Segmentation/clustering on normalized data

```
# Library and paths
rm(list=ls())
library(tidyverse)
library(dplyr)
library(ggplot2)
library(data.table)
library(factoextra)
library(gfpop)
library(mclust)
```

Data description, filtering and normalization

```
Rep <-"/Users/lebarbier/Desktop/Projets/IFCAM-genomics/Programs/Single-Cell"
setwd(Rep)
dataDir <- "../../Data/Single-Cell/"
RawCountsFile <- "T17225-counts.tsv"
```

Raw count data

```
RawCounts <- as.data.frame(fread(paste0(dataDir, RawCountsFile)))
rownames(RawCounts) <- RawCounts[, 1];RawCounts <- RawCounts[, -1]
RawCounts[1:5, 1:5]
```

```
                AAACCTGAGAGAGCTC AAACCTGAGAGCTTCT AAACCTGAGCTGCAAG
ENSG00000000003                0                0                0
ENSG00000000419                0                0                0
ENSG00000000457                0                0                0
ENSG00000000460                0                0                0
ENSG00000000938                0                0                0
                AAACCTGAGGCCCTTG AAACCTGAGGCTAGCA
ENSG00000000003                0                0
ENSG00000000419                1                0
ENSG00000000457                0                0
ENSG00000000460                0                0
ENSG00000000938                0                0
```

```
NbCell <-nrow(RawCounts)
NbGene <- ncol(RawCounts)
```

Il y a 27760 cellules et 7437 gènes.

Filtering

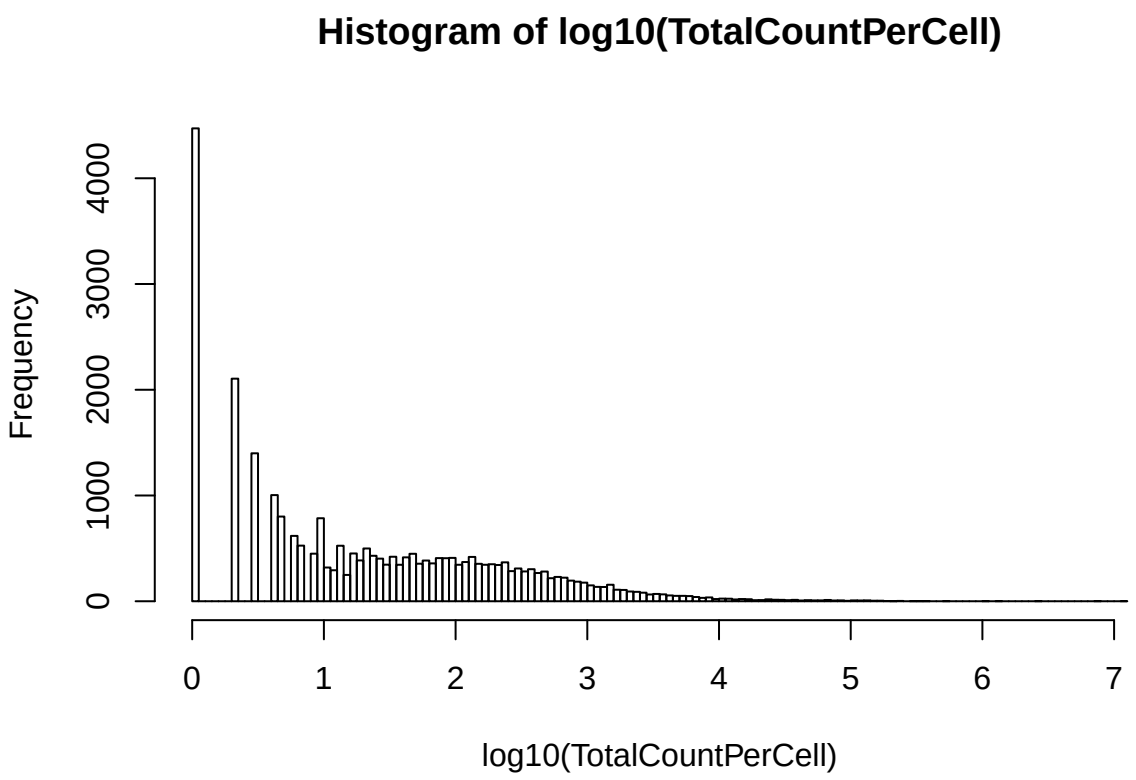
On the cells and first normalization

Nombre total par cellule

```
TotalCountPerCell <- RawCounts %>% rowSums
TotalCountPerCell %>% as.tibble() %>% summarise_all(.,funs(n(),min,mean,median,max))
```

```
# A tibble: 1 x 5
      n   min  mean median      max
  <int> <dbl> <dbl>  <dbl>    <dbl>
1 27760     1 1647.     17 11253007
```

```
hist(log10(TotalCountPerCell), breaks=sqrt(NbCell))
```



Expression moyenne par cellules.

```
ThresholdMeanExp <- 1
NumberCellKeep <-which(rowMeans(RawCounts)>=ThresholdMeanExp)
RawCountsFilterCell <- RawCounts[NumberCellKeep,]
NbCellAfterRmv <- nrow(RawCountsFilterCell)
```

On enleve les cellules qui ont une expression moyenne inférieure à 1. Après ce filtrage, il reste 385 cellules.

Normalisation w.r.t. nombre total par cellule

```
NormCounts <- RawCountsFilterCell/TotalCountPerCell[NumberCellKeep]
```

On the genes

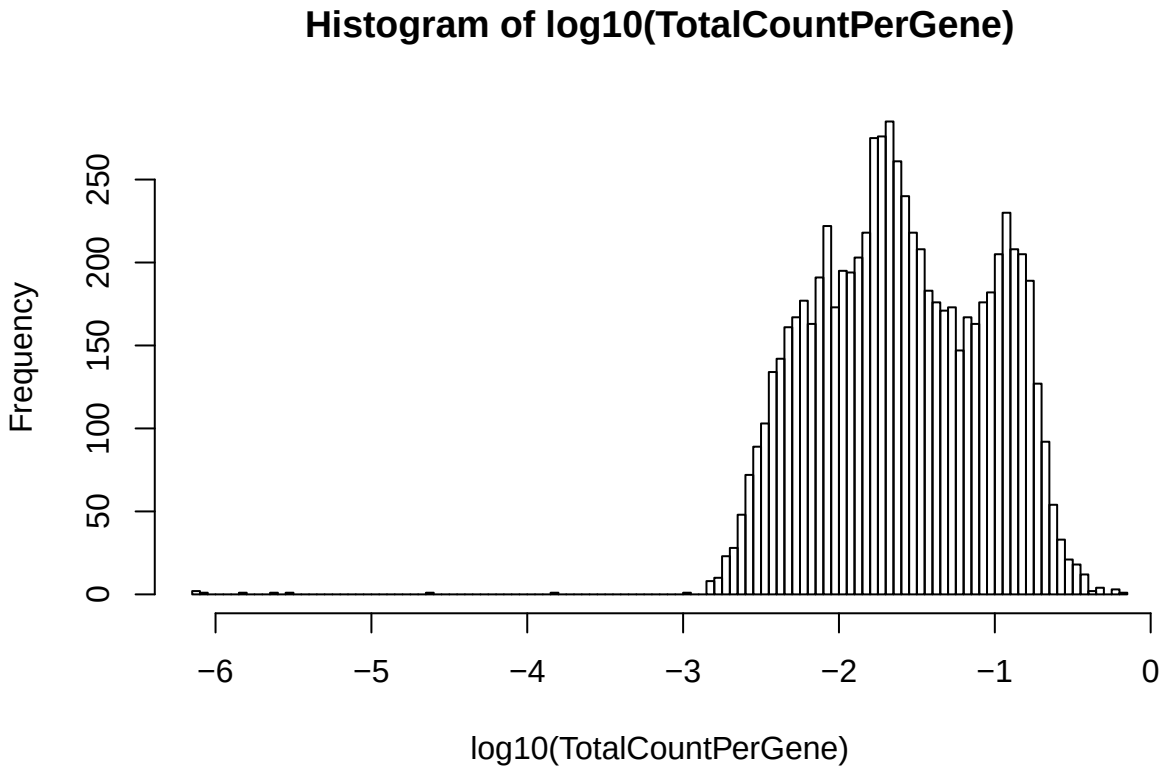
Total par gene

```
TotalCountPerGene <- NormCounts %>% colSums
TotalCountPerGene %>% as.tibble %>% summarise_all(.,funs(n(),min,mean,median,max))
```

```
# A tibble: 1 x 5
      n   min   mean median   max
```

```
  <int> <dbl>  <dbl>  <dbl> <dbl>
1  7437     0 0.0518 0.0239 0.680
```

```
hist(log10(TotalCountPerGene), breaks=sqrt(NbGene))
```

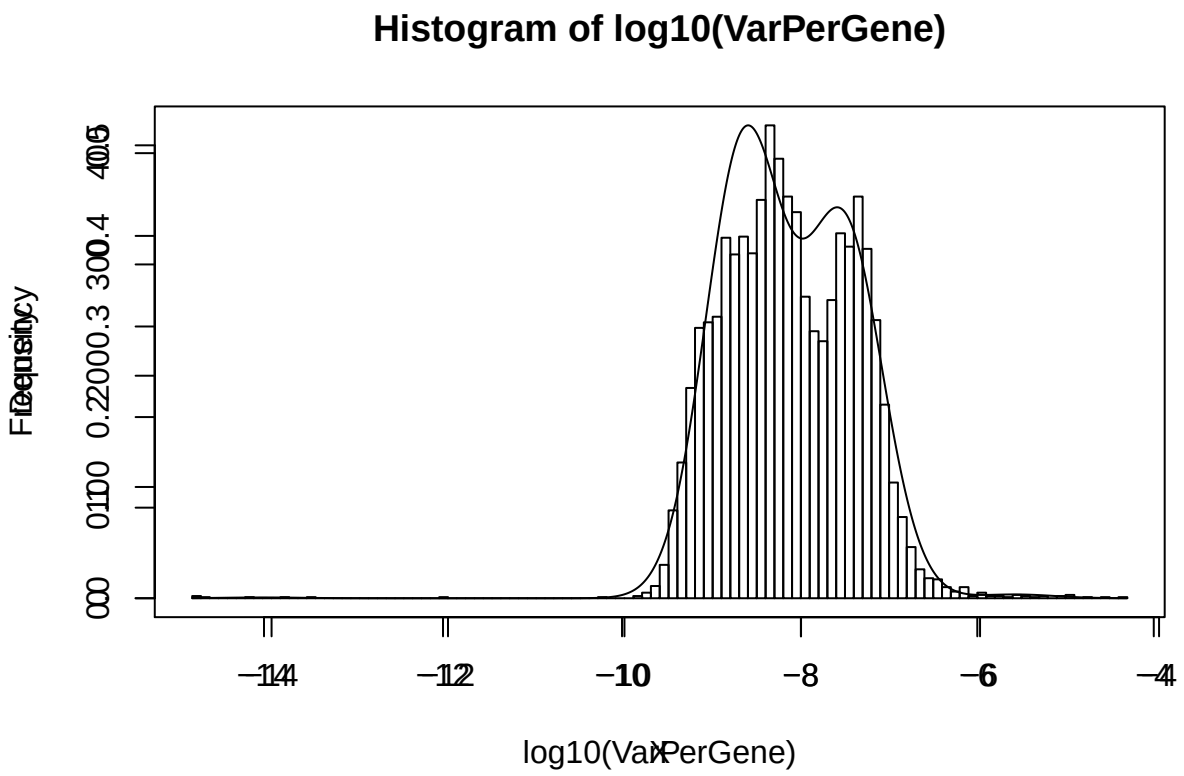


On enlève les gènes d'expression nulle

```
NormCounts <- NormCounts[,-which(TotalCountPerGene==0)]
```

Variance par gène. Je retire les gènes de trop petite variance.

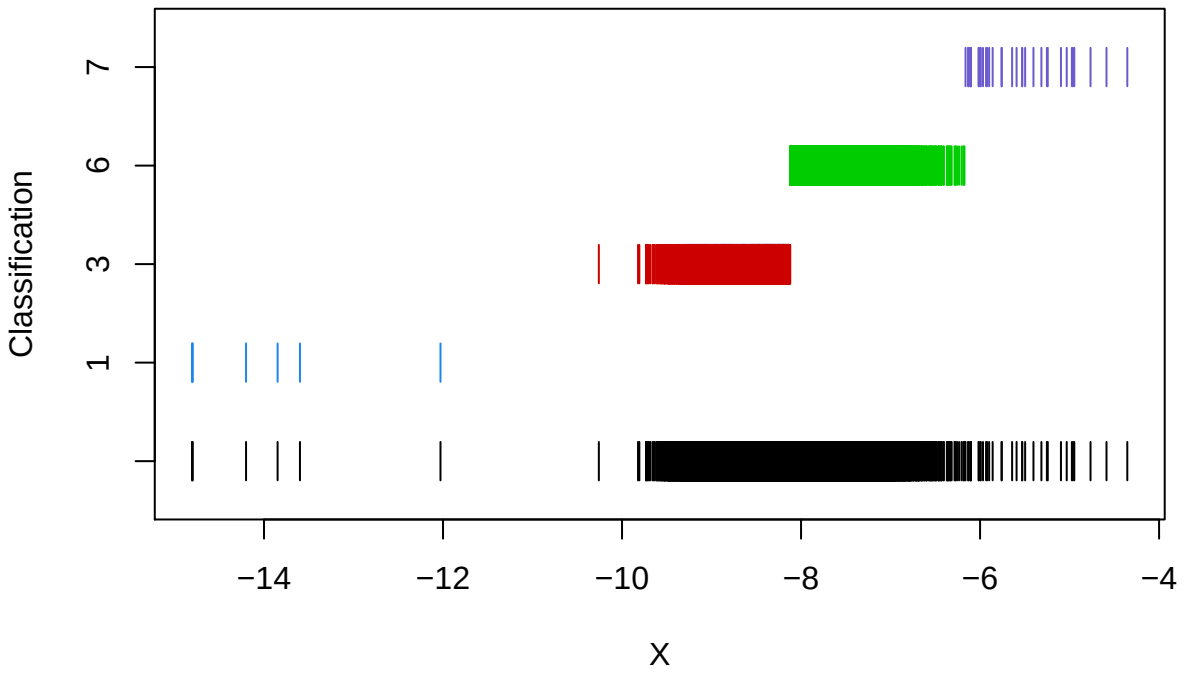
```
VarPerGene <- colMeans(NormCounts**2)-colMeans(NormCounts)**2
hist(log10(VarPerGene),100)

X=log10(VarPerGene)
group.VarPerGene <- Mclust(X,modelNames = "E")
hist(log10(VarPerGene),100)
par(new=T)
plot(group.VarPerGene,what="density")
```




```
table(group.VarPerGene$classification)
```

```
   1    3    6    7
   7 3925 3468   35
```

```
plot(group.VarPerGene, what = "classification")
```



```
NormCountsF <- NormCounts[,-which(group.VarPerGene$classification<=2)]
NbGeneAfterRmv <- ncol(NormCountsF)
```

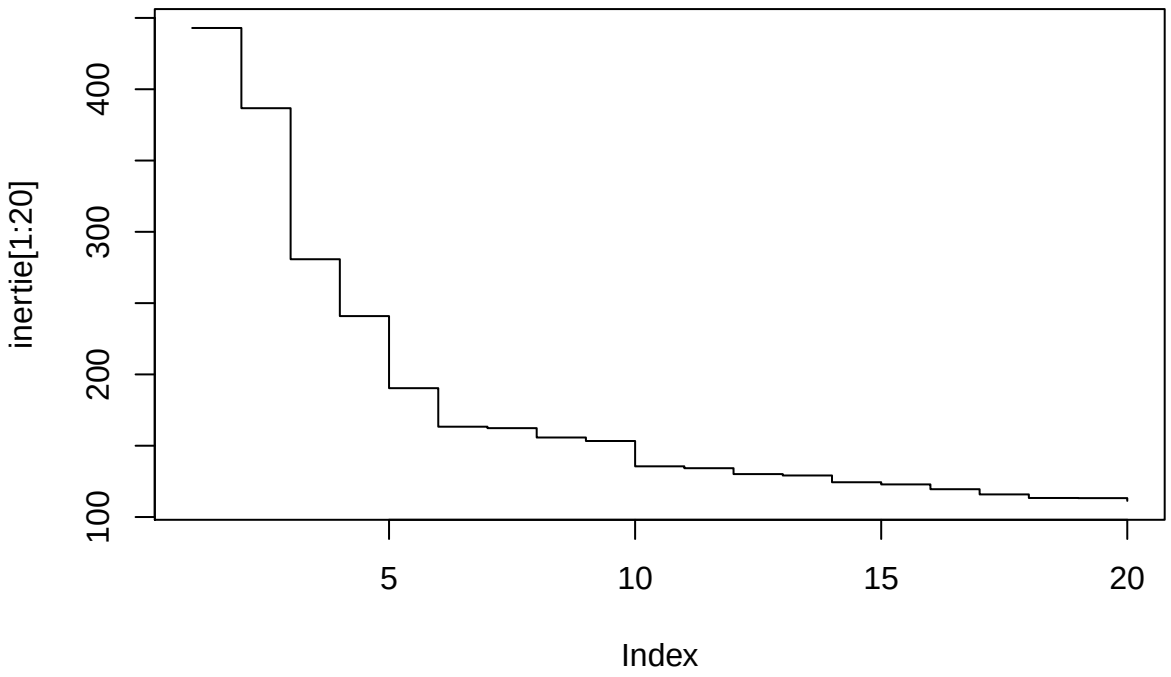
Il reste 7428.

Compute scaled expression profiles for cells

```
NormCountsF.scaled <- t(NormCountsF) %>% scale %>% t()
p <- NbGeneAfterRmv
n <- NbCellAfterRmv
```

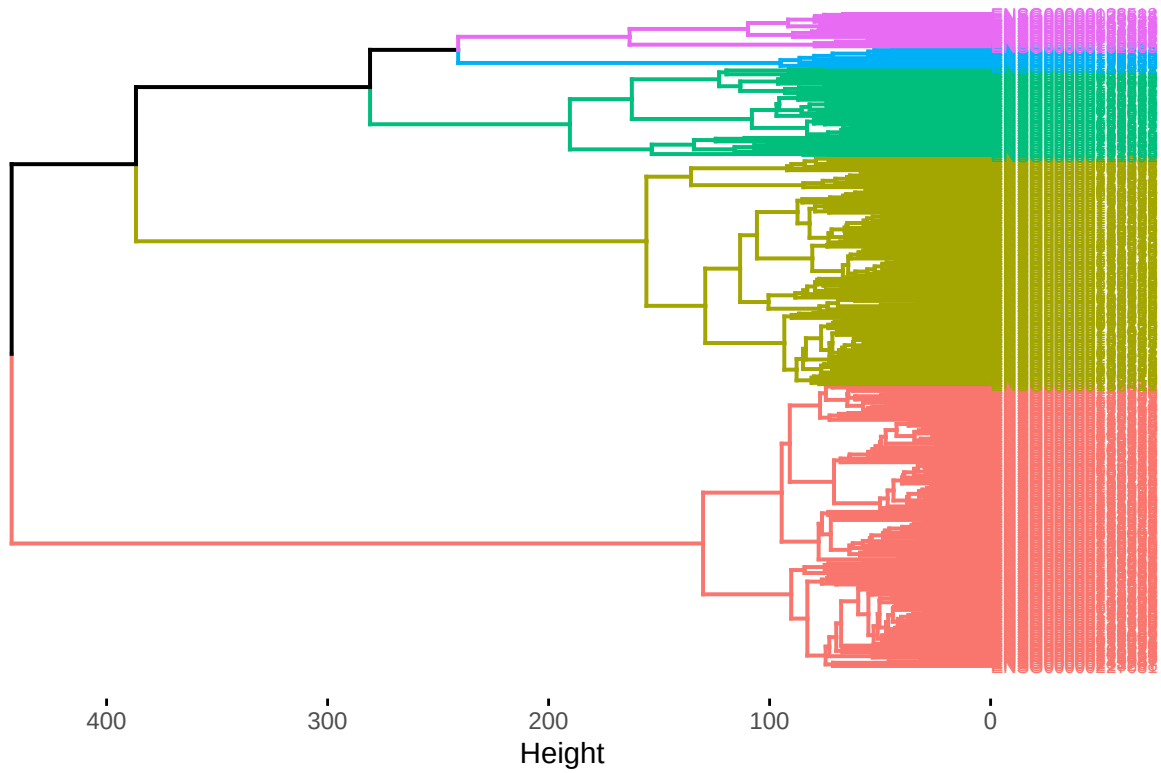
Cell clustering on the normalized data

```
Dist.Cell <- dist(x = NormCountsF.scaled)
HclustCell <- hclust(d = Dist.Cell,method = "ward.D2")
inertie <- sort(HclustCell$height, decreasing = TRUE)
plot(inertie[1:20], type = "s")
```

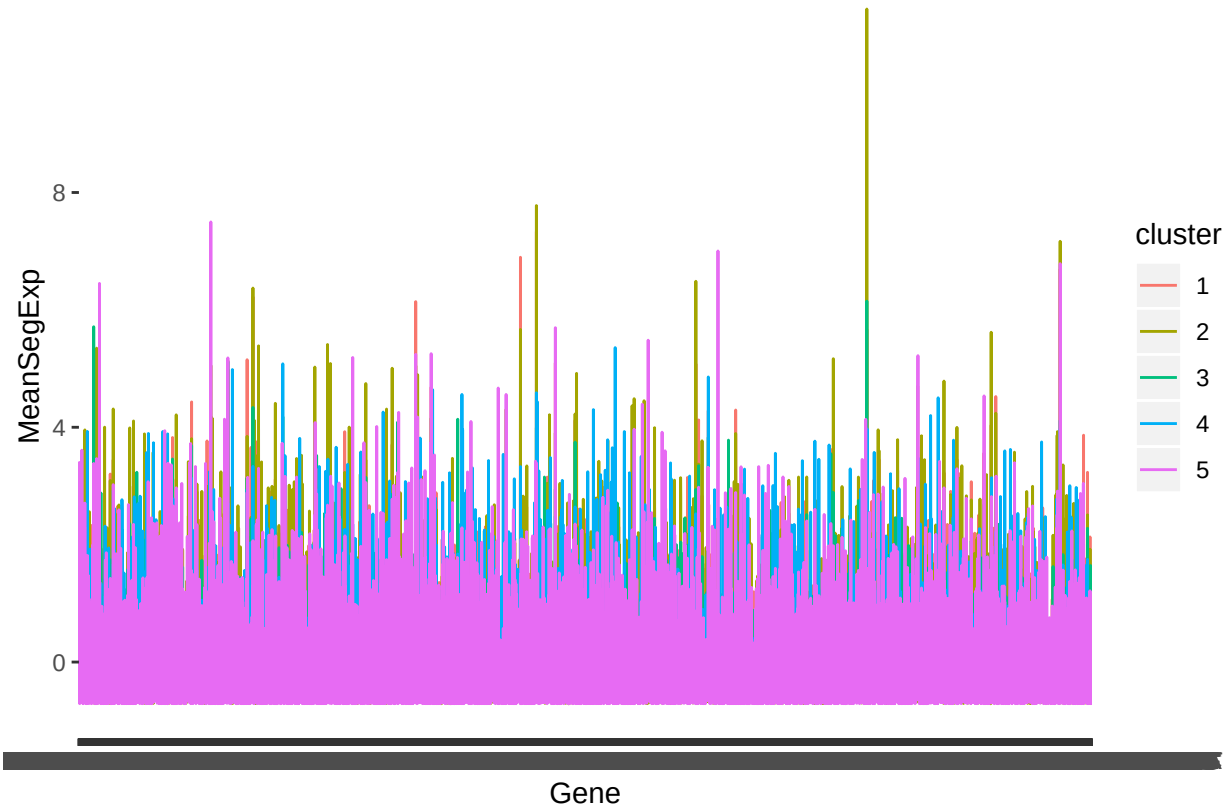



```
NbClust=5
#Dendrogramme
fviz_dend(HclustCell,horiz = TRUE, cex = 0.5, k = NbClust, color_labels_by_k = TRUE)
```

Cluster Dendrogram




```
Cluster.Cell <- cutree(HclustCell, k = NbClust)
MeanProfileByClust <-
  NormCountsF.scaled %>% as.tibble %>% mutate(cluster = as.factor(Cluster.Cell)) %>% group_by(cluster) %

# Mean expression per group
MeanProfileByClust %>%
  ggplot(aes(x = Gene, y = MeanSegExp, group = cluster)) +
  geom_line(aes(color = cluster))
```




```
# Proportion de profils dans chaque groupe
PropClust.Cell <- Cluster.Cell %>%tibble %>% setnames("cluster") %>%  group_by(cluster) %>% summarise(N
PropClust.Cell
```

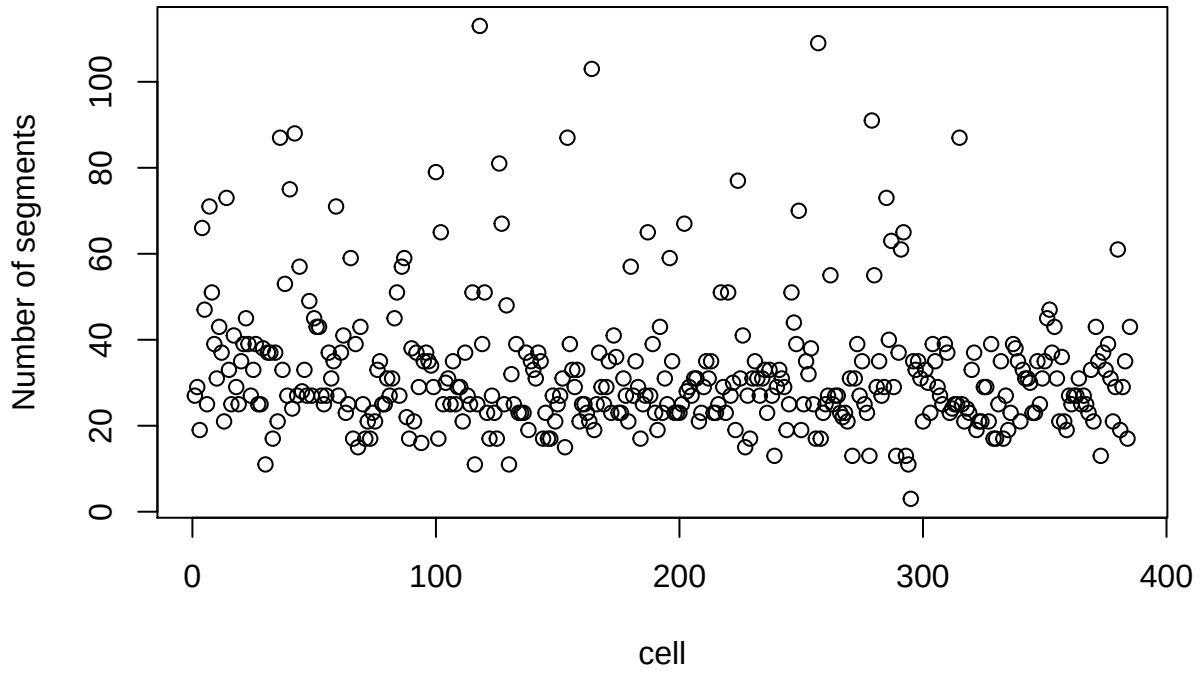
```
# A tibble: 5 x 2
  cluster NbCell
    <int>  <int>
1       1    135
2       2    166
3       3     52
4       4     20
5       5     12
```

Segmentation/Clustering

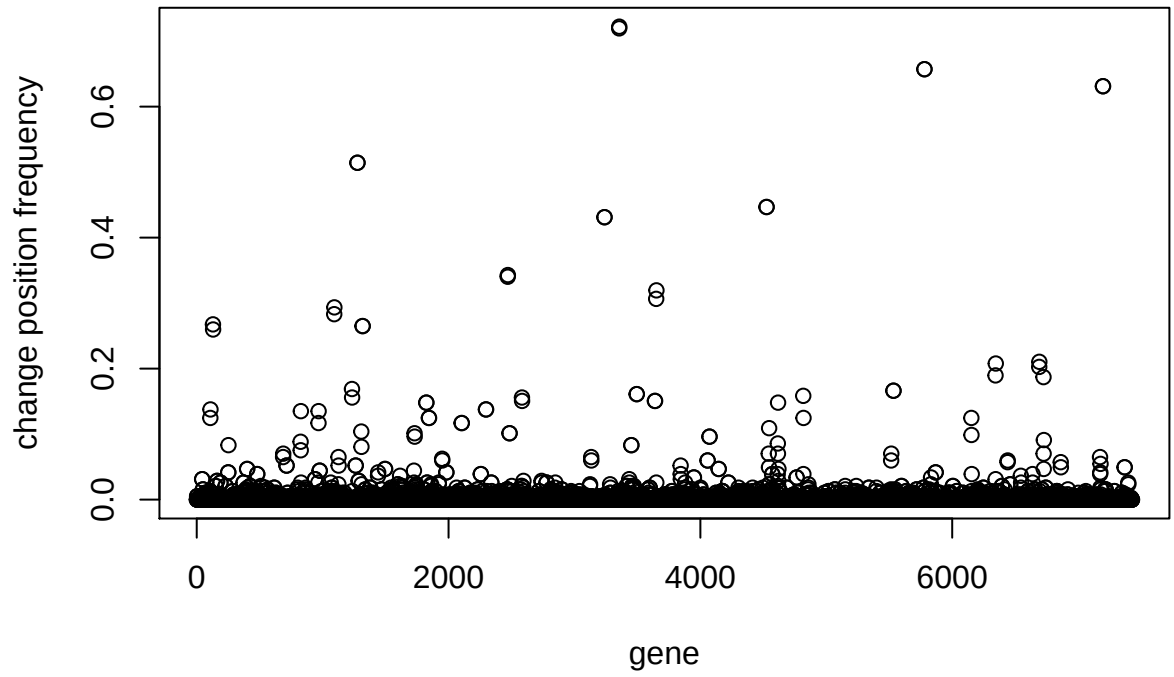
Segmentation with fgpop (faster than Segmentor3Isback) and a penalty function 2*log(p)

```
Seg <- function(data,i){
  signal <- as.numeric(data[i,])
  n <- ncol(data)
  myGraphStd <- graph(penalty = 2*log(n), type = "std")
  ResSeg=gfpop(data =  signal, mygraph = myGraphStd, type = "mean")
  rupt=ResSeg$changepoints
  Kselect<- length(rupt)
```

```
  rupt.bin <- rep(0,n)
  rupt.bin[rupt] <- 1
  rupt.bin[n] <- 0
  y.pred.per.segment <-ResSeg$parameters
  y.pred <- rep(y.pred.per.segment,diff(c(0,rupt)))
  return(list(Kselect=Kselect,rupt=rupt,rupt.bin=rupt.bin,y.pred=y.pred))
}


R <- purrr::map(1:n,~Seg(NormCountsF.scaled,.x) )

CellKselect <-  map_dbl(R,~ .x$Kselect)
CellRupt.mean.pos <-R %>% purrr::map(.,"rupt.bin") %>% do.call(rbind,.) %>% colMeans(.)
CellPred <- R %>% purrr::map(.,"y.pred") %>% do.call(rbind,.) %>% as.data.frame()
colnames(CellPred) <- colnames(NormCountsF.scaled)
rownames(CellPred) <- rownames(NormCountsF.scaled)

#Graphes
#Nombre de segments par profil
plot(1:n,CellKselect,ylab="Number of segments",xlab="cell")
```

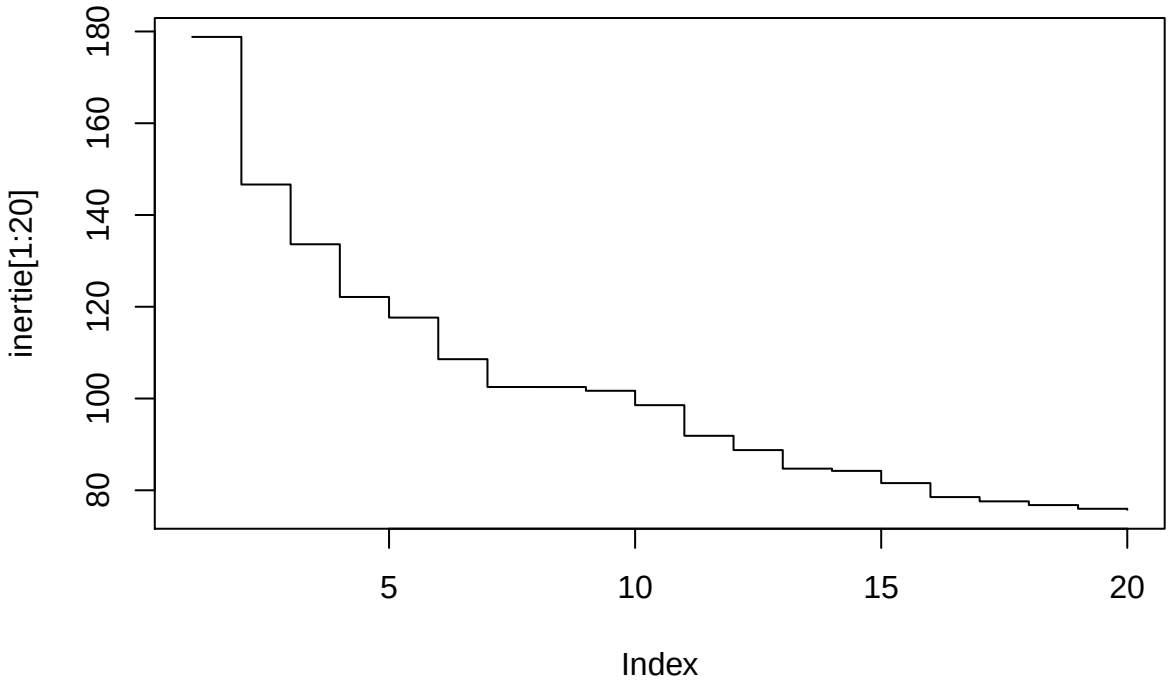



```
#fréquence des ruptures
plot(1:p,CellRupt.mean.pos,ylab="change position frequency",xlab="gene")
```



Clustering

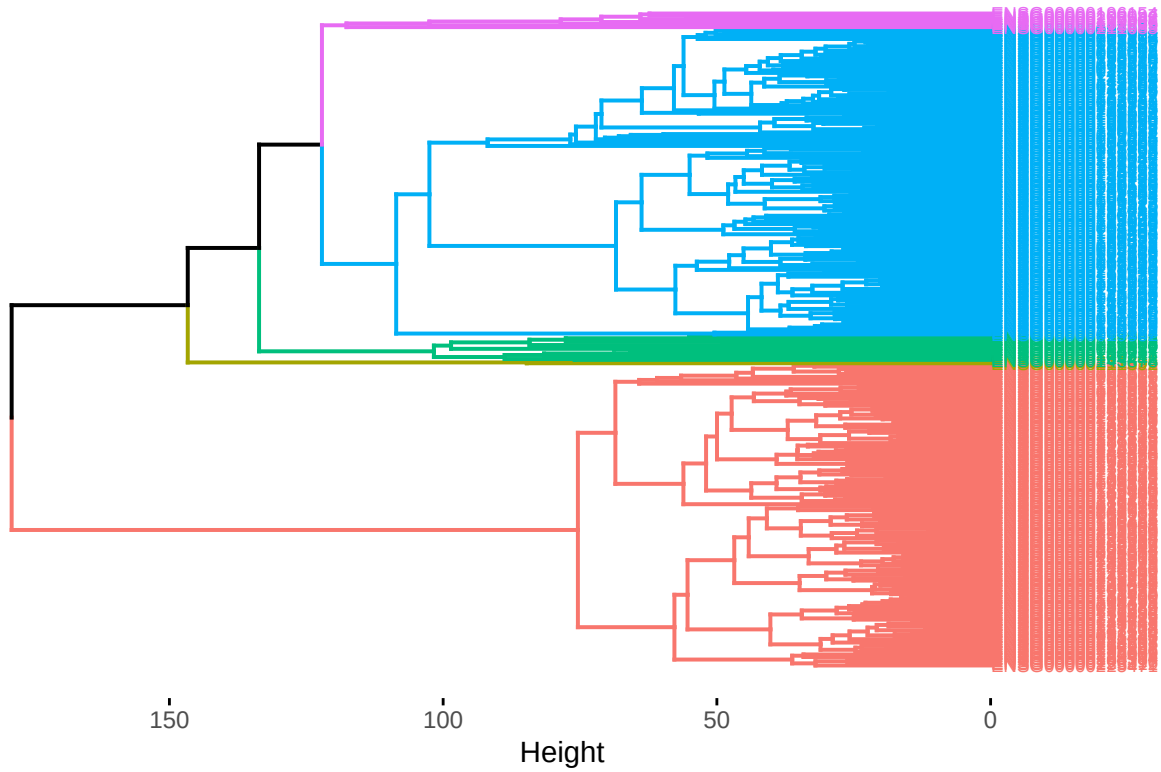
```
Dist.CellSeg <- dist(CellPred)
HclustCellSeg <- hclust(d = Dist.CellSeg,method = "ward.D2")
inertie <- sort(HclustCellSeg$height, decreasing = TRUE)
plot(inertie[1:20], type = "s")
```



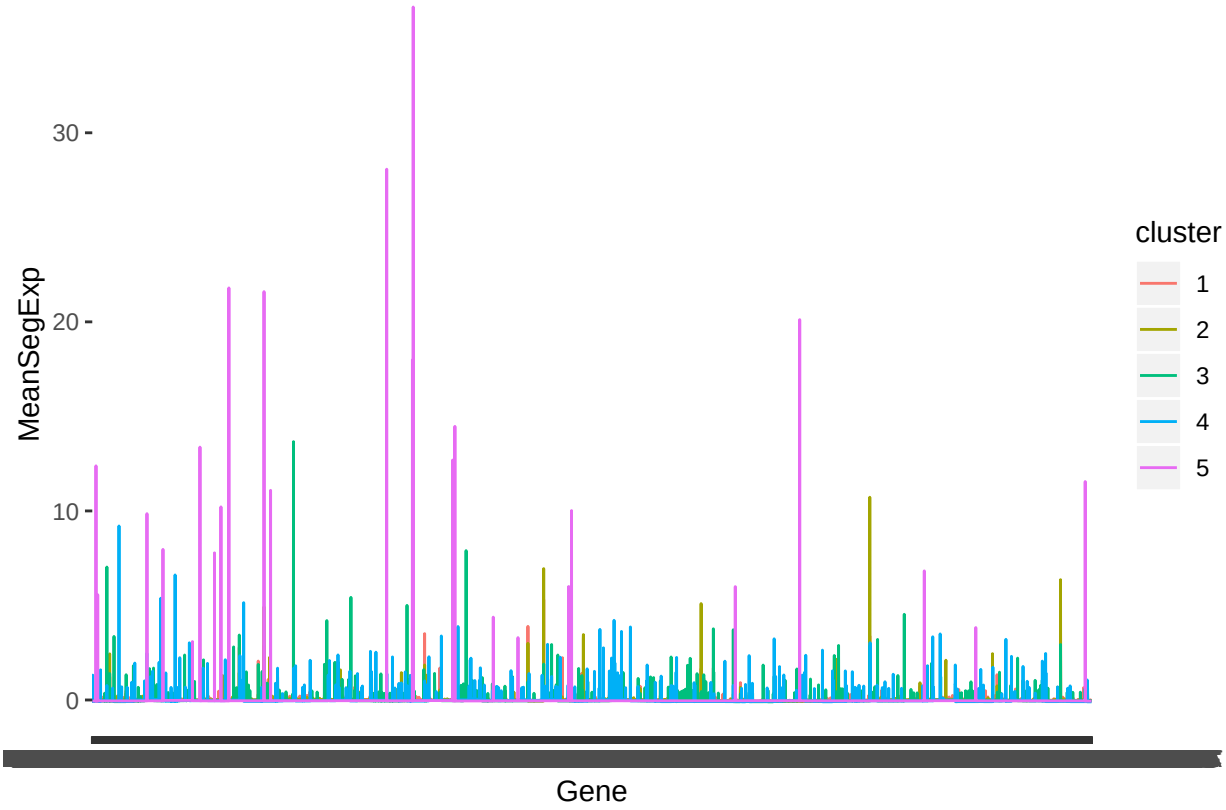

```
NbClustSeg=5
#Dendrogramme
```

```
fviz_dend(HclustCellSeg,horiz = TRUE, cex = 0.5, k = NbClustSeg, color_labels_by_k = TRUE)
```

Cluster Dendrogram




```
Cluster.CellSeg <- cutree(HclustCellSeg, k = NbClustSeg)
MeanProfileCellSegByClust <-
  CellPred %>% mutate(cluster = as.factor(Cluster.CellSeg)) %>% group_by(cluster) %>%  summarise_all(fu

# Profils moyens segmentés dans chaque groupe
MeanProfileCellSegByClust %>%
  ggplot(aes(x = Gene, y = MeanSegExp, group = cluster)) +
  geom_line(aes(color = cluster))
```




```
# Proportion de profils dans chaque groupe
PropClust.CellSeg <- Cluster.CellSeg %>%tibble %>% setnames("cluster") %>%  group_by(cluster) %>% summa
PropClust.CellSeg
```

```
# A tibble: 5 x 2
  cluster NbCell
    <int>  <int>
1       1    181
2       2    178
3       3     15
4       4      9
5       5      2
```

```
# Comparaison entre les deux classif
table(Cluster.CellSeg,Cluster.Cell,dnn=c("Seg","WithoutSeg"))
```

```
   WithoutSeg
Seg   1   2   3   4   5
  1 123   7  29  10  12
  2  11 159   7   1   0
  3   1   0  12   2   0
  4   0   0   2   7   0
  5   0   0   2   0   0
```