December 2018

High-Throughput Polygenic Biomarker Discovery Using Condition-Specific Gene Coexpression Networks

William Louis Poehlman
Clemson University, wpinvest153@gmail.com

HIGH-THROUGHPUT POLYGENIC BIOMARKER DISCOVERY USING CONDITION-SPECIFIC GENE COEXPRESSION NETWORKS

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Genetics

---

by
William L. Poehlman
December 2018

Accepted by:
Dr. F. Alex Feltus, Committee Chair
Dr. Julia Frugoli
Dr. Stephen Kresovich
Dr. Feng Luo
Dr. Hong Luo

ABSTRACT

Biomarkers can be described as molecular signatures that are associated with a trait or disease. RNA expression data facilitates discovery of biomarkers underlying complex phenotypes because it can capture dynamic biochemical processes that are regulated in tissue-specific and time-specific manners. Gene Coexpression Network (GCN) analysis is a method that utilizes RNA expression data to identify binary gene relationships across experimental conditions. Using a novel GCN construction algorithm, Knowledge Independent Network Construction (KINC), I provide evidence for novel polygenic biomarkers in both plant and animal use cases.

Kidney cancer is comprised of several distinct subtypes that demonstrate unique histological and molecular signatures. Using KINC, I have identified gene correlations that are specific to clear cell renal cell carcinoma (ccRCC), the most common form of kidney cancer. ccRCC is associated with two common mutation profiles that respond differently to targeted therapy. By identifying GCN edges that are specific to patients with each of these two mutation profiles, I discovered unique genes with similar biological function, suggesting a role for T cell exhaustion in the development of ccRCC.

Medicago truncatula is a legume that is capable of atmospheric nitrogen fixation through a symbiotic relationship between plant and rhizobium that results in root nodulation. This process is governed by complex gene expression patterns that are dynamically regulated across tissues over the course of rhizobial infection. Using de novo RNA sequencing data generated from the root maturation zone at five distinct time points, I identified hundreds of genes that were differentially expressed between control and inoculated plants at specific time points. To discover genes that were co-regulated during this experiment, I constructed a GCN using the KINC software. By combining GCN clustering analysis with differentially expressed genes, I present evidence for novel root nodulation biomarkers. These biomarkers suggest that temporal regulation of pathogen response related genes is an important process in nodulation.

Large-scale GCN analysis requires computational resources and stable data-processing pipelines. Supercomputers such as Clemson University's Palmetto Cluster provide data storage and processing resources that enable terabyte-scale experiments. However, with the wealth of public sequencing data available for mining, petabyte-scale experiments are required to provide novel insights across the tree of life. I discuss computational challenges that I have discovered with large scale RNA expression data mining, and present two workflows, OSG-GEM and OSG-KINC, that enable researchers to access geographically distributed computing resources to handle petabyte-scale experiments.

DEDICATION

This dissertation is dedicated to my parents, who provided endless support during my studies.

ACKNOWLEDGMENTS

I would like to acknowledge the support of my advisor, Dr. Alex Feltus, for his support and inspiration. He gave me the freedom to pursue unique and collaborative research outlets while ensuring that my time as a student was well managed, and provided me with an endless number of opportunities to travel the country and present my research. I would also like to acknowledge the support of the Open Science Grid staff for their prompt and dedicated support. In particular, I would like to thank Mats Rynge for his valuable assistance and consulting efforts. He went above and beyond his duties to provide technical support and advice, while teaching me valuable skills that transformed my career goals as a scientist. I acknowledge the support of the Frugoli lab at Clemson University for generating data used in this dissertation, and would like to thank Elise Schnabel for providing valuable feedback and insight on my data analysis. I would also like to thank Bill Paseman for his inspiration, without which I would not have embarked on the mission to better understand kidney cancer.

TABLE OF CONTENTS

Page

Table of Contents (Continued) Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

List of Figures (Continued)

Figure Page

CHAPTER ONE

INTRODUCTION

Biomarker Discovery

Reductionism refers to the scientific approach of explaining a complex phenomenon through a small number of discrete measurements [1]. While molecular biology approaches to dissect the cause of complex phenotypes are often described as reductionist methods, they are complementary to the holistic approaches of systems genetics that embrace the complexity of biological systems [2]. The human population faces many challenges that must be addressed through a combination of reductionist and holistic approaches. The second leading cause of death in America is cancer, a disease that demonstrates a seemingly unlimited number of molecular drivers [3, 4]. As an example, efforts to identify precise mechanisms for kidney cancer development have pinpointed specific genetic lesions associated with various clinical subtypes [5-7]. However, advanced stage kidney cancer remains an incurable disease in most patients [8]. Agriculture is another field that must be improved to support the growing human population. Experts estimate that we must double the current rate of food production by the year 2050 in order to support the growth of our human population [9]. One challenge in crop productivity is the application of nitrogen fertilizer, which is consumed at a rate of over 100 tons per year globally [10]. Root nodulation is a process that allows several legume species to fix atmospheric nitrogen through a symbiotic rhizobial infection, reducing the need for nitrogen fertilizer application [11, 12]. This process is governed by complex gene expression patterns that result in signaling cascades in specific tissues of the root [13, 14]. While the model legume *Medicago truncatula* has been used to study these complex expression dynamics, there is potential to translate this trait to other crops due to its evolutionary conserved pathways that are involved in a fungal symbiosis called mycorrhiza [12, 15]. Gaps in our understanding of how legumes have acquired the specialized functions to enable root nodulation must be solved before this trait can be translated into other crops.

This dissertation addresses the challenges mentioned above through a holistic approach to biomarker discovery. By utilizing the common systems genetics technique of gene coexpression network (GCN) analysis using a novel algorithm, I demonstrate evidence for a specific set of candidate biomarkers involved in kidney cancer and root nodulation. The results demonstrate that a holistic approach such as GCN analysis can be combined with other sources of data to obtain a set of hypotheses that can be tested using reductionist approaches. Chapter 2 describes the identification of gene correlations that are specific to a distinct clinical subtype of kidney cancer, and Chapter 3 applies a similar approach to identify root nodulation biomarkers. During this process I encountered significant computational challenges in data storage and processing. As the wealth of publicly available data grows, opportunity for data mining is matched by computational limitations. Chapters 4 and 5 discuss the development of computational workflows that enable researchers to access grid computing resources across the country in order to process and interpret large volumes of genomic data. This dissertation describes a novel approach to polygenic biomarker identification and demonstrates specific use-cases for this framework.

Gene Expression Biomarkers

Biomarkers are measurable blends of biological molecules that are associated with a specific trait or disease. These molecules can be genetic markers such as DNA base composition and sequence, biochemical markers such as RNA signatures, metabolite profiles such as cholesterol levels, or physical attributes such as cell morphology [16]. Data-driven approaches to biomarker discovery involve analyzing large volumes of next generation sequencing (NGS) data. Genetic biomarkers such as single nucleotide polymorphisms (SNPs) can be used to identify genetic changes that are associated with specific traits. However, such biomarkers are often not causal but are located in proximity to a causal genomic sequence [17, 18]. Alternately, gene products such as RNA molecules can be quantified and associated with phenotypes. Such gene products are dynamic and must be measured in a specific tissue under specific conditions to be consistent [19]. Gene expression patterns are important because epigenetic changes can cause disease or affect phenotypes without altering any underlying DNA sequences [20, 21]. Identifying patterns of gene expression that are specific to a certain trait or disease thus becomes a valuable technique for biomarker discovery.

Clustering algorithms have demonstrated that RNA expression profiles can sort samples into meaningful groups. Commonly used techniques include principal component analysis and k-means clustering [22, 23]. Another dimensionality reduction technique, t-Distributed Stochastic Neighbor Embedding (t-SNE), has been applied to transcriptomes from 25 human tissue types [24, 25]. This technique has also been applied to data from 19 cancer types, revealing relatedness and distinctions between cancer types [24]. Roche at al. applied Dynamic Quantum Clustering (DQC) [26], a clustering algorithm that utilizes variations in features of the data to reveal relationships between datasets, to RNA expression data from five cancer subtypes. DQC revealed specific biomarkers, but even after removing these biomarkers the data could be clustered into cancer sub-groups using a large number of random genes [27]. Thus, techniques to identify biomarkers that specifically affect biochemical pathways related to a disease or trait may pave the road for personalized medicine, plant and animal breeding, and fundamental biological discovery.

Differential gene expression analysis is a popular technique to compare the expression levels of genes, treated as independent variables, between two or more conditions to identify biomarkers [28]. For example, Lu et al. identified 29 genes whose expression levels could be used to identify tissues that had been exposed to radiation [29]. While this technique is useful for identifying up and down-regulated genes, there are often hundreds to thousands of differentially expressed genes between two conditions, making it difficult to find useful biomarkers. GCN analysis is a holistic approach to deciphering complex gene interaction patterns by performing correlation analysis of gene expression values between biological samples [30]. GCN analysis can be performed across hundreds to thousands of samples to identify robust gene correlation patterns [31]. Thus, mining petabytes of RNA expression data that are publicly available becomes a feasible method for biomarker identification. This dissertation will provide an overview of GCN analysis as a method for discovering polygenic biomarkers and discuss the computational challenges that arise from such efforts.

Gene Coexpression Network Analysis

GCN analysis is a method that can be used to discover polygenic biomarkers. A GCN is a graph in which nodes in the graph represent genes and edges represent connections between genes [32]. Edges are discovered through correlation analysis of all possible pairwise gene combinations in the genome. For each gene to gene comparison, a Spearman or Pearson correlation is typically performed across all available biological samples. A representative GCN edge with a positive correlation is presented in Figure 1.1. A graph of significant edges can then be extracted using hard-threshold techniques such as random matrix theory [31, 33], or soft-threshold techniques such as WGCNA [34]. GCN networks can then be used to identify clusters of genes, modules, which are highly connected to each other in the graph. The link community module (LCM) is a method of module detection that allows for a given node in the GCN to be a member of multiple modules [35]. This makes sense given that genes can be pleiotropic, functioning in different pathways, tissues, and time [36]. GCN modules tend to be co-functional, thus guilt-by-association inferences can be made about the function or regulation of genes in a module [30, 37]. In addition, these GCN modules can serve as polygenic biomarkers. For example, Geschwind et al. identified GCN modules that were specific to brain regions in humans and chimpanzees. While the modules that corresponded to some regions of the brain were well conserved across species, others were not [38]. In this case, the researches constructed GCN modules from different datasets separately and then compared the resulting gene clusters. However, knowledge-independent methods of identifying polygenic biomarkers through GCN analysis can provide the same power while identifying rare biomarkers that would otherwise not be discovered.

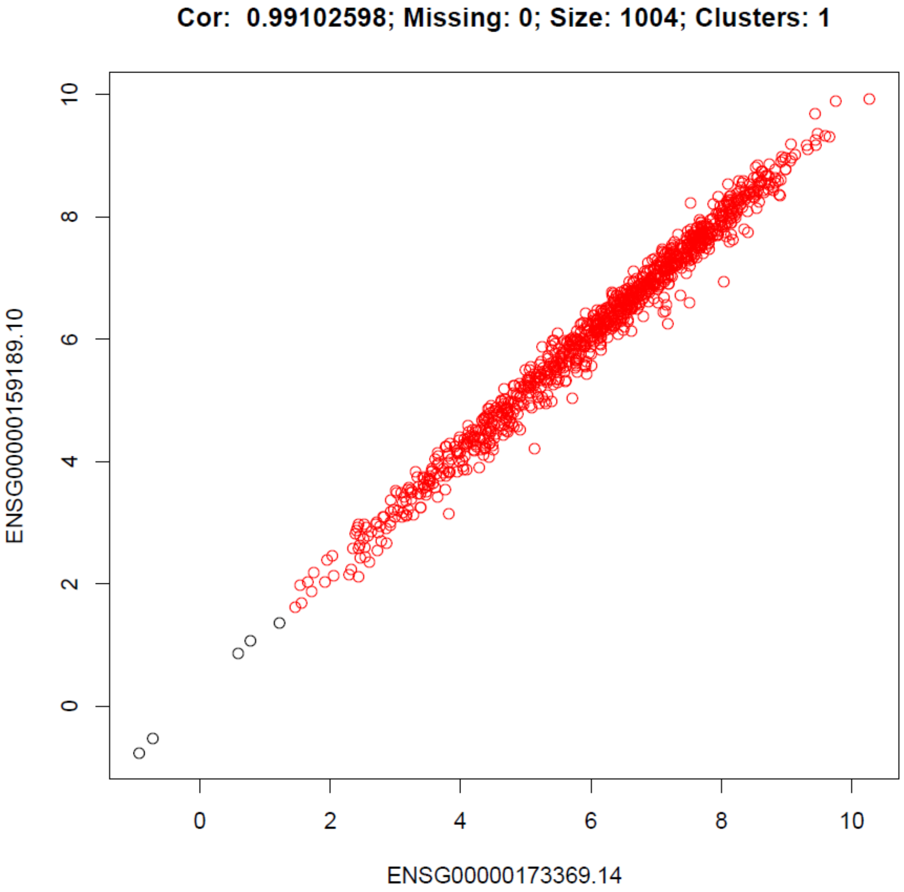


Figure 1.1 A representative GCN edge. In this edge, there is only one cluster of samples present. The resulting correlation value is not specific to any subset of samples or condition, but results in a significant edge in the GCN.

Challenges in GCN analysis include extrinsic noise due to variation in input samples, statistical noise due to inappropriate use of correlation metrics, and technical noise due to error in RNA library preparation, sequencing, or software tools used to quantify gene expression. Correlation metric choice has a large impact on the resulting GCN topology [39]. Furthermore, inappropriate usage of a correlation test can result in false edges in the GCN. For example, a Pearson correlation assumes that the input data is linear, continuous, and free of outliers [40]. In GCN analysis, researchers often conduct Pearson correlations across the genome without checking these assumptions. This issue becomes apparent when a GCN constructed using Pearson correlations is compared to a GCN constructed using Spearman correlations, revealing very little similarity in the resulting networks [41]. These violations may be the result of extrinsic noise in the input datasets, causing multi-modal distributions of expression. One method to address this noise is to cluster input samples and construct separate GCNs from each cluster. Feltus et al. utilized k-means clustering to sort 7,105 expression datasets, then constructed 86 GCNs from the resulting clusters. These 86 GCNs were able to cover 94.7% of the genome space, compared to only 15.9% when one GCN was constructed with all of the data [42]. While this study clustered samples using the gene expression values across the entire genome, it also possible to cluster samples on a gene by gene basis prior to performing correlation analysis. Ficklin et al. demonstrated the use of Gaussian Mixture Models (GMMs) in clustering samples based on the local pairwise gene expression values for a given correlation test [41]. Using this method, clusters were identified in every gene to gene comparison in the genome prior to performing correlation analysis, which results in GCN edges that are specific to subset of the input samples. This resulted in a higher concordance in a GCN constructed using Pearson correlation compared to a GCN constructed using Spearman correlation, demonstrating that this method can be used to reduce the negative impacts of extrinsic noise [41]. During my PhD studies, I contributed to a software development collaboration between Washington State University and Clemson University led by Professor Stephen Ficklin. To address the issues described above, we developed a novel GCN construction algorithm, KINC, that clusters input samples prior to performing correlation analysis on a given gene pair.

Knowledge Independent Network Construction

KINC is a software package that constructs GCNs from heterogeneous expression datasets. As described in Chapter 5, three steps must be executed to build a GCN using KINC: KINC *similarity*, KINC *threshold*, and KINC *extract* (https://github.com/SystemsGenetics/KINC). KINC *similarity* constructs a correlation matrix using the provided input file. Before performing a correlation test on a given gene pair, samples are clustered using GMMs. A correlation test using Spearman or Pearson is then performed on each cluster individually. This process is repeated for every possible pairwise gene combination, which typically reaches billions of comparisons for a eukaryotic genome. This process is computationally demanding, often requiring advanced computing resources to complete in a reasonable amount of time (Chapter 5). Next, KINC *threshold* is performed to identify a significance threshold. For a given gene pair, if the absolute value of the correlation is above the significance threshold, it will be included in the final GCN. KINC uses random matrix theory (RMT) to identify this cutoff. To identify this threshold, RMT iterates through successively lower threshold values and looks for the distribution of eigenvalues in the similarity matrix to change from Gaussian to Poisson. Once the threshold is identified, the GCN can be extracted using the KINC *extract* command.

KINC is unique from other methods of GCN construction because it has the ability to identify condition-specific edges [41]. Condition-specific edges occur when a subset of the samples present in a cluster, and this cluster produces a significant correlation value. In the event that only one cluster of samples is identified, a GCN edge will not be condition-specific. Figure 1.1 demonstrates an example of an edge that is not condition-specific. In contrast, Figure 1.2 demonstrates an edge that is specific to only a subset of the input samples. In this example, two sample clusters were identified. The samples highlighted in red were members of the cluster that produced the significant correlation. Similar edges that are specific to a subset of input samples can be annotated for attributes in the input data. For example, a Fisher's exact test can be performed to test for overrepresentation of a particular attribute in the cluster that produced a given GCN edge. The KINC.R package (https://github.com/SystemsGenetics/KINC.R) facilitates such statistics by providing functions for loading the input GCN and performing edge enrichment tests.

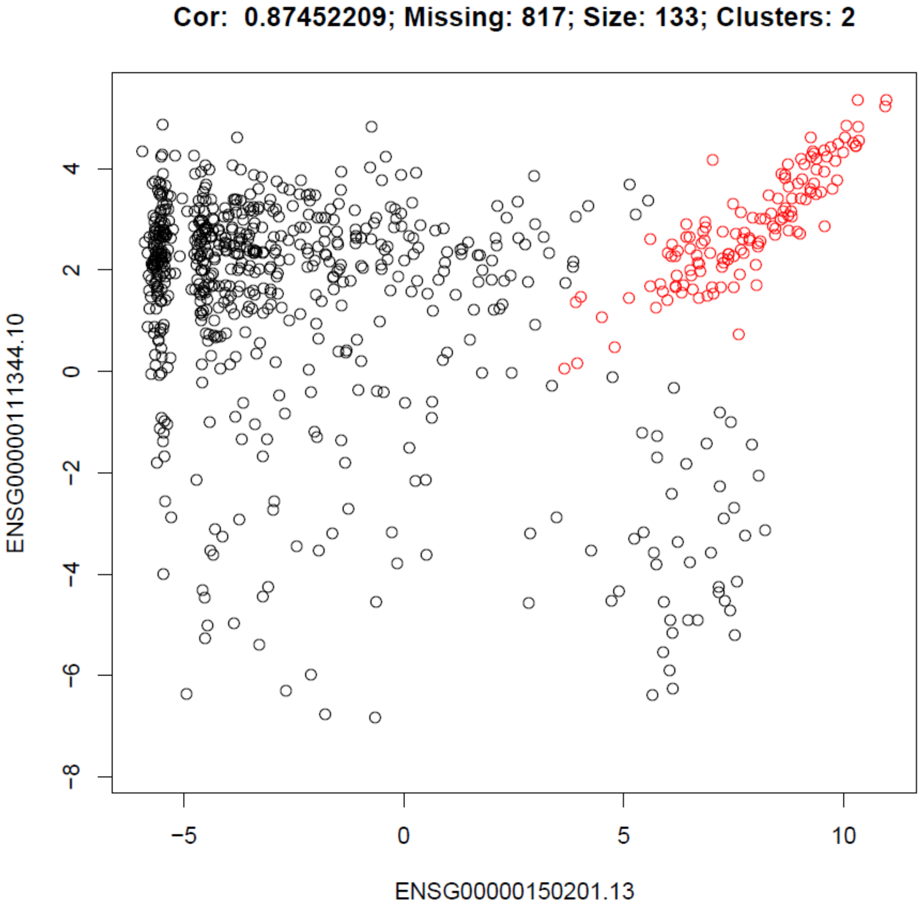


Figure 1.2 A representative condition-specific GCN edge that was generated using the KINC software. The cluster highlighted in red was used to conduct a Spearman correlation, resulting in a significant edge in the resulting GCN. Samples not in this cluster did not produce a significant correlation value.

KINC enables GCNs to be constructed from a large number of diverse datasets. For example, Dunwoodie et al. used KINC to construct GCNs for cancerous and non-cancerous brain samples [43]. By comparing LCM modules from these networks, glioblastoma (GBM)-specific coexpression patterns were identified. The genes from this module were up-regulated and hypomethylated relative to the other brain samples, thus identifying these genes as biomarkers for GBM [43]. Given that KINC can identify condition-specific GCN edges, it is possible to identify rare and specific coexpression patterns from a large dataset without prior knowledge of the underlying sample clusters. Thus, exploiting a large number of publicly available datasets that span hundreds of different conditions becomes possible. Dunwoodie utilized a binary gene detection method to identify a list of biomarker genes by identifying condition-specific GCN modules. However, edge-based grouping is an equally powerful biomarker detection method. In Chapter 2, I discuss the construction of a condition-annotated kidney cancer GCN. The list of biomarkers produced by this study are binary edge-based biomarkers. In Chapter 3, I discuss the construction of a root nodulation GCN, which I utilize to produce a list of biomarker genes from GCN modules, similar to the approach the Dunwoodie et al. utilized. KINC provides a novel method for edge-based biomarker discovery, an approach that is ideal for complex experiments that span thousands of datasets.

Data Mining Resources

Public databases host a wealth of RNA expression data. The Cancer Genome Atlas (TCGA) hosts normalized RNA expression data, mutation profiles, and clinical attributes that correspond to patients from 37 cancer types. In some cases, these samples include data from primary tumors, metastatic tumors, and adjacent normal tissue. In Chapter 1, I constructed a kidney cancer GCN using 1,021 kidney cancer patients, and incorporated DNA mutation data to identify GCN edges that were specific to distinct clinical subtypes of cancer. The data available on TCGA also spans varying tumor stages and patient attributes such as gender, age, ethnicity, alcohol history, etc. These datasets are freely available for download from the TCGA Genomic Data Commons (https://portal.gdc.cancer.gov/). The Genotype-Expression (GTEx) project is another source of human gene expression and mutation data (https://gtexportal.org/home/). This database contains data from over 700 individuals and spans 53 tissues (Figure 1.3). In contrast to the TCGA, all data hosted on GTEx was obtained from healthy tissue. Both TCGA and GTEx provide raw transcript and gene counts in addition to normalized expression values. Thus, comparing expression of tumor samples to normal samples becomes possible at a large scale. Resources such as Gene Expression Profiling Interactive Analysis (GEPIA) have enabled convenient access to pre-computed differential expression analysis through a web interface [44]. Large scale data commons are not limited to human samples. For example, the European Bioinformatics Institute provides an expression data from 791 plant experiments [45, 46].

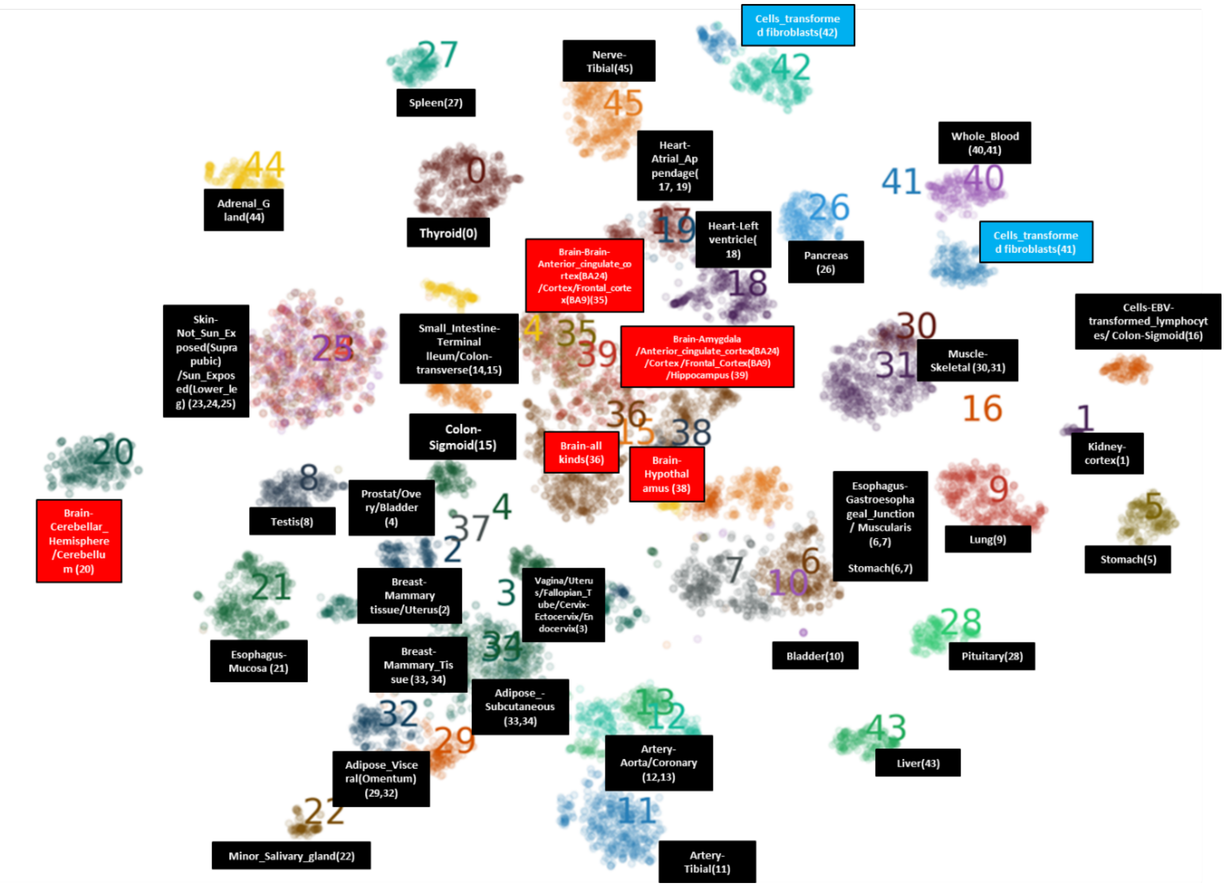


Figure 1.3 Gene Expression Samples available through the GTEx portal. Samples were clustered using 1000 iterations of t-SNE followed by consensus cluster identification. Figure credit: Yuqing “Iris” Hang <yhang@g.clemson.edu>

In addition to databases that host preprocessed expression quantification and mutation profiles, a large volume of raw sequencing datasets is available. It is now common practice to upload sequencing reads to the Sequence Read Archive (SRA), which is hosted by the National Center for Biotechnology Information (https://www.ncbi.nlm.nih.gov/sra/docs/). The SRA hosts over 8 petabytes of raw, unprocessed sequencing reads from hundreds of organisms, which includes over 4 petabytes of public access data (Figure 1.4). The growth of this data has been parabolic over the past decade and will continue to grow as high-throughput sequencing techniques become a staple in molecular biology and clinical diagnostic labs. De novo sequencing experiments are also worthy of data mining. In chapter 3, I discuss the construction of a root nodulation GCN using 30 samples that were generated as part of an NSF-funded grant at Clemson University (PGRP award # 1444461). As the growth of DNA sequencing continues at a rapid rate, the need for stable computing resources grows exponentially.

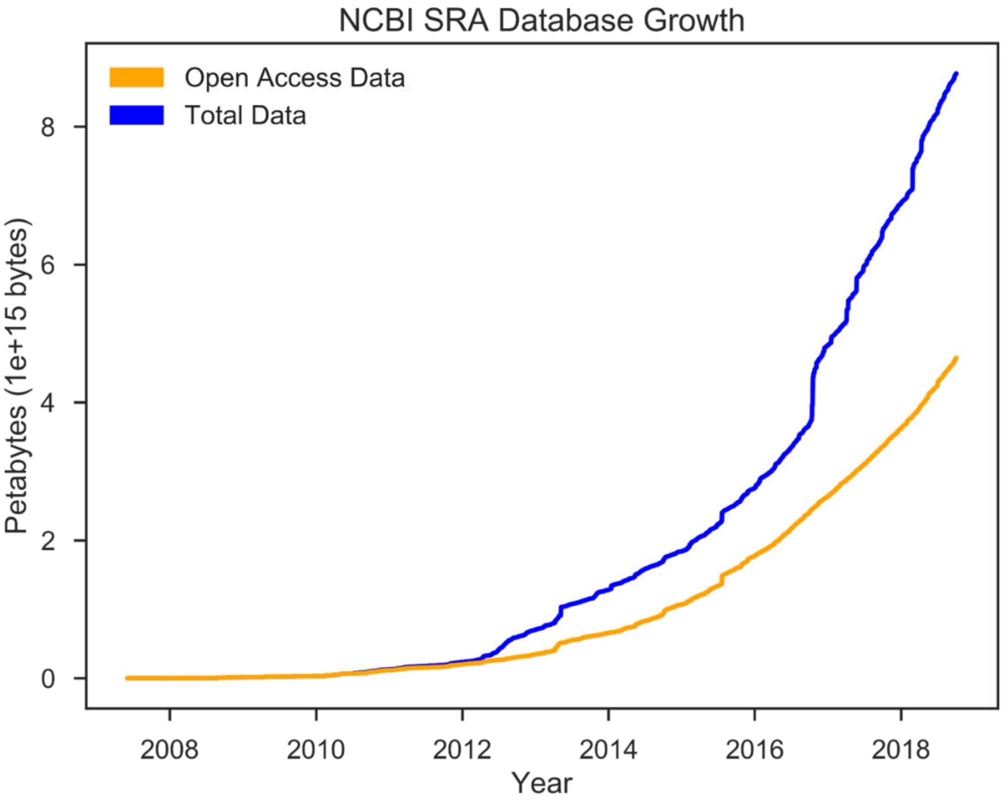


Figure 1.4 Growth of DNA sequencing data available in the SRA archive. This graph was produced using publicly available data (https://www.ncbi.nlm.nih.gov/sra/).

Computational Workflows and Cyberinfrastructure

Processing NGS datasets requires computational resources beyond those available on a desktop computer or laptop. DNA sequencing reads are stored in FASTQ files, which are text files that contain nucleotide strings that represent sequencing reads and quality scores that represent the confidence that a given nucleotide was called correctly [47]. These files typically contain a minimum of 10 million reads, which results in 40 million lines per file. Significant memory (RAM) and disk storage are necessary to process these files. In addition, software tools and computational pipelines are necessary to interpret the data in a manner that becomes useful to the biologist. An example RNA sequencing data processing workflow is presented in Figure 1.5. SRR771467 is a human RNAseq dataset that is stored on the SRA database. This dataset contains 28,013,763 paired-end sequencing reads which comprise 17 GB of uncompressed text files. To process this data, the raw reads are purged of poor quality reads and adapter sequences using Trimmomatic [48]. The resulting clean FastQ files total 15.6 GB in size. These cleaned reads are mapped to the GRCh38 [49] reference genome, resulting in a 18 GB SAM alignment file. This alignment file is filtered, sorted, and compressed into a BAM using Samtools [50], resulting in a file that is 2.7 GB. Transcript abundances are quantified using StringTie [51, 52], producing GTF files that gene expression values can be parsed from. In total, 39 GB of files were produced by processing this single dataset. As experiments scale to hundreds or thousands, terabytes of data are quickly generated. The researcher must locate appropriate computing resources to handle the volume of data that their experiment will generate.

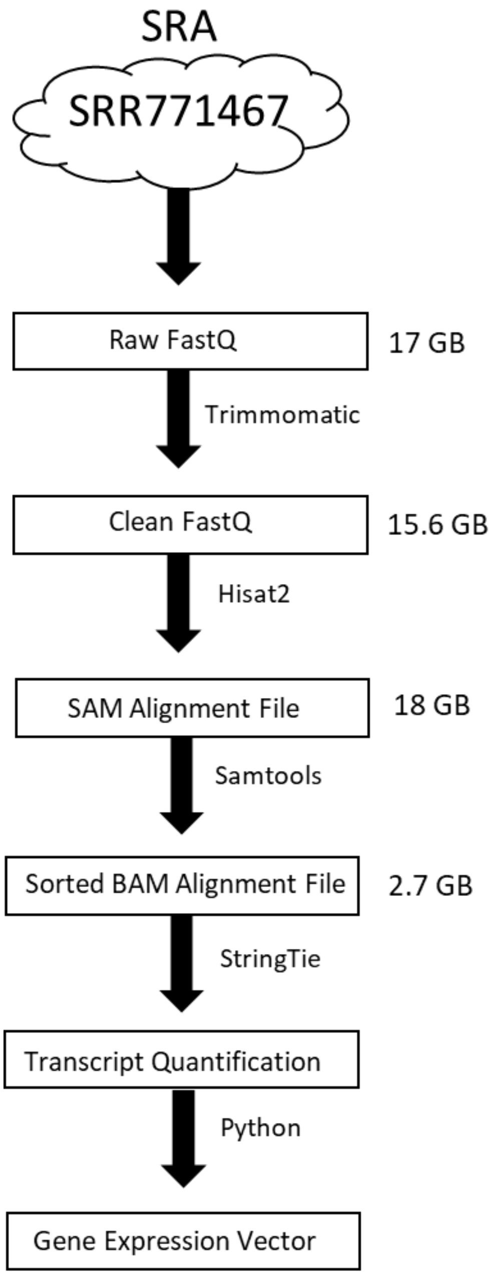


Figure 1.5 An example RNAseq workflow. SRR771467 was processed using Trimmomatic, Hisat2, Samtools, and StringTie software packages.

Computing clusters are a common source of resources for genomic researchers to utilize for NGS data processing. A high performance computing (HPC) cluster is comprised of many computers, referred to as nodes, which are connected to each other through local networks. These nodes can be comprised of diverse computer hardware, with some nodes having large volumes of disk storage and memory [53, 54]. A user can access a computing cluster through a single login host, and have access to hundreds to thousands of computers. Thus, computation can be easily scaled by running multiple tasks as the same time [53]. Tasks are submitted to a scheduler node that decides which computer to send the task to. On university campuses that have HPC clusters, resources are often provisioned equally among users, with user's jobs being placed into queues when all nodes are busy. For example, Clemson University's Palmetto cluster uses the PBS job scheduling system to manage the submission of computing tasks from users. Even though the Palmetto cluster is a Top 200 ranked supercomputer in the world, there is often a saturation of resources, resulting in long wait time for users to complete their tasks. During my PhD research, I encountered bottlenecks in my ability to produce results using the Palmetto Cluster alone. As a result, I was forced to develop cyberinfrastructure skills through collaborations with diverse scientists. Scientists performing terabyte to petabyte scale experiments must obtain additional computing resources such as cloud computing or grid computing.

Grid computing refers to the utilization of geographically distributed computing resources from a remote host [55]. The Open Science Grid (OSG) is a grid computing resource that is available free of charge to US based researchers. Universities across the

United States participate in the OSG. When a node is not being used at a local institution, this resource becomes available to researchers on the OSG. The HTCondor job scheduler enables users to submit thousands of jobs to the OSG, without the user needing to specify where the job will run [56]. A researcher on the OSG has access to these opportunistic resources all across the country, allowing for thousands of compute jobs to be run at the same time, which would not be possible on a local HPC cluster such as the Palmetto Cluster. In chapter 4, I discuss the development of an RNA sequencing data processing workflow that can run on the OSG, and compare this workflow to a comparable workflow on the Palmetto Cluster. However, workflows on the Open Science Grid encounter troubles that HPC clusters such as the palmetto cluster do not. Computing tasks are expected to be small on the OSG, typically only 2 GB of RAM and 10 GB of disk storage are available for a job on the OSG. Thus, users must carefully monitor the progress of jobs that they submit to the OSG, because job failure is common. Job failure occurs when the hardware on a computer fails, or when the owner of the computer reclaims the resource that they own while a job is running. Given that a workflow run on the OSG will typically submit thousands of jobs, monitoring and resubmitting failed jobs by hand becomes nearly impossible. Workflow managers solve this problem by automating error detection and job submission.

A workflow manager is a piece of software that interacts with a job scheduler node to automatically submit and monitor user-defined tasks. When a user submits a workflow that utilizes a workflow manager, they do not have to submit jobs one by one, and they do not have to execute various stages of a pipeline individually. The Pegasus

Workflow Management system is used on the OSG to automate HTCondor job submission [57]. Pegasus handles data movement between resources, job submission, error detection, and task dependencies. A Pegasus workflow is described as an abstract workflow, meaning the execution environment is not defined when a workflow is designed. At the time of workflow submission, Pegasus attaches the execution environments to the workflow, allowing for computing resources to be identified by the workflow. This abstraction allows Pegasus workflows to be portable, meaning they can be executed in different computing environments with minimal modifications. Pegasus is crucial to enabling genomics workflows on the OSG, but the workflows can be difficult to design and implement for new users to the OSG. Thus, developing Pegasus workflows for common genomics applications serves a valuable purpose. NGS sequencing datasets can be processed in parallel by splitting large FastQ files into small pieces to run on the OSG. However, this technique is not common practice in the Genomics community and requires significant testing. In Chapter 4, I discuss the OSG-GEM workflow which splits FastQ files into small pieces to process in parallel. Other genomics workflows such as GCN construction are naturally ideal for the OSG, since billions of correlation tests can be performed independently. In Chapter 5, I discuss the OSG-KINC workflow, which enables massively parallel gene correlation analysis on the OSG. Constructing reproducible and automated workflows for these use-cases will enable researchers to tap into the grid computing resources of the OSG.

A Diverse Contribution to Science

This dissertation presents biological evidence for specific biomarkers involved in kidney cancer and root nodulation. During my studies I contributed to the development of a novel GCN construction algorithm called KINC [41], which enables edges in a GCN to be annotated for specific attributes or conditions. By combining mutational profiles and differential gene expression analysis to the GCNs described in this dissertation, I provide evidence that specific sets of genes are important to kidney cancer and root nodulation. The resulting biomarkers are candidates for functional validation for their potential causative role in these processes. I demonstrate that a holistic approach such as GCN analysis can be incorporated with other common systems biology techniques to distill a list of thousands of genes into a small list of highly specific biomarkers. During this process I encountered computational roadblocks that prevented me from generating results in the timeframe of my PhD studies. As a result, I developed collaborations with computer engineers and scientists across diverse disciplines. Ultimately, the Open Science Grid enabled me to utilize geographically distributed computing resources across the United States to generate these results [58]. I led the development of two automated computational workflows, OSG-GEM and OSG-KINC, which enable researchers to tap into these resources to process large genomic datasets [59, 60]. The approaches described in this dissertation can be applied to the endless wealth of data that is available for mining in online database, as well as data that is generated *de novo* in a molecular biology lab.

References

1. Regenmortel, M.H.V.V., *Reductionism and complexity in molecular biology.* EMBO Reports, 2004. **5**(11): p. 1016-1020.

2. Fang, F.C. and A. Casadevall, *Reductionistic and Holistic Science.* Infection and Immunity, 2011. **79**(4): p. 1401-1404.

3. Siegel, R.L., K.D. Miller, and A. Jemal, *Cancer statistics, 2016.* CA Cancer J Clin, 2016. **66**(1): p. 7-30.

4. Hanahan, D. and R.A. Weinberg, *Hallmarks of cancer: the next generation.* Cell, 2011. **144**(5): p. 646-74.

5. Ricketts, C.J., et al., *The Cancer Genome Atlas Comprehensive Molecular Characterization of Renal Cell Carcinoma.* Cell Rep, 2018. **23**(1): p. 313-326.e5.

6. Hsieh, J.J., et al., *Overcome Tumor Heterogeneity-Imposed Therapeutic Barriers through Convergent Genomic Biomarker Discovery: A Braided Cancer River Model of Kidney Cancer.* Seminars in cell & developmental biology, 2017. **64**: p. 98-106.

7. Hsieh, J.J., et al., *Genomic classifications of renal cell carcinoma: a critical step towards the future application of personalized kidney cancer care with pan-omics precision.* J Pathol, 2018. **244**(5): p. 525-537.

8. Hsieh, J.J., et al., *Renal cell carcinoma.* Nature reviews. Disease primers, 2017. **3**: p. 17009-17009.

9. Taiz, L., *Agriculture, plant physiology, and human population growth: past, present, and future.* Theoretical and Experimental Plant Physiology, 2013. **25**: p. 167-181.

10. NATIONS, F.A.A.O.O.T.U. *Current world fertilizer trends and outlook to 2015*. 2011 [cited 2016 August 15]; Available from: http://www.fao.org/3/a-av252e.pdf.

11. Suzaki, T. and M. Kawaguchi, *Root nodulation: a developmental program involving cell fate conversion triggered by symbiotic bacterial infection.* Curr Opin Plant Biol, 2014. **21**: p. 16-22.

12. Oldroyd, G.E., *Speak, friend, and enter: signalling systems that promote beneficial symbiotic associations in plants.* Nat Rev Microbiol, 2013. **11**(4): p. 252-63.

13. Sun, J., et al., *Crosstalk between jasmonic acid, ethylene and Nod factor signaling allows integration of diverse inputs for regulation of nodulation.* Plant J, 2006. **46**(6): p. 961-70.

14. Boscari, A., et al., *Expression dynamics of the Medicago truncatula transcriptome during the symbiotic interaction with Sinorhizobium meliloti: which role for nitric oxide?* Plant Physiol, 2013. **161**(1): p. 425-39.

15. Wang, E., et al., *A common signaling process that promotes mycorrhizal and oomycete colonization of plants.* Curr Biol, 2012. **22**(23): p. 2242-6.

16. Mayeux, R., *Biomarkers: potential uses and limitations.* NeuroRx, 2004. **1**(2): p. 182-8.

17. Vignal, A., et al., *A review on SNP and other types of molecular markers and their use in animal genetics.* Genet Sel Evol, 2002. **34**(3): p. 275-305.

18. Johnson, A.D., *SNP bioinformatics: a comprehensive review of resources.* Circulation. Cardiovascular genetics, 2009. **2**(5): p. 530-536.

19. Jones, B., *Layers of gene regulation.* Nature Reviews Genetics, 2015. **16**: p. 128.

20. Gao, D., J.G. Herman, and M. Guo, *The clinical value of aberrant epigenetic changes of DNA damage repair genes in human cancer.* Oncotarget, 2016. **7**(24): p. 37331-37346.

21. Allis, C.D. and T. Jenuwein, *The molecular hallmarks of epigenetic control.* Nature Reviews Genetics, 2016. **17**: p. 487.

22. Berkhin, P., *A Survey of Clustering Data Mining Techniques*, in *Grouping Multidimensional Data: Recent Advances in Clustering*, J. Kogan, C. Nicholas, and M. Teboulle, Editors. 2006, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 25-71.

23. Lever, J., M. Krzywinski, and N. Altman, *Principal component analysis.* Nature Methods, 2017. **14**: p. 641.

24. Taskesen, E., et al., *Pan-cancer subtyping in a 2D-map shows substructures that are driven by specific combinations of molecular characteristics.* Scientific Reports, 2016. **6**: p. 24949.

25. van_ der_ Maaten, L.J.P.H., G. E, *Visualizing High-Dimensional Data Using t-SNE.* Journal of Machine Learning Research, 2008. **9**: p. 2579–2605.

26. Weinstein, M. and D. Horn, *Dynamic quantum clustering: a method for visual exploration of structures in data.* Phys Rev E Stat Nonlin Soft Matter Phys, 2009. **80**(6 Pt 2): p. 066117.

27. Roche, K.E., et al., *Sorting Five Human Tumor Types Reveals Specific Biomarkers and Background Classification Genes.* Scientific Reports, 2018. **8**(1): p. 8180.

28. Love, M.I., W. Huber, and S. Anders, *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.* Genome Biology, 2014. **15**(12): p. 550.

29. Lu, T.-P., et al., *Identification of Gene Expression Biomarkers for Predicting Radiation Exposure.* Scientific Reports, 2014. **4**: p. 6293.

30. Wolfe, C.J., I.S. Kohane, and A.J. Butte, *Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks.* BMC Bioinformatics, 2005. **6**: p. 227.

31. Gibson, S.M., et al., *Massive-scale gene co-expression network construction and robustness testing using random matrix theory.* PLoS One, 2013. **8**(2): p. e55871.

32. Butte, A.J., et al., *Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks.* Proc Natl Acad Sci U S A, 2000. **97**(22): p. 12182-6.

33. Luo, F., et al., *Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory.* BMC Bioinformatics, 2007. **8**: p. 299.

34. Langfelder, P. and S. Horvath, *WGCNA: an R package for weighted correlation network analysis.* BMC Bioinformatics, 2008. **9**: p. 559.

35. Ahn, Y.Y., J.P. Bagrow, and S. Lehmann, *Link communities reveal multiscale complexity in networks.* Nature, 2010. **466**(7307): p. 761-4.

36. Paaby, A.B. and M.V. Rockman, *The many faces of pleiotropy.* Trends in genetics : TIG, 2013. **29**(2): p. 66-73.

37. Aoki, K., Y. Ogata, and D. Shibata, *Approaches for extracting practical information from gene co-expression networks in plant biology.* Plant Cell Physiol, 2007. **48**(3): p. 381-90.

38. Oldham, M.C., S. Horvath, and D.H. Geschwind, *Conservation and evolution of gene coexpression networks in human and chimpanzee brains.* Proceedings of the National Academy of Sciences, 2006. **103**(47): p. 17973-17978.

39. Weighill, D.A. and D. Jacobson, *Network Metamodeling: Effect of Correlation Metric Choice on Phylogenomic and Transcriptomic Network Topology*, in *Network Biology*, I. Nookaew, Editor. 2017, Springer International Publishing: Cham. p. 143-183.

40. Schober, P., C. Boer, and L.A. Schwarte, *Correlation Coefficients: Appropriate Use and Interpretation.* Anesthesia & Analgesia, 2018. **126**(5): p. 1763-1768.

41. Ficklin, S.P., et al., *Discovering Condition-Specific Gene Co-Expression Patterns Using Gaussian Mixture Models: A Cancer Case Study.* Sci Rep, 2017. **7**(1): p. 8617.

42. Feltus, F.A., et al., *Maximizing capture of gene co-expression relationships through pre-clustering of input expression samples: an Arabidopsis case study.* BMC Syst Biol, 2013. **7**: p. 44.

43. Dunwoodie, L.J., et al., *Discovery and validation of a glioblastoma co-expressed gene module.* Oncotarget, 2018. **9**(13): p. 10995-11008.

44. Tang, Z., et al., *GEPIA: a web server for cancer and normal gene expression profiling and interactive analyses.* Nucleic Acids Res, 2017. **45**(W1): p. W98-w102.

45. Petryszak, R., et al., *Expression Atlas update—an integrated database of gene and protein expression in humans, animals and plants.* Nucleic Acids Research, 2016. **44**(D1): p. D746-D752.

46. Petryszak, R., et al., *Expression Atlas update—a database of gene and transcript expression from microarray- and sequencing-based functional genomics experiments.* Nucleic Acids Research, 2014. **42**(D1): p. D926-D932.

47. *Andrews S. FASTQC. A quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. Accessed 29 September 2014*.

48. Bolger, A.M., M. Lohse, and B. Usadel, *Trimmomatic: a flexible trimmer for Illumina sequence data.* Bioinformatics, 2014. **30**(15): p. 2114-2120.

49. *GENCODE*.

50. Li, H., et al., *The Sequence Alignment/Map format and SAMtools.* Bioinformatics, 2009. **25**(16): p. 2078-2079.

51. Pertea, M., et al., *StringTie enables improved reconstruction of a transcriptome from RNA-seq reads.* Nature biotechnology, 2015. **33**(3): p. 290-295.

52. Pertea, M., et al., *Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie, and Ballgown.* Nature protocols, 2016. **11**(9): p. 1650-1667.

53. G.Sravanthi, B.Grace, and V.kamakshamma, *A review of High Performance Computing.* IOSR Journal of Computer Engineering (IOSR-JCE), 2014. **16**(1): p. 36-43.

54. Qian, D., *High performance computing: a brief review and prospects.* National Science Review, 2016. **3**(1): p. 16-16.

55. Talukdar, V., et al., *Changing from computing grid to knowledge grid in life-science grid.* Biotechnol J, 2009. **4**(9): p. 1244-52.

56. Thain, D., T. Tannenbaum, and M. Livny, *Distributed computing in practice: the Condor experience: Research Articles.* Concurr. Comput. : Pract. Exper., 2005. **17**(2-4): p. 323-356.

57. Deelman, E., et al., *Pegasus, a workflow management system for science automation.* Future Gener. Comput. Syst., 2015. **46**(C): p. 17-35.

58. Ruth, P., et al., *The open science grid.* Journal of Physics: Conference Series, 2007. **78**(1): p. 012057.

59. Poehlman, W.L., et al. *OSG-KINC: High-throughput gene co-expression network construction using the open science grid*. in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2017.

60. Poehlman, W.L., et al., *OSG-GEM: Gene Expression Matrix Construction Using the Open Science Grid.* Bioinformatics and Biology Insights, 2016. **10**(5814-BBI-OSG-GEM:-Gene-Expression-Matrix-Construction-Using-the-Open-Science-Gr.pdf): p. 133-141.

CHAPTER TWO

LINKING BINARY GENE RELATIONSHIPS TO DRIVERS OF RENAL CELL CARCINOMA REVEALS CONVERGENT FUNCTION IN ALTERNATE TUMOR PROGRESSION PATHS

William L. Poehlman[1], James J. Hsieh[2], and F. Alex Feltus[1]

[1]Clemson University Department of Genetics & Biochemistry, Clemson SC, USA.
[2]Molecular Oncology, Department of Medicine, Siteman Cancer Center, Washington University, St Louis, MO, USA.

Submitted October 21st: *Cancer Research*

Abstract

Renal cell carcinoma (RCC) subtypes are characterized by distinct molecular profiles. Using RNA expression profiles from 1,009 RCC samples, we constructed a condition-annotated gene coexpression network (GCN). The RCC GCN contains binary gene coexpression relationships (edges) specific to conditions including RCC subtype and tumor stage. As an application of this resource, we discovered RCC GCN edges and edge sets (modules) that were associated with genetic lesions in known RCC driver genes, including VHL, a common initiating clear cell RCC (ccRCC) genetic lesion, and PBRM1 and BAP1 which are early genetic lesions in the Braided Cancer River Model (BCRM). Since ccRCC tumors with PBRM1 mutations respond to chemotherapy differently than tumors with BAP1 mutations, we focused on ccRCC-specific edges associated with tumors that exhibit alternate mutation profiles: VHL-PBRM1 or VHL-BAP1. We found specific blends of genes and molecular functions associated with these two mutation paths. Despite these mutation-associated edges having unique genes, they

were enriched for the same immunological functions suggesting a convergent functional role for alternate gene sets consistent with the BCRM. The condition annotated RCC GCN described in this report is a novel data mining resource for the assignment of polygenic biomarkers and their relationships to RCC tumors with specific molecular and mutational profiles.

Introduction

Renal cell carcinoma (RCC) is a type of cancer that originates from tubular epithelial cells of the kidney. Subtypes of RCC – clear cell, papillary, and chromophobe– demonstrate unique molecular and histological profiles (1). In recent years, hundreds of RCC tumors from The Cancer Genome Atlas (TCGA; (2,3)) and other sources have been deeply analyzed for genes underlying tumor etiology and progression. While many biomarkers have been associated with RCC, there are few causal genes with consistent and stable genetic lesions driving RCC.

In the case of the most common RCC subtype, ccRCC, several biomarkers have been discovered with variable prevalence between individual tumors. The VHL gene is a common initiating mutation, leading to an accumulation of lipids and glycogens in the tissue (4). Loss of VHL function is insufficient to develop ccRCC. Epigenetic regulators such as PBRM1 and BAP1 – which act as tumor suppressors – are frequently mutated and associated with distinct clinical outcomes in ccRCC patients (5). Loss of function of another chromatin-modifying gene – KDM5C – is also associated with unique clinical outcome (6). BAP1 mutations occur at a near mutually exclusive manner from PBRM1 mutations, and tumors respond to standard of care molecularly-targeted drugs differently depending on which mutations they have (6,7). Other common ccRCC mutations include a histone methyltransferase – SETD2 – and the mTOR kinase which plays key roles in cell growth (8). These biomarkers are clearly relevant to understanding ccRCC biology, but aberrations in these genes are not always consistent between tumors and probably do not fully explain ccRCC tumor progression.

Biomarker inconsistency, a prime motivation for personalized medicine, can partly be attributed to tumor heterogeneity which is a genotyping challenge given that certain regions of a tumor may contain mutations that are unique from other regions of the tumor. A Braided Cancer River Model (BCRM) has defined stages of mutation accumulation that lead to clear cell RCC (ccRCC) (9): initiating, early, intermediate, and speedy mutations. A key aspect of this model is that genetic pathways can operate in parallel to drive tumorigenesis, suggesting that mutations in different genes at various stages of the model can result in convergent evolution of cancer cells (7,9). Thus, targeting parallel genetic pathways with similar phenotypic outputs becomes a challenge in treating and preventing cancer. Polygenic biomarker discovery may provide insight on these parallel pathways and suggest possible therapeutic targets. Given that mutations in chromatin-modifying genes will greatly alter mRNA expression levels (4), identifying RCC-subtype specific gene expression patterns may pave the way for more robust drug targeting.

One method to discover novel biomarkers is through gene coexpression network (GCN) analysis. A GCN is a graph of nodes and edges, where nodes are gene products (e.g. mRNA) and edges are binary relationships between genes (e.g. Spearman correlation). A network of significant edges can be extracted using random matrix theory (RMT) (10,11) or a via soft thresholding to identify functional modules as per WGCNA (12). Gene modules of tightly connected nodes are partitioned from the GCN using techniques such as link communities (13). Modules are then tested for enrichment in known biochemical activity, allowing inference of novel gene function (14,15). Knowledge Independent Network Construction (KINC) is a software package that builds GCNs and tracks the conditions under which significant edges exist (16). Prior to performing correlation analysis for a given gene pair, KINC uses Gaussian Mixture Models (GMMs) to detect one or more sample clusters in the pairwise expression data. Each sample cluster in each pairwise gene comparison is tested for correlation. This procedure reduces extrinsic noise due to sample variation, and since the samples are tracked it is possible to test each edge for overrepresentation of an attribute or condition (e.g. sex, tumor subtype, tumor stage). For example, Dunwoodie et al. used KINC to identify a gene coexpression module that is specific to glioblastoma, an aggressive form of brain cancer (17). Thus, KINC is an appropriate method to discover condition specific gene relationships in a complex mixture of gene expression profiles.

The purpose of this study was to use KINC to identify RCC subtype-specific GCN edges. In addition, we searched for GCN edges specific to tumors with co-occurring mutations in known genes relevant to ccRCC. The GCN was constructed from 1,009 RCC RNAseq datasets from TCGA which included the three major RCC subtypes. These datasets span various tumor stages as well as clinical attributes such as gender and vital status. We assigned GCN edges to ccRCC tumor subsets that have accumulated specific sets of known driver mutations.

Materials and Methods

Input Data and Gene Expression Matrix Construction. All available gene expression quantification (FPKM) files for TCGA-KIRC, TCGA-KIRP, and TCGA-KICH patients were downloaded in May 2018 using the CentOS7 binary distribution of the GDC Data Transfer Tool [https://gdc.cancer.gov/access-data/gdc-data-transfer-tool]. 1,021 samples were downloaded – each containing measurement of 60,483 genes – and aggregated into a gene expression matrix (GEM). The preprocessCore R library was used to preprocess the input GEM (23). Following log base 2 transformation of the data, outlier samples were detected using a Kolmogorov-Smirnov test (KS Dval > 0.15). A total of 12 outlier samples were removed, and the matrix was quantile normalized to reduce technical noise. Clinical annotations were downloaded directly from the GDC website [https://portal.gdc.cancer.gov/]. Mutation profiles for 843 RCC patients were downloaded from Supplemental Table 1 of Ricketts et al. (22). This table provides mutation profiles for the 16 genes listed in Table 2.2. All disruptive mutation types were converted to a simple “Mutation/No Mutation” attribute prior to edge enrichment. In the event that a sample in the RCC GEM was not present in this mutation table, all 16 genes were annotated as “No Mutation”. For co-occurring mutation tests, patients with VHL mutations and mutually exclusive mutations in PBRM1 and BAP1 were assigned the “Mutation” attribute.

Sample Clustering. One thousand iterations of t-SNE were performed using the parallel Python implementation [https://github.com/DmitryUlyanov/Multicore-TSNE]. A perplexity of 30 was used. Clustering of each embedding was performed using the HDBSCAN Python library [https://pypi.python.org/pypi/hdbscan/]. Consensus clusters were identified using the Cluster_Ensembles Python library[https://pypi.org/project/Cluster_Ensembles/], with a minimum cluster size of 10.

Gene Co-expression Network Construction. The OSG-KINC workflow [https://github.com/feltus/OSG-KINC](24) was utilized to execute 50,000 KINC similarity jobs on the Open Science Grid with the following arguments: '*./kinc similarity --method sc --clustering mixmod --criterion ICL --min_obs 30 --th 0'*. Output was transferred to Clemson University's Palmetto Cluster and uncompressed. KINC threshold was executing using the following arguments: '*./kinc threshold --min_csize 30 --clustering mixmod --method sc --th_method sc --th 0.95 --max_modes 5'*. A significance threshold of 0.819100 was identified and the GCN was extracted using the following KINC extract arguments: '*./kinc extract --min_csize 30 --clustering mixmod --method sc --th_method sc --th 0.819100 --max_modes 5'*. A representative GCN edge can be found in Supplemental Figure 2.2.

Edge Enrichment Analysis. Edge enrichment for mutations and clinical attributes was performed using the KINC.R package [https://github.com/SystemsGenetics/KINC.R].

Mutations were coded as present or absent in a tumor according to annotations found in (22). For co-occurring mutation enrichment, a “Mutation” tumor had to have both VHL-PBRM1 (but no BAP1) or VHL-BAP1 (but no PBRM1) mutations. A Fisher’s exact test with a Hochberg p-value correction was used as the default arguments to the *analyzeNetCat* function. Edges were considered to be significantly enriched for a given attribute or set of attributes if the adjusted p value was less than 0.001. Due to the low number of tumors with co-occurring mutation groups (106 VHL/PBRM1, 28 VHL/BAP1), only edges with a cluster size of 250 or smaller were considered for Table 2.3 and Table 2.4.

Module Detection and Enrichment Analysis. Link Community Modules (25) were detected using the linkcomm R package (21). The “single” hcmethod was used with a minimum module size of 3 edges. Functional enrichment of LCM modules as performed using the FUNC-E package [https://github.com/SystemsGenetics/FUNC-E], which uses a Fisher’s exact test similar to the David method of functional enrichment (26). For cross-module comparisons, enriched terms were considered significant if the Fisher’s P value was less than 0.001.

<u>Results</u>

We downloaded and parsed 1,021 gene expression quantification files representing clear cell renal cell carcinoma (KIRC), papillary renal cell carcinoma (KIRP), and chromophobe renal cell carcinoma (KICH) into a 1,021 x 60,483 gene expression matrix (GEM). The GEM contained 860 samples that are annotated for specific tumor stages and 128 samples that are annotated as "Solid Tissue Normal". In addition, there are 33 primary tumor samples that were not annotated for a specific tumor stage. The matrix was log base 2 transformed and 12 outlier samples were removed. Following quantile normalization of the GEM, we performed 1,000 iterations of t-distributed stochastic neighbor embedding (t-SNE) (18) and circumscribed clusters using HDBSCAN(19) and the Cluster Ensembles method (20) (Figure 2.1). Four clusters were identified: Cluster 1 (solid tissue normal enriched; FDR = 4.03E-67); Cluster 2 (KIRP enriched; FDR = 4.88E-83); Cluster 3 (KICH enriched; FDR= 6.84E-40); and Cluster 4 (KIRC enriched; FDR = 5.32E-70). The sample to cluster assignment is available in Supplemental Table 2.1.

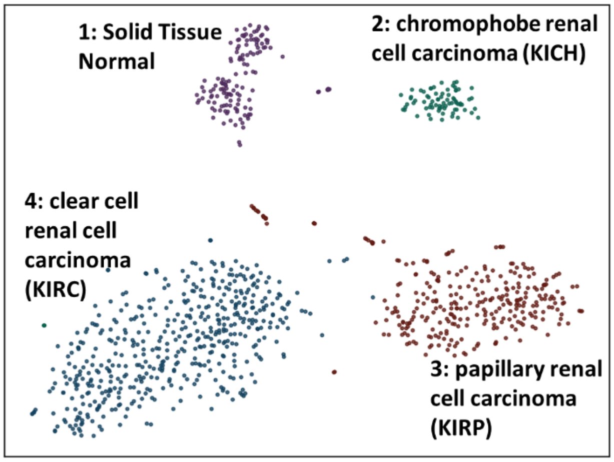


Figure 2.1 Overview of TCGA RCC Expression Data. A total of 128 "solid tissue normal" kidney samples and 860 "primary tumor" samples with were used in this study. Shown are four consensus clusters each with a unique color identified from 1000 t-SNE runs.

Using the preprocessed GEM as input, a condition-annotated GCN was constructed using KINC. This RCC GCN contains 4,121 nodes, 10,451 edges, and demonstrates scale-free topology (R2=0.933; Figure 2.2). Edges in the GCN were tested for enrichment of cancer type, tissue type, tumor stage, and vital status (Table 2.1). The RCC GCN with enrichment p-values for every edge is available in Supplemental Table 2.2. Edges that were enriched (adj. $p < 0.001$) for "Solid Tissue Normal" were extracted to produce a "non-tumor" GCN (Supplemental Table 2.3). Edges that were enriched for "Primary Tumor" were extracted to produce a "tumor" GCN (Supplemental Table 2.4). The non-tumor GCN had 1416 nodes and 3605 edges. The tumor GCN had 623 nodes and 2361 edges (Supplemental Figure 2.1). The number of condition-enriched edges in each of the three GCNs is shown in Table 2.1.

Link community modules (LCM) were identified for each GCN ((21); Supplemental Table 2.5), and functional enrichment tests were performed on each module. Each GCN contains LCMs that were enriched for GO, Reactome, MIM, Pfam, and Interpro annotations. A full list of functionally enriched modules in the RCC GCN is available in Supplemental Table 2.6. Notably, the non-tumor GCN contains LCM modules that are enriched (Fisher's Pval < 0.01) for terms related to MET signaling, which is absent in the RCC GCN. The RCC and non-tumor GCN both have modules enriched for VEGF and Notch signaling (Supplemental Tables 2.7 & 2.8).

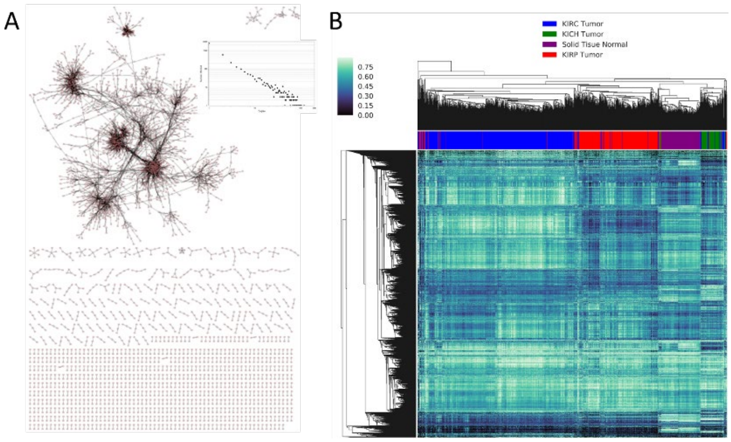


Figure 2.2 Renal cell carcinoma (RCC) gene coexpression network. (A) The RCC GCN demonstrates scale-free topology and contains 4,121 nodes and 10,451 edges. (B) A gene expression intensity heatmap of the 4,121 genes is shown.

To test if edges where specific to tumors with mutations in known RCC genes, we downloaded somatic mutation profiles for 16 genes that are relevant to RCC (22) and detected edges enriched in known ccRCC driver mutations (Table 2.2). In order to place mutations into a BRCM mutation context, we next identified edges in the tumor GCN that were specific to patients with co-occurring VHL and BAP1 mutations (Table 2.3). In addition, we identified edges in the tumor GCN that are specific to patients with co-occurring VHL and PBRM1 mutations (Table 2.4).

While some genes are common to the two edge lists in Tables 2.3-3.4 (CD96, SH2D1A SIRPG, SLA2, SLAMF6), each list contains unique genes that are members of the tumor GCN. Comparing the genes in Table 2.3 to the genes in Table 2.4 reveals similar biological function. Enrichment (Fisher's Pval < 0.001) of GO terms related to T cell activation and immune response are shared between these lists: adaptive immune response (GO:0002250), T cell activation (GO:0042110), positive regulation of natural killer cell mediated cytotoxicity (GO:0045954), and regulation of immune response (GO:0050776).

Discussion

We constructed a condition-annotated RCC GCN and detected edges that are specific to cancer subtype, tissue type, tumor stage, and unique mutation profile. This GCN is a novel data-mining resource for polygenic biomarker assignment to clinically relevant RCC sub-types. To link novel genes to known drivers of ccRCC, we identified 8 edges that are specific to KIRC primary tumors that contain VHL and BAP1 mutations and compared these to 27 edges that are specific to KIRC primary tumors that contain VHL and PBRM1 mutations. These expanded ccRCC driver mutations represent two possible selection routes through the BCRM. We demonstrate that the tumor GCN edges associated with these two sets of mutations contain different genes with similar biological function. Thus, two unique sets of genes can be regulated and selected for in different tumors yielding the same functional result.

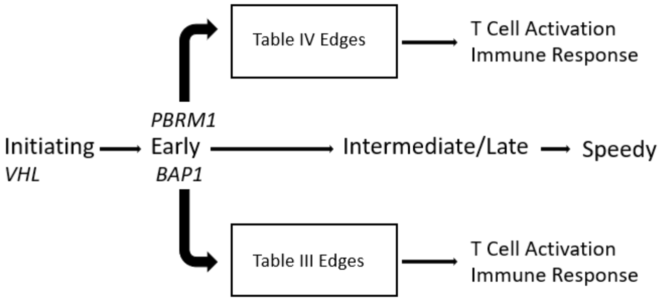


Figure 2.3 Convergent Gene Coexpression Functions in the Braided Cancer River Model. The Braided Cancer River Model was expanded to include gene coexpression function. GCN edges specific to patients with common ccRCC mutation profiles are enriched for functional annotation terms associated with T cell activation and immune response.

Several of the GCN edges associated with mutated gene sets are associated with T cell activation and immune response. The genes in Table 2.3 and Table 2.4 are both enriched for the following GO ontology terms: adaptive immune response

(GO:0002250), T cell activation (GO:0042110), positive regulation of natural killer cell mediated cytotoxicity (GO:0045954), and regulation of immune response (GO:0050776). Identifying ccRCC edges associated with these functions supports the finding of Ricketts et al. (22) that immune signatures related to T cell response are up-regulated in ccRCC compared to other RCC subtypes.

Regardless of whether the patient has co-occurring VHL and BAP1 mutations or co-occurring VHL and PBRM1 mutations, T cell activation genes form coordinated co-expression in the tumor (Figure 2.3). It has been shown that T cell exhaustion occurs when T cells are chronically activated due to infection or inflammation (23). Over time, the T cells lose their function due to increased expression of inhibitory receptors (24,25). We hypothesize that T cell exhaustion is a component in the progression of ccRCC. As evidence, we present binary gene relationships in Table 2.3 that have been characterized for their role in T cell exhaustion in cancer. TIGIT is an inhibitory receptor that is expressed on the surface of T cells and is associated with poor prognosis in melanoma patients (24,26). TIGIT is often co-expressed with LAG3, an inhibitory receptor that migrates to the surface of T cells during chronic inflammation, contributing to T cell dysfunction (24,27,28). While LAG3 is not present in Table 2.3 or Table 2.4, we detected seven KIRC-specific edges that contain LAG3 (Supplemental Table 2.2), implicating this gene in ccRCC regardless of tumor mutation path. We also found TIGIT to be coexpressed with SH2D1A and SLAMF6 in Table 2.3, which are coexpressed with UBASH3A in Table 2.4. SH2D1A is a lymphocyte-activating protein that interacts with SLAMF6 to stimulate natural killer (NK) and T cell activity (29-31). SLA2 — a transcription factor that controls expression of genes that regulate T cell development (32) — is also present in Table 2.3 and Table 2.4.

Further, Table 2.4 contains unique cancer biomarkers that are involved in T cell function. LCK is a tyrosine kinase that functions in normal T-cell development. When this gene becomes mutated and the protein becomes overexpressed, it becomes a proto-oncogene by promoting cellular proliferation and immortality (33). UBASH3A is a T-cell ubiquitin ligand protein that disrupts T cell receptor signaling by promoting accumulation of inhibitory receptors and T cell apoptosis under certain conditions (34). Overexpression of UBASH3A is associated with poor prognosis in metastatic breast cancer (35), and the gene is also associated with autoimmune disorders such as Lupus Erythematosus (36). UBASH3A is present in 14 of the 27 edges in Table 2.4, highlighting its importance in ccRCC patients with co-occurring VHL and PBRM1 mutations. It is coexpressed with CD96, an immune checkpoint receptor that plays inhibitory roles in NK cell activity (37). As we found in Table 2.3, CD96 is expressed on the surface of T cells with TIGIT, which has also demonstrated inhibitory effects on NK cell function in addition to contributing to T cell exhaustion (38,39). We also found UBASH3A to be coexpressed with a surface antigen expressed on the surface of T cells, CD2, which has been found to play key roles in NK cell stimulation (40). Other T cell receptor proteins that we found to be coexpressed with UBASH3A include CD3D and CD3E, which play positive roles in lymphocyte production (41). The tumorigenic role of UBASH3A should be further explored given its dominant presence in the edges of Table 2.4. Given that different sets of mutations are associated with unique edges in Tables 2.3

and 2.4 that are related to T cell function, we have extended the BCRM to include GCN edges that demonstrate convergent function (Figure 2.3).

Interestingly, Table 2.4 contains 11 non-coding RNA genes: DARS-AS1, RP11-789C17.5, AC116366.6, CCDC147-AS1, RP11-981G7.6, AF064858.3, AC073115.2, AF064858.1, AC073115.7, AC011352.3, and AC011352.1. Non-coding RNAs are thought to play key roles in cancer by altering gene expression levels through recruitment of chromatin-modifying enzymes or by directly targeting RNA-binding proteins (42,43). Notably, the antisense RNA DARS-AS1 was found to be correlated with TCRGC2, a T cell receptor(44) gene, suggesting that this non-coding RNA might play a role in suppressing healthy T cell function. We also detected four edges: RP11-981G7.6-LINCR-0001, AF064858.3- AF064858.1, AC011352.3- AC011352.1, and AC073115.2-AC073115.7 that are each comprised of two long non-coding RNAs that are correlated with each other. We speculate that these non-coding RNAs are targeting parallel genetic pathways during cancer development as per the BCRM. Identification of similar GCN edges can help tackle the challenge of tumor heterogeneity by identifying novel genes and pathways that synchronously contribute to the hallmarks of cancer.

The condition-annotated GCNs described in this report provide a novel data-mining resource for discovering polygenic biomarkers of RCC. By linking edges to mutations in specific genes, we provide a framework for identifying edges that are relevant to specific clinical subtypes of RCC. In addition, this provides a resource for patients who may have genotyped tumors – but no RNA expression data — to link somatic mutations with therapeutic targets developed from genes in this GCN.

Interestingly, the non-tumor GCN is larger than the tumor GCN and has a larger number of condition-specific edges. It is possible that accumulation of driver mutations in the tumor results in gene expression changes in adjacent normal tissue. These gene expression changes may lead to metastasis, tumor growth, or recurrence. Thus, in addition to edges in the tumor GCN, edges in the "non-tumor" GCN may be important biomarkers or potential therapeutic targets.

While this report focused on edges associated with ccRCC driver mutations, the ccRCC-specific edges that were not mutation-associated are worthy of further exploration. For example, one could model these edges in the context of tumor stages as a "time-series" to identify GCN edge patterns acquired or lost during tumor development. With genome-wide mutation profiles, a deeper analysis could test for edge associations beyond the handful of known mutation drivers examined in this report. Finally, our GCN analysis focused on ccRCC but is applicable to other RCC subtypes. We detected 103 edges that are specific to KIRP tumors and 37 edges that are specific to KICH tumors. We suspect that fewer edges were detected for these RCC subtypes due to the smaller number of available TCGA samples relative to ccRCC patients. Regardless, exploration of these additional binary biomarkers is a valuable resource for characterizing the differential molecular and histological presentation of RCC subtypes.

Acknowledgments

This work utilized Clemson University’s Palmetto Cluster and the Open Science Grid. The OSG is supported by the National Science Foundation and the U.S. Department of Energy’s Office of Science. J.J.H. is supported by NIH R01 CA223231. J.J.H. is a paid consultant for Eisai Inc. We acknowledge the assistance of M. Rynge, D. Balamurugan, and the OSG support staff for technical support and assistance. We also acknowledge Bill Paseman for bringing his valuable insights and inspiration to this study.

Tables

Table 2.1. GCN Topology & Attribute-Enriched Edges

	RCC-GCN	Tumor-GCN	Normal-GCN
Nodes	4121	623	1416
Edges	10451	2361	3605
<k>	5.066	7.576	5.089
R2	0.933	0.838	0.850
Patient KIRC	6288	1909	2362
Paitent KIRP	275	103	50
Patient KICH	1807	37	1651
Primary_Tumor	2361	2361	0
Solid_Tissue_Normal	3605	0	3605
Tumor_stage_i	54	16	20
Tumor_stage_ii	129	3	100
Tumor_stage_iii	432	22	385
Tumor_stage_iv	1770	24	1697
VitalStatus_alive	9	1	7
VitalStatus_dead	2620	280	1987

Table 2.2. GCN Edge-RCC mutation Association

Mutation	Gene Description	RCC-GCN	Tumor-GCN	Normal-GCN
VHL	von Hippel-Lindau tumor suppressor	5282	1755	2330
PBRM1	polybromo 1	4254	1362	2274
SETD2	SET domain containing 2	265	67	170
KDM5C	lysine demethylase 5C	41	33	1
BAP1	BRCA1 associated protein 1	41	29	0
PTEN	phosphatase and tensin homolog	1	0	0
MTOR	mechanistic target of rapamycin kinase	441	31	386
TP53	tumor protein p53	154	4	121
PIK3CA	PI3-kinase catalytic subunit alpha	3	2	0
MET	MET proto-oncogene, RTK	16	1	9
FAT1	FAT atypical cadherin 1	0	0	0
NF2	neurofibromin 2	2	0	0
KDM6A	lysine demethylase 6A	3	0	0
SMARCB1	SWI/SNF related	1	0	0
NFE2L2	nuclear factor, erythroid 2 like 2	2	0	1
STAG2	stromal antigen 2	0	0	0

Table 2.3. KIRC Tumor Edges Associated with Co-Occuring VHL and BAP1 Mutations

GeneA	GeneB	GeneA Description	GeneB Description	Module	Notes
ENSG00000183918;SH2D1A	ENSG00000181847;TIGIT	SH2 domain containing 1A	T cell immunoreceptor with Ig and ITIM domains	TM0006	&,:
ENSG00000181847;TIGIT	ENSG00000162739;SLAMF6	T cell immunoreceptor with Ig and ITIM domains	SLAM family member 6	TM0006	&,:
ENSG00000181847;TIGIT	ENSG00000153283;CD96	T cell immunoreceptor with Ig and ITIM domains	CD96 molecule	TM0006	&,‡,†,*
ENSG00000181847;TIGIT	ENSG00000101082;SLA2	T cell immunoreceptor with Ig and ITIM domains	Src like adaptor 2	TM0006	&,‡,†,*
ENSG00000198846;TOX	ENSG00000049249;TNFRSF9	thymocyte selection associated high mobility group box	TNF receptor superfamily member 9	NA	&,‡,†,*
ENSG00000153563;CD8A	ENSG00000049249;TNFRSF9	CD8a molecule	TNF receptor superfamily member 9	NA	&,‡,†,*
ENSG00000163508;EOMES	ENSG00000049249;TNFRSF9	eomesodermin	TNF receptor superfamily member 9	NA	&,‡,†,*
ENSG00000181847;TIGIT	ENSG00000089012;SIRPG	T cell immunoreceptor with Ig and ITIM domains	signal regulatory protein gamma	NA	&,‡,†,*

& = Spearman Correlation > 0.80; ‡ = Padj KIRC Patient < 0.001; † = Padj Primary Tumor < 0.001; * = Padj VHL and BAP1 Mutations < 0.001

Table 2.4. KIRC Tumor Edges Associated with Co-Occuring VHL and PBRM1 Mutations

GeneA	GeneB	GeneA Description	GeneB Description	Module	Notes
ENSG00000160185;UBASH3A	ENSG00000153283;CD96	ubiquitin associated and SH3 domain containing A	CD96 molecule	TM0023	&,‡,†,*
ENSG00000183918;SH2D1A	ENSG00000160185;UBASH3A	SH2 domain containing 1A	ubiquitin associated and SH3 domain containing A	TM0023	%,‡,†,*
ENSG00000162739;SLAMF6	ENSG00000160185;UBASH3A	SLAM family member 6	ubiquitin associated and SH3 domain containing A	TM0023	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000101082;SLA2	ubiquitin associated and SH3 domain containing A	Src like adaptor 2	TM0023	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000116824;CD2	ubiquitin associated and SH3 domain containing A	CD2 molecule	TM0021	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000089012;SIRPG	ubiquitin associated and SH3 domain containing A	signal regulatory protein gamma	TM0021	&,‡,†,*
ENSG00000277734;TRAC	ENSG00000160185;UBASH3A	T cell receptor alpha constant	ubiquitin associated and SH3 domain containing A	TM0021	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000137078;SIT1	ubiquitin associated and SH3 domain containing A	signaling threshold regulating transmembrane adaptor 1	TM0021	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000147168;IL2RG	ubiquitin associated and SH3 domain containing A	interleukin 2 receptor subunit gamma	TM0021	&,‡,†,*
ENSG00000167286;CD3D	ENSG00000160185;UBASH3A	CD3d molecule	ubiquitin associated and SH3 domain containing A	TM0021	&,‡,†,*
ENSG00000182866;LCK	ENSG00000160185;UBASH3A	LCK proto-oncogene, Src family tyrosine kinase	ubiquitin associated and SH3 domain containing A	TM0021	&,‡,†,*
ENSG00000198851;CD3E	ENSG00000160185;UBASH3A	CD3e molecule	ubiquitin associated and SH3 domain containing A	TM0021	&,‡,†,*
ENSG00000163564;PYHIN1	ENSG00000160185;UBASH3A	pyrin and HIN domain family member 1	ubiquitin associated and SH3 domain containing A	NA	&,‡,†,*
ENSG00000231890;DARS-AS1	ENSG00000227191;TCRGC2	DARS antisense RNA 1	T Cell Receptor Gamma Constant 2	NA	&,‡,†,*
ENSG00000281881;SPRY4-IT1	ENSG00000109920;FNBP4	SPRY4 intronic transcript 1	formin binding protein 4	NA	&,‡,†,*
ENSG00000161405;IKZF3	ENSG00000160185;UBASH3A	IKAROS family zinc finger 3	ubiquitin associated and SH3 domain containing A	NA	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000143851;PTPN7	ubiquitin associated and SH3 domain containing A	protein tyrosine phosphatase, non-receptor type 7	NA	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000104814;MAP4K1	ubiquitin associated and SH3 domain containing A	mitogen-activated protein kinase kinase kinase kinase 1	NA	&,‡,†,*
ENSG00000160185;UBASH3A	ENSG00000005844;ITGAL	ubiquitin associated and SH3 domain containing A	integrin subunit alpha L	NA	&,‡,†,*
ENSG00000263970;RP11-789C17.5	ENSG00000054148;PHPT1	Antisense RNA	phosphohistidine phosphatase 1	NA	&,‡,†,*
ENSG00000272505;RP11-981G7.6	ENSG00000253641;LINCR-0001	lincRNA	uncharacterized LINCR-0001	NA	&,‡,†,*
ENSG00000234290;AC116366.6	ENSG00000197536;C5orf56	Antisense RNA	chromosome 5 open reading frame 56	NA	&,‡,†,*
ENSG00000237721;AF064858.3	ENSG00000235888;AF064858.1	lincRNA	lincRNA	NA	&,‡,†,*
ENSG00000231233;CCDC147-AS1	ENSG00000184277;TM2D3	CCDC147 antisense RNA 1	TM2 domain containing 3	NA	&,‡,†,*
ENSG00000251320;AC011352.3	ENSG00000248362;AC011352.1	lncRNA	lncRNA	NA	&,‡,†,*
ENSG00000218227;RPL19P9	ENSG00000204677;FAM153C	Ribosomal Protein L19 Pseudogene 9	family with sequence similarity 153 member C	NA	&,‡,†,*
ENSG00000237471;AC073115.2	ENSG00000229628;AC073115.7	lincRNA	lincRNA	NA	&,‡,†,*

& = Spearman Correlation > 0.80; % = Spearman Correlation < -0.80; ‡ = Padj KIRC Patient < 0.001; † = Padj Primary Tumor < 0.001; * = Padj VHL and PBRM1 Mutations < 0.001

References

1. Linehan WM, Walther MM, Zbar B. The genetic basis of cancer of the kidney. The Journal of urology **2003**;170:2163-72

2. Li Y, Kang K, Krahn JM, Croutwater N, Lee K, Umbach DM*, et al.* A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data. BMC Genomics **2017**;18:508

3. Weinstein JN, Collisson EA, Mills GB, Shaw KR, Ozenberger BA, Ellrott K*, et al.* The Cancer Genome Atlas Pan-Cancer analysis project. Nat Genet **2013**;45:1113-20

4. The Cancer Genome Atlas Research N. Comprehensive molecular characterization of clear cell renal cell carcinoma. Nature **2013**;499:43

5. Hsieh JJ, Purdue MP, Signoretti S, Swanton C, Albiges L, Schmidinger M*, et al.* Renal cell carcinoma. Nature reviews Disease primers **2017**;3:17009

6. Hsieh JJ, Le V, Cao D, Cheng EH, Creighton CJ. Genomic classifications of renal cell carcinoma: a critical step towards the future application of personalized kidney cancer care with pan-omics precision. The Journal of pathology **2018**;244:525-37

7. Hsieh JJ, Manley B, Khan N, Gao J, Carlo MI, Cheng EH. Overcome Tumor Heterogeneity-Imposed Therapeutic Barriers through Convergent Genomic Biomarker Discovery: A Braided Cancer River Model of Kidney Cancer. Seminars in cell & developmental biology **2017**;64:98-106

8. Guertin DA, Sabatini DM. Defining the role of mTOR in cancer. Cancer cell **2007**;12:9-22

9. Hsieh JJ, Cheng EH. A braided cancer river connects tumor heterogeneity and precision medicine. Clinical and Translational Medicine **2016**;5:42

10. Gibson SM, Ficklin SP, Isaacson S, Luo F, Feltus FA, Smith MC. Massive-scale gene co-expression network construction and robustness testing using random matrix theory. PLoS One **2013**;8:e55871

11. Luo F, Yang Y, Zhong J, Gao H, Khan L, Thompson DK*, et al.* Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. BMC Bioinformatics **2007**;8:299

12. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics **2008**;9:559

13. Ahn Y-Y, Bagrow JP, Lehmann S. Link communities reveal multiscale complexity in networks. Nature **2010**;466:761-4

14. Wolfe CJ, Kohane IS, Butte AJ. Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. BMC Bioinformatics **2005**;6:227

15. Srihari S, Ragan MA. Systematic tracking of dysregulated modules identifies novel genes in cancer. Bioinformatics **2013**;29:1553-61

16. Ficklin SP, Dunwoodie LJ, Poehlman WL, Watson C, Roche KE, Feltus FA. Discovering Condition-Specific Gene Co-Expression Patterns Using Gaussian Mixture Models: A Cancer Case Study. Sci Rep **2017**;7:8617

17. Dunwoodie LJ, Poehlman WL, Ficklin SP, Feltus FA. Discovery and validation of a glioblastoma co-expressed gene module. Oncotarget **2018**;9:10995-1008

18. van_ der_ Maaten LJPH, G. E. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research **2008**;9:2579–605

19. McInnes L, Healy J, Astels S. hdbscan: Hierarchical density based clustering. 2017.

20. Strehl A, Ghosh J. Cluster ensembles --- a knowledge reuse framework for combining multiple partitions. J Mach Learn Res **2003**;3:583-617

21. Kalinka AT, Tomancak P. linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. Bioinformatics **2011**;27:2011-2

22. Ricketts CJ, De Cubas AA, Fan H, Smith CC, Lang M, Reznik E*, et al.* The Cancer Genome Atlas Comprehensive Molecular Characterization of Renal Cell Carcinoma. Cell reports **2018**;23:313-26.e5

23. Yi JS, Cox MA, Zajac AJ. T-cell exhaustion: characteristics, causes and conversion. Immunology **2010**;129:474-81

24. Wang JC, Xu Y, Huang ZM, Lu XJ. T cell exhaustion in cancer: Mechanisms and clinical implications. Journal of cellular biochemistry **2018**;119:4279-86

25. Wherry EJ. T cell exhaustion. Nature Immunology **2011**;12:492

26. Chew GM, Fujita T, Webb GM, Burwitz BJ, Wu HL, Reed JS*, et al.* TIGIT Marks Exhausted T Cells, Correlates with Disease Progression, and Serves as a

Target for Immune Restoration in HIV and SIV Infection. PLoS Pathogens **2016**;12:e1005349

27. Andrews LP, Marciscano AE, Drake CG, Vignali DAA. LAG3 (CD223) as a Cancer Immunotherapy Target. Immunological reviews **2017**;276:80-96

28. Andrews LP, Marciscano AE, Drake CG, Vignali DA. LAG3 (CD223) as a cancer immunotherapy target. Immunological reviews **2017**;276:80-96

29. Nagy N, Cerboni C, Mattsson K, Maeda A, Gogolak P, Sumegi J*, et al.* SH2D1A and SLAM protein expression in human lymphocytes and derived cell lines. International journal of cancer **2000**;88:439-47

30. Eisenberg G, Engelstein R, Geiger-Maor A, Hajaj E, Merims S, Frankenburg S*, et al.* Soluble SLAMF6 Receptor Induces Strong CD8<sup>+</sup> T-cell Effector Function and Improves Anti-Melanoma Activity <em>In Vivo</em>. Cancer Immunology Research **2018**;6:127-38

31. Wu N, Zhong MC, Roncagalli R, Perez-Quintero LA, Guo H, Zhang Z*, et al.* A hematopoietic cell-driven mechanism involving SLAMF6 receptor, SAP adaptors and SHP-1 phosphatase regulates NK cell education. Nat Immunol **2016**;17:387-96

32. Jin L, Gong Y, Chen F, Zheng H, Li M, Xiong B. Transcription factor SLA2 regulated genes predict the survival of breast cancer patients. 2017. 2895-902 p.

33. Shi M, Cooper JC, Yu CL. A constitutively active Lck kinase promotes cell proliferation and resistance to apoptosis through signal transducer and activator of transcription 5b activation. Mol Cancer Res **2006**;4:39-45

34. Tsygankov AY. TULA-family proteins: Jacks of many trades and then some. Journal of cellular physiology **2018**

35. Rui X, Li Y, Jin F, Li F. TMPRSS3 is a novel poor prognostic factor for breast cancer. International journal of clinical and experimental pathology **2015**;8:5435-42

36. Diaz-Gallo LM, Sanchez E, Ortego-Centeno N, Sabio JM, Garcia-Hernandez FJ, de Ramon E*, et al.* Evidence of new risk genetic factor to systemic lupus erythematosus: the UBASH3A gene. PloS one **2013**;8:e60646

37. Chan CJ, Martinet L, Gilfillan S, Souza-Fonseca-Guimaraes F, Chow MT, Town L*, et al.* The receptors CD96 and CD226 oppose each other in the regulation of natural killer cell functions. Nat Immunol **2014**;15:431-8

38. Blake SJ, Dougall WC, Miles JJ, Teng MW, Smyth MJ. Molecular Pathways: Targeting CD96 and TIGIT for Cancer Immunotherapy. Clinical cancer research : an official journal of the American Association for Cancer Research **2016**;22:5183-8

39. Dougall WC, Kurtulus S, Smyth MJ, Anderson AC. TIGIT and CD96: new checkpoint receptor targets for cancer immunotherapy. Immunological reviews **2017**;276:112-20

40. Liu LL, Landskron J, Ask EH, Enqvist M, Sohlberg E, Traherne JA*, et al.* Critical Role of CD2 Co-stimulation in Adaptive Natural Killer Cell Responses Revealed in NKG2C-Deficient Humans. Cell Rep **2016**;15:1088-99

41. Hellstrom I, Ledbetter JA, Scholler N, Yang Y, Ye Z, Goodman G*, et al.* CD3-mediated activation of tumor-reactive lymphocytes from patients with advanced cancer. Proceedings of the National Academy of Sciences of the United States of America **2001**;98:6783-8

42. Tsai M-C, Spitale RC, Chang HY. Long intergenic non-coding RNAs – New links in cancer progression. Cancer research **2011**;71:3-7

43. Ching T, Peplowska K, Huang S, Zhu X, Shen Y, Molnar J*, et al.* Pan-Cancer Analyses Reveal Long Intergenic Non-Coding RNAs Relevant to Tumor Diagnosis, Subtyping and Prognosis. EBioMedicine **2016**;7:62-72

44. Ping Y, Liu C, Zhang Y. T-cell receptor-engineered T cells for cancer treatment: current status and future directions. Protein & Cell **2018**;9:254-66

CHAPTER THREE

IDENTIFYING TEMPORALLY REGULATED ROOT NODULATION BIOMARKERS IN *MEDICAGO TRUNCATULA* USING TIME SERIES GENE COEXPRESSION ANALYSIS

William L. Poehlman[1], Elise Schnabel[1], Suchitra Chavan[1], Julia Frugoli[1], and F. Alex Feltus[1]

[1]Clemson University Department of Genetics & Biochemistry, Clemson SC, USA.

To be submitted: *Plant Physiology*

Abstract

Root nodulation results from a symbiotic relationship between plant host and rhizobium. Synchronized gene expression patterns over the course of rhizobial infection result in activation of pathways that are unique from the highly conserved pathways that enable mycorrhizal symbiosis. To detect nodulation-specific biomarkers, we performed RNA sequencing of 30 *Medicago truncatula* root maturation zone samples at five distinct time points. These samples included plants inoculated with *Sinorhizobium meliloti* and control plants that did not receive any rhizobium. Following gene expression quantification, we identified 1,758 differentially expressed genes across all time points. We constructed a gene coexpression network (GCN) from this data and identified Link Community Modules (LCMs) that were comprised entirely of differentially expressed genes at specific time points. These LCMs included genes that were up-regulated at 24 hours following inoculation, suggesting an activation of allergen family genes and carbohydrate-binding gene products in response to rhizobium. We identified LCMs that were comprised entirely of genes that were down-regulated at 24 hours post-inoculation.

These modules suggest that down-regulating specific genes at 24 hours may result in decreased jasmonic acid production and an increase in cytokinin production. We also discovered LCMs that were composed entirely of genes that were down-regulated at 48 hours. These modules suggest that coordinated down-regulation of a specific set of genes involved in lipid biosynthesis may play a key role in nodulation. The modules identified in this manuscript provide a novel data mining resource for identifying polygenic biomarkers that are associated with root nodulation.

Introduction

Root nodulation is a symbiotic process in which a plant host allows rhizobium to colonize in meristematic root tissue called nodules. The plant provides carbon to the rhizobium in exchange for ammonium that is produced by atmospheric nitrogen fixation [1]. *Medicago truncatula* is a model plant organism that produces indeterminant nodules that persistently produce nodules from the meristem [2]. In response to inoculation with rhizobium such as *Sinorhizobium meliloti*, metabolic pathways are activated to achieve nodulation. Nod factor lipoproteins that are released by the rhizobium interact with receptor-like kinases in the plant, resulting in a spike in calcium oscillations in the nucleus of the cell that activates signaling pathways necessary to produce nodules [3]. These signaling pathways result in the production of proteins that allow the rhizobium to enter and colonize the host plant (infection thread formation), and nodule organogenesis results from rapid cortical cell division [4, 5]. The autoregulation of nodulation (AON) is activated by long-distance signaling pathways upon nodule inception. AON prevents the plant from producing too many nodules which would be harmful to the plant due to excess carbon consumption [6].

Root nodulation has evolved from another form of symbiosis called mycorrhiza, which is a conserved symbiosis with fungi across many plant species. However, conserved genes involved in mycorrhization have adapted unique functions to achieve root nodulation. For example, the LysM receptor-kinase gene has been duplicated in nodulating legumes, resulting in a copy that functions in root nodulation a copy that functions in mycorrhization [3]. As a result of such duplications, plants such as M. truncatula can achieve either mycorrhization or nodulation through activation of alternate signaling pathways. Temporally coordinated gene expression patterns are necessary to initiate and regulate root nodule formation. Transcriptome profiling has identified genes that are induced upon inoculation with rhizobium or nod factor. The NIN transcription factor is a master regulator of nodulation, playing roles in nodule organogenesis in cortical and epidermal root cells [7]. Other key genes that are induced upon rhizobial infection, termed nodulin genes, have been identified [8, 9]. While differential gene expression analysis of root transcriptomes has helped to identify such genes, analyzing the whole root tissue is likely diluting the signal of genes that are dynamically involved in nodule organogenesis. For example, CRE1, a cytokinin receptor, is expressed only in the root cortex and plays a key role in nodulation [7]. The root maturation zone is a region of the root that contains rapidly replicating cells [10]. In *M. truncatula*, this region of the root moves upwards through development and is a site of nodule formation [11, 12]. Analyzing the transcriptome of this portion of the root may reveal gene expression dynamics that were not detectable from whole-root tissue. Identifying packages of genes that are spatially and temporally regulated to induce nodulation remains a challenge.

Gene coexpression network (GCN) analysis is a method that can be applied to elucidate complex gene expression patterns over the time course of root nodulation. A GCN is a graph in which nodes represent genes and edges represent correlations between genes [13]. Typically, a Pearson or Spearman correlation is conducted across all available samples. Significant edges can then be extracted using techniques such as Random Matrix Theory (RMT) [14, 15], or a soft-threshold can be used to identify functional modules in techniques such as WGCNA [16]. Clustering techniques such as Link Communities can be used to identify clusters of genes in the GCN (modules) that are highly connected to each other, suggesting that they share common function or regulatory mechanism [17]. Knowledge Independent Network Construction (KINC) is a software package that constructs GCNs that contain condition-specific edges. Prior to performing correlation analysis on a given gene pair, KINC identifies sample clusters using Gaussian mixture models (GMMs) [18]. A correlation test is performed for each cluster separately, allowing significant GCN edges to be detected that are specific to only a subset of the input samples. These edges can then be annotated for attributes including genotype, phenotype, or experimental condition. Given that the minimum number of samples needed to conduct a correlation test is typically 20 to 30 samples, experiments with 30 samples or less typically can only identify edges that are not condition-specific. However, the GCN can still be used to identify sets of genes that demonstrate similar expression patterns. Thus, genes with similar temporal expression patterns can be identified.

The aim of this study was to identify polygenic root nodulation biomarkers that demonstrate time point-specific gene expression patterns. To achieve this aim, we performed RNAseq on 30 *M. truncatula* maturation zone samples across five distinct time points: 0 hours, 12 hours, 24 hours, 48 hours, and 72 hours. We identified differentially expressed genes between control and inoculated samples at each time point, and constructed a GCN from these samples. We identified LCM modules from this GCN and overlaid differentially expressed genes to identify modules that were differentially expressed at specific time points (Figure 3.1).

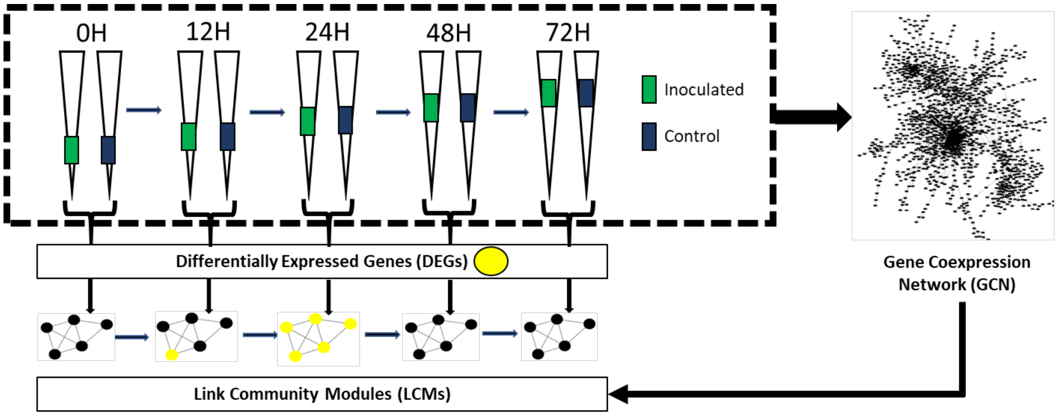

0H
12H
24H
48H
72H
Inoculated
Control
Differentially Expressed Genes (DEGs)
Link Community Modules (LCMs)
Gene Coexpression Network (GCN)

Figure 3.1 Experimental Overview. Differentially expressed genes between control and inoculated samples were identified at each time point. A GCN was constructed from all 30 samples, and LCMs were identified. Differentially expressed LCMs were identified by overlaying DEGs from each time point onto the LCMs

Results

RNAseq was performed on 30 maturation zone samples at five distinct time points: zero hours, 12 hours, 24 hours, 48 hours, and 72 hours post inoculation. At each time point, we analyzed three biological replicates of control samples that were not inoculated with rhizobium and three biological replicates of samples that were inoculated by rhizobium. We identified genes that were differentially expressed between control and inoculated samples at each time point, resulting in a total of 1,758 DEGs across all time points. While we detected 36 genes that were differentially expressed from twelve hours through seventy-two hours, the majority of the DEGs were unique to specific time points. We detected five unique DEGs at zero hours, 149 unique DEGs at 12 hours, 652 unique DEGs at 24 hours, 321 unique DEGs at 48 hours, and 317 unique DEGs at 72 hours (Figure 3.2). A heatmap of these DEGs demonstrates that samples can be clustered based on expression differences between control and inoculated samples (Figure 3.3). A normalized gene expression matrix (GEM) constructed from these thirty samples was used to construct a GCN with KINC. The resulting GCN contains 4,067 nodes and 7,854 edges, demonstrating scale-free topology ($R^2 = 0.799$). Figure 3.4 demonstrates a representative GCN edge from two genes that are down-regulated in inoculated samples at the 24 hour timepoint. We detected 161 LCMs that contained at least three genes, with the largest LCM containing 128 genes (Table S2). Figure 4.5 demonstrates a representative LCM that is composed of genes with the same expression patterns.

We detected 53 unique differentially expressed genes that were present in LCMs. Nine of the LCMs that we detected were comprised entirely of genes that were differentially expressed at specific time points. We detected modules that were up-regulated at 24 hours: M0004 and M0006 (Table 3.1). M0004 and M0006 are both enriched for PFAM terms PF01190 ("Pollen proteins Ole e I like") and PF09478 ("Carbohydrate binding domain CBM4") (Table S3). Conversely, we detected modules that were down-regulated at 24 hours: M0021, M0055, M0064, and M0072 (Table 3.2). M0021 is enriched for KEGG K13416 ("BAK1; brassinosteroid insensitive 1-associated receptor kinase 1 [EC:2.7.10.1 2.7.11.1]"). M0055 is enriched for PFAM PF06351 ("Allene oxide cyclase"). M0064 is enriched for GO:0008299 ("isoprenoid biosynthetic process"), GO:0004452 ("isopentenyl-diphosphate delta-isomerase activity"), K01823 ("idi, IDI; isopentenyl-diphosphate delta-isomerase [EC:5.3.3.2]"), K01597 ("MVD, mvaD; diphosphomevalonate decarboxylase [EC:4.1.1.33]"), K00787 ("FDPS; farnesyl diphosphate synthase [EC:2.5.1.1 2.5.1.10]"), PF00348 ("Polyprenyl synthetase"), and PF00288 ("GHMP kinases N terminal domain") (Table S3). We also detected modules that were down-regulated at 48 hours: M0032, M0118, and M0132 (Table 3.3). M0032 and M0132 are both enriched for K15401 ("CYP86A1; fatty acid omega-hydroxylase [EC:1.14.-.-]"). M0132 is also enriched for PF04535 ("Domain of unknown function (DUF588)") (Table S3.3).

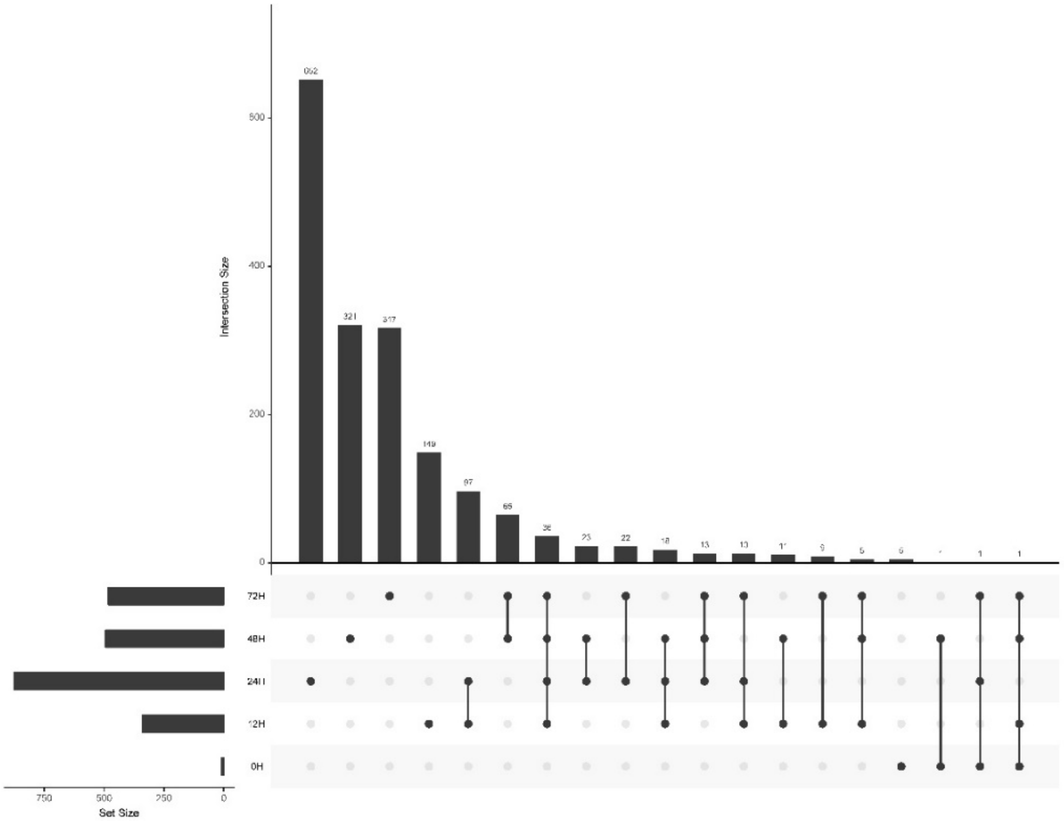


Figure 3.2 Upset plot of differentially expressed genes. A line connecting two dots indicates that a subset of genes was differentially expressed in both time points.

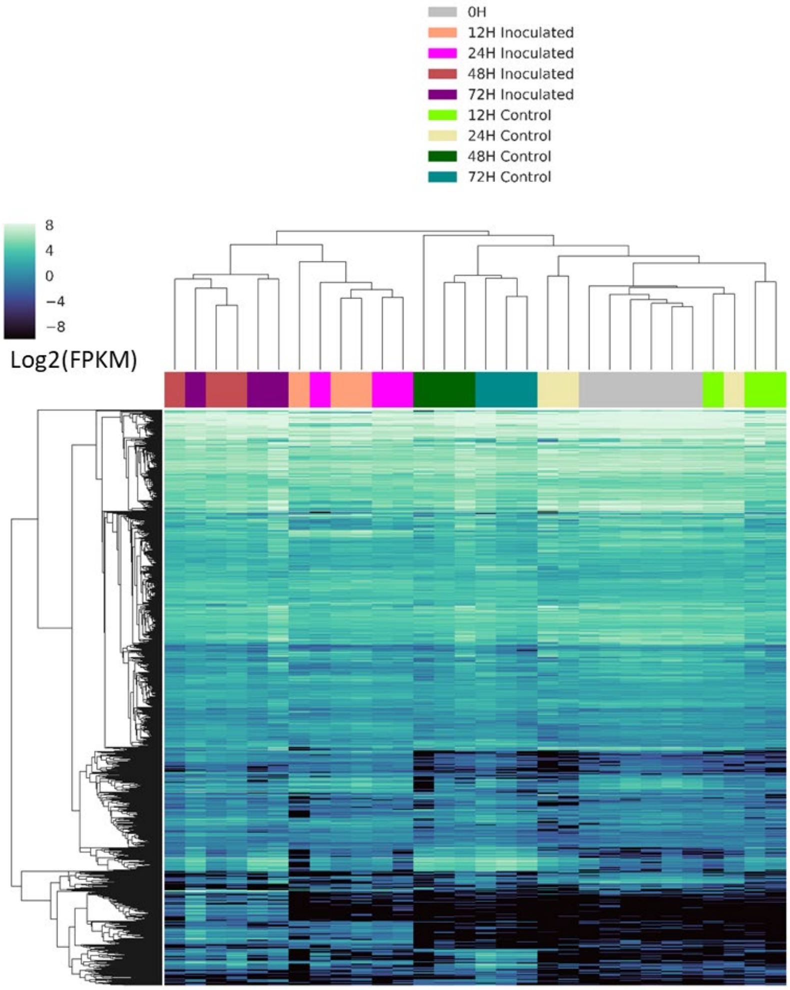


Figure 3.3 Overview of normalized expression levels of all DEGs. Samples were clustered and visualized using the seaborn clustergram function

Table 3.1. Up-Regulated GCN Modules at 24 Hours Post-Inoculation

Gene ID	Gene Description	LCM Module	LogFC	Padj
Medtr8g042900	pectinesterase/pectinesterase inhibitor	M0004	3.23	4.17E-07
Medtr7g102770	pollen Ole e I family allergens	M0004	2.82	1.23E-04
Medtr3g071470	pollen Ole e I family allergens	M0004	2.76	7.86E-04
Medtr4g074960	endo-1,4-beta-glucanase	M0004	2.62	5.99E-03
Medtr2g035120	disease-resistance response protein	M0004	1.73	8.41E-03
Medtr7g102770	pollen Ole e I family allergens	M0006	2.82	1.23E-04
Medtr3g071470	pollen Ole e I family allergens	M0006	2.76	7.86E-04
Medtr4g074960	endo-1,4-beta-glucanase	M0006	2.62	5.99E-03
Medtr4g109880	adenine nucleotide alpha hydrolase superfamily protein	M0006	1.98	1.33E-02

Table 3.2. Down-Regulated GCN Modules at 24 Hours Post-Inoculation

Gene ID	Gene Description	LCM Module	LogFC	Padj
Medtr3g070860	leucoanthocyanidin dioxygenase-like protein	M0021	-4.36	6.33E-05
Medtr2g008380	somatic embryogenesis receptor-like kinase	M0021	-2.70	8.74E-05
Medtr3g013890	3-oxo-delta(4,5)-steroid 5-beta-reductase-like protein	M0021	-2.28	4.85E-04
Medtr3g102730	3-oxo-delta(4,5)-steroid 5-beta-reductase-like protein	M0021	-2.00	1.60E-03
Medtr8g018570	seed linoleate 9S-lipoxygenase	M0055	-4.82	3.12E-08
Medtr3g070860	leucoanthocyanidin dioxygenase-like protein	M0055	-4.36	6.33E-05
Medtr7g417750	allene oxide cyclase	M0055	-2.63	1.97E-02
Medtr3g013890	3-oxo-delta(4,5)-steroid 5-beta-reductase-like protein	M0055	-2.28	4.85E-04
Medtr3g102730	3-oxo-delta(4,5)-steroid 5-beta-reductase-like protein	M0055	-2.00	1.60E-03
Medtr1g112230	mevalonate diphosphate decarboxylase	M0064	-2.49	4.97E-11
Medtr2g027300	geranylgeranyl pyrophosphate synthase	M0064	-1.71	2.50E-20
Medtr7g080060	isopentenyl-diphosphate delta-isomerase	M0064	-1.70	9.35E-04
Medtr8g018570	seed linoleate 9S-lipoxygenase	M0072	-4.82	3.12E-08
Medtr3g070860	leucoanthocyanidin dioxygenase-like protein	M0072	-4.36	6.33E-05
Medtr7g085120	Nod factor-binding lectin-nucleotide phosphohydrolase	M0072	-3.82	1.37E-06
Medtr7g417750	allene oxide cyclase	M0072	-2.63	1.97E-02

Table 3.3. Down-Regulated GCN Modules at 48 Hours Post-Inoculation

Gene ID	Gene Description	LCM Module	LogFC	Padj
Medtr5g014100	anionic peroxidase swpb3 protein	M0032	-3.28	3.71E-05
Medtr2g062600	Lipid transfer protein	M0032	-3.24	1.54E-06
Medtr8g089300	CASP POPTRDRAFT-like protein	M0032	-2.88	4.25E-03
Medtr5g070010	cytochrome P450 family-dependent fatty acid hydroxylase	M0032	-2.87	1.72E-05
Medtr5g064530	leguminosin group485 secreted peptide	M0118	-3.13	1.75E-06
Medtr0097s0070	CASP POPTRDRAFT-like protein	M0118	-3.04	4.96E-06
Medtr4g415290	glycerol-3-phosphate acyltransferase	M0118	-2.80	1.20E-05
Medtr1g071720	Lipid transfer protein	M0118	-2.46	4.90E-03
Medtr2g062600	Lipid transfer protein	M0132	-3.24	1.54E-06
Medtr5g064530	leguminosin group485 secreted peptide	M0132	-3.13	1.75E-06
Medtr2g009450	leguminosin group485 secreted peptide	M0132	-3.10	1.22E-05
Medtr0097s0070	CASP POPTRDRAFT-like protein	M0132	-3.04	4.96E-06
Medtr8g079050	GDSL-like lipase/acylhydrolase	M0132	-3.00	2.61E-08
Medtr8g089300	CASP POPTRDRAFT-like protein	M0132	-2.88	4.25E-03
Medtr5g070010	cytochrome P450 family-dependent fatty acid hydroxylase	M0132	-2.87	1.72E-05
Medtr4g415290	glycerol-3-phosphate acyltransferase	M0132	-2.80	1.20E-05
Medtr3g463060	cytochrome P450 family-dependent fatty acid hydroxylase	M0132	-2.68	2.61E-06

Methods

RNAseq Data Processing

The PBS-GEM workflow[https://github.com/wpoehlm/PBS-GEM] was utilized to process RNA sequencing reads on Clemson University's Palmetto Cluster. Poor quality sequences and adapters were removed using Trimmomatic-0.38 [19]. Cleaned reads were mapped to the Mt4.0v1 reference genome using hisat2-2.1.0 [20] with the following parameters: *hisat2 --rna-strandedness RF --min-intronlen 20 --maxintronlen 7000 -p 4 --downstream-transcriptome-assembly*. SAM alignment files were filtered to retain only unique primary alignments (MAPQ 60), sorted, and converted to BAM files using samtools-1.8[21]. Reference gene abundances were estimated using stringtie-1.3.4d [22, 23] with the following options: *stringtie –G –e –B –A.*

Differential Gene Expression Analysis

Raw gene counts were calculated using the prepDE.py script that is provided with the StringTie Package[https://ccb.jhu.edu/software/stringtie/dl/prepDE.py]. Differential expression analysis was performed using the DESeq2 R package [24], which internally normalizes for library size. Genes with total read counts of less than 50 were excluded from analysis. Control and inoculated samples were compared separately at each timepoint (0H, 12H, 24H, 48H, and 72H) using the *DESeqDataSetFromMatrix* function with the following formula: *design = ~ condition*. Genes with an adjusted p value of less than 0.05 were considered to be significant.

Gene Expression Matrix (GEM) Preparation

Gene-level FPKM (fragments per kilobase of gene per million read pairs) were extracted from the gene abundance output files produced by StringTie and merged into a gene expression matrix (GEM) using a perl script. The matrix was log2 transformed and preprocessed using the preprocessCore R library [25] to detect outliers and reduce technical noise. Pairwise Kolmogorov-Smirnov (KS) tests were performed to test for outlier samples (KS Dval > 0.15). No outlier samples were detected. The matrix was quantile normalized using the *normalize.quantiles* function. This normalized GEM was used to construct a gene coexpression network (GCN). Heatmaps and expression plots were generated using the *clustermap* and *tsplot* functions from the Seaborn Python package[https://seaborn.pydata.org/]

Coexpression Network Analysis

The OSG-KINC[https://github.com/feltus/OSG-KINC] [26] workflow was utilized to execute 10,000 KINC similarity jobs on the Open Science Grid with the following parameters: *kinc similarity--method pc --clustering mixmod --criterion ICL --min_obs 20*. Output was transferred to Clemson University's Palmetto Cluster and decompressed. KINC threshold was executed with the following parameters: *kinc threshold --min_csize 20 --clustering mixmod --method pc --th_method pc --max_modes 5*. A significance threshold of 0.946100 was identified, and the GCN was extracted using the following KINC extract parameters: *kinc extract --clustering mixmod --method pc --th_method pc --*

th 0.946100 --max_modes 5. Link Community Modules (LCM) were identified with the linkcomm R package [27], using the "single" hcmethod and a minimum cluster size of 3.

Discussion

We identified differentially expressed genes between control and inoculated samples at five distinct time points. As shown in Figure 3.2, the majority of these genes were unique to one specific time point. Thus, finding useful biological signal from hundreds of genes at each time point became a challenge. We used the GCN to identify genes that demonstrated similar expression over the time series. Figure 3.4 demonstrates how two genes with similar expression patterns over time produced a high correlation value. Even though the edge was not condition specific, we detected differential gene expression at the 24 hour time point. We then detected LCMs from this GCN to identify clusters of genes that all demonstrated similar expression patterns. As shown in Figure 3.5, LCM M055 is comprised entirely of genes that are down-regulated in inoculated samples at the 24 hour time point. Expression of these genes drops at the 12 hour timepoint and then is restored at the 24 hour time point in control samples while the expression in the inoculated samples slowly rises. We detected 161 LCMs that demonstrate coordinated expression patterns and overlaid differentially expressed genes at each time point to these LCMs. We were able to detect nine LCMs that were composed entirely of genes that were either up or down-regulated at a specific time point.

The two modules (M0004 and M0006) that are composed of up-regulated genes at 24 hours are enriched for PFAM term PF01190 ("Pollen proteins Ole e I like"). Pollen allergen genes have undergone a high degree of duplication and purifying selection, suggesting that they maintain significant biological function [28]. Chen et al. [28] characterized the function of allergen gene families in Arabidopsis and rice, demonstrating that allergen genes in Arabidopsis often have unique functions compared to rice. Some of these functions include defense response to bacterium and cell redox homeostasis, two processes that are involved in root nodulation. The genes in Table 3.1 are also enriched for PF09478 ("Carbohydrate binding domain CBM49"), a group of cellulases often associated with cell wall hydrolysis [29]. Notably, Table 3.2 contains a pectinesterase gene (Medtr8g042900) and a disease response gene (Medtr2g035120). We speculate that the up-regulated genes in Table 3.1 are involved in pathogen response or cell wall remodeling.

Table 3.2 contains GCN modules that are down-regulated in inoculated samples at 24 hours. M0072 and M0055 both contain a gene related to jasmonic acid synthesis: Medtr7g417750 (allene oxide cyclase). Suppression of this gene has been shown to reduce jasmonic acid (JA) levels in hairy roots of *M. truncatula*, lowering the plant's ability to achieve mycorrhization [30]. While JA seems to play a positive role in mycorrhization, it has been demonstrated to negatively impact root nodulation by inhibiting nod-factor induced calcium oscillations in the nucleus of the cells [31]. Interestingly, JA and cytokinin were found to have antagonistic roles in Arabidopsis xylems [32]. We speculate that down-regulation of genes in Table 3.2 results in a decrease in JA production and an increase in cytokinin biosynthesis, contributing to root nodulation by shutting down alternate pathways that would otherwise enable mycorrhizal symbiosis. We found Medtr7g085120, a Nod factor-binding lectin-nucleotide phosphohydrolase, to be down-regulated in inoculated samples at this time point. This protein was found to be necessary for rhizobial and mycorrhizal symbiosis in *Lotus japonicus*, a determinate nodulating plant [33]. Previous studies that analyzed RNA expression levels of whole-root tissue found this gene to be up-regulated early in the course of *S. meliloti* transfection in *M. truncatula*. We speculate that the cellular composition of the tissue used in our study demonstrates unique expression of this gene compared to the whole-root samples previously analyzed [8].

Table 3.3 contains two modules, M0032 and M0132, that are enriched for KEGG ontology term K15401 ("fatty acid omega-hydroxylase"). All three modules (M0032, M0132, and M0118) contain genes that are annotated as "lipid transfer protein". Lipids play diverse roles in plant physiology, such as signaling pathways involved in plant defense [34, 35]. Notably, Medtr4g415290 – a glycerol-3-phosphate acyltransferase (GPAT) gene, is down-regulated in both M0132 and M0118. GPAT enzymes catalyze the first step of membrane phospholipid biosynthesis [34, 36]. Another GPAT gene in *M. truncatula*, RAM2, was found to be necessary for fungal mycorrhization through its involvement in cutin biosynthesis [37]. Other genes involved in lipid biosynthesis are present in Table 3.3: Medtr5g070010 ("cytochrome P450 family-dependent fatty acid hydroxylase"), Medtr8g079050 ("GDSL-like lipase/acylhydrolase"), and Medtr3g463060 ("cytochrome P450 family-dependent fatty acid hydroxylase"). We hypothesize that down-regulation of genes in Table 3.3 helps to inhibit synthesis of specific fatty acids that would otherwise play a negative role in root nodulation. M0032 also contains a peroxidase protein, Medtr5g014100. Given that peroxidases are often involved in stimulating plant defense against pathogens, we hypothesize that down-regulation of this gene helps to enable rhizobial infection [38].

We hypothesize that many of the genes in Tables 3.1, 3.2, and 3.3 are involved in pathogen response. Given that the genes in Table 3.1 are up-regulated in inoculated samples, these genes might play a role in normal pathogen response, while the down-regulated genes in Tables 3.2 and 3.3 could play important roles in nodulation. As evidence, we compared these tables to genes that have been found to be dysregulated in NAD1 mutants. NAD1 (nodules with activated defense 1) is a gene that is necessary for maintaining rhizobial symbiosis in *M. truncatula* roots [39-41]. In NAD1 mutants, brown pigmentation accumulates in the nodules following the release of rhizobium from the infection thread, resulting in nodule necrosis. Wang et. al. performed transcriptome profiling of NAD1 mutants to compare with control plants at 21 days post inoculation [40]. Out of the six total genes in Table 3.1, three were up-regulated in NAD1 mutants (Medtr3g071470, Medtr4g109880, Medtr7g102770). Out of the 10 total genes in Table 3.2, five were up-regulated in the NAD1 mutants (Medtr8g018570, Medtr3g070860, Medtr7g417750, Medtr3g102730, Medtr3g013890), while one gene was down-regulated in the mutants (Medtr7g085120). Out of the 11 total genes in Table 3.3, six were up-regulated in the mutants (Medtr0097s0070, Medtr3g463060, Medtr5g070010, Medtr8g079050, Medtr4g415290, Medtr5g064530), while one gene was down-regulated

(Medtr5g014100). Given that NAD1 plays a key role in regulating immune response to rhizobium, genes that are up-regulated in NAD1 mutants may play key roles in nodulation [40]. Thus, we speculate that the down-regulation of genes in Table 3.2 and 3.3 help to suppress innate immune responses that would otherwise prevent rhizobial colonization in nodules.

The differentially expressed LCMs that we characterized provide novel polygenic biomarkers for root nodulation. Further research is needed to determine if the expression patterns of these genes are causative biomarkers, or if they are simply an effect of root nodulation or pathogen defense pathways. Regardless, these LCMs revealed biochemical differences between control and inoculated samples over the course of root infection. This study provides a novel list of differentially expressed genes from the maturation zone of *M. truncatula* roots. While we focused on the LCMs that were composed only of genes that were differentially expressed, other LCMs in which a subset of the genes were differentially expressed are worthy of further investigation. It is possible that genes that did not meet our significance cut off for differential expression are co-regulated with genes that did. To improve our resolution of gene expression patterns relevant to root nodulation, we will perform laser capture micro dissection to isolate specific cell types for gene expression quantification. This will amplify signals that were otherwise diluted by using a mixture of cell types from the maturation zone. This report describes a framework for identifying polygenic biomarkers that will be applied future experiments.

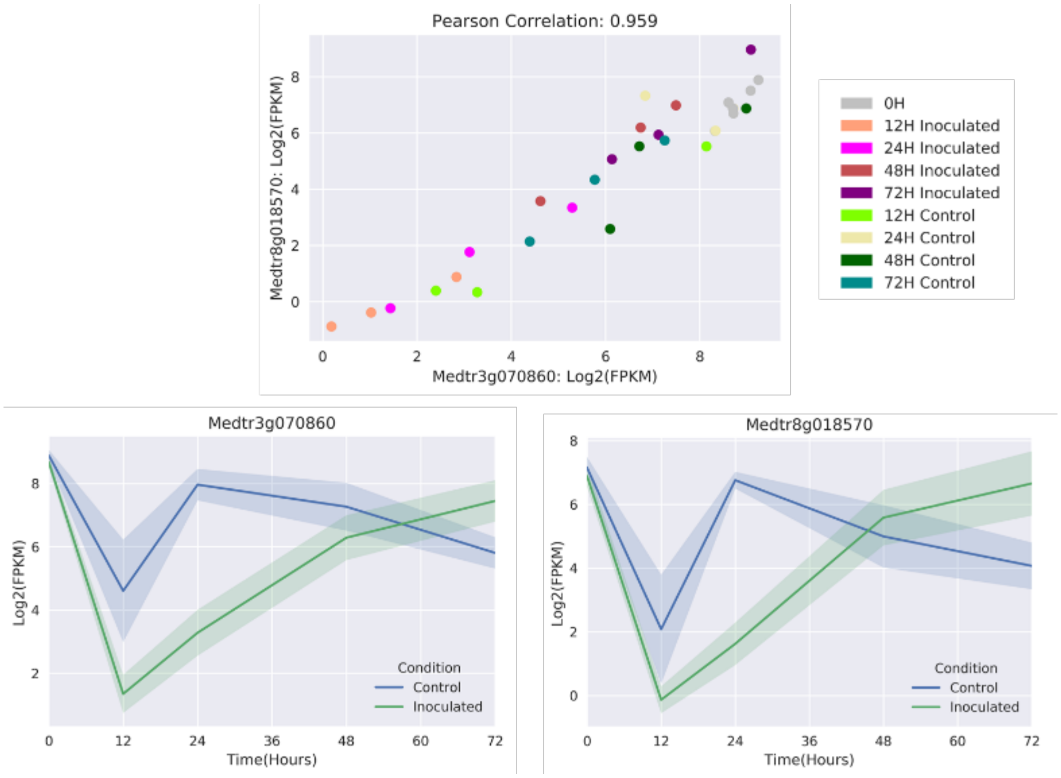


Figure 3.4 In a representative GCN edge, two genes produce a high correlation value across all samples. Expression plots reveal that both genes demonstrate differential expression at the 24 hour time point.

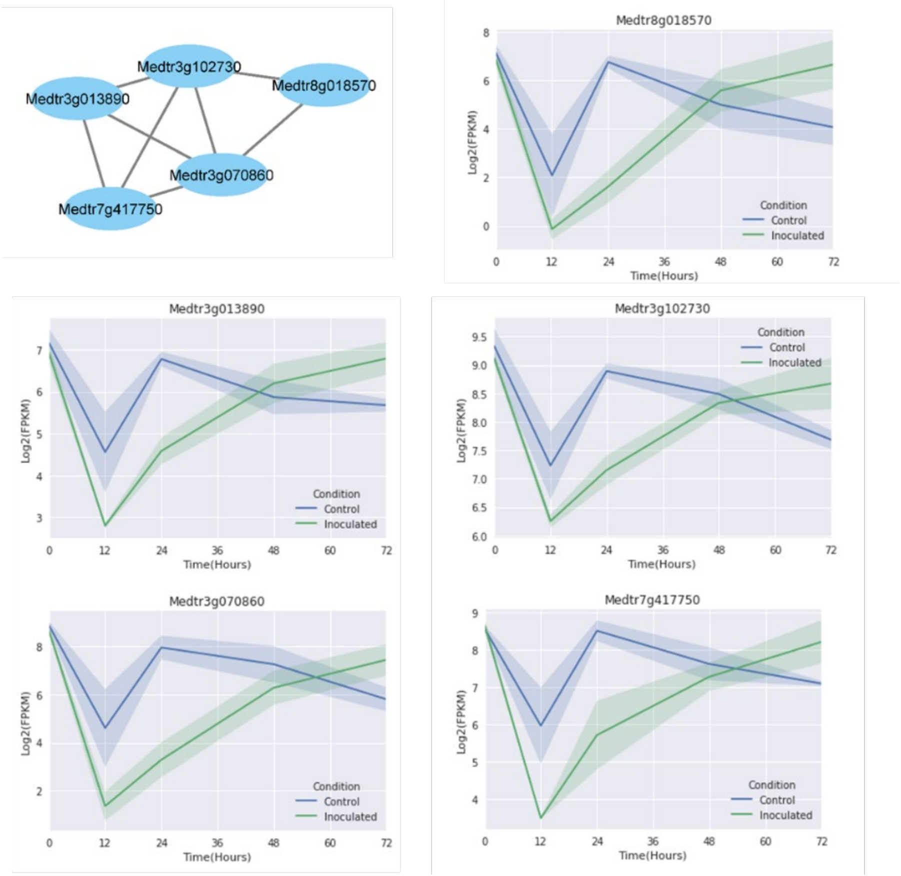


Figure 3.5 In a representative LCM, all genes demonstrate consistent expression patterns**.**

Acknowledgements

This work was funded by NSF PGRP Award # 1444461. The analysis presented in this manuscript was generated using Clemson University's Palmetto Cluster, and the Open Science Grid which is funded by the National Science Foundation and the U.S. Department of Energy's Office of Science.

References

1. Suzaki, T. and M. Kawaguchi, *Root nodulation: a developmental program involving cell fate conversion triggered by symbiotic bacterial infection.* Curr Opin Plant Biol, 2014. **21**: p. 16-22.
2. Gage, D.J., *Infection and Invasion of Roots by Symbiotic, Nitrogen-Fixing Rhizobia during Nodulation of Temperate Legumes.* Microbiology and Molecular Biology Reviews, 2004. **68**(2): p. 280-300.
3. Oldroyd, G.E., *Speak, friend, and enter: signalling systems that promote beneficial symbiotic associations in plants.* Nat Rev Microbiol, 2013. **11**(4): p. 252-63.
4. Jones, K.M., et al., *How rhizobial symbionts invade plants: the Sinorhizobium-Medicago model.* Nat Rev Microbiol, 2007. **5**(8): p. 619-33.
5. Long, S.R., *Genes and signals in the rhizobium-legume symbiosis.* Plant Physiol, 2001. **125**(1): p. 69-72.
6. Kawaguchi, T.S.a.M., *Systematic Regulation of Root Nodule Formation*, in *Advances in Biology and Ecology of Nitrogen Fixation*, P.T. Ohyama, Editor. 2014, InTech.
7. Vernie, T., et al., *The NIN Transcription Factor Coordinates Diverse Nodulation Programs in Different Tissues of the Medicago truncatula Root.* Plant Cell, 2015. **27**(12): p. 3410-24.

8. Larrainzar, E., et al., *Deep Sequencing of the Medicago truncatula Root Transcriptome Reveals a Massive and Early Interaction between Nodulation Factor and Ethylene Signals.* Plant Physiol, 2015. **169**(1): p. 233-65.

9. El Yahyaoui, F., et al., *Expression profiling in Medicago truncatula identifies more than 750 genes differentially expressed during nodulation, including many potential regulators of the symbiotic program.* Plant Physiol, 2004. **136**(2): p. 3159-76.

10. Hayashi, K., J. Hasegawa, and S. Matsunaga, *The boundary of the meristematic and elongation zones in roots: endoreduplication precedes rapid cell expansion.* Sci Rep, 2013. **3**: p. 2723.

11. Moreau, S., et al., *Transcription reprogramming during root nodule development in Medicago truncatula.* PLoS One, 2011. **6**(1): p. e16463.

12. Oldroyd, G.E. and J.A. Downie, *Coordinating nodule morphogenesis with rhizobial infection in legumes.* Annu Rev Plant Biol, 2008. **59**: p. 519-46.

13. Wolfe, C.J., I.S. Kohane, and A.J. Butte, *Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks.* BMC Bioinformatics, 2005. **6**: p. 227.

14. Luo, F., et al., *Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory.* BMC Bioinformatics, 2007. **8**: p. 299.

15. Gibson, S.M., et al., *Massive-scale gene co-expression network construction and robustness testing using random matrix theory.* PLoS One, 2013. **8**(2): p. e55871.

16. Langfelder, P. and S. Horvath, *WGCNA: an R package for weighted correlation network analysis.* BMC Bioinformatics, 2008. **9**: p. 559.

17. Ahn, Y.Y., J.P. Bagrow, and S. Lehmann, *Link communities reveal multiscale complexity in networks.* Nature, 2010. **466**(7307): p. 761-4.

18. Ficklin, S.P., et al., *Discovering Condition-Specific Gene Co-Expression Patterns Using Gaussian Mixture Models: A Cancer Case Study.* Sci Rep, 2017. **7**(1): p. 8617.

19. Bolger, A.M., M. Lohse, and B. Usadel, *Trimmomatic: a flexible trimmer for Illumina sequence data.* Bioinformatics, 2014. **30**(15): p. 2114-2120.

20. Kim, D., B. Langmead, and S.L. Salzberg, *HISAT: a fast spliced aligner with low memory requirements.* Nature methods, 2015. **12**(4): p. 357-360.

21. Li, H., et al., *The Sequence Alignment/Map format and SAMtools.* Bioinformatics, 2009. **25**(16): p. 2078-2079.

22. Pertea, M., et al., *StringTie enables improved reconstruction of a transcriptome from RNA-seq reads.* Nature biotechnology, 2015. **33**(3): p. 290-295.

23. Pertea, M., et al., *Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie, and Ballgown.* Nature protocols, 2016. **11**(9): p. 1650-1667.

24. Love, M.I., W. Huber, and S. Anders, *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.* Genome Biology, 2014. **15**(12): p. 550.

25. *preprocessCore: A collection of pre-processing functions*. 2018.

26. Poehlman, W.L., et al. *OSG-KINC: High-throughput gene co-expression network construction using the open science grid*. in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2017.

27. Kalinka, A.T. and P. Tomancak, *linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type.* Bioinformatics, 2011. **27**(14): p. 2011-2.

28. Chen, M., et al., *Origin and Functional Prediction of Pollen Allergens in Plants.* Plant Physiology, 2016. **172**(1): p. 341-357.

29. Urbanowicz, B.R., et al., *A tomato endo-beta-1,4-glucanase, SlCel9C1, represents a distinct subclass with a new family of carbohydrate binding modules (CBM49).* J Biol Chem, 2007. **282**(16): p. 12066-74.

30. Isayenkov, S., et al., *Suppression of allene oxide cyclase in hairy roots of Medicago truncatula reduces jasmonate levels and the degree of mycorrhization with Glomus intraradices.* Plant Physiol, 2005. **139**(3): p. 1401-10.

31. Sun, J., et al., *Crosstalk between jasmonic acid, ethylene and Nod factor signaling allows integration of diverse inputs for regulation of nodulation.* Plant J, 2006. **46**(6): p. 961-70.

32. Jang, G., et al., *Antagonistic interaction between jasmonic acid and cytokinin in xylem development.* Scientific Reports, 2017. **7**(1): p. 10212.

33. Roberts, N.J., et al., *Rhizobial and mycorrhizal symbioses in Lotus japonicus require lectin nucleotide phosphohydrolase, which acts upstream of calcium signaling.* Plant Physiol, 2013. **161**(1): p. 556-67.

34. Waschburger, E., et al., *Genome-wide analysis of the Glycerol-3-Phosphate Acyltransferase (GPAT) gene family reveals the evolution and diversification of plant GPATs.* Genet Mol Biol, 2018. **41**(1 suppl 1): p. 355-370.

35. Pinot, F. and F. Beisson, *Cytochrome P450 metabolizing fatty acids in plants: characterization and physiological roles.* Febs j, 2011. **278**(2): p. 195-205.

36. Takeuchi, K. and K. Reue, *Biochemistry, physiology, and genetics of GPAT, AGPAT, and lipin enzymes in triglyceride synthesis.* Am J Physiol Endocrinol Metab, 2009. **296**(6): p. E1195-209.

37. Wang, E., et al., *A common signaling process that promotes mycorrhizal and oomycete colonization of plants.* Curr Biol, 2012. **22**(23): p. 2242-6.

38. Almagro, L., et al., *Class III peroxidases in plant defence reactions.* J Exp Bot, 2009. **60**(2): p. 377-90.

39. Yu, H., et al., *Suppression of innate immunity mediated by the CDPK-Rboh complex is required for rhizobial colonization in Medicago truncatula nodules.* New Phytol, 2018. **220**(2): p. 425-434.

40. Wang, C., et al., *NODULES WITH ACTIVATED DEFENSE 1 is required for maintenance of rhizobial endosymbiosis in Medicago truncatula.* New Phytologist, 2016. **212**(1): p. 176-191.

41. Domonkos, A., et al., *NAD1 Controls Defense-Like Responses in Medicago truncatula Symbiotic Nitrogen Fixing Nodules Following Rhizobial Colonization in a BacA-Independent Manner.* Genes (Basel), 2017. **8**(12).

CHAPTER FOUR

OSG-GEM: GENE EXPRESSION MATRIX CONSTRUCTION USING THE OPEN SCIENCE GRID

William L. Poehlman[1], Mats Rynge[2], Chris Branton[3], D. Balamurugan[4], and Frank A. Feltus[1*]

[1]Clemson University, Department of Genetics & Biochemistry, SC 29634, USA; [2]University of Southern California, Information Sciences Institute, Marina Del Rey, CA 90292, USA; [3]Center for Computation & Technology, Louisiana State University, Baton Rouge, Louisiana 70803, USA; [4]University of Chicago, Computation Institute, Chicago, IL 60637, USA.

Published: *Bioinformatics and Biology Insights.* August 2nd 2016.
https://doi.org/10.4137/BBI.S38193

Abstract

High throughput DNA sequencing technology has revolutionized the study of gene expression while introducing significant computational challenges for biologists. These computational challenges include access to sufficient computer hardware and functional data processing workflows. Both of these challenges are addressed with our scalable, open source Pegasus workflow for processing high throughput DNA sequence datasets into a gene expression matrix (GEM) using computational resources available on the Open Science Grid (OSG). We detail usage of the workflow (OSG-GEM), discuss workflow design, inspect performance data, and assess accuracy in mapping paired-end sequencing reads to a reference genome. A target OSG-GEM user is proficient with the Linux command line and possesses basic bioinformatics experience. The user may run this workflow directly on the OSG or adapt it to novel computing environments.

Software Availability

OSG-GEM is open sourced under the GNU GPL License v2 and available at github.com/feltus/OSG-GEM.

Introduction

There is a molecular detection revolution underway in molecular biology. Biologists can now determine the dynamics of gene expression by sequencing and counting hundreds of millions of RNA and DNA molecules. This method is called next-generation sequencing (NGS) or high throughput sequencing (HTS) of DNA[1], which is steadily becoming a cost effective way to achieve diverse tasks including comparing DNA sequences of individuals for genetic analysis (genotyping by sequencing[2]); sequencing and counting RNA molecules after conversion to DNA to measure steady state RNA expression through the construction of a gene expression matrix (GEM) (RNAseq[3,4]); identifying organisms in environmental samples (metagenomics[5,6]); and many other applications. In essence, biologists can now "observe" molecular information flow from genomes that will have as much impact in understanding biological systems as the microscopy revolution of the 17th century.

There are several HTS platforms, including those from Illumina[7], Ion Torrent[8,9], and Pacific Biosciences[10], each with their own nuances. Each system creates a large quantity (often in the millions) of short DNA sequences (<200 base pairs called *reads*) that are encoded in chromosomal intervals (i.e. genes) with specific sequences that are unique to the species and individual. It would be ideal to capture the sequence of the entire DNA molecule without error, but high quality sequences are often obtain from one end of the molecule (single-end reads) or as pairs from both ends of the molecule (paired-end reads). It should be noted that Pacific Biosciences captures longer reads at the expense of a higher error rate. Thus, a key aspect of HTS DNA analysis involves aligning a large number of short DNA sequences to a smaller number of large reference genome DNA sequences that have been painstakingly discovered for many organisms. The HTS DNA data lifecycle and typical computational workflow are shown in Figure 4.1.

HTS DNA data files can be quite large and require complex computational workflows that extract a quantitative biological measurement. After sequencing is complete, a HTS DNA dataset is a concatenation of DNA sequence strings and metadata that include base pair call accuracy encoding (quality scores) as well as sample and instrument information. The datasets are stored in standard formats including FASTQ[11] and SRA[12]. Of note, SRA files can be manipulated and converted into FASTQ with the NCBI *sra-toolkit*[13]. Raw DNA reads often contain sequence contamination and poor quality reads, and must be cleaned before downstream processing. A Java application called *Trimmomatic*[14] performs this pre-processing task.

Once cleaned, reads are mapped to a reference genome[15] or transcriptome sequence set[16]. Several short-read genome aligners may be used for this, including bowtie2[17], bwa and variants[18-20], SOAP[21] and others, all of which create an alignment file, often in the SAM/BAM format[22]. The SAM/BAM file can be processed to extract sequence variants to the reference genome as well as count molecules that were sequenced at specific positions in the reference sequence. In the case of the RNAseq workflow, a gene expression matrix (GEM) can be constructed where each row is a known gene transcript and a column is a vector of gene expression intensities (i.e. RNA molecule count output detected for all genes in the sample). Molecule count information can be determined by the "tuxedo" suite of software that includes Tophat[23], Cufflinks[24], HISAT[25], and StringTie[26]. It should be noted that there is a plethora of other software that processes HTS reads, including GATK[27], Galaxy[28], and R/Bioconductor[29] to name but a few.

Processing HTS DNA datasets requires significant hardware resources. While it is possible to crunch these datasets on lab workstations, high-performance computing, high-throughput computing, and even big data systems may be required as the end user scales up the number of samples while datasets get richer and larger. One system that is highly scalable for HTS DNA workflow execution is the Open Science Grid (OSG[30]), a U.S. based consortium of over 100 universities and national laboratories set up to share distributed high throughput computing resources. A major stakeholder community of the OSG includes Large Hadron Collider physicists. As the OSG has matured, the benefits of the infrastructure have become apparent to experiments in other fields of science, including genomics, as well as universities to serve their local users' computational needs.

When the OSG resource contributors do not need their full capacity -- for example when an instrument is down for maintenance and no new data is produced -- the unused cycles on the compute resource can be shared back to the OSG community.

These opportunistic cycles add up to over 100 million core hours annually, and it is those cycles which are used for the OSG-GEM workflow. To access the OSG, our project utilized OSG Connect, which provides a simple but feature-rich interface to the OSG. Services used in this work include submit hosts, used to submit and manage jobs, and Stash, a multi-petabyte file storage service. Stash is a centralized storage system, but provides a number of access methods such as web, Globus[31], or other file transfer, and sharing tools such as distributed data caching close to the compute resources.
The OSG supports high throughput computing (HTC) via HTCondor[32]. HTCondor is a high throughput batch system for managing jobs on distributed resources. In a typical HTC workflow, several tasks are concurrently executed on independent machines that are connected through a network. Many scientific computations are suitable for HTC, including molecular screening, parameter sweeps, and statistical sampling. HTC systems have potential to accelerate GEM construction as a large quantity short sequences from HTS are processed. The GEM workflow developed for the OSG may be modified for transfer to any HTC systems, including a local campus cluster, grid or cloud.

The Pegasus Workflow Management System enables the execution of large-scale computational workflows on a variety of infrastructures[33]. Pegasus workflows are described as abstract directed acyclic graphs (DAG) which describe the tasks and data dependencies, but not the execution environment specifics. The reason for this abstract representation is that it provides portability for the workflow. The same workflow can be planned into an executable workflow for different resources at different times. This planning step, going from an abstract DAG to an executable workflow, is where Pegasus adds nodes to the graph such as data management nodes, and applies transformations to the graph, such as task clustering and workflow reduction based on already existing data products.

The OSG Gene Expression Matrix (OSG-GEM) workflow described in this article is a distributed computing mechanism to process RNAseq paired-end Illumina DNA sequence datasets into expression matrices using the tuxedo suite of software. We provide details on how the Pegasus-based workflow is organized, as well as usage and evaluation of OSG-GEM. OSG-GEM is adaptable to alternative methods of processing of HTS DNA datasets, as well as tuning or replacing the described software applications. OSG-GEM is freely available on GitHub.

Workflow Usage

OSG-GEM workflow overview. The OSG-GEM workflow is capable of processing hundreds to thousands of paired-end Illumina HTS DNA datasets in FASTQ format in parallel on OSG. Output is a two-column matrix of gene identifiers and normalized RNA expression intensities. These matrices can be stitched together to create larger GEMs for an organism, suitable for downstream analysis including gene co-expression matrix construction[34,35] (GCN in Figure 4.1) and differential gene expression profiling[36] (DEG in Figure 4.1). In order to execute the workflow, the user will need an account on the OSG[37], HTS DNA datasets in FASTQ format, and a reference genome or transcript assembly with associated gene annotations in GTF/GFF3 format. These files are either placed in a specific OSG-GEM directory or via paths defined in the *osg-*

gem.config file. The OSG-GEM workflow can be obtained from github[38] which contains the most up to date usage documentation. A test dataset is cloned with the workflow, which utilizes human chromosome 21 from the GRCh38 build of the human reference genome[39] along with a small dataset containing 200,000 human sequences (from SRR1825962[40]). The user can submit this reduced test dataset to become familiar with workflow setup and execution.

Pre-workflow steps. As shown in Figure 4.2, the first end user decision is to decide whether the Hisat2 or Tophat2 method will be used. We recommend Hisat2, since the developers are no longer supporting further development of Tophat2 (according to their website). We also recommend that the user become familiar with the application documentation for each method. If the Hisat2 method is chosen, the user must accumulate the reference genome sequence file in FASTA format[41] and gene location annotations in GTF format[42]. If the Tophat2 method is selected, the user must accumulate the reference genome sequence file in FASTA format and gene location annotations in GFF format[42]. Reference genome indices must be constructed using *hisat2-build*[43] or *bowtie2-build*[44]. In order to guide accurate mapping of sequencing reads independently from one another, annotated splice site information must be provided. For Hisat2, the built in *hisat2_extract_splice_sites.py* script generates a tab delimited list of splice junctions that allows the user to disable discovery of novel splice junctions[25]. Tophat2 can map reads directly to a reference transcriptome by generating index files of all sequences that are present in the reference genome annotation[23]. A reference genome annotation file in the GFF3 format is provided to guide RNA molecule counting using either StringTie[26] or Cufflinks[24].

OSG-GEM workflow setup. To setup an OSG-GEM workflow, the user must modify the *osg-gem.config* file to select software options and point to input data for recognition by Pegasus. First, the user must identify a reference prefix ($REF_PREFIX) that will be used to name all reference genome files used by the workflow. Next, the user must provide the file path to a forward FASTQ file and to a reverse FASTQ file. FASTQ filenames must end with .forward_1.fastq.gz or .forward_1.fastq to signify forward sequencing reads, and .reverse_2.fastq.gz or .reverse_2.fastq to signify reverse sequencing reads. Finally, the user must select 'True' or 'False' for each software option. Once the *osg-gem.config* file is appropriately modified, the user must place the necessary reference genome files in the *reference* directory of the workflow, with filenames containing the $REF_PREFIX that was specified in the *osg-gem.config* file.

If the user selects Hisat2 as 'True', the following files must be present in the *reference* directory:

$REF_PREFIX.fa,

$REF_PREFIX.1.ht2 … $REF_PREFIX.N.ht2,

$REF_PREFIX.Splice_Sites.txt,

$REF_PREFIX.gff3

If the user selects Tophat2 as 'True', the following files must be present in the *reference* directory:

$REF_PREFIX.fa,

$REF_PREFIX.1.bt2 … $REF_PREFIX.N.bt2,

$REF_PREFIX.rev.1.bt2

$REF_PREFIX.rev.2.bt2,

$REF_PREFIX.transcriptome_data.tar.gz,

$REF_PREFIX.gff3

For example, a user cloned OSG-GEM into ‘/stash2/user/username/GEM_test’, and placed input FASTQ files for dataset ‘TEST’ in ‘/stash2/user/username/Data’. To process this dataset using Hisat2 and StringTie with the GRCh38 build of the human reference genome, the *osg-gem.config* file would be modified as follows:

```
[reference]
reference_prefix = GRCh38
[inputs]
forward = /stash2/user/username/Data/TEST_1.fastq.gz
reverse = /stash2/user/username/Data/TEST_2.fastq.gz
[config]
tophat2 = False
hisat2 = True
cufflinks = False
```

stringtie = True

OSG-GEM Workflow Execution. Once the user submits the workflow by running the *submit* script, a list of all reference files recognized by Pegasus will print to the screen, as well as commands that can be used to monitor the workflow. If no reference files were found or multiple software options for alignment or quantification were selected, Pegasus will produce an error message.

The Pegasus workflow manager directs the execution of tasks in the workflow. In order to parallelize execution of read trimming and mapping while keeping hardware requirements low, the workflow splits input FASTQ files into files of 20,000 sequences on the OSG stash filesystem. To minimize filesystem I/O, input is read from disk and written only once by piping compressed input to *gunzip,* and piping the results to a python script that splits the files. To keep the number of files within each filesystem directory manageable, the hierarchical structure of the workflow is established at this step. Each sub-workflow manages the processing of 1,000 forward and 1,000 reverse FASTQ files.

An example input dataset contains 80 million sequences split into 1,000 chunks (20,000 sequences each) that will be managed by four DAG subworkflows (Figure 4.3). For each subworkflow, Pegasus creates a set of job submission scripts whose execution is managed by DAGMan and implemented by the HTCondor job submission system. A job consists of trimming (*Trimmomatic*) and mapping (*Hisat2* or *Tophat2*) sequences to the reference genome. After a job is completed, BAM-format alignment results are transferred back to a temporary OSG filesystem and then submitted back to an OSG compute node for an initial merge. Upon completion of all DAG subworkflows, a BAM file from each DAG subworkflow is transferred to an OSG compute node to generate the final *merged.bam* file. The final BAM file is then used to generate molecule counts which are represented as a column in a gene expression matrix (GEM).

Workflow Evaluation

Workflow Speed. Total OSG-GEM workflow runtime was compared with total runtime of an equivalent workflow processed on the Clemson University Palmetto Cluster (Figure 4.4). The first 5,000,000 sequences of dataset NCBI SRR1825962 were mapped against the GRCh38 build of the human reference genome. The corresponding comprehensive gene model annotation was downloaded[39] (Gencode Release 24) as GTF and GFF3 files. This dataset was processed using either a combination of Tophat2-Cufflinks or Hisat2-StringTie. The OSG-GEM workflows were submitted with requests of 6 GB of RAM and 30 GB of disk storage per job. An IBM DX340 machine with an allocation of 14 GB of RAM and 111 GB of available local_scratch node storage was requested for each job on the Palmetto Cluster. For OSG-GEM workflows, files were split into 20,000 sequence pieces as described previously, while the jobs on the Palmetto Cluster processed the dataset as complete FASTQ files. Total OSG-GEM walltime was documented using the *pegasus-statistics* command, and job walltime on the Palmetto Cluster was documented using the *qstat* command. The cumulative job walltime for each OSG-GEM subcomponent in an example workflow is shown in Figure 4.5.

Workflow Accuracy. The first 5,000,000 sequences of NCBI dataset SRR1825962 were processed as described above. To confirm the accuracy of the OSG-GEM workflow, gene expression values generated by each workflow were compared with results from the same tasks performed on the Palmetto Cluster without input file splitting (Figure 4.6). A tab delimited list of splice sites was provided to guide mapping of reads using Hisat2 with novel splice junction discovery disabled. Reads were mapped to the reference transcriptome directly using Tophat2, with novel splice junction and insertion-deletion discovery disabled. The Hisat2-StringTie OSG-GEM workflow produced identical results with the Palmetto Cluster, while the Tophat2-Cufflinks workflow resulted in a high correlation (Pearson's R = 0.99). These results indicate no loss of accuracy using the OSG-GEM workflow.

Discussion

We have described an open source OSG-GEM workflow to process HTS DNA datasets in the OSG distributed compute environment. The output of OSG-GEM, the gene expression matrix, is a focal data structure for multiple downstream analyses that could also be adapted to the OSG. Given the nature of the OSG, the workflow is highly scalable, adaptable, and available to a broad research community. OSG-GEM is in an active state of development, and we are continually working to synchronize OSG-GEM with new software applications and hardware resources available for OSG job submission.

This workflow serves as a valuable resource in a variety of situations. First, scientists without institutional access to high performance computing clusters may utilize the OSG to process RNASeq datasets without paying the cost of commercial cloud providers. Second, there is a significant development period to create and tune a complex workflow on the OSG or local computer. OSG-GEM is a solid baseline to use as-is or extend to other purposes. Third, as input dataset size continues to swell in size and quantity, hardware requirements will become more challenging, especially with competition for resource allocation on campus computing clusters. The ability to split large input datasets to process in parallel on the OSG will alleviate some of these issues by democratizing the resources available to analyze large datasets.

The goal of OSG-GEM is to construct accurate GEMs as quickly as possible for which there is potential for optimization. There is an impactful balance between resources requested, queue time, and job failure rate, all of which can potentially increase the performance of this workflow for a given dataset size. Resources can be balanced by requesting more RAM or more disk space that should result in fewer failed jobs, but could result in longer queue times. Job failure can be caused by requesting insufficient resources, or by problems on one or more nodes, such as exceeding local disk storage or hardware failure. In addition to failed jobs, we have found two "run-away" jobs that complete in an exceptionally long time, greatly influencing the final wall time (Figure 4.5 inset). If problematic nodes were avoided, OSG-GEM should complete in a fraction of the time shown in Figure 4.5.

As shown in Figure 4.4, the total workflow walltime of OSG-GEM workflows was greater than that of equivalent workflows processed on the campus Palmetto Cluster. Basic properties of the OSG make comparison to cluster resources difficult. We used the OSG via the OSG Connect system and thus had opportunistic access to the currently unused compute resources. A small percentage of our opportunistic jobs had to be restarted as a resource owner reclaims the resources for their own work. Such restarts might increase the overall walltime of the workflow. In addition, there is a large set of variables for the resource supply and demand equation on the OSG, including the number of available resources with varying system properties, the number of active users and what resources they require, and HTCondor user priorities. All of these variables change over time. However, it is only when doing performance tests that a user has to be concerned about these variables. For data processing, OSG users enjoy an automatic fair-share based work to resource matching.

Data access is also a factor when comparing execution on a campus resource versus the OSG. The campus resources usually have a local file system connected with a high speed, low latency network. The distributed nature of OSG means that jobs starting up on some remote resource will have to transfer or access data remotely over a wide area network. In the case of the OSG-GEM workflow, Pegasus handles these transfers transparently. Input data to a job is pulled in via parallel HTTP connections to the OSG Connect Stash filesystem, and potential output data is transferred back to Stash over SSH. These transfers do not show up in the runtime of the individual tasks, but can add up and affect the overall walltime.

In conclusion, the OSG-GEM workflow is a robust method for processing RNASeq datasets to generate gene expression matrices that serve as input for downstream applications. In the future, we intend to develop linked workflows that build upon the GEM datatype. OSG-GEM is functional and under active development. We are adapting OSG-GEM to evolving OSG infrastructure and tuning it to our needs, and we point the reader to examine the current build and documentation at github.com/feltus/OSG-GEM.

Figures

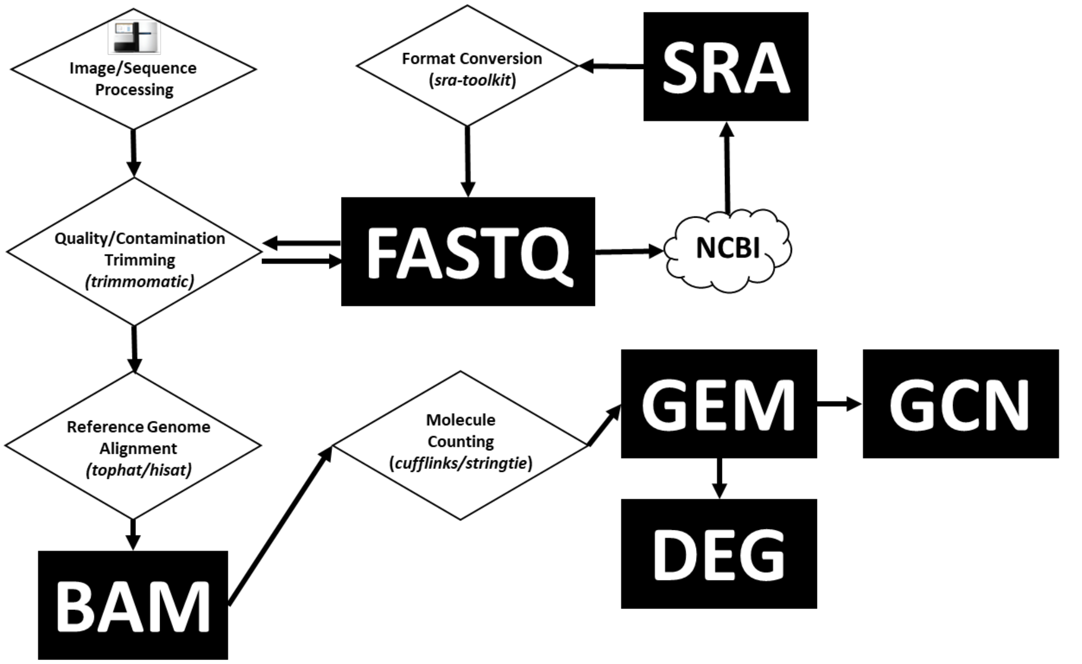

Image/Sequence Processing
Format Conversion (*sra-toolkit*)
SRA
Quality/Contamination Trimming *(trimmomatic)*
FASTQ
NCBI
Reference Genome Alignment *(tophat/hisat)*
Molecule Counting (*cufflinks/stringtie*)
GEM
GCN
DEG
BAM

Figure 4.1 DNA Sequence File Lifecycle. A DNA sequence starts its life as a TIFF image stack from a DNA sequencing instrument. Raw images are converted to a FASTQ text file and preprocessed or deposited into repositories such as the National Center for Biotechnology Information (NCBI) as Short Read Archive (SRA) files. Cleaned FASTQ files are mapped to a reference genome and converted to a BAM alignment file. BAM files can be mined for gene expression vectors that can be bundled into a gene expression matrix (GEM). GEMs are a stable data structure that can be mined for differentially expressed genes (DEGs) or used to construct Gene Co-expression Networks (GCNs) and processed by other workflows.

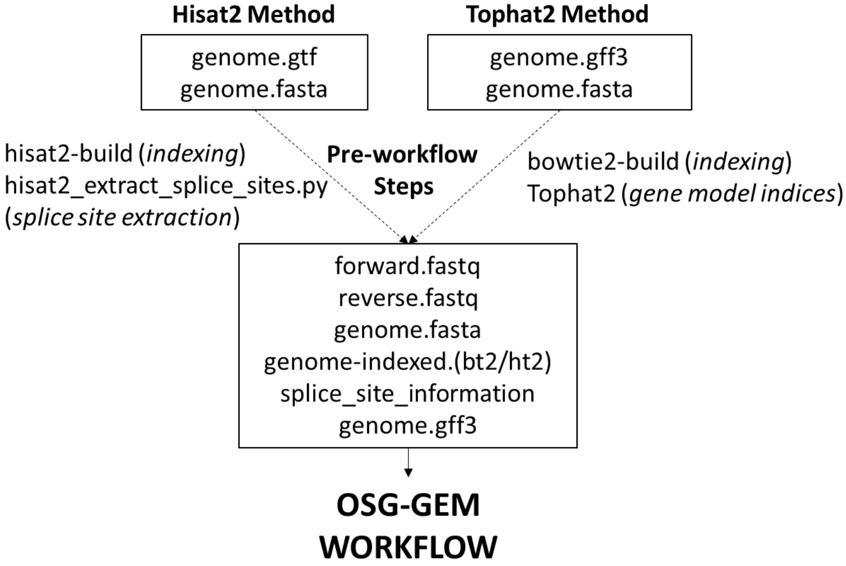


Figure 4.2 Preparation of Input Files for the Gene Expression Matrix Construction Workflow on the Open Science Grid (OSG-GEM). Required input files for either the Hisat2 or Tophat2 method are shown in boxes. The user provides paired-end DNA sequences in FASTQ format (forward/reverse) which can be extracted from SRA format files with the NCBI SRA Toolkit. The reference genome (genome) in FASTA format must be indexed using either the Hisat2 or Bowtie2 application. Built into the Hisat2 software package, the hisat2_extract_splice_sites.py script can generate a tab delimited list of splice sites using a reference annotation file in GTF format. Tophat2 can generate a set of gene model indices from GFF3 or GTF format files that contain splice site information in the form of a reference transcriptome. FASTQ file locations are defined in the osg-gem.config file and all other files are placed in the reference directory of the OSG-GEM workflow.

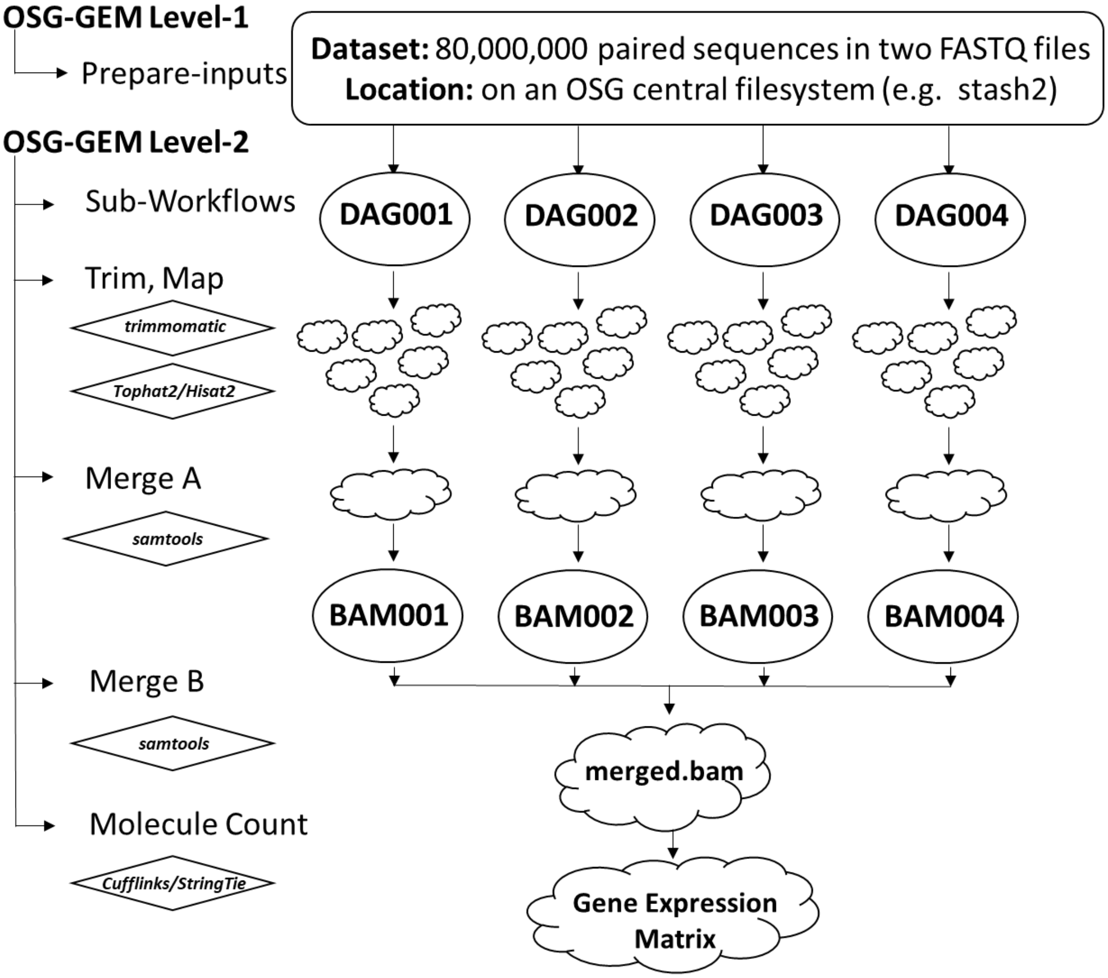


Figure 4.3 OSG-GEM Pegasus Workflow Diagram for a Representative HTS DNA Sequence Dataset. The workflow is managed by Pegasus and divided into two phases called levels 1 and 2. In level 1, input FASTQ files are split into an appropriate size for OSG compute nodes. In level 2, a specific quantity of split sequence files are managed by a finite number of DAGMan sub-workflows based on input file size. DAGMan manages the submission of jobs in the workflow, which results in trimming of FASTQ files, mapping to a reference sequence, merging alignment files, and quantifying RNA expression levels. Upon completion of all DAG subworkflows a final merged.bam file

is created. The final BAM file is used to count molecules for parsing into a gene expression matrix (GEM).

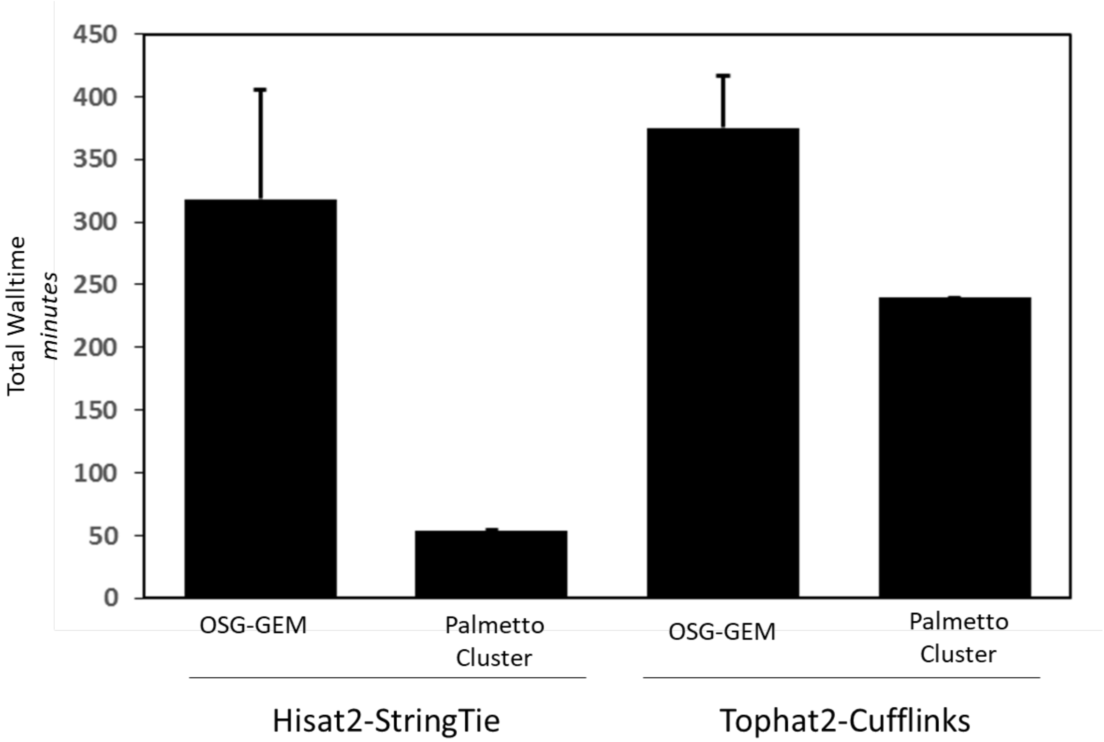


Figure 4.4 Walltime Comparison Between the OSG and Palmetto Cluster. Total workflow walltime of OSG-GEM workflows was compared with total walltime of equivalent workflows processed as single jobs on Clemson University's Palmetto Cluster. A representative dataset containing 5,000,000 paired-end sequencing reads was mapped to the human reference genome followed by RNA molecule quantification using a combination of Hisat2-StringTie or Tophat2-Cufflinks. Error bars represent standard error of the mean (n=3).

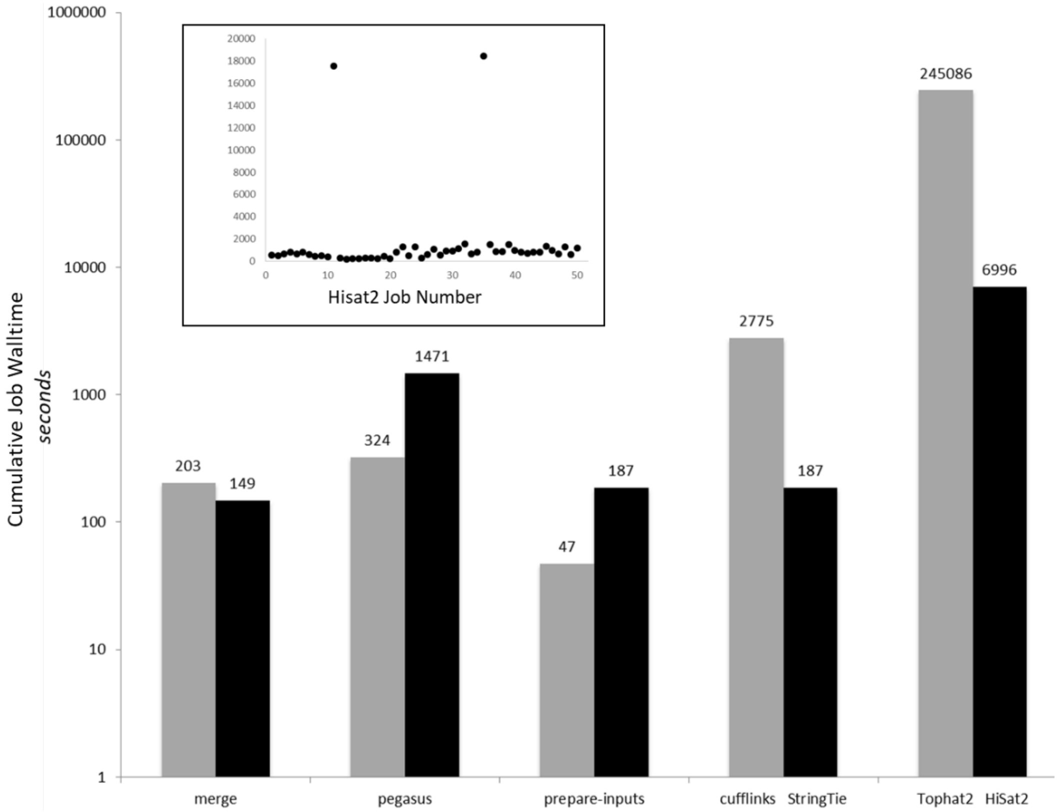


Figure 4.5 OSG-GEM Component Performance. A 5,000,000 sequence dataset was processed using the Hisat2-StringTie and Tophat2-Cufflinks methods of OSG-GEM. The cumulative walltime for each step of the workflow is shown for TopHat2-Cufflinks (gray bars) and Hisat2-StringTie (black bars). The inset scatterplot presents the walltime of each Hisat2 job in the Hisat2-StringTie workflow.

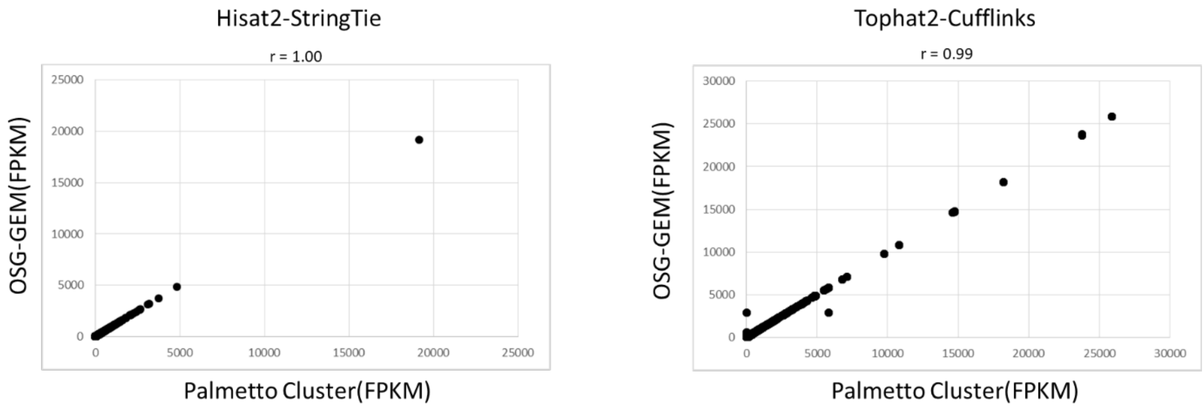


Figure 4.6 GEM Accuracy After Pre-processing. Gene expression vectors generated by processing a 5,000,000 sequence dataset using either the Hisat2-StringTie or the Tophat2-Cufflinks method on the Open Science Grid (OSG-GEM workflow) and single jobs on the Palmetto Cluster were compared. FPKM = Fragments Per Kilobase of Exon per Million Mapped Reads. Pearson correlation coefficients were calculated for each comparison.

Acknowledgements

This research was supported by the U.S. National Science Foundation Award #1447771 entitled “PXFS: ParalleX Based Transformative I/O System for Big Data.” This research was done using resources provided by the Open Science Grid, which is supported by the National Science Foundation and the U.S. Department of Energy's Office of Science. Clemson University is acknowledged for generous allotment of compute time on the Palmetto Cluster.

References

1. Altman RB, Prabhu S, Sidow A, et al. A research roadmap for next-generation sequencing informatics. *Science translational medicine.* 2016;8(335):335ps310.

2. Elshire RJ, Glaubitz JC, Sun Q, et al. A Robust, Simple Genotyping-by-Sequencing (GBS) Approach for High Diversity Species. *Plos One.* 2011;6(5):e19379.

3. Dillies MA, Rau A, Aubert J, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief Bioinform.* 2013;14(6):671-683.

4. Conesa A, Madrigal P, Tarazona S, et al. A survey of best practices for RNA-seq data analysis. *Genome Biol.* 2016;17(1):13.

5. Warnecke F, Luginbuhl P, Ivanova N, et al. Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite. *Nature.* 2007;450(7169):560-565.

6. Wooley JC, Godzik A, Friedberg I. A primer on metagenomics. *PLoS Comput Biol.* 2010;6(2):e1000667.

7. Illumina. 2016; http://www.illumina.com/.

8. Fisher T. 2016; https://www.thermofisher.com/us/en/home/brands/ion-torrent.html.

9. Yeo ZX, Wong JC, Rozen SG, Lee AS. Evaluation and optimisation of indel detection workflows for ion torrent sequencing of the BRCA1 and BRCA2 genes. *BMC Genomics.* 2014;15:516.

10. Pacific Biosciences. 2016; http://www.pacb.com/.

11. Wikipedia. FASTQ Format. 2016; https://en.wikipedia.org/wiki/FASTQ_format.

12. Kodama Y, Shumway M, Leinonen R. The Sequence Read Archive: explosive growth of sequencing data. *Nucleic Acids Res.* 2012;40(Database issue):D54-56.

13. sra-tools. 2016; https://github.com/ncbi/sra-tools.

14. Trimmomatic. www.usadellab.org/cms/?page=trimmomatic. 2013; http://www.usadellab.org/cms/index.php?page=trimmomatic.

15. E pluribus unum. *Nature Methods.* 2010;7:331.

16. Haas BJ, Papanicolaou A, Yassour M, et al. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nat Protoc.* 2013;8(8):1494-1512.

17. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nat Methods.* 2012;9(4):357-359.

18. Abuin JM, Pichel JC, Pena TF, Amigo J. SparkBWA: Speeding Up the Alignment of High-Throughput DNA Sequencing Data. *PLoS One.* 2016;11(5):e0155461.

19. Jo H, Koh G. Faster single-end alignment generation utilizing multi-thread for BWA. *Bio-medical materials and engineering.* 2015;26 Suppl 1:S1791-1796.

20. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics.* 2009;25(14):1754-1760.

21. Li R, Li Y, Kristiansen K, Wang J. SOAP: short oligonucleotide alignment program. *Bioinformatics.* 2008;24(5):713-714.

22. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics.* 2009;25(16):2078-2079.

23. Trapnell C, Pachter L, Salzberg SL. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics.* 2009;25(9):1105-1111.

24. Trapnell C, Williams BA, Pertea G, et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology.* 2010;28(5):511-515.

25. Kim D, Langmead B, Salzberg SL. HISAT: a fast spliced aligner with low memory requirements. *Nat Methods.* 2015;12(4):357-360.

26. Pertea M, Pertea GM, Antonescu CM, Chang TC, Mendell JT, Salzberg SL. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat Biotechnol.* 2015;33(3):290-295.

27. McKenna A, Hanna M, Banks E, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* 2010;20(9):1297-1303.

28. Hillman-Jackson J, Clements D, Blankenberg D, Taylor J, Nekrutenko A. Using Galaxy to perform large-scale interactive data analyses. *Curr Protoc Bioinformatics.* 2012;Chapter 10:Unit10 15.

29. Gentleman RC, Carey VJ, Bates DM, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.* 2004;5(10):R80.

30. Pordes R, Petravick D, Kramer B, et al. The open science grid. *Journal of Physics: Conference Series.* 2007;78(1):012057.

31. Foster I. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. *IEEE Internet Computing.* 2011;15(03):70-73.

32. Thain D, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the Condor experience. *Concurrency and computation: practice and experience.* 2005;17(2-4):323-356.

33. Deelman E, Vahi K, Juve G, et al. Pegasus: a Workflow Management System for Science Automation. *Future Generation Computer Systems.* 2015;46:17-35.

34. Feltus FA, Ficklin SP, Gibson SM, Smith MC. Maximizing capture of gene co-expression relationships through pre-clustering of input expression samples: an Arabidopsis case study. *BMC Syst Biol.* 2013;7:44.

35. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics.* 2008;9:559.

36. Vu TN, Wills QF, Kalari KR, et al. Beta-Poisson model for single-cell RNA-seq data analyses. *Bioinformatics.* 2016.

37. The Open Science Grid. 2016; https://www.opensciencegrid.org/.

38. Poehlman W. OSG-GEM. 2016; https://github.com/feltus/OSG-GEM.

39. GENCODE. 2016; http://www.gencodegenes.org.

40. Blakeley P, Fogarty NM, del Valle I, et al. Defining the three cell lineages of the human blastocyst by single-cell RNA-seq. *Development (Cambridge, England).* 2015;142(18):3151-3165.

41. Wikipedia. FASTA Format. 2016; https://en.wikipedia.org/wiki/FASTA_format.

42. ENSEMBL. GTF/GFF Format. 2016; http://useast.ensembl.org/info/website/upload/gff.html.

43. HISAT2_BUILD. 2016; https://ccb.jhu.edu/software/hisat2/manual.shtml#the-hisat2-build-indexer.

44. BOWTIE2-BUILD. 2016; http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#the-bowtie2-build-indexer.

CHAPTER FIVE

OSG-KINC: HIGH-THROUGHPUT GENE CO-EXPRESSION NETWORK CONSTRUCTION USING THE OPEN SCIENCE GRID

William L. Poehlman[*], Mats Rynge[‡], D. Balamurugan[§], Nicholas Mills[†], and Frank A. Feltus[*]
[*]Department of Genetics and Biochemistry,
Clemson University, Clemson, SC 29634
[†]Holcombe Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
[‡]Information Sciences Institute,
University of Southern California, Marina Del Rey, CA 90292
[§]Computation Institute,
University of Chicago, Chicago, IL 60637

Published: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Date of Conference**:** 13-16 Nov. 2017. Date Added to IEEE ***Xplore*:** 18 December 2017. Electronic ISBN**:** 978-1-5090-3050-7. DOI: 10.1109/BIBM.2017.8217938

Abstract

Gene Co-expression Network (GCN) analysis is a method to characterize the complexity underlying biological systems. With an increasing availability of datasets available for mining complex gene expression patterns, novel algorithms and computational frameworks must be developed to take advantage of the wealth of information. OSG-KINC is a Pegasus workflow that enables highly parallel execution of KINC –

Knowledge Independent Network Construction – using resources available on the Open Science Grid (OSG). A yeast GCN was constructed using the OSG-KINC workflow, providing an example GCN resource for biological hypothesis testing. Timing experiments demonstrate that the number of jobs submitted by the user significantly affects the performance of the workflow. An overview of workflow usage, bottlenecks, and efforts for improvement is provided. OSG-KINC is freely available at https://github.com/feltus/OSG-KINC under GNU General Public License version 3.

Introduction

High-throughput DNA sequencing technology enables high- resolution quantification of gene expression by counting RNA molecules. Thus, RNA sequencing (RNAseq) has become a common technique for biological hypothesis testing [34], [49]. RNAseq datasets are text files where each byte encodes a DNA base pair (A, T, G, C) underlying the source of the RNA transcript, associated probability that each base pair call is correct (quality score), or metadata on the experiment. Since each experiment produces information for hundreds of millions of base pairs for tens to thousands of samples, processing raw RNAseq datasets requires significant hardware resources.

A variety of platforms and scientific workflows have been developed to enable researchers to process RNAseq data [22], [33], [36]. However, the fundamental output of RNAseq analysis, normalized gene expression values, remains a stable data source that may be mined for biological information. Normalized gene expression vectors for all samples can be merged into a Gene Expression Matrix (GEM) for downstream analysis.

For example, systems genetics approaches to understanding the basis of complex traits involve interpreting multiple data types including transcriptomes in GEMs, metabolomes, genomes, and various forms of phenotypic data [12]. Understanding these complex properties of biological systems are quite promising but the computation remains a challenge [8], [38].

One method to address the complexity of biological processes is through gene co-expression network (GCN) analysis. A GCN is constructed from a GEM and is represented as a graph in which nodes are genes or RNA transcripts and edges that connect nodes represent gene co-expression. Correlation analysis is performed, typically using Pearson or Spearman statistics, on a pairwise basis across all combination of gene output quantified in the input GEM [17], [46]. A natural GCN exhibits scale-free behavior, and highly interconnected nodes in the graph — modules — can be parsed and characterized. Insight on the dynamics of complex gene expression patterns may be gained from these modules, and the function of genes may be characterized through guilt-by-association inferences [7], [48]. A variety of tools for constructing a GCN are available, including WGCNA [28], RMTGeneNet [21], and petal [35]. Typically, correlation analysis is performed across all available samples.

Knowledge Independent Network Construction (KINC) is a software package that builds GCNs from mixed-condition input GEM datasets [3]. In contrast to GCN construction tools that perform correlation analysis across all available samples, KINC uses Gaussian Mixture Models (GMMs) to identify clusters of input samples based on pairwise gene expression patterns [19]. Correlation analysis is then performed for each cluster, allowing for edges in the resulting GCN to be annotated based on the type of samples that are present in the identified clusters. By identifying distinct modes in the input data prior to performing correlation analysis, condition-specific gene expression patterns may be identified.

To build a GCN with KINC software, three steps must be executed: KINC similarity, KINC threshold, and KINC extract. KINC similarity performs GMM clustering and correlation analysis across all pairwise gene combinations. KINC threshold identifies a significance threshold using Random Matrix Theory (RMT) thresholding [30]. KINC extract uses the threshold identified by RMT to extract significant correlations. While KINC threshold and KINC extract have low computational requirements, KINC similarity using GMMs requires thousands to millions of CPU hours to complete. This is due to the fact that a typical eukaryotic reference genome will require billions of pairwise comparisons. For each comparison, GMM parameters are estimated by iterative execution of the Expectation-Maximization (EM) algorithm [16], which is a computationally challenging task [23]. Following identification of GMM clusters, Spearman or Pearson correlation is calculated within each cluster separately. The number of comparisons performed by KINC similarity is equal to (n(n-1))/2 where n represents the number of rows in the input matrix.

For example, if an input GEM has measurements for 50,000 genes, KINC will perform 1,249,975,000 comparisons. This is the minimum number of correlation tests that will be performed, with multiple correlations being calculated for genes that demonstrate multiple GMM modes of expression. While GCN software that does not perform clustering prior to correlation analysis can typically be run on a single CPU, the use of GMMs by KINC is a challenging task, requiring computation to be split into small pieces and run in parallel. The KINC software requires a parameter that specifies the number of computing jobs that will be performed, as well as a job index number that corresponds to a given subset of the input matrix. Each pairwise comparison of gene expression can be performed independently of one another. Thus, execution of KINC similarity can be easily parallelized when multiple CPUs are available.

OSG-KINC is a Pegasus [15] workflow that is configured to run on the Open Science Grid (OSG) [37]. The OSG provides opportunistic access to unused compute cycles from data centers across the United States. Computation that can be split into small pieces, requiring small amounts of memory and disk space per job, is well suited for the OSG. Due to the large number of jobs that can be submitted, the heterogeneity of the compute resources available, and the ability of resource-owners to reclaim compute cycles, job failure is expected and must be carefully monitored. The Pegasus Workflow Management system addresses these challenges by monitoring workflow progress and job completion. Failed jobs are automatically detected and resubmitted, using DAGMan as the meta-scheduler to HTCondor [44]. By default, the workflow uses 1 GB of RAM per job. In the event of a job failure, a failed-job-callout script is invoked that modifies the submit script for the corresponding job to retry the job with 5 GB of RAM.

Pegasus workflows are designed to portable between execution environments [14]. At the time of workflow submission, a dax-generator script generates an XML file that represents the necessary workflow tasks. The pegasus-plan command is then executed to link computational tasks to compute and staging sites using a provided sites.xml catalogue. Thus, a user must modify the sites catalogue and the arguments that are passed into pegasus-plan, rather than the design of the workflow, to enable execution in a different environment.

The OSG-KINC workflow is freely available on Github [5]. The workflow contains a pre-compiled KINC Linux binary that is executable on the OSG. This binary is transferred to the compute sites during workflow execution and dynamic library dependencies are pulled from CVMFS [1], [42], a read-only, heavily cached, distributed filesystem that hosts software modules available on the OSG. To run the workflow with a new input dataset, the user must place a tab-delimited GEM in the task-files directory of the workflow. The OSG stash filesystem is used to stage output files during workflow execution, and output will be transferred to the user's /local- scratch directory upon completion of all KINC jobs. OSG-KINC will automatically identify the input matrix dimensions to pass as arguments into the KINC compute jobs. The user must specify how many pieces the computation will be split into at time of submission. Full instructions for proper input matrix format and workflow submission can be found in the README.md file on https://github.com/feltus/OSG-KINC.

Input into the OSG-KINC workflow may be generated using the OSG-GEM workflow [4]. This workflow processes raw Illumina RNAseq datsets into a GEM containing gene expression intensities across all samples and all annotated RNA transcripts in the genome [36]. Once the OSG-KINC workflow has run, the user must transfer output files to another system to perform KINC threshold and KINC extract

(Figure 5.1). This is due to the nature of the KINC threshold jobs: a large number of files must be iterated over using a single CPU, and it may take days to weeks to identify the threshold. In addition, the OSG-KINC workflow will generate terabytes of output data for a typical experiment. Transferring a large amount of data to a compute node or having a long-running local job is not well-suited for the OSG. Thus, OSG-KINC is designed to only perform the computation that performs well with a high degree of parallelism on distributed compute resources. An overview of a possible workflow to generate a GCN using raw data hosted by the National Center for Biotechnological Information (NCBI) is provided in Figure 5.1. Globus [20] may be used to transfer the large volume of output files from KINC similarity to a cloud or HPC resource.

Results

Use Case: Yeast GCN. *Saccharomyces cervisiae* is a species of yeast that serves as a model organism for genetic studies, and plays important roles in industrial processes including carbohydrate fermentation [11], [40], [43]. While the mechanisms underlying control of gene expression have been thoroughly studied in yeast, environmental stress has been shown to play a large role in the dynamics of gene expression [13], [32], [47]. Thus, KINC has the potential to identify novel gene co-expression patterns from a variety of input gene expression datasets.

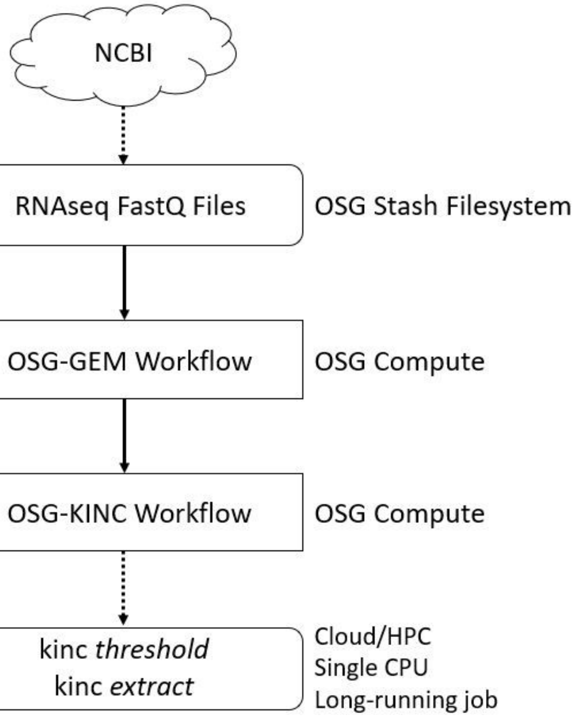


Figure 5.1 GCN construction workflow using the OSG and local HPC resources. Raw FastQ files are downloaded onto the OSG Stash filesystem from the NCBI database. The OSG-GEM workflow can be utilized to process the raw data using OSG resources. The output from OSG-GEM is used as input into the OSG-KINC workflow. The output from OSG-KINC is transferred to a cloud or HPC resource for RMT thresholding and GCN extraction.

A yeast GEM was constructed using 439 *S. cervisiae* paired- end Illumina RNAseq datasets downloaded from the NCBI Sequence Read Archive database [29]. Raw reads were cleaned using Trimmomatic-0.33 [10], mapped to the R64 build of the reference genome [18] using hisat2-2.0.1-beta [27], and RNA transcript abundances were quantified using cufflinks-2.2.1 [45]. A Kolmogorov-Smirnov test was performed to identify outlier samples based on the global distribution of FPKM (fragments per kilobase per exon of million mapped reads) values. 251 outlier samples were removed, and the remaining matrix was log2 transformed and quantile normalized using the preprocessCore [9] R library to reduce technical noise between samples. The preprocessed yeast GEM was input into the OSG-KINC workflow. This native yeast GEM is included in the Github repository as a unit test file. To demonstrate the portability of OSG-KINC between environments, the Chameleon Cloud [31] was used to build a GCN from this dataset. Upon completion of OSG-KINC, RMT identified a significance threshold of 0.8501 which was used to extract the GCN.

The resulting GCN contains 2966 nodes that are connected by 6766 edges, and demonstrates scale-free topology with an average connectivity of 4.270 (Figure 5.2). To identify groups of highly connected nodes in the graph, Link Community Modules (LCM) were identified using the linkcomm R package [26]. LCM uses hierarchical clustering to identify clusters of nodes, allowing for a given node to be a member of multiple clusters [6]. 318 unique LCM modules were identified. These modules may be further investigated to identify novel gene expression patterns driving biological processes such as fermentation under varying biological conditions.

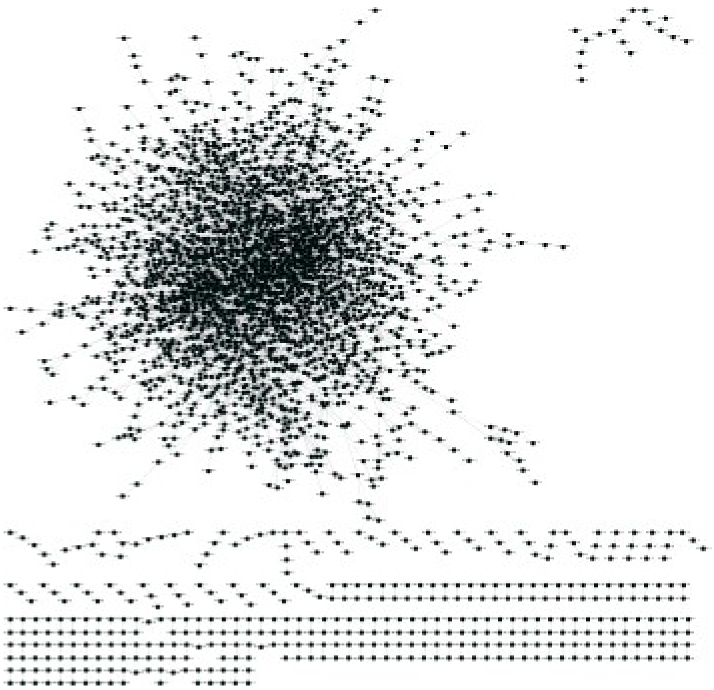


Figure 5.2 A Yeast GCN was constructed using the input data provided in the OSG-KINC Github repository. After OSG-KINC execution, RMT thresholding, and network extraction, the resulting graph was visualized using Cytoscape [41].

Workflow Performance. Total workflow walltime was recorded after executing OSG- KINC on the OSG using 1000 jobs compared to 8000 jobs. Each test was submitted three times, making sure that multiple workflows were not running at the same time. As shown in Figure 5.3, the workflow took longer to run when submitting with a larger number of jobs. While the average job walltime was significantly lower when submitting more jobs, the over- head of queue time, time required by Pegasus and HTCondor to submit and monitor each job, and a larger number of retried jobs may have lowered the efficiency of the workflow (Table 5.1). Workflow walltime will show significant variation between experiments, due to differences in the availability of opportunistic resources over time as well as the reliability of hardware that is received by each job. On average, the peak number of running jobs was higher when submitted with 1000 jobs (as noted from the OSG user dashboard). However, even in the event where the workflow submitted with 8000 jobs had a higher peak number of running jobs, the workflow took longer to run. Thus, the number of jobs that are submitted plays a large role in the overall runtime of the workflow.

Conclusion

OSG-KINC was developed to enable high-throughput GCN construction using resources available on the OSG. While the workflow has been optimized for usability, there remain bottlenecks in the GCN construction process. Failed jobs and overhead associated with job scheduling and execution play a detrimental role in workflow performance. Furthermore, the user must select an appropriate number of jobs to submit for a given input dataset. The optimal number of jobs depends on both the number of rows and columns in the input matrix, as well as the density of non-missing gene expression values. Thus, the user may need to submit test runs to determine an efficient number of jobs to submit. In addition, output from the OSG-KINC workflow must be transferred to a different computing environment to complete the GCN construction process.

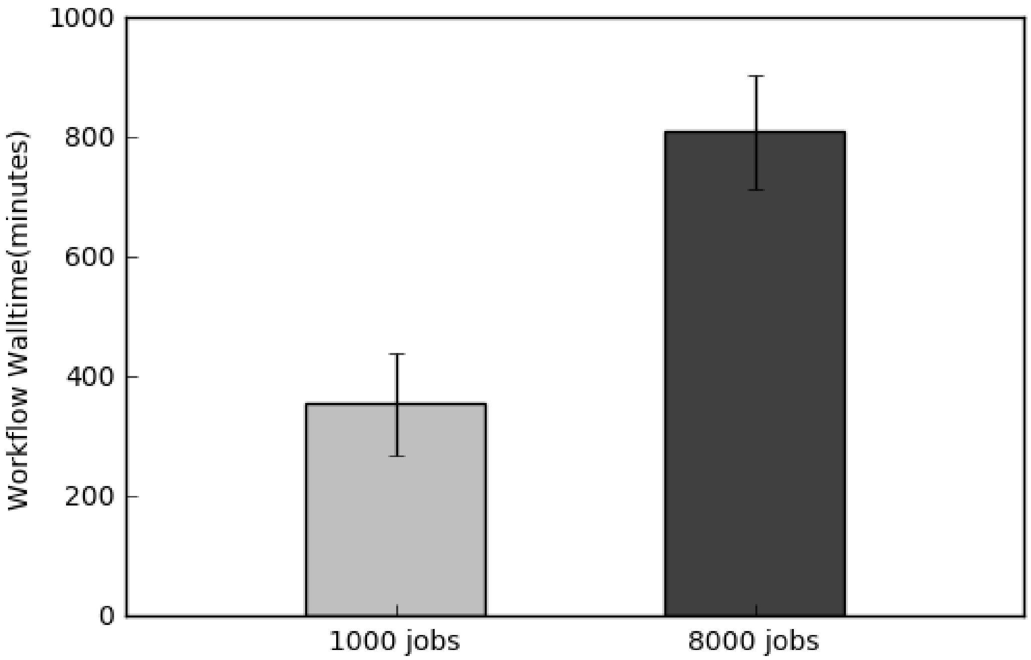
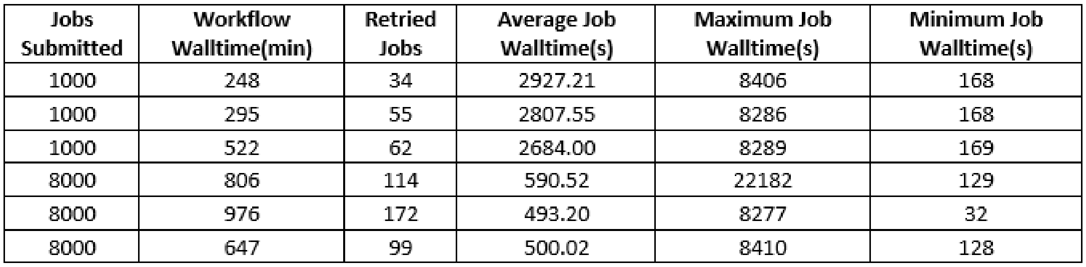


Figure 5.3 The total workflow walltime (n=3) was compared for 1000 and 8000 OSG-KINC job submissions. The mean of three replicates was plotted using the pyplot [24] library. Error bars represent with standard error of the mean calculated using the scipy [25] stats library with default function arguments.



Jobs Submitted	Workflow Walltime(min)	Retried Jobs	Average Job Walltime(s)	Maximum Job Walltime(s)	Minimum Job Walltime(s)
1000	248	34	2927.21	8406	168
1000	295	55	2807.55	8286	168
1000	522	62	2684.00	8289	169
8000	806	114	590.52	22182	129
8000	976	172	493.20	8277	32
8000	647	99	500.02	8410	128

Table 5.1 Workflow execution stats were gathered using the pegasus- statistics -s all command. The total workflow walltime, number of retried jobs, average job walltime, maximum job walltime, and minimum job walltime were recorded for three workflows submitted with 1000 jobs and three workflows submitted with 8000 jobs.

OSG-KINC is a core workflow utilized by the SciDAS project (National Science Foundation Award #1659300), which aims to enable petascale data processing by alleviating the problems discussed above. Efforts to further develop the OSG-KINC workflow will reduce bottlenecks in GCN construction. First, efforts are underway to GPU-optimize the KINC source code. Recently, support for iRODS [2], [39] has been added to the OSG-KINC workflow. This allows the user to stage input, intermediate, and output files on a remote iRODs server, which prevents excessive data movement between resources. The ability to access distributed computing resources such as the Open Science Grid in combination with stable HPC and cloud resources will enable OSG-KINC to execute all three stages of the GCN construction process - KINC similarity, KINC threshold, and KINC extract - without the need to transfer output of the workflow to local HPC resources for downstream processing. Efforts to optimize data movement between resources, task to resource matching, and user input will reduce bottlenecks in GCN construction and other workflows. In its current state, OSG-KINC provides a stable resource highly parallel gene correlation analysis using distributed computing resources provided by the OSG.

Acknowledgement

The authors would like to thank the OSG support staff for their technical assistance. This research utilized resources provided by the Open Science Grid, which is supported by the National Science Foundation (NSF Award #1148698) and the U.S. Department of Energy's Office of Science, as well as the Chameleon Cloud (NSF

Award #1743358). Clemson University is acknowledged for generous allotment of compute time on the Palmetto Cluster. Portions of this project were supported by the National Science Foundation (NSF) SciDAS project (Award #1659300).

References

[1] CVMFS - OSG Documentation. http://opensciencegrid.github.io/docs/worker-node/install-cvmfs/.

[2] irods: Open Source Data Management Software, 2017. https://github.com/irods/irods.

[3] KINC, 2017. https://github.com/SystemsGenetics/KINC.

[4] OSG-GEM, 2017. https://github.com/feltus/OSG-GEM.

[5] OSG-KINC, 2017. https://github.com/feltus/OSG-KINC.

[6] Y. Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. Nature, 466(7307):761–764, Aug 2010.

[7] K. Aoki, Y. Ogata, and D. Shibata. Approaches for extracting practical information from gene co-expression networks in plant biology. Plant Cell Physiol., 48(3):381–390, Mar 2007.

[8] N. S. Baliga, J. L. Bjorkegren, J. D. Boeke, M. Boutros, N. P. Crawford, A. M. Dudley, C. R. Farber, A. Jones, A. I. Levey, A. J. Lusis, H. C. Mak, J. H. Nadeau, M. B. Noyes, E. Petretto, N. T. Seyfried, L. M. Steinmetz, and S. C. Vonesch. The State of Systems Genetics in 2017. Cell Syst, 4(1):7–15, Jan 2017.

[9] Bolstad BM. preprocessCore: A collection of pre-processing functions.

[10] A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics, 30(15):2114–2120, Aug 2014.

[11] A. M. Cavanaugh and S. L. Jaspersen. Big Lessons from Little Yeast: Budding and Fission Yeast Centrosome Structure, Duplication, and Function. Annu. Rev. Genet., Sep 2017.

[12] M. Civelek and A. J. Lusis. Systems genetics approaches to understand complex traits. Nat. Rev. Genet., 15(1):34–48, Jan 2014.

[13] E. de Nadal, G. Ammerer, and F. Posas. Controlling gene expression in response to stress. Nat. Rev. Genet., 12(12):833–845, Nov 2011.

[14] Ewa Deelman, Gurmeet Singh, Mei-Hui Su, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, G. Bruce Berriman, John Good, Anastasia Laity, Joseph C. Jacob, and Daniel S. Katz. Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Scientific Programming Journal, 13(3):219–237, 2005.

[15] Ewa Deelman, Karan Vahi, Gideon Juve, Mats Rynge, Scott Callaghan, Philip J. Maechling, Rajiv Mayani, Weiwei Chen, Rafael Ferreira da Silva, Miron Livny, and Kent Wenger. Pegasus, a workflow management system for science automation. Future Gener. Comput. Syst., 46(C):17– 35, May 2015.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38, 1977.

[17] Abbasali Emamjomeh, Elham Saboori Robat, Javad Zahiri, Mahmood Solouki, and Pegah Khosravi. Gene co-expression network reconstruction: a review on computational methods for inferring functional information from plant-based expression data. Plant Biotechnology Reports, 11(2):71–86, Apr 2017.

[18] S. R. Engel, F. S. Dietrich, D. G. Fisk, G. Binkley, R. Balakrishnan, M. C. Costanzo, S. S. Dwight, B. C. Hitz, K. Karra, R. S. Nash, S. Weng, E. D. Wong, P. Lloyd, M. S. Skrzypek, S. R. Miyasato, M. Simison, and J. M. Cherry. The reference genome sequence of Saccharomyces cerevisiae: then and now. G3 (Bethesda), 4(3):389–398, Mar 2014.

[19] S. P. Ficklin, L. J. Dunwoodie, W. L. Poehlman, C. Watson, K. E. Roche, and F. A. Feltus. Discovering Condition-Specific Gene Co-Expression Patterns Using Gaussian Mixture Models: A Cancer Case Study. Sci Rep, 7(1):8617, Aug 2017.

[20] I. Foster. Globus online: Accelerating and democratizing science through cloud-based services. IEEE Internet Computing, 15(3):70–73, May 2011.

[21] S. M. Gibson, S. P. Ficklin, S. Isaacson, F. Luo, F. A. Feltus, and M. C. Smith. Massive-scale gene co-expression network construction and robustness testing using random matrix theory. PLoS ONE, 8(2):e55871, 2013.

[22] B. A. Gruning, J. Fallmann, D. Yusuf, S. Will, A. Erxleben, F. Eggen- hofer, T. Houwaart, B. Batut, P. Videm, A. Bagnacani, M. Wolfien, S. C. Lott, Y. Hoogstrate, W. R. Hess, O. Wolkenhauer, S. Hoffmann,

A. Akalin, U. Ohler, P. F. Stadler, and R. Backofen. The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic Acids Res., Jun 2017.

[23] C. He, H. Fu, C. Guo, W. Luk, and G. Yang. A fully-pipelined hardware design for gaussian mixture models. IEEE Transactions on Computers, 66(11):1837–1850, Nov 2017.

[24] J. D. Hunter. Matplotlib: A 2d graphics environment. Computing In Science & Engineering, 9(3):90–95, 2007.

[25] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[26] A. T. Kalinka and P. Tomancak. linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. Bioinformatics, 27(14):2011–2012, Jul 2011.

[27] D. Kim, B. Langmead, and S. L. Salzberg. HISAT: a fast spliced aligner with low memory requirements. Nat. Methods, 12(4):357–360, Apr 2015.

[28] P. Langfelder and S. Horvath. WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics, 9:559, Dec 2008.

[29] R. Leinonen, H. Sugawara, and M. Shumway. The sequence read archive. Nucleic Acids Res., 39(Database issue):19–21, Jan 2011.

[30] F. Luo, Y. Yang, J. Zhong, H. Gao, L. Khan, D. K. Thompson, and J. Zhou. Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. BMC Bioinformatics, 8:299, Aug 2007.

[31] Joe Mambretti, Jim Chen, and Fei Yeh. Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn). In Proceedings of the 2015 International Conference on Cloud Computing Research and Innovation (ICCCRI), ICCCRI '15, pages 73–79, Wash- ington, DC, USA, 2015. IEEE Computer Society.

[32] J. E. McCarthy. Posttranscriptional control of gene expression in yeast. Microbiol. Mol. Biol. Rev., 62(4):1492–1553, Dec 1998.

[33] N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin. The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences. PLoS Biol., 14(1):e1002342, Jan 2016.

[34] U. Nagalakshmi, K. Waern, and M. Snyder. RNA-Seq: a method for comprehensive transcriptome analysis. Curr Protoc Mol Biol, Chapter 4:1–13, Jan 2010.

[35] J. Petereit, S. Smith, F. C. Harris, and K. A. Schlauch. petal: Co-expression network modelling in R. BMC Syst Biol, 10 Suppl 2:51, Aug 2016.

[36] W. L. Poehlman, M. Rynge, C. Branton, D. Balamurugan, and F. A. Feltus. OSG-GEM: Gene Expression Matrix Construction Using the Open Science Grid. Bioinform Biol Insights, 10:133–141, 2016.

[37] Ruth Pordes, Don Petravick, Bill Kramer, Doug Olson, Miron Livny, Alain Roy, Paul Avery, Kent Blackburn, Torre Wenaus, Frank Wrthwein, Ian Foster, Rob Gardner, Mike Wilde, Alan Blatecky, John McGee, and Rob Quick. The open science grid. Journal of Physics: Conference Series, 78(1):012057, 2007.

[38] Y Qin, HK Yalamanchili, J Qin, B Yan, and J Wang. The current status and challenges in computational analysis of genomic big data. Big Data Research, 2015.

[39] Arcot Rajasekar, Reagan Moore, Chien-yi Hou, Christopher A. Lee, Richard Marciano, Antoine de Torcy, Michael Wan, Wayne Schroeder, Sheau-Yen Chen, Lucas Gilbert, Paul Tooby, and Bing Zhu. iRODS Primer: Integrated Rule-Oriented Data System. Morgan and Claypool Publishers, 2010.

[40] M. N. Rezaei, E. Dornez, P. Jacobs, A. Parsi, K. J. Verstrepen, and C. M. Courtin. Harvesting yeast (Saccharomyces cerevisiae) at different physiological phases significantly affects its functionality in bread dough fermentation. Food Microbiol., 39:108–115, May 2014.

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

[42] Jamie Shiers, Frank Olaf Berghaus, Jakob Blomer, Gerardo Ganis, Sunje Dallmeier-Tiessen, Tibor Simko, and German Cancio Melia. CERN Services for Long Term Data Preservation. Technical Report CERN-IT- Note-2016-004, CERN, Geneva, Jul 2016.

[43] J. Steensels and K. J. Verstrepen. Taming wild yeast: potential of conventional and nonconventional yeasts in industrial fermentations. Annu. Rev. Microbiol., 68:61–80, 2014.

[44] Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the condor experience. Concurrency - Practice and Experience, 17(2-4):323–356, 2005.

[45] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat. Biotechnol., 28(5):511–515, May 2010.

[46] S. van Dam, U. Vosa, A. van der Graaf, L. Franke, and J. P. de Ma- galhaes. Gene co-expression analysis for functional classification and gene-disease predictions. Brief. Bioinformatics, Jan 2017.

[47] S. D. Willis, A. K. M. N. Hossian, N. Evans, and M. J. Hickman. Measuring mRNA Levels Over Time During the Yeast S. cerevisiae Hypoxic Response. J Vis Exp, (126), Aug 2017.

[48] Cecily J. Wolfe, Isaac S. Kohane, and Atul J. Butte. Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. BMC Bioinformatics, 6(1):227, Sep 2005.

[49] S. Zhao, W. P. Fung-Leung, A. Bittner, K. Ngo, and X. Liu. Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. PLoS ONE, 9(1):e78644, 2014.

CHAPTER SIX

CONCLUSION

The results presented in this dissertation demonstrate the utility of condition-specific gene coexpression network (GCN) analysis as a biomarker discovery tool. Utilization of a novel GCN construction algorithm, Knowledge Independent Network Construction (KINC), was essential to the work presented in this dissertation. KINC is capable of identifying condition-specific GCN edges by performing sample clustering before correlation analysis for every gene pair comparison [1]. The chapters discussed in this dissertation are early use-cases of the KINC software, demonstrating its application in biomarker discovery using two unique datasets.

In Chapter 2, I discuss the construction of a GCN using 1,009 kidney cancer datasets. These datasets spanned conditions such as cancer subtype, tumor stage, and patient gender. In addition, I curated mutation profiles for the corresponding patients, which allowed me to identify GCN edges that were specific to patients with specific mutations. By comparing the GCN edges that were specific to two common kidney cancer mutation profiles, I discovered two lists of biomarkers that contained unique genes. However, these gene lists were both enriched for biological function related to T cell activation and immune response, revealing convergent function of alternate genetic lesions. While the data analyzed in Chapter 2 spanned over 1,000 samples, the data analyzed in Chapter 3 was generated from a much smaller de novo RNA sequencing experiment.

Chapter 3 presents the construction of a root GCN using 30 root maturation zone samples spanning control and inoculated samples at five time points. Differential gene expression analysis revealed hundreds of up and down-regulated genes at specific time points, which were difficult to translate into meaningful biomarkers. While the small sample size in this experiment made it impossible to detect specific GCN edges that were unique to time point or inoculated samples, the GCN was utilized to identify nodulation biomarkers. By performing clustering of nodes in the GCN, functional modules were identified that demonstrated consistent expression patterns across samples over time. Three of these modules were comprised entirely of genes that were differentially expressed at one specific time point. These results demonstrate that combining GCN analysis with other common biomarker discovery techniques can reduce a list of biomarkers from thousands of genes down to small lists containing less than 20 genes.

Performing the experiments described above required significant computational resources and stable data processing pipelines. During my PhD studies, I encountered significant roadblocks in my ability to generate insights from large RNA sequencing datasets in a reasonable timeframe. As a result, I ported my core workflows into the Pegasus workflow management system [2] which allowed me to utilize the grid computing resources of the Open Science Grid (OSG) [3]. Chapter 4 discusses the development of an RNA sequencing data processing workflow, OSG-GEM, which is executable on the OSG infrastructure [4]. The results demonstrate that sequence FastQ files can be split into small pieces to process in parallel, and still generate the same result as the un-split files. The results also highlight bottlenecks in this process, as demonstrated by longer computational run time of a single dataset processed on the OSG compared to the same dataset processed on the Palmetto Cluster at Clemson University. Regardless, this workflow enables users to scale up their experiments to hundreds or thousands of samples without overloading their local computing cluster or paying for cloud credits. Chapter 5 discusses the development of the OSG-KINC workflow, which enables users to perform genome-wide correlation analysis on the OSG [5]. This workflow was critical to generating results with KINC, as thousands of computers are necessary to perform this analysis. Still, this chapter discusses bottlenecks in the GCN construction process, such as the need to transfer output from the OSG-KINC workflow to a large-memory node that the OSG does not provide.

In conclusion, this dissertation contributes to science by demonstrating that a common systems genetics approach, GCN analysis, can be applied in unique ways as a method for biomarker discovery from RNA sequencing data. The computational challenges that I encountered during this work resulted in the need to develop workflows that enabled execution of genomics workflows on geographically distributed grid computing resources. By applying these workflows to an animal and a plant case study, I identified specific biomarkers that can be used as candidates for functional validation. These results demonstrate that a holistic approach of dissecting the basis of complex traits can be used to identify a specific set of candidate biomarkers.

References

1. Ficklin, S.P., et al., *Discovering Condition-Specific Gene Co-Expression Patterns Using Gaussian Mixture Models: A Cancer Case Study.* Sci Rep, 2017. **7**(1): p. 8617.

2. Deelman, E., et al., *Pegasus, a workflow management system for science automation.* Future Gener. Comput. Syst., 2015. **46**(C): p. 17-35.

3. Ruth, P., et al., *The open science grid.* Journal of Physics: Conference Series, 2007. **78**(1): p. 012057.

4. Poehlman, W.L., et al., *OSG-GEM: Gene Expression Matrix Construction Using the Open Science Grid.* Bioinformatics and Biology Insights, 2016. **10**(5814-BBI-OSG-GEM:-Gene-Expression-Matrix-Construction-Using-the-Open-Science-Gr.pdf): p. 133-141.

5. Poehlman, W.L., et al. *OSG-KINC: High-throughput gene co-expression network construction using the open science grid*. in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2017.

APENDIX A - LICENSING INFORMATION

Chapter four was published under the CC-BY-NC 3.0 license, allowing full non-commercial reuse for this dissertation. The version included in this dissertation was the original submitted version.

https://creativecommons.org/licenses/by-nc/3.0/

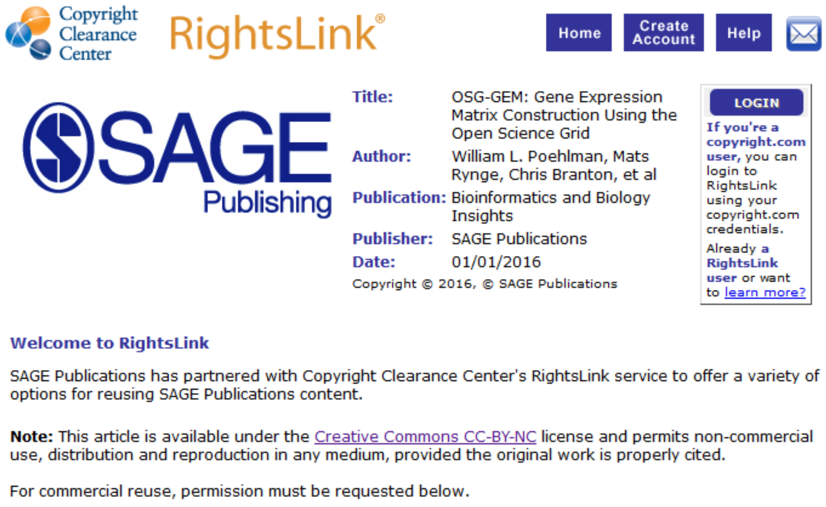
http://journals.sagepub.com/doi/abs/10.4137/BBI.S38193

OSG-GEM: GENE EXPRESSION MATRIX CONSTRUCTION USING THE OPEN SCIENCE GRID

William L. Poehlman[1], Mats Rynge[2], Chris Branton[3], D. Balamurugan[4], and Frank A. Feltus[1*]

[1]Clemson University, Department of Genetics & Biochemistry, SC 29634, USA; [2]University of Southern California, Information Sciences Institute, Marina Del Rey, CA 90292, USA; [3]Center for Computation & Technology, Louisiana State University, Baton Rouge, Louisiana 70803, USA; [4]University of Chicago, Computation Institute, Chicago, IL 60637, USA.

Published: *Bioinformatics and Biology Insights.* August 2nd 2016.
https://doi.org/10.4137/BBI.S38193



Title: OSG-GEM: Gene Expression Matrix Construction Using the Open Science Grid
Author: William L. Poehlman, Mats Rynge, Chris Branton, et al
Publication: Bioinformatics and Biology Insights
Publisher: SAGE Publications
Date: 01/01/2016
Copyright © 2016, © SAGE Publications

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.
Already a RightsLink user or want to learn more?

Welcome to RightsLink

SAGE Publications has partnered with Copyright Clearance Center's RightsLink service to offer a variety of options for reusing SAGE Publications content.

Note: This article is available under the Creative Commons CC-BY-NC license and permits non-commercial use, distribution and reproduction in any medium, provided the original work is properly cited.

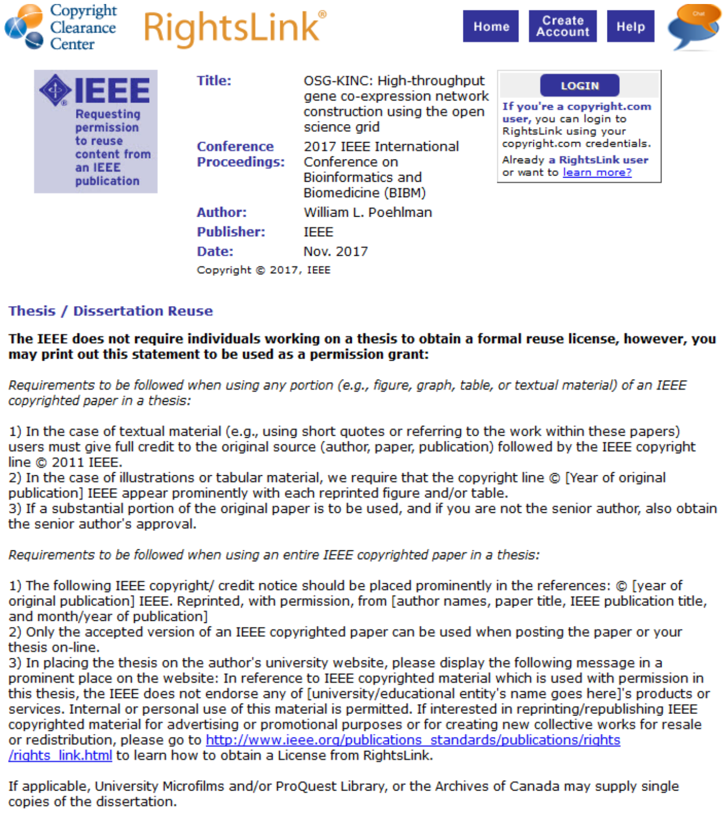
For commercial reuse, permission must be requested below.

Chapter five was published under IEEE copyright, which permits reuse for a dissertation.

OSG-KINC: HIGH-THROUGHPUT GENE CO-EXPRESSION NETWORK CONSTRUCTION USING THE OPEN SCIENCE GRID

William L. Poehlman[*], Mats Rynge[‡], D. Balamurugan[§], Nicholas Mills[†], and
Frank A. Feltus[*]
[*]Department of Genetics and Biochemistry,
Clemson University, Clemson, SC 29634
[†]Holcombe Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
[‡]Information Sciences Institute,
University of Southern California, Marina Del Rey, CA 90292
[§]Computation Institute,
University of Chicago, Chicago, IL 60637

Published: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Date of Conference**:** 13-16 Nov. 2017. Date Added to IEEE ***Xplore*****:** 18 December 2017. Electronic ISBN**:** 978-1-5090-3050-7. DOI: 10.1109/BIBM.2017.8217938


Copyright Clearance Center
RightsLink®
Home
Create Account
Help
IEEE
Requesting permission to reuse content from an IEEE publication
LOGIN

Title: OSG-KINC: High-throughput gene co-expression network construction using the open science grid

Conference Proceedings: 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)

Author: William L. Poehlman

Publisher: IEEE

Date: Nov. 2017

Copyright © 2017, IEEE

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.
Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.