Statistical Analysis of Microarray Data

Clustering

Jacques van Helden

Jacques.van-Helden@univ-amu.fr
Aix-Marseille Université, France
Technological Advances for Genomics and Clinics
(TAGC, INSERM Unit U1090)
http://jacques.van-helden.perso.luminy.univ-amu.fr/

FORMER ADDRESS (1999-2011)
Université Libre de Bruxelles, Belgique
Bioinformatique des Génomes et des Réseaux (BiGRe lab)
http://www.bigre.ulb.ac.be/

Contents

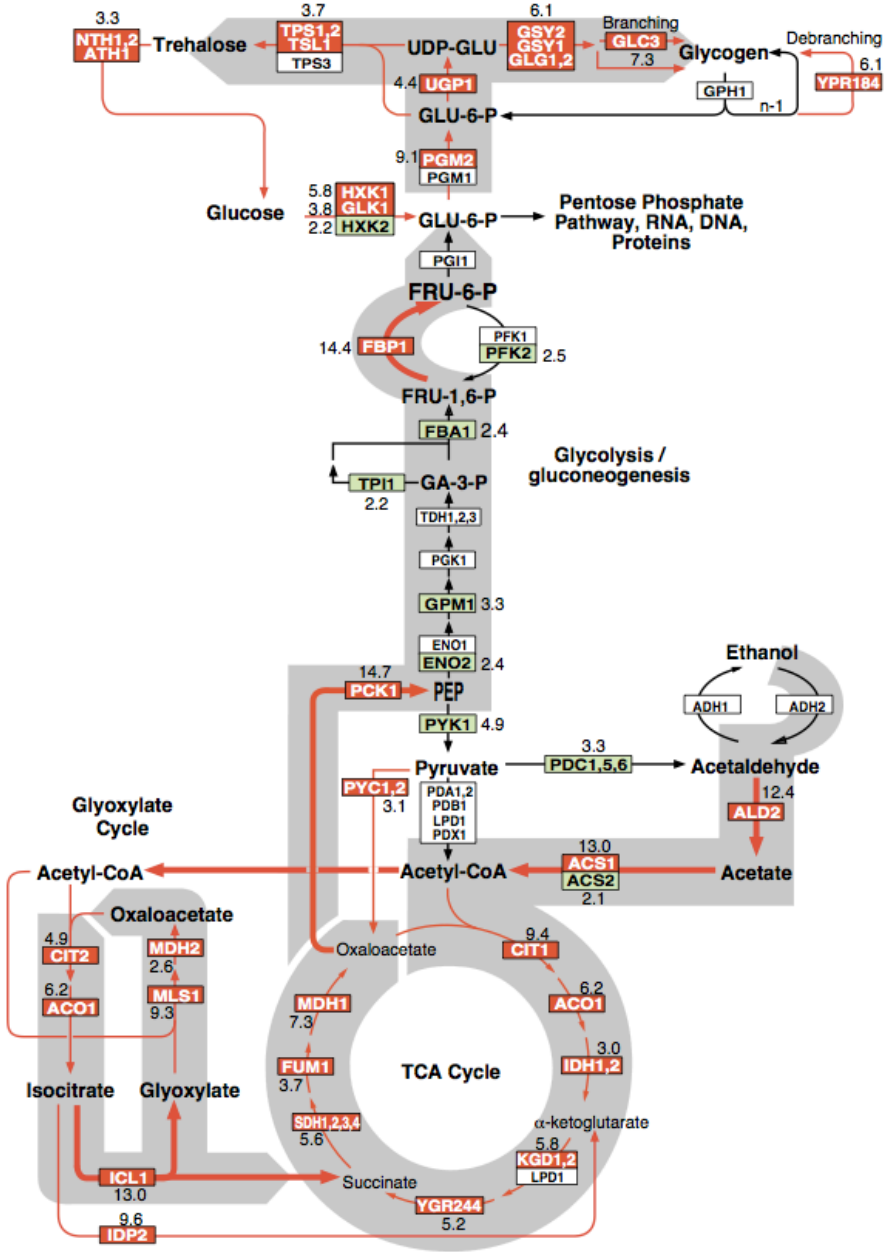
Introduction to clustering

- Clustering is an *unsupervised* approach
  - Class discovery: starting from a set of objects, group them into classes, without any prior knowledge of these classes.
- There are many clustering methods
  - hierarchical
  - k-means
  - self-organizing maps (SOM)
  - knn
  - ...
- The results vary drastically depending on
  - clustering method
  - similarity or dissimilarity metric
  - additional parameters specific to each clustering method (e.g. number of centres for the k-mean, agglomeration rule for hierarchical clustering, ...)

Data sets

Diauxic shift

- DeRisi et al published the first article describing a full-genome monitoring of gene expression data.
- This article reported an experiment called “diauxic shift” with with 7 time points.
- Initially, cells are grown in a glucose-rich medium.
- As time progresses, cells
  - Consume glucose -> when glucose becomes limiting
    - Glycolysis stops
    - Gluconeogenesis is activated to produce glucose
  - Produce by-products -> the culture medium becomes polluted/
    - Stress response



- DeRisi et al. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science (1997) vol. 278 (5338) pp. 680-6

Cell cycle data

- Spellman et al. (1998)
- Time profiles of yeast cells followed during cell cycle.
- Several experiments were regrouped, with various ways of synchronization (elutriation, cdc mutants, …)
- ~800 genes showing a periodic patterns of expression were selected (by Fourier analysis)

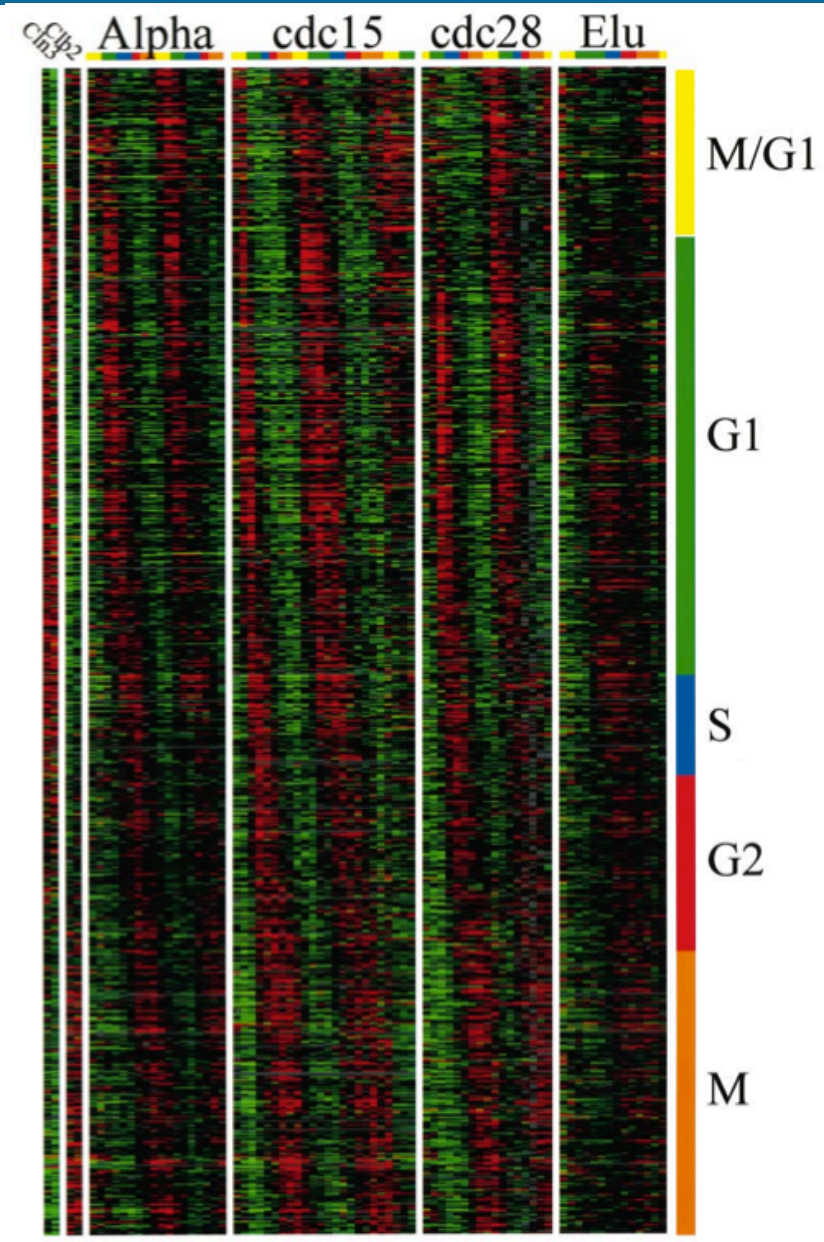
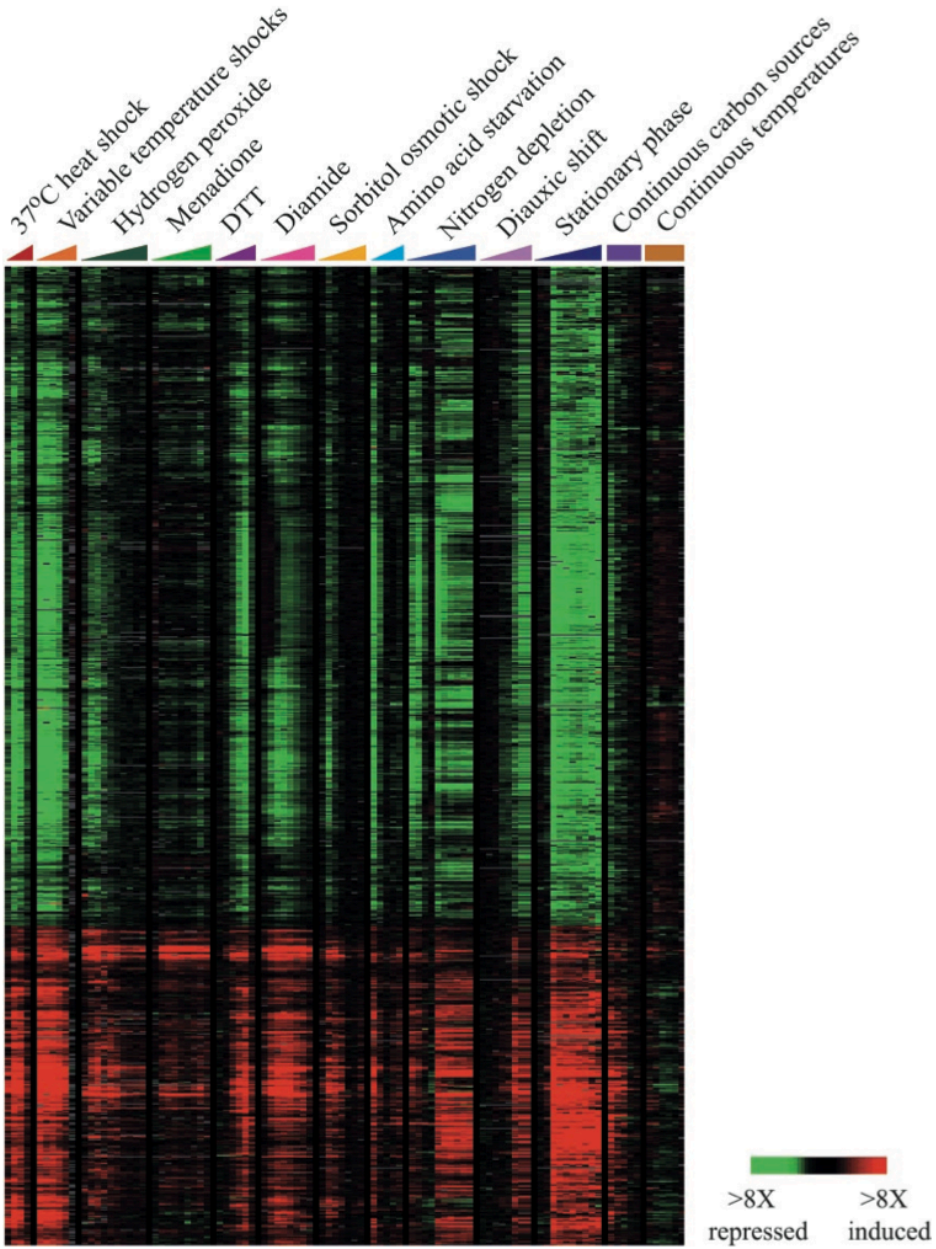


Figure 1.

- Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell (1998) vol. 9 (12) pp. 3273-97

Gene expression data – response to environmental changes

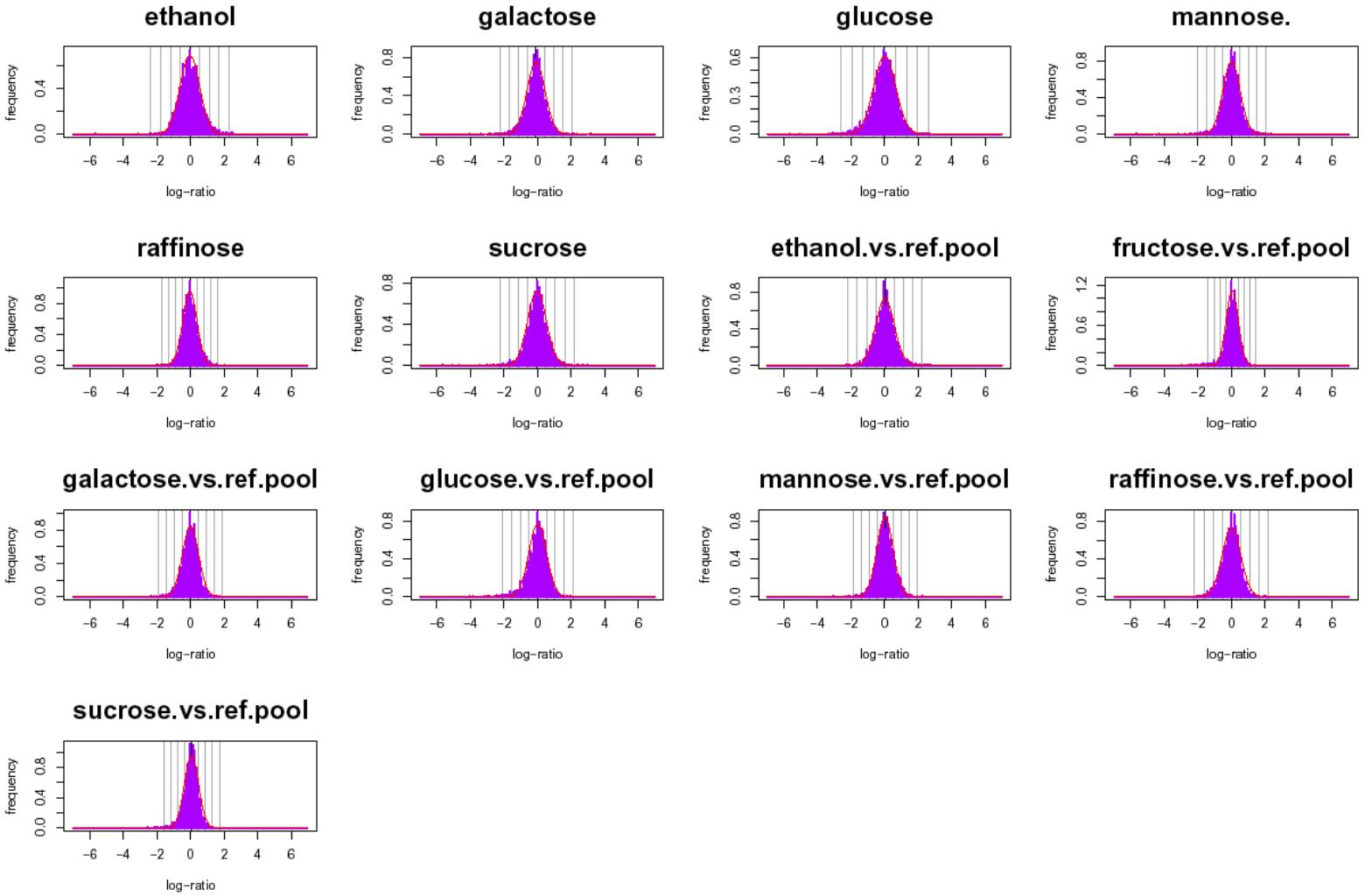
- Gasch et al. (2000), 173 chips (stress response, heat shock, drugs, carbon source, …)



- Gasch et al. Genomic expression programs in the response of yeast cells to environmental changes. Mol Biol Cell (2000) vol. 11 (12) pp. 4241-57.

Gene expression data - carbon sources

- Gasch et al. (2000), 173 chips (stress response, heat shock, drugs, carbon source, …)
- We selected the 13 chips with the response to different carbon sources.

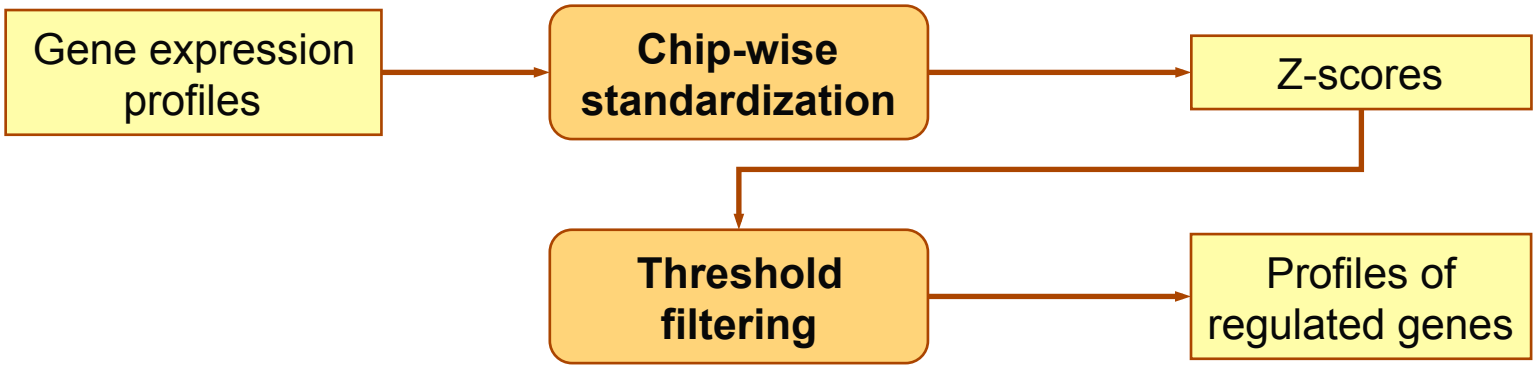


- Gasch et al. Genomic expression programs in the response of yeast cells to environmental changes. Mol Biol Cell (2000) vol. 11 (12) pp. 4241-57.

Data standardization and filtering

- For the cell cycle experiments, genes had already been filtered in the original publication. We used the 800 selected genes for the analysis.
- For the diauxic shift and carbon source experiments, each chip contain >6000 genes, most of which are un-regulated.
- Standardization
  - We applied a chip-wise standardization (centring and scaling) with robust estimates (median and IQR) on each chip.
- Filtering
  - Z-scores obtained after standardization were converted
    - to P-value (normal distribution)
    - to E-value (=P-value*N)
  - Only genes with an E-value < 1 were retained for clustering.

Filtering of carbon source data



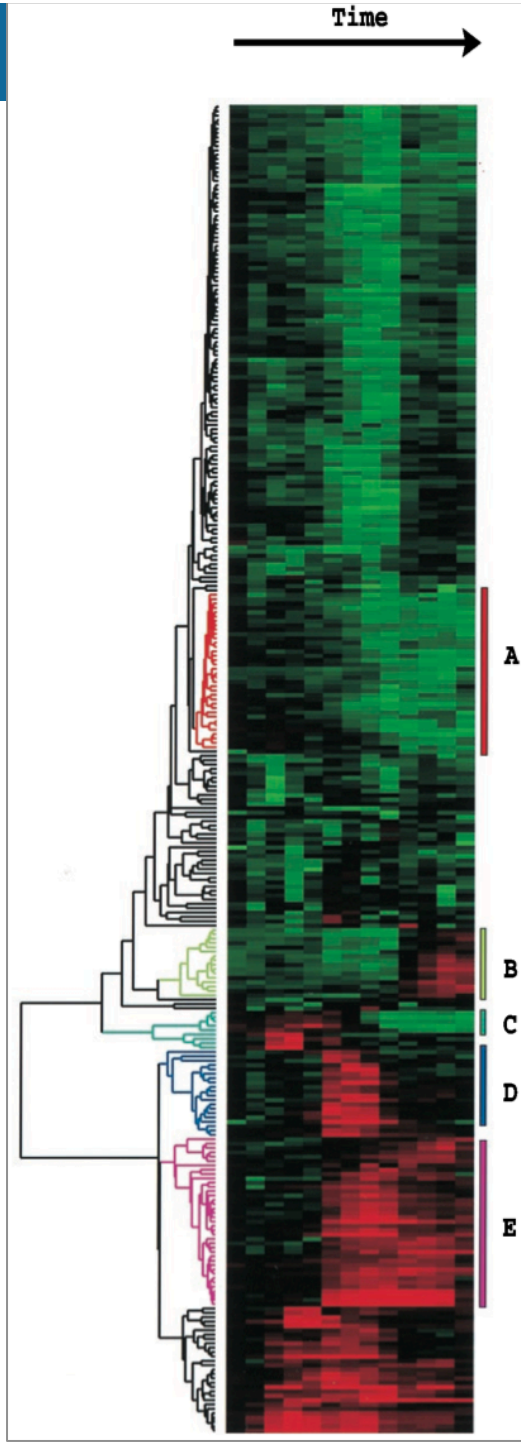
Carbon sources

ORF	ethanol	galactose	glucose	mannose.	raffinose	sucrose	ethanol.vs.ref.pool	fructose.vs.ref.pool	galactose.vs.ref.pool	glucose.vs.ref.pool	mannose.vs.ref.pool	raffinose.vs.ref.pool	sucrose.vs.ref.pool
YAL066W	0.71	-1.87	-1.15	-4.90	-1.85	-3.81	-3.34	-0.96	-3.41	-1.04	0.36	-1.55	-0.87
YAR008W	-2.70	0.36	0.03	-4.90	-0.94	-0.53	0.64	-0.53	-2.57	-0.73	0.38	-1.75	-0.55
YAR071W	-5.43	-1.22	2.73	-0.44	-0.24	3.24	-6.69	1.10	-5.21	1.39	-0.70	0.22	2.94
YBL005W	1.40	3.05	3.97	4.92	1.18	5.52	-0.53	0.79	-0.84	-1.00	1.12	-2.26	1.23
YBL015W	4.00	0.28	-3.46	-3.65	-2.38	-4.94	3.26	-4.64	0.59	-3.76	-1.62	1.08	-5.37
YBL043W	3.91	-1.16	-4.89	-4.90	-1.61	-4.76	4.47	-6.97	-0.61	-6.67	-7.12	0.78	-9.73
YBR018C	-9.68	5.53	-8.66	-11.19	-13.49	-10.23	-9.81	-15.15	6.32	-10.89	-13.01	-12.10	-13.73
YBR019C	-9.68	6.16	-7.77	-11.19	-12.09	-9.17	-9.42	-12.93	6.07	-10.58	-10.90	-9.08	-11.97
YBR020W	-9.68	6.05	-8.66	-11.19	-13.49	-10.23	-10.04	-12.70	6.83	-12.82	-13.01	-8.95	-14.94
…	…	…	…	…	…	…	…	…	…	…	…	…	…

Hierarchical clustering

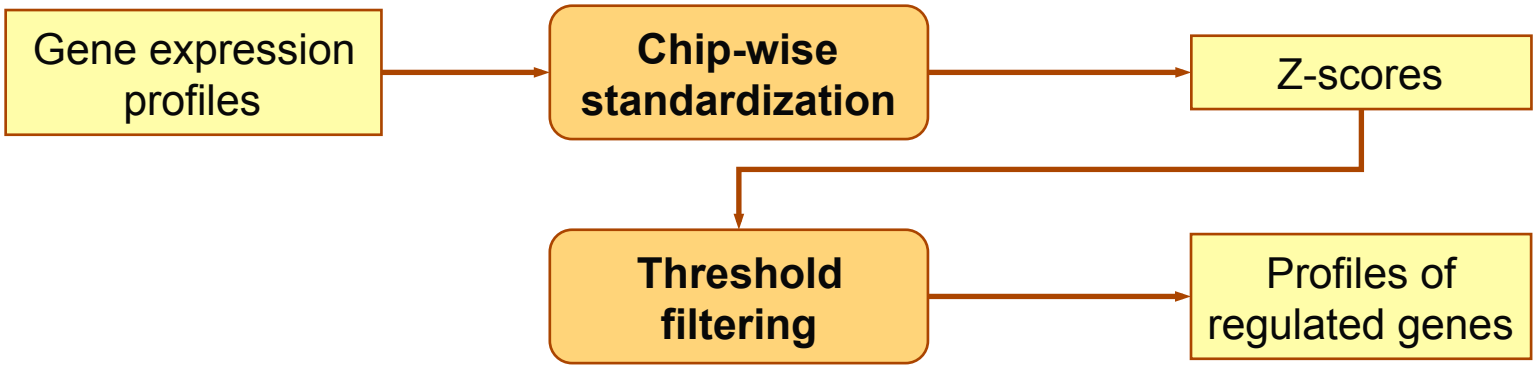
Hierarchical clustering of expression profiles

- In 1998, Eisen et al.
  - Implemented a software tool called *Cluster*, which combine hierarchical clustering and heatmap visualization.
  - Applied it to extract clusters of co-expressed genes from various types of expression profiles.



- Eisen et al. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A (1998) vol. 95 (25) pp. 14863-8

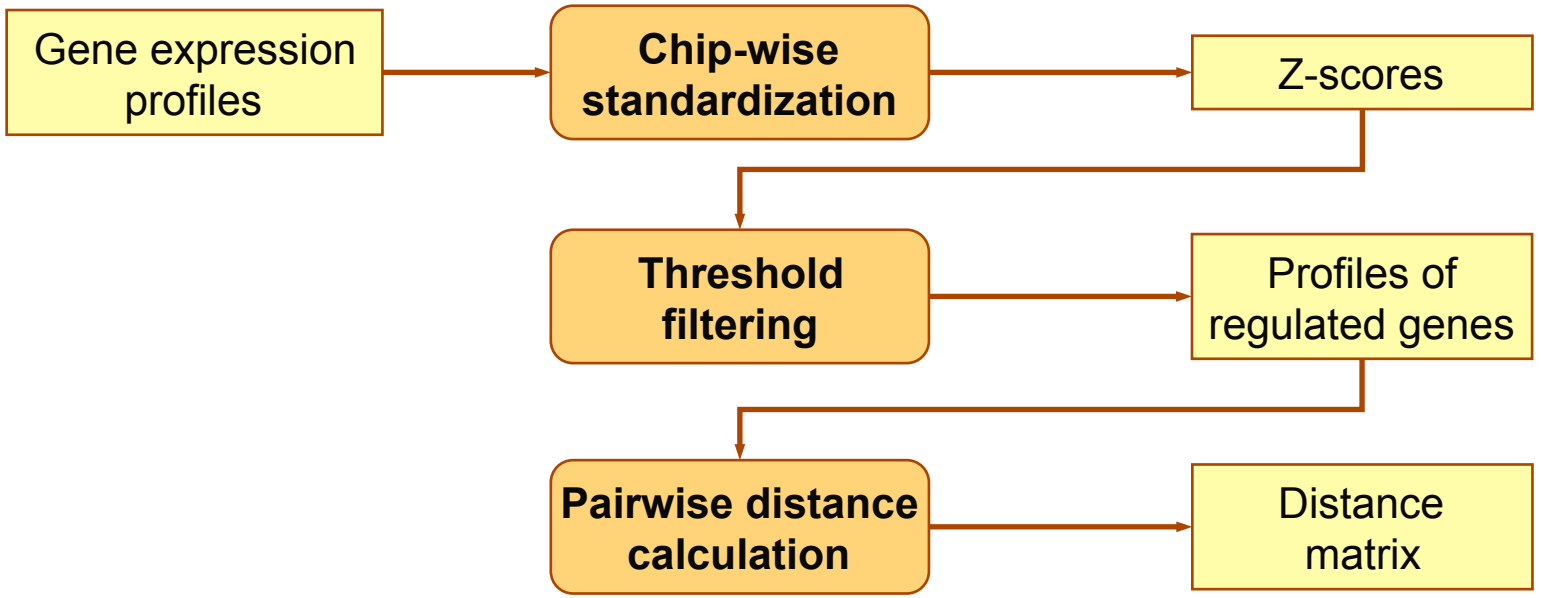
Clustering with gene expression data



Carbon sources

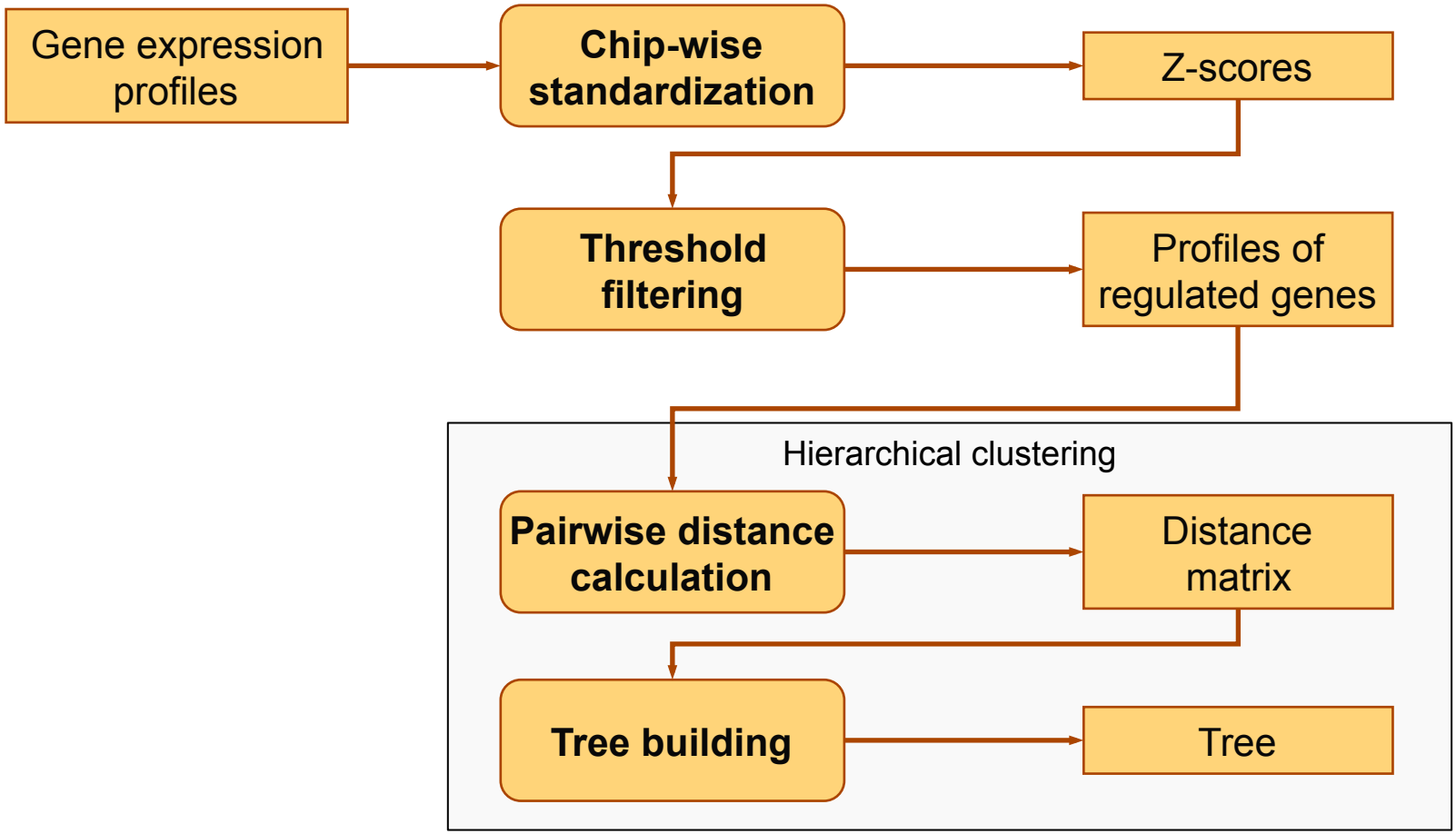
ORF	ethanol	galactose	glucose	mannose.	raffinose	sucrose	ethanol.vs.ref.pool	fructose.vs.ref.pool	galactose.vs.ref.pool	glucose.vs.ref.pool	mannose.vs.ref.pool	raffinose.vs.ref.pool	sucrose.vs.ref.pool
YAL066W	0.71	-1.87	-1.15	-4.90	-1.85	-3.81	-3.34	-0.96	-3.41	-1.04	0.36	-1.55	-0.87
YAR008W	-2.70	0.36	0.03	-4.90	-0.94	-0.53	0.64	-0.53	-2.57	-0.73	0.38	-1.75	-0.55
YAR071W	-5.43	-1.22	2.73	-0.44	-0.24	3.24	-6.69	1.10	-5.21	1.39	-0.70	0.22	2.94
YBL005W	1.40	3.05	3.97	4.92	1.18	5.52	-0.53	0.79	-0.84	-1.00	1.12	-2.26	1.23
YBL015W	4.00	0.28	-3.46	-3.65	-2.38	-4.94	3.26	-4.64	0.59	-3.76	-1.62	1.08	-5.37
YBL043W	3.91	-1.16	-4.89	-4.90	-1.61	-4.76	4.47	-6.97	-0.61	-6.67	-7.12	0.78	-9.73
YBR018C	-9.68	5.53	-8.66	-11.19	-13.49	-10.23	-9.81	-15.15	6.32	-10.89	-13.01	-12.10	-13.73
YBR019C	-9.68	6.16	-7.77	-11.19	-12.09	-9.17	-9.42	-12.93	6.07	-10.58	-10.90	-9.08	-11.97
YBR020W	-9.68	6.05	-8.66	-11.19	-13.49	-10.23	-10.04	-12.70	6.83	-12.82	-13.01	-8.95	-14.94
…	…	…	…	…	…	…	…	…	…	…	…	…	…

Hierarchical clustering on gene expression data



	YAL066W	YAR008W	YAR071W	YBL005W	YBL015W	YBL043W	YBR018C	YBR019C	YBR020W	YBR054W	…
YAL066W	0.00	6.82	12.99	16.33	11.64	17.39	36.41	32.52	36.07	12.00	…
YAR008W	6.82	0.00	11.70	13.69	12.58	18.18	37.51	33.46	37.18	12.36	…
YAR071W	12.99	11.70	0.00	13.32	21.77	26.62	42.48	38.48	42.15	21.09	…
YBL005W	16.33	13.69	13.32	0.00	19.52	25.04	44.95	41.16	44.62	17.86	…
YBL015W	11.64	12.58	21.77	19.52	0.00	8.51	34.47	30.79	33.77	6.46	…
YBL043W	17.39	18.18	26.62	25.04	8.51	0.00	31.74	28.64	30.90	11.13	…
YBR018C	36.41	37.51	42.48	44.95	34.47	31.74	0.00	5.12	4.66	35.84	…
YBR019C	32.52	33.46	38.48	41.16	30.79	28.64	5.12	0.00	4.81	32.58	…
YBR020W	36.07	37.18	42.15	44.62	33.77	30.90	4.66	4.81	0.00	35.63	…
YBR054W	12.00	12.36	21.09	17.86	6.46	11.13	35.84	32.58	35.63	0.00	…
…	…	…	…	…	…	…	…	…	…	…	…

Hierarchical clustering on gene expression data

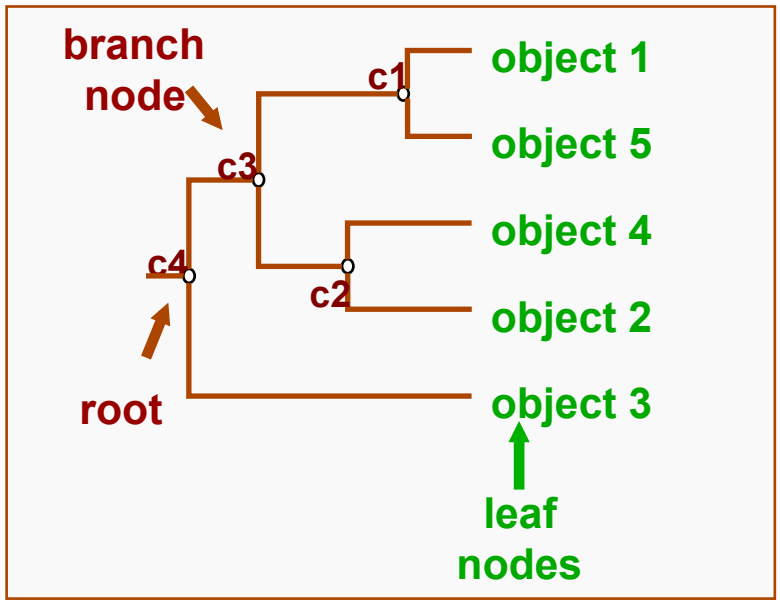


Principle of tree building

Distance matrix

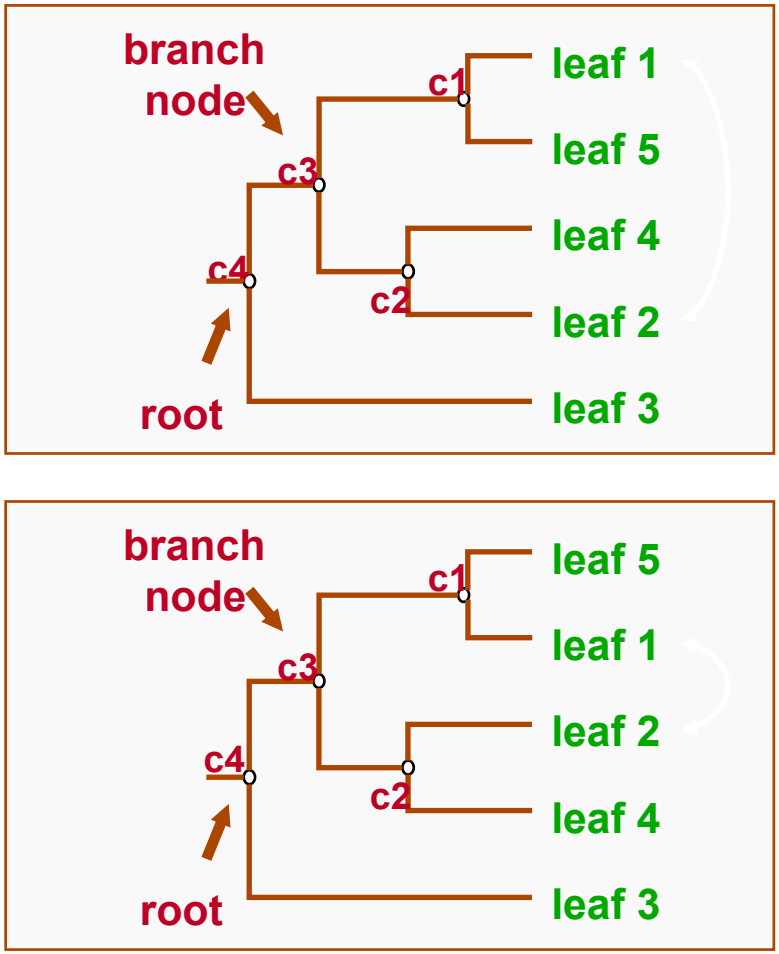
	object 1	object 2	object 3	object 4	object 5
object 1	0.00	4.00	6.00	3.50	1.00
object 2	4.00	0.00	6.00	2.00	4.50
object 3	6.00	6.00	0.00	5.50	6.50
object 4	3.50	2.00	5.50	0.00	4.00
object 5	1.00	4.50	6.50	4.00	0.00

Tree representation



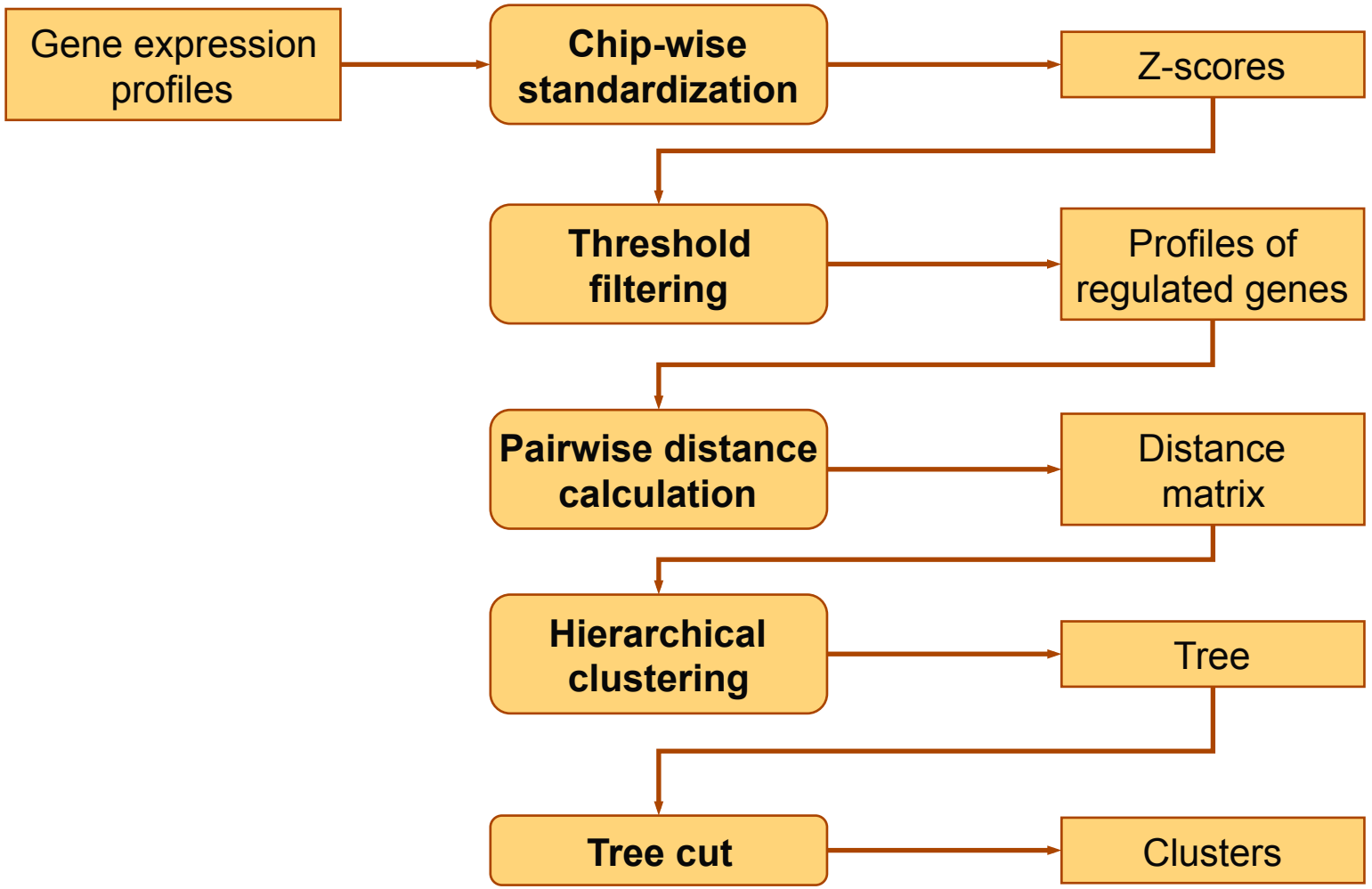
- Hierarchical clustering is an aggregative clustering method
  - takes as input a distance matrix
  - progressively regroups the closest objects/ groups
- One needs to define a (dis)similarity metric between two groups. There are several possibilities
  - **Average linkage**: the average distance between 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
- Algorithm
  - (1) Assign each object to a separate cluster.
  - (2) Find the pair of clusters with the shortest distance, and regroup them in a single cluster
  - (3) Repeat (2) until there is a single cluster
- The result is a tree, whose intermediate nodes represent clusters
  - N objects → N-1 intermediate nodes
- Branch lengths represent distances between clusters

Isomorphism on a tree



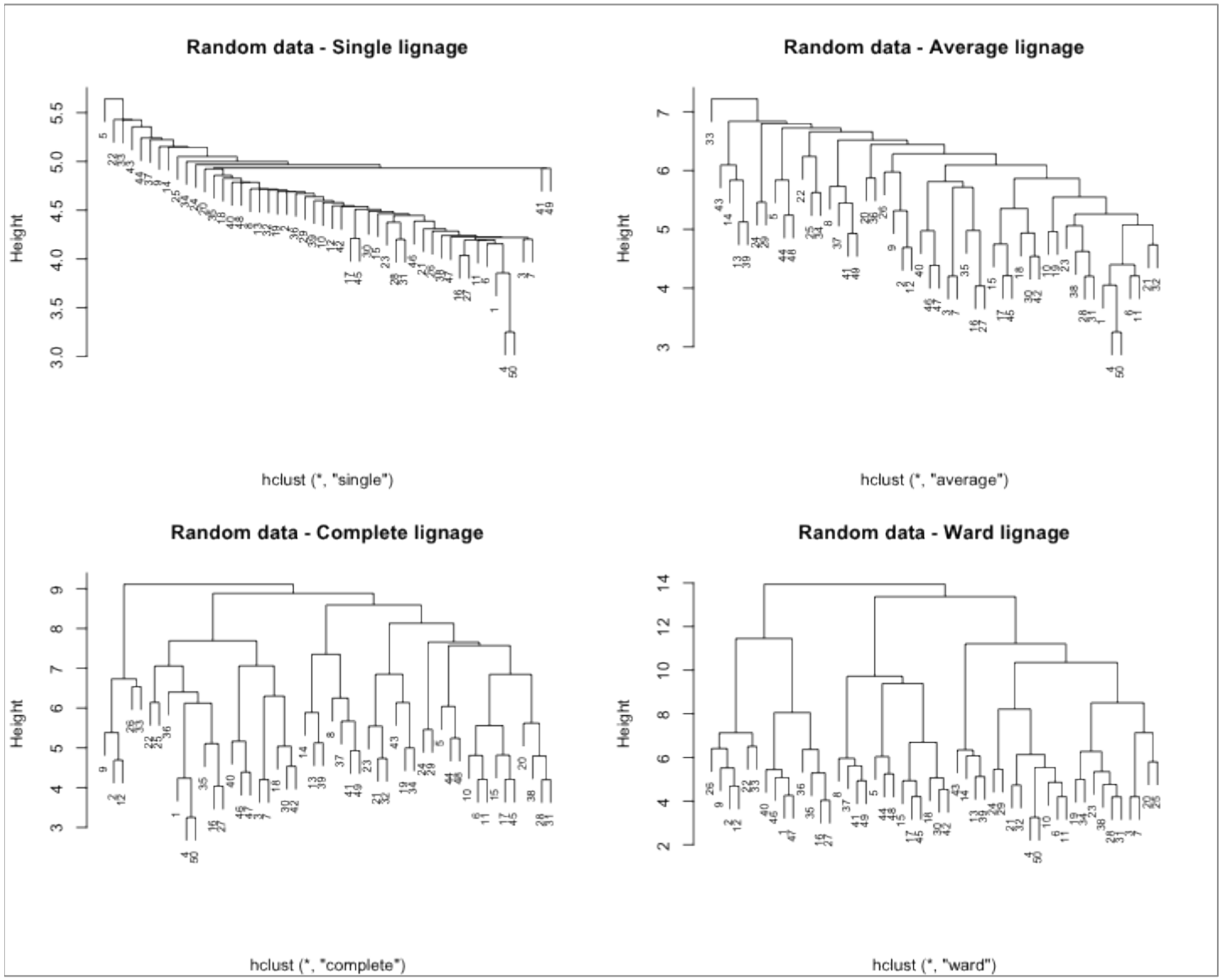
- In a tree, the two children of any branch node can be swapped. The result is an ***isomorphic tree***, considered as equivalent to the intial one.
- The two trees shown here are equivalent, however
  - Top tree: leaf 1 is far away from leaf 2
  - Bottom tree: leaf 1 is neighbour from leaf 2
- The vertical distance between two nodes does NOT reflect their actual distance !
- The distance between two nodes is the ***sum of branch lengths***.

Hierarchical clustering on gene expression data



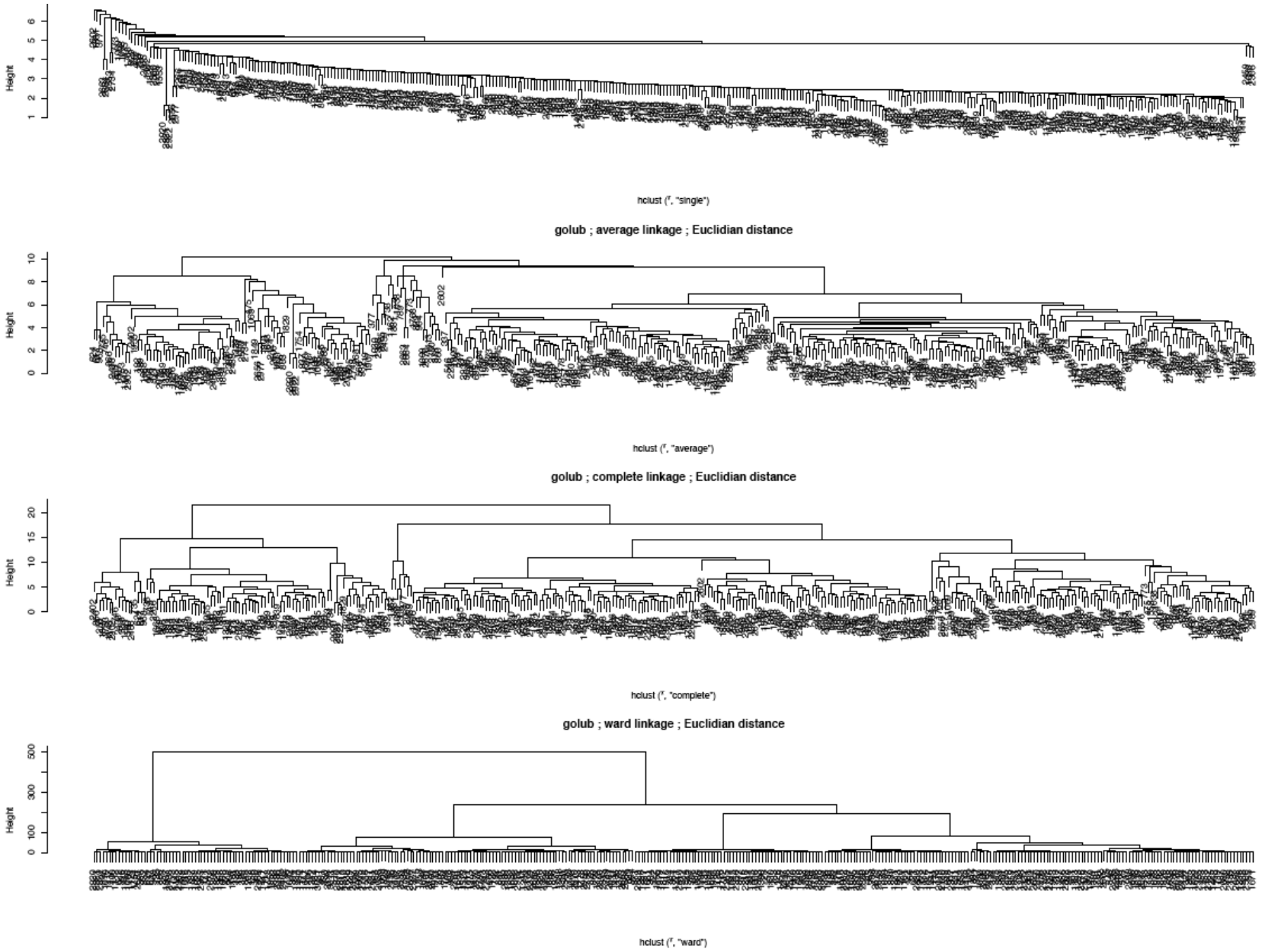
Impact of the agglomeration rule

- The choice of the agglomeration rule has a strong impact on the structure of a tree resulting from hierarchical clustering.

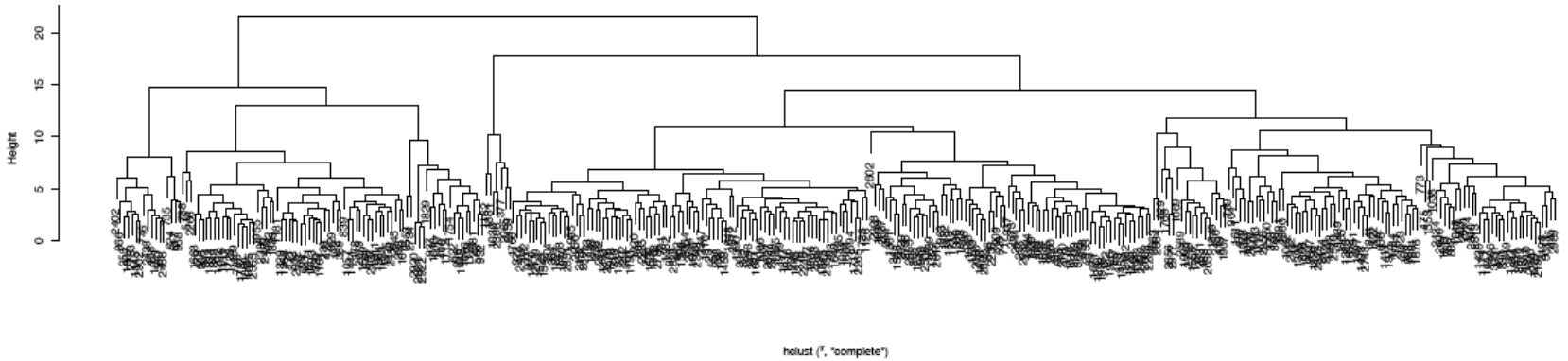


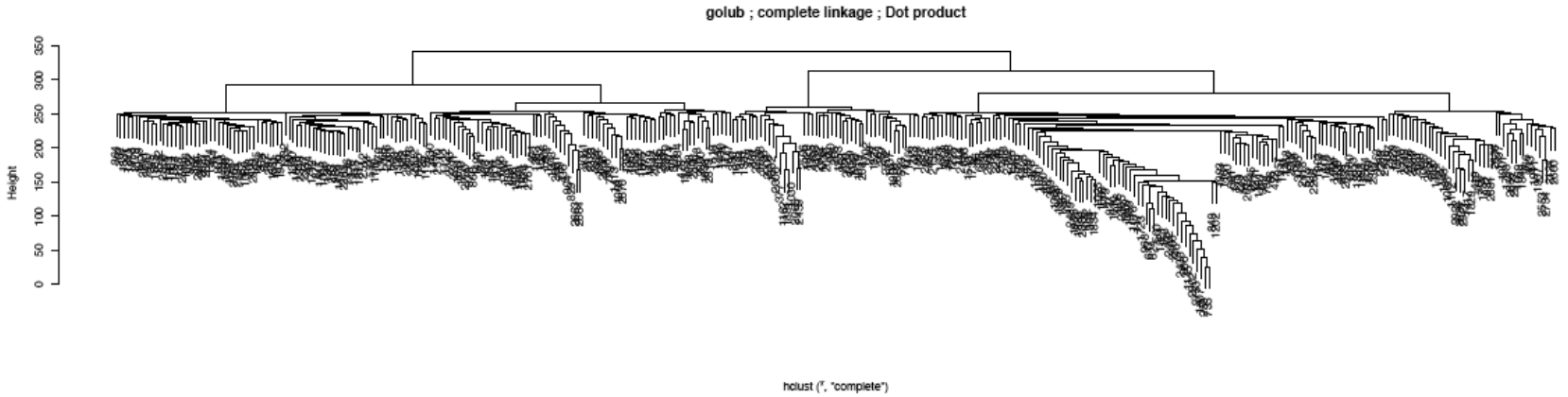
- Those four trees were built from the same distance matrix, using 4 different agglomeration rules.
- The clustering order is completely different.
- Single-linkage typically creates nesting clusters (“Matryoshka dolls”).
- Complete and Ward linkage create more balanced trees.
- **Note:** the matrix was computed from a matrix of random numbers. The subjective impression of structure are thus complete artifacts.

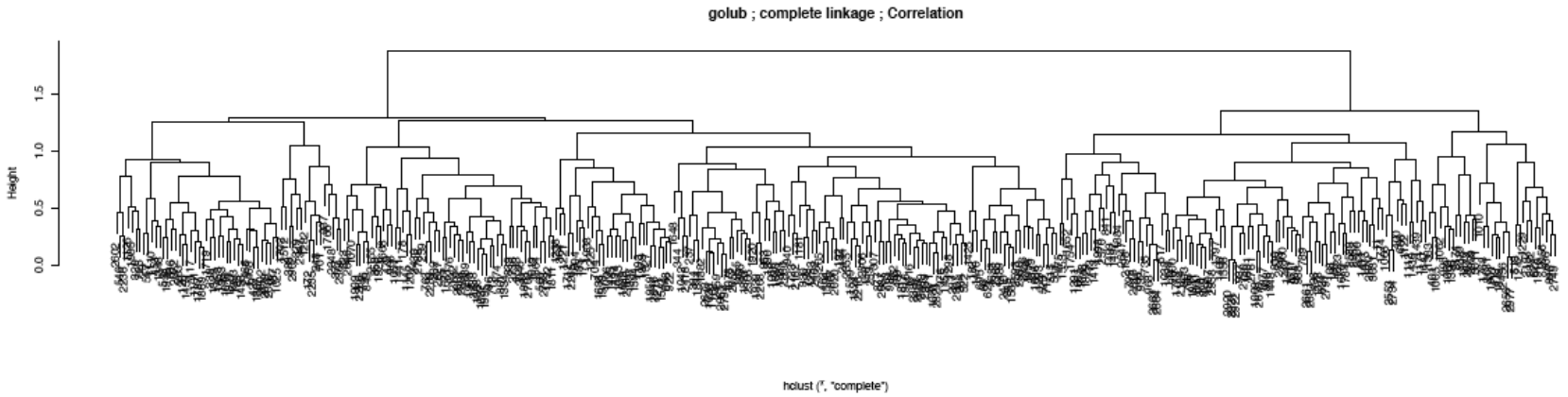
Golub 1999 - Impact of the linkage method (Euclidian distance for all the trees)



Golub 1999 - Effect of the distance metrics (complete linkage for all the trees)

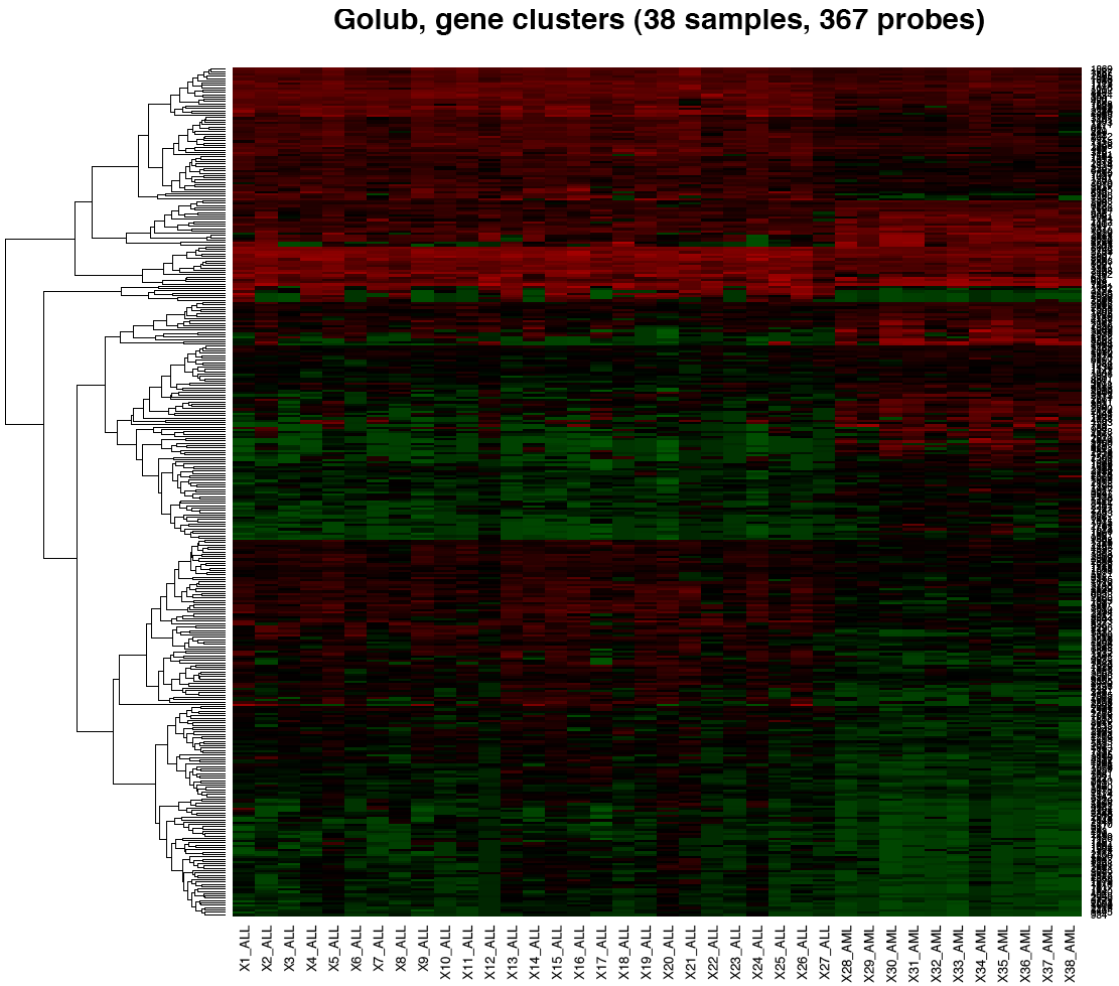




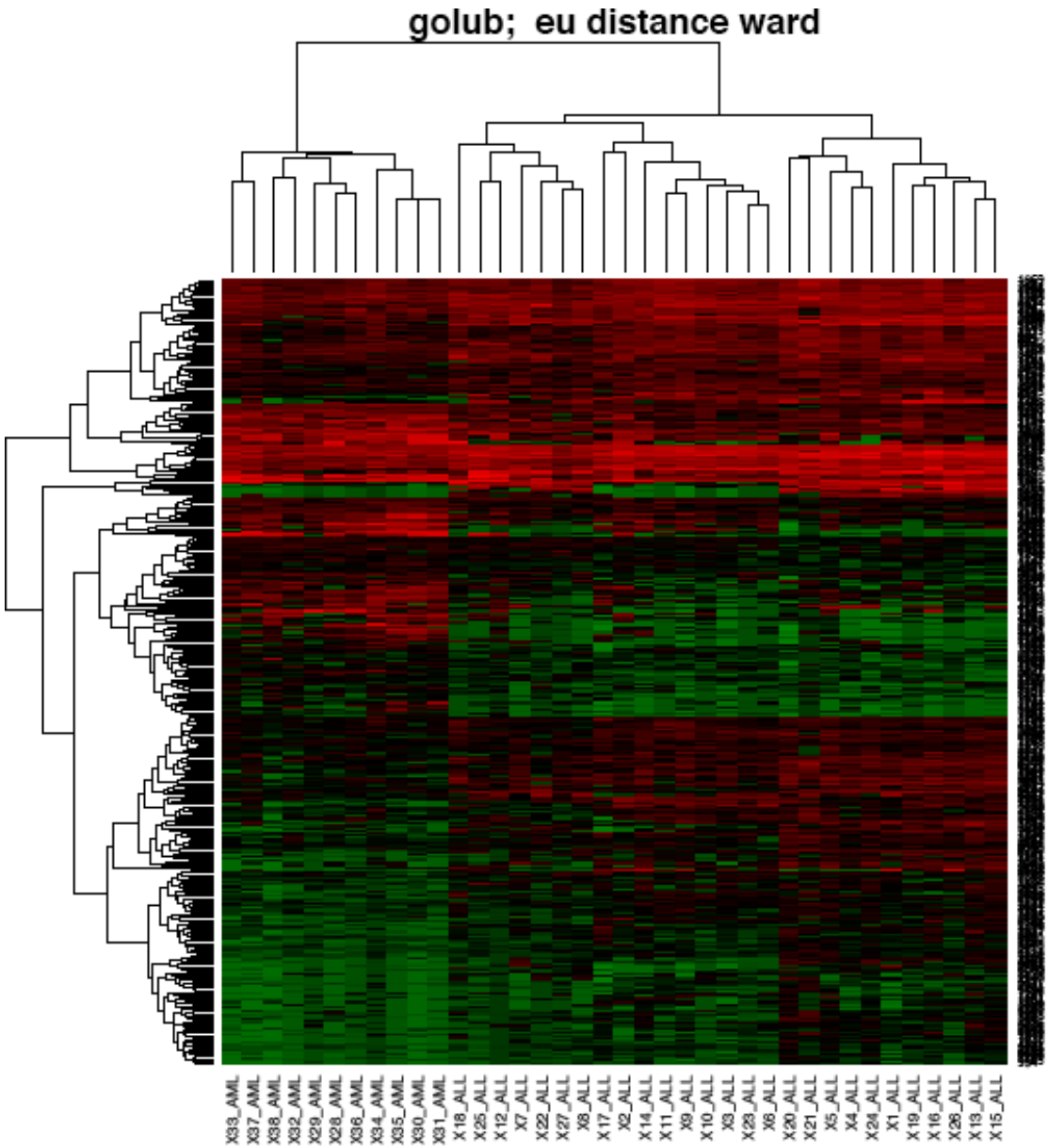


Golub 1999 - Gene clustering

- Gene clustering highlights groups of genes with similar expression profiles.

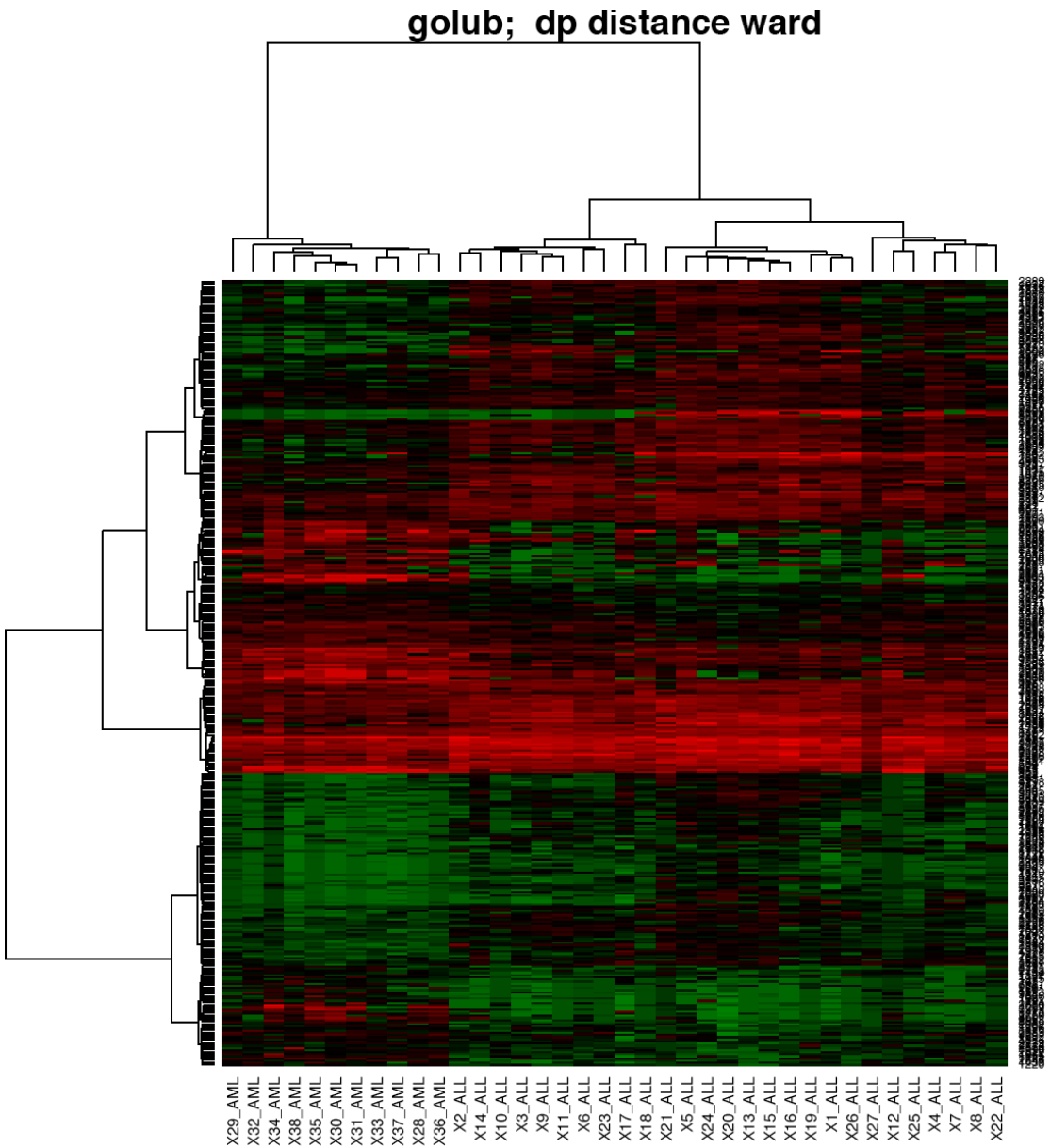


Golub 1999 - Ward Biclustering - Euclidian distance



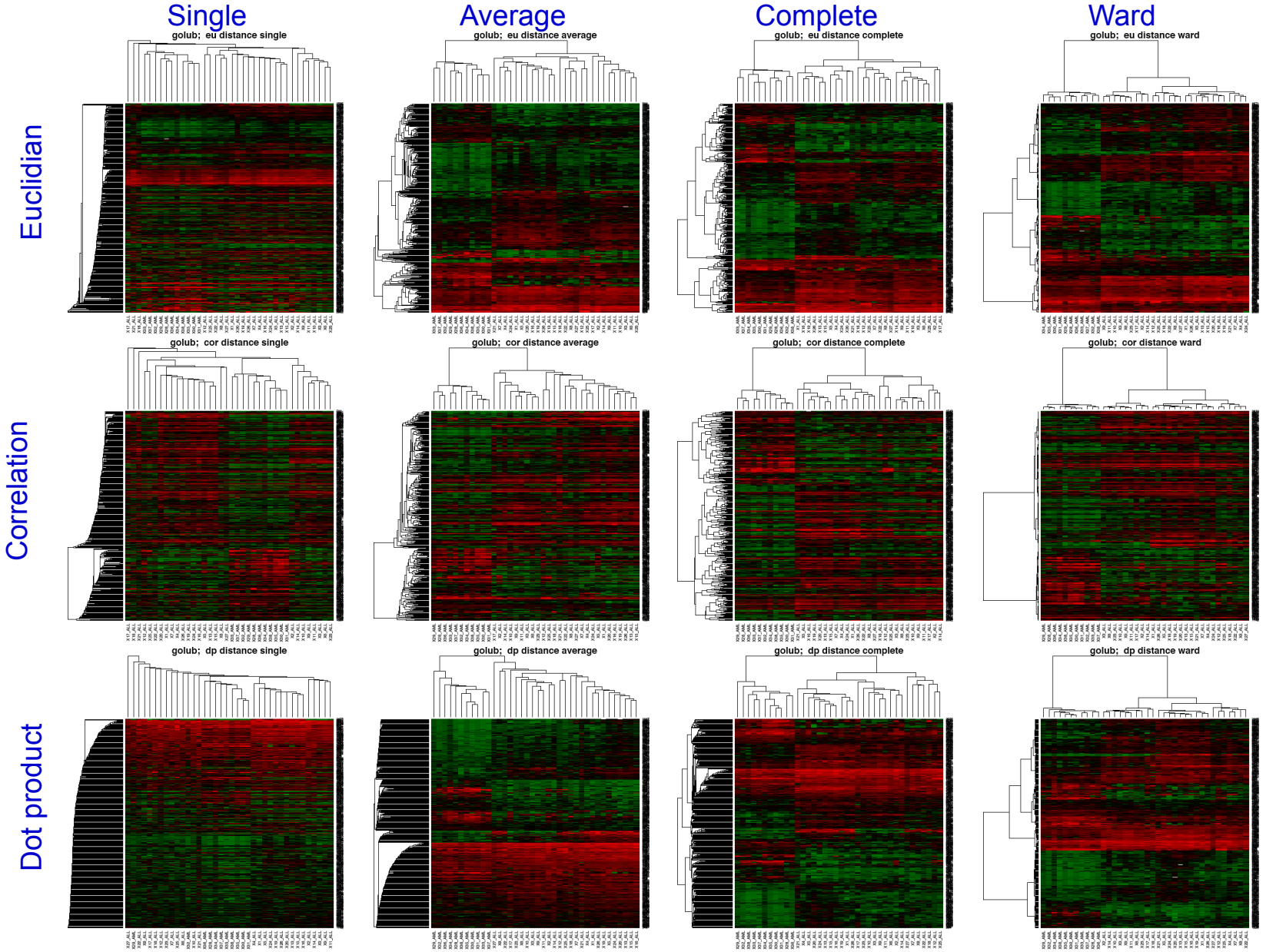
- Biclustering consists in clustering the rows (genes) and the columns (samples) of the data set.
- This reveals some subgroups of samples.
- With the golub 1999 data set
  - The AML and ALL patients are clearly separated at the top level of the tree
  - There are apparently two clusters among the AML samples.

Golub 1999 - Ward Biclustering - Dot product distance

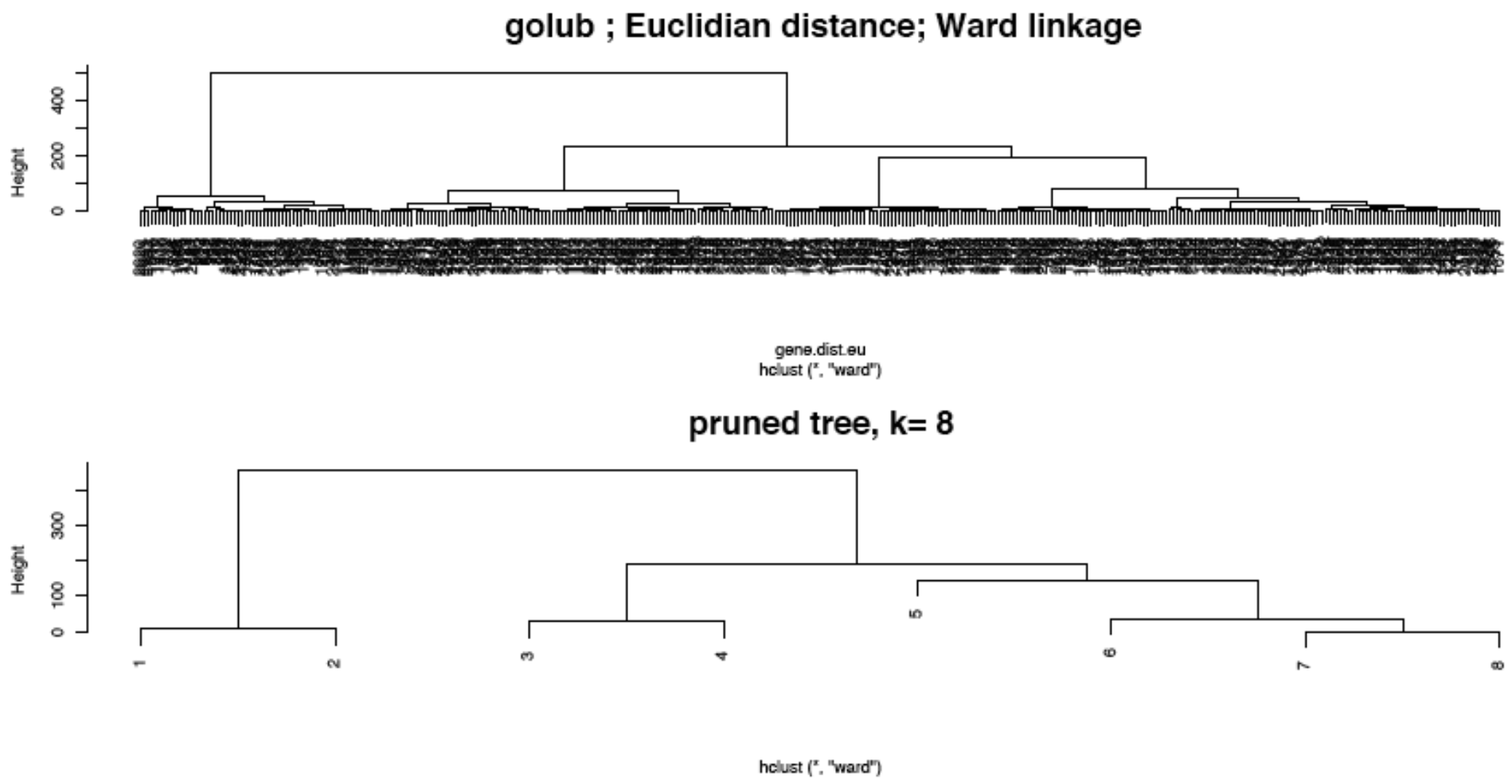


- Biclustering consists in clustering the rows (genes) and the columns (samples) of the data set.
- This reveals some subgroups of samples.
- With the golub 1999 data set
  - The AML and ALL patients are clearly separated at the top level of the tree
  - There are apparently two clusters among the ALL samples. Actually these two clusters correspond to distinct cell subtypes: T and B cells, respectively.

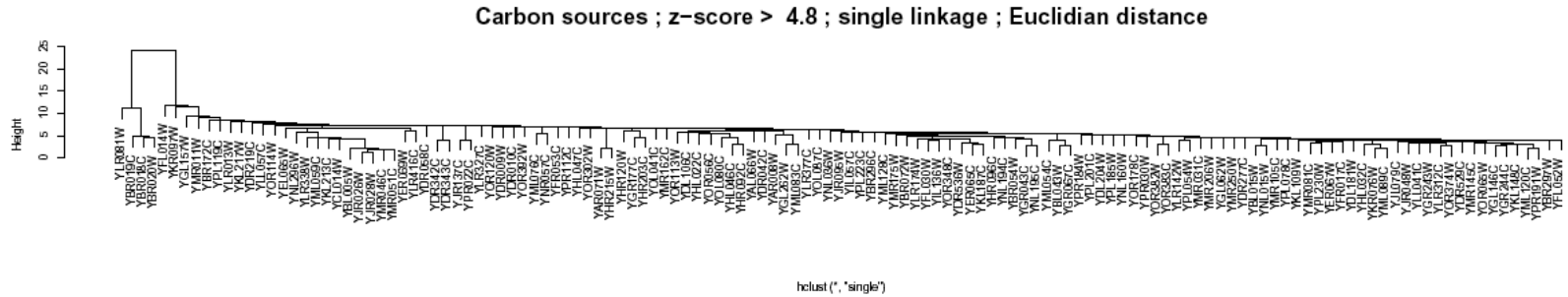
Impact of distance metrics and agglomeration rules

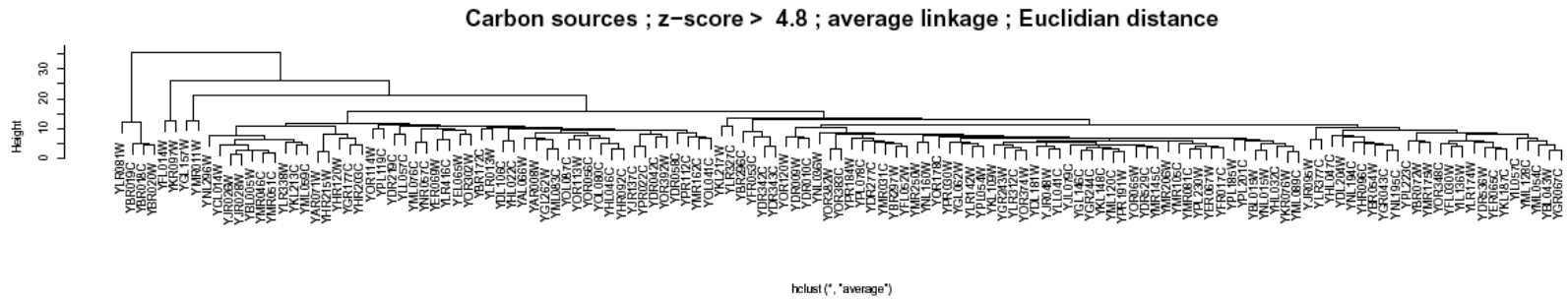


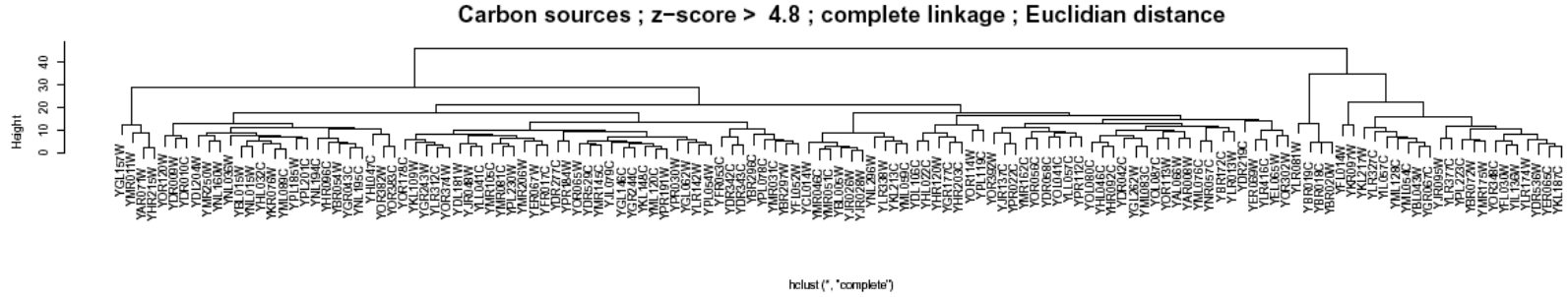
Golub 1999 - Pruning the tree

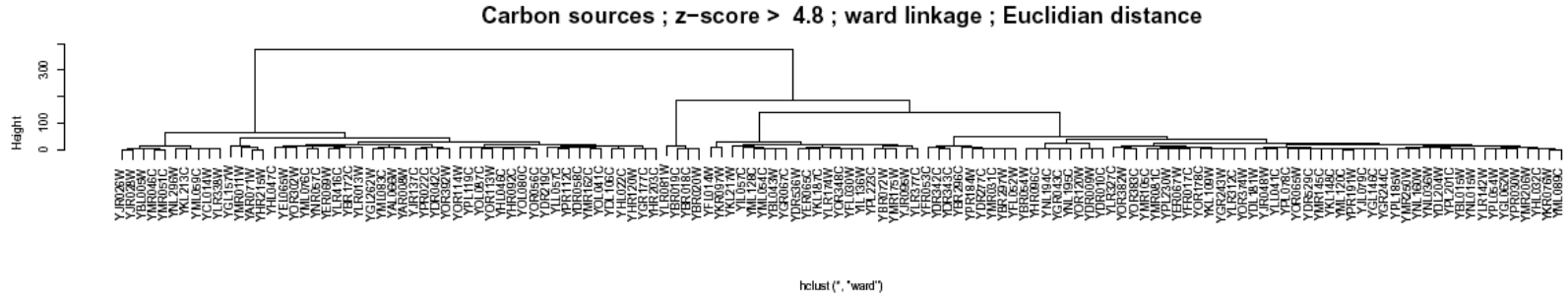


Impact of the linkage method

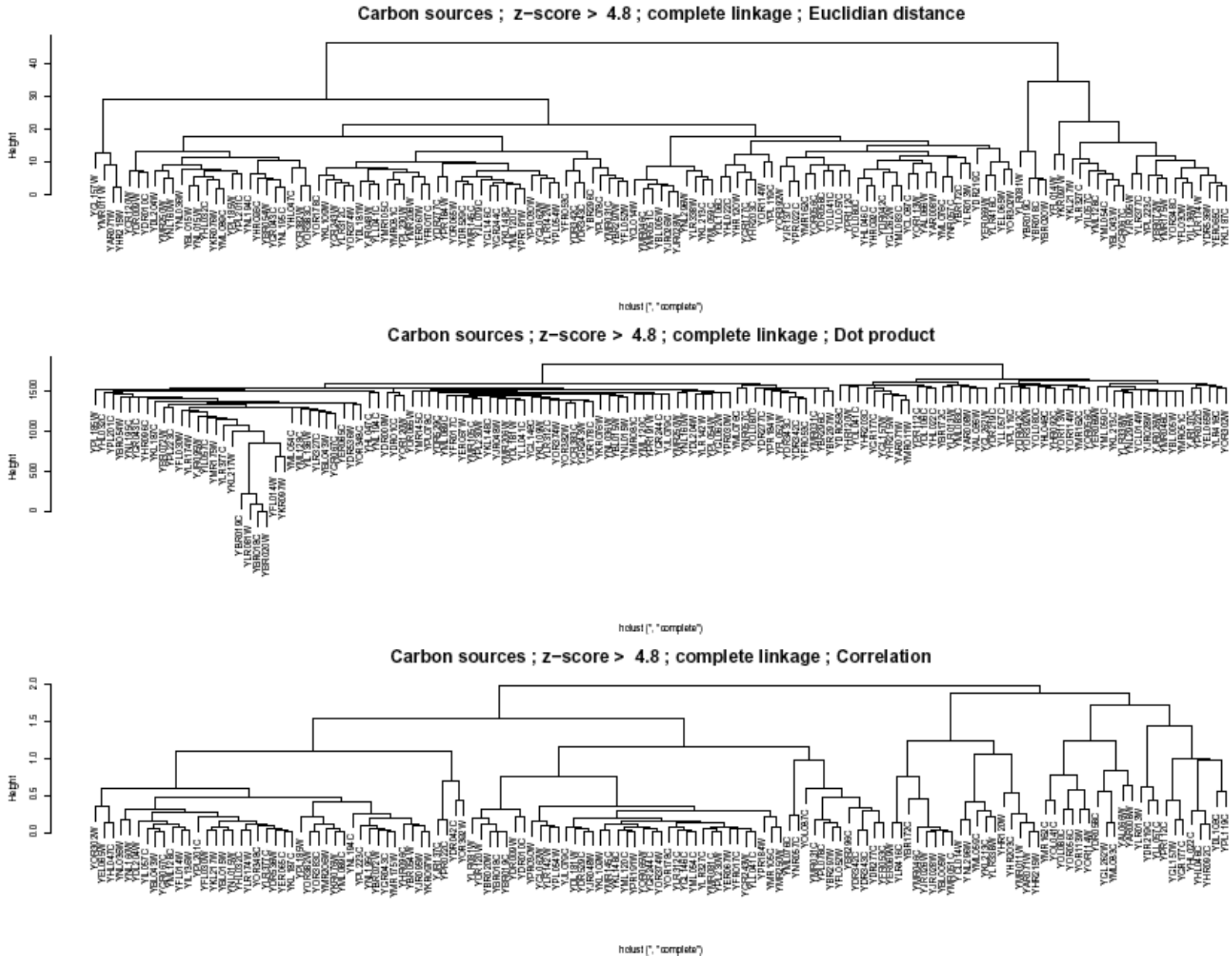




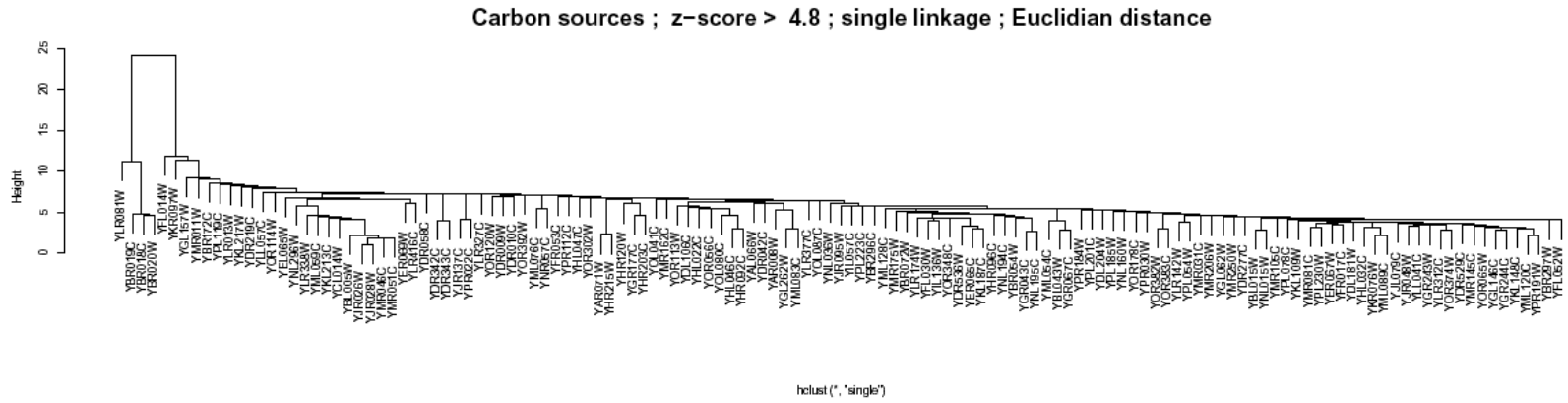


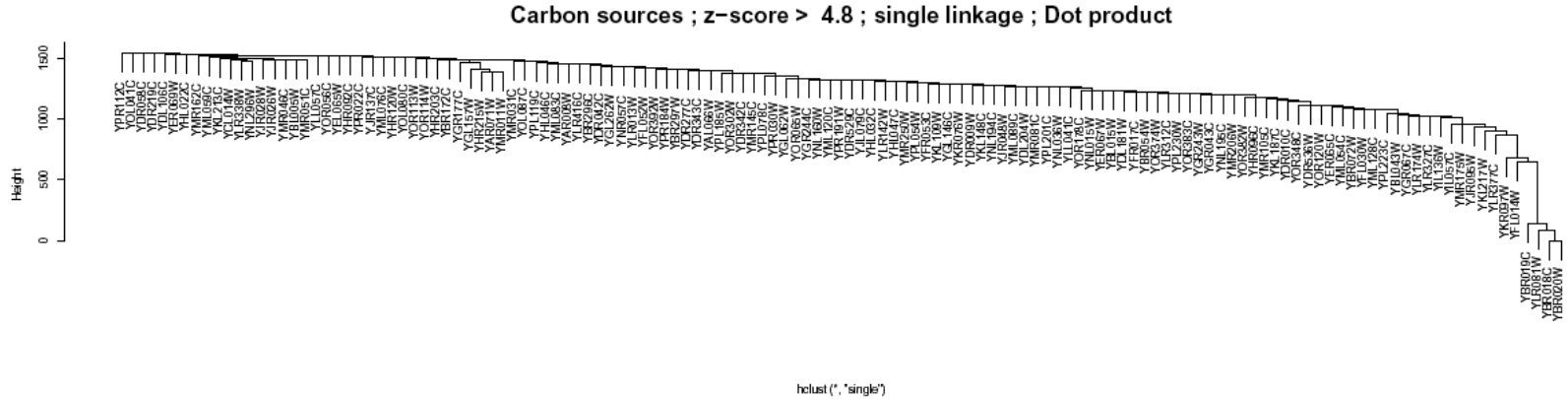


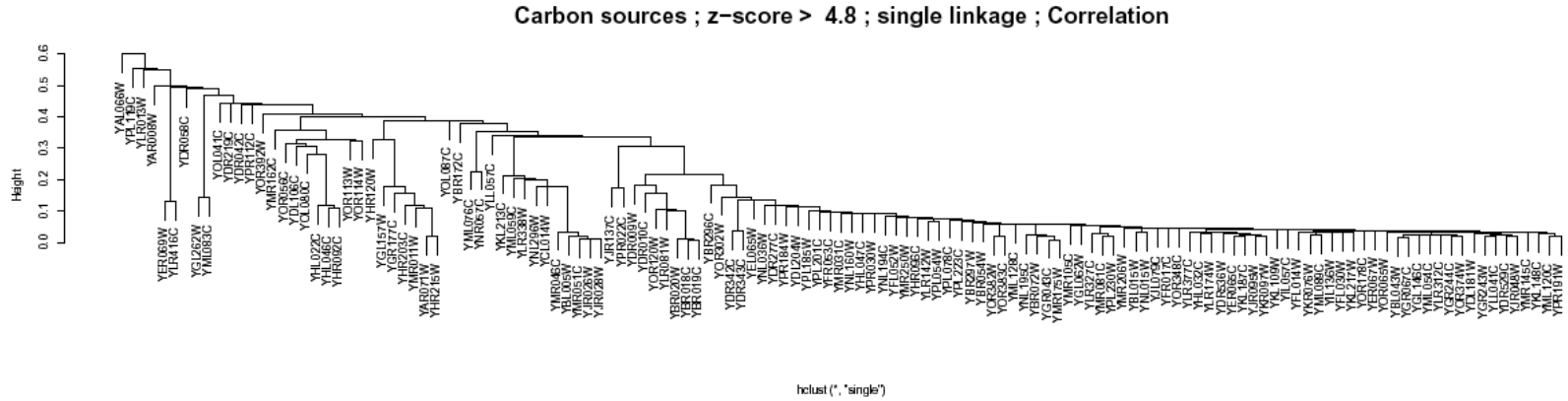
Impact of the distance metric - complete linkage

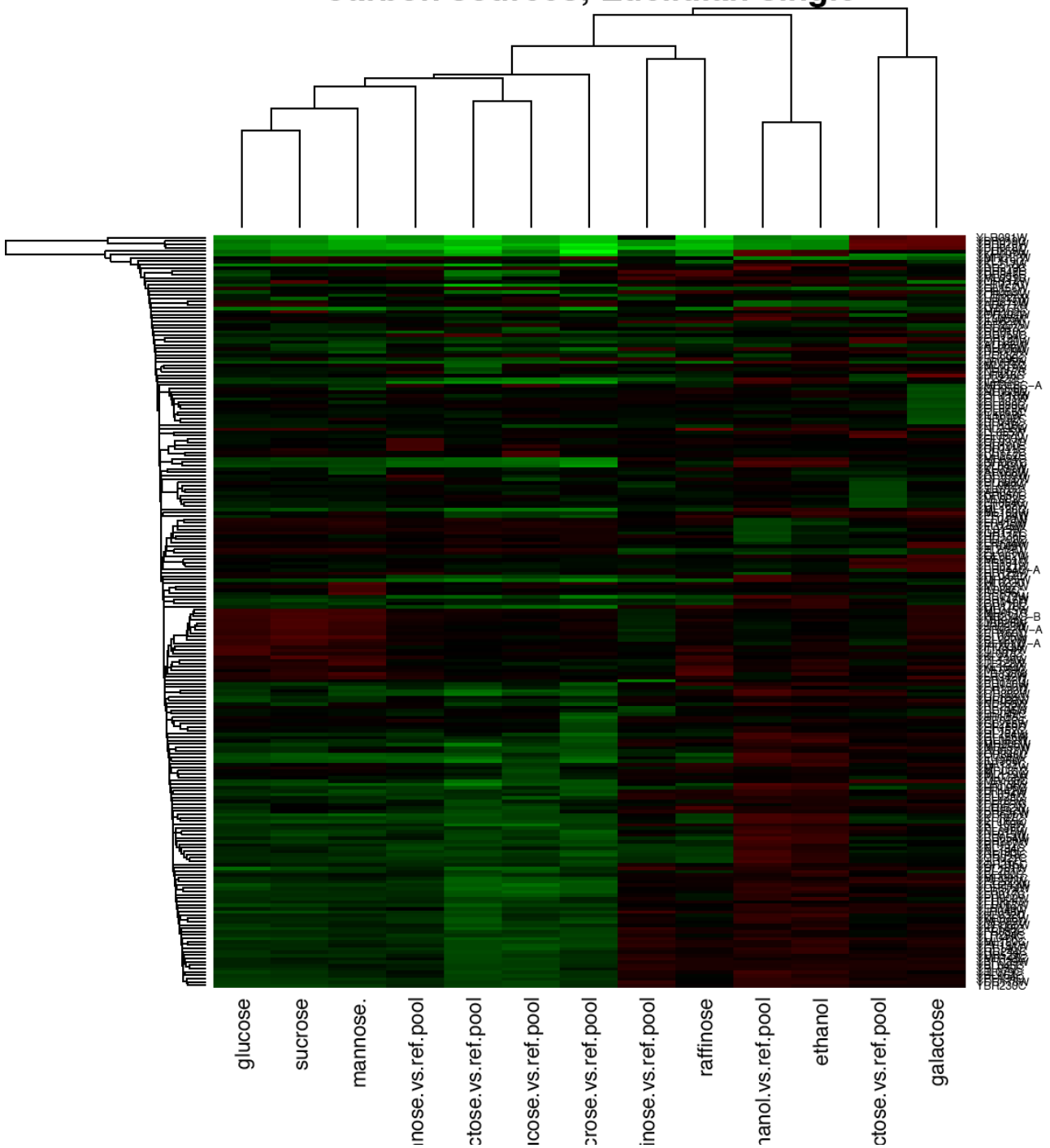


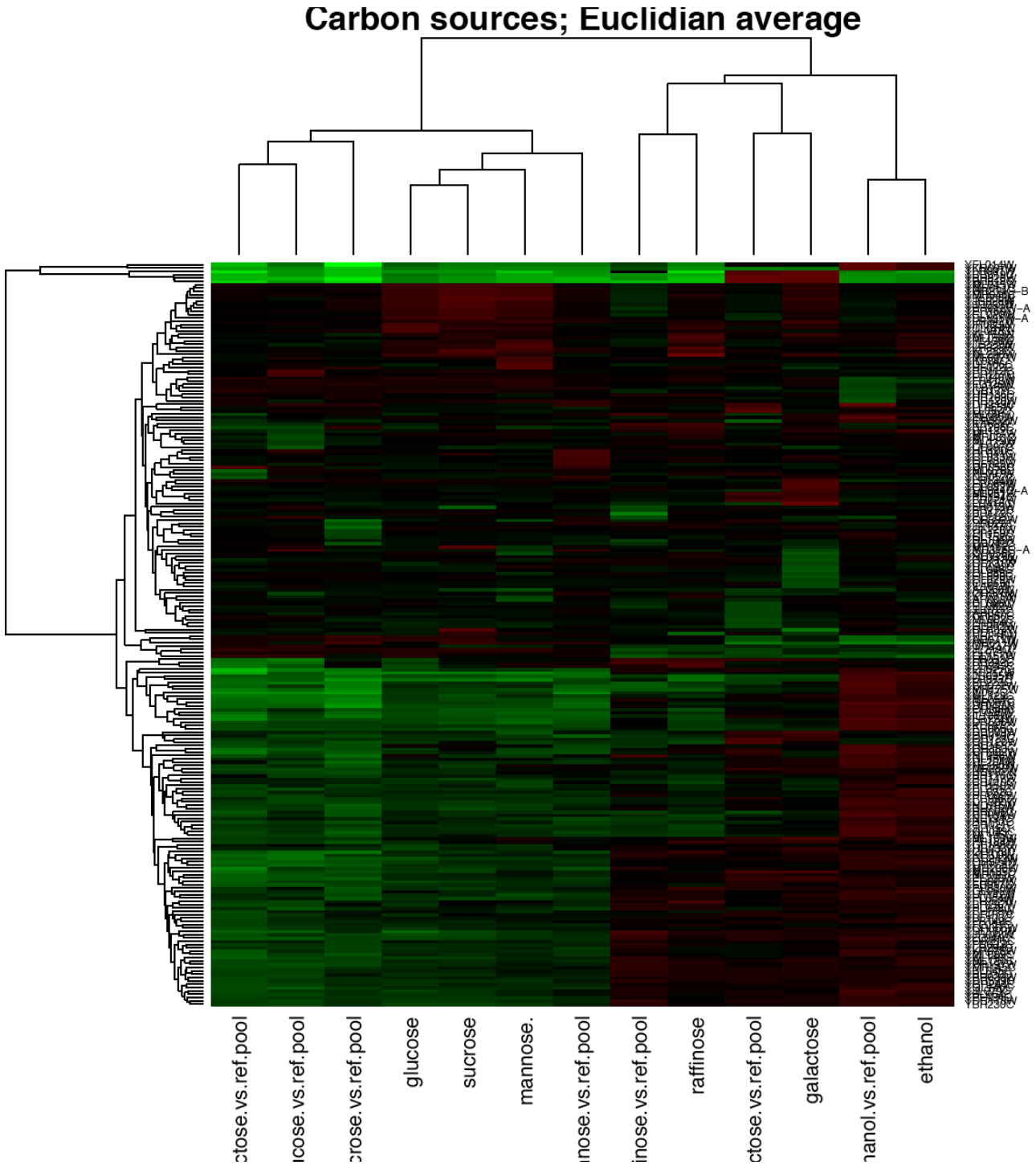
Ipact of the distance metric - single linkage

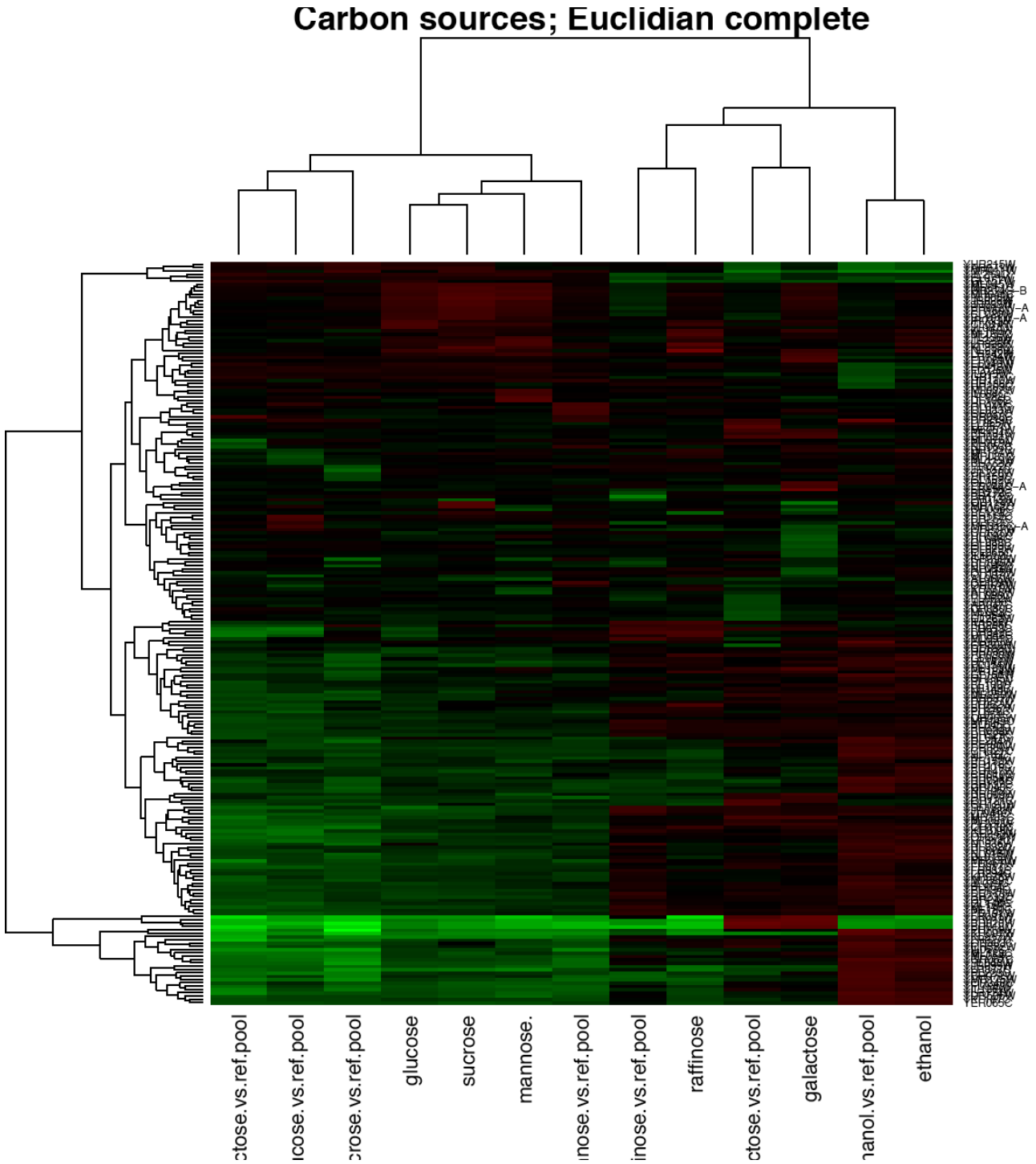


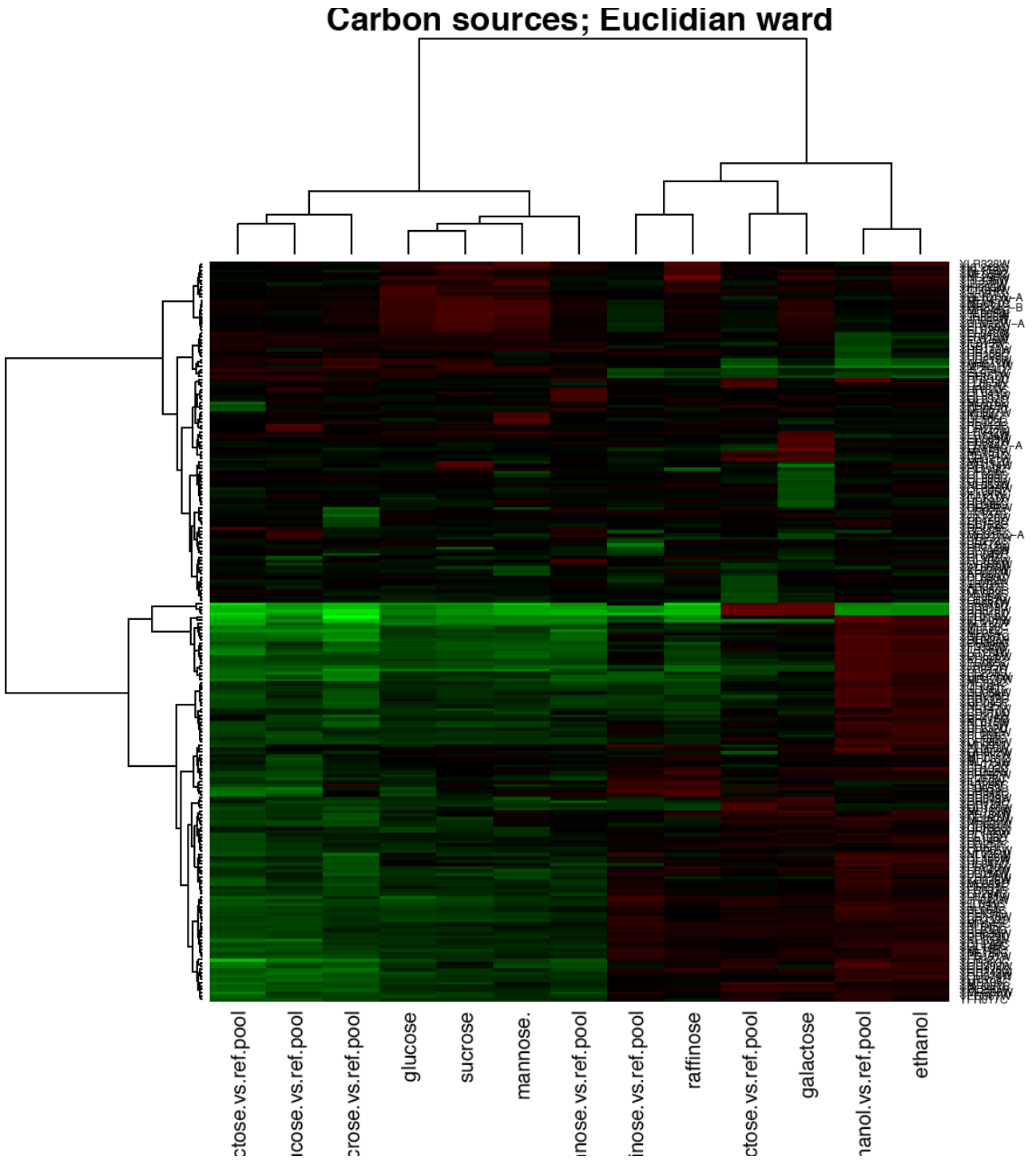





Carbon sources; Euclidian single
glucose
sucrose
mannose.
nose.vs.ref.pool
ctose.vs.ref.pool
cose.vs.ref.pool
rose.vs.ref.pool
inose.vs.ref.pool
raffinose
hanol.vs.ref.pool
ethanol
ctose.vs.ref.pool
galactose

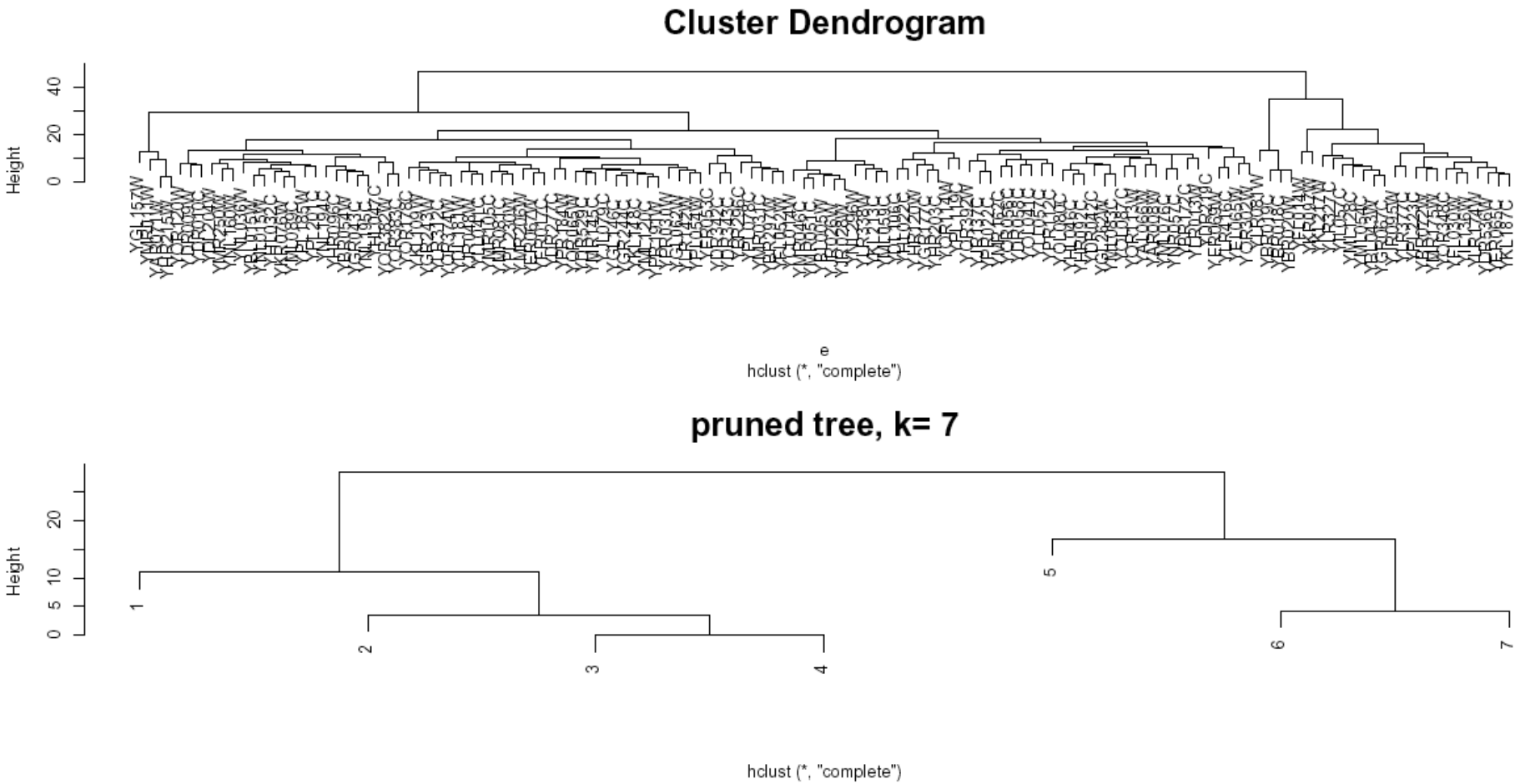

Carbon sources; Euclidian average
ctose.vs.ref.pool
lcose.vs.ref.pool
crose.vs.ref.pool
glucose
sucrose
mannose.
inose.vs.ref.pool
inose.vs.ref.pool
raffinose
ctose.vs.ref.pool
galactose
hanol.vs.ref.pool
ethanol


Carbon sources; Euclidian complete
ctose.vs.ref.pool
lcose.vs.ref.pool
crose.vs.ref.pool
glucose
sucrose
mannose.
inose.vs.ref.pool
inose.vs.ref.pool
raffinose
ctose.vs.ref.pool
galactose
hanol.vs.ref.pool
ethanol

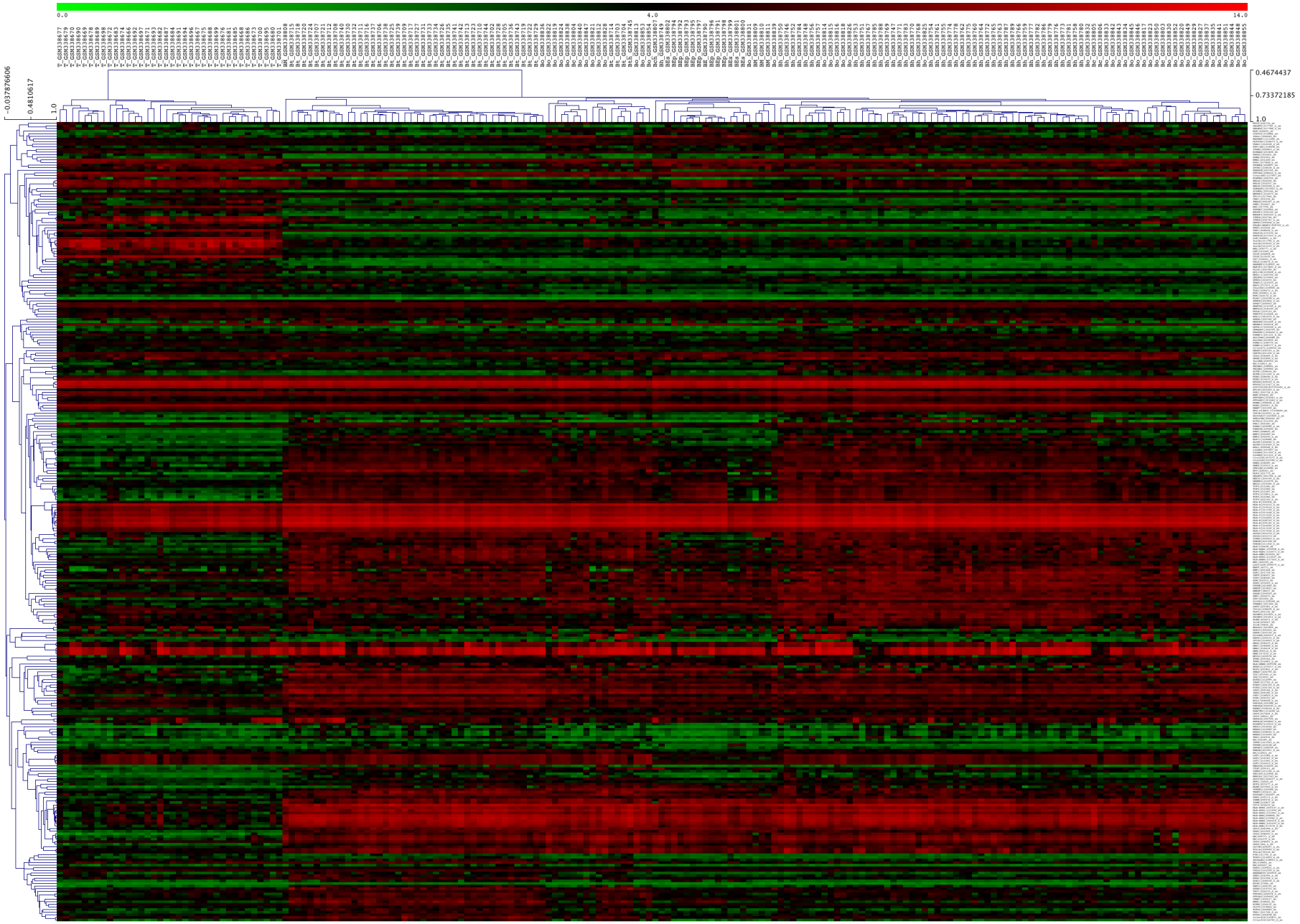

Carbon sources; Euclidian ward
ctose.vs.ref.pool
ıcose.vs.ref.pool
crose.vs.ref.pool
glucose
sucrose
mannose.
ınose.vs.ref.pool
inose.vs.ref.pool
raffinose
ctose.vs.ref.pool
galactose
ıanol.vs.ref.pool
ethanol

Pruning and cutting the tree

- The tree can be cut at level k (starting from the root), which creates *k* clusters
- A k-group partitioning is obtained by collecting the leaves below each branch of the pruned tree



Den Boer 2009 – Hierarchical clustering

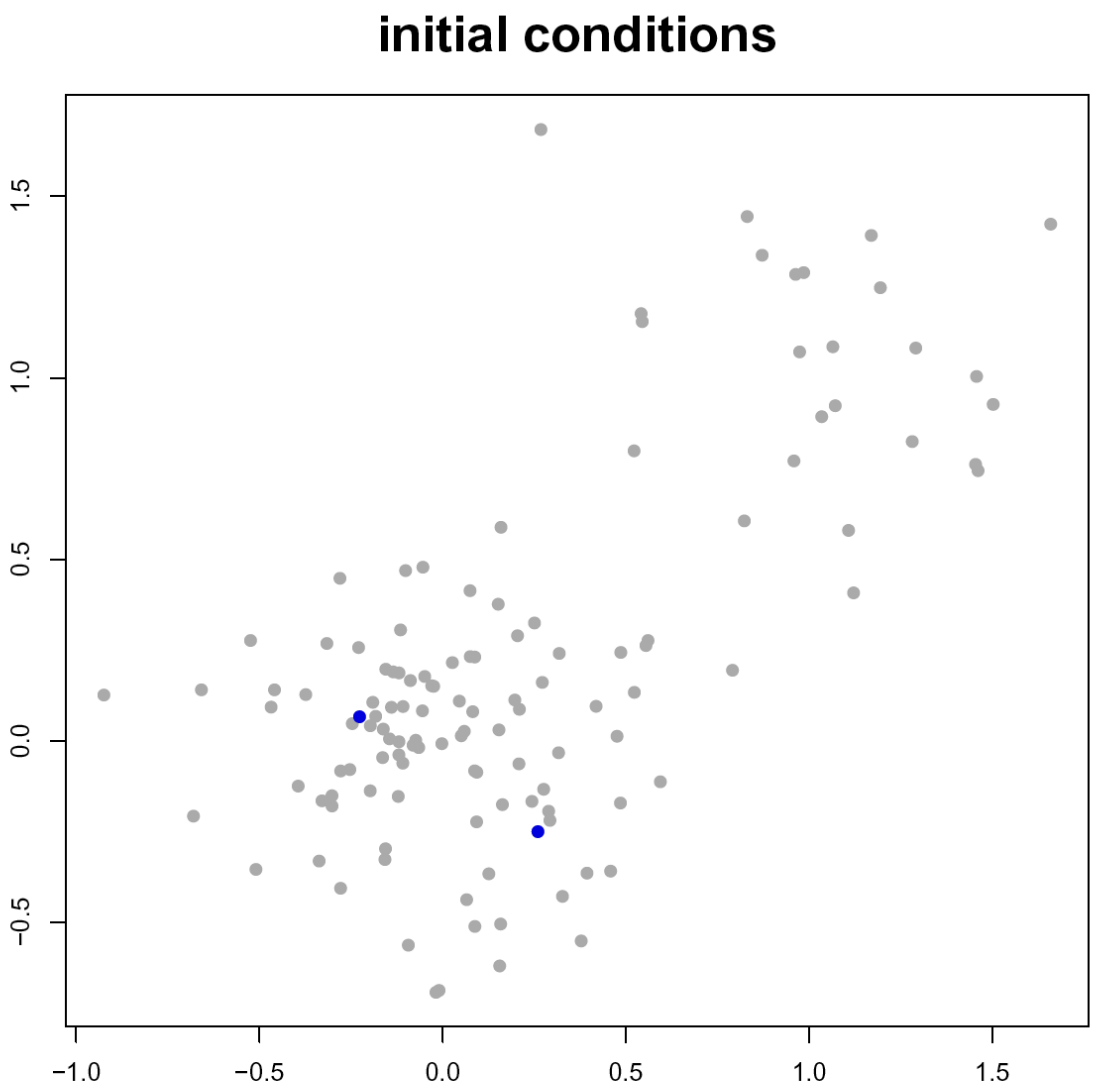


K-means clustering

Clustering around mobile centres

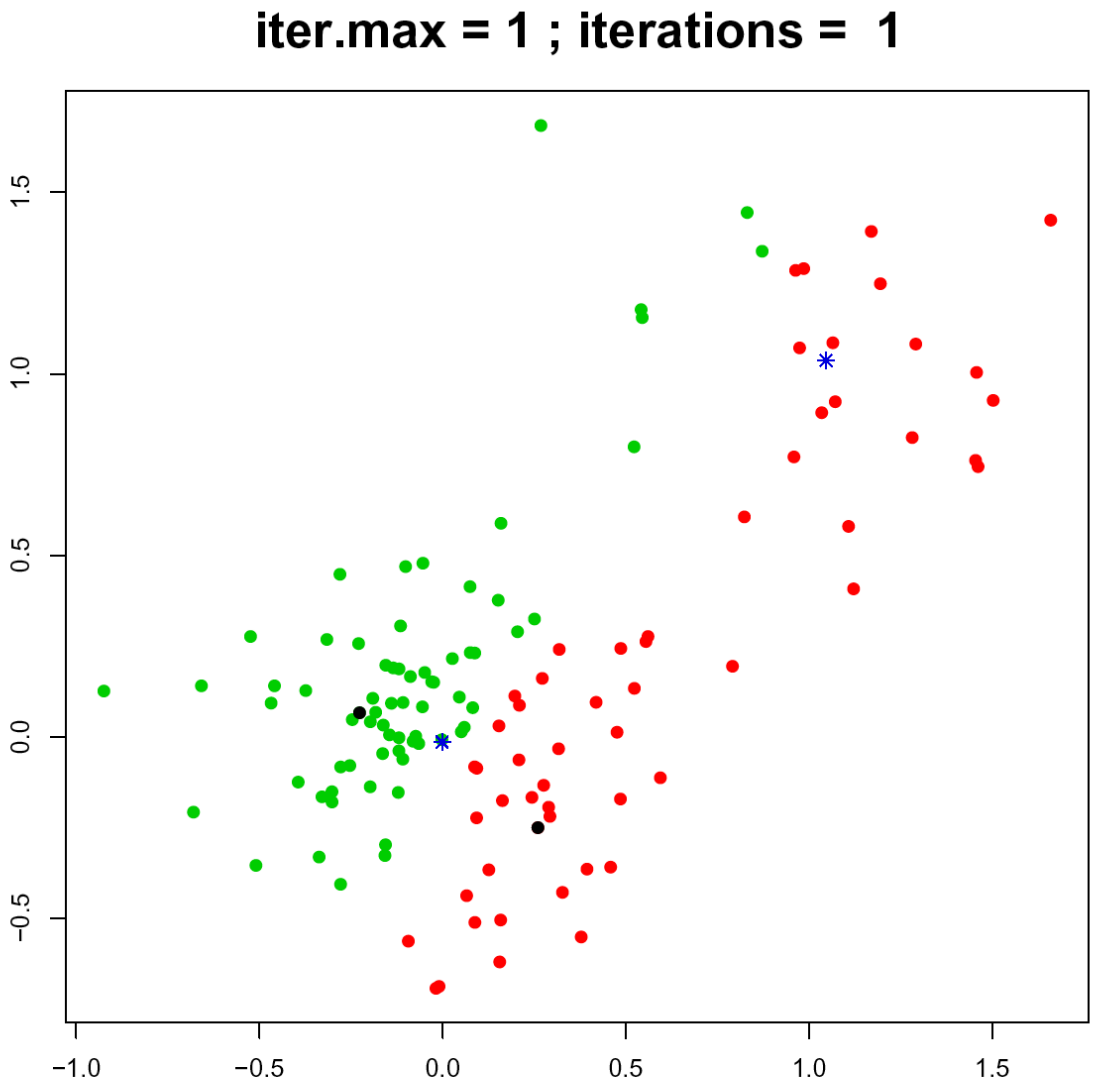
- The number of centres (k) has to be specified a priori
- Algorithm
  - (1) Arbitrarily select k initial centres
  - (2) Assign each element to the closest centre
  - (3) Re-calculate centres (mean position of the assigned elements)
  - (4) Repeat (2) and (3) until one of the stopping conditions is reached
    - the clusters are the same as in the previous iteration
    - the difference between two iterations is smaller than a specified threshold
    - the max number of iterations has been reached

Mobile centres example - initial conditions



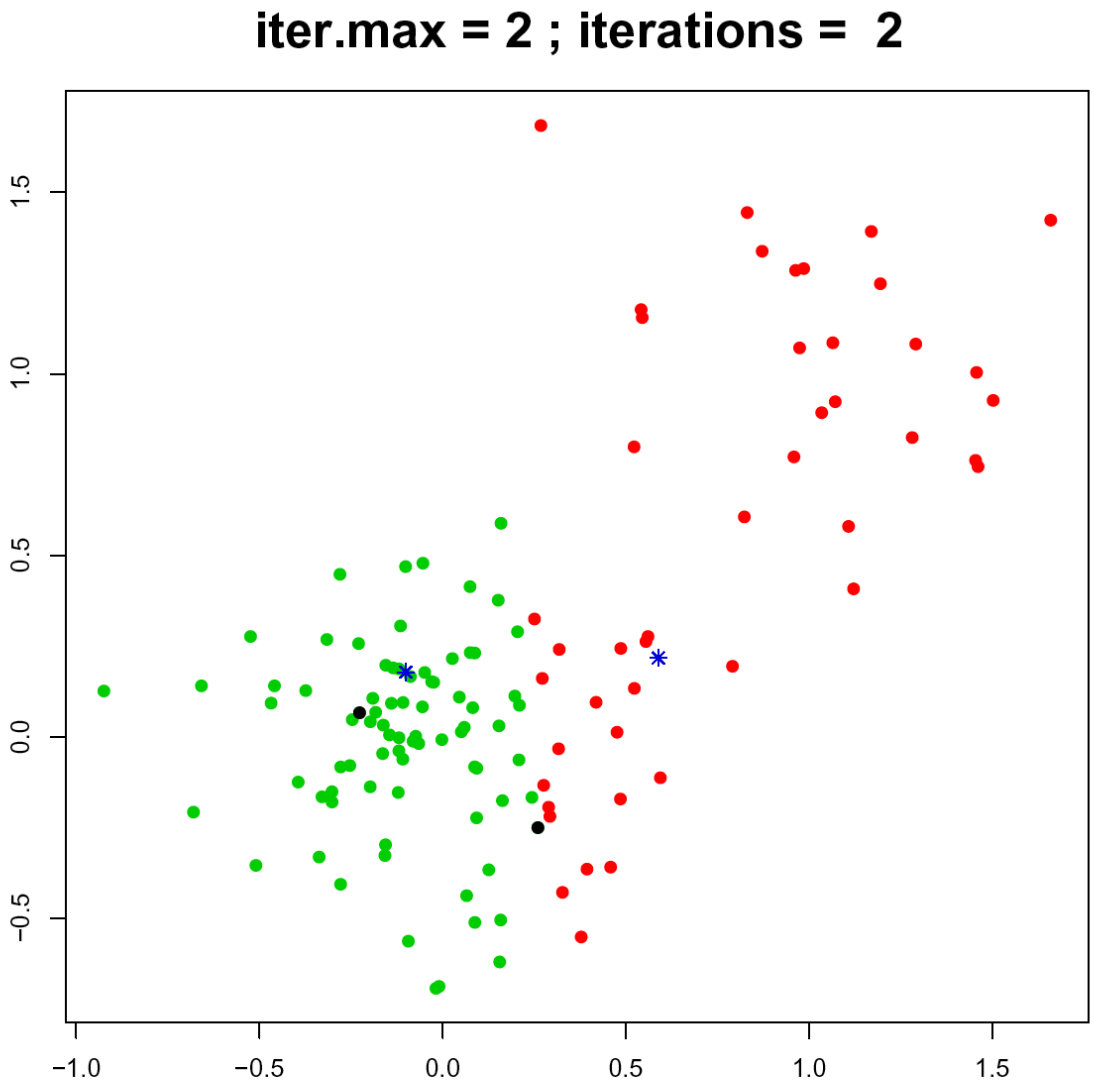
- Two sets of random points are randomly generated
  - 200 points centred on (0,0)
  - 50 points centred on (1,1)
- Two points are randomly chosen as seeds (blue dots)

Mobile centres example - first iteration



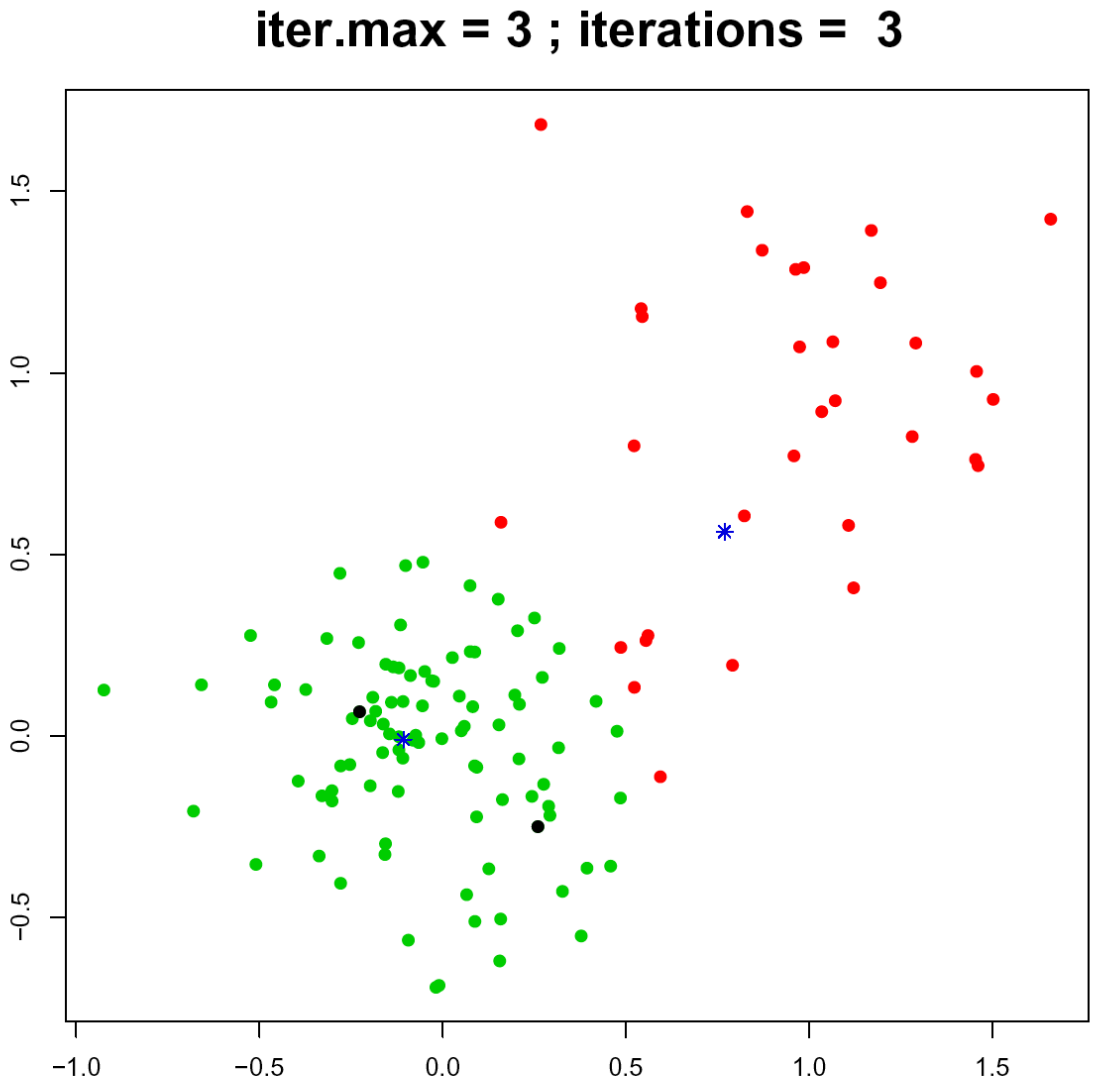
- Step 1
  - Each dot is assigned to the cluster with the closest centre
  - Centres are re-calculated (blue star) on the basis of the new clusters

Mobile centres example - second iteration



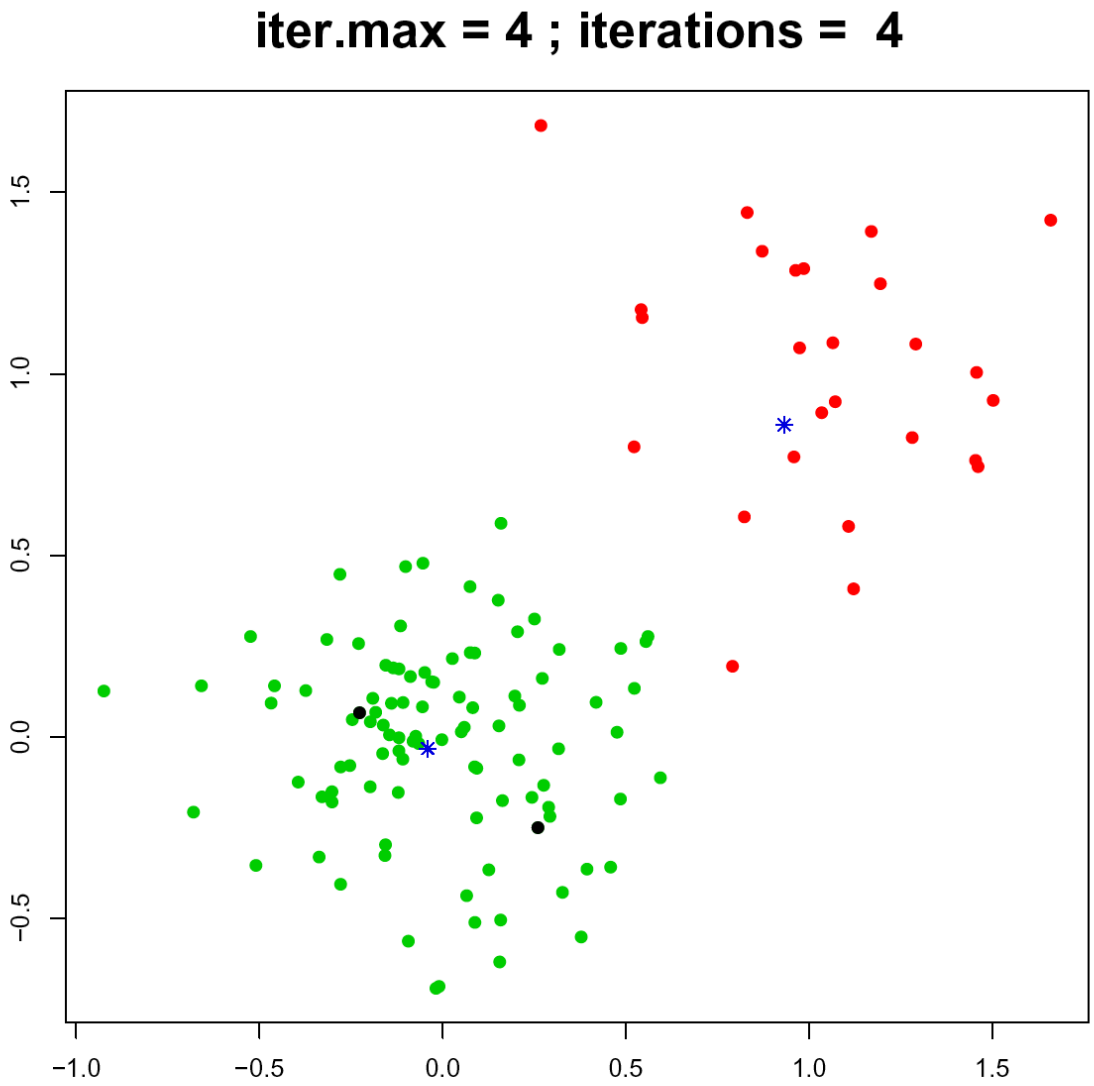
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

Mobile centres example - after 3 iterations



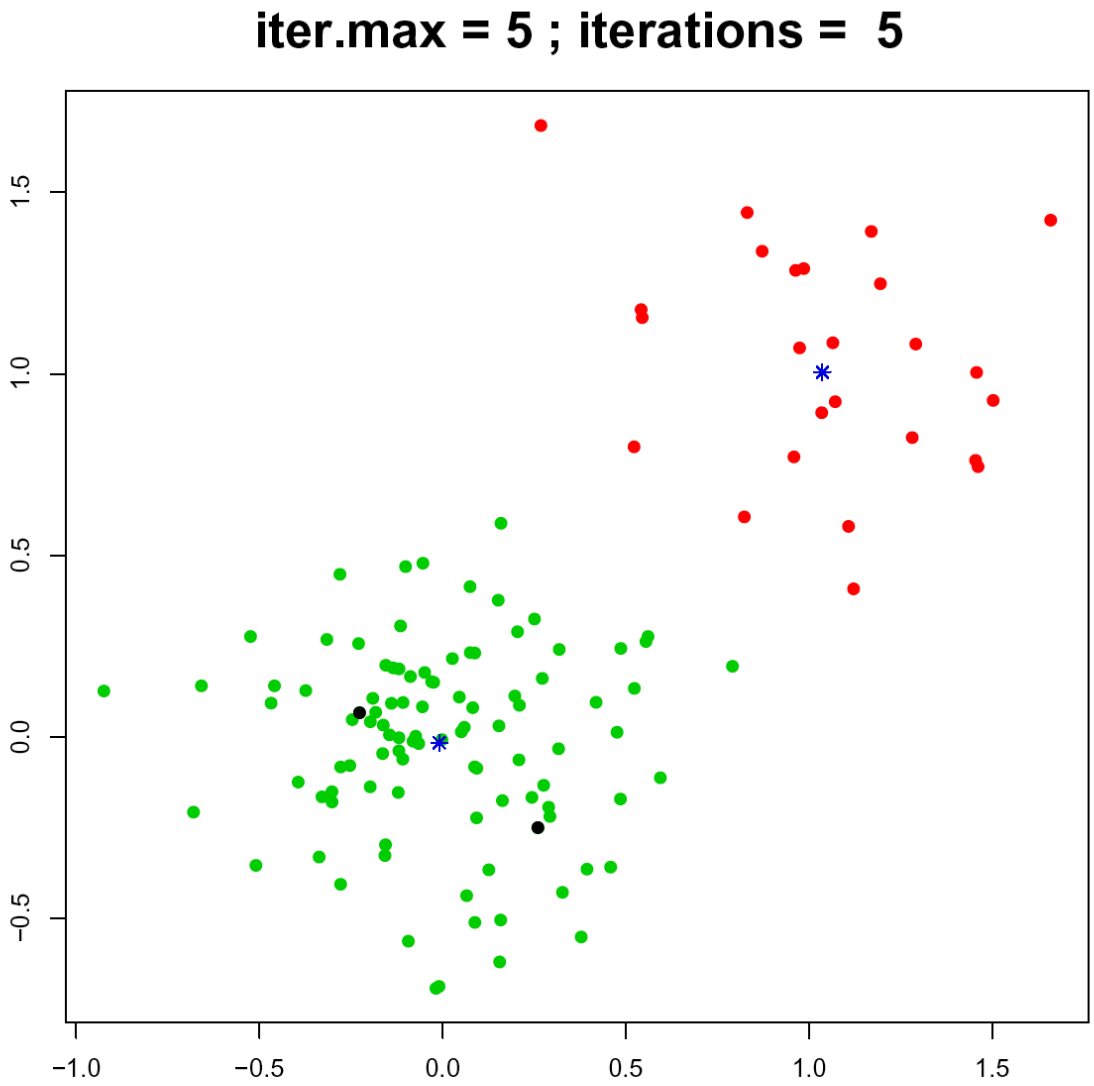
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

Mobile centres example - after 4 iterations



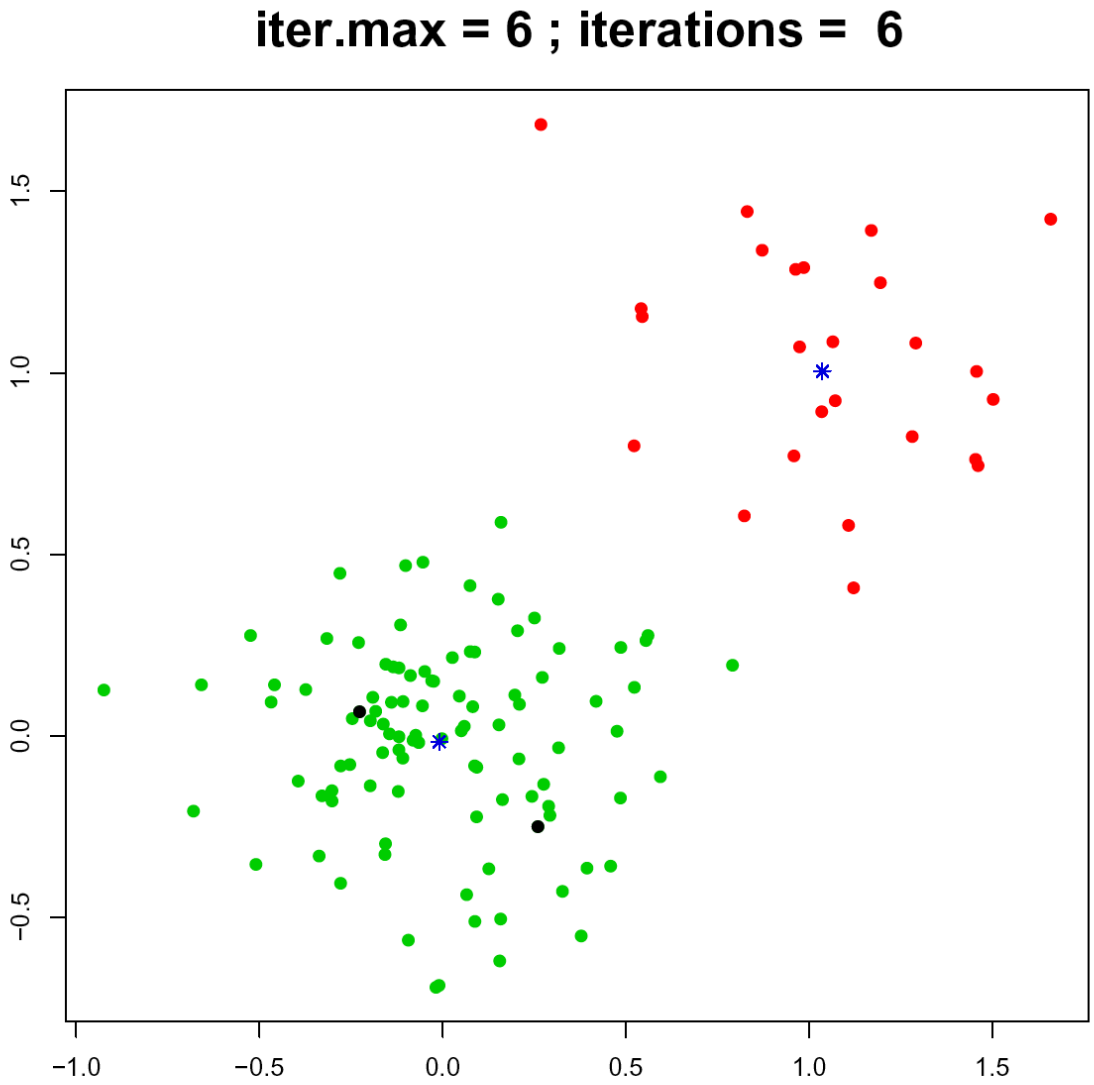
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

Mobile centres example - after 5 iterations



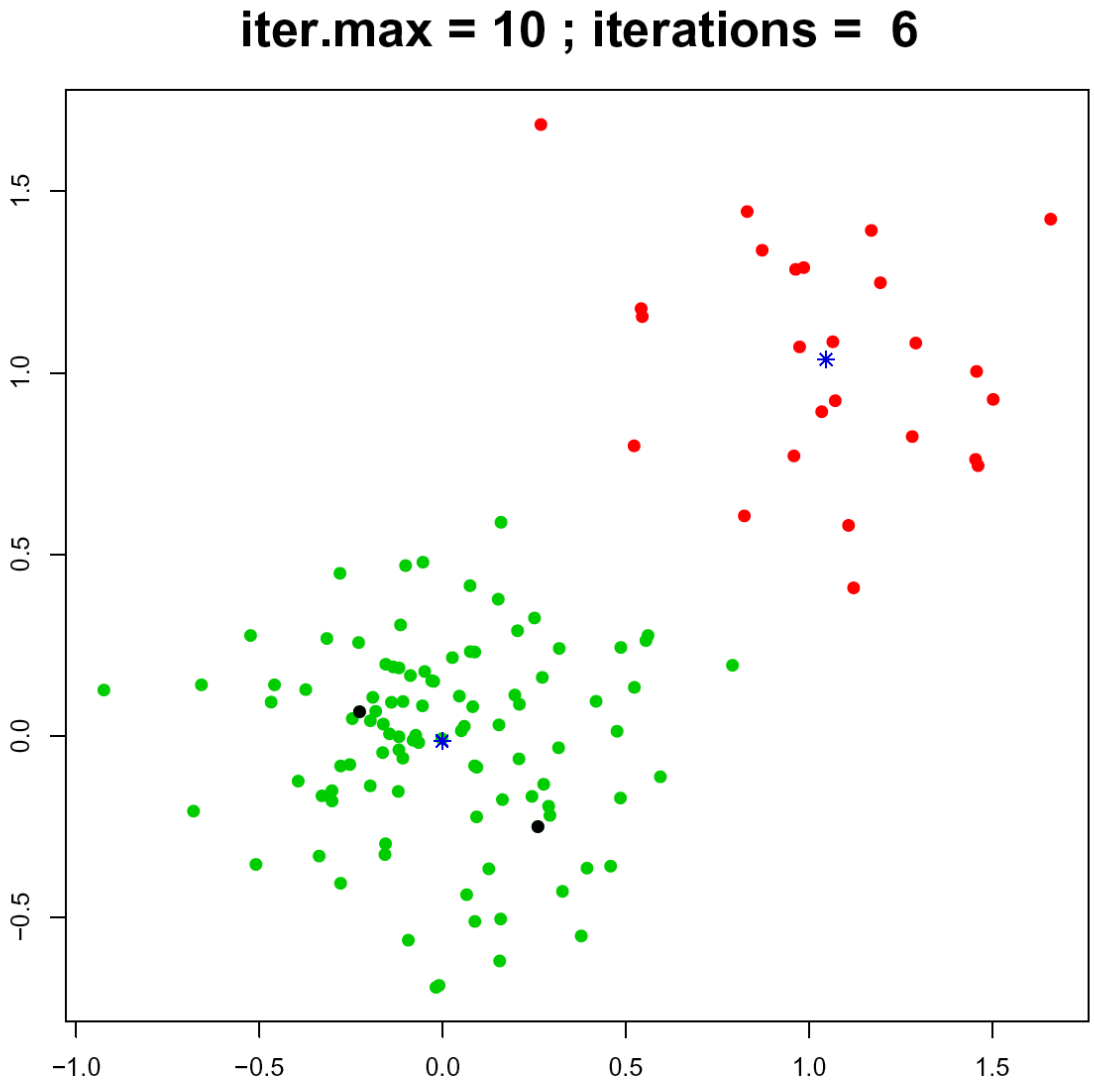
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

Mobile centres example - after 6 iterations



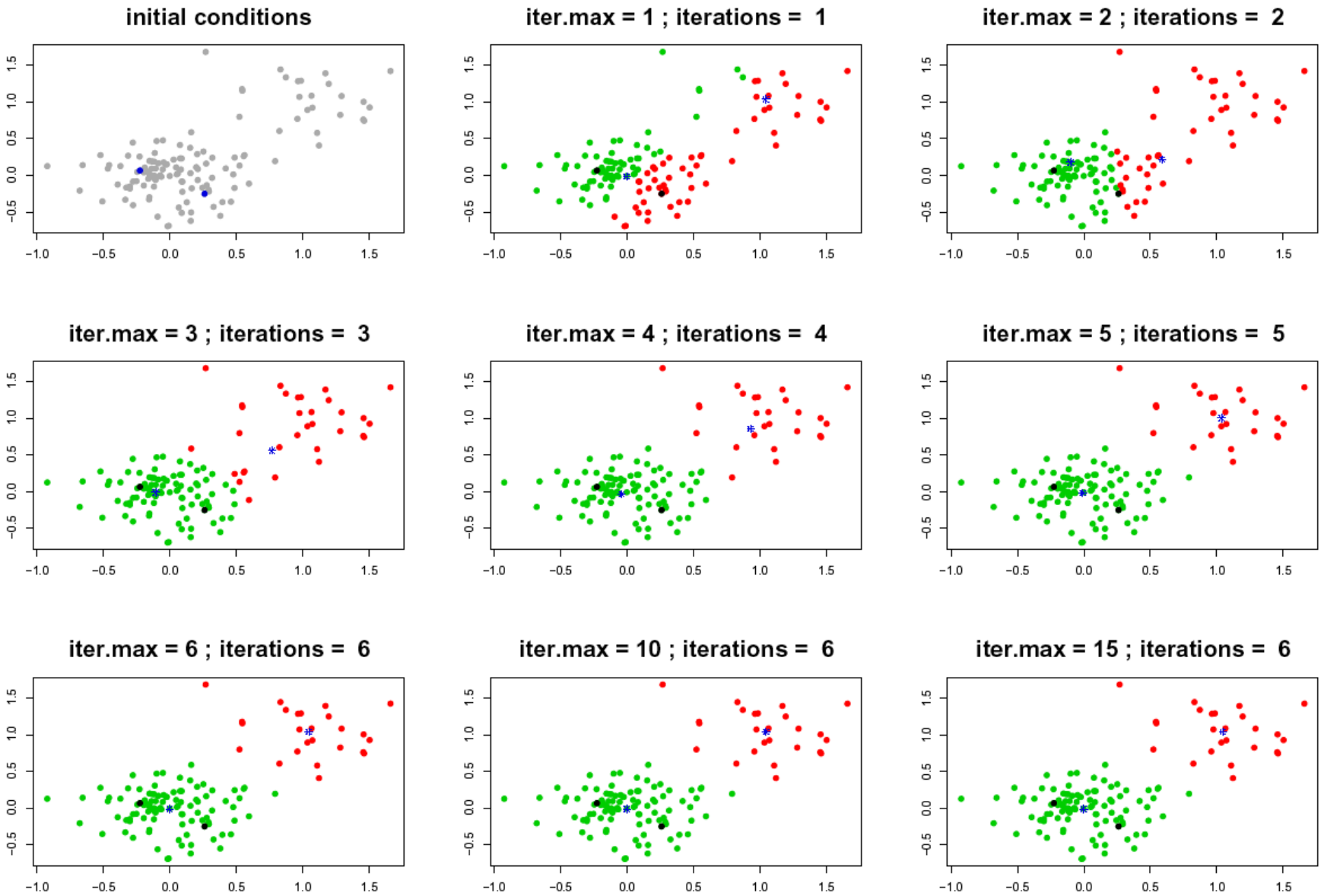
- At each step,
  - points are re-assigned to clusters
  - centres are re-calculated
- Cluster boundaries and centre positions evolve at each iteration

Mobile centres example - after 10 iterations



- After some iterations (6 in this case), the clusters and centres do not change anymore

Mobile centres example - random data



K-means clustering

- K-means clustering is a variant of clustering around mobile centres
- After each assignation of an element to a centre, the position of this centre is re-calculated
- The convergence is much faster than with the basic mobile centre algorithm
  - after 1 iteration, the result might already be stable
- K-means is time- and memory-efficient for very large data sets (e.g. thousands of objects)

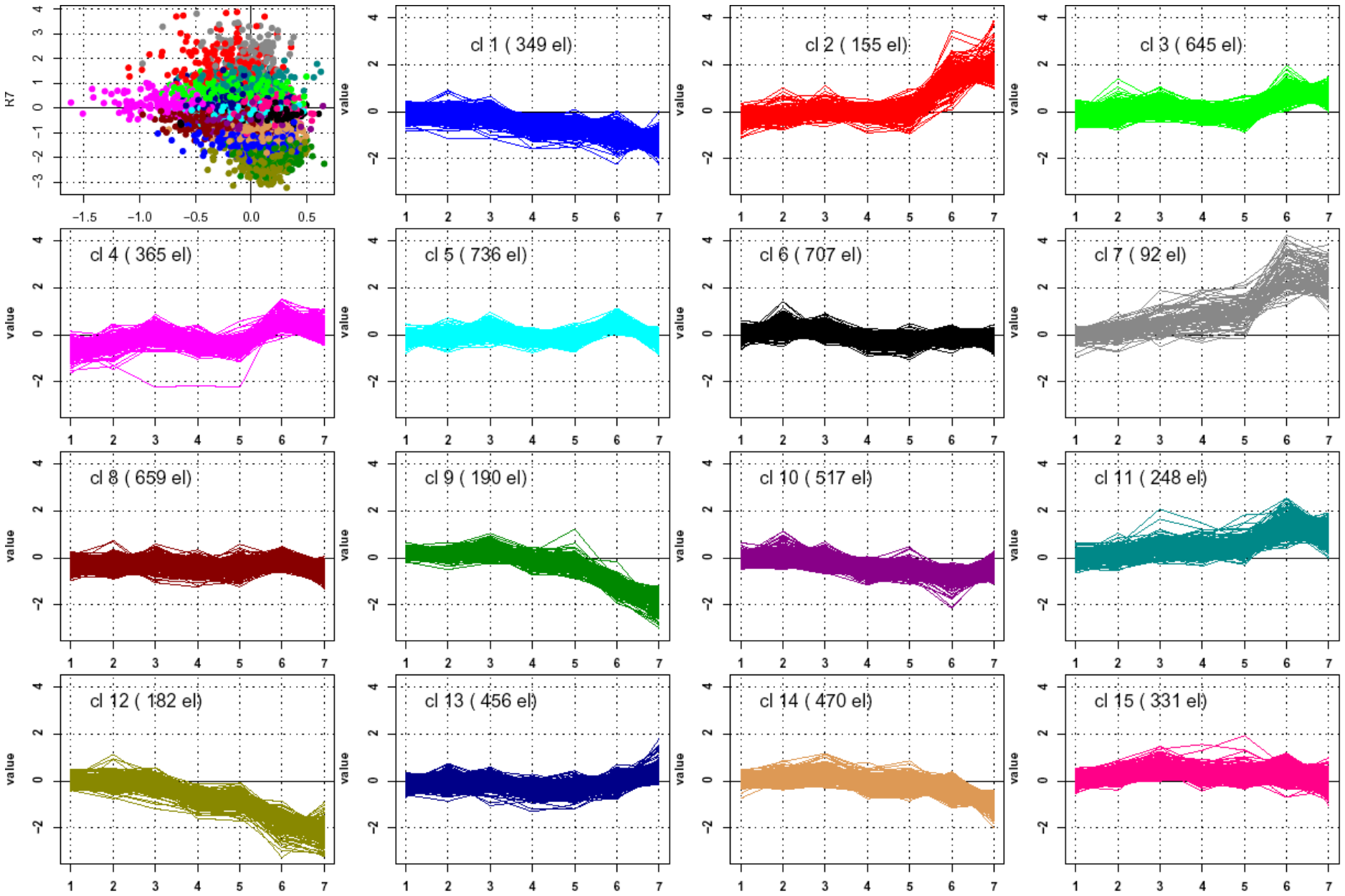
Clustering with gene expression data

- Clustering can be performed in two ways
  - Taking genes as objects and conditions/cell types as variables
  - Taking conditions/cell types as objects and genes as variables
- Problem of dimensionality
  - When genes are considered as variables, there are many more variables than objects
  - Generally, only a very small fraction of the genes are regulated (e.g. 30 genes among 6,000)
  - However, all genes will contribute equally to the distance metrics
  - The noise will thus affect the calculated distances between conditions
- Solution
  - Selection of a subset of strongly regulated genes before applying clustering to conditions/cell types

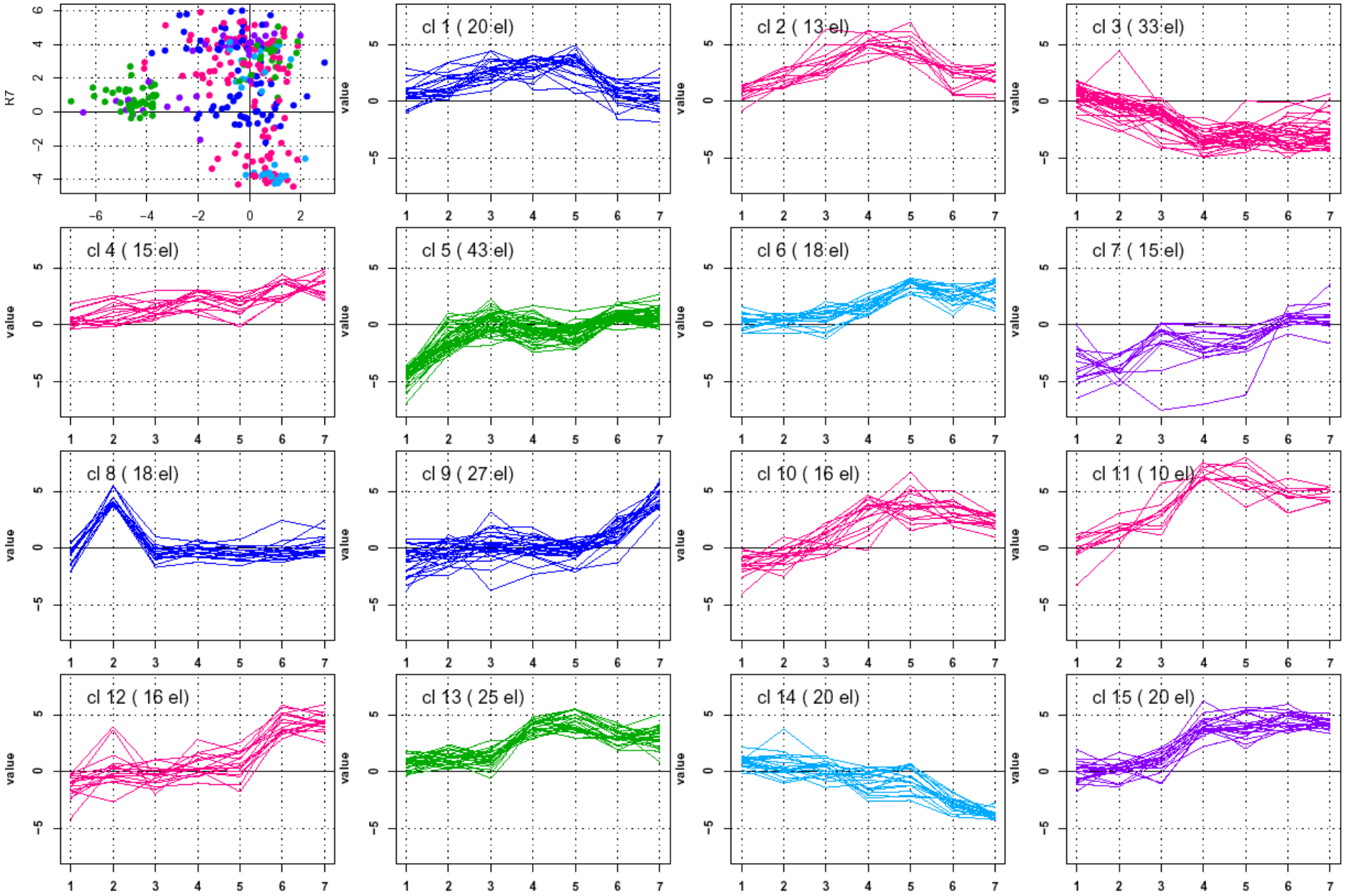
K-means clustering

- K-means clustering is a variant of clustering around mobile centres
- After each assignation of an element to a centre, the position of this centre is re-calculated
- The convergence is much faster than with the basic mobile centre algorithm
  - after 1 iteration, the result might already be stable
- K-means is time- and memory-efficient for very large data sets (e.g. thousands of objects)

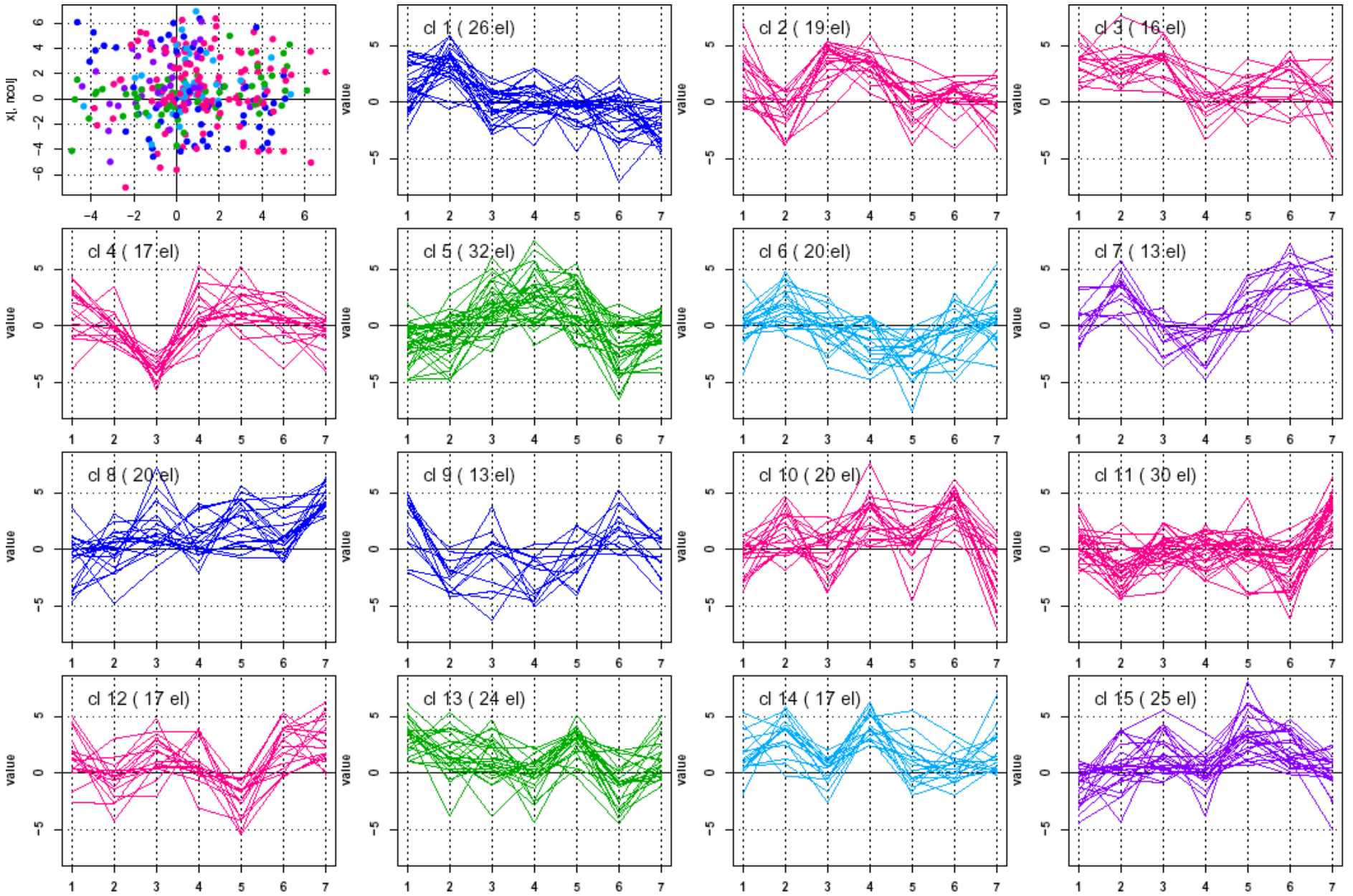
Diauxic shift: k-means clustering on all genes



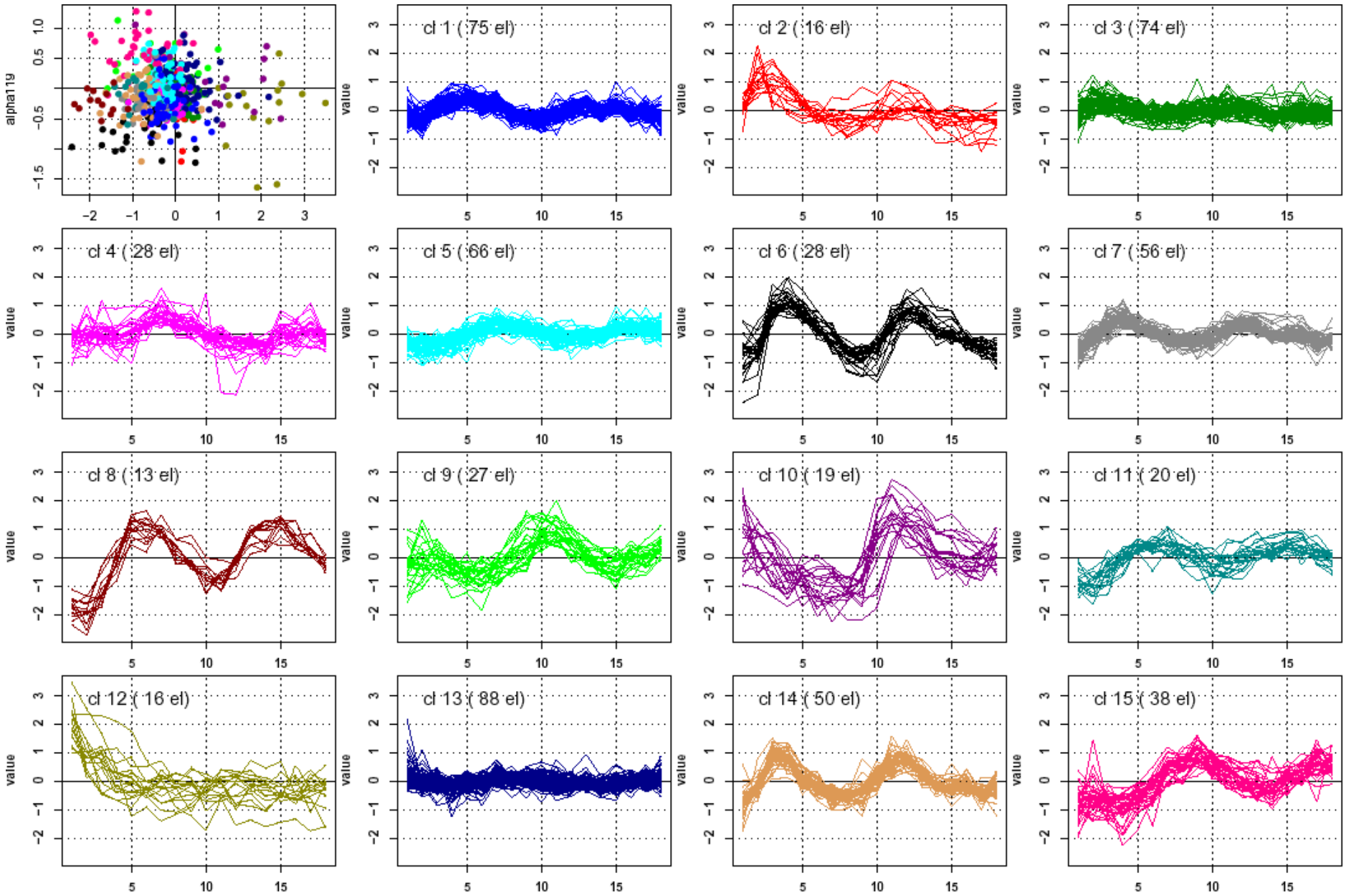
Diauxic shift: k-means clustering on filtered genes



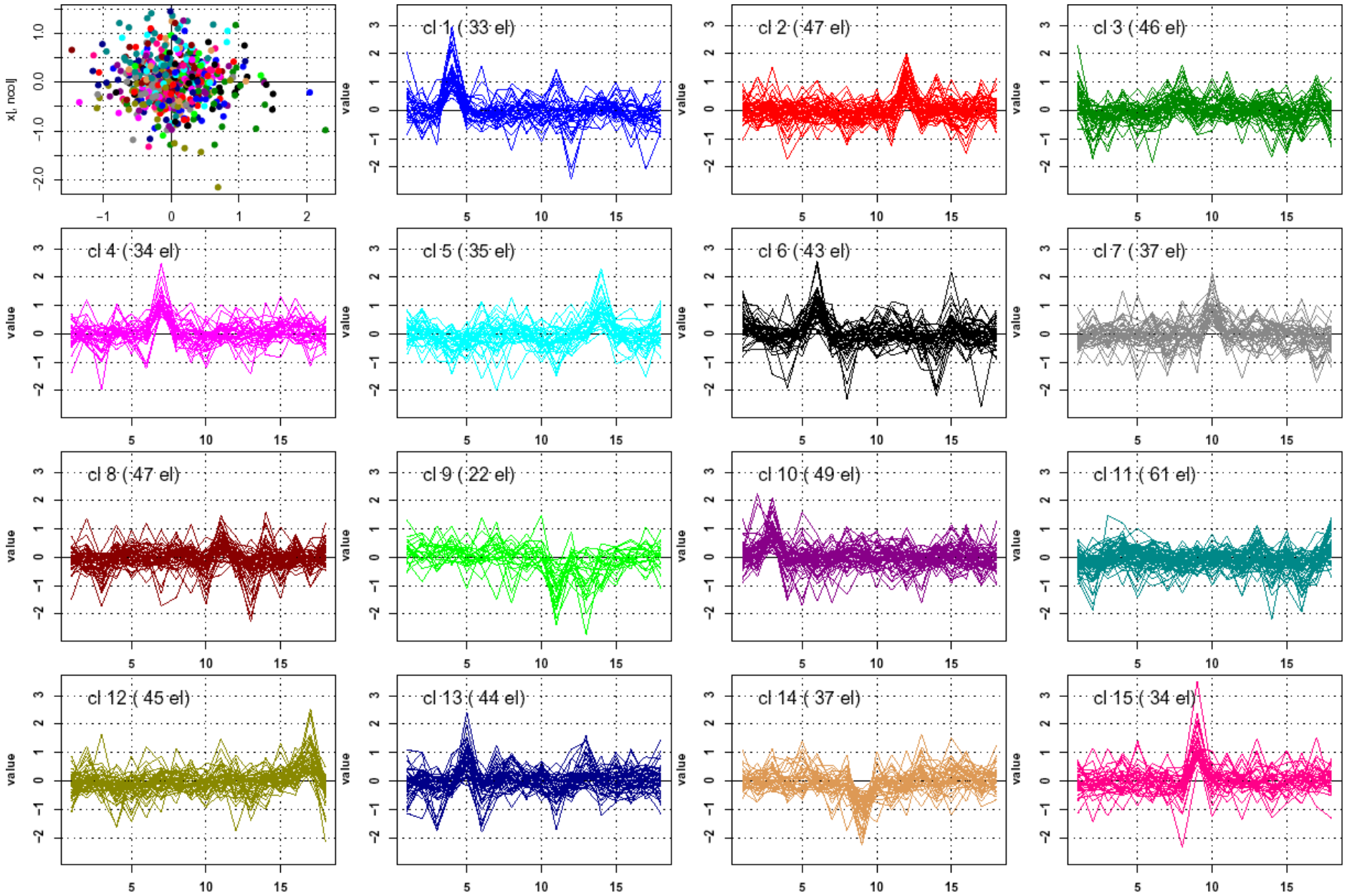
Diauxic shift: k-means clustering on permuted filtered genes



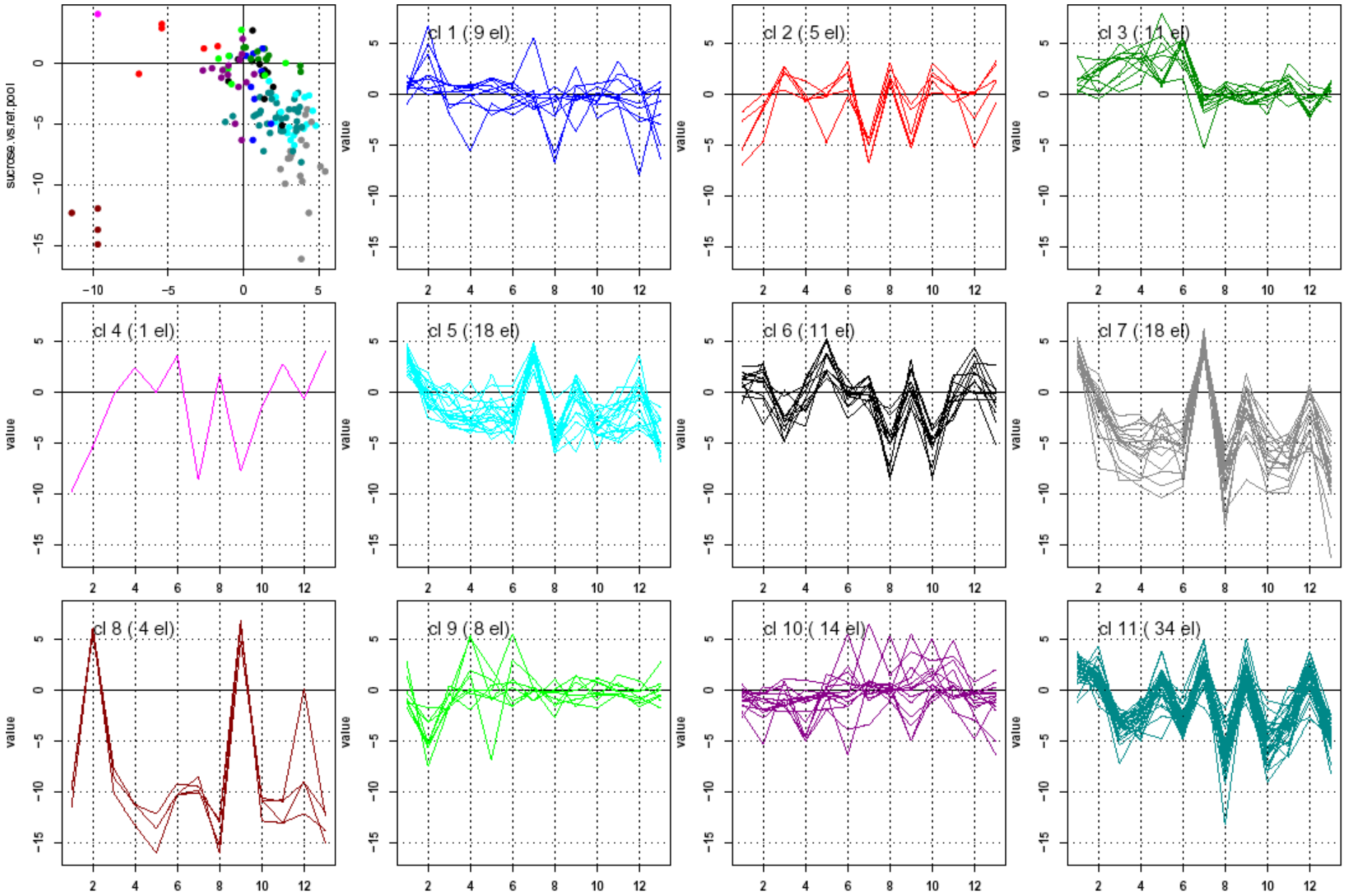
Cell cycle data: K-means clustering



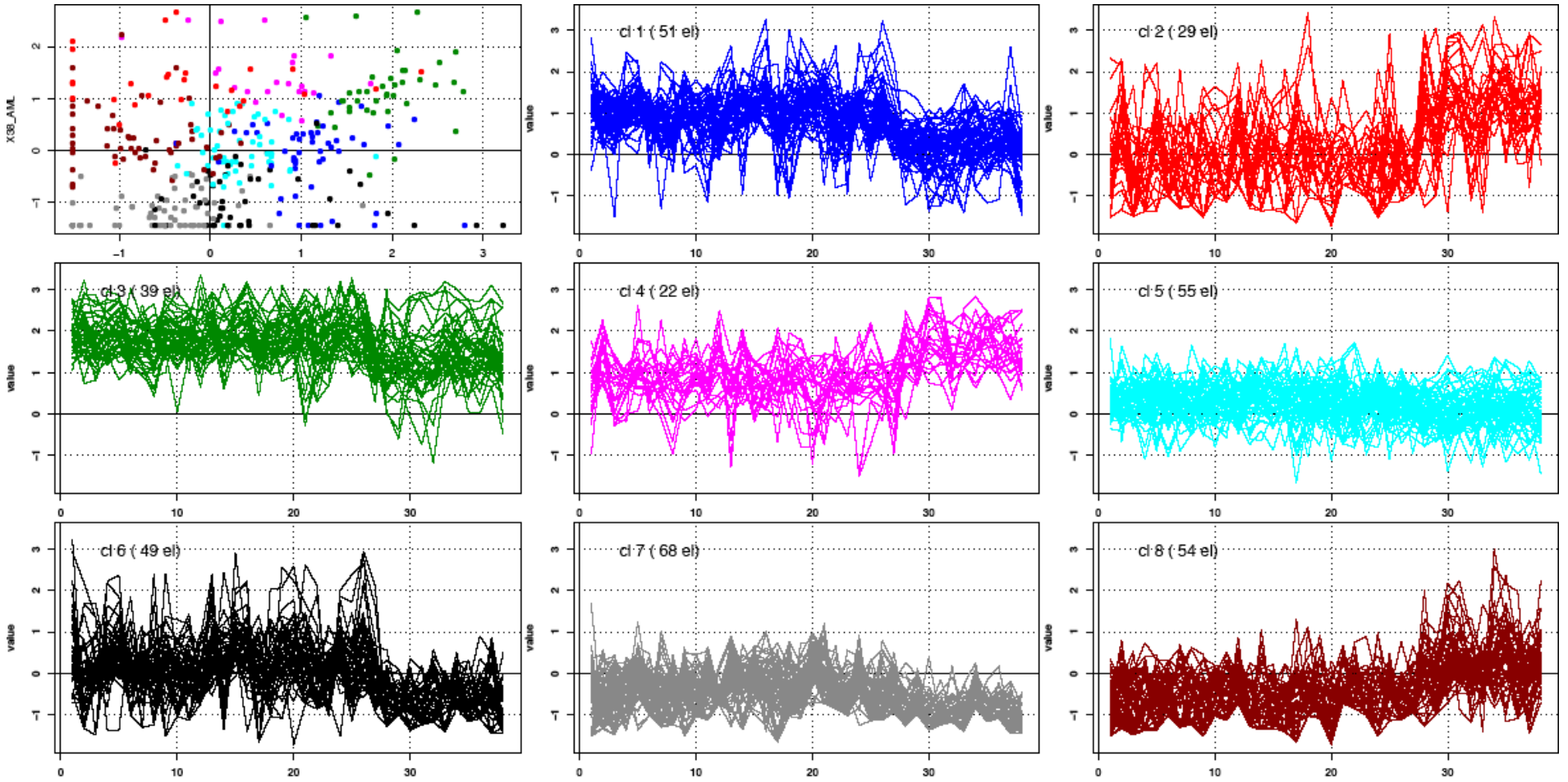
Cell cycle data: K-means clustering, permuted data



Carbon sources: K-means clustering



Golub - K-means clustering



K-means clustering - summary

- Strengths
  - Simple to use
  - Fast
  - Can be used with very large data sets
- Weaknesses
  - The choice of the number of groups is arbitrary
  - The results vary depending on the initial positions of centres
  - The R implementation is based on Euclidian distance, no other metrics are proposed
- Solutions
  - Try different values for k and compare the result
  - For each value of k, run repeatedly to sample different initial conditions
- Weakness of the solution
  - Instead of one clustering, you obtain hundreds of different clustering results, totaling thousands of clusters, how to decide among them

Evaluation of clustering results

How to evaluate the result ?

- It is very hard to make a choice between the multiple possibilities of distance metrics, clustering algorithms and parameters.
- Several criteria can be used to evaluate the clustering results
  - **Consensus**: using different methods, comparing the results and extracting a consensus
  - **Robustness**: running the same algorithm multiple times, with different initial conditions
    - Bootstrap
    - Jack-knife
    - Test different initial positions for the k-means
  - **Biological relevance:** compare the clustering result to functional annotations (functional catalogs, metabolic pathways, ...)

Comparing two clustering results

- If two methods return partitions of the same size, their clusters can be compared in a confusion table
- Optimal correspondences between clusters can be established (permuting columns to maximize the diagonal)
- The consistency between the two classifications can then be estimated with the hit rate
- Example :
  - Carbon source data, comparison of k-means and hierarchical clustering

hierarchical clustering / k-means clustering

	k1	k2	k3	k4	k5	k6	k7	Sum
h1	0	0	2	18	14	1	0	35
h2	0	0	0	4	0	0	0	4
h3	0	0	0	0	10	0	0	10
h4	40	0	10	0	0	9	0	59
h5	2	12	0	0	0	5	0	19
h6	0	0	0	0	0	0	4	4
h7	0	2	0	0	0	0	0	2
Sum	42	14	12	22	24	15	4	133

hierarchical clustering / k-means clustering

	k4	k3	k5	k1	k2	k7	k6	Sum
h1	18	2	14	0	0	0	1	35
h2	4	0	0	0	0	0	0	4
h3	0	0	10	0	0	0	0	10
h4	0	10	0	40	0	0	9	59
h5	0	0	0	2	12	0	5	19
h6	0	0	0	0	0	4	0	4
h7	0	0	0	0	2	0	0	2
Sum	22	12	24	42	14	4	15	133

Correspondence between clusters

hierarchical	h1	h2	h3	h4	h5	h6	h7
k-means	k4	k3	k5	k1	k2	k7	k6

Matches 84 Hit rate 63.2%

Mismatches 49 Error rate 36.8%

Evaluation of robustness - Bootstrap

- The bootstrap consists in repeating r times (for example r=100) the clustering, using each time
  - Either a different subset of variables
  - Or a different subset of objects
- The subset of variables is selected randomly, with resampling (i.e. the same variable can be present several times, whilst other variables are absent.
- On the images the tree is colored according to the reproducibility of the branches during a 100-iterations bootstrap.