TP2

Justine Guégan - j.guegan-ihu@icm-institute.org
Guillaume Meurice - guillaume.meurice@gustaveroussy.fr

24 novembre 2016

Les données nécessaires à cette séance se trouvent sur le site web :

https://pf-bb.github.io/CentraleSupelec-R-genomics/TP2.html.

Objectif

L'objectif du TP est d'étudier la modulation de l'expression des gènes entre des échantillons de cancer du sein et des échantillons non tumoraux. Pour cela, des expériences de RNA-seq ont été réalisées.

Répondez aux questions dans un document Rmarkdown produisant un fichier **PDF** ou **HTML**.

Données

Les ARNm de 17 échantillons de tumeurs du sein de 3 types, HER2 positif (HER2), triple négatif (TNBC), non triple négatif (Non-TNBC), et 3 échantillons de sein normal (épithelium) ont été séquencés par Illumina HiSeq2000. Les données brutes sont disponibles sur le site du NCBI, sous sra, étude SRP032789.

Ces données ont été utilisées dans les 2 articles suivants :

- J.Eswaran et al. Transcriptomic landscape of breast cancers through mRNA sequencing. Scientific Report (2012) article
- J.Eswaran et al. RNA sequencing of cancer reveals novel splicing alterations. Scientific Report (2013) article

Afin d'obetnir la table d'expression utilisée dans ce TP, les données brutes (fastq) ont préalablement été nettoyées, alignées sur le génome de référence humain hg19 avec l'outil STAR, puis la quantification a été faite avec l'outil feature-count. Cet outil génère un tableau avec en ligne les gènes, et en colonnes les indivuds. On y retrouve l'annotation des gènes ainsi que les valeurs d'expression brutes, aussi appelées "comptages".

Pré-requis

Ce TP nécessite le chargement de 2 packages :

- package pheatmap

```
install.packages("pheatmap")
```

- package RColorBrewser

```
install.packages("RColorBrewser")
```

Analyse des données

1. Lecture des fichiers de données

> **Question** : Chargez en mémoire la table de comptage (fichier `counts.txt`) et la description des échantillons (fichier `annot_sample.txt`).
> Expliquez chacune des options utilisée.

```
comptage = read.table("counts.txt", sep="\t", header=TRUE)
design = read.table("annot_sample.txt", sep="\t", header=TRUE)
```

> **Question** : Quelle est la classe des objets chargés et quelles en sont les dimensions ? Affichez un extrait de chaque objet.

```
class(comptage)
```

```
## [1] "data.frame"
```

```
class(design)
```

```
## [1] "data.frame"
```

```
dim(comptage)
```

```
## [1] 26423    26
```

```
dim(design)
```

```
## [1] 20  4
```

```
head(comptage)
```

```
##    Geneid
## 1   SGIP1
## 2   AZIN2
## 3 SLC45A1
## 4  NECAP2
## 5   CLIC4
## 6   AGBL4
##
## 1 chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr
## 2                                                                                 chr1;chr1;chr1;chr1;chr
## 3                                                                                              chr1;chr
## 4                                                                                                   chr
## 5
```

```
## 6                                                                    chr1;chr1;chr1;chr1;chr1;chr1;chr
##
## 1 66999252;66999639;67091530;67098753;67101627;67105460;67108493;67109227;67126196;67133213;67136678
## 2
## 3
## 4
## 5
## 6
##
## 1 66999355;67000051;67091593;67098777;67101698;67105516;67108547;67109402;67126207;67133224;67136702
## 2
## 3
## 4
## 5
## 6
##                                                       Strand Length SRR1027171
## 1 +;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+;+  11104        156
## 2                               +;+;+;+;+;+;+;+;+;+;+;+   2569         80
## 3                                     +;+;+;+;+;+;+;+;+   2527         20
## 4                                       +;+;+;+;+;+;+;+   2081        646
## 5                                           +;+;+;+;+;+   4434       3915
## 6                             -;-;-;-;-;-;-;-;-;-;-;-;-   2988         28
##   SRR1027172 SRR1027173 SRR1027174 SRR1027175 SRR1027176 SRR1027177
## 1         63        218        129        188         76        254
## 2         31        133        234         95         51        171
## 3          4         30         12         15         13         11
## 4        375        532       1053        840        831       1946
## 5       1274       4826       4925       7318       6271       7969
## 6          3         35         13         16         17         40
##   SRR1027178 SRR1027179 SRR1027180 SRR1027181 SRR1027182 SRR1027183
## 1        221        203        350        115         47        139
## 2        118        140        148        126         99         59
## 3         52         17         23         23         58          3
## 4        621        591        623        809        398        460
## 5       2939       2861       4255       3387       1649       2578
## 6         36         29         34         28         27          6
##   SRR1027184 SRR1027185 SRR1027186 SRR1027187 SRR1027188 SRR1027189
## 1         88        189        124         34        104        108
## 2        112         41         25         42        128        200
## 3          5          3          5         10         37         58
## 4        539        593        400        309        778       1175
## 5       5789       4143       1990       2187      10046      15334
## 6          2          7         19          5         12          9
##   SRR1027190
## 1        228
## 2        145
## 3         78
## 4       1048
## 5      26998
## 6          7
```

```r
head(design)
```

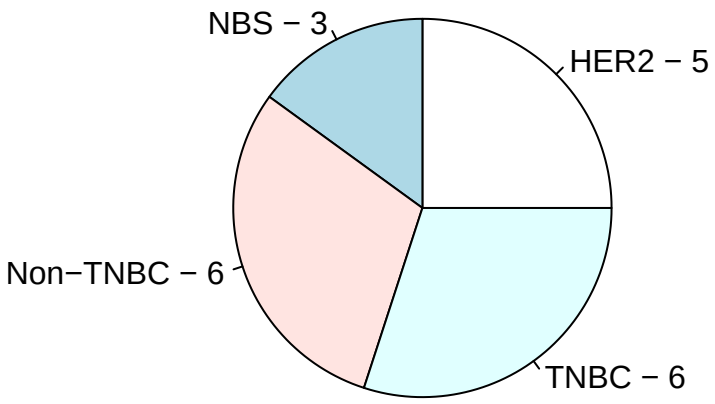
```
##   sampleName   sampleId experiment condition
```

```
## 1      TNBC1 SRR1027171 GSM1261016      TNBC
## 2      TNBC2 SRR1027172 GSM1261017      TNBC
## 3      TNBC3 SRR1027173 GSM1261018      TNBC
## 4      TNBC4 SRR1027174 GSM1261019      TNBC
## 5      TNBC5 SRR1027175 GSM1261020      TNBC
## 6      TNBC6 SRR1027176 GSM1261021      TNBC
```

2. Exploration des données

Question : Indiquez le nombre d'échantillons par `condition`. Représenter cette répartition sous forme de pie chart.

```
tpie = table(design$condition)
pie(table(design$condition),labels = paste0(names(tpie)," - ",tpie), main = "Répartition des échantillon
```



Question : A partir du fichier de comptages, créer un tableau contenant uniquement les données d'annotations des gènes et un tableau contenant uniquement les valeurs de comptages.
Sur le tableau de comptage, modifier les noms des lignes pour qu'ils correspondent aux noms des gènes, et les noms de colonnes pour qu'ils correspondent aux noms d'échantillons (`sampleName`).

```
annotgene = comptage[,1:6]
comptage = comptage[,-c(1:6)]
rownames(comptage) = annotgene$Geneid
colnames(comptage) = design$sampleName
```

Question : Pour chaque échantillon, indiquez les statistiques simples sur les valeurs de comptages (moyenne, médiane, min, max, 1er et 3ème quartiles). Quelle fonction permet de retourner très simplement toutes ces statistiques ?

```
summary(comptage)
```

```
##      TNBC1              TNBC2              TNBC3
##  Min.   :     0.0   Min.   :     0.00   Min.   :     0.0
```

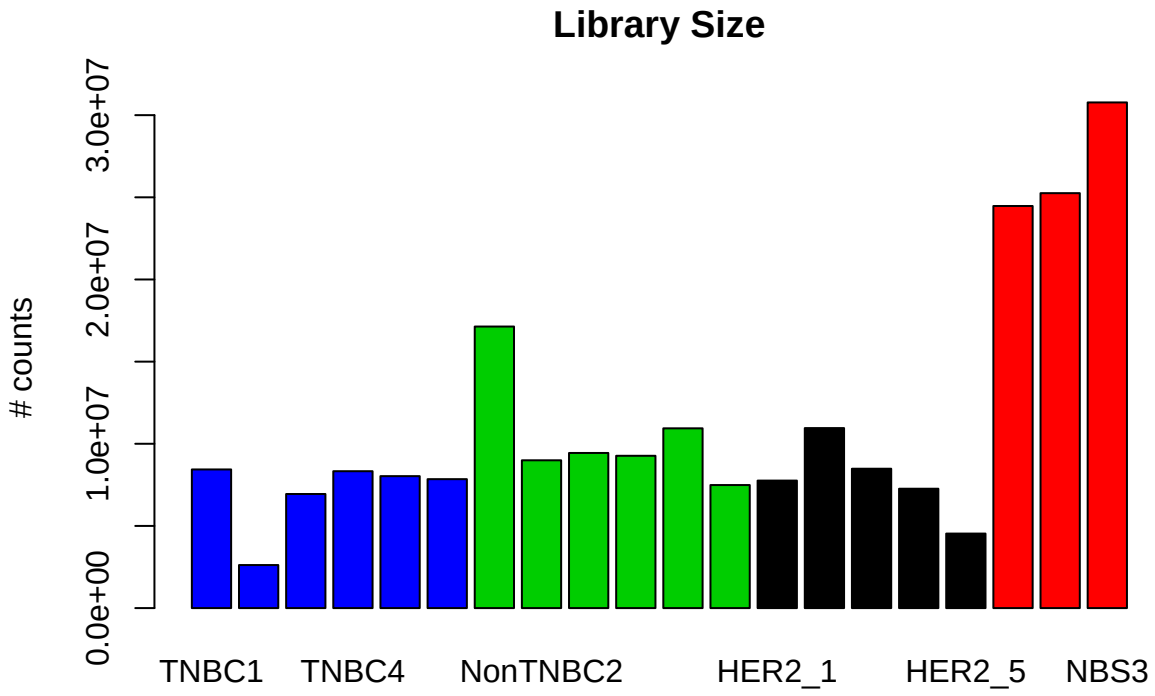
```
##  1st Qu.:     8.0   1st Qu.:     0.00   1st Qu.:    10.0
##  Median :    47.0   Median :     8.00   Median :    48.0
##  Mean   :   319.3   Mean   :    99.11   Mean   :   262.8
##  3rd Qu.:   286.0   3rd Qu.:    77.00   3rd Qu.:   219.0
##  Max.   :115866.0   Max.   :221245.00   Max.   :159453.0
##      TNBC4              TNBC5              TNBC6
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     2.0   1st Qu.:     3.0   1st Qu.:     3.0
##  Median :    31.0   Median :    32.0   Median :    30.0
##  Mean   :   315.2   Mean   :   303.9   Mean   :   296.9
##  3rd Qu.:   253.0   3rd Qu.:   242.0   3rd Qu.:   234.5
##  Max.   :512073.0   Max.   :561500.0   Max.   :255265.0
##     NonTNBC1           NonTNBC2           NonTNBC3
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     3.0   1st Qu.:     4.0   1st Qu.:     4.0
##  Median :    51.0   Median :    37.0   Median :    40.0
##  Mean   :   648.5   Mean   :   340.4   Mean   :   357.3
##  3rd Qu.:   529.0   3rd Qu.:   286.0   3rd Qu.:   305.0
##  Max.   :364193.0   Max.   :435453.0   Max.   :382686.0
##     NonTNBC4           NonTNBC5           NonTNBC6            HER2_1
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     4.0   1st Qu.:     5   1st Qu.:     2.0   1st Qu.:     1.0
##  Median :    40.0   Median :    46   Median :    28.0   Median :    32.0
##  Mean   :   350.7   Mean   :   414   Mean   :   283.4   Mean   :   293.6
##  3rd Qu.:   274.0   3rd Qu.:   339   3rd Qu.:   235.0   3rd Qu.:   236.0
##  Max.   :368364.0   Max.   :575036   Max.   :537307.0   Max.   :353192.0
##      HER2_2             HER2_3             HER2_4             HER2_5
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     1.0   1st Qu.:     1   1st Qu.:     1.0   1st Qu.:     1.0
##  Median :    34.0   Median :    29   Median :    26.0   Median :    17.0
##  Mean   :   414.6   Mean   :   321   Mean   :   274.9   Mean   :   171.6
##  3rd Qu.:   329.5   3rd Qu.:   250   3rd Qu.:   204.0   3rd Qu.:   140.0
##  Max.   :451252.0   Max.   :449500   Max.   :402736.0   Max.   :133319.0
##       NBS1               NBS2               NBS3
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:     1.0   1st Qu.:     1.0   1st Qu.:     1
##  Median :    61.0   Median :    66.0   Median :    66
##  Mean   :   926.1   Mean   :   955.7   Mean   :  1165
##  3rd Qu.:   591.0   3rd Qu.:   662.5   3rd Qu.:   735
##  Max.   :302434.0   Max.   :310990.0   Max.   :219276
```

3. Normalisation

Un biais important en RNA-seq est la profondeur de séquençage de chaque échantillon, aussi appelée taille de la librairie.

> **Question** : Représentez en barplot la taille de librairie de chaque échantillon. Adaptez la couleur des barplot en fonction de la `condition` (1 couleur par `condition`). Existe-t-il un biais dans cette expérience ? Lequel ?

```
barplot(colSums(comptage), main="Library Size", ylab = "# counts", col = as.numeric(design$condition))
```



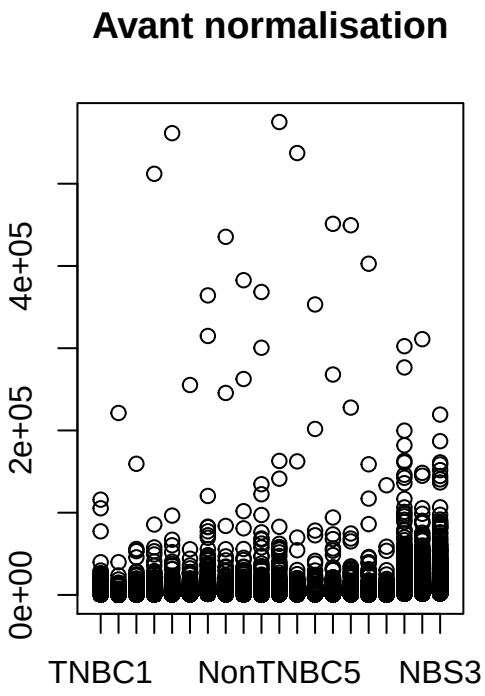
Pour corriger ce biais, nous allons normaliser les données. La taille de la librairie est souvent appelée, en RNA-seq, le *sizefactor*. Si les comptages des gènes non différentiellement exprimés sont, en moyenne, 2 fois plus grands dans un échantillon que dans un autre (car la librairie été séquencée 2 fois plus profond), le *sizefactor* du premier échantillon devrait être 2 fois plus grand que l'autre échantillon.

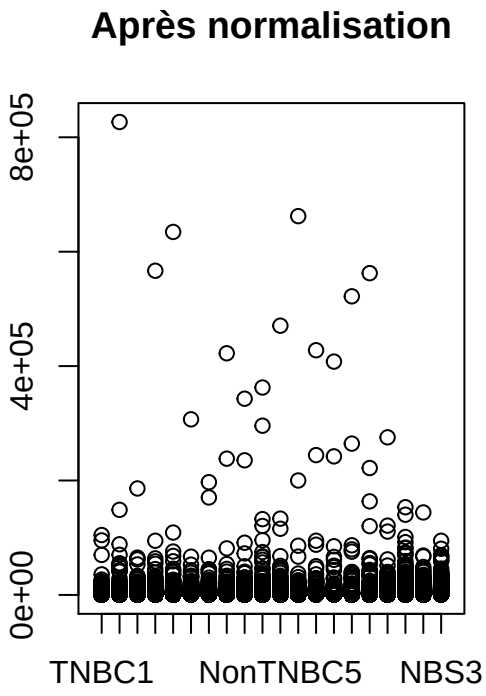
Question : Chargez en mémoire le fichier de comptages normalisés `counts_normalized.txt`.

```
normcount = read.table("counts_normalized.txt", sep="\t", header=TRUE)
```

Question : Afin d'observer l'effet de la normalisation sur les données, représenter sous forme de boxplot les données avant et après normalisation.
Représentez ces 2 graphiques sur une même fenêtre.

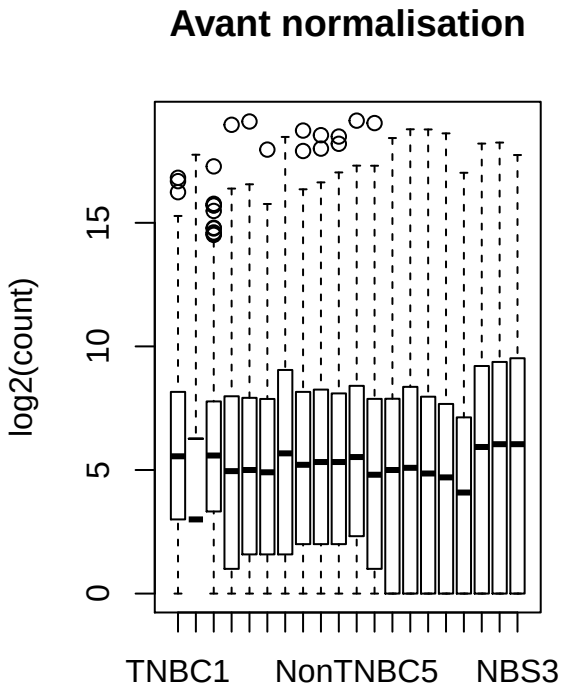
```
par(mfrow=c(1,2))
boxplot(comptage, main="Avant normalisation")
boxplot(normcount, main="Après normalisation")
```

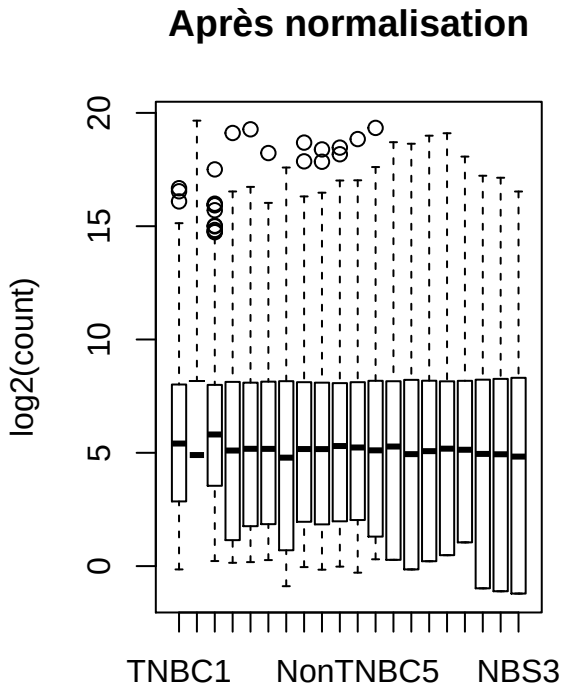




Question : Afin d'avoir une meilleure représentation des données, transformez les matrices de comptages non normalisés et normalisés en log2 (fonction `log2`). Représentez une nouvelle fois les 2 boxplots, après transformation des valeurs en log2.

```
comptagelog = log2(comptage)
normcountlog = log2(normcount)
par(mfrow=c(1,2))
boxplot(comptagelog, main="Avant normalisation", ylab = "log2(count)")
boxplot(normcountlog, main="Après normalisation", ylab = "log2(count)")
```

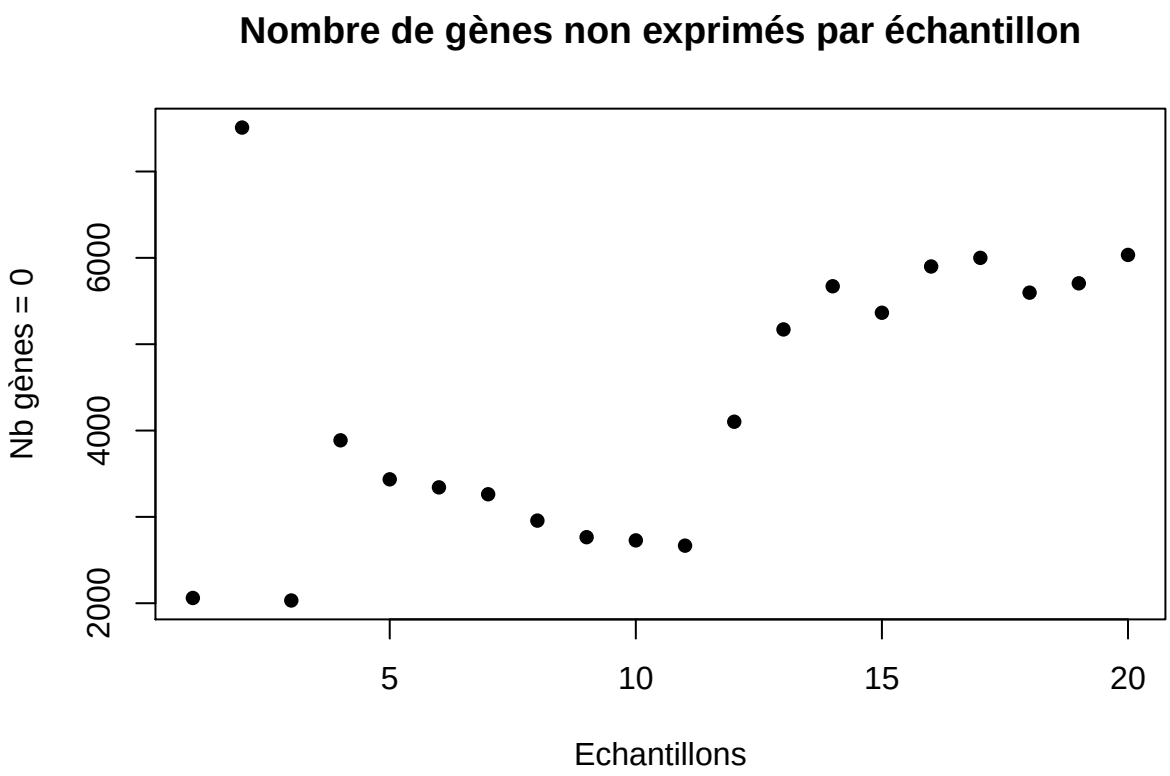




En RNA-seq, il est courant d'avoir des valeurs de comptages à 0, ce qui correspond à des gènes non exprimés.

> **Question** : A partir de la matrice de comptage normalisés, affichez un graphique représentant le nombre de gènes ayant des comptages nuls en fonction des échantillons. Que remarquez-vous ?

```
nbZero = apply(normcount,2,function(r){length(which(r == 0)) })
plot(nbZero, main="Nombre de gènes non exprimés par échantillon", xlab = "Echantillons", ylab="Nb gènes
```



> **Question** : Combien de gènes ne sont jamais exprimés chez tous les échantillons ? Créez une nouvelle matrice ne contenant pas ces gènes. Affichez les dimensions de cette nouvelle matrice. Transformez cette matrice en log2. Quelle est la valeur minimale par échantillon ? Pourquoi ? Afin d'éviter cette valeur, gênante pour la suite des analyses, nous allons ajouter un pseudocount de 1 avant de passer en log2. Crée cette nouvelle matrice.

```
normcount0 = normcount[ rowSums(normcount) > 0, ]
dim(normcount0)
```

```
## [1] 25504    20
```

```
normcount0log = log2(normcount0)
normcount0log = log2(normcount0 + 1 )
```

Pour la suite du TP, nous travaillerons uniquement sur cette nouvelle matrice de comptages, c'est à dire la matrice de comptages normalisés, en log2, sans les gènes non exprimés chez tous les échantillons.

4. Analyse non supervisée

Nous allons visualiser la proximité relative des observations, grâce à une **Analyse en Composantes Principales**. Il s'agit d'une méthode d'analyse multivariée par réduction de dimension. Les composantes

principales sont des combinaisons linéaires des variables. Elles ont pour contraintes de maximiser la variance entre les observations et d'être orthogonales entre elles. Le nombre de composantes est égal au rang de la matrice des données. On utilise la fonction `prcomp` de `R base`.

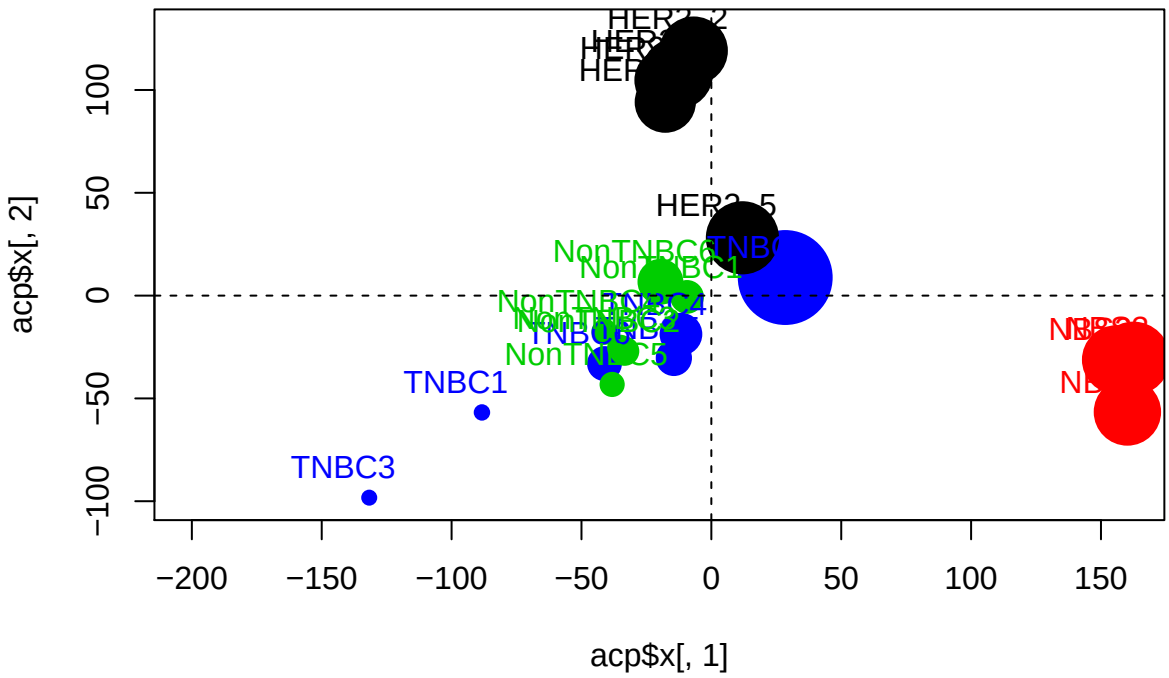
Question :

- Transposez votre matice de comptage grâce à la fonction `t()`.

- Calculez les composantes grâce à la fonction `prcomp` avec les options `scale=TRUE` et `center=TRUE`. Combien y a-t-il de composantes ?
- Représentez graphiquement les observations, c'est à dire les échantillons, en fonction des deux premières composantes, colorez les points en fonction de la colonne "condition" et changez la taille des points (paramètre `cex` de la fonction `plot`) en fonction du nombre de gènes non exprimés du tableau de données.
- Ajoutez le noms des échantillons sur le graphique généré (fonction `text()`)

- Ajoutez les lignes x=0 et y=0 en pointillé.
- Interprétez cette ACP.

```
acp = prcomp(t(normcount0log),center = TRUE, scale = TRUE)
nbZero = apply(normcount0log,2,function(r){length(which(r == 0)) })
plot(acp$x[,1], acp$x[,2], pch=16, col=as.numeric(design$condition), cex=nbZero/1000, ylim=c(-100,130),
text(acp$x[,1]-10, acp$x[,2]+15, labels = design$sampleName, col=as.numeric(design$condition))
abline(v=0, h=0, lty=2)
```



Il est possible de visualiser la proximité relative des observations, grâce à une autre méthode : **le clustering**. Cette methode consiste à calculer une distance entre les profils transcriptomique, puis à les regrouper de proche en proche. Il existe de nombreuses méthodes pour calculer une distance entre deux profils (distance Euclidienne, disante de Manahatan . . .) et de nombreuses méthode pour agréger les profils les plus semblables entre eux.

Question :

- Chargez les library `pheatmap` et `RColorBrewer` si ce n'est pas déjà fait.
- Appliquez la fonction `dist` (avec les paramètres par défaut) à la matrice de comptage transposée. Quelle est la classe de l'objet généré ?
- Créer un vector de couleurs de dégradé de bleu grâce à la commande suivante : `colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)`.
- Générez une heatmap sample-to-sample grâce à la fonction `pheatmap`, et y ajouter une ligne d'annotations correspondant à l'annotation `condition`.
- Interprétez ce clustering.

```
library(pheatmap)
library(RColorBrewer)
sampledist = dist(t(normcount0log))
sampledistMatrix = as.matrix(sampledist)
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
annot = as.data.frame(design$condition)
rownames(annot) = design$sampleName
colnames(annot) = "condition"
pheatmap(sampledistMatrix,col=colors, annotation_col = annot)
```