elaborator

Seriation of laboratory parameters

Currently, there are three options for seriation

AI sorted:	Use of intelligent algorithms to locate laboratory parameters with similar changes close to each other.
As in input:	This is the most flexible option and allows the a user-defined sorting of the lab parameters. The parameters are sorted in the order as they occur in the input dataset.
Alphabetically (default):	Sort lab parameters alphabetically

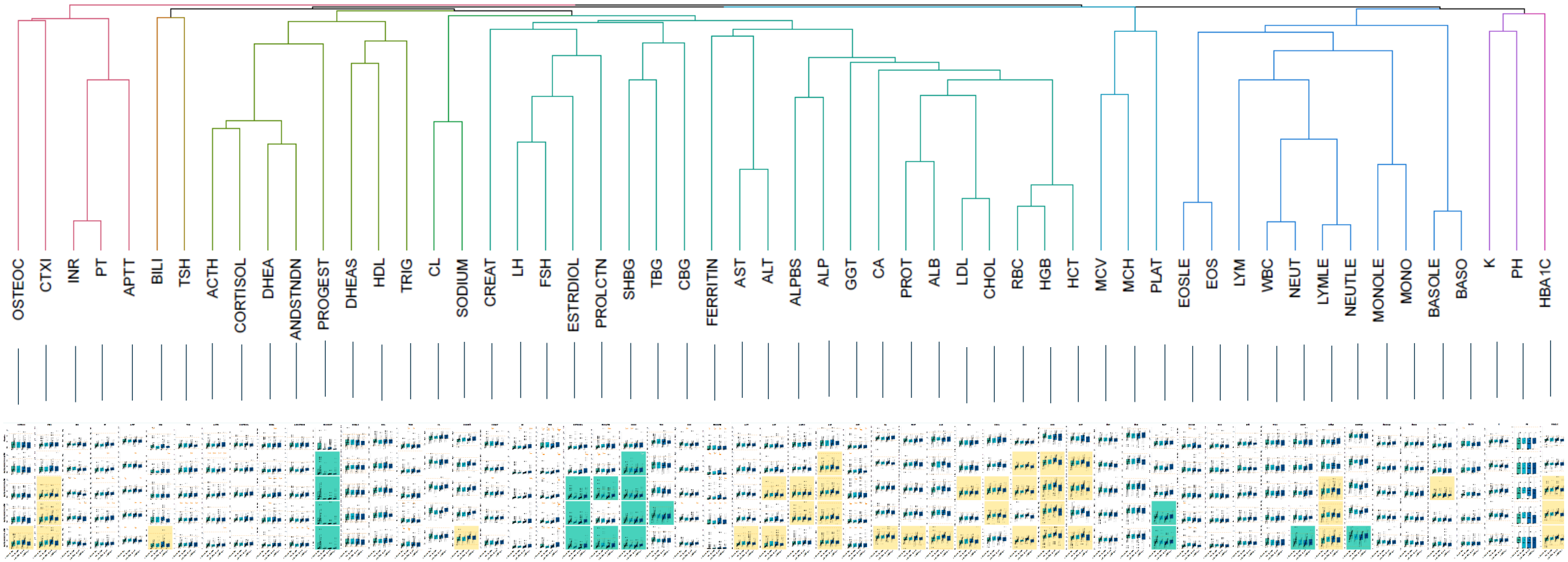
This manual focusses on AI sorting.

Why using AI for seriation of lab parameters?

- Seriation algorithms can locate laboratory parameters with similar changes close to each other.
- This enables the user to see which laboratory parameters have similar changes and makes exploration through the e**lab**orator even more efficient.
- Changes refer to the change from one visit to another visit. The two visits are to be selected by the user in the application.

AI for seriating laboratory parameters – An example

Use of intelligent algorithms makes exploration even more efficient



AI seriation in the e**lab**orator

A seriation of laboratory parameters is defined through a

// Distance measure for assessing similarity of lab parameters, and

// Seriation algorithm for locating similar lab parameters close to each other.

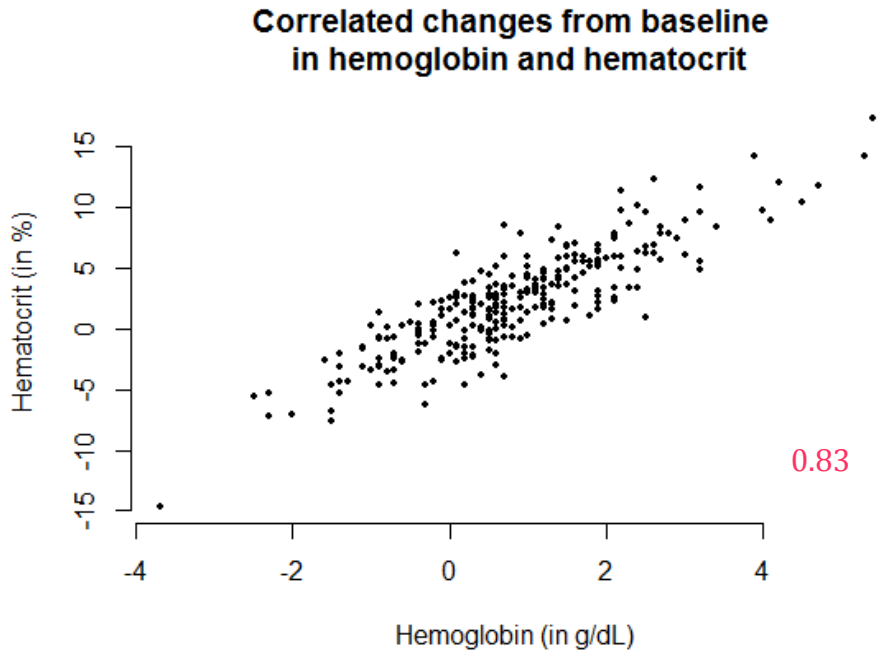
The choice of a reasonable distance measure is essential and cannot be changed by the user in the e**lab**orator.

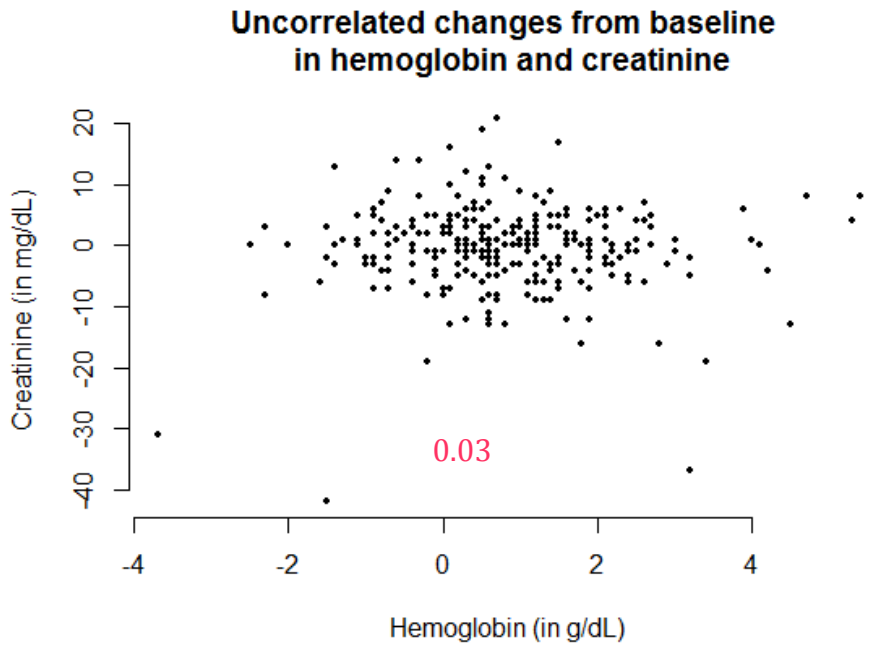
There are several different seriation algorithms, there is no ‚best' one. The user can change the algorithm used in the e**lab**orator. This manual will give a description for each of them to help in understanding how seriation works.

Assessing similarity of laboratory parameters via distance measure

Correlation-based distance

// Correlation describes if changes of lab parameters are correlated (left) or uncorrelated (right).

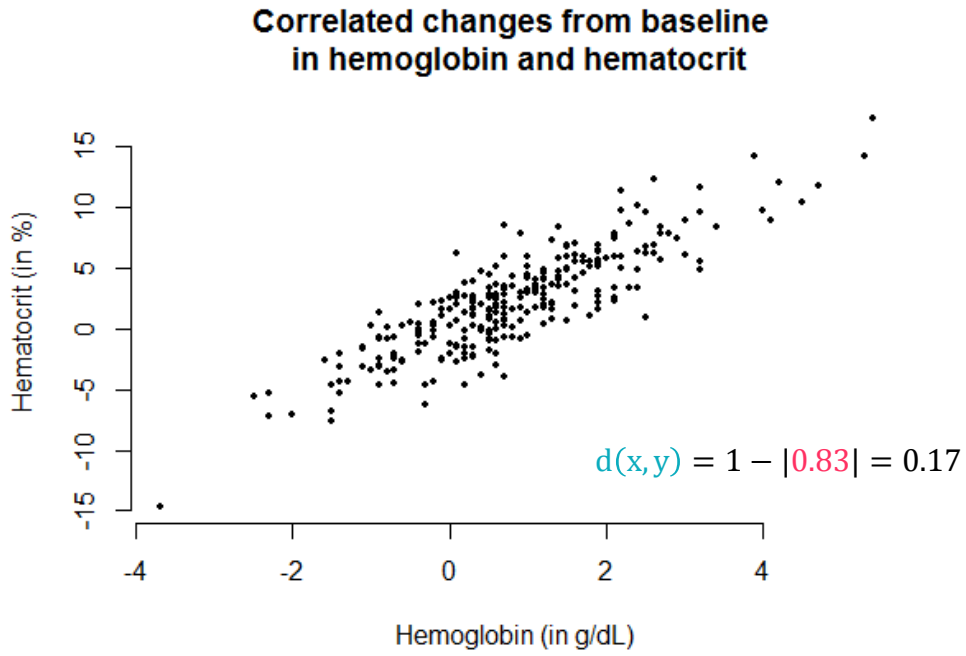


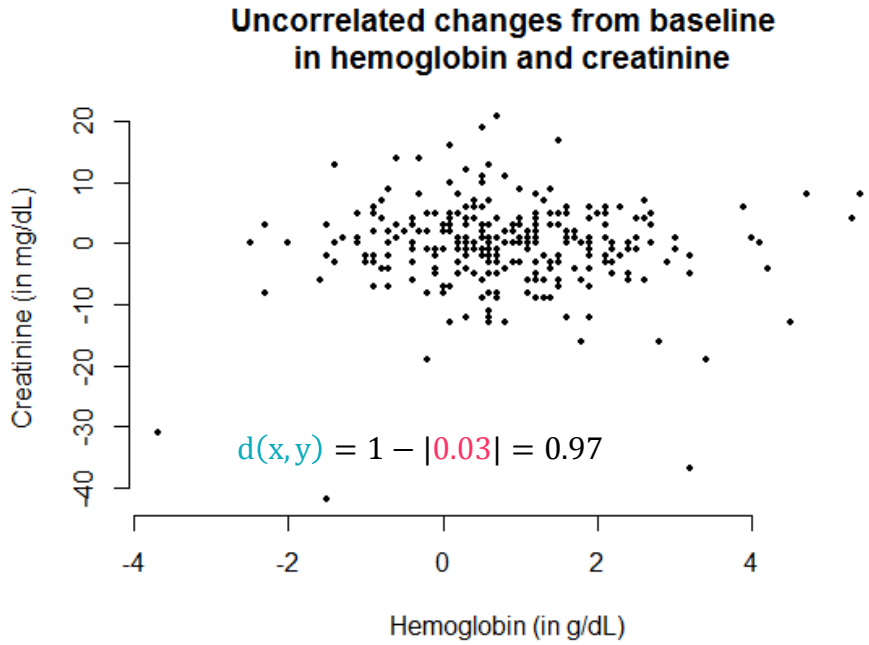


// We want lab parameters with high correlation (here: hemoglobin & hematocrit) being located close to each other.

Correlation-based distance

// Correlation-based distance: $d(x, y) = 1 - |\rho_{xy}|$, with ρ_{xy} denoting the correlation coefficient (Spearman)

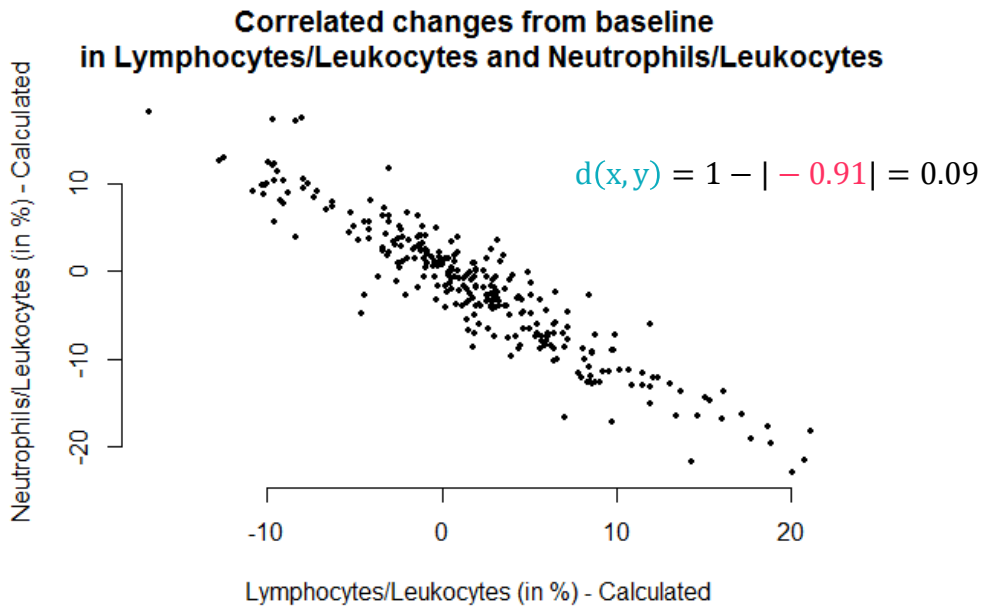




// Distances close to 0 indicate that lab parameters have similar changes and shall be located close to each other. Distances close to 1 indicate dissimilar changes.

Correlation-based distance

// Correlation-based distance: $d(x, y) = 1 - |\rho_{xy}|$, with ρ_{xy} denoting the correlation coefficient (Spearman)



// Lab parameters with high negative correlation shall be located close to each other as well.

Locating similar laboratory parameters close to each other by seriation algorithms

Seriation algorithms

- Several seriation algorithms are available in the e**lab**orator and can be selected by the user.
- The implementation of these algorithms is completely based on the R package seriation (Hahsler et al, 2008).
  - An intuitive and simplified description of the methods is provided in the following.
  - For details, please see information available at https://cran.r-project.org/web/packages/seriation/

Hahsler, M., Hornik, K., & Buchta, C. (2008). Getting Things in Order: An Introduction to the R Package seriation. *Journal of Statistical Software, 25*(3), 1-34.

Seriation algorithms PART 1

The following seriation algorithms based on unsupervised learning are available in e**lab**orator:

// “OLO_average” → **O**ptimal **L**eaf **O**rdering and average linkage

// “OLO_complete” → **O**ptimal **L**eaf **O**rdering and complete linkage

// “OLO_single” → **O**ptimal **L**eaf **O**rdering and single linkage

// “OLO_ward” → **O**ptimal **L**eaf **O**rdering and ward linkage

// “GW_average” → **G**ruvaeus **W**ainer heuristic and average linkage

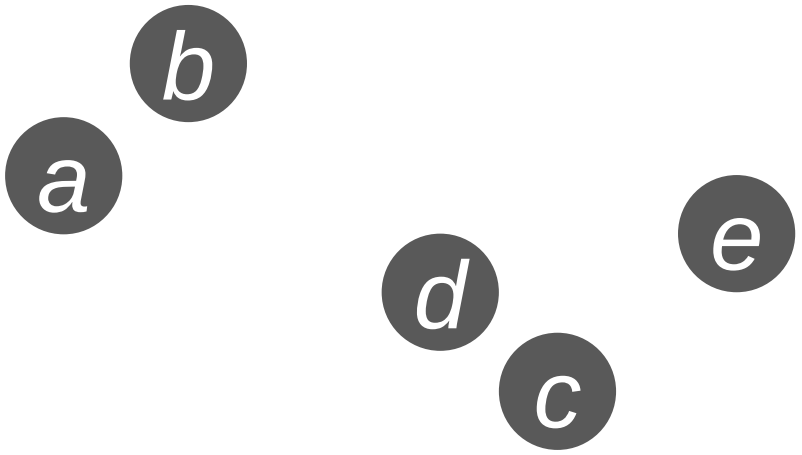
// “GW_complete” → **G**ruvaeus **W**ainer heuristic and complete linkage

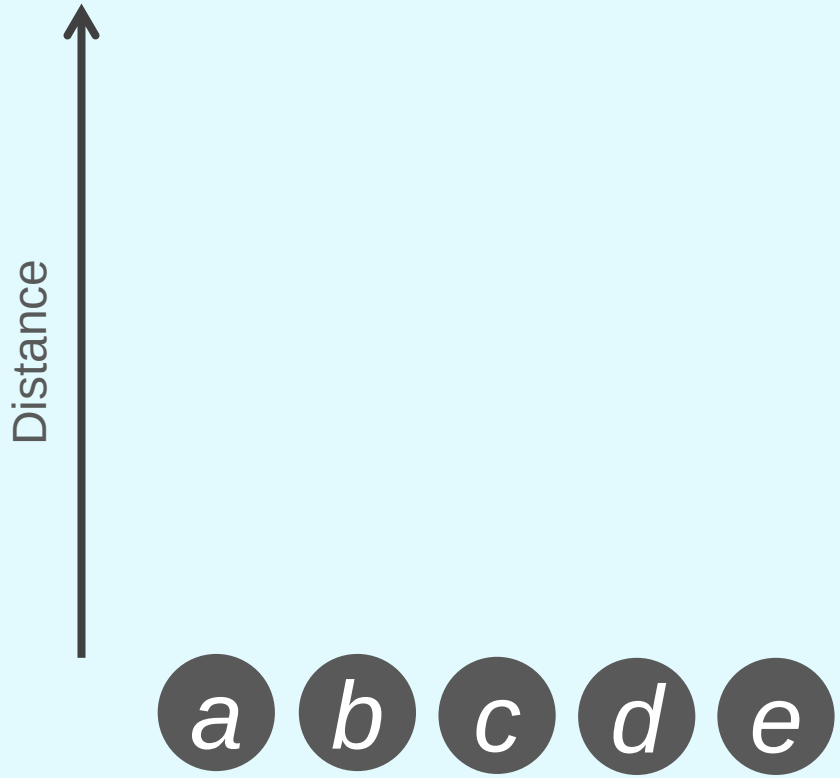
// “GW_single” → **G**ruvaeus **W**ainer heuristic and single linkage

// “GW_ward” → **G**ruvaeus **W**ainer heuristic and ward linkage

Agglomerative clustering

An illustration of an unsupervised learning method

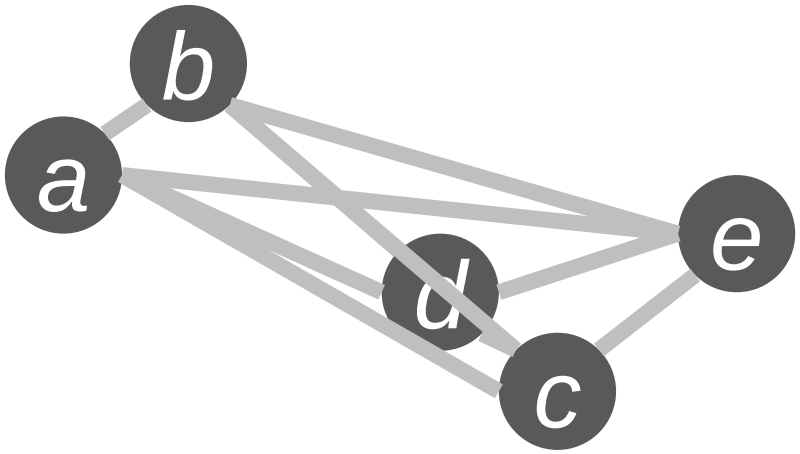


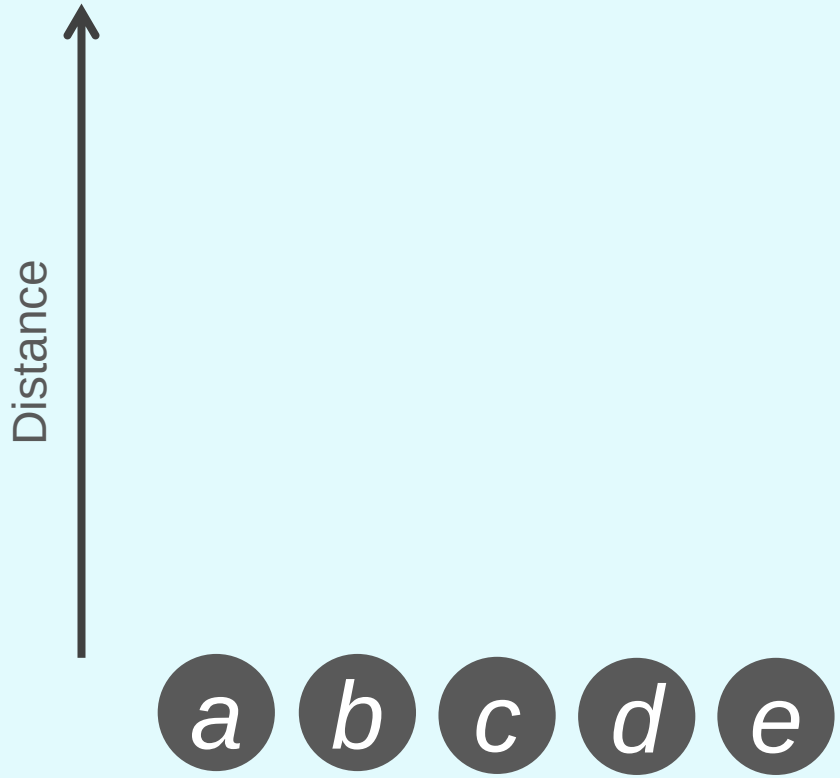


Each lab parameter forms an own cluster

Agglomerative clustering

An illustration of an unsupervised learning method

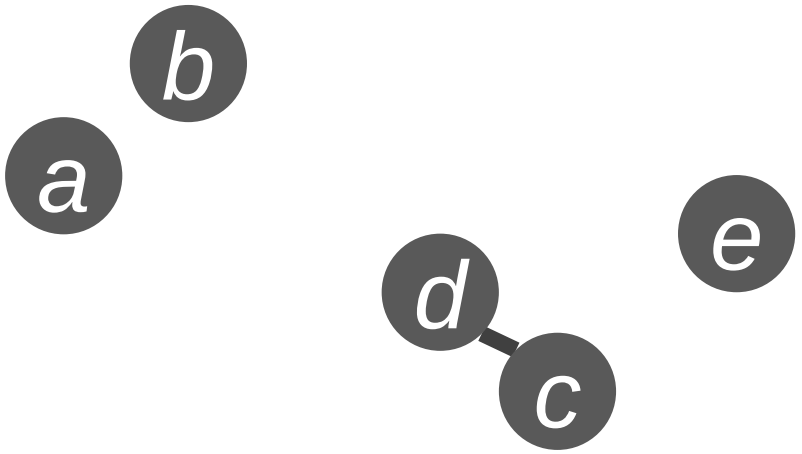


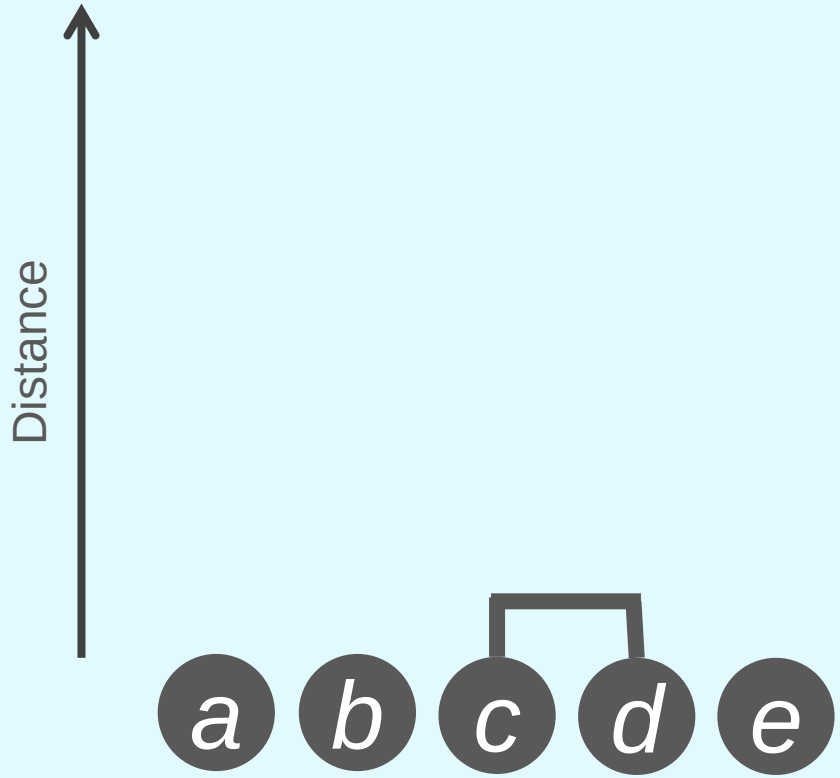


Compute distance between each pair of laboratory parameters

Agglomerative clustering

An illustration of an unsupervised learning method

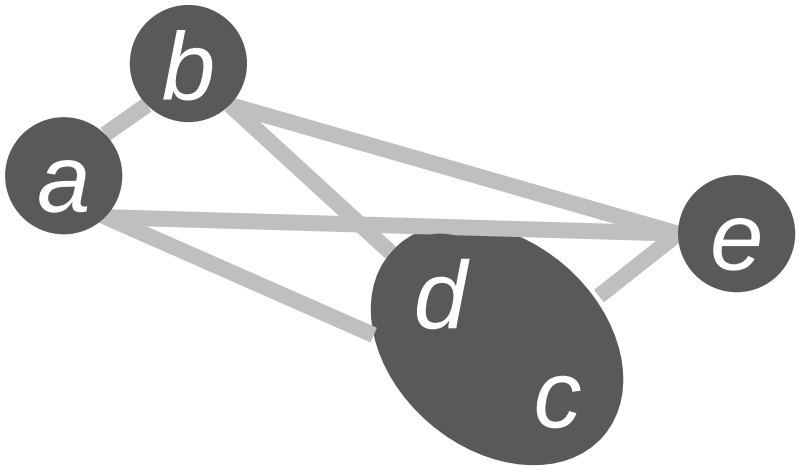


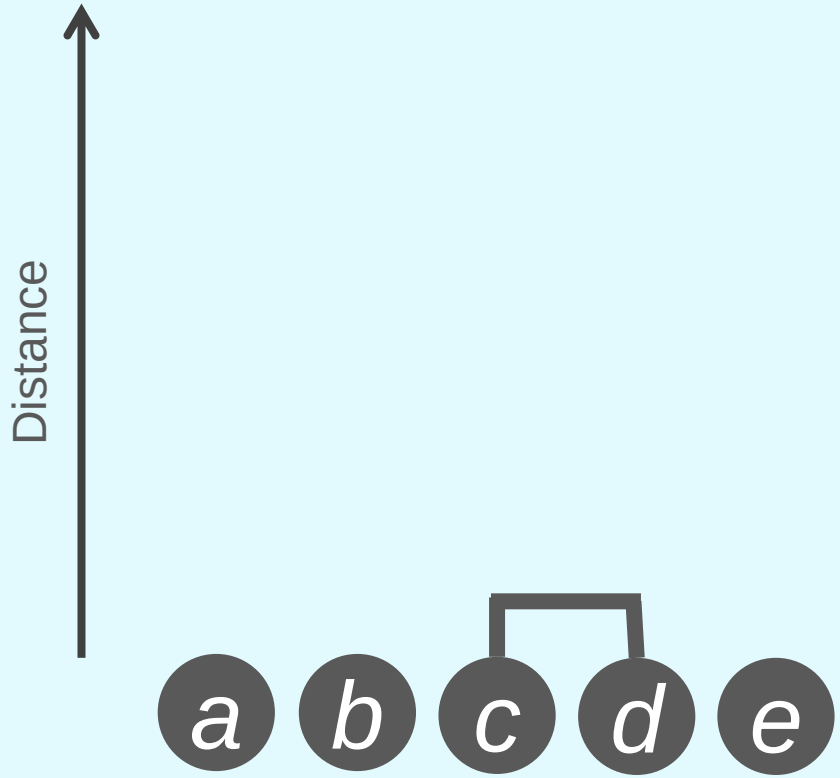


Lab parameters with smallest distance are merged

Agglomerative clustering

An illustration of an unsupervised learning method

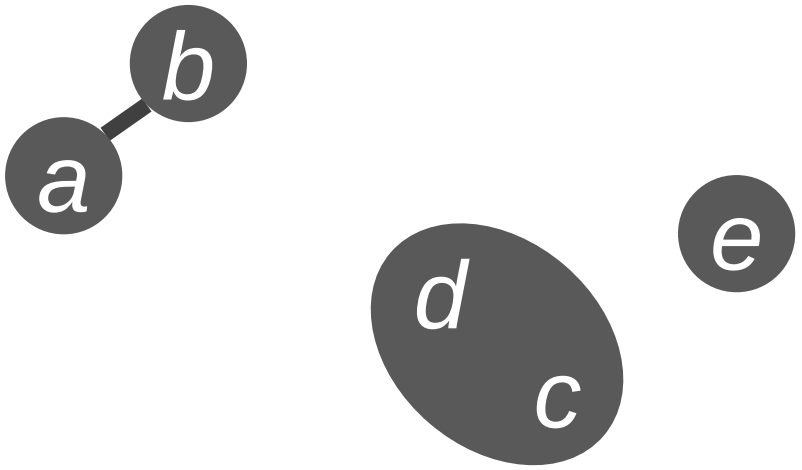


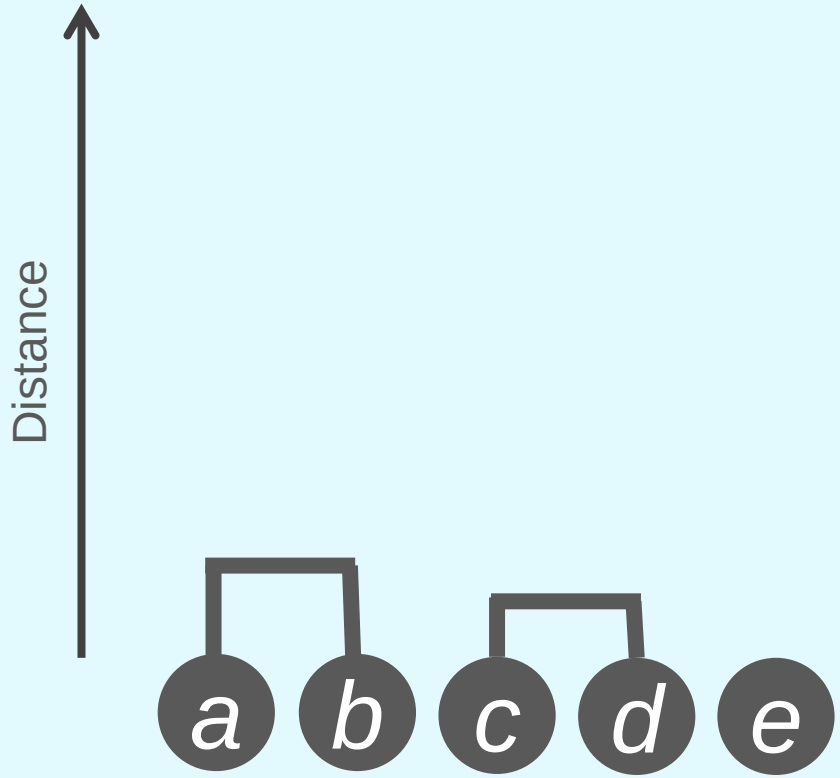


Compute distance between each pair of laboratory parameters

Agglomerative clustering

An illustration of an unsupervised learning method

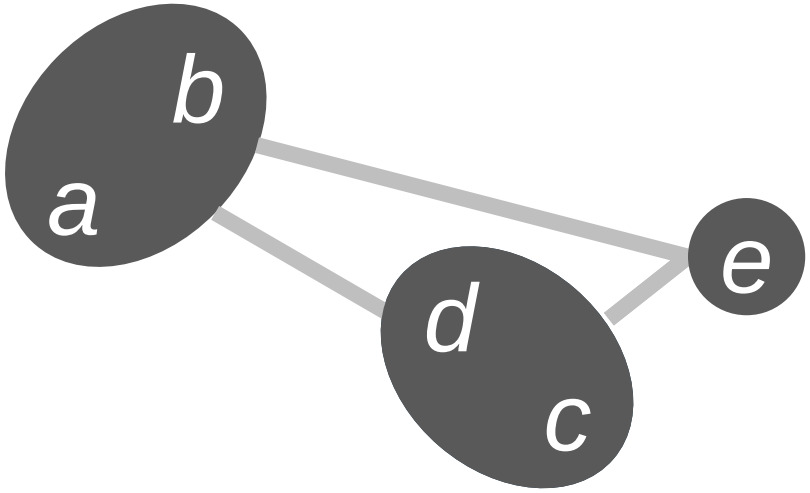


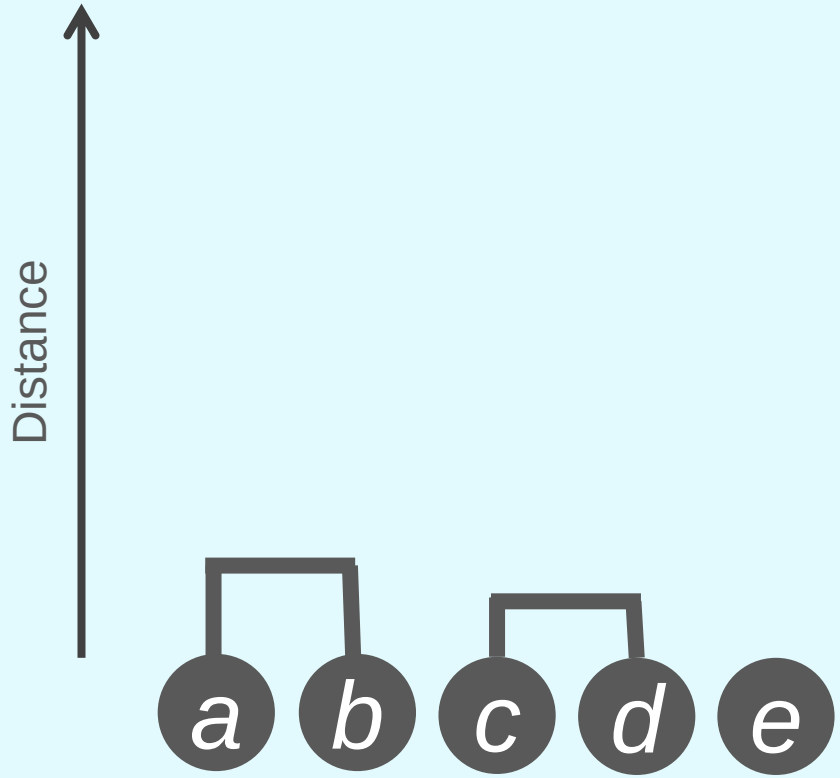


Lab parameters with smallest distance are merged

Agglomerative clustering

An illustration of an unsupervised learning method

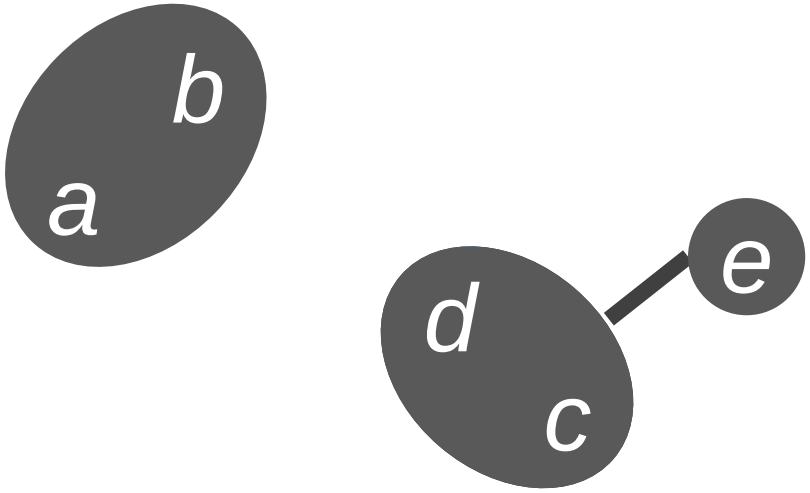


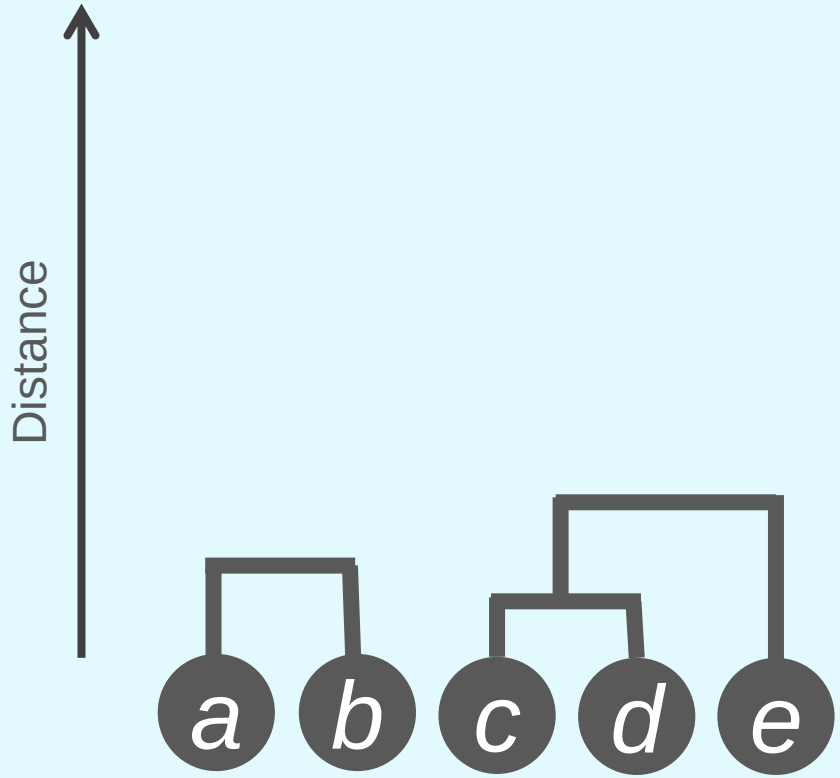


Compute distance between each pair of laboratory parameters

Agglomerative clustering

An illustration of an unsupervised learning method

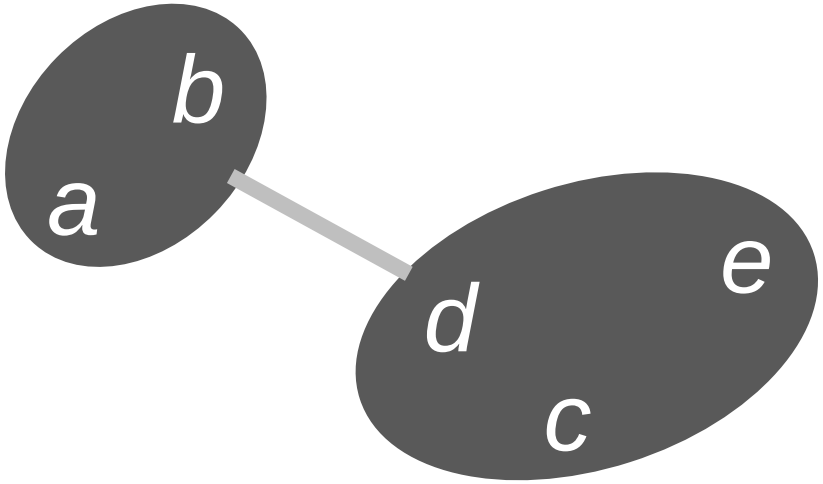


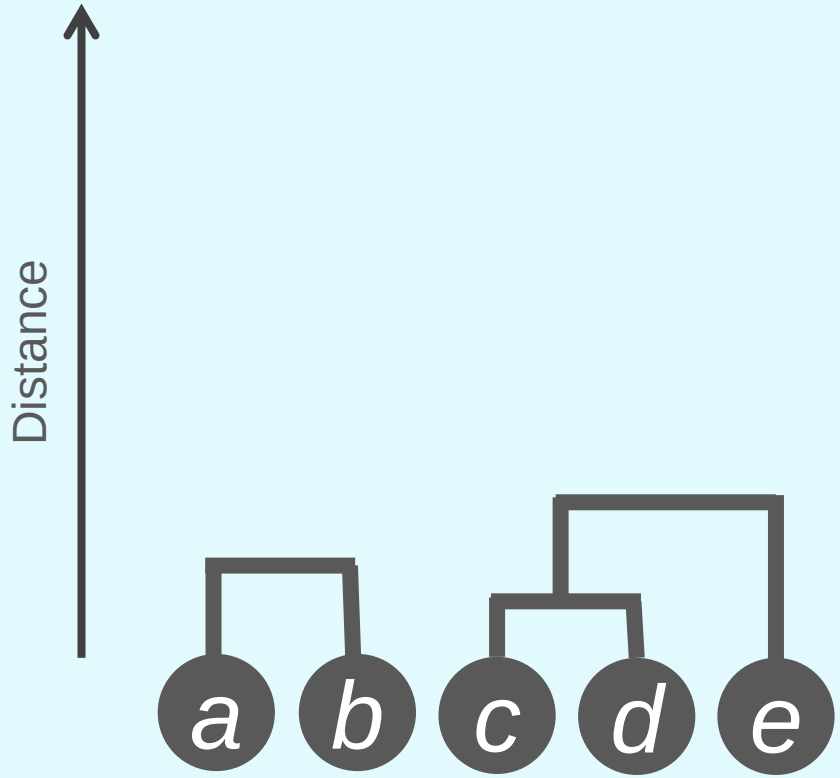


Lab parameters with smallest distance are merged

Agglomerative clustering

An illustration of an unsupervised learning method

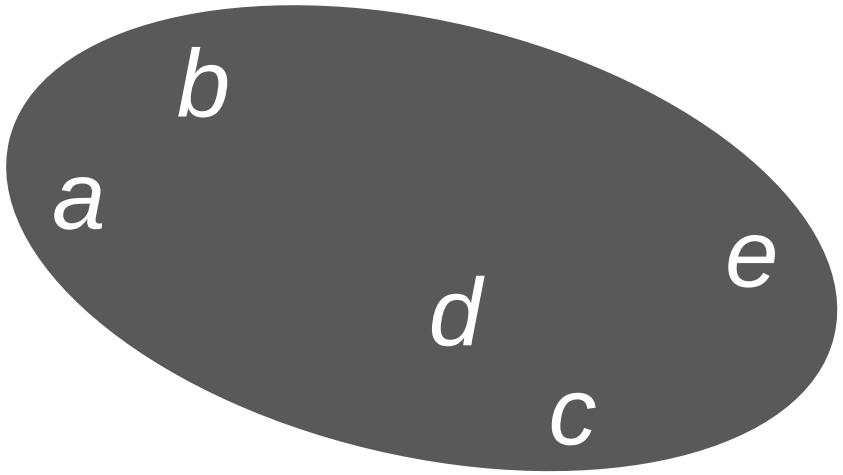




Compute distance between each pair of laboratory parameters

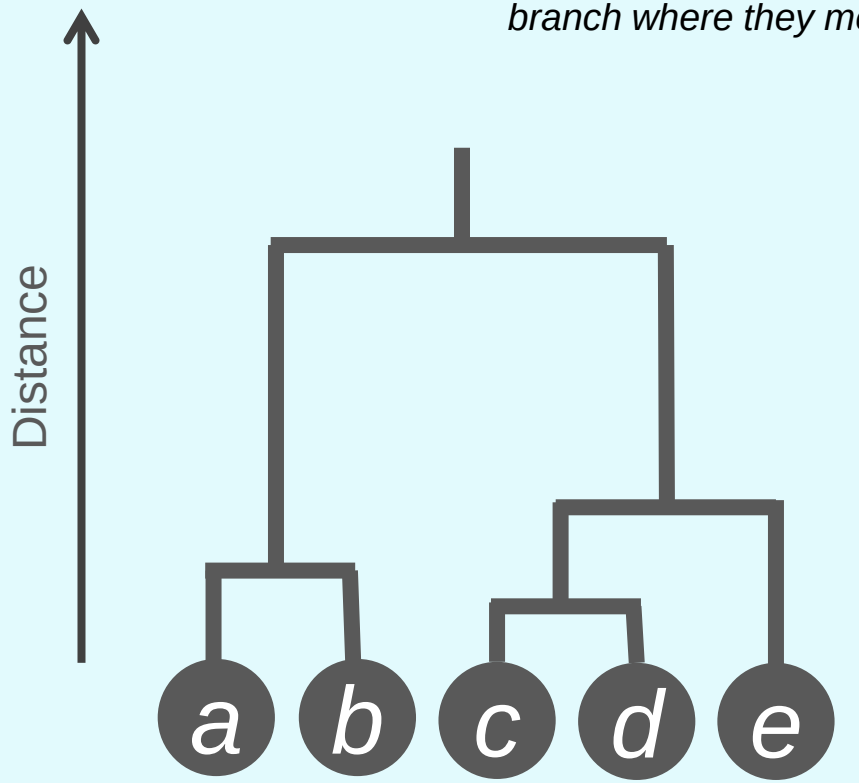
Agglomerative clustering

An illustration of an unsupervised learning method



Lab parameters with smallest distance are merged

The similarity of two clusters / lab parameters is encoded in the height of the branch where they merge.



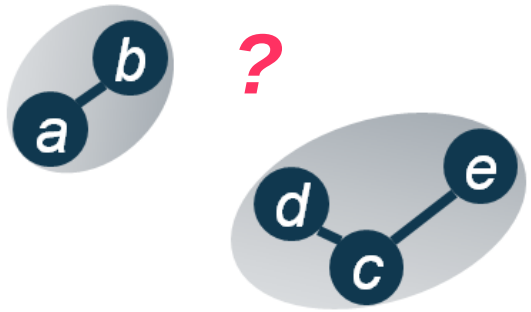
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$

Distance between clusters of laboratory parameters

Linkage schemes

// Computing distance between pairs of laboratory parameters is straightforward.

// But how to compute distance between clusters of laboratory parameters?



Distance between clusters of laboratory parameters

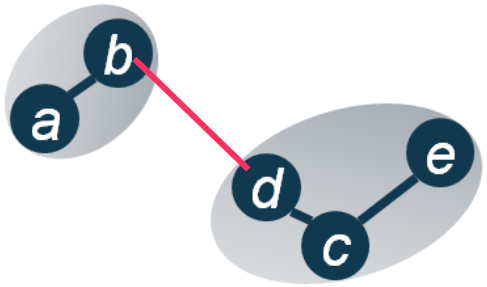
Linkage schemes

// Computing distance between pairs of laboratory parameters is straightforward.

// But how to compute distance between clusters of laboratory parameters?

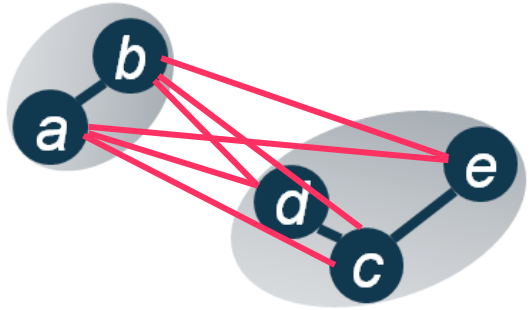
// *Different linkage schemes exist!*

Single linkage



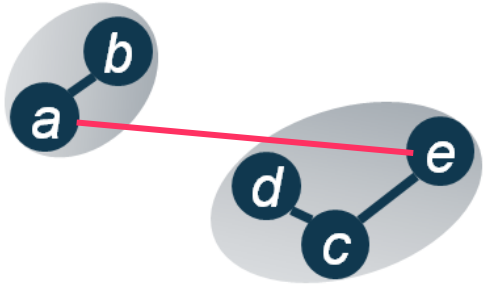
→ OLO_single
→ GW_single

Average linkage



→ OLO_average
→ GW_average

Complete linkage

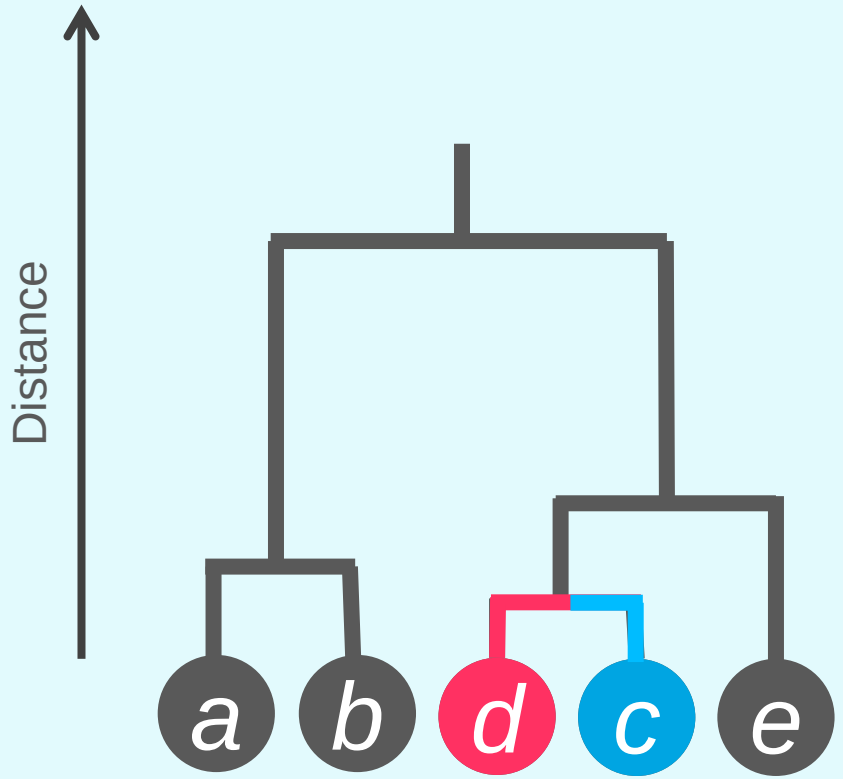


→ OLO_complete
→ GW_complete

Dendrogram

- Graphical tool to display the result of a hierarchical clustering
- Dendrogram might be used to order lab parameters
- BUT: dendrogram is not unique!

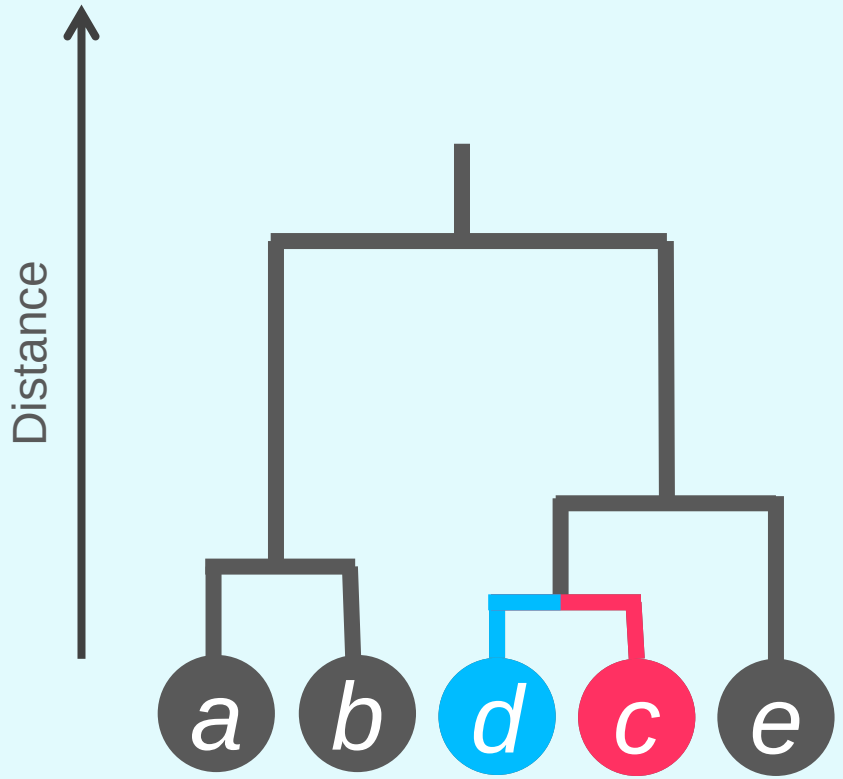
  There are 2^{n-1} possible orderings consistent with tree structure



Dendrogram

- Graphical tool to display the result of a hierarchical clustering
- Dendrogram might be used to order lab parameters
- BUT: dendrogram is not unique!

  There are 2^{n-1} possible orderings consistent with tree structure

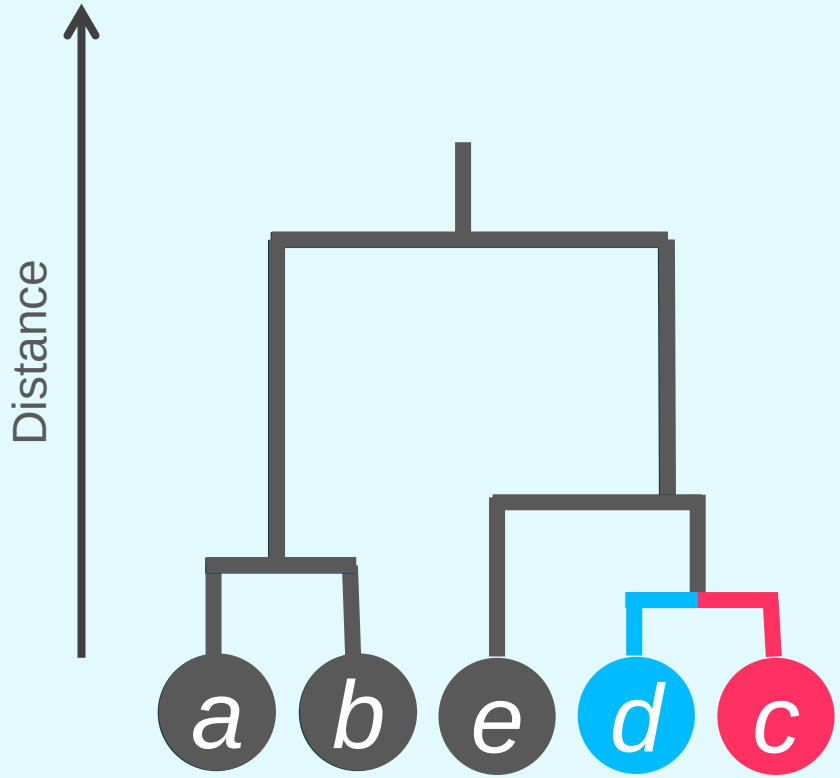


Dendrogram

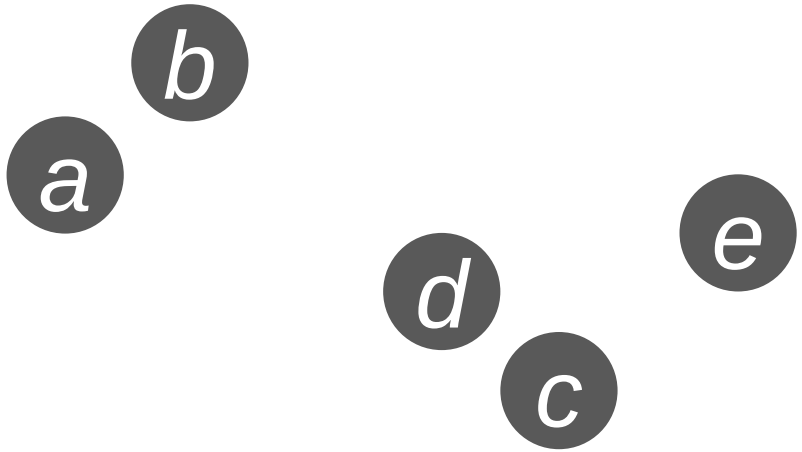
- Graphical tool to display the result of a hierarchical clustering
- Dendrogram might be used to order lab parameters
- BUT: dendrogram is not unique!

  There are 2^{n-1} possible orderings consistent with tree structure

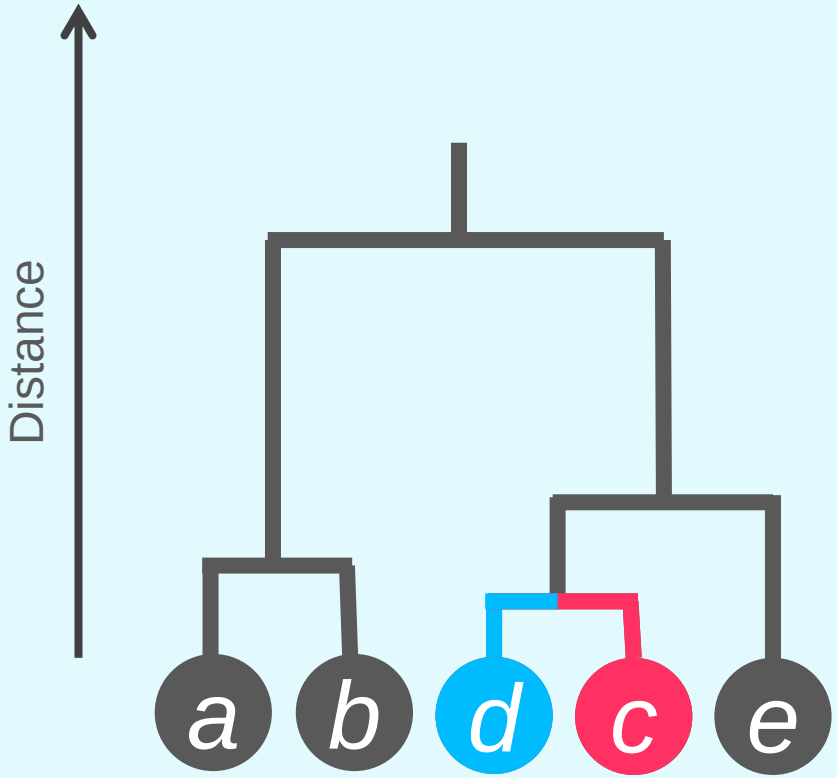
- Rotation methods rotate branches such that neighbored lab parameters are most similar.
  - Optimal leaf ordering (OLO)
  - Gruvaeus Wainer heuristic (GW)



Dendrogram



- Rotation methods rotate branches such that neighbored lab parameters are most similar.
  - Optimal leaf ordering (OLO)
  - Gruvaeus Wainer heuristic (GW)



$a \rightarrow b \rightarrow d \rightarrow c \rightarrow e$

Rotation methods for dendrogram

- Optimal leaf ordering (OLO)
  - Rough idea:

    Given a dendrogram, find an equivalent dendrogram which maximizes the sum of similarity of any adjacent objects.

  - Conducted after a dendrogram has been constructed

- Gruvaeus Wainer heuristic (GW)
  - Rough idea:

    After merging two clusters / lab parameters, order the branches of the dendrogram in such a way that the lab parameters at the edge of adjacent subtrees are most similar.

  - Part of the construction of a dendrogram (additional step in the iteration)
  - Faster than OLO but less optimal (heuristic)

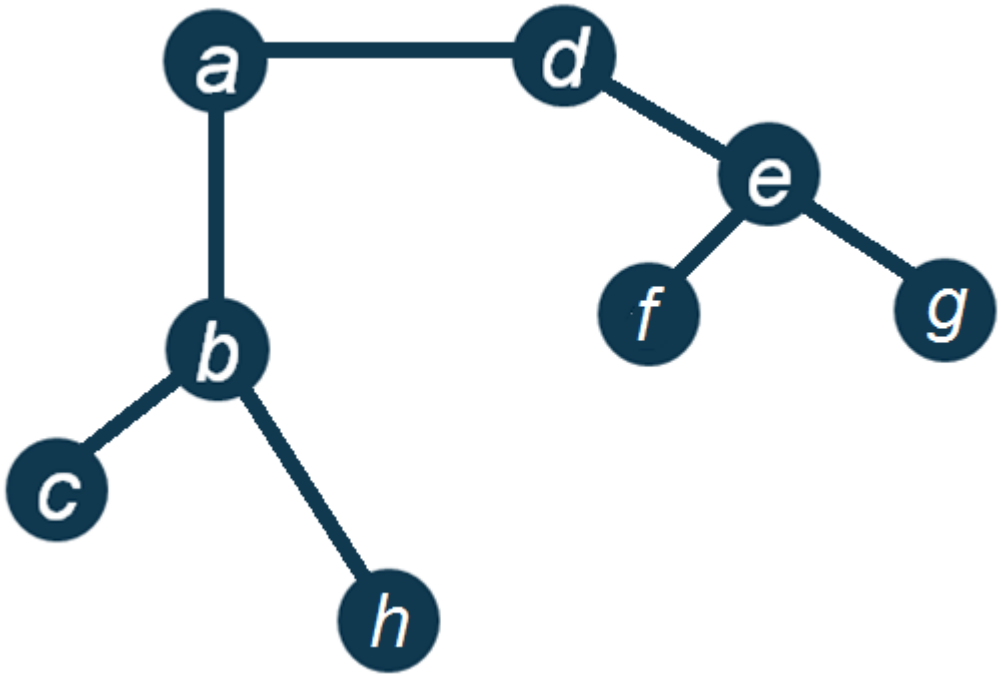
Seriation algorithms PART 2

The following seriation algorithms based on minimizing the sum of distances between laboratory parameters are available in e**lab**orator:

// “VAT” → **V**isual **A**ssessment of **T**endency

// “TSP” → **T**ravelling **S**alesperson **P**roblem

VAT algorithm (Visual Assessment of Tendency)

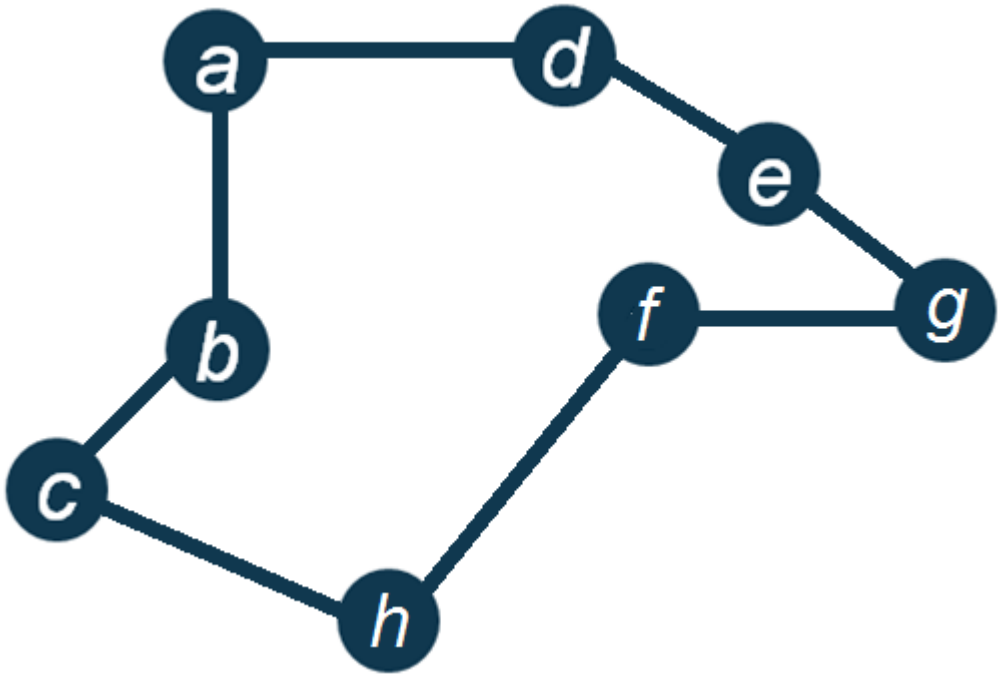


- Rough idea:
  - Finding a path that connects all lab parameters while minimizing the sum of the distances (minimal spanning tree)

$$c - b - h - a - d - e - f - g$$

Bezdek, J.C. and Hathaway, R.J. (2002): VAT: a tool for visual assessment of (cluster) tendency. *Proceedings of the 2002 International Joint Conference on Neural Networks* (IJCNN ’02) 3, 2225–2230

TSP algorithm (Travelling Salesperson Problem)



- Rough idea:
  - Finding a path that connects all lab parameters while minimizing the sum of the distances with restriction that each lab parameter can only be ‚visited' once

$$h - c - b - a - d - e - g - f$$

Hahsler M, Hornik K (2007). TSP - Infrastructure for the Traveling Salesperson Problem. *Journal of Statistical Software*, 23(2), 1-21.

Seriation algorithms PART 3

The following seriation algorithms are based seriating lab parameters in such a way that for each lab parameter the more proxy a lab parameter is located, the more dissimilar it is:

// „ARSA“ → **A**nti-**R**obinson seriation by **S**imulated **A**nnealing

// “BBURCG” → **B**ranch & **B**ound **U**nweighted **R**ow and **C**olumn **G**radient

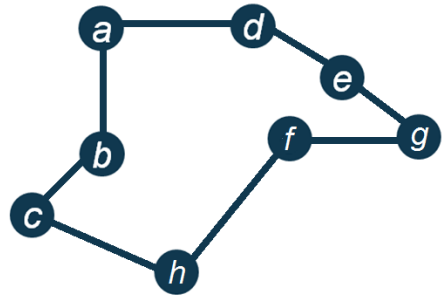
// “BBWRCG” → **B**ranch & **B**ound **W**eighted **R**ow and **C**olumn **G**radient

Note: For very large numbers of lab parameters runtimes might be very large for branch & bound algorithms.

ARSA, BBURCG, BBWRCG

- Rough idea:
  - For any possible seriation count the no. of violations.
  - Find the seriation with the smallest number of violations from this concept.
  - The magnitude of violations can be ignored (**U**nweighted, i.e. BBURCG; ARSA) or taken into account (**W**eighted, i.e. BBWRCG).

- Examples for violations of the seriation $h - c - b - a - d - e - g - f$:
  - e and f are more similar than f and g but are more proxy
  - f and d are more similar than d and a but are more proxy
  - f and d are more similar than d and b but are more proxy
  - …



ARSA, BBURCG, BBWRCG

- Rough idea (cont.):
  - In practice there are many possible seriations ($\frac{factorial(no.\,lab\ parameters)}{2}$) such that not all can be evaluated. So-called ‚Partial enumeration methods‘ are used:
    - branch & bound (BBURCG, BBWRCG)
    - simulated annealing (ARSA) [heuristic used for large no. lab parameters]

- Examples for violations of the seriation $h - c - b - a - d - e - g - f$:
  - e and f are more similar than f and g but are more proxy
  - f and d are more similar than d and a but are more proxy
  - f and d are more similar than d and b but are more proxy
  - …