PCA_PAM50 : Analyse ACP à partir de l'expression des 50 gènes de la PAM50

Liste des fonctions nécessaires :

Fonction R	Description
as.matrix()	convertir un objet en matrice
scale()	centrer et reduire
boxplot()	Réprésenter la distribution des variables quantitatives
dudi.pca()	Réaliser une ACP
get_pca_var()	renvoie les résultats de l'ACP pour les variables
get_pca_ind()	renvoie les résultats de l'ACP pour les individus
dist()	calcul des distances entre colonnes
hclust()	clustering à partir de distances calculées
cutree()	renvoie les groupes calculés par le clustering pour une valeur de k donnée

Data : Data_Pam50.csv

La signature PAM50, développée par Parker et *al.* (1), est une liste de 50 gènes qui peuvent aider à prédire les sous-types de cancer du sein des patients. Dans cette analyse, vous devrez analyser les données des patients en fonction de l'expression des 50 gènes.

Les données correspondent à l'expression de 50 gènes (colonnes) pour 72 patients (lignes). Une colonne supplémentaire correspond au sous type réel de chaque patient.

(1): Parker et al., Supervised risk predictor of breast cancer based on intrinsic subtypes. J Clin Oncol. 2009 Mar 10;27(8):1160-7. doi : 10.1200/JCO.2008.18.1370.

1- Charger et visualiser les données

```
setwd("/Users/patricia/Documents/UEs/_S7_STAT_2019//PCA/")
data=read.table("Data_Pam50.csv",header=T,row.names=1,sep=",",dec = ".")
# Afficher les premières lignes pour les 5 première colonnes
head(data[,1:5])
# Réprésenter la distribution des variables quantitatives avec la fonction boxplot()
boxplot(data[2:50])
# centrer et reduire avec la fonction scale()
data2=scale(data[2:50])
```

```r
# données "normées" : gammes de variation et unités rendues homogénes
boxplot(data2)
```

2- Réaliser une ACP avec la fonction dudi.pca (package nécessaire: ade4)

- Analyser les données à l'aide d'une ACP (sans prednre en compte la première colonne qui contient une variable qualitative pour les sous types des patients)
- Les variables / colonnes correspondent aux 50 gènes de la PAM50
- Les individus / lignes correspondent aux patients
- Normaliser les données grâce à l'argument scale de la fonction dudi.pca()
- Réprésenter les valeurs d'inerties de l'ACP pour définir le nombre de composantes à retenir

```r
library(ade4)
library(factoextra)
?dudi.pca
# Comprendre les arguments: scale, scannf et nf de la fonction dudi.pca()
PCA.res=dudi.pca(data[2:51],scale=TRUE, scannf=FALSE, nf=ncol(data[2:51]))
PCA.res
PCA.res$eig
# cumulative percentage of variance
inertia.dudi(PCA.res)
# Visualiser le pourcentage de variances expliquée cumulée.
# Sélection du nombre d'axe par la méthode du coude
fviz_eig(PCA.res)
```

3- Analyser la qualité de la réprésentation des données : variables et individus

```r
# Résultats des variables / gènes de la PAM50
res.var <- get_pca_var(PCA.res)
#head(res.var$coord )          # Coordonnées
head(res.var$contrib)       # Contributions aux axes
head(res.var$cos2)            # Qualité de représentation
# Résultats des individus / Patients
res.ind <- get_pca_ind(PCA.res)
#head(res.ind$coord)           # Coordonnées
head(res.ind$contrib)         # Contributions aux axes
head(res.ind$cos2)            # Qualité de représentation
```

4- Visualiser la qualité de la réprésentation des données : variables et individus avec le package

Basé sur *http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/80-acp-dans-r-avec-ade4-scripts-faciles/*

- Graphique des individus. Coloration en fonction du cos2 (qualité de représentation). Les individus similaires sont groupés ensemble.
- Graphique des variables. Coloration en fonction de la contribution des variables. Les variables corrélées positivement sont du même côté du graphique. Les variables corrélées négativement sont sur des côtés opposés du graphique.
- Graphique Biplot des individus et des variables

```r
# Graphique des individus. Coloration en fonction du cos2 (qualité de représentation).
# Les individus similaires sont groupés ensemble.
```

```r
fviz_pca_ind(PCA.res,
             col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
)
# Graphique des variables. Coloration en fonction de la contribution des variables.
# Les variables corrélées positivement sont du même côté du graphique.
# Les variables corrélées négativement sont sur des côtés opposés du graphique.
fviz_pca_var(PCA.res,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
)
# Graphique Biplot des individus et des variables
fviz_pca_biplot(PCA.res, repel = TRUE,
                col.var = "#2E9FDF",
                col.ind = "#696969"
)
```

5- Ajouter de l'information sur la PCA avec une variable qualitative (labels des patients extraits de la première colonne)

- Utiliser une variable qualitative pour colorer les individus de la PCA avec le package factoextra
- Utiliser une variable qualitative pour colorer les individus de la PCA avec le package ade4

```r
# Les variables qualitatives / catégorielles peuvent être ajoutées sur l'ACP.
# les individus par groupes.

# Première alternative avec le package factoextra
fviz_pca_ind(PCA.res,
             col.ind = data$subtype, # colorer par groupes
             palette = c("#00AFBB",  "#FC4E07", "#E7B800","#B2182B", "#D6604D"),
             addEllipses = TRUE, # Ellipses de concentrations
             ellipse.type = "confidence",
             legend.title = "Groups",
             repel = TRUE)
# Première alternative avec le package ade4
groups <- as.factor(data$subtype)
s.class(PCA.res$li,# $li correspond aux lignes/résidus/patients
        fac = groups,  # colorer par groupes
        col = c("#00AFBB",  "#FC4E07", "#E7B800","#B2182B", "#D6604D")
)
```

6- Regroupement des patients avec une méthode de clustering hierarchique par comparaison de la distance euclideienne des données d'expression

- Calculer de la distance entre les patients d'après l'expression des 50 genes.
- *Clustering* des données de distances pour identifier les 5 groupes de patients.
- Générer un plot des résultats de *clustering*.
- Visualiser des résultats avec une heatmap

```r
# Calcul de la distance de l'expression des 50 genes.
mat_distance = dist(data[,2:50])
```

```r
head(mat_distance)
# Clustering des données de distances pour identifier les 5 groupes de patients.
CAH <- hclust(mat_distance, method = "average")
# Plot des résultats de clustering.
plot(CAH)
# Regourpement des patients en 5 groupes avec la fonction cutree
grpecluster <- cutree(CAH, k = 5)
head(grpecluster)
# Visualisation des résultats avec la fonction heatmap()
# Nécessite de convertir les résultats de distance dans une matrice
# avec la fonction as.matrix()
heatmap(as.matrix(mat_distance))
```

7- Combinaison des résultats de clustering avec l'ACP

- Visualisation de l'ACP en utilisant une nouvelle variable qualitative correspondant aux 5 groupes de patients obtenus d'après le *clustering*

```r
res <- scatter(PCA.res, clab.row = 0, posieig = "none")
s.class(
  PCA.res$li,
  fac = as.factor(grpecluster),
  col = c("#00AFBB",  "#FC4E07", "#E7B800","#B2182B", "#D6604D")
)
```