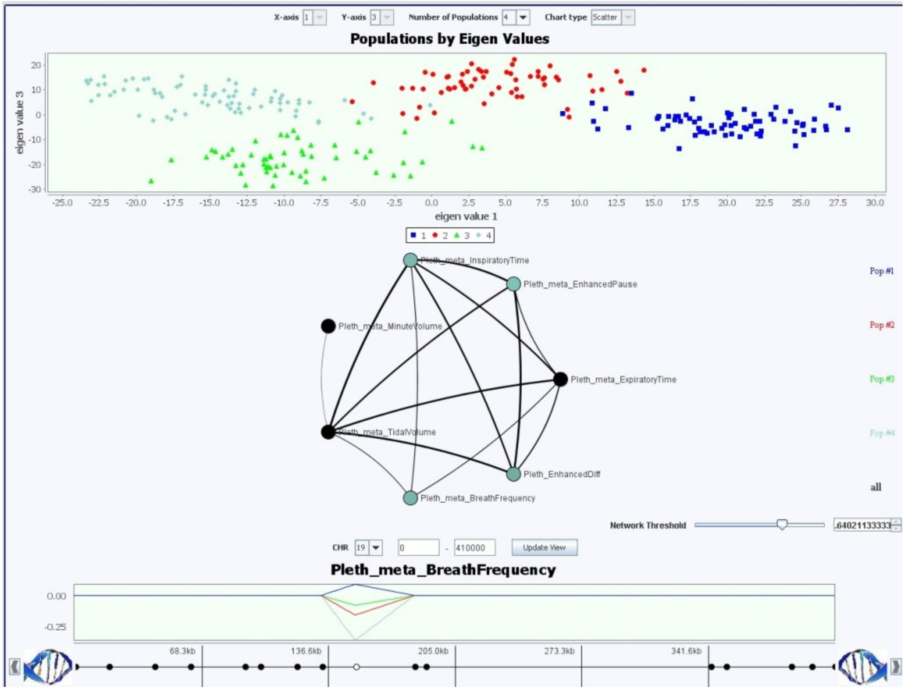
the *genome view*. Let's say I am specifically interested in a genomic region on chromosome 2 (base-pair 560000) in results generated from running GFlasso on the yeast data. From the association tree view, the I browse to this genomic location, selects several SNPs in the region, and colors the tree by association to these SNPs. Each node in the tree is then colored by strength of association to these SNPs, white represents a strong association to the genome location and black represents no association. As seen in Figure 3.11, a non-leaf node is colored by the strength of the strongest association of all the traits it represents.

I am interested to find the genes with the strongest associations to these SNPs. From the root of the tree, I follow the white nodes to browse down the tree until I find the genes (leaves, shown in Figure 3.11). Interestingly, this part of the tree only had two genes associated to this genomic locus. I look these genes up in UniProt through GenAMap's links to find out what they are and why they might be affected by mutations in this genomic location. I can also use the genome browser to link to the SNP location in the *Saccharomyces* Genome Database (SGD 2011).

Further exploration in the tree will allow the analyst to find associations between the genes and SNPs, identify whether other related genes in the tree are also associated, and discover the common GO enrichment of associated branches in the tree.

3.3.5 A case study on a mouse GWA dataset

The second dataset that I consider is a mouse dataset (Johannesson et al. 2009). This dataset has measurements for 179 clinical traits and 12546 SNPs for 269 mice. Using a dataset with clinical traits and SNPs allows geneticists to identify SNPs that are associated with a particular disease trait of interest. In this example, I will focus on traits related to asthma in mice. These views are also referenced in Section 4.2.



X-axis 1 Y-axis 3 Number of Populations 4 Chart type Scatter

Network Threshold .64021133333

CHR 19 0 - 410000

Figure 3.12: The population association view is an integrated view enabling the exploration of association strengths across different populations. The Manhattan plot shows the association strengths across each population for a given trait, and the traits are colored by the color of the population with the strongest association to that trait.

3.3.6 The population structure view

From my knowledge of the data source, I believe that there is population structure in the data. I explore the population structure through the *population structure view* (the population structure view is integrated into the top half of Figure 3.12) after running machine learning to generate population structure (Pritchard et al. 2000) and eigenvalues from the data. Individuals are plotted according to their eigenvalues, and colored according to population assignment. I dynamically color the plot for different numbers of populations. I can also see the number of individuals assigned to each population using a pie chart. I find that the mice split up into four distinct subpopulations across the first five eigenvalues.

3.3.7 The population association view

In the *population association view* (Figure 3.12), I explore the results from MPGL on the mouse data with four populations. GenAMap integrates the population structure view, the network view, and the genome view to help analysts explore these associations. I explore the overall network and identify seven traits related to asthma for further exploration.

I want to find the SNPs associated with these traits. I color the genome by association to these traits, and identify a SNP on chromosome 19 that is strongly associated to at least one trait. I select this SNP, able now to ignore the rest of the genome, and color the traits in the network that are associated to this SNP. Each trait is colored by the color of the population with the largest β value (association). I find four asthma traits associated with this SNP, with the strongest association in each case being the association to population #4. I investigate the association of each of these traits one-by-one by adding the Manhattan plot to the genome view. For the trait *breath frequency*, I find that population #4 and #1 are strongly associated with this SNP, more than population #2 and #3. I investigate this association by linking dbSNP (Sherry et al. 2001) through GenAMap. I suspect that this SNP on chromosome 19, or one close-by, plays a role in asthma in mice.

3.3.8 Conclusions

In conclusion, GenAMap provides many different views and tools to enable different types of structured association analysis. In the next section, I will discuss the design of a three-way (SNP-gene-trait) visualization toolkit that I have implemented in GenAMap.

3.4 Three-way Visualizations

The GFlasso-gGFlasso analysis (Section 2.4) of a genome-transcriptome-phenome dataset leaves the genetics analyst with a large, complex sea of data to interpret. In addition to the gene-gene and trait-trait relationships, potentially hundreds of SNP-to-gene and gene-to-trait associations are identified. The analyst must explore this data to pinpoint the associations that lead to insight into disease.

The analyst might explore the data with different strategies, depending on what questions he/she is interested in. In one scenario, the analyst comes into the study with questions about specific traits. The analyst explores the network relationships between these traits and looks for associated gene networks. He/she then examines the gene networks for genomic associations, leading to the discovery of SNPs that perturb the gene networks associated with the phenotypes of interest. Alternatively, the analyst starts with a genomic region of interest. In this case, the analyst first considers the genomic region and its associations to genetic pathways. He/she then identifies the traits associated with the discovered genes. In either scenario, analysts have to filter through the dataset in an exploratory fashion. As visualization methods are particularly adept at guiding the identification of interesting data through exploratory analysis (Fekete et al. 2008), visualization can be a powerful tool in either scenario.

To design a visualization system that would facilitate the exploration of three-way association results with gene and trait structure I follow Shneiderman's mantra (Shneiderman, 1996): 1) *overview first*, 2) *zoom and filter*, and 3) *details on demand*. In this section, I discuss my design of the visualizations, which are implemented in GenAMap using Java and the JUNG toolkit (Madahain et al. 2005). A video overview of these visualization tools is available online at http://sailing.cs.cmu.edu/genamap/threeway.html.

3.4.1 Overview first: an introduction to the three-way visualization

When the analyst first examines three-way association results in GenAMap, he/she is presented with an overview of the data (Figure 3.13, 3.14). Traits are represented as blue hexagons and edges in the trait network are displayed as weighted (by strength of correlation) gray lines. Genes are grouped according to GFlasso results: genes associated with common SNPs form a group. Visually, gene groups are represented as circles where the size of the circle represents the size of the group. Edges between genes groups are shown as black lines, the thickness of the line representing the number of edges between the genes in the two groups.

GenAMap enables the analyst to explore the three-way results through two different layouts. The KK-layout is designed to present trait-trait and gene-gene structures separately (Figure 3.13). The positions of traits are determined by the KK-layout algorithm (Kamada and Kawai 1989) and gene groups are plotted in a half-circle with the ten largest groups placed in an arc.

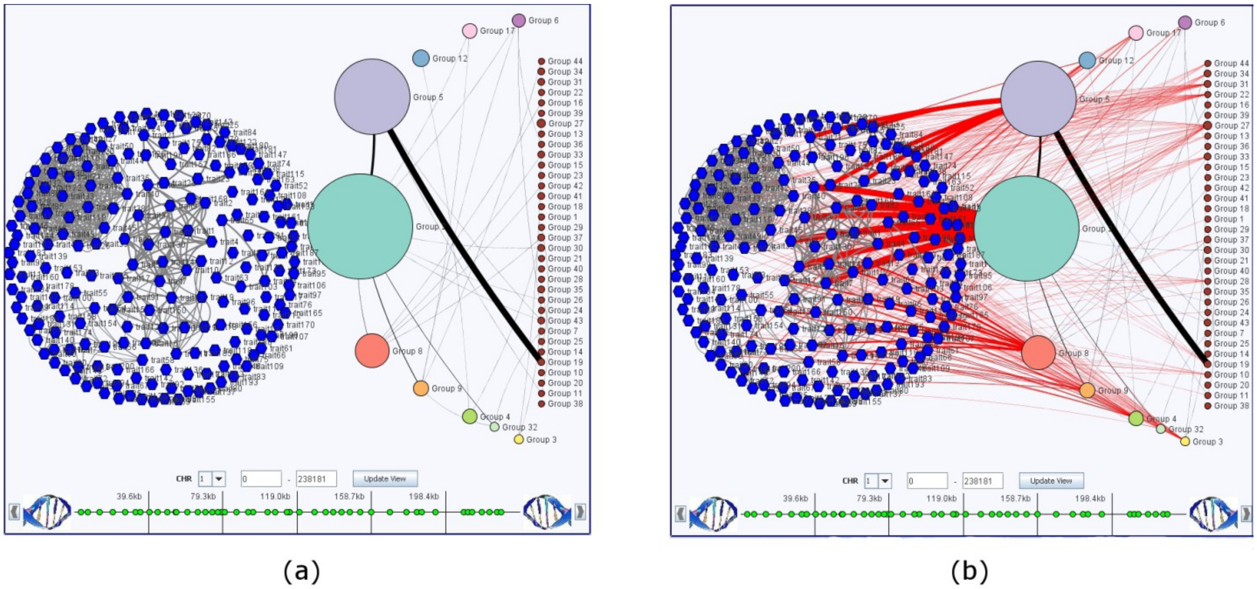


Figure 3.13: Overview of three-way association results using the KK-layout. (a) The structure (edges) of the traits (blue hexagons) and gene groups (circles) are shown without showing the association edges. (b) The red association edges from gene groups to traits are shown. SNPs are represented at the bottom of the display as green circles in GenAMap's genome browser.

Association edges are shown in red. When the analyst hovers over a gene group, a tool-tip reports the group's gene count and significant gene ontology (GO) enrichment.

Analysts can also explore three-way association results using a force-directed layout (FDL). In this layout, gene and trait nodes repel each other and edges act as springs that pull nodes together. The analyst can adjust the repulsion and attraction parameters to adjust the display. In Figure 3.14, I show the data from Figure 3.13 now represented in the FDL. In Figure 3.14a, the association and correlation edge spring tension is high, causing connected nodes to pull into a tightly clustered group. As the spring tension is relaxed (Figure 3.14b), the structure of the connected gene-trait clusters is visible. In both the KK-layout and FDL, the analyst can customize the display by turning labels on and off, adjusting parameters, or manually repositioning nodes.

GenAMap enables the analyst to simultaneously explore SNP-gene and gene-trait associations. The SNP-gene association strengths are visualized using color-encoding. For example, the analyst selects SNPs of interest and clicks to color all genes by association. The

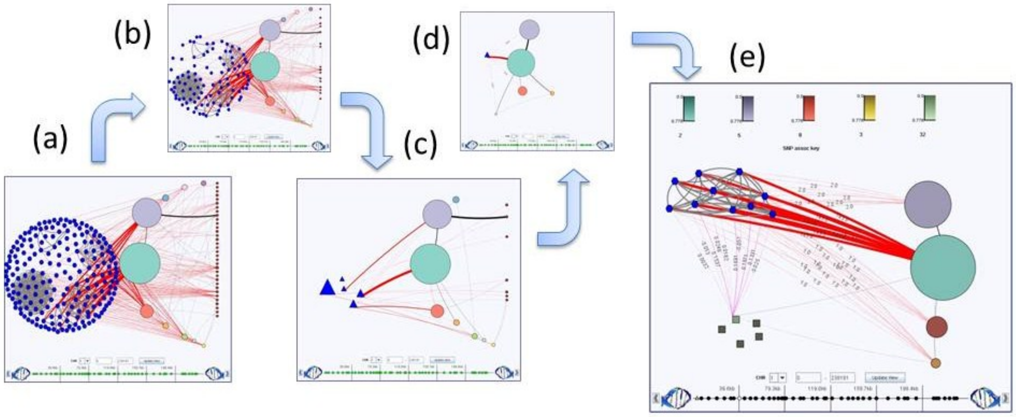


Figure 3.15: Demonstration of zoom and filter tools in GenAMap. (a) The analyst starts with all data visible and filters data based on association resulting in (b). He adjust the association edge threshold and groups traits into trait groups (c). Trait groups and gene groups not connected/association with a trait group are removed (d). A trait group and gene groups of interest are expanded and the analyst considers genome-transcriptome-phenome association simultaneously (e).

gene view updates so that the brightness of the color of each group represents its strongest association to the selected SNPs. Alternatively, the analyst selects gene groups of interest and colors the SNPs by association. SNPs are then colored: white represents a strong association and black represents no association. In Figure 3.15e I demonstrate both of these encodings: SNPs are colored by association to the teal gene group, and groups are colored by association to selected SNPs.

3.4.2 Zoom and filter: identifying interesting signals in the data

GenAMap is designed to enhance the analyst's ability to filter through the dataset to identify genome-transcriptome-phenome associations. I demonstrate GenAMap's filter and zoom tools through a series of steps shown in Figure 3.15. In Figure 3.15a, the data are loaded in the KK-layout. The analyst uses a filter to removes all genes and traits without associations (Figure 3.15b). Next, the analyst adjusts the association edge threshold to remove weak associations. Interested in the largest gene group (Group 2, teal color), he/she removes all groups that do not have a network edge to Group 2. Each connected component in the trait graph is then collapsed into a *trait group* (represented as a triangle) to simplify the display (Figure 3.15c). To explore only the trait group with the most associations to Group 2 (thickest edge), all trait groups and gene groups not connected to this trait group are manually removed (Figure 3.15d). Finally, the analyst expands the trait group and a strongly-associated gene group. He colors the SNPs by association to the now-visible genes (squares) in this gene group. After identifying associated SNPs, the analyst colors all genes by association to the identified SNPs (Figure 3.15e). In summary, GenAMap allows the analyst to filter and explore based on network connectivity, association, edge thresholds, and grouping strategies. These strategies can be employed in any three-way analysis, starting from traits, SNPs, or genes.

3.4.3 Details on demand: resources for further exploration

Once the analyst has found interesting gene-trait associations, GenAMap directly links the analyst to more information. The analyst can directly link to the UniProt database (The UniProt Consortium 2011), Google search, or to dbSNP (Sherry et al. 2001). GO information for genes is available through GenAMap's integration with BiNGO (Maere et al. 2005). The analyst can also consult association strengths for the genes using interactive Manhattan plots, as shown in Figure 3.16.

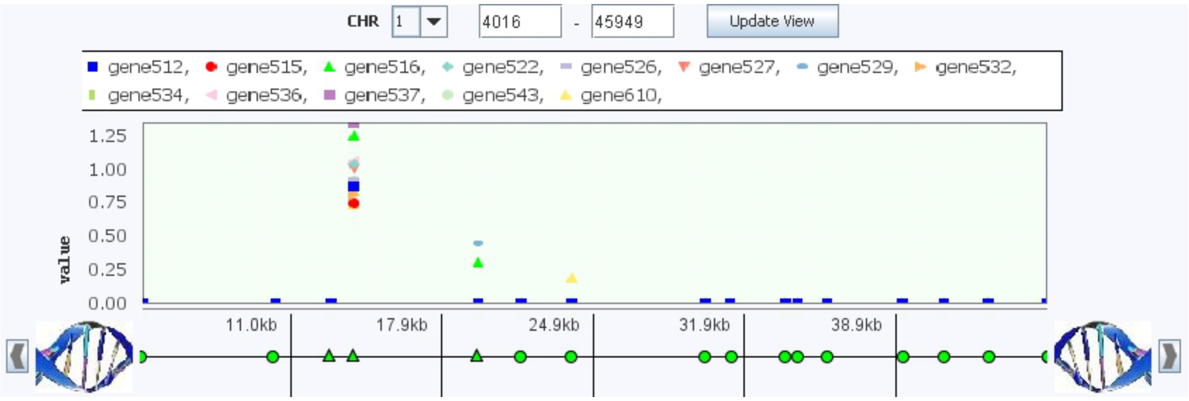

CHR
1
4016
-
45949
Update View
■ gene512, ● gene515, ▲ gene516, ◆ gene522, ▬ gene526, ▼ gene527, ● gene529, ▶ gene532,
▮ gene534, ◀ gene536, ■ gene537, ● gene543, ▲ gene610,
value
1.25
1.00
0.75
0.50
0.25
0.00
11.0kb
17.9kb
24.9kb
31.9kb
38.9kb

Figure 3.16: A Manhattan plot of the SNP-gene association results for all genes in one gene group.

4. Using GenAMap in Association Analysis

In this chapter, I present three genetics analyses that I have conducted using structured association mapping and GenAMap. The first analysis is an analysis using a yeast gene expression dataset. I show that by using GFlasso, I uncover interacting eQTL hotspots. In the second analysis, I use structured association mapping and GenAMap to analyze a human asthma dataset; the analysis leads to the discovery of a potentially new asthma gene. Finally, I use structured association mapping and GenAMap to analyze a mouse dataset to explore the three way associations between the genome, gene expression data in three tissues, and clinical trait measurements.

4.1 GFlasso analysis leads to insight into *Saccharomyces cerevisiae* gene regulation by uncovering interacting eQTL hotspots

In this section, I present the results from a study that is joint work with Dr. Seyoung Kim, Dr. John Woolford, and Dr. Eric Xing. Although I was the lead on the project and performed the analyses, I did so with significant input from the other authors on the paper. This study was an exciting application of structured association mapping and GenAMap to a yeast eQTL dataset.

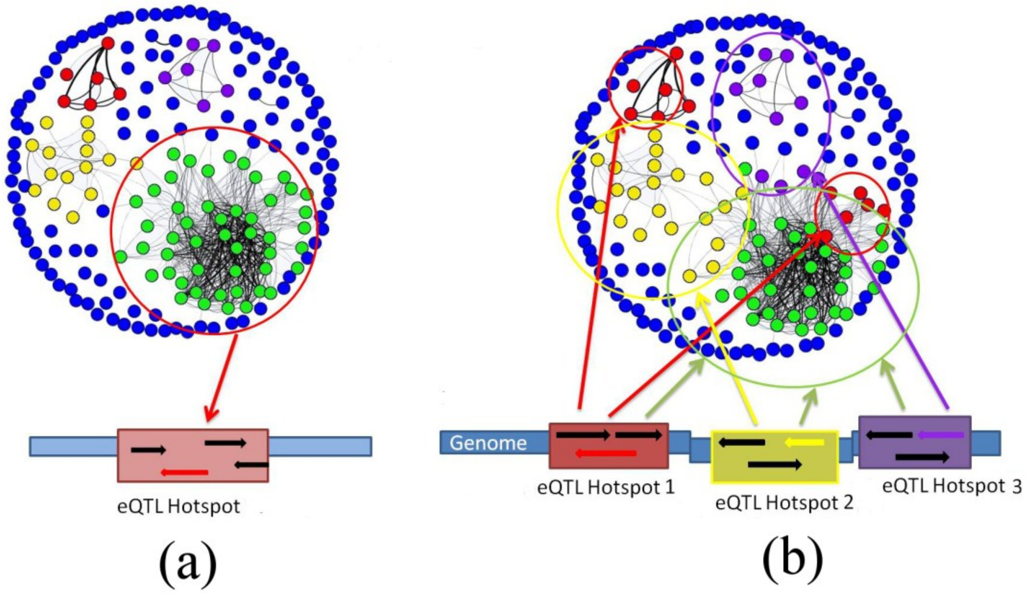


Figure 4.1: An illustration of our main results. (a) Previous analyses of the yeast eQTL dataset reported eQTL hotspots, a module of multiple genes controlled by the same genomic locus. (b) In our GFlasso analysis of the same dataset, we not only found eQTL hotspots, but also discovered interaction among multiple eQTL hotspots, where the same module of multiple genes is controlled by multiple eQTL hotspots. This figure was created using GenAMap.

In this study, we use GFlasso to reanalyze the yeast eQTL dataset available from Brem and Kruglyak (2005) with a new focus on uncovering the genetic basis behind the coupled gene-expression traits. The dataset includes the genome-wide profiling of expression levels and SNPs for 112 recombinant progeny from two parent strains, a laboratory strain and a wild vineyard strain. We chose this particular dataset because it has been previously analyzed using different computational methods, providing a useful test bed for comparing GFlasso with other methods. Since GFlasso leverages the gene network in eQTL analysis to combine information across correlated traits, it has the potential to achieve greater statistical power and discover relatively weak association signals that were missed in previous analyses. In fact, we found that our analysis of the yeast eQTL dataset using GFlasso provided new insights into the complex interaction between genetic variations and the transcriptome in yeast.

Many of the previous computational analyses of this same dataset reported regions in the genome, coined eQTL hotspots, which control the expression level of gene clusters that are highly enriched for a common function (Zhu et al. 2008; Lee et al. 2009). This suggests a coordinated genetic control of gene modules. Also, by examining the eQTL hotspot regions in the genome, these studies identified candidate regulators whose genetic variations lead to a perturbation of the gene cluster's gene expression levels.

While our GFlasso analysis rediscovered these previously reported eQTL hotspots and their regulators, we identified additional novel eQTL hotspots of biological significance. More interestingly, we found that these novel eQTL hotspots interact with other eQTL hotspots to affect a common set of genes. Although the presence of eQTL hotspots has been reported previously, to our knowledge, our analysis is the first to reveal interactions among multiple eQTL hotspots. We performed in-depth analyses of the three groups of interacting eQTL hotspots that we have uncovered. Based on the shared function of the genes perturbed by the eQTL hotspots in each group, we name the three groups of eQTL hotspots the ribosome biogenesis (Ribi) group, the telomere group, and the retrotransposon group. We suggest candidate regulators for each group of hotspots and provide indirect validation by identifying mutations in the promoter and coding regions of the candidate regulators using the full genome sequences available from public databases. The presence of missense mutations and promoter mutations, combined with the association signals found by GFlasso, provides strong evidence that the candidate regulators are true regulators. As additional evidence, previous studies have shown many of these candidate regulators to be involved in the functional role predicted by GFlasso.

Our in-depth analysis of each group of eQTL hotspots provides new insight into gene regulation in yeast. In our analysis of the Ribi group, we found a coordinated regulation of ribosome biogenesis by multiple genomic loci and identified candidate regulators that affect genes involved in ribosome biogenesis either directly or indirectly. In our analysis of the telomere group, we discovered candidate regulators in multiple genomic regions that likely play a coordinated role in telomere silencing. Finally, in the retrotransposon group, we discovered the coordinated effects of 17 retrotransposon insertions on the resulting expression signal for retrotransposons. The results from our GFlasso analysis provide important insight into future eQTL studies to consider the use of structured association mapping to uncover weak signals, and also to consider interaction among genomic regions when identifying regulatory genes.

4.1.1 The discovery of interacting eQTL hotspots

We applied GFlasso to analyze a genome-wide eQTL dataset generated from a cross between the BY4617 (BY) strain (isogenic to yeast strain S288c) and the vineyard RM11-1a (RM) strain of *Saccharomyces cerevisiae*, baker's yeast (Brem and Kruglyak 2005). The dataset consists of these two parent strains and 112 recombinant progeny. We considered the 1260 unique SNP markers on all 16 chromosomes, which cover nearly the entire genome at a resolution of about 20kb. We used these SNP markers to find associations to the mRNA expression levels for 5637 genes (genes with more than 30% missing values were excluded from analysis).

GFlasso assumes that a network for the gene-expression traits is available as prior knowledge, and GFlasso leverages this network in a structured sparse regression framework to identify associations between genetic loci and multiple traits that are tightly connected in the network. In our preprocessing step, we constructed a scale-free and modular network from the gene-expression data (see Section 4.1.3 for more detail) (Zhang and Horvath 2005; Zhu et al.

2008). We used the resulting topological overlap matrix as our gene-expression network. Once we estimated the parameters for association strengths using GFlasso (see Section 4.1.3), we carried out an in-depth biological analysis.

Gene modules under regulation of common genetic loci

We examined the eQTLs estimated by GFlasso for clusters of genes controlled by common genetic loci. We divided up the genome into 428 genomic regions based on the linkage disequilibrium (LD) between the SNPs in this dataset (see Section 4.1.3). We define an *eQTL module* as all of the genes that map to the same genomic region. We define an *eQTL hotspot* as a genomic region whose eQTL module is greater than 40 genes. An eQTL hotspot and its corresponding eQTL module imply a pleiotropic effect of the genetic locus on co-regulated genes in a common pathway. We note that in our definition of an eQTL module, a gene could be a member of multiple eQTL modules, each associated with different eQTL hotspots. In a

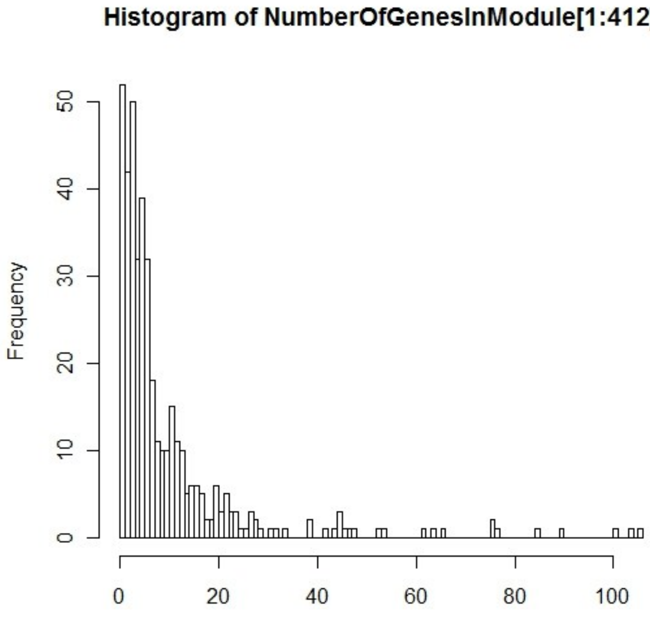


Figure 4.2: Histogram of the number of genes in each of the 428 eQTL modules. We considered the genomic regions corresponding to eQTL modules with more than 40 genes as eQTL hotspots. The largest 13 eQTL modules (with more than 110 genes per eQTL module) are not shown.

biological system, this corresponds to the situation of interacting genetic loci, where the expression of the gene is affected by multiple *trans*-acting loci as well as a possible *cis*-acting locus.

Although we found many eQTL modules, in this study we focus on those eQTL modules with greater than 40 genes, that is, only those that map to eQTL hotspots (Figure 4.2). We present an analysis of 22 such eQTL modules that vary in size from 42 to 722 genes. Ten of the 22 corresponding eQTL hotspots were novel discoveries in this dataset. The other 12 eQTL hotspots overlapped with the 13 eQTL hotspots that had been reported in previous analyses of the same dataset; all 13 previous eQTL hotspots were recovered as two previously discovered eQTL hotspots were combined in our analysis (Yvert et al. 2003; Zhu et al. 2008; Lee et al. 2009).

The common association that an eQTL hotspot has to all the genes in an eQTL module suggests that the region harbors regulators that influence the expression levels of the genes in the eQTL module. We list some candidate regulators located in *cis* to each eQTL hotspot in Table 4.1. All the genes within 20kb from the eQTL hotspot are potential candidates, but because many genes are located in *cis* to each eQTL hotspot, we select transcription factors, genes in the eQTL module, and other genes involved in the pathway of the eQTL module to list here. In Table 4.1, we compare our results with those obtained from a computational analysis using a Bayesian network modeling approach (Zhu et al. 2008), *Lirnet* (Lee et al. 2009), and known and possible regulators based on literature search (Yvert et al. 2003). In general, we found that the results were consistent between GFlasso and previous analyses. For example, in eQTL hotspot 4 located on chromosome III around 200kb, GFlasso found 62 genes in the eQTL module; five of these genes, *MATALPHA1*, *MATALHPA2*, *PHO87*, *BUD5*, and *TAF2*, are located in *cis* to this eQTL hotspot and therefore they are candidate regulators of the eQTL module. Consistent with our results, three previous analyses discussed *MATALPHA1* as a regulator for genes in this eQTL module, and *Lirnet* additionally suggested *MATALPHA2* and *TBK1*. As these candidate regulators lie in *cis* to this eQTL hotspot region, the genetic variation in this region may directly influence the activity or expression of these regulators, which then influence the expression of other genes in the eQTL module.

Table 4.1 - The eQTL hotspots and their candidate regulators from GFlasso and other previous analyses. A * represents a previously discovered eQTL hotspot.

eQTL Hotspot	**eQTL module size**	***cis* genes in eQTL module**	(Yvert et al. 2003)	(Zhu et al. 2008)	(Lee et al. 2009)
***II:380000**	106	*NRG1 TIP1 TAT1 TEC1 ECM33*	none	none	*RDH54 SEC18 SPT7*
***II:560000**	722	*AMN1 MAK5 CNS1 TBs1 TOS1 ARA1 SUP45 CSH1 RPB5 SDS24 ENP1 REI1*	*AMN1 MAK5*	*AMN1 CNS1 TBS1 TOS1 ARA1 SUP45 CSH1*	*AMN1 CNS1 TOS2 ABD1 PRP5 TRS20*
***III:100000**	225	*LEU2 ILV6 NFS1 CIT2 PGS1 RER1 HIS4 FRM2 KCC4*	*LEU2*	*LEU2 ILV6 NFS1 CIT2 MATALPHA1*	*LEU2 ILV6 PGS1*
***III:200000**	62	*MATALPHA1 MATALPHA2 PHO87 BUD5 TAF2*	*MATALPHA1*	*MATALPHA1*	*MATALPHA1 MATALPHA2 TBK1*
IV:1500000	46	*YRF1-1 YDR539W YDR541C*	-	-	-
***V:110000**	45	*URA3 NPP2*	*URA3*	*URA3*	*URA3 NPP2 PAC2*
V:350000	618	*RPS24A RPS8B RTT105*	-	-	-
V:420000	405	*LCP5 NSA2*	-	-	-
V:460000	42	*YER138C UBP5 RTR1*	-	-	-
VII:50000	350	*RAI1 TAD1 KAP114*	-	-	-
***VIII:110000**	147	*GPA1 YAP3 ERG11 LAG1 SHO1 ETP1 YLF2 LEU5*	*GPA1*	*GPA1*	*GPA1 STP2 NEM1*
X:20000	48	*YJL225C VTH2 FSP2 REE1*	-	-	-
XII:610000	53	*TOP3*	-	-	-
***XII:680000**	185	*HAP1 NEJ1 SSP120*	*HAP1*	*HAP1*	*HAP1 NEJ1 GSY2*
XII:780000	44	*REC102 PEX30 FKS1 GAS2*	-	-	-
***XII:1070000**	54	*YRF1-4 YRF1-5 YLR464W YLR462W*	*SIR3*	*YRF1-4 YRF1-5 YLR464W*	*SIR3 HMG2 ECM7*
***XII:70000**	76	*MDM1*	none	none	*ARG81 TAF13 CAC2*
***XIV:490000**	448	*SAL1 TOP2 MKT1 THO2 MSK1 TPM1 LAT1 SWS2*	none	*SAL1 TOP2*	*TOP2 MKT1 MSK1*
XIV:550000	45	*COG6 YIP3 HDA1*	-	-	-
XV:90000	85	*ZEO1 RFC4 HM11 INO4 NDJ1 SKM1 HAL9*	-	-	-
***XV:180000**	406	*PHM7 ATG19 WRS1 RFC4*	none	*PHM7*	*PHM7 ATG19 BRX1*
***XV:59000**	120	*LSC1 YOR131C RAS1 INP53 OST2 PIN2*	*CAT5*	none	*CAT5 ADE2 ORT1*

Multiple interacting eQTL hotspots with pleiotropic effects on common eQTL modules

In order to see if the genes in an eQTL module controlled by an eQTL hotspot share function, we performed a gene ontology (GO) enrichment analysis using the GO Slim annotation from the *Saccharomyces* Genome Database (SGD 2011). As can be seen in Table 4.2, many of the eQTL modules were significantly enriched for common GO categories, indicating the genes in the eQTL module share function. For example, the eQTL module associated with the eQTL hotspot at XII:680kb is enriched for genes annotated to *lipid metabolic process* (GO category, *p*-value = 1.3e-14) in biological process (GO category type), and the eQTL module associated with the

Table 4.2 - GO enrichment analysis of eQTL modules/hotspots found by GFlasso. Previously discovered hotpots are annotated with a *.

Group	eQTL Hotspot	eQTL module size	GO Category	*p*-value	GO size (overlap)	GO Annotation (Zhu et al. 2008)
Ribi Group	***II:560000**	722	nucleolus ribosome biogenesis	1.4e-81 5.4e-62	224 (148) 311 (156)	Cytoplasm organization and biogenesis
	V:350000	618	ribosome biogenesis nucleolus	7.0e-104 5.5e-89	311 (184) 224 (129)	
	V:420000	405	nucleolus Ribosome biogenesis	8.0e-80 6.8e-66	224 (118) 311 (124)	
	VII:50000	350	nucleolus ribosome biogenesis	7.6e-97 5.6e-83	224 (124) 311 (131)	
Telomere Group	IV:150000	46	cellular component unknown helicase activity	6.1e-21 1.8e-17	683 (33) 80 (15)	
	X:20000	48	cellular component unknown helicase activity	1.6e-22 1.4e-15	683 (35) 80 (14)	
	XII:780000	44	helicase activity cellular component unknown	8.3e-10 1.2e-13	80 (15) 683 (26)	
	*XII:1070000	54	helicase activity cellular component unknown	9.3e-19 1.1e-16	80 (15) 683 (27)	none
Retrotransposon Group	V:460000	42	none	-	-	
	*VIII:110000	147	conjugation site of polarized growth	1.6e-11 7.6e-6	119 (20) 211 (18)	conjugation, RNA binding
	XV:90000	85	none	-	-	
Other eQTL modules	*II:380000	106	none	-	-	
	*III:100000	168	cell. amino acid derivative proc transferase activity	6.3e-31 2.6e-7	215 (55) 623 (51)	Organic acid metabolism
	*III:200000	62	conjugation response to chem stimulus	5.1e-7 4.3e-6	119 (10) 400 (16)	Response to chemical stimulus
	*V:110000	45	carbohydrate met. process cell. aromatic comp. met proc	1.2e-4 1.5e-4	249 (9) 65 (5)	
	XII:610000	53	none	-	-	
	*XII:680000	185	lipid metabolic process ER	1.3e-14 3.9e-10	249 (36) 350 (36)	Lipid metabolism, ER
	*XIII:70000	76	cell. amino acid derivative proc	1.2e-4	215 (11)	
	*XIV:450000	448	translation structural molecule activity	2.3e-28 1.4e-15	373 (97) 325 (7)	Protein biosynthesis, intracellular transport
	XIV:550000	45	none	-	-	
	*XV:180000	406	carbohydrate met. process protein modification process	3.5e-5 4.8e-4	249 (36) 544 (21)	Gen of precursor met & energy
	*XV:590000	76	gen of precursor met. & energy mitochondrial envelope	2.6e-35 3.8e-20	168 (42) 82 (37)	Gen of precursor met & energy

XIV:450kb eQTL hotspot is enriched for *translation* (GO Category, *p*-value = 2.3e-28) in biological process (GO category type). For the previously reported eQTL hotspots, our results were consistent with those from previous GO enrichment analysis.

In addition, we performed enrichment analyses on each of the eQTL modules using two knockout datasets (Chua et al. 2006; Hughes et al. 2000) and four transcription factor binding datasets (MacIssac et al. 2006; Zhu et al. 2009; Harbison et al. 2004; Lee et al. 2002). From the transcription factor target enrichment results, we found that genes in each eQTL module involved in the same GO process were also generally enriched for the binding of a common

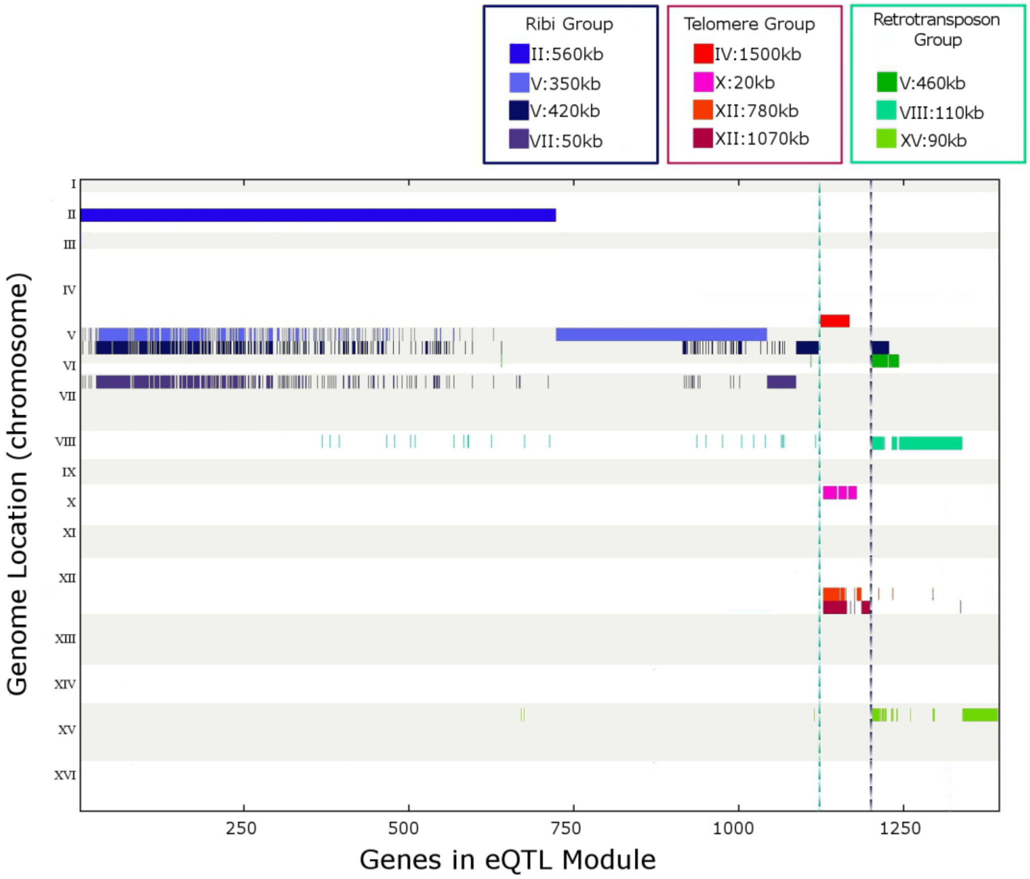


Figure 4.3: eQTL hotspots and their overlapping eQTL modules found by GFlasso. GFlasso found that many genes (*x*-axis) are jointly influenced by the same genetic loci (*y*-axis), suggesting that these eQTL hotspots perturb an overlapping set of genes. We group these eQTL hotspots into the Ribi, telomere, and retrotransposon groups according to their overlaps in the corresponding eQTL modules. For example, the first 722 genes (plotted along the *x*-axis) all belong to the eQTL module for II:560kb, and many of these genes also belong to the eQTL modules derived from three other eQTL hotspots: V:350kb, V:420kb, and VII:50kb.

transcription factor and a knockout perturbation as has been shown before (Zhu et al. 2008).

Interestingly, as we considered the results from the GO enrichment analysis and the transcription factor and knockout analyses, we noticed groups of eQTL modules that were enriched for common GO annotations, transcription factor binding, and knockout signatures. For example, the eQTL modules associated to eQTL hotspots II:560kb, V:350kb, V:420kb, and VII:50kb were all significantly enriched for *ribosome biogenesis* and *PBF2* binding. Furthermore, as shown in Figure 4.3, these four eQTL modules had a large overlap in member

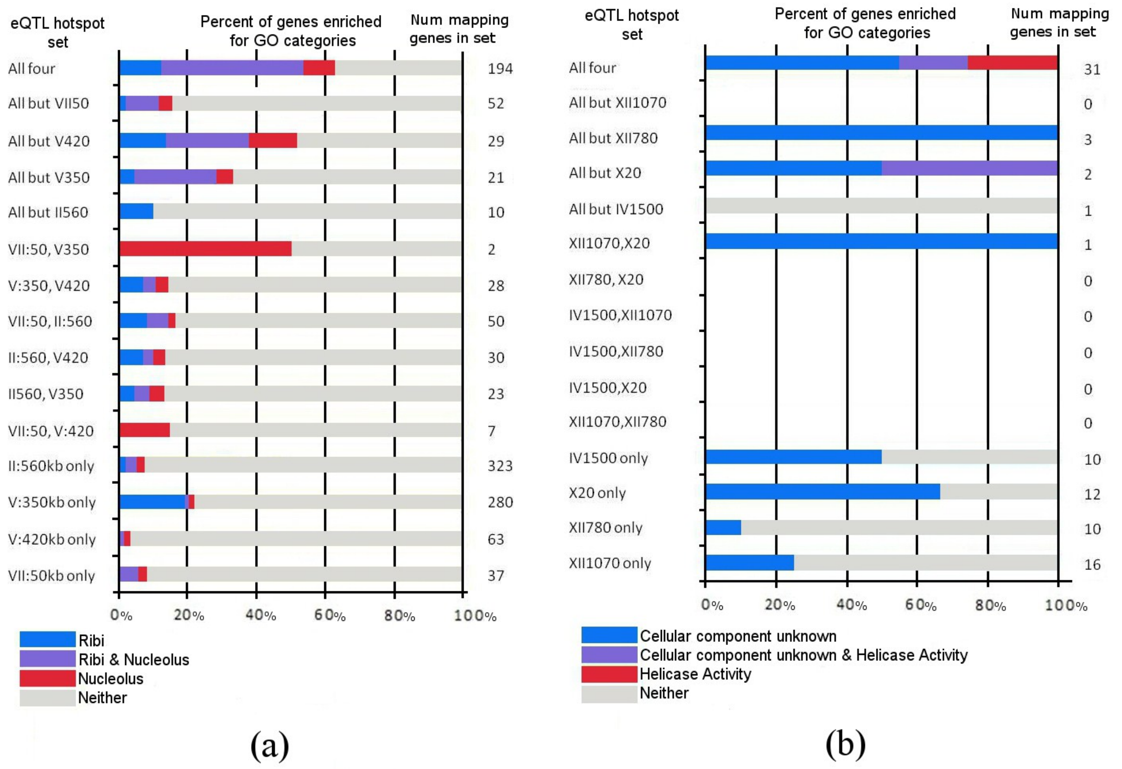


Figure 4.4: GO enrichment analysis for eQTL hotspots with overlapping eQTL modules. For both the (a) Ribi group and the (b) Telomere group, we divide all of the genes in the overlapping eQTL modules into different sets (rows) based on which of the eQTL hotspots in each eQTL hotspot group they are mapped to. Genes that map to all four eQTL hotspots are placed in the set labeled "All four" in the top row. Within each of the sets, we show the percentage of genes mapping to the eQTL hotspot(s) that are annotated to one or both of the top GO enrichment categories for the eQTL hotspot group's eQTL modules along the *x* axis with different colors. In our analysis, we focus our attention on the genes with associations to three or more eQTL hotspots in a group, as they are enriched for a common function.

genes. This suggests that a large number of genes in the overlap of the four eQTL modules are commonly influenced by these different eQTL hotspots. As the eQTL modules share the common GO annotation of *ribosome biogenesis* (Ribi), we call this group of eQTL hotspots the Ribi group (Figure 4.4).

Additionally, we found another group of eQTL modules, associated with eQTL hotspots IV:1500kb, X:20kb, XII:780kb, and XII:1070kb, all significantly enriched for *helicase activity*. Again, the eQTL modules corresponding to these eQTL hotspots shared a large fraction of member genes (Figure 4.3), and many of the genes that are shared by these four eQTL modules were annotated with common GO terms (Figure 4.4). Since our in-depth analysis of this group of eQTL modules revealed that they are involved in telomere activity as we discuss in the next subsection, we name this group of eQTL hotspots the telomere group. As we considered the telomere group and the Ribi group, we noticed that the genes present in many of the overlapping eQTL modules in a group had a higher GO enrichment than genes only present in one or two of the eQTL modules in the group (Figure 4.4). Thus, in subsequent analysis we focus primarily on these genes.

In addition to the Ribi and telomere groups of eQTL hotspots, we searched for other groups of eQTL hotspots with an overlap of more than 20 genes in the corresponding eQTL modules. Using this criterion, we identified one additional group of eQTL modules mapping to eQTL hotspots at V:460kb, VIII:110kb, and XV:90kb (Table 4.2). Although we did not find a common GO enrichment for these eQTL modules, our in-depth functional analysis revealed that many of the genes are involved in retrotransposon biology. Thus, we name this group the retrotransposon group.

We note that only one of the eQTL hotspots associated with each of the Ribi, telomere, and retrotransposon groups has been found in previous analyses of the same dataset, while all of the other eQTL hotspots associated to the groups of eQTL modules are novel discoveries from our GFlasso analysis. Thus, to our knowledge, these interacting eQTL hotspots with a common eQTL module have not been found in any of the previous analyses of this dataset. Thus, our GFlasso and GO enrichment analysis provides new insight into the genetic control of gene expression in a cell, especially the pleiotropic effect of multiple interacting genetic loci.

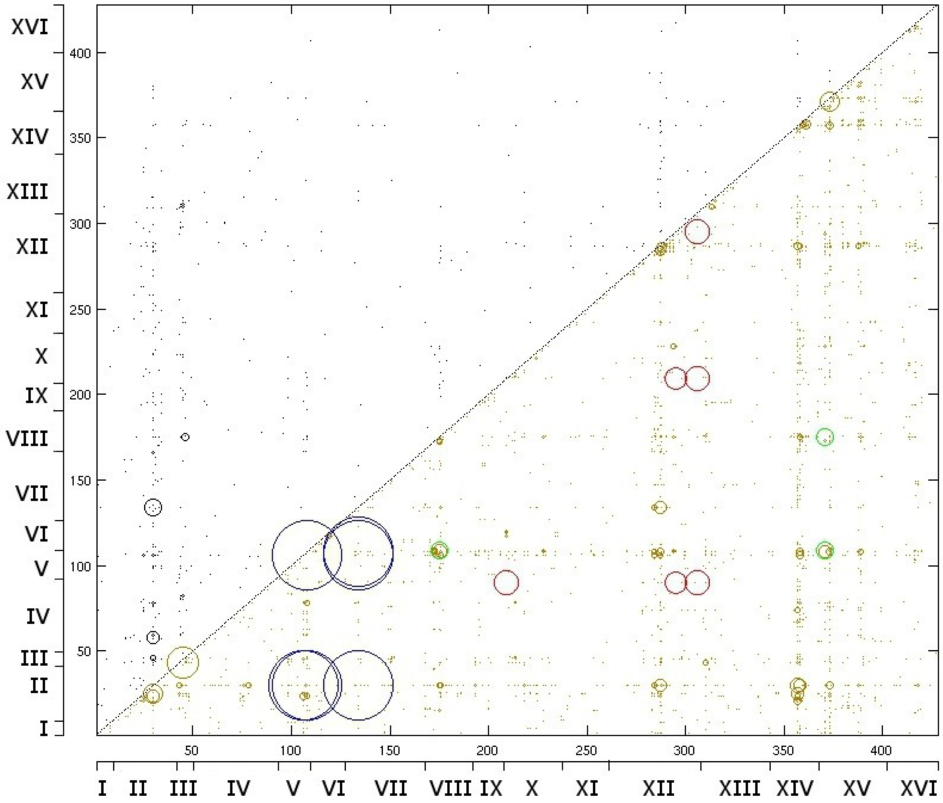


Figure 4.5: Comparison of interacting genome locations found by GFlasso with epistatically interacting SNPs found by an analysis using a previously described non-parametric method (Brem et al. 2005). The results from the nonparametric method are shown on the upper diagonal, and the GFlasso results are shown on the lower diagonal. The size of the circle is proportional to the number of genes in the overlap of the two eQTL modules of the two hotspots. The interactions reported in the main text are colored in blue for the Ribi group, red for the telomere group, and green for the Retrotransposon group. Each genome bin corresponds to one of the 428 eQTL modules found by GFlasso.

We compared the GFlasso results to those from a previous study which looked for epistatic interactions in this dataset (Brem et al. 2005). We replicated the analysis, but found few overlaps between the results from the two analyses; the two studies share only one gene affected by the same two genomic bins (see Figure 4.5). We believe that this difference arises from the different characteristics of the two methods used in the analyses. GFlasso considers the whole gene-expression network to find associations between SNPs and a group of genes with highly correlated expression levels. Meanwhile, the analysis in Brem et al. (2005) examined expression for each gene individually. As GFlasso tends to focus on pleiotropic effects by combining information across multiple genes in the gene network, it found interactions of genomic regions with pleiotropic effects such as the Ribi, telomere, and retrotransposon groups of eQTL hotspots. On the other hand, these signals were missed in the analysis by Brem et al. (2005), which looked for epistatic interactions among loci with effects on individual genes.

Multiple genes in three eQTL hotspots affect Ribi expression levels directly and indirectly

Given the evidence of multiple interacting eQTL hotspots from the GFlasso and enrichment analyses, we performed an in-depth biological analysis of the three groups of eQTL hotspots: the Ribi, telomere, and retrotransposon groups. We determined the functional role that these interacting loci play on the overlapping set of genes and identified potential candidate regulators that these loci harbor. By further examining the DNA sequence of these potential candidate regulators, we found that many of them have missense or promoter mutations between the two strains. The presence of such mutations indicates a potential change of function or expression level. Below, we present our in-depth analysis of the Ribi, telomere, and retrotransposon groups.

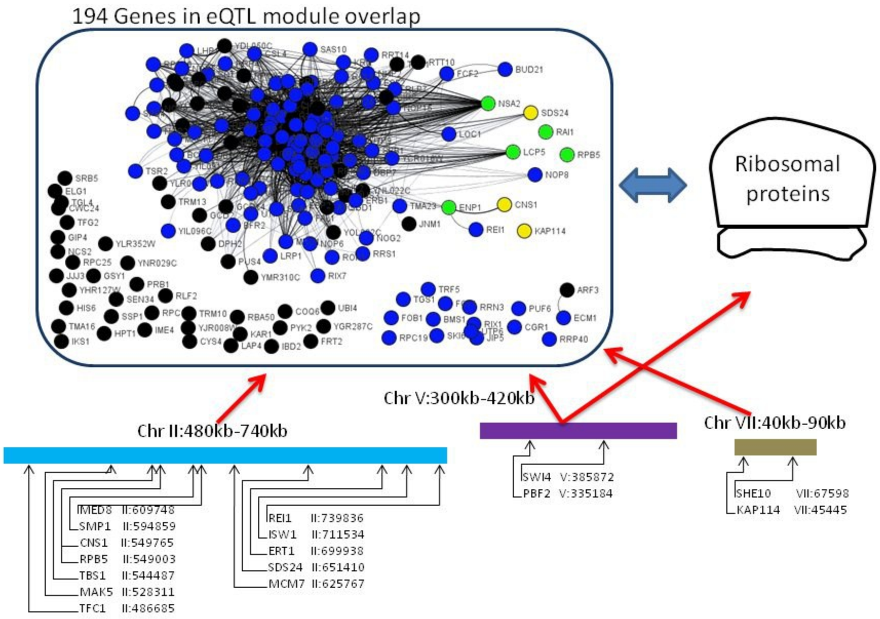


Figure 4.6: An illustration of gene regulation in the Ribi group of eQTL hotspots. We found four eQTL hotspots on chromosomes II, V, and VII that are all associated with the same 194 genes. 122 of the genes in this 194-gene overlap were annotated to the GO category of ribosome biogenesis (Ribi) or nucleolus (shown as blue nodes in the graph). The genes involved in Ribi are generally assembly factors that assemble rRNA and ribosomal proteins into the ribosomal unit in the nucleus. We also found an association from the V:350kb eQTL hotspot to the ribosomal proteins. The expression levels of the ribosomal proteins are tightly coupled with the expression of the Ribi genes. Additionally, we found eight genes (shown as green and yellow nodes in the graph) in the overlap that were located in *cis* to one of these eQTL hotspots. The green nodes represent genes located in *cis* that are annotated for the Ribi or nucleolus GO categories, while the yellow nodes represent genes located in *cis* to one of the eQTL hotspots with a different GO annotation (see Table 4.3). This figure was created using GenAMap.

We first consider the Ribi group, which consists of four eQTL hotspots located on chromosomes II:560kb, V:350kb, V:420kb, and VII:50kb. The corresponding eQTL modules overlap, with 194 genes in the overlap. 122 of these 194 genes have the GO annotation for *ribosome biogenesis* (GO category, type of biological function) and/or *nucleolus* (GO category, type of cellular compartment). We looked at each eQTL hotspot in the group to find mutations that would potentially perturb the expression levels of the genes either directly through transcription or indirectly through a feedback loop. The creation of a Ribi protein is a multi-step process, and there are many steps along the pathway where transcriptional feedback could potentially occur. First the genes encoding the RPs (ribosomal proteins) and Ribi assembly factors must be transcribed, the transcripts translated, and then the proteins imported into the nucleus where the Ribi assembly factors assemble rRNA with the RPs into functional ribosomes.

To limit our search for candidate regulators *cis* to the four eQTL hotspots in the Ribi group, we considered genes that were located in *cis* to one of the four eQTL hotspots and either 1) were also found in the 194 gene overlap, implying an association to all four hotspots (8 overall, shown as green and yellow nodes in Figure 4.6), 2) were also known to be involved in Ribi (*MAK5*, *UTP7*, and *PBF2*), or 3) were annotated as a *DNA binding protein* (12 overall, listed as DNA binding in Table 4.3). We list the candidate genes from our search in Table 4.3. Also, in Figure 4.6 we provide a visual overview of the regulation of these eQTL modules. Although there might be other genes in *cis* to these eQTL hotspots that affect expression levels of the genes in these eQTL modules, we believe our criteria for candidate regulators led us to many of the interesting possibilities. We discuss the candidate regulators for the Ribi group by eQTL hotspot in the paragraphs below.

In both this analysis and the telomere analysis, we examine our candidate regulators by comparing the full coding and promoter sequence for the BY and RM strains. Since only 1260 unique (2956 overall) SNPs were genotyped in the eQTL dataset, these SNPs serve only as genetic markers rather than an exhaustive list of genetic polymorphisms between the two strains. Once the GFlasso analysis points us to the genomic regions or eQTL hotspot around a genetic marker, we can compare the full sequences available from public databases (see Section 4.1.3) to identify missense and promoter mutations.

When we considered the candidate genes located in *cis* to eQTL hotspot V:350kb, we found the transcriptional regulator *PBF2*, which is known to regulate Ribi gene expression; we also noted an association from the V:350kb hotspot to the RPs that could have an indirect effect of Ribi gene expression. Concerning *PBF2,* the Ribi genes are regulated transcriptionally through the *PAC* and *RRPE* promoter motifs, present upstream of most Ribi genes (Hughes et al. 2000). Recently, *PBF2* was shown to bind to the *PAC* motif and to regulate Ribi gene expression (Zhu et al. 2009). *PBF2* is also located in *cis* to the V:350kb eQTL hotspot and has nine missense mutations between the BY and RM strains. *PBF2* is differentially expressed between the progeny that have the BY allele and those that have the RM allele (*p*-value=3.0e-2), although no *cis*-association was found by GFlasso. These mutations suggest that *PBF2* functions differently in the RM and BY strain, influencing Ribi gene expression directly. An additional signal from this eQTL hotspot is the association from V:350kb to the expression levels of 144 of the 330 RP genes. The Ribi assembly factors depend on the presence of RPs to create ribosomes. Thus, it is likely that when the RP level changes in a cell, feedback loops will regulate the Ribi gene level to maintain the ideal ratio of RPs to Ribi genes in the nucleus. Therefore, the perturbation of RP gene expression by a SNP *cis* to eQTL hotspot V:350kb could have an indirect effect on Ribi gene expression.

Located *cis* to the VII:50kb eQTL hotspot, our criteria singled out the *KAP114* gene. *KAP114* is a nuclear importer that could have an indirect effect on Ribi gene transcription. *KAP114* is one of the 194 genes in all four eQTL modules. Nuclear import is an important step in Ribi; the Ribi proteins are translated in the cytoplasm and then *KAP* proteins import them into the nucleus, where they assemble the RPs and rRNA into ribosomes (Sydorskyy et al. 2003). Although *KAP114* has not yet been implicated in Ribi protein import, Ribi proteins and RPs account for most of the incoming nuclear traffic in a cell (Sydorskyy et al. 2003). Therefore, if transport into the nucleus were affected, the rate of Ribi transcription could be affected through a feedback loop.

Table 4.3 - Candidate regulators in the Ribi group of eQTL hotspots

eQTL Hotspot	Candidate	GO Category	Differentially Expressed?	Mutations	Function
II:560kb	TFC1	DNA binding	No	3 missense	RNA Pol III subunit
	MAK5	Ribi	No	5 missense, 4 indel	60S ribosome processing
	TBS1	DNA binding	6.7e-20	7 missense	Unknown
	RPB5	Nucleolus	9.6e-3	promoter has 7 SNPs and 1 indel	RNA Poly subunit
	CNS1	Protein folding	3.6e-9	promoter has 7 SNPs and 1 indel, 1 missense	TPR-containing co-chaperone
	SMP1	DNA binding	2.7e-2	promoter has 7 SNPs and 2 indels	Transcription factor that regulates osmotic stress
	MED8	DNA binding	1.2e-2	Promoter has 2 SNPs, 2 missense	RNA Poly II mediator complex
	MCM7	DNA binding	No	6 missense	DNA ATPase activity
	SDS24	Molecular function unknown	7.2e-7	2 missense, 2 promoter SNPs	Involved in cell separation during budding
	ERT1	DNA binding	2.2e-2	5 missense, 8 SNPs in promoter and a 5 base insertion	Transcriptional regulator of nonfermentable carbon utilization
	THI2	DNA binding	No	None	Zinc finger protein
	ENP1	Ribi	No	None	40S ribosomal subunit synthesis
	ISW1	DNA binding	No	1 missense	ATPase, DNA and nucleosome binding
	REI1	Ribi	No	3 missense	Cytoplasmic pre-60S factor
V:350kb	UTP7	Ribi	No	None	Processing of 18S rRNA
	RAD51	DNA binding	No	None	Strand exchange protein
	PBF2	Ribi	3.0e-2	9 missense	PAC binding factor
	SWI4	DNA binding	No	2 missense	Transcriptional activator
V:420kb	NSA2	Ribi	1.4e-5	None	Constituent of 60S pre-ribosomal particles
	LCP5	Ribi	2.6e-3	1 missense	Involved in maturation of 18S rRNA
	YER130C	DNA binding	No	1 missense	Unknown function
VII:70kb	RAI1	Ribi	No	None	Required for pre-rRNA processing
	RTF1	DNA binding	No	None	Subunit of RNA Pol II
	KAP114	Protein import into nucleus	1.2e-8	9 missense	Karyopherin

Now, we consider the candidates on chromosome II. Due to the large size of the eQTL hotspot on chromosome II:560kb, we had many candidate regulators to consider. Here we report an interesting mutation in the promoter region between *CNS1* and *RPB5*. *CNS1* and *RPB5* are both associated to all four eQTL hotspots in the Ribi group. There are seven SNPs and an indel in the promoter region, which could potentially affect the expression of both genes. *RPB5* is a component of RNA polymerase; a change in expression levels would directly perturb Ribi gene levels. Also, previous computational studies list *CNS1* as a possible Ribi regulator, although its

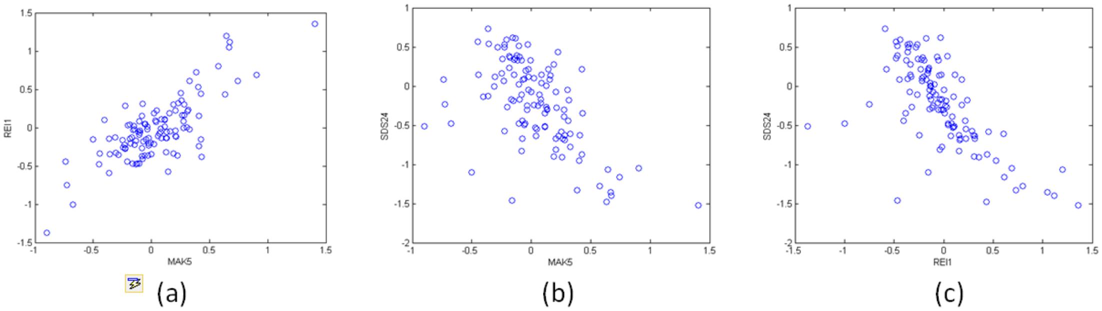


Figure 4.7: Correlation of gene expression among *MAK5*, *REI1*, and *SDS24*. All of the three genes are located in *cis* to eQTL hotspot II:560kb. *MAK5* is located in II:528kb, REI1 in II:740kb, and *SDS24* in II:651kb. (a) *MAK5* is positively correlated with *REI1* (r^2 = 0.7719). (b) *MAK5* is negatively correlated with *SDS24* (r^2 = -0.6569). (c) *REI1* is negatively correlated with *SDS24* (r^2 = -0.5680)

involvement is unclear (Lee et al. 2009; Zhu et al. 2008). As we further considered this eQTL hotspot, we noticed a correlation pattern between the genes located around II:500kb with the genes located around II:650kb, and we found a similar pattern between the genes located around II:700kb with the genes located around II:650kb. For example, *MAK5* (II:528kb) and *SDS24* (II:651kb) are negatively correlated (*r*=-.57). *CNS1* (II:549kb) also follows a similar pattern with *SDS24* (*r*=-.80). *MAK5* and *CNS1* are strongly correlated with *ENP1* and *REI1* (*r*=.80), while *ENP1* (II:712kb) and *REI1* (II:739kb) are also negatively correlated with *SDS24* (see Figure 4.7). These data suggest the presence of some regulator around II:650kb that negatively affects the expression of Ribi genes. This regulator could possibly be *SDS24*, or another gene located close by.

To summarize the in-depth analysis of the Ribi group, we found a complex interaction among genes encoded in the eQTL hotspots. The candidate regulators that we have identified either affect Ribi gene expression directly or could affect transcription levels indirectly through involvement with nuclear import or by influencing RP expression levels. We have also identified other mutations, which GFlasso results suggest affect Ribi expression.

eQTL hotspots harbor mutations in *NUP2*, *RIF2*, and *SIR3* that potentially affect telomere silencing

We now consider the interaction among the four eQTL hotspots in the telomere group: IV:1500kb, X:20kb, XII:780kb, and XII:1070kb. Each of these eQTL hotspot are associated with the same 31 genes, and six additional genes are associated with three of the four eQTL hotspots. All 37 genes lie in telomere regions and three (IV:1500kb, X:20kb, and XII:1070kb) of the four eQTL hotspots also lie in telomere regions. This suggests that there is coordinated or interacting regulation among these four eQTL hotspots to turn on the expression of telomere genes. GO enrichment analysis found that these genes are enriched for the GO functional annotation "helicase activity" (*p*-value = 2e-17) and the GO component annotation "cellular component unknown" (*p*-value = 2e-17). This suggests that these genes share common function and interact with DNA as a helicase. Because many of the genes do not have a cellular component annotation, they might be understudied or not ordinarily expressed.

We considered the known function of each gene in the set individually and found seven yeast *YRF1* genes (*YRF1-1*, *YRF1-2*, *YRF1-3*, *YRF1-4*, *YRF1-5*, *YRF1-6*, and *YRF1-7*). We also found that the other 30 genes were all either "proteins of unknown function," or "helicase-like proteins encoded within the telomeric Y' element" (SGD, 2011). Because the annotations were common across the genes in the set, we performed a sequence BLAST against *YRF1-1* and found that 36 of the 37 genes had high homology to the *YRF1-1* transcript (BLAST eValue < 1e-36). The 29 non-*YRF1* genes had no known functionality, despite their homology to the *YRF1* genes. We conclude that these genes are copies of *YRF1* in the yeast telomeres. The homology also suggests that these genes cross-hybridize to each other's probes on the microarray; if any one of the genes regulated by these four eQTL hotspots is expressed, all of the genes would appear to be expressed on the microarray. The homology of this module has been previously observed (Zhu et al. 2008), however, in the RM wild type strain, the *YRF1* genes have a significantly lower expression level than in the BY mutant strain (*t*-test *p*-value = 1.1789e-26), which cannot be explained by the hypothesis of cross-hybridization. It appears that there is some kind of regulation turning on, or failing to turn off, at least one of the *YRF1* genes in the BY strain.

In order to explain the difference in *YRF1* and *YRF1*-like gene expression between the RM and BY strains, we considered what is known about the *YRF1* genes. The *YRF1* genes are known to be a backup plan for telomerase. Telomerase is the protein complex essential for maintaining telomere length (Cohn and Blackburn 1995). The loss of telomerase results in the gradual shortening of the telomeres and in eventual cell arrest, unless the telomeres are lengthened through the *YRF1* pathway (Straatman and Louis 2007). There are many copies of *YRF1* located in yeast telomeres (Yamada et al. 1998). *YRF1* genes are not expressed in wild type cells, probably due to telomere silencing. However, it appears that as the telomeres shorten, silencing information is removed, leading to the expression of the *YRF1* genes (Yamada et al. 1998). The *YRF1* genes contain several helicase motifs and are believed to extend the telomeres through DNA homologous recombination, largely because other helicases participate in homologous recombination and genes important in homologous recombination are essential for survival without the proper function of telomerase (Yamada et al. 1998).

Therefore, one possibility to explain the expression of *YRF1* is impaired telomerase function. Telomerase is made up of five proteins, two of which are functionally essential: *TLC1* and *EST2* (Yamada et al. 1998). *EST2*, a reverse transcriptase, is located in *cis* to the eQTL hotspot at XII:780kb in the telomere group. We considered the sequence of the RM11-1a strain against the S288c strain and found five missense mutations in the *EST2* transcript (*EST2* was not differentially expressed between the two strains, *p*-value > .2), including an R to Q mutation in the reverse transcriptase domain. These mutations suggest that *EST2* could be impaired in its function, allowing for the shortening of telomeres and the activation of the *YRF1* genes. However, we conclude that the loss of *EST2* function is unlikely, given the popularity of the S288c strain and its use as a "normal" control for functional telomerase in yeast telomere studies (Straatman and Louis 2007). We therefore suggest other pathways that potentially regulate *YRF1* gene expression.

Another possibility to explain the *YRF1* gene expression is the loss of telomere silencing genes. *NUP2* (XII:780kb) and *RIF2*/*SIR3* (XII:1070kb), are telomere silencing genes *cis* to hotspots in the telomere group. Dilworth et al. (2005) report that *NUP2* (part of the nuclear pore complex) localizes in the nucleus with yeast telomeres; ChIP-chip experiments reveal that *NUP2* has a telomere binding preference. These results, combined with the association found by the GFlasso, suggest that *NUP2* (seven missense mutations) is a player in telomere silencing. In regards to *RIF2, RAP1* is also known to be involved in telomere silencing (Shore 1997), and in this role it is assisted by *RIF2*; deletions in *RIF2* affect telomere length (Teixeira et al. 2004). *SIR3* is also recruited to the telomere regions by *RAP1*, and cells lacking telomerase have increased concentration levels of the *SIR3* protein (Straatman and Louis 2007). *SIR3* (twelve missense mutations) and *RIF2* (six missense mutations) are both located in *cis* to the XII:1070kb eQTL hotspot. Previous computational analyses of this dataset have similarly implicated *RIF2* as a possible regulator of telomere genes (Lee et al. 2006).

A final possibility to explain the *YRF1* gene expression is the loss of silencing sequence in the DNA, leading to the expression of a *YRF1* transcript. *YRF1-1* (IV:1500kb) and *YJL225C* (X:20kb) both have mutations that could have this effect. *YRF1-1* has a five-base indel located

186 bases upstream in its promoter region, and *YJL225C* has a ten-base indel located 237 bases upstream in its promoter region. These mutations could potentially remove silencing information for these genes; the genotype at the eQTL hotspot is indicative of the expression level in both cases (*p*-value = 1e-4 and 1e-6 respectively).

In conclusion, we investigated three possibilities where mutations in the BY strain could lead to the expression of at least one *YRF1* gene transcript. We suggest that telomerase is not impaired in the BY strain, however, mutations in telomere silencing genes and in promoter regions of *YRF1* genes are likely candidates that may work together or in parallel to either turn on or silence *YRF1* gene expression.

Table 4.4 - Candidate regulators in the telomere group of eQTL hotspots.

eQTL Hotspot	Candidate	Differentially Expressed?	Mutations	Function
IV:1500kb	*YRF1-1*	6.4e-6	5 base insertion 186 nt upstream	*YRF1* gene
X:20kb	*YJL225C*	1.1e-4	10 base insertion 237nt upstream	*YRF1*-like gene
XII:780kb	*EST2*	No	5 missense	Telomerase component, reverse transcriptase
	NUP2	No	7 missense	Nuclear pore protein involved in telomere silencing
XII:1070kb	*SIR3*	5.3e-3	12 missense	Involved in telomere silencing
	RIF2	No	6 missense	Involved in telomere silencing

GFlasso uncovers 17 retrotransposon insertions

The retrotransposon group of eQTL hotspots are located on V:460kb, VIII:110kb, and XV:90kb. The eQTL modules for each of these eQTL hotspots differ in size (42, 85, and 147 genes), although they all influence a common set of 20 genes, with 35 genes associated with two of the three eQTL hotspots. Although the VIII:110kb eQTL module is enriched for conjugation (due to its close proximity with the mutated *GPA1* gene (Yvert et al. 2003)), neither of the other two eQTL modules are enriched for a GO category. From our analysis, we found that the nonoverlapping genes in these eQTL modules are not related to the genes involved in the overlap of the eQTL modules.

When we considered each of the 35 genes, we found that 15 of the genes are highly homologous (BLAST score of less than 1e-200 when queried against each other) and are annotated as Ty retrotransposons in the SGD database. We investigated the significance of finding 15 retrotransposons in the same eQTL module. A computational study (Kim et al. 1998) identified 331 retrotransposons in the yeast genome, 94 of which correspond to retrotransposon genes listed in the SGD. Of these 94 genes, 21 are included in the Brem and Kruglyak (2005) dataset; it is unlikely that 15 of these 21 genes would end up in the same eQTL module of size 35 (*p*-value=1.27e-30).

In yeast there are five types of retrotransposons, referred to as the Ty genes: Ty1, Ty2, Ty3, Ty4, and Ty5, with Ty1 being the most frequent in the genome (Kim et al. 1998). Retrotransposons are scattered throughout the genomes of eukaryotes and function like a virus that is transcribed into an mRNA intermediate. This mRNA intermediate, through a reverse transcriptase, is then inserted back into the genome as cDNA, playing an important role in genome evolution (Kim et al. 1998; Boeke and Sandmeyer 1991). It is estimated that Ty retrotransposon mRNA accounts for about 1% of the total mRNA in a cell. However an insertion into the chromosomal DNA only happens between 10^{-7} and 10^{-8} times per cell division cycle, suggesting that the insertion of the Ty genes is regulated post-transcriptionally (Krastanova et al. 2005; Curcio et al. 1990). The transcriptional regulation of the Ty genes happens through the TATA-box and other information in the promoter, and has been linked to the suppressor of transposition (*SPT*) genes and the *STE* genes (Krastanova et al. 2005).

Due to the sequence homology of these genes, it is probable that the observed co-expression is a result of cross-hybridization. However, we were interested to find genes in the three associated eQTL hotspots that could account for the transcriptional diversity between the strains. We found a few candidate *STP* and *STE* genes based solely on location, *SPT15* at V:464kb and *STE20* at VIII:94kb. However, sequence analysis revealed that *SPT15* is perfectly conserved between the RM and the BY strain. *STE20* had eight missense mutations and a few SNPs in its promoter region.

Interestingly, we found a retrotransposon located in *cis* to each of the three eQTL hotspots; each homologous to the 15 retrotransposons discovered in the eQTL module overlap. These three retrotransposons, *YER138C* (V:449kb), *YHL009W-B* (VIII:85kb), and *YOL104W-A* (XV:118kb) are present in the S288c strain, but not in the RM strain sequence. This could be due to errors in the assembly of the RM sequence, but it likely that the insertion happened after these two strains diverged. We additionally considered each of the 15 Ty1 genes in the eQTL module overlap among these three interacting eQTL hotspots and found that only one Ty1 gene was present in both strains. Thus, we have found 17 total (14 in the dataset and 3 in *cis* to eQTL hotspots) retrotransposon insertions between the BY and RM strain, leaving open the possibility for other insertions as well. Additionally, among the 20 genes in the overlapping set of 35 that were not retrotransposons, we found that 13 of them were within 10kb of a retrotransposon site and therefore could be expressed differently between the two strains because of the retrotransposon.

GFlasso has therefore uncovered a case where retrotransposon insertions have occurred since the BY and RM strain diverged. The occurrence of such retrotransposon insertion events in separate populations is not surprising and can be found by the direct comparison of the genome sequences of the two yeast strains. However, our analysis shows that in the absence of the full genome sequence information, GFlasso has the potential to discover systematic sequence differences such as gene insertions by investigating their impact on the expression levels solely based on an eQTL dataset.

4.1.2 Discussion of yeast results

Many of the previous methods for discovering eQTLs from genotype and gene-expression datasets have been concerned with testing the hypothesis of association between an individual genotype and the expression of each gene (Gilad et al. 2008). However, there is a great deal of evidence that the elements in the genome and transcriptome interact with each other in performing a biological function; it has been widely recognized that the computational methods for detecting eQTLs should take into account this complex interaction pattern. Although there have been attempts towards this goal (Zhu et al. 2008), GFlasso is the only computational method that directly maps the quantitative-trait (gene-expression) network to genotypes, explicitly combining information across multiple correlated traits to increase the power of detecting association. In this study, we re-analyzed the eQTL dataset (Brem and Kruglyak 2005) from the genetic cross of two yeast strains (BY and RM) using GFlasso and discussed the new insights into yeast gene regulation that were provided by our analysis. The yeast eQTL dataset provides an excellent test-bed for comparing various computational methods, as it has been extensively analyzed. We showed that GFlasso led to significant biological findings that had not been discovered by other methods, and we demonstrated the potential of GFlasso for future analyses of eQTL datasets that are becoming available for various organisms, tissue types, and diseases.

While the pleiotropic control of multiple genes by a genetic locus, called an eQTL hotspot, has been previously reported in analyses of many different eQTL datasets, our GFlasso analysis of the yeast eQTL dataset revealed another layer of complexity in gene regulation by uncovering the pleiotropic effect of multiple genetic loci on multiple genes. The literature has yet to report this type of pleiotropic effects of multiple interacting genetic loci. Although our analysis in this study was focused on yeast, we suspect that the pleiotropic regulation of genes by multiple interacting eQTL hotspots is commonplace in many other eQTL datasets. Our results show that it may be worthwhile to revisit eQTL datasets with this new perspective of interacting eQTL hotspots, especially as more powerful computational methods become available. Furthermore, our results demonstrate the advantages of using structured association mapping in future studies to uncover weak signals and also of considering interaction among genomic regions when identifying regulatory genes.

Our close investigation of the three groups of interacting eQTL hotspots that control an overlapping set of genes led to new insights into Ribi gene regulation, telomere silencing, and retrotransposon activity and suggested potential regulators. By identifying missense and promoter mutations in the full DNA sequence of the candidate regulators, we provided strong evidence that these candidate regulators influence the gene expression levels of many genes in these biological pathways. In addition, we showed that prior studies of these individual genes in the literature support many of the hypotheses that the candidate regulators have the functional role suggested by our analysis. As yeast is one of the model organisms that have been studied extensively, a plethora of information is already available. This information includes the full genome sequence as well as detailed investigations of many of the genes; we were able to confirm the results of our analysis by comparing the results with this information. For many complex diseases, in other organisms where the same kind of extensive knowledge base is not yet available, we expect GFlasso to serve as a powerful computational tool for new discoveries.

Our in-depth analyses of the three groups of interacting eQTL hotspots opens up many research questions on the regulation of Ribi genes and telomere silencing that need to be further investigated in follow-up studies. Although GFlasso analysis suggests that the candidate regulators on different eQTL hotspots affect the same set of genes in a coordinated manner, understanding the exact mechanism of such coordination would require further research. For example, *NUP2* and *SIR3* on two different loci in the telomere group of eQTL hotspots have been found to regulate telomere silencing in both GFlasso analysis and previous studies of these genes. However, exactly how this interaction between these two genes occurs remains unexplored. This novel interaction could lead to further insight into how *NUP2* is involved in telomere silencing and perhaps uncover further interactions between various genetic loci that turn genes on and off.

Finally, GFlasso is designed to identify additive effects of multiple genetic loci on correlated traits, and thus, the effects of multiple eQTL hotspots on each gene were also additive. An interesting future research direction would be to consider epistatic interactions among multiple eQTL hotspots, where the effect of a given eQTL hotspot is not independent of the genotypes of other eQTL hotspots. As detecting epistatic effects on individual genes is widely known as a computationally intensive task, the more challenging problem of detecting epistatic effects on multiple genes with pleiotropic effects would require a significant advance in computational tools.

4.1.3 Methods used in yeast study

Preprocessing yeast eQTL dataset

We used the expression and gene expression data collected from 112 segregants (114 strains) from a cross between BY4716 and RM11-1a (Brem and Kruglyak 2005). The dataset has 2956 SNPs across the 16 chromosomes of the yeast genome. Additionally, for each segregant, microarray expression data are available for 6216 genes. After pre-processing the microarray data with standard procedures, we discarded the genes whose expression values are missing for more than 30% of the strains, and used the gene expression values for the other 5637 genes in our GFlasso analysis. We imputed the missing values for these genes using *k*-nearest-neighbor imputation method (Troyanskaya et al. 2001).

In the genotype data, we found that many of the adjacent SNPs are in complete linkage disequilibrium (LD) across all 114 strains. After discarding those adjacent SNPs with perfect correlations, we were left with 1260 SNPs, which we used in our GFlasso analysis.

Creating a network from gene expression data

GFlasso (Kim and Xing 2009) takes a gene-interaction network as input, along with the genotype and gene-expression data, and performs a correlated association analysis to identify genomic regions that perturb correlated traits in the network. In order to obtain a gene-interaction network to use as an input to GFlasso, we used the algorithm for learning a topological overlap matrix as described in Zhang & Horvath (2005) that was applied to the same dataset in Zhu et al. (2008). The resulting network has the properties of being modular and scale free with a few hub genes having high connectivity and controlling many other genes. Genes that appear correlated in this network often share common functionality and therefore are likely to be regulated by the same regions of the genome (Zhu et al. 2008).

More specifically, we applied the soft-thresholding methodology (Zhang and Horvath 2005) to gene expression data to learn the gene interaction network that can be used in GFlasso (Kim and Xing 2009). Below, we provide the details of the procedure for network generation.

1. We computed a correlation matrix of size 5637x5637 for 5637 genes, and took the absolute values of its elements.
2. For different values of η ranging from 1 to 20, we raised each value in the matrix element-wise to the η power and obtained 20 different matrices. For each of the 20 matrices, we perform the following procedure. We regressed $\log(p(k))$ on $\log(k)$ to find r^2, where k is the number of neighbors (degree) of each gene (calculated as the sum over all columns for each row for the gene of the matrix), and $p(k)$ is the degree distribution of the network. $p(k)$ for each gene was found empirically by creating a 50-bin histogram for all genes in the network based on k, and then dividing the number of genes in the same bin as the gene in question by the total number of genes.
3. Out of the 20 networks, we selected the scale-free network corresponding to the lowest value of η that gives r^2>.8. This was obtained at η =5, which is consistent with the previous reports from using the same method on these data (Zhu et al. 2008). Similar to the previous application of this method, we found that the network degree distribution is described as $p(k)\sim k^{1.47}$ and r^2=.84.
4. We used the scale-free network to create a topological overlap matrix (TOM). After setting all diagonal elements to zero, a TOM was found by calculating ω for each element in the matrix as previously described (Zhang and Horvath 2005). We multiplied each entry in the TOM by its original sign in the correlation matrix to preserve positive and negative edge weights.

5. We discarded edges for weak interactions by thresholding the TOM at 0.10, since these weak edges only add to the computational cost of GFlasso without increasing the power of detecting association.

GenAMap

We used the processing pipeline that is implemented in GenAMap (Section 2.2.4) to run GFlasso on the yeast data to procure the results. These steps involved finding connected components and using spectral clustering to separate the genes into parallel runs for GFlasso. In contrast to the proximal gradient method used in Section 2.2.4, we use a coordinate descent approach to optimize GFlasso given regularization parameters (Kim and Xing 2009).

Identifying mutations between the strains

In order to identify the genotypic differences in genes across two strains, we downloaded the RM11-1a sequence for each protein of interest from the *Saccharomyces cerevisiae* RM11-1a sequencing project website (Broad Institute of Harvard and MIT 2011) and used it as the query for BLASTp search (NCBI 2011), limiting the results to the *Saccharomyces cerevisiae* S288c strain. The full protein sequences that we obtained as results for each query were used to identify mutations between the two strains. In the cases of promoter mutations, we took the 500 bases upstream of the gene in the RM sequence and performed a BLASTn search (NCBI 2011) in the BY strain.

SNP LD analysis

In order to interpret the results from GFlasso, we group SNPs in high LD into bins, and group the association results from GFlasso analysis according to these SNP bins. In other words, within each bin of SNPs in high LD, we treat the associations from each member SNP to any phenotypes as an association to the common genomic locus.

We applied the greedy algorithm for tag SNP selection to group the SNPs (Carlson et al. 2004). We first created a matrix of size 1260x1260, whose element contains the r^2 LD statistic between SNPs, and set all elements with r^2<0.80 to zero, since we considered only those SNP pairs with r^2>0.80 as in high LD. Then, we applied the following iterative procedure.

1. Find the pair of SNPs with the highest LD by selecting the element with the highest value in the matrix. We call the selected two SNPs “founders”.
2. Combine the founder SNPs and the SNPs located between the founder SNPs to create a new bin.
3. Expand the bin (in Step 2) by adding all of adjacent SNPs that are in high LD (r^2 > 0.8) with the founder SNPs.
4. Repeat Steps 1-3 until all SNPs have been assigned to a bin or form a bin of a single SNP with no significant LD with other SNPs.

eQTL Module creation and analysis

We created eQTL modules for each of the 428 bins by finding all genes in the ***B*** matrix that had a nonzero value for any SNP in the bin. We then picked the largest, most interesting bins for further analysis. We left out contiguous bins that were mapped to the same genes as a neighboring bin we had already chosen. For bins with eQTL modules larger than 40 genes, we call them *eQTL hotspots*.

For each bin, we performed enrichment tests using the GO Slim annotation (http://www.yeastgenome.org/). We also performed enrichment tests using two knockout datasets and four transcription factor (TF) binding datasets. We used Fisher’s exact test to perform all of our enrichment calculations.

We performed knock-out enrichment analyses using two knockout datasets. The first knockout data set that we used is the yeast compendium data set (Hughes et al. 2000). We followed the procedure used by Zhu et al (2008) to preprocess the data. The second knockout data set that we used is from Chua et al (2006). In our preprocessing of this dataset, we considered a gene to be affected by a knockout if its *z*-score was greater than 2.58 or less than -2.58.

We used four different TF datasets to test for TF enrichment. The first data that we used was a ChIP-on-chip dataset (MacIssac et al. 2006). We used the results from this dataset that corresponded to a threshold of .005 and a stringent cutoff. The second dataset that we used is from Lee et al (2002), and the third data set is from Harbison et al (Harbison et al. 2004). For both of these datasets we used the conservative threshold of .001 to indicate a binding preference. Finally, we performed enrichment tests using the Protein Binding array results from the Supplementary material of Zhu et al. (2009). We used the top 200 genes reported for each TF for our enrichment analysis.

We found that the same patterns in the TF and knockout datasets that we found in the GO enrichment analysis. eQTL modules that overlapped were enriched for binding by the same TFs, or enriched for perturbation by the same knockout genes. eQTL modules mapping to eQTL hotspots were not just enriched for common function (GO category), but were also enriched for TF binding to a particular factor and/or knockout-signature of a particular gene.

4.2 GenAMap analysis on SARP and CSGA asthma data uncovers new asthma gene

In this section I use GenAMap to analyze a human dataset collected for the study of asthma. The data were collected through the Severe Asthma Research Program (SARP) and the Cooperative

Study for the Genetics of Asthma (CGSA). This data set is an excellent dataset for demonstrating GenAMap because it complements the analysis of the yeast data, showing the versatility of GenAMap to perform analyses appropriate for different datasets from different species. Additionally, this dataset has been collected from individuals in two distinct populations, a non-Hispanic white and an African American population, allowing me to show analyses in GenAMap that use population structure. Through my analysis of the asthma data using GenAMap, I have uncovered a potentially new asthma gene, which is undergoing further validation.

The combination of the SARP and CGSA datasets created a dataset with 1745 individuals genotyped for 752256 SNPs. In addition to case/control assignment, 18 other clinical traits are available. Due to implementation constraints, GenAMap cannot import datasets of larger than 5000 SNPs for 2000 individuals. As a preprocessing step to find potentially informative SNPs, I used PLINK (Purcell et al. 2007) to run the chi-square test on each of the SNPs against the asthma phenotype (cases/controls) in each population. I chose a significance cut-off at 2.5e-3 to select 3785 SNPs that were associated with asthma in one of the two populations (three SNPs were associated with asthma in both populations at this significance level). These 3785 SNPs and the 19 traits were imported into GenAMap for analysis.

4.2.1 Assessing population structure

The first step I took upon importing the data into GenAMap was to explore the population structure of the data. I compared race assignments for the individuals found in two ways: 1) self-reported race and 2) race assignments generated from running Structure (Pritchard et al. 2000) (with GenAMap) on the data assuming two populations. In the population view, GenAMap plots individuals by Eigenvalue; the analyst can explore 2D plots for the first five Eigenvalues. I found that the individuals separate into two distinct populations (Figure 4.8). I also found that the

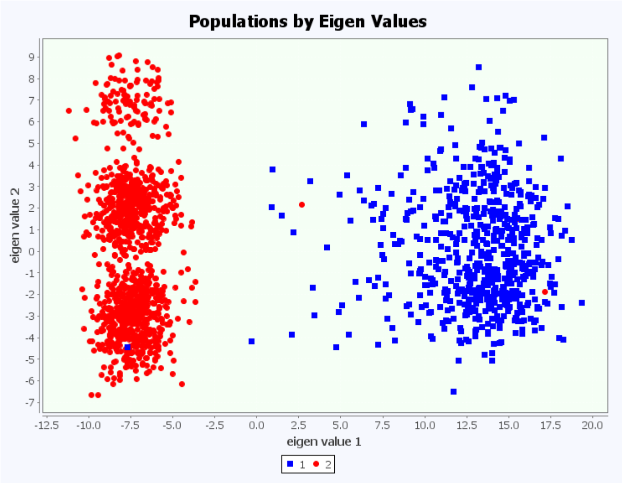


Figure 4.8: Analyzing population structure in GenAMap. I used GenAMap to explore population structure and self-reported race. Population assignments are plotted by individual by Eigenvalue. In this plot, self-reported race is plotted according to the first two Eigenvalues. The plot shows clear separation between the populations.

population assignments made by running Structure strongly agreed with self-reported race (less than 10 differences in classification).

4.2.2 Exploring association by population

Given the strong separation between populations, I chose to perform simple, baseline association analyses that find associations separately in each population. GenAMap provides four simple statistics to explore associations by population (Curtis et al. 2011). These four analyses are automated and run in parallel in GenAMap. The four analyses are 1) the Wald (qualitative traits)

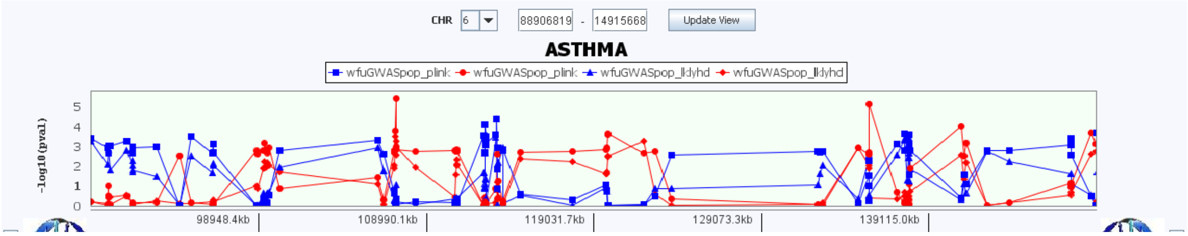
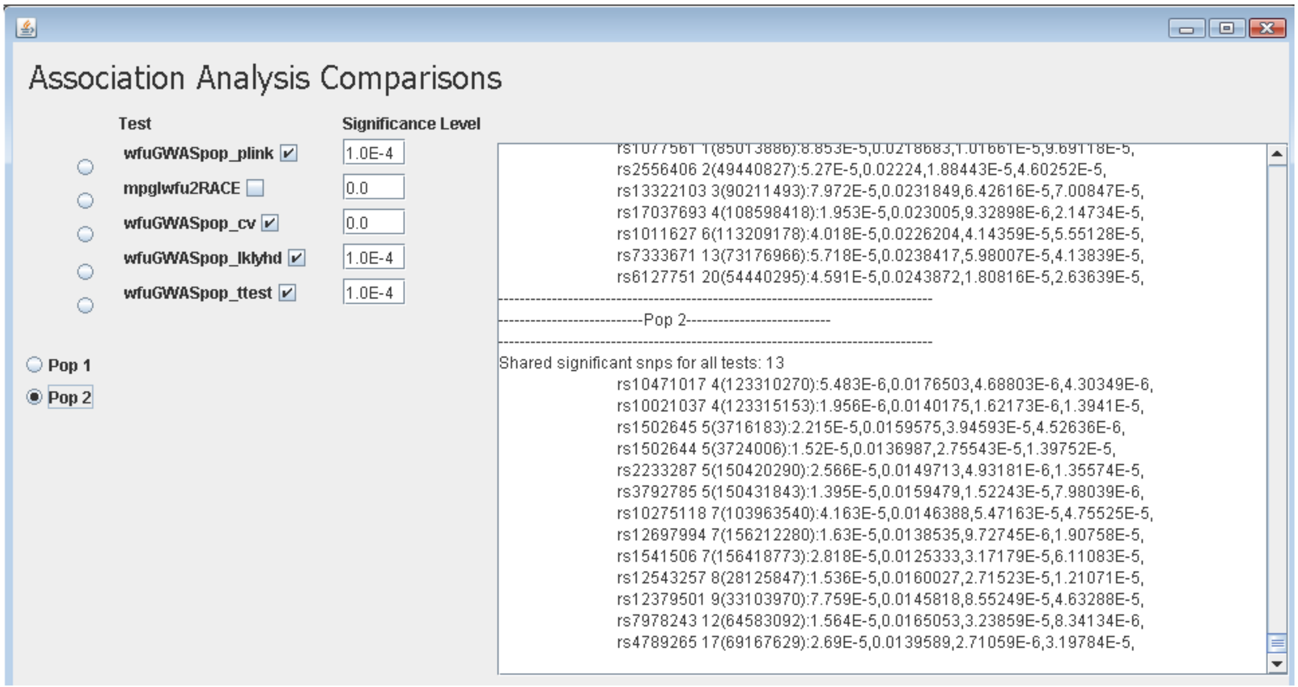


Figure 4.9: Interactive Manhattan plot for population data. I show the results of two tests looking for genetic associations to asthma in GenAMap. The blue lines represent population 1 (African American population) and the red lines represent population 2 (non-Hispanic white population). Different tests are represented by different shapes in the plot.

or chi-squared likelihood (binary traits) test as implemented by PLINK, 2) a two-sided t-test on the phenotype distribution by genotype, 3) a likelihood test (Wu et al. 2010), and 4) a cross-ten validation score by linear regression.

In Figure 4.9, I present an overview of the results from running this test on the asthma data. Figure 4.9 shows a region on chromosome 6 where many associations were found to the trait "Asthma" in the data. In this interactive Manhattan plot, I can add and remove tests, and I can also add and remove populations. For example, in Figure 4.9 I show the results from the PLINK and likelihood tests for both populations. From the plot, it is readily observable that the two tests found the same associations at approximately the same level of significance. I also recognized the interesting pattern that the SNPs associated with asthma in the African American



Association Analysis Comparisons

	Test	Significance Level
○	wfuGWASpop_plink ☑	1.0E-4
○	mpglwfu2RACE □	0.0
○	wfuGWASpop_cv ☑	0.0
○	wfuGWASpop_lklyhd ☑	1.0E-4
○	wfuGWASpop_ttest ☑	1.0E-4

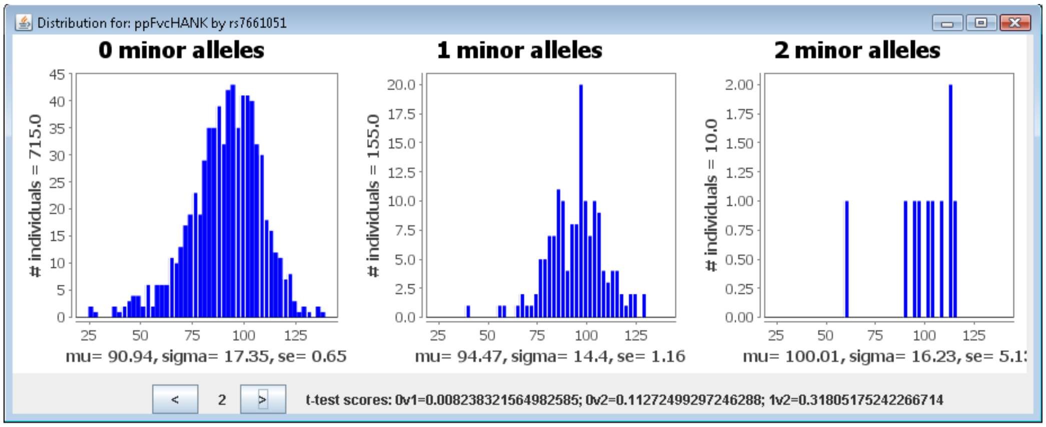
○ Pop 1
● Pop 2

```
                rs1077561 1(85013886):8.853E-5,0.0218683,1.01661E-5,9.69118E-5,
                rs2556406 2(49440827):5.27E-5,0.02224,1.88443E-5,4.60252E-5,
                rs13322103 3(90211493):7.972E-5,0.0231849,6.42616E-5,7.00847E-5,
                rs17037693 4(108598418):1.953E-5,0.023005,9.32898E-6,2.14734E-5,
                rs1011627 6(113209178):4.018E-5,0.0226204,4.14359E-5,5.55128E-5,
                rs7333671 13(73176966):5.718E-5,0.0238417,5.98007E-5,4.13839E-5,
                rs6127751 20(54440295):4.591E-5,0.0243872,1.80816E-5,2.63639E-5,
-----------------------------------------------------------------------------
-------------------------Pop 2-------------------------
-----------------------------------------------------------------------------
Shared significant snps for all tests: 13
                rs10471017 4(123310270):5.483E-6,0.0176503,4.68803E-6,4.30349E-6,
                rs10021037 4(123315153):1.956E-6,0.0140175,1.62173E-6,1.3941E-5,
                rs1502645 5(3716183):2.215E-5,0.0159575,3.94593E-5,4.52636E-6,
                rs1502644 5(3724006):1.52E-5,0.0136987,2.75543E-5,1.39752E-5,
                rs2233287 5(150420290):2.566E-5,0.0149713,4.93181E-6,1.35574E-5,
                rs3792785 5(150431843):1.395E-5,0.0159479,1.52243E-5,7.98039E-6,
                rs10275118 7(103963540):4.163E-5,0.0146388,5.47163E-5,4.75525E-5,
                rs12697994 7(156212280):1.63E-5,0.0138535,9.72745E-6,1.90758E-5,
                rs1541506 7(156418773):2.818E-5,0.0125333,3.17179E-5,6.11083E-5,
                rs12543257 8(28125847):1.536E-5,0.0160027,2.71523E-5,1.21071E-5,
                rs12379501 9(33103970):7.759E-5,0.0145818,8.55249E-5,4.63288E-5,
                rs7978243 12(64583092):1.564E-5,0.0165053,3.23859E-5,8.34134E-6,
                rs4789265 17(69167629):2.69E-5,0.0139589,2.71059E-6,3.19784E-5,
```

Figure 4.11: Comparing association tests. GenAMap allows analysts to compare association results across tests. In this dynamic query tool, the analyst can select which tests to include in the comparison and the significance level of each test. They can see which SNP-trait associations are significant across all tests, and also which associations are unique to any given test. Here, I list 13 SNPs that had significant association to asthma across all four tests.

population were not found to be associated with asthma in the non-Hispanic white population and vice-versa. This pattern was consistent across all chromosomes.

Finally, I used the dynamic query system in GenAMap to compare results from different association tests. The capabilities of this system are highlighted in Figure 4.10. Figure 4.10 shows the results from a query that compares the four different tests in population 2. In the query dialog, I selected the four tests that I wanted to compare, set the significance level of the tests (1e-4), and chose what populations I wanted to consider. By performing this analysis, I found 13 SNPs that were associated with asthma in population 2 (non-Hispanic whites), listed in Figure 4.10.


Distribution for: ppFvcHANK by rs7661051

mu= 90.94, sigma= 17.35, se= 0.65

mu= 94.47, sigma= 14.4, se= 1.16

mu= 100.01, sigma= 16.23, se= 5.1:

t-test scores: 0v1=0.008238321564982585; 0v2=0.11272499297246288; 1v2=0.31805175242266714

Figure 4.11: Frequency distribution of asthma trait by genotype at rs7661051. In this figure, I explore the association of an asthma trait (ppFvcHank) with a SNP on chromosome 4. I notice a significant difference in the trait distribution in population 2 between individuals with no minor alleles and individuals with at least one minor allele, suggesting that this allele has a protective affect against asthma.

4.2.3 Using structured association mapping to analyze the asthma data

In addition to the simple statistical tests, I also ran MPGL (Puniyani et al. 2010) on the asthma data. MPGL finds associations by population, but also assumes that diseases may have similar causal SNPs across the different populations. I used a small subset of SNPs to run MPGL on the data; I took all SNPs with a *p*-value of less than 1e-4 in either population, giving a total of 131

SNPs (60 from population 1 and 71 from population 2 with no overlaps). Using these SNPs, MPGL found 12 SNPs associated with asthma related traits in at least 1 population.

MPGL itself incorporates information between the populations when finding signals, however, it does not correct for the different frequencies of SNPs and cases/controls between the populations. For example, in this data population 1 has 56.6% cases and population 2 has 66.2% cases. I wanted to validate the 12 discovered associations; thus I performed a series of tests on each SNP-trait pair that was uncovered by MPGL:

1. The trait had to be associated with the genotype using the Wald test through PLINK at the .01 significance level.
2. The trait distribution had to differ (according to a t-test) between individuals with the major and minor alleles at the .01 significance level.
3. The linear model that includes the SNP had to perform better than the null model at the .01 significance level.

Given these stringent tests, I found that only rs7661051, which is located on chromosome 4q at base pair 78380281, was has an association with the trait ppFVCHank in the non-Hispanic white population that could pass all three tests. None of the other SNP-trait pairs could pass these tests, suggesting the association we found was due to other artifacts.

I will briefly report the statistics found for this SNP-trait association pair: rs7661051 was associated with asthma (*p*-value = 9.01e-5) in the non-Hispanic white population, and was also associated with ppFVCHank (*p*-value = .005215) in the non-Hispanic white population. ppFVCHank levels are higher in individuals with a minor allele at the rs7661051 locus (*p*-value = .0082) in the non-Hispanic white population. A linear model that includes rs7661051 predicts ppFVCHank levels better than a null model (likelihood *p*-value = .0053) in the non-Hispanic white population.

rs7661051 is located in a non-coding region on chromosome 4q. The closest genes to this region are *CXCL13* and *CCNG2*. In a separate, coordinated investigation, Dr. Sally Wenzel found that *CXCL13* was differentially expressed between cases and controls in collected expression data (q-value is about .03). The gene is also associated with increasing wheeze, chest tightness, and shortness of breath. *CCNG2* was not differentially expressed. Given these results, further steps are being done to consider *CXCL13* as a potential asthma gene.

In summary, my results found a significant SNP, rs7661051, that was associated with clinical lung function trait "Forced Vital Capacity (FVC), % predicted" in the non-Hispanic white population. Cases have a lower level of FVC (*p*-value = 4.1e-17), thus rs7661051 could potentially impact asthma in non-Hispanic whites with effects to alter FVC. The distribution of this trait is significantly different between individuals with no minor alleles and individuals with a minor allele (Figure 4.11, *p*-value = 8.24e-3). Further verification work is ongoing.

4.3 Structured association mapping analysis of NIH heterogeneous stock mice data

One resource that is available for association studies is the NIH heterogeneous stock mice dataset (Johannesson et al. 2009). The dataset consists of 460 mice that have been genotyped for 12,545 markers and phenotyped for 97 traits (Valdar et al. 2006). Additionally, expression profiling was recently added to the dataset from the liver, lung, and brain (Huang et al. 2009). The expression profiling was done in the liver and lung for 260 genotyped mice, and in the brain for 460 mice. This dataset has been studied for SNPs associated with the mice phenotypes and for eQTLs. Cutting-edge online webpages are available to explore these associations and to investigate the strength of association on a trait-by-trait or a genomic-location basis. Thus, this dataset provides an excellent test bed to demonstrate a structured association analysis.

However, to date, the effect of an eQTL on a genome-phenome association is assumed based on the location of the eQTL in the genome. While this offers some insight into the mechanisms behind the association, the discovery of three-way genome-transcriptome-phenome associations can be enhanced using structured association mapping. As I will show in this section, by using the GFlasso-gGFlasso strategy (Curtis et al. 2012) to find three-way associations, I not only uncover SNP-trait associations, but also find SNP-gene-trait associations that uncover some of the biological mechanisms behind the associations.

To prepare the data for analysis, I preprocessed the expression data from each tissue (hippocampus, liver, and lung) using lumi (Du et al. 2008). In each tissue, I retained all probes that had a significant signal (d < .05) for at least 95% of the mice. I limit this study to 218 mice that have gene expression measurements across all three tissues. I imputed missing phenotypic traits using k-nearest neighbor imputation (Troyanskaya et al. 2001), and I excluded all phenotypes missing values for more than 30% of the mice. In summary, my dataset included 218 mice. Each mouse is genotyped for 12545 SNPs, has measurements for 173 phenotypic traits, and has gene expression level measurements from the liver (7102 probes), lung (9698 probes) and hippocampus (9733 probes).

4.3.1 Genetic network analysis

I imported SNP data, gene expression data from three tissues, and phenotypic trait data into GenAMap. I analyzed the data using GenAMap in three steps: genetic network analysis, eQTL analysis, and three-way genome-transcriptome-phenome analysis. In this subsection, I will discuss the results from the network analysis of the three tissues in GenAMap.

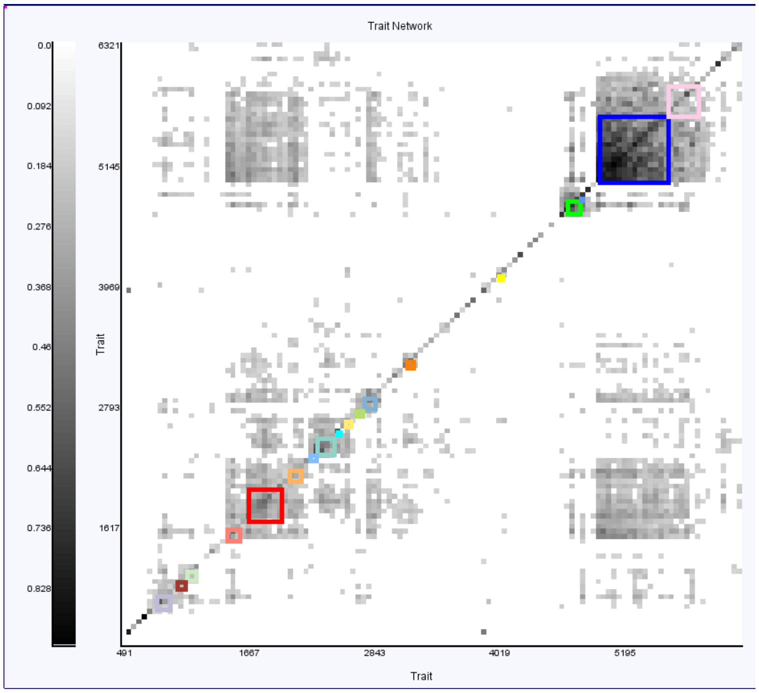


Figure 4.12: Mouse gene network analysis. I used GenAMap to create gene-gene networks from the expression data for each tissue. GenAMap finds the top 20 connected modules and GO and eQTL enrichment for each module. Here, I show the gene-gene network generated using the hippocampus gene expression data.

After importing the data into GenAMap, I used GenAMap to construct genetic networks for each tissue using the soft-thresholding method described by Zhang and Horvath (2005). I also used GenAMap to run PLINK (Purcell et al. 2007) to find SNP-gene associations. GenAMap automatically considers all *p*-values less than 1e-3 to be significant, which, although naïve in its approach, is a sufficient cutoff that allows an overview of the associations in the dataset. I used GenAMap to cluster each of the three gene networks by hierarchical clustering and to run a dynamic programming algorithm (Zhu et al. 2008) to find the top 20 connected gene modules in each network. For each of the 20 connected gene modules, GenAMap also performs enrichment analyses for the modules in terms of eQTL enrichment and GO category. I used the GO slim annotation and the associations found by PLINK for this analysis. In Figure 4.12, I show the annotated network generated from the gene expression data from the brain. The top connected modules identified by GenAMap are outlined in color.

I found that the gene networks were quite dissimilar across the three tissues. For each tissue, I found the number of unique genes (some genes are represented by multiple probes) and the number of unique edges between genes. I compare the three tissues in Table 4.5. While many genes (78% of the genes in the liver dataset) are shared between the three networks, few edges are common across all three networks (only 14% of the edges in the liver gene network are common across all three tissues). Because the set of genes included in each network are similar, I suggest that the differences between the networks are due to a difference in regulatory patterns of expression across the three tissue types.

Similarly, I found the gene modules found in each network to be distinct. Specifically, I found little overlap between the genes in each of the top 20 modules identified across tissues. Also, I noticed a difference in the GO and eQTL enrichments for the modules across tissues. In the liver, I found nine gene modules that were enriched for a GO category; these are listed in Table 4.6 and include enrichments for *mitochondrion*, *catalytic activity*, and *generation of metabolites and energy*. While the hippocampus network had eight modules enriched for a GO category, only two matched the liver enrichments and different GO categories were represented including *ribosome*, *calcium ion binding*, and *transport*.

Table 4.5 – Comparison of gene networks across mouse tissues

Tissue	**# genes**	**% genes shared w/ brain**	**% genes shared w/ liver**	**%genes shared w/ lung**	**% genes shared across tissues**	**# network edges**	**% edges shared w/ brain**	**% edges shared w/ liver**	**% edges shared w/ lung**	**% edges shared by all tissues**
Brain	7960	100	59.6	78.1	57.6	170982	100	8.9	16.6	4.2
Liver	5879	80.8	100	86.8	78.0	48768	31.1	100	22.9	14.8
Lung	7968	78.1	64.0	100	57.6	105933	26.8	10.5	100	6.8

The eQTL enrichments for the modules were also different across the three tissues. Of note, I found five modules in the lung gene expression network that were significantly associated with enrichment for association to the SNP rs3023797 on chromosome 2. No modules in the liver or brain were significantly enriched for association with this SNP. Interestingly, rs3023797 is located in the exon region of the gene *Ttf1*, transcription termination factor, RNA polymerase I (Sherry et al. 2001). *Ttf1* has previously been shown to have important regulatory roles in lung function and development in mice (Reynolds et al. 2010). These results, therefore, suggest that mutations in *Ttf1* affect the expression patterns in the lung, but not in the other tissue types. Similarly, six gene modules in the lung network also had an enrichment for association to chromosome 12 (26000000), which was not seen in the other two tissues. This suggests that there is a second mutation that is affecting lung expression patterns, but not hippocampus or liver expression.

Table 4.6 – Gene modules with GO enrichment in the liver network

Module number	**# genes in module**	**eQTL location**	**eQTL *p*-value**	**GO Category**	**GO *p*-value**
1	446	11 (4877160)	1.47E-57	mitochondrion	3.80E-04
2	104	17 (61151939)	6.10E-07	catalytic activity	1.96E-04
4	201	14 (9353843)	7.42E-114	ion channel activity	2.02E-04
5	97	19 (20354841)	3.38E-31	mitochondrion	1.11E-13
8	89	17 (61151939)	1.81E-07	cytoplasm	3.73E-04
12	45	13 (56818025)	2.56E-10	regulation of gene expression epigenetic	5.59E-05
14	22	1 (76152963)	8.61E-07	generation of metabolites and energy	6.28E-04
15	34	19 (21138174)	4.18E-10	ER	7.08E-04
20	20	6 (42868138)	1.31E-11	nucleic acid binding	2.34E-04

4.3.2 eQTL analysis using GFlasso

Given the modularity of the gene expression networks, I used GenAMap to run GFlasso (Kim and Xing 2009) to identify eQTLs for each tissue type. GenAMap uses cross-ten-validation to find optimal values for λ and γ using a linear search strategy (documented online: http://sailing.cs.cmu.edu/genamap). I downloaded all results from GenAMap and found all SNP-gene associations. SNPs within 2MB of each other and associated with the same gene are counted as the same association. I used the genomic locations of all genes (Blake et al. 2011) to classify associations as *cis*- or *trans*-associations. I define an association as a *cis* association when the SNP and gene are located on the same chromosome and located within 10MB of each other.

The GFlasso results were quite different in the different tissues. In the liver, GFlasso identified six SNP-gene associations; all six associations were cis associations. The results from the lung were similarly sparse, with 25 SNP-gene associations discovered. GFlasso found one *trans*-association and 24 *cis*-associations. Overall, GFlasso found two *cis* SNP-gene associations that were common across all three tissues (*Gps2* and *Psmb6*), one *cis*-association common between liver and lung (*C4b*), and four *cis*-associations common between lung and hippocampus (*Mrpl15*, *Hsd17b11*, *Rpl21*, *Hbb-b1*).

In contrast to the sparsity of the GFlasso results in liver and lung, GFlasso found many eQTLs using the hippocampus data. Specifically, GFlasso identified 467 SNP-gene associations for 103 SNPs and 268 genes. GFlasso identified 138 *cis*-associations and 329 *trans*-associations in the dataset. 79 genes were associated with more than one SNP, and 6 SNPs were associated with more than 20 genes. Although the sparsity of the results for liver and lung is surprising, our results are consistent with previous reports (Huang et al. 2009) that found that "*trans*-eQTLs are twice as common as *cis*-" in the brain, and that *trans*-eQTLs are more common in the brain than in the other two tissues. Because the strength of GFlasso is to identify SNPs that affect multiple-correlated genes, it is no surprise that it was able to identify many trans-eQTLs in the hippocampus gene expression data as compared to the other two tissue samples. Because the results from the hippocampus are the most interesting, I focus on these signals in the remainder of this subsection.

I present an overview of the SNP-gene association results for the brain gene measurements in Figure 4.13. From the overview, I note in particular one long horizontal line, suggesting the discovery of an eQTL hotspot that regulates many genes in trans. I also note the presence of other, shorter horizontal lines, including some short lines that overlap with some of the genes of the largest eQTL hotspot. I used GenAMap to discover the location of the SNP that was associated with these genes, rs8244120 on chromosome 14. I used GenAMap to find rs8244120 in dbSNP and found that it is located in the exon coding region of two genes: *Tmem55b* and *Apex1*. *Apex1* has been annotated for GO categories such as *DNA binding* and *DNA demethylation*, suggesting that *Apex1* potentially regulates the expression of other genes through its interaction with the genome.

To better understand the genes associated with this genomic region, I used GenAMap to create a subset of all genes associated with rs8244120. GenAMap identified 140 genes associated with rs8244120 in the GFlasso results. I performed a GO enrichment analysis using GenAMap on these genes to see if they shared common annotations. In fact, GenAMap found that the associated genes were enriched for the GO category *cell projection* (*p*-value = 2.65e-5). Cell projection is defined as "A prolongation or process extending from the cell, e.g. a flagellum or axon" (Binns et al. 2009). Indeed, many of the genes in this subset are annotated to GO categories indicating involvement in brain function (e.g. *Gas7*, *Nrp1*, *Stx1a* are annotated to the GO category *neuron projection development*). I selected the 22 genes annotated with the cell projection annotation and saved them as a subset for further analysis. These 22 genes were enriched for many GO annotations including *cell projection* (*p*-value = 1.4911e-27), *neuron projection* (*p*-value = 5.3239e-17), *axon* (*p*-value = 3.3027e-9) and *dendrite* (*p*-value = 2.6457e-7).

I was interested to consider the associations of the identified cell projection genes to the SNPs (Figure 4.14). I plotted the Manhattan plot of the associations for the genes across GenAMap's genome browser. I noticed that all of the genes were associated with rs8244120, as expected, but that many genes had other associations as well. Two of the genes were also associated to rs13482353 (also on chromosome 14, 56MB away), and three of the genes were

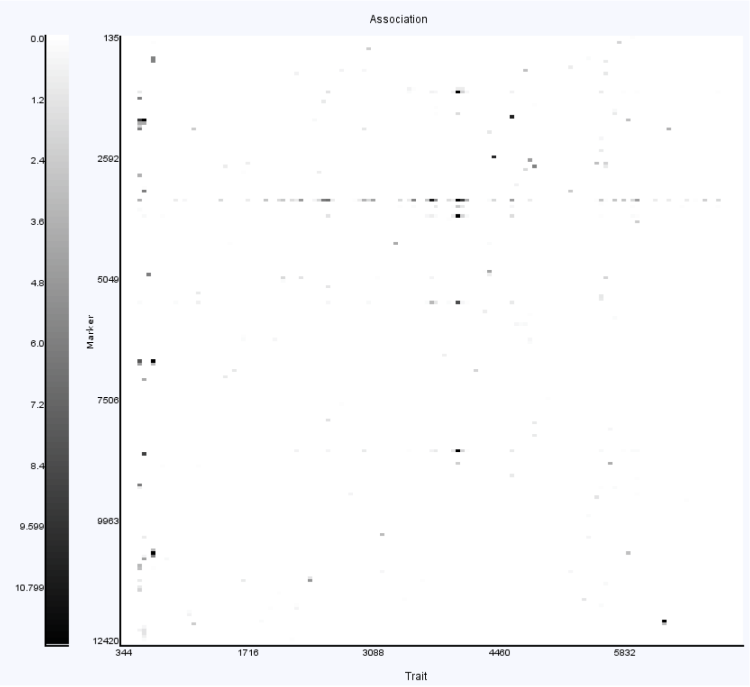
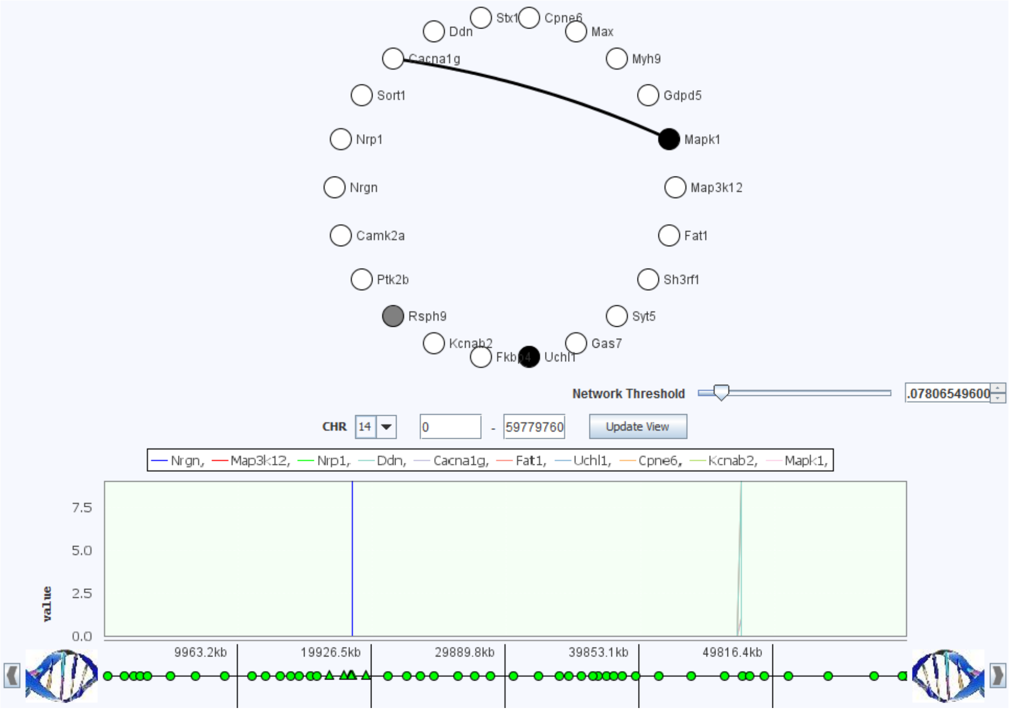


Figure 4.13: eQTLs found in hippocampus tissue. I used GenAMap to find SNP-gene associations in the hippocampus gene expression data using GFlasso. In this figure, I show the overview of the results in GenAMap. This is a heat chart representation of the associations, where SNPs are represented along the *y* axis and the clustered genes are represented along the *x* axis. I have zoomed into the section of the gene graph where there are the most associations. I note an eQTL hotspot (represented by a horizontal line of associations).

associated with rs3722205 on chromosome 18. I looked into these two SNPs in more detail and found that there are 27 genes associated with rs13482353, 25 of which are also associated with rs8244120. I also found that 25 of the 27 genes associated with rs3722205 are also associated with rs8244120. These results suggest that these SNPs may interact in some way to regulate gene expression in the mouse hippocampus.

I also investigated an unrelated set of 41 genes that are associated with rs1348069 on chromosome 10 in the GFlasso results. I found that these 41 genes are enriched for several GO categories including *extracellular ligandgated ion channel activity* (*p*-value = 2.2018e-4), *membrane depolarization* (*p*-value = 2.5393e-4), and *synaptic transmission* (*p*-value = 5.1675e-4). rs1348069 is in the intron region of *Slc5a8*, a gene that has been annotated to the GO



Network Threshold .07806549600

CHR 14 0 - 59779760

Figure 4.14: Association of axon genes to chromosome 14. I found that rs8244120 on chromosome 14 was associated with 140 genes enriched for cell projection, implying function in neuronal axons. Here, I show 22 of these genes in GenAMap's node-link view, colored by the strength of association to rs8244120. White genes are strongly associated and black genes are weakly associated (gray is intermediate). I found that some of the genes were also associated with another SNP on chromosome 14 (shown) and some of the genes were associated with a SNP on chromosome 18 (not shown).

category *ion transport*, suggesting that this SNP may play a role in cell ion signaling by affecting these genes through altering the function or expression of *Slc5a8*.

4.1.3 Joint three-way genome-transcriptome-phenome analysis using GFlasso-gGFlasso

Table 4 – GFlasso-gGFlasso associations matching previous results (Valdar et al. 2006)

SNP	Chr	Gene	Trait
rs13459079	4	*C1qb*	Alkaline phosphatase
rs4226889	7	*Nsmce1*	Weight at 6 weeks
rs3718803	11	*Pcdh20*	Aspartate Transaminase
rs3023277	11	*Psmb6*	Mean corpuscular haemglobin
rs6326787	11	*Gabrd*	Startle response
rs6380524	11	*Ube2g1*	Startle response
rs4229111	11	*Mpp3*	Startle response
rs1348295	17	*H2-T22*	CD4+/CD8+
rs1348295	17	*H2-T22*	%CD4+/CD3+
rs1348295	17	*H2-T22*	%C8+ cells

Given the results of the eQTL analysis, I determined to run gGFlasso to find associations from the brain tissue genes to the clinical trait phenotypes. I ignored all traits that were marked as "Covariates," since these were largely dates, experimenter ids, and other variables such as gender and litter. Overall, GFlasso-gGFlasso found 759 SNP-gene-trait associations. These associations included 138 associations to the X chromosome, which were ignored due to possible gender effects. The results of GFlasso-gGFlasso thus consist of 621 associations between 98 SNPs on 18 chromosomes to 156 genes that are associated with 94 phenotypic traits.

I compared the GFlasso-gGFlasso results with the top 29 results reported using a SNP-trait association method (Valdar et al. 2006). I found nine matches where GFlasso-gGFlasso found a SNP-gene-trait association that matched the previously reported SNP-trait associations. I list these matches in Table 4.7. The GFlasso-gGFlasso results suggest associated genes that help to explain the SNP-trait associations that were previously discovered.

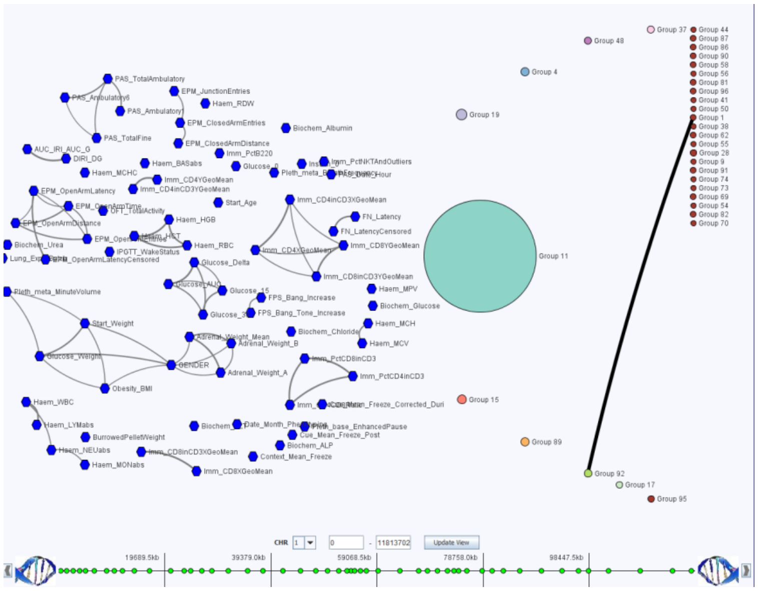


Figure 4.15: Overview of three way GFlasso-gGFlasso association analysis. I show the overview of the trait-network and gene-network from GenAMap for the GFlasso-gGFlasso analysis; associations are not shown. In this visualization, circles represent groups of genes, associated to the same regions in the genome. Hexagons represent traits. The edges between genes or between traits represent the connections in the gene or trait network. In this data, I note that there are very few edges between gene groups. The largest gene group is the teal group, representing genes associated with the eQTL hotspot on chromosome 14. The trait network consists of small sub-groups of related traits.

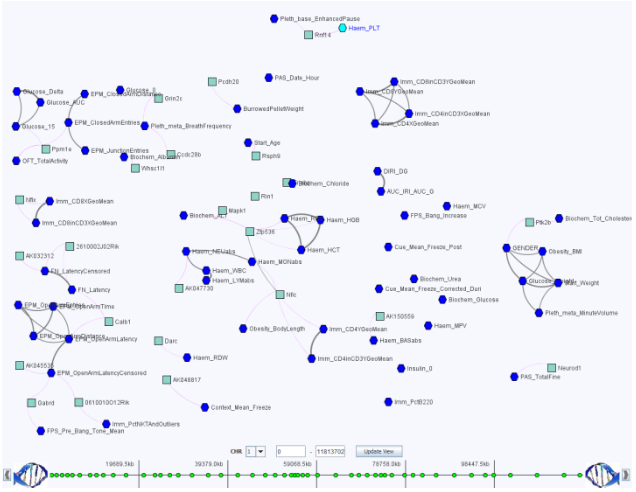
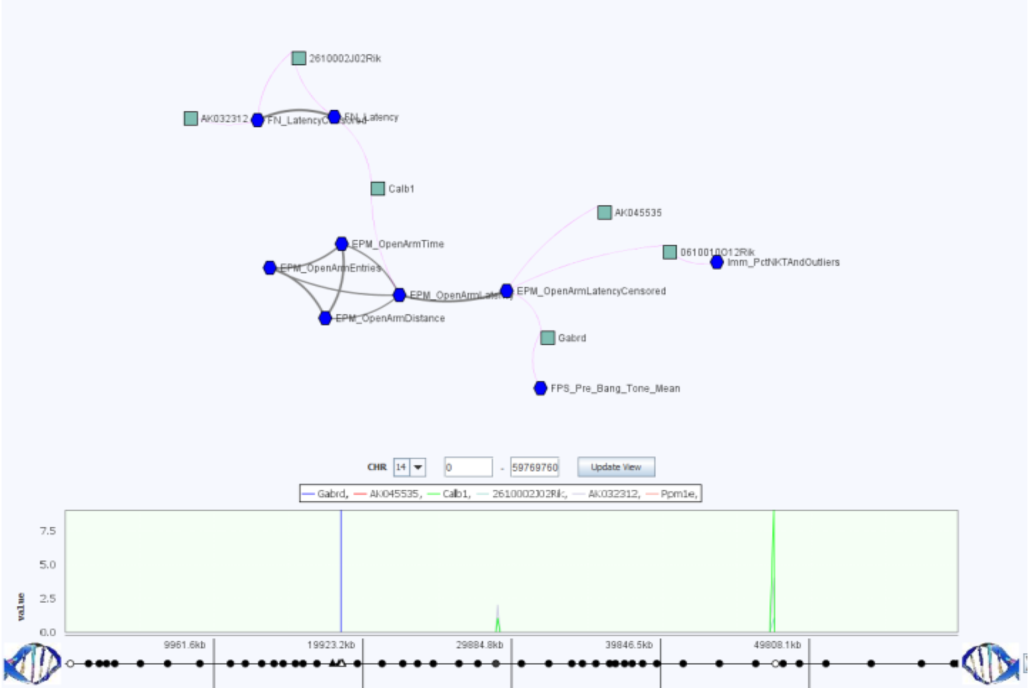


Figure 4.16: Gene-trait associations for genes associated with chromosome 14. I used GenAMap to explore the joint three-way associations corresponding to the eQTL hotspot on chromosome 14. I removed all other gene groups and then expanded the group to see the individual genes (squares). I filtered out all traits without associations to these genes.

I used GenAMap to drill down to explore the specific associations in the results. First, I considered the overall structure of the gene and trait data (Figure 4.15), noting that the largest gene group was associated with the eQTL hotspot on chromosome 14 as discovered in the previous subsection. To better understand the associations of these genes to the phenotypic traits,



CHR 14 0 - 59769760

Figure 4.17: Joint SNP-gene-trait associations from chromosome 14. I found a subnetwork of traits and associated genes involved in brain function. These genes were also associated with the overlapping eQTL hotspots on chromosome 14.

I used GenAMap to zoom into these genes and the associated traits, filtering out all other genes, traits, and associations (Figure 4.16). After exploring the results, I was especially interested in six genes that were found to be associated with sub-networks of anxiety traits (*Elevated plus maze open arm time*, *distance*, *latency*, etc.) due to the probable link between the brain and the traits themselves. In Figure 4.17, I show these traits, the correlations between traits (represented as gray lines between traits), and the gene-trait associations (pink lines between genes and traits). I also considered the associations of these genes to the genome and found that the genes were associated with two regions on chromosome 14. These results are consistent with previous findings that found two peaks on chromosome 14 associated with these traits (Valdar et al. 2006). Furthermore, the results also suggest potential mechanisms for these associations. For example, consider *Calb1*, a gene associated with the two eQTL hotspots and the anxiety traits. *Calb1* has been annotated to the axon, and knockout mice are known to show severe impairment in motor coordination (Blake et al. 2011). Similarly, *Gabrd* is also associated with one eQTL hotspot and these traits. *Gabrd* knockouts have increased postpartum depression and anxiety, along with other disorders, and *Gabrd* is annotated to be involved in ion transport (Blake et al. 2011). Thus, current knowledge supports the model that the GFlasso-gGFlasso results uncover: mutations on chromosome 14 affect the expression levels of *Calb1* and *Gabrd* in the hippocampus to affect anxiety traits such as *Elevated plus maze open arm time*.

I also considered other associations that GFlasso-gGFlasso uncovered. For example, the GFlasso-gGFlasso results found an association between chromosome 17 and immunology traits (CD4+/CD8+, %CD4+/CD3+, and %CD8+), which was also reported as a strong signal using the simple SNP-trait association method. I was interested to see if GFlasso-gGFlasso provided further mechanistic insight into this association. I used GenAMap to drill down to this association (Figure 4.18). I found a gene group consisting of four genes that were associated with these three immunology traits. One gene, *H2-T22*, was associated with all three correlated immunology traits. *H2-T22* was found to be associated with rs13482952 on chromosome 17, which is 3.2 mega-bases away from the *H2-T22* coding region. Given that the resolution for this cross is about 2MB, this SNP likely affects expression of *H2-T22* in *cis*. In fact, this region on chromosome 17 is part of the mouse H2 region, the major histocompatibility complex (MHC). The H2 region is the mouse ortholog to the human HLA region and encodes genes involved in the mouse immune response (Stuart 2010). To summarize the H2 genes in mice, there are two classes of H2 genes. Some H2 genes (class I) are expressed in virtually all cells and display “self” antigens, while others (class II) are expressed only in antigen-presenting cells (Kumanovics et al. 2002). The immunology traits associated with *H2-T22*: CD8+, CD4+, and CD3+, refer to proteins on the surface of immune response cells that bind to the antigens on the surface of other cells in the organism. *H2-T22* has been annotated as a membrane protein (Blake et al. 2011), and it likely participates in this immune response pathway. As the immune response is common across all cell types, one would expect to find this association in all cells, including the brain tissue.

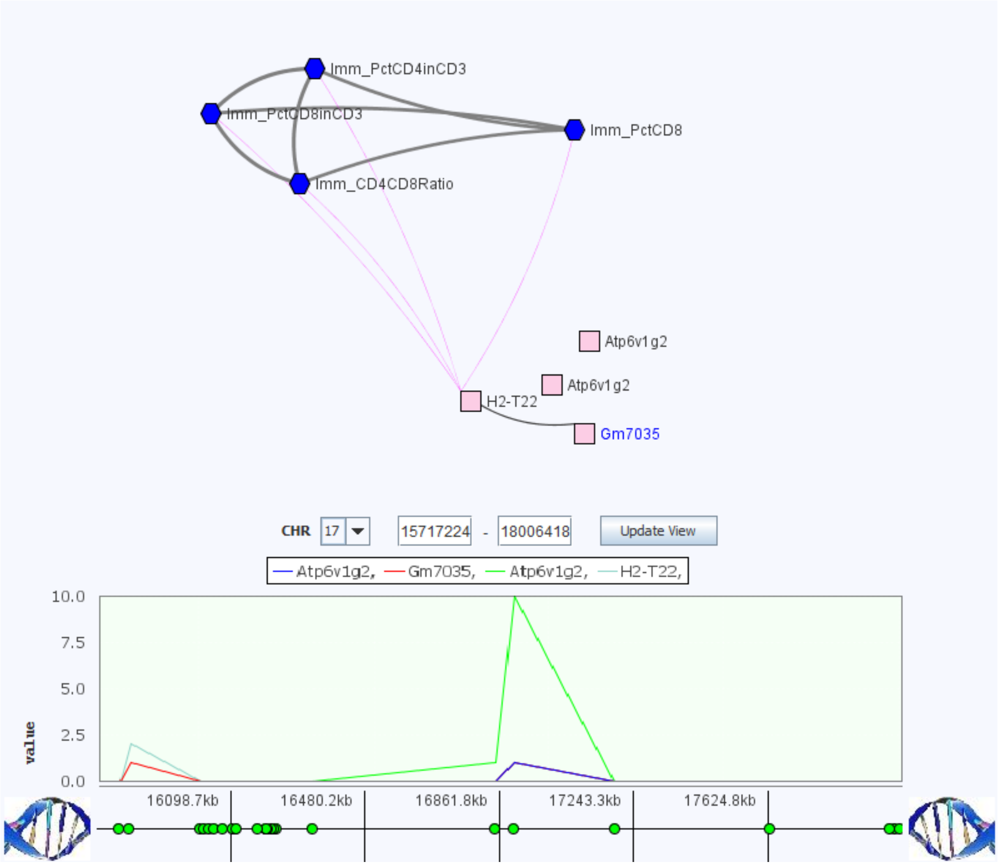


Figure 4.18: Immunity associations from chromosome 17. I found a small group of genes associated with the H2 region on chromosome 17. I also found that these genes were associated with a subset of immunology traits.

5. GenAMap Evaluation

If we consider academic research in two camps, invention and discovery, then this dissertation is clearly a hybrid of the two. On the one hand, the biological analyses that I have led clearly fall under the premise of discovery: I have uncovered new biology that was not known before. These discoveries have been supported by experiment and additional literature data. However, in the development of inventions, such as gGFlasso and the visualizations in GenAMap, the validation of the method becomes more difficult. In the case of gGFlasso, I compared results to different research using simulation. However, this is not possible in the case of the visualization tools, as GenAMap is a pioneer software system in structured association mapping visualization.

The question then remains, how to validate the new visualizations and software that were designed and built in this dissertation? In part, the results presented in the biological analyses conducted using GenAMap help to validate the visualizations and software, as the software was used to conduct the analysis. In addition to the biological application of the software, however, I have conducted additional studies and collected other data to track the impact of the software. In this chapter, I present the results from a qualitative user study that was used to evaluate GenAMap and its ability to facilitate structured association mapping analysis. I also report a few tracking measures that suggest the impact that GenAMap has made in the broader community. Although the impact of inventions may be difficult to track, the results presented in this chapter suggest that GenAMap has been a significant contribution in the association mapping field.

5.1 User study results

I performed a preliminary qualitative user study to assess the utility of my visualization techniques and to get feedback on steps we could take to improve the visualizations. I recruited PhD students and post-docs with specific research interests in genetics from two universities. I had eight volunteers participate in the study: seven PhD students and one post-doc. There were four male participants, and four females. All of the participants are involved in genetics research with an emphasis on machine learning development. I assessed the level of expertise in association mapping of the candidates based on three criteria: 1) self-rated expertise in association mapping, 2) self-reported participation in an association mapping project, and 3) the ability to explain what an eQTL study entails. Using these criteria, the participants consisted of four experts in association mapping (met two or more criteria) and four non-experts.

Each participant met with me privately in a standard office space. I guided them through five different tutorials. GenAMap was run on a standard desktop-computer with a 22-inch screen. Participants were encouraged to think-aloud as they used GenAMap; they were given semi-structured tasks to explore the tools with guidance and on their own. I asked for verbal feedback at each stage of the study. Sessions lasted about an hour; I took notes of all comments throughout the session, and users filled in a survey upon completing the evaluation. The post-use survey was presented to the user as a written survey after completion of the semi-structured tasks. Users had the option to report their name with their responses, or to remain anonymous. I was available for clarification questions during the survey; otherwise the survey was completed independently. Upon completion of the survey, users returned the written responses to me to be filed with the notes from the think-aloud session.

5.1.1 Survey results

The post-use-survey had twelve questions where the user had to rate the software on a scale from 1-to-5, with 5 representing a high score. Overall, the users reported that GenAMap allowed them to explore association results better than other tools (average score 5.0) and that GenAMap allowed them to get an overall feel for the structure in the data (4.75). They all agreed that GenAMap lead to insight that was not available using other tools (4.71), and that they would recommend GenAMap to other researchers (4.75). The lowest scores from the survey were in regards to the usability of the system. While the utility scores just mentioned were high, the participants did not agree as strongly that GenAMap was easy to learn (average score 3.75), or that the visualization strategies were always easy to understand (4.0). However, the lowest rating that we had from any user was a score of 3 for any of these questions.

In the free response part of the survey, the users were prompted for the most useful part of GenAMap that they explored. Four of the users specifically mentioned the incorporation of outside data, including GO category analysis and external databases. In fact, when asked specifically about external links, seven of the participants responded very positively. When asked what views led to the greatest insight, seven participants specifically mentioned a visualization strategy that incorporated multiple views that allowed them to explore the association results between the genome and the traits represented in some structure.

5.1.2 Think-aloud comments and results

I presented three integrated views to explore the results of association analysis to the users: the association tree view (Section 3.3.4), the population association view (Section 3.3.7), and the network association view (Section 3.3.2). None of the users had seen association results presented in such a way previously.

All the users reported that they liked each of the visualizations. The tree view was met with some reservation, but in the end, users found that they could think of different uses for it. The users gave me many ideas of different queries they wanted to be added to explore the tree, and one user specified that the tree view was his favorite visualization technique. All the users mentioned that GenAMap was an improvement over how they would normally do these types of studies. For example, six of the users mentioned that the tree view was a more convenient way to explore the results than MATLAB or another command line interface.

Users also liked the integration of the different views of the structure of the data. Five users specifically mentioned that they would have had to use a combination of tools or done the work by hand in order to complete a similar analysis. One user remarked, "By myself I would have to go back and forth between the human genome browser and a network viewer. It is really nice that it is integrated into the software." Five of the users specifically mentioned that they liked being able to explore the genome and the gene-gene network in the same integrated tool. One user said, "The ability to interact with the network and the genome is excellent," and another commented that "it really puts it into perspective." Five users specifically mentioned that using GenAMap was easier, more convenient, and saved time by allowing them to more systematically explore the data.

Many of the feature requests the users offered were related to documentation and exporting data. Five users wanted better documentation and links incorporated into the software so they could more easily identify what different plots and charts represented. Additionally, five users mentioned that they would have liked the ability to export data from GenAMap for further analysis using more specialized tools.

I include further discussion of these results in Section 7, the conclusions of the dissertation.

5.2 The popularity of GenAMap

As an informal means to track the impact of GenAMap, I have kept statistics on the visitors that have visited GenAMap's website on the SAILING Lab webpage. While many people may visit the webpage and not download or request a log in to GenAMap, the software can contribute to the field by supporting association mapping and advancing the importance of visualization in biology. If the long-range vision is to create a culture where cutting-edge machine learning research and biology truly come together via software visualization, then the popularity of the software is one plausible metric that can help explain the impact of GenAMap.

I present Figure 5.1, which describes the activity on the GenAMap website since GenAMap was released in February 2011. I report on the number of unique visitors to the website each month, the number of page views, and the average time on the website. I also note the number of users who have requested an account to GenAMap.

GenAMap was highlighted in a tutorial in ISMB 2011 (Xing and Kim 2011). At the same time, GenAMap 3.0 was released, which was a near-final version of the software. GenAMap was getting around 20 visitors a month before that time, but the number of visitors has increased dramatically since then to about 80 visits per month. October 2011 was also an eventful month as five users requested login accounts to use GenAMap. Another interesting trend is the average number of minutes on the website. About the time that I posted the video tutorials on the GenAMap website, the number of minutes on the website increased significantly. About this same time, the paper describing the three-way association visualizations was reviewed for inclusion in the Pacific Symposium on Biocomputing (PSB). Either of these events could account for the increase in time spent on the website.

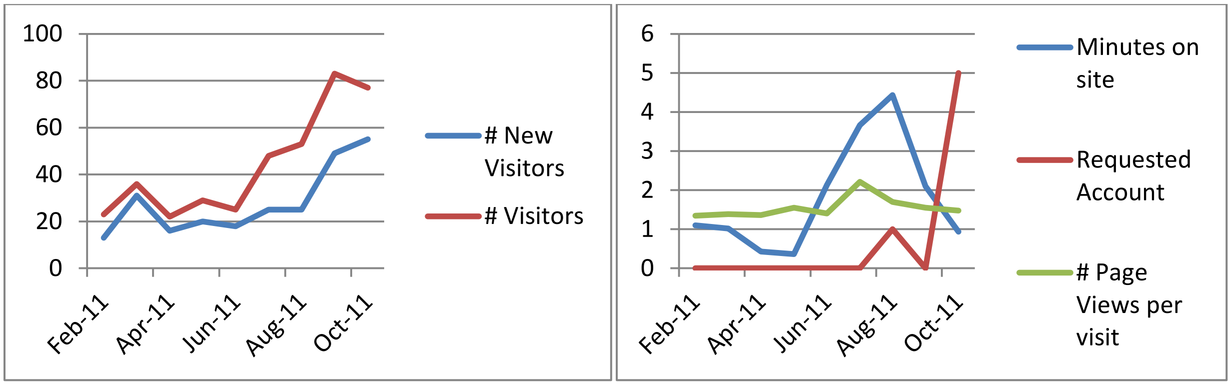


Figure 5.1: Measures of the popularity of GenAMap on the world wide web. I track the number of visitors that visit the website each month, as well as how long the visitors spend on the site and how many pages they view.

6. TVNViewer

As discussed in Chapter 1, biological relationships, such as those between proteins, between transcription factors and DNA binding sites, and between cells, differ across different tissues and change over time. Recent studies have shown that changes in network architecture from the cell cycle and in response to diverse stimuli are quite significant (Luscombe et al. 2004). Despite the dynamic nature of these interactions, the general scientific paradigm has often been to consider these relationships as static entities. However, many recent studies have advanced the field through insight from network dynamics across time or tissue in a variety of species including: human blood leukocyte response (Calvano et al. 2005), rice regulatory hierarchies of gene expression (Jiao et al. 2009), temporal interaction in *C. elegans* (Dupuy et al. 2007), and correlated changes in gene expression between mouse tissues (Keller et al., 2008). The study of network dynamics has the potential to produce crucial discoveries in gene regulation, the cell cycle, and cancer progression.

Representing biological relationships visually through networks is considered a good way to demonstrate the interplay between different genes or proteins. Network representations of such data range from the small and simple networks to large, complex networks that represent thousands of genes. The potential for network visualization to aid in our fundamental understanding of biological relationships has resulted in an explosion of software platforms and visualization toolkits (Section 1.5). However, despite the complex array of software available, there remains the unmet need for software to explore dynamic networks (Pavlopoulos et al. 2008; Suderman and Hallett 2007).

To fill this gap, I led the development of TVNViewer, a free, open-source website built specifically for the visualization and analysis of biological networks that change over time and space. Both circle and force-directed layouts are available for users to scroll through a series of networks, providing an immediate and natural way to identify changes in the network. Additionally, TVNViewer gives the user access to analysis tools to gain insight into how the degree of the nodes in the network changes over time. Although TVNViewer was developed primarily with biological interactions in mind, it can be used to visualize any type of dynamic network.

6.1 TVNViewer Overview

TVNViewer is built for the visualization of small to moderate datasets of up to 500 nodes for gene-gene interactions, or if genes are grouped into a descriptor category (such as a GO category annotation), TVNViewer can handle up to 5000 nodes classified by up to 100 descriptors. TVNViewer accepts a series of up to 50 networks in text or xml format; detailed instructions on how to prepare files for upload are available through TVNViewer's import wizard. TVNViewer supports three visualization paradigms: a gene-gene interaction paradigm, a two-tiered gene-gene interaction paradigm, and a descriptor paradigm where nodes are classified according to some descriptor category. Each different visualization paradigm provides a slightly different way to explore the network; examples of each paradigm are available by running the examples available on the TVNViewer website. TVNViewer can be used through a temporary session to upload data, or users also have the option to create a user account on the server (free of charge) in order to save data for future access.

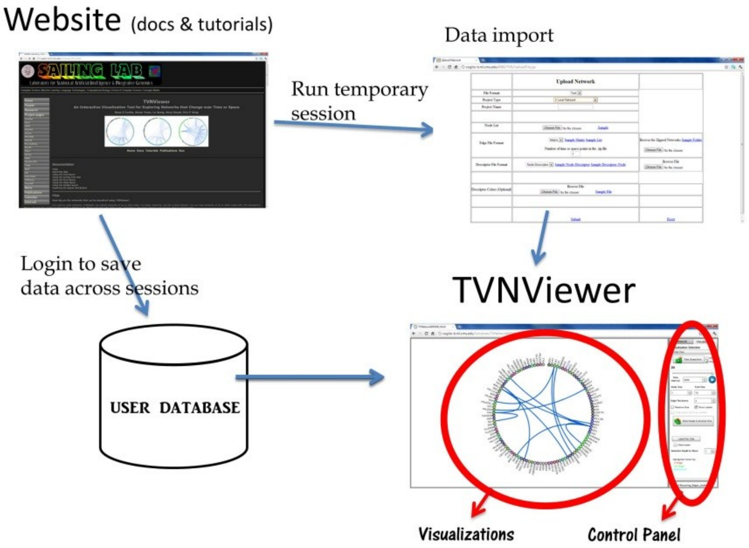


Figure 6.1: An overview of the TVNViewer system. TVNViewer is designed as an online tool. Users can access TVNViewer directly from http://sailing.cs.cmu.edu/tvnviewer. The website has a series of documents, tutorials, and examples that teach how to use the tool. TVNViewer can be run using a temporary session or a login. If a user login is created, analysts can upload datasets to persist across sessions. The TVNViewer display is a dual display with the visualizations on the left, and the control panel on the right. The analyst interacts with both the visualization and the control panel to explore the data and customize the display.

6.2 Implementation and Design

TVNViewer runs as a online visualization tool, an overview of the system is shown in Figure 6.1. TVNViewer can be accessed from http://sailing.cs.cmu.edu/tvnviewer. The website is simply designed; a series of html pages provide background information on TVNViewer and allow analysts to link directly to TVNViewer with or without a login. Additionally, the website presents several resources for analysts to learn how to use TVNViewer. First, the TVNViewer website includes online documentation that reviews each of the features available in TVNViewer; the website also provides a series of video tutorials which show how to use the main features of the software. Finally, the analyst has access to explore five example, preloaded networks in TVNViewer with data specifically designed to show off the visualizations in TVNViewer.

An analyst can choose to create a login or to upload data into a temporary session. Analysts who create a login can store up to ten datasets directly on the TVNViewer website; thus, the data will persist after the session is over. Data for TVNViewer is stored securely on the website by user information stored in a MySQL database. Analysts can upload data onto the website formatted in a tab-delimited or an xml format, described in the online documentation. TVNViewer runs a separate Apache process to convert the analyst's data into a TVNViewer-readable JSON file format and to store it on the website server.

TVNViewer itself is implemented using Adobe ActionScript, and thus runs on all major browsers with the freely-available Adobe Flash plug-in. TVNViewer is an open-source project (source code is available from the main website). To implement TVNViewer, I took advantage of Flare, a recent, powerful visualization toolkit. Flare is an easily-customized, open-source web-visualization project that provides many cutting-edge visualization strategies for the visualization of different data types (flare.prefuse.org). I significantly expanded the available templates to customize the visualizations into what is now available in TVNViewer. All images and views created by TVNViewer can be exported by the analyst to a .PNG or .PDF file. As an example of how these images can be downloaded to create a visual summary of network rewiring, I present Figure 6.2.

TVNViewer was designed using a simple Model-View-Control architecture. The nodes and edges for each gene for each time step are stored directly in the model. The analyst uses

TVNViewer’s control panels to adjust the view of the data in the model or by interacting with the visualization directly.

In addition to providing the different visual representations of the data for the analyst that I have described thus far, TVNViewer allows the analyst to customize the network views to find the preferred visual representation. Specifically, the analyst can adjust the size of the data nodes, choose to have the data nodes sized based on degree, adjust the font size of the labels, or change the visual thickness of the edges. Based on the size of the analyst’s screen, TVNViewer dynamically resizes the visualization to ensure that all labels and nodes fit within the visualization window. The thickness of each edge in TVNViewer represents its weight in the network. The analyst can adjust the thickness of the edges via the maximum threshold control. Additionally, the analyst can select what edges and node labels are visible in the visualization. For example, consider the case where a network has many edges with a low weight. In this case,

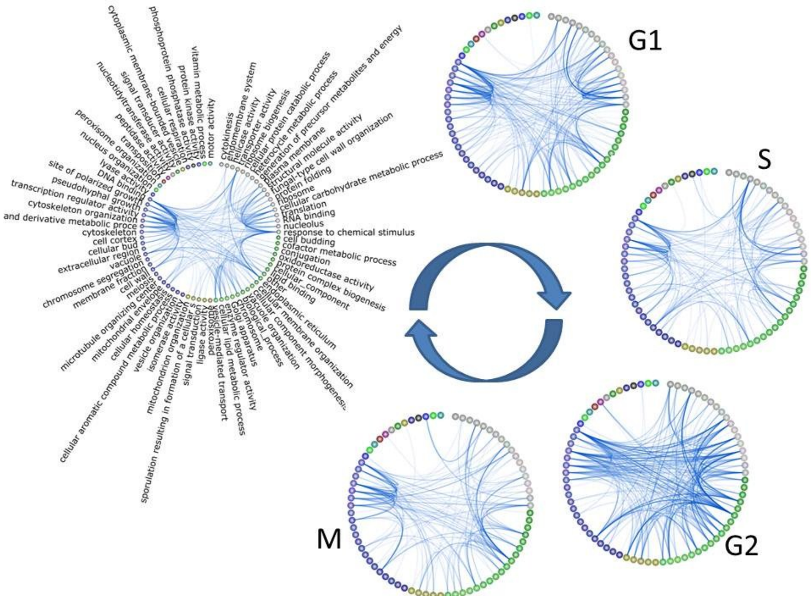

G1
S
G2
M
cytokinesis
endomembrane system
helicase activity
transporter activity
ribosome biogenesis
cellular protein catabolic process
heterocycle metabolic process
generation of precursor metabolites and energy
plasma membrane
structural molecule activity
fungal-type cell wall organization
protein folding
ribosome
cellular carbohydrate metabolic process
translation
RNA binding
nucleolus
response to chemical stimulus
cell budding
cofactor metabolic process
conjugation
oxidoreductase activity
protein complex biogenesis
cellular_component
lipid binding
other
endoplasmic reticulum
cellular membrane organization
cellular component morphogenesis
vacuole organization
biological_process
chromosome
Golgi apparatus
enzyme regulator activity
cellular lipid metabolic process
vesicle-mediated transport
peroxisome
ligase activity
signal transduction
sporulation resulting in formation of a cellular s
mitochondrion organization
isomerase activity
vesicle organization
cellular aromatic compound metabolic process
cellular homeostasis
mitochondrial envelope
microtubule organizing center
meiosis
cell wall
membrane fraction
chromosome segregation
vacuole
extracellular region
cellular bud
cell cortex
cytoskeleton
and derivative metabolic proce
cytoskeleton organization
transcription regulator activity
pseudohyphal growth
site of polarized growth
DNA binding
lyase activity
nucleus organization
peroxisome organization
transposition
peptidase activity
nucleotidyltransferase activity
signal transducer activity
cytoplasmic membrane-bounded vesicle
cellular respiration
phosphoprotein phosphatase activity
protein kinase activity
vitamin metabolic process
motor activity

Figure 6.2: TVNViewer dynamic network visualization. This figure was created from TVNViewer’s visualization tools. A simulated dataset is shown at five stages with distinct differences in the relationships between nodes. In the TVNViewer, the user can use these types of visualizations to find and highlight how the network changes.

the analyst increases the minimum edge threshold and all edges below this threshold disappear, revealing the remaining interactions. Another scenario is where the analyst is interested in only a handful of genes or gene groups. In this scenario, the analyst can remove all other labels from the visualization, highlighting the specific genes of interest. Providing customizable, interactive visualizations like these allows analysts to enhance their own cognition by putting their knowledge into the analysis. Rather than constantly having to remember numeric or ordinal values for edge weights, for example, the visualization off-loads those considerations to the visual cortex, allowing the analyst to focus on analytic activities rather than the trivia of edge weights which are only valuable for the analyst in so far as they generate insights (Liu and Stasko 2010).

6.3 Case Studies: Using TVNViewer for Dynamic Network Analysis

In this section, I present seven of TVNViewer's visualizations available for dynamic network analysis. I demonstrate how the visualizations in TVNViewer facilitate dynamic network analysis through the analysis of two real datasets. The first dataset is a yeast (*Saccharomyces cerevisiae*) microarray dataset that contains 5610 genes measured at 25 time points across two cell cycles (Pramila et al. 2006). The networks at each time point have been recovered using Time-Varying Dynamic Bayesian Networks (TV-DBN) (Song et al. 2009). The second dataset is a breast cancer progression and reversal dataset (Petersen et al. 1992); breast cells grown in a 3D culture start out as normal cells, become malignant (cancerous), and are then reverted by drugs that inhibit various signaling pathways. The networks have been recovered using Treegl (Parikh et al. 2011). I show how TVNViewer can be used to expose the similarities and differences of these cells states to reveal the effectiveness of various drugs.

6.3.1 One-level network circle view

An important challenge in dynamic network analysis is the recognition of subtle changes in the network topology over time. I designed the one-level network circle view to enable analysts to explore the rewiring of a gene network. In the one-level network circle view, the analyst sees all the genes in the dataset aligned in a circle layout. The genes are represented by circles (nodes) and the connections between genes are represented by edges (lines between nodes). The genes are clustered to minimize the number of edges going across the circle, keeping most edges local

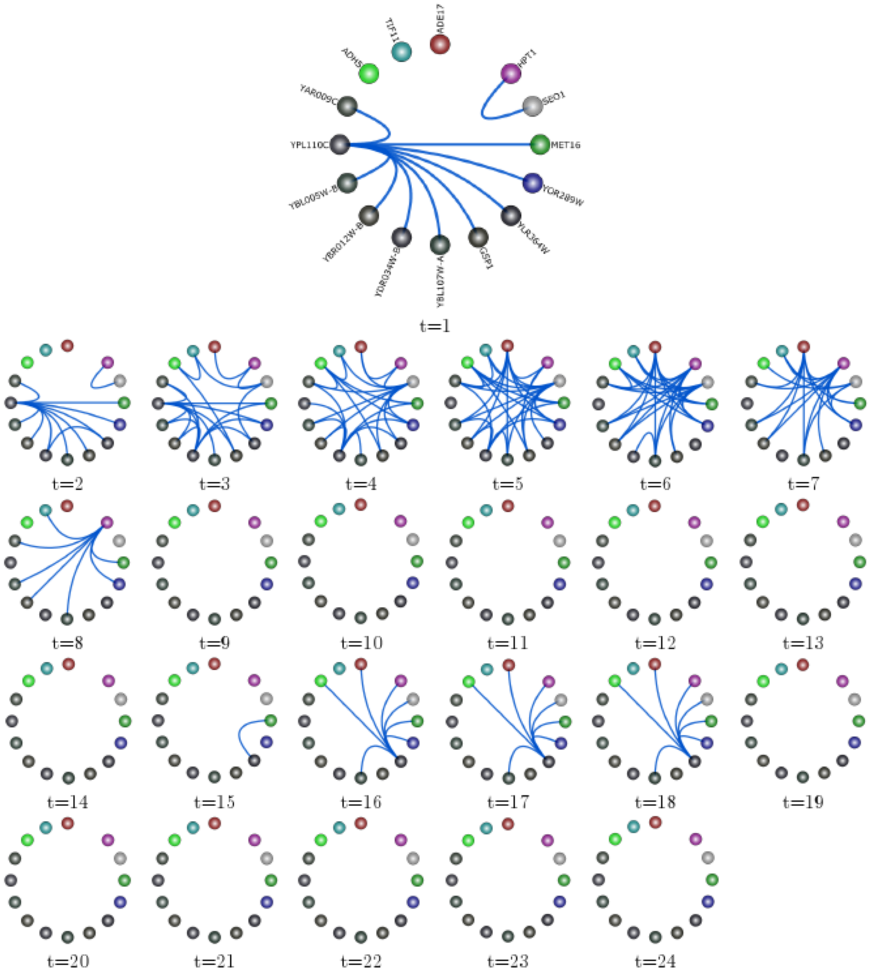


Figure 6.3: One-level gene network view. I demonstrate how the one-level gene network view in TVNViewer can be used to explore the rewiring of a subnetwork of genes generated from the yeast cell cycle data. The network rewires across two different cell cycles. The first cycle occurs during t=1-12, and the second cycle is from t=13-24. The network is most active during the initial phases of the cell cycle, which coincide with the G1 phase.

to tight clusters of genes around the edge of the circle. Genes are colored by this clustering; the analyst can see the clustering tree via the sorting tree view.

In the one-level network circle view, the analyst can step through the sequence of networks in the dataset to explore the rewiring of the gene networks. TVNViewer updates the display in real time, allowing the analyst to see the edges between genes at each time point. This feature is demonstrated in Figure 6.3, which shows a subnetwork of genes at 24 time points from a large network derived from yeast gene expression data. The top graph in Figure 6.3 represents the gene network at Time 1, and all nodes are labeled by the names of the genes they represent. To enhance the figure's readability, TVNViewer is used to remove gene name labels in the graphs representing the other time points. The 24 time points in this figure represent two cell cycles. The first cell cycle occurs between time point 1 and 12 and the second cell cycle occurs between time point 13 and 24. Figure 6.3 shows how the one-level gene network view in TVNViewer makes the appearance and disappearance of edges in the network readily accessible to the analyst, without the awkward integration or customization required by other network visualization tools. The analyst can quickly identify that this particular network is active in the beginning of the each cell cycle which corresponds to the G1 phase of the cell cycle.

6.3.2 Two-level network with GO annotations

While the one-level network is useful for analyzing small gene graphs, in many cases, however, there more genes than can be visualized using a simple circle view. In this case, it is often more helpful to group similar genes by function (*i.e.*: gene ontology (GO) category) and then to visualize the interactions amongst the groups. TVNViewer provides a two-level network view specifically designed to enable the analyst to explore a high level view of the network, via visualizing interactions at the group level, while still being able to zoom in to explore individual gene interactions.

For example, consider analyzing a T4 malignant breast cancer cell network that was reverse engineered from microarray data using Treegl (Parikh et al. 2011). Since there are 5440 genes (nodes), analyzing the actual network at a gene-level would be difficult. To create a graph for analysis, TVNViewer groups the genes according to second-level GO process groups; the interactions between these groups are visualized with TVNViewer as shown in Figure 6.4A. The thickness of an edge between two GO groups A and B corresponds to how many genes from A interact with those in B. For example, it is clear that cell death and proliferation, both indicative of cancer, are active in the network.

TVNViewer allows the analyst to filter information in the group graph to focus attention on specific groups and interactions. Consider for example, the GO group *cell killing* in Figure 6.4A. By clicking on this group the analyst removes other interactions from view and highlights only those genes involved with cell killing as shown in Figure 6.4B.

While the overall view of gene-group interactions is useful, sometimes the analyst needs to zoom in to see gene-gene interactions. TVNViewer supports this type of exploration one gene group at a time. Double clicking on the label of *cell killing* in Figure 6.4A expands the group to show all the genes that are in this group, Figure 6.4C. This view shows the gene-to-group and gene-to-gene interactions of the genes involved in *cell killing*. As before, clicking on a particular gene will filter out all interactions that do not involve that gene (for example the *TUBB* gene shown in 6.4D). In this case, from the visualization, it is clear that the *TUBB* gene (tubulin beta) interacts with genes from many groups, most notably the signaling process and biological adhesion groups. This makes sense since *TUBB* encodes proteins that are important to GTP binding and GTPase activity, in addition to its involvement in the structure of the cytoskeleton. Thus, the two level view can help the analyst get both a high level view of large, evolving regulatory networks, while simultaneously allowing him to zoom in on particular genes for a more detailed analysis. All the dynamic exploration tools that were presented for the one-level network view are also available in the two-level network view.

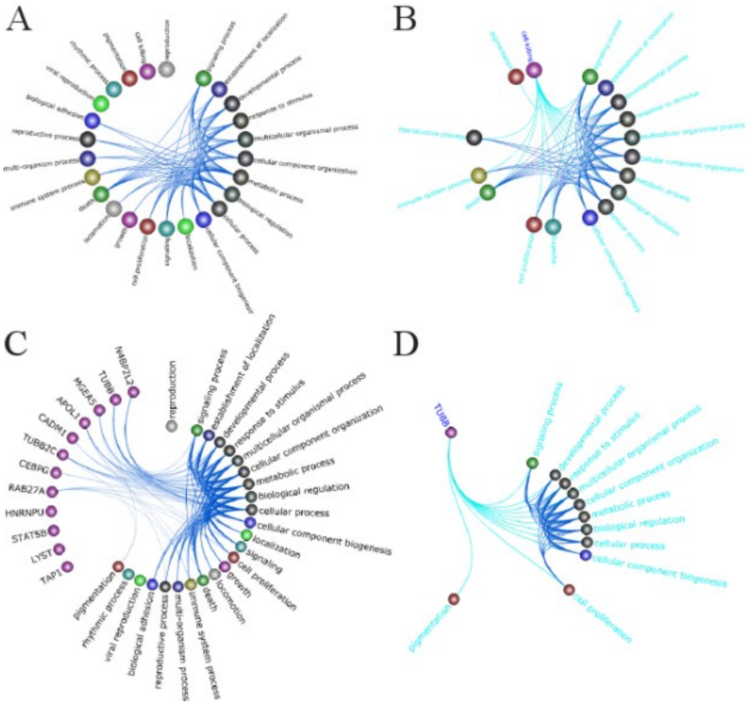


Figure 6.4: Two-level gene network view. In TVNViewer's two-level network view, the genes are grouped by GO category, and the analyst can explore the overall topology of the network or zoom into the small-scale gene-gene interactions. A) An overview of the network. Genes involved in cell death and proliferation are especially active. B) The analyst can select a GO category of interest (cell killing) to observe the specific interactions of the genes in that group. C) Groups can be expanded to reveal the genes involved in the group and their interactions with other groups. To illustrate this feature, I have expanded the *cell killing* group. D) By selecting genes, the analyst can observe the interactions of specific genes (in this case *TUBB*) with the rest of the network.

6.3.3 Force View

I present the force view (Figure 6.5) as an alternative layout to the circle view. The force-view provides the same information as the circle view in a force-directed layout. This layout can be used to study the different connected components of the network and to observe how these change over time. In Figure 6.5A I present an example subnetwork from the yeast data. In this

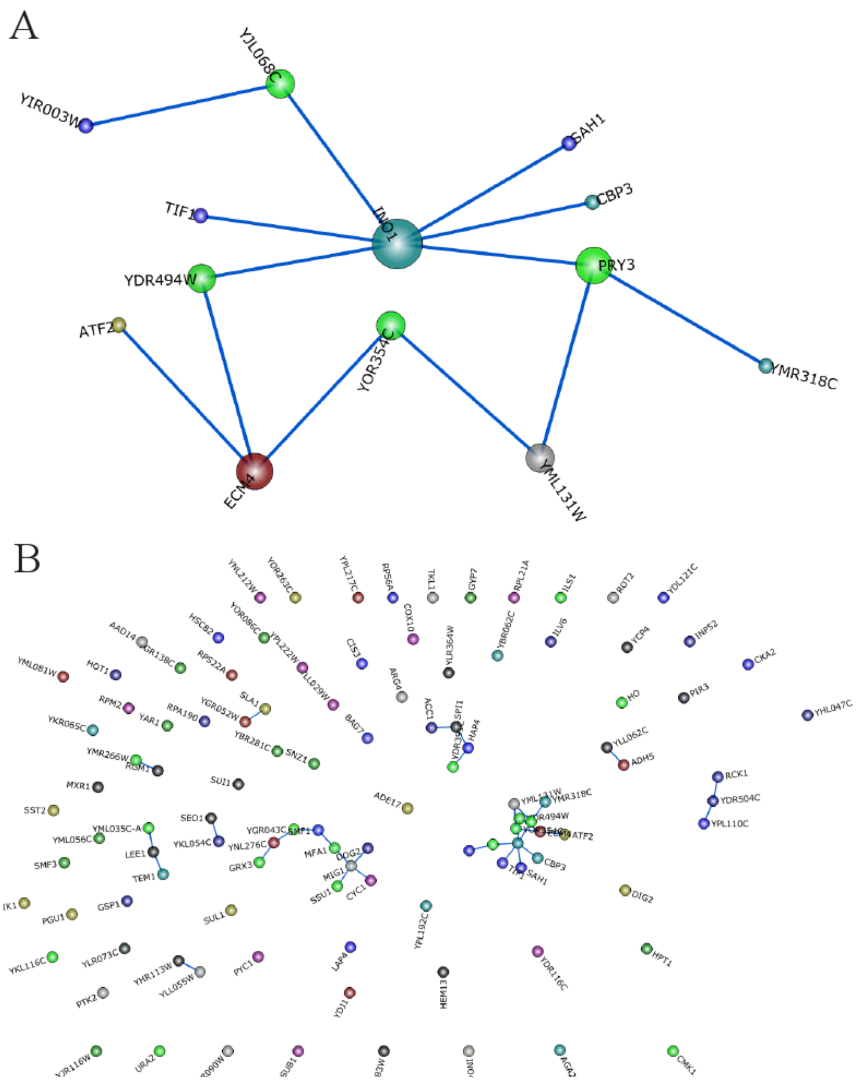


Figure 6.5: Force View. Dynamic relationships between genes can be visualized using a force-directed layout. This view contains the same information as the circle view, but provides a different layout to explore different features in the data. The analyst can produce a force view for each time point. A) A subnetwork of genes is shown. The nodes are sized by degree, *INO1* is clearly shown as a hub gene in the subnetwork. B) When looking at a network in the force-view, connected components of genes cluster together. As the analyst scrolls across time, he can explore how the clusters change.

case, the size of each node represents its degree, thus larger nodes represent hub genes. This view provides a concise representation of the relationships between genes in this subnetwork and updates as the analyst browses across time or space. The force view can also be used to see an overview of the network, as shown in Figure 6.5B. All connected components in the network group together into tight clusters in the force view, allowing the analyst to see how many connected components (clusters) are in the network, how these gene clusters evolve over time, and how many genes are in each cluster. By selecting any of the clusters, all other gene clusters are filtered out and the selected cluster expands to show the relationships between genes in the cluster, as in Figure 6.5A.

6.3.4 Directed Graphs

TVNViewer can be used to visualize both directed and undirected graphs. Directed graphs are valuable if an analyst is interested in cases where the direction of the edge is significant, such as in a regulatory cascade. The initial layout of the graph is not changed in the case of directed graphs for the circle and force views. However, as the analyst hovers over different genes, TVNViewer will highlight all of the gene's in-edges in red, out-edges in green, and bidirectional edges in cyan. If an analyst is interested in one particular gene or gene group, he can select that particular node and TVNViewer will isolate that node and show only the genes connected to it. For example, in Figure 6.6A, *MIG1* is selected in the yeast dataset; all the edges connected to it are highlighted in red indicating that they are in-edges, implying that they may have a regulatory relationship with *MIG1*. However, in Figure 6.6B, the selected node *INO4* has only out-degree nodes since the edges connected to it are green. This suggests that these genes may be regulated by *INO4*. These regulatory relationships may change across time or space, and the analyst can use TVNViewer to trace these relationships using directional information.

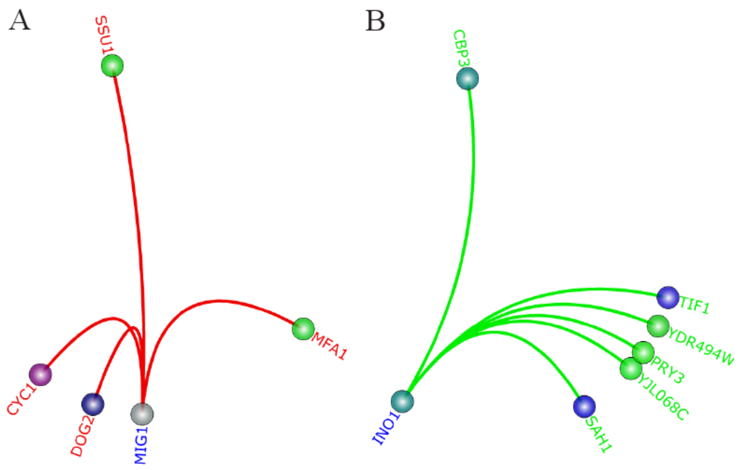


Figure 6.6: Directed edges. TVNViewer also supports datasets with directed or undirected edges. In this case, I show two genes in the yeast dataset with different edge patterns. A) *MIG1*is shown with only in-edges colored red, suggesting that it is regulated by multiple genes. B) *INO1* is shown with only out-edges, suggesting that it regulates the expression of the genes highlighted in green.

6.3.5 Stack view

While the circle and force layouts allow analysts to understand how genetic networks rewire over time or space, these plots are better fit for exploring how specific interactions between genes or gene ontology groups change over time. However, it is often important to explore how the overall network topology and node function changes across time. For example, an analyst may want to explore what cellular processes are active at different stages in the cell cycle. To facilitate this analysis, TVNViewer provides the stack view visualization. In the stack view, I have adopted visualization techniques developed for a very different data set and problem space (selecting a baby's name), taking advantage of features of human visualization processing such as area estimation versus linear magnitude (Wattenberg and Kriss 2006). Further, I implement smooth filtering transitions to drill-down and rescale the plot to match user intent and aid in the analytic process by removing clutter and focusing attention on areas of interest.

In Figure 6.7, I use the stack view to show how the gene activities in a evolving yeast network change in degree over the course of the cell cycle. Genes are grouped using GO annotation. In this view, the out-degree of each GO category is stacked, one on top of the others. Thus, the variation in individual GO categories is clear, and the overall variation in out-degree is emphasized. This visualization clearly shows that the network is active during the G1 phase; by interacting with this view, it is readily observable that genes in the GO categories ATP binding, electron transport chains, and phospholipase C activity are especially active during G1 times.

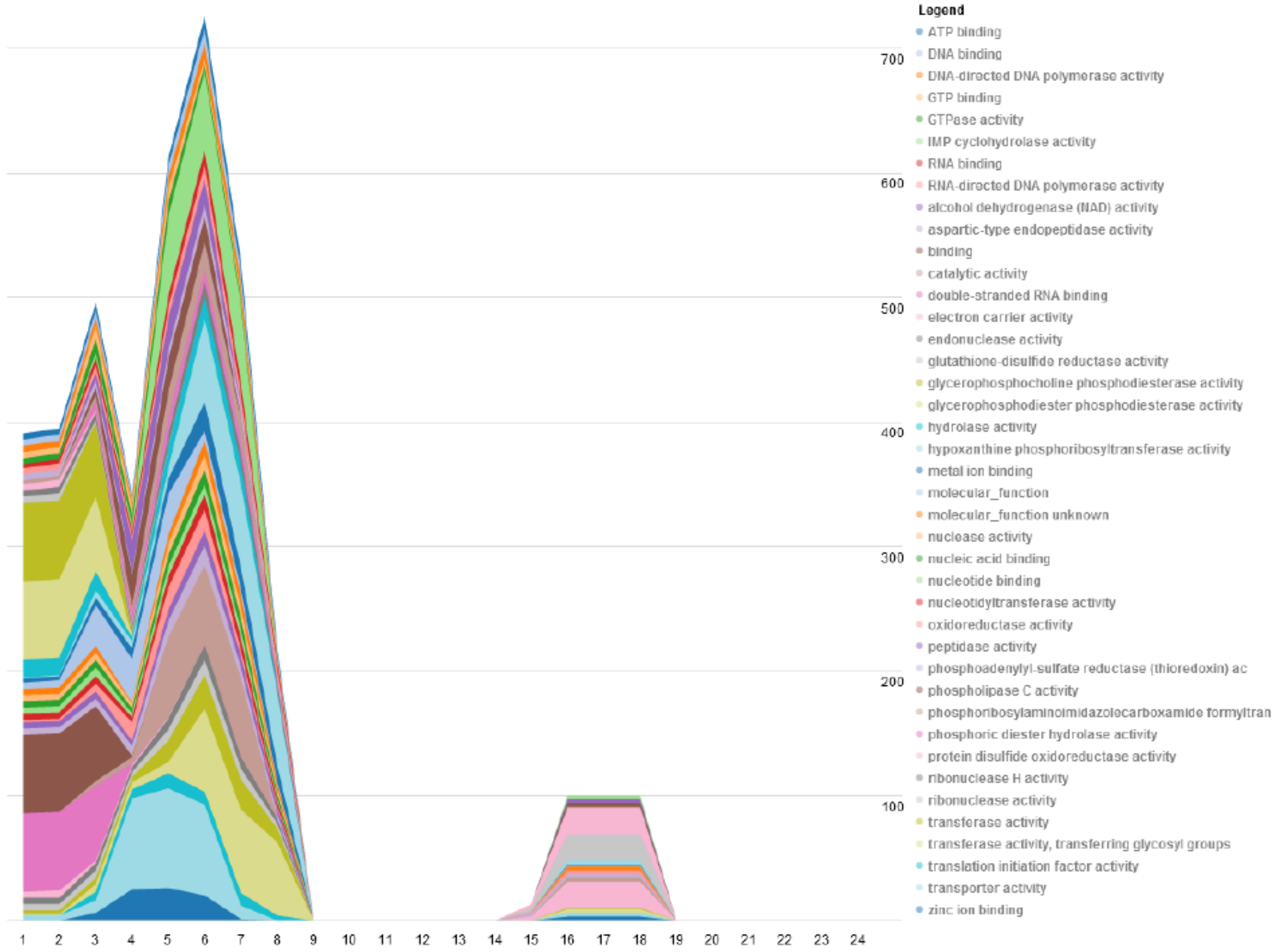


Figure 6.7: Stack View. The stack view allows analysts to get a general overview of how the gene degree of the entire network changes across time. In the yeast cell cycle data, there are two distinct time periods with high activity. It is also observable that different gene groups contribute to the height of the stack differently at different times. By hovering over these groups, the analyst can identify the groups and observe their evolution.

This is expected as these are all functions involved in cellular respiration, which is the signature activity of the G1 phase of the cell cycle. The analyst identifies the gene group by hovering over the category in the stack; a tool tip displays both the GO category and its degree at the given time point.

The stack view can be customized via a filtering system to enable the investigation of interesting patterns in the data. Using the filter box and listing interesting categories in a common delimited list, or by double clicking on the plot, the analyst specifies categories of interest. Additionally, if the analyst is interested in specific genes, he can drill down past the group level and generate stack plots of specified genes of interest.

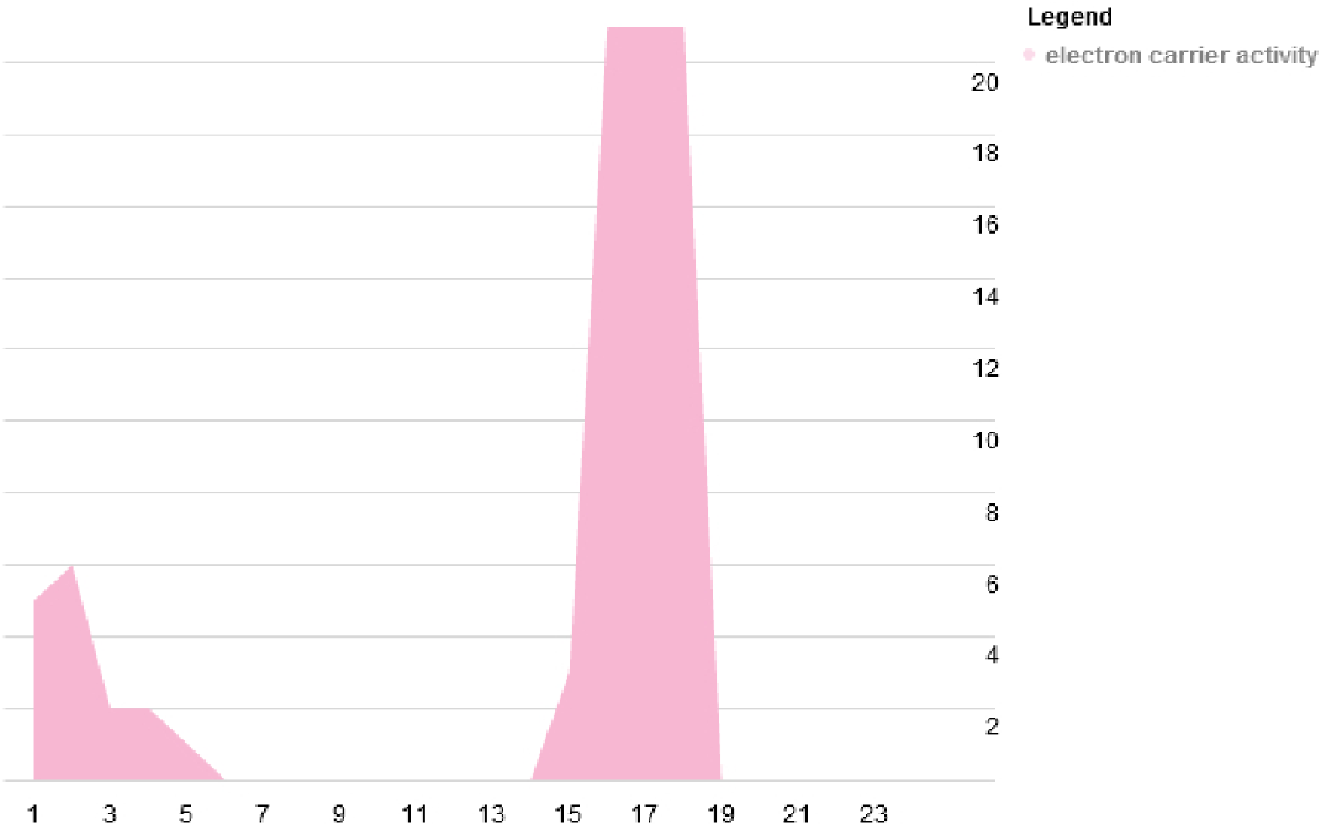


Figure 6.8: Using Filters in the Stack View. TVNViewer allows analysts to filter the stack view to isolate specific gene groups or specific genes and how they evolve across time. Here, the analyst considers how the degree of genes involved in electron carrier activity change across the two yeast cell cycles.

Indeed, selection and filtering are some of the most basic functions that information visualizations should provide for exploratory analysis. Although relatively simple to implement in a visualization, the impact provided by these features is substantial. By allowing analysts to rapidly and simply subset their data while highlighting items of interest, I allow analysts to play "what if" scenarios, which may combine a number of highlights or filters. These visualization features, comparable to dynamic queries, drastically lower the cost of exploring and experimenting with the data and evaluating the outcome of varying queries in comparison to database queries or other approaches (Shneiderman 1994). In Figure 6.8, I show how the stack view can be filtered to examine the electron carrier activity gene group. Figure 6.8 shows that genes related to electron carrier activity are active between time points 1-6 and 14-19. The timing is consistent with G1-phase which occurs at the beginning of each cell cycle. This observation is expected biologically; the cell is growing during G1, and thus cellular respiration, which requires electron carrier activity, should be active.

6.4.6 Timeline View

Although the stack view allows analysts to get a feel for how the overall network changes over time, it is difficult for the analyst to understand how the degree of specific genes changes. To make this information accessible, I implemented the Timeline View, shown in Figure 6.9. The timeline view is similar to the stack view in that it shows the out-degree of each node. However, in this view the degrees are not stacked, but are shown as a linear scatter plot. Hence, the analyst can readily observe how the degree of each gene changes across time or space and also compare trajectories in comparison to other genes. Using a linear view instead of a stack view enhances the analyst's ability to make comparisons instead of examining the contribution of an element to the total effect. As with the stack view, the plot can be filtered to specific genes of interest.

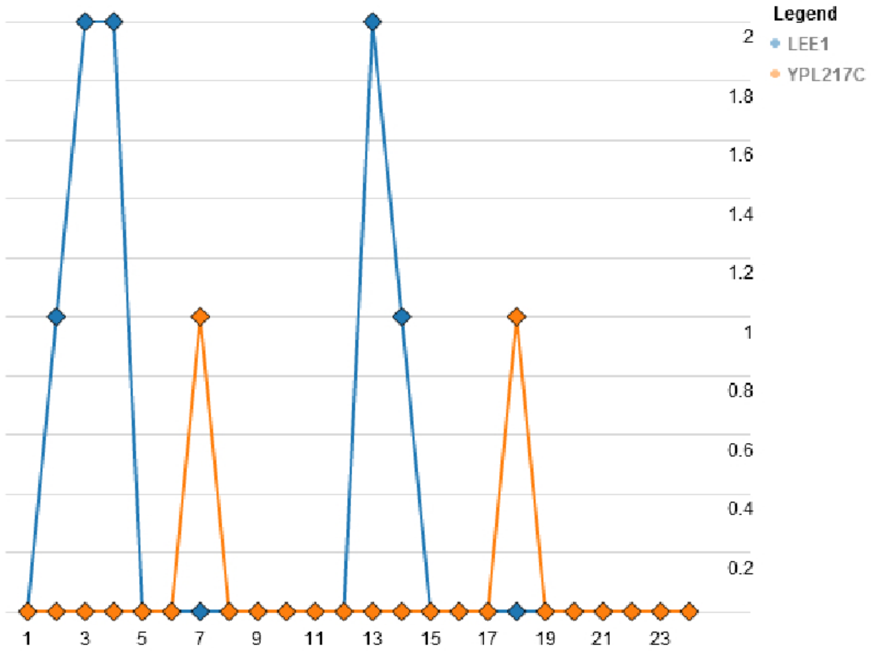


Figure 6.9: Timeline View. In the timeline view, analysts can observe how the degree of gene groups or genes changes over time. For example, in the yeast dataset *LEE1* and *YPL217C* both exhibit a cycling pattern over the two cell cycles with peaks at different points in the cell cycle.

Figure 6.9 shows the timeline view of two selected genes in a yeast network, *LEE1* and *YPL217C*. *LEE1* is shown in blue and peaks at time point 3 and time point 13. However, *YPL217C*, shown in orange, is not active until time point 6 and peaks at time point 7 in the first cell cycle. This suggests *LEE1* is involved in the G1-phase whereas *YPL217C* is active in S-phase. By comparing the activity of genes at different points in the cell cycle, one can determine when certain transcription factors are active, helping to elucidate the roles that these genes may have.

6.4.7 Degree Distribution

TVNViewer allows analysts to further examine macroscopic trends as the network evolves through the degree distribution view. In Figure 6.10 shows the degree distributions over an S1 normal breast cell (S1, shown in A), a T4 malignant cancerous breast cell (B), a breast cell reverted by an MMP inhibitor (C), a breast cell reverted by a PI3K-MAPKK inhibitor (D).

As one can see, the degree distributions change over carcinogenesis progression and reversion. For example the MMP inhibited cells have a larger number of low degree nodes, compared to the other networks. For larger networks, this visualization can be used to access the scale-freeness quality of the networks, compare the degree distributions of in-edges and out-edges, and see how these distributions change over time.

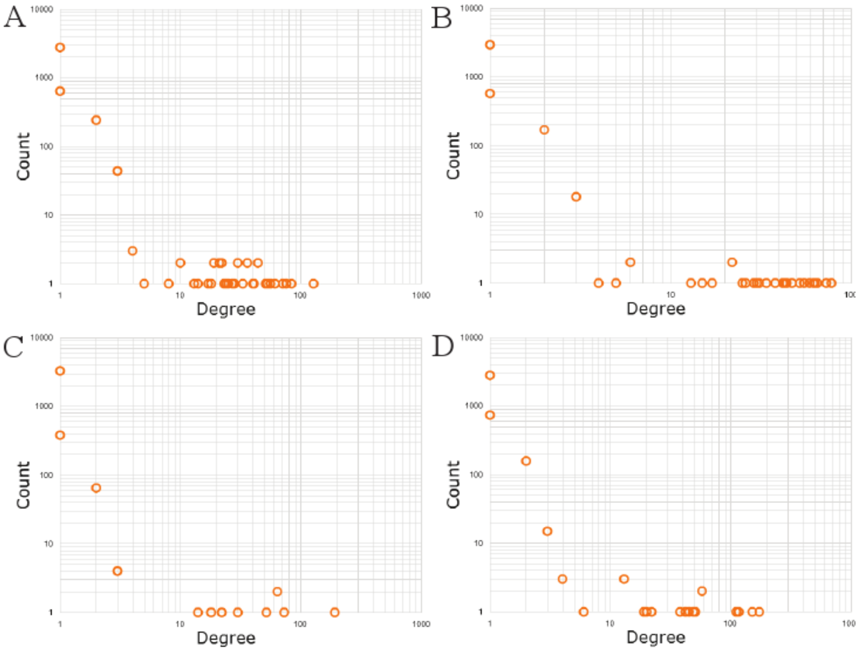


Figure 6.10: Degree Distribution. The distribution of the degree of each node in the network can be displayed as a scatter plot. Here, the distribution of genes in the breast cancer dataset is shown on the log-scale at four states. A) S1 (normal) B) T4 (malignant) C) MMP-T4R (revert) D) PI3K-MAPKK-T4R (revert). One can see that the degree distribution changes as the cell becomes malignant and is then reverted, underscoring the functional differences of each cell state.

6.4 Biological Analysis Using TVNViewer

The first two subsections (Sections 6.4.1 and 6.4.2) in this section are from a paper describing TVNViewer (Curtis et al. 2012). The purpose of these sections in the original paper was to demonstrate TVNViewer through real biological analysis using the previously described (Section 6.3) yeast and breast cancer datasets. These two sections are excellent examples of real biological studies using TVNViewer. They are also excellent examples on how *other* analysts have been able to utilize the tool for biological study. Similarly, the third and fourth subsections (Sections 6.4.3 and 6.4.4), are also derived from biological analysis using TVNViewer (Parikh et al. 2011). Although I was not involved directly in the biological analysis, I worked with the analyst so he could use TVNViewer. I also made suggestions during the drafting of the paper to enhance the use of TVNViewer's visualizations to describe the reported biological results. This section, then, shows how TVNViewer has been used by analysts not involved in its development to discover and present biological results.

6.4.1 Analysis of temporally dependent gene-gene interactions across the yeast cell cycle

Budding yeast (*Saccharomyces cerevisiae)* serves as an excellent model for dynamic network learning because the molecular mechanisms of the cell cycle control system is known in detail (Chen et al. 2004). Budding yeast follows the eukaryotic cell cycle, which is a cyclic progression of events during which the cell grows, replicates its genetic material, and divides into two daughter cells. Each daughter cell then contains all the information necessary to repeat the sequences of events. This process is divided into 4 distinct phases (Cooper 2000). The first is G1-phase (gap 1), which is the interval between mitosis and DNA synthesis. During G1-phase, the cell is metabolically active and is growing. This is followed by S-phase (synthesis) during which DNA replication occurs. The cell continues to grow during the second gap phase G2 (gap 2) and then begins to divide in the M or mitosis phase. For the purposes of our study, we group the G2 and M phase together and refer to it as G2M.

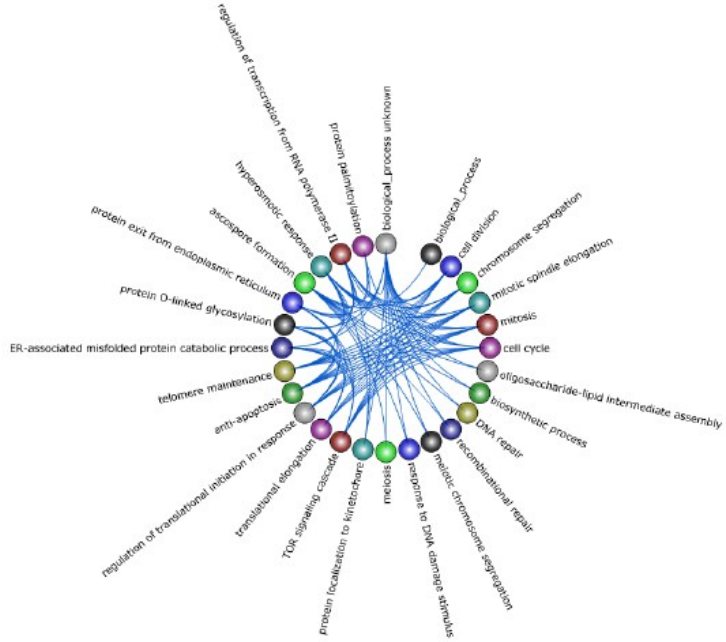

regulation of transcription from RNA polymerase II
protein palmitoylation
biological_process unknown
biological_process
cell division
chromosome segregation
mitotic spindle elongation
mitosis
cell cycle
oligosaccharide-lipid intermediate assembly
biosynthetic process
DNA repair
recombinational repair
meiotic chromosome segregation
response to DNA damage stimulus
meiosis
protein localization to kinetochore
TOR signaling cascade
translational elongation
regulation of translational initiation in response
anti-apoptosis
telomere maintenance
ER-associated misfolded protein catabolic process
protein O-linked glycosylation
protein exit from endoplasmic reticulum
ascospore formation
hyperosmotic response

Figure 6.11: – G2M active genes in yeast The subnetwork shown is a selection of yeast genes that were found to be active during the G2M phase. As a result, the functional groups describe biological processes that occur in the G2-phase of the cell cycle and the final phase which is mitosis. Specifically, groups like DNA repair are indicative of G2-checkpoint and groups such as chromosome segregation annotate genes involved in mitosis.

Studying the yeast cell cycle is a fitting scenario for utilizing TVNViewer as both an exploratory tool and a method of validation. We first generate a series of networks across time from yeast gene expression data using TV-DBN (Song et al. 2009). Then we select sub-networks that are active during certain cell cycle phases and observe their temporal activity as it relates to their function. For example, Figure 6.11 shows a network with a selection of genes that were found to be active during the G2M-phase. Here, we observe functional groups that are clearly relevant. We see GO terms such as cell division, chromosome segregation, mitosis, mitotic spindle elongation, and telomere maintenance. These are events that are characteristic of the mitosis component of the G2M phase. In addition, we also observe functional groups like DNA repair, recombinational repair, and response to DNA damage stimulus which are representative of the G2-phase. There are several checkpoints in the eukaryotic cell cycle that ensure that the genome is complete and correct before cell division occurs. One of the major checkpoints occurs

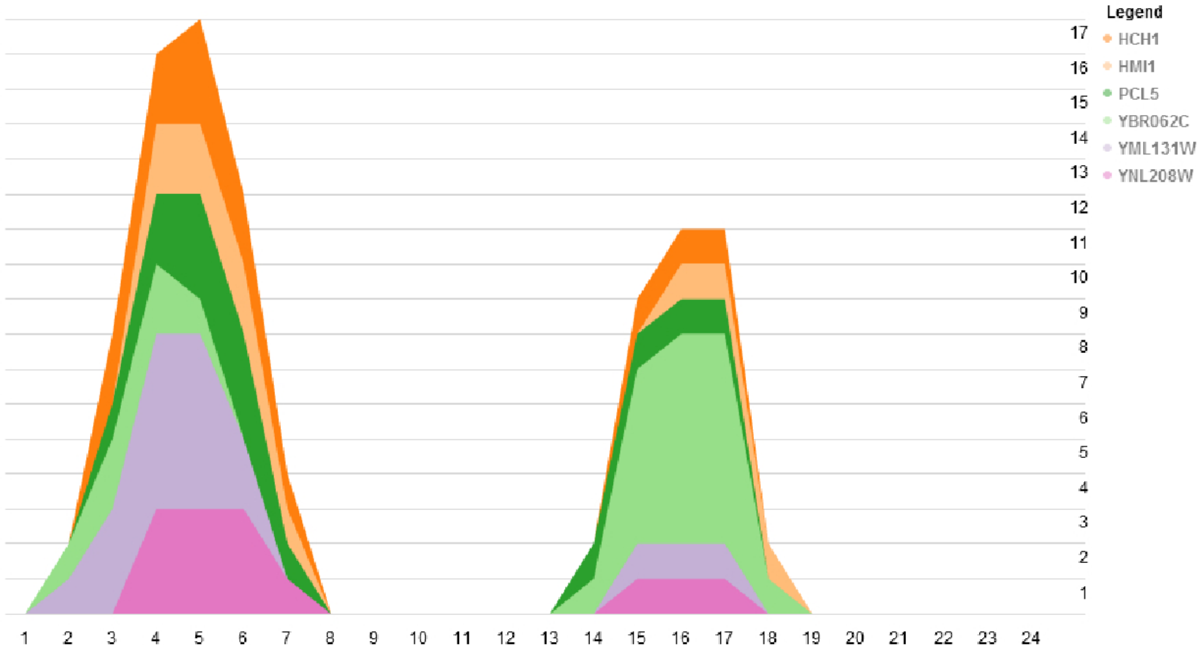


Figure 6.12: – Genes active during S phase in yeast. The plot shown is generated from a selection of yeast genes active in S-phase. The stack view shown illustrates the recurring activity of particular genes over time. Here, we can easily identify the time and shape of interaction patterns that repeat across cell cycles. The peak times are around time points 4-5 in the first cell cycle and 16-17 in the second cell cycle.

in the G2 phase whereby cells are arrested in this phase in response to damaged or unreplicated DNA (Cooper 2000). This allows the cell to repair the damage before cell division continues and the genetic material is passed to the daughter cells. From this information, we can conclude that the functions of the genes in this network are aligned with what we expect from genes that are active in G2M.

An important characteristic of cell cycle data is that it is repetitive. Thus, we should observe recurring patterns in the time-varying networks. For example, a network showing a different selection of genes, active in S-phase, are visualized in Figure 6.12.The colored layers of plots clearly indicate that the interactions between the genes repeat over the two cell cycles; the first cell cycle occurs between time points 1-12 and the second during time points 13-24. Note that the out-degree during the first cell cycle peaks at around time point 4-5 and then 16-17 in the

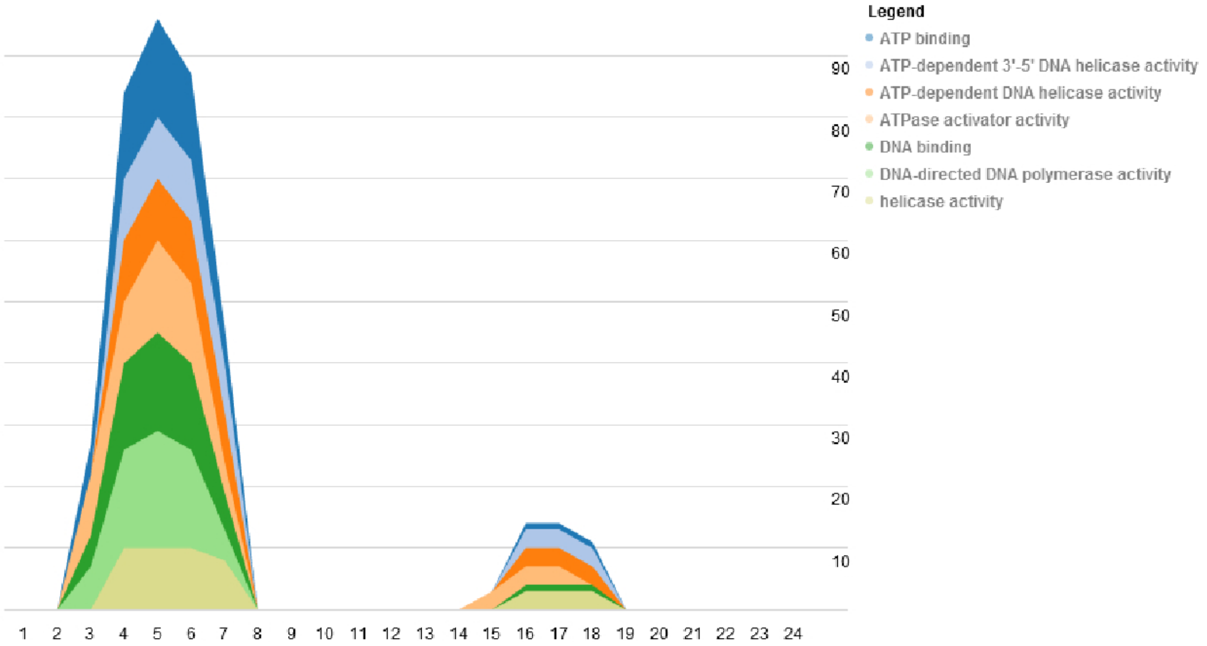


Figure 6.13: Gene functional groups active during S phase in yeast By annotating the genes from Figure 6.12 using GO functional groups, we can observe the recurring functional groups. In this example, DNA binding, helicase activity, and DNA-directed DNA polymerase activity are all molecular mechanisms that we expect to occur during S-phase.

second cell cycle. Thus, the peaks coincide with about the same time in each cell cycle; this indicates that the interactions between genes are consistent for each cell cycle.

If we take the same subnetwork as shown in Figure 6.12 and annotate the genes using GO functional groups, we can observe which functional groups are active over the time series (Figure 6.13). Again, the colored layers show the GO groups peak in the middle of the first cell cycle, then diminish until around the same point in the second cell cycle. The GO terms listed are also relevant to S-phase as they indicate the presence of genes involved in DNA binding, helicase activity and ATP binding. These are all necessary events during DNA synthesis, as helicase uses ATP in order to unwind the double helix in preparation for DNA replication.

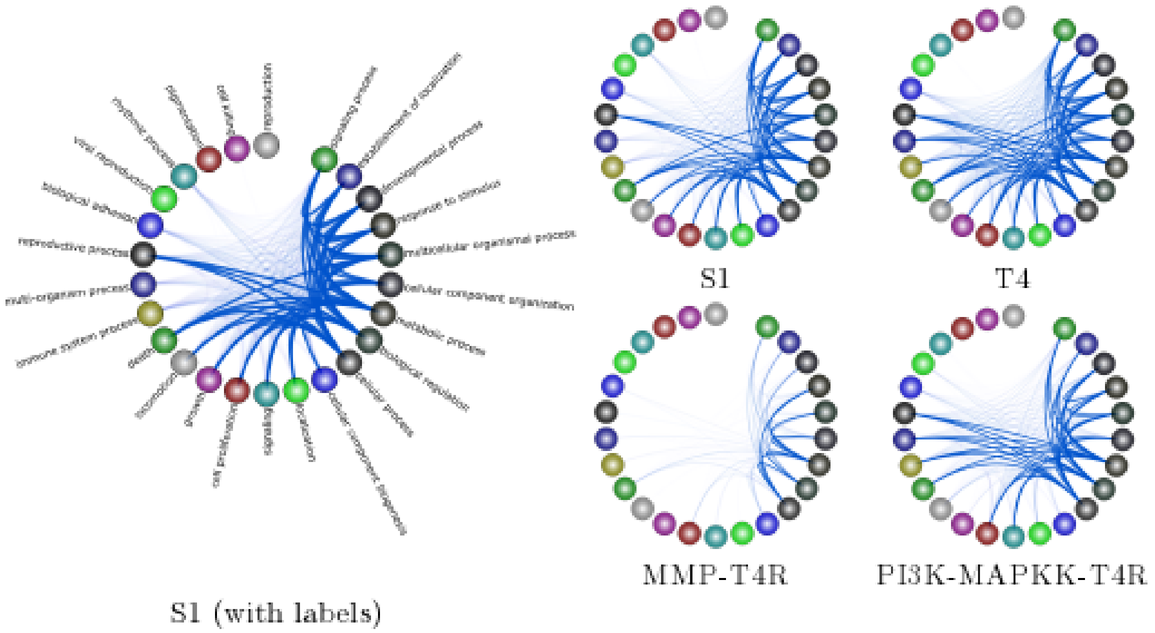
From this preliminary overview of the functional significance of the genes provided by TVNViewer, we can then focus on particular genes and investigate supporting biological literature that can both confirm and explain why these genes interact. For instance, the gene

HMI1 was found to be a DNA helicase and experimental results indicated that it localized in the mitochondria and was required for the maintenance of the functional mitochondrial genome (Sickmann et al. 2003). The unwinding activity of the helicase requires ATP hydrolysis and has a 3’ to 5’ polarity (Monroe et al. 2005). Another gene in the subnetwork is *YNL208W*. While not much is known about the function of *YNL208W*, the protein was detected in purified mitochondria (Sedman et al. 2000). Interestingly, experimental evidence places both *HMI1* and *YNL208W* at the same cellular location, supporting the prediction by our network that these genes interact.

Studying developmental processes such as the yeast cell cycle requires the integration of temporal and functional information. We identify the recurring patterns of the gene sub-networks such as the repetitive activity of particular genes that are pertinent to S-phase. We also generate an overview of the functional roles of the genes in the network and determine whether that is consistent with the timing of the network activity. For example, it makes biological sense that DNA replication occurs in the S-phase. This analysis can be used as a starting point to explore the biological literature in order to link the gene-gene interactions and formulate a summarizing regulatory mechanism.

6.4.2 Exploring the progression and reversal of breast cancer

Using TVNViewer, we also investigate the progression and reversion of breast cancer cells using dynamic network analysis. Functional analysis of 3D culture models of breast cancer has led to a deeper understanding of the effect of a cell’s microenvironment on tumorgenesis and metastasis (Petersen et al. 1992). It was found that micro-environmental factors and signaling pathways have a dramatic influence on the growth dynamics and malignancy of the cells (Weaver et al. 1997; Itoh et al. 2007). Furthermore, treatment with inhibitors of various signaling molecules



S1 T4

MMP-T4R PI3K-MAPKK-T4R

S1 (with labels)

Figure 6.14: Breast cancer analysis using GO biological process functional annotation. Here we present a summary of our results from the analysis of the breast cancer data using the GO biological process functional annotations for the genes. We present the network derived from the original cells (S1), a network derived from the cancer cells (T4), and then networks derived from the reverted cells. Nodes signify GO biological process groups and the relative thickness of the edges between groups represents the number of genes that interact between the two groups.

causes reversion of T4 cells into morphologically-normal-looking cells (T4R cells). Our objective is to analyze the functional differences amongst the different cell states.

We first used Treegl (Parikh et al. 2011) to reverse engineer gene networks for each cell state. As shown in Figure 6.14, compared to S1 cells, T4 cells display increased activities in cell proliferation and signaling, both of which are indicative of cancer. Furthermore, we see that that the T4 network exhibits significantly more interaction with the extracellular matrix and other components related to the cell membrane such as the vesicle (Figure 6.15). This is expected since it has been found that a cell's interaction with its microenvironment affects tumorgenicity and metastasis (Bissell and Labarge 2005). Finally, one can see that the T4 network also displays increased signal transducer activity and oxidoreductase activity (Figure 6.16). Signal transducers and activators of transcription, especially those associated with cytokine and growth factor

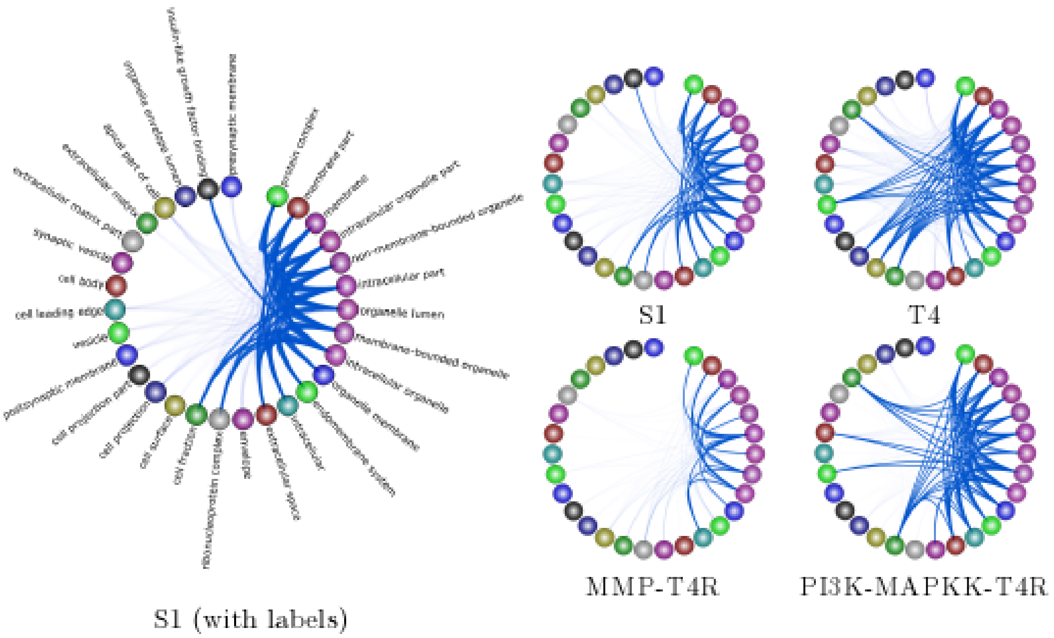


Figure 6.15: Breast cancer analysis using GO cellular component annotation. Here we present a summary of our results from the analysis of the breast cancer data using the GO cellular component annotations for the genes. The network derived from the original cells is denoted by S1, the network from the cancer cells is denoted T4, and the networks from the reverted cells are labeled MMP-T4R and MAPKK-T4R.

activity have been implicated in tumorigenesis (Weaver and Silva 2007). Breast cancer cells are also associated with increased oxidoreductase activity (Adams et al. 1991).

As we can readily observe from the figures, the T4R cells are different from the S1 and T4 cells, but are also distinct from each other. The MMP-T4R network is very sparse and thus has few interactions. Notably, cell proliferation and other indicators of cancer are absent in MMP-T4R cells. On the other hand, the PI3K-MAPKK-T4R cells still display considerable cell proliferation and interaction with the extracellular matrix. PI3K-MAPKK –T4R cells also exhibit more activity such as tetrapyrole binding, demethylase activity and carbohydrate binding, all of which are absent in the other cell states. Collectively, these data suggest that although T4 cells can be morphologically reverted back to the normal-looking T4R cells, the underlying molecular mechanisms in the reverted cells are different from those in either S1 or T4 cells and from one another.

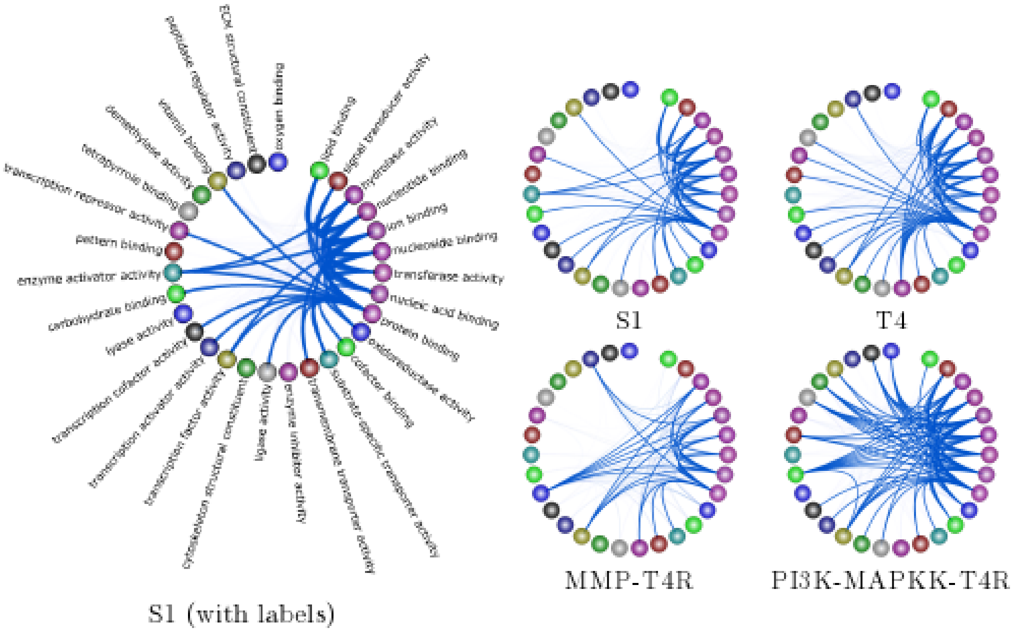

peptidase regulator activity
ECM structural constituent
oxygen binding
vitamin binding
demethylase activity
tetrapyrrole binding
transcription repressor activity
pattern binding
enzyme activator activity
carbohydrate binding
lyase activity
transcription cofactor activity
transcription activator activity
transcription factor activity
cytoskeleton structural constituent
ligase activity
enzyme inhibitor activity
transmembrane transporter activity
substrate-specific transporter activity
cofactor binding
oxidoreductase activity
protein binding
nucleic acid binding
transferase activity
nucleoside binding
ion binding
nucleotide binding
hydrolase activity
signal transducer activity
lipid binding
S1 (with labels)
S1
T4
MMP-T4R
PI3K-MAPKK-T4R

Figure 6.16: Breast cancer analysis using GO molecular function annotation. Here we present a summary of our results from the analysis of the breast cancer data using the GO molecular function annotations for the genes. The network derived from the original cells is denoted by S1, the network from the cancer cells is denoted T4, and the networks from the reverted cells are labeled MMP-T4R and MAPKK-T4R.

6.4.3 Treegl results on breast cancer data set

In this section, we again consider the results from running Treegl to recover the networks in the breast cancer data. The networks exhibit many different topologies reflecting their underlying biological differences. To shed more light on these differences, Figure 6.17 shows the interactions among the second level GO groups in the recovered networks. The thickness of a link between two groups is proportional to the number of edges present between genes that are members of these GO groups. T4 cells display increased activities in cell proliferation and signaling, both indicative of their malignant state, compared to S1 cells. The T4R cells lie somewhere in between: MMP-T4R cells tend to have only a few interactions, since the network is quite sparse. While both the PI3K-MAPKK-T4R and EGFR-ITGB1 networks show reduced

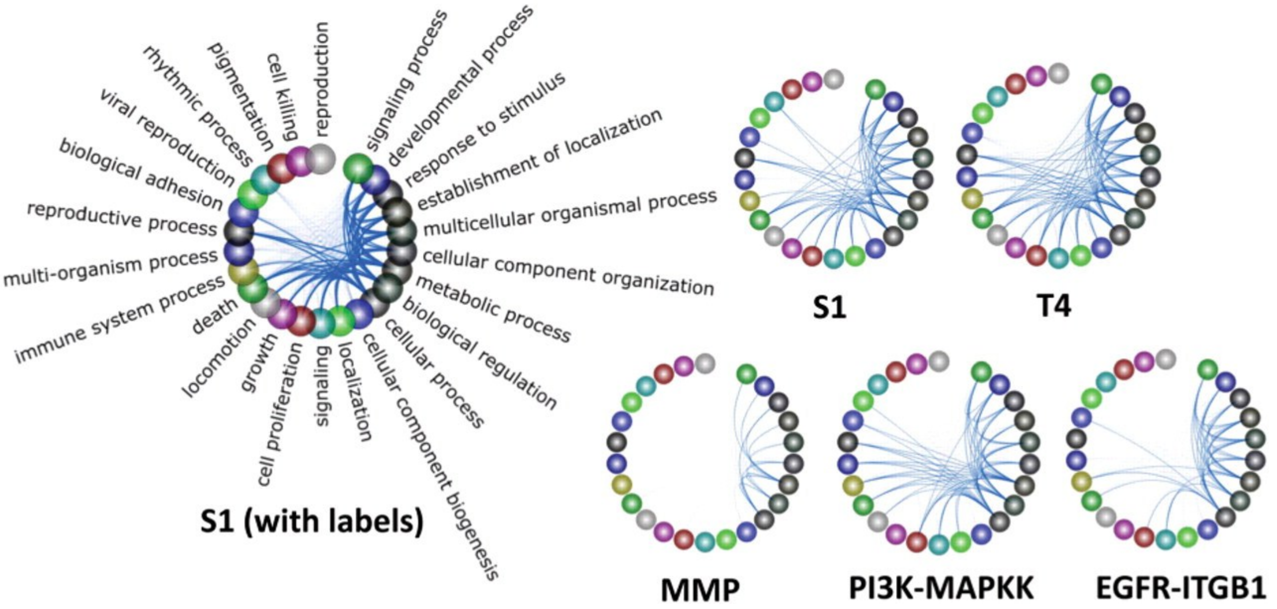


Figure 6.17: Overview of results for the identified networks. Note that the nodes on the circles are not actual genes but correspond to GO process groups. The thickness of a line between two GO groups A and B is proportional to how many genes in A interact with those in B.

activities in growth and locomotion compared to S1 cells, the former network has more activities in cell proliferation and reduced signaling than the latter one. Taken together, these data suggest that although T4 cells can be morphologically reverted back to the normal-looking T4R cells, the underlying molecular mechanisms in the reverted cells are different from those in either S1 or T4 cells.

6.4.4 Analysis of hubs in T4 network

Finally, to identify potential novel drug targets in T4 cells, we examined several hubs which have high degrees as well as their neighborhood genes in these cells. Figure 6.18 shows the sub-networks of 5 hubs: ANXA3, CA9, HSF2BP, PTGS2 and SCG5. As expected, many of the functional gene groups enriched in the sub-networks reflect our intuition that these hubs interact closely with genes influential in cancer.

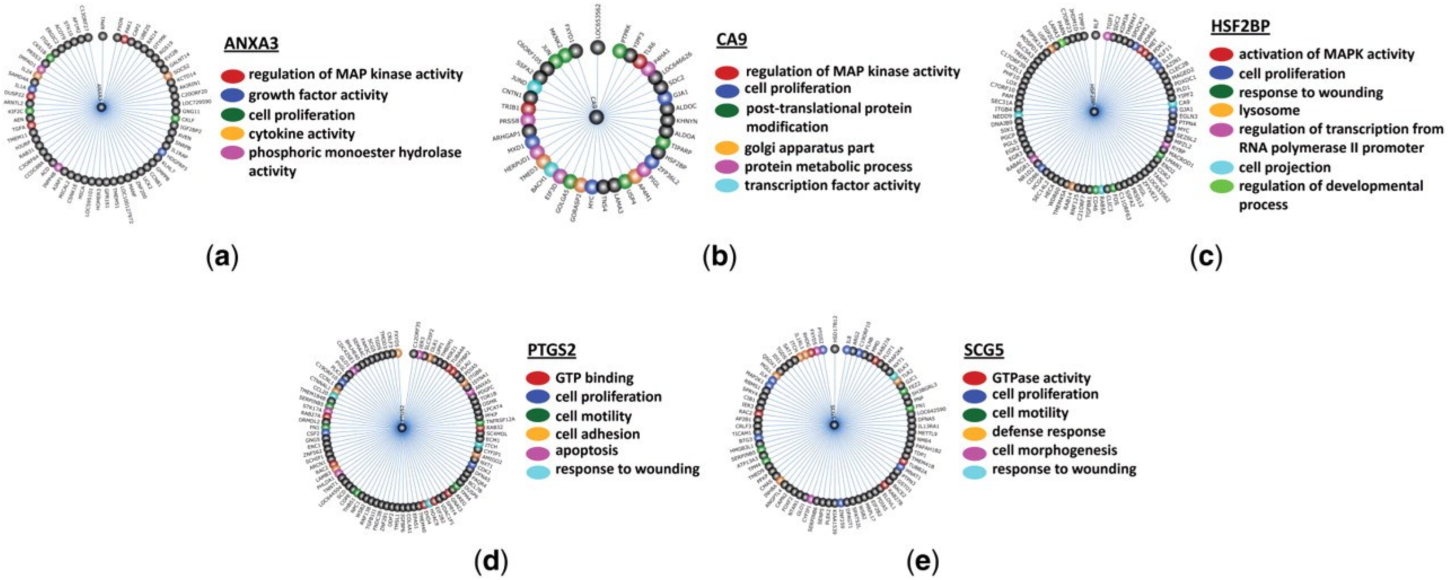


Figure 6.18: Neighborhoods of a few high-degree hubs in T4 cells. A few enriched GO groups are highlighted in the sub-networks as shown.

1. **ANXA3** (degree: 61)—encodes a protein belonging to the annexin family, and is known to play a role in the regulation of cell growth and is thought to be a biomarker of cancer (Jung et al. 2010). In the ANXA3-subnetwork, it interacts with a number of genes related to cell proliferation, growth factor activity, and the MAP kinase signaling pathway, the latter of which is known to be one of the key signaling pathways in T4 cells (Liu et al. 2004).
2. **CA9** (degree: 37)—encodes carbonic anhydrase IX. It has been implicated in cell proliferation, and has been found to be important in renal cell carcinoma (Jubb et al. 2004). We see that CA9's neighborhood consists of genes involved in cell proliferation, the MAP kinase signaling pathway, golgi apparatus part, and transcription factor activity.
3. **HSF2BP** (degree: 80)—encodes heat shock transcription factor binding protein. Like the previous two hubs, HSF2BP has neighbors related to cell proliferation and the MAP kinase signaling pathway. It also has neighbors related to ‘response to wounding’ which

is known to be linked with tumorigenesis and tumor development (Chang et al. 2005; Fukumura et al. 1998).

4. **PTGS2** (degree: 88)—encodes prostaglandin-endoperoxide synthase 2, which is a key enzyme in prostaglandin biosynthesis. Previous evidence suggests that it is associated with risk of breast cancer (Langsenlehner et al. 2006). Again, we see neighbors participating in similar activities to the previous hubs, such as cell proliferation and wound healing. Another interesting group is cell motility which suggests that the subnetwork of PTGS2 potentially plays a role in tumor cell spread (Yamazaki et al. 2005).
5. **SCG5** (degree: 78)—encodes secretogranin V, which has been found to be involved in medullary carcinoma (Marcinkiewicz et al. 1988) as well as human lung cancer (Roebroek et al. 1989). Again many of its neighbors are involved in cell proliferation, response to wound healing, and cell motility. Another interesting group of neighbors is those related to GTPase activity; as oncogenes happen to be members of the family of GTPases (Sahai and Marshall 2002), this group of genes may also have activities implicated in cancer.

In summary, these results suggest that hubs with high degrees in the T4 network contribute to the growth, proliferation, and malignancy of T4 cells, and thus may serve as potential novel targets for breast cancer treatment.

6.5 Use and popularity of TVNViewer

Similar to GenAMap, I have kept statistics on how the TVNViewer website has been hit by visitors. While similar to the statistics kept for GenAMap, TVNViewer is different because an analyst must visit the website each time they run the tool. Over the course of the past 9 months, I have seen the number of visits increase to over 100 per month, with an average of almost 2 minutes on the website each visit. The number of visits peaked in the spring when the first paper describing the TVNViewer paper was published online. The number of visits has held steady since then. The website continues to average more than two page views per visit, despite the large number of returning visitors and relatively small number of pages on the website.

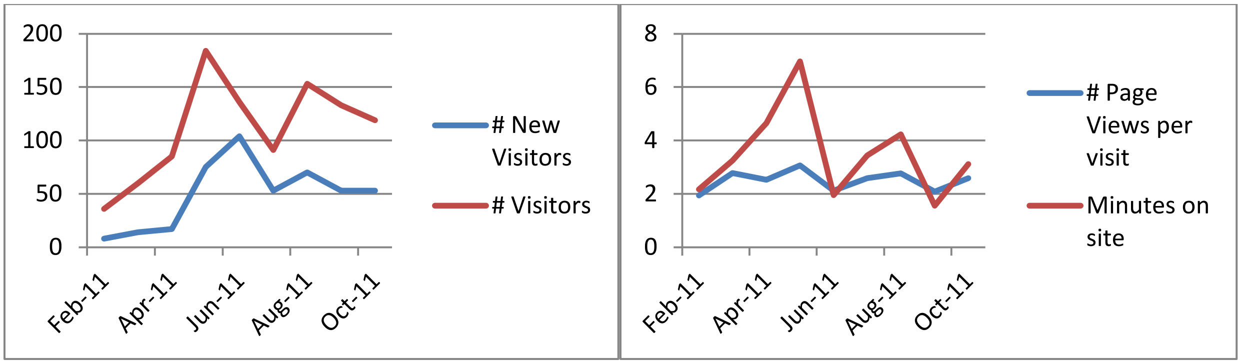


Figure 6.19: Measures of the popularity of TVNViewer on the world wide web. I track the number of visitors that visit the website each month, as well as how long the visitors spend on the site and how many pages they view.

7. Conclusions

In the post-GWAS era, the challenges facing geneticists are varied and difficult. While there is great potential for discovery and advancement with the ever-growing data available to geneticists, it is easy to drown in the multi-dimensional complexity and to slide by without taking advantage of the rich treasure-trove of information. Given sophistication of methods needed to analyze and interpret results from the combination of genome, transcriptome, and phenome data, the inter-reliance between the fields of human genetics, molecular biology, machine learning, and information visualization is paramount. It is the frequent and successful collaboration between researchers in biology, information visualization, and machine learning that will allow geneticists to fully capture the potential of the vast amount of data available.

In this dissertation, I have described the development of GenAMap, a visual analytics software platform for GWAS and eQTL studies. GenAMap is a suite of algorithmic tools that provide ready-to-use access to cutting-edge machine learning research in GWAS and eQTL analysis. Not only have I built GenAMap to provide access to state-of-the art analytic methods, I have also designed visualizations to enable analysts to explore the sea of data that results from these types of algorithms. By building on tried-and-tested visualization principles, I have developed visualization strategies that will enable analysts to explore association results from any analysis. Through multiple-coordinated views, I provide analysts with the ability to explore the structure in the genome, transcriptome, and phenome simultaneously, while considering associations between the data types. I provide instant access to online databases, GO annotations, and association strengths. These tools enable the analyst to explore the data in ways that would not be possible using command-line query tools.

As the amount of data available to biologists continues to grow at an increasing rate, biological studies will need to rely on advances in large scale statistics and machine learning more and more. The integration of machine learning advances into genetics study could become a bottleneck if the distribution and acceptance of state-of-the-art methods is not improved. In this dissertation, I have also proposed a new deployment strategy that makes the latest machine learning technology in genetics association mapping available to genetics analysts. I have created an automatic processing system called Auto-SAM, which automates five state-of-the-art structured association mapping algorithms. Auto-SAM is also integrated into GenAMap. I have demonstrated that Auto-SAM enables genetics analysts to run a variety of structure and association mapping algorithms without the effort to format the data and customize the implementations. I anticipate that Auto-SAM will enable genetics analysts to incorporate structured association mapping algorithms in their GWAS analysis pipelines, potentially enhancing discovery and leading to new genetics insight.

In addition to the development of GenAMap, in this work, I have reported the development of TVNViewer, a new visualization tool built for exploring the dynamic relationships between genes across a time cycle or in response to environment or disease. I have shown that TVNViewer facilitates the analysis of a yeast and breast cancer dataset in ways that would not be possible using other gene network visualization tools. Specifically, I have created a tool that enables analysts to explore dynamic networks in real time, highlighting important changes in network topology and gene-gene interactions. TVNViewer provides a clean interface that can be used to compare networks, investigate interesting signals, and adjust the display to highlight important information.

In TVNViewer, I have also taken advantage of network statistics such as gene degree distribution and gene degree to provide further analysis tools to enable analysts to explore the general patterns of how these networks rewire. By adding interaction to the visualization, I enable seamless transitions to facilitate data exploration without overwhelming the analyst with information. Enabling and simplifying the process of playing "what if" scenarios allows for rapid hypothesis testing, which can give the analyst a better feel for the dataset while leaving the possibility open to serendipitous insights.

Many problems faced by biologists are a near perfect match for visualization. The data biologists confront are dauntingly vast. Machine learning and statistics have enabled us to gain invaluable comprehension over these data and allow us to test specific theories or hypotheses. Visualization is naturally the next step, as it has been demonstrated that visualization can greatly aid in exploratory data analysis when the insight or information of interest cannot be extracted automatically. Thus, visual analytics systems that combine the strengths of machine learning and visualization have the potential to greatly enhance genetics analysis.

To combat the increasing complexity of genetics analysis, I argue that research must follow a pattern of collaboration and cooperation between disciplines, even those as vastly different as genetics, information visualization, and machine learning. I believe that GenAMap and TVNViewer serve as an exemplary case of this type of multi-disciplinary collaboration to build a suite of tools and visualizations based on cutting-edge machine learning technology.

7.1 Lessons learned from the user study

The results from my qualitative user study suggest that GenAMap made analyzing results from structured association mapping easier and saved time, while providing additional insight. The users in the study liked how GenAMap incorporated multiple views to provide a feel for the structure of the genome and the traits while exploring the associations. They felt that the coordinated visualization helped to put things into perspective and avoided unnecessary and awkward integration of specialized tools. Users also felt that GenAMap helped them to focus their attention on important associations and that GenAMap was an improvement over the command line scripts they normally use. GenAMap also has several resources to link to outside information such as GO annotation, SNP pages, and UniProt. These proved to be a key feature in GenAMap to aid researchers in the analysis of the data. GenAMap was able to help the users focus their attention on the important signals, and then quickly direct their attention to the outside sources that could explain the signals.

Despite the improvement that GenAMap has over current applications, the user study led to further development in GenAMap. Specifically, I continued the development of GenAMap to add more links to outside information, as requested by the users. Additionally, even though GenAMap incorporates much of the pipeline for association analysis, the participants in our study suggested that they need to export the data for additional specialized analysis. I have now incorporated this feature in the tool. Additionally, I have worked to provide more legends, keys, and consistency to the tool based on user feedback. Although users were able to understand the visualization strategies in GenAMap, many felt that with a few more legends, color bars, and tooltips, the tool would be easier to pick up and use without consulting documentation or tutorials.

My experience with GenAMap and the user study suggests these five general rules for building an adequate biological visualization system: 1) the ability to focus the user's attention on the important information in complex data sets, especially large arrays of multi-dimensional data, 2) the ability to coordinate multiple views when analyzing connections between different data types, 3) the ability to link out directly from the tool to outside information and biological databases in order to strongly integrate into existing work flows, 4) the ability to export intermediate results for further specialized analysis, and 5) intuitive displays with legends, tooltips, and color bars to enable the user to understand the data presented to them. I hope that these guidelines will prove useful in the development of future biological visualizations.

I expect that the development of GenAMap will make the analysis of structured association mapping available to more genetics researchers by moving the analysis away from command line scripts into a visual system where users can explore associations, structure, and small-scale interaction.

7.2 Lessons learned from the tool development

In this dissertation, I have reported on the development of two different tools: GenAMap and TVNViewer. Both of these tools are visualization tools for genetics research, and both of these tools were motivated by advances in machine learning. GenAMap was motivated by the promise of structured association mapping in association research, and TVNViewer was motivated by the promise of dynamic-network recovery and analysis.

Despite the similarities between the motivations behind the two software platforms, the actual developmental and deployment strategies were quite different. TVNViewer was built as an open-source, website-based tool. GenAMap was built as a distributable desktop application that stores data and runs algorithms on a distributed webserver. TVNViewer was built as a postanalysis visualization tool; while GenAMap was built to not only provide exploratory visualizations, but also to automate the execution of structured association mapping algorithms.

Besides the differences in development architecture, the tools have also followed different paths in how they have evolved into the research community. In this section, I will speculate on the differences between the tools and how their development is illustrative of software applications research.

7.2.1 Problems solved

The development of a visual analytics tool requires specific design decisions on three different levels: data management, analysis tools, and visualizations (Fekete et al. 2011). Different toolkits and development strategies are better at different things – for example, one toolkit might be better at visualizing large datasets while another might be better for complex statistical analysis on smaller datasets. Balancing these three dimensions in a toolkit is difficult, and a perfect solution has not been created yet. Thus, the decision of what toolkit to use is made early in the design and development of the tool, and has a broad impact on how the tool is developed. In terms of these three dimensions of visual analytics software, my approach to the design of TVNViewer and GenAMap has been different.

In TVNViewer, I took an approach that focused primarily on the visualization of small to moderately sized datasets. By limiting the size of the data to around 5000 nodes, I focused primarily on the visualization, and thus I didn't need to worry about designing the tool to handle large amounts of data. I also required all network construction (analysis) steps to be performed independently, outside the tool itself, isolating the tool from computationally intense processes. Thus, TVNViewer is a visualization tool built for the exploration of the rewiring behavior of dynamic networks. The decision to focus primarily on visualization allowed me to use a powerful visualization toolkit (Flare). This focused strategy appears to have been effective, as the TVNViewer website has been visited almost 1000 times (995 times) from Feb 1 to Oct 31, 2011, with nearly 3000 page views.

On the other hand, in GenAMap I approached the problem of creating a software platform for structured association mapping from all three levels. GenAMap has its own data management system that allows analysts to explore and analyze datasets up to 20,000 SNPs or genes. GenAMap is integrated with a complex, parallel-processing environment to run structured association mapping algorithms; and GenAMap is also a visualization tool that provides visualizations for very small to large datasets. Because I have attacked the problem at all three levels, the design of GenAMap was significantly different than TVNViewer, especially in the design of the local/distributed system architecture of the software. My decisions on each level also affected the other levels as well. As one example of the interdependence between layers, structured association mapping algorithms are not scalable (currently) to a genome-wide scale, and so I made the decision to not build GenAMap to support genome-wide scale data in its data management or visualization system. To date, while GenAMap has not been as popular as TVNViewer, it has drawn considerable interest with 396 visits and 630 page views from Feb 1 to Oct 31, 2011, with six users requesting accounts to the software during that time.

7.2.2 Development Strategies

The field of visual analytics is a growing field, and the problems that researchers are studying are varied and interesting. Perhaps the most common approach to building a visual analytics system is the collaboration between the software research team and the analysts. For example, Kulkami et al. collaborated closely with a team of biologists to develop an end-to-end system for analyzing time-lapse images of neural cells (2011); Albers et al. used established visualization principles to build a genomic alignment visualization system in collaboration with four groups of biologists that wanted to explore large genome alignment datasets (2011); and Malik et al. collaborated with the US Coast Guard to build a system that helps manage maritime resources (2011). Many (or most) visual analytics systems result from collaborations between computer scientists and analysts. It has been noted that an important lesson for building visual analytics systems is to “maintain a steady exchange of ideas” between developers and the analysts during all stages of development (Bertini et al. 2011).

TVNViewer was built using this model of collaboration between developer and analyst. TVNViewer was born when Dr. Le Song approached me to help solve unmet visualization needs in dynamic network analysis. I maintained close contact with Dr. Song and others that were performing these kinds of analyses through the development of TVNViewer. Upon the completion of the development of the tool, it has been used for other biological studies in yeast and breast cancer by my collaborators.

On the other hand, GenAMap deviates from this paradigm of visual analytics tool development. Instead of collaborating with others who were performing structured association analysis, I was the one who was performing the structured association analyses. This strategy deviates from the norm of building a visual analytics system: in this rare scenario, the analyst is the developer. There are obvious advantages to this strategy, as there is no disconnect in communication between the biologist and the software engineer. However, the strategy could be held up by a potential disconnect between my analysis and the analysis strategies of other researchers. It thus remains to be seen if the methods I have used to solve my own analytic problems extend well to other researchers.

To expound on this point, a recent study found several roadblocks that analysts have using new visual analytics software systems (Kwon et al. 2011). Of note, one of the road blocks is that many analysts wanted the software to be able to perform analyses that were not built into the software. The authors called this roadblock: "Failure to match expectations and functionality." The authors demonstrate this point by describing users who would spend a lot of time trying to figure out how to make the software do what it was simply not able to do, saying: "There has to be a way …" If this is indeed a common roadblock, one might reason that every analyst comes to a new software system with a certain perception of how to use the software and what kind of conclusions they want to uncover. In fact, reflecting on my own user study, there were cases where users would want to perform analyses that I had not considered to include in GenAMap. Also, although the interpretation of the visualizations in GenAMap were obvious to me, they were not always obvious to others. This point is interesting to consider in light of the model of the development of GenAMap. Will other analysts approach structured association mapping in the same way that I approached it? If this is the case, at least in part, then the tools built into GenAMap will be a natural extension of the analysis. As the GenAMap system acquires more users, this will be an interesting point to consider, shedding light on the advantages and disadvantages of the analyst building the visual analytics system himself. I have attempted to overcome this problem by providing extensive tutorials (more than 45 minutes of video) on the GenAMap website. In any case, this model of development supported the creation of a system that enabled the biological analyses that I performed.

7.3 Future Work

My work with GenAMap has shown that structured association mapping, coupled with powerful visualization tools, has the potential to accelerate biological discovery. However, the initial work with GenAMap is only the beginning of what can become a powerful tool for structured association analysis in the future. In particular, future work with GenAMap can build on the continued development of structured association mapping algorithms, visualizations built for other dimensions of association data, and improved collaboration with genetics analysts.

The development of GFlasso, TreeLasso, and MPGL is only the beginning of structured association mapping algorithms. The potential of this new generation of GWAS algorithms to enhance genetics analysis by leveraging structure inherent in biological data continues to motivate the development of improved and more powerful algorithms. As structured association mapping matures, the new algorithms will need to be incorporated into GenAMap. The incorporation of these algorithms will be facilitated by GenAMap's flexible architecture; however, there may be new challenges arise with the development of these algorithms. Additionally, as the algorithms scale-up to larger datasets, GenAMap will also need to adjust with smarter data management to quickly store and access larger datasets.

Similarly, the development of the visualizations in GenAMap is only the beginning of what can become a powerful suite of visualization tools for GWAS and eQTL analysis. GenAMap supports the visualization of genetic networks, population structure, and feature data for SNPs. However, there are other dimensions of GWAS data that can be visualized and exploited in association analysis. For example, as structured association mapping begins to take advantage of other genomic structures, such as LD and epistasis, the visualizations in GenAMap can be adapted, expanded, or added to in order to present these data types and allow analysts to explore these data structures while considering associations. Also, as the visualization of networks improves, the visualizations in GenAMap can be enhanced to incorporated external data sources. For example, GenAMap could be expanded to automatically query the web for genomic locations of genes in an analysis. Then, GenAMap could report statistics of *cis* and *trans* associations found in an analysis.

Perhaps GenAMap will be most improved as the user base expands to more genetics analysts who use and give feedback on the visualizations and their utility. Input from new users will improve the usability of the software. Also, my experience with the user study suggested that each analyst comes into an association study with a different perspective on association studies and the types of analyses that they want to perform. GenAMap will benefit greatly from the additional insights and perspectives of external collaborators.

7.4 Final Word

In this dissertation, I have contributed new two new tools for genetics analysis. TVNViewer enables dynamic network analysis by providing new visualizations to explore how genetic networks rewire across time and in response to different environments. GenAMap proposes a new model of machine learning algorithm development, where algorithms are appropriately scaled and automated such that they are accessible to non-specialized users. GenAMap also proposes new visualization strategies that use multiple-coordinated views to explore the structure of the genome and multiple-correlated traits while considering genome and phenome structure. I have presented a new visualization paradigm in GenAMap that enables analysts to explore joint three-way genome-transcriptome-phenome associations. Finally, I demonstrate GenAMap through the discovery of interacting hotspots in yeast, the discovery of a potentially new asthma gene in humans, and the discovery of genome-gene-trait associations in a mouse dataset.

Contributing Work

This work is based on eight papers and one abstract that have been published or are currently under review. Because the text for this dissertation is mostly derived from these publications, I briefly outline my contributions to each of these projects to ensure that my contribution to the work is clearly stated.

1) Curtis RE, Yin J, Kinnaird P, Xing EP. 2012. Finding genome-transcriptome-phenome association with structured association mapping and visualization in GenAMap. Pacific Symposium on Biocomputing 17:327-338.
2) Curtis RE, Kinnaird P, Xing EP. 2011. GenAMap: Visualization Strategies for Structured Association Mapping. *IEEE Symposium on Biological Visualization* 1:87-95.
3) Curtis RE, Yuen A, Song L, Goyal A, Xing EP. 2011. TVNViewer: An interactive visualization tool for exploring networks that change over time or space. *Bioinformatics* 27(13):1880-1881.
4) Parikh A, Wu W, Curtis RE, Xing EP, Reverse Engineering Tree-Evolving Gene Networks Underlying Developing Biological Lineages, *the Nineteenth International*

Conference on Intelligence Systems for Molecular Biology (ISMB 2011). *Bioinformatics* 27(13):i196-i204.

5) Curtis RE, Goyal A, Xing EP. Enhancing the usability and performance of structured association mapping algorithms using automation, parallelization, and visualization in the GenAMap software system. *Under review.*
6) Curtis RE, Xiang J, Parikh A, Kinnaird P, Xing EP. Enabling dynamic network analysis through visualization tools in TVNViewer. *Under Review.*
7) Curtis RE, Kinnaird P, Kim S, Lee S, Yin J, Puniyani K, Wenzel S, Bleecker E, Meyers DA, Xing EP. GenAMap: visual analytics software for structured association mapping. *Under Review*.
8) Curtis RE, Kim S, Woolford J, Xu W, Xing EP. GFlasso analysis on yeast data uncovers multiple interacting eQTL hotspots. *Under Review.*
9) Curtis RE, Wenzel S, Meyers DA, Bleecker E, Xing EP. 2011. Population analysis of asthma genome-wide association data using GenAMap (Abstract #688T). Presented at the 12th International Congress of Human Genetics/61st Annual Meeting of The American Society of Human Genetics, October 13, 2011, Montreal, Canada.

Contribution to 1: This was a collaborative project between Peter Kinnaird of the Human Computer Interaction Institute, Junming Yin of the Lane Center, and me. The project was my idea; I wanted to lead a project that would allow for the development of visualization strategies for three-way genome-transcriptome-phenome associations. Peter and I discussed the design of the visualizations, after which I implemented and refined our design in GenAMap. Junming and I planned the algorithmic strategy, the two-way GFlasso-gGFlasso. Junming derived the update equations, and I implemented the algorithm. Junming planned the simulation, and I ran the experiments. I drafted the text with input from Peter and Junming, and I performed the biological analysis.

Contribution to 2: This project was the first paper on GenAMap, presented to the visualization community. Peter Kinnaird helped me to plan the presentation and to plan the user study. He also provided some input into the text. In this paper, I was the primary contributor who developed the software, wrote the text, and conducted the user evaluation.

Contribution to 3: This is the first paper describing TVNViewer. I was the lead author on the project; Anuj and Amos were able to contribute significantly in their roles as code developers, and Le was instrumental in getting the project started. I drafted the manuscript, came up with the initial prototypes for the software, and led the development of the software.

Contribution to 4: This was the first paper that was not a visualization paper that presented results using TVNViewer. In this paper, Ankur Parikh describes a new algorithm called Treegl and applies it to a breast cancer dataset. I had some input into the manuscript, but my primary role was to work with Ankur to teach him how to use TVNViewer. I suggested the visualizations that would best show his data, and two of his figures in the paper are a result from our collaboration.

Contribution to 5: This paper describes my initial work designing the GenAMap automation system, which I call Auto-SAM. After I had the system up and running, Anuj helped to incorporate some of the newer algorithms into Auto-SAM, which is why he is also listed as an author. I designed the system, implemented most of it, and drafted the manuscript.

Contribution to 6: This paper was a collaborative work between me, Jing Xiang, and Ankur Parikh. We wanted to present TVNViewer in a systematic way with real data to show off the full capabilities of the software. Jing was a co-lead on the project because of her experience working with the software on biological data. This was a truly collaborative work as we planned the paper strategy and then worked together to write the text and create the figures for each section. Peter Kinnaird also contributed insight from the visualization community as we drafted the work.

Contribution to 7: This is the work in which I explain the full GenAMap software and its contribution to biology. The co-authors on the paper had input into the presentation of my work and helped to lay its foundations through algorithm development and data sharing. In this paper, I write up use-cases of the software, demonstrate the different tools I have incorporated into the software, and also perform a biological analysis using GenAMap.

Contribution to 8: This was my first project at CMU. Dr. Seyoung Kim was my mentor on the project and helped me to get it started and provided feedback throughout the project. Dr. John Woolford contributed many hours looking over results and discussing possible experiments. Dr. Woolford carried out experiments in collaboration with Dr. Wenjie Xu. I directed the project, ran the analyses, and drafted the text.

Contribution to 9: This project was an analysis using GenAMap on the asthma data. I designed new visualization strategies for exploring interactive Manhattan Plots. I also performed biological analyses that led to the discovery of the asthma gene. Dr. Sally Wenzel provided access to the data and provided feedback throughout the process.

Bibliography

Adams EF, Rafferty B, White MC. 1991. Interleukin 6 is secreted by breast fibroblasts and stimulates 17β-oestradiol oxidoreductase activity in MCF-7 cells: possible paracrine regulation of breast oestradiol levels. *Int J Cancer* **49**: 118-121.

Ahmed A, Xing EP. 2009. Recovering time-varying networks of dependencies in social and biological studies. *PNAS* **106**(29): 11878.

Albers D, Dewey C, Gleicher M. 2011. Sequence Surveyor: Leveraging Overview for Scalable Genomic Alignment Visualization. *IEEE Transacations on Visualization and Computer Graphics* **17**(12): 2392-2401.

Anderson CA, Boucher G, Lees CW, Franke A, D'Amato M, Taylor KD, Lee JC, Goyette P, Imielinski M, Latiano A et al. 2011. Meta-analysis identifies 29 additional ulcerative colitis risk loci, increasing the number of confirmed associations to 47. *Nature genetics* doi:10.1038/ng.764.

Basso K, Margolin A, Stolovitzky G, Klein U, Dalla-Favera R, Califano A. 2005. Reverse engineering of regulatory networks in human B cells. *Nature Genetics* **37**(4): 382-390.

Bertini E, Perer A, Plaisant C, Santucci. 2008. BELIV'08: Beyong time and errors: novel evaluation methods for information visualization. In *CHI '08 extended abstracts on Human factors in computing systems*.

Bertini E, Strobelt H, Braun J, Deussen O, Groth U, Mayer TU, Merhof D. 2011. HiTSEE: A Visualization Tool for Hit Selection and Analysis in High-Throughput Screening Experiments. *IEEE Symposium on Biological Data Visualization* **1**: 95-102.

Binns D, Dimmer E, Huntley R, Barrell D, O'Donovan C, Apweiler R. 2009. QuickGO: a web-based tool for Gene Ontology searching. *Bioinformatics* **25**(22): 3046-6.

Birol I, Jackman SD, Nielsen CB, Qian JQ, Varhol R, Stazyk G, Morin RD, Zhao Y, Hirst M, Schein JE et al. 2011. De novo transcriptome assembly with ABySS. *Bioinformatics* **25**(21): 2872-2877.

Bissell MJ, Labarge MA. 2005. Context, tissue plasticity, and cancer; are tumour stem cells also regulated by the microenvironment? *Cancer Cell* **7**: 17-23.

Blake JA, Bult CJ, Kadin JA, Richardson JE, Eppig JT, Mouse Genome Database Group. 2011. The Mouse Genome Database (MGD): a premier model organism resource for mammalian genomics and genetics. *Nucleic Acids Res* **39**(suppl 1): D842-D848.

Boeke JD, Sandmeyer SB. 1991. Yeast transposable elements. In *The molecular and cellular biology of the yeast Saccharomyces: genome dynamics, protein synthesis, and energetics*. (ed. Broach JR, Jones EW, Pringle J) pp.193-261. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York.

Bonsignore EM, Dunne C, Rotman D, Smith M, Capone T, Hansen DL, Shneiderman B. 2009. First Steps to Netviz Nirvana: Evaluating Social Network Analysis with NodeXL. *Computational Science and Engineering* **4**: 332-339.

Breitkreutz BJ, Stark C, Tyers M. 2003. Osprey: a network visualization system. *Genome Biol* **4**(3): R22.

Brem RB, Kruglyak L. 2005. The landscape of genetic complexity across 5700 gene expression traits in yeast. *Proc Natl Acad Sci USA* **102**(5): 1572-1577.

Brem RB, Storey JD, Whittle J, Kruglyak L. 2005. Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature* **436**(7051): 701-703.

Brem RB, Yvert G, Clinton R, Kruglyak L. 2002. Genetic dissection of transcriptional regulation in budding yeast. *Science* **296**(5568): 752-5.

Broad Institute of Harvard and MIT. 2010. Saccharomyces cerevisiae RM11-1a sequencing project. http://www.broad.mit.edu.

Buckingham SD. 2008. Scientific Software: seeing the SNPs between us. *Nature Methods* **5**: 903-908.

Califano A, Butte A, Friend S, Ideker T, Schadt EE. 2011. Integrative Network-based Association Studies: Leveraging cell regulatory models in the post-GWAS era. *Nature Precedings* **5732**(1): doi:10.1038.

Calvano SE, Wenzhong X, Richards DR, Felciano RM, Baker HV, Cho RJ, Chen RO, Brownstein BH, Cobb JP, et al. 2005. A network-based analysis of systemic inflammation in humans. *Nature* **437**: 1032-1037.

Card SK, Mackinlay JD, Shneiderman B. 1998. Information Visualization: Using Vision to Think. *Morgan-Kaufmann* San Francisco, California.

Card S, Mackinlay J, Shneiderman B, Kaufmann M, 1999. *Readings in Information Visualization*.

Carlson CS, Eberle MA, Rieder MJ, Yi Q, Kruglyak L, Nickerson DA. 2004. Selecting a Maximally Informative Set of Single-Nucleotide Polymorphisms for Association Analyses Using Linkage Disequilibrium. *AJHG* **74**(1): 106-120.

Castle T, 2009. *JHeatChart*. [Online] Available at: http://freshmeat.net/projects/jheatchart.

Chang HY, Nuyten DS, Sneddon JB, Hastie T, Tishirani R, Sorlie T, Dai H, He YD, van't Verr LJ, Bartelink H et al. 2005. Robustness, scalability, and integration of a wound-response gene expression signature in predicting breast cancer survival. *PNAS* **102**: 3738.

Chen K.C., Calzone L., Csikasz-Nagy A., Cross F.R., Novak B., Tyson J.J. 2004. Integrative analysis of cell cycle control in budding yeast. *Mol Biol Cell* 3841-62.

Chen X, Kim S, Lin Q, Carbonell JG, Xing EP. 2010. Graph-Structured Multi-task Regression and an Efficient Optimization Method for General Fused Lasso. *CoRR*.

Chen W, Liang L, Abecasis GR. 2009. GWAS GUI: a graphical browser for the results of whole-genome association studies with high-dimensional phenotypes. *Bioinformatics* **25**(2): 284-285.

Chen WY, Song Y, Bai H, Lin CJ, Chang EY. 2010. Parallel Spectral Clustering in Distributed Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* url: http://www.cs.ucsb.edu/~wychen/sc.

Chen Y, Zhu J, Lum PY, Yang X, Pinto S, MacNeil DJ, Zhang C, Lamb J, Edwards S, Sieberts SK et al. 2008. Variations in DNA elucidate molecular networks that cause disease. *Nature* **452**: 429-435.

Chua G, Morris QD, Sopko R, Robinson MD, Ryan O, Chan ET, Frey BJ, Andrews BJ, Boon C, Hughes TR. 2006. Identifying transcription factor functions and targets by phenotypic activiation. *PNAS* **103**(32): 12045-12050.

Cohn M, Blackburn EH. 1995. Telomerase in yeast. *Science* **269**(5222): 396-400.

Cookson W, Liang L, Abecasis G, Moffatt M, Lanthrop M. 2009. Mapping complex disease traits with global gene expression. *Nature Reviews Genetics* **10**: 184-194.

Cooper G.M., 2000. *The Cell, 2nd Ed.* Sunderland (MA): Sinauer Associates.

Cox D, Patterson R, 2011. *NSFNET growth until 1995*. [Online] Available at: http://www.caida.org/projects/internetatlas/gallery/nsfnet/index.xml.

Curcio MJ, Hedge AM, Bocke JD, Garfinkel DJ. 1990. Ty RNA levels determine the spectrum of retrotranposition events that activate gene expression in Saccharomyces cerevisiae. *Mol Gen Genet* **220**: 213-221.

Curtis RE, Goyal A, Xing EP. 2012. Enhancing the usability and performance of structured association mapping algorithms using automation, parallelization, and visualization in the GenAMap software system. *Under Review*.

Curtis RE, Kim S, Woolford JL, Xu W, Xing EP. 2012. GFlasso analysis on yeast data uncovers multiple interacting eQTL hotspots. *Under Review*.

Curtis RE, Kinnaird P, Kim S, Lee S, Yin J, Puniyani K, Wenzel S, Bleecker E, Meyers DA, Xing EP. 2012. GenAMap: visual analytics software for structured association mapping. *Under Review*.

Curtis RE, Kinnaird P, Xing EP. 2011. GenAMap: Visualization Strategies for Association Mapping. *IEEE Symposium on Biological Data Visualization* **1**: 87-95.

Curtis RE, Wenzel S, Myers DA, Bleecker E, Xing EP. 2011. Population analysis of asthma genome-wide association data using GenAMap. *Presented at the 61st Annual Meeting of the American Society of Human Genetics*.

Curtis RE, Xiang J, Parikh A, Kinnaird P, Xing EP. 2012. Enabling dynamic network analysis through visualization in TVNViewer. *Under Review*.

Curtis RE, Xing EP. 2010. GenAMap: An Integrated Analytic and Visualization Platform for GWA and eQTL Analysis. In *Proceedings of the 18th International Conference on Intelligent Systems for Molecular Biology (ISMB)*.

Curtis RE, Yin J, Kinnaird P, Xing EP. 2012. Finding Genome-Transcriptome-Phenome Associations with Structured Association Mapping and Visualization in GenAMap. *Pacific Symposium on Biocomputing* **17**: 327-338.

Curtis RE, Yuen A, Song L, Goyal A, Xing EP. 2011. TVNViewer: An interactive visualization tool for exploring networks that change over time or space. *Bioinformatics* **27**(13): 1880-1881.

Devlin B, Roeder K, Wasserman L. 2003. Analysis of multilocus models of association. *Genetic Epidemiology* **25**: 36-47.

Dilworth DJ, Tackett AJ, Rogers RS, Yi EC, Christmas RH, Smith JJ, Siegel AF, Chait BT, Wozniak RW, Aichison JD. 2005. The mobile nucleoporin Nup2p and chromatin-bound Prp20p function in endogenous NPC-mediated transcriptional control. *J Cell Biol* **171**(6): 955-65.

Dubois PC A, Trynka G, Franke L, Hunt KA, Romanos J, Curtotti A, Zhernakova A, Heap GA R, Adany R, Aromaa A et al. 2010. Multiple common variants for celiac disease influencing immune gene expression. *Nature genetics* **42**: 295-302.

Du P, Kibbe WA, Lin SM. 2008. lumi: a pipeline for processing Illumina microarray. *Bioinformatics* **24**(13): 1547-1548.

Dupuy D, Bertin N, Hidalgo CA, Venkatesan K, Tu D, Lee D, Rosenberg J, Svrzikapa N, Blanc A, Carnec A et al. 2007. Genome-scale analysis of in vivo spatiotemporal promoter activity in Caenorhabditis elegans. *Nat Biotechnol* **25**(6): 663-8.

Efron B, Bastie T, Johnstone I, Tibshirani R. 2004. Least Angle Regression. *Annals of Statistics* **32**(2): 407-499.

Emilsson V, Thorleifsson G, Zhang B, Leonardson AS, Zink F, Zhu J, Carlson S, Helgason A, Walters GB, Gunnarsdottir S et al. 2008. Genetics of gene expression and its effect on disease. *Nature* **452**(27): 423-430.

Fekete JD, Hemery PL, Baudel T, Wood J. 2011. Obious: A Meta-Toolkit to Encapsulate Information Visualization Toolkits - One Toolkit to Bind Them All. *IEEE Conference on Visual Analytics Science and Technology* **6**: 89-98.

Fekete JD, vanWijk JJ, Stasko JT, North C. 2008. The Value of Information Visualization. *LNCS* **4950**: 1-18.

Forer L, Schonherr S, Weissensteiner H, Haider F, Kluckner T, Gieger C, Wichmann HE, Specht G, Kronenberg F, Kloss-Brandstatter A. 2010. CONAN: copy number variation analysis software for genome-wide association studies. *BMC Bioinformatics* **11**(318): doi:10.1186/1471-2105-11-318.

Franke A, McGovern DP B, Barrett JC, Wang K, Radford-Smith GL, Ahmad T, Lees CW, Balschun T, Lee J, Roberts R et al. 2010. Genome-wide meta-analysis increases to 71 the number of confirmed Crohn's disease susceptibility loci. *Nature genetics* **42**: 118-1125.

Friedman J, Hastie T, Tibshirani R. 2007. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**(3): 432-441.

Friedman J, Hastie T, Tibshirani R. 2010. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* **33**(1): 1-22.

Fukumura D, Xavier R, Sugiura T, Chen Y, Park EC, Lu N, Selig M, Nielsen G, Taksir T, Jain RK et al. 1998. Tumor induction of VEGF promoter activity in stromal cells. *Cell* **94**: 715-724.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettlin M, Dudoit S, Ellis B, Gautier L, Ge Y. 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology* **5**: R80.

Ge D, Zhang D, Need AC, Marin O, Fellay J, Telenti A, Goldstein DB. 2008. WGAViewer: Software for Genomic Annotation of Whole Genome Association Studies. *Genome Res* **18**(4): 640-3.

Gilad Y, Rifkin SA, Pritchard JK. 2008. Revealing the architecture of gene regulation: the promise of eQTL studies. *Treds Genet* **24**(8): 408-145.

Gilbert D, 2010. *JFreeChart open source library*. [Online] Available at: http://www.jfree.org/jfreechart/index.html.

Golden Helix, 2011. *About Golden Helix*. [Online] Available at: http://www.goldenhelix.com/Company/about.html [Accessed 2011].

GTEx, 2010. *GTEx (Genotype-Tissue Expression) eQTL Browser*. [Online] Available at: http://www.ncbi.nlm.nih.gov/gtex/test/GTEX2/gtex.cgi [Accessed 2011].

Harbison CT, Gordon DB, Lee TI, Rinaldi JN, MacIssac KD, Danford TW, Hannett NM, Tagne JB, Reynolds B, Yoo J et al. 2004. Transcriptional regulatory code of a eukaryotic genome. *Nature* **431**: 99-104.

Heer J, Boyd D. 2005. Vizster: Visualizing Online Social Networks. *IEEE Symposium on Information Visualization* **InfoVis**: 5.

Heer J, Perer A. 2011. Orion: A System for Modeling, Transformation and Visualization of Multidimensional Heterogeneous Networks. *IEEE Conference on Visual Analytics Science and Technology* **6**: 49-58.

Hindorff LA, MacArthur J, (European Bioinformatics Institute), Wise A, Junkins HA, Hall PN, Klemm AK, Manolio TA, 2011. *A Catalog of Published Genome-Wide Associations Studies*. [Online] Available at: www.genome.gov/gwastudies.

Hindorff LA, Sethupathy P, Junkins HA, Ramos EM, Mehta JP, Collins FS, Manolio TA. 2008. Potential etiologic and functional implications of genome-wide association for human diseases and traits. *PNAS* **106**(23): 9362-9367.

Holm K, Melum E, Franke A, Karlsen TH. 2010. SNPexp - A web tool for calculating and visualizing correlation between HapMap genotypes and gene expression levels. *BMC Bioinformatics* **11**(600): doi:10.1186/1471-2105-11-600.

Hsu Y, Zillikens MC, Wilson SG, Farber CR, Demissie S, Soranzo N, Bianchi EN, Grundberg E, Liang L, Richards JB et al. 2010. An integration of genome-wdie association study and gene expression profiling to prioritize the discovery of novel susceptibility loci for osteoporosis-related traits. *PLoS Genetics* **6**(6): e1000977.

Huang GJ, Shifman S, Valdar W, Johannesson M, Yalcin B, Taylor MS, Taylor JM, Mott R, Flint J. 2009. High resolution mapping of expression QTLs in heterogeneous stock mice in multiple tissues. *Genome Research* **19**: 1133-1140.

Hughes JD, Estep PW, Tavazoie S, Church GM. 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *J Mol Biol* **296**(5): 1205-1214.

Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, Dai H, He YD et al. 2000. Functional discovery via a compendium of expression profiles. *Cell* **102**(1): 109-26.

Hu Z, Hung JH, Wang Y, Change YC, Hugan CL, Huyck M, Delisi C. 2009. VisANT 3.5: multi-scale network visualization, analysis, and inference based on the gene ontology. *Nucleic Acids Res* **37**: W115-121.

Hu Z, Ng DM, Yamada T, Chen C, Kawashima S, Mellor J, Linghu B, Kanehisa M, Stuart JM, C DeLisi. 2007. VisANT 3.0: new modules for pathway visualization, editing, prediction, and construction. *Nucl. Acids Res* **35**(suppl 2): W625-W632.

Huttenhower C, Mehmood SO, Troyanskaya OG. 2009. Graphle: Interactive exploration of large, dense graphs. *BMC Bioinformatics* **10**: 417.

Itoh M, Nelson C, Myers C, Bissell M. 2007. Rap1 integrates tissue polarity, lumen formation, and tumorigenic potential in human breast epithelial cells. *Cancer Research* **67**(10): 4759.

Jayawardena M, Toor S, Holmgren S, 2010. *Computational and visualization tools for genetic analysis of complex traits*. Uppsala University.

Jiao Y, Tausta SL, Gandotra N, Sun N, Liu T, Clay NK, Ceserani T, Chen M, Ma Ligeng, Holford M et al. 2009. A transcriptome atlas of rice cell types uncovers cellular, functional and developmental hierarchies. *Nat Genet* **41**: 258-263.

Johannesson M, R Lopez-Aumatell, Stridh P, Diez M, Tuncel J, Blazquez G, Martinez-Membrives E, Canete T, Vicens-Costa E, Graham D et al. 2009. A resource for the simultaneous high-resolution mapping of multiple quantitative trait loci in rats: the NIH heterogeneous stock. *Genome Res* **19**(1): 150-8.

Jubb AM, Pham TQ, Hanby AM, Frantz GD, Peale FV, Wu TD, Koeppen HW, Hillan KJ. 2004. Expression of vascular endothelial growth factor, hypoxia inducible factor 1α, and carbonic anhydrase IX in human tumours. *J. Clin. Pathol* **57**: 504.

Jung E, Moon H, Park S, Cho B, Lee S, Jeong C, Ju Y, Jeong S, Lee Y, Choi S et al. 2010. Decreased annexin A3 expression correlates with tumor progression in papillary thyroid cancer. *Proteomics* **4**: 528-527.

Kamada T, Kawai S. 1989. An algorithm for drawing general indirect graphs. *Information Processing Letters* **31**(1): 7-15.

Keim DA, Mansmann F, Schneidewind J, Thomas J, Ziegler H. 2008. Visual Analytics: Scope and Challenges. In *Visual Data Mining*. (ed.) pp.10.1007/978-3-540-71080-6 6. Springer-Verlag Berlin, Heidelberg.

Keller MP, Choi YJ, Wang P, Davis DB, Rabaglia ME, Oler AT, Stapleton DS, Argmann C, Schueler KL, Edwards S et al. 2008. A gene expression network model of type 2 diabetes links cell cycle regulation in islets with diabetes susceptibility. *Genome Res* **18**: 706-716.

Kim JM, Vanguri S, Boeke JD, Gabriel A, Voytas DF. 1998. Transposable elements and genome organization: a comprehensive survery of retrotransposons revealed by the complete Sacchararomyces cerevisiae genome sequence. *Genome Res* **8**(5): 464-478.

Kim S, Xing EP. 2009. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genetics* **5**(8): e1000587.

Kim S, Xing EP. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*.

Krastanova O, Hadzhitodorov M, Pesheva M. 2005. Ty Elements of the yeast Saccharomyces cerevisiae. *Biotchnol Biotc Eq* **19**(3): 19-26.

Kulkarni I, Mistry SY, Cummings B, Gopi M. 2011. A Visual Navigation System for Querying Neural Stem Cell Imaging Data. *IEEE Conference on Visual Analytics Science and Technology* **6**: 209-218.

Kumanovics A, Takada T, Lindahl KF. 2002. Genomic Organization of the Mammalian MHC. *Annual Review of Immunology* **21**: 629-657.

Kump KL, Bradbury PJ, Wisser RJ, Buckler ES, Belcher AR, Oropeza-Rosas MA, Zwonitzer JC, Kresovich S, McMullen MD, Ware D et al. 2011. Genome-wide association study of quantitative resistance to southern leaf blight in the maize nested association mapping population. *Nature genetics* **43**: 163-168.

Kwon BC, Fisher B, Yi JS. 2011. Visual Analytic Roadblocks for Novice Investigators. *IEEE Conference on Visual Analytics Science and Technology* **6**: 1-9.

Langsenlehner U, Yazdani-Biuki B, Eder T, Renner W, Wascher TC, Paulweber B, Weitzer W, Samonigg H, Krippl P. 2006. The cyclooxygenase-2 (PTGS2) 8473T>C polymorphism is associated with breast cancer risk. *Clin. Cancer Res.* **12**: 1392.

Lee SI, Dudley AM, Drubin D, Silver PA, Krogan NJ, Pe'er D, Koller D. 2009. Learning a prior on regulatory potential from eQTL data. *PLoS Genet* **5**(1): e1000358.

Lee SI, Pe'er D, Dudley AM, Church GM, Koller D. 2006. Identifying regulator mechanisms using individual variation reveals key role for chromatin modification. *PNAS* **103**(38): 14062-7.

Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, Hannet NM, Harbison CT, Thompson CM, Simon I et al. 2002. Transcriptional Regulatory Networks in Saccharomyces cerevisiae. *Science* **298**(5594): 799-804.

Lee S, Zhu J, Xing EP. 2010. Adaptive Multi-Task Lasso: with Application to eQTL Detection. In *Advances in Neural Information Processing Systems 23 (NIPS)*.

Lettre G, Rioux JD. 2008. Autoimmune diseases: insights from genome-wide association studies. *Hum. Mol. Genet.* **17**(R2): R116-R121.

Li Z, Chan C. 2004. Inferring pathways and networks with a Bayesian framework. *The FASEB Journal* **18**(6): 746-748.

Liu H, Radisky D, Wang F, Bissell M. 2004. Polarity and proliferation are controlled by distinct signaling pathways downstream of PI3-kinase in breast epithelial tumor cells. *Journal of Cell Biology* **164**(4): 603.

Liu Z, Stasko J. 2010. Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective. *IEEE Transactions on Visualization and Computer Graphics* **16**(6): 999-1008.

Lozano AC, Abe N, Liu Y, Rossert S. 2009. Grouped graphical Granger modeling for gene expression regulatory network discovery. *Bioinformatics* **25**(12): i110-i118.

Luscombe NM, Babu MM, Yu H, Snyder M, Teichmann SA, Gerstein M. 2004. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* **431**: 308-312.

MacIssac KD, Wang T, Gordon DB, Gifford DK, Stormo GD, E Fraenkel. 2006. An improved map of conserved regulatory sites for Saccharomyces cerevisiae. *BMC Bioinformatics* **7**: 113.

Madahain JO, Fisher D, Smyth P, White S, Boey YB. 2005. Analysis and Visualization of Network Data using JUNG. **VV**(II).

Maere S, Heymans K, Kuiper M. 2005. BiNGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in biological networks. *Bioinformatics* **21**: 3448-3449.

Malik A, Maciejewski R, Maule B, Ebert DS. 2011. A Visual Analytics Process for Maritime Resource Allocation and Risk Assessment. *IEEE Conference on Visual Analytics Science and Technology* **6**: 219-228.

Manolio RA, Collins FS, Cox NJ, Goldstein DB, Hindorff LA, Hunter DJ, McCarthy MI, Ramos EM, Cardon LR, Chakravarti A et al. 2009. Finding the missing heritability of complex disease. *Nature* **461**: 747-753.

Marcinkiewicz M, Benijannet S, Falgueyret JP, Seidah NG, Schurch W, Verdy M, Cantin M, Chretien M. 1988. Identification and localization of 7B2 protein in human, porcine, and rat thyroid gland and in human medullary carcinoma. *Endocrinology* **123**(2): 866-73.

Martin O, Valsesia A, Telenti A, Xenarios I, Stevenson BJ. 2009. AssociationViewer: a scalable and integrated software tool for visualization of large-scale variation data in genomic context. *Bioinformatics* **25**(5): 662-663.

Martin A, Ward M. 1995. High Dimensional Brushing for Interactive Exploration of Multivariate Data. In *Proceedings of IEEE Visualization*.

McCarthy MI, Hirschorn JN. 2008. Genome-wide association studies: potential next steps on a genetic journey. *Hum. Mol. Genet.* **17**(R2): R156-R165.

McPherson R, Pertsemlidis A, Kavaslar N, Stewart A, Roberts R, Cox DR, Hinds DA, Pennacchio LA, Tybjaerg-Hansen A, Folsom AR et al. 2007. A common allele on chromsome 9 associated with coronary heart disease. *Science* **316**(5830): 1488-91.

Meyer M, Munzner T, DePace A, Pfister H. 2010. MulteeSum: A Tool for Comparative Spatial and Temporal Gene Expression Data. *IEEE Transactions on Visualization and Computer Graphics* **16**(6): 908-917.

Monroe D, Leitzel A, Klein H, Matson S. 2005. Biochemical and genetic characterization of Hmilp, a yeast DNA helicase involved in the maintenance of mitochondrial DNA. *Yeast* **22**(16): 1269-1286.

Montgomery SB, Dermitzakis ET. 2009. The resolution of the genetics of gene expression. *Human Molecular Genetics* **18**(2): R211-R215.

Moore JH, Asselbergs FW, Williams SM. 2010. Bioinformatics challenges for genome-wide association studies. *Bioinformatics* **26**(4): 445-455.

Mueller M, Goel A, Thimma M, Dickens NJ, Aitman TJ, Mangion J. 2005. eQTL Explorer: integrated mining of combined genetic linkage and expression experiments. *Bioinformatics* **22**(4): 509-511.

Narayanan K, Li J. 2010. MAVEN: a tool for visualization and functional analysis of genome-wide association results. *Bioinformatics* **26**(2): 270-272.

NCBI, 2011. *NCBI*. [Online] Available at: http://blast.ncbi.nlm.nih.gov/Blast.cgi.

Nicolae DL, Gamazon E, Zhang W, Duan S, Dolan ME, Cox NJ. 2010. Trait-Associated SNPs are More Likely to Be eQTLs: Annotation to Enhance Discovery from GWAS. *PLoS Genetics* **6**(4): e1000888.

Nielsen CB, Jackman SD, Birol I, Jones SJ M. 2009. ABySS-Explorer: Visualizing genome sequence assemblies. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2009)* **15**(6): 881-8.

North C, Shneiderman B. 2000. Snap-Together Visualization: Can Users Construct and Operate Coordinated Views. *Intl. Journal of Human-Computer Studies, Academic Press* **53**(5): 715-739.

Parikh AP, Wu W, Curtis RE, Xing EP. 2011. TREEGL: reverse engineering tree-evolving gene networks underlying developing biological lineages. *Bioinformatics* **27**(13): i196-i204.

Pavlopoulos GA, Wegener AL, Schneider R. 2008. A survey of visualization tools for biological network analysis. *BioData Mining* **1**: 12.

Pendergrass SA, Dudek SM, Crawford DC, Ritchie MD. 2010. Synthesis-View: visualization and interpretation of SNP association results fro multi-cohort, multi-phenotype data and meta-analysis. *BioData Mining* **3**(10): doi:10.1186/1756-0381-3-10.

Petersen O, Ronnov-Jessen L, Howlett A, Bissell M. 1992. Interaction with basement membrane serves to rapidly distinguish growth and differentiation pattern of normal and malignant human breast epithelial cells. *PNAS* **89**(19): 9064.

Pettersson F, Morris AP, Barnes MR, Cardon LR. 2008. Goldsurfer2 (Gs2): A comprehensive tool for the analysis and visualization of genome wide association studies. *BMC Bioinformatics* **9**(138): doi:10.1186/1471-2105-9-138.

Postma DS, Koppelman GH. 2009. Genetics of Asthma: Where are we and where do we go? *Proceedings of the American Thoracic Society* **6**: 283-287.

Pramila T, Wu W, Miles S, Noble WS, Breeden LL. 2006. The forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. *Gene & Development* **20**(16): 2266-78.

Pritchard JK, Stephens M, Donnelly P. 2000. Inference of Population Structure Using Multilocus Genotype Data. *Genetics* **155**: 945-959.

Pruim RJ, Welch RP, Sanna S, Teslovich TM, Chines PS, Gliedt TP, Boehnkie M, Abecasis GR, Willer CJ. 2010. LocusZoom: regional visualization of genome-wide association scan results. *Bioinformatics* **26**(18): 2336-2337.

Puniyani K, Kim S, Xing EP. 2010. Multi-population GWA mapping via multi-taks regularized regression. *Bioinformatics* **26**(12): i208-i216.

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA R, Bender D, Maller J, Skalr P,de Bakker, P I W, Daly MF, Sham PC. 2007. PLINK: a toolset for whole-gehome association and population-based linkage analysis. *American Journal of Human Genetics* 81.

R Development Core Team, 2011. *R: A Language and Environment for Statistical Computing*. [Online] Available at: http://www.R-project.org.

Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y. 2008. SimpleMKL. *Journal of Machine Learning Research* **9**: 2491-2521.

Reynolds PR, Allison CH, Willnauer CP. 2010. TTF-1 regulates α5 nicotinic acetylcholine receptor (nAChR) subunits in proximal and distal lung epithelium. *Respiratory Research* **11**(175): doi: 10.1186/1465-9921-11-175.

Robinson JW, Hartemink AJ. 2010. Non-stationary dynamic Bayesian networks: Learning Non-Stationary Dynamic Bayesian Networks. *Journal of Machine Learning Research* **11**: 3647-3680.

Roebroek AJ M, Martens GJ M, Dults AJ, Schalken JA, van Bokhoven A, Wagenaar SS, M Vande Ven W J. 1989. Differential expression of the gene encoding the novel pituitary polypeptide 7B2 in human lung cancer cells. *Cancer Research* **49**: 4154-4158.

Rosenbloom KR, Dreszer TR, Pheasant M, Barber GP, Meyer LR, Pohl A, Raney BJ, Wang T, Hinrich AS, Zweig AS et al. 2009. ENCODE whole-genome data in the UCSC Genome Browser. *Nucleic Acids Research* **38**: 620-625.

Sahai E, Marshall C. 2002. RHO-GTPases and cancer. *Nat Rev Cancer* **2**: 133-142.

Schadt EE. 2009. Molecular networks as sensors and drivers of common human diseases. *Nature* **461**: 218-223.

Schadt EE, Molony C, Chudin E, Hao K, Yang X, Lum PY, Kasarskis A, Zhang B, Wang S, Suver C et al. 2008. Mapping the genetic architecture of gene expression in human liver. *PLoS Biol* **6**(5): e107.

Schafer J, Strimmer K. 2005. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics* **21**: 754-764.

Sedman T, Kuusk S, Kivi S, Sedman J. 2000. Mitochondrial Genome in Saccharomyces cerevisiae. *Mol Cell Bio* **20**(5): 1816-1824.

Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N. 2003. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet* **34**: 166-176.

SGD, 2011. *Saccharomyces Genome Database*. [Online] Available at: http://yeastgenome.org [Accessed 2011].

Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research* **13**(11): 2498-504.

Shen L, Kim S, Risacher SL, Nho K, Swaminathan S, West JD, Foroud T, Pankratz N, Moore JH, Sloan CD et al. 2010. Whole genome association study of brain-wide imaging phenotypes for identifying quantitative trait loci in MCI and AD: A study of the ADNI cohort. *NeuroImage* **53**(3): 1051-1063.

Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K. 2001. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.* **29**(1): 308-11.

Shneiderman B. 1994. Dynamic Queries for Visual Information Seeking. *IEEE Software* **11**(6): 70-77.

Shneiderman B. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc 1996 IEEE Visual Languages*. Boulder, CO.

Shore D. 1997. Telomere length regulation: getting the measure of chromosome ends. *Biol Chem* **387**(7): 591-7.

Sickmann A, Reinders J, Wagner Y, Joppich C, Zahedi R, Meyer HE, Schonfisch B, Perschil I, Chacinska A, Guiard B et al. 2003. The proteome of Saccharomyces cerevisiae mitochondria. *PNAS* **100**(23): 13207-12.

Silveira AC, Morrison MA, Ji F, Xu H, Reinecke JB, Adams SM, Arneberg TM, Janssian M, Lee J, Yuan Y et al. 2010. Convergence of linkage, gene expression and association data demonstrates the influence of the RAR-related orphan receptor alpha (RORA) gene on neovascular AMD: A systems biology based approach. *Vision Research* **50**(7): 698-715.

Smith MA, Shneiderman B, Milic-Frayling N, Rodrigues EM, Barash V, Dunne C, Capone T, Perer A, Gleave E. 2009. Analyzing (social media) networks with NodeXL. In *Proceedings of the fourth international conference on Communities and technologies*.

Song L, Kolar M, Xing EP. 2009. KELLER: estimating time-varying interactions between genes. *Bioinformatics* **12**(i128): 25.

Song L, Kolar M, Xing EP. 2009. Time-Varying Dynamic Bayesian Networks. *Proceedings of the 23rd Neural Information Processing Systems*.

Stegle O, Parts L, Durbin R, Winn J. 2010. A Bayesian Framework to Account for Complex Non-Genetic Factors in Gene Expression Levels Greatly Increases Power in eQTL Studies. *PLoS Comp Bio* **6**(5): e1000770.

Straatman KR, Louis EJ. 2007. Localization of telomeres and telomere-association proteins in telomerase-negative Saccharomyces cerevisiae. *Chromosome Res* **15**: 1033-1050.

Stranger BE, Nica AC, Forrest MS, Dimas A, Bird CP, Beazley C, Ingle CE, Dunning M, Flicek P, Koller D et al. 2007. Population genomics of human gene expression. *Nat Genet* **39**: 1217-1224.

Stuart PE, Nair RP, Ellinghuas E, Ding J, Tejasvi T, Gudjonsson JE, Li Y, Weidinger S, Eberlein B, Gieger C et al. 2010. Genome-wide association analysis identifies three psoriasis susceptibility loci. *Nature genetics* **42**: 1000-1004.

Suderman M, Hallett M. 2007. Tools for visually exploring biological networks. *Bioinformatics* **23**(20): 2651-2659.

Sydorskyy Y, Dilworth DJ, Yi EC, Goodlett DR, Wozniak RW, Aitchison JD. 2003. Intersection of the Kap123p-mediated nuclear import and ribosome export pathways. *Mol Cell Biol* **23**(6): 2042-2054.

Teixeira MT, Arneric M, Sperisen P, Lingner J. 2004. Telomere length homeostasis is achieved via a switch between telomerase- extendable and -nonextendable states. *Cell* **117**(3): 323-325.

Thain D, Tannenbaum T, Livny M. 2005. Distributed computing in practice: the Condor experience. *Concurrency - Practice and Experience* **17**(2-4): 323-356.

The GoDARTS and UKPDS Diabetes Pharmacogenetics Study Group & The Wellcome Trust Case Control Consortium 2. 2010. Common variants near ATM are associated with glycemic response to metformin in type 2 diabetes. *Nature genetics* **43**: 117-120.

The UniProt Consortium. 2011. Ongoing and future developments at the Universal Protein Resource. *Nucleic Acids Res.* **39**: D214-D219.

Tian F, Bradbury PJ, Brown PJ, Hung H, Sun Q, Flint-Garcia S, Rocheford TR, McMullen MD, Holland JB, Buckler ES. 2011. Genome-wide association study of leaf architecture in the maize nested association mapping population. *Nature genetics* **43**: 159-162.

Tibshirani R. 1996. Regression shrinkage and selection via the lasso. *Royal Statist Soc B* **58**(1): 267-288.

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB. 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**(6): 520-525.

Turner S, Bush W., 2009. *Visualizing sample relatedness in a GWAS using PLINK and R*. [Online] Available at: http://gettinggeneticsdone.blogspot.com/2009/10/visualizing-sample-relatedness-in-gwas.html [Accessed 2011].

Valdar W, Solberg LC, Gauguler D, Burnett S, Klenerman P, Cookson WO, Taylor MS, Rawlins JN, Mott R, Flint J. 2006. Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nat Genet* **172**: 1783-1797.

van Ham F, Wattenberg M, Viegas FB. 2009. Mapping Text with Phrase Nets. *IEEE Transactions on Visualization and Computer Graphics* **15**(6): 1169-1176.

Waring SC, Rosenberg RN. 2008. Genome-Wide Association Studies in Alzheimer Disease. *Arch Neurol* **65**(3): 329-334.

Wasserman L, Roeder K. 2009. High-dimensional variable selection. *Ann Stat* **37**(5A): 2178-2201.

Wattenberg M, Kriss J. 2006. Designing for Social Data Analysis. *IEEE Transcations on Visualization and Computer Graphics* **12**(4): 549-557.

Weaver V, Petersen O, Wang F, Larabell C, Briand P, Damsky C, Bissell M. 1997. Reversion of the malignant phenotype of human breast cells in three dimensional culture and in vivo by integrin blocking antibodies. *Journal of Cell Biology* **137**(1): 231.

Weaver AM, Silva CM. 2007. Signal tranducer and activator of transcription 5b: a new target of breast tumor knase/protein tyrosine kinase 6. *Breast Cancer Res* **9**: R79.

West MA L, Kim K, Kliebenstein DJ, van Leeuwen H, Michelmore RW, Doerge RW, St. Clair DA. 2007. Global eQTL mapping reveals the complex genetic architecture of transcript-level variation in Arabidopsis. *Genetics* **175**(3): 1441-1450.

Wu T T, Chen Y F, Hastie T, Sobel E, Lange K. 2009. Genome-wide association analysis by lasso penalized regression. *Bioinformatics* **25**(6): 714-721.

Wu J, Devlin B, Ringquist S, Trucco M, Roeder K. 2010. Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genetic Epidemiology* **34**(3): 275-285.

Xing EP, Kim S. 2011. Modern Statistical Methods for Genetic Association Study: Structured Genome-Transcriptome-Phenome Association Analysis. *Tutorial at the Nineteenth International Conference on Intelligence Systems for Molecular Biology (ISMB 2011)*.

Yaguchi H, Togawa K, Moritani M, Itakura M. 2005. Identification of candidate genes in the type 2 diabetes modifier locus using expression QTL. *Genomics* **85**(5): 591-599.

Yamada M, Hayatsu N, Matsuura A, Ishikawa F. 1998. Y'-Help1, a DNA helicase encoded by the yeast subtelomeric Y' element is induced in survivors defective for telomerase. *J Biol Chem* **273**(50): 33360-33366.

Yamazaki D, Kurisu S, Takenawa T. 2005. Regulation of cancer cell motility through actin reorganization. *Cancer Sci.* **96**(7): 379-86.

Yang TP, Beazley C, Montegomery SB, Dimas AS, Gutierrez-Arcelus M, Stranger BE, Deloukas P, Dermitzakis ET. 2010. Genevar: a database and Java application for the analysis and visualization of SNP-gene associations in eQTL studies. *Bioinformatics* **26**(19): 2474-2476.

Yeager M, Chatterjee N, Ciampa J, Jacobs KB, Gonzalez-Bosquet JG, Hayes RB, Kraft P, Wacholder S, Orr N, Berndt S et al. 2009. Identification of a new prostate cancer susceptibility locus on chromosome 8q24. *Nature Genetics* **41**: 1055-1057.

Yvert G, Brem RB, Whittle J, Akey JM, Foss E, Smith EN, Mackelprang R, Kruglyak L. 2003. Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nat Genet* **35**: 57-64.

Zamar D, Tripp B, Ellis G, Daley D. 2009. Path: a tool to facilitate pathway-based genetic association analysis. *Bioinformatics* **25**(18): 2444-6.

Zhang B, Horvath S. 2005. A General Framework for Weighted Gene Co-Expression Network Analysis. *Statistical Applications in Genetics and Molecular Biology 4(1)*.

Zhao H, Zhang X. 2010. An Evaluation of Gene Module Concepts in the Interpretation of Gene Expression Data. *Fronteirs in Computational and Systems Biology* **15**: 331-349.

Zhu C, Byers KJ, McCord RP, Shi Z, Berger MF, Newburger DE, Saulrieta K, Smith Z, Shah MV, Radhakrishnan M et al. 2009. High-resolution DNA-binding specificity analysis of yeast transcription factors. *Genome Res* **19**(4): 556-66.

Zhu J, Zhang B, Smith EN, Drees B, Brem RB, Kruglyak L, Bumgarner RE, Schadt EE. 2008. Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. **40**(7): 854-861.

Zou W, Aylor DL, Zeng ZB. 2007. eQTL Viewer: visualizing how sequence variation affects genome-wide transcription. *BMC Bioinformatics* **8**(7): doi:10.1186/1471-2105-8-7.