Practical Bioinformatics

Mark Voorhies

5/23/2013

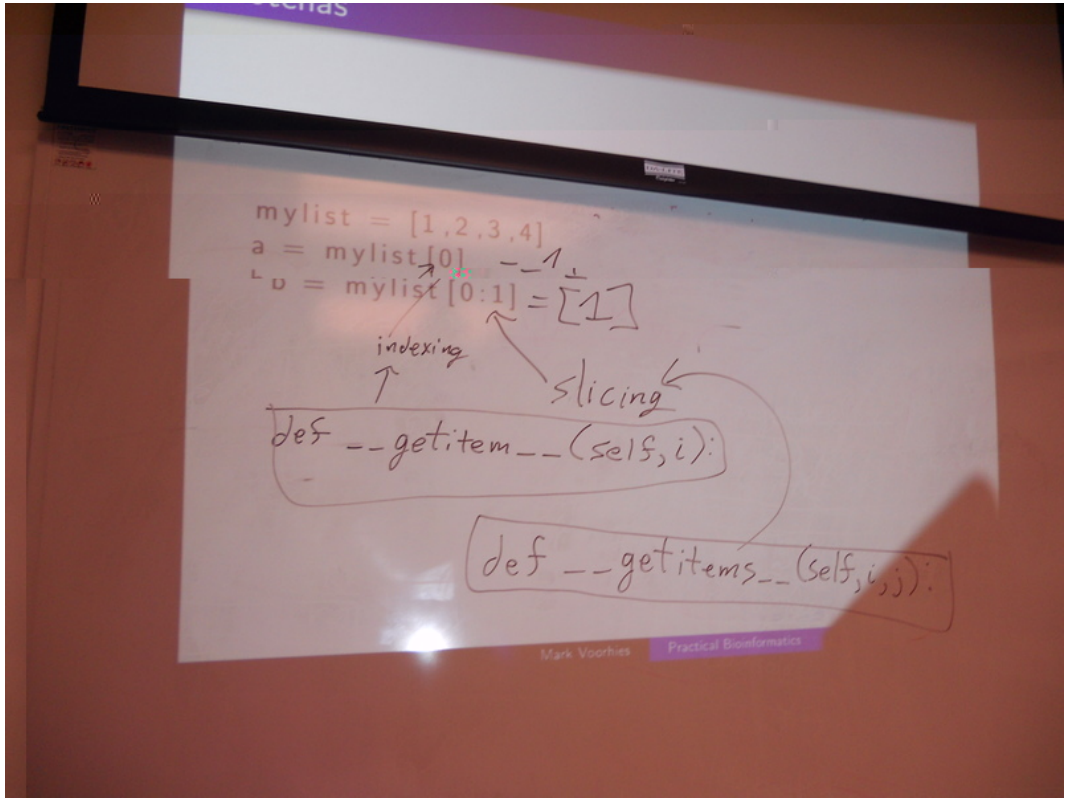
Gotchas

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

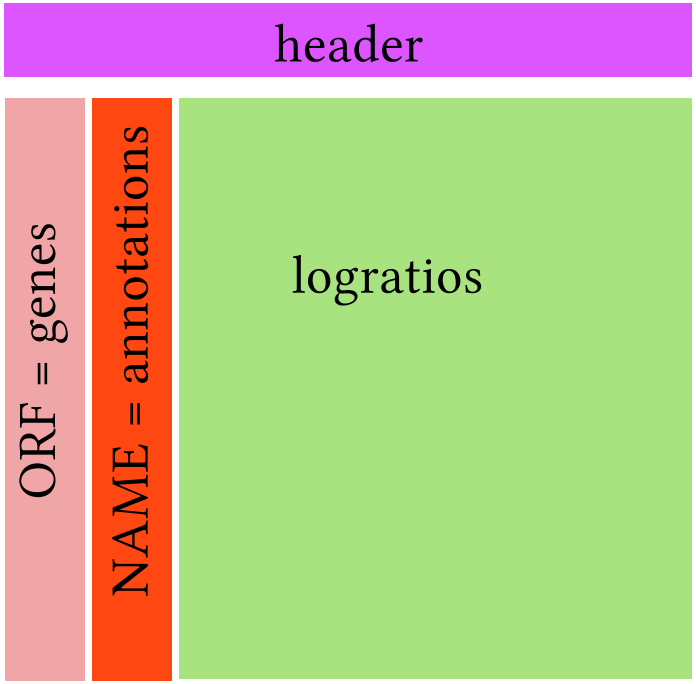
Gotchas

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

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

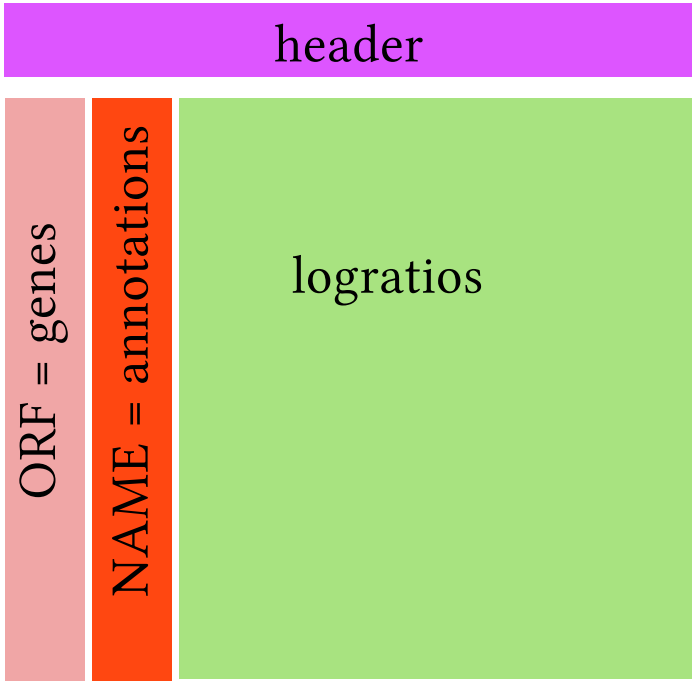
Whiteboard Image



supp2data.cdt



supp2data.cdt




```
[["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 distances

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 distances

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 distances

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 distances

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

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

Pearson distances

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 shu ing controls in gure 3 of the Eisen paper (removing correlations among genes and/or arrays).

Clustering exercises – Negative controls

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

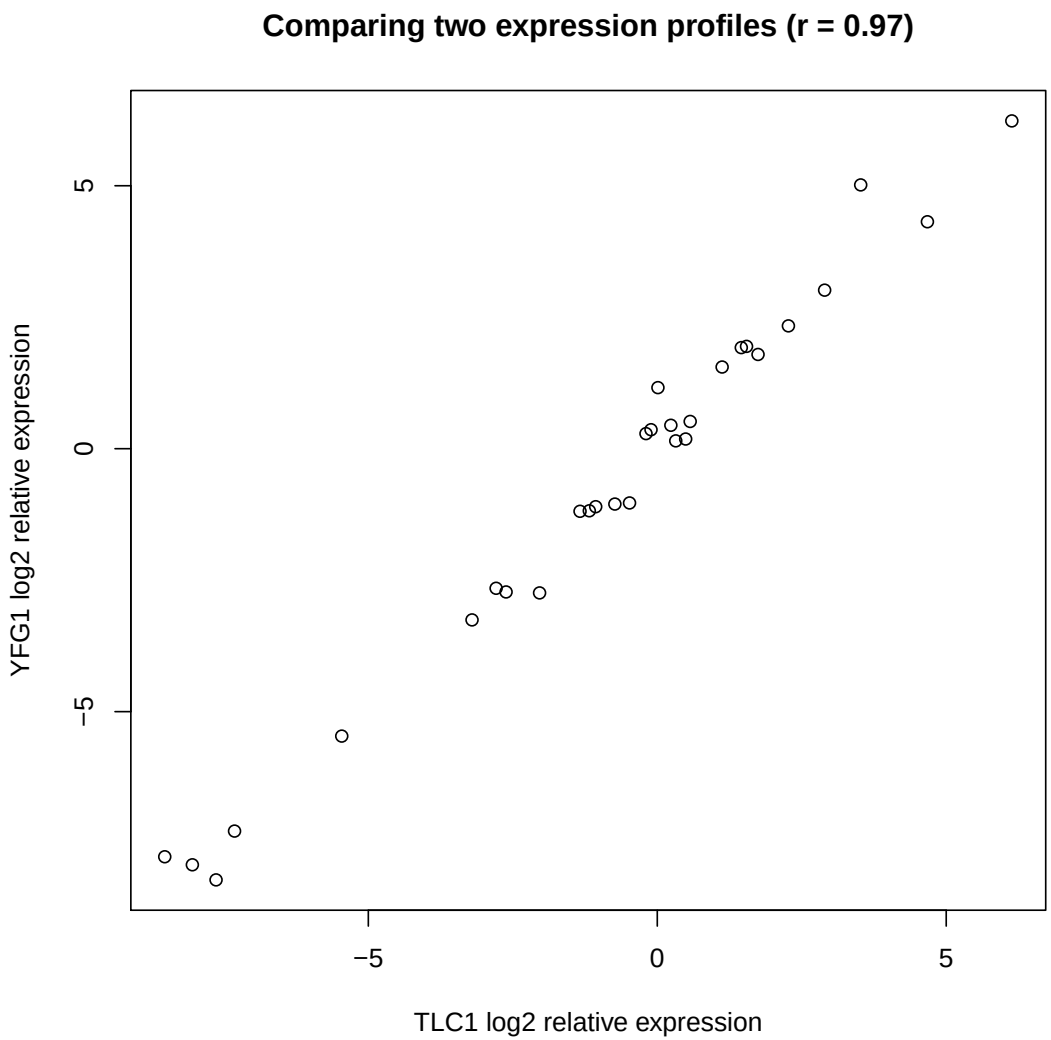
```
def shuffleGenes(self, seed = None):
    """Shuffle expression matrix by row."""
    import random
    if(seed != None):
        random.seed(seed)
    indices = range(len(self.genes))
    random.shuffle(indices)
    genes = [self.geneName[i] for i in indices]
    self.geneName = genes
    annotations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

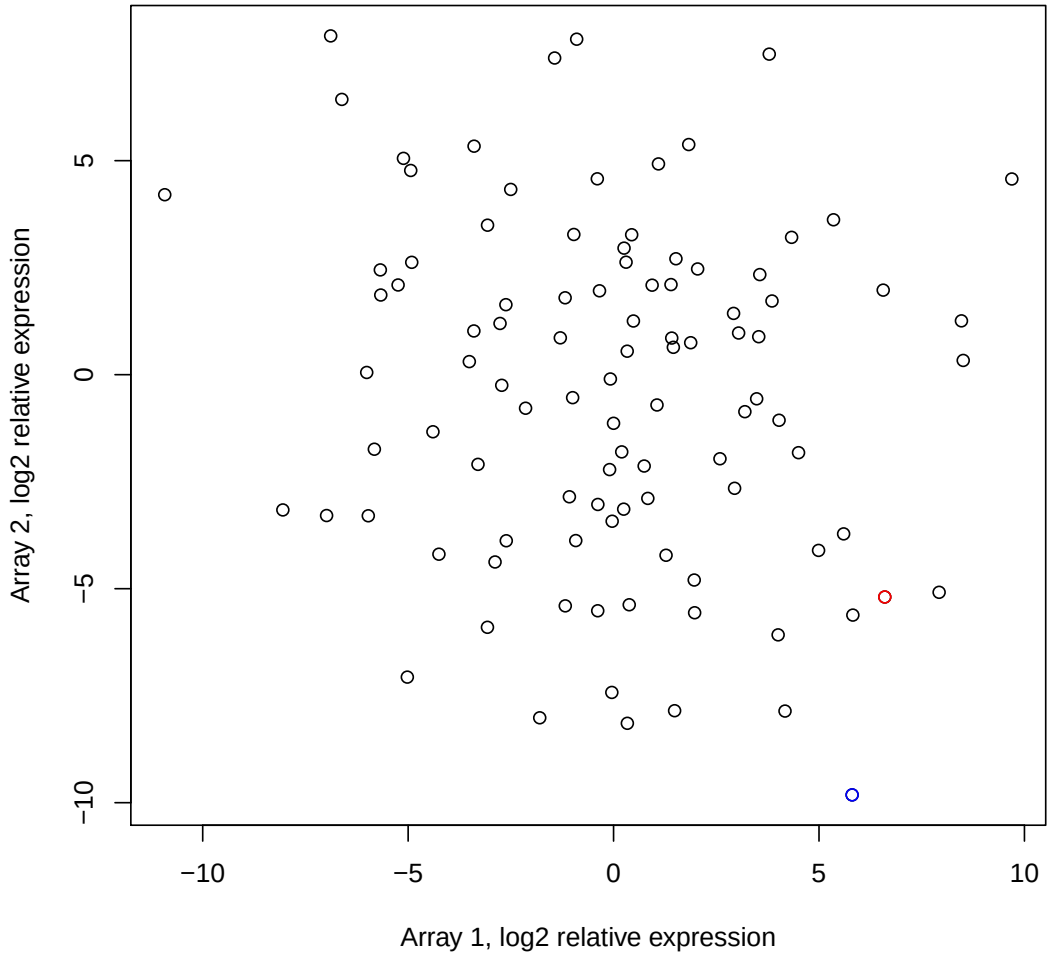
Write functions to reproduce the shu ing controls in gure 3 of the Eisen paper (removing correlations among genes and/or arrays).

```
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])):
```

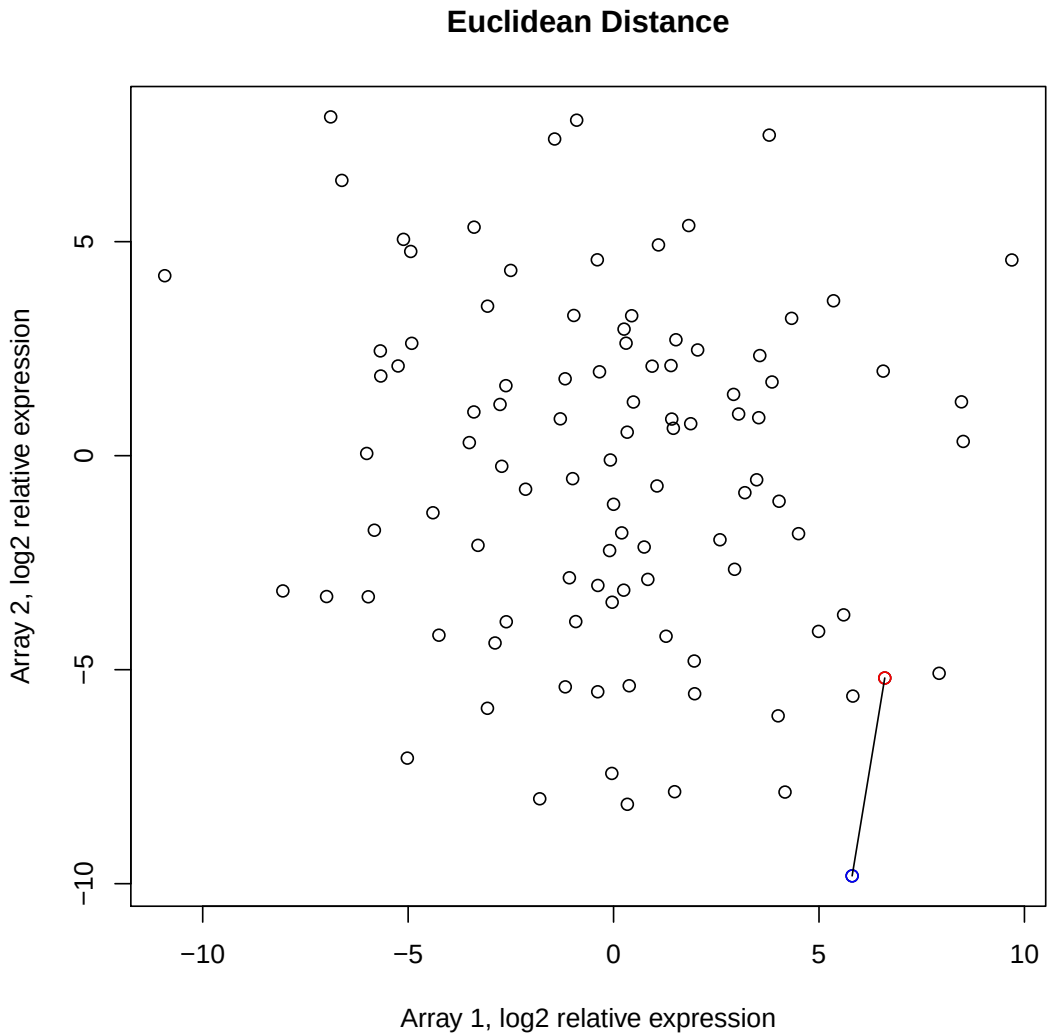
Comparing all measurements for two genes



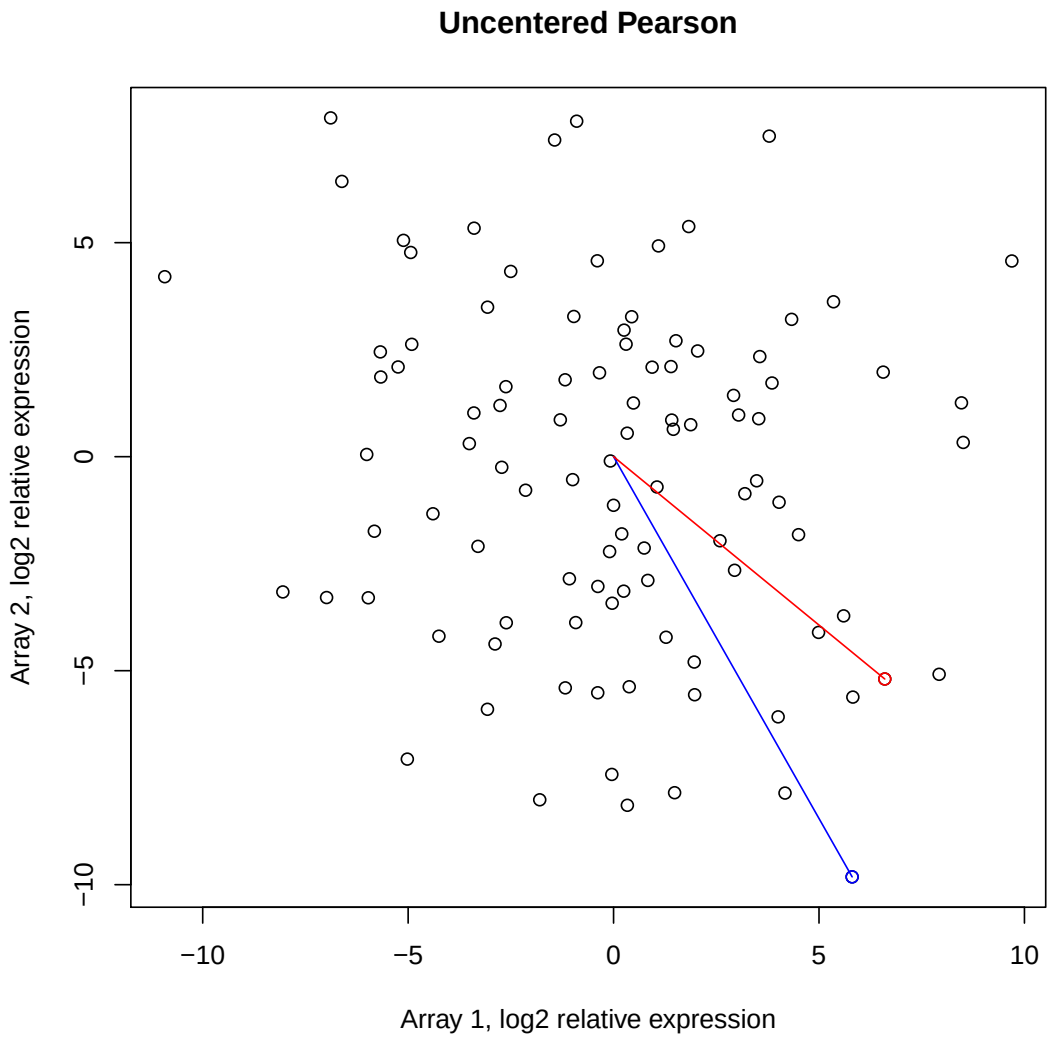
Comparing all genes for two measurements



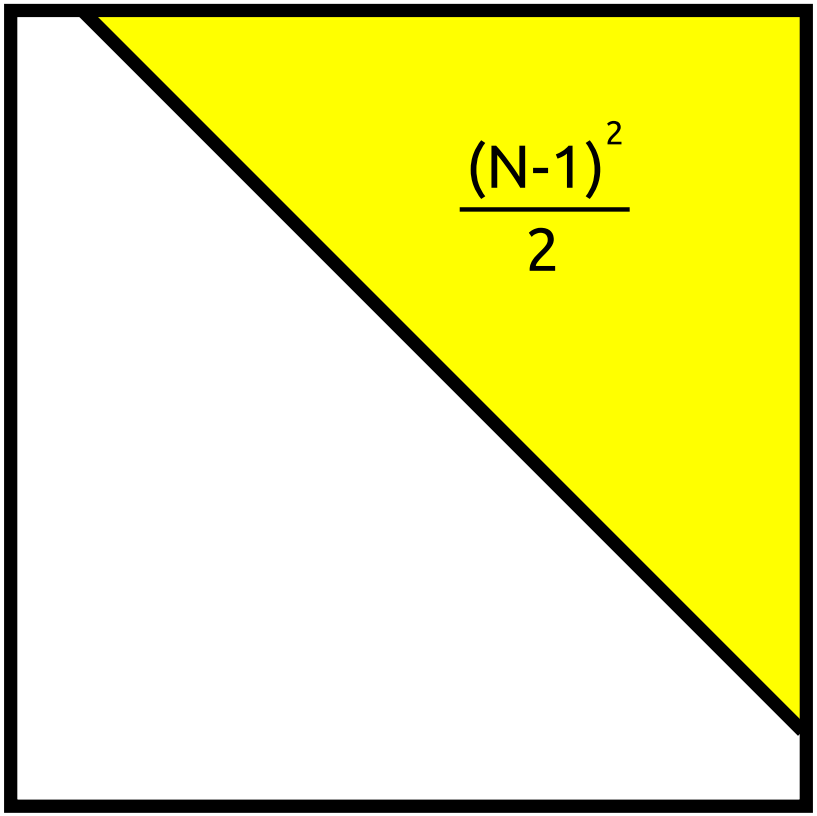
Comparing all genes for two measurements



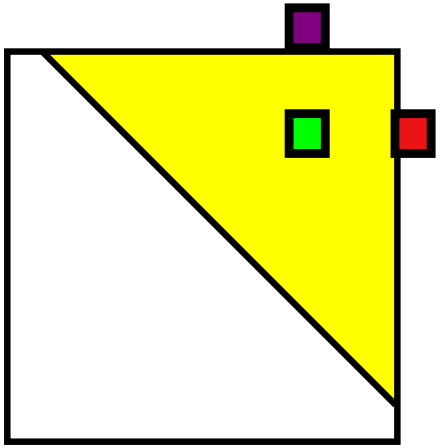
Comparing all genes for two measurements

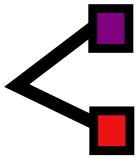


Measure all pairwise distances under distance metric

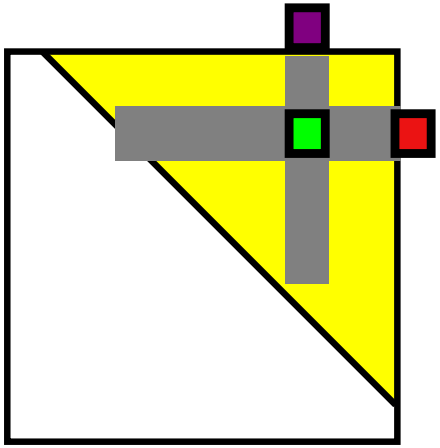


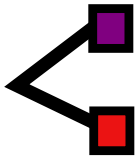
Hierarchical Clustering



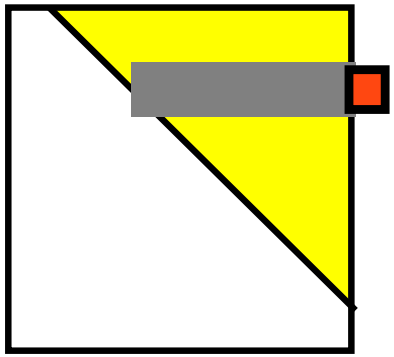


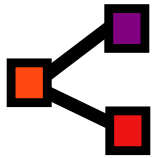
Hierarchical Clustering



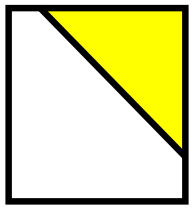


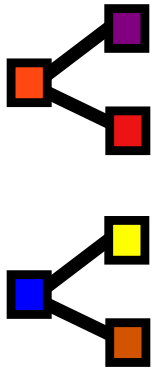
Hierarchical Clustering



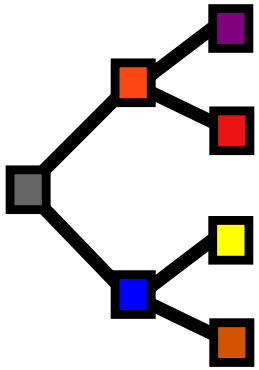


Hierarchical Clustering





Hierarchical Clustering



Scripting Cluster

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)"

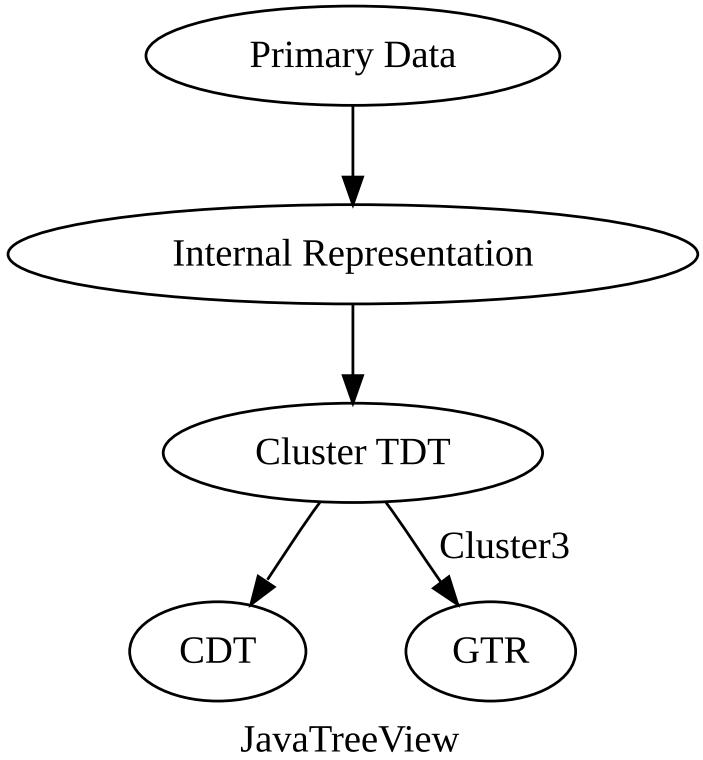
Use the `subprocess` module to run command-line programs from within Python.

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)

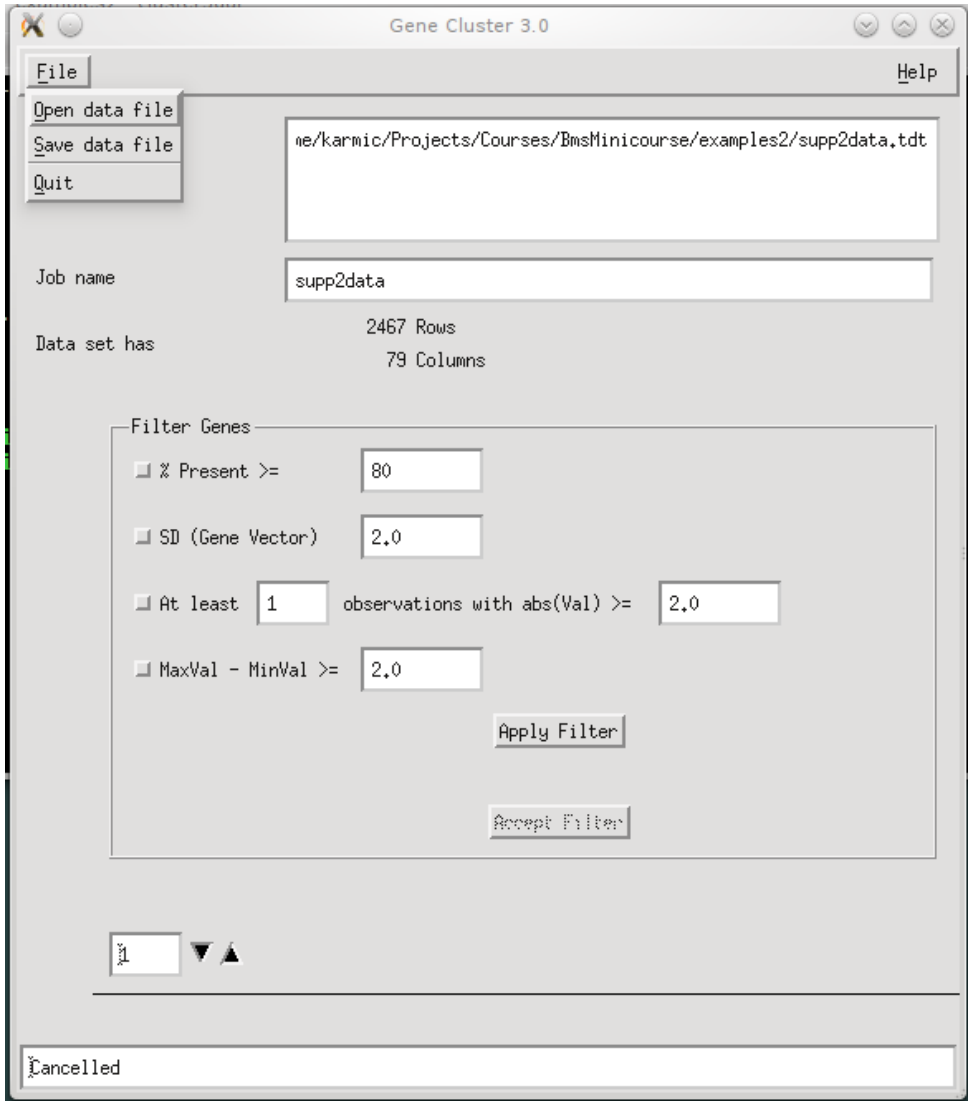
Scripting the Protocol

```
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"))
```

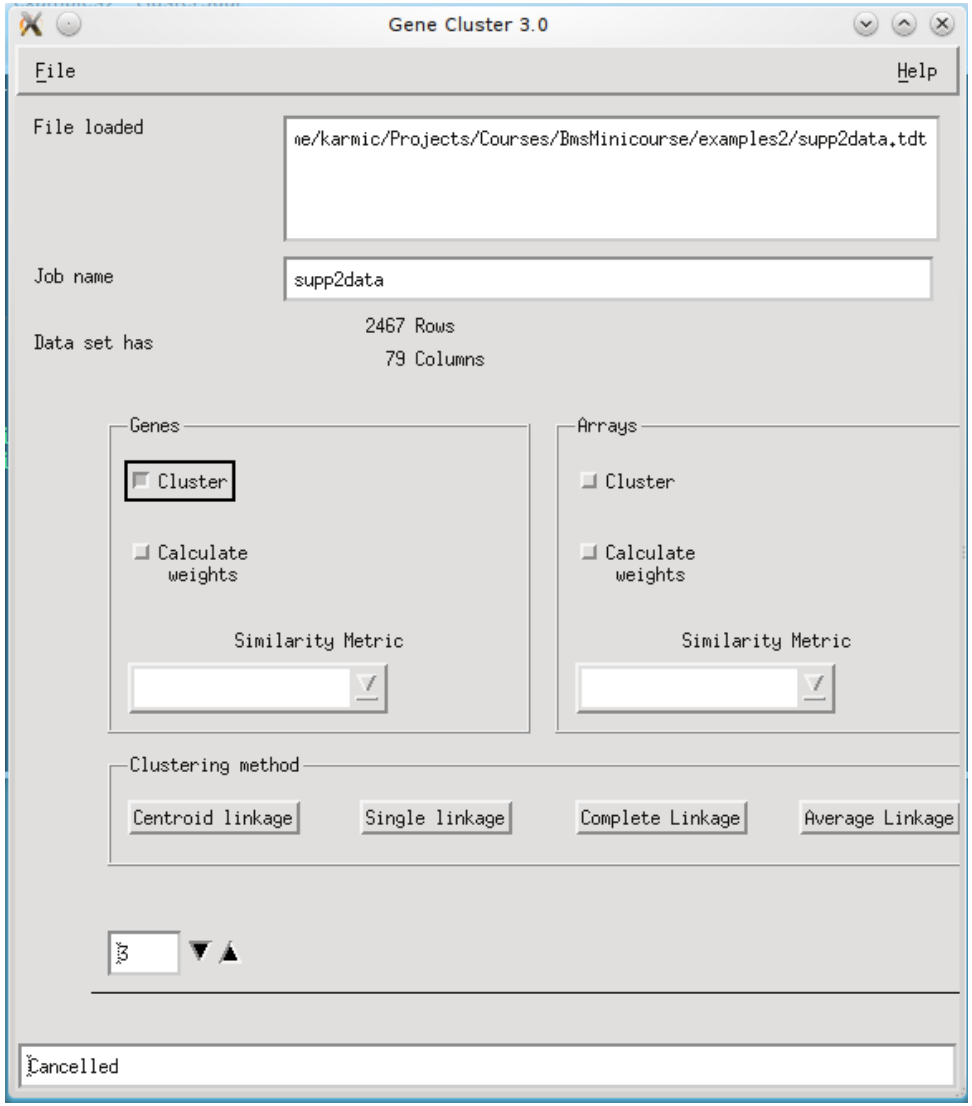


Using the Cluster3 GUI

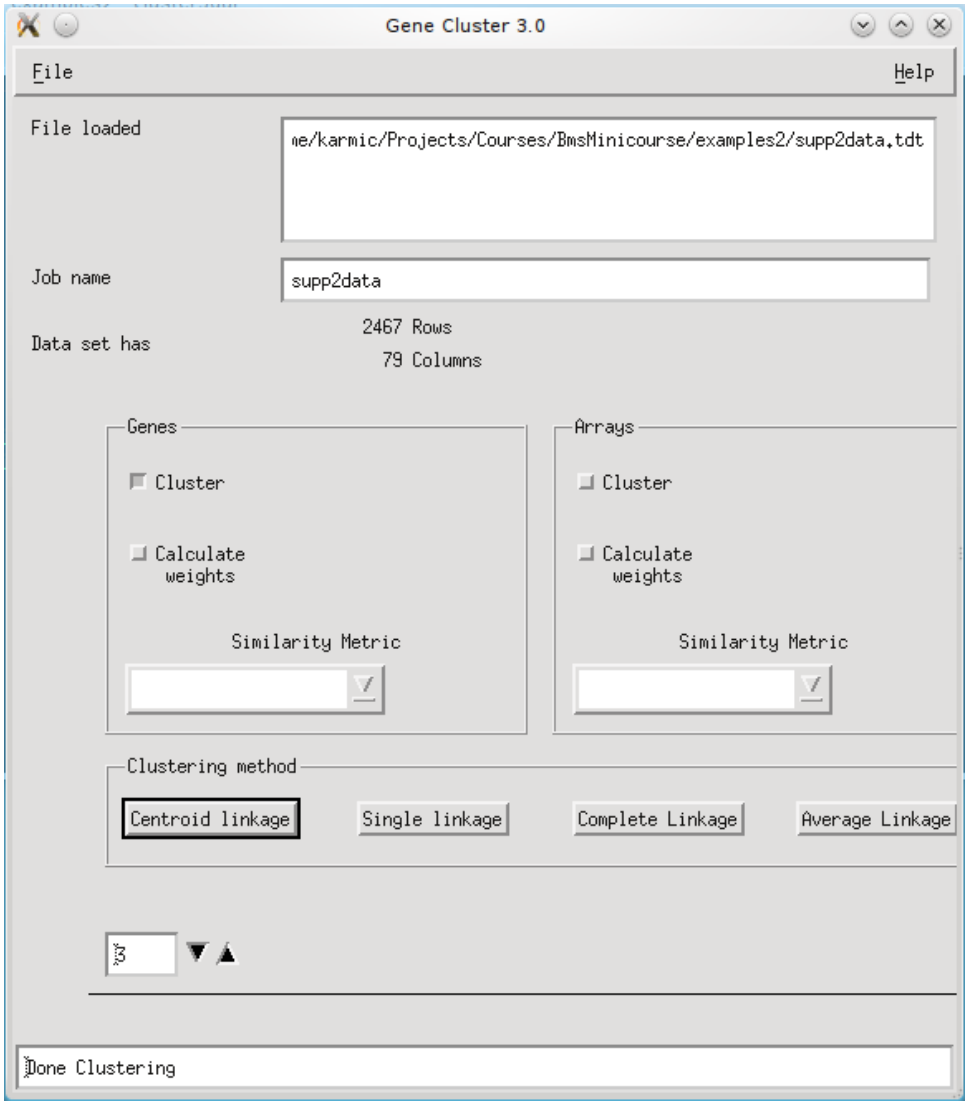
Load your data



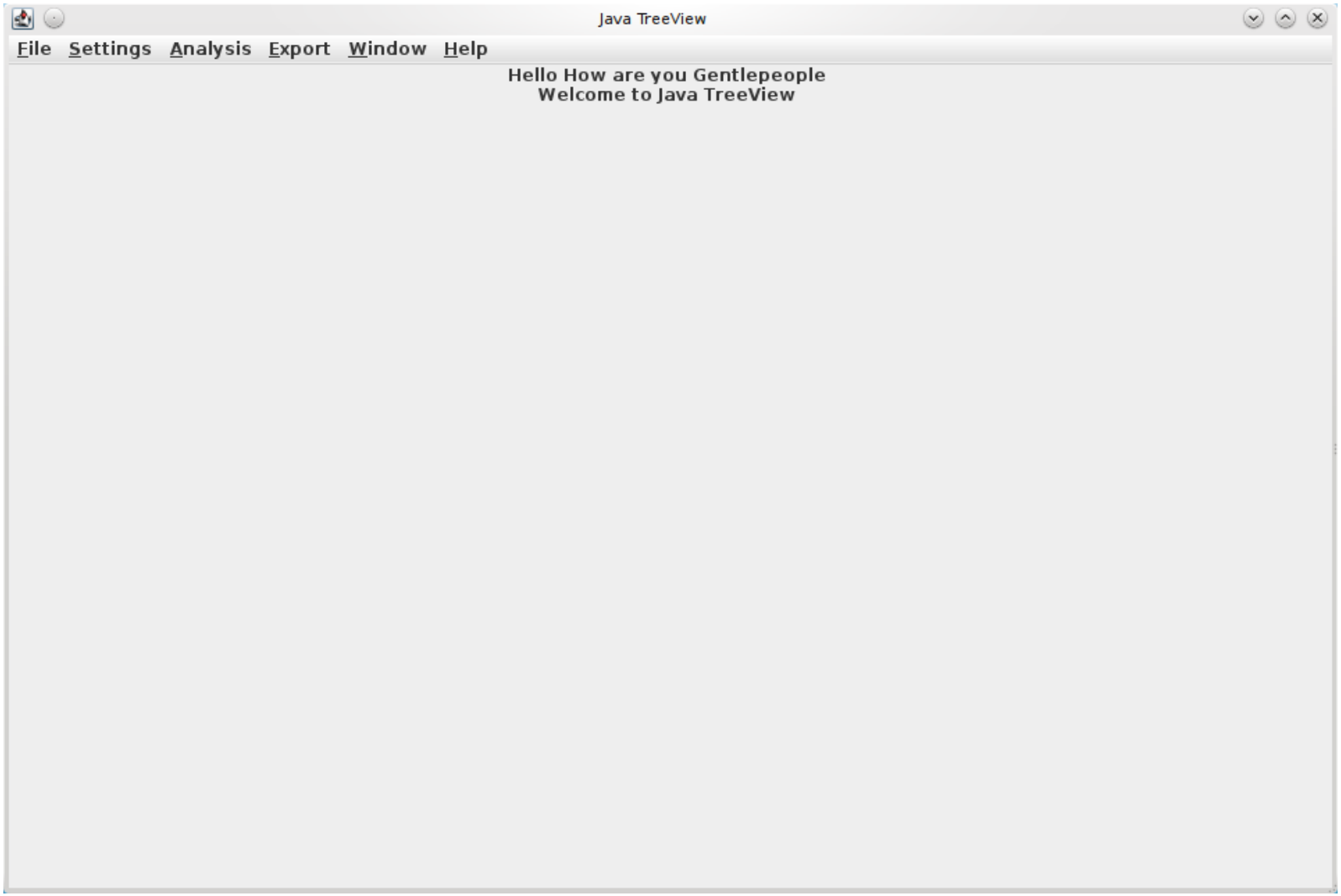
Choose distance function



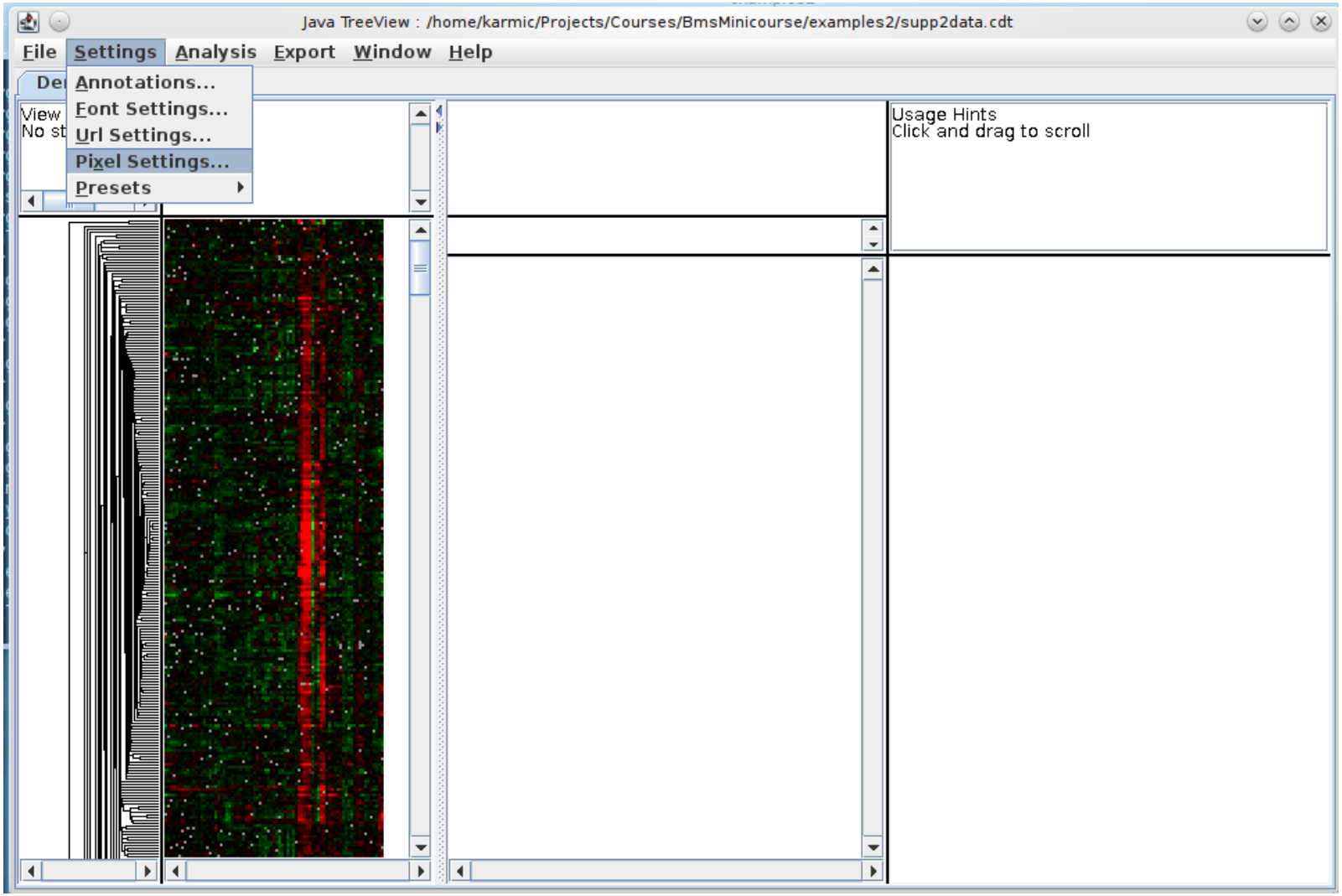
Choose linking method



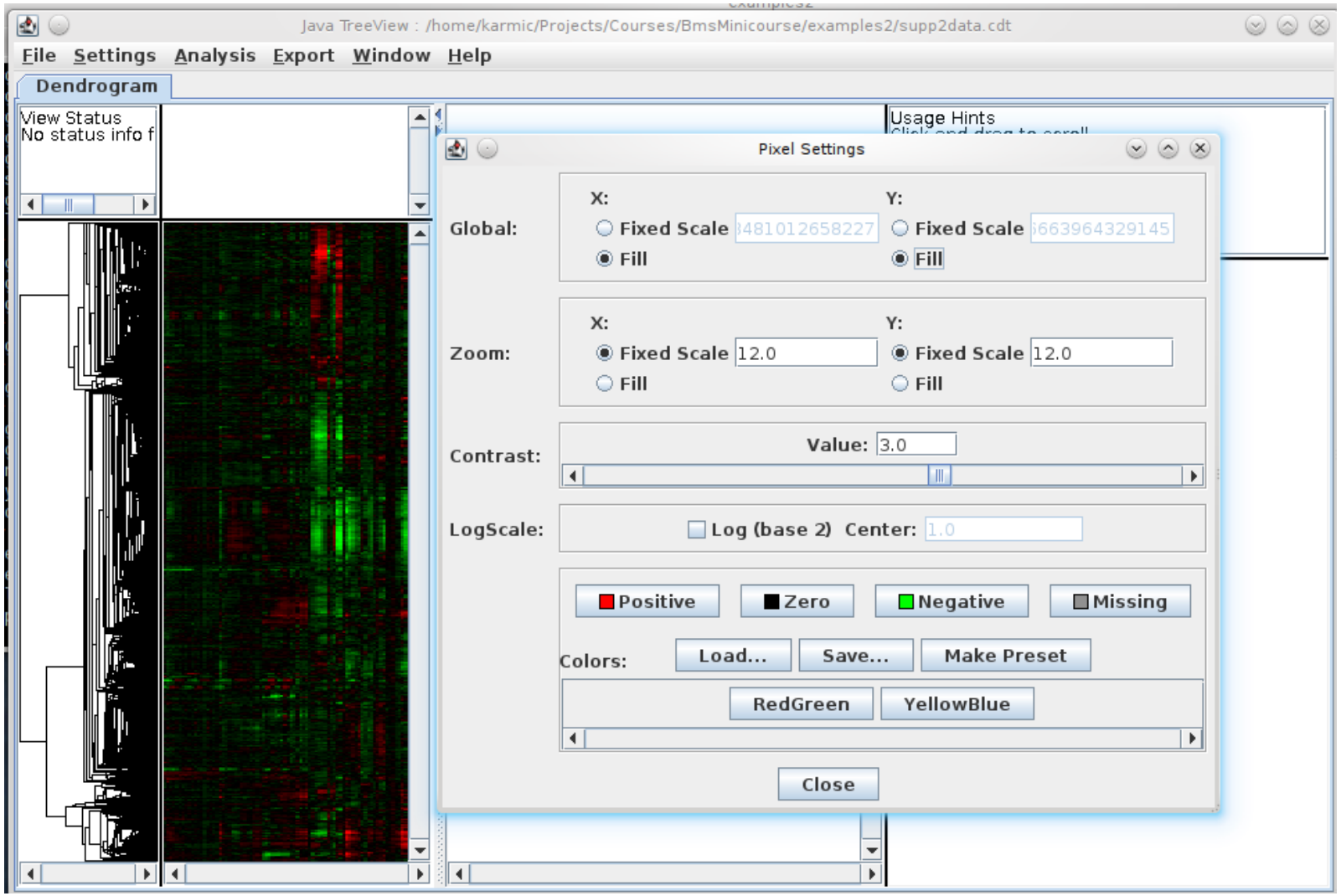
Using JavaTreeView



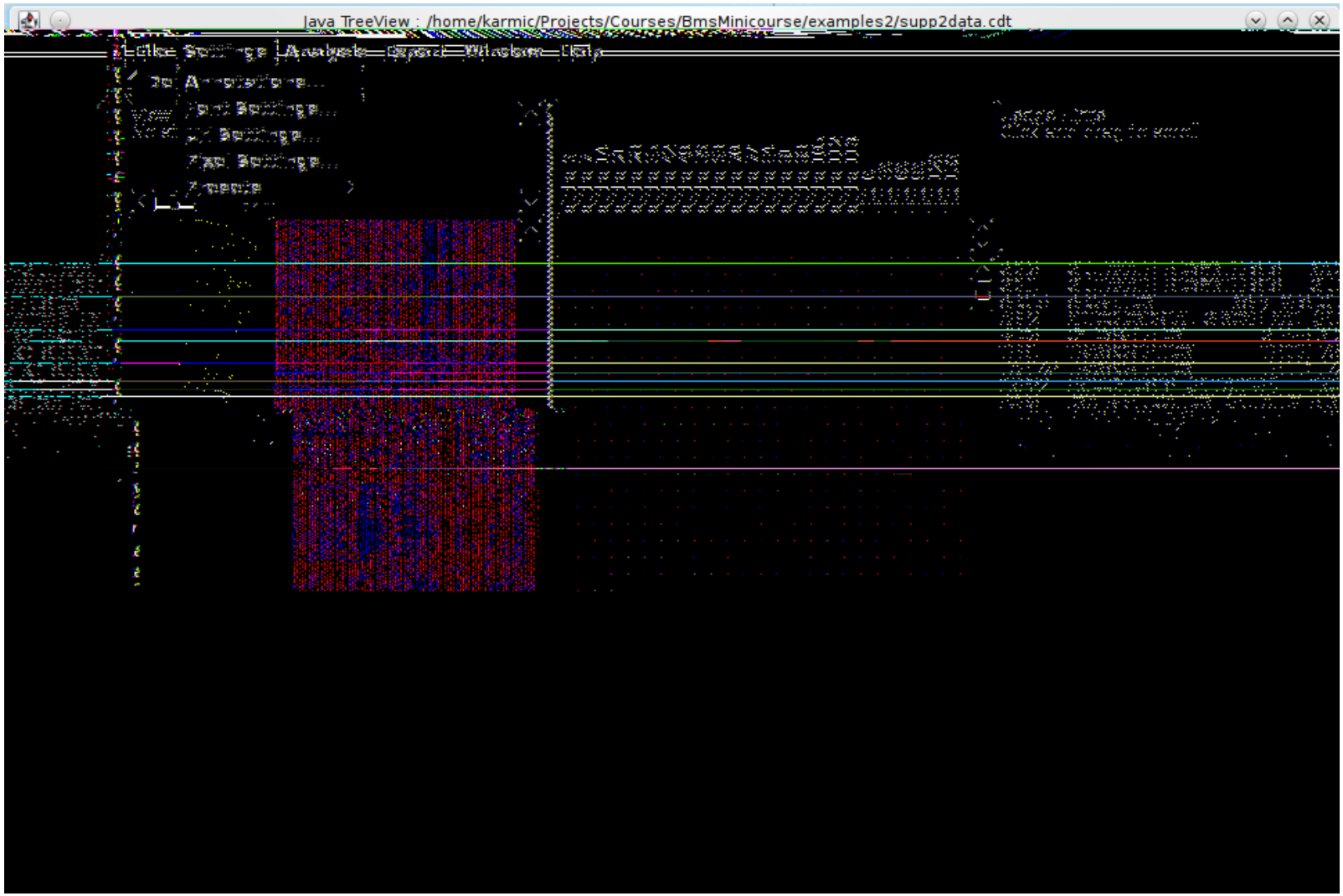
Adjust pixel settings for global view



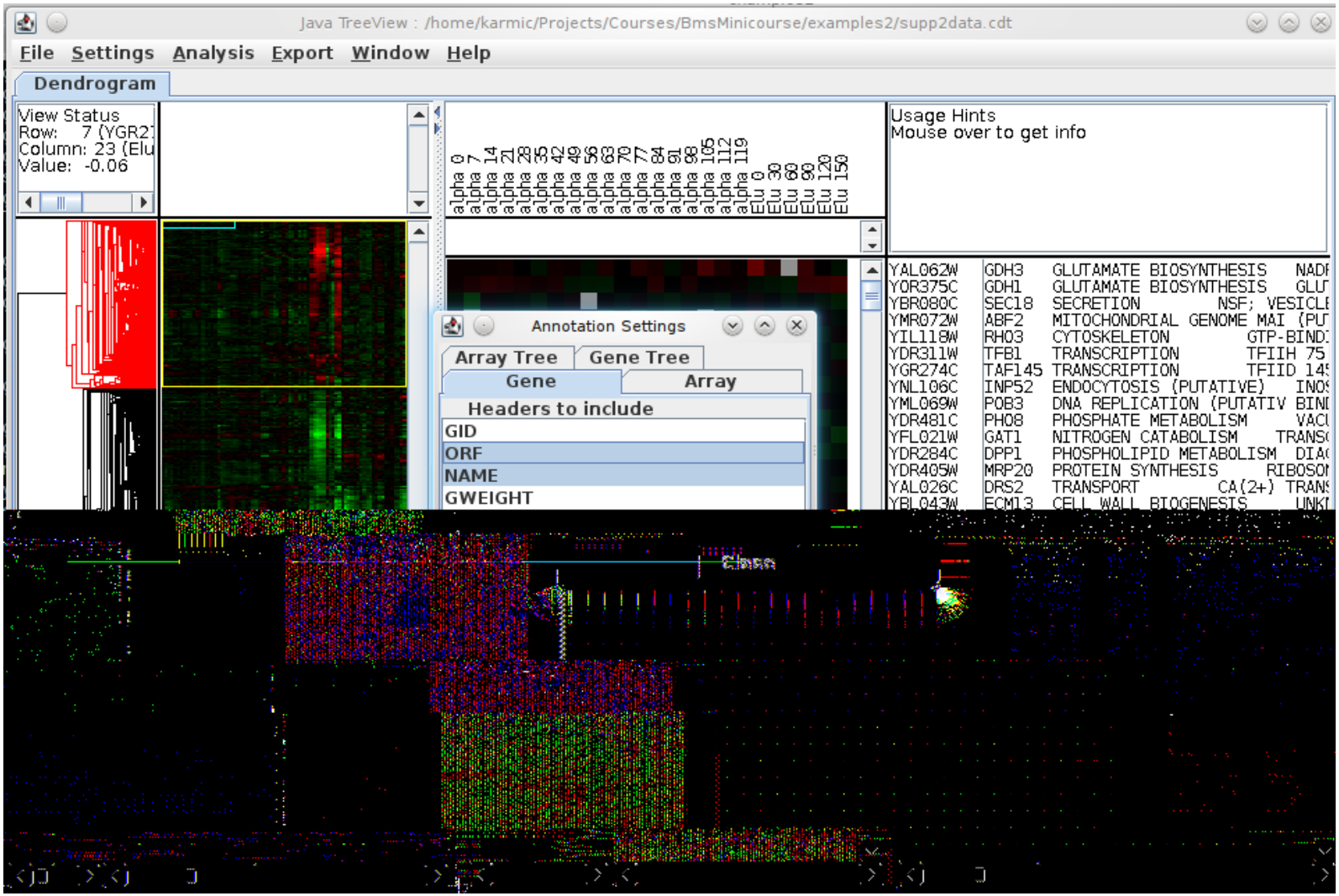
Adjust pixel settings for global view



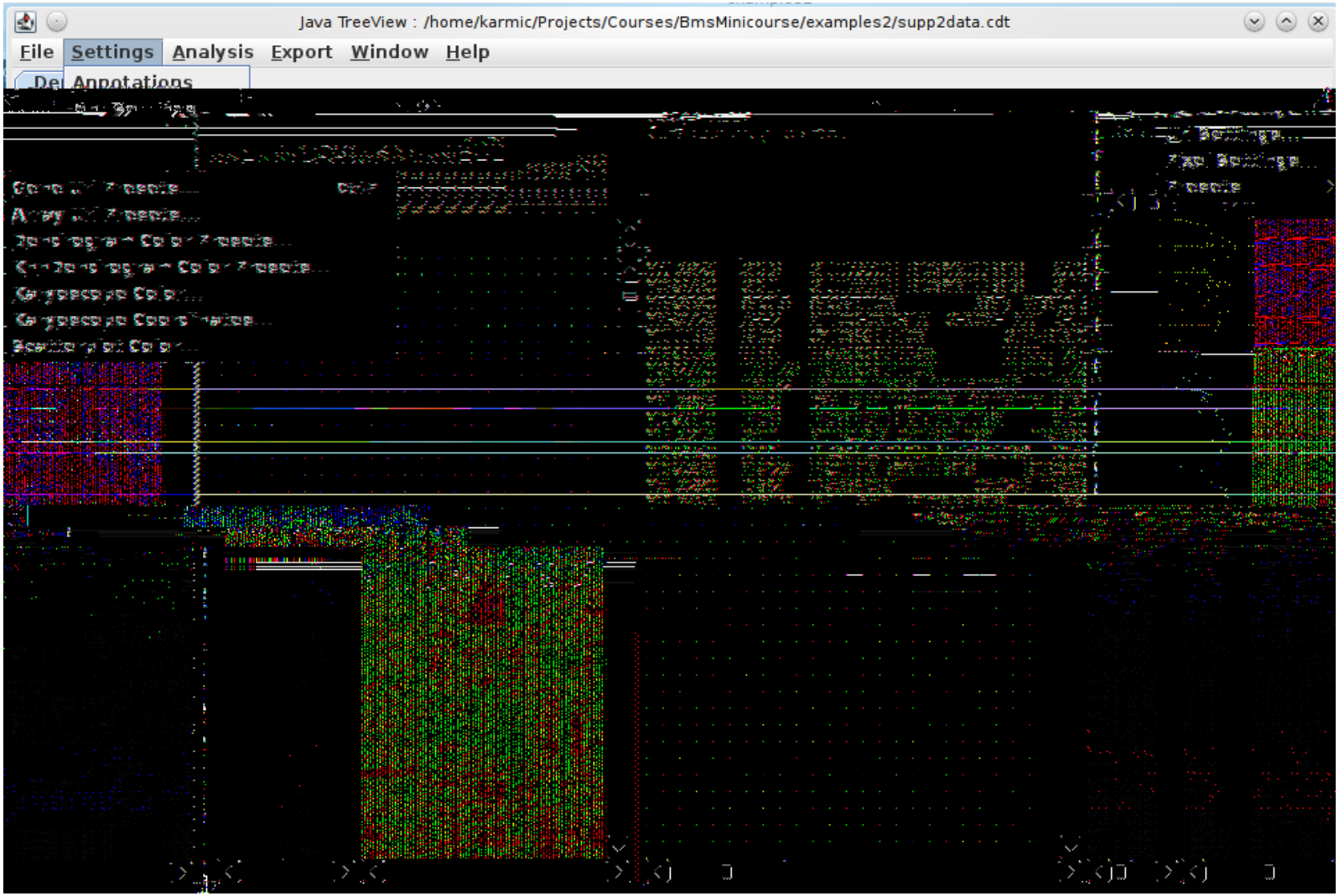
Select annotation columns



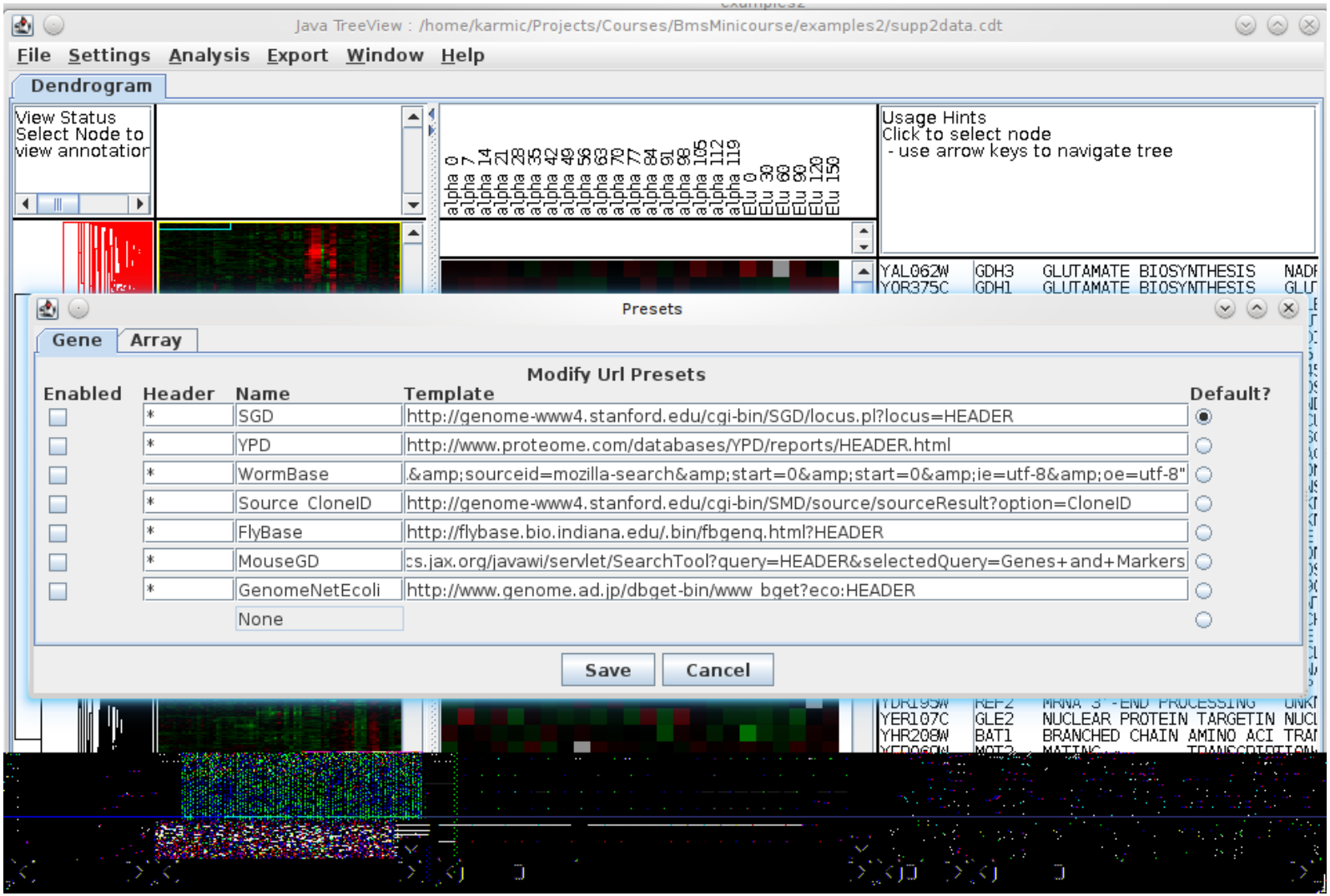
Select annotation columns



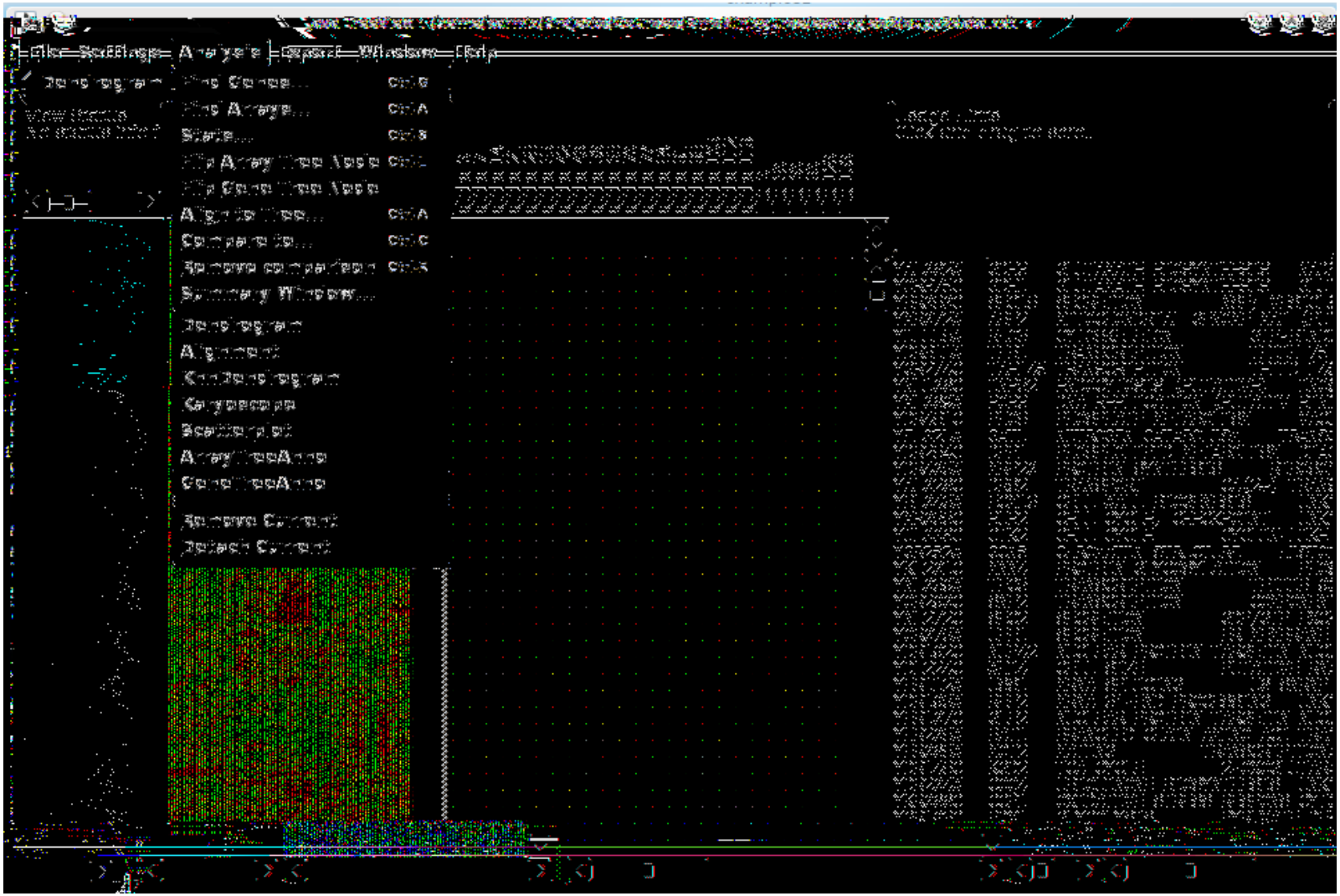
Select URL for gene annotations



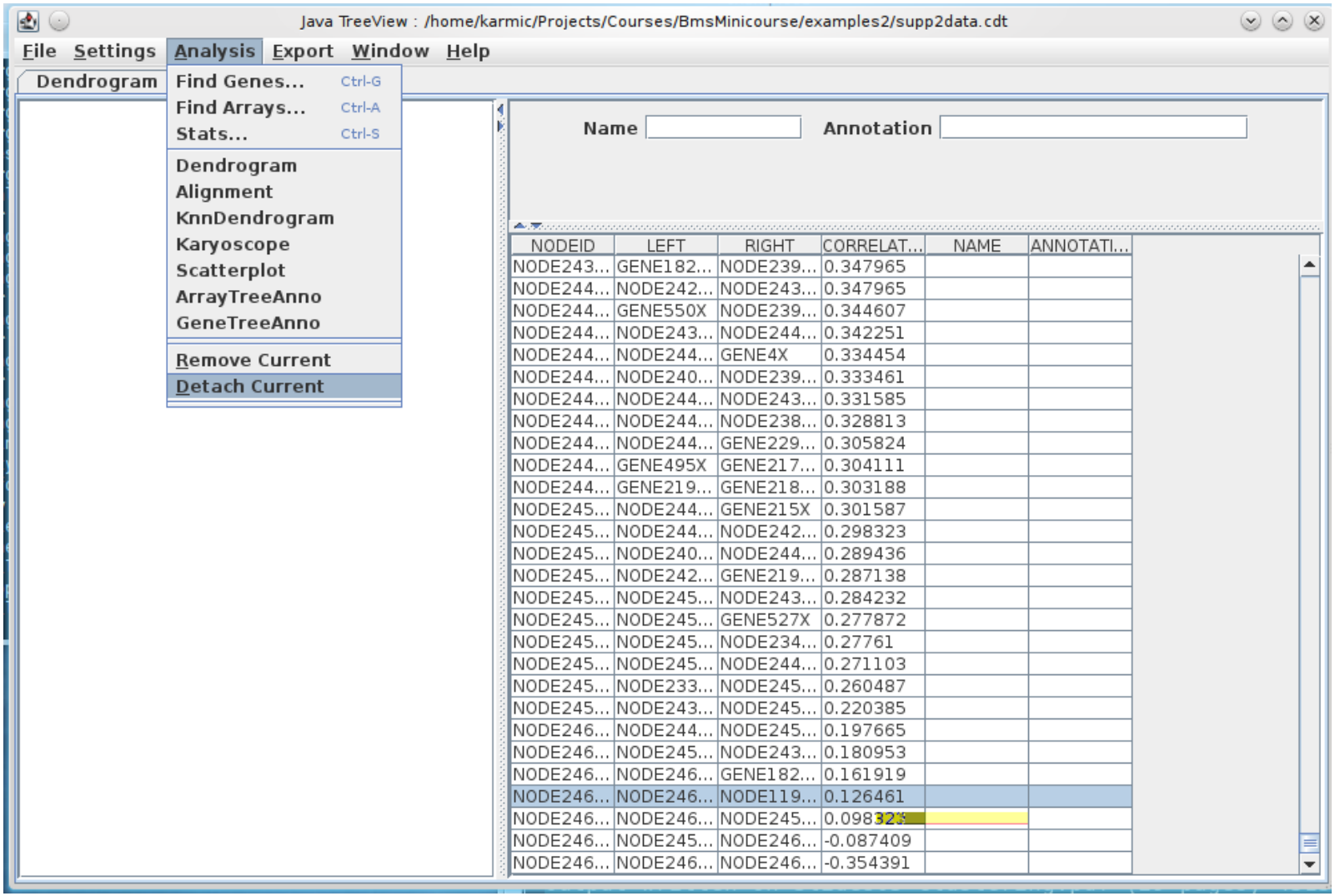
Select URL for gene annotations



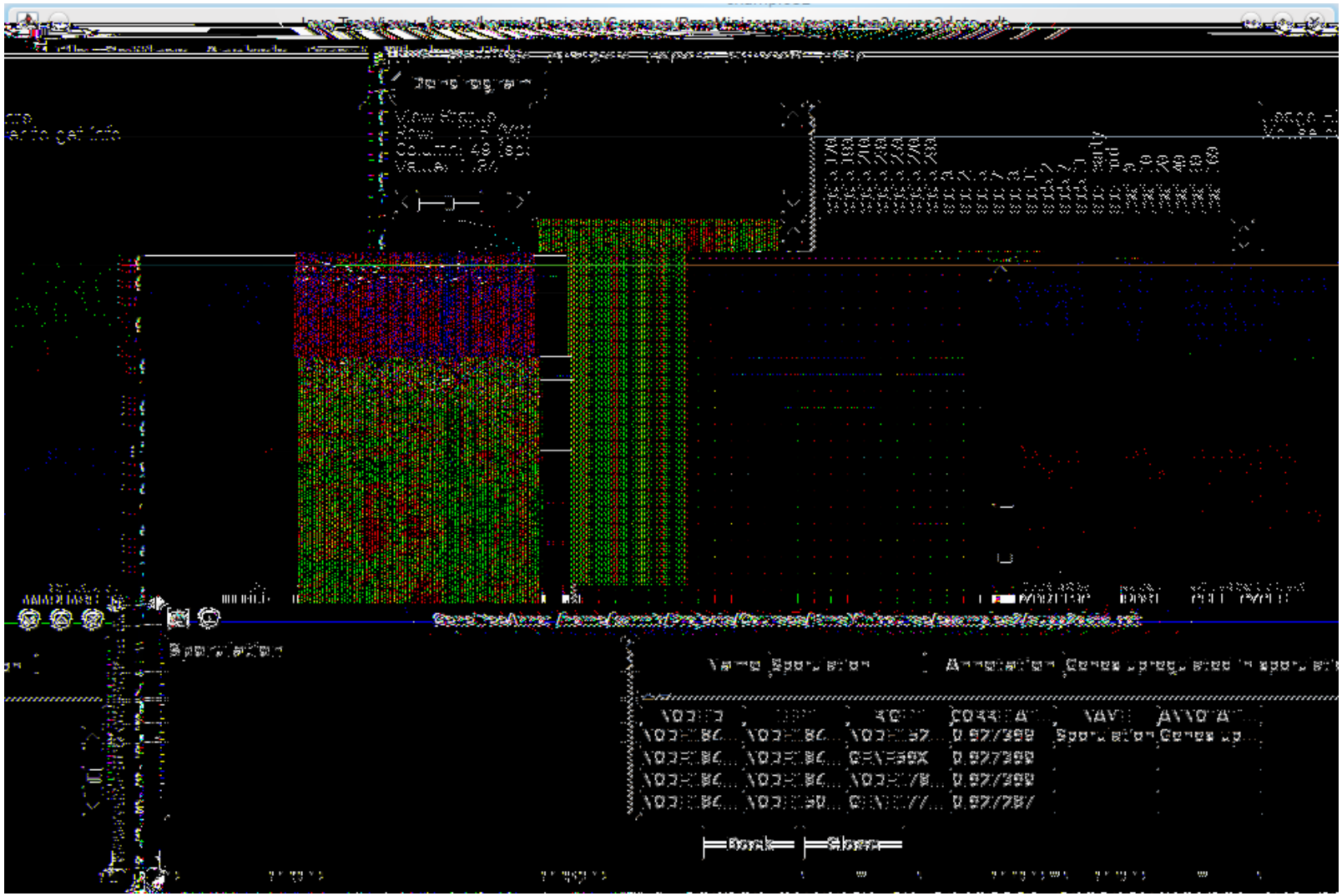
Activate and detach annotation window



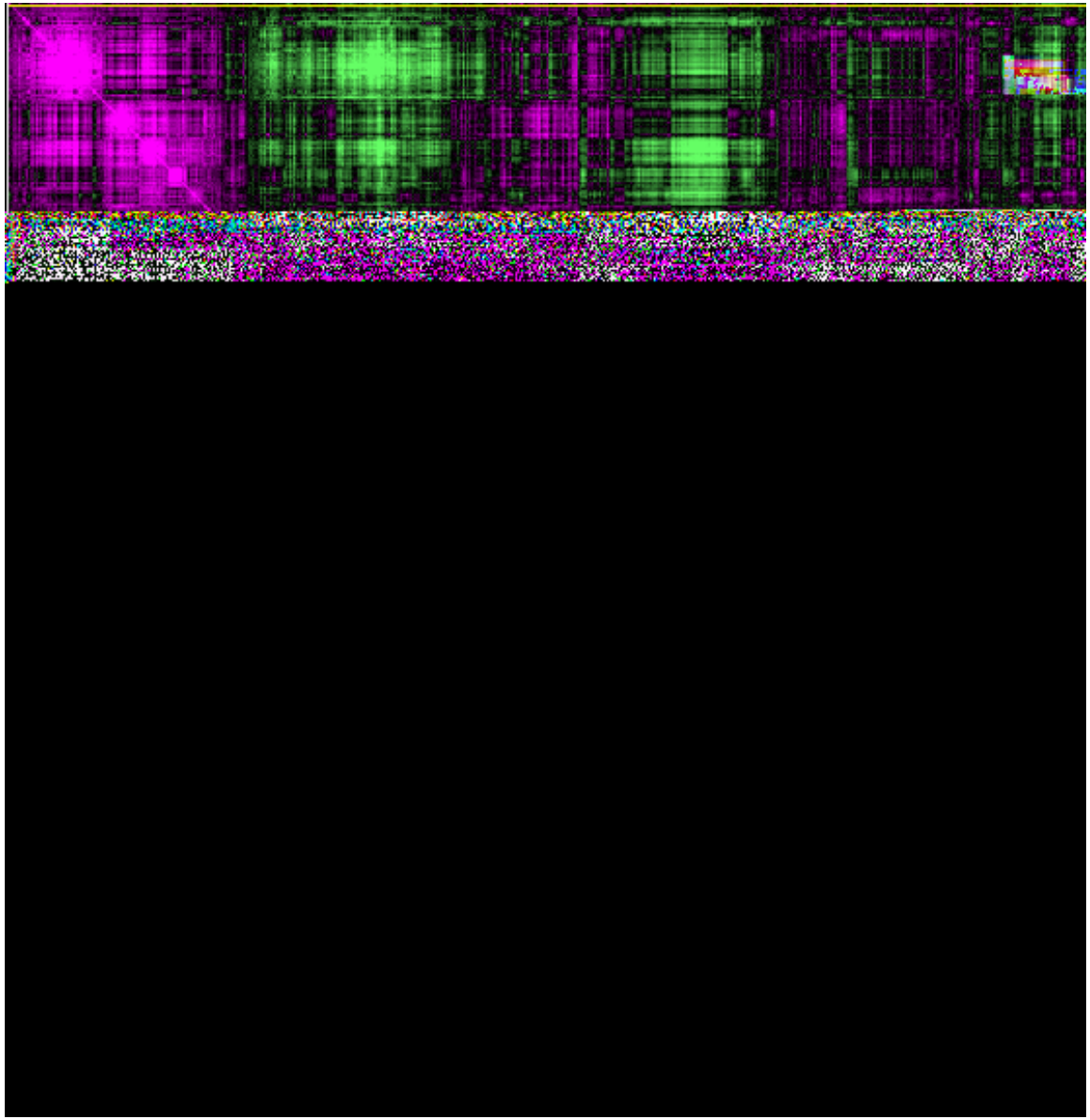
Activate and detach annotation window



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)))
```

Homework

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

Explore the gure 2 data with Cluster3 and JavaTreeView.

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

Whiteboard Image