Université de Montpellier

Faculté des Sciences

M2 Statistiques et Sciences des Données

TP1 - Expression génétique et données de comptage

HAX004X

Auteurs:
Anas Zakroum

Enseigné par:
Pre. Alice Cleynen

21 février 2023

Table des matières

Prise en main des données

Nous disposons d'un jeu de données dénombre chez 27 souris près de 15 000 gènes à l'aide de leur Ensembl ID. Ces souris ont été divisés en deux groupes, chaque groupe ayant reçu un traitement différent : CD8 (au nombre de 15) et PBS (au nombre de 12).
Chaque ligne du jeu de donnée correspond à un gène particulier. Les individus sont représentés en colonne.

Regardons l'expression pour les deux groupes d'un gêne choisi arbitrairement

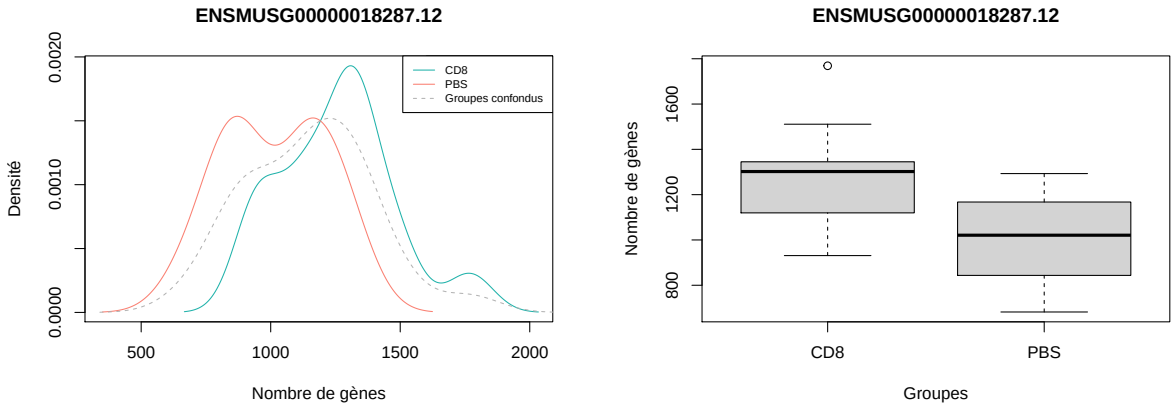

FIGURE 1 – Expression du gène ENSMUSG00000018287.12 chez les deux groupes de souris.

L'expression des gènes varie d'un individu à un autre et semble varier selon les différents groupes pour ce gène. Il existe aussi de la variabilité dans le niveau d'expression entre les gènes. Certains d'ordre de gandeur de dizaines, d'autres de milliers mais aussi de centaines de milliers voir de millions.

Par exemple le gène 203 du jeu de donnée est de faible expression chez les deux groupes.

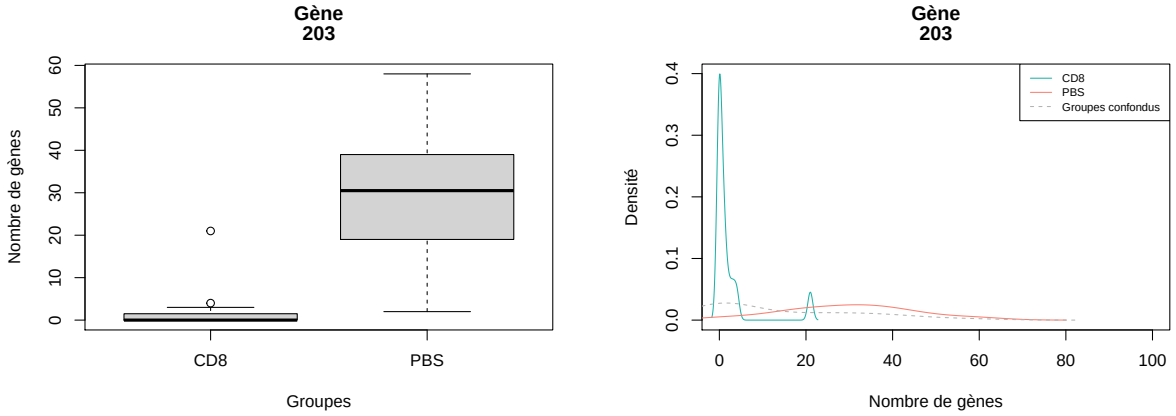

FIGURE 2 – Expression du gène ENSMUSG00000001508.15, numéro 203 chez les deux groupes.

Le gène 11100 est d'expression plutôt forte ou encore le gène 12876 dont l'expression diffère significativement d'un groupe à un autre.

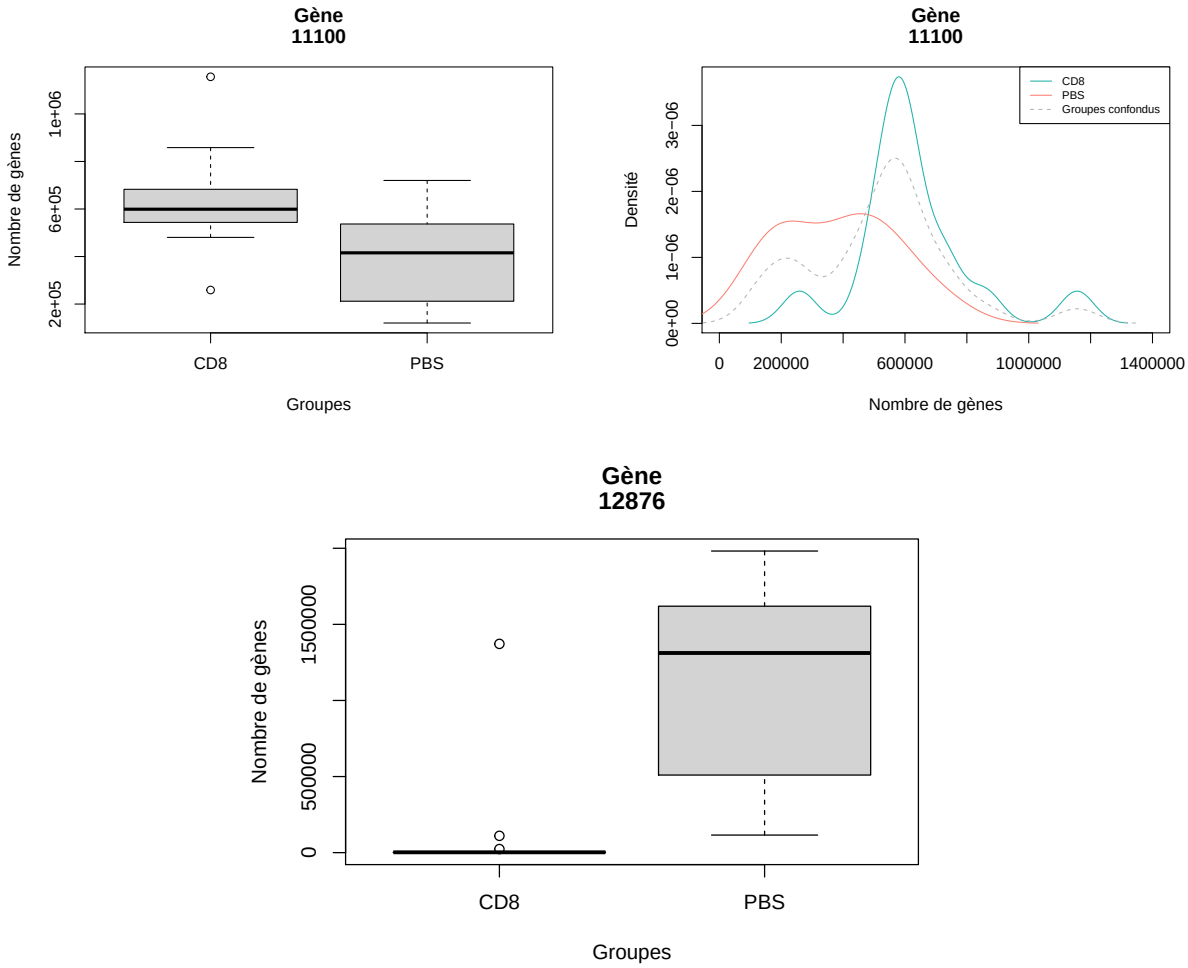

FIGURE 3 – Expression des gènes ENSMUSG00000056071.12 et ENSMUSG00000076613.4, numéros 11100 et 12876.

Le choix des niveaux a été fait sur la base d'une hypothèse biologique et d'un diagnostic naïfs des quantiles de l'expression génétique chez les 27 souris.

L'hypothèse considéré est que l'ordre de grandeur de l'expression d'un gène serait réparti de la même manière chez les individus de même espece (sauf anomalie). Le nombre important de gènes et la grande variabilité dans leur expression nous a poussé à considérer la médiane plutôt que la moyenne pour décider de ce qu'est un niveau d'expression moyen. Les niveaux d'expression faible et fort ont été choisi sur la base d'une trop grande deviance du seuil indiqué par la médiane. Un gène d'expression moyenne est comme celui présenté à la toute première figure.

Le gène 12876 remet en cause l'hypothèse biologique considérée : boite à moustache de la Figure 3.
Le traitement aurait il un effet sur l'expression génétique ?

1 Loi de Poisson

Nous nous intéressons au modèle suivant décrivant le comptage du gène g pour l'individu i :

$$Y_{gi} \sim \mathcal{P}(\mu_{gi}) \tag{1}$$

Avec

$$\log \mu_{gi} = \alpha_g + \beta_g \log N_i.$$

La loi de Poisson appartient à la famille exponentielle, la fonction de lien naturelle est $\eta \mapsto e^{\eta}$.

Nous avons ici une dépendance à l'echelle log du nombre d'ARN-m pour le gène $g$en fonction du nombre total de reads pour le même individu.
Le paramètre α_g représente la pente à l'origine de la relation linéaire reliant μ_g et N_i (à l'échelle log). Il joue le rôle d'un seuil de base auquel on s'attend à voir grandir au fur et à mesure qu'on lit de reads (N_i).

Dans un contexte de comparaison de l'effet d'un traitement sur les données de comptage, on s'attend à ce que le traitement (si il a un effet) agisse seulement sur le seuil de base et non sur la quantité de départ de reads.
En considérant l'appartenance à des groupes différents, donc un effet seulement sur : α_g et non le β_g qui n'est pas sensé être dépendant du traitement, mais de la quantité de reads.

Ainsi le modèle devient

$$Y_{gij} \sim \mathcal{P}(\mu_{gij}) \tag{2}$$

Avec

$$\log \mu_{gij} = \alpha_{gj} + \beta_g \log N_{ij}.$$

où $j \in \{1, 2\}$ désignant les groupes PBS et CD8.

Afin de rendre le paramètre π reliant le nombre de reads du gène g au nombre de reads total par la $\mu_{gi} = N_i \pi_g$ (et représentant la fraction réelle d'ARN provenant du gène g) plus lisible, nous procédons au changement d'échelle suivant :

$$\tilde{N}_i = \log\left(N_i \times 10^{-6}\right).$$

id	664	665	...
N_i	2.744×10^7	1.899×10^7	...
$\tilde{N}_i$	3.312196	2.944138	...

TABLE 1 – Transformation de l'échelle du nombre total de reads.

Nous implémentons dans ce qui suit le modèle linéaire généralisé (1) pour le gène d'expression moyenne numéro 11100 sans prendre en compte le traitement

TABLE 2 – Résultats de la régression de Poisson du modèle (1), groupes confondus.

	Dependent variable :
	exmed
$\hat{\beta}_g$	1.022^{***}
	(0.029)
$\hat{\alpha}_g$	3.833^{***}
	(0.093)
Observations	27
Log Likelihood	−193.943
Akaike Inf. Crit.	391.886
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

Regardons maitenant chaque groupe séparément à l'aide du modèle(2). Les résultats de la régression donnent :

	Dependent variable			
	Exmed CD8			Exmed PBS
$\hat{\beta}_{g,CD8}$	0.964***		$\hat{\beta}_{g,PBS}$	1.221***
	(0.042)			(0.065)
$\hat{\alpha}_{g,CD8}$	4.018***		$\hat{\alpha}_{g,PBS}$	3.239***
	(0.138)			(0.197)
Observations	15			12
Log Likelihood	-93.392			-94.617
Akaike Inf. Crit.	190.784			193.235
Note	*p<0.1 ; **p<0.05 ; ***p<0.01			

TABLE 3 – Estimation des coefficients du modèle (2) visants à comparer les deux groupes pour le gène g

Les probabilités critiques sont toutes significatives au seuil de 1% pour le test de nullité des coefficients, regardons les intervalles de confiance :

	$\hat{\alpha}_g$	$\hat{\beta}_g$
Groupes confondus	[3.651, 4.015]	[0.965, 1.079]
PBS	[2.852, 3.623]	[1.093, 1.348]
CD8	[3.748, 4.288]	[0.881, 1.047]

TABLE 4 – Intervalles de confiance à 95% des estimateurs pour les deux modèles

Observation

Pour le seuil de base $\hat{\alpha}_g$, nous observons une difference statistiquement significative (bien que la borne supérieur PBS soit proche de la borne inférieure CD8). Pour $\hat{\beta}_g$, les IC se chevauchent.

Pour revenir à la relation $\log(\mu_{gj}) = \alpha_{gj} + \beta_g N_i$, ceci nous indique une pente équivalente quelque soit le groupe, mais une ordonnée à l'origine différente pour chaque groupe.
Le traitement à t il alors un effet ? Pour ce gène les niveaux d'expression pour chaque groupe sont très proche, donc peut être pas pour ce gène.

Question 6 :

Nous modifions le modèle en rajoutant une covariable désignant l'appartenance à un groupe et entrainons le nouveau modèle généralisé dans le but d'estimer ses paramètres. Nous cherchons un modèle sans interaction.
On utilise `glm(exp~TC + Treatment, family = 'poisson', data=df)`

La paramétrisation de `R` désigne une modalité de réference pour chaque variable qualitative. D'après les résultats de la régression de la Table **??** ci-dessous, la modalité de référence est le groupe CD8. Voici les résultats obtenus.

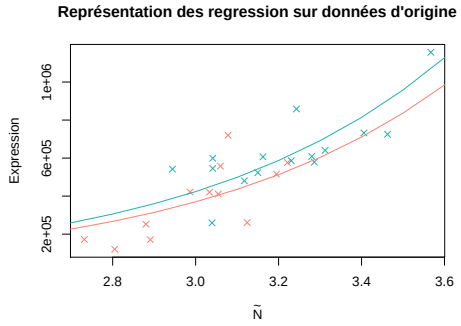

FIGURE 4 – Représentation de la regression avec données d'origine.

	Dependent variable :
	exp
TC	1.631***
	(0.002)
TreatmentPBS	−0.136***
	(0.001)
Constant	8.063***
	(0.005)
Observations	27
Log Likelihood	−461,526.700
Akaike Inf. Crit.	923,059.300
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

TABLE 5 – Résultats de la régression du modèle avec la covariable de groupe.

Pour le gène 12876, nous trouvons une différence dans l'expression selon le traitement.

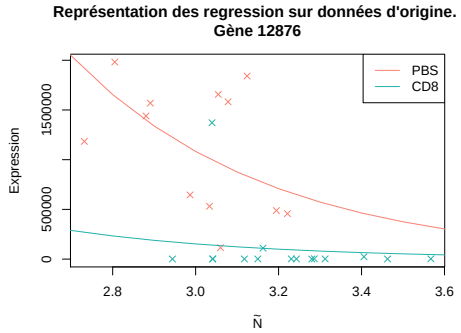

FIGURE 5 – Représentation de la regression avec données d'origine.

	Dependent variable :
	exp
TC	−2.120***
	(0.002)
TreatmentPBS	1.959***
	(0.001)
Constant	18.295***
	(0.006)
Observations	27
Log Likelihood	−5,010,509.000
Akaike Inf. Crit.	10,021,024.000
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

TABLE 6 – Résultats de la régression du modèle avec la covariable de groupe.

Question 7

Nous sommes en présence de deux modèles emboités et souhaitons savoir quel est le modèle le plus adapté. On teste alors la nullité de la covariable Traitement, qui est présente dans un seul des deux modèles.

```
Model 1: exp ~ TC + Treatment
Model 2: exp ~ TC
```

Nous procédons à un test de rapport de vraisemblance.

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	24	922651.40			
2	25	962581.80	-1	-39930.40	$< 2.2\ e^{-16}$

Au regard de la probabilité critique et du seuil de deviance, nous rejettons l'hypothèse de l'égalité des modèles.

2 Loi binomiale négative

Il est possible de modéliser les données d'expression par une loi binomiale négative via le modèle suivant :

$$Y_{gij} \sim \mathcal{NB}(\mu_{gij}, \phi_g) \tag{3}$$

Avec

$$\log \mu_{gij} = \mu_{gj} + \beta_g \log N_{ij}.$$

Question 9 :

μ_{gij} correspond au nombrede reads pour le gène g d'un individu i appartenant au groupe j.
N_{ij} représente le nombre de reads total pour l'individu i appartenant au groupe j.

Le modèle suppose une relation linéaire à l'echelle logarithmique entre le nombre de reads μ_{gij} d'un gène g qu'on "s'attend" à lire pour un individu i appartenant au groupe j et le nombre total de reads N_{ij}.

Le modèle se base sur l'hypothèse biologique que le nombre moyen de reads d'un gène g dépend non seulement d'un seuil de base correspondant au gène mais aussi du groupe dans lequel se trouve l'individu : μ_{gj} et dont l'accroissement dépend du nombre de reads de départ et d'une proportion propre au gène : β_g qui caracterise la pente.

Enfin ϕ_g correspond au coefficient de dispersion de la binomiale négative.

Question 10 : En choisissant le groupe PBS comme référence, nous trouvons

TABLE 7 – Résultats de la régression du modèle (3).

	Dependent variable :
	exp
TC	1.244***
	(0.163)
Constant	3.169***
	(0.491)
Observations	12
Log Likelihood	−70.685
θ	178.727** (84.937)
Akaike Inf. Crit.	145.369
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

Question 11 :

Notre modèle `glm.nb()` est stocké dans l'objet `fit_binomial`.

Pour calculer la valeur de ϕ_g, on calcule l'inverse de `fit_binomial\$theta`.

Question 12 :

On implémente le modèle pour le gène d'expression faible numéro 203

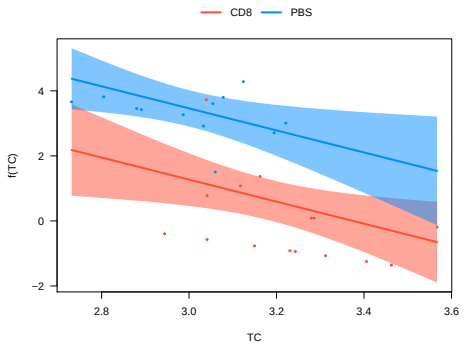

FIGURE 6 – Représentation de la regression

	Dependent variable :
	exp
TC	−3.391**
	(1.422)
TreatmentPBS	2.191***
	(0.500)
Constant	11.444**
	(4.537)
Observations	27
Log Likelihood	−81.213
θ	1.062*** (0.410)
Akaike Inf. Crit.	168.426
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

TABLE 8

Nous observons une différence statistiquement significative. Le gène est différentiellement exprimé entre les groupes.

Question 13 :
Nous reprenons la même question pour le gène d'expression forte numéro 11100.

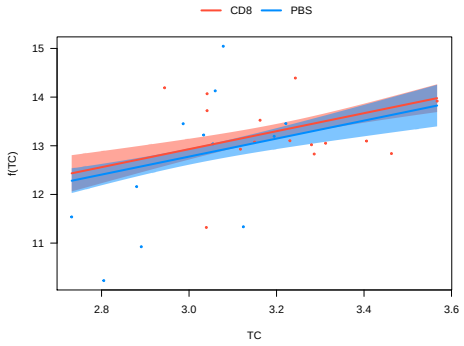

FIGURE 7 – Représentation de la regression pour le gène d'expression forte.

	Dependent variable :
	exp
TC	1.849***
	(0.358)
TreatmentPBS	−0.153
	(0.137)
Constant	7.384***
	(1.154)
Observations	27
Log Likelihood	−358.911
θ	11.584*** (3.109)
Akaike Inf. Crit.	723.822
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

TABLE 9 – Résultats de la régression pour le gène d'expression forte

Pour ce gène, nous n'observons pas une différence statistiquement significative. Pas de différence dans les expression de ce gène pour les deux groupes.

3 Tests multiples

On souhaite tester l'expression différentielle des gènes les plus fortment exprimés.

Question 14 :

Il est important d'effectuer une correction aux tests multiples plus il y a d'hypothèses à tester, plus le risque d'erreur grandit ;

Supposons qu'on dispose de n hypothèses à tester avec un niveau de signification de $\alpha = 5\%$. La probailité d'observer au moins un résultat significatif est :

$$\begin{aligned}\mathbb{P}(\text{au moins un résultat significatif}) &= 1 - \mathbb{P}(\text{aucun résultat significatif}) \\ &= 1 - (1 - 0.05)^n\end{aligned}$$

Pour $n = 13$ hypothèses à tester, $\mathbb{P}(\text{au moins un résultat significatif}) = 0.51$. Ce qui veut dire que même si aucun résultat n'est réellement significatif, la probabilité d'en trouver un qui l'est, relève du hasard. Cette probabilité augmente d'autant plus que le nombre d'hypothèses à tester est important.

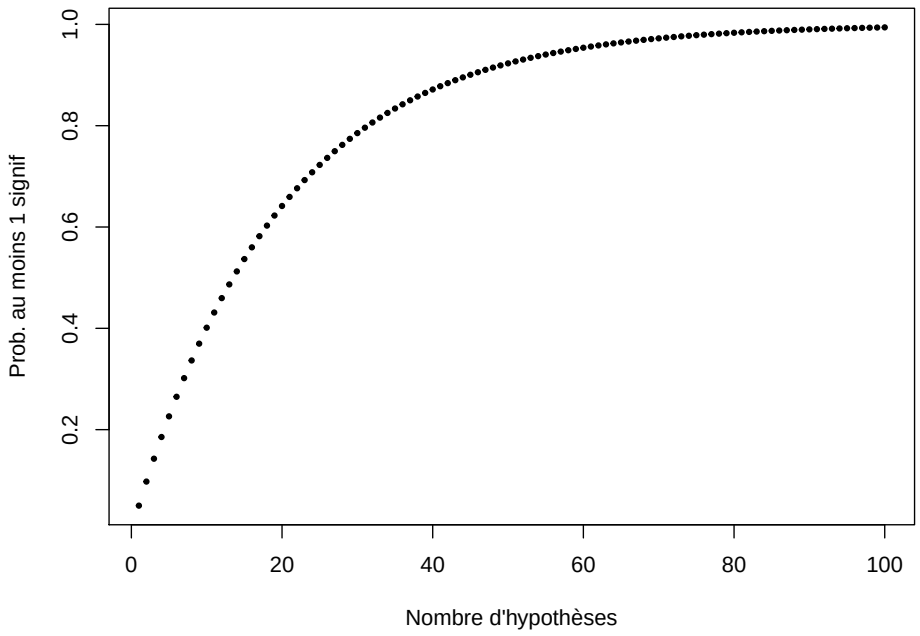

FIGURE 8 – Évolution des probabilités de trouver un résultat significatif en fonction du nombre d'hypothèses à tester.

En génomique, on peut se retrouver à tester un nombre d'hypothèses très grand. D'où l'importance de procéder à des corrections.

Question 15 :

Implémentation de la procédure de Benjamini et Hochberg pour une liste donnée de p-valeurs, et un niveau de contrôle α.

```
bh<- function(pval, alpha){
  pval_size <- length(pval)
  pval_ordered <- sort(pval)
  treshold <- alpha * c(1:pval_size)/pval_size
  x <- which(pval_ordered < treshold)
  x <- max(x)
  selected <- c()
  if(x > 0){
    selected <- order(pval)[1:x]
  }
  return(selected)
}
```

Question 16 :
Le gène d'expression forte fait partie de la liste. La fonction pvalue_NB pour le gène 11100 donne une probabilité critique de 0.24.

Question 17 :
Notre gène ne figure pas dans la liste `bcor`. Au regard de la probabilité critique, il n'est pas différentiellement exprimé.

Question 18 :

— Avant correction : 665 gènes différentiellement exprimés.
— Avec procédure Bonferroni : 23 gènes différentiellement exprimés.

Les méthodes Bonferroni, Hochberg et Hommel sont beaucoup plus strictes d'après la Figure 9.

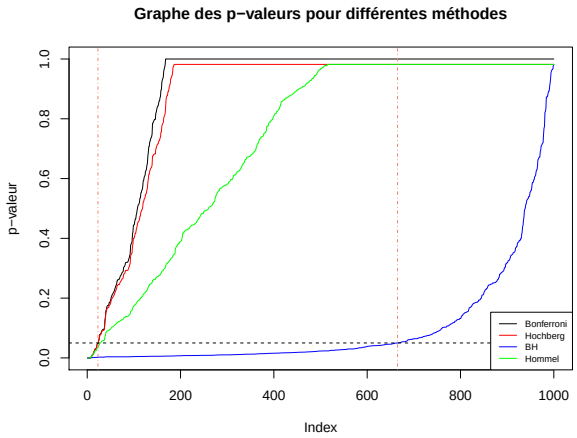

FIGURE 9 – Graphique des p-valeurs par différentes méthodes de correction

4 Comparaison des modèles

Question 19

Les modèles de Poisson et Binomiale négative sont des modèles emboités. Le modèle de Poisson est un cas particulier de la Binomiale négative. Pour cela un test du rapport de vraisemblance peut être effectué.

Annexe

Code en annexe

```
library(stargazer)

cd8 <-which(Design == "CD8")
pbs <-which(Design == "PBS")

random_gene_cd8  <- as.numeric(matrixcounts[vect_of_genes[3],cd8])
random_gene_pbs  <- as.numeric(matrixcounts[vect_of_genes[3],pbs])
conf_dsty <- density(as.numeric(matrixcounts[vect_of_genes[3],]))

cd8_dsty <- density(random_gene_cd8)
pbs_dsty <- density(random_gene_pbs)


plot(cd8_dsty$x, cd8_dsty$y, col = "lightseagreen",
     type = 'l',
     main = paste(c("Gène", vect_of_genes[3]), sep = ""), xlab = "Nombre de gènes",
     ylab = "Densité")

plot(pbs_dsty$x, pbs_dsty$y, col = "lightseagreen",
     type = 'l',
     main = paste(c("Gène", vect_of_genes[3]), sep = ""), xlab = "Nombre de gènes",
     ylab = "Densité")

lines(pbs_dsty$x, pbs_dsty$y, col = "salmon")
lines(conf_dsty$x,conf_dsty$y, lty = 2, col = "gray70")
legend(x = "topright", col = c("lightseagreen", "salmon", "gray70"),
       legend = c("CD8", "PBS", "Groupes confondus"), lty = c(1,1,2), cex = 0.7)


# boxplot
r_gene <- as.numeric(matrixcounts[vect_of_genes[3],])
df_random <- data.frame(r_gene,Design)
boxplot(df_random$r_gene ~ df_random$Treatment,
        xlab = "Groupes", ylab = "Nombre de gènes", main = paste(c("Gène", vect_of_ge


vect_of_genes <- c(203, 11100, 11235)
# forte 1110
# faible 203
# interessant 12876


# question 3
Nis <- apply(matrixcounts, 2, sum)
Ni_tilde <- log(Nis*(10^(-6)))
```

```r
TableM=data.frame(exp=ExMed,TC=Ni_tilde,Group=Design)

# question 4
Ni_tilde_cd8 <- Ni_tilde[cd8]
Ni_tilde_pbs <- Ni_tilde[pbs]
exmed <- as.numeric(ExMed)
exmed_pbs <- exmed[pbs]
exmed_cd8 <- exmed[cd8]
pois_fit <- glm(exmed ~ Ni_tilde, family = poisson)
pois_cd8 <- glm(exmed_cd8 ~ Ni_tilde_cd8, family = poisson)
pois_pbs <- glm(exmed_pbs ~ Ni_tilde_pbs, family = poisson)

## IC
ci_all <- confint(pois_fit)
ci_cd8 <- confint(pois_cd8)
ci_pbs <- confint(pois_pbs)




######
## Question 6
######

# glm
ExHigh <- as.numeric(matrixcounts[11100,])
df <- data.frame(exp=ExHigh, TC=Ni_tilde,Treatment=Design)
fit_ni <- glm(exp~TC+Treatment, family = 'poisson', data=df)


# graphique
x <- seq(0.5,5,by=0.1)
y1 <- fit_ni$coefficients[1] + fit_ni$coefficients[2] * x
y2 <- fit_ni$coefficients[1] + fit_ni$coefficients[3] + fit_ni$coefficients[2]*x

plot(exp~Ni_tilde, data = df,
     pch=4,col=ifelse(Design[,1]=="PBS", "salmon", "lightseagreen"),
     ylab = "Expression", xlab= expression(tilde(N)),
     main = "Représentation des regression sur données d'origine. \n Gène 12876 ")
lines(x, exp(y1), col="lightseagreen")
lines(x, exp(y2), col="salmon")
legend("topright", c("PBS","CD8"), col=c("salmon","lightseagreen"), lty=1)


######
## Question 7
```

```
######



pois_fit <- glm(exp~TC, family = 'poisson', data=df)

anova(fit_ni, pois_fit,test = "LRT")


######
## Question 10
######

df_med <- data.frame(exp=exmed, TC=Ni_tilde,Treatment=Design)
fit_binomial <- glm.nb(exp~TC, data=df_med, subset = (Treatment == "PBS"))
summary(fit_binomial)

# phi_g
dispersion <- fit_binomial$theta
dispersion <- 1/dispersion


######
## Question 12
######
exlow <- as.numeric(matrixcounts[203,])
df <- data.frame(exp=exlow, TC=Ni_tilde,Treatment=Design)
fit_bin_low <- glm.nb(exp~TC+Treatment, data=df)
summary(fit_bin_low)



visreg(fit_bin_low, by="Treatment", xvar="TC", overlay=T)


######
## Question 13
######
exhigh <- as.numeric(matrixcounts[11100,])
df <- data.frame(exp=exhigh, TC=Ni_tilde,Treatment=Design)
fit_bin_high <- glm.nb(exp~TC+Treatment, data=df)
summary(fit_bin_high)
confint(fit_bin_high)
visreg(fit_bin_high, by="Treatment", xvar="TC", overlay=T)

bh<- function(pval, alpha){
  pval_size <- length(pval)
  pval_ordered <- sort(pval)
```

```
  treshold <- alpha * c(1:pval_size)/pval_size
  x <- which(pval_ordered < treshold)
  x <- max(x)
  selected <- c()
  if(x > 0){
    selected <- order(pval)[1:x]
  }
  return(selected)
}


######
## Question 15
######

hyp_test <- function(alpha, number_of_hypotheses){
        res <- 1 - (1-alpha)^number_of_hypotheses
        return(res)
}

curve(hyp_test(0.05,x), from = 0,to = 100, xlab =)

vect_of_n <- 1:100
prob <- hyp_test(0.05,vect_of_n)
plot(vect_of_n, prob, type = 'b', pch = 19, xlab ="Nombre d'hypothèses", ylab = "Prob
hyp_test(0.05, 13)



######
## Question 16
######



p_values <- sapply(HighList,pvalue_NB)
# 836
p_values[which(HighList==11100)]


######
## Question 17
######

alpha <- 0.05
bcor <- bh(p_values,alpha)
bcor[836]
which(bcor == 836)
```

```r
######
## Question 18
######

plot(sort(p.adjust(p_values, method = "bonferroni")),ylab = "p-valeur", pch = 19, cex
     main = "Graphe des p-valeurs pour différentes méthodes")
lines(1:length(sort(p.adjust(p_values, method = "hochberg"))), sort(p.adjust(p_values
lines(1:length(sort(p.adjust(p_values, method = "BH"))), sort(p.adjust(p_values, metho
lines(1:length(sort(p.adjust(p_values, method = "hommel"))), sort(p.adjust(p_values, m
abline(h = 0.05, lty = 2)

abline(v = 665, lty = 4, col = "salmon")
abline(v = 23, lty = 4, col = "salmon")
legend(x = "bottomright", col = c("black", "red", "blue", "green"),
      legend = c("Bonferroni", "Hochberg", "BH", "Hommel"), lty = c(1,1,1,1), cex = 0
```