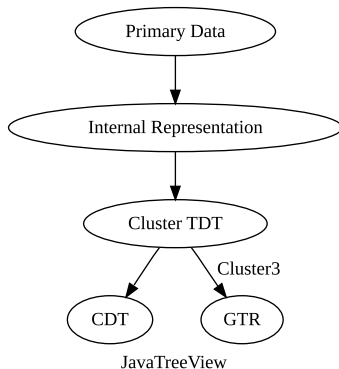


Practical Bioinformatics

Mark Voorhies

4/25/2011

Clustering protocol



How can we discover how we should format a file?

Whiteboard Image

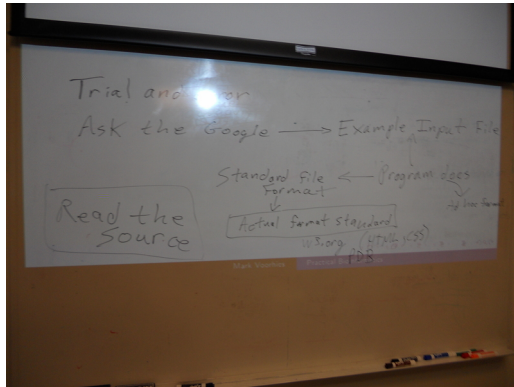
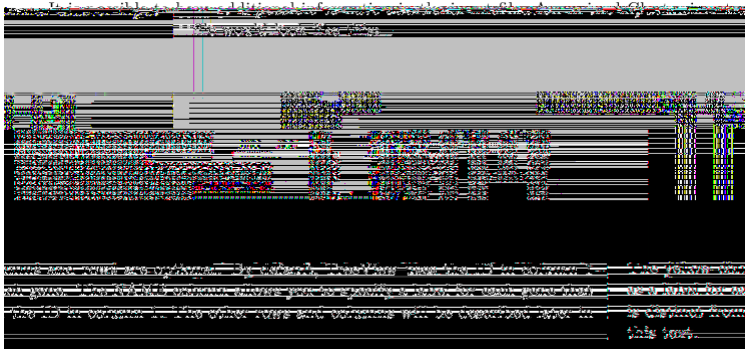


Figure 2.16. Screenshot of Generalized CDT file in Excel

P12_color3.cdt										
	A	B	C	D	E	F	G	H	I	J
1	GID	YCRF	NAME	PlateID	Spot	FGCOLOR	Sector	GWEIGHT	yB4n053 5 h	yB4n059 7 h
2	FGCOLOR					#000000			#999900	#999905
3	EWEIGHT					#000000		1	1	1
4	GENE5998X	YML051W	1709 GAL80	25570	7590	#000000	40	1	0.07	-0.27
5	GENE6516X	YNL064C	2407 YDJ1	25571	8170	#000000	43	1	0.94	0.6
6	GENE6229X	YBR035C	3774 PDX3	25558	7853	#000000	41	1	0.38	-0.54
7	GENE6272X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
8	GENE6273X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
9	GENE6274X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
10	GENE6275X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
11	GENE6276X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
12	GENE6277X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
13	GENE6278X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
14	GENE6279X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
15	GENE6280X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
16	GENE6281X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
17	GENE6282X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
18	GENE6283X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
19	GENE6284X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
20	GENE6285X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
21	GENE6286X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
22	GENE6287X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
23	GENE6288X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
24	GENE6289X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
25	GENE6290X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
26	GENE6291X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
27	GENE6292X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
28	GENE6293X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
29	GENE6294X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
30	GENE6295X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
31	GENE6296X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
32	GENE6297X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
33	GENE6298X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
34	GENE6299X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0
35	GENE6300X	YGL115W	5593 TPO3	25564	7630	#000000	40	1	0.85	0

RTFM: Cluster3



How does Cluster3 handle newlines?

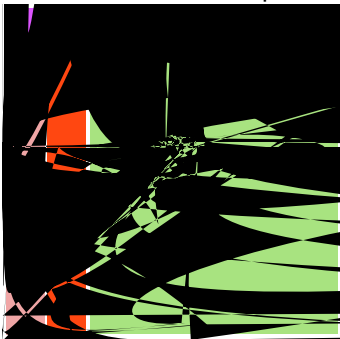
```
static char* GetLine(FILE* inputfile)
/* The function GetLine reads one line from the inputfile , and returns it as a
 * null-terminated string. If inputfile is at EOF, a null pointer is returned.
 * The calling routine should free the char* returned by GetLine.
 */
{ int c;
  int n = 0;
  int size = 1023;
  char* line = malloc((size+1)*sizeof(char));
  while ((c = getc(inputfile))!=EOF && c!='\r' && c!='\n')
  { if (n == size)
    { size *= 2;
      line = realloc(line ,(size+1)*sizeof(char));
    }
    line[n] = (char)c;
    n++;
  }
  if (c=='\r')
  { c = getc(inputfile);
    if (c!='\n' && c!=EOF) ungetc(c,inputfile);
  }
  if (n==0 && c==EOF)
  { free(line);
    return 0;
  }
  line[n] = '\0';
  line = realloc(line ,(n+1)*sizeof(char));
  return line;
}
```

How does Cluster3 handle delimiters?

```
static char* tokenize(char* s)
{
    char* p = s;
    while (1)
    {
        if (*p=='\0') return NULL;
        if (*p=='\t')
        {
            *p = '\0';
            return p+1;
        }
        p++;
    }
    /* Never get here */
    return NULL;
}
```

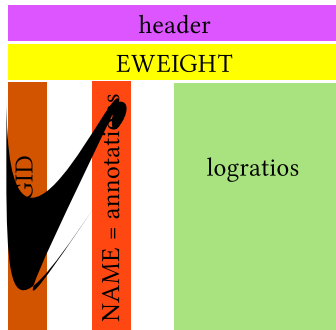
The CDT file format

Minimal CLUSTER input



- Tab delimited (`\t`)
- UNIX newlines (`\n`)
- Missing values $\rightarrow$ empty cells

Cluster3 CDT output



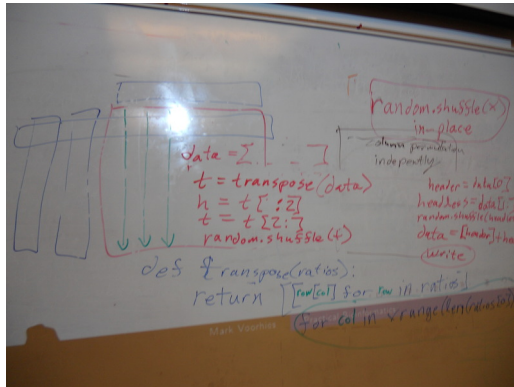
Be liberal in what you accept, strict in what you emit

```
def writerow(out, row):
    out.write(
        # Use tab delimiters
        "\t".join(
            # Normalize all whitespace (including \t, \r, and \n) and "
            # to "normal" spaces.
            [re.sub(r'[\s]',' ',str(i)) for i in row])
        # Use UNIX-style newlines
        +"\n")

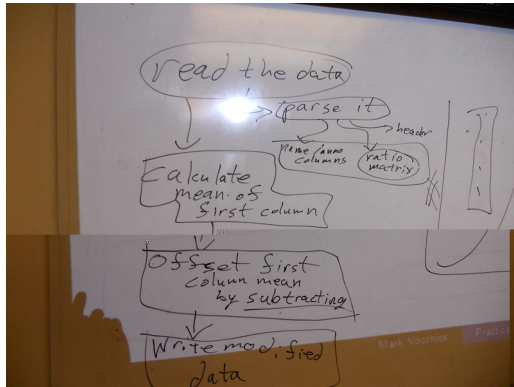
def writectd(filename, matrix,
              row_names, row_annotations, columns):
    out = open(filename, "w")
    writerow(out, ["UNIQID", "NAME"] + columns)
    for (uniqid, name, row) in zip(row_names, row_annotations, matrix):
        writerow(out, [uniqid, name] + row)
    out.close()
```

Strategies for Shuffling

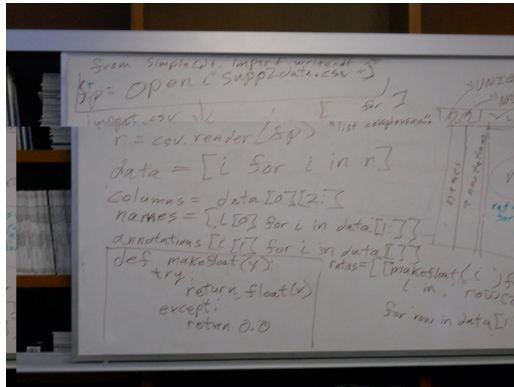
Whiteboard Image



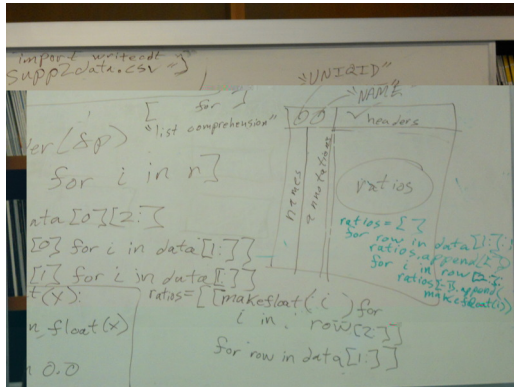
Whiteboard Image



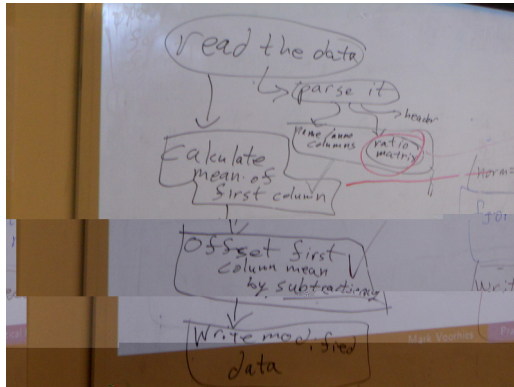
Whiteboard Image



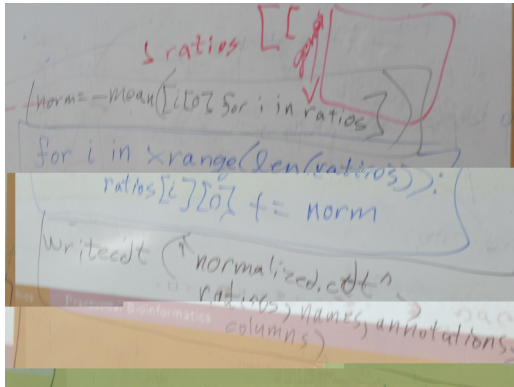
Whiteboard Image



Whiteboard Image



Whiteboard Image



Handwritten code on a whiteboard. The code is written in blue ink. A red box highlights the first line of code, and a red arrow points to the variable 'ratios' in the second line. The code is as follows:

```
norm = mean([c[0] for i in ratios])  
for i in xrange(len(ratios)):  
    ratios[i][0] += norm  
writecdt(normalized.cdt,  
          ratios, names, annotations,  
          columns)
```

Below the code, there is a pink sticky note with the text: "Practical Bioinformatics".

Using the Cluster3 GUI

Gene Cluster 3.0

File Help

File loaded

Job name

Data set has Rows Columns

Filter Genes

☐ % Present $\geq$ 80

☐ SD (Gene Vector) 2.0

☐ At least 1 observations with $\text{abs}(\text{Val}) \geq$ 2.0

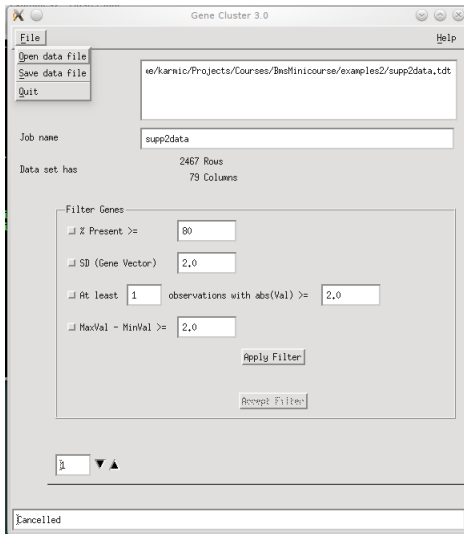
☐ MaxVal - MinVal $\geq$ 2.0

Apply Filter

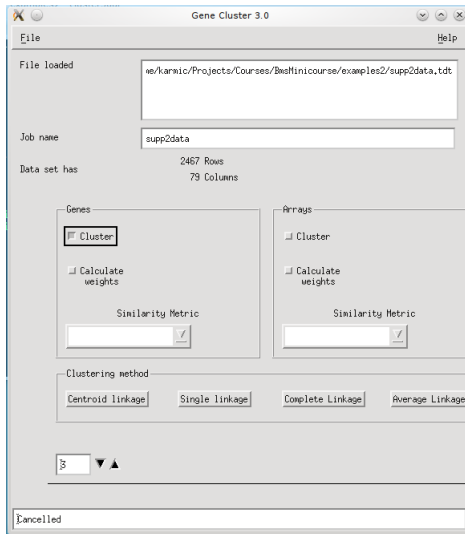
Accept Filter

1 ▼▲

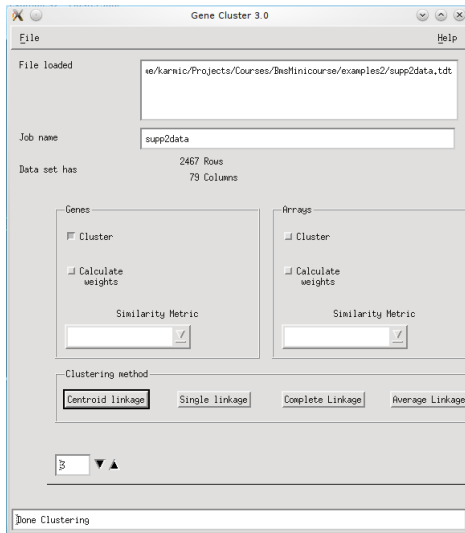
Load your data



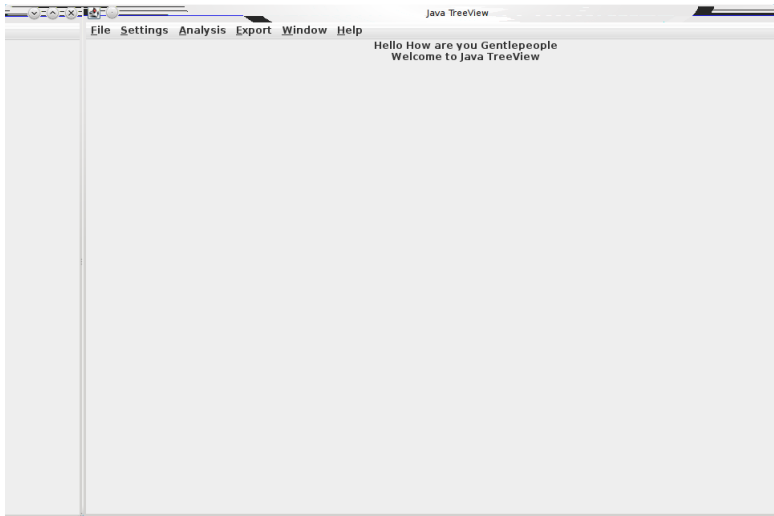
Choose distance function



Choose linking method

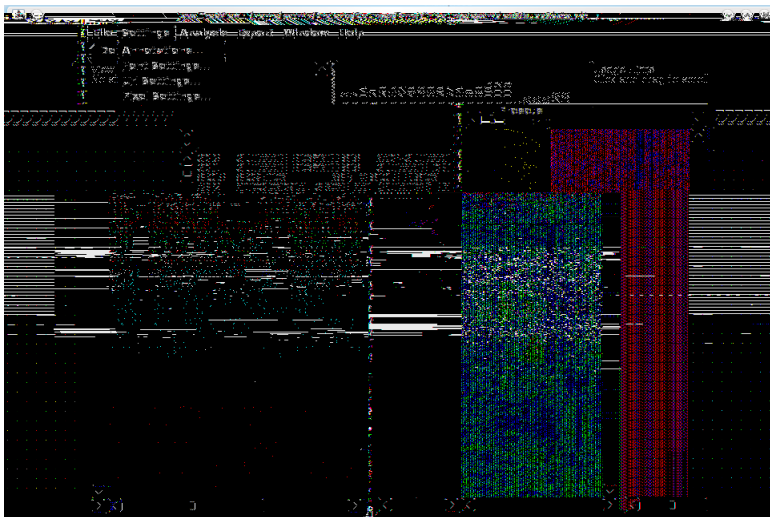


Using JavaTreeView



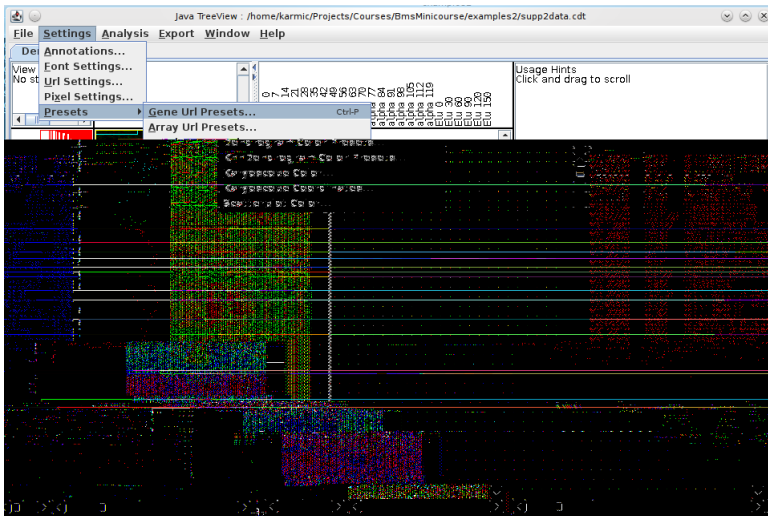
[illegible]

Select annotation columns

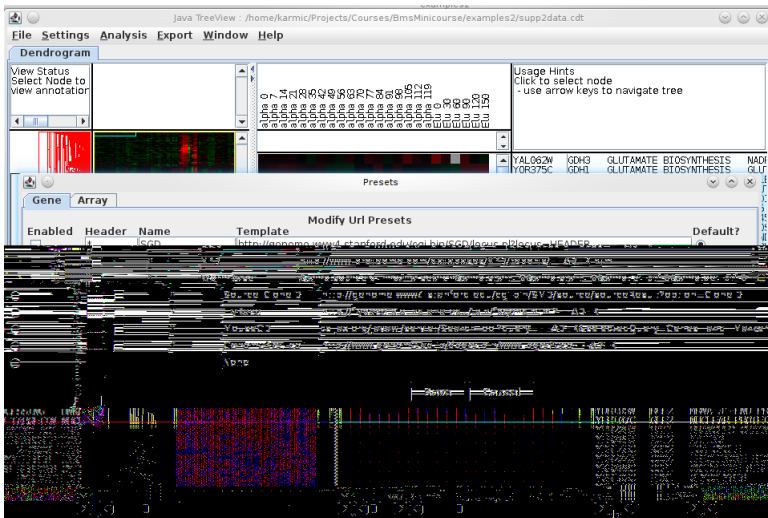


The screenshot shows the Java TreeView application. The main window displays a tree structure with a large, multi-colored rectangular area in the center, likely representing a complex data structure or a large tree. To the right, a detailed view of a selected node is shown, displaying a list of items with various attributes and a 'Class' label. The interface includes standard window controls (minimize, maximize, close) and a menu bar at the top.

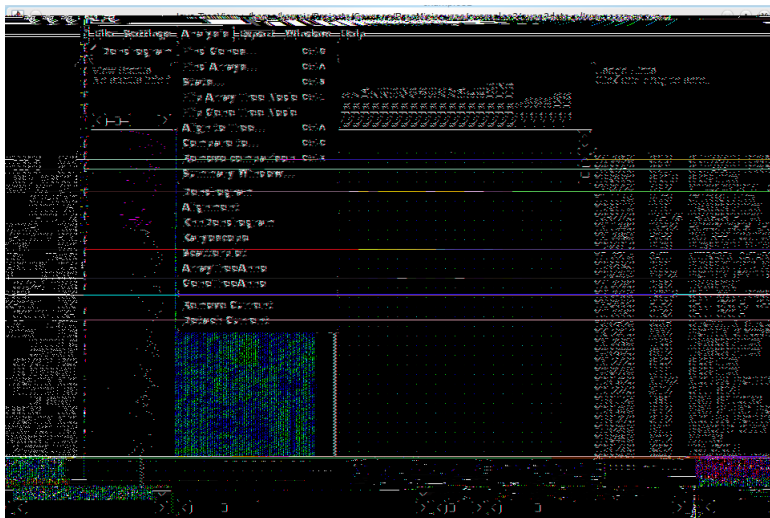
Select URL for gene annotations



Select URL for gene annotations



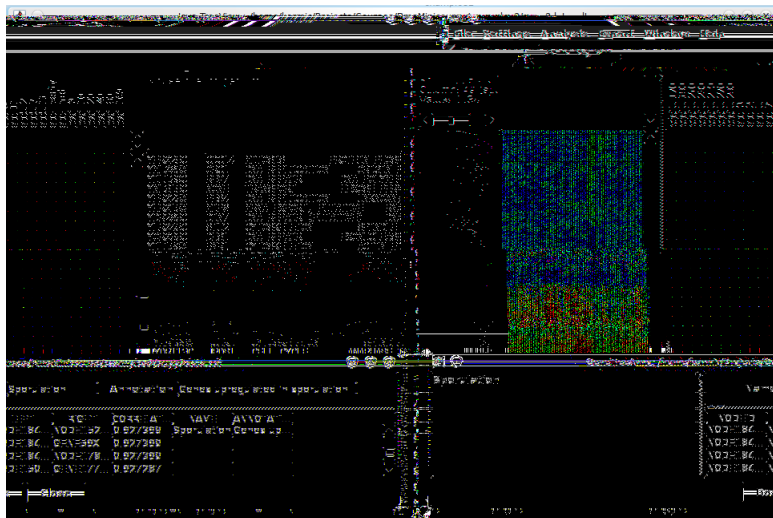
Activate and detach annotation window



Activate and detach annotation window

[illegible]

Activate and detach annotation window



Calculating the Distance Matrix

Pearson similarity

$$s(x, y) = \frac{\sum_i^N (x_i - x_{offset})(y_i - y_{offset})}{\sqrt{\sum_i^N (x_i - x_{offset})^2} \sqrt{\sum_i^N (y_i - y_{offset})^2}} \quad (1)$$

Pearson distances

Pearson similarity

$$s(x, y) = \frac{\sum_i^N (x_i - x_{offset})(y_i - y_{offset})}{\sqrt{\sum_i^N (x_i - x_{offset})^2} \sqrt{\sum_i^N (y_i - y_{offset})^2}} \quad (1)$$

Pearson distance

$$d_{uncentered}(x, y) = 1 - s(x, y) \quad (2)$$

Pearson distances

Pearson similarity

$$s(x, y) = \frac{\sum_i^N (x_i - x_{\text{offset}})(y_i - y_{\text{offset}})}{\sqrt{\sum_i^N (x_i - x_{\text{offset}})^2} \sqrt{\sum_i^N (y_i - y_{\text{offset}})^2}} \quad (1)$$

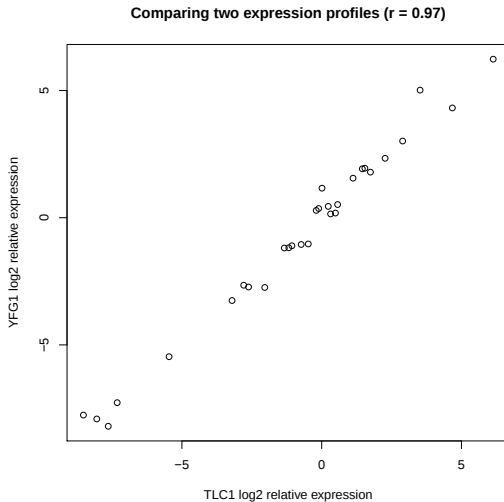
Pearson distance

$$d_{\text{uncentered}}(x, y) = 1 - s(x, y) \quad (2)$$

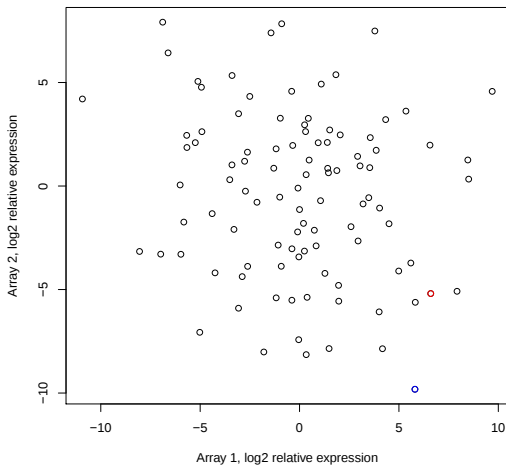
Euclidean distance

$$\frac{\sum_i^N (x_i - y_i)^2}{N} \quad (3)$$

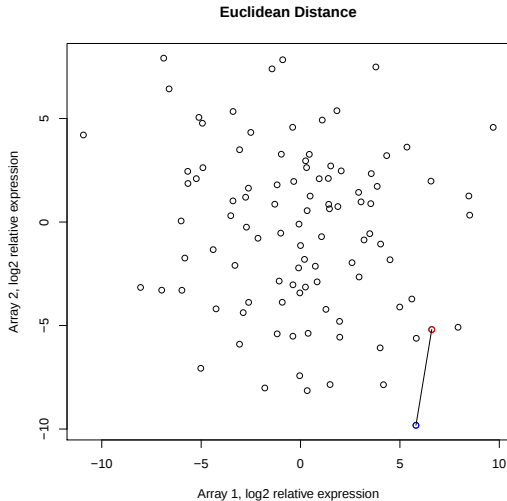
Comparing all measurements for two genes



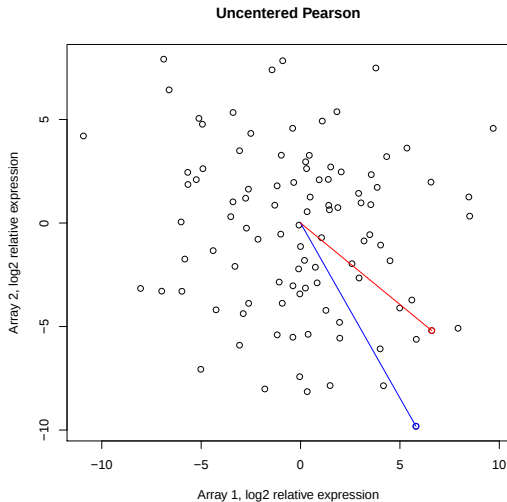
Comparing all genes for two measurements



Comparing all genes for two measurements



Comparing all genes for two measurements



Special cases for Pearson in Cluster3

```
int flag = 0;
/* flag will remain zero if no nonzero combinations of mask1 and mask2 are
 * found.
 */
int i;
for (i = 0; i < n; i++)
{ if (mask1[index1][i] && mask2[index2][i])
  { double term1 = data1[index1][i];
    double term2 = data2[index2][i];
    double w = weight[i];
    result += w*term1*term2;
    denom1 += w*term1*term1;
    denom2 += w*term2*term2;
    flag = 1;
  }
}
if (!flag) return 0.;
if (denom1==0.) return 1.;
if (denom2==0.) return 1.;
result = result / sqrt(denom1*denom2);
result = 1. - result;
return result;
```