# False discovery rate control with unknown null distribution: illustrations on real data sets

Etienne Roquain and Nicolas Verzelen

2020-12-16

## Contents

This vignette illustrates some of the results delineated in Roquain and Verzelen (2020) for classical case/control data sets.

```
require("sansSouci.data") || remotes::install_github("pneuvial/sanssouci.data")
```

```
library(sda)
library(multtest)
library(Equalden.HD)
library(locfdr)
library(plot.matrix)
```

Set the seed of the random number generator for numerical reproducibility of the results:

```
set.seed(20200924)
```

# 1 Theoretical null approach

In this section, we consider classical case/control real data sets and we show that the theoretical null distribution $\mathcal{N}(0,1)$ can be inadequate to describe the overall behavior of the test statistics $(Y_i, 1 \le i \le n)$.

## 1.1 Standard analysis: the prostate cancer data set

The Singh et al. (2002) prostate cancer data set provides gene expression measures for two populations : a control group, which corresponds to “healthy” cells, and a case group, which corresponds to tumoral tissues. Based on this data set, we aim at identifying which genes are differentially expressed between the two groups.

The data are given under the form of a $d \times n$ matrix $X$, $d = d_0 + d_1$, where $d_0$ (resp. $d_1$) denotes the number of replications of the $n$-dimensional measurements for control (resp. case) individuals. The matrix $X$ is derived as follows:

```r
data(singh2002)
prostate=singh2002
X=prostate$x
#prostate$y
#dim(X)
d=dim(X)[1]
n=dim(X)[2]
d0=sum(prostate$y==prostate$y[1]) # control group
d1=d-d0 # case group
```

Note that the control group is given by the $d_0$ first rows of $X$.

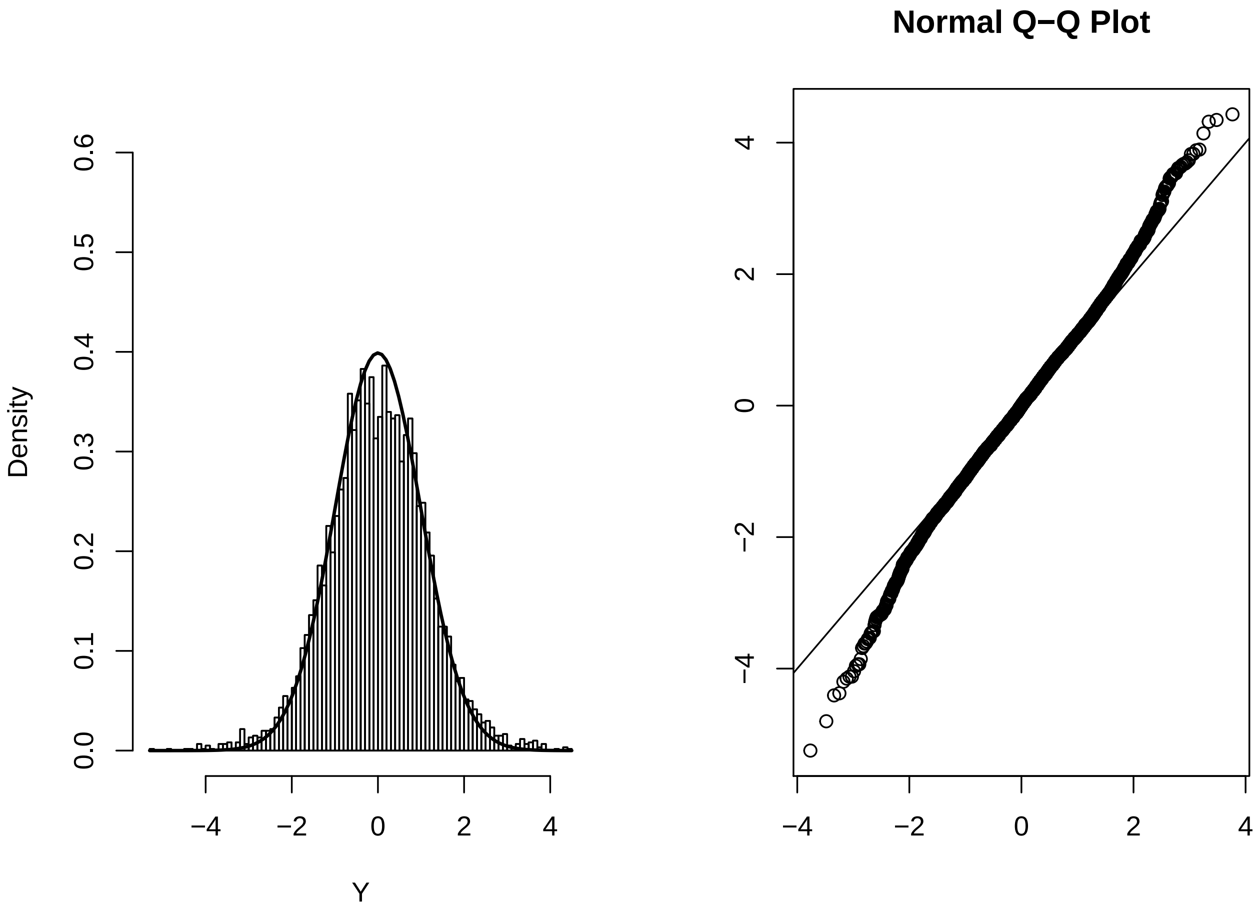
The vector of observed test statistics $Y = (Y_1, \dots, Y_n)$ is then built from $X$ via a standard $t$-test and a suitable Gaussian normalization:

```r
getY=function(X,d0,d){
  pvalues=apply(X,2,function(data)
    t.test(data[1:d0],data[(d0+1):d],var.equal=TRUE,alternative = c("greater"))$p.value)
  Y=-qnorm(pvalues)
}
Y=getY(X,d0,d)
```

Assuming that each line of $X$ follows a two-sample Gaussian model, the $p$-value corresponding to a gene $i$ such that the “control” group has the same mean than the “case” group has a marginal distributions which is uniform. Hence, if we are confident in this Gaussian modeling, the above normalization ensures that the marginal distribution of the corresponding $Y_i$ under the null is $\mathcal{N}(0, 1)$. Henceforth, this null distribution is referred as the *theoretical null distribution*.

We display how the theoretical null fits the overall shape of the histogram of the $Y_i$’s:

```r
par(mfrow=c(1,2))
hist(Y,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="")
curve(dnorm(x),lwd=2,add=TRUE)
hist(Y,nclass=70,freq=FALSE,add=TRUE)
qqnorm(Y,xlab="",ylab="")
abline(a=0,b=1)
```



We note that the fit is acceptable, so that it seems reasonable to run the standard BH procedure with this theoretical null, as follows.

```
BH=function(Y,alpha){
    n=length(Y)
    pvalues=2*(1-pnorm(abs(Y)))
    sortpvalues=sort(pvalues)
    rejet = sortpvalues <= alpha*1:n/n
    threshold=0
    if(sum(rejet)>0) threshold = max(which(rejet))
    rejectedset=which(pvalues<=alpha*threshold/n)
  nbrejections=length(rejectedset)
    return(nbrejections)
}
```

```
nbrejections=BH(Y,alpha=0.1)
```

Assuming that the BH procedure is well controlling the FDR in that context, it suggests that the 59 identified genes contained at most $\alpha = 10\%$ of false discoveries (on average).

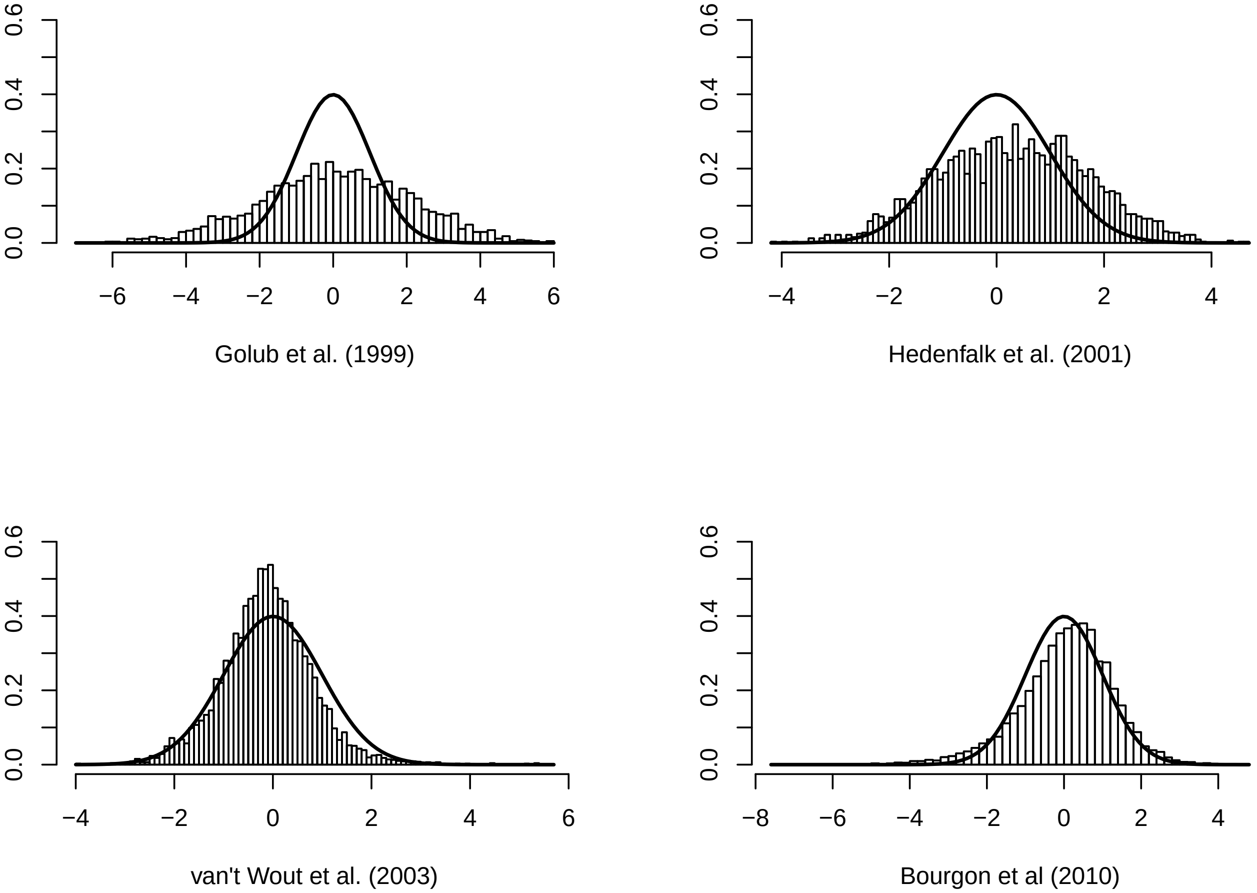
## 1.2 Criticism of theoretical null

We apply below the same pipe-line of analyses to the four data sets Golub et al. (1999), Hedenfalk et al. (2001), Wout et al. (2003) and Bourgon, Gentleman, and Huber (2010).

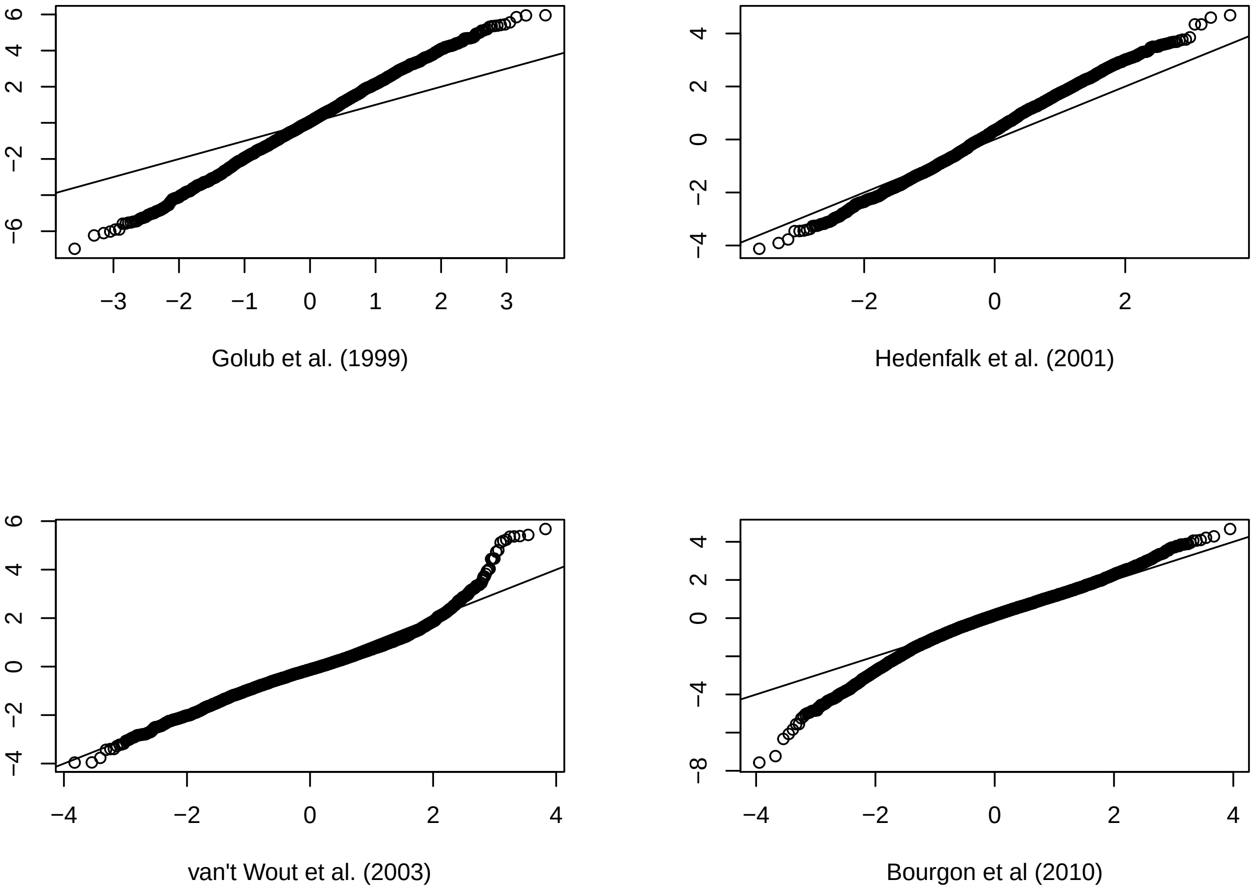
```r
# Golub et al. (1999) gene expression dataset
data(golub)
X=t(golub)
d=length(golub.cl)
d0=sum(golub.cl==golub.cl[1]) # control
d1=d-d0 # case
Ygolub=getY(X,d0,d)
nbrejectionsgolub=BH(Ygolub,alpha=0.1)
# Hedenfalk et al. (2001) breast cancer dataset
data(Hedenfalk)
X=t(Hedenfalk)
d=dim(X)[1]
d0=7
Yheden=getY(X,d0,d)
nbrejectionsheden=BH(Yheden,alpha=0.1)
# van’t Wout et al. (2003) HIV data set
data(hivdata)
Ywout=hivdata
nbrejectionshiv=BH(Ywout,alpha=0.1)
# Bourgon et al (2010) acute lymphoblastic leukemia (ALL) data set
data(expr_ALL, package = "sansSouci.data")
X=t(expr_ALL)
X=X[c(which(rownames(X)=="NEG"), which(rownames(X)=="BCR/ABL")),]
d=dim(X)[1]
d0=length(which(rownames(X)=="NEG"))
YBourgon=getY(X,d0,d)
nbrejectionsALL=BH(YBourgon,alpha=0.1)
```

The four fits of the corresponding theoretical nulls are displayed below.

```r
par(mfrow=c(2,2))
hist(Ygolub,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="",xlab="Golub et al. (1999)",ylab="")
curve(dnorm(x),lwd=2,add=TRUE)
hist(Ygolub,nclass=70,freq=FALSE,add=TRUE)
hist(Yheden,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="",xlab="Hedenfalk et al. (2001)",ylab="
curve(dnorm(x),lwd=2,add=TRUE)
hist(Yheden,nclass=70,freq=FALSE,add=TRUE)
hist(Ywout,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="",xlab="van't Wout et al. (2003)",ylab="
curve(dnorm(x),lwd=2,add=TRUE)
hist(Ywout,nclass=70,freq=FALSE,add=TRUE)
hist(YBourgon,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="",xlab="Bourgon et al (2010)",ylab=""
curve(dnorm(x),lwd=2,add=TRUE)
hist(YBourgon,nclass=70,freq=FALSE,add=TRUE)
```



```
par(mfrow=c(2,2))
qqnorm(Ygolub,xlab="Golub et al. (1999)",ylab="",main="")
abline(a=0,b=1)
qqnorm(Yheden,xlab="Hedenfalk et al. (2001)",ylab="",main="")
abline(a=0,b=1)
qqnorm(Ywout,xlab="van't Wout et al. (2003)",ylab="",main="")
abline(a=0,b=1)
qqnorm(YBourgon,xlab="Bourgon et al (2010)",ylab="",main="")
abline(a=0,b=1)
```



The four fits are rather poor and it seems that many $Y_i$'s do not follow the theoretical null distribution. This could imply that most alternative hypotheses are true, which would imply that most genes are differentially expressed. Alternatively, this could be due to the fact that the theoretical null is inadequate here and that, under the null (no differential expression of gene $i$), the $Y_i$'s follow a different distribution.
Unless the statistician has a strong belief that most genes are differentially expressed, the theoretical null is inadequate here. In particular, in a sparse situation where there are only a few alternative, relying on the theoretical null $\mathcal{N}(0,1)$ could be desastrous.

This observation was originally made in by Efron (2008) who also gave some possible explanations for this phenomenon. In particular, he hypothetized that the presence of confounding factors or strong correlations between the genes: while the (unconditional) marginal distribution of $Y_i$ could be standard normal, the empirical distribution of the sample $(Y_i, 1 \leq i \leq n)$ would be different.

## 2 Empirical null approaches

The previous section shows that the theoretical null can be inadequate. In this section, we show that fitting an empirical null distribution to the data can yield to a more meaningful result.

As investigated by Efron (2008), a part of the phenomenon delineated in the above section can be captured by rescaling the null appropriately by some mean $\theta$ and standard deviation $\sigma$. Indeed, the scaling parameters $(\theta, \sigma)$ can be considered as factors disturbing all the measurements simultaneously, that induce some dependencies between the $Y_i$'s when some of these two are random. Namely, if $Y_i = \theta + \sigma\xi_i$ under the null for $\xi_i$ i.i.d. $\mathcal{N}(0,1)$, with $\theta \sim \mathcal{N}(0, 1-\sigma^2)$ (and independent), then the null distribution is $\mathcal{N}(0,1)$ *unconditionnally on the scaling* but becomes $\mathcal{N}(\theta, \sigma^2)$ *conditionnally on the scaling*. To this respect, estimating the conditional null could be more meaningful that the unconditional one, and solve partially the gap observed in the previous pictures. In addition, conveniently, the measurements $Y_i$'s are independent conditionnally on the scaling parameters in this case, at least those under the nulls.

This suggests to consider the Huber model (with fixed mixture) used in Roquain and Verzelen (2020) where the observations $(Y_i, 1 \leq i \leq n)$ are assumed to be *independent*, with most of them coming from some unknown null distribution $F_0$, typically some scaled Gaussian, while the remaining ones are let arbitrary. The aim here is to estimate $F_0$ so that the corresponding (plug-in) BH procedure has good performances.

### 2.1 A new goodness of fit test for the null distribution

First, let us investigate the criticism raised by Efron, by testing whether the null c.d.f. $F_0$ could be equal to the theoretical null c.d.f. $\Phi$, that is, the standard Gaussian tail. To this end, the new goodness of fit test developed in Roquain and Verzelen (2020) can be used. This test requires the statistician to set a prescribed upper bound $\overline{\pi}$ on the proportion of true alternatives in the data.

Recall that this test reject the hypothesis $F_0 = \Phi$ if

$$\exists k \in \{0, \dots, n\}, \text{ such that } \tilde{a}_n(k; \Phi) > \tilde{b}_n(k; \Phi), \tag{1}$$

where

$$\begin{aligned}
\tilde{a}_n(k; F_0) &= 0 \vee \frac{\max_{0 \leq \ell \leq k} \left\{\ell/n - (1-\overline{\pi}) F_0(Y_{(\ell)})\right\} - c_{n,\alpha}}{\overline{\pi}};\\
\tilde{b}_n(k; F_0) &= 1 \wedge \frac{\min_{k \leq \ell \leq n} \left\{\ell/n - (1-\overline{\pi}) F_0(Y_{(\ell+1)})\right\} + c_{n,\alpha}}{\overline{\pi}},
\end{aligned}$$

where $c_{n,\alpha} = \{-\log(\alpha/2)/(2n)\}^{1/2}$ and $\alpha$ is the level of the test.

These quantities and the test can be derived as follows.

```
getcnalpha=function (n,alpha,pibar) sqrt(-(1-pibar)*log(alpha/2)/(2*n))

getankbnkF0=function(alpha,F0,pibar,Y){
  n=length(Y)
  cnalpha=getcnalpha(n,alpha,pibar)
  sortY=sort(Y)
  truca=c(0,sapply(1:n,function(l) l/n-(1-pibar)*F0(sortY[l])))
  ank=pmax(0,(cummax(truca)-cnalpha)/pibar)
  #plot(ank)
  trucb=c(sapply(0:(n-1),function(l) l/n-(1-pibar)*F0(sortY[l+1])),1)
  bnk=pmin(1,(cummin(trucb[(n+1):1])[(n+1):1]+cnalpha)/pibar)
  #plot(bnk)
  return(matrix(c(ank,bnk), n+1,2,byrow=FALSE))
}

acceptF0=function(alpha,F0,pibar,Y){
 n=length(Y)
 res=getankbnkF0(alpha,F0,pibar,Y)
 return(sum(res[,1]<=res[,2])>n)
}
```

```
getpvalueF0=function(alpharange,F0, pibar,Y){
  accept=sapply(alpharange, function(alpha) acceptF0(alpha,F0,pibar,Y))
  pvalue=1
  set=which(accept==FALSE)
  if(length(set)>0) pvalue=alpharange[min(set)]
  return(pvalue)
}

alpharange=sort(exp(-seq(0.1,20,1)),decreasing = FALSE)
pibar=0.1 #allows quite dense signal

tab <- data.frame("Data" = c("Golub","Heden", "HIV","ALL"),
                  "p-value" = c(getpvalueF0(alpharange,pnorm,pibar,Ygolub),
                  getpvalueF0(alpharange,pnorm,pibar,Yheden), getpvalueF0(alpharange,pnorm,pibar,Ywout)
                  getpvalueF0(alpharange,pnorm,pibar,YBourgon)
                  ),check.names = FALSE, row.names = NULL)
knitr::kable(tab)
```

| Data | p-value |
|---|---:|
| Golub | 0e+00 |
| Heden | 0e+00 |
| HIV | 0e+00 |
| ALL | 2e-06 |

We have set $\bar{\pi} = 0.1$ in the example thereby allowing for possibly dense signal. The test rejects the null hypothesis $F_0 = \Phi$ for all four data sets.

## 2.2 Gaussian empirical null

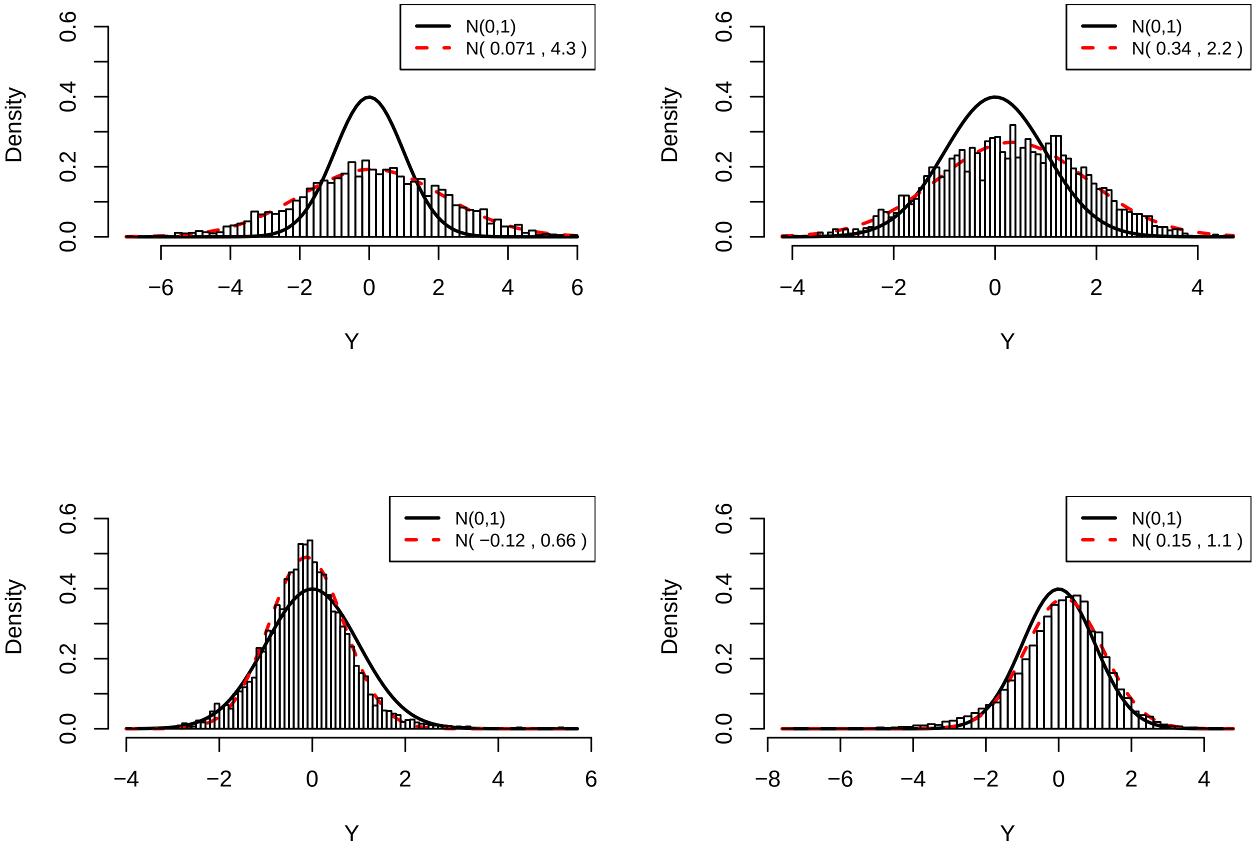
Since the theoretical null cannot be used for $F_0$, we should investigate the delicate task of estimating $F_0$ from the data. Following Efron's argument, we assume that $F_0$ is Gaussian, with some scaling parameters $\theta, \sigma^2$. Hence, the null estimation problem boils down to inferring these scaling parameters. For this, we rely on the classical optimal robust estimators, that is the median $\widetilde{\theta}$ and the median of absolute deviation $\widetilde{\sigma}$ (MAD) of the sample, respectively.

```
getthetatilde=function(Y) quantile(Y,1/2,type=1)
getsigmatilde=function(Y) quantile(abs(Y-getthetatilde(Y)),1/2,type=1)/sqrt(qchisq(1/2,d=1))
```

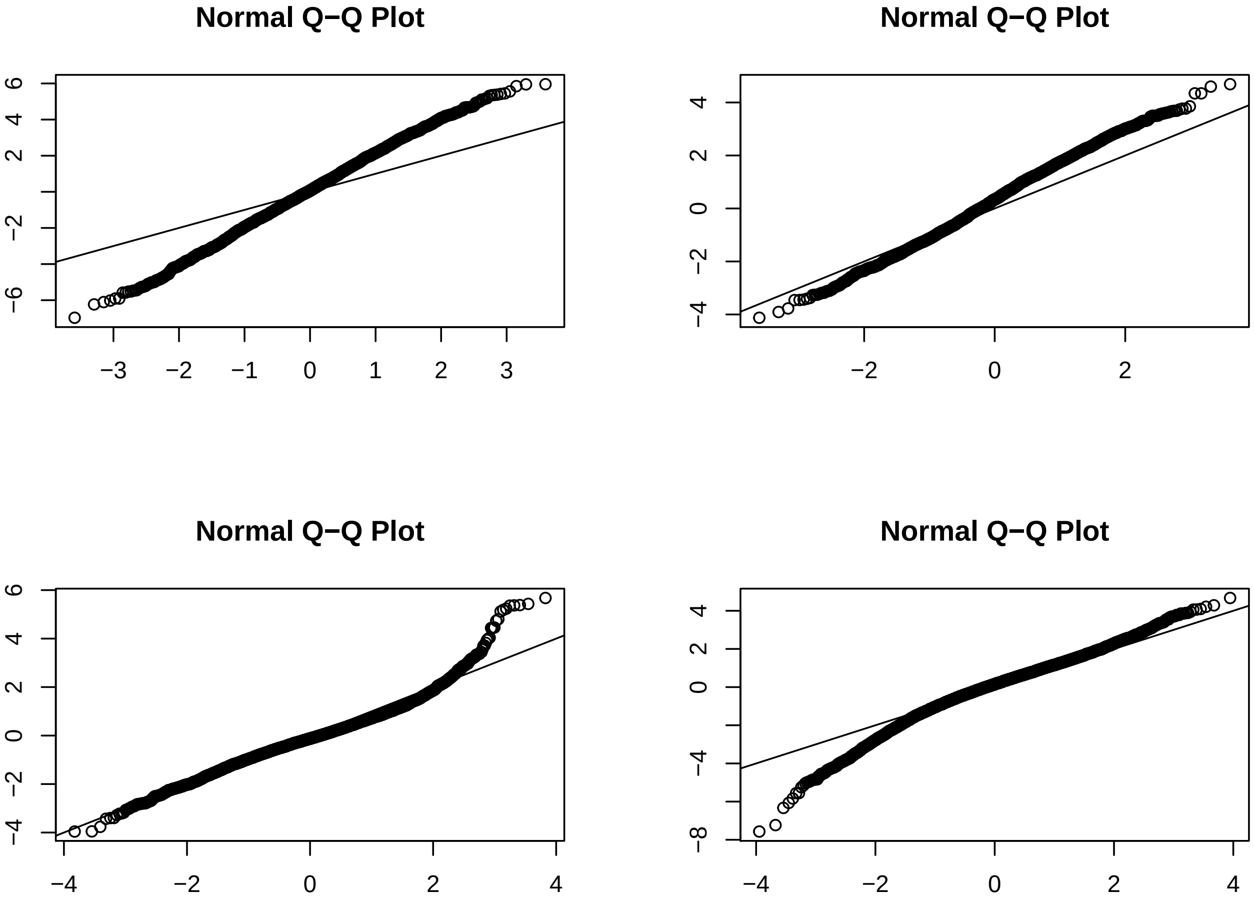
The empirical null distributions $\mathcal{N}(\widetilde{\theta}, \widetilde{\sigma}^2)$ are displayed below for the four data sets considered above.

```
plotempfit=function(Y){
  thetatilde=getthetatilde(Y)
  sigmatilde=getsigmatilde(Y)
  hist(Y,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="")
curve(dnorm(x),lwd=2,add=TRUE)
curve(dnorm(x,thetatilde,sigmatilde),lwd=2,lty=2,add=TRUE,col="red")
hist(Y,nclass=70,freq=FALSE,add=TRUE)
legend("topright",c("N(0,1)",
        paste("N(",signif(thetatilde,2),",",signif(sigmatilde^2,2),")")),
       lwd=c(2,2),lty=c(1,2),col=c("black","red"),cex=0.8)
return(c(BH(Y,alpha=0.1),BH((Y-thetatilde)/sigmatilde,alpha=0.1)))
}
```

```
par(mfrow=c(2,2))
nbrejetgolub=plotempfit(Ygolub)
nbrejetheden=plotempfit(Yheden)
nbrejetWout=plotempfit(Ywout)
nbrejetBourgon=plotempfit(YBourgon)
```



```
par(mfrow=c(2,2))
qqnorm(Ygolub,xlab="",ylab="")
abline(a=0,b=1)
qqnorm(Yheden,xlab="",ylab="")
abline(a=0,b=1)
qqnorm(Ywout,xlab="",ylab="")
abline(a=0,b=1)
qqnorm(YBourgon,xlab="",ylab="")
abline(a=0,b=1)
```



## 2.3 Plug-in BH procedure

The plug-in BH procedure is the Benjamini Hochberg procedure Benjamini and Hochberg (1995) used with empirical Gaussian null distributions. In Roquain and Verzelen (2020), it has been shown that, if the proportion of true alternatives is small enough in the data, that is, if the sparsity is small enough, this plug-in BH procedure mimicks the oracle-BH procedure. Here, we report the number of rejections of the plug-in BH procedure for the four above data sets.

```
tab <- data.frame("Data" = c("Golub","Heden", "Wout","Bourgon"),
                  "theoretical BH procedure" = c(nbrejetgolub[1],nbrejetheden[1], nbrejetWout[1],nbreje
                  "plug-in BH procedure" = c(nbrejetgolub[2],nbrejetheden[2], nbrejetWout[2],nbrejetBou
knitr::kable(tab)
```

| Data | theoretical BH procedure | plug-in BH procedure |
|---|---:|---:|
| Golub | 876 | 0 |
| Heden | 124 | 0 |
| Wout | 22 | 111 |
| Bourgon | 251 | 182 |

It is apparent that the discovered variables highly depend on the plugged null distribution. For instance, for the data Golub et al. (1999), the theoretical BH procedure makes 876 discoveries, while the empirical BH procedure does not make any discovery. For Wout et al. (2003), this is the other way around ; the theoretical BH procedure makes only 22 discoveries, while the empirical BH procedure makes 111 discoveries. Hence, the theoretical BH procedure could generate a lot of false discoveries (or false non-discoveries). This reinforces the interest in suitably estimating the null before applying the BH procedure.

## 2.4 Label-permuted empirical null

In (multiple) testing, a popular method for estimating the null is to use permutations of the labels. Let us investigate how this compares to the previous method in our context.

The permutation method consists in switching randomly the labels case/control in the data base to mimick the situation where the individuals are exchangeable accross the sample. Since there are several variables, there are basically two ways to achieve this:

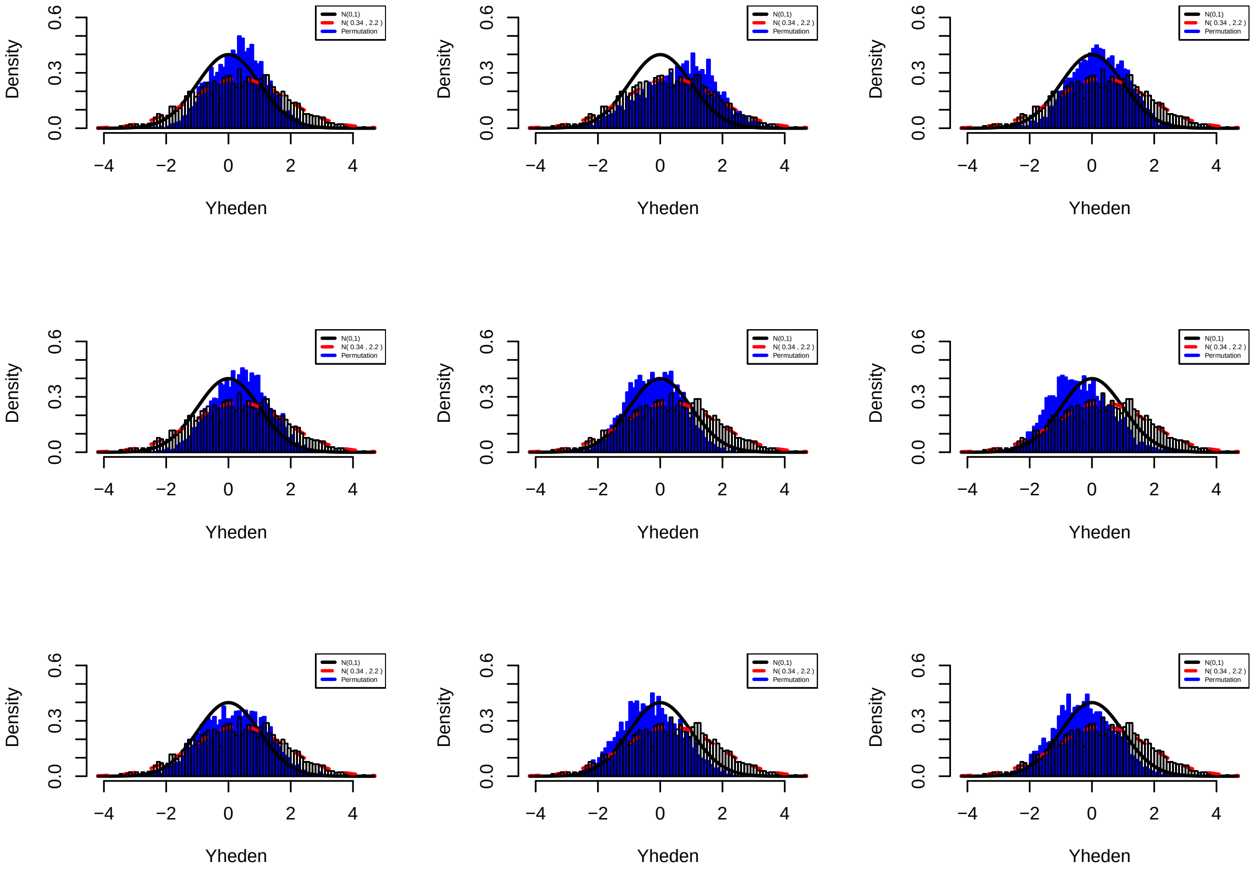
- either we consider a permutation per variable, that is, each column of the matrix $X$ is permuted independently. In that case, the dependence structure accross the variables is lost;
- or the permutations act simultaneously on all the variables, that is, the permutation operation is applied to the whole lines of the matrix $X$. In that case, the dependence structure accross the variables is maintained;

We focus on the second situation which keeps the dependence structure information. We can compute the permuted measurements $X^{(b)}$, $b = 1, \dots, B$, obtained by applying $B$ permutations to the lines of $X$. Then, applying each time our function `getY(X,d0,d)`, we obtain a sample $(Y^{(b)}, b = 1, \dots, B)$ of variables in $\mathbb{R}^n$.

```
getpermY=function(X,d0,d,B){
  Yperm=sapply(1:B, function(b)  getY(X[sample(d),],d0,d))
  return(Yperm)
}
```

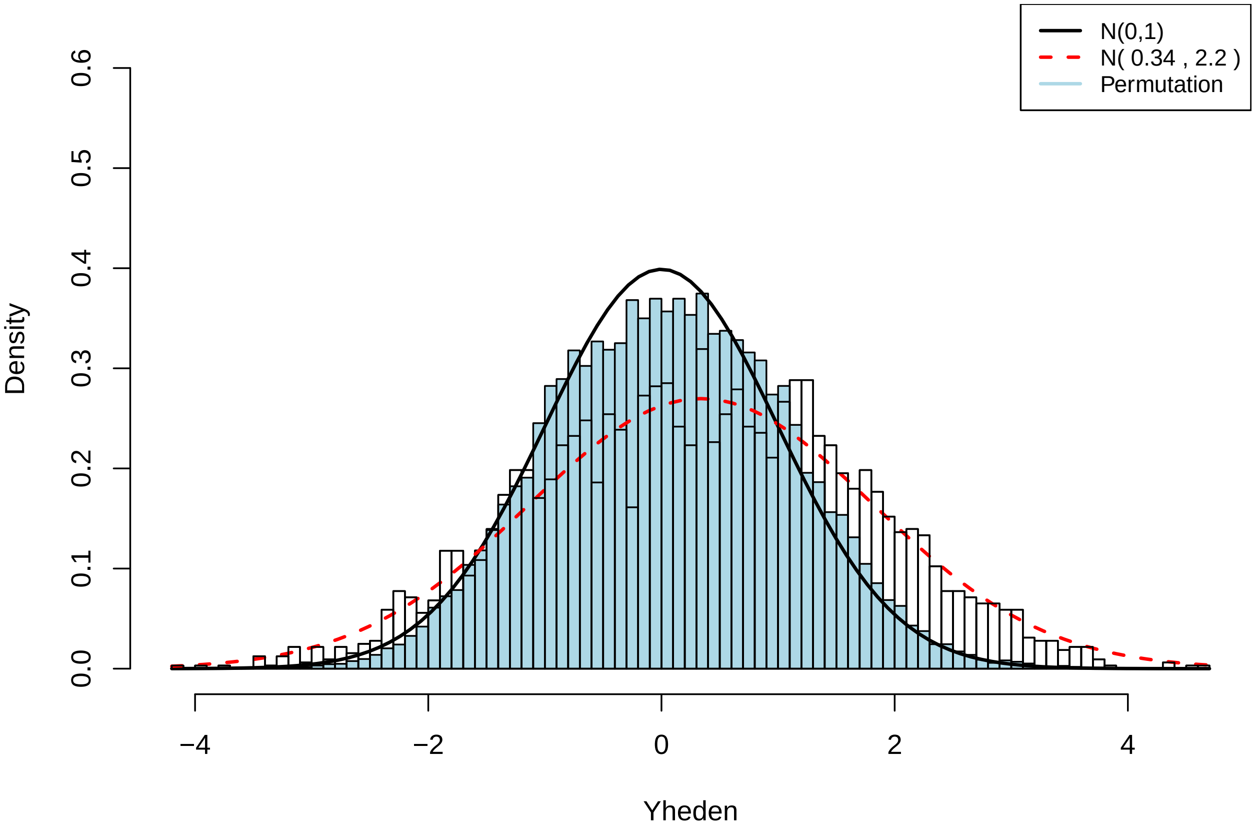
Doing so, the empirical distribution of each $Y^{(b)}$ can be used to approximate the overall null distribution. Let us apply this for the data set Hedenfalk et al. (2001).

```
data(Hedenfalk)
X=t(Hedenfalk)
d=dim(X)[1]
d0=7
B=9
Yperm=getpermY(X,d0,d,B)
thetatilde=getthetatilde(Yheden)
sigmatilde=getsigmatilde(Yheden)

par(mfrow=c(sqrt(B),sqrt(B)))
for (b in 1:B){
  hist(Yheden,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="")
hist(Yperm[,b],nclass=70,freq=FALSE,add=TRUE,col="blue",border="blue")
  curve(dnorm(x),lwd=2,add=TRUE)
curve(dnorm(x,thetatilde,sigmatilde),lwd=2,lty=2,add=TRUE,col="red")
hist(Yheden,nclass=70,freq=FALSE,add=TRUE)
legend("topright",c("N(0,1)",
        paste("N(",signif(thetatilde,2),",",signif(sigmatilde^2,2),")"),"Permutation"),
       lwd=c(2,2,2),lty=c(1,2,1),col=c("black","red","blue"),cex=0.4)
}
```



Unfortunately, these permutation-based null estimations vary from one permutation to another. Besides, this does not seem to fit the empirical distribution. To obtain a stable estimate, one could try to concatenate all the measurements $Y^{(b)}$, for $b = 1, \ldots, B$ and to use the resulting empirical distribution as an estimation of the null distribution.

```
  hist(Yheden,nclass=70,ylim=c(0,1.6/sqrt(2*pi)),freq=FALSE,main="")
hist(Yperm,nclass=70,freq=FALSE,add=TRUE,col="lightblue")
  curve(dnorm(x),lwd=2,add=TRUE)
curve(dnorm(x,thetatilde,sigmatilde),lwd=2,lty=2,add=TRUE,col="red")
hist(Yheden,nclass=70,freq=FALSE,add=TRUE)
legend("topright",c("N(0,1)",
        paste("N(",signif(thetatilde,2),",",signif(sigmatilde^2,2),")"),"Permutation"),
       lwd=c(2,2,2),lty=c(1,2,1),col=c("black","red","lightblue"),cex=0.8)
```



Unfortunately, the corresponding estimator is close to the theoretical null $\mathcal{N}(0,1)$ and not to the empirical distribution of the data. A possible explanation is that the structure of dependence of the variables is suppressed by the concatenation operation.

## 3 Confidence region with a stability indicator for the rejected set

In Roquain and Verzelen (2020), it has been shown that using the plug-in BH procedure is safe when the data are sufficiently sparse (only few true alternatives), but that no method can mimick the oracle plug-in BH procedure otherwise, in the minimax sense. This means that, without enough sparsity, there exists a model configuration where the oracle plug-in BH procedure is out of reach. An example of such a configuration is given in the proof of the lower bound in Roquain and Verzelen (2020). In this configuration, the numerical experiments in Roquain and Verzelen (2020) shows that classical procedure indeed fails: either the power is low, or the FDR control is lost (as it is the case for the `locfdr` package).

However, this minimax result is pessimistic as, for some other distributions of the alternatives, plug-in could still be possible. This raises the challenge of assessing the performance of plug-in for the data at hand.

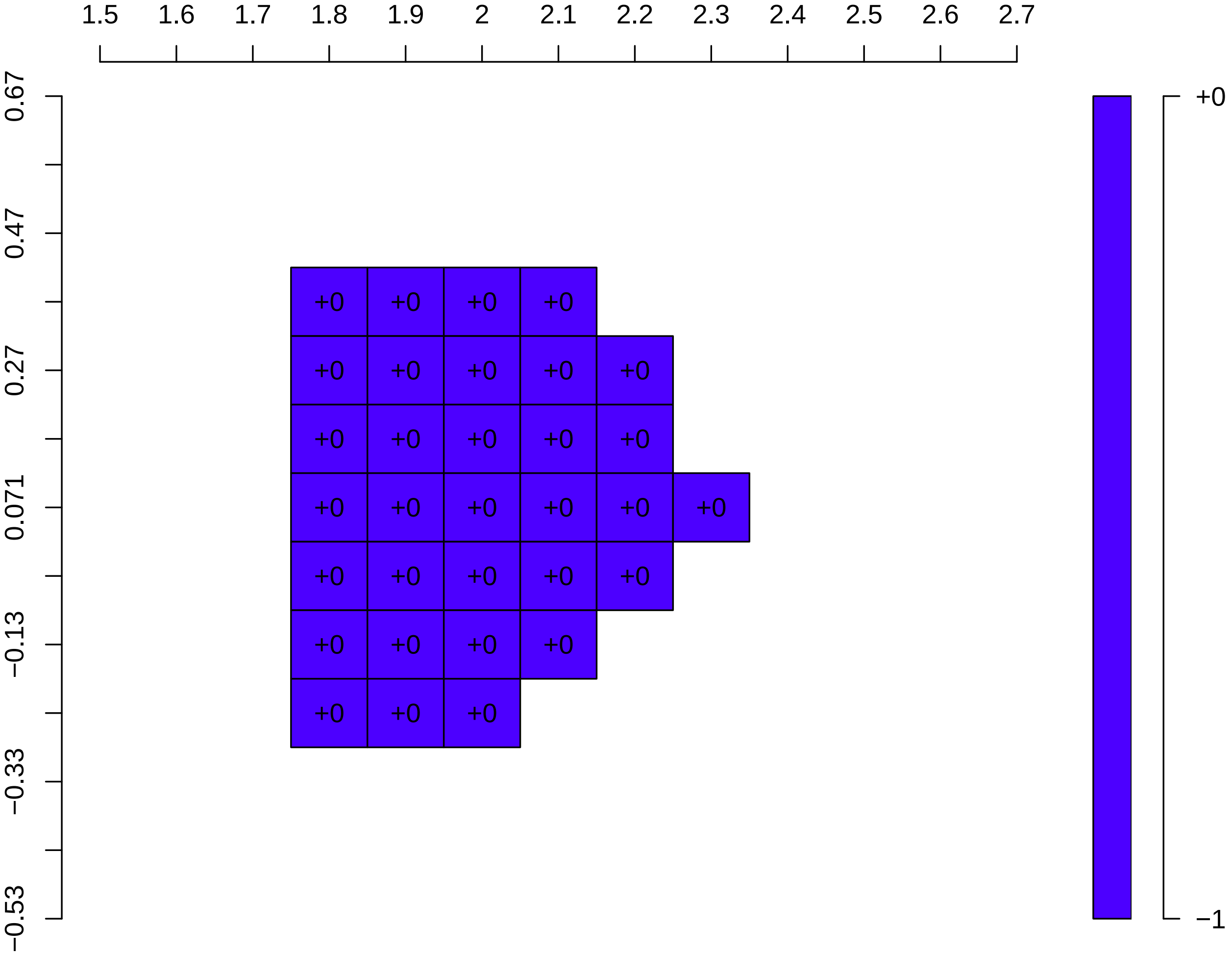
Given these facts, how could the user validate the conclusion of her empirical null procedure?

Keeping the assumption that the null distribution is Gaussian $\mathcal{N}(\theta,\sigma^2)$, we advise to draw a confidence map for $\theta$ and $\sigma$ as described above and as given in Section 6 of (Roquain and Verzelen 2020). For this, we report all the parameters $(\theta,\sigma^2)$ such that $F_0=\mathcal{N}(\theta,\sigma^2)$ is inside the confidence region with coverage, say, 90%. To get more insight, we can display at each point $(\theta,\sigma^2)$ of the region, the number of rejections of the corresponding plug-in BH procedure.

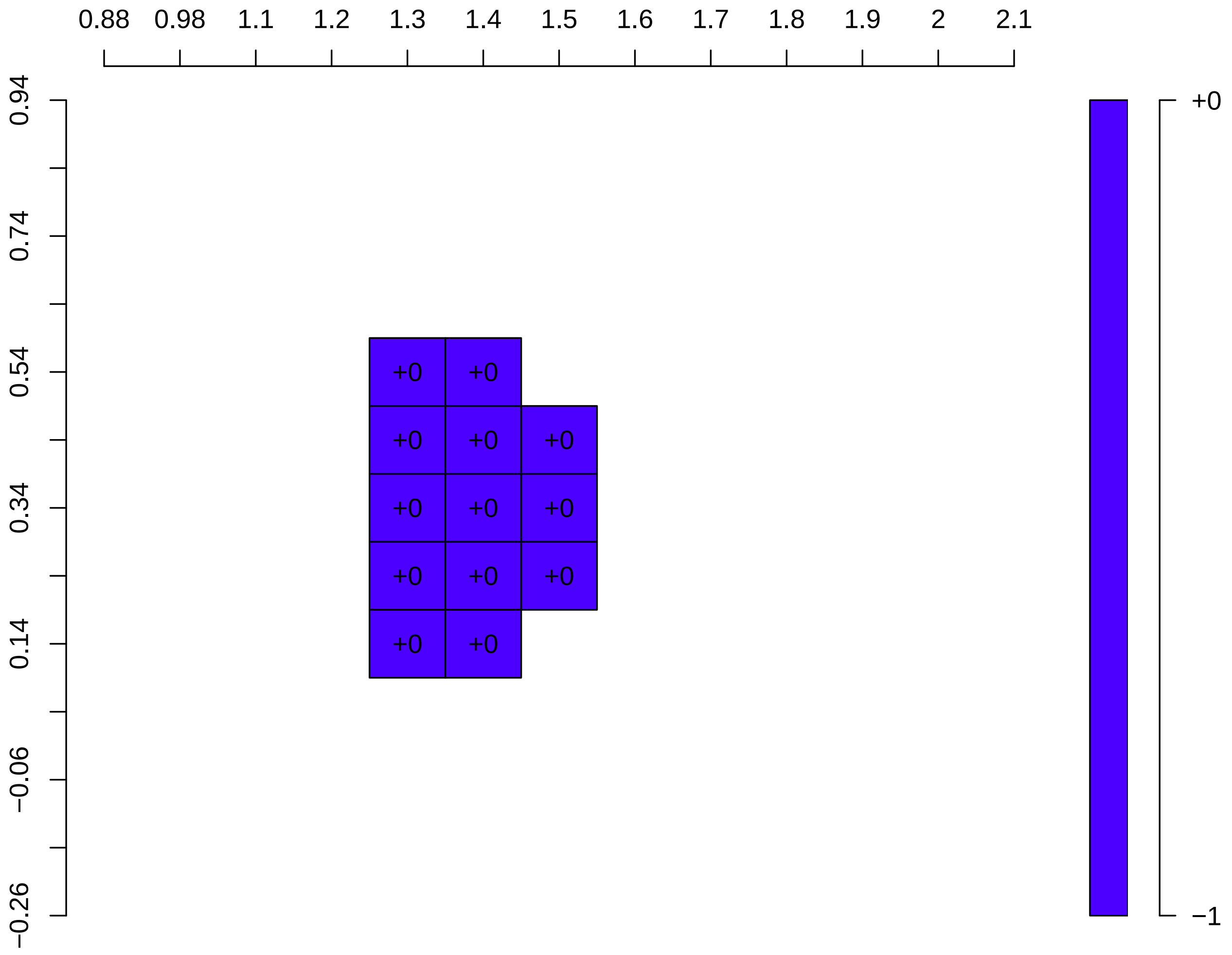
```
getBHrejectionsCR=function(Y,alpha,pibar,thetarange,sigmarange){
  N=length(thetarange)
Admis=matrix(0,N,N)
for (i in 1:N){
  for (j in 1:N){
    Admis[i,j]=NA
    F0=function(x) pnorm((x-thetarange[i])/sigmarange[j])
    if(acceptF0(alpha,F0,pibar,Y)){
      Admis[i,j]= BH((Y-thetarange[i])/sigmarange[j],alpha=0.1)
    }
  }
}
return(Admis)
}
```

On the four above data sets, this parameter confidence region can be displayed as follows:

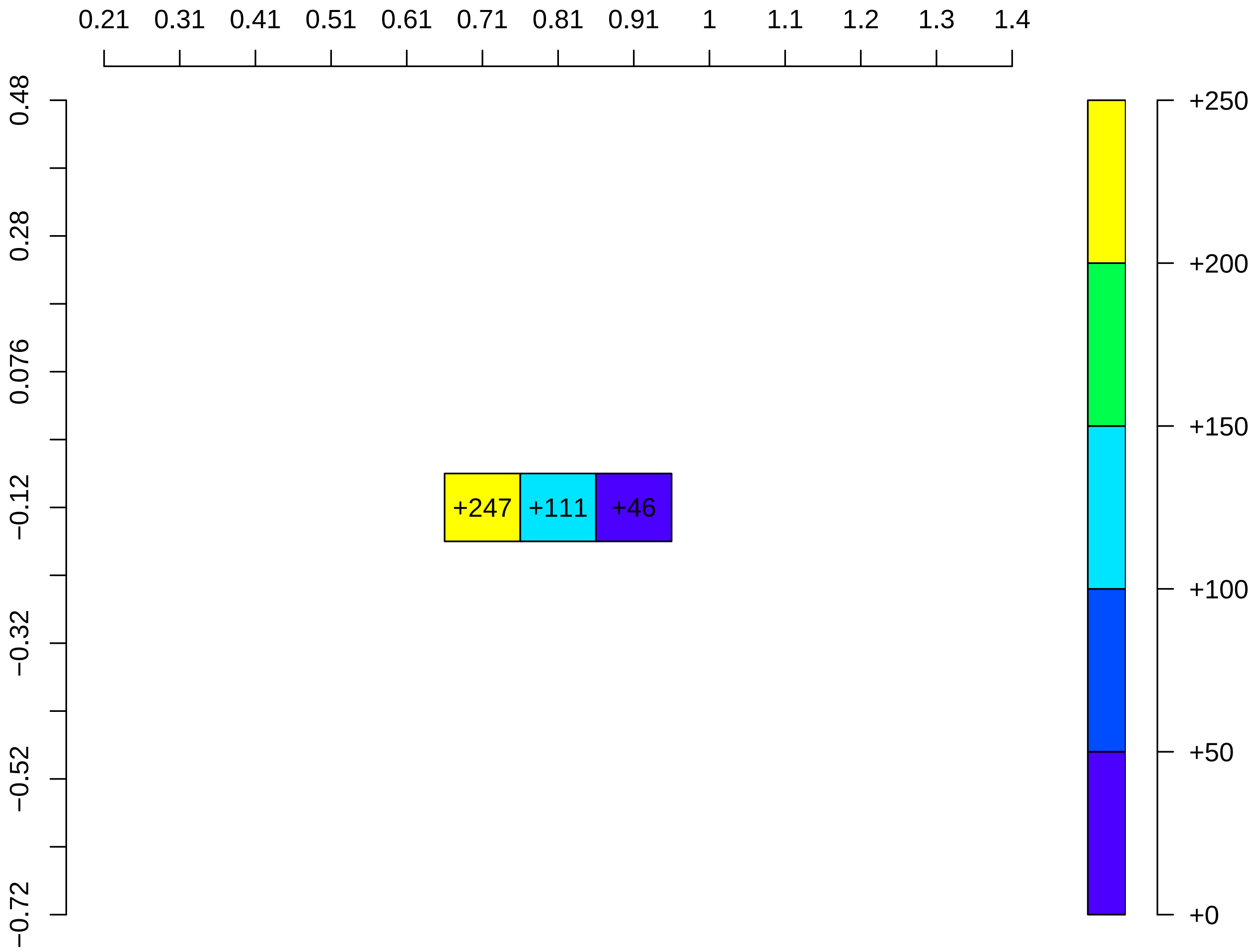
```
plotconfidenceregion=function(Y){
  thetatilde=getthetatilde(Y)
  sigmatilde=getsigmatilde(Y)
  N=13
  thetarange=seq(thetatilde-0.6,thetatilde+0.6,length.out=N)
  sigmarange=seq(sigmatilde-0.6,sigmatilde+0.6,length.out=N)
  Admis=getBHrejectionsCR(Y,alpha=0.1,pibar=0.1,thetarange,sigmarange)
  par(mar=c(0.1, 2.1, 2.1, 4.5))
  plot(Admis[N:1,],xlab="",ylab="",digit=0,
  col=topo.colors,na.cell=FALSE,
  axis.row=axis(side=3,labels=signif(sigmarange,2),at=1:N),
  axis.col=axis(side=2,labels=signif(thetarange,2),at=1:N),line=1,main="")
}

#par(mfrow=c(2,2))
plotconfidenceregion(Ygolub)
```



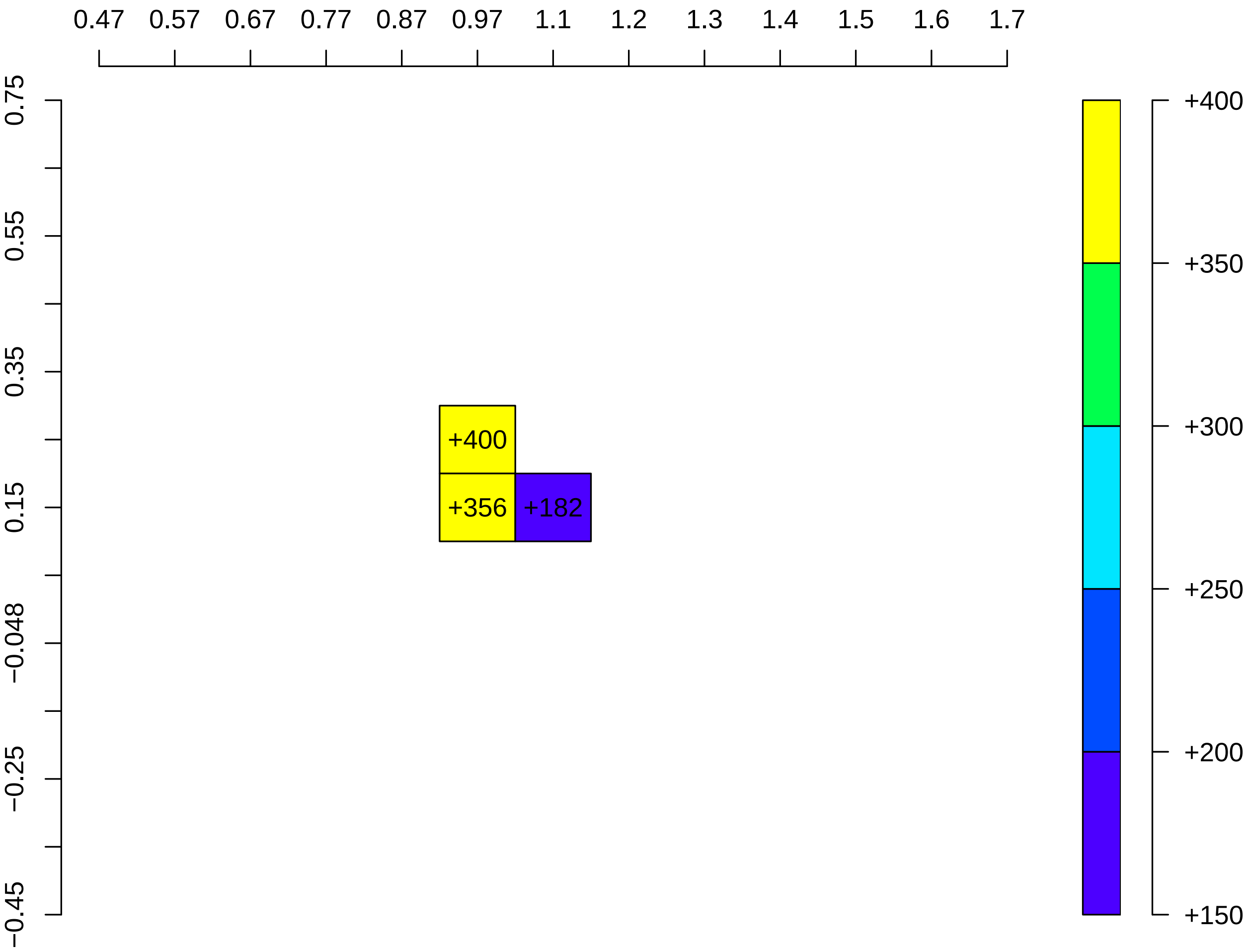
```
plotconfidenceregion(Yheden)
```



```r
plotconfidenceregion(Ywout)
```



```
plotconfidenceregion(YBourgon)
```



In each picture, the confidence region in the scaling $(\theta, \sigma)$ corresponds to the colored pixels. In each of these pixels, the displayed number is the rejection number of the plug-in BH procedure at level $\alpha = 0.1$ using the corresponding scaling.

By definition, the oracle BH procedure belongs to this parameter confidence region with probability at least 90%. Hence, the minimum rejection number of plug-in BH in the confidence region is a lower bound on the best plug-in BH procedure rejection number. If this number is zero or very low, the user should certainly be cautious and declare no variables as significant. If this number is large enough, then the oracle BH procedure promises to find some signal in the data, and thus, the user might apply the plug-in BH procedure (or any other null-Gaussian-based estimation technics, like `locfdr` type algorithm) with more confidence. Alternatively, an extra care could be to declare as significant the variables rejected by *all* the plug-in BH procedures of the region, that is, by considering rejection sets rather than rejection numbers.

Importantly, this method provides an insight different than the minimax approach. The obtained confidence region adapts to the shape of the overall empirical c.d.f. of the measurements. Hence, it is not based on a least favorable configuration (lower bound), but rather accounts for the particular compatibility of the data with respect to the family of Gaussian null distribution.

Following these recommandations on the data sets considered above, the user thus might be cautious for the data Golub et al. (1999) and Hedenfalk et al. (2001) and might declare many findings for the data Wout et al. (2003) and Bourgon, Gentleman, and Huber (2010). These conclusions markedly differ from the ones of the theoretical BH procedure.

To conclude, we have shown in this vignette that the effect of estimating the null distribution can be substantial and lead to very different conclusions from an analysis using the theoretical null. Let us finally note that, in fact, the `null estimation effect` can be even stronger than the `test multiplicity effect` itself. For the data Golub et al. (1999), we have seen that the theoretical BH make 876 discoveries, which are probably all false discoveries according to the above confidence region. Let us now compute the rejection number of the thresholding procedure at level $\alpha = 10\%$, that correctly rescales the data but does not perform any multiple testing correction.

```
nbrejetnocorrection=sum(2*(1-pnorm((Ygolub-getthetatilde(Ygolub))
                                   /getsigmatilde(Ygolub)))<=0.1)
tab <- data.frame("Method" = c("Empirical BH procedure","Theoretical BH procedure",
                               "Non corrected empirical procedure"), "Rejection number" = c(nbrejetgolu
knitr::kable(tab)
```

| Method | Rejection number |
|:---|---:|
| Empirical BH procedure | 0 |
| Theoretical BH procedure | 876 |
| Non corrected empirical procedure | 138 |

While the theoretical BH makes 876 (presumably false) discoveries, the non corrected procedure only makes 138 (presumably false) rejections. From this perspective, if one aims at avoiding false discoveries, the issue of estimating the null can be even more crucial than the issue of taking into account the multiplicity of the tests.

# 4 Session information

```
sessionInfo()
#> R version 3.6.2 (2019-12-12)
#> Platform: x86_64-apple-darwin15.6.0 (64-bit)
#> Running under: macOS Mojave 10.14.6
#>
#> Matrix products: default
#> BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> attached base packages:
#> [1] parallel  stats     graphics  grDevices utils     datasets  methods
#> [8] base
#>
#> other attached packages:
#>  [1] plot.matrix_1.4      locfdr_1.1-8         Equalden.HD_1.2
#>  [4] multtest_2.42.0      Biobase_2.46.0       BiocGenerics_0.32.0
#>  [7] sda_1.3.7            fdrtool_1.2.15       corpcor_1.6.9
#> [10] entropy_1.2.1        sansSouci.data_0.2.0
#>
#> loaded via a namespace (and not attached):
#>  [1] Rcpp_1.0.5      knitr_1.28      magrittr_1.5    MASS_7.3-51.4
#>  [5] splines_3.6.2   lattice_0.20-38 rlang_0.4.7     highr_0.8
#>  [9] stringr_1.4.0   tools_3.6.2     grid_3.6.2      xfun_0.12
#> [13] htmltools_0.4.0 yaml_2.2.1      survival_3.1-8  digest_0.6.25
#> [17] Matrix_1.2-18   evaluate_0.14   rmarkdown_2.1   stringi_1.4.6
#> [21] compiler_3.6.2  stats4_3.6.2
```

# 5 Reproducibility

To re-build this vignette from its source, use:

```
rmarkdown::render("vignette.Rmd", output_format = "pdf_document")
# To keep intermediate files, add option 'clean = FALSE'
rmarkdown::render("vignette.Rmd", output_format = "html_document")
```

# References

Benjamini, Yoav, and Yosef Hochberg. 1995. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.” *J. Roy. Statist. Soc. Ser. B* 57 (1): 289–300.

Bourgon, Richard, Robert Gentleman, and Wolfgang Huber. 2010. “Independent filtering increases detection power for high-throughput experiments.” *PNAS*. doi:10.1073/pnas.0914005107/-/DCSupplemental.www.pnas.org/cgi/doi/10.1073/pnas.0914005107.

Efron, Bradley. 2008. “Microarrays, Empirical Bayes and the Two-Groups Model.” *Statist. Sci.* 23 (1): 1–22. doi:10.1214/07-STS236.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, et al. 1999. “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring.” *Science* 286 (5439). American Association for the Advancement of Science: 531–37. doi:10.1126/science.286.5439.531.

Hedenfalk, Ingrid, David Duggan, Yidong Chen, Michael Radmacher, Michael Bittner, Richard Simon, Paul Meltzer, et al. 2001. “Gene-Expression Profiles in Hereditary Breast Cancer.” *New England Journal of Medicine* 344 (8): 539–48. doi:10.1056/NEJM200102223440801.

Roquain, E., and N. Verzelen. 2020. “False Discovery Rate Control with Unknown Null Distribution: Is It Possible to Mimic the Oracle?” *Submitted*.

Singh, Dinesh, Phillip G Febbo, Kenneth Ross, Donald G Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, et al. 2002. “Gene Expression Correlates of Clinical Prostate Cancer Behavior.” *Cancer Cell* 1 (2). Elsevier: 203–9.

Wout, Angélique B van’t, Ginger K Lehrman, Svetlana A Mikheeva, Gemma C O’Keeffe, Michael G Katze, Roger E Bumgarner, Gary K Geiss, and James I Mullins. 2003. “Cellular Gene Expression Upon Human Immunodeficiency Virus Type 1 Infection of Cd4+-T-Cell Lines.” *Journal of Virology* 77 (2). Am Soc Microbiol: 1392–1402.