

# GEWORKBENCH 1.0

## USER MANUAL

VERSION 1.5

(FEBRUARY 2008)



**caBIG**<sup>™</sup> *cancer Biomedical  
Informatics Grid*<sup>™</sup>

an initiative of the National Cancer Institute



**Columbia University**  
**Joint Centers for Systems Biology**  
**Center for Computational Biology and Bioinformatics**  
**Herbert Irving Cancer Center**

**National Centers for Biomedical Computing - MAGNet**  
**National Cancer Institute Center for Bioinformatics**

**caBIG<sup>™</sup>**

**AMDeC - Academic Medicine Development Company**

Note: Previous versions of this program appeared under the names BioWorks and caWorkbench

Manual Revision History:

Version 1.0 October 14<sup>th</sup>, 2004

Version 1.1 June 21<sup>st</sup>, 2005

Version 1.2 August 1<sup>st</sup>, 2006

Version 1.3 September 14, 2006

Version 1.4 January 9, 2008

Version 1.5 February 28, 2008

# Copyright and License page

## SOFTWARE LICENSE AGREEMENT

Copyright ©2004-2008 Columbia University.

This software was developed by Columbia University in conjunction with First Genetic Trust and the National Cancer Institute, and so to the extent government employees are co-authors, any rights in such works shall be subject to Title 17 of the United States Code, section 105.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

2. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Columbia University, First Genetic Trust and the National Cancer Institute."

If no such end-user documentation is to be included, this acknowledgment shall appear in the software itself, wherever such third-party acknowledgments normally appear.

3. This license does not authorize the incorporation of this software into any proprietary programs.

4. THIS SOFTWARE IS PROVIDED "AS IS," AND ANY EXPRESSED OR IMPLIED WARRANTIES, (INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) ARE DISCLAIMED. IN NO EVENT SHALL THE COLUMBIA UNIVERSITY OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5. Below is the list of all third party software used in geWorkbench and their license information.

This product includes software developed by the Apache Software Foundation. Batik, Xerces, and Xalan are part of Apache XML project. Byte Code Engineering Library, POI, Jakarta Commons are part of Jakarta project, Axis is part of Apache Web Services project. Log4J is part of Apache Logging Services project. ObjectRelationalBridge is part of the Apache DB project. All aforementioned Apache projects are trademarks of The Apache Software Foundation. For further open source licensing issues pertaining to Apache Software Foundation, visit:

<http://www.apache.org/LICENSE>

This product includes software developed by NCI Center for Bioinformatics (NCICB). caBIO is part of caCORE project. caArray is cancer array informatics project. For more information, visit: [http://ncicb.nci.nih.gov/core/caBIO/technical\\_resources/core\\_jar/license](http://ncicb.nci.nih.gov/core/caBIO/technical_resources/core_jar/license)  
<http://ncicb.nci.nih.gov/download/caarraylicense.jsp>

This product may include the following software:

Cytoscape by the Institute for Systems Biology, University of California at San Diego, Memorial Sloan-Kettering Cancer Center and Institut Pasteur.

NetX by J. Maxwell, ODE For Java by Tim Schmidt.

OpenJGraph by Jesus M. Salvo, Jr.

Java Excel API by Andy Khan.

JMOL by molvisions.com

BioJava by BioJava.org.

JSCi by Mark Hale.

Ensemble for Java by the Sanger Institute and the European Bioinformatics Institute.

JGraph by JGraph Ltd.

Those software are licensed under the Lesser General Public License. For more information,

visit:

<http://www.gnu.org/copyleft/lesser.html>

This product may include the following software:

Bayesian Network tools in Java by Kansas State University.

Java Hidden Markov Models (JAHMM) by Jean-Marc François.

JFreeChart by David Gilbert.

the Ostermiller utils by Stephen Ostermiller.

Weak by the University of Waikato

Those software are licensing under General Public License. For more information, visit:

<http://www.gnu.org/copyleft/gpl.html>

This product may include the following software:

ArrayExpress by the European Bioinformatics Institute.

Ogsa from Globus Alliance.

JDOM by Jason Hunter and Brett McLaughlin.

Looks by Karsten Lentzsch.

PureTLS by Eric Rescorla.

SkinLF by Frédéric Lavigne.

Jaxen by The Werken Company.

Dom4J by MetaStuff, Ltd.

Piccolo by the University of Maryland.

Those software are licensing under BSD or BSD style License. For more information, visit:

<http://www.gnu.org/philosophy/license-list.html#OriginalBSD>

This product may include following public domain softwares:

AntLR by Terence Parr.

Distributions by the University of Edinburgh

Java Matrix Package by MathWorks and NIST.

SplashBitmap by Kai Blankenhorn

This product may include the following software:

AspectJ by the Eclipse Foundation.

JUnit by Erich Gamma and Kent Beck.

AntLR by Terence Parr.

Distributions by the University of Edinburgh

Java Matrix Package by MathWorks and NIST.

WSDL4j by IBM, Inc.

Those software are licensing under Common Public License. For more information, visit:

<http://www.eclipse.org/legal/cpl-v10.html>

This product may include the following software:

Eleritec Docking Framework by Marius. This software is under MIT license. For more information,

visit: <http://www.eleritec.net/>

This product may include the following software:

NetComponents by Original Reusable Objects, which is under it own license. For more information,

visit: <http://www.savarese.org/oro/downloads/NetComponentsLicense.html>

All other product names mentioned herein and throughout the entire project are trademarks of their respective owners.

<b><i>Members of the Development Team<sup>1</sup></i></b>		
<b><i>Development</i></b>	<b><i>User's Guide</i></b>	<b><i>Program Management</i></b>
<i>Names of developers</i>	<i>Names of technical writers and reviewers</i>	<i>Names of program managers</i>
Andrea Califano	Kenneth Smith	Aris Floratos
Aris Floratos	Eileen Daly	Kenneth Smith
Matt Hall		
Michael Honig		
Christine Hung		
Bernd Jagla		
Kiran Keshav		
Kauschal Kumar		
Manjunath Kustagi		
John Watkinson		
Xiaoqing Zhang		
<sup>1</sup> All contributors are currently or former members of the Joint Centers for Systems Biology, Columbia University, New York, NY.		



<b>Contacts and Support</b>	
Training contact	N/A
Support contact	<a href="http://gforge.nci.nih.gov/forum/?group_id=78">http://gforge.nci.nih.gov/forum/?group_id=78</a>

<b>LISTSERV Facilities Pertinent to software teams</b>		
<b>LISTSERV</b>	<b>URL</b>	<b>Name</b>
geWorkbench	<a href="http://gforge.nci.nih.gov/forum/?group_id=78">http://gforge.nci.nih.gov/forum/?group_id=78</a>	geWorkbench Open Discussion Forum
caBIO_Users	<a href="https://list.nih.gov/archives/cabio_users.html">https://list.nih.gov/archives/cabio_users.html</a>	caBIO Users Discussion Forum
caBIO_Developers	<a href="https://list.nih.gov/archives/cabio_developers.html">https://list.nih.gov/archives/cabio_developers.html</a>	caBIO Developers Discussion Forum
caDSR_Users	<a href="https://list.nih.gov/archives/cadsr_users.html">https://list.nih.gov/archives/cadsr_users.html</a>	Cancer Data Standards Repository
NCIEVS-L Listserv	<a href="https://list.nih.gov/archives/ncievs-l.html">https://list.nih.gov/archives/ncievs-l.html</a>	NCI Vocabulary Services Information
CAARRAY_DEVELOPERS-L	<a href="https://list.nih.gov/archives/caarray_developers-l.html">https://list.nih.gov/archives/caarray_developers-l.html</a>	caARRAY Developers Forum
CAARRAY_MAGE-OM_API	<a href="https://list.nih.gov/archives/caarray_mage-om_api.html">https://list.nih.gov/archives/caarray_mage-om_api.html</a>	caArray MAGE-OM API Forum
CAGRID_DEVELOPERS	<a href="https://list.nih.gov/archives/cagrid_developers.html">https://list.nih.gov/archives/cagrid_developers.html</a>	caGRID Developers Forum
CAGRID_USERS-L	<a href="https://list.nih.gov/archives/cagrid_users-l.html">https://list.nih.gov/archives/cagrid_users-l.html</a>	caGRID Users Forum



# Table of Contents

<b>TABLE OF CONTENTS .....</b>	<b>I</b>
<b>1 INTRODUCTION TO THE MANUAL .....</b>	<b>1</b>
1.1 CHANGES SINCE THE PREVIOUS MANUAL, VERSION 1.4 .....	1
1.2 CHANGES SINCE THE PREVIOUS MANUAL, VERSION 1.3 .....	1
1.3 ORGANIZATION OF THE GUIDE.....	2
1.4 GETTING STARTED WITH GEWORKBENCH.....	2
1.5 DOCUMENT TEXT CONVENTIONS .....	3
<b>2 OVERVIEW OF THE SOFTWARE.....</b>	<b>5</b>
2.1 MOTIVATION .....	5
2.2 INTRODUCTION TO GEWORKBENCH.....	5
2.3 THE GEWORKBENCH APPROACH TO INTEGRATED GENOMICS .....	6
2.4 COMPONENTS OF GEWORKBENCH .....	7
2.5 SUPPORTED DATATYPES.....	7
<b>3 VISUAL INTERFACE AND DATA MANAGEMENT.....</b>	<b>9</b>
3.1 LAYOUT OF THE GEWORKBENCH 1.0 INTERFACE .....	9
<i>Menu Bar</i> .....	10
<i>Data management area (1)</i> .....	10
<i>Set selection and management (3)</i> .....	10
<i>Visualization and Analysis tools (2 and 4)</i> .....	10
3.2 ONLINE HELP.....	10
3.3 WORKING WITH DATA FILES .....	11
3.3.1 <i>Workspaces – A brief overview</i> .....	11
3.4 THE PROJECT FOLDERS COMPONENT .....	12
3.4.1 <i>Example of opening a local microarray data file</i> .....	17
3.4.2 <i>Other file operations – Merging and renaming</i> .....	20
<b>4 QUERYING CAARRAY .....</b>	<b>23</b>
4.1 SEARCHING CAARRAY USING MAGE ANNOTATIONS .....	23
<b>5 MICROARRAY DATA ANALYSIS .....</b>	<b>29</b>
5.1 SET SELECTION (THE MARKERS/ARRAYS/PHENOTYPES) COMPONENTS.....	29
5.1.1 <i>Marker Sets</i> .....	29
5.1.2 <i>Set Activation and Manipulation</i> .....	32
5.1.3 <i>Array/Phenotype Sets</i> .....	32
5.1.4 <i>The Commands Menu</i> .....	32
5.2 THE VIEW WINDOW.....	33
5.2.1 <i>The Microarray Viewer</i> .....	35
5.2.2 <i>The Expression Profiles Tool</i> .....	37
5.2.3 <i>The Color Mosaic View</i> .....	37
5.2.4 <i>The Tabular View</i> .....	39
5.2.5 <i>The Marker Annotations and caBIO Pathways Views</i> .....	40
5.2.6 <i>The Image Viewer</i> .....	41
5.2.7 <i>Scatter Plot</i> .....	43
5.2.8 <i>Expression Value Distribution</i> .....	44
5.2.9 <i>Reverse Engineering with Cytoscape</i> .....	46
THE ANALYSIS/ANNOTATION WINDOW .....	47
5.3 FILTERING OPERATIONS .....	47
5.3.1 <i>Normalization Tools</i> .....	49
5.3.2 <i>Dataset History</i> .....	50
5.4 THE ANALYSIS TOOLS .....	50

5.4.1 Hierarchical Clustering:.....	51
5.4.2 Self Organizing Map (SOM).....	52
5.4.3 The Dataset Annotation Tool.....	54
5.4.4 The Experiment Info Tool.....	55
<b>6 SEQUENCE ALIGNMENT.....</b>	<b>57</b>
6.1 OVERVIEW.....	57
6.2 BLAST.....	57
6.3 TUTORIAL.....	57
6.3.1 Running BLAST.....	57
6.3.2 Analyzing the Results.....	61
6.4 COMPONENT LAYOUT AND OPERATION.....	61
6.4.1 Component Visual Elements.....	61
6.5 REFERENCES.....	63
<b>7 PATTERN DISCOVERY.....</b>	<b>64</b>
7.1 OVERVIEW.....	64
7.2 TUTORIAL.....	64
7.2.1 Discovery analysis.....	65
7.2.2 Hierarchical analysis.....	68
7.2.3 Exhaustive analysis.....	69
7.3 VISUALIZATION OF PATTERN DISCOVERY RESULTS.....	70
7.4 COMPONENT VISUAL ELEMENTS.....	70
7.5 REFERENCES.....	73
<b>8 PROMOTER ANALYSIS.....</b>	<b>75</b>
8.1 OVERVIEW.....	75
8.2 TUTORIAL.....	75
8.2.1 Transcription Factor signature analysis.....	75
8.2.2 Transcription Factor signature discovery.....	77
8.3 COMPONENT LAYOUT AND OPERATION.....	78
8.3.1 Component Visual Elements.....	78
8.4 REFERENCES.....	79
<b>9 ANALYSIS OF VARIANCE (ANOVA).....</b>	<b>80</b>
9.1 OVERVIEW.....	80
9.2 DEFINITIONS OF KEY TERMS.....	80
9.3 GUI LAYOUT.....	81
9.4 PARAMETERS SUBTAB.....	82
9.5 SERVICES SUBTAB.....	83
9.6 OUTPUTS AND VISUALIZATION.....	84
9.6.1 Project Folders.....	84
9.6.2 Tabular Viewer.....	85
9.6.3 Color Mosaic.....	85
9.7 DATASET HISTORY.....	86
9.8 RUNNING AN ANOVA JOB.....	87
9.8.1 Setup.....	87
9.8.2 Choose Settings.....	88
9.8.3 Run the calculation.....	89
9.9 REFERENCES.....	89
<b>10 USING CAGRID ANALYTICAL SERVICES.....</b>	<b>91</b>
10.1 USING CAGRID-BASED REMOTE ANALYTICAL SERVICES.....	91
10.2 HIERARCHICAL CLUSTERING.....	92
10.3 SELF-ORGANIZING MAPS (SOM).....	95
10.4 ARACNE.....	98

<i>Introduction</i> .....	98
<i>Prerequisites:</i> .....	99
<i>Understanding the Parameters</i> .....	99
<i>Running an ARACNE Calculation</i> .....	102
<i>Viewing the Results</i> .....	102
<b>11 USING CASCRIP TO AUTOMATE ACTIONS</b> .....	<b>104</b>
11.1 USING CASCRIP TO AUTOMATE TASKS .....	104
<b>APPENDIX A.    ERROR MESSAGES/INDICATORS AND PROBLEM RESOLUTIONS</b> .....	<b>111</b>
<b>APPENDIX B.    REFERENCES</b> .....	<b>112</b>
SCIENTIFIC PUBLICATIONS .....	112
TECHNICAL MANUALS/ARTICLES .....	112
CABIG MATERIAL .....	112
CACORE MATERIAL .....	113
<b>APPENDIX C.    GLOSSARY</b> .....	<b>114</b>

# 1 Introduction to the Manual

This manual is intended for the users of geWorkbench. It is directed at the bench scientist and bioinformatician. This manual does not provide installation instructions for geWorkbench nor for grid services. While extensive explanations of how to use the software are given in this manual, further tutorial information can be found on-line at [www.geworkbench.org](http://www.geworkbench.org).

This manual explains the basic principles, design goals, and uses of geWorkbench, centered around the single or joint analysis of gene expression microarray and sequence data.

This manual will cover the basic operations of geWorkbench and its core components. This revision of the manual pertains primarily to release 1.0.6. New modules for geWorkbench continue to be developed, and manual pages for them may be made available separately at [www.geworkbench.org](http://www.geworkbench.org).

## ***1.1 Changes since the previous manual, version 1.4***

This release of the manual adds a chapter on the ANOVA component for Analysis of Variance calculations. This chapter

## ***1.2 Changes since the previous manual, version 1.3***

This release of the manual incorporates material that was previously released as a separate supplement, entitled “Advanced Services”. Topics include:

1. use of geWorkbench within the context of the caGRID infrastructure. Several analytical routines already supported directly within geWorkbench have been developed as formal caGRID services, with an appropriate service interface present within geWorkbench. They are initially intended to be used in the analysis of microarray data. They are:
  - a. Hierarchical Clustering
  - b. SOM (Self-Organizing Maps)
  - c. ARACNE (a gene network reverse-engineering tool)

2. a query interface for caARRAY which allows searches on available annotation fields
3. use of the caSCRIPT scripting language developed specifically for geWorkbench to automate the running of repetitive or complex tasks.

### **1.3 Organization of the Guide**

<b><i>Chapter in geWorkbench User Manual Supplement 1</i></b>	<b><i>Chapter Contents</i></b>
Chapter 1	An introduction to using this Manual
Chapter 2	A general introduction to geWorkbench, and a high-level description of the advanced services covered in this manual.
Chapter 3	Description of the geWorkbench User Interface, including layout, basic file operations, and using Online Help
Chapter 4	Querying a remote instance of caArray for microarray data.
Chapter 5	This chapter covers the basic mechanisms for dealing with sets of markers and arrays, performing analysis and viewing results. The major common components of geWorkbench are covered.
Chapter 6	The use of BLAST within geWorkbench
Chapter 7	The use of the Pattern Discovery component for sequence analysis (SPLASH)
Chapter 8	The use of the Promoter component for analyzing promoter sequence elements.
Chapter 9	Accessing remote computational services via caGrid.
Chapter 10	The caScript language can be used to automate tasks within geWorkbench
Appendix A	Covers common error messages and problems
Appendix B	References
Appendix C	Glossary

### **1.4 Getting Started with geWorkbench**

To get started with geWorkbench you may refer to the following sections of this manual:

- Review Chapter 2 for a brief overview of the software
- Review Chapter 3 to learn about the Graphical User Interface, including basic file operations.
- Refer to Chapters 4 through 8 cover the basic a description of how to use the core modules of geWorkbench.
- For information on remote access to services via caGrid, consult Chapter 9.
- To learn how to use caScript to automate tasks, consult Chapter 10.

Detailed instructions and step-by-step tutorials on how to install and run geWorkbench are available online at <http://www.geworkbench.org/>.

## 1.5 Document Text Conventions

The following table shows various typefaces to differentiate between regular text and menu commands, keyboard keys, and text that you type. This illustrates how conventions are represented in this guide.

<b>Convention</b>	<b>Description</b>	<b>Example</b>
Bold & Capitalized Command Capitalized command > Capitalized command	Indicates a Menu command Indicates Sequential Menu commands	<b>Admin &gt; Refresh</b>
TEXT IN SMALL CAPS	Keyboard key that you press	Press ENTER.
TEXT IN SMALL CAPS + TEXT IN SMALL CAPS	Keyboard keys that you press simultaneously	Press SHIFT + CTRL and then release both.
Boldface type	Options that you select in dialog boxes or drop-down menus. Buttons or icons that you click.	In the Open dialog box, select the file and click the Open button.
<i>Italics</i>	Used to reference other documents, sections, figures, and tables.	<i>caCORE Software Development Kit 1.0 Programmer's Guide</i>
<i>Italic boldface type</i>	Text that you type	In the New Subset text box, enter <i>Proprietary Proteins</i> .
Courier typestyle	Used for filenames, directory names, commands, file listings, source code examples and anything that would appear in a Java program, such as methods, variables, and classes.	URL_definition ::= url_string
Note:	Highlights a concept of particular interest	Note: This concept is used throughout the installation manual.



Warning!	Highlights information of which you should be particularly aware.	Warning! Deleting an object will permanently delete it from the database.
{}	Curly brackets are used for replaceable items.	Replace {root directory} with its proper value such as c:\cabio

Table 1. 1 Document Conventions

## 2 Overview of the Software

### **2.1 Motivation**

Recent advances in high-throughput genomic technologies, spurred on in part through the Human Genome Project, have opened the flood-gates to many different types of biological data. For example, NCBI provides open access to genome sequences of over 1000 organisms; nucleotide and protein sequences (e.g., GenBank, RefSeq, Swiss-Prot, PIR etc.), 3D macromolecular structures; population study data sets, catalogs of human disease genes, genetic markers or tagged-sites database (SNP, EST, STS), molecular modeling and genome mapping information. These developments directly influence biomedical research. However, making use of this cornucopia of information is difficult for investigators because most laboratories lack the tools to integrate the data into their own studies.

Although a large selection of bioinformatics software tools is available, these have been developed as individual software programs and do not readily interface with other software. Differences in application design, programming language used for implementation, and input/output requirements restrict their use to certain operating systems, and/or impose data reformatting requirements. Furthermore, management of any complex biological data (e.g. combining output from two different gene-expression clustering tools) usually requires custom programming, because even though concepts such as a gene expression cluster are well understood and ubiquitous in the literature, their representation has not been standardized.

### **2.2 Introduction to *geWorkbench***

*geWorkbench* 1.0 is an open-source platform for bioinformatics data analysis . It supports a growing collection of self-contained software modules for management, analysis and visualization of a range of biological research data. It also provides integration of external databases and services into the local desktop client. The overriding goal of *geWorkbench*1.0 is to provide biomedical researchers with a user-friendly application that can link the analysis of disparate data types. It is an extension of a project originally sponsored by the National Cancer Institute Center for Bioinformatics (NCICB) to develop tools for microarray data analysis (*caWorkBench* 1.0).

*geWorkbench* has been primarily constructed for analysis of data derived from gene expression microarray experiments, and allows pulling in many different resources to this

end, including sequence, gene ontology, promoter analysis, and standard analytic techniques such as the t-test, hierarchical clustering, and gene network reverse-engineering.

geWorkbench has a modular, component-based design. New modules can easily be written and added as the need arises. A primary aim is to allow easy integration of different forms of data analysis. Such integration removes the common hindrance of needing to reformat data for each different type of analysis undertaken.

Extensive documentation and training material for geWorkbench can be found on its main website at <http://www.geworkbench.org/>. There are wiki-based tutorials there for almost all components of the application. These tutorials are more applied in nature than the material in the printed manual. The software can be downloaded via links found on the 'Download' section of that site. Those links refer to the actual archival location of the software, which is the GForge site maintained by the NCICB. All official releases of the software can be downloaded from that site.

This manual provides a detailed view of the modules that make up the core functionality of geWorkbench1.0. Tutorials and examples are available on the application website, [www.geworkbench.org](http://www.geworkbench.org). The application download area can be reached from that site or directly from <http://gforge.nci.nih.gov/projects/geWorkbench/>. Information about additional modules not covered in this manual can also be found at [www.geworkbench.org](http://www.geworkbench.org).

### ***2.3 The geWorkbench Approach to Integrated Genomics***

We believe that biomedical researchers will be best served by the establishment of a standardized, fully integrated bioinformatics software infrastructure (such as geWorkbench1.0) that supports not only heterogeneous data and models, but also algorithms, management, and visualization tools that can be seamlessly integrated and distributed within the biomedical scientific community. It is this realization that is the central motivation of the geWorkbench1.0 framework. The latter, then, attempts to address the following needs:

1. Sharing not just a growing set of biological data types and data sets, but also a growing set of application software tools to manipulate them.
2. Allowing different modules to interact with each other based on their semantic compatibility (that is, the programmatic interfaces they implement). This is like having two individuals that communicate not just because they have a “vocabulary” translating individual words into their native languages but because they know how they are assembled into meaningful sentences and concepts (semantics).
3. Supporting automatic event-driven computations and data analysis and visualization workflows in a distributed environment that operates transparently to the end-users.

## **2.4 Components of geWorkbench**

**geWorkbench:** geWorkbench v.1.0 is a Java application which is run on the User's local Windows, Macintosh or Linux workstation. This main application also serves as a front-end client to a number of external computational and data services. Such services already present in geWorkbench include the ability to run BLAST jobs on NCBI servers, and to retrieve gene, pathway and sequence information from sources such as UC Santa Cruz and the NCICB. A built-in, interpreted, Java-like scripting language, caSCRIPT, can be used to automate tasks within geWorkbench.

**caGRID:** caGrid is a project sponsored by the National Cancer Institute to link and make available data stored in Cancer Centers throughout the United States. . caGrid provides a mechanism for all data and parameters passed on the grid to be of known, registered types, to facilitate interoperability. geWorkbench can be used as a client for such grid services and several such modules have been implemented.

**caArray:** caArray is a MIAME-compliant database system for microarray data developed by the National Cancer Institute. geWorkbench supports querying against such databases and retrieval of expression information.

Central to data integration in geWorkbench1.0 is a mechanism that allows independently built tools and data sources to communicate in a meaningful fashion. This mechanism, termed "component semantics interoperability," facilitates construction of complex biomedical applications from simple components, much like building complex assemblies from Lego pieces. It is implemented through the exchange or broadcast of well-defined messages.

## **2.5 Supported Datatypes**

geWorkbench currently is oriented towards the integration of microarray gene expression and sequence data. Examples of data types handled by include:

- microarray gene expression data (Affymetrix, GenePix)
- Sequence data
  - (e.g., DNA, RNA, protein sequences)
- complex multi-dimensional data-types
  - biochemical pathways
  - gene regulatory pathways

Examples of external data sources and services provided through geWorkbench1.0 include

- Cluster-accelerated version of BLAST
- Server-side implementation of pattern discovery algorithms.
- GoldenPath genome sequence retrieval.
- Access to NCI databases including
  - CGAP gene annotations
  - BioCarta pathway diagrams
  - caArray gene expression data

# 3 Visual Interface and Data Management

## 3.1 Layout of the geWorkbench1.0 Interface

Figure 3-1 shows a screenshot of geWorkbench1.0's graphical interface. The workspace is divided into 4 resizable panels whose functionality is further defined by the folder tabs running across the top of each panel. Each panel can be arbitrarily resized by clicking on an edge of that panel's frame and dragging the mouse. In addition, the triangular shaped wedges (on the left sides of the horizontal separators and at the top of the vertical separator) can be clicked on to maximize that frame vertically and/or horizontally.

Each of these configurable panels is described in more detail in the sections that follow; the purpose of this section is to provide an overall orientation. Moving from left to right and from top to bottom, these panels include a *Project* window (1), a *View* window (2), a *Selection* window (3), and an *Analysis/Annotation* window (4).

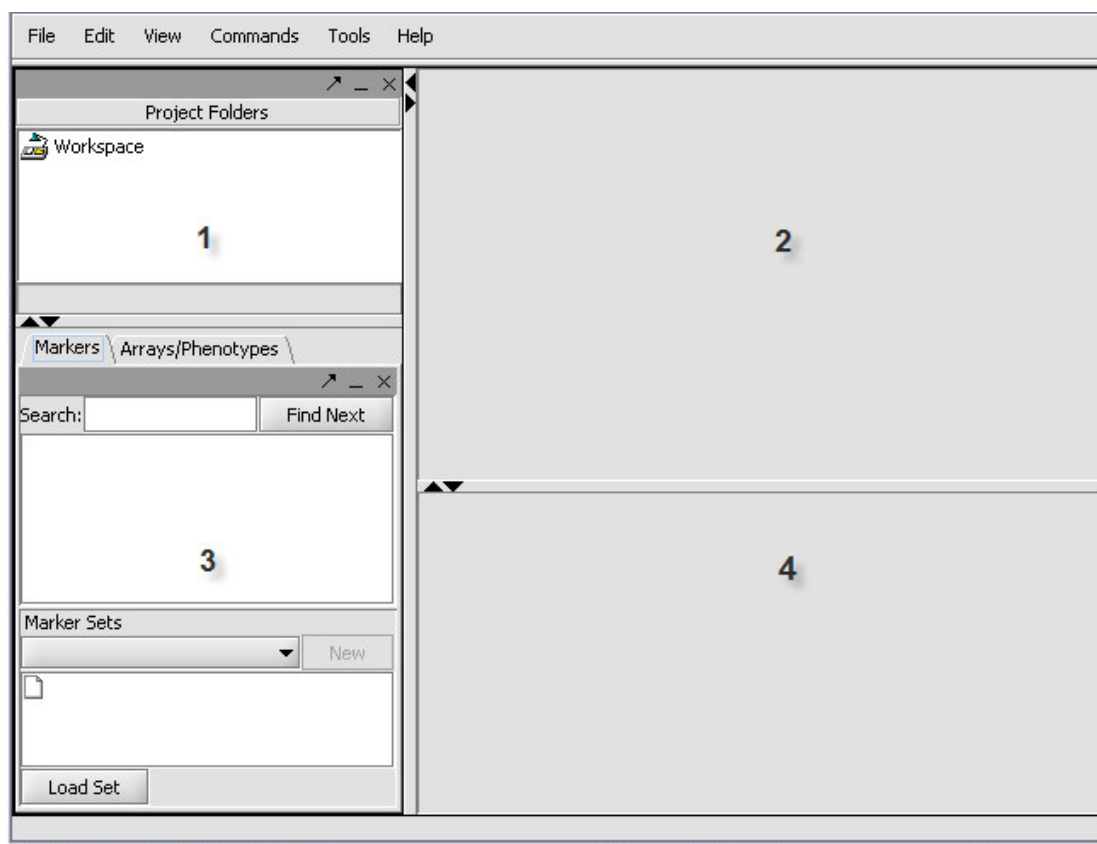


Figure 3-1 Layout of the geWorkbench1.0 Graphical Interface

## Menu Bar

The GUI provides a menu bar at top with a standard choice of commands. Many commands that are available in the menu bar are also available by right-clicking on data objects.

### Data management area (1)

Working with geWorkbench involves creating a project within the top-level workspace. Open data files and the results of data transformation or analysis are stored within a project. A workspace can contain more than one project at a time, allowing data to be organized as desired. A workspace and all the projects and data within it can be saved and later reloaded.

### Set selection and management (3)

A key feature of geWorkbench is the ability to work with defined sets of markers or arrays. This allows subsets of data to be analyzed, and allows for passing of selected subsets of data between different components. For example, the t-test can be used to create a list of markers showing a significant difference in expression between two states, and this list can then be used to retrieve relevant sequences or annotations.

### Visualization and Analysis tools (2 and 4)

geWorkbench works such that only the visualization and analysis components relevant to the type of dataset currently selected in the Project Folders area (1) are displayed through tabs in their respective areas (2 and 4). Thus choosing a microarray dataset will result in a different set of tabs being displayed as compared with those seen when a nucleotide sequence file is selected. When a new data file is loaded, or an analysis produces a new data set, not only is it added to the Project area (1), but an appropriate viewer in the Visualization area (2) is automatically selected.

## 3.2 Online Help

Figure 3-2 shows the **Online Help** interface. **Online Help** is found as a menu item under **Help** on the top menu bar. **Online Help** is provided for all geWorkbench modules which have been included in a formal release. They focus on the actual use of particular controls within a given module, e.g. button actions, definition of parameters etc.

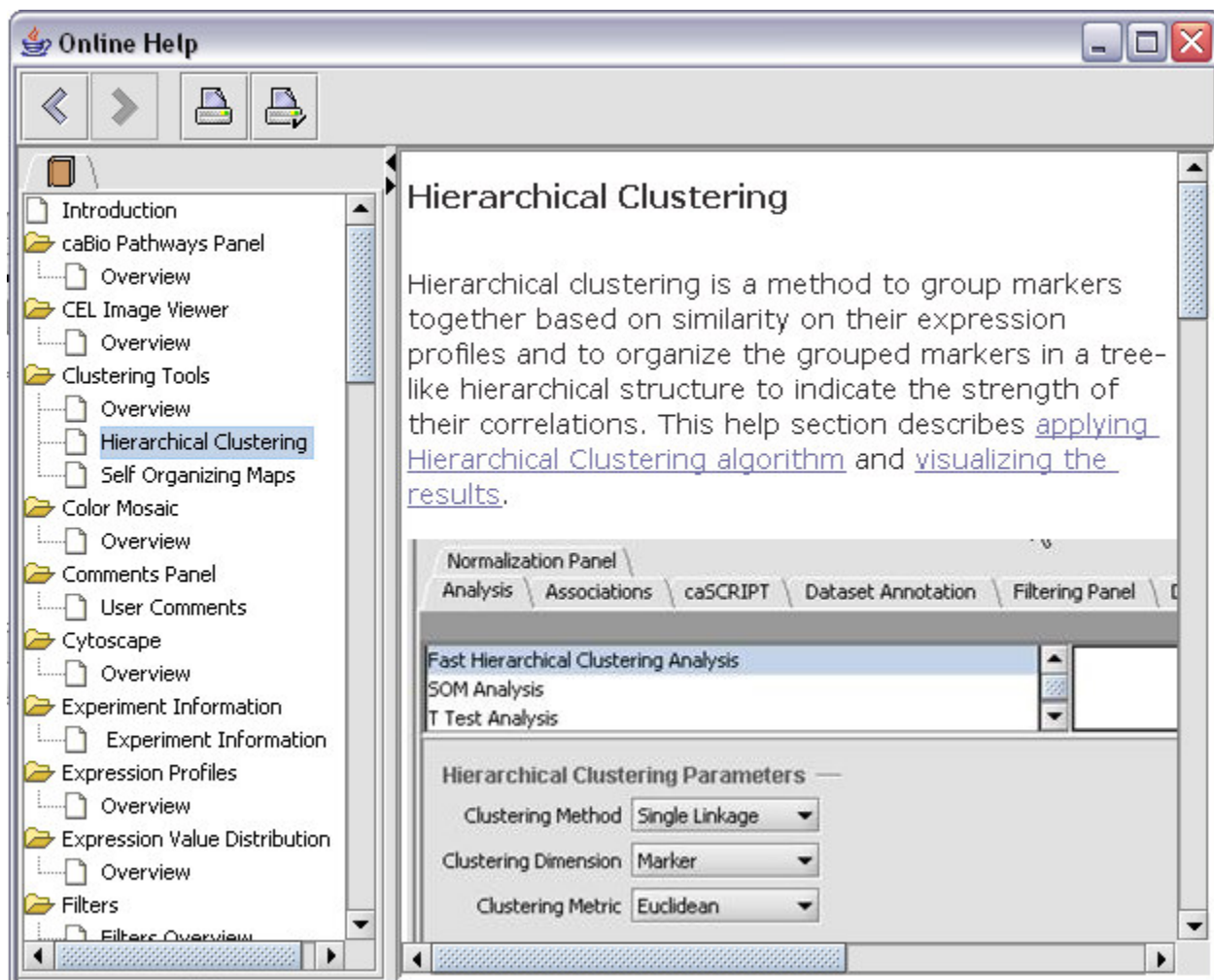


Figure 3-2 Online Help

## 3.3 Working with Data Files

### 3.3.1 Workspaces – A brief overview

The top level of organization of data in geWorkbench is the Workspace. A Workspace can contain any number of Projects, which are used to organize data and results.

geWorkbench1.0 organizes data files using a *workspace/project* paradigm. A project is analogous to a “virtual” folder, as it allows individual data sets to be grouped together without modifying their physical storage locations. Once a project’s data sets have been defined, it is possible to open, save, or close all data sets in that project with a single action.



A typical use of the project facility is to associate multiple data sets from the same experiment with one another in a single project folder. In addition to loaded data files, other types of data generated during the session—e.g., images, results from various steps analysis such as clustering, etc.—are also saved associated with their parent dataset in a particular project.

Multiple projects can, in turn, be managed within a single workspace. A user can create a project in a workspace, delete an existing project from a workspace, or rename a project. The application supports handling an arbitrary number of projects in a single workspace.

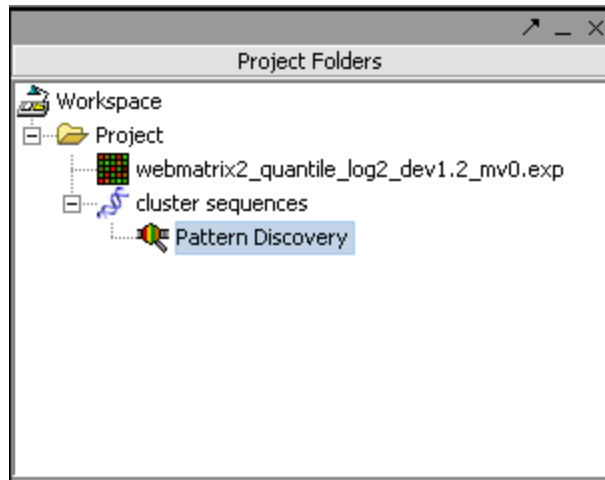
In summary, a workspace may contain multiple projects, which may themselves contain a variety of raw, filtered, normalized, or otherwise annotated microarray data sets. Workspaces are saved as files with a *.wsp* extension. Projects can only be saved and/or accessed as part of a workspace. **Note – in current versions of geWorkbench, the saved workspace is dependent on the particular version of Java installed on the machine, and thus is not suitable for long-term data storage.**

In addition to the controls provided within each individual component, a main menu bar appears at the top of the screen. The first menu option, **File**, is used to manage the opening, creating, deleting, and saving of workspaces, projects, and files. Most of the options included in the main menu bar are applicable to the Project and Marker/Phenotype windows, and are described below.

### **3.4 The Project Folders component**

The Project Folders component provides a centralized area for managing projects and files in the current workspace. Operations on this window are controlled by the **File** and **Edit** options in the main menu bar. Most of these menu options can also be accessed by right clicking the mouse when it is located over an appropriate element in the Project or Marker/Phenotype window.

The operations described here all manipulate the workspace, projects, files, and images that are visible in the Project Folders component. Only a single workspace can be open at one time, but multiple projects, files, and images can be managed within that workspace. The Project Folders component uses a hierarchical treelike structure to manage these elements, as shown by the example in Figure 3-3.



**Figure 3-3 An Example Project Tree**

Used to capture a “working session,” the workspace is represented by a folder icon at the very top of the file hierarchy into which all of the data generated during a user session can be subsumed. Items contained in a workspace include projects, which may themselves contain a variety of raw, filtered, normalized or otherwise annotated microarray data sets.

Multiple projects can be accommodated under one workspace heading. Multiple data sets and derivatives thereof can be grouped within a single project. It is important to note, however, that some operations require data sets to be part of the same project, and in some cases, in the same file. For instance, two microarrays cannot be viewed side by side unless they have been merged into one file. Moreover, two data sets cannot be merged into one file unless they are included in the same project.

Figure 3-4 illustrates the expanded options for the first three operations under **File**, i.e., **Open**, **Save**, and **New**. These operations can be selectively applied to open, save, or create workspaces, projects, and files,.

Not all operations are applicable to all types of elements however. As noted above, projects can only be opened, saved, or created as part of a workspace. Similarly, files can only be opened and/or saved as part of a project, and cannot be explicitly created. Selecting the **File**→**Open**→**File** operation without having first defined a project in which to open that file will generate a prompt advising the user to first select a project in the Project window.

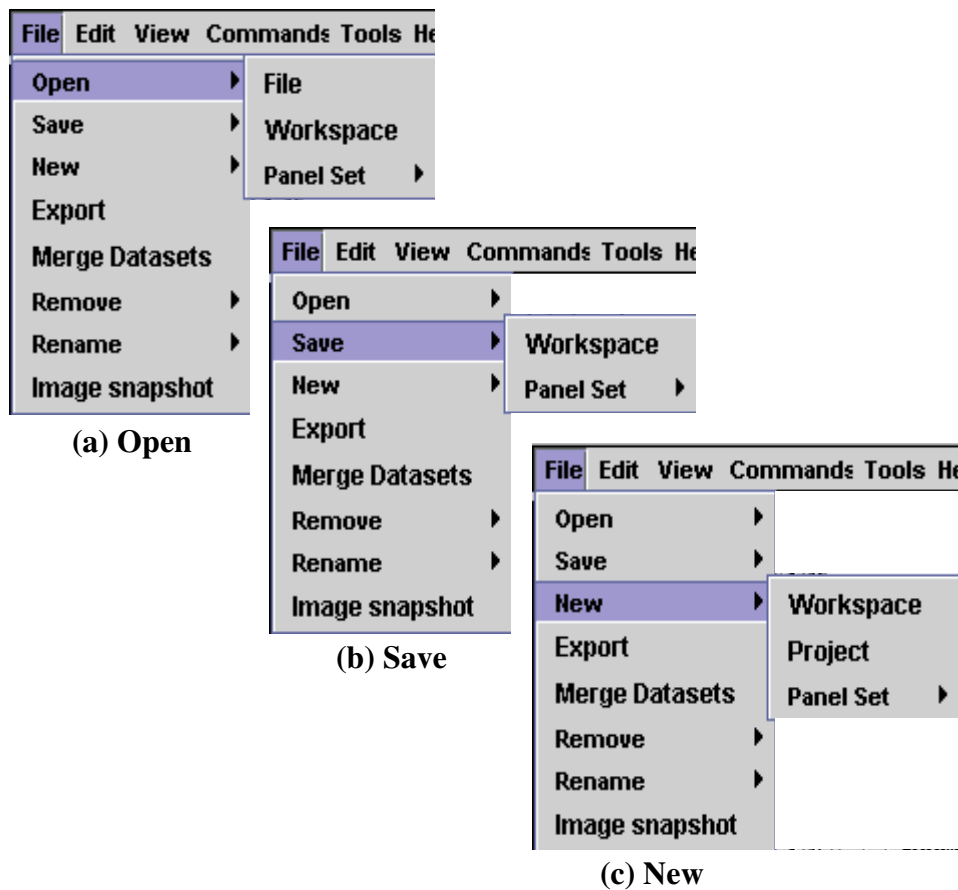


Figure 3-4 Open, Save, and New Operations

Several of the operations in the drop-down **File** menu are not applicable to the Project window. We list these operations here for completeness, but defer any further discussion to the relevant sections.

When the application starts up, a blank workspace is created. A new project can be created in the workspace by selecting **File→New→Project** from the drop-down menu (

Figure 3-4(c)). Data files can then be added to the project by selecting that project in the Project window and using the **File→Open→File** option (

Figure 3-4 (a)). The workspace can be saved in a *.wsp* file using the **File→Save→Workspace** option and reopened at any future time using the **File→Open→Workspace** option.

The **File→Save** option in

Figure 3-4(b) is applicable to workspaces only; files and projects are implicitly saved as constituents of a workspace. The **File→New** option in

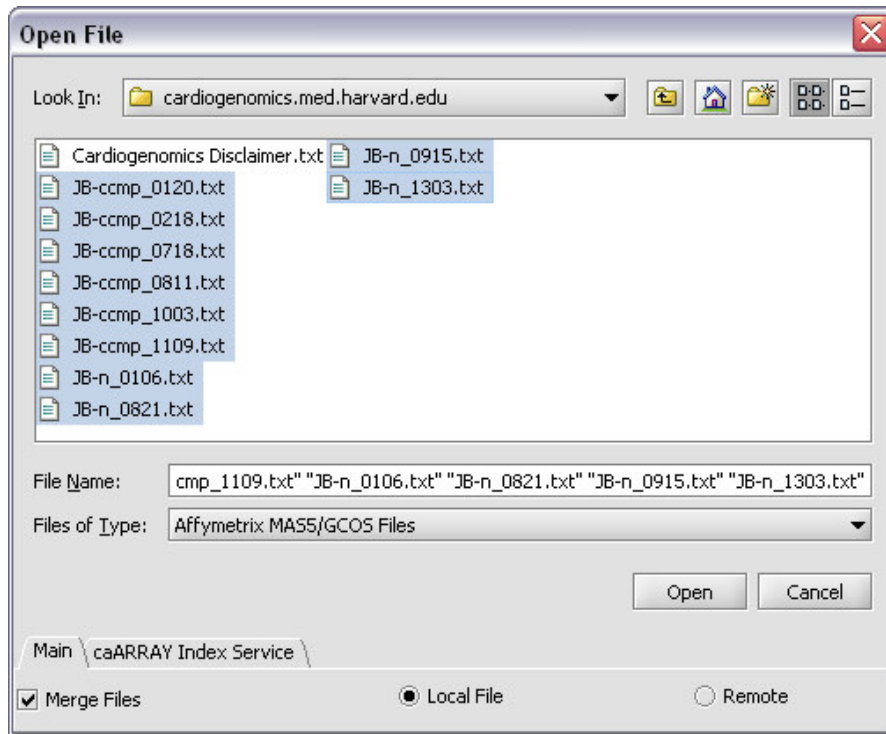
Figure 3-4(c) is only applicable to workspaces and projects; “new” data sets can only be derived by performing certain operations on existing data. These derived data sets will, however, be saved as part of the project in which they were derived.

The default option in the pop-up "Open File" dialog box is to open a local file, as indicated by the pre-selected radio button at the bottom of the dialog box in Figure 3-5(a). Selecting the appropriate file type from the pull-down list in the local file dialog box will display all files of that type. Selecting a file from the list of those available and clicking on the **Open** button or double-clicking will then close the file dialog box and add that file to the current project. A small set of example datasets are available with the download package, in the *Sample Data* directory, and additional examples are available as part of the Tutorial dataset at [www.geworkbench.org](http://www.geworkbench.org).

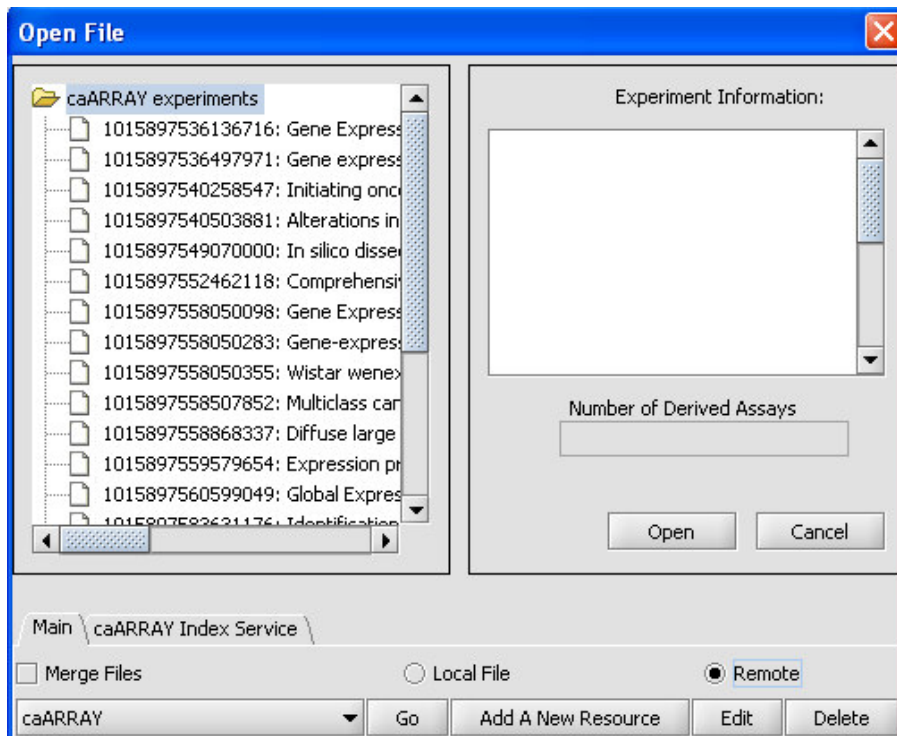
Alternatively, to access remote data sets stored on a remote source such as NCI's caArray server, you must first select the remote radio button at the bottom of the dialog box, as shown in Figure 3-5(b). As seen there, the left-hand panel lists the files available for selection, and a scrollable text window on the right displays information about the currently selected experiment.

To open a file, begin by right clicking on the selected experiment in the left panel. Selecting a file from that list and pressing **Open** in the right panel will import that file to the current project.

In summary, the **File→Open** command can be used to add files to a selected project and to open workspaces. Files can be opened from a local disk, and in the case of microarray gene expression data from a remote instance of caArray, and are always saved as part of a project. Finally, opening or creating a new workspace will drop any unsaved work that has been done in the current workspace, so be sure to save your work before performing either of these actions.



(a)



(b)

Figure 3-5 The Local (a) and Remote (b) Open File Dialog Boxes

### 3.4.1 Example of opening a local microarray data file

#### Prerequisites

Certain aspects of the functionality of geWorkbench currently depends on microarray annotation files. For example, such files are supplied by Affymetrix for their microarray chips. Due to licensing restrictions, these Affymetrix files are not distributed with geWorkbench as part of formal releases. The examples in this portion of the User Manual do not depend on annotation information. If you nonetheless would like to work with the full functionality of geWorkbench, the relevant file for the dataset used in this manual can be downloaded from the Affymetrix.com support web site. The file is named “HG\_U95Av2\_annot.csv”.

#### Example

geWorkbench includes sample data files. In the example below we will open a small microarray data file, `web100.exp`. This file is in a custom format used by geWorkbench, which is termed the **Affymetrix File Matrix** format. It contains results from a number of different microarray chips that have been merged into one dataset.

Opening a file in a new Project (see Figure 3-6)

1. Right-click on Workspace and select **New Project**.
2. Right-click on **Project** and select **Open File(s)**.
3. By default, the file browser should open in the geWorkbench data directory. Select **File of Type** to be **Affymetrix File Matrix**.
4. Select the file `web100.exp`.

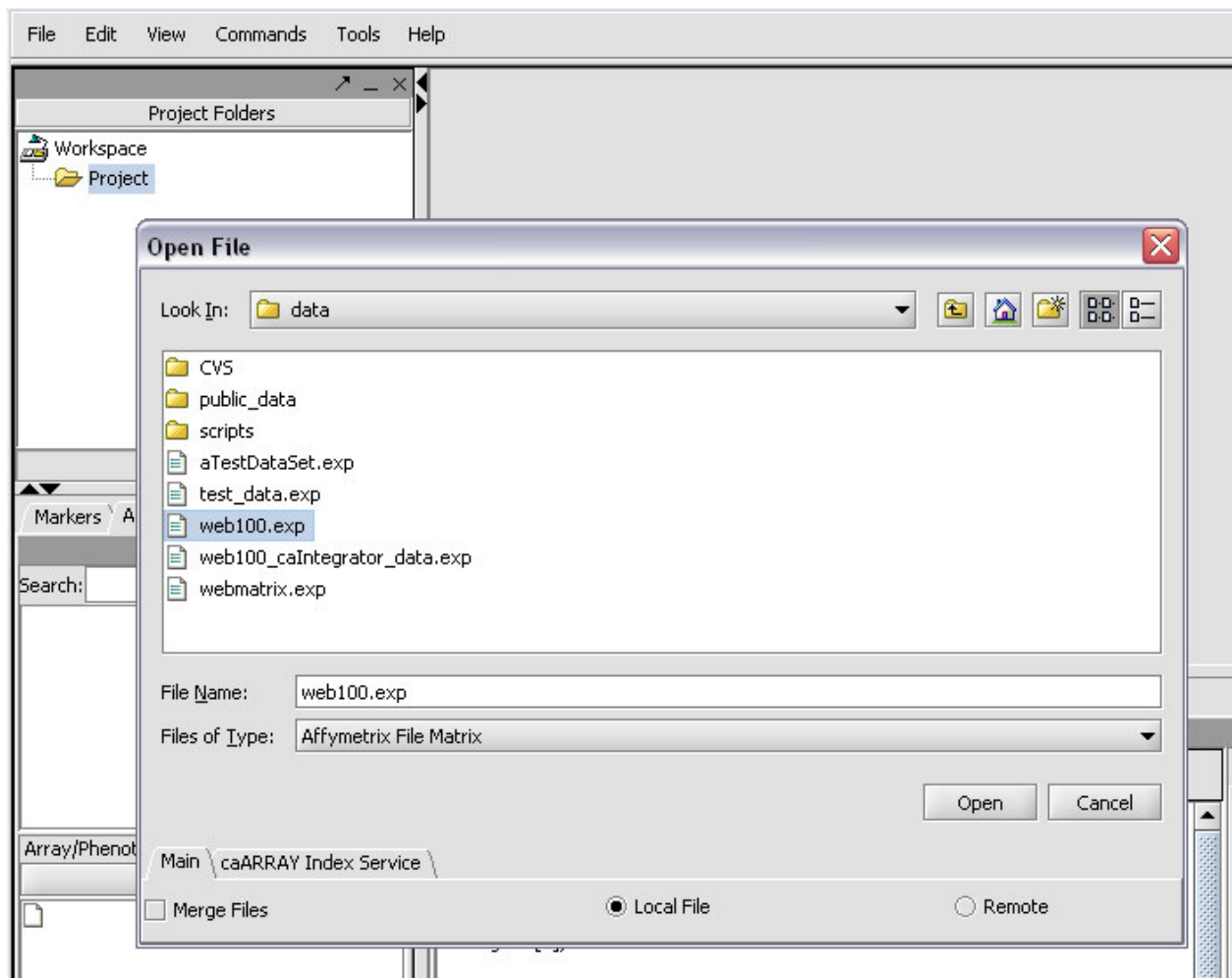


Figure 3-6 Opening a file in a project

5. A box with information about annotation files will appear. Click **Continue**.
6. The file browser will open at the root of the geWorkbench installation directory (Figure 3-7). If the file `HG_U95Av2_annot.csv` is present, just press the **Open** button. If you have downloaded it to another directory, please navigate to that directory and open the file. If you do not have the file, just press **Cancel** and proceed without the annotation file.

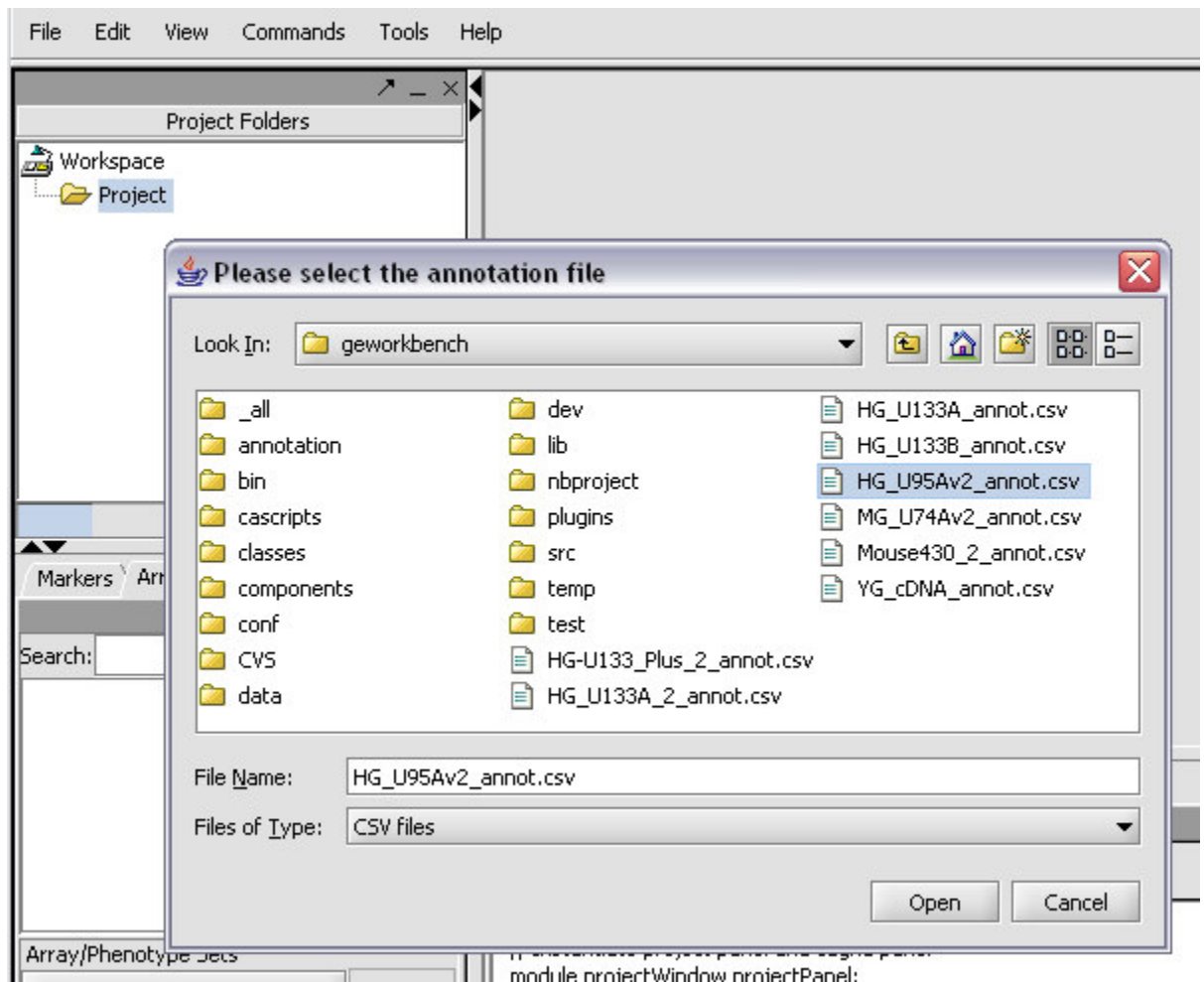


Figure 3-7 Opening the annotation file

The opened data file is now shown within the **Project Folders** area at upper left in the GUI. All components relevant to acting on microarray data have now appeared in the interface Figure 3-8.



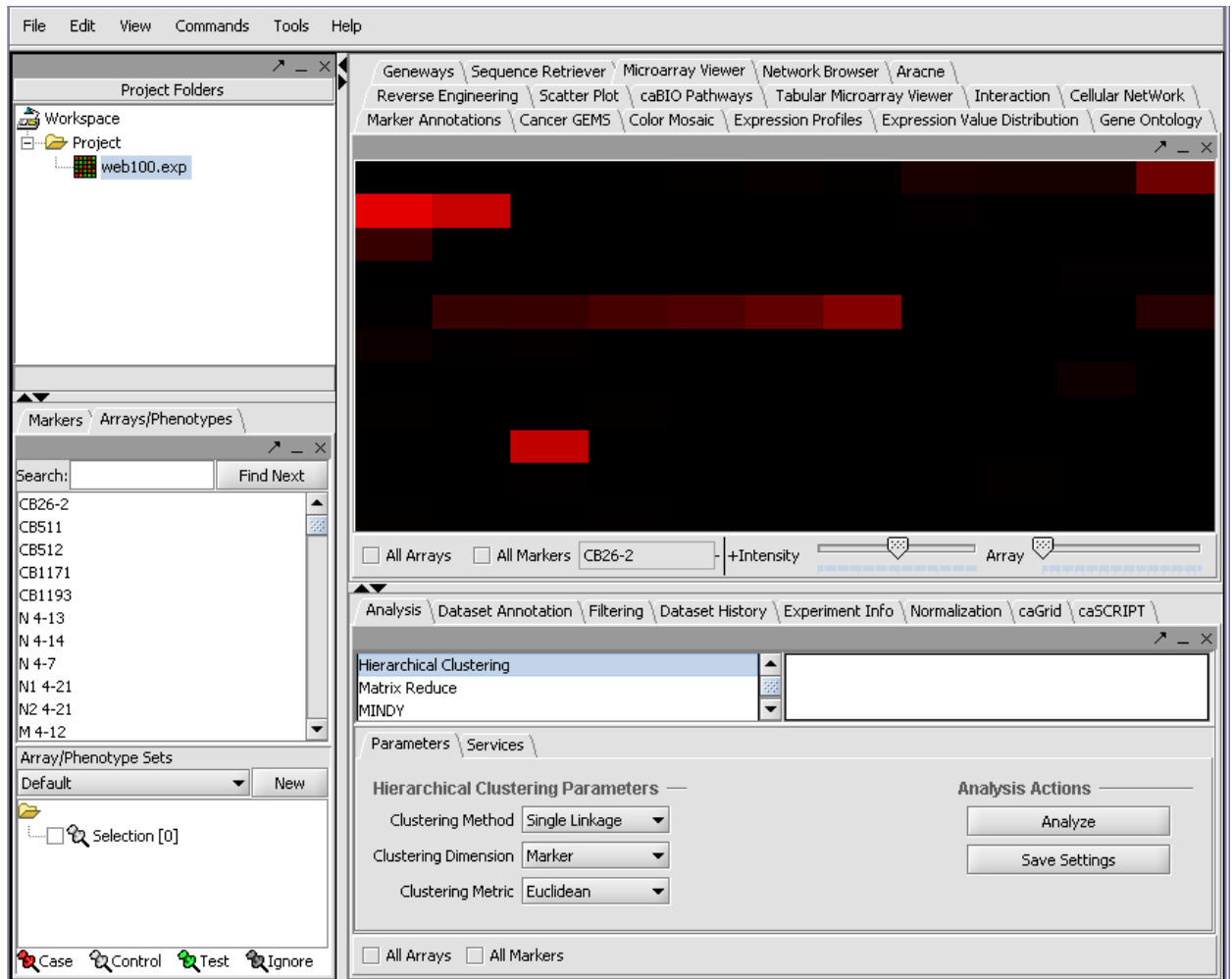


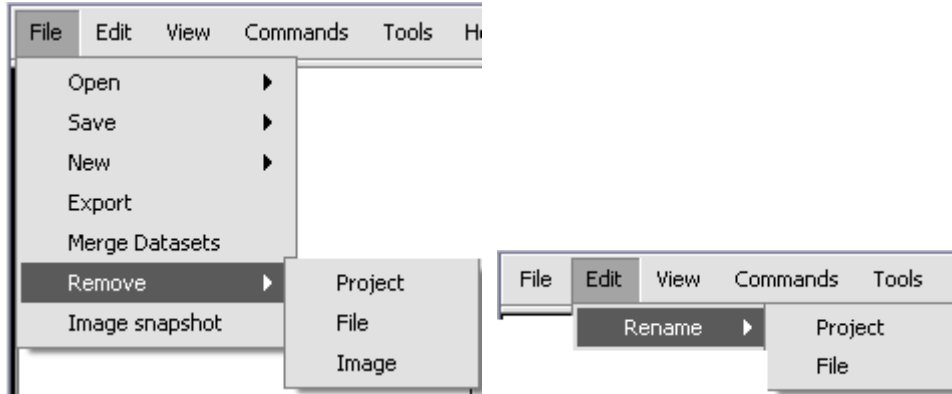
Figure 3-8 geWorkbench GUI showing all microarray-related modules

### 3.4.2 Other file operations – Merging and renaming

The next two options in the **File** pull-down menu are **Export** and **Merge Datasets**. The export option can be used to save the data in a data file or a selected image (see Section 3.1.1) in a new format. The **Merge Datasets** operation combines two or more microarray data sets generated using the same platform to produce a single set (Chapter 3). *Only those data sets included in the same project can be merged.*

Figure 3-9 shows the expanded drop-down menus associated with the next two file operation. The **File→Remove** option is used to remove images and files from a project, and projects and marker panels from a workspace. In all cases, the user must first select the object to be removed in the Project window before executing the operation.

It is possible to rename projects and data files from within the Project Folder component by right-clicking on the desired element and using the shortcut pop-up menu. The drop-down **Edit->Rename** menu in the main menu bar also provides this option.



**Figure 3-9 Removing and Renaming**

The last option in the **File** drop-down menu is the **Image snapshot** operation, which is used to take snapshots in the View window and is described in Section 5.2.6 . For convenience, all of the file operations are summarized in Table 3.4-2.

**Table 3.4-1**

Command	Arguments	Action
<b>File→Open</b>	files, workspaces,	Opens a new workspace, or a file in the current workspace. Clicking the “Merge Files” checkbox will combine the selected microarray files into a single dataset.
<b>File→Save</b>	workspaces	Saves the workspace
<b>File→New</b>	workspaces, projects,	Creates a new workspace, or a new project in the current workspace.
<b>File→Export</b>	images, files	Saves an image or data set in a new format.
<b>File→Merge Datasets</b>	files	Merges two or more microarray data sets to generate a single combined set.
<b>File→Remove</b>	projects, files, images	Removes files and/or images from a project, or projects from a workspace.
<b>File→Image snapshot</b>	objects in the View window	Takes a snapshot of an object in the View window.

**Table 3.4-2 Summary of File Operations in the Main Menu bar**



## 4 Querying caARRAY

This chapter describes how geWorkbench can query remote instances of a caARRAY database. caARRAY currently uses a MAGE-OM compliant API which allows searching on a number of common annotation data fields, such as species, array type and tissue type.

### **4.1 Searching caARRAY using MAGE annotations**

caARRAY is a microarray gene expression repository developed by the NCICB which supports storing and querying of annotated datasets. The annotations conform to the MAGE (Microarray Gene Expression) model. It should be noted that actual datasets may be only partially or sparsely annotated, but the model has the capability to store, in a rigorous fashion, almost any type of information about an experiment.

In the current implementation, geWorkbench can query against four types of annotations supported by caARRAY:

- Tissue type (available choices are displayed)
- Chip Platform (e.g. Affymetrix, Agilent etc.)
- Organism
- Principal Investigator

The following example illustrates constructing a query against caARRAY.

1. Create a new project as described previously.
2. Right-click on the **Project** entry and select **Open File(s)**.
3. Click the **Remote** radio button. This will cause the Open File popup to switch to the remote file interface (Figure 4-1).

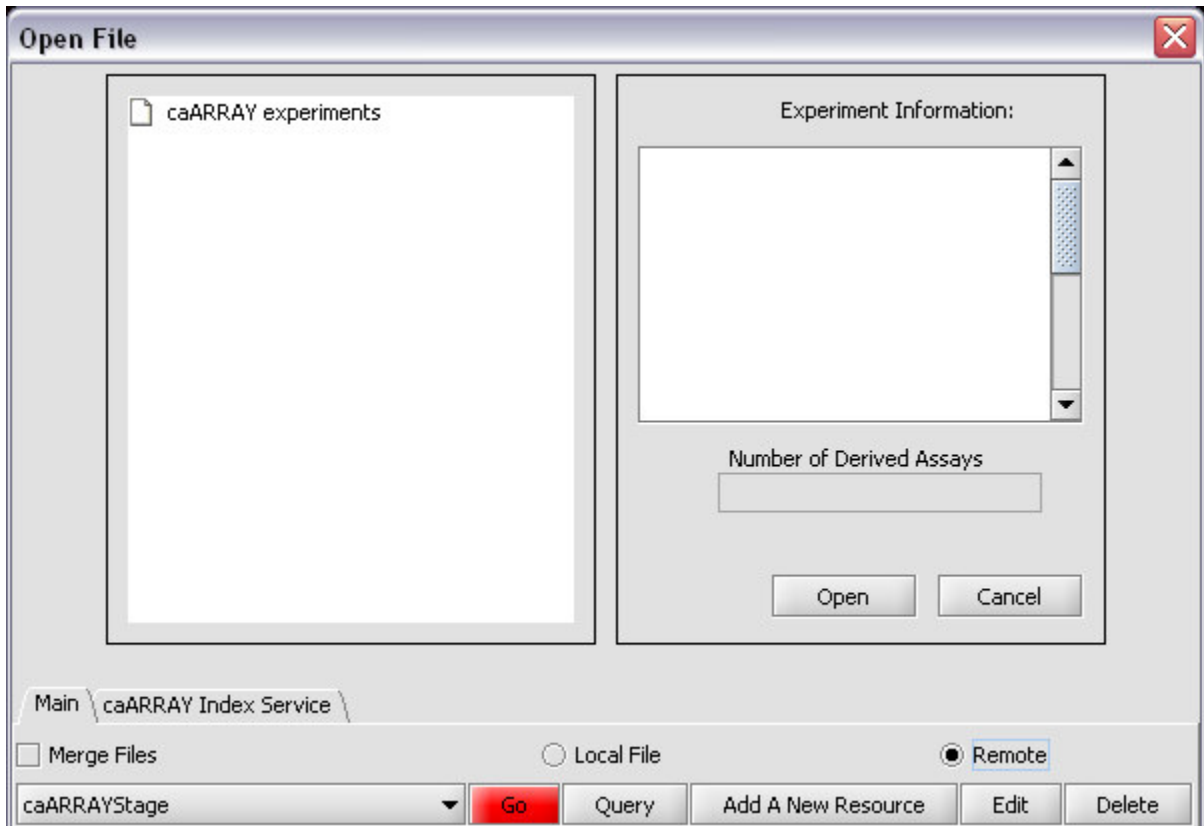


Figure 4-1 The Remote Open File interface

You can inspect the **caARRAY Index Service** tab (Figure 4-2). As more services become available, the desired service can be entered here:

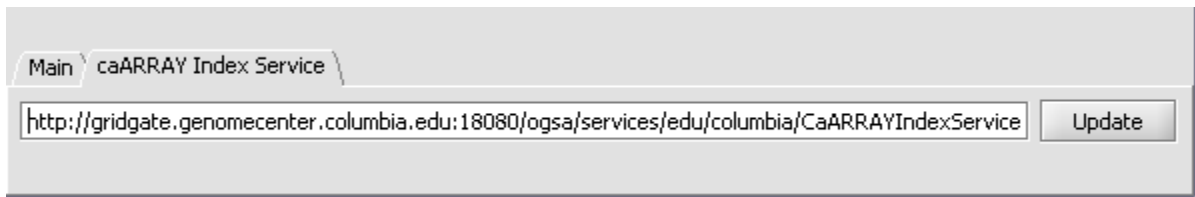


Figure 4-2 Inspecting the caARRAY Index Service entry

Clicking on the red **Go** button would retrieve all available experiments. Instead, we will construct a query.

4. Click on **Query** (see above) to build a MAGE keyword search. The query builder appears (Figure 4-3).

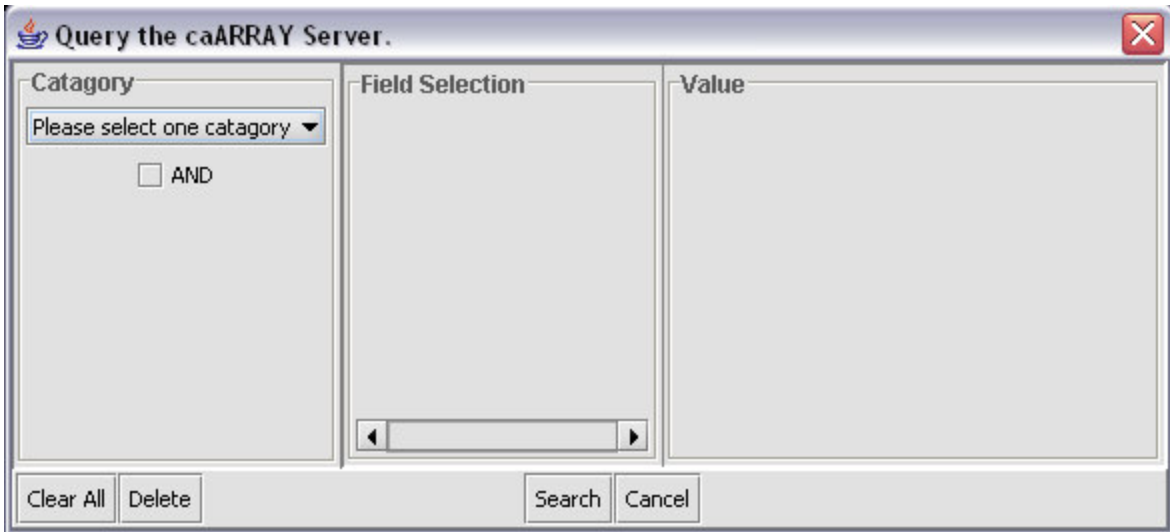


Figure 4-3 The caARRAY MAGE query interface

5. Under Category, select **Experiments** (Figure 4-4). The available search field types will be displayed. Here we will search on **Tissue Type**. Highlighting this field shows available tissue types.
6. Here we have selected the tissue type **ovary**.

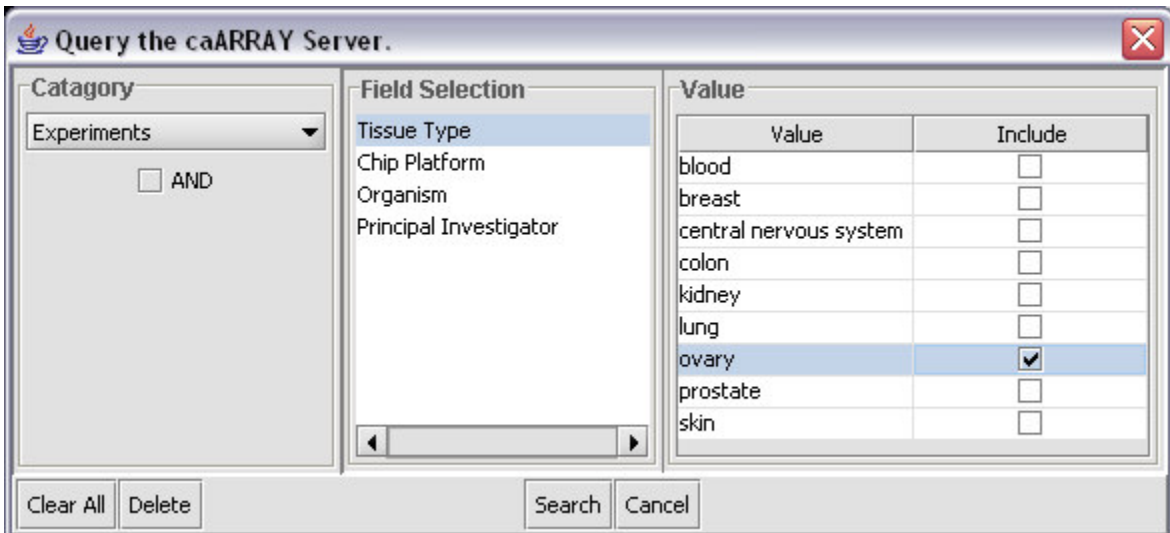


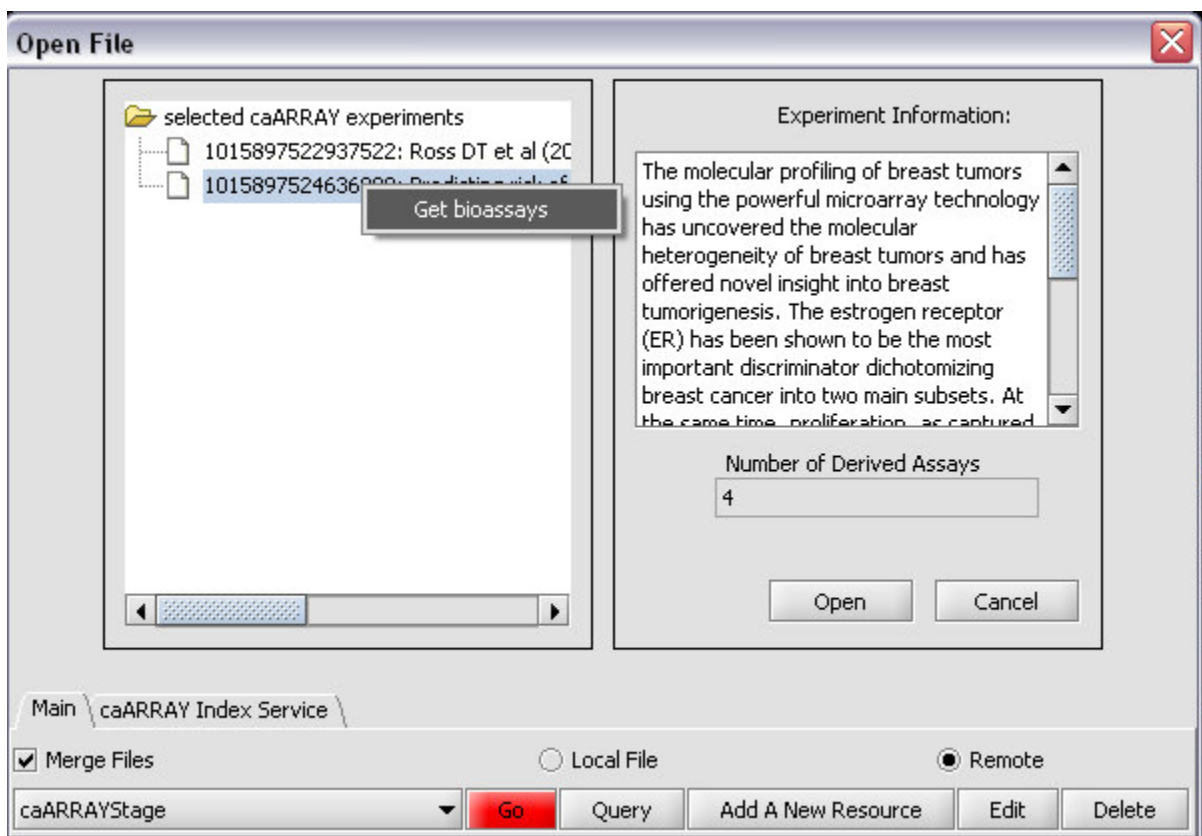
Figure 4-4 Constructing a new caARRAY query

**Notes on query building –**

- Use of the **AND** checkbox – As currently implemented in

caARRAY, **AND** queries can only include items from different fields as defined above. That is, you could search for a **Tissue Type AND a Principal Investigator**". If the **AND** box is not checked, only the currently visible field will be used in the query.

- Implied **AND** within a field – If e.g. in Tissue Type you were to select two different tissues, the query would be for type A **AND** type B. This is a limitation of the current caARRAY implementation. No **OR** functionality is yet available.
7. Click **Search**. (see Figure 4-4 above) Experiments matching the search term are returned (Figure 4-5).
  8. Selecting an experiment will display information about it in the boxes at right, as here we have selected the second entry.
  9. After selecting an entry, right click on it – the **Get bioassays** button appears.



**Figure 4-5 Retrieving the list of bioassays for an experiment**

10. Clicking on **Get bioassays** returns the available bioassays for this

experiment. (**This hidden step is the most easily overlooked one in geWorkbench**). The returned bioassays are shown in Figure 4-6.

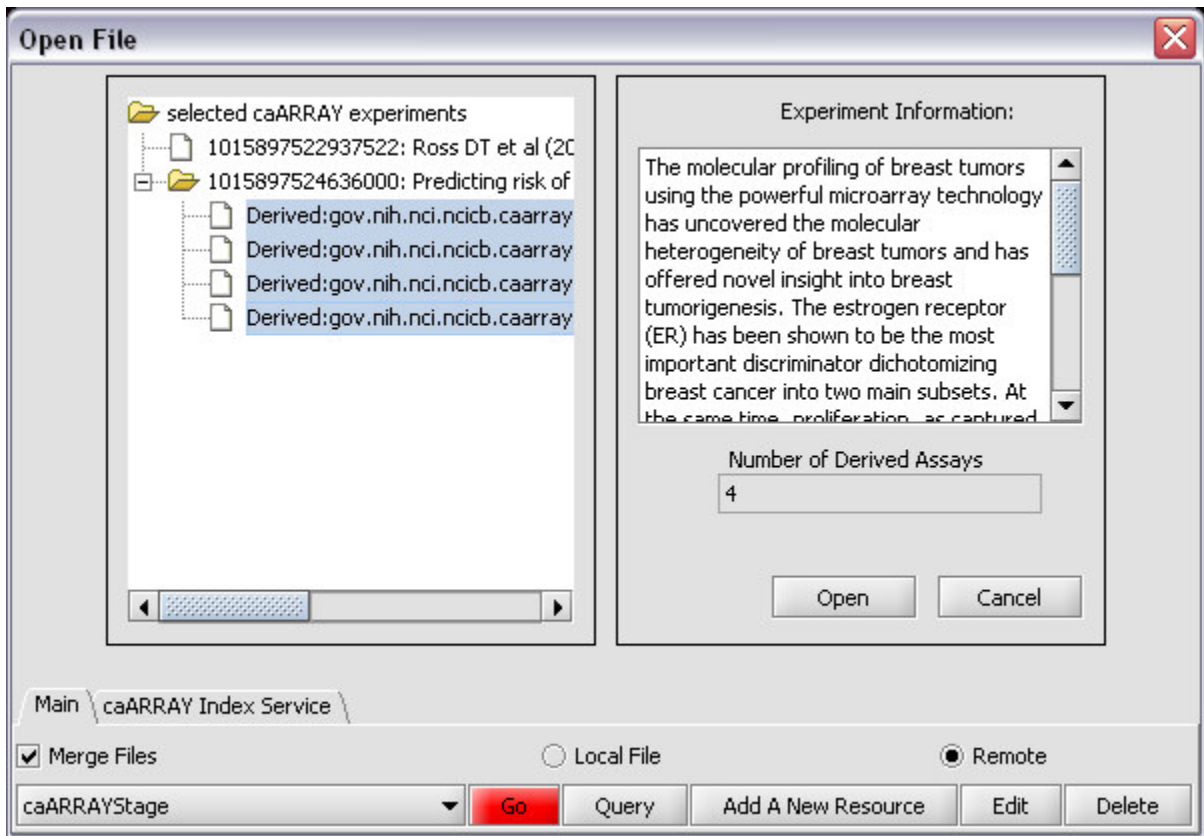


Figure 4-6 View of retrieved bioassays

11. Click on the **Merge Files** checkbox at lower left (Figure 4-6) so that the four files will be merged into a single geWorkbench dataset after they are downloaded. Highlight the desired bioassays.
12. Click **Open** (located in the **Experiment Information** box). The datasets are downloaded, merged, and inserted into the current **Project** (Figure 4-7). In the **Arrays/Phenotypes** component below, the four individual arrays can be seen.



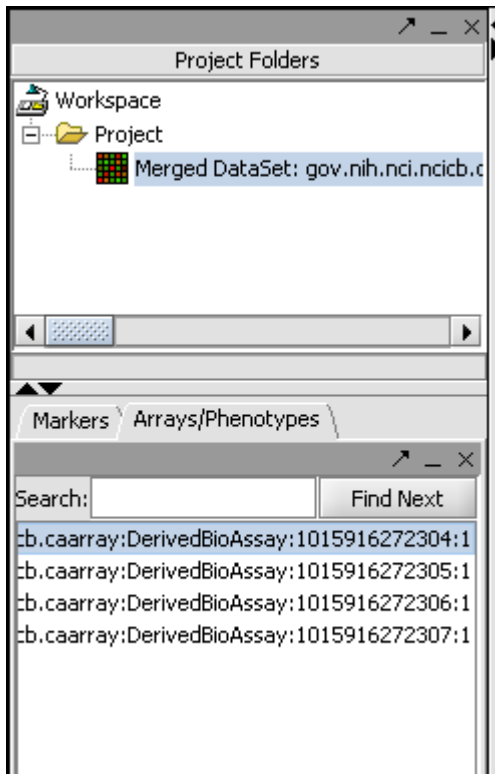


Figure 4-7 The merged dataset retrieved from caARRAY as displayed in the Project Folder

13. The merged dataset can be renamed if desired by right-clicking on it and selecting **Rename**.
14. Did you forget to check the **Merge** checkbox before download? You can merge the files after download by selecting menu item **File > Merge Datasets**.

# 5 Microarray Data Analysis<sup>1</sup>

geWorkbench1.0 provides a comprehensive and extensible suite of open-source desktop software tools that can be applied to the analysis, visualization, and annotation of microarray data. In addition to the analysis and visualization tools routinely found in microarray software tools today, geWorkbench1.0 provides an enhanced environment via its integration with the Cancer Bioinformatics Infrastructure Objects (caBIO)<sup>2</sup>. This integration provides geWorkbench1.0 users with access to publicly available microarray data on a remote NCI server; to the CGAP web site's gene annotation pages, and to the pathway visualization diagrams generated by BioCarta. This last capability allows users to view the observed microarray data in the context of metabolic and signal transduction pathways.

The workbench is intended to support a variety of the input formats in which microarray data are found; its open-ended design supports the extension of the software to accept additional formats as needed. The present version of geWorkbench1.0 supports Affymetrix (.txt, MAS 4.0/5.0), Expression Matrix (.exp) and GenePix (.gpr) files. A simple plug-in framework allows users to further define and use any input format they wish. Similarly, this plug-in framework supports the addition of any number of user-defined filters, normalizers, and analysis algorithms.

This chapter provides an overview of a rather complex software suite, and assumes that the user has some experience with microarray data analysis. The discussion which follows outlines procedures for loading data files, for using visualizations, and for annotating data.

## 5.1 Set Selection (the Markers/Arrays/Phenotypes) components

### 5.1.1 Marker Sets

The term *marker* is used generically to represent several different things in geWorkbench. When working with microarrays, the term marker refers to a gene probe (in other cases, it can be individual items from other data sets, such as sequences). The definition of what constitutes a gene probe in turn depends on the type of microarray platform. On Affymetrix platforms, gene probes are oligonucleotides synthesized on the microarray chip *in situ*. On other platforms (e.g. GenePix), gene probes are oligonucleotides or cloned DNA fragments deposited and immobilized on the substrate by various techniques.

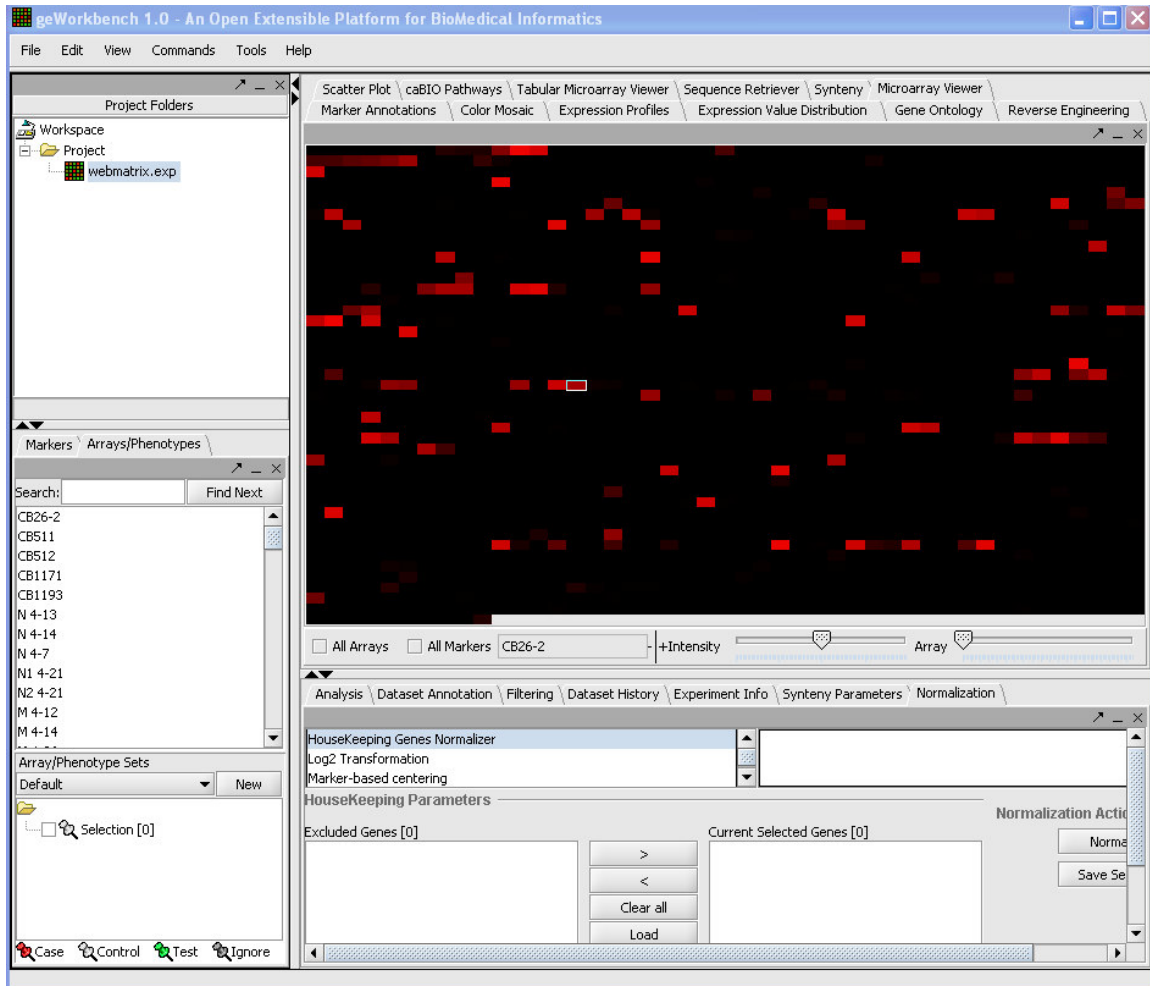
---

<sup>1</sup> This chapter originally derived from the NCICB User Applications Manual, Chapter 5.

<sup>2</sup> The NCICB Technical Guide provides a detailed description of the caBIO project and its application programming interface (API).

As soon as a specific microarray is selected in a project folder, the entire complement of markers on that array is displayed in the *Selection* area under the **Markers** tab. For example, in

Figure 5-1, the user has just created a new project and loaded a single file—*webmatrix.exp* (from the *example data* directory)—to that project.



**Figure 5-1 geWorkbench1.0 Display Immediately After Loading a Data File**

A *marker set* is a user-defined grouping of several markers, e.g. gene probes. Typically, these sets probe for genes of specific interest, of importance for certain disease or developmental states, or for characteristic changes in gene expression that may be hallmarks of a tumor in a particular tissue.

A master list of all gene probes in the currently selected data set is displayed under the **Markers** tab, and a color-coded image of the corresponding gene expression measurements is shown in the View window's **Microarray Viewer**. Individual or groups of gene probes can be selected from this master list and added to smaller marker sets for use in a specific study. The sets are managed in the smaller window immediately below

the master list. A marker set can be saved by right-clicking on it and selecting **Save**. Clicking on the **Load Set** button loads saved sets.

Any number of sets can be created, and they can be grouped as desired. A new group can be created using the **New** button in the Sets area below the master list. A new set can be created manually by selecting markers in the master list and right-clicking, then selecting **Add to Set** from the pop-up menu. A prompt will appear, for the set name. More commonly, new sets of markers will be returned from an analysis step, for example from hierarchical clustering or a t-test. Figure 5-2 shows a set called “cluster\_tree\_84\_markers” containing a set of 84 gene probes., and a second set containing 12 probes.,.



Figure 5-2 Two different gene probes sets in the default group.

### 5.1.2 Set Activation and Manipulation

A gene set, once created, can be *activated*. This notifies other components that this subset of data is available for analysis or visualization as a distinct entity. Any number of sets can be in the activated state at one time. A set is activated simply by checking the box before its name, or from a right-click menu or from the top menu bar Commands

As described in the next section, most visualization tools, including in the **Microarray Viewer** provide **All Arrays and All Markers** checkboxes. By default neither is checked. In this state, if no sets are activated, then all Markers/Arrays are considered active implicitly. If any set has been activated, then only that set will be used, unless the respective “All” button is checked..

A set’s **Activate** option can be used to activate all of the markers or arrays in that set. Similarly, the **Deactivate** option will deactivate all markers or arrays in the set.

As mentioned in the previous section, it is also possible to save and load panel sets independent of the workspace where they were created. While it does not make sense to load a panel set generated from a data set that is not currently loaded, this facility can be useful when several saved workspaces share common data. The **Load Set** option allows users to load panel sets defined outside the current workspace. The user has the option of assigning meaningful names to the sets, using the **Rename** operation.

Finally, individual marker or array sets can also be explicitly renamed, activated, deactivated and deleted from the group. All of the elements listed in the *Marker/Phenotype* windows have pop-up menus associated with them.

### 5.1.3 Array/Phenotype Sets

geWorkbench1.0 uses the term *phenotype* to refer to any user-defined grouping of microarrays. These microarrays will often share some common property that in most cases is phenotypic, although this is not a requirement. For example, one such “phenotype” might represent a disease state such as tumor tissue samples, with a second “phenotype” defined as a collection of experiments performed on normal tissue samples.

Like the **Markers** component, the **Arrays/Phenotypes** component has two portions: the top portion lists the arrays included in the selected data set, and the bottom portion (titled “Array/Phenotype Sets”) lists any user-defined array groupings and sets.

For data sets involving a single array, only that array is present in the top portion. But in the case of multiarray data sets, the display becomes more interesting: each experiment that was included in the set is displayed as a potentially separate phenotype.

Analogous to the procedures for selecting markers or gene probes into gene panels, arrays are selected and grouped—according to the user’s preferences—into phenotype sets. Each array in the top portion of the phenotype window has an associated pop-up menu with options **Add To Set** and **Clear Selection**.

### 5.1.4 The Commands Menu

Many of the commands for manipulating marker and phenotype sets are also available from the **Commands** menu option in the main menu bar, as shown in Figure 5-3.

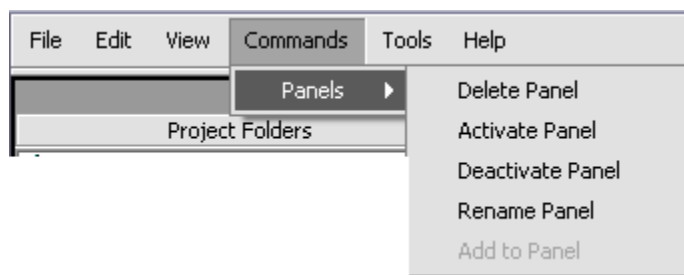


Figure 5-3 The Commands Menu

## 5.2 The View Window

The View window is in a sense the main work area as it provides all of the visualization tools in geWorkbench1.0. Folder tabs running across the top of the screen provide access to these tools, a basic selection of which are summarized in Table 5.2-1 and described in more detail below.

Table 5.2-1 Microarray Visualization Tools in the View Window

<b><u>Visualization Tool</u></b>	<b><u>Description</u></b>
caBIO Pathways	Displays BioCarta pathway diagrams for selected genes.
Color Mosaic	A color mosaic representation of measurements, with each array in one column and each probe in one row.
Dendrogram	Displays tree-structured diagrams (dendrograms) reflecting the results of hierarchical clustering analysis.
Expression Distribution	Value Display a graph of the distribution of expression values for a set of markers from a particular hybridization.
Expression Profiles	Displays the expression of genes across several arrays/hybridizations.
Image Viewer	Displays snapshot images taken from whole screen views.
Marker Annotations	Allows users to retrieve and display CGAP annotations for genes within a marker panel.
Microarray Viewer	Displays expression measurements as spots over a red-green color spectrum (absolute scale) or a red-blue spectrum (relative scale).
Scatter Plot	Allows the plotting of a single microarray chip or marker against other chips or markers in the project. Useful for presenting a visual picture of markers that have changed under different experimental conditions.
SOM Clusters	Displays the results of self-organizing map cluster analysis.

## Visualization Tool

Tabular Microarray Panel

## Description

Presents the numerical values of the expression measurements in a table format; each row represents an individual probe and the columns display the signal and background intensities and intensity ratios

It is important to note that the viewing and analysis tools displayed depend on the type of data currently selected in the Project Folders area.. In addition, the windows vary greatly in information content and small displays can sometimes prohibit a complete view of data for the more complex windows. In order to maximize the display of information it is often useful to detach the display window, by clicking on the arrow facing up and to the right on the view panel. See the image below that shows a detached promoter window. The arrow in the top right now points to the bottom right, indicating the window is detached.

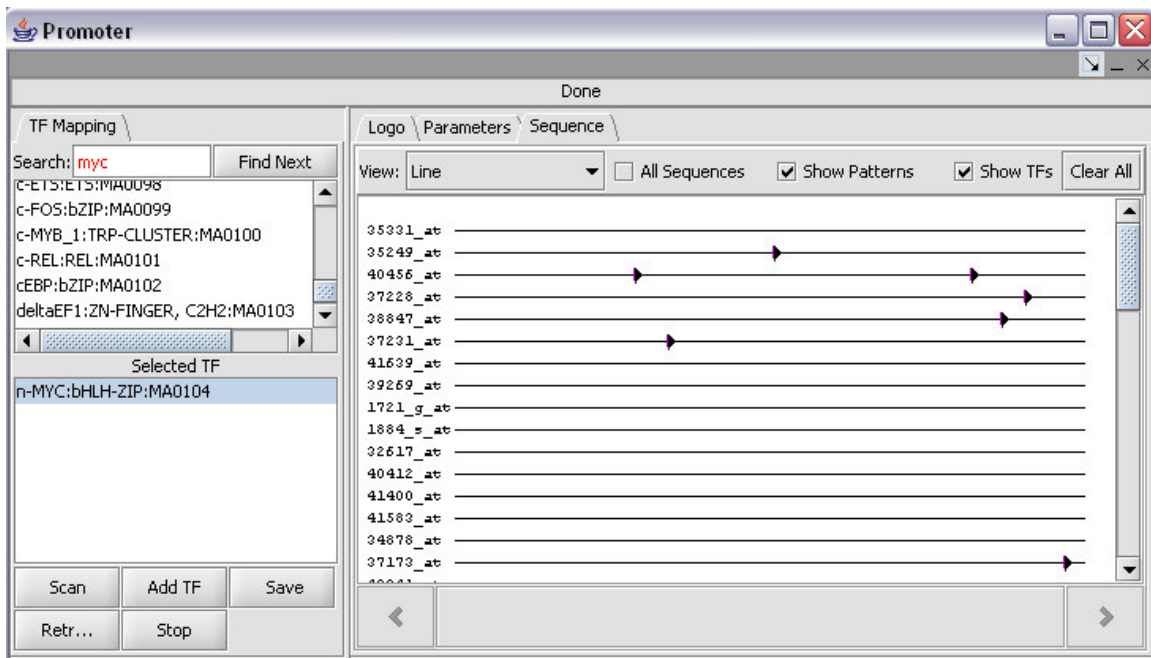


Figure 5-4 A detached window.

Some of the visualization tools are only applicable to data sets involving more than one array; others are enhanced by applying filters and/or normalizations to the data, and two of the tools are only applicable to clustered data—the Dendrogram and SOM Clusters tools. This section provides a quick tour of the general capabilities of those tools that can be applied to unclustered data.

### 5.2.1 The Microarray Viewer

The **Microarray Viewer** is the default visualization tool in the View window, and is displayed when the application is first started. As each new microarray data file is opened, that data set becomes the currently selected one, and the data is displayed in the **Microarray Viewer**. The image displays color-coded levels of gene expression, using the absolute color scale these vary gradually from red (positive values) through black (zero) to green (negative values). The interpretation of course depends on the specifics of the data loaded. The density of the data points in this screen is determined by the number of probes on the array. The Microarray Viewer has four controls at the bottom of the panel, shown in Figure 5-5.



Figure 5-5 Graphical Controls in the Microarray Panel

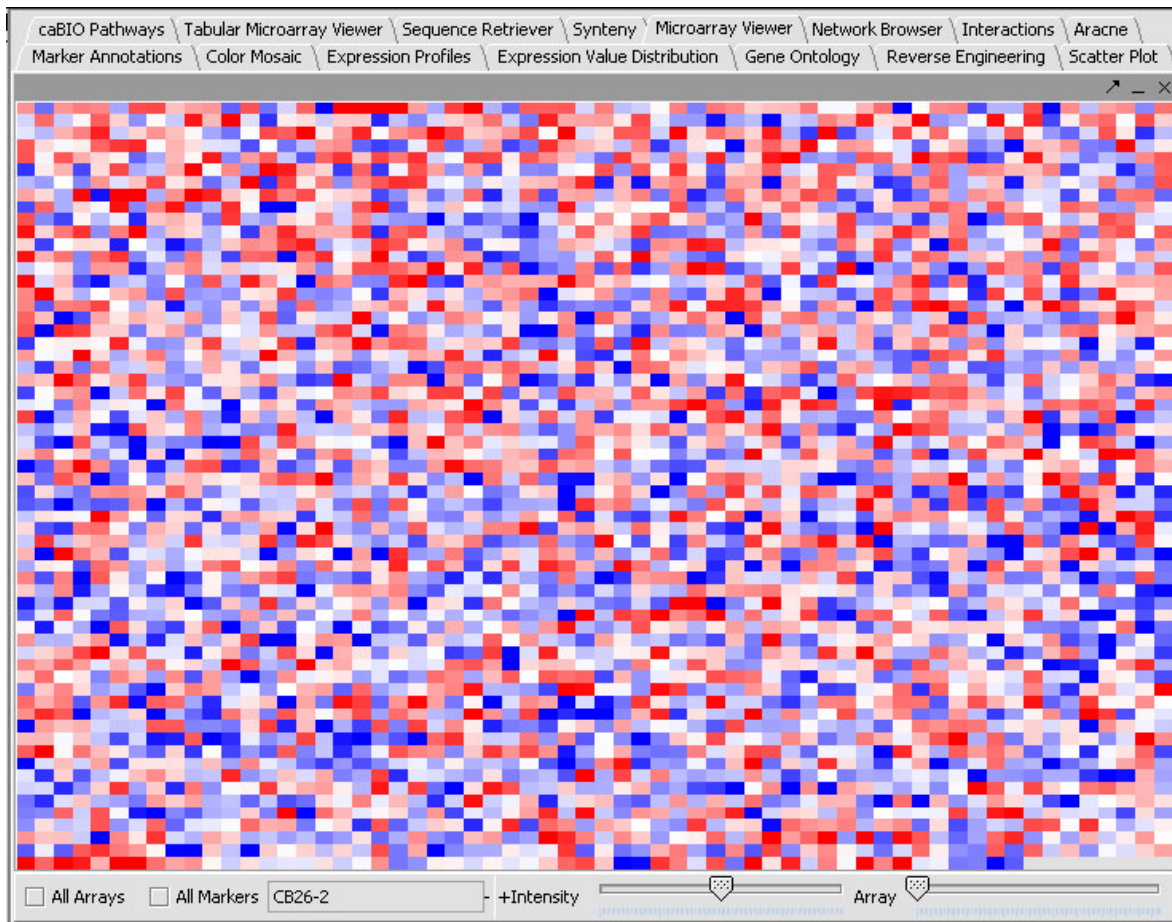
The two checkboxes to the left, **All Arrays** and **All Markers**, determine which data points are included in the display. If neither is checked, then the entire data set is shown. The **All Arrays** control is useful when working with data sets comprising multiple arrays. In this case, only those arrays that are included in a currently *activated* phenotype set will be displayed (see previous section). Similarly, de-selecting the **All Markers** checkbox can be used to view only those probes that are currently included in an activated marker set.

The scrollbar to the right of the two checkboxes is active only when a multi-array data set is being viewed. In this case the individual microarrays are displayed from left to right, and the scroll bar can be used to jump from one microarray to the next. The entry point to each of the individual chip displays is indicated by a tick mark on the scrollbar.

Right-clicking on the panel will provide the following menu items:

1. **Show Marker**: highlights a particular marker in the display for reference purposes.
2. **Remove Marker**: unselects markers selected by **Show Marker**.
3. **Image Snapshot**: save an image under the dataset node in the **Project Tree**.





**Figure 5-6** Display of a single filtered Affymetrix array with about 3,000 features in the Microarray viewer, using the relative display preference.

The **Microarray Viewer** provides an overview of the chip(s) under investigation and can be used for ascertaining the quality of the data—i.e., the uniformity of the hybridizations, the compatibility of intensities between chips, and so forth. Each feature on the chip can be accessed with a small cursor box and highlighted. Left-clicking the mouse will then highlight the corresponding probe in the master list contained in the Gene Panel window. This association between the **Microarray Viewer** and the **Markers** component facilitates the selection of individual probes for inclusion on explicit marker sets, as the user can then right click the selected probe and simply select **Add to Set**.

Only those gene sets that are currently activated will be displayed when the **All Markers** checkbox is unchecked. Any number of smaller predefined marker sets can be activated and displayed in this zoomed view.

Phenotype sets are used to create experiment groups. For instance, in a multi-array data set containing arrays from both normal and tumor tissue, these samples can be divided into appropriate sets, which can then be used for example in setting up statistical tests. Like their marker set counterparts, phenotype sets can be selectively activated and displayed using the controls described above.

## 5.2.2 The Expression Profiles Tool

The **Expression Profile** view makes it possible to visualize changes in the gene expression levels across different hybridizations. This is useful especially in the analysis of time course or dose response experiments. Since the tool generates a graphical representation of *relative* expression levels across two or more arrays, the **Expression Profile** view can only be applied to datasets containing multiple arrays.

After loading, , merging and normalizing the desired data sets, the user may also wish to apply marker and phenotype panels in order to zoom in on the expression behavior of a subset of genes and/or hybridizations.

Figure 5-7 shows a sample expression profile,” A set of 84 genes showing similar expression was returned following hierarchical clustering analysis and activated.

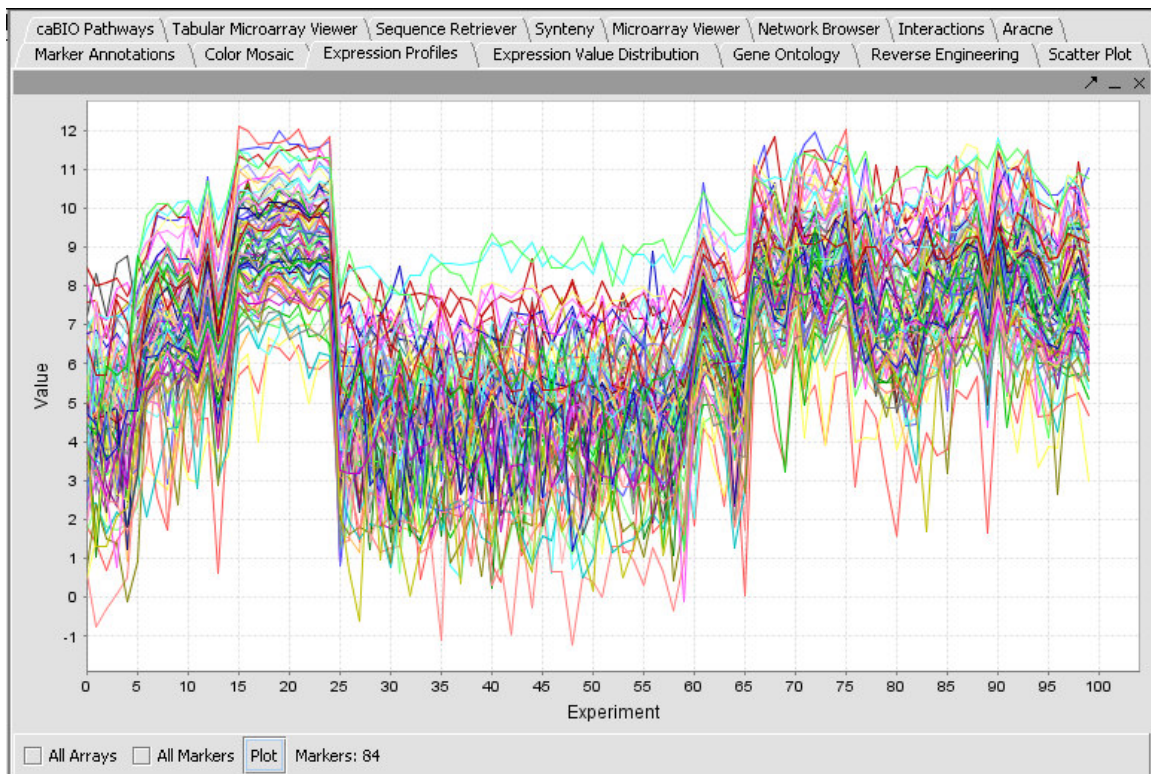


Figure 5-7 An Expression Profile Using Phenotype and Marker Panels

## 5.2.3 The Color Mosaic View

When a multiple array data set is viewed in the Color Mosaic view, the gene expression levels across all of the microarrays are displayed as a color coded image, where each

column in the image corresponds to one of the microarrays, and each row corresponds to a particular gene probe.

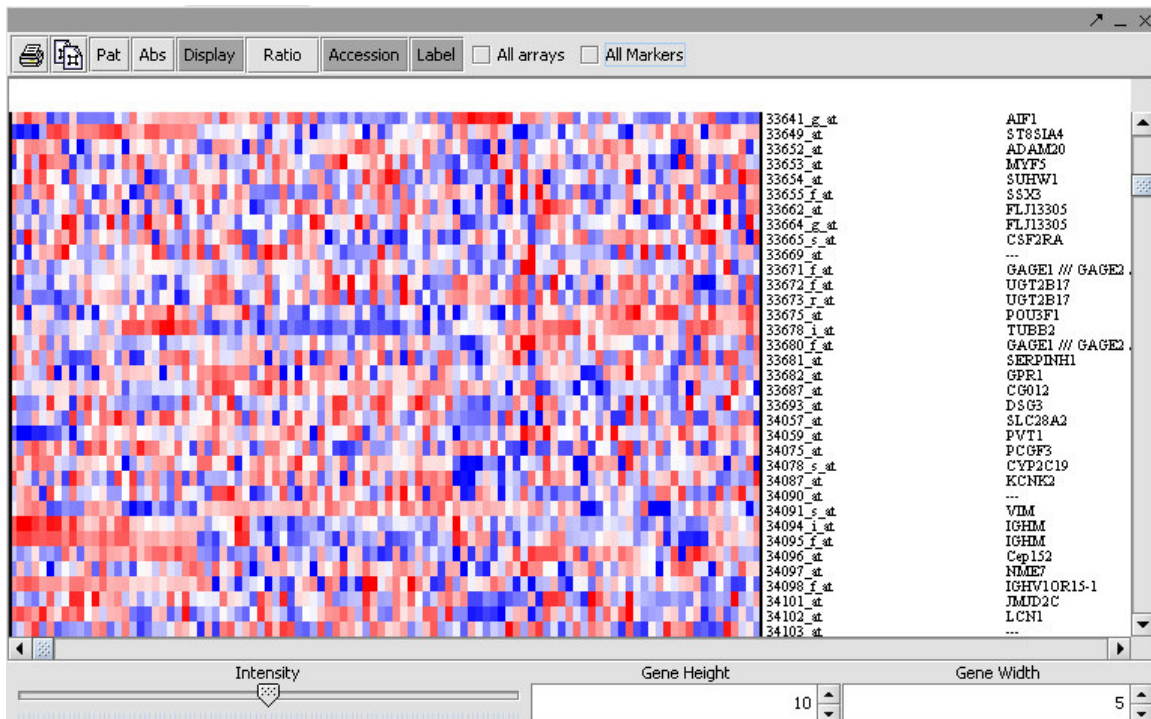
Figure 5-8 shows a color mosaic generated for a single microarray data file (the mosaic is displayed only when the “Display” button is activated)..The graphical controls provided in the Color Mosaic window can be seen in

Figure 5-8 and in Figure 5-9, and include **the following** checkboxes **and** buttons:

**Table 5.2-2**

<u>Button/ Checkbox</u>	<u>Description</u>
Display	Display selected data.
All Arrays	Will display all arrays even if array sets are activated.
All Markers	Will display all markers (probes) even if marker sets are activated.
Abs	(not used)
Accession	Displays accession Numbers
Ratio	(not used)
Label	Displays gene names
Pat	(not used)

A mouse over tool tip controls for changing the height and width of the displayed genes, and a slider for modifying the intensity of the color codings.



**Figure 5-8 A Color Mosaic View of a Single Data Set**

The height and width controls increase or decrease the dimensions of the tiles as well as the associated labels when these are displayed. The slider increases or decreases the thresholds used to define the color codings. Clicking on a tile in the mosaic will highlight the corresponding gene probe in the Marker Panel window, where it can be picked and added to a marker panel if desired.



**Figure 5-9 Graphical controls for the Color Mosaic View**

### 5.2.4 The Tabular View

Like the **Color Mosaic**, the **Tabular View** can be used to obtain a side-by-side comparison of the observed intensities for each gene probe over multiple chips. The display in this case however, shows the numerical values in a simple table format.

As with most of the other panels in the View window, checkboxes are provided for displaying selected phenotype and marker sets.

Figure 5-10 shows a **Tabular View** generated for the *webmatrix.exp* data set included with the distribution in the *example data* directory.

Marker	CB26-2	CB511	CB512	CB1171	CB1193	N 4-13	N 4-14	N 4-7	N1
AFFX-MurIL...	4.15	4.81	1.98	4.42	4.13	2.93	2.18	0.8	2
AFFX-MurF...	2.69	3.99	6.03	4.96	0.77	4.76	5.08	1.91	5
AFFX-BioB...	8.4	6.64	7.28	7.94	7.32	7.97	6.67	6.29	7
AFFX-BioB...	9.62	7.05	7.59	8.43	8.07	8.3	6.94	6.66	7
AFFX-BioB...	8.64	5.39	4.64	7.47	6.85	7.91	6.55	5.67	5
AFFX-BioB...	4.21	5.86	6.71	5.82	4.38	6.53	6.16	5.58	5
AFFX-BioDn...	7.5	6.6	5.01	7.98	7.1	6.5	5.48	6.35	4
AFFX-CreX...	7.17	6.92	6.23	6.65	7.01	7.01	4.77	4.8	6
AFFX-CreX...	6.81	7.42	6.53	7.48	8.07	7.94	6.25	6.27	3
AFFX-hum...	12.06	10.87	10.92	8.23	8.4	12.31	10.42	11.11	10
AFFX-LysX...	4.28	4.44	2.3	3.37	1.33	1.57	4.5	4.2	4
AFFX-LysX...	2.47	2.93	3.4	2.92	5.05	4.62	3.61	4.53	2
AFFX-LysX...	4.01	3.92	4.13	1.48	3.49	-0.02	0.05	3.5	2
AFFX-ThrX...	1.62	4.4	2.4	1.77	4.22	3.67	3.67	5.35	4
AFFX-TrpnX...	3.47	5.3	5.52	3.15	3.86	5.21	4.8	4.5	3
AFFX-TrpnX...	3.55	2.05	1.78	0.09	-0.27	1.92	3.1	4.87	2
AFFX-HUIML...	4.59	7.48	7.58	6.61	7.03	9.59	8.57	7.74	8

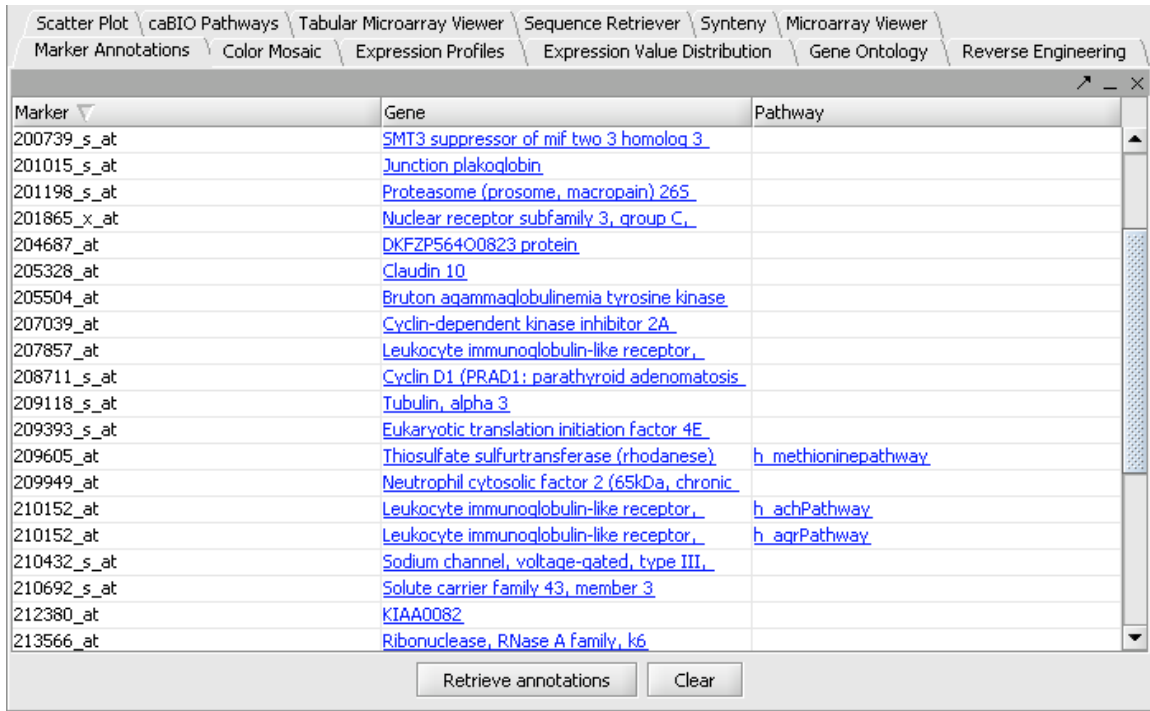
**Figure 5-10 The Tabular View**

Note – values filtered out by applying one of the masks described in Section 5.3 will be highlighted in yellow.



## 5.2.5 The Marker Annotations and caBIO Pathways Views

The **Marker Annotations View** window is used to view and retrieve CGAP annotations for selected genes. A collection of annotations for a set of genes can be viewed when those genes are included in a marker set. The annotations can be retrieved by activating the marker set, and pushing **Retrieve annotations..**



The screenshot shows a software window titled "Marker Annotations View" with a menu bar containing: Scatter Plot, caBIO Pathways, Tabular Microarray Viewer, Sequence Retriever, Synteny, Microarray Viewer, Marker Annotations, Color Mosaic, Expression Profiles, Expression Value Distribution, Gene Ontology, and Reverse Engineering. The main area contains a table with three columns: Marker, Gene, and Pathway. The table lists 20 marker genes with their corresponding gene names and associated pathways. At the bottom of the window are two buttons: "Retrieve annotations" and "Clear".

Marker	Gene	Pathway
200739_s_at	<a href="#">SMT3 suppressor of mif two 3 homolog 3</a>	
201015_s_at	<a href="#">Junction plakoglobin</a>	
201198_s_at	<a href="#">Proteasome (prosome, macropain) 26S</a>	
201865_x_at	<a href="#">Nuclear receptor subfamily 3, group C,</a>	
204687_at	<a href="#">DKFZP564O0823 protein</a>	
205328_at	<a href="#">Claudin 10</a>	
205504_at	<a href="#">Bruton agammaglobulinemia tyrosine kinase</a>	
207039_at	<a href="#">Cyclin-dependent kinase inhibitor 2A</a>	
207857_at	<a href="#">Leukocyte immunoglobulin-like receptor,</a>	
208711_s_at	<a href="#">Cyclin D1 (PRAD1; parathyroid adenomatosis</a>	
209118_s_at	<a href="#">Tubulin, alpha 3</a>	
209393_s_at	<a href="#">Eukaryotic translation initiation factor 4E</a>	
209605_at	<a href="#">Thiosulfate sulfurtransferase (rhodanese)</a>	<a href="#">h_methioninepathway</a>
209949_at	<a href="#">Neutrophil cytosolic factor 2 (65kDa, chronic</a>	
210152_at	<a href="#">Leukocyte immunoglobulin-like receptor,</a>	<a href="#">h_achPathway</a>
210152_at	<a href="#">Leukocyte immunoglobulin-like receptor,</a>	<a href="#">h_agrPathway</a>
210432_s_at	<a href="#">Sodium channel, voltage-gated, type III,</a>	
210692_s_at	<a href="#">Solute carrier family 43, member 3</a>	
212380_at	<a href="#">KIAA0082</a>	
213566_at	<a href="#">Ribonuclease, RNase A family, k6</a>	

Figure 5-11 The Marker Annotations View

Figure 5-11 shows a part of the **Marker Annotations View** for a selected set. The first column lists the marker gene, the second column lists the name of the gene, and the last column lists any pathways associated with that gene. Clicking on the gene name will open a new browser window displaying the CGAP annotation page for that gene. Clicking on a pathway will make the **caBIO Pathways** window active in the View window, and display the corresponding BioCarta pathway. Figure 5-12 shows one such pathway diagram. All the data can be sorted by clicking on the column header at the top of each column. The “Clear” functionality will clear all annotations, not just the selected annotation.

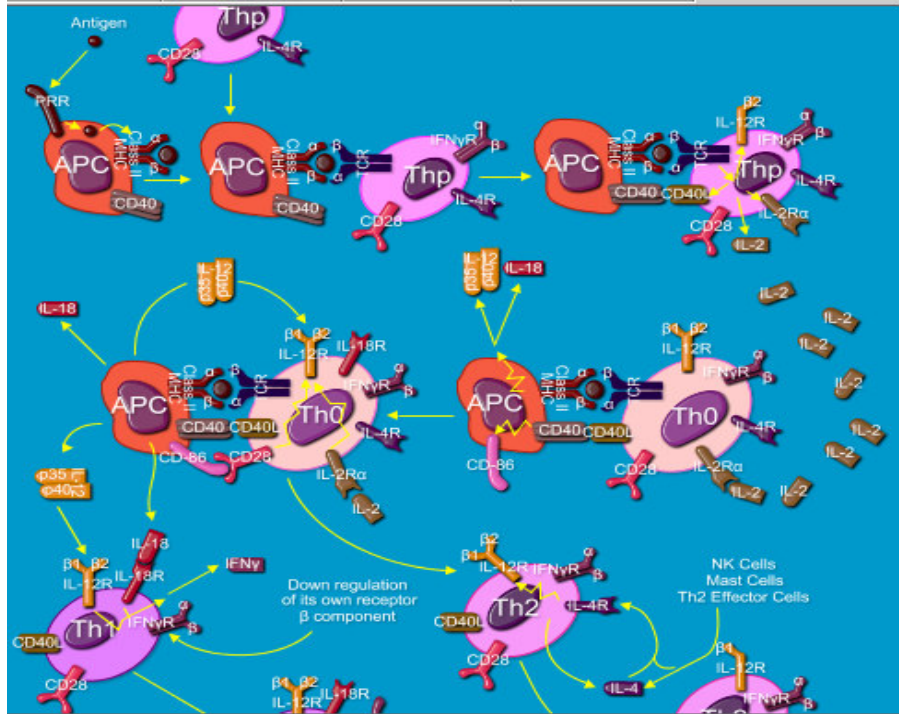


Figure 5-12 The caBIO Pathways View

### 5.2.6 The Image Viewer

Several of the visualization tools provide a means of capturing a snapshot of a selected region of the display. For example, right clicking on any point in the **Microarray Viewer** or the **Dendrogram** component will cause a pop-up text control to appear, with one of the options being **Image Snapshot**. Left-clicking on this control will create a snapshot of whatever is currently visible in the display view, and store that image under the

associated data file in the Project Folders component.

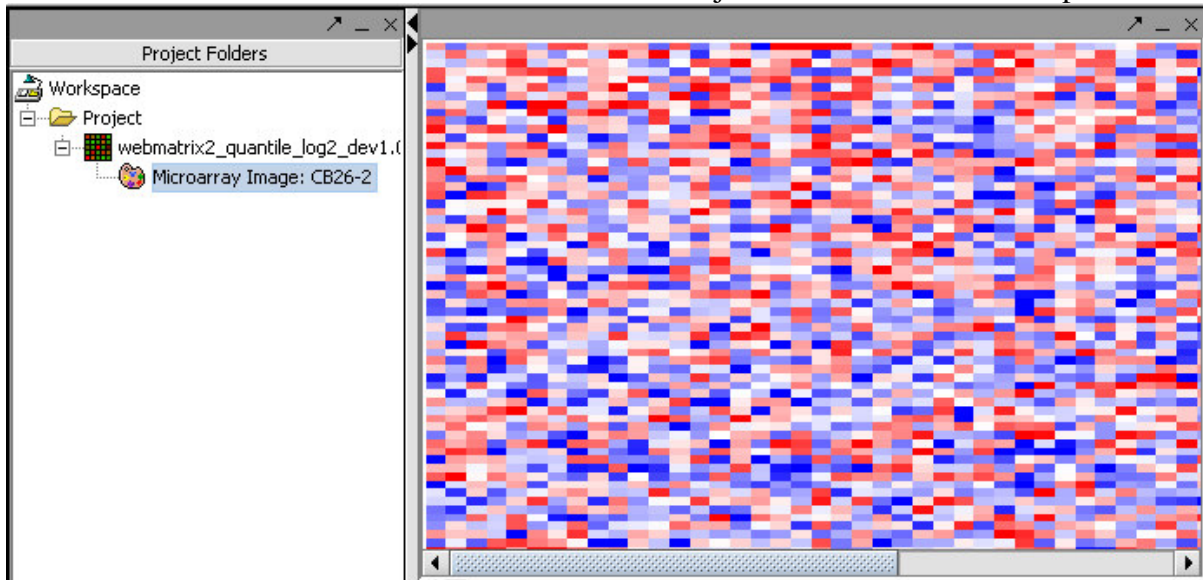
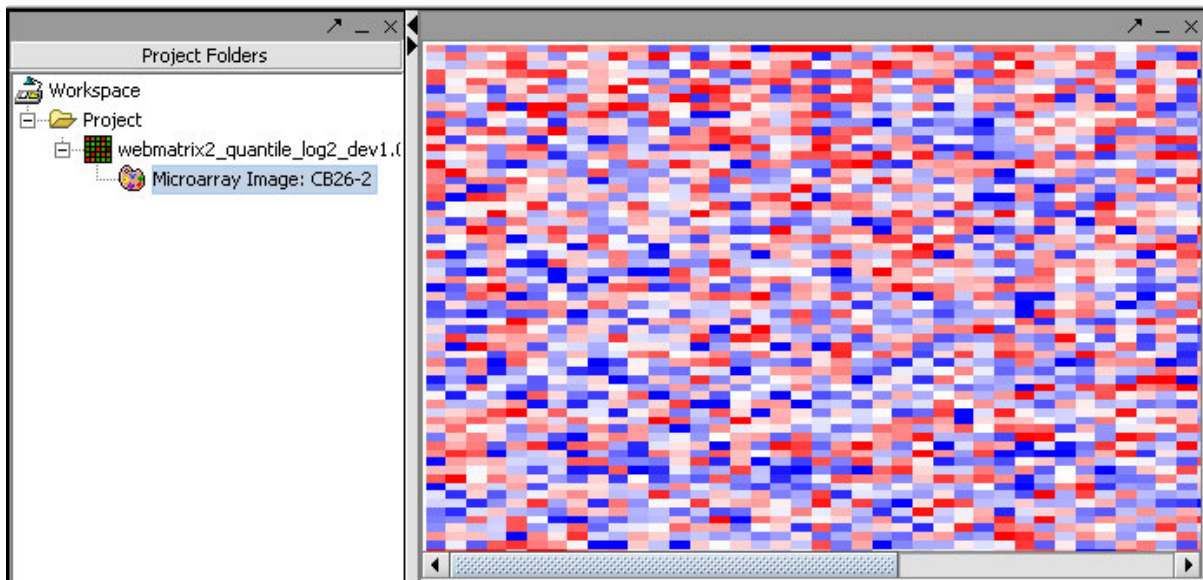


Figure 5-13 is a snapshot captured from the **Microarray Viewer**. The snapshot is stored in the **Project Folders** under the dataset from which it originated. An image can be saved in BITMAP, TIFF, JPEG, or PNG format by selecting the image in the Project window, and selecting the **Export** option from the drop-down file menu.- Note that when the image file is selected, the only viewer type available is the Image Viewer – no tabs are present.



**Figure 5-13 The Image Viewer**

The steps used to capture an image and to view and save the stored snapshot are:

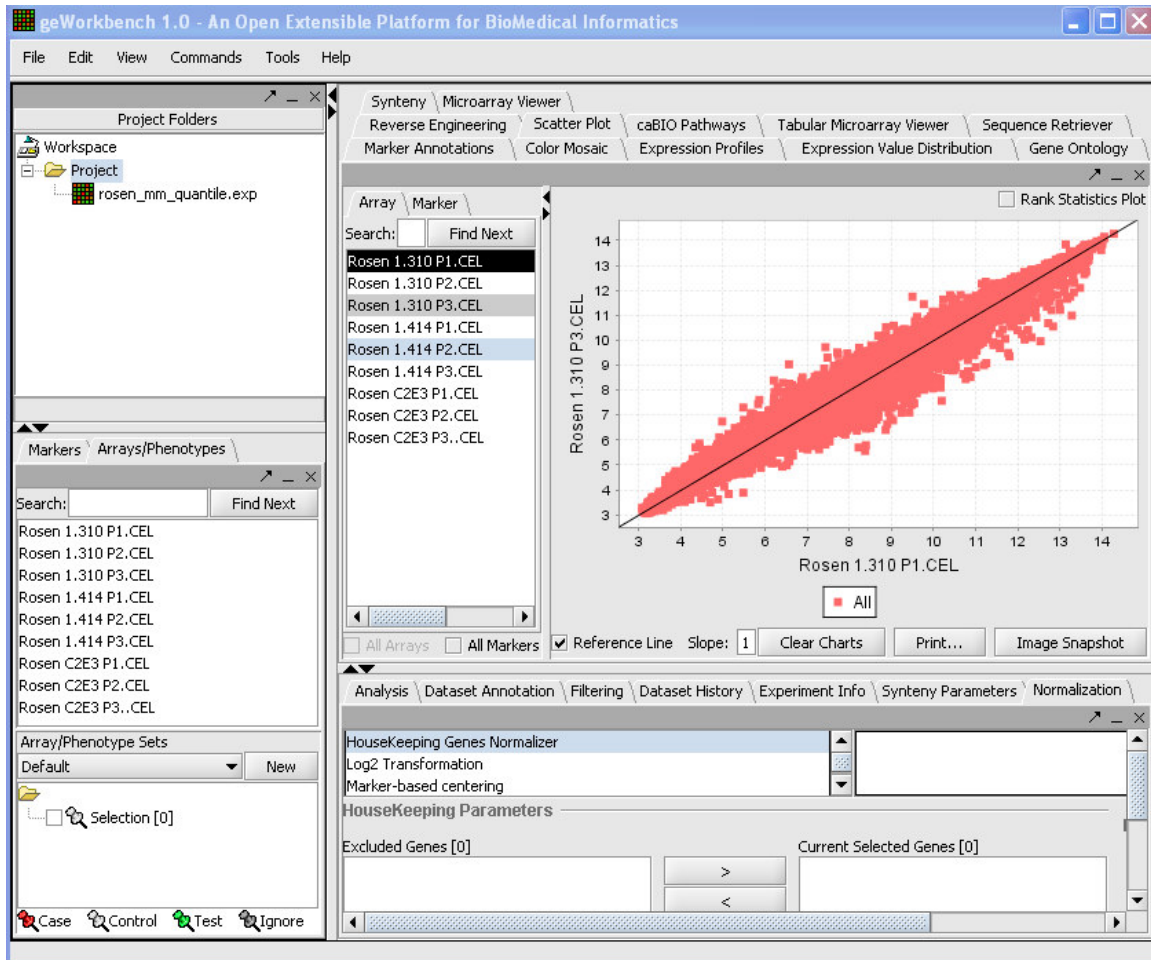
1. In the **Project** window, select the data set for which you would like to capture an image.

2. In the View window, select a visualization tool.
3. Right click on any point in the tool's display and left click on **Image Snapshot**.
4. In the **Project** window, expand the associated data set (if is not already open) by clicking on the +-like icon to the left of the data set.
5. Click on the stored image to bring it up in the **Image Viewer**.
6. Select the image in the **Project** window, and use the **Export** option from the file menu to save it in BITMAP, TIFF, JPEG, or PNG format.

### 5.2.7 Scatter Plot

The scatter plot feature of geWorkbench1.0 is useful for a visual comparison of up to 6 pairs of markers or arrays. The user selects an initial marker or array which is plotted on the x-axis of the Scatter plot graph and highlighted in black on the arrays/phenotype or marker selection panel. The user can then select a second marker or array which is plotted on the y-axis and highlighted in grey on the arrays/phenotype or marker selection panel. If markers are selected as the axes, then each point represents an array. If arrays are chosen for the axes, then each point represents a marker. Additional markers or arrays that are selected are also highlighted in grey, and are used as the y-axis for additional graphs. A light blue color is shown in the panel when a marker or array is deselected from the list.





**Figure 5-14 Scatter Plot**

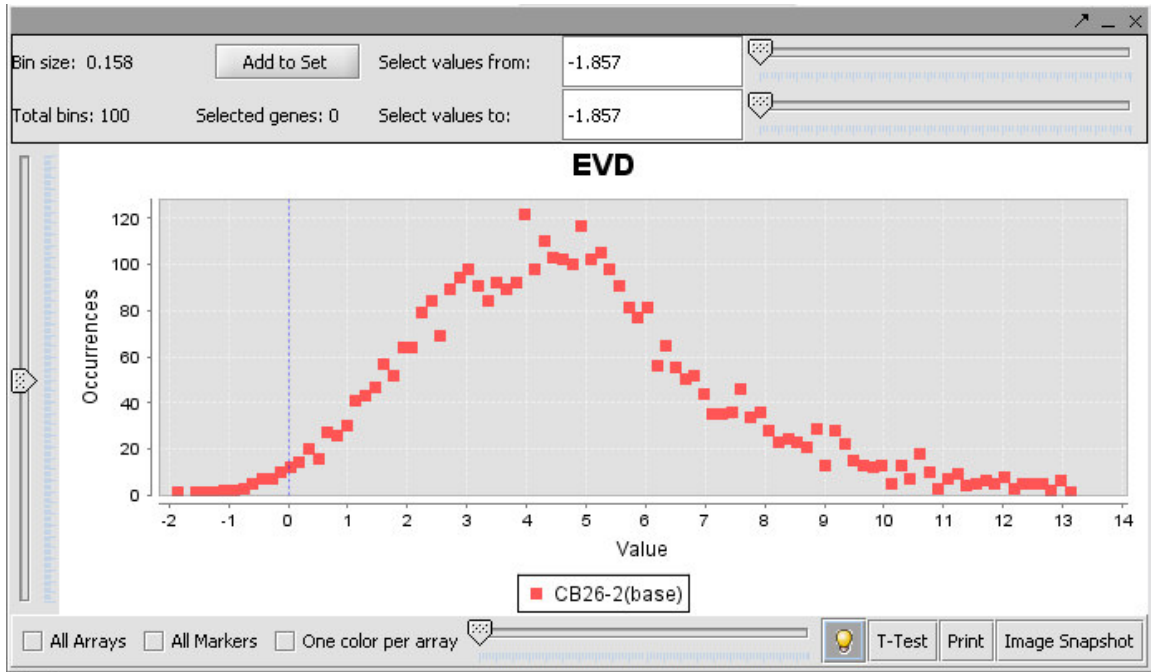
The picture above shows the ScatterPlot in geWorkbench1.0. In general one would use either the Array or Marker tab in this component along with the search functionality in order to find the desired marker or arrays to be plotted. Then it is simply a matter of picking the x-axis marker or array, and then selecting the desired marker or array for the y-axis to plot it against. As usual, the the “All Markers” or “All Arrays” checkboxes can be used to override any activated sets of Markers or Arrays..

The checkbox for “Rank Statistics Plot” transform the data from being plotted by expression value to being plotted by their rank. The “Clear Charts” button removes all charts, the “Print” button allows for printing, and the “ImageSnapshot” button takes a snapshot of the Scatterplot. The “Reference Line” checkbox simply adds or removes the reference line from the Scatterplot.

## 5.2.8 Expression Value Distribution

The expression value distribution component plots the expression values of microarray markers for the selected hybridized array or arrays. Both can be specified in the selection panel in the markers and arrays/phenotypes tabs respectively and should immediately be

plotted on the graph. The expression value distribution is useful for determining the difference in expression levels between sets of markers under difference conditions, a T-Test can be used to detect markers with significantly different expression. The controls for the expression value distribution are shown below.



**Figure 5-15 Expression Value Distribution**

### Controls:

**Bin size and Total Bins:** Each expression range on the EVD corresponds to a “bin”. The size of each bin and total bins are displayed at the upper left corner of the component. The number of total bins equals  $(m - M) / \text{number of bin size}$ .



**Tooltip:** When a user scrolls over a plot, the enabled tool tip displays the name of the array. The tooltip display can be disabled by selecting the tooltip icon. The disabled tooltip icon appears grayed out.

**Change the base array:** The slider can be used to navigate through the microarrays in the selected dataset. As the user moves the slider to the next tick which represents a microarray, the grid is updated with positions representing the marker values in the Microarray being displayed. The base array always labels with the red color.

**All Markers and All Arrays Checkbox:** These checkboxes override any activated sets of Markers or Arrays, causing all to display.

**One Color per Array:** The EVD lines and plots appear color-coded, based on the color preferences of the arrays panel. All the arrays in a set are shown with the same color. The user can modify the color display to reflect a unique color per array by selected the checkbox **One Color per Array**.

**Legend:** A legend appears at the bottom of the plot indicating which color and shape corresponds to which panel. Modification of image display preferences are described in details in Preferences. The base array always lists as the first one.

**Image Snapshot:** Clicking on this button creates image snapshots that are saved in the project.

**Zoom:** Zooming in can be done by left or right- clicking and dragging down and to the right over a region of the image. Zooming out is done similarly but dragging up and to the left..

**T-Test:** This button computes a t-test statistic for each marker and updates the graph title from EVD to T-Test and disables the base array slider. Case and control panels must be created and classified (see Marker Sets section for additional information).

**Print:** Prints the displayed EVD. The print dialog pop up is displayed to support printer selection.

**Selected Values from/to:** The starting/ending X axis locations that allow markers to be selected graphically and filtered. The number of current selected genes is displayed next to the Total bins. “Add to Set” button adds the selected genes to the Markers component.

### 5.2.9 Reverse Engineering with Cytoscape

The Reverse Engineering component uses the information theory concept of mutual information to infer interactions between genes and gene products from microarray expression data. It calculates the information that the expression pattern of one gene carries about the expression of another gene, that is, it is a pairwise calculation. From a single “hub gene” a network can be reversed engineered and then viewed in the Cytoscape plugin. A set by step procedure is shown below on how to do this if available on the wiki at:

[http://wiki.c2b2.columbia.edu/workbench/index.php/Tutorial\\_-\\_Reverse\\_Engineering](http://wiki.c2b2.columbia.edu/workbench/index.php/Tutorial_-_Reverse_Engineering)

The wiki also includes instruction on how to use the Cytoscape plugin in the context of reverse engineering. Additional details on how to operate the cytoscape plugin can be found at the cytoscape website (<http://www.cytoscape.org/>).

## ***The Analysis/Annotation Window***

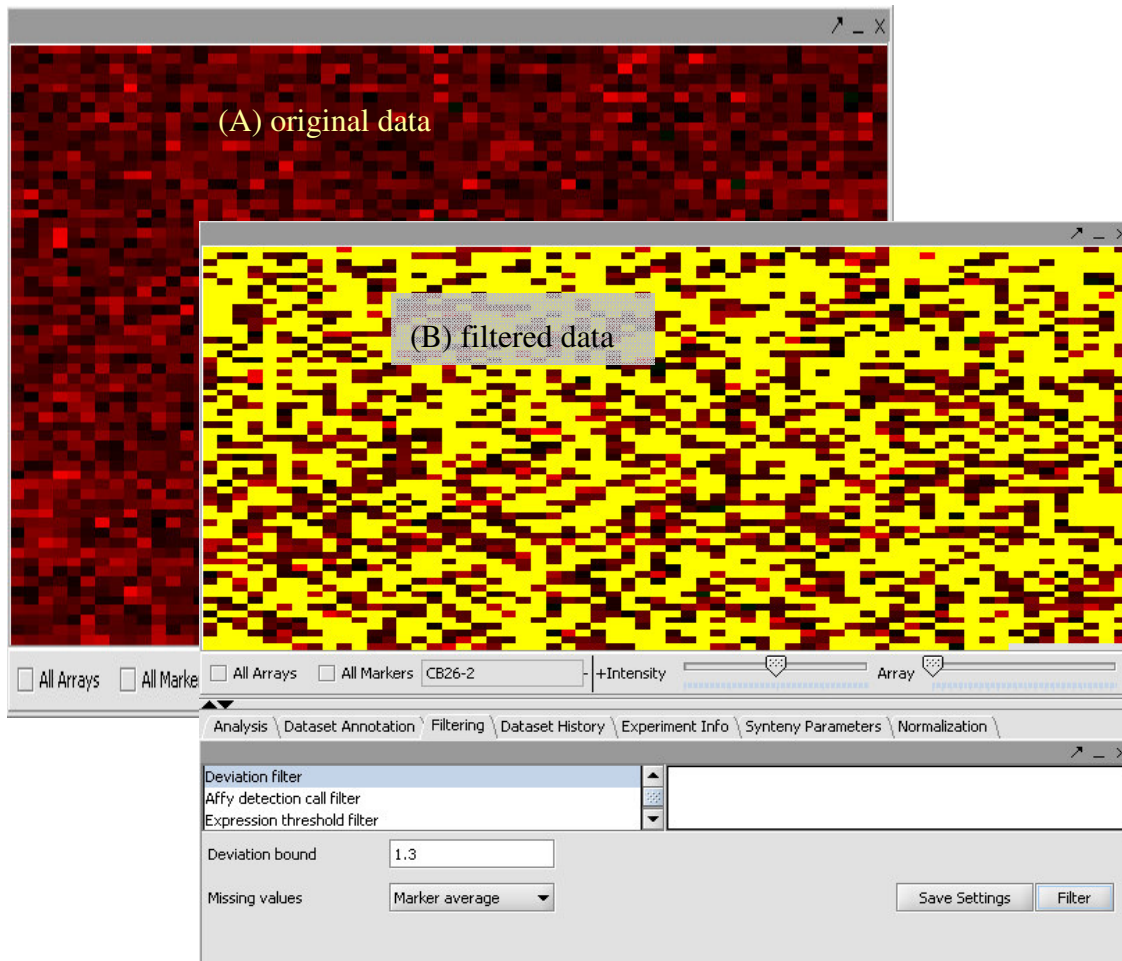
The tab-indexed tools in this last component area include facilities for filtering, normalizing and analyzing data, along with components for viewing the history of operations that have been performed on a data set and general experiment and annotation information. All of the filtering, normalization, and analysis tools include a **Save Settings** option, which saves the parameters used in the analysis or processing step with the workspace.

Filters are used to remove data points when some data quality or signal criteria are not met. As a result of applying a filter, the status call of a questionable data point may be reset to "missing," or alternatively, the data point may be eliminated altogether from the data set. In the later case, all measurements for that marker (across all chips in the data set being filtered) will be eliminated. In contrast, normalizers do not change the status or remove individual markers, but re-scale the observed intensities, usually in preparation for some type of analysis. Filtering or normalizing is done directly on a dataset in geWorkbench, and does not create a new copy of the data. Copies of data in a particular state can be created manually by saving a dataset to a file.

### ***5.3 Filtering Operations***

The Filtering Panel contains several filters that allow the software to set certain values to missing. For instance, the *Affy detection call* filter allows the user to filter out undesirable values on the basis of the Affymetrix calls ("present," "absent" or "missing").

Figure 5-16 shows the result of applying this filter to remove from analysis all markers with a call of "absent". In the **Microarray Viewer** display of the filtered data, all of the missing data points now appear as yellow squares in the heat map. Table 5.3-1 summarizes the filters that are available from a pull-down list in the **Filtering** component. *Please note that all filtering and normalization functions change the original dataset and do not keep it intact.*



**Figure 5-16 Microarray Viewer showing data before (A) and after (B) filtering.**

**Table 5.3-1 The Filtering Panel Toolset**

<b><u>Filter</u></b>	<b><u>Description</u></b>
Missing values	Discards all markers that have “missing” measurements in at least <i>n</i> microarrays, where <i>n</i> is defined by the user. Another filter must first be applied however, in order to generate the missing values upon which this filter can operate.
Deviation	Sets as missing all markers whose measurements deviate below a given value across all microarrays.
Expression threshold	Sets as missing all markers whose measurements are inside (or outside) a user-defined range.
Affy detection call	Applicable to Affymetrix data only. Sets all measurements whose detection status is any user-defined combination of A, P or M as

	missing (Absent, Present, Marginal).
2-channel threshold	Applicable to 2-channel arrays (Genepix) data only. Defines applicable ranges for each channel, and sets all values for which either channel intensity is inside (or outside) the defined range as missing.
Genepix Flags	Removes values flagged in Genepix file

### 5.3.1 Normalization Tools

Before comparing multiple microarrays, the user must first ensure that the observed values therein have been made “comparable” through a process of normalization. The normalization panel offers the user several gene-centric or array (tissue)-centric methods that are summarized in Table 5.3-2.

**Table 5.3-2 The Normalization Panel Toolset**

<b><u>Normalization Tool</u></b>	<b><u>Description</u></b>
Missing value calculation	Replaces every missing value with either the mean value of that marker across all microarrays or with the mean measurement of all markers in the microarray where the missing value is observed.
Log2 Transformation	Applies a log2 transformation to all measurements in a microarray.
Threshold Normalizer	All data points whose value is less than (or greater than) a user-specified minimum (maximum) value are raised (reduced) to that minimum (maximum) value
Marker-based centering	Subtracts the mean (median) measurement of a marker profile from every measurement in the profile
Array-based centering	Subtracts the mean (median) measurement of a microarray from every measurement in that microarray.
Mean-variance normalizer	For every marker profile, the mean measurement of the entire profile is subtracted from each measurement in the profile and the resulting value is divided by the standard deviation.
Housekeeping Normalizer	Genes Uses a set of house-keeping genes to normalize expression of all genes on all arrays such that the averaged expression value of house-keeping genes is constant across all microarrays.

## Normalization Tool

## Description

Quantile Normalization

Assumes distribution of probe intensities is nearly the same in all samples and calculates the quantile of each value and normalizes it against a reference chip.

### 5.3.2 Dataset History

geWorkbench1.0 provides a convenient system for electronic tracking of all actions. As noted, each time a new file or image is generated, that file appears in the **Project Tree Window** as a new node occurring beneath the data set from which it was derived. In addition, the **Dataset History** window displays a list of all of the operations that were performed on both the currently selected data set as well as on all of its “parent” data sets in the **Project Tree**. Figure 5-17 shows the history window for a data normalization workflow.

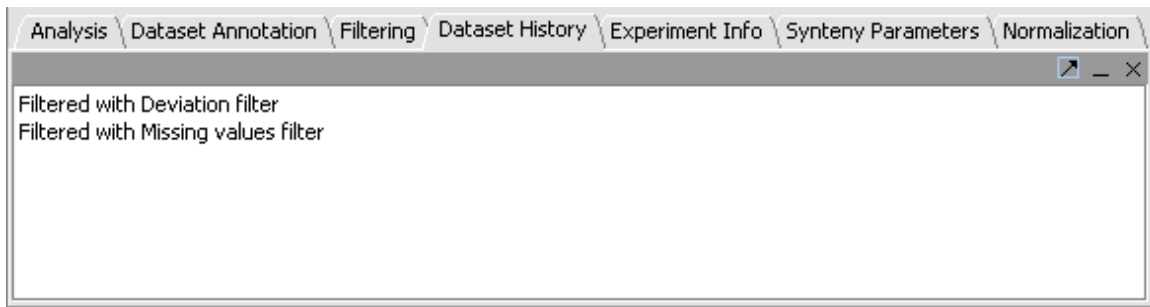
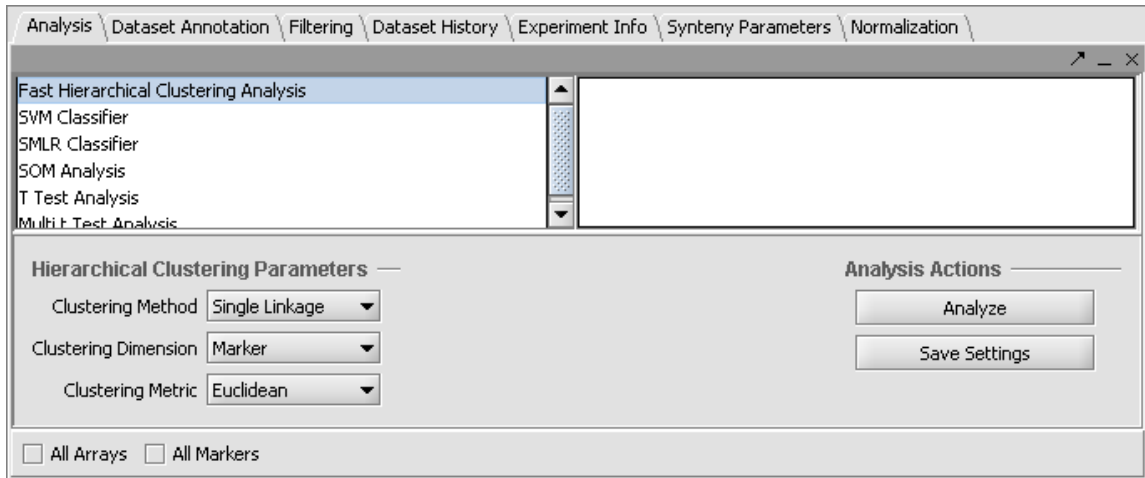


Figure 5-17 The Dataset History Window

## 5.4 The Analysis Tools

The Analysis area (Figure 5-18) contains access to a number of clustering and statistical analysis tools.. Analysis parameters can be saved to the file system. Analysis results are displayed in the separate viewing region of the application, and datasets are placed in the Project Folders area beneath their parent dataset.. Some of the algorithm implementations are adapted from TIGR's MEV<sup>5</sup> software.



**Figure 5-18** The analysis component.

### 5.4.1 Hierarchical Clustering:

geWorkbench1.0 provides two clustering methods in the **Analysis** panel, Hierarchical clustering and SOM (Self-Organizing Maps). Hierarchical clustering groups markers on the basis of similarities in their expression profiles, and outputs a hierarchical tree that can be viewed in the **Dendrogram** component (see Figure 5-19). Clustering can also be done in the array dimension if desired.

The parameters used for this analysis method are Clustering Method, Clustering Dimension and Clustering Metric. The clustering dimension determines whether clustering should be done against the markers, arrays, or both. The remaining two determine how the clustering is performed and are described further in the following:

The clustering methods provided are:

**Table 5.4-1 Hierarchical clustering methods**

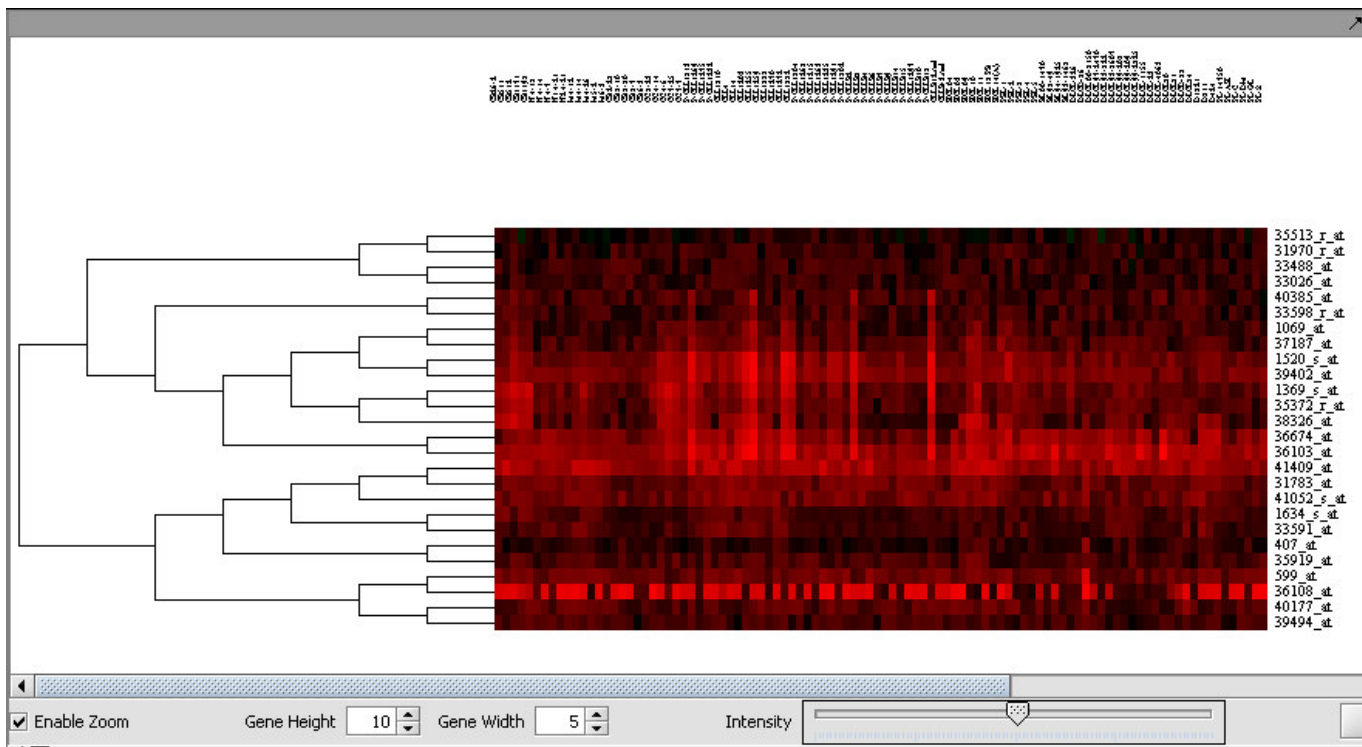
<b><u>Hierarchical Tool</u></b>	<b><u>Description</u></b>
Single linkage	The distances are measured between each member of one cluster each member of the other cluster. The minimum of these distances is considered the cluster-to-cluster distance.
Average linkage	The average distance of each member of one cluster to each member of the other cluster is used as a measure of cluster-to-cluster distance.
Total Linkage	The distances are measured between each member of one cluster each member of the other cluster. The maximum of these distances is considered the cluster-to-cluster distance.



Distances metrics available for cluster computations are:

1. Euclidean
2. Pearson's Correlation
3. Spearman Rank

The Dendrogram view is very flexible. The image can be scrolled. The cluster tree can be saved as an image. The sizes of the grid in the mosaic representing genes can be altered. The color intensities can be altered. A zoom feature allows subtrees to be selected, viewed, and saved as Marker sets. Mouse-over information (expression value) can be toggled with the light bulb button at lower right. Finally, image snapshots of the dendrogram can be taken.



**Figure 5-19** An example of viewing a subtree from Hierarchical Clustering in the Dendrogram component.

### 5.4.2 Self Organizing Map (SOM)

An implementation of SOM<sup>3</sup> is provided. The available parameters are:

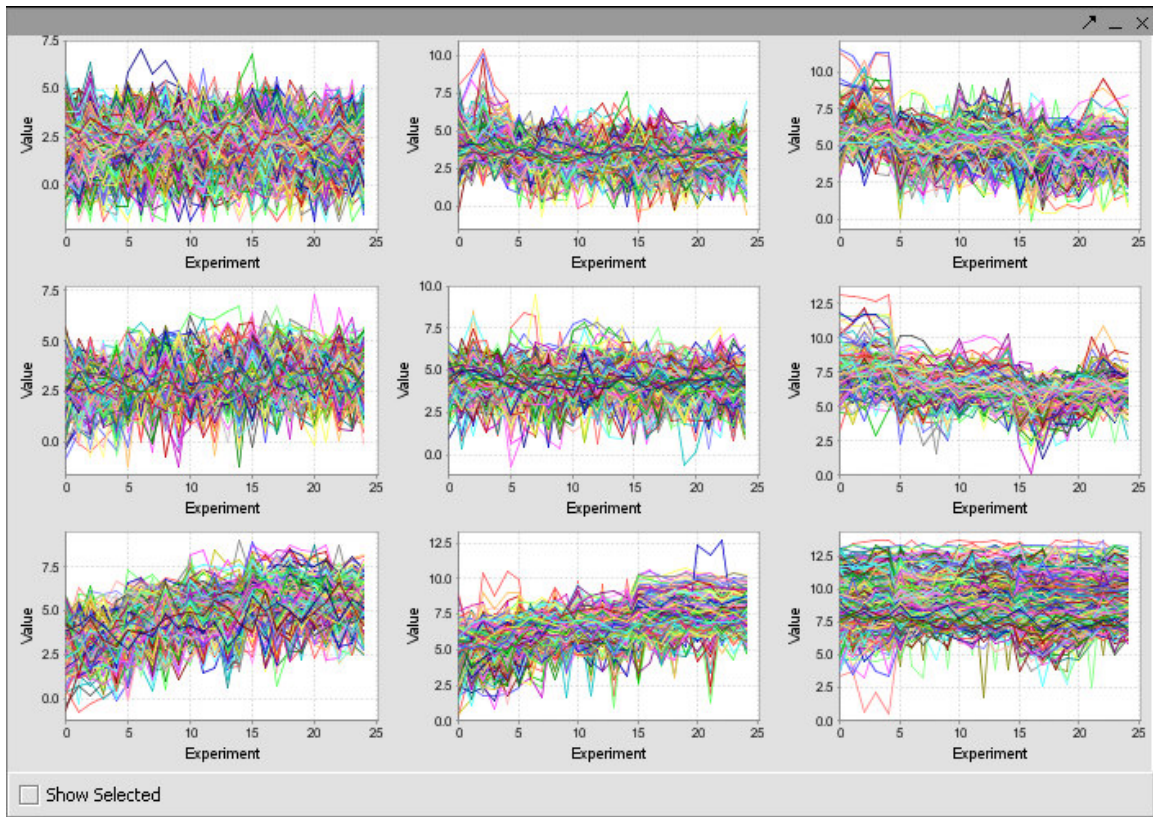
**Table 5.4-2** SOM parameters

<u>SOM Tool</u>	<u>Description</u>
-----------------	--------------------

<u>SOM Tool</u>	<u>Description</u>
Rows	The number of rows that the user desires in the resulting SOM.
Columns	The number of columns that the user desires in the resulting SOM .
<u>Radius</u>	When using the bubble neighborhood parameter this float value is used to define the extent of the neighborhood. If an SOM vector is within this distance from the winning node (the cluster to which an element has been assigned) then that node (and SOM vector) is considered to be in the neighborhood and its SOM vector is adapted.
Iteration	The number of times the dataset will be presented to the Map. Each expression element will be presented this number of times to train the nodes.
Alpha	This value is used to scale the change of individual SOM vectors when a new expression vector is associated with a node.
Functions	<p>The neighborhood options indicate the conventions (formulas) used to update (adapt) an SOM vector once an expression vector has been added into a node's neighborhood.</p> <p><b>Bubble:</b> This option uses the provided radius (see above) to determine which surrounding SOM nodes are in the neighborhood and therefore are candidates for adaptation. When this option is selected the Alpha parameter for scaling the adaptation is used directly as provided from the user.</p> <p><b>Gaussian:</b> This option forces all SOM vectors in the network to be adapted regardless of proximity to the winning node. In this case the Alpha parameter is scaled based on the distance between the SOM vector to be adapted and the winning node's SOM vector.</p>

SOM analysis uses self-organizing neural nets to identify genes with similar expression patterns, and maps expression profiles into the cells of user defined grids. The **SOM Clusters View** (Figure 5-20) can then be used to explore the resulting maps. It contains a grid of profile graphs depicting each cluster in the SOM results. The number of cells seen in the results grid equals the product of the number of rows and columns provided in the analysis input. Clusters resulting with no entries are shown as empty profile graphs in the plot. Each profile in a grid cell corresponds to the gene profile of the corresponding gene in the input dataset. Each cluster can be viewed alone by selecting the **Show**

**selected** checkbox and selecting the appropriate grid cell. Finally, image snapshots of the SOM grid as well as zoomed-in images can also be taken.



**Figure 5-20 The SOM Clusters View window**

geWorkbench1.0's implementation of these algorithms is based on their implementation in the [Multi Experiment Viewer \(MEV\)](#) platform, which is freely available from The Institute for Genomic Research (TIGR).

### **5.4.3 The Dataset Annotation Tool**

This panel provides a simple text window for adding any textual information that the user wishes to associate with a particular data set. Examples might include annotations found on the CGAP web site, questions that arise during the analysis which the user may wish to pursue at a later time, actions that were taken that are not otherwise tracked by geWorkbench1.0, etc.

Cut and paste operations are supported to facilitate importing and/or exporting text to and from this window. In particular, as the "parent" data set's annotations are not inherited by the "child" nodes, the user may wish to copy and paste some of these as new data sets are derived. Any text entered in this window will be saved and retrieved with the experiment when the workspace is reopened.

geWorkbench1.0 supports user annotation/comments for images that appear in the Project Folders component. First an image must be created by one of the components, for instance through the Expression Value Distribution component as described in section 3.2.8. Then the user can attach an annotation by right mouse clicking on the generated image to create a new annotation label for the image. The user can also used Edit -> Rename -> File to accomplish the same task.

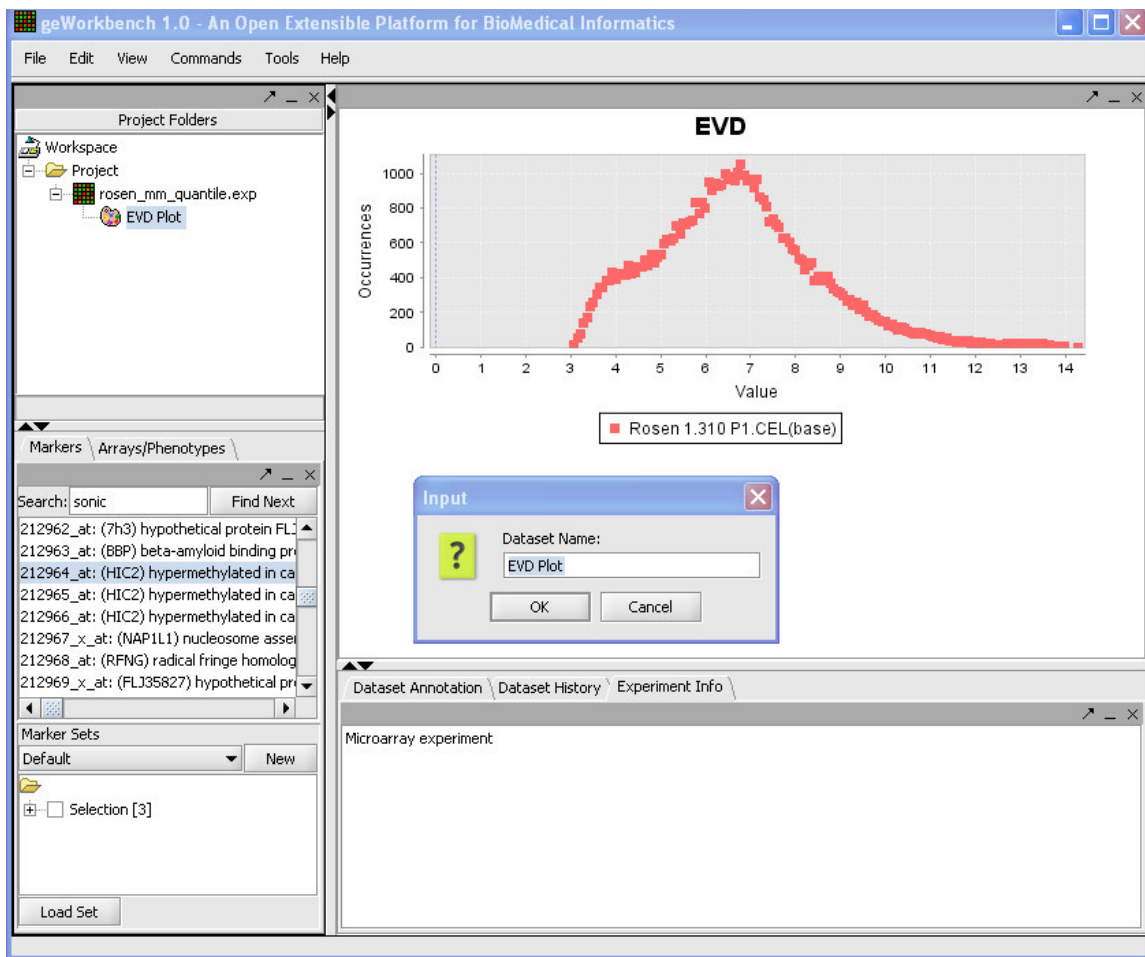


Figure 5-21 Adding a dataset annotation

#### 5.4.4 The Experiment Info Tool

This read-only text window displays the textual preamble that precedes the data in most experiments. While it is not possible to modify the text in this display, the user can copy

that text if desired into a **Dataset Annotation** panel. For example, when two independent data sets are merged to form a new data set, the latter has no experiment information associated with it. Using copy and paste operations, the user can copy the experiment information from each of the original data sets into the **Dataset Annotation** window for the merged data.

### Software Tools and References:

1. **GeneChip Arrays.** 2003. Affymetrix. First and most comprehensive whole human genome expression array. <http://www.affymetrix.com>
2. **Eisen, M.B., Spellman P.T., Brown P.O., and Botstein D.** 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA 95:14863-14868.
3. **Kohonen, T.** 1997. Self-organizing Maps. Springer-Verlag, Berlin.
4. **Tamayo P., Slonim D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E.S., and Golub T.R.** 1999. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. USA 96:2907-2912.
5. **Saeed A.I., Sharov V., White J., Li J., Liang W., Bhagabati N., Braisted J., Klapa M., Currier T., Thiagarajan M., Sturn A., Snuffin M., Rezantsev A., Popov D., Ryltsov A., Kostukovich E., Borisovsky I., Liu Z., Vinsavich A., Trush V. and Quackenbush J.** 2003. TM4: a free, open-source system for microarray data management and analysis. Biotechniques 34(2):374-8.
6. **Covitz P.A., Sahni H., Gustafson S., and Buetow K. National Cancer Institute Center for Bioinformatics.** 2002. Cancer Bioinformatics Infrastructure Objects (caBIO): An open-source, object oriented API for biomedical informatics. Objects in Bio- & Chem-Informatics: <http://lsr.omg.org/oibc2002/>
7. **GenePix Scanner.** Axon Instruments. [http:// www.axon.com](http://www.axon.com)
8. **Microarray Gene Expression Data Society (MGED).** Microarray and Gene Expression (MAGE). <http://www.mged.org/Workgroups/MAGE/mage.html>
9. **Cancer Bioinformatics Objects (caBIO).** National Cancer Institute Center for Bioinformatics. <http://ncicb.nci.nih.gov/core/caBIO>
10. **Hegde, P., R. Qi, K. Abernathy, C. Gay, S. Dharap, R. Gaspard, J.E. Hughes, E. Snesrud, N. Lee, and J. Quackenbush.** 2000. A concise guide to cDNA microarray analysis. BioTechniques 29:548-556.

# 6 Sequence Alignment

## 6.1 Overview

The comparison of sequences is central to the study of genes and genomes, and can illuminate function, regulation and evolution. For example, highly conserved protein coding sequences imply structural similarity and tend to be causally related to a common function. Therefore, in studying a gene or other sequence, it is important first to detect all significant similarities between the encoded amino acid or nucleic acid query sequence and any accumulated protein or nucleic acid sequence data, for example that contained in the national sequence databases.

## 6.2 BLAST

BLAST (Basic Local Alignment Search Tool) is designed to rapidly search for pair-wise alignments between a query protein or nucleic acid sequence and each sequence in one or more sequence database. Since BLAST detects local as well as global alignments, regions of similarity embedded in otherwise unrelated sequences are detected. Both types of similarity may provide important clues to the function of uncharacterized proteins and suggest evolutionary relationships. While BLAST is well suited for the rapid identification of orthologous sequences, it will fail to find sequences homologous to a query unless they match a minimum number of consecutive residues and may miss distant relationships obscured by insertions and deletions. BLAST is run via a remote service, with the option provided of using a version hosted at Columbia, or dispatching the job to servers at NCBI. Most of the same options found in standard BLAST implementations are offered.

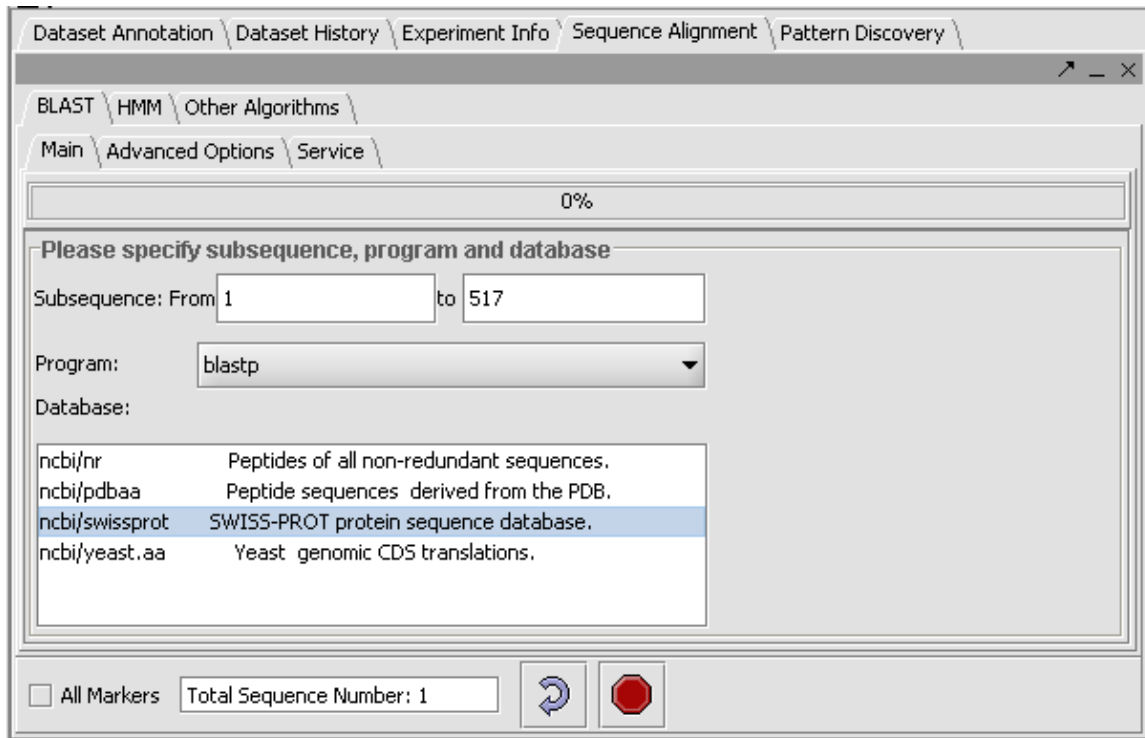
## 6.3 Tutorial

In this tutorial we will provide an example of a BLAST search. The other Sequence Alignment algorithms use substantially the same format. For this purpose, we will load a sequence into the Project Folders component, run a BLAST search.. The FASTA formatted amino acid sequence (filename NP\_077744-Wilms.fasta) for the human Wilm's tumor gene used in this example can be found among the sample data files on the geWorkbench1.0 website and downloaded to your local machine.

### 6.3.1 Running BLAST

We will begin by opening the above mentioned sequence file NP\_077744-Wilms.fasta in the Project Folders component. Now press the Sequence Alignment panel tab and then press the BLAST subpanel tab (Figure 6-1). You will see in the Subsequence text boxes that the sequence boundaries are From 1 to 517; this is the full length of the input sequence. We

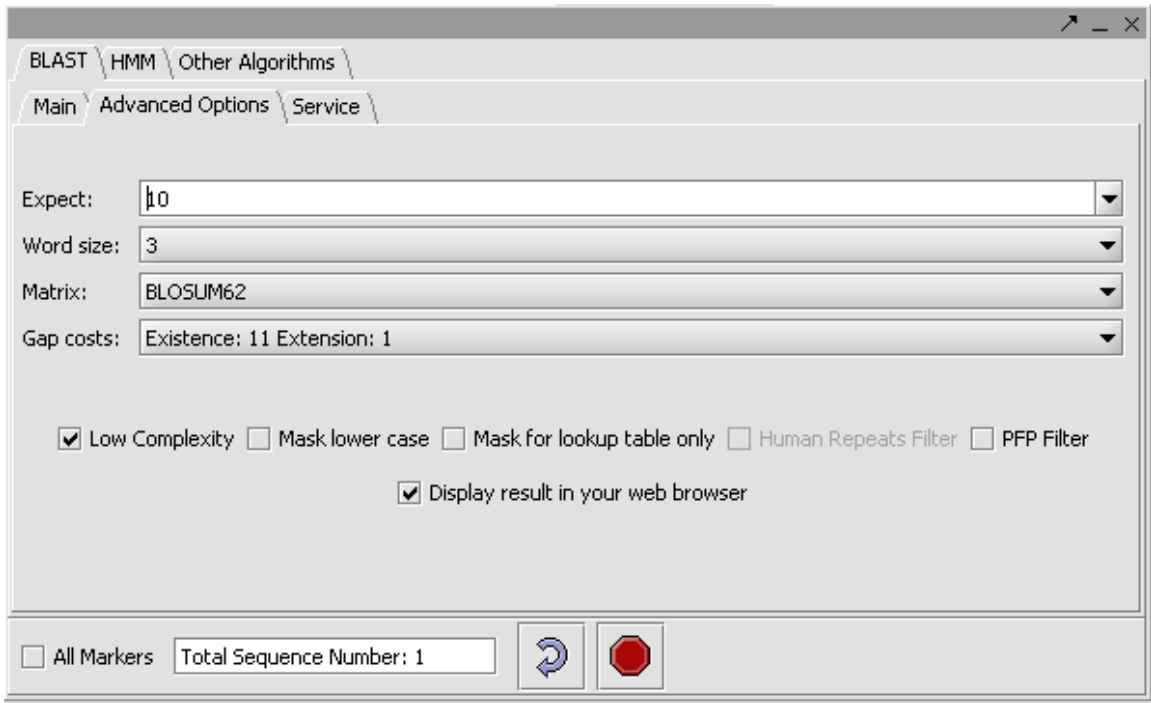
may want to narrow the interval, depending on our search results, but we'll leave it as is for now. Notice that, when the panel is first opened, the **Database** text box is empty. We next “Select a **Program**” from the dropdown list. When we select **blastp**, a list of databases appropriate to our program choice appears in the text box. We will highlight **ncbi/swissprot**.



**Figure 6-1 Sequence Alignment - Running BLAST**

Next, open the **Advanced\_Options** tab. Figure 6-2 shows the defaults for other BLASTP options, we will accept these default values.

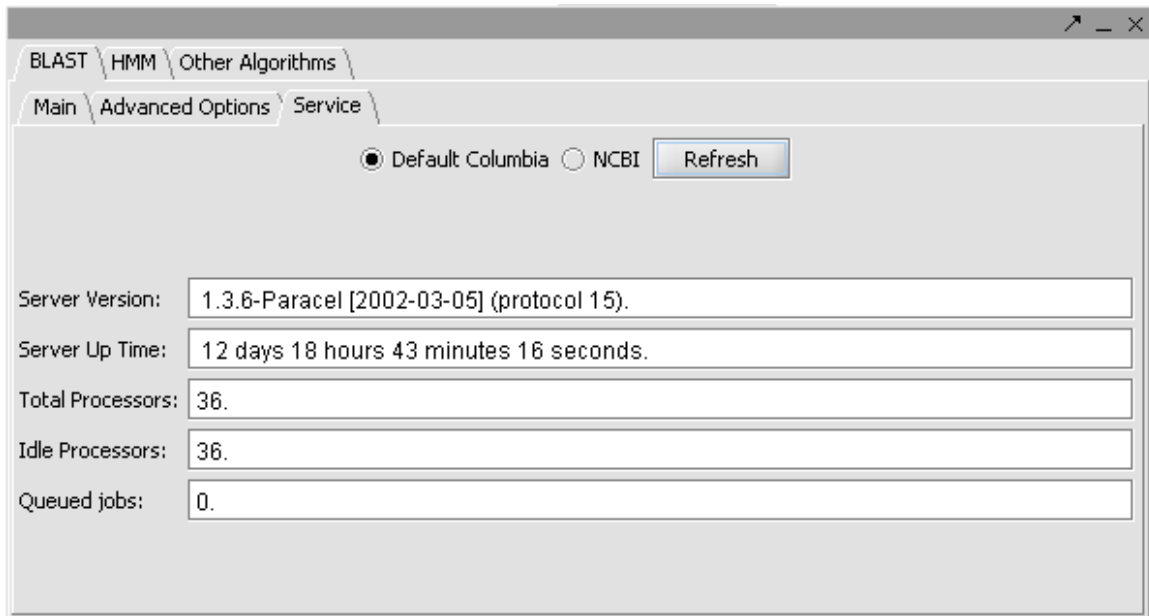





**Figure 6-2** Setting advanced options for sequence alignment.

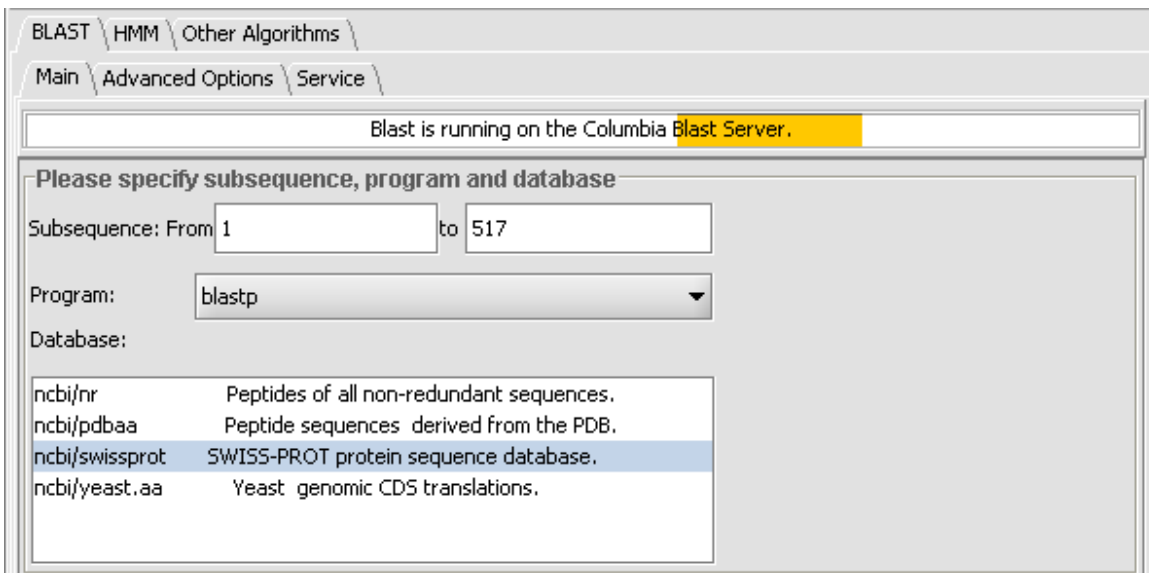
Before we run the job, we may want to check the **Service\_Info** (Figure 6-3). The BLAST job can run on a server at Columbia or on an NCBI server. We have here hit the refresh button to retrieve the Columbia server status. If we see that there are few **Idle Processors** or many **Queued jobs**, we may expect a slower return of our results. We may also want to check the server info if our results are slow to return after submission.





**Figure 6-3 Checking BLAST server info.**

Now go back to the BLAST subpanel and press the button . When we do, the **Progress bar** is set in motion (Figure 6-4).



**Figure 6-4 A BLAST job in progress.**

### 6.3.2 Analyzing the Results

When the job is finished, a stand-alone window is generated that shows the BLAST results in the Paracel BLAST format (similar to the NCBI BLAST results display), with hyperlinks to NCBI data pages and BLAST alignments. We can also look at the results in the **Alignment Results Panel** in the **View Area** (Figure 6-5).

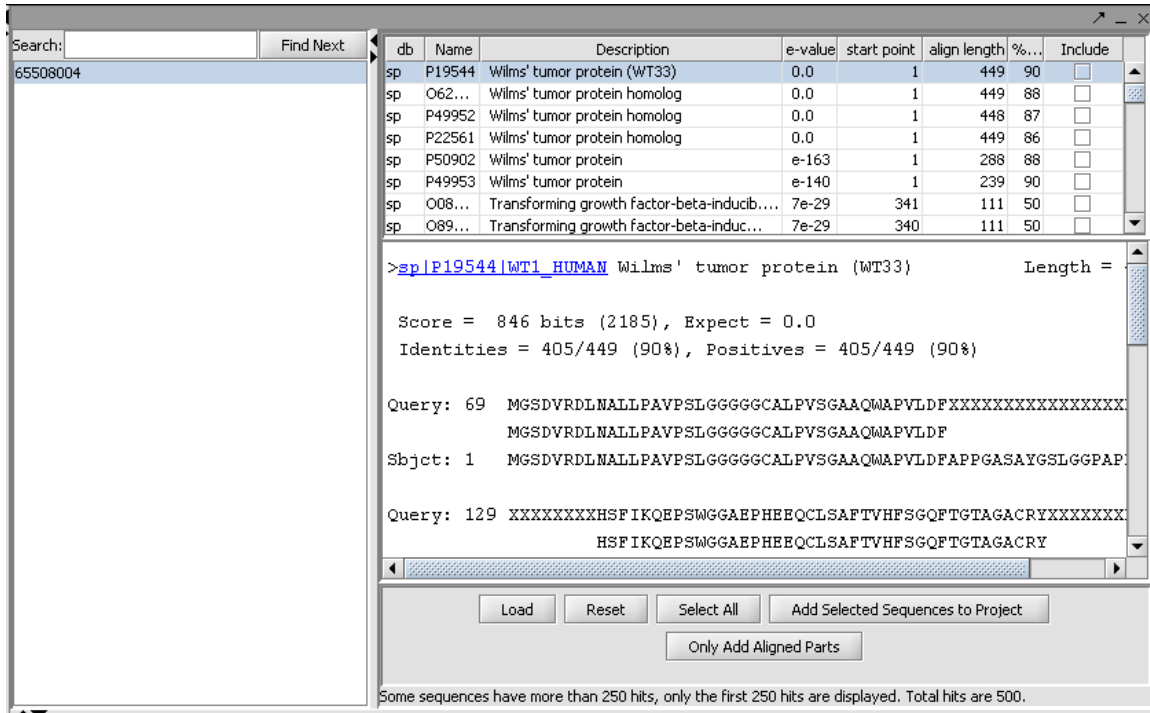


Figure 6-5 Alignment Results Panel display of BLAST analysis

The panel shows the same BLAST hit data as does the NCBI format. You will see at the far right of the panel, an additional column **Include**, with check boxes in each row. You can check any number of boxes, or press **Select All** to check all boxes automatically. As they are checked, the button **Add Selected Sequences to Project** turns yellow. Pressing this button will now add the five selected sequences to the Project Panel as a new dataset node at the same level as the original sequence file. This dataset can now be analyzed by other geWorkbench1.0 components.

## 6.4 Component Layout and Operation

### 6.4.1 Component Visual Elements

This section describes the various visual elements of the **Sequence Alignment Panel** shown in Figure 6-1.

**Sequence Alignment:** This tab opens the Sequence Alignment Panel. This panel has five tabs.

**BLAST:** This tab provides access to Paracel-enhanced BLAST algorithms and NCBI databases and controls for choosing BLAST parameters. The following are visual elements of the BLAST subpanel:

**Progress bar:** The oscillating yellow bar indicates that search is in progress. When the job is finished, the progress bar shows the time and date.

**Subsequence:** This pair of type-in text boxes allows the user to set subsequence boundaries (default is the full-length sequence).

**Database:** This listbox is initially empty and brings up a selectable list of databases after choosing a BLAST program. For `blastn`, `tblastn` and `tblastx`, the listbox choices are: `ncbi/ntl`, `ncbi/pdbnt`, and `ncbi/yeast.nt`. For `blastp` and `blastx`, the listbox choices are: `ncbi/nr`, `ncbi/pdbaa`, `ncbi/swissprot`, and `ncbi/yeast.aa`. The selected database will be highlighted blue.

**Program:** This drop-down list provides choices among the following programs: `blastn`, `blastp`, `blastx`, `tblastn`, and `tblastx`.

**BLAST:** This button connects the panel to the server and initiates a BLAST search.

**Stop:** This button disconnects the server and ends the search.

**Advanced\_Options:** This tab accesses controls for setting adjustable parameters for BLAST and Smith-Waterman searches. The following are visual elements of the AO subpanel:

**Frame shift penalty:** This dropdown list applies to `blastx`. It sets a penalty for out-of-frame (OOF) alignments (default is **NO OOF**).

**Query genetic code:** This dropdown list allows the user to choose which genetic code is to be used in the `blastx` translation of the query choices. Other genetic codes are **Vertebrate Mitochondrial**, **Yeast Mitochondrial**, **Invertebrate Mitochondrial**, **Echinoderm Mitochondrial** and **Euplotid Nuclear** (default is **Standard**).

**Matrix:** This dropdown list offers selections among similarity matrices. Choices are **BLOSUM 50**, **BLOSUM62**, **BLOSUM100**, and **BLOSUM150** (default is `dna.mat`).

**Expect:** This dropdown list enables the user to set the **EXPECT** values; i.e., the statistical significance thresholds for reporting matches against database sequences. (Default is **10**).

**PF filter:** This checkbox enables the **Paracel Filtering Package** pre-filtering option (default is **CHECKED**).

**Low Complexity:** This checkbox enables the **Paracel low-complexity filter** to mask segments of the query sequence that have low complexity.

**Mask the lookup table only:** This checkbox enables the user to mask only for purposes of constructing the lookup table used by BLAST.

**Display result in your web browser:** This checkbox allows the user to specify that the results be displayed in the web browser (default is CHECKED).

**Server\_Info:** This tab controls access to sequence alignment servers and displays information about them. The following are visual elements of the SI subpanel:

**Connect:** This button establishes a connection to the server. Details appear in text boxes below.

**Refresh:** This button updates server info in text boxes.

**Stop:** This button terminates the search.

**Server Version:** This text box shows that shows that the server is 1.3.6-Paracel [2002-03-05] (protocol 15).

**Server Up Time:** This text box shows how long the server has been running.

**Total Processors:** This text box shows how many processors (currently 40) in the server.

**Idle Processors:** This text box shows the number of idle processors.

**Queued jobs:** This text box shows the number of jobs in the server queue.

## **6.5 References**

### *BLAST*

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." *J. Mol. Biol.* 215:403-410.

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D.J. (1997) "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucleic Acids Res.* 25:3389-3402.

# 7 Pattern Discovery

## 7.1 Overview

Sequence Pattern Discovery is the process of identifying nucleotide or amino acid arrangements, also called motifs that are enriched in a set of sequences. Such motifs may identify regions that have been preserved by evolution and which therefore may play a key functional or structural role. geWorkbench1.0 currently provides three modes of Sequence Pattern Discovery: Regular Discovery, Hierarchical Discovery, and Exhaustive Discovery.

Regular Discovery is based on the algorithm SPLASH (Califano, A., 2000); it generates a list of all regular expression patterns (*motifs*) that satisfy a user-defined **minimum support** and a **minimum density criteria**. The former determines the minimum number of times a pattern must occur in the sequence set to be reported. This can also be expressed as the minimum percent of sequences that must contain the pattern. The latter determines how sparse the pattern can be, in other words the minimum number of matching characters  $k$  (any character except for the dot character “.”) over a window of predefined length  $w$ .

SPLASH-based motif discovery is extremely efficient and can process most large protein super-families in a few minutes on a conventional workstation. Discovery is uniquely effective in identifying sparse patterns using extremely low-density constraints, and the results obtained with Discovery can provide the core for a large number of more specific local alignments.

Exhaustive Discovery starts from a relatively high minimum support (e.g. patterns occurring in 75% of the sequences) and it progressively reduces the support, until a statistically significant pattern is discovered. Discovered patterns are reported and then masked in the sequence set so that they are no longer discovered. Then the process continues iteratively until the minimum support reaches a lower user-defined limit. Exhaustive Discovery, thus, produces a list of non-overlapping motifs in order of support.

Hierarchical Discovery is based on the top-down clustering algorithm CASTOR (Liu and Califano, 2003); it proceeds similarly to Exhaustive discovery, except that each time a pattern is reported, the set is split into sequences containing and sequences not containing the pattern. Discovery continues hierarchically in the individual split subsets. This produces a binary tree of sequence sets and associated patterns. Discovery stops when the sets become smaller than a user-defined limit or when statistically significant patterns can no longer be discovered. In addition, HMM models are also generated.

## 7.2 Tutorial

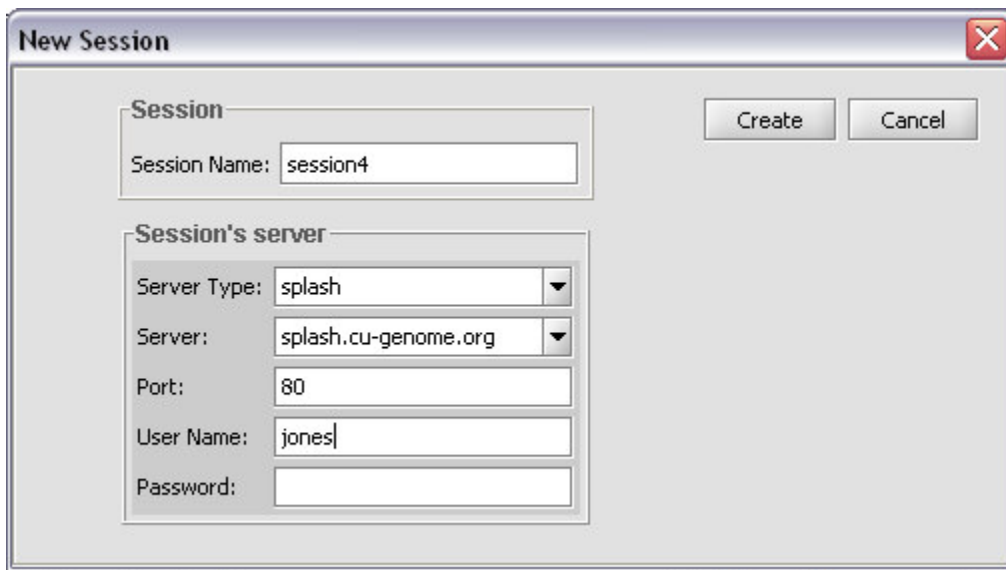
In this tutorial we will present examples of **Normal**, **Hierarchical**, and **Exhaustive** analyses. To demonstrate **Normal**, we will load a dataset of 254 amino acid sequences containing H1 histone sequences and attempt to discover a common motif in at least 95% of the sequences. (This file, H1H5\_HistoneDB\_NHGRI.fasta is available for download from

the geWorkbench1.0 website as part of the tutorial data). To demonstrate **Hierarchical**, and **Exhaustive** analyses, we will search the same dataset using different constraints. We will then describe three panels that can display the resulting patterns. These are **Sequence**, **Position Histogram**, and **Patterns(Promoter)**.

### 7.2.1 Discovery analysis

In this example, we begin the **Discovery** process by loading the H1-histone sequence dataset from local files into the **Project Folders** component. After loading the **Project Folders**, you can see the list of sequence IDs for the dataset by clicking on the **Markers** tab in the Selection Area.. You can also see a graphic display of the dataset by opening the **Sequence** component in the Visual Area. In this view each sequence is represented by a line proportional to its length, preceded by the sequence ID. Now open the **Pattern Discovery** component in the Analysis area. In the default view, you will see the Normal radio button has been selected and the Basic sub-panel is open with default parameter values. Change the settings in the text boxes to **Support: 80%**, **Min. Tokens: 7**, **Density Window: 12**, and **Density Tokens: 4**. Next, open the **Advanced** sub-panel tab and uncheck **Exact Only** to activate the BLOSUM50 similarity matrix.

To begin the **Discovery** process, press **Execute icon** . This brings up a **New Session** dialog box.



**Figure 7-1** Pattern Discovery dialog box

Enter the following: **Server: splash.cu-genome.org, port: 80**. For the session and User Name, you can enter any convenient values (if in the future a login is required to access remote servers, then a valid User Name and Password will need to be entered). Note, subsequent searches in the same session will not elicit the dialog box. Then press **Create**. Looking back at the **Pattern Discovery** panel, you will see the following series of progress

bar text messages: **Uploading**, **Processing seeds**, **Discovering**, **Collating**, and **Done**. The **Discovery Table** will then fill with information on the discovered motifs. The table in Figure 7-2 shows that seven similar patterns were found. If you select a motif, it will be highlighted in blue. Here, the motif [NDE][RK].G.S...[ILM].[RK].[ILMV] was selected. The table shows that it is found once in each of 209 of the 254 input sequences, spans 14 tokens and contains 7 full character tokens. Right-clicking on the table elicits a pop-menu, described in Section .7.4

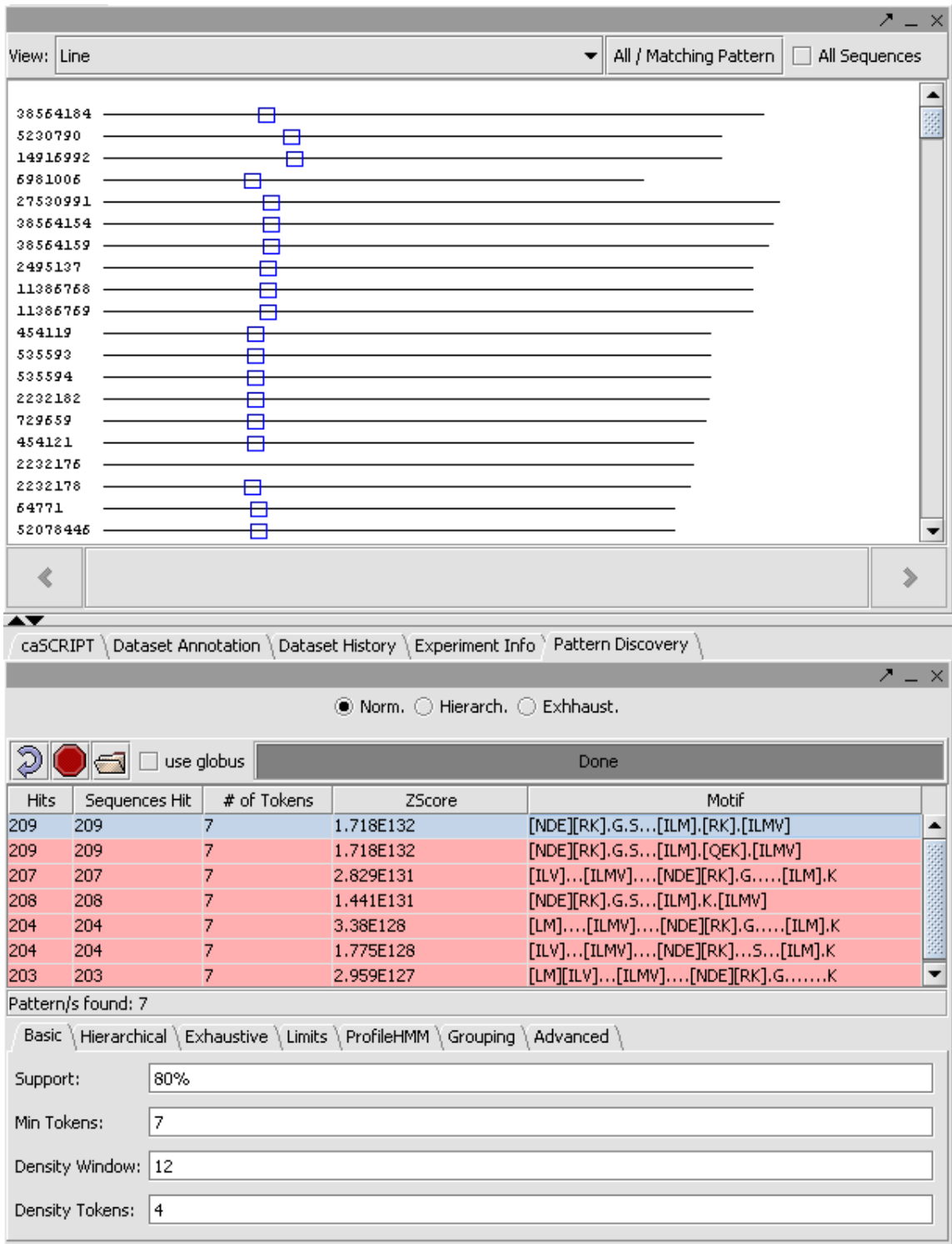



Figure 7-2 Normal pattern discovery



## 7.2.2 Hierarchical analysis

For an explanation of hierarchical analysis see ["http://www.research.ibm.com/splash/Hyerarchical/HierarchicalDiscovery.htm"](http://www.research.ibm.com/splash/Hyerarchical/HierarchicalDiscovery.htm).

To perform Hierarchical analysis on the histone dataset:

1. Select the Hierarc radio button.
2. Then set the Basic constraints as in the Normal example:
3. Leave the Advanced constraints as they were.
4. In the Hierarchical sub-panel, further options can be adjusted, but we will use the default values (Min. Cluster Size: 10, and Min. Pattern Number: 10).
5. Press the Execute icon  to initiate the search (Note – this search may take a long time). A gray progress bar is displayed while the search is in progress.
6. Results: Motifs and their frequency are listed in the Display Area as nodes in expandable folders (Figure 7-3).

The primary node shows the total number of sequences (254) in the dataset. Expanded folders show motifs and number. Folders expand until hierarchical constraints are reached.

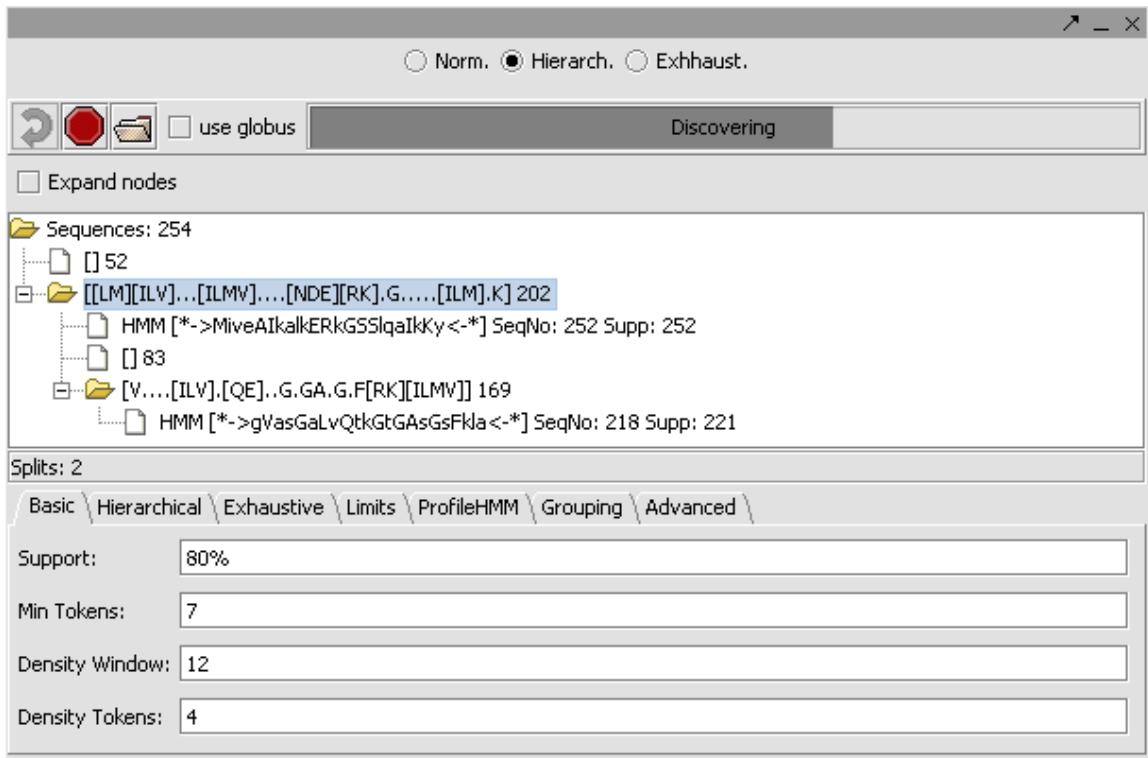
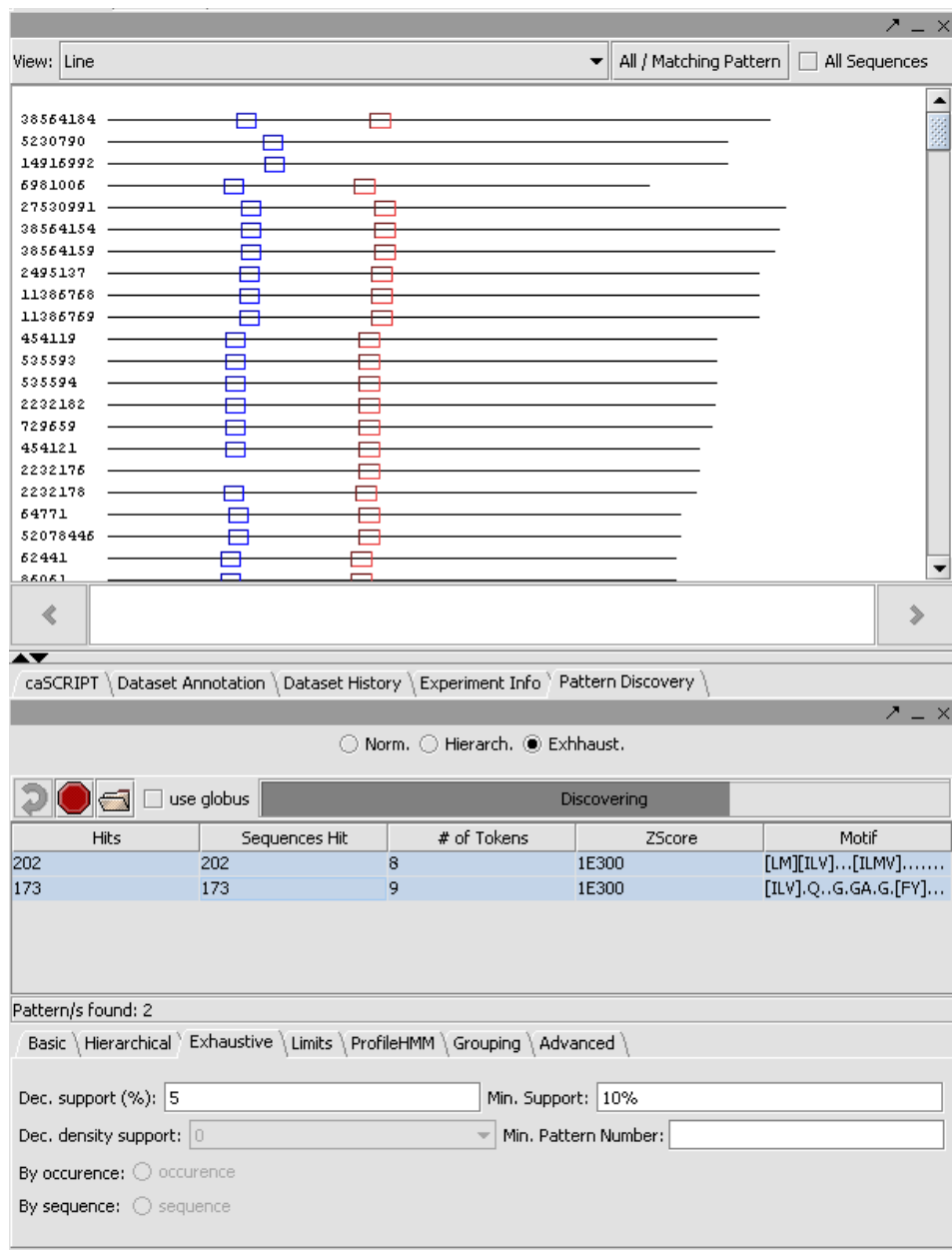


Figure 7-3 Hierarchical pattern discovery

### 7.2.3 Exhaustive analysis

In this third example, Exhaustive analysis was applied for Pattern Discovery on the same dataset, using the same Basic and Advanced constraints as in the hierarchical example. However, it is possible to set additional constraints, specified in the Exhaustive sub-panel. Here, we left the default parameters for Dec. Support(%) at 5 and Min Support at 10% (Figure 7-4).



**Figure 7-4 Exhaustive pattern discovery in progress, two non-overlapping results highlighted and displayed in the Sequence component.**

### 7.3 Visualization of Pattern Discovery Results

Various aspects of selected motifs can be effectively displayed in the **Sequence**, **Position Histogram**, and **Promoter** components in the **Visual Area**. This is applicable to all three of the **Discovery** modes. To display the motifs in the **Position Histogram** as in Figure 7-5 select the motifs as in Figure 7-2, press the **Position Histogram** tab, and press **Plot Position**. The same motifs are displayed as in **Sequence** component shown above (Figure 7-4). Notice, each motif is represented by a specific color and these colors are the same in both displays..

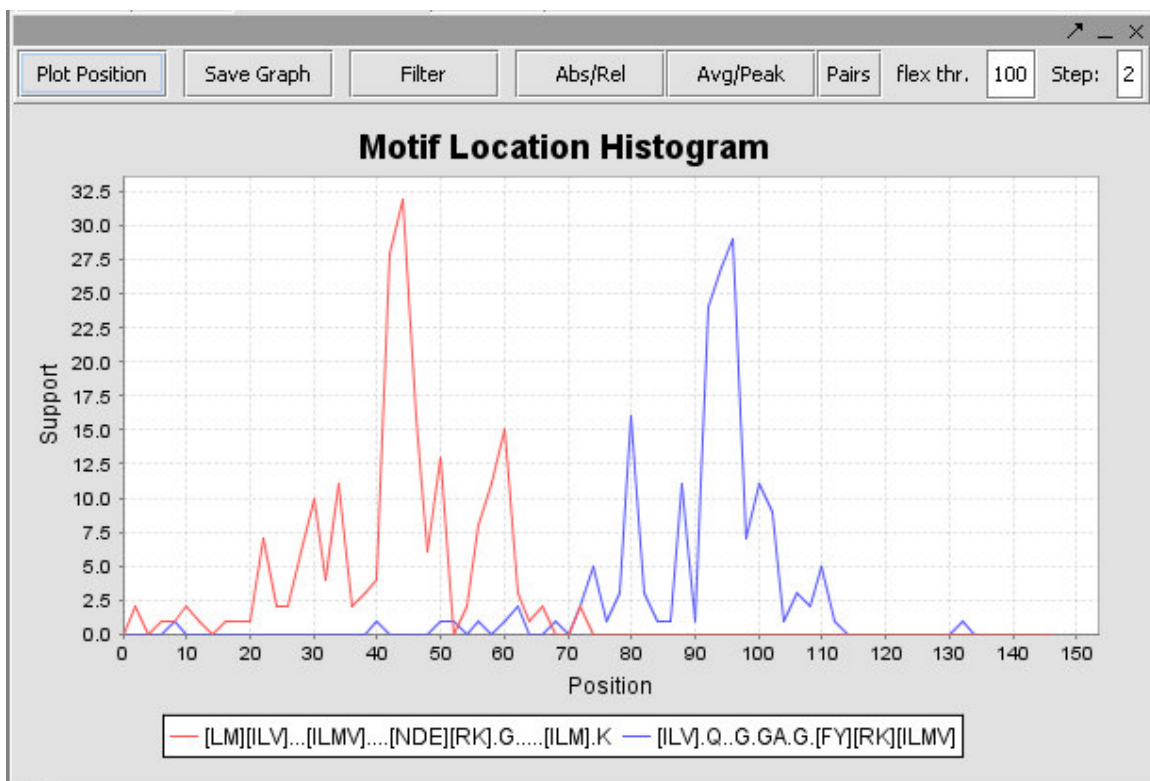


Figure 7-5 Position Histogram

### 7.4 Component Visual Elements

This section describes the various visual elements of the **Pattern Discovery** component (see the [SPLASH](#) page at IBM for further details).

**Display Area:** this is where input datasets and search results are shown either in graphical or textual representation.

**Normal:** this radio button implements Normal Pattern Discovery.

**Hierarc:** this radio button implements Hierarchical analysis.

**Exhaustive:** this button implements Exhaustive analysis.



**Execute:** this command button brings up **New Session** dialog box to start **Discovery**.



**Stop:** this command button stops a search in progress.



**(Load):** this command button loads **Discovery** results from a local file.

**Progress Bar:** in the narrow band above the **Display Area**, an animated slider pulsates back and forth as long as the search is ongoing.

The following are columns in the **Sequence Discovery Table**:

**Hits:** this is the total number of times a motif appears in a sequence dataset.

**Sequences Hit:** this is the total number of different sequences in which the motif is found.

**# of Tokens:** this is the number of full-character tokens in the motif.

**Zscore:** this is a measure of how often the motif would be found in a random set of sequences of the same size and composition as the project dataset.

**Motif:** this is a sequence of tokens, which may be full character or wildcard. Periods (...) correspond to wild cards. Parentheses identify “either/or” tokens that satisfy the BLOSUM matrix.

The following are elements in the pop-up menu, elicited by right-clicking on the **Discovery Table**:

**Mask Pattern:** this menu item filters sequences containing selected motifs, so that they are not re-discovered or displayed in the **Sequence Panel** when **Discovery** is executed.

**Unmask all Patterns:** this menu item removes all the masks applied by **Mask Pattern**.

**Save Patterns (Regex Only):** this menu item saves the sequences of the motifs to a local file.

**Save Selected Patterns:** this menu item saves selected motifs from the **Discovery Table** to a local file. The saved table can be re-loaded into **geWorkbench1.0** by pressing the **Load** button

**Save All Patterns:** this menu item saves the complete **Discovery Table** to a local file. The saved table can be re-loaded into **geWorkbench1.0** by pressing the **Load** button

**Add Patterns to Project:** this menu item saves the results in the Discovery Table as a node in the Project Folder.

The following are visual elements in the **Basic** subpanel:

**Basic:** this tab opens a subpanel for defining motif parameters.

**Support:** this sets the minimum number or % of sequences in the set containing the shared motif. Type in a % sign to indicate %, e.g. 80% or 80 sequences (no percent sign).

**Min. Tokens:** sets the minimum number of density tokens in the density window.

**Density window:** is a window within the motif that counts tokens and wild cards.

**Density tokens:** are the full character tokens (not wildcards) in the density window.

The following are visual elements in the **Hierarchical** subpanel:

**Hierarchical:** this tab opens a subpanel for setting **Hierarchical** specific parameters,

**Min. Cluster Size:** this sets a lower limit to number of sequences that will be searched for a shared motif.

**Min. Pattern Number:** this sets a lower limit to the number of sequences that must contain a shared motif to be included in the hierarchy.

The following are parameters in the **Exhaustive** subpanel:

**Dec support (%):** this sets the size of intervals by which support level is decremented in successive searches (default is 5).

**Min Support:** this sets the lower limit on the percentage of sequences that must contain a specific motif (default is 10%).

**Dec. density support:** INACTIVE

**Minimum Pattern Number:** this sets a lower limit on the number of motifs in a cluster.

**By occurrence:** INACTIVE

**By sequence:** INACTIVE

The following are visual elements in the **Limits** subpanel:

**Limits:** this tab opens a subpanel for setting maximum pattern number and run time.

**Max Pattern Number:** **this** limits the number of patterns to discover.

**Max Run Time (sec):** this limits search time.

The following are visual elements in the **ProfileHMM** subpanel:

**ProfileHMM:** this tab opens a subpanel for setting parameters for profile-hidden Markov models (pHMMs).

**Entropy Threshold:** N/A

**Conserved Region Extension:** bases on either side of conserved region to be considered.

**Sliding Window Size:** bases considered in sequence being searched.

The following are visual elements in the **Grouping** subpanel:

**Grouping:** this tab opens a subpanel for setting Grouping parameters.

**Type:** feature is disabled.

**Size:** N/A

The following are visual elements in the **Advanced** subpanel:

**Advanced:** this tab opens a subpanel for setting **Advanced** parameters.

**Exact Only:** this check box, if unchecked, activates use of BLOSUM matrices.

**Count sequences:** this check box allows you to sort patterns by number of occurrences, number of distinct sequences in which they occur, length, or Zscore.

**ZScore:** calculate and use the ZScore (which is a measure of the statistical significance) to filter patterns to display,

**BLOSUM50:** this is the default substitution, or similarity, matrix used for polypeptide motif discovery; others available in the scroll panel are **BLOSUM100** and **BLOSUM150**

**Similarity Threshold:** a measure of the stringency of the search.

**Minimum ZScore:** **minimum value of ZScore for pattern to be significant.**

## **7.5 References**

Califano A. SPLASH: structural pattern localization analysis by sequential histograms. *Bioinformatics* **16**:341-57 (2000).



# 8 Promoter Analysis

## 8.1 Overview

Regulation of gene expression is accomplished through the interaction of transcription factors (TFs) and their binding sites. Computational analysis of transcription factor binding sites (TFBS) is commonly performed in two fashions:

1. By searching putative regulatory sequences against a collection of known transcription factor DNA-binding signatures, represented as a position weight matrices (PWMs) [Lawrence and Reilly, 1990].
2. By discovering new putative DNA-binding motifs using algorithms such as SPLASH [Califano, 2000] and AlignACE, [Roth *et al*, 1998].

Both approaches are readily implemented in geWorkbench1.0 and can easily interoperate. For instance, motifs newly discovered using SPLASH and known TF signatures can be mapped on a set of putative regulatory genomic sequences. With respect to the known signatures, geWorkbench1.0 uses the JASPAR Transcription Factor Binding Profile Database ([http://jaspar.cgb.ki.se/cgi-bin/jaspar\\_db.pl](http://jaspar.cgb.ki.se/cgi-bin/jaspar_db.pl)). JASPAR is an open-access, non-redundant, curated set of transcription factor DNA binding preferences in multicellular eukaryotes. It also allows loading new TF signatures using a simple file format.

geWorkbench provides many avenues through which genes can be chosen for TFBS analysis. Gene selections can be made e.g. in the Gene Ontology browser component, through results of hierarchical clustering or in the reverse engineering component. Upstream sequences can then be retrieved for these genes from the UCSC GoldenPath database, by defining the length in base-pairs of the genomic sequence upstream and downstream of the transcription initiation point. The latter can then be used either to match known TF signatures or to discover new ones.

## 8.2 Tutorial

In this tutorial we will discuss two examples. In the first, we will load putative regulatory genomic sequences for a set of genes that have a common GO function. We will attempt to find known TF signature matches. In the second we will show how this component interoperates with the Project Folders component and the Pattern Discovery component to identify new putative DNA-binding signatures.

### 8.2.1 Transcription Factor signature analysis

For this tutorial you will need to be connected to the Internet, as genomic sequences will be retrieved from the UCSC server. The tutorial will also include use of the GO (Gene Ontology) component, so you will need to load into geWorkbench1.0 a dataset which defines the list of genes used for the analysis, which will be associated with their corresponding GO terms. For this we will use two example files available on the geWorkbench1.0 website: A file containing a set



of gene expression results from an Affymetrix Microarray experiment, webmatrix2.exp and its annotation file, HG\_U95Av2\_annot.csv.

Setup:

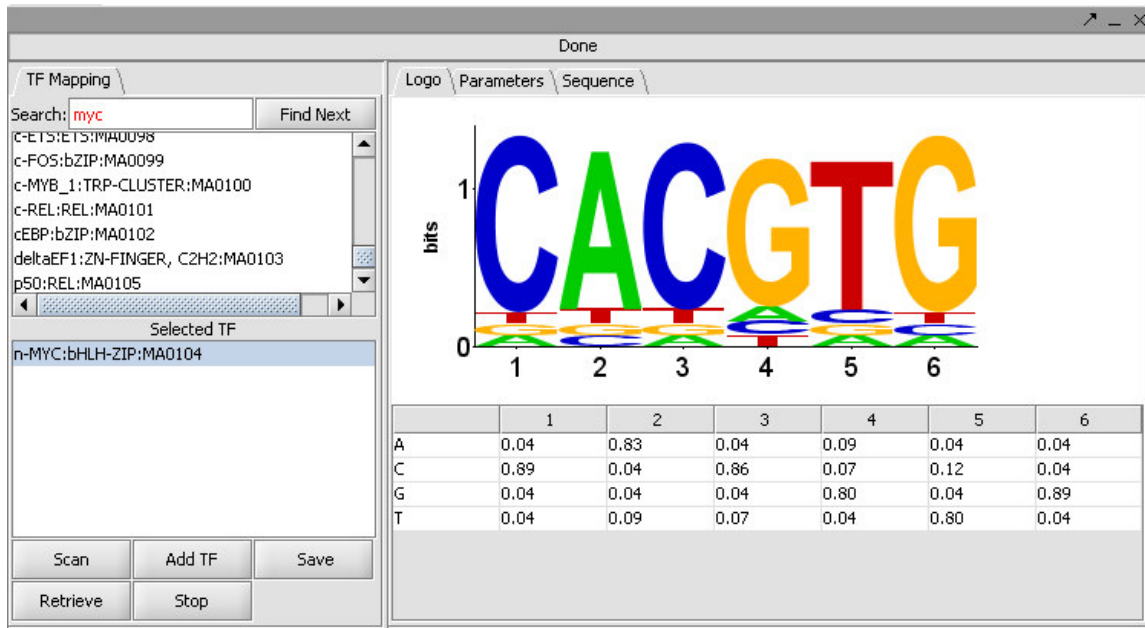
1. Load the file webmatrix2.exp, which is of type “Affymetrix File Matrix”. It will be recognized as type HG\_U95Av2 and the appropriate annotation file will be loaded. You may have to obtain this file from the Affymetrix NETAFFX website first (login required).
2. geWorkbench will build an annotation database for the experiment..

Select the **GO Term** component in the **View Area**. Under **Process**, click on the **+** sign of the following nodes to navigate to the **cell death** process: **biological process**, **cellular process**, **cellular physiological process**, and **cell death**. The **cell death** node has 370 associated entries. Right-click on the node and select **Add to Set -> Reference List** from the popup menu. These will be added to the **Markers** component as a new set labeled “cell death”. It has 83 members. **Activate** the set by checking the box in front of it in the **Markers** component. The set of 83 genes will be available to any other component in geWorkbench1.0.

You can use the **Sequence Retriever** component to retrieve upstream sequences for your list of genes. You might choose -2,000 to +2000, for example. Then, click on the **Get sequence** button. This retrieves genomic sequences corresponding to 2kbp (kilobase-pairs) upstream and 2kbp downstream of the transcription initiation site of each of the 18 genes selected in the **GO** component. Sequences are retrieved from the **UCSC GoldenPath** server [<http://genome.ucsc.edu/cgi-bin/hgGateway>]. This may take a minute or so.

After the sequences have been retrieved, you will see them appear as lines preceded by the gene access number in the rightmost box in the **Promoter** component. The line represents the full 4000 base pairs of each sequence. Note that the line for some sequences may be shorter or missing altogether. This is because there may be overlap between the selected 4Kbp and other genes or because data may be missing in the database.

We are now ready to test these genomic sequences for the presence of putative TF signatures. In the upper list box (**TF List**), double-click on the signature called n-MYC:bHLH-ZIP:MA0104. This will move the signature to the lower list box (**Selected TF**) and will display a “sequence logo” representation of the TF signature on the bottom of the component (Figure 8-1) [Schneider and Stephens, 1990].



**Figure 8-1 Promoter Logo view.**

Now click the button **Scan**. You will notice that a green progress bar will start oscillating. This shows that the currently selected signature is being calibrated against the selected sequences to adjust the sensitivity of the search. After a short while, the green progress bar will stop oscillating and you will see arrowheads appear on a number of the sequences. These are putative binding sites for the Myc complex. You can double click on the sequences to see the textual representation of the DNA and the TF signature match, shown as a colored rectangle. Note that if multiple signatures are selected in the **Selected TF** box, each one will be represented by a rectangle of a different color. If you mouse-over the colored box, the logo is replaced with the name and sequence position of the TF.

### 8.2.2 Transcription Factor signature discovery

Let us now attempt to discover additional putative binding sites by using the **SPLASH Pattern Discovery** algorithm. First, let us create a set corresponding to the selected genomic sequences using the **Add To Project** button. When asked for a name of the data set, type “cell death GO” or any other suitable description. You will now see a new DNA icon appearing in the **Project Folder** (upper left quadrant of geWorkbench1.0), with the corresponding name. If desired, you can right click on the icon to save the data set to local files using an appropriate filename.

Select **Pattern Discovery** in the **Selection Area** and set the following parameters: **Support 70%**, **Min Tokens: 7**, **Density Window: 5**, **Density Tokens 4**. Then chose the **Exhaustive** mode and press the **Execute** button. Make sure that the appropriate SPLASH server information is selected (Server: splash.cu-genome.org, port: 80) and click the **Create** button. The progress bar is activated and the motifs, as they are discovered, are added to a growing list in the **Pattern Discovery** table. After the

discovery of, for example, six motifs, pressing the **Stop** button terminates the search. Selecting the motifs (this highlights them in blue) displays their positions in the **Promoter component**. Figure 8-2 is a three-panel geWorkbench1.0 display of such an example..

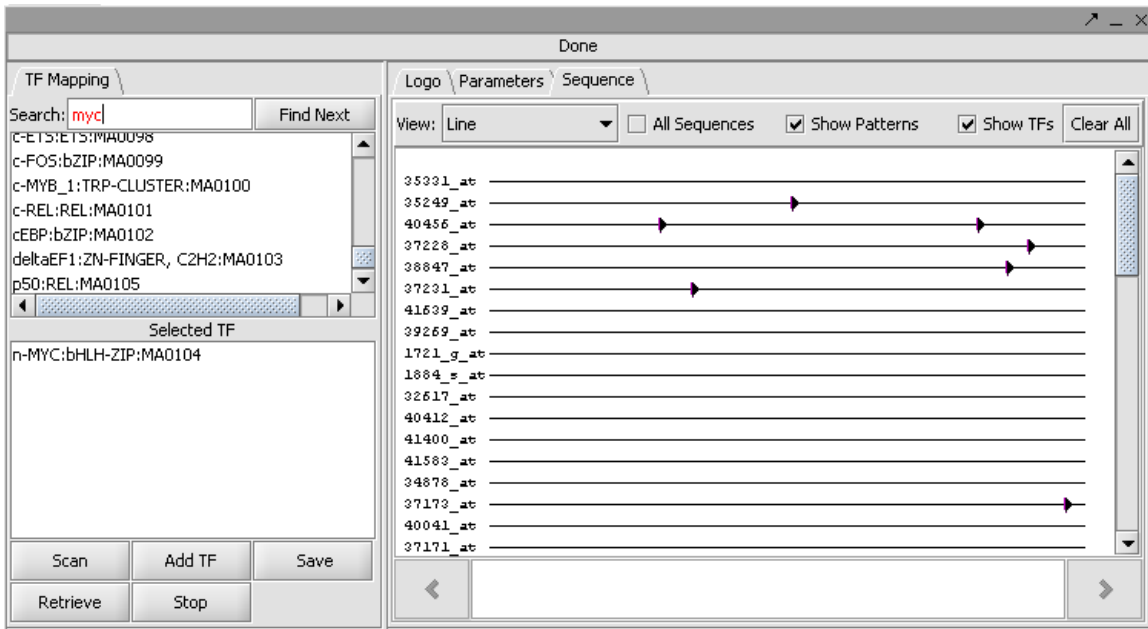


Figure 8-2 TFBS motif hits displayed in the Promoter component

## 8.3 Component Layout and Operation

### 8.3.1 Component Visual Elements

This section describes the various visual elements of the **Promoter component**, shown in **Error! Reference source not found.**

**Display Area:** this is where input datasets and search results are shown either in graphical or textual representation.

**Transcription factor list:** this list box contains the names of available transcription factors signatures from Jaspar or those that have been loaded from files.

**Selected TFs:** this list box shows transcription factors that have been activated and which can be searched against the available genomic sequences by clicking on the **Mapping** button. Double-clicking on a TF name clears it from the active list.

**Scan:** pressing this button starts the search.

**Progress Bar:** in the narrow band below the text input fields, an animated slider moves back and forth as long as the search is ongoing.

**Add TF:** this button adds transcription factors from local files to the Selected TFs list. The local .txt file is a PWM, with rows (top to bottom) corresponding to weights for A, C, T, and G and columns (left to right) corresponding to sequence positions 1 – n.

**Retrieve this button** .....

**Save:** this button saves results to local data files.

**Add to Project:** this button adds sequences from the Promoter Panel to Project Panel.

## **8.4 References**

Califano A. SPLASH: structural pattern localization analysis by sequential histograms. *Bioinformatics* **16**:341-57 (2000).

Lawrence, C.E. and Reilly, A.A. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*.**7**:41-51 (1990).

Lenhard. B. and Wasserman, W. TFBS: Computational framework for transcription factor binding site analysis. *Bioinformatics*. **18**:1135-6 (2002).

Roth FP, Hughes JD, Estep PW, Church GM. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat Biotechnol*. **16**:939-45 (1998).

Sandelin, A., Alkema, W., Engström, P., Wasserman, W. and Lenhard, B. *Nucleic Acids Res*. **32** Database Issue. (2004).

Schneider, T.D. and Stephens, R.M. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*. **18**:6097-100 (1990).

# 9 Analysis of Variance (ANOVA)

## 9.1 Overview

The ANOVA (ANalysis of VAriance) algorithm is used to determine whether any significant difference in the means exist in a dataset composed of more than two groups of experimental results. It could be used in a drug trial where one group was given placebo and several other groups were each given a different drug. Or the groups could be composed of tissue samples from different organs, and one would like to see if there was any difference in response to some treatment.

The implementation of ANOVA being used is taken from a distribution of MeV (<http://www.tm4.org/m25ev.html>). MeV is a microarray analysis and visualization tool developed at The Institute for Genomic Research (TIGR), which is now known as the J. Craig Venter Institute. Additional details about the implementation of the ANOVA algorithm can be found in the MeV manual ([http://www.tm4.org/documentation/MeV\\_Manual\\_4\\_0.pdf](http://www.tm4.org/documentation/MeV_Manual_4_0.pdf)). This version of ANOVA implements only a one-way (single-factor) analysis.

Groups of samples are defined using the standard geWorkbench Arrays/Phenotypes set mechanism, described in Section 5.1 . The user enters a critical P-value to use in judging the significance of the calculated statistic (an F-statistic) for each gene. When the ANOVA algorithm is run, an F-statistic is calculated for each gene (or marker). Any corrections chosen by the user are applied, and a list of significant markers is returned to the user. This process will be described in detail in the following.

Two types of P-value corrections are provided to account for the effects of multiple testing – the fact that the tests are being run on thousands of genes at one time. The first four P-value corrections deal with the Family-Wise Error Rate – the chance of getting a single false positive result. This is often too stringent a criterion. The second type of correction, the False Discovery Rate calculation, allows one to specify what fraction of false discoveries is acceptable. This should allow more true positives to be detected (greater sensitivity), at the price of additional false positives. The point is that the experimenter can make this choice based on the economics of testing the hypotheses that are generated.

## 9.2 Definitions of key terms

Term	Description
F-statistic	A value calculated for a particular case using the F-distribution.
F-distribution	The F-distribution arises from the ratio of the variances of two normally distributed statistics (chi-squared distributions). See <a href="http://www.statistics4u.info/fundstat_eng/cc_distri_fisher_f.html">http://www.statistics4u.info/fundstat_eng/cc_distri_fisher_f.html</a>
P-value	A probability value calculated from the theoretical F-distribution or by permutation. It is the probability that an F-statistic a least this large would be obtained under the null hypothesis of no difference in means.

### 9.3 GUI Layout

Figure 9-1 below shows the basic layout of the ANOVA component, which is located in the Analysis tab in geWorkbench. The ANOVA component itself contains two subtabs: Parameters and Services. The various parameters shown are described in the following section.

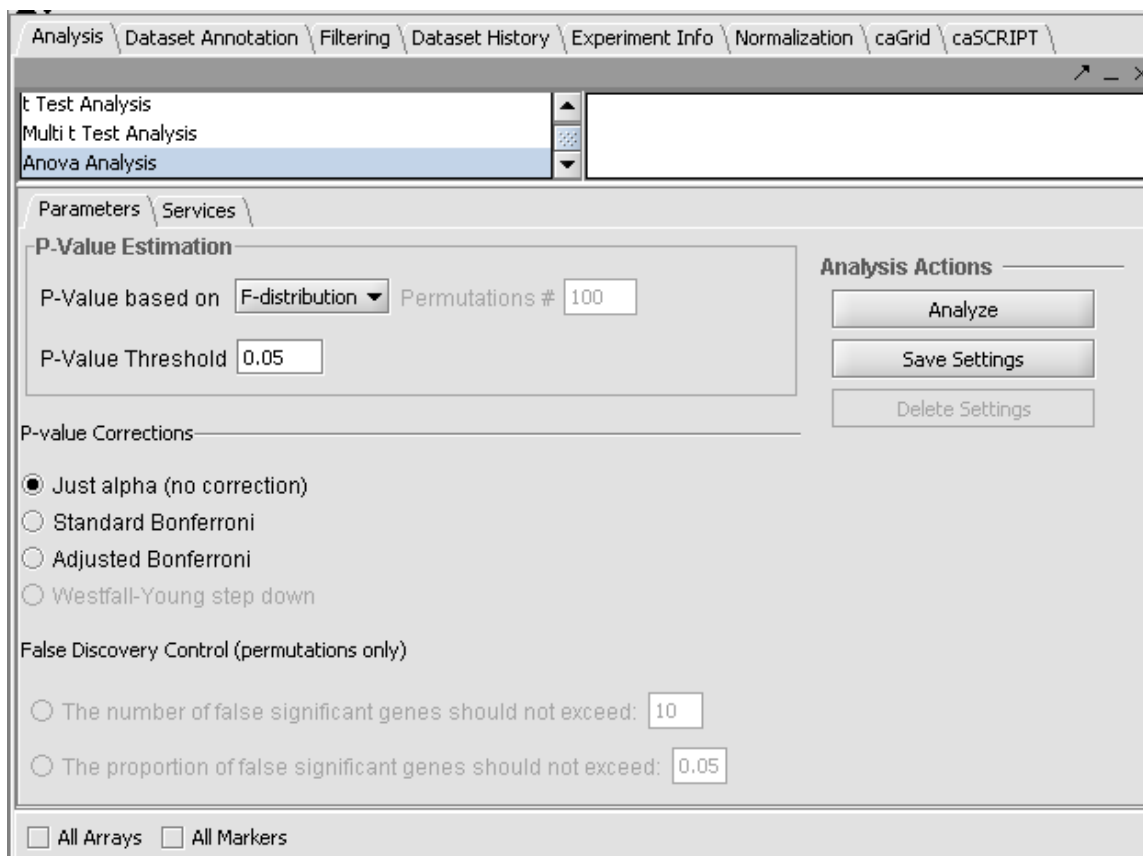
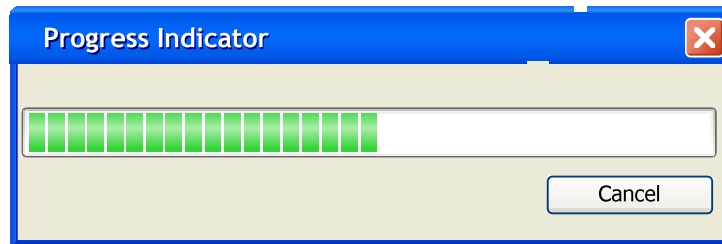


Figure 9-1 The ANOVA component

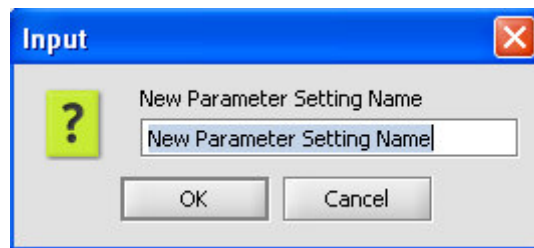
## 9.4 Parameters Subtab

- P-value Estimation: The p-value can be estimated either using the F-distribution or using a permutation method. These two choices are available as pulldown menu items.
  - P-Value Estimation based on F- distribution
  - P-Value Estimation based on Permutations
    - Number of permutations – if permutations are selected, the Permutations # box becomes active.
- P-Value Threshold
  - The user can set the desired critical P-value (alpha) here, by which determinations of significance for a statistic are made. Values from 0 to 1 can be chosen. The default is 0.05.
- P-value Corrections: The p-value can be used directly as a cutoff for determining significant genes, or it can be corrected.
  - Just Alpha (no correction)
  - Standard Bonferroni (the cutoff value (alpha) is divided by the number of tests (genes or markers) before being compared with the p-values).
  - Adjusted Bonferroni . This presumably implements the Holm-Bonferroni correction. For each successive P-value in an ordered list of P-values, the divisor for alpha is decremented by one and then the result compared with the P-value. The effect is to slightly reduce the stringency of the Bonferroni correction.
  - Westfall-Young Step-Down maxT. (Enabled only if permutations have been enabled). This correction can be used when it is believed the individual tests are not independent – note that in a gene expression experiment, groups of genes with correlated expression are to be expected. It uses the test statistic T rather than a P-value as the value that is corrected. This method involves using permutations of the data to calculate the distribution of the test statistic.
- False Discovery Control: This is enabled (only) when p-values based on permutation have been selected.
  - The user must select the false discoveries in terms of either
    - maximum number of falsely discovered significant genes (an integer), or
    - the proportion of falsely discovered significant genes (a decimal fraction).
  - The confidence of the above calculation will be specified by the value of alpha chosen in the parameters above (p-value cutoff).
- Analysis Actions
  - Analyze – launch the ANOVA calculation When the Analyze button is hit, a progress bar (Figure 9-2) may appear indicating that the calculation is continuing.



**Figure 9-2 Run Progress Indicator**

- Save Settings – Pressing this button will bring up a text entry box (Figure 9-3) that will allow the current settings to be saved as a named list. The saved settings are stored in a list at the top right of the Analysis component. From that list, any desired saved parameter set can be chosen.



**Figure 9-3 New Parameter Setting input**

## **9.5 Services Subtab**

The Services subtab (Figure 9-4) allows the user to select which ANOVA compute server to use. There are two types of compute servers options supported: Local and Grid. Local runs the local code directly within geWorkbench; Grid allows the user to pick from among any available caGrid-enabled versions of ANOVA. Further information about using caGrid services in geWorkbench is available in Section 10.1



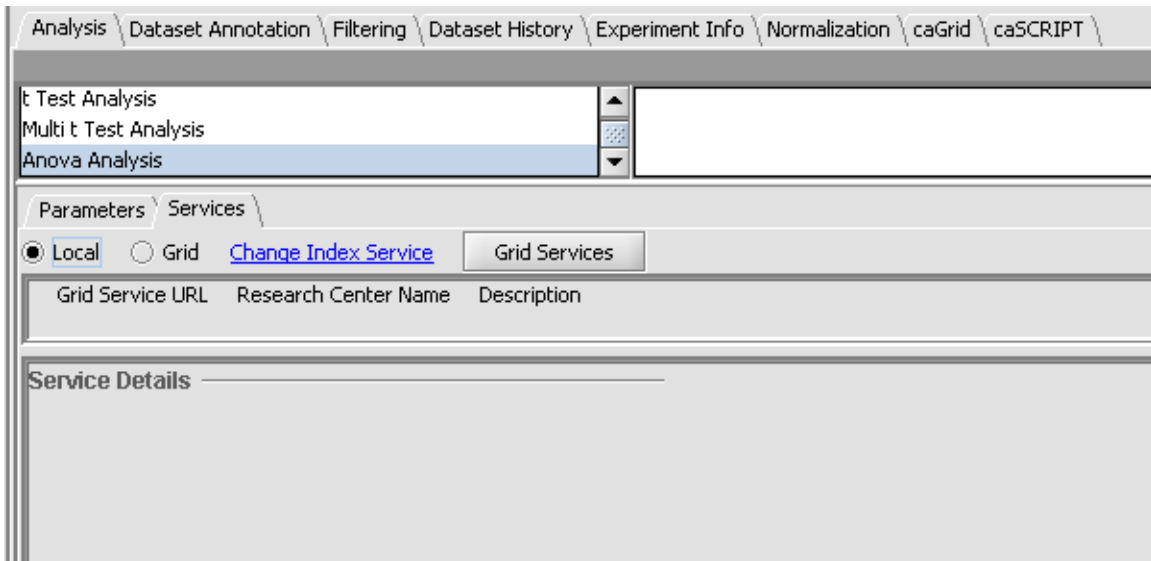


Figure 9-4 Services Subtab

## 9.6 Outputs and Visualization

### 9.6.1 Project Folders

If the ANOVA analysis is successful, the project panel is updated to reflect the addition of an ANOVA results data node (Figure 9-5). This node contains a list of markers that have met the significance criterion, along with their p-values and other information. The list of significant markers is also saved as a new group in the Markers component.

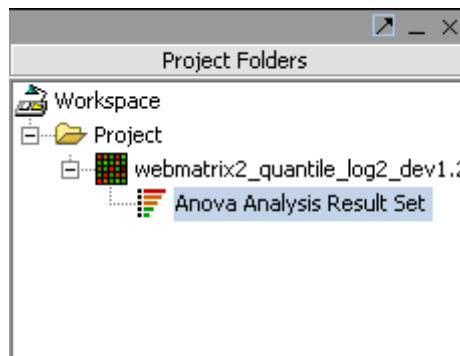


Figure 9-5 An ANOVA result set in the Project Folders component

## 9.6.2 Tabular Viewer

The contents of the result node can be viewed in the ANOVA Tabular Viewer component (Figure 9-6).

Marker Name	P-Value	F-statistic	GC Tumor_Mean	GC Tumor_Std	non-GC B-cell_Mean	non-GC B-cell_Std	GC B-cell_Mean	GC B-cell_Std	non-GC Tumor_Mean	non-GC Tumor_Std
IGL@	0.00E+00	46.19	7.28	1.41	10.81	0.54	10.39	0.52	7.01	1.50
FCGR1	0.00E+00	44.13	7.89	1.10	8.57	0.60	6.16	0.98	9.47	0.84
TNIP	0.00E+00	73.74	9.91	1.07	12.01	0.32	9.40	0.65	12.11	0.64
TUBB2C	0.00E+00	51.76	11.25	0.69	10.22	0.59	12.13	0.41	9.38	0.95
KIF14	0.00E+00	51.11	8.49	0.99	6.45	0.92	9.69	0.25	5.52	1.58
RPL39L	0.00E+00	51.58	6.14	1.01	4.63	0.49	7.18	0.38	4.22	0.90
CR1	0.00E+00	58.19	4.91	1.41	8.37	0.88	7.58	0.73	4.45	1.09
CDC45L	0.00E+00	59.53	7.44	0.67	5.02	1.32	7.97	0.32	4.58	1.26
CD1C	0.00E+00	46.22	9.18	1.79	10.94	0.34	8.84	0.42	7.08	0.98

Figure 9-6 ANOVA Tabular Viewer

The Tabular Viewer presents a read-only spreadsheet of the significant genes sorted by p-value in ascending order (from most significant to least significant). This table is exportable in .csv format. In this view, the columns displayed can be altered in the **Display Preference** window (Figure 9-7), and the display can be sorted by the values in any column. The columns available for display are:

- Marker Name: The marker name display is the minimum length to ensure uniqueness.
- F-Statistic:
- P-Value:
- For each Marker group:
  - Mean: Displays the mean value of the group.
  - Standard Deviation: Displays the mean value of the group.

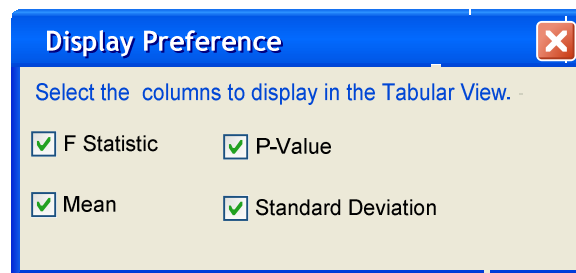


Figure 9-7 Tabular Viewer Display Preferences

## 9.6.3 Color Mosaic

The Color Mosaic (Figure 9-8) presents a heat map, with grouped arrays across the top in the columns, and individual markers in the rows.

Display – pressing this button draws the Color Mosaic display  
Accession – turns on or off display of marker accession numbers  
Label – turns on or off the display of the marker labels.

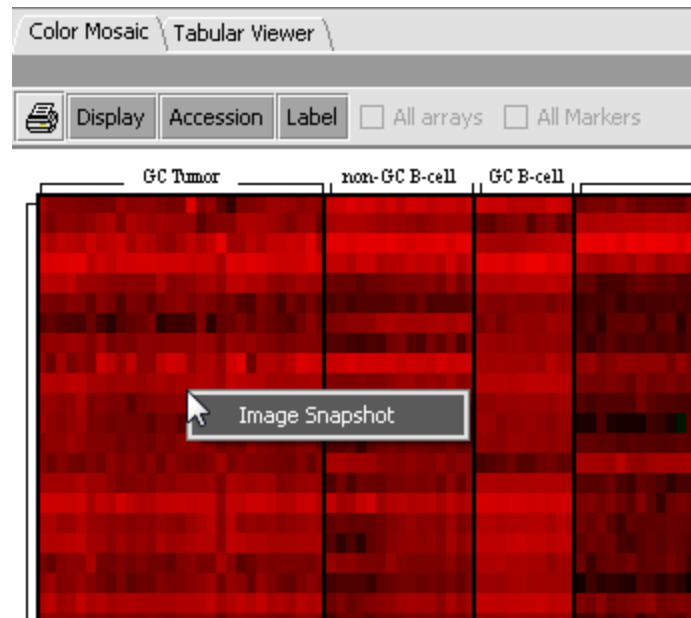


Figure 9-8 Color Mosaic display of ANOVA data.

Right-clicking on the Color Mosaic depiction brings up an “Image Snapshot” button. This will store a picture of the color mosaic view in the Project Folders component.

## 9.7 Dataset History

Complete information about the ANOVA run is displayed in the Dataset History component.

The following is captured in the Dataset History of the ANOVA result node. Each input parameter must be recorded as well as activated marker groups and the markers in each group.

For example:

Generated with ANOVA run with parameters.

-----

P Value estimation: F-Distribution

P Value threshold: 0.05

correction-method: bonferroni

Group Name 1

Member a

Member b

Member c

Group Name 2

Member a

Member b

Member c

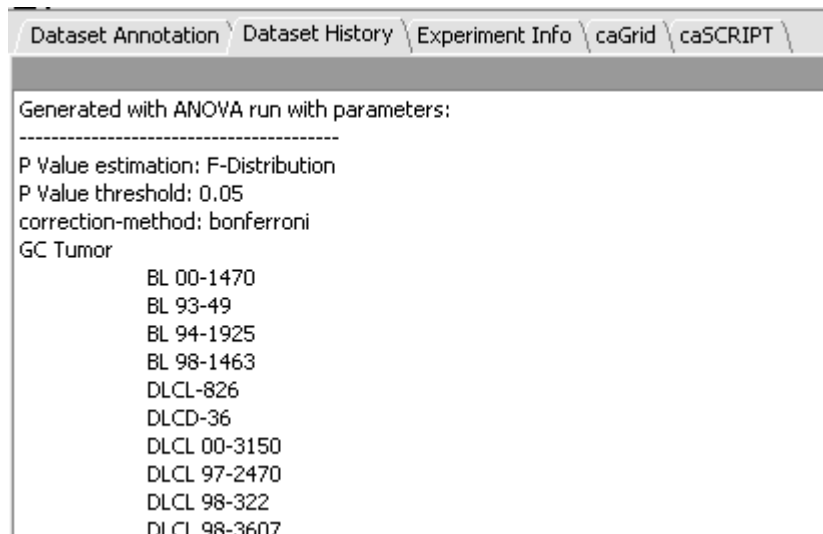
Group Name 3

Member a

Member b

Member c

A portion from an actual example is shown in the screenshot below (Figure 9-9).



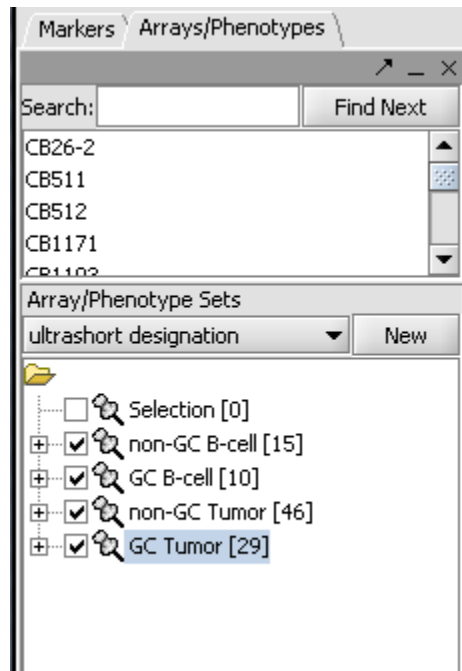
**Figure 9-9 Partial Dataset History from an ANOVA run.**

## **9.8 Running an ANOVA job**

### **9.8.1 Setup**

- An expression file must be loaded into the Project Folder.
- Microarray data should be log-transformed, so as to provide a dataset with an approximately normal distribution.

- Three or more array groups must be defined and activate activated in the Arrays/Phenotypes component (Figure 9-10).

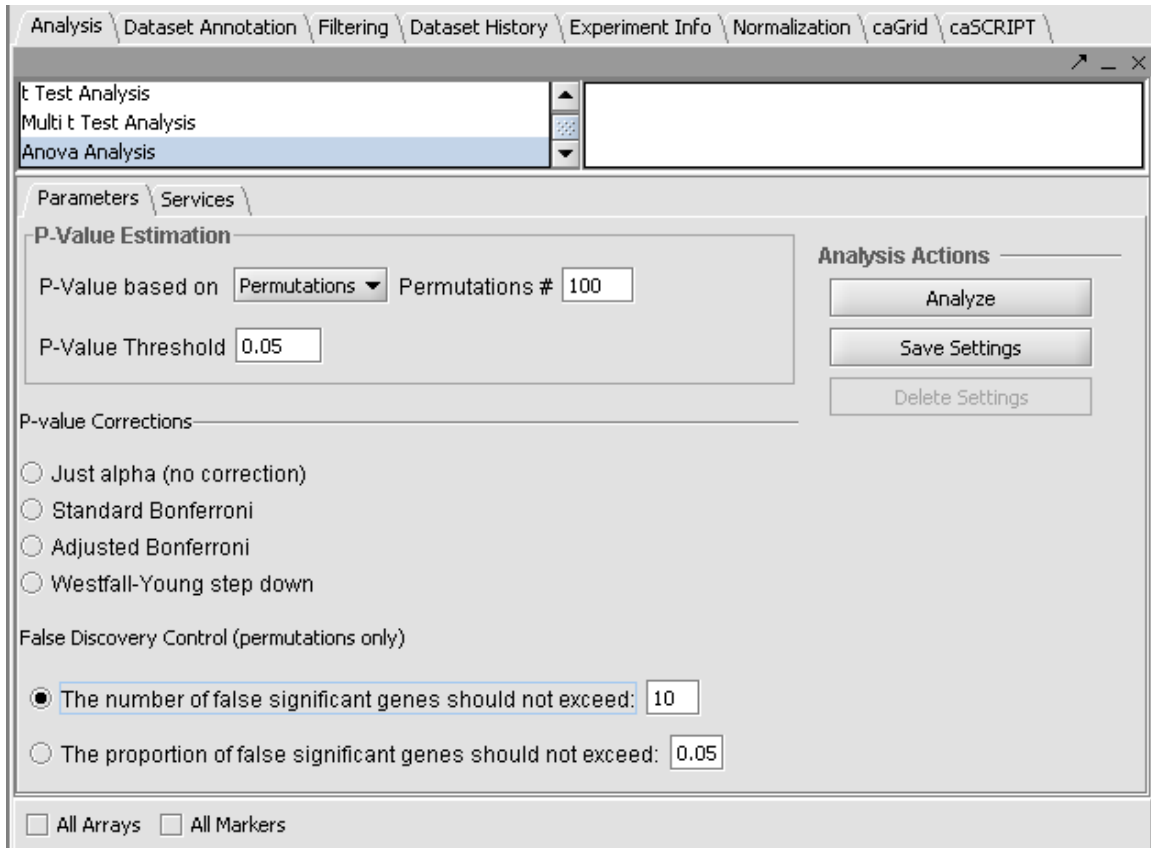


**Figure 9-10** Array set selection for ANOVA

- Marker groups may be defined and activated in the Markers component of the geWorkbench framework. If one or more marker groups is activated, only those markers will be used in the analysis – however, if the All Markers check box is selected, all markers will be used. If no marker sets are activated, all markers are used by default.

### 9.8.2 Choose Settings

The parameters for the particular type of ANOVA run desired should be chosen. For example, one may wish to perform a run with False Discovery Control. To enable this option, the P-Value estimation method must be set to Permutations (Figure 9-11):



**Figure 9-11 Setting up ANOVA for False Discovery Control**

You may also choose to run the calculation locally (the default) or using a caGrid based server on the Services tab.

The settings can be saved by clicking the **Save Settings** button.

### **9.8.3 Run the calculation**

The ANOVA calculation is launched by hitting the **Analyze** button. Results will be displayed in either the Tabular Viewer or Color Mosaic components, as described above.

## **9.9 References**

Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A, Kostukovich E,

Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J. TM4: a free, open-source system for microarray data management and analysis. *Biotechniques*. 2003 Feb;34(2):374-8.

[http://www.tm4.org/documentation/TM4\\_Biotechniques\\_2003.pdf](http://www.tm4.org/documentation/TM4_Biotechniques_2003.pdf)

Yongchao Ge, Sandrine Dudoit, and Terence P. Speed. Resampling-based multiple testing for microarray data analysis. Technical Report 633. Department of Statistics, University of California, Berkeley. <http://www.stat.berkeley.edu/tech-reports/633.pdf>

# 10 Using caGRID Analytical Services

This chapter describes how geWorkbench can be used to dispatch jobs to remote servers using the caGRID infrastructure. Three services which have been implemented are Hierarchical Clustering, Self Organizing Maps, and ARACNE.

Topics covered in this chapter include:

- Using caGRID-based remote analytical services
- Hierarchical Clustering
- Self-Organizing Maps (SOM)
- ARACNE

## ***10.1 Using caGRID-based remote analytical services***

Three microarray analysis routines already available within geWorkbench have been deployed as caGRID analytical services: Hierarchical Clustering, Self-Organizing Maps (SOM), and ARACNE. The purpose is to remove large-scale calculations from the user's desktop machine, instead running them on appropriately scaled server systems. The remote systems could scale to cluster computers or other major hardware platforms as demand necessitates. This could be of particular interest for jobs requiring or benefiting from parallel programming, large memory, or which have long run-times. caGRID provides a standardized way to develop, deploy, and interact with these remote services.

Information about the hierarchical clustering and SOM routines is available in either or both of the geWorkbench User Manual and the tutorials available on <http://www.geworkbench.org/>. This material will not be repeated in detail in this supplement, as the only new feature is the availability of remote execution over caGRID. The material on ARACNE however is new.

This chapter will primarily focus on invoking the grid-based services. Use of all three routines begins with loading a microarray dataset. An example of doing so has already been provided in Chapter 3, using the supplied data file `web100.exp`. Such datasets are two-dimensional, in that typically results from several experiments (chips) have been merged into a single data array, with genes on the vertical axis and the individual experiments on the horizontal axis when viewed in spreadsheet format.



## 10.2 Hierarchical Clustering

Hierarchical clustering is a method used to group data based on a measure of similarity. The two-dimensional microarray datasets used in geWorkbench can be clustered based on expression profiles for genes, for single arrays, or both. For example in clustering by gene, if the expression pattern for gene A across all experiments is very similar to that of gene B, then A and B will tend to cluster together.

### Example

1. Load a microarray dataset, for example `web100.exp` as shown in Chapter 3 of this manual.
2. In the **Array/Phenotype** component at lower left in the GUI, select which arrays should be used in the analysis (Figure 10-1). Here we have activated all the arrays by checking the boxes next to each group. (The same could be accomplished by not checking any boxes, since the default is to include all arrays if nothing is selected).

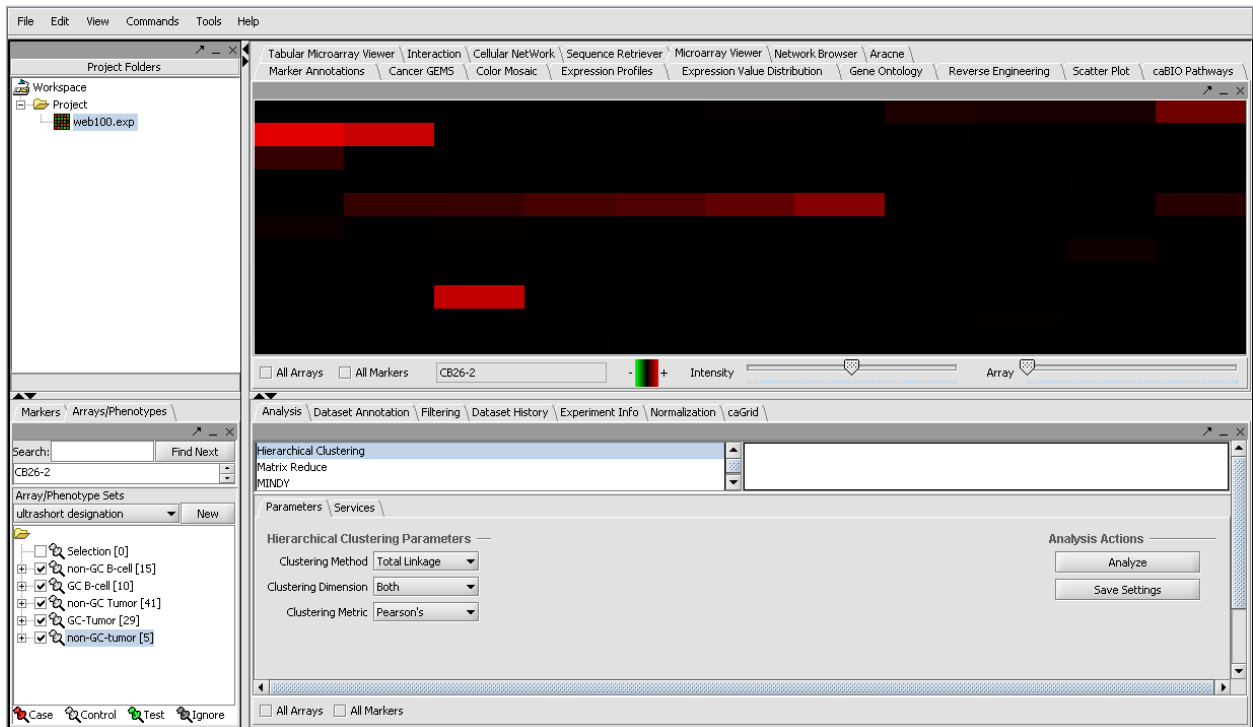
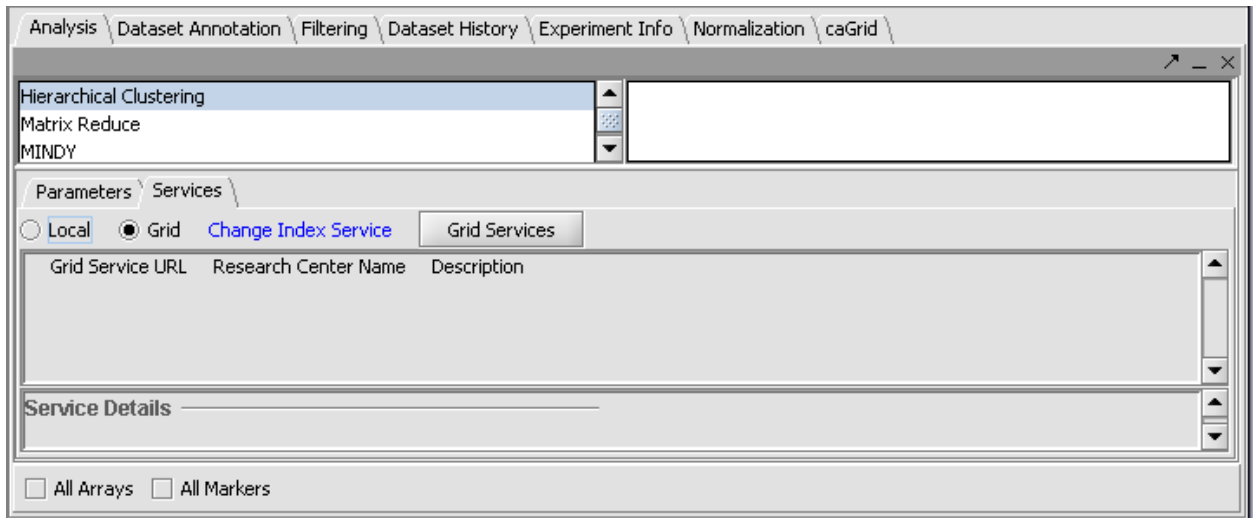


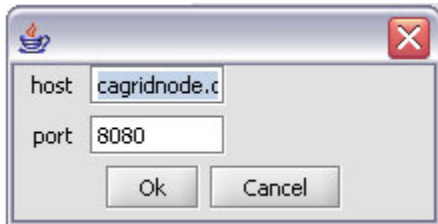
Figure 10-1 Activating the array sets

3. In the **Analysis** tab at lower right in the GUI, select **Hierarchical Clustering**. Select the **Services** tab (Figure 10-2). Click on the blue text **Change Index Service**. This will bring up a small input box.



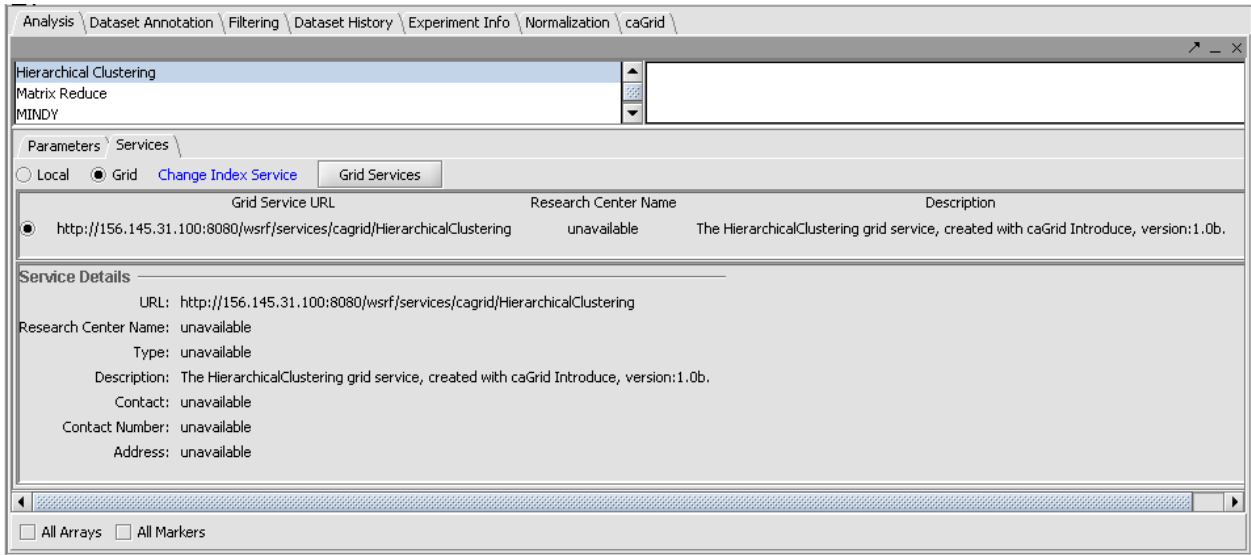
**Figure 10-2 Setting the Grid Index Service**

4. For **host**, (if it is not already present) enter `cagridnode.c2b2.columbia.edu` (Figure 10-3). Leave the port set to 8080. Press **OK**.



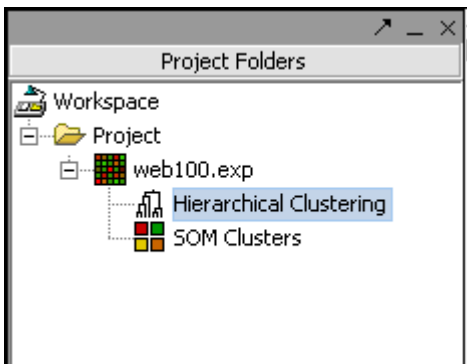
**Figure 10-3 Adding a new grid node**

5. Pressing **Grid Services** will display the available services (Figure 10-4). Select the available **Hierarchical Clustering** service. Once a service is selected, its details will display in the **Service Details** box below.

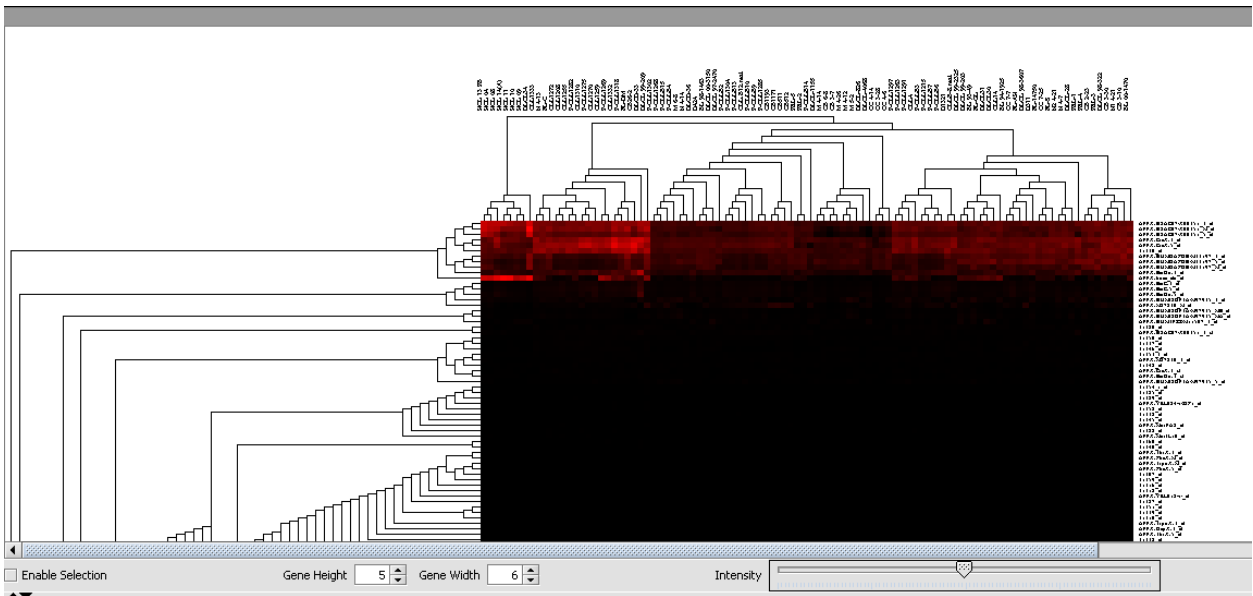


**Figure 10-4 Selecting an available Hierarchical Clustering service**

6. Now go back to the **Parameters** tab (see Figure 10-1 above) and set the following parameters for this example clustering operation:
  - a. Clustering Method: **Total Linkage**
  - b. Clustering Dimension: **Both**
  - c. Clustering Metric: **Pearson's**
7. Click **Analyze**. The computation is carried out on the remote server and returned to geWorkbench, where the results are entered into the Project as a new data node (Figure 10-5), and displayed in the **Dendrogram Viewer** component (Figure 10-6).



**Figure 10-5 Result sets displayed in the Project Folder**



**Figure 10-6 Hierarchical Clustering Dendrogram display**

8. Within the Dendrogram display, the data can be manipulated in many ways, and clusters of genes can be selected and stored as named sets in the **Marker** component for further analysis (see the Clustering tutorial on <http://www.geworkbench.org/> for details).

### **10.3 Self-Organizing Maps (SOM)**

The SOM algorithm is used to divide a data set into a predetermined number groups based on similarity. Here we illustrate running SOM through the grid interface.

#### **Example**

1. Start as in the previous example by loading a microarray dataset, such as `web100.exp`. If it is already loaded, there is no need to reload it.
2. In the **Analysis** tab, select **SOM**.
3. Changing the Index Service, which should not be necessary, has been described above under Hierarchical Clustering.
4. Select an available **SOM** service. Once selected, the details will display in the area below it (Figure 10-7).



**Figure 10-7 SOM Grid Services**

5. We will accept the default parameters, except setting the **Function** to **Gaussian** instead of Bubble (Figure 10-8):
  - a. **Rows:** 3
  - b. **Columns:** 3
  - c. **Radius:** 3
  - d. **Iterations:** 4000
  - e. **Alpha:** 0.8
  - f. **Function:** Gaussian

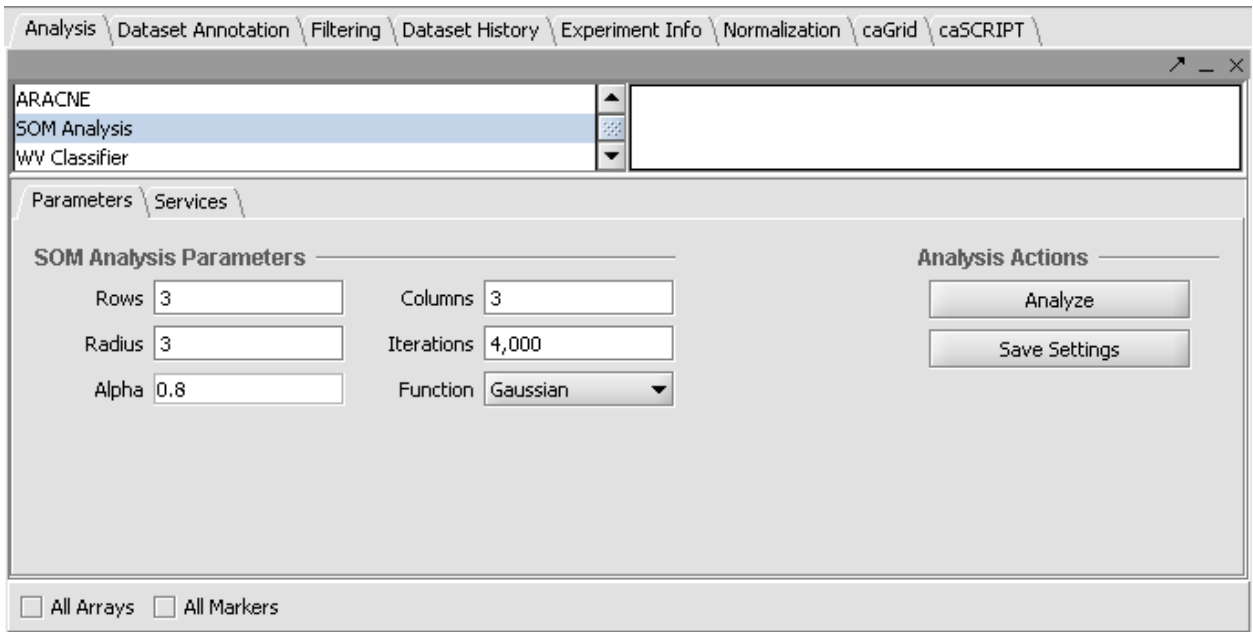


Figure 10-8 Setting SOM parameters

- Click **Analyze**. The result will be returned from the remote server and displayed in the SOM Viewer component as a 3x3 array of graphs. Each contains a set of expression profiles that are, within the boundaries established by the parameters chosen, similar to each other. Several clearly different groups can be distinguished(Figure 10-9)

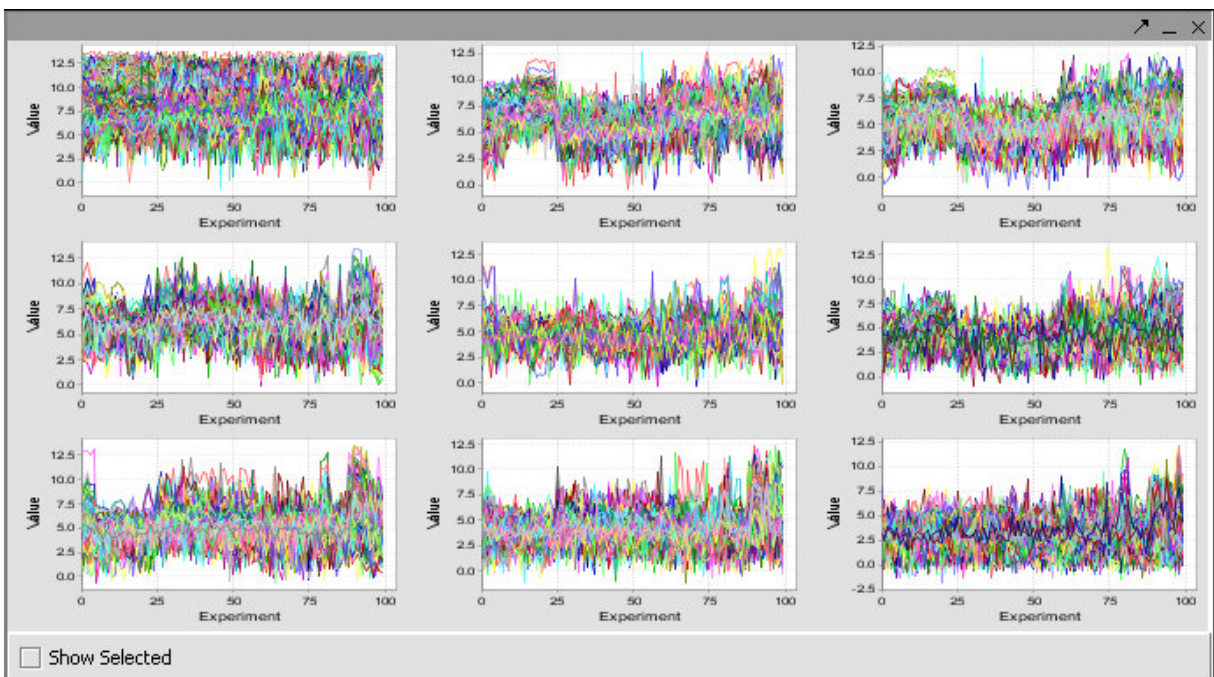


Figure 10-9 SOM Viewer

7. The genes represented in any given graph can be selected and returned to the Markers component for further analysis.
8. The result is also placed in the **Project Folders** component of the GUI beneath its parent data set. In this figure, both a hierarchical clustering dataset and a SOM result are present. Either can be viewed simply by selecting it (Figure 10-10) here.

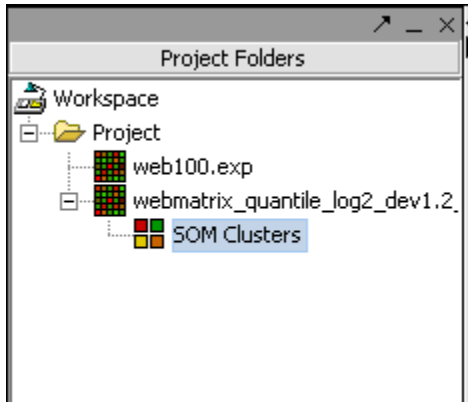


Figure 10-10 Project Folder with results nodes

## 10.4 ARACNE

### Introduction

ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) can be used to infer regulatory interactions within a set of microarray data. Its use has been described in detail in reference Margolin et al. 2006, #1.

The ARACNE algorithm is based on the calculation of mutual information (MI). It has been designed to overcome a number of problems with other methods, for example by allowing calculation on continuous-valued data rather than requiring discretized data, and needing no assumptions about the underlying network topology. The method is not without potential limitations, and the user should consult the references for more information on the usefulness of the algorithm in any particular type of investigation.

ARACNE calculates the mutual information between specified pairs of markers across a set of multiple microarray gene-expression experiments. In an optional second step, an information theoretic property known as the Data Processing Inequality (DPI) (Margolin et al. 2006, #2) is used to attempt to remove indirect interactions, that is, those mediated

by another marker. The algorithm in principal could be used on any type of interaction data, not just gene expression. In geWorkbench however only gene expression data may currently be submitted to the algorithm. The output of ARACNE is an adjacency matrix, showing the strength of interaction for each calculated pair of markers.

### Prerequisites:

There should be at least 100 microarrays in the dataset to allow reliable calculation of the mutual information. However, please note than on such large datasets, only calculation of the mutual information of a particular gene marker with the rest of the dataset should be undertaken on a desktop-class machine. An all-against-all calculation generally requires use of a cluster computer, potentially via the grid service facility if such a service becomes available. Otherwise, ARACNE can be obtained as a stand-alone program for use on a local computational cluster.

The data used should show a significant dynamic range, for example through sampling multiple phenotypes, or through use of experimental perturbations. Uninformative genes, such as those with low mean expression, should likely be filtered out before running ARACNE.

### Understanding the Parameters

There are several choices that can be made as to how ARACNE will be run on a given dataset. Here we will outline what these choices are, and provide an introduction on how to choose appropriate parameters for your own particular case. The figure below (Figure 10-11) shows the ARACNE component within geWorkbench and the parameters which can be set.

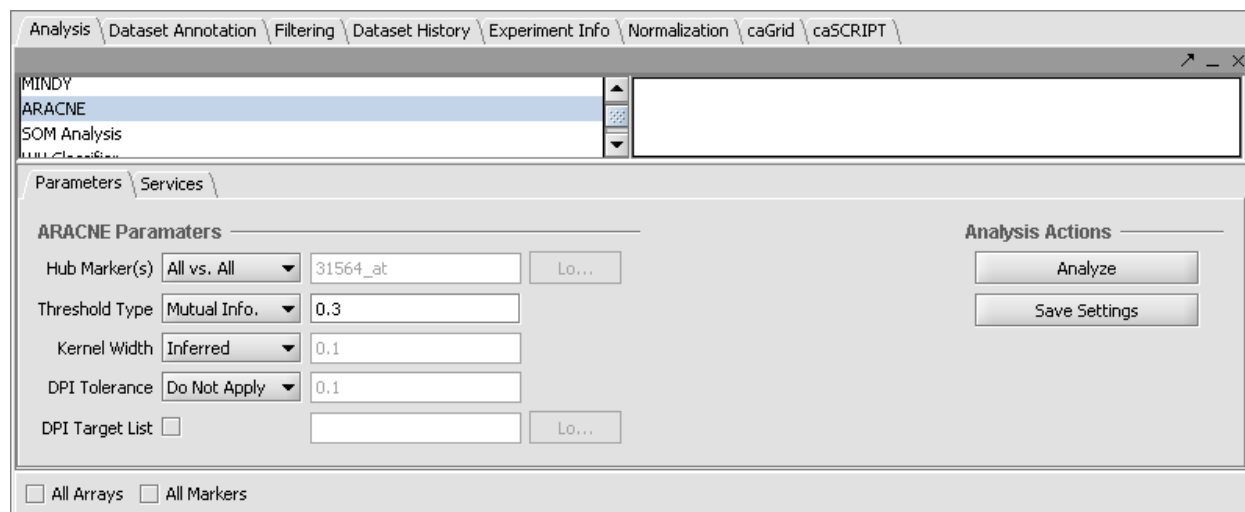


Figure 10-11 The ARACNE component GUI



### **Hub Markers** – Using a hub gene to limit calculations

An ARACNE run can either compute the MI values of every (active) marker against every other, or it can use a list of one or more hub genes to limit the number of calculations required. For large datasets, the all-against-all calculation is not feasible on a local desktop machine, and instead a computational cluster should be used if available as a grid service. If hub genes are used, the MI values are only calculated between each hub gene and all others (or all other markers activated in the Markers component).

- Options:
  - **All vs. All** – Compute the mutual information of each marker against all others (or all markers activated in the Markers component against each other).
  - **List** – Compute the mutual information of each marker in the list against all others (or all those activated in the Markers component).

### **Threshold Type** – P-value or Mutual Information Score

The threshold value above which potentially interacting pairs will be reported can be set either as a p-value or directly as a mutual information score. The mutual information score is more useful when reevaluating an already calculated adjacency matrix, an option present in the standalone version of ARACNE. Margolin et al. 2006, #1 gives a suggested method of choosing a reasonable p-value:

Divide ..” the desired number of false-positives (generally a small integer) by the number of tests performed, calculated as the number of distinct probe pairs. For example, a threshold of  $1e-7$  will lead to about five expected false-positives for a data set with around 10,000 probes, because 10,000 choose 2 (i.e., about  $5e7$ ) probe pairs are tested”.

- Options:
  - **Mutual Info** – use a specified mutual information value, ranging from from 0 to 1.
  - **P-Value** – specify a p-value. This will be converted internally to a mutual information threshold as described above and in Margolin et al. 2006, #1.

### **Kernel Width** – a parameter for the Mutual Information calculation

The parameter used in the mutual information calculation is the kernel width of the Gaussian operator. ARACNE will calculate a default value based on the sample size. The user may also specify a kernel width; methods to calculate an optimal value are given in Margolin et al. 2006, #1. However, for most uses the default value should be adequate, as the algorithm has been shown to be robust with respect to this parameter.

- Options:
  - **Inferred** – accept the default value
  - **Specify** – enter an explicit value.

### **DPI Tolerance** – Removing indirect interactions using DPI

As described in Margolin et al. 2006, #1,

“Many statistical dependencies between gene expression profiles arise from cascades of transcriptional interactions that correlate the expressions of many genes that do not interact directly. ARACNE provides an option to eliminate interactions that are likely to be indirect by applying ...the DPI (described in detail in Margolin et al. 2006, #2). The DPI requires accurate estimation of Mutual Information (MI) ranks; as MI values cannot be estimated exactly with finite data, a tolerance is used to compensate for errors in the estimate that might affect these ranks. Empirically, values between 0 (no tolerance) and 0.15 (15%) tolerance should be used, as larger values tend to cause high false-positive rates”.

- Options:
  - **Do Not Apply** – do not run the DPI calculation
  - **Apply** – run the DPI calculation with the specified tolerance to remove potential indirect interactions.

### **DPI Target List** (checkbox) – Generating a network of transcription factors

If you wish to reconstruct a network only involving transcription factors, you can include a list of such genes (which have been annotated as transcription factors (TFs)) whose interactions are not to be eliminated by DPI in favor of interactions consisting of two non-TFs. “This partially alleviates the problem associated with highly correlated non-interacting genes, such as those involved in stable complex formation, which violate some of the assumptions required for application of the DPI. This feature is described in greater detail in the online Supplementary Manual” (Margolin et al. 2006, #1) (<http://amdec-bioinfo.cu-genome.org/html/ARACNE.htm>).

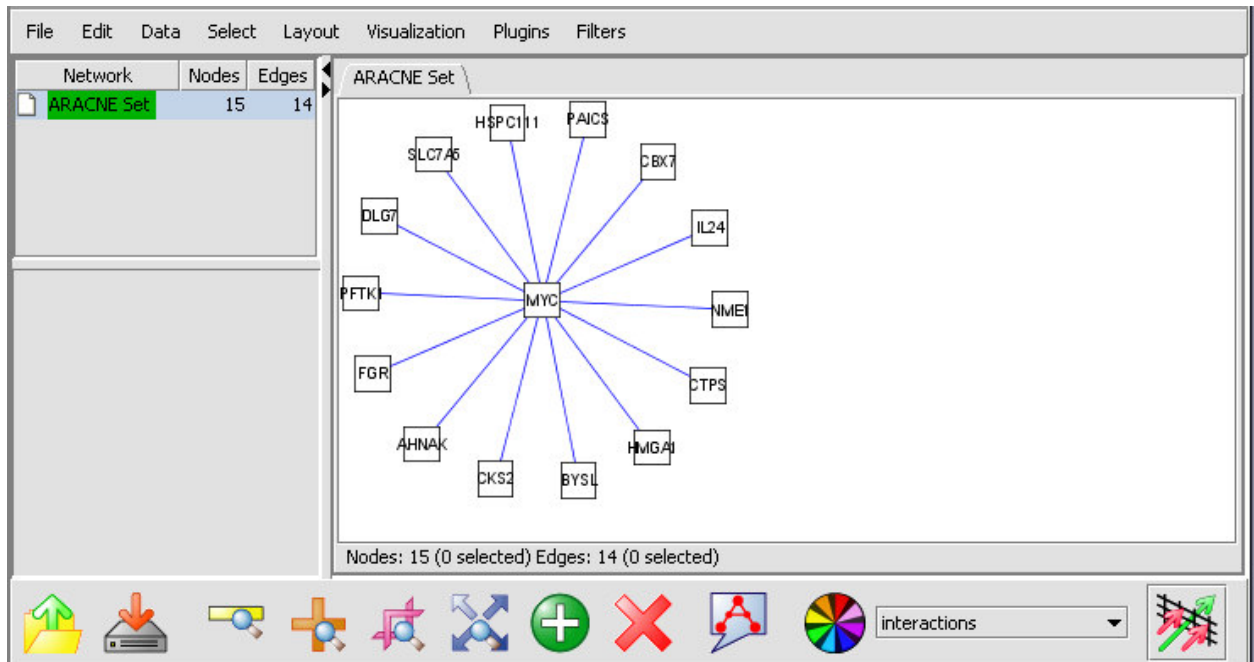
## Running an ARACNE Calculation

### Example

1. A microarray dataset must be loaded in the **Projects** component. Review the Prerequisites section above for details.
2. The **ARACNE** component can be found in the **Analysis** section (tab).
3. Set the parameters depending on the desired type of run as described above in “Understanding the Parameters”
4. If a remote ARACNE grid service is to be used, go to the **Services** tab in the **ARACNE** component and chose the desired service, e.g. as already described above for Hierarchical Clustering.
5. Push the **Analyze** button. When complete, the results will be displayed within geWorkbench in the **Cytoscape** component.

### Viewing the Results

The figure below (Figure 10-12) shows an example of viewing the results of an ARACNE run using a single hub gene. The results are displayed in the **Cytoscape** component. This component has a number of options for controlling how the network graph will be displayed. It is an external component to geWorkbench and full documentation on its use can be found at <http://www.cytoscape.org/>.



**Figure 10-12 Display of ARACNE-generated adjacency matrix in Cytoscape**

## References

- (1) Reverse engineering cellular networks. Adam A Margolin, Kai Wang, Wei Keat Lim, Manjunath Kustagi, Ilya Nemenman & Andrea Califano. (2006) *Nature Protocols* **1**, pp 662-671
- (2) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A. (2006) *BMC Bioinformatics* **7**,S7.

# 11 Using caSCRIPT to automate actions

This chapter describes how caSCRIPT, a scripting language developed for geWorkbench, can be used to automate common or complex tasks in geWorkbench. The caSCRIPT environment includes a visual editor with access to the available methods and variables.

This chapter covers the single topic:

- Using caSCRIPT to automate tasks

## ***11.1 Using caSCRIPT to automate tasks***

geWorkbench has a built-in scripting language, caSCRIPT. This language is similar to Java. The actual language is described in a separate technical document (see the technical manual entitled “geWorkbench\_Technical\_Guide-Grid\_Services.pdf”). It provides direct access to all of the modules in geWorkbench. Scripting allows any sequence of steps in a workflow to be automated, and allows them to be repeated as desired. The example presented here will illustrate how caSCRIPT can be used to execute a caGRID-based SOM calculation. The basics of running a SOM calculation on a caGRID node have already been covered above in 10 .

The caSCRIPT component, shown below in Figure 11-1, contains a default demonstration script. It executes SOM clustering. A line to run Hierarchical Clustering has been commented out, but can be included by simply removing the comment symbol (*//*). For this example the user can paste in a script which will perform hierarchical clustering:

```
void main() {
    // Instantiate project panel and cagrid panel
    module projectWindow projectPanel;
    module expressionFileFilter expFileFormat;
    module cagrid cagrid;
    string urls[1];

    // Load a microarray set
    projectPanel.loadDataSet("data/web100.exp",
expFileFormat);
```

```

datatype DSMicroarraySet mset = projectPanel.getDataSet();

// Get services
string url =
cagrid.getServiceUrl("cagridnode.c2b2.columbia.edu", 8080,
"HierarchicalClustering");
print url;

// Do clustering
datatype DSHierClusterDataSet cluster =
cagrid.doClustering(mset, "Total", "Both", "Pearson", url);
print cluster.getLabel();

// Add cluster to project panel
projectPanel.addDataSetNode(cluster);
}

```

The caSCRIPT component contains three separate areas (Figure 11-1) – the script display at left, a list of available methods at top-right, and an area to display the details of any selected method at bottom right. These aid in constructing new scripts (not covered in this manual).

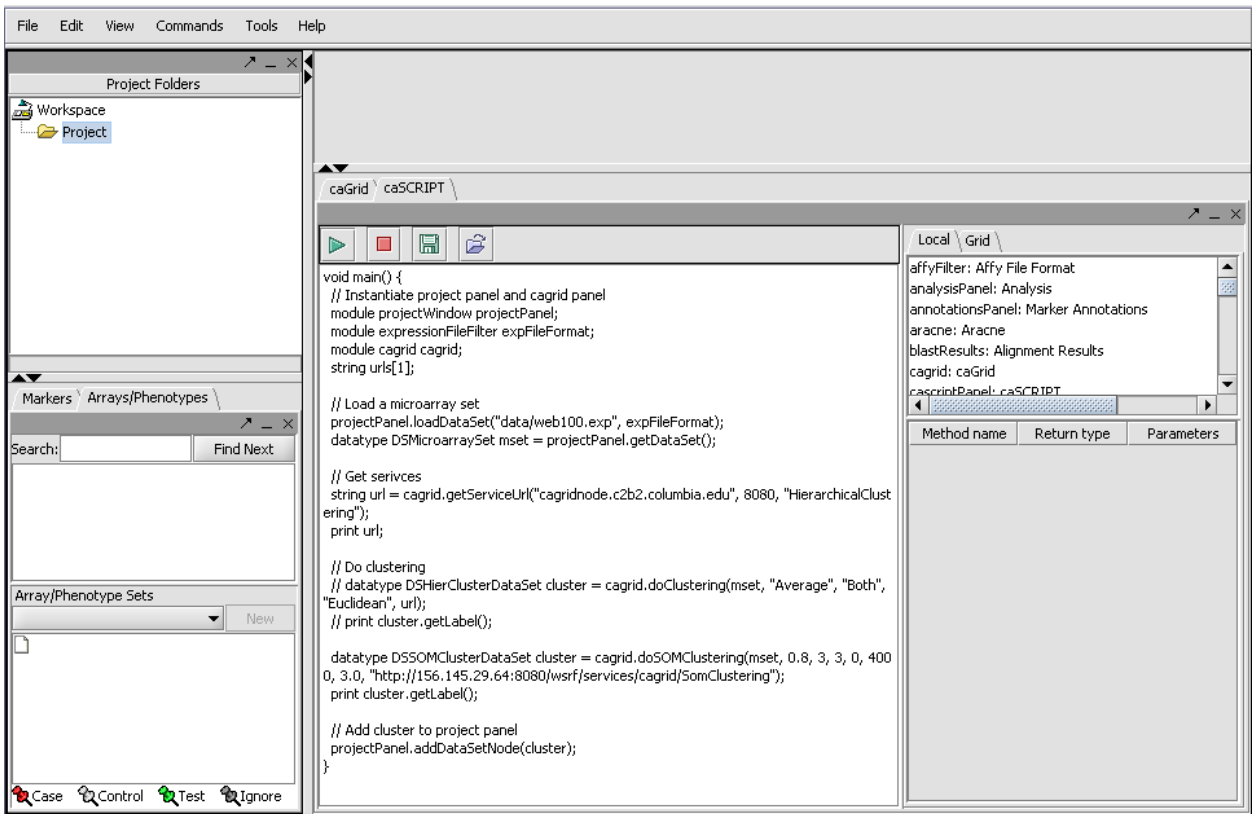


Figure 11-1 The caSCRIPT interface

Figure 11-2 below shows an example of listing a local method, which displays its properties just below it.

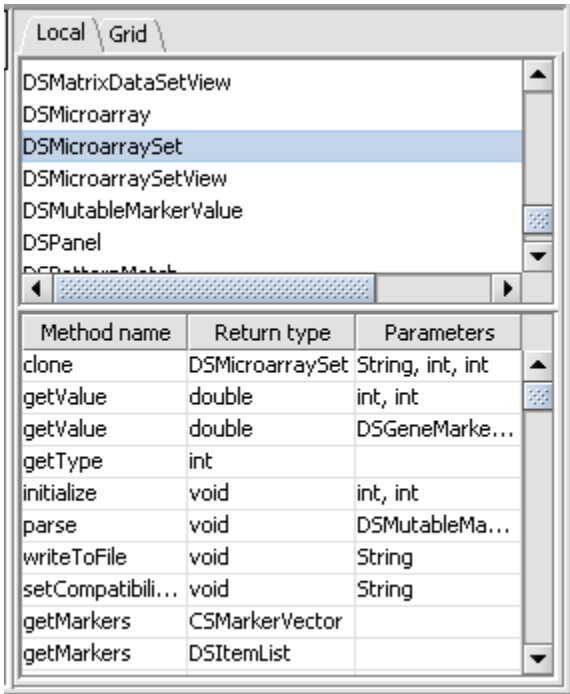


Figure 11-2 caSCRIPT methods selector

The **Grid** tab at right on the caSCRIPT component (Figure 11-3) reveals an interface for changing the **caGRID Index Service** and for discovering the analytical services the chosen node offers. The figure below shows that after the **Discover** button has been pushed, the **Hierarchical Clustering** service is offered. This information could be used in constructing a new script. When a discovered service is selected, its details are displayed in the area below.

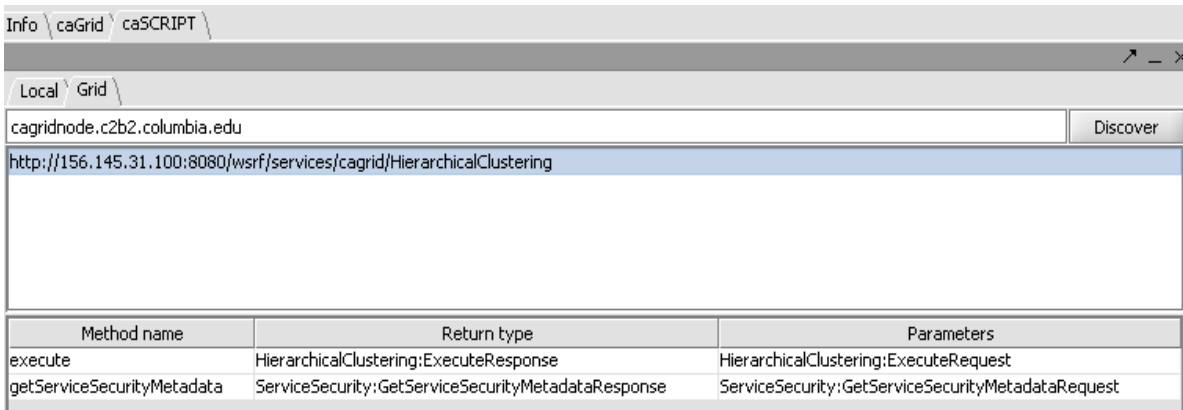


Figure 11-3 Discovering available services



The **caSCRIPT** component has four main controls, shown below in Figure 11-4. They are from left:

**Play** – execute the current script.

**Stop** – stop execution of the script.

**Save** – save the current script to disk.

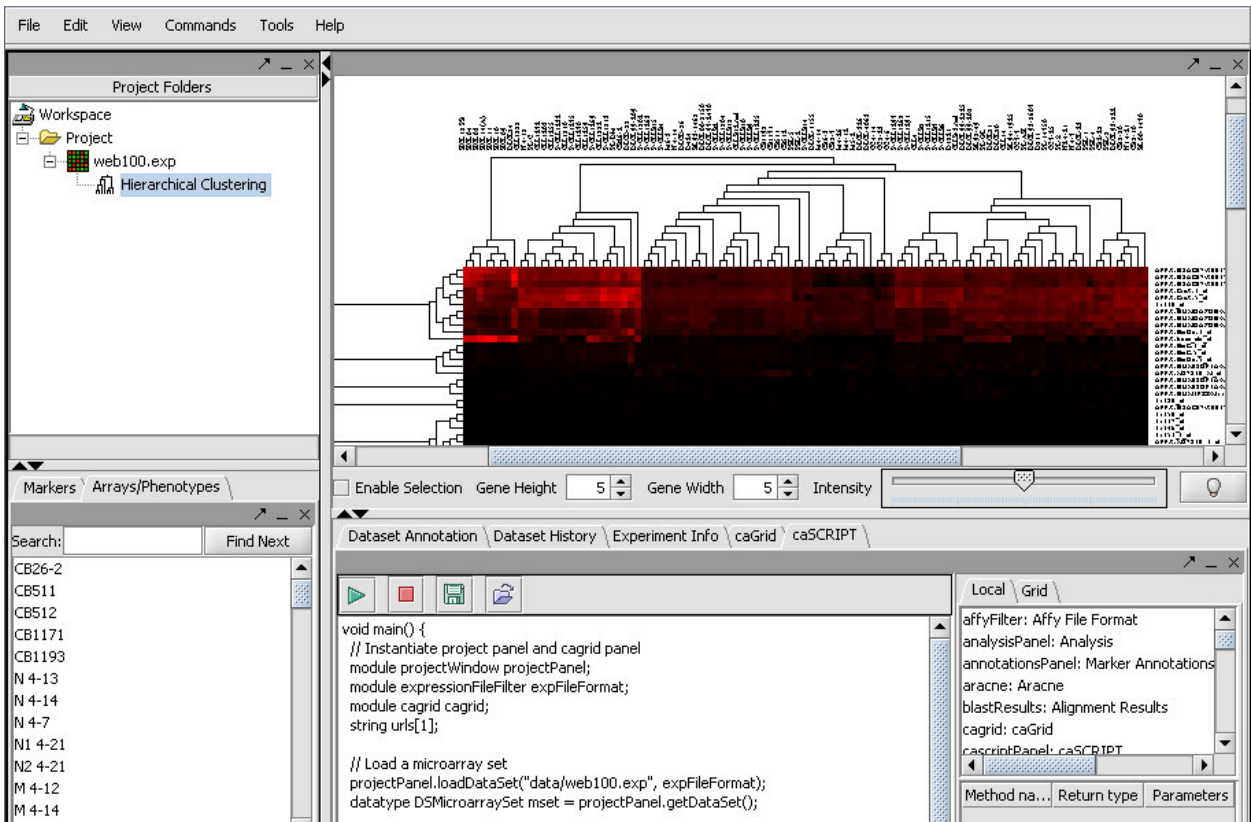
**Open** - opens a file browser to locate a saved script.



**Figure 11-4 caSCRIPT controls**

**Example:**

1. To run the hierarchical clustering example, select and copy the test script shown above into the **caSCRIPT** component.
2. Press the left-arrow shaped **Play** button on the **caSCRIPT** component. The task will be executed on the remote server and the results returned to the **Dendrogram** component, as shown in Figure 11-5 below.



**Figure 11-5 The Dendrogram component displaying results of hierarchical clustering executed using caSCRIPT**

The result of running the default script, which executes SOM clustering, is shown in Figure 11-6 below:

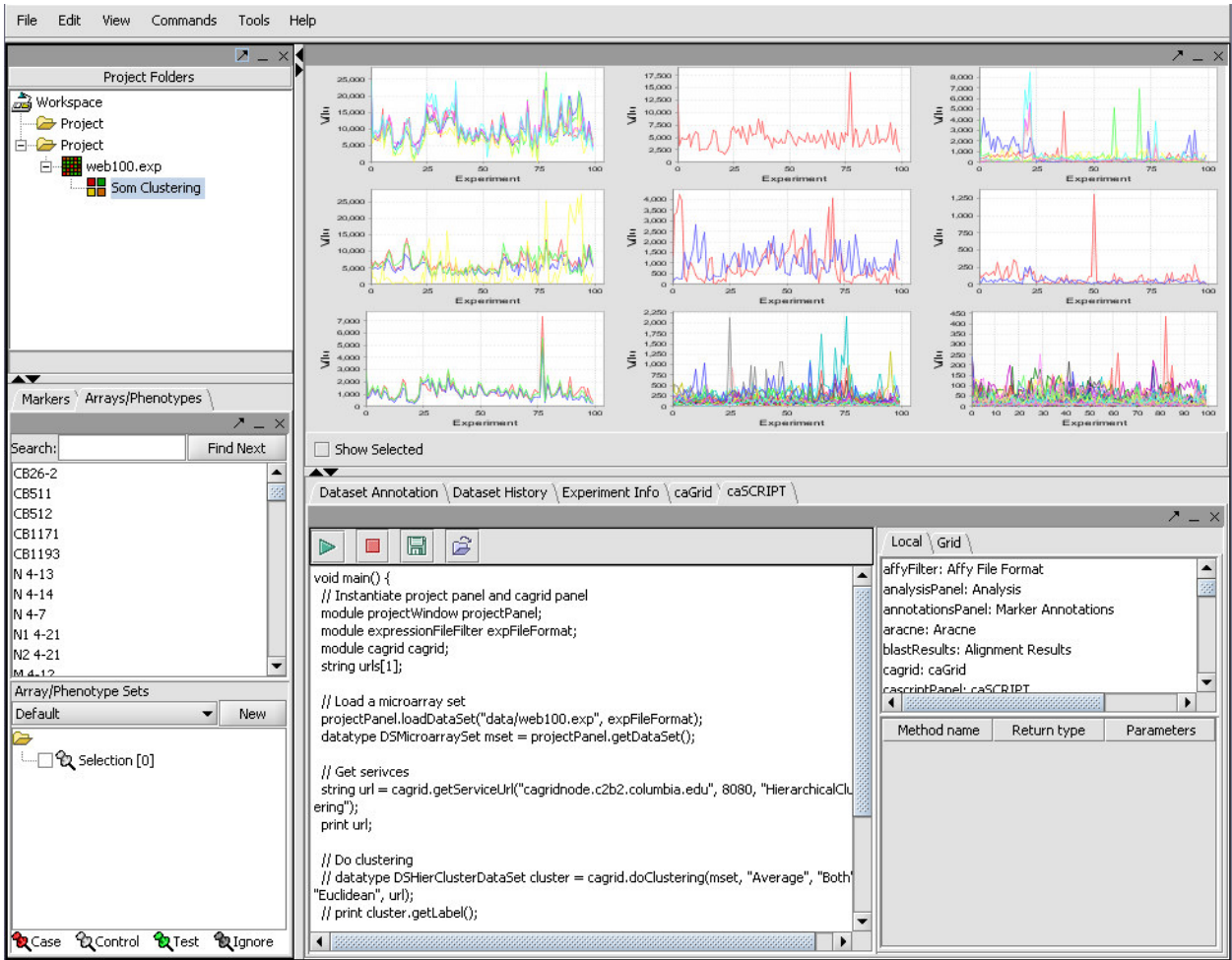


Figure 11-6 Results of SOM clustering example using caSCRIPT

# Appendix A. Error Messages/Indicators and Problem Resolutions

**Why is the desired caGRID service not available?**

Until services are installed in an official caBIG index service, their availability may vary.

**After running a large local calculation, my Windows computer seems slow.**

There are apparently some problems, at least with Windows XP, if the operating system is pushed into swapping – that is, some of the contents of memory are written to disk. Even after geWorkbench has been exited, the slow behavior may persist. In extreme cases, simply reboot the computer. We recommend running geWorkbench in a computer with at least 1 GB of memory, while 2 GB or more will greatly increase the size of calculations possible.

**How do I increase the memory allocated to geWorkbench?**

There is a file in the geWorkbench root directory called UILauncher.lax. There is a line there which specifies the Java heap size:

**lax.nl.java.option.java.heap.size.max=640678989**

Here it is shown set to about 640 MB. You can experiment with increasing this, subject to the amount of memory in your machine and demands on it from other applications.

**Where else can I look for help?**

(Note – this method applies to the packaged distribution version of geWorkbench)

Please see the main geWorkbench website at <http://www.geworkbench.org/>. Of particular interest will be the following sections:

1. FAQs
2. Known Issues
3. Tutorials

# Appendix B. References

## **Scientific Publications**

4. Reverse engineering cellular networks. Adam A Margolin, Kai Wang, Wei Keat Lim, Manjunath Kustagi, Ilya Nemenman & Andrea Califano. (2006) Nature Protocols 1, pp 662-671
5. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A. (2006) BMC Bioinformatics 7,S7.

## **Technical Manuals/Articles**

1. National Cancer Institute. "caCORE 2.0 Technical Guide", [ftp://ftp1.nci.nih.gov/pub/cacore/caCORE2.0\\_Tech\\_Guide.pdf](ftp://ftp1.nci.nih.gov/pub/cacore/caCORE2.0_Tech_Guide.pdf)
2. Java Programming: <http://java.sun.com/learning/new2java/index.html>
3. Extensible Markup Language: <http://www.w3.org/TR/REC-xml/>
4. XML Metadata Interchange: <http://www.omg.org/technology/documents/formal/xmi.htm>
5. **geWorkbench Technical Guide – Grid Services** – This manual is available at the geWorkbench caBIG GForge site in the documentation section: <http://cabigcv.nci.nih.gov/viewcv/viewcv.cgi/caworkbenchcabig/Documentation/>. The manual is titled “geWorkbench\_Technical\_Guide-Grid\_Services.pdf”.

The direct link is

[http://cabigcv.nci.nih.gov/viewcv/viewcv.cgi/caworkbenchcabig/Documentation/geWorkbench\\_Technical\\_Guide-Grid\\_Services.pdf](http://cabigcv.nci.nih.gov/viewcv/viewcv.cgi/caworkbenchcabig/Documentation/geWorkbench_Technical_Guide-Grid_Services.pdf)

## **caBIG Material**

caBIG: <http://cabig.nci.nih.gov/>

caBIG Compatibility Guidelines:

[http://cabig.nci.nih.gov/guidelines\\_documentation](http://cabig.nci.nih.gov/guidelines_documentation)

***caCORE Material***

**caCORE:** <http://ncicb.nci.nih.gov/core>

**caBIO:** <http://ncicb.nci.nih.gov/core/caBIO>

**caDSR:** <http://ncicb.nci.nih.gov/core/caDSR>

**EVS:** <http://ncicb.nci.nih.gov/core/EVS>

**CSM:** <http://ncicb.nci.nih.gov/core/CSM>

## Appendix C. Glossary

Following is a list of terms and their definitions.

<b>Term</b>	<b>Definition</b>
API	Application Programming Interface
caArray	cancer Array Informatics
caBIG	cancer Biomedical Informatics Grid
caBIO	Cancer Bioinformatics Infrastructure Objects
caCORE	cancer Common Ontologic Representation Environment
caDSR	Cancer Data Standards Repository
caMOD	Cancer Models Database
CDE	Common Data Element
CGAP	Cancer Genome Anatomy Project
CMAP	Cancer Molecular Analysis Project
CVS	Concurrent Versions System
EVS	Enterprise Vocabulary Services
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JAR	Java Archive
Javadoc	Tool for generating API documentation in HTML format from doc comments in source code ( <a href="http://java.sun.com/j2se/javadoc/">http://java.sun.com/j2se/javadoc/</a> )
MAGE	MicroArray Gene Expression
MAGE-OM	MicroArray Gene Expression - Object Model
MGED	Microarray Gene Expression Data
MO	MGED Ontology
NCI	National Cancer Institute
NCICB	National Cancer Institute Center for Bioinformatics
SDK	Software Development Kit
SQL	Structured Query Language
UI	User Interface
URL	Uniform Resource Locators