Classification method in single particle analysis

Cluster Analysis

Pawel A. Penczek
Pawel.A.Penczek@uth.tmc.edu
The University of Texas – Houston Medical School

Overview

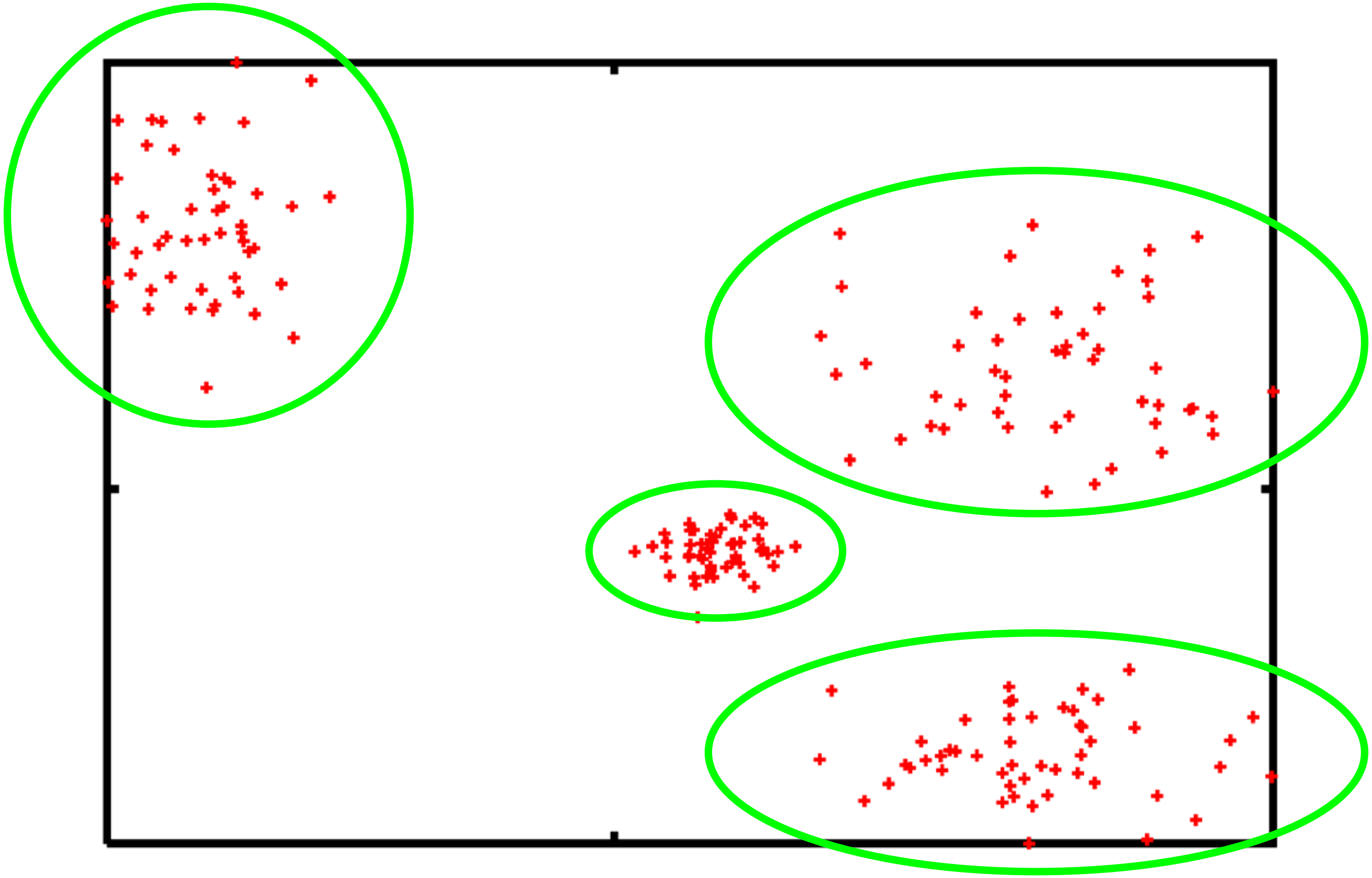
- Background
- Hierarchical Methods
- *K*-Means
- Clustering in single particle analysis
- Structure determination in EM as a classification problem

Background

- *Clustering* is the process of identifying natural groupings in the data
- *Unsupervised learning* technique
  - No predefined class labels
- Classic text is *Finding Groups in Data* by Kaufman and Rousseeuw, 1990
- Two types: (1) hierarchical, (2) *K*-Means

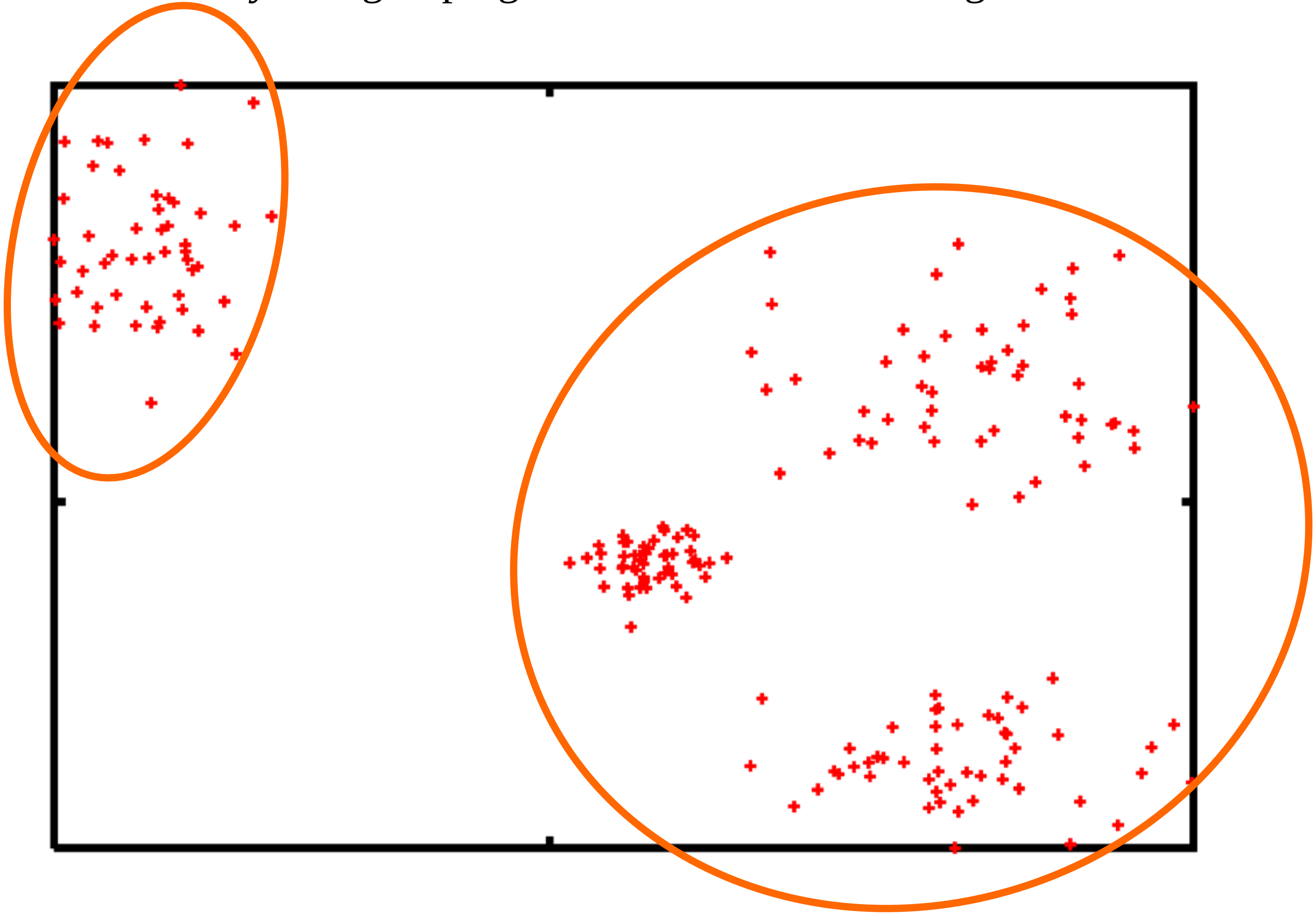
What is a cluster?

Cluster analysis – grouping of the data set into homogeneous classes.



What is a cluster?

Cluster analysis – grouping of the data set into homogeneous classes.



Two unresolved questions.

1. What is a cluster?

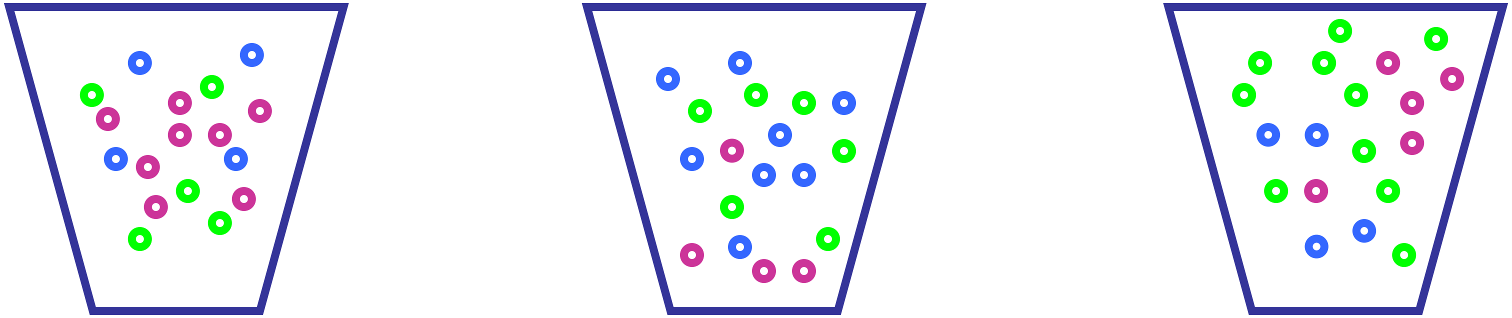
   Lack of a mathematical definition, can vary from one application to another.

2. How many clusters there are?

   Depends on the adopted definition of a cluster, also on the preference of the user.

Clustering is an intractable problem.

Distribute *n* distinguishable objects into *k* urns.

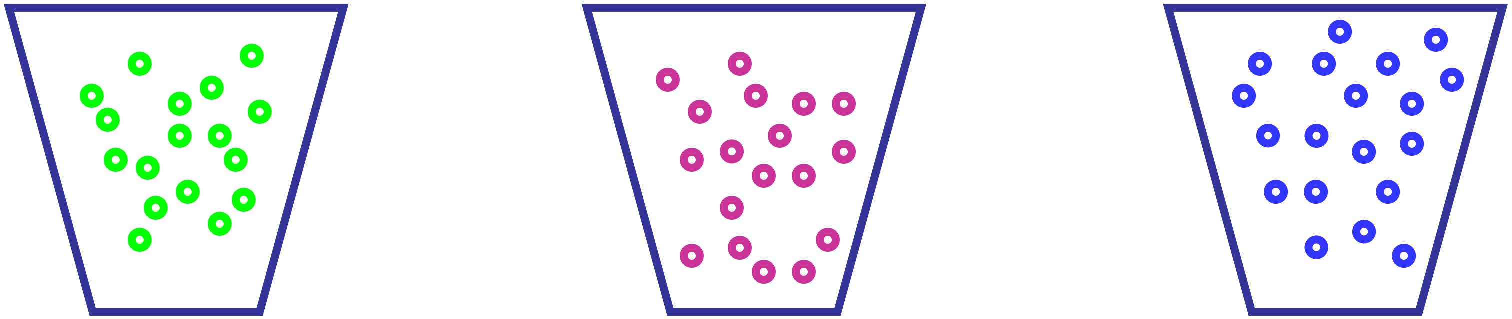


$k^{\boldsymbol{n}}$ possibilities.

If k=3 and n=100, the number of combinations is ~10^{47}!

Clustering is an intractable problem.

Distribute *n* distinguishable objects into *k* urns.

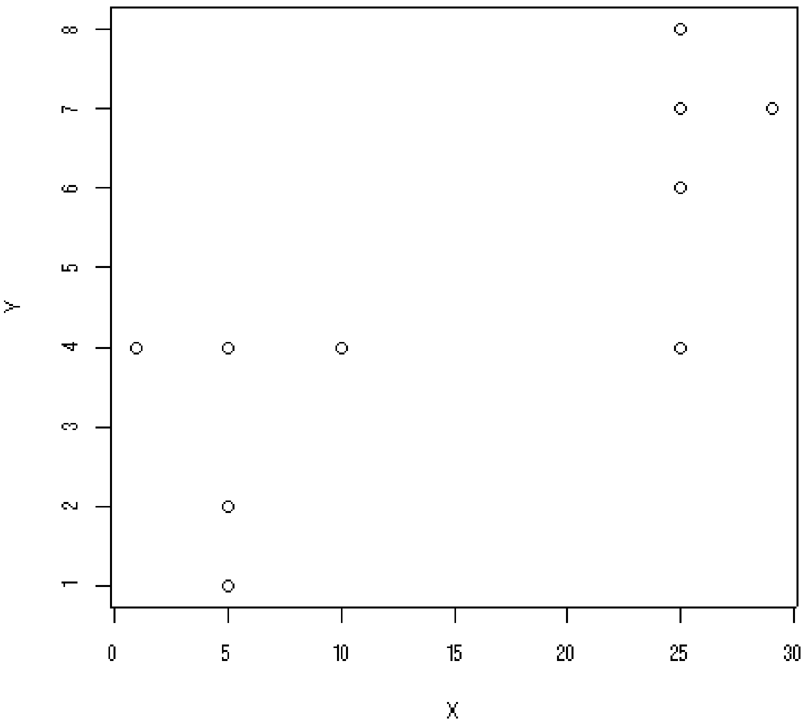


$k^{\boldsymbol{n}}$ possibilities.

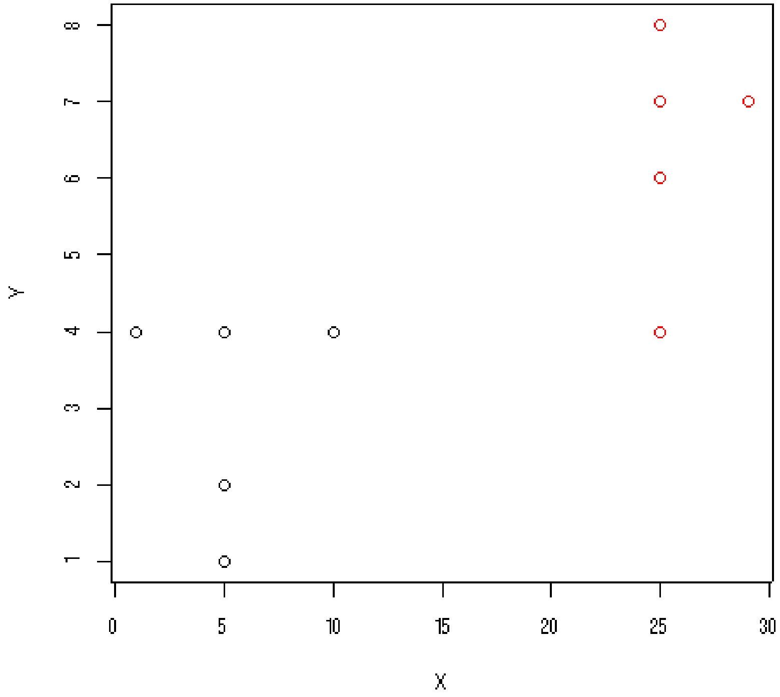
If *k*=3 and *n*=100, the number of combinations is ~10^{47}!

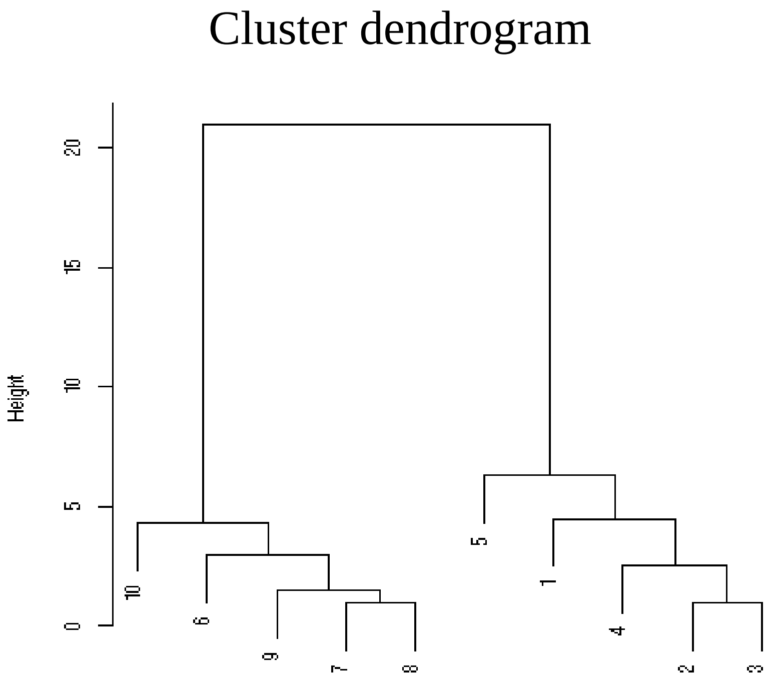
Clustering

X	Y
1	4
5	1
5	2
5	4
10	4
25	4
25	6
25	7
25	8
29	7

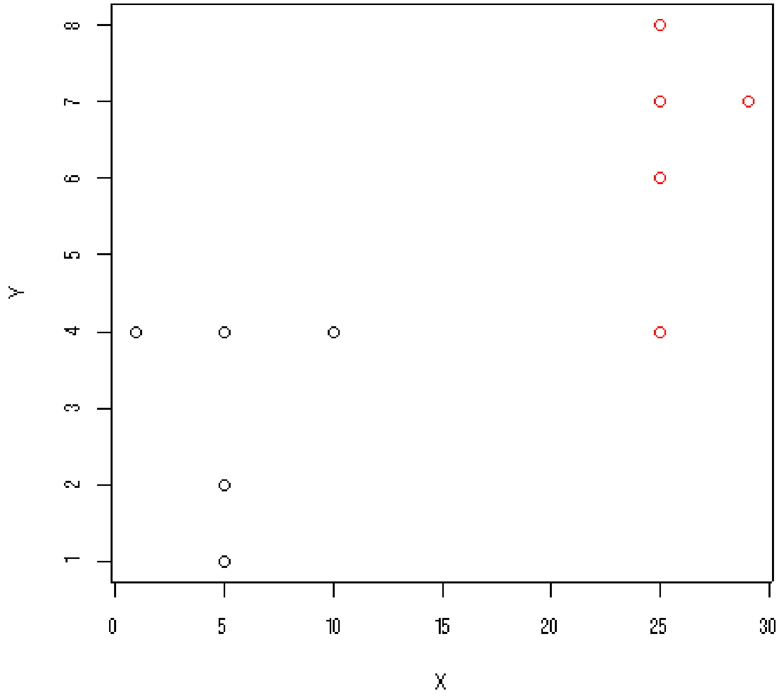


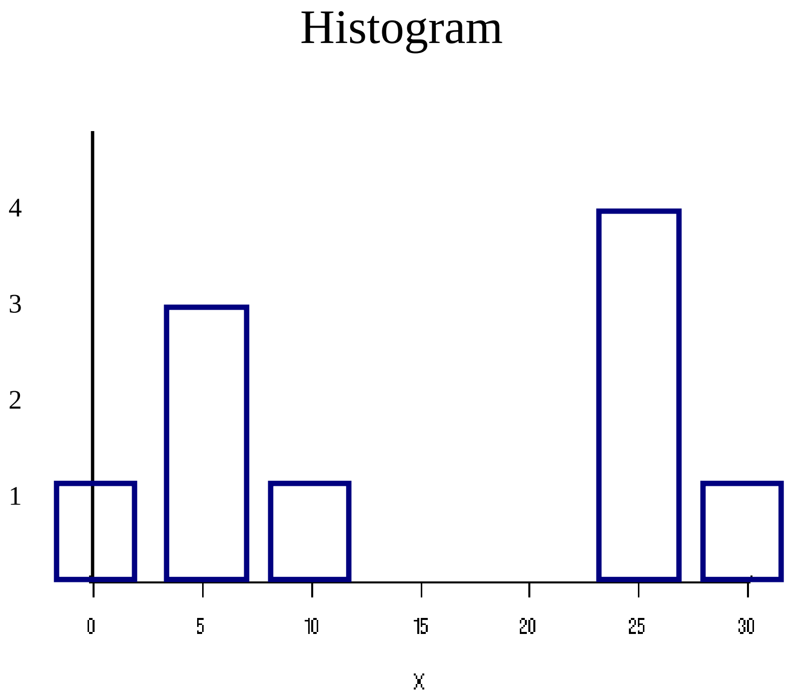
Visualizations



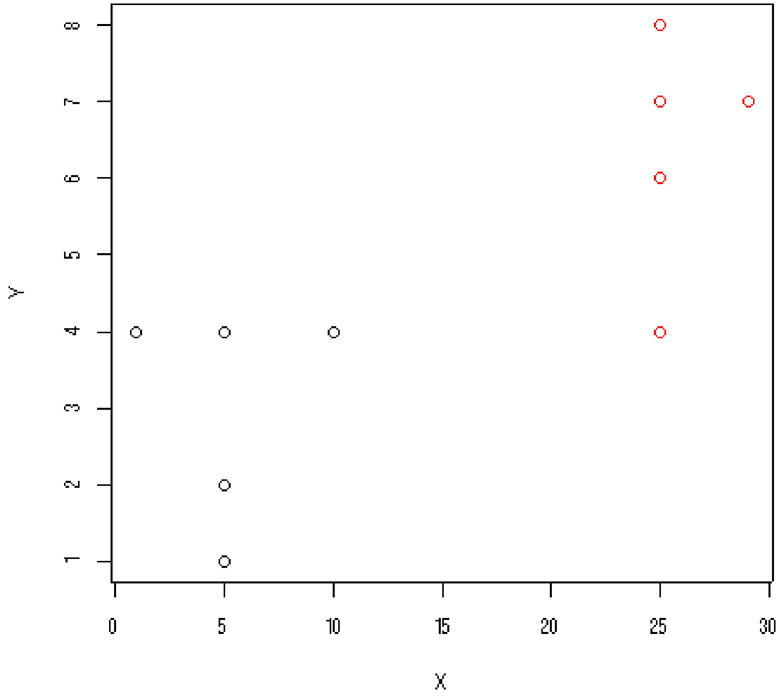

Cluster dendrogram

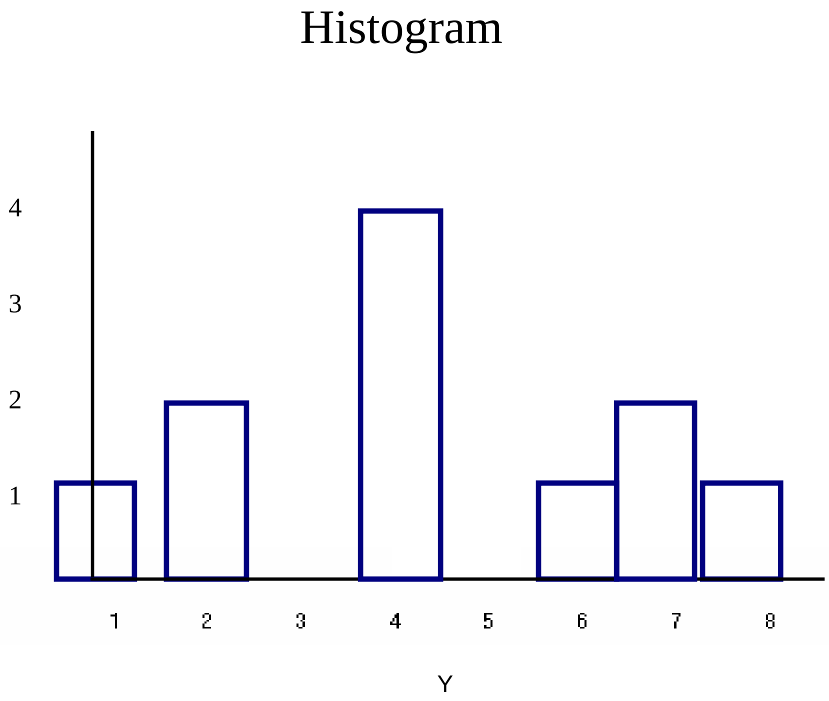
Visualizations





Visualizations





Data available in the form of pair-wise ‘dissimilarities’

- Hierarchical clustering algorithms use a *dissimilarity matrix* as input

	Ford Escort	Nissan Xterra	Land Rover	Honda Accord	Ford Mustang
Ford Escort		different	different	similar	different
Nissan Xterra			similar	different	different
Land Rover				different	different
Honda Accord					different
Ford Mustang					

Hierarchical Methods

- Top-down (descendant)
- Bottom-up (ascendant)

Top-Down vs. Bottom-Up

- Top-down or *divisive* approaches split the whole data set into smaller pieces
- Bottom-up or *agglomerative* approaches combine individual elements

Agglomerative Nesting

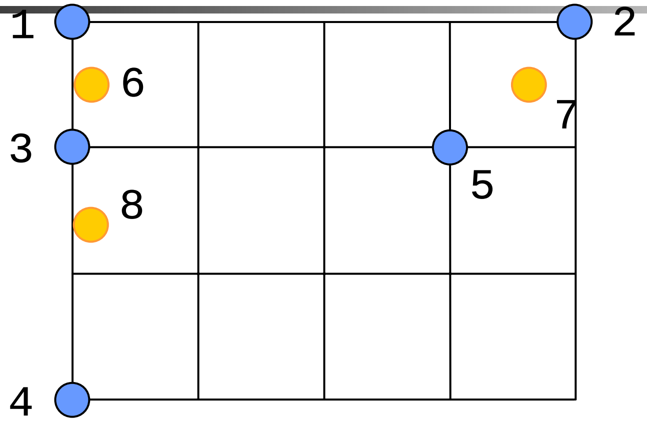
- Combine clusters until one cluster is obtained
  - Initially each cluster contains one object
  - At each step, select the two “most similar” clusters

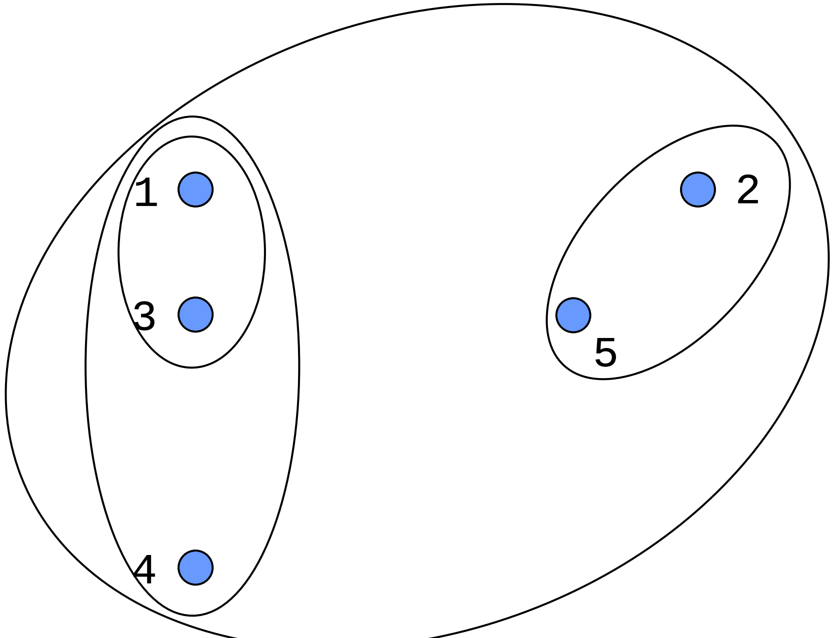
$$d(R,Q) = \frac{1}{|R||Q|} \sum_{\substack{i \in R \\ j \in Q}} diss(i, j)$$

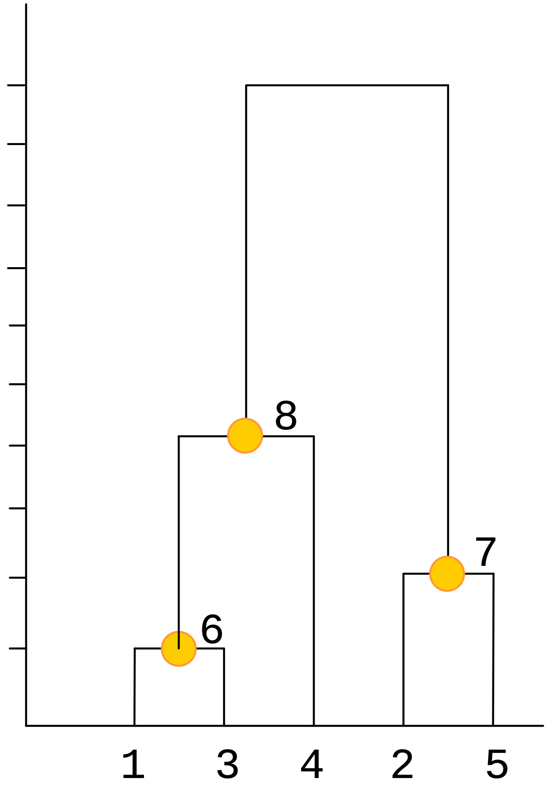
Hierarchical ascendant clustering

```
Algorithm: HAC
Input:    D      the matrix of pair-wise dissimilarities
Output:   Tree   a dendrogram

Assign each of $N$ objects to its own class
For $k$=2 to $N$ do
    Find the closest (most similar) pair of clusters and merge them into a single cluster;
    Store the information about merged cluster and merging threshold in a dendrogram;
    Compute distances (similarities) between the new cluster and each of the old clusters;
Enddo
```

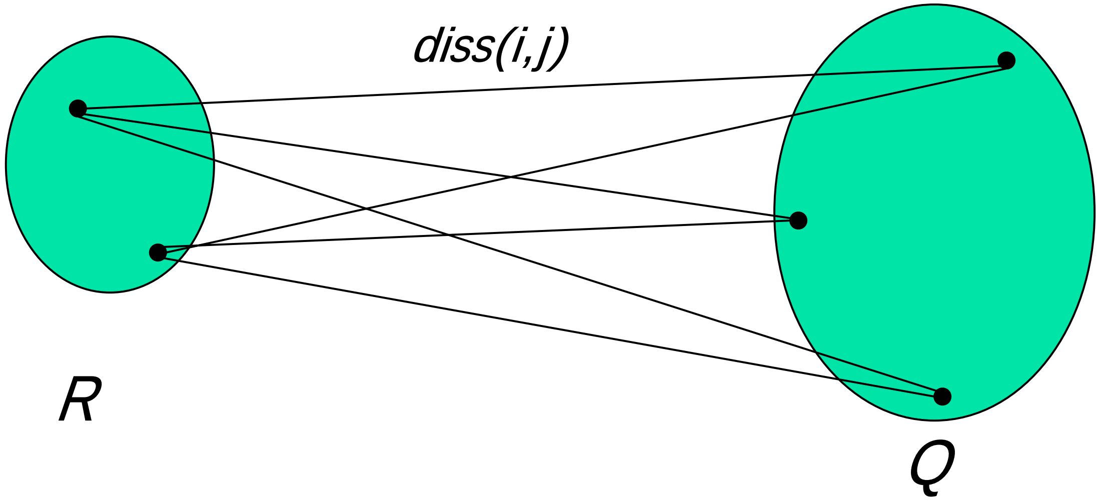
Hierarchical Ascendant Classification
Agglomerative







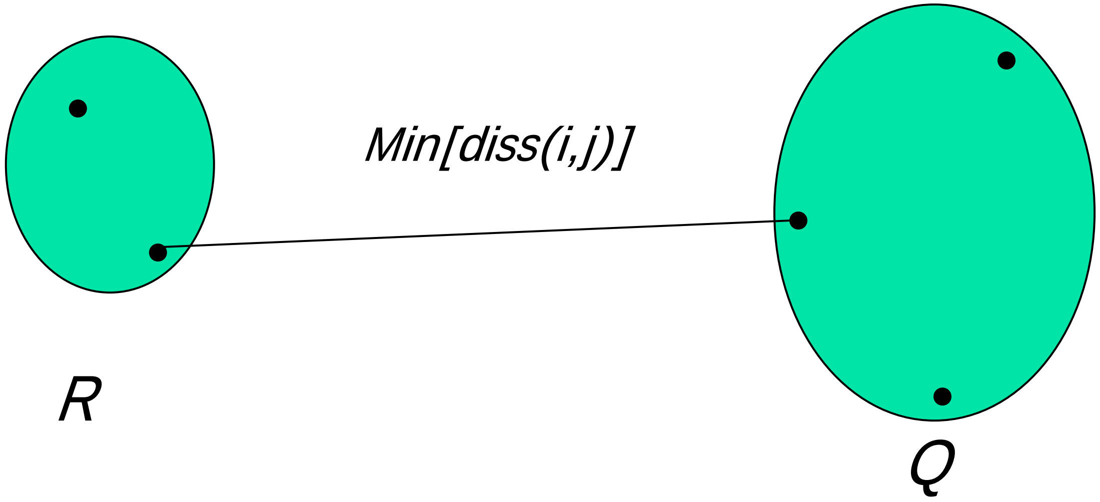
Cluster Dissimilarities



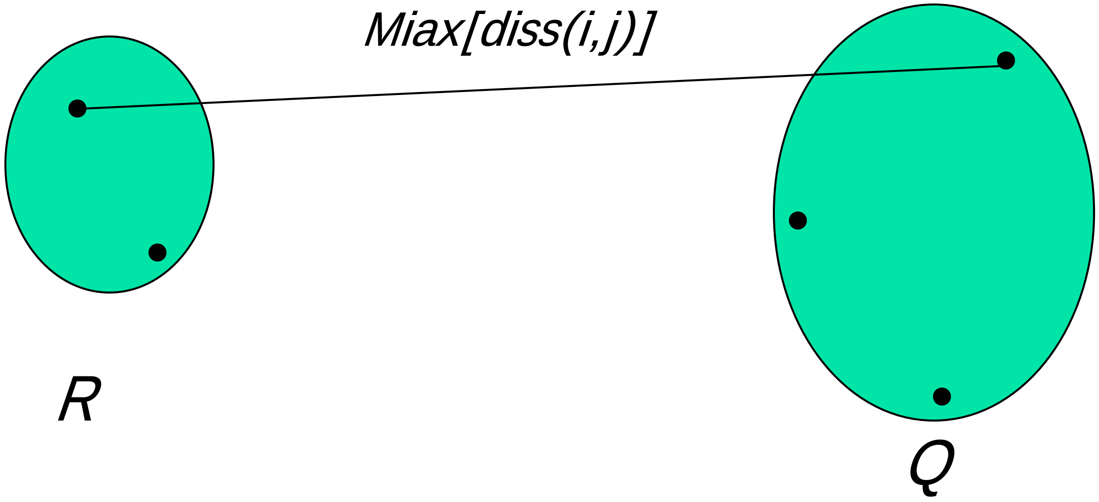
Merging criteria

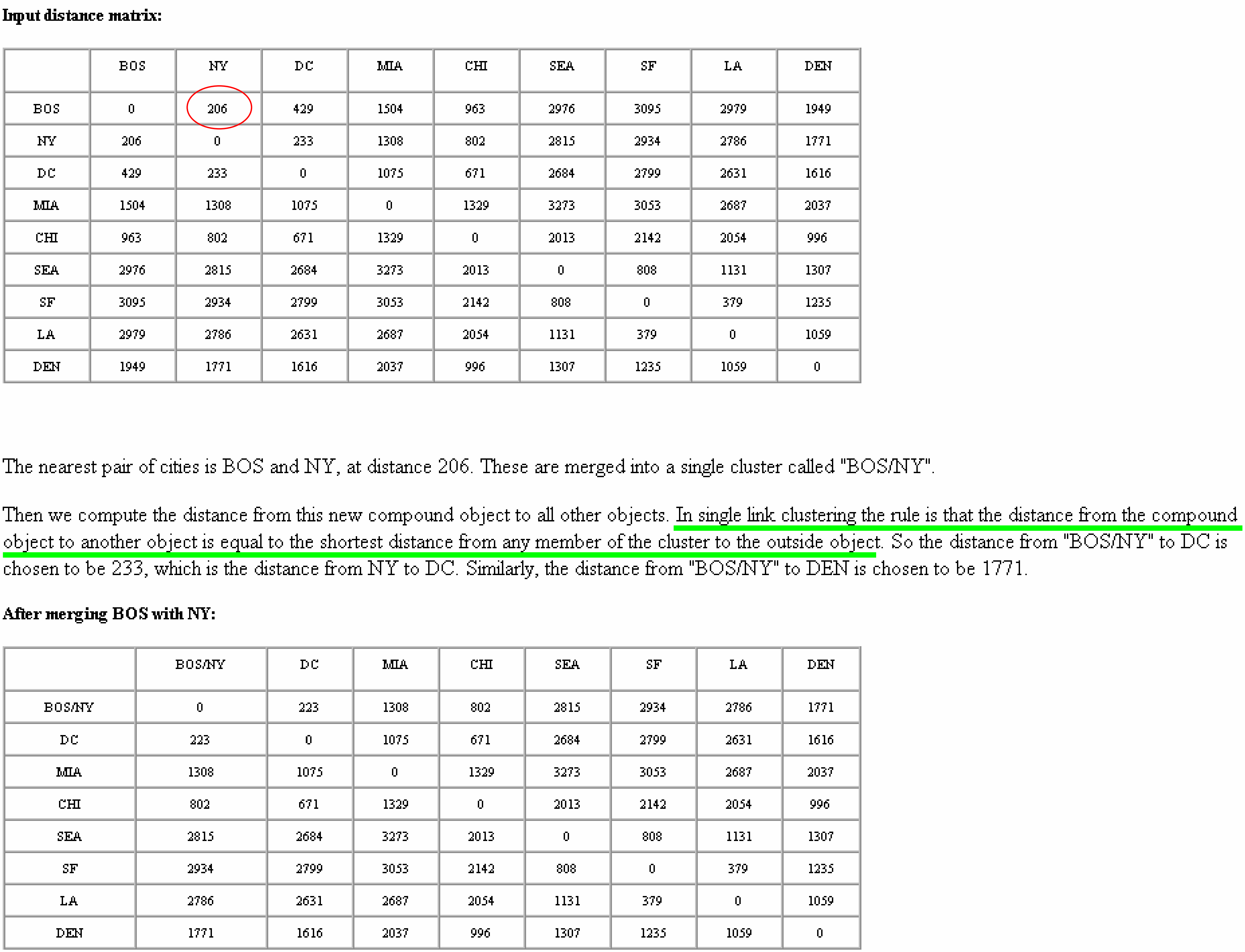
- The dissimilarity between clusters can be defined differently
  - Minimum dissimilarity between two objects
    - Single linkage
  - Maximum dissimilarity between two objects
    - Complete linkage
  - Average dissimilarity between two objects
    - Average method
  - Ward’s method
    - Interval scaled attributes
    - Error sum of squares of a cluster

Single linkage



Complete linkage





Input distance matrix:

	BOS	NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NY	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

The nearest pair of cities is BOS and NY, at distance 206. These are merged into a single cluster called "BOS/NY".

Then we compute the distance from this new compound object to all other objects. In single link clustering the rule is that the distance from the compound object to another object is equal to the shortest distance from any member of the cluster to the outside object. So the distance from "BOS/NY" to DC is chosen to be 233, which is the distance from NY to DC. Similarly, the distance from "BOS/NY" to DEN is chosen to be 1771.

After merging BOS with NY:

	BOS/NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY	0	223	1308	802	2815	2934	2786	1771
DC	223	0	1075	671	2684	2799	2631	1616
MIA	1308	1075	0	1329	3273	3053	2687	2037
CHI	802	671	1329	0	2013	2142	2054	996
SEA	2815	2684	3273	2013	0	808	1131	1307
SF	2934	2799	3053	2142	808	0	379	1235
LA	2786	2631	2687	2054	1131	379	0	1059
DEN	1771	1616	2037	996	1307	1235	1059	0

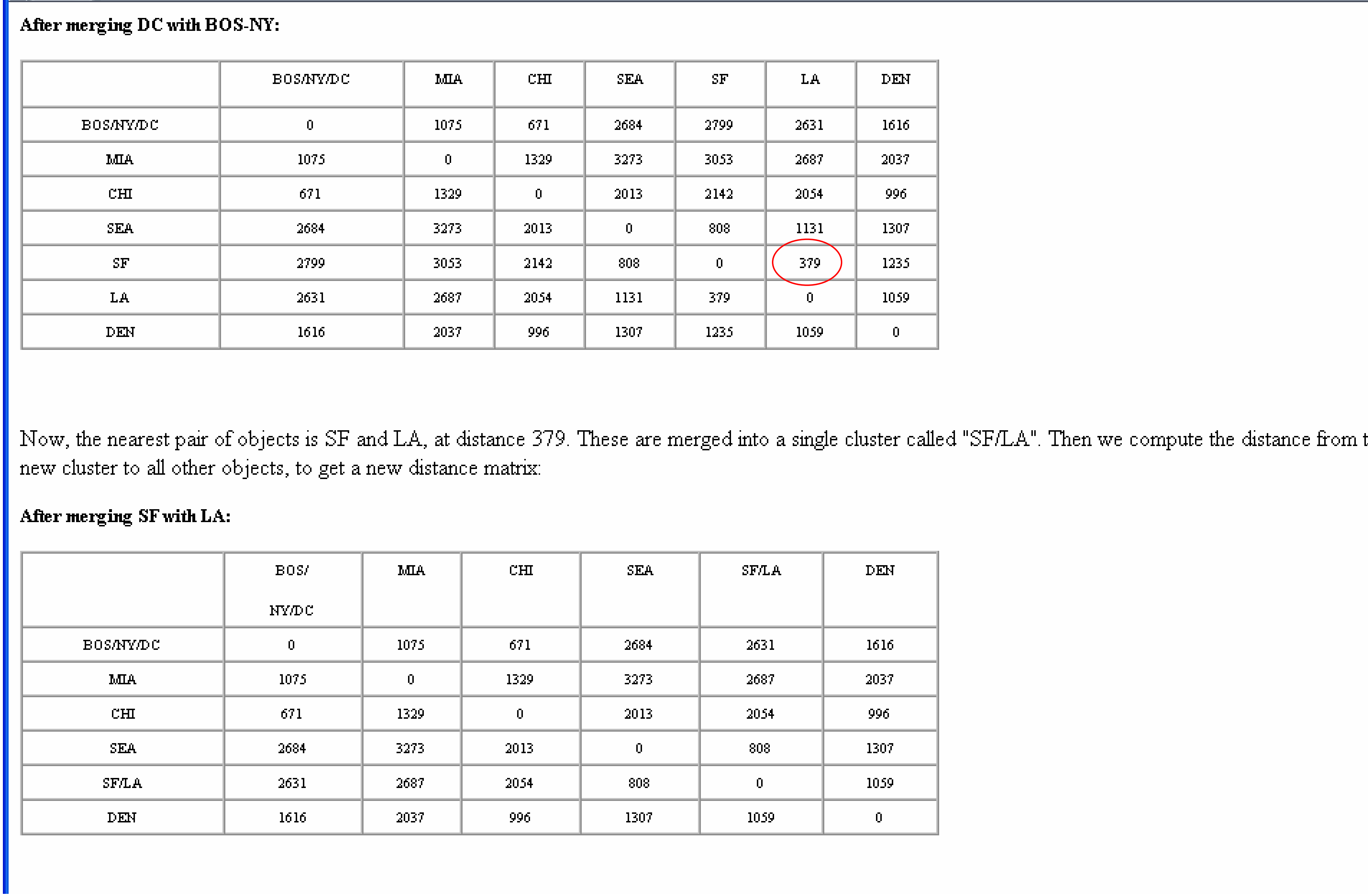
After merging BOS with NY:

	BOS/NY	DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY	0	223	1308	802	2815	2934	2786	1771
DC	223	0	1075	671	2684	2799	2631	1616
MIA	1308	1075	0	1329	3273	3053	2687	2037
CHI	802	671	1329	0	2013	2142	2054	996
SEA	2815	2684	3273	2013	0	808	1131	1307
SF	2934	2799	3053	2142	808	0	379	1235
LA	2786	2631	2687	2054	1131	379	0	1059
DEN	1771	1616	2037	996	1307	1235	1059	0

The nearest pair of objects is BOS/NY and DC, at distance 223. These are merged into a single cluster called "BOS/NY/DC". Then we compute the distanc from this new cluster to all other clusters, to get a new distance matrix:

After merging DC with BOS-NY:

	BOS/NY/DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY/DC	0	1075	671	2684	2799	2631	1616
MIA	1075	0	1329	3273	3053	2687	2037
CHI	671	1329	0	2013	2142	2054	996
SEA	2684	3273	2013	0	808	1131	1307
SF	2799	3053	2142	808	0	379	1235
LA	2631	2687	2054	1131	379	0	1059
DEN	1616	2037	996	1307	1235	1059	0



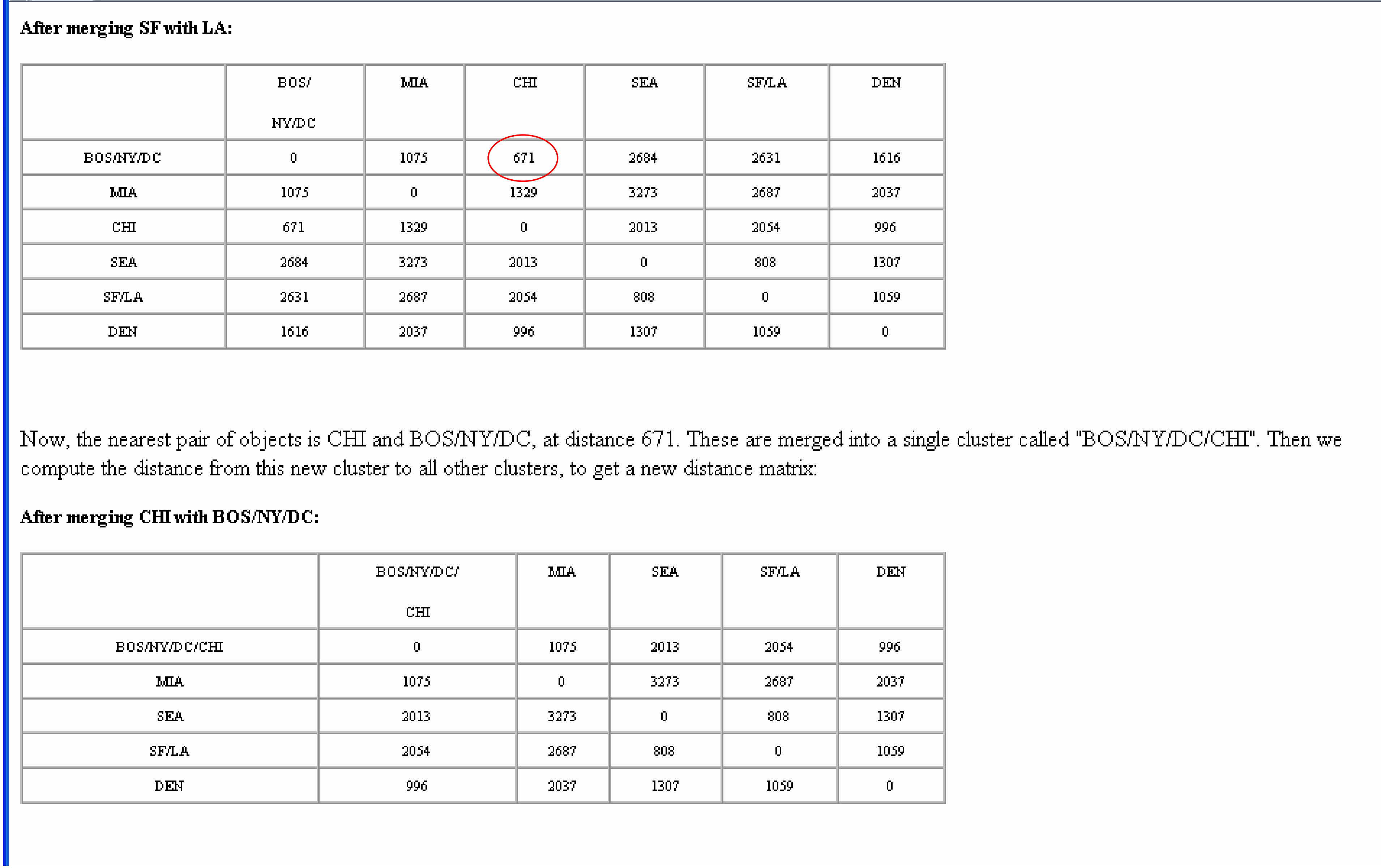
After merging DC with BOS-NY:

	BOS/NY/DC	MIA	CHI	SEA	SF	LA	DEN
BOS/NY/DC	0	1075	671	2684	2799	2631	1616
MIA	1075	0	1329	3273	3053	2687	2037
CHI	671	1329	0	2013	2142	2054	996
SEA	2684	3273	2013	0	808	1131	1307
SF	2799	3053	2142	808	0	379	1235
LA	2631	2687	2054	1131	379	0	1059
DEN	1616	2037	996	1307	1235	1059	0

Now, the nearest pair of objects is SF and LA, at distance 379. These are merged into a single cluster called "SF/LA". Then we compute the distance from t
new cluster to all other objects, to get a new distance matrix:

After merging SF with LA:

	BOS/ NY/DC	MIA	CHI	SEA	SF/LA	DEN
BOS/NY/DC	0	1075	671	2684	2631	1616
MIA	1075	0	1329	3273	2687	2037
CHI	671	1329	0	2013	2054	996
SEA	2684	3273	2013	0	808	1307
SF/LA	2631	2687	2054	808	0	1059
DEN	1616	2037	996	1307	1059	0



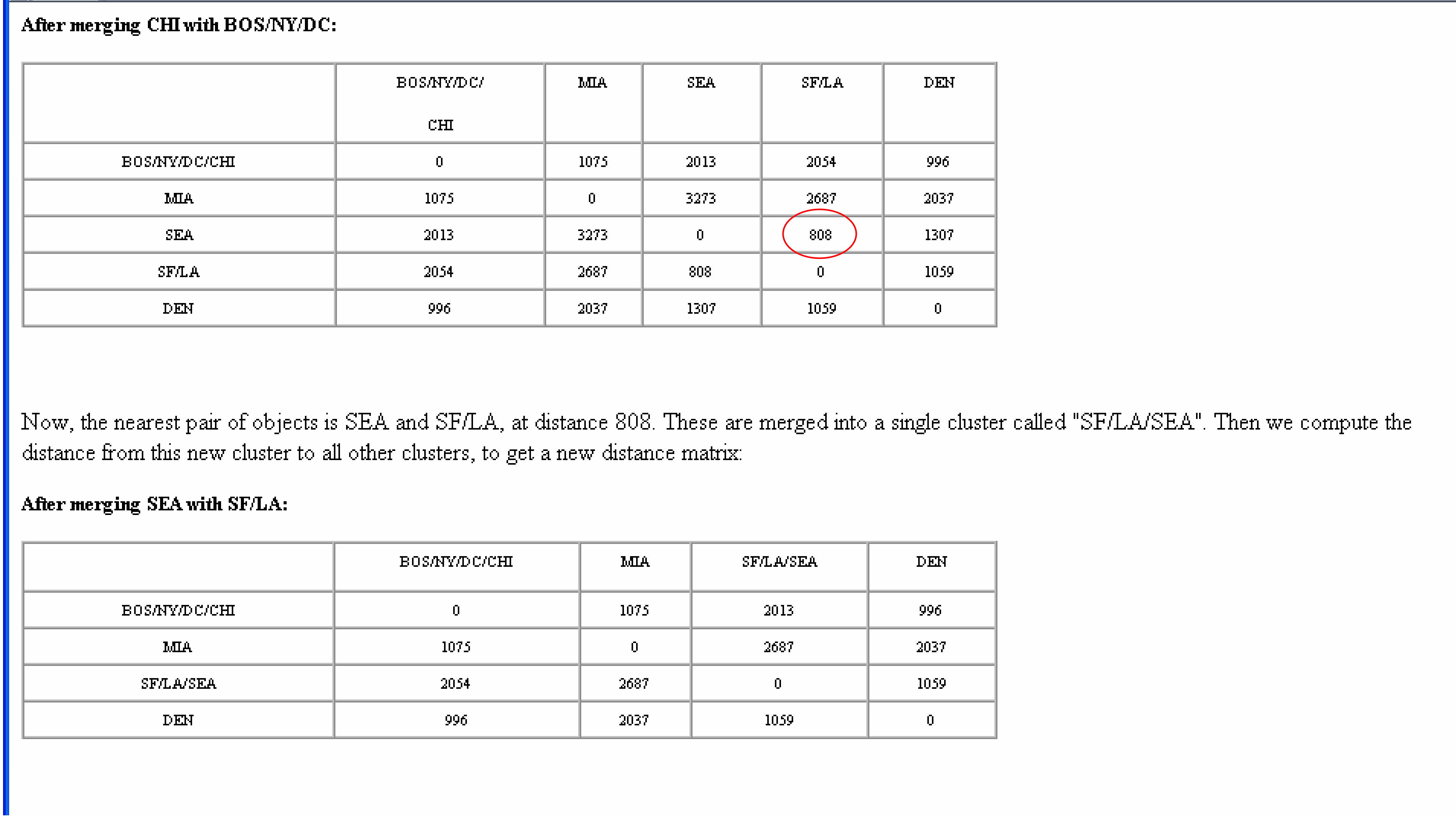
After merging SF with LA:

	BOS/NY/DC	MIA	CHI	SEA	SF/LA	DEN
BOS/NY/DC	0	1075	671	2684	2631	1616
MIA	1075	0	1329	3273	2687	2037
CHI	671	1329	0	2013	2054	996
SEA	2684	3273	2013	0	808	1307
SF/LA	2631	2687	2054	808	0	1059
DEN	1616	2037	996	1307	1059	0

Now, the nearest pair of objects is CHI and BOS/NY/DC, at distance 671. These are merged into a single cluster called "BOS/NY/DC/CHI". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix:

After merging CHI with BOS/NY/DC:

	BOS/NY/DC/CHI	MIA	SEA	SF/LA	DEN
BOS/NY/DC/CHI	0	1075	2013	2054	996
MIA	1075	0	3273	2687	2037
SEA	2013	3273	0	808	1307
SF/LA	2054	2687	808	0	1059
DEN	996	2037	1307	1059	0



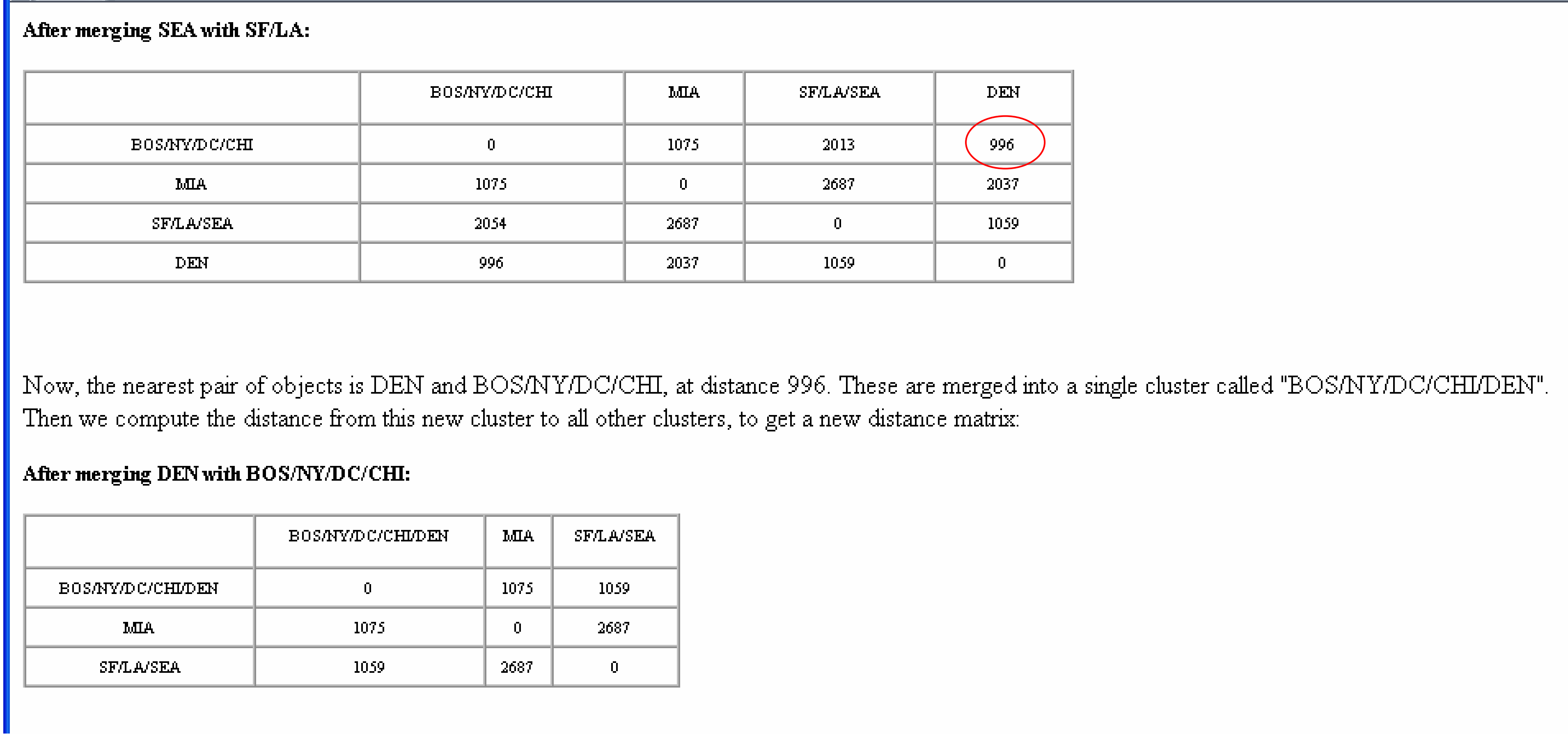
After merging CHI with BOS/NY/DC:

	BOS/NY/DC/ CHI	MIA	SEA	SF/LA	DEN
BOS/NY/DC/CHI	0	1075	2013	2054	996
MIA	1075	0	3273	2687	2037
SEA	2013	3273	0	808	1307
SF/LA	2054	2687	808	0	1059
DEN	996	2037	1307	1059	0

Now, the nearest pair of objects is SEA and SF/LA, at distance 808. These are merged into a single cluster called "SF/LA/SEA". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix:

After merging SEA with SF/LA:

	BOS/NY/DC/CHI	MIA	SF/LA/SEA	DEN
BOS/NY/DC/CHI	0	1075	2013	996
MIA	1075	0	2687	2037
SF/LA/SEA	2054	2687	0	1059
DEN	996	2037	1059	0



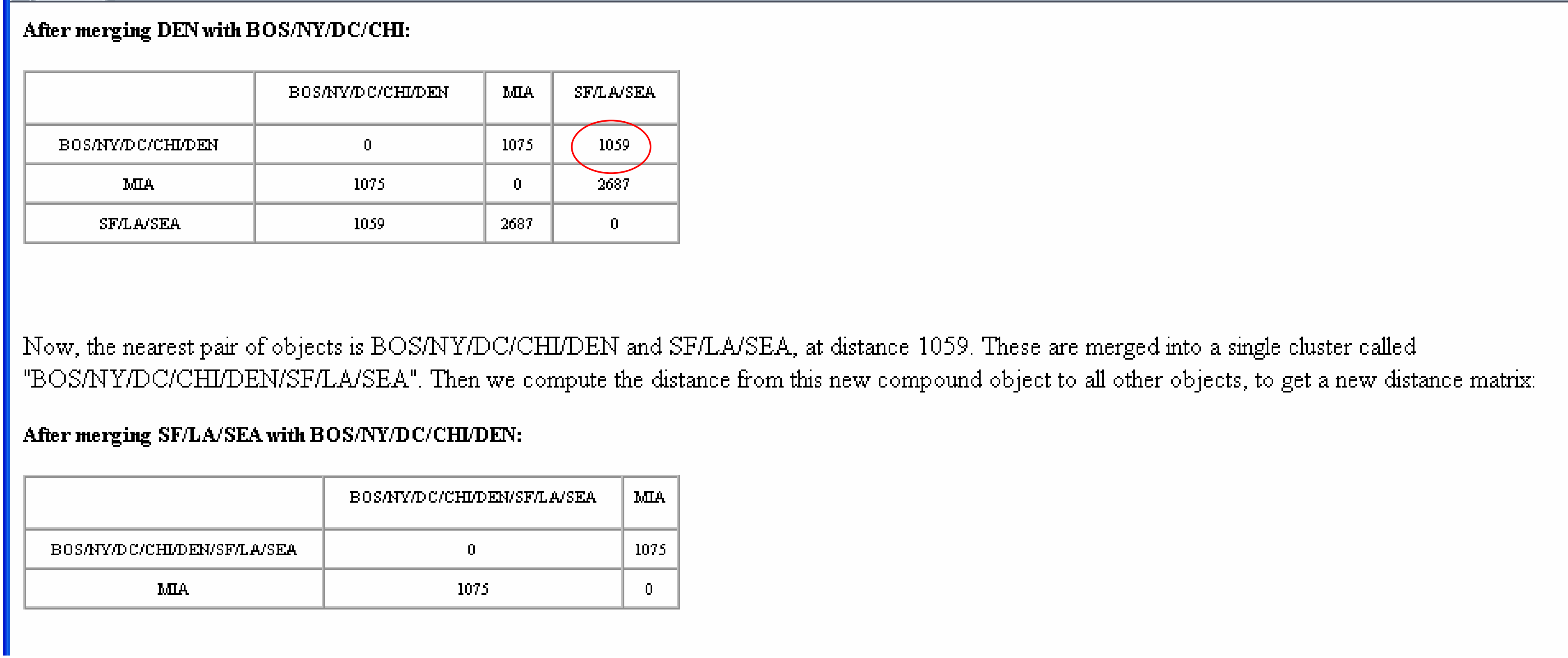
After merging SEA with SF/LA:

	BOS/NY/DC/CHI	MIA	SF/LA/SEA	DEN
BOS/NY/DC/CHI	0	1075	2013	996
MIA	1075	0	2687	2037
SF/LA/SEA	2054	2687	0	1059
DEN	996	2037	1059	0

Now, the nearest pair of objects is DEN and BOS/NY/DC/CHI, at distance 996. These are merged into a single cluster called "BOS/NY/DC/CHI/DEN". Then we compute the distance from this new cluster to all other clusters, to get a new distance matrix:

After merging DEN with BOS/NY/DC/CHI:

	BOS/NY/DC/CHI/DEN	MIA	SF/LA/SEA
BOS/NY/DC/CHI/DEN	0	1075	1059
MIA	1075	0	2687
SF/LA/SEA	1059	2687	0



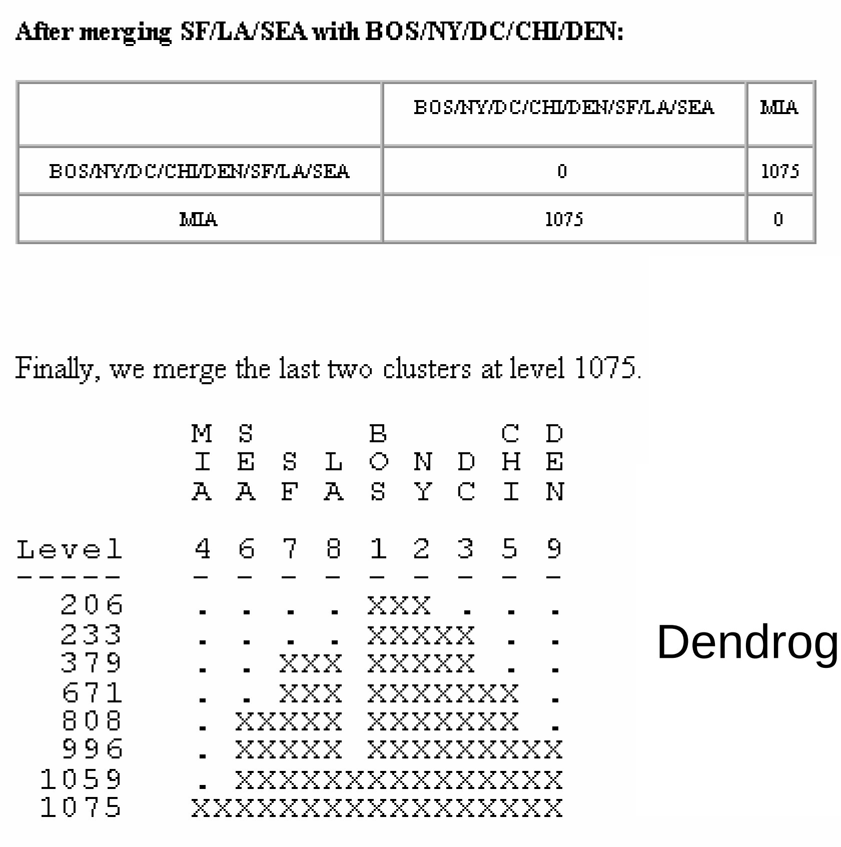
After merging DEN with BOS/NY/DC/CHI:

	BOS/NY/DC/CHI/DEN	MIA	SF/LA/SEA
BOS/NY/DC/CHI/DEN	0	1075	1059
MIA	1075	0	2687
SF/LA/SEA	1059	2687	0

Now, the nearest pair of objects is BOS/NY/DC/CHI/DEN and SF/LA/SEA, at distance 1059. These are merged into a single cluster called "BOS/NY/DC/CHI/DEN/SF/LA/SEA". Then we compute the distance from this new compound object to all other objects, to get a new distance matrix:

After merging SF/LA/SEA with BOS/NY/DC/CHI/DEN:

	BOS/NY/DC/CHI/DEN/SF/LA/SEA	MIA
BOS/NY/DC/CHI/DEN/SF/LA/SEA	0	1075
MIA	1075	0



After merging SF/LA/SEA with BOS/NY/DC/CHI/DEN:

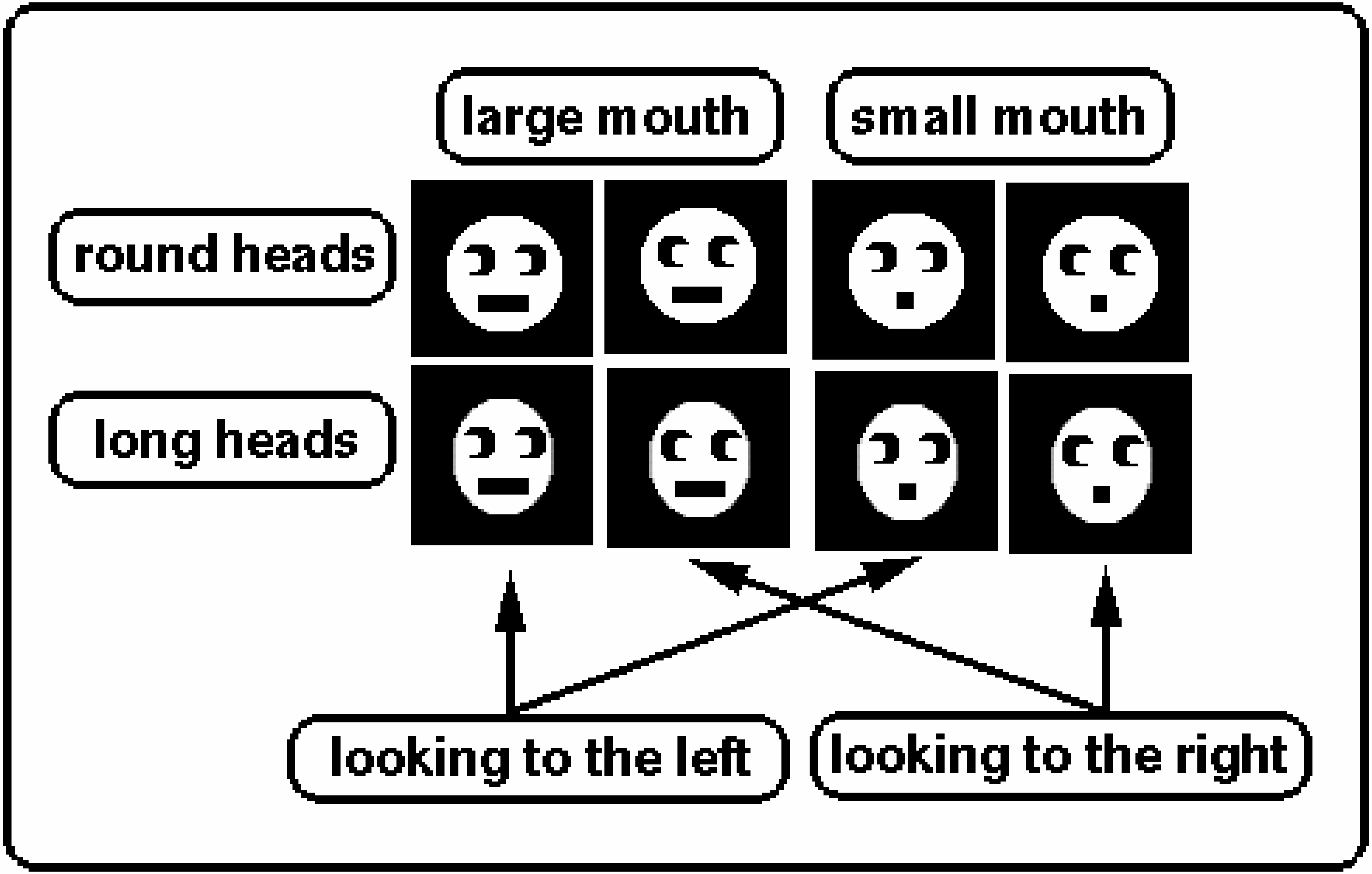
	BOS/NY/DC/CHI/DEN/SF/LA/SEA	MIA
BOS/NY/DC/CHI/DEN/SF/LA/SEA	0	1075
MIA	1075	0

Finally, we merge the last two clusters at level 1075.

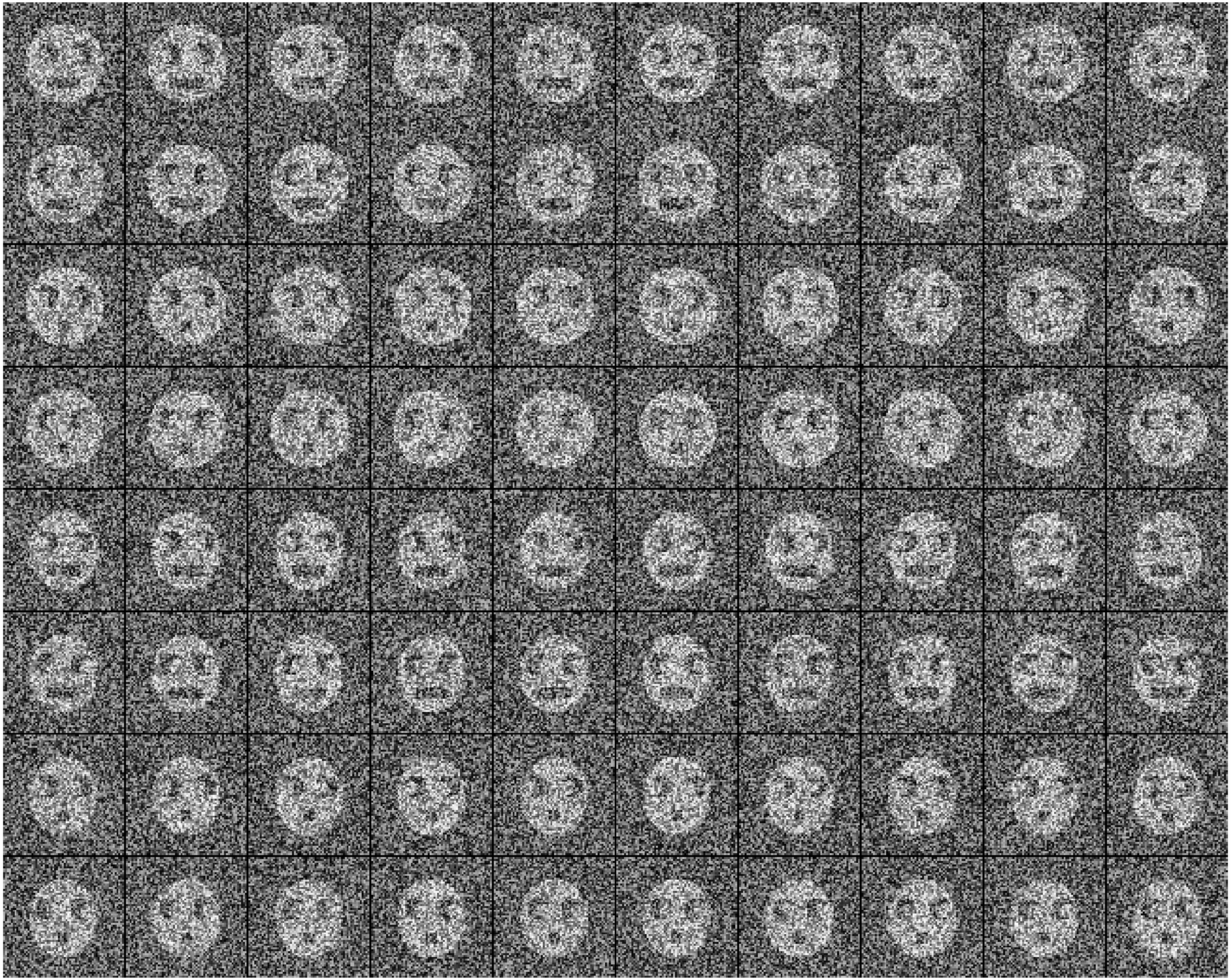
```
        M S       B       C D
        I E S L   O N D   H E
        A A F A   S Y C   I N

Level   4 6 7 8   1 2 3   5 9
-----   - - - -   - - -   - -
  206   . . . .   XXX .   . .
  233   . . . .   XXXXX   . .
  379   . . XXX   XXXXX   . .
  671   . . XXX   XXXXXXX   .
  808   . XXXXX   XXXXXXX   .
  996   . XXXXX   XXXXXXXXXXX
 1059   . XXXXXXXXXXXXXXXXXXX
 1075   XXXXXXXXXXXXXXXXXXXXX
```

Dendrogram (history of merging steps).

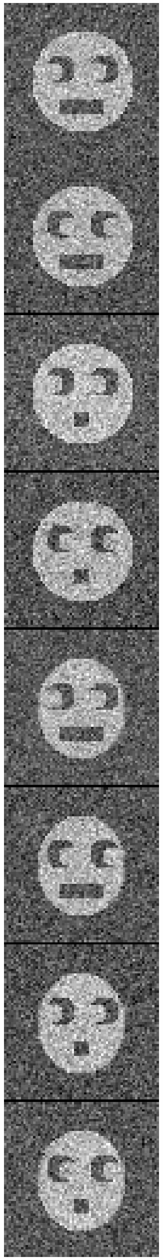
Brétaudière JP and Frank J (1986) Reconstitution of molecule images analyzed by correspondence analysis: A tool for structural interpretation. *J. Microsc.* **144**, 1-14.

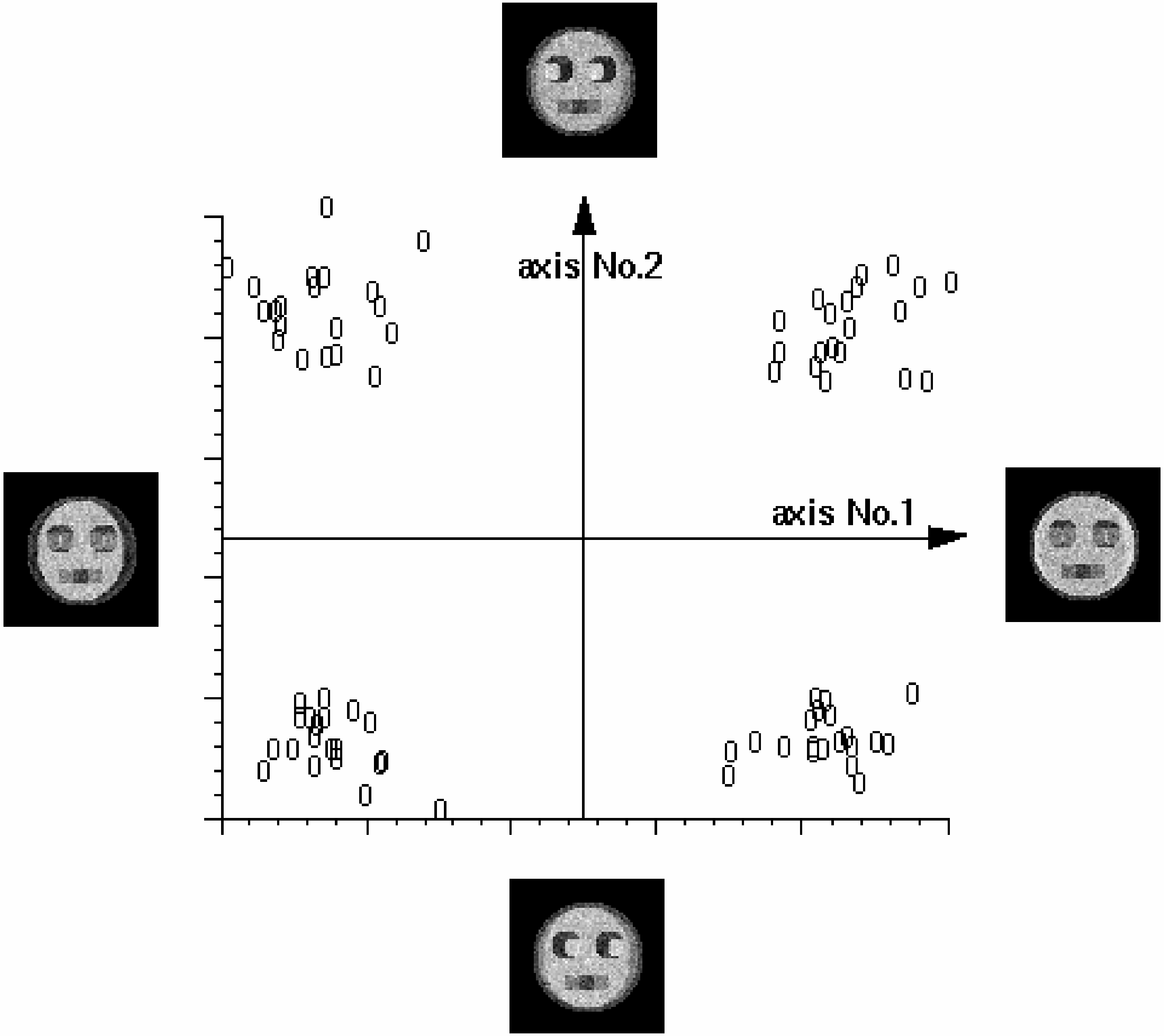


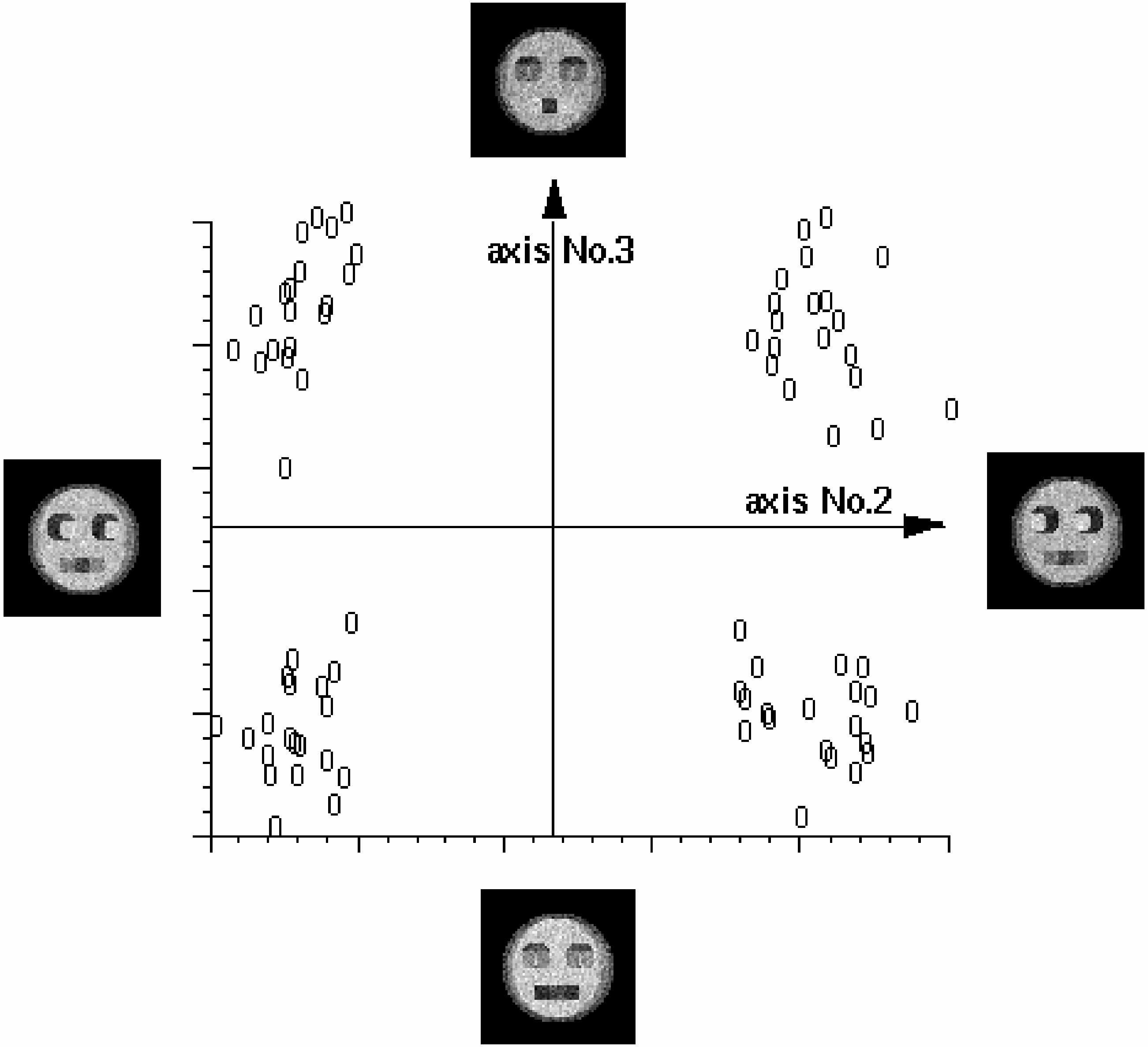
10 copies of the 8 types of heads + random noise

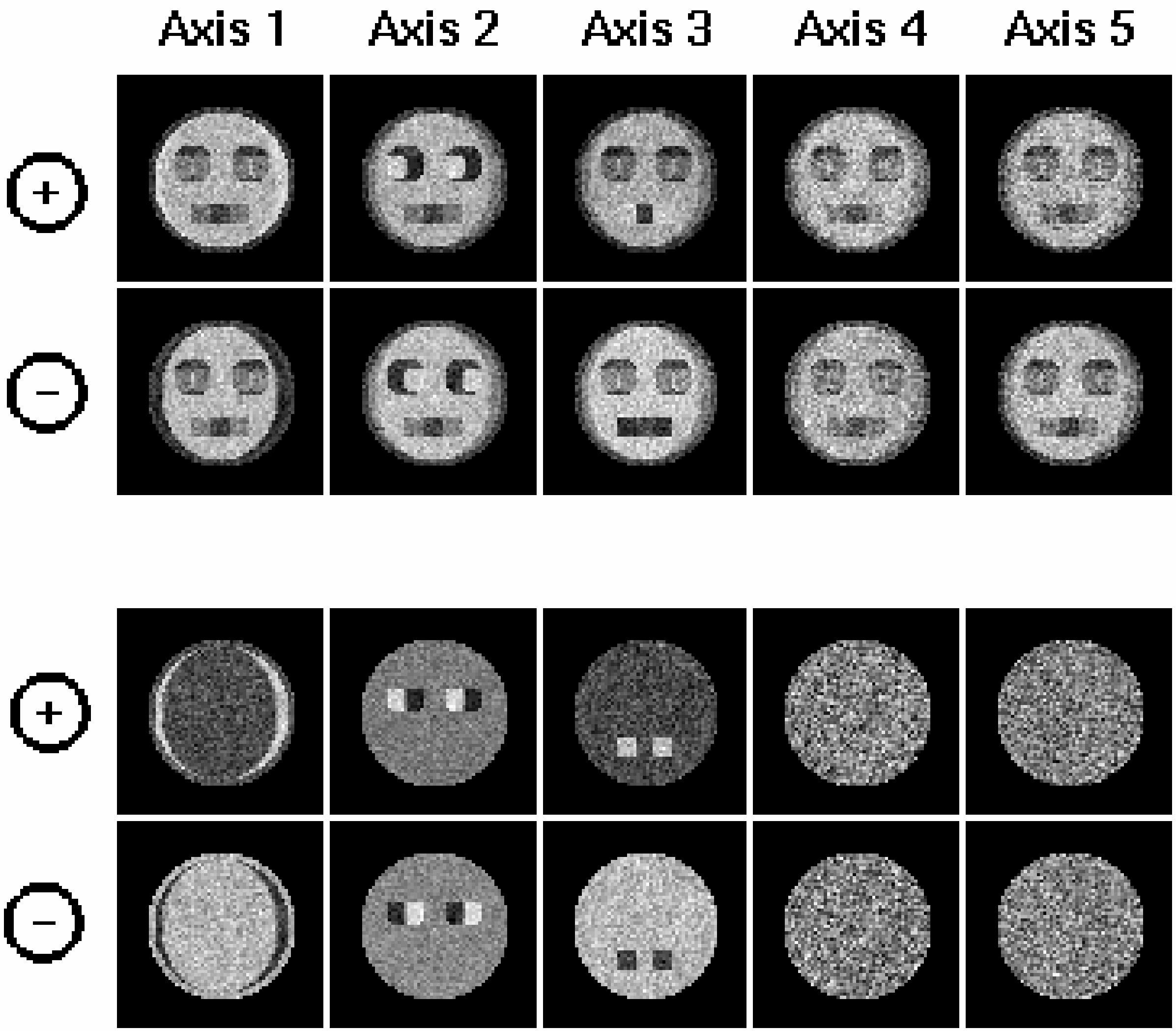


Averages




axis No.2
axis No.1


axis No.3
axis No.2

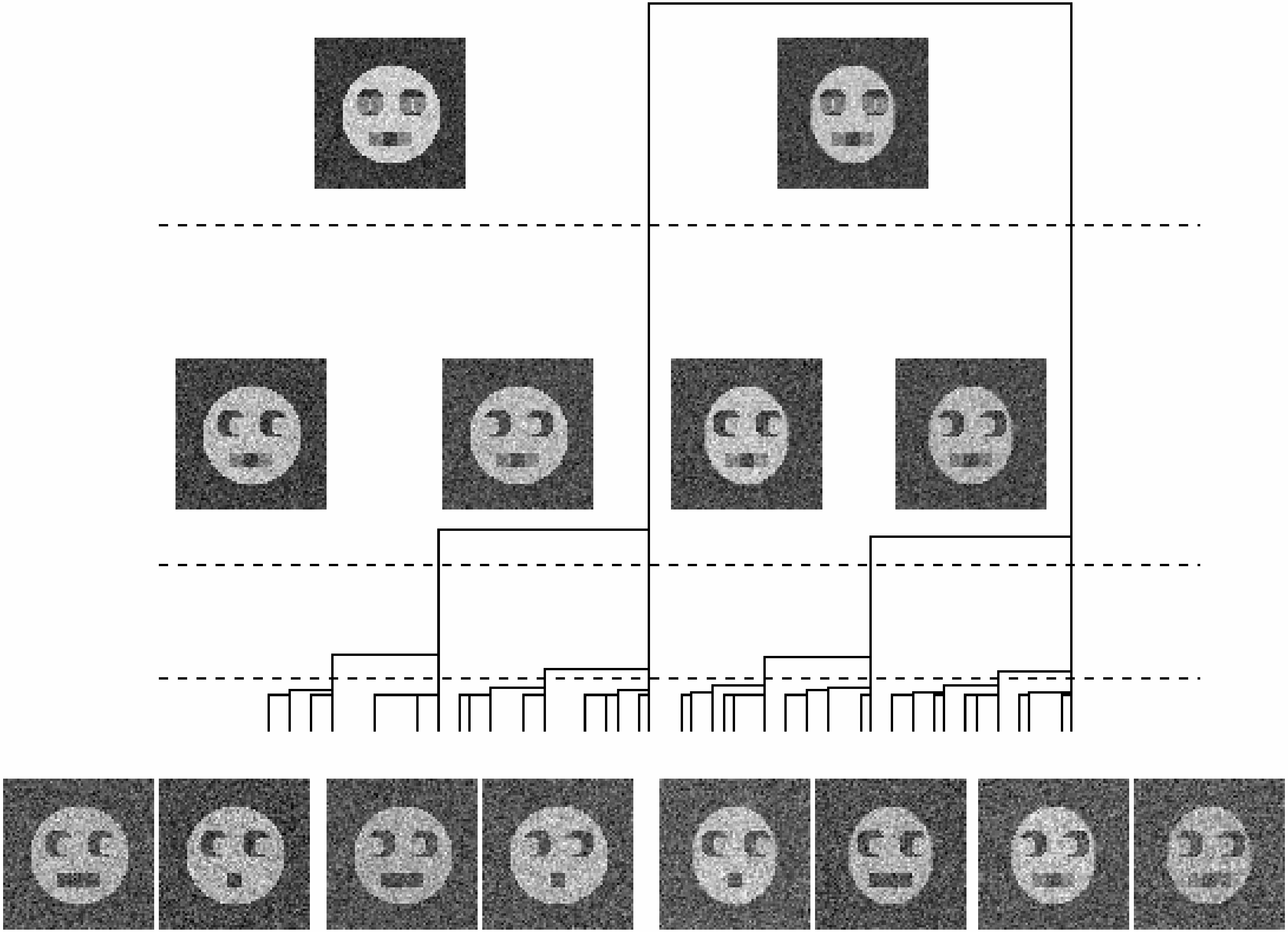


Reconstituted images

Importance images

(Mv Heel, Ph.D Thesis)

HIERARCHICAL ASCENDENT CLASSIFICATION

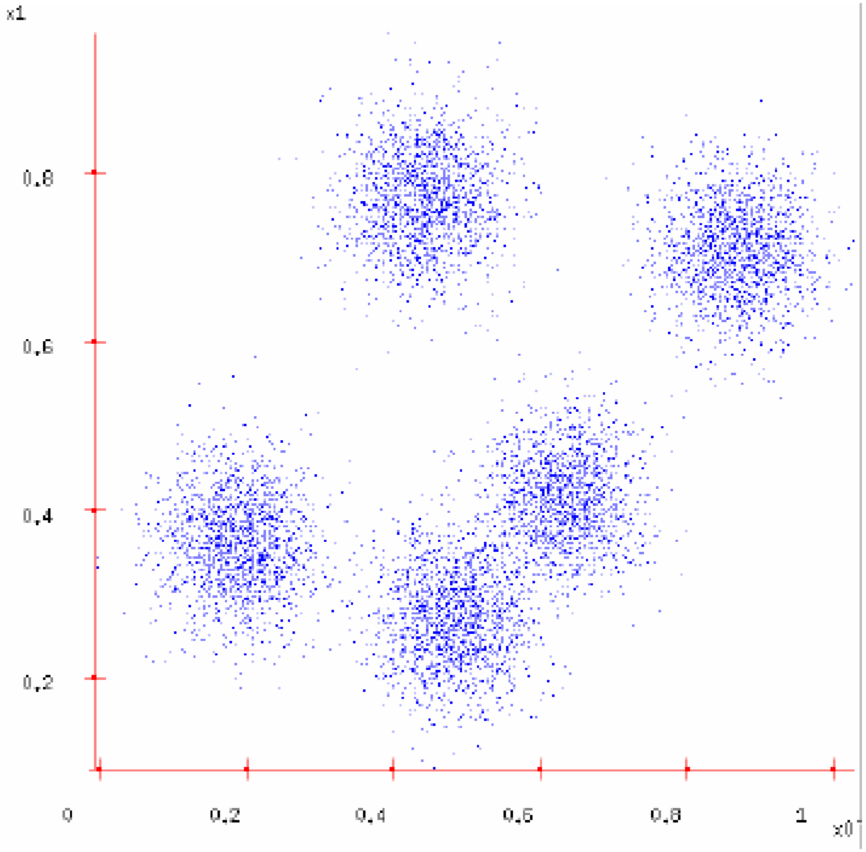


K-Means

Find a partition of a dataset such that objects within each class are closer to their class centers (averages) that to other class centers.

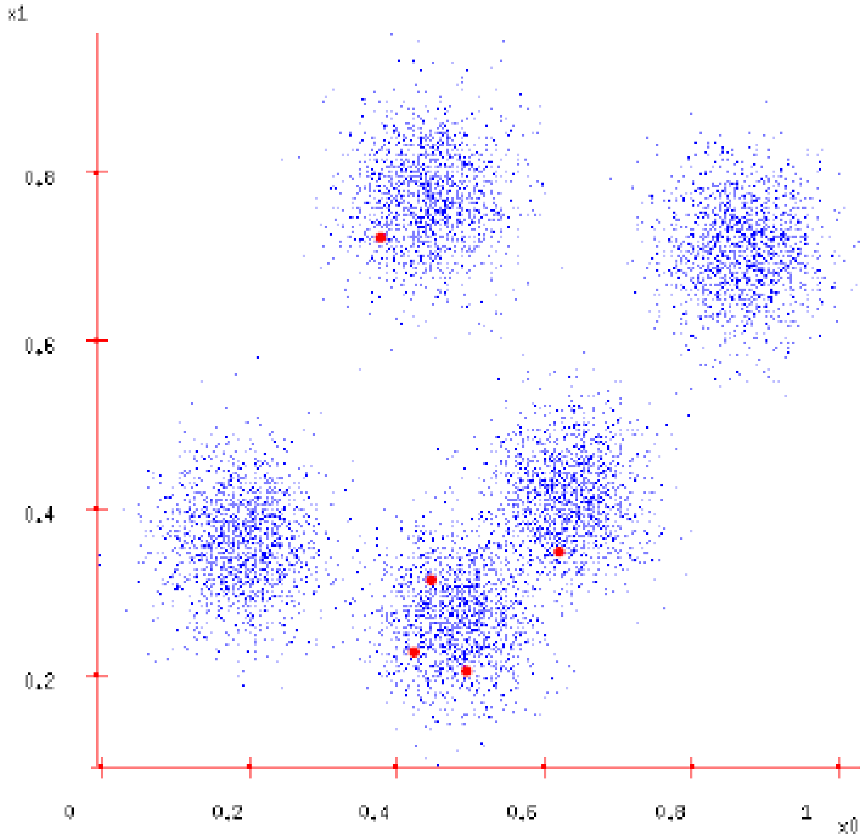
K-Means

1. Set the number of groups K



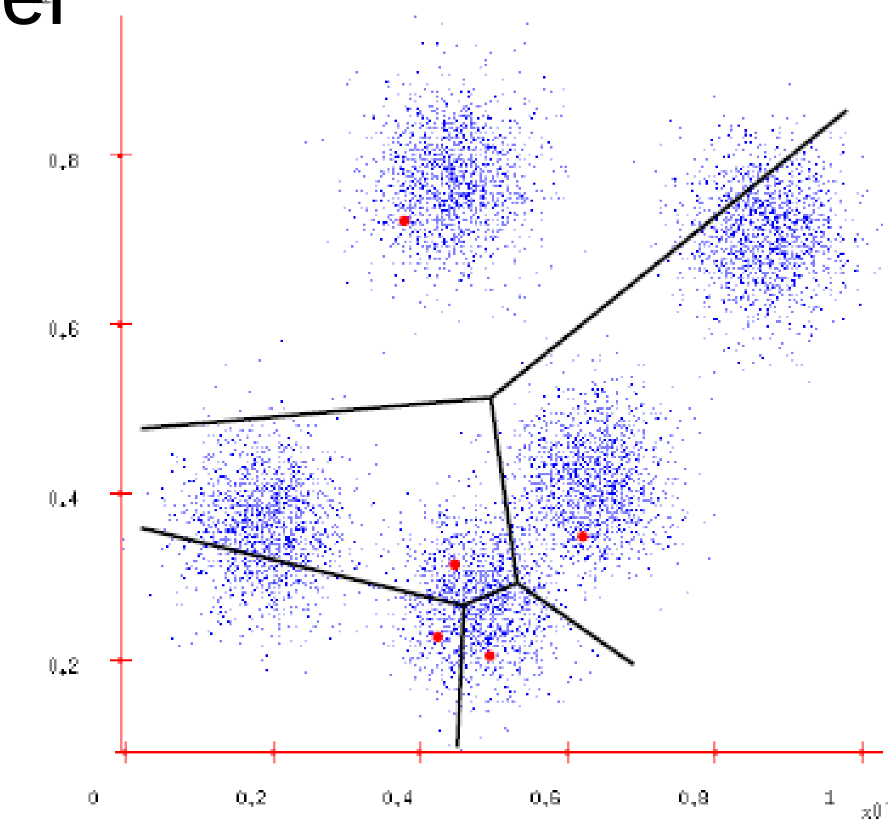
K-Means

2. Randomly select K class centers



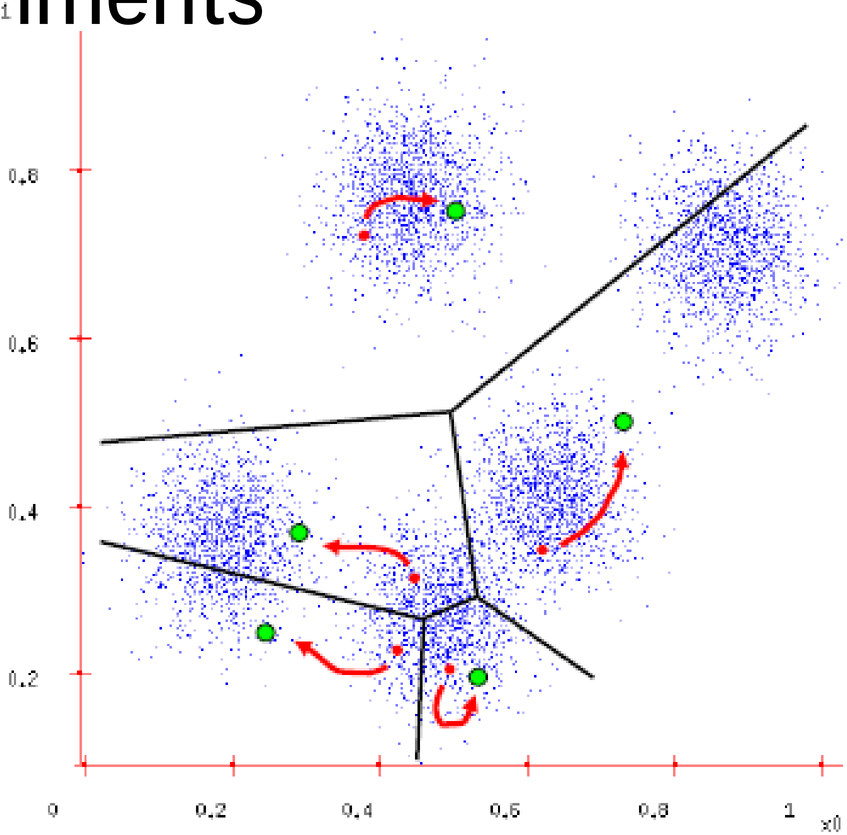
K-Means

3. Assign each point to its nearest class center



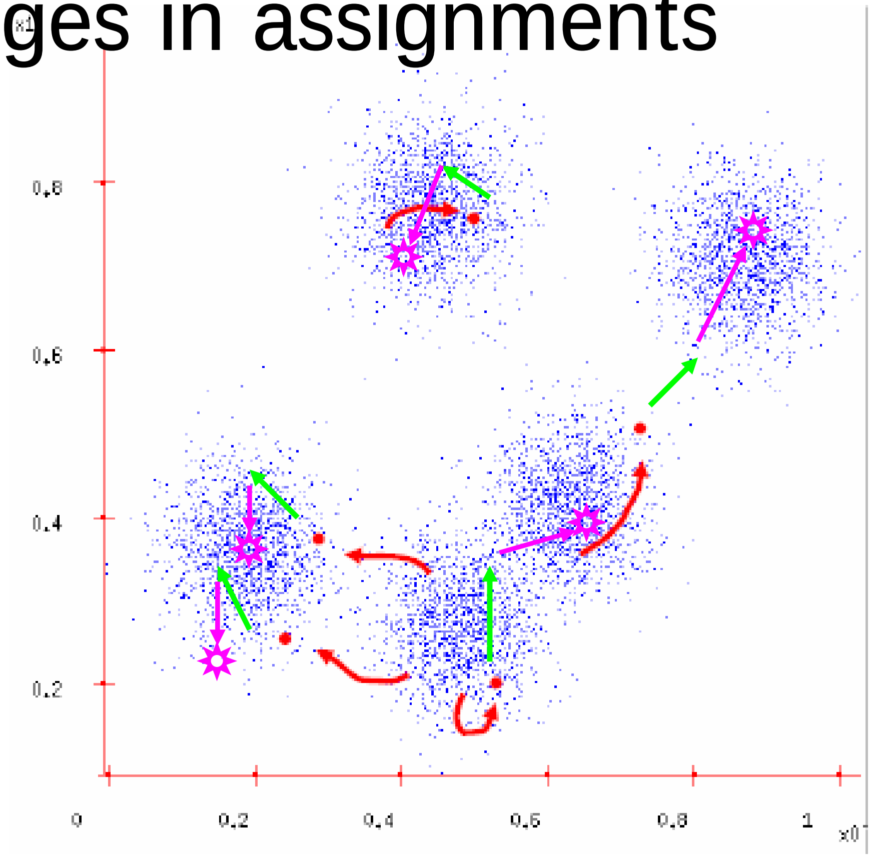
K-Means

4. Recompute class centers based on new assignments



K-Means

5. Repeat steps 4 & 5 until no further changes in assignments



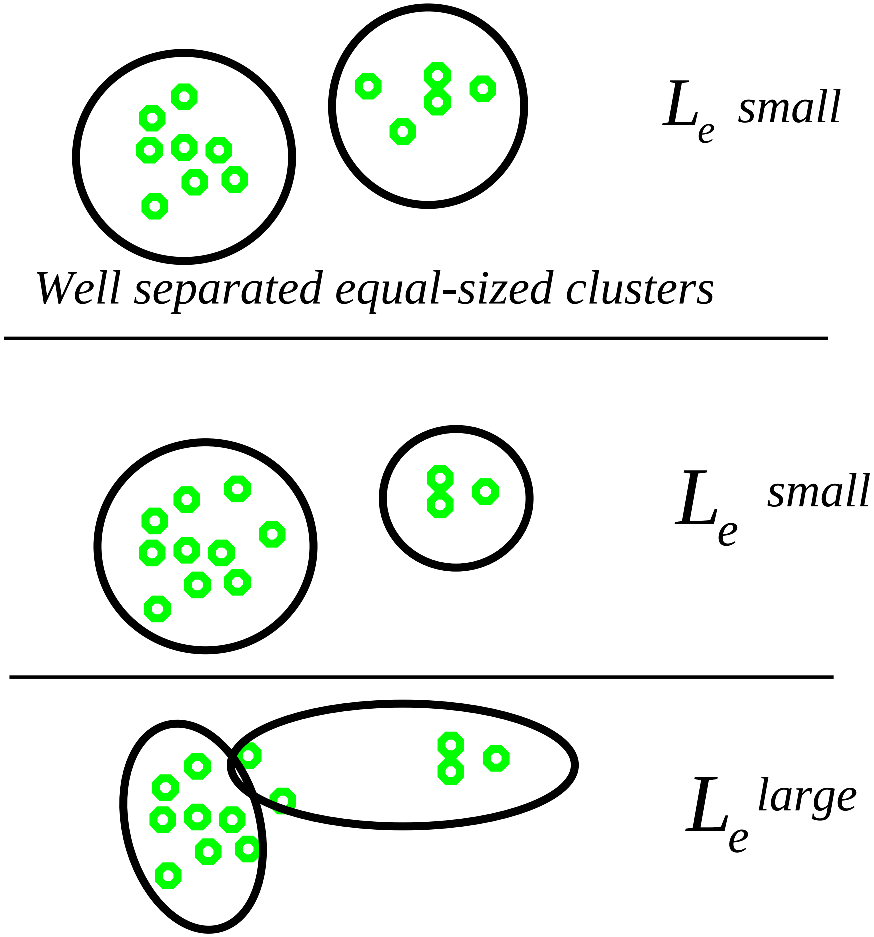
K-Means

- The algorithm steps are (J. MacQueen, 1967):
- Choose the number of clusters, *k*.
- Randomly generate *k* clusters and determine the cluster centers, or directly generate *k* random points as cluster centers.
- Assign each point to the nearest cluster center.
- Recompute the new cluster centers.
- Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed).

K-Means Clustering
The Sum-of-Squared-Error Criterion

$$\mathbf{m}_k = \frac{1}{n_k} \sum_{i \in C_k}^{n_k} \mathbf{x}_i$$

$$L_e = \sum_{k=1}^{c} \sum_{i \in C_k}^{n_k} \left\| \mathbf{x}_i - \mathbf{m}_k \right\|^2$$



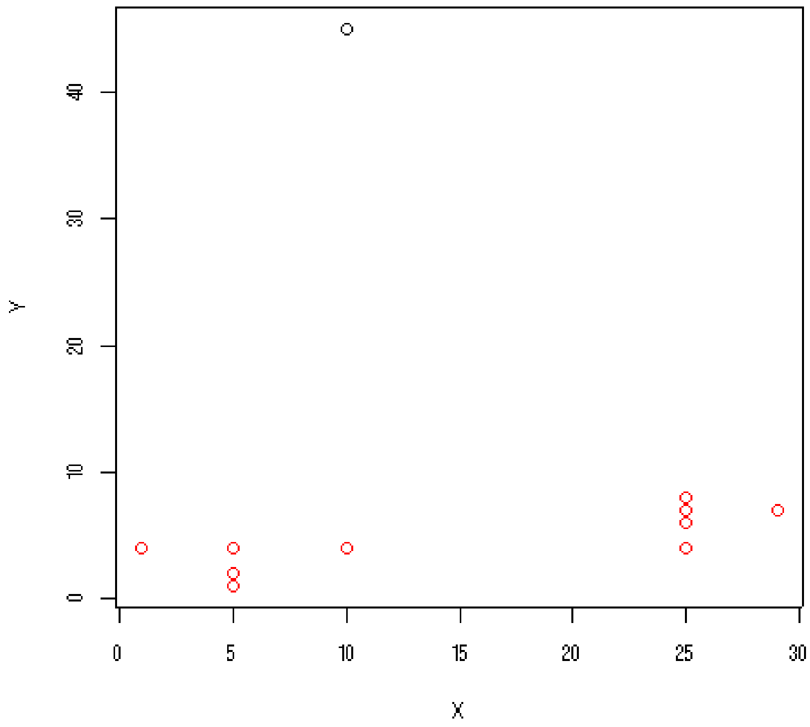
SSE K-Means

```
Algorithm: K-means
Input:     $k$        number of clusters
           $t$        number of iterations
           data       the data, $n$ samples
Output:    C          a set of $k$ clusters

$cent$ = arbitrarily select $k$ objects as initial centers
compute centers and criteria $L_k$ for all clusters
do
  do (randomly select $sample$ $\mathbf{x}$ in $data$)

    if(reassignment of $\mathbf{x}$ from its current cluster decreases $L$)
       reassign $\mathbf{x}$;
       update averages and criteria for two clusters;

until(no change in $L$  in $n$ attempts)
End
```

K-Means Summary

- Based on a mathematical definition of a cluster (SSE)
- Very simple algorithm
- $O(knt)$ time complexity
- Circular cluster shape only
- Guaranteed to converge in a finite number of steps
- Is not guaranteed to converge to a global minimum
- Outliers can have very negative impact

Outliers



Optimum number of clusters

- *Hierarchical clustering:*
  by eye

- *K*-*means* (*moving averages*):
  by eye

- SSE *K-means*:
  dispersion criteria

Optimum number of clusters in SSE K-means

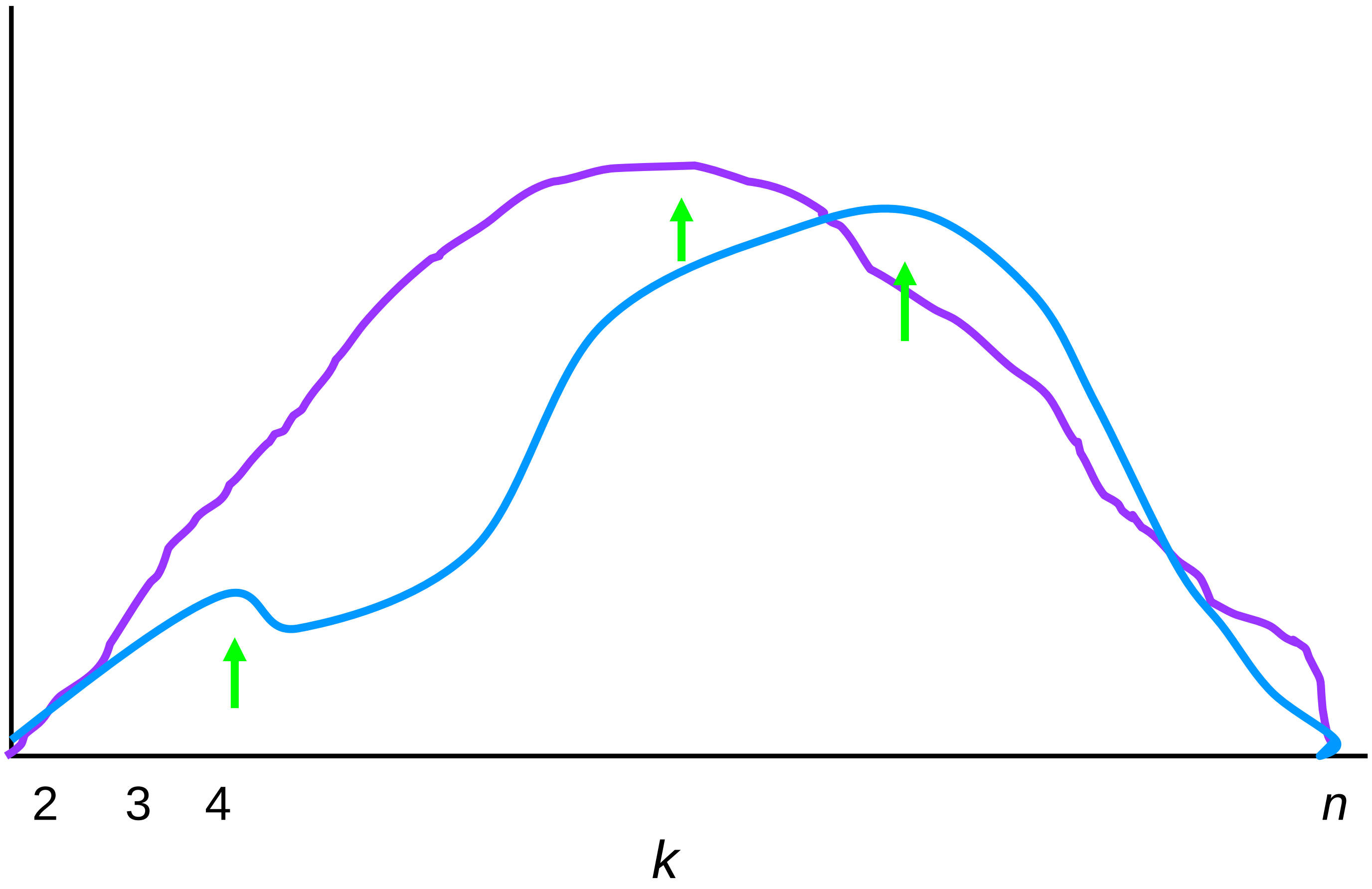
- Tr(**B**), trace of between-groups sum of squares matrix (between-groups dispersion)
- Tr(**W**), trace of within-groups sum of squares matrix (within-groups dispersion)
- Coleman criterion: $$C = Tr\left(\mathbf{B}\right) * Tr\left(\mathbf{W}\right)$$
- Harabasz criterion: $$H = \frac{Tr\left(\mathbf{B}\right) \big/ \left(k-1\right)}{Tr\left(\mathbf{W}\right) \big/ \left(n-k\right)}$$

Optimum number of clusters in SSE K-means

C, H

$$C = Tr\left(\mathbf{B}\right) * Tr\left(\mathbf{W}\right)$$

$$H = \frac{Tr\left(\mathbf{B}\right) \big/ \left(k-1\right)}{Tr\left(\mathbf{W}\right) \big/ \left(n-k\right)}$$



Other clustering methods used in EM

1. Fuzzy k-means
2. Self-organizing maps

Self-organizing map (SOM)

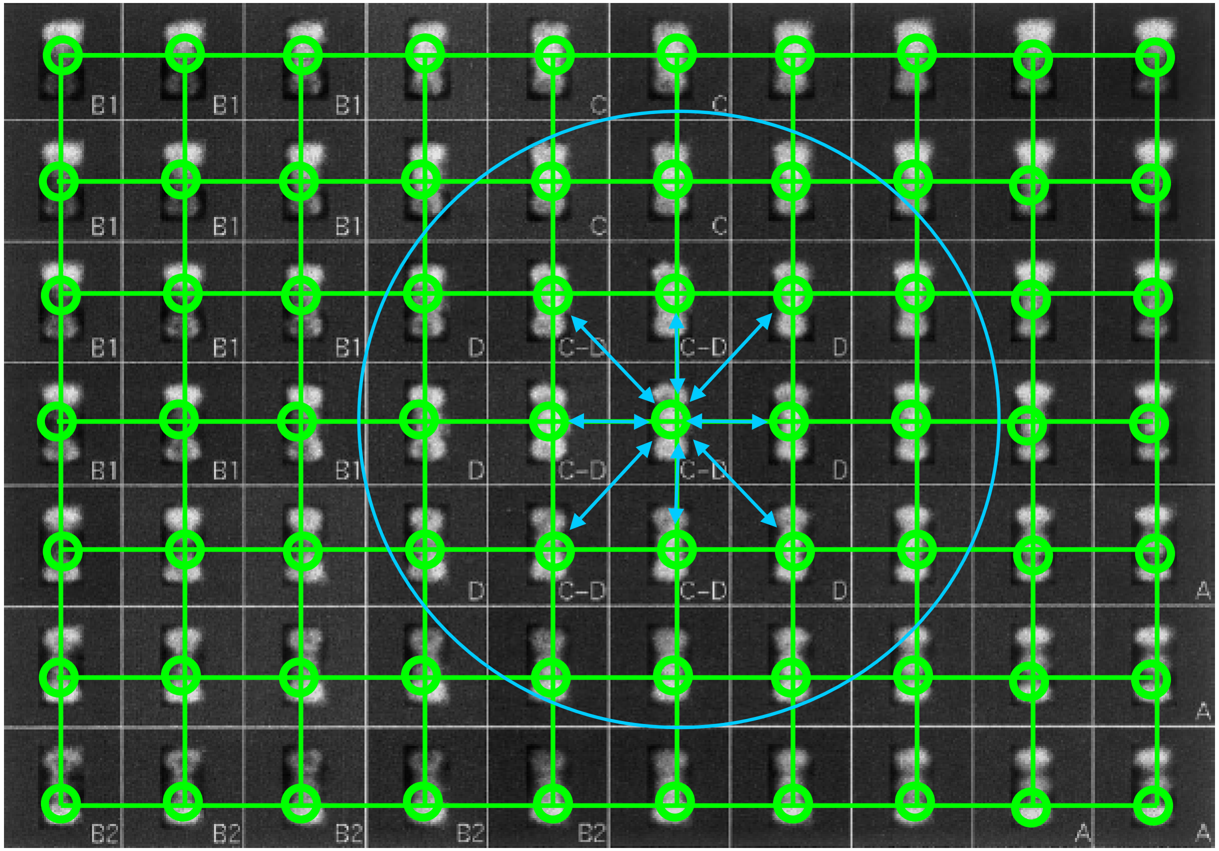


FIG. 6. Results of KerDenSOM for the Large T-Antigen double hexamers. A 10 × 7 map with a square topology was used. A Gaussian kernel with a deterministic annealing strategy varying the regularization factor ϑ from 300 down to 250 in 20 steps was applied. The best resulting map according to the generalized cross-validation criterion was selected with a $\vartheta = 250$ and a $\alpha = 0.94$. 300 iterations and a stopping criteria of 1×10^{-7} were also used.

Pascual-Montano *et al*., 2001. A novel neural network technique for analysis and classification of EM single-particle images. J. Struct. Biol. 133, 233-245

What does it have to do with single particle analysis?!?

- Regretfully, very little...
  - No accounting for image formation model
  - No accounting for the fact that images originate (or should originate) from the same object
  - No method developed specifically for single particle analysis

All key steps in single particle analysis can be well understood when formulated as clustering problem

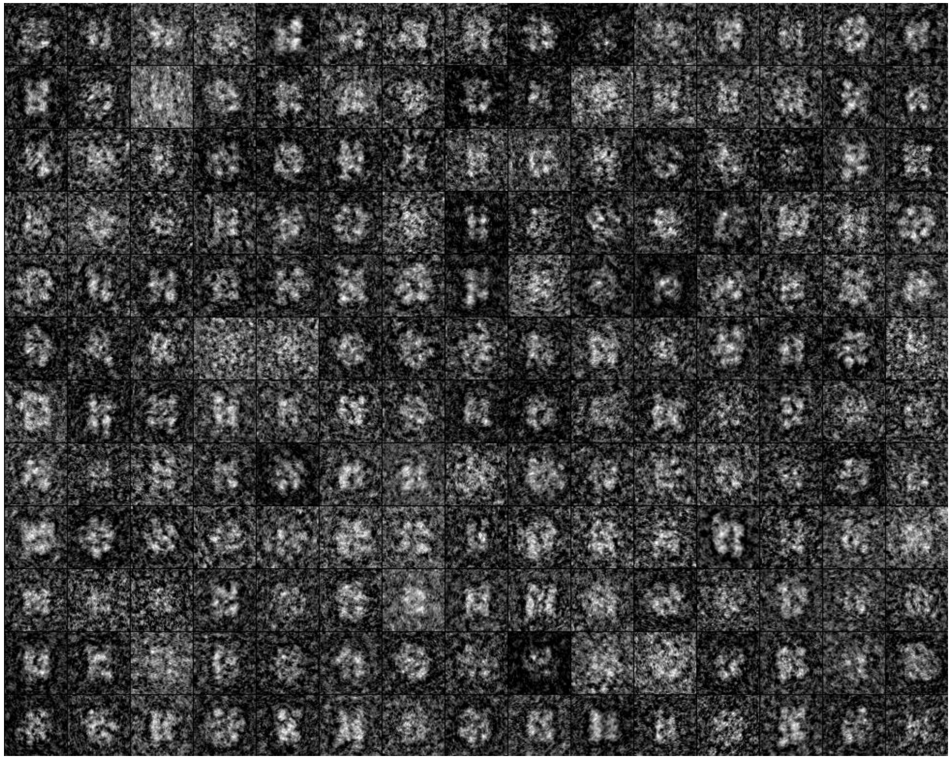
1. Multi-reference 2-D alignment

2. Ab initio structure determination

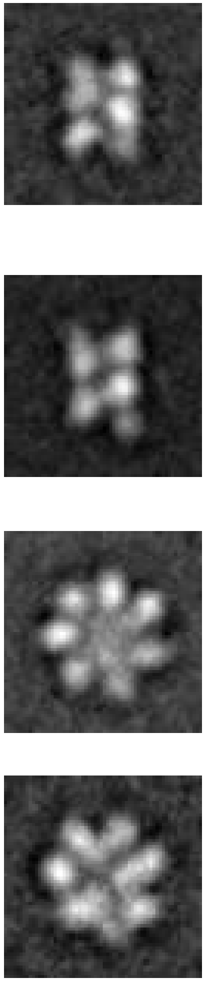
3. 3-D structure refinement (projection matching)

4. 3-D multi-reference alignment

2-D multi-reference alignment



n images (objects)

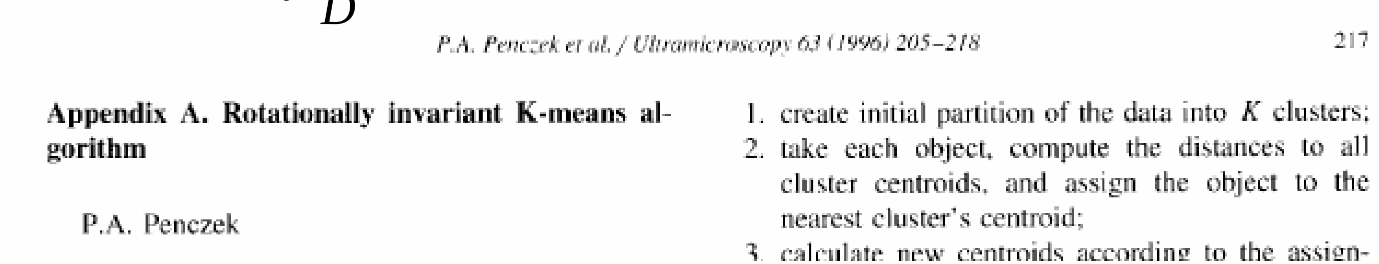


k averages (clusters)

2-D multi-reference alignment

K-means clustering with the distance defined as a minimum Euclidean distance over the permissible range of values of rotation and translation.

$$d^2 = \min_{\alpha,s_x,s_y} \int_D \left| f\left(\mathbf{T}\left(\alpha,s_x,s_y\right)\mathbf{x}\right) - g\left(\mathbf{x}\right)\right|^2 d\mathbf{x}$$



P.A. Penczek et al. / Ultramicroscopy 63 (1996) 205–218 217

Appendix A. Rotationally invariant K-means algorithm

P.A. Penczek

1. create initial partition of the data into K clusters;
2. take each object, compute the distances to all cluster centroids, and assign the object to the nearest cluster's centroid;
3. calculate new centroids according to the assign-

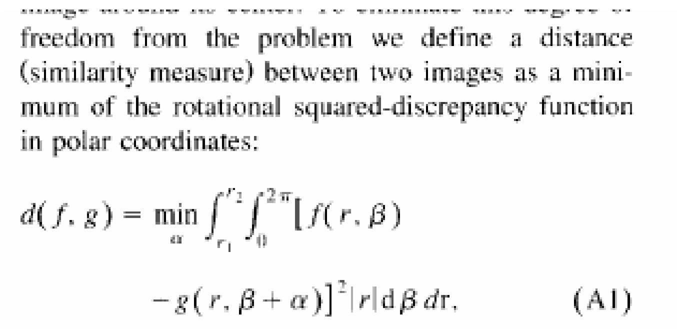
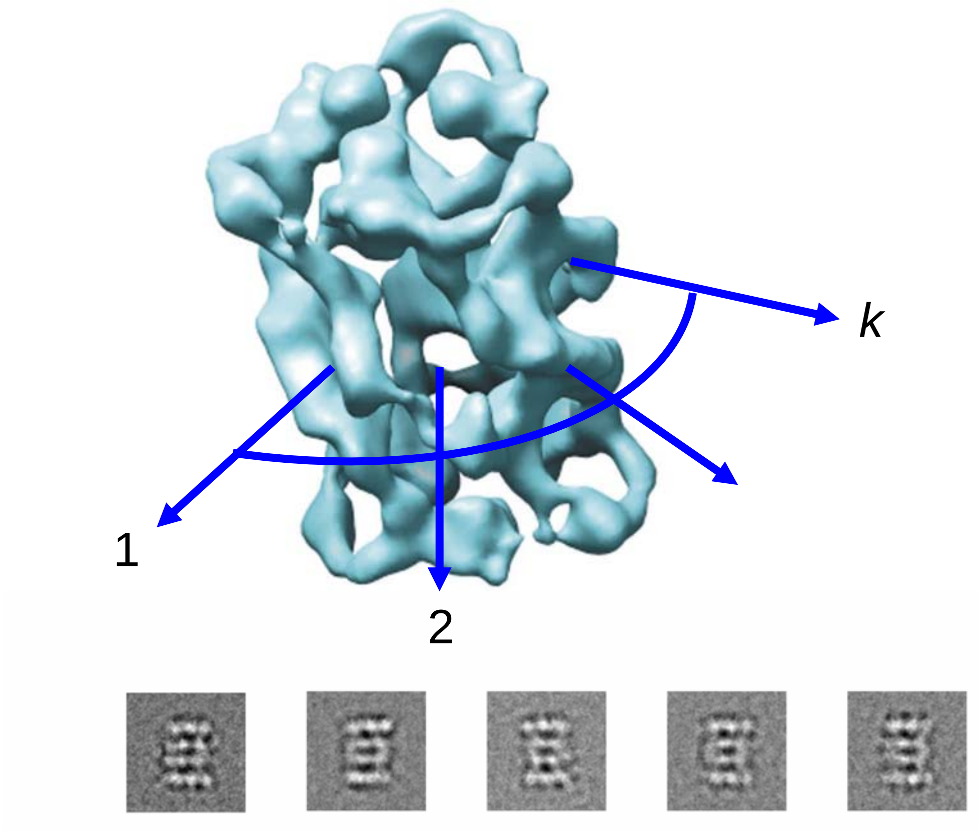


image around its center. To eliminate this degree of freedom from the problem we define a distance (similarity measure) between two images as a minimum of the rotational squared-discrepancy function in polar coordinates:

$$d(f,g) = \min_{\alpha} \int_{r_1}^{r_2} \int_0^{2\pi} \left[f(r,\beta) - g(r,\beta+\alpha) \right]^2 |r| \, \mathrm{d}\beta \, \mathrm{d}r, \qquad \text{(A1)}$$

Ab initio structure determination

Set of orthoaxial projections



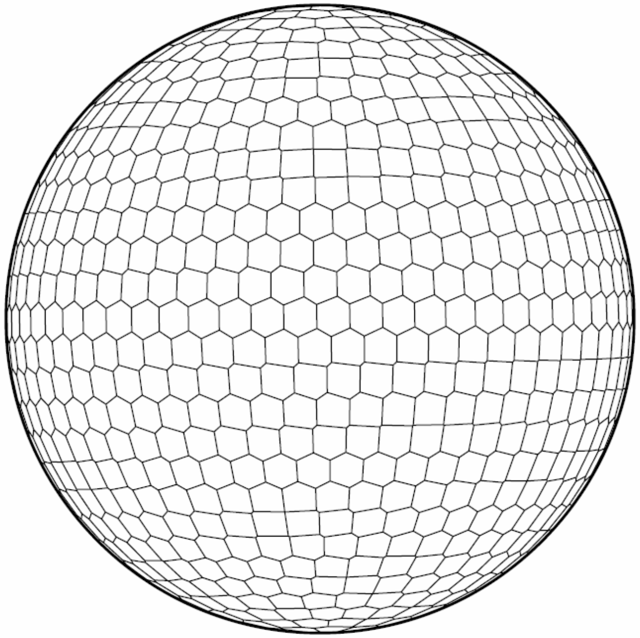
This is clustering problem with *k* orthoaxial projection directions spanning a Self Organizing 1D Map (a circle).

Interactions between *k* nodes are given by the overlap between projections in Fourier space.

Sidewinder (Phil Baldwin)
Pullan, L., […] Penczek, P. A., 2006. Structure 14, 661. Supplement

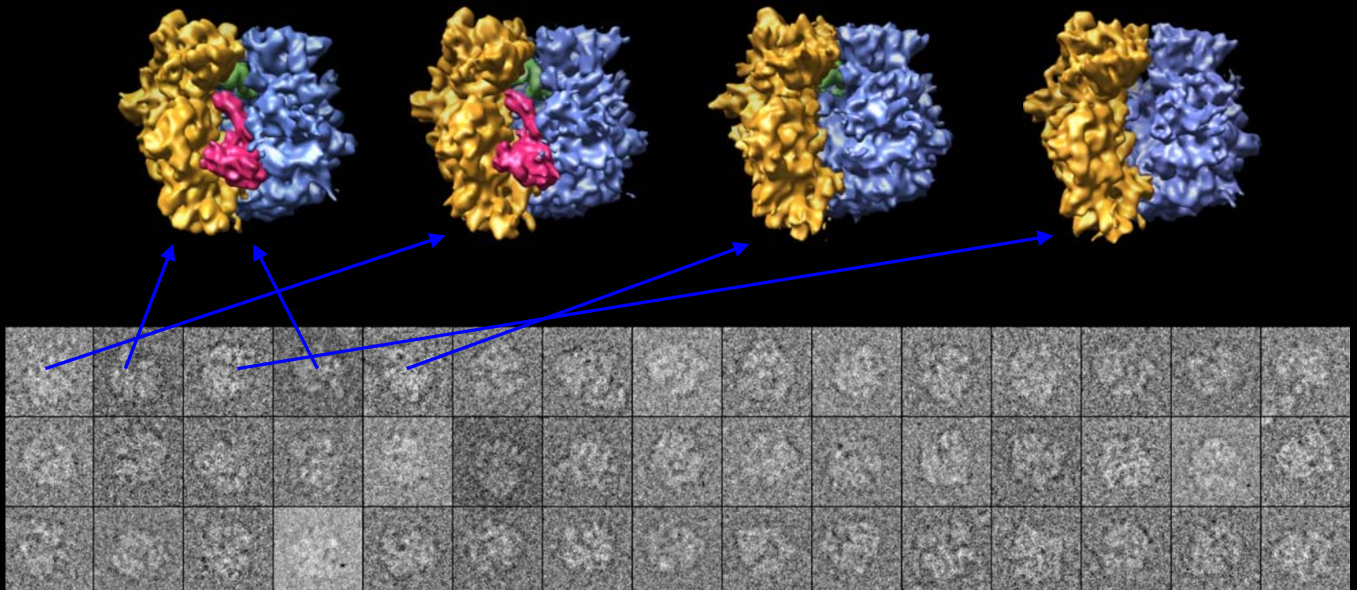
3-D projection matching

- For exhaustive search, the problem is discretized and a quasi-uniform set of *k* projection direction (clusters) is selected.
- *n* experimental projections have to be assigned to *k* projection directions using a similarity measure that is defined as a minimum distance over the permissible range of orientation parameters.
- The problem can be seen as SOM where interactions between nodes are adjustable and determined by the reconstruction algorithm.



3-D multi-reference alignment

- *k* 3-D structures (class averages)
- *n* experimental projections have to be assigned to *k* structures.



3-D multi-reference alignment

- *k* 3-D structures (class averages)
- *n* experimental projections have to be assigned to *k* structures.

In fact, the problem of 3-D multi-reference alignments has three levels:

1. *K*-means of assigning *n* experimental projections to *k* structures.

2. 2-D alignments of subsets of projections assigned to the same structure and projection direction.

3. K-means of assigning a subset of *m* experimental projections to *p* projection directions for a given structure.

Neither of these problems can be solved independently, so a likelihood of finding a good solution for the combination of three is slim.

Conclusions

- *Clustering* is the process of identifying natural groupings in the data; however, the notion of what constitutes a group (or a cluster) can be subjective.
- Clustering algorithms provide fast insight into structure in the data (data mining).
- Clustering algorithms can be heuristic (hierarchical, moving averages) or seek to minimize a functional defining a notion of a partition (Sum-of-Squared Error K-Means).
- There are no clustering algorithms that would guarantee optimum partition of the data, even if the goal is mathematically defined.
- All key steps of single particle analysis can be seen as attempts to cluster the data – this not only underlines complexity of the problem, but also provides inspiration for the development of new, robust approaches.