Victor Vannobel Alexandre Verzura

Introduction aux tests multiples et application à des données d'expression de gènes

Mai 2023

Merci madame Duval pour tout le temps que vous nous avez accordé.

Introduction aux tests multiples et application à des données d'expression de gènes

VANNOBEL Victor, VERZURA Alexandre

Mai 2023

Table des matières

Introduction

Ce document traite des résultats de recherche sur les méthodes correctives pour des tests multiples. L'objectif de cette étude est de présenter les problèmes rencontrés lors de réalisations de tests multiples puis d'apporter des solutions pour contrôler les différentes erreurs possibles lors de ces tests. Pour ce faire, on se place dans le cadre de l'étude du génome humain. La problématique est donc la suivante : comment définir si un gène du génome humain est codant ou non.

---

1 Rappels sur les tests

Un test sur un paramètre inconnu θ, lié à des observations, est une procédure statistique permettant de rejeter ou non une hypothèse sur θ avec une probabilité d'erreur controlée.

1.1 Notions et définitions

Soit un n-échantillon $(X_1,...,X_n)$ tel que les X_i sont indépendants et identiquement distribués (noté i.i.d.) selon la loi $\mathbb{P}_\theta$. Pour construire un test sur cet échantillon il faut dans un premier temps définir les hypothèses confrontées.
L'hypothèse nulle $H_0 : \theta \in \Theta_0$ contre $H_1 : \theta \in \Theta_1$ avec $\Theta_0 \cap \Theta_1 = \emptyset$.
Afin d'obtenir de l'information sur le paramètre inconnu, il convient de créer un estimateur convergent vers la vraie valeur du paramètre θ.
Rappelons la définition de la convergence en loi : On dit qu'une suite de variables aléatoires $(X_n)_{n\geq 1}$ à valeurs dans un espace métrique (E,d) converge en loi vers X si pour toute fonction continue bornée ϕ de E dans $\mathbb{R}$, $\lim\limits_{n\to+\infty} \mathbb{E}[\phi(X_n)] = \mathbb{E}[\phi(X)]$.
Ensuite, on construit une statistique de test T liée à l'estimateur de θ c'est à dire une variable aléatoire définie comme une fonction mesurable d'un n-échantillon : $T = T_n = g_n(X_1,...,X_n)$.
La statistique est construite telle que son comportement est différent sous H_0 et sous H_1.
Ainsi, nous pouvons construire une zone de rejet pour T de niveau α notée R_α, qui est une partie de l'espace probabilisable telle que $\mathbb{P}_{H_0}(T \in R_\alpha) \leq \alpha$.
Enfin, si notre statistique observée notée T_{obs} appartient à la zone de rejet R_α on rejette l'hypothèse H_0 et on conlut que l'hypothèse H_1 est vraie.

Un test statistique admet deux types d'erreurs distinctes dites de première et de seconde espèce. L'erreur de première espèce α correspond au niveau du test.
L'erreur de seconde espèce $\beta = \mathbb{P}_{H_1}(T \notin R_\alpha)$ correspond à la probabilité de conclure à tord H_0.

Exemple concret

Prenons un exemple concret pour mieux visualiser le principe du test :
Soit $(X_1,...,X_n)$ un n-échantillon i.i.d. de loi $\mathcal{N}(\theta,\sigma^2)$ avec θ le paramètre inconnu et σ connu.
On a donc $\forall i \in [\![1,n]\!]$ $X_i \in \mathbb{R}$ et $\theta \in \mathbb{R}$.
Nous voulons confronter les hypothèses $H_0 : \theta = 0$ contre $H_1 : \theta > 0$.
Ici, $\Theta_0 = \{0\}$ et $\Theta_1 =]0,+\infty[$ (on note que $\Theta_0 \cap \Theta_1 = \emptyset$ mais que $\Theta_0 \sqcup \Theta_1 \neq \mathbb{R}$).
Nous allons dans un premier temps estimer le paramètre θ inconnu. On a d'après la loi forte

des grands nombres : $\overline{X_n} = \sum_{i=1}^{n} X_i \underset{n\to+\infty}{\to} \mathbb{E}(X_1) = \theta$. $\overline{X_n}$ est un bon estimateur du paramètre inconnu θ car il converge (ici presque sûrement) vers ce dernier. On sait que $T_n = \sqrt{n}\frac{\overline{X_n}-\theta}{\sqrt{V_n^*}} \sim T$ sous H_0 où $T \sim \mathcal{T}(n-1)$ et $V_n^* = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X_n})^2$ est la variance empirique corrigée.

On pose $T_n = \sqrt{n}\frac{\overline{X_n}-\theta_0}{\sqrt{V_n^*}} = \sqrt{n}\frac{\overline{X_n}}{\sqrt{V_n^*}}$. En remarquant que la statistique T_n a tendance à être plus grande sous l'hypothèse H_1 plutôt que sous H_0, on a une zone de rejet de la forme $R_\alpha = \{T_n \geq k_\alpha\}$.

En choisissant $q_{\alpha,n-1}$ tel que $\mathbb{P}(T \geq q_{\alpha,n-1}) \leq \alpha$, il vient $\mathbb{P}_{H_0}(\sqrt{n}\frac{\overline{X_n}}{\sqrt{V_n^*}} \geq q_{\alpha,n-1}) \leq \alpha$. et $R_\alpha = \left\{T_n \geq q_{\alpha,n-1}\right\}$.

En prenant un échantillon de taille n=10 et un niveau $\alpha = 0.05$ pour des $X_i \sim \mathcal{N}(\theta, \sigma^2 = 4)$ on obtient le quantile $q_{0.05,9} = 1.833$. Finalement, $T_n \in R_\alpha \Leftrightarrow \overline{X_n} \geq \frac{1.833\times\sqrt{V_n^*}}{\sqrt{n}} \simeq 0.58\sqrt{V_n^*}$

Si $T_n \in R_\alpha$ on rejette H_0 et on conclut H_1.

1.2 P-valeur et propriétés

Cette partie a pour but d'introduire et de visualiser ce qu'est une p-valeur. Ainsi nous étudierons dans un premier temps la définition de la p-valeur et son utilité dans la prise de décision d'un test.

1.2.1 Définition

Soit T une statistique et $\alpha \in]0,1[$ pour tester l'hypothèse H_0 contre l'hypothèse alternative H_1. On note R_α la zone de rejet du test.

On a observé T_{obs}, on définit la p-valeur comme suit : p-val $= \inf\left\{\alpha \in]0,1[,\ T_{obs} \in R_\alpha\right\} = \alpha^*$.

La p-valeur d'un test est le plus petit niveau pour lequel on rejette H_0.

Ainsi, pour $\alpha \in]0,1[$, si $\alpha^* \leq \alpha$ alors on rejette H_0, sinon, $\alpha < \alpha^*$ et on ne rejette pas H_0.

On peut interpréter les p-valeurs obtenues de manière suivante :

$\alpha^* \leq 0.01$, très forte présomption contre H_0

$0.01 < \alpha^* \leq 0.05$, forte présomption contre H_0

$0.05 < \alpha^* \leq 0.1$, faible présomption contre H_0

$0.1 < \alpha^*$, peu ou pas de présomption contre H_0

On rappelle que la majorité des tests sont effectués avec des valeurs de risque α qui varient entre 0.01 et 0.1.

1.2.2 Méthode de calcul pour une loi symétrique

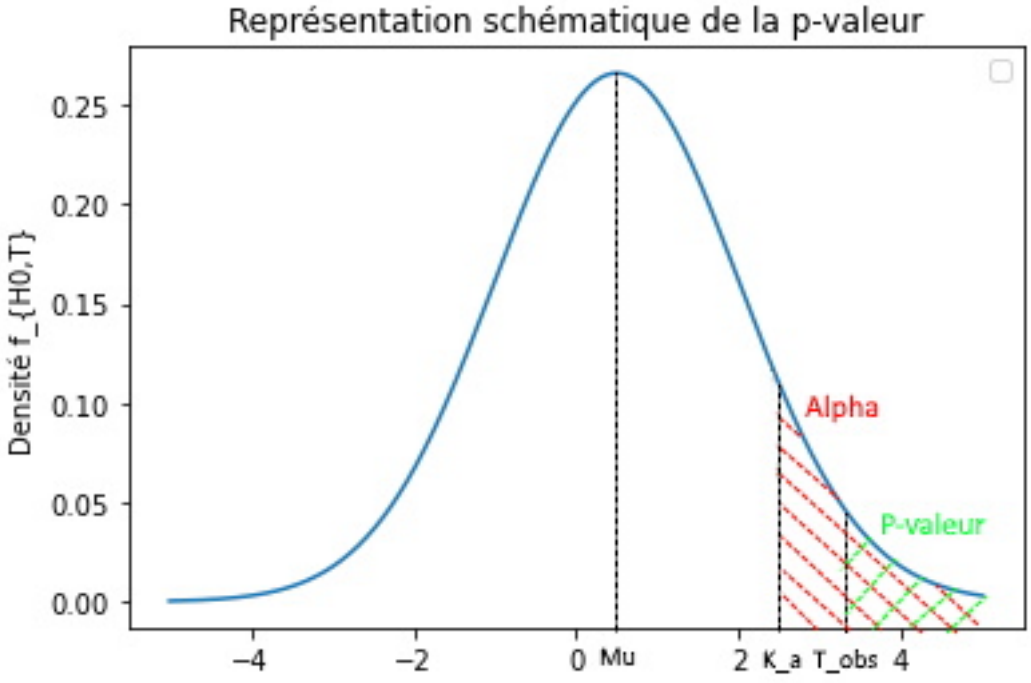
Soit $X_1, \ldots, X_n$ un n-échantillon et T une statistique de loi f symétrique.

La zone de rejet pour un test bilatéral au niveau α s'écrit $R_\alpha = \left\{|T| \geq k_{\alpha/2}\right\}$ où $\mathbb{P}\left(T \geq k_{\alpha/2}\right) = \mathbb{P}\left(T \leq -k_{\alpha/2}\right) = \frac{\alpha}{2}$.

On a observé $T_{obs} = T(X_1, \ldots, X_n)$ et donc $\mathbb{P}\left(T \geq |T_{obs}|\right) = \frac{\alpha^*}{2}$ d'où $\alpha^* = 2(1 - F(|T_{obs}|))$ où F est la fonction de répartition de la loi f et α^* est la p-valeur de notre échantillon.

1.2.3 Représentation schématique

En reprenant les notations précédentes, on représente graphiquement la densité $f_{H_0,T}$ d'une statistique T sous l'hypothèse H_0.



(Figure 1) Représentation schématique de la p-valeur.

Cette représentation graphique met en évidence que le calcul de la p-valeur permet de se conforter ou non vis-à-vis de la conclusion du test. En effet, la p-valeur donne un degré de confiance sur le résultat du test. On observe ici que $k_\alpha < T_{obs}$ et que p-val < α, donc on rejette H_0.

1.3 Cas du test de Student

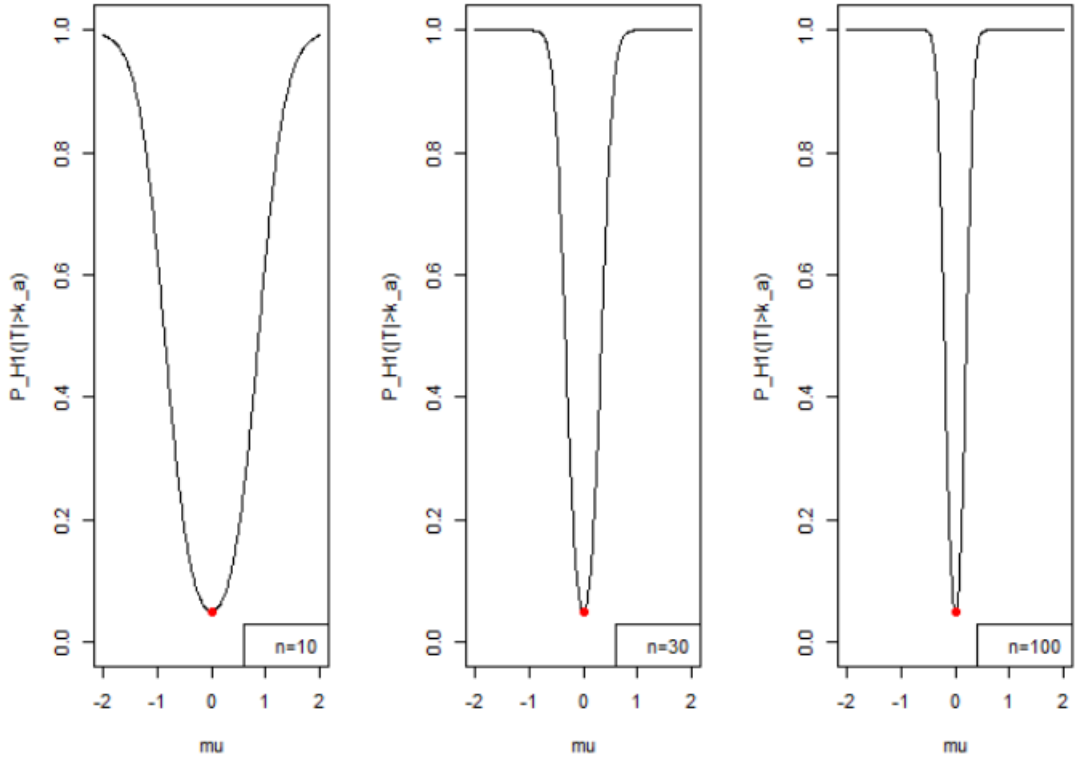
1.3.1 Test de nullité

Soit $n \in \{10, 100, 1000\}$, $X_1, \dots, X_n$ i.i.d. de loi $\mathcal{N}(\mu, \sigma^2)$ avec $\alpha = 0.05$ et μ,σ^2 inconnus.
On propose les hypothèses suivantes : $H_0 : \mu = \mu_0 = 0$ contre $H_1 : \mu \neq \mu_0$.
La loi de la moyenne empirique : $\overline{X_n} = \frac{1}{n}\sum_{i=1}^{n} X_i$ est une $\mathcal{N}(\mu, \frac{\sigma^2}{n})$.
On choisit la statistique $T = \sqrt{n}\dfrac{\overline{X_n} - \mu_0}{\sqrt{V_n^*}}$ où $V_n^* = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X_n})^2$ est la variance empirique corrigée.
Sous H_0, T suit une loi de Student à n-1 dégrés de liberté : $T \sim \mathcal{T}(n-1)$. La zone de rejet est de la forme $R_\alpha = \{T \leq -k_{\alpha/2}\} \cup \{T \geq k_{\alpha/2}\}$.
On va chercher $k_{\alpha/2} : 1 - \mathbb{P}(T < k_{\alpha/2}) = 0.025$ d'où $k_{\alpha/2} = F_{n-1}(0.975)$ avec F_{n-1} la fonction de répartition d'une $\mathcal{T}(n-1)$.
On a donc $R_\alpha = \{|T| \geq F_{n-1}(0.975)\}$ ainsi que $\alpha^* = 2(1 - F_{n-1}(|T_{obs}|))$

Sous H_1, nous avons que $\lim\limits_{n\to+\infty} |T| = +\infty$.

En effet, nous observons que $T = \sqrt{n} * \dfrac{\overline{X_n} - \mu_0}{\sqrt{V_n^*}} = \sqrt{n} * \dfrac{\overline{X_n} - \mu_1}{\sqrt{V_n^*}} + \sqrt{n} * \dfrac{\mu_1 - \mu_0}{\sqrt{V_n^*}} = A_n + B_n$ (en notant $\mu = \mu_1 \neq \mu_0$). On remarque alors que A_n suit asymptotiquement une $\mathcal{N}(0,1)$ (d'après le lemme de Slutsky) et que $|B_n| \underset{n\to+\infty}{\to} +\infty$ et donc que $|T| \underset{n\to+\infty}{\to} +\infty$.

De plus, pour $n \in \mathbb{N}^*$, en notant F_n la fonction de répartition d'une $\mathcal{T}(n)$ et F celle d'une $\mathcal{N}(0,1)$, nous avons que $\forall n \in \mathbb{N}^*$, F_n est bijective sur $\mathbb{R}$ de bijection réciproque F_n^{-1}. Comme $F_n \underset{n\to+\infty}{\to} F$, on a que $F_n^{-1} \underset{n\to+\infty}{\to} F^{-1}$ et donc que les quantiles d'une loi de Student tendent vers ceux d'une loi normale centrée réduite, ce qui permet d'écrire $\lim\limits_{n\to+\infty} \mathbb{P}_{H_1}\left(|T| > k_{\alpha,n}\right) = 1$.



(Figure 2) Représentation graphique de la puissance du test pour différentes valeurs de n, $\alpha = 5\%$ et $\sigma^2 = 1$.

Le graphique ci-dessus obtenu à partir du code 1 met en évidence que la puissance du test, c'est à dire la probabilité d'accepter H_1 à raison, augmente lorsque la taille de l'échantillon augmente. On remarque par ailleurs que pour seulement $n = 100$ et en notant μ_1 la vraie moyenne de l'échantillon, si $|\mu_0 - \mu_1| \geq 0.5$ alors la puissance du test sera quasiment de 1.

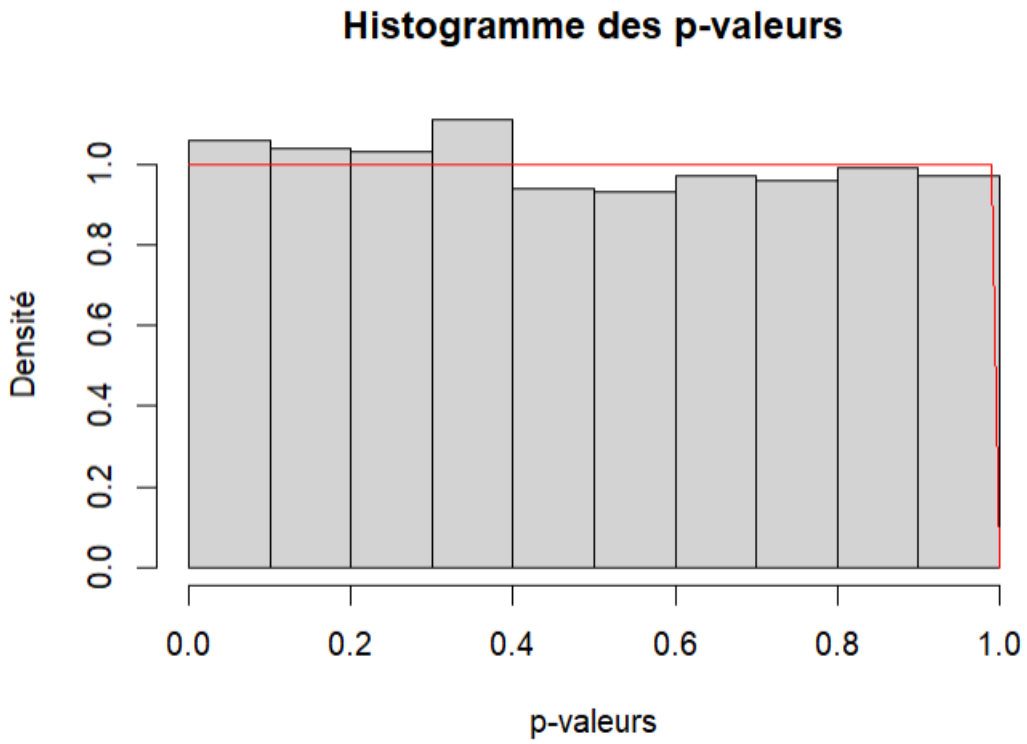
1.3.2 Application numérique sur R

On simule 1000 échantillons i.i.d. de taille n fixé et de loi $\mathcal{N}(0,1)$ grâce au code 2 et on effectue un test de Student pour chaque échantillon. À priori, on sait seulement que notre échantillon suit une loi $\mathcal{N}(\mu, \sigma^2)$ avec μ et σ^2 des paramètres inconnus.

On va faire un test de nullité sur le paramètre μ, c'est-à-dire $H_0 : \mu = 0$ contre $H_1 : \mu \neq 0$.

On calcule pour chaque échantillon la p-valeur qu'on stocke ensuite dans un vecteur, puis on affiche sur un graphe un histogramme des valeurs de notre vecteur ainsi que la densité

d'une loi uniforme sur [0,1] et on s'aperçoit graphiquement que la loi des p-valeurs suit une $\mathscr{U}([0,1])$ sous l'hypothèse H_0.



(Figure 3) Histogramme des p-valeurs.

Nous allons maintenant prouver que la loi des p-valeurs est une $\mathscr{U}([0,1])$ sous H_0 :
Notons F la fonction de répartition de notre statistique T, $F^{-1}(x) = \inf\{t \in \mathbb{R},\ F(t) \leq x\}$ l'inverse généralisée de F et $Y = F(T) \in]0,1[$. Nous avons :

$$\begin{aligned}
\mathbb{P}_{H_0}(Y \leq x) &= \mathbb{P}_{H_0}\left(F^{-1}(Y) \leq F^{-1}(x)\right) \quad \text{car } F^{-1} \text{ est croissante} \\
&= \mathbb{P}_{H_0}\left(T \leq F^{-1}(x)\right) \\
&= F(F^{-1}(x)) \\
&= x \quad \text{pour } x \in]0,1[.
\end{aligned}$$

Si $x \leq 0$, la probabilité ci-dessus vaut 0 et si $x \geq 1$, elle vaut 1. La loi des p-valeurs est donc une $\mathscr{U}([0,1])$ sous H_0.

Maintenant nous allons comparer les résultats théoriques obtenus à la main avec ceux fournis par l'ordinateur grâce à la fonction t.test du logiciel R (code 3).
On fait un test de nullité sur un n-échantillon i.i.d. de loi $\mathscr{N}(\mu,1)$, avec $n \in \{10,100,1000\}$ au risque $\alpha = 0.05$. On teste si la moyenne μ de nos échantillons vaut 0 cependant nous allons la faire varier et elle pourra valoir 0 ou 0.1 ou 1.
Un intervalle de confiance pour μ est donné par $IC(1-\alpha) = \left[\overline{X}_n - t^{n-1}_{\alpha\backslash 2}\frac{\sqrt{V^*_n}}{\sqrt{n}}, \overline{X}_n + t^{n-1}_{\alpha\backslash 2}\frac{\sqrt{V^*_n}}{\sqrt{n}}\right]$ où $t^{n-1}_{\alpha\backslash 2}$ est le quantile d'ordre $1-\frac{\alpha}{2}$ d'une loi de Student à n-1 degrés de liberté et vérifie $\mathbb{P}\left(T \leq t^{n-1}_{\alpha\backslash 2}\right) = 1-\frac{\alpha}{2}$.

Un intervalle de confiance bootstrap pour la p-valeur est donné par $\hat{IC}(1-\alpha) = \left[\hat{\theta}^*_{\lceil B\frac{\alpha}{2}\rceil}, \hat{\theta}^*_{\lceil B(1-\frac{\alpha}{2})\rceil}\right]$ où $\hat{\theta}^*_{\lceil B\frac{\alpha}{2}\rceil}$ et $\hat{\theta}^*_{\lceil B(1-\frac{\alpha}{2})\rceil}$ sont respectivement les $\lceil B\frac{\alpha}{2}\rceil$-ème et $\lceil B(1-\frac{\alpha}{2})\rceil$-ème valeurs par ordre croissant de nos statistiques bootstrapées $\hat{\theta}^*_b$ obtenues sur nos B échantillons boostraps.
On détermine ensuite des intervalles de confiance pour différentes valeurs de n et μ, qu'on a renseignés dans le tableau suivant :

	mu = 0		mu = 0,1		mu = 1	
n=10	$\bar{x}$=0.068	pval=0.86	$\bar{x}$=0.42	pval=0.11	$\bar{x}$=1,24	pval=0.01
	IC=[-0.76 , 0.89]	IC=[0.05 , 0.92]	IC=[-0.12 , 0.95]	IC=[0.001 , 0.657]	IC=[0.32 , 2.17]	IC=[5e-4 , 0.1]
n=100	$\bar{x}$=-0.036	pval=0.71	$\bar{x}$=0.21	pval=0.02	$\bar{x}$=0.95	pval=1e-15
	IC=[-0.23 , 0.15]	IC=[0.02 , 0.95]	IC=[0.04 , 0.39]	IC=[8e-5 , 4.6e-1]	IC=[0.75 , 1.15]	IC=[0.00 , 3.1e-12]
n=1000	$\bar{x}$=-0.027	pval=0.39	$\bar{x}$=0.103	pval=0.0008	$\bar{x}$=1.02	pval=0.00
	IC=[-0.09 , 0.03]	IC=[0.008 , 0.921]	IC=[0.04 , 0.16]	IC=[3e-7 , 4.5e-2]	IC=[0.96 , 1.09]	IC=[0.00 , 0.00]

(Figure 4) Exemples de résultats de tests de Student pour différentes valeurs de n et de μ.

On remarque que plus n est grand et plus la vraie moyenne de notre échantillon augmente, plus la p-valeur se rapproche de 0 ce qui nous ammène à rejeter l'hypothèse comme quoi notre échantillon est de moyenne nulle. En revanche pour un μ proche de 0 (0.1) et pour une taille d'échantillon n=10, les résultats nous ammènent à accepter H_0 à tord pour un niveau de test à 5% ce qui nous suggère de faire les tests avec des échantillons de tailles plus importantes.

On peut par ailleurs vérifier que nous ne nous sommes pas trompés dans nos calculs théoriques en vérifiant que nos résultats et ceux de la fonction t.test coïncident :

```
        One Sample t-test

data:  X
t = 1.2566, df = 999, p-value = 0.2092
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.02240529  0.10218552
sample estimates:
 mean of x
0.03989011

> print(c(mean,T,pval))
[1] 0.03989011 1.25656167 0.20920614
```

(Figure 5) Sortie graphique d'un test de Student sur R.

1.3.3 Et avec un échantillon non gaussien ?

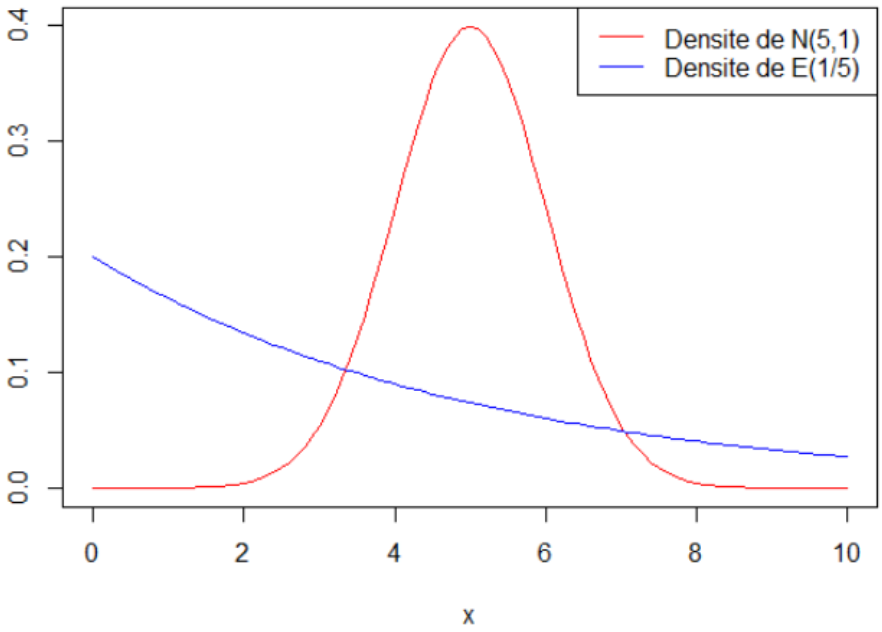
Nous allons maintenant effectuer le même test mais cette fois-ci pour un échantillon de taille moins conséquente, par exemple pour n=10 et $(Y_i)_{1\leq i\leq n}$ de loi $\mathscr{E}(\frac{1}{\mu})$ avec $\mu = 5$ ainsi que notre risque $\alpha = 0.05$.
Posons les hypothèses $H_0 : \mu = 8$ contre $H_1 : \mu \neq 8$.
Le test de Student suppose que les données suivent une distribution normale. Cette hypothèse n'est plus vérifiée ici et nous risquons d'obtenir des résultats trompeurs.

```
data:  Y
t = -0.13428, df = 9, p-value = 0.8961
alternative hypothesis: true mean is not equal to 8
95 percent confidence interval:
  2.26668 13.09082
sample estimates:
mean of x
 7.678749

> sort(Y)
 [1]  0.7849390  0.9248121  2.6682110  3.0227183  3.5722547  6.8331700  8.6976377  9.2367452 17.0453898 24.0016092
```

(Figure 6) Sortie graphique d'un test de Student pour un échantillon de loi $\mathscr{E}(\frac{1}{5})$ avec n=10.

Nous apercevons dans l'exemple ci-dessus que notre échantillon contient deux valeurs (24 et 17.05) très éloignées de la moyenne à 7.68. Cela s'explique par le fait que les valeurs extrêmes sont relativement fréquentes, en raison de la lourde queue de la distribution exponentielle, ce qui signifie que les valeurs extrêmes ont tendance à avoir un impact important sur les estimations de la moyenne.
En effet, le tirage de ces deux valeurs a faussé les résultats du test puisque l'on a obtenu une moyenne empirique plus importante que celle attendue, et du fait que l'on a supposé la moyenne de notre échantillon de 8, le test de Student nous conforte dans l'idée d'accepter cette hypothèse (p-valeur de 0.9), à tord puisque la vraie moyenne est 5...



(Figure 7) Comparaison des densités d'une $\mathscr{N}(5,1)$ et d'une $\mathscr{E}(\frac{1}{5})$.

En rouge la densité d'une $\mathscr{N}(5,1)$ et en bleu celle d'une $\mathscr{E}(1/5)$, on obersve le phenomène de queue lourde qui caractérise la distribution exponentielle.

Cependant, plus notre échantillon contient d'observations moins le phénomène sera impactant comme le montre l'exemple suivant pour n=100 :

```
data:  Y
t = -6.1331, df = 99, p-value = 1.775e-08
alternative hypothesis: true mean is not equal to 8
95 percent confidence interval:
 3.957027 5.933585
sample estimates:
mean of x
 4.945306
```

(Figure 8) Sortie graphique d'un test de Student pour un échantillon de loi $\mathscr{E}(\frac{1}{5})$ avec n=100.

La p-valeur obtenue est relativement petite par rapport à notre risque α, ce qui nous amène fortement à rejeter H_0. On remarque par ailleurs que sur plusieurs autres essais la conclusion reste la même, on rejette H_0 pour H_1, ce qui suggère que le test devient relativement plus précis lorsque la taille d'échantillon augmente.
C'est normal, puisque dans le cas où le nombre d'échantillons est suffisament grand, le TCL garantit que la loi de $\sqrt{n}\dfrac{\overline{Y_n} - \mu_Y}{\sigma_Y}$ sera asymptotiquement une $\mathscr{N}(0,1)$ et donc que sous H_0 notre statistique T suivra approximativement une loi de Student, ce qui signifie que le test de Student pourra être utilisé pour effectuer des tests sur la moyenne. En effet, dans le cas où la disitribution n'est pas normale mais que la moyenne empirique suit une loi normale, T suivra quand même une loi de Student sous H_0.

En résumé, le test de Student suppose d'utiliser des distributions normales, où des valeurs extrêmes ont peu de chances d'être obtenues et ont donc peu d'impact sur la moyenne. En utilisant une distribution exponentielle, on perd cette spécificité et en combinant à une petite taille d'échantillon, cela rend les résultats du test de Student peu fiables.

2 Tests multiples appliqués à l'expression de gènes

Les puces à ADN sont des outils utilisés pour étudier les schémas d'expression des gènes. Elles permettent de mesurer l'activité des gènes en étudiant les niveaux d'expression de milliers de gènes simultanément.

2.1 Modèle simple avec erreur gaussienne

Prenons par exemple $j \in \{1,...,5000\}$, $n \in \{1,...,10\}$, $p \in [0,1]$ ainsi que des quantités d'intérêt $\theta_j \sim \mathscr{B}(p)$, des bruits $\epsilon_{j,n} \sim \mathscr{N}(0,\sigma^2)$ et posons $Y_{j,n} = \theta_j + \epsilon_{j,n}$ (pour chaque j, on a n observations). Selon la modélisation, lorsque $\theta_j = 0$ on dira que le gène ne s'exprime pas et à l'inverse lorsque $\theta_j = 1$, on dira qu'il s'exprime. Posons les hypothèses $H_{0,j} : \theta_j = 0$ et $H_{1,j} : \theta_j > 0$.

Lorsque l'on s'aperçoit que l'on a trop de faux négatifs, cela peut signifier relativement à notre risque α, que la moyenne (sur n) de $Y_{j,n}$ est plutôt proche de 0 (tout en ayant $\theta_j = 1$) et que la variance est plutôt élevée (a une importance car petit échantillon), le tout pour avoir une statistique T pas trop grande et une p-valeur pas trop petite de sorte à conclure à tord H_0. En réduisant la variance, nous allons concentrer notre moyenne en 1 et augmenter notre statistique T de sorte à diminuer la p-valeur et rejeter H_0 à raison. Il est important de noter que nous ne pouvons ni augmenter la valeur que peut prendre n (l'échantillon sur n doit rester petit) ni modifier la variance du bruit car en pratique ce paramètre n'est pas contrôlable.

Lorsque l'on s'aperçoit que l'on a trop de faux positifs, cela peut signifier que la moyenne est trop éloignée de 0 (tout en ayant $\theta_j = 0$), on pourrait alors réduire la variance pour avoir une moyenne (sur n) des $Y_{j,n}$ plus proche de 0 mais alors n'aurions pas plus d'informations sur notre statistique T puisqu'à la fois $\overline{Y_{j,n}}$ et V_n^* se rapprochent de 0. Néanmoins, réduire notre risque α réduira les faux positifs mais cela augmentera d'un autre coté les faux négatifs.
Théoriquement, le nombre de faux positifs est donné par $FP = J(1-p)\alpha$.
En effet :

$$\begin{aligned}
\mathbb{E}\Big[\sum_{i=1}^{J} \mathbb{1}_{Y_i \geq t_\alpha^{n-1},\, \theta_i = 0}\Big] &= \sum_{i=1}^{J} \mathbb{E}\big[\mathbb{1}_{Y_i \geq t_\alpha^{n-1},\, \theta_i = 0}\big] \\
&= J\mathbb{E}\big[\mathbb{1}_{Y \geq t_\alpha^{n-1},\, \theta = 0}\big] \\
&= J\mathbb{P}\big(Y \geq t_\alpha^{n-1},\, \theta = 0\big) \\
&= J\mathbb{P}\big(\theta = 0\big)\mathbb{P}\big(Y \geq t_\alpha^{n-1} | \theta = 0\big) \\
&= J(1-p)\alpha
\end{aligned}$$

Voici ci-dessous un exemple de réalisation du modèle à l'aide du code 5 :

```
> methode1(10,alpha,mu,sd,J,p)
[1] "Nb vrais sites, Resultats, Nb faux pos, Nb faux neg"
[1] 255 463 238  30 255
> J*alpha*(1-p)
[1] 237.5
```

(Figure 9) Réalisation du code 5 pour J = 5000, $n = 10$, $\alpha = 0.05$, $p = 0.05$, $\mu = 0$ et $\sigma = 1$.

Pour cette réalisation, on conclut selon notre modèle qu'il y'a 463 vrais sites. En réalité, il y'en a exactement 255 et sur notre résultat de 463, 238 sont des faux positifs. Il y'a 30 faux négatifs donc 30 sites où le gène s'exprimait qui n'ont pas été retenus. On remarque également que le nombre de faux positifs théorique est égal à l'arrondi près au nombre de faux positifs observé.
En l'absence de correction, le nombre de faux positifs est plus élevé que le nombre de faux négatifs et près de la moitié des résultats sont faux.
La dernière valeur de la liste correspond à une vérification : on retire les faux positifs et ajoute les faux négatifs aux résultats, le nombre obtenu doit coïncider avec le nombre de vrais sites.

Nous aimerions réduire le nombre de faux négatifs, nous allons donc prendre $\sigma = 0.75$:

```
> methode1(10,alpha,mu,sd,J,p)
[1] "Nb vrais sites, Resultats, Nb faux pos, Nb faux neg"
[1] 241 493 254   2 241
> J*alpha*(1-p)
[1] 237.5
```

(Figure 10) Réalisation du code 5 pour $\sigma = 0.75$.

On remarque qu'en réduisant la variance du bruit, les faux négatifs diminuent effectivement. Malheureusement, ceci n'est qu'à titre expérimental puisqu'il n'est pas possible en pratique de modifier la variance du bruit, c'est pourquoi nous allons rester avec une variance valant 1 pour nos données simulées.

Nous nous apercevons dans les deux exemples précédents que le nombre de faux positifs est trop élévé et nous souhaitons donc le réduire. Nous allons répéter la même opération mais cette fois-ci en réduisant le risque α de sorte à ce que notre nouveau risque soit 5 fois plus petit. Nous nous attendons à une diminution du nombre de faux positifs mais à une augmentation du nombre de faux négatifs.

```
> methode1(10,alpha,mu,sd,J,p)
[1] "Nb vrais sites, Resultats, Nb faux pos, Nb faux neg"
[1] 256 208  49  97 256
> J*alpha*(1-p)
[1] 47.5
```

(Figure 11) Réalisation du code 5 pour $\alpha = 0.01$.

Par rapport à la première réalisation, on constate qu'il y'a 5 fois moins de faux négatifs pour 3 fois plus de faux positifs. Il semble y'avoir des améliorations en jouant sur le risque α. Cette observation introduit la partie 3 dans laquelle il sera question d'apporter de bonnes corrections sur le risque α selon différentes méthodes. Néanmoins il ne sera pas possible de réduire définitivement à 0 le nombre de faux positifs.

2.2 On s'attend à obtenir des faux positifs

Soient les variables aléatories i.i.d. $(\epsilon_i)_{1\leq i\leq J}$ suivant une loi $\mathcal{N}(\mu,\sigma^2)$, $\mu \in \mathbb{R}$, $\sigma^2 > 0$ et $\alpha \in]0,1[$.
Nous allons démontrer que $\lim\limits_{J\to+\infty} \mathbb{P}\big(\max\limits_{1\leq i\leq J} |\epsilon_i| \geq \alpha\sqrt{2\sigma^2\ln(J)}\big) = 1$.

Etape préliminaire : détermination d'un équivalent du reste de l'intégrale de Gauss.
Pour cela, nous allons démontrer le résultat suivant : Soit $a \in \mathbb{R}$, f et g deux applications continues et intégrables sur $[a,+\infty[$ à valeurs strictement positives, $f(x) = \underset{x\to+\infty}{o}(g(x))$ entraîne $\int_x^{+\infty} f(t)\mathrm{d}t = \underset{x\to+\infty}{o}\big(\int_x^{+\infty} g(t)\mathrm{d}t\big)$.

En effet, $f(x) = \underset{x\to+\infty}{o}(g(x))$ est équivalent à $\forall \epsilon > 0, \exists x_0 \in [a,+\infty[, \forall\, t \in [a,+\infty[, x_0 \le t \Rightarrow 0 < f(t) \le \epsilon g(t)$.
Soit $x \in [a,+\infty[$, f et g sont intégrables sur $[x,+\infty[$. De plus, par croissance de l'intégrale, $\forall x \in [a,+\infty[, x_0 \le x \Rightarrow 0 < \int_x^{+\infty} f(t)\mathrm{d}t \le \epsilon \int_x^{+\infty} g(t)\mathrm{d}t$ et donc que $\int_x^{+\infty} f(t)\mathrm{d}t = \underset{x\to+\infty}{o}\big(\int_x^{+\infty} g(t)\mathrm{d}t\big)$.

Nous allons maintenant utiliser ce résultat pour déterminer un équivalent du reste de l'intégrale de Gauss :

$$\begin{aligned}
\text{Prenons } 2\mu \le x, \quad \mathrm{I}(x) = \int_x^{+\infty} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}\mathrm{d}t &= \int_x^{+\infty} \frac{\frac{t-\mu}{\sigma^2}}{\frac{t-\mu}{\sigma^2}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}\mathrm{d}t \\
&= \Big[-\frac{e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}}{\frac{t-\mu}{\sigma^2}}\Big]_x^{+\infty} - \int_x^{+\infty} \frac{e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}}{\left(\frac{t-\mu}{\sigma}\right)^2}\mathrm{d}t \quad \text{par IPP} \\
&= \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\frac{x-\mu}{\sigma^2}} - \int_x^{+\infty} \frac{e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}}{\left(\frac{t-\mu}{\sigma}\right)^2}\mathrm{d}t \\
&= \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\frac{x-\mu}{\sigma^2}} - \mathrm{J}(x)
\end{aligned}$$

Or nous avons que $\frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\left(\frac{x-\mu}{\sigma}\right)^2} = \underset{x\to+\infty}{o}\big(e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}\big)$ et d'après le résultat démontré précédemment que $\mathrm{J}(x) = \underset{x\to+\infty}{o}\big(\mathrm{I}(x)\big)$. D'après la dernière égalité, on a :

$$1 + \frac{\mathrm{J}(x)}{\mathrm{I}(x)} = \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\mathrm{I}(x)\left(\frac{x-\mu}{\sigma^2}\right)} \quad \text{avec } \frac{\mathrm{J}(x)}{\mathrm{I}(x)} = \underset{x\to+\infty}{o}(1)$$

On obtient alors que $\lim\limits_{x\to+\infty} \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\mathrm{I}(x)\left(\frac{x-\mu}{\sigma^2}\right)} = 1$ et donc que $\int_x^{+\infty} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}\mathrm{d}t \underset{x\to+\infty}{\sim} \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\frac{x-\mu}{\sigma^2}}$.

Nous pouvons désormais effectuer le calcul souhaité :

$$\begin{aligned}
\mathbb{P}\big(\max_{1\le i\le \mathrm{J}} |\epsilon_i| \ge \alpha\sqrt{2\sigma^2 ln(\mathrm{J})}\big) &= 1 - \mathbb{P}\big(\max_{1\le i\le \mathrm{J}} |\epsilon_i| < \alpha\sqrt{2\sigma^2 \ln(\mathrm{J})}\big) \\
&= 1 - \prod_{i=1}^{\mathrm{J}} \mathbb{P}\big(|\epsilon_i| < \alpha\sqrt{2\sigma^2 \ln(\mathrm{J})}\big) \\
&= 1 - \mathbb{P}\big(|\epsilon_1| < \alpha\sqrt{2\sigma^2 \ln(\mathrm{J})}\big)^{\mathrm{J}}
\end{aligned}$$

$$
\begin{aligned}
&= 1 - \left(\mathrm{F}\left(\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right) - \mathrm{F}\left(-\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right)\right)^{\mathrm{J}} \\
&= 1 - \left(2\mathrm{F}\left(\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right) - 1\right)^{\mathrm{J}} \\
&= 1 - e^{\mathrm{J}\ln\left(2\mathrm{F}\left(\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right)-1\right)} \\
&= 1 - e^{\mathrm{J}\ln\left(1+2\mathrm{F}\left(\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right)-2\right)} \\
&\underset{\mathrm{J}\to+\infty}{\sim} 1 - e^{2\mathrm{J}\left(\mathrm{F}\left(\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right)-1\right)} \\
&\underset{\mathrm{J}\to+\infty}{\sim} 1 - e^{-2\mathrm{J}\left(\int_{\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}}^{+\infty} \frac{e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}}{\sqrt{2\pi}\sigma}\mathrm{d}t\right)} \\
&\underset{\mathrm{J}\to+\infty}{\sim} 1 - e^{\frac{-2\mathrm{J}e^{-\frac{1}{2}\left(\frac{\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}-\mu}{\sigma}\right)^2}}{\sqrt{2\pi}\sigma\frac{\alpha\sqrt{2\sigma^2\ln(\mathrm{J})}-\mu}{\sigma^2}}} \quad \text{d'après l'équivalent démontré plus tôt} \\
&\underset{\mathrm{J}\to+\infty}{\sim} 1 - e^{\frac{-2\mathrm{J}e^{-\alpha^2\ln(\mathrm{J})}}{\sqrt{2\pi}\alpha\sqrt{2\ln(\mathrm{J})}}} \\
&\underset{\mathrm{J}\to+\infty}{\sim} 1 - e^{\frac{-\mathrm{J}^{1-\alpha^2}}{\alpha\sqrt{\pi\ln(\mathrm{J})}}} \underset{\mathrm{J}\to+\infty}{\to} 1
\end{aligned}
$$

On obtient bien que $\lim_{\mathrm{J}\to+\infty} \mathbb{P}\left(\max_{1\le i\le \mathrm{J}} |\epsilon_i| \ge \alpha\sqrt{2\sigma^2\ln(\mathrm{J})}\right) = 1$, ce qui conclut la démonstration.
En pratique, le résultat établi ci-dessus signifie que si l'on fait un test par site, alors on aura presque sûrement des faux positifs.

3 Les corrections

En effectuant des tests indépendement sur les J sites, le nombre moyen de faux positifs est souvent bien supérieur au nombre de gènes vraiment exprimés et donc les résultats sont essentiellement constitués d'erreurs. On se propose de présenter différentes corrections qui permettront de réduire ces erreurs.

3.1 Différents types de corrections

Dans cette partie, on considère toujours le modèle $Y_{j,n} = \theta_j + \epsilon_{j,n}$ avec pour tout $(j,n) \in [\![1,5000]\!] \times [\![1,10]\!]$; $\theta_j \sim \mathscr{B}(p)$ et $\epsilon_{j,n} \sim \mathscr{N}(0,\sigma^2)$, θ_j étant la variable définissant si le site j est codant et $\epsilon_{j,n}$ le bruit blanc de mesure. On suppose les variables indépendantes et identiquement distribuées.
Les données d'études seront simulées numériquement, on peut donc faire l'hypothèse d'une variance unitaire pour le bruit blanc et on fixe une probabilité p=0.05 pour les variables θ_j.

3.1.1 Correction de Bonferroni

La correction de Bonferroni est une méthode cherchant à diminuer le nombre de faux positifs en réduisant le niveau de chaque test j. Selon la correction de Bonferroni, pour un risque $\alpha \in]0,1[$ et pour $j \in [\![1,J]\!]$, nous rejettons $H_{0,j}$ si $p_j \leq \alpha_{BONF}$ avec p_j la j-ème p-valeur et $\alpha_{BONF} = \frac{\alpha}{J}$. Ainsi, avec ce nouveau niveau α_{BONF} pour chaque site, le nombre de faux positifs théorique (vu à la partie 2.1) devient FP=$J(1-p)\alpha_{BONF} = (1-p)\alpha$.
Le graphique ci-dessous représente les exemples de résultats pour des tests avec la correction de Bonferroni, la moyenne des résultats est réalisée sur 50 exemples de tests multiples.

	Nbr_vrai_site	Nbr_accept_H1	Faux_positif	Faux_negatif
Exemple 1	251.0000	1.00	0.00	250.00
Exemple 2	257.0000	4.00	0.00	253.00
Exemple 3	252.0000	0.00	0.00	252.00
Exemple 4	260.0000	2.00	0.00	258.00
Exemple 5	258.0000	3.00	0.00	255.00
Exemple 6	250.0000	4.00	0.00	246.00
Exemple 7	234.0000	2.00	0.00	232.00
Moyenne	251.7143	2.38	0.02	244.56

(Figure 12) Exemples de résultats de tests multiples par correction de Bonferroni pour $\alpha = 0.05$, J = 5000 et p, n, μ, σ inchangés.

On remarque qu'il n'y a quasiment pas de faux positifs (2%) et donc que nous avons globalement toujours conclu H_0 à raison. En revanche, les taux de faux négatifs obtenus sont énormes, ce qui nous laisse penser que nous ne concluons quasiment jamais H_1 avec cette correction. La liste des résultats est fiable mais quasiment vide. En effet, en faisant J=5000 tests avec la correction de Bonferroni, le nouveau niveau α_{BONF} fait drastiquement chuter la puissance de chaque test. Pour $\alpha = 5\%$, le niveau corrigé est de $\alpha_{BONF} = 0.001\%$.

En utilisant la correction de Bonferroni pour des trop grandes valeurs de J (ici J=5000), l'hypothèse $H_{0,j}$ sera quasiment toujours acceptée à tord ou à raison, ce qui signifie qu'il y'aura très peu de faux positifs, beaucoup de faux négatifs et surtout que l'on va passer quasiment à coté de tous les sites pour lesquels les gènes s'expriment. On obtiendra donc une liste de résultats très petite mais fiable.

3.1.2 Correction de Šidák

Selon la correction de Šidák, on rejette $H_{0,j}$ si la p-valeur $p_j \leq \alpha_{SID}$ avec $\alpha_{SID} = 1-(1-\alpha)^{\frac{1}{J}}$. Pour $\alpha = 0.05$ et $J = 5000$, $\alpha_{SID} = 1.025861e-05$ et $\alpha_{BONF} = 1e-05$, la correction de Šidák est légèrement moins conservatrice que celle de Bonferroni.
Les résultats obtenus à l'aide d'une correction de Šidák sont similaires à ceux obtenus à l'aide d'une correction de Bonferroni comme le montre la figure 13 :

	Nbr_vrai_site	Nbr_accept_H1	Faux_positif	Faux_negatif
Exemple 1	251.0000	1.00	0.00	250.0
Exemple 2	257.0000	4.00	0.00	253.0
Exemple 3	252.0000	0.00	0.00	252.0
Exemple 4	260.0000	2.00	0.00	258.0
Exemple 5	258.0000	3.00	0.00	255.0
Exemple 6	250.0000	6.00	0.00	244.0
Exemple 7	234.0000	2.00	0.00	232.0
Moyenne	251.7143	2.44	0.02	244.5

(Figure 13) Exemples de résultats de tests multiples par correction de Šidák pour $\alpha = 0.05$, $J = 5000$ et p, n, μ, σ inchangés.

3.1.3 Correction de Holm-Bonferroni

Le principe de la correction de Holm-Bonferroni est le suivant :
- Les p-valeurs de chaque test sont triées par ordre croissant.
- En notant p_j la j-ème p-valeur par ordre croissant, si $p_j \leq \frac{\alpha}{J+1-j}$, alors on rejette $H_{0,j}$.

	Nbr_vrai_site	Nbr_accept_H1	Faux_positif	Faux_negatif
Exemple 1	251.0000	1.00	0.00	250.00
Exemple 2	257.0000	4.00	0.00	253.00
Exemple 3	252.0000	0.00	0.00	252.00
Exemple 4	260.0000	2.00	0.00	258.00
Exemple 5	258.0000	3.00	0.00	255.00
Exemple 6	250.0000	4.00	0.00	246.00
Exemple 7	234.0000	2.00	0.00	232.00
Moyenne	251.7143	2.38	0.02	244.56

(Figure 14) Exemples de résultats de tests multiples par correction de Holm-Bonferroni pour $\alpha = 0.05$, $J = 5000$ et p, n, μ, σ inchangés.

On constate également pour cette correction que peu de sites sont acceptés mais que la liste des résultats est fiable.

Les trois corrections présentées ci-dessus appartiennent à une même famille de tests appelée correction de typer FWER (Family-Wise Error Rate) dont le principe est d'assurer que le taux de faux positifs ne dépasse pas un certain seuil. C'est pourquoi peu voire pas de gènes ne passent les réstrictions. Le principe des corrections FWER est de réduire drastiquement la puissance pour chaque test. On peut alors se demander pourquoi utiliser ces corrections pour des tests multiples? Le choix d'un type de correction dépend des hypothèses du modèle ainsi que des résultats recherchés. Ces trois corrections ont pour objectif d'assurer au maximum un petit taux de faux positifs sous l'hypothèse d'observations indépendantes.

3.1.4 Correction de Benjamini-Hochberg

La correction de Benjamini-Hochberg tolère un plus fort taux de faux positifs. Le principe est le suivant :
- Les p-valeurs de chaque test sont triées par ordre croissant.
On corrige ensuite les p-valeurs de chaque test j comme ci-dessous :
- Pour $j \in \{1, ..., J\}$, $p_{j-\mathrm{BH}} = p_j \times \frac{J}{j}$, avec p_j la j-ème p-valeur par ordre croissant. Si $p_{j-\mathrm{BH}} \leq \alpha$ alors on rejette $H_{0,j}$.

	Nbr_vrai_site	Nbr_accept_H1	Faux_positif	Faux_negatif
Exemple 1	251.0000	25.0	1.00	227.00
Exemple 2	257.0000	26.0	2.00	233.00
Exemple 3	252.0000	34.0	6.00	224.00
Exemple 4	260.0000	25.0	1.00	236.00
Exemple 5	258.0000	53.0	8.00	213.00
Exemple 6	250.0000	49.0	5.00	206.00
Exemple 7	234.0000	38.0	2.00	198.00
Moyenne	251.7143	29.8	1.62	218.74

(Figure 15) Exemples de résultats de tests multiples par correction de Benjamini-Hochberg.

On remarque qu'avec la correction de Benjamini-Hochberg, les nombres de sites acceptés sont bien plus élevés que pour les corrections précédentes sans pour autant que les nombres de faux positifs explosent. Cette correction fait partie de la famille des corrections de type FRD pour False Rate Discovery.
Les corrections de type FRD sont des approches statistiques utilisées pour contrôler le rapport entre le nombre de faux positifs et le nombre de tests réalisés. Elles sont particulièrement utiles dans les études de grande dimension où de nombreux tests sont effectués simultanément.

3.1.5 Correction de Benjamini-Yekutieli

La correction de Benjamini-Yekutieli est une variante de la correction de Benjamini-Hochberg pour des hypothèses de dépendance. C'est aussi une correction de type FDR. À l'image de la procédure de Benjamini-Hochberg, le principe est de trier les p-valeurs par ordre croissant puis de corriger selon que les tests sont indépendants ou non.
La procédure est expliquée ci-dessous :
- On trie les p-valeurs de chaque test par ordre croissant.
On corrige ensuite les p-valeurs de chaque test j comme ci-dessous :
- Si les tests sont indépendants, $\lambda_J = 1$ et on est rammené à la correction de Benjamini-Hochberg. Sinon, sous des hypothèses de dépendance, $\lambda_J = \sum_{i=1}^{J} \frac{1}{i}$.
- Pour $j \in [\![1, J]\!]$, $p_{j-BY} = p_j \times J\frac{\lambda_J}{j}$, avec p_j la j-ème p-valeur par ordre croissant et λ_j calculé comme ci-dessus. Si $p_{j-BY} \leq \alpha$ alors on rejette $H_{0,j}$.

On notera que l'algorithme présenté ci-dessus est utilisé pour des variables dont on ne peut pas supposer l'indépendance (comme dans la partie 4). C'est pourquoi les résultats de cet algorithme ne seront pas présentés dans la partie 3 (l'algorithme est trop conservatif, la moyenne d'hypothèses alternatives acceptées est très inférieure à 1 pour 5000 tests).
Nous allons donc utiliser dans la partie 3 une variante de l'algorithme de Benjamini-Yekutieli.
Ainsi la dernière étape sur la boucle pour $j \in [\![1, J]\!]$ devient : $p_{J-BY} = p_j \times \frac{J}{\lambda_j}$ avec $\lambda_j = \sum_{i=1}^{j} \frac{1}{i}$

	Nbr_vrai_site	Nbr_accept_H1	Faux_positif	Faux_negatif
Exemple 1	240	5.00	0.00	235.00
Exemple 2	235	2.00	0.00	233.00
Exemple 3	260	7.00	0.00	253.00
Exemple 4	265	1.00	0.00	264.00
Exemple 5	258	5.00	0.00	253.00
Exemple 6	258	4.00	0.00	254.00
Exemple 7	262	2.00	0.00	260.00
Moyenne	254	4.78	0.14	239.94

(Figure 16) Exemples de résultats de tests multiples par correction alternative de Benjamini-Yekutieli.

Dans la suite de la partie 3, par abus de notation, on appellera correction de Benjamini-Yekutieli la correction alternative expliquée ci-dessus. Dans la partie 4, on utilise la correction de Benjamini-Yekutieli originalle. La figure 16 montre que la correction de Benjamini-Yekutieli est moins conservatrice que les corrections de type FWER mais plus que celle de Benjamini-Hochberg.

3.2 Comparaison des corrections

Afin de choisir la correction statistique qui semble la plus intéréssante à appliquer sur des données réelles, on crée un tableau représentant la moyenne du nombre de conclusion

H1, la moyenne du nombre de faux positifs et la moyenne du nombre de faux négatifs pour chaque correction.

	Moyenne_vrai_site	Moyenne_accept_H1	Moyenne_faux_positif	Moyenne_faux_negatif
Bonferroni	245.7143	2.74	0.04	250.84
Sidak	245.7143	2.78	0.04	250.80
Benjamini-Hochberg	245.7143	33.82	1.88	221.60
Holm Bonferroni	245.7143	2.74	0.04	250.84
Benjamini Yekutieli	245.7143	5.38	0.22	248.38

(Figure 17) Comparaison des moyennes des résultats pour chaque correction appliquée à la même base de données simulée.

Ce tableau met en évidence les différences de puissance et de contrôle du taux de fausses découvertes pour chaque correction. Ainsi, les corrections de type FWER (Bonferroni, Šidák et Holm-Bonferroni) sont des corrections très conservatrices (les tests statistiques sont très peu puissants). Le nombre de faux positifs est extrêmement faible ce qui nous permet d'être très confiants pour les résultats obtenus par ces méthodes malgrès le fait qu'ils soient peu nombreux.
Les corrections de type FDR comme la correction de Benjamini-Yekutieli ou encore celle de Benjamini-Hochberg sont des corrections qui se veulent moins conservatrices et donc plus puissantes. En effet, on remarque que le taux d'acceptation pour la correction de Benjamini-Yekutieli est deux fois plus important que pour les corrections de type FWER. Cependant, le taux de faux positifs est aussi beacuoup plus important avec cette correction (facteur supérieur à 5 par rapport aux corrections de type FWER). Un facteur 5 est une très forte différence quand on regarde le taux de faux positifs de manière relative aux corrections précédentes. On peut en revanche dire que dans le contexte de l'étude, avoir en moyenne un taux de 0.22 faux positifs pour 5000 tests est acceptable.
La méthode qui semble la plus efficace avec un taux de faux positifs de 1.88 est la méthode de Benjamini-Hochberg. En effet, cette méthode de tests multiples accepte en moyenne beaucoup plus de fois l'hypothèse alternative. Contrairement aux autres corrections étudiées, la méthode de Benjamini-Hochberg accepte 33.82 sites testés sur les 5000 sites. Cette méthode est donc beaucoup plus efficace malgré le fait que son taux de faux positifs (1.88) est plus important que les autres méthodes.

3.3 Influence de la variance du bruit

Nous avons vu à la partie 2.1 que la variance du bruit gaussien a un fort impacte sur les tests statistiques en raison du petit nombre d'observations (n=10). Le bruit dù à la mesure ne peut pas être modifié car il dépend de critères liés aux conditions de l'étude. Cependant, il semble intéréssant de regarder l'évolution des moyennes des corrections pour différentes variances (figure 18). La figure ci-dessous est donc la compilation de la figure 17 pour cinq variances précises. Les variances choisies sont de $\{0.9; 0.95; 1; 1.05; 1.1\}$. La variance augmente toutes les 5 lignes.

	Nbr_vrai_site	Nbr_accept_H1	Faux_positif	Faux_negatif
var=0.9 ; Bonferroni	240.1429	4.00	0.04	242.34
Sidak	240.1429	4.02	0.04	242.32
Benjamini-Hochberg	240.1429	70.72	3.32	178.90
Holm Bonferroni	240.1429	4.00	0.04	242.34
Benjamini Yekutieli	240.1429	9.14	0.10	237.26
var=0.95 ; Bonferroni	251.1429	3.02	0.02	246.60
Sidak	251.1429	3.06	0.02	246.56
Benjamini-Hochberg	251.1429	50.76	2.42	201.26
Holm Bonferroni	251.1429	3.02	0.02	246.60
Benjamini Yekutieli	251.1429	6.58	0.12	243.14
var=1 ; Bonferroni	245.7143	2.74	0.04	250.84
Sidak	245.7143	2.78	0.04	250.80
Benjamini-Hochberg	245.7143	33.82	1.88	221.60
Holm Bonferroni	245.7143	2.74	0.04	250.84
Benjamini Yekutieli	245.7143	5.38	0.22	248.38
var=1.05 ; Bonferroni	256.7143	1.68	0.04	250.62
Sidak	256.7143	1.72	0.04	250.58
Benjamini-Hochberg	256.7143	20.10	1.02	233.18
Holm Bonferroni	256.7143	1.68	0.04	250.62
Benjamini Yekutieli	256.7143	3.22	0.04	249.08
var=1.1 ; Bonferroni	241.8571	1.46	0.02	249.44
Sidak	241.8571	1.46	0.02	249.44
Benjamini-Hochberg	241.8571	11.34	0.66	240.20
Holm Bonferroni	241.8571	1.46	0.02	249.44
Benjamini Yekutieli	241.8571	2.38	0.02	248.52

(Figure 18) Comparaison des moyennes des résultats en fonction de la variance pour chaque correction étudiée.

On remarque que lorsque la variance augmente, le nombre de fois où l'on accepte H_1 diminue. Le nombre de faux positifs semble être stable pour les méthodes de types FWER (compris entre 0.02 et 0.04), et le taux d'acceptation de H_1 reste très faible ($\leq$ 4.02). Les méthodes FWER semblent donc stable pour des bruits de variances σ^2 proches de 1.
Les méthodes de types FDR comme les corrections de Benjamini-Hochberg et Benjamini-Yekutieli ont de plus fortes variations de résultats en fonction de la variance. La méthode BH accepte en moyenne 70 fois l'hypothèse alternative pour un bruit de variance 0.9 contre 11 fois pour un bruit de variance 1.1 tandis que le taux de faux positifs passe de 3.32 à 0.66. Ces résultats ne sont pas étonnant, le taux de faux positifs sur le taux d'acceptation est à-peu-près toujours égal à 5% car cette méthode maintient un taux d'erreurs commises inférieur à 5%. La méthode de Benjamini-Yekutieli alternative conserve un taux d'erreurs inférieur à 5% mais souvent plus bas (de l'ordre de 1%).

Conclusion

Finalement, la méthode a employer pour réaliser des tests mutliples dépend des attentes vis-à-vis des résultats. Si l'on veut minimiser le taux de faux positifs en acceptant de perdre de la puissance sur les tests, la méthode de Benjamini-Yekutieli semble la plus efficace. Si l'on veut maximiser les découvertes tout en acceptant un taux de faux positifs non négligeable, la méthode de Benjamini-Hochberg semble être la plus efficace.

4 Application à des données réelles

Le scientifique Todd Golub publie en 1999 une base de données d'expression de gènes. Les données récoltées sont parmi les plus connues en bioinformatique.
Nous allons utiliser une partie de ces données grâce à la base de données "golub" sur R.

4.1 Présentation de la base de données "golub"

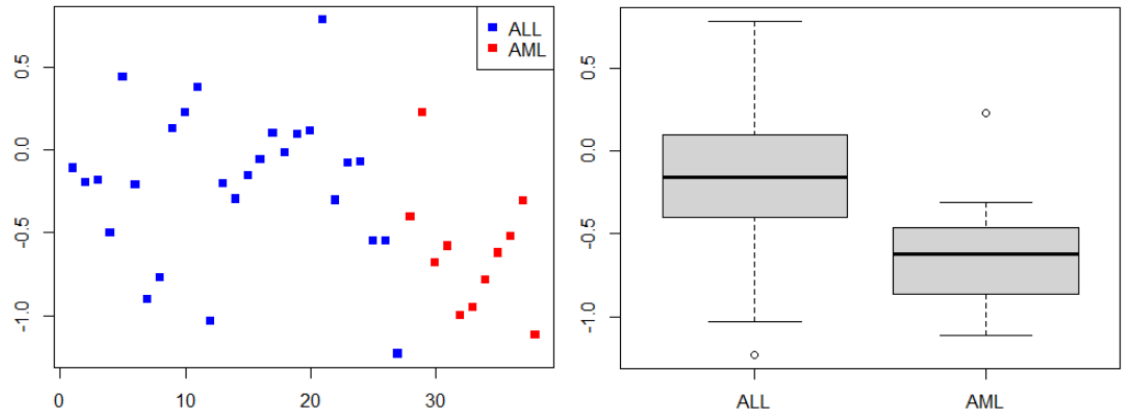
La base de données "golub" contient 3051 gènes en ligne et 38 patients atteints de leucémie en colonne. Sur les 38 patients, les 27 premiers sont atteints de leucémie aiguë lymphoblastique (ALL) et les 11 autres de leucémie aiguë myéloïde (AML). Une deuxième base de données nommée "golub.gnames" contient respectivement les index, ID et noms des gènes étudiés.
Voici un aperçu de la base de données "golub" pour les 6 premiers gènes et les 5 premiers patients :

AFFX-HUMISGF3A/M97935_MA_at	-1.45769	-1.39420	-1.42779	-1.40715	-1.42668
AFFX-HUMISGF3A/M97935_MB_at	-0.75161	-1.26278	-0.09052	-0.99596	-1.24245
AFFX-HUMISGF3A/M97935_3_at	0.45695	-0.09654	0.90325	-0.07194	0.03232
AFFX-HUMRGE/M10098_5_at	3.13533	0.21415	2.08754	2.23467	0.93811
AFFX-HUMRGE/M10098_M_at	2.76569	-1.27045	1.60433	1.53182	1.63728
AFFX-HUMRGE/M10098_3_at	2.64342	1.01416	1.70477	1.63845	-0.36075

(Figure 19) Aperçu de la base de données "golub".

Pour l'expression des différents gènes de la base de données "golub", nous pouvons utiliser le modèle suivant où pour le j-ème gène et pour le i-ème patient, $Y_{j,i} = \theta_{j,i} + \epsilon_{j,i}$.On dira que le gène est réprimé lorsque $\theta_{j,i} = -1$, on dira que le gène est exprimé lorsque $\theta_{j,i} = 1$ et enfin on dira que le gène n'est ni réprimé, ni exprimé (est dans un état stable) lorsque $\theta_{j,i} = 0$. Dans le cadre de notre étude, nous supposerons que les erreurs sont gaussiennes.

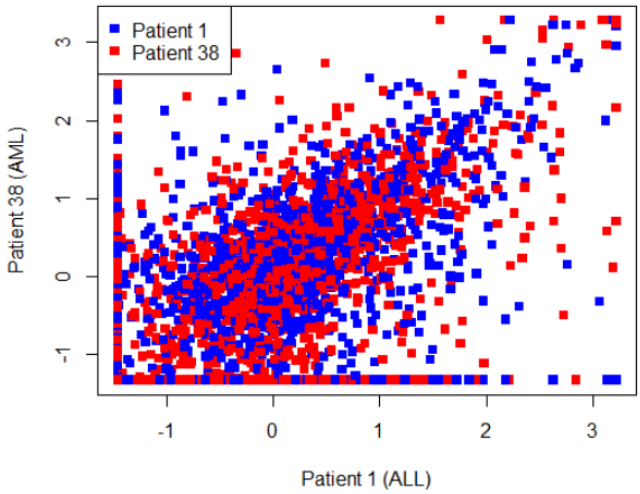
Prenons par exemple le 720-ème gène nommé "FRG1 mRNA" et affichons le nuage de points ainsi que le box-plot :



(Figure 20) Nuage de points et box-plot des données du gène 720.

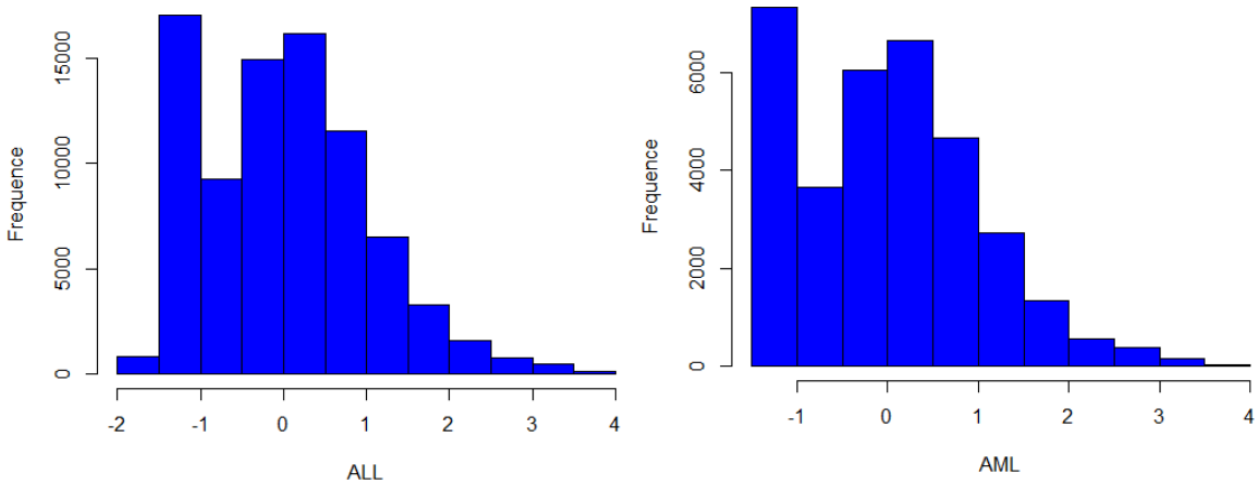
On remarque que le gène ne s'exprime pas de la même manière selon la maladie du patient. Pour les patients atteints de ALL, on lit une moyenne de données de -0.19 contre -0.61 pour ceux atteints de AML. On aurait donc tendance à dire que le gêne est stable dans le cas où le patient est atteint de ALL alors que dans le cas où il est atteint de AML, qu'il est réprimé.

Au lieu de comparer les patients sur un seul gène, nous pouvons aussi en comparer deux sur tous les gènes :



(Figure 21) Représentation des données de gènes des patients 1 et 38.

On observe que la grande majorité de données des gènes pour les deux patients est contenue dans l'intervalle $[-1,1]$, et si on décide de tracer les histogrammes des valeurs totales en fonction des maladies on obient le graphe suivant :

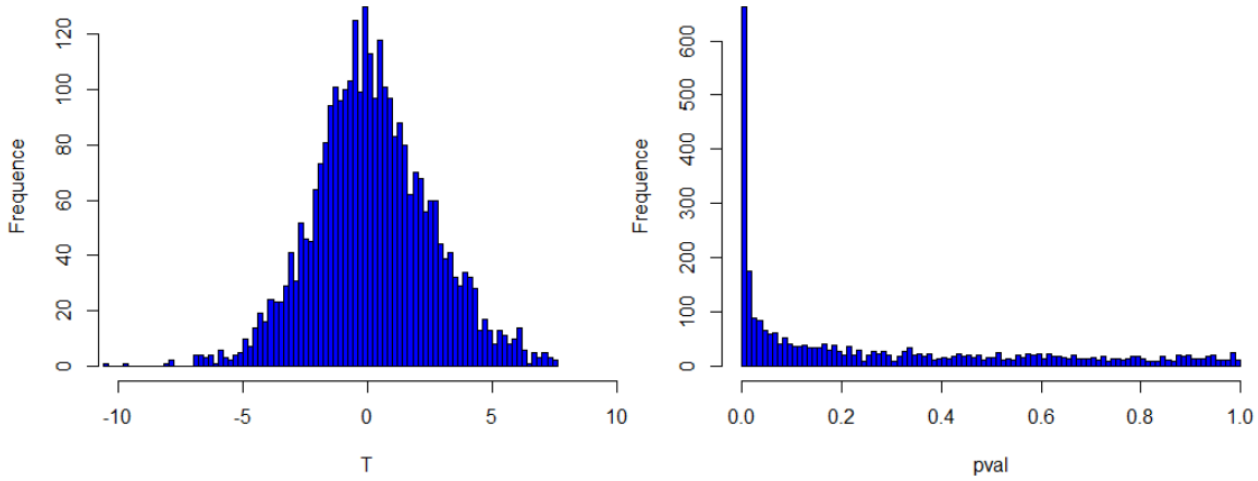


(Figure 22) Histogrammes de l'expression des gènes des patients ALL et AML.

4.2 Comparaison de l'expression des gènes entre patients ALL et AML

Nous allons maintenant comparer les gènes des patients atteints de ALL et de AML. On dispose de $(X_1,...,X_n)$ de loi $\mathcal{N}(\mu_1,\sigma_1^2)$ et $(Y_1,...,Y_m)$ de loi $\mathcal{N}(\mu_2,\sigma_2^2)$. On suppose que les variances ne sont pas égales. Nous allons introduire la statistique suivant : $T = \dfrac{\overline{X_n} - \overline{Y_n}}{\sqrt{\frac{V_X}{n} + \frac{V_Y}{m}}}$.

Posons $H_0 : \mu_1 = \mu_2$ et $H_1 : \mu_1 \neq \mu_2$. Sous H_0, $T \sim \mathcal{T}(a)$ avec a non entier et nous allons effectuer le test de Welch avec la commande t.test(x, y, var.equal=FALSE) de R.

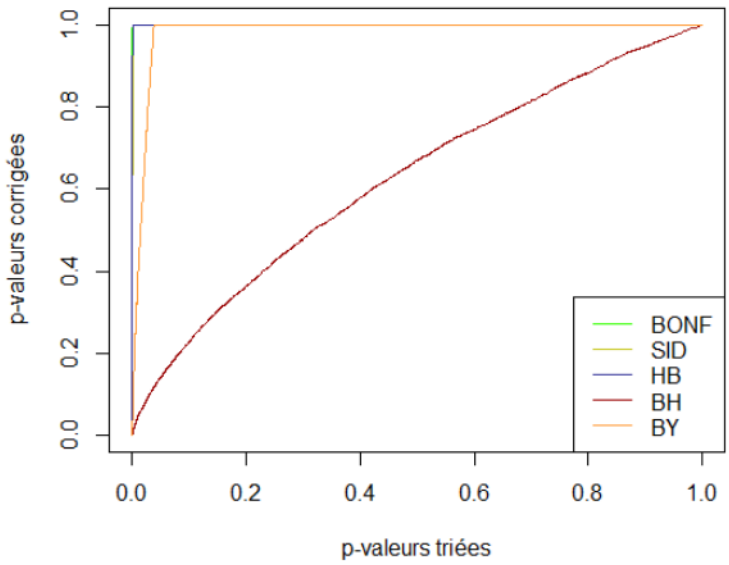


(Figure 23) Histogrammes des valeurs de T et des p-valeurs.

Nous avons affiché en figure 23 les histogrammes des valeurs des statistiques T et des p-valeurs pour les 5031 gènes de l'étude.

4.3 Prise de décision

Il s'agit maintenant de décider quels gènes s'expriment différemment selon que le patient est atteint de ALL ou de AML.
Pour ce faire, nous allons appliquer les corrections présentées en partie 3 aux p-valeurs obtenues précédemment.



(Figure 24) P-valeurs corrigées en fonction des p-valeurs triées.

À noter en figure 24 que si la p-valeur corrigée était supérieure à 1 alors nous la fixions à 1. On remarque que pour les trois corrections de type FWER, les p-valeurs corrigées se capent très vite à 1. Pour les corrections de type FRD, ce phénomène arrive légèrement plus tard pour la correction de Benjamini-Yekutieli et les p-valeurs corrigées selon Benjamini-Hochberg évoluent différemment des autres.

Les résultats obetnus sont affichés en figure 25.

	BONF	SID	HB	BH	BY
Resultats	103	103	103	695	287

(Figure 25) Résultats de l'analyse.

Ces résultats sont nos décisions concernant le nombre de gènes qui s'expriment différement en fonction de la maladie des patients.
On constate que les résultats obtenus à partir des corrections de type FWER sont les mêmes et représentent 3.4% des gènes totaux. Selon la correction de Benjamini-Yekutieli, 9.4% des gènes totaux s'expriment différement en fonction de la maladie des patients. Enfin, selon la correction de Benjamini-Hochberg, 22.8% soit près d'un quart des gènes totaux étudiés s'expriment de manière différente selon que le patient est ALL ou AML.

En conclusion, nous deciderons selon nos besoins. Si nous souhaitons une grande confiance dans nos résultats, alors nous accepterons 103 gènes avec les corrections de type FWER.
Si nous nous laissons au contraire une certaine tolérance par rapport aux faux positifs, alors nous accepterons 287 ou 695 gènes selon la correction de type FRD utilisée. À noter que la correction de Benjamini-Yekutieli s'utilise en pratique lorsqu'il n'y a pas indépendance entre les sites, ce qui est le cas ici, et est donc particulièrement adaptée à la situation pour tenir compte de la dépendance des sites.

5 Codes

5.1 Partie 1

Code 1 :

```
# Puissance
n<-30
f<-function(mu){
  X<-rnorm(n,mu,1)
  T<-sqrt(n)*(mean(X)-mu)/sqrt(var(X))
  #ifelse(mu>=0,1-pt(qt(0.95,n-1)-sqrt(n)*(mu)/sqrt(var(X)),n-1),(1-pt(
    qt(0.95,n-1)+sqrt(n)*(mu)/sqrt(var(X)),n-1)))
  return(1-pt(qt(0.975,n-1)-sqrt(n)*(mu)/sqrt(var(X)),n-1)+pt(-qt
    (0.975,n-1)-sqrt(n)*(mu)/sqrt(var(X)),n-1))
}
plot(seq(-2,2,0.01),f(seq(-2,2,0.01)),type='l',xlab='mu',ylab='P_H1(|T
    |>k_a)',ylim=c(0,1))
points(0,0.05,col='red',pch=19)
legend("bottomright", legend = c("n=30"))
```

Code 2 :

```
# Fonction p-valeurs
liste_pval<-function(n,N,mu=0,mu0=0){
  a<-c()
  for(i in 1:N){
    X<-rnorm(n,mu,1)
    T<-sqrt(n)*(mean(X)-mu0)/sqrt(var(X))
    p<-2*(1-pt(abs(T),n-1))
    a=append(a,2*(1-pt(abs(T),n-1)))
  }
  return(a)
}
# Affichage des resultats
hist(liste_pval(1000,1000),freq=FALSE,main ="Histogramme des p-valeurs"
    ,xlab="p-valeurs",ylab="Densite")
curve(dunif(x), add=TRUE, col="red") # La loi des p-val est une loi
    uniforme sur [0,1]
```

Code 3 :

```
# Fonction test de Student/Intervalle de confiance
Student<-function(n,mu,mu0,B){
  # Intervalle de confiance a 95% pour mu
  X<-rnorm(n,mu,1)
  mean<-mean(X)
  T<-sqrt(n)*(mean(X)-mu0)/sqrt(var(X))
  pval<-2*(1-pt(abs(T),n-1))
  print("Moyenne, T, pval")
  print(c(mean,T,pval))
  inter1<-c(mean-qt(0.975,n-1)*sqrt(var(X))/sqrt(n),mean+qt(0.975,n-1)*
    sqrt(var(X))/sqrt(n))
  print("Intervalle de confiance a 95% pour la moyenne")
  print(inter1)
  # Intervalle de confiance a 95% bootstrap pour la p-valeur
  a<-c()
  VecBoot<-replicate(B,sample(X,n,replace=TRUE))
  for(i in 1:B){
    Tb<-sqrt(n)*(mean(VecBoot[,i])-mu0)/sqrt(var(VecBoot[,i]))
    pb<-2*(1-pt(abs(Tb),n-1))
    a=append(a,pb)
  }
  a=sort(a) # Rangement de a par ordre croissant
  inter2<-c(a[ceiling(B*0.05/2)],a[ceiling(B*(1-0.05/2))]) # Intervalle
    de confiance bootstrap a 95% pour la pval
  print("Intervalle de confiance a 95% pour la pval")
  return(inter2)
}
```

Code 4 :

```
# Test de Student pour un echantillon de loi exponentielle
n<-100
mu=5
mu0=8
Y<-rexp(n,1/mu) # Moyenne de l'echantillon : mu
mean<-mean(Y)
t.test(Y,mu=mu0,conf.level=0.95)
T<-sqrt(n)*(mean(Y)-mu0)/sqrt(var(Y))
pval<-2*(1-pt(abs(T),n-1))
print(c(mean,T,pval))
sort(Y)
```

5.2 Partie 2

Code 5 :

```
# Tests multiples
methode1<-function(n,alpha,mu,sd,J){
  faux_pos<-0
  faux_neg<-0
  T<-rep(0,J)
  pval<-rep(0,J)
  decision<-rep(0,J)
  theta<-rep(0,J)
  for(i in 1:J){
    bruit<-rnorm(n,mu,sd) #vecteur
    theta[i]<-rbinom(1,1,0.05)
    Y<-theta[i]+bruit
    T[i]<-sqrt(n)*(mean(Y)-0)/sqrt(var(Y))
    pval[i]<-1-pt(T[i],n-1)
    if(pval[i] <= alpha){
      decision[i]<-1
    }
    else{
      decision[i]<-0
    }
    if(decision[i]==1 & theta[i]==0){
      faux_pos=faux_pos+1
    }
    if(decision[i]==0 & theta[i]==1){
      faux_neg=faux_neg+1
    }
  }
  print("Nb vrais sites, Resultats, Nb faux pos, Nb faux neg")
  return(c(sum(theta),sum(decision),faux_pos))
}
```

5.3 Partie 3

Code 6 :

```
# Methode Holm-Bonferroni
HB<-function(n,alpha,mu,sd,J,p){
  faux_pos<-0
  faux_neg<-0
  T<-rep(0,J)
  pval<-rep(0,J)
  decision<-rep(0,J)
```

```
  theta<-rep(0,J)
  for(i in 1:J){
    bruit<-rnorm(n,mu,sd) #vecteur
    theta[i]<-rbinom(1,1,p)
    Y<-theta[i]+bruit
    T[i]<-sqrt(n)*(mean(Y)-0)/sqrt(var(Y))
    pval[i]<-1-pt(T[i],n-1)
  }
  pval_s<-sort(pval)
  for(j in (J):1){
    if(pval_s[j]<alpha/(J+1-j)){
      decision[j]=1
      if(theta[order(pval)[j]]==0){
        faux_pos<-faux_pos+1
      }
    }
    else{
      decision[j]=0
      if(theta[order(pval)[j]]==1){
        faux_neg<-faux_neg+1
      }
    }
  }
  print("Nb vrais sites, Resultats, Nb faux pos, Nb faux neg")
  print(c(sum(theta),sum(decision),faux_pos,faux_neg,sum(decision) -
   faux_pos + faux_neg))
}
```

Code 7 :

```
# Methode Benjamini-Hochberg
benj<-function(n,alpha,mu,sd,J,p){
  faux_pos<-0
  faux_neg<-0
  T<-rep(0,J)
  pval<-rep(0,J)
  decision<-rep(0,J)
  theta<-rep(0,J)
  for(i in 1:J){
    bruit<-rnorm(n,mu,sd) #vecteur
    theta[i]<-rbinom(1,1,p)
    Y<-theta[i]+bruit
    T[i]<-sqrt(n)*(mean(Y)-0)/sqrt(var(Y))
    pval[i]<-1-pt(T[i],n-1)
  }
  pval_s<-sort(pval)
```

```
  for(j in (J-1):1){
    if(pval_s[j]*J/j<alpha){
      decision[j]=1
      if(theta[order(pval)[j]]==0){
        faux_pos<-faux_pos+1
      }
    }
    else{
      decision[j]=0
      if(theta[order(pval)[j]]==1){
        faux_neg<-faux_neg+1
      }
    }
  }
  print("Nb vrais sites, Resultats, Nb faux pos, Nb faux neg")
  print(c(sum(theta),sum(decision),faux_pos,faux_neg,sum(decision) -
   faux_pos + faux_neg))
}
```

Code 8 :

```
Stud_multi<-function(n,alpha,mu,sigma,J,p){
  T<-rep(0,J)   #Stat de test
  alpha_sid<-1-(1-alpha)**(1/J)  # Les nouveau niveau pour les
   corrections de Sidak et Bonferroni
  alpha_bonf<-alpha/J
  pvalm<-rep(0,J)
  decision<-rep(0,J)
  thetam<-rep(0,J)
  for(i in 1:J){ # Simulation des J echantillons
    bruit<-rnorm(n,mu,sigma**2) #vecteur
    thetam[i]<-rbinom(1,1,p)
    Y<-thetam[i]+bruit
    T[i]<-sqrt(n)*(mean(Y)-0)/sqrt(var(Y))
    pvalm[i]<-1-pt(T[i],n-1)}
  tableau2 <- data.frame(thetam,pvalm) # tableau des p-valeurs et
   valeur de theta

  pval2<-sort(pvalm) # On re ordonne les p-valeurs pour les correction
   Benjamini-Hochberg (BH), Holm-Bonferroni (HB) et Benjamini-Yekutieli
   (BY)
  pval_BH<-rep(0,J)
  pval_HB<-rep(0,J)
  pval_BY<-rep(0,J)
  Nbr_accept_sidak<-0
  Nbr_accept_bonf<-0
```

```
  Nbr_accept_HB<-0
  Nbr_accept_BH<-0
  Nbr_accept_BY<-0

  attach(tableau2)
  tableau <- tableau2[order(pvalm),] # on re ordonne le tableau en
   fonction des p-valeurs (ordre croissant)
  detach(tableau2)
lambda<-0
  for (i in 1:J){ # on ajuste les p-valeurs en fonction des correction
   ( HB, BH et BY)
    lambda<-lambda+1/i
    pval_BH[i]<-pval2[i]*J/i
    pval_HB[i]<-pval2[i]*(J-i+1)
    pval_BY[i]<-pval2[i]*J/lambda}
tableau$pval_BH<-pval_BH
tableau$pval_HB<-pval_HB
tableau$pval_BY<-pval_BY

for ( j in 1:J){   #On effectue les test
  if(tableau[j,2]<=alpha_sid){
    Nbr_accept_sidak<-Nbr_accept_sidak+1}
  if(tableau[j,2]<=alpha_bonf){
    Nbr_accept_bonf<-Nbr_accept_bonf+1}
  if(tableau[j,4]<=alpha){
    Nbr_accept_HB<-Nbr_accept_HB+1}
  if(tableau[j,3]<=alpha){
    Nbr_accept_BH<-Nbr_accept_BH+1}
  if(tableau[j,5]<=alpha){
    Nbr_accept_BY<-Nbr_accept_BY+1}}

fo_pos_sid<-0
fo_pos_HB<-0
fo_pos_BH<-0
fo_pos_bonf<-0
fo_pos_BY<-0

fo_neg_sid<-0
fo_neg_BH<-0
fo_neg_HB<-0
fo_neg_bonf<-0
fo_neg_BY<-0

for (i in 1:J){ # Calcul des faux positifs et negatifs pour chaque
   corrections
  if(tableau[i,1]==0){
```

```
    if(tableau[i,2]<alpha_sid){fo_pos_sid<-fo_pos_sid+1}
    if(tableau[i,2]<alpha_bonf){fo_pos_bonf<-fo_pos_bonf+1}
    if(tableau[i,4]<alpha){fo_pos_HB<-fo_pos_HB+1}
    if(tableau[i,3]<alpha){fo_pos_BH<-fo_pos_BH+1}
    if(tableau[i,5]<alpha){fo_pos_BY<-fo_pos_BY+1}}

  if(tableau[i,1]==1){
    if(tableau[i,2]>=alpha_sid){fo_neg_sid<-fo_neg_sid+1}
    if(tableau[i,2]>=alpha_bonf){fo_neg_bonf<-fo_neg_bonf+1}
    if(tableau[i,4]>=alpha){fo_neg_HB<-fo_neg_HB+1}
    if(tableau[i,3]>=alpha){fo_neg_BH<-fo_neg_BH+1}
    if(tableau[i,5]>=alpha){fo_neg_BY<-fo_neg_BY+1}}}

Sidak<-c(Nbr_accept_sidak,fo_pos_sid,fo_neg_sid)
BH<-c(Nbr_accept_BH,fo_pos_BH,fo_neg_BH)
HB<-c(Nbr_accept_HB,fo_pos_HB,fo_neg_HB)
Bonf<-c(Nbr_accept_bonf,fo_pos_bonf,fo_neg_bonf)
Vrai<-c(sum(thetam),0,0)
BY<-c(Nbr_accept_BY,fo_pos_BY,fo_neg_BY)
retour<-data.frame(Vrai,Bonf,Sidak,BH,HB,BY) # on renvoi un tableau (
   data frame) avec les resultats
return(t(retour))}
```

5.4 Partie 4

Code 9 :

```
# Code donnees reelles
data(golub)
row.names(golub)=golub.gnames[,3]
gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL","AML"))
golub[1,gol.fac=="ALL"]
boxplot(golub[720,] ~ gol.fac, method="jitter")
mean(golub[720,1:27])
mean(golub[720,28:38])
cat<-rep("A",27)
cat<-append(cat,rep("B",11))
plot(golub[720,],col=ifelse(cat == "A","blue","red"),pch=15)
legend("topright", legend = c("ALL","AML"), col = c("blue","red"), pch
   = c(15, 15))
##
plot(golub[,1], golub[,38], xlab = 'Patient 1 (ALL)', ylab = 'Patient
   38 (AML)',col=c("blue","red"),pch=c(15,15))
legend("topleft", legend = c("Patient 1","Patient 38"), col = c("blue",
   "red"), pch = c(15, 15))
```

```
##
hist(golub[,1:27],col="blue",xlab="ALL",ylab="Frequence",breaks=10)
hist(golub[,28:38],col="blue",xlab="AML",ylab="Frequence",breaks=10)
##
X<-rep(0,3051)
pval<-rep(0,3051)
for(i in 1:3051){
  test<-t.test(golub[i,1:27],golub[i,28:38], alternative = "two.sided",
   var.equal=FALSE,conf.level=0.95 )
  X[i]<-test$statistic
  pval[i]<-test$p.value
}
hist(X,breaks=100,xlim=c(-10,10),col="blue",xlab="T",ylab="Frequence")
hist(pval,breaks=100,col="blue",ylab="Frequence",xlab="pval")
##
pval_s<-sort(pval)
pval_bonf<-rep(0,3051)
res_bonf<-0
for(i in 1:3051){
  pval_bonf[i]<-pval_s[i]*3051
  if(pval_s[i]<=0.05/3051){
    res_bonf<-res_bonf+1
  }
}
pval_sid<-rep(0,3051)
res_sid<-0
for(i in 1:3051){
  pval_sid[i]<-1-(1-pval_s[i])^3051
  if(pval_s[i]<=1-(1-0.05)^(1/3051)){
    res_sid<-res_sid+1
  }
}
pval_HB<-rep(0,3051)
res_HB<-0
for(i in 1:3051){
  pval_HB[i]<-pval_s[i]*(3051+1-i)
  if(pval_s[i]<=0.05/(3051+1-i)){
    res_HB<-res_HB+1
  }
}
pval_BH<-rep(0,3051)
res_BH<-0
for(i in 1:3051){
  pval_BH[i]<-pval_s[i]*3051/i
  if(pval_s[i]<=0.05*i/3051){
    res_BH<-res_BH+1
```

```
  }
}
pval_BY<-rep(0,3051)
res_BY<-0
for(i in 1:3051){
  L<-3051*sum(c(1/(1:3051)))
  pval_BY[i]<-pval_s[i]*L/i
  if(pval_s[i]<=i*0.05/L){
    res_BY<-res_BY+1
  }
}
##
p.adjust(pval_s,method="bonferroni")
sum((p.adjust(pval_s,method="BH")<0.05))
sum((p.adjust(pval_s,method="BY")<=0.05))
sum((p.adjust(pval_s,method="bonferroni")<0.05))
##
plot(pval_s,ifelse(pval_bonf>=1,1,pval_bonf),col="#33FF00",type='l',
   xlab="p-valeurs triees",ylab="p-valeurs corrigees")
points(pval_s,ifelse(pval_sid>=1,1,pval_sid),col="#CCCC33",type='l')
points(pval_s,ifelse(pval_HB>=1,1,pval_HB),col="#333399",type='l')
points(pval_s,ifelse(pval_BH>=1,1,pval_BH),col="#990000",type='l')
points(pval_s,ifelse(pval_BY>=1,1,pval_BY),col="#FF9933",type='l')
legend("bottomright", legend = c("BONF","SID","HB","BH","BY"), col = c(
   "#33FF00","#CCCC33","#333399","#990000","#FF9933"), lty = c
   (1,1,1,1,1))

res<-data.frame(matrix(nrow=1,ncol=5))
rownames(res)<-c("Resultats")
colnames(res)<-c("BONF","SID","HB","BH","BY")
res[1,1]<-res_bonf
res[1,2]<-res_sid
res[1,3]<-res_HB
res[1,4]<-res_BH
res[1,5]<-res_BY
```