Clustering, cont’

Genome 373

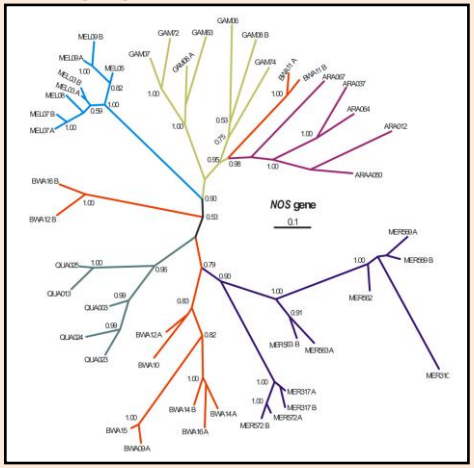
Genomic Informatics

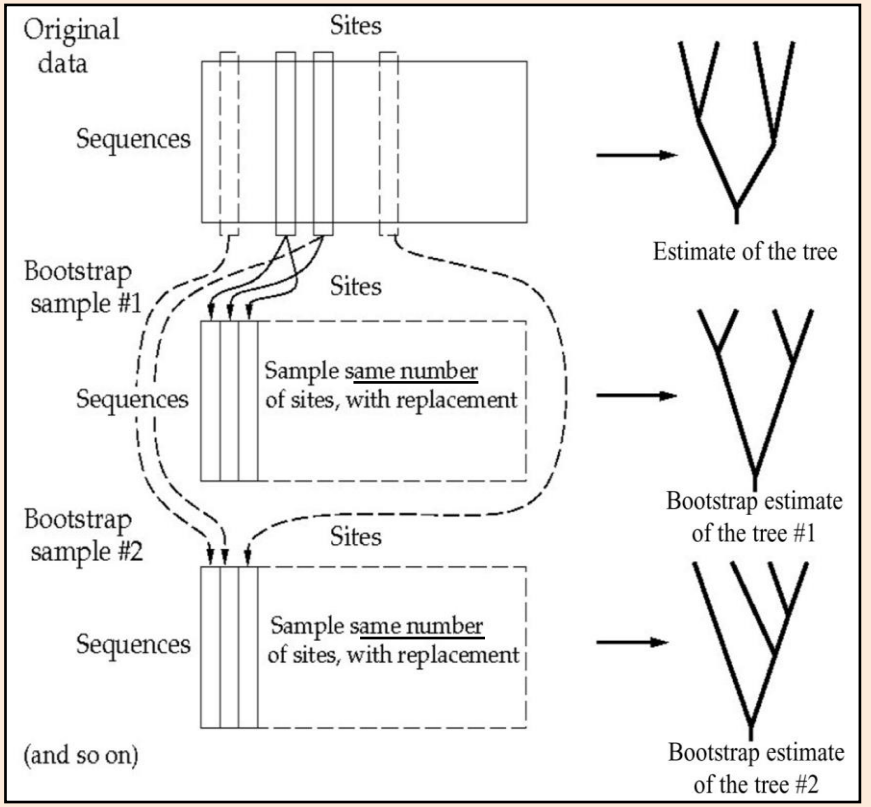
Elhanan Borenstein

Some slides adapted from Jacques van Helden

A quick review

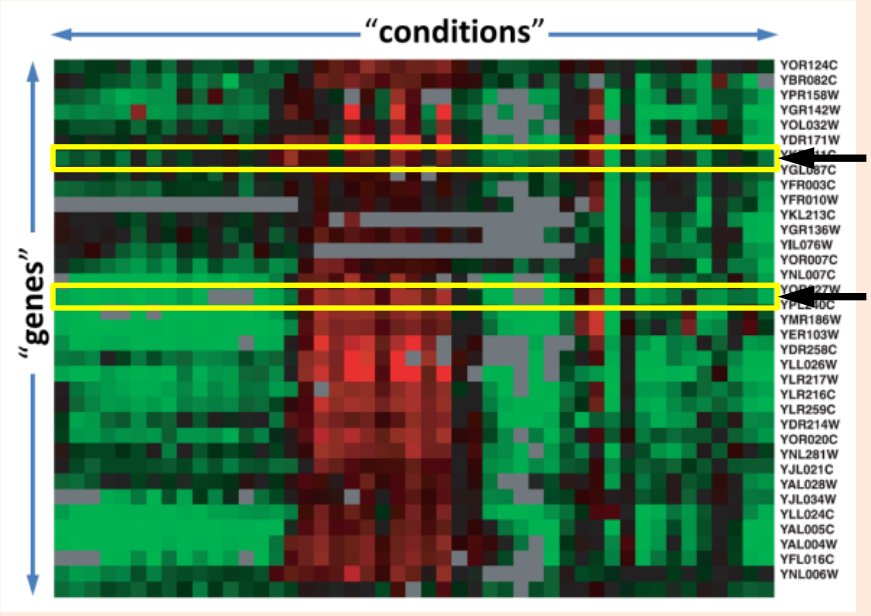
- **Improving the search heuristic:**
  - Multiple starting points
  - Simulated annealing
  - Genetic algorithms
- **Branch confidence and bootstrap support**





A quick review

- **Clustering:**
  - The clustering problem: **homogeneity vs. separation**
  - Why clustering
  - The number of possible clustering solutions



gene x

[0.1, 0.0, 0.6, 1.0, 2.1, 0.4, 0.2, 0.3, 0.5, 0.1, 2.1]

gene y

[0.2, 1.0, 0.8, 0.4, 1.4, 0.5, 0.3, 2.1, 1.2, 3.4, 0.1]

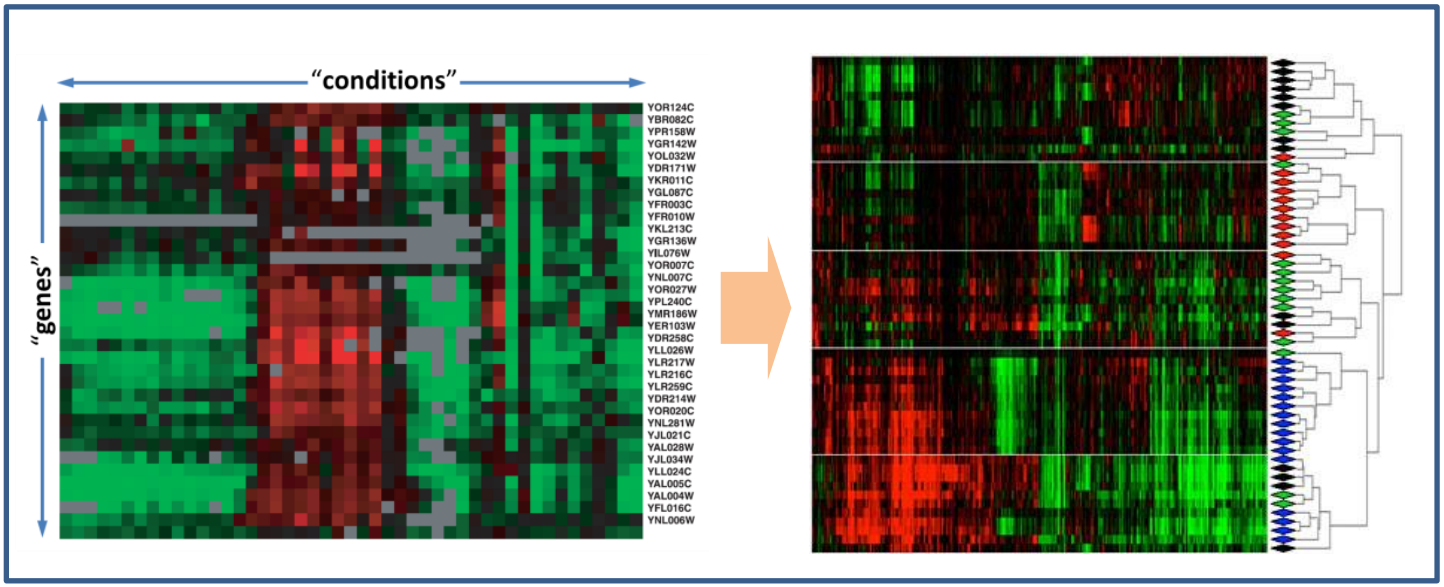
One problem, numerous solutions

- Many algorithms:
  - Hierarchical clustering
  - k-means
  - self-organizing maps (SOM)
  - Knn
  - PCC
  - CLICK

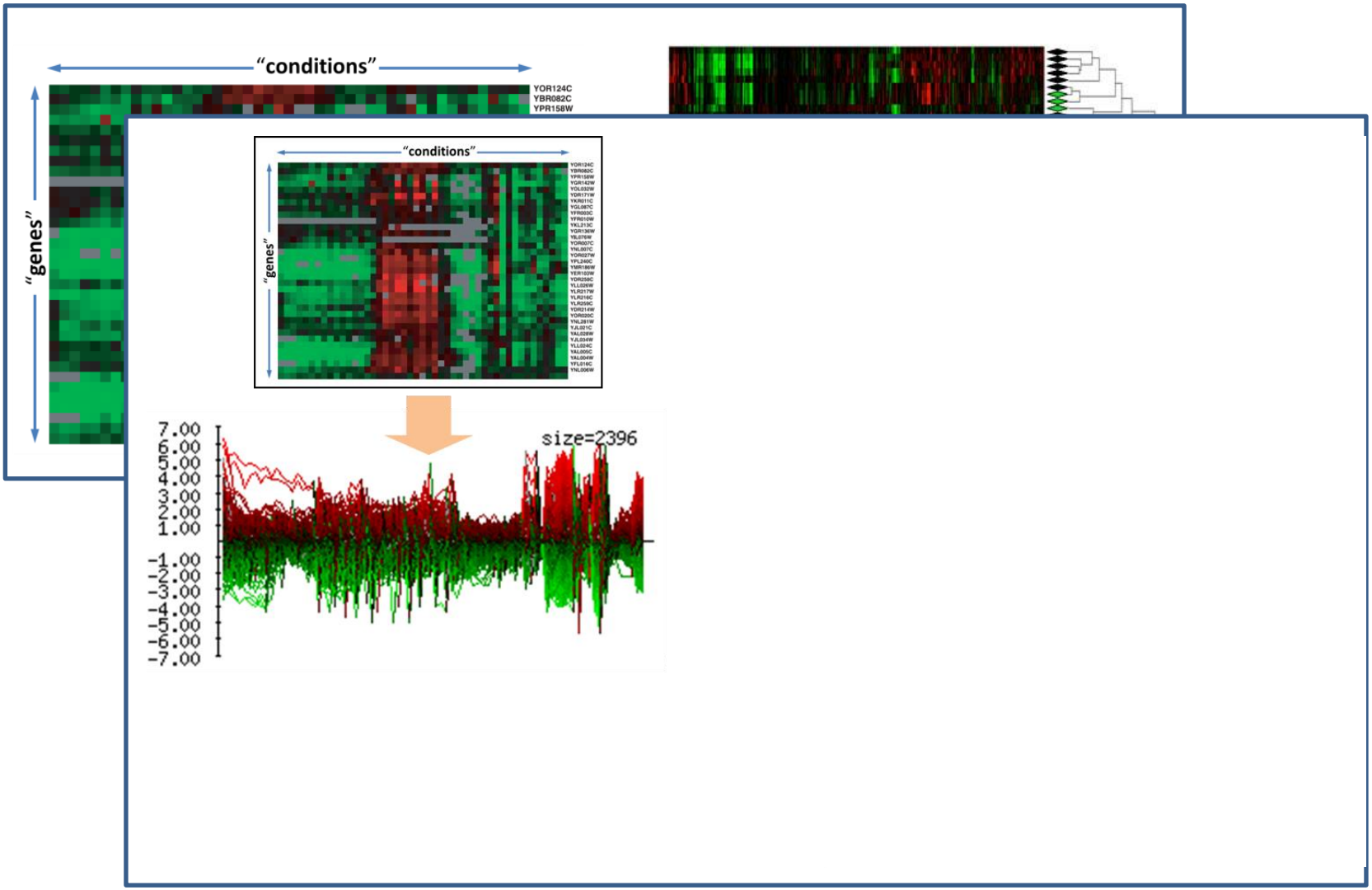
- There are many formulations of the clustering problem; most of them are **NP-hard**

- The results (i.e., obtained clusters) can vary drastically depending on:
  - Clustering method
  - Parameters specific to each clustering method

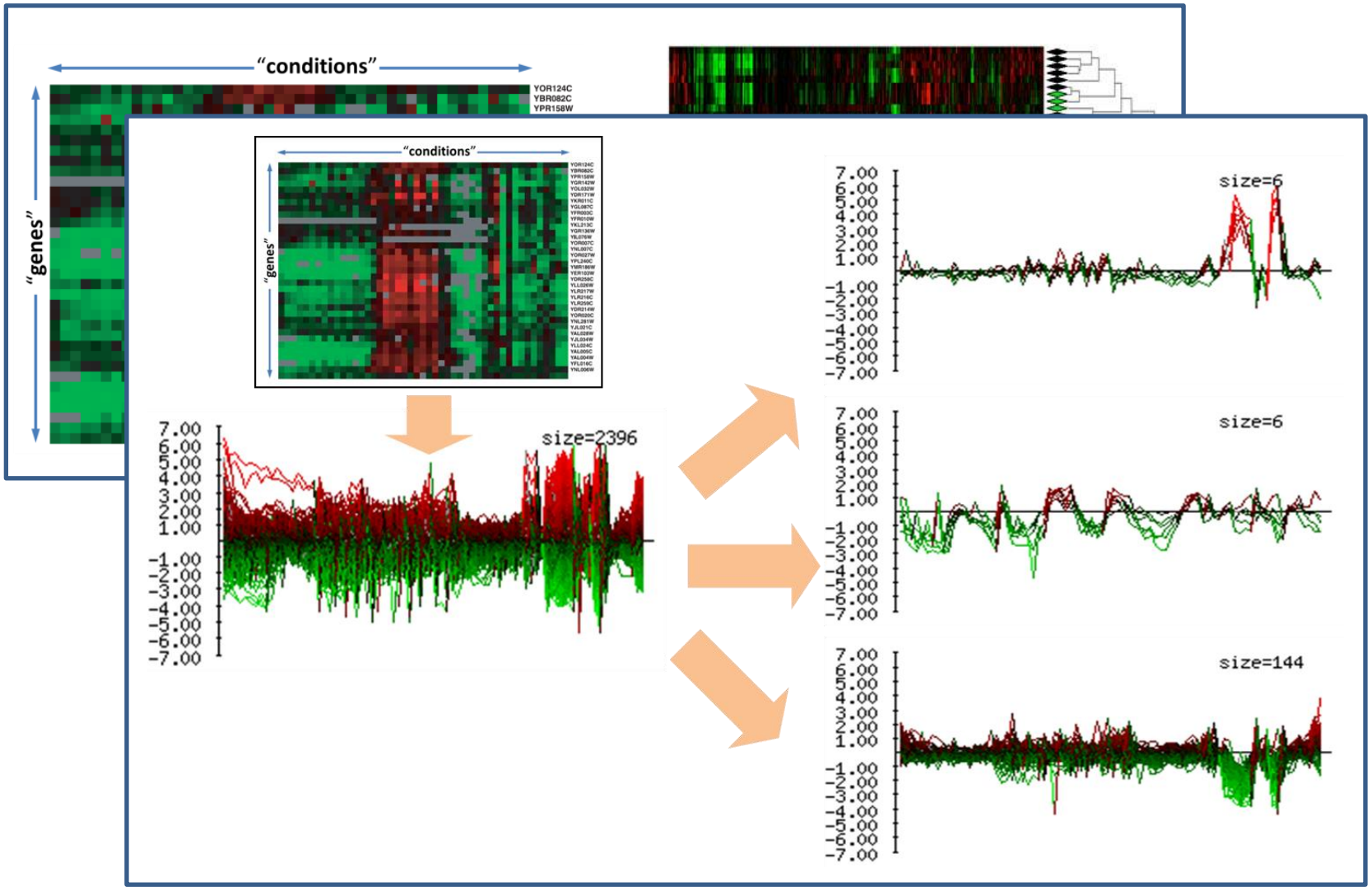
Different views of clustering …



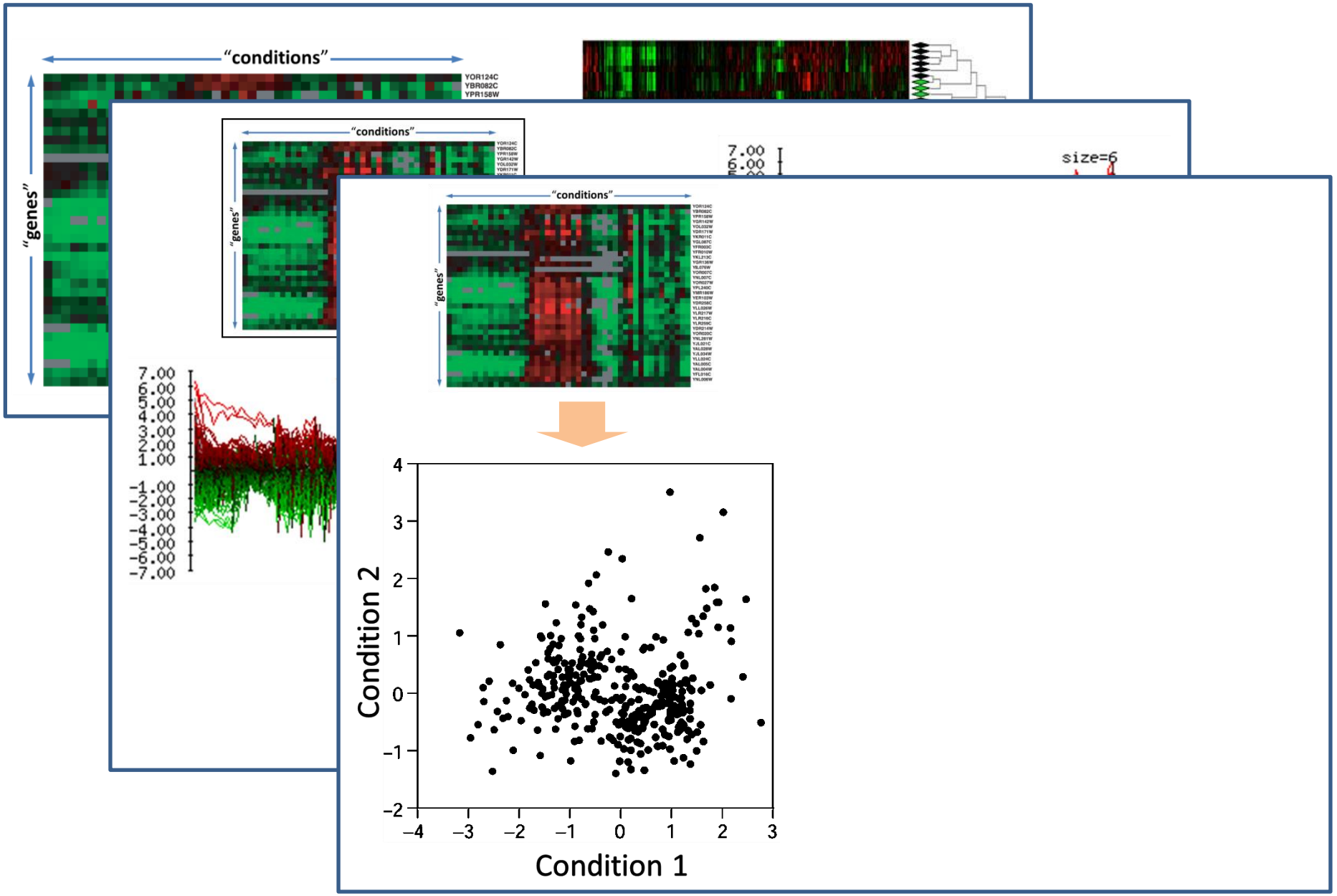
Different views of clustering …



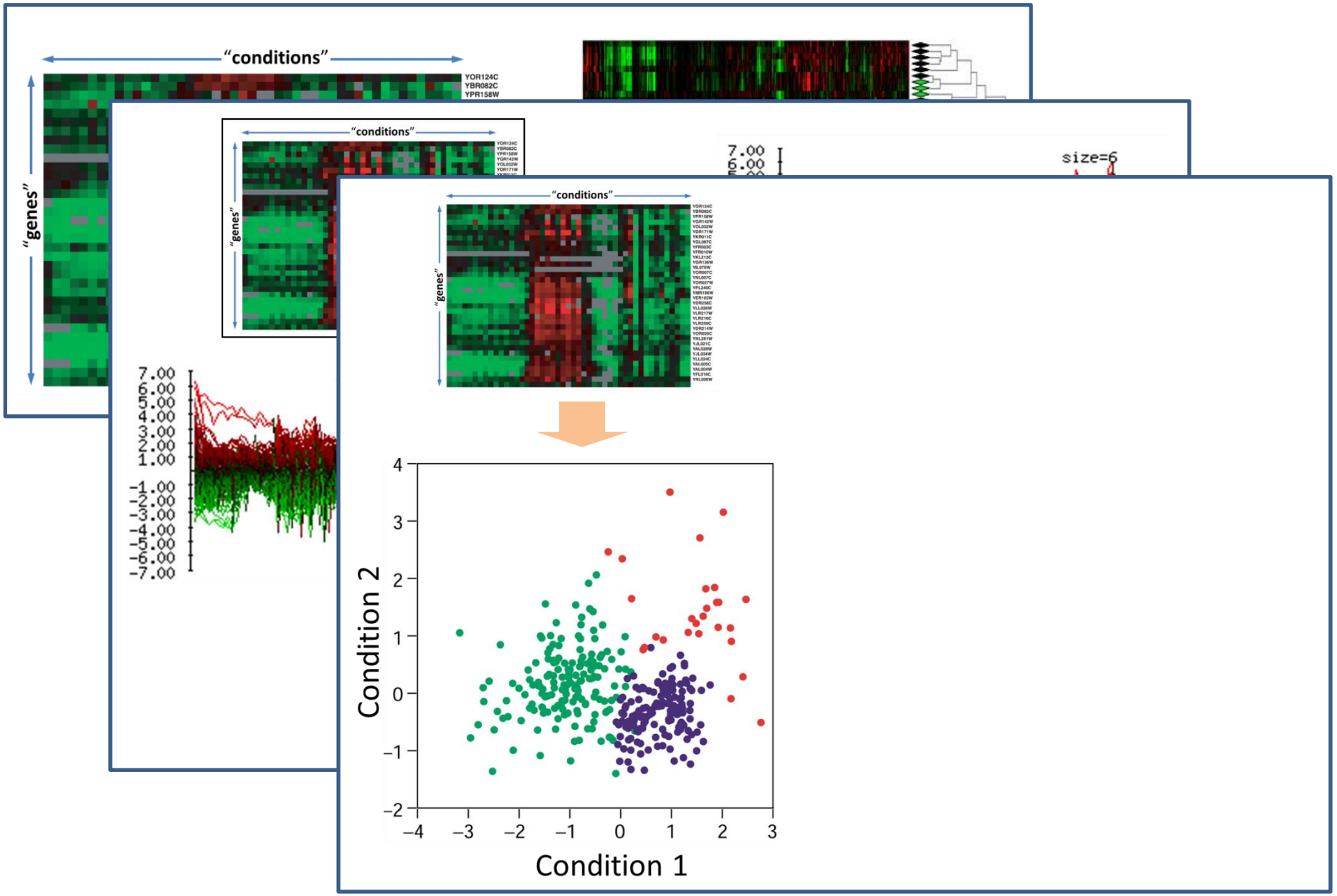
Different views of clustering …



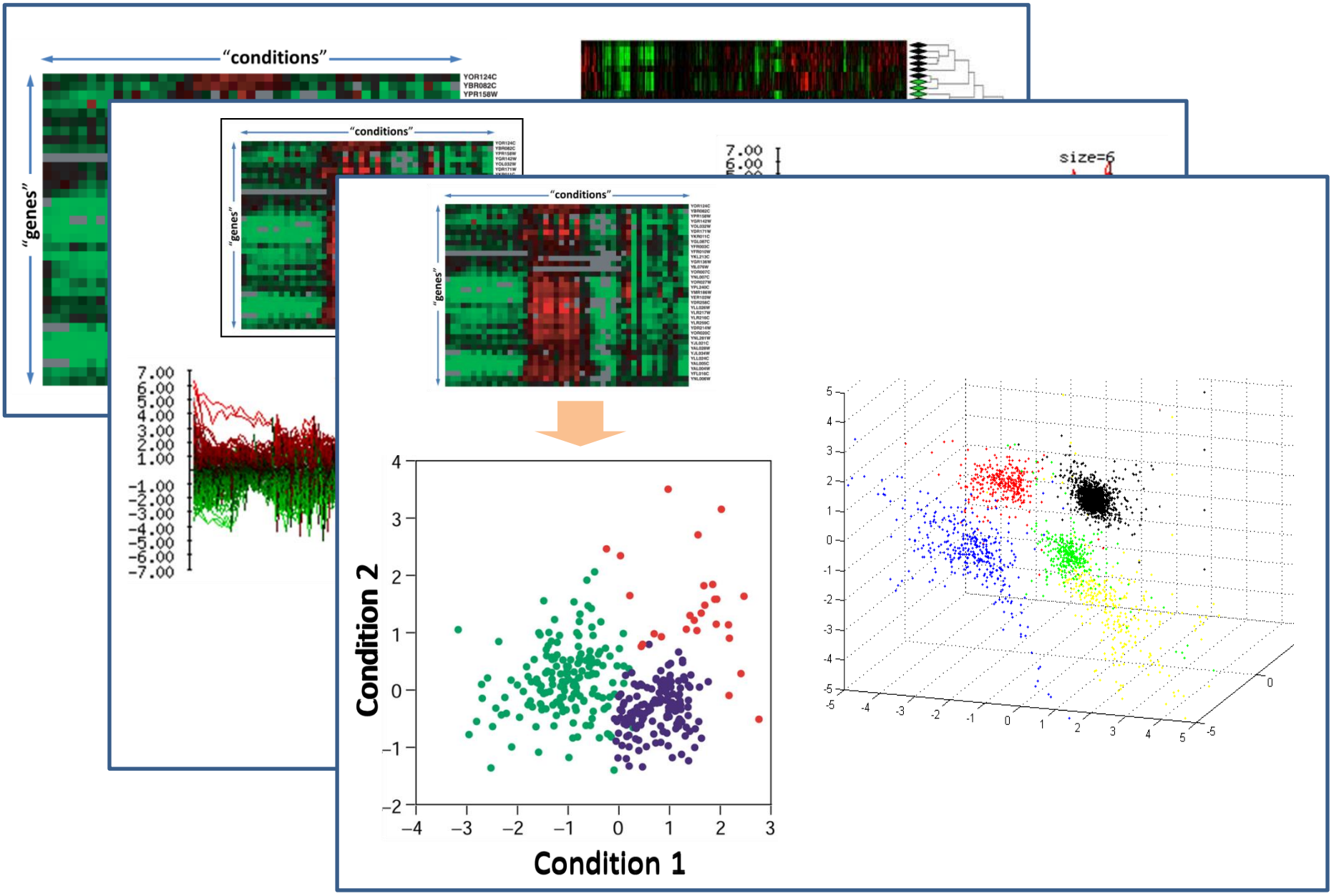
Different views of clustering …



Different views of clustering …

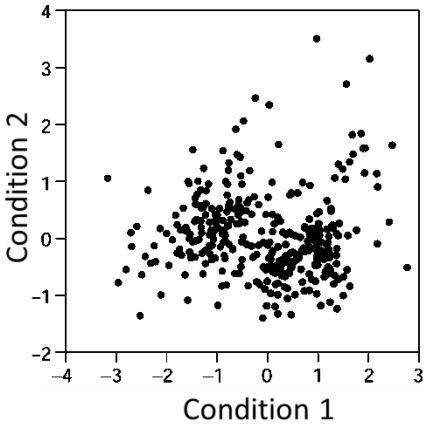


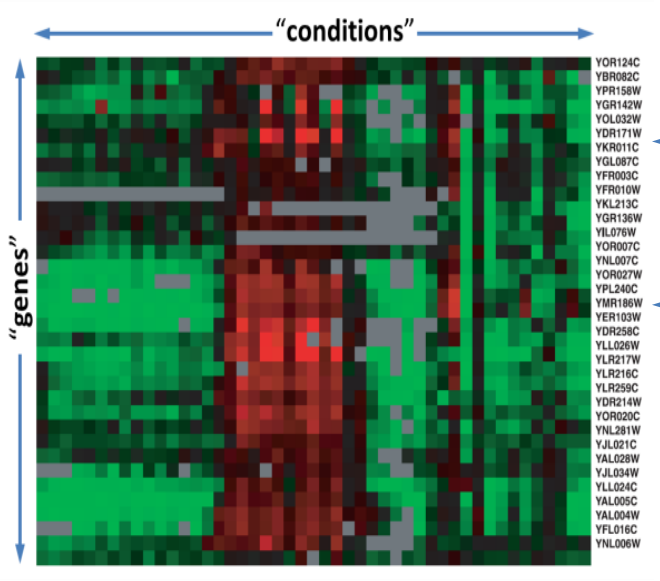
Different views of clustering …



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)





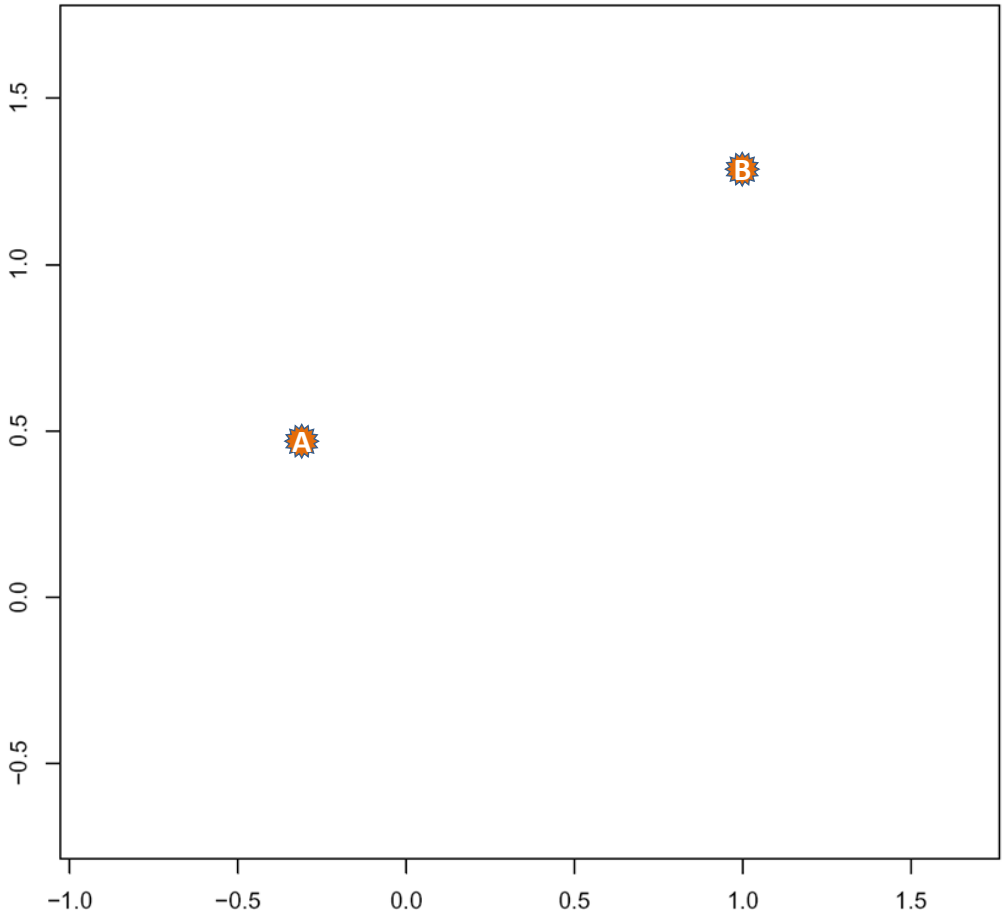
"Point" 1 : [0.1 0.0 0.6 1.0 2.1 0.4 0.2]

"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

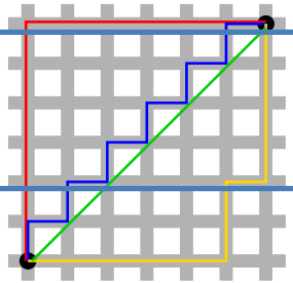
$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- 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**)



- **Correlation**

  (Pearson, Spearman, Absolute Value of Correlation, etc.)

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?

Hierarchical clustering

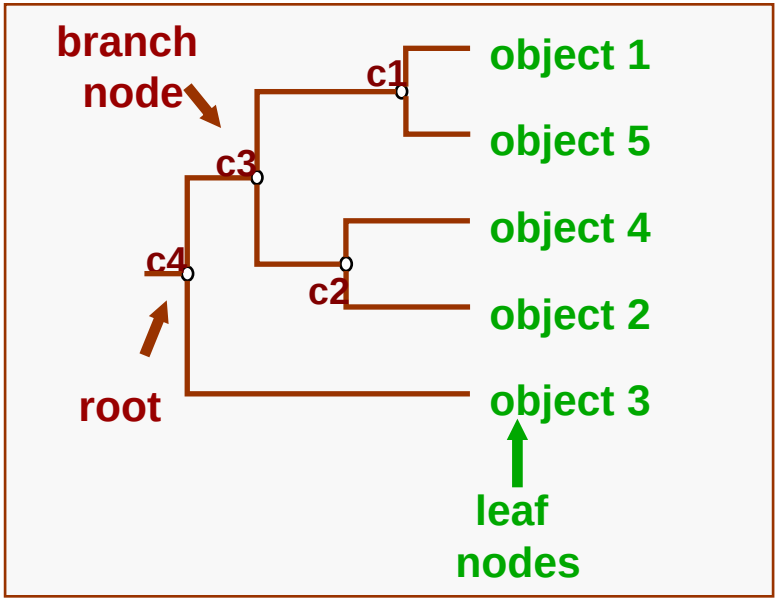
Hierarchical clustering

- **Hierarchical** clustering is an **agglomerative** clustering method
  - Takes as input a distance matrix
  - Progressively regroups the closest objects/groups

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.

- The result is a tree, whose intermediate nodes represent clusters

- Branch lengths represent distances between clusters

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.

Linkage (agglomeration) methods

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.

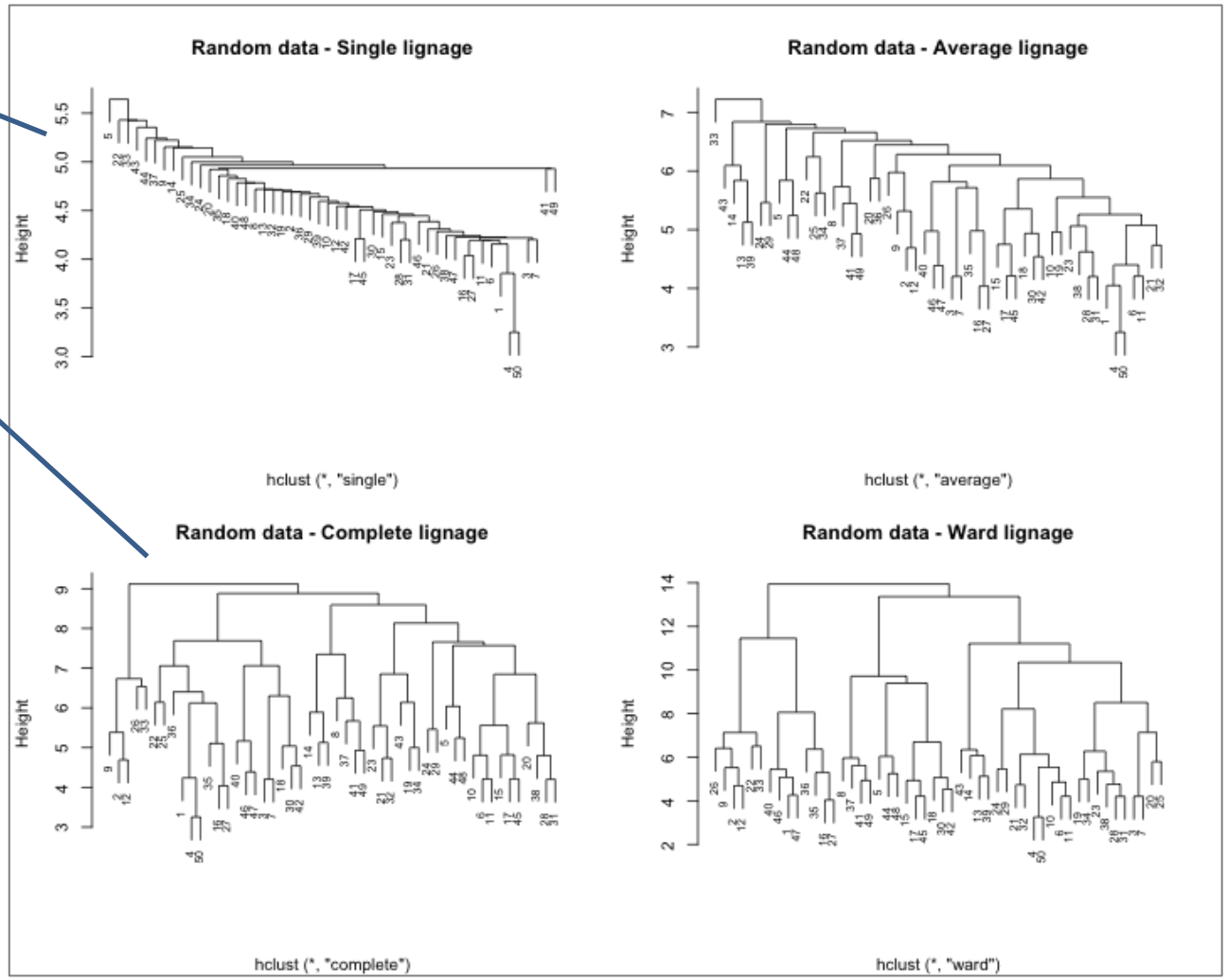
- One needs to define a (dis)similarity metric between two **groups**. There are several possibilities
  - **Average linkage:** the average distance between objects from groups A and B
  - **Single linkage:** the distance between the closest objects from groups A and B
  - **Complete linkage:** the distance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

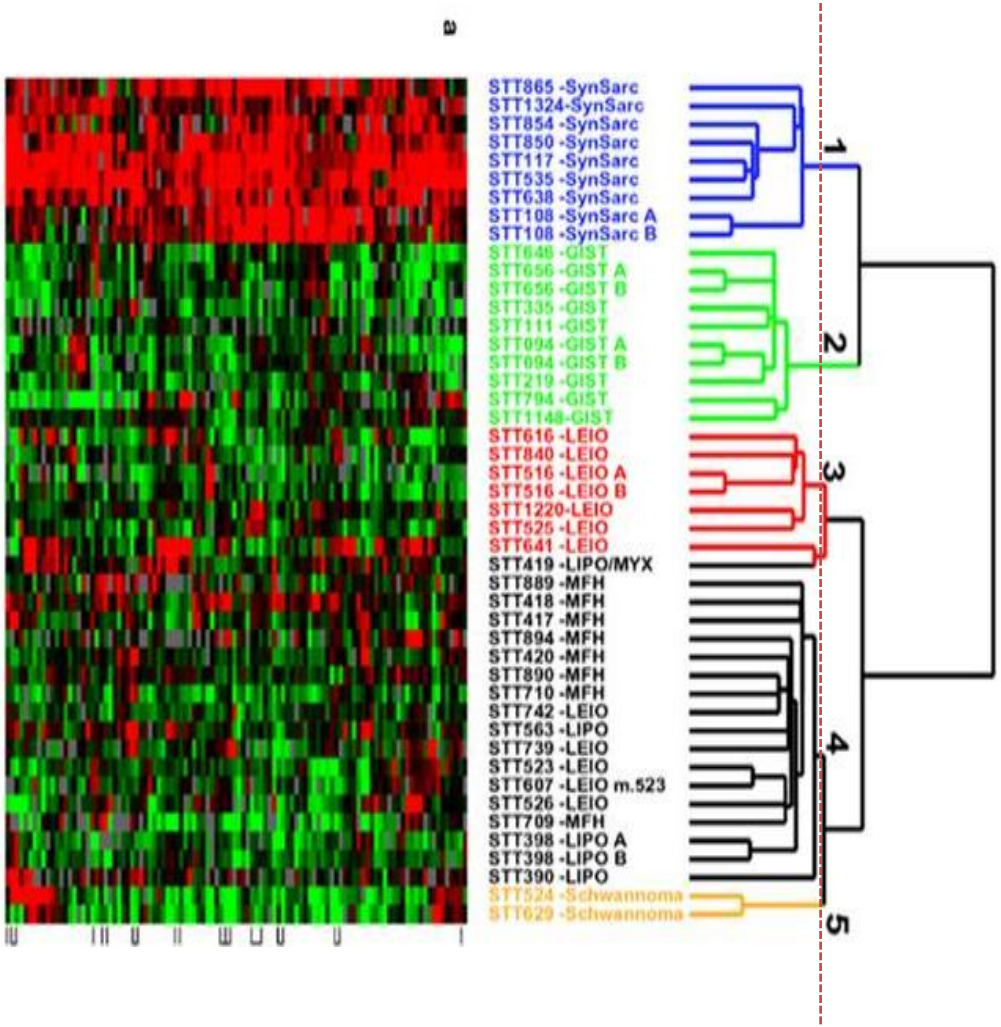
Single-linkage typically creates nesting clusters

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

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;
- In most cases, **heuristic methods** or approximations are used.