BioVLAB-Microarray: Microarray Data Analysis in Virtual Environment

Youngik Yang, Jong Youl Choi, Kwangmin Choi
School of Informatics
Indiana University
Bloomington, IN, U.S.A.
{yiyang, jychoi, kwchoi}@indiana.edu

Marlon Pierce
Pervasive Technology Labs
Indiana University
Bloomington, IN, U.S.A.
mpierce@indiana.edu

Dennis Gannon
Microsoft Research
Redmond, WA, U.S.A.
Dennis.Gannon@microsoft.com

Sun Kim
Center for Genomics and Bioinformatics,
School of Informatics
Indiana University
Bloomington, IN, U.S.A.
sunkim2@indiana.edu

Abstract—**Microarray technology is a high-throughput experimental technique that can measure expression levels of hundreds of thousands of genes simultaneously. To interpret massive data from gene-expression microarray experiments, biologists encounter computational and analytical challenges. This is especially challenging for small research labs that lack local computing and bioinformatics expertise. Here, we introduce a virtual analysis system for microarray gene expression data in computing clouds with flexible and configurable GUI workflow engine so that biologists are able to analyze the data in many angles without worrying about computational and bioinformatics issues.**

Keywords-component; cloud computing; Microarray; workflow engine;

I. INTRODUCTION

Recently, biologists and medical scientists can measure expression of hundreds of thousands genes simultaneously using microarray technology. Analysis of data from high-throughput microarray technology remains a big challenge in the field of bioinformatics for its huge volume of data. In addition to its huge data volume, analysis of microarray data involves many computational tools for searching genes of interest, clustering, component analysis, network analysis, to find meaningful interpretations from different angles.

Due to its huge size of the data, analysis tasks are usually computationally-intensive and time-consuming and require high-performance computing power. Recently, virtual computing clouds are drawing attentions for its flexibility since users can create a high-performance cluster of any number of virtual computing units in an on-demand manner. However, building an efficient computation environment and integrating it with various bioinformatics applications is a challenging task to normal users even with a cloud computing cluster. This is a computing infrastructure issue.

Another issue in microarray data analysis is to execute multiple of analysis tasks as a single batch job. A typical batch job for analysis involves a series of execution of different analysis methods and it is often getting tedious due to its repetitive execution with slightly different parameters in each run. This is called an exploratory data analysis issue.

To allow small biology labs to utilize gene expression microarray data, both issues should be taken care of. We have developed a system for microarray data analysis in virtual environment, called BioVLAB Microarray, using a graphical workflow composer, XBaya [1], coupled with Amazon Elastic Computing Cloud (EC2), Simple Storage Service (S3), and Microsoft Application-Based Storage. Contribution of our system is three-fold: i) providing a suite of microarray analysis applications which can utilize remote high-performance computing resources such as computing clouds or public Web services, ii) providing an easy-to-use and reconfigurable workflow system in which a workflow composition requires no system knowledge of working environment and users can repeatedly execute the same workflow with different parameter settings, and iii) building a Web portal where an administrator can manage inventories of applications that a user can use for his/her workflow composition and also users can manage their data.

The rest of the paper is organized as follows. We discuss about the motivation of our system in Section II and related work in Section III. Section IV explains microarray experiments and gene expression network analysis tasks. In section V, we describe the BioVLAB Microarray system architecture. Section VI shows result from an experiment using a microarray data using our system. Section VII concludes the paper with discussion on future works.

II. MOTIVATION

Analysis of microarray data requires various tasks such as searching similar patterns of genes with a target gene, network analysis, extracting differentially expressed genes, clustering, and component analysis (See Fig. 1). Running

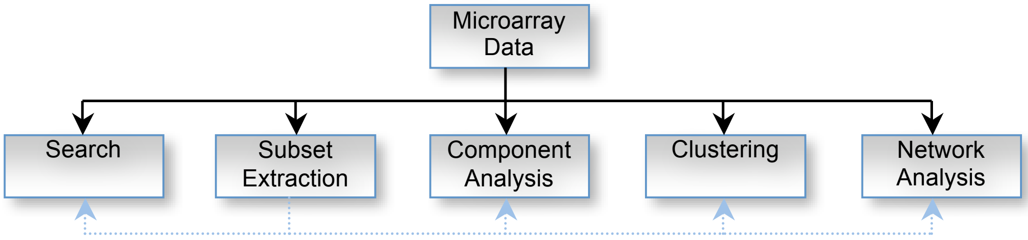


Figure 1. Common task of Microarray analysis: gene expression data obtained from a microarray experiment is analyzed in various ways such as finding a set of genes which show similar expression with a target gene, extracting differentially expressed genes, component analysis, clustering, and network analysis. Output of subset extraction which is a set of differentially expressed gene, can be input of other analyses such as component analysis, clustering, and network analysis.

application software to perform an analysis task is often not trivial for biologists, especially when it is written for Linux or Unix environment which is often the case in Bioinformatics. In addition, to combine multiple analysis tasks, an output of a task needs to be an input to another analysis, which is even more difficult for many biologists. Use of a workflow composer can alleviate this technical burden significantly since all application software tools are visible and multiple tools can be easily combined by the drag-and-drop feature of a workflow composer. A well defined and composed workflow can be repetitively used with different parameter settings and can also be modified easily. Coupling a workflow composer with a computing cloud will allow biologists in small research labs to analyze microarray data on their desktop or notebook without worrying about many computational and bioinformatics issues.

Motivated by this, we built a workflow execution environment of microarray gene expression data analysis, named BioVLAB Microarray. This system is a reusable, reconfigurable, high-level workflow execution workbench powered by computing clouds in which a user can do various microarray gene expression analyses without deep knowledge of computing environment and resources.

III. Related Work

A number of research projects have been conducted to build a workflow system in the field of biology and bioinformatics. Those efforts can be categorized into two main directions: one is to develop efficient and user-friendly workflow composers and execution engines and the other is to use distributed and heterogeneous computing resources, such as a Grid system, combined with a workflow system to enhance computing capability. There are many bioinformatics workflows and workflow engines. SIBIOS [2] has been developed to address an issue of dynamic workflow execution and interoperability between distributed and heterogeneous bioinformatics services. BioWBI [3] is a Web tool to provide researchers with a virtual workspace for sharing data with collaborators and a graphical workflow composer. KDE Bioscience [4] is a Java-based platform that integrates more than 60 bioinformatics tools and provides a GUI-based workflow composer and its execution engine.

The other research direction to build a workflow system is to use a group of distributed and heterogeneous computing resources, called Grid system. Among many efforts, Taverna [5], Triana [6], Kepler [7], GNARE [8], and RENCI-Bioportal [9] are worth mentioning. Taverna [5] has been developed as a part of the myGrid project and it is a workflow system running bioinformatics Web services and existing bioinformatics applications over distributed resources. Triana [6] is designed to offer more a general approach to integrate with other Grid based systems, such as Globus and GridLab, or service oriented system like Web services. Kepler [7] is a scientific workflow system based on a dataflow-oriented model, so called an actor-oriented model. GNARE [8] is a bioinformatics server equipped with automated workflows and a Grid-based computational backend to perform high throughput analysis of genomes with an aid of the workflow engine called GADU which can access to Grid resources.

Our BioVLAB Microarray system is similar to other workflow systems such as Taverna [5], Triana [6], Kepler [7], and GNARE [8], in terms of utilizing the workflow concept. However, our BioVLAB Microarray system is different from those systems in that we enable a user to use on-demand virtual computing powers supplied by a computing cloud service and various public resources. Using virtual instances, instead of using persistent resources, is more flexible and cost-efficient way for a small group of users who needs, only sporadically, high performance computing infrastructures for running computation-intensive bioinformatics applications. Contrast to our previous work [10], the BioVLAB Microarray system is integrated with bioinformatics applications for microarray analysis and more public resources such as Microsoft Application-Based Storage.

IV. Gene Expression And Expression Netowrk

At a given time, only a subset of genes in a cell is expressed corresponding to environmental changes and its own changing needs. To study the dynamics of cell activity, experiments encompassing a whole set of genes is required. The microarray technology is a recently developed high-throughput experimental method for the purpose. A single microarray experiment can measure expression levels of hundreds of thousands of genes within a cell. This high-throughput experiment gives an insight of how a cell responds to changed conditions [11].

Microarray is useful in many ways. First, previously unknown function(s) of a gene can be characterized by finding similar expression patterns of genes with known function. Second, co-expression of many genes can used to build a gene transcription network, which can reveal how genes interact with each other. Third, identifying genes related to certain disease and detecting effectiveness of new drugs can be also tested in terms of clinical data.

BioVLAB Microarray can be used to perform integrated analysis of microarray data for two research goals: gene expression analysis and expression network analysis. In section 4.1, searching similar pattern of genes against a target gene, subset extraction of expression set, unsupervised approach of gene expression analysis, and component analysis are discussed. Interaction of gene products is another important issue in microarray data analysis. In section 4.2, we discuss about constructing gene expression networks from a microarray data.

A. Gene expression analysis

1) Searching for similar expression patterns: We often need to find a set of genes that have similar expression patterns to a gene of interest. When the function of a target gene is unknown, inspecting function of genes with expression patterns similar to the target gene may reveal the function of the target gene. Even when the function of the target gene is known, we can have an insight how similarly/dissimilarly genes are expressed in given conditions. Note that this type of analysis is not precisely defined, but rather exploratory, thus repeated analysis from different angles is necessary.

2) Subset extraction: One important research question is to identify genes that are differentially expressed. The criterion for differentially expressed genes is often subjective. Thus this is typically an exploratory analysis task and often combined with statistical evaluation. Since many genes are involved, there is higher chance of observing difference in gene expression randomly and a rigorous statistical testing for dealing with multiple experiments is needed. A widely used method for adjusting multiple testing correction False Discovery Rate (FDR) [12].

3) Clustering: Many genes in a microarray experiment may have no known protein functions previously. Also, genes with similar expression patterns may need to be studied as a group. Clustering has been widely used in analysis of microarray data. Traditional unsupervised learning methods, such as k-means clustering [13] and hierarchical clustering, have been widely used. Quality threshold clustering (QT clustering) [14] determines clusters based on maximum diameter instead of a number of clusters in k-means clustering. Self-organizing map (SOM) [15] has widely used. Recently, biclustering methods [16] are getting attention. Clustering methods such as k-means clustering, hierarchical clustering, and SOM considers only one direction of data either rows (genes) or columns (samples/conditions) but not both. On the other hand, biclustering methods are two-way clustering methods where they train both directions and reach to clusters in both ways.

4) Component analysis: Hidden structure of microarray gene expression data may be revealed by component analysis. Principal component analysis (PCA) and independent component analysis (ICA) [17] are two widely used component analysis.

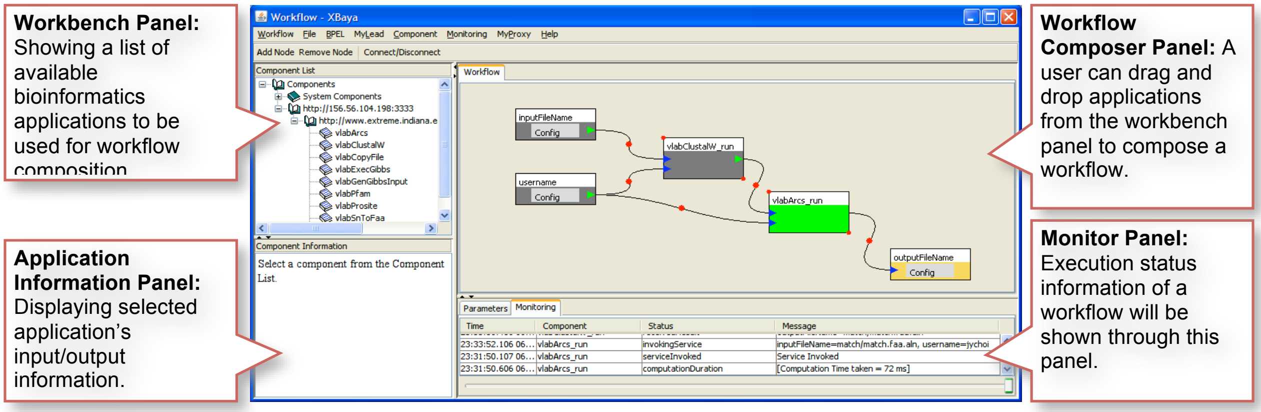
B. Expressed protein network analysis

1) Network analysis: The network analysis approach is a powerful analysis tool to understand real-world phenomena. From the viewpoint of network theory, a world is a network of inter-connected node. Here, nodes represent the individual actors and connections represent the relationships between actors. Relationship among nodes within the network often carries more robust information than each node. One of the main interests in network analysis is exposing hidden structures from a given network. Network structures can be revealed by identifying several network properties using graph theory: betweenness, closeness, centrality, clustering coefficient, cohesion, density, path length, radiality, and structural holes. Three models of random network, scale-free network, and hierarchical network are used to interpret a given network.

2) PPI network analysis: Network analysis can be applied to several biological systems, especially to protein-protein interaction (PPI) network. PPI networks are central to most biological processes, including metabolic pathway, cellular signaling, and gene expression control. The systematic identification of PPIs and their network plays a key role in understanding how a cellular system works. Protein–protein interactions are known to be a scale-free network [18]. Many biological networks also show outstanding modularity in structure and it is believed that these structural modules reflect higher-level functional organization of cellular components. Since PPI patterns correlate with the protein's functional properties, PPI maps can be utilized to uncover potential biological role of proteins with unknown functional classification. Also, by adding expression correlation score to network edges, we may see inter-protein crosstalk during gene expression control. Transcripts of protein-coding genes are matched to PPI.

V. BioVLAB Microarray System

Microarray data analysis tasks are typically exploratory as we discussed in the previous section. Thus no system – no matter how well designed it is – can meet users' diverse requirement. BioVLAB Microarray workflow system uses a graphical workflow composer (See Fig. 2) to select, combine, and execute various analysis tasks, invoking application software packages on a computing cloud, Amazon Elastic Computing Cloud (EC2) and Simple Storage Service (S3) or Microsoft Application-Based Storage. Use of a cloud computer allows users to build instant and flexible parallelizable clusters. With those resources, users can save time and efforts by utilizing more powerful external computing resources and reducing the burden of power-constraint local systems.



Components
System Components
http://156.56.104.198:3333
http://www.extreme.indiana.e
vlabArcs
vlabClustalW
vlabCopyFile
vlabExecGibbs
vlabGenGibbsInput
vlabPfam
vlabProsite
vlabSnToFaa

Select a component from the Component List.

Time	Component	Status	Message
23:33:52.106 06...	vlabArcs_run	invokingService	inputFileName=match/match.faa.aln, username=jychoi
23:31:50.107 06...	vlabArcs_run	serviceInvoked	Service Invoked
23:31:50.606 06...	vlabArcs_run	computationDuration	[Computation Time taken = 72 ms]

Workbench Panel: Showing a list of available bioinformatics applications to be used for workflow composition

Application Information Panel: Displaying selected application's input/output information.

Workflow Composer Panel: A user can drag and drop applications from the workbench panel to compose a workflow.

Monitor Panel: Execution status information of a workflow will be shown through this panel.

Figure 2. Composing a workflow by using XBaya: a user can (a) add a node by doing drag-and-drop one of available applications from the workbench panel and (b) create an edge by pushing and releasing a mouse pointer between two nodes of a workflow.

The purpose of our system is three-fold: i) providing user-friendly, graphic-user-interface (GUI) workflow composer and execution engine, ii) enabling users to use remote computing powers, such as computing clouds, e.g., Amazon EC2/S3, and public Web services, such as NCBI services and Microsoft Application-Based Storage, with ease and flexibility, and iii) providing an easy-to-use Web portal in which users can manage the system and data. In the following, we discuss more details of our system.

A. BIOVLAB Microarray system architecture

Our system consists of the following three main components: workflow composer and execution engine, application services which can utilize Amazon EC2/S3 and Microsoft Application-Based Storage service, and Web portal.

1) Workflow composer and execution engine: The workflow concept has been introduced in the scientific communities to execute a batch of multiple tasks by reducing a user's involvement and enables a user to repeat the same task easily. A workflow is a directed acyclic graph where each inner node is an application to execute and an edge between two nodes represents a flow of data. A starting node and an end node in a workflow graph correspond to an input and an output data respectively. For the workflow composition and execution, our system uses a graphical workflow composer, called XBaya [1].

By using XBaya, a user can easily compose a workflow by doing drag-and-drop from the workbench that lists software applications available on the system and execute the workflow instantly. During the execution of a workflow, XBaya can monitor status of workflow execution. Detailed status-messages will be displayed in the monitor panel as shown in Fig. 2. For more information about using XBaya, refer to [1].

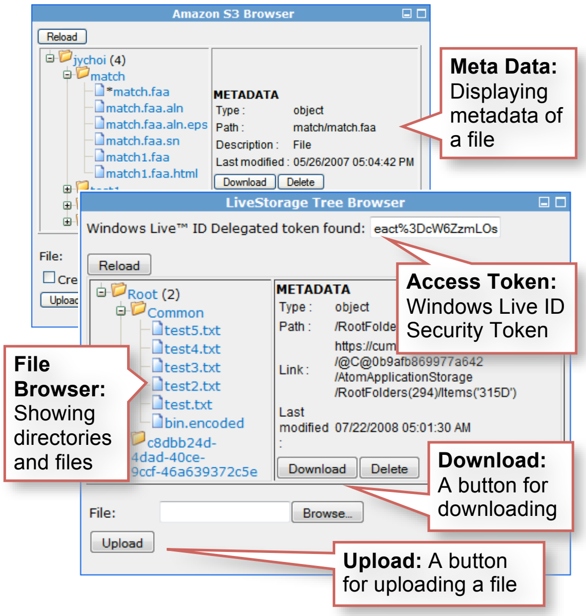
2) Application service: Many biological applications are developed stand-alone and platform-dependent. Thus interoperability among applications is a major problem and a user has a great difficulty in executing multiple applications in a single environment. To overcome this problem, application services have been proposed. An application service is a Web service that can invoke applications through simple Web messages known as Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). In our system, we used the Generic Service Toolkit, known as Gfac [19], to convert any command-line bioinformatics application into a Web service, which is accessible by XBaya.

Some bioinformatics applications require a powerful high performance or parallelizable computation environment. Our system is designed to utilize remote high-performance resources, such as computing cloud of Amazon EC2 in which a user can create any number of virtual computing instances running in parallel. Our system can also use public remote storage services, such as Amazon S3 and Microsoft Application-Based Storage, as a storage service to store intermediate or final output of workflow execution. Thus, a user in our system can choose to use both of them without worrying about installation or management of computing resources. For easy-to-use access and management of data stored in the remote services, we provide a Web portal in our system.

3) Web portal: The management of our system, as an administrator, or the access of stored data, as a user, can be done though the portal interface in our system, called Web portal. We built our Web portal by using the Open Grid Computing Environments (OGCE) Portal [20].

As an administrator, management of registered application that users are allowed to execute can be done easily through the Gfac's registry portlet that is used in our system. User management and access control can also be done through the portlet interface provided by OGCE.

In our system, a user can upload and download inputs and outputs of bioinformatics applications, which can be stored in the remote storage services, such as Amazon EC2 and Microsoft Application-Based Storage, through our Web portal simply by using a Web browser (Fig. 3).



jychoi (4)
match
*match.faa
match.faa.aln
match.faa.aln.eps
match.faa.sn
match1.faa
match1.faa.html

METADATA
Type : object
Path : match/match.faa
Description : File
Last modified : 05/26/2007 05:04:42 PM

Windows Live™ ID Delegated token found: eact%3DcW6ZzmLOs

Root (2)
Common
test5.txt
test4.txt
test3.txt
test2.txt
test.txt
bin.encoded
c8dbb24d-4dad-40ce-9ccf-46a639372c5e

METADATA
Type : object
Path : /RootFolde
Link : https://cum /@C@0b9afb869977a642 /AtomApplicationStorage /RootFolders(294)/Items('315D')
Last modified : 07/22/2008 05:01:30 AM

File:

Meta Data: Displaying metadata of a file

Access Token: Windows Live ID Security Token

File Browser: Showing directories and files

Download: A button for downloading

Upload: A button for uploading a file

Figure 3. A Web interface for browsing files in Amazon S3 and Microsoft Application-Based Storage: a user can simply use a Web browser to explore files stored in those services.

B. Analysis resources

R [21] is a widely used language for microarray data analysis. Bioconductor [22] that is implemented in R was integrated to analyze gene expression data. Many built-in functions in R as well as other R packages have been integrated into our system. Public databases such as NCBI GEO and DIP were directly/indirectly integrated to our system.

1) Data acquisition: Microarray data can be obtained from public data repositories. Here, we connect to the NCBI GEO database to get microarray data using GEOquery package [23].

2) Gene expression search: Genes having similar expression pattern with a target gene are often interested. We used similarity metric such as correlation.

3) Differentially expressed genes: To extract differentially expressed gene, we used the limma package [24].

4) Clustering: Built-in R functions `kmeans` and `hclust` were used for k-means clustering and hierarchical clustering, respectively. Additional clustering packages were used such as flexclust package [25] for QT clustering, biclust [26] for biclustering, and self-organizing map [27] for SOM.

5) Component analysis: Two popular analysis methods, principle component analysis (PCA) and independent component analysis (ICA), were incorporated. PCA were executed by calling R built-in function `princomp`. For ICA, we used fastICA package [28].

6) Network resources: Information of protein-protein interaction of a genome was retrieved from the Database of Interacting Proteins (DIP) database [29]. DIP catalogs experimentally determined protein–protein interactions that were manually reviewed by experts or were collected automatically using computational prediction methods. Connected components in a generated network were computed using Perl graph package. To visualize a network, we used GraphViz package.

VI. EXPERIMENTS

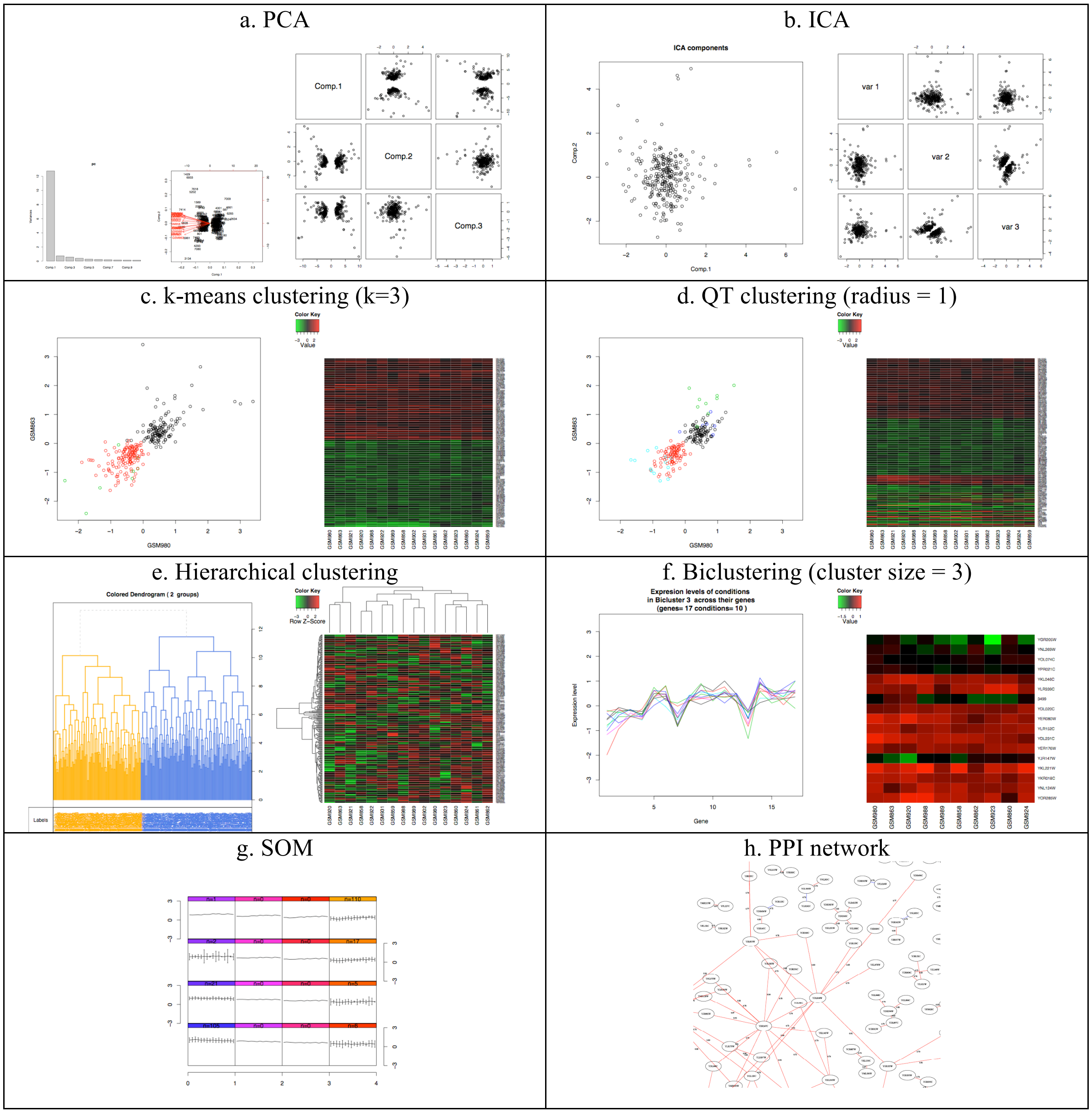
GDS38 gene expression data set [30] was chosen for an input gene expression data. It was remotely retrieved from the NCBI GEO database by an application in XBaya and used throughout experiments. It is time-series gene expression data to observe cell cycle in *Saccharomyces cerevisiae* yeast genome. There are 7680 spots in each 16 samples where each sample was taken every 7 minutes as cells went through cell cycle.

Fig. 4 shows a graphical experiment summary of component analysis, clustering, and network analysis. For component analysis and clustering, only subset of genes that are differentially expressed was used. Differentially expressed genes were extracted using the limma package with an adjusted P-value less than 0.05.

Outputs of two component analyses, PCA and ICA, were summarized graphically in Fig. 4.a and 4.b. For PCA in Fig. 4.a, screeplot, biplot, and pair plot with top 3 dimensions were shown. Fig. 4.b shows the result of ICA as a scatter plot and a pairwise scatterplot with top 3 dimensions.

Clustering analysis results are shown in Fig. 4.c through Fig. 4.g. For *k*-means clustering in Fig. 4.c, a scatter plot and a heat map visualize the clustering result. A cluster size *k* was set to 3. QT clustering result is shown in Fig. 4.d with a scatter plot and a heat map. The radius threshold of clustering quality was set to 1. Fig. 4.e shows two hierarchical plots, one-way hierarchical clustering and two-way hierarchical clustering shown as a heat map. The bi-clustering summary is shown in Fig. 4.f. Cheng and Church method [31] was used for bi-clustering in which a cluster size was set to 3. Graphical summary of a self-organizing map was shown in Fig. 4.g. For SOM, *x* dimension and *y* dimension were set to 4.

Fig. 4.f shows the graphical summary of a PPI network. Given positive/negative correlation cutoff values, a gene network was generated from a correlation matrix by referring to DIP. Since our approach combined gene expression correlation between genes and their protein-protein interaction information, the resulting network representation visualized sub-networks of genes that might participate in positive/negative feedback control of interesting cellular processes. These networks were represented as undirected graph. Sub-networks were identified by detecting connected component and visualized using the GraphViz package. Nodes represent genes (or their protein products). Red and blue colors were used for edges to represent positive correlation and blue for negative correlation respectively. Each correlation score was labeled over edge line.



a. PCA

b. ICA

c. k-means clustering (k=3)

d. QT clustering (radius = 1)

e. Hierarchical clustering

f. Biclustering (cluster size = 3)

g. SOM

h. PPI network

Figure 4. Graphical experiment summary: see detail in text

VII. Conclusion

In this paper, we introduced a system for microarray gene expression data analysis in a virtual environment, BioVLAB Microarray. Coupling computing clouds and a user-friendly GUI workflow execution engine, we constructed a workflow system that can be used by biologists without worrying about computer infrastructure and bioinformatics issues. Our approach is an effective system design principle that allows biologists, especially those in small research labs, to utilize massive high throughput biological data. We believe that the approach introduced in this paper can be applied to many biological applications that need to deal with massive data.

The system introduced in this paper is useful but needs further development. First of all, there are a lot more software packages used for gene expression microarray data and there will be new application packages that need to be included in BioVLAB Microarray. In addition, biology and medical sciences are under-going a revolution to study biological system-level study, *a.k.a* systems biology, by utilizing data from high throughput instruments such as various types of microarray data (including gene expression data), sequence data from massively parallel sequencing methods, mass spectronomy, etc. Thus what biologists need is a system of great flexibility to integrate heterogeneous computing resources such as clouds, grid, and even multi-core desktop, altogether. Also, since there have been many public databases and applications available, the system would be able to integrate those valuable resources such as NCBI, KEGG, PDB, netblast, etc. A system architecture to deal with this situation is a new research problem that needs to be further studied.

We have begun to explore a 5-tier approach: a gateway, clouds, grid, desktop, and public resources. Since collaboration is also crucial issue in biological experiments, our system is being developed towards a collaborative workbench backed up by metadata catalog and search.

Acknowledgment

The work is partially supported by NSF MCB 0731950 and a MetaCyt Microbial Systems Biology grant from Lilly Foundations.

References

[1] S. Shirasuna and D. Gannon, "XBaya: A graphical workflow composer for the web services architecture," Indiana University 2006.

[2] M. Mahoui, L. Lu, N. Gao, N. Li, J. Chen, O. Bukhres, and Z.B. Miled, "A Dynamic Workflow Approach for the Integration of Bioinformatics Services," *Cluster Computing,* vol. 8, Oct. 2005, pp. 279-291.

[3] P. Leo, C. Marinelli, G. Pappada, G. Scioscia, L. Zanchetta, and Padova, "BioWBI: an Integrated Tool for building and executing Bioinformatic Analysis Workflows," in *Bioinformatics Italian Society Meeting (BITS 2004)*, 2004.

[4] Q. Lu, P. Hao, V. Curcin, W. He, Y. Li, Q. Luo, Y. Guo, and Y. Li, "KDE bioscience: platform for bioinformatics analysis", *J Biomed Inform*, vol. 39(4), Aug 2006, pp. 440-50, doi:10.1016/j.jbi.2005.09.001.

[5] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and enactment of bioinfor-matics workflows," *Bioinformatics,* vol. 20, pp. 3045-3054, Nov. 2004.

[6] I. Taylor, M. Shields, I. Wang, and A. Harrison, "Visual Grid Workflow in Triana," *Journal of Grid Computing,* vol. 3, pp. 153-169, September 2005.

[7] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E.A. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the Kepler system: Research Articles," *Concurr. Comput. : Pract. Exper. ,* vol. 18, pp. 1039-1065, Aug. 2006.

[8] D. Sulakhe, M. D'Souza, M. Syed, A. Rodriguez, Y. Zhang, E. Glass, M. Romine, and N. Maltsev, "GNARE – A Grid-based Server for the Analysis of User Submitted Genomes," *Accepted for publication in Nucleic Acids Res. (special issue),* vol. NAR-00335-Web-B-2007.R1, 2007.

[9] L. Ramakrishnan, M. Reed, J. Tilson, and D. Reed, "Grid Portals for Bioinformatics," in *Second International Workshop on Grid Computing Environments (GCE),* Workshop at SC'06, 2006.

[10] J.Y. Choi, Y. Yang, S. Kim, and D. Gannon, "V-Lab-Protein: Virtual Collaborative Lab for Protein Sequence Analysis," in *IEEE Workshop on High-Throughput Data Analysis for Proteomics and Genomics, Workshop at BIBM 2007* Sillicon Valley, 2007.

[11] "Microarrays: Chipping away at the mysteries of science and medicine," in *A Science Primer*, 2008: NCBI. http://www.ncib.nlm.nih.gov/About/primer/microarrays.html

[12] W.J. Ewens and G.R. Grant, *Statistical Methods in Bioinformatis*, 2nd Editioin ed.: Springer, 2005.

[13] J.A. Hartigan and M.A. Wong, "A K-means Clustering Algorithm," *Applied Statistics,* vol. 28, pp. 100-108, 1979.

[14] L.J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," *Genome Research,* vol. 9, pp. 1106-1115, 1999.

[15] T. Kohonen, *Self-Organizing Maps*: Springer, 2000.

[16] S.C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 1, pp. 24-25, 2004.

[17] P. Comon, "Independent Component Analysis, a new concept?," in *Signal Processing*. vol. 36: Elsevier, 1994, pp. 287-314.

[18] A.L. Barabasi and Z.N. Oltvai, "Network biology: understanding the cell's functional organization," *Nat Rev Genet,* vol. 5, pp. 101-113, Feb. 2004 2004.

[19] G. Kandaswamy, L. Fang, Y. Huang, S. Shirasuna, S. Marru, and D. Gannon, "Building Web Services for Scientific Grid Applications," *IBM Journal of Research and Development,* vol. 50, pp. 249-260, 2006.

[20] J. Alameda, M. Christie, G. Fox, J. Futrelle, D. Gannon, M. Hategan, G. Kandaswamy, G. v. Laszewski, M.A. Nacar, M. Pierce, E. Roberts, C. Severance, and M. Thomas, "The Open Grid Computing Environments collaboration: portlets and services for science gateways," *Concurrency and Computation Practice and Experience* vol. 19, p. 921, 2007.

[21] R. Ihaka and R. Gentleman, "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics,* vol. 5, pp. 299-314, 1996.

[22] R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Yang, and J. Zhang, "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology,* vol. 5, p. R80, 2004.

[23] S. Davis and P.S. Meltzer, "GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor," *Bioinformatics,* vol. 23, pp. 1846-1847, July 15, 2007 2007.

[24] G.K. Smyth, "Linear models and empirical Bayes methods for assessing differential expression in microarray experiments," *Statistical Applications in Genetics and Molecular Biology,* vol. 3, 2004.

[25] F. Leisch, "flexclust: Flexible Cluster Algorithms," R package, 2005.

[26] S. Kaiser and F. Leisch, "A Toolbox for Bicluster Analysis in R," University of Munich Department of Statistics: Technical Reports, No.28, 2008.

[27] J. Yan, "som: Self-Organizing Map," R package, 2004.

[28] J.L. Marchini, C. Heaton, and B.D. Ripley, "fastICA: FastICA Algorithms to perform ICA and Projection Pursuit," R package, 2007.

[29] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, and D. Eisenberg, "The Database of Interacting Proteins: 2004 update," *Nucl. Acids Res.,* vol. 32, pp. D449-451, January 1, 2004 2004.

[30] "GDS38: Cell cycle, alpha-factor block-release time course." vol. 2008: NCBI GEO. http://www.ncbi.nlm.nih.gov/geo/gds/gds_browse.cgi?gds=38

[31] Y. Cheng and G. M. Church, "Biclustering of Expression Data," in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 2000, pp. 93-103.