Exemple de correction du TP4 : test dans le modèle de bruit blanc gaussien

M2 Statistique, 2019-2020

Statistique mathématique en grande dimension et applications, cours de M. Roquain

Exercice 1

1)

```
library(sda)
```

```
## Loading required package: entropy
```

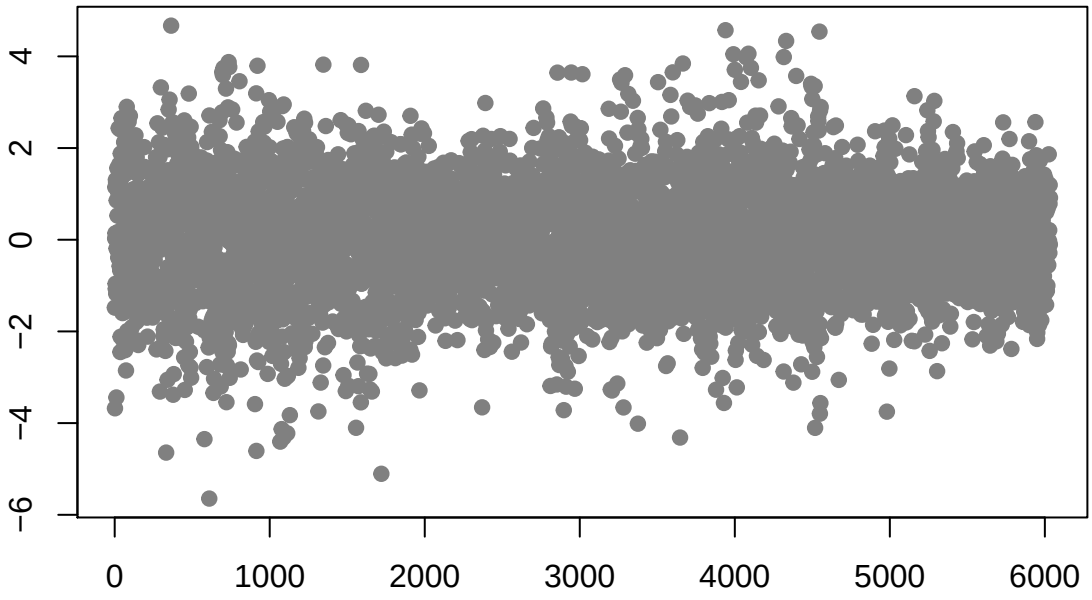
```
## Loading required package: corpcor
```

```
## Loading required package: fdrtool
```

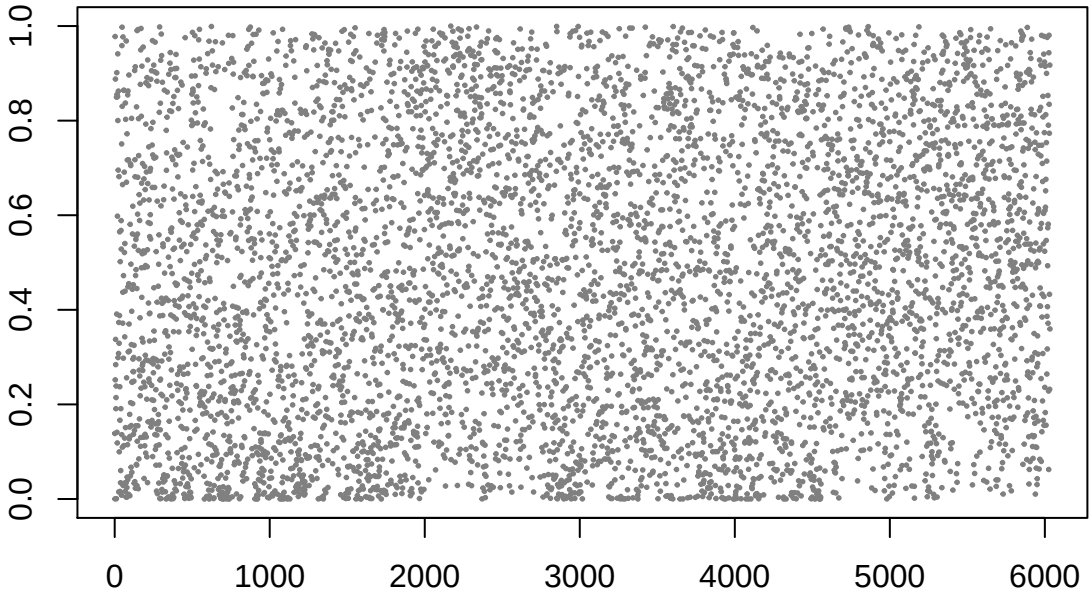
```
data(singh2002)

prostate=singh2002
X=prostate$x
dim(X)
```

```
## [1]  102 6033
```

```
n=dim(X)[1]
m=dim(X)[2]
n0=sum(prostate$y==prostate$y[1]) # groupe sain
n1=n-n0 #groupe malade
computestat=function(data) t.test(data[1:n0],data[(n0+1):n],var.equal=TRUE)$stat
Z=apply(X,2,computestat)
plot(Z,xlab="",ylab="",pch=19,col=gray(0.5))
```



```r
pvalues=2*(1-pnorm(abs(Z)))
plot(pvalues,xlab="",ylab="",pch=19,col=gray(0.5),cex=0.25)
```

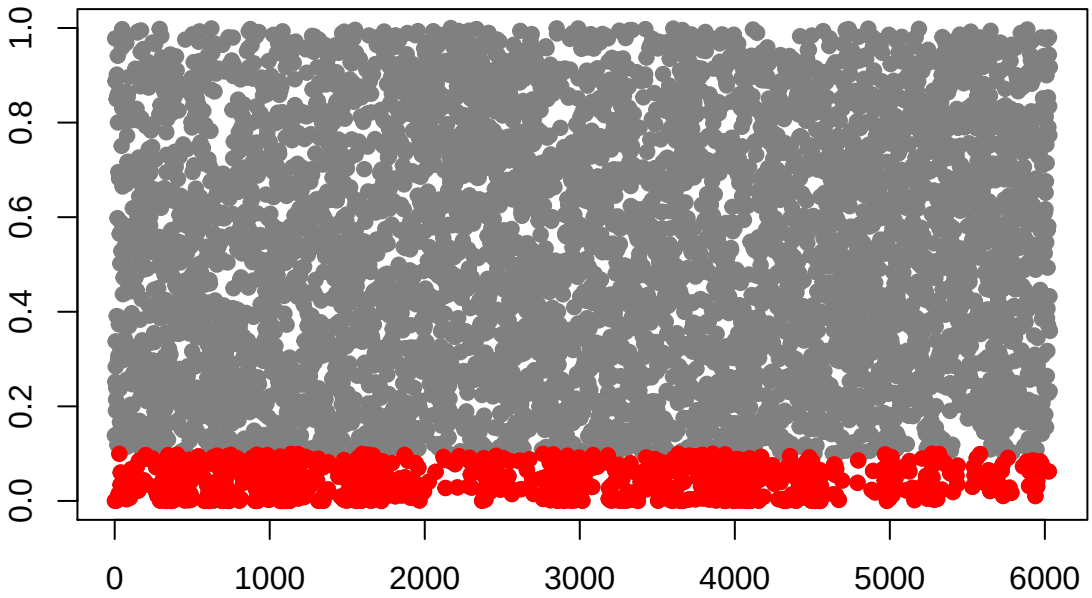


2)

```r
alpha=0.1
sum(pvalues<=alpha)
```

```
## [1] 817
```

```r
plot(pvalues,xlab="",ylab="",pch=19,col=gray(0.5))
points((1:m)[pvalues<=alpha],pvalues[pvalues<=alpha],pch=19,col="red")
```



Il s'agit de la procédure de test multiple non corrigée, elle n'assure pas le contrôle du nombre de faux positifs. Donc on ne peut `rien` conclure.

3)

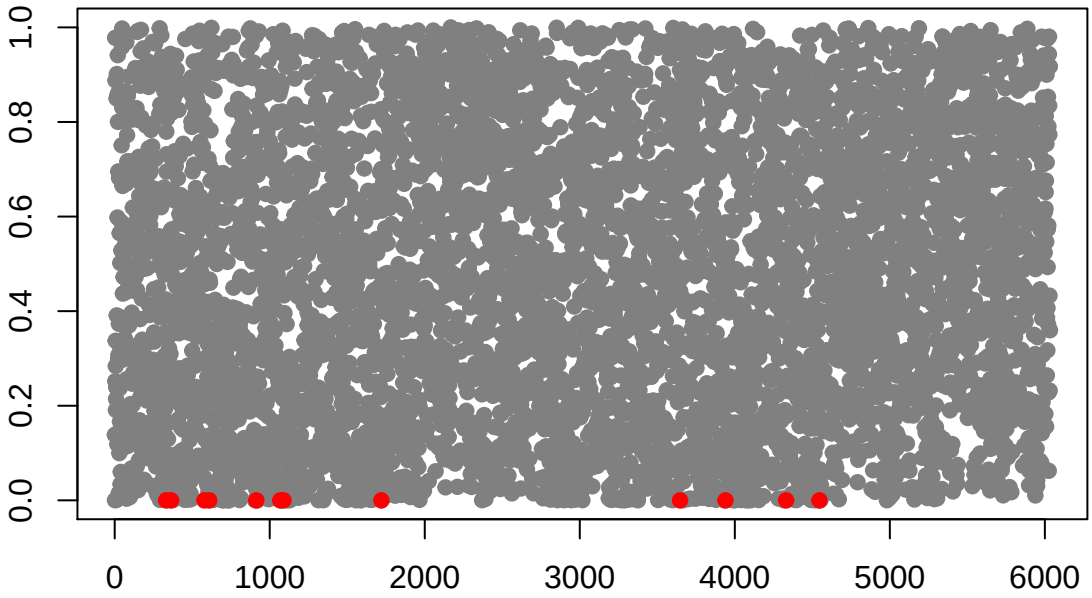
```
alpha=0.1
t=alpha/m
which(pvalues<=t)
```

```
##  [1]  332  364  579  610  914 1068 1089 1720 3647 3940 4331 4546
```

```
sum(pvalues<=t)
```

```
## [1] 12
```

```
plot(pvalues,xlab="",ylab="",pch=19,col=gray(0.5))
points((1:m)[pvalues<=t],pvalues[pvalues<=t],pch=19,col="red")
```



Ainsi, avec probabilité supérieure à 90%, cet ensemble de gènes ne contient aucun faux positifs, c'est-à-dire uniquement des gènes différentiellement exprimés.

4)

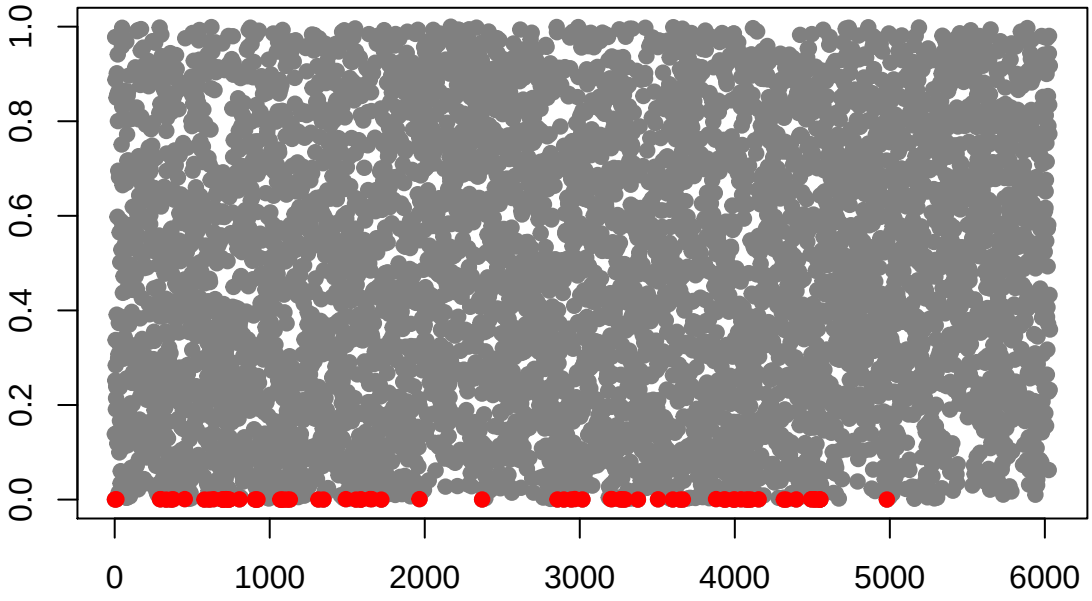
```
alpha=0.1
sortpvalues=sort(pvalues)
set=which(sortpvalues<=alpha*1:m/m)
kchap=0
if(length(set)>0) kchap=max(set)
t=alpha*kchap/m
which(pvalues<=t)
```

```
##  [1]    2   11  292  298  332  364  377  452  579  610  637  694  698  702
## [15]  718  721  735  739  805  905  914  921 1068 1077 1089 1113 1130 1314
## [29] 1346 1491 1557 1588 1589 1647 1659 1720 1966 2370 2856 2897 2945 2968
## [43] 3017 3200 3208 3260 3269 3282 3292 3375 3505 3600 3647 3665 3879 3930
## [57] 3940 3991 4000 4040 4073 4088 4104 4154 4316 4331 4396 4492 4496 4515
## [71] 4518 4546 4549 4552 4981
```

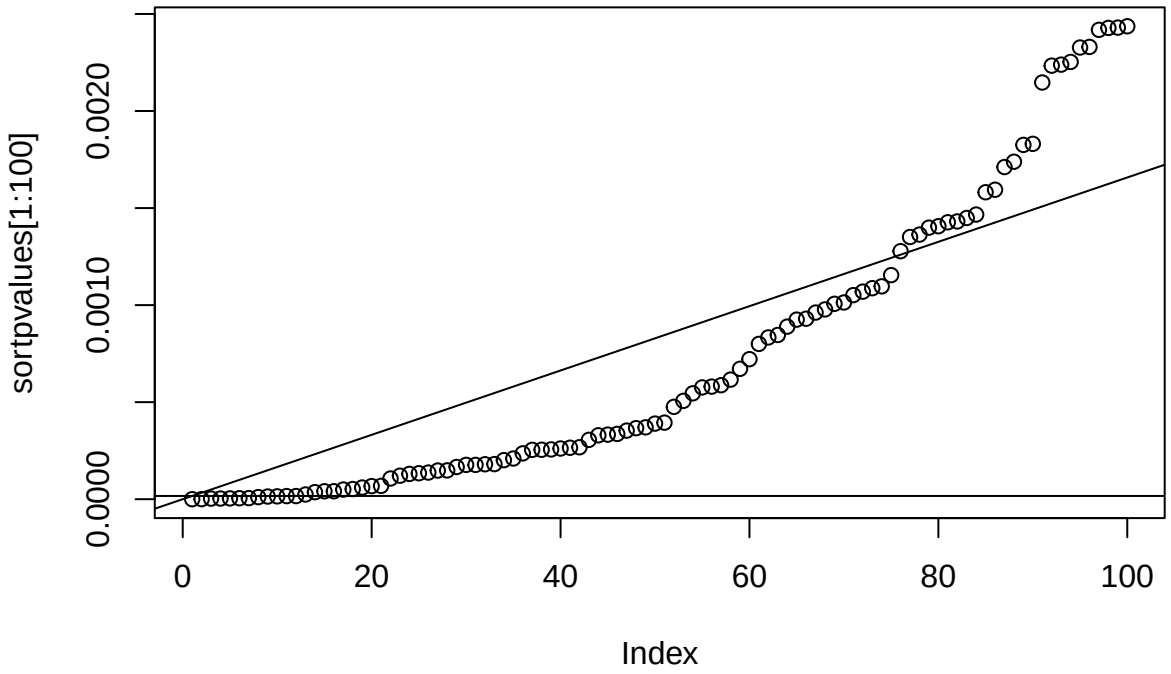
```
sum(pvalues<=t)
```

```
## [1] 75
```

```
plot(pvalues,xlab="",ylab="",pch=19,col=gray(0.5))
points((1:m)[pvalues<=t],pvalues[pvalues<=t],pch=19,col="red")
```




```r
plot(sortpvalues[1:100])
abline(a=0,b=alpha/m)
abline(h=alpha/m)
```



L'interprétation est qu'en moyenne, parmi les gènes sélectionnés, 10% peuvent être des faux positifs. On voit bien que prendre un critère plus permissif permet de faire davantage de découverte. On comprend bien pourquoi ce critère a du succès.

Exercice 2

On voit bien que le seuillage de BH "suit" la quantité de signal: sous une grande sparsité, il s'agit essentiellement de la procédure de Bonferroni. Si beaucoup de signal est présent, alors la procédure de BH ressemble à une

procédure non corrigée. On dit que la procédure s'adapte à la quantité de signal dans les données, ce qui est une propriété importante en grande dimension.