

GenViewer Tutorial / Manual

Table of Contents

Importing Data Files.....	2
Configuration File.....	2
Primary Data	4
Primary Data Format:.....	4
Connectivity Data.....	5
Module Declaration File Format	5
Module Regulator File Format.....	6
Phenotype Regulator File Format	6
Annotation Data.....	6
Annotation File Format.....	6
Annotation Description File Format.....	7
Location Data	7
Marker Location.....	7
Gene Location	7
Visually Browsing the Network.....	7
Filtering by Edge Weight:.....	7
Filtering by Experimental Class.....	8
Filtering by Expression Dataset.....	8
Recentering.....	8
Resetting	8
Navigation & Manipulation.....	8
Zooming:	8
Picking and Transforming:	8
Searching & Navigating.....	8
Search.....	8
Search Results	8
Cluster search results:.....	8
Probe search results:.....	9

Gene search results:.....	9
Ranking Vertices.....	9
Going to Network View.....	9
Going to Cluster View	9
GO Enrichment.....	9

Importing Data Files

Import your data files into GenViewer via a configuration file ([SEE EXAMPLE](#)), which functions as a method to import all of your data.

GenViewer is a visualization program designed to handle data of several different types. These data fall under one of four categories 1) Primary Data 2) Connectivity Data and 3) Annotation Data 4) Location data.

We will use the included mouse dataset as an example, and will refer to it throughout this manual.

Consider the following experimental set-up:

We have two experimental classes of mice (X male mice and Y female mice). For each mouse, we do a microarray study for the liver, adipose, muscle and brain tissue types, resulting in four microarray datasets for each mouse, specifying the levels of expression for a set of probes. In addition, we genotype each of the mice to identify the variations at a set of marker loci, and measure the phenotype values for a series of metrics.

Using an alternate analysis program, [such as Su-In's algorithm](#), we construct a partition of probes into modules, and a regulatory network connecting modules with markers, and phenotypes.

Configuration File

Configuration files have several sections, each of which contains a list of files to load; sections are denoted by a colon suffix. The sections are:

- 1) ExpClasses:
 - a. ExpClasses, a contraction for Experimental Classes represents a way of separating between two different experimental classes within a single data set. In our example, we distinguish between the male and female datasets, M and F respectively.
 - b. FORMAT: In the line following the "ExpClasses:" label, list all experimental classes in tab-delimited fashion


```
>>M F
```
- 2) Genotype:

- a. GenViewer currently only supports a single genotype file. The genotype file describes the alleles for a set of loci and experimental classes.
- b. FORMAT: In the line following the “Genotype:” label, specify the file and its label.

```
>>genotype:  
>>genotype.txt genotype
```

3) Phenotype:

- a. GenViewer currently only supports a single phenotype file. The phenotype file describes measured values for a set of user determined metrics. In our example, these include the weight (in grams), length(in cm), abdominal fat, other fat, total fat, etc.
- b. FORMAT: In the line following the “Phenotype:” label, specify the phenotype file and its label.

```
>>phenotype:  
>>phenotype.txt phenotype
```

4) Expression:

- a. Expression data, most commonly microarray data, specifies measured expression values for a set of probes. We can specify several datasets (eg. Liver, adipose, muscle, brain) and must include unique file labels, which will be used to refer to probes in different datasets. Probe names must be unique within expression files, but can be duplicated between expression files.
- b. FORMAT: In the lines following “Expression:” label, list the expression files and their labels.

```
>>expression:  
>>liver_f.txt liver  
>>adipose.txt adipose  
>>muscle.txt muscle  
>>brain_f.txt brain
```

5) Clusters:

- a. Clusters specify how different sets of probes are grouped together within the regulatory network. Clusters do not overlap (a probe can belong to at most 1 cluster), and clusters can only contain probes from a single expression file (a cluster cannot have probes belonging to both liver and adipose).
- b. FORMAT: In the lines following the “Clusters:” label, list the cluster files.

```
>>clusters:  
>>mems.txt
```

6) Cluster_Regulators:

- a. Clusters can only be regulated by probes and by genotype markers. The cluster regulator files specify the weights for which probes and genotype markers regulate clusters.
- b. FORMAT: In the lines following the “Cluster_Regulators” label, list the cluster regulator files.

```
>>cluster_regulators:  
>>mems.txt
```

7) Phenotype_Regulators:

- a. Phenotypes can only be regulated by clusters. The cluster phenotype regulator files specify the weights for which the clusters regulate phenotypes.
- b. FORMAT: In the lines following the “Phenotype_Regulators:” label, list the phenotype regulator files, one per line.

```
>> phenotype_regulators:  
>>reguls.txt
```

8) Marker_Locations:

- a. Marker Locations specify a location for the markers specified in the genotype file. Only a single Marker Location file can be specified.
- b. FORMAT: In the lines following the “Marker_Locations:” label, list the marker location files, one per line.

```
>>Marker_Locations:  
>>marker_locations.txt
```

9) Gene_Locations:

- a. Gene Locations specify the locations for genes
- b. FORMAT: In the lines following the “Gene_Locations:” label, list the gene location files, one per line.

```
>>Gene_Locations:  
>>marker_locations.txt
```

10) Annotations:

- a. GenViewer currently only supports a single annotation file, common to all of the expression sets. The annotation file specifies the relationship between the probes specified in the expression files, gene names and GO code annotations.
- b. FORMAT: In the lines following the “Annotations:” label, specify the annotation file.

```
>>annotation.txt
```

11) Annotation Descriptions:

- a. GenViewer currently only supports a single annotation description file. The annotation description file specifies a description for each GO term, which allows for easier viewing.
- b. FORMAT: In the lines following the “Annotation Descriptions:” label, specify the annotation description file.

```
>>goTerms.txt
```

All Files are Tab Delimited.

Primary Data

Primary data is the measured values for gene expression (most commonly microarray), phenotype, and genotype data ([SEE EXAMPLE FOR CLARIFICATION](#)). Several expression files can be loaded, but only one genotype file and one phenotype file can be loaded in the same configuration file. All the primary data files share the same format.

Primary Data Format:

Each data file represents a matrix of values representing measured values. The Columns represent experiments, and rows represent the metrics. Expression data files would have values corresponding to measured expression levels corresponding to each probe; Genotype data files would have (0,0.5,1) to enumerate the sequences found at each marker; Phenotype data files would have measured values corresponding to user defined metrics (eg. Weight, length, height, amount of fat, etc.)

Headers:

Header Line 1: The first line specifies the names for each of the experiments in tab delimited form. The first entry is a placeholder.

```
>>ExpNames      1      2      3
```

Header Line 2: The second line specifies the experimental class each experiment belongs to. The first entry is a placeholder.

```
>>ExpClasses    F      M      M
```

Data: The first entry specifies the metric name, and the remaining entries specifies the measured value corresponding to the metric and experiment.

Expression Data:

```
>>10024401266   1.89882 -0.188049 -0.00798
```

Genotype Data:

```
>>mCV22542926  1      0.5    0.5
```

Phenotype Data:

```
>>Weight_g     -0.0609521 -0.675634 -1.2519
```

Connectivity Data

Connectivity Data describes the regulatory network that connects all of the primary data together. Probes from the expression data can be partitioned into modules or clusters (used interchangeably). Modules are described in Module Declaration Files, which assign groupings of probes to a module. Modules can be regulated by probes and genes, and these connections are described in the Module Regulator File. Phenotypes can be regulated by Modules. Each experimental class can have its own independent regulatory network.

Module Declaration File Format

Header: The first line of this file is for humans only, it is ignored by GenViewer.

Data: Each line specifies the probes belonging to each cluster. The first entry of each line specifies a module name. The second entry of each line specifies the label for an expression file. The remaining entries for each line specify the names of the probes belonging to each module. Each module name, expression file label combination must be unique, but modules names can be repeated for different expression files.

```
>>1      liver      10024402862      10024393539      10024407122
```

```
>>2    liver    10024412083    10024414265    10024402824
>>1    adipose   10024399675    10024401143
```

Module Regulator File Format

Header: The first line of this file is for humans only, it is ignored by GenViewer.

Data: Each line specifies the regulators for a particular cluster and experimental class. The entries are as follows:

- 1) Name of module
- 2) Experimental class
- 3) Label of expression file to which the module belongs
- 4) "regulators" key word
- 5) List of probes followed by weights
- 6) "genotypes" key word
- 7) List of genotypic markers followed by weights

```
>>1    F        liver    regulators    10024402750    0.054038 10024400762    -0.414839
        genotypes
>>1    F        liver    regulators    genotypes s    3685837 0.273988 rs3687724 0.131793
```

Phenotype Regulator File Format

Header: The first line of this file is for humans only, it is ignored by GenViewer

Data: Each line specifies the regulators for a unique phenotype. Regulation of phenotypes cannot be separated by experimental class as in the module regulator files. The first entry of each line specifies the name of the phenotype, and the remaining entries represent ordered 3-tuples of expression file label, module name, and weight.

```
>>weight_g    liver    9    -0.0545797    liver    21    -0.0446198    adipose    151
-0.0112608
>>Liver_UC    liver    186    -0.0236161    liver    198    -0.0200897    adipose    13
0.0841958 adipose 121    0.048837 muscle 118    0.0240462 muscle 135    -0.067248
```

Annotation Data

Annotation Data is used to annotate different probes with their GO associations. These annotations are split into two files, the Annotation File and the Annotation Description File. The annotation file describes the relationships between probes and GO annotations. The Annotation Description File assigns a description to GO codes.

Annotation File Format

Header: None

Data: Each line specifies a set of gene names and GO code annotations for a unique probe name. The data is formatted in a tab-delimited list as follows:

- 1) Unique probe name, which may correspond to an entry in an Expression File

- 2) Gene Name
- 3) Gene Name
- 4) Gene Identifier
- 5) List of GO terms

Probes are grouped together based upon the gene identifier in the 4th entry. If no GO names are available, then “None” should be used as a placeholder. GO terms must be expressed in the format “GO:XXXXXX”.

>> 10024406979	ri 4831422J22 PX00101P24 4685	A930038C07Rik	68169	GO:0005576
GO:0005575	GO:0008372	GO:0003673	GO:0005615	GO:0044421

Annotation Description File Format

Header: None

Data: Each line specifies a description for a unique GO term. These descriptions will be used to make the data more readable. **Annotation Description Files are made available on the website for recent GO files.** The first entry of each line specifies the GO term, and the remainder of the line is used as the description

>>GO:0000001	Process	mitochondrion inheritance
>>GO:0000002	Process	mitochondrial genome maintenance
>>GO:0000003	Process	reproduction

Location Data

Marker Location

Gene Location

Visually Browsing the Network

The Network View (show below) provides a visualization of the connectivity data. Vertices within the graph describe modules, phenotypes and genotypes, while the edges represent the aggregated regulator weights. For example, if five probes in module 1 regulate module 2, this is represented as a regulation of module 1 by module 2 with weight equal to the sum of the individual weights.

To facilitate browsing of large / dense networks, the graph can be manipulated in several ways, discussed below.

Filtering by Edge Weight:

The regulatory network can often be large and complex, making the layout difficult to interpret or use. We can filter out edges based upon specific criteria via the Navigation Panel. We can set maximum, minimum, minimum absolute value for the edges. Vertices in the graph that have no edges in or out are removed from the graph, often making everything much simpler. The checkbox beside the weights is used to determine whether we should filter by the associated edge criteria.

Filtering by Experimental Class

Different experimental classes can have different regulatory networks. We can turn these networks on and off via the controls in the graph panel.

Filtering by Expression Dataset

Expression datasets do not overlap; inclusion of vertices associated with expression datasets can be toggled via the controls in the graph panel.

Recentering

Even with filtering options, there can be too many nodes and edges for a graph to be easily interpretable. We can focus on a specific part of the network by selecting a specific set of nodes and clicking “Recenter”. This will present a graph containing only the nodes and edges that are immediate upstream or downstream neighbors of the selected nodes. The number of steps away from the selected set of nodes can be changed via the “Upstream Neighbors” and “Downstream Neighbors” options in the graph panel.

Resetting

Resetting is a way of returning to the global view. This zooms the graph all the way out.

Navigation & Manipulation

Layout of the nodes is not always ideal, especially for large or complex graphs. Click and drag a node to move it around.

Zooming: The scroll-wheel can be used to zoom in and out of a graph

Picking and Transforming: There are two modes that can be used to manipulate the graph. In picking mode, the closest node is selected and dragged. Use this mode to rearrange graphs. In Transforming mode, clicking and dragging will move the graph. This is useful if the graph is zoomed so only a portion of it is visible.

Searching & Navigating

Search

Large graphs can be cumbersome to manipulate. If the name of a node is known beforehand, it can be found by using the search functionality. Currently the items that will be searched include: name of probe, genotypic marker, or phenotype, name of gene and the name of cluster.

Search Results

Clicking on a search result will bring up panel containing more information regarding the selected item.

Cluster search results: Selecting a cluster search result will bring up a panel listing cluster members, any upstream or downstream regulators, and it’s GO enrichment. In order to see GO enrichment, GO analysis must first be performed on the associated dataset. ([SEE GO ENRICHMENT](#)).

Probe search results: Selecting a probe search result will bring up a panel showing It's name, associated genes, associated cluster and dataset.

Gene search results: Selecting a gene search result will bring up a list of associated probes.

Ranking Vertices

It can be helpful to identify nodes that are particularly “well connected” within the entire graph. Currently, we only support ranking of nodes on a global scale based upon the total number of in and out edges.

Going to Network View

Selecting Network will automatically open a network frame if one is not open already, and center it on the selected nodes. This is the fastest way to identify a cluster.

Going to Cluster View

Selecting “Cluster View” will open a cluster view for each selected cluster.

GO Enrichment

Selecting GO enrichment from the menu-bar will calculate GO enrichments for clusters. False Discovery Rate (FDR) correction is used to correct for false positives due to the large number of hypotheses. Users can specify the datasets to enrich, minimum GO cluster size, maximum GO cluster size, minimum FDR P-Vale, and FDR alpha. FDR values are calculated independently for each selected dataset.