Introduction to Transcriptomics Analysis

Class 14 - Downstream Analysis I
Practice about Clustering.

INSTRUCTOR:
Aureliano Bombarely
Department of Bioscience
Universita degli Studi di Milano
aureliano.bombarely@unimi.it

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

Data source

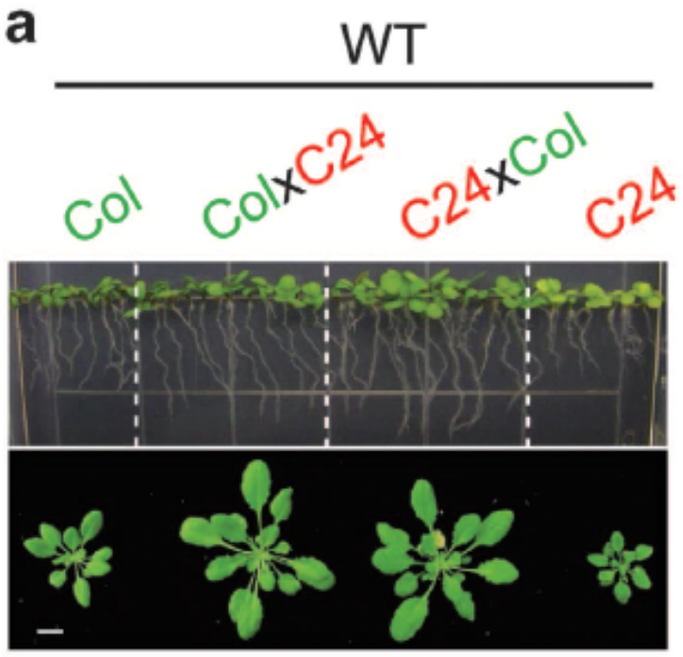
OPEN

ARTICLE

Citation: Cell Discovery (2016) 2, 16027; doi:10.1038/celldisc.2016.27
www.nature.com/celldisc

The chromatin remodeler DDM1 promotes hybrid vigor by regulating salicylic acid metabolism

Qingzhu Zhang[1], Yanqiang Li[1], Tao Xu[2], Ashish Kumar Srivastava[1], Dong Wang[1], Liang Zeng[1], Lan Yang[1], Li He[1], Heng Zhang[1], Zhimin Zheng[1], Dong-Lei Yang[1], Cheng Zhao[1], Juan Dong[3], Zhizhong Gong[4], Renyi Liu[1], Jian-Kang Zhu[1,5]



From Exercise 12.1.4

- `ge_table`

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 1: Hierarchical clustering for CummeRBund.

Preparation before the exercise:

1- Select the matrix with the expression data for the DE genes

```
dge_table = ge_table[ge_table$pval <= 0.05,]

row.names(dge_table) = dge_table$id

dge_table = dge_table[,c(2:13)]
```

```
> head(dge_table)
    FPKM.Artha_C24_Rep1 FPKM.Artha_C24_Rep2 FPKM.Artha_C24_Rep3 FPKM.Artha_C24xCol_Rep1 FPKM.Artha_C24xCol_Rep2
52            4.6609248           5.1396839            3.079771               2.5309586               3.378501
68            5.2725765           4.5371768            8.231462               3.4939138               2.216994
89            7.2489841           7.2659231            8.432974               6.6637972               7.858001
100          17.9162695          18.9155772           18.475026              15.4499275              16.456770
351          48.8143705          45.0038931           50.663231              43.5557659              43.797780
414           0.5825911           0.9154612            2.379365               0.9631538               1.085114
    FPKM.Artha_C24xCol_Rep3 FPKM.Artha_Col_Rep1 FPKM.Artha_Col_Rep2 FPKM.Artha_Col_Rep3 FPKM.Artha_ColXC24_Rep1
52                 1.709900           3.4443220            2.904350            2.124660                2.198507
68                 4.606045           5.3011560            4.958441            7.526265                3.490425
89                 6.695981           9.6040859            8.452558            8.593418                7.003947
100               18.942256          18.0459464           19.824542           20.891862               16.149299
351               48.262689          47.3690306           47.803516           49.435798               46.198851
414                3.324734           0.3462995            1.842947            2.315747                1.500637
    FPKM.Artha_ColXC24_Rep2 FPKM.Artha_ColXC24_Rep3
52                 2.543655                1.965995
68                 2.586179                6.467795
89                 6.063115                7.802854
100               16.409068               18.655557
351               43.592466               45.313312
414                1.015271                2.407164
```

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.1: Calculating gene expression distance.

The goal of the exercise is calculate a distance matrix for the gene expression table.

Steps:

1. Run dist() on the dge_table object (it will run an euclidean distance by default):

```
gene_dist = dist(dge_table)
```

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.2: Performing the clustering.

The goal of the exercise is calculate to run and hierarchical classification on the distance matrix and then select a number of clusters

Steps:

1. Run hclust() on the gene_distance object using the “ward.D” method:

```
gene_hclust = hclust(gene_dist, method = "ward.D")
```

2. Plot the hierarchical clustering

```
plot(gene_hclust, labels = FALSE)

abline(h = 1000, col = "brown", lwd = 2)
```

3. Select a cutoff with six clusters

```
gene_cluster = cutree(gene_hclust, k = 6)
```

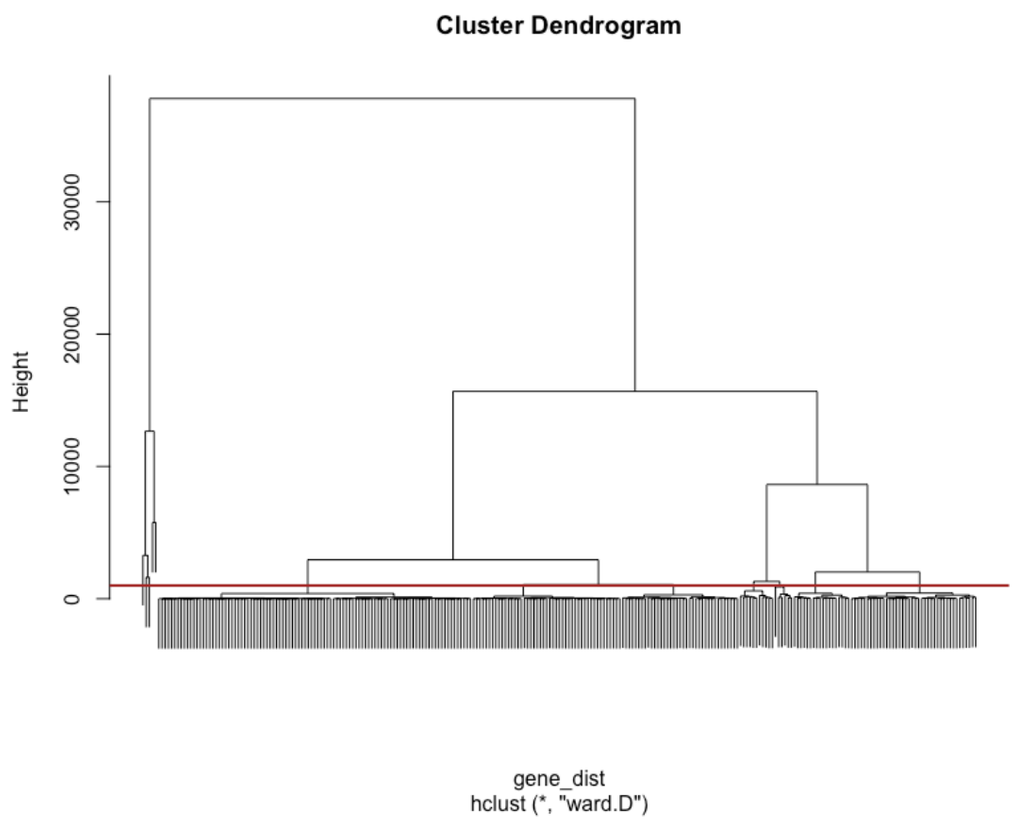
- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.2: Performing the clustering.

The goal of the exercise is calculate to run and hierarchical classification on the distance matrix and then select a number of clusters

Steps:

2. Plot the hierarchical clustering



- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.2: Performing the clustering.

The goal of the exercise is calculate to run and hierarchical classification on the distance matrix and then select a number of clusters

Steps:

3. Select a cutoff with six clusters.

```
gene_cluster_k6 = cutree(gene_hclust, k = 6)
```

4. Retrieve the number of genes per cluster.

```
table(gene_cluster_k6)
```

5. Select a new cutoff with three clusters.

```
gene_cluster_k3 = cutree(gene_hclust, k = 3)
```

6. Retrieve the number of clusters again

```
table(gene_cluster_k3)
```

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.3: Visualising the clusters.

The goal of the exercise is visualise the clusters retrieved in the previous step. We will use K=3 since K=6 produces clusters with only one member.

Steps:

1. Transform the gene_cluster_k3 into a tibble.

```
library(tidyverse)
gene_cluster_k3 = cutree(gene_hclust, k = 3) %>%
+ enframe() %>%
+ dplyr::rename(gene = name, cluster = value)
```

2. Retrieve the expression data

```
dge_means_table = ge_table[ge_table$pval <= 0.05,]
row.names(dge_means_table) = dge_means_table$id
dge_means_table = dge_means_table[,c(14,16,18,20)]
dge_means_table$gene = row.names(dge_means_table)
gde_means_ttable= dge_means_table %>% gather("condition", "expression",
-gene)
gde_means_ttable = gde_means_ttable[order(gde_means_ttable$gene),]
```

- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.3: Visualising the clusters.

The goal of the exercise is visualise the clusters retrieved in the previous step. We will use K=3 since K=6 produces clusters with only one member.

Steps:

3. Join the expression and the cluster tibbles.

```
gde_mean_cluster = gde_means_ttable %>% inner_join(gene_cluster_k3, by
= "gene")
```

4. Plot the clusters with ggplot2

```
gde_mean_cluster %>% ggplot(aes(condition, expression)) +
  geom_line(aes(group = gene)) +
  geom_line(stat = "summary", colour = "brown", size = 1.5, aes(group = 1)) +
  facet_grid(rows = vars(cluster))
```

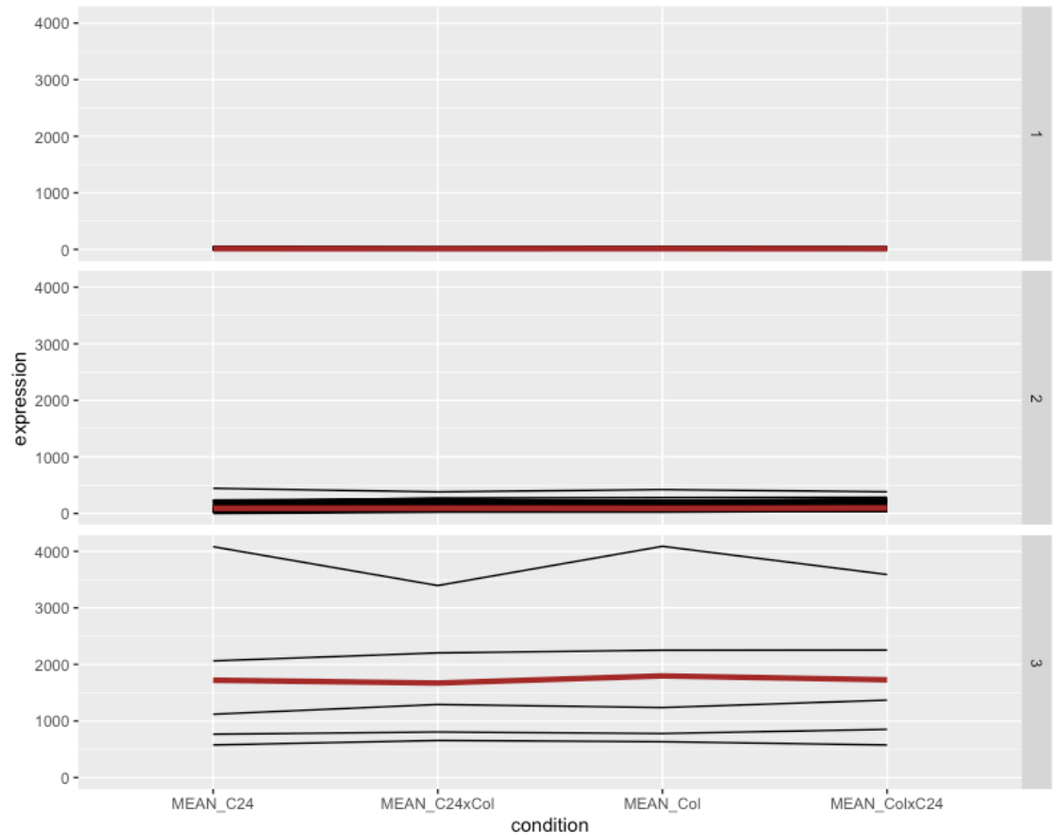
- Exercise 1: Hierarchical clustering for CummeRBund.

Exercise 1.3: Visualising the clusters.

The goal of the exercise is visualise the clusters retrieved in the previous step. We will use K=3 since K=6 produces clusters with only one member.

Steps:

4. Plot the clusters with ggplot2



Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 2: K-means clustering for CummeRBund.

Preparation before the exercise:

1- (Same than before) Select the matrix with the expression data for the DE genes

```
dge_table = ge_table[ge_table$pval <= 0.05,]

row.names(dge_table) = dge_table$id

dge_table = dge_table[,c(2:13)]
```

```
> head(dge_table)
    FPKM.Artha_C24_Rep1 FPKM.Artha_C24_Rep2 FPKM.Artha_C24_Rep3 FPKM.Artha_C24xCol_Rep1 FPKM.Artha_C24xCol_Rep2
52            4.6609248           5.1396839            3.079771               2.5309586               3.378501
68            5.2725765           4.5371768            8.231462               3.4939138               2.216994
89            7.2489841           7.2659231            8.432974               6.6637972               7.858001
100          17.9162695          18.9155772           18.475026              15.4499275              16.456770
351          48.8143705          45.0038931           50.663231              43.5557659              43.797780
414           0.5825911           0.9154612            2.379365               0.9631538               1.085114
    FPKM.Artha_C24xCol_Rep3 FPKM.Artha_Col_Rep1 FPKM.Artha_Col_Rep2 FPKM.Artha_Col_Rep3 FPKM.Artha_ColXC24_Rep1
52                 1.709900           3.4443220            2.904350            2.124660                2.198507
68                 4.606045           5.3011560            4.958441            7.526265                3.490425
89                 6.695981           9.6040859            8.452558            8.593418                7.003947
100               18.942256          18.0459464           19.824542           20.891862               16.149299
351               48.262689          47.3690306           47.803516           49.435798               46.198851
414                3.324734           0.3462995            1.842947            2.315747                1.500637
    FPKM.Artha_ColXC24_Rep2 FPKM.Artha_ColXC24_Rep3
52                 2.543655                1.965995
68                 2.586179                6.467795
89                 6.063115                7.802854
100               16.409068               18.655557
351               43.592466               45.313312
414                1.015271                2.407164
```

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 2: K-means clustering for CummeRBund.

Exercise 2.1: Running different K's

The goal of the exercise is to run K-means clustering with a range of K's.

Steps:

1. Run the Kmer clustering for K=3

```
gene_cluster_kmeans3 = kmeans(gene_dist, 3)
```

2. Get the number of genes per cluster

```
table(gene_cluster_kmeans3$cluster)
```

3. Run now for K=2 to K=10

```
clusters = list()
kn=0
for(i in 2:10) {
    kn=kn+1
    print(kn, i)
    clusters[[kn]] = kmeans(gene_dist, i)
}
```

Outline of Topics

- Exercise 1: Hierarchical clustering for CummeRBund.
  - Exercise 1.1: Calculating gene expression distance.
  - Exercise 1.2: Performing the clustering.
  - Exercise 1.3: Visualising the clusters.
- Exercise 2: K-means clustering for CummeRBund.
  - Exercise 2.1: Running different K's.
  - Exercise 2.2: Selecting the optimal number of K's

- Exercise 2: K-means clustering for CummeRBund.

Exercise 2.2: Selecting the optimal number of K's

The goal of the exercise is to run select the right K based in the package fpc.

Steps:

1. Run fps for each of the cluster

```
library(fpc)

cluster.stats(gene_dist, clusters[[1]]$cluster, clusters[[2]]$cluster)
```