Algorithms

Clustering

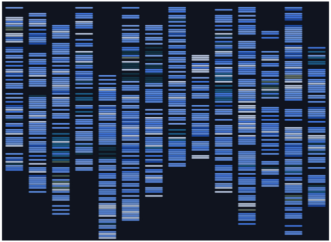
Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2019/2020 · April 30, 2020

Motivation

Example: Discover groups of genes with similar functions, regardless of their particular role.



Motivation

Example: Discover groups of genes with similar functions, regardless of their particular role.

Clustering problem: To partition a set of experimental data into groups (clusters) such that:

- Data points within the same cluster are highly similar.
- Data points in different clusters are very different.

This is a common problem in biology and other data sciences.

Motivation

Example: Discover groups of genes with similar functions, regardless of their particular role.

Clustering problem: To partition a set of experimental data into groups (clusters) such that:

- Data points within the same cluster are highly similar.
- Data points in different clusters are very different.

This is a common problem in biology and other data sciences.

What is a "good cluster"?
What does it mean to be "similar"?

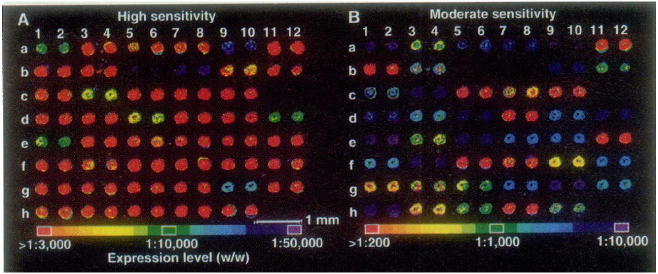
Example: Gene expression analysis

Sequence comparison can be misleading; genes with the same function may have no sequence similarity at all.

Example: Gene expression analysis

Sequence comparison can be misleading; genes with the same function may have no sequence similarity at all.

Instead, study the expression levels (the amount of mRNA produced) at different points in time:



Schena et al, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray", 1995

Example: Gene expression analysis

Sequence comparison can be misleading; genes with the same function may have no sequence similarity at all.

Instead, study the expression levels (the amount of mRNA produced) at different points in time:

Time	1 hr	2 hr	3hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

expression matrix: each number is some measured intensity

Schena et al, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray", 1995

Example: Gene expression analysis

Sequence comparison can be misleading; genes with the same function may have no sequence similarity at all.

Instead, study the expression levels (the amount of mRNA produced) at different points in time:

Time	1 hr	2 hr	3hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

similar rows $\approx$ related genes

expression matrix: each number is some measured intensity

Schena et al, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray", 1995

Distances

In order to group genes together, we need to define a distance between rows in the expression matrix.

Distances

In order to group genes together, we need to define a distance between rows in the expression matrix.

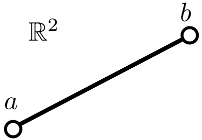
Each row is seen as a point in $\mathbb{R}^3$, e.g. $g_1 = (10.0, 8.0, 10.0)$.

Distances

In order to group genes together, we need to define a distance between rows in the expression matrix.

Each row is seen as a point in $\mathbb{R}^3$, e.g. $g_1 = (10.0, 8.0, 10.0)$.

Euclidean distance:

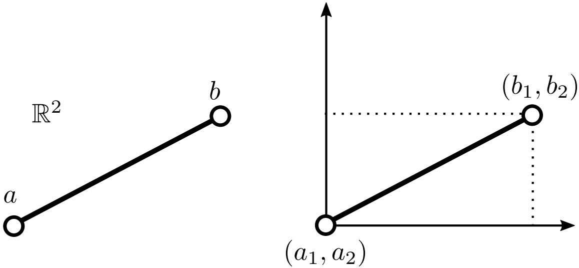


Distances

In order to group genes together, we need to define a distance between rows in the expression matrix.

Each row is seen as a point in $\mathbb{R}^3$, e.g. $g_1 = (10.0, 8.0, 10.0)$.

Euclidean distance:

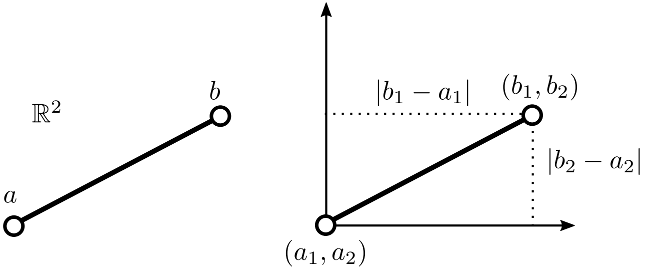


Distances

In order to group genes together, we need to define a distance between rows in the expression matrix.

Each row is seen as a point in $\mathbb{R}^3$, e.g. $g_1 = (10.0, 8.0, 10.0)$.

Euclidean distance:

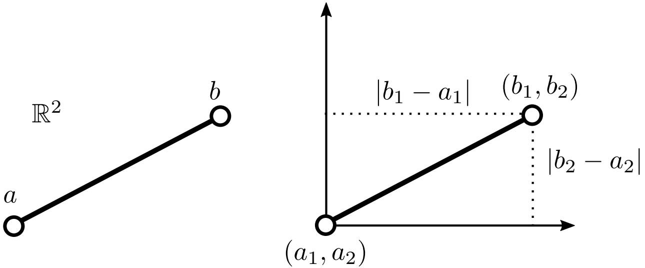


Distances

In order to group genes together, we need to define a distance between rows in the expression matrix.

Each row is seen as a point in $\mathbb{R}^3$, e.g. $g_1 = (10.0, 8.0, 10.0)$.

Euclidean distance:



Pythagoras' theorem: $d(a,b) = (|b_1 - a_1|^2 + |b_2 - a_2|^2)^{\frac{1}{2}}$

L_p distance

One can generalize to different power coefficients $p \geq 1$:

$$(|a_1 - b_1|^2 + |a_2 - b_2|^2)^{\frac{1}{2}}$$
$$\Downarrow$$
$$(|a_1 - b_1|^p + |a_2 - b_2|^p)^{\frac{1}{p}}$$

L_p distance

One can generalize to different power coefficients $p \geq 1$:

$$(|a_1 - b_1|^2 + |a_2 - b_2|^2)^{\frac{1}{2}}$$
$$\Downarrow$$
$$(|a_1 - b_1|^p + |a_2 - b_2|^p)^{\frac{1}{p}}$$

As well as generalize from 2 dimensions to k dimensions:

$$(\sum_{i=1}^{k} |a_i - b_i|^p)^{\frac{1}{p}}$$

L_p distance

One can generalize to different power coefficients $p \geq 1$:

$$(|a_1 - b_1|^2 + |a_2 - b_2|^2)^{\frac{1}{2}}$$
$$\Downarrow$$
$$(|a_1 - b_1|^p + |a_2 - b_2|^p)^{\frac{1}{p}}$$

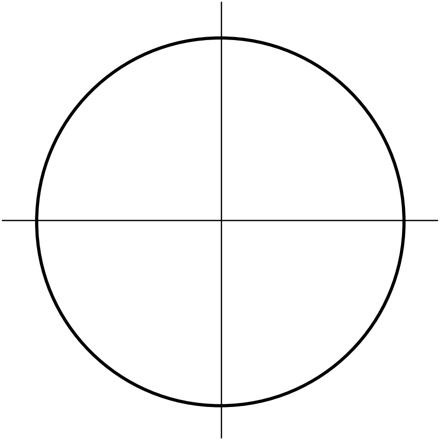
As well as generalize from 2 dimensions to k dimensions:

$$(\sum_{i=1}^{k} |a_i - b_i|^p)^{\frac{1}{p}}$$

This definition gives us the L_p distance between points in $\mathbb{R}^k$.

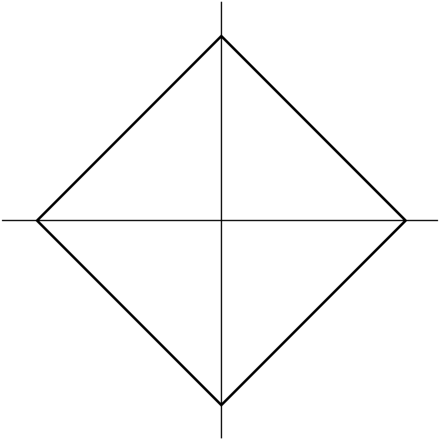
L_p unit balls in $\mathbb{R}^2$

For $p = 2$, we get the usual intuitive idea of a circle:



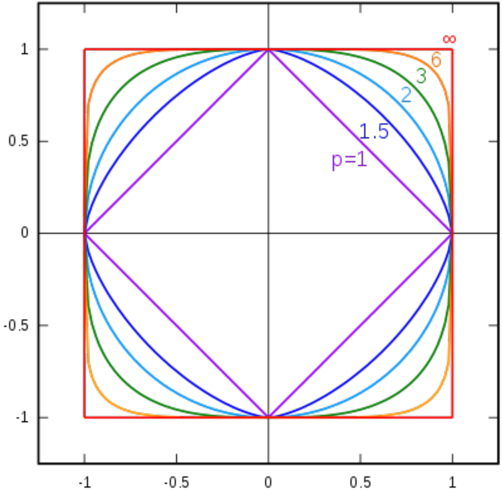
L_p unit balls in $\mathbb{R}^2$

For $p = 1$, we get a diamond-shaped boundary:



L_p unit balls in $\mathbb{R}^2$

For general $p \geq 1$, we get a general notion of "ball":



Example: Gene expression analysis

Consider the genes as points in $\mathbb{R}^3$:

Time	1 hr	2 hr	3hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

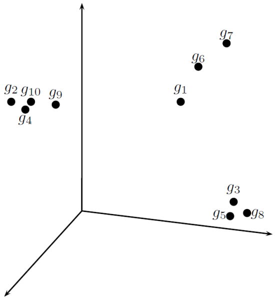
similar rows $\approx$ related genes

Example: Gene expression analysis

Consider the genes as points in $\mathbb{R}^3$:

Time	1 hr	2 hr	3hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

similar rows $\approx$ related genes



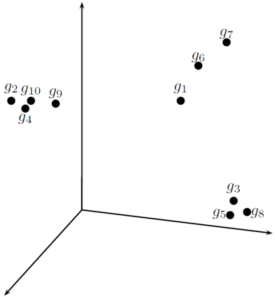
Example: Gene expression analysis

Consider the genes as points in $\mathbb{R}^3$:

Time	1 hr	2 hr	3hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

similar rows $\approx$ related genes

Let us look at the L_2 distances between each pair of genes:

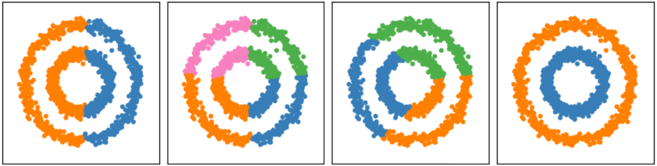


	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

distance matrix

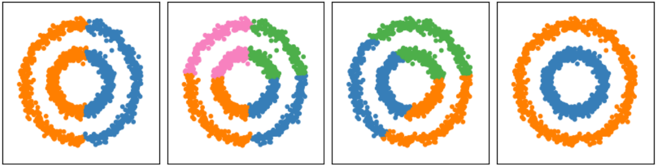
Good clusters

Data points can be clustered in many possible ways:



Good clusters

Data points can be clustered in many possible ways:

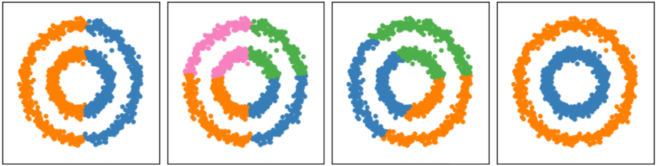


We need to define a quality criterion.

- $d(i,j)$ should be small if i and j belong to the same cluster.
- $d(i,j)$ should be large if i and j belong to a different cluster.

Good clusters

Data points can be clustered in many possible ways:



We need to define a quality criterion.

- $d(i,j)$ should be small if i and j belong to the same cluster.
- $d(i,j)$ should be large if i and j belong to a different cluster.

Also: how many clusters do we want to find?

k-means algorithm

Let us be given a set of n elements $\{x_1, \ldots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

k-means algorithm

Let us be given a set of n elements $\{x_1, \ldots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

Algorithm:

1. Choose k elements $\{c_1, \ldots, c_k\}$ (called seed).

k-means algorithm

Let us be given a set of n elements $\{x_1, \dots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

Algorithm:

1. Choose k elements $\{c_1, \dots, c_k\}$ (called seed).
2. For the remaining $n - k$ elements, find the nearest c_i for $i = 1 \dots k$.
   This generates k clusters.

k-means algorithm

Let us be given a set of n elements $\{x_1, \ldots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

Algorithm:

1. Choose k elements $\{c_1, \ldots, c_k\}$ (called seed).
2. For the remaining $n - k$ elements, find the nearest c_i for $i = 1 \ldots k$. This generates k clusters.
3. For each cluster, compute its centroid and assign it to c_i.

k-means algorithm

Let us be given a set of n elements $\{x_1, \ldots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

Algorithm:

1. Choose k elements $\{c_1, \ldots, c_k\}$ (called seed).
2. For the remaining $n - k$ elements, find the nearest c_i for $i = 1 \ldots k$. This generates k clusters.
3. For each cluster, compute its centroid and assign it to c_i.
4. Repeat from step (2).

k-means algorithm

Let us be given a set of n elements $\{x_1, \ldots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

Algorithm:

1. Choose k elements $\{c_1, \ldots, c_k\}$ (called seed).
2. For the remaining $n - k$ elements, find the nearest c_i for $i = 1 \ldots k$. This generates k clusters.
3. For each cluster, compute its centroid and assign it to c_i.
4. Repeat from step (2).

The algorithm alternates between assignment and centroid computation.

k-means algorithm

Let us be given a set of n elements $\{x_1, \ldots, x_n\}$ that we want to cluster.

Assume we know the number of clusters $k < n$ in advance.

Algorithm:

1. Choose k elements $\{c_1, \ldots, c_k\}$ (called seed).
2. For the remaining $n - k$ elements, find the nearest c_i for $i = 1 \ldots k$. This generates k clusters.
3. For each cluster, compute its centroid and assign it to c_i.
4. Repeat from step (2).

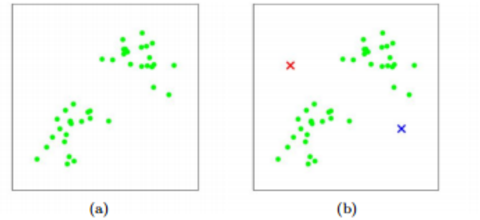
The algorithm alternates between assignment and centroid computation.

Termination criterion: For example, when the centroids stop changing.

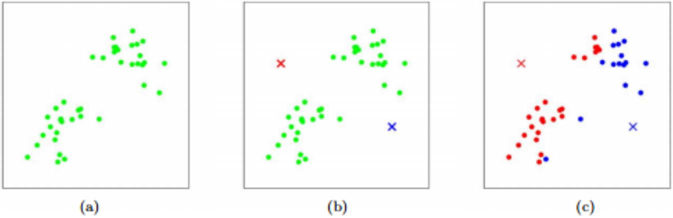
k-means algorithm: Example

(a)

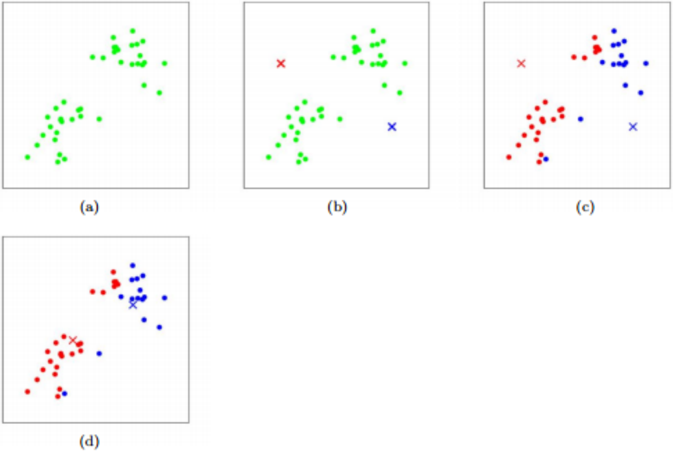
k-means algorithm: Example



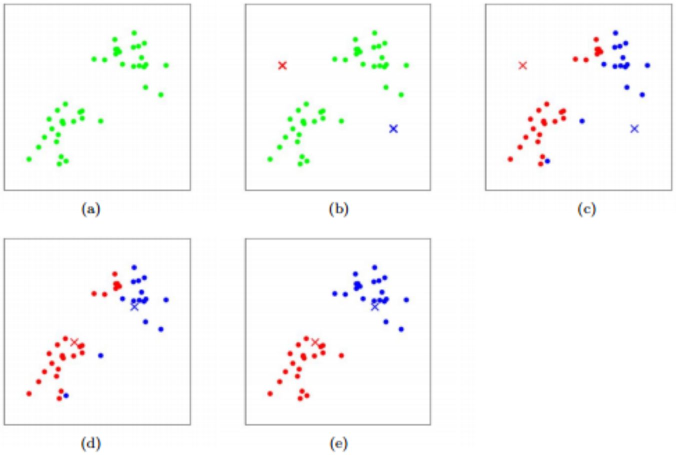
k-means algorithm: Example



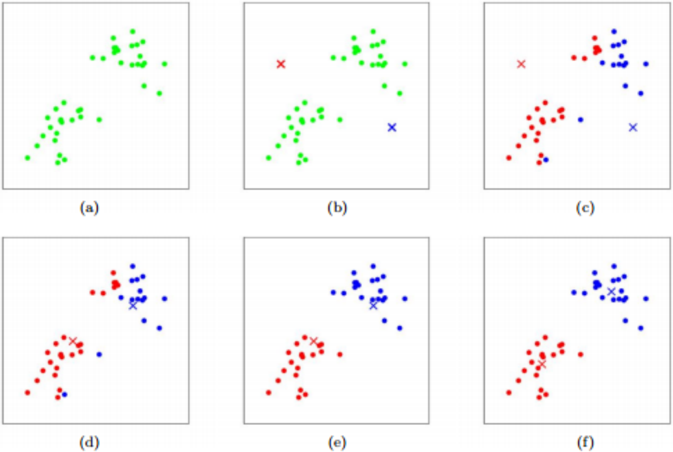
k-means algorithm: Example



k-means algorithm: Example



k-means algorithm: Example



k-means algorithm: Details

The seed is often chosen randomly.

k-means algorithm: Details

The seed is often chosen randomly.

Multi-start strategy: Run the algorithm many times with different seeds, keep the best solution (e.g. minimum intra-cluster average distance).

k-means algorithm: Details

The seed is often chosen randomly.

Multi-start strategy: Run the algorithm many times with different seeds, keep the best solution (e.g. minimum intra-cluster average distance).

The centroid c_i of the i-th cluster $\{y_1, \ldots, y_m\}$ is simply the mean:

$$c_i = \frac{1}{m} \sum_{j=1}^{m} y_j$$

where the summation and division are applied for each coordinate.

k-means algorithm: Details

The seed is often chosen randomly.

Multi-start strategy: Run the algorithm many times with different seeds, keep the best solution (e.g. minimum intra-cluster average distance).

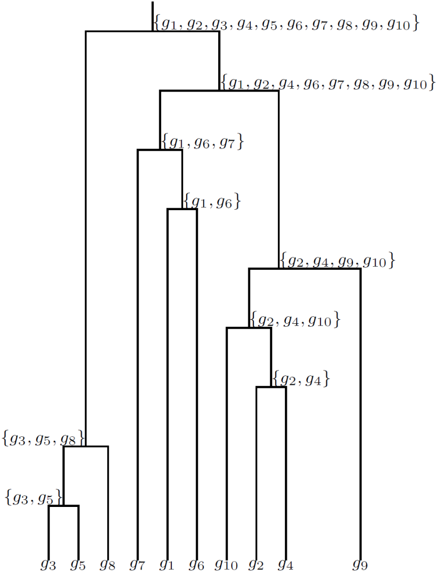
The centroid c_i of the i-th cluster $\{y_1, \ldots, y_m\}$ is simply the mean:

$$c_i = \frac{1}{m} \sum_{j=1}^{m} y_j$$

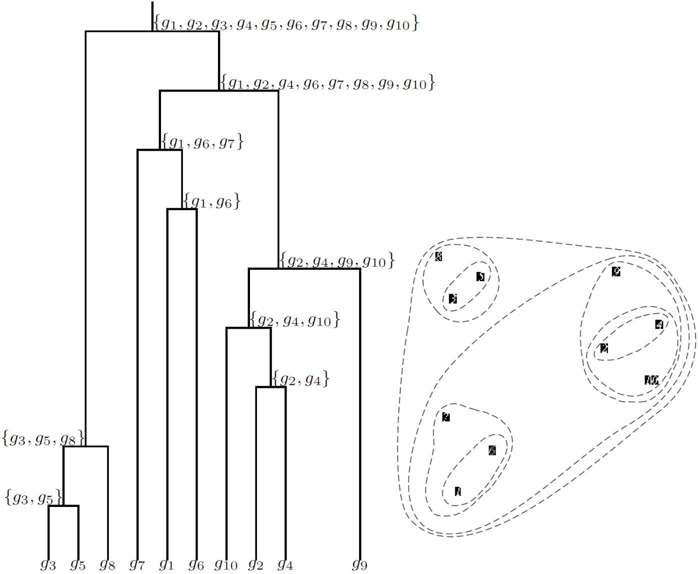
where the summation and division are applied for each coordinate.

Termination: Check if the cluster assignment does not change anymore, or if centroids stop moving significantly.

Hierarchical clustering



Hierarchical clustering



Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.
- Each edge has a length.

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.
- Each edge has a length.
- Shortest paths between leaves are entries in the distance matrix.

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.
- Each edge has a length.
- Shortest paths between leaves are entries in the distance matrix.

Overall idea:

- Given a distance matrix, it progressively generates n clusters.

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.
- Each edge has a length.
- Shortest paths between leaves are entries in the distance matrix.

Overall idea:

- Given a distance matrix, it progressively generates n clusters.
- The largest cluster has n single-element clusters (the leaves).

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.
- Each edge has a length.
- Shortest paths between leaves are entries in the distance matrix.

Overall idea:

- Given a distance matrix, it progressively generates n clusters.
- The largest cluster has n single-element clusters (the leaves).
- The second-largest combines the two closest clusters from the largest.

  Thus, it has $n-1$ clusters.

Hierarchical clustering

Hierarchical clustering organizes elements into a tree where each level is a cluster and the leaves are the individual elements.

- Clusters are related via inclusion.
- Each edge has a length.
- Shortest paths between leaves are entries in the distance matrix.

Overall idea:

- Given a distance matrix, it progressively generates n clusters.
- The largest cluster has n single-element clusters (the leaves).
- The second-largest combines the two closest clusters from the largest.

  Thus, it has $n-1$ clusters.
- In general, the i-th cluster combines the two closest clusters from the $(i-1)$-th cluster.

Hierarchical clustering

HIERARCHICALCLUSTERING($\mathbf{d}, n$)

Form n clusters, each with 1 element

Construct a graph T by assigning an isolated vertex to each cluster

Hierarchical clustering

```
HIERARCHICALCLUSTERING(d, n)
Form n clusters, each with 1 element
Construct a graph T by assigning an isolated vertex to each cluster
while there is more than 1 cluster
      Find the two closest clusters C_1 and C_2
return T
```

Hierarchical clustering

HIERARCHICALCLUSTERING$(\mathbf{d}, n)$

Form n clusters, each with 1 element
Construct a graph T by assigning an isolated vertex to each cluster
while there is more than 1 cluster
    Find the two closest clusters C_1 and C_2
    Merge C_1 and C_2 into new cluster C with $|C_1| + |C_2|$ elements
return T

Hierarchical clustering

```
HierarchicalClustering(d, n)
Form n clusters, each with 1 element
Construct a graph T by assigning an isolated vertex to each cluster
while there is more than 1 cluster
    Find the two closest clusters C_1 and C_2
    Merge C_1 and C_2 into new cluster C with |C_1| + |C_2| elements
    Compute distance from C to all other clusters



return T
```

As in k-means, different distance definitions yield different results.

Hierarchical clustering

HIERARCHICALCLUSTERING($\mathbf{d}, n$)

Form n clusters, each with 1 element
Construct a graph T by assigning an isolated vertex to each cluster
while there is more than 1 cluster
    Find the two closest clusters C_1 and C_2
    Merge C_1 and C_2 into new cluster C with $|C_1| + |C_2|$ elements
    Compute distance from C to all other clusters
    Add a new vertex C to T and connect to vertices C_1 and C_2
return T

As in k-means, different distance definitions yield different results.

Hierarchical clustering

HIERARCHICALCLUSTERING($\mathbf{d}, n$)

Form n clusters, each with 1 element
Construct a graph T by assigning an isolated vertex to each cluster
while there is more than 1 cluster
    Find the two closest clusters C_1 and C_2
    Merge C_1 and C_2 into new cluster C with $|C_1| + |C_2|$ elements
    Compute distance from C to all other clusters
    Add a new vertex C to T and connect to vertices C_1 and C_2
    Remove rows and columns of $\mathbf{d}$ corresponding to C_1 and C_2
    Add a row and column to $\mathbf{d}$ for the new cluster C
return T

As in k-means, different distance definitions yield different results.

Hierarchical clustering

HIERARCHICALCLUSTERING($\mathbf{d}, n$)

Form n clusters, each with 1 element
Construct a graph T by assigning an isolated vertex to each cluster
while there is more than 1 cluster
    Find the two closest clusters C_1 and C_2
    Merge C_1 and C_2 into new cluster C with $|C_1| + |C_2|$ elements
    Compute distance from C to all other clusters
    Add a new vertex C to T and connect to vertices C_1 and C_2
    Remove rows and columns of $\mathbf{d}$ corresponding to C_1 and C_2
    Add a row and column to $\mathbf{d}$ for the new cluster C
return T

As in k-means, different distance definitions yield different results.

At the end, we have one large cluster that contains all the others.

Exercises

Implement the k-means clustering algorithm.

Test for different values of k with the given data (see course webpage).

Send me your code + some screenshots of your results.

Suggested reading

Chapters 10.1, 10.2 and 10.3 of:

“An Introduction to Bioinformatics Algorithms”, Jones and Pevzner