Hierarchical clustering

Rebecca C. Steorts, Duke University

STA 325, Chapter 10 ISL

Agenda

- K-means versus Hierarchical clustering
- Agglomerative vs divisive clustering
- Dendogram (tree)
- Hierarchical clustering algorithm
- Single, Complete, and Average linkage
- Application to genomic (PCA versus Hierarchical clustering)

From K-means to Hierarchical clustering

Recall two properties of K-means clustering:

1. It fits exactly K clusters (as specified)
2. Final clustering assignment depends on the chosen initial cluster centers

- Assume pairwise dissimilarites d_{ij} between data points.
- Hierarchical clustering produces a consistent result, without the need to choose initial starting positions (number of clusters).

Catch: choose a way to measure the dissimilarity between groups, called the linkage

- Given the linkage, hierarchical clustering produces a sequence of clustering assignments.
- At one end, all points are in their own cluster, at the other end, all points are in one cluster

Agglomerative vs divisive clustering

Agglomerative (i.e., bottom-up):

- Start with all points in their own group
- Until there is only one cluster, repeatedly: merge the two groups that have the smallest dissimilarity

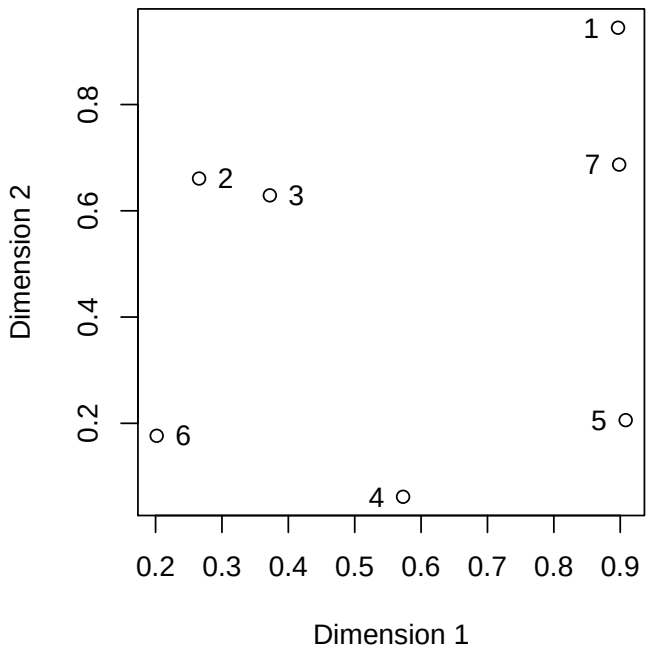
Divisive (i.e., top-down):

- Start with all points in one cluster
- Until all points are in their own cluster, repeatedly: split the group into two resulting in the biggest dissimilarity

Agglomerative strategies are simpler, we'll focus on them. Divisive methods are still important, but you can read about these on your own if you want to learn more.

Simple example

Given these data points, an agglomerative algorithm might decide on a clustering sequence as follows:



Step 1: $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}$;

Step 2: $\{1\}, \{2, 3\}, \{4\}, \{5\}, \{6\}, \{7\}$;

Step 3: $\{1, 7\}, \{2, 3\}, \{4\}, \{5\}, \{6\}$;

Step 4: $\{1, 7\}, \{2, 3\}, \{4, 5\}, \{6\}$;

Step 5: $\{1, 7\}, \{2, 3, 6\}, \{4, 5\}$;

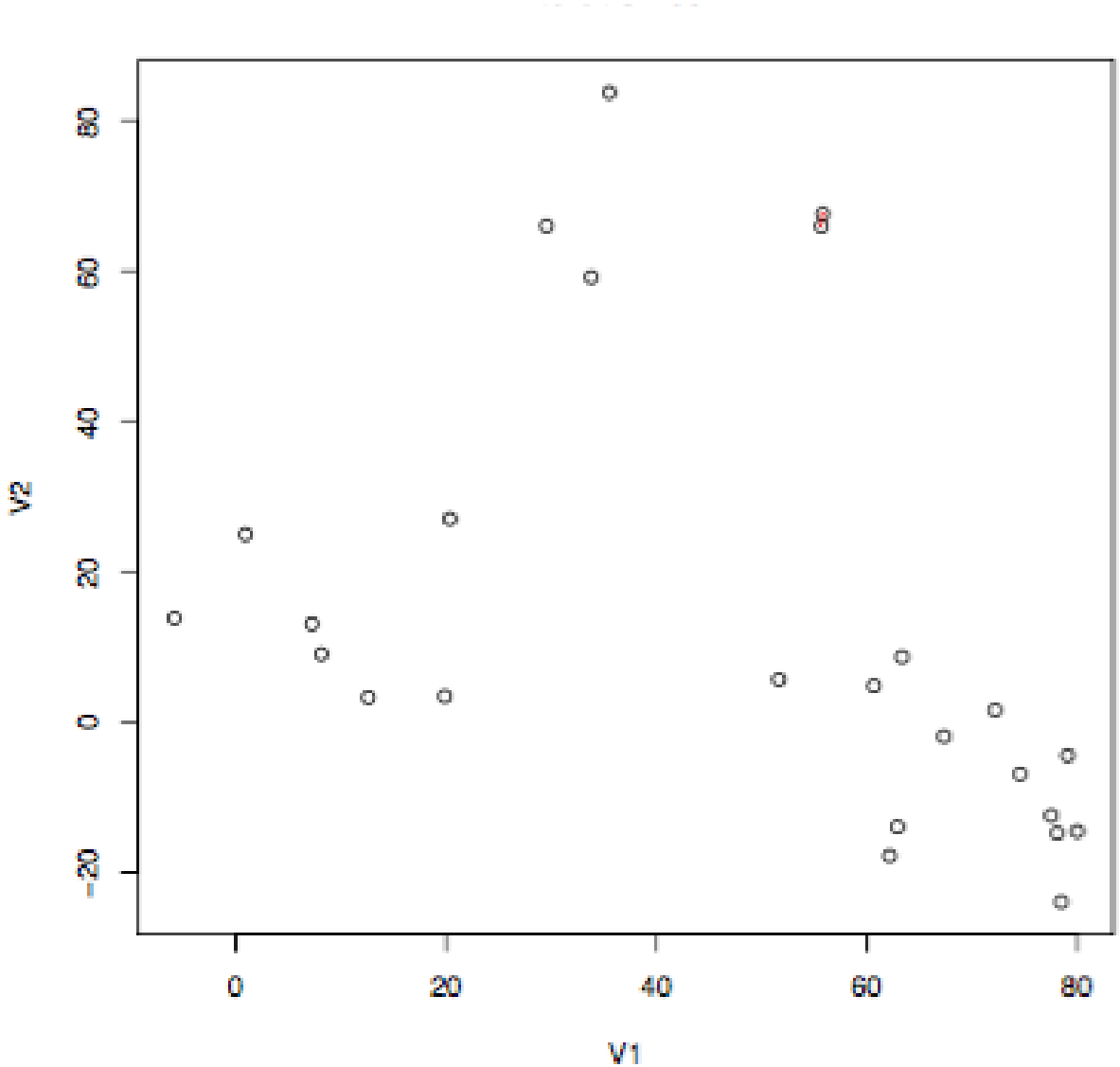
Step 6: $\{1, 7\}, \{2, 3, 4, 5, 6\}$;

Step 7: $\{1, 2, 3, 4, 5, 6, 7\}$.

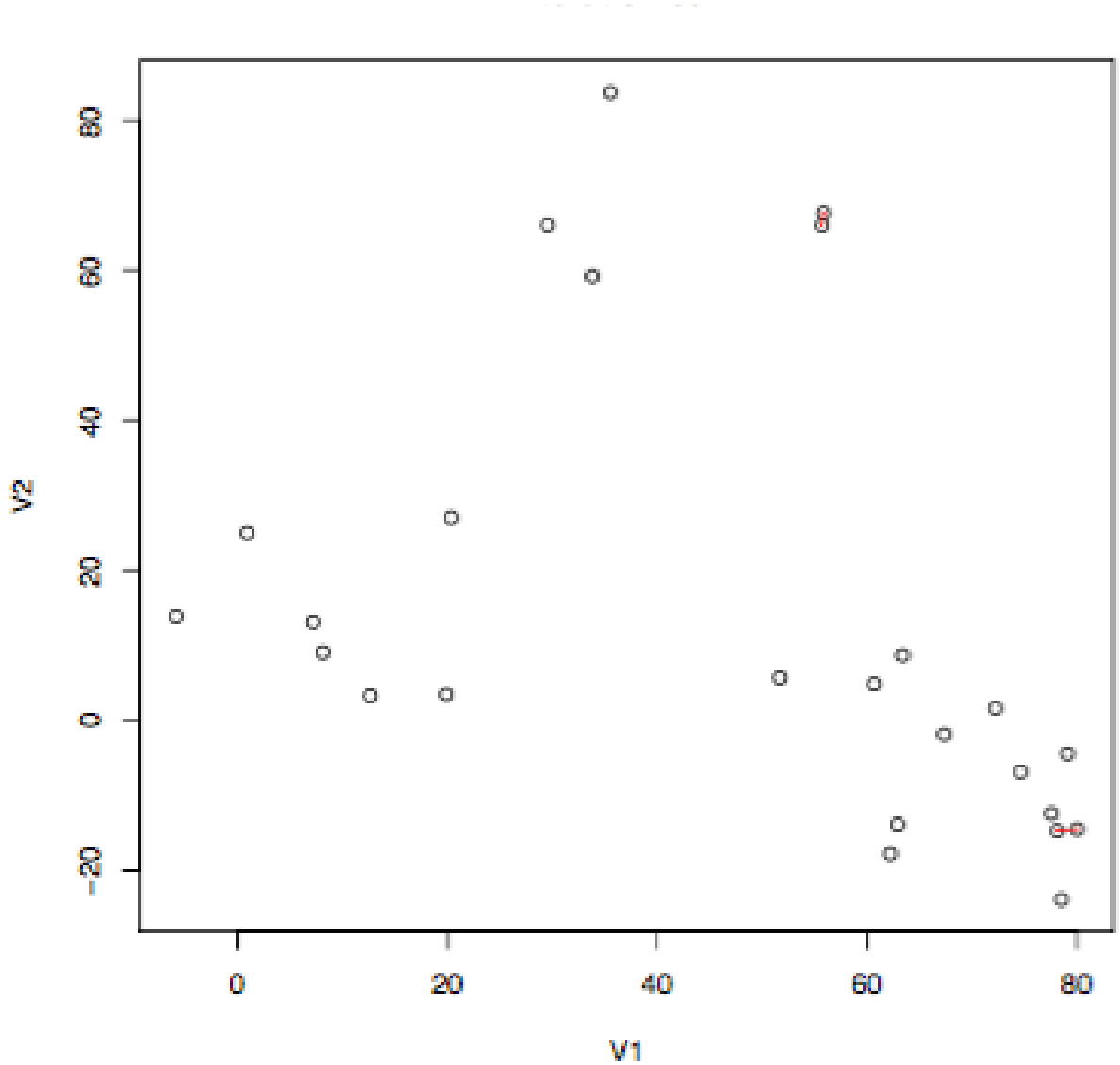
Algorithm

1. Place each data point into its own singleton group.
2. Repeat: iteratively merge the two closest groups
3. Until: all the data are merged into a single cluster

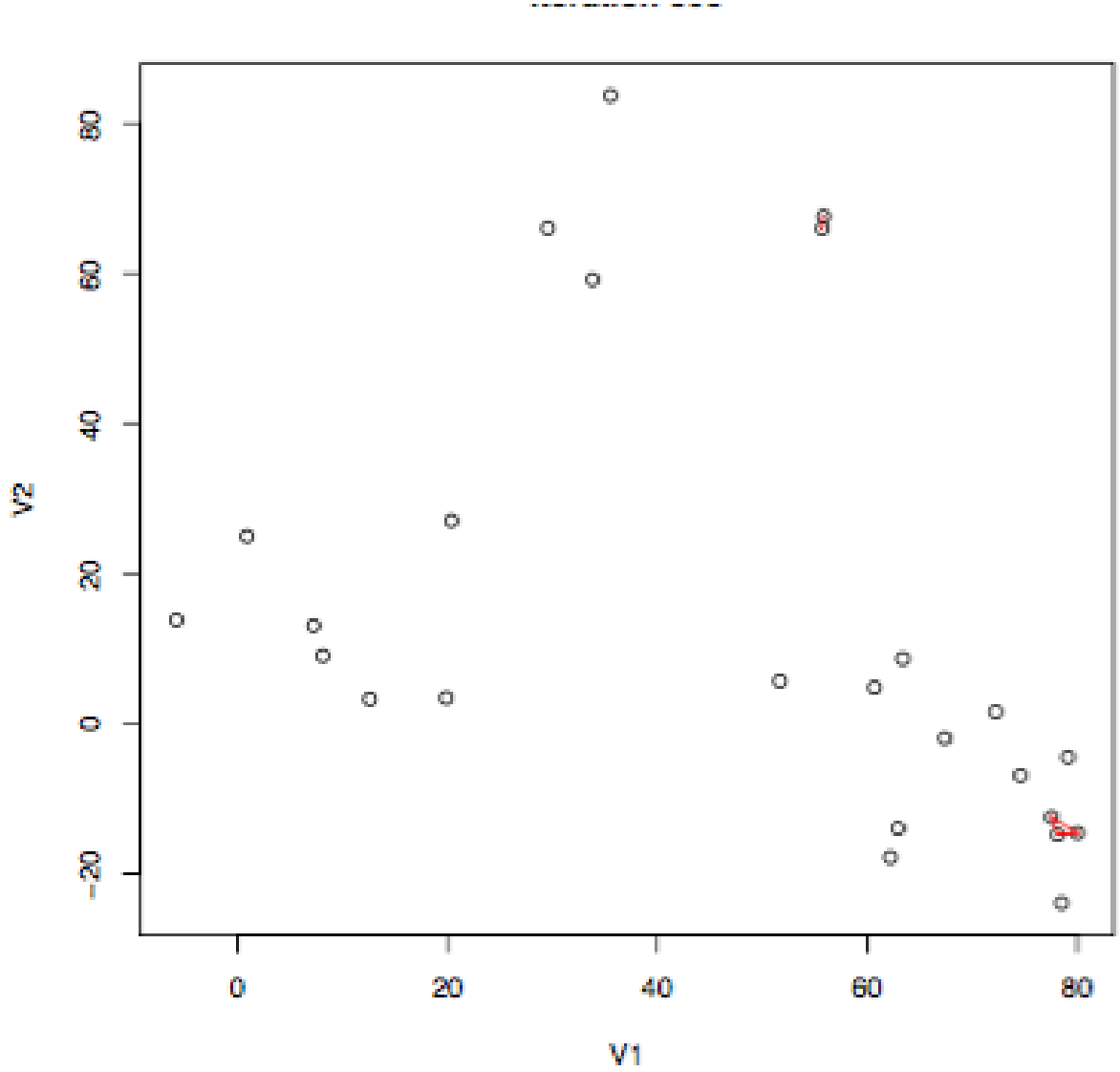
Example



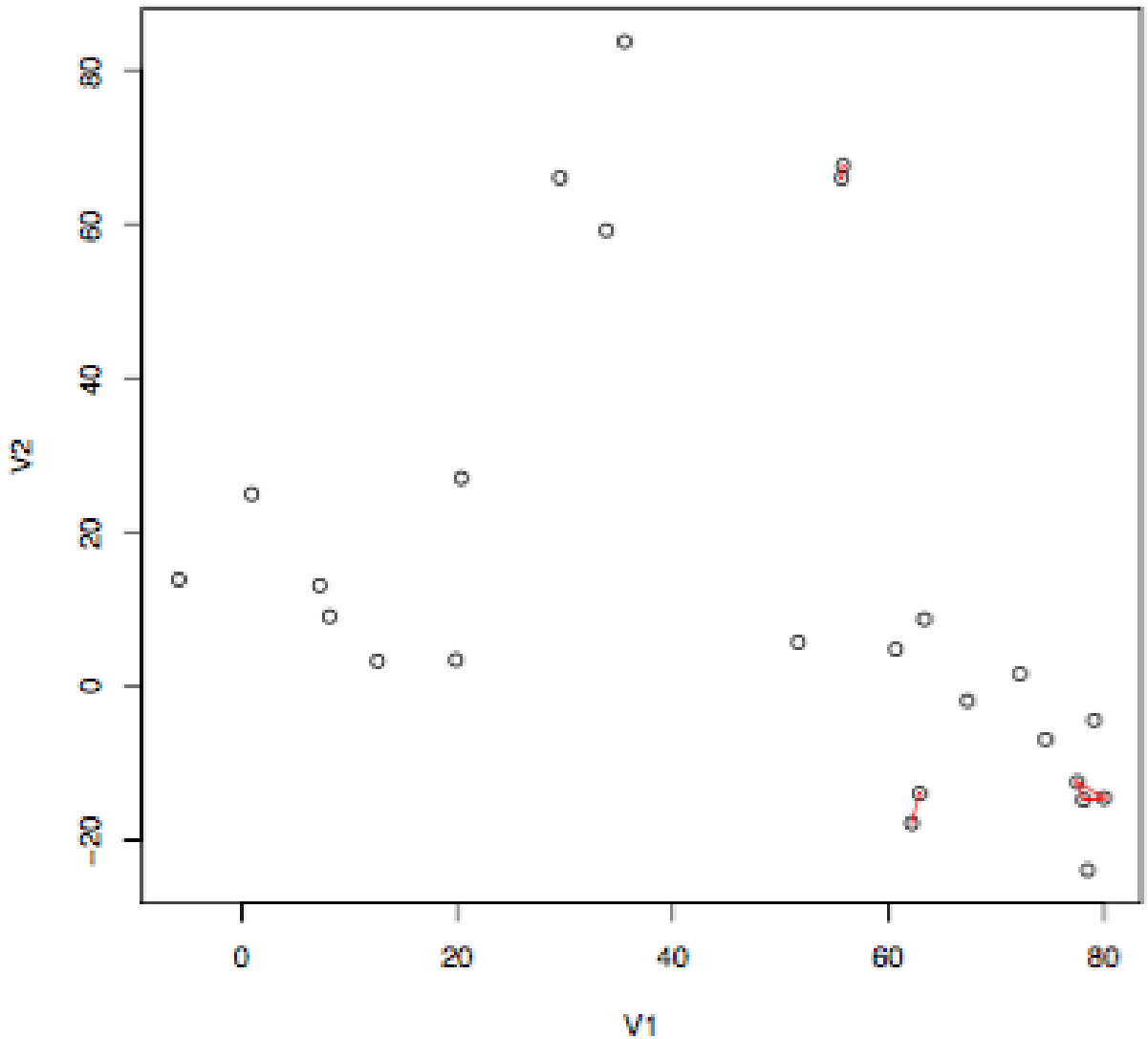
Iteration 2



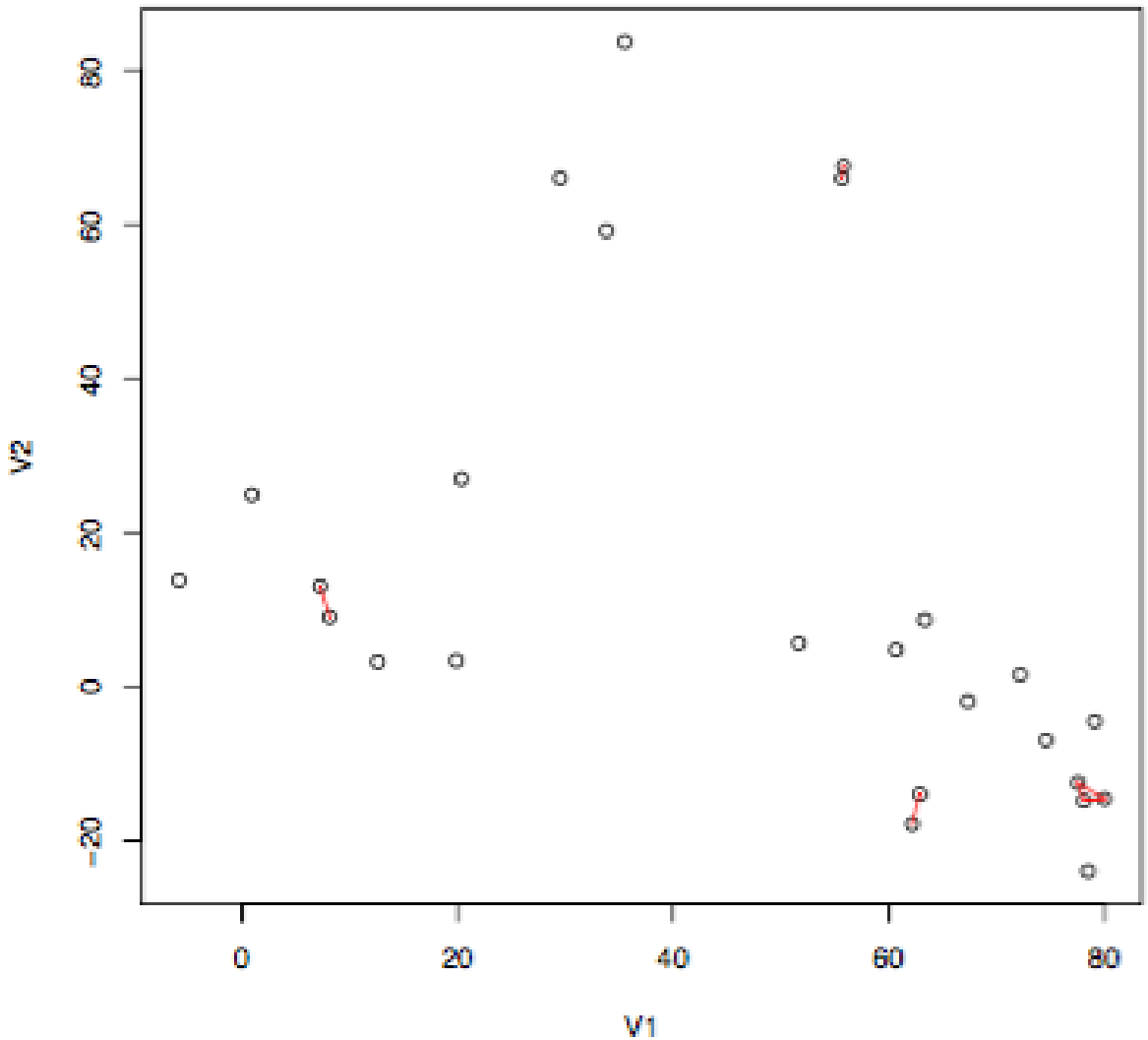
Iteration 3



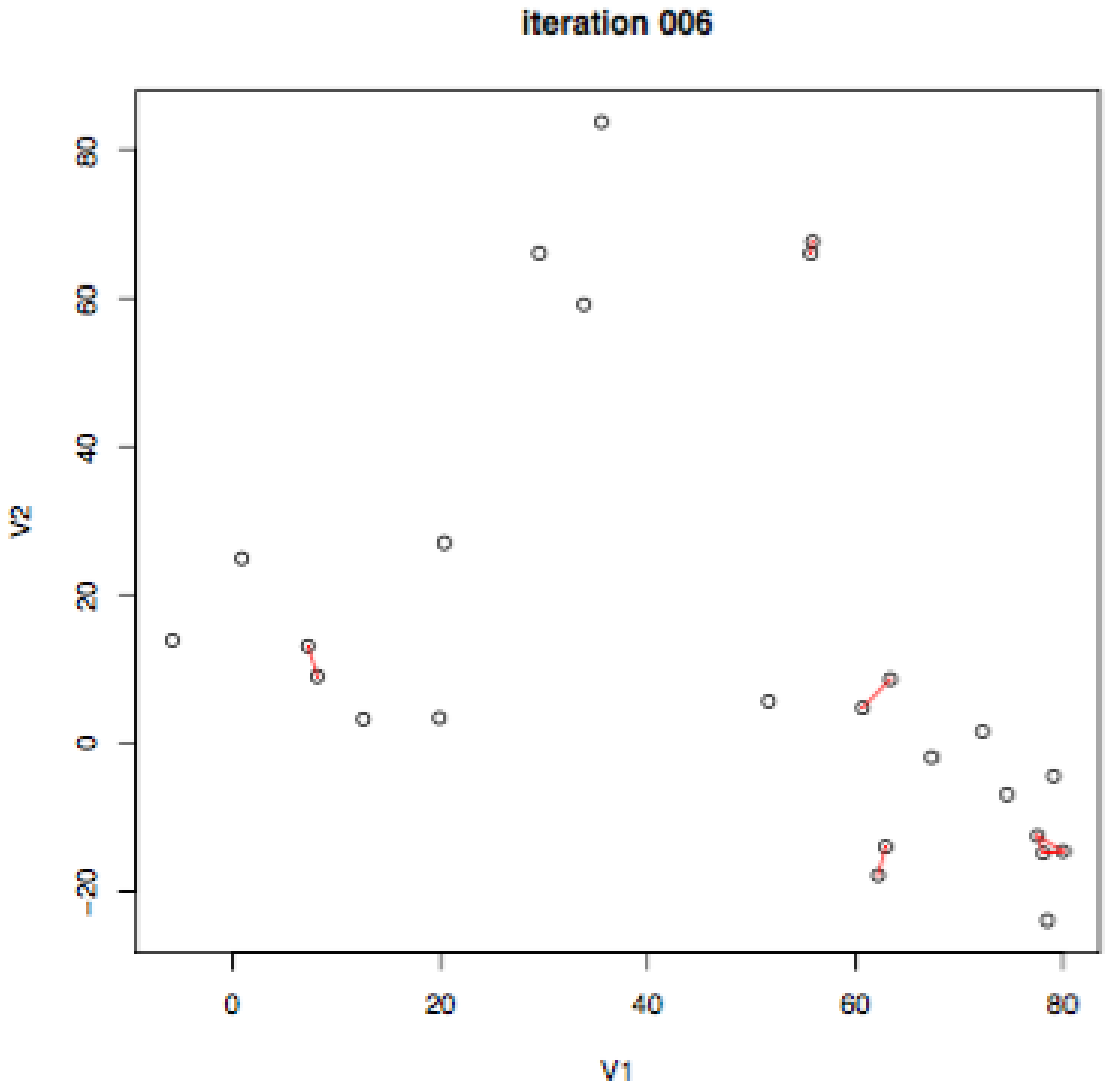
Iteration 4



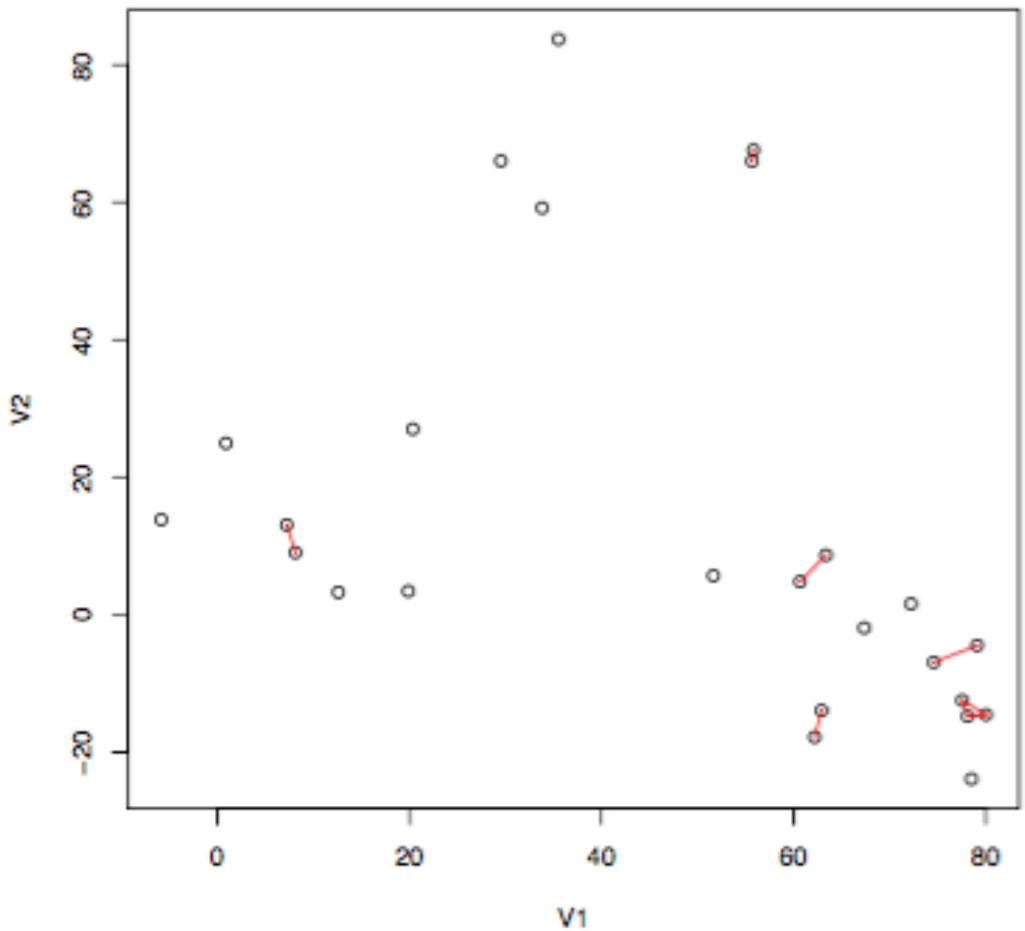
Iteration 5



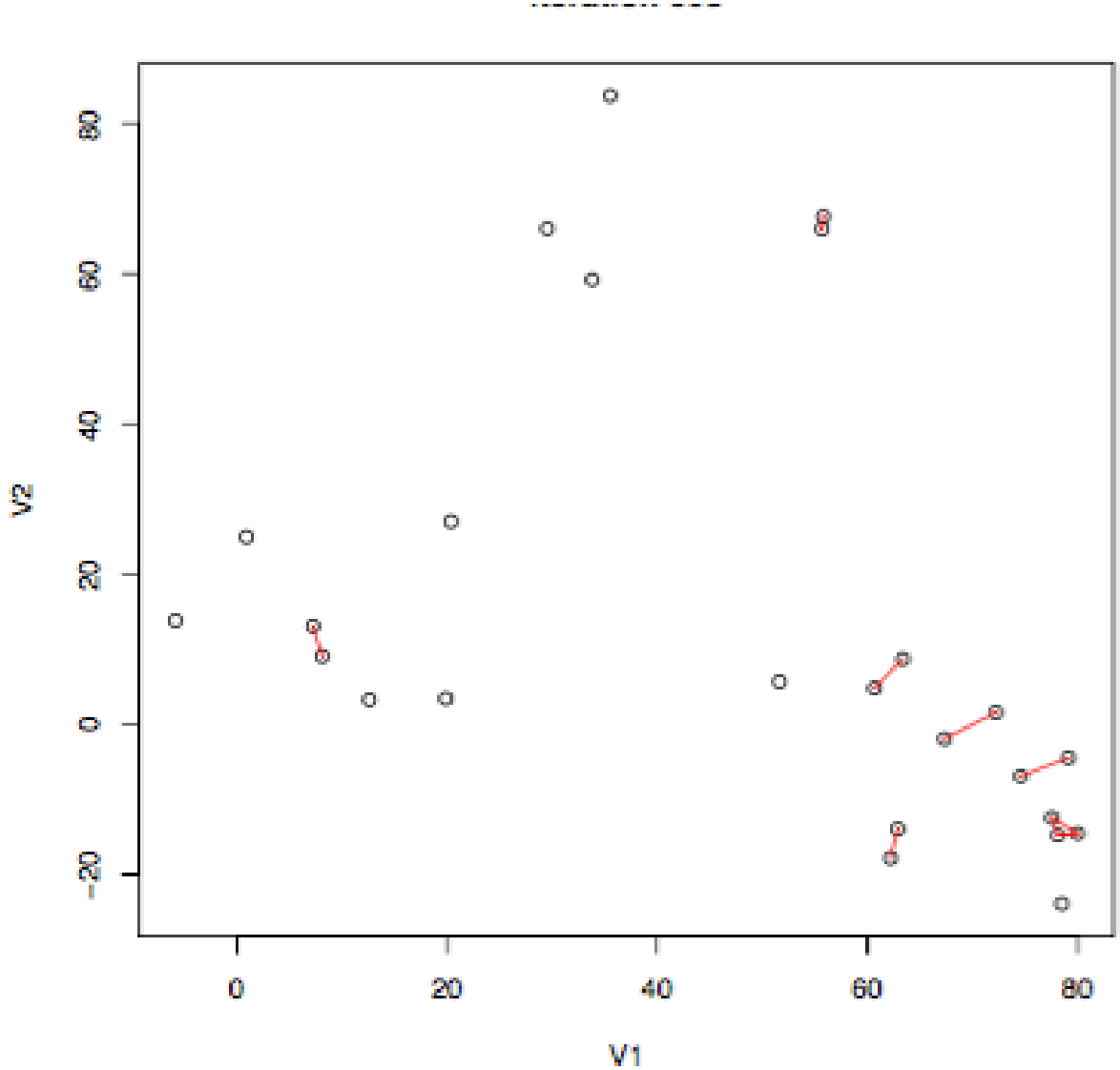
Iteration 6



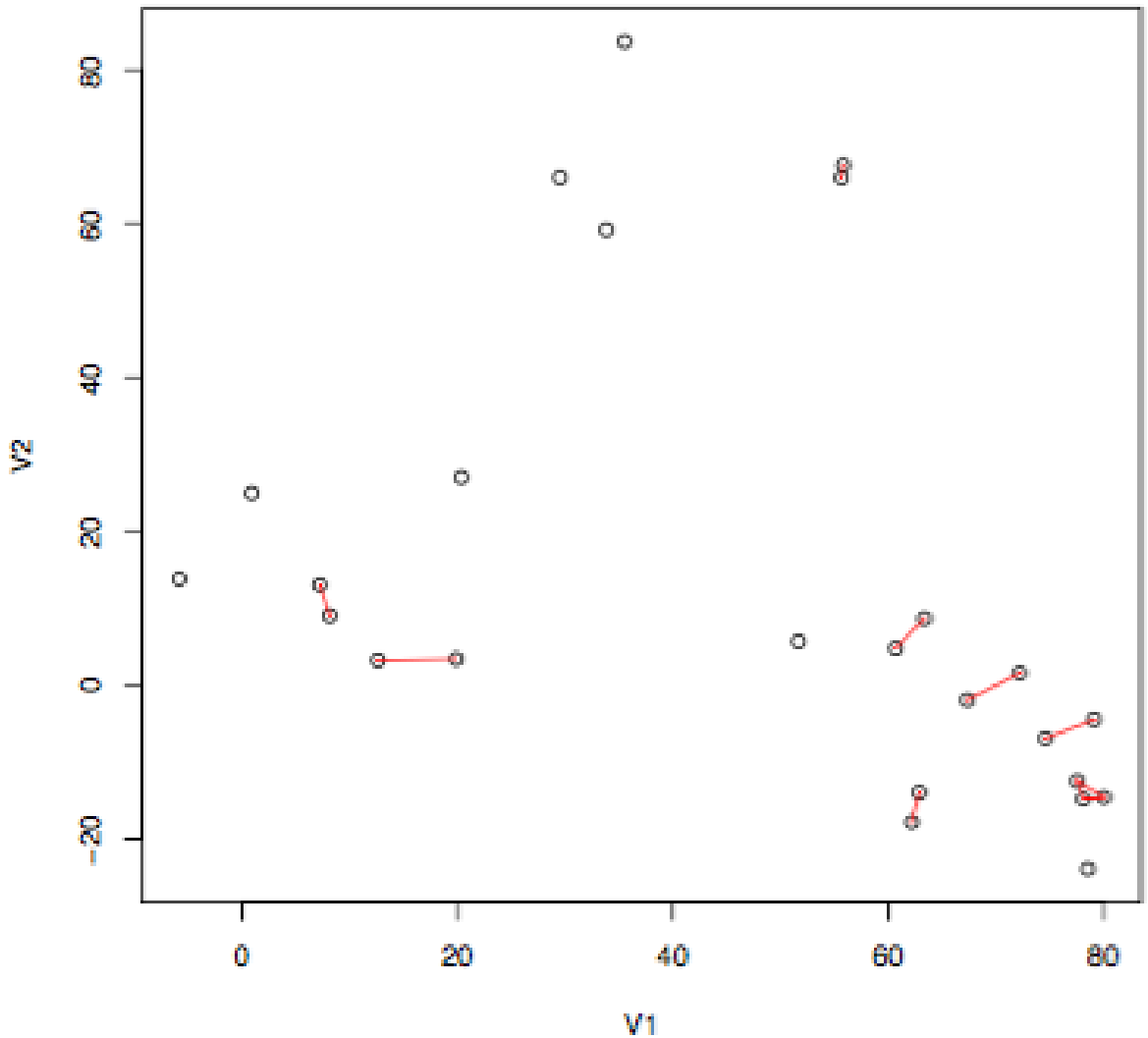
Iteration 7



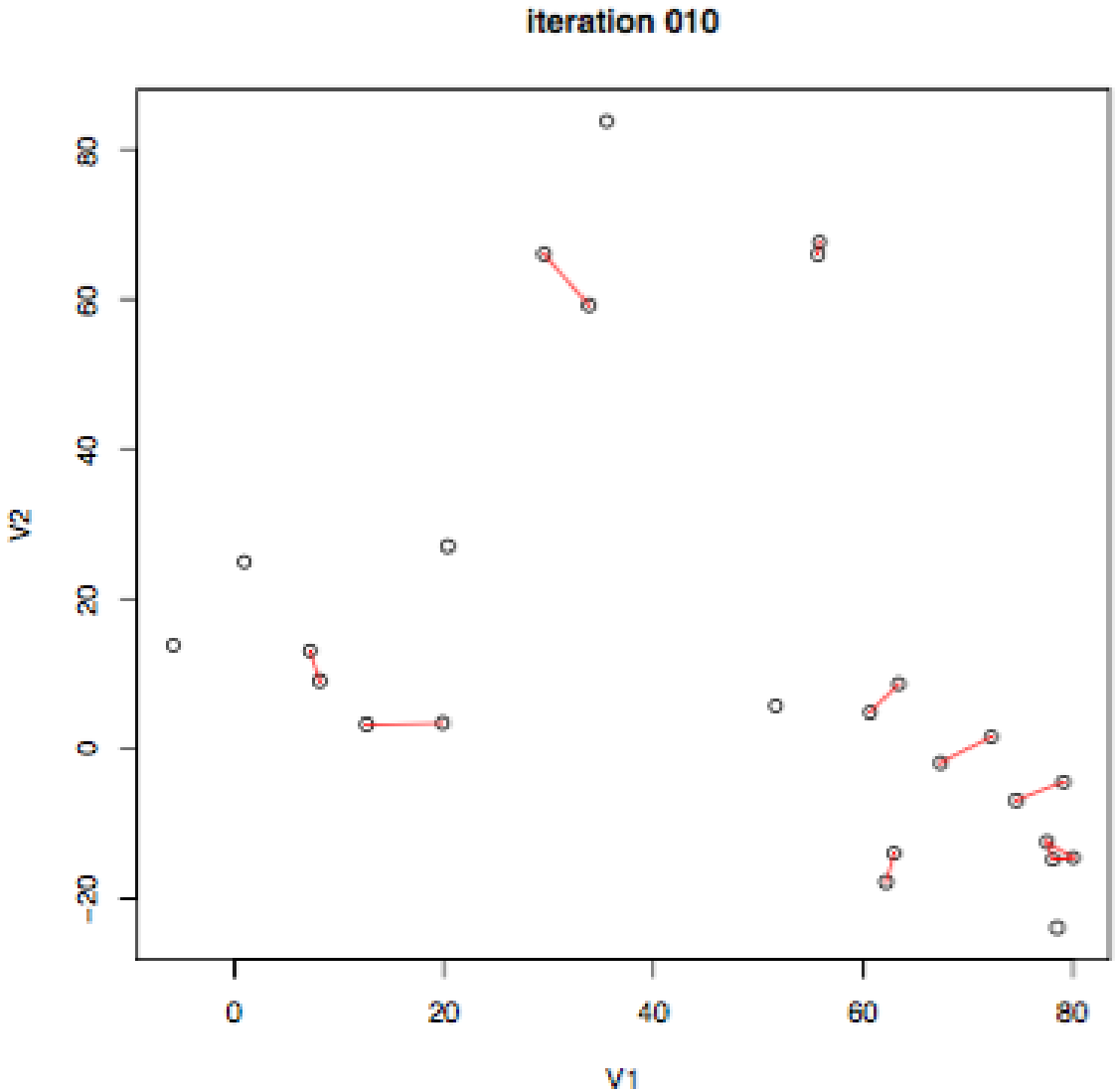
Iteration 8



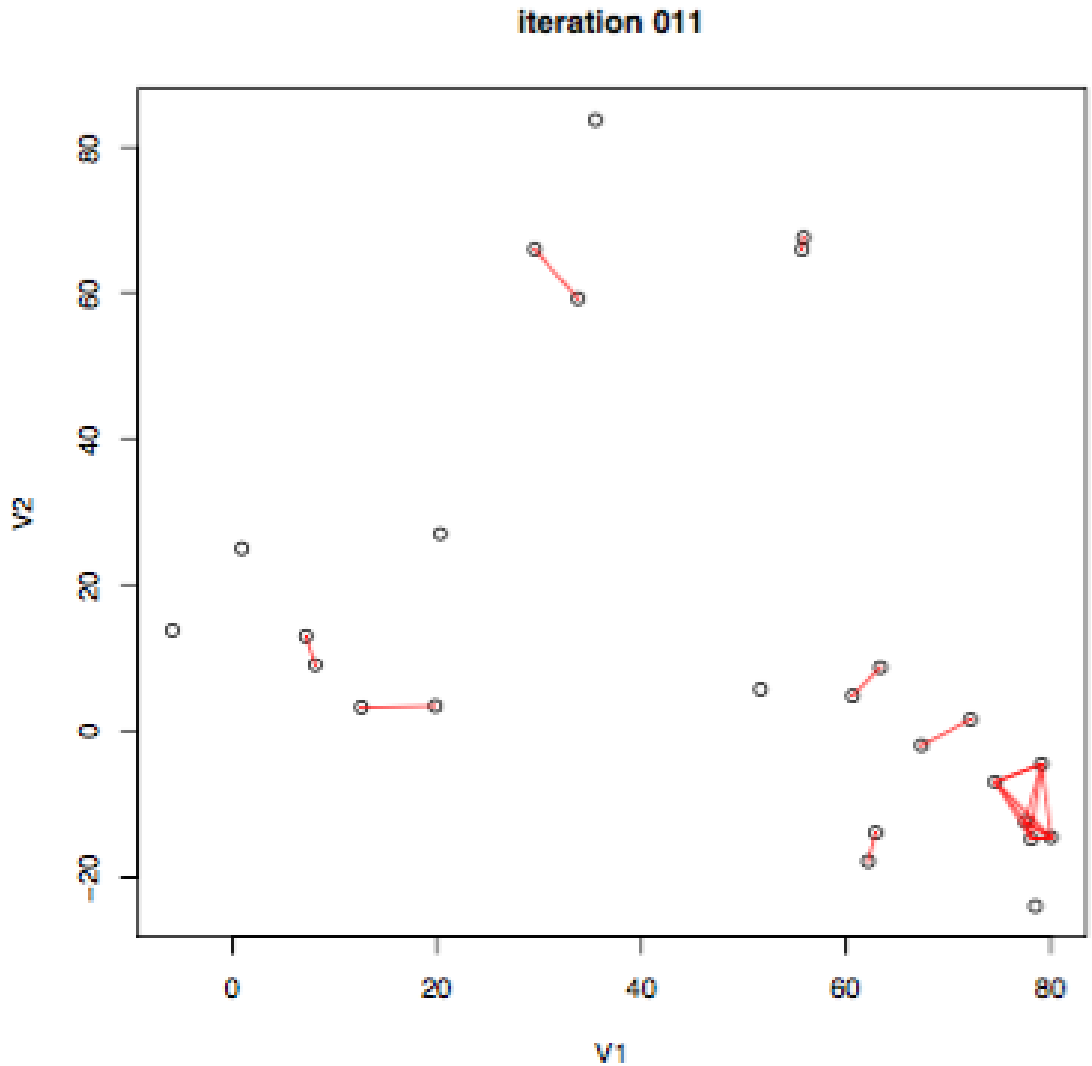
Iteration 9



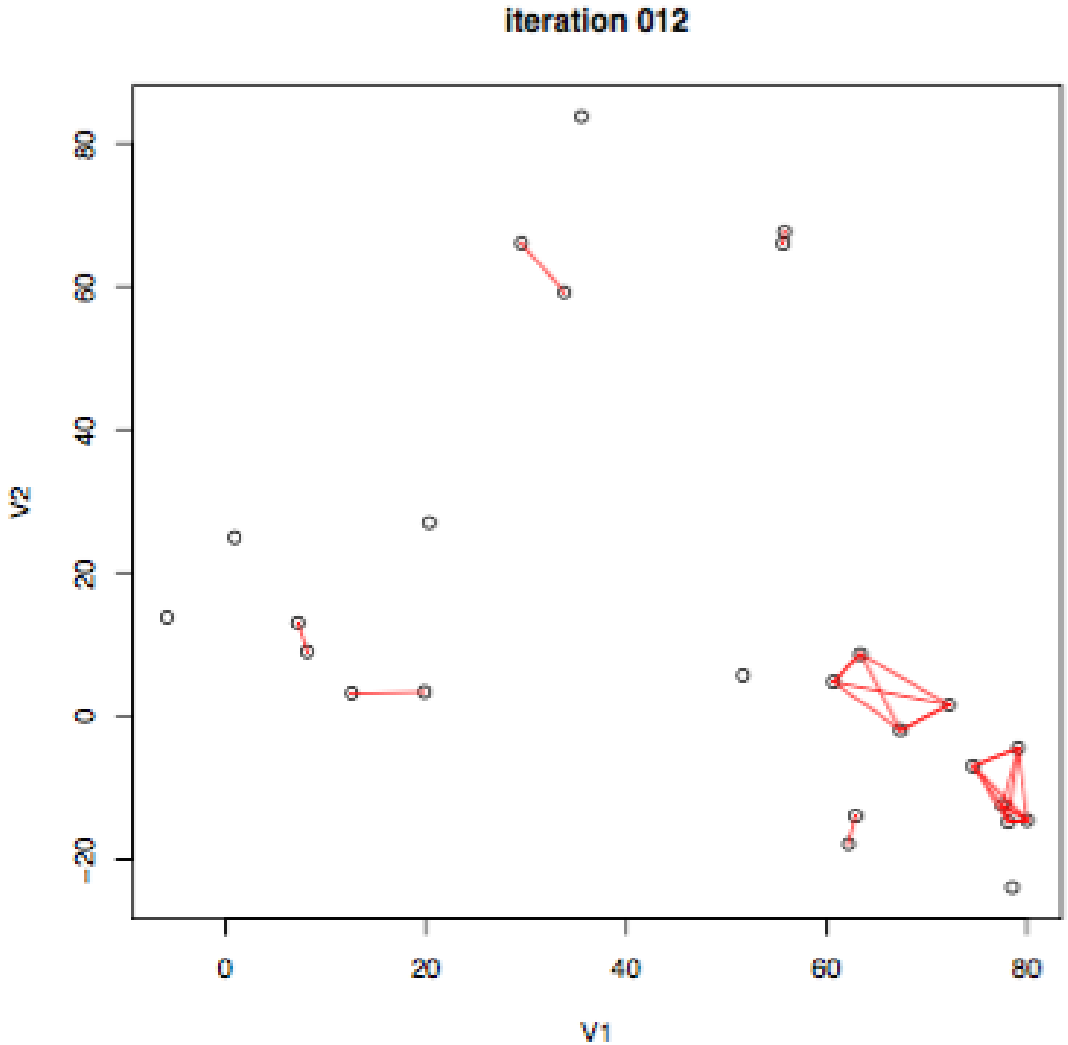
Iteration 10



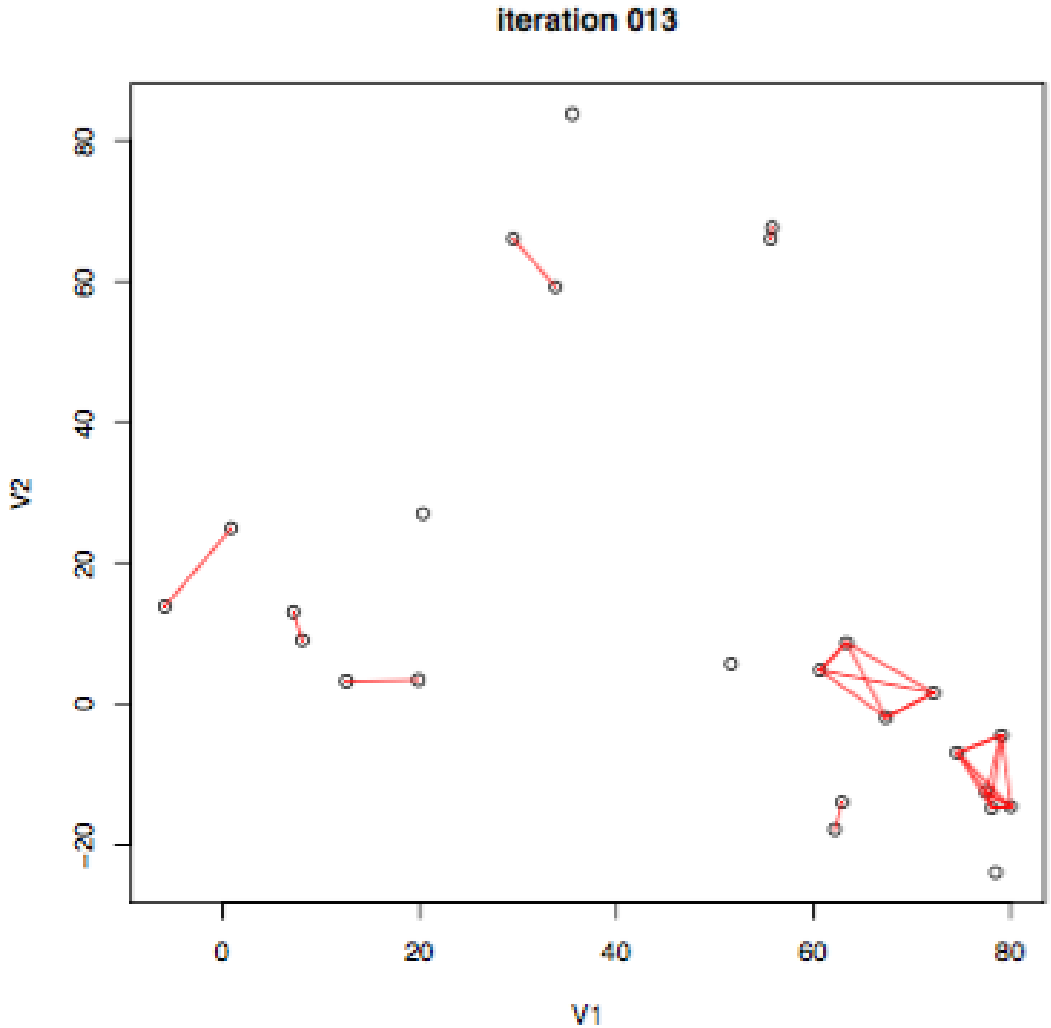
Iteration 11



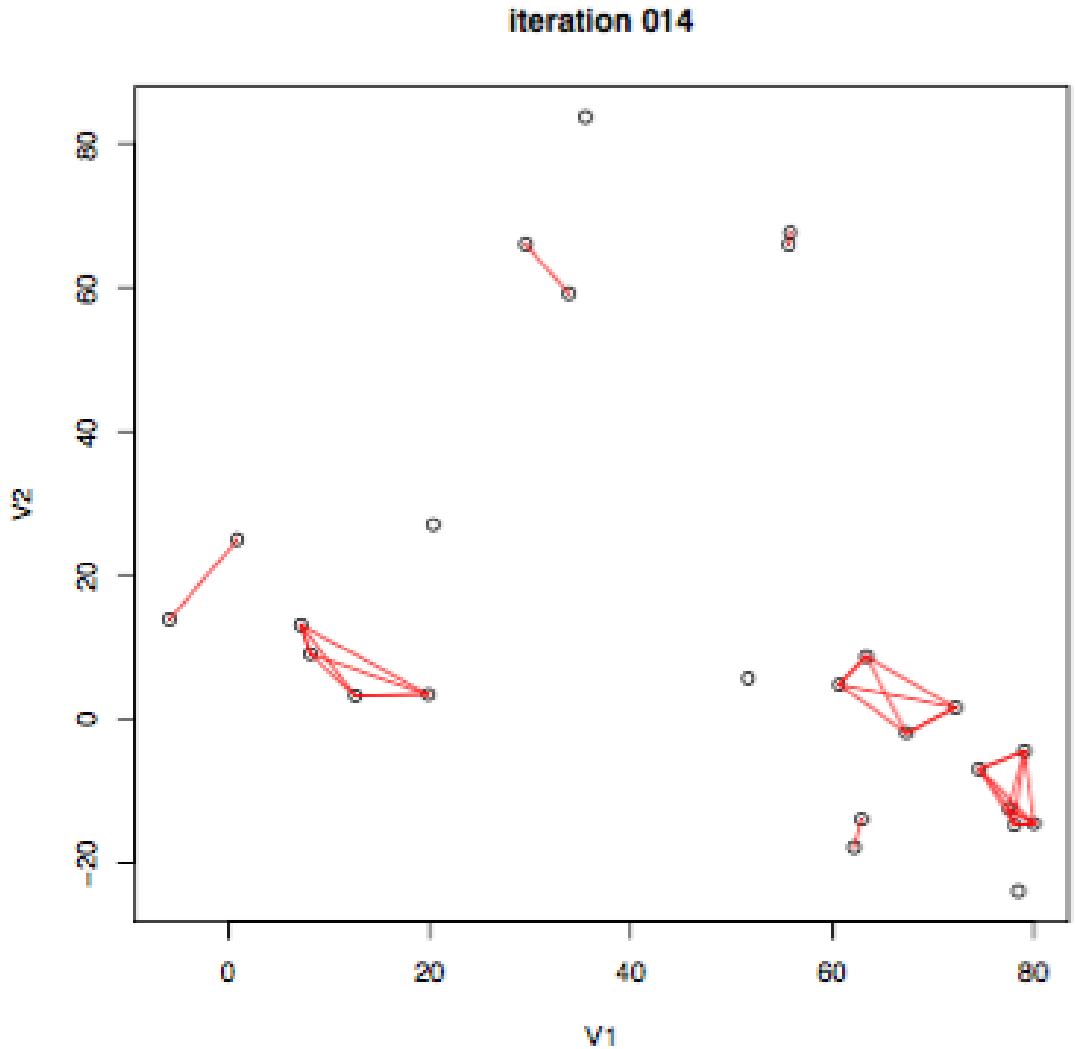
Iteration 12



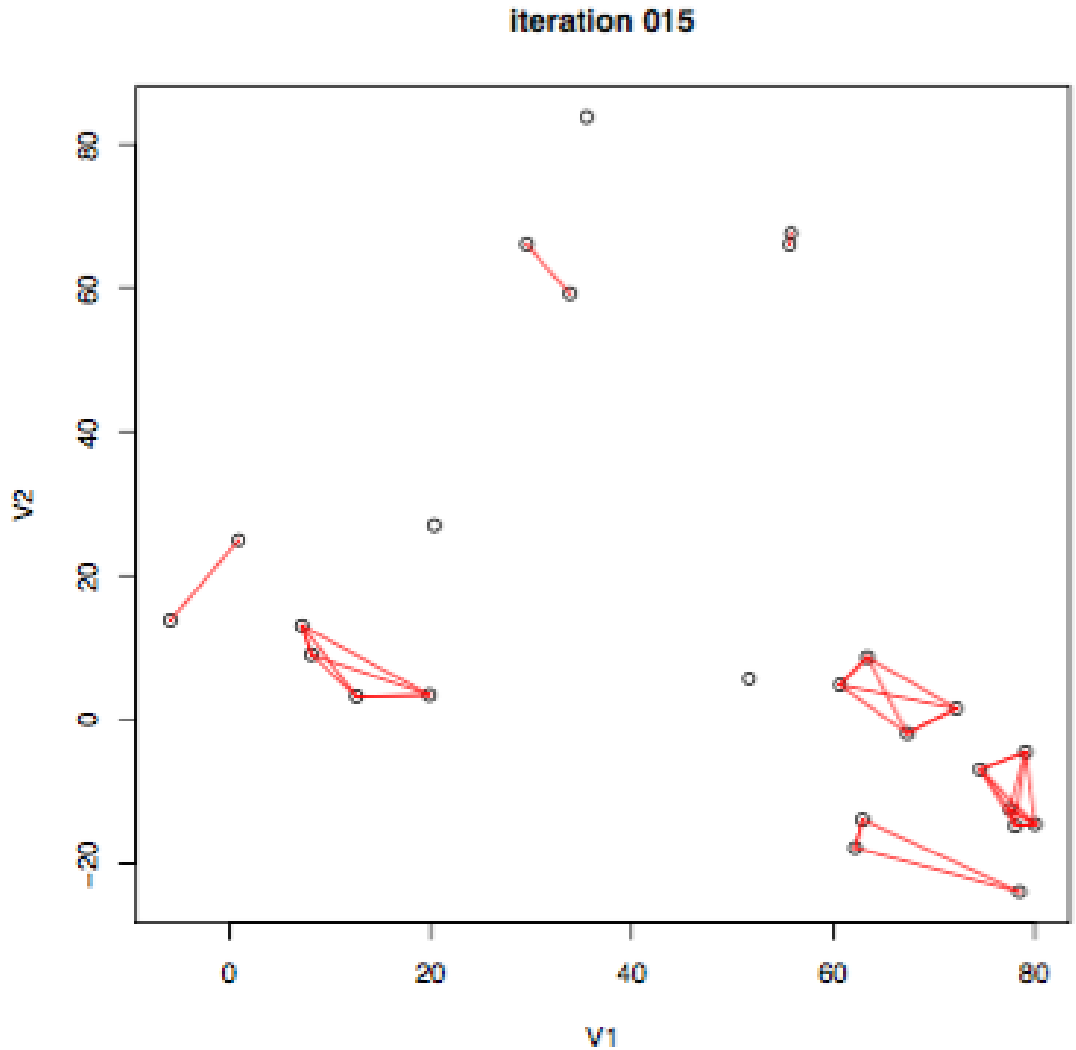
Iteration 13



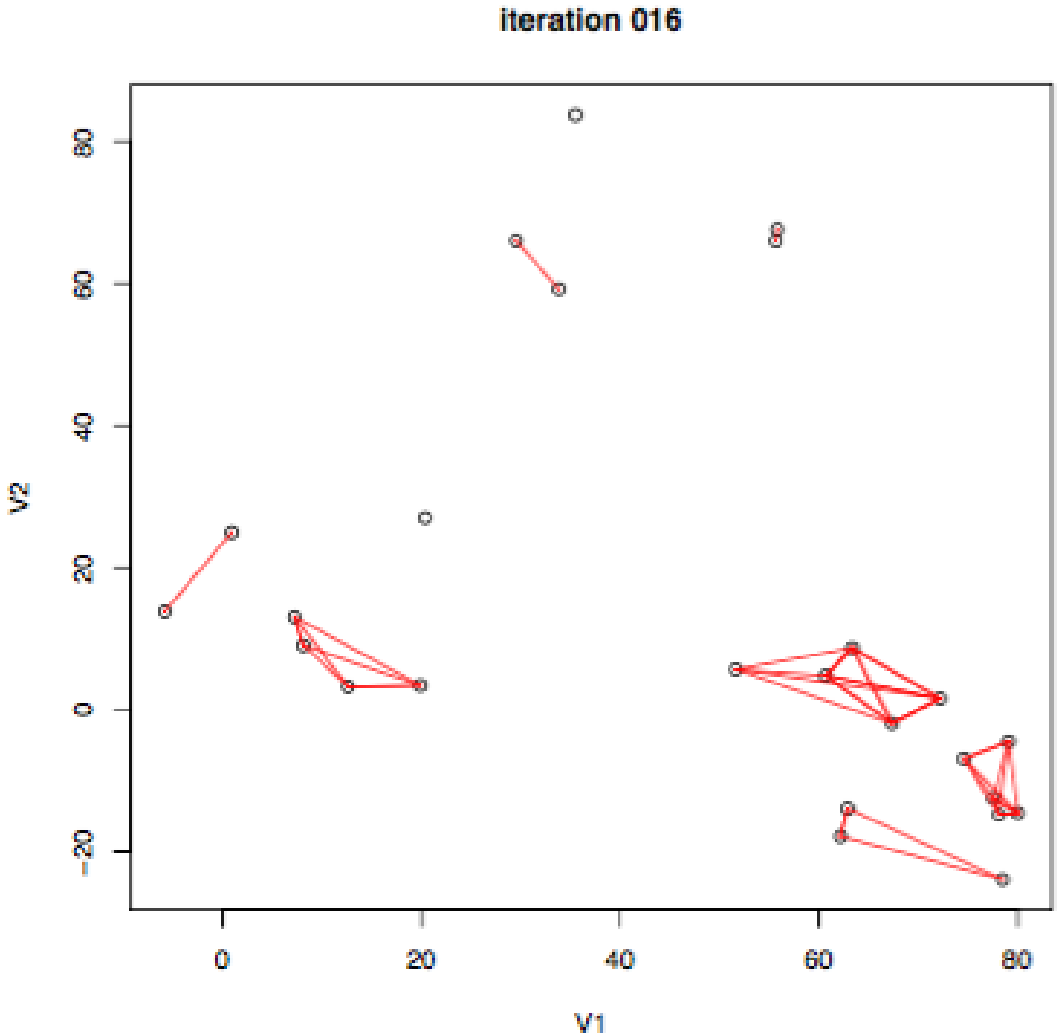
Iteration 14



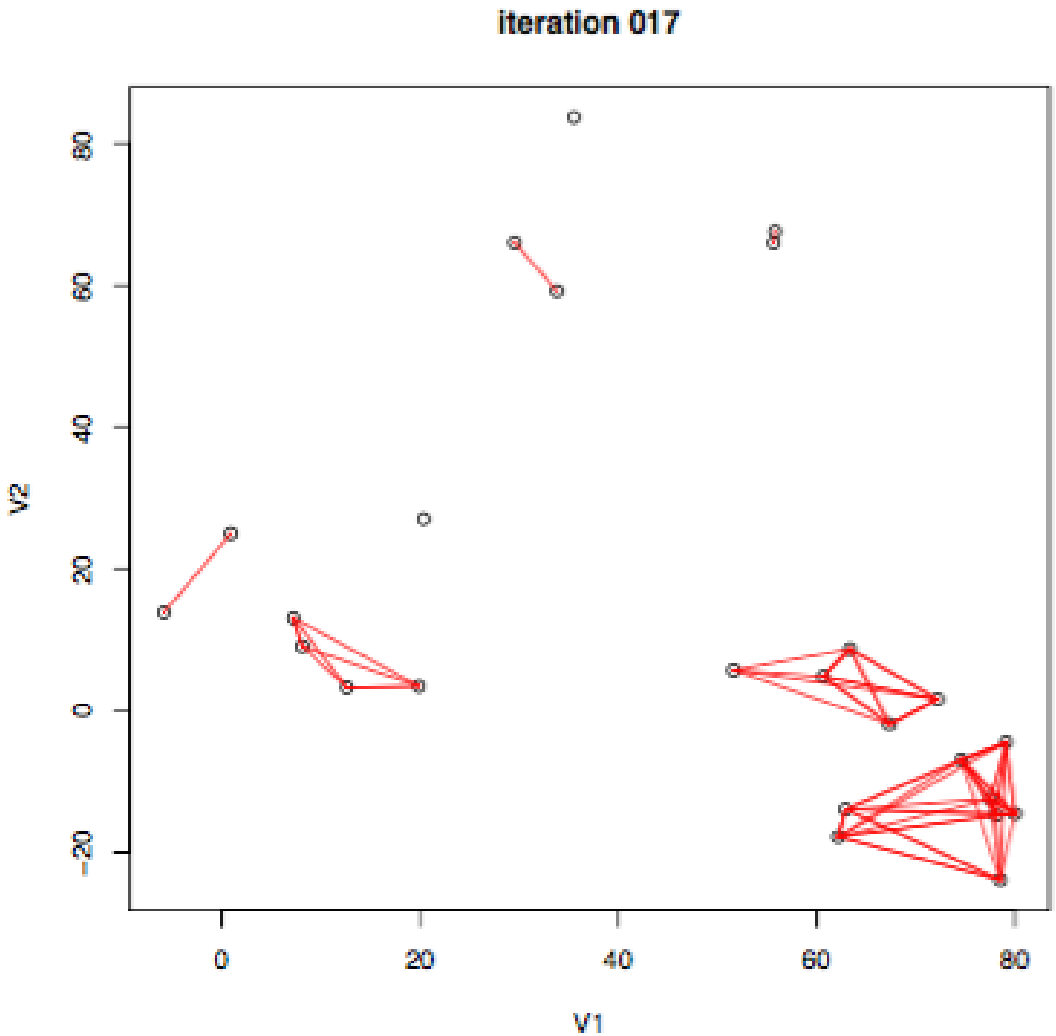
Iteration 15



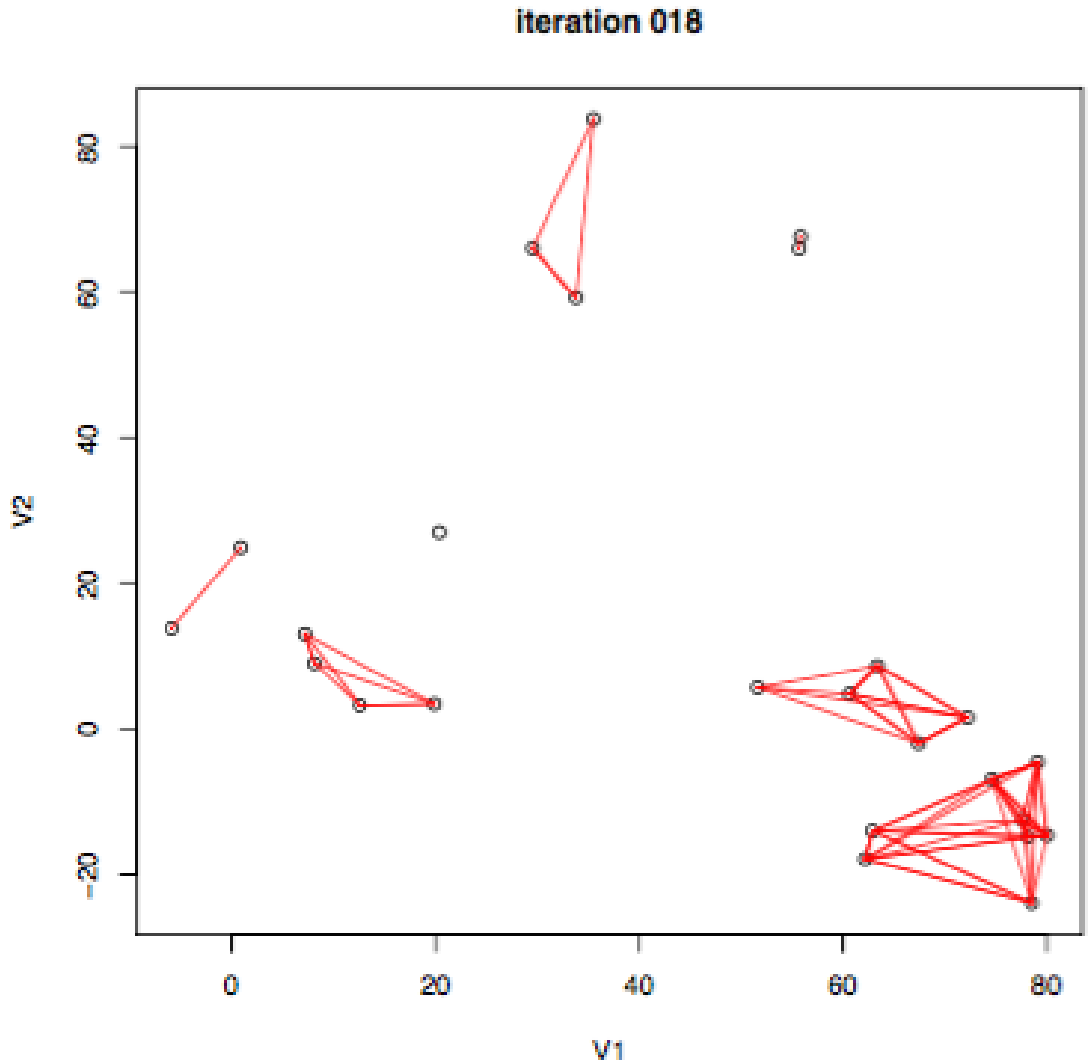
Iteration 16



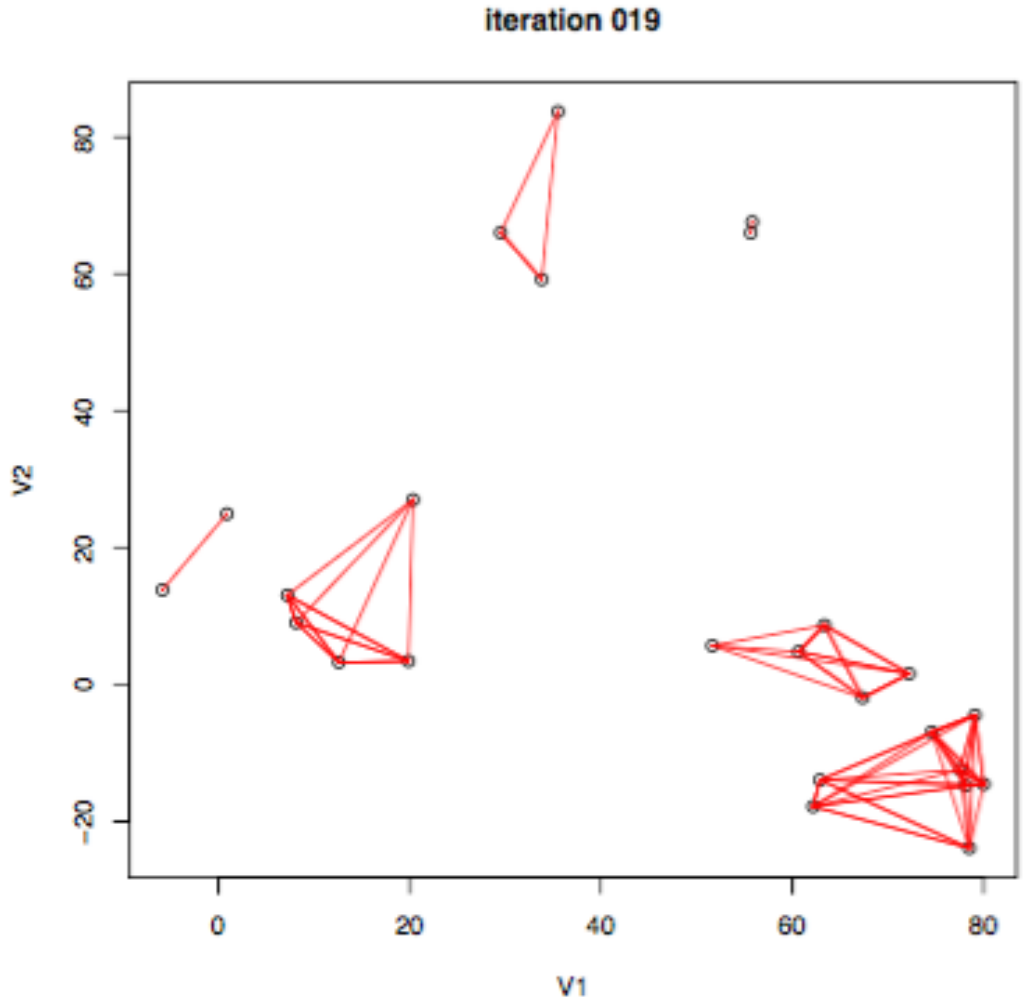
Iteration 17



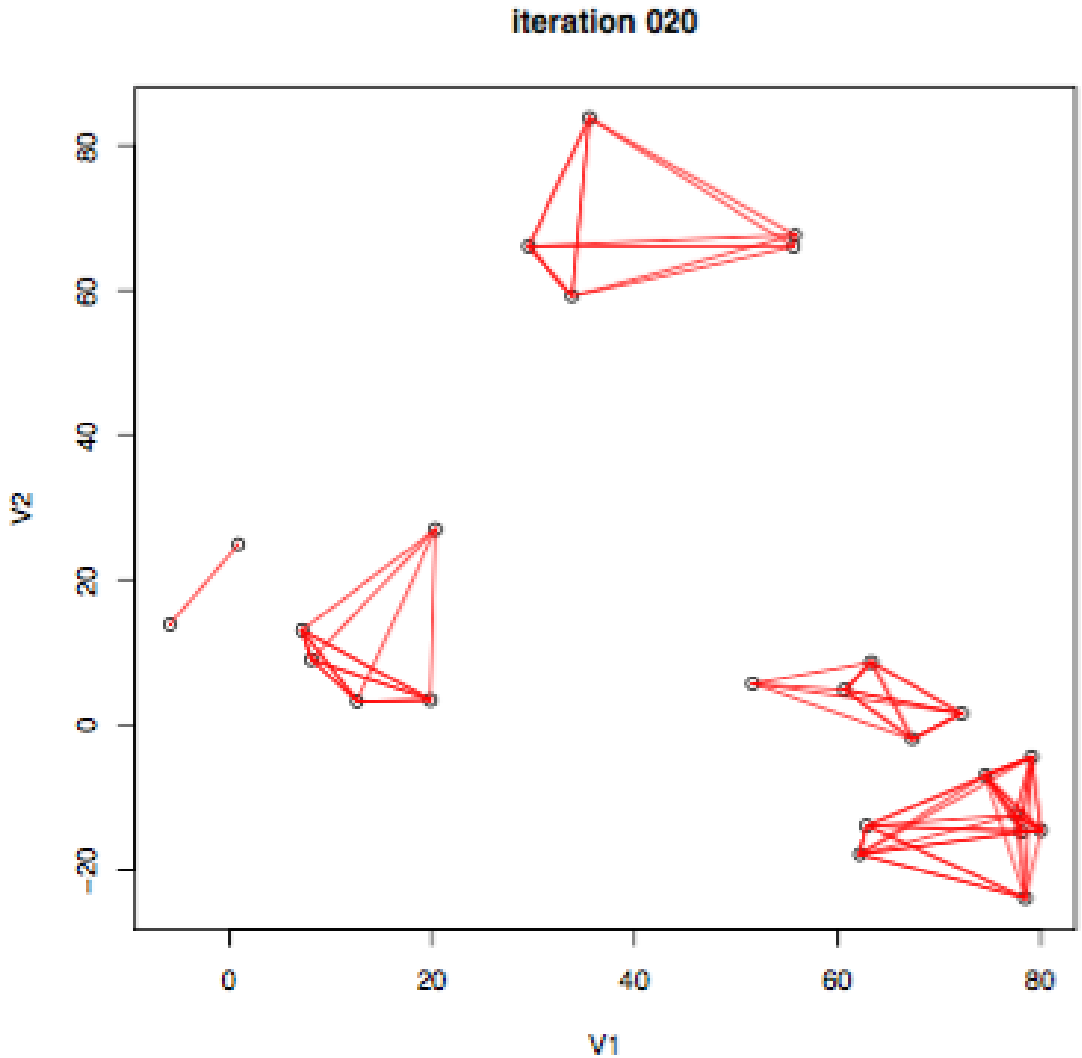
Iteration 18



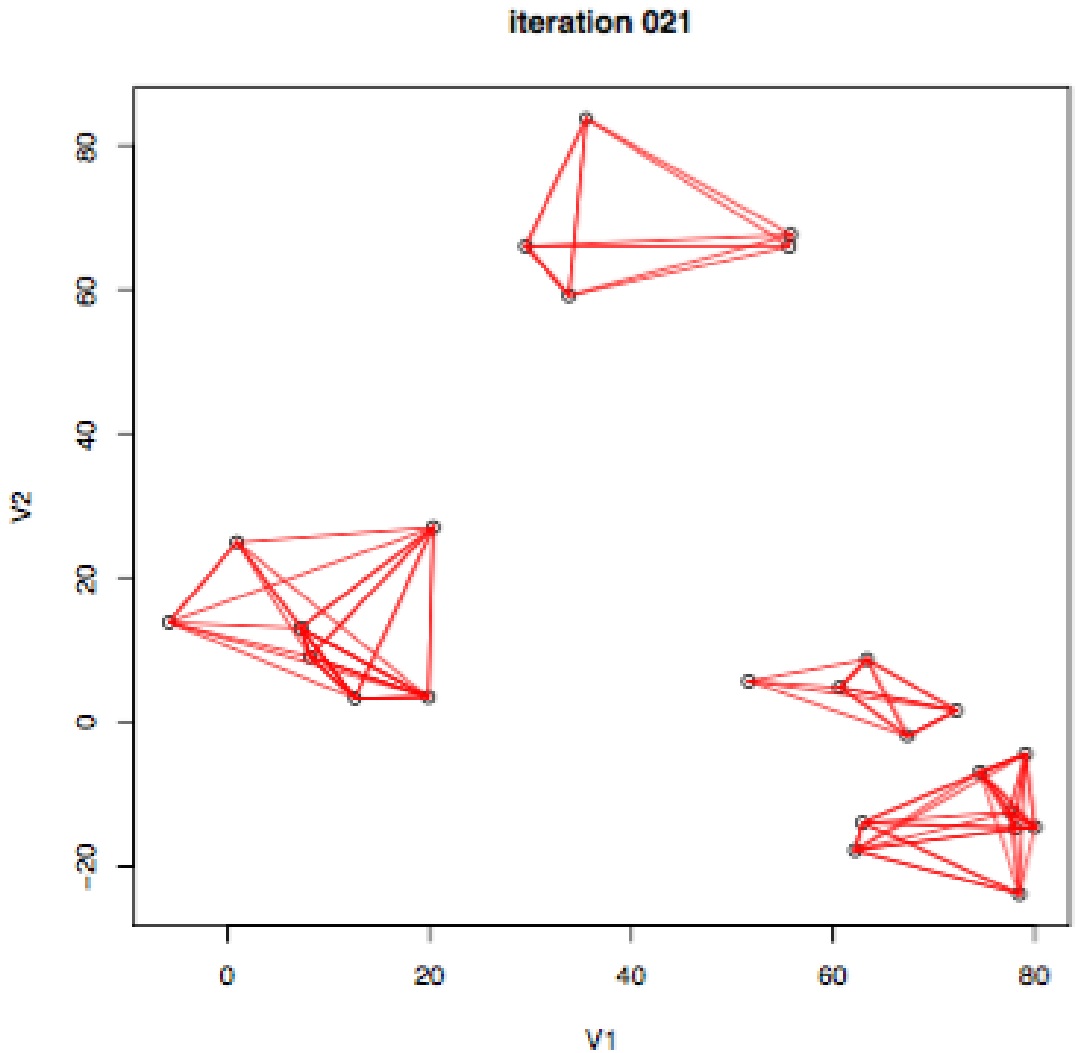
Iteration 19



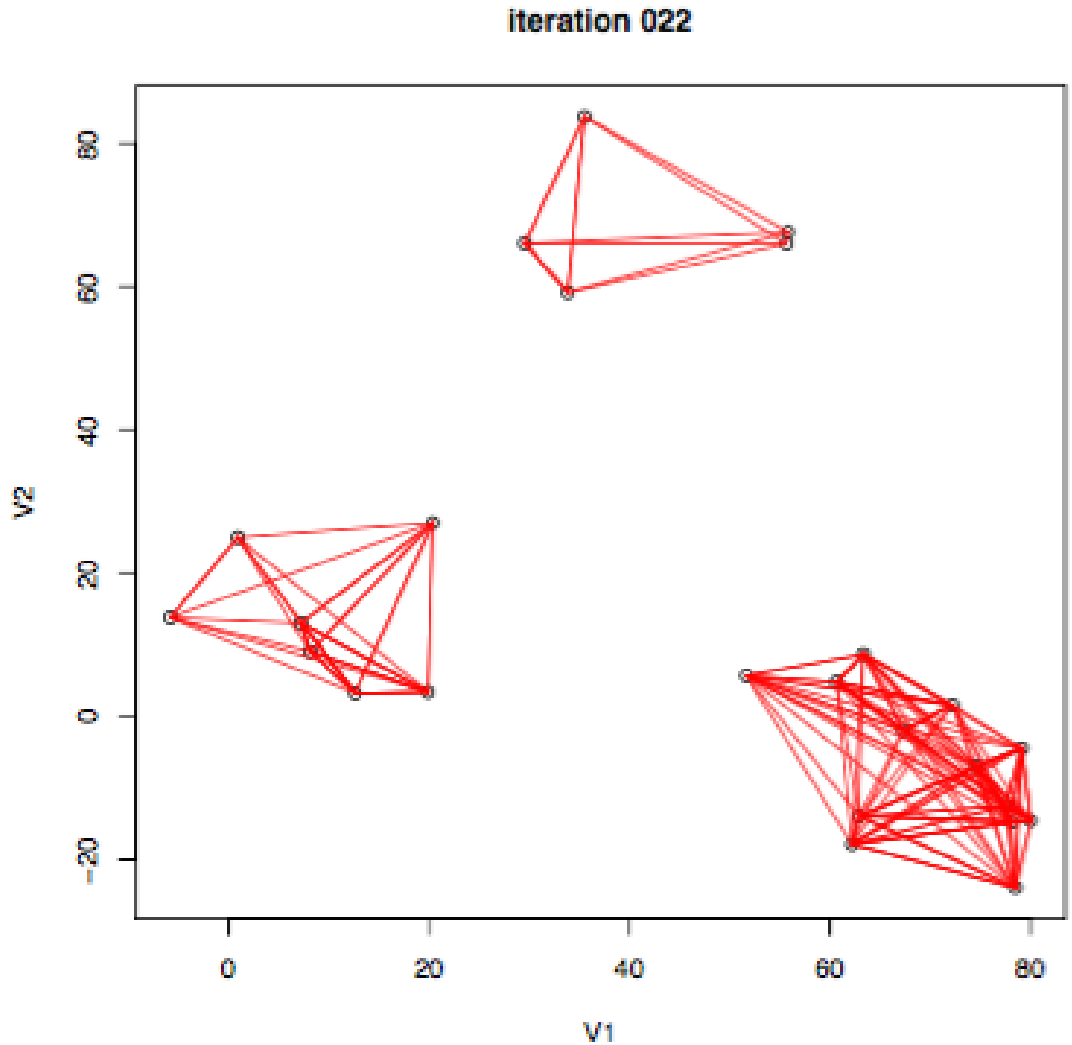
Iteration 20



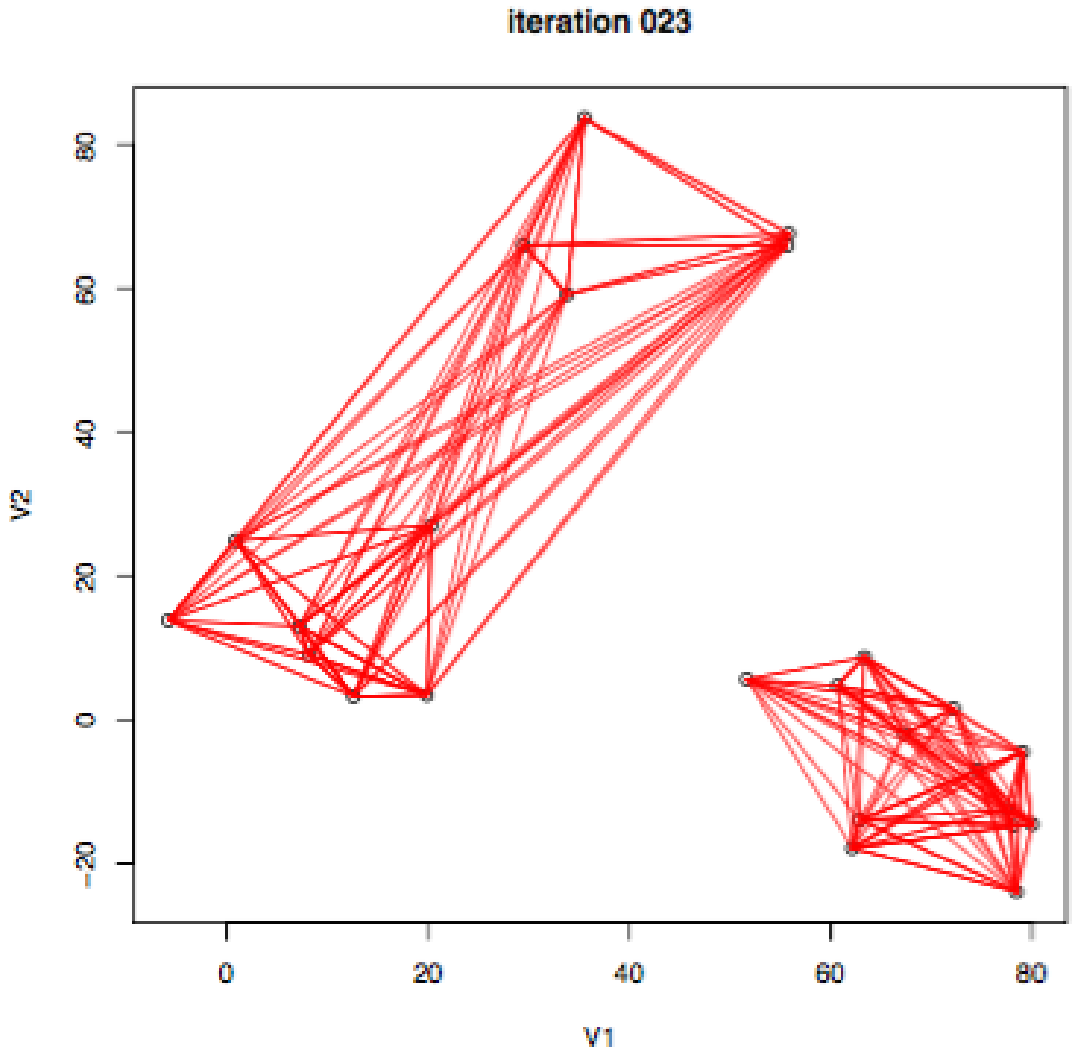
Iteration 21



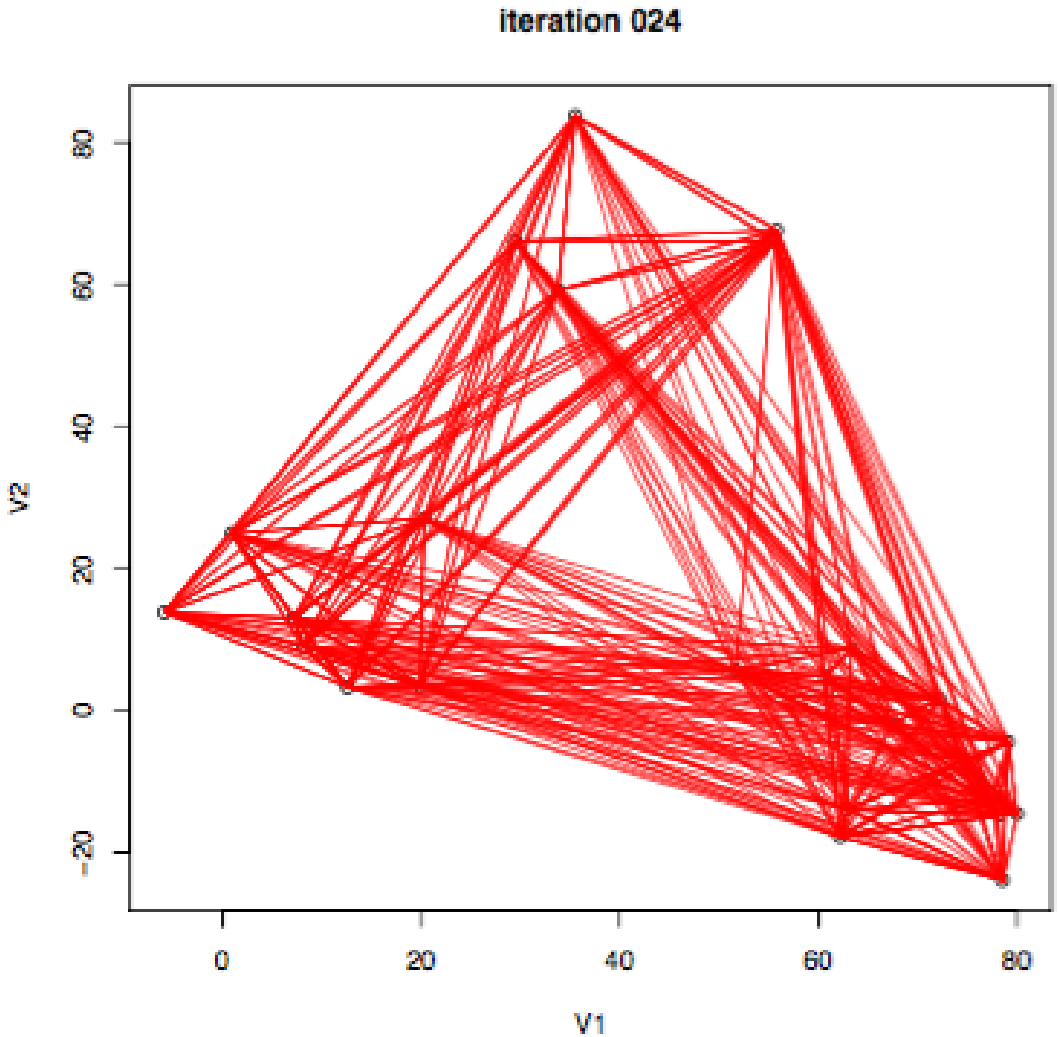
Iteration 22



Iteration 23



Iteration 24



Clustering

Suppose you are using the above algorithm to cluster the data points in groups.

- How do you know when to stop?
- How should we compare the data points?

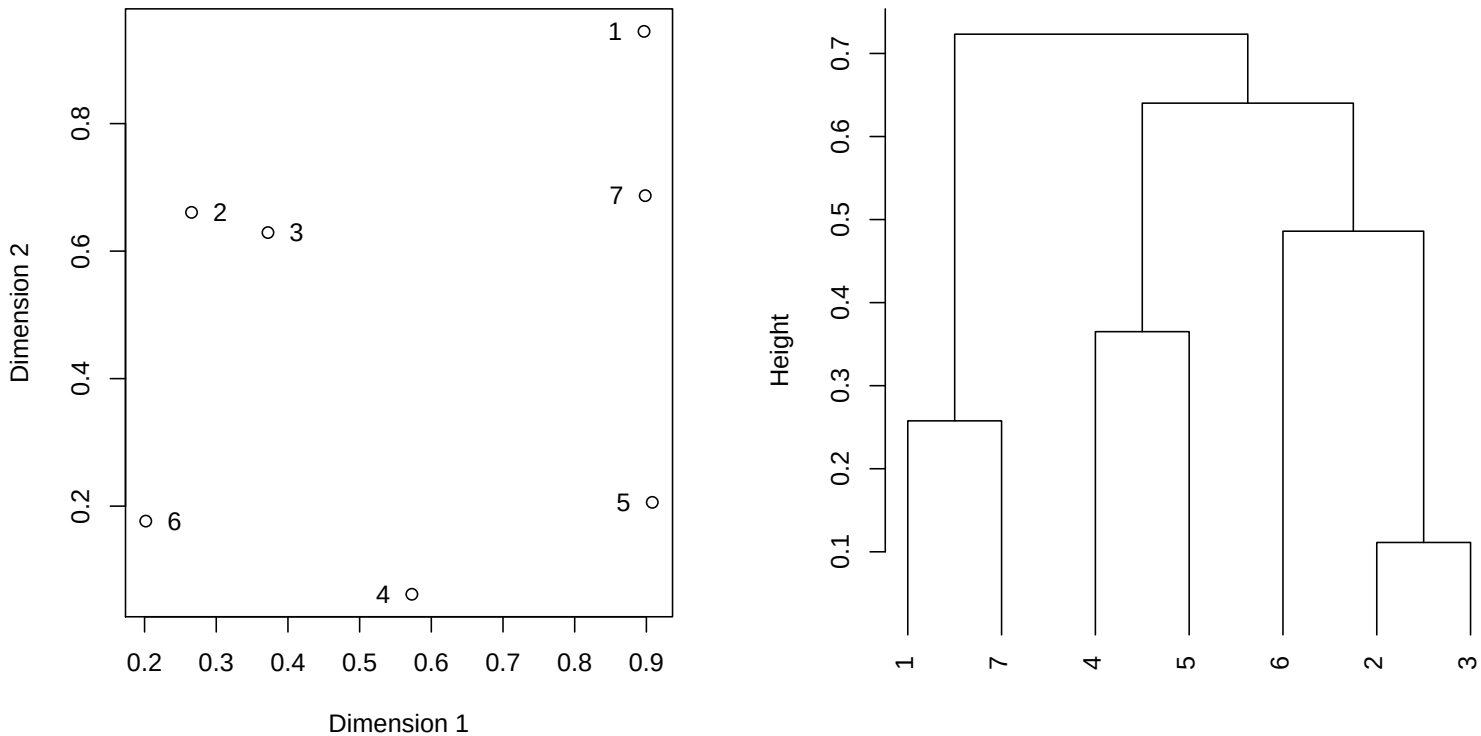
Let's investigate this further!

Agglomerative clustering

- Each level of the resulting tree is a segmentation of the data
- The algorithm results in a sequence of groupings
- It is up to the user to choose a "natural" clustering from this sequence

Dendogram

We can also represent the sequence of clustering assignments as a dendrogram:



Note that cutting the dendrogram horizontally partitions the data points into clusters

Dendogram

- Agglomerative clustering is monotonic
- The similarity between merged clusters is monotone decreasing with the level of the merge.
- Dendrogram: Plot each merge at the (negative) similarity between the two merged groups
- Provides an interpretable visualization of the algorithm and data
- Useful summarization tool, part of why hierarchical clustering is popular

Group similarity

Given a distance similarity measure (say, Eucliclean) between points, the user has many choices on how to define intergroup similarity.

1. Single linkage: the similiarity of the closest pair

$$d_{SL}(G, H) = \min_{i \in G, j \in H} d_{i,j}$$

2. Complete linkage: the similarity of the furthest pair

$$d_{CL}(G, H) = \max_{i \in G, j \in H} d_{i,j}$$

3. Group-average: the average similarity between groups

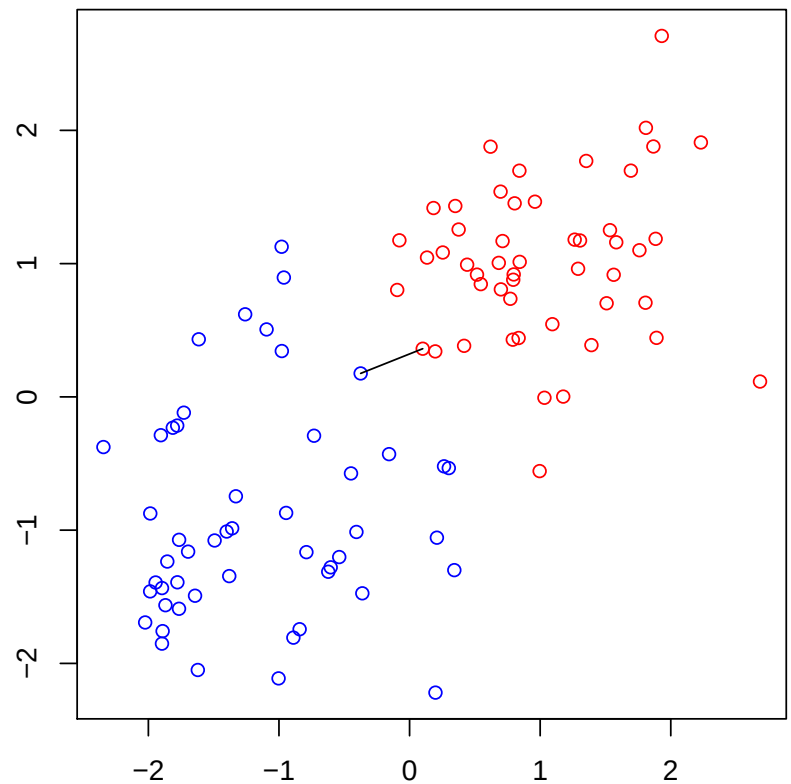
$$d_{GA} = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{i,j}$$

Single Linkage

In single linkage (i.e., nearest-neighbor linkage), the dissimilarity between G, H is the smallest dissimilarity between two points in opposite groups:

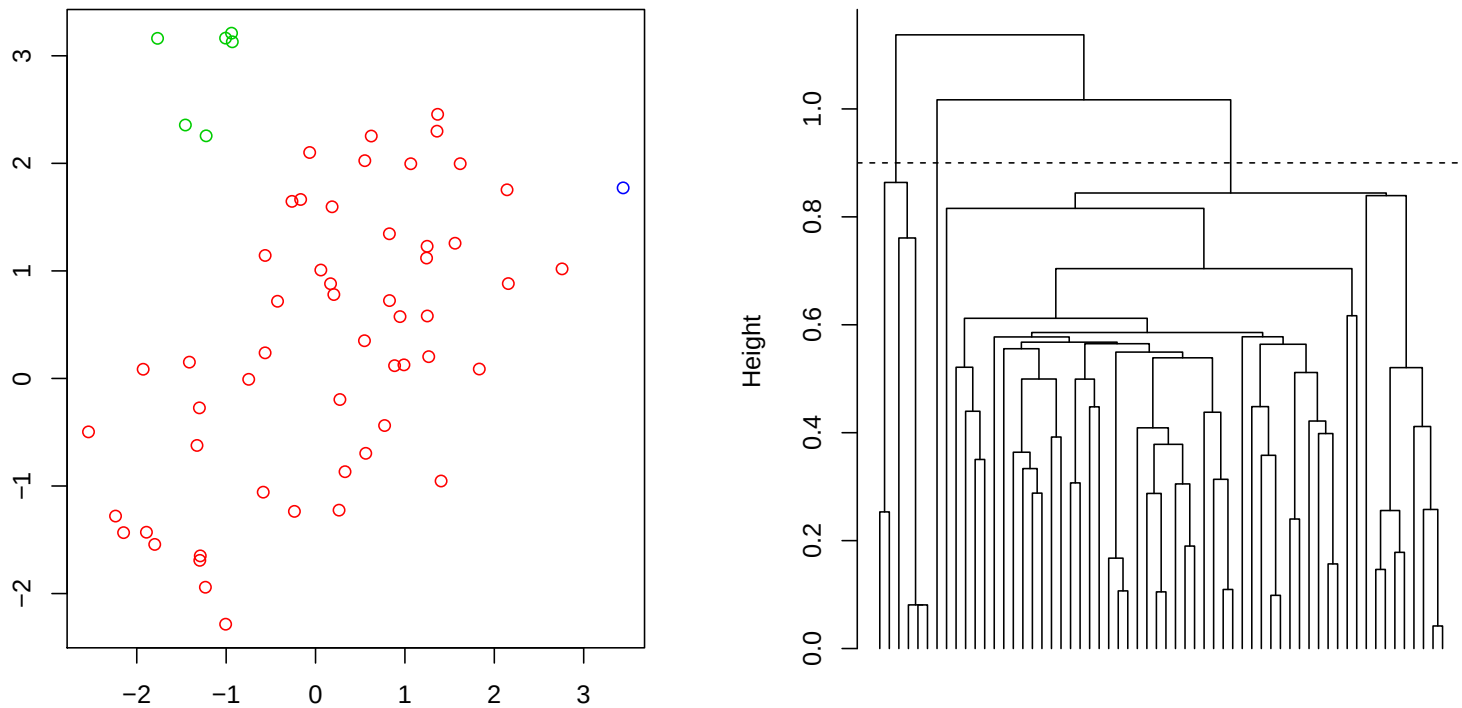
$$d_{\text{single}}(G, H) = \min_{i \in G, j \in H} d_{ij}$$

Example (dissimilarities d_{ij} are distances, groups are marked by colors): single linkage score $d_{\text{single}}(G, H)$ is the distance of the closest pair



Single Linkage Example

Here $n = 60$, $X_i \in \mathbb{R}^2$, $d_{ij} = \|X_i - X_j\|_2$. Cutting the tree at $h = 0.9$ gives the clustering assignments marked by colors



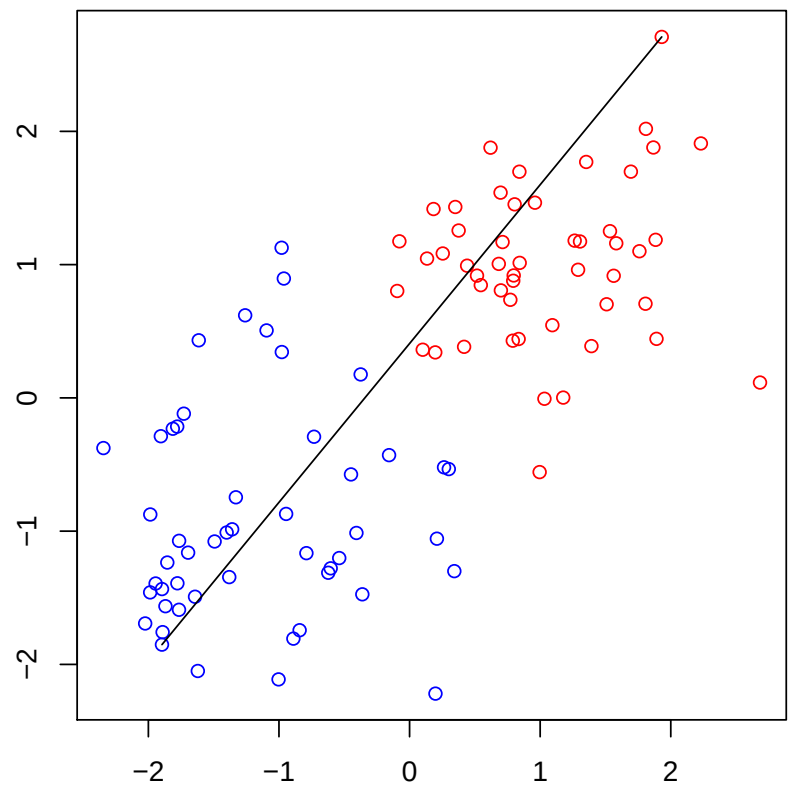
Cut interpretation: for each point X_i, there is another point X_j in its cluster with $d_{ij} \leq 0.9$

Complete Linkage

In complete linkage (i.e., furthest-neighbor linkage), dissimilarity between G, H is the largest dissimilarity between two points in opposite groups:

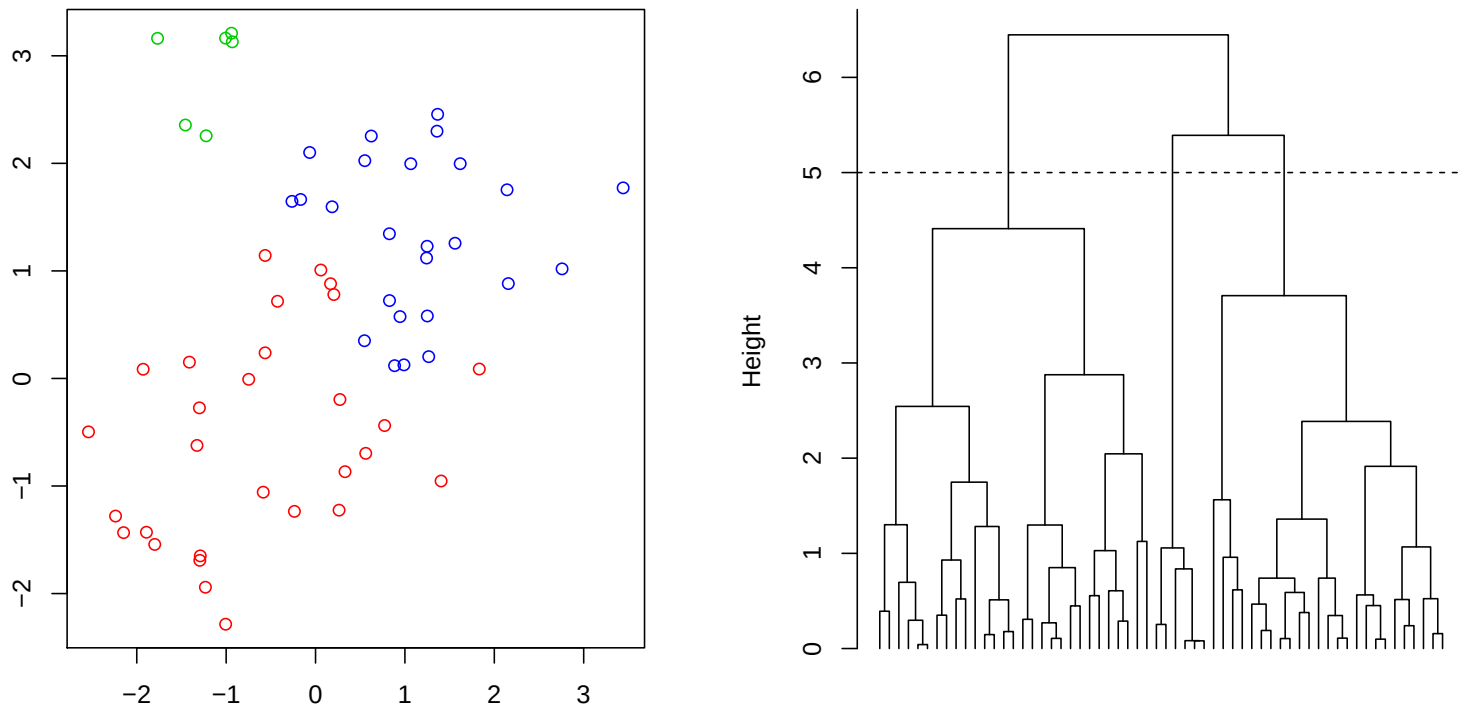
$$d_{\text{complete}}(G, H) = \max_{i \in G, j \in H} d_{ij}$$

Example (dissimilarities d_{ij} are distances, groups are marked by colors): complete linkage score $d_{\text{complete}}(G, H)$ is the distance of the furthest pair



Complete Linkage Example

Same data as before. Cutting the tree at $h = 5$ gives the clustering assignments marked by colors



Cut interpretation: for each point X_i, every other point X_j in its cluster satisfies $d_{ij} \leq 5$

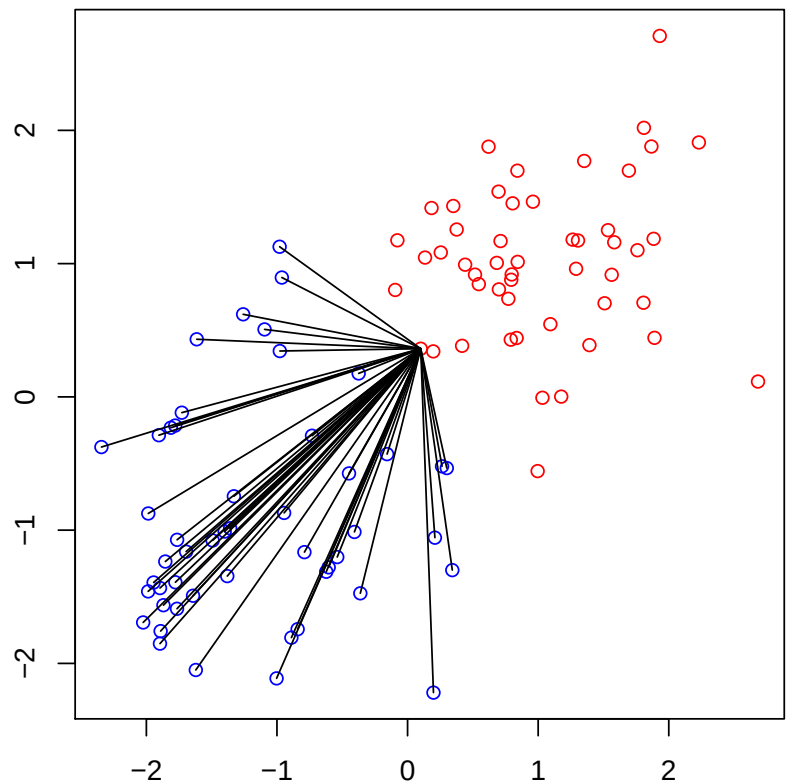
Average Linkage

In average linkage, the dissimilarity between G, H is the average dissimilarity over all points in opposite groups:

$$d_{\text{average}}(G, H) = \frac{1}{n_G \cdot n_H} \sum_{i \in G, j \in H} d_{ij}$$

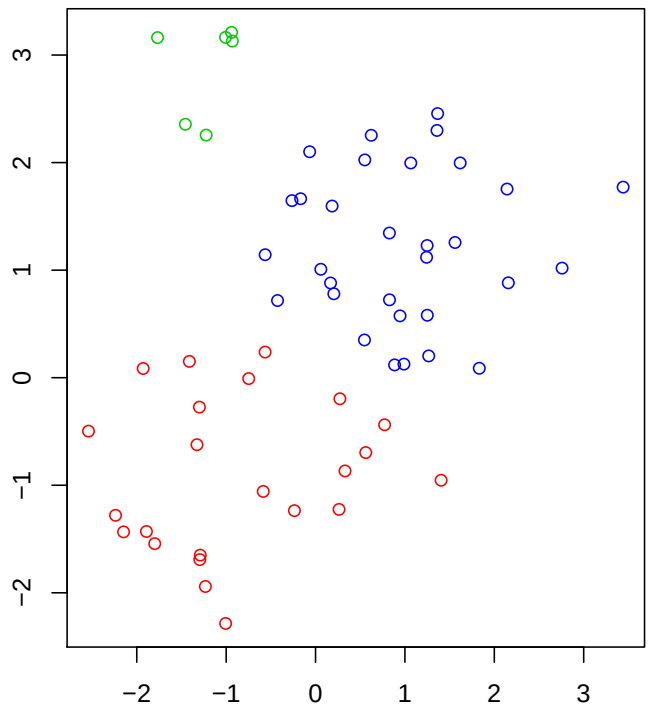
Example (dissimilarities d_{ij} are distances, groups are marked by colors): average linkage score $d_{\text{average}}(G, H)$ is the average distance across all pairs

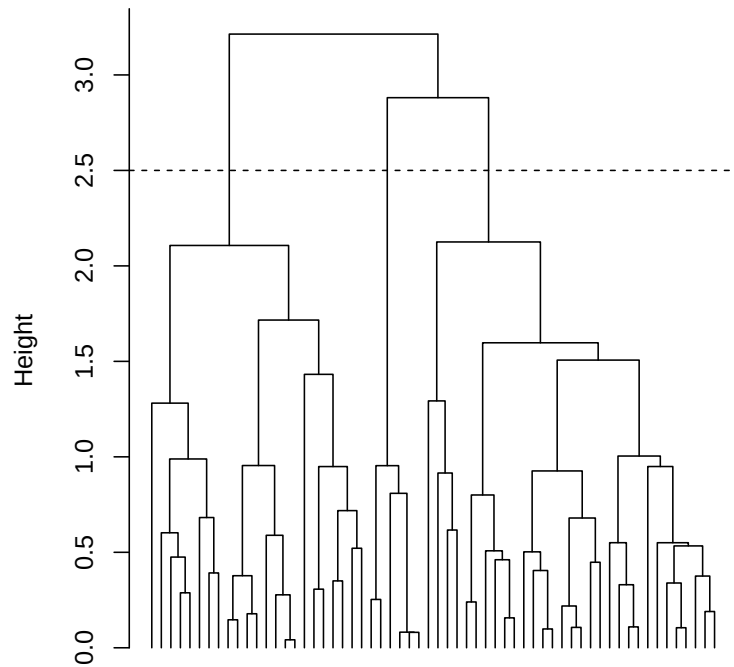
(Plot here only shows distances between the blue points and one red point)



Average linkage example

Same data as before. Cutting the tree at $h = 2.5$ gives clustering assignments marked by the colors





Cut interpretation: there really isn't a good one!

Properties of intergroup similarity

- Single linkage can produce "chaining," where a sequence of close observations in different groups cause early merges of those groups
- Complete linkage has the opposite problem. It might not merge close groups because of outlier members that are far apart.
- Group average represents a natural compromise, but depends on the scale of the similarities. Applying a monotone transformation to the similarities can change the results.

Things to consider

- Hierarchical clustering should be treated with caution.
- Different decisions about group similarities can lead to vastly different dendrograms.
- The algorithm imposes a hierarchical structure on the data, even data for which such structure is not appropriate.

Application on genomic data

- Unsupervised methods are often used in the analysis of genomic data.
- PCA and hierarchical clustering are very common tools. We will explore both on a genomic data set.
- We illustrate these methods on the NCI60 cancer cell line microarray data, which consists of 6,830 gene expression measurements on 64 cancer cell lines.

Application on genomic data

```
library(ISLR)
nci.labs <- NCI60$labs
nci.data <- NCI60$data
```

- Each cell line is labeled with a cancer type.
- We do not make use of the cancer types in performing PCA and clustering, as these are unsupervised techniques.
- After performing PCA and clustering, we will check to see the extent to which these cancer types agree with the results of these unsupervised techniques.

Exploring the data

```
dim(nci.data)
```

```
## [1]   64 6830
```

```
# cancer types for the cell lines
nci.labs[1:4]
```

```
## [1] "CNS"   "CNS"   "CNS"   "RENAL"
```

```
table(nci.labs)
```

```
## nci.labs
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

PCA

```
pr.out <- prcomp(nci.data, scale=TRUE)
```

PCA

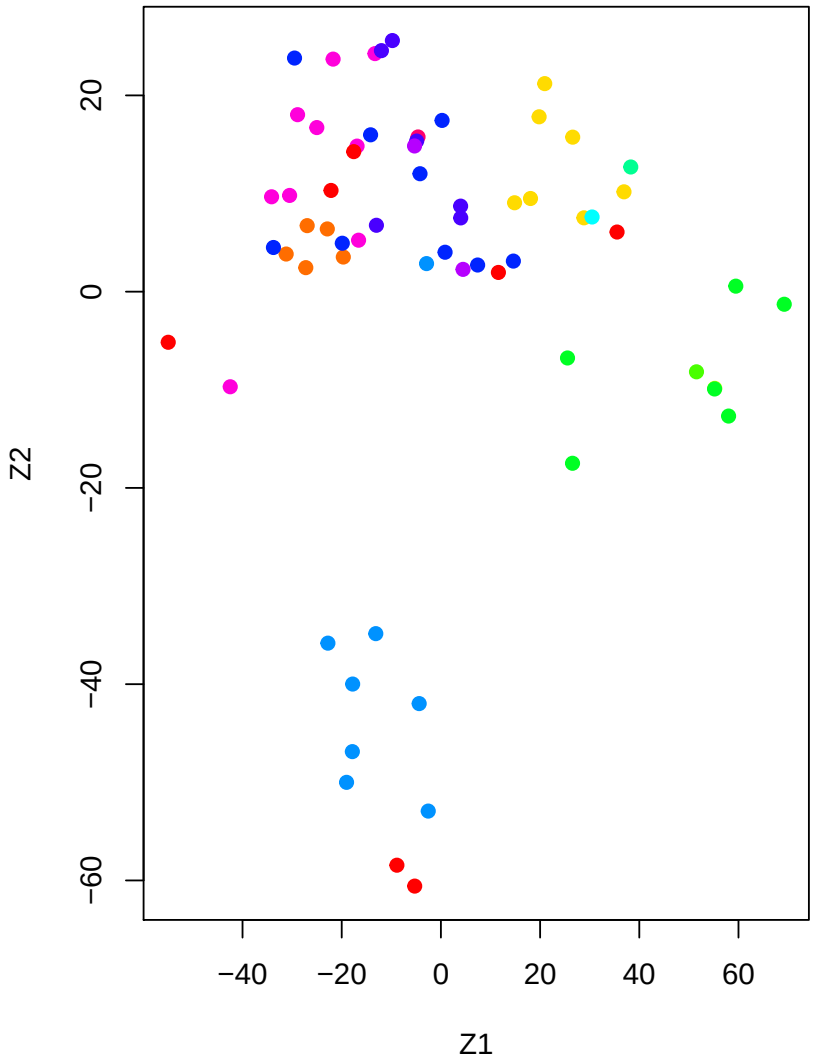
We now plot the first few principal component score vectors, in order to visualize the data.

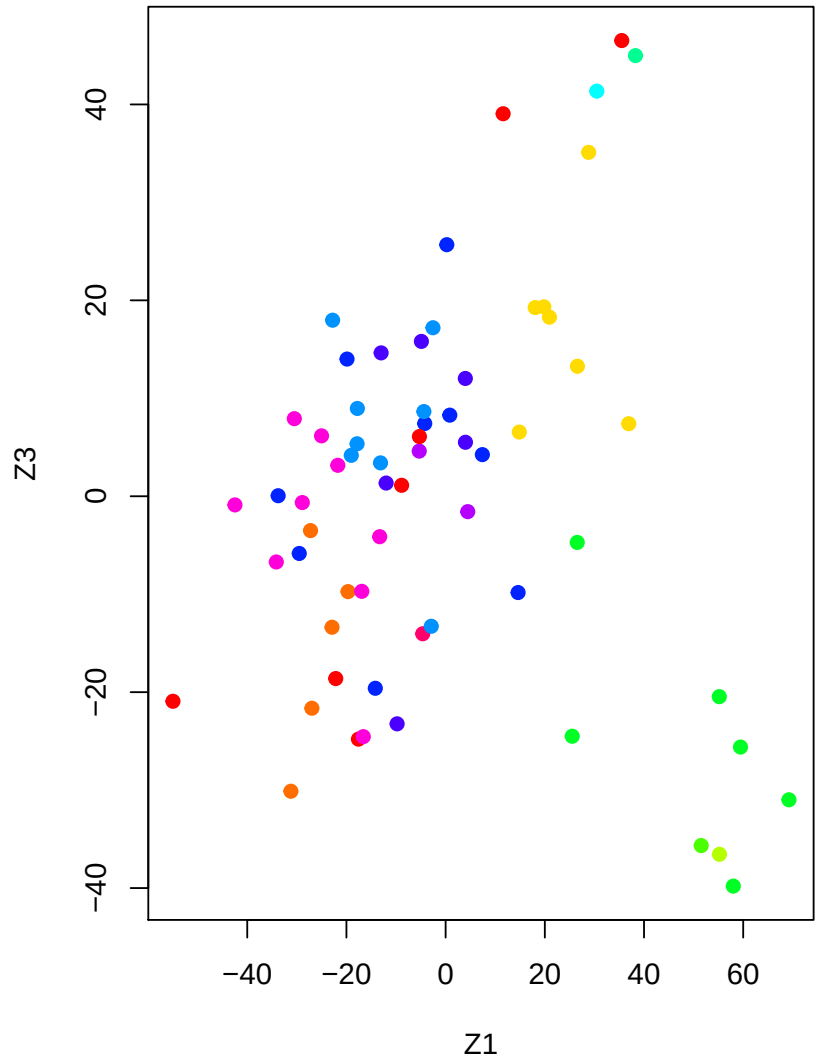
First, we create a simple function that assigns a distinct color to each element of a numeric vector. The function will be used to assign a color to each of the 64 cell lines, based on the cancer type to which it corresponds.

Simple color function

```
#Input: positive integer, vector
#Output: vector containing that
  #number of distinct colors
Cols=function(vec){
  cols<-rainbow(length(unique(vec)))
  return(cols[as.numeric(as.factor(vec))])
  }
```

PCA





On the whole, cell lines corresponding to a single cancer type do tend to have similar values on the first few PC score vectors. This indicates that cell lines from the same cancer type tend to have pretty similar gene expression levels.

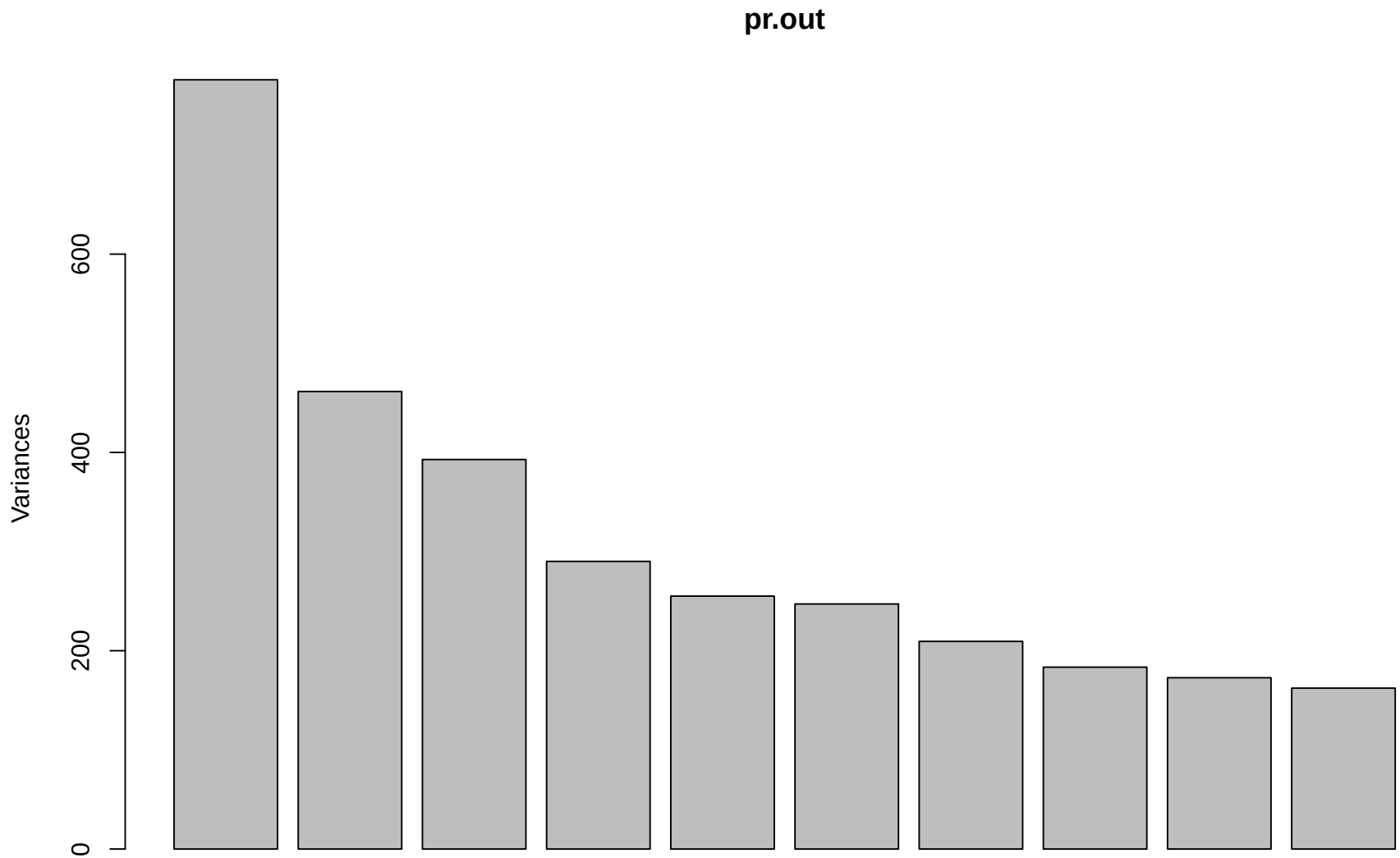
Proportion of Variance Explained

```
summary(pr.out)
```

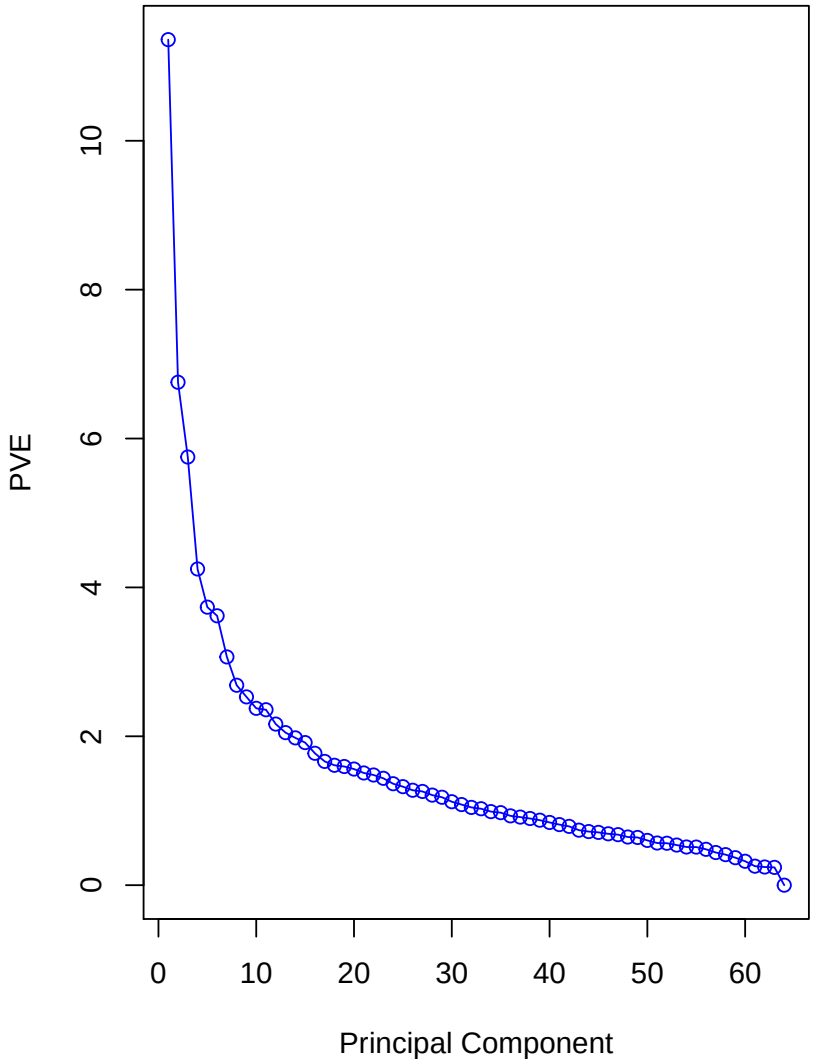
```
## Importance of components%s:
##                            PC1      PC2      PC3      PC4      PC5
## Standard deviation     27.8535 21.48136 19.82046 17.03256 15.97181
## Proportion of Variance  0.1136  0.06756  0.05752  0.04248  0.03735
## Cumulative Proportion   0.1136  0.18115  0.23867  0.28115  0.31850
##                            PC6      PC7      PC8      PC9     PC10
## Standard deviation     15.72108 14.47145 13.54427 13.14400 12.73860
## Proportion of Variance  0.03619  0.03066  0.02686  0.02529  0.02376
## Cumulative Proportion   0.35468  0.38534  0.41220  0.43750  0.46126
##                           PC11     PC12     PC13     PC14     PC15
## Standard deviation     12.68672 12.15769 11.83019 11.62554 11.43779
## Proportion of Variance  0.02357  0.02164  0.02049  0.01979  0.01915
## Cumulative Proportion   0.48482  0.50646  0.52695  0.54674  0.56590
##                           PC16     PC17     PC18     PC19    PC20
## Standard deviation     11.00051 10.65666 10.48880 10.43518 10.3219
## Proportion of Variance  0.01772  0.01663  0.01611  0.01594  0.0156
## Cumulative Proportion   0.58361  0.60024  0.61635  0.63229  0.6479
##                           PC21    PC22    PC23    PC24    PC25    PC26
## Standard deviation     10.14608 10.0544 9.90265 9.64766 9.50764 9.33253
## Proportion of Variance  0.01507  0.0148 0.01436 0.01363 0.01324 0.01275
## Cumulative Proportion   0.66296  0.6778 0.69212 0.70575 0.71899 0.73174
##                           PC27   PC28    PC29    PC30    PC31    PC32
## Standard deviation     9.27320 9.0900 8.98117 8.75003 8.59962 8.44738
## Proportion of Variance 0.01259 0.0121 0.01181 0.01121 0.01083 0.01045
## Cumulative Proportion  0.74433 0.7564 0.76824 0.77945 0.79027 0.80072
##                           PC33    PC34    PC35    PC36    PC37    PC38
## Standard deviation     8.37305 8.21579 8.15731 7.97465 7.90446 7.82127
## Proportion of Variance 0.01026 0.00988 0.00974 0.00931 0.00915 0.00896
## Cumulative Proportion  0.81099 0.82087 0.83061 0.83992 0.84907 0.85803
##                           PC39    PC40    PC41   PC42    PC43   PC44
## Standard deviation     7.72156 7.58603 7.45619 7.3444 7.10449 7.0131
## Proportion of Variance 0.00873 0.00843 0.00814 0.0079 0.00739 0.0072
```

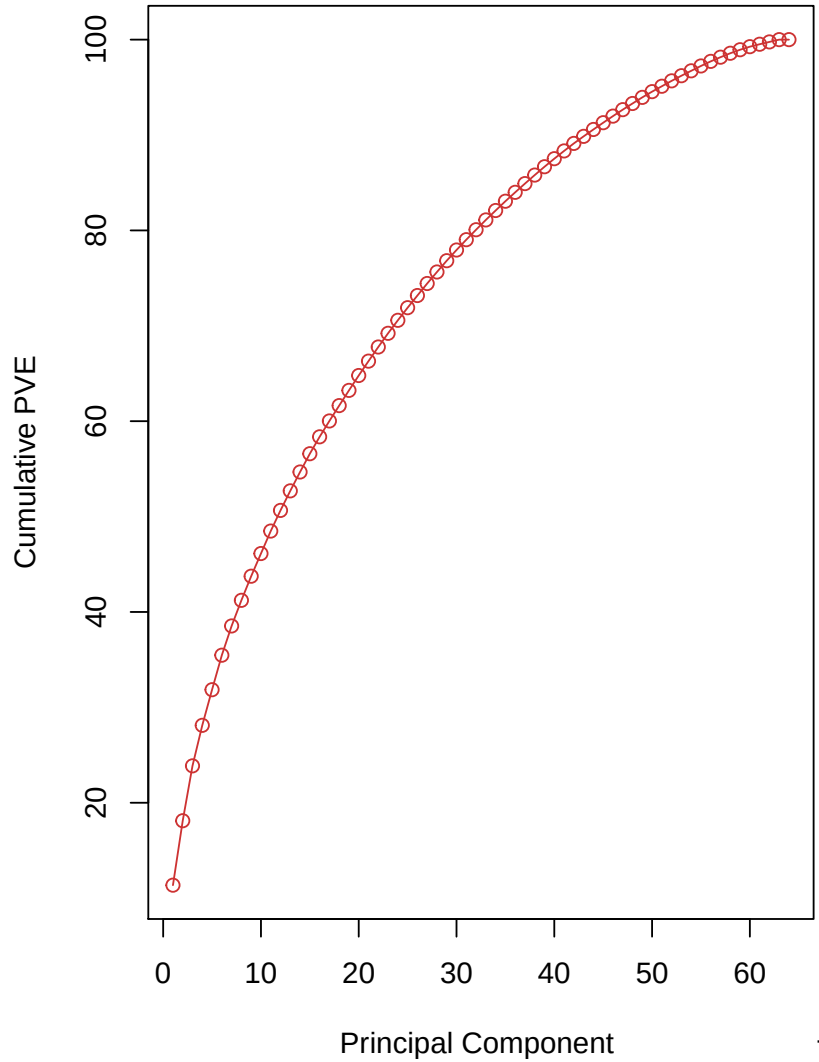
Proportion of Variance Explained

```
plot(pr.out)
```



PCA





The proportion of variance explained (PVE) of the principal components of the NCI60 cancer cell line microarray data set. Left: the PVE of each principal component is shown. Right: the cumulative PVE of the principal components is shown. Together, all principal components explain 100 % of the variance.

Conclusions from the Scree Plot

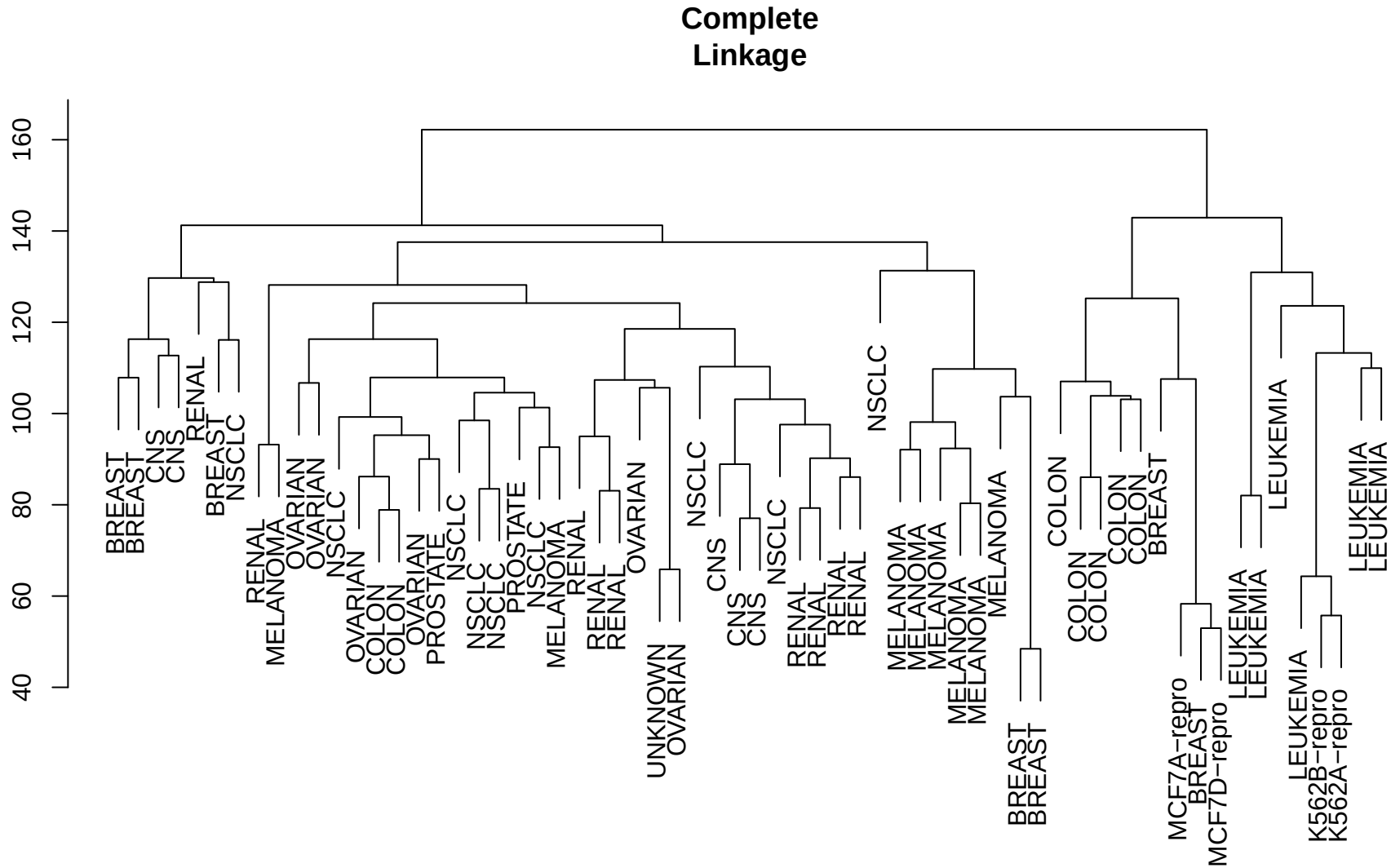
- We see that together, the first seven principal components explain around 40% of the variance in the data.
- This is not a huge amount of the variance.
- Looking at the scree plot, we see that while each of the first seven principal components explain a substantial amount of variance, there is a marked decrease in the variance explained by further principal components.
- That is, there is an elbow in the plot after approximately the seventh principal component. This suggests that there may be little benefit to examining more than seven or so principal components (though even examining seven principal components may be difficult).

Hierarchical Clustering to NCI60 Data

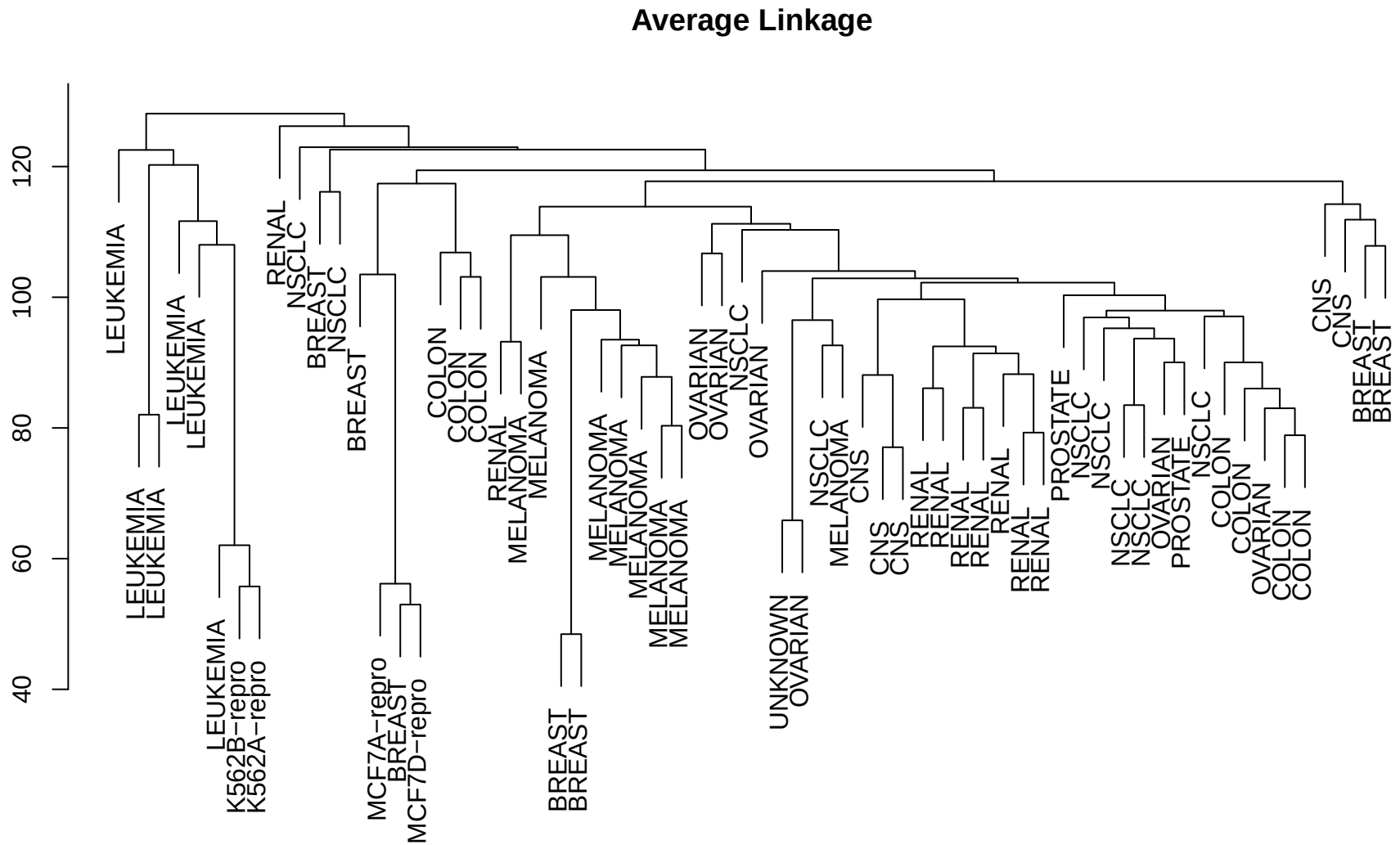
- We now proceed to hierarchically cluster the cell lines in the NCI60 data, with the goal of finding out whether or not the observations cluster into distinct types of cancer.
- To begin, we standardize the variables to have mean zero and standard deviation one.
- As mentioned earlier, this step is optional and should be performed only if we want each gene to be on the same scale.

```
sd.data=scale(nci.data)
```

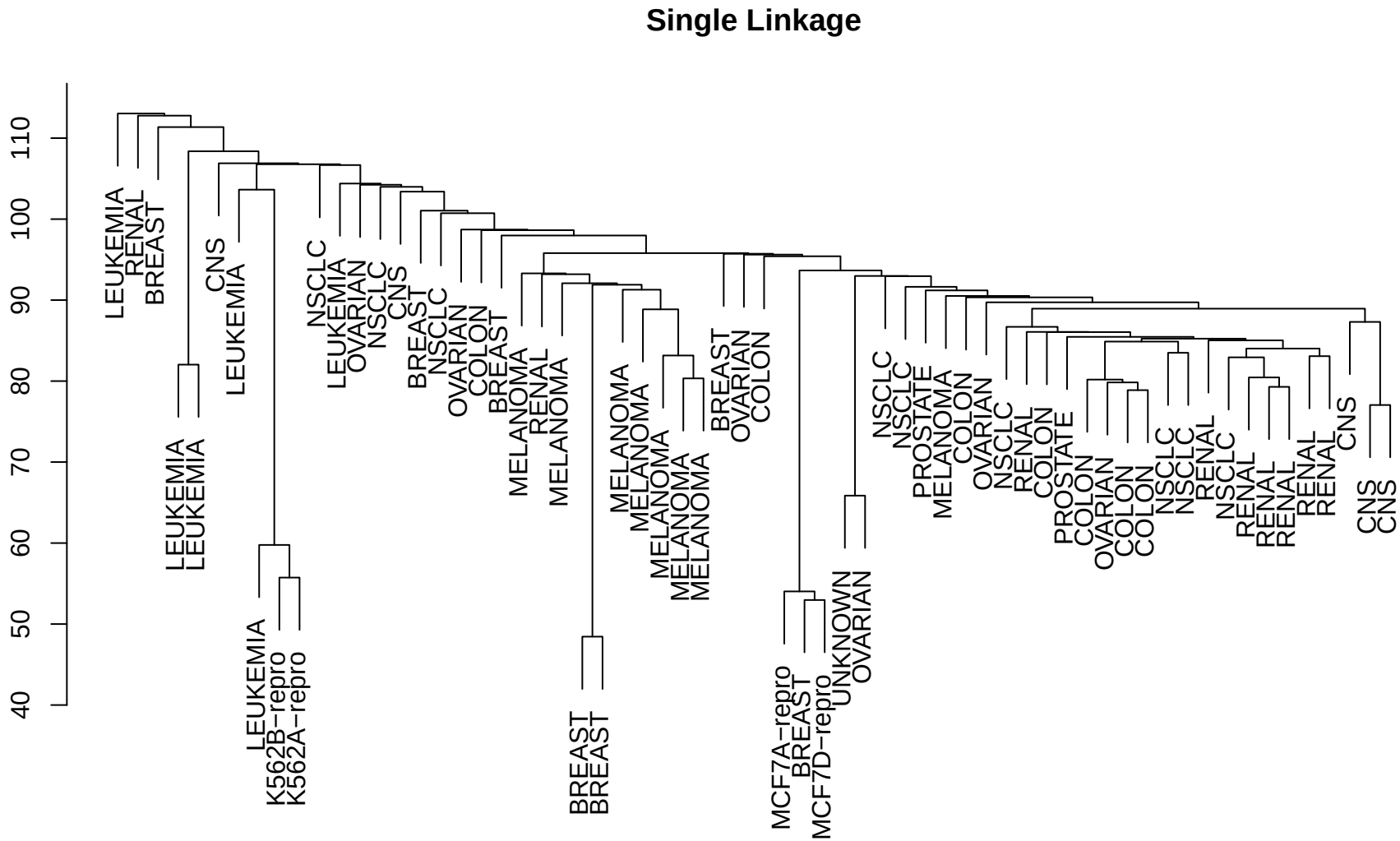
Hierarchical Clustering to NCI60 Data



Hierarchical Clustering to NCI60 Data



Hierarchical Clustering to NCI60 Data



We see that the choice of linkage certainly does affect the results obtained.

Hierarchical Clustering to NCI60 Data

- Typically, single linkage will tend to yield trailing clusters: very large clusters onto which individual observations attach one-by-one.
- On the other hand, complete and average linkage tend to yield more balanced, attractive clusters.
- For this reason, complete and average linkage are generally preferred to single linkage.

Complete linkage

- We will use complete linkage hierarchical clustering for the analysis that follows.
- We can cut the dendrogram at the height that will yield a particular number of clusters, say four.

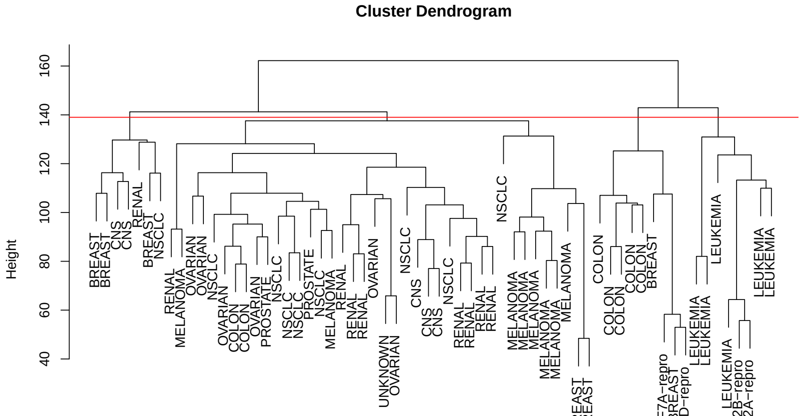
```
hc.out=hclust(dist(sd.data))
hc.clusters=cutree(hc.out,4)
table(hc.clusters,nci.labs)
```

```
##             nci.labs
## hc.clusters BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro
##           1      2   3     2           0           0        0           0
##           2      3   2     0           0           0        0           0
##           3      0   0     0           1           1        6           0
##           4      2   0     5           0           0        0           1
##             nci.labs
## hc.clusters MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
##           1           0        8     8       6        2     8       1
##           2           0        0     1       0        0     1       0
##           3           0        0     0       0        0     0       0
##           4           1        0     0       0        0     0       0
```

Complete linkage

All the leukemia cell lines fall in cluster 3, while the breast cancer cell lines are spread out over three different clusters. We can plot the cut on the dendrogram that produces these four clusters.

This is the height that results in four clusters. (It is easy to verify that the resulting clusters are the same as the ones we obtained using cutree(hc.out,4))



K-means versus complete linkage?

How do these NCI60 hierarchical clustering results compare to what we get if we perform K-means clustering with $K = 4$?

```
set.seed (2)
km.out <- kmeans(sd.data, 4, nstart=20)
km.clusters <- km.out$cluster
table(km.clusters, hc.clusters )
```

```
##             hc.clusters
## km.clusters  1  2  3  4
##            1 11  0  0  9
##            2  0  0  8  0
##            3  9  0  0  0
##            4 20  7  0  0
```

K-means versus complete linkage?

- We see that the four clusters obtained using hierarchical clustering and K-means clustering are somewhat different.
- Cluster 2 in K-means clustering is identical to cluster 3 in hierarchical clustering.
- However, the other clusters differ: for instance, cluster 4 in K-means clustering contains a portion of the observations assigned to cluster 1 by hierarchical clustering, as well as all of the observations assigned to cluster 2 by hierarchical clustering.