

Practical Bioinformatics

Mark Voorhies

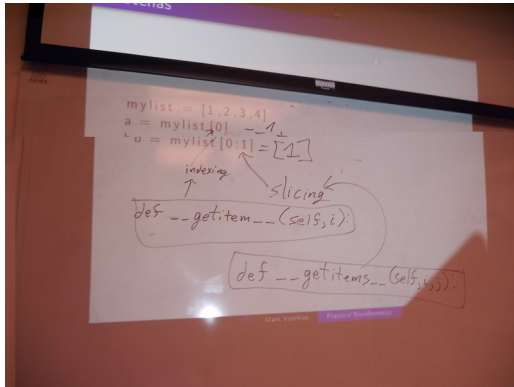
5/23/2013

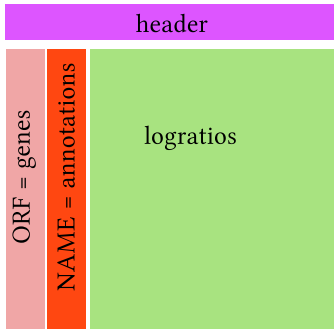
```
mylist = [1,2,3,4]  
a = mylist[0]  
b = mylist[0:1]
```

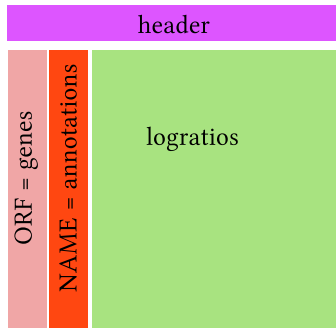
```
mylist = [1,2,3,4]  
a = mylist[0]  
b = mylist[0:1]
```

```
a == 1  
b == [1]
```

Whiteboard Image







```
[["YBR166C", "YOR357C", "YLR292C", ...],
 ["TYR1 ...", "GRD19 ...", "SEC72 ...", ...],
 [[ 0.33, -0.17, 0.04, -0.07, -0.09, ...],
 [-0.64, -0.38, -0.32, -0.29, -0.22, ...],
 [-0.23, 0.19, -0.36, 0.14, -0.40, ...],
 ...]
]
```

Pearson similarity

$$s(x, y) = \frac{1}{N} \sum_i^N \left(\frac{x_i - x_{offset}}{\phi_x} \right) \left(\frac{y_i - y_{offset}}{\phi_y} \right)$$

$$\phi_G = \sqrt{\sum_i^N \frac{(G_i - G_{offset})^2}{N}}$$

Pearson similarity

$$s(x, y) = \sum_i^N \left(\frac{x_i - x_{offset}}{\phi_x} \right) \left(\frac{y_i - y_{offset}}{\phi_y} \right)$$

$$\phi_G = \sqrt{\sum_i^N (G_i - G_{offset})^2}$$

Pearson similarity

$$s(x, y) = \frac{\sum_i^N (x_i - x_{offset})(y_i - y_{offset})}{\sqrt{\sum_i^N (x_i - x_{offset})^2} \sqrt{\sum_i^N (y_i - y_{offset})^2}}$$

Pearson similarity

$$s(x, y) = \frac{\sum_i^N (x_i - x_{\text{offset}})(y_i - y_{\text{offset}})}{\sqrt{\sum_i^N (x_i - x_{\text{offset}})^2} \sqrt{\sum_i^N (y_i - y_{\text{offset}})^2}}$$

Pearson distance

$$d_{\text{uncentered}}(x, y) = 1 - s(x, y)$$

Pearson similarity

$$s(x, y) = \frac{\sum_i^N (x_i - x_{offset})(y_i - y_{offset})}{\sqrt{\sum_i^N (x_i - x_{offset})^2} \sqrt{\sum_i^N (y_i - y_{offset})^2}}$$

Pearson distance

$$d_{uncentered}(x, y) = 1 - s(x, y)$$

Euclidean distance

$$\frac{\sum_i^N (x_i - y_i)^2}{N}$$

Clustering exercises – Negative controls

Write functions to reproduce the shuffling controls in figure 3 of the Eisen paper (removing correlations among genes and/or arrays).

Clustering exercises – Negative controls

Write functions to reproduce the shuffling controls in figure 3 of the Eisen paper (removing correlations among genes and/or arrays).

```
def shuffleGenes(self, seed = None):
    """ Shuffle expression matrix by row. """
    import random
    if (seed != None):
        random.seed(seed)
    indices = range(len(self.genes))
    random.shuffle(indices)
    genes = [self.geneName[i] for i in indices]
    self.geneName = genes
    annotations = [self.geneAnn[i] for i in indices]
    self.geneAnn = genes
    num = [self.num[i] for i in indices]
    self.num = num
```

Clustering exercises – Negative controls

Write functions to reproduce the shuffling controls in figure 3 of the Eisen paper (removing correlations among genes and/or arrays).

```
def shuffleGenes(self, seed = None):
    """ Shuffle expression matrix by row. """
    import random
    if (seed != None):
        random.seed(seed)
    indices = range(len(self.genes))
    random.shuffle(indices)
    genes = [self.geneName[i] for i in indices]
    self.geneName = genes
    annotations = [self.geneAnn[i] for i in indices]
    self.geneAnn = genes
    num = [self.num[i] for i in indices]
    self.num = num

def shuffleRows(self, seed = None):
    """ Permute ratio values within rows. """
    import random
    if (seed != None):
        random.seed(seed)
    for i in self.num:
        random.shuffle(i)
```

Clustering exercises – Negative controls

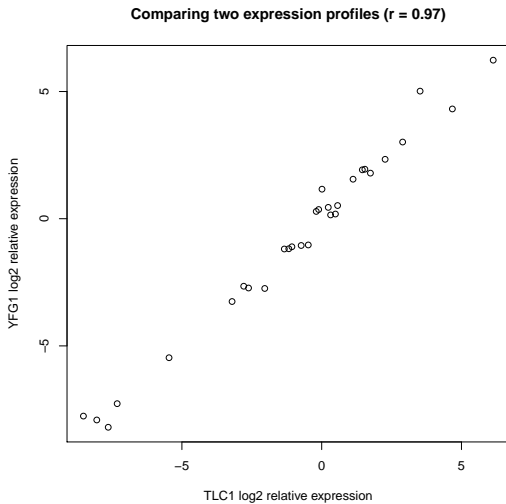
Write functions to reproduce the shuffling controls in figure 3 of the Eisen paper (removing correlations among genes and/or arrays).

```
def shuffleGenes(self, seed = None):
    """Shuffle expression matrix by row."""
    import random
    if (seed != None):
        random.seed(seed)
    indices = range(len(self.genes))
    random.shuffle(indices)
    genes = [self.geneName[i] for i in indices]
    self.geneName = genes
    annotations = [self.geneAnn[i] for i in indices]
    self.geneAnn = genes
    num = [self.num[i] for i in indices]
    self.num = num

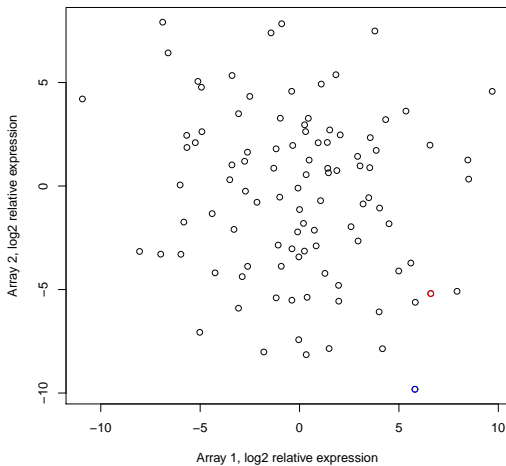
def shuffleRows(self, seed = None):
    """Permute ratio values within rows."""
    import random
    if (seed != None):
        random.seed(seed)
    for i in self.num:
        random.shuffle(i)

def shuffleCols(self, seed = None):
    """Permute ratio values within columns."""
    import random
    if (seed != None):
        random.seed(seed)
    # Transpose the expression matrix
    cols = []
    for col in xrange(len(self.num[0])):
```

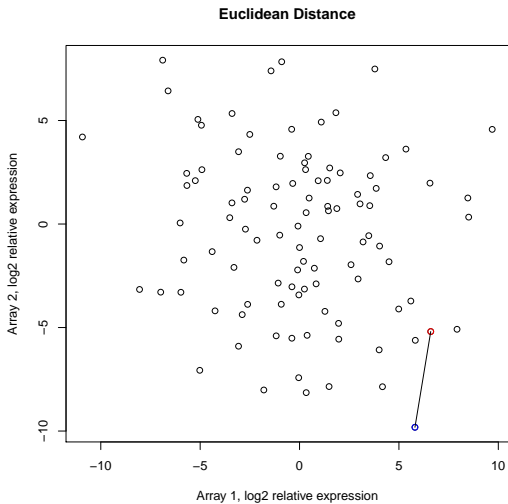

Comparing all measurements for two genes



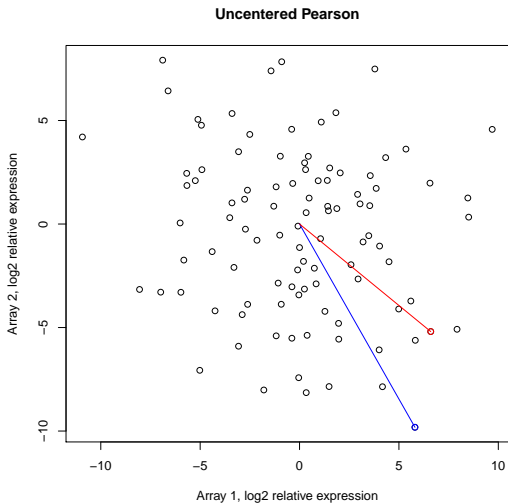
Comparing all genes for two measurements



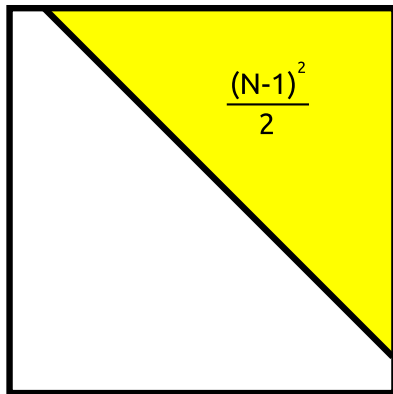
Comparing all genes for two measurements



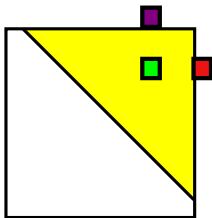
Comparing all genes for two measurements



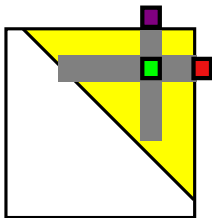
Measure all pairwise distances under distance metric



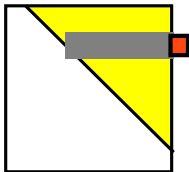
Hierarchical Clustering



Hierarchical Clustering



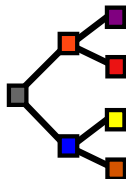
Hierarchical Clustering



Hierarchical Clustering



Hierarchical Clustering



Running Cluster3 from the command line

- /Applications/Cluster.app/Contents/MacOS/Cluster
- /Program Files/Stanford University/Cluster3/Cluster.com

Command-line programs are like functions

\man program" is like \help(function)"

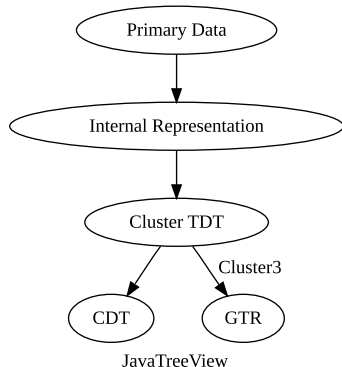
Use the `subprocess` module to run command-line programs from within Python.

USAGE: cluster [options]

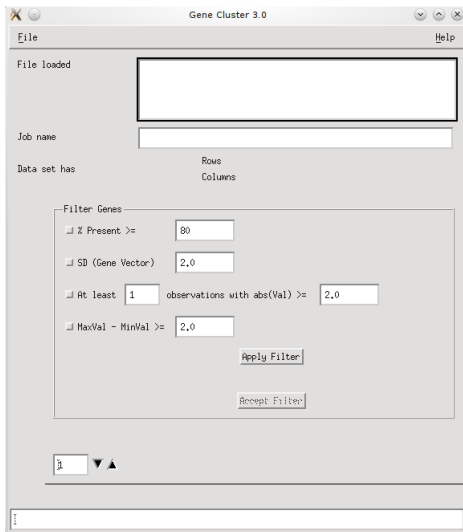
-f filename	File loading
-u jobname	Allows you to specify a different name for the output files (default is derived from the input file name)
-g [0..8]	Specifies the distance measure for gene clustering 0: No gene clustering 1: Uncentered correlation 2: Pearson correlation 3: Uncentered correlation, absolute value 4: Pearson correlation, absolute value 5: Spearman's rank correlation 6: Kendall's tau 7: Euclidean distance 8: City-block distance (default: 0)
-m [msca]	Specifies which hierarchical clustering method to use m: Pairwise complete-linkage s: Pairwise single-linkage c: Pairwise centroid-linkage a: Pairwise average-linkage (default: m)

Scripting the Protocol

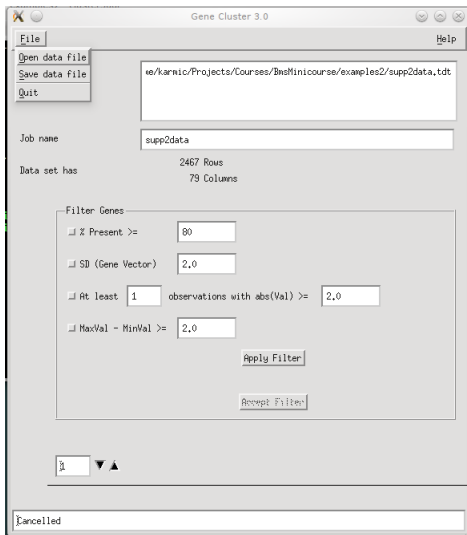
```
from subprocess import check_call
check_call(
    # Which program to run
    ("cluster",
     # Input file
     "-f", "supp2data.tdt",
     # Output prefix
     "-u", "supp2data.Uncentered.Complete",
     # Clustering method: complete linkage
     "-m", "m",
     # Distance function: uncentered Pearson
     "-g", "1"))
```



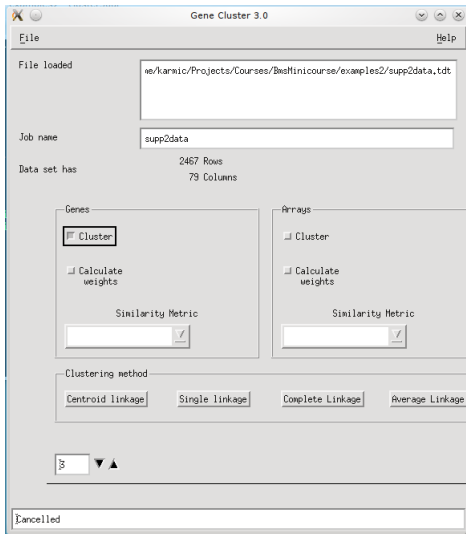
Using the Cluster3 GUI



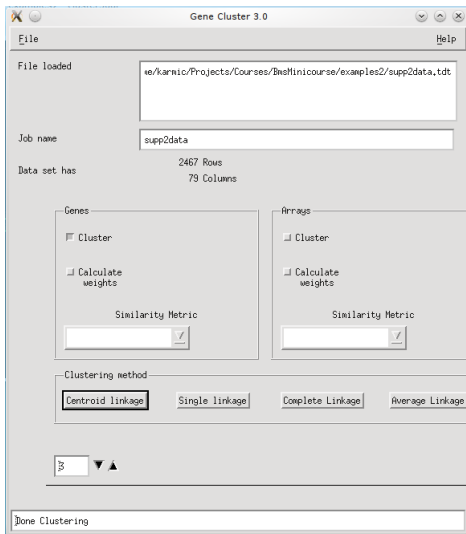
Load your data



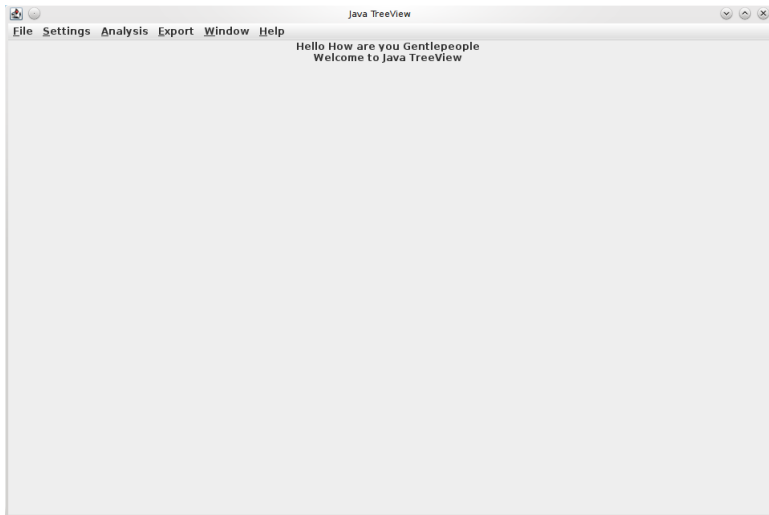
Choose distance function



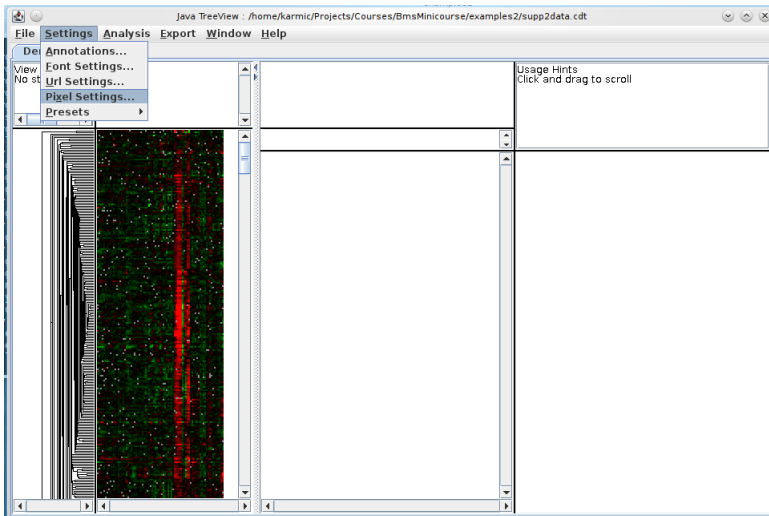
Choose linking method



Using JavaTreeView



Adjust pixel settings for global view



Adjust pixel settings for global view

The screenshot shows the Java TreeView application window. The main view displays a heatmap with a dendrogram on the left. A 'Pixel Settings' dialog box is open in the foreground, allowing for adjustments to the heatmap's appearance. The dialog includes the following controls:

- Global:** Radio buttons for 'Fixed Scale' (value: 481012658227) and 'Fill'. The 'Fill' option is selected.
- Zoom:** Radio buttons for 'Fixed Scale' (value: 12.0) and 'Fill'. The 'Fixed Scale' option is selected.
- Contrast:** A slider with a 'Value' of 3.0.
- LogScale:** A checkbox for 'Log (base 2) Center' (value: 1.0), which is currently unchecked.
- Colors:** Four color selection buttons: 'Positive' (red), 'Zero' (black), 'Negative' (green), and 'Missing' (grey). Below these are 'Load...', 'Save...', and 'Make Preset' buttons, and a dropdown menu showing 'RedGreen' and 'YellowBlue' color schemes.

The background window shows a menu bar with 'File', 'Settings', 'Analysis', 'Export', 'Window', and 'Help'. The title bar indicates the file path: 'java TreeView : /home/karmic/Projects/Courses/BmsMinicourse/examples2/supp2data.cdt'.

Select annotation columns

The screenshot shows a Java TreeView application window titled "java TreeView: /home/kermic/Projects/Courses/BmsMinicourse/examples2/supp2data.cdt". The interface is dark-themed. On the left, there is a file tree view showing a directory structure with folders like "Annotations" and "Annotations2". The main area displays a large table of data. The table has several columns, with the first column containing a list of identifiers. A vertical selection bar is visible on the left side of the table, indicating that a column is selected. The table contains multiple rows of data, with some cells containing text and others containing numerical values. The bottom right corner of the application window shows standard window controls (minimize, maximize, close) and a search icon.

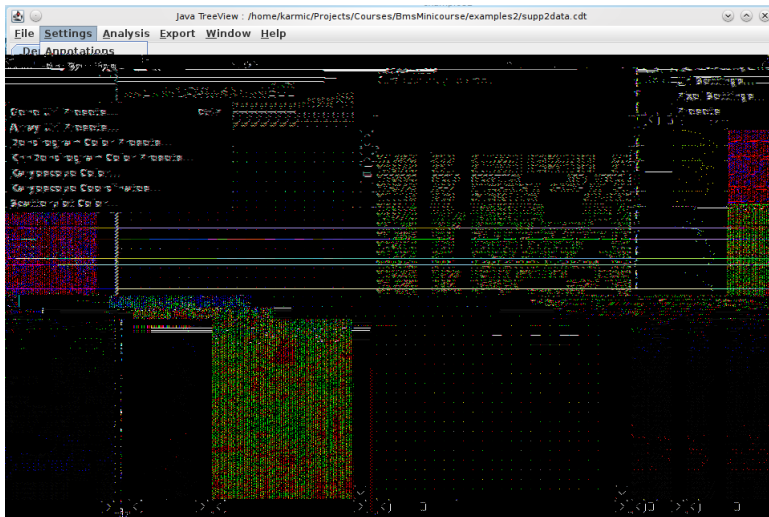
Select annotation columns

The screenshot displays the Java TreeView application window. The title bar reads "Java TreeView : /home/karmic/Projects/Courses/BmsMinicourse/examples2/supp2data.cdt". The menu bar includes "File", "Settings", "Analysis", "Export", "Window", and "Help".

The main interface is divided into several sections:

- Dendrogram:** Located on the left, it shows a hierarchical clustering tree. A specific cluster is highlighted in red.
- View Status:** Located in the top-left corner, it displays "Row: 7 (YGR2)", "Column: 23 (ELU)", and "Value: -0.06".
- Usage Hints:** Located in the top-right corner, it contains the text "Mouse over to get info".
- Annotation Settings Dialog:** A modal dialog box is open in the center, titled "Annotation Settings". It has tabs for "Array Tree" and "Gene Tree". Under the "Gene Tree" tab, there are sub-sections for "Gene" and "Array". The "Headers to include" list contains: "GID", "ORF", "NAME", and "GWEIGHT".
- Gene List:** On the right side, a list of genes is shown with their associated biological processes. For example, "YAL062W" is associated with "GDH3" and "GLUTAMATE BIOSYNTHESIS".
- Heatmap:** The main area of the window is a heatmap where each row represents a gene and each column represents a data point. The color scale ranges from dark blue (low values) to red (high values). A prominent red vertical band is visible on the left side of the heatmap, corresponding to the red cluster in the dendrogram.

Select URL for gene annotations



Select URL for gene annotations

The screenshot shows the Java TreeView application window. The title bar indicates the file path: `/home/karmic/Projects/Courses/BmsMinicourse/examples2/supp2data.cdt`. The menu bar includes **File**, **Settings**, **Analysis**, **Export**, **Window**, and **Help**. The **Dendrogram** tab is active, showing a tree structure with nodes labeled 'alpha' and a heatmap below it. A 'Usage Hints' box is visible, stating: 'Click to select node - use arrow keys to navigate tree'. A 'Presets' dialog box is open, titled 'Modify Url Presets', with 'Gene' and 'Array' tabs. The dialog contains a table of presets with columns for 'Enabled', 'Header', 'Name', 'Template', and 'Default'.

Enabled	Header	Name	Template	Default
<input type="checkbox"/>	*	SGD	http://genome-www4.stanford.edu/cgi-bin/SGD/locus.pl?locus=HEADER	<input checked="" type="radio"/>
<input type="checkbox"/>	*	YPD	http://www.proteome.com/databases/YPD/reports/HEADER.html	<input type="radio"/>
<input type="checkbox"/>	*	WormBase	http://www.wormbase.org/cgi-bin/locate.cgi?locus=HEADER&start=0&start=0&ie=utf-8&oe=utf-8	<input type="radio"/>
<input type="checkbox"/>	*	Source CloneID	http://genome-www4.stanford.edu/cgi-bin/SMD/source/sourceResult?option=CloneID	<input type="radio"/>
<input type="checkbox"/>	*	FlyBase	http://flybase.bio.indiana.edu/bin/fbqeng.html?HEADER	<input type="radio"/>
<input type="checkbox"/>	*	MouseGD	http://www.jax.org/avaw/Servlet/SearchTool?query=HEADER&selectedQuery=Genes+and+Markers	<input type="radio"/>
<input type="checkbox"/>	*	GenomeNetEcoli	http://www.genome.ad.jp/dbget-bin/www_bget?eco:HEADER	<input type="radio"/>
<input type="checkbox"/>		None		<input type="radio"/>

Buttons: **Save** **Cancel**

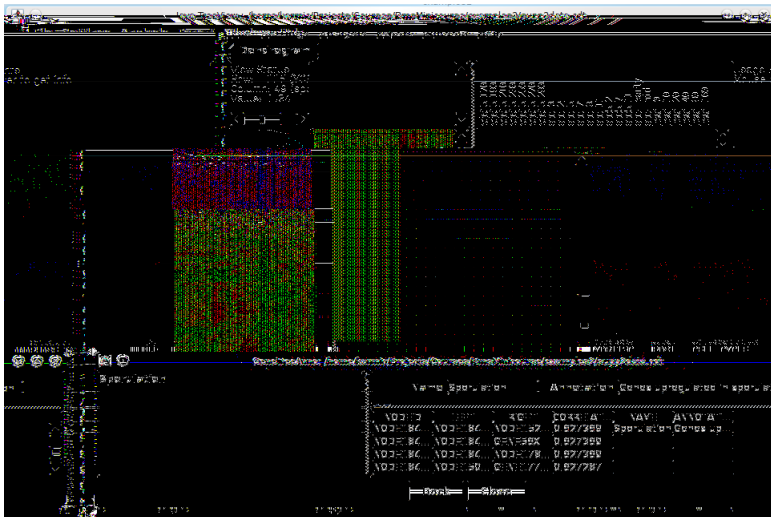
Activate and detach annotation window

The screenshot shows the Java TreeView application window. The title bar reads "Java TreeView : /home/karmic/Projects/Courses/BmsMinicourse/examples2/supp2data.cdt". The menu bar includes "File", "Settings", "Analysis", "Export", "Window", and "Help". The "Analysis" menu is open, showing options: "Find Genes...", "Find Arrays...", "Stats...", "Dendrogram", "Alignment", "KnnDendrogram", "Karyoscope", "Scatterplot", "ArrayTreeAnno", "GeneTreeAnno", "Remove Current", and "Detach Current".

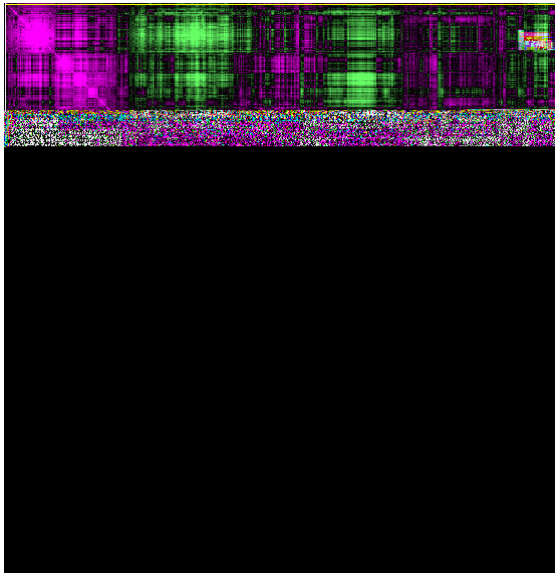
Below the menu, there are input fields for "Name" and "Annotation". The main area contains a table with the following columns: NODEID, LEFT, RIGHT, CORRELAT..., NAME, and ANNOTATI... The table lists various nodes and their associated gene IDs and correlation values.

NODEID	LEFT	RIGHT	CORRELAT...	NAME	ANNOTATI...
NODE243...	GENE182...	NODE239...	0.347965		
NODE244...	NODE242...	NODE243...	0.347965		
NODE244...	GENE550X	NODE239...	0.344607		
NODE244...	NODE243...	NODE244...	0.342251		
NODE244...	NODE244...	GENE4X	0.334454		
NODE244...	NODE240...	NODE239...	0.333461		
NODE244...	NODE244...	NODE243...	0.331585		
NODE244...	NODE244...	NODE238...	0.328813		
NODE244...	NODE244...	GENE229...	0.305824		
NODE244...	GENE495X	GENE217...	0.304111		
NODE244...	GENE219...	GENE218...	0.303188		
NODE245...	NODE244...	GENE215X	0.301587		
NODE245...	NODE244...	NODE242...	0.298323		
NODE245...	NODE240...	NODE244...	0.289436		
NODE245...	NODE242...	GENE219...	0.287138		
NODE245...	NODE245...	NODE243...	0.284232		
NODE245...	NODE245...	GENE527X	0.277872		
NODE245...	NODE245...	NODE234...	0.27761		
NODE245...	NODE245...	NODE244...	0.271103		
NODE245...	NODE233...	NODE245...	0.260487		
NODE245...	NODE243...	NODE245...	0.220385		
NODE246...	NODE244...	NODE245...	0.197665		
NODE246...	NODE245...	NODE243...	0.180953		
NODE246...	NODE246...	GENE182...	0.161919		
NODE246...	NODE246...	NODE119...	0.126461		
NODE246...	NODE246...	NODE245...	0.09892		
NODE246...	NODE245...	NODE246...	-0.087409		
NODE246...	NODE246...	NODE246...	-0.354391		

Activate and detach annotation window



Clustering exercises – Visualizing the distance matrix



Clustering exercises – Scripting Cluster

Modify the clustering protocol script to run Cluster3 multiple times on the same input, varying distance metric and/or clustering method. Be sure to give the output files distinct names.

Clustering exercises – Scripting Cluster

Modify the clustering protocol script to run Cluster3 multiple times on the same input, varying distance metric and/or clustering method. Be sure to give the output files distinct names.

```
metrics = ("None",
          "Uncentered",
          "Pearson",
          "UncenteredAbs",
          "PearsonAbs",
          "Spearman",
          "Kendall",
          "Euclidean",
          "City")
linkage = (("Complete", "m"),
          ("Single", "s"),
          ("Centroid", "c"),
          ("Average", "a"))

# Loop over all 32 possible methods
print "Starting hierarchical clustering runs..."
from subprocess import check_call
for metric in xrange(1, len(metrics)):
    print "    ", metrics[metric], "..."
    for (linkname, link) in linkage:
        print "        ", linkname
        check_call(("cluster", "-f", "shuffled.txt",
                    "-u", ".".join(("shuffled",
                                    metrics[metric],
                                    linkname))),
                    "-m", link, "-g", str(metric)))
```

If you haven't done so already, read the PNAS paper

Explore the figure 2 data with Cluster3 and JavaTreeView.

- Can you find/reproduce the clusters described in the paper?
- Are the annotations consistent with the current annotations in SGD?
- Are there other patterns that you can find in the data?
- What follow-up experiments are prompted by this analysis?

Whiteboard Image

