QUEEN'S UNIVERSITY BELFAST
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

# Data Analysis & Visualisation

**CSC3062**

**BEng (CS & SE), MEng (CS & SE), BIT & CIT**

Dr Reza Rafiee

Semester 1 2019

# Consensus clustering[1]

Assume, we are given a dataset for the purpose of clustering analysis

1) Multiple runs of a clustering algorithm
   a) Determine the number of clusters and assess the stability of the discovered clusters
   b) In k-means clustering: with using random restart

2) Aggregating the cluster (label) results of different clustering algorithms

[1] Ensemble clustering

# Consensus clustering[1]

Assume, we are given a dataset for the purpose of clustering analysis

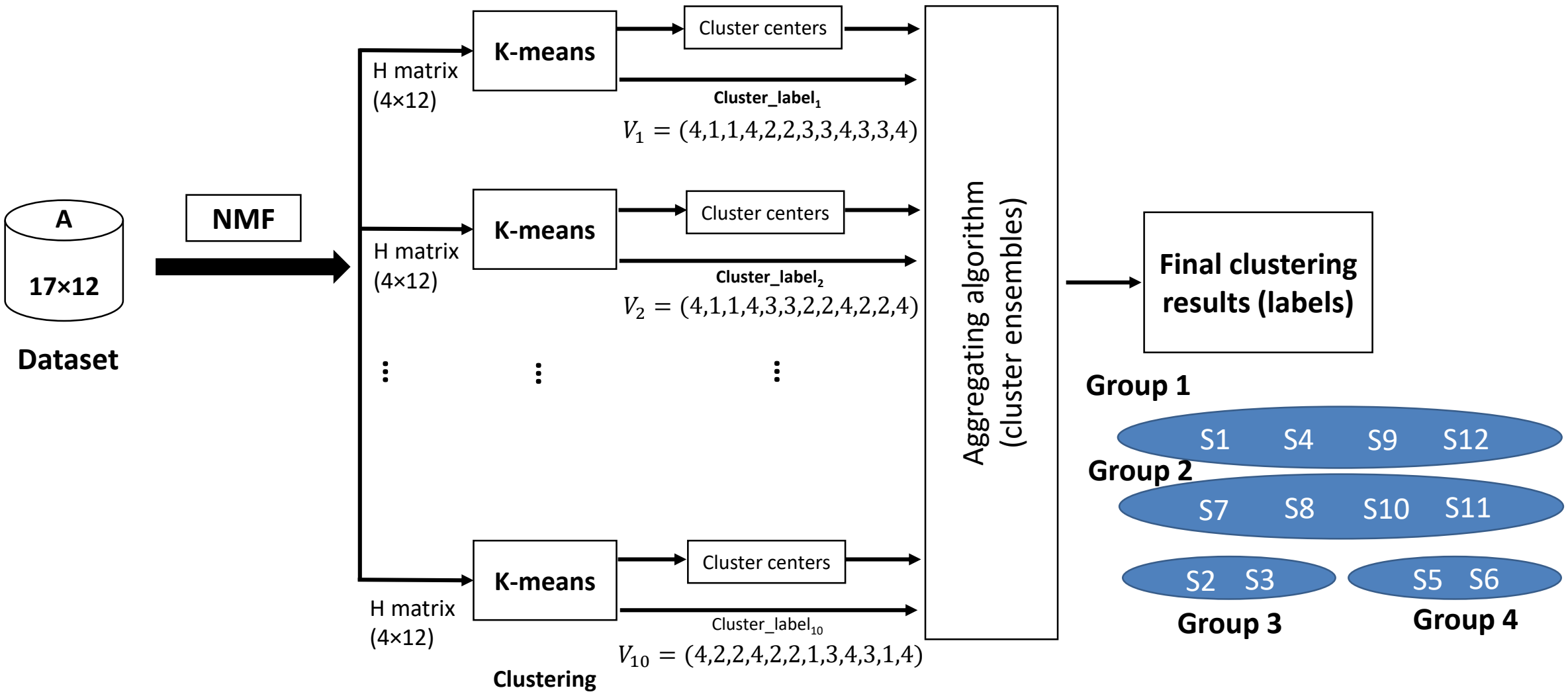
1) Multiple runs of a clustering algorithm
   a) Determine the number of clusters and assess the stability of the discovered clusters
   b) In k-means clustering: with using random restart

2) Aggregating the cluster (label) results of different clustering algorithms

[1] Ensemble clustering

# Summary: consensus approach

## 1) Multiple runs of a clustering algorithm




A comprehensive Ensemble approach for unsupervised clustering using NMF projection and k-means clustering

# Main clustering approaches

- **Partitioning algorithms**: Make different partitions heuristically and then evaluate them by some criteria (k-means & PAM)
- **Hierarchy algorithms**: Create a hierarchical decomposition of data using some criteria
- **Density-based algorithms**: based on connectivity and density functions
- **Model-based algorithms**: A model is assumed for each cluster and the idea is to find the best fit of a model

# Main clustering approaches

- **Partitioning algorithms**: Make different partitions heuristically and then evaluate them by some criteria (k-means & PAM)
- **Hierarchy algorithms**: Create a hierarchical decomposition of data using some criteria
- **Density-based algorithms**: based on connectivity and density functions
- **Model-based algorithms**: A model is assumed for each cluster and the idea is to find the best fit of a model

# Partitioning approaches

- **Partitioning method**: Construct a partition of a ***n*** samples into a set of ***k*** clusters
- Given a ***k***, find a partition of ***k*** *clusters* that optimises the chosen partitioning criterion
  - **Global optimal**: exhaustively calculate (construct) all partitions
  - **Heuristic methods**: *k-means* and *k-medoids* algorithms
    - *k-means* (MacQueen'67): Each cluster is represented by **the center of the cluster**
    - *k-medoids* or PAM (**Partition around medoids**) (Kaufman & Rousseeuw'87): Each cluster is represented by **one of the samples in the cluster**

# Partitioning around medoid (PAM)

- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other data points (samples) in the cluster is minimum.
- The difference between **k-means** and **k-medoids** is analogous to the difference between **mean** and **median**: where mean indicates the average value of all data items collected, while median indicates the value around that which all data items are evenly distributed around it.

# Partitioning around medoid (PAM)

- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other data points (samples) in the cluster is minimum.
- The difference between **k-means** and **k-medoids** is analogous to the difference between **mean** and **median**: where mean indicates the average value of all data items collected, while median indicates the value around that which all data items are evenly distributed around it.

# Partitioning around medoid (PAM)

- Partitioning around medoid **(PAM)** is the robust version of the K-means algorithm.
- Both algorithms attempt to minimize the squared-error (i.e., cost function) but the K-medoid algorithm is **more robust to noise** than K-means algorithm.

## Algorithm

1) Initialisation: select **k** random samples (data points) as the medoids.
2) Associate each data point to the closest medoid by using any common distance metric methods.
3) While the cost decreases: for each medoid **m**, for each data point $\mathbf{S_i}$ which is not a medoid:
   a. Swap **m** and $\mathbf{S_i}$, associate each data point to the closest medoid, recompute the cost.
   b. If the total cost is more than that in the previous step, undo the swap.

PAM is less sensitive to outliers than other partitioning algorithms.

# PAM clustering in R

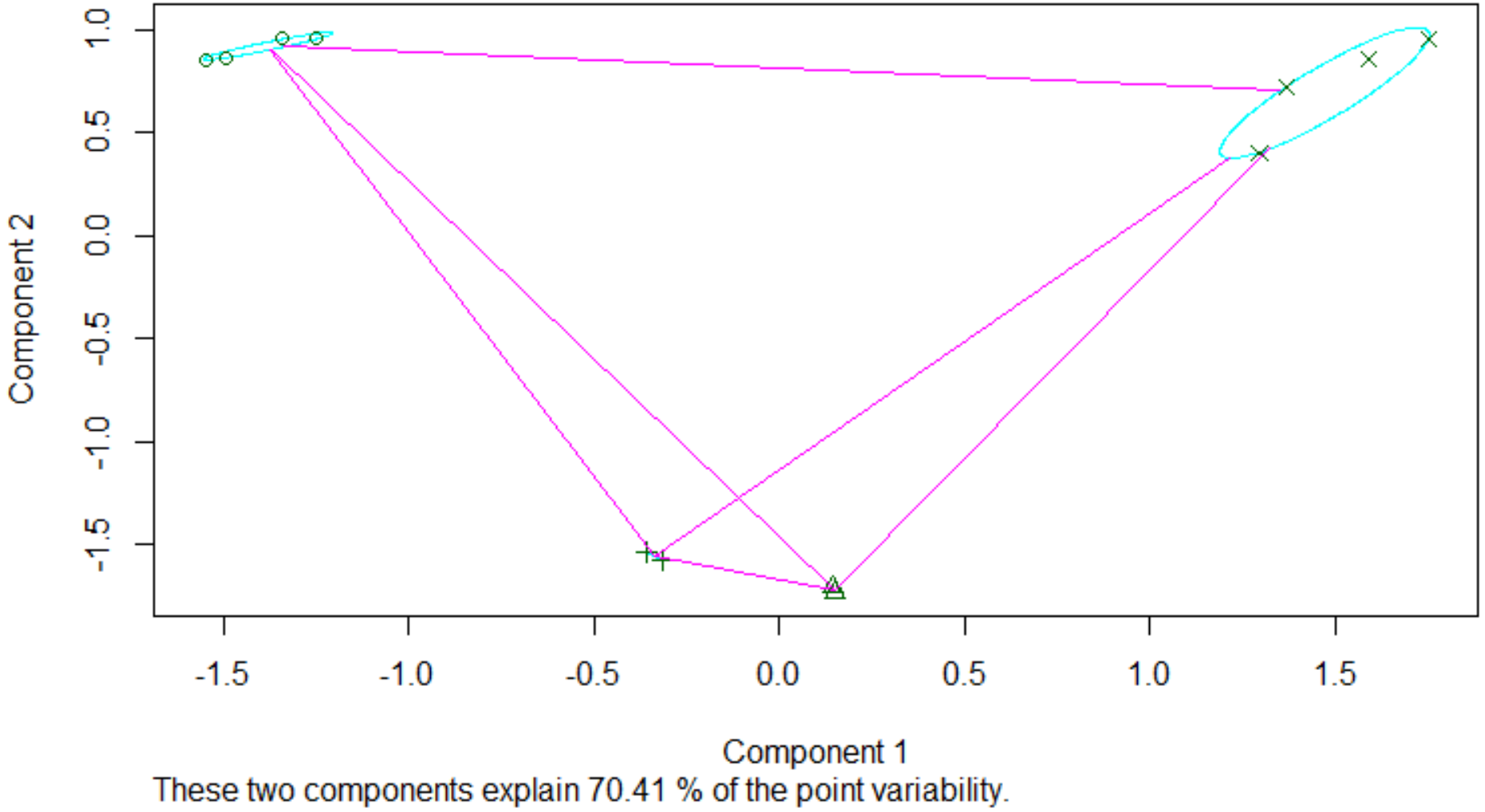
```
#-----------------------------------------------------------------------------------------
# Consider only 12 samples out of 220 with 4 metagenes
setwd("D:/Live")  # change the path to your working directory including the following csv file
Small_dataset_cluster_analysis <- read.csv("H_matrix_17_8_k4_4.csv",row.names = 1)
rownames(Small_dataset_cluster_analysis) <- c("Metagene_1","Metagene_2","Metagene_3","Metagene_4")
min(Small_dataset_cluster_analysis)  # [1] 4.14e-70
max(Small_dataset_cluster_analysis)  # [1] 9.434869
Small_dataset_cluster_analysis_0To1 <- Data_Range_Into_01(Small_dataset_cluster_analysis)
min(Small_dataset_cluster_analysis_0To1) # [1] 0
max(Small_dataset_cluster_analysis_0To1) # [1] 1
# PAM clustering algorithm
library(cluster) # k-medoid (PAM) function is in this library
Pam_Model <- pam(t(Small_dataset_cluster_analysis_0To1),k = 4)  # 4 features (metagenes) * 12 samples
plot(Pam_Model)
Pam_Model$medoids
Pam_Model$clustering
#-----------------------------------------------------------------------------------------
```

# PAM clustering in R

```
#-----------------------------------------------------------------------------------
# Consider only 12 samples out of 220 with 4 metagenes
setwd("D:/Live")  # change the path to your working directory including the following csv file
Small_dataset_cluster_analysis <- read.csv("H_matrix_17_8_k4_4.csv",row.names = 1)
rownames(Small_dataset_cluster_analysis) <- c("Metagene_1","Metagene_2","Metagene_3","Metagene_4")
min(Small_dataset_cluster_analysis)  # [1] 4.14e-70
max(Small_dataset_cluster_analysis)  # [1] 9.434869
Small_dataset_cluster_analysis_0To1 <- Data_Range_Into_01(Small_dataset_cluster_analysis)
min(Small_dataset_cluster_analysis_0To1) # [1] 0
max(Small_dataset_cluster_analysis_0To1) # [1] 1
# PAM clustering algorithm
library(cluster) # k-medoid (PAM) function is in this library
Pam_Model <- pam(t(Small_dataset_cluster_analysis_0To1),k = 4)  # 4 features (metagenes) * 12 samples
plot(Pam_Model)
Pam_Model$medoids
Pam_Model$clustering
#-----------------------------------------------------------------------------------
```

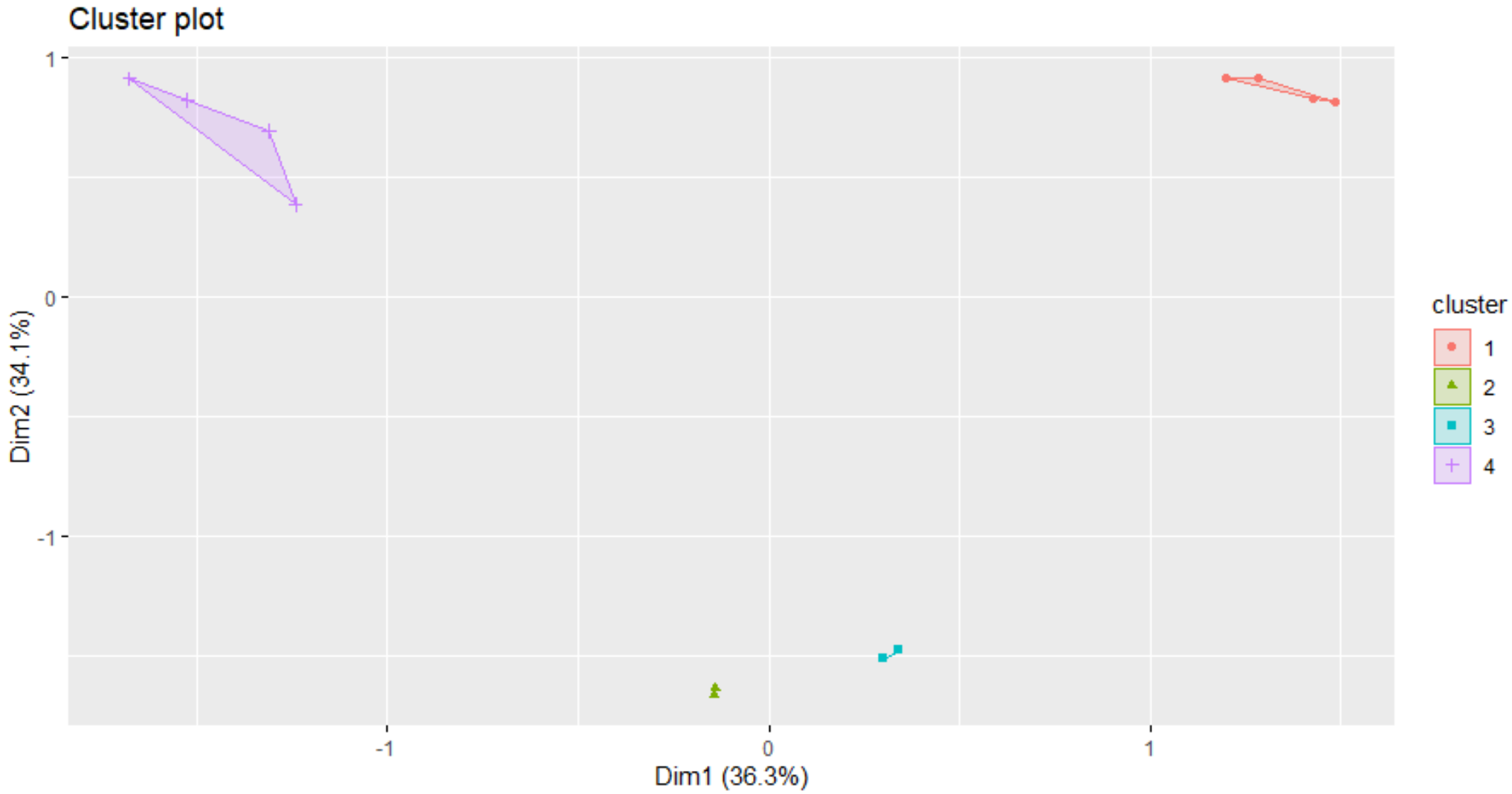
# PAM clustering in R

```
Pam_Model <- pam(t(Small_dataset_cluster_analysis_0To1),k = 4)  # 4 features (metagenes) * 12 samples
plot(Pam_Model)
```



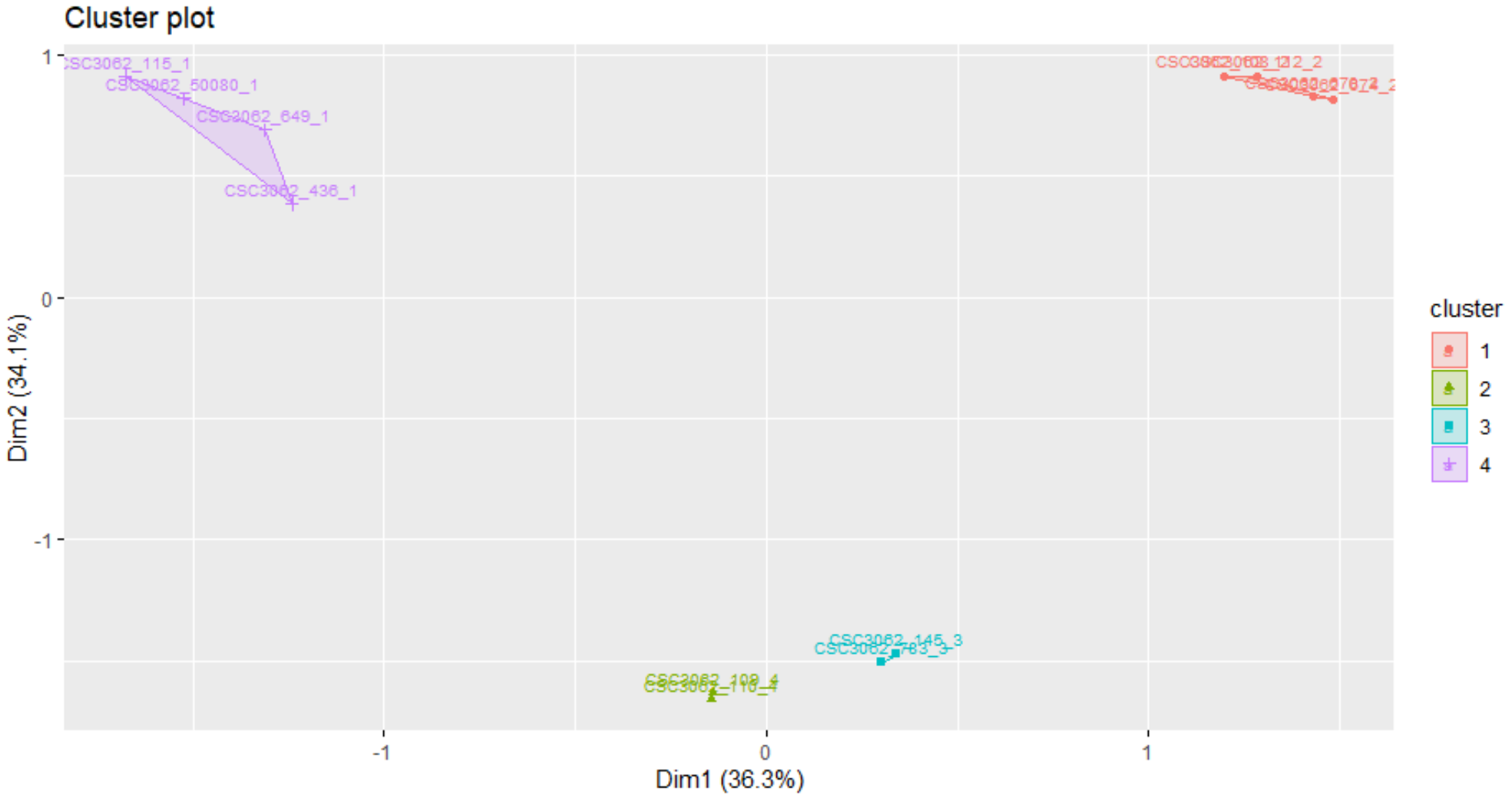
# Visualising PAM clustering results

```
# Visualize
fviz_cluster(Pam_Model,geom = "point",shape = NULL,labelsize = 8)
```



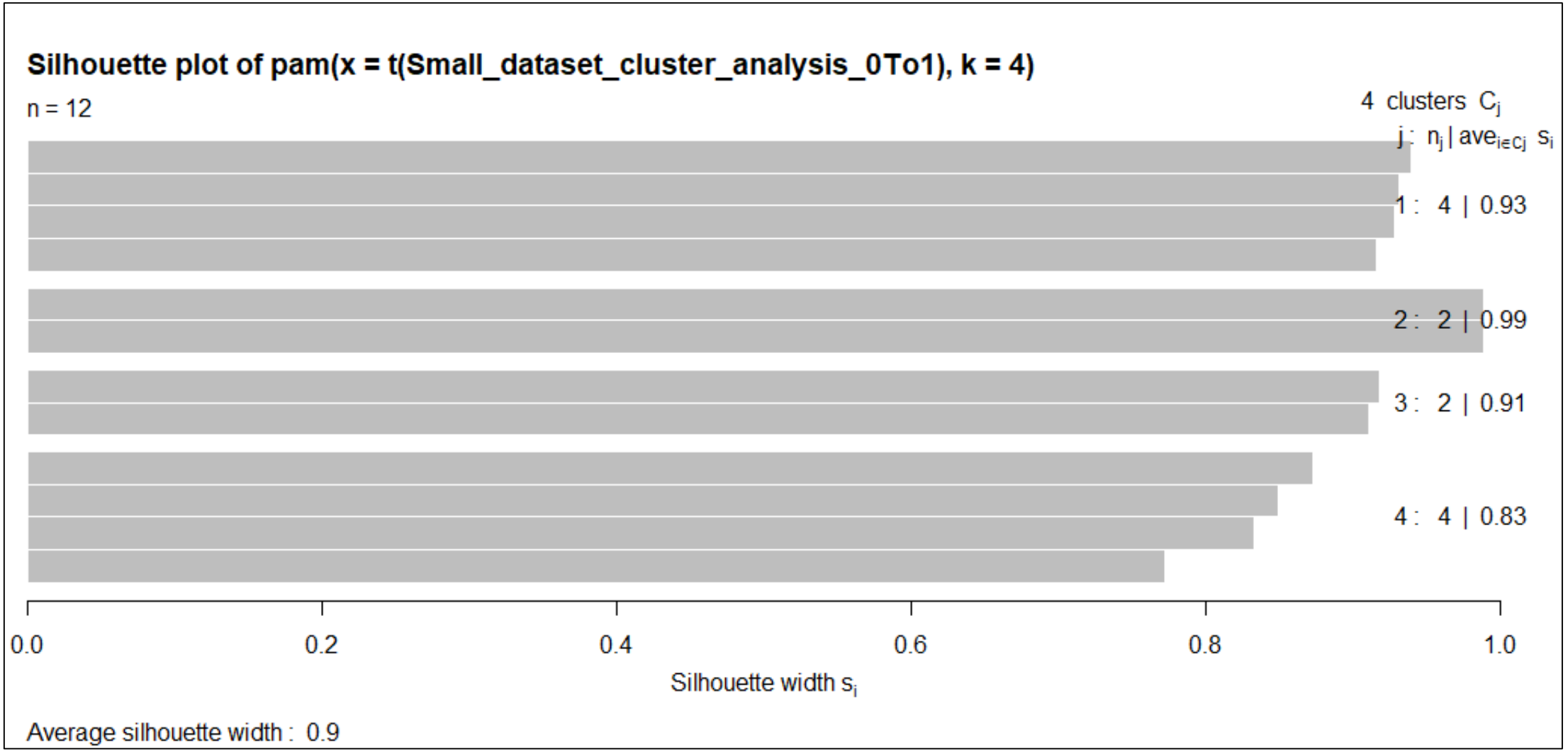
# Visualising PAM clustering results

```
library("factoextra")
fviz_cluster(Pam_Model,shape = NULL,labelsize = 8)
```



# Cluster validity using silhouette

```
Pam_Model <- pam(t(Small_dataset_cluster_analysis_0To1),k = 4)  # 4 features (metagenes) * 12 samples
plot(Pam_Model)
```



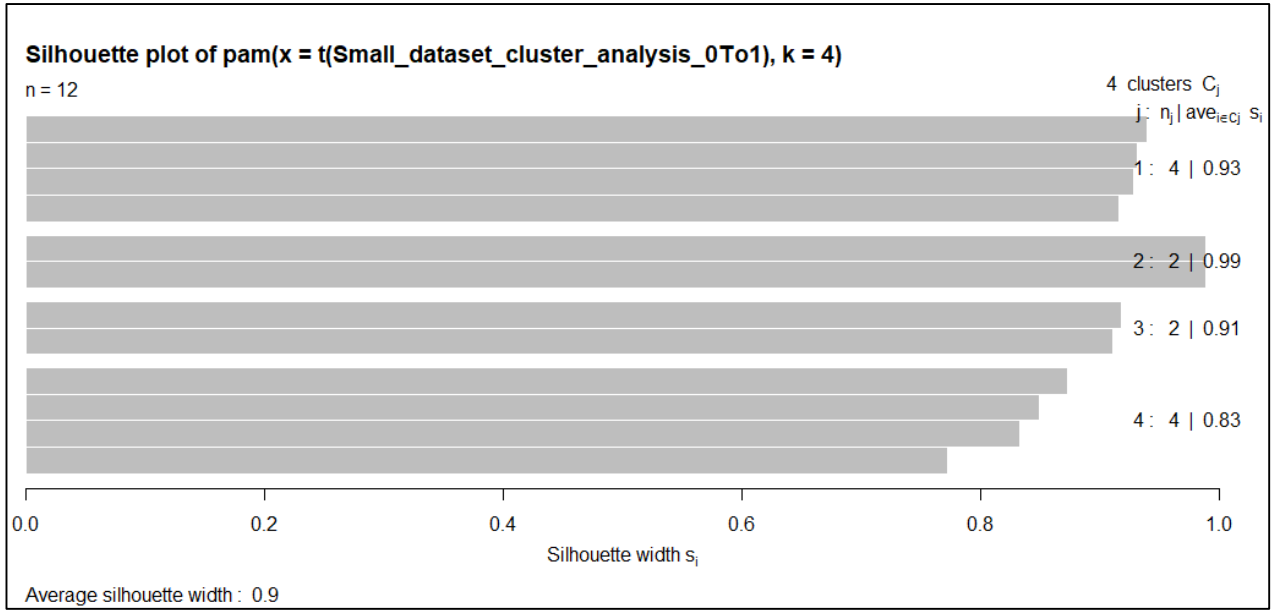
# Silhouette information

```
> Pam_Model$silinfo
$widths
               cluster neighbor sil_width
CSC3062_112_2        1        2 0.9395337
CSC3062_670_2        1        2 0.9313826
CSC3062_674_2        1        2 0.9280016
CSC3062_108_2        1        2 0.9158198
CSC3062_110_4        2        3 0.9884642
CSC3062_109_4        2        3 0.9883783
CSC3062_145_3        3        2 0.9176832
CSC3062_783_3        3        2 0.9103288
CSC3062_50080_1      4        2 0.8723002
CSC3062_649_1        4        2 0.8487563
CSC3062_115_1        4        2 0.8322591
CSC3062_436_1        4        2 0.7719721

$clus.avg.widths
[1] 0.9286844 0.9884212 0.9140060 0.8313219

$avg.width
[1] 0.90374

>
```



# Medoids as cluster representatives

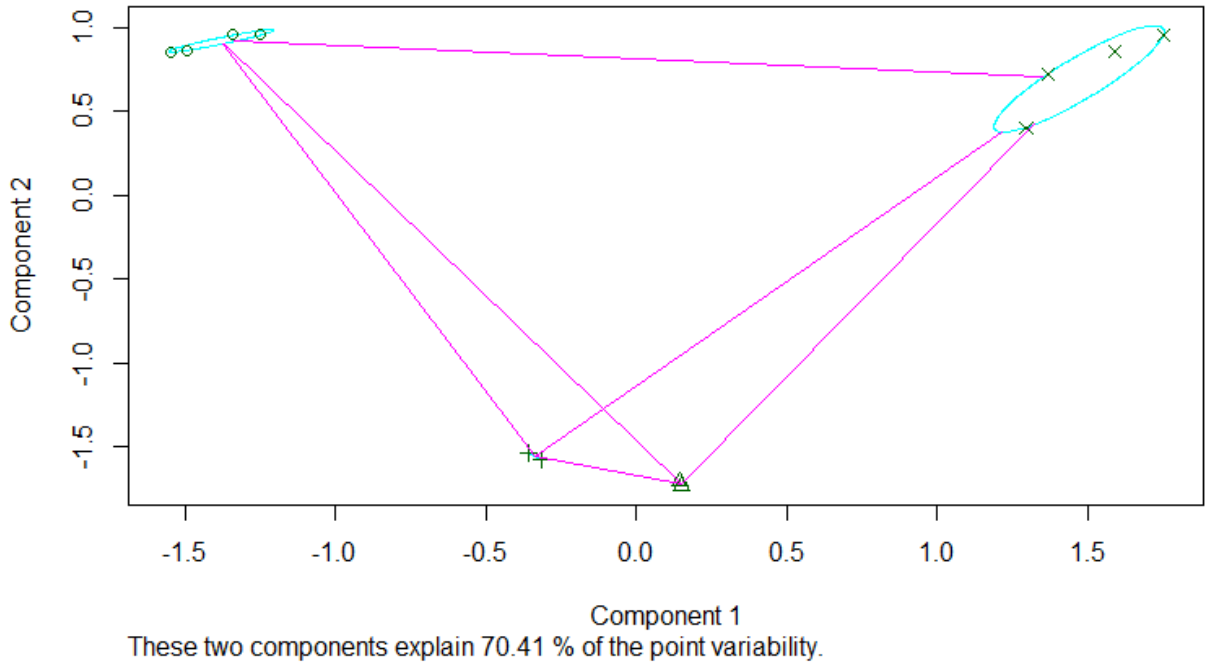
```
Pam_Model$medoids
```

```
> Pam_Model$medoids
                  Metagene_1   Metagene_2   Metagene_3   Metagene_4
CSC3062_112_2   7.633172e-02 4.172066e-27 2.634450e-34 9.671474e-01
CSC3062_110_4   2.654951e-40 5.625382e-01 1.629874e-28 4.787113e-29
CSC3062_783_3   3.608274e-32 5.022959e-02 6.117725e-01 1.660626e-34
CSC3062_50080_1 8.314318e-01 1.158338e-18 1.691378e-40 4.684605e-20
>
```

# PAM clustering results

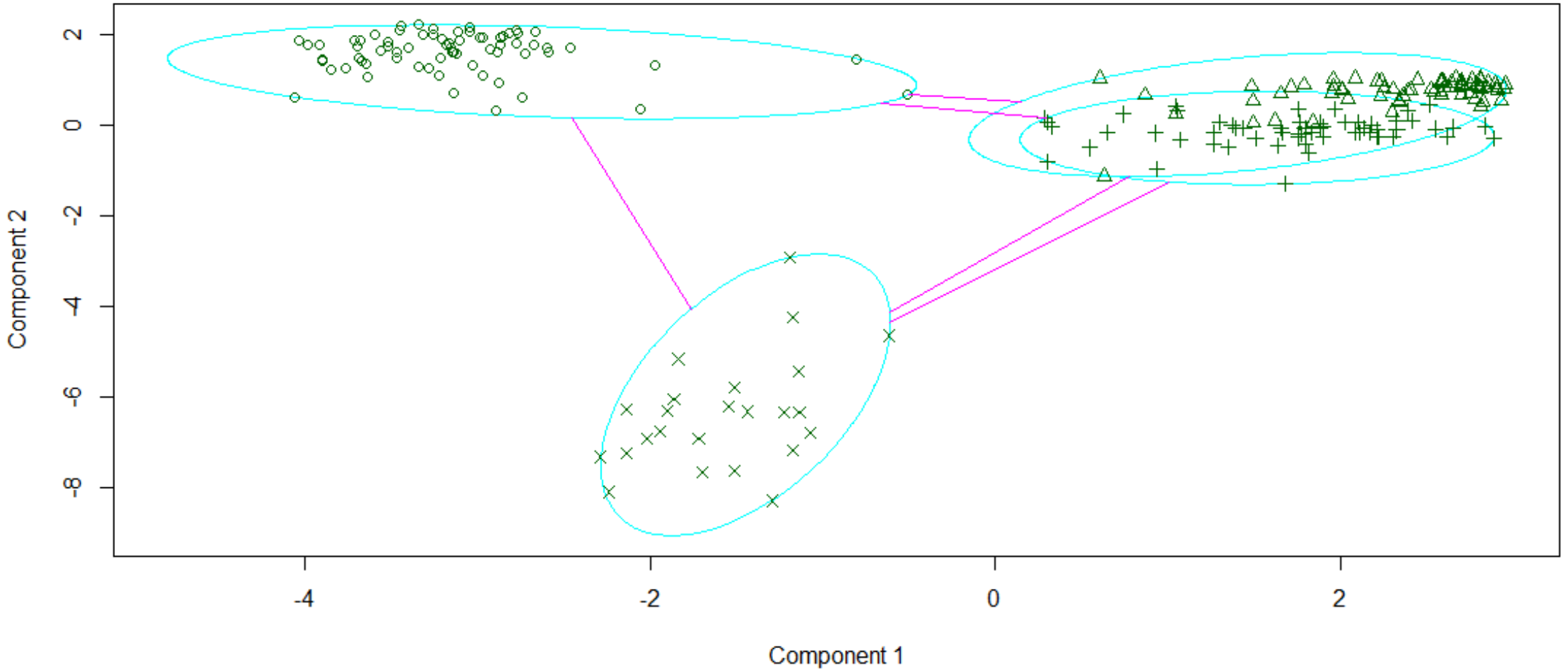
```
Pam_Model$clustering
```

```
> Pam_Model$clustering
  CSC3062_108_2   CSC3062_109_4   CSC3062_110_4   CSC3062_112_2   CSC3062_783_3   CSC3062_145_3   CSC3062_649_1   CSC3062_115_1
              1               2               2               1               3               3               4               4
  CSC3062_670_2 CSC3062_50080_1   CSC3062_436_1   CSC3062_674_2
              1               4               4               1
>
```



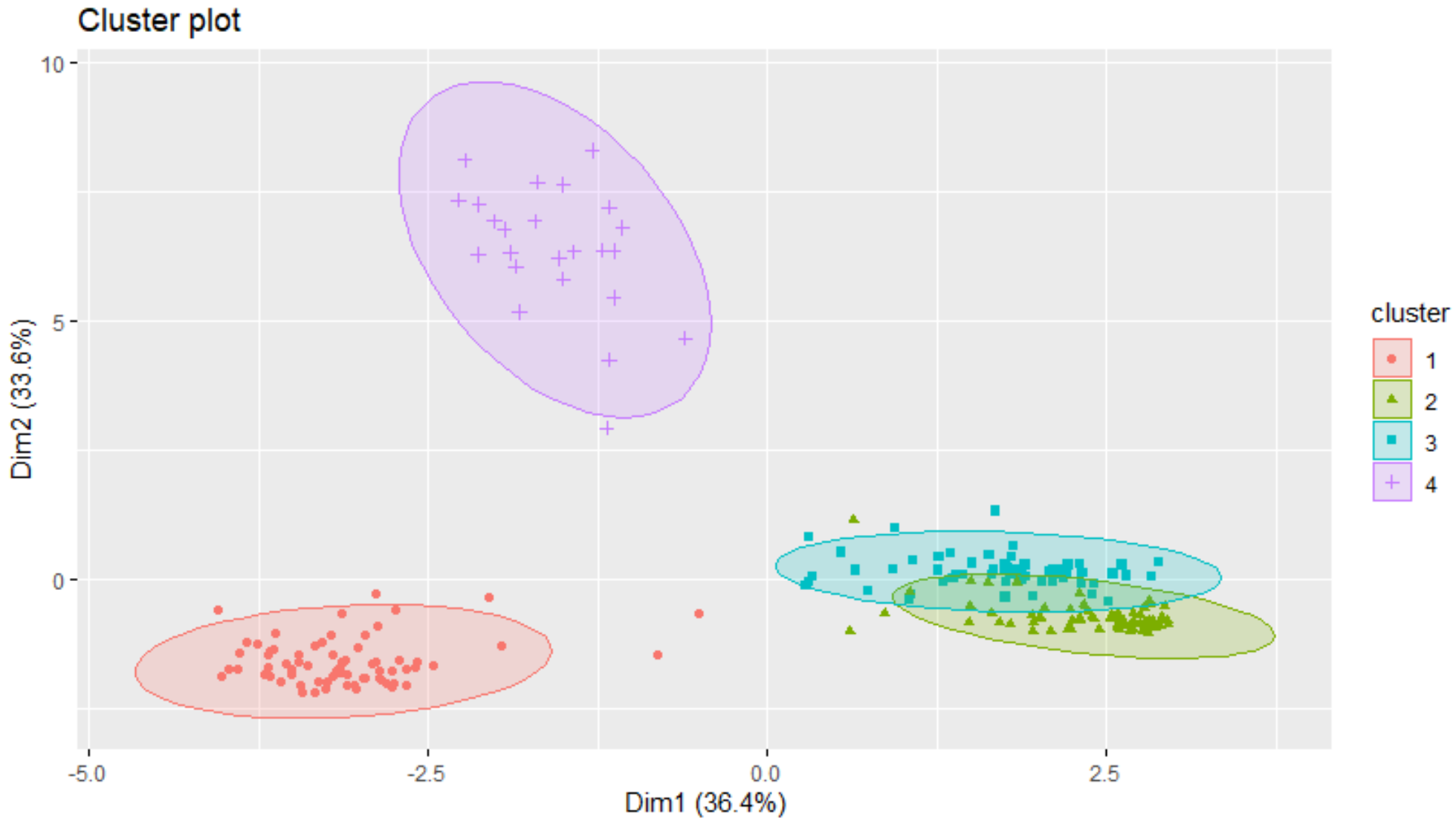
# PAM clustering in R (n=220 samples)

```
#-------------------------------------------------------------------------------------------
Pam_Model <- pam(t(Complete_dataaset_220),k = 4)  # 17 features * 220 samples
```



# PAM clustering in R (n=220 samples)

```
# Visualise the result of clustering
fviz_cluster(Pam_Model,geom = "point",shape = NULL,labelsize = 8,ellipse.type = "norm")
```



# Any Questions?