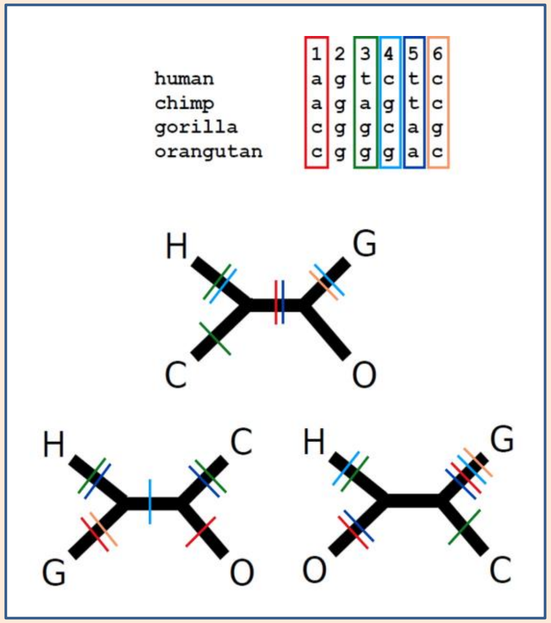
A quick review

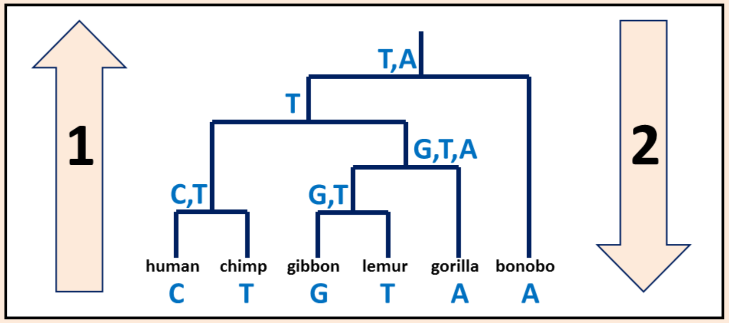
- **The parsimony principle:**
  - Find the tree that requires the fewest evolutionary changes!
- A fundamentally different method:
  - Search rather than reconstruct
- **Parsimony algorithm**
  1. Construct all possible trees
  2. For each site in the alignment and for each tree count the minimal number of changes required
  3. Add sites to obtain the total number of changes required for each tree
  4. Pick the tree with the lowest score



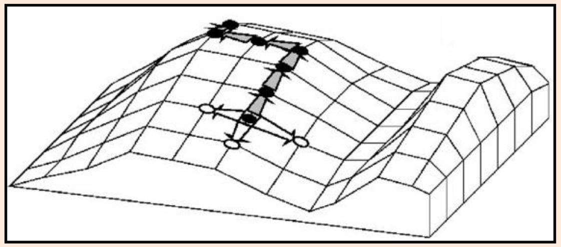
	1	2	3	4	5	6
human	a	g	t	c	t	c
chimp	a	g	a	g	t	c
gorilla	c	g	g	c	a	g
orangutan	c	g	g	g	a	c

A quick review – cont’

- Small vs. large parsimony



- Fitch’s algorithm:
  1. **Bottom-up phase**: Determine the set of possible states
  2. **Top-down phase**: Pick a state for each internal node



- Searching the tree space:
  - Exhaustive search, branch and bound
  - Hill climbing with Nearest-Neighbor Interchange
- Branch confidence and bootstrap support

Phylogenetic trees: Summary

Parsimony Trees:

1) *Construct all possible trees* ***or search the space of possible trees***
2) *For each site in the alignment and for each tree count the minimal number of changes required* ***using Fitch's algorithm***
3) *Add all sites up to obtain the total number of changes for each tree*
4) *Pick the tree with the lowest score*

Distance Trees:

1) *Compute pairwise corrected distances.*
2) *Build tree by sequential clustering algorithm (UPGMA or Neighbor-Joining).*
3) *These algorithms don't consider all tree topologies, so they are very fast, even for large trees.*

Maximum-Likelihood Trees:

1) *Tree evaluated for likelihood of data given tree.*
2) *Uses a specific model for evolutionary rates (such as Jukes-Cantor).*
3) *Like parsimony, must search tree space.*
4) *Usually most accurate method but slow.*

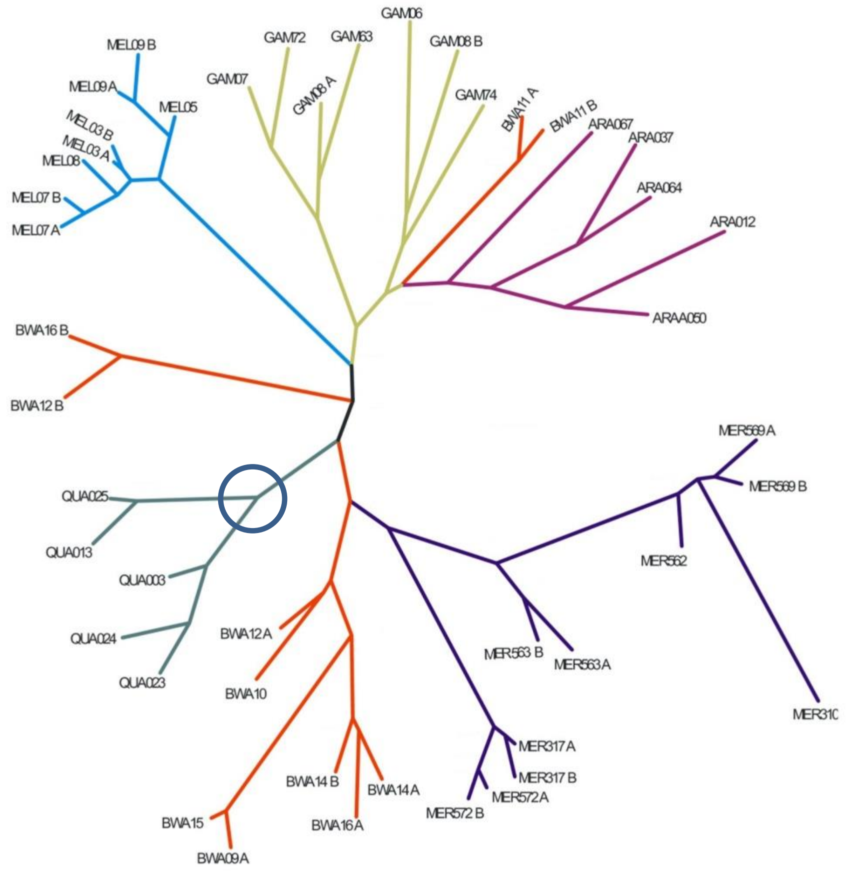
Branch confidence

How certain are we that this is the correct tree?

Can be reduced to many simpler questions - how certain are we that each **branch point** is correct?

For example, at the circled branch point, how certain are we that the three subtrees have the correct content:

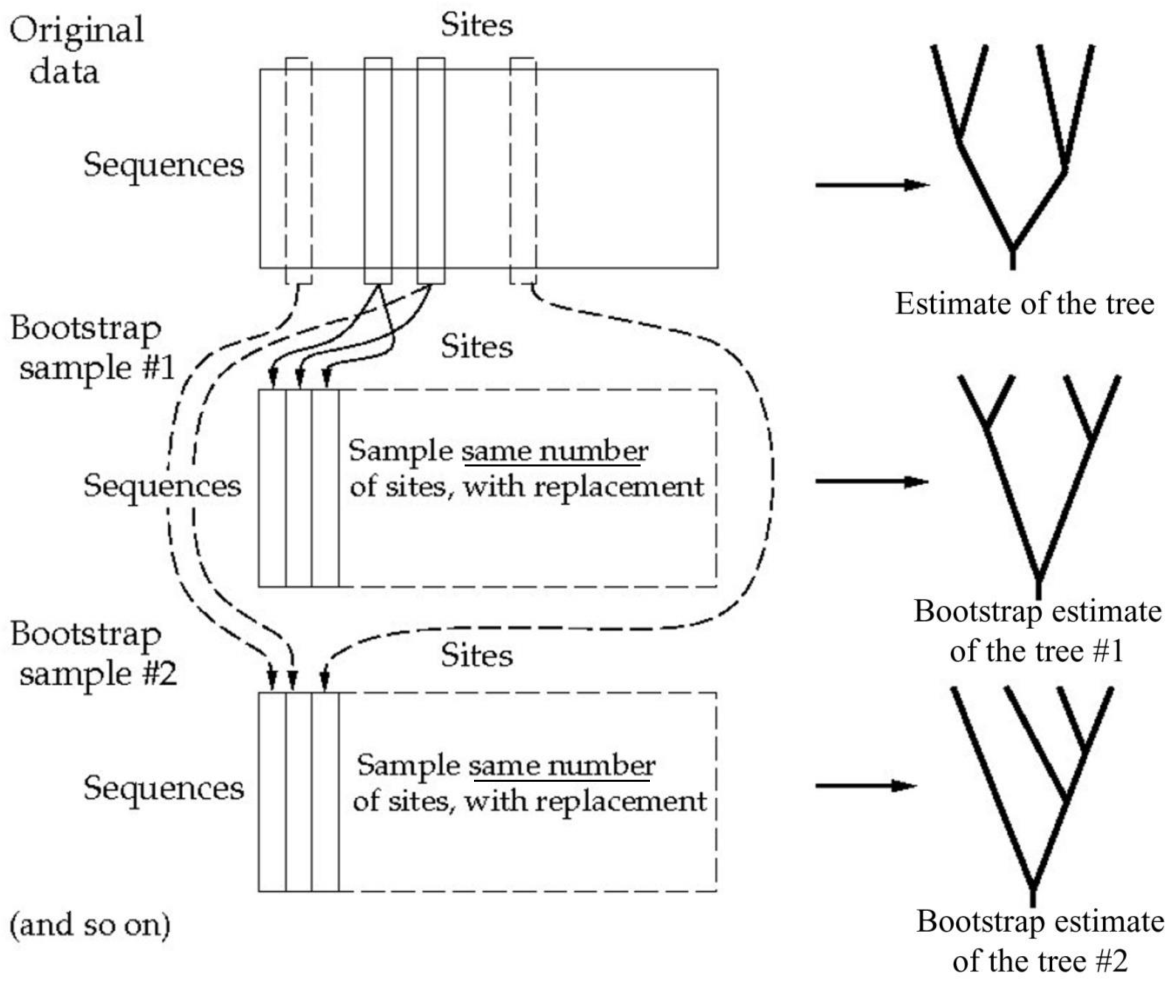
subtree1: QUA025, QUA013
Subtree2: QUA003, QUA024, QUA023
Subtree3: everything else



Bootstrap support

Most commonly used branch support test:

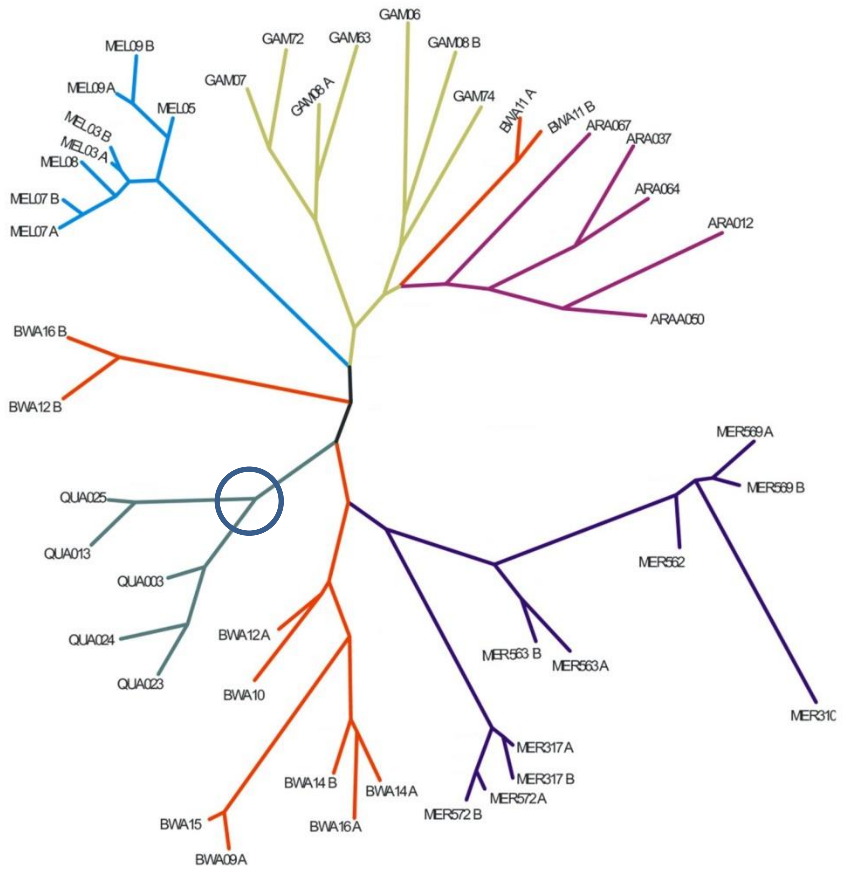
1. *Randomly sample alignment sites (with replacement).*
2. *Use sample to estimate the tree.*
3. *Repeat many times.*


Original data
Sites
Sequences
Estimate of the tree
Bootstrap sample #1
Sites
Sequences
Sample same number of sites, with replacement
Bootstrap estimate of the tree #1
Bootstrap sample #2
Sites
Sequences
Sample same number of sites, with replacement
Bootstrap estimate of the tree #2
(and so on)

(sample with replacement means that a sampled site remains in the source data after each sampling, so that some sites will be sampled more than once)

Bootstrap support

For each branch point on the computed tree, count what fraction of the bootstrap trees have the same subtree partitions (regardless of topology within the subtrees).

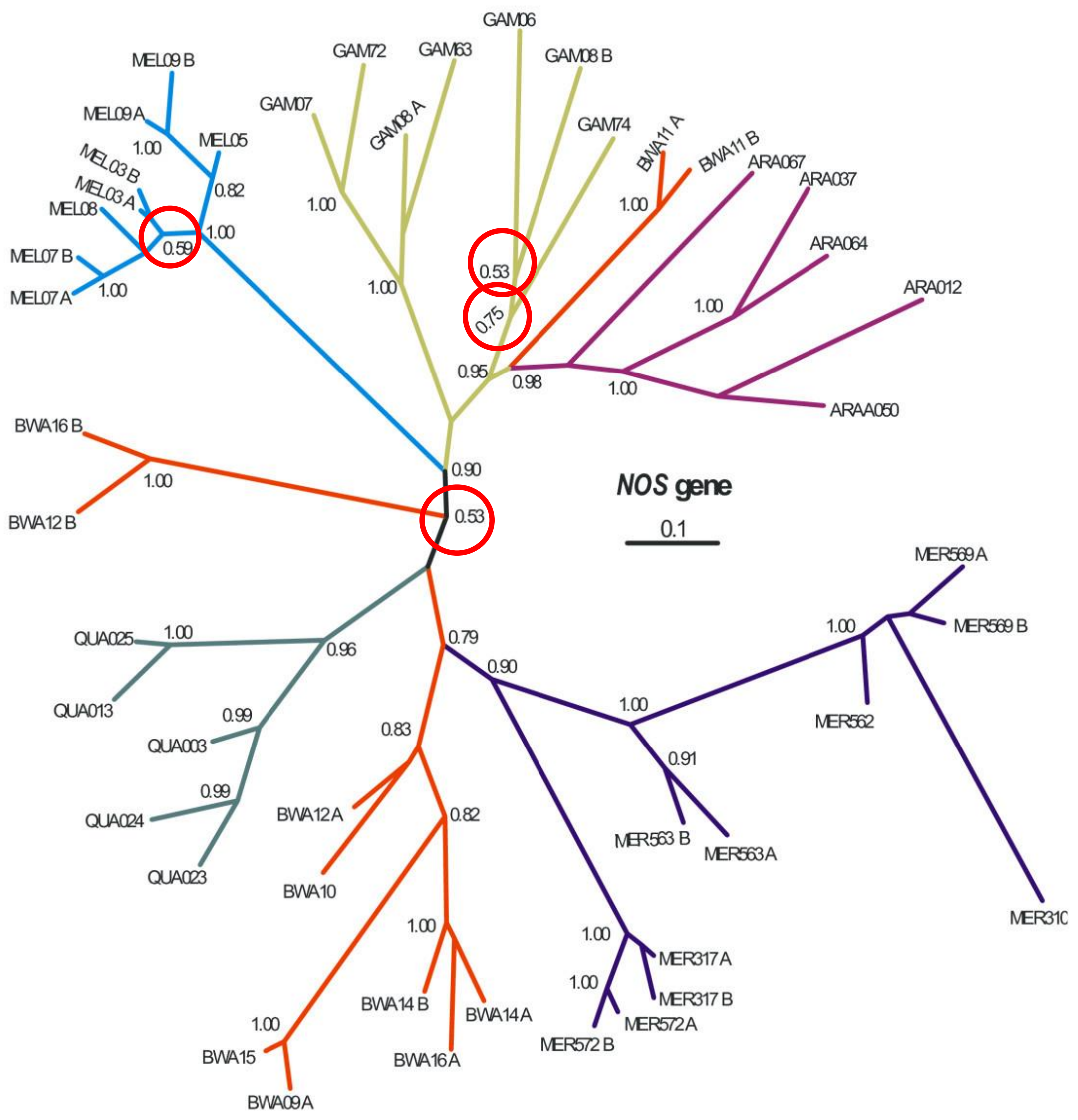


For example at the circled branch point, what fraction of the bootstrap trees have a branch point where the three subtrees include:

Subtree1: QUA025, QUA013
Subtree2: QUA003, QUA024, QUA023
Subtree3: everything else

This fraction is the **bootstrap support** for that branch.

Original tree figure with branch supports
(here as fractions, also common to give % support)



low-confidence branches are marked

Clustering

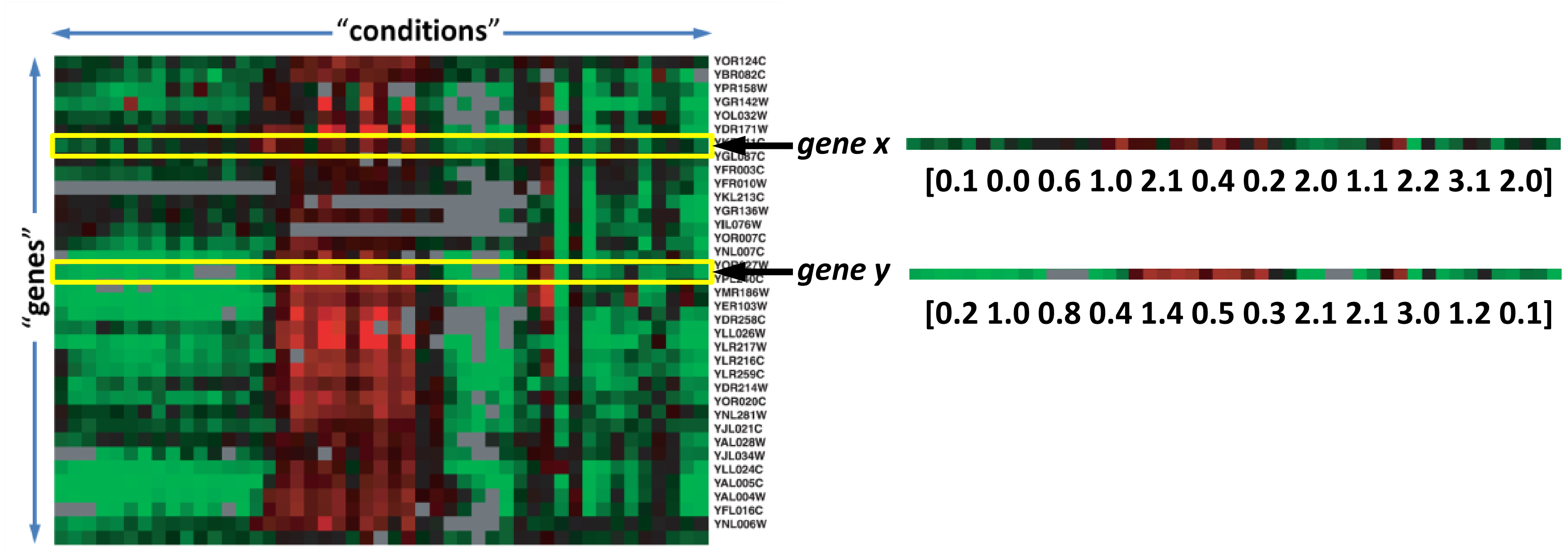
Genome 373

Genomic Informatics

Elhanan Borenstein

Some slides adapted from Jacques van Helden

Many different data types, same structure

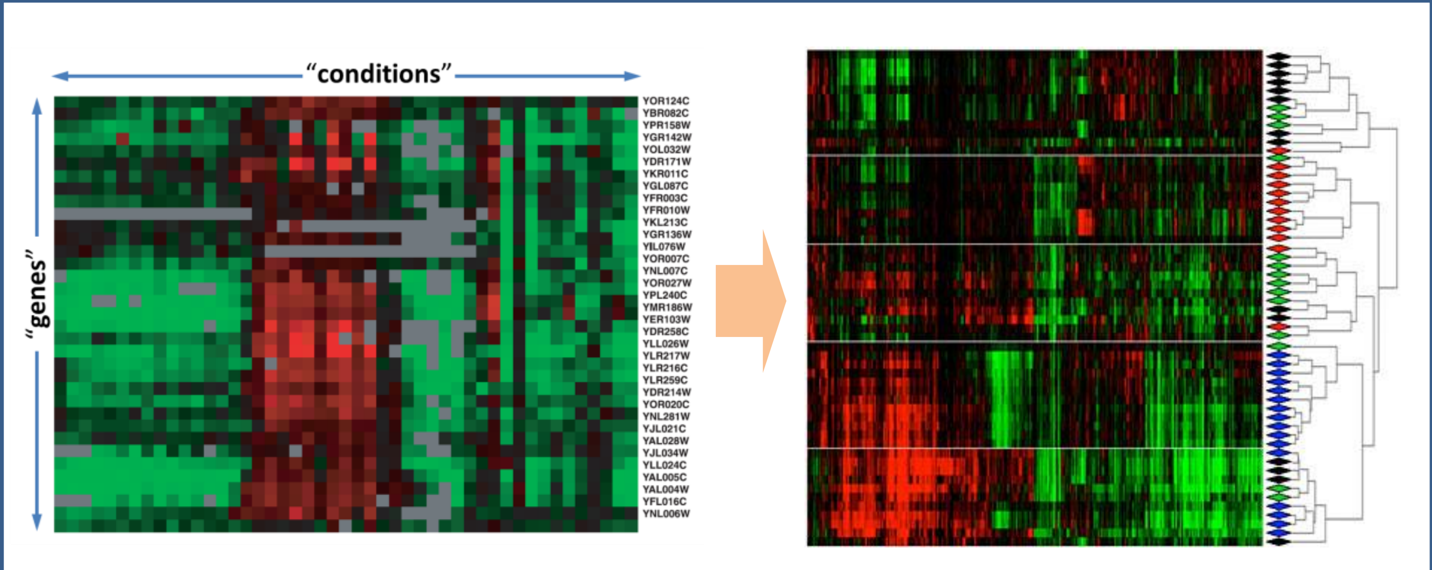


[0.1 0.0 0.6 1.0 2.1 0.4 0.2 2.0 1.1 2.2 3.1 2.0]

[0.2 1.0 0.8 0.4 1.4 0.5 0.3 2.1 2.1 3.0 1.2 0.1]

The clustering problem

- **The goal of gene clustering process is to partition the genes into distinct sets such that genes that are assigned to the same cluster are "similar", while genes assigned to different clusters are "non-similar".**



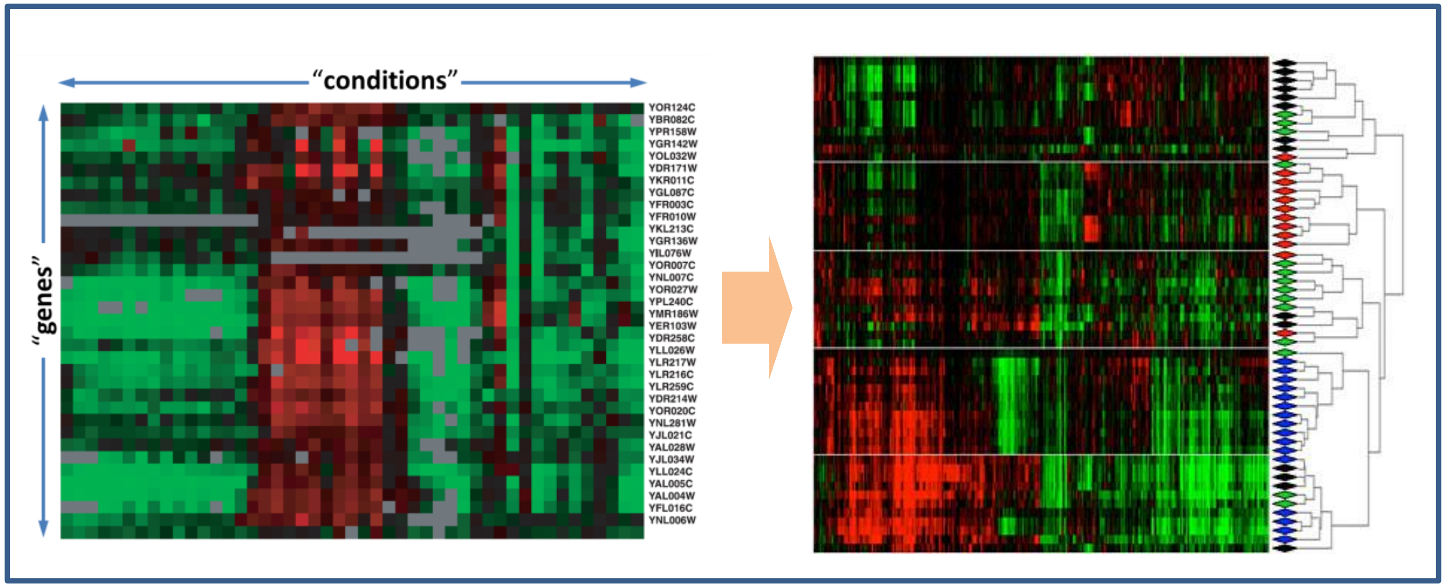
Why clustering

Why clustering

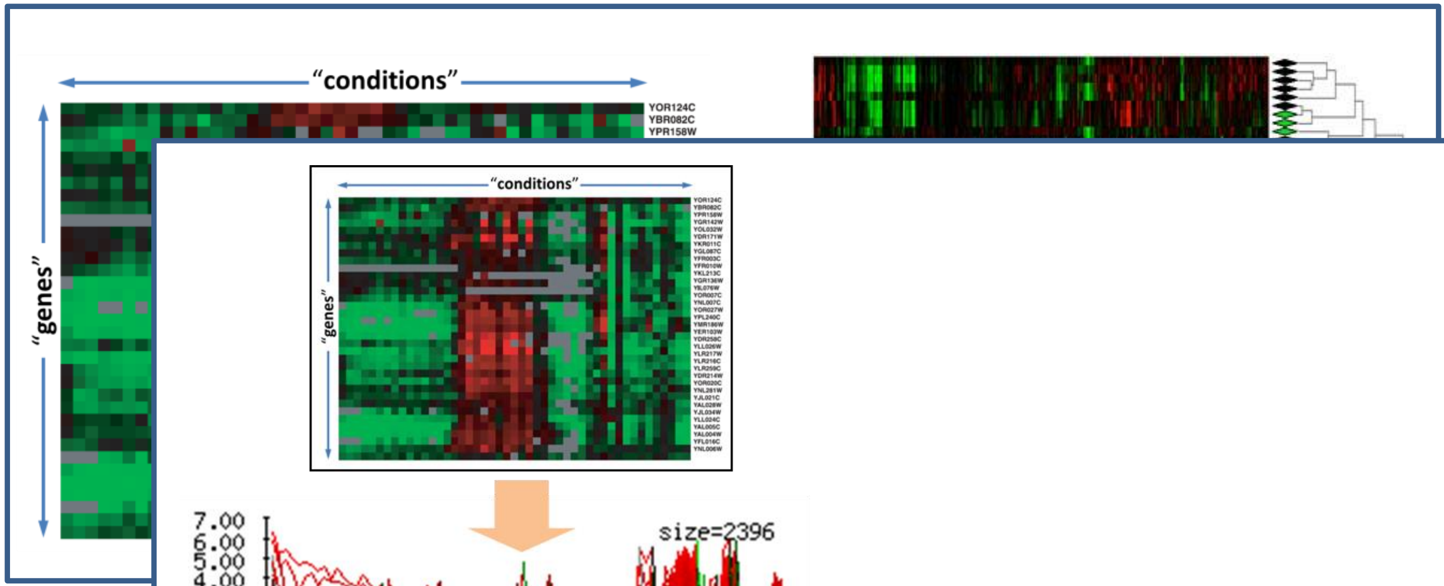
- Clustering genes or conditions is a basic tool for the analysis of expression profiles, and can be useful for many purposes, including:
  - Inferring functions of unknown genes
    (assuming a similar expression pattern implies a similar function).
  - Identifying disease profiles
    (tissues with similar pathology should yield similar expression profiles).
  - Deciphering regulatory mechanisms: co-expression of genes may imply co-regulation.
  - **Reducing dimensionality.**

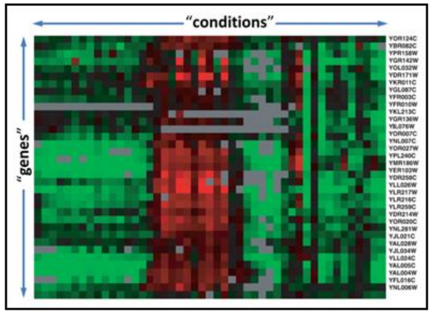
Why is clustering a hard computational problem?

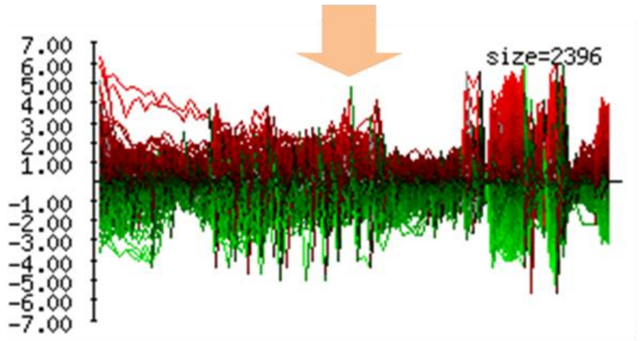
Different views of clustering …



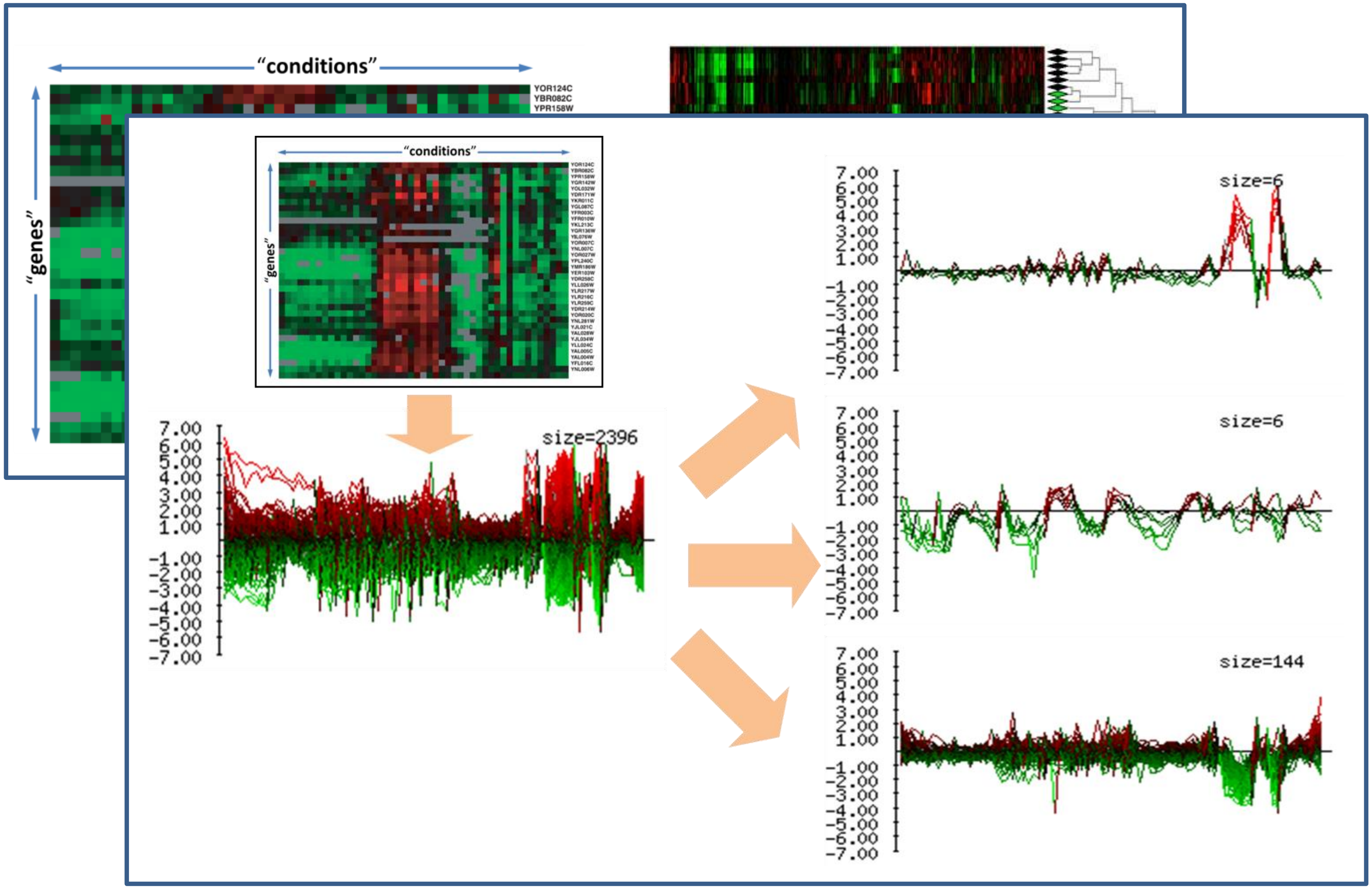
Different views of clustering …



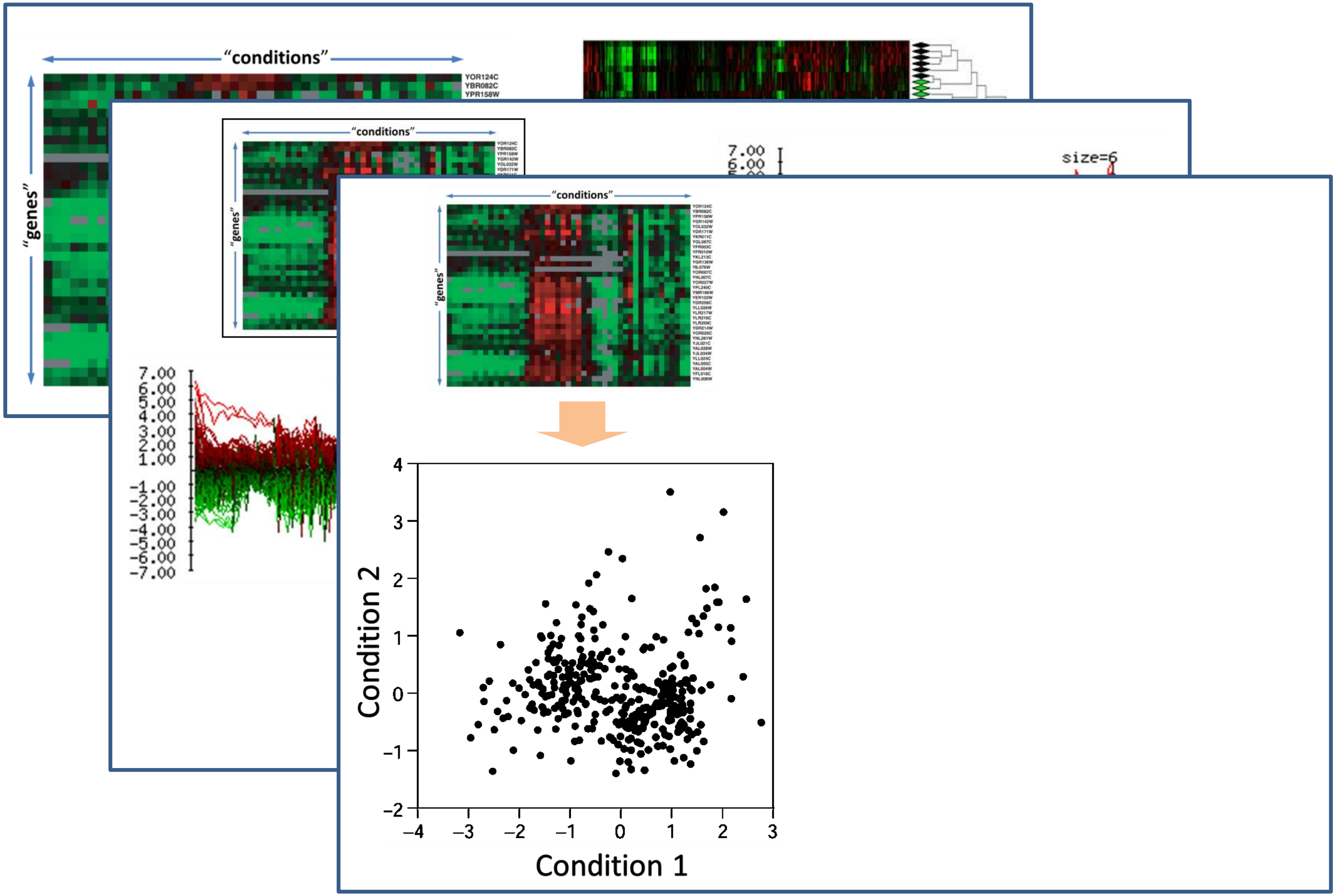




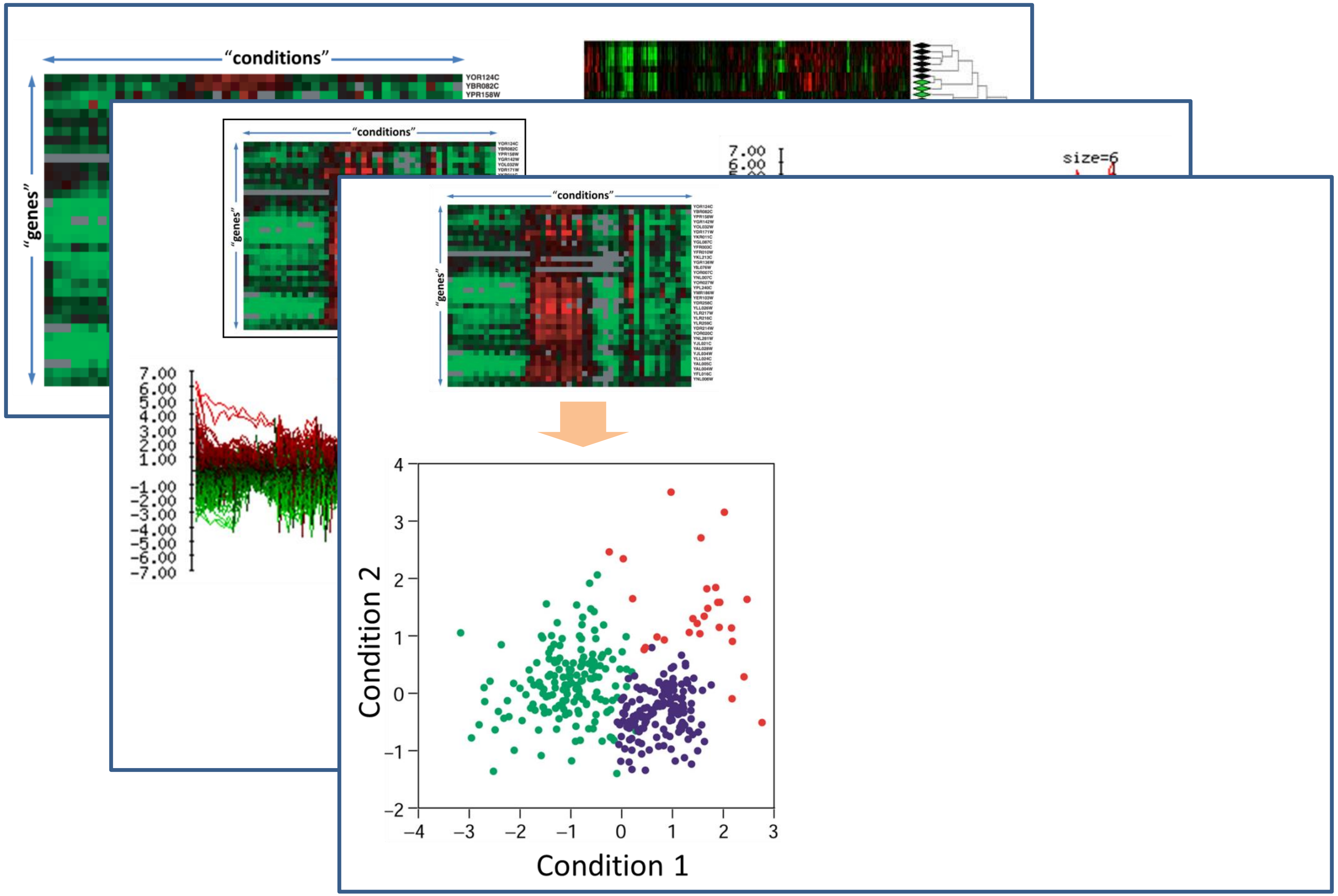
Different views of clustering …



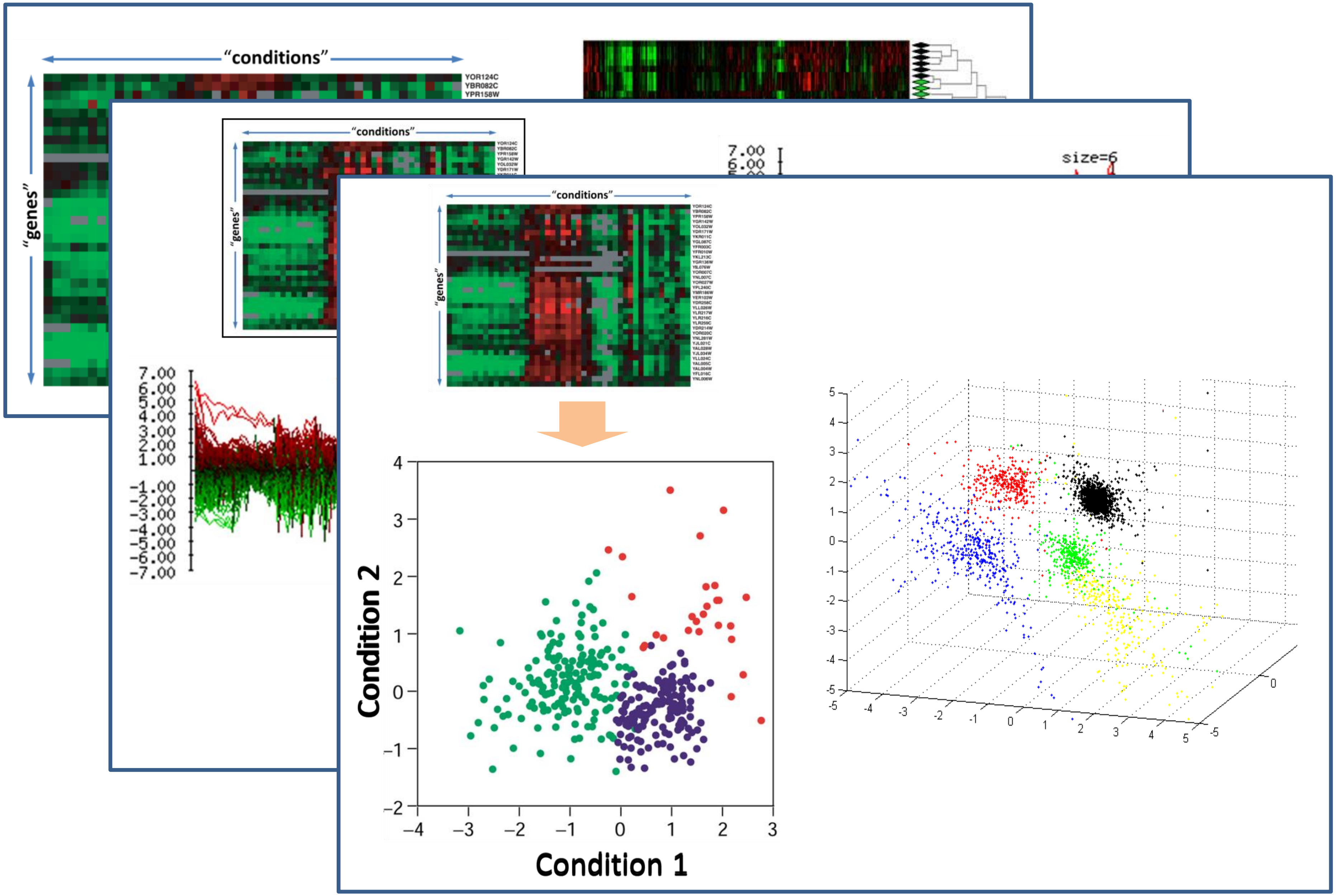
Different views of clustering …



Different views of clustering …

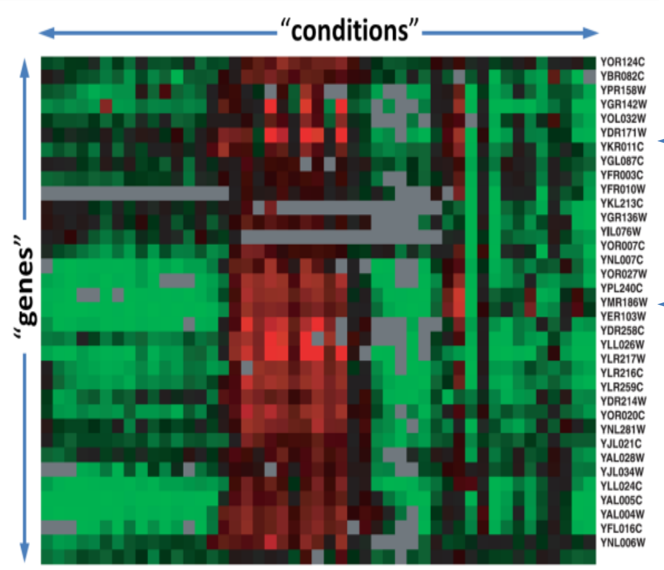


Different views of clustering …



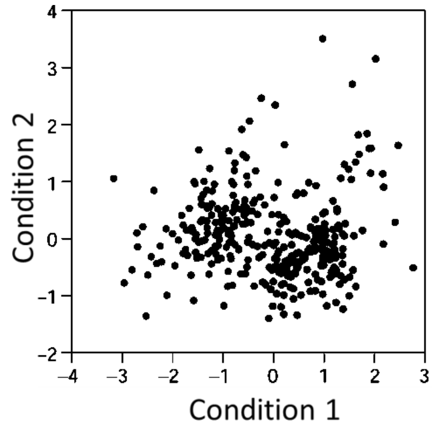
Measuring similarity/distance

- An important step in many clustering methods is the selection of a distance measure (**metric**), defining the distance between 2 data points (e.g., 2 genes)



"Point" 1 : [0.1 0.0 0.6 1.0 2.1 0.4 0.2]

"Point" 2 : [0.2 1.0 0.8 0.4 1.4 0.5 0.3]

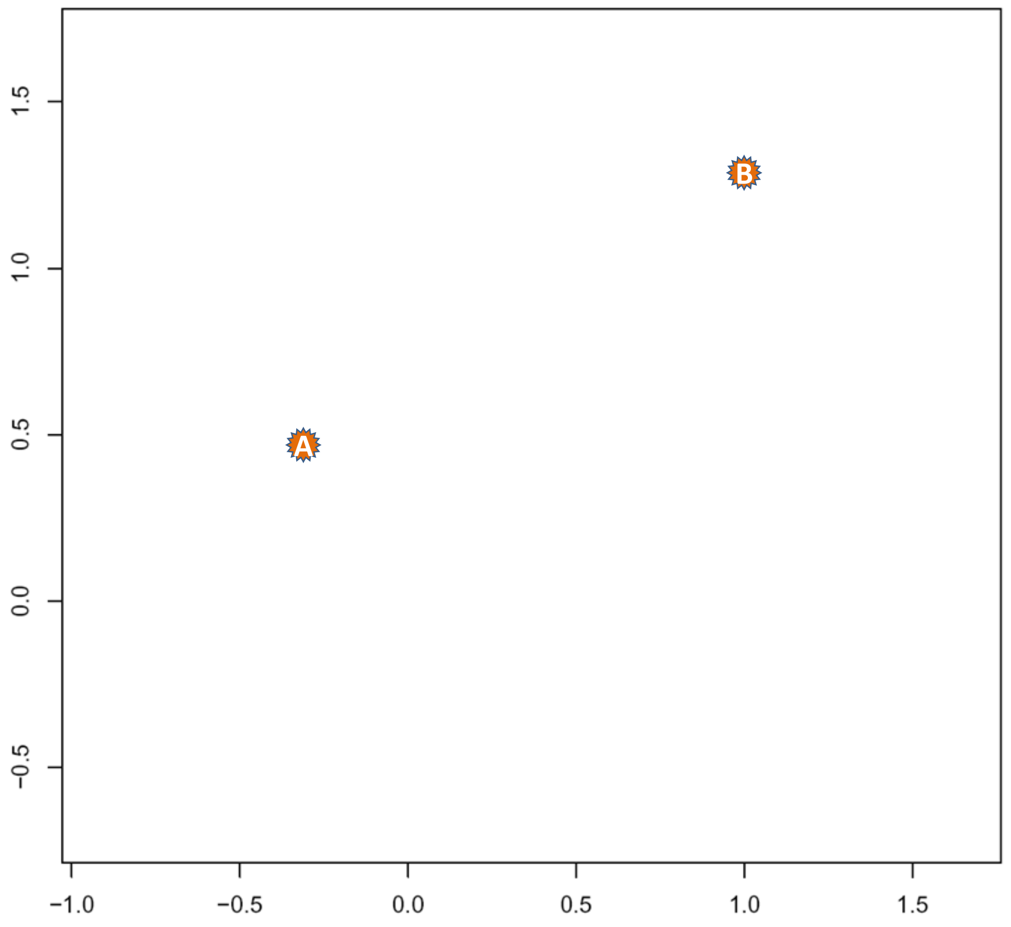


Genes are points in the multi-dimensional space R^n

(where n denotes the number of conditions)

Measuring similarity/distance

- So … how do we measure the distance between two point in a multi-dimensional space?



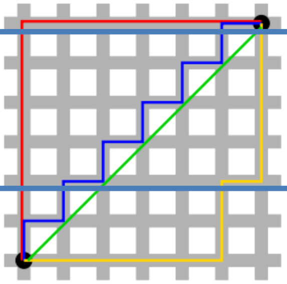
Measuring similarity/distance

- So … how do we measure the distance between two point in a multi-dimensional space?

p-norm

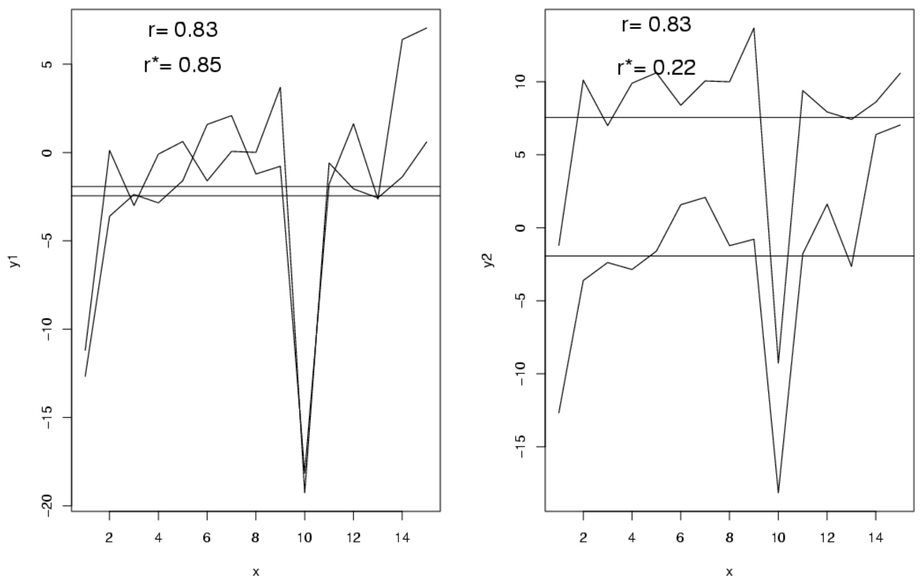
$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

- Common distance functions:
  - The **Euclidean** distance $\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}.$ ← ***2-norm***
    (a.k.a "distance as the crow flies" or distance).
  - The **Manhattan** distance ← ***1-norm***
    (a.k.a **taxicab distance**)
  - The **maximum** norm ← ***infinity-norm***
    (a.k.a **infinity distance**)
  - The **Hamming** distance
    (number of substitutions required to change one point into another).



Correlation as distance

- Another approach is to use the correlation between two data points as a distance metric.
  - Pearson Correlation
  - Spearman Correlation
  - Absolute Value of Correlation



Metric matters!

- The metric of choice has a marked impact on the shape of the resulting clusters:
  - Some elements may be close to one another in one metric and far from one anther in a different metric.

- Consider, for example, the point (x=1,y=1) and the origin.
  - What’s their distance using the 2-norm (Euclidean distance)?
  - What’s their distance using the 1-norm (a.k.a. taxicab/ Manhattan norm)?
  - What’s their distance using the infinity-norm?

The clustering problem

- A good clustering solution should have two features:

1. **High homogeneity**: homogeneity measures the similarity between genes assigned to the same cluster.

2. **High separation**: separation measures the distance/dis-similarity between clusters.
   (If two clusters have similar expression patterns, then they should probably be merged into one cluster).

The “philosophy” of clustering

- “**Unsupervised learning**” problem
- **No single solution is necessarily the true/correct!**
- There is usually a **tradeoff** between homogeneity and separation:
  - More clusters → increased homogeneity but decreased separation
  - Less clusters → Increased separation but reduced homogeneity
- Method matters; metric matters; definitions matter;
- There are many formulations of the clustering problem; most of them are **NP-hard (why?)**.
- In most cases, **heuristic methods** or approximations are used.

One problem, numerous solutions

- Many algorithms:
  - Hierarchical clustering
  - k-means
  - self-organizing maps (SOM)
  - Knn
  - PCC
  - CAST
  - CLICK
- The results (i.e., obtained clusters) can vary drastically depending on:
  - Clustering method
  - Parameters specific to each clustering method (e.g. number of centers for the k-mean method, agglomeration rule for hierarchical clustering, etc.)

Hierarchical clustering

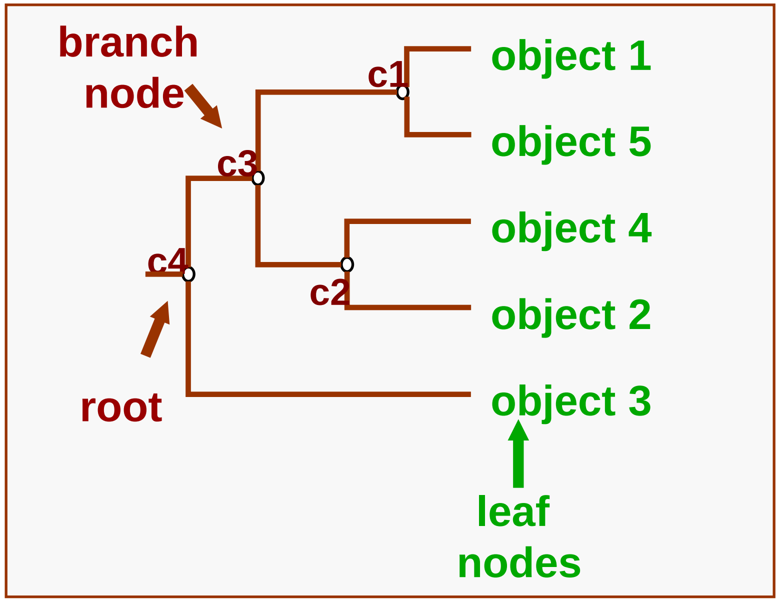
Hierarchical clustering

- An **agglomerative** clustering method
  - Takes as input a distance matrix
  - Progressively regroups the closest objects/groups
  - The result is a tree - intermediate nodes represent clusters
  - Branch lengths represent distances between clusters

Distance matrix

	object 1	object 2	object 3	object 4	object 5
object 1	0.00	4.00	6.00	3.50	1.00
object 2	4.00	0.00	6.00	2.00	4.50
object 3	6.00	6.00	0.00	5.50	6.50
object 4	3.50	2.00	5.50	0.00	4.00
object 5	1.00	4.50	6.50	4.00	0.00

Tree representation



mmm…
Déjà vu anyone?

Hierarchical clustering algorithm

1. Assign each object to a separate cluster.
2. Find the pair of clusters with the shortest distance, and regroup them into a single cluster.
3. Repeat 2 until there is a single cluster.

Hierarchical clustering algorithm

1. Assign each object to a separate cluster.
2. **Find the pair of clusters with the shortest distance, and regroup them into a single cluster.**
3. Repeat 2 until there is a single cluster.

Hierarchical clustering

1. Assign each object to a separate cluster.
2. **Find the pair of clusters with the shortest distance, and regroup them into a single cluster.**
3. Repeat 2 until there is a single cluster.

- One needs to define a (dis)similarity metric between two **groups**. There are several possibilities
  - **Average linkage:** the average distance between objects from groups A and B
  - **Single linkage:** the distance between the closest objects from groups A and B
  - **Complete linkage:** the distance between the most distant objects from groups A and B

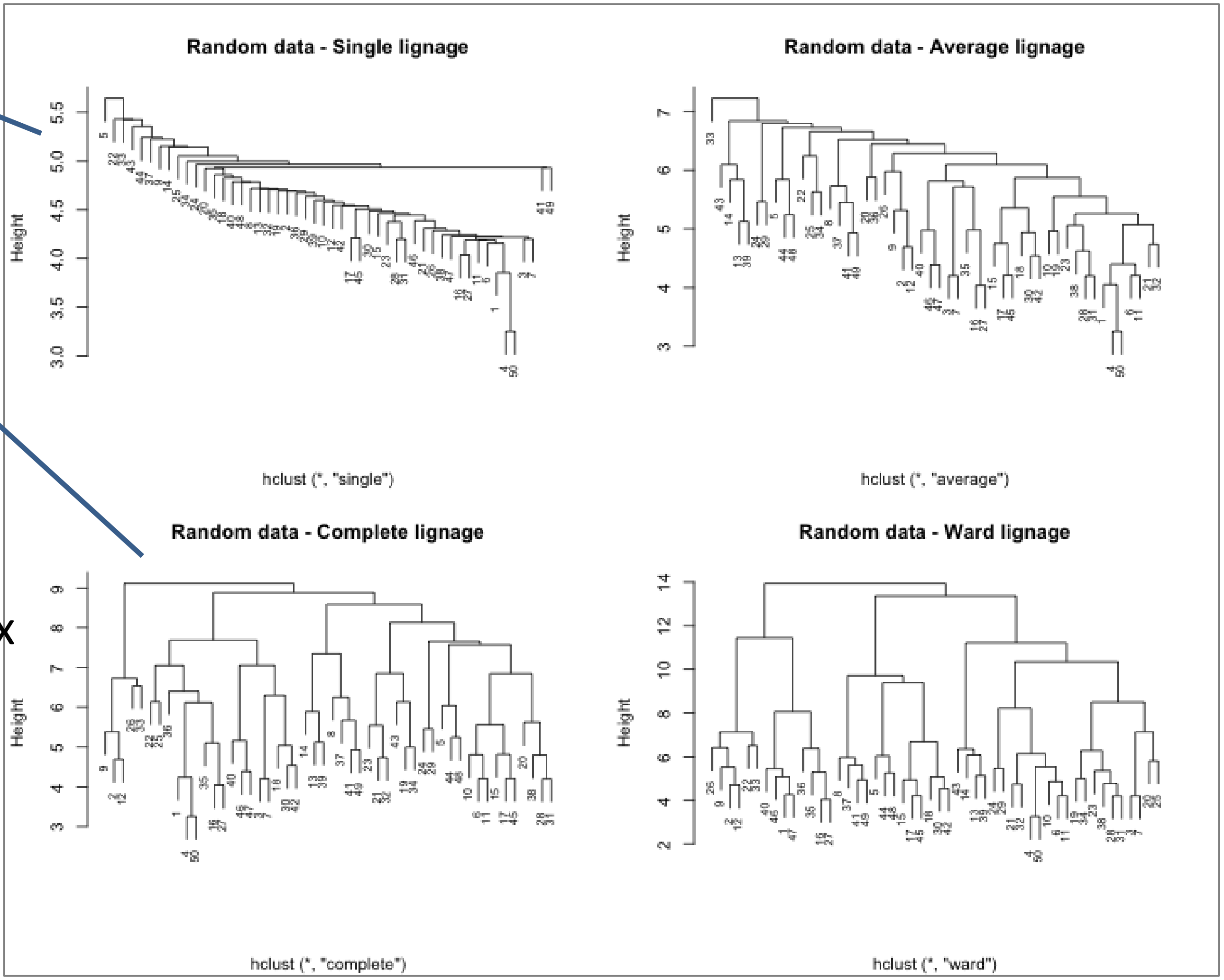
Impact of the agglomeration rule

- These four trees were built from the same distance matrix, using 4 different agglomeration rules.

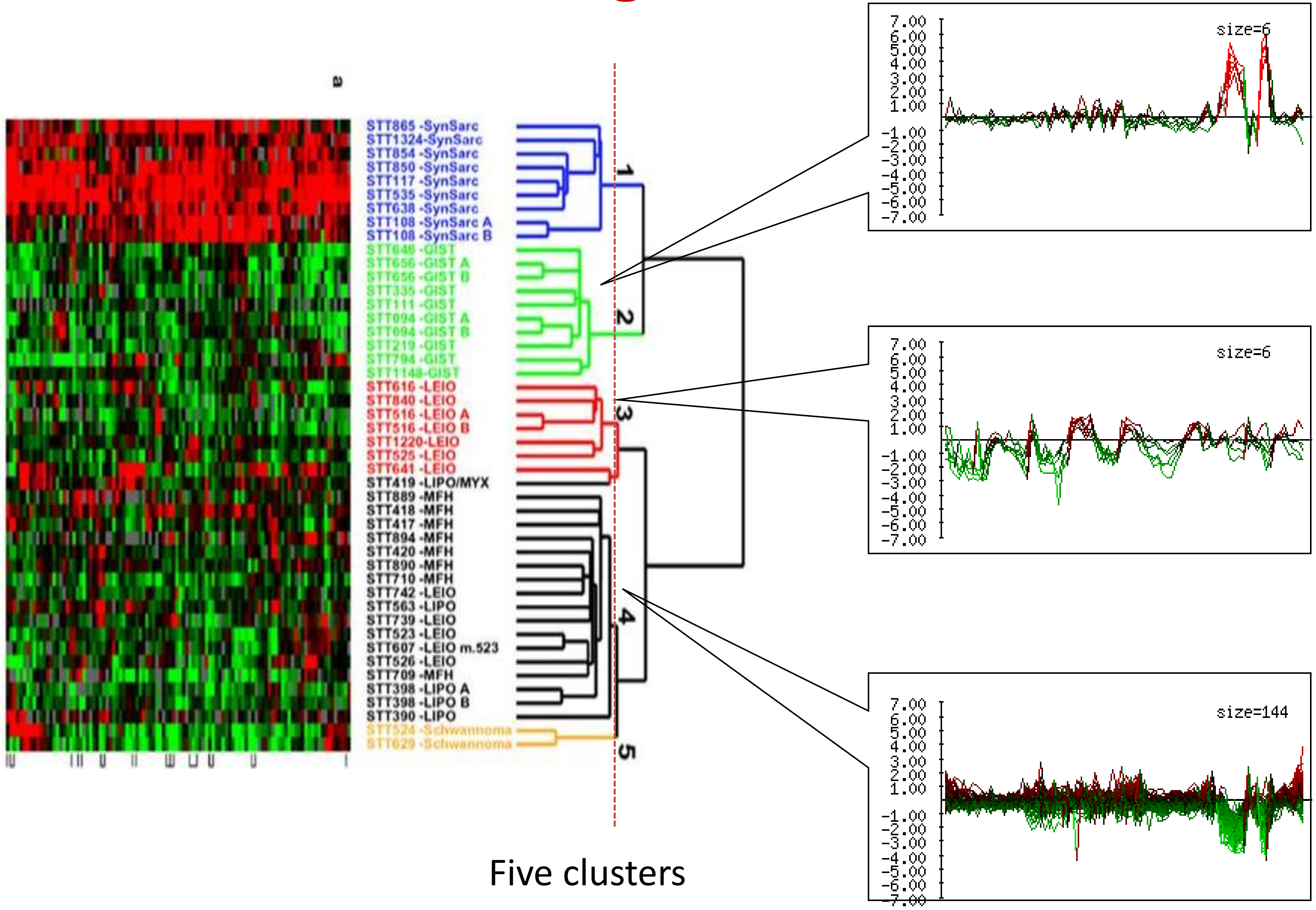
Single-linkage typically creates nesting clusters

Complete linkage create more balanced trees.

Note: these trees were computed from a matrix of random numbers. The impression of structure is thus a complete artifact.



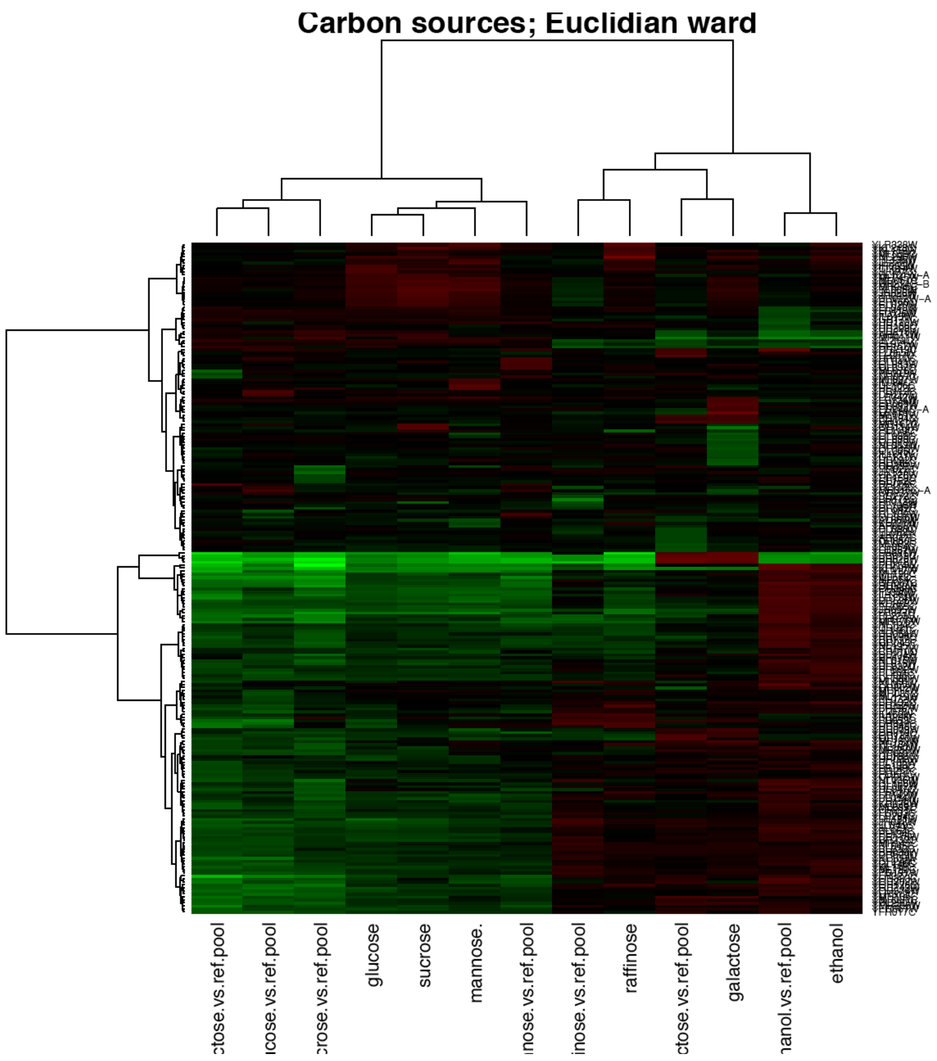
Hierarchical clustering result



Five clusters

Clustering in both dimensions

- We can cluster genes, conditions (samples), or both.


Carbon sources; Euclidian ward
ctose.vs.ref.pool
icose.vs.ref.pool
crose.vs.ref.pool
glucose
sucrose
mannose.
nose.vs.ref.pool
inose.vs.ref.pool
raffinose
ctose.vs.ref.pool
galactose
nanol.vs.ref.pool
ethanol