# Segmentation/clustering on raw data

```
# Library and paths
rm(list=ls())
library(tidyverse)
library(dplyr)
library(ggplot2)
library(Segmentor3IsBack)
library(data.table)
library(factoextra)
Rep <-"/Users/lebarbier/Desktop/Projets/IFCAM-genomics/Programs/Single-Cell"
setwd(Rep)
dataDir <- "../../Data/Single-Cell/"
```

## Data importation

```
tab <- as.data.frame(fread(paste0(dataDir, 'T17225-counts.tsv')))
rownames(tab) <- tab[, 1];tab <- tab[, -1]
tab[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(tab)
NbGene <- ncol(tab)
```

Il y a 27760 cellules et 7437 gènes.

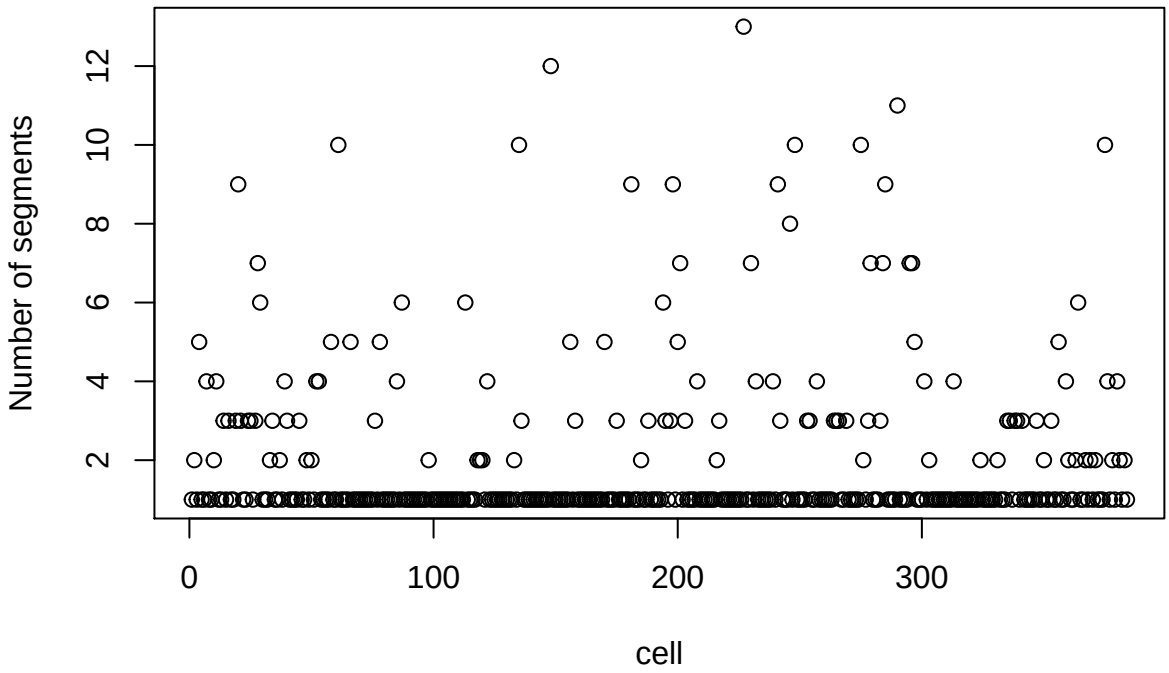
## Raw data and Filtering

```
thres <- 1
data <- tab[which(rowMeans(tab)>=thres), which(colMeans(tab)>=thres)]
data <- data[-which(rownames(data) %in% "ENSG00000211592"),]
n <- nrow(data); p <- ncol(data)
```

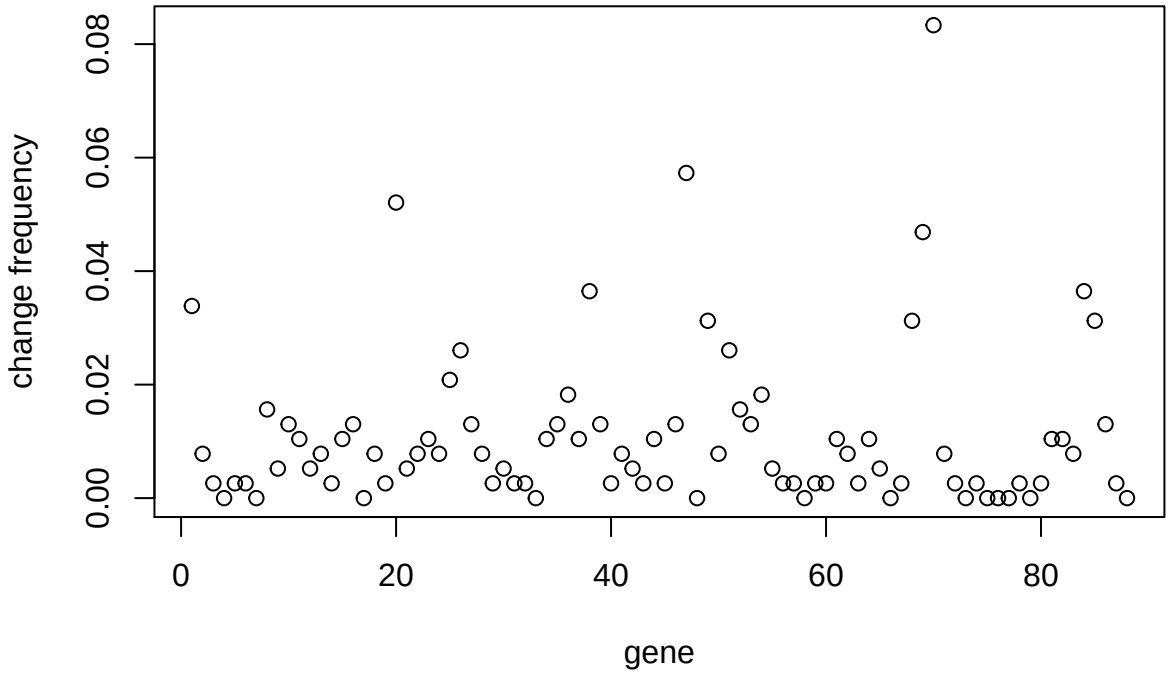
On ne garde que les cellules et les gènes pour lesquels le signal moyen est supérieur à 1. Il reste 384 cellules et 88 gènes.

### Segmentation of the expression profil of each cell

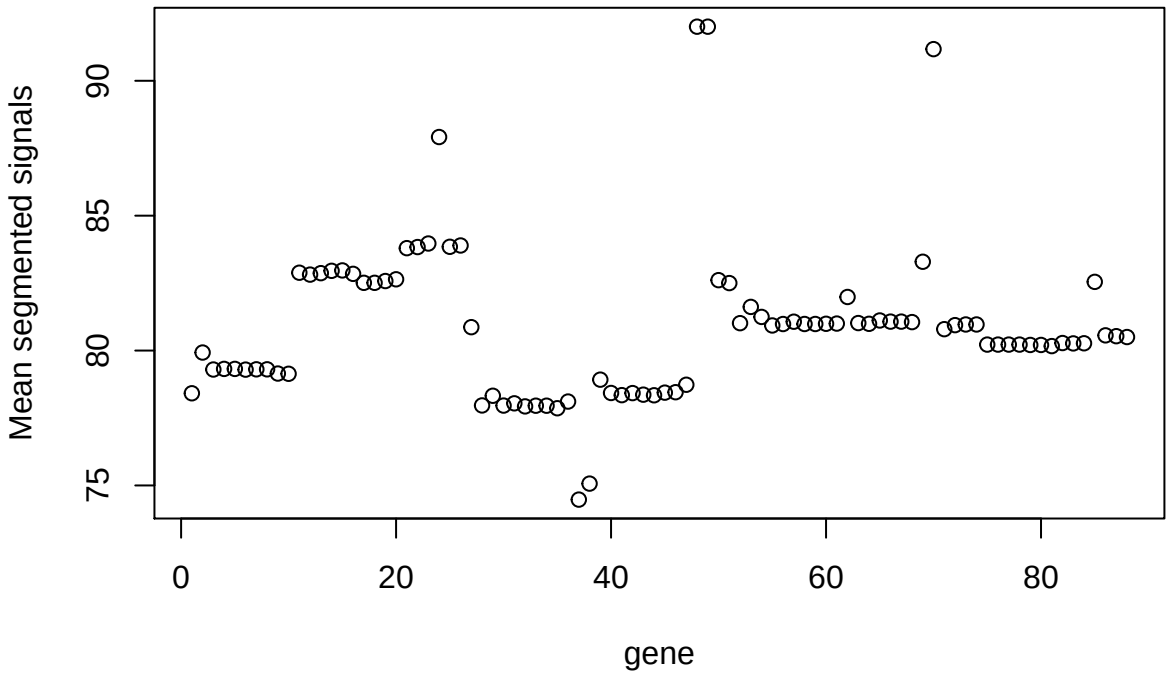
```
Seg <- function(i){
  signal <- as.numeric(data[i,])
  res <- Segmentor(signal, model=3, Kmax=30, keep=TRUE);
  Kselect<-SelectModel(res, penalty='oracle', keep=FALSE)
  rupt <- getBreaks(res)[Kselect,1:Kselect]
  rupt.bin <- rep(0,p)
  rupt.bin[rupt] <- 1
  rupt.bin[p] <- 0
  if (Kselect ==1){
    y.pred.per.segment <- mean(signal)
  } else {
    proba <- getParameters(res)[Kselect,1:Kselect]
    phi <- getOverdispersion(res)
    y.pred.per.segment <-phi*(proba)/(1-proba)
  }
  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))
}

SegFileName <- paste0("CellSeg_thresCellGene",thres,".rds",sep="")
if(!file.exists(SegFileName)){
  R <- 1:n %>% map(Seg)
  saveRDS(R,SegFileName)
} else {
  R <- readRDS(SegFileName)
}

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

#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 frequency",xlab="gene")
```



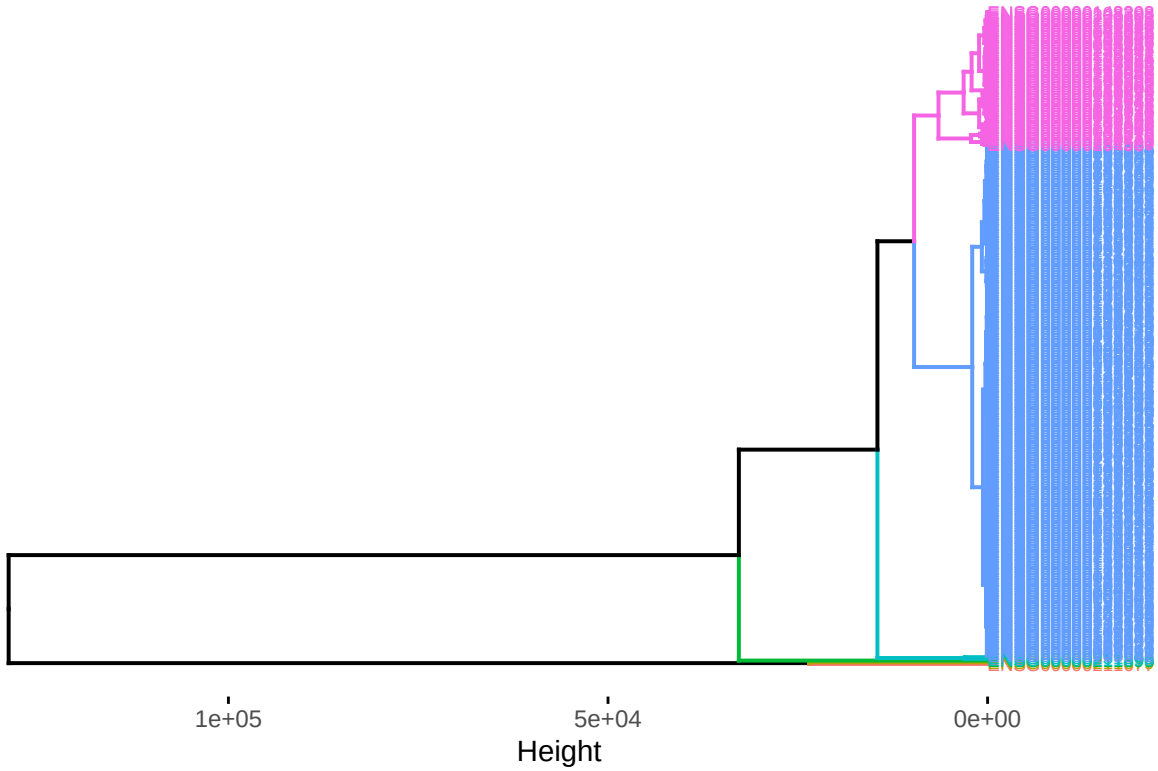

```
#Moyennes des profils segmentés
plot(colMeans(CellPred),ylab="Mean segmented signals",xlab="gene")
```



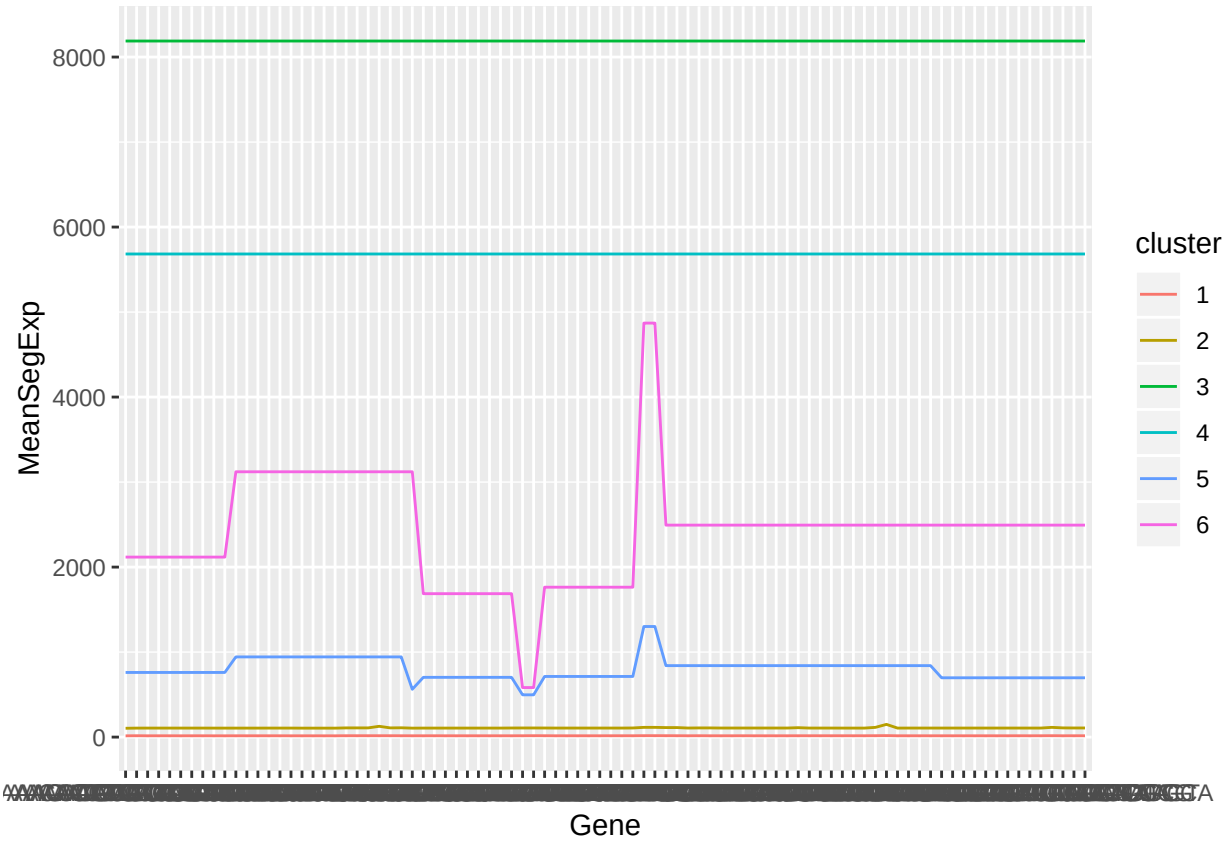

### Clustering of the cells par CAH

```
Dist.CellProf <- dist(x = CellPred)
HclustCellProf <- hclust(d = Dist.CellProf,method = "ward.D2")
NbClust=6
#Dendrogramme
fviz_dend(HclustCellProf,horiz = TRUE, cex = 0.5, k = NbClust, color_labels_by_k = TRUE)
```

Cluster Dendrogram




```
clusterProf <- cutree(HclustCellProf, k = 6)
MeanProfileByClust <-
  CellPred %>% mutate(cluster = as.factor(clusterProf)) %>% group_by(cluster) %>%  summarise_all(funs(m

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



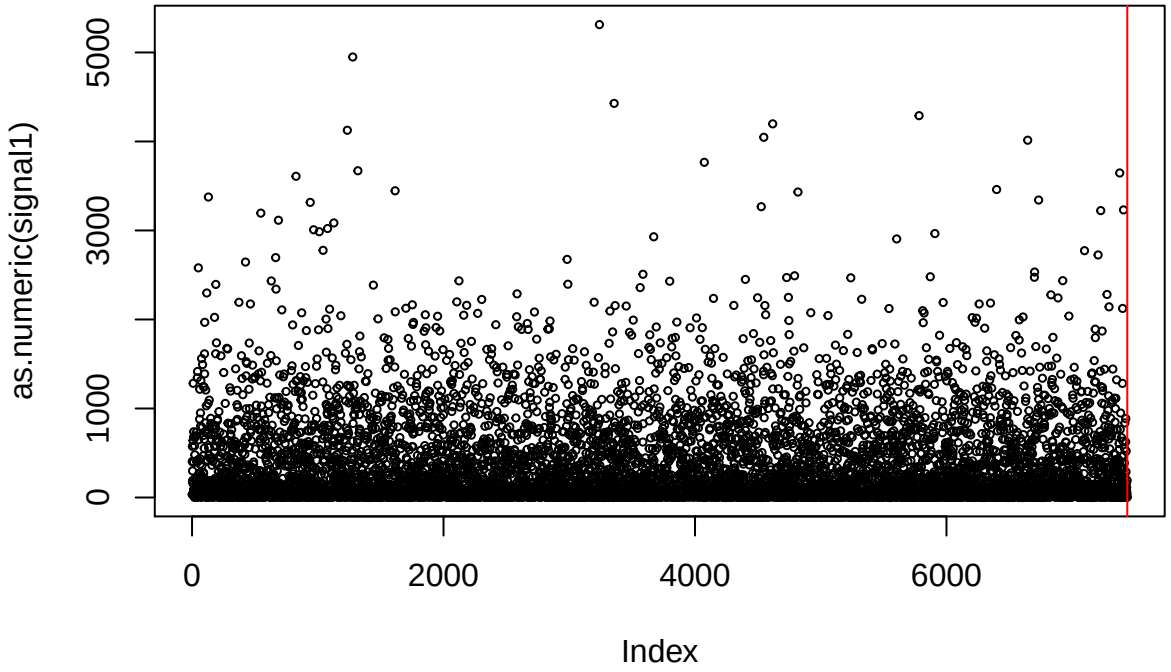

```
# Proportion de profils dans chaque groupe
PropCellClustProf <- clusterProf %>%tibble %>% setnames("cluster") %>%  group_by(cluster) %>% summarise
PropCellClustProf
```

```
# A tibble: 6 x 2
  cluster NbCell
    <int>  <int>
1       1    300
2       2     79
3       3      1
4       4      1
5       5      2
6       6      1
```

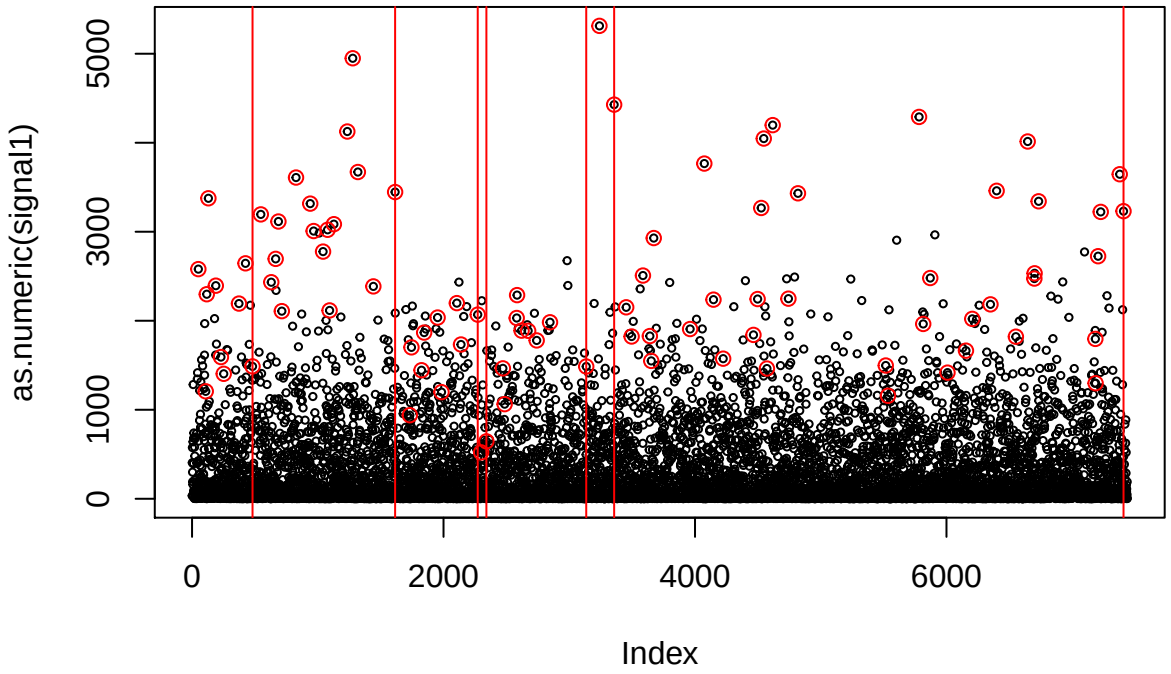
### Signal in cluster 6 (the unique)

```
c=6
signal <- data %>% as.tibble %>% mutate(cluster=clusterProf) %>% filter(cluster==c)
rupt <- R[[which(clusterProf==c)]]$rupt


#Segmentation with all the genes
thres <- 1
data1 <- tab[which(rowMeans(tab)>=thres),]
data1 <- data1[-which(rownames(data1) %in% "ENSG00000211592"),]
KeepGene <- which(colMeans(tab)>=thres)
```

```
signal1 <- data1 %>% as.tibble %>% mutate(cluster=clusterProf) %>% filter(cluster==c)
res1 <- Segmentor(as.numeric(signal1), model=3, Kmax=30, keep=TRUE);
Kselect1<-SelectModel(res1, penalty='oracle', keep=FALSE)
rupt1 <- getBreaks(res1)[Kselect1,1:Kselect1]
plot(as.numeric(signal1),cex=0.5)
abline(v=rupt1,col="red")
```

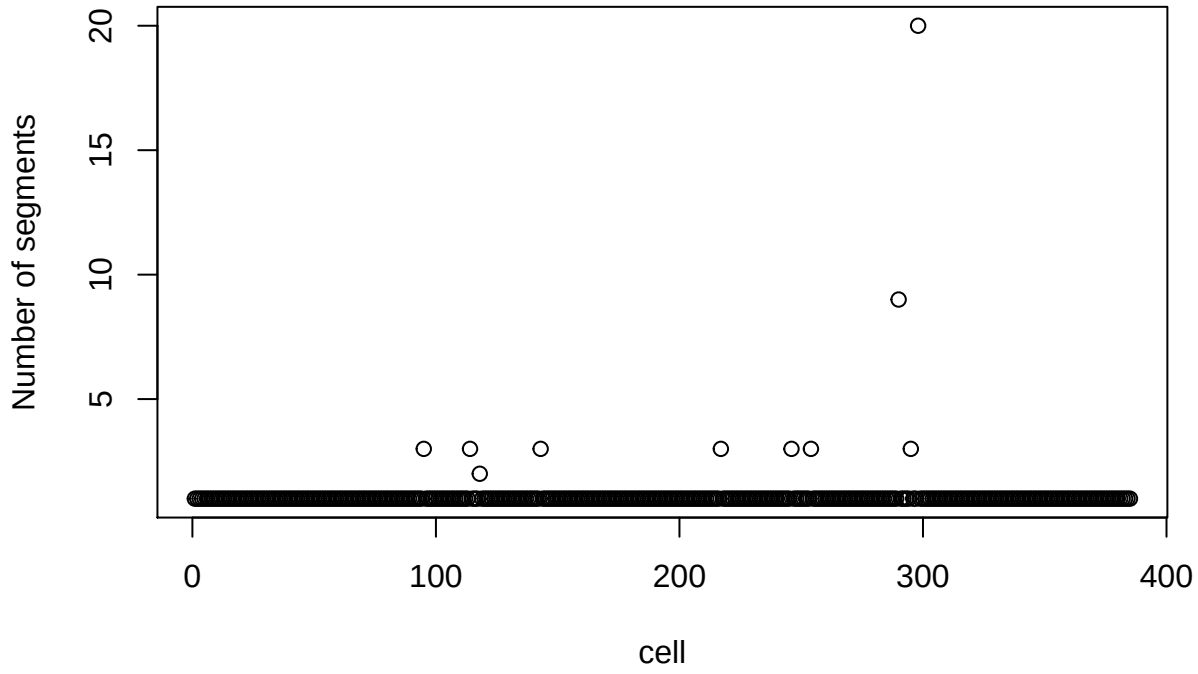



```
#Profil avec tous les genes, ceux gardés après fitering en rouge, segmentation obtenu sur ce profil fil
plot(as.numeric(signal1),cex=0.5)
points(KeepGene,signal1[KeepGene],col="red")
abline(v=KeepGene[rupt],col="red")
```

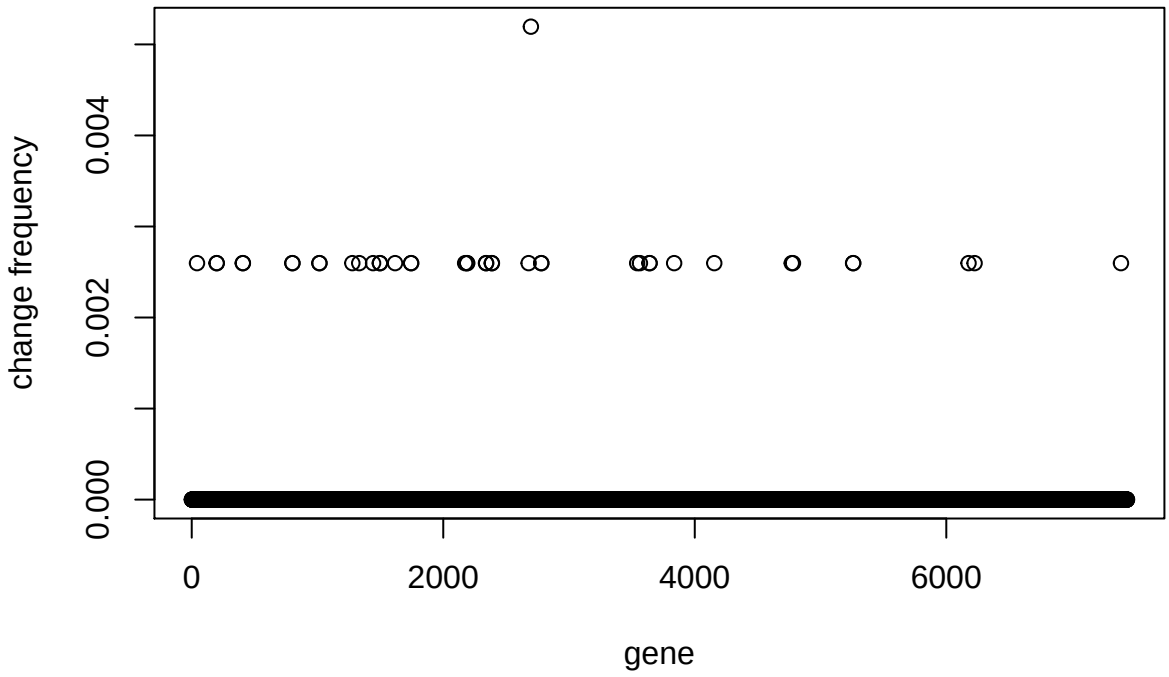


### Segmentation and clustering on all the genes

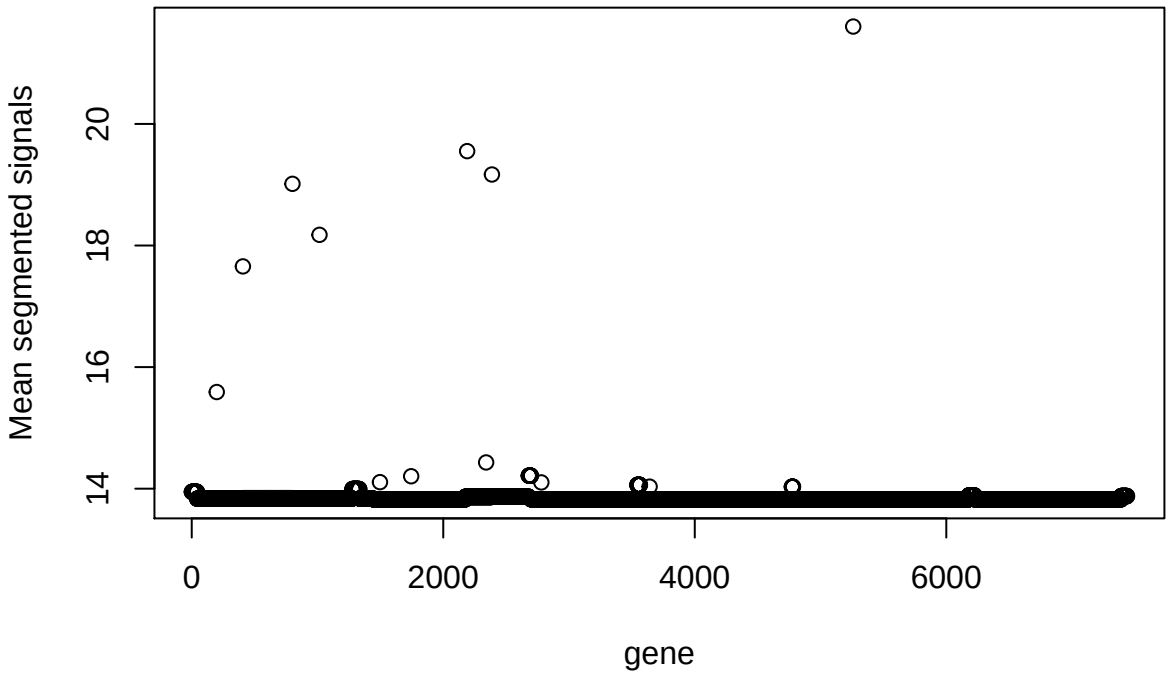
```
data_TotGene <- tab[which(rowMeans(tab)>=thres),]
n <- nrow(data_TotGene); p <- ncol(data_TotGene)
SegFileName <- paste0("CellSeg_thresCell_TotGene",thres,".rds",sep="")
R_TotGene <- readRDS(SegFileName)
CellKselect_TotGene <-  map_dbl(R_TotGene,~ .x$Kselect)
CellRupt.mean.pos_TotGene <-R_TotGene %>% map(.,"rupt.bin") %>% do.call(rbind,.) %>% colMeans(.)
CellPred_TotGene <- R_TotGene %>% map(.,"y.pred") %>% do.call(rbind,.) %>% as.data.frame()
colnames(CellPred_TotGene) <- colnames(data_TotGene)
rownames(CellPred_TotGene) <- rownames(data_TotGene)

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




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

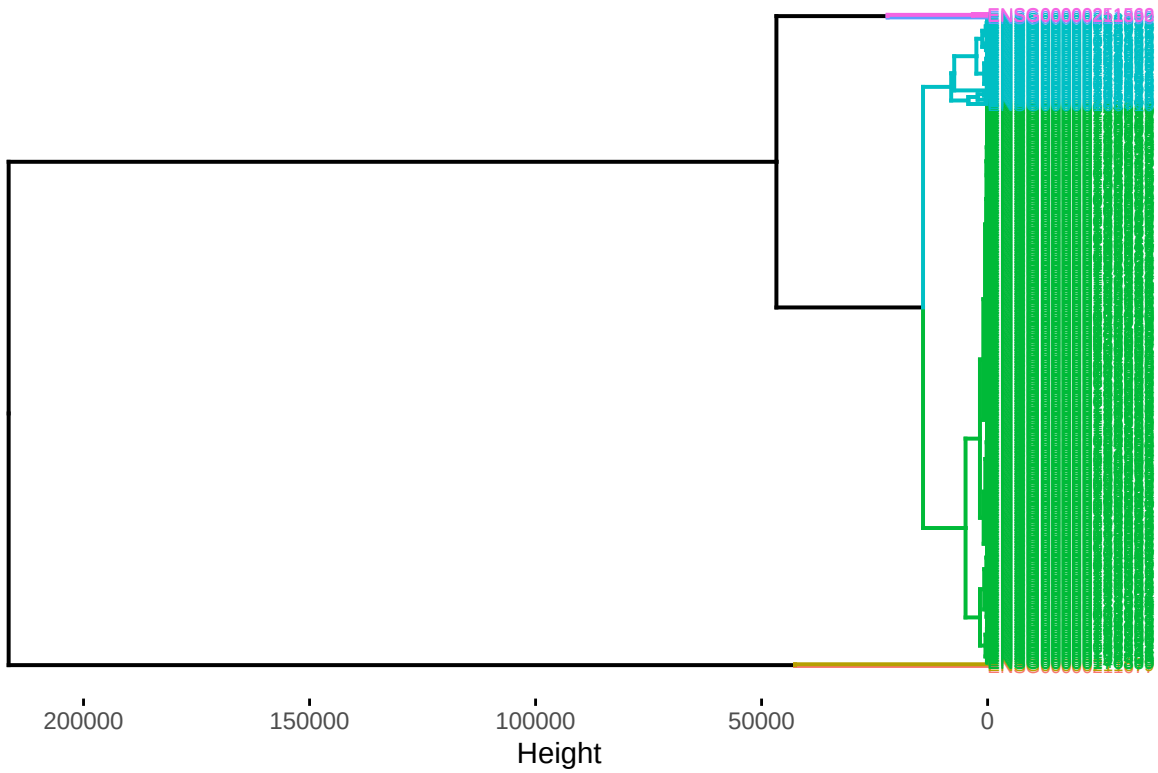



```
#Moyennes des profils segmentés
plot(colMeans(CellPred_TotGene),ylab="Mean segmented signals",xlab="gene")
```

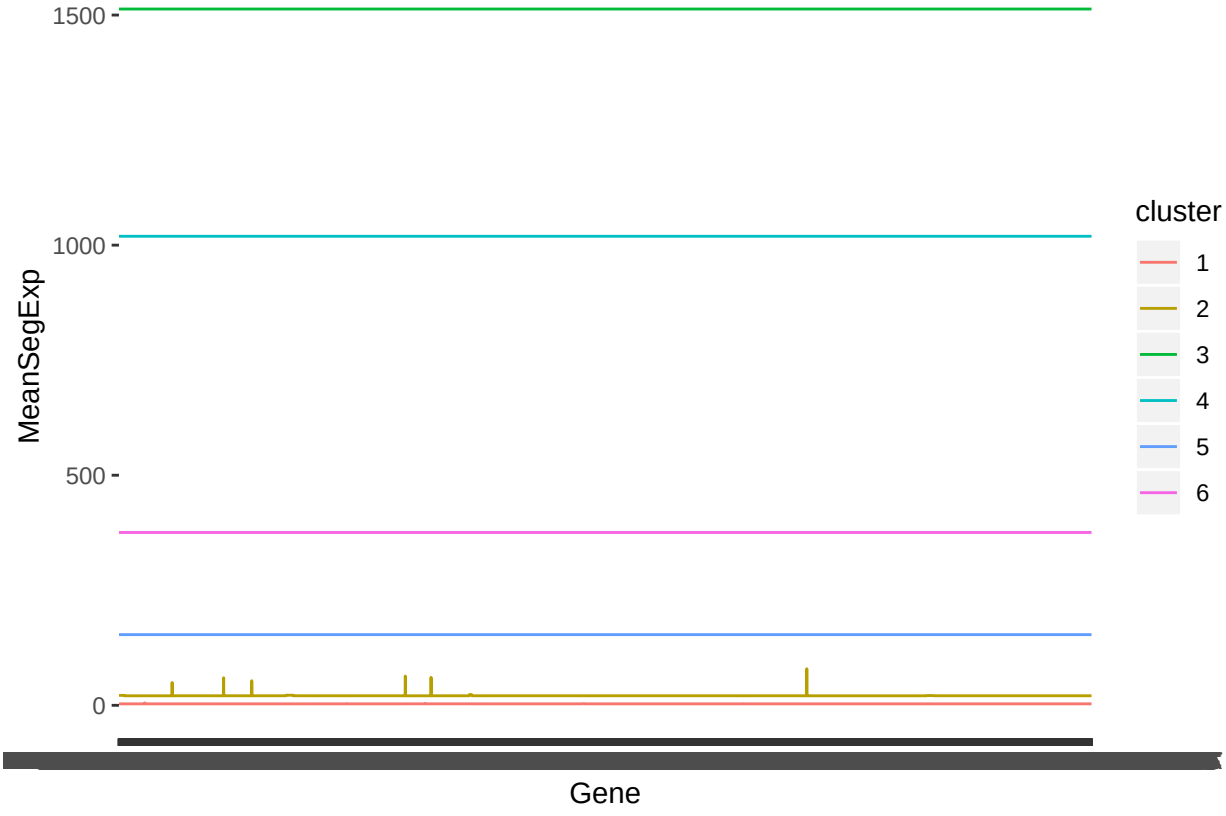



```
#Clustering
Dist.CellProf_TotGene <- dist(x = CellPred_TotGene)
HclustCellProf_TotGene <- hclust(d = Dist.CellProf_TotGene,method = "ward.D2")
NbClust=6
fviz_dend(HclustCellProf_TotGene,horiz = TRUE, cex = 0.5, k = NbClust, color_labels_by_k = TRUE)
```

Cluster Dendrogram



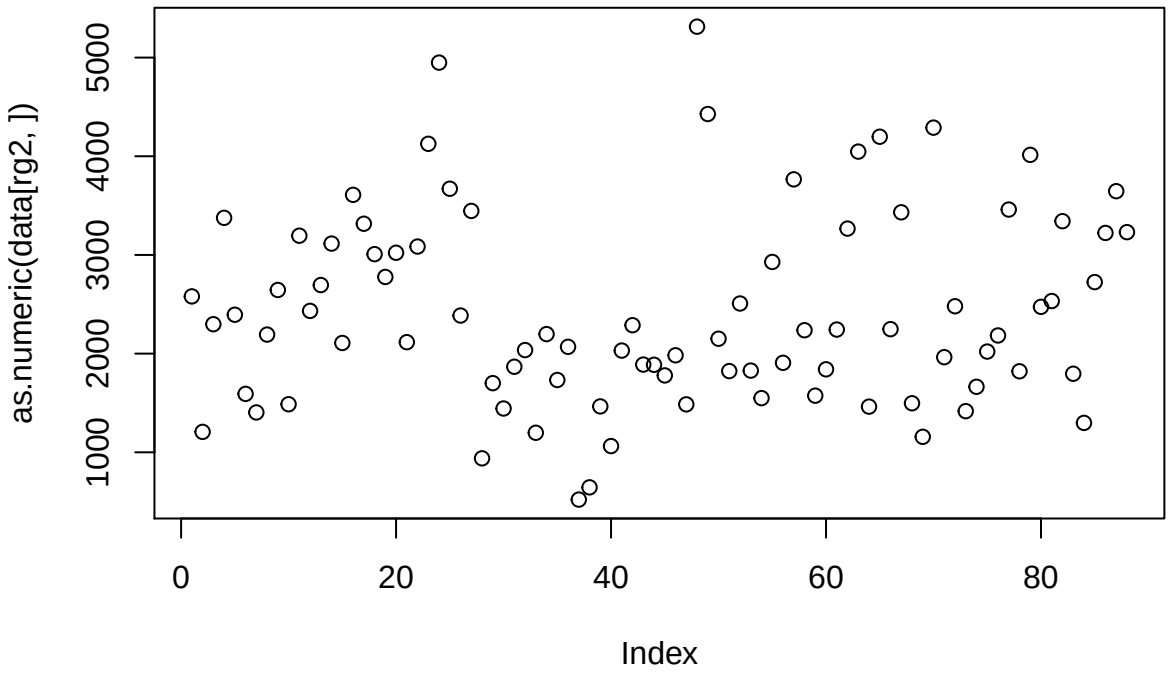

```
clusterProf_TotGene <- cutree(HclustCellProf_TotGene, k = 6)
MeanProfileByClust_TotGene <-
  CellPred_TotGene %>% mutate(cluster = as.factor(clusterProf_TotGene)) %>% group_by(cluster) %>%  summa

MeanProfileByClust_TotGene %>%
  ggplot(aes(x = Gene, y = MeanSegExp, group = cluster)) +
  geom_line(aes(color = cluster))
```




```
PropCellClustProf_TotGene <- clusterProf_TotGene %>%tibble %>% setnames("cluster") %>%  group_by(cluste
PropCellClustProf
```

```
# A tibble: 6 x 2
  cluster NbCell
    <int>  <int>
1       1    300
2       2     79
3       3      1
4       4      1
5       5      2
6       6      1
```

```
rg1=which(clusterProf_TotGene==6)
rg2=which(clusterProf==6)
plot(as.numeric(data[rg2,]))
```




```
plot(as.numeric(data_TotGene[rg1,]))
```