

# Distance Metrics

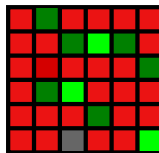
Mark Voorhies

4/27/2017

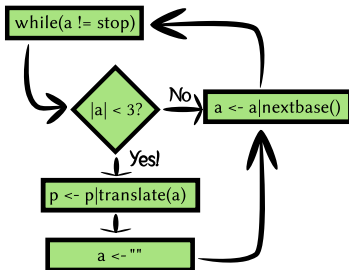
# Anatomy of a Programming Language

$f(x)$  

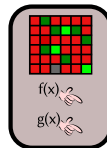
functions



data structures



control statements



objects

# Anatomy of a Programming Language

$f(x)$  

functions

```
def f(x,y):  
    return x*y  
  
from math import sqrt
```

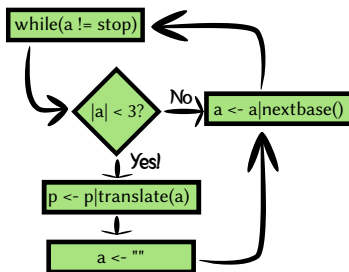
# Anatomy of a Programming Language

```
1  
1.2  
"my string"  
["my", "list"]  
my_file = open("my_file.txt")  
("my", "tuple")  
[["my", "multi"],  
 ["dimensional", "list"]]
```



data structures

# Anatomy of a Programming Language



```
while (a != -1):  
    a = x.find("ATG")  
  
for line in open("x"):  
    L.append(line[:-1])
```

control statements

# Anatomy of a Programming Language

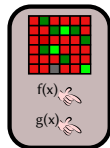
```
"GGGATGCATCAT".find("ATG")
```

```
L = [3,4,5]
```

```
L.append(7)
```

```
L += [6,7]
```

```
open("1.txt").readlines()
```

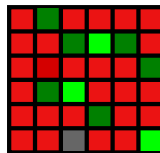


objects

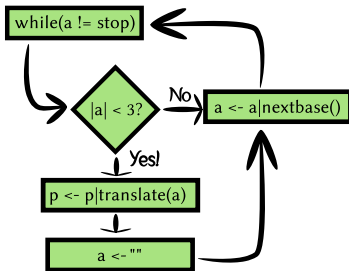
# Anatomy of a Programming Language

$f(x)$  

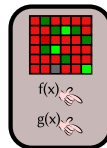
functions



data structures

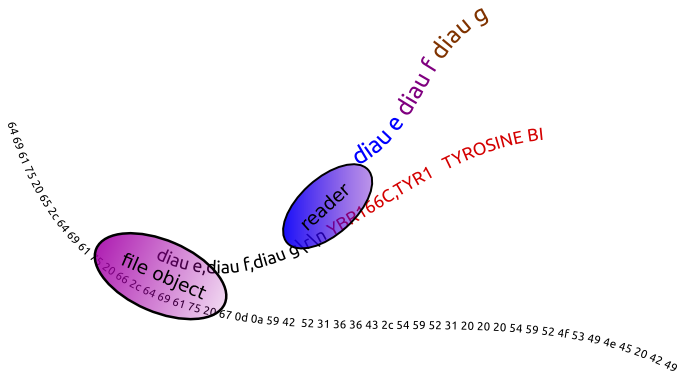


control statements



objects

# Generators are like polymerases: iterable but not indexable





# Fun with logarithms

In log space, multiplication and division become addition and subtraction:

$$\begin{aligned}\log(xy) &= \log(x) + \log(y) \\ \log(x/y) &= \log(x) - \log(y)\end{aligned}$$

# Fun with logarithms

In log space, multiplication and division become addition and subtraction:

$$\begin{aligned}\log(xy) &= \log(x) + \log(y) \\ \log(x/y) &= \log(x) - \log(y)\end{aligned}$$

Therefore, exponentiation becomes multiplication:

$$\log(x^y) = y \log(x)$$

# Fun with logarithms

In log space, multiplication and division become addition and subtraction:

$$\begin{aligned}\log(xy) &= \log(x) + \log(y) \\ \log(x/y) &= \log(x) - \log(y)\end{aligned}$$

Therefore, exponentiation becomes multiplication:

$$\log(x^y) = y \log(x)$$

Also, we can change of the base of a logarithm like so:

$$\log_A(x) = \log(x) / \log(A)$$

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

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

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

Pearson similarity

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

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

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

Pearson distance

$$d(x, y) = 1 - s(x, y)$$



Pearson similarity

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

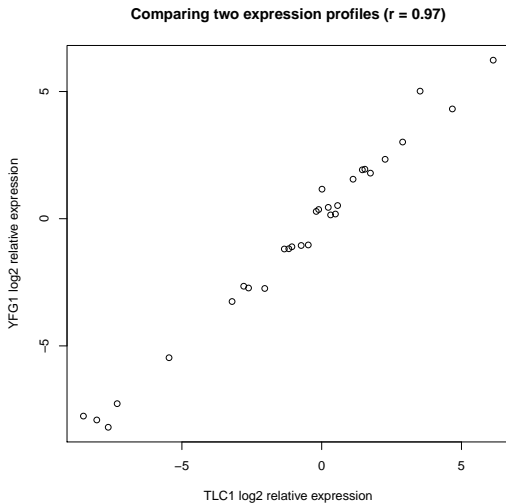
Pearson distance

$$d(x, y) = 1 - s(x, y)$$

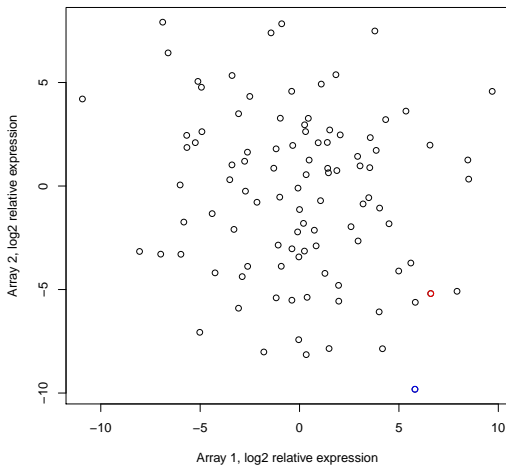
Euclidean distance

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

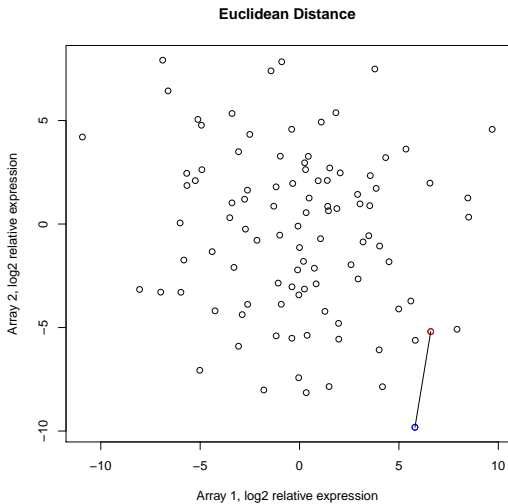
# Comparing all measurements for two genes



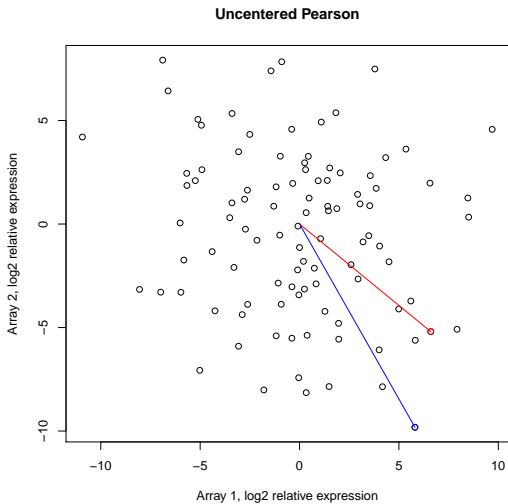
# Comparing all genes for two measurements



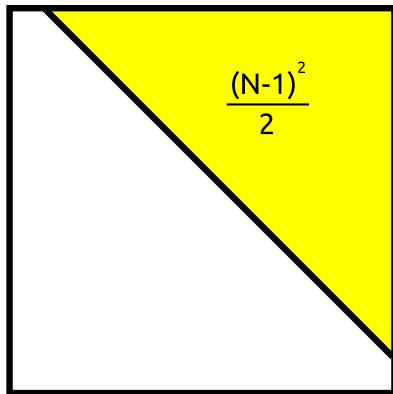
# Comparing all genes for two measurements



# Comparing all genes for two measurements

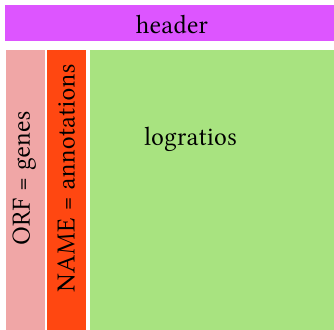


# Measure all pairwise distances under distance metric

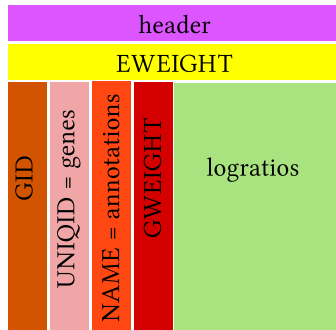


# The CDT file format

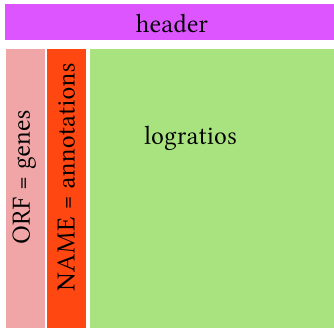
## Minimal CLUSTER input



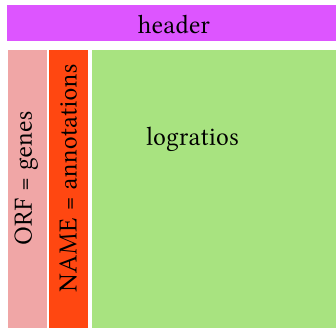
## Cluster3 CDT output



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







```
[["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, ...],
 ...]
]
```

# Homework

- 1 Install biopython via Canopy (*or whatever you're using. If this doesn't work, install Cluster3*)
- 2 Write a function to calculate all pairwise Pearson correlations for the yeast expression profiles.
- 3 Save the results of your pairwise correlation calculation in the CDT format described in the JavaTreeView manual.
- 4 Read PNAS 95:14863
- 5 Try the first two problems, replacing the Pearson correlation with the distance metric from the PNAS paper or with one of the distance metrics from the Cluster3 manual.