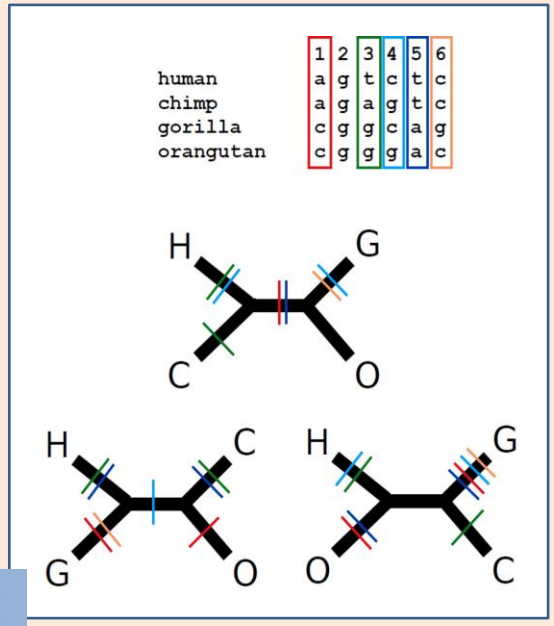
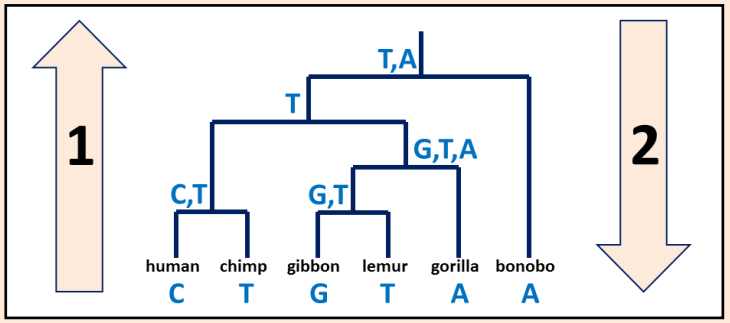
A quick review

- **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 — **Too many!**
  2. For each site in the alignment and for each tree count the minimal number of changes required — **The small parsimony problem**
  3. Add sites to obtain the total number of changes required for each tree
  4. Pick the tree with the lowest score

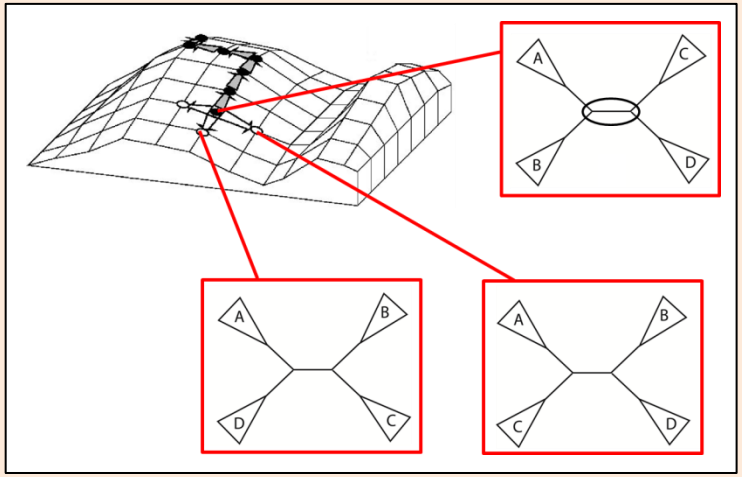


A quick review – cont'

- **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



The parsimony algorithm

1) *Construct all possible trees* ***or search the space of possible trees using NNI hill-climb***
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* ***or search until no better tree can be found***

Search algorithm - Review

How can we improve the search algorithm and increase our chances of finding the optimal tree?

How can we apply this algorithm to solve other problems?

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:

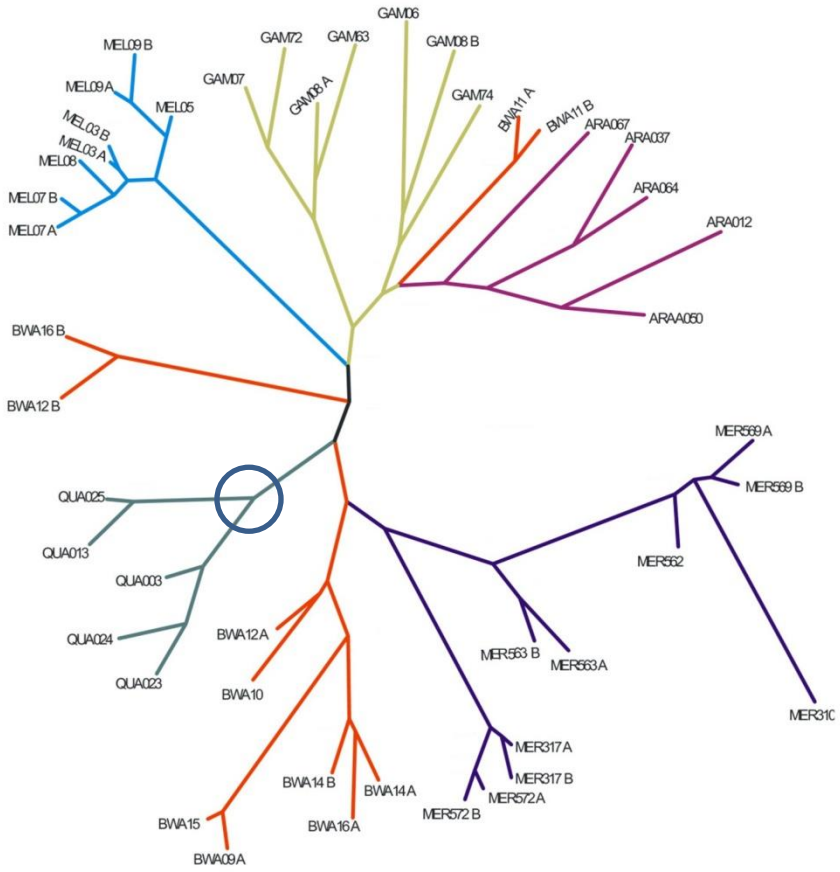
1) *Tree evaluated for likelihood of data given tree.*
2) *Uses a specific model for evolutionary rates (such as Jukes-Cantor).*
3) *Like parsimony, must search tree space.*
4) *Usually most accurate method but slow.*

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:

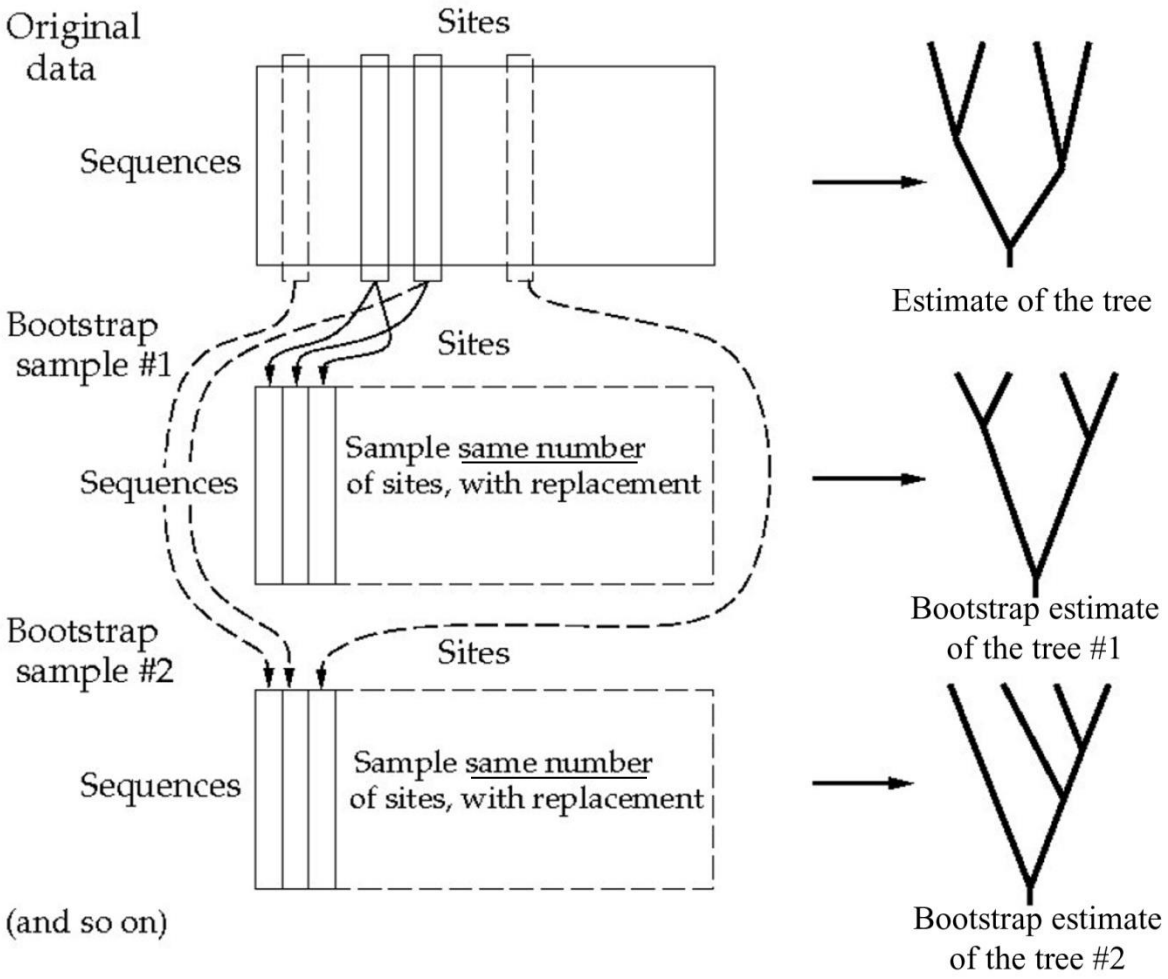


subtree1: QUA025, QUA013
Subtree2: QUA003, QUA024, QUA023
Subtree3: everything else

Bootstrap support

Most commonly used branch support test:

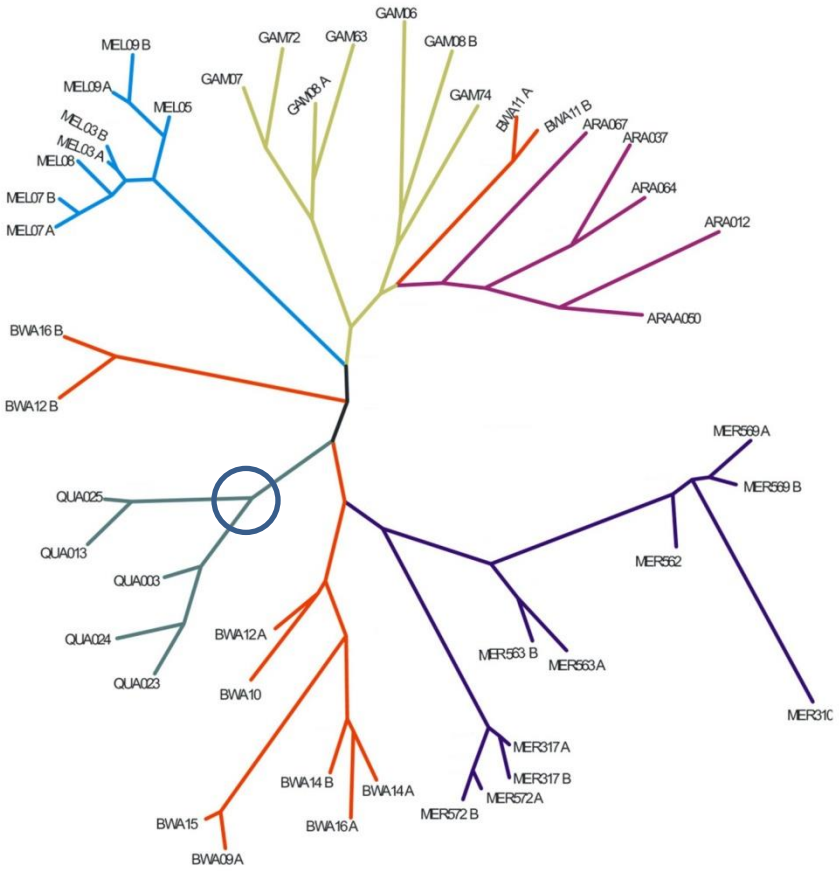
1. *Randomly sample alignment sites (with replacement).*
2. *Use sample to estimate the tree.*
3. *Repeat many times.*



(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).



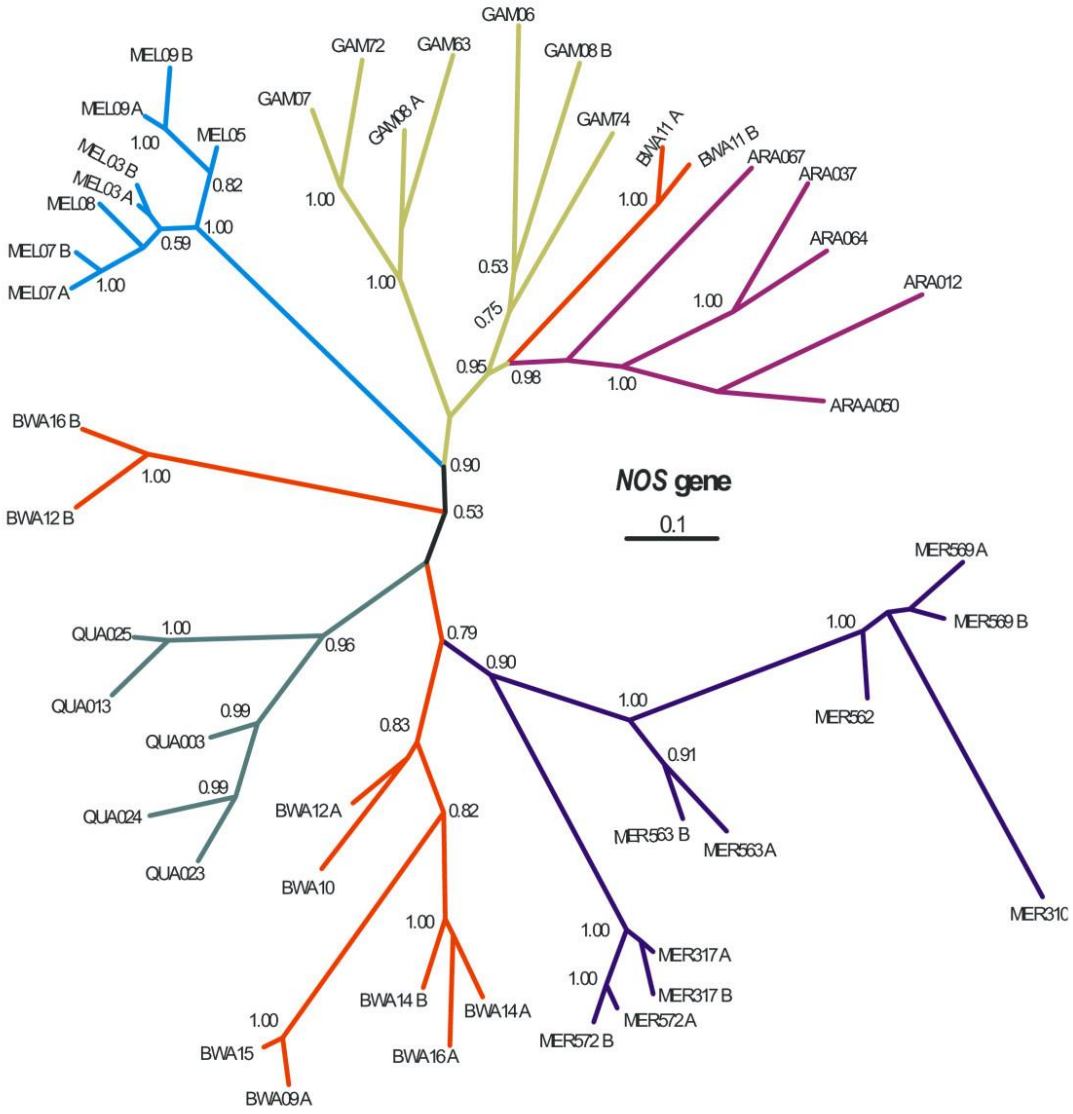
For example at the circled branch point, what fraction of the bootstrap trees have a branch point where the three subtrees include:

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)



Clustering

Genome 373

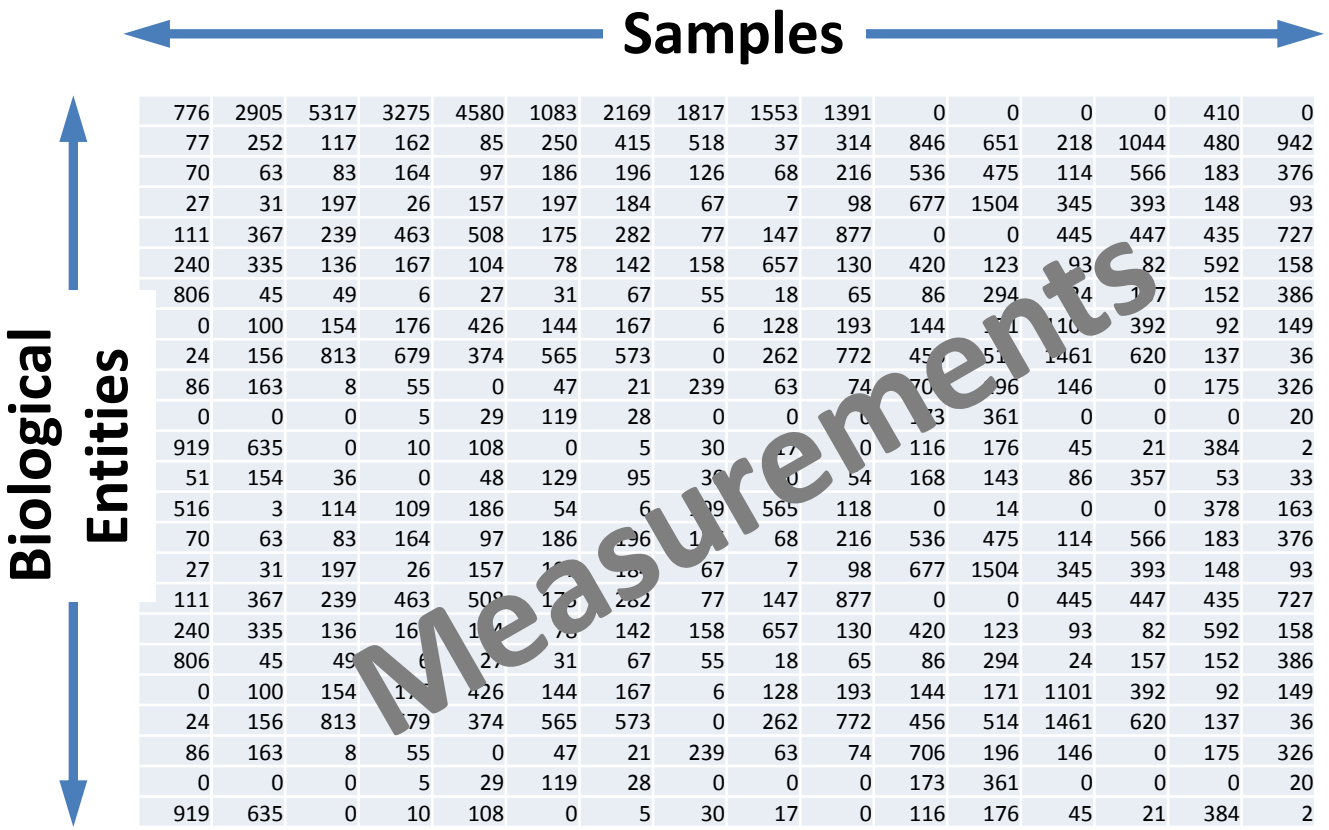
Genomic Informatics

Elhanan Borenstein

Some slides adapted from Jacques van Helden

A common data structure in high-throughput biology

A common data structure in high-throughput biology



776	2905	5317	3275	4580	1083	2169	1817	1553	1391	0	0	0	0	410	0
77	252	117	162	85	250	415	518	37	314	846	651	218	1044	480	942
70	63	83	164	97	186	196	126	68	216	536	475	114	566	183	376
27	31	197	26	157	197	184	67	7	98	677	1504	345	393	148	93
111	367	239	463	508	175	282	77	147	877	0	0	445	447	435	727
240	335	136	167	104	78	142	158	657	130	420	123	93	82	592	158
806	45	49	6	27	31	67	55	18	65	86	294	24	157	152	386
0	100	154	176	426	144	167	6	128	193	144	171	1101	392	92	149
24	156	813	679	374	565	573	0	262	772	456	514	1461	620	137	36
86	163	8	55	0	47	21	239	63	74	706	196	146	0	175	326
0	0	0	5	29	119	28	0	0	0	173	361	0	0	0	20
919	635	0	10	108	0	5	30	17	0	116	176	45	21	384	2
51	154	36	0	48	129	95	39	0	54	168	143	86	357	53	33
516	3	114	109	186	54	6	99	565	118	0	14	0	0	378	163
70	63	83	164	97	186	196	1	68	216	536	475	114	566	183	376
27	31	197	26	157	1	18	67	7	98	677	1504	345	393	148	93
111	367	239	463	508	17	282	77	147	877	0	0	445	447	435	727
240	335	136	16	1	7	142	158	657	130	420	123	93	82	592	158
806	45	49	6	2	31	67	55	18	65	86	294	24	157	152	386
0	100	154	17	426	144	167	6	128	193	144	171	1101	392	92	149
24	156	813	67	374	565	573	0	262	772	456	514	1461	620	137	36
86	163	8	55	0	47	21	239	63	74	706	196	146	0	175	326
0	0	0	5	29	119	28	0	0	0	173	361	0	0	0	20
919	635	0	10	108	0	5	30	17	0	116	176	45	21	384	2

A common data structure in high-throughput biology

- *Samples*
- *Conditions*
- *Stimuli*
- *Time points*
- *Tissues*
- *Disease states*
- *Locations*
- *Cell types*
- *…*

- *Expression*
- *Abundance*
- *Count*
- *Rate*
- *…*

Samples

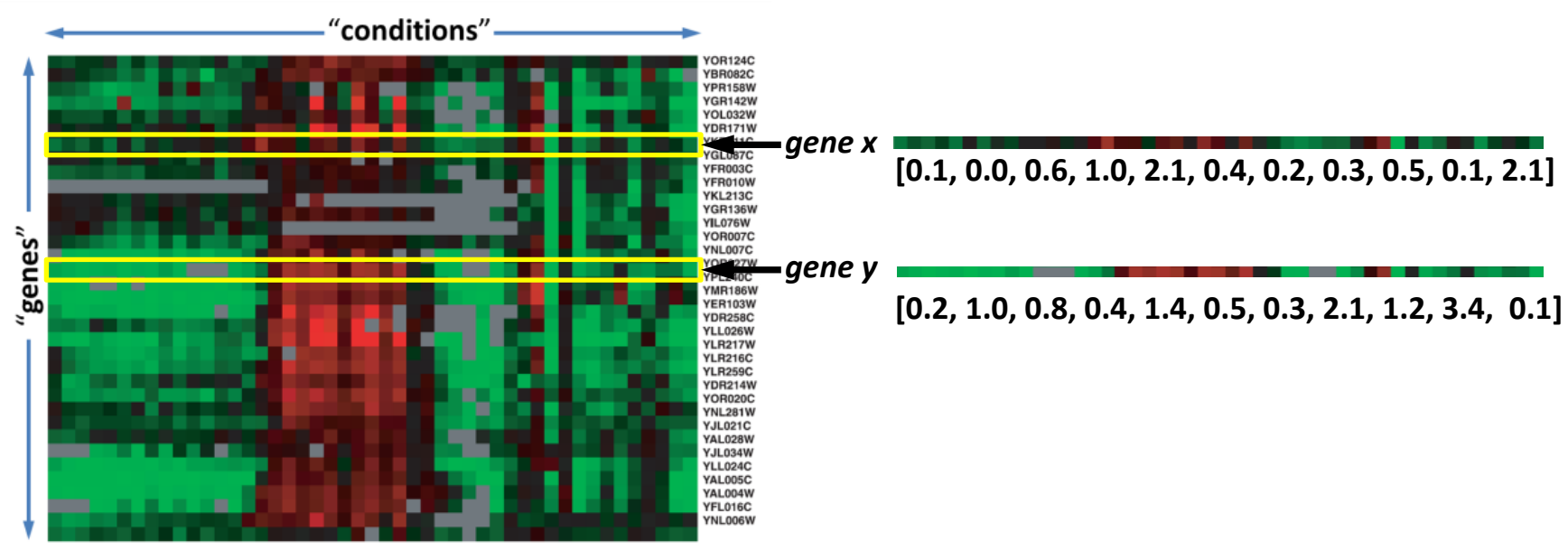
Biological Entities

Measurements

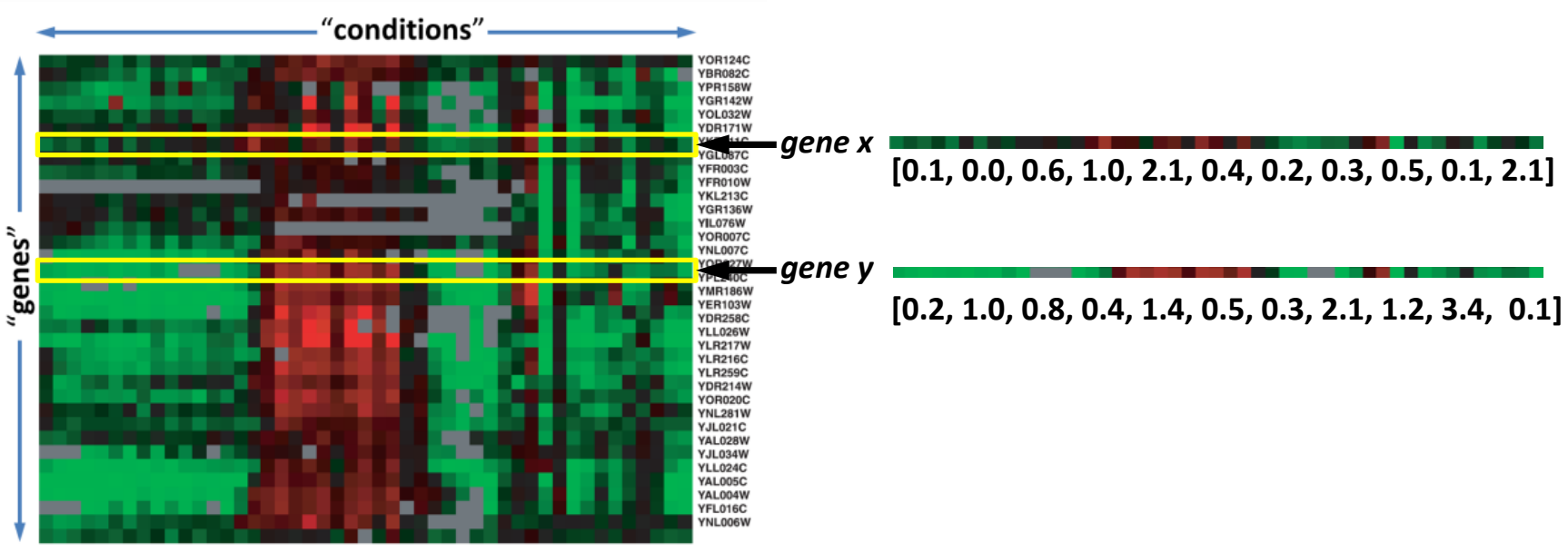
776	2905	5317	3275	4580	1083	2169	1817	1553	1391	0	0	0	0	410	0
77	252	117	162	85	250	415	518	37	314	846	651	218	1044	480	942
70	63	83	164	97	186	196	126	68	216	536	475	114	566	183	376
27	31	197	26	157	197	184	67	7	98	677	1504	345	393	148	93
111	367	239	463	508	175	282	77	147	877	0	0	445	447	435	727
240	335	136	167	104	78	142	158	657	130	420	123	93	82	592	158
806	45	49	6	27	31	67	55	18	65	86	294	24	157	152	386
0	100	154	176	426	144	167	6	128	193	144	171	1101	392	92	149
24	156	813	679	374	565	573	0	262	772	456	514	1461	620	137	36
86	163	8	55	0	47	21	239	63	74	706	196	146	0	175	326
0	0	0	5	29	119	28	0	0	0	173	361	0	0	0	20
919	635	0	10	108	0	5	30	17	0	116	176	45	21	384	2
51	154	36	0	48	129	95	30	0	54	168	143	86	357	53	33
516	3	114	109	186	54	6	99	565	118	0	14	0	0	378	163
70	63	83	164	97	186	196	1	68	216	536	475	114	566	183	376
27	31	197	26	157	1	18	67	7	98	677	1504	345	393	148	93
111	367	239	463	508	17	282	77	147	877	0	0	445	447	435	727
240	335	136	16	1	78	142	158	657	130	420	123	93	82	592	158
806	45	49	6	27	31	67	55	18	65	86	294	24	157	152	386
0	100	154	17	426	144	167	6	128	193	144	171	1101	392	92	149
24	156	813	679	374	565	573	0	262	772	456	514	1461	620	137	36
86	163	8	55	0	47	21	239	63	74	706	196	146	0	175	326
0	0	0	5	29	119	28	0	0	0	173	361	0	0	0	20
919	635	0	10	108	0	5	30	17	0	116	176	45	21	384	2

- *Genes*
- *Proteins*
- *Transcripts*
- *Species*
- *….*

A common data structure in high-throughput biology

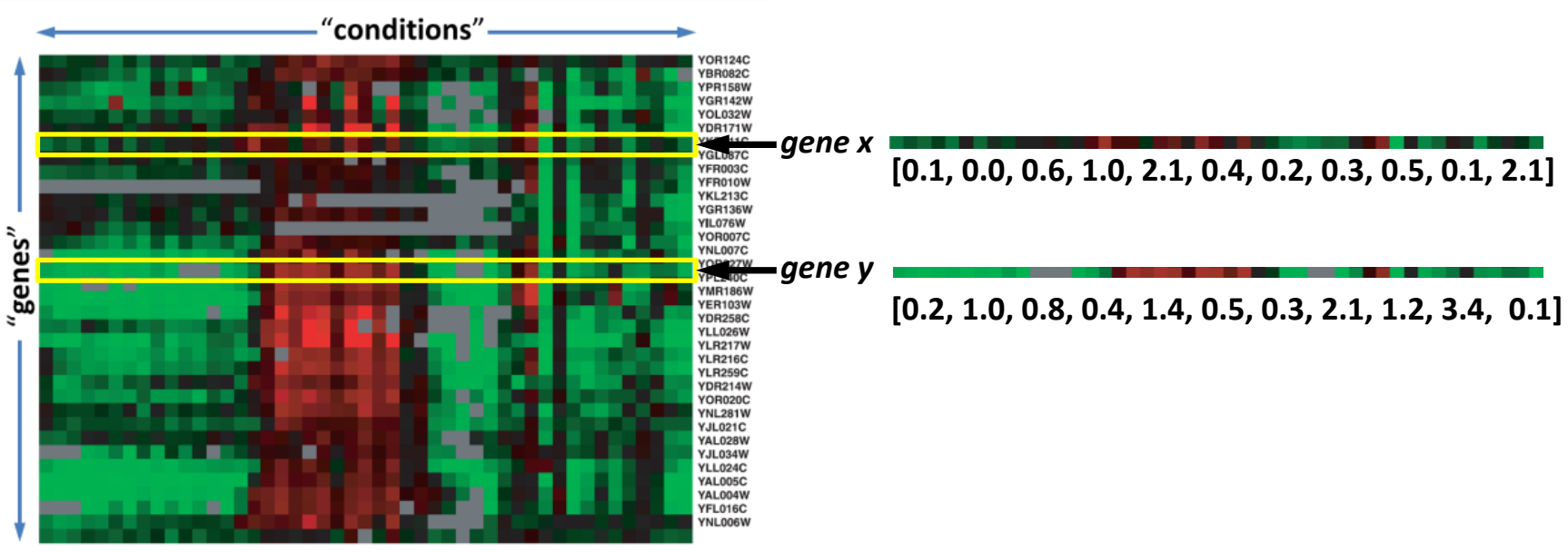


The clustering problem

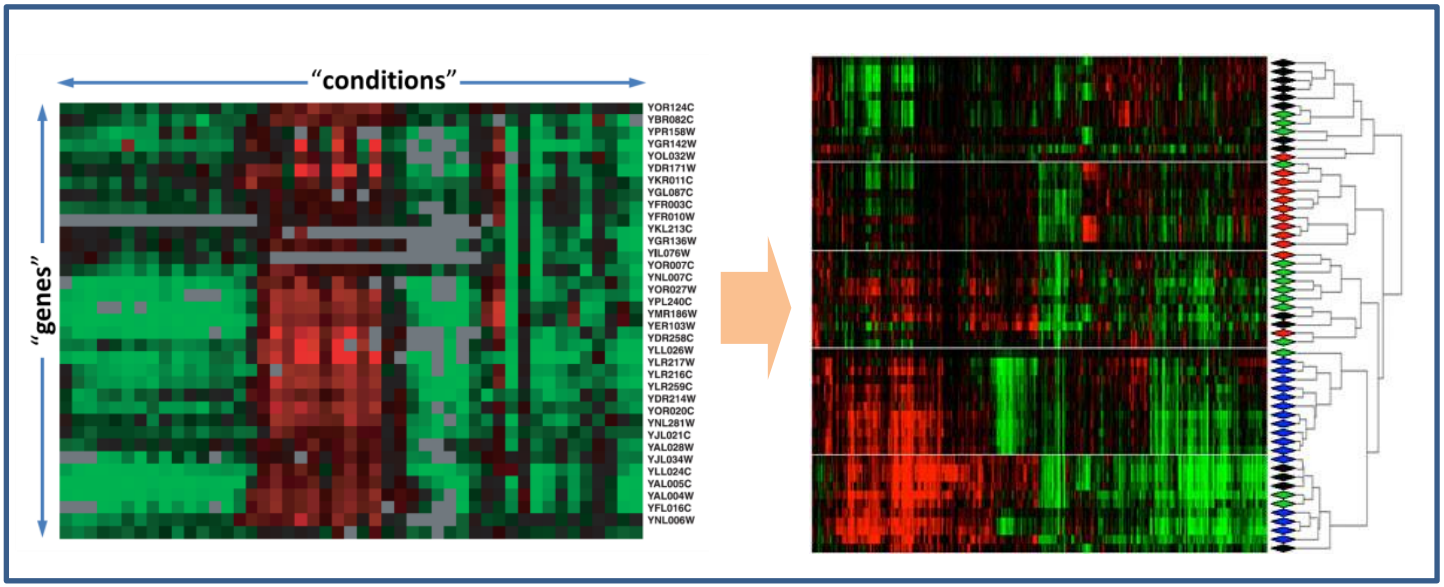


The clustering problem

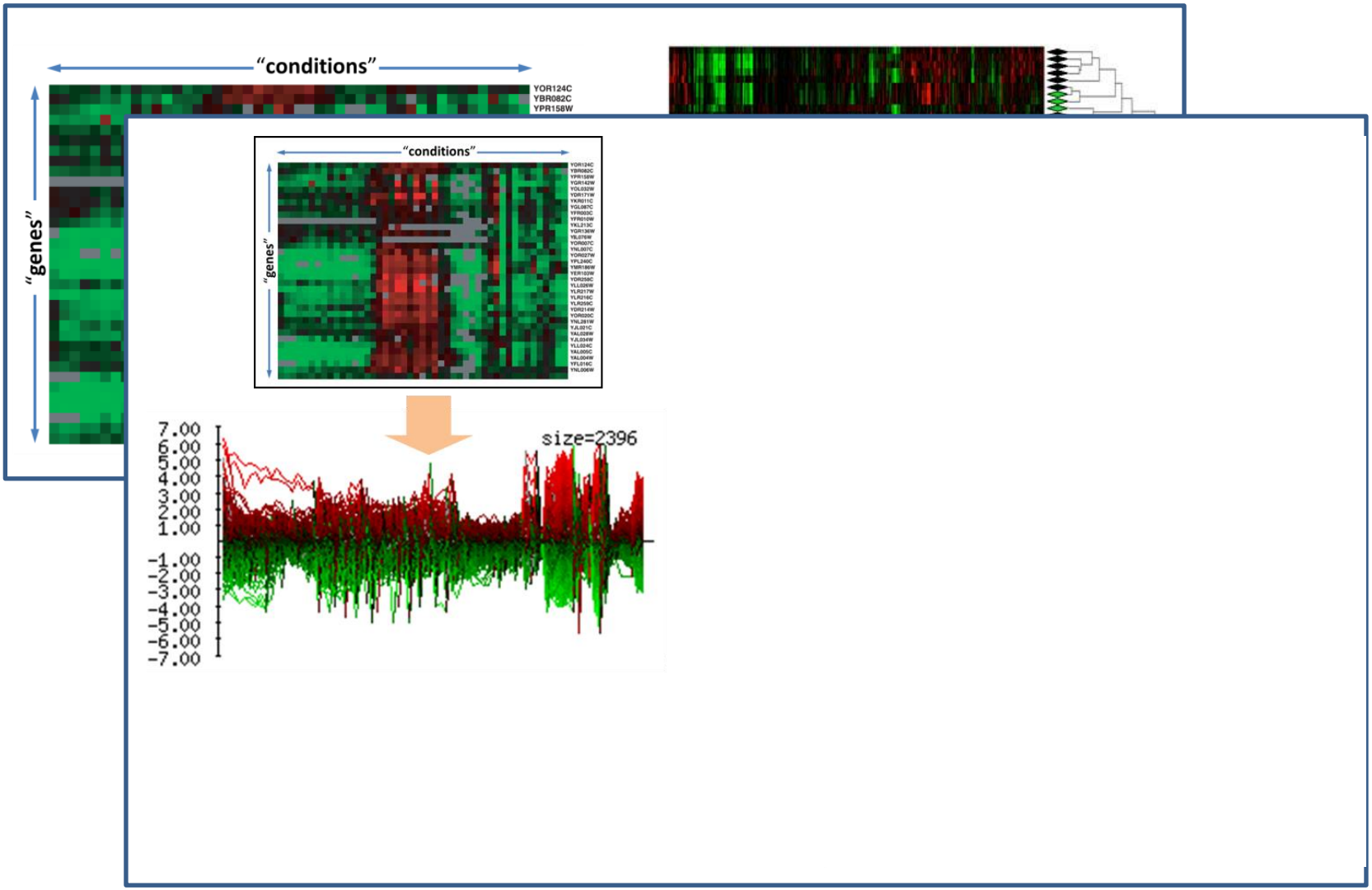
- **The goal of gene clustering process is to partition the genes into distinct sets such that genes that are assigned to the same cluster are “similar”, while genes assigned to different clusters are “non-similar”.**



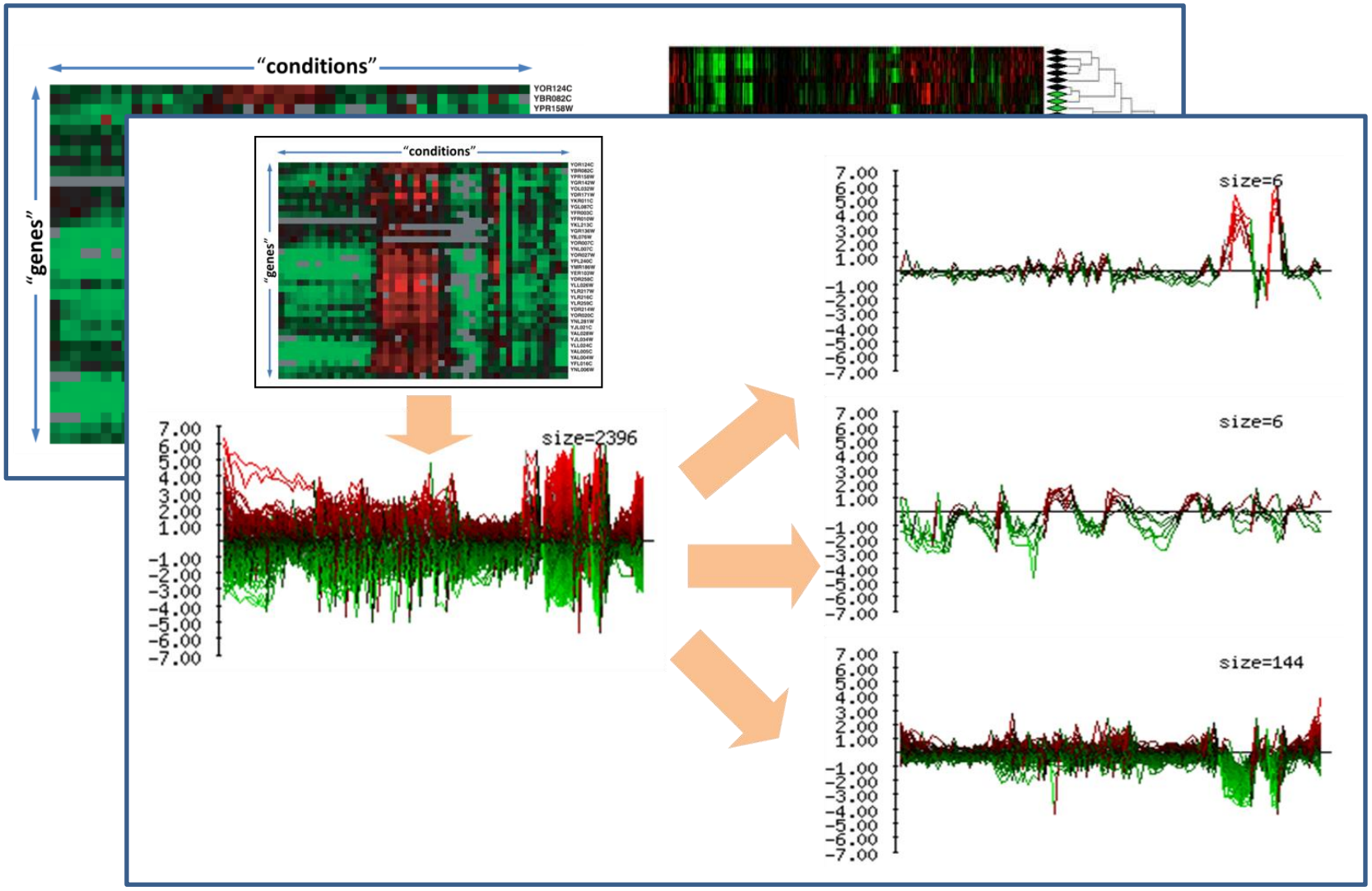
Different views of clustering …



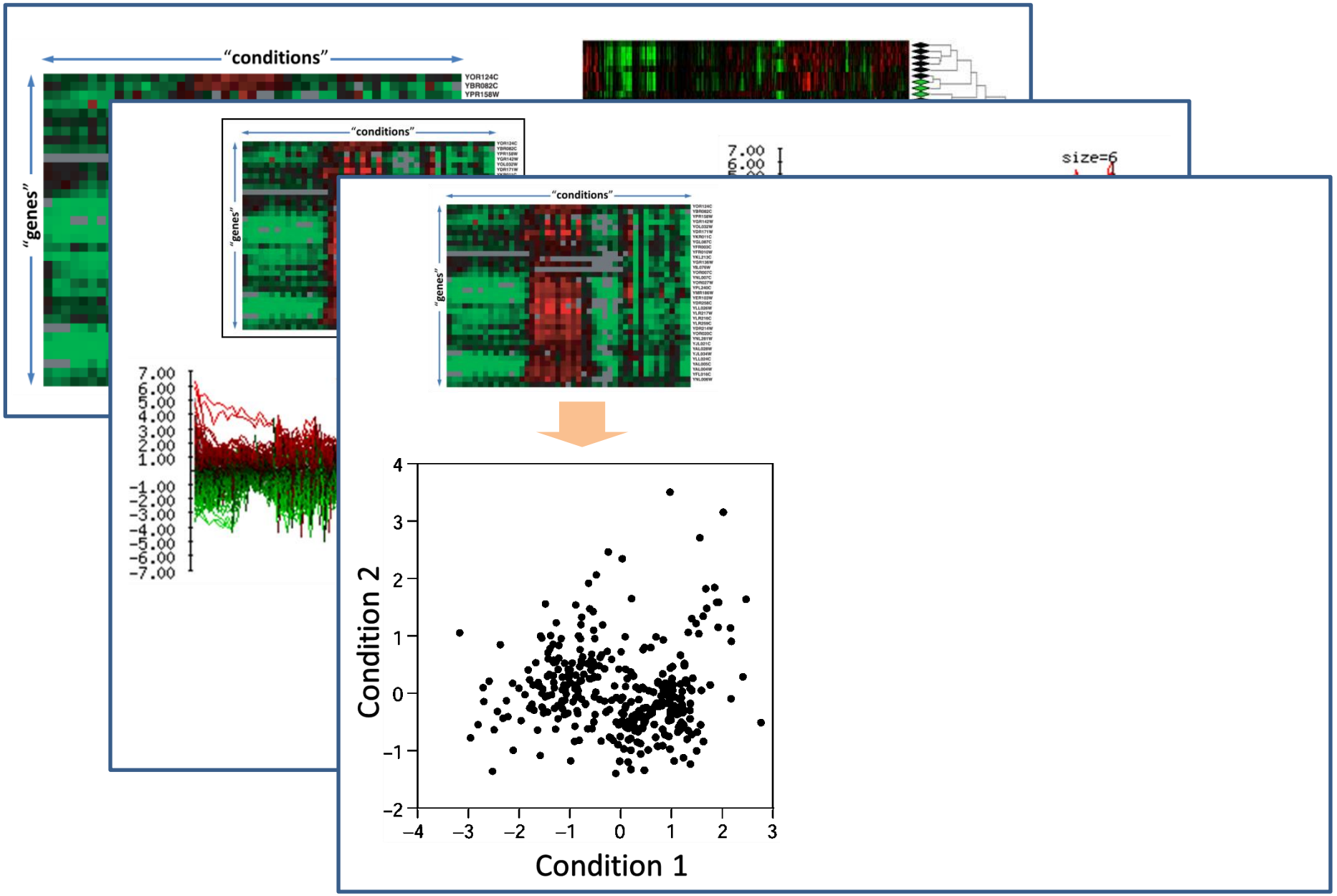
Different views of clustering …



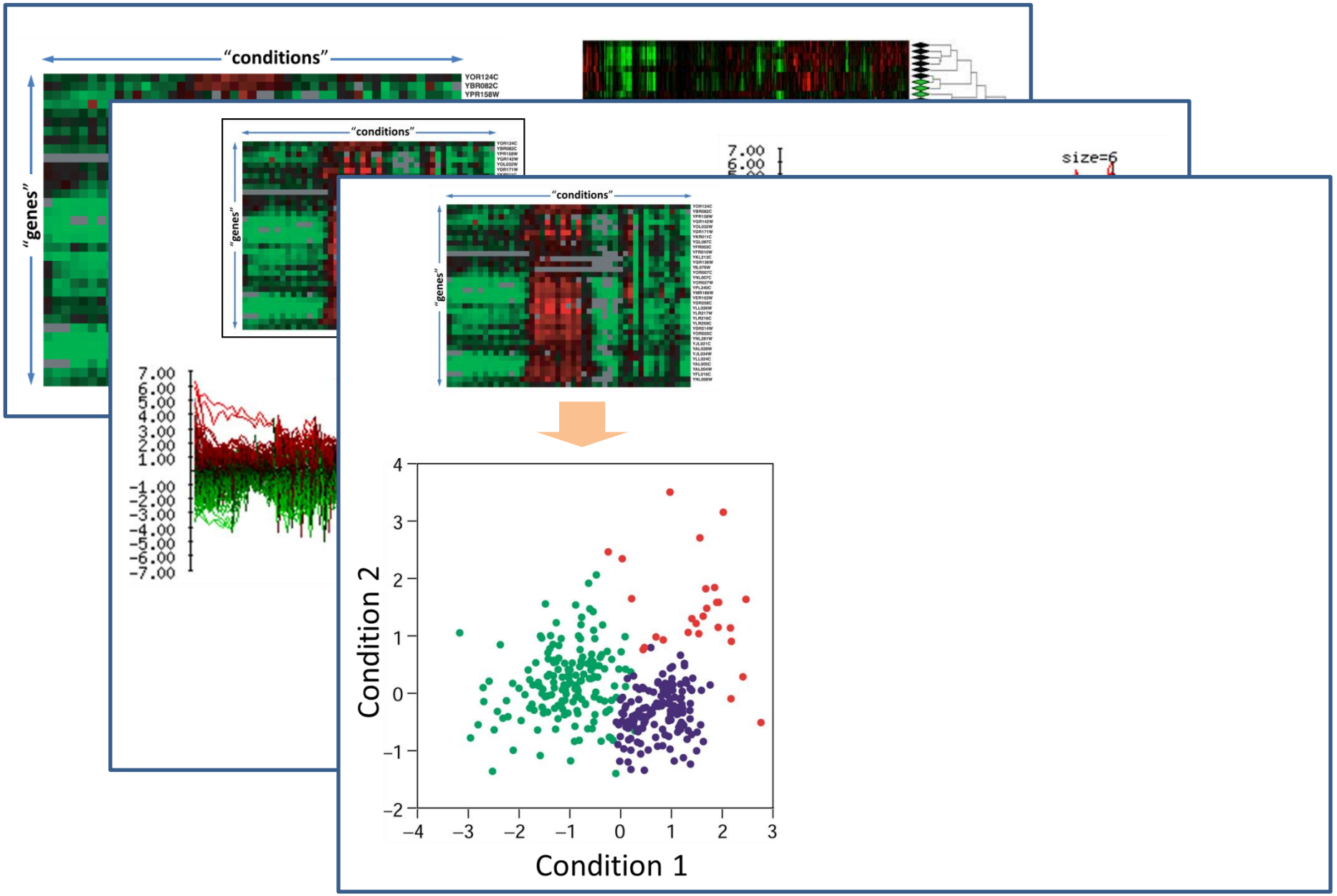
Different views of clustering …



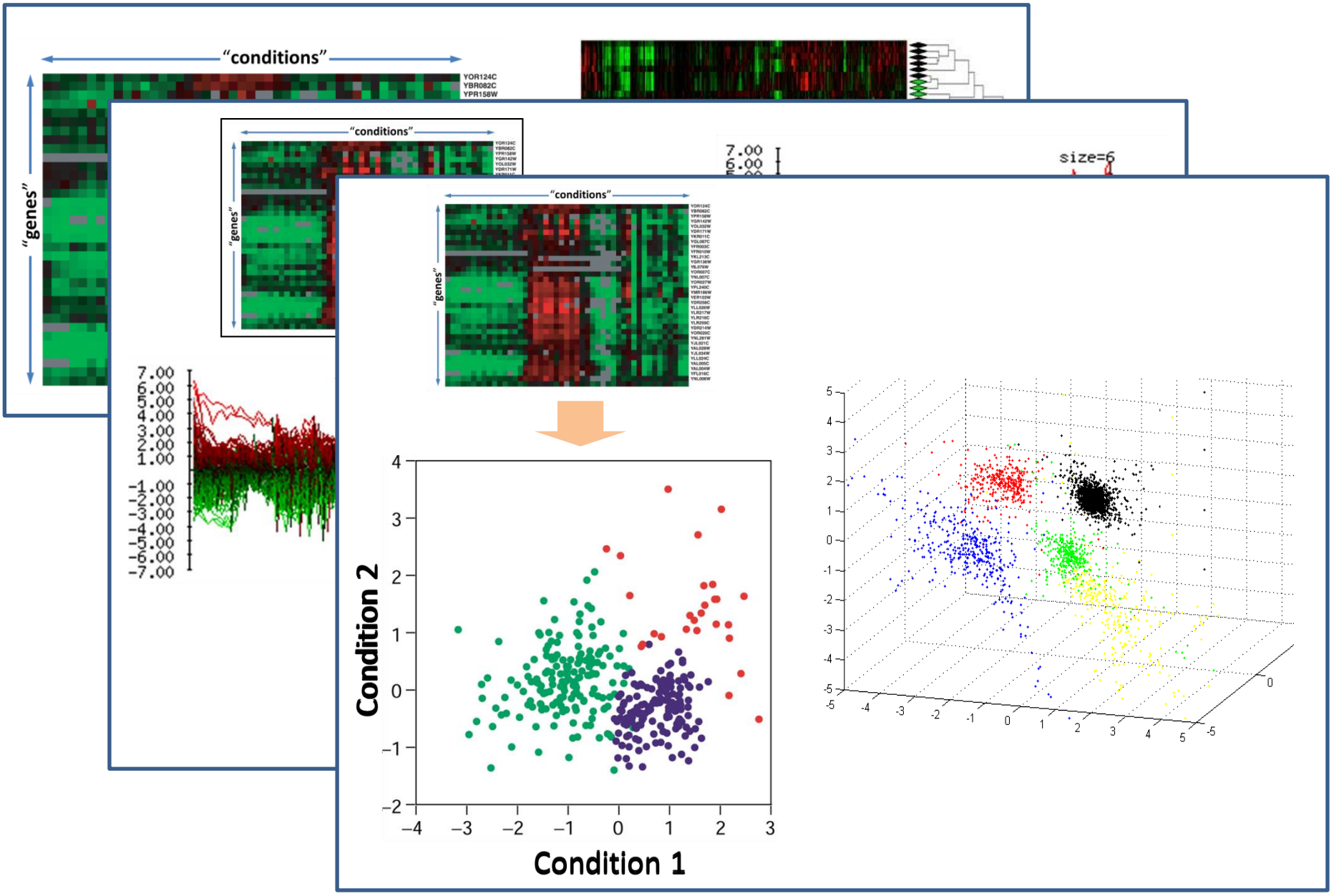
Different views of clustering …



Different views of clustering …

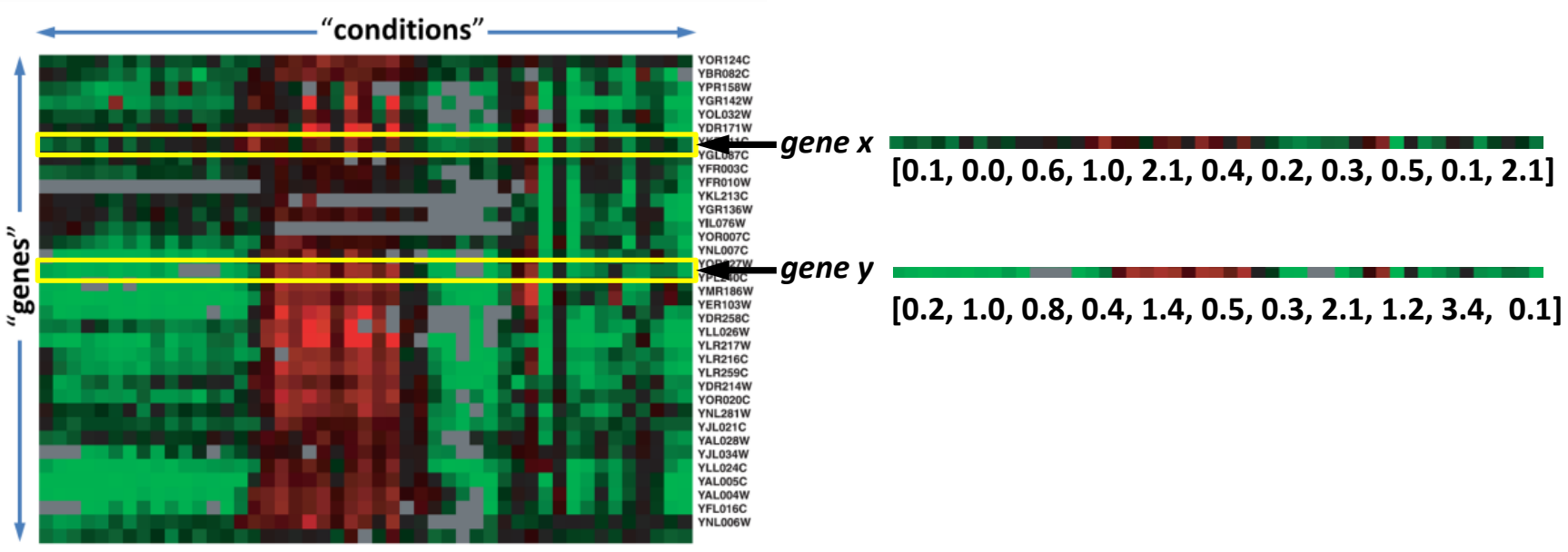


Different views of clustering …



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”.**



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).

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.**

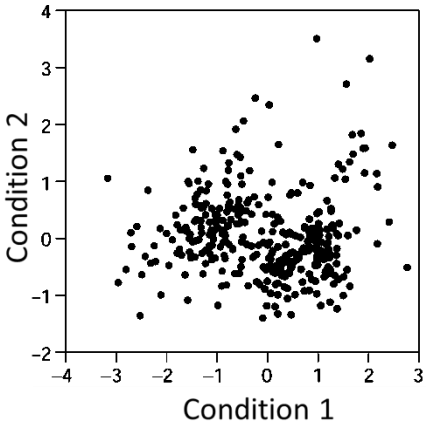
Why is clustering a hard computational problem?

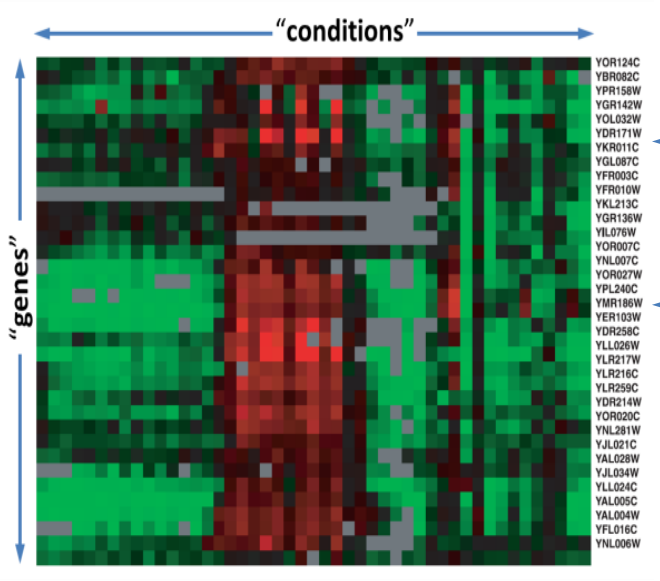
One problem, numerous solutions

- 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 (why?)**.
- 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.)

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)





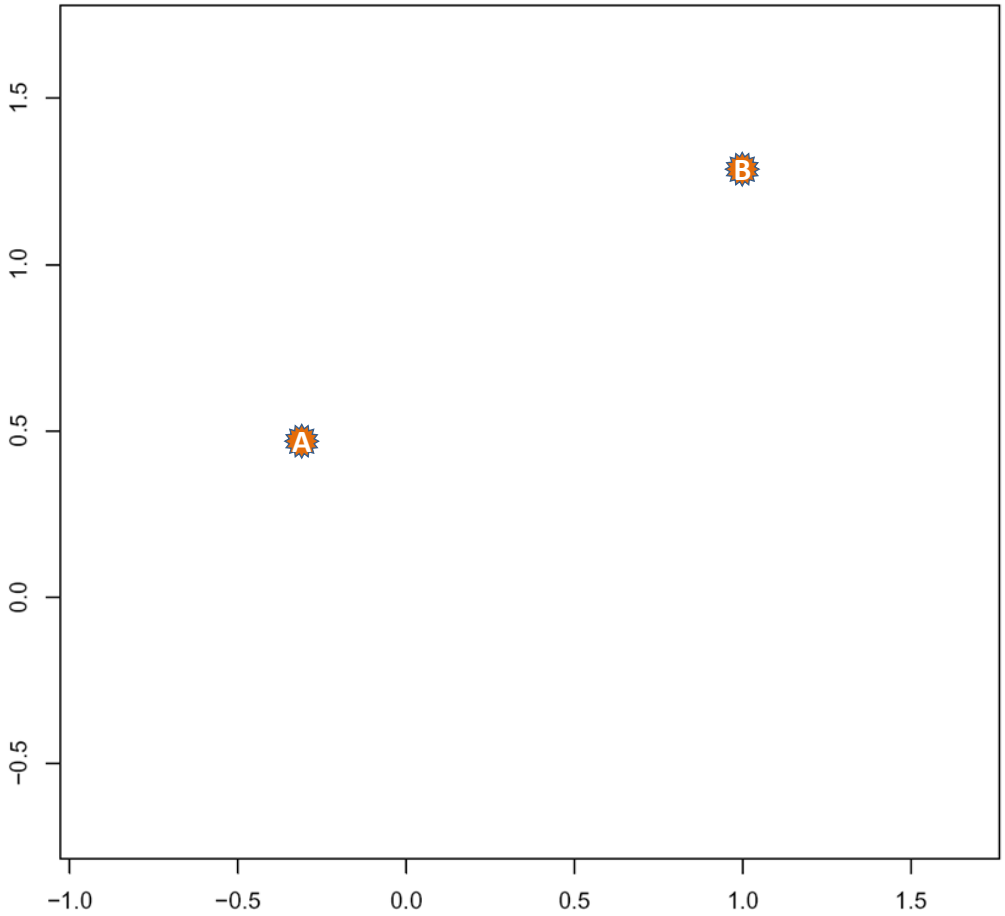
"Point" 1 : [0.1 0.0 0.6 1.0 2.1 0.4 0.2]

"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

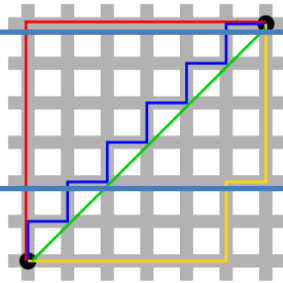
$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- 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 (x=0,y=0).
  - 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

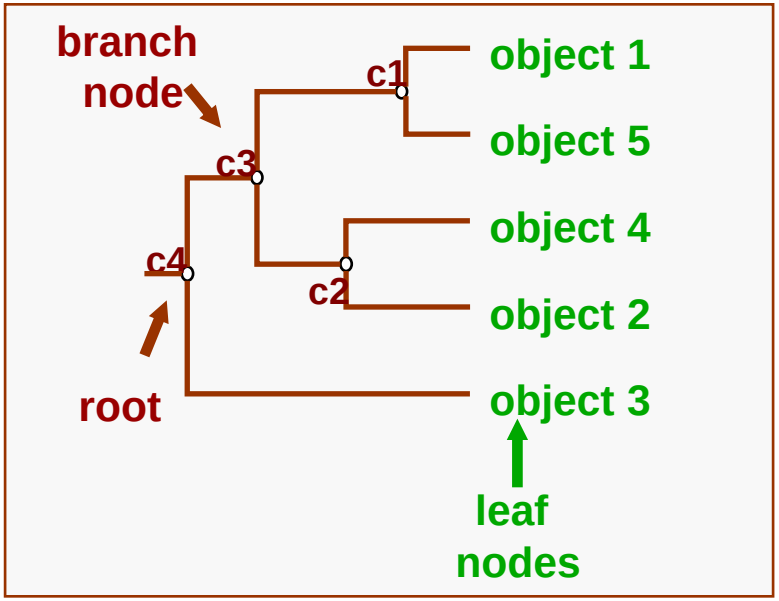
Hierarchical clustering

- **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

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

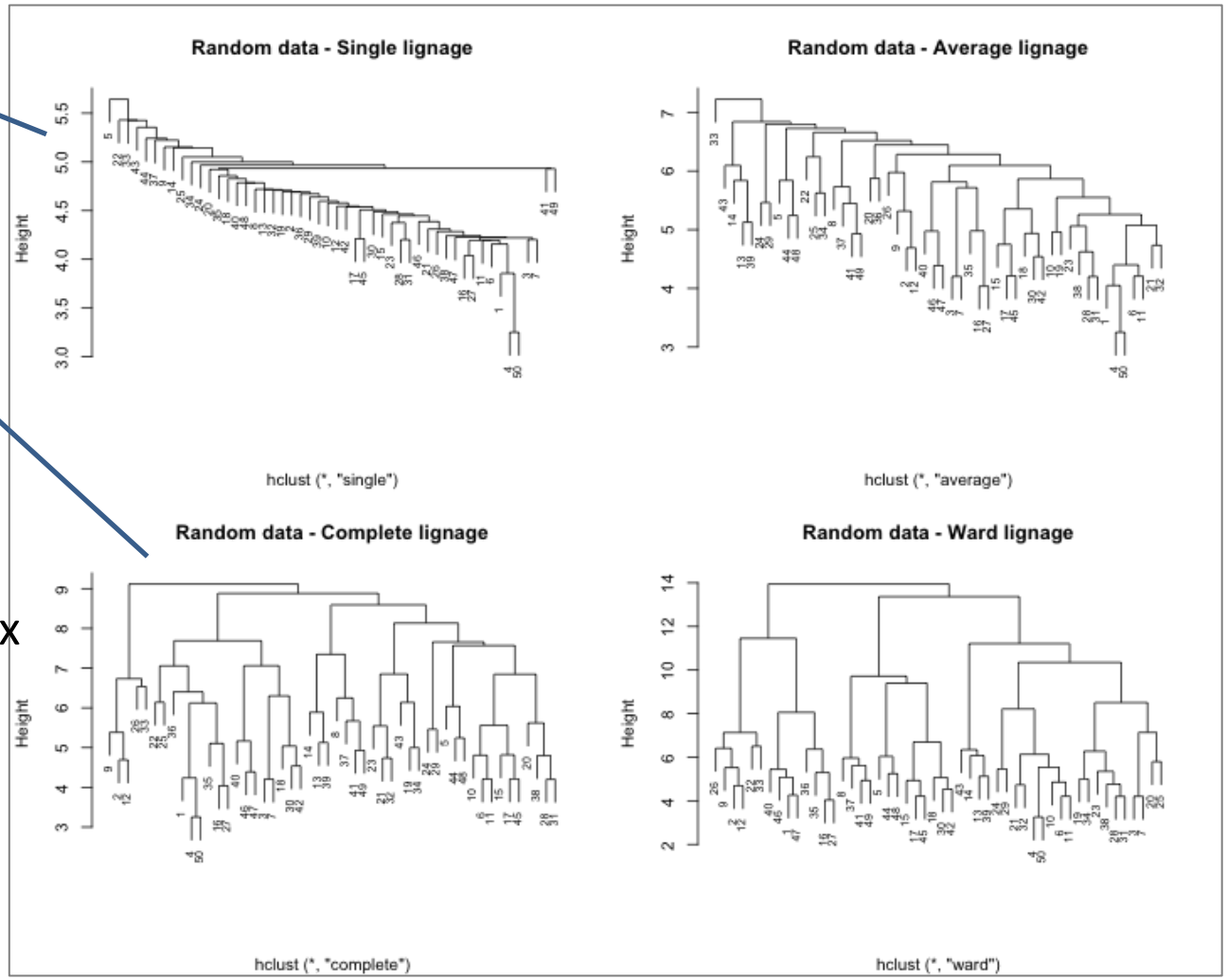
Impact of the agglomeration rule

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

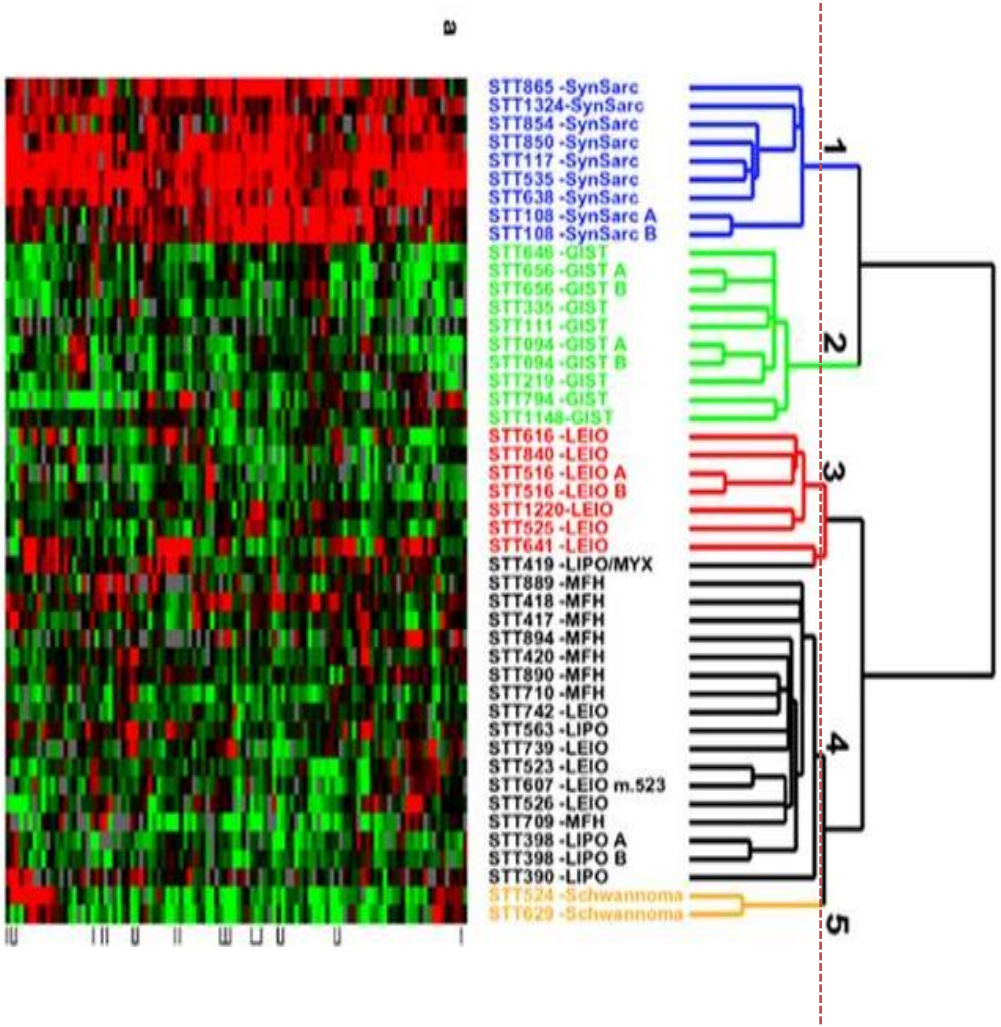
Single-linkage typically creates nesting clusters

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 - Summary

- “**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.