OMICS DATA EXPLORATION: ACROSS SCALES AND DIMENSIONS

by

Gang Su

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Bioinformatics)
in the University of Michigan
2013

Doctoral Committee:

Professor Brian D. Athey, Co-Chair

Research Associate Professor Fan Meng, Co-Chair

Professor Charles F. Burant

Associate Research Scientist Barbara Mirel

Assistant Professor Maureen Sartor

DEDICATION

I dedicate this thesis especially to my grandparents. My grandfather grew up in a shattered nation, and survived in countless brutal battles through WWII, but he never lost his hopes and disciplines, and lived happily even to this day. Whenever I talk with him, I realize that my peaceful life today is nothing but a gift from those great sacrifices the older generations have made – and I should struggle to make my own contributions in this new era and to make something better for the future. He wished his grandson could reach the summit of knowledge someday on behalf of the family, and now I am proudly fulfilling his wish. My grandmother, loved her family all her life, and loved me every second since my first day.

I also dedicate this work to my parents, it's with their persistent love and care that I had a carefree and beautiful childhood. My father was the inspiration for me to go out and see the world. As a sailor in his early years, his footprints covered the majority of the globe, leaving only Europe and North America uncovered due to the political segregation back then. I am now completing the left pieces of this big puzzle, and hopefully I will have much more in the future to pass on to. My mother developed my interest in reading, drawing and later in science. Her devotion and love to the family made me strong and resilient.

And to my friends – I always believed that those who surround us define us. I am so grateful to the great experiences with them, and it's beyond words.

Last but not the least; I dedicate this work to dear Yoyo: my great friend, my career partner, and my life-long companion. She gave me the feeling of completeness, and made me a better individual everyday. There are so much more to explore ahead of us, in the writing of our stories together. Get ready; a new chapter is right ahead.

ACKNOWLEDGEMENTS

I would like to first acknowledge my advisor, Dr Fan Meng, for years of patient guidance and support through the course of my PhD. Also to sincerely thank Dr Brian Athey and Charles Burant, for rigorous and insightful academic advising, Dr Barbara Mirel for expert knowledge on user-interface design and workflows, as well as Dr Maureen Sartor for statistics and the structuring of the thesis. I couldn't have come this far without this excellent interdisciplinary committee. I would also like to thank Dr David States and Steve Qin in the early rotation advising when I began my PhD. Last but not least, I would like to especially thank Dr Margit Burmeister. She has been a great mentor and a great friend, and offered me enormous help for both research and life.

I would also like to acknowledge my friends. Ann Arbor has been a second hometown to me and my life is tied with the people here. PhD did take sometime and I have met several batches of friends. Their support and accompany gave me ideas, happiness, and a lot of memorable moments. I sincerely thank you all, for everything.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The rapid development and adoption of high throughput technologies has led to an avalanche of omics data, including those from genome, transcriptome, proteome and metabolome, from individual laboratories as well as global-scale collaborative efforts. The major ensuing challenge is then how to analyze, explore and extract new biomedical knowledge from such omics datasets. This thesis attempted to address some of these challenges by 1) developing novel tools for flexible searching, clustering and visualizing omics networks and pathways 2) developing novel robust statistical workflows to identify confident associations that lead to discovery of new cell-line specific bio-signatures from NCI-60 omics datasets with high variability and missing measurements, and most notably, 3) conceiving and developing a novel visual data exploration model, the CoolMap, to bring multi-scale, versatile and flexible visual data mining capabilities to structured two-dimensional omics datasets. CoolMap's unique capabilities were demonstrated through several use cases including a mother-child nutrient/epigenetics study, and enables efficient and flexible identification of strongly correlated high-level ontological concepts as well as low-level specific measurements for data-driven hypothesis generation.

Chapter 1 Introduction

I would like to begin my thesis by paraphrasing an Indian allegory[i]:

A merchant brought an elephant to the Emperor. Knowing no one has ever seen this creature before, the Emperor summoned ten most knowledgeable people to his palace and put them to a test. He blindfolded them and sent them to touch around the elephant, and then tell him what they felt the elephant should look like. The first person grabbed the leg of the elephant and said the elephant should have the shape of a pillar. The second person held the trunk of the elephant and stated the elephant felt like a snake. The third person touched the ear of the elephant and told the Emperor the elephant was interestingly flat like a giant tropical leaf. The last person got hold of the tail of the elephant and claimed that he was holding a snake. The Emperor then ordered them to take off the cloth around their eyes and said: if you draw your conclusion from limited observations, you will miss the big picture. ***We must analyze the problem as a whole.***

What does this allegory tell us? Even about two thousand years ago, people in Asia already understood the importance of studying a problem in its entirety; or in other words, in a 'systems' manner. This requires the following practices: first of all, compile a dataset that contains as complete and detailed information as possible; secondly, analyze the problem in each information domain, identify intra-domain associations, and make domain-specific inferences; and most importantly, build cross-domain relationships and generate a comprehensive, integrated and systems conclusion. The last step produces a new entry of knowledge that can be used to create solutions for existing challenges or answer new questions.

For sciences that are heavily reliant on measurements such as Physics, Chemistry and Biology, a major obstacle to analyzing a particular matter in using the systems view has been how quickly, accurately and affordably raw data could be obtained. Taking DNA sequencing as an example, the initial cost of obtaining raw sequence data could be as high as $10 for 1

[i] For more details: http://en.wikipedia.org/wiki/Blind_men_and_an_elephant

[ii] http://www.singularity.com/charts/page73.html

base pair (bp) in 1990[ii], not to mention the cost for data storage and analysis. Therefore in the pre-high-throughput era when obtaining large amount of measurement data was impractical, researchers focused on specific and manageable research problems: often only a handful of genes or proteins are studied in a well-controlled study. Many key molecules were identified and carefully studied, such as the tumor suppressor p53 gene, the transcription modulator gene NF-κB, etc. However, for processes that involve an ensemble of factors such as obesity, hair-loss or cancer, it is necessary to study 'snapshots' of the whole biological system to avoid being 'blindfolded'. While the reductionist approaches predominantly drove the major progresses in biomedical research in the last century, holistic interpretation and understanding of the data are becoming increasingly more necessary for complex processes.

With the rapid development in chemistry, biomedical engineering and information sciences, the world has entered an era of information explosion. The human genome project reached the first milestone in 2001, when the first draft genome containing 3 Gigabase (Gb) was released under an international collaborative effort in 2001. The cost of sequencing dropped to about $10k for 1 Megabase (Mb) base pairs (bp) and as low as a quarter per 1Mb in 2010. With the arrival of Next Generation Sequencing (NGS), now it's now possible to sequence a human genome at 30X coverage in less than 10 days with cost less than $10k[1]. Meanwhile, more cost-effective, efficient and precise oligonucleotide based arrays gradually replaced traditional blotting technology in high-throughput experiments. Molecular signatures were discovered that not only reflected pivot elements in disease processes, but also could be used as phenotypic classifiers for pathological diagnostics. Amongst this context, the term 'Omics', referring to study "all constituents considered collectively" in a systems view, has become increasing popular. The most referred ones are genomics, transcriptomics, proteomics and metabolomics, and the integrated omics analysis[2].

[ii] http://www.singularity.com/charts/page73.html

1.1 The Omics Landscape

Genomics is the systematic elucidation of a target organism's genome[3]. It entails the determination of DNA sequences, and characterization of the structure and function of genomic elements. With the latest break-through in sequencing technology, currently there are about 60 fully sequenced genomes for multicellular eukaryotic organisms available in the Ensembl database since the first draft completion of the human genome project[iii]. The availability of assembled genomes provide the critical scaffolds needed for a wide variety of biomedical research, such as understanding the evolution of genes and species and identifying differences between individuals (polymorphisms). With the continuous decreasing cost of sequencing, personal genome is becoming affordable to the general public and personalized medicine is expected to deliver more cost-effective treatments[4–7]. Genomics data are deposited in several major repositories, such as UCSC[8,9], NCBI[10], Ensembl[11]. Although genomics research has clarified many biological problems, the static DNA sequences cannot accurately represent the dynamic metabolic and physiological state of the organism. Therefore Genomics is often studied along with other downstream Omics datasets.

Transcriptomics is the quantitative study of the transcriptome that consists of the complete set of transcripts in a cells or tissues, including mRNAs that are directly used as encoding templates for protein synthesis, and regulatory RNAs such as non-coding RNAs and small RNAs[12]. The term was first proposed by Charles Auffray and was one of the early members in the Omics family[13]. Transcript expression profile is directly associated with developmental, physiological and pathological processes[14].

A variety of technologies have been developed and applied to Transcriptomics research, such as hybridization based Microarrays offered by Affymetrix[iv] and Illumina[v]. The microarray technology remarkably facilitates the holistic analysis of transcriptome. The

[iii] Ensembl: http://useast.ensembl.org/index.html

[iv] http://www.affymetrix.com/

[v] http://www.illumina.com/

recently developed RNA-Seq technology, considered as the successor of microarrays, has demonstrated the capability of accurately measure transcription activity digitally on single base-resolution, in a high-throughput manner, with high dynamic range, high reproducibility and sensitivity and can be executed at relatively low cost[13,14]. These advances offered new holistic and detailed insights into the transcriptome. Bioinformatics Tools supporting RNA-Seq, such as designing RNA-Seq experiments, rapid short sequence alignment, detecting splice junctions, identification of differentially expressed genes, or transcriptome visualization are also publicly available[15–19]. The plethora of publicly available Transcriptomics data, such as those deposited in public domains such as NCBI GEO[20] and European Bioinformatics Institute (EBI)[21,22], also stimulated the rapid development of computational and statistical methods that could be immediately applied to other Omics datasets.

Transcriptomics have been applied to understand research topics such as cell differentiation, cell cycle, development and carcinogenesis. It has been widely applied in diagnostics and biomarker discovery. Golub et al. demonstrated that molecular signatures from selected genes effectively carry sufficient information to classify different leukemia types in their cornerstone paper[23]. The RNA-Seq technology also opened new frontiers in transcriptomics such as more accurate transcript start site and exon boundary mapping, strand-specific measurements, much more precise characterization of alternative splicing patterns, detection of gene fusion, de novo transcriptomics, in which no prior reference genome is known, and Single Cell Transcriptomics (small sample Transcriptomics) that measurement is done with very little sample materials[12–14,24,25].

Proteomics concerns the characterization of all proteins' peptide sequence, structure, function, abundance and interactions in a cell. Proteins are the actual molecular machinery that execute the commands encoded in the genome[26]. Although transcriptomics can now accurately capture gene expression profiles, there's another layer of regulation from mRNA to protein. Gene expression can be regulated at the mRNA level via alternative splicing and siRNAs, and proteins are subject to one or more post-translational modifications such as phosphorylation, methylation, glycosylation, unbiquitination, glycation, etc[vi]. Processes such

[vi] http://en.wikipedia.org/wiki/List_of_sequenced_eukaryotic_genomes

as protein localization, transport, protein-protein and protein-DNA interaction, are also impossible to be captured by Transcriptomics alone. Moreover, Proteomics is also more sensitive to the cellular state change and external stimuli[26,27].

There are several technologies developed to cater proteomics needs. Protein sequence and quantity can be determined using Mass Spectrometry (MS)[28]; physical protein-protein interactions can be validated using Yeast-2-Hybrid systems or high-throughput protein chips[29]; proteins from a sample can be separated using 2D gels; protein structures can be characterized using X-ray crystallography or Nuclear Magnetic Resonance (NMR) spectroscopy; in vivo protein-DNA interactions can be investigated using ChIP-seq[30]; protein localization can be directly observed using fluorescence imaging. Rich proteomics data encompassing sequence, structure, function, abundance and interaction are publicly available online, such as Proteomics IDEntifications database (PRIDE)[31], RCSB Protein Data Bank (PDB)[32], UniProt[33], Reactome[34] and MiMI[35] for protein-protein interaction. Similar to Transcriptomics, Proteomics data have also been applied to medical practices such as biomarker discovery and disease diagnostics such as cancer and diabetes[25,36,37], and to other domains such as ecology and population biology[38]. It has also been integrated with Transcriptomics to reveal dynamics from transcript to protein and the ensemble of interactions to underlying molecular mechanisms[39–41].

Metabolomics, or metabolic profiling, consists of the quantitative characterization of metabolites in a biological system[42]. Metabolites are rapid fluctuating and interchangeable small molecules (<1500 Da) that directly participate in the cellular regulatory processes and capture the physiological state of the cell. Currently there are about more than 8k annotated metabolites and hundreds of metabolic pathways stored in databases such as the Human Metabolome Database (HMDB)[43,44], METLIN[45], Edinburgh Human Metabolic Network (EHMN)[46], and Kyoto Encyclopedia of Genes and Genomes (KEGG)[47], and used by programs for analyzing metabolomics data, such as Metsscape[48,49]. Generally, there are three major metabolic profiling techniques[50]: targeted quantitative analysis that profiles only a limited number but well-characterized metabolites; unbiased profiling that measures a large number of metabolites (from a couple hundred to a few thousand) simultaneously, and metabolic fingerprinting that captures the entire 'snapshot' of the metabolic state of a cell culture or tissue.

Along with the technological advances in other Omics, the technological advances have also remarkably improved the measurement of metabolites. The pipeline usually consists of a molecular separation procedure, such as Gas Chromatography (GC), Liquid Chromatography (LC), High-Performance Liquid Chromatography (HPLC), Ultra Performance Liquid Chromatography (UPLC), Capillary Electrophoresis (CE), that is then coupled with a characterization procedure, such as Mass-Spectrometry (MS), Nuclear Magnetic Resonance (NMR) spectroscopy, and Fourier Transform Infrared Spectroscopy (FT-IR)[50–53].

Metabolomics has a variety of applications, such as plant sciences and agriculture to capture the states of growth, development and plants' response to external stimuli[54], pharmaceuticals and drug discovery[53,55], environmental sciences, natural product research[56], food sciences to benchmark processing, quality and safety[57]. As it has been demonstrated that cells usually undergo significant metabolic changes in pathological processes such as obesity, diabetes, complex disorders and cancer, metabolic profiling is also used in non-invasive early disease diagnostics, prognostics and biomarker discovery[53,58–61]. After careful examination and testing, many statistical and computational methods developed for gene expression profiles can be directly applied to Metabolomics[50,62].

Cross-Omics Studies. With this rapid development in each individual omics study, and because each omics dimension captures one aspect of the biological state of the cell, research using multiple omics would reveal novel biological hypotheses involving complex interactions among the different omics or pathways that would otherwise be impossible to be mined from a single omics data-type alone, and the results have a wide spectrum of applications in Bioinformatics and Biomedical sciences, such as cross-omics network building[2], modeling of cellular responses[63], regulatory pathways and network reconstruction[64] in plant biology[41], elucidating pathways in microbial systems[40,65,66] and understanding of human diseases[67], and in association with pharmacogenomics for personalized drug therapy[68]. Recently, there have also been proposals of integrating omics data into Electronic Health Record (EHR)s[69,70], such as the high quality personal phenomics (the systematic measurement of the physical and biological traits of organisms) could also become a crucial component for future diagnostics supporting systems for doctors and public health practitioners[71]. Rui Chen et. al proposed a integrative personal omics profile (iPOP) system

that integrates genomic, transcriptomic, proteomic, metabolomics and autoantibody profiles for medical risks assessment[72]. Many Tools have also been developed to provide out-of-the-box solutions from cross-omics research, such as the Bioconductor suite[73], integrOmics[74], MeltDB[75], etc. A comprehensive list of tools can be found in Ghosh et. al[76] It is expected that with the further development of omics such as phenomics, interactomics, etc. and the development in analytical methods and hardware, integrated omics analysis would be even more effective in understanding bioilogical systems.

1.2 Analytical Methods for Omics Datasets

The most frequently asked questions in omics studies, is how do we detect the key molecules, or key pathways and networks that consist of such molecules, that contribute to the observed phenomenon? Different from the pre-omics era when there were only a handful of candidates to manually analyze, now researchers have to mine signals from tens of thousands of genes, proteins or metabolites. More importantly, it is crucial to identify high-level concepts (protein complexes, biological processes, pathways) that contribute to the observations in order to develop hypotheses that link molecular events to high-level biological functions. High performance methods with small false discovery rate (FDR)[77] and that are robust to high levels of noise have become critical. In general, this includes of methods that filter out a small set of molecular features, cluster molecular profiles based on their expression signatures, and map to existing knowledge of interaction networks or creating new networks with observed associations.

For filtering molecular profiles that contain most signals, a Variable Selection (VS) procedure is often used to identify a subset of molecular profiles that best differentiates different sample classes. This can be done using any of the several statistical routines, such as conventional t-test and ANOVA[78], adjusted t test like Significance Analysis of Microarrays (SAM)[79], shrinkage t-test with adjusted variance considering multicollinearity (Shrinkage t)[80,81], a permutation-based approach or constructing new variables using combinations of existing ones to reduce the data dimensionality such as the principle component analysis (PCA)[82], independent component analysis (ICA)[83], non-negative matrix factorization methods (NMF)[84,85], support vector machines (SVM)[86], K-Nearest Neighbors (KNN), Random Forest

coupled Variable Selection (VarSelRF)[87], Discriminant Function Analysis (DFA)[50]. Regression models with variable selection can also be built using Partial Least Squares (PLS) and can then be used for prediction[50]. The filtered subset of molecular profiles carries the most variability of the data and can be used for future classification of samples.

Another approach is to cluster omics variables based on their expression profiles across samples. Clustering is a process that assigns variables to groups while simultaneously minimizing the intra-cluster differences and maximizing inter-cluster differences using a distance function[88]. Because variables that tend to change synchronously are more likely to be involved in the same biological processes, clustering could reveal high-level molecular mechanisms better than the individual omics variables. The most frequently used methods are agglomerative hierarchical clustering (HClust), K-means, Markov clustering, Bayesian clustering, density based clustering, and community structure detection[88–92]. Because some hub genes participate in multiple biological processes, fuzzy clustering and probabilistic clustering methods were also proposed to allow multiple cluster membership assignments[50,88]. Clustering is also an unsupervised method, meaning no prior membership knowledge is used and memberships are only assigned based on distances computed from data.

After obtaining omics profiles, it is then possible to gain a high level view of the individual molecular entities using the interrelated relationships. There are two strategies: the first strategy is to map the identified molecules to known statistically significantly over-represented pathways or networks using an enrichment analysis. This can be done in a slew of offline and online pipelines for single or modular variable discovery, such as Bioconductor/R, Gene Set Enrichment Analysis (GSEA), DAVID, LRpath and ConceptGen, commercial services like MetaCore, etc.[35,62,73,93,94]. On the other hand, it's also possible to reconstruct a de novo network using the Omics profiles, using correlation/coexpression data such as ARACNE[95], WGCNA[96]. An inferred network can also be queried against an interaction network database using network mapping methods such as SAGA or VANLO[97,98]. Integrating several Omics data types could improve the quality of resultant networks and models[99].

1.3 Visual Exploration of Omics Datasets

Complementary to automated pipelines or statistical methods to generate a list of 'research hotspots' is the visual exploration process: the results can be plotted using a variety of static or interactive methods and the researcher then could directly explore the data and look for interesting features. Frequently used visualization methods for Omics data includes Scatter Plots, Profile Plots and HeatMaps[100]. For example, a numeric gene expression profile matrix, with each row mapped to a gene, each column mapped to a sample and cell for the corresponding expression value, can be color-coded and plotted as a heatmap. Aforementioned clustering memberships can also be superimposed, or visualized using schemes proposed by Hibbs et. al[101], which is much easier for human interpretation rather than raw data. Other annotations can be added such as the hierarchical trees or cluster memberships[102]. For direct comparison of multiple Omics data, Baran et. al proposed a color scheme for three way comparisons[103]. OmicsViz can illustrate large omics datasets across several sources[104]. Dimension reduction methods such as PCA, Multi-Dimensional Scaling (MDS) and Non-linear dimensionality reduction algorithm such as t-statistic Stochastic Neighbor Embedding (t-SNE) were also applied to more accurately reflect co-expressed genes in visualizations[105]. Networks and pathways from databases or generated from experiments can be visualized in various ways using standalone software like R, Cytoscape, NAViGaTOR, VisAnt, TranscriptomeBrowser, Pajek, VANTED, GenMAPP, MetaCore and QluCore, or online applications like Oncomine, KEGG, Lichen and Omics Viewer [100,106–108]. Hendrik et al. proposed a slew of visualization recommendations for cross-domain knowledge visualization such as image stacking, network comparison, and multi-modal alignment[109]. Gehlenborg's review compiled a comprehensive list of frequently used software for Omics data visualization[100].

1.4 Bioinformatics Challenges in the Omics Era

With the rapid technological development and decreasing cost, researchers are now less concerned about obtaining raw data. Instead, analyzing and interpreting the exponential growth of data has become the new challenge – even simply handling massive high-

throughput data has become as major challenge to many investigators. I hereby summarized the five major Bioinformatics challenges for the Omics era.

Storage: Although the cost of storage has dropped significantly to about one dime per Gb, there's still an ongoing race between data generation and data storage. Just an example, there are already over one million GEO microarray datasets uploaded to this day. Furthermore, the cost for hardware (servers), service (power, network), and maintenance (labor, backup, transfer, data consistency and security) also stacks up quickly with the growth of data. For example, to install a mirror of UCSC genome browser, it would require about a compute system with about 8 CPU cores, 64Gb memory, and 40 Tb disc storage for a typical setup[vii]. It was estimated that by the end of 2012, the total number of expression datasets has surpassed one million[110]. Even though free public domain data deposit (NCBI GEO[20], UCSC genome center[8], KEGG[47], Reactome and MiMI[34,35]) and low-cost cloud storage (Amazon and Google) have alleviated the economic hardship for small research groups, it remains a big challenge with the rapid instrumental development of high-throughput, high-resolution data generation. Besides hardware, the massive data also incurs challenges to databases. The traditional relational databases such as used for the UCSC genome browser[8] have been competent for manage traditional data, but are less suitable for complex biomedical experiments, free-from and heterogeneous data, media rich data such as images and 3D models[111]. Key-value based big-table implementations such as Google LevelDB, and distributed storage system for structured data such as Big-table[viii], along with adaptive informatics[111], can be utilized to tackle such challenges. However, this brings in other complications such as data synchronization across different physical servers or data centers. Wiesinger et. al compiled a list of crucial questions to be addressed for data and knowledge management for cross-Omics projects[112]. Ghosh et. al reviewd the software tools and resources, as well as workflow designs pertinent to integrated platforms designed for systems biology[76].

[vii] http://genome-source.cse.ucsc.edu/gitweb/?p=kent.git;a=tree;f=src/product

[viii] http://research.google.com/archive/bigtable.html

Query: Assuming the storage demand can be met, the next immediate challenge is how to quickly locate the data of interest. It is now entirely impossible to browse a data warehouse manually and pick entries of interest without queries. Taking full-text search as an example, as of 2012 there are over one million publications indexed in medline[ix]. Pubmed currently indexes over 5600 journals[x] and a Google Scholar search of cancer will return over 3.7 million results[xi]. A search for gene expression profiles with the term MAPK[xii] would return about half million results in the NCBI GEO Profiles database[20]. Even for a research institution, it is impractical to manually explore all these data and because many of the hypotheses are built upon new observations and previous knowledge, information query and retrieval are vital to the knowledge generation process.

Notably, there are two major issues related to data query: first of all, the query syntax must not be over-complicated for non-information-science-savvy people to obtain desired results, yet adequately flexible to tailor customized search needs. The aforementioned PubMed and Google both offer the user simple search that are optimized for day-to-day use and composite search with syntax qualifiers for building complex criteria, or more structured semantic searches. Secondly, the results should be returned relatively fast so that the user may iteratively improve queries based on 'draft' search attempts, but also ordered sufficiently informative and unbiased so that the user wouldn't need to browse several result pages for desired outcome. Researches have shown that most people hardly browse search results appear beyond page one. Such practices may lead to significant biases and consequences in biological research. Development of standardized semantics and structured grammar such as MESH has improved the search efficiency and accuracy[113].

Analysis: The large size of raw data introduced many new issues to the existing data analysis work-flow, both computationally and theoretically. Let's use N to denote the size of data.

[ix] http://dan.corlan.net/cgi-bin/medline-trend?Q=

[x] http://www.nlm.nih.gov/bsd/num_titles.html

[xi] http://scholar.google.com/scholar?hl=en&q=cancer&btnG=&as_sdt=1%2C23&as_sdtp=

[xii] http://www.ncbi.nlm.nih.gov/geoprofiles?term=mapk

For data structures and algorithms that grow even on the order of N^2, personal computers quickly become incapable. For example, the Human Genome U133A 2.0 Array contains 14500 well-characterized human gene profiles[xiii]. If we would like to generate a correlation matrix of all the expression profiles from a single U133A chip alone using 32 bit double, this alone would require around 850 Mb Ram without intermediate storages. More complicated analytical procedures will likely require much more memory and storage; therefore it would be very demanding even to analyze a single piece of big data set, not to mention comparative studies that involve a multiple of datasets. For algorithms that have a time complexity beyond logarithmic growth quickly become impractical to be applied to large datasets. Distributed, cloud computing or using heuristics can dramatically improve performance or the resource requirement in some conditions, but for researchers without such capabilities it is still a tremendous challenge.

Another aspect of related to big data analysis is the theoretical challenges. Obtaining tens of thousands measurements simultaneously indeed boosted the discovery efficiency, but at the cost of less through investigation for individual hypothesis. The potential pitfalls include multiple testing, multivariate outliers, missing values, fuzzy clustering memberships, etc. Taking outliers as an example, Pearson correlation coefficient (PCC) has been widely used in determining the association between two data vectors. It has been shown that even a few outliers, or multi-modal distributed data could significantly affect the resultant correlation[114]. For small datasets, significant candidate correlations can be plotted and visually validated for data peculiarities. However, it would be impossible to manually check all false positive and false negative signals. Even using the multivariate outlier detection methods and robust methods that are outlier resistant, there is still a good chance that a strong signal could be resulted from wrong measurements or noise. Statistical modeling also becomes difficult with tens of thousands variables – even using variable selection methods there could still be thousands of variables left, and they may contribute to a biological process collectively. Moreover, because tens of thousands of hypothesis are tested simultaneously, a considerable number of signals may simply be a result of fluctuations. False-discovery of Omics variables may have a significantly impact on subsequent pathway analyses[115]. Therefore, how to

[xiii] http://www.affymetrix.com/estore/browse/products.jsp?productId=131537#1_1

efficiently and correctly perform pathway analysis to identify the key elements on such big datasets reproducibly are yet to be addressed. Many papers have proposed best practices[116].

Meta analysis, which tests hypotheses from pooled data, has gained very strong popularity in the recent decade. With the vastly available data on the public domain, and the general notion that larger sample size would lead to more statistical power, it is very tempting to conduct research on merged data from different studies, especially different Omics measurements. Many previous researches have demonstrated the efficacy of this strategy. However, there are intricacies that must be addressed for data merging, such as the experiment condition differences, processing of missing values and data normalization.

Exploration: Even though many analysis pipelines, such as sequence assembly, network clustering and pattern matching now can be executed very efficiently without a graphical user interface (GUI), however visual exploration still holds a very important role in Bioinformatics for general researchers. Translating tabulated data into plots, figures and interactive visualizations help data-driven hypothesis generation: the user can quickly identify strong signals that are related to his or her knowledge domain and trigger the intellectual reasoning and discovery process. There is a plethora of publicly available applications, such as Oncomine, Cytoscape, UCSC genome browser, just to name a few[100]. The omics era also imposed significant challenges to the analysts even with the state-of-the-art hardware and software. It is difficult to build high-performance yet interactive visualization that can run on general consumer computer hardware, while feeding the user an overwhelming amount of information could even be counter-productive. For example, Michael B. Eisen demonstrated the effective visualization of clusters of genes that share similar expression profiles in his seminal publication[102]. However, although the heatmap evidently showed overall expression patterns of selected genes, even for this subset that only contained of a few hundred genes and less than a hundred conditions, it already became difficult to examine rows or columns in detail, not to mention if a similar heatmap is plotted for the entire dataset that contains around 8,000 genes. The yeast BIND network from Cytoscape contains over 30,000 proteins and 30,000 interactions, and the visualizations often result in a dense 'hair-ball'[117]. Expanding the SNP track in UCSC genome browser around MAPK9 would reveal all the SNPs in the

region[xiv], which could take several scrolls even on a 27 inch Apple monitor with 2560X1440 resolution to fully display. These visualization techniques are more or less deficient of scalability – at certain datasets of certain size would render the visualization ineffective, which neither reveals much of the underlying network structure nor helping to generate new hypotheses using merely eyeball inspection. To cope with this challenge, the idea of 'multi-scale' visualization was proposed[118]. Employing data clustering hierarchies, or external ontologies, to reduce and visualize raw data at various summarization levels could ease the analytical difficulty. For example, instead of investigating the individual gene expression profile change for tens of thousands of genes, the researcher could first screen the pathways or certain Gene Ontology (GO) terms which contain groups of genes and check whether interesting signals could be detected, and then dive deeper into the details and identify which genes would be the drivers of the signals[119]. A gene interaction network could also first be plotted as high-level pathway modules, and then gradually dwelt down to intermediate clusters and finally lowest level gene-gene pairwise interactions[118]. Many online applications that contain one or more of such exploration methods, such as Oncomine[120], VANTED[107], Qlucore[xv], have been developed to tackle such challenges. However, new 'future-proof', scalable, flexible and versatile visualization models are still yet to be developed.

Exchange: The challenge related to data exchange is two fold. The first difficulty is mapping and conversion of data from different sources. At the inception of the Bioinformatics era, many research institutions developed their own annotation system for genes, transcripts and proteins simultaneously. Because of the lack of standardization (*one of the most important achievement from the first Emperor of China, Qin, is the national standardization of metrics so that people wouldn't have to carry around a dozen of different types money for businesses*), these annotations still coexisted. The Clone[121] offers more than 20 ID conversions and the David gene conversion tool[122] offers more than 40. Some of these ID mappings, such as probe to gene, gene to protein, or gene to transcript, could be one to many, many to one or many to many. Furthermore, many software programs have their own input-output formats – a node-

[xiv] http://genome.ucsc.edu/cgi-bin/hgTracks?position=chr5:179673028-179707608&hgsid=322350387&knownGene=pack&hgFind.matches=uc021yjl.1

[xv] http://www.qlucore.com

edge typed network can be stored as an edgelist, an adjacency matrix, structured markup files (XML, JSON), etc. Different databases also usually have their own data processing and formatting rules. Together these variabilities make a typical cross-omics analysis that involves the analysis of multiple datasets using several software utilities very difficult. A significant amount of labor and resources are usually invested only to pre-process the raw data.

The second difficulty is to exchange data across research institutions and researchers. As the big problems that require expertise from life sciences, statistics, information and computational sciences are unlikely to be analyzed by a single research group alone, how to effectively share data and analysis workload across multiple sites is also critical. Not only terabytes of raw data and accompanying annotations need to be hosted and shared efficiently, but also the shared data need to be interpreted and explored consistently at different sites. This requires each site to have comparable hardware and software setup. Online workspace that allows the researchers to collaborate virtually on data exploration from different geographic sites[123], re-playable workflows such as DAVID[93], EzArray[124], MeRy-B[125], as well as public and cloud storage such as Dropbox (https://www.dropbox.com/) and Peer to peer data sharing such as Tranche, have been developed to address these challenges.

To summarize, the big data era opened doors to many more possibilities – along with the new challenges that call for innovative, disruptive and collaborative solutions.

1.5 My Thesis Work

Through the course of my PhD research, I attempted to probe into several omics datasets, and tried to explore and mine new knowledge. With the aforementioned Bioinformatics challenges in mind, I have been developing tools that could better help understand the structure of networks, identify functional modules in a network, create meaningful visualizations, and applications that offers flexible and efficient search capabilities. I worked on a comprehensive transcriptomics-metabolomics study and proposed novel workflows to identify outliers that could lead to the identification of unknown biological processes and serve as biomarkers, as well as compute reliable correlations for a large number of pairwise comparisons of data with high variability and multivariate outliers. And finally for the highlight of the thesis, and to humbly address the imminent Bioinformatics challenges in this

big data era, I worked with my mentors and advisors to develop a new visual exploration model for large, complex, heterogeneous and structured biological data. While developing CoolMap, I have integrated not only my working experiences with the omics datasets, but also my experience with graphics software, trying to design a new future-proof, extensible and high performance visualization paradigm. I hope this generic exploratory model can potentially be extended to other disciplines and address existing problems in data-driven information mining.

1.6 Organization of this Thesis

Chapter 1 gives an overview of the omics era, the current methodologies and Bioinformatics challenges. Chapter 2 describes the development of a series of Cytoscape plugins for searching, clustering and visualizing omics networks. Chapter 3 presents results of bio-signature discovery using robust statistical methods on cross-omics datasets. Chapter 4 discusses the design, implementation and usage scenarios of the CoolMap paradigm. Chapter 5 concludes the work of this thesis.

1.7 Professional Experience

Expected Publications 2013

Su G *et al*, CoolMap: Multiscale and Multidimensional exploration of omics data

Su G *et al*, Data exploration using CoolMap and Cytoscape: cross-application interoperability through public APIs

Collaborative transcriptomics-metabolomics with Dr Charles Burant's data

Fan GF, Fu CX, **Su G**, Lin G, Pappin D, Lucito R and Tonks NK. PTP1B Dephosphorylates and Antagonizes Brk-mediated IGF-1R Signaling in Ovarian Cancer, in preparation

First Author Publications

Su G, Burant CF, Beecher CW, Athey BD, Meng F. Integrated metabolome and transcriptome analysis of the NCI60 dataset. BMC Bioinformatics. 2011 Feb 15;12 Suppl 1:S36.

Su G, Kuchinsky A, Morris JH, States DJ, Meng F. GLay: community structure analysis of biological networks. Bioinformatics. 2010 Dec 15;26(24):3135-7.

Su G, Athey BD, Meng F. GSearcher: agile attribute querying for biological networks. Bioinformatics. 2010 Dec 15;26(24):3138-9. Epub 2010 Nov 18

Collaborative Work

Qin ZH, Bilenky M, **Su G**, Robertson G, Jones S, MotifOrganizer: A scalable multi-stage model-based clustering approach for grouping conserved non-coding elements in mammalian genomes, Front Biosci (Elite Ed). 2013 Jan 1;5:785-97.

Yan QY, Peng B, **Su G**, Cohan BE, Major T, Meyerhoff ME. Measurement of Tear Glucose Levels with Amperometric Glucose Biosensor/Capillary Tube Configuration, Anal. Chem. Sep 30, 2011.

Morris JH, Apeltsin L, Newman A, Baumbach J, Wittkop T, **Su G**, Bader GD, Ferrin TE. clusterMaker: A Multi-algorithm Clustering Plugin for Cytoscape, BMC bioinformatics 12(1), 436

Zöllner S, **Su G**, Stewart WC, Chen Y, McInnis MG, Burmeister M. Bayesian EM algorithm for scoring polymorphic deletions from SNP data and application to a common CNV on 8q24. Genet Epidemiol. 2009 May;33(4):357-68.

All papers and citations can be found at the Google Scholar profile link:

http://scholar.google.com/citations?user=NWtVOsMAAAAJ

Conference Presentations

Su G, Burant CF, Beecher CW, Athey BD, Meng F. Integrated metabolome and transcriptome analysis of the NCI60 dataset**.** The ninth Asia Pacific Bioinformatics Conference (APBC), Inchon, Korea, Jan 2011

Su G, Kuchisky A, Morris JH, States DJ. GLay plugin for Cytoscape, Cytoscape Annual retreat and Intelligent Systems for Molecular Biology (ISMB) birds of a feather session, Toronto, 2008.

Su G, David J. States. Application community structure algorithms to protein-protein interaction networks. Ohio Collaborative Conference on Bioinformatics (OCCBIO) 2008, Toledo, 2008

Book

Cytoscape Complex Networks Analysis How-to. Packt Publishing Limited. In progress. Expected 2013.

Journal Reviewer

Peer Reviewed Papers for the following journals:

PLOS Genetics	http://www.plosgenetics.org/
BMC Bioinformatics	http://www.biomedcentral.com/bmcbioinformatics/
Journal of Clinical Bioinformatics	http://www.jclinbioinformatics.com/
MDPI Cancers	http://www.mdpi.com/journal/cancers

Chapter 2 Tools for Exploration of Omics Networks

2.1 Introduction

As stated in Chapter 1, the omics era has produced massive amount of raw data that characterize many facets of organisms. One of the biggest challenges is how to make sense of the data for life sciences research groups without strong statistical or computational capabilities. There are two different paths of resolutions: the first is a 'black box' strategy, in which an automated analytical application takes the user input and feed the users human interpretable results. All intermediate steps are encapsulated so that a general user can use the 'typical' routines for most commonly executed tasks. For example, by using Gene Set Enrichment Analysis (GSEA)[94] or online pipelines such as David[93], the user can immediately obtain enriched genes and pathways from the supplied datasets. The advantages and disadvantages of this approach are very clear-cut: while it is efficient to perform 'factory' pipelines automatically and efficiently, it is error-prone if the user does not understand the underlying hypotheses and mechanisms thoroughly. An alternative strategy is the visual-exploration: by plotting the microarray experiment and clustering results as a heatmap, or an inferred protein-protein interaction network using a network view, the researcher could intuitively identify hotspots that could lead to new knowledge discovery. Nevertheless, this method is not as efficient, and due to the human analytical capability and computational / screen estate constraints, the visual exploration quickly becomes ineffective when the heatmap grows more than a few hundred rows and columns or a network contains more than a few hundred nodes. In practice, the two strategies are usually coupled – an Omics dataset is first processed to generate a subset of potential candidates, and each candidate is then visually explored and validated.

Networks are one of the common visualization formats for omics data. A network, or equivalently a graph in mathematics terms, is a set of nodes (interchangeably, vertices) or edges. In life sciences, nodes are usually mapped to biological entities and edges are mapped

to relationships. A network can be directed or undirected; a regulatory pathway can be modeled as a directed network while a pairwise correlation network can take the undirected form. There are many publicly available databases for molecular interaction networks, such as the Biological General Repository for Interaction Datasets (BioGRID)[126], Reactome[34], Michigan Molecular Interaction (MiMI)[35], Biological Interaction Network Database (BIND)[127]. All molecular interaction data can be pooled together to capture the entire interaction map of a cellular organism, and such a pooled interaction network can be named as 'interactome'. A network can be first analyzed using 'black box' pipelines such as the Boost Graph library[xvi] and its parallel variants, JUNG[xvii]and igraph[xviii], and then interactively visualized using Cytoscape, Prefuse, VizAnt or NAViGaTOR[100]. Cytoscape is one of the most popular open source network software for biological network data exploration. One reason for its outstanding success is the plugin framework and strong community support – it is very easy for third party developers to integrate the 'black box' analytical functions into Cytoscape's visualization framework. There were hundreds of plugins developed for network generation, cluster analysis, heatmap visualization, remote database query, ontology analysis, etc. For a complete list of current Cytoscape plugins, please refer to the list here[xix].

There are several questions researchers would like to ask when exploring a biological network. First of all, what's the overall structure of the network? Does it consist of a single fully connected component, in which one node can reach any other node via a limited number of steps, or many small disconnected cliques? Is the network very dense with a large number of edges, or relatively sparse with only a few 'hub' nodes? Secondly, is it possible to partition the network into smaller clusters, or communities, so that each cluster contains densely interconnected nodes? The clusters could contain nodes share similar attributes or frequently interact with each other, which implies certain biological functions. Thirdly, given a very large visualized network, how can one find nodes that suit some complex criteria? For

[xvi] http://www.boost.org/doc/libs/1_52_0/libs/graph/doc/

[xvii] http://jung.sourceforge.net/

[xviii] http://igraph.sourceforge.net/

[xix] http://apps.cytoscape.org/apps/

example, in a molecular interaction network, we would like to find all the nodes that are annotated with a specific Gene Ontology term such as 'apoptosis'; or in a gene regulatory network coupled with microarray data, we would like to highlight all the differential-expressed genes. Corresponding to Chapter one, we need solutions for visualization, analysis and query challenges.

When I began to use Cytoscape in the summer of 2008, there were not many tools available to address these challenges. Existing ones generally had difficulty with increasingly bigger and more complex data. I noticed there were quite a few 'black box' analysis libraries available, and would notably address the aforementioned questions if added to Cytoscape. Developer rule number one: don't reinvent the wheel.

2.2 GLay and igraph

The GLay project was initially developed as a student project in Google Summer of Code 2008 (GSoC) and was presented in the Intelligent Systems for Molecular Biology conference (ISMB) 2008. The project was continuously developed afterwards with a rich set of community structure detection algorithms and network layout functions, with the hope that it would be more useful for general network data analysis.

2.2.1 Community Structure and igraph

Many real world networks, including social network, biological interaction network, citation network, etc., are scale-free networks in which a few 'hub' nodes contain a very large number of edges, and the rest of the nodes have very few edges. The overall degree (number of edges a node) distribution asymptotically follows the power law. The overall effect of a scale free network is that nodes tend to cluster into communities. It is then possible to partition the network into communities and elucidate their functions respectively. Namely, using the divide and conquer strategy.

Community Structure Detection is a popular way to partition a network. A network can be subdivided into communities without using attribute data to compute distance matrices similar to the Markov Clustering (clusterMaker); the resultant communities maximizes intra-community connections and minimizes inter-community connections. These algorithms

have been applied to social sciences to identify groups that share similar interests or behaviors (Mark Newman). Many high-performance and scalable community detection algorithms have been applied to molecular interaction networks to reveal functional modules. It has been demonstrated that some of these algorithms are capable of clustering mega-scale social networks. Usually a community structure algorithm uses the modularity score Q, proposed by Mark Newman[128], to benchmark the quality of a community structure. This score reflects the quality of the computed community structure with reference to a reference random Erdos-Renyi network, in which the probability of having an edge connecting an arbitrary pair of nodes is uniform. The modularity takes value between 0 and 1, for 0 equivalent to a community structure computed on a comparable Erdos-Renyi network and 1 for perfect community structure. Some researchers have pointed out that using only the modularity score may lead to resolution issues: the number of communities found does not scale well with the size of the network and some large communities may also possess strong community structure. Iterative methods, quality functions other than modularity score and module checking criteria have been proposed to improve the quality of community detection algorithms[129,130].

The igraph(http://igraph.sourceforge.net/) is a C based library for comprehensive graph analysis[131]. It has interfaces to many major programming environments such as R, Python and Ruby. This library provides a rich collection of community detection algorithms and a variety of layout algorithms for very large networks.

2.2.2 Existing Network Clustering and Layout Functions in Cytoscape

Cytoscape is not preinstalled with any network clustering functions and all clustering functions are contributed from the community. There are several frequently used Cytoscape plugins developed for functional module detection available, such as MCode[132], NeMo[133] and ClusterMaker[92]. However, there are some limitations to these applications. Some algorithms in ClusterMaker, such as hierarchical or k-means clustering, require additional numerical attributes to compute a dissimilarity matrix. Others, such as MCode and NeMo, are only engineered to find small and densely intra-connected local clusters without clustering all the nodes in a network network. For example, when applied to a network from Michigan Molecular Interaction (MiMI) containing 11884 nodes and 88134 edges with default

parameters, MCODE produced 105 clusters, in which 52 clusters contain less than five nodes. Therefore, it may not be suitable for global partioning of large networks for exploratory analysis. In addition, some of these plugins were not tailored for large networks NeMo threw an error when executed on the same network on a 2.67GHz Intel Core i7 machine. Before GLay, no plugin offered a comprehensive collection of efficient community detection algorithms, which could profoundly improve cluster analysis for Cytoscape.

Besides network clustering, visualizing very large networks is also a big challenge for Cytoscape. It is very desirable to plot the network in such a way that the node proximity reflects the network topology. Force based layout algorithms, using ball-string model simulation, are frequently used to generate layouts on scale-free networks. However, generating a force-based layout on a large network not only consumes considerable resources and time, but also rarely produces any informative outcome. It is suggested that direct application of a force-based algorithm on a dense network with greater than 500 nodes could produce a massive hairball-like mess[134]. Therefore some additional procedures, such as brushing and visual attribute mapping may be necessary. Cytoscape is shipped with several layout functions, with the organic layout and force-direct layouts as most popular ones. However, the bundled Force-based layouts can fail when the network size grow beyond 10,000 nodes using default parameters. Therefore, adding scalable and versatile layout algorithms, along with the community detection functions would significantly improve the visual analytic capability of Cytoscape on large Omics datasets.

2.2.3 Implementation of GLay

The core of GLay was developed as a Cytoscape plugin with high-performance community analysis and graph layout functions ported from the igraph C library[131]. The bridging was constructed via Java native access[xx] interface. JNA serves as a high level wrapper to encapsulate the complexities of the Java Native Interface[xxi]. Functions can be written in prototype format in Java and they are mapped automatically to the native code declarations. Also JNA supports native object mapping that allows the direct coupling of Java objects and

[xx] JNA https://jna.dev.java.net

[xxi] JNI http://en.wikipedia.org/wiki/Java_Native_Interface

native objects (such as primitives and arrays) and thus facilitate efficient data transfer. The functions ported from igraph C library were only compiled under Windows 32/64 bit platform, but could be easily extended to other platforms by recompiling and C codes and configuring JNA accordingly. A Google Summer of Code 2011 project managed to port some igraph functions to Cytoscape under Mac OSX[xxii].

The general workflow is done as follows: when user issues a command to use GLay-igraph for community structure detection or network layout, the current network is automatically processed as the input network, with edge directionality, duplication and self-looping removed. Such a network standardization step will make the resultant community structures from different community structure detection algorithms comparable as well as improving the overall performance, as empirically, edge directionality generally don't affect the outcome of community structure. Upon completion of an analysis, the community memberships are sent back to Cytoscape and a custom layout is created to color-code the node membership assignments, along with a custom node attribute for memberships. The user may browse the resultant community structure with the built-in GLay navigator panel.

Currently GLay has implemented the following community structure detection algorithms: Edge-betweenness[89], Fast-greedy[135,136], Label propagation[137], Leading eigenvector[138], Spin glass[139] and Walk trap[140]. Because of the distinct heuristics of algorithms, scalability, running speed and the resultant community structures vary. Some algorithms, such as the leading eigenvector algorithm, works well on a small network of a few hundred nodes but may not be suitable for very large networks. Others are optimized for large datasets but may be less accurate. For example, the fast greedy algorithm may produce communities with skewed community size distribution because of the greedy optimization of the modularity score[135]. Users may test different algorithms and evaluate performance by various benchmarks such as modularity, number of communities and community size distributions. Some empirical guidelines are provided to help the user determine the optimal algorithm for their specific dataset in the following sections.

[xxii] http://code.google.com/p/google-summer-of-code-2011-nrnb/

For layout algorithms, GLay offers the following: Fruchterman Reiingold[141], graphopt (http://www.schmuhl.org/graphopt/), Kamada Kawai (force based spring layout)[142], Large Graph Layout (LGL)[143], Multidimensional scaling (MDS)[144], Reingold Tilford (Hierarchical and Circular)[145]. These algorithms are capable of efficiently layout very large networks or generate characteristic views such as hierarchical or circular trees. A key advantage of GLay layout is that it allows the layout calculations of various algorithms to initiate from the current network layout state. This adds significant flexibility since it enables the user to progressively refine the layout by either fun-tuning parameters or using different layout algorithms together. For example, when attempting to calculate a force-based layout for a very large network, the user may specify a small number of iterations to obtain a draft layout, and then gradually refine this layout b adding more iteration, tuning the parameters or even combing multiple layout algorithms together. Once done, the user may super impose a community or partitioning structure on the resultant layout to investigate network topology. We have supplemental information on the plugin homepage and igraph library documentation[131].

2.2.4 GLay Usage Cases

We have tested GLay on datasets of various scale and structure. GLay demonstrated substantial performance leverage in both network decomposition and layout generation over existing Cytoscape solutions. For example, using GLay to subdivide the MiMI human Interactome that contains 11884 nodes and 88134 edges takes 0.7 seconds using the label propagation algorithm and 20s using the fast greedy algorithm on a Intel Core i7 machine with 2.67GHz CPU clock. MCODE takes around 198 seconds to find functional modules (clusters). Generating a layout on this network using the Fruchterman Reingold grid algorithm takes about 20 seconds, where as the Cytoscape built-in force directed and spring embedded algorithms both reported failure during execution with default configurations and 1.5G heap space. This demonstrates that Java-C hybrid model has dramatic performance advantage processing large network works in Cytoscape. This hybrid model adds additional cost of maintaining multiple code-bases across platforms, but it can take advantage of the high performance statistics, analysis and numerical processing libraries.

Figure 2-1 shows using Fruchterman Reingold grid layout on the Cytoscape built-in BIND human dataset, consists of 17961 nodes and 30156 edges. It demonstrates the effectiveness of superimposing community structure on top of a force-based layout. The red circle indicates a group of highly interacting immunoglobulins which don't have strong interactions with other proteins.

Users may navigate and explore communities of genes with the GLay browser. For example, clicking the community entry in the community browser table will select all nodes belonging to that community. The user can then create a new subnetwork or nested network (metanodes) from the selected nodes, extract gene lists from the attribute browser or incorporate other experimental data for various research interests.

In addition, GLay can provide qualitatively different results from existing solutions. Figure 2-2 shows a side-by-side comparison of MCODE using default parameters and GLay using the fast greedy algorithm. It can be seen that by using the default parameters, MCODE produces much smaller clusters than GLay, leaving the majority of the nodes un-clustered. Therefore, GLay outperforms MCODE in terms of structural partitioning of the original network. In addition, overall GLay has higher sensitivity than MCODE at the trade-off of specificity, which made it more suitable for functional interpretation. For example, one cluster in MCODE contains five genes, with four genes function in the MAPK pathway. The equivalent GLay cluster contains 25 genes. Submitting these genes to DAVID[93] reveals one enriched functional cluster for the MCODE cluster and nine enriched functional cluster for the GLay cluster. As some of the genes such as *cdc28* and *ste12* are involved in multiple regulation processes, the GLay cluster recovered more biological-relevant information than the equivalent MCODE cluster. As GLay contains a variety of community detection algorithms, the user may also evaluate the performance of these algorithms using prior knowledge from their datasets and the inferred functional modules.

In summary, GLay capitalizes on the power of highly optimized C code from several social network analysis and network layout algorithms to improve scalability of Cytoscape for large networks. GLay helped address the increasing needs for analysis and visualization of large-scale networks. The proposed novel Java-C hybrid model implementation could also benefit the Cytoscape plugin developer community.

2.3 Comparison of Different Community Structures

As many community detection algorithms have been proposed, there has been very little work done to evaluate the actual performance of these algorithms. Most of these algorithms rely on the optimization of the modularity score proposed by Mark Newman[89]. However, sometimes this single measure is insufficient to judge the resultant community structures as has been demonstrated[129,130]. There has been previous research on evaluation of the community structure algorithms[146]. We also therefore ventured into this problem and proposed some empirical evaluation guidelines.

As mentioned in previous sections, the performance of community structure algorithms relies heavily on the topology of the input data and the algorithmic heuristics. Although a community detection algorithm could generate a partition on any input networks, networks lacking the small world feature (such as an Erdos-Renyi random network in which the probability of having an edge connecting any pair of nodes is equal) will not produce meaningful community structures. Also occasionally the resultant clusters may only arise as artifacts of the algorithms[135]. The challenge is that there is no good 'golden-standard' to benchmark the algorithm performances – the frequently used Karate club network is far too small for today's practical uses. It is then usually reliant to the expertise, knowledge and discretion of the analyzer to determine which is the optimal algorithm for a given dataset[129].

2.3.1 Materials and Methods

A Boolean rat interaction network (rat interactome) was constructed from MiMI protein-protein interaction data, with edge duplication and directionality removed. The largest connected component of this network contains 3664 genes and 39150 interactions. This interactome data demonstrates the typical small-world property of nicely fitted power-law node degree distribution (with the fitted alpha = 1.608).

We analyzed the following community structure detection algorithms used in GLay from the igraph package: leading-eigenvector (LE), spin-glass (SG), walk-trap (WT), fast-greedy (FG) and label-propagation (LP). The Clauset, Newman and Moore – Wakita and Tsurumi (CNM-WT) algorithm was also evaluated using the executables from the authors. We

proposed a set of rules including modularity, scalability, reproducibility, robustness and interpretability as quality measures, which will be discussed in detail in the Results section.

The performances of the algorithms were also assessed by mapping communities to biological annotation data using standard enrichment analysis. We calculated the total number of enriched KEGG pathways and GO Biological Process (BP) terms using Fisher's exact test with R SubpathwayMiner package and topGO package, respectively. All parameters are set using package recommended values. The cut-off for KEGG pathway enrichment is FDR < 0.05. The cut-off for GO enrichment is p-value <0.01. Gene ids are also permuted for 100 times to compute the mean and standard deviation of total number of matches for the null distribution. All analysis was done on Depression Centre Cluster from University of Michigan, Ann Arbor.

2.3.2 Algorithm Evaluation Criteria

To choose the optimal community detection method for our dataset, we proposed the following criteria for performance evaluation. Table 2-1 summarizes the comparison of these algorithms.

- **Modularity**. The algorithm must be capable of producing a community structure significantly different from random partitioning, which is measured by modularity score[89]. Most algorithms can perform reasonably well on our dataset because the topology of rat Interactome (such as power-law fit of degree distribution) is significantly different from a random Erdos-Renyi network, which has expected modularity score of 0. We also compared the agreement of the community structure produced by different algorithms. Result showed that even though the differences in modularity scores are small (Table 2-1, modularity ranges from 0.65 – 0.74), the community structures are very dissimilar. For example, the modularity score is almost identical (0.71) for EB and LP, but the agreement between the two resultant community structures, measured by Adjusted Rand Index (ARI, ranges from 0-1 with 1 for perfect match and 0 for comparison of two different random partitions)[147], is only 0.15. This phenomenon demonstrates that optimization using modularity score is not sufficient as there may be a large number of dissimilar solutions that produce similar modularity score.

- **Scalability**. From the summary of running time we can see that even though these algorithms produce community structure with very close modularity scores, the running time vary dramatically. Algorithm such as LP, which is capable of partitioning a network in near linear time, takes less than 1 second to finish whereas a single run of SG takes more than 10 minutes. The original Edge betweenness algorithm requires more than 8 hours of running time. As the size and complexity of biological networks grow rapidly, scalable algorithms such as FG, KWT, LP and WT are more favorable to do quick and draft partition of very large networks.
- **Reproducibility**. Some non-deterministic algorithms, such as SG and LP, produce different community structure in each execution because of its stochastic nature, even though these community structures produce similar modularity scores. The researcher will need to run the algorithm repeatedly to find the best solution or generate a 'mean' community. We estimated the dispersion[148] of SG and LP, which is a score between 0 and 1 representing how often two nodes are consistently in the same community after repeated runs. The dispersion for SG is 0.82 and 0.93 LP after 100 repeated runs, which demonstrates that communities from LP are much more reproducible thus easier to replicate.
- **Robustness**. As the real world dataset is always incomplete and error prone, the algorithm must have sufficient robustness to cope with noise. To evaluate the robustness of algorithms, the rat Interactome was perturbed by randomly adding (ad), removing (rm) or rewiring (rw: edges were randomly broken down and reconnected while maintaining the degree distribution of the original network) a certain number of edges. As we don't have a prior probabilistic distribution of edge error rates, we assume its uniform in entire Interactome and perturbed all edges at equal probability. The interrogated network is then analyzed with the same procedure and the produced community structure is compared with that produced from the original dataset using ARI. The network is perturbed 10 times for any given number of edges subject to change and the average ARI and dispersion score is computed. Figure 2-3 shows the 'deterioration curve' of ARI and dispersion score generated by 6 algorithms on the perturbed networks. It can be seen that ARI and dispersion for FG decreases rapidly even with only a small number of perturbations because of FG's greedy optimization of modularity score. KWT as an improved version of FG

with cluster size balancing did show better performance dispersion score, which implies it's more resistant to noises. LP and SG bear no signs of 'deterioration' because of their stochastic nature, and consistent with previous analysis in reproducibility, LP outperforms SG as shown both in ARI and dispersion curves. WT outperforms all the other algorithms in both ARI and dispersion. It is capable of producing very stable communities with about 1% of the edges in the network perturbed.

- **Interpretability**. It has been shown that some algorithms such as FG, despite of their outstanding scalability, tend to produce community structure with artefacts because of the heuristics[135]. Some of these artefacts don't undermine modularity score severely but could make the results difficult to interpret. We evaluated the 'balance' of the community size distribution in skewness and kurtosis, as shown in table 1. Large value of skewness and kurtosis indicate unbalanced community size distribution. For example, FG, LP, FG and WT tend to produce communities with skewed size distribution, with a large number of singletons and very small communities and several very large communities. WT produces 541 communities with 438 containing less than 5 nodes, which could also inflate ARI, dispersion and enrichment analysis. It is also not suitable for 'draft' partitioning of a large network for further investigation. KWT out-performs all other algorithms as it's designed to account for unbalanced community size distribution.

From these analysis we can see that the choice of algorithm is data and research interest dependent, and we can rank each algorithm based these proposed criteria. For very large networks, scalability is the utmost concern and FG, KWT , LP and WT are good choices. If the researcher prefers to do a rough cut of the network with relatively a small number of communities, FG, KWT and SG are favorable. If the network is with high variability (such as a correlation network generated from a gene expression dataset), then a robust algorithm such as WT, or non-deterministic algorithm such as LP and SG will work better. In addition, the researcher could pick the algorithm, which produces approximately desired number of communities if this prior knowledge is available; or the size of communities with known genes or proteins approximate those annotated pathway or network modules.

2.3.3 Algorithm Evaluation and Biological Inferences

We performed enrichment analysis of communities from FG, KWT, LE, LP, SG and WT, using KEGG pathways and GO Biological Process (BP) terms. The total number of enriched pathways and terms are counted for the result of each algorithm. For LP and SG the analysis was repeated 50 times to compute the mean and standard deviation. In order to assess significance, gene ids were permuted to generate a random network and the communities were analyzed using the same procedure.

It can be seen from Table 2-2 that communities from each algorithm is capable of capturing a much larger number of enriched KEGG pathways and GO BP terms than the permuted network. Some algorithms such as LP and WT, captured much more enriched pathway and GO terms. This could be resulted from the larger number and smaller communities they produce, as the permuted communities from LP and WT also captured more communities than the other algorithms. In this case, even though the enrichment is very high, it's very difficult to interpret therefore other algorithms such as KWT may be favored. The performance of FG, KWT, LE and SG are then very close in terms of biological inference. Neverthless, partitioning the interaction network would reveal densely interacting modules for further functional interpretation.

2.4 GSearcher

A biological network, in which genes and proteins are modeled as nodes and interactions are represented as edges, can be associated with various attribute data such as gene annotation or expression profiles. As stated earlier, one challenge for network exploration is effective and efficient searching. As the size of networks and amount of attribute data accumulate, highly flexible and scalable search solutions become increasingly necessary.

Cytoscape has a lightweight but powerful tabulated attribute data structure to store annotation data that are linked with the loaded networks. It provides some internal functions for searching, such as Quick Find and Filters. However, this is often insufficient for the user to quickly filter for a subset of nodes based on a given criteria. Cytoscape Enhanced Search Plugin (ESP) has been developed to incorporate more search capabilities using the Java Lucene text search engine functions[149]. The added fuzziness dramatically improved the

search flexibility and accuracy. However, the major drawback of these search solutions is that they still use the conventional (or legacy) submission-wait mechanism. The user typically supplies a query, hit the 'Enter' key to begin the search and then waits for the results to be shown in the default attribute browser. The submit-wait process must be repeated to compare different queries, to correct errors and to progressively improve a query. This process not only creates the perception of search slowness by forcing the user to wait for complete results from unsatisfactory preliminary searches, but also interrupts any coherent thought processes. Many modern search engines, such as iTunes and Google search, updates the search result instantly from the users' input without waiting for the user to notify the server by the 'Enter' key. This interactive auto-completion and live-feedback model enables the user to complete a query from live feedback, dramatically improving the efficiency of searches and the aesthetic appeal of interaction with the software.

Current Cytoscape search tools also have some issues that undermine the efficiency and accuracy of searching. First of all, all these tools use the default attribute browser to display results. This may result in a user interface function collision with other plugins (such as MCODE) or functions that also use the default attribute browser to display data. Secondly, these tools only select nodes or edges matching the search criteria, without indicating where in the attribute table matches occur. The user can only locate the position of matches by manual scanning, which can be very difficult for browsing the result from a fuzzy search in a very large attribute table with many columns. Thirdly, it is difficult for the user to compare matching and non-matching attributes as non-matching attributes are always hidden by current tools. Comparing matching and non-matching entries would also help the user refine the search criteria. Finally, all these search methods only support a subset of fuzzy matching rules. For example, wildcards currently are not allowed at the beginning of a query, which prevents the user from suffix-initiated searching. GSearcher addresses these issues by providing a fully interactive, highly flexible search interfaced that supports full Java regular expression (regex).

2.4.1 Design and Implementation

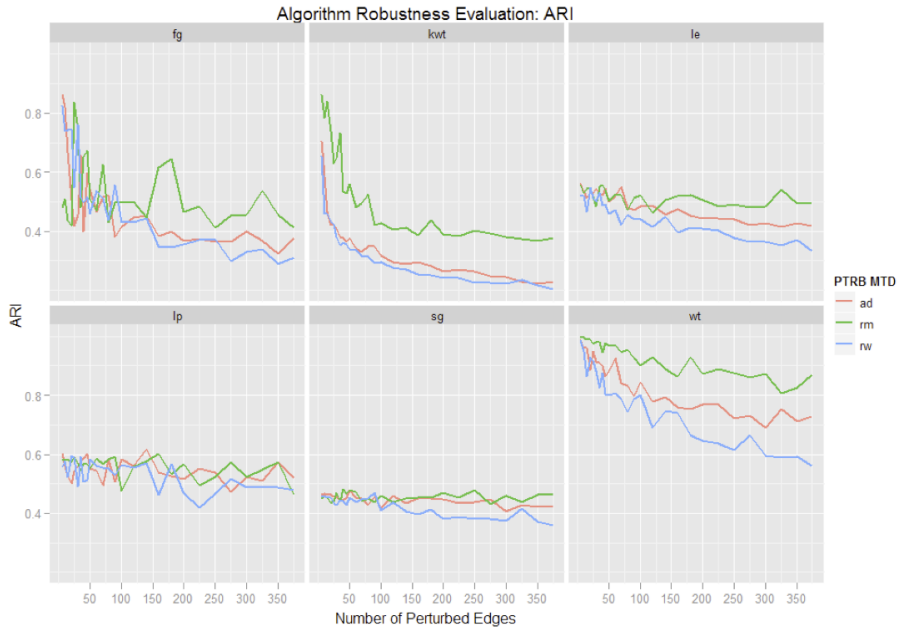
GSearcher is built on JDK 6. We used GlazedList[xxiii] library as the underlying data model. Upon initialization, the current network's attribute data are transformed into a specific GlazedList table model for high performance searching, sorting and updating. This transformation is very efficient; GSearcher only takes 792ms to transform a Michigan Molecular Interaction (MiMI)[35] human interaction network of 11884 nodes and 88134 edges with 21 attribute fields on a 2.67 GHz Intel Core i7 920 PC. In comparison, ESP takes about 4 s for indexing on the same computer. The numerical primitive data types (Double, Float, Integer) are preserved; Hash/Array attributes are flattened into Strings. Subsequent searching on the same network does not require table model reloading (re-indexing).

In order to provide interactive feedback and result sorting, browsing and highlighting, we built GSearcher's independent data browser using JXTable[xxiv]. This browser listens to user input and updates search results interactively independent from the default data browser. Attribute entries that match the query are highlighted in the table. The user may either remove non-matching attribute rows from the browser, or keep them in the browser to compare with the matching rows. Similar to the default browser, the selected rows in the result table are dynamically linked to the network view, but now the user may either 'select' or 'highlight' nodes or edges. When the nodes/edges are highlighted, the selection state is preserved so that the search can be performed independently with minimal interference to other Cytoscape functions (Figure 2-3 Robustness test on community algorithms.

These two figures demonstrate the robustness of algorithms when the rat Interactome network is perturbed with certain number of edges added (ad), removed (rm) or rewired (rw), measured in ARI (the agreement with result from original communities) and dispersion (the reliability of community structure).

[xxiii] http://publicobject.com/glazedlists

[xxiv] http://swinglabs.org/


Algorithm Robustness Evaluation: ARI
fg
kwt
le
lp
sg
wt
ARI
0.8
0.6
0.4
50 100 150 200 250 300 350
Number of Perturbed Edges
PTRB MTD
ad
rm
rw

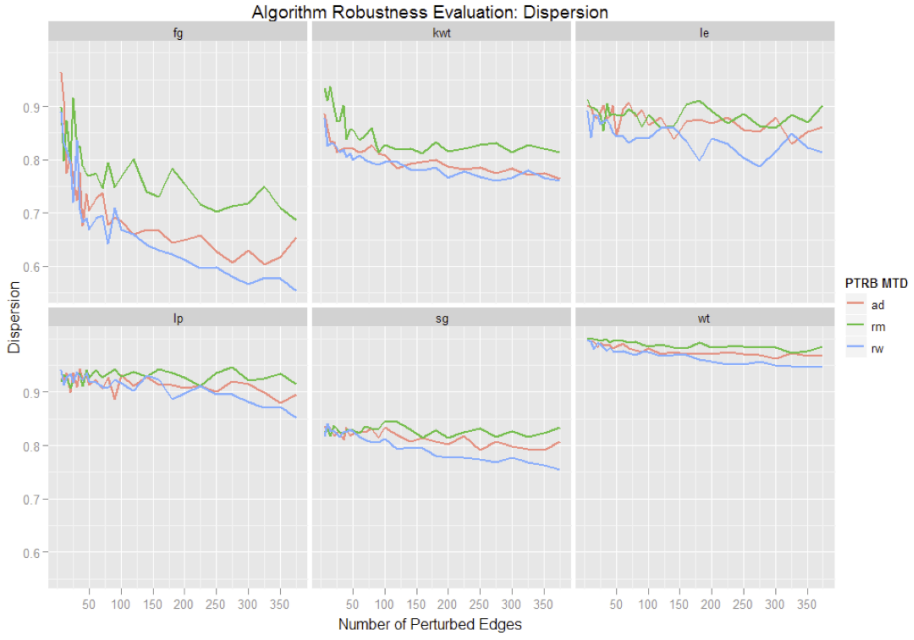

Algorithm Robustness Evaluation: Dispersion
fg
kwt
le
lp
sg
wt
Dispersion
0.9
0.8
0.7
0.6
50 100 150 200 250 300 350
Number of Perturbed Edges
PTRB MTD
ad
rm
rw

Figure 2-4). Rows in the browser can be sorted by a single click on the column header, either numerically or alphabetically depending on the primitive data types. Undesired attributes can be removed from the view and the search pool by hiding the browser table columns.

2.4.2 Search Capabilities

Basic Search: Quick search mode enables the user to apply a single query on all or selected attribute fields. There are currently six different matching modes:

- Terms Anywhere (M): allows a match to occur anywhere in the attribute table. Multiple keywords can be submitted separated by spaces. Unlike ESP, the default operator for joining multiple terms is AND. This is more similar to typical online searching.
- Begins with Phrase: only matches a phrase at the beginning of attributes.
- Reg Exp: the query term is treated as a JAVA regular expression instance.
- Exact: the query term must match an attribute perfectly.
- Phrase: the query is treated as a phrase with spaces preserved.
- Exclude Phrase: the query is treated as a phrase, and attributes that do not contain the phrase are highlighted.

GSearcher provided some search functions that were unavailable for Cytoscape for version 2.6.x or lower. For example, querying 'nuclease' on the MiMI network using 'Terms Anywhere (M)' returns 116 matches, while ESP only returns 13 matches. 'endonuclease' and 'ribonuclease' were left out by ESP because suffix matching is not allowed. Using regular expression, the user can build even more flexible rules. Using 'CDC\d+' as a regular expression query will only match attributes beginning with 'CDC', followed by a number, such as 'CDC16'. The ESP syntax only allows 'CDC*'in which the wildcard cannot be refined to represent a set of characters. 'biological\?_\s+function' will match not only 'biological_function' and 'biological-function', but also 'biological function'—which allows fuzzy matching to span over spaces. '(?!ATP)binding.*' will match a binding term NOT following ATP, such as 'RNA binding'. Therefore, by incorporating regular expression,

GSearcher substantially supplements current Cytoscape search functions. A comprehensive reference of regular expressions can be found at the Sun Java tutorial[xxv].

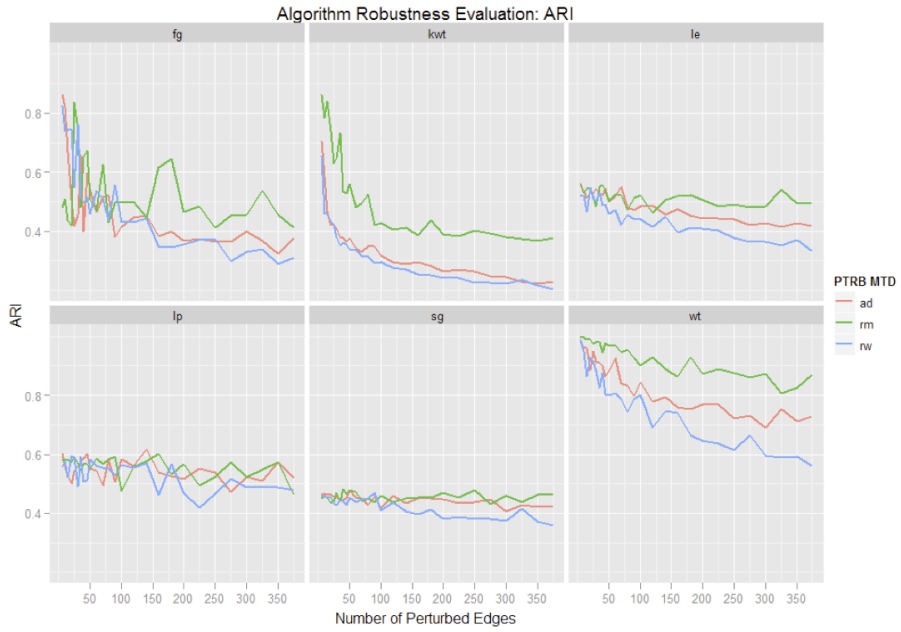
Advanced Search: While Quick search applies the same search criteria to one or multiple attributes, the 'Advanced search' combines an arbitrary number of Quick Search filters that can be casted on different attributes. Filters can be joined either with AND, which indicates all filters must be satisfied, or OR, where at least one filter is satisfied. There are currently three types of filters:

- Text filter: each text filter is one implementation of Quick Search.
- Threshold filter: compare a numerical attribute with a certain threshold value, using numerical comparison operators (such as >).
- Range filter: test whether a numerical attribute is within the specific range.

The combination of filters offers users great flexibility when querying the network. Figure 2-3 Robustness test on community algorithms.

These two figures demonstrate the robustness of algorithms when the rat Interactome network is perturbed with certain number of edges added (ad), removed (rm) or rewired (rw), measured in ARI (the agreement with result from original communities) and dispersion (the reliability of community structure).

[xxv] http://docs.oracle.com/javase/tutorial/essential/regex/


Algorithm Robustness Evaluation: ARI
fg
kwt
le
lp
sg
wt
ARI
0.8
0.6
0.4
50 100 150 200 250 300 350
Number of Perturbed Edges
PTRB MTD
ad
rm
rw

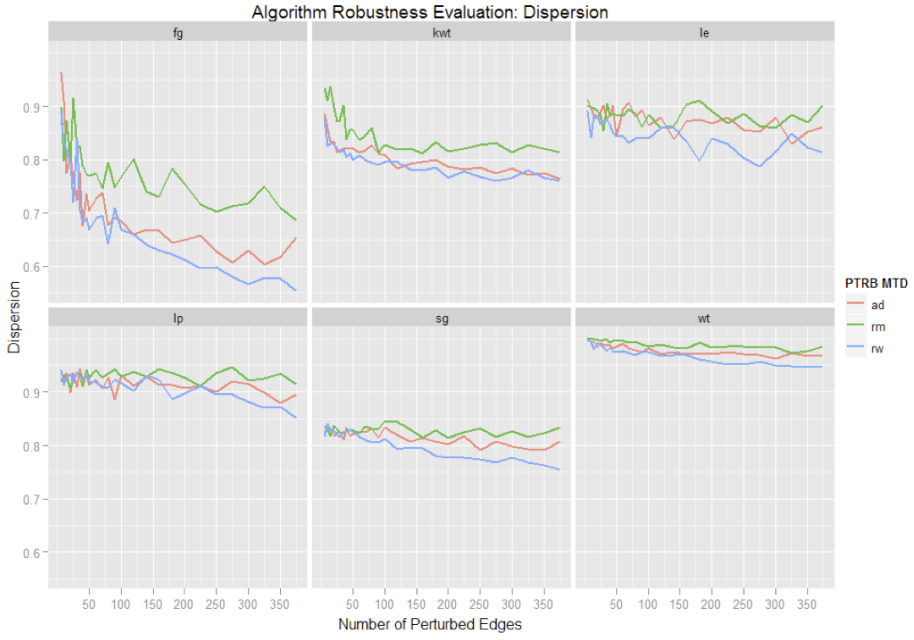

Algorithm Robustness Evaluation: Dispersion
fg
kwt
le
lp
sg
wt
Dispersion
0.9
0.8
0.7
0.6
50 100 150 200 250 300 350
Number of Perturbed Edges
PTRB MTD
ad
rm
rw

Figure 2-4 call-out box also illustrates an example of incorporating a threshold filter and a regular expression filter.

To conclude, the GSearcher provides Cytoscape users with an interactive interface and full regular expression support for building complex queries. It has demonstrated the improved flexibility and interactivity that supplement the current Cytoscape search functions, help researchers navigate large attribute datasets and facilitate exploratory analysis of biological networks.

2.5 Other Related Tools

I have also developed and contributed to some other tools for network exploration. My Java implementation of fast-greedy community detection algorithm and its derivatives balanced for community size distribution was incorporated to clusterMaker[92]. I co-mentored a Google Summer of Code project (GSoC) in 2009 for a phylogenetic tree layout plugin for Cytoscape[xxvi], and mentored a project in GSoC 2011 for a Cytoscape igraph plugin (similar to GLay) on Mac OSX in 2011[xxvii].

Another plugin I developed is the NodeFilter for network expansion and traversal. In the previous section, we discussed the application of attribute-based filtering of Cytoscape networks. As the networks grow larger, a very common task is to select a subset of nodes from a certain criteria to pinpoint the nodes or edges of interest. On one hand, in certain scenarios, it is necessary to filter nodes based on the structure of the network: for example, a researcher may need to find all the first or second neighbors of a certain node, or extend the current network by fetching all nodes connected with a certain node of interest from external databases such as MiMI. One particular usage of NodeFilter (Figure 2-5) is to selectively hide the 'hub' nodes in a scale free network if they become obtrusive to network exploration. The node filter provides a series of tools for the user to conveniently filter nodes based on their connectivity with other nodes. After an initial selection, the user is able

[xxvi] http://apps.cytoscape.org/apps/with_author/Chinmoy%20Bhatiya

[xxvii] http://apps.cytoscape.org/apps/igraphplugin

to show only the first neighbors of a certain set of nodes. This set of nodes is called 'anchors', and they will not be visible in the subsequent operations unless the user releases them. The user then can progressively unhide other nodes to extend the current network, or fetch data remotely to grow the current network from the MBNI Brainarray concept mapping.

2.6 Summary

Thanks to the continuous development of open-source software like Cytoscape and strong community support, visual exploration of omics networks has remarkably facilitated data-driven hypotheses generation. Not only the structure of the networks can be clearly visualized, but also a variety of attribute data, such as Gene Ontology (GO), gene family, pathways and experimental measurement such as gene expression profiles from cross-omics studies or time-series experiments can be super-imposed on a network visualization. Nevertheless, the on-going Omics era will generate even bigger, more complex data with even more dimensions. Tools like GLay and GSearcher have demonstrated the effectiveness of facilitating the clustering and filtering of big biological networks.

One challenge yet to be addressed is the visualization model. With the expansion of data, simply adding data points into the current view, such as increasing the number of nodes or edges of a network or the number of rows or columns in a heatmap will eventually run out of screen estate and resources. As stated in Chapter one, one strategy is to add hierarchies to the data view. By the time of this thesis, the latest release of Cytoscape (2.8.x and the 3.x beta) both permit nested networks – a network can be nested within a node to generate a 'network of network'. It is especially useful, for example, to collapse genes that share the same GO term into a single node that represents a functional group. Similar idea can be extended to other visual exploration models, which will be discussed in later chapters.

Table 2-1 Community Algorithm performance comparison.

Each algorithm is performed with 50 repeated runs. The standard deviations of modularity, skewness, kurtosis and the number of communities from non-deterministic algorithms (LP and SG) are also provided.

Algorithm	Running Time (Secs)	Modularity	Skewness	Kurtosis	Number of Communities
FG	2.88±0.16	0.71	3.16	10.68	38
KWT	0.17±0.06	0.66	0.50	-1.10	23
LE	20.81±0.07	0.65	2.76	8.16	56
LP	0.14±0.06	0.71±0.01	7.18±1.72	64.6±28.38	175±8.75
SG	739.23±65.5	0.74±0.13	2.94±0.53	9.49±3.09	24±0.80
WT	2.50±0.09	0.70	9.45	106.45	541

Table 2-2 KEGG pathway and GO biological process (BP) enrichment for communities.

Gene ids are permuted 100 times for each algorithm to obtain the average total. Numbers show the enriched KEGG pathways and GO terms for the null distribution.

Algorithm	Total enriched KEGG pathways (Original)	Total enriched KEGG pathways (Permuted)	Total enriched GO terms (Original)	Total enriched GO terms (Permuted)
FG	139	3.76±4.05	1317	157.78±36.79
KWT	141	3.02±2.48	1321	140.3±29.19
LE	137	3.25±3.16	1310	222.24±39.71
LP	292±26	17.10±9.57	4015±232	580.49±65.71
SG	166±9	3.68±4.24	1544±63	145.9±28.87
WT	291	12.94±10.99	5372	1605.66±124.69

Figure 2-1 GLay result illustrations.

The top figure illustrates the community structure on the galFiltered network shipped with Cytoscape[106]. The bottom figure illustrates the fast-greedy community structure superimposed on Frutcherman Reingold grid layout from the largest component of Cytoscape human BIND dataset, consists of 17961 nodes and 30156 edges. Note that nodes belong to the same community tend to aggregate spatially, which resulted in clusters with good visual separation. The red circle indicates a group of highly interacting immunoglobulins.

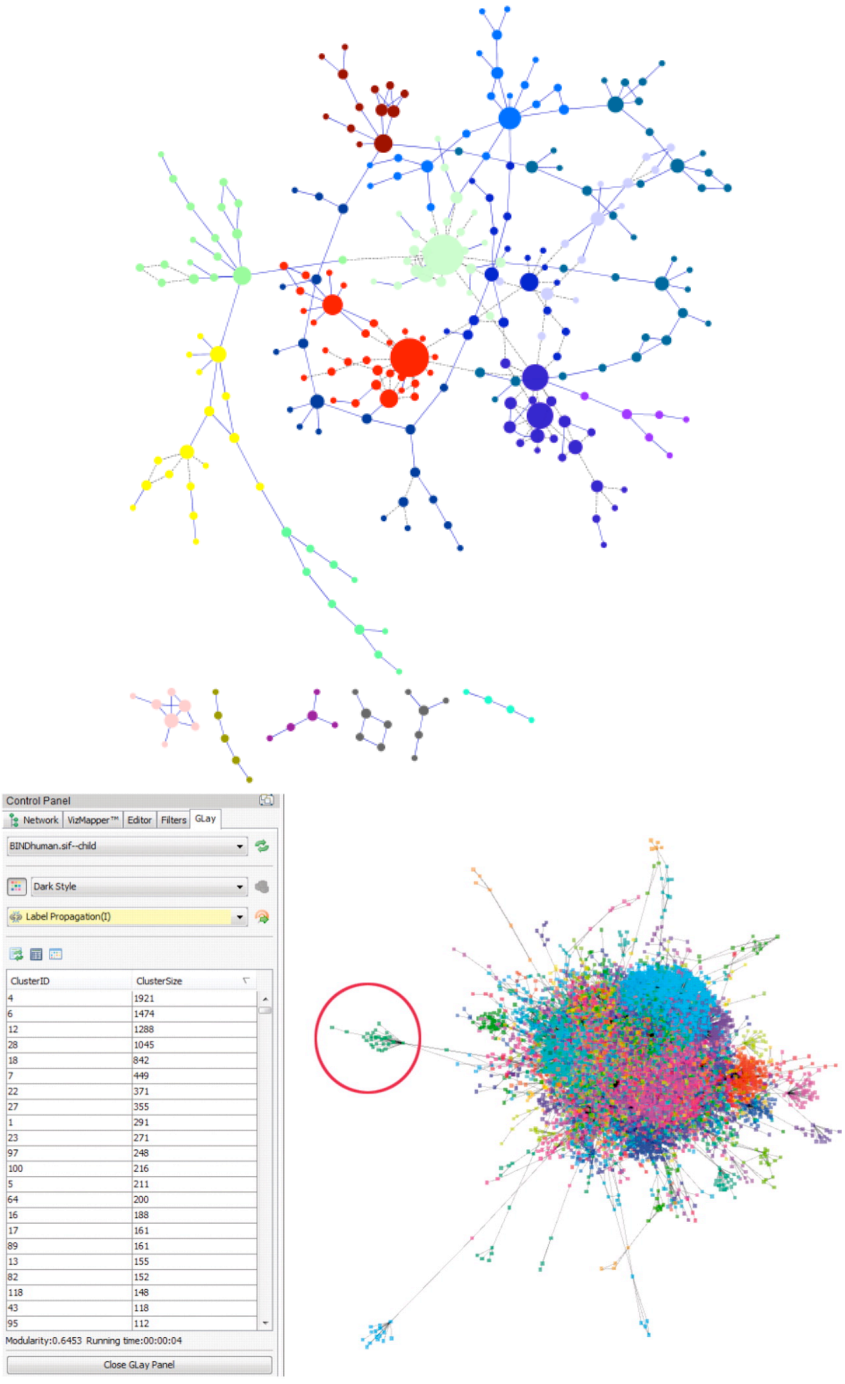


Figure 2-2 Comparison of GLay and MCODE results.

Comparison between clusters produced by MCODE with default parameters and GLay using fast greedy algorithm on Cytoscape bundled galFiltered dataset. The node color is determined by the corresponding cluster membership. Left: MCODE clusters. The un-clustered genes are hidden from view. Right: GLay fast-greedy clusters. (A) A MCODE cluster, in which four out of five genes are associated with MAPK pathway. The corresponding cluster in GLay contains 25 genes, including more genes in MAPK pathway, cell cycle and ion binding. (B) A GLay cluster not identifiable by MCODE. This cluster consists of six genes, with four related to RNA processing.

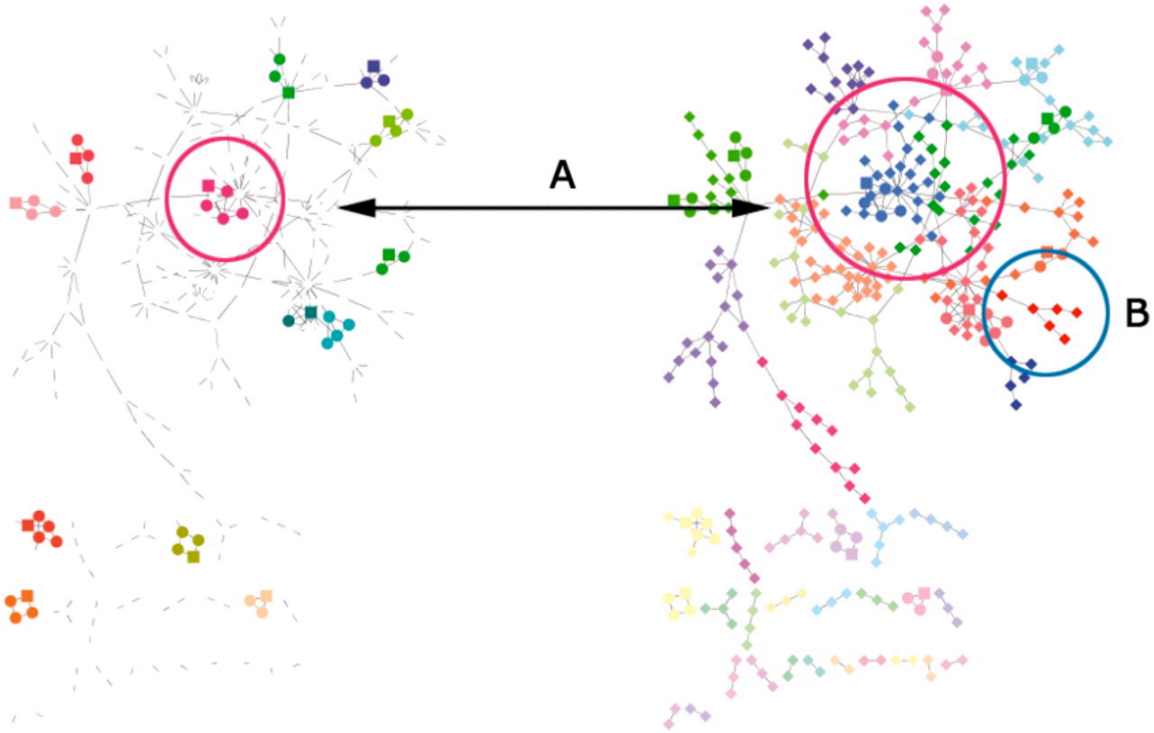
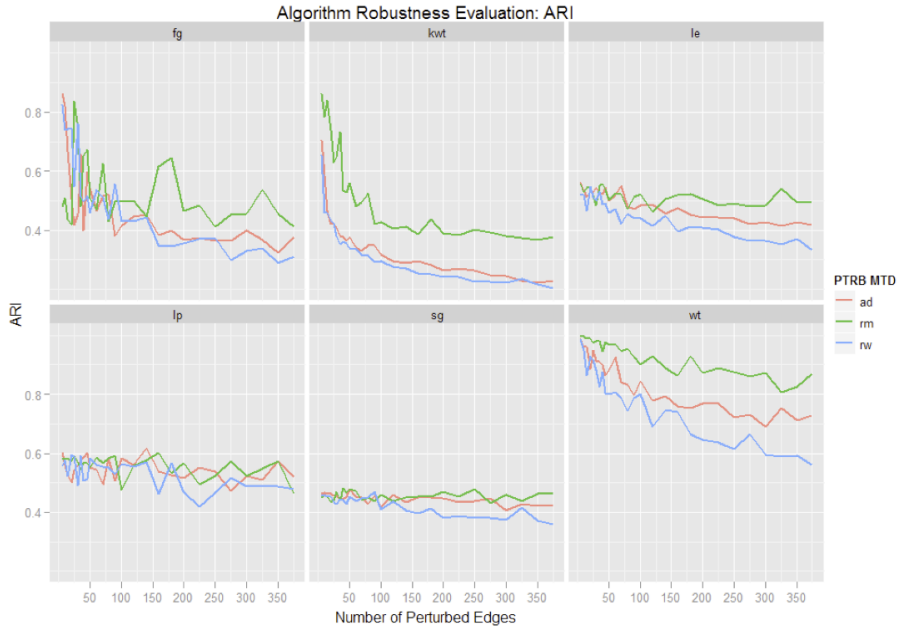


Figure 2-3 Robustness test on community algorithms.

These two figures demonstrate the robustness of algorithms when the rat Interactome network is perturbed with certain number of edges added (ad), removed (rm) or rewired (rw), measured in ARI (the agreement with result from original communities) and dispersion (the reliability of community structure).



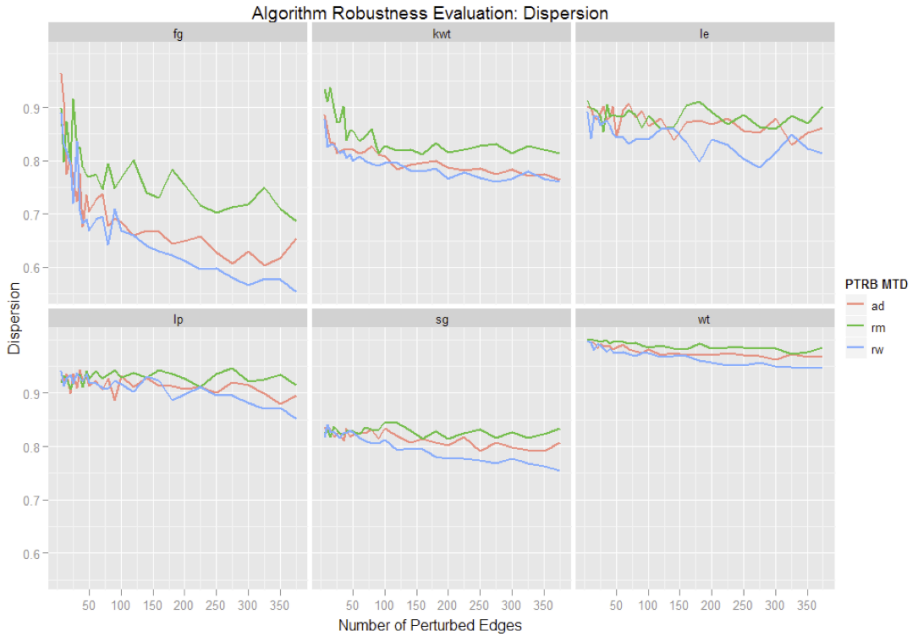
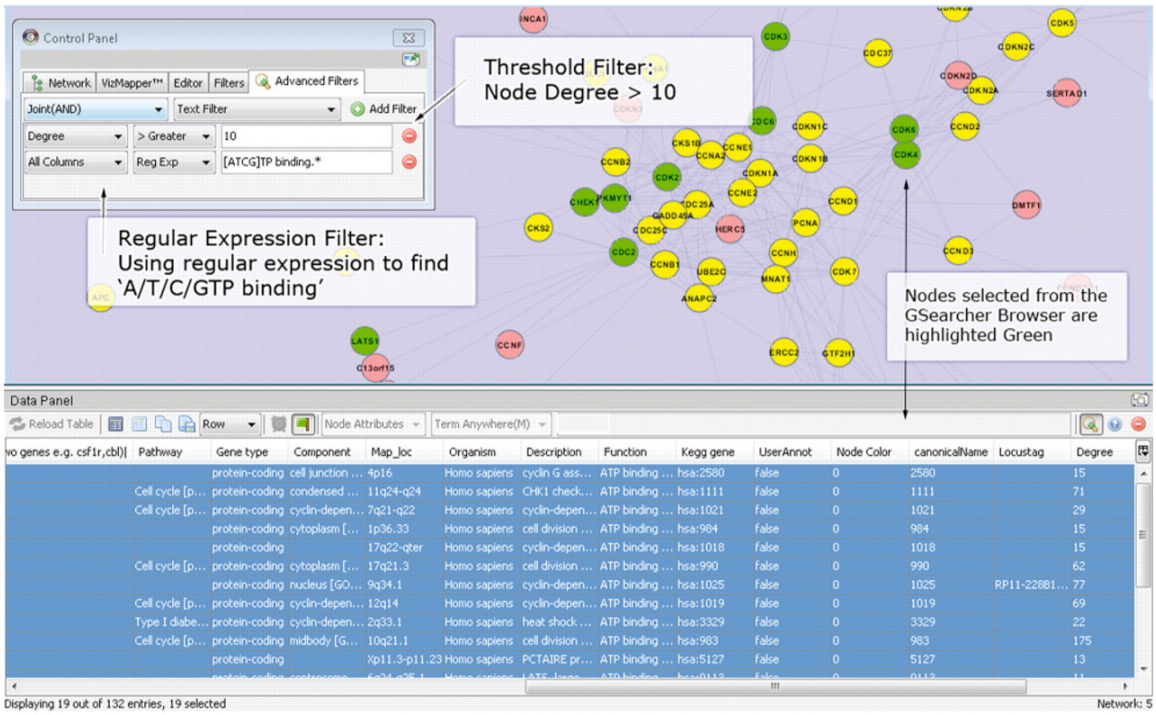


Figure 2-4 GSearcher result illustration.

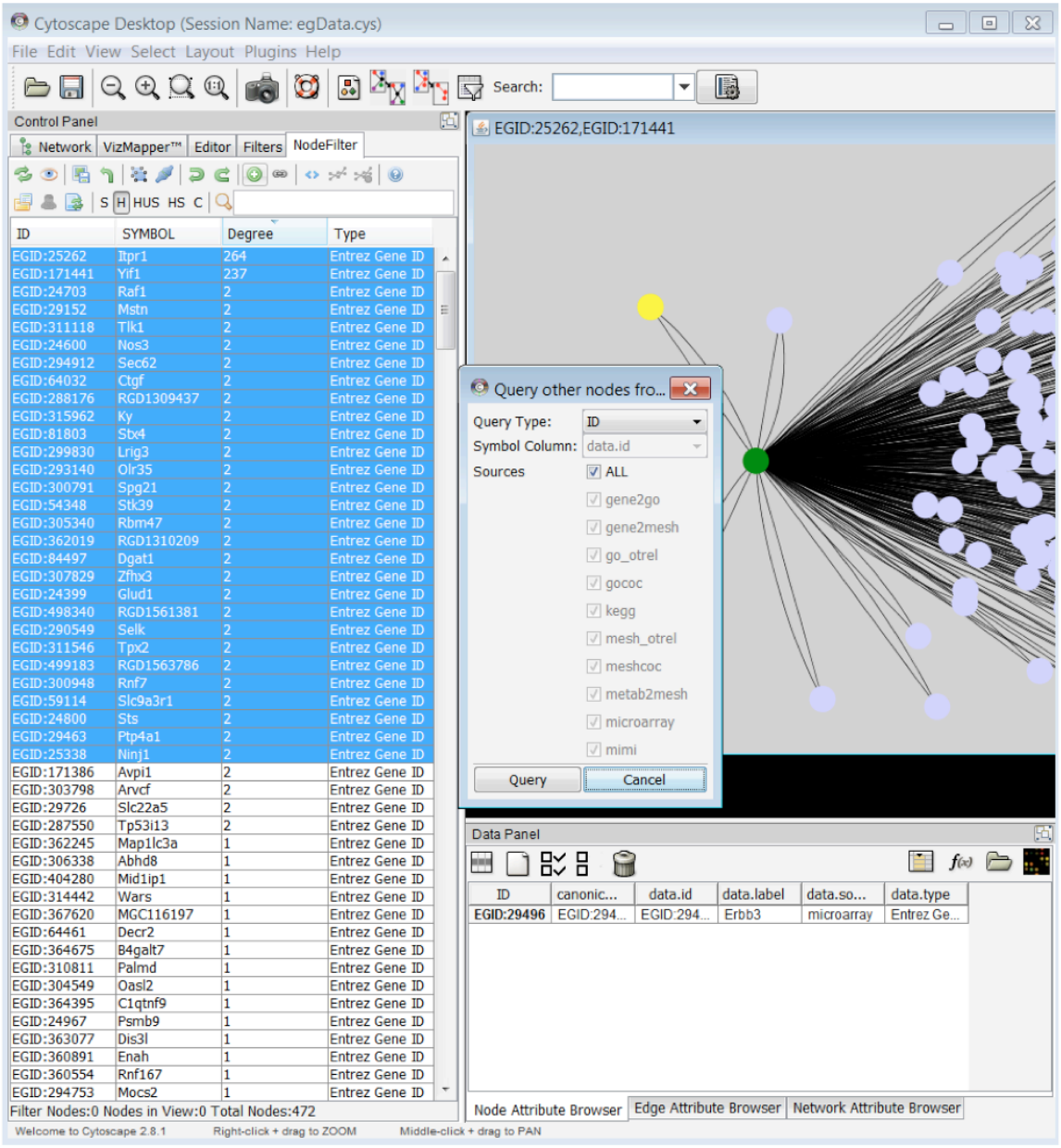
A screen shot of GSearcher on MiMI human Interactome. A subnetwork is created from nodes with their attributes containing the keyword 'cyclin'. Nodes can be dynamically selected from the GSearcher browser using conditions in the two illustrated filters. These nodes are highlighted in green. In contrast, previous selections by the user or other plugins are highlighted in yellow, demonstrating how GSearcher interact with other plugins with minimal interfererence.



vo genes e.g. csf1r,cbl)|	Pathway	Gene type	Component	Map_loc	Organism	Description	Function	Kegg gene	UserAnnot	Node Color	canonicalName	Locustag	Degree
		protein-coding	cell junction ...	4p16	Homo sapiens	cyclin G ass...	ATP binding ...	hsa:2580	false	0	2580		15
	Cell cycle [p...	protein-coding	condensed ...	11q24-q24	Homo sapiens	CHK1 check...	ATP binding ...	hsa:1111	false	0	1111		71
	Cell cycle [p...	protein-coding	cyclin-depen...	7q21-q22	Homo sapiens	cyclin-depen...	ATP binding ...	hsa:1021	false	0	1021		29
		protein-coding	cytoplasm [...	1p36.33	Homo sapiens	cell division ...	ATP binding ...	hsa:984	false	0	984		15
		protein-coding		17q22-qter	Homo sapiens	cyclin-depen...	ATP binding ...	hsa:1018	false	0	1018		15
	Cell cycle [p...	protein-coding	cytoplasm [...	17q21.3	Homo sapiens	cell division ...	ATP binding ...	hsa:990	false	0	990		62
		protein-coding	nucleus [GO...	9q34.1	Homo sapiens	cyclin-depen...	ATP binding ...	hsa:1025	false	0	1025	RP11-226B1...	77
	Cell cycle [p...	protein-coding	cyclin-depen...	12q14	Homo sapiens	cyclin-depen...	ATP binding ...	hsa:1019	false	0	1019		69
	Type I diabe...	protein-coding	cyclin-depen...	2q33.1	Homo sapiens	heat shock ...	ATP binding ...	hsa:3329	false	0	3329		22
	Cell cycle [p...	protein-coding	midbody [G...	10q21.1	Homo sapiens	cell division ...	ATP binding ...	hsa:983	false	0	983		175
		protein-coding		Xp11.3-p11.23	Homo sapiens	PCTAIRE pr...	ATP binding ...	hsa:5127	false	0	5127		13

Figure 2-5 Node Filter result illustration.

Illustrates Node Filter with the major functions. The left panel displays the search results and the user may select of filter out nodes of interest. The popup-panel allows the user to query the database and extend the current network from the selected node (yellow). Green highlights one of the marked nodes by Node Filter. The Node filter can work seamlessly with built-in Node Attribute Explorer and other Cytoscape plugins.



Chapter 3 Cross-Omics Knowledge-mining

3.1 Introduction

As stated in Chapter 1, the 'omics boom' has made it possible for high-throughput integrative data analysis and Biomarker discovery. As each omics data type captures one aspect of the data, by pooling together the jigsaw puzzles from different omics measurements it is potentially possible to obtain a more complete picture of the underlying mechanisms than study each of the Omics data alone. Because of the relatively low cost and time requirements, the integrated transcriptomics-metabolomics analysis has become one of the popular cross-omics paradigms.

Overall, metabolites are transients, intermediate and end products of cellular processes thus the profile of metabolites provides a snapshot of the physiological state of a cell complementary to its transcriptome and proteome, which manifest functional status of more upstream events. The size of Metabolome is about 1-2 orders of magnitude smaller than the transcriptome and the proteome; the steady state concentration of a metabolite usually reflect the combinatorial effects of multiple upstream factors such as environmental stimuli, nutrition availability and genomic structure influences. Conceivably, this property may make metabolites better biomarkers for certain conditions as they responses rapidly to physiological changes. In addition, identifying relationships between metabolites and genes/genome structures (genomics), gene expression (transcriptomics) and protein expression (proteomics) can potentially help to elucidate molecular mechanisms involved in a variety pathophysiological processes, such as oncology, diabetes, etc.

Studies on gene expression profile and proteomics data have shown significant molecular signatures. It is generally accepted that meta-analysis of the pooled omics data would notably improve our knowledge of biological pathways and processes. The quantitative study of gene to metabolite associations may not only complement the qualitative knowledge of certain

metabolic reactions, but also unravel novel biological processes. These associations could also potentially help us identify unknown metabolites by reducing the search space, classify and validate unknown cell-lines on metabolomics level and serve as biomarkers.

Previous studies have investigated significant associations between the transcriptome and metabolome, which add another layer of quantitative inferences to gene-wise correlation. Ferrara et. al have coupled metabolic and transcriptional profiling to construct causal networks which demonstrates fluctuation of gene expression in changes of metabolite availability[150]. Xu et. al have performed integrated pathway analysis on rat urine metabolic profiles and kidney Transcriptomic profiles to study the underlying mechanism in toxicology of model nephrotoxicants[151]. Nam et. al demonstrated that the integrative study of Transcriptomics and Metabolomics could effectively identify metabolic biomarkers for breast cancer[152]. Some online databases, such as Cornell Tomato Functional Genomics Database (TFGD)[153], provides a quick access portal to query correlations between selected metabolites and gene expression profiles for plant sciences. Nevertheless, the noisy nature of the Metabolome data, limited number of metabolites with known structures, and the lack of metabolite-gene interaction annotations despite databases like Edinburgh Human Metabolic Network (EHMN)[46] and BiGG[154], the indirect nature of potential gene-metabolite relationships, and the lack of large scale Metabolome study data in the public domain, still limit the knowledge mining of metabolites and their regulation in normal and disease processes.

Conceivably, cancer samples provide excellent opportunities for identifying metabolic biomarkers and gene-metabolome relationships due to the dramatic function alternations at the molecular level in cancer tissues. For example, cancer tissues usually exhibit more than 10-fold changes in the expression level of many genes in numerous microarray studies. Recently, the collaborative NCI60 project from the Developmental Therapeutics Program (DTP) of the National Cancer Institute (NCI) has made extensive measurements of various Omics data publicly available, including microarray, Metabolomics, Proteomics, Epigenetics, etc. While the number of Metabolome and Metabolome related studies and biomarker discoveries associated with cancer is increasing rapidly in targeted studies, there is still no literature on comprehensive analysis of metabolic features and their regulatory mechanisms of difference cancer types across multiple Omics data at the time of this analysis.

There are three major challenges in Metabolome analysis. First of all, the number of clearly characterized metabolites is much smaller (a few hundred to a few thousand) when comparing with transcriptome (Tens of thousands). Many metabolites are involved in multiple processes, which makes it difficult to establish clear-cut associations between a metabolite and other Omics profiles. Secondly, metabolites are small interchangeable molecules that fluctuate rapidly to the changes in various external signals. Therefore measurement of metabolites is more susceptible to the experimental designs and conditions. The resultant metabolic profiles are more likely to be non-normally distributed, with multi-modes or outliers. Some of the classic analytical methods, such as Pearson Correlation Analysis (PCC), Principle Component Analysis (PCA) or Linear Regression models may fail to work properly. It has been demonstrated that using robust correlation measures instead of PCC or Spearman correlation could improve the estimate[155,156]. Thirdly, it's more difficult to validate the significant findings in Metabolome than Transcriptome or Proteome. The annotated data for gene-wise/protein-protein associations accumulated from literature-mining and molecular interaction databases are much more comprehensive for Transcriptome/Proteome than Metabolome. Although some databases, such as EHMN, have compiled gene to metabolite relationships from metabolic pathways, the scale and scope of such data are still quite small[46]. Moreover, some metabolite to gene associations may not be attributable to known primary metabolic reactions, but rather via much more subtle and unknown processes. Systematic validation pipelines are also not readily available for the metabolome, whereas for the transcriptome/proteome there's a rich collection of toolkits available for researchers (GSEA[94], DAVID[93], etc).

In this cross-omics study, we investigated whether different cancer cell lines have distinct metabolic signatures and whether available Metabolome data for NCI-60 cell lines could be suitable for cancer subtype classification. We also attempted to identify distinct gene-metabolic relationships in cancer cell lines, where gene expression or genomic structure changes may be associated with synergistical fluctuations in metabolic profiles. Our expected result is that the combined Metabolome and Transcriptome analysis would reveal some abnormal regulatory relationships in some of the NCI-60 dataset not possible by either Metabolome or Transcriptome study alone. Such abnormal regulatory relationships will be the starting point for follow-up wet lab studies for understanding the metabolic and signal

transduction pathways involved in cancer genesis. We also proposed a heatmap visualization method for bidirectional hierarchical heatmaps that eventually leads to the development of the CoolMap described in chapter 4.

3.2 The analysis of NCI-60 Dataset

3.2.1 Materials and Methods

Data Preprocessing: NCI-60 data preprocessing: raw molecular datasets were downloaded from DTP web portal (March 2007 release), consist of 57 cell-lines in 9 cancer types have both microarray and metabolomics data. We used our in-house Entrez-based Custom CDF version 12 to derive gene-level expression data from the NCI-60 Affymetrix Genechip CEL files[157]. The metabolite data averaged over triplicate experiments were manually compared with a reference dataset to identify annotate imputed values (Beecher, unpublished data) that accounts for 33.9% of the raw dataset. The imputed data could significantly bias the inferences drawn from subsequent statistical analysis, thus were excluded. All metabolite names were manually compared with KEGG metabolite database and assigned a KEGG compound ID whenever possible. The preprocessing step produced 11961 and 6089 gene expression profiles from u133a and u133b chip respectively, and quantitative data for 124 known and 218 unknown compounds.

Statistical Analysis: All statistical analyses were performed in R[xxviii]. The matrix RV coefficient was computed using FactoMineR package[158]. Classification and variable selection were performed by R randomForest[xxix] and varSelRF package[87]. The robust correlations were computed with robust package[xxx], using the parameter pair-wise Quadrant Correlation (pairwiseQC). Multidimensional outliers were identified by package mvoutlier[114]. Optimal robust correlation estimation was selected using methods described in previous sections. We

[xxviii] http://www.r-project.org/

[xxix] http://cran.r-project.org/web/packages/randomForest/index.html

[xxx] http://cran.r-project.org/web/packages/robust/index.html

ran our computation on MBNI cluster of ~100 nodes and loaded all results into our Oracle database server for integrative analysis from multiple datasets.

Resources used for validating significant correlations: The annotated gene to metabolite relationship data kindly offered by the EHMN project is used for identifying biological relevant known gene-metabolite relationships revealed by our analysis. We also compiled a local version of KEGG and DAVID 2008 for further validations. In order to associate known mutations and CNVs in NCI-60 cell lines to abnormal gene-metabolite relationships, we have built a local copy of COSMIC[159] dataset from Sanger and Tumorscape[160] from Broad institute for cell-line specific point-mutation and copy number variations (CNV), respectively.

3.2.2 Results

3.2.2.1 Metabolomic Signature of Cancer Cells from Different Tissue Origins

The first question we would like to address is whether cancer cell lines exhibit tissue-origin specific metabolic signatures. In our initial analysis, we tried classification analysis similar to those performed in microarray[161] to classify 57 cell-lines into 9 cancer types from metabolite profiles. However, regardless of method used, the classification error on metabolomics data is much higher than that from microarray (~0.51 for metabolomics and ~0.34 for microarray). Several factors could have contributed to this phenomenon. First of all, it has been shown that some of the cell lines may be assigned wrong labels (Unoffical discussions with Beecher). Some cancer samples, such as breast, contain a mixture of different tissue types with high intra-class heterogeneity. In addition, some cancer classes have very few samples. Prostate cancer only has two samples, which is insufficient to generalize adequate features for separation from other cell lines. Moreover, the high variability of metabolic profiles may also contribute to the large classification error. The Matrix RV coefficient (RVC) gives an estimate of correlations between different matrices. The RVC between u133a and u133b chip is 0.91, which demonstrates the high concordance between different microarray measurements. The RVC between u133a and metabolomics is 0.11 and RVC between u133b and metabolomics is 0.09. This implies that there could be strong disagreement between Transcriptome and Metabolome (disagreement also occur between Transcriptome and

Proteome[40], or even microarray and RNA-Seq[14], but much weaker), which may also elevate classification error.

As the metabolite profiles are unable to correctly classify the entire set of cell lines, we then further inspected whether these classifiers could perform well in a subset of samples. We progressively removed the cancer class that contributed most to the out-of-bag (OOB) error estimate and recomputed classification errors. After removing Prostate, Breast Ovarian and CNS, metabolite classifiers can reach comparable performance with microarray classifiers Figure 3-1. This indicates that the disagreement between Metabolome and Transcriptome may arise from a subset of the samples with high variability or heterogeneity. Metabolome, if mined with robust methods, still captures many functional features of the cell and therefore the association between Metabolome and Transcriptome should provide us with new insights on the underlying biological processes.

3.2.2.2 Correlation Analysis

Although the limited number of cell lines for each cancer type and the noisy nature of Metabolome data prevent the use of Metabolome data alone as the cancer cell line classifier, we hypothesized that different cell lines should share some basic regulatory and metabolic processes essential for cell growth and metabolism. It is likely that although different cancer cell lines may have very different levels of gene expression and metabolism, the steady-state relationship between genes and metabolites in the same or highly coupled pathways should be maintained across different cell lines in the absence of dramatic genomic changes such as gene mutation and copy number variation (CNV). Identifying such gene-metabolite relationships at steady state will help us better understand the underlying molecular mechanisms related to metabolic change. They will also help us to infer potential metabolic change based on expression data or vice versa. On the other hand, if a few cell lines deviated significantly from the gene-metabolite relationships exhibited by the rest of the samples, the related genes in such outlier cell lines are likely to be silenced (no expression) or stimulated (over expression) due to genomic structural changes. Consequently, we are interested in both high correlations, which reflects steady state trend over all samples, and outliers, which reflects signatures in specific cell lines. The latter is usually overlooked in high-throughput analysis; even though outliers could have strong biological significance, systematic removal

of outliers could lead to better clustering, classification and prediction accuracy of classifiers[114,162,163].

The classic method of inspecting the relatedness of two quantitative features is by computing the Pearson Correlation (PCC). While PCC has been proven to work well in many previous gene expression studies[164], the correlation is susceptible to inflation by outliers, which made it not suitable to analyze high throughput data with high background noise. As describe before, metabolic profiles have demonstrated high variability, and we have identified many cases in our gene-metabolite correlation analysis where a large number of high PCC correlations were merely artifacts resulting from a few extreme outliers (discussed in detail later). On the other hand, high background noise also affects PCC, which would underestimate the association if missing values were replaced with very small base values. To tackle this challenge, we proposed to use robust correlation estimates instead of PCC. Robust correlations offer a more precise estimate of the true association in the presence of multi-dimensional outliers, given a sufficient sample size. We chose Pair-wise Quadrant Correlation (PQC) for its good computational performance. In our following correlation analysis, we propose to use a novel integrative analysis with both PCC and PQC. Gene-metabolite pairs with high PCC and high PQC tend to demonstrate true linear correlation across all cancer types, which imply a general functional association between gene and metabolite profiles; gene-metabolite pairs with high PCC and low PQC are possibly resulted from a few extreme outliers, which are potentially linked to cell-line-specific signatures. A collection of robust correlation estimation methods were evaluated using simulated data and the results are presented later in this chapter.

3.2.2.3 Metabolite – Metabolite Correlations

We have computed both PCC and PQC for metabolite-metabolite correlations. As metabolites with fewer than 10 valid measurement values were removed from the analysis results as they tend to inflate PCC and PQC because of the small sample sizes. Table 3-1 shows the top metabolite pairs with both high PCC and PQC. It can be seen that there are high correlations between compounds of different forms (L-allo-theronine ~ threonine), between different amino-acids (leucine ~ methionine) and between mixture and compounds (isobar6 includes valine and betaine ~ valine). There are also some high correlations between unknown compounds with known compounds, such as the correlation between tyramine

and X-4043 that is almost perfectly linear. We could then reason that these unknown compounds are therefore either structurally or functionally closely associated with the known compounds, utilizing this information for prediction. Unfortunately, the MS data were missing from the NCI-60 datasets; otherwise it would be possible to search online databases such as HMDB[43,44] to predict the unknown metabolites.

3.2.2.4 Metabolite – Gene Correlations

We computed the PCC and PQC for all gene expression profiles and metabolite measurements. Similarly, metabolites with less than 10 valid measurements were removed. Genes with low expression profiles in the majority of the samples due to tissue/cell-type specificity were also removed. Table 3-2 shows the top 10 gene-metabolite associations.

To investigate the biological significances of the correlation analysis, we mapped all the known gene-compound associations from EHMN to our NCI60 analysis. 721 entries of associations, including 352 genes and 81 known metabolites, were mapped. The percent of gene and metabolite mapping to known pathways were very low due to the current shortage of metabolite-gene annotations in existing databases. Besides, the fraction of gene-metabolite pairs with moderately high correlation among all the mapped pairs is also very small. We chose the top 9 pathways in EHMN mapping ordered by the number of genes contained in each pathway. There are several possible explanations to the low match ups of significant correlations to EHMN data. First of all, the high level of abnormal regulation in the cancer cell lines may have masked any global mechanism such that it's difficult to identify high gene to metabolite correlations across all cell lines. High correlations may be identifiable in cancer subtypes but not applicable in this case due to the small sample sizes in each cancer class. Secondly, metabolites display higher variability than gene expression profiles, which could reduce the consistency of coupled gene and compound expression levels. Nevertheless, some of the gene-compound pairs does show high association in across all cell-lines. For example, gene AKR1B1 which reduces L-Arabitol to L-Arabinose with EHMN reaction ID R01758 and R01759, is associated with L-Arabitol with PQC of 0.69 and PCC of 0.36, which implies an association between metabolite level and gene activity. However, it is surprising that most of the direct enzyme-metabolite relationships could not be mapped to high correlations (Figure 3-2). There are several explanations to the low correspondence of significant correlations to EHMN data. A simple explanation is most of the annotatable

gene-metabolite relationships in the NCI-60 dataset may not be the rate limiting factor in the related pathways. It is also possible that the high level of abnormal regulation in the cancer cell line may have masked global mechanism such that it is difficult to identify high gene to metabolite correlations across all cell lines.

Interestingly the grouping of genes in the same pathway in Figure 3-3 enables us to detect many metabolites that exhibit high correlations with other genes in the same pathway. For example, although phospoenolpyruvate does not overlap with any of the EHMN annotated direct reaction genes, it has high correlation with GPI and ALDOA, two of the genes in he glycolysis module that are known to be highly regulated by hypoxia-inducible factor1l alpha and such regulation is related to the aggressive phenotype of hepatocellular carcinoma. Consequently, ALDOA, GPI and other genes highly correlated with phosphoenolpyruvate in our multi-cancer cell line analysis may suggest that these genes has more significant regulatory or rate limiting roles in glycolysis than genes such as ENO1, ENO2 that are directly related to reactions involving phosphoenolpyruvatge in these cancer cell lines. Naturally, not all genes in the same pathway strongly correlate with each other since genes in the same pathway are not always changed in the same direction.

Further investigation would be worthwhile for high correlation between metabolites and other genes (i.e. those not directly involved in the specific metabolic reactions) in the same pathway, as such un-annotated relationship are likely to help us identify speed limiting enzymes in a pathway, key regulatory genes of related pathways or novel metabolic mechanisms.

3.2.2.5 Outlier Analysis

In our previous analysis we used PQC in addition to PCC to identify molecule pairs with true high correlation. We have also identified many cases where cell lines have one or a few gene-metabolite pairs with much higher expression values than the rest of the other samples, which directly produce inflated PCC and very low PQC. To systematically identify these cases, we used R package mvoulier to detect multidimensional outliers, and recomputed the PCC and PQC scores after outlier removal. Our empirical rule shows that when PCC > 0.6 and PQC < 0.3 (Preferably close to 0), and the number of multidimensional outliers is smaller than 3, the high PCC is most likely to be an artifact from very few extreme outliers.

To explore the biological significance of these outliers, we compared the outlier cases detected by the criteria mentioned above to the Sanger Cosmic Database[159]. Sanger Cosmic contains 177 mutation entries of 28 genes in our 57 NCI-60 cell lies. Interestingly, the top two outliers detected by our approach, the NOTCH1-X-2005 relationship in the MOLT-4 cell line and the KRAS-X-2690 relationship in the OVCAR_5 cell line, have known gene mutations in those two genes, respectively, in the Sanger Cosmic database. The common feature of these two gene-metabolite pairs is high PCC and low PQC before outlier removal and low PCC and low PQC after the outlier removal. From **Error! Reference source not found.**Figure 3-3 we can verify that the inflated PCC were indeed a product of single cell-line outliers. Besides, the sample sizes of these two cases are significantly large (22 for the NOTCH1-X-2005 pair and 26 for the KRAS-X-2690 pair) so that the outlier is not likely to be a random fluctuation as a result of small sample size.

Since our analysis does not take advantage of any cell line or gene mutation information, the fact that our top two outliers overlaps with documented gene mutations in the Sanger Cosmic dataset from the unbiased analysis suggest that the mutations may be the cause of such abnormal relationships. In addition to point mutations, we also compared the outlier analysis results with Copy Number Variation (CNV) data from the Broad Institute. Only 34 out of 57 samples from NCI60 have CNV data., but we have also found some gene-metabolite outlier pairs that are consistent with CNV outliers. For example, BRIP1, with CNV of 14.22 in cell line MCF7, has PCC of 0.90 and PQC of 0.008 and one outlier. The corresponding compound, X-3363, maybe associated specifically to this copy number variation.

The fact that some top ranked gene-metabolite relationship outliers detected by our approach matches with known genomic structure changes related to the same genes in the corresponding cell lines strongly suggest the usefulness of our analysis method in identifying potential molecular mechanisms related to Metabolome and Transcriptome changes.

3.2.3 Summary

Our analysis results on the NCI-60 Metabolome and Transcriptome data suggest that 1) while there are metabolic signatures associated with cancer subtypes, the small sample size and the high noise level in the current NCI-60 Metabolome dataset makes it unsuitable for

cancer subtype classification purposes. 2) There are indeed biologically meaningful high correlation gene-metabolite pairs across NCI-60 cell lines, identifiable by robust correlation estimates. 3) Most strikingly, there are several example of abnormal gene-metabolite coupling that can be directly linked to known gene mutations or copy number variations. These findings will help us to investigate the molecular mechanisms involved in metabolic changes in some of the NCI-60 cell lines through more targeted wet lab experiments.

The high correlations as well as outliers can be utilized to aid the progressive prediction of unknown metabolites based on annotations in existing pathway databases and literature. For example, the high correlation of an unknown compound with a known gene and in particular, multiple known genes in a pathway can dramatically reduce the search space for the unknown compound, since the most likely candidates will be structurally related molecules or known metabolites from related pathways. We plan to compare the Mass Spectrometer (MS) features of these unknown compounds from the predicted candidate pool and conduct wet-lab experiments for validation. Since we have discovered that many unknown metabolites are strongly associated with each other but not with any known compounds. Correct determination even a small fraction of them would facilitate the identification of the rest, which also will in turn improve our understanding of the molecular processes and pathways involving these molecules.

Our result suggests that the 'house-keeping' gene-metabolite association which dominates fundamental metabolic processes may be more informative when studied in homogeneous cell samples or cell-lines with fewer changes in metabolic state than those of cancer cell-lines, as the mapping of annotated gene-metabolite interactions from databases to computed PCC/PQC scores is quite poor. Nevertheless, some significant outliers can be nicely matched to gene-specific mutations in a few cell lines by our proposed integrative method. Most of the current correlation analyses in gene-expression/gene-metabolite association are performed with PCC only, which not only could be misleading because of multidimensional outliers, but also ignored the significance of these outliers. We have shown that some of these outliers may be a direct 'phenotype' of cell-specific mutations, and are worthy of further investigation of underlying biological causes.

While the diversity in cancer cell lines masked the underlying common metabolic mechanisms, mutations in genes may result in changes in metabolic state, accompanied by drastic fluctuations in certain metabolites. The metabolite-metabolite and gene-metabolite correlations can be explored to facilitate clarification of the structure and function of these compounds, which will not only help us add more paths to the metabolic pathway and networks, but also adds another quantitative dimension to this knowledgebase.

3.3 Evaluation of Robust Correlations

As described in previous text, there are two common procedures in analyzing omics data: to find variables that captures the most differences across sample types, and find variables with expression profiles that are consistent across different samples. The most popular approaches to address these two questions are classification and correlation. Other inferences, such as a clustering or network view of the correlations, can also be built[165]. As the scale and complexity of Omics data continue to grow and software that provide analytical aid becomes increasingly easier to use, the biological inferences may fall victim of the errors and noises of the input data if not handled with caution, especially when the analytical methods are not noise-resistant (i.e. robust). Given the large number of Omics profiles analyzed at the same time, it is impractical to manually validate every comparison. Automated methods have been proposed: Hardin et al. addressed the problem and proposed using Tukey's biweight as a robust substitute for Pearson Correlation and applied to real world gene expression analysis[116,155].

The noise in the experimental data can be attributed to several factors. Taking microarrays as an example: first of all, there is individual variability across different biological samples. Some tissues, such as colon or breast, contain many different cell types. The growth and nutritional state of the cell may also significantly influence its gene-expression profiles. Secondly, variability, or systematic error, is introduced by the machine when operated by different technicians or under different experimental conditions. Some genes with very low expression may fall below the dynamic range of the machine. As a result, they may be assigned with a baseline value or marked as missing. These missing values are then confounded with genes with actual expression values near the lower bound of the dynamic

range. On the other hand, experiment errors, such as faulty sample or wrong reagent concentration and non-optimal experiment condition sometimes incur very large outliers, which could significantly bias classic analytical methods. Thirdly, the probe definition of the gene-chip is also error-prone. Dai et al. have demonstrated that a non-optimal mapping of probes to the genome may have a very significant effect on biological inferences[157]. Fourthly, the deregulated pathways of certain samples such as cancer cells may cause some genes to exert very non-normal distributed profiles. The violation of near normally distributed data may make non-robust methods invalid. Fifthly, the multi-dimensional outliers may be hard to detect because of the masking effects[166]. Visual inspections are also difficult when the number of gene-pairs is very large. The end-product, contaminated by the joint effect of all these factors, can be significantly undermined. For example, as we discussed previously the NCI-60 Metabolomics dataset from National Cancer Institute's Developmental Therapeutics Program (DTP http://dtp.nci.nih.gov), contains the measurement of ~ 350 metabolites from ~ 60 cancer cell-lines in 9 different cancer classes. This data also has more than 34% missing values, which were substituted with small, imputed baseline values. Therefore to draw any inferences using non-robust methods from this dataset is difficult. Even though this most comprehensive Metabolomics dataset has been published in the public domain for about three years, there is still no available literature drawn from this dataset.

We therefore deem it very important to evaluate various robust methods in the context of such noisy data. The most of the focus of this analysis will be on correlation, but some of the methods can be adapted to evaluate classification straightforwardly. In this analysis, we analyzed the resistance of seven different correlation estimators (Pearson ρ, Spearman γ, Kendall τ, Hardin's Tukey Biweight, Fast MCD, M-estimate and pairwiseQC) on simulated data contaminated with various outlier and noise structures. We then applied the robust correlation estimator to a Transcriptomics-Metabolomics association study with NCI 60 cell lines. The simulation study and the analysis on the experimental data demonstrated that the classic correlation estimators, even including the naive robust estimators such as Spearman ρ and Kendall τ could produce very misleading results when data are contaminated with missing values and extreme outliers. Robust estimators, on the other hand, may over-estimate the actual correlation for data with a small sample size. Therefore we proposed an integrative approach using both classic and robust correlation to identify highly correlated

gene-metabolite pairs, as well as extreme outliers, which could indicate cell-line specific pathological processes.

3.4 Materials and Methods

Correlation Estimators: The most frequently used correlation estimate is the Pearson Correlation Coefficient (PCC), as the covariance standardized over variances:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(X-\mu_Y))}{\sigma_X \sigma_Y}$$

This correlation estimator ranges from -1 to 1, with 1 indicating perfect positive linear association and -1 indicating perfect negative association. It has been demonstrated that PCC is not outlier resistant. The major reason is that neither the sample mean estimator nor the sample variance estimator is outlier resistant. One quick fix could be to replace the μ and σ as the robust versions, such as median, trimmed mean for sample mean and median absolute deviation (MAD) for sample variance or apply winsorization or huberizing procedure on the data. However, this may cause the PCC ρ to fall outside of [-1,1].

Charles Spearman coined the Spearman's rank correlation coefficient which computes the correlation based on transformed ranks:

$$r = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

Here d is the difference between ranks of X and Y[167]. It is more extreme outlier resistant than PCC because the large outliers' numerical values are transformed into integer ranks. Spearman Υ is frequently used as a robust substitute for PCC.

Kendall's τ is another non-parametric rank-based correlation measure between two random variables. It is calculated as the difference between the number of concordant pairs and the number of discordant pairs:

$$\tau = \frac{N_c - N_d}{\frac{1}{2}n(n-1)}$$

Here N_c denotes the number of concordant pairs, and N_d denotes the number of discordant pairs. It is also considered as a robust correlation estimator[168].

Hardin et. al proposed a robust measure using Tukey's biweight as a robust measure[155]. The idea is to use proper outlier-resistant estimators of location and scatter from the biweight iteration scheme to replace the sample mean and the sample variance in the Pearson Correlation formula:

$$r'_{X,Y} = \frac{s'_{XY}}{s'_X s'_Y}$$

The Fast Minimum Covairance Determinant (MCD) method is an improved robust estimator from MCD method using the subsampling technique[166]. This method can determines multivariate location and scatter by finding *h* observations out of *n* total observations (typically number of samples) for *p* variables (typically number of genes, for gene pairs p = 2), whose classical covariance matrix has the lowest determinant. The FMCD location estimate is then the average of these h points, and the scatter is the covariance matrix of these matrices. Similar to Hardin's approach, the robust location and scale estimators are then plugged into Pearson's formula to obtain the FMCD correlation estimate.

The M-estimate proposed by Maronna is a solution to the system equations provided as below:

$$\frac{1}{n}\sum_{i=1}^{n} u_1[\sqrt{(x_i - t)'V^{-1}(x_i - t)}](x_i - t) = 0$$

$$\frac{1}{n}\sum_{i=1}^{n} u_2[\sqrt{(x_i - t)'V^{-1}(x_i - t)}](x_i - t)(x_i - t)' = V$$

Here *u1* and *u2* are functions satisfying a set of general assumptions. The resultant t is the multivariate location estimate and V is the multivariate scatter estimate[169]. Pair-wise correlations can be directly obtained from V.

Marrona and Zamar recently proposed a pair-wise estimation method modified from the Gnanadesikan-Kettenring robust covariance estimate, using robust versions of variances to replace principle components of corresponding directions, then transform back to the original coordinates to produce a orthogonalized Gnandesikan-Kettenring estimator (OGK, pairwiseGK)[170]. Based on the similar idea, the pair-wise quadrant estimator can also be computed (QC, pairwiseQC).

It should be noted that most of the correlation estimators are capable of estimating multivariate location and scatter. The pair-wise correlation can be obtained from the estimated covariance matrix. However, as the dimension of data grows rapidly, it becomes very demanding to run such programs on a desktop computer without parallelization. For example, the memory limit for 32 bit operating system is 3.5 GB (Gigabytes), and to store a covariance matrix of all the 14,500 genes on the U133A chip alone requires 1.6 GB memory with 64bit double. Some meta-analysis even requires computing correlation across several different chips or Omics data, which makes the estimation of covariance matrix even more difficult. It has been proposed by Chilson et. al that some steps in the computation the covariance matrix can be parallelized[171]. Another way to alleviate the problem is to compute the correlation estimators in a pair-wise manner, and then rebuild covariance matrix from pair-wise estimates. Hardin's method employs such an approach[155]. This scheme not only significantly reduces the memory requirement, but also enables simultaneous computation of pair-wise correlation utilizing multi-core desktop computer or computer clusters.

There are some other distance measures which capture the relatedness of two probability distributions, such as Kullback-Leibler divergence (KLD)[172] and mutual information (MI)[173]. The key advantage is that they can capture non-linear association. To apply these methods, the raw data must be transformed into a probability distribution using a 2D histogram. However, as the $n << p$ for most of the Omics data, the small sample size could significantly distort KLD and MI estimates. Therefore we did not compare the efficacy of these methods to the correlation estimators, but for datasets with large n they are worthy of further investigation.

Identification of Outliers: Hardin suggested using robust measure of correlation to 'flag' gene pairs of poor quality whenever there's a disagreement between the classic correlation

and the robust correlation[155]. It is also very important to flag samples which are consistently shown as outliers, as they most likely reflect the profile of cell-lines which severely deviate from the other samples or merely faulty experiments. Graphical analyses and statistical tests, such as Quantile-Quantile (Q-Q) plot and the Dixon's test[174], can be applied to identify univariate outliers. However, multivariate outliers may only appear to be extreme values in a subset of directions, which made them difficult to be detected using simple extension of univariate outlier detection methods. Figure 3-4 demonstrates a bivariate case in which outliers only reside on one direction.

Filzmoser et al. proposed an outlier detection method by weighing each observation using principle components in high space[114]. As shown in Figure 3-4, this method effectively detected all the artificial outliers, with a small number of false positives. Therefore we used this method to flag outliers in the subsequent Transcriptomics-Metabolomics analysis to investigate cell-line specific characteristics.

Data Simulation: We simulated bivariate normally distributed data with noises and outliers resemble the structure in the real world Transcriptomics-Metabolomics data. There are six different data structures:

- Bivariate normal simulated from a given correlation matrix.
- Bivariate normal, with small values resemble imputed missing values added in one direction.
- Bivariate normal, with small values resemble imputed missing values added in both directions.
- Bivariate normal, with extreme outliers at both directions (First Quadrant).
- Bivariate normal, with extreme outliers at only one direction (Second Quadrant).
- Bivariate normal, with extreme outliers at either one direction (Second and Third Quadrant)

These six different data structures are illustrated in **Error! Reference source not found.**Figure 3-5, which represent the noise structure we observed in the Transcriptomics-Metabolomics study. The bivariate normally distributed values were generated with mean of 0.0 and σ of 1.0. For missing values, we randomly chose a subset of observations and set the

values on one direction to be the smallest value on that direction. For outliers, we simulated at the center of 4.0 and σ of 0.2 in the corresponding directions. In each simulation, we tested the sample size of 25, 50 and 100, and prior correlation of -0.98, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75 and 0.98. Various percentages of small values/outliers are also tested. Each test is performed 100 times and mean and standard error for the correlation estimators are calculated.

Computational Environment: The computations are performed using R project for statistical computing (http://www.r-project.org/). Pearson, Spearman and Kendall correlations are computed using function cor. The robust correlations are computed using the robust package (http://cran.r-project.org/web/packages/robust/index.html). Hardin's Tukey's biweight code was obtained from the paper's supplemental information. The outliers are identified using the mvoutlier package (http://cran.r-project.org/web/packages/mvoutlier/index.html). The bivariate normal data are simulated using the MASS package (http://cran.r-project.org/web/packages/MASS/index.html). The transformations between correlation and covariance matrices are performed using the MBESS package (http://cran.r-project.org/web/packages/MBESS/index.html). The analysis was done on an Intel Core i7 920 computer with 12GB memory. We also did some of the computation in parallel on a linux cluster with ~ 100 nodes.

3.5 Results

Result from the simulated Data: We investigated the behavior of seven correlation estimators with the presence of various systematically added errors that resemble the observed noise from the Transcriptomic-Metabolomic data. We use PCC, SP, KD, TB, FMCD, M and QC to represent Pearson, Spearman, Kendall, Hardin's Tukey biweight, M-estimate and pairwiseQC correlation estimators.

The stdRandom tab shows simulation results without the addition of missing values or extreme outliers. We can see that: 1) Sample size does have an effect on correlation estimators; the standard error of all correlation estimators increases when sample size decreases from 100 to 25. For example, when the actual correlation is 0.98, the PCC is 0.98 $\pm$ 0.0039 when sample size is 100, 0.98 $\pm$ 0.0059 when sample size is 50 and 0.98 $\pm$ 0.0082.

2) The standard error for robust estimators is larger possibly due to algorithmic characteristics, such as the subsampling for FMCD. 3) For all sample sizes and all correlations, the Kendall correlation (KD) consistently under-estimates the actual correlation. For example, for sample size 100, the average Kendall correlations for 100 repeated runs at actual correlation of 0.98, 0.75, 0.5, 0.25 and 0 are 0.87, 0.54, 0.33, 0.17 and -0.0018, respectively. 4) FMCD shows a small degree of underestimation of correlations when sample size reduced from 100 to 50. All the other robust estimators have comparable performance with the classic PCC and Spearman estimators, at all actual correlations and sample sizes. In summary, when the data and are approximately normally distributed and not contaminated with extreme outliers or missing values, PCC and Spearman estimators are still favorable due to their faster execution speed.

Next we examined the performance of these estimators when the data are tainted with missing values, along either direction or both directions. The NCI60 metabolomics data contain many small and identical values, which substituted missing values as 'baseline' values. We then similarly pushed a certain percentage of values towards the baseline to replicate the structure of these missing values. The results are shown in the msRandom1D and msRandom2D tab. It can be seen that 1) the addition of even a few dummy missing values could significantly reduce the classic correlation estimators. For sample size 100 and the actual correlation is 0.98, the adding of 10 dummy missing values can reduce PCC from 0.98 to 0.74 and SP from 0.98 to 0.80, and the addition of missing values appear to have a smaller effect on KD, which is reduced from 0.87 to 0.72. It can be seen that contradictory to general perceptions, the rank based SP and KD are not resistant enough to counter even a few outliers. On the other hand, all the robust correlations at this level are almost identical to the actual correlation of 0.98, which demonstrated the effectiveness of their resistance. 2) The sample size only affects the standard error but not the value of correlation estimates. 3) For 10% missing values, TB, FMCD, M and QC works equally well for all sample sizes. However, when the percentage increased to 20%, TB demonstrated severe performance deterioration. For example, when the sample size is 100 and actual correlation is 0.98, TB is reduced to 0.78 when 20% of the values are replaced by 1D dummy missing values and 0.57 when 30% of the values are replaced. FMCD, M and QC still have reasonably good performance when 30% of the values were replaced at the sample size of 25. This is

inconsistent with the default TB breakdown value of 0.2, which indicates the maximum percentage of allowed 'bad' values when the algorithm is still able to obtain good estimates. We tested percentage from 15% to 20% and result showed that the actual breakdown point may be smaller than the theoretical breakdown point. As it is impossible to know the percentage of outliers in the data a priori, it is worthy investigating the relationship between theoretical breakdown point and actual breakdown point for TB as using excessive large breakdown point may incur over-estimation. 4) The results for 2D missing values are similar to that of 1D missing values. There are some slight improvements for all estimators possibly because the missing values along both directions could offset each other. The robust estimators demonstrated much superior performance when the data are tainted with missing values, with FMCD, M and QC outperforming all the other estimators.

Finally, we tested the effect of a few extreme outliers that severely deviate from the rest of the data. The tabs outliers1, outliers2 and outliers23 list results of outliers added to the first, second, second+third quadrant, as described in the Materials and Methods section. The result shows that 1) Adding outliers in the first Quadrant will severely change PCC when the The simulation study only demonstrated the efficacy of robust correlation estimators when either missing values or outliers are present. The actual Transcriptomics-Metabolomics data contain more complex noise than we have attempted to reproduce here. However, the robust estimators FMCD, M, QC and TB have shown much better performance over the classic PCC, SP and KD estimators. Many recent microarray/metabolomics correlation analyses still use PCC or SP as outlier-resistant correlation estimators, and we can see from the simulation analysis that those estimates could significantly deviate from the actual correlation are less affected than PCC but also have underestimation when the actual correlation is negative. TB performed equally well with FMCD, M and QC, but overestimated positive correlation and underestimated negative correlation when 15 outliers are added to the sample of size 100. 2) FMCD and M can give reasonably good estimates even 10 outliers are added to the first Quadrant for the sample size of 25. 3) The results for outliers added to the second quadrant and third quadrant are similar. FMCD and M consistently outperform all the other correlation estimators.

The simulation study only demonstrated the efficacy of robust correlation estimators when either missing values or outliers are present. The actual Transcriptomics-Metabolomics data

contain more complex noise than we have attempted to reproduce here. However, the robust estimators FMCD, M, QC and TB have shown much better performance over the classic PCC, SP and KD estimators. Many recent microarray/metabolomics correlation analyses still use PCC or SP as outlier-resistant correlation estimators, and we can see from the simulation analysis that those estimates could significantly deviate from the actual correlation.

As the visual inspection for tens of thousands of such gene/metabolite pairs is impractical, using robust correlation estimators will produce much better and more confident estimators.

A problem of using robust correlation is related to significance estimation. For PCC, we can test against the null hypothesis of $\rho = 0$ using t test. However, for robust estimators the t is not guaranteed to have approximate t-distribution. Nevertheless, we can retrieve a nominal p-value by plugging the ρ obtained from robust correlation estimators. Another approach is to use permutation tests or bootstrapping if the sample size n is sufficiently large. As permutation tests will dramatically increase the intensity of computation, it is only feasible to run with parallelization on a computer cluster. The p-values can be corrected using Holm-Bonferroni procedure or by FDR using methods proposed by Strimmer[77].

We also have observed in some cases that when the sample size is very small ($n < 10$, data not shown), sometimes the robust estimators may significantly over-estimate the actual correlations. For example, because FMCD uses a subsampling technique, it is very likely for the algorithm to detect a few points that have strong correlation when the n is very small. Therefore it is very important to understand the behavior of robust correlation estimators before application to real world data.

3.6 Discussion

In this chapter, we evaluated the performance of seven correlation estimators when missing values and extreme outliers are present in the data using both systematic simulation and experimental data. Our results suggest that even though the classic correlations perform very well on bivariate normally distributed data, the Pearson, Spearman and Kendall correlation estimators are unable to obtain good estimates on noisy omics data. Spearman and Kendall

estimators do have a weak degree of resistance, but they fail quickly when systematic missing values and errors are added. As it is very difficult to verify all variable pairs using graphical plots for very large datasets, using these classical methods could produce very misleading correlations for subsequent clustering or synthesized networks. The robust correlation estimators have shown superior performance and the strong biological relevance of our results also indicates that the analysis strategy we developed based on the combination of PCC, QC and outlier detection is a powerful approach for integrative analysis of noisy Omics datasets. Our pipeline also flags problematic samples for further investigation in addition to Hardin's scheme that only flags gene-pairs with outliers. In doing so, we facilitate the characterization of sample-specific features. On the other hand, the resistance of robust estimators also varies with regard to data size and noise structure, due to the different algorithmic design. For example, the subsampling of FMCD may incur over-estimation on very small datasets, and Hardin's biweight by default will have a lower breakdown point than the other robust estimators. It is then critical to understand both the behavior of the robust estimators and the characteristics of the experimental data to obtain the best correlation estimates. Furthermore, as the computation of robust correlation estimators is more intensive than the computation of classic estimators, it is more favorable to parallelize large meta-analyses, which involve computing correlations across many Omics datasets.

Conceivably, high correlations as well as outliers can be utilized to aid the progressive prediction of unknown metabolites based on annotations in existing pathway databases and literature. For example, the high correlation of an unknown metabolite with a known gene and in particular, multiple known genes in a pathway can dramatically reduce the search space for the unknown metabolite, since the most likely candidates will be structurally related molecules or known metabolites from related pathways. Correct characterization of even a small fraction of the unknown metabolites would facilitate the identification of the rest, which will in turn improve our understanding of the molecular processes and pathways involving these molecules.

Table 3-1 Top 10 highly associated compound-compound pairs.

With PPC and PQC both > 0.9

Compound A	Compound B	PPC	PQC
L-allo-threonine	threonine	0.996946	0.992606
tyramine	X-4043	0.996749	0.994182
3-phospho-d-glycerate	glyceraldehyde	0.983365	0.988978
Isobar6 includes valine, betaine	valine	0.982947	0.972611
X-1713	X-4027	0.982945	0.980403
X-1111	X-4019	0.974755	0.985652
leucine	methionine	0.968212	0.958808
creatinine	X-3176 (possible creatine)	0.967171	0.958087
anthranilic acid	valine	0.966016	0.940412
leucine	methionine	0.964969	0.974087

Table 3-2 Top 10 highly associated gene-compound pairs.

With PPC and PQC both > 0.65

Compound	Gene	PPC	PQC
X-2005	CD7	0.905804	0.665047
taurine	PREB	0.850938	0.714942
X-2005	CDK6	0.84396	0.654139
taurine	CALCRL	0.78668	0.660189
X-2005	LOC390940	0.779039	0.659253
X-2724	MEPCE	0.744048	0.698484
X-3090	ALPK1	0.73805	0.821748
X-2005	IRF1	0.736982	0.654989
X-2139	TMEM77	0.732769	0.684709
X-2724	KBTBD2	0.732675	0.725019

Figure 3-1 Estimated Out-of-Bag Error (OOB) Error with regard to progressive cancer class removal.

At each step, cancer classes contribute most to classification error were removed and OOB error were recalculated. B: Breast cancer. P: Prostate Cancer. O: Ovarian Cancer. C: CNS. The left plot shows OOB errors on entire dataset and the right plot shows OOB errors on a subset of classifiers selected by varSelRF. The improvement of variable selection is more remarkable for microarray because of the much larger classifier pool to select from (11961 v.s. 342). Metabolite classifiers can achieve average OOB error of 0.28 when B,P,O,C are progressively removed, reduced from 0.51 from the full set. The OOB error for u133a was reduced from 0.34 to 0.18. The average OOB errors from the random cancer class removal are plotted with the orange dashed line.

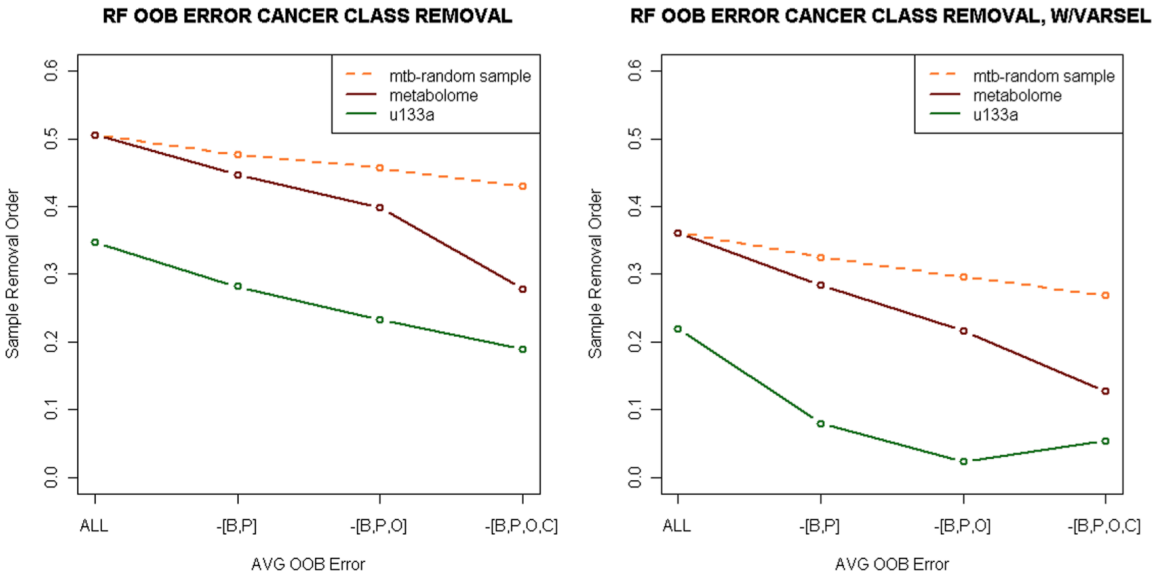


Figure 3-2 Aggregated Heatmap view of gene-metabolite relationship organized according to the KEGG pathway.

Mapping of EHMN gene-metabolite association data to robust correlation matrix heatmap. A: Rows are genes grouped by pathway names, columns are metabolites also grouped by corresponding pathway names. Green circles mark the position of a specific reaction that couples a metabolite and a gene from EHMN. Orange Light cells indicate positive PQC, with black mapped to 0 and orange to be 1. It can be seen from that figure that even though there are patterns of gene-metabolite clustering, very few high correlations can be mapped to known reactions. B: Gene-metabolite correlation in the Glycolysis pathway. C: The gene-gene correlation within the Glycolysis pathway.

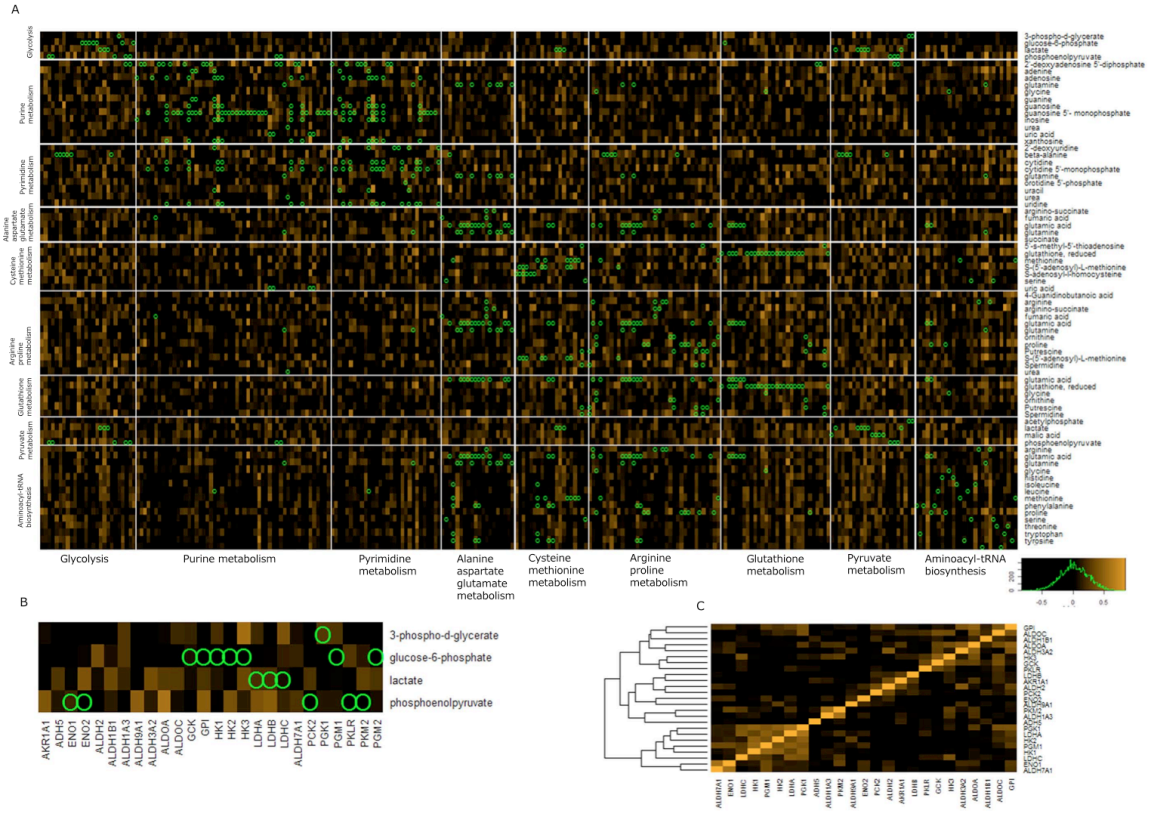
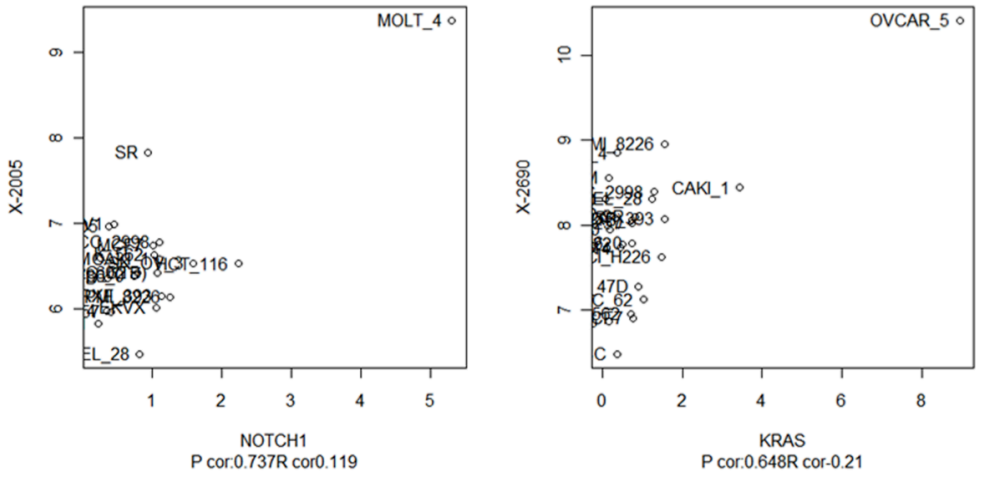


Figure 3-3 Outlier Analysis.

Outlier analysis shows extreme gene-metabolite outlier pairs could be resulted from cell-specific mutations. Top scatter plots: NOTCH1 ~ X2005, with outlier in cell line MOLT_4 and KRAS~X-2690, with outlier in OVCAR5. It can be seen that the high PCC were both artefacts of extreme outliers. Middle table: PCC and PQC of the corresponding pairs, before and after outlier removal. R_QC: PQC. R_QC_RM: PQC after outlier removal. P_Pearson: PCC. R_P_RM: PCC after outlier removal. Bottom table: annotated mutation records from Sanger Cosmic database, directly matched to these two outlier pairs.



Meta_ID	Gene_ID	R_QC	R_Pearson	Outlier_Num	R_QC_RM	R_P_RM	Outlier_Num
371809	4851	0.11852	0.737259	1	0.086509912	0.116586	0
371840	3845	-0.21132	0.648059	2	-0.209199795	0.138102	0

MOLT_4	905958	NOTCH1	p.P2515fs*4	c.7544_7545delCT	Variant of unknown origin	Heterozygous
OVCAR_5	905969	KRAS	p.G12V	c.35G>T	Reported in another cancer sample as somatic	Homozygous

Figure 3-4 Illustration of bivariate outliers.

'o' denotes data generated from a bivariate normal distribution centered at (0,0) and scattered with covariance of (1, 0.8, 0.8, 1). The outliers are added at the center of (0,4) and scattered with the covariance of (1,0.2,0.2,1). '+' denotes the outliers flagged by method proposed by Filzmoser et. al.

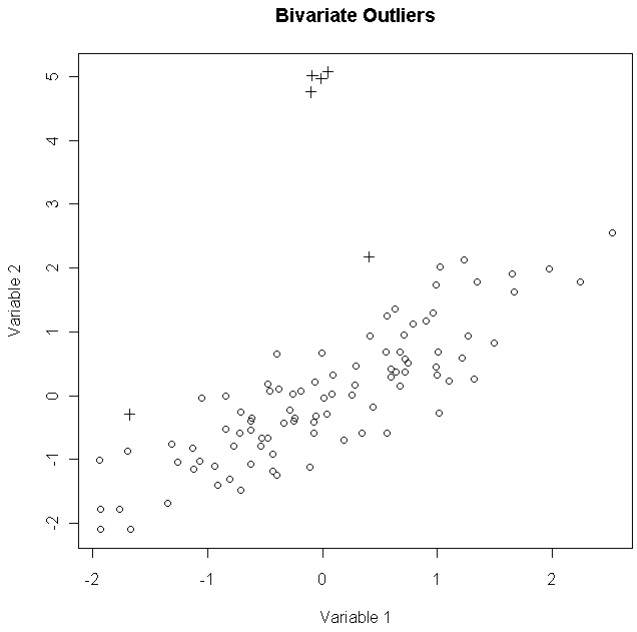
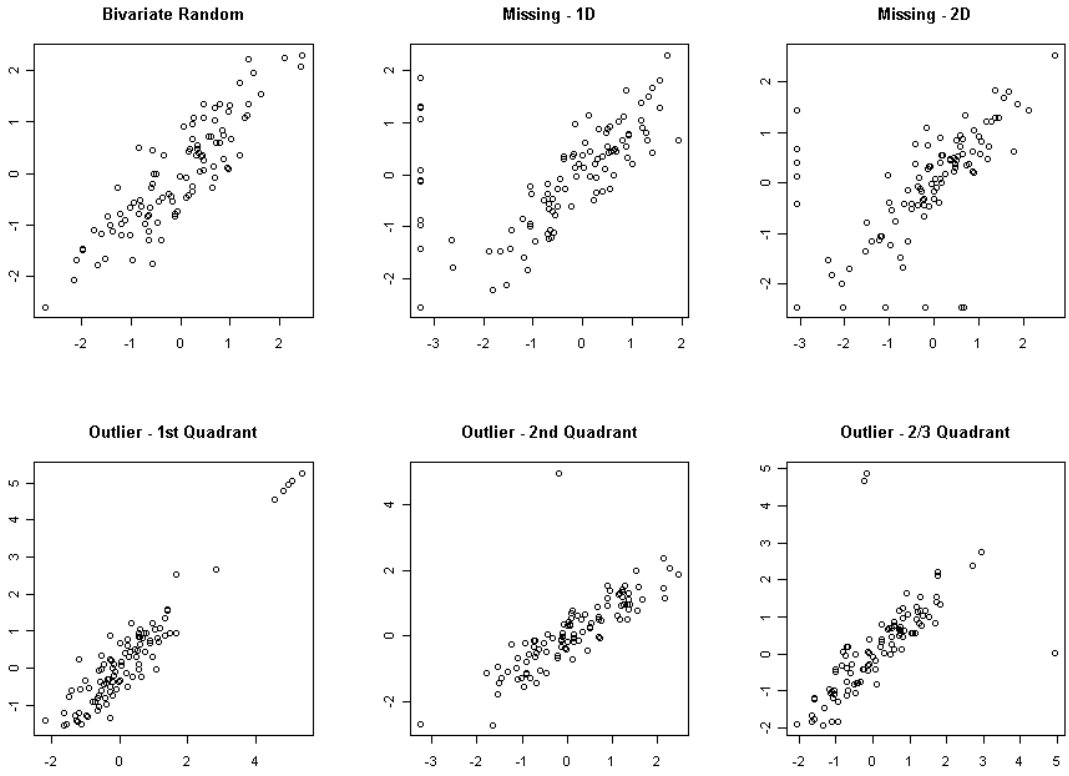


Figure 3-5 Simulated data structure with artificially added outliers.

Bivariate random is simulated with bivariate normal random number generator with given location and scatter. Missing 1D denotes data which is similar to bivariate random but contains missing values along one direction and the missing values were replaced with base values. Missing 2D denotes data that contains missing values along both directions. Outliers are simulated bivariate normal data with outliers added for both directions, only one direction and either one direction, respectively.



Chapter 4 CoolMap for Interactive Exploration of Omics Data

4.1 Introduction

As discussed in previous chapters, the Omics movement has produced a large number of molecular profiles, usually stored in matrix (profile-sample) format or (profile-profile) correlation format. Such datasets are usually visualized as heatmaps using value to color mapping. As the size of such data matrices grow, it becomes increasingly difficult to explore such Omics datasets interactively. There are several challenges to the classic heatmap visualization: first, a modern chip may contain tens of thousands of molecular of profiles and it's impractical to display the resultant heatmap into a single monitor screen without a significant loss of information. Therefore a scaled-down version of the heatmap is usually displayed and artifacts may occur due to the image resizing process. Secondly, many heatmaps with hierarchical clustering lack the interactivity of data exploration, such as the heatmap renderers in R[xxxi], R packages such as ggplot2 and bioconductor, infoViz and Matlab[xxxii]. It is often easy to identify patterns around a certain heatmap region but difficult to obtain exact underlying values from the plot. Thirdly, when clustering gene expression data, each gene only appears once in the resultant heatmap view. Some of the genes may actually have similar expression profiles with genes belong to different pathways or modules; enforcing a single membership may lead to inaccurate interpretation. Fourthly, it's difficult to examine the relationships on a higher concept level – instead of investigating gene-to-gene or gene-to-metabolite associations, there are many research questions to be addressed from the raw pairwise data: how pathways or modules associate or interact? What's the intra-cluster variability and inter-cluster variability of a resultant clustering result? For replicate

[xxxi] http://www.r-project.org/

[xxxii] http://www.mathworks.com/products/matlab/

experiments, is there a way to examine all the raw values from several datasets interactively into a single view? Is there a way to visually compare a clustering result with regard to annotated pathways, or have several Ontologies (GO biological process and molecular functions) visualized simulatenously? In the previous NCI 60 analysis, we rendered a heatmap of gene-metabolite correlation with the molecules arranged in KEGG pathways groups. Even with only a few hundred rows and columns on each side, the heatmap was already quite difficult to interpret. It would be much more informative if we could obtain a summary matrix, that contains representative correlation values (such as max, min or median) at intersecting pathways. Then we only need to drive down into pathways that show strong correlation signals and identify the key molecules that drive such correlations.

In the following sections, I will first provide a brief history of matrix visualizations, followed by addressing how CoolMap can help to answer these research questions. The idea was originally formulated by my mentor Fan Meng, and then I evolved the idea from the features of conventional heatmaps. Many design concepts were inspired by best practices of popular software platforms and modern web-tools such as Cytoscape, Google Map and Photoshop.

4.2 The origins of key design concepts

Before diving into the details of matrix visualization, I provide a brief description of the evolution of the whole design process. The inception of Web 2.0 revolutionized the way people browse remote data - the experience has become more dynamic, interactive and fluid. People spent much less time waiting for the system to respond, and more time exploring and analyzing data using modern data visualization platforms. As the size and complexity of biomedical data continue to grow rapidly with the development new high-throughput technologies, it has become more and more critical to provide efficient and effective ways for biomedical researchers to explore these datasets and generate hypotheses. The first attempt was to try the idea of 'heterogeneous aggregation' along only one axis. The resultant dynamic genome browser (DGB) prototype developed back in 2008 brought the idea of 'heterogeneous zooming' of tracks. As the majority of genome browsers use one single zoom for all the tracks, DGB allows users to look at tracks with different zoom levels, with tracks containing more details such as SNPs or genomic sequences zoomed in much deeper

and tracks such as chromosomal bands only viewed at the overview level. The tracked are all synchronized at the center and scrolled at different speeds. In this way the user could examine certain regions in detail without losing the high-order concept context around. It was one of my earliest attempts to build data visualization applications with hierarchical view structures.

Another inspiration comes from Google Maps. It is very common to have adaptive rendering capability in Geographic Information Systems (GIS). At different zooming levels, the visualization provides different levels of detail to the user, from displaying only the name of a city to all the street-level view. Most current heatmap visualization is uniform at different zooming levels – each pixel is represented with a color regardless the sizes they take. With the introduction of ontology nodes, it is possible to offer more information to the researcher while the region of interest is zoomed in. For example, the exact numerical value that underlies a certain cell can be displayed instead of color when the size of the cell is sufficiently large. Pie charts, box plots or statistics can also be overlayed on Ontology node intersections. With these added capabilities, the heatmap can provide the user with much more contextual information to facilitate the analytical and discovery process.

4.3 A brief history of Matrix Visualization

There are many ways to represent a two or multi-way associative data. For very small datasets, it is often desirable to use a tree-diagram. The tree-diagram resembles a hash-to-hash data structure, and works well for unordered data. For larger two-way or three-way datasets, a table is more suitable. A two-way numeric table can be considered as a 'flattened' view of a three dimensional data, with each cell value indicating the z-position with regard to the corresponding xy position. For larger tables, it is generally more appealing to use certain visual cues to substitute numbers for more intuitive observations. For example, bars or shape sizes can be used to fill each cell[175]. More frequently, the cell values can also map to color schemes (continuous color gradient) or shade values (black and white). For example, Czekanowski employed a discrete mapping of five shading levels diagram to illustrate a 2D table[176]. Due to its intuitiveness and ease of implementation, heatmap representation has been implemented in a wide scope of disciplines, such as archaeology, cartography, sociology,

and psychology. It was not until Eisen's 1998 paper that this technique was applied widely to gene expression data[102], and it still remains the prime choice to visualize chip-like data. Many software packages have been developed to make static or interactive heatmaps, such as R (heatmap, gplots, ggplot, Bioconductor), Matlab, Gene Pattern, VistaClara, InfoViz, TreeViewer, JTreeMap, ClusterMaker, ConceptGen, BioSearch2D, etc[73,92,177–180]. However, many of these heatmap implementations are static that can't be interactively explored. It is also difficult to associate structured annotation data in these applications.

There are also several additional challenges to create a meaningful heatmap – the heatmap should not only function as a simple translation of a numeric space to a color space, but also unravel the otherwise inconspicuous structure of the data. To achieve this goal, the rows and columns should be permuted to maximize the distance of the most dissimilar entries and minimize the distance of the most similar ones. Due to the large number of possible permutations (n!m!), it is very important to use an efficient and effective algorithm. There are currently three major approaches: hierarchical (agglomerative/divisive) clustering, partitional clustering and row/column seriation.

Hierarchical clustering is a greedy step-wise algorithm to progressively merge/remove the most similar/dissimilar rows or columns, based on a certain distance criteria (single, complete, average, median, complete, ward, centroid). This algorithm is moderately efficient (depending on implementation, the performance can vary from around $O(N^2)$ or $O(N^3)$[181], and probably the most appealing feature is that it produces a hierarchical tree that clearly reveals the underlying grouping structure if the data is clearly bi-modal or contains a small number of modes. However, as almost any dataset can produce a hierarchical tree, the validity of this approach is often questioned if the clustering quality is unevaluated since the hierarchical tree may simply be an artifact of the algorithm rather than a true reflection of data structure. Moreover, as each left and right node of the tree can be flipped independently (total of 2^{n-1}), the resultant hierarchical tree may not best reflect the data structure. Some leaf-repositioning algorithms have been developed to optimize the node flipping[182], but are not scalable for large datasets.

The partitional algorithms, such as K-means and Model based iterative clustering, attempt to place entities into groups to maximize inter-group difference and minimize intra-group

difference. If the number of clusters is known *a priori*, this approach often out-perform the hierarchical algorithms in terms of cluster homogeneity. Sometimes the hierarchical and partitional procedures are even combined for better performance[183]. The problem of visualizing a heatmap using the result from a partitional clustering algorithm is that the best intra-cluster orders are unknown, which is trivial for functional inference but important for heatmap plotting. Therefore, some heuristics or other methods are needed to find an optimal intra-cluster order for heatmap visualization.

The matrix seriation (or reorderable matrix) methods are the least popular approaches in the Bio-domain. The main reason is possibly that in a typical chip analysis, it is more important to find a good cluster membership than to determine the optimal ordering of rows and columns of the heatmap. Furthermore, because most of the heatmaps in publications only plot a subset of selected gene expression profiles (or other signals), these sets usually represent the most drastic contrast across certain groups to provide the reader a visual cue of the degree or overall pattern, and the hierarchical clustering usually works fairly well. However, their effectiveness decays when the underlying data contain multiple small fuzzy groupings or other peculiarities. Seriation, by definition, is to find an ordering for an array of objects such that the position of the object reflects the relatedness to its neighbors. For example, to seriate the dissimilarity matrix for genes on an expression chip, the matrix would be permuted such that the least dissimilarity values are placed close to diagonal – an approximate anti-robinson matrix. The first documented formulation was probably from Petrie, where he used sorted matrix to represent tomb relic's data. Some algorithms have been developed, such as those listed here[184]. It was only until recently that some software packages have been made available, such as PermutMatrix[185], GAP[186] and the 'seriation' package for R. The major drawback for seriation algorithms is that they are not very scalable (could be $O(N^3)$ even $O(N^4)$), a combined clustering-seriation scenario has been proposed to only utilize seriation to refine the ordering within small clusters.

Some methods have been developed to optimize the display of the hierarchical tree. Treemap place the tree nodes in alternating rows and columns and the size of the compartment can be mapped to the tree node attribute values. However, it is still quite difficult to apply these methods to matrix visualization. Although it is possible combine various views and update them simultaneously to provide multiple aspects of the data to the

user at the same time, the scale and complexity of the current datasets may cause a 'mind' overflow of the user[187].

Even with proper permutation of rows and columns, the current heatmap visualization applications make little use of the clustering or ontology structure other than plotting along sides of the heatmap only for indication purposes. It is difficult to browse large heatmaps, sort a region of a heatmap, or associate structural data to rows and columns. Using gene chips as an example, the user may want to take advantage of gene-annotation information such as ontology or function, to only show genes linked to certain pathways or biological processes in a heatmap, or to place genes in two clusters side by side for comparison, or even draw a heatmap of pathways versus pathways from gene expression profiles. As described before, we have encountered such difficulties when analyzing the NCI60 dataset, and subsequently spurred the design of CoolMap.

4.4 The overall design of CoolMap

There have been some previous attempts of integrate multi-level tree into a heatmap typed visualization, but each of these implementations have deficiencies. The Matrix Zoom[119] and JTreeView[188] are both capable of plotting a hierarchical tree along the heatmap and the user may select a branch for in-depth view, but the interactivity of the tree diminishes quickly when the size of the heatmap grows. The user has no easy way to only investigate a branch of an ontology tree only at a certain level independently. There's no way to manipulate the tree nodes, such as to expand, collapse or hide certain nodes from the view while keeping the rest of the view regions in place. In addition, these trees can only be single inheritance trees, usually being a result from hierarchical clustering. As a result, it is impossible to visualize two ontology terms or pathways side by side, with shared child nodes. Yet this is a very common occurrence, as many pathways contain shared genes and metabolites. Furthermore, the width and height of the heatmap cell in all these applications can only be changed using a single zoom multiple, which makes it difficult to examine heatmaps with very large number of rows (genes) and very small number of samples (columns). As a result, the majority of the heatmaps only provide a very coarse overview of the underlying data

characteristics. We believe by integrating ontology trees and agile/flexible rendering pipelines, the capability of heatmaps can be significantly enhanced to explore current datasets.

We developed the CoolMap specifically to address the above issues. Table 4-1 Feature comparison of CoolMap with some other Tools

	R(gplots-heatmap.2)	JTreeView	MatrixZoom	CoolMap
Interactive	No	Yes	Yes	Yes
User-rearrange row/column order	No	No	No	Yes
Rendering other data-types	No	Yes	No	Yes
Rendering other than color	No	Yes	No	Yes
Annotation overlays	Yes	No	No	Yes
Clustering	Yes	Yes	Yes	Under-development
Node Expansion/Collapse	No	No	Yes	Yes
Multiple Ontology	No	No	No	Yes
Multiple Plot Link	No	Yes	Yes	Under-development
Extract Sub-portion	Programmatically	Yes	Yes	Yes
Search/Filter	Programmatically	No	No	Yes

Figure 4-1 illustrates the basic concept of CoolMap. Suppose rows and columns of a matrix can be aggregated with an ontology tree, and the node of the tree then can be folded bi-directionally. The resultant cell can then be replaced with a summary value using a compatible aggregation function, such as the mean, median, quantiles, or variance. The ontologies can be a result of hierarchical clustering, or user defined groups, pathways, networks, molecular ontologies or phylogenetic trees exported from databases. Using this design, it is then possible to view the data at a higher concept level. The details can then be shown contextually via overlays, or expansion of the node pairs of interest. A significant amount of peripheral functions were also developed to support the interactive visualization of CoolMap. The following list summarizes the main features of CoolMap:

Major features:

- Capable of exploring 2D data with arbitrary format. Due to the extensible interface, CoolMap potentially can render any type of data with a compatible data loader, aggregator and renderer. The current release only supports numeric data, but can be easily extended to string, character, boolean, image or even composite data types.

- An ontology with arbitrary structure, but without self-loops, can be loaded. An ontology term is a group term that contains an arbitrary number of rows or columns from the base matrix. There can also be hierarchical structure between the ontology terms (such as Gene Ontology or intermediate nodes in the result of hierarchical clustering).

- Rows or columns can be added, removed, or reordered manually or programmatically. For example, the row and column order from matrix seriation can be used to reorder the CoolMap in view. The ontology nodes can be mixed with base nodes in the same view for side-by-side comparison. Ontology nodes can be expanded to reveal ontology terms on the lower level.

- The CoolMap in view can be dragged around; hovering the mouse near a cell immediately shows the underlying value in the matrix; the size of each cell can be adjusted using the resizing grid so that cells of interest can take more space and reveal more details.

- A renderer can be assigned to determine how the view matrix is presented. It is very easy to extend the renderer interface with a variety of visualization methods. Right now the renderer supports color mapping and size (bar).

- An annotator can render the base rows and columns associated with a matrix on top of the renderer layer. This interface can also be extended to render the underlying group of cells as a variety of graphics (boxplots, scatter plots, etc).

- It's possible to use several criteria to filter the CoolMap with filters. Cells in view that pass the composite criteria will be shown. Others will be masked out from view. The user may use numeric or fuzzy string filters. Multiple filters can be joined together using logic AND or OR.

- The user may use the Ontology browser, and ontology table to search for ontology terms of interest, inspect the hierarchical structure about that term, and then add selected terms to at any row/column locations in view.

- A bridge to R is available to both the user and plugin developers to incorporate R functions, such as hierarchical clustering, correlation, k-means, seriation, etc to CoolMap.

More details can be found in the appendix. Figure 4-2 shows a screenshot with a majority of the CoolMap widgets. A feature comparison is listed in Table 4-1.

4.5 Application of CoolMap to Exploratory Data Analysis

4.5.1 Nutrition experiment with multiple categorical designs

To illustrate how CoolMap could facilitate the exploration of microarray data, we tried to replicate the exploratory process from a nutrition study targeted at identifing how saturated fatty-acid rich diet could trigger obesity-linked proinflammatory gene expression in adipose tissues[189]. The sample consists of 20 abdominally overweight subjects and they received Saturated fatty acid diets (SFA) or monounsaturated fatty acid rich diets (MUFA) over an

eight-week span. The samples were categorized into factors such as the experiment protocol, the gene expression profiles were analyzed and the authors discovered that consumption of SFA diet triggered elevated expression of genes involved in inflammation processes in adipose tissues, whereas the MUFA diet led to anti-inflammatory gene expression fingerprints. We therefore used their dataset to examine how exactly the change of expression profiles can be visually explored. The experiment protocol, time, individual and gender can all be used as group ontologies in the view.

Figure 4-3 shows the overall expression well-known immune related genes as shown in Figure 3 of the original paper. By using CoolMap, it's possible to access both gene-level expression and probe level expression, which is not immediately possible before. From the comparison figure we can see both gene IL6R and CD209 have two probes in the probeset. While the expression levels of IL6R probes are quite similar, the probe expression of CD209 are quite different (~3 for NuGO_eht0340260_at and ~8 for 207277_at). As usually the average expression value across all probes in a probeset is used as the gene expression value, the overall value for CD209 may be problematic in this case. Potentially for a gene that could be mapped to a larger number of probes, it would be helpful to examine the individual probe expression values when examining filtered genes before further functional inference. This demonstrates the capability of CoolMap to be used for quality control when a molecular profile can be mapped to several child profiles, such as time series, technical replicates, protein or gene families, etc.

Figure 4-4 shows the exploration of gene expression change in SFA and MUFA diet groups. The original paper used heatmap[189] to illustrate the increased expression of immune related genes. The same set of genes were exported from the GEO data files (some genes were unable to be found, possibly due to the change of probe mapping annotations) and illustrated in CoolMap. By using the experiment groups, it is possible to examine the change of gene expression on the SFA and MUFA level instead of each individual gene. Overall by using mean and median expression fold change, the SFA group indeed shows overall increased expression in immune related genes. B,C,D. Meanwhile we found that different from the original heatmap, CD209 did not show elevated gene expression in the SFA group. As CD209 is mapped to two probes, expansion of CD209 reveals that the probe 207277_at shows overall elevated expression in SFA group while NuGO_eht0340260_at showed little

change in gene expression over all samples. It is then likely that the NuGO_eht0340260_at could be a probe with low sensitivity (also shows that this probe has low expression value comparing with 207277_at), or it could be a wrong mapping.

These two examples together demonstrate that CoolMap is capable of replicate heatmap functions, and yet provide more insights on higher level (experiment groups) as well as lower level (individual probes) interactive explorations, which were not possible to be done with existing heatmap-based applications.

4.5.2 Analysis of Mother-Child Nutrient/Epigenome Interaction

Another test study of CoolMap is on a collaborative research project with Dr Charles Burant, aiming at understanding how mother/child nutrient transfer and epigenetics could affect the development of the child. Mother-Child placental transfer of nutrients, such as fatty acid, is critical to fetus's prenatal and postnatal growth. We collected a dataset with targeted serum metabolomics profiling and methylation profiling. There are two major research questions we would like to address: first of all, how does metabolite levels and gene methylation correlate between mother and child? And more specifically, how the genetics change in mother lead to the epigenetics change in child, and how metabolite (diet) level in mother across placenta transfer eventually lead to the epigenetics change in child? Also What are the predicted changes in gene expression, and subsequently the potential changes in the child's metabolism? There are several key genes we would like to explore, such as the Insulin-like growth factor 2 (IGF2 http://www.ncbi.nlm.nih.gov/gene/3481) which affects development and growth, Peroxisome Proliferator-activated Receptor Alpha (PPARa http://www.ncbi.nlm.nih.gov/gene/5465) that regulates lipid metabolism and Estrogen Receptor Alpha (ESR1 http://www.ncbi.nlm.nih.gov/gene/2099) that also regulates development. The gene methylation profiles were collected from blood samples from 15 mothers (in the first trimester) and maternal and infant cord blood collected during delivery. Directed metabolomics analysis was performed to quantify serum amino acids, acyl-carnitines and the total plasma fatty acids from both the mother and the child. Ontology headers are also created to utilize CoolMap for multi-level exploration of the datasets (Line 1 methylation, ESR Methylation sites, IGF2 Methylation sites, PPARa Methylation, BCAA

AcylCarnitines, MediumChain Acyl Carnitines, Long Chain AcylCarnitines, Aromatic Amino Acids, Saturated Fatty Acids, Monounsaturated Fatty Acids, PUFA, etc.)

In order to investigate the association between the profile of these biomarkers between the mother and the child, we computed the Pearson Correlation Coefficient on these data along with a number of clinical parameters, including the preconception weight, weight gain during pregnancy, term fetus weight, among others. Data were (z transformed; in case of missing values, only complete observations were used to compute correlation). The resultant correlation matrix was also clustered to find close associations. By using CoolMap, instead of browsing the entire dataset, we could of first identify which groups of biomarkers that have high correlation between the mother and the child, then dive-in to further identify the members in a group that drive the high correlation. Filtering, annotator and a combination of operations will also facilitate the knowledge discovery process. **Figure 4-5** shows the aggregated group view makes it much easier to understand the overall trend of the data than the heatmap drawn at the base level. From the aggregated view, we can immediately identify the highly correlated mother-child measurement groups, such as BCAA AcylCarnitines (0.45), Long Chain AcylCarnitines (0.34), PPARa methylation (0.52), ESR Methylation (0.32) are highly correlated overall between mother and child.

The next step is to identify the diving forces of these high correlations within a group (Figure 4-6). Overlaying a boxplot on PPARa shows that the spread of correlation across PPARa gene methylation (mean 0.52) is quite small. This implies that the methylation profiles between mother and child is strongly correlated, and indicates the genetic factor on the correlated gene expression profiles. Using the same method on BCAA AcylCarnitines (0.45), we can see that the spread of correlation is much wider, and because of the C4 AcylCarnitines, the overall correlation was brought down much lower. The mean methylation correlation between ESR mother and child is much lower, only 0.32. The boxplot overlay shows that the correlation spread is quite wide. Expanding the corresponding nodes to the child level reveals the heterogeneity of site-specific methylation correlation; with site-3 have a much higher correlation (0.838) than the other members. The ESR methylation site-1, on the other hand, has a moderately high correlation with mother valine (0.42), which implies a potential metabolic factor to gene expression profiles. By use the other search and filter functions of CoolMap, we could identify other highly correlated

pairs within an ontology/group context (Figure 4-7). By using multi-level CoolMap exploratory analysis, we could identify interesting signals in the data from a high level of view, and then investigate the lower level details to mine for more details from the data for hypothesis generation. The identified highly correlated methylation / metabolite pairs can lead to further targeted experiments and pathway/functional module elucidation.

4.6 Other Applications

4.6.1 Data Quality Control

As described in chapter 3, many omics data may contain abnormally distributed data, missing values, or extreme outliers. It is then very helpful to quickly investigate the input data and identify the number, distribution and pattern of missing values. shows the CoolMap visualization of baseline metabolite measurement from the investigational weight management clinic. We can see that there are quite a few missing values for baseline 11,13 and 14. Furthermore, the glutamine level seems to be way too high. Quickly adjusting the range of color mapping will better reflect the details in low value regions.

4.6.2 Data with large amount of missing values

List typed data with each row containing an uneven number of elements or sparse matrices (for networks) with a large number of missing values can be difficult to visualize using traditional heatmaps. Conceptually, even if there are missing values in a row or column group, it should still be possible to obtain representative or summary values that would drastically reduce the number of missing values. Illustrates and example of using CoolMap on DNA methylation data. This data was obtained from four squamous cell carcinoma cell lines (two positive for human pappilomavirus HPV, and two negative for HPV) using the illumine HumanMethylation27k BeadChip platform[190]. By building ontology groups for methylation sites, the original matrix view can be translated into a much more condensed view. Expansion of methylation groups shows the number of methylation sites for each gene and the values. It can be seen that some genes such as CDKN2A and CDKN2B have many more methylation sites than others, using the average would eliminate the intra-group

heterogeneity and the possibility of studying which site is actually the key that drives the underlying process.

4.6.3 Sequence Analysis

As CoolMap can visualize data types other than numeric values, we also developed a renderer variant that can be used to explore sequence alignment and motifs. Transcription factors usually bind to a sequence in a sequence specific manner, and the resultant binding site are usually represented as consensus sequences, position weight matrices or sequence logos[191]. CoolMap can be used to display the consensus sequence using the IUPAC degeneracy letter, while still preserve the underlying individual sequence information. Illustrates an example of CRP binding site taken from Schultz et al[192]. The consensus sequence can be aggregated at different levels to illustrate the variability of conservation. Base and GC content can also be overlaid on top of the sequence logos for underlying base distribution details.

4.7 Summary

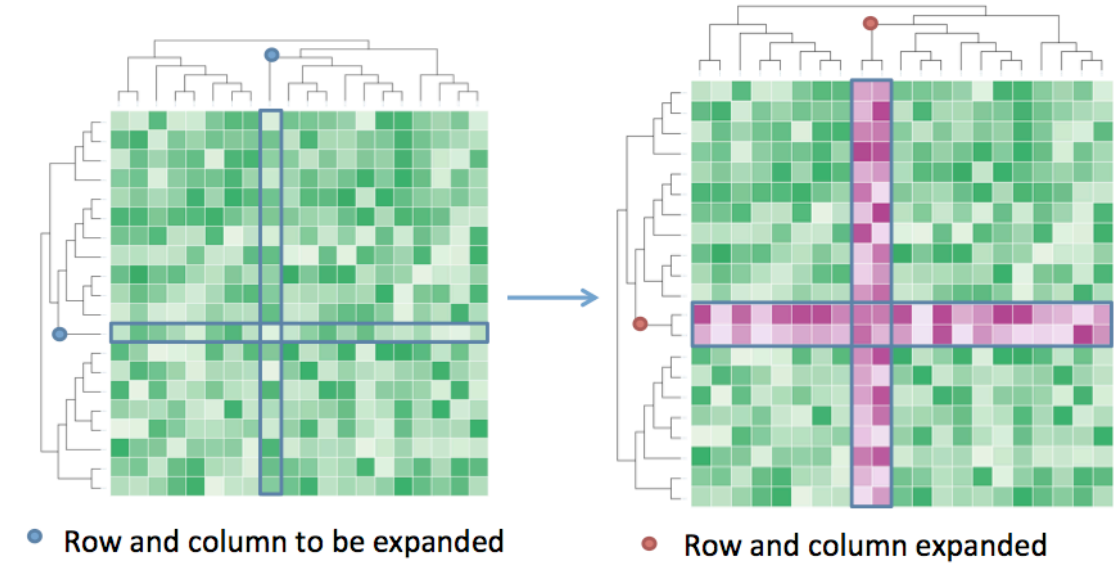
CoolMap extended the visualization capability of traditional heatmap by incorporating ontology structure along both axes. From the usage examples we demonstrated that CoolMap is capable of rendering multi-scale information to the researcher for rapid exploratory analysis and hypothesis generation. The ontology-aided data aggregation can help researchers understand the signals of the data using structural knowledge. We hope that with this new heatmap-based model, and the versatile rendering options, streamlined user-interface and a slew of auxiliary functions, CoolMap will improve the data-driven functional interpretation process.

Table 4-1 Feature comparison of CoolMap with some other Tools

	R(gplots-heatmap.2)	JTreeView	MatrixZoom	CoolMap
Interactive	No	Yes	Yes	Yes
User-rearrange row/column order	No	No	No	Yes
Rendering other data-types	No	Yes	No	Yes
Rendering other than color	No	Yes	No	Yes
Annotation overlays	Yes	No	No	Yes
Clustering	Yes	Yes	Yes	Under-development
Node Expansion/Collapse	No	No	Yes	Yes
Multiple Ontology	No	No	No	Yes
Multiple Plot Link	No	Yes	Yes	Under-development
Extract Sub-portion	Programmatically	Yes	Yes	Yes
Search/Filter	Programmatically	No	No	Yes

Figure 4-1 the basic concept diagram of the CoolMap.

In the CoolMap implementation, the rows and columns of a data matrix can be aggregated (collapsed and expanded using a designated aggregation function). Top: illustrates the basic concept. A row and column at any aggregation level can be collapsed into a summarization view, or expanded to show the underlying details. The bottom figure illustrates the organizational flow of CoolMap: in the first step, the data objects in the raw matrix are aggregated into the view matrix with a designated aggregator. The aggregator translates raw data type R into view object type V. The renderer then renders the view matrix into a graphic CoolMap representation. An example flow: the raw data matrix contains gene expression values in double numeric format. The view matrix contains gene ontology groups and the aggregator translates the raw double numeric format into mean log2 transformed expression values. The number to color renderer finally converts the resultant view matrix into a heatmap representation.



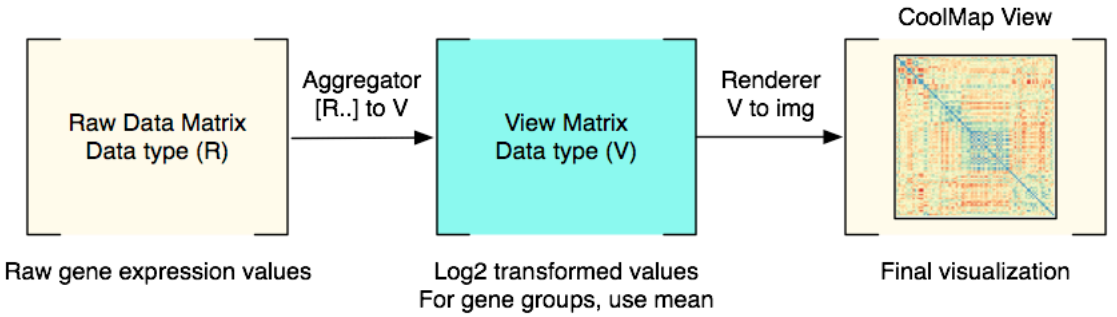


Figure 4-2 CoolMap screenshot.

A CoolMap screenshot lists the majority of of currently developed modules, including CoolMap listers, filters, ontology browser, annotation renderer, color renderer, data browser table, etc.

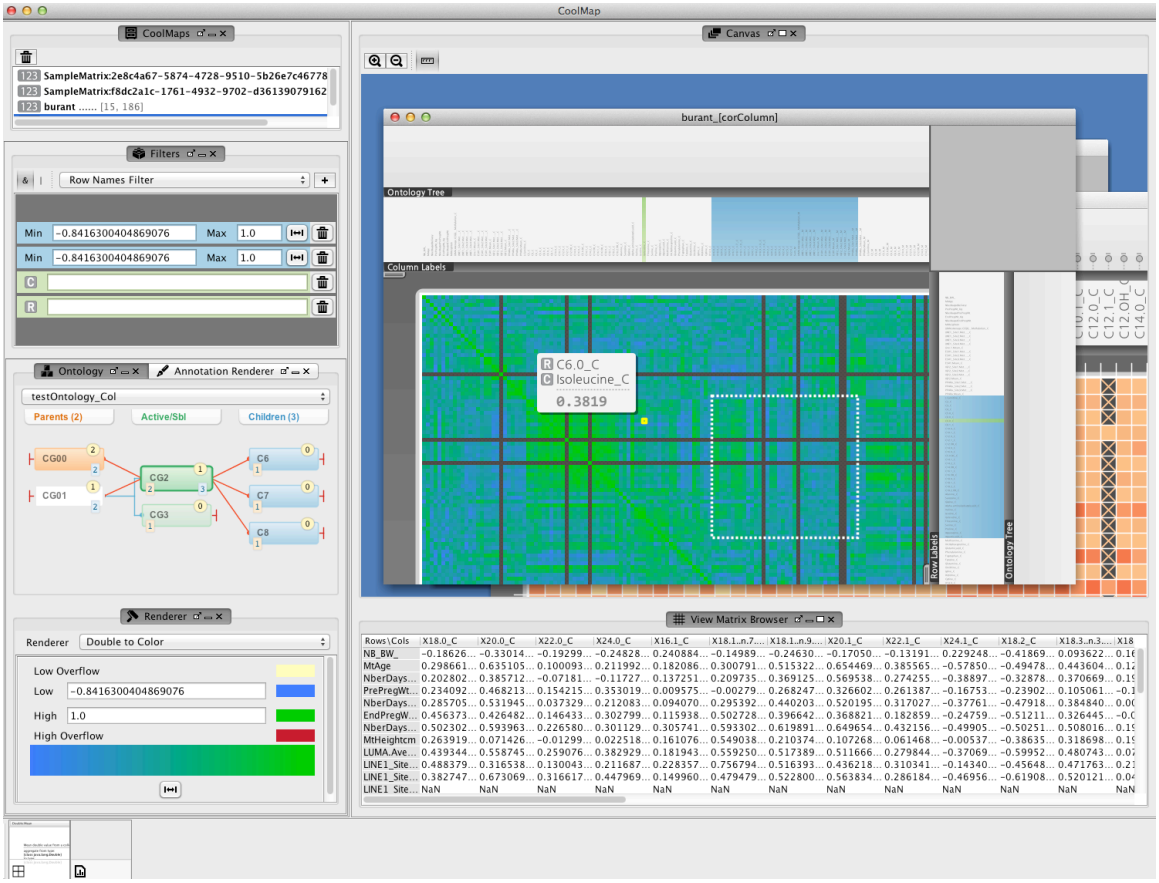


Figure 4-3 Quality control of the GDS3678 dataset from figure 3 in the original paper.

The genes in figure 3 of the original paper were arranged in CoolMap view. Several genes, such as HLA-DQA1, PPARg, HLA-DMA, etc., are missing. The majority of the probes are 1 to 1 match to gene annotations. There are two genes that have 2 probes in the probeset, IL6R and CD209. After expansion, it can be seen that the two probe expression of IL6R (yellow rectangles) are quite uniform. However, the two probes of CD209, 207277_at and NuGO_eht0340260_at, have very dissimilar expression profiles, with 207277_at ~ 8 across all samples and NuGO_eht0340260_at ~3. If the resultant expression value is taken as the average, it could be worthy of concern. This demonstrates that CoolMap could be utilized to investigate raw data at different hierarchies. The two groups on the horizontal axis denotes Saturated Fatty Acid (SFA) group and Mono Saturated Fatty Acid (MUFA) group, respectively.

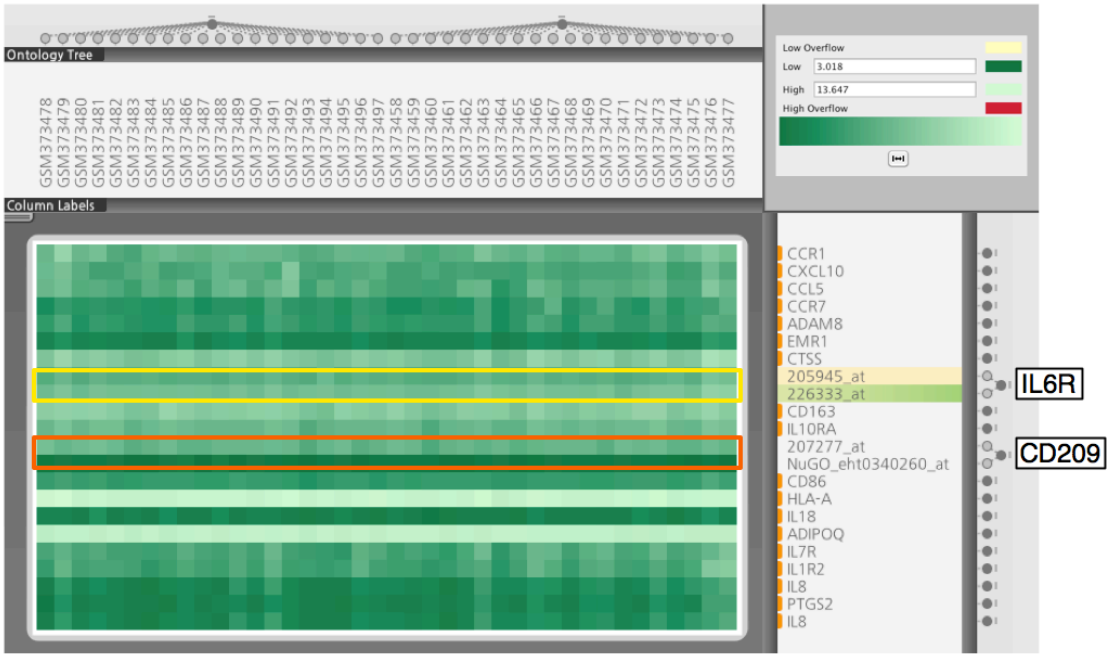


Figure 4-4 Illustration of elevated expression of immune related genes in Saturated Fatty Acid (SFA) diet set.

The original paper used a heatmap to illustrate the elevated gene expression in SFA set over an eight-week diet span compared with the MUFA set (A). The value is mapped from -0.5 to 0.5 for fold change, from green to red. We illustrated the same set of genes (some can't be found in the downloaded dataset, probably due toe the change of probe definitions) using CoolMap (center). The samples can be aggregated using the experiment design (http://www.ncbi.nlm.nih.gov/geo/gds/profileGraph.cgi?gds=3678) . In the aggregated view, it's much clearer from the mean or median fold change that the immune related genes do show elevated expression. Color is coded from orange to blue, from -0.5 to 0.5. A: Original heatmap visualization. B: CoolMap replication with matching genes. C: aggregated mean. D: aggregated median. The SFA group demonstrates elevated overall gene expression. E: CD209 did not show marked elevation of expression with the GEO data annotation, expansion of CD209 shows that the probe 207277_at shows elevated expression in SFA and is consistent with A. However, the NuGO_eht0340260_at didn't show any marked expression. Therefore this probe is either a wrong match, or has very low sensitivity.

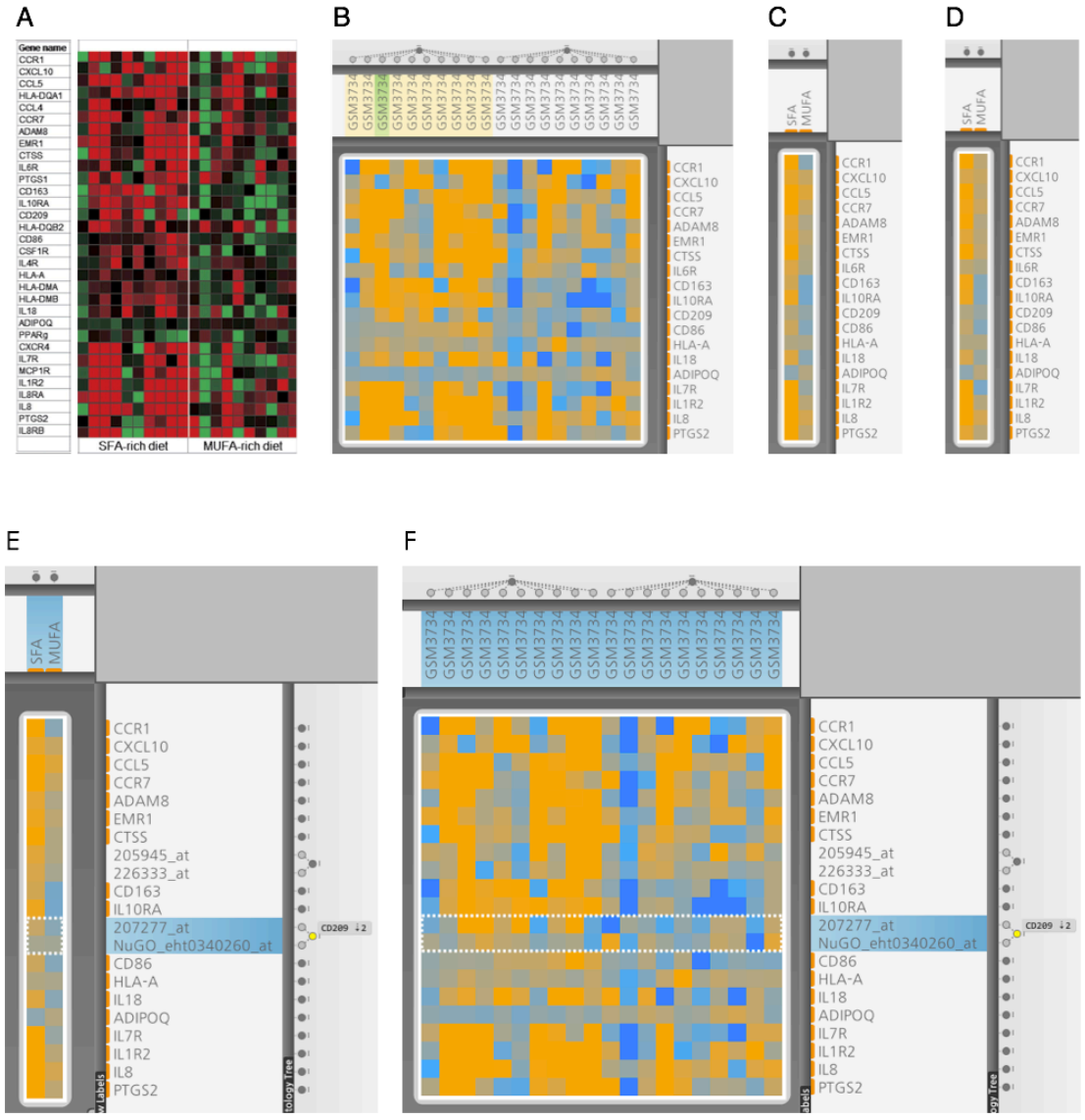


Figure 4-5 Condense the raw data into methylation / metabolite groups.

A: illustrates the efficient usage of ontologies to reduce the original pairwise correlation matrix into a condensed and manageable group view. From the condensed view we can identify the highly correlated groups, such as BCAA acylcarnitines (0.45), long chain acylcarnitines (0.34), PPARa methylation (0.52), ESR methylation (0.32).

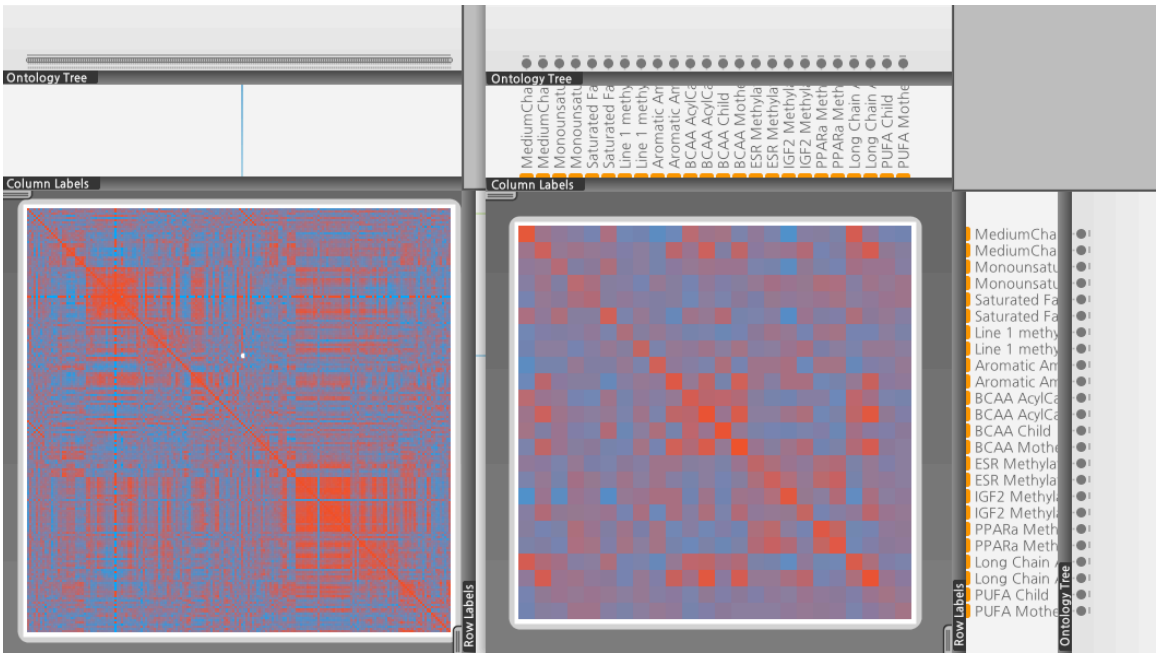


Figure 4-6 Multi-level exploration of highly correlated mother-child pairs.

Top row: the PPARa mother-child correlation. The boxplot overlay shows that the overall correlation is quite high. Center row: the BCAA-acylcarnitines mother-child correlation. The spread of data is bigger, and the C4-acylcarnitine mother-child correlation is quite low. Bottom row: the ESR1 boxplot and expanded views illustrate the methylation site heterogeneity.

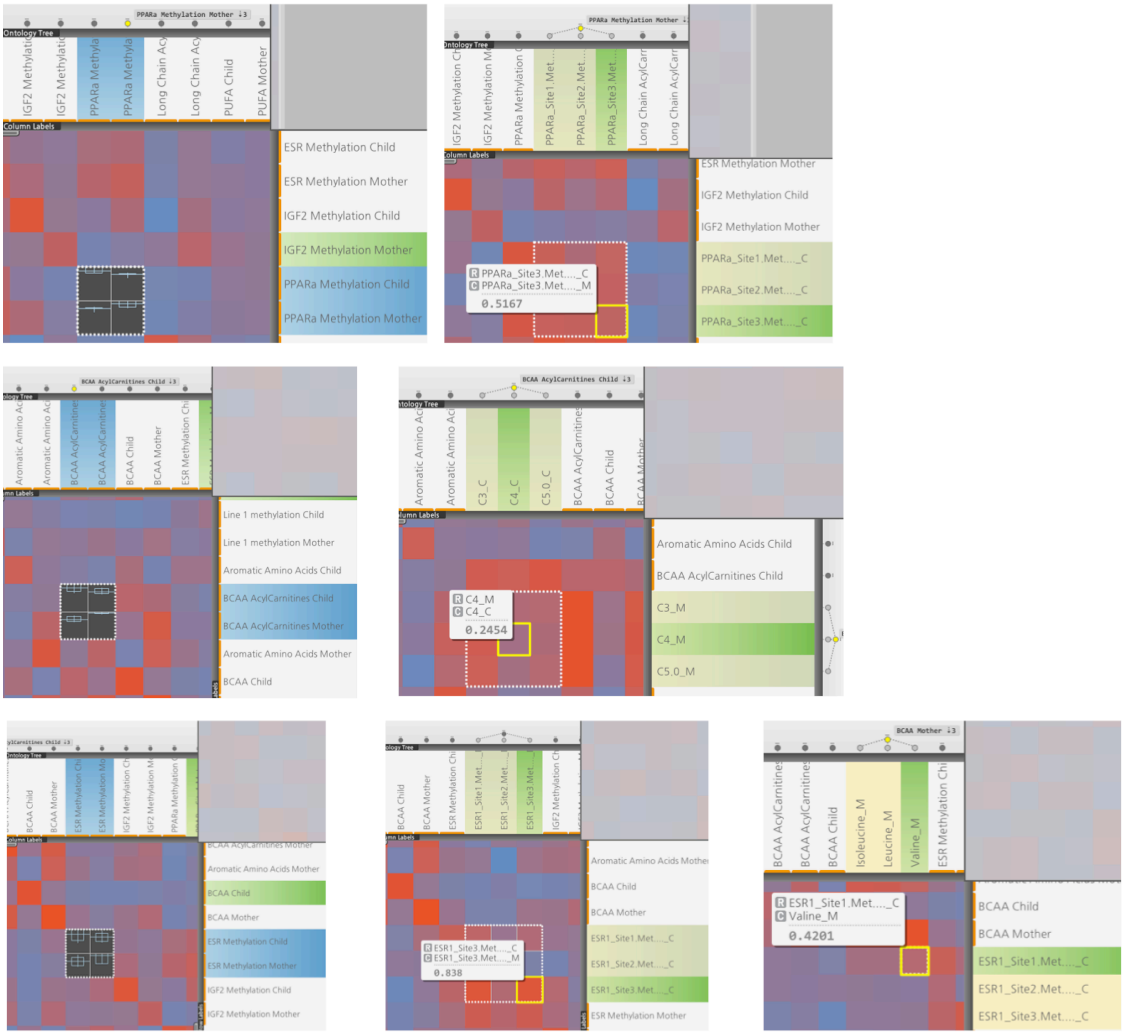


Figure 4-7 Using CoolMap for interactive knowledge discovery.

A: Using filter to remove low correlations out of view would significantly reduce the exploration space (shows only correlation between 0.5 – 1.0). The highlighted rectangle shows high mother to child correlation with X20.4. B: Even though there are strong correlations of LINE1 methylation between mothers and children samples, there are no strong correlation of LINE1 identified between mother and child. Further expansion shows that there's indeed no strong correlation signal. C: ESR1 methylation has strong correlation between mother and child, and it can be seen the ESR1 site 3 is the main driver for the strong correlation.

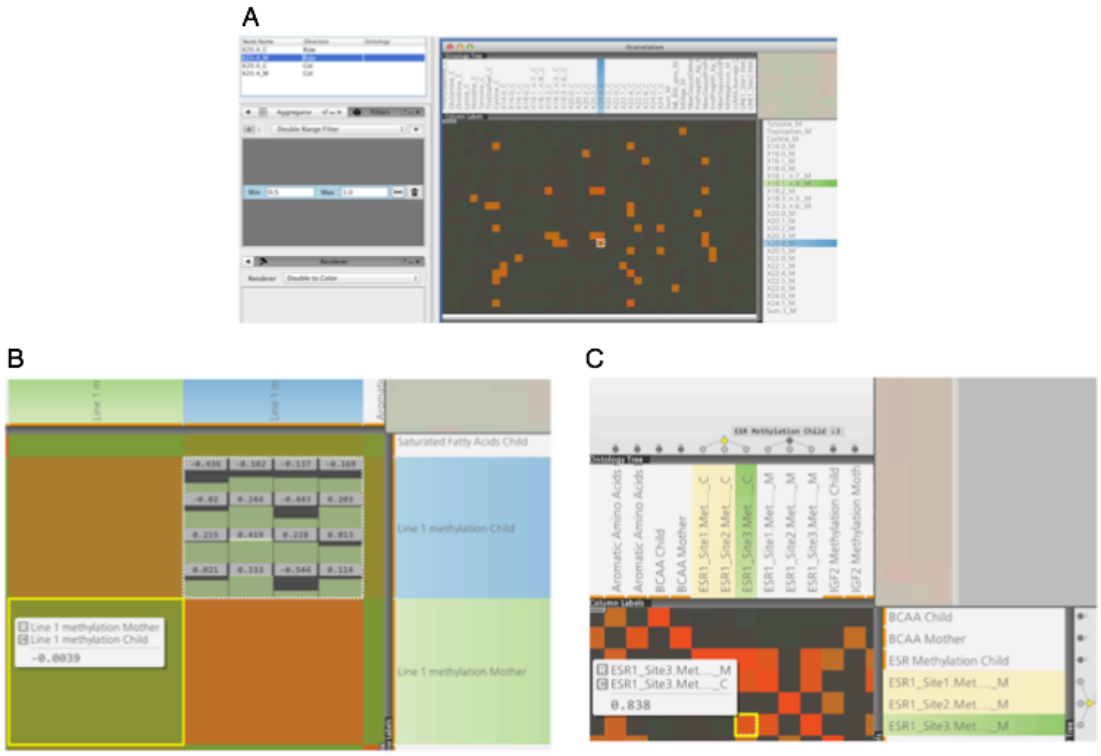


Figure 4-8 CoolMap data quality inspection. We can quickly identify CoolMap regions with missing values or other peculiarities. Top: The color scale yellow to orange is mapped from 0.0 – 1087.92. Note that the center column of glutamine, has values much higher than other metabolites. Bottom: adjusting the color mapping from 0.0 – 100.0 to yellow – orange would reveal more details in the low value regions.

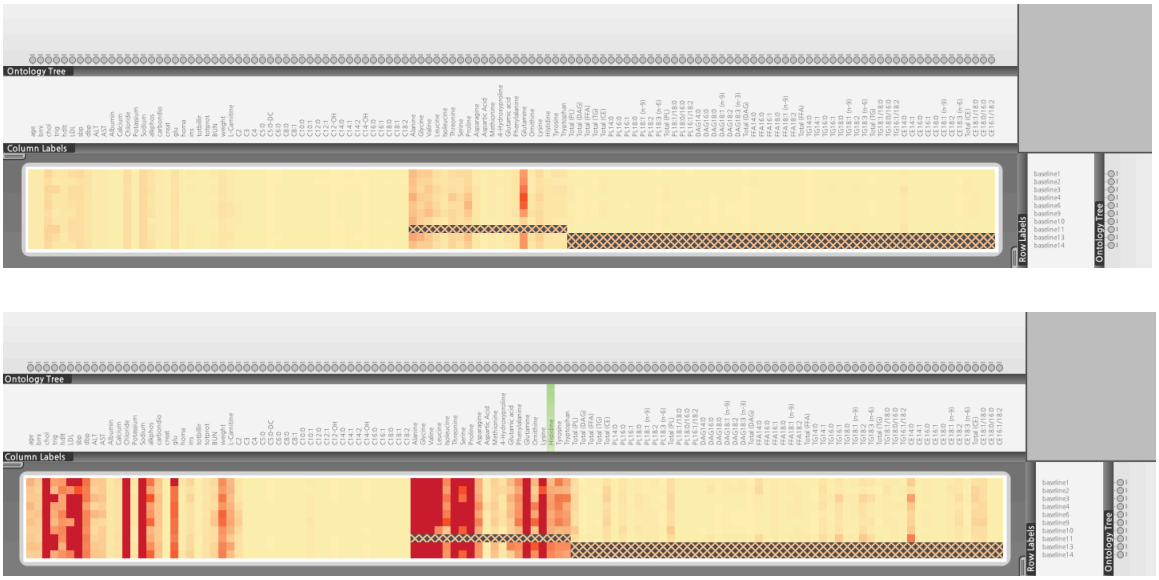


Figure 4-9 Illustration of CoolMap on unpublished methylation data from Maureen

A illustrates the average methylation values and expression values condensed by sites, with Caski.1 and Caski.2 expanded. B illustrates expansion of four methylation groups. Because genes have different number of methylation sites, it's usually difficult to illustrate such list-typed data using heatmap. Using only the average value from all sites may be biased for gene with a large number of methylation sites. Expansion of methylation groups shows CDKN2A and CDKN2B both have around 10 methylation sites. The scale is green (-2.0) to red (2.0). Orange dotted regions indicate missing values.

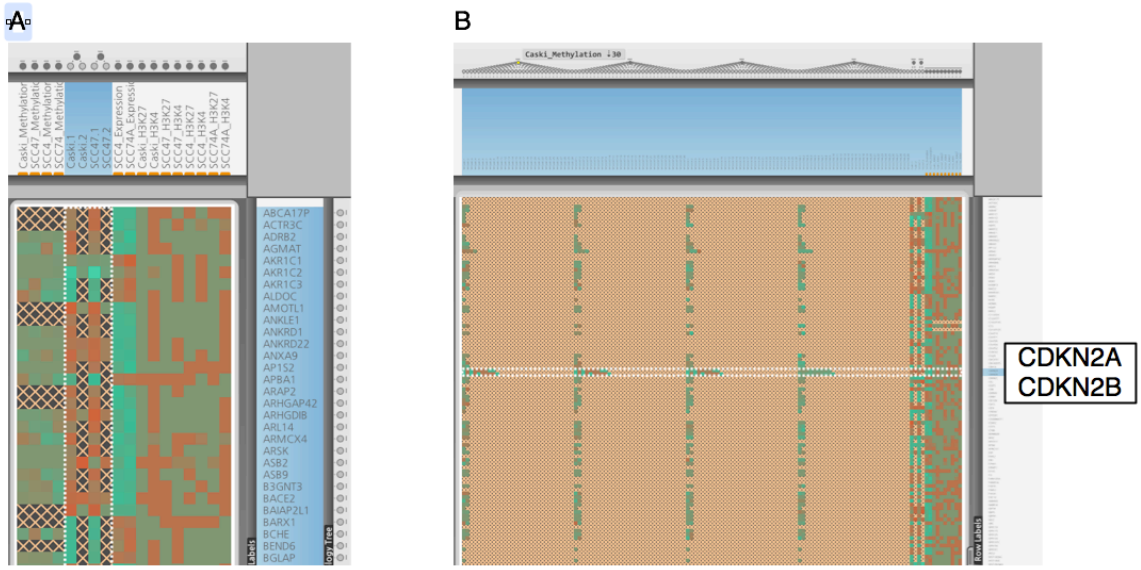
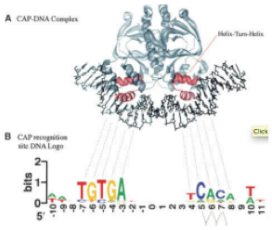
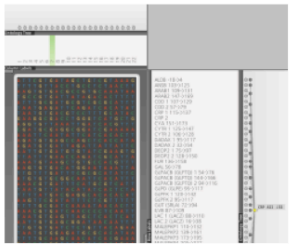


Figure 4-10 Illustration of using CoolMap for sequence analysis

Top-left: a sequence logo that illustrates the CRP (Catabolite Activator Protein) binding site, 49 sequences. Bottom-left: the fully expanded view of all the sequences, T(red), G(yellow), A(green) and C(blue). Top-right: the sequences are now collapsed into the CRP family, rendered using degenerate letters following the IUPAC rule. Note that the conserved pattern of TGTGA … TCACA is shown, with additional WWs on each side. Center-right: the base percentages are overlaid as an annotation. Center-bottom: GC content is shown.

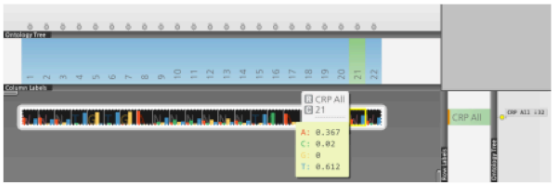


Sequence Logo

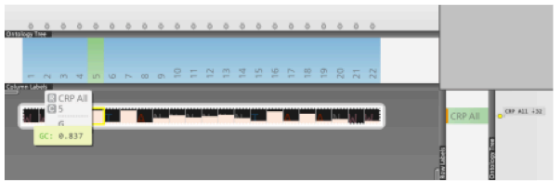


Full Sequence View

Consensus View



Consensus View with base percentage overlay



Consensus View with GC content overlay

Chapter 5 Conclusion

In the course of my PhD research, I began by making incremental changes to existing exploration models for Omics data. I developed the GLay plugin for Cytoscape to extend network clustering capabilities with a variety of community structure detection algorithms, GSearcher to offer flexible, intuitive and fuzzy search tools for filtering nodes or edges of interest from big attribute tables, Node Filter to interactively explore networks with node connectivity and expand networks from biological concept interaction database. I developed a novel workflow, using robust statistical methods coupled with classic methods to analyze Transcriptomics-Metabolomics data with multivariate outliers and a high level of noise, to identify new metabolite-metabolite and gene-metabolite relationships in NCI 60 transcriptomics-metabolomics data, which could be used for subsequent characterization of unknown metabolites and biomarker discovery. In this process, I began to realize that the growing Omics data size and complexity made it difficult to continue to use existing models for effective visual exploration. Therefore with the guidance of my advisor Fan Meng and the thesis committee, I designed a novel model that facilitate data-driven exploration of omics datasets that is scalable to the growing sizes of datasets in the foreseeable future. After about two years and over five complete overhauls, we came up with the CoolMap, in hopes that it would allow researchers to make better use of the enormously information rich omics data.

Where's the future of omics data exploration and analysis? With the tabletification of terminal computation structure and permeation of cloud storage, we could imagine the following change: computational model will regress to its early master-slave model. The client side will be lighter, with less computational power but bigger, more versatile screen estate (high resolution multi screen), more flexible user interactions (touch interfaces) and connectivity (wireless), and the server side will be cheaper but bigger (cloud, rentable servers), it would be expected that a large amount of Omics data will be stored in online repositories using more normalized formats, and heavy-duty statistical analysis, such as network inference, enrichment analysis, clustering and classification will all be done remotely in batch

and parallel. The analysis cycle will be much more efficient - the users will spend less time conducting local analysis, but more time analyzing results. It will also more efficient to collaborate online, and use reproducible workflows. Many open source software groups such as the developers for Cytoscape already use Google Hangout for lightweight teleconferencing and git for community development and code deposit. This model could easily be extended to other collaborative online research projects.

With this ongoing trend, it is then critical to develop novel and intuitive visual exploratory analysis tools for the future datasets. The heterogeneous view design of CoolMap enables users to use a single view to investigate data at different aggregation levels. With the continuous improvement of measurement resolution, it would become impossible to manually investigate every piece of detail of a largess dataset. Having rapid search functions, efficient and robust statistic methods and visualization that can constantly adapt to the focus of the analyzer and provide the most relevant contextual information will be critical to the analysis of future omics datasets. The visual exploration model of Cytoscape for biological network analysis has become a paradigm[193]: the user could switch between a global overview of the entire network structure and the detailed neighborhood view of a gene within a sub-network representing a protein complex or a pathway; the network statistics can be computed using NetworkAnalyzer[194], the external link tools such as the Agilent literature search plugin can be used to retrieve external annotations of a gene or protein on the fly via one click[195]; the current gene or protein in a network can be expanded using MiMI or Metscape for inspecting the interaction pattern[35,49]; external reference networks or pathways can be imported directly from databases such as Reactome or KEGG[34,47]. The researcher has all the tools needed and these functions can be accessed conveniently and contextually. This loosely coupled 'core framework' + 'peripheral services' model is proven its effectiveness for visual exploratory analysis applications.

Nevertheless, network representation is a qualitative approximation of the actual quantitative interactions detected in omics data. The construction of a co-expression network from omics datasets as proposed by many previous researches[165,196–199] all requires a 'cut-off' – a certain value that indicates the qualification of an interaction edge. If the distribution of pairwise distances is very flat or multi-modal, a slight change in cut-off may drastically change the number of edges in the network, which would subsequently affect the overall

topology of the resultant network clustering of networks modules. Furthermore, the strength of pairwise association can be much better reflected using a heatmap view than in a network view, in which the pairwise association is usually visualized using the thickness of the edge and/or color + transparency, which quickly becomes extremely difficult to interpret with massive number stacking edges when the network grows to more than a few hundred nodes or edges. One reason of using network to interpret the quantitative molecular interactions within an Omics dataset is the lack of suitable heatmap-based visualization methods. With CoolMap's hierarchical view capabilities it is possible to visualize the interactions between higher order concepts such as Gene Ontology terms (cellular component, biological process, molecular function), molecular pathway modules or clustering results. Ontologies are also becoming increasingly important in biological knowledge mining. It helps reorganize the data in structured way and in our case, help the users normalize data from different sources understand the data across multiple scales, which is especially important for interactive exploration of large-scale data. Software programs like protégé can be used to build, edit and explore ontology trees. As the data size and complexity continue to grow, barely flatten out big data on the base variable level would be difficult for humans to comprehend. Using ontologies in CoolMap has demonstrated its superior capability of multi-scale visualization. Once an interesting overall intra-high-order-concept is identified, the details can be inspected in detail by expanding the corresponding nodes. In this way all the quantitative data are preserved and can be accessed at anytime during the analysis. In the future more capable ontology loading, remote retrieval, generation and editing functions will be incorporated into CoolMap.

It would also be very beneficial to integrate both the network view (for simplified interaction landscape of the omics dataset) and the CoolMap view (for pairwise interaction details at various hierarchy levels). Similar applications have been developed for traditional heatmaps[92,178]. We developed one prototype, 'heatnet', to illustrate the efficacy of such approaches using CoolMap.

For future development, we envision that the CoolMap model can be implemented for general genome data browser. The current genome browser models such as UCSC[8] and Ensembl[11] were developed more than a decade ago. The view can be changed at different 'zoom' which represents aggregation levels along the genomic axis. The UCSC genome

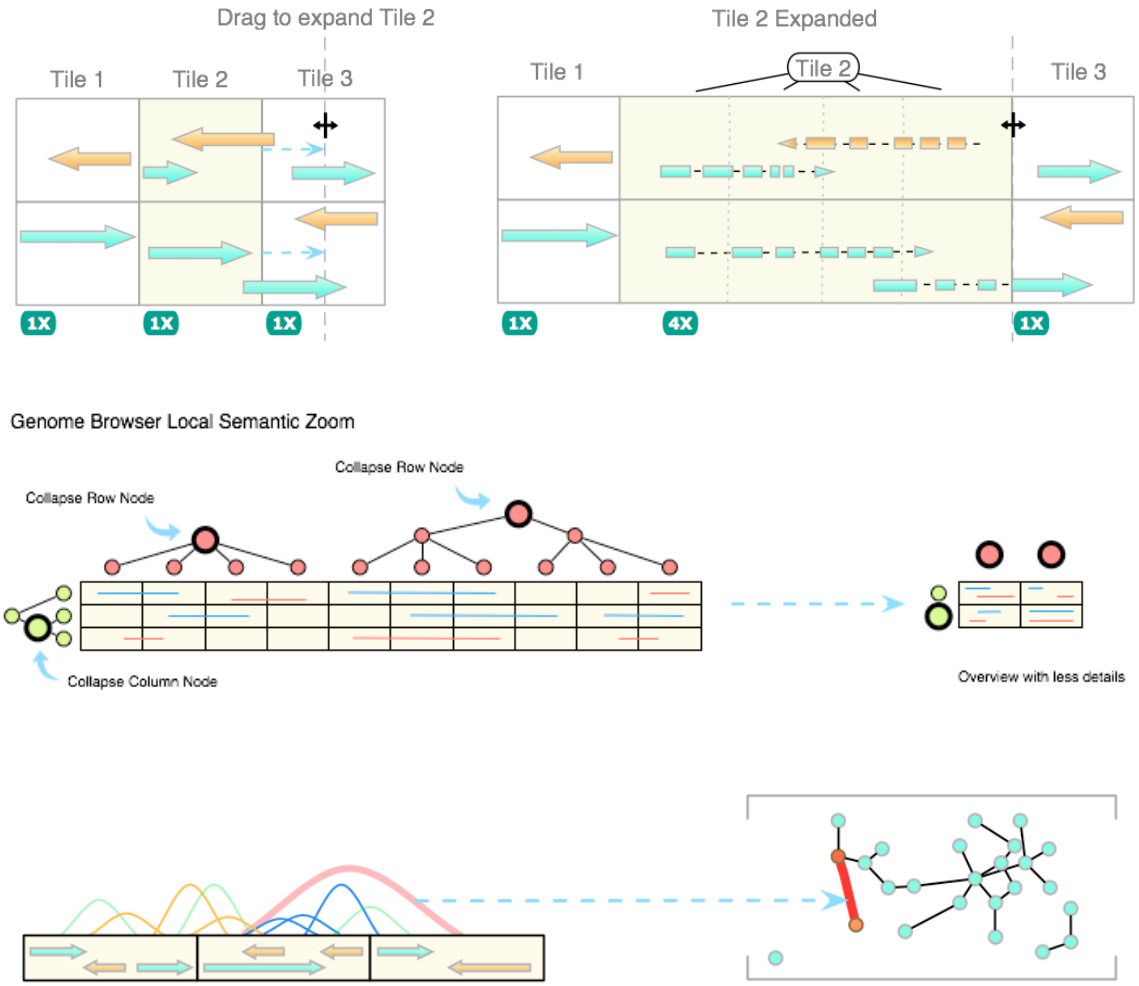
browser allows each track to be displayed at several predefined visual aggregations, such as 'dense', 'pack' or 'full'. At the inceptions of genome browsers, the majority of the genome is associated with positional annotation data; therefore this linear model has proven to be very useful. However, with the accumulation of functional annotation, structural and experimental data, the linear representation of the genome is becoming difficult for integrated visual exploration. For example, it's difficult to visualize long distance genomic interactions such as long distance upstream enhancers, linkage disequilibrium and chromosomal interactions as the target regions may be very far away from each other on the linear scale. It is also not easy to visualize quantitative omics data such as RNA-Seq results from multiple series or Metabolomics measurements in time series. Furthermore, the current genome browser model only allows data 'tracks' to be different along the vertical axis. With the incoming personal omics era, it's conceivable that heterogeneous data with different provenance would be aggregated into the same view. It would then be possible to juxtapose genomics (SNPs), transcriptomics (gene expression), metabolomics (metabolomics fingerprints) and proteomics (biomarker proteins) data along the horizontal axis, and multiple samples along the vertical axis to obtain an aggregated overview of integrated omics datasets. We think the CoolMap model, with flexible concept aggregation along both axes, will serve as an excellent starting candidate for future development of omics data browsers. Figure 5-1 shows some conceptual benefits the CoolMap model could bring to the genome browser. The CoolMap model could also be extended for other usages, such adaptive display of geographical information, aggregation of spreadsheets and user interface design, to name a few.

In the end, what's the future of the development of omics to people's day-to-day lives? At the current development rate, we can only expect that omics will be even more affordable, portable, accurate, interpretable and complete. The embedded devices with the development of smart tablets may derive a variety of micro measurement devices, that we can wear or mount on head, waist, legs or arms 24/7, and generate a time stamped, cloud-synced fingerprint of people's physiological data. Such data could be incorporated as part of the electronic health record (EHR) and benefit the diagnostics and prognostics enormously – personal, customized medicine will no longer be a dream. The future Omics Visual Explorers will be indispensable for life science researchers, medical practitioners, or even

'civilian scientists' to explore and make best use of such datasets. Such Omics Visual Explorers can also potentially serve as great support systems for decision-making – accurate and fast diagnostics, along with cost-effective cures will be more accessible to the general public. I hope my thesis work, as a whole, will humbly contribute to this on going movement.

Figure 5-1 Conceptual extension of next generation genome browser from CoolMap.

Top: semantic zoom along the x-axis will enable visualization of a certain region in detail while keep the surrounding regions in high level overview. Middle: the browser can have ontological structures on both axes to make it possible to view in a highly aggregated form. Bottom: long distance interactions can be addressed in the aggregated form and interlinked to network or other forms of views.



APPENDIX

CoolMap Implementation Details

The CoolMap is composed of two major components: the map visualization framework along with a variety of user interface widgets, and underlying data structure that contains the numeric matrices and the row/column multiple inheritance ontology tree. The CoolMap is implemented in Java Development Kit version 6 (JDK6), and some of the statistical analysis functions are ported from the R statistical analysis framework (http://www.r-project.org/). The aim is to replicate the Cytoscape model, with a core visualization canvas and numerous interfaces that can be harnessed by third party developers.

The core concept of CoolMap is to enable aggregation both along the rows and columns of a data matrix. Initially the design was focused on only working with numerical values – if a row or column node is represented as a group node that contain a number of child nodes, the corresponding cell will be a summarization value (such as mean, minimum or maximum). Later on I changed the design of CoolMap enable the data matrix contain arbitrary data types, such as strings, hyperlinks, or images. As long as a matrix of data objects can be aggregated into a single object, a custom renderer can be developed to visualize such data objects. Table 4-1 Feature comparison of CoolMap with some other Tools

	R(gplots-heatmap.2)	JTreeView	MatrixZoom	CoolMap
Interactive	No	Yes	Yes	Yes
User-rearrange row/column order	No	No	No	Yes
Rendering other data-types	No	Yes	No	Yes
Rendering other than color	No	Yes	No	Yes
Annotation overlays	Yes	No	No	Yes
Clustering	Yes	Yes	Yes	Under-

				development
Node Expansion/Collapse	No	No	Yes	Yes
Multiple Ontology	No	No	No	Yes
Multiple Plot Link	No	Yes	Yes	Under-development
Extract Sub-portion	Programmatically	Yes	Yes	Yes
Search/Filter	Programmatically	No	No	Yes

Figure 4-1 illustrates the basic design and workflow of CoolMap. This way the user can have a much-condensed view of a very big dataset – for example, instead of inspecting each individual genes and individual samples in an experiment, the researcher may look at the summary of gene pathways and sample groups. If anything particular of interest is discovered, then the researcher may expand the rows and columns for more in-depth view of the details.

Basic Data Structure

There are four core concepts in the CoolMap data structure:

- The base matrix: an interface of 2D matrix that holds the source matrix data. Source data can take any Java Object type.
- The aggregator: defines methods that aggregate a region in the base matrix (some rows and columns combinations, not necessarily consecutive) into a single value. The resultant view matrix can hold data objects of a different type from the source type.
- The view matrix: holds data that will be immediately rendered into view via a renderer. The renderer is capable of translating the view object type into arbitrary type of graphics.
- The ontology: holds mappings of labels (nodes) to groups of rows or columns in the source matrix data, as well as the hierarchical relationship between these labels. The ontology can have arbitrary number of parents as well as children.

Next let's explain each of the component in detail:

Base Matrix: In the traditional heat-map like visualization, once the matrix is loaded into view, all rows and columns are one to one mapped to the original dataset, even though they can be re-ordered or sorted by a variety of clustering algorithms. The CoolMap's base matrix is an abstract 2D matrix that can hold any type of data, such as Double (numeric), String (character), Image or even composite objects or user defined classes. It is defined as an interface, which means that the underlying storage of the base matrix does not necessarily need to be a matrix form, as long as it complies to the interface definition and returns

function values such as getNumberOfRows(), getNumberOfColumns(), getValueAt(int row, int column), getRowLabel(int row), getColLabel(int col) etc. An example is to define a sparse matrix, and use a hash table to store node to node pairwise interaction data for a network, or define a remote 'base' matrix, with values being pulled remotely from a server. The possibility of coupling the base matrix interface to different data types made it very flexible for CoolMap to load a variety of data by only writing new custom base matrix definitions, instead of imposing major changes to the software structure. The CoolMap is preloaded with base matrix that can handle numeric data (Double, Integer, etc.). Other matrices can be easily added later via the plugin interfaces.

Aggregator: an aggregator is also an interface that defines functions of summarizing groups of values in the base matrix into a single value. For example, if the groups of rows for genes belong to a certain pathway, and the groups of columns belong to a certain cancer type, the summary value can be the mean expression value of all the genes in that pathway, from that specific cancer type. There are four possible summarization patterns:

1. Single cell: an aggregation at [row, column] is returned.
2. Single row: an aggregation at [row,columns[]] is returned.
3. Single column: an aggregation at [rows[],column] is returned.
4. Sub matrix: an aggregation at [rows[], columns[]] is returned.

The region in the base matrix need not to be continuous, as the array rows[] and columns[] can contain arbitrary combinations of row and column indices. The summarized single values are then put together to produce a view matrix. The aggregator can produce a view matrix containing a different data type from the base matrix. For example, the base matrix can contain numeric values, and the aggregator can transform the base values into Boolean (true or false) by a comparing to a threshold. As illustrated in Table 4-1 Feature comparison of CoolMap with some other Tools

	R(gplots-heatmap.2)	JTreeView	MatrixZoom	CoolMap
Interactive	No	Yes	Yes	Yes
User-rearrange row/column order	No	No	No	Yes
Rendering other data-types	No	Yes	No	Yes

Rendering other than color	No	Yes	No	Yes
Annotation overlays	Yes	No	No	Yes
Clustering	Yes	Yes	Yes	Under-development
Node Expansion/Collapse	No	No	Yes	Yes
Multiple Ontology	No	No	No	Yes
Multiple Plot Link	No	Yes	Yes	Under-development
Extract Sub-portion	Programmatically	Yes	Yes	Yes
Search/Filter	Programmatically	No	No	Yes

Figure 4-1, it is also possible to simply execute data transformation (such as log2) using the aggregator. This adds a layer of flexibility of visualizing the base data.

View Matrix: A view matrix is a 2D matrix that simply holds data generated from an aggregator. The rows and columns of a view matrix can be customized by the user, using both rows and columns from the base matrix and ontology nodes, which are mapped to groups of base rows or columns. If the aggregator is simply a pass-through (pass the value without modification) and the rows and columns of the view matrix are identical to the base matrix, the rendered view matrix will reflect exactly the base matrix. By choosing aggregators and row/column nodes combinations, the view matrix can reflect a subset, or an aggregated view of the base matrix. The flexible arrangement of the view matrix enables the researchers to browse data from a very condensed way, or only focus on regions of focus interest.

Ontology tree: An Ontology tree holds a multi-tree of nodes that can be mapped to the rows or columns of the source data. For example, an Ontology node can be a Gene Ontology term/KEGG pathway that contains a number of genes from the base matrix, or an intermediate node from the merge result of a hierarchical agglomerative clustering. Each node in an Ontology tree can have arbitrary number of parent nodes, and arbitrary number child nodes. When the ontology nodes are added into the rows or columns of the view matrix, the corresponding cell value is then replaced with the summary value generated by the designated aggregator. The most striking difference between the CoolMap's Ontology and other heatmap implementations (such as Eisen's tree[102]) is that the CoolMap's row or column annotations are stored in a multi-tree[200]. In a conventional single tree, such as a hierarchical tree or Gene Ontology, each tree-node contains only a single parent, but can have multiple children. This single tree does not work when duplicate rows or columns occur in the view – for example, the result from a fuzzy clustering or Non-Negative Matrix Factorization (NMF) will have ambiguous gene membership assignments, which could require some genes be placed more than once in the heatmap; or if the user would like to inspect the gene expression heatmap results of two pathways, the pathway may contain shared house-keeping genes. In these scenarios, the recursive traversal of trees may fail because the multi parent structure may lead to loops, which is not allowed. The expansion/collapsing mechanisms of row or column nodes also require careful concern to tackle user-interaction conflicts.

Auxiliary Functions

In addition to the basic data structures, some peripheral classes are also developed for a variety of CoolMap operation and visualization capabilities. The most important ones are CoolMap Controller, Renderer and Annotation Renderer.

The **Controller** handles all the operations that alter the view matrix. As described before, the layout of the view matrix is determined by its row and column nodes, whenever rows or columns in the view matrix are changed (expanded, collapsed, moved or deleted), the view matrix will need to be updated accordingly. It would not be very economical to update the entire matrix every time using the aggregators as many of the changes happen locally. Therefore, the expansion, collapsing or reordering or the view matrix will only trigger local changes. When a view matrix is created, the active Ontology nodes are stored in the lists of ActiveRowNodes and ActiveColumnNodes. In addition, the expanded row or column tree nodes are stored in the ActiveRowTreeNodes and ActiveColTreeNodes. When a node is expanded, only nodes in the lowest level (that determine the layout of the view matrix) are stored in the active nodes lists (Figure). All other nodes are 'pushed' up into a tree branch.

The following operations can be performed to alter the rows and columns view matrix:

1. **Insert nodes**: insert new row or column nodes into the view matrix.
2. **Remove nodes**: remove rows or columns from the view matrix.
3. **Expand to child nodes**: remove the current node, and insert its immediate child nodes in the Ontology tree at the original position.
4. **Expand to base**: remove the current node, and replace the node with its base matrix row or column labels.
5. **Expand node to all child nodes:** remove the current node, and insert all its leaf child nodes in the Ontology tree at the original position.
6. **Collapse a node**: remove all child nodes of the selected node from the active node list, and place this node back to the active node list.
7. **Shift nodes:** Shift a region of rows or columns from its original location to a new location. Used for the drag and drop operation so the user may rearrange nodes.
8. **Rearrange nodes:** Rearrange the all the rows or columns, or a subset of the rows and columns with new ordering (such as an optimized row/column order that

minimizes the distances between adjacent rows or columns, computed from a Matrix seriation algorithm[201]).

Whenever these operations are performed, only the regions need to be updated are computed using the aggregator; the rest of the unchanged regions are copied over to the new view matrix. For example, when the user expands a column node, the other columns are copied to the corresponding location in the new view matrix. The sub-region corresponding to the newly inserted child nodes is computed, inserted and only the subregion is redrawn. Also when the view matrix to be updated is large enough (more than 200 rows / columns), it is divided into 4 sub regions and updated using parallel threads. By using these optimizations, even updating a view matrix with thousands of rows and columns take very little time (less than 1 sec). In our test case of a matrix of (5000 rows by 5000 columns, with 1000 row groups and 1000 column groups), the multi-threaded version can speed up the update process by 3~4 fold.

One of the major challenges is how to make it intuitive to handle node operations when there are duplicate nodes, since the Ontology tree is a multi-inheritance tree and the rows and columns of the view matrix can be arranged very flexibly. The user may get lost if multiple copies of the nodes with the same labels are presented without clear indication of what the underlying ontologies they are associated with. The solution is to preserve the reference to the 'ontology' each node is descended from. In this way, multiple occurrences of the same node in the matrix view can be differentiated, and therefore prevent collisions in collapse operations. For example, if the user displays a matrix view containing all genes from MAPK (KEGG:ko04010), Chemokine signaling pathway (KEGG:ko04062) and Apoptosis (KEGG:ko04210), there will be three copies of NFKB1(KEGG:k02580) in the view as NFKB1 functions in all these three pathways. Each NFKB1 would then be have a reference to the parent pathway respectively, and it can be displayed in the row/column node browser (discussed later). The ontology tree is still structured as a multi-tree in which each node can have multiple parent and child nodes, with one condition that no loops are allowed (any node can not trace back to itself). The ontology tree branches in view, resulted from node expansion or other tree generation algorithms such as hierarchical clustering, are always presented as a single tree. This enables the user to explore any partial branch in an ontology tree with ease, and even make comparisons across multiple ontology trees. Once the layout

of the view matrix is determined, the view matrix is then passed to the renderer to translate from data to graphics.

Renderer: The render is also written as an interface. A render takes a value in the view matrix, and translate it into a graphics object. Similar to the other basic components, the renderer can be extended to render any type of data with any type 2D visualization. For example, a numeric value can be rendered as a colored rectangle (traditional heatmap), bars, circles or even texts and images. The only thing needed to do is to override the corresponding abstract render methods. Similar to updating the matrix, the renderer also use multiple threads to update large graphics objects if necessary, to ensure performance.

One distinct feature of the renderer interface is the adaptive rendering. CoolMap allows each cell of the view matrix take different render sizes (to be covered later), so that the user may investigate regions with more detail while keeping the context in view. The renderer defines three methods of rendering: SD, Normal and HD. The SD mode is used for generate thumbnails or the view matrix cells are very small, when speed is preferred over quality. The normal mode is used when the cells are moderately large, and the HD mode is used when the cells are sufficiently large so that more detailed graphics and annotations may be possible to be overlaid. In each of these abstract functions, a Graphics2D object, as long with the coordinates in the view matrix, row/column label and the dimension of the cell are passed for writing the custom renderer. In addition, the renderer can also return an optional ToolTip for the underlying value to be displayed in the mouse hover.

CoolMap comes with two basic renderers, the Double to Color renderer and the Double to Bar renderer. The Double to Color renderer is a straightforward implementation of the conventional heatmap. The user may determine a color scale (currently a two-color system, beginning to end), and the min/max values these two colors are mapped too. All immediate values in the view matrix are then interpolated using the color scale and rendered as rectangles. The user may also assign colors for under- and over-flows, for values that are higher or lower than the min/max values. The color to bar maps double values to bar heights in each cell, with the min value of 0 height and max value mapped to the full cell height. Other Renders can be written easily by implementing the renderer interface.

Annotation Renderer: As a cell in the view matrix can actually reflect a summary of a group of values if the row and/or the column node is an ontology node, sometimes it's necessary to have a quick look at the underlying data before expanding the row or column nodes. For example, if the row node is a gene pathway, and the column node is a group of samples, the base matrix is a table of gene expression profiles, and the summary is the maximum. When the researcher sees a high expression value for that group of samples in that pathway, the next question would be, which gene-sample actually has the high expression? And what about the expression profiles in the rest of the group? An annotation is the overlay of a visualization of the details of the underlying data. Similar to the renderer, it is also defined as an interface and has SD, normal and HD renders functions for adaptive rendering. In addition, the base matrix elements are also accessible within the render functions, so that details of the members can be plotted instead of a single summary value. For example, a box plot of the underlying values within two ontology nodes can be plotted as an annotation instead of the single value summary (such as the mean value), to reveal additional details within the data. It is very easy to extend the interface to write custom annotation renderers.

Standardized operations: To implement undo/redo automated operations, all operations are defined as a subclass of an operation class. For every redo-able operation, a reverse operation is defined accordingly (such as expand or collapse a node). Once an operation is performed, it is stored in the operations stack and can be reversed accordingly.

The Visualization Canvas

With all the underlying models, it is critical to provide an intuitive and flexible interface to utilize the exploratory capability of CoolMap. Because of the distinct features (custom graphics rendering, multi-tree ontology nodes as rows and columns), nothing off the shelf immediately satisfies our needs, although we have tried a variety of libraries such as G the graphics library, Java FX, Processing, etc. We eventually developed a custom visualization panel to render the view matrix and offer contextual ontology node operations. The display panel have high performance, provides the user with some smooth-scrolling experience similar to popular browsing software such as the Google Map, with rich interactivity and information prompts. After several trials, I adopted a multi-layer design that is both easily programmable, and extensible.

The root container is a JLayeredPane. The JLayeredPane allows arbitrary number of Swing Components to be stacked on top of each other, and thus separate different functional components into various layers. Furthermore, each layer can be blended together using layer alpha (An alpha value determines the transparency of the pixel), and mask each other's mouse event functions. The user may pan the view, select nodes of interest to show detailed or alternative annotations, permute rows or columns, or expand/collpase certain row/column nodes. We have implemented the following layers:

- **Base layer**: a static background, which can be a uniform color or gradient.
- **CoolMap layer**: this layer stores the rendered view matrix (the CoolMap view) using a designated renderer, with zoomX and zoomY determine the default width and height or each cell respectively. If the visible region of the view matrix is larger than the viewport (the dimension of the display panel), only a portion of the view matrix plus some buffering region is rendered. When the user scrolls the heatmap, or jumps to a different region on the heatmap, the boundary conditions are checked. If the current rendered view matrix is insufficient to fill the viewport, a new sub portion of the heatmap will be rendered as replacement. This mechanism offers the user a smooth scrolling experience with minimal number of re-rendering.
- **Annotation layer**: This layer is a container for the aforementioned annotation renderer, using brushing (ref) to add context-specific annotations to the base visualization. Each annotation is a small visualization of the cell value, which can be generated with or without other associated data. Each annotation should return two objects: 1) an Image which can be placed on top of the corresponding cell of the matrix view image; 2) an Image contains a tooltip. As rendering annotations can be more computationally intensive than rendering the view matrix, only the 'selected' cells are rendered with annotation overlay. So far there are two kinds of annotations:
    1. **Sub-render of the base matrix**: if the current row or cell is an annotation node, the underlying base matrix can be rendered to replace the single summary value. The user then could alternate between the summary value and the base matrix for the selected nodes.
    2. **Bar representation of the cell value**: sometimes the color may not be the best way to contrast values, especially when the differences of the values are

small or due to color insensitivity of the human eye. A bar representation can be placed on the selected cells, with the bar height representing the cell value.

- **Mask Layer**: The mask layer provides an interface to block certain cells in the view matrix with a dark shade. This layer is utilized by the filter widget (will be described later) to show only cells that pass certain criteria.
- **Highlight Layer**: This layer provides a single method to highlight a certain region in the view matrix, to indicate a change just took place (rows/columns were added, removed or reordered)
- **Selection Layer**: The user may perform standard selection operations on this layer to trigger a rectangular selection region, with single mouse click selection, and shift + click for area selection. As mentioned before, selection will trigger rendering of corresponding annotations, if any annotation renderer is set to be active. The selections can also be set by clicking on the row labels and column labels (described later), or programmatically.
- **Hover layer**: Whenever the mouse cursor is hovered within a heatmap, a hover grid is drawn to indicate the current active row – column coordinate. A tool tip is shown to display the cell value (the toString() method from the underlying object). If the cell is currently selected, a secondary tip will be shown to display the tooltip from the corresponding annotation.
- **CoolMap mouse listener layer**: Instead of assigning mouse event to individual layers, this layer captures all mouse events, such as mouse move, drag, single click, double click, etc., and update the coordinate parameters. All layers will utilize the parameters to make necessary updates.
- **Grid Layer**: Another distinct feature of CoolMap is it allows each cell to have its own dimension. This is done via the Grid Layer to remove the mouse event conflict between cell resizing and map pan. When the grid layer is activated (Hot Key Alt-C), solid lines mark the boundaries of the underlying cells are drawn on top of the other CoolMap layers below. The user will then be able to adjust the size of each cell by dragging the cell bounds, just like adjusting the dimension of cells in an excel spreadsheet. The adjusted cell dimensions will persist when the global zoom is changed.

- **Row/Column Drawer Container**: These two layers are containers for additional layers (Row/Column Drawer) that contain annotation information that is only related to rows or columns, such as the row/column labels and row/column tree. As this layer is placed on top of the CoolMap mouse listener layer, the mouse events captured on these two layers will override the heatmap mouse events. We have also implemented an interface to make it possible to add third party row/column drawers. Currently we have:
    1. **Row/Column label Drawer**: The label drawer displays the row and column node names. A row or column can be selected by single mouse left click, or shift + left click to select a region of rows or columns. The selected region can also be dragged around to rearrange row/column orders.
    2. **Row/Column Ontology Tree:** The Ontology Tree Panel displays the active nodes and their parent nodes. The user may expand ontology nodes to child nodes, to base nodes or collapse tree nodes.

Each layer can be turned on or off independently; further layers could be added in the future to extend more functions.

Widgets

The CoolMap application is developed using dockable framework with Multiple Document Interface (MDI), which means more than one CoolMap instance can be displayed and explored at the same time. The dockable framework (Sanaware Javadocking) allows the user to arrange functional widgets freely (similar to arranging the floating tool panels in Adobe Photoshop). In addition, only widgets of interest can be shown to create a 'workspace' for a certain analysis. For example, if the user only want to browse a heatmap typed data without using any of the ontology nodes, the Ontology related widgets could be hidden from the view. This makes the entire user interface highly customizable and flexible. So far the following widgets have been developed. Other widgets can be easily added by extending the widget interface.

- **Canvas widget**: contains all the active CoolMap instances. Each of the CoolMap instance can be moved, resized, maximized or minimized. The selected CoolMap instance will be assigned as the 'Active' CoolMap object.

- **CoolMap list widget**: contains all the loaded CoolMap instances. The user may switch the active CoolMap, read additional description, or delete CoolMap(s).
- **Aggregator widget**: contains a list of available aggregators. Currently, only numeric aggregators are provided, such as Max, Min, Median, Mean, Standard Deviation, Variance and Sum. Aggregator will always check with the base matrix to determine whether they are compatible. Only compatible aggregators can be assigned.
- **Renderer Widget**: contains a list of available renderers. Each renderer also has a User Interface panel for configuration of parameters. When a renderer is assigned to a CoolMap object, a new instance is created so that each CoolMap instance can have its own copy of renderer and set of parameters.
- **Annotation Renderer widget**: contains a list of available annotation renderers. Similar to the renderer widget, each it ensures render compatibility, and each CoolMap instance maintains its own copy of the Annotation Renderer and set of parameters.
- **View Matrix Value widget**: this widget displays a table for the values in the selected region. The values are produced using the 'toString()' method of the objects in the View Matrix. This widget allows the user to look at the rendered view matrix and its underlying values simultaneously.
- **Radar widget**: this widget offers an overview of the entire rendered map (thumbnail) the user may rapidly jump to regions of interest on the map. The map will also be changed whenever the layout, or the renderer of the view matrix was changed.
- **Active Node Searcher widget**: this widget lists all the active row/column nodes for the current active CoolMap instance. The user may search for a row or column using a keyword. The search also supports Java regular expression for fuzzy search. Single click on a search result will immediately bring the CoolMap view centered at the selected row/column node.
- **Ontology Browser widget:** this widget allows the user to search ontology and add intermediate nodes of interest to the view. The use may select a loaded Ontology, and the table shows the number of parent and child nodes, as well as their labels. The user may add a single or multiple entries of ontology nodes at the end of the current view matrix, or insert at the current selected region.

- **Ontology Display widget:** this custom developed widget shows the organizational relationship between the parent nodes and child nodes. As the CoolMap Ontology is a multi-tree instead of a single inheritance tree, the standard JTree cannot be used to display the proximity of an Ontology term. The parent, child and siblings or a node can be selected respectively, and added to the active CoolMap View.
- **Filter widget:** the filter widget provides a mechanism to use the mask layer to 'block' certain nodes out of the view based on certain criteria. The filter is also defined as an interface so that a variety of filtering mechanisms can be incorporated. Currently CoolMap provides three kinds of filters: filter by value (range), by row node label or column node label. Multiple filters can be applied at the same time, the user may also determine whether to use 'AND' (all active filtering criteria must be satisfied) or 'OR' at least one filtering criteria must be satisfied.

The widget interface can be extended by plugin development similar to the approach adopted by Cytoscape. We believe that this opening framework will incorporate community effort to continuously improve the CoolMap framework.

Integration with Cytoscape

As have mentioned in many literature, linking different views can help the user better understand distinct aspects of the underlying data. There have been some previous efforts to integrate a heatmap display in Cytoscape, such as ClusterMaker and VistaClara, but these implementations lack rich interactivity, and row/column annotation data integration. The CoolMap can be bundled as a Cytoscape plugin to provide such two-way link-view functions. We have developed a prototype, heatnet, to illustrate the feasibility and effectiveness of using such network-Coolmap linked-view

BIBLIOGRAPHY

1. McGinn, S. & Gut, I. G. DNA Sequencing-Spanning the Generations. *New biotechnology* (2012).doi:10.1016/j.nbt.2012.11.012

2. Joyce, A. R. & Palsson, B. Ø. The model organism as a system: integrating "omics" data sets. *Nature reviews. Molecular cell biology* **7**, 198–210 (2006).

3. Lockhart, D. & Winzeler, E. Genomics, gene expression and DNA arrays. *NATURE-LONDON-* (2000).at <http://128.197.54.216/reading/L22-microarrays.pdf>

4. Ball, M. P. *et al.* A public resource facilitating clinical use of genomes. *Proceedings of the National Academy of Sciences of the United States of America* **109**, 11920–7 (2012).

5. Ginsburg, G. S. & Willard, H. F. Genomic and personalized medicine: foundations and applications. *Translational research : the journal of laboratory and clinical medicine* **154**, 277–87 (2009).

6. Cordero, P. & Ashley, E. A. Whole-genome sequencing in personalized therapeutics. *Clinical pharmacology and therapeutics* **91**, 1001–9 (2012).

7. Lunshof, J. E. *et al.* Personal genomes in progress: from the human genome project to the personal genome project. *Dialogues in clinical neuroscience* **12**, 47–60 (2010).

8. Kent, W. J. *et al.* The human genome browser at UCSC. *Genome research* **12**, 996–1006 (2002).

9. Chan, P. P., Holmes, A. D., Smith, A. M., Tran, D. & Lowe, T. M. The UCSC Archaeal Genome Browser: 2012 update. *Nucleic acids research* **40**, D646–52 (2012).

10. Pruitt, K., Tatusova, T. & Maglott, D. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic acids research* (2005).at <http://nar.oxfordjournals.org/content/33/suppl_1/D501.short>

11. Hubbard, T., Barker, D. & Birney, E. The Ensembl genome database project. *Nucleic acids …* (2002).at <http://nar.oxfordjournals.org/content/30/1/38.short>

12. Blow, N. Transcriptomics: The digital generation. *Nature* (2009).at <http://www.nature.com/nature/journal/v458/n7235/full/458239a.html>

13. McGettigan, P. A. Transcriptomics in the RNA-seq era. *Current Opinion in Chemical Biology* (2013).doi:10.1016/j.cbpa.2012.12.008

14. Wang, Z., Gerstein, M. & Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics* **10**, 57–63 (2009).

15. Busby, M. A., Stewart, C., Miller, C., Grzeda, K. & Marth, G. Scotty: A Web Tool For Designing RNA-Seq Experiments to Measure Differential Gene Expression. *Bioinformatics* (2013).doi:10.1093/bioinformatics/btt015

16. Lepoivre, C. *et al.* TranscriptomeBrowser 3.0: introducing a new compendium of molecular interactions and a new visualization tool for the study of gene regulatory networks. *BMC bioinformatics* **13**, 19 (2012).

17. Li, R., Yu, C., Li, Y., Lam, T. & Yiu, S. SOAP2: an improved ultrafast tool for short read alignment. … (2009).at <http://bioinformatics.oxfordjournals.org/content/25/15/1966.short>

18. Trapnell, C., Pachter, L. & Salzberg, S. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* (2009).at <http://bioinformatics.oxfordjournals.org/content/25/9/1105.short>

19. Wang, L., Feng, Z., Wang, X. & Zhang, X. DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics* (2010).at <http://bioinformatics.oxfordjournals.org/content/26/1/136.short>

20. Barrett, T., Troup, D. & Wilhite, S. NCBI GEO: mining tens of millions of expression profiles—database and tools update. *Nucleic acids* … (2007).at <http://nar.oxfordjournals.org/content/35/suppl_1/D760.short>

21. Rodriguez-Tome, P. & Stoehr, P. The european bioinformatics institute (EBI) databases. *Nucleic acids* … (1996).at <http://nar.oxfordjournals.org/content/24/1/6.short>

22. Labarga, A. & Valentin, F. Web services at the European bioinformatics institute. *Nucleic acids* … (2007).at <http://nar.oxfordjournals.org/content/35/suppl_2/W6.short>

23. Golub, T., Slonim, D. & Tamayo, P. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science* (1999).at <http://www.sciencemag.org/content/286/5439/531.short>

24. Ozsolak, F. & Milos, P. M. RNA sequencing: advances, challenges and opportunities. *Nature reviews. Genetics* **12**, 87–98 (2011).

25. Kussmann, M., Raymond, F. & Affolter, M. OMICS-driven biomarker discovery in nutrition and health. *Journal of biotechnology* **124**, 758–87 (2006).

26. De Hoog, C. L. & Mann, M. Proteomics. *Annual review of genomics and human genetics* **5**, 267–93 (2004).

27. Tan, H. T., Lee, Y. H. & Chung, M. C. M. Cancer proteomics. *Mass spectrometry reviews* **31**, 583–605

28. Julka, S. & Regnier, F. Quantification in proteomics through stable isotope coding: a review. *Journal of proteome research* (2004).at <http://pubs.acs.org/doi/abs/10.1021/pr0340734>

29. Pandey, A. & Mann, M. Proteomics to study genes and genomes. *Nature* (2000).at <http://www2.hawaii.edu/~wsu/nature_405_837_2000.pdf>

30. Buck, M. J. & Lieb, J. D. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics* **83**, 349–360 (2004).

31. Martens, L. *et al.* PRIDE: the proteomics identifications database. *Proteomics* **5**, 3537–45 (2005).

32. Rose, P. W. *et al.* The RCSB Protein Data Bank: new resources for research and education. *Nucleic acids research* **41**, D475–D482 (2013).

33. Consortium, U. The universal protein resource (UniProt). *Nucleic Acids Res* (2008).at <http://scholar.google.com/scholar?hl=en&q=UniProt&btnG=&as_sdt=1,23&as_sdtp=#0>

34. Joshi-Tope, G. & Gillespie, M. Reactome: a knowledgebase of biological pathways. *Nucleic acids* … (2005).at <http://nar.oxfordjournals.org/content/33/suppl_1/D428.short>

35. Tarcea, V. G. *et al.* Michigan molecular interactions r2: from interacting proteins to pathways. *Nucleic acids research* **37**, D642–6 (2009).

36. Mikami, T., Aoki, M. & Kimura, T. The application of mass spectrometry to proteomics and metabolomics in biomarker discovery and drug development. *Current molecular pharmacology* **5**, 301–16 (2012).

37. Petricoin, E. F., Zoon, K. C., Kohn, E. C., Barrett, J. C. & Liotta, L. A. Clinical proteomics: translating benchside promise into bedside reality. *Nature reviews. Drug discovery* **1**, 683–95 (2002).

38. Karr, T. L. Application of proteomics to ecology and population biology. *Heredity* **100**, 200–6 (2008).

39. Barker, B. M. *et al.* Transcriptomic and proteomic analyses of the Aspergillus fumigatus hypoxia response using an oxygen-controlled fermenter. *BMC genomics* **13**, 62 (2012).

40. Nie, L., Wu, G., Culley, D. E., Scholten, J. C. M. & Zhang, W. Integrative analysis of transcriptomic and proteomic data: challenges, solutions and applications. *Critical reviews in biotechnology* **27**, 63–75

41. Fukushima, A., Kusano, M., Redestig, H., Arita, M. & Saito, K. Integrated omics approaches in plant systems biology. *Current opinion in chemical biology* **13**, 532–8 (2009).

42. Patti, G., Yanes, O. & Siuzdak, G. Innovation: Metabolomics: the apogee of the omics trilogy. *Nature Reviews Molecular Cell Biology* (2012).at <http://www.nature.com/nrm/journal/v13/n4/abs/nrm3314.html>

43. Wishart, D. S. *et al.* HMDB 3.0--The Human Metabolome Database in 2013. *Nucleic acids research* **41**, D801–7 (2012).

44. Wishart, D. S. *et al.* HMDB: a knowledgebase for the human metabolome. *Nucleic Acids Research* **37**, D603–D610 (2009).

45. Smith, C., O'Maille, G. & Want, E. METLIN: a metabolite mass spectral database. *Therapeutic drug* … (2005).at <http://journals.lww.com/drug-monitoring/Abstract/2005/12000/METLIN__A_Metabolite_Mass_Spectral_Database.16.aspx>

46. Ma, H. *et al.* The Edinburgh human metabolic network reconstruction and its functional analysis. *Molecular systems biology* **3**, 135 (2007).

47. Kanehisa, M. The KEGG database. *Novartis Foundation symposium* **247**, 91–101; discussion 101–3, 119–28, 244–52 (2002).

48. Gao, J., Tarcea, V. & Karnovsky, A. Metscape: a Cytoscape plug-in for visualizing and interpreting metabolomic data in the context of human metabolic networks. … (2010).at <http://bioinformatics.oxfordjournals.org/content/26/7/971.short>

49. Karnovsky, A., Weymouth, T. & Hull, T. Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data. … (2012).at <http://bioinformatics.oxfordjournals.org/content/28/3/373.short>

50. Blekherman, G. *et al.* Bioinformatics tools for cancer metabolomics. *Metabolomics* **7**, 329–343 (2011).

51. Zhang, A., Sun, H., Wang, P., Han, Y. & Wang, X. Modern analytical techniques in metabolomics analysis. *The Analyst* **137**, 293–300 (2012).

52. Patti, G. J., Yanes, O. & Siuzdak, G. Innovation: Metabolomics: the apogee of the omics trilogy. *Nature reviews. Molecular cell biology* **13**, 263–9 (2012).

53. Ma, Y., Zhang, P., Yang, Y., Wang, F. & Qin, H. Metabolomics in the fields of oncology: a review of recent research. *Molecular biology reports* **39**, 7505–11 (2012).

54. Nadella, K. Metabolomics in Agriculture. *Omics: a journal of* … (2012).at <http://online.liebertpub.com/doi/abs/10.1089/omi.2011.0067>

55. Nordström, A. & Lewensohn, R. Metabolomics: moving to the clinic. *Journal of neuroimmune pharmacology : the official journal of the Society on NeuroImmune Pharmacology* **5**, 4–17 (2010).

56. Rochfort, S. Metabolomics reviewed: a new "omics" platform technology for systems biology and implications for natural products research. *Journal of natural products* **68**, 1813–20 (2005).

57. Cevallos-Cevallos, J. M. & Reyes-De-Corcuera, J. I. Metabolomics in food science. *Advances in food and nutrition research* **67**, 1–24 (2012).

58. Blekherman, G., Laubenbacher, R. & Cortes, D. Bioinformatics tools for cancer metabolomics. *Metabolomics* (2011).at <http://www.springerlink.com/index/5J403L2T30328383.pdf>

59. Ellis, D. I. & Goodacre, R. Metabolic fingerprinting in disease diagnosis: biomedical applications of infrared and Raman spectroscopy. *The Analyst* **131**, 875–85 (2006).

60. Sreekumar, A. *et al.* Metabolomic profiles delineate potential role for sarcosine in prostate cancer progression. *Nature* **457**, 910–4 (2009).

61. German, J., Watkins, S. & Fay, L. Metabolomics in practice: emerging knowledge to guide future dietetic advice toward individualized health. *Journal of the American dietetic* … (2005).at <http://www.sciencedirect.com/science/article/pii/S000282230501031X>

62. Huang, D. W., Sherman, B. T. & Lempicki, R. A. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research* **37**, 1–13 (2009).

63. Jennen, D. *et al.* Integrating transcriptomics and metabonomics to unravel modes-of-action of 2,3,7,8-tetrachlorodibenzo-p-dioxin (TCDD) in HepG2 cells. *BMC systems biology* **5**, 139 (2011).

64. Zhu, J. *et al.* Stitching together multiple data dimensions reveals interacting metabolomic and transcriptomic networks that modulate cell regulation. *PLoS biology* **10**, e1001301 (2012).

65. Zhang, W., Li, F. & Nie, L. Integrating multiple "omics" analysis for microbial biology: application and methodologies. *Microbiology (Reading, England)* **156**, 287–301 (2010).

66. Kint, G., Fierro, C., Marchal, K., Vanderleyden, J. & De Keersmaecker, S. C. J. Integration of "omics" data: does it lead to new insights into host-microbe interactions? *Future microbiology* **5**, 313–28 (2010).

67. Romero, R. *et al.* The use of high-dimensional biology (genomics, transcriptomics, proteomics, and metabolomics) to understand the preterm parturition syndrome. *BJOG : an international journal of obstetrics and gynaecology* **113 Suppl** , 118–35 (2006).

68. Lesko, L. J. & Schmidt, S. Individualization of drug therapy: history, present state, and opportunities for the future. *Clinical pharmacology and therapeutics* **92**, 458–66 (2012).

69. Hammond, W. The EHR Chronicle in USA and Europe from the 60's to the future. *epj-observatoriet.dk* at <http://epj-observatoriet.dk/konference2004/PowerPoints/WilliamEHammond.pdf>

70. Blobel, B. & Pharow, P. Analysis and evaluation of EHR approaches. *Methods of information in medicine* (2009).at <http://www.schattauer.de/en/magazine/subject-areas/journals-a-z/methods/contents/archive/issue/662/manuscript/11042/download.html>

71. Gurwitz, D. High-Quality Phenomics are Crucial for Informative Omics Studies. *Drug Development Research* (2012).at <http://onlinelibrary.wiley.com/doi/10.1002/ddr.21025/full>

72. Chen, R. *et al.* Personal omics profiling reveals dynamic molecular and medical phenotypes. *Cell* **148**, 1293–307 (2012).

73. Gentleman, R. & Carey, V. Bioconductor: open software development for computational biology and bioinformatics. *Genome …* (2004).at <http://genomebiology.com/2004/5/10/R80>

74. Lê Cao, K.-A., González, I. & Déjean, S. integrOmics: an R package to unravel relationships between two omics datasets. *Bioinformatics (Oxford, England)* **25**, 2855–6 (2009).

75. Neuweger, H. *et al.* MeltDB: a software platform for the analysis and integration of metabolomics experiment data. *Bioinformatics (Oxford, England)* **24**, 2726–32 (2008).

76. Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.-Y. & Kitano, H. Software for systems biology: from tools to integrated platforms. *Nature reviews. Genetics* **12**, 821–32 (2011).

77. Strimmer, K. A unified approach to false discovery rate estimation. *BMC bioinformatics* **9**, 303 (2008).

78. Cui, X. & Churchill, G. Statistical tests for differential expression in cDNA microarray experiments. *Genome Biol* (2003).at <http://www.biomedcentral.com/content/pdf/gb-2003-4-4-210.pdf>

79. Tusher, V., Tibshirani, R. & Chu, G. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the …* (2001).at <http://www.pnas.org/content/98/9/5116.short>

80. Opgen-Rhein, R. & Strimmer, K. Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach. *Statistical Applications in Genetics and …* (2007).at <http://www.bepress.com/context/sagmb/article/1252/type/pdf/viewcontent>

81. Zuber, V. & Strimmer, K. Gene ranking and biomarker discovery under correlation. *Bioinformatics (Oxford, England)* **25**, 2700–7 (2009).

82. Yeung, K. & Ruzzo, W. Principal component analysis for clustering gene expression data. *Bioinformatics* (2001).at <http://bioinformatics.oxfordjournals.org/content/17/9/763.short>

83. Lee, S. & Batzoglou, S. Application of independent component analysis to microarrays. *Genome biology* (2003).at <http://www.biomedcentral.com/content/pdf/gb-2003-4-11-r76.pdf>

84. Carmona-Saez, P. Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC …* (2006).at <http://www.biomedcentral.com/1471-2105/7/78/>

85. Fogel, P., Young, S., Hawkins, D. & Ledirac, N. Inferential, robust non-negative matrix factorization analysis of microarray data. *Bioinformatics* (2007).at <http://bioinformatics.oxfordjournals.org/content/23/1/44.short>

86. Rakotomamonjy, A. Variable selection using svm based criteria. *The Journal of Machine Learning Research* (2003).at <http://dl.acm.org/citation.cfm?id=944977>

87. Diaz-Uriarte, R. GeneSrF and varSelRF: a web-based tool and R package for gene selection and classification using random forest. *BMC bioinformatics* **8**, 328 (2007).

88. Belacel, N., Wang, Q. & Cuperlovic-Culf, M. Clustering methods for microarray gene expression data. *Omics : a journal of integrative biology* **10**, 507–31 (2006).

89. Newman, M. & Girvan, M. Finding and evaluating community structure in networks. *Physical Review E* **69**, 026113– (2004).

90. Fraley, C. & Raftery, A. E. Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association* **97**, 611–631 (2002).

91. Qin, Z. S., Bilenky, M., Su, G. & Jones, S. J. M. MotifOrganizer: a scalable model-based motif clustering tool for mammalian genomes . *Frontiers in bioscience (Elite edition)* **E5**, 785–797 (2013).

92. Morris, J. H. *et al.* clusterMaker: a multi-algorithm clustering plugin for Cytoscape. *BMC bioinformatics* **12**, 436 (2011).

93. Dennis, G. *et al.* DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome biology* **4**, P3 (2003).

94. Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* **102**, 15545–50 (2005).

95. Margolin, A. A. *et al.* ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics* **7 Suppl 1**, S7 (2006).

96. Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics* **9**, 559 (2008).

97. Tian, Y., Mceachin, R., Santos, C. & Patel, J. SAGA: a subgraph matching tool for biological graphs. *Bioinformatics* (2007).at <http://bioinformatics.oxfordjournals.org/content/23/2/232.short>

98. Brasch, S., Linsen, L. & Fuellen, G. VANLO--interactive visual exploration of aligned biological networks. *BMC bioinformatics* **10**, 327 (2009).

99. Lo, K. *et al.* Integrating external biological knowledge in the construction of regulatory networks from time-series expression data. *BMC systems biology* **6**, 101 (2012).

100. Gehlenborg, N. *et al.* Visualization of omics data for systems biology. *Nature Methods* **7**, S56–S68 (2010).

101. Hibbs, M. A., Dirksen, N. C., Li, K. & Troyanskaya, O. G. Visualization methods for statistical analysis of microarray clusters. *BMC bioinformatics* **6**, 115 (2005).

102. Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America* **95**, 14863–14868 (1998).

103. Baran, R., Robert, M., Suematsu, M., Soga, T. & Tomita, M. Visualization of three-way comparisons of omics data. *BMC Bioinformatics* **8**, 72 (2007).

104. Xia, T. & Dickerson, J. OmicsViz: Cytoscape plug-in for visualizing omics data across species. *Bioinformatics* (2008).at <http://bioinformatics.oxfordjournals.org/content/24/21/2557.short>

105. Bushati, N., Smith, J., Briscoe, J. & Watkins, C. An intuitive graphical visualization technique for the interrogation of transcriptome data. *Nucleic acids research* **39**, 7380–9 (2011).

106. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research* **13**, 2498–504 (2003).

107. Junker, B. H., Klukas, C. & Schreiber, F. VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC bioinformatics* **7**, 109 (2006).

108. Arakawa, K., Kono, N., Yamada, Y., Mori, H. & Tomita, M. KEGG-based pathway visualization tool for complex omics data. *In Silico Biology* **5**, 419–423 (2005).

109. Rohn, H., Klukas, C. & Schreiber, F. Creating views on integrated multidomain data. *Bioinformatics (Oxford, England)* **27**, 1839–45 (2011).

110. Baker, M. Gene data to hit milestone. *Nature* **487**, 282–3 (2012).

111. Millard, B. L., Niepel, M., Menden, M. P., Muhlich, J. L. & Sorger, P. K. Adaptive informatics for multifactorial and high-content biological data. *Nature methods* **8**, 487–93 (2011).

112. Wiesinger, M. *et al.* Data and knowledge management in cross-Omics research projects. *Methods in molecular biology (Clifton, N.J.)* **719**, 97–111 (2011).

113. Lowe, H. & Barnett, G. Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. *JAMA: the journal of the American Medical …* (1994).at <http://jama.ama-assn.org/content/271/14/1103.short>

114. Filzmoser, P., Maronna, R. & Werner, M. Outlier identification in high dimensions. *Computational Statistics & Data Analysis* **52**, 1694–1711 (2008).

115. Rog, C. J., Chekuri, S. C. & Edgerton, M. E. Challenges of the information age: the impact of false discovery on pathway identification. *BMC research notes* **5**, 647 (2012).

116. Hardin, J. & Rocke, D. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis* (2004).at <http://www.sciencedirect.com/science/article/pii/S0167947302002803>

117. Su, G., Kuchinsky, A., Morris, J. H., States, D. J. & Meng, F. GLay: community structure analysis of biological networks. *Bioinformatics* **26**, 3135–3137 (2010).

118. Praneenararat, T., Takagi, T. & Iwasaki, W. Integration of interactive, multi-scale network navigation approach with Cytoscape for functional genomics in the big data era. *BMC genomics* **13 Suppl 7**, S24 (2012).

119. Abello, J. & Ham, F. van Matrix zoom: A visual interface to semi-external graphs. *Information Visualization, 2004. INFOVIS …* (2004).at <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1382907>

120. Rhodes, D. & Yu, J. ONCOMINE: a cancer microarray database and integrated data-mining platform. … *(New York, NY)* (2004).at <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1635162/>

121. Alibés, A., Yankilevich, P., Cañada, A. & Díaz-Uriarte, R. IDconverter and IDClight: conversion and annotation of gene and protein IDs. *BMC bioinformatics* **8**, 9 (2007).

122. Huang, D. W. *et al.* DAVID gene ID conversion tool. *Bioinformation* **2**, 428–30 (2008).

123. Dai, M., Wang, P., Jakupovic, E., Watson, S. J. & Meng, F. Web-based GeneChip analysis system for large-scale collaborative projects. *Bioinformatics (Oxford, England)* **23**, 2185–7 (2007).

124. Yuerong, Z., Yuelin, Z. & Wei, X. EzArray: A web-based highly automated Affymetrix expression array data management and analysis system. *BMC Bioinformatics* (2008).at <http://en.scientificcommons.org/28178128>

125. Ferry-Dumazet, H., Gil, L. & Deborde, C. MeRy-B: a web knowledgebase for the storage, visualization, analysis and annotation of plant NMR metabolomic profiles. *BMC plant …* (2011).at <http://www.biomedcentral.com/1471-2229/11/104/>

126. Chatr-Aryamontri, A. *et al.* The BioGRID interaction database: 2013 update. *Nucleic acids research* **41**, D816–23 (2012).

127. Bader, G. D. *et al.* BIND--The Biomolecular Interaction Network Database. *Nucleic acids research* **29**, 242–5 (2001).

128. Newman, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* **103**, 8577–82 (2006).

129. Fortunato, S. & Barthélemy, M. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America* **104**, 36–41 (2007).

130. Ruan, J. & Zhang, W. Identifying network communities with a high resolution. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics* **77**, 016104 (2008).

131. Csardi, G. & Nepusz, T. The igraph software package for complex network research. *interjournal.org* at <http://interjournal.org/manuscript_abstract.php?361100992>

132. Bader, G. D. & Hogue, C. W. V An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics* **4**, 2 (2003).

133. Rivera, C. G., Vakil, R. & Bader, J. S. NeMo: Network Module identification in Cytoscape. *BMC bioinformatics* **11 Suppl 1**, S61 (2010).

134. Merico, D., Gfeller, D. & Bader, G. D. How to visually interpret biological data using networks. *Nature biotechnology* **27**, 921–4 (2009).

135. Wakita, K. & Tsurumi, T. Finding Community Structure in Mega-scale Social Networks. 9 (2007).at <http://arxiv.org/abs/cs/0702048>

136. Clauset, A., Newman, M. & Moore, C. Finding community structure in very large networks. *Physical Review E* **70**, 066111– (2004).

137. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical review. E, Statistical, nonlinear, and soft matter physics* **76**, 036106 (2007).

138. Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E* **74**, 036104– (2006).

139. Reichardt, J. & Bornholdt, S. Statistical mechanics of community detection. *Physical Review E* **74**, 016110– (2006).

140. Pons, P. & Latapy, M. Computing Communities in Large Networks Using Random Walks. *Computer and Information Sciences - ISCIS 2005* **3733**, 284–293 (2005).

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

142. Kamada, T. & Kawai, S. An algorithm for drawing general undirected graphs. *Information processing letters* (1989).at <http://www.cse.ohio-state.edu/~chaudhua/Temporary/wi11_888/Papers/Kamada_GraphDrawing_Letters89.pdf>

143. Adai, A. T., Date, S. V, Wieland, S. & Marcotte, E. M. LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. *Journal of molecular biology* **340**, 179–90 (2004).

144. Brandes, U. & Pich, C. Eigensolver Methods for Progressive Multidimensional Scaling of Large Data. *Graph Drawing* **4372**, 42–53 (2007).

145. Reingold, E. M. & Tilford, J. S. Tidier Drawings of Trees. *IEEE Transactions on Software Engineering* **SE-7**, 223–228 (1981).

146. Lancichinetti, A., Fortunato, S. & Radicchi, F. Benchmark graphs for testing community detection algorithms. *Physical Review E* (2008).at <http://pre.aps.org/abstract/PRE/v78/i4/e046110>

147. Steinley, D. Properties of the Hubert-Arable Adjusted Rand Index. *Psychological Methods; Psychological Methods* (2004).at <http://psycnet.apa.org/journals/met/9/3/386/>

148. Kim, H. & Park, H. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics* (2007).at <http://bioinformatics.oxfordjournals.org/content/23/12/1495.short>

149. Ashkenazi, M., Bader, G. D., Kuchinsky, A., Moshelion, M. & States, D. J. Cytoscape ESP: simple search of complex biological networks. *Bioinformatics* **24**, 1465–1466 (2008).

150. Ferrara, C. T. *et al.* Genetic networks of liver metabolism revealed by integration of metabolic and transcriptional profiling. *PLoS genetics* **4**, e1000034 (2008).

151. Xu, E. Y. *et al.* Integrated pathway analysis of rat urine metabolic profiles and kidney transcriptomic profiles to elucidate the systems toxicology of model nephrotoxicants. *Chemical research in toxicology* **21**, 1548–61 (2008).

152. Nam, H., Chung, B. C., Kim, Y., Lee, K. & Lee, D. Combining tissue transcriptomics and urine metabolomics for breast cancer biomarker identification. *Bioinformatics (Oxford, England)* **25**, 3151–7 (2009).

153. Fei, Z. *et al.* Tomato Functional Genomics Database: a comprehensive resource and analysis package for tomato functional genomics. *Nucleic acids research* **39**, D1156–63 (2011).

154. Schellenberger, J., Park, J. O., Conrad, T. M. & Palsson, B. Ø. BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. *BMC bioinformatics* **11**, 213 (2010).

155. Hardin, J., Mitani, A., Hicks, L. & VanKoten, B. A robust measure of correlation between two genes on a microarray. *BMC bioinformatics* (2007).at <http://www.biomedcentral.com/1471-2105/8/220>

156. Bickel, D. Robust cluster analysis of microarray gene expression data with the number of clusters determined biologically. *Bioinformatics* (2003).at <http://bioinformatics.oxfordjournals.org/content/19/7/818.short>

157. Dai, M. *et al.* Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data. *Nucleic acids research* **33**, e175 (2005).

158. Lê, S., Josse, J. & Husson, F. FactoMineR: An R package for multivariate analysis. *Journal of statistical software* (2008).at <http://sebastien.ledien.free.fr/unofficial_factominer/docs/article_FactoMineR.pdf >

159. Forbes, S. A. *et al.* COSMIC (the Catalogue of Somatic Mutations in Cancer): a resource to investigate acquired mutations in human cancer. *Nucleic acids research* **38**, D652–7 (2010).

160. Beroukhim, R. *et al.* The landscape of somatic copy-number alteration across human cancers. *Nature* **463**, 899–905 (2010).

161. Wang, H. *et al.* Comparative analysis and integrative classification of NCI60 cell lines and primary tumors using gene expression profiling data. *BMC genomics* **7**, 166 (2006).

162. Shieh, A. D. & Hung, Y. S. Detecting outlier samples in microarray data. *Statistical applications in genetics and molecular biology* **8**, Article 13 (2009).

163. Pearson, R., Gonye, G. & Schwaber, J. Outliers in microarray data analysis. … *of Microarray Data Analysis III* (2004).at <http://www.springerlink.com/index/L21604201R2475L7.pdf>

164. Quackenbush, J. Computational analysis of microarray data. *Nature Reviews Genetics* (2001).at <http://theory.bio.uu.nl/BPA/pdf/Obligatory_reading/Quackenbush.pdf>

165. Ruan, J., Dean, A. K. & Zhang, W. A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC systems biology* **4**, 8 (2010).

166. Rousseeuw, P. J. & Driessen, K. Van A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**, 212–223 (1999).

167. Wissler, C. The Spearman Correlation Formula. *Science (New York, NY)* (1905).at <http://www.ncbi.nlm.nih.gov/pubmed/17836577>

168. Brophy, A. An algorithm and program for calculation of Kendall's rank correlation coefficient. *Behavior Research Methods* (1986).at <http://www.springerlink.com/index/Y701535N04285666.pdf>

169. Maronna, R. Robust M-estimators of multivariate location and scatter. *The annals of statistics* (1976).at <http://www.jstor.org/stable/10.2307/2957994>

170. Maronna, R. & Zamar, R. Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics* (2002).at <http://www.tandfonline.com/doi/abs/10.1198/004017002188618509>

171. Chilson, J., Ng, R., Wagner, A. & Zamar, R. Parallel computation of high-dimensional robust correlation and covariance matrices. *Algorithmica* (2006).at <http://www.springerlink.com/index/D822W24777900163.pdf>

172. Kullback, S. & Leibler, R. On information and sufficiency. *The Annals of Mathematical Statistics* (1951).at <http://www.jstor.org/stable/10.2307/2236703>

173. Paninski, L. Estimation of entropy and mutual information. *Neural Computation* (2003).at <http://www.mitpressjournals.org/doi/abs/10.1162/089976603321780272>

174. Rorabacher, D. Statistical Treatment for Rejection of Deviant Values: Critical Values of Dixon Q Parameter and Related Subrange Ratios at the 95 percent Confidence Level. *Anal. Chem* **63**, 139 – 146 (1991).

175. Siirtola, H. & Mäkinen, E. Constructing and reconstructing the reorderable matrix. *Information Visualization* **4**, 32–48 (2005).

176. Wilkinson, L. The History of the Cluster Heat Map. *American Statistician* **63**, 179–184 (2009).

177. Reich, M., Liefeld, T., Gould, J. & Lerner, J. GenePattern 2.0. *Nature* … (2006).at <http://www.nature.com/ng/journal/v38/n5/full/ng0506-500.html>

178. Kincaid, R., Kuchinsky, A. & Creech, M. VistaClara: an expression browser plug-in for Cytoscape. *Bioinformatics* **24**, 2112–2114 (2008).

179. Miller, C., Hooper, S. & Lecheler, L. InfoViz for Education: Using Interactive Information Visualization to Enhance Sense-and Decision-Making for Teachers and Students. *World Conference on E-Learning in* … (2009).at <http://www.editlib.org/p/32722/>

180. Sartor, M. & Mahavisno, V. ConceptGen: a gene set enrichment and gene set relation mapping tool. … (2010).at <http://bioinformatics.oxfordjournals.org/content/26/4/456.short>

181. Walter, B., Bala, K., Kulkarni, M. & Pingali, K. Fast agglomerative clustering for rendering. *2008 IEEE Symposium on Interactive Ray Tracing* 81–86 (2008).doi:10.1109/RT.2008.4634626

182. Bar-Joseph, Z., Gifford, D. K. & Jaakkola, T. S. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics* **17 Suppl 1**, S22–S29 (2001).

183. Thalamuthu, A., Mukhopadhyay, I., Zheng, X. & Tseng, G. C. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics* **22**, 2405–2412 (2006).

184. Liiv, I. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining* **3**, 70–91 (2010).

185. Caraux, G. & Pinloche, S. PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order. *Bioinformatics* **21**, 1280–1281 (2005).

186. Wu, H.-M., Tien, Y.-J. & Chen, C.-H. GAP: A graphical environment for matrix visualization and cluster analysis. *Computational Statistics & Data Analysis* **54**, 767–778 (2010).

187. North, C. & Shneiderman, B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. *AVI 00 Proceedings of the working conference on Advanced visual interfaces* 128–135 (2000).doi:10.1145/345513.345282

188. Saldanha, A. J. Java Treeview -- extensible visualization of microarray data. *Bioinformatics* **20**, 3246–3248 (2004).

189. Van Dijk, S. J. *et al.* A saturated fatty acid-rich diet induces an obesity-linked proinflammatory gene expression profile in adipose tissue of subjects at risk of metabolic syndrome. *The American journal of clinical nutrition* **90**, 1656–64 (2009).

190. Sartor, M. A. *et al.* Genome-wide methylation and expression differences in HPV(+) and HPV(-) squamous cell carcinoma cell lines are consistent with divergent mechanisms of carcinogenesis. *Epigenetics : official journal of the DNA Methylation Society* **6**, 777–87 (2011).

191. Su, G., Mao, B. & Wang, J. MACO: a gapped-alignment scoring tool for comparing transcription factor binding sites. *In Silico Biology* **6**, 307–310 (2006).

192. Schultz, S. C., Shields, G. C. & Steitz, T. A. Crystal structure of a CAP-DNA complex: the DNA is bent by 90 degrees. *Science (New York, N.Y.)* **253**, 1001–7 (1991).

193. Cline, M. S. *et al.* Integration of biological networks and gene expression data using Cytoscape. *Nature protocols* **2**, 2366–82 (2007).

194. Assenov, Y., Ramírez, F., Schelhorn, S.-E., Lengauer, T. & Albrecht, M. Computing topological parameters of biological networks. *Bioinformatics (Oxford, England)* **24**, 282–4 (2008).

195. Vailaya, A. *et al.* An architecture for biological information extraction and representation. *Bioinformatics (Oxford, England)* **21**, 430–8 (2005).

196. Zheng, Z.-L. & Zhao, Y. Transcriptome comparison and gene coexpression network analysis provide a systems view of citrus response to "Candidatus Liberibacter asiaticus" infection. *BMC Genomics* **14**, 27 (2013).

197. Stuart, J., Segal, E., Koller, D. & Kim, S. A gene-coexpression network for global discovery of conserved genetic modules. *Science* (2003).at <http://www.sciencemag.org/content/302/5643/249.short>

198. Reverter, A., Hudson, N. & Wang, Y. A gene coexpression network for bovine skeletal muscle inferred from microarray data. *Physiological …* (2007).at <http://physiolgenomics.physiology.org/content/28/1/76.short>

199. Choi, J., Yu, U., Yoo, O. & Kim, S. Differential coexpression analysis using microarray data and its application to human cancer. *Bioinformatics* (2005).at <http://bioinformatics.oxfordjournals.org/content/21/24/4348.short>

200. Caseau, Y. Efficient handling of multiple inheritance hierarchies. *ACM SIGPLAN Notices* **28**, 271–287 (1993).

201. Hahsler, M., Hornik, K. & Buchta, C. Getting Things in Order: An introduction to the R package seriation. (2007).at <http://epub.wu.ac.at/852/>