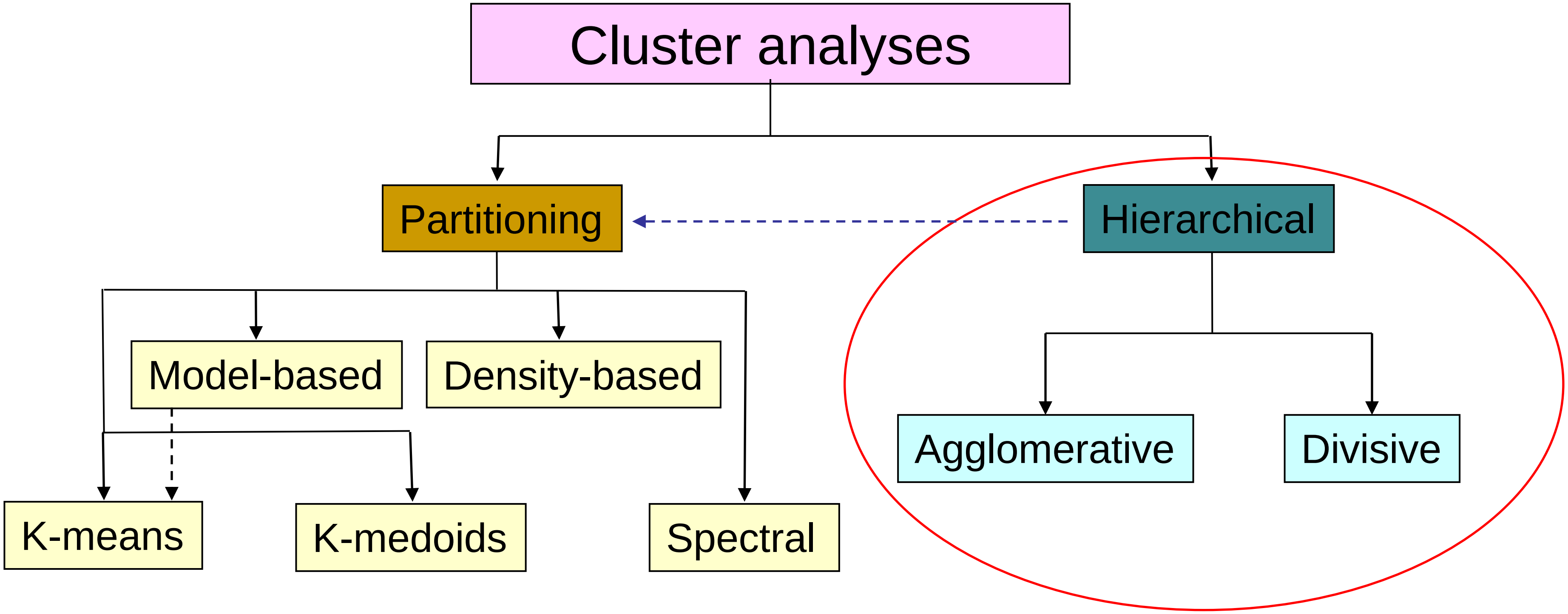
Cluster analysis II

Hierarchical clustering

Radoslav Harman
KAMS FMFI UK

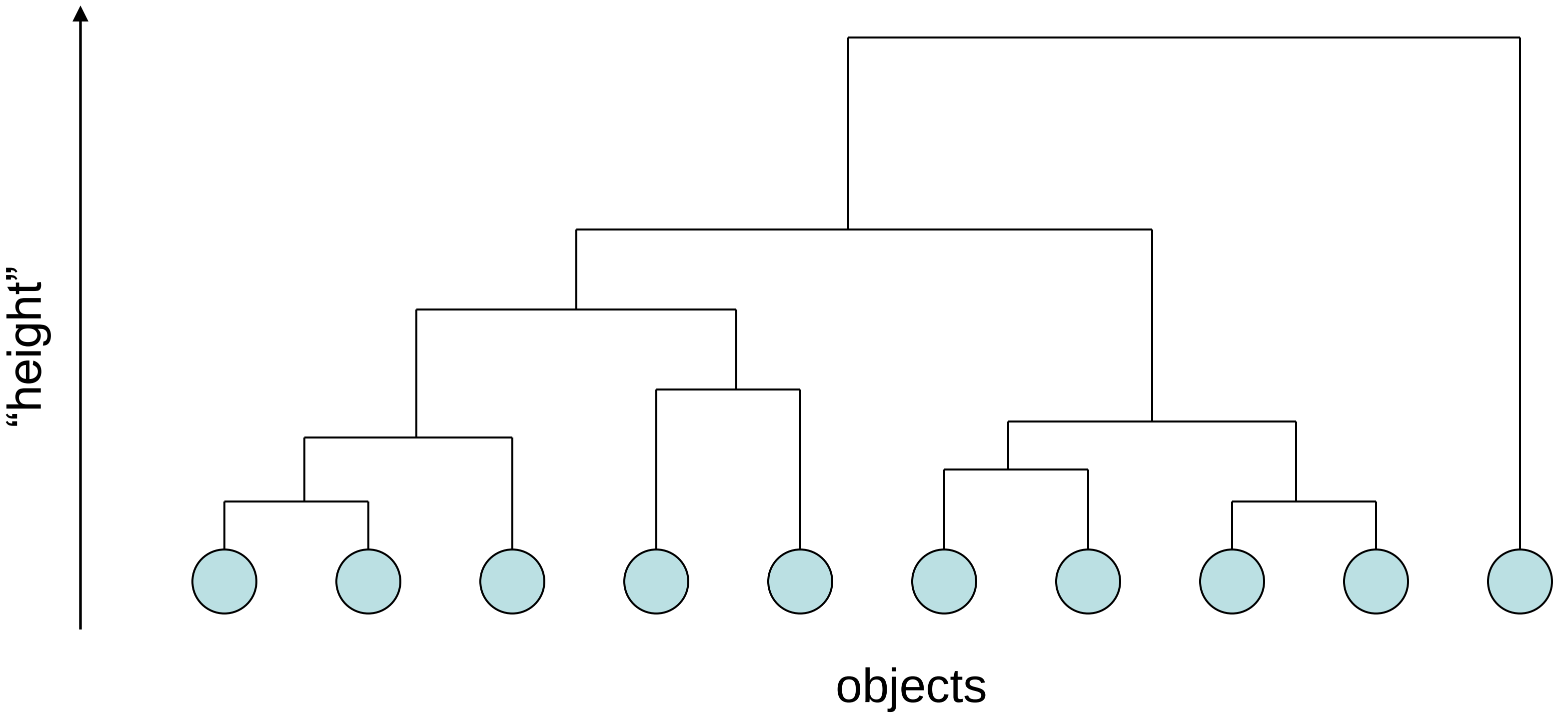
Structure of cluster analyses



Hierarchical clustering: Produces a "**hierarchy of clusters**" visualized by a "**dendrogram".**

Compared to partitioning methods: Generally **slower** computation, but more **informative** output. **Dissimilarities** can typically be used directly. Defined by an **algorithm**, not an objective.

Example of a dendrogram



The dendrogram is created either:

•„bottom-up“ (**aglomerative**, or ascending, clustering), or

•„top-down“ (**divisive**, or descending, clustering).

Agglomerative clustering

Algorithm:

- Create the initial set of „top-level“ (“active”) clusters: formed by individual objects (each object forms an individual top-level cluster).

- While there are more than one top-level clusters do:

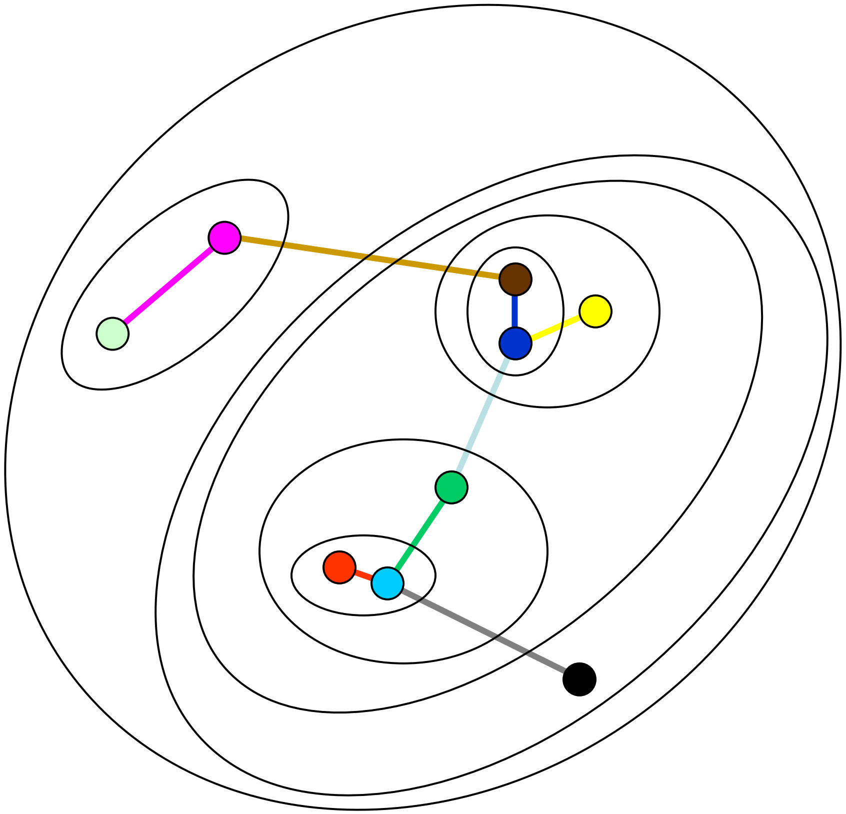
  - Find the two top-level clusters with the smallest **mutual intercluster distance** and join them into a new top-level cluster. (The two clusters that have been joined cease to be top-level clusters.)

Different measures of distance between clusters provide different variants: Single linkage, Complete linkage, Average linkage, Ward’s distance, ...

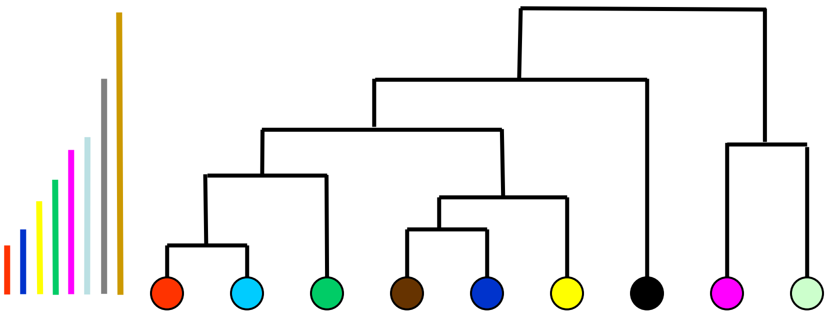
Single linkage in agglomerative clustering

- The distance of two clusters is the dissimilarity of the least dissimilar objects of the clusters:

$$D_S(C_i, C_j) = \min_{r \in C_i, s \in C_j} d(r, s)$$



Dendrogram

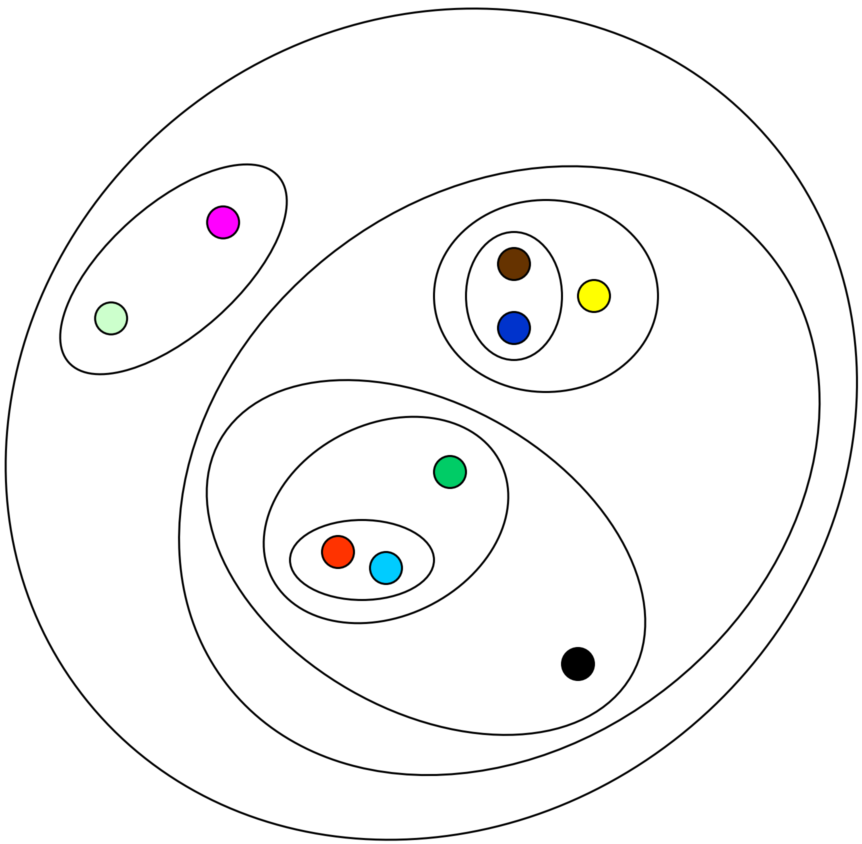


Note that we effectively constructed a minimum-weight spanning tree. The “naïve” implementation of the single-linkage clustering = the **Kruskall’s algorithm**.

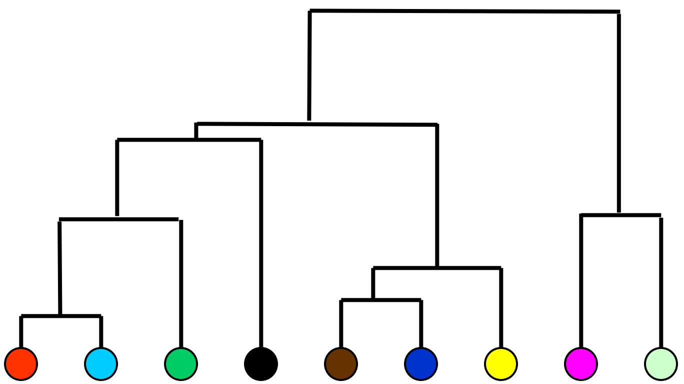
Average linkage in agglomerative clustering

- The distance of two clusters is the average of mutual dissimilarities of the objects in the clusters:

$$D_A(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{r \in C_i} \sum_{s \in C_j} d(r,s)$$



Dendrogram



Other methods of measuring a distance of clusters in agglomerative clustering

- **Complete linkage**: the distance of clusters is the dissimilarity of the most dissimilar objects:

$$D_C(C_i, C_j) = \max_{r \in C_i, s \in C_j} d(r, s)$$

- **Ward's distance**: Requires that for each object r we have the real vector of features x_r. (The matrix of dissimilarities is not enough.) It is the difference between "an extension" of the two clusters combined and the sum of the "extensions" of the two individual clusters.

$$D_W(C_i, C_j) = \sum_{m \in C_i \cup C_j} \rho^2(x_m, c_{ij}) - \left[\sum_{r \in C_i} \rho^2(x_r, c_i) + \sum_{s \in C_j} \rho^2(x_s, c_j) \right]$$

$c_{ij}, c_i, c_j \ldots$ the centroids of $C_i \cup C_j, C_i, C_j$

$\rho \ldots$ the distance between vectors

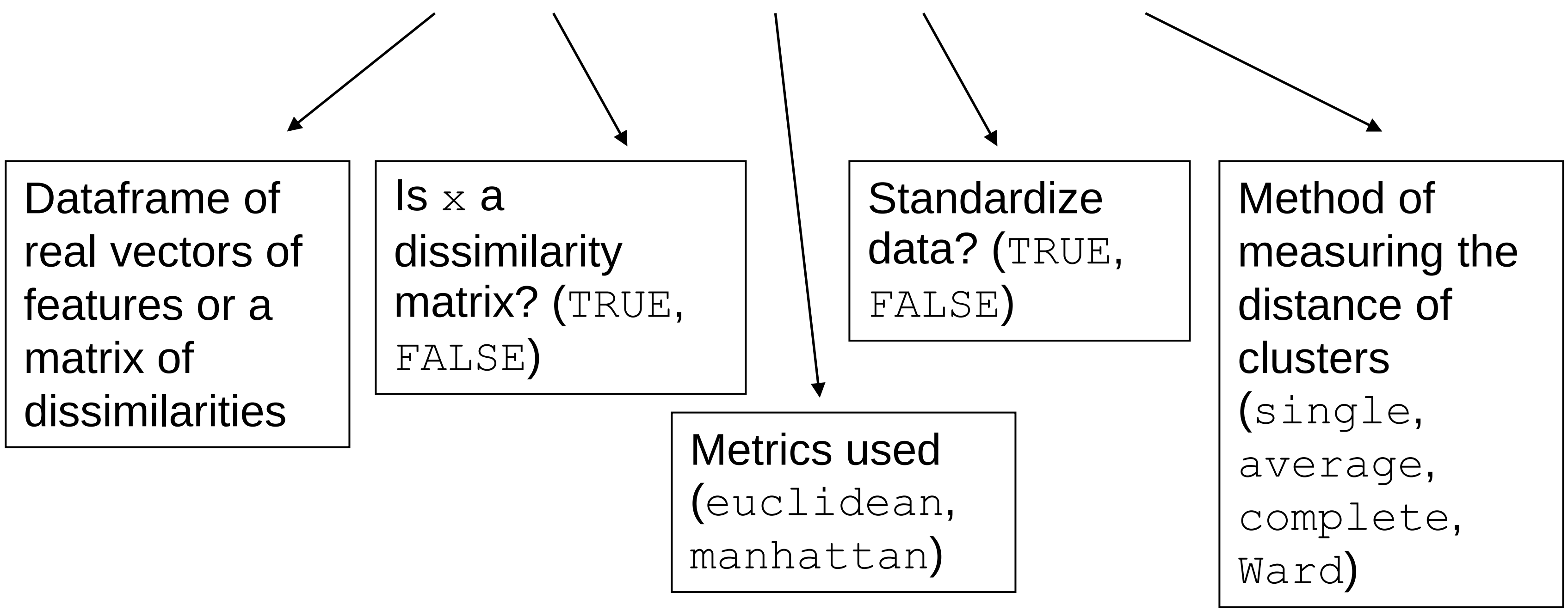
- Gazillion of other linkages

Computational issues of agglomerative clustering

- **Complexity:** At least quadratic complexity with respect to the number of objects. Naïve implementation has cubic complexity.

In R (library `cluster`):

```
agnes(x, diss, metric, stand, method, …)
```



Divisive clustering

Algorithm (Specification by Kaufman and Rousseeuw):

- Form a single cluster consisting of all objects.
- For each “bottom level” (“active”) cluster with at least two objects:
  - Find the “most eccentric” object that initiates a “splinter group”. (The object that has maximal average dissimilarity to other objects.)
  - Find all objects in the cluster that are more similar to the “most eccentric” object than to the rest of the objects. (For instance, the objects that have higher average dissimilarity to the eccentric object than to the rest of the objects.)
  - Divide the cluster into two subclusters accordingly.
- Continue until all “bottom level” clusters consist of a single object.

Divisive clustering in general is similar to the hierarchical clustering for the nodes of a network, for instance the **Girvan-Newman algorithm**, which sequenatially removes the edges of a network which have the maximum „edge betweeness“.

Illustration of the divisive clustering

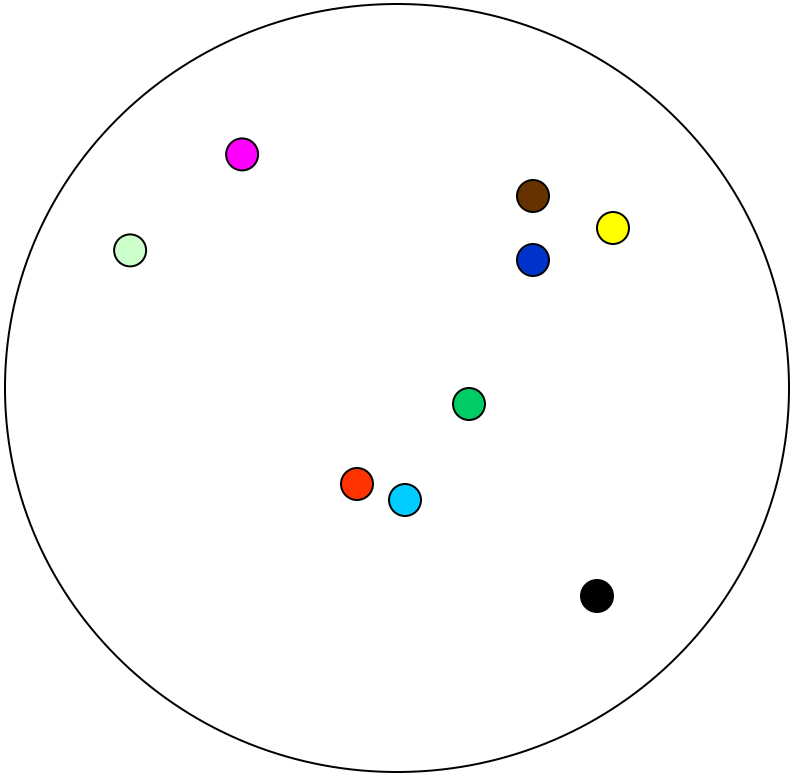


Illustration of the divisive clustering

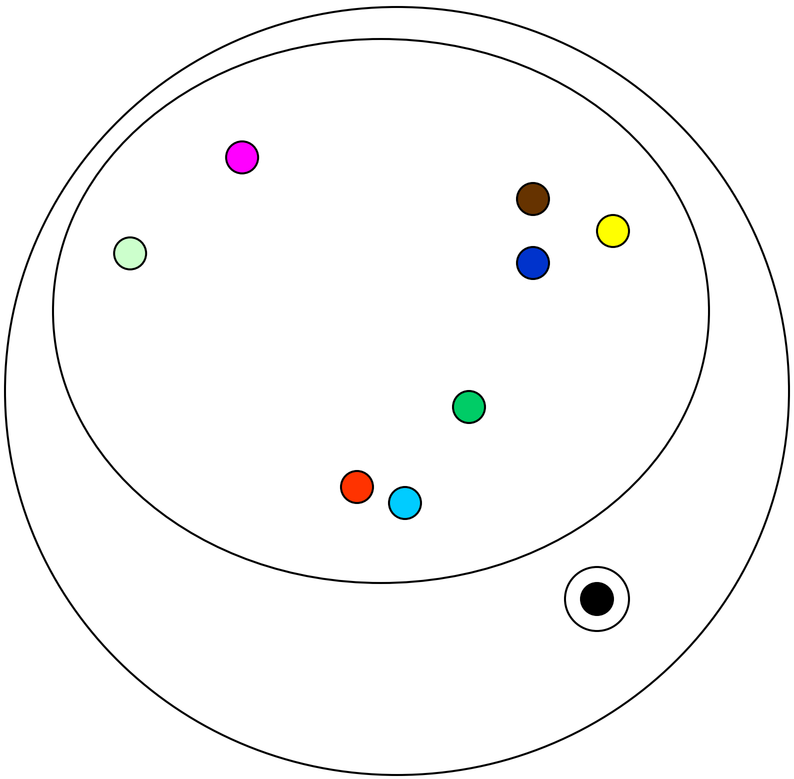


Illustration of the divisive clustering

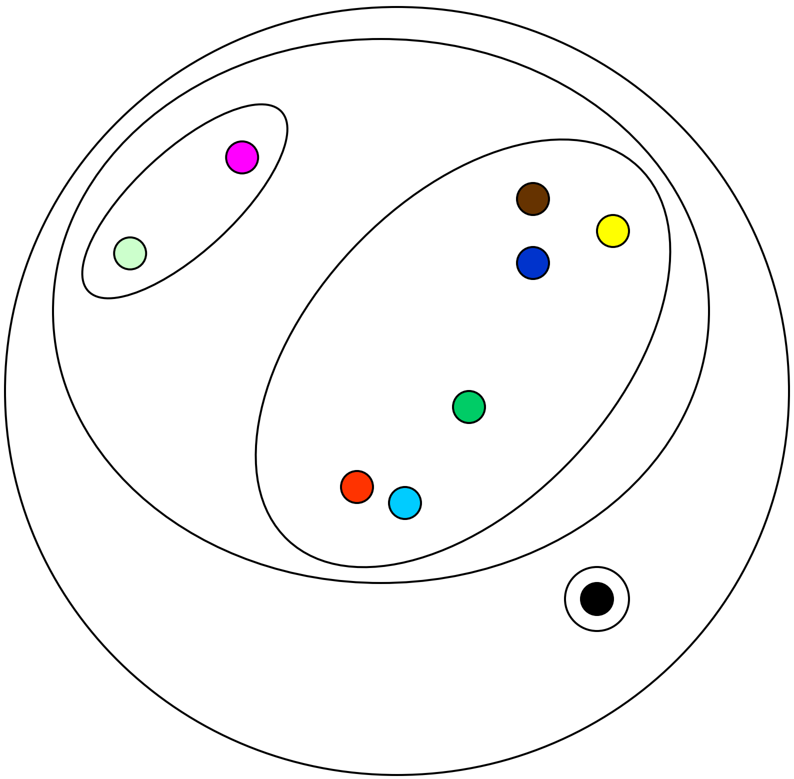


Illustration of the divisive clustering

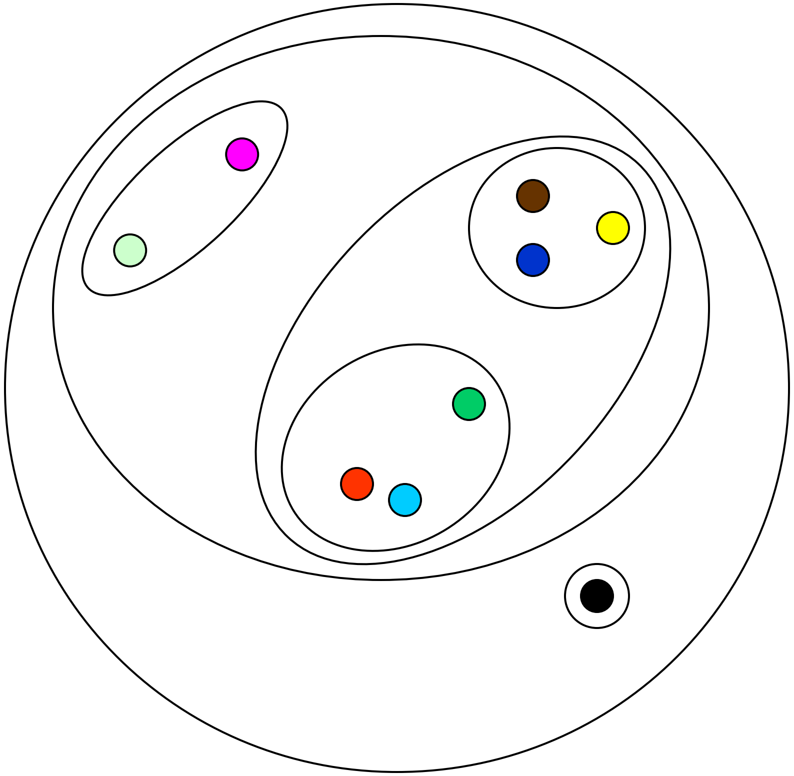


Illustration of the divisive clustering

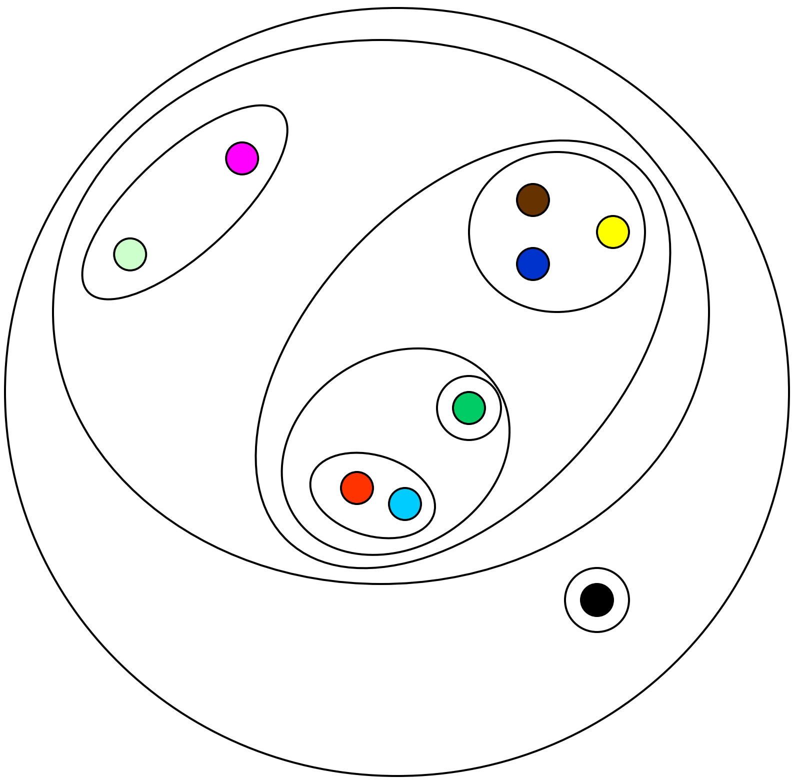


Illustration of the divisive clustering

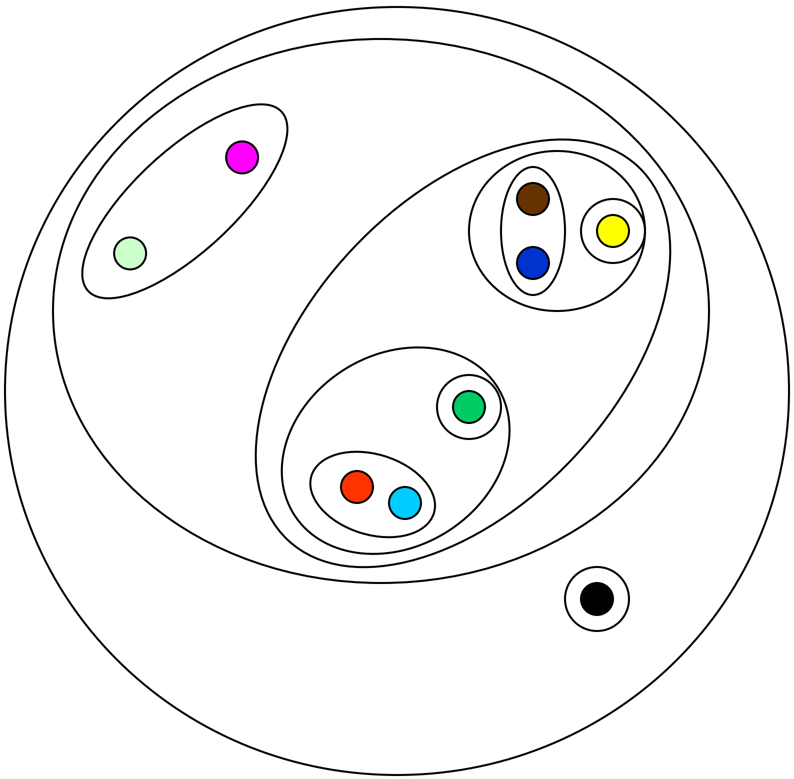


Illustration of the divisive clustering

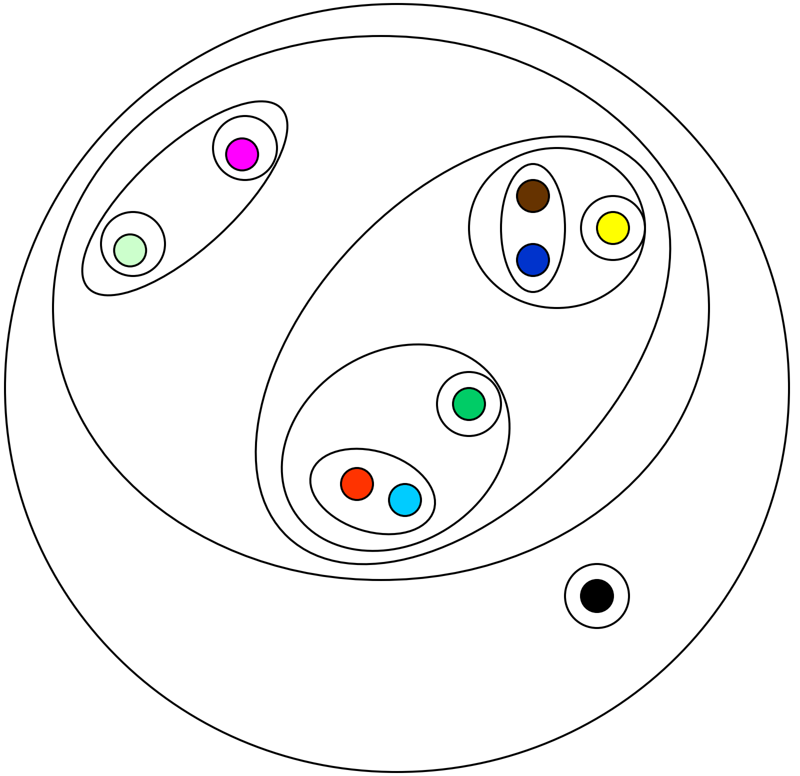


Illustration of the divisive clustering

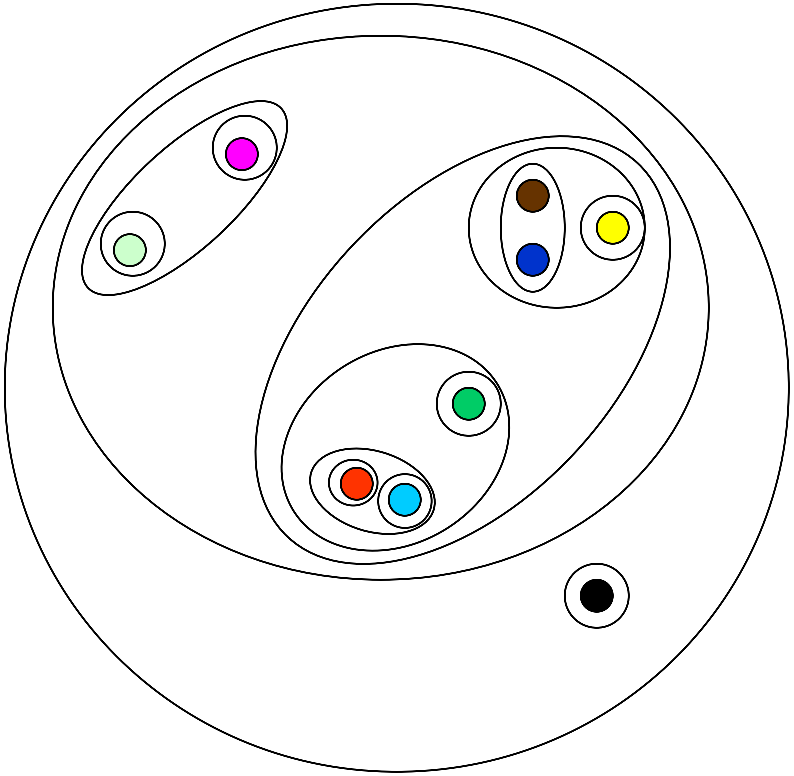


Illustration of the divisive clustering

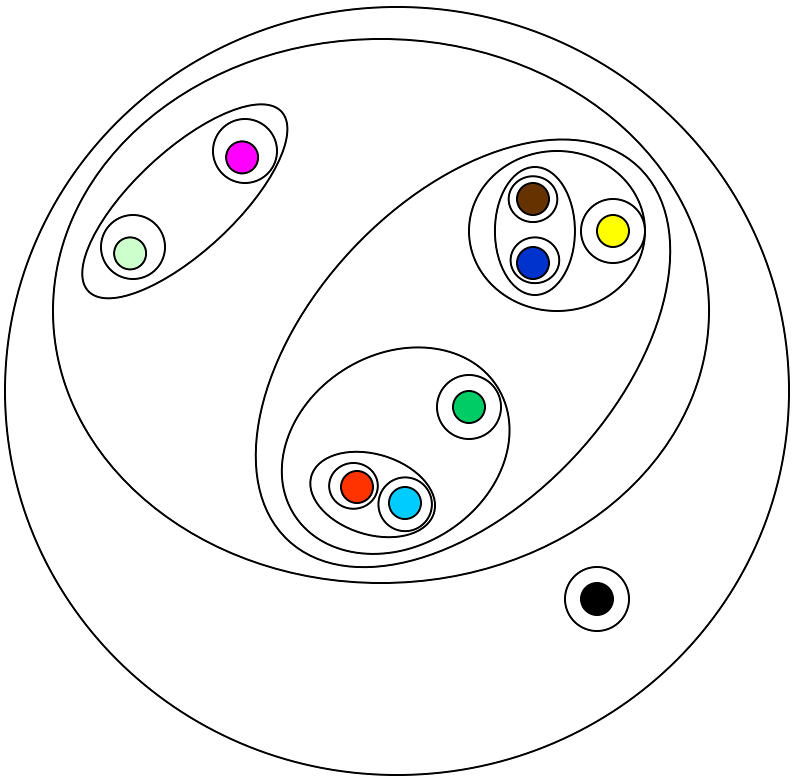
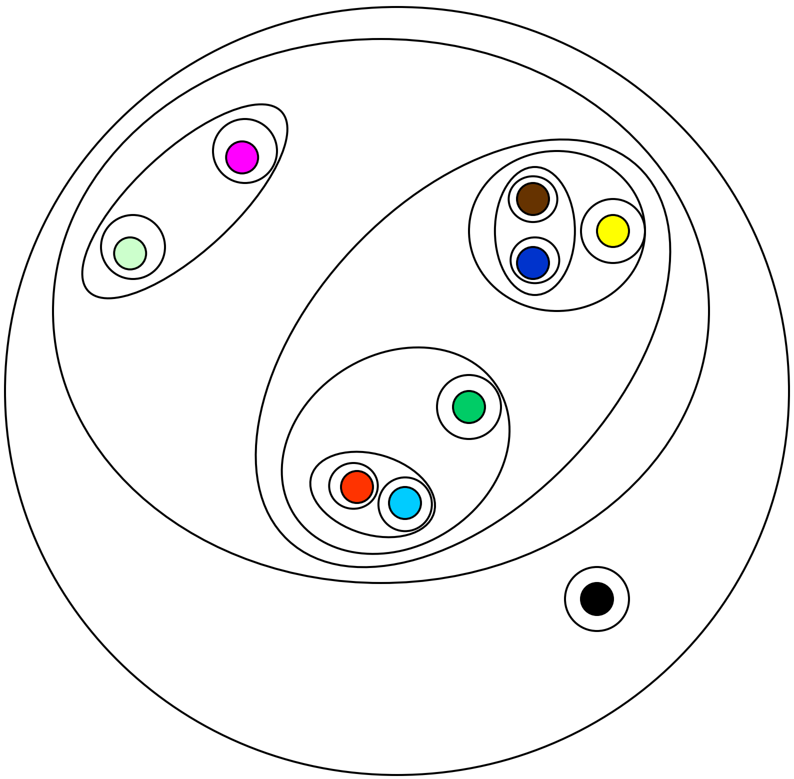
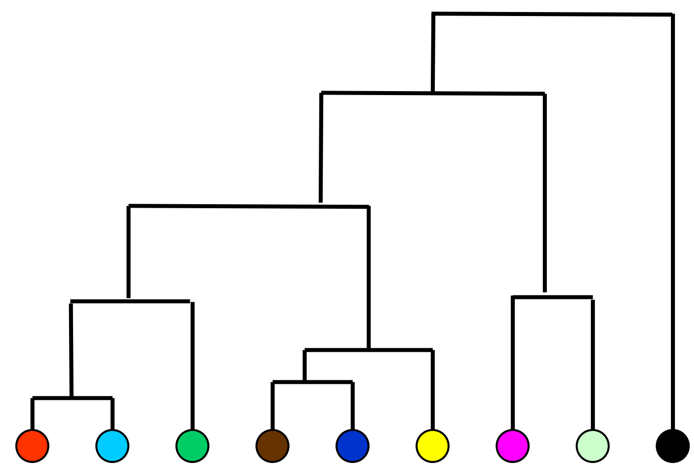


Illustration of the divisive clustering



Dendrogram



Computational issues of divisive clustering

- **Complexity:** At least linear with respect to the number of objects (depending on implementation and a on the kind of the „splitting subroutine“).

In R (library `cluster`):

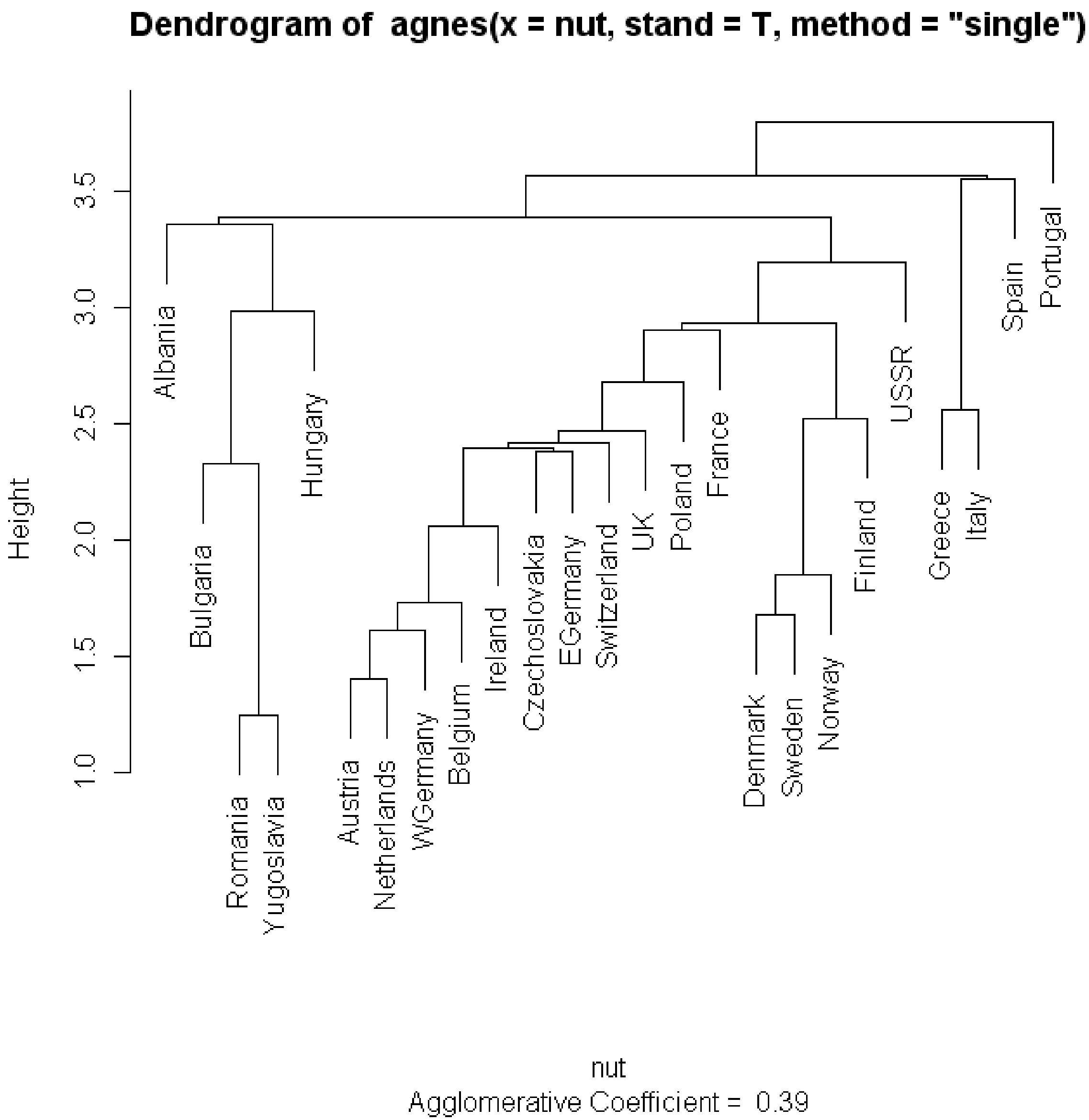
```
diana(x, diss, metric, stand, …)
```

- `x`: Dataframe of real vectors of features or a matrix of dissimilarities
- `diss`: Is `x` a dissimilarity matrix? (`TRUE`, `FALSE`)
- `metric`: Metrics used (`euclidean`, `manhattan`)
- `stand`: Standardize data? (`TRUE`, `FALSE`)

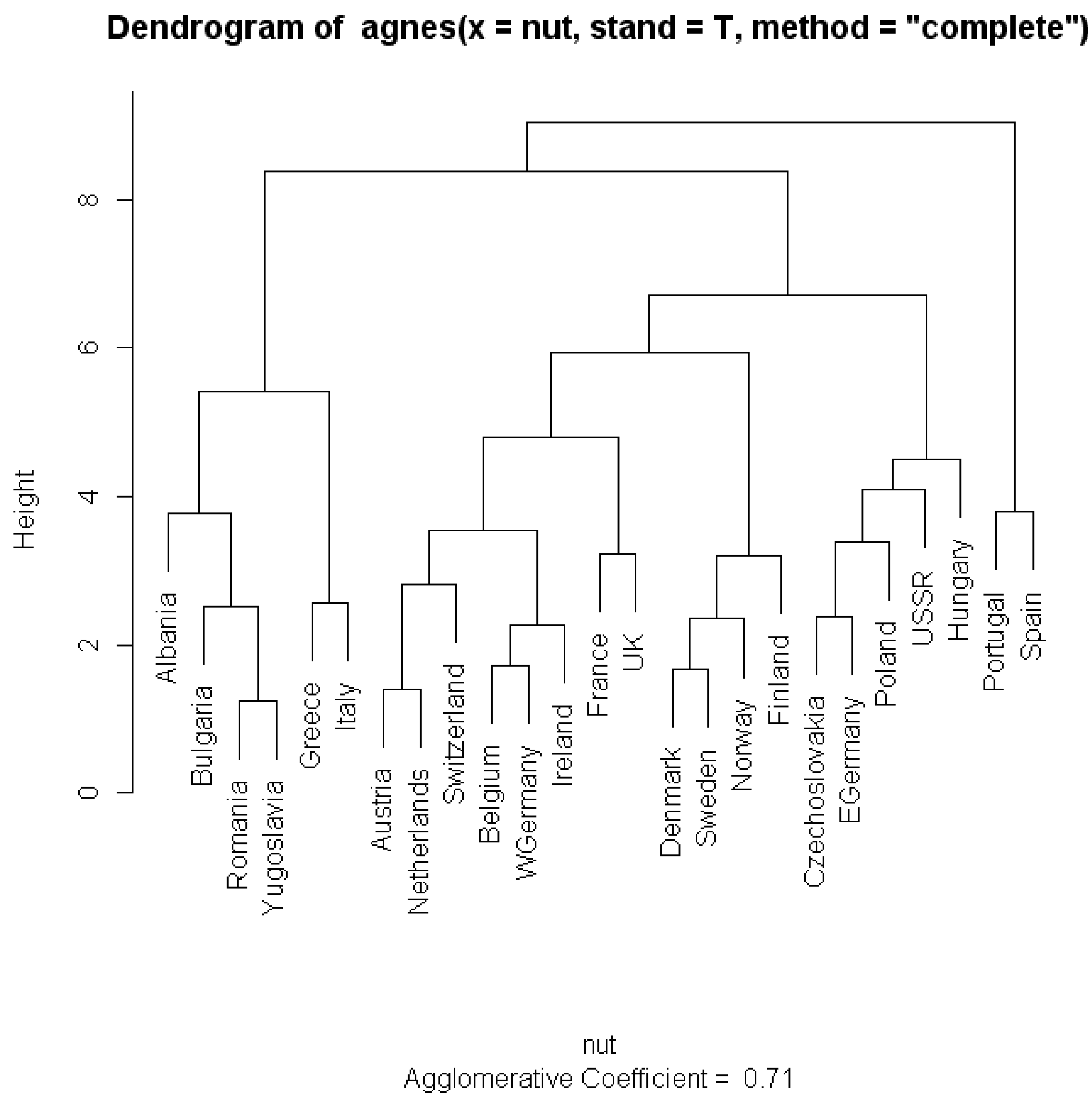
Comparison of hierarchical clustering methods

- n=25 objects - European countries (`Albania, Austria, Belgium, Bulgaria, Czechoslovakia, Denmark, EGermany, Finland, France, Greece, Hungary, Ireland, Italy, Netherlands, Norway, Poland, Portugal, Romania, Spain, Sweden, Switzerland, UK, USSR, WGermany, Yugoslavia`)
- p=9 dimensional vectors of features - consumption of various kinds of food (`Red Meat, White Meat, Eggs, Milk, Fish, Cereals, Starchy foods, Nuts, Fruits/Vegetables`)

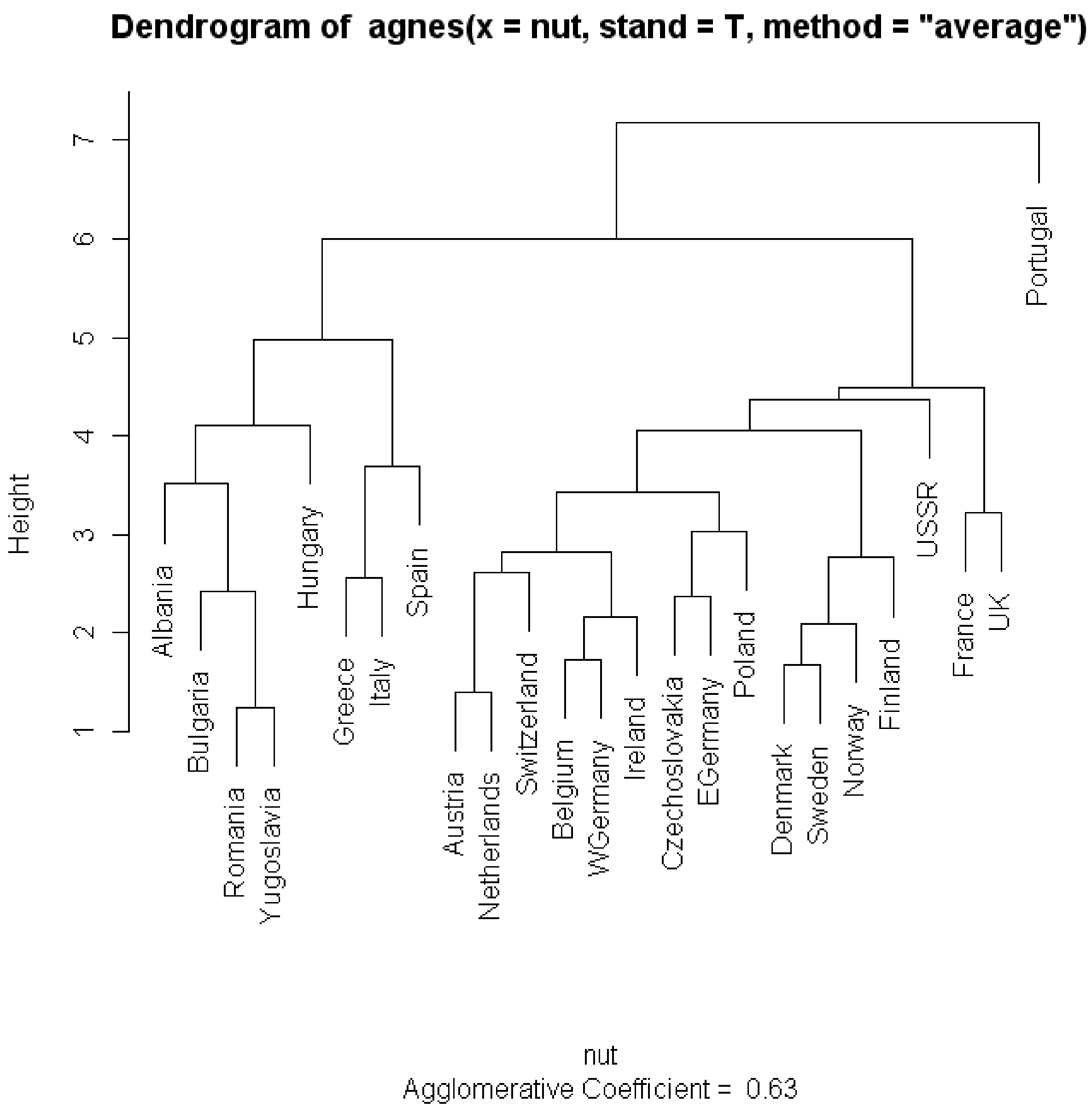
Agglomerative - single linkage



Agglomerative - complete linkage



Agglomerative - average linkage



Divisive clustering