2010

Methods for integrated biochemical pathway analysis

John Louis Van Hemert
Iowa State University

Methods for integrated biochemical pathway analysis

by

John L. Van Hemert

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Interdepartmental Major: Bioinformatics and Computational Biology

Program of Study Committee:
Julie A. Dickerson, Co-major Professor
Basil Nikolau, Co-major Professor
Roger P. Wise
Gustavo MacIntosh
Peng Liu

Iowa State University

Ames, Iowa

2010

DEDICATION

I would like to dedicate this thesis to my wife Jessica, without whose support I would not have been able to complete this work. She has shown great perseverance and love by sacrificing much to allow me to pursue this degree. It is fitting, then, that this work be dedicated to the one who has sacrificed so selflessly over several years. I would also like to thank my friends and family for their loving support during the writing of this work. Specifically my parents, Gary and Mary Beth Van Hemert, whose roles entailed constant encouragement.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

ACKNOWLEDGEMENTS x

ABSTRACT xi

1. INTRODUCTION 1

2. MONTE CARLO RANDOMIZATION TESTS FOR LARGE-SCALE ABUNDANCE DATASETS ON THE GPU 5

Abstract 5

Introduction 6

Strengths 8

Weaknesses 8

Computational methods and theory 8

Data representation 8

Data manipulation 9

Output 10

Test statistic functions 12

Samples of typical program runs 12

Specifications 14

Hardware requirements 14

Software requirements 14

3. PATHWAYACCESS: CELLDESIGNER PLUGINS FOR PATHWAY DATABASES 20
Abstract . . . 20
Summary: . . . 20
Availability: . . . 20
Introduction . . . 21
Functionality . . . 21
Pathway Integration Across Databases . . . 22
Creating New PathwayAccess Plugins . . . 23
Examples . . . 23
Impact . . . 24
Discussion of Technical Solutions . . . 24
Importing and integrating pathways . . . 25
Caching database queries . . . 27
Committing a model to a database . . . 28
Pseudocode . . . 29
4. DISCRIMINATING OMICS RESPONSE GROUPS IN BIOCHEMICAL PATHWAY NETWORKS . . . 37
Abstract . . . 37
Introduction . . . 37
Background Terminology . . . 39
Existing uses of flow simulation . . . 41
Other graph theoretical metrics and properties . . . 42
Modeling directed random walks . . . 42
Summarizing random walk flow between groups of nodes . . . 43
Modeling and testing values in Θ, $\Theta^{(rev)}$, and $\Theta^{(tot)}$ . . . 45
Applications Using Our Web Tool . . . 50
Use Case: LipidA inhibition in *E. coli* . . . 50
Discussion . . . 60

5. EXPRESSION PLATFORM INTEGRATION AND INSIGHTS INTO THE GRAPEVINE'S RESPONSE TO SHORT WINTER DAYS 63

Abstract 63

Introduction 64

Results 67

Platform Integration for Exon Quantification 67

Single-Platform Analysis of the Full Photoperiod-induced Bud Dormancy Data 68

Discussion 75

Platform Integration 75

Dormancy-related Genes 75

Methods 81

Platform Integration Technical Study 82

Single-Platform Analysis of the Full Photoperiod-induced Bud Dormancy Data 85

APPENDIX A. GRAPEVINE DNA SEQUENCING PROJECTS 97

APPENDIX B. ANNOTATION DATABASES FOR VITIS 114

APPENDIX C. LARGE TABLES AND FIGURES 121

LIST OF TABLES

Table 4.1 Interpretation of different flow simulations and tests. 60

Table 5.1 Transcription Factor Families 68

Table 5.2 Existing data points for the two microarray platforms. 82

Table C.1 Icons and buttons on the web tool. 122

Table C.2 Significant up-regulated *E. coli* reactions. 123

Table C.3 Significant down-regulated *E. coli* reactions. 125

Table C.4 Overrepresented Annotation: Cluster 1 128

Table C.5 Cluster 1 Pathways. 129

Table C.6 Overrepresented Annotation: Cluster 3 130

Table C.7 Cluster 3 Pathways. 132

Table C.8 Overrepresented Annotation: Cluster 5 133

Table C.9 Cluster 5 Pathways. 134

Table C.10 Cluster 6 Pathways. 135

Table C.11 Overrepresented Annotation: Cluster 7 136

Table C.12 Cluster 7 Pathways. 138

Table C.13 Overrepresented Annotation: Cluster 8 139

Table C.14 Cluster 8 Pathways. 140

Table C.15 Significant pathways from Pinot Noir-Corvina Indels 148

LIST OF FIGURES

Figure 1.1 Research Roadmap 2

Figure 2.1 Permutation test schematic 15

Figure 2.2 GPU memory allocation 16

Figure 2.3 GPU verification on CPU 17

Figure 2.4 Speedup x number of columns 18

Figure 2.5 Speedup x number of variables 18

Figure 2.6 Speedup x number of permutations 19

Figure 3.1 Dataflow for PathwayAccess plugins 22

Figure 4.1 General assessment of M_{ij} fit to Exponential 46

Figure 4.2 Erlang assessments 48

Figure 4.3 Our PathwayFlow webtool 51

Figure 4.4 CHIR-090 binds LpxC 52

Figure 4.5 Flow plots for up-regulated *E. coli* pathways 53

Figure 4.6 Flow plots for up-regulated *E. coli* reactions 54

Figure 4.7 Venn diagram for up-regulated *E. coli* pathways 55

Figure 4.8 The integrated pathways containing significant successor reactions 56

Figure 4.9 The saccharolipid Kdo2-Lipid A 57

Figure 4.10 The integrated pathways containing significant predecessor reactions 58

Figure 4.11 Flow plots for down-regulated *E. coli* pathways 59

Figure 4.12 A model for *E. coli* responses to lipid A inhibition 61

Figure 5.1 Filtering misaligned Affymetrix probes 69

Figure 5.2 Integrated Exon Estimability 70

Figure 5.3 Exon estimates compare well with RNAseq data 71

Figure 5.4 Some expression profiles 72

Figure 5.5 Centroids of gene expression profiles over the seven time points for the 770 gene data set 73

Figure 5.6 Transcription Factor Homologues 74

Figure 5.7 The Swappable Prediction Model 84

Figure 5.8 Estimating the Δ sampling distribution 92

Figure 5.9 Δ-based p- and q-values 93

Figure A.1 Cross-assembly mapping is a bipartite graph problem 101

Figure A.2 Distribution of Cross-platform MegaBlast hit scores. 103

Figure A.3 Cross-platform MegaBlast hit positions 104

Figure A.4 Cross-platform MegaBlast hit positions 3D 105

Figure A.5 Opposite-strand inversion alignment odds ratio 106

Figure A.6 Full chromosome MegaBlast alignments 107

Figure A.7 Full chromosome MegaBlast alignments in GBrowse 108

Figure A.8 Corvina genes detected in Fasoli et al atlas 110

Figure A.9 GBrowse tracks for Corvina indels 112

Figure A.10 PathwayFlow plots for Corvina indels 113

Figure B.1 A VitisMart use case. 116

Figure B.2 VitisMart is integrated with our GBrowse server. 117

Figure B.3 VitisCyc summary statistics. 118

Figure B.4 Chromosome 1 in VitiCyc. 119

Figure B.5 The glycolipid biosynthesis pathway in VitiCyc. 120

Figure C.1 Automatic ID lookup service 121

Figure C.2 Plots for selecting perturbed probes from the data. 127

Figure C.3 Probes linked by common exons 141

Figure C.4 Cross-platform probe correlations 142

Figure C.5 Normalizing for Integration 143

Figure C.6 Probe-Exon Systems 144

Figure C.7 Year effect distributions 145

Figure C.8 Control probes fluorescence 146

Figure C.9 Assessing multivariate Normality 146

Figure C.10 Selecting a minimum variance cutoff 147

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Julie A. Dickerson for her guidance, patience and support throughout this research and the writing of this thesis. She is an exemplary advisor who always know exactly how to guide her students to success. I would also like to thank my co-advisor, Dr. Basil Nikolau, whose expertise and insights are invaluable. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Roger P. Wise, Dr. Gustavo MacIntosh, and Dr. Peng Liu. Without their feedback, this work would not be possible. I would additionally like to thank collabroators Anne Y. Fennell, Grant Cramer, and Mario Pezzotti. Their guidance and support enabled me to learn much of the necessary methods for this work. Lastly, but not least, I would like to thank Trish Stauble for her advice and coordination during my time as a student in this interdepartmental program.

This work was funded by three fellowships: Graduate Assistance in Areas of National Need (US Department of Education), Grape Research Collaboration Netowrk Student Travel Stipend (National Science Foundation (NSF) grant DBI 0741876), and James Cornette Fellowship (the Iowa State University Bioinformatics and Computational Biology program). Much of the work was also funded by two other NSF grants: DBI 0604755 and EEC 0813570.

ABSTRACT

The common goal for biological research is to develop models for the biological processes we seek to understand. Such models, in the form of biochemical pathway networks which describe the physical interactions between a living cell's genes, transcripts, proteins, and metabolites (”Omics”), accumulate in different repositories for several model organisms as well as non-model organisms. This thesis presents a set of integrated statistical bioinformatics tools that address key problems in integrating large-scale Omics datasets with pathway network models. A hardware accelerated non-parametric Omics mining method (Monte Carlo on the GPU) allows faster screening of custom test statistics and functions. A software platform for mining pathway databases (PathwayAccess) confers knowledge integration and comparison. Omics and pathway mining are combined for a novel method for statistically discriminating functionally meaningful subnetworks for their interaction with lists of entities mined from Omics data, so that software can intelligently mine large and complex pathway databases to answer a wide variety of questions and generate hypotheses (Discriminating Omics Response Groups in Pathways). The method, called PathwayFlow, can discriminate pathways, reactions, metabolite classes, or any other biological entity grouping (Response Groups), and automatically accounts for connectivity-caused biases in the pathway network. It also differentiates between regulators (or inputs) and regulatees (or outputs) for a given Query List of Omics entities. It is applied to three real datasets: a simple *E. coli* gene expression dataset which validates the method, a more complex *Vitis* gene expression dataset which complements functional enrichment analysis (Grapevine's Response to Short Days), and an ultra-high throughput re-sequencing dataset for assessing genetic differences between two wine grape varieties (DNA Sequencing Appendix).

1. INTRODUCTION

Omics refers to the quantification of the entirety of something in a living cell. Whether it is genomics, transcriptomics, proteomics, metabolomics, or other -omics, most studies involve four general steps:

1. Design the experiment. This includes a careful consideration of the objectives of the study and plans for sampling.

2. Generate the data. This is when the experiment is run; samples are generated/collected, and biotechnologies such as sequencing or microarrays are used.

3. Mine the data. This is the first directly computational step (although design should consider statistical requirements for sampling), where data points collected from the biotechnologies are pre-processed and relavent entities (ie differentially expressed genes) are mined from the full population.

4. Interpret the results. This is usually the most intellectually creative and challenging step. Given lists mined from the data, scientists must reconcile them with existing biological knowledge as well as hypothesize new models for the biological processes that were activated or perturbed by the experimental treatments.

The purpose of this dissertation is to communicate my contribution to our ability to conduct the final two general steps: Data Mining and Results Interpretation, specifically in the metabolic pathway context. These steps are closely linked, not only because they are computational, but because data mining results directly affect interpretation and hypothesis generation; During the data mining step, the interpretation step must be considered so that the desired behaviors are mined and during the interpretation step, one must understand how the omics lists were mined from the data. Figure 1.1 illustrates the relevant research processes and how they relate to one another as well as their focus in the following chapters.

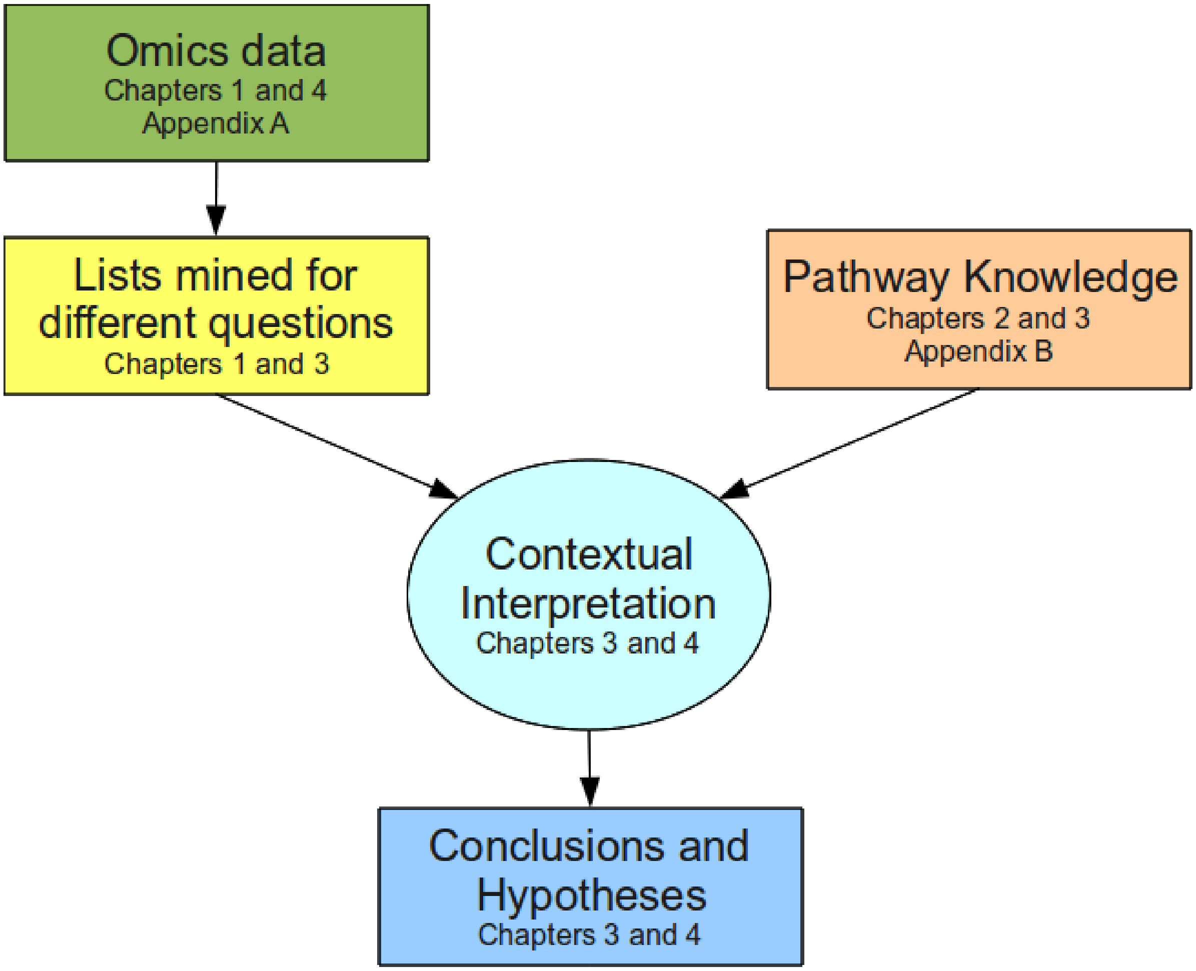


Figure 1.1 Research processes and the chapters which discuss them.

This document is organized into four chapters, each a manuscript that is either already published in a peer reviewed journal or will be submitted to one. The first chapter, Monte Carlo Randomization Tests for Large-scale Abundance Datasets on the GPU (Van Hemert and Dickerson (2010a), Chapter 2 on page 5), was published in Computational Methods and Programs in Biomedicine in June 2010 and discusses the challenge of mining lists of entities from Omics data and presents a hardware-accelerated Monte Carlo based method for doing so. It is an important component to the general work because mining lists from Omics data is directly related to interpretation and pertains to the data mining step.

The second chapter, PathwayAccess: CellDesigner Plugins for Pathway Databases (Van Hemert and Dickerson (2010b), Chapter 3 on page 20) was published in Bioinformatics in July, 2010 and presented at the 2010 PathwayTools Workshop at the Stanford Research Institute and discusses the challenges of curating and integrating different pathway network model repositories and presents a software for doing so. It is important because an understanding of the existing pathway databases is necessary for processing such data and knowledge and pertains to the interpretation step, specifically the pathway context.

The third chapter, Discriminating Omics Response Groups in Biochemical Pathway Networks (to be submitted, Chapter 4 on page 37), presents a novel method for statistically modeling and discriminating subnetworks from a pathway network using sound hypothesis tests. Inputs include a list of Omics entities (i.e. differentially expressed genes), a metabolic pathway network on which to base interpretation, and a preset definition of response groups to be discriminated; response groups can be any delineation of metabolic entities in the network such as pathways, reactions, or chemical compound classes. Current implementations of the tool only support pathway or reaction response groups. The output is a visualization of the hypothesis tests used to discriminate significant response groups along with lists of said response groups. The method is validated with a web tool use case for analyzing differentially expressed *E. coli* genes in the EcoCyc pathway network and a novel model for *E. coli* response to Lipid A deprivation is posited. This chapter pertains to the interpretation step and is the main computational contribution of this work by integrating concepts from the first two chapters.

The fourth chapter, Expression Platform Integration and Insights into the Grapevine's Response to Short Winter Days (to be submitted, Chapter 5 on page 63), focuses on grapevine data and includes two parts: First, a technical gene expression integration study with a novel method for exon-specific quantification and verified by a comparison to RNAseq data. This section is a key component to the work because understanding the relationships between different biotechnologies is necessary for processing Omics data. It also uses some of the same computational theory as the third chapter, but for modeling relationships between exons and the microarray probes that measure them instead of pathway networks. Second, a novel multivariate data mining effort to mine time-dependent activity from gene expression in the paradormant buds of *Vitis riparia* along with a functional analysis using conventional category enrichment tests and our novel pathway flow modelling method described in the third chapter. This section is also important because it applies a novel mining method to real data and then complements the flow analysis method in the third chapter with a traditional category enrichment analysis for the resulting gene lists to generate new hypotheses for the grapevine's response to shorter photoperiods. This chapter discusses all four general Omics steps and uses novel methods for the third and fourth steps.

Two appendices include content from other bioinformatic work conducted during this time. Much of the work presented in the appendices was done at the University of Verona under the supervision of Dr. Mario Pezzotti and funded by a student travel stipend awarded by the Grape Research Collaboration Network. Appendix A on page 97 discusses challenges and solutions for processing ultra-high throughput data from Next-Generation Sequencing projects. This section is highly relevant to the main document because it shows exposure and understanding of new biotechnologies and the computational methods that are being developed to process their data. Appendix B on page 114 discusses solutions for curating the results and integrating them with functional annotation, which is a non-trivial task when working with multiple collaborators at different locations and different sets of functional annotation.

2. MONTE CARLO RANDOMIZATION TESTS FOR LARGE-SCALE ABUNDANCE DATASETS ON THE GPU

A paper published in Computational Methods and Programs in Biomedicine

John L. Van Hemert and Julie A. Dickerson

Abstract

Statistical tests are often performed to discover which experimental variables are reacting to specific treatments. Time-series statistical models usually require the researcher to make assumptions with respect to the distribution of measured responses which may not hold. Randomization tests can be applied to data in order to generate null distributions non-parametrically. However, large numbers of randomizations are required for the precise p-values needed to control false discovery rates. When testing tens of thousands of variables (genes, chemical compounds, or otherwise), significant q-value cutoffs can be extremely small (on the order of 10^{-5} to 10^{-8}). This requires high-precision p-values, which in turn require large numbers of randomizations. The NVIDIA® Compute Unified Device Architecture® (CUDA®) [1] platform for General Programming on the Graphics Processing Unit (GPGPU) was used to implement an application which performs high-precision randomization tests via Monte Carlo sampling for quickly screening custom test statistics for experiments with large numbers of variables, such as microarrays, Next-Generation sequencing read counts, chromotographical signals, or other abundance measurements. The software has been shown to achieve up to more than 12 fold speedup on a Graphics Processing Unit(GPU) when compared to a

[1] NVIDIA

powerful Central Processing Unit(CPU). The main limitation is concurrent random access of shared memory on the GPU. The software is available from the authors.

Introduction

Statistical models provide a detailed analysis of the extremity of observed data and are always based on some number of assumptions. These assumptions usually consider the relationships between the test subjects and treatments as well as the nature in which a subject responds to a treatment. Randomization tests provide a non-parametric measure of the extremity of an observation which does not require these assumptions. While some methods such as Knijnenburg et al. (2009) attempt to approximate sampling distribution behavior in the tails by fitting curves to simple permutation sets, exact permutation is the only way to be sure of tail behavior for complex statistics.

Time series experiments violate the usual assumption of independence in statistical testing because samples from different time points are inherently related through time. One method which does not require independence is the generation of a null distribution by reordering data labels in all possible permutations and calculating the value of a test statistic for each permutation. The test statistic calculated from the observed abundance profile can then be tested under the null hypothesis, by comparing it to the generated null distribution. See Figure 2.1 for an illustration of the general randomization test procedure. A test statistic can be any meaningful function of the abundance profile of a variable. While a student's T statistic is an example of a simple parametric test statistic, other functions can be used with this application, such as comparing different distance metrics between treatments through time. When the number of samples over the time points is prohibitively large, a Monte Carlo simulation simply samples randomly from the population of all possible permutations in order to estimate the true null distribution. Gene expression datasets created using microarray or Next-Generation technology are prime candidates for our method because these datasets contain tens to hundreds of observation on thousands to tens of thousands of variables.

Generating very large numbers of permutations is essential for multiple testing correction.

Consider an experiment with 20,000 abundance profiles. Sampling 100 random permutations for each of the 20,000 variables may be computationally convenient. However, the significance cutoff for rejecting the null hypothesis for a variable must be corrected for multiple tests (there are actually 20,000 tests for this single experiment). Without such correction, a Type I Error Rate of 0.05 would produce $20,000 * 0.05 = 1,000$ false positives on average. The simplest method, known as Bonferroni correction (Holm, 1979), divides the desired false discovery rate by the number of tests. If the FDR is 0.05 (a common selection), then the significance cutoff would be $0.05/20,000 = 0.0000025$. Clearly, the precision of sampling only 100 permutations for an abundance profile is inadequate. Sampling 100 random permutation statistics results in p-values of 0.0, 0.01, 0.02, up to 1.0; the smallest non-zero p-value is 0.01, which is much larger than the cutoff value. These examples show that there is a granularity associated with permutation statistics which limits the sensitivity of the overall test. For our example with 20,000 abundance profiles permuted 100 times, there is a granularity of $1/100 = 0.01$, so profiles with true exact permutation p-values between 0.0000025 (the cutoff value) and 0.01 would not be considered interesting, because they would usually yield zero random statistics more extreme than their corresponding observed statistics. On average, sampling just 100 permutations would yield p-values of 0.0 and declare them as interesting, resulting in false positives. For this example, a granularity of $1/10^7 = 10^{-7}$ would be necessary to discriminate between true random statistics of 0.0000025 and 0.0000026; false Discovery Rate would not be correctly controlled with less than 10^7 permutations for each variable. Alarmingly, under a null hypothesis with a uniform distribution of p-values, an experiment with 20,000 variables would be expected to produce $(0.01 - .0000025) * 20,000 = 200$ unintended significant p-values.

Permuting so many abundance profiles so many times each can be computationally prohibitive. Fortunately, the process is parallelizable on a graphics processing unit (GPU). GPUs are designed to quickly update information on pixel displays in parallel. GPUs are usually able to carry out specific hardware-tuned calculations meant for graphics display. NVIDIA's CUDA platform provides a C-like programming language and compiler enabling general programming of standard and user-defined functions on the GPU. The following sections describe the algorithm and speedup results for high-precision randomization tests on a GPU. NVIDIA's CUDA (Corporation, 2007) platform for GPGPU was used to implement an application which performs high-precision randomization tests via Monte Carlo sampling for quickly screening custom test statistics for abundance data. The software achieves up to more than 12 fold speedup on a GPU when compared to a powerful CPU.

Strengths

Conducting large numbers of randomizations for each variable enhances multiple testing correction and improves confidence in results. The massively parallel GPU can be a powerful tool for randomization tests on abundance data when the experiment is complex, parametric assumptions are unmet, and high *p*-value precision is necessary for correct FDR control.

Weaknesses

Shared GPU memory creates a speedup bottleneck for data-dependent applications. On a multi-cored computing architecture with shared memory, the processing speed of this type of application depends on the architecture's ability to allow multiple threads (cores) to access memory simultaneously. The architecture used here is CUDA compute capability 1.0, which limits the number of simultaneous coalesced memory accesses to 16 and does not offer large enough on-chip cached memory for typical abundance datasets. Future plans include redesign of the application using a more advanced architecture such as CUDA compute level 1.3 which may provide more flexibility in coalescing memory accesses between threads. Goals of the improved application would include all the those for the current one, plus providing a speedup over the compared CPU architecture significantly greater than 12X.

Computational methods and theory

Data representation

Abundance experiment data are usually stored as text files in the form of delimited tables. Rows are labeled by unique identification numbers or accessions representing different variables.

Columns are labeled by different treatments, which may include time points at which each variable's abundance level is measured under different conditions and at different time points. A cell value in the table is a real number representing abundance level for a specific variable under a specific treatment. It is important to note that there are several methods for normalizing raw abundance data, resulting in this tabular form. Such methods include MAS5.0, RMA, and GCRMA (Irizarry et al., 2003b).

Data manipulation

Data preparation

The application reads a tabular abundance data file into host memory and then copies it to device global memory, where it can be read by all threads after the randomization test kernel launches. Before the test statistic calculation begins, another kernel transposes the input data. The data must be transposed to allow for coalesced GPU global memory accesses (see Figure 2.2). See Corporation (2007) for details on memory access coalescence. The kernel used to efficiently transpose the data is provided with the CUDA toolkit as an example project [7].

Permute column indices

Column index permutation is parallelized by thread block. Each data row (abundance profile) is permuted according to permuted indices shared by each thread block. This way, random permutation need only be conducted once for each permutation requested by the user and not r times for each permutation requested where r is the number of data rows. This also allows for coalesced memory accesses when permuted data rows. The kernel parallelizes an algorithm for pseudo-random number generation similarly to the Mersenne Twister project example included in the NVIDIA CUDA Software Development Kit (SDK) (Pdlozhnyuk, 2007). The Mersenne Twister algorithm is a large-period bit vector-based method for pseudo-random number generation and is ideal for implementation on the GPU. The pseudo-random numbers generated are used to shuffle the data row indices (0 through the number of data columns less 1) using the modern implementation of the Fisher-Yates Shuffle described in (Durstenfeld, 1964).

Compare observed statistic with permutation-based statistic

With a shared index permutation now in hand, data row processing is parallelized by thread within each block. Each thread in a thread block is assigned a different data row (abundance profile). Each thread then stores its data row in permuted order according to the index permutation. It then calculates the test statistic and compares it to the same test statistic for the observed (un-permuted) data row (abundance profile), which has been calculated and stored in global GPU memory previously by a different kernel. If the randomly permuted data results in a more extreme test statistic, a count of permutations at least as extreme as the data row is incremented. When finished, this count will be divided by the number of permutations, resulting in a p-value which approaches an exact limit as the number of permutations grows. Note that the meaning of ”more extreme” depends on the meaning of the test statistic. For example, some test statistics may require a comparison of the observation's and permutation's distance from zero, rather than a simple comparison of the observation and permutation test statistics. See the pseudocode for the thread kernel in Algorithm 1.

Output

When the permutations are complete for all variables, the count of randomizations at least as extreme as each variable is divided by the number permutations executed for each variable and written to a data file as p-values. These p-values can then be quickly plotted in a histogram or other diagnostics in other software. If the p-values do not invalidate the test statistic, they can then be converted to FDR-controlling statistics such as q-values (Storey, 2003; Storey and Tibshirani, 2003; Storey et al., 2004), and used to create lists of interesting variables for further biological analysis.

```
input  : input data matrix location in device memory
input  : scratch work area location in device memory
input  : result location in device memory
input  : number of rows,columns,permutations
output: a count for each row in the data matrix. these represent the number of
        randomly permuted test statistics that were more extreme than the test
        statistic observed for each respective row
forall permutations do in parallel by thread block
    initialize ordered column indices in shared block memory, $S$;
    initialize random column indices in shared block memory, $R$;
    forall columns do in parallel by thread within each block
        store random column index in $R$;
    end
    one thread in each block shuffle $S$ according to $R$;
    foreach row $i$ do in parallel by thread within each block
        permute row $i$ according to $S$;
        calculate test statistic for permuted row $i$;
        if random permuted test statistic more extreme than observed then
            increment count for $i$;
        end
    end
end
```

Algorithm 1: Randomization test CUDA kernel

Test statistic functions

Any function of a data row can be used as a test statistic. Typical functions are usually measures of distance or similarity between treatment groups. These measures are then compared similarly to the more restricted contrasts approach in Analysis of Variance. A set of pre-coded test statistics for analyzing two- or three-factor experiments is provided in source code for custom applications, including comparisons of distance metrics such as Canberra distance and Euclidean distance as well as comparisons of similarity measures such as Pearson, Spearman, or Kendall correlation. Equation 2.1 illustrates the use of Euclidean distance for a complex three-factor experiment. x_{ijk} is the abundance measured under the i^{th} treatment for the first factor, the j^{th} treatment for the second factor, and the k^{th} treatment for the third factor. Here, the statistic T is essentially a measurement of the interaction between the first and second factors, in the Euclidean space of the third factor. More complex functions can be written into the source code in the well-labeled statistic function section for custom applications, providing test flexibility and complexity scalability.

$$T_{EucDist} = \sqrt{\sum_k (x_{11k} - x_{12k})^2} - \sqrt{\sum_k (x_{21k} - x_{22k})^2} \tag{2.1}$$

Samples of typical program runs

Testing for speedup and correctness was conducted using simulated data as input. An R script (R Development Core Team, 2010) was written to simulate preprocessed abundance datasets of comparable size to 3-factor experiments found on PlexDB.org (Shen et al., 2005). To test for correctness, a linear model was used to simulate variable behavior through time in two factors. Equation 2.2 describes the linear model. One variable out of the 100 simulated was randomly selected to exhibit an interaction effect.

$$y_{ijk} \quad = \quad \mu + \alpha_i + \beta_j + (\alpha_i \beta_j) + \gamma_k + \epsilon_{ijk} \tag{2.2}$$

where

μ = The mean abundance for all measurements

α_i = the effect of the first factor's i^{th} level

β_j = the effect of the second factor's j^{th} level

γ_k = the effect of the third factor's k^{th} level (the k^{th} time point)

$(\alpha_i\beta_j)$ = the effect of the interaction between α_i and β_j

ϵ_{ijk} $\sim$ $N(0,1)$(Normally distributed random noise)

The model in Equation 2.2 was used to simulate 100 variables. These 100 variables were then analyzed using the GPU application with the statistic generated by Equation 2.1 permuted 1000 times for each variable. The same was done using an R script and the resulting p-values compared. The results were nearly identical, showing correct results from the GPU application. See Figure 2.3.

For speedup profiling, datasets were generated containing different population sizes (data matrix rows) and treatment population sizes (data matrix columns). The number of permutations was also adjusted for testing. A much simpler model for simulation was used for these test to provide easy flexibility in the changing parameters (numbers of data matrix rows, columns, and permutations). Each measurement in the data matrix is simply a pseudo-random number sampled from a Uniform distribution between 0 and 1. Though this model is not realistic, it quickly generates a data matrix of any size for speedup profiling.

Compute time and memory usage for this application are affected by data size and the number of permutations requested. Speedup and GPU memory usage were profiled when increasing each of three values: The number of input data columns (treatments), the number of input data rows (variables), and the number of permutations generated.

Figure 2.4 shows the profiling results when increasing only the number of data columns in input data. Speedup of over 10 is achieved except for two cases. It is unclear why 800 and 1,000 data columns consistently produced small speedup. These slow-downs occurred for many different randomly simulated datasets of different sizes. CUDA runtime register and memory usage details may hide the cause for this (see Corporation (2007)). Figures 2.5 and

2.6 show profiling results when increasing the number of data rows and the number of requested permutations, respectively.

Specifications

This application uses both shared and global GPU memory such that usage is well within the bounds of most NVIDIA CUDA-supporting hardware. Concurrent memory accesses are coalesced to maximize random access bandwidth. Testing and speedup calculation was conducted using an Intel(R) Core(TM)2 CPU X6800 at 2.93GHz with 4096 KB cache and an NVIDIA GeForce 8800 Ultra GPU with 16 multiprocessors at 1.51 GHz with 768 MB global memory (804,585,472 bytes) and 16 KB shared memory per thread block (16,384 bytes).

Hardware requirements

1. NVIDIA CUDA graphics card with Compute Capability 1.0 or higher.

Software requirements

1. NVIDIA CUDA driver for the selected
   graphics card (available from NVIDIA.com).

2. NVIDIA SDK (available from NVIDIA.com).

Availability: This software source is available for on a Subversion (SVN) server at https://subversion.vrac.iastate.edu/Subversion/RandTestGPU/svn/RandTestGPU/.
Note that it requires the NVIDIA SDK mentioned above. Support is available upon request.

Funding: This project is funded by the National Science Foundation Plant Genome Research, DBI 0604755.

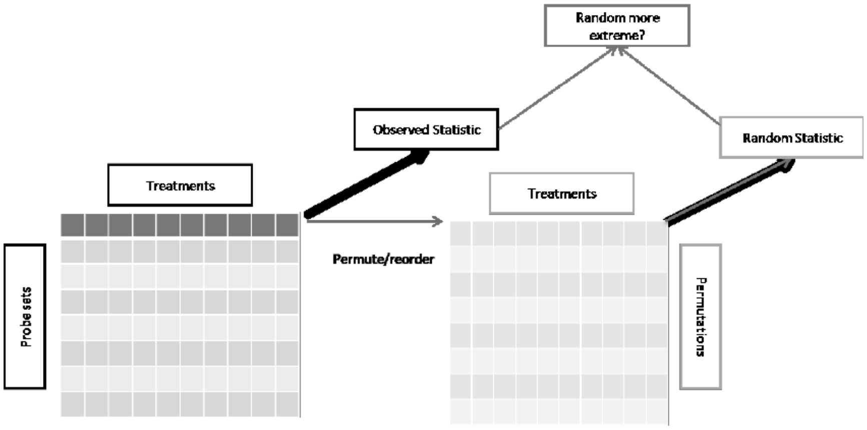
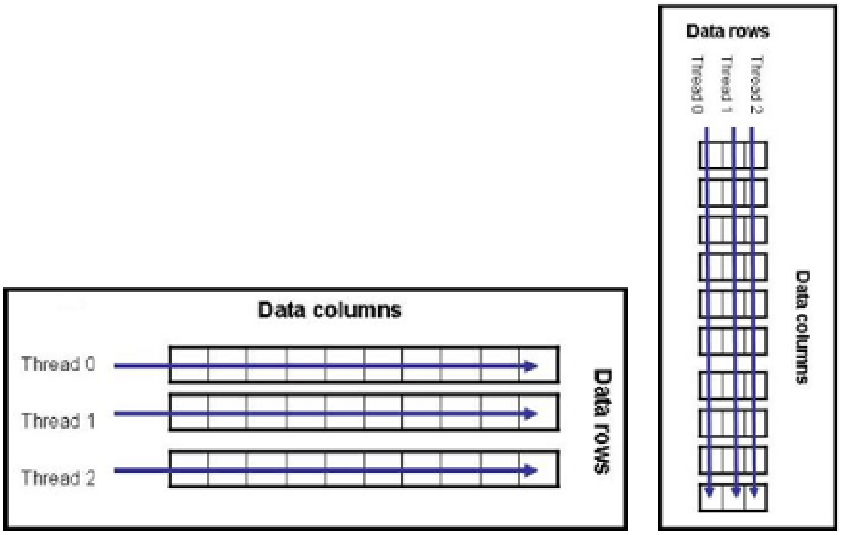


Figure 2.1 Each abundance profile is reordered (permuted) a large number of times and then each permutation is used to calculate a random statistic which is then compared to the observed statistic. The right-hand block represents the set of permutations generated and processed for the first data row in the left-hand block.



(a) Tabular structure commonly representing preprocessed abundance data stored in a typical two-dimensional matrix. This is the un-transposed input data.

(b) The same 3x10 data table after transposition.

Figure 2.2 Transposing abundance data for column-major coalesced memory accesses. Note that as the threads walk along their respective variables, they are accessing contiguous cells in memory in (b) and not in (a). Coalesced accesses require that the threads access contiguous cells in memory simultaneously.(Corporation, 2007)

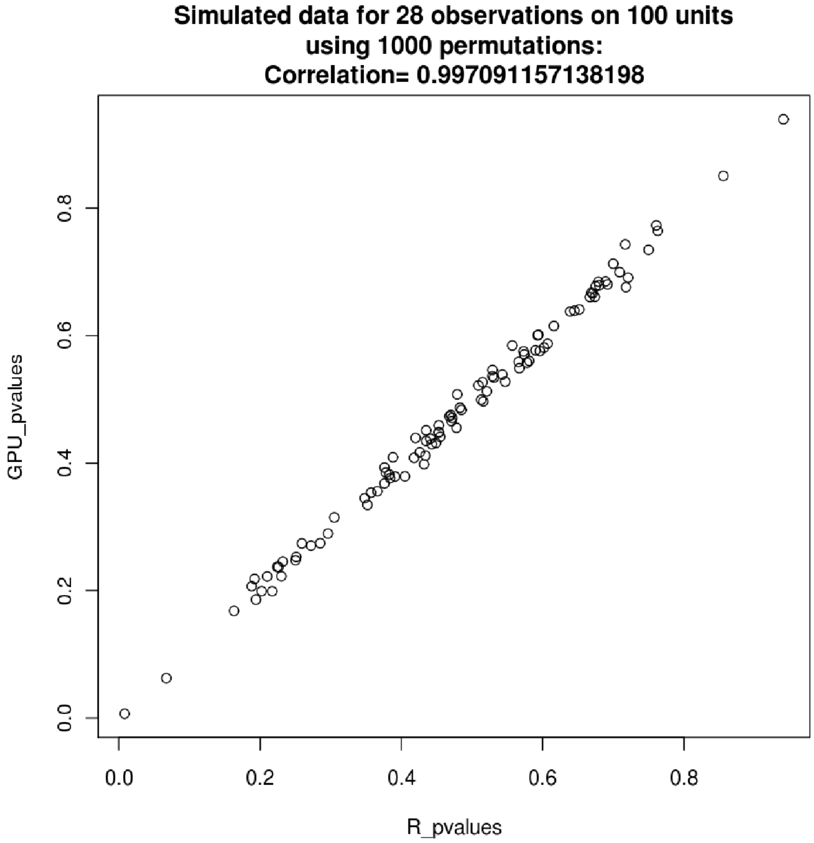


Figure 2.3 Negligible differences in p-values were observed due to word size and randomization differences between the GPU and R approaches.

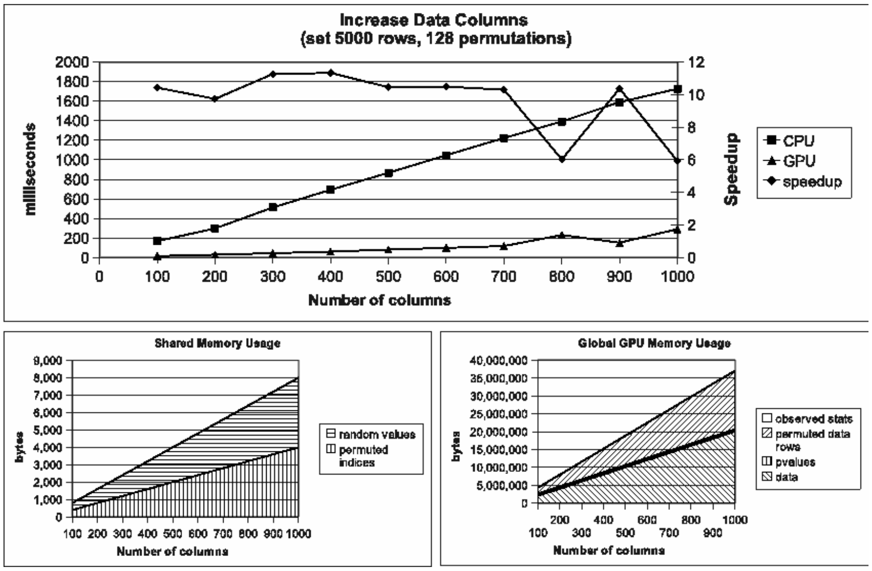


Figure 2.4 Wall clock time and speedup (top), shared memory usage (bottom left), and global device memory usage (bottom right) when increasing the number of input data columns (treatments and/or replications). Speedup of 6-10X is achieved while operating well within memory limitations.

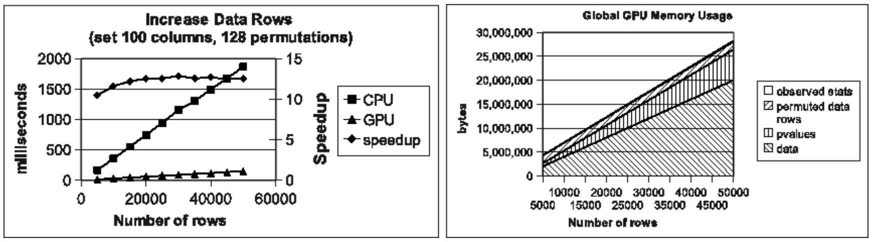


Figure 2.5 Wall clock time and speedup (left), and global device memory usage (right) when increasing the number of input data rows (variables). Speedup of 10-12X is achieved while operating well within memory limitations. Shared memory usage does not change when changing only the number of input data rows.

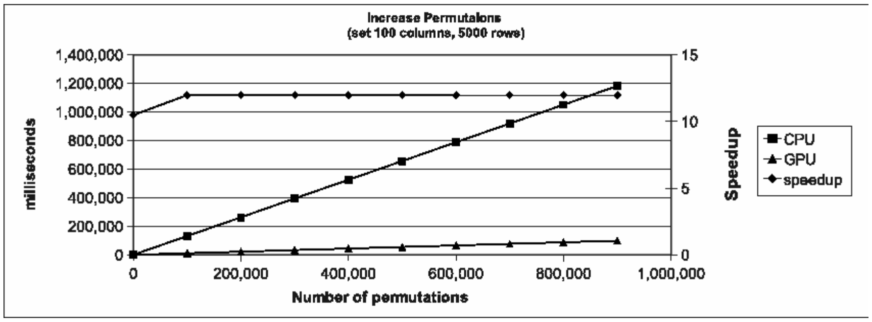


Figure 2.6 Wall clock time and speedup when increasing only the number of permutations requested. Speedup of 12 is achieved. Memory usage does not change when changing only the number permutations.

3. PATHWAYACCESS: CELLDESIGNER PLUGINS FOR PATHWAY DATABASES

A paper published in Bioinformatics

John L. Van Hemert and Julie A. Dickerson

Abstract

Summary:

CellDesigner provides a user-friendly interface for graphical biochemical pathway description. Many pathway databases are not directly exportable to CellDesigner models. PathwayAccess is an extensible suite of CellDesigner plugins which connect CellDesigner directly to pathway databases using respective Java application programming interfaces (API's). The process is streamlined for creating new PathwayAccess plugins for specific pathway databases. Three PathwayAccess plugins, MetNetAccess, BioCycAccess, and ReactomeAccess, directly connect CellDesigner to the pathway databases MetNetDB, BioCyc, and Reactome. PathwayAccess plugins enable CellDesigner users to expose pathway data to analytical CellDesigner functions, curate their pathway databases, and visually integrate pathway data from different databases using standard Systems Biology Markup Language (SBML) and Systems Biology Graphical Notation (SBGN).

Availability:

Implemented in Java, PathwayAccess plugins run with CellDesigner version 4.0.1 and were tested on Ubuntu Linux, Windows XP and 7, and MacOSX. Source code, binaries, documen-

tation, and video walkthroughs are freely available at
http://vrac.iastate.edu/~jlv.

Introduction

CellDesigner (Funahashi et al., 2008) is a tool for graphically building biochemical pathway models which integrate model representation by Systems Biology Markup Language (SBML) (Hucka et al., 2003) with graphical representation by Systems Biology Graphical Notation (SBGN) (Le Novere et al., 2009). There exist many databases providing Application Programming Interface (API) libraries enabling programmatic queries. These API libraries include many biologically meaningful objects which carry out intuitive functions. For example, a Pathway object can report the set of Reaction objects it contains, a Protein Complex object can report the Monomer objects which contsruct it, and a Metabolite object might report its SMILES and InChi codes. The problem is that a Pathway object in one API is not the same as a Pathway object in the API of a different database; The same biological concept is represented using independently developed in-silico representations, preventing any single application from communicating and integrating across databases.

Functionality

PathwayAccess plugins directly interact with pathway databases so that the user can download one or more pathways to a CellDesigner model and upload (or commit) a CellDesigner model to a database. Figure 3.1 shows a dataflow diagram for typical use of the PathwayAccess plugins.

The PathwayAccess plugin framework confers three major benefits, depending on whether individual database API's support data retrieval and modification. Firstly, the plugins make pathways stored in remote databases available to the powerful modeling and simulation functionality already provided by CellDesigner. Secondly, SBGN implemented by CellDesigner provides a standard representation for biologists to curate pathway databases; the user can create a pathway model and commit it to the database of his choice. A user can also download

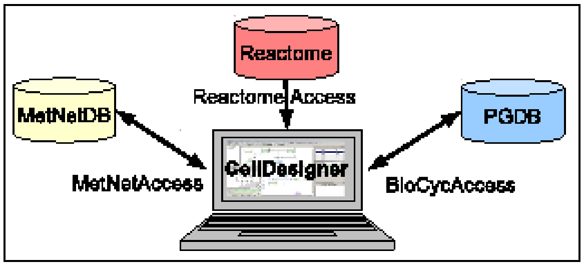


Figure 3.1 Dataflow for PathwayAccess plugins. PathwayAccess plugins use respective APIs to communicate with different pathway databases and integrate data in CellDesigner. As indicated by arrows, depending on functionality supported by the datasource, dataflow is uni- or bi-directional.

a pathway model from a database, edit it, and commit it back to the database, either replacing the original pathway or creating a different version. Thirdly, CellDesigner can be effective in visually comparing and integrating pathway data from one or many different databases; metabolic networks can be downloaded directly into CellDesigner and integrated into custom super-pathways. CellDesigner can export pathways into files for loading into other software such as Cytoscape (Shannon et al., 2003), where SBGN is an ancilliary feature to network analysis functions.

Since CellDesigner and most datasources' user interfaces provide good automatic layouts, layouts are left to the datasources and CellDesigner independently.

Pathway Integration Across Databases

When PathwayAccess plugins download pathways, they are integrated with the growing model in memory. CellDesigner is suited to support integration because it uses the XML-based SBML data model not only for file storage, but also for objects in memory– ideal for representing annotations integrated from different sources. Among other annotations, PathwayAccess stores synonyms this way, enabling it to match integrated objects in the same subcellular compartment that may be named differently across databases. To prevent duplicate reactions in integrated pathways, a reaction hashing algorithm calculates a unique integer for every combination of reaction substrates, products, and catalysts (see Additional Material). Each

PathwayAccess plugin has a unique, but editable highlight color, which can be used to color the model objects downloaded using that plugin. Objects from multiple databases are colored by mixing the colors of the plugins that downloaded them.

Creating New PathwayAccess Plugins

The PathwayAccess framework includes a core library plus one or more independent plugins. A plugin developer can easily create a new CellDesigner plugin which communicates with any pathway database providing a Java API. Simply create a new CellDesigner plugin object using the PathwayAccess library and define a set of simple database query operations, depending on whether the plugin will support download and/or saving a model to the database. To create a PathwayAccess plugin which downloads a pathway, define 18 simple functions such as get the synonyms of an object (pathway, metabolite, gene, etc). To design a commit feature, define nine simple functions such as add substrates to a reaction in the database. With these simple operations defined for communicating with a database, PathwayAccess handles all interaction both with CellDesigner and the database, similarly to Cytoscape's Data Integration Request For Comments (Killcoyne and Pico, 2009), and provides a way to enrich objects beyond the annotation used for integration.

Examples

Three PathwayAccess plugins, MetNetAccess, BioCycAccess, and ReactomeAccess were created. In addition to representing biological objects differently, each uses a different communication protocol: SQL, Sockets, and Web Services, respectively.

BioCycAccess: Download and Commit to a PGDB. BioCyc databases are individually deployed for specific organisms and purposes (Karp (2005); Karp et al. (2005), http://www.biocyc.org). BioCycAccess uses JavaCycO, our new library wrapped around the JavaCyc API (Mueller et al., 2005; Krummenacker et al., 2005), running in client mode to connect to a BioCyc Pathway Genome Database (PGDB) that is running JavaCycO in server mode. It supports both downloading and committing pathways.

ReactomeAccess: Download from Reactome. Reactome is a large repository for pathways (Vastrik et al., 2009). ReactomeAccess supports downloading pathways from Reactome directly into CellDesigner models via an API wrapped around Reactome's Web Services.

MetNetAccess: Download and Commit to MetNet. MetNetAccess provides CellDesigner access to the pathway database MetNetDB using MetNetAPI (Sucaet and Wurtele, 2010), which is wrapped around SQL queries. It supports both downloading and committing pathways. MetNetDB is an integrated pathway database that currently includes Arabidopsis *thaliana*, yeast, soybean, and the grapevine. MetNetAccess has been used to curate many pathways for different organisms in MetNetDB (Wurtele et al., 2007). MetNet allows public downloading of data, but only registered curators may modify data in MetNetDB.

Impact

The PathwayAccess suite of CellDesigner plugins is a powerful tool for researchers who work with metabolic pathway data and wish to take advantage of graphical and computational CellDesigner features. By directly accessing and publishing to pathway databases, decentralized pathway integration and comparison is made possible over simply saving and loading SBML files. While three PathwayAccess plugins have been released, the practical scope of the PathwayAccess library is as wide as the number of databases to which CellDesigner can connect because communication requires a Java API. MetNetAccess, BioCycAccess, ReactomeAccess and future PathwayAccess plugins enable CellDesigner users to expose pathway data to analytical CellDesigner functions as well as visually integrate and curate pathway data from different databases using standard SBGN– something which has been previously prevented by disparate in-silico representations of biological objects.

Discussion of Technical Solutions

The following sections were included as as supplementary technical description of this work.

Importing and integrating pathways

PathwayAccess effectively communicates with pathway databases to integrate pathway models in CellDesigner. The main challenge of integration is preventing redundant objects–both species and reactions.

Preventing duplicate species

During an import, there are two sides to our solution: On the database side, a plugin must be able to retrieve all synonyms of a generic pathway object that is to become a CellDesigner species. The PathwayAccessPlugin abstract class requires all extending subclasses (which are the specific plugins, such as BioCycAccess) implement a function that retrieves all synonyms from its database for a given generic pathway object. If the database does not support synonyms, the plugin should at least return a list where the single member is the name of the generic object.

On the CellDesginer side, imported species must maintain a list of synonyms that is persistent through file saves. For this we designed a simple XML schema that is inserted into the CellDesigner Species Notes, which are seen on the botton right corner of the CellDesigner screen when a species is selected (see procedure AddAnnotation in Section 3 of this document). Using XML this way confers several benefits: 1) PathwayAccess annotations are easily parsed using the libSBML library, which is the core of CellDesigner, 2) CellDesigner models are saved in XML format by default, so the XML annotations fit nicely within these saved files and are persistent, and 3) our schema is simple enough that the PathwayAccess annotations are human-readable 4) PathwayAccess annotation remains attached to species object in CellDesigner and SBML. There are two issues, however: 1) The SBML specification does not allow custom XML be added to the Notes field of objects, and 2) if a user adds other text to the Notes field of an object, PathwayAccess is unable to parse the XML. Despite these, issues, PathwayAccess plugins do perform their goals in CellDesigner as along as the user does not add text to the Notes fields of objects or mind seeing SBML warnings during model saves and loads. Also note that PathwayAccess considers object IDs and names specific to databases to be synonyms as well, and IDs are always first searched in the existing model to find the same ID from the same database without searching synonyms.

Before a pathway is imported to a CellDesigner model, a dictionary of synoyms mapped to species objects is built from all existing species in the model, and used to look up speces by synonym (see procedure ImportPathway in Section 3 of this document).

Preventing duplicate reactions

Even if duplicate species are created, duplicate reactions can easily appear when integrating overlapping pathways. Our solution represents the parts of a reaction (inputs and outputs) as a string and converts that string to a unique integer using Java's hash code function for strings (see procedure ReactionHash in Section 3 of this document). The key is that we can build complete CellDesigner reactions in memory *before* adding them to the model on the screen. This way, we can build each imported reaction in memory, where its inputs and outputs (reactants, modifiers, and products) are species that have been either newly created or looked up using IDs and synonyms (if the reaction is indeed a duplicate, all inputs and outputs will have been found in the model and not created). CellDesigner assigns model-specific IDs to objects (species and reactions) per SBML standards, so we build the reaction parts-representing string using these SBML IDs. The type of the reaction is also added to the string. Currently, the location of the reaction is ignored because of the volatility of this annotation in different databases for reactions. The ID members of a reaction-representing string are always sorted before creating the hash value of their concatenation. This way, if a two reactions have exactly the same inputs and outputs and they are of the same reaction type, they have the same reaction hash value. See procedure ImportReaction in Section 3 of this document.

Before a pathway is imported to a CellDesigner model, a dictionary of reaction hash values mapped to reaction objects is built from all existing reactions in the model, and used to look up reactions by hash value (see procedure ImportPathway in Section 3 of this document).

Caching database queries

Database query performance is dependent on the individual plugins that use the PathwayAccess library. As stated in the manuscript, our three PathwayAccess plugins use three different communication protocols, each with their own performance strengths and weaknesses.

BioCycAccess

BioCycAccess uses our software, JavaCycO (http://vrac.iastate.edu/∼jlv/javacyc), to connect to a local or remote BioCyc database. Communication uses a socket protocol that issues Lisp code to and from the server, plus some special commands we developed for searching. JavaCycO operates in two modes: 1) client mode, and 2) server mode. Within the BioCycAccess plugin, JavaCycO operates in client mode. In client mode, JavaCycO maintains a cache of objects in local memory in the database so that their information need not be repeated loaded.

To connect to a remote BioCyc database, BioCycAccess must communicate with a server running JavaCycO in server mode along with the PathwayTools software. In both modes, JavaCycO maintains a search cache mapping IDs, names, and synonyms to lists of objects in the database, which is used to lookup existing objects during a pathway commit to the database. If the client instance does not have a synonym in its cache, it asks the server instance to search for it. Since the server mode instance of JavaCycO is persistent across many clients, most clients will build their own cache by issuing search queries to the server; if a server mode instance of JavaCycO runs long enough and is used by enough clients, it will build a cache close to the complete database and become very fast when searching for synonyms.

ReactomeAccess

ReactomeAccess uses an API we designed that wraps around the webservice provided by Reactome.org to query its database. ReactomeAccess includes a custom connection object we designed to maintain a cache of Reactome objects as they are loaded via the webservice.

MetNetAccess

MetNetAccess communicates with MetNetDB using
MetNetAPI (http://metnet3.vrac.iastate.edu/api/), which is a collection of objects wrapped around SQL queries to the MetNetDB MySQL database. MetNetAccess leaves caching to MetNetDB, as MySQL databases do maintain query caches.

Committing a model to a database

Only PathwayAccess plugins that connect to databases supporting data modification can commit a CellDesigner model to the database. JavaCycO supports free data modification (only expose your development PGDBs to JavaCycO for now), so BioCycAccess can commit a model to a PGDB. MetNetAPI supporst data modification with user authentication, so MetNetAccess can only commit a model to MetNet if the user has logged into MetNetDB with a privileged user account. Reactome does not support data modification, so it is not an option for ReactomeAccess.

All PathwayAccess plugins must implement three object initialization functions: initialize a pathway, initialize a reaction, and initialize a species (aka 'entity'). Each of these functions take as input the analogous object from CellDesigner (CellDesigner model is to generic pathway object as CellDesigner reaction is to generic reaction object as CellDesigner species is to generic entity object) and first must search the database for the species in the database. Recall that all PathwayAccess annotation is available withing the CellDesigner species object because it is contained in the Notes field as XML; If the species came from the database, it's database-specific ID is stored there, along with all names and synonyms from all databases it came from. Other database-specific annotations are stored there as well. For example, MetNetDB uses a confidence annotation, BioCyc databases provide a comment field, and both MetNetDB and BioCyc share an EC field for zero or more EC numbers. If a match is found, plugins must *clear* the object so that all information for it in the CellDesigner model can overwrite existing information about it in the database. Else, the plugin must create the object in the database and return it in an 'empty' state, whatever that may mean for the particular database and API. It is left to the plugin to handle multiple hits in the database. MetNetAccess and BioCycAccess both use special convenience functions of the PathwayAccess library to prompt the user to select one out of the multiple hits from the database when such a search result occurs, and they remember the selection for subsequent attempts to initialize the object during the same commit operation.

PathwayAccess plugins must also implement functions to add inputs (reactants and modifiers) and outputs (products) to reaction objects in the database.

Database API's manipulate biological object locally in memory and if they support modifying the database, they provide some sort of object commit function that actually writes the object to the database. All PathwayAccess plugins must implement a commit object method that takes as input one of its generic pathway database objects and writes it to the database. See procedure CommitModel in Section 3 of this document.

Pseudocode

The following procedures are the important operations in PathwayAccess that allow for integration from different databases as well as curation. See http://vrac.iastate.edu/~jlv/pathwayaccess/ for documentation, binaries, and source code.

This pseudocode is somewhat detailed to communicate exactly how and when the procedures interact. Note that all of these procedures are implemented in the PathwayAccess library and *not* in the plugins; *new plugin developers never need to implement these procedures* because we have already done so and provided the PathwayAccess library for it. They only need to implement simpler database communication functions. In pseudocode below, these required functions are used when the procedures "ask plugin to..." These statements are first listed to communicate the requirements for creating a new PathwayAccess plugin. See them in more detail by clicking the API Documentation link at http://vrac.iastate.edu/~jlv/pathwayaccess/.

Acknowledgement Thank you to Yves Sucaet and Eve Wurtele for developing MetNetAPI and to Jesse Walsh for testing JavaCycO. This work is funded by the National Science Foundation Plant Genome Research Program, DBI 0604755, and EEC 0813570.

Procedure Calls to plugin that developer must implement

```
// This is not a procedure, but the set of lines in the following procedures
   which ask the plugin for something.
GenericObjectsList $\leftarrow$ ask plugin for all of genericPathwayObject's reactions;
id $\leftarrow$ ask plugin for ID of genericReactionObject;
GenericObjectsList $\leftarrow$ ask plugin for all of genericReactionObject's reactants;
GenericObjectsList $\leftarrow$ ask plugin for all of genericReactionObject's modifiers;
GenericObjectsList $\leftarrow$ ask plugin for all of genericReactionObject's products;
id $\leftarrow$ ask plugin to retrieve its database's unique ID of object;
name $\leftarrow$ ask plugin to retrieve the name of object;
type $\leftarrow$ ask plugin to retrieve the CellDesigner type of object ;  // CellDesigner constants
   like SIMPLE_COMPOUND or GENE
location $\leftarrow$ ask plugin to retrieve the subcellular location of object;
Synonyms $\leftarrow$ ask plugin to retrieve any synonyms of object;
genericPathwayObject $=$ ask plugin to intialize and commit the current CellDesigner model as
   a pathway;
genericReactionObject $\leftarrow$ ask plugin to initialize Reaction;
object $\leftarrow$ ask plugin to initialize and commit species;
ask plugin to add object to genericReactionObject as a reactant;
object $\leftarrow$ ask plugin to initialize and commit species;
ask plugin to add object to genericReactionObject as a modifier;
object $\leftarrow$ ask plugin to initialize and commit species;
ask plugin to add object to genericReactionObject as a product;
ask plugin to commit genericReactionObject;
ask plugin to add genericReactionObject to genericPathwayObject;
```

Procedure ImportPathway(*plugin,genericPathwayObject,model*)

```
input  : a PathwayAccess plugin, plugin
input  : a generic pathway object defined by the plugin's database API, genericPathwayObject
// First, build a mapping from reaction hashes to reactions already in the
   model
initialize ReactionHashes;
ReactionsList ← ask current CellDesigner model for all existing reactions;
foreach Reaction in ReactionsList do
    hash ← ReactionHash(Reaction);
    add (hash ⇒Reaction) to ReactionHashes;
end
// Next, build a mapping from species names and synonyms to species already
   in the model
initialize SpeciesDictionary;
SpeciesList ← ask current CellDesigner model for all existing species;
foreach species in SpeciesList do
    foreach name in all NAME annotations for species do
        add (name ⇒species) to SpeciesDictionary;
    end
end
// Next, begin importing generic reactions
GenericObjectsList ← ask plugin for all of genericPathwayObject's reactions;
foreach genericReactionObject in GenericObjectsList do
    Reaction ← ImportReaction(plugin,genericReactionObject);
    id ← ask plugin for ID of genericReactionObject;
    AddAnnotation(Reaction, "plugin.ID", id);
end
```

Procedure `ImportReaction`(*plugin,genericReactionObject,ReactionHashes*)

```
input  : a PathwayAccess plugin, plugin
input  : a generic reaction object defined by the plugin's database API, genericReactionObject
input  : a dictionary of reaction hash values mapped as keys to reactions in the model,
         ReactionHashes. This is populated before every new download operation by reading
         the XML node HASH stored in all reactions' Notes attributes. See procedure
         ReactionHash.
output: result, either a newly created CellDesigner reaction object, or the matching reaction
         that already exists in the model
result ← create new, empty CellDesigner reaction;
GenericObjectsList ← ask plugin for all of genericReactionObject's reactants;
foreach object in GenericObjectsList do
    species ← ImportSpecies(plugin,object);
    add species to result as a reactant;
end
GenericObjectsList ← ask plugin for all of genericReactionObject's modifiers;
foreach object in GenericObjectsList do
    species ← ImportSpecies(plugin,object);
    add species to result as a modifier;
end
GenericObjectsList ← ask plugin for all of genericReactionObject's products;
foreach object in GenericObjectsList do
    species ← ImportSpecies(plugin,object);
    add species to result as a product;
end
hash ← ReactionHash(result);
if ReactionHashes not contains key hash then                // it is a new reaction
    add result to current CellDesigner model;
    add (hash ⇒result) to ReactionHashes;
    return result;
else                                                  // it is a redundant reaction
    destroy result;
    return ReactionHashes lookup hash;
end
```

Procedure ReactionHash(*reaction*)

```
input  : a CellDesigner reaction, Reaction
output: hash, a unique integer for the type, reactants, modifiers, and products of Reaction
initialize PartsList ;          // a list of strings representing the parts of Reaction
add Reaction's type to PartsList;
foreach species in all Reaction's reactants, modifiers, and products do
    add species ID to PartsList;
end
sort PartsList;
PartsString ← convert PartsList to a single concatenated string;
hash ← JavaStringHashCode(PartsString);                  // Java's hashing algorithm
return hash;
```

```
Procedure ImportSpecies(plugin,object)

input  : a PathwayAccess plugin, plugin
input  : a generic biological object defined by the plugin's database API, object
input  : a dictionary of names, synonyms, and database IDs, SpeciesDictionary.
output: result, either a new Species, or existing match.
id ← ask plugin to retrieve its database's unique ID of object;
name ← ask plugin to retrieve the name of object;
type ← ask plugin to retrieve the CellDesigner type of object ;   // CellDesigner constant
location ← ask plugin to retrieve the subcellular location of object;
if location is unknown then
    location ← cytosol;
end
Synonyms ← ask plugin to retrieve any synonyms of object;
key ← 'plugin id';                                  // search for ID from same database
if SpeciesDictionary contains key then
    result ← SpeciesDictionary lookup key;
else
    key ← 'location type name';                                   // search for name
    if SpeciesDictionary contains key key then
        result ← SpeciesDictionary lookup key;
    else
        foreach synonym in Synonyms do
            key ← 'location type synonym';                     // search for a synonym
            if SpeciesDictionary contains key key then
                result ← SpeciesDictionary lookup key;
                break;                                                  // be greedy

        end
    end
    if result null then                   // No match found.  Create a new species.
        result ← create new species of type type in location named name;
    end
    AddAnnotation(result, ”plugin.ID”, id));
    AddAnnotation(result, ”plugin.NAMES”, name ∪ Synonyms);
    add to SpeciesDictionary mapping 'plugin id'⇒result;
    add to SpeciesDictionary mapping 'location type name'⇒result;
    foreach synonym in Synonyms do
        key ← 'location type synonym';
        add (key ⇒result) to SpeciesDictionary;
    end
end
return result;
```

Procedure AddAnnotation(*sbase,label, Values*)

```
input  : a CellDesigner SBML object, sbase
input  : an annotation label, label
input  : a set of values to add under label, Values
// PathwayAccess annotations are stored by building a simple XML tree in the
   Notes attribute of a CellDesigner species or reaction:
// <List name="label part 1">
//   <List name="label part 2">
//    <Item value="Values item 1">
//    <Item value="Values item 2">
//    <Item value="Values item 3">
// label can be a hierarchy path such as "MyPlugin.NAMES"
LabelParts ← split label on delimiter ('.');
notesXML ← get Notes attribute for sbase;
XMLtarget ← find XML List node in notesXML referred to by LabelParts;
if XMLtarget null then
    XMLtarget ← build XML List node in notesXML referred to by LabelParts;
end
add Values as XML Items under XMLtarget;
```

Procedure `CommitModel(`*plugin*`)`

```
input : a PathwayAccess plugin, plugin
organism ← ask user to select organism from those in plugin's database;
ask the plugin to validate the model for commit; // this is where MetNetAccess checks
    for privileged user authentication
if invalid then
    show error message;
    return;
end
genericPathwayObject = ask plugin to intialize and commit the current CellDesigner model as
    a pathway;
ReactionsList ← ask current CellDesigner model for all existing reactions;
foreach Reaction in ReactionsList do
    genericReactionObject ← ask plugin to initialize Reaction;
    foreach species in Reaction reactants do
        object ← ask plugin to initialize and commit species;
        ask plugin to add object to genericReactionObject as a reactant;
    end
    foreach species in Reaction modifiers do
        object ← ask plugin to initialize and commit species;
        ask plugin to add object to genericReactionObject as a modifier;
    end
    foreach species in Reaction products do
        object ← ask plugin to initialize and commit species;
        ask plugin to add object to genericReactionObject as a product;
    end
    ask plugin to commit genericReactionObject;
    ask plugin to add genericReactionObject to genericPathwayObject;
end
```

4. DISCRIMINATING OMICS RESPONSE GROUPS IN BIOCHEMICAL PATHWAY NETWORKS

A paper to be submitted to Nucleic Acids Research

John L. Van Hemert [1,2,3] and Julie A. Dickerson[1,2,3,4]

Abstract

Analysis of Omics experiments generates lists of entities (genes, metabolites, etc) selected based on specific behavior. Functional interpretation of these lists usually entails some sort of catorgy enrichment tests using functional annotations like Gene Ontology terms. We present a method for interpreting Omics lists in the context of metabolic pathway and regulatory networks using directed stochastic modeling of the networks themselves. We also present web tool for using our method and a proof of concept application to an *E. coli* transcriptomics data set where we used the web tool to confirm common knowledge of the importance of Lipid A and posit a model for *E. coli* response to Lipid A deprivation. Intuitively, the main theme is response to osmotic stress, but we also were able to detect other responses that are supported by the literature.

Introduction

Analysis of Omics experiments generates lists of entities (genes, metabolites, etc) selected based on specific behavior. Common practice is to leverage existing functional knowledge of

[1] Electrical and Computer Engineering
[2] Bioinformatics and Computational Biology
[3] Iowa State University, Ames, Iowa
[4] Author for correspondence

the entities in a list by further listing the functional annotations assigned to the members of the list. Category enrichment analysis generally refers to testing the null hypothesis that the distribution of functional annotation in the entity list is similar to the distribution of functional annotation for all entities (Nettleton et al., 2008; Barry et al., 2005; Subramanian et al., 2005; Maere et al., 2005). If that hypothesis is rejected, one or more of the functional annotations in the entity list is either over- or under-represented, and a general functional response or perturbation is inferred for the experimental treatment and specific test used to generate the entity list. For example, most plant biologists mine a number of sets of genes from results that exhibit an expected behavior of biological interest and then map the selected genes to static functional annotation and then manually or computationally determine which functions are associated with the behavior in the experiment (Nettleton et al., 2008), (Maere et al., 2005). Knowledge is often digitally stored as networks, whether it is ontological (Ashburner et al., 2000; Cordero et al., 2009; Avraham et al., 2008) or biochemical (e.g., Reactome (Vastrik et al., 2009), KEGG (Okuda et al., 2008), PathwayTools/BioCyc (Krummenacker et al., 2005), and MetNetDB (Wurtele et al., 2007)). This makes functional analysis much more complex than simple set comparisons, requiring more complex tools like MapMan (Thimm et al., 2004; Rotter et al., 2009; Usadel et al., 2009), Array2KEGG (Kim et al., 2010), or KEGG Spider (Antonov et al., 2008) to name a few. Most plant pathways are stored in AraCyc, PlantCyc (both BioCyc Pathway Genome Databases (PGDBs)), and MetNetDB. These resources provide web-based access to simple pathway visualizations, searching, links to other databases, as well as some basic analysis tools.

Category enrichment is unable to directly infer causality; if a functional annotation term is enriched in an entity list, we cannot determine whether the function is somehow causing the perturbation of the members of the entity list, or members of the entity list are themselves a response to some other signal and the enriched function is a response to the entities.

At the same time, biochemical pathway models are accumulating in central repositories such as BioCyc, MetNetDB, Reactome, and KEGG databases. These pathway models use and assign functions to entities by placing them in networks of chemical reactions. Pathway annotation has also been used in category enrichment where entities are annotated with the names of the pathways in which they participate. Unfortunately, pathway annotation enrichment suffers from the same general problems as category enrichment plus it fails to consider the inter-connectivity and reactive relationships between different entities, reactions, and pathways; it is no different from common functional annotation enrichment analysis.

Our purpose in this work is to provide a methodology and tool for discriminating groups of entities (*Response Groups*) in a pathway network which are highly connected to a *Query List* of entities which results from a previous selection from Omics data. Such a tool has several requirements:

1. Receive as input a biochemical pathway network structure
2. Receive as input a Query List of entities referred to by nodes in the pathway network. Entities in a Query List could be any combination of genes, enzymes, chemical compounds, or reaction events in the pathway network.
3. Receive as input a definition of Response Groups to discriminate. Response Group compartmentalization must be flexible; Response Groups can be the set of all functional pathways in the network, all reactions in the network, or the set of all compound classes in the network, for example.
4. Response Groups must be able to overlap on entities; Entities, both members and non-members of the Query List, must be able to be members of multiple Response Groups.
5. The set of all Response Groups need not cover then entire pathway network; not all nodes in the pathway network are guarranteed to be a member of any Response Group.

Background Terminology

A graph or network is a set of vertices or nodes connected by a set of edges.

A connected component in a graph is a set of nodes and edges where there is a path from each node in the connected component to all other nodes in the connected component.

An adjacency matrix, $C_{N \times N}$, is a square matrix representation of a graph with N nodes, where C_{ij} indicates the weight of the edge from node i to node j. Zeros indicate the absence of an edge and unweighted graphs use the same weight value througout all existing edges. Undirected graphs have the property of symmetry where $C_{ij} = C_{ji}$.

A stochastic matrix is a matrix whose rows and/or columns sum to one. If the rows sum to one, the matrix is *right stochastic*. If the columns sum to one, the matrix is *left stochastic*. If both the rows and columns sum to one, the matrix is *doubly stochastic*.

A state transition probability matrix, $A_{N \times N}$, is a stochastic adjacency matrix where A_{ij} represents the probability of a transition from state (or node) j if the system is currently in state i.

A sparse matrix is one which contains mostly zeroes. A sparse matrix can be stored in a way that avoid storing zero-values, saving space and compute time in operations on the sparse matrix. In contrast, a *dense* matrix is one with relatively few zeroes. Large, dense matrices are difficult to process because they contain such a large number of values.

Omics refers to high-throughput biological experiments which quantify a large number of variables (thousands, or even millions) simultaneausly during specific treatments or perturbations. For example, *genomics* refers to the study of an organism's *genome*, *transcriptomics* refers to the study of all RNA-encoded gene transcripts in an organism's *transcriptome*, and *metabolomics* refers to the study of all metabolites in an organism's *metabolome*. Analysis of Omics experiments generates lists of entities (genes, metabolites, etc) selected based on specific behavior.

Existing uses of flow simulation

Graphical Clustering

Flow simulation is used by the tool MCL ("Markov Clustering") (van Dongen, 2000). Input is an undirected, weighted network of nodes edges. The algorithm take successive powers of the stochastic state transition probability matrix, with an inflation step at each iteration based on a single inflation parameter which degrades low-flowing edges until they vanish, creating a set of connected components which represent the resulting clusters. This method is useful for clustering data based on the structure of some meaningful undirected graph representing it, such as a correlation network as in Mao et al. (2009).

The Random Walk Kernel

Graph kernels are functions which take as inputs adjacency matrices for two graphs and return as results some metric that usually compares the two networks (Vishwanathan et al., 2010). A random walk kernel is a kernel which conducts operations on the input matrices which simulate random walks along the edges of the input matrices' networks. Towfic et al. (2010) have used a state transition probability matrix multiplication called the random walk kernel to infer homologues from protein interaction networks.

Exsting methods model undirected flow. Many kernels and other stockastic flow-based methods for processing networks assume the network is *undirected.* This means edges in the network pair their respectively connected nodes in no particular order. Conversely, *directed* networks' edges have a specific ordering; one of the nodes a directed edge connects is the *source* and the other node is the *target.* The direction of the edges goes from *source* to *target.* However, biochemical pathways are often modeled as directed networks. Nodes in a pathway network represent both physical entities such as genes or enzymes, as well as intangible events such as chemical reactions. Edges between such nodes represent interaction (ie regulation or conversion) and/or participation in an event (ie catalysis of a reaction). Direction is necessary to indicate the direction of reactions, ie, which participants are catabolized and which are anabolized in a particular reaction.

Other graph theoretical metrics and properties

If we model metabolism as a directed network, there are many well-studied metrics and methods for analysis available. *Betweenness* is the measure of a graph object's (node or edge) centrality in the graph by counting the number of shortest pathways between all nodes in the graph pass through it. *Degree/Hubness* is the number of edge connections to a particular node in a graph. *Density* is a measure of the number of edge connections in a graph, calculated by dividing the number of edges in the graph by the maximum non-redundant edges possible (Opsahl et al., 2010). *Scale-free* networks have a degree distribution that follow a power law, which is a relationship between two quantities (here, it is node degree and node degree frequency) where one quantity is a power function of the other. Natural networks are often scale-free because it is often the case where there are a few highly connected central hubs in the network while the rest of the nodes are less connected (Barabasi and Bonabeau, 2003).

Modeling directed random walks

Directed flow simulation is possible using stochastic state transition probability matrices, but the matrices are not guarranteed to be doubly-stochastic. Here, we conceptualize only right-stochastic state transition probability matrices by using them to represent a random walk on the network of a finite number of steps; in a given step in a random walk on the graph represented by right-stochastic state transition probability matrix, $A_{N\times N}$, if we are standing at node i, we must take a step somewhere, so the sum $\sum A_{i\cdot}$ must equal one. On the other hand, if a random walk in $A_{N\times N}$ lands on node j, the sum $\sum A_{\cdot j}$ may be less or greater than one, indicating the walk hit j, but may not have come from any other node, or have an invalide probability greater than one, respectively. We will avoid the non-left-stochastic contradiction by only considering random ”forward” steps from i to j on edges A_{ij}.

For a state stransition probability matrix, $A_{N\times N}$, the state stransition probability matrix in exactly w steps is A^k; the probability of transitioning from state i to state j in w steps is

${A^w}_{ij}$. Therefore, the sum of successive powers of A up to A^w, $M_{N\times N}$ would be the matrix of hit rates in a random walk of length w steps (Equation 4.3).

$$\begin{aligned} A_{N\times N} &= \text{The directed, non-symmetric state transition probability matrix} && (4.1)\\ A_{ij} &= C_{ij}/\sum C_{i\cdot} && (4.2)\\ M_{N\times N} &= \sum_{s=1}^{w} A^s, \text{ for a random walk of } w \text{ steps} && (4.3) \end{aligned}$$

Given a biochemical pathway network represented by a weighted adjacency matrix, $C_{N\times N}$, we can row-stochastize it to fit the form of A in Equation 4.1 by dividing the values in C by the sum of their respective row as in Equation 4.2. This simple process is not often used on large networks (thousands of nodes or more) because computational space limitations. Indeed, adjacency matrices for most networks are sparse, but as successive powers are summed, the result quickly because dense and difficult to process. Fortunately, biochemical pathway networks, while sparse, contain several hub nodes, which are highly connected to the rest of the network (ie, water and energy molecules), allow relatively short random walk models (10-20 steps) to cover most of a pathway network. The resulting matrix M is the matrix of hit rates on random walks between nodes; M_{ij} is the hit rate at j of random walks of size w steps starting at i. Generally, we call this metric 'random walk flow'.

Summarizing random walk flow between groups of nodes

Our original problem involves a list of nodes in a pathway network and comparing it to different groups of nodes in the same network. For example, our query list might be a list of genes that are differentially expressed under a specific condition and the response groups could be the functional pathways defined by common biological knowledge (a pathway is a subset of nodes and edges in the entire network that are commonly associated with a specific process or function, ie glycolysis). To summarize the flow between a query list and a given response group, we take the sum of flow (or random walk hit rates) between nodes in the query list and nodes in the response groups. This is a simple matrix operation using a reponse group membership indicator matrix, $\Upsilon_{N\times G}$, where $\Upsilon_{ng} = 1$ if node n is a member of response group g and zero otherwise. The matrix product of the matrices $M_{N\times N}$ and $\Upsilon_{N\times G}$, $\Psi_{N\times G}$ contains the the sums of flow from each node and the nodes in each response group (Equation 4.4). We then take the matrix product of an indicator vector, $Q_{1\times N}$, and $\Psi_{N\times G}$, where $Q_n = 1$ if n is in the query list and zero otherwise. The result, $\Theta_{1\times G}$, is the vector of sums of flow from the nodes in the query list to the nodes in each response group (Equation 4.5).

$$\begin{aligned} \Psi_{N\times G} &= M_{N\times N}\Upsilon_{N\times G} && (4.4) \\ & \quad \text{where } \Upsilon_{ng} = 1 \text{ if node } n \text{ is in response group } g\text{, 0 otherwise} \\ \Theta_{G\times 1} &= (Q'_{N\times 1}\Psi_{N\times G})' = (Q'_{N\times 1}\,(M_{N\times N}\Upsilon_{N\times G}))' && (4.5) \\ & \quad \text{where } Q_n = 1 \text{ if node } n \text{ is in the query list, 0 otherwise} \end{aligned}$$

Reversing directionality

The previous formulation results in random walk flow summarizations from the query list to response groups, ie pathways which genes in a query list regulate. The question of what is regulating the query list, or signalling its members to change behavior, is often equally or even more interesting. We can reverse direction rearranging the matrix multiplication to result in another vector of flow rates for each response group, only these represent flow summaries from the response groups to the query list (Equations 4.7-4.9). We must first re-initialized the random walk rate matrix, A, as $A^{(rev)}$ by left-stochastizing the adjacency matrix (Equation 4.6) because reverse direction focuses on backtracking the directed graph using arrival probabilities which are represented by columns in $A^{(rev)}$. If we sum the flow rates in both directions at the Ψ step, we get flow rate summaries between the query list and each response group overall (in both directions) (Equation 4.10).

$$\begin{aligned} A^{(rev)}_{ij} &= C_{ij}/\sum C_{\cdot j} && (4.6) \\ M^{(rev)}_{N\times N} &= \sum_{s=1}^{w} A^{(rev)^{s}} && (4.7) \end{aligned}$$

$$\begin{aligned}\Psi^{(rev)}_{G\times N} &= \Upsilon'_{N\times G}M^{(rev)}_{N\times N} &(4.8)\\ \Theta^{(rev)}_{G\times 1} &= \Psi^{(rev)}_{G\times N}Q_{N\times 1} = \left(\Upsilon'_{N\times G}M^{(rev)}_{N\times N}\right)Q_{N\times 1} &(4.9)\\ \Theta^{(tot)}_{G\times 1} &= (Q'_{N\times 1}(\Psi_{N\times G} + (\Psi^{(rev)}_{G\times N})'))' &(4.10)\end{aligned}$$

Modeling and testing values in Θ, $\Theta^{(rev)}$, and $\Theta^{(tot)}$

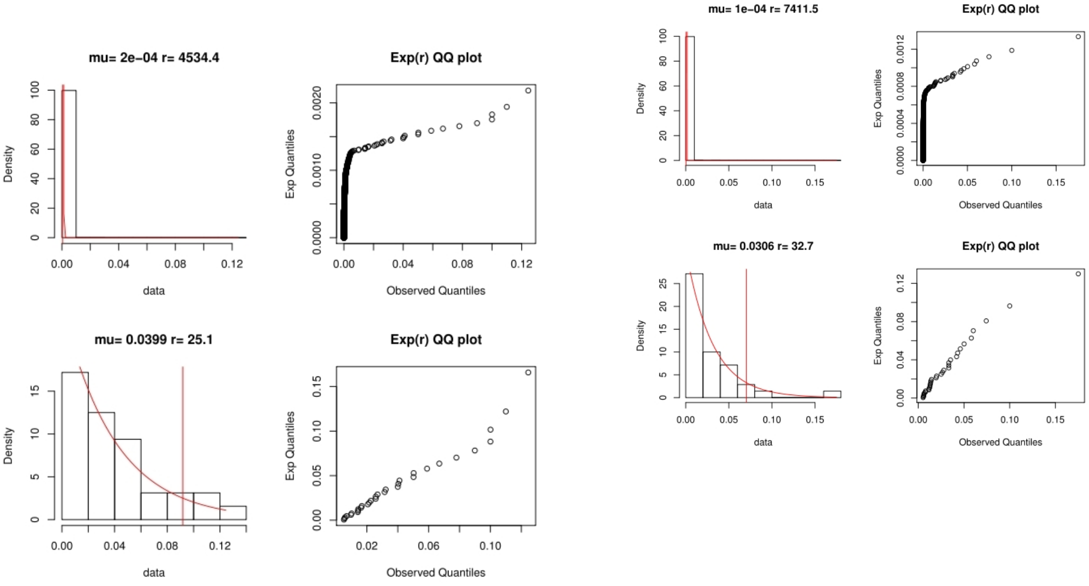
After obtaining metrics for random walk flow between the query list and each response group, we would like to discriminate which metrics are significantly high; these are the response groups which are highly connected to the query list in a specific direction. This might be accomplished by a statistical test of the null hypothesis that the observed random walk flow between a response group g and the query list is equal to that of a randomly selected query list and g. There are two complicating considerations for designing such a test.

1. *Response group size.* The number of nodes and edges in a reponse group is variable. Therefore, we must account for the assumption that larger response groups are more likely to have higher random walk flow with a query list than smaller response groups. Using mean flows instead of sums between the nodes in the query list and each response group may account for this, because it would penalize larger resopnse groups. However, means would also complicate the matrix operations we use to summarize the flows. Further, mean metrics are susceptible to outliers, which could bias our model.

2. *Response group connectedness.* In addition to size, response groups have different general connectivity with the rest of the pathway network due to its inherent structure. This can also cause bias in flow metrics where more connected response groups are more likely to have higher flows with a query list than smaller response groups. While there is likely a correlation between response group size and connectedness, it is not guarranteed, so we must account for all combinations of size and connectivity.

The underlying flow distribution

We begin with a bootstrapped assessment of the distribution of values in the M matrices (Equations 4.3 and 4.7). Most of these non-negative values are near zero, with a skewed upper tail containing those higher random walk flow relationships. A common probability distribution with these properties is the Exponential distribution, which is often used to model waiting times for an event to occur, such as the time until a light bulb will burn out. If we plot our bootstrapped sample from observed flows for random walks of $w = 10$ steps on the EcoCyc pathway network (Keseler et al., 2009), we see a good fit to an Exponential distribution (Figure 4.1) for values more distant from zero.



(a) M

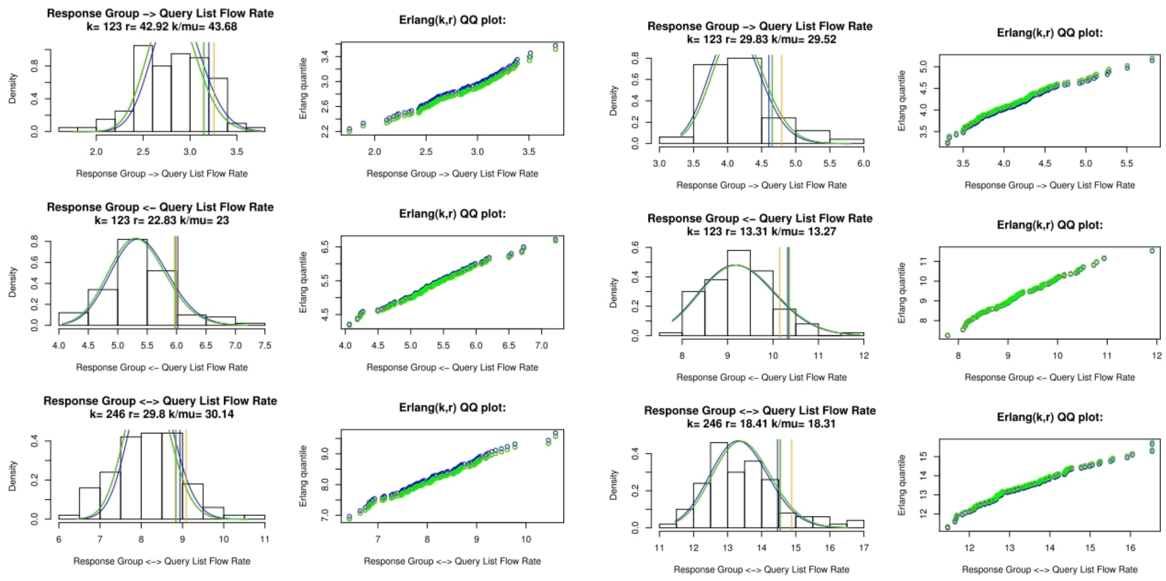
(b) $M^{(rev)}$

Figure 4.1 General assessment of fit to an Exponential distribution for values in M (a) and $M^{(rev)}$ (b) on a given pathway network. Each assessment includes a histogram with a fit Exponential density and Quantile-Quantile plot for all values in the matrix (top) and values greater than 0.005 (bottom) from 10-step random walk simulations on the EcoCyc pathway network.

The Exponential distribution has many useful properties such as memorylessness ($P_{Exp_\lambda}(X > t + \delta | X > t) = P_{Exp_\lambda}(X > \delta)$, where λ is the rate parameter, t is a length of waiting time, and $t + \delta$ is a longer waiting time). Another useful property is that the sum of k independent and identically distributed Exponential random variables with rate parameter λ follows what is called the Erlang distribution with shape parameter k and rate parameter λ. The Erlang distribution is a special case of the Gamma distribution where the shape parameter is an integer. Since our matrix multiplication in Equations 4.5 and 4.9 actually sum M values for a given query list in each response group, we can assume that the values in the Θ vectors each follow a different Erlang distribution with same shape parameter equal to the size of the query list and different rate parameters (Equation 4.13). We can use the same model for the reverse direction (Equation 4.14) and total in both directions (Equation 4.15), where k is doubled because the values are summed twice- once for forward and once for the reverse direction.

$$\begin{aligned}
M_{ij} &\sim Exp(\lambda) \text{ ,where } \lambda \text{ is the inverse of the mean of all values in } M \qquad (4.11)\\
M_{ij}^{(rev)} &\sim Exp(\lambda^{(rev)}) \text{ ,where } \lambda^{(rev)} \text{ is the inverse of the mean of } M^{(rev)} \qquad (4.12)\\
\Theta_g &\sim Erlang(k, \lambda_g) \text{ ,where } k \text{ is the size of the query list} \qquad (4.13)\\
\Theta_g^{(rev)} &\sim Erlang(k, \lambda_g^{(rev)}) \qquad (4.14)\\
\Theta_g^{(tot)} &\sim Erlang(2k, \lambda_g^{(tot)}) \qquad (4.15)
\end{aligned}$$

Assessing the Erlang-based model involves a Monte Carlo simulation where, for a given query list size, k, we repeatedly draw a random query list of k entities out of the pathway network and compute Θ, $\Theta^{(rev)}$, and $\Theta^{(tot)}$ each draw, building a multivariate (in the number of response groups) sampling distribution for each Θ. For a given reponse group, we then fit an Erlang distribution to the simulated results using the convenient Erlang distribution property that the rate parameter equals the ratio of the shape parameter to the mean. With this ternary relationship, we can estimate the rate parameter and by taking the ratio of the shape parameter (k) to the mean of the Monte Carlo simulation. As with the Exponential distribution above, for a given query list size and response group, we can then inspect fit by plotting the histogram of the Monte Carlo values with the density of the fit Erlang distribution as well as creating a Quantile-Quantile plot for each of Θ, $\Theta^{(rev)}$, and $\Theta^{(tot)}$ (Figure 4.2).



(a) tRNA charging pathway (106 nodes)

(b) putrescine degradation II (30 nodes)

Figure 4.2 Erlang assessments for arbitrarily selected pathway response groups after random walk simulations of $w = 10$ steps on the EcoCyc pathway network and 100 Monte Carlo simulations of flow rates with a query list of size $k = 123$. The three rows are assessments of the Θ_g, $\Theta_g^{(rev)}$, and $\Theta_g^{(tot)}$ values, respectively, where g is the tRNA charging pathway (a) and putrescine degradation II (b).

Hypothesis Testing

After establishing an Erlang-based model for our test statistics, Θ, $\Theta^{(rev)}$, and $\Theta^{(tot)}$, we can define a null hypothesis to test for each Θ vector and each response group. We stated earlier that the goal is to test the case where there is no flow relationship between a response group g and the query list, so our null hypothesis, H_o, is that the unknown true rate parameter, λ_g^*, equals the Monte Carlo-estimated λ_g, which can be interpreted as the rate parameter for flows between unrelated query lists and response groups (Equation 4.16). We make the analogous null hypotheses for the other two Θ vectors (Equations 4.17 and 4.18).

$$
\begin{aligned}
H_0 &: \lambda_g^* = \lambda_g && (4.16)\\
H_0^{(rev)} &: \lambda_g^{*(rev)} = \lambda_g^{(rev)} && (4.17)\\
H_0^{(tot)} &: \lambda_g^{*(tot)} = \lambda_g^{(tot)} && (4.18)\\
& && (4.19)
\end{aligned}
$$

The alternative hypothesis should reflect a high flow rate between the query list and the response group g; a random query list drawn from the set of nodes which are biologically linked to response group g would follow an Erlang distribution with the same shape parameter, k, but a larger rate parameter, λ_g. Therefore, the counter hypotheses are the upper-tailed alternatives in Equations 4.20, 4.21, and 4.22.

$$
\begin{aligned}
H_A &: \lambda_g^* > \lambda_g && (4.20)\\
H_A^{(rev)} &: \lambda_g^{*(rev)} > \lambda_g^{(rev)} && (4.21)\\
H_0^{(tot)} &: \lambda_g^{*(tot)} > \lambda_g^{(tot)} && (4.22)
\end{aligned}
$$

And we reject H_0 if the observed Θ_g falls above the $(1-\alpha)$ percentile of the Erlang distribution with shape k and rate λ_g, where α is a selected Type I Error Rate, which is the rate at which the null hypothesis is rejected incorrectly (Equations 4.23, 4.24, 4.25).

$$
\begin{aligned}
\text{Reject } H_0 \text{ if} \quad & P_{Erl_{k,\lambda_g}}(X > \Theta_g) \le \alpha && (4.23)\\
\text{Reject } H_0^{(rev)} \text{ if} \quad & P_{Erl_{k,\lambda_g^{(rev)}}}(X > \Theta_g^{(rev)}) \le \alpha && (4.24)\\
\text{Reject } H_0^{(tot)} \text{ if} \quad & P_{Erl_{k,\lambda_g^{(tot)}}}(X > \Theta_g^{(tot)} \le \alpha && (4.25)
\end{aligned}
$$

Multiple testing correction

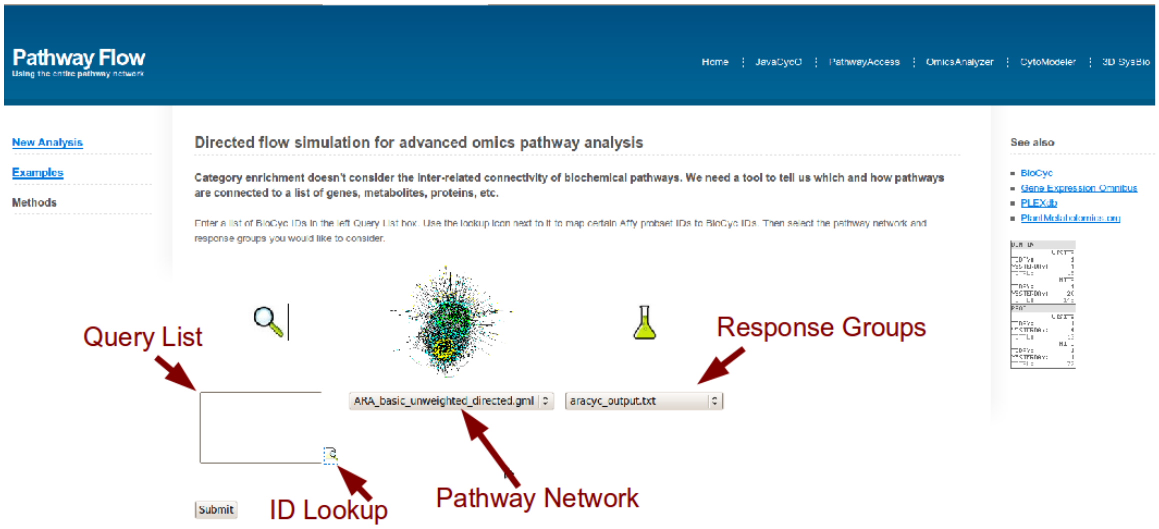
When multiple hypothesis tests are conducted simultaneaously, the Family Wise Error Rate (FWER) is inflated by the number of tests; If we conduct 10 tests, each with $\alpha = 0.01$, each test has a 1% probability of making a Type I Error, but the overall probability of making a Type I Error is the sum of each α, or $0.01 \times 10 = 0.1$. The multiple testing problem has been a focus for microarray processing because families of tests are conducted on thousands of genes in this field creating strong demand for clever correction methods. The most straight-forward and conservative correction, named for Bonferronni Holm (1979), simply uses a corrected α value for tests equal to the original desired Type I Error Rate divided by the number of tests, $\alpha' = \frac{\alpha}{m}$, where m is the number of tests. Several more complex methods exist which focus on the False Discovery Rate and estimate parameters for the specific distribution that p-values follow for microarray experiments (Storey, 2003; Storey and Tibshirani, 2003; Storey et al., 2004; Fodor et al., 2007), where p-values are uniformly distributed between zero and one with a spike near zero containing the relatively large set of genes perturbed by the experiment. Unfortunately, p-values for response groups are not always expected to follow such a distribution because there are often only a few significant response groups in one of our analyses. For this reason, we discretionarily use Bonferronni correction to correct for multiple testing where an independent test is conducted for each response group.

Applications Using Our Web Tool

In order to validate our method on real data and present our web tool, we will walk through a use case where we use the web tool (Figure 4.3) to discriminate response groups from the EcoCyc pathway network.

Use Case: LipidA inhibition in *E. coli*

The data for this use case comes from the GEO (Barrett et al., 2009) dataset accession GDS3597 by Zhu et al. (2009), who investigated transcriptional regulation by FabR of the fatty acid biosynthesis genes fabA and fabB in the presence of endogenous and/or exogenous unsaturated fatty acids. Among other factors in their experiment, gene expression was measured in a control and treatment with CHIR-090, an antibiotic which inhibits the biosynthesis of Lipid A (Figure 4.4, Barb et al. (2007)). Lipid A is the anchor by which lipopolysaccharides attach



Directed flow simulation for advanced omics pathway analysis

Category enrichment doesn't consider the inter-related connectivity of biochemical pathways. We need a tool to tell us which and how pathways are connected to a list of genes, metabolites, proteins, etc.

Enter a list of BioCyc IDs in the left Query List box. Use the lookup icon next to it to map certain Affy probset IDs to BioCyc IDs. Then select the pathway network and response groups you would like to consider.

Figure 4.3 The main page for beginning an analyses with our web tool. It takes three simple inputs: 1) a list of BioCyc IDs which can be looked up using our mapping service (Figure C.1), 2) the pathway network, and 3) the response groups to discriminate.

to the outer membrane of gram-negative bacteria, which provide much of the cell's structural stability and are also recognized by immune systems.

The query lists were generated using GEO's T-test data analysis tool. Single-tailed tests at the 90% confidence level for "control < treatment" and "treatment > control" created a query list for up-regulated genes and down-regulated genes, respectively. These lists are not actually genes, but probeset identification numbers which do not exist in our reference pathway network, EcoCyc. Fortunately, our web tool includes a mapping service for the Affymetrix probeset IDs on the platform used by GEO dataset GDS3597, which takes as input the list of probeset IDs and presents us with the corresponding EcoCyc IDs, which our web tool can process (ID Lookup on Figure 4.3 and Figure C.1).

Up-regulated genes

123 EcoCyc genes were identified from the list of probesets switched higher when lipid A synthesis was inhibited. We can then discriminate each of two sets of response groups

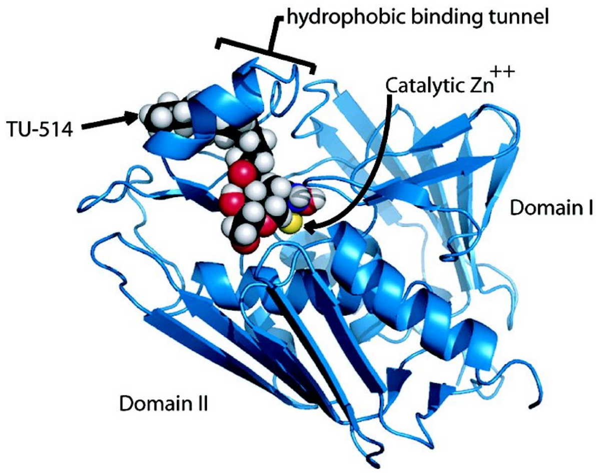
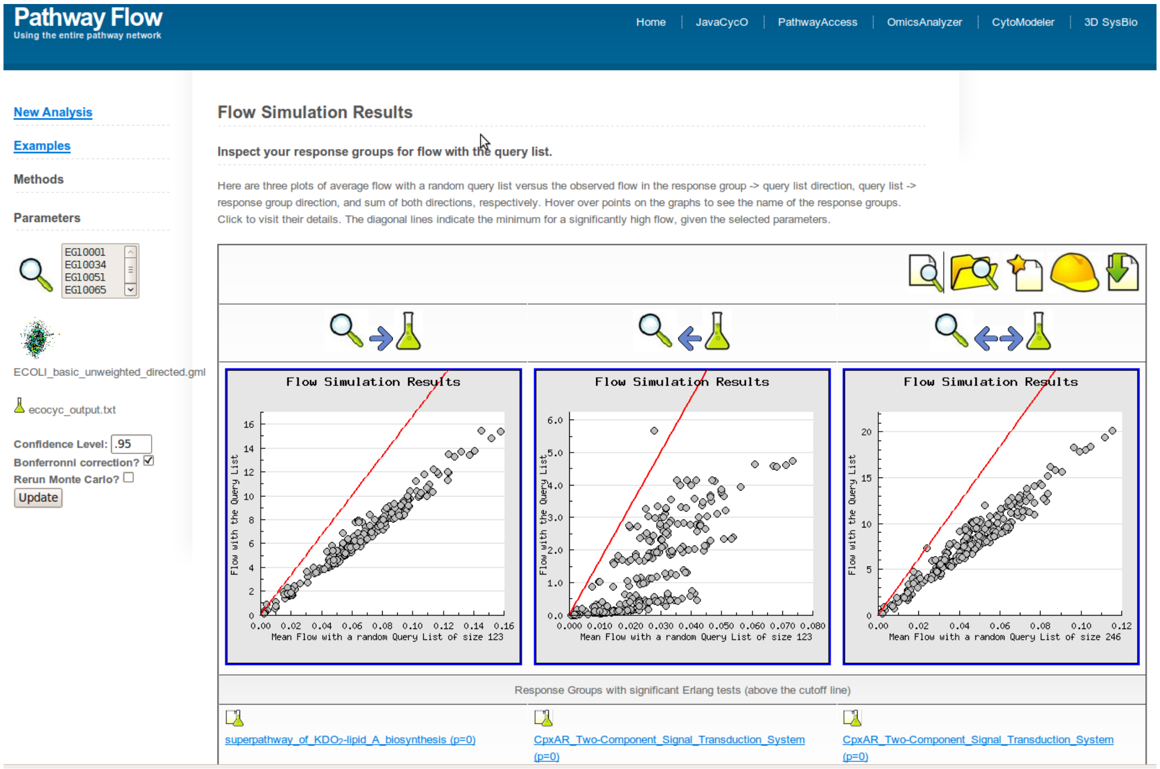


Figure 4.4 CHIR-090 binds the LpxC enzyme, preventing it from catalyzing the committed step in Lipid A biosynthsis. Image from Barb et al. (2007).

(Response Groups on Figure 4.3).

Pathway Response groups were discriminated first. When we click Submit, the web tool displays our current parameters on the left with three plots: one for each of the forward direction, reverse direction, and total, respectively (Figure 4.5). The default is to use Bonferronni correction at the 95% confidence level and the red cutoff lines are drawn accordingly. Response groups that fall above the red lines have significantly high flow with the query list and are listed below with *p*-values. We can also hover over response groups in the plots to see their names and *p*-values. There is also a set of icons and buttons to help navigate the web tool (Table C.1).

The superpathway of KDO$_2$-lipid A biosynthesis is the only pathway that is a significant successor (downstream in the directed pathway network) to our query list of up-regulated genes, with a *p*-value less than 0.0001. This is the expected result when the cells are unable to produce the lipid A they require for membrane structure; they are increasing their efforts to produce more lipid A. The CpxAR Two-Component Signal Transduction System is the only significant predecessor (upstream) pathway to our query list of up-regulated genes, with a *p*-value less than 0.0001. This is a signalling system which senses cell envelope stress (Wolfe



Pathway Flow
Using the entire pathway network

Home | JavaCycO | PathwayAccess | OmicsAnalyzer | CytoModeler | 3D SysBio

New Analysis

Examples

Methods

Parameters

EG10001
EG10034
EG10051
EG10065

ECOLI_basic_unweighted_directed.gml

ecocyc_output.txt

Confidence Level: .95
Bonferronni correction? ☑
Rerun Monte Carlo? □
Update

Flow Simulation Results

Inspect your response groups for flow with the query list.

Here are three plots of average flow with a random query list versus the observed flow in the response group -> query list direction, query list -> response group direction, and sum of both directions, respectively. Hover over points on the graphs to see the name of the response groups. Click to visit their details. The diagonal lines indicate the minimum for a significantly high flow, given the selected parameters.

Response Groups with significant Erlang tests (above the cutoff line)

superpathway_of_KDO$_2$-lipid_A_biosynthesis (p=0)

CpxAR_Two-Component_Signal_Transduction_System (p=0)

CpxAR_Two-Component_Signal_Transduction_System (p=0)

Figure 4.5 Results are visualized with a plot of the response groups for each direction and the total. The Y-axes are Θ, $\Theta^{(rev)}$, and $\Theta^{(tot)}$, respectively. The X-axes are the inverses of λ, $\lambda^{(rev)}$, and $\lambda^{(tot)}$, respectively, which are also the expected values of the Θ's for the respective Monte Carlo simulations. The red lines mark the null hypothesis rejection cutoff, given the confidence level, correction, and λ value (X-coordinate).

et al., 2008), which is also expected because the we can interpret our results as evidence for CpxAR signalling the increased expression of the genes in our query list. The CpxAr system responds to cell envelope stress and regulates transcription of the porin genes ompF and ompC, and a loss of function mutation in cpxAr can result in increased transcription of ompC and decreased transcription of ompF (Batchelor et al., 2005).

Reaction response groups can give us a more precise idea of which events in the pathway network are related to a query list. We start a new analysis, enter the same query list, select the same EcoCyc network, but select a different set of response groups: the EcoCyc reactions. Each response group contains one reaction event node in the network plus all of the participants, both input and output. Now, results plots are not of pathways, but reactions in EcoCyc (Figure 4.6). Discrimination of reactions produce much longer lists of significant response groups. To get an idea of how many are significant in each direction, we click the *Download PDF* button to see the same plots along with a Venn Diagram of the counts of significant response groups (Figure 4.7).

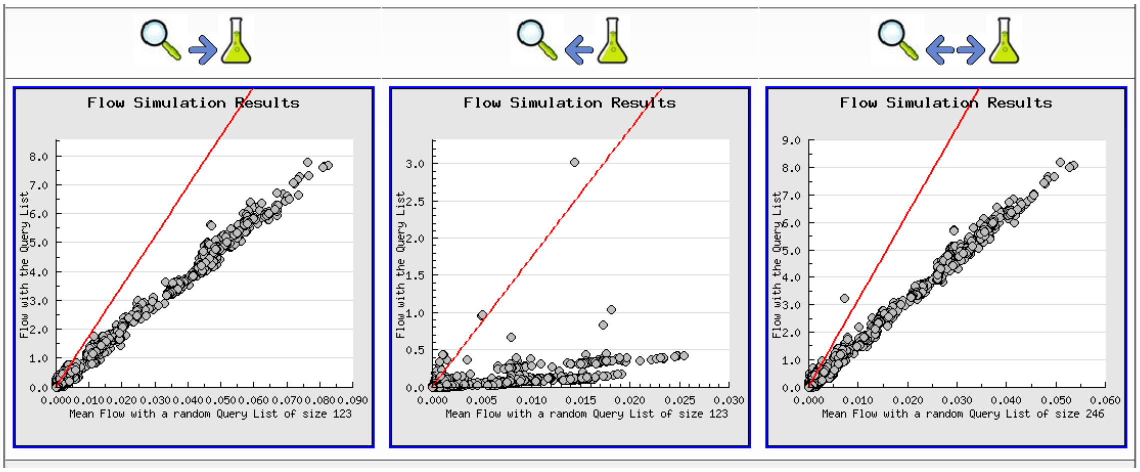


Figure 4.6 Points on these scatter plots represent EcoCyc reactions. Again, reactions above the red lines are significantly related to the query list containing up-regulated genes.

If we click the *Response Group IDs* button, we can download the BioCyc IDs of the significant successor (forward direction) reactions. We then used the JavaCycO software (Van Hemert and Dickerson, 2010b) and Cytoscape (Shannon et al., 2003) to visualize the significant reactions within their integrated network of respective pathways and highlight the members of the query list and the significant reactions (Figure 4.8). Blue-marked reactions are described in Table C with descriptions taken directly from EcoCyc. These reactions include several phospholipid-building reactions, which is consistent with our conclusion from the pathway response groups that if lipid A synthesis is inhibited, cells invest in compensating for its depletion. We also see in the list the Arabinose-5-phosphate isomerase reaction, which produces the first precursor to keto-deoxyoctulosonate (”KDO”). KDO is an antigen that is anchored to the outer membrane by lipid A (Figure 4.9) (Raetz et al., 2006). Since the genes in our query list are up-regulated, cells not only respond to lipid A inhibition by attempting to produce

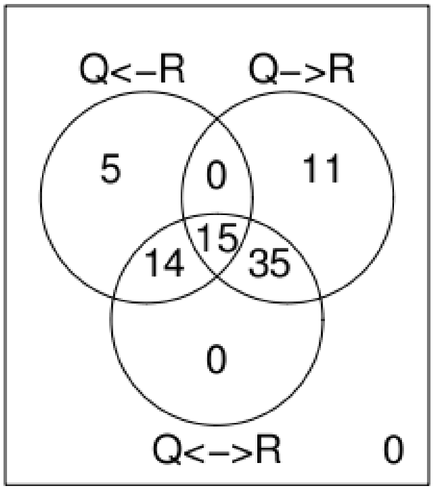


Figure 4.7 The circle labeled "Q< −R" is the set of significant response groups in the reverse direction. The circle labeled "Q− >R" is the set of significant response groups in the forward direction. The circle labeled "Q< − >R" is the set of significant response groups in the tests for flow in both directions.

more lipid A (the anchor), but they also respond by attempting to produce more KDO. A hypothesis might be that cells use the same sensing mechanism to determine their amount of functioning lipid A and KDO. Further, the OmpR phosporylation reaction is significant. OmpR is phosphorylated by EnvZ when osmotic pressure drops in the cell disrupting homeostasis. Phosphyrlated OmpR binds promoters for the ompF and ompC genes which code for the OmpF and OmpC porins (Batchelor et al., 2005). We could hypothesize that the inhibition of lipidA disrupts osmotic homeostasis and the cell responds by attempting to produce more pressure-relieving porins.

We can conduct the same analysis for the significant predecessors (reverse direction) reactions to our query list containing up-regulated genes. The list or significant reactions is shorter and the list of significant reactions that are part of pathways is shorter still (Figure 4.10). The significant reactions in the pathways are listed in Table C. This list is made up of several reactions which phosphorylate nitrate and nitrite sensing response proteins and others which activate ArcAB, which has recently been found to not only regulate general anaerobic growth, but also plays a role in resistance to reactive oxygen compounds (Loui et al., 2009).

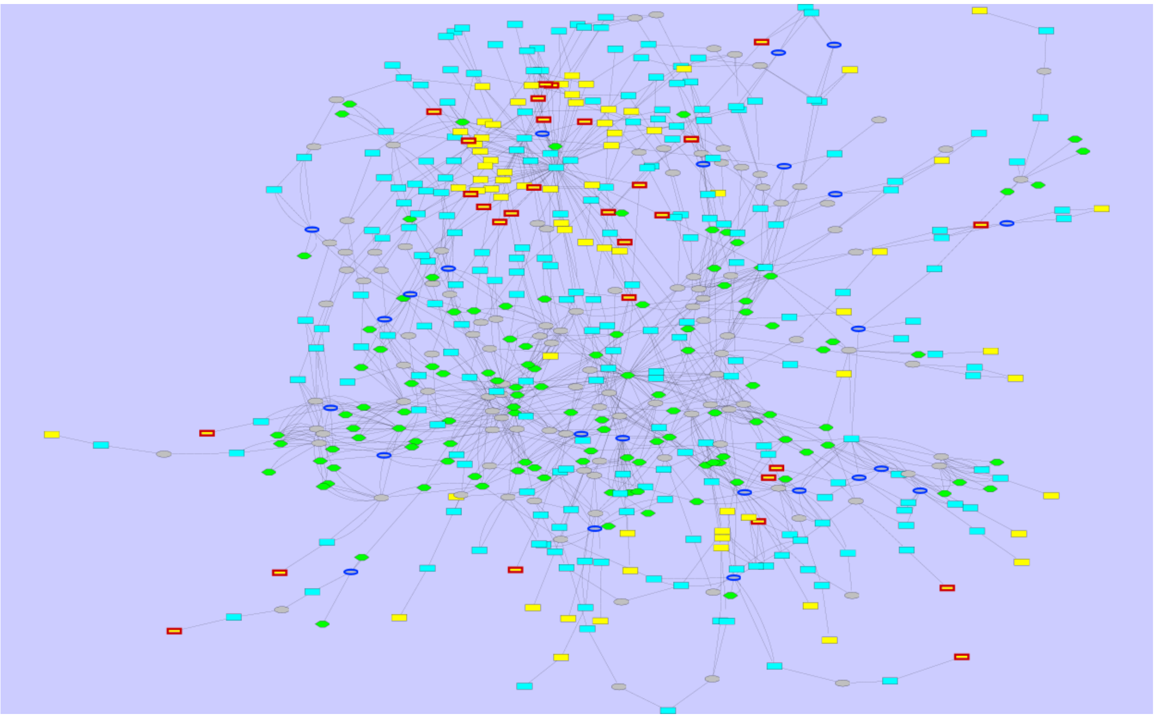


Figure 4.8 The integrated pathways containing significant successor reactions. Significant successor reactions are marked blue while the members of the query list which happen to also be in this subset of pathways are marked red. Node types are color-codes as follows: Yellow=Gene, Turquoise=Proteins, Green=Metabolites, Grey=Reactions.

Down-regulated genes

81 EcoCyc genes were identified from the list of probesets switched to lower expression when lipid A synthesis was inhibited.

Pathway response groups are plotted in Figure 4.11. When conservatively using Bonferronni correction, two pathways were significant, but some pathways seemed to be plotted very near the significance cutoff line. We adjusted the confidence level to 99% and unchecked the Bonferronni option to be slightly less conservative with our Erlang tests, which resulted in a total of four significant pathways instead of two. The TorSR and ZraSR Two-Component Signal Transduction Systems are the significant successor (forward direction) pathways to the query list. The TorSR system regulates use of Trimethylamine N-oxide (TMAO), which is both an

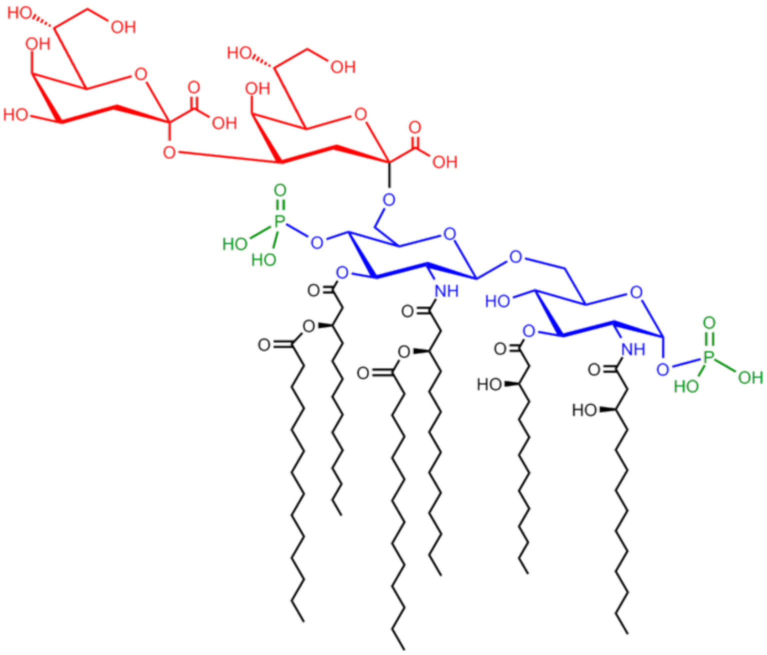


Figure 4.9 The saccharolipid Kdo2-Lipid A. Glucosamines are blue, KDO is red, acyl chains are black and phosphate groups in green. (Raetz et al., 2006)

osmoprotectant and alternative electron acceptor during anaerobic respiration (Ansaldi et al., 2000). The ZraSR system senses toxic levels of zinc and lead in the periplasm. The CpxAr Two-Component Signal Transduction System and Acetoacetate Degradation to Acetyl CoA pathways are the siginificant predecessors (reverse direction) to the query list. Recall that the CpxAr system also appeared in the signifcant predecessor pathways of the up-regulated query list, a contradiction that might be explained by incorrect selection of confidence intervals for the query list generation and/or the Erlang tests. In this case, the CpxAr system has an extremely low *p*-value, so if we adjust the confidence level for the Erlang test, it will not drop out of either the up-regulated pathways or the down-regulated pathways. If we adjust the confidence level for the T-tests used to generate the up- and down-regulated gene lists, we can check whether the Erlang test results change. After entering query lists based on T-test at the 95% confidence level, results for the up-regulated pathways in both forward and reverse directions as well as down-regulated successors (forward direction) remained constant, while down-regulated

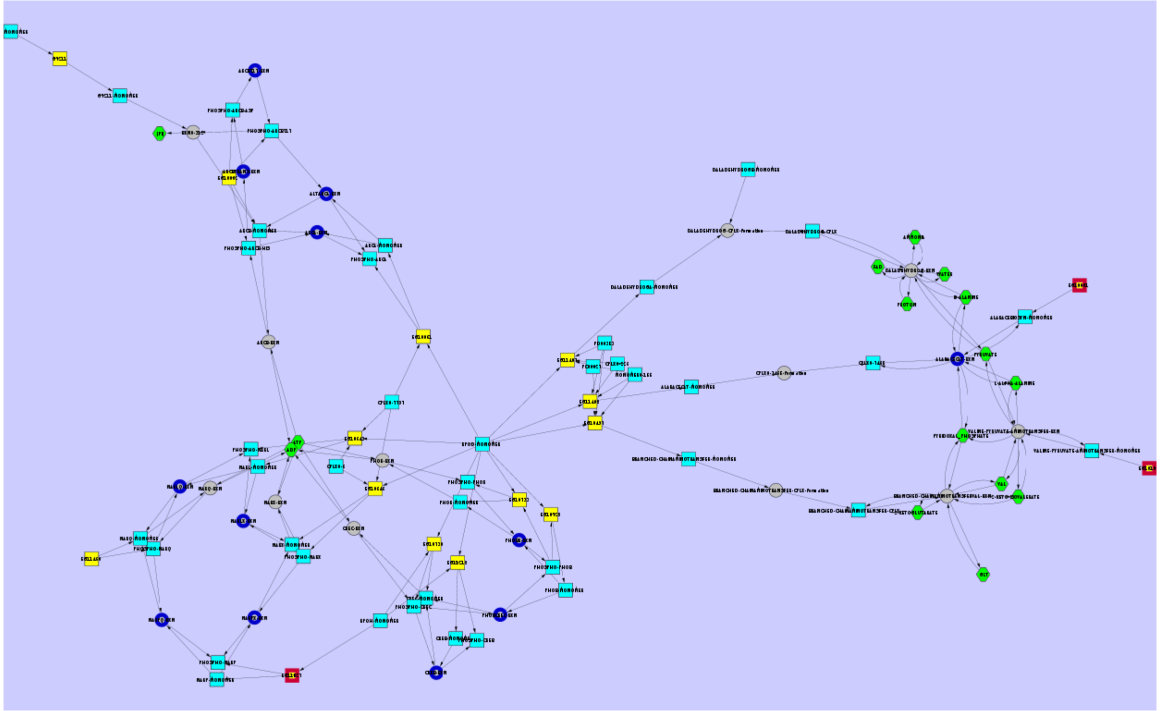
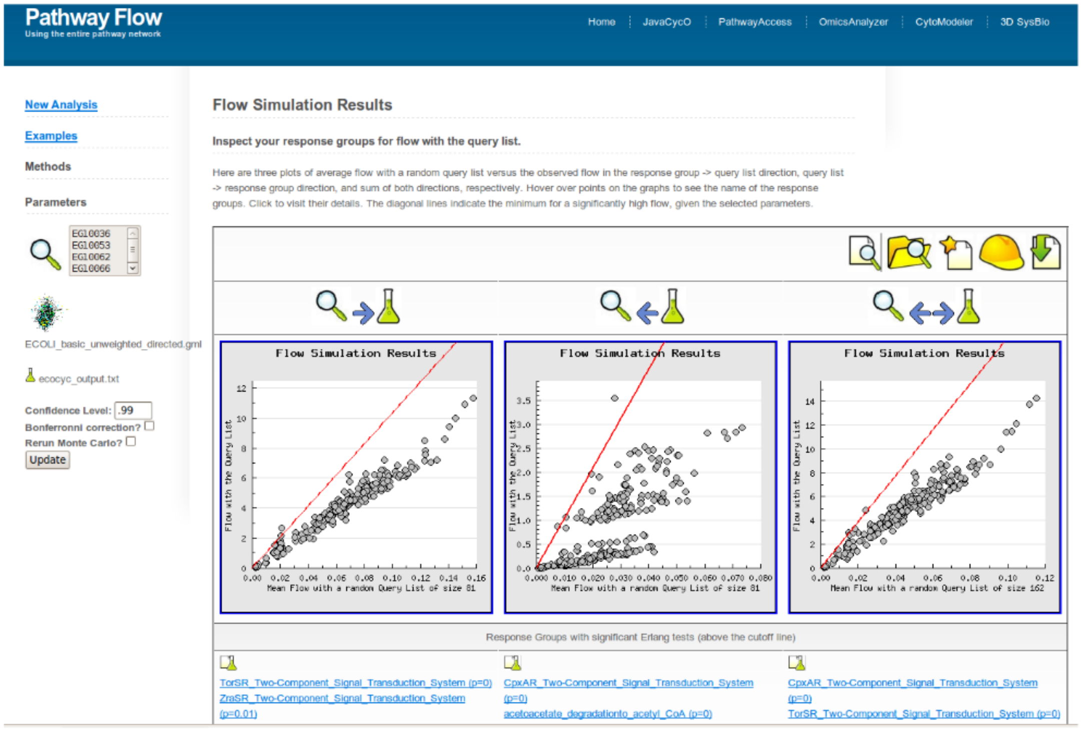


Figure 4.10 The integrated pathways containing significant predecessor reactions. Significant predecessor reactions are marked blue while the members of the query list which happen to also be in this subset of pathways are marked red. Node types are color-codes as follows: Yellow=Gene, Turquoise=Proteins, Green=Metabolites, Grey=Reactions.

predecessors (reverse direction) changed from CpxAr to the DpiAB Two-Component Signal Transduction System, which regulates citrate fermentation genes. The DpiAB system is also known to interrupt chromosome duplication in the SOS response (Yamamoto et al., 2008). Acetoacetate degredation feeds carbon energy into the TCA cycle (Pauli and Overath, 1972) and genes for this production are negatively regulated by ArcA.

Generating hypotheses

After completing our web-based analysis of the two query lists of genes, we can hypothesize a model for *E. coli* decision making when lipid A is inhibited. We clearly saw activity relavent to the cell's boundary (envelope and periplasm), which is consistent with our understanding of the utility of lipid A. We can further use our results to postulate a model for the cell's priorities



Flow Simulation Results

Inspect your response groups for flow with the query list.

Here are three plots of average flow with a random query list versus the observed flow in the response group -> query list direction, query list -> response group direction, and sum of both directions, respectively. Hover over points on the graphs to see the name of the response groups. Click to visit their details. The diagonal lines indicate the minimum for a significantly high flow, given the selected parameters.

EG10036
EG10053
EG10062
EG10066

ECOLI_basic_unweighted_directed.gml

ecocyc_output.txt

Confidence Level: .99
Bonferronni correction? □
Rerun Monte Carlo? □

Response Groups with significant Erlang tests (above the cutoff line)

TorSR_Two-Component_Signal_Transduction_System (p=0) ZraSR_Two-Component_Signal_Transduction_System (p=0.01)	CpxAR_Two-Component_Signal_Transduction_System (p=0) acetoacetate_degradationto_acetyl_CoA (p=0)	CpxAR_Two-Component_Signal_Transduction_System (p=0) TorSR_Two-Component_Signal_Transduction_System (p=0)

Figure 4.11 The same plots from our web tool as Figure 4.5, but the query list here contains the down-regulated genes.

when it is under this type of stress using Table 4.1 to organize interpretation of the response groups. Our model is as follows (entries in Table 4.1 are in "()"): Lipid A inhibition causes a breakdown of the cell's structure and osmotic stress, which the cell senses and responds with several different decisions. Firstly, it activates the genes to produce both the inhibited lipid A (1) and the KDO (2) that the lipid A should be anchoring to the cell membrane. It also shifts priorities away from growth (6,9), toxin sensing in the periplasm (5,9), and osmoprotectant production (7). OmpR activation is increased because both the OmpC and OmpF porins production require it (3), but since the promoter for ompf has higher affinity for OmpR-P than the promoter for ompc, ompF transctription is specifically decreased using a separate mechanism (4) so that only OmpC porins are produced (Figure 4.12). Most of these inferences are consistent with the literature, and we can hypothesize that the cell knows that the osmotic stress is caused by structural insufficiencies and not by a severe change in solute concentrations, so it chooses *not* to produce osmoprotectant.

Table 4.1 Interpretation of different flow simulations and tests.

	Successors (forward)	Predecessors (reverse)
Up-reg	**Activated by the query list** 1. KDO_2-lipid A biosynthesis 2. Arabinose-5-phosphate isomerase 3. OmpR phosporylation	**Activate the query list** 4. CpxAR signalling 5. nitrate and nitrite sensors 6. ArcAB
Down-reg	**De-activated by the query list** 7. TorSR signalling 8. ZraSR signalling	**De-activate the query list** 9. DpiAB signalling

Discussion

The main weakness of our method is sensitivity to missing information from the pathway network; our method does not directly infer new pathway models. Rather, it presents existing, complex knowledge about pathways in the context of a list of entities to generate hypotheses. If an entity in a query list is not understood, the best we can do with our method is assume ”guilt by association” and infer its involvement in the response groups we associate with the well-understood entities in the query list. This is especially true for query lists made entirely of genes because genes are usually leaves on branches of the pathway network with flow only from the gene to the rest of the network via translation to enzymes; these cases cannot produce results for the reverse direction because there are no flows into the query list. Reverse flow results are only possible when the pathway network contains an adequate amount of gene-regulatory relationships, which are represented by edges and flows into genes. Cycles and feedback loops might create ambiguity between significant successor and predecessor response groups.

We have developed a method and tool which leverages organism-wide pathway models for interpreting Omics data and generating hypotheses. It accomplishes our original objectives:

1. Receive as input a biochemical pathway network structure, a Query List of entities, and

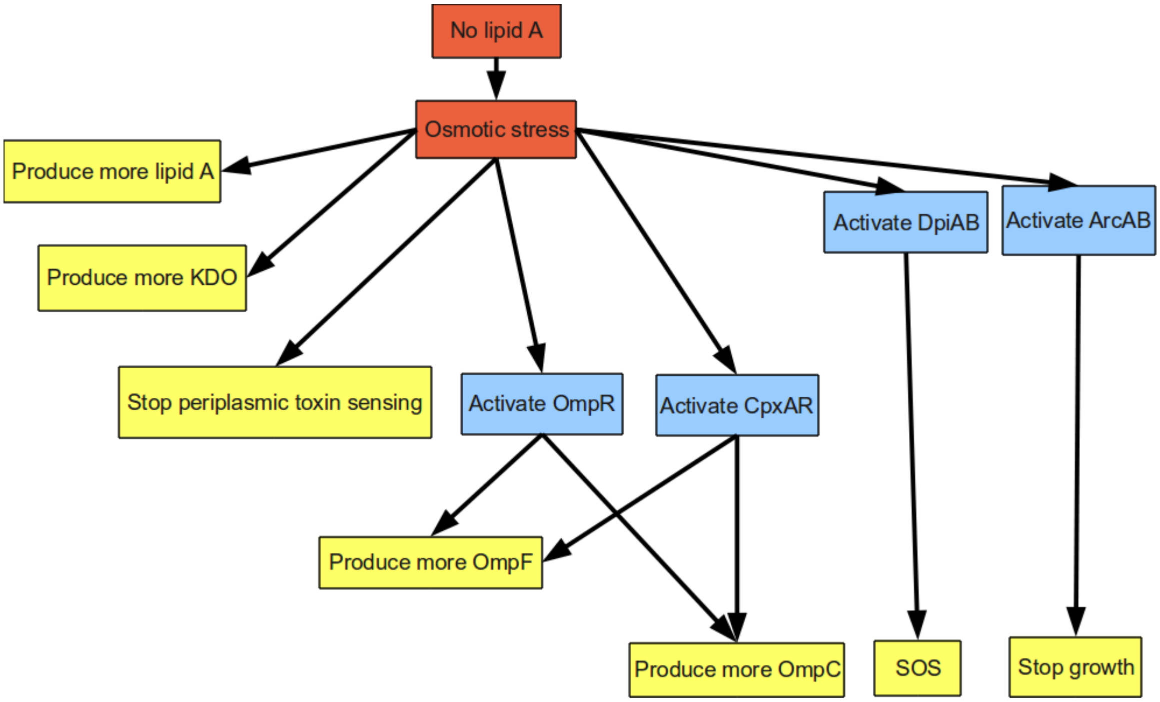


Figure 4.12 A model for *E. coli* responses to lipid A inhibition based on results from our web tool and confirmed by literature. Red boxes are the initial signals, blue boxes are the intermediate responses, and yellow boxes are the final responses.

a set of Response Groups to discriminate.

2. Visually and interactively present hypothesis test results for decision support and discretionary test parameter adjustments.

3. Entities in a Query List could be any combination of genes, enzymes, chemical compounds, or reaction events in the pathway network.

4. Response Groups can be the set of all functional pathways in the network, all reactions in the network, or the set of all compound classes in the network, for example.

5. Response Groups can overlap on entities.

6. The set of all Response Groups need not cover then entire pathway network.

7. The hypothesis test accounts for both Response Group size and inherent connectivity with the rest of the network.

We used our web tool to interpret Omics data from a simple *E. coli* microarray dataset, verified the results with the literature, and generated new hypotheses. Future work includes application to more diverse Omics datasets which include compounds and enzymes. Our tool is compatible with output from the Markov Clustering software (MCL) by van Dongen (2000) and we intend to investigate Response Groups defined by graphical clusters mined from large metabolice networks. Lastly, the web tool is to be fully integrated with the PLEXdb.org (Shen et al., 2005) website.

5. EXPRESSION PLATFORM INTEGRATION AND INSIGHTS INTO THE GRAPEVINE'S RESPONSE TO SHORT WINTER DAYS

A paper to be submitted to Plant Physiology

John L. Van Hemert [1], Erin E. Boggess[1,2], Alberto Ferrarini [2], Massimo Delledonne[4], and Mario Pezzotti[4], Anne Fennell [3], and Julie A. Dickerson[1]

Abstract

Besides being economically significant, The *Vitaceae* (the grape family) provides a unique domestication history as well as strong responses to environmental signals, such as winter dormancy. Mining gene expression data for biomarkers and pathway activity is a key component to understanding the mechanisms controlling such responses in plants. In this paper, we share our approach to pre-processing the data, exploratory analysis, additional data filtering, and clustering transcriptomics data. Pre-processing included a technical study aimed at comparing and integrated different expression platforms. Using results from the technical study, we were able to extract dormancy-related genes from an original set of 16436, and then form biologically meaningful clusters that supported interpretation of signaling activity and regulatory activity.

Our results provide insight into possible cellular mechanisms that occur at the onset of grape dormancy, by investigating known biological processes related to overrepresented annotation as well as examination of possible experimental variation within a treatment type over time are necessary, we proposed biological explanations for our computational results. These

[1] Electrical and Computer Engineering, Bioinformatics and Computational Biology, Iowa State University, Ames, Iowa

[2] Biotechnology, University of Verona, Italy

[3] Horticulture, South Dakota State University

explanations are the beginnings of new biological models for the underlying mechanisms during photoperiod-induced bud dormancy in *V. riparia*. We observed transcriptional responses to environmental signals as well as different groups of genes responding to one another.[4]

Introduction

The *Vitaceae* (grape) is comprised of a diverse collection of species that have been bred to grow in a variety or climate conditions. In addition to being an economically important crop, the grapevine represents a unique domestication history and sensitivity to environmental changes and signals. For example, many grape species undergo a period of endodormancy, or a non-growth phase, that typically corresponds to an inactive winter rest and is brought on by conditions within the plant itself. Endodormancy regulation of grape buds is necessary for plant survival during inclement winter conditions. The endodormancy phase is triggered by the onset of a shorter photoperiod corresponding to the shorter days of winter in the Northern hemisphere (Vergara and Perez, 2010; Kuhn et al., 2009; Perez et al., 2007; Noriega et al., 2007; Fennell et al., 2005).

V. riparia is a species of grape that is cultivated in North America which is known for its cold hardiness. The biological processes that *V. riparia* buds undergo at the onset of endodormancy are not well understood, mainly due to small amounts of relevant tissue. Our task in this project is to use multivariate methods to identify which genes, functional groups, and pathways participate in endodormancy activities by examining the plant transcriptome measured by Affymetrix Vitis Gene Chips (microarrays).

Multiple technologies exist that quantify the level at which genes are expressed. For the past decade or so, microarrays (considered high-throughput technology) have been the tool for that purpose. Of these, many different microarray platforms have been developed using a wide range of design parameters including but not limited to oligo-nucleotide length, microchip print technology, background fluorescence baselines, and oligo-nucleotide sequence selection.

[4] **Author contributions:** Van Hemert conducted and wrote the technical comparison portion with advice and discussions with Ferrarini, Delledonne, Pezzotti, Fennell, and Dickerson. Van Hemert and Boggess co-wrote much of the single-platform Affymetrix-based analysis, specifically the GO Over-representation analysis portions. Van Hemert developed the Δ-based MANOVA statistics and conducted the pathway flow analyses.

It is not uncommon to find several to thousands of experiments using different microarray platforms on the same organism or tissue (Barrett et al., 2009). Meta-analysis, or pooled reanalysis with increased statistical power, of such an organism requires some sort of integration of the datasets generated by each platform (Xiong et al., 2010).

In the past, microarray platform integration has been of interest to medical researchers because they sometimes need to increase power by merging different studies done on different platforms with different patients in order to answer similar questions. The first problem in platform integration is that there are usually genes uniquely measured by each platform. A common solution is to omit those genes and join the experiments on the common genes measured, which are mapped to microarray oligos according to sequence similarity to the latest gene models. Subsequent steps might integrate at different levels, with fluorescence intensities being the lowest level (Garrett-Mayer et al., 2008; Warnat et al., 2005; Shen et al., 2004; Parmigiani et al., 2002), and platform-specific statistics (Rhodes et al., 2002) or gene sets (Choi et al., 2003; Zenoni et al., 2010) being higher levels. The main problem is the trade-off between data level (lower is more precise) and confounded biological and platform effects, which are less prevalent at higher data levels. Approaches to circumvent this problem range from clustering all merged data to normalized observations against cluster averages (Shabalin et al., 2008), to fitting merged data to models which attempt to accounts for platform effects (Choi et al., 2003), to converting fluorescence intensities to platform-specific ranks or quantiles (Shen et al., 2004). Platform integration methods are usually evaluated by calculating accuracy and specificity rates when comparing results to common knowledge of a well-studied gene family such as estrogen receptors in breast cancer studies (Tsiliki et al., 2009), leaving the specific causes of errors to speculation; integration can be a form of benchmarking if the integration method provides a metric for how well platforms integrate. Model-based approaches such as Choi et al. (2003) and Xiong et al. (2010) are able to produce such metrics.

In more recent years, new methods appeared which measure gene expression using what are considered ultra-high-throughput technologies. Next-Generation Sequencing actually observes the nucleotide sequences of millions of segments from a sample which vary in length according to the specific technology. These reads are then assembled into longer segments using various bioinformatics tools and eventually create a set of replicons from the biological sample. Other tools are used to predict the locations and structure of gene models on the replicons. Deep sequencing and RNAseq refer to heavy sequencing of short reads of RNA samples followed by alignments of the reads to a pre-built genome. The depth to which reads overlap (called coverage) indicates the level of expression of a particular region in the genome. Coverage for a specified region has been quantified using the number of reads aligned to the region per the region length in kilobases per the total number of aligned reads, reads per kilobase of exon model per million mapped reads (RPKM) (Mortazavi et al., 2008).

Some platforms, like the new Nimblegen Vitis chips, were designed based on gene models predicted from the genome and their probe positions are known. Other platforms, like the Affymetrix 16k gene chip for Vitis, are based on older EST libraries and probes are placed on the new genome using sequence alignments. Even conservative sequence alignments can place probes incorrectly. Our other task is to use experimental data to identify Affymetrix probes incorrectly placed on the genome and show how data from two or more platforms can be integrated for more sensitive Next-Gen-like analysis of exon-level expression as well as refine functional annotation based on experimental data.

Results

Platform Integration for Exon Quantification

Alignment Filtering

We filtered Affymetrix probe alignments by fitting a mixed univariate normal model to the correlation values and classifying pairs as ”false” when in the lower population and there is low uncertainty ($< .01$). We can observe the filtered consensus alignments in GBrowse and omit them from functional annotation transferred to lists of differentially expressed Affymetrix probesets via gene models. With these data, we were able to filter 5,487 (2.75%) false Affymetrix probe placements on the genome coming from 3,150 consensus alignments (Figure 5.1).

When we fit the multivariate linear model to our real data, we obtain estimates for the true expression of individual exons. These estimates are called ”indirectly measured exons” and can be added to the set of exons which are ”directly measured” by single probes fully contained withing the exon. Depending on the genome coverage of the two expression platforms, we cannot obtain an estimate for every exon in the genome; we only have estimates for exons that are part of exon systems where the number of probes is at least as large as the number of exons. Therefore, platform integration can improve exon coverage by creating more such systems. In fact, the number of estimable exons for our integrated data set is larger than the sum of the numbers of estimable exons for each platform's data set fit separately (Figure 5.2).

Comparison to RNAseq data. To validate the exon expression estimations, we compared them to RNAseq data using a separate data set where technical replicates were run on both Nimblegen Vitis chips and transcript-sequenced using Illumina short-read sequencing. We also compared the standard gene-level summarization of expression units with RNAseq data (Figure 5.3. A small amount of correlation is lost when using our multivariate exon estimations, but an increase in noise is expected when the focus shifts from the larger gene summaries to more specific exon summaries (estimates) where there are one or more exon estimates for each gene estimate. This is a small tradeoff in cases where exon expression is the desired unit

Table 5.1 Transcription Factor Families

Family	**Odds Ratio**	***p*-value**
Cluster 2		
bZIP	9.3	0.00000
C3H	8.9	0.00000
CO-like	40.1	0.00000
Cluster 7		
CPP	52.7	0.00000
NAC	4.2	0.00000
ERF	3.0	0.00000
C2H2	2.9	0.00004
bHLH	2.3	0.00038

of expression, ie alternative splicing detection. Further, exon estimates correlate with RNAseq data as well as Affymetrix' Human tiling array in (Agarwal et al., 2010), whose plots look very similar to ours, including the lower-left region looking flat indicating zero correlation for expression below the noise level.

Single-Platform Analysis of the Full Photoperiod-induced Bud Dormancy Data

We mined 770 dormancy-related genes (see Figure 5.4 for examples) from the data and clustered them into 8 groups according to endodormant *V. riparia* expression patterns through time (Figure 5.5). Functional annotations and pathways associated with different clusters include cell wall restructuring, stress responses, and shift from energy use and growth to energy storage in starch. Two of the clusters (2 and 7) likely contain transcription factors (Figure 5.6). Each of these transcription factor-rich groups contain homologues to different transcription factor families. Cluster 2 is enriched with bZIP, C3H, and CO-like transcription factors (Guo et al., 2008). Cluster 7 is enriched with BHLH, C2H2, CPP, ERF, and NAC transcription factors. See Table 5.1 for these lists of families along with odds ratios (proportion in cluster to proportion in genome) and hypergeometric test *p*-values.

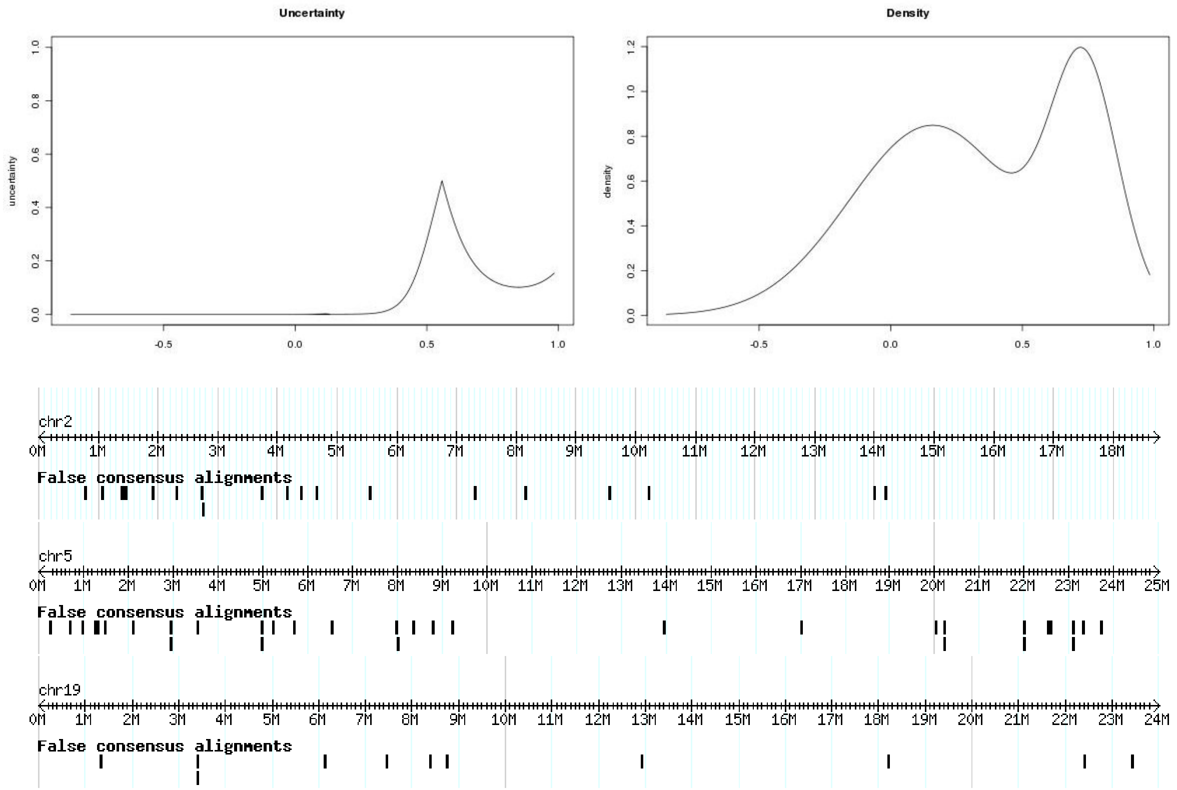


Figure 5.1 The univariate model fit to the distribution of linked probe correlations (above) and the positions of Affymetrix probes which fall into the lower population with low uncertainty (below). Top-right shows the univariate 2-Gaussian model fit to the top-right histogram in Figure C.4. The left mode is the population of poor, zero-center linked probe correlations and the right mode is the population of linked probes where the Affymetrix probe is correctly aligned. Top-left shows the plot of uncertainty when attempted to classify a linked probe as good or bad, given a correlation value (x-axis, same as top-right). We selected Affymetrix probes from linked probes in the lower population with low uncertainty for removal from chromosome alignments. Some examples are plotted in GBrowse in the lower three plots.

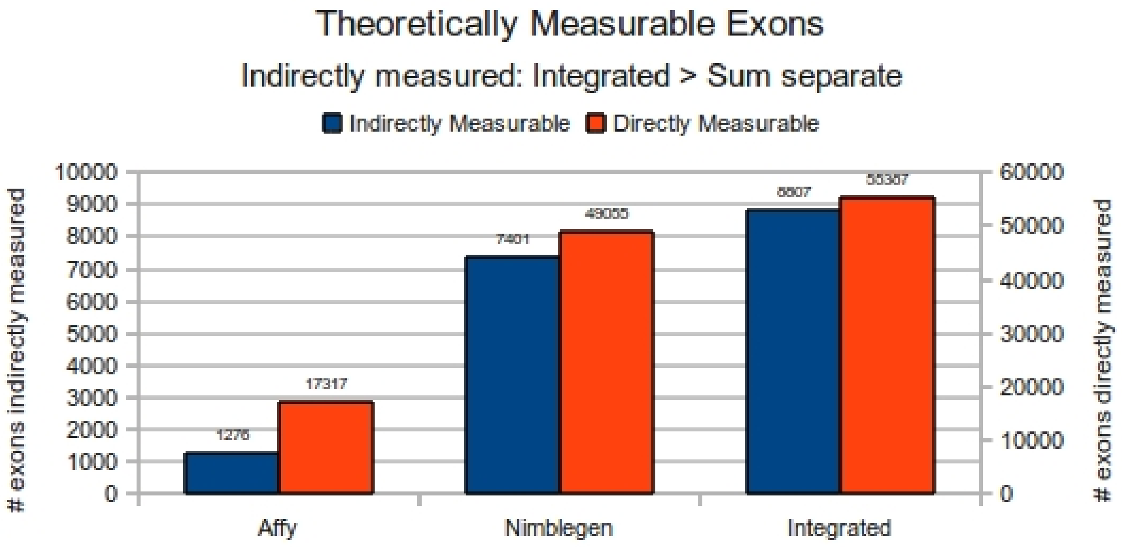


Figure 5.2 Estimability of exons improves when platforms are integrated. Red bars show the number of estimable grapevine exons by unspliced microarray probes (”Directly Measurable”) for each separate platform, Affymetrix (”Affy”) and Nimblegen, and after integration by pooling and normalization. Blue bars show the number of estimable exons by applying our probe-exon system model (”Indirectly Measurable”). These are the exons missed by standard exon summaries because they are only measured by spliced probes. Notice that the sum of two individual blue bars is less than the blue bar after integration.

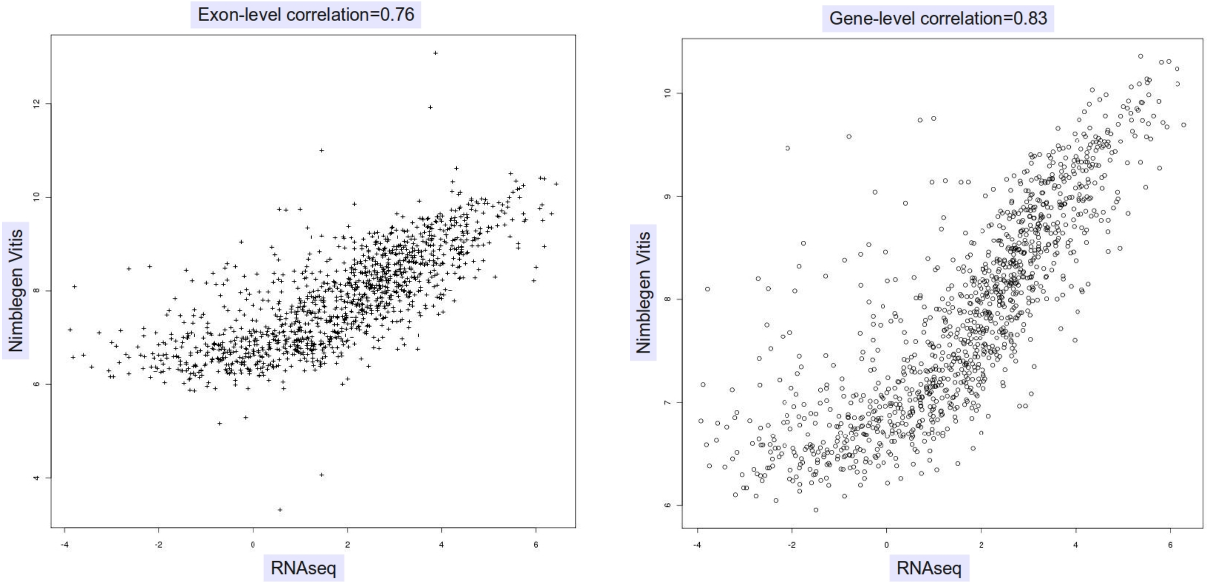


Figure 5.3 Exon estimates using our probe-exon system model compare well with RNAseq data. Points in the left plot are gene-level estimates and points in the right plot are exon-level estimates using our multivariate model. Microarray values were RMA normalized (including log-transformation) and RNAseq FPKM (Trapnell et al., 2010) values were log-transformed. We would expect a decline in correlation using our probe-exon system model due to a larger number of data points alone. Other causes may include non-linearity in splicing effects and cross-hybridization.

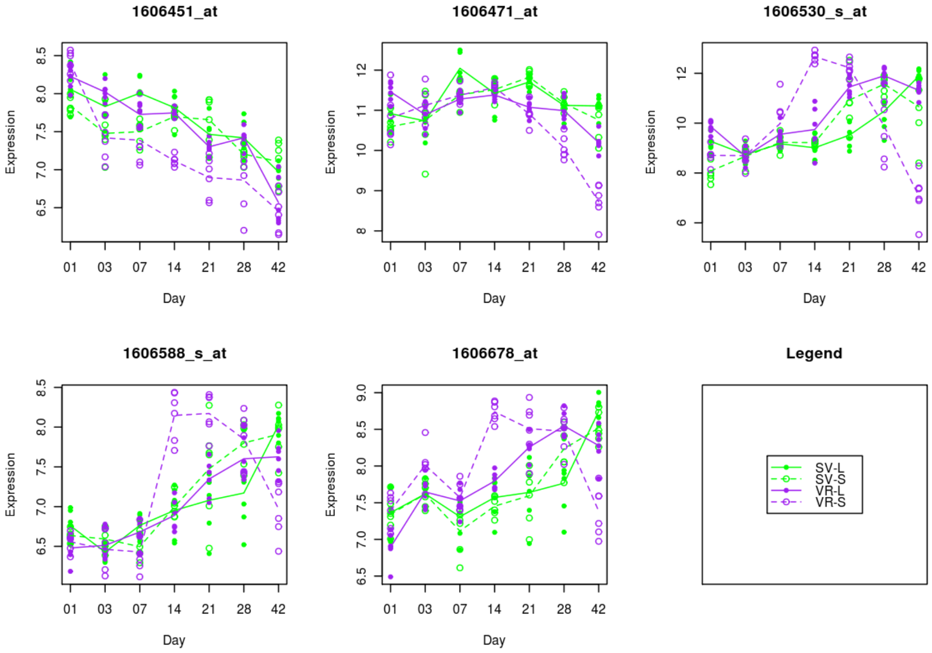


Figure 5.4 Five selected gene profiles from the 770 dormancy-related genes. Each probeset has four profiles– one for each treatment level combination and lines plot replicate means. We mined for probesets with *V. riparia* short-day (purple, dashed lines) expression unique for all other treatments. Notice how *V. riparia* behaves differently during short photoperiod in all of these representative examples.

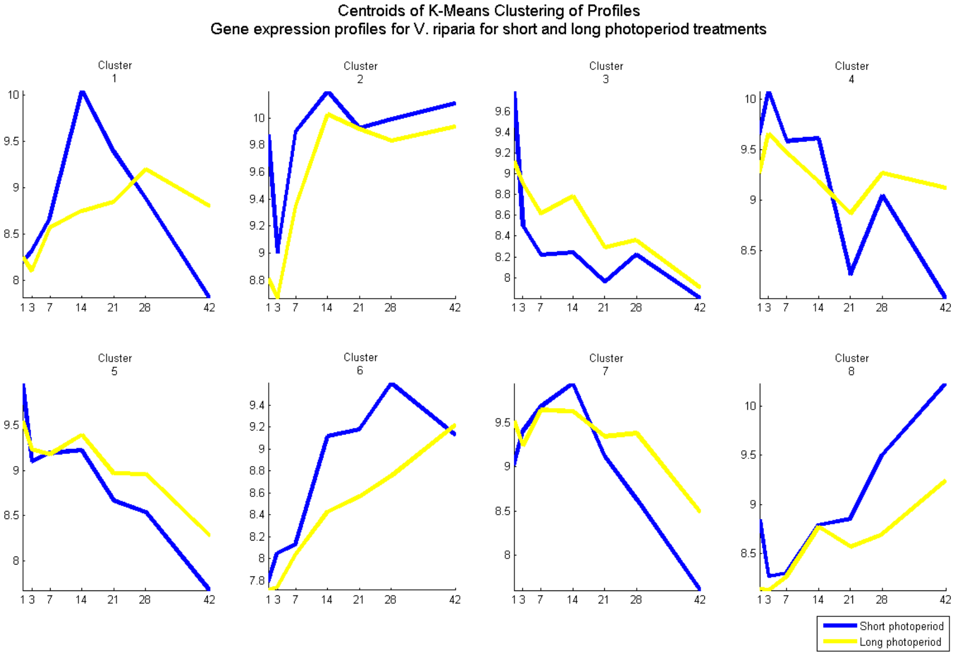


Figure 5.5 Centroids of gene expression profiles over the seven time points for the 770 gene data set. Mean expression values were calculated from the values in the original clusters (which themselves were median expression values over replicates) for each time point for *V. riparia* under the short photoperiod treatment (blue). This calculation was repeated with the corresponding *V. riparia* long photoperiod data (yellow). Negative controls had 50% and 75% quantiles of 6.04 and 6.21 respectively. No centroids exhibited expression levels in this range, meaning these genes have non-zero expression levels. Of note are Cluster 1, which contains cell wall restructuring genes, Clusters 2 and 7, which probably contain transcription factors, and Clusters 5 and 8, which contain energy use and storage genes, respectively.

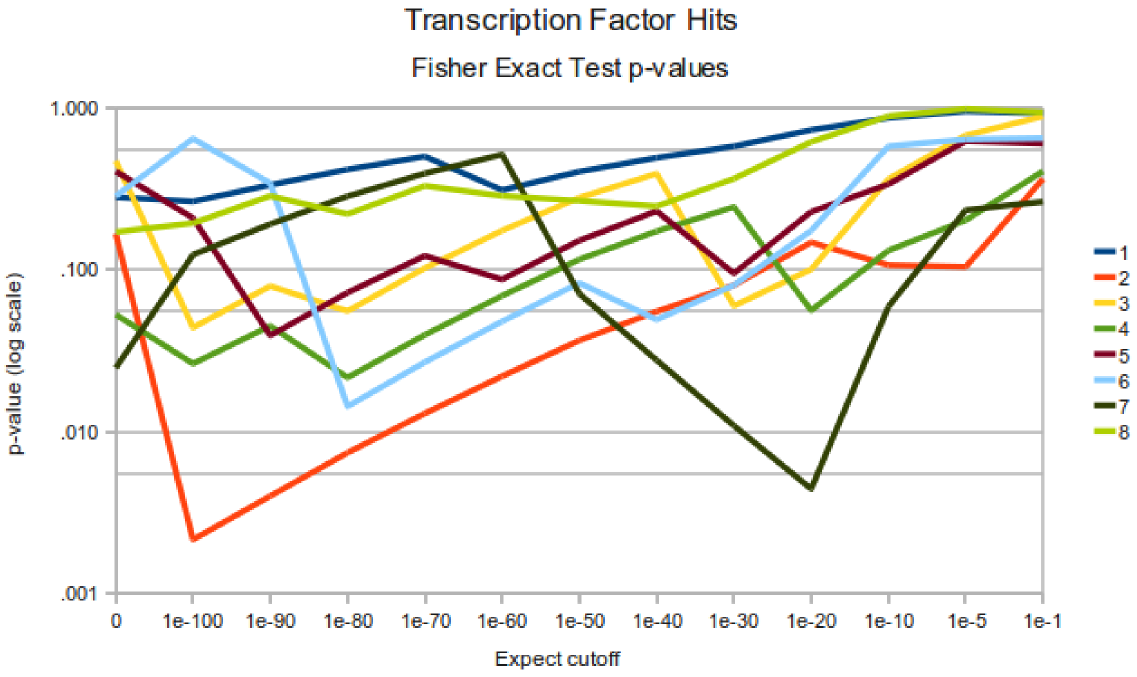


Figure 5.6 Each cluster was compared to a set of reference transcription factors from PlantTFDB (Guo et al., 2008) made up of transcription factors from *Vitis vinifera*, *Populus trichocarpa*, *Glycine max*, *Oryza sativa*, and *Arabidopsis thaliana* using BLASTP. Each series is one of the eight clusters and the x-axis is the expect-value cutoff for hits as it is made less conservative. The (log-scaled) y-axis shows Fisher's Exact Test p-values for the null hypothesis that the proportion of BLASTP hits in the cluster equals that of the entire genome. Clusters 2 and 7 reach the most significant p-values meaning they may represent transcriptional regulation patterns.

Discussion

Platform Integration

Microarray platform integration can improve exon coverage using our probe-exon system model. This can be useful if data exist for the same treatments on different platforms and either the goal is to integrate and summarize or the goal is to compare the platforms at the exon level. It also provides a model which takes advantage of spliced microarray probes, conferring greater flexibility in microarray design; spliced probes become an asset instead of a problem to be avoided. The probe-exon system model also accommodates overlapping gene models as well as cross-hybrization events between probes by accounting for the distribution of sample oligonucleotide fragments across multiple probes. Future work includes investigating non-linear cross-hybridization relationships between probes and exons for even more flexibility in microarray design.

Dormancy-related Genes

In this study, we successfully applied and adapted multivariate methods to effectively reduce a high-dimensional data set to a collection of genes that demonstrate dormancy-related activity. We then applied clustering methods to classify genes of interest into groups that exhibit similar expression profiles over time. For each gene cluster, we searched for over-representation of biological annotations, discriminated highly connected VitiCyc pathways, and compared them to a reference set of transcription factors in order to form new hypotheses about cell response to dormancy induction.

***V. riparia* genes in Cluster 1** show a spike in expression after about two weeks of short-photoperiod, while expression is relatively constant during the longer photoperiod. After about one month of short photoperiod, these genes' expression drop well below their long photoperiod levels. Overrepresented annotations in this group include lipid transport, carbohydrate metabolic process (Biological Process); Hydrolase activity, polygalacturonase activity (Molecular Function), and integral to membrane (Cellular Component) (Table C.4). Plants experience variation in photoperiod during the growing season, and must decide whether the variation is due to growing season weather, or season change in Autumn. This cluster shows that a strong transcriptional response occurs after about two weeks as the decision is made that it is, in fact, time to enter dormancy. Pathway flow analysis also showed an initial spike then decline in both energy use and cell wall activity (Table C.5). The behavior of genes in Cluster 1 appears to be some sort of signalling response. However, since a relatively low count and percentage of Cluster 1 genes were found in our transcription factor reference (Figure 5.6), it probably does not involve direct transcription regulation of other genes.

***V. riparia* genes in Cluster 2** behave in a relatively parallel manner, with a strong drop in expression in the first few days, followed by a sharp increase for the rest of the experiment. At day 1, gene expression is much higher during short photoperiod than long. Since all plants were under the same photoperiod treatment (long photoperiod) prior to the time experiment, this group may represent a response to stimuli related to the experimental process (technical perturbations), which are unobservable without more information. We were unable to reject the null hypothesis of no enriched GO annotations for any genes in this group. The same is true for pathway flow analysis at the 95% confidence level with Bonferonni correction. However, a relatively high percentage of the genes in Cluster 2 are highly homologous to our transcription factor reference set suggesting that Cluster 2 might contain transcription factor genes (Figure 5.6). Behavior similarity to Cluster 8 indicates Cluster 2 may contain the transcriptional regulators of genes in Cluster 8.

***V. riparia* genes in Cluster 3** also showed approximately parallel time profiles under long and short photoperiod, with significantly lower expression during short. Overrepresented annotations in Cluster 3 include photosynthesis terms (Biological Process); glyceraldehyde-3-phosphate dehydrogenase, NADP oxidoreductase, FMN binding, transcription repressor activity (Molecular Function); and photosystems terms (Cellular Component) (Table C). Since we conducted tests for stronger photoperiod effects in *V. riparia* than *Seyval*, the difference in profiles we see for *V. riparia* is larger than any difference that exists between photoperiod treatment in *Seyval*; the photosystems in *V. riparia* are more sensitive to changes in photoperiod. This might be necessary to recognize the environmental signals to enter dormancy. In addition we noticed that these photosynthesis genes decreased expression over the time course under the long photoperiod treatment, while we would expect that it is constant. Further investigation with a pathway flow analysis showed (without Bonferroni correction) that gene expression for enzymes that catalyze many energy storage pathways are shutting down during the throughout the experiment (Table C.7). It remains to be discovered why these genes also decrease expression under the long photoperiod treatment.

For *V. riparia* genes in Cluster 4, which shows a sharp drop in expression for short days when compared to long days, we were unable to reject the null hypothesis for any annotations. However, pathway flow analysis without Bonferroni correction results in a short list of pathways including 13-LOX biosynthesis and phospholipases. This could be due to an over-representation of phospholipase and lipoxygenase enzyme-coding genes in the cluster. It might also contain unknown transcription factors, as Figure 5.6 shows a relatively high percentage of Cluster 4 genes are homologous to our reference transcription factor set. Further, only five of the 61 genes in Cluster 4 existed in VitiCyc at the time of analysis.

Cluster 5 behavior is similar to Cluster 3 and pathway flow analysis shows similar redox pathways (Table C.9). Overrepresented annotations include acid phosphatase activity, lipoxygenase (Molecular Function); apoplast, cell wall, and integral to membrane (Cellular Compartment) (Table C.8). It is known that zinc inhibits cell wall acid phosphatases. While not statistically significant, we did observe several genes annotated as zinc ion binding in Clusters 2 and 4, where *V. riparia* genes exhibit higher expression during short photoperiod than long for roughly the first two weeks. This happens to be roughly the time at which Cluster 5 genes (over-representing acid phosphatase) begin to decrease in expression. It is possible that some underlying mechanism uses zinc to regulate acid phosphatases as *V. riparia* enters dormancy. Acid phosphatase activity is associated with the NADP+ salvage pathway in VitiCyc (Van Hemert et al., 2010), and Cluster 5 seems to exhibit a relationship with Cluster

8, described below.

In Cluster 6, *V. riparia* genes steadily increase expression at a faster rate during short photoperiod than long, for about one month, then become roughly equal after 42 days. We were unable to reject the null hypothesis for GO term enrichment on genes in this group. We able to discriminate interesting pathways using the Pathway Flow tool including resveratrol biosynthesis, which is associated with biotic and abiotic stress, and wax esters biosynthesis (Table C.10).

***V. riparia* genes in Cluster 7** show a clear spike in expression around two weeks after short photoperiod begins which does not occur during long photoperiod. These genes then switch off during the rest of the time series, while, during long photoperiod, they remain relatively constant until day 42. Overrepresented annotations in this cluster include phenylpropanoid metabolism, lipid transport, chromatin and nucleosome assembly, and cell wall organization (Biological Process); structural constituent of cell wall (Molecular Function); extracellular region, nucleus, chromosome, chromatin, and nucleosome (Cellular Component) (Table C). This suggests that *V. riparia* bud cells increase chromatin production after about two weeks of short photoperiod treatment and proceed to pack their DNA to prevent further growth and enter dormancy. After about three weeks, these genes follow a sharp decline in expression, which may be caused by the dormant state, when excess DNA packages is no longer necessary. This support recent findings in Horvath (2009). Since VitiCyc lacks gene regulatory information, our pathway flow analysis cannot check the chromatin and nucleosome GO annotation results. However, it did detect cell wall-related pathways (Table C.12). A relatively high number and percentage of members of Cluster 7 also showed homology to our transcription faction reference set (Figure 5.6). Behavior similarity to Cluster 1 indicates Cluster 7 may contain the transcriptional regulators of genes in Cluster 1.

Lastly, *V. riparia* genes in Cluster 8 increase in expression over the time course. Overrepresented annotations in this group include RNA polyadenylation, response to oxidative stress (Biological Process); chitin binding, chitinase, antioxidant activity, peroxidase activity, phosphorylase, vitamin binding, and polynucleotide adenyltransferase activity (Molecular Function) (Table C.13). The overall function of genes in Cluster 8 is to respond to oxidative stress. Cluster 5 contains genes involved in recycling of energy molecules like NADP+, which results in production of antioxidants. We propose an inverse relationship between Clusters 5 and 8 where bud cells adapt to an energy system change seen in Cluster 5 by protecting themselves from oxidative damage using mechanisms represented in Cluster 8. Pathway flow analysis reveals energy storage pathways including sorbitol degradation into D-fructose and starch biosynthesis (Table C.14).

Likely transcription factor clusters include Clusters 2 and 7 because they reach the most significant hit-enrichment *p*-values in Figure 5.6. Besides using cluster behavior to hypothesize which clusters are regulated by these respective transcription factor groups, we can use their functional annotation to explain the difference expect value cutoffs at which each cluster reaches high significance. Cluster 2 reaches low *p*-values around very conservative expect value cutoffs (1e-100) while Cluster 7 reaches low *p*-values at higher, less conservative cutoffs (1e-20). Cluster 2 is not well annotated, but Cluster 7 is enriched with chromatin remodelling-related GO annotation. The transcription factor clusters also map to different transcription factor families and have expression patterns similar to other non-transcription factor clusters; Cluster 7 may represent the transcriptional regulators for the cell wall reorganizing members of Cluster 1 and Cluster 2 may represent the transcriptional regulators of the energy-storage-related genes in Cluster 8. Future work includes closer examination of these putative transcription factors and their families along with promoter binding site detection of their respective response cluster members.

Our results provide insight into possible cellular mechanisms that occur at the onset of grape dormancy, by investigating known biological processes related to overrepresented annotation as well as examination of possible experimental variation within a treatment type over time are necessary, we proposed biological explanations for our computational results. These explanations are the beginnings of new biological models for the underlying mechanisms during photoperiod-induced bud dormancy in *V. riparia*. We observed transcriptional responses to environmental signals as well as different groups of genes responding to one another. Lastly, we have shown that GO term enrichment and pathway flow analysis can complement each other in hypothesis generation. Further, our results detected pathways in VitiCyc that should be removed because they do not occur in plants.

Future work requires re-clustering of the genes using a higher K value in order to glean information about more intricate variations in expression profiles. Although we chose 8 clusters for this project because it allowed for demonstration of data trends and minimized cluster overlap, a much larger k is suitable for identifying smaller sets of genes that are involved in the same process or are subject to the same regulatory mechanisms. Smaller clusters will also be capable of more precisely differentiating between gene expression profiles and we can merge clusters at our discretion.

We also plan to examine locations of genes in each cluster on the *V. riparia* genome using our extensive web-based annotation system, which contains an interconnected GBrowse and annotation BioMart. Recall that our data actually measures probes, or transcripts that are portions of genes. Physical proximity of probes on the genome and correlated expression profiles can suggest that they belong to the same gene. This exploration may also identify genes that are co-transcribed and alternative splicing events. Currently this effort has produced an experiment-wide information system at http://vitis.student.iastate.edu/VV18/ with cluster behavior plots and cluster gene lists which link to both GBrowse views of gene locations and annotations in our BioMart database.

Methods

Six biological replicates from each genotype (*Seyval* and *V. riparia*) were grown in a greenhouse under fixed conditions. Of each group of six, three were grown with a 13 hour (short) photoperiod and the other three were grown under a 15 hour (long) photoperiod. Each replicate was sampled at seven time points: days 1, 3, 7, 14, 21, 28, and 42. All plant specimens were two to six year old vines. Prior to applying the photoperiod treatment, all plants were grown under long day conditions until all vines reached 12-15 nodes. Vines were then randomized into groups for photoperiod treatments. Temperature was maintained at 25±3 during the day and 20±3 at night for all treatments. Samples were extracted from the plants using a novel bud RNA extraction technique (Anne Fennell, unpublished). Microarray experiments were performed for each RNA sample using GeneChip *Vitis vinifera* (grape) by Affymetrix. This experiment, performed in 2007, was repeated in 2008 using the same replication. These two years will be referred to as "year 1" and "year 2". The shorter photoperiod (13 hours of sunlight per day) is intended to simulate the condition of approaching winter - the time at which the plant enters endodormancy. The longer photoperiod (15 hours) is used as a baseline from which we measure the effect of the shorter period. To differentiate between genetic activity related to endodormancy and other activities (e.g., photosynthesis), another grape species, *Seyval*, was examined. *Seyval* is a white wine grape plant that does not enter endodormancy. By collecting data for both cultivars, it is possible to identify differentially expressed genes across the photoperiod treatments, and then extract the subset specific to *V. riparia* for further analysis of importance to endodormancy.

The entire experiment for both years resulted in 167 Affymetrix microarray hybridizations because one replicate sample was accidentally lost or destroyed. Each Affymetrix *V. vinifera* microarray platform contains 16436 probes in all, made up of transcripts from *V. vinifera* as well as transcripts from other *Vitis* species, based on a large expressed sequence tag (EST) database in 2003. Additionally, the microarray platforms have positive and negative controls, which are constantly expressed and not expressed, respectively, for all samples.

All Affymetrix data were downloaded from the Plant Expression Database (PLEXdb.org)

(Wise et al., 2007), experiment VV18. At the time of this project, the data are not published and remain available only to PlexDB user accounts which are part of the Fennell-led study.

Platform Integration Technical Study

Before the single-platform Affymetrix results, the first part of this work entailed a technical comparison of the data based on a subset of the data points where samples were hybridized on Nimblegen "MD" Vitis and Affymetrix Vitis platforms in technical replicates (Table 5.2). This preliminary technical study resulted in a better functional annotation for interpretation of the full Affymetrix results plus a new model for platform benchmarking and integration.

Table 5.2 Existing data points for the two microarray platforms.

	Day 21	Day 28	Day 42
Long Days	3 Affy 3 MD	-	3 Affy 3 MD
Short Days	3 Affy 3 MD	3 Affy 3 MD	3 Affy 3 MD

Selecting Perturbed Probes

We selected perturbed probes on each platform by comparing their Coefficients of Variation (CV) to that of the respective platforms' selected controls. We selected 32,024 (12.2%) Affymetrix probes and 11,287 (10.0%) Nimblegen MD probes (Figure C.2).

Linked Probes

Two linked probes are expected to similar expression patterns under the same treatments based on their positions in the genome. We link probes across platforms when they share the same exon. We can model and visualize the these links with a graphical network where nodes are probes and edges link linked probes (Figure C.3).

We examined Affymetrix-MD linked probes where both the unspliced 60-bp MD and 25-bp Affymetrix probes share an exon in the gene models. True linked probes should be correlated and were selected using empirical FDR-corrected p-values from 10,000 random linked probes' correlation values. For 8,962 (46.6%) of the linked probes, we reject the null hypothesis that the Affymetrix probe is misaligned to the genome where either Affymetrix or MD is perturbed at the 95% confidence level (Figure C.4).

Normalizing for Integration

In order to integrate expression values from different platforms, we must eliminate any platform effect while maintaining the treatment effects. We mean-centered and mean-scaled all probes using Level Scaling (van den Berg et al., 2006) (Equation 5.1). We then fit both a "full" linear model with a platform effect (Equation 5.2) and a "reduced" model without a platform effect (Equation 5.3). We tested how much the fit improved under the full model. Level Scaling removed the platform effect completely (Figure C.5).

$$\begin{aligned}
\hat{x}_{icdpr} &= \frac{x_{icdpr} - \bar{x}_{ic\cdots}}{\bar{x}_{ic\cdots}} && (5.1)\\
\hat{x}_{icdpr} &= \mu_i + \tau_{ic} + \alpha_{id} + \beta_{ip} + \epsilon_{icdpr} && (5.2)\\
\hat{x}_{icdpr} &= \mu_i + \alpha_{id} + \beta_{ip} + \epsilon_{icdpr} && (5.3)\\
i &= 1 \ldots 2024 \text{ (”true” linked probes)}\\
c &= 1 \ldots 2 \text{ (platforms)}\\
d &= 1 \ldots 3 \text{ (days)}\\
p &= 1 \ldots 2 \text{ (photoperiods)}\\
r &= 1 \ldots 3 \text{ (replicates)}
\end{aligned}$$

Integration using a Swappable Gene Prediction model

With continued development of new gene expression detection platforms, quantifying expression with respect to gene models is non-trivial. From older platforms such as cDNA microarrays to short-read-based probe chips to transcriptome sequencing, different platforms measure different units in an attempt to measure the same thing: gene expression. When we have a reference genome we can place measured probes or reads on specific genome positions, but there are often multiple solutions. Further complication comes from incorporating experimentally supported and predicted gene models. This systems approach to gene expression quantification can be modeled with four variable, or swappable, levels of information: 1) a genome, 2) base-specific expression levels, 3) gene model predictions, and 4) functional annotation (Figure 5.7). For each combination of the four levels, expression data from different platforms are integrated into meta-probes and quantified with respect to different gene models and compared for biological interpretation including alternatively spliced genes. We would like to be able to swap gene models and transcriptomics platforms in and out of our interpretation system. The problem is that different gene models will align with probes differently, causing misinterpretation of gene splicing and expression, or missing it altogether.

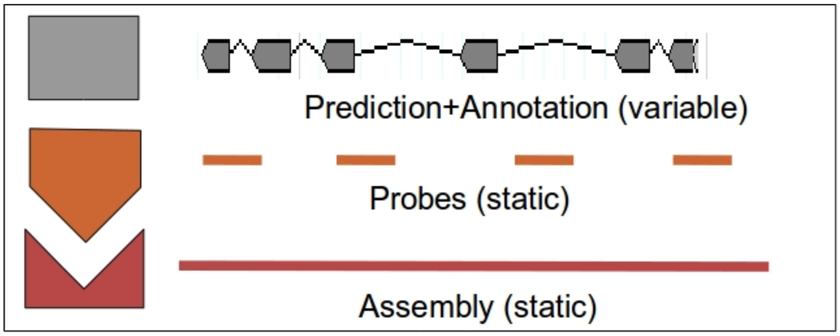


Figure 5.7 We would like to be able to mix and match expression platforms and gene models. If we assume the genome assembly and microarray probe alignments to positions on that assembly are relatively static (do not change more often then every few years), we seek the ability to interpret expression quantification from the probes in the context of any gene model prediction and its accompanying functional annotation.

Probe-Exon Systems. A Probe-Exon System is a small network of exons linked by probes which overlap them. In a probe-exon system network, exons are nodes and probes are edges linking the nodes (Figure C.6).

For a given probe-exon system, if we assume that the observed fluorescence of a probe equals the weighted sum of the exons which it overlaps, we can often fit a linear model to the data using Multivariate Multiple Regression and Least Squares optimization (Equations 5.4-5.8).

$$m = 1 \ldots M \text{treatments}$$

$$n = 1 \ldots N \text{probes in system}$$

$$w = 1 \ldots W \text{exons in system}$$

$$X_{nm} = \text{the RMA-normalized fluorescence of probe } n \text{ under treatment } m$$

$$Y_{nm}{}^{(0)} = \frac{X_{nm} - \bar{X}_{n\cdot}}{\bar{X}_{n\cdot}} \text{ (Level-scaling (van den Berg et al., 2006))} \tag{5.4}$$

$$Y_{(n \in \Upsilon)m} = \frac{med(Y_{Pm})}{\text{coverage of } P \text{ on } w} \tag{5.5}$$

$$\text{where } \Upsilon \text{ is all unspliced probes, } w \text{ is the exon of probe } n,$$

$$\text{and } P \text{ is the set of all unspliced probes on } \xi$$

$$Z_{nw} = \frac{\text{coverage of probe } n \text{ on exon } w}{\text{total coverage of probe } n} \tag{5.6}$$

$$[Y]_{N \times M} = [Z]_{N \times W} [\beta]_{W \times M} + [\epsilon]_{N \times M} \tag{5.7}$$

$$[\hat{\beta}]_{W \times M} = (Z'Z)^{-1} Z'Y \tag{5.8}$$

Single-Platform Analysis of the Full Photoperiod-induced Bud Dormancy Data

Data Normalization

The Robust Multi-array analysis (RMA) (Irizarry et al., 2003a) algorithm is used for background correction and normalization, and median polish (Irizarry et al., 2003b) for probeset summarization. After RMA, the data is structured as a 167 × 16436 array of log-transformed probe abundance data, where 167 is the number of hybridizations and 16436 is the number of probes (with controls removed).

To satisfy a full factorial, model-based analysis, the missing hybridization was imputed by calculating the mean of the existing two replicates for each gene and treatment combination. Specifically, each probeset has only two hybridizations from *Seyval* under the 15-hour (longer days) treatment at day one. The third imputed replicate value for each probeset was added for each probeset after RMA normalization and summarization.

For our purposes, a non-zero year effect (calculated using Equation 5.9) is not biologically interesting, but prohibits pooling of replicates from each year; Visual inspection of the histograms in Figure C.7 reveals that all gene-treatment combinations exhibit a consistent year effect. There is a slight leftward skew that may be explained if each distribution is actually the sum of two distributions; a major population of genes with a non-zero year effect centered below zero, and a minor population of genes with year effect centered at zero. In order to pool replicates across years, we adjusted all values from year 2 according to Equation 5.10.

$y_{igpd} = \overline{x_{igpd1\cdot}} - \overline{x_{igpd2\cdot}}$ where y_{igpd} is the year effect for the i^{th} probeset of the g^{th} genotype under the p^{th} photoperiod treatment measured at the d^{th}day, and x_{igpdyr} is the RMA normalized fluorescence value for the i^{th} probeset of the g^{th} genotype under the p^{th} photoperiod treatment measured at the d^{th}day in the y^{th} year and the r^{th} replicate. (5.9)

$x^{(cor)}_{igpd2r} = x_{igpd2r} + y_{igpd}$ where y_{igpd} and x_{igpdyr} follow Equation 5.9 and $x^{(cor)}_{igpd2r}$ is the year-effect-corrected x_{igpdyr} (5.10)

Data filtering

In microarray data analysis, scientists traditionally apply two initial filtering strategies: filtering by low absolute value and filtering by low variance. Typically, both filters involve some arbitrary cutoff (e.g., bottom 20 percent of genes) to determine which genes will be excluded from further analysis. Removing genes with low absolute value is motivated by inaccuracies in microarray experiment measurements for probes with intensities near zero. Removing genes with low variance across experimental conditions will eliminate genes from the data that are not likely to be involved in biological processes related to the organism response of interest. These are considered ”uninteresting” genes given the motivation of the study (van Iterson et al., 2010).

For our microarray data, instead of employing these filters, we propose using the control probe data for filtering genes. Our method begins by examining the variation that exists in control probes. The control probes on the microarray platform consist of positive controls, corresponding to genes that are always ”on”, and negative controls, corresponding to sequences that do not exist in these species and are therefore always ”off”. Because of the consistent behavior of the controls, it is possible to inspect the variance across our experimental conditions and interpret this as experimental error or microarray platform noise. Now, considering every gene on the chip, if its variance is similar to the technical noise, we are unable to attribute behavior to any treatment and the gene should be eliminated from further analysis.

The Affymetrix Vitis gene chip contains 166 control probesets including both negative controls which are known not to match any Vitis genes as well as positive controls which are known to be relatively constantly expressed in all or most Vitis plant cells. The negative controls are based on a Bacterial Artificial Chromosome (BAC). The positive controls are based on highly conserved actin genes which are expressed similarly for all cells. Visual inspection of boxplots in Figure C.8 for expression of control probes allows for identification of negative and positive controls based on their mean expression values. The positive control group was used as a model for the error variance due to the microarray platform. This subset was chosen because the variability of the positive controls was larger than the variability of the negative controls; a difference that may be a consequence of increased measurement error for larger probe intensities. Because we are interested in genes that are expressed under at least one experimental condition, it is most appropriate to use the larger variability of the positive controls as a criterion for filtering.

Inspecting multivariate normality for 16436 distributions is far from trivial. However, using the positive controls, we can inspect the multivariate normality of technical noise for expressed genes. To do this, we first applied a data reorganization of the positive controls. Initially, each of the 20 positive controls were measured 167 times, plus 1 imputed measurement. We rearranged the data matrix of positive controls to a $(20 * 6 = 120) \times (168/6 = 28)$ matrix where each row is a single replicate.

The Quantile-Quantile plot in Figure C.9 of statistical distances from each of the 120 positive controls to their 28-dimensional mean against theoretical χ^2 quantiles shows relatively good fit, except for some outliers in the tails. We conclude that the technical noise for expressed genes follows something near a multivariate normal distribution. We can also safely conclude that technical noise in any subset of the measurements on a gene follow multivariate normal distributions, and technical noise on each measurement on a gene follow univariate normal distributions.

Next, we calculated the sample variance, S^2, for each of the 120 positive control replicates by Equation 5.11 and used the resulting distribution of sample variances as a reference for discriminating variable non-control probesets. We applied the same data reorganization to all non-control probesets in order to compare their variances across treatments to that of the positive controls. Each non-control probeset has six replicates under all treatments, but we wish to calculate a single representative variance for each probeset. To do this, we calculated the mean variance of all six replicates for each probeset using Equation 5.12.

$$S^2_{control_{ir}} = \frac{1}{n-1}\sum_{g,p,d}\left[(x_{igpdr} - \overline{x_{i\cdots r}})^2\right] \tag{5.11}$$

$$n = 28 \text{ (the sample size)}$$

$$S^2_{control_{ir}} = \text{the sample variance for the } i^{th} \text{ control probeset in the } r_{th} \text{ replicate}$$

$$x_{igpdr} = \text{the RMA normalized fluorescence value for the } i^{th} \text{ control probeset of the } g^{th} \text{ genotype under the } p^{th} \text{ photoperiod treatment measured at the } d^{th}\text{day for the } r^{th} \text{ replicate.}$$

$$S_i^2 = \frac{1}{R}\sum_r\left[\frac{1}{n-1}\sum_{g,p,d}\left[(x_{igpdr} - \overline{x_{i\cdots r}})^2\right]\right] \tag{5.12}$$

$$= \text{the mean sample variance for all } R = 6 \text{ replicates of the } i^{th} \text{ test probeset}$$

$$S^2_{control_{ir}}{}^{(VR)} = \frac{1}{n-1}\sum_{p,d}\left[(x_{i1pdr} - \overline{x_{i1\cdot\cdot r}})^2\right] \tag{5.13}$$

$$S_i^{2(VR)} = \frac{1}{R}\sum_r\left[\frac{1}{n-1}\sum_{p,d}\left[(x_{i1pdr} - \overline{x_{i1\cdot\cdot r}})^2\right]\right] \tag{5.14}$$

A principal components analysis at this stage (not shown) revealed sufficient separation between genotypes, but other effects were not easily identified in the first few directions of variability. This is probably because applying both Equations 5.11 and 5.12 across both genotypes resulted in an over-representation of probesets whose targets exist in one genotype and not the other and do not necessarily respond to the other treatments. Many of these probesets (data not shown) exhibit large variances, but only due to the genotype factor, which is not of biological interest here. To avoid these genotype-specific probesets, our filtering used adjusted versions of Equations 5.11 and 5.12 where a stronger filter was applied to the data: instead of filtering based on variance across all experimental conditions, we filtered genes based on variance across only the treatments to *V. riparia*. This modification to the filter reduces the effect of genotype and eliminates genes that did not vary across the photoperiod treatments within *V. riparia*. By using Equations 5.13 and 5.14 for the controls and non-controls, respectively, we selected only probesets with high expression variance as *V. riparia* specifically entered endodormancy.

A plot of the number of genes versus their quantile relative to the positive control variance data revealed intuitive trends in the data. As shown in Figure C.10, most of the genes exhibited very little variability across experimental conditions and could quickly be eliminated from analysis. For our purposes, we have selected all genes above the 75^{th} percentile in this figure. This corresponds to 1304 genes, or approximately the top 7.9 % of the original set of genes.

Multivariate analysis of variance

In an effort to mine biologically meaningful genes out of the full set, we conducted a series of Multivariate Analyses of Variance (MANOVA) on each of the 1304 filtered probesets. An interaction effect between genotype and photoperiod is biologically relevant because we are searching for genes perturbed by the short photoperiod due to dormancy processes and not photosynthetic processes. To test this, we applied a 2-way MANOVA to each probeset with the multivariate linear model in Equation 5.15. Notice that a separate model is fit to each probeset, which is represented by a 24 treatments × 7 time points matrix of observations. The model assumes multivariate normality, which can safely assume for chip effects, but not necessarily other effects. It also assumes homogeneous variance-covariance matrices between each treatment group. We neglected to formally test for these features for any probesets and assume they are true.

Each gene is a 24 × 7 matrix

$$\begin{aligned}
\vec{x}_{igpr} &= \vec{\mu}_i + \vec{\tau}_{ig} + \vec{\beta}_{ip} + \vec{\gamma}_{igp} + \vec{\epsilon}_{igpr} \\
i &= 1 \ldots 1304 \text{ (probesets)} \\
g &= 1 \ldots 2 \text{ (genotypes)} \\
p &= 1 \ldots 2 \text{ (photoperiods)} \\
r &= 1 \ldots 6 \text{ (replicates)} \\
0 &= \sum_g \vec{\tau}_{ig} = \sum_p \vec{\beta}_{ip} = \sum_g \vec{\gamma}_{igp} = \sum_p \vec{\gamma}_{igp} \\
\vec{\epsilon}_{igpr} &\sim iidN_p(\vec{0}, \mathbf{\Sigma_i}) \qquad (5.15)
\end{aligned}$$

We used Wilk's Lambda, a common MANOVA test statistic, calculated with Equation 5.16, to compare the sums of squares and cross products for the interaction (SSP_{int}) and residual error (SSP_{res}). We compared its Bartlett-scaled test statistic to a χ^2 distribution with $(2-1)(2-1)7 = 7$ degrees of freedom. Unfortunately, this method was unable to further discriminate genes; a significant interaction effect ($\vec{\gamma}_{igp}$ in Equation 5.15) was not found for any gene. This may be due to the small number of replicates (6).

$$\Lambda = \frac{|SSP_{res}|}{|SSP_{int} + SSP_{res}|}, \Lambda \to 0 \Rightarrow \text{strong interaction effect} \qquad (5.16)$$

We altered our approach to mining dormancy-related genes from the 1304 high-variance-in-*V. riparia* genes by applying a One-way MANOVA (Equation 5.17) to each gene, but for the two genotypes separately. This would result in two test statistics for each gene which could be compared with the goal of selecting those with a more significant test statistic in *V. riparia*. The new test statistic, Delta (Δ), compares the time-multivariate photoperiod effect, which is measured using Equation 5.18, for *Seyval* and *V. riparia* by taking the log-ratio of one to the other, as in Equation 5.19.

Each gene is a 12 × 7 matrix

fit twice (once for each genotype)

$$\begin{aligned}
\vec{x}_{igpr} &= \vec{\mu}_i + \vec{\beta}_{ip} + \vec{\epsilon}_{igpr} \\
i &= 1 \ldots 1304 \text{ (probesets)} \\
g &= 1 \ldots 2 \text{ (genotypes)} \\
p &= 1 \ldots 2 \text{ (photoperiods)} \\
r &= 1 \ldots 6 \text{ (replicates)} \\
\sum_p \vec{\beta}_{ip} &= 0 \\
\vec{\epsilon}_{igpr} &\sim iidN_p(\vec{0}, \mathbf{\Sigma_i})
\end{aligned} \tag{5.17}$$

$$\Lambda = \frac{|SSP_{res}|}{|SSP_{phot} + SSP_{res}|}, \Lambda \rightarrow 0 \Rightarrow \text{strong photoperiod effect} \tag{5.18}$$

$$\Delta = \ln \frac{\Lambda_{VR}}{\Lambda_{SV}}, \Delta \downarrow \Rightarrow \text{stronger photoperiod effect in } \textit{V. riparia} \tag{5.19}$$

Like the variance filtering, we compared Delta for each gene to the distribution of Deltas from all 166 controls. We were able to use all controls because expression value location and scale do not affect the Delta statistic, and Delta is calculated for each probeset independently. While we neglect to attempt to show it analytically, Delta for our controls clearly follows a Normal distribution, when checking the Q-Q Normal plot in Figure 5.8. We also see non-centrality in the histogram of control Deltas. There seems to be some factor causing generally stronger photoperiod effects in *Seyval*. Since we are examining controls, there should only be a technical (not biological) explanation for this. There may be some feature of *Seyval* that causes its RNA to amplify and hybridize more variably than *V. riparia* and is not corrected by our normalization technique. We estimated the Delta sampling distribution's parameters (μ and σ) with mean and sample variance of the controls Delta sample, respectively.

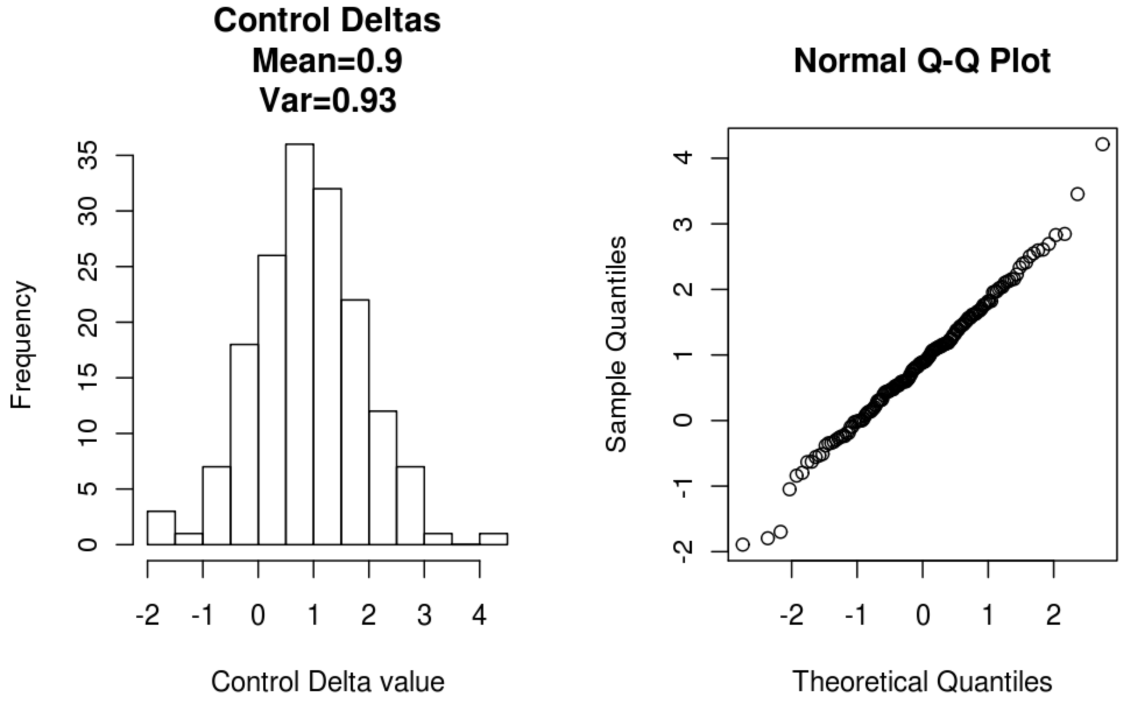


Figure 5.8 We used the sample of controls to estimate the parameters of the Δ sampling distribution. For each of these controls, the Δ statistic was calculated using Equation 5.19. The histogram of Δ statistics for all controls is on the left and the Normal Quantile-Quantile plot is on the right. The sampling distribution of Δ very closely resembles a Normal distribution with mean 0.9 and variance 0.93. We used this distribution to test significance (in the lower tail) of the experimental Δ statistics.

For genes with Delta values in the lower tail of the Normal distribution with parameters estimated by the controls, we can reject the null hypothesis that their photoperiod effect is similar in both genotypes. We calculated p-values for the 1304 genes, shown in a histogram in Figure 5.9. False Discovery Rate was corrected into q-values according to (Storey, 2003). Both distributions appear very appropriate for mining genes for perturbations; the distribution of p-values is generally uniform, except for a tall mode near zero, made up of the sizeable set of genes perturbed by the treatments ((Fodor et al., 2007)). There is also a smaller upper mode, indicating a second set of genes which were perturbed by short photoperiod more in *Seyval* than in *V. riparia*. 770 Vitis probesets' Delta values fall below the fifth percentile of the sampling distribution. These are the genes perturbed by short photoperiod more in *V.*

riparia than in *Seyval*, which, after our filtering and multivariate testing, we can assume are at least mostly dormancy-related genes. Figure 5.4 plots five of the 770 gene profiles, showing that we have detected genes perturbed by photoperiod in *V. riparia*.

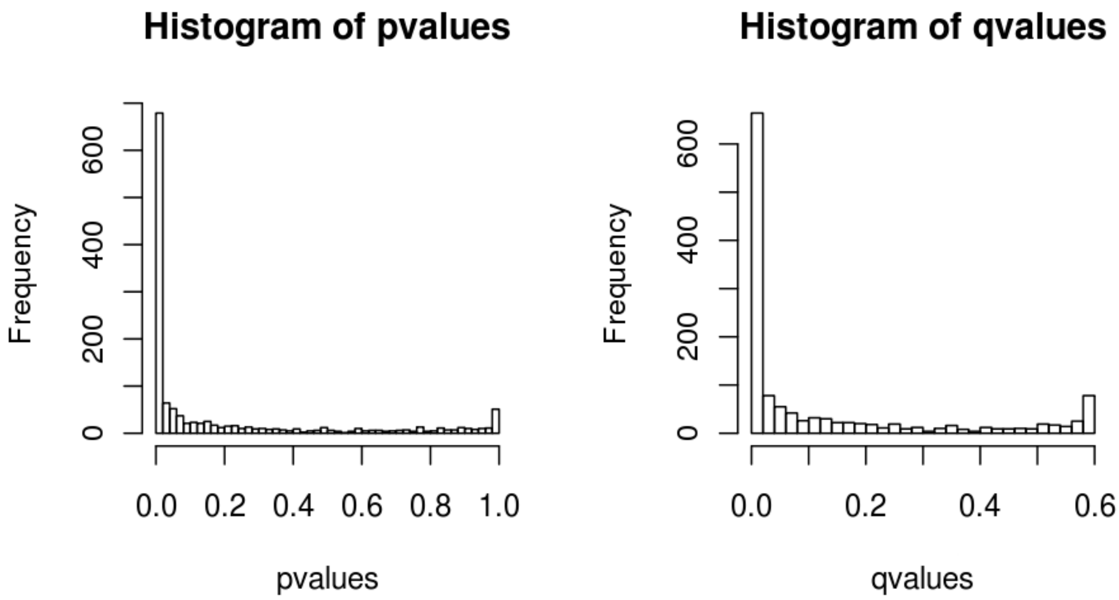


Figure 5.9 Both p-values and FDR-corrected q-values form good histograms, where many probesets are of the null hypothesis (not dormancy-related), but many form a spike near zero, represented the set of dormancy-related probesets. The small spikes on the upper tails represent probesets measuring genes which are actually perturbed by photoperiod more in *Seyval* than *V. riparia*. We analyze the lower-tail genes in this work.

***K*-means clustering**

We performed K-means clustering using the median expression value of *V. riparia* samples given a short photoperiod treatment for each time point per gene. The use of medians for all replicates was to prevent outliers from strongly influencing the expression summary for each gene. This results in a 7 × 770 data matrix where the number of rows corresponds to the number of time points (days 1, 3, 7, 14, 21, 28, and 42) and the number of columns is the number of previously identified dormancy-related genes (Figure 5.5).

The goal of clustering genes by their expression over the time series is to identify groups of genes that exhibit similar behavior in *V. riparia* with a short photoperiod treatment. Biologically, genes that are highly correlated with each other are often involved similar functions. They may also be subject to the same regulatory processes or co-transcribed.

K-means clustering uses an iterative algorithm to assign objects to k clusters such that the distances from objects within a cluster to its center is minimized. Both k, the number of clusters, and the distance metric are user-specified. Correlation was the distance metric used for our clustering because it captures the relation between genes that is most interesting for our study. The choice of $k = 8$ was motivated by hierarchical clustering and silhouette plots not shown.

Functional Analysis of the 8 Clusters

Biologically, it is of interest to explore functional and regulatory relations that exist within groups of genes that are highly correlated. We used Gene Ontology (GO) (Ashburner et al., 2000) annotations to investigate individual clusters and also search for overall biological significance of the 770 dormancy-related genes.

Gene Ontology is a collection of controlled biological terms used to define gene products properties. GO is comprised of three separate ontologies; Cellular Component, Biological Process, and Molecular Function. Each forms its own network of terms to describe characteristics of genes. Each ontology is constructed such that terms are nodes and they are linked by edges that describe defined relationships (e.g., ”sigma factor activity” is a ”transcription initiation activity”). The full collection of terms and links within an ontology forms a directed acyclic graph. Each gene in the 770 data set was mapped to a GO identification number that corresponds to GO annotation. Probesets were first mapped to the newest version of the Grapevine genome produced by the French-Italian Grape consortium (Delledonne, 2009). Probeset consensus sequences were aligned using BLAT (Kent, 2002) to all chromosomes in the genome assembly. Then, probesets were mapped to predicted gene models by identifying probesets and gene models which overlap by at least four bases. False hits from our technical correlation comparison were removed. Individual probes were then mapped to their respective positions on the genome, according to their positions in the probeset consensus sequences. A computational annotation effort at the University of Padova (unpublished) has produced (among others) GO annotation for the gene models, which were then attached to the genes' respective overlapping probesets. The mapping is not 1:1, and in many cases, many genes will have the same annotation. In addition, approximately half of the 770 genes are not currently annotated and cannot be included in the following data enrichment process.

In order to identify commonalities within clusters, we used a hypergeometric test to identify any annotations that are overrepresented within each group. Benjamini and Hochberg False Discovery Rate (FDR) correction was used to correct for multiple comparisons. Our criteria for overrepresented categories was a test for the cluster versus the entire annotated network for our organism. To say an annotation is overrepresented in a cluster is to reject the null hypothesis that the ratio of genes in the cluster with that specific annotation to the size of the annotated cluster is less than or equal to the ratio of all genes annotated with that specific annotation to all annotated genes (an odds ratio). Because of potential bias in annotation for each of the three ontologies, we chose to perform an individual test of significance per ontology at the 90% confidence level. All tests were executed in Cytoscape (Shannon et al., 2003) using the BiNGO plugin (Maere et al., 2005).

We also used our PathwayFlow web tool to discriminate VitiCyc pathways linked to the genes in each cluster by entering the members of each cluster as a query list. VitiCyc lacks gene regulatory relationships so we can only mine forward-direction query list successors. Members of each cluster were also compared to a set of reference transcription factors from PlantTFDB (Guo et al., 2008) made up of transcription factors from *Vitis vinifera*, *Populus trichocarpa*, *Glycine max*, *Oryza sativa*, and *Arabidopsis thaliana* (Figure 5.6).

Acknowledgment

The authors would like to thank Dr. Alicia Carriquiry for statistical guidance on this project. Data generation was funded by the National Science Foundation Plant Genome Research, DBI 0604755. Multivariate Affymetrix data processing was also partly funded by EEC 0813570. Platform integration was funded by the Grape Research Collaboration Network NSF-funded student travel program (DBI 0741876). 'Grazie mille' to Sara Zenoni and Marianna Fasoli for running many of the hybridizations.

APPENDIX A. GRAPEVINE DNA SEQUENCING PROJECTS

John L. Van Hemert [1], Jerome Grimplet [2], Marianna Fasoli [3], Alberto Ferrarini[3], Massimo Delledonne[3], and Mario Pezzotti[3], and Julie A. Dickerson[1]

On the transfer of functional annotation from one grapevine genome assembly to another

During my work on *Vitis* related projects, multiple assemblies and gene model predictions were created for the Grapevine. A major challenge was how to handle these different versions of important information and the different analyses based on them.

Abstract

In 2007 a draft assembly and gene prediction of the grapevine was made public for the scientific community. Since then, a new assembly which added more Sanger Sequencing reads to the assembly pool produced a new genome version with superior base coverage. Before the new version was created, much functional annotation was performed on the previous genome. In order to most efficiently annotate the new version, it is important to leverage as much completed work as possible by transferring ”8X” annotation to the ”12X” version of the genome. The 8X and 12X assemblies+predictions of the grapevine genome were compared to answer the question, ”Can we uniquely map 8X predicted genes to 12X predicted genes?” Predicted genes were compared between the two genome versions. Results show that while the assemblies and gene structure predictions are too different to make a complete mapping between the two,

[1] Electrical and Computer Engineering, Bioinformatics and Computational Biology, Iowa State University, Ames, Iowa

[2] Science Institute of Vine and Wine, Rioja University, Spain

[3] Biotechnology, University of Verona, Italy

interesting structures appear which enlighten our understanding of the transition from one genome version to the next.

Definitions

1. 8X: The grape genome published in 2007 by the French-Italian Consortium with average 8.4X assembly coverage (Jaillon et al., 2007).

2. 12X: The yet unpublished genome by the same group with increased coverage to 12X average (Delledonne, 2009).

3. Gene prediction: the computational prediction of ORFs, genes, UTRs, and CDS for a genome sequence.

4. Genome annotation: A mapping between predicted genes on a genome and functions, locations, processes, mutants, homologs, etc.

5. Chip annotation: A mapping between microarray probesets and functions, locations, processes, mutants, homologs, etc.

6. ”V0”: The assembly and gene prediction of the 12X genome by Genoscope. This will be included with the initial 12X paper/publication.

7. ”V1”: The improved assembly and gene prediction of the 12X genome by the Padova group. This will be released immediately after the ”V0” publication. This is also the prediction the Nimblegen chips are based on.

8. ”Sister genes”: Two versions of the same gene from different assemblies. Not to be confused with paralogs, which are homologous sequences from the same assembly.

9. ”Alignment series”: A group of local sequence alignments which appear to line up on a diagonal line when plotted on their respective chromosomes’ positions.

Methods

Megablast Megablast is an multiple sequence alignment tool designed for comparing nucleotide sequences which differ due to sequencing errors. It operates similarly to BLAST, but does not allow for affine gap penalties which attempt to model sequence indels in evolution (Altschul et al., 1990; Zhang et al., 2000). Megablast was used to compare 8X and 12X sequences where 8X sequences were the query set and 12X sequences were the subject database. For the 12X prediction, "V1" was used, which is the latest version of gene model predictions on the 12X assembly. Default Megablast parameters were used because results would be further filtered in a later step.

Entire predicted genes were compared The 8X and 12X assemblies are accompanied by respective gene structure predictions, which contain different types of subsequence predictions. These include genes, mRNAs, UTRs, introns, exons, and inter-genic spaces. Besides comparing the full chromosome assemblies, any set of one or more of these subsequence types could be used for comparison. Per the predictions, genes contain mRNAs, which contain UTRs, introns, and exons. Since open reading frame can be generally defined as a region of the genome which is potentially protein-coding, we can called these predicted gene regions ORFs. In this study, because we are detecting sequencing variation and not evolution, these complete ORFs were compared between the 8X and 12X assemblies + predictions.

Chromosomes were aligned Each assembly produced 19 ordered chromosomes plus a twentieth unknown chromosome which contains contigs which could not be assigned to any of the 19 chromosomes. Of these 20 chromosomes, many are accompanied by smaller partner chromosome labeled as random. These random chromosomes contain contigs which were assigned to the respective chromosomes, but could not be assembled in order with the other contigs. Expectedly, the 12X assembly contains smaller unknown and random chromosomes. Gene structure predictions were performed by Genoscope and the Padova group on all (ordered, random and unknown) chromosomes, producing ORFs to compare in all chromosomes. However, some analyses, such as those considering position information, must omit the random and unknown chromosomes.

Sequence homology presents a Cardinality Problem Megablast results in a many-to-many relationship between 8X and 12X ORFs. Hypothetically, the 8X prediction could define a long gene on a specific locus, while slightly different assembly in 12X version could cause a prediction of several separate genes spanning the same nucleotides. This results in many 12X predicted genes aligning almost perfectly with the same 8X gene. Of course, the reverse is also possible. Further, paralogous domains cause a confounding web of links between sets of genes. The degree to which a gene is linked to multiple sister genes in the other version is called cardinality. When we model the sister gene hits as a graph, where nodes are genes and edges represent Megablast hits, we create a bipartite graph where one side is the set of 8X sequences and the other side is the set of 12X sequences. Edges between the two sets indicate sequence homology hits (Figure A.1).

Hits were ranked and bests were selected The cardinality problem can be approached by ranking hits for each gene and selecting the best hit out of many for a gene with cardinality greater than one. This approach assumes that this best hit on a gene is the only real match and should be assigned as the one and only sister gene. Ranking and selecting best hits must be done ”in both directions.” That is, to assign the best hit for each gene in one assembly version and then assigning the best hit for each gene of the other assembly out of the remaining links. The resulting unique one-to-one mapping depends on which ”direction” is ranked and selected first. For this study, the 8X genes were first ranked and selected because this resulted in a larger number of unique one-to-one sister gene mappings. A measure of the alignment coverage was used to rank and select mappings.

Alignment coverage was used for scoring It is assumed that a good pair of sister genes will produce a local alignment which spans most of both gene sequences. When searching for these cases, we can define a function of the megablast alignment results which reflects alignment coverage. The function used for this study is simply a percentage comparing the

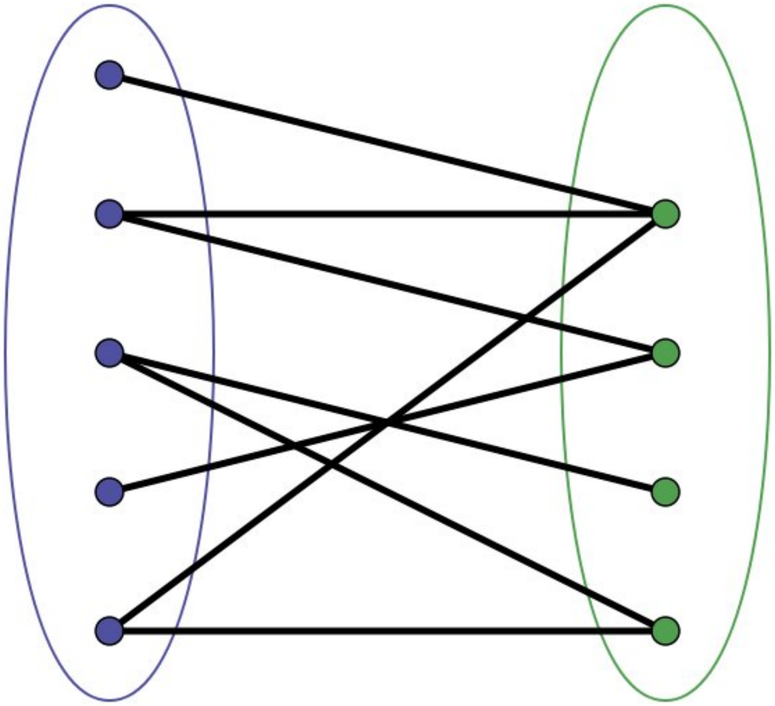


Figure A.1 Cross-assembly mapping is a bipartite graph problem; if gene models from one assembly on the left side are mapped to gene models from the other assembly on the right side, we are bound to find one-to-many and even many-to-many relationships.

length of the alignment region to the sum o the lengths of the respective hit ORFs. Indeed, this reflects the same measure as the normalized bit score on which BLAST's expect value is based, but the alignment coverage score is more intuitive here. Results are similar if not identical to using the bit score.

Results

Score distribution shows much noise When naively selecting best hits in both directions, we have a set of 21461 unique one-to-one mappings between sister genes on any chromosome (ordered, random, and unknown). Of these, 5182 pairs involve a random chromosome and 4109 involve an "Unknown" chromosome. While these are the hits which had the best alignment coverage for each respective gene, the alignment coverage values are far from perfect and reflect many false positives. Of this one-to-one mapping, we can plot a histogram of the alignment coverage scores.

The histogram is actually the sum of two different distributions. The first is clear in the left mode. This is a form of the Karlin distribution (Karlin and Altschul, 1990), which is the distribution of random best sequencing alignment scores from false sister gene pairs. The rightward mode is from a normal distribution folded at a maximum value of 100% alignment coverage. This shows the distribution of alignment coverage scores from true sister genes. We can draw the two separate distributions by eye. If we define a cutoff score around 85, we can mark the areas which indicate the number of true positives, false positives, true negatives, and false negatives (TP, FP, TN, FN, respectively). From this, it is clear that, at best, only a few thousand of the mappings are correct (Figure A.2).

Gene order series are observed in alignments Despite poor structure in the sister gene mappings based on alignment coverage, we can see interesting structure when considering the Megalast hits with respect to gene position. A plot of the relative chromosome positions in each of the 21461 unique sister gene pairs shows much structure. When sister pairs from the same chromosome are color coded as such, we see many pairs in series along the diagonal. This shows that sister genes are detected in line with their sisters on the other genome version. Black points indicate sister gene pairs which come from different chromosomes. Many of these do show series structure. Most of the black series are mapping from an 8X random or Unknown chromosome to a 12X ordered one (Figure A.3).

3D plots show that high coverage scores indicate "good series" We can plot the sister gene pair positions along with a third dimension to visualize scores in this context. A 3D view of the 21461 sister gene pairs shows that strong alignment coverages coincide with gene order series (Figure A.4). On paper, the 3D visualization is difficult to interpret, but using Ggobi (Swayne and Buja, 2004), the user is free to rotate the plot and see these signals.

Similar 3D plots were created for each of the ordered chromosome (Figure A.4). These

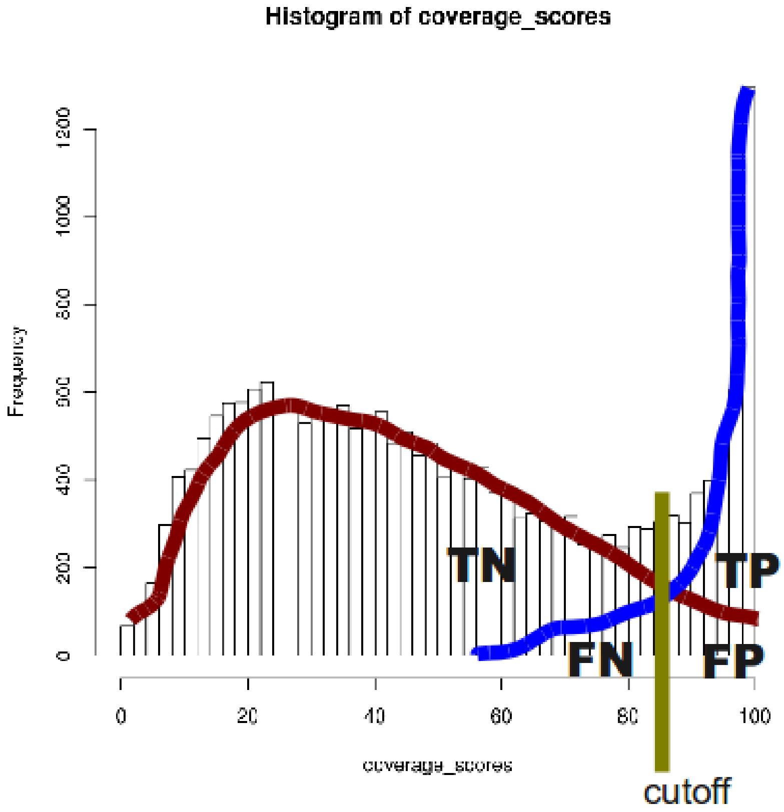


Figure A.2 Distribution of Cross-platform MegaBlast hit scores. We see the sum of two populations: false hits and true hits.

plots do not show unique ranked-and-selected sister gene pairs, but all Megablast hits where both genes came from the particular chromosome plotted. Since the genes are not ranked and selected, we clearly see low-scoring noise hits off the series and strong signals from the hits on the series, which are generally along the diagonal. The shows that we need not manually or computationally detect the sister gene pair series based using chromosome positions, which is a non-trivial signal processing task. Rather, we only need to select high alignment coverage scoring pairs. This partly validates the mass rank-and-select method for creating the unique one-to-one mapping.

Gene order inversions are observed Some alignment position plots show one or more sister gene pair series along a negatively sloped diagonal (Figures A.3 and A.4). These series

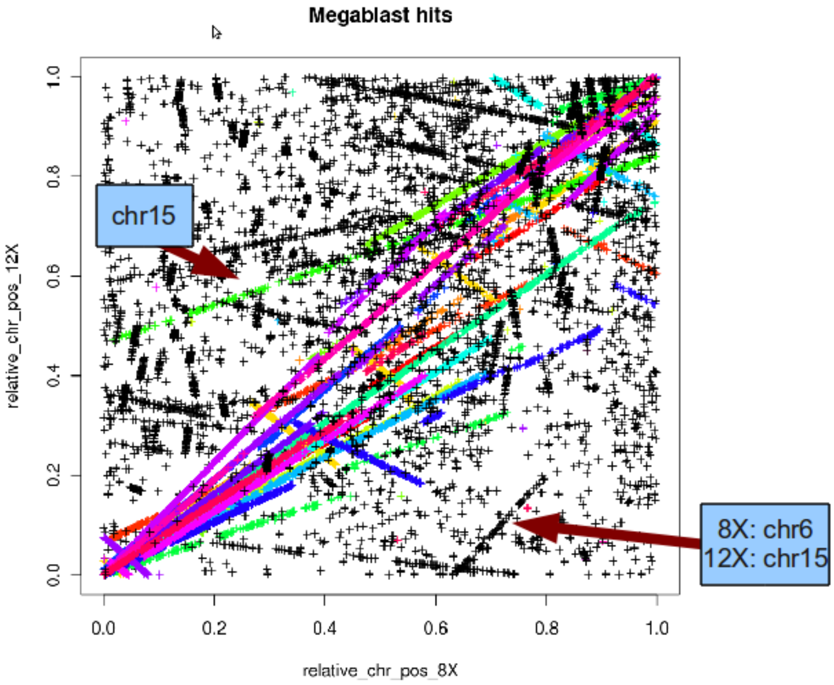


Figure A.3 Cross-platform MegaBlast hit positions. The X-coordinate is the relative position on the 8X chromosome and the Y-coordinate is the relative position on the 12X chromosome. Colors represent a hit between gene models on the same respective chromosome. Black points are for hits between gene models on differnent respective chromosomes.

indicate gene order inversions where a series of 8X genes are aligning with 12X sister genes in the reverse order. Notice that these series show scores as high as the positively correlation series.

Gene order inversions are at the assembly level Are these genes reversed in order at the assembly or prediction level? In order to answer this question, an ad-hoc pipeline for manual subsection of hits from the position plot in R-Ggobi was created. Using this pipeline, sister gene pairs were selected from the negatively sloped series from the alignment position plot for chromosome 12, which contains the most inversions. A Fisher test was conducted to test the null hypothesis that the ratio of opposite strand hits is the same for two sets of hits: 1) hits on the inversions and 2) all other hits. The test resulted in an immeasurably small p-value,

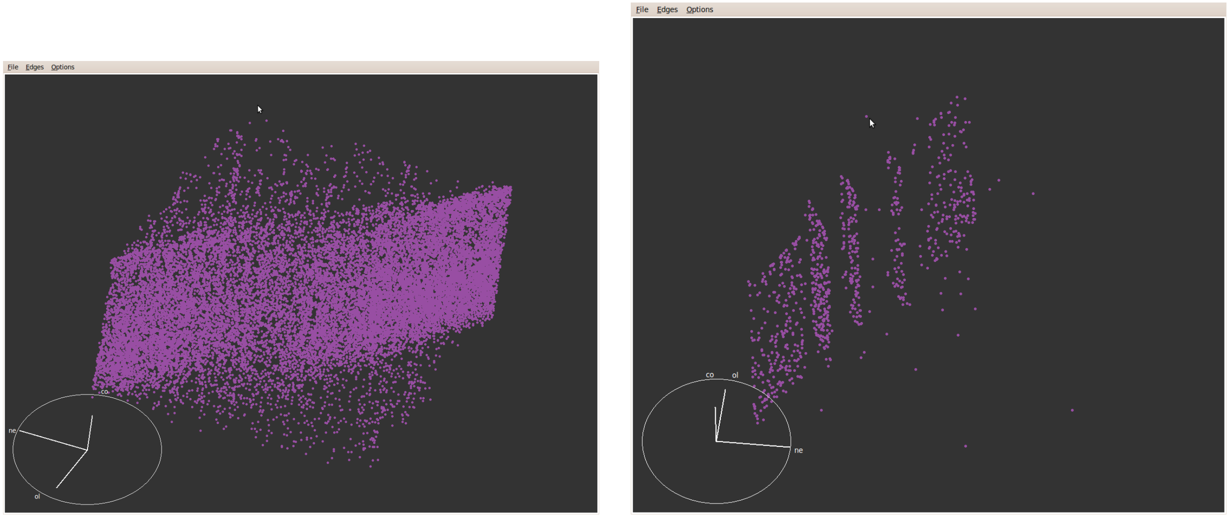
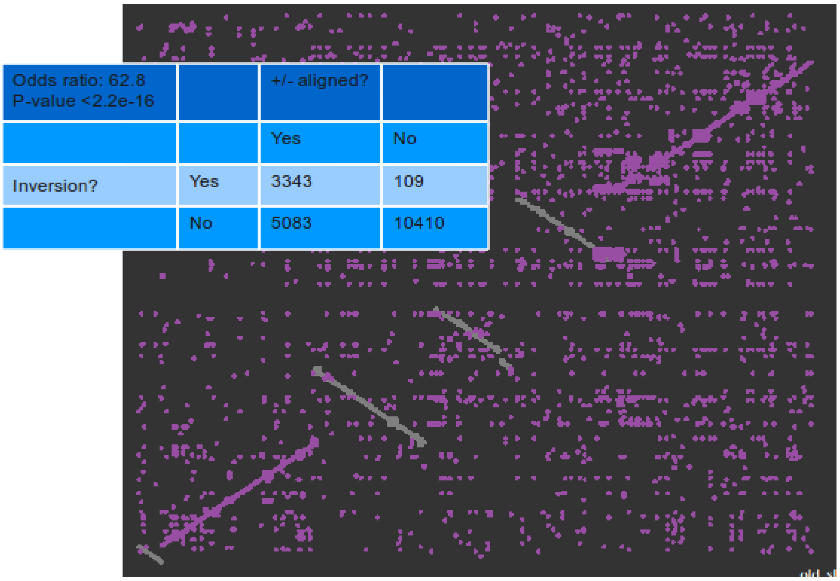


Figure A.4 3D plots of respective relative chromosome hit positions plus a third axis for alignment score for all hits (left) and for hits between 8X chromosome 1 and 12X chromosome 1 (right). Most other ordered chromosomes show a similar pattern.

allowing confident rejection of this hypothesis and to accept the alternative that sister gene pairs on inversions strongly tend to be opposite strand alignments. This indicates that the inversions are at the assembly level and were probably caused by inadequate linkage mapping in the 8X version (assuming the 12X is more correct). (Figure A.5).

Inversions are probably poorly assembled contigs in 8X Large assembly inversions are probably caused by sequencing errors on the inversion flanks or ends. These errors are probably caused by low coverage in these regions. A 3D plot where the third dimension is the assembly coverage for 8X may show that the ends of the inversions have low values while plotting the 12X assembly coverage may show better values in the same region. A plot where the third dimesion show the difference between 12X and 8X coverage may show a similar signal.

Drastic changes in Chromosome 15 are observed Returning to the plot of all ranked-and-selected sister gene pairs, notice that the chromosome 15 series begins midway through the 12X assembly. In addition, the beginning of chromosome 15 in the 12X assembly aligns best with a portion of chromosome 6 in the 8X assembly. While simple cis-inversions are



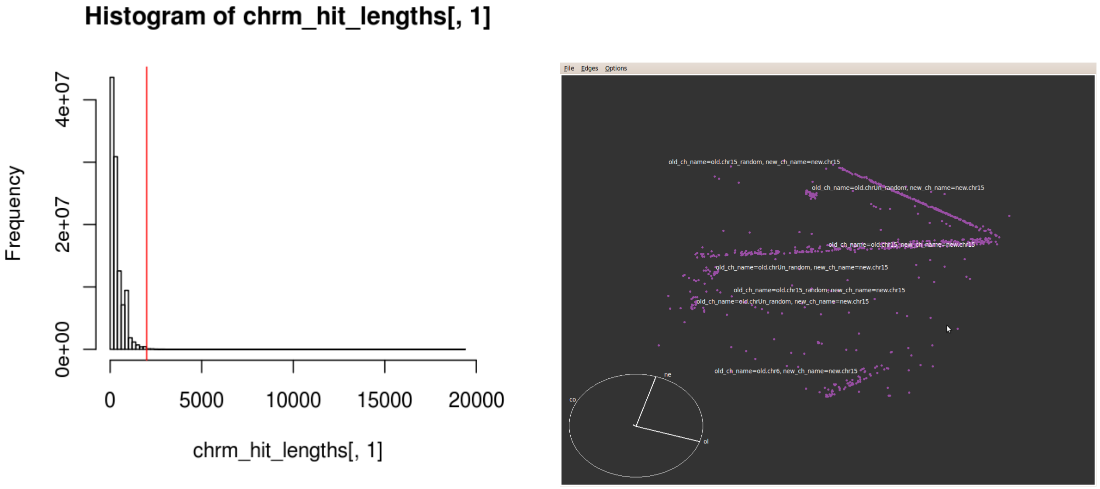
Odds ratio: 62.8 P-value <2.2e-16		+/- aligned?	
		Yes	No
Inversion?	Yes	3343	109
	No	5083	10410

Figure A.5 The cross-directional inversion alignment odds ratio shows that inversion hits were opposite-strand alignments while non-inversions were same strand (+) alignments.

minor side-effects of linkage mismapping, trans-inversions show a more severe and less recoverable inadequacy in the 8X sequence. A similar alignment plot for full chromosome alignments shows similar structure to the ORF alignment plots (Figure A.6). All 8X chromosomes were Megablast'ed against all 12X chromosomes using the same procedure as the ORF comparison. Only hits of length 2 kb or more were used because this reduced the number of hits from over 1.1e8 to less than 300000 as shown in the histogram of all chromosome hit lengths (Figure A.6). Placing the resulting local alignments on a gbrowse (GMOD, 2010) track, we confirm the tendency for inversion regions to align on opposite strands (Figure A.7).

Discussion

Visualization is important for comparative genomics Chromosome comparison is a large scale pattern recognition problem. The best available tool for pattern recognition is often the human mind. Therefore, visualization techniques are important for chromosome comparisons. For this study, a the multivariate data visualization tool Ggobi was used in concert with the sequence visualization tool Gbrowse. In particular, chromosome position plots of alignment



(a) Chromosome alignments scores.

(b) Chromosome alignments scores by positions.

Figure A.6 Distribution of full chromosome MegaBlast hit scores (A.6(a), the red virtical line is the right-most value used.) When inspecting these alignments using the 3D plot, we can see that the 12X (labeled ”new”) chromosome 15 ORFs align well with 8X ORFs from chromosomes 15, 6, random 15, and random Unknown A.6(b).

hits with a third score dimension conferred easy pattern recognition, while Gbrowse presented smaller scale examples of the patterns detected.

A possible dynamic programming approach to series detection If there had not been such a high correspondence between diagonal alignment series and high alignment coverage scores, these series might have been computationally detected. One possible method for this is a modified Smith-Waterman dynamic programming approach. Instead of tracing a path through a matrix of base pair alignments between two different sequences optimizing an overall alignment score, we would trace a path through a matrix of Megablast hits between two different chromosomes optimizing an overall position correlation. The caveat is that the alignment score can be calculated incrementally at the margin, adding base matches or gap one by one, and hit position correlation must be re-calculated for each increment for all previously included hits in the path trace, leading to a sort of inner dynamic programming problem. Future work may entail implementing such an algorithm and testing is accuracy in marking the series of

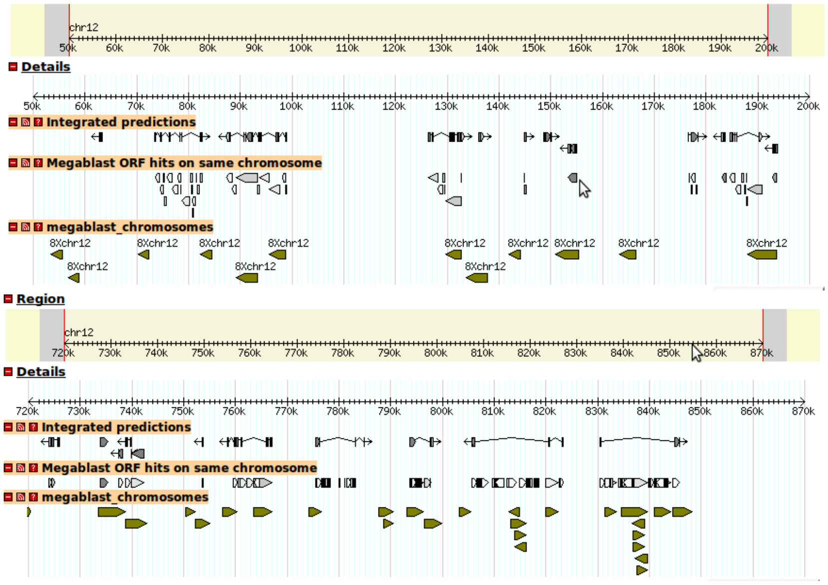

chr12
50k 60k 70k 80k 90k 100k 110k 120k 130k 140k 150k 160k 170k 180k 190k 200k
Details
Integrated predictions
Megablast ORF hits on same chromosome
megablast_chromosomes
8Xchr12
Region
chr12
720k 730k 740k 750k 760k 770k 780k 790k 800k 810k 820k 830k 840k 850k 860k 870k
Details
Integrated predictions
Megablast ORF hits on same chromosome
megablast_chromosomes

Figure A.7 We can see that full chromosome hits are opposite strand within an inversion (top) and positive strand outside of inversions (bottom).

alignment hits on the diagonals (or inversions).

Conclusion Creating a unique mapping between predicted 8X genes and 12X genes requires consistent gene predictions between the two genome assembly versions. Apparently, the 12X assembly yielded such different sequences and gene predictions that most sister gene pairs detected by the rank-and-select approach are random noise. Only a few thousand pairs are clearly correct matches. These sister gene mappings can transfer previously curated 8X gene functional annotation to a minority of the 12X genes. Due to severe differences between the assemblies caused by low-coverage and linkage mismapping, the remaining 12X sequences must be functionally annotated ”de novo” using the same pipelines used to annotate the 8X genes.

Processing DNA re-sequencing data

The purpose of this project was to investigate putative indel regions in Corvina genome compared to Pinot Nior genome after Illumina resequencing.

Background

A highly homozygous Pinot Noir was sequenced using Sanger technology for average read coverage of 12X. A heterzygous Corvina cultivar was sequenced using Illumina technology and the reads aligned to the 12X Pinot Noir reference genome. Read coverage for the Corvina (CV) reads was compared to read coverage for the Pinot Noir (PN) reads. Coverage was generally equal, but some areas showed much high coverage for CV than PN and others showed little or no coverage for CV. The former are interpreted as possible PN regions that are duplicated in CV. The latter are interpreted as possible deletions from PN to CV, also stated as genomic regions that exist in PN but not CV. Regions were classified as CV Duplications, CV Deletions, or Equal (insignificant read coverage difference). According to the data file defining these regions, there are 747 CV Deletion regions and 156 CV Duplication regions.

Data was visualized using Gbrowse tracks

Four gbrowse tracks were created. The first track plots the read coverage difference values and colors positive (more CV coverage than PN) values green and negative (less CV coverage than PN) red. A second and third track show the regions where the read coverage difference is far enough from zero to be significant in either direction and colored similarly to the plot. A fourth track defines ”Unknown” regions that were nearly but not significantly different from zero coverage difference.

Deletions refuted by Nimblegen chip data

To check the above putative deletion regions, genes which overlap the deletion regions were checked in a current expression atlas study for CV done on the Nimblegen microarray designed based on predicted gene models from the 12X PN genome assembly. The study currently has 20 organ x developmental stage combinations with three biological replicates each. Data was discretized to a matrix of 0’s and 1’s where 1 indicates a gene is expressed higher than two standard deviations above the mean expression of a pool of negative control probes in at least 2 of the 3 replicates. 0 indicates otherwise. If a gene is part of a true deletion, it should not be expressed in any of the 20 organ-stages. Surprisingly, the distribution of "on-conditions" for the putative CV-deleted genes shows that most are actually detected on the microarray (Figure A.8). Only 270 of these putative CV-deleted genes were never detected by the microarray analysis. GBrowse tracks were used to visualize the relationship between the putative Corvina deletions and detected genes in the Fasili et al atlas study (Figure A.9). When plotting these tracks on gbrowse along with the track mentioned above, one can easily see which putative CV deletions are supported or refuted. Interestingly chromosome 1 contains many putative CV deletions and there are no overlapping genes to support or refute them.

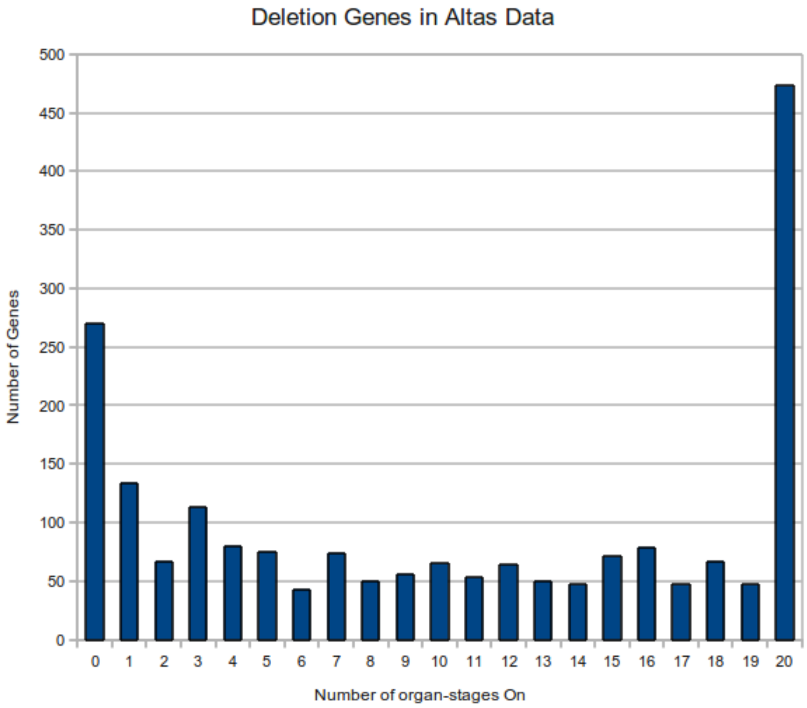


Figure A.8 Corvina genes detected in Fasoli et al atlas.

Functional annotation of the CV Indels

s Fasta sequence file were generated by extracting the PN genome regions defined by the putative CV duplications and deletions respectively. Since there are 156 CV duplication regions, there are 156 sequences in the duplication fasta file. For the deletions regions, only regions were extracted which do no overlap a gene detected in the atlas study. This reduced the 747 putative deletion regions to 208 regions. These 208 regions were extracted from the PN 12X genome and placed in a fasta sequence file. Each of these DNA sequence files were compared to the REFSEQ database of plant proteins using blastx. As a first look at the results, a list was prepared for each class (deletions and duplications) of blastx hits which meet the following criteria:

1. expect value $< 1e - 20$
2. not of Vitis
3. not a "hypothetical protein”
4. not a "predicted protein”
5. not an "ORF”
6. not a Rice locus

Criteria 2-6 remove REFSEQ hits whose names are not meaningful. Many hits were removed this way, but 2048 remain for CV duplication regions and 1237 remain for CV deletion regions. These lists are too long to present here, but we can conduct a functional analysis using these lists with our PathwayFlow web tool (Figure A.10). Using our tool, we find that those genes both missing from and duplicated in Corvina grapes are involved in several pathways known to produce the differences between wines, such as sugars and anthocyanins metabolism (Table C).

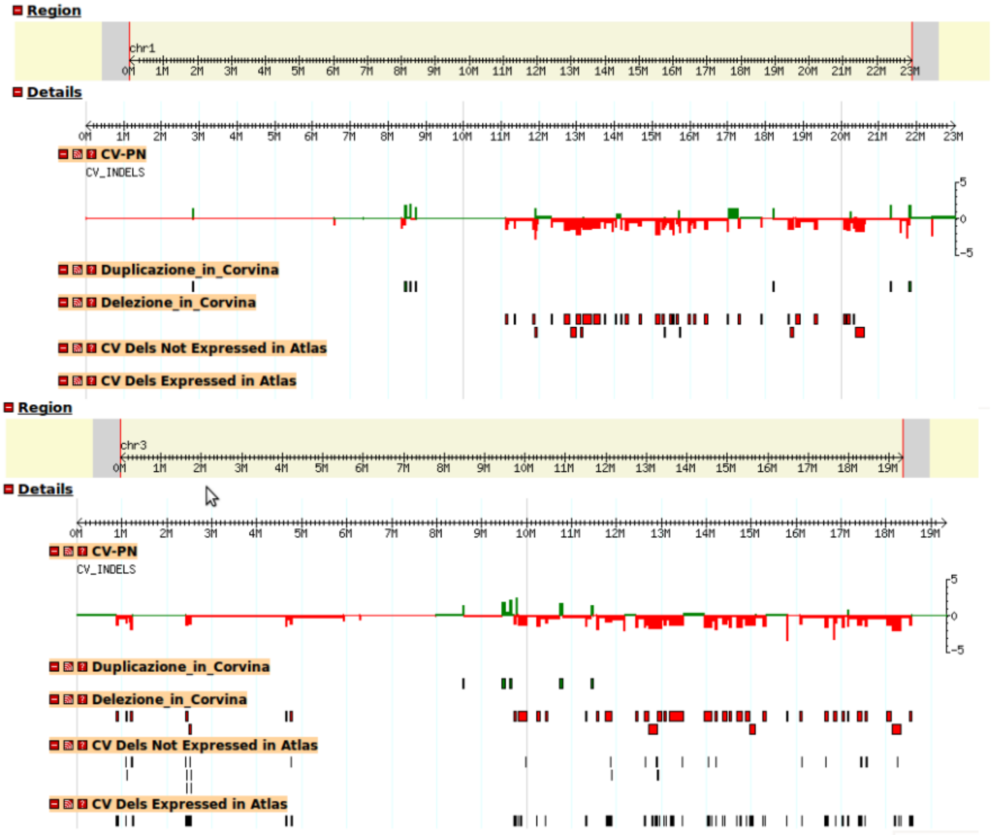

Region
chr1
0M 1M 2M 3M 4M 5M 6M 7M 8M 9M 10M 11M 12M 13M 14M 15M 16M 17M 18M 19M 20M 21M 22M 23M
Details
0M 1M 2M 3M 4M 5M 6M 7M 8M 9M 10M 11M 12M 13M 14M 15M 16M 17M 18M 19M 20M 21M 22M 23M
CV-PN
CV_INDELS
5
0
-5
Duplicazione_in_Corvina
Delezione_in_Corvina
CV Dels Not Expressed in Atlas
CV Dels Expressed in Atlas
Region
chr3
0M 1M 2M 3M 4M 5M 6M 7M 8M 9M 10M 11M 12M 13M 14M 15M 16M 17M 18M 19M
Details
0M 1M 2M 3M 4M 5M 6M 7M 8M 9M 10M 11M 12M 13M 14M 15M 16M 17M 18M 19M
CV-PN
CV_INDELS
5
0
-5
Duplicazione_in_Corvina
Delezione_in_Corvina
CV Dels Not Expressed in Atlas
CV Dels Expressed in Atlas

Figure A.9 GBrowse tracks for Corvina indels. The first defines the genes which overlap putative CV deletions and were not expressed in any of the 20 organ-stages. These genes support the putative CV deletions. The second track defines genes which were detected in at least one organ-stage. These genes are evidence of falsely predicted CV deletions. (Top: Chromosome 1, Bottom: Chromosome 3)

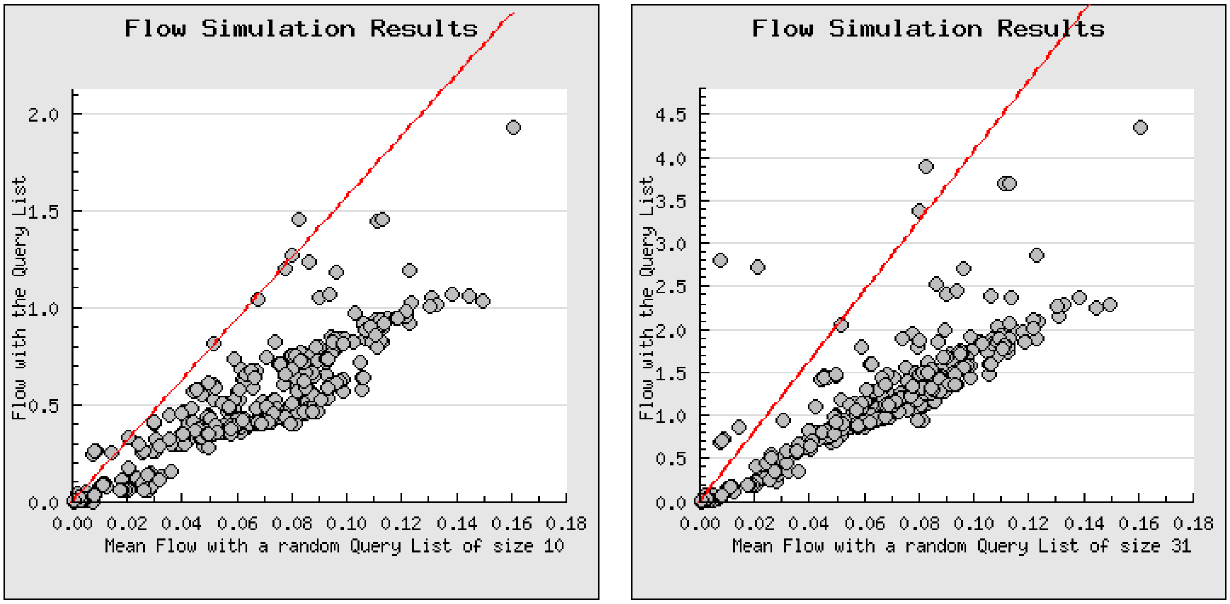


Figure A.10 Forward pathway flow for Corvina deleted genes (left) and duplicated genes in Corvina (right)

APPENDIX B. ANNOTATION DATABASES FOR VITIS

During my work on *Vitis* related projects, necessity dictated central organization of new annotation efforts by the community.

VitisMart: An Annotation BioMart for *Vitis*

Grapevine genetics has made large advances recently. After a draft assembly with 8X average coverage of the Pinot Noir genome was release and published in 2007 (Jaillon et al., 2007), a second draft was developed by the same French-Italian consortium that has average coverage of 12 Sanger reads and computationally predicted gene models on each of the chromosomes. These initial gene models were called ”V0.” A subsequent set of gene models was predicted based not only on sequence, but also on RNAseq data from a simple berry development experiment (Delledonne, 2009).

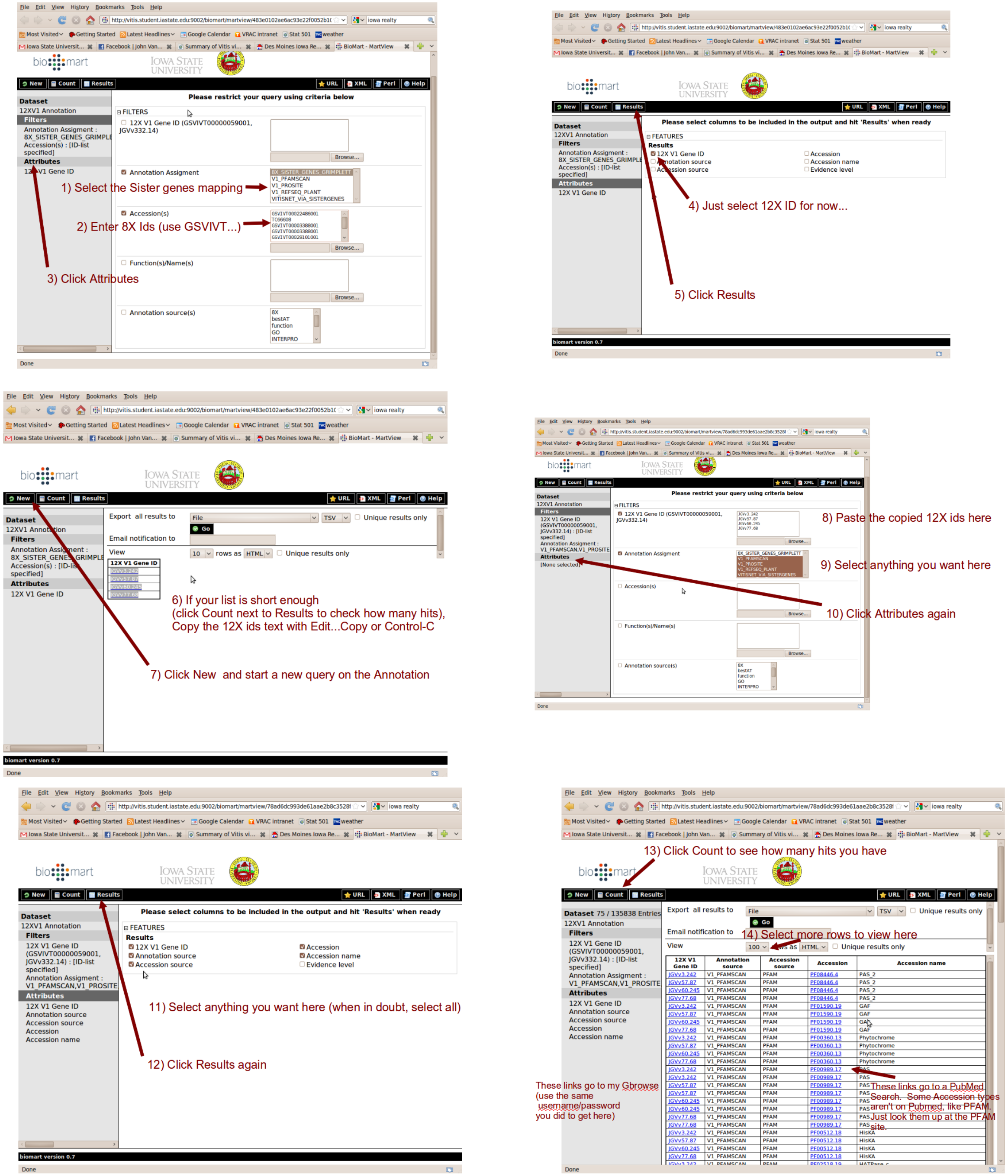
In order to create a central, query-able resource for these and other functional annotations, we created a BioMart (Smedley et al., 2009; Haider et al., 2009) database. BioMart instances are a semi-automatic rearrangement of an existing database schema into an internal, de-normalized form used for a standard web, webservice, and programming interface. Biomarts have a heirarchical organization where Databases contain Datasets and Datasets contain Records, which can be *filtered* based on search criteria to provide lists of matching Records’ *attributes*. We created a simple BioMart containing a single Database, which contains a single Dataset, which contains Records of 12X V1 gene model functional annotation from different sources including a mapping from VitisNet (Grimplet et al., 2009) 8X-based annotation to 12X V1 genes by Jerome Grimplet and the following annotations based on Giorgio Valle’s lab’s annotation pipline on 12X V1 gene models:

1. Gene Ontology terms
2. RefSeq (plants) proteins
3. Pfam domains
4. Prosite domains
5. UNIPROT proteins

A use case for mapping 8X genes to 12X V1 genes is presented in Figure B.1. The results of BioMart query link to corresponding genes on our GBrowse server, which houses all tracks used for this and other projects (Figure B.2).

VitiCyc: A Pathway Genome Database for *Vitis*

While the Plant Metabolomics Network (Lysenko et al., 2009) hosts a BioCyc database for *Vitis*, it is not comprehensive and lacks the 12X V1 genomic data that remained private and unpublished during this work. We created a Pathway Genome Database (PGDB) using the PathwayTools software (Krummenacker et al., 2005), which takes as input a diverse set of specially-formatted gene structure and annotation files and outputs a fully integrated PGDB, which is web-browsable and integrated into all of our tools and information systems. We used the 12X assembly with the V1 gene structure predictions with the fine-tuned functional annotation from Giorgio Valle's laboratory in Padova, Italy plus publicly available annotation of the *Vitis vinifera* chloroplast and mitochondrial genomes. The next few figures include screenshot from a VitiCyc web browsing session displaying general statistics, a section of chromosome 1, and an example pathway (Figures B.3, B.4, B.5).



1) Select the Sister genes mapping

2) Enter 8X Ids (use GSVIVT...)

3) Click Attributes

4) Just select 12X ID for now...

5) Click Results

6) If your list is short enough
(click Count next to Results to check how many hits),
Copy the 12X ids text with Edit...Copy or Control-C

7) Click New and start a new query on the Annotation

8) Paste the copied 12X ids here

9) Select anything you want here

10) Click Attributes again

11) Select anything you want here (when in doubt, select all)

12) Click Results again

13) Click Count to see how many hits you have

14) Select more rows to view here

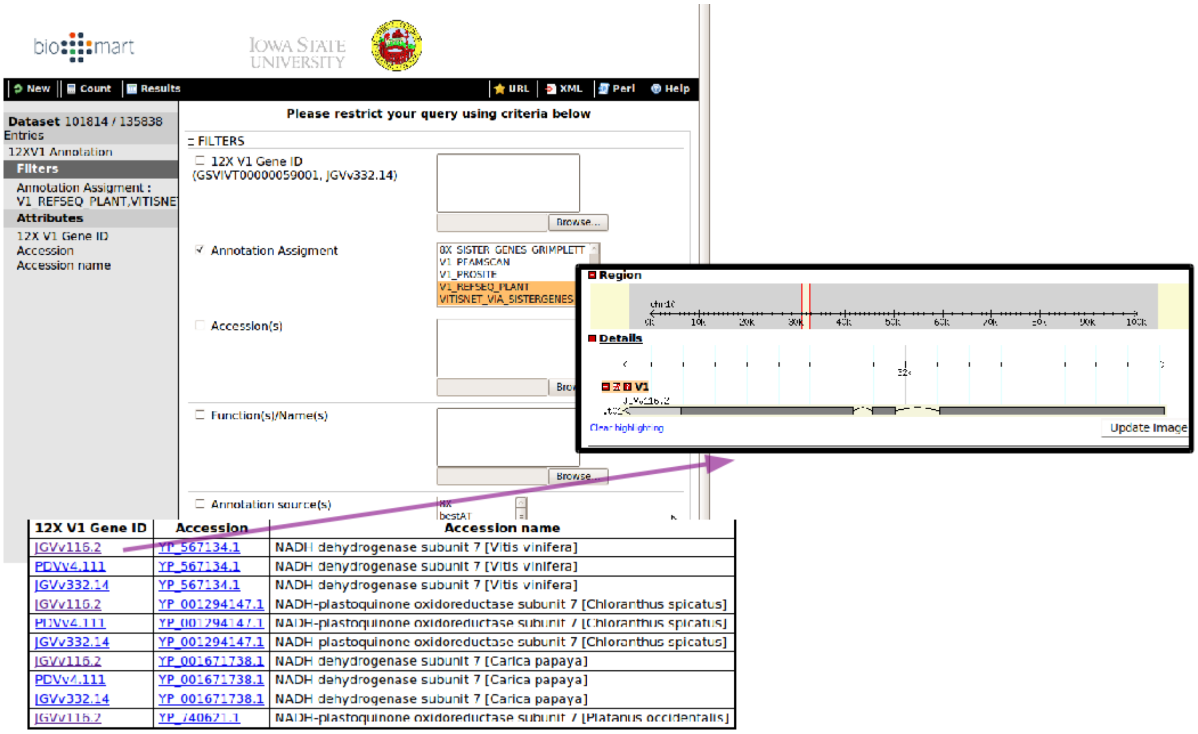
These links go to my Gbrowse (use the same username/password you did to get here)

These links go to a PubMed Search. Some Accession types aren't on Pubmed, like PFAM. Just look them up at the PFAM site.

Dataset 75 / 135838 Entries

12X V1 Gene ID	Annotation source	Accession source	Accession	Accession name
JGVv3.242	V1_PFAMSCAN	PFAM	PF08446.4	PAS_2
JGVv57.87	V1_PFAMSCAN	PFAM	PF08446.4	PAS_2
JGVv60.245	V1_PFAMSCAN	PFAM	PF08446.4	PAS_2
JGVv77.68	V1_PFAMSCAN	PFAM	PF08446.4	PAS_2
JGVv3.242	V1_PFAMSCAN	PFAM	PF01590.19	GAF
JGVv57.87	V1_PFAMSCAN	PFAM	PF01590.19	GAF
JGVv60.245	V1_PFAMSCAN	PFAM	PF01590.19	GAF
JGVv77.68	V1_PFAMSCAN	PFAM	PF01590.19	GAF
JGVv3.242	V1_PFAMSCAN	PFAM	PF00360.13	Phytochrome
JGVv57.87	V1_PFAMSCAN	PFAM	PF00360.13	Phytochrome
JGVv60.245	V1_PFAMSCAN	PFAM	PF00360.13	Phytochrome
JGVv77.68	V1_PFAMSCAN	PFAM	PF00360.13	Phytochrome
JGVv3.242	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv3.242	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv57.87	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv57.87	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv60.245	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv60.245	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv77.68	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv77.68	V1_PFAMSCAN	PFAM	PF00989.17	PAS
JGVv3.242	V1_PFAMSCAN	PFAM	PF00512.18	HisKA
JGVv57.87	V1_PFAMSCAN	PFAM	PF00512.18	HisKA
JGVv60.245	V1_PFAMSCAN	PFAM	PF00512.18	HisKA
JGVv77.68	V1_PFAMSCAN	PFAM	PF00512.18	HisKA
JGVv3.242	V1_PFAMSCAN	PFAM	PF02518.19	HATPase_c

Figure B.1 A VitisMart use case.



Dataset 101814 / 135838 Entries
12XV1 Annotation

Filters

Annotation Assigment : V1_REFSEQ_PLANT,VITISNET

Attributes

12X V1 Gene ID
Accession
Accession name

Please restrict your query using criteria below

FILTERS

□ 12X V1 Gene ID (GSVIVT00000059001, JGVv332.14)

☑ Annotation Assigment: 8X_SISTER_GENES_GRIMPLETT, V1_PFAMSCAN, V1_PROSITE, V1_REFSEQ_PLANT, VITISNET_VIA_SISTERGENES

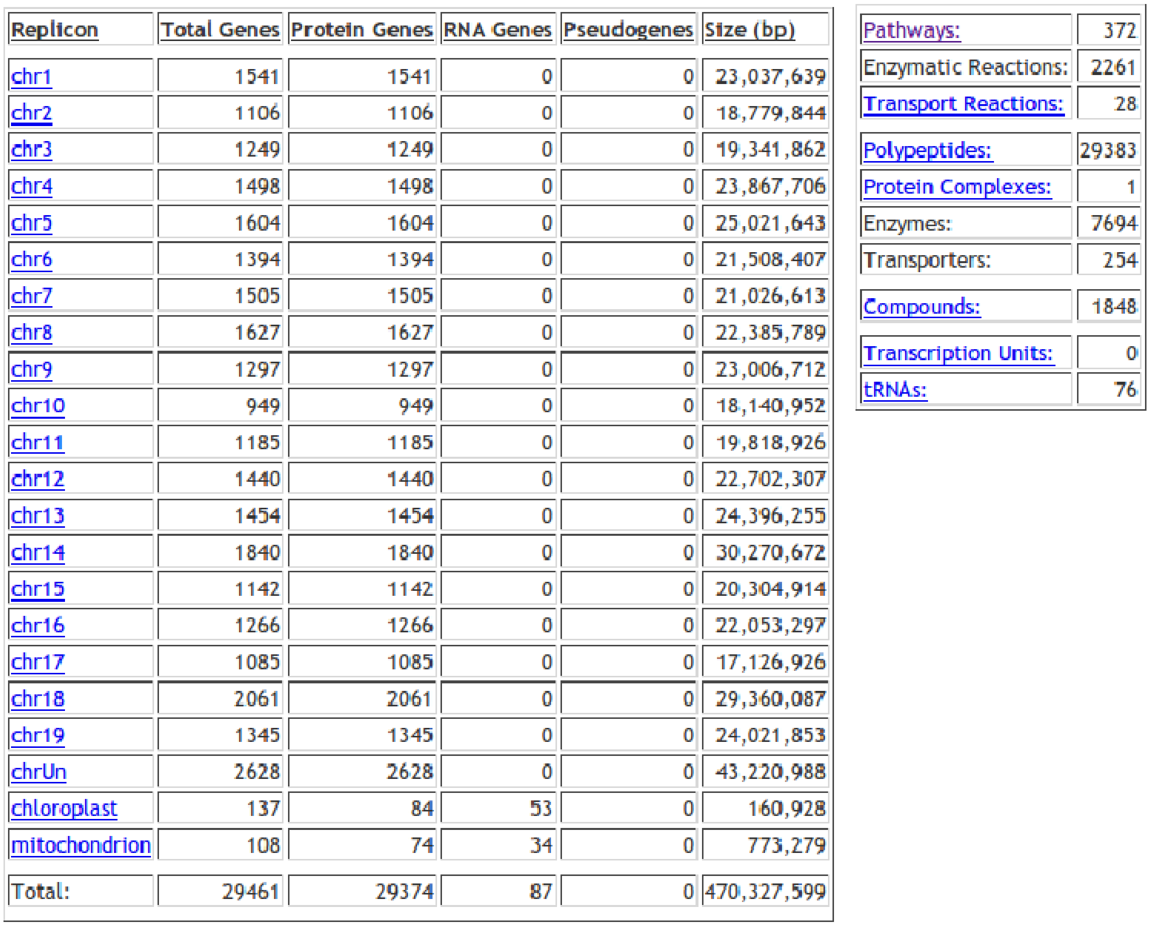
□ Accession(s)

□ Function(s)/Name(s)

□ Annotation source(s): 8X, bestAT

12X V1 Gene ID	Accession	Accession name
JGVv116.2	YP_567134.1	NADH dehydrogenase subunit 7 [Vitis vinifera]
PDVv4.111	YP_567134.1	NADH dehydrogenase subunit 7 [Vitis vinifera]
JGVv332.14	YP_567134.1	NADH dehydrogenase subunit 7 [Vitis vinifera]
JGVv116.2	YP_001294147.1	NADH-plastoquinone oxidoreductase subunit 7 [Chloranthus spicatus]
PDVv4.111	YP_001294147.1	NADH-plastoquinone oxidoreductase subunit 7 [Chloranthus spicatus]
JGVv332.14	YP_001294147.1	NADH plastoquinone oxidoreductase subunit 7 [Chloranthus spicatus]
JGVv116.2	YP_001671738.1	NADH dehydrogenase subunit 7 [Carica papaya]
PDVv4.111	YP_001671738.1	NADH dehydrogenase subunit 7 [Carica papaya]
JGVv332.14	YP_001671738.1	NADH dehydrogenase subunit 7 [Carica papaya]
JGVv116.2	YP_740621.1	NADH-plastoquinone oxidoreductase subunit 7 [Platanus occidentalis]

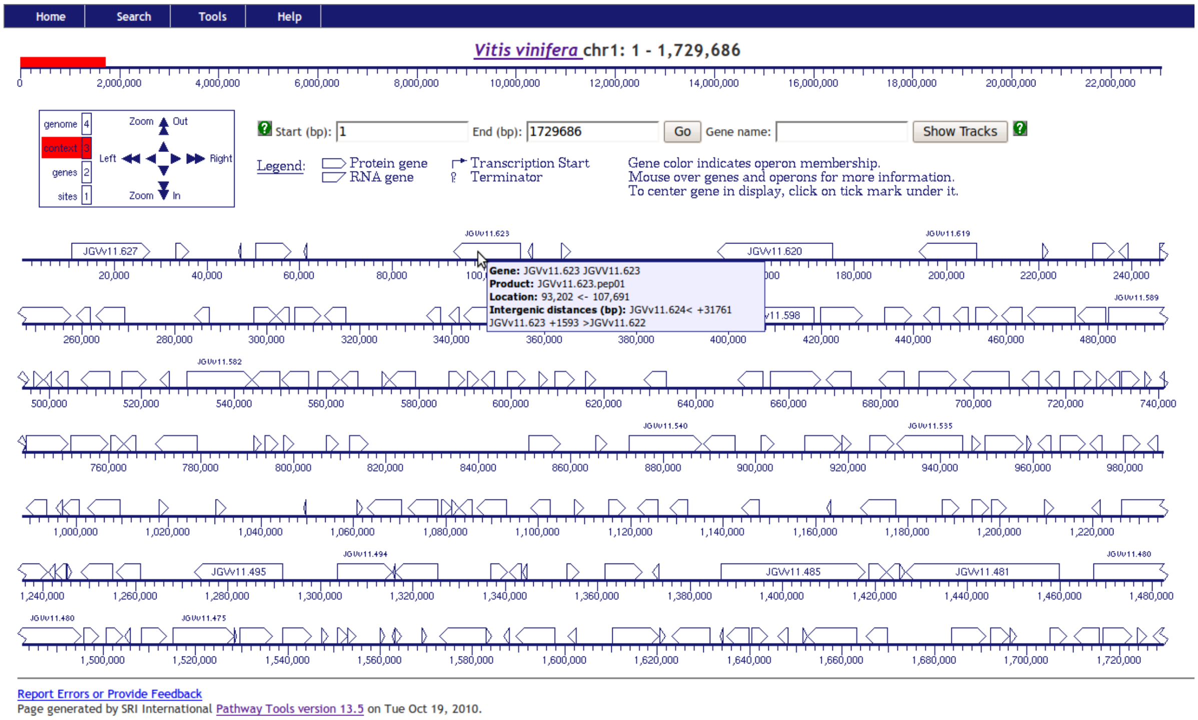
Figure B.2 VitisMart is integrated with our GBrowse server.



Replicon	Total Genes	Protein Genes	RNA Genes	Pseudogenes	Size (bp)
chr1	1541	1541	0	0	23,037,639
chr2	1106	1106	0	0	18,779,844
chr3	1249	1249	0	0	19,341,862
chr4	1498	1498	0	0	23,867,706
chr5	1604	1604	0	0	25,021,643
chr6	1394	1394	0	0	21,508,407
chr7	1505	1505	0	0	21,026,613
chr8	1627	1627	0	0	22,385,789
chr9	1297	1297	0	0	23,006,712
chr10	949	949	0	0	18,140,952
chr11	1185	1185	0	0	19,818,926
chr12	1440	1440	0	0	22,702,307
chr13	1454	1454	0	0	24,396,255
chr14	1840	1840	0	0	30,270,672
chr15	1142	1142	0	0	20,304,914
chr16	1266	1266	0	0	22,053,297
chr17	1085	1085	0	0	17,126,926
chr18	2061	2061	0	0	29,360,087
chr19	1345	1345	0	0	24,021,853
chrUn	2628	2628	0	0	43,220,988
chloroplast	137	84	53	0	160,928
mitochondrion	108	74	34	0	773,279
Total:	29461	29374	87	0	470,327,599

Pathways:	372
Enzymatic Reactions:	2261
Transport Reactions:	28
Polypeptides:	29383
Protein Complexes:	1
Enzymes:	7694
Transporters:	254
Compounds:	1848
Transcription Units:	0
tRNAs:	76

Figure B.3 VitisCyc summary statistics.



Vitis vinifera chr1: 1 - 1,729,686

Start (bp): 1 End (bp): 1729686 Gene name:

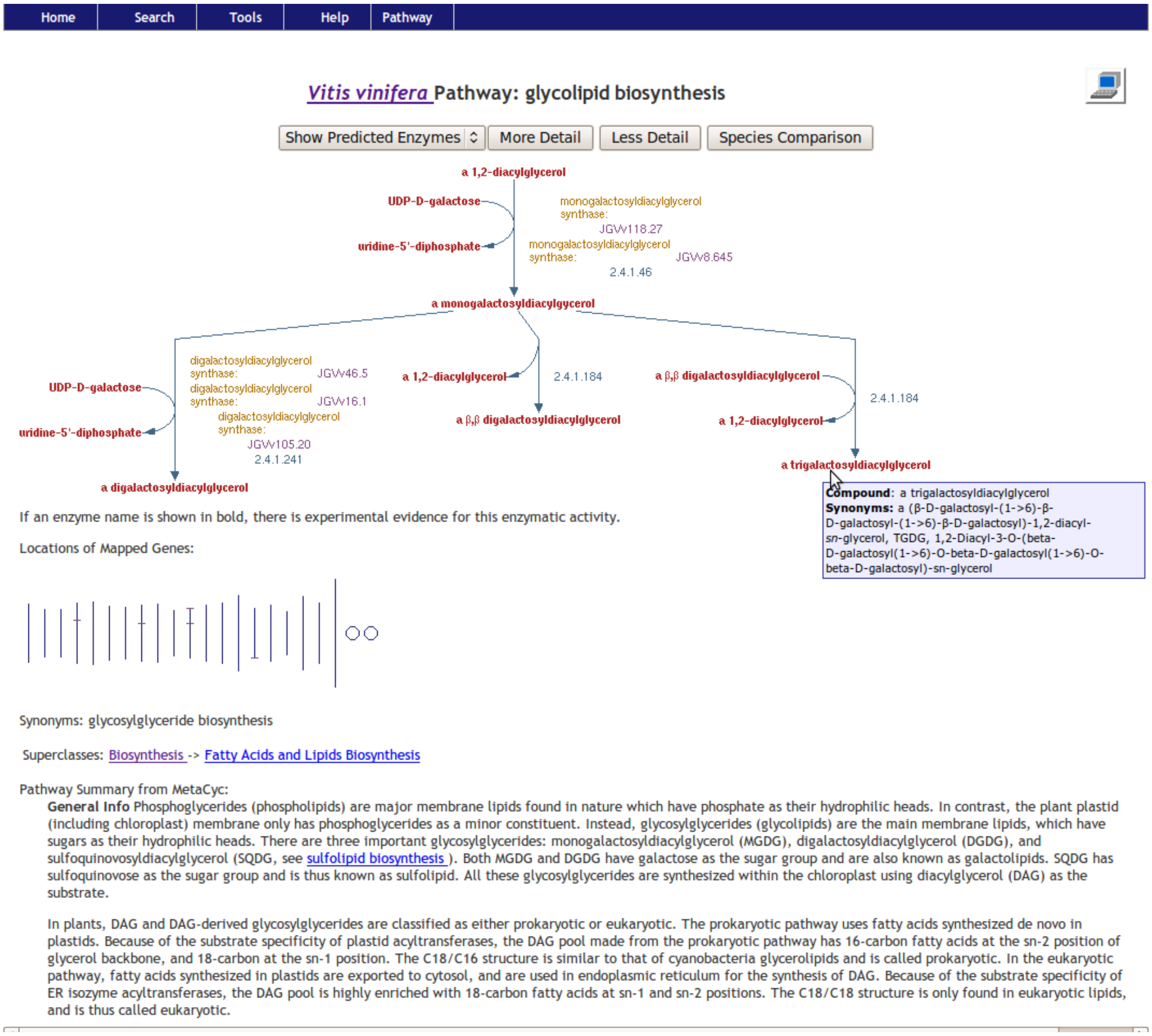
Legend: Protein gene RNA gene Transcription Start Terminator

Gene color indicates operon membership.
Mouse over genes and operons for more information.
To center gene in display, click on tick mark under it.

Gene: JGVv11.623 JGVV11.623
Product: JGVv11.623.pep01
Location: 93,202 <- 107,691
Intergenic distances (bp): JGVv11.624< +31761
JGVv11.623 +1593 >JGVv11.622

Report Errors or Provide Feedback
Page generated by SRI International Pathway Tools version 13.5 on Tue Oct 19, 2010.

Figure B.4 Chromosome 1 in VitiCyc.


Home
Search
Tools
Help
Pathway
Show Predicted Enzymes
More Detail
Less Detail
Species Comparison
a 1,2-diacylglycerol
UDP-D-galactose
uridine-5'-diphosphate
monogalactosyldiacylglycerol synthase:
JGVV118.27
monogalactosyldiacylglycerol synthase:
JGVV8.645
2.4.1.46
a monogalactosyldiacylglycerol
UDP-D-galactose
uridine-5'-diphosphate
digalactosyldiacylglycerol synthase:
JGVV46.5
digalactosyldiacylglycerol synthase:
JGVV16.1
digalactosyldiacylglycerol synthase:
JGVV105.20
2.4.1.241
a digalactosyldiacylglycerol
a 1,2-diacylglycerol
2.4.1.184
a β,β digalactosyldiacylglycerol
a β,β digalactosyldiacylglycerol
2.4.1.184
a 1,2-diacylglycerol
a trigalactosyldiacylglycerol

Vitis vinifera Pathway: glycolipid biosynthesis

Compound: a trigalactosyldiacylglycerol
Synonyms: a (β-D-galactosyl-(1->6)-β-D-galactosyl-(1->6)-β-D-galactosyl)-1,2-diacyl-*sn*-glycerol, TGDG, 1,2-Diacyl-3-O-(beta-D-galactosyl(1->6)-O-beta-D-galactosyl(1->6)-O-beta-D-galactosyl)-sn-glycerol

If an enzyme name is shown in bold, there is experimental evidence for this enzymatic activity.

Locations of Mapped Genes:

Synonyms: glycosylglyceride biosynthesis

Superclasses: Biosynthesis -> Fatty Acids and Lipids Biosynthesis

Pathway Summary from MetaCyc:

General Info Phosphoglycerides (phospholipids) are major membrane lipids found in nature which have phosphate as their hydrophilic heads. In contrast, the plant plastid (including chloroplast) membrane only has phosphoglycerides as a minor constituent. Instead, glycosylglycerides (glycolipids) are the main membrane lipids, which have sugars as their hydrophilic heads. There are three important glycosylglycerides: monogalactosyldiacylglycerol (MGDG), digalactosyldiacylglycerol (DGDG), and sulfoquinovosyldiacylglycerol (SQDG, see sulfolipid biosynthesis). Both MGDG and DGDG have galactose as the sugar group and are also known as galactolipids. SQDG has sulfoquinovose as the sugar group and is thus known as sulfolipid. All these glycosylglycerides are synthesized within the chloroplast using diacylglycerol (DAG) as the substrate.

In plants, DAG and DAG-derived glycosylglycerides are classified as either prokaryotic or eukaryotic. The prokaryotic pathway uses fatty acids synthesized de novo in plastids. Because of the substrate specificity of plastid acyltransferases, the DAG pool made from the prokaryotic pathway has 16-carbon fatty acids at the sn-2 position of glycerol backbone, and 18-carbon at the sn-1 position. The C18/C16 structure is similar to that of cyanobacteria glycerolipids and is called prokaryotic. In the eukaryotic pathway, fatty acids synthesized in plastids are exported to cytosol, and are used in endoplasmic reticulum for the synthesis of DAG. Because of the substrate specificity of ER isozyme acyltransferases, the DAG pool is highly enriched with 18-carbon fatty acids at sn-1 and sn-2 positions. The C18/C18 structure is only found in eukaryotic lipids, and is thus called eukaryotic.

Figure B.5 The glycolipid biosynthesis pathway in VitiCyc.

APPENDIX C. LARGE TABLES AND FIGURES

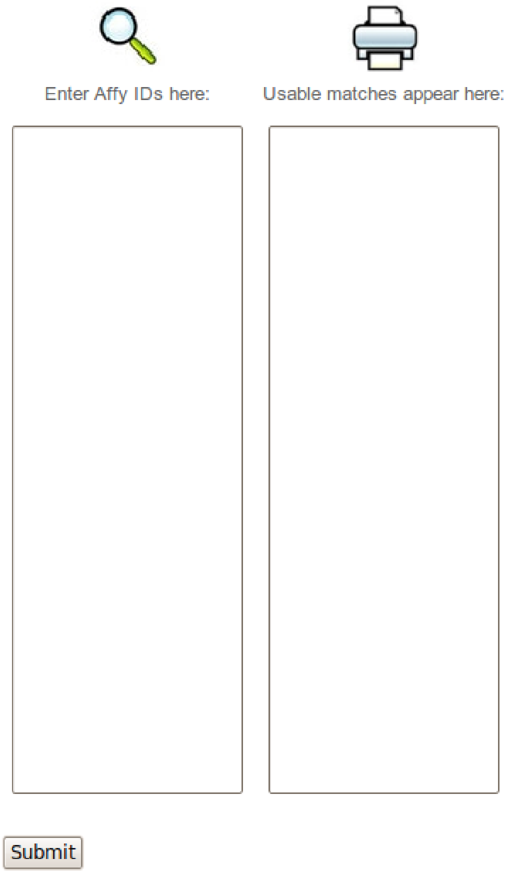


Figure C.1 Enter your IDs in the box on the left to receive the corresponding BioCyc IDs in the box on the right if they exist.

Table C.1 Icons and buttons on the web tool.

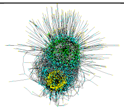
🔍	*Query List.* This represents the Query List.
⚗	*Response Groups.* This represents the Response Groups.
	*Pathway Network.* This represents the selected pathway network.
🔍→⚗	*Query List* → *Response Groups.* This represents flow from the Query List to the Response Groups.
🔍←⚗	*Query List* ← *Response Groups.* This represents flow from the Response Groups to the Query List.
🔍↔⚗	*Query List* ←→ *Response Groups.* This represents the sum of flow between the Query List and Response Groups.
🔍	*Lookup IDs.* This button takes you to the BioCyc IDs of your query list.
📂	*Results Files.* This button takes you to a directory containing output files from your analysis.
☆	*Hard Link.* This button provides a hard link that can be bookmarked and saved for loading the same analysis at a later time.
⛑	*Quality Control.* This button takes you to a directory containing Erlang distribution assessments for a random sample of response groups.
⬇	*Download PDF.* This button downloads a PDF of the three plots along with a Venn diagram of the significant response groups in each.
⚗	*Response Groups IDs.* This button downloads a list of the significant response groups' BioCyc IDs.

Table C.2 Blue-marked reactions in Figure 4.8. Descriptions are copied directly from EcoCyc.

CDP-diacylglycerol–glycerol-3-phosphate 3-phosphatidyltransferase	This reaction is at a branch point and is the committed step in the biosynthesis of the acidic phospholipids.
Acyl-[acyl-carrier-protein]–UDP-N-acetylglucosamine O-acyltransferase	This is the first unique reaction in the biosynthesis of the phosphorylated glycolipidlipid A in the outer membrane of E.coli.
Arabinose-5-phosphate isomerase	This reaction interconverts D-arabinose-5-phosphate and D-ribulose-5-phosphate. D-arabinose-5-phosphate is a precursor to KDO.
NARPQ-RXN	In this reaction nitrate/nitrite response regulator NarP is phosphorylated by phospho-NarQ sensor protein.
1-acylglycerol-3-phosphate O-acyltransferase	This is the second step in de novo phospholipid biosynthesis in which a second fatty acid is esterified at C2 of the glycerol moiety.
CDP-diacylglycerol–serine O-phosphatidyltransferase	This reaction is at a branch point and is the committed step for the synthesis of phosphatidylethanolamine.
BAROMP-RXN	In this reaction the trancriptional regulatory protein OmpR is phosphorylated by transphosphorylation from the phospho-BarA protein.
D-galacturonate isomerase	Part of the galacturonate pathway.
Alanine racemase	No summary
UHPA-RXN	In this reaction the transcriptional regulatory protein UhpA is phosphorylated by phospho-UhpB proteinthereby becoming activated.
Tagatose-bisphosphate aldolase	This reaction is part of the galactitol and N-acetylgalactosamine catabolism pathways.
Glucuronate isomerase	Part of the glucuronate pathway.

Table C.2 (Continued)

4-α-glucanotransferase	This reaction synthesizes maltodextrins of longer length.
Adenylosuccinate lyase	This is the eighth step in the de novo purine biosynthesis.
Chorismate mutase	This reaction is the first step after the chorismate branch point in the biosynthesis of both phenylalanine and tyrosine.
Diaminopimelate epimerase	This is the sixth and last step in the synthesis of diaminopimelateand the penultimate step in the synthesis of lysine.
NARPX-RXN	In the presence of both nitrate and nitritenitrate/nitrite response regulator NarP is phosphorylated by the sensor kinase-phosphotransferase phospho-NarX.
trans-2-decenoyl-[acyl-carrier-protein]isomerase	No summary
RXN0-947	No summary
OMPR-RXN	In this reaction the transcriptional regulatory protein OmpR is phosphorylated by transphosphorylation from osmolarity sensor kinase-phosphotransferase EnvZ. OmpR regulates transcription of the membrane porin genes ompC and ompF.
acyl-ACP:sn-glycerol-3-phosphate 1-O-acyltransferase	No summary
palmitoleoyl[acyl-carrier-protein]-dependent acyltransferase	No summary
Threonine aldolase	No summary
Glucose dehydrogenasepyrroloquinoline-quinone	Involved in the transfer of electrons from the non-phosphorylated aldose sugars to the electron transport chain. Aldose sugar dehydrogenase is associated with the outer membrane under certain physiological conditions while glucose dehydrogenase localizes to the inner membrane.

Table C.3 Blue-marked reactions in Figure 4.10. Descriptions are copied directly from EcoCyc.

NARLQ-RXN	In this reaction nitrate/nitrite response regulator NarL is phosphorylated by the phospho-NarQ sensor protein.
PHOBCREC-RXN	In this reaction the phosphate regulon transcriptional regulatory protein PhoB is phosphorylated by the sensor kinase-phosphotransferase CreC-his-P.
NARPQ-RXN	In this reaction nitrate/nitrite response regulator NarP is phosphorylated by phospho-NarQ sensor protein.
ARCBTRANS-RXN	In this reaction the sensor kinase-phosphotransferase ArcB undergoes intramolecular transphosphorylation. The phosphoryl group is transferred from the histidine residue H292 to an aspartate residue.
PHOBR-RXN	In this reaction the phosphate regulon transcriptional regulatory protein PhoB is phosphorylated by phosphate regulon sensor kinase-phosphotransferase PhoR-his-P.
NARPX-RXN	In the presence of both nitrate and nitritenitrate/nitrite response regulator NarP is phosphorylated by the sensor kinase-phosphotransferase phospho-NarX.
ALTARCA-RXN	In this reaction respiration control protein ArcA is phosphorylated by sensor kinase-phosphotransferase ArcB-his717-P. This is an alternative route to ArcA activation.

Table C.3 (Continued)

ARCA-RXN	In this reaction the respiration control protein ArcA is phosphorylated by the sensor kinase-phosphotransferase ArcB-his292-P protein. This is an alternative route to ArcA activation.
Alanine racemase	No summary
CREB-RXN	In this reaction the transcriptional regulatory protein CreB is phosphorylated by sensor kinase-phosphotransferase CreC-his-P.
ARCB717-RXN	In this reaction sensor kinase-phosphotransferase ArcB undergoes intramolecular transphosphorylation.
NARLX-RXN	In the presence of nitratenitrate/nitrite response regulator NarL is phosphorylated by the sensor kinase-phosphotransferase NarX. In the presence of nitritethe NarL protein is dephosphorylated with the sensor protein NarX acting as a phospho-NarL phosphatase. In the presence of nitrate the nitrate/nitrite response regulator NarL can be phosphorylated by both nitrate/nitrite sensor kinase-phosphotransferases phospho-NarX and phospho-NarQ. In this activated state phospho-NarL can act as both an activator of nitrate and nitrite reductase transcription and repressor of fumarate reductase transcription. Both actions switch anaerobic respiration to utilization of either nitrate or nitrite as electron acceptors. In the presence of nitrite the NarL protein is phosphorylated only by phospho-NarQ. The sensor protein NarX acts as a phospho-NarL phosphatase under these conditions countering the sensor protein NarQ phosphorylation to some extent.

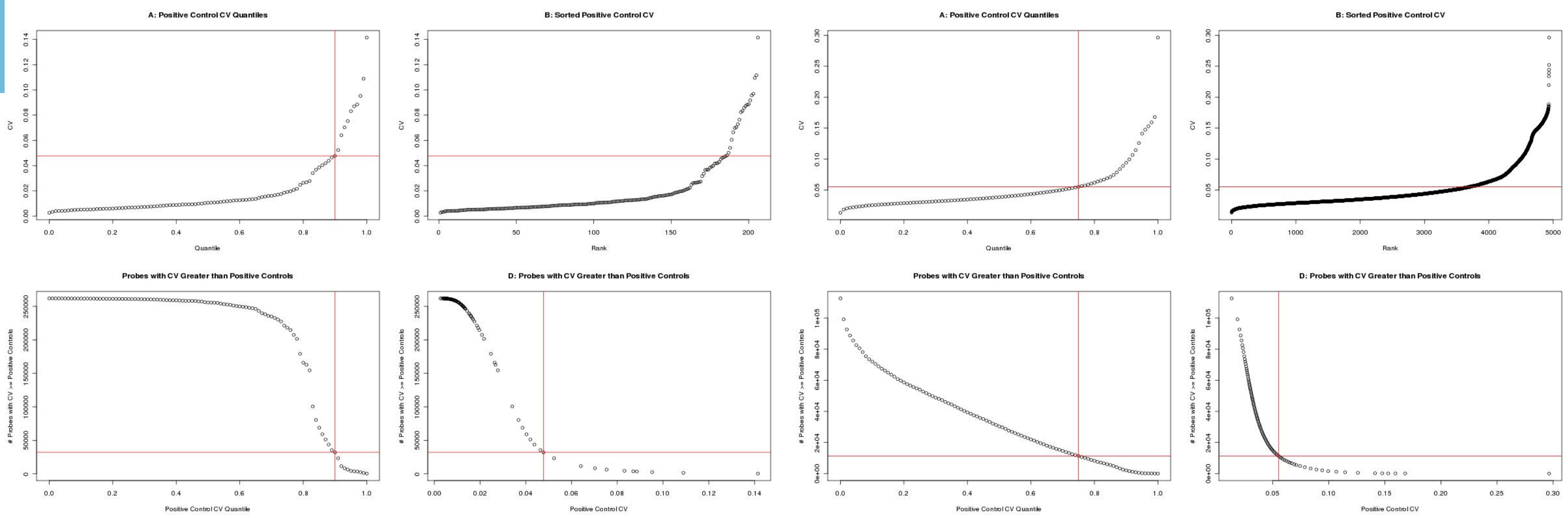


Figure C.2 Plots for selecting perturbed probes from the data.

Table C.4 Overrepresented Annotation: Cluster 1

Biological Process							
GO-ID	p-value	corr p-value	Hits in subcluster	Hits in ontology	Subcluster size	Ontology size	Description
6869	2.4913E-5	1.8435E-3	4	35	23	4479	lipid transport
5975	1.5139E-3	5.6014E-2	7	356	23	4479	carbohydrate metabolic process
Molecular Function							
4553	1.1464E-5	3.3761E-4	6	167	19	5697	hydrolase activity, hydrolyzing O-glycosyl compounds
16798	1.2277E-5	3.3761E-4	6	169	19	5697	hydrolase activity, acting on glycosyl bonds
4650	4.6677E-4	8.5575E-3	2	10	19	5697	polygalacturonase activity
Cellular Component							
16021	5.9884E-3	4.4743E-2	5	530	7	2504	integral to membrane
31224	7.4572E-3	4.4743E-2	5	556	7	2504	intrinsic to membrane
44425	1.4696E-2	4.5198E-2	5	646	7	2504	membrane part
16020	1.5066E-2	4.5198E-2	6	963	7	2504	membrane

Table C.5 Successor pathways with high flow from Cluster 1 genes. Discriminated at the 95% confidence level with Bonferronni correction.

VitiCyc Pathway	Description from MetaCyc
HGA biosynthesis and degradation	Homogalacturonan (HGA) is a linear chain of 14-linked a-D-galacturonic acid residues in which some of the carboxyl groups are methyl esterified. HGA accounts for up to 60 percent of the pectin found in primary cell walls of all plants.
rutin biosynthesis	Rutin is a flavonol glucoside that is widespread in the plant kingdom and has been especially found in the Polygonaceae family. Rutin has several interesting pharmacological properties (e.g. antioxidative activity) that have been exploited in human medicine and nutrition. Quercetin-3-rhamnoside and its rutinosidei.e. rutin has been demonstrated to possess antimalarial activity which is an ubiquitous activated nucleotide sugar readily available for metabolic processes in plants.
DIMBOA-glucoside degradation	Cyclic hydroxamic acids and particularly their aglycone forms have been reported to be involved in the defense of plants against a wide range of pathogens and insects. The aglycone forms are autotoxic and are generally found stored in plants in their 2-O-β-D-glucoside form. The toxic aglycone is enzymatically released by the action of β-glucosidases the origin of which can be from the compromised plant or the pathogen itself.
glycolipid biosynthesis	Phosphoglycerides (phospholipids) are major membrane lipids found in nature which have phosphate as their hydrophilic heads. In contrast the plant plastid (including chloroplast) membrane only has phosphoglycerides as a minor constituent. Instead glycosylglycerides (glycolipids) are the main membrane lipids which have sugars as their hydrophilic heads.
cyclopropane fatty acid (CFA) biosynthesis	Cyclopropane fatty acid (CFA) synthase catalyzes a modification of the acyl chains of phospholipid bilayers through methylenation (using S-adenosyl-L-methionine) of unsaturated fatty acyl chains to their cyclopropane derivatives.

Table C.6 Overrepresented Annotation: Cluster 3

Biological Process							
GO-ID	p-value	corr p-value	Hits in subcluster	Hits in ontology	Subcluster size	Ontology size	Description
15979	1.8937E-17	2.2157E-15	14	58	41	4479	photosynthesis
9765	5.2605E-17	3.0774E-15	10	18	41	4479	photosynthesis, light harvesting
19684	2.2698E-15	8.8524E-14	10	24	41	4479	photosynthesis, light reaction
6091	2.6224E-8	7.6706E-7	10	108	41	4479	generation of precursor metabolites and energy
Molecular Function							
8943	2.3112E-4	9.1293E-3	2	4	36	5697	glyceraldehyde-3-phosphate dehydrogenase activity
4365	2.3112E-4	9.1293E-3	2	4	36	5697	glyceraldehyde-3-phosphate dehydrogenase (phosphorylating) activity
50662	1.2296E-3	3.2379E-2	6	195	36	5697	coenzyme binding
16564	2.4627E-3	4.8638E-2	2	12	36	5697	transcription repressor activity
48037	4.5800E-3	7.2365E-2	6	253	36	5697	cofactor binding
30528	7.3537E-3	8.5331E-2	6	279	36	5697	transcription regulator activity
16620	7.5609E-3	8.5331E-2	2	21	36	5697	oxidoreductase activity, acting on the aldehyde or oxo group of donors, NAD or NADP as acceptor
10181	9.0373E-3	8.9243E-2	2	23	36	5697	FMN binding
16491	1.1687E-2	9.7495E-2	9	611	36	5697	oxidoreductase activity
16903	1.2341E-2	9.7495E-2	2	27	36	5697	oxidoreductase activity, acting on the aldehyde or oxo group of donors

Table C.6 (Continued)

Cellular Component							
9536	6.7225E-14	1.3781E-12	12	52	29	2504	plastid
9507	6.7225E-14	1.3781E-12	12	52	29	2504	chloroplast
30090	9.2008E-5	1.2574E-3	4	22	29	2504	photosystem
34357	1.8210E-4	1.8665E-3	4	26	29	2504	photosynthetic membrane
43227	3.7285E-4	3.0574E-3	19	836	29	2504	membrane-bounded organelle
9523	5.7917E-4	3.9576E-3	3	15	29	2504	photosystem II
43231	1.2657E-3	7.4133E-3	18	831	29	2504	intracellular membrane-bounded organelle
44444	1.5818E-3	8.1068E-3	14	554	29	2504	cytoplasmic part
16020	2.1375E-2	9.4437E-2	17	963	29	2504	membrane
9538	2.3033E-2	9.4437E-2	1	2	29	2504	photosystem I reaction center
43229	2.8748E-2	9.5990E-2	19	1160	29	2504	intracellular organelle
43226	2.8748E-2	9.5990E-2	19	1160	29	2504	organelle
5737	3.0436E-2	9.5990E-2	14	757	29	2504	cytoplasm

Table C.7 Successor pathways with high flow from Cluster 3 genes. Discriminated at the 99% confidence level without Bonferronni correction.

glycolysis I, II, IV	In plants glycolysis is the predominant pathway fueling respiration.
aspartate degradation II	In this eukaryotic route of aspartate degradationaspartate is converted to malate as part of the reversible malate-aspartate shuttle. This pathway spans the mitochondrial and cytoplasmic spaces transferring reducing equivalents across the mitochondrial membrane. It is one of several shuttle mechanisms used to transfer electrons from cytosolic NADH produced by glycolysis into the mitochondrionbecause NADH itself cannot cross the inner mitochondrial membrane.
glycolysis III (Thermotoga)	A mis-prediction in VitiCyc, but energy-related nonetheless.
formaldehyde assimilation III (dihydroxyacetone cycle)	A mis-prediction in VitiCyc, but energy-related nonetheless. This might be mis-predicted in VitiCyc due to its interaction with the pentose phosphate pathway (commone ribulose phophate and fructose-bisphophate rearranging and transketolase enzymes).
13-LOX and 13-HPL pathway	Lipoxygenases (LOX) are ubiquitous enzymes in eukaryotes. Depending on the positional specificities they are further classified as 9-LOX and 13-LOX converting poly unsaturated fatty acids to 9-hydroperoxides and 13-hydroperoxidesrespectively. In plants most reported LOXs are 13-LOXs which is important in the biosynthesis of the plant hormone jasmonic acid. 13-HPL activity has been detected in many plants. The aldehydes and alcohols produced by 13-HPL give the flavors and tastes to plantssuch as the cucumber odor.
pentose phosphate pathway (non-oxidative branch)	The pentose phosphate pathway is one of the three essential pathways of central metabolism. In addition this pathway is an important source of NADPH which is also needed for biosyntheses.
pentose phosphate pathway (partial)	A mis-prediction in VitiCyc, but energy-related nonetheless.
D-mannose degradation	D-mannose can serve as a total source of carbon and energy for growth of *E. coli* and is also used in plants
glycine biosynthesis	No summary
gluconeogenesis	Gluconeogenesis is the generation of glucose from non-sugar carbon substrates.
Calvin-Benson-Bassham cycle	The Calvin cycle is the major CO_2 fixation pathway found in all in green plants and many autotrophic bacteria.

Table C.8 Overrepresented Annotation: Cluster 5

GO-ID	p-value	corr p-value	Hits in subcluster	Hits in ontology	Subcluster size	Ontology size	Description
Molecular Function							
16165	1.8487E-3	7.7648E-2	2	9	42	5697	lipoxygenase activity
3993	2.3001E-3	7.7648E-2	2	10	42	5697	acid phosphatase activity
16701	2.7981E-3	7.7648E-2	2	11	42	5697	oxidoreductase activity, acting on single donors with incorporation of molecular oxygen
16702	2.7981E-3	7.7648E-2	2	11	42	5697	oxidoreductase activity, acting on single donors with incorporation of molecular oxygen, incorporation of two atoms of oxygen
51213	3.9313E-3	8.7276E-2	2	13	42	5697	dioxygenase activity
Cellular Component							
16021	3.7665E-3	5.7440E-2	9	530	17	2504	integral to membrane
5576	3.9496E-3	5.7440E-2	3	49	17	2504	extracellular region
31224	5.3093E-3	5.7440E-2	9	556	17	2504	intrinsic to membrane
16020	7.1999E-3	5.7440E-2	12	963	17	2504	membrane
48046	8.4471E-3	5.7440E-2	2	21	17	2504	apoplast
44425	1.4986E-2	7.6388E-2	9	646	17	2504	membrane part
5618	1.7974E-2	7.6388E-2	2	31	17	2504	cell wall
30312	1.7974E-2	7.6388E-2	2	31	17	2504	external encapsulating structure

Table C.9 Successor pathways with high flow from Cluster 5 genes. Discriminated at the 95% confidence level with Bonferronni correction.

13-LOX and 13-HPL pathway	Lipoxygenases (LOX) are ubiquitous enzymes in eukaryotes. Depending on the positional specificities they are further classified as 9-LOX and 13-LOX converting poly unsaturated fatty acids to 9-hydroperoxides and 13-hydroperoxidesrespectively. In plants most reported LOXs are 13-LOXs which is important in the biosynthesis of the plant hormone jasmonic acid. 13-HPL activity has been detected in many plants. The aldehydes and alcohols produced by 13-HPL give the flavors and tastes to plantssuch as the cucumber odor.
pentose phosphate pathway (non-oxidative branch)	The pentose phosphate pathway is one of the three essential pathways of central metabolism. In addition this pathway is an important source of NADPH which is also needed for biosyntheses.
pentose phosphate pathway (partial)	A mis-prediction in VitiCyc, but energy-related nonetheless.
sorbitol degradation I	Sorbitol is a hexitolone of several acyclic polyols or sugar alcohols found in higher plants. The translocated sorbitol is ’unloaded’ in fruit tissues and eventually converted to fructose *via* a NAD$^+$-dependent sorbitol dehydrogenase which is a key enzyme for the metabolic utilization of sorbitol and plays an important role in supplying carbon during fruit development of those plants which use sorbitol as the main product of photosynthesis.
butanediol biosynthesis and degradation	Bacterial pyruvate fermentation. Mis-predictions in VitiCyc, but energy-related nonetheless.

Table C.10 Successor pathways with high flow from Cluster 6 genes. Discriminated at the 99% confidence level without Bonferronni correction.

resveratrol biosynthesis	Resveratrol is the major polyphenol produced in a restricted number of plant genera with *Vitis Arachis* and *Pinus* as the prime representatives where it is formed as a general response to biotic and abiotic stress.
betanidin degradation	The violet to red betacyaninse.g.FRAME:PWY-5399are a subclass of the betalainsseeFRAME:PWY-5394and occur almost exclusively in the order of Caryophyllales and their degradation remains highly unknown. Causes discoloration and breakdown of betacyanins in Beta vulgaris under uptake of oxygen.
wax esters biosynthesis II	Wax ester biosynthesis appears to be carried out by sets of unrelated enzymes particularly the acyltransferases three distinct types have been identified; the wax synthases from jojoba are the first kind, a second type described in this pathway was identified in Acinobacter calcoaceticus. A third type identified in mammals is involved in the wax production within sebum and mebium glands.
acyl-CoA hydrolysis	Fatty acids are often found in the cell in the activated form of an acyl-coA. Acyl-CoAs are used in the biosynthesis of many cellular products and components. In plants they are involved in the biosynthesis of membrane lipids, seed storage lipids, wax, and suberin.
acyl carrier protein metabolism	All polyketide synthases, fatty-acid synthases and non-ribosomal peptide synthases require post-translational modification of their constituent acyl-carrier-protein (ACP) domains to become catalytically active.
ascorbate glutathione cycle	Many metabolic processes (including respiration, photosynthetic electron transport and oxidation of glycolate in photorespiration) generate active oxygen species such as singleton oxygen and hydrogen peroxide. The toxic singleton oxygen and hydrogen peroxide can damage membrane lipids and certain enzymes and thus interrupt the cell function. The ascorbate-glutathione cycle scavenges hydrogen peroxide.

Table C.11 Overrepresented Annotation: Cluster 7

Biological Process							
GO-ID	p-value	corr p-value	Hits in subcluster	Hits in ontology	Subcluster size	Ontology size	Description
6334	2.3010E-10	1.8187E-8	9	41	48	4479	nucleosome assembly
6323	4.5469E-10	1.8187E-8	9	44	48	4479	DNA packaging
31497	4.5469E-10	1.8187E-8	9	44	48	4479	chromatin assembly
22607	8.5345E-10	2.5011E-8	9	47	48	4479	cellular component assembly
6333	1.0421E-9	2.5011E-8	9	48	48	4479	chromatin assembly or disassembly
6325	8.3524E-9	1.6705E-7	9	60	48	4479	establishment and/or maintenance of chromatin architecture
65004	1.3058E-8	2.2385E-7	9	63	48	4479	protein-DNA complex assembly
7001	1.7359E-8	2.6038E-7	9	65	48	4479	chromosome organization and biogenesis
16043	7.9083E-8	1.0544E-6	15	275	48	4479	cellular component organization and biogenesis
65003	6.6837E-6	8.0205E-5	9	129	48	4479	macromolecular complex assembly
6996	8.0257E-6	8.7553E-5	10	168	48	4479	organelle organization and biogenesis
43933	1.3080E-5	1.3080E-4	9	140	48	4479	macromolecular complex subunit organization
7047	5.4236E-4	4.6488E-3	5	64	48	4479	cell wall organization and biogenesis
45229	5.4236E-4	4.6488E-3	5	64	48	4479	external encapsulating structure organization and biogenesis
9664	2.2824E-3	1.8259E-2	2	7	48	4479	plant-type cell wall organization and biogenesis
9698	5.8163E-3	4.1962E-2	2	11	48	4479	phenylpropanoid metabolic process
6869	5.9446E-3	4.1962E-2	3	35	48	4479	lipid transport

Table C.11 (Continued)

Molecular Function							
5199	3.0222E-6	3.4756E-4	3	4	53	5697	structural constituent of cell wall
Cellular Component							
786	5.1336E-11	9.2405E-10	9	36	28	2504	nucleosome
32993	5.1336E-11	9.2405E-10	9	36	28	2504	protein-DNA complex
5717	1.4503E-10	1.7403E-9	9	40	28	2504	chromatin
44427	4.5396E-10	4.0857E-9	9	45	28	2504	chromosomal part
5694	1.0695E-8	7.7006E-8	9	63	28	2504	chromosome
44422	4.8744E-3	2.1520E-2	10	371	28	2504	organelle part
44446	4.8744E-3	2.1520E-2	10	371	28	2504	intracellular organelle part
43228	5.3800E-3	2.1520E-2	10	376	28	2504	non-membrane-bounded organelle
43232	5.3800E-3	2.1520E-2	10	376	28	2504	intracellular non-membrane-bounded organelle
5634	6.3001E-3	2.2681E-2	13	589	28	2504	nucleus
5576	1.6363E-2	5.3551E-2	3	49	28	2504	extracellular region

Table C.12 Successor pathways with high flow from Cluster 7 genes. Discriminated at the 95% confidence level with Bonferronni correction.

D-mannose degradation	D-mannose can serve as a total source of carbon and energy for growth of *E. coli* and is also used in plants
HGA biosynthesis and degradation	Homogalacturonan (HGA) is a linear chain of 14-linked a-D-galacturonic acid residues in which some of the carboxyl groups are methyl esterified. HGA accounts for up to 60 percent of the pectin found in primary cell walls of all plants.
cellulose biosynthesis	Cellulose is present universally in the cell walls of plants.

Table C.13 Overrepresented Annotation: Cluster 8

Biological Process							
GO-ID	p-value	corr p-value	Hits in subcluster	Hits in ontology	Subcluster size	Ontology size	Description
43631	1.1019E-3	7.4375E-2	2	5	48	4479	RNA polyadenylation
31123	1.1019E-3	7.4375E-2	2	5	48	4479	RNA 3'-end processing
6979	1.9965E-3	8.9840E-2	3	24	48	4479	response to oxidative stress
Molecular Function							
19842	4.5252E-5	5.2492E-3	6	62	61	5697	vitamin binding
4645	3.3603E-4	1.9490E-2	2	3	61	5697	phosphorylase activity
4652	1.1047E-3	2.9287E-2	2	5	61	5697	polynucleotide adenylyltransferase activity
4553	1.8867E-3	2.9287E-2	7	167	61	5697	hydrolase activity, hydrolyzing O-glycosyl compounds
30170	1.8967E-3	2.9287E-2	4	50	61	5697	pyridoxal phosphate binding
16684	2.0145E-3	2.9287E-2	3	24	61	5697	oxidoreductase activity, acting on peroxide as acceptor
4601	2.0145E-3	2.9287E-2	3	24	61	5697	peroxidase activity
16798	2.0198E-3	2.9287E-2	7	169	61	5697	hydrolase activity, acting on glycosyl bonds
30976	2.2880E-3	2.9490E-2	2	7	61	5697	thiamin pyrophosphate binding
1871	4.8024E-3	4.2856E-2	2	10	61	5697	pattern binding
30247	4.8024E-3	4.2856E-2	2	10	61	5697	polysaccharide binding
8061	4.8024E-3	4.2856E-2	2	10	61	5697	chitin binding
5488	5.0939E-3	4.2856E-2	51	3888	61	5697	binding
48037	5.1723E-3	4.2856E-2	8	253	61	5697	cofactor binding
16209	8.7098E-3	6.7356E-2	3	40	61	5697	antioxidant activity
4568	1.2289E-2	8.9094E-2	2	16	61	5697	chitinase activity

Table C.14 Successor pathways with high flow from Cluster 8 genes. Discriminated at the 95% confidence level with Bonferronni correction.

alanine biosynthesis III	Alanine is an essential component of protein (as L-alanine) and peptidoglycan (as a roughly 3:1 mix of D- and L-alanine). Only about a tenth of total alanine synthesized is incorporated into peptidoglycan.
sorbitol degradation I	Sorbitol is a hexitolone of several acyclic polyols or sugar alcohols found in higher plants. The translocated sorbitol is ’unloaded’ in fruit tissues and eventually converted to fructose *via* a NAD$^+$-dependent sorbitol dehydrogenase which is a key enzyme for the metabolic utilization of sorbitol and plays an important role in supplying carbon during fruit development of those plants which use sorbitol as the main product of photosynthesis.
starch and glycogen biosynthesis	Glycogen and starch, megadalton-sized glucose polymers, are the major reservoir of readily available energy and carbon compounds in most living organisms ranging from archaea, eubacteria and yeasts up to higher eukaryotes including plants and animals.

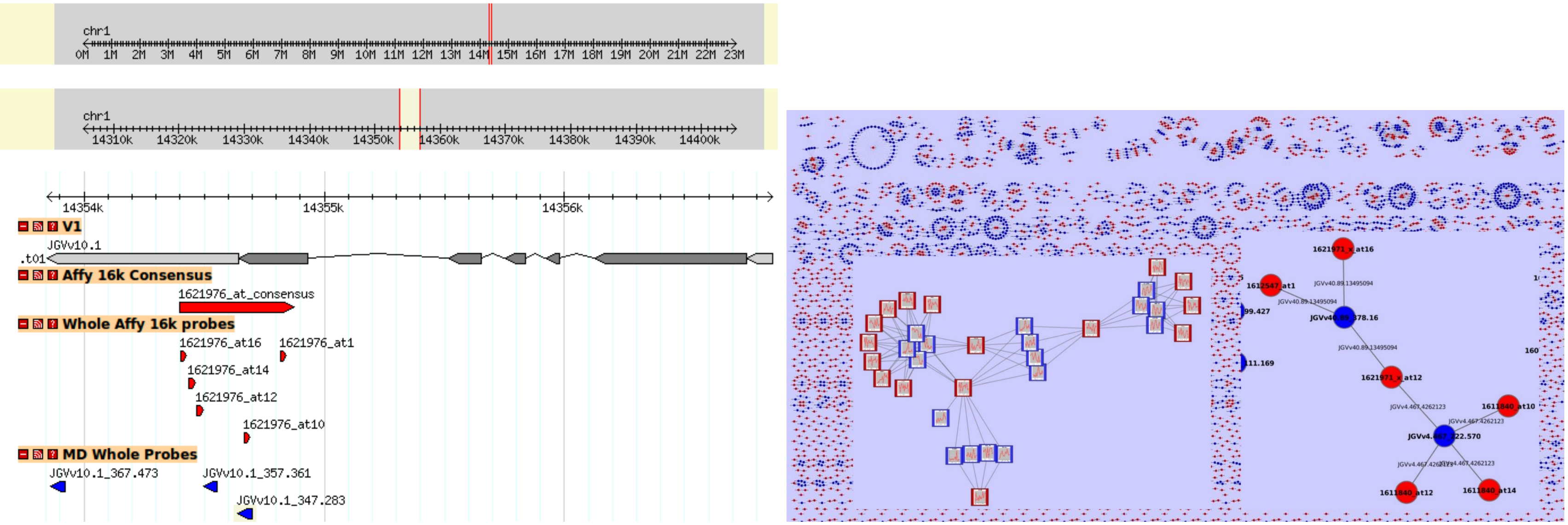


Figure C.3 The Nimblegen platform used was designed based on gene models from the latest grapevine genome, so their positions on the chromosomes are known. After mapping Affymetrix probes to chromosomal positions via probeset consensus BLAT, we can link pairs of cross-platform probes (ie one Affymetrix probe and one Nimblegen probe) based on common exons in a gene model; if two probes measure the same exon, they should be highly correlated in any dataset where the exon's expression is perturbed. See an example of Affymetrix (red) and Nimblegen (blue) probes with a common exon on the left. On the right, we have modeled these probe relationships with networks of probe-nodes linked by common exon-edges.

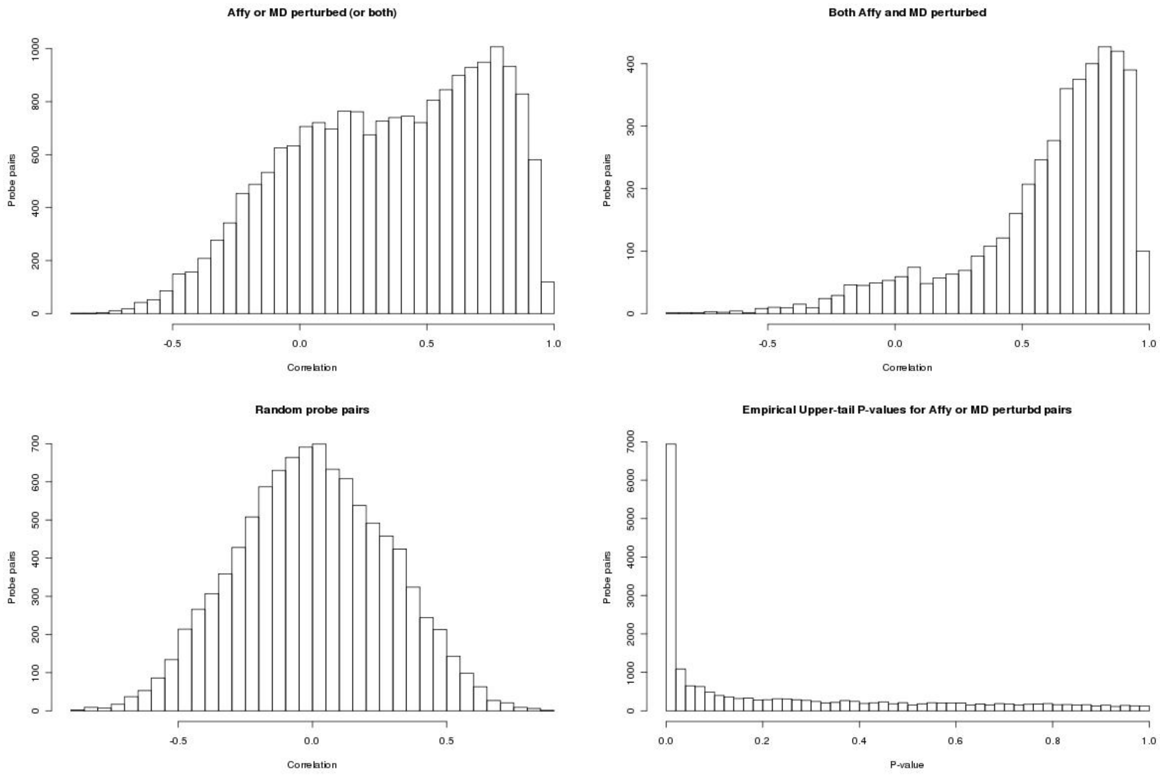


Figure C.4 Distributions of correlations between cross-platform pairs of probes linked by common exons. We aim to remove Affymetrix probe-genome alignments associated with poor correlation with exon-linked Nimblegen probes. Top-left shows the histogram of cross-platform probe pair correlations where the either the Affymetrix ("Affy") probe, Nimblegen ("MD") probe, or both probes are perturbed in expression. This appears to be a mixture of two populations: probe pairs with correlations normally distributed around zero and probe pairs with correlations approaching one. Top-right shows a histogram of correlations where both probes are perturbed. Requiring that both are perturbed greatly reduces the population of zero-correlated pairs. Bottom-left shows a correlations of randomly selected probe pairs and bottom-right shows a histogram of p-values for the pairs in top-left, evaluated using the sampling distribution generated for bottom-right.

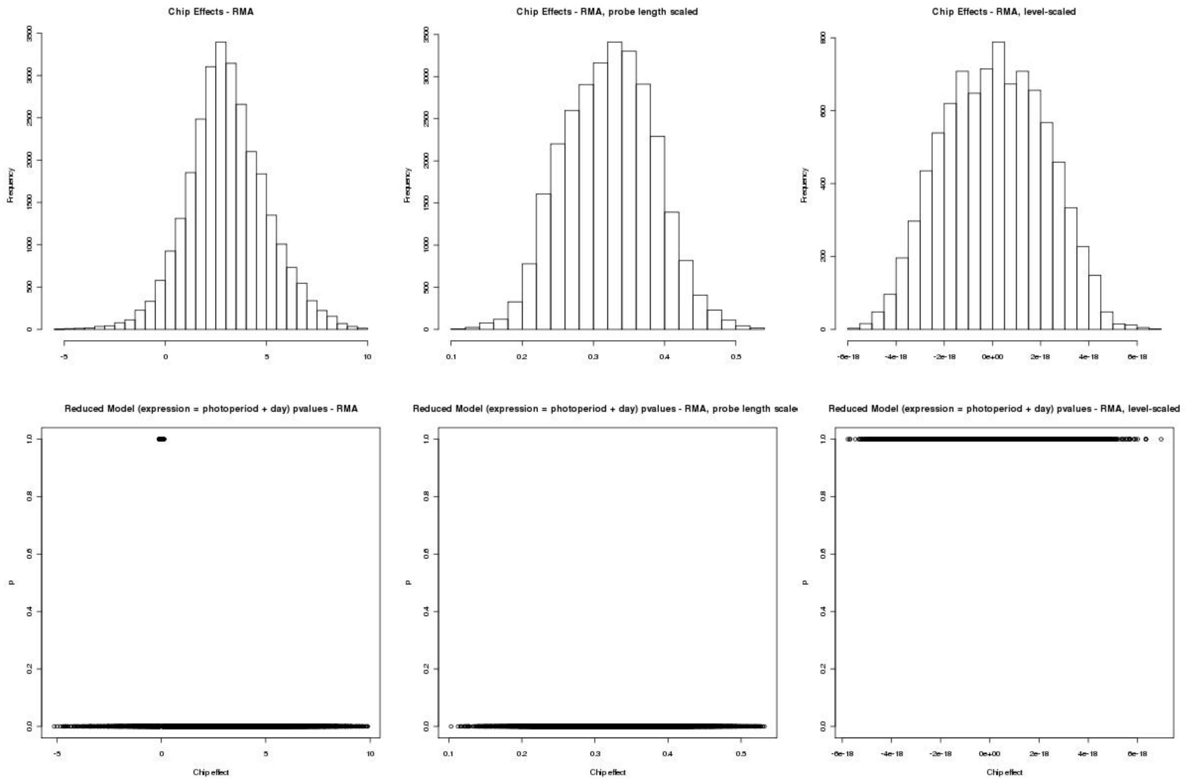


Figure C.5 After pooling the two platforms, we fit a linear model which accounts for a ”chip effect.” This chip effect measures the significance of the platform quantification differences and is non-zero both for RMA and RMA plus probe length scaling (left and center columns, respectively). However, the chip effect vanishes when we use level-scaling, and the chip effects are tightly distributed around zero (top-right). The bottom row shows *p*-values for each probeset when testing the null hypothesis that the chip effect equals zero.

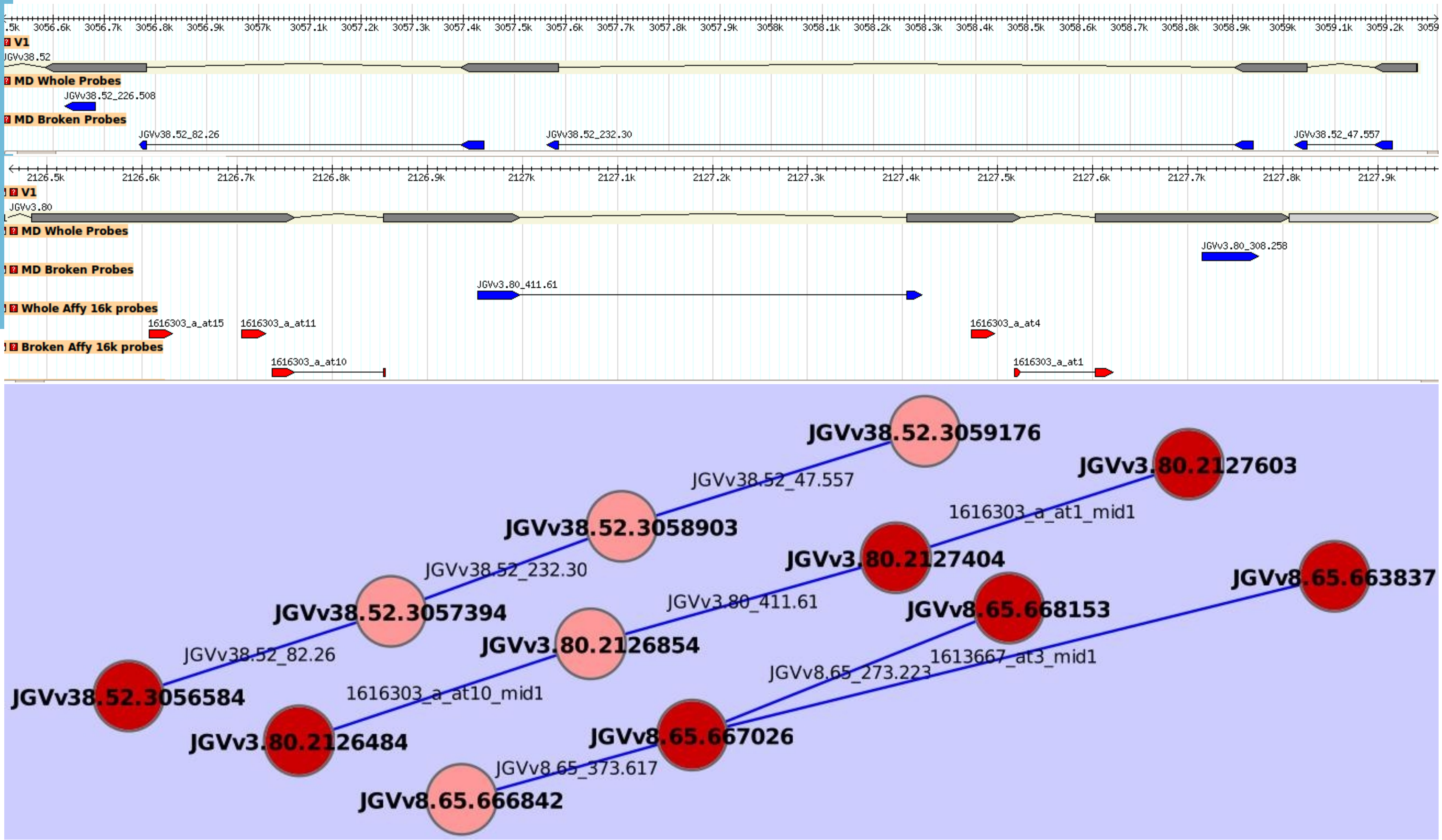


Figure C.6 Probe-Exon Systems are small networks (bottom) where exons are nodes linked by probes spliced between them. Darker exon-nodes are exons containing unspliced probes. Above, see the gene models and aligned probes in GBrowse. Both Affymetrix and Nimblegen probes can be spliced across exons, linking the exons. Spliced probes link exons where the observed probe fluorescence values are weighted sums of the unknown expressions of the exon(s) which then measure.

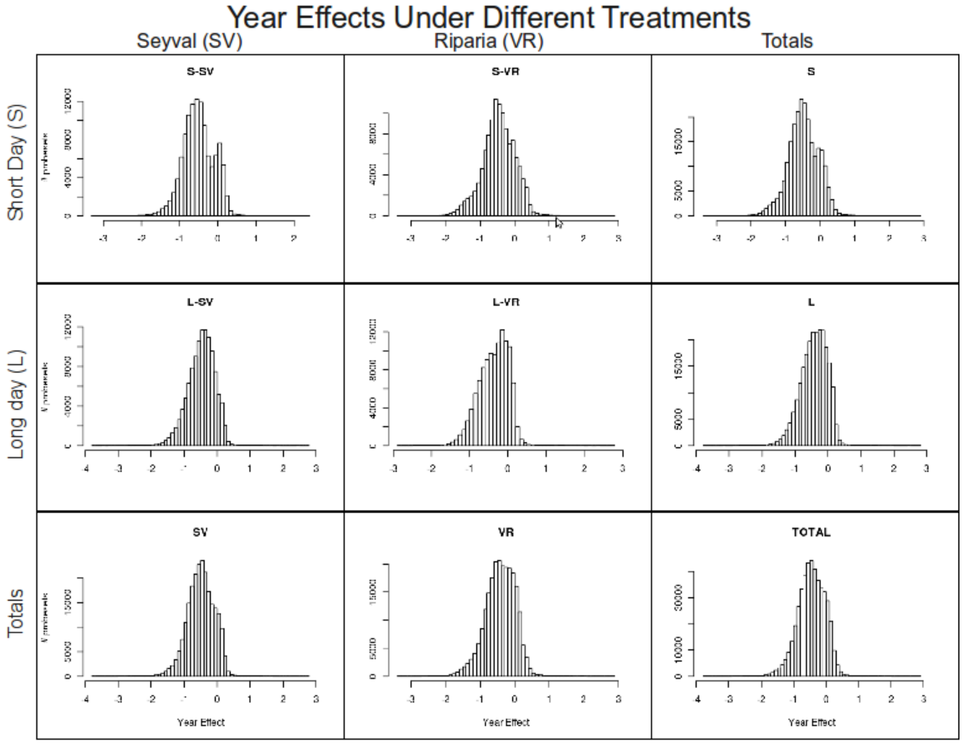


Figure C.7 Histograms of year effects as calculated by Equation 5.9. The plausibility of two different populations is most apparent in the bimodal distribution of year effect under the short day treatment for *Seyval* (top left). While this strong bimodality in the short day × *Seyval* treatment carries through to the totals in the bottom row and right hand column of histograms, they other joint distributions (S-VR, L-VR, L-SV) also show a slight skew which may be cause by two very convoluted distributions.

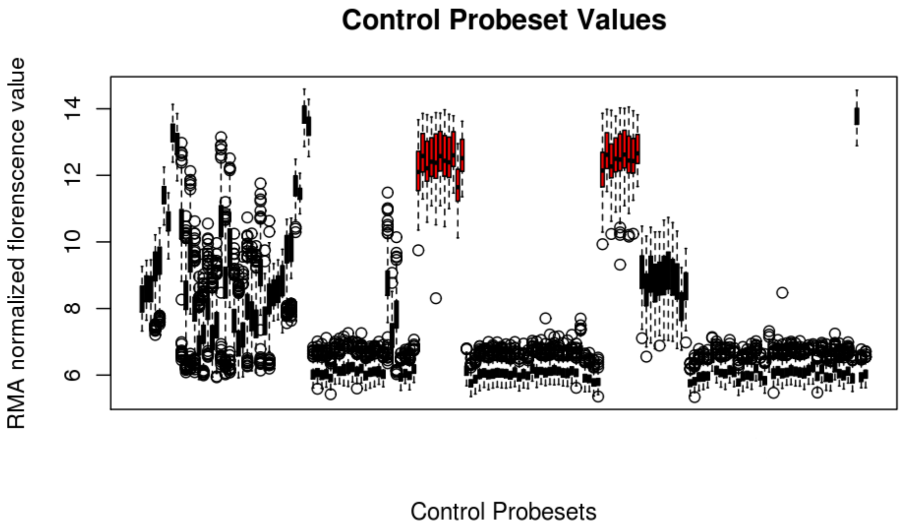


Figure C.8 Boxplots for expression in each hybridization for all control probesets. The 20 selected as positive controls are shown in red. These are the controls used as expressed references for variability caused by noise.

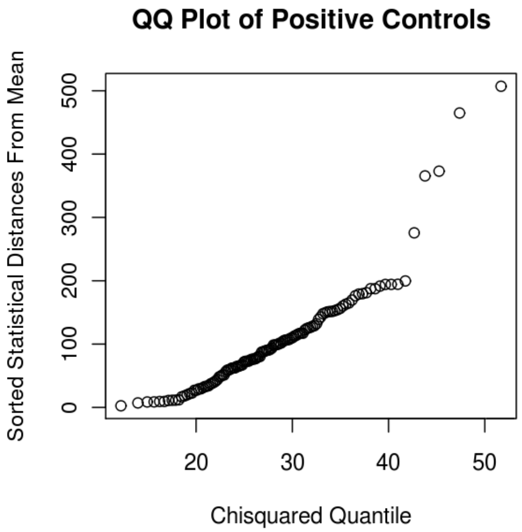


Figure C.9 The Quantile-Quantile plot checking for multivariate normality in positive controls shows good fit, except for outliers in the tails. Since the quantiles of controls statistical distance from their centroid line up with the theoretical χ^2 quantiles, we can assume that the multivariate distribution of noise is multivariate Normal.

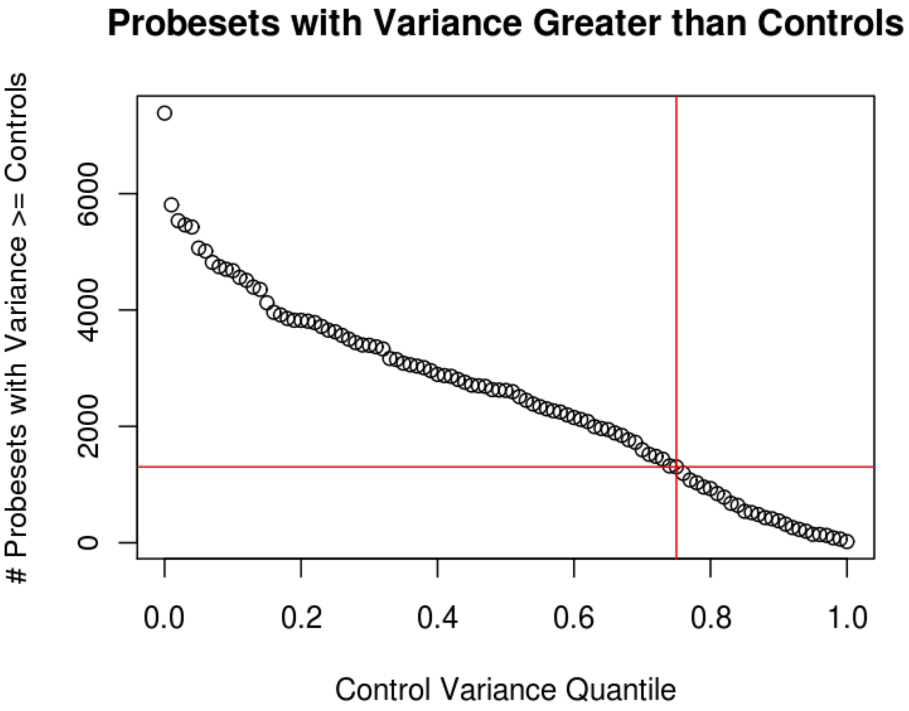


Figure C.10 As we increase the control percentil cutoff (x-axis), the number of experimental probesets with variance at least as large (y-axis) decreases. In order to mine only the probesets perturbed by the treatments, we selected a variance cutoff equal to the 75^{th} percentile of the positive control variances, and selected experimental probesets at least as variable as this, resulting in 1,304 perturbed probesets, making up 7.9% of the entire set.

Table C.15 Significant successor pathways to a query list of genes in Pinot Noir but missing from Corvina (”Del”) and assembled with high depth (”Dup”), and based on RefSeq mapping (with a confidence level of 95% and no Bonferronni correction)

VitiCyc Pathway	Description from MetaCyc	Del p-value	Dup p-value
anthocyanin biosynthesis (pelargonidin 3-O-glucosidecyanidin 3-O-glucoside)	responsible for the red to magenta coloration of flowers and fruits of plants	0.02	0.01
anthocyanin biosynthesis (delphinidin 3-O-glucoside)	responsible for introducing blue tones to the floral organs of plants	0.05	0.03
homogalacturonan biosynthesis	Cell wall activity during ripening	0.02	< 0.0001
xylan biosynthesis	Cell wall activity during ripening	< 0.0001	< 0.0001

Table C.15 (Continued)

rutin biosynthesis	a flavonol glucoside that is widespread in the plant kingdom, and has been especially found in the Polygonaceae family. Rutin has several interesting pharmacological properties (e.g. antioxidative activity)	< 0.0001	< 0.0001
phospholipases	Plant signal transduction, such as responding to auxin stimulation of growth, pathogen and elicitor.	0.04	> 0.05
glycolipid biosynthesis	Major constituent of skin and pulp	0.0001	< 0.0001
phospholipid biosynthesis II	The major structural components of biological membranes. Some phospholipids (i.e. phosphatidylinositol) also serve as important lipid-signaling molecules	0.0001	> 0.05
gentiodelphin biosynthesis	An unusual stable anthocyanin conferring deep blue flower colors to species of the genus Gentiana	0.04	> 0.05
momilactone biosynthesis	Responsible for wine aromas	> 0.05	< 0.0001
UDP-galactose biosynthesis	Sugars metabolism responsible for wine flavors	> 0.05	0.04
triacylglycerol degradation	Responsible for wine flavors	> 0.05	< 0.0001

BIBLIOGRAPHY

Agarwal, A., Koppstein, D., Rozowsky, J., Sboner, A., Habegger, L., Hillier, L. W., Sasidharan, R., Reinke, V., Waterston, R. H., and Gerstein, M. (2010). Comparison and calibration of transcriptome data from rna-seq and tiling arrays. *BMC Genomics*, 11:16.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

Ansaldi, M., Simon, G., Lepelletier, M., and Mejean, V. (2000). The torr high-affinity binding site plays a key role in both torr autoregulation and torcad operon expression in escherichia coli. *Journal of Bacteriology*, 182(4):961–966.

Antonov, A. V., Dietmann, S., and Mewes, H. W. (2008). Kegg spider: interpretation of genomics data in the context of the global gene metabolic network. *Genome Biology*, 9(12):11.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., Sherlock, G., and Gene Ontology, C. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Avraham, S., Tung, C. W., Ilic, K., Jaiswal, P., Kellogg, E. A., McCouch, S., Pujar, A., Reiser, L., Rhee, S. Y., Sachs, M. M., Schaeffer, M., Stein, L., Stevens, P., Vincent, L., Zapata, F., and Ware, D. (2008). The plant ontology database: a community resource for plant structure and developmental stages controlled vocabulary and annotations. *Nucleic Acids Research*, 36:–449.

Barabasi, A. L. and Bonabeau, E. (2003). Scale-free networks. *Scientific American*, 288(5):60–69.

Barb, A. W., McClerren, A. L., Snehelatha, K., Reynolds, C. M., Zhou, P., and Raetz, C. R. H. (2007). Inhibition of lipid A biosynthesis as the primary mechanism of CHIR-090 antibiotic activity in Escherichia coli. *BIOCHEMISTRY*, 46(12):3793–3802.

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I. F., Soboleva, A., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Muertter, R. N., and Edgar, R. (2009). Ncbi geo: archive for high-throughput functional genomic data. *Nucleic Acids Research*, 37:–885.

Barry, W. T., Nobel, A. B., and Wright, F. A. (2005). Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics*, 21(9):1943–1949.

Batchelor, E., Walthers, D., Kenney, L. J., and Goulian, M. (2005). The escherichia coli cpxa-cpxr envelope stress response system regulates expression of the porins ompf and ompc. *Journal of Bacteriology*, 187(16):5723–5731.

Choi, J. K., Yu, U., Kim, S., and Yoo, O. J. (2003). Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19:–84.

Cordero, F., Pensa, R. G., Visconti, A., Ienco, D., and Botta, M. (2009). *Ontology-Driven Co-clustering of Gene Expression Data*, volume 5883, pages 426–435. Springer-Verlag Berlin, Berlin.

Corporation, N. (2007). *NVIDIA CUDA Programming Guide*.

Delledonne, M. (2009). The ”v1” gene models predicted for the new ”12x” grape genome.

Durstenfeld, R. (1964). Algorithm-235 - random permutation g6. *Communications of the Acm*, 7(7):420–420.

Fennell, A., Mathiason, K., and Luby, J. (2005). *Genetic segregation for indicators of photoperiod control of dormancy induction in Vitis species*, pages 533–539.

Fodor, A. A., Tickle, T. L., and Richardson, C. (2007). Towards the uniform distribution of null p-values on affymetrix microarrays. *Genome Biology*, 8(5).

Funahashi, A., Matsuoka, Y., Jouraku, A., Morohashi, M., Kikuchi, N., and Kitano, H. (2008). Celldesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the Ieee*, 96(8):1254–1265.

Garrett-Mayer, E., Parmigiani, G., Zhong, X. G., Cope, L., and Gabrielson, E. (2008). Cross-study validation and combined analysis of gene expression microarray data. *Biostatistics*, 9(2):333–354.

GMOD (2010). Generic Model Organism Database (GMOD).

Grimplet, J., Cramer, G. R., Dickerson, J. A., Mathiason, K., Van Hemert, J., and Fennell, A. Y. (2009). Vitisnet: ”omics” integration through grapevine molecular networks. *Plos One*, 4(12).

Guo, A. Y., Chen, X., Gao, G., Zhang, H., Zhu, Q. H., Liu, X. C., Zhong, Y. F., Gu, X. C., He, K., and Luo, J. C. (2008). Planttfdb: a comprehensive plant transcription factor database. *Nucleic Acids Research*, 36:–966.

Haider, S., Ballester, B., Smedley, D., Zhang, J. J., Rice, P., and Kasprzyk, A. (2009). Biomart central portal-unified access to biological data. *Nucleic Acids Research*, 37:–23.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Horvath, D. (2009). Common mechanisms regulate flowering and dormancy. *Plant Science*, 177(6):523–531.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J. H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novere, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J., and Forum, S. (2003). The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531.

Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003a). Summaries of affymetrix genechip probe level data. *Nucleic Acids Research*, 31(4).

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003b). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264.

Jaillon, O., Aury, J. M., Noel, B., Policriti, A., Clepet, C., Casagrande, A., Choisne, N., Aubourg, S., Vitulo, N., Jubin, C., Vezzi, A., Legeai, F., Hugueney, P., Dasilva, C., Horner, D., Mica, E., Jublot, D., Poulain, J., Bruyere, C., Billault, A., Segurens, B., Gouyvenoux, M., Ugarte, E., Cattonaro, F., Anthouard, V., Vico, V., Del Fabbro, C., Alaux, M., Di Gaspero, G., Dumas, V., Felice, N., Paillard, S., Juman, I., Moroldo, M., Scalabrin, S., Canaguier, A., Le Clainche, I., Malacrida, G., Durand, E., Pesole, G., Laucou, V., Chatelet, P., Merdinoglu, D., Delledonne, M., Pezzotti, M., Lecharny, A., Scarpelli, C., Artiguenave, F., Pe, M. E., Valle, G., Morgante, M., Caboche, M., Adam-Blondon, A. F., Weissenbach, J., Quetier, F., Wincker, P., and French-Italian, P. (2007). The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. *Nature*, 449(7161):463–5.

Karlin, S. and Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the United States of America*, 87(6):2264–2268.

Karp, P. D. (2005). Biocyc pathway database collection and the pathway tools software. *Abstracts of Papers of the American Chemical Society*, 229:044.

Karp, P. D., Ouzounis, C. A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahren, D., Tsoka, S., Darzentas, N., Kunin, V., and Lopez-Bigas, N. (2005). Expansion of the biocyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research*, 33(19):6083–6089.

Kent, W. J. (2002). Blat - the blast-like alignment tool. *Genome Research*, 12(4):656–664.

Keseler, I. M., Bonavides-Martinez, C., Collado-Vides, J., Gama-Castro, S., Gunsalus, R. P., Johnson, D. A., Krummenacker, M., Nolan, L. M., Paley, S., Paulsen, I. T., Peralta-Gil, M., Santos-Zavaleta, A., Shearer, A. G., and Karp, P. D. (2009). EcoCyc: A comprehensive view of Escherichia coli biology. *Nucleic Acids Research*, 37:D464–D470.

Killcoyne, S. and Pico, A. (2009). Cytoscape RFC: Data Integration.

Kim, J. S., Kim, S. J., Park, H. W., Youn, J. P., An, Y. R., Cho, H., and Hwang, S. Y. (2010). Array2kegg: Web-based tool of kegg pathway analysis for gene expression profile. *Biochip Journal*, 4(2):134–140.

Knijnenburg, T. A., Wessels, L. F. A., Reinders, M. J. T., and Shmulevich, I. (2009). Fewer permutations, more accurate p-values. *Bioinformatics*, 25(12):–161.

Krummenacker, M., Paley, S., Mueller, L., Yan, T., and Karp, P. D. (2005). Querying and computing with biocyc databases. *Bioinformatics*, 21(16):3454–3455.

Kuhn, N., Ormeno-Nunez, J., Jaque-Zamora, G., and Perez, F. J. (2009). Photoperiod modifies the diurnal expression profile of vvphya and vvphyb transcripts in field-grown grapevine leaves. *Journal of Plant Physiology*, 166(11):1172–1180.

Le Novere, N., Hucka, M., Mi, H. Y., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M. I., Wimalaratne, S. M., Bergman, F. T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villeger, A., Boyd, S. E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T. C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G. M., Goryanin, I., Kell, D. B., Sander, C., Sauro, H., Snoep, J. L., Kohn, K., and Kitano, H. (2009). The systems biology graphical notation. *Nature Biotechnology*, 27(8):735–741.

Loui, C., Chang, A. C., and Lu, S. W. (2009). Role of the arcab two-component system in the resistance of escherichia coli to reactive oxygen stress. *BMC Microbiology*, 9.

Lysenko, A., Hindle, M. M., Taubert, J., Saqi, M., and Rawlings, C. J. (2009). Data integration for plant genomics-exemplars from the integration of arabidopsis thaliana databases. *Briefings in Bioinformatics*, 10(6):676–693.

Maere, S., Heymans, K., and Kuiper, M. (2005). Bingo: a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449.

Mao, L. Y., Van Hemert, J. L., Dash, S., and Dickerson, J. A. (2009). Arabidopsis gene co-expression network and its functional modules. *BMC Bioinformatics*, 10.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature Methods*, 5(7):621–628.

Mueller, L. A., Solow, T. H., Taylor, N., Skwarecki, B., Buels, R., Binns, J., Lin, C. W., Wright, M. H., Ahrens, R., Wang, Y., Herbst, E. V., Keyder, E. R., Menda, N., Zamir, D., and Tanksley, S. D. (2005). The sol genomics network. a comparative resource for solanaceae biology and beyond. *Plant Physiology*, 138(3):1310–1317.

Nettleton, D., Recknor, J., and Reecy, J. M. (2008). Identification of differentially expressed gene categories in microarray studies using nonparametric multivariate analysis. *Bioinformatics*, 24(2):192–201.

Noriega, X., Burgos, B., and Perez, F. J. (2007). Short day-photoperiod triggers and low temperatures increase expression of peroxidase rna transcripts and basic peroxidase isoenzyme activity in grapevine buds. *Phytochemistry*, 68(10):1376–1383.

Okuda, S., Yamada, T., Hamajima, M., Itoh, M., Katayama, T., Bork, P., Goto, S., and Kanehisa, M. (2008). Kegg atlas mapping for global analysis of metabolic pathways. *Nucleic Acids Research*, 36:–423.

Opsahl, T., Agneessens, F., and Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251.

Parmigiani, G., Garrett, E. S., Anbazhagan, R., and Gabrielson, E. (2002). A statistical framework for expression-based molecular classification in cancer. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 64:717–736.

Pauli, G. and Overath, P. (1972). Ato operon - a highly inducible system for acetoacetate and butyrate degradation in escherichia-coli. *European Journal of Biochemistry*, 29(3):553.

Pdlozhnyuk, V. (2007). Parallel mersenne twister.

Perez, F. J., Rubio, S., and Ormeno-Nunez, J. (2007). Is erratic bud-break in grapevines grown in warm winter areas related to disturbances in mitochondrial respiratory capacity and oxidative metabolism? *Functional Plant Biology*, 34(7):624–632.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Raetz, C. R. H., Garrett, T. A., Reynolds, C. M., Shaw, W. A., Moore, J. D., Smith, D. C., Ribeiro, A. A., Murphy, R. C., Ulevitch, R. J., Fearns, C., Reichart, D., Glass, C. K., Benner, C., Subramaniam, S., Harkewicz, R., Bowers-Gentry, R. C., Buczynski, M. W., Cooper, J. A., Deems, R. A., and Dennis, E. A. (2006). Kdo(2)-lipid a of escherichia coli, a defined endotoxin that activates macrophages via tlr-4. *Journal of Lipid Research*, 47(5):1097–1111.

Rhodes, D. R., Barrette, T. R., Rubin, M. A., Ghosh, D., and Chinnaiyan, A. M. (2002). Meta-analysis of microarrays: Interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer. *Cancer Research*, 62(15):4427–4433.

Rotter, A., Camps, C., Lohse, M., Kappel, C., Pilati, S., Hren, M., Stitt, M., Coutos-Thevenot, P., Moser, C., Usadel, B., Delrot, S., and Gruden, K. (2009). Gene expression profiling in susceptible interaction of grapevine with its fungal pathogen eutypa lata: Extending mapman ontology for grapevine. *BMC Plant Biology*, 9.

Shabalin, A. A., Tjelmeland, H., Fan, C., Perou, C. M., and Nobel, A. B. (2008). Merging two gene-expression studies via cross-platform normalization. *Bioinformatics*, 24(9):1154–1160.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504.

Shen, L. H., Gong, J., Caldo, R. A., Nettleton, D., Cook, D., Wise, R. P., and Dickerson, J. A. (2005). Barleybase - an expression profiling database for plant genornics. *Nucleic Acids Research*, 33:–614.

Shen, R. L., Ghosh, D., and Chinnaiyan, A. M. (2004). Prognostic meta-signature of breast cancer developed by two-stage mixture modeling of microarray data. *BMC Genomics*, 5.

Smedley, D., Haider, S., Ballester, B., Holland, R., London, D., Thorisson, G., and Kasprzyk, A. (2009). Biomart - biological queries made easy. *BMC Genomics*, 10.

Storey, J. D. (2003). The positive false discovery rate: A bayesian interpretation and the q-value. *Annals of Statistics*, 31(6):2013–2035.

Storey, J. D., Taylor, J. E., and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 66:187–205.

Storey, J. D. and Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16):9440–9445.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550.

Sucaet, Y. and Wurtele, E. (2010). MetNetAPI: A flexible method to access biological network data from MetNet. *BMC Research Notes*.

Swayne, D. F. and Buja, A. (2004). *Exploratory visual analysis of graphs in Ggobi*. Compstat 2004: Proceedings in Computational Statistics.

Thimm, O., Blasing, O., Gibon, Y., Nagel, A., Meyer, S., Kruger, P., Selbig, J., Muller, L. A., Rhee, S. Y., and Stitt, M. (2004). Mapman: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant Journal*, 37(6):914–939.

Towfic, F., VanderPlas, S., Oliver, C. A., Couture, O., Tuggle, C. K., Greenlee, M. H. W., and Honavar, V. (2010). Detection of gene orthology from gene co-expression and protein interaction networks. *BMC Bioinformatics*, 11.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–174.

Tsiliki, G., Zervakis, M., Ioannou, M., Sanidas, E., Stathopoulos, E., Tsiknakis, M., and Kafetzopoulos, D. (2009). *Multi-platform data integration in microarray analysis*. 2009 9th International Conference on Information Technolology and Applications in Biomedicine. Ieee, New York.

Usadel, B., Poree, F., Nagel, A., Lohse, M., Czedik-Eysenberg, A., and Stitt, M. (2009). A guide to using mapman to visualize and compare omics data in plants: a case study in the crop species, maize. *Plant Cell and Environment*, 32(9):1211–1229.

van den Berg, R. A., Hoefsloot, H. C. J., Westerhuis, J. A., Smilde, A. K., and van der Werf, M. J. (2006). Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genomics*, 7.

van Dongen, S. (2000). *Graph Clustering by Flow Simulation*. PhD dissertation, University of Utrecht.

Van Hemert, J. L. and Dickerson, J. A. (2010a). Monte carlo randomization tests for large-scale abundance datasets on the gpu. *Comput Methods Programs Biomed*.

Van Hemert, J. L. and Dickerson, J. A. (2010b). PathwayAccess: CellDesigner plugins for pathway databases. *Bioinformatics*.

Van Hemert, J. L., Pezzotti, M., and Dickerson, J. A. (2010). Viticyc.

van Iterson, M., Boer, J. M., and Menezes, R. X. (2010). Filtering, FDR and power. *BMC Bioinformatics*, 11:11.

Vastrik, I., D'Eustachio, P., Schmidt, E., Gopinath, G., Croft, D., de Bono, B., Gillespie, M., Jassal, B., Lewis, S., Matthews, L., Wu, G., Birney, E., and Stein, L. (2009). Reactome: a knowledge base of biologic pathways and processes (vol 8, pg 39, 2007). *Genome Biology*, 10(2).

Vergara, R. and Perez, F. J. (2010). Similarities between natural and chemically induced bud-endodormancy release in grapevine vitis vinifera l. *Scientia Horticulturae*, 125(4):648–653.

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. (2010). Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242.

Warnat, P., Eils, R., and Brors, B. (2005). Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6.

Wise, R. P., Caldo, R. A., Hong, L., Shen, L., Cannon, E., and Dickerson, J. A. (2007). Barleybase/plexdb - a unified expression profiling database for plants and plant pathogens. *Methods in Molecular Biology*, pages 347–363. Times Cited: 0 Edwards, D.

Wolfe, A. J., Parikh, N., Lima, B. P., and Zemaitaitis, B. (2008). Signal integration by the two-component signal transduction response regulator cpxr. *Journal of Bacteriology*, 190(7):2314–2322.

Wurtele, E. S., Li, L., Berleant, D., Cook, D., Dickerson, J. A., Ding, J., Hofmann, H., Lawrence, M., Lee, E. K., Li, J., Mentzen, W., Miller, L., Nikolau, B. J., Ransom, N., and Wang, Y. (2007). *Metnet: Systems biology tools for arabidopsis.* Concepts in Plant Metabolomics.

Xiong, H. L., Zhang, Y., Chen, X. W., and Yu, J. S. (2010). Cross-platform microarray data integration using the normalised linear transform. *International Journal of Data Mining and Bioinformatics*, 4(2):142–157.

Yamamoto, K., Matsumoto, F., Oshima, T., Fujita, N., Ogasawara, N., and Ishihama, A. (2008). Anaerobic regulation of citrate fermentation by citab in escherichia coli. *Bioscience Biotechnology and Biochemistry*, 72(11):3011–3014.

Zenoni, S., Ferrarini, A., Giacomelli, E., Xumerle, L., Fasoli, M., Malerba, G., Bellin, D., Pezzotti, M., and Delledonne, M. (2010). Characterization of transcriptional complexity during berry development in vitis vinifera using rna-seq. *Plant Physiology*, 152(4):1787–1795.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000). A greedy algorithm for aligning dna sequences. *Journal of Computational Biology*, 7(1-2):203–214.

Zhu, K., Zhang, Y. M., and Rock, C. O. (2009). Transcriptional regulation of membrane lipid homeostasis in escherichia coli. *Journal of Biological Chemistry*, 284(50):34880–34888.