Supplement: Year 5 (6/01/2010-5/31/2011):

# DBI 0445666: Conceptual data integration for the VirtualPlant

**Project Participants**

## 1. What people have worked on your project?

Principal Investigator:

* Gloria M. Coruzzi (NYU Biology)

Co-Principal Investigators:

* Rodrigo A. Gutiérrez (NYU Biology)
* Dennis E. Shasha (NYU Courant, Math and Computer Sciences)

Senior Personnel:

* Daniel Tranchina (NYU Biology, NYU Courant, Math and Computer Sciences)

Graduate Students:

* Muller Casey (NYU Courant, Math and Computer Sciences)

Project Director:

* Manpreet S. Katari (NYU Biology)

Post-doc

* Arthur Goldberg (NYU Biology)

Programmers:

* Jonathan Kelfer (M.S. Computer Science)
* Lee Parnell (M.S. Computer Science)
* Rebecca Davidson (B.S. Computer Science)
* Roberto Jimeno

## 2. What other organizations have been involved as partners?

**Digital Image Design, Inc. (DiDi)**:

Type: Collaborative research (organization's staff work with project staff on the project)

We continue to consult with Mr. W. Bradford Paley from Digital Image Design, for the development of novel data visualization tools. Mr. Paley is an interface designer with extensive experience in applications that interact and manipulate large volumes of data.

## 3. Project Activities

Our long-term goal is to enable researchers to integrate, analyze and visualize genomic data to understand how internal and external perturbations affect processes and networks controlling plant growth and development. In this NSF project, we start with data integration of the known relationships among genes, proteins and molecules (extracted from public databases and/or generated with predictive algorithms) as well as experimental measurements under many different treatments. We go beyond data integration to conceptual integration by using novel visualization techniques to render the multivariate information in visual formats that facilitate extraction of biological concepts. We also use mathematical and statistical methods to help summarize the data. We implement and combine these approaches in a system we term "VirtualPlant". Whereas our project relates specifically to Arabidopsis, the data structures, algorithms, and visualization tools are designed in a species-independent way. Thus the informatic, mathematical, statistical and visualization tools that we develop can be used to model the cellular and physiological responses of any organism for which genomic data is available.

The VirtualPlantsoftware is unique among software platforms that analyze genomic data (e.g. MapMan, Cytoscape, Genevestigator), because VirtualPlantis designed to enable iterative cycles of data analysis, integration and hypothesis generation which is a hallmark of Systems Biology. This is possible due to the unique “Cart”feature of VirtualPlantwhich enables users to run data analysis and visualization tools on collections of genes in the “Cart” save the output back into the Cart*.* This E-commerce like *modus operandi* allows for multiple iterative cycles of data analysis, integration and visualization that fuel new cycles of experimentation and analysis, finally producing more focused and experimentally testable hypotheses.

*VirtualPlant Community Outreach*: We currently have more than 700 registered users from more than 30 countries on VirtualPlant from academia and industry and thousands of non-registered users from around the world who are actively and effectively using our VirtualPlant software system since its public release in year 2 of the grant, June 2006. Biologists and computer scientist are using VirtualPlant for the purpose it was designed for, to support the analysis of original genomic data generated by the researchers themselves. During all stages of software development, we have reached out to plant biologists within and beyond the PI’s lab for beta testing. These laboratories include: Philip Benfey (Duke) a developmental biologist who beta tested Sungear, Nigel Crawford (UCSD) a metabolic biologist who beta tested (and has written papers using) BioMaps and Sungear, Dominique Bergman (Stanford) a developmental biologist who beta tested the gene networks function in VirtualPlant*,* Amy Litt (New York Botanical Garden) an evolutionary biologist who helped beta test orthology tools we have developed (OrthologID) (Chiu et al., 2006), and Loreto Houigue (Pontificia Universidad Catolica de Chile) a researcher focusing on abiotic stress responses who beta tested set operations and gene expression correlation tools. We have also demonstrated the features of VirtualPlantannually at various national and international conferences including bioinformatics workshops at the Arabidopsis meeting. On July 28th, 2008 we held our first VirtualPlant course at NYU, which trained 6 scientists in a hands on workshop.

## 4. Training and development

The following post-docs, PhD and undergraduate students have contributed to the VirtualPlant project and were trained under this project:

Post-doctoral scientist:

* Manpreet S. Katari (NYU Biology)
* Arthur Goldberg (NYU Biology)
* Gabriel Krouk (NYU Biology)

Students:

* Steve Nowicki (UG student, NYU College of Arts & Sciences)
* Chris Poultney (PhD student, NYU Courant)
* Varuni Prabhakar (UG student, Barnard College)
* Jason Reisman (PhD student, NYU Courant)
* Saurabh Kumar (PhD student, NYU Courant)
* Ana F. Arroja (MS student, NYU Courant)
* Ranjita Iyer (MS student, NYU Courant)
* Donghan Wang (MS student, NYU Courant)
* Jesse Lingeman (MS student, NYU Courant)
* Michael Schizas (MS student, NYU Biology)

Programmers:

* Jonathan Kelfer (B.S. Computer Science)
* Lee Parnell (M.S. Computer Science)
* Rebecca Davidson (AB Computer Science)
* Roberto Jimeno

## 5. Outreach activities

Presentations on VirtualPlant were given by the PI and co-PIs in the following educational and industrial settings:

Coruzzi (PI)

Mendel Biotech, Seminar: Nitrogen networks & the VirtualPlant, March 3, 2010.

Syngenta Fellows Symposium on Yield, Dec 18, 2007. Invited speaker.

Monsanto, Plant Systems Biology, May 2007. Invited speaker.

Shasha (co-PI)

“Biocomputational Puzzles” University of Montpellier, France. February 1, 2007.

“Biocomputational Puzzles” Ecole Polytechnique de Lausanne, Switzerland. November 11, 2006.

“Fast Calculations of Simple Primitives in Time Series” Universite Marne la Vallee, France. November 7, 2006.

Katari, M.S., Aceituno, F.F., Nowicki, S.D., Shasha, D.E., Coruzzi, G.M., and Gutierrez, R.A. VirtualPlant: A software platform to support Systems Biology research in the post-genomic era. *Garnish* Dec. 2006. Edition 5, pp. 25-26. A short article summarizing the database and the tools available on VirtualPlant was published in a newsletter:

## 6. Journal publications

*In preparation* - Gutiérrez RA, Prabhakar V, Colombo T, Chiaromonte F., Coruzzi GM (2011). “BioSkew/TextArc: A novel statistical and visualization tool for text data mining” (In preparation).

X. Zhang, D. Shasha, Y. Song and J. T. L. Wang (In Press) “Fast Elastic Peak Detection for Mass Spectrometry Data Mining,” ***IEEE Transactions on Knowledge and Data Engineering***.

Krouk G, Mirowski P, Lecun Y, Shasha DE, Coruzzi GM (2010 Dec 23) “Predictive network modeling of the high-resolution dynamic plant transcriptome in response to nitrate.” ***Genome Biol*** 11(12):R123

Cibrian-Jaramillo A, De la Torre-Barcena JE, Lee EK, Katari MS, Little DP, Stevenson DW, Martienssen R, Coruzzi GM, DeSalle R (2010) “Using phylogenomic patterns and gene ontology to identify proteins of importance in plant evolution.” ***Genome Biol Evol*** Jul 12;2:225-39.

Huang-Wen Chen, Sunayan Bandyopadhyay, Dennis E. Shasha, and Kenneth D. Birnbaum (2010) “Estimation of genome-wide redundancy in Arabidopsis thaliana,” ***BMC Evolutionary Biology*** 10:357; doi:10.1186/1471-2148-10-357

Krouk G, Crawford NM, Coruzzi GM, Tsay YF (2010) “[Nitrate signaling: adaptation to fluctuating environments](http://www.ncbi.nlm.nih.gov/pubmed/20093067?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum&ordinalpos=2)” ***Curr Opin Plant Biol***. 13(3):266-73

Katari MS, Nowicki SD, Aceituno FF, Nero D, Kelfer J, Thompson LP, Cabello JM, Davidson RS, Goldberg AP, Shasha DE, Coruzzi GM, Gutierrez RA (2010) “[VirtualPlant: A software platform to support Systems Biology research](http://www.ncbi.nlm.nih.gov/pubmed/20007449?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum&ordinalpos=1)” ***Plant Physiol***. 152: 500-515.

Di Natale R, Ferro A, Giugno R, Mongiovi M, Pulvirenti A and Shasha D (2010) "SING: Subgraph search In Non-homogeneous Graphs" ***BMC Bioinformatics***, 11:96doi:10.1186/1471-2105-11-96

Ruffel S, Krouk G, Coruzzi GM (2010) “[A Systems View of Responses to Nutritional Cues in Arabidopsis: Towards a Paradigm Shift for Predictive Network Modeling](http://www.ncbi.nlm.nih.gov/pubmed/19939945?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum&ordinalpos=2)” ***Plant Physiol***. 152:445-52.

Nero D, Kelfer J, Katari MS, Tranchina D, Coruzzi GM (2009) [In Silico Evaluation of Predicted Regulatory Interactions in Arabidopsis thaliana.](http://www.ncbi.nlm.nih.gov/pubmed/20025756?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum&ordinalpos=1) ***BMC Bioinformatics***. 10:435.

Lagana A, Forte S, Giudice A, Arena MR, Puglisi PL, Giugno R, Pulvirenti A, Shasha D, Ferro A (2009) "miRo: a miRNA knowledge base" ***Database: The Journal of Biological Databases and Curation***: 2009, doi 10.1093/database/bap008

Krouk G, Tranchina D, Lejay L, Cruikshank AA, Shasha DE, Coruzzi GM, and Gutiérrez RA. (2009). A systems approach uncovers restrictions for signal interactions regulating genome-wide responses to nutritional cues in Arabidopsis. ***PLoS Compt Biol****.* 5, 1-12.

Aceituno FF, Moseyko N, Rhee SY, Gutiérrez RA (2008). The rules of gene expression in plants: Organ identity and gene body methylation are key factors for regulation of gene expression in Arabidopsis thaliana. ***BMC Genomics***9, 438.

Diego Reforgiato, Rodrigo Gutierrez, Dennis Shasha “GraphClust: A Method for Clustering Databases of Graphs” ***Journal of Information & Knowledge Management (JIKM)*** Volume: 7, Issue: 4 (December 2008)

Gutierrez RA, Stokes TL, Thum K, Xu X, Obertello M, Katari MS, Tanurdzic M, Dean A, Nero DC, McClung CR, Coruzzi GM “Systems approach identifies an organic nitrogen-responsive gene network that is regulated by the master clock control gene CCA1.” **Proc Natl Acad Sci** U S A (2008 Mar 25)

Gifford ML, Dean A, Gutierrez RA, Coruzzi GM, Birnbaum KD “Cell-specific nitrogen responses mediate developmental plasticity.’ **Proc Natl Acad Sci U S A** (2008 Jan 15)

Thum KE, Shin MJ, Gutierrez RA, Mukherjee I, Katari MS, Nero D, Shasha D, Coruzzi GM (2008) “An integrated genetic, genomic and systems approach defines gene networks regulated by the interaction of light and carbon signaling pathways in Arabidopsis”. **BMC Syst Biol.** 2008; 2: 31.

Poultney C, Gutiérrez RA, Katari MS, Gifford ML, Paley WB, Coruzzi GM and Shasha DE (2007) “Sungear: Interactive visualization, exploration and functional analysis of genomic datasets”. B**ioinformatics**, 23: 259-61.

Gutiérrez RA, Lejay LV, Dean A, Chiaromonte F, Shasha DE, Coruzzi GM (2007). Qualitative network models and genome-wide expression data define carbon/nitrogen-responsive molecular machines in Arabidopsis. **Genome Biol** 8, R7.

**Computer Science conference publications:**

Bonnici V, Di Natale R, Ferro A, Giugno R, Mongiovi M, Pigola G, Pulvirenti A, Shasha D (2009) "Enhancing Graph Database Indexing By Suffix Tree Structure" ***BITS***

## 7. Other specific products

**Data:**

VirtualPlant Platform: The VirtualPlant platform synthesizes the known relationships among genes, proteins and molecules (extracted from public databases and/or generated with predictive algorithms) as well as experimental measurements under many different treatments. The Arabidopsis multinetwork in VirtualPlant currently has ~10,000 nodes (genes) and 230,000 interactions between them (for a description see Gutierrez et al., 2007). This tool has been used to identify testable regulatory networks that have been experimentally validated (see Gutierrez et al., 2008). We have also developed new visualization techniques and have integrated tools developed by other NSF-funded projects, to render the multivariate information in visual formats that facilitate extraction of biological concepts.

Processed Microarray Data: Microarray experiments were obtained using the NASC Affywatch Subscription. The CEL file from all the experiments were normalized using the gcrma, which is part of the BioConductor package available for the R programming language. The result is loaded in the VirtualPlant database and is available to the public for download from the VirtualPlant website.

Pair-wise correlation data for all probes in the Affymetrix ATH1 gene chip: The normalized values were correlated using the Spearman method and only the significant correlations were recorded in the database. The correlation values are used by several methods including gene network analysis where the correlation of a transcription factor and its target can be used to filter the predictive regulatory edges.

Gene network data: The Gene network data is comprised of two different components, the multi-network and regulatory predictions. The multi-network consists of gene interactions based on information from metabolic pathways and protein-protein interactions. The regulatory predictions were made using known DNA binding motifs of transcription families. A gene was a putative target of a transcription factor if the target gene contained the known motif in its upstream region. The regulatory network can be used in conjunction with the pair-wise correlation data mentioned above to reduce the number of false positives.

**Database**

VDB (VirtualPlant DataBase): The VirtualPlant database is a “light-weight” data warehouse, which contains some of the most commonly used data types. Some of the major data types include, metabolic pathways from KEGG, protein-protein interactions from BIND and Interolog databases, and GeneOntology and Gene annotations from TAIR. The database also contains processed data obtained by analyzing publicly available Microarray experiments. Finally the database also stores User profiles and the contents of their gene carts, which are essentially gene lists.

**Data Availability**

The VirtualPlant database is accessible via the VirtualPlant website. The website allows users to query the database and view the data in a user-friendly GUI. In many cases the website provides a link where the User can download and store the data on their computer locally.

**Software**

VirtualPlant: VirtualPlant can be accessed via the World Wide Web using the URL http://www.virtualplant.org/. Here Users can create an account to store their data sets and use many tools to analyze their genomic data such as microarray experiments. The website does not require a password and is available for free when used for non-for-profit purposes.

Sungear: Sungear is a rapid, visually interactive and biologist-driven exploration of standard questions on many experiments at a genomic scale. Sungear can represent an arbitrary number of experiments/lists. The names of the lists are located around the vertices of a polygon. The position of a circle and its arrows indicates which list(s) it is associated with. The size of a circle is proportional to the number of genes it contains.

GeneLights: GeneLights is a module within Sungear that allows Users to examine expression patterns of the genes that are present in Sungear. This gives the User the extra capability of selecting genes based on their expression patterns in addition to their Gene Ontology annotation.

CellStorm: CellStorm will allow us to visualize the Gene Network data in a subcellular context. A graphical representation of the subcellular components will be associated with each other based on their gene content and the type of interaction that exists between the genes.

BioMaps: BioMaps takes one or more set of genes and analyzes the functional terms (GO or MIPS functional terms) associated to the genes in each set. BioMaps determines which functional terms are statistically over-represented as compared to a background population (e.g. Arabidopsis genome). The output is presented in either a table format, which can be downloaded in a Microsoft Excel format, or a graphical representation based on the Hierarchy of the functional annotation.

Gene Network: Gene network analysis allows users to query our Gene Network data and displays the results in a graph using Cytoscape, an open source project that can be launched using Java Web Start. The tool also allows Users to include and exclude interactions before displaying the graph.

Supernode Network: The Supernode network helps summarize the results of a Gene network analyis. It is conceptually the combination of BioMaps and Gene Network analysis. The genes in the gene network are grouped (Supernode) based on their functional annotation and they are associated with other Supernodes with edges determined from the Gene Network Data. The size of the supernode is determined by the number of genes it contains.

CytoDiff: A plugin for Cytoscape that allows Users to color code their network graph based on experimental data that is loaded in their account. This interactive plugin allows users to select “Control” and “Treated” samples, which it then uses to calculate the color to plot on the network. The comparison can also be saved as “slide”. Once the user has created several slides, they can play it back as a movie. This feature allows the user to visualize time series data dynamically. The plugin also allows the Users to create new network based on gene lists in their GeneCart without having to relaunch the Cytoscape application.

VPLayout: A plugin for Cytoscape that redraws the network based on the coordinated provided by the user. This plugin is useful to visualize data derived from statistical methods, such as principal component analysis, in the form of a network.

**Software Availability**

VirtualPlant is accessible from the World Wide Web. Users will be able to access the data visualization and analysis tools freely utilizing a normal web browser. Specific parts of the software will be patented. All software is available upon request to non-for-profit organizations.

## 8. Contributions within discipline

The production version of tools developed under the Virtual Plant project has been made available to the community through our main web site (http://www.virtualplant.org). The BioMaps tool has been used in the analysis of plant genomic data by researchers in other labs including Crawford lab (UCSD), Wildermuth Lab (UC Berkeley), Doerner Lab (University of Edinburgh), Raikhel Lab (UC Riverside), Chory Lab (Salk Institute).

**Seminar presentations on VirtualPlant have been disseminated via the following:**

Coruzzi (PI):

**2010**

CSHL Plant Genome Course: Plant Systems Biology, July 2010

NSF US-EU Taskforce on Plant Biotechnology: Speaker, June 2, 2010

**2009**

CSHL Plant Genomes: Genes, Networks and Applications, March 4-7, 2009.

**2008**

19th International Conference on Arabidopsis Research, Montreal, July 23-27, 2008 (Session Chair and Plenary speaker: Systems Biology Plenary Session).

Society for Experimental Biology (UK) Symposium on Systems Biology the Society.

Marseille, France July 7-9, 2008 (Keynote speaker).

NSF iPlant Meeting, CSHL, May 2008 (Session moderator; Systems Biology).

6th Annual Keen Lecture, UC Riverside, Genome Center Jan 18, 2008.

**2007**

Keystone Symposium, Systems Biology and Regulatory Networks, Mar 22-27, 2007, Steamboat Springs (Plenary Speaker).

CSHL 5th meeting on Systems Biology: Global Regulation of Gene Expression March 28 - April 1, 2007. (Invited Speaker).

**2006**

**“Systems biology approaches to analysis of metabolic and regulatory networks of** Arabidopsis”at the 17th International Conference on Arabidopsis Research 2006. Madison, WI.

ISPMB Meeting; Plenary Speaker, Systems Biology, Adelaide, Australia Aug. 20-25, 2006

Society for Developmental Biology, Ann Arbor MI, June 17-19, 2006 (Plenary Speaker).

**2005**

Systems Biology Symposium, Plant Biotech Denmark, Nov 2005 (Plenary Speaker).

2nd Tri-National Arabidopsis Meeting: Neuchåtel, Switzerland, Aug 24-27, 2005.

Annual ASPB Meeting, New Approaches for Integrating Plant Genomes & Function. Seattle, WA, July 2005 (Plenary Speaker).

CSHL Arabidopsis Genome Course, July 2005 (Lecturer).

Frontiers in Plant Biology: Genomics & Beyond: Missouri Symposium, April 27-30, 2005.

Gutierrez (co-PI):

**2008**

Major Symposium at the American Society of Plant Biologists Conference, Mérida, México, 2008.

Symposium at the Sociedad Argentina de Fisiología Vegetal, Rosario, Argentina, 2008.

Workshop at the International Arabidopsis Conference on Arabidopsis Research, Montreal Canada, 2008.

Symposium Speaker at the Panamerican Association for Biochemistry and Molecular Biology Conference, Aguas de Lindoia Brazil, 2008.

Katari (Project Director):

**2010**

Systems Biology Workshop at Cold Spring Harbor Laboratories

Poster Presentation at New York University Tech Expo

**2008**

A Systems Approach to Nitrogen Regulatory Networks and the VirtualPlant. Network Biology, Hinxton, UK. August 2008.

VirtualPlant: A Software platform to support Systems Biology research in the post genomic era.New York Area Plant Molecular Biology, Adelphi University, New York, June 2008.

Nitrogen Regulatory networks and plant systems biology. Department of Energy GTL Systems Biology Network/Knowledgebase Workshop, Bethesda MD, May 2008.

iPLANT Collaborative Inaugural Symposium, Cold Spring Harbor Laboratory, NY, May 2008.

**2007**

18th International Conference on Arabidopsis Research 2007. Beijing, China. (Workshop presentations).

**2006**

17th International Conference on Arabidopsis Research 2006. Madison, WI. (Session and workshop presentations).

## 9. Contributions to other disciplines

Our goals are to adapt VirtualPlant for use in other model organisms including human. With this objective in mind, we have designed our software and data warehouse to be independent of the type of data and the data source. The database and all the utility scripts that go with it can work for any species. Moreover, one can work with multiple species at one time. The species of an object can be specified in the database using the NCBI Taxonomy ID of the species in the Taxon column of the OBJECT table. When loading the database, the loadVDB.pl script checks the taxid parameter in the configuration file and assigns the taxon to all of the objects in the file. The capability of storing multiple species allows VirtualPlant to be used for comparative studies. It will also allow the user to go from one species to another with the click of a button. We are currently integrating OrthologID (http://nypg.bio.nyu.edu/orthologid/) and ViCoGenTA (http://www.vicogenta.org), two tools for comparative genomic analysis. We are also adding to our VirtualPlant database network information derived from our work on Nitrogen Networks in Plants. Thus, the VirtualPlant project constitutes the strategic node for integrating and synergizing with our two other NSF grants:

**NSF Arabidopsis 2010 Genome Grant (MCB0929338)**: “Arabidopsis 2010: Nitrogen Networks in Plants” P.I. Gloria Coruzzi, Co-PIs Dennis Shasha (NYU Courant), Nigel Crawford (UCSD). Years 9-12 (7/2009-6/2013).

### NSF Plant Genome Grant (DBI0421604): ”Genomics of Comparative Seed Evolution” P.I. Gloria Coruzzi, Co-PIs Dennis Shasha (NYU Courant) Dennis Stevenson (NYBG), Richard McCombie (CSHL), Robert Martienssen (CSHL) Robert DeSalle (AMNH). Years 1-5 (10/2004-2/2010).

## 10. Contributions to human resources development

Training of students and post doctoral fellows in Bioinformatics and Visualization tools.

Post-doctoral scientist:

* Manpreet S. Katari (NYU Biology)
* Arthur Goldberg (NYU Biology)

Students:

* Steve Nowicki (UG student, NYU CAS)
* Chris Poultney (PhD student, NYU Courant)
* Varuni Prabhakar (UG student, Barnard College)
* Jason Reisman (PhD student, NYU Courant)
* Saurabh Kumar (PhD student, NYU Courant)
* Ana F. Arroja (MS student, NYU Courant)
* Ranjita Iyer (MS student, NYU Courant)
* Jonathan Kelfer (MS student, NYU Courant)
* Donghan Wang (MS student, NYU Courant)
* Jesse Lingeman (MS student, NYU Courant)

Programmers:

* Jonathan Kelfer (MS Computer Science)
* Lee Parnell (MS Computer Science)
* Rebecca Davidson (BS Computer Science)
* Roberto Jimeno

## 11. Contributions to resources for research and education

Distributed material from presentations on plant genomics and the VirtualPlant for High School and College Teachers at:

* High School Education: Recent Advances in Science, Oct. 22, 2005 “Plant Genomics & Networks”, Science Education curriculum (High School), NYU Steinhardt School of Education, Department of Teaching and Learning. This lecture was attended by teachers and students in NYC High Schools. Powerpoint presentation distributed to teachers.
* College Education. FRN Faculty Resource Network Summer 2005 Workshop on BIO 2010: Integrative Approaches to Teaching Life Sciences. Lecture: Plant Genomics and Systems Biology.

Graduate education: CSHL Arabidopsis Genome Course, Lecturer, July 2005: Genomics and the Virtual Plant.

July 2008 – VirtualPlant Course 101. We held our first one day course about VirtualPlant. There were a total of 5 students, 2 were from outside of New York University. The course was successful and we plan to organize another one this year.

## 12. Contributions beyond science and engineering

As part of this project, we have been working with W. Bradford Paley (Digital Image Design, Inc) who is a consultant on design of Graphical User Interfaces. Mr. Paley is an interaction designer, who brings visual interface ideas to clients on Wall Street, the design/art fields, and now to science in the context of the “VirtualPlant” Project. The VirtualPlant project has adapted and adopted TextArc, a text mining tool developed by Brad Paley. [Have we really used this?] We are in the process of implementing a statistical analysis tool that can be used with TextArc to give quantitative value to this tool. The statistical applications that we develop as part of VirtualPlant, can be used in all applications of TextArc to mining text data from any source: literature, stock market and to other aspects of the business world.

We also have regular meetings/discussions with Corinna Cortés (Director of Google Research, New York) and Mehryar Mohri (NYU-Courant) to explore the use of machine learning methods to predict the function of Arabidopsis genes. [I don’t think this is true any more] Our biology-driven problems are similar to those encountered in the analysis of the World Wide Web. Thus, the ideas emerging from our discussions and the results of using the prediction algorithms we are testing could help the general problem of label prediction where the amount of unlabeled data far exceeds that of the labeled training sets.

In addition to our intellectual contributions, our object system architecture and database design is flexible enough that can be used to represent arbitrary data types in non-biological contexts.

**Project Findings**

**Major findings from the activities identified**

We are on track with the accomplishment of our Aims within the time schedule proposed in our three year grant and two year supplement. We have a live version of the VirtualPlant system (http://www.virtualplant.org) that has been operational since June 2006, with various data visualization and analysis tools available. A series of screenshots demonstrating the use of VirtualPlant can be found in http://virtualplant.bio.nyu.edu/screenshots/. The two-year supplement (Years 4 & 5; 08-10) was awarded to enable us accomplish two additional aims related to the original goals of the VirtualPlant parent application to develop tools for: 1. Dynamic Network Modeling, and 2. Comparative Genomics Analysis. Please see sections labeled “Supplement Year 5” for progress made during this past year on these two goals.

**Application of VirtualPlant to develop testable biological hypotheses**: We applied the tools in VirtualPlant to the analysis of microarray expression data, created models, and built hypotheses for four independent studies from which the biological hypotheses generated by VirtualPlant were developed and validated in four proof-of principle studies outlined below.

***The first study*** looked at the interaction between genes regulated by carbon and nitrogen signals (Gutierrez et al. 2007). Here, the VirtualPlant multi-network was used to obtain a global yet detailed view of how genome-wide regulation of gene expression is affected by carbon and nitrogen nutrient treatments. This analysis revealed a set of “molecular machines” comprised of highly connected genes in metabolic, cellular or signaling pathways, whose expression is regulated by C and N metabolites. One such CN-responsive subnetwork is involved in responses to auxin, as it contains 13 genes in the auxin response pathway (including the auxin receptor), 5 auxin efflux carriers and 2 auxin transport proteins. Validation using time-course studies suggest that the phytohormone auxin acts as a regulator of plant growth in response to C and/or N availability. In addition, the CN responsive gene network was shown to contain a significant proportion of regulatory proteins including: 299 transcription factors and 27 genes that are known targets of miRNAs. This latter result implicated a new role for miRNAs in post-transcriptional regulation of gene expression by CN metabolite signals in plants.

***In the second study***, VirtualPlant was used to analyze the microarray data from *cli186,* an Arabidopsis mutant defective in carbon and light signal integration, was analyzed to identify the genes, biological processes and gene networks affected (Thum et al 2008). BioMaps analysis of the list of the 216 misregulated genes showed an overrepresentation of genes in “Energy”, “Metabolism”, and “Photosynthesis”. When these 216 genes were used to query the Arabidopsis multinetwork in VirtualPlant, a subnetwork of 60 interconnected genes was identified, including six transcription factors and their downstream targets in “Amino acid metabolism” (ASN1, GDH1, and ANS1), “Carbohydrate metabolism” and “Glycolysis /gluconeogenesis”

***In a third study***, microarrays were utilized to distinguish genes that respond to organic (e.g. glutamate) and inorganic (e.g. nitrate) nitrogen signals in Arabidopsis. In this study, VirtualPlant allowed the identification of a new molecular mechanism of transcriptional regulation of nitrogen metabolic pathways (Gutierrez et al., 2008). Using the multinetwork tool in VirtualPlant, we generated a hypothesis that was experimentally validated demonstrating that nitrogen acts as an input to the circadian clock via nitrogen regulation of CCA1, a central gene in clock control (Gutierrez et al., 2008).

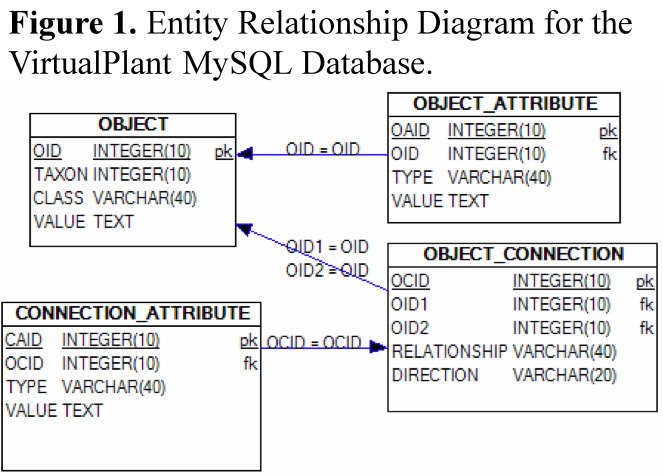
***Finally, in a fourth study***, we used VirtualPlant to identify cell specific nitrogen response networks in plants and identified a role for miRNAs in cell-specific N-signaling (Gifford et al., 2008). In each of these examples, the tools in VirtualPlant allowed the authors (biologists) to analyze their data at a systems level and generated a hypothesis that was experimentally validated uncovering a new mechanism that controls plant responses to nitrogen at a systems wide level.

**Development of the VirtualPlant software platform**. In addition to the significant biological findings uncovered using the VirtualPlant software platform, we have also made major changes to the architecture of VirtualPlant software and implemented new features during the third year. We upgraded the VirtualPlant code from CGI to MODPERL version of CGI, which maintains one instance of the application thus improving the stability and speed of the application. This was a critical and necessary upgrade to accommodate for high web traffic we were attracting and the computer intensive applications available on VirtualPlant. One of the new features we implemented is a method for TRACKING Users and the VirtualPlant functions they execute. As of May of 2010, VirtualPlant has over 700 registered Users since the project first started, and this new tracking tool has enabled us to evaluate and upgrade the VirtualPlant features that the community finds most useful.

**Supplement Year 4: Dynamic Network Analysis**. To enhance our ability to visualize and analyze dynamic networks we have added several features to VirtualPlant as plugins that interface with Cytoscape. Currently, we have developed three plugins: 1) VirtualPlant plugin: allows users to access their VP Cart from Cytoscape, 2) CytoDiff Plug in: allows users to visualize their network based on time series data that they have loaded into VirtualPlant, and 3) VPLayout pulgin: allows users to load a user defined mapping of the network. The details of these three new features of VirtualPlant and progress on the specific areas of the original proposal are described below.

### Aim 1. To integrate genomic data for all Arabidopsis genes.

During the first year of the grant, we constructed a “light-weight” data warehouse with *selected* information about molecular entities (*e.g.* gene models, gene annotation, functional classification), the molecular network in the plant cell (metabolic associations and regulatory interactions in public databases, among others) with experimental measurements (publicly available mRNA measurements from whole genome microarray experiments; over 1500 hybridizations). We continue to add novel data types (e.g. RNA:RNA interactions) and perfect the way the User can access the database. This database is the data platform for the VirtualPlant software system described below. Our VirtualPlant software contains a module that automatically refreshes this data on a regular basis.



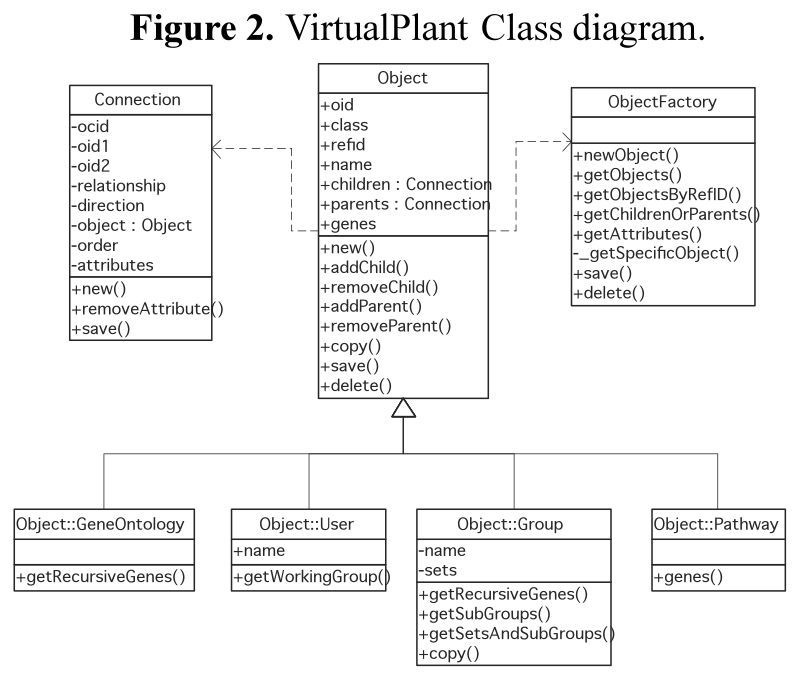
**Database.** Data persistence in the VirtualPlant system is facilitated by the open source MySQL database server version 5.0. This version contains important features, such as nested queries and transactions, which allow us to write few queries and thus facilitate performance and reliability. Our database schema uses a parsimonious design inspired by the LIMBO system (Philippi 2004) and e-commerce sites, with only 4 tables that allows for maximum flexibility to accommodate disparate data types. This schema is sufficient to house most of the data for VirtualPlant, and is easily applicable to many other applications. The object system architecture and database design have proved flexible enough to be used in a non-biological context (not shown). Figure 1 shows the entity relationship diagram for the database. The OBJECT table stores everything from genes, to GO terms, to user accounts. The type of object is distinguished by a CLASS field. The value of this field designates the Perl class used to represent the object in the software. For example, if the value of CLASS is ‘GeneOntology’, the ObjectFactory class (see below) will look to instantiate a new object using the code found in geneontology.pm if it exists, otherwise it will use the default object.pm. The various attributes of the objects are stored in the relational table OBJECT\_ATTRIBUTE, which is connected to OBJECT through the OID field. Attributes are simple name/value pairs and allow for fields like name, description, and any other arbitrary text needed to describe the object. Objects are connected to one another as parent/child through the OBJECT\_CONNECTION table. The object referred to in OID1 is the parent, and that in OID2 is the child. The nature of the relationship is characterized in the RELATIONSHIP table, which typically has values like “GeneOntology2Gene”, which would indicate that the GO term in OID1 is related to the gene in OID2. Since there may be relevant information about the relationship itself, the CONNECTION\_ATTRIBUTE table allows us to store information such as the literature citation that resulted in the connection between a gene and a GO term. With careful attention to the indices and storage parameters of the database, we have found this design to provide high performance with a very small number of distinct queries.

**Supplement Year 4: Comparative Genomics features to VirtualPlant**: By the end of year 3, we started to expand the database schema to allow for multiple species. We wanted a database schema that took advantage of the strategy discussed about but also to be species specific. The solution was to create a different database for each species (ARTH\_DB and ORSA\_DB) and one general database with information not related to any species, GENERAL\_DB. All three databases contain the same tables as discussed above but simply different data. There is also an extra table in GENERAL\_DB for tracking purposes. The GENERAL\_DB contains user specific information and the connection of each user to their carts in the different species. ARTH\_DB and ORSA\_DB contain the cart data and genome annotations, which are discussed in detail below. The advantage to using this setup is that it will be easy for us to setup additional databases as the number of species supported by VirtualPlant increase. Additionally, the database performance for a given species is not penalized for another. For example if someone wants to work on the Rice version of VirtualPlant the users on the Arabidopsis version of VirtualPlant should not affect them.

**Database content.** To facilitate the process of loading and updating the database, we have created a software module that can automatically create or refresh the database. The software downloads a list of files from the sources specified in a configuration file (e.g. The Arabidopsis Information Resource), parses them and then loads them into the database. This configuration file also contains the information regarding how the file should be parsed. Thus, the data loading software is indifferent to the data source and format. If new files need to be added to the database, we can simply modify the configuration file instead of creating a new parser or load routine each time. In addition, the data loading scripts are species-independent. The user needs only to supply the NCBI taxonomy ID in the configuration file to indicate what species should be loaded in the database. Our group focuses on Arabidopsis research; hence our current database contains Arabidopsis data. During the second year, we incorporated small RNA:RNA interaction data in collaboration with Dr. Pamela J. Green (Delaware Biotechnology Institute, University of Delaware). During the third we have incorporated even a larger set of known miRNA interactions from the ASRP project (Gustafson et al 2005). This data is useful to gain insight into post-transcriptional control of gene expression networks. In addition, our database holds the most recently updated versions of: 1) Arabidopsis Annotation version 6 from TAIR (ftp.arabidopsis.org). 2) Gene Ontology terms and their association to Arabidopsis genes (http://www.geneontology.org). 3) MipFuncat functional categories and their association to Arabidopsis genes (ftp.mips.gsf.de). 4) Affymetrix probes from AG and ATH1 chips and their association to Arabidopsis genes downloaded from TAIR (ftp.arabidopsis.org). 5) Biochemical pathways, including enzymes, reactions, and small molecules (ftp.genome.jp). 6.) Protein interaction data from Bind (ftp.blueprint.org). 7) Regulatory interaction data from the AGRIS database (arabidopsis.med.ohio-state.edu/). 8) We have also purchased a subscription to “NASC Affy Watch”, where a DVD is provided containing all the recent publicly available microarray experiments. All the experiments were normalized in R using the gcRMA package (Irizarry, Hobbs et al. 2003). Significant correlation of genes was determined across all the publicly available experiments using Pearson correlation and the standard test for correlation. The results, both normalized and correlation values, are stored in the VirtualPlant database in dedicated tables. Normalized and correlation data can be queried using the Affy probe ID through the VirtualPlant user interface.

The current data housed in VirtualPlant supports many queries. For example, a user may want to obtain a list of genes that are correlated with the *COP9* gene throughout a series of microarray experiments. This is a simple query for many existing databases, but VirtualPlant takes this one step further. We want to know the genes that are correlated with COP9 in the publicly available microarray experiments and what processes these correlated genes participate in. The corresponding SQL query took 0.01s, and suggests that genes involved in Ribosome, RNA polymerase, Pyrimidine metabolism, and Purine metabolism (Gene Ontology categories) are correlated (cutoff ≥ 0.7) with *COP9* across the publicly available experiments from NASC. Moreover, with one click of the mouse, one can start the Cytoscape software (see below) and visually inspect the types of interactions between these genes (e.g. physical interactions for ribosomal proteins, metabolic interactions in the case of pyrimidine metabolism).

**Supplement Year 4: Database Content – expanding species diversity (Oryza sativa):** Rice gene annotations were obtained from the Gramene (Ware et al. 2002). Gramene also provided the Gene Ontology association for the predicted rice genes. We have parsed the data from BIND and KEGG for rice and are in the process of creating the Multi-Network for rice.

**VirtualPlant system architecture.** To allow the highest degree of flexibility and ease of refactoring as the software continues to grow, we have devoted careful attention to the software architecture. The systems architecture for the VirtualPlant software is based on common standards for software development. Given the widespread use of Perl in the bioinformatics community, VirtualPlant is written in OO Perl using a Model-View-Controller (MVC) design and other well-established patterns. The MVC paradigm has allowed the core team of three software developers to focus efforts on their individual areas of expertise. One person can focus on user interaction issues, while another works on the data analysis methods, and the third focuses on data storage and the queries to fetch that data. This allows three different people to work on interlinked aspects of the same feature at the same time. By parceling out the work by skill set rather than by feature, we ensure the best possible code is written for each part of those features. This architecture also allows the GUI developer to work on his part of the system for some feature, even though the database might not be ready to accommodate the data for that feature. We have found this *modus operandi* to be exceptionally productive for our small team of software developers.

**Model.** The software components that correspond to the *Model* portion of the MVC paradigm are those classes that represent the data structures used in the application. For example, the class called Gene represents the data and methods appropriate for modeling a gene. The software for the *Model* is organized into a hierarchy of classes centered on the Object class. Most classes inherit from Object. Figure 2 presents a diagram of the classes used. All Object objects and the subclasses thereof are constructed using the Factory design pattern. This pattern allows us to automatically generate objects of the correct class without having to know in advance which type of object is necessary. For example, this allows us to pass an array of object ids to the ObjectFactory::getObjects method, and receive an array of objects of varying types (i.e., User, Gene, etc.) without having to know in advance which ids correspond to which class. The Factory design pattern also means that we can create and destroy sub-classes of Object at will by creating and deleting files in a certain directory. This flexibility allows us to create complex new functionality without having to change large amounts of code. We have also employed a lazy instantiation scheme to reduce the number of queries that are run for any given page request. The attributes, parents, and children of an object are never populated at the time the object is created, as very often this information is not needed. For example, in the search results, the only information needed is the id and name of the object. Instead, any attempt to view or change the attributes, children, or parents triggers a function that populates that data on demand. In this way, the object itself determines which data it needs, rather than relying on the programmer to pre-populate it with data that may or may not be required in the current context.

**View.** The *View* component of the MVC architecture is implemented through a combination of one class, called Output.pm, that is part of VirtualPlant, and a templating system, called HTML::Template that is publicly available from CPAN. *Model* functions process any data as necessary and pass that data, usually in the form of object references, back to the *Controller*, which directs the data to the appropriate function in Output.pm. Output.pm then reformats the data for processing by the templating system. The templating system expects to receive the name of a template file and a hash of name-value pairs. The template files are plain HTML files with certain specialized markup recognized by the HTML::Template. This system means that our output markup and data processing logic are completely divorced from each other. Separating data processing logic from output provides us with two benefits. First, it is trivial to change the look-and-feel of the output at any time, without having to know too much about the data that drives that output. We can take advantage of team members with skills in HTML, but not in Perl code, to help with the software development effort. Second, it means that we can easily provide output in more than one format. For example, it would not take much to provide a mobile-device version of VirtualPlant (*i.e.*, mobile phone, or PDA version), simply by providing alternate templates, and providing directives to the controllers to use them.

**Controller.** The *Controller* components are those Perl scripts that directly take input from the web browser and coordinate the actions of the *Model* and *View* components. The main script that drives VirtualPlant, called virtualplant.cgi, contains very little code itself. After determining some conFiguration parameters and establishing a connection to the database, it uses a parameter, called action, which it gets from the query string or posted form data, to determine which of the controllers to invoke. A call to virtualplant.cgi?action=analysis will result in the passing of control to Analysis.pm. These sub-controllers then look for a parameter called cmd to determine which of their functions to call. The URI virtualplant.cgi?action=Output&cmd=viewGeneSet will result in the invocation of the function viewGeneSet() in the file Output.pm, which displays the details for a particular gene set. Once the *Controller* Figures out which of its functions to call, that function is invoked and becomes responsible for determining which *Model* classes and display methods need to be called. For example, in the call to viewGeneSet, the *Controller* will try to fetch the GeneSet object from the database. However, if no set ID is provided on the query string, the *Controller* will display an error screen, instead of the details page for viewing that gene set. This ability to switch to different *View* components based on the results of actions by the *Model* components allows us to easily construct a large complex application with fairly minimal code.

**MODPERL.** In keeping with our ongoing efforts to improve end-user experience and ensure that VirtualPlant is as extensible as possible, during our third year we transitioned from vanilla CGI to ModPerl. Traditionally CGI applications function independently from the HTTP server. During the request/response cycle, the application is launched as a separate process. All necessary environment and request information is passed to the CGI application via standard input, and the CGI response is communicated via standard output and standard error. In a ModPerl conFiguration, the Perl interpreter is embedded within Apache, enabling the server to cache code, and serve replies entirely from volatile memory. More importantly, this enables us to insert and retrieve information from the Apache request/response cycle however our design patterns dictate.

With the Apache application programming interface (API) at our disposal, we have improved user authentication, session processing, header and cookie parsing methods, inter-module communication, application-wide security features, and transitioned to Perl's taint mode. ModPerl has also enabled us to further optimize our database. With a persistent interpreter all database queries can be funneled into a single database connection, eliminating startup and tear-down overhead and providing us with the opportunity to increase MySQL buffer and cache sizes. In all, responsiveness has improved in the range of 200%-2000%.

**TRACKING.** In order to determine how active our over 500 registered users are and which of the tools are their favorites we created a tracking table in our database. All actions that users perform are recorded along with the time of event. This tells us how often and when the users log into VirtualPlant. This also tells us the different applications that they are using. Examples of the different type of queries that are possible are shown in the Figure 3. The Pie Chart in Figure 3a shows the distribution of the number of times each function was called. The four most popular functions in VirtualPlant are “Intersection”, “BioMaps”, “Sungear”, and “GeneNetwork”. The bar graph in Figure 3b shows that an average of 40 unique users have “logged in” each month in the past 4 months. The total number of unique visitors in the last 4 months is 87, which means that there are several users that consistently log in every month. Future enhancements to the TRACKING feature is the 1.) “UserHistory” so users can go back and look at the previous commands they have executed and 2.) Graphical User Interface for an Administrator which will query the TRACKING table for useful statistics like the ones in Figure 3.

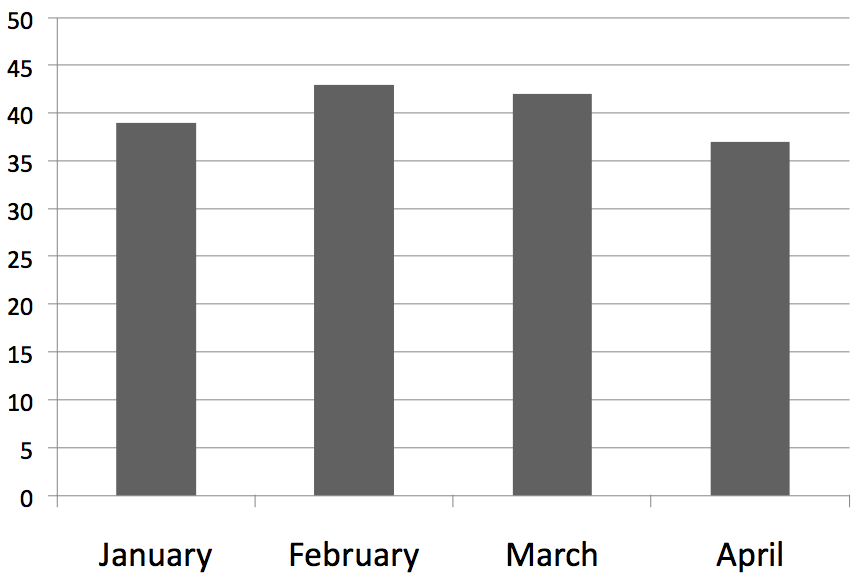
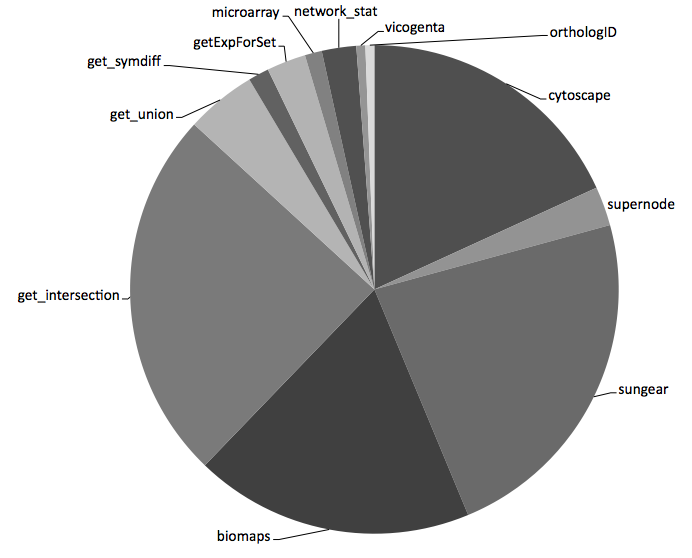


Figure 3

A

B

**Tutorial.** To make the VirtualPlant website easier to use, we have started to create a tutorial with step-by-step examples of how to use the different tools on VirtualPlant and also how to use the different features. The tutorial is accessible from the home page by clicking on the “Help” icon. Currently they are static html files with instructions that a user can click through, however we are in the process of creating step-by-step instructional videos as well.

**Supplement Year 5: Stress Testing of VirtualPlant:** As the numbers of users of VirtualPlant grows (currently 500 registered users), we want to make sure that VP can handle the load of all the users and also the load of large datasets. To this end we have performed stress testing where we created a program that heavily loads VirtualPlant and measures its performance. The program is built on Perl’s HTTPD::Bench::ApacheBench module. It stresses VirtualPlant by issuing multiple, concurrent HTTP Requests to the same tool. It measures the response time for each request. We can increase the rate of requests until the server becomes overloaded, as shown in Figure 4. This helps us determine the load our tools can handle.

We’ve addressed performance problems identified by stress testing in several ways: 1) offloading computation to the cluster, 2) re-designing slow algorithms, 3) limiting input sizes, and 4) re-coding slow implementations.

Figure 4: Response time of VirtualPlant as a function of the number of concurrent requests to BioMaps. Our stress test program produced this data, by issuing a range of concurrent requests to BioMaps and measuring VirtualPlant’s response time. Each request asked for a BioMaps of a set of 200 genes.

**Supplement Year 6: Semi-Automated Testing of VirtualPlant using Selenium:** Complementary to the stress test performed by our program using HTTPD::Bench::ApacheBench, we are using Selenium to help us create a semi-automated method for testing all the applications on VirtualPlant. Selenium is a suite of tools for automating web application testing. Selenium's plug-in for Firefox can "record" the interaction between a user and Firefox. Then, the interaction can be "played" back and the output is compared with the output on the recorded interaction. After comparing both outputs, Selenim's plug-in indicates whether both are similar or different. Different outputs are marked as an indication for possible errors.

A sequence of small interactions with the web application VirtualPlant have been recorded and successfully used to detect changes in its behavior. Selenium is essentially being used as a simple "regression testing" tool. It is expected that by extending the set of test sequences VirtualPlant will continue its development at a faster pace and with a higher level of quality assurance than before the utilization of this automated testing tool.

**Supplement Year 5:** **Virtual machine port of VirtualPlant**: VirtualPlant uses over 3 dozen software systems, including over 30 Perl modules such as RSPerl, BioPerl, GraphViz and DBI, and many other systems such as Apache, MySQL, modperl, Java, R and Tomcat. As the system is deployed on only a single web site, it has not been worth our effort to create an automated installation program. However, demand has arisen for two types of additional VirtualPlant installations: 1) development and testing environments, and 2) deployable installations that could be provided to private users, such as firms and labs with high secrecy requirements. To meet this need we've ported VirtualPlant to a virtual machine (VM) that we can replicate as needed to meet these demands. The installation's stack is VirtualPlant code, on the software systems above, on Ubuntu 9.04, on VMWare Workstation. However, to reduce the load inside the VM, and share data between several VMs, the VM accesses an external MySQL server. If we are satisfied with the perform of VirtualPlant in the VM and we deploy it to corporate users then we will consider running all production VirtualPlant installations inside the VM so that we only need to support a single version. Think of this as a low-effort partial productization tactic.

**Supplement Year 6: Virtual machine port of VirtualPlant:** We successfully deployed Virtualplant 1.1 and VirtualPlant 1.2 on a VM. We found the performance lost using VM is minimal, therefore the benefits outweigh the limitations of running VirtualPlant on a VM.

**Supplement Year 5:** **Construct a VirtualPlant compute job service on the cluster**: Several of the VirtualPlant analysis programs are very compute-intensive and embarrassingly parallel. For example, correlation takes *O(n2)* time where *n* is the size of the gene set being analyzed, and biomaps takes *O(n+t)* where *t* is the number of GO terms in the gene set.

We've purchased an 8-node, 64-core cluster with NSF funds. The cluster runs the Rocks cluster management software and the Sun Grid Engine (SGE) job scheduler. We've architected and prototyped a service that works as follows:

1. VirtualPlant receives a web request that launches a compute-intensive job.
2. VirtualPlant sends the request, including the program identifier and its inputs, to the compute job service.
3. VirtualPlant notifies the user that the request may take time to service.
4. The compute job service transfers inputs and parameters to the cluster, queues the job in SGE, monitors the job's execution, and returns the results (or error output) to VirtualPlant.

We will soon beta-test this system on some of our production analysis traffic.

**Supplement Year 5:** **Parallelization of Correlation of Gene Expression Measurements on the Cluster**: One of the most compute-intensive analyses offered by VirtualPlant is correlation of gene expression measurements because the compute time takes O(n2) time where n is the size of the gene set and a gene set may contain 1000s of genes. For example, it takes over 10 minutes to compute the correlation of 4000 genes. While correlation is straightforwardly parallelized, implementation on a cluster still presents some important design decisions:

1. How should tasks be started in parallel and input data be distributed to nodes?
2. How should correlation pairs be mapped to nodes? Can the sum of the size of the input over all nodes be minimized?
3. How should results be merged into a single output file?

Our parallel correlation performs quite well. To minimize response time it parallelizes correlation's startup (task initiation and input data load) and completion (multi-stage parallel merge of results). Measurements show a near linear speedup of correlation computations for gene sets of 1000 genes on 56 cores. These results have been reported in a paper submitted by Wang and Goldberg to the Fourth International Workshop on Scientific Workflows. Figure 5 shows preliminary results.

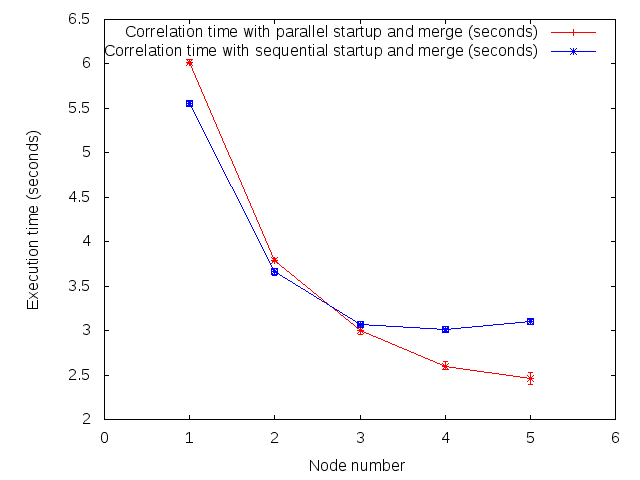


Figure 5: Response Time of Correlation on the Cluster for 1000 Genes as a Function of Parallelism. Performance improves as the startup and merging of results are parallelized.

**Supplement Year 5:** **Multi-species data load**: We're expanding VirtualPlant to support multiple species: One of the goals of the two year supplement was to extend the capabilities of VirtualPlant to comparative genomic analysis. To this end we have enabled VirtualPlant to support multispecies analysis, as described for the fully sequenced species below:

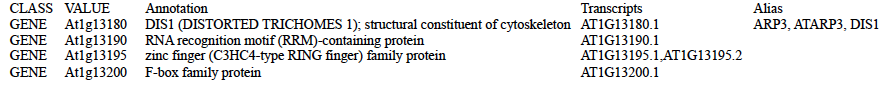
| **Common name** | **Data source** | **Genus** | **Species** | **Ecotype** | **GO annotated** |
| --- | --- | --- | --- | --- | --- |
| Thale cress | TAIR | Arabidopsis | Thaliana | Columbia | True |
| Rice | MSU Rice Genome Annotation Project | Oryza | Sativa | Japonica | True |
| Grapevine |  | Vitis | Vinifera | TBD | True |
| Poplar | Plantgdb.org | Populus | Trichocarpa | TBD | True |
| Sorghum |  | Sorghum | Bicolor | TBD | True |
| Purple false brome |  | Brachypodium | Distachyon | TBD | False |
| Moss | Joint Genome Institute | Physcomitrella | Patens | TBD | True |
| Spikemoss | Joint Genome Institute | Selaginella | Moellendorffii |  | False |
| Corn |  | Zea | Mays | TBD | True |

To enable this expansion of VP to multispecies, we built software that downloads genomes from the Internet and processes them for convenient installation in VirtualPlant's database. The processing extracts gene names and descriptions. It also obtains GO terms, KEGG pathways, the relationships between pathways, and the genes that encode for proteins used in each pathway.

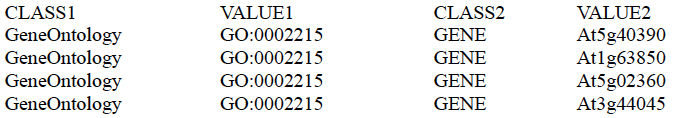
We have also architected and written code that computes pairwise BLAST match strength between all genes in a pair of genomes, and stores strong matches in VirtualPlant's database. This code runs on the cluster. We will integrate these matches with orthologs among the same set of species, as determined by OrthologID. These steps are preparing us for full multi-species support in VirtualPlant.

**Supplement Year 6: Semi-Automated process of creating a species database:**

In order to facilitate creation of new species databases we decided to define two types of files VPObjects and VPConnections. We have developed software that will take these two files and create the VirtualPlant database for the species and prepare its corresponding files for use by VirtualPlant tools such as BioMaps and Sungear. VPObjects defines different elements, such as Genes and GeneOntology terms, with their attributes. VPObjects is a tab-delimited text file where the first column defines the CLASS, for example Gene, GeneOntology, followed by some unique identifier, such as the AGI locus ID from TAIR for Arabidopsis. The following columns can be different attributes and the type of attribute is defined by the column header. See table below.



VPConnections defines relationships between the different elements. These can be associations such as GeneOntology terms associated to Gene ID or Parent-Chjld relationships as found in GeneOntology. VPConnections is also a tab-delimited file where the first column is the Class of the first element (parent), followed by the unique identifier. The next two columns are the Class of the second element followed by its unique identifier. See below for example of GeneOntology to GeneID connections file.



After the files are loaded into the database a perl script creates necessary files for Sungear and BioMaps.

These short steps are enough to have a functional species database.

Documentation describing the two files in detail can be found on http://virtualplant.bio.nyu.edu/docs/

**Supplement Year 6: Semi-Automated process of updating a species annotation:**

Using scripts described above we were able to create instances of Tair9 and Tair10. In addition we also created a tool in VirtualPlant that allows for users to convert their Tair8 gene lists to Tair9 and Tair9 to Tair10 using files provided by TAIR. The tool warns the user if there are any changes to annotations of genes in their list(s). This gives the user the option to accept or reject the changes.

**Aims 2 and 3: To visualize, analyze and summarize multivariate data for the generation of testable biological hypotheses**. During the second and third year of this project, we continue to develop and integrate new software tools that allow users to access, query, analyze and visualize the data stored in our data warehouse to generate biological hypotheses. Because our tools integrate data analysis with visualization, we present progress on **Aim 2** (visualization) and **Aim 3** (analysis tools) together in the following sections.

The main (but not exclusive) entry point to the VirtualPlant system is through a web-accessible Perl program (available from http://www.virtualplant.org). The user can navigate the system through a Graphical User Interface (GUI) and easily access methods to visualize and/or analyze the available data. Whenever possible, the data is presented in various “biology-centric” views (*e.g.* hierarchical, interaction) accessed from individual windows that are inter-linked. Selection of the data in one view updates corresponding data in the other views. Biological entities (e.g. gene products and metabolites), gene properties and/or functional relationships are readily accessible from these views for molecular level analysis. VirtualPlant has been designed using the familiar paradigms of an e-commerce site. By adding genes to a “cart”, users can perform a number of “checkout” functions. These functions are the various data analysis and visualization tools. Most checkout functions allow the user to send the results of the analysis back to the cart for further processing. These tools are described below.

**Set Operations.** We support the set operations union, intersect, and symmetric difference. All set operations can be simultaneously performed on 2 or more sets. A union combines the contents of two or more sets to create a new set. An intersect is the set of genes common to all queried sets. The symmetric difference is the union minus the intersection. In addition to the combinatorial operations, sets can also be renamed, deleted, or assigned to a “group”. Groups, functioning like folders in a file system, can be arbitrarily deeply nested and are used to organize gene sets. For example, if the user is working on two different projects, the sets for each project can be organized into separate groups.

**Supplement Year 5: Probes:** A user can upload a list of Affymetrix ATH1 microarray probes. VirtualPlant converts the list into a list of genes, and adds it to the user’s cart. The user can review the probe to gene conversion. Also, the correlation tool has been extended to calculate the correlation in gene expression among sets of probes.

**BioMaps.** BioMaps takes one set of genes and analyzes the contents of that set for over-represented terms (GO or MIPS functional terms). The data is presented as a table, showing the GO terms that are over-represented and the list of genes annotated to each term. Additionally, a graph image is presented that shows GO terms as nodes, with the relevant genes attached to them. A p-value cut off can be set to determine which GO terms are significant, and different annotation ontologies can be used.

**Supplement Year 5: Biomaps:** BioMaps has been improved to support two additional name spaces in the GO Hierarchy, MIPS Funcat, and Oryza Sativa gene annotations. This version can be easily configured to handle other ontologies and species.

BiomapNSF.tiffFigure 6: Interactive BioMaps. This BioMaps runs in ActionScript. We see the GO terms that annotate a list of 30 genes, the start of which appears at screen bottom. Term colors indicate their p-values, as shown in the scale in the upper left corner. The ‘transporter activity’ node is selected, and all terms and edges on the path from the root to its deepest children are highlighted.

Additionally, we have prototyped a new fully interactive version BioMaps in Flare, an ActionScript library for visualizations that run in the Adobe Flash Player. This tool supports many interactive operations to enable the researchers to interactively analyze the biological significance of gene lists. It inputs a gene list and loads the subset of the GO DAG that represents all GO terms that annotate genes in the list. Each GO term is labeled with a p-value, the probability that it might appear with the observed frequency in the gene list by chance. BioMaps Flare displays the DAG using Flare’s tree layout algorithm.

Users can zoom the GO DAG to any level of detail, and select GO terms. They can also query the GO terms and use union, intersect, and difference set operations to form new genelists and send them to the cart. Also, they can adjust a slider that sets a p-value threshold to which hides GO terms whose p-values exceed the threshold. A png image of the biomaps DAG can be saved.

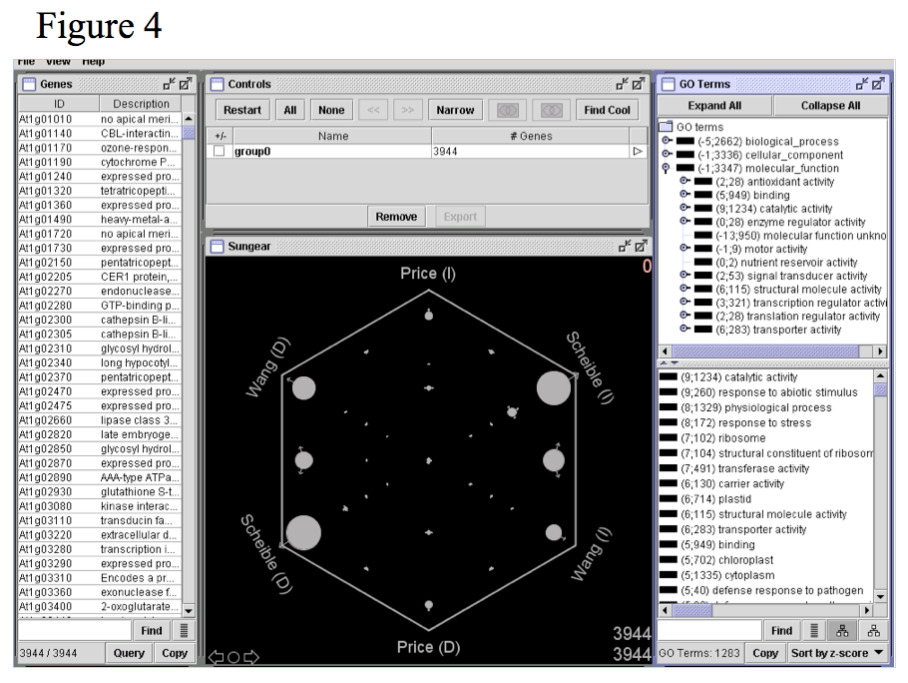
**Sungear.** Many software tools are available to analyze genomic data and other large data sets, but no existing tool supports a rapid, visually interactive and biologist-driven exploration of natural questions on many experiments at a genomic scale. Sungear is a tool built for just this purpose (Poultney, Gutierrez et al. 2007). Simple operations executed using Sungear enable the user to identify genome-wide responses that are robust across a series of microarray experiments, while linked GO annotation features allow the user to develop biological hypotheses about the processes that respond in these data. Sungear can also be applied to compare genomes, for example to quickly uncover the patterns of conservation of gene function across the tree of life. In addition to its biological applications, Sungear can be applied to any area involving comparisons of multiple large data sets, as shown by a case study for the analysis of baseball statistics (Poultney et al., 2007). The Sungear interface presents four windows to the user: 1. the Sungear plot, 2. the Gene list, 3. the GO terms, and 4. the navigation/export controls (Figure 7). These four windows are linked with one another, so that selections in one window will immediately be reflected in the other windows. Figure 7 shows the use of Sungear to represent regulated genes from Arabidopsis microarray experiments carried out in three different laboratories. The experiment names are listed around the polygonal vertices (hereinafter called “anchors”), and these names are linked to the lists of genes in those experiments (e.g. regulated genes as discussed below). A “gear” or "vessel" is a circle within the polygon with arrows pointing to one or more anchors. The size of a vessel, for example pointing to anchors A1 and A2, is proportional to the number of genes in experiments A1 and A2 but not other experiments. The location of the vessel within the polygon is largely determined by the position of all anchors with which it is associated. The number of vessels that could be present corresponds to all possible subsets of the experiments, including the null subset. So, if there are X experiments (represented by an X-gon) there can potentially be 2X vessels. However, the actual number of vessels that will be visualized depends on the dataset. For example, there may be fewer than 2X vessels, because some possible intersections may contain no genes. Whereas the Venn diagram representation does not extend beyond X = 3, Sungear can represent an arbitrary number of experiments/lists, depending only on the researcher's willingness to understand a visual display having many anchors and associated vessels.

Figure 7: Sungear.

The Sungear display offers visual and quantitative information about the numbers of genes that are, for example, similarly regulated in any combination of the experiments analyzed. Thus, a cursory look at the Sungear window can quickly answer a common question posed by biologists when analyzing expression data. In addition, Sungear provides other views of the vessel contents, including gene lists (left side in Figure 7), a Gene Ontology (GO) (Ashburner, Ball et al. 2000) hierarchy (upper right in Figure 7), and a list of over-represented GO terms (lower right in Figure 7). In Sungear, one or more vessels, genes, anchors and/or GO terms may be selected for analysis at any time. Because of the associated GO annotations, Sungear can also be used to query for the regulation of specific processes. When querying, it is useful to distinguish Boolean "and-functionality" from "or-functionality". And-functionality arises, for example, when a researcher wants to select all those genes that are in vessel <X> and also satisfy GO term <Y>. By contrast, or-functionality would be used when, for example, seeking genes that are either in vessel <X> or satisfy GO term <Y>. Using a familiar combination of mouse clicks, shift and alt keys, Sungear can readily support these queries as well as many other logical operations including exclusion (i.e. genes in experiment A and B, but not experiment C).

In the second year, we expanded the use of Sungear to a number of different organisms, i.e. Oryza sativa, and demonstrated that it can be used for any organism and hierarchical grouping of genes, and indeed for non-biological applications as well. We worked to make Sungear available to several other labs. We also used Sungear in a new way to find common features among similar experiments (i.e. with same experimental factor) from different laboratories. We also made a number of technical improvements to Sungear. The major improvements were: improved communication with external programs (including, but not limited to, VirtualPlant); better set analysis tools and visualization controls; and a host of interface improvements designed to make the tool easier for novice users.

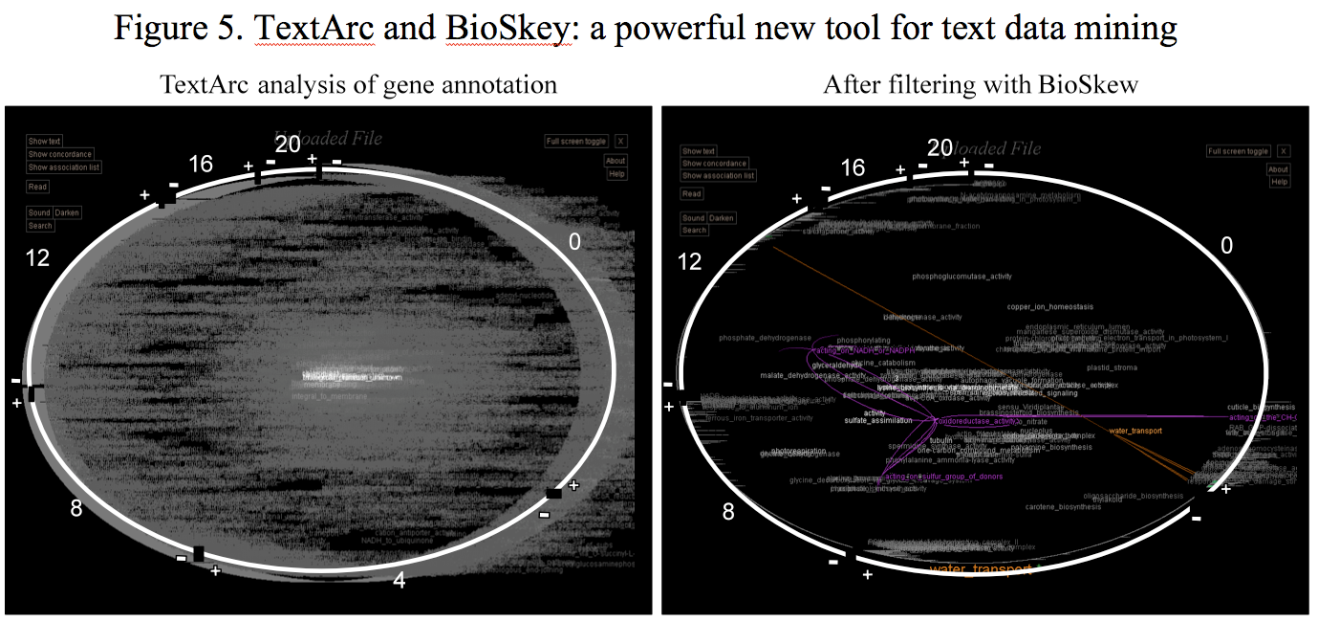
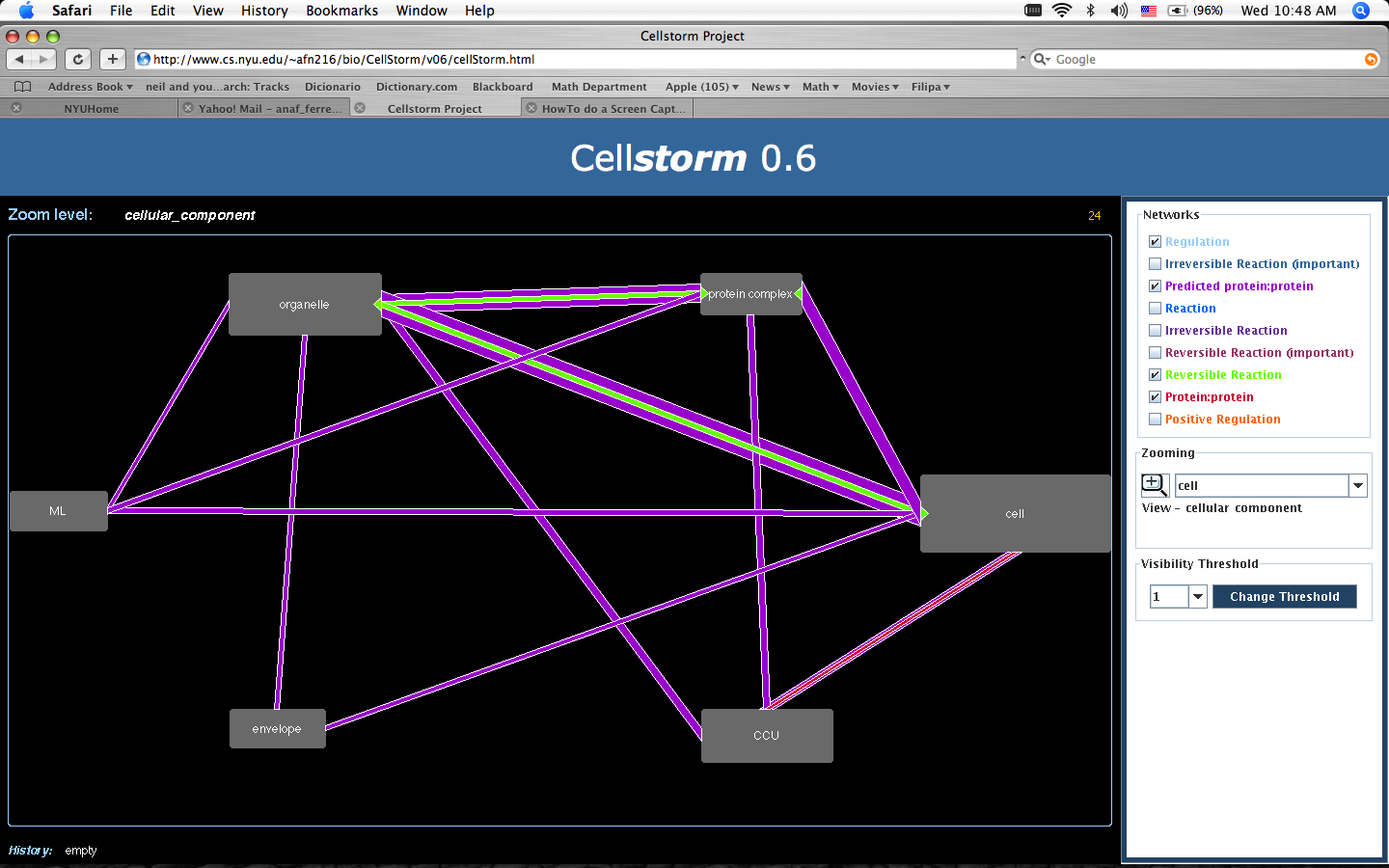
**BioSkew.** BioSkew is a novel analysis tool that measures the “skewness” of associated text in ordered lists. It takes in lists of ordered sets of words and statistically analyses it for words that may not be randomly and evenly distributed either within each ordered list or across all the lists. For example, the tool can be used to find GO terms that are non-randomly distributed in an ordered list of regulated genes from microarray data across or within different treatments. While microarray data with genes ordered by regulation level and associated with GO terms is an immediate and easy example, the program algorithm is general enough to accept text of any form provided it is ordered in a meaningful way. BioSkew calculates the cumulative deviation for each word, from and evenly distributed distribution, for each ordered list (for within-treatment analyses) and/or over all the lists (for across-treatment analyses). The significance values are estimated by comparing the observed discrepancies to the distribution obtained from recalculating the discrepancies on randomizing the word sets of the list. TextArc, a visualization tool developed by Brad Paley, is being used as a front end for these statistical analyses (Figure 8). TextArc takes in formatted text and places it on the perimeter of an ellipse. The words are arranged inside the ellipse depending on the positions of the instances of each word on the perimeter. TextArc can be used in conjunction with the BioSkew to view only words that are statistically significantly skewed. Figure 8 illustrates the use of BioSkew/TextArc to analyze microarray experiments that were performed with plant tissue samples harvested at different times of the day. The numbers in the outside of the list indicate hours after dawn. The white arcs correspond to the beginning and end of each data set. The plus and minus sign indicate the order of the gene lists from the most induced to the most depressed respectively. The panel on the left represents all the text annotation associated with the genes regulated at the different time points. The panel on the right shows those words that were significantly skewed (p<0.01) as determined by BioSkew. TextArc allows the user to click on individual words to find their location on the datasets (orange lines). In this case, we find that genes annotated to “water transport” are skewed, as they are among the most highly induced at the beginning of the day. TextArc also identifies related terms (pink arcs) based on their pattern of occurrence on the gene lists.

Figure 8: TextArc and Bioskew: a powerful tool for text mining.

**CellStorm.** Within eukaryotic cells, the activity of gene products is specific to subcellular components where proteins and other gene products (e.g. small RNAs) interact with one another through transcription control, protein:protein interactions, RNA:RNA interactions, metabolic relationships and others. In the second year, we began the design and implementation of new software called “CellStorm”. The goal of CellStorm is to give a visual representation indicating which subcellular components each gene is active in and the network interactions among genes in different subcellular components. Because there are thousands of such interactions and some (like metabolic ones) are directional, CellStorm gives size and color cues to indicate the type and quantity of edges between subcellular components as well as arrowheads to indicate directionality. Figure 9 shows a typical screenshot of the CellStorm program when analyzing a list of 500 genes. The image shows the gene ontology terms from the cellular\_component ontology that have at least 1 gene annotated from the input list. These subcomponents are represented by grey rectangles with the corresponding label. The size of the rectangle is proportional to the number of annotated genes. So, in this case, the subcomponent "cell" has more genes from the input list, followed by organelle and so on. The image also shows the interactions between the genes annotated to each term. The interactions are selected on the right side panel. In this case, the network generated contain the following interactions: Regulation, predicted protein:protein, reversible metabolic reaction, and known protein:protein. The gene networks are generated with the gene network tool described below. The thickness of the line that connects to rectangles is proportional to the number of existing interactions (i.e. gene pairs). In this case, the "Predicted protein:protein" between organelle and cell is the link that contains by far more interactions, followed by organelle - protein complex. CellStorm is not yet integrated with VirtualPlant.



**Figure 9.** Network highways that within the Cellular Component Ontology

**Networks.** VirtualPlant currently supports two main types of networks: 1.) Gene networks: these are networks that include genes and metabolites as nodes and all the interactions available in the public domain (metabolic, regulatory, physical, etc.) 2.) Super node networks: these networks contain nodes that group one or more genes based on their function.

**Gene Networks.** Sets selected for analysis, generated with any of the tools in the VirtualPlant system (e.g. Sungear, BioMaps) can be combined with information about metabolic, protein-protein, and other types of molecular interaction data to generate gene networks. A form is provided by VirtualPlant that allows the user to select the different types of interaction to be used to create the network graph. The form also allows users to use correlation values from selected experiments to filter their data. VirtualPlant generates network files and automatically launches the Cytoscape software (Shannon, Markiel et al. 2003) with the network and supporting data using Java Web Start technology. This allows the user to use all the powerful features available from within Cytoscape (e.g. plugins for data analysis) for network analysis.

**Supernode Networks.** In many cases Gene Networks can be large and complex. To simplify the analysis of large gene networks we have created the Supernode Network tool. Supernode analysis allows the User to group genes based on their function/annotation. Users can currently group genes based on pathway annotation from KEGG, or ontology annotation from the Gene Ontology Consortium. For example, grouping genes by the KEGG pathways will generate 1 node that represents all the genes that belong to the same pathway in the KEGG database (*e.g.* nitrate assimilation). This tool is extremely useful, because it greatly reduces the complexity of networks generated with large sets of genes. It also allows the user to determine the regulatory interactions that may be controlling a specific pathway. The VirtualPlant plugin for Cytoscape allows the User to send the genes represented by 1 or more supernodes back to the VirtualPlant gene cart.

**Supplement Year 4: Cytoscape Plugins for connecting to User’s Cart in VirtualPlant:** We have created a plugin for Cytoscape that allows Users to log into their VirtualPlant cart. This plugin enables the user to

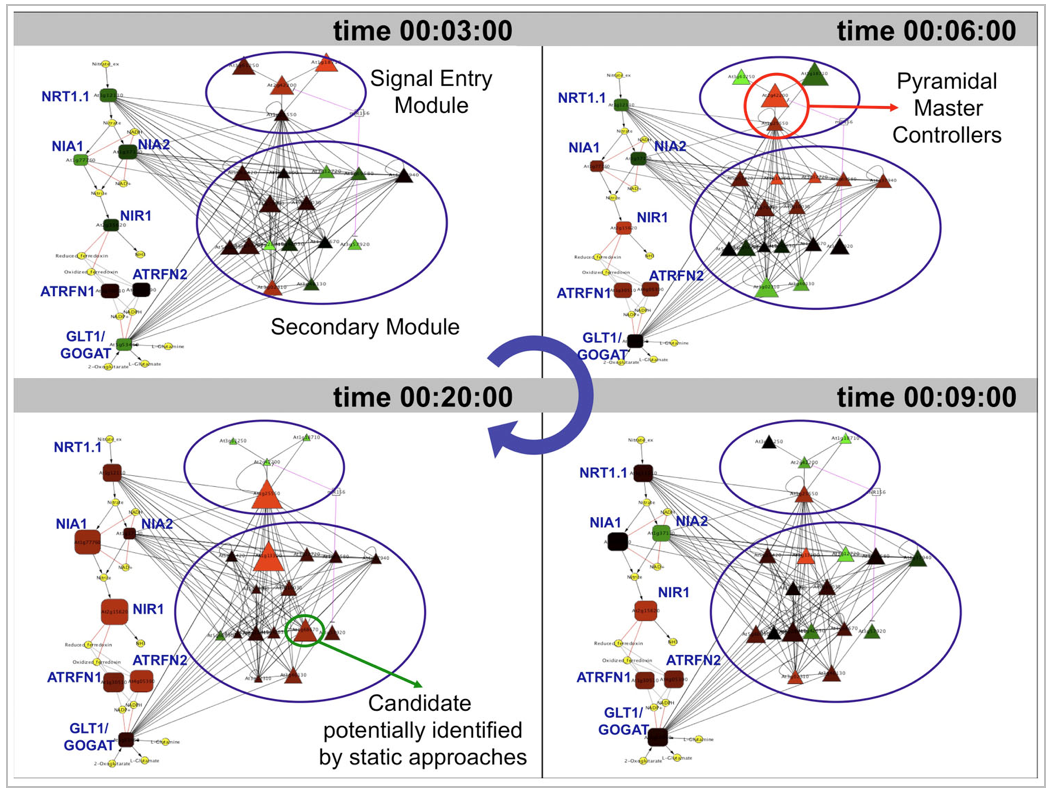
a) Send lists of selected genes back to VirtualPlant for further analysis of this new set of genes.

b) Highlight nodes in their network by selecting a gene list from their cart.

c) Load their experimental data to be used in the Cytodiff plugin described below.

**Supplement Year 4: Cytoscape Plugin for Vizualizing time series data:.** CytoDiff is a plugin for Cytoscape that allows quick visualization of changing gene expressions between microarray experiments in any gene network. Given a gene network and gene expression values CytoDiff performs a statistical analysis on the expression values and visually maps this analysis onto the network in the form of a color gradient and size change. A gene whose expression was statistically repressed between two experiments will be clearly shown as a small green node, while a gene that was greatly expressed would be a large red. Any gene whose expression change was not significant becomes a more muted color and smaller so that significant changes stand out. The analysis is done using the log ratio of the gene expression of the gene in the treated sample versus the control. This tool is particularly useful when comparing experiments or comparing a set of time course experiments where changes in the network can be seen and a flow of expression values seen. In the fourth year, we expanded this plugin to support time series data. The user is essentially saving their statistical comparison in what we refer to as a slide. Once the user has created several slides they can play it back like a slide show presentation. An example of the utility of Cytodiff is shown in Figure 10. In this example, we analyzed a fine scale time series data of plants treated with Nitrogen for 3, 6, 9, and 20 minutes. After determining which genes are regulated by Nitrogen, we asked how this N-regulatory network changed over time. We compared gene expression across two time points sequentially (e.g. 0/3min, 3/6min, 6/9min, and 9/20min) and used Cytodiff to visualize the changes. Cytodiff displays changes in gene expression by changes in node size (larger=higher expression) and color (red=induction, green= repression, grey=no change) (Fig. 10). This dynamic network visualization enabled us to identify an N-signal entry TF module consisting of three interacting TFs induced by nitrate at a very early time-point (3 min) that are positioned as potential “pyramidal master regulators” of a downstream module of interacting TFs, which we refer to as “secondary modules”. Based on their predicted position at the top of this dynamic regulatory network, we hypothesize that T-DNA knockout mutants of these proposed “master regulators” will have the most influence on the nitrate-response response of the network.

Figure 10: Using Cytodiff to visualize time series data



**Supplement Year 4: Cytoscape Plugin for changing layout based on PCA output: VPLayouts:** The layouts plugin allows users to upload a file of coordinates defining the position of all the nodes in the graph. It’s a simple plugin but it opens up a window of many possibilities. We were inspired by the idea that we may want to display our network in a layout that was determined by a statistical analysis tool such as principal component analysis (PCA). Now we can concentrate on creating layout algorithms that will help us visualize and study the data in different ways.

**Comparative Analysis:**

**Vicogenta and OrthologID.** Vicogenta (Katari et al, unpublished) and OrthologID (Chiu, Lee et al. 2006) were created by members of the New York Plant Genomics Consortium for use in their NSF Grant “Genomics of Comparative Seed Evolution” (NSF Plant Genome Grant: DBI-0421604). The integration of these tools into VirtualPlant has tremendously aided the members of the Consortium in their work. Vicogenta (Viewer for Comparing Genomes to Arabidopsis) stores top blast matches of other plant species against Arabidopsis. A list of genes can query the Vicogenta database to identify which genes have matches to the species selected. Since it is based on the GMOD database the User can visualize the alignments using GBrowse. A list of Arabidopsis genes common among the species selected can be sent back to VirtualPlant for further analysis. OrthologID is a database of orthologous genes determined within a character-based phylogenetic framework. The output is a gene family tree of Orthologous genes from completed genomes which currently includes Arabidopsis thaliana, Oryza sativa, Populus trichocarpa and Chlamydomonas reinhardtii as the outgroup. The interaction between Virtualplant and Vicogenta and OrthologID are still in the preliminary stages. They will evolve considerably as we start adding more genomes to VirtualPlant.

**Supplement Year 5: Comparative Genomes and BLAST** – We added a function to VP that enables users to use BLAST to align sequences to genomes and proteomes of Arabidopsis and Rice. Users provide protein or nucleotide sequences from any species in FASTA format, a GenBank Accession number or GI numbers. If provided in either of the latter two formats, VirtualPlant retrieves the sequences from the GenBank database at NCBI via a web service. Once the sequences are obtained, a genome wide BLAST is performed again the currently selected species within VirtualPlant. Several blast programs are available, including blastn, blastx, blastp, tblastn, and tblastx. Other options provided are the e-value cutoff and top n hits. After the results are obtained they are displayed in a web form from which the user can select genes to form a new gene list in the cart.

**Microarray data analysis.** We have purchased a subscription to “NASC Affy Watch”, where a DVD is provided on a regular basis containing all the recent publicly available microarray experiments. All the experiments were normalized in R using the gcRMA package (Irizarry, Hobbs et al. 2003). Significant correlation of genes was determined across all the publicly available experiments using Pearson correlation and the standard test for correlation. The results – both normalized and correlation values – are stored in the VirtualPlant database in dedicated tables. Normalized and correlation data can be queried using the Affy probe ID through the VirtualPlant user interface. The graph is created using FLEX, a state-of-the-art web application development software that greatly facilitates presenting and visualizing complex data. Currently, the VirtualPlant database contains 3140 hybridizations that correspond to 298 experiments from the public domain, spanning a wide range of experimental factors. The microarray data includes important data sets such as those generated by the AtGenExpress project. The publicly available microarray data has been normalized in the VirtualPlant database, allowing researchers to skip the normalization step when performing an analysis on these slides.

In addition to the public domain data, users can upload their own data in the form of Affymetrix CEL files. We are using the open source celutil (http://www.bioinformatics.org/celutil/) application to extract metadata from CEL files, and where necessary convert to uncompressed ASCII data to allow us to parse the intensity matrices into Perl data structures for normalization.

VirtualPlant supports a variety of methods for the determination of differentially expressed genes in microarray data. Using normalized data, VirtualPlant currently supports 5 different statistical analysis methods for determining differentially expressed genes including the popular log base 2 ratio, t-test with correction for multiple testing and Rank Products. Three of the methods implemented leverage on the multitest package (Dudoit, van der Laan et al. 2004) available from the Bioconductor project and implemented for the R programming language. This list of differentially regulated genes is automatically added to the user’s gene cart from where the user continues analyzing their data with the tools described above.

For processor-intensive actions such as data normalization or the statistical analysis to determine differentially expressed genes across several slides, we have implemented an asynchronous queuing system to allow users to continue to use the VirtualPlant application while long-running procedures complete. Upon completion of these actions, the user is notified via email and is directed back to the cart to view the results of their analysis.

**Supplement Year 5: MultiExperiment Viewer (MeV)** (Saeed et al 2006) Webstart: MeV is a Java application that allows users to analyze their microarray data. In year 5, we created an interface that allows VirtualPlant users to launch MeV Webstart from their experiment in their user cart. Once launched, the user can analyze data as they normally would using any of the built-in genomics analysis tools. Once the analysis is complete, the user can save clusters of genes, and upload them back into the VirtualPlant gene cart.

**Supplement Year 5: Motif Retrieval Analysis:** This tool examines nucleotide sequences in upstream promoter regions of annotated genes and identifies over-represented sequence motifs. The tool takes as input a list of known motifs or, if not provided, MEME (Bailey and Elkan, 1994) can obtain a list of predicted motifs. It also takes as input a genelist and a background genelist for statistics. The search can examine either 1kb or 3kb upstream in the promoter. If MEME is used, then the width and number of motifs may be set. The output data is presented as a table showing the position weight matrices (PWMs) calculated with Fisher’s exact test. There is also is an image representation of the PWM, on which the user can click to view the PWM as text and get more details about it. This tool is currently being tested prior to being released.

**Supplement Year 5: Motif Cart:** Users can save PWMs from various analyses into the motif cart. The PWM can then later be used to search sequences in other sequences of the same or other species. This feature is still under development.

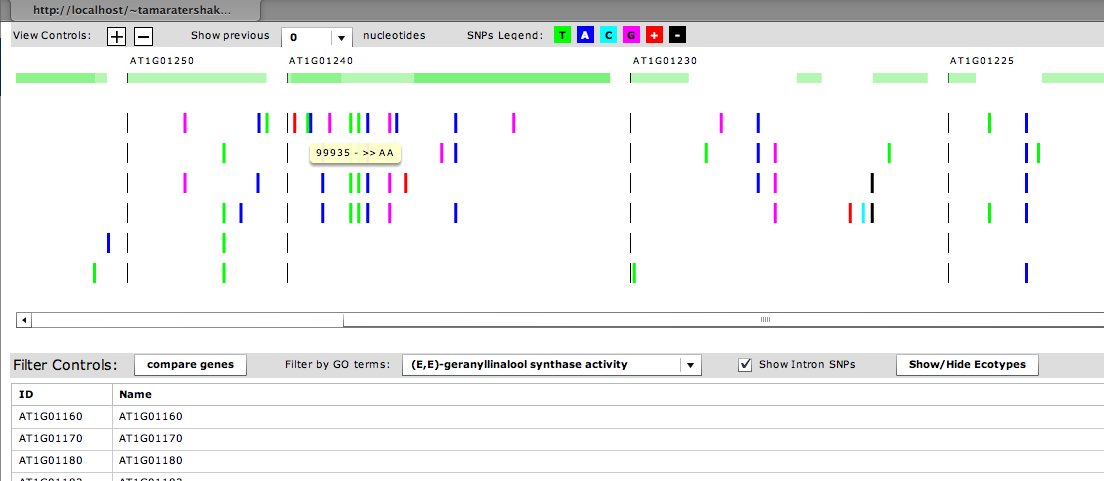
**Supplement Year 5: Genesect:** Genesect is a non-parametric randomization test that determines whether the overlap between two gene lists is higher or lower than expected by chance. Genesect randomly selects elements from the user-specified background population for each of two lists equal in size to the input lists. Then it counts the size of the two random lists’ intersection. Typically, Genesect intersects about 1000 pairs of random lists, which provides adequate statistics about the size of the intersection of the input lists.

Genesect outputs a p-value and a z-score – the number of standard deviations by which the size of the observed gene list overlap differs from the mean of the size of intersection of the random sets. Genesect can be performed on 2 or more gene sets. The output of the function provides a matrix with each cell containing the p-value or z-score for a pair of gene lists. Users can then click on the hyperlinks to add the desired gene lists to the cart**.**

**Supplement Year 6: Model Simplification Anova:** In experiments where there are two or more explanatory variables (factors), Analysis of Variance (ANOVA) is often used to determine whether there is interaction between the factors. We are designing a program that will identify the simplest model that explains the interaction. The input for our two-way ANOVA statistical program is gene expression data consisting of two factors: a dichotomous categorical input variable and a categorical variable consisting of three or more levels. For example the first factor may correspond to nitrogen treatment (high or low) and the second to genotype (three or more). The output for each gene gives the categories significant effects or the interaction between levels of either factor deemed to be significant. For each locus the program creates a model using groups of factors as the independent variables and gene expression as the dependent variable. It then simplifies the model (reducing the number of coefficients) as much as possible as its easier to pick up significant effects in simpler models with fewer parameters. The idea is to test for the alternative hypothesis that there is a difference among the groups. Since the program tests for interactions between variables it will allow us to see if the genotype factor has a dependency on the levels of the treatment factor. In other words, you will be able to see if there is an effect with nitrogen treatment but only in specific genotypes.

**Supplement Year 6: SNP/InDel Viewer:**

We have started development on a Flash-based tool to compare SNPs, insertions, and deletions (hereafter, SNPs for simplicity) across ecotypes. Instead of showing a complete linear portion of the chromosome as most gene browsers do, this tool shows only the genes selected by the user. The SNPs are shown below for all ecotypes in such a way that it is easy to see when the same SNP occurs across ecotypes, and to see concentrations of SNPs. The user can zoom in or out, choose whether to see intron SNPs, opt to see 500 or 1000 of the gene promoter and filter the gene list by GO term.

****

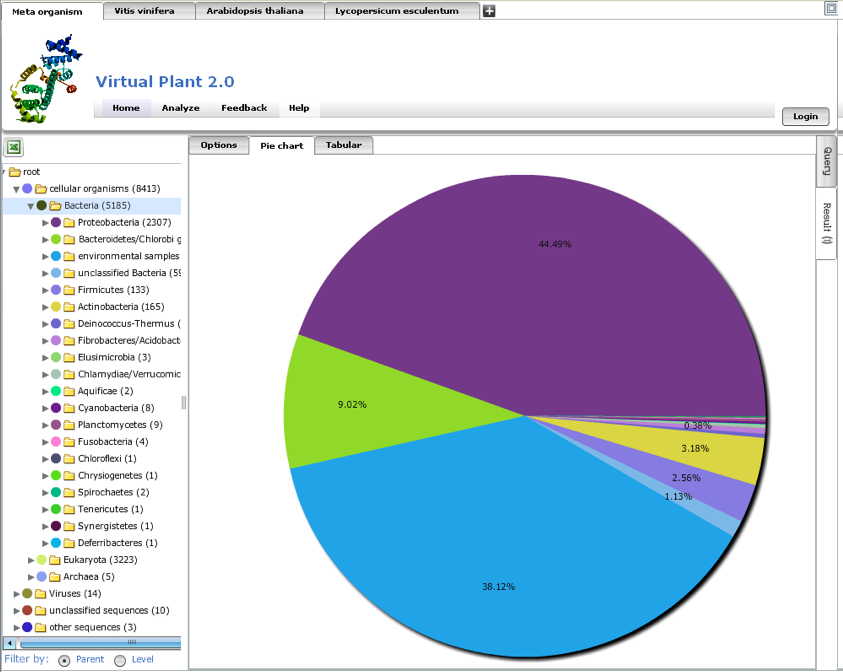
**Supplement Year 6: RNA-seq and CHIP-seq Analysis Pipelines:** We have developed pipelines for analyzing sequences generated by Illumina sequencing platform. It is well known that sequenced generated by Illumina tend to have lower quality on the 3’. To improve the quality of the matches we first trim from the 3’ of the sequences until a high quality base is reached. Despite efforts to minimize the number of ribosomal RNA (rRNA), we find majority of the reads map to the rRNA. It is thus essential to filter the sequences and remove all reads that 1) match rRNA, 2) match vector sequences, or 3) contain low complexity sequences. These searches are doneWe then take the reads and map them to the genome using the Burrows-Wheeler algorithm (BWA) (Li and Durbin 2009) or Bowtie and TopHat package (Trapnell 2009). The results are loaded into the GMOD database (Stein 2002) using scripts provided by the BioPerl (Stajich 2002) project. The reason for using GMOD is that it is paired with Gbrowse, which allows us to visualize the sequence alignments to the genome sequence. For RNA-seq we are currently using Cufflinks (Trapnell 2010) to determine which genes are differentically expressed, and for CHIP-seq we are using MACS (Zhang 2008) to identify peaks.

Nearly all the steps discussed above require knowledge in Bioinformatics in order to analyze the sequence data. We are in the process of integrating the steps described above into a workflow management software system, such as Galaxy (Blankenberg 2010), onto VirtualPlast so users can start analyzing next-generation data.

**Supplement Year 6: Metagenomics sequence analyzer:** This tool takes a fasta format file as input. The software groups identical sequences in the input file and then offers the user the following filter options:

* Filter out low complexity sequences.
* Filter out sequences outside an specified size range.
* Filter out sequences with less than an specified percentage of identity obtained against the best hit(s) in sequence alignments (see below).
* Filter out sequences with ambiguous base callings (e.g N,Y,R)

The sequences are compared against the non-redundant database from NCBI using BLASTN. The software takes the best hit(s) for each sequence and counts the taxon identifiers associated to the subject sequence. The count is weighted by the total number of hits to different species. The weighted counts for each taxon ID are represented in the NCBI phylogenetic tree and summarized in a pie chart (similar to the GO Pie currently implemented in VirtualPlant)



**Source code control system.** We are using the Concurrent Version System (CVS) to manage the source code for VirtualPlant. CVS allows us to manage changes to the code, prepare versions for release, review the history of changes, and determine who made those changes. Unlike some commercial source code control systems, CVS allows multiple developers to edit the same file at the same time. In addition to the CVS command-line utilities, we also use the ViewCVS software to provide a GUI for browsing the CVS repository through a web browser interface. We were experiencing some difficulty with CVS so by the end of year 3 we switched to Subversion (SVN) available at <http://subversion.tigris.org/>.

**Defect tracking system.** To allow the development team to keep track of problems reported by the beta testers and track features for future implementation, we are using the open source Mantis Bug Tracking System (http://mantisbt.sourceforge.net). The software allows users to generate reports of the problems they find while using the VirtualPlant system. Mantis generates emails which are sent to alert the development team of the nature and severity of the problem. The defect tracking system in conjunction with CVS allows us to rapidly respond to problems and deploy new releases of the VirtualPlant code that address the problems. Mantis also generates reports that allow us to identify areas of the code that are particularly problematic, facilitating targeted code refactoring and implementation of changes to the software architecture that would result in increased stability for specific features.

**User input.** Currently we are using two email groups (“virtualplant” and “virtualplant-discuss”) available from Google Groups (http://groups.google.com) for communication between software developers and Users. These email lists were made public upon release of the first version of the software and we expect them to be an important avenue for discussing technical as well as practical aspects of the use of the VirtualPlant software. In addition, VirtualPlant contains several links and forms that allow users to send questions, comments or suggestions to the software developers. In fact, the latter has been the preferred method for user communication up to date.

**References:**

## Ashburner, M., C. A. Ball, et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." Nature Genetics 25(1): 25-29.

Bailey, T. L. and Elkan, C (1994). "Fitting a mixture model by expectation maximization to discover motifs in biopolymers", Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28-36,

Blankenberg, D., G. Von Kuster, et al. (2010). "Galaxy: a web-based genome analysis tool for experimentalists." Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.] **Chapter 19**: Unit 19 10 11-21.

## Chiu, J. C., E. K. Lee, et al. (2006). "OrthologID: automation of genome-scale ortholog identification within a parsimony framework." Bioinformatics: btk040.

## Dudoit, S., M. J. van der Laan, et al. (2004). "Multiple Testing. Part I. Single-Step Procedures for Control of General Type I Error Rates." Statistical Applications in Genetics and Molecular Biology 13(1): Article 13.

## Irizarry, R. A., B. Hobbs, et al. (2003). "Exploration, normalization, and summaries of high density oligonucleotide array probe level data." Biostat 4(2): 249-264.

Katari, Manpreet S., Steve D. Nowicki, Felipe F. Aceituno, Damion Nero, Jonathan Kelfer, Lee Parnell Thompson, Juan M. Cabello, Rebecca S. Davidson, Arthur P. Goldberg, Dennis E. Shasha, Gloria M. Coruzzi and Rodrigo A. Gutiérrez, VirtualPlant: A Software Platform to Support Systems Biology Research, Plant Physiology 152:500-515 (2010).

Kent, W. J. (2002). "BLAT--the BLAST-like alignment tool." Genome Res **12**(4): 656-664.

Langmead, B., C. Trapnell, et al. (2009). "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." Genome Biol **10**(3): R25.

Li, H. and R. Durbin (2010). "Fast and accurate long-read alignment with Burrows-Wheeler transform." Bioinformatics **26**(5): 589-595.

Mortazavi, A., B. A. Williams, et al. (2008). "Mapping and quantifying mammalian transcriptomes by RNA-Seq." Nature methods **5**(7): 621-628.

## Philippi, S. (2004). "Light-weight integration of molecular biological databases." Bioinformatics 20(1): 51-57.

## Poultney, C. S., R. A. Gutierrez, et al. (2007). "Sungear: interactive visualization and functional analysis of genomic datasets." Bioinformatics 23(2): 259-261.

Saeed AI, Bhagabati NK, Braisted JC, Liang W, Sharov V, Howe EA, et al. [TM4 microarray software suite](http://www.ncbi.nlm.nih.gov/pubmed/16939790). . Methods in Enzymology. 2006;411:134-93.

## Shannon, P., A. Markiel, et al. (2003). "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks." Genome Res. 13(11): 2498-2504.

Stajich, J. E., D. Block, et al. (2002). "The Bioperl toolkit: Perl modules for the life sciences." Genome Res **12**(10): 1611-1618.

Stein, L. D., C. Mungall, et al. (2002). "The generic genome browser: a building block for a model organism system database." Genome Res **12**(10): 1599-1610.

Trapnell, C., L. Pachter, et al. (2009). "TopHat: discovering splice junctions with RNA-Seq." Bioinformatics **25**(9): 1105-1111.

Trapnell, C., B. A. Williams, et al. (2010). "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation." Nature biotechnology **28**(5): 511-515.

## Thum KE, Shin MJ, Gutierrez RA, Mukherjee I, Katari MS, Nero D, Shasha D, Coruzzi GM (2008) “An integrated genetic, genomic and systems approach defines gene networks regulated by the interaction of light and carbon signaling pathways in Arabidopsis.” BMC Syst Biol 2: 31

Wang, Donghan (Jarod), and Arthur Goldberg, Enhancing the Performance of a Parallel Correlation Computation, submitted to the IEEE 2010 Fourth International Workshop on Scientific Workflows.

Ware D, Jaiswal P, Ni J, Pan X, Chang K, Clark K, Teytelman L, Schmidt S, Zhao W, Cartinhour S, McCouch S, Stein L. (2002) “Gramene: a resource for comparative grass genomics.” Nucleic Acids Res. 2002 Jan 1;30(1):103-5.

Zhang, Y., T. Liu, et al. (2008). "Model-based analysis of ChIP-Seq (MACS)." Genome Biol **9**(9): R137.