1 L'équipe BIOS au sein de l'Unité Mixte de Recherche : PRC (Physiologie de la Reproduction et des Comportements) de l'Institue Nationale de la Recherche Agronomique

1.1 Organisation de l'unité et de l'équipe

Il y a 11 équipes de recherche et une vingtaine d'étudiants en thèse dans cette unité. L'équipe BIOS (Biologie et Bioinformatique des Systèmes de Signalisation) est quant à elle constituée de 22 personnes.

1.2 Thèmes de recherches

L'équipe BIOS cherche à développer de nouvelles approches afin de contrôler l'activité de certains récepteurs. Dans cette équipe les recherches concentre sur l'étude des récepteurs FSH au sein des cellules germinales. L'hormone FSH est une hormone qui agit sur le fonctionnement de cellule impliquée dans la production de cellules reproductrices.

1.3 La modélisation, un point original de BIOS

Romain Yvinec et Anne Poupon sont les deux mathématiciens de l'équipe BIOS. Cette équipe développe des travaux de recherche qui se situes à l'interface entre l'informatique, la biologie et les mathématiques. Le développement de nouvelles technologies de mesure donne accès à de nombreuses données. Ce qui permet de modéliser des protéines, d'identifier de nouveaux gènes ou protéines ou de comprendre le fonctionnement de certains recepteurs. D'où l'utilisation de modèles mathématiques et de calcul numérique.

2 introduction a l'analyse différentielle protéomique et objectif du stage

2.1 Travail en amont : comprendre l'univers de l'analyse différentielle sur des données -omiques

A partir du mois de Janvier, j'ai commencé à lire des cours et des articles sur les données de RNAseq, et tout au long du stage, j'ai appris leur utilité ainsi que leur potentialité à travers des conversations ou des conférences.

2.2 Intitulé du stage et objectifs

L'intitulé du stage est le suivant : Analyse statistique de données de transcriptome, traductome, et protéome. Application au réseau FSH. L'objectif

du stage est multiple, il consiste dans une première partie à comprendre et à résoudre une variété de tests statistiques, il consistera en parallèle à comprendre la nécessite d'un ajustement du risque dans le cadre d'hypothèses multiples (FDR et FWER). Nous nous attarderons aussi sur les méthodes de normalisation et aux packages DESeq2 et EdgeR. Enfin si le temps le permet, nous pourrons essayer de mettre en évidence des chemins de signalisation à l'aide d'un logiciel Cytoscape.

3 Théorie des tests et comparaison des résultats obtenus en prenant en compte les hypothèses sous-jacentes

L'objectif de ces tests statistiques est d'identifier et de quantifier l'effet de l'hormone FSH sur la cellule de Sertoli. Pouvons-nous dire que l'expression des gènes est significativement différentes entre la situation stimulé et la situation contrôle. On quantifie l'expression d'un gène dans une cellule grâce à la quantité d'ARNm présente dans la cellule. Pour identifier l'ARNm au gène de départ, on utilise des méthodes de RNA sequencing (RNAseq) ou de sequençage à haut débit de l'ARNm.

Les données sur lesquels je travaille sont des données issues de RNAseq. Nous avions 4 répliquas pour chaque gènes (plus de 31000 gènes) dans les deux situations NS et FSH. Romain ayant identifié des anomalies dans un des répliquas, a décidé dans enlever un, ce qui laisse 3 répliquas pour chaque condition. De plus notre jeux de données est divisé en 3 fractions, libre (0 ribosome), 80s (1 ribosomes), polysomiale (¿ 2 ribosomes). Qui sont issues de la même expérience.

L'objectif commun est de comparer les moyennes de deux distributions entre deux situations donc HO : mNS = mFSH et H1 mNS != mFSH avec NS pour Non Stimulé et FSH pour stimuler à l'aide de l'hormone FSH.

Pour chaque test, il y a une partie théorique qui consiste a énoncer les hypothèses, donner un début de preuve et si possible le résoudre analytiquement. Il y a ensuite une partie appliquée dans laquelle je code le test sur R et le compare à une fonction prédéfinie.

3.1 Test de comparaison des variances

Pour pouvoir réaliser le test de Student, il est nécessaire de savoir si nos observations ont la même variance entre les deux conditions NS et FSH. En effet si non, il nous faut utiliser le test de Welch.

3.2 Test de Welch

Test qui s'appuie sur le comportement asymptotique de la statistique de test. Difficile à justifier théoriquement car on manipule des échantillons avec au maximum 3 individus (problème de coût) donc on peut qu'au prix d'erreur

d'approximation appliquer le Théorème Central Limite afin d'en conclure au rejet ou bien à la conservation de l'hypothèse H0. Nous avons décidé de tester la normalité de nos données afin de vérifier l'exactitude des résultats. Dans notre cas, il semblerait effectivement que l'on soit loin d'une distribution normale mais par contre très proche d'une distribution de poisson ou négative binomiale. Ainsi le test de Welch est obtenu grâce à la statistique suivante : Supposons que μ_1 et μ_2 représente les moyennes de nos deux populations sous NS et sous FSH. On veut tester les hypothèses suivantes :

$$H0 : \mu_1 = \mu_2 \tag{1}$$
$$H1 : \mu_1 ! = \mu_2 \tag{2}$$

On utilise la statistique de test suivante :

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{3}$$

Avec n1 le nombre d'observation dans la condition une et n2 le nombre d'observations dans la condition deux. Enfin, s_1 et s_2 sont les standards errors. Enfin, il nous faut estimer le degré de liberté à l'aide de l'équation de Welch-Sattertwaite grâce à l'équation suivante :

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{1}{n_1 - 1}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_2^2}{n_2}\right)^2}$$

3.3 Test du Maximum de Vraisemblance et Test de Wald

3.3.1 Test du Maximum de Vraisemblance

Étant donné que les données de concentration d'ARNm dans une cellule suivent des lois discrètes de Poisson ou Binomiale Négative Romain m'a proposé de construire le test du Ratio de Maximum de vraisemblance grâce à la vraisemblance obtenue avec des distributions de Poisson

Ainsi notre modèle de référence sera $\theta 1$ pour le paramètre de la loi de Poisson de la population de gènes NS et $\theta 2$ pour le paramètre de la loi de Poisson de la population de gènes sous FSH (stimulé):

$$X \sim P(\theta 1) \tag{4}$$
$$Y \sim P(\theta 2) \tag{5}$$

La fonction de vraisemblance est la suivante pour une loi de poisson avec 6 observations :

$$X = (x1, x2, x3) \tag{6}$$
$$Y = (y1, y2, y3) \tag{7}$$

$$L(\theta 1, \theta 2, X, Y) = \prod_{i=1}^{3} \exp(-\theta 1) \times \frac{\theta 1^{xi}}{xi!} \times \prod_{i=1}^{3} \exp(-\theta 2) \times \frac{\theta 2^{yi}}{yi!} \tag{8}$$

Pour obtenir le maximum de vraisemblance, on calcul le maximum du log de la fonction si dessus, ce qui donne :

$$\max_{\theta 1, \theta 2} \log(L(\theta 1, \theta 2, X, Y)) \tag{9}$$

On obtient le maximum grâce aux dérivées partielles de la fonction de vraisemblance, ce qui donne :

$$\frac{\partial \log(L(\theta 1, \theta 2, X, Y))}{\partial \theta 1} = 0 \; ie \; -3 + \sum_{i=1}^{3} \frac{xi}{\theta 1} = 0 \; donc \; \frac{\sum_{i=1}^{3} xi}{3} = \theta 1 \tag{10}$$

De même par symétrie :

$$\frac{\partial \log(L(\theta 1, \theta 2, X, Y))}{\partial \theta 2} = 0 \; ie \; -3 + \sum_{i=1}^{3} \frac{yi}{\theta 2} = 0 \; donc \; \frac{\sum_{i=1}^{3} yi}{3} = \theta 2 \tag{11}$$

Ainsi on obtient les estimateurs que l'on obtient grâce à la méthode des moments

On peut maintenant calculer le test du ratio de maximum de vraisemblance, sous les hypothèses suivantes :

$$\text{H0} : \theta 1 = \theta 2$$
$$\text{H1} : \theta 1 \neq \theta 2$$

Dans notre cas $\Theta 1 \in R^2$ et $\Theta 2 \in R^3$ donc puisque le test du ratio de vraisemblance suit une loi du khi2(à démontrer) de degré de liberté égal à :

$$df = dim \; \Theta 2 - dim \; \Theta 1 \tag{12}$$

La valeur du test est la suivante :

$$\Phi = 2 \times log(\frac{\sup_{\theta \in \Theta 1 \cup \Theta 2} L(\theta 1, \theta 2, X, Y)}{\sup_{\theta \in \Theta 1} L(\theta 1, \theta 2, X, Y)}) \sim \chi^2_{df} \tag{13}$$

On peut le démontrer

Dans notre cas sous H1:

$$\widehat{\theta 1} = \frac{\sum_{i=1}^{3} x_i}{3} \tag{14}$$

$$\widehat{\theta 2} = \frac{\sum_{i=1}^{3} y_i}{3} \tag{15}$$

Dans notre cas sous H0, puisque $\theta 1 = \theta 2$.
On pose V = (x1,x2,x3,y1,y2,y3):

$$\widehat{\theta 0} = \frac{\sum_{i=1}^{6} v_i}{6} \tag{16}$$

Donc la valeur de notre test pour chaque gène est la suivante:

$$\Phi = 2 \times log(\frac{L(\widehat{\theta 1}, \widehat{\theta 2}, X, Y)}{L(\widehat{\theta 0}, V)} \sim \chi_1^2 \tag{17}$$

On l'obtient grâce au code suivant:

```
ratiotest <- function(x1,x2,x3,y1,y2,y3){
        l0 = sum(x1,x2,x3,y1,y2,y3)/6
        l1 = sum(x1,x2,x3)/3
        l2 = sum(y1,y2,y3)/3
        v <- c(x1,x2,x3,y1,y2,y3)
        L0 <- 1
        L1 <- 1
        for (i in 1:3){
                L0 <- L0 * (exp(-6*l0)* l0^(v[i]+v[3+i]))/
                (factorial(v[i])*factorial(v[3+i]))
                L1 <- L1 * (exp(-3*l1)*l1^v[i]/factorial(v[i])*
                exp(-3*l2)*l2^v[i+3]/factorial(v[3+i]))
                }
        T <- 2*log(L1/L0)
        p_value <-  1-(pchisq(T,1))
        return(p_value)}
```

On compare les valeurs obtenus grâce à la fonction prédéfinie de R à l'aide d'un modèle Linéaire Généralisé qui admet des spécifications concernant la distribution, dans notre cas une distribution de Poisson.

Notre modèle est alors, Y pour le nombre de reads pour chaque gènes dans les différents échantillons pour lib,pol et 80s, X1 pour chaque gènes NS = 1 et

FSH = 0, X2 pour chaque gènes NS = 0 et FSH = 1 et X0 pour chaque gènes 1 pour tous les répliquas.

Par exemple :

Si x1,x2,x3 donnent les comptes non stimulés pour les trois répliquas obtenus pour un gène
Si y1,y2,y3 donnent les comptes stimulés pour les trois répliquas obtenus pour un gène

Y = (x1,x2,x3,y1,y2,y3)
X1 = (1,1,1,0,0,0)
X2 = (0,0,0,1,1,1)
X0 = (1,1,1,1,1,1) = pas de différence entre non stimulé et stimulé

Notre modèle sous H1 est le suivant :
$Y_i = a \times X1_i + b \times X2_i + \epsilon_i$ avec ϵ pour le terme d'erreur et i pour le i ème gènes

Sous H0, c'est :
$Y_i = (a+b) \times X0_i + \epsilon'_i$

Ainsi on cherche à estimer les paramètres a et b, et on cherche à savoir si ces paramètres sont significativement différents ou non. Le code suivant renvoi la p-valeur.

```
ratiotestR <- function(x1,x2,x3,y1,y2,y3){
        Y = c(x1,x2,x3,y1,y2,y3)
        X0 = c(0,0,0,1,1,1)
        X1 = c(1,1,1,0,0,0)
        X2 = c(1,1,1,1,1,1)
        model0 <- glm(Y ~ X2-1, family = poisson )
        model1 <- glm(Y ~ X0 + X1-1, family = poisson )
        A <- anova(model0, model1, test="LRT")
        T <- lrtest(model0,model1) #les deux sont possibles
        p_value <- A$`Pr(>Chi)`
        return(p_value[2])}
```

3.3.2 Test de Wald

Ce test s'appui les maximums de vraisemblance des deux distributions ie $\theta 1$ et $\theta 2$ qui sont estimés grâce aux moyennes empiriques (11) et (12).

L'objectif de ce test est de comparer sous H0 les deux moyennes et de statuer si elles sont significativement différentes ou non. Nos hypothèses sont les suivantes :

$$\text{H0} : \theta 1 = \theta 2$$
$$\text{H1} : \theta 1 \neq \theta 2$$

Pour ce faire la statistique de Wald est la suivante :

$$W = n_{\cdot} \times (R^t\theta - r)(R^t \times I^{-1}(\theta) \times R)^{-1}(R\theta - r) \sim \chi^2_{dim(R)} \tag{18}$$

Avec : R = $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$; $\theta = \begin{pmatrix} \theta 1 \\ \theta 2 \end{pmatrix}$; r = 0 ; la matrice de Fisher est obtenu en calculant :

$$I(\theta)_{ij} = -\frac{1}{n}\mathbb{E}(\frac{\partial^2 \log(L(\theta, X, Y))}{\partial\theta_i\, \partial\theta_j}) \tag{19}$$

n = le nombre de répliquas total Avec :

$$L(\theta, X, Y) = \sum_{i=1}^{3} \exp^{-(\theta 1+\theta 2)} + \sum_{i=1}^{3} x_i \times \log(\frac{\theta 1}{x_i!}) + y_i \times \log(\frac{\theta 2}{y_i!}) \tag{20}$$

$$\frac{\partial L(\theta, X, Y)}{\theta 1} = -3 + \sum_{i=1}^{3} \frac{x_i}{\theta 1} \tag{21}$$

$$\frac{\partial L(\theta, X, Y)}{\theta 2} = -3 + \sum_{i=1}^{3} \frac{y_i}{\theta 2} \tag{22}$$

Donc :

$$\frac{\partial^2 \log(L(\theta, X, Y))}{\partial^2\theta_1} = -\sum_{i=1}^{3} \frac{x_i}{\theta_1^2} \tag{23}$$

$$\frac{\partial^2 \log(L(\theta, X, Y))}{\partial^2\theta_2} = -\sum_{i=1}^{3} \frac{y_i}{\theta_2^2} \tag{24}$$

Pour les deux autres dérivées partielles, on obtient 0, donc la matrice d'information de Fisher est une matrice diagonale de la forme :

$$I(\theta) = \begin{pmatrix} \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{pmatrix} \tag{25}$$

On utilise des estimateurs de θ, on sait que c'est la moyenne empirique. En définitive notre statistique de test suit une loi du khi deux à 2 - 1 degré de liberté.

3.4 Test exact de Fisher

L'objectif d'identifier des gènes differentiellement exprimés est de pouvoir ensuite faire une analyse plus approfondie sur leurs rôles biologique dans la cellule. Pour ce faire la statistique de test de Fisher a l'avantage d'être une statistique exacte qui ne repose pas sur des résultats asympotitiques, très avantageux en RNAseq car nous n'avons pas beaucoup de répliquas car ils sont très couteux à obtenir. Cependant, il semblerait que ce test soit interessant justement lorsque l'on a qu'un seul répliqua. En effet on le calcul grâce à la table de contingence suivante :

	NS	FSH
k_{ijl}	k_{ijNS}	k_{ijFSH}
K_{-ijl}	K_{jNS}	K_{jFSH}

Avec k_{ijNS} le nombre de compte pour le gène i dans la condition NS et le répliquas j, k_{ijFSH} le nombre de compte pour le gène i dans la condition FSH, K_{jNS} pour la taille de la librairie dans la condition NS pour le répliquas j sans le nombre de compte pour le gène i. De même, K_{jFSH} est la taille de la librairie dans la condition FSH pour le répliquas j sans le nombre de compte pour le gène i. Ainsi on compare si grâce à la statistique de R, Fisher.test les deux distributions en faisant le ratio.

4 Prise en compte de la multiplicité des Tests

4.1 FDR : basic concept

L'intérêt des méthodes qui prennent en compte la multiplicité des tests est de contrôler pour un nombre important de test le risque de première espèce, c'est à dire le nombre de faux négatifs

Selon Anat Reiner adn Yoav Benjamini[1] dans leur article, lorsque beaucoup d'hypothèses sont testés, la probabilité que une erreur de type 1 soit commise augment grandement avec le nombre d'hypothèses. Ainsi en Microarray le problème est saillant, en effet nous avons plus de 24000 gènes pour lesquels on veut savoir si ils sont différentiablement exprimés ou non, c'est à dire que nous avons 24000 tests à réaliser, dont l'indépendance est un autre problème. En effet dans une cellule l'expression d'un gène est co-régulé, en effet on peut considérer qu'il y a une certaine quantité d'ARNm qui peut être produite dans la cellule si un gène est très fortement exprimé par rapport aux autres alors les autres gènes verront leur expression diminuer. Il y a donc une interdépendance entre l'expression des différents gènes et donc entre les tests que nous faisons.

Le False Discovery Rate est la proportion observée de test rejetant H0 alors que H0 est vrai sur le nombre d'hypothèses rejetées

Contrôler le False discovery Rate permet de révéler la proportion d'argent

investie inutilement dans la première étape sans pour autant annuler tout si il y a une seule erreur de type 1.

	Conserve H0	**Rejette H0**	**Total**
H0 vraie	U	V	m0
H1 vraie	T	S	m - m0
	m - R	R	m

4.2 Comment calculer sur un jeux de données le False Discoverie Rate

Théoriquement : FDR = $\mathbb{E}$(V/R) si $R > 0$ sinon 0
On peut donner une première interprétation du FDR, si le FDR est contrôlé à un certain niveau α alors la probabilité de détecter un gène différentiellement exprimé qui ne l'ai pas est inférieur à α.

Pour calculer le FDR, nous allons utiliser la première méthode trouvée par Benjamini et Hochberg(1995) qui consiste à utiliser les p-valeurs ordonnées pour trouver k.
Chaque p-valeur est comparée, pour un niveau de FDR q, à q×i/m.
m = le nombre d'hypothèses.

$$k = \max\{i : P_{(i)} \leq q \times \frac{i}{m}\} \tag{26}$$

On rejette alors toutes les hypothèses $H_{(1)}, H_{(2)}, ..., H_{(k)}$.

Permet de bien contrôler le risque de première espèce si les tests sont indépendants entre eux. Cependant il existe des méthodes de permutation qui réintroduisent de l'indépendance dans des données liées.

Notamment grâce aux méthode de boostrap, l'avantage de cese méthodes est de permettre de répondre à des problèmes statistiques qui ne peuvent pas être résolu par la théorie à grands renfort de calculs sur ordinateurs.
Explication naïve : Bootstrap méthode permet d'obtenir un estimateur à l'aide de tirages aléatoires au sein de l'échantillon. D'un échantillon, on construit plusieurs échantillons en tirant aléatoirement dans un unique échantillon de départ.
Bootstrap permet d'obtenir quelque chose de normal, s'appui sur le principe de tirage aléatoire avec remise dans un échantillon qui est lui un échantillon d'une population d'intérêt.

L'algorithme ci-dessous renvoi pour un vecteur de p-valeurs et un niveau de risque α donnés la valeur du False Discovery Rate selon la méthode ci-dessus.

```
FDR <- function(vect, alpha){
        m <- length(vect)
        fdr = c(0,0)
        for (i in 1:length(vect)){
                if (vect[i] <= i*alpha/m){
                        fdr[1]= i
                        fdr[2]= vect[i]
                }
        }
        return(fdr)}
```

Cette méthode renvoi la même valeur du False Discovery Rate que celle déjà programmée dans R

On peut aussi chercher à calculer la p-valeurs ajustée grâce à une autre méthode qui consiste à calculer la p-valeurs ajustée pour chaque p-valeur suivant la formule:

$$P_j^{BH} = min_{j \leq i}\{P_i \times \frac{m}{i}\} \tag{27}$$

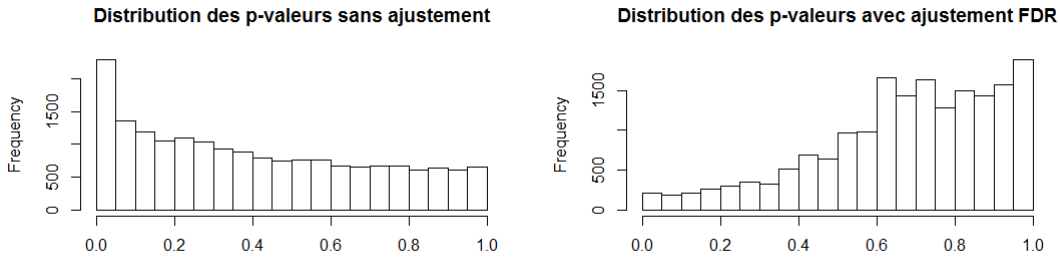
Ce qui donne l'algorithme suivant:

```
adjpvalue <- function(pvalue){
                pvalue <- sort(pvalue)
                pvalue1 <- rep(0,length(pvalue))
                for (i in 1:m){
                        p = 1
                        for (j in i:m){
                                mi =min((m/j)*pvalue[j],1)
                                if (mi<= p){
                                        p = mi}}
                        pvalue1[i] = p}
                return(pvalue1)}
```

4.3 Application

Nous pouvons comparer les résultats en terme de p-valeurs avec ou sans la méthode de Benjamini et Hochberg. Nous allons utiliser, les p-valeurs obtenus grâce à DESeq2 sur la fraction polysomiale.
Lors de notre étude, bien que beaucoup plus rigoureux de prendre en compte l'aspect multiple des tests, nous avons choisi de travailler avec les p-values non ajustées. En effet, on peut comparer les distributions des p-value sans et avec ajustement, on remarque qu'il y a beaucoup moins de petite p-value après ajustement. On utilise les p-values obtenues sur la fraction Polysomiale

grâce à DESeq2. On divise par 10 le nombre de p-value en dessous du seuil 0.05, de 2286, on passe à 215.

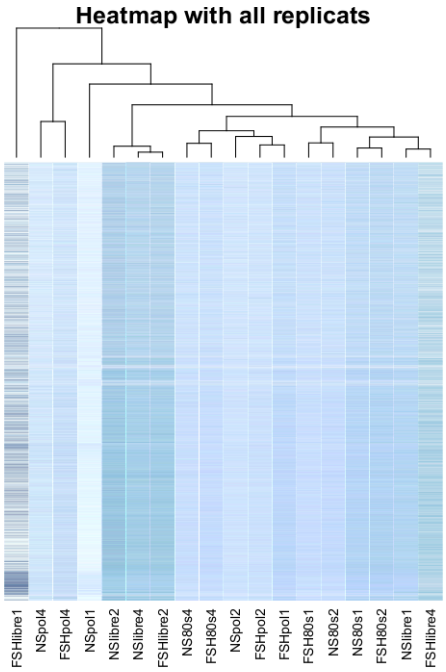


5 Observation qualitative des données

Dans cette section nous allons refaire la démarche suivie par Romain Yvinec afin de regrouper les répliquas ente eux. Nous utilisons l'Analyse en Composante principale, le clustering avec la distance euclidienne et une visualisation des différents groupes à l'aide d'un heatmap.

5.1 Heatmap

Un Heatmap est un graphique en trois dimensions, les counts pour chaque gènes et pour chaque répliquas sont représentés à l'aide d'une couleur, plus elle est foncée plus la quantité d'ARNm pour le gène i dans le répliqua j est grande. Sur les axes des abscisses et des ordonnées, l'algorithme faire des groupes.

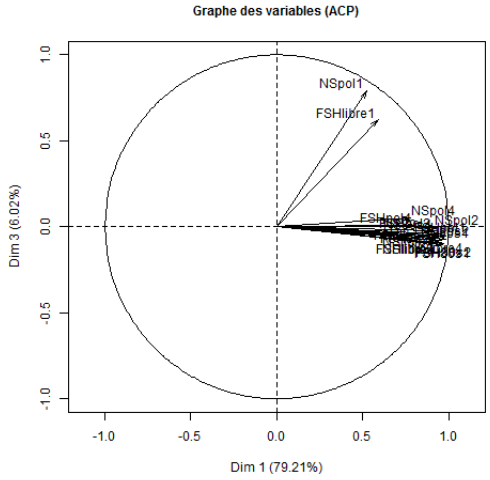


On remarque que plus la case pour g_{ij} est bleu plus le nombre de counts pour ce gène est important. De plus on constate qu'il y a deux colonnes qui sont isolées celle de NSpol1 et FSHlibre1, tandis que les autres sont relativement similaire.

Nous allons voir que l'Analyse en composante principale nous donne des résultats qui corroborent cette première analyse.

5.2 ACP

L'analyse en composante principale utilise la matrice de variance-covariance afin de réduire le nombre de vecteurs à 2 ou 3 pour pouvoir les analyser. Dans notre cas on remarque que le vecteur propre associé à la troisième composante principale est lié essentiellement aux deux répliquas FSHlibre1 et NSpol1. Ainsi cela nous permet d'identifier des répliquas qui ne se comportent pas comme le reste des répliquas. C'est à dire dans note cas répliquas, à calculer les valeurs propres associées aux vecteurs afin d'en faire une typologie.

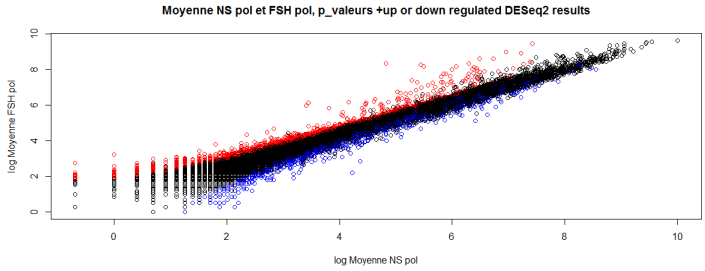


Les graphiques précédent nous ont amené à supprimer deux répliquas de notre analyse statistique NSpol1 et FSHlib1

5.3 Quelle est la meilleur représentation des données afin d'avoir des groupes qui ont un sens ?

Nous nous intéressons maintenant aux données sans NSpol1 et FSHlibre1, comment représenter les données pour e, faire des groupes les plus homogènes possible. Nous pouvons comparer une représentation directement avec les réplicats et avec les moyennes des réplicats pour chaque groupe. On obtient avec la deuxième représentation des groupes plus cohérent. On constate que les lib,pol et 80s FSH et pol sont rassemblés ce qui semble vouloir dire que la variabilité entre les différents groupes de ribosomes est plus forte que la distance entre FSH et Pol. Pour faire les groupes, on utilise la méthode hclust de R. est-ce que j'introduis les clusters ou non ?

Un des moyen de représenter nos résultats est de superposer plusieurs dimension. En effet, on peut vouloir vérifier, si les gènes différentiellement exprimée se sont bien ceux qui ont un log de la moyenne des repliquas plus élevé ou plus faible dans une condition plutôt qu'une autre. C'est à dire, est-ce que es gènes qui sont dit diffférentiellement exprimés sont loin de l'axe de la regression linéaire. On peut constater que c'est globalement le cas.



En bleu se sont les gènes down regulated dont le logFC obtenu avec DESeq2 est négatif et la p-valeurs associé à ce gène est inférieur ou égale à 0.05. Inversant en bleu se sont les gènes up regulated, c'est à dire dont le log de Fold Change est significativement différents de 0, au ris que 0.05.

6 Méthodes de normalisation des données

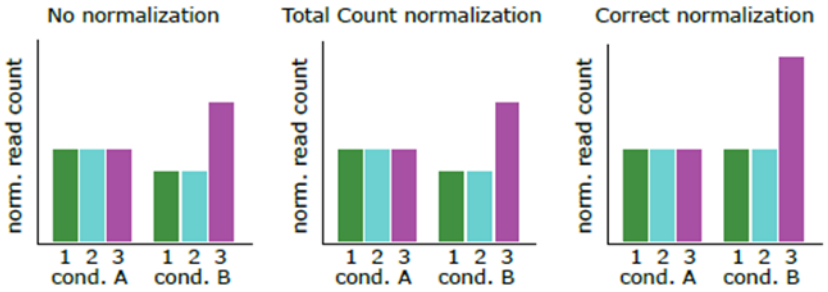
6.1 Pourquoi normaliser les données issues de RNAseq

L'objectif de toutes les méthodes de normalisation est de réduire autant que faire ce peux l'impact des variations aléatoire lié aux conditions biologiques différentes entre les expériences, afin d'identifier les observations qui ont réellement changées d'É entre la condition stimulé et non stimulé. Dans notre cas, cela se résume à une augmentation ou une diminution significative du nombre de Reads ou à un Fold change significativement différent de 0 pour chaque gène.

Il y a des facteurs aléatoires qui viennent affecter l'analyse différentielle entre les gènes au sein d'un même échantillon. Par exemple la taille des ARNm ou la présence de GC, augmentent soit la stabilité des séquences nucléotidiques ou simplement la zone de lecture et de ce fait augmente le nombre de lectures de ces ARNm lors du séquençage.

Il y a aussi des facteurs biologiques ou techniques qui affectent le nombre de Reads entre deux échantillons, l'objectif de la normalisation peut-être de remédier à ces différences entre échantillons. Une bonne normalisation doit permettre que les gènes non-différentiellement exprimés aient la même quantité de reads entre les différents échantillons.

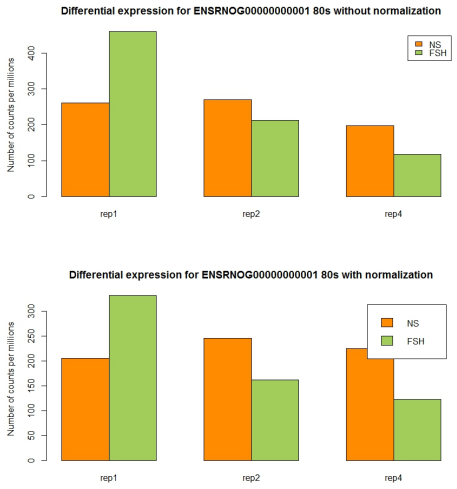
Bullard et al ont montré que les méthodes de normalisation en RNA sequencing était un des facteurs déterminants les résultats de l'expérience. D'où la nécessité de s'y intéresser.



Graphique de Ciaran Evans, Johanna Hardin et Daniel Stoebel

Ce graphique illustre l'objectif de la normalisation des données de RNAseq. En effet, après normalisation, les gènes qui sont réellement différentiellement exprimés sont facilement identifiable et ne sont pas pollués ps d'autres gènes qui aurait été diffférentiellement exprimé seulement à cause de variation expérimentales. Faire avec mes données le même graphique pour un gène donné. A commenter.

J'ai observé l'effet de la normalisation de EdgeR sur les Compte par Million des échantillons 80s, cela donne les deux graphiques suivant pour le premier gène de notre base de donnée:



6.2 Comment normaliser les données ?

Le nombre de reads permet de quantifier le niveau d'expression d'un gène. On peut considérer celui-ci comme une variable aléatoire. Les méthodes de normalisation forment deux groupes:

- la normalisation à l'intérieur d'un même échantillon, nécessaire du fait de la variabilité d'expression et de lectures des gènes induite par la longueur des gènes ou par la présence prononcé de liaison GC.
- La normalisation entre différents échantillons, nécessaire du fait de la variabilité d'expression et de lectures des gènes induite par les facteurs techniques(environnements, manipulation)

Dans cette section nous nous intéresserons essentiellement à la variabilité entre les échantillons, c'est à dire nous ne voulons pas analyser des résultats dans lesquelles la différence d'expression entre deux échantillon serait dû à des différences techniques entre échantillons.

Comme on fait Ciaran Evans,Johana Hardin et Daniel Stoebel, nous souhaitons nous intéresser aux méthodes de normalisation entre échantillon et à leurs hypothèses sous-jacentes. Ainsi il existe trois méthodes de normalisation : la normalisation par la taille de l'échantillon (library size), la normalisation par la distribution, la normalisation en s'appuyant sur des gènes contrôles.

6.2.1 Filtrer les données

Cette étape sert à retirer les gènes peu exprimés ou même ayant aucun ARNm séquencés. Notre critère de filtre est le suivant : on retire tous les gènes dont le compte par million pour un répliquas est inférieur à deux et ont le compte par million au niveau du gène est inférieur à 3.

6.2.2 La normalisation par la taille de l'échantillon

L'hypothèse principale est que l'expression totale des gènes entre les échantillons est la même. Cette méthode est relativement simple, elle consiste à normaliser par la taille de l'échantillon chaque échantillon. Cf [mette la référence] voir comment on calcule le counts par million (cpm) à l'intérieur d'un échantillon.

6.2.3 La normalisation par la distribution

Méthode qui repose sur le fait que les gènes non-différentiellement exprimés et différentiellement exprimés (DE) suivent a même loi de probabilité et ont donc la même distribution.

Une des méthodes est la normalisation par les quantiles, elles consiste à imposer une certaine distribution aux données en remplaçant les quantiles des différents échantillon par des quantile similaires entre échantillons qui est la moyenne pour chaque quantiles des quantiles de tous les échantillons. Deux autres méthodes cherchent à calculer un facteur de taille pour chaque échantillon, c'est le cas des méthodes employés par EdgeR et DESeq. Pour DESeq celui-ci est calculé de telle manière à ce que les différents facteurs représentent les "sequencing depth" des différents échantillons.

Soit $k_{i,j}$ le nombre de read identifiés pour un gène i issu d'un échantillon j. L'expression du facteur de taille $\hat{s}_j$ pour DESeq est la suivante :

$$s_j = median_i\{\frac{k_{ij}}{(\prod_{v=1}^{m} k_{iv})^{1/m}}\} \tag{28}$$

m est le nombre d'échantillon, c'est la médiane par rapport aux gènes d'un échantillon normalisé par un facteur commun à tous les échantillons qui est $(\prod_{v=1}^{m} k_{iv})^{1/m}$.

Pour TMM (Trimmed Mean of the M-values), la méthode employée par EdgeR, il utilise le même principe, en choisissant un échantillon de référence ; et compare le nombre de counts pour chaque échantillon par rapport à l'échantillon de référence afin de calculer le size factor.

6.3 Comparaison des méthodes de normalisation employées par : EdgeR, DESeq

Avant de normaliser les données, ces packages permettent grâce aux compte par millions (cpm) de supprimer les lignes qui n'ont pas beaucoup de compte.

Ce choix est justifié par le fait qu'un gène avec peu de compte en global ou dans certains répliquas aura de toute façon moins de chance d'être différentiellement exprimés et il biaise les conclusion car un changement même faible pour un gène qui n'a que 10 counts pour tous les répliquas NS et FSH compris aura un impact fort sur le test qui nous permet de déterminer si le gène est différentiellement exprimés ou non. En effet un changement de 10 revient à doubler le nombre de compte entre les deux situations.
Ainsi, on retire tous les gènes qui ont soit des comptes par millions faible au niveau de tous les gènes soit des comptes par million faible au niveau au moins d'un répliquas. Le compte par million est déjà une forme de normalisation car il est égale à :

$$Kpm_{i,j} = 10^6 * \frac{c_{i,j}}{lib.size_j} \tag{29}$$

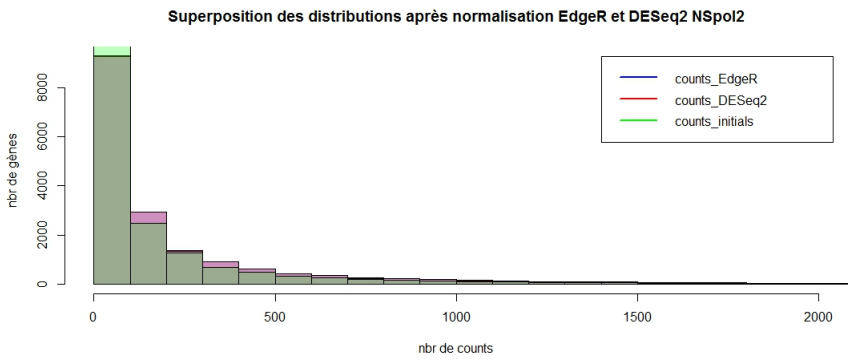
Concernant les notations, on utilise $c_{i,j}$ pour le nombre de reads pour un gènes i et un répliquas j, $\rho_j = 1$ si situation stimulé et égal 0 si situation non stimulé. $Lib.size_j = \sum_{i=1}^{length(rep_j)} c_{i,j}$.

EdgeR et DESeq2 ont tous les deux une méthode de normalisation des données dite globale, c'est à dire qu'ils appliquent un facteur d_j qui s'applique au niveau du répliquas. Il s'intitule le size factor dans les deux packages.

On obtient les facteurs de normalisation suivant pour EdgeR et DEseq2, on retrouve bien pour edgeR un échantillon qui sert de référence c'est le FSH80s répliqua 4.

NSlib1	**NSlib2**	**NSlib4**	**FSHlib2**	**FSHlib4**	**NSpol2**	**NSpol4**	**FSHpol1**
DESeq							
1,38	1,99	1,70	1,76	1,55	0,66	0,37	1,00
EdgeR							
1,26	1,28	1,25	1,28	1,26	0,88	0,43	0,97
FSHp2	**FSHpol4**	**NS80s1**	**NS80s2**	**NS80s4**	**FSH80s1**	**FSH80s2**	**FSH80s4**
DESeq							
0,73	0,58	1,20	0,88	0,80	0,84	1,33	0,85
EdgeR							
0,86	0,50	1,18	1,12	0,95	1,26	1,24	1,00

Pour savoir si notre jeu de donnée est bien normalisé, on peut faire plusieurs choses : une première technique utilisée dans la plus part des articles qui comparent les méthodes de normalisation est de dessiner un graphique avec en ordonnée le True positve Rate et en abscisse le False Positive Rate, ce qui donne la courbe de ROC. Le True positive qui est (cf Figure) égale à S/R, c'est à dire sur le nombre d'hypothèse rejetée alors que H1 est vrai, c'est la puissance du test. Le False Positive rate, c'est le risque de première espèce, c'est à dire le nombre de test qui rejette l'hypothèse H0 alors que celle-ci est vraie. Cependant



dans notre cas, nous n'avons pas de génome de référence et nous n'avons pas décidé de simuler nos variables aléatoire.

Ainsi, nous avons décidés de comparer les méthodes de normalisation à l'aide de trois critères, en applicant la même statistiques de tests aux deux jeux de données :

- Comparer les distributions des deux jeux de données pour un répliquas.
- Le nombre de gène significativement différentiellement exprimés pour les deux méthodes.
- Comparer les distributions des p-valeurs entre les deux jeux de données avec le même test.

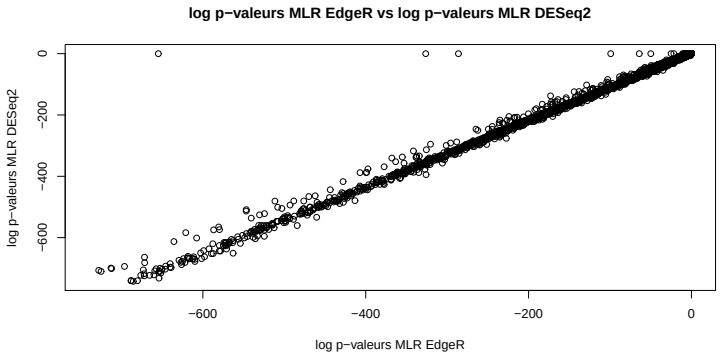
En premier, nous pouvons comparer la distribution des counts normalisés d'un répliqua entre deux méthode de normalisation. Nous nous sommes intéressés à l'échantillon NSpol2 et nous obtenons l'histogramme des distributions suivant qui montre qu'il n'y a pas de différences fondamentales entre les deux distributions après normalisation, on superpose aussi les counts non normalisés en vert.

On se rend compte que les distributions avant normalisation et après normalisation ne sont pas du tout les mêmes, par contre la différence entre la distribution des counts dans NSpol2 normalisés par EdgeR et DESeq2 est très faible.En effet, on ne voit qu'une toute petite partie en noir sur les graphiques où celles-ci ne se superposent pas.

On utilise ensuite le ratio du maximum de vraisemblance et la partie polysomiale des ARNm pour comparer le nombre de gène différentiellement exprimés entre les deux méthodes, communs aux deux méthodes. Pour EdgeR, on détecte 9298 gènes DE. Pour DESeq2, on détecte 9457 gènes DE. On détecte 8917

gènes différentiellement exprimés communs aux deux méthodes de normalisation toujours avec le maximum de vraisemblance. On dit qu'un gène est différentiellement exprimé si sa p-valeur est en dessous de 0.05, car à ce moment on peut rejeter l'hypothèse H0 et dire que les moyennes sont significativement différentes entre NSpol et FSHpol. On voit que la méthode de normalisation de DESeq2 permet de détecter un peu plus de gène différentiellement exprimé sur notre jeu de données. Seulement, on observe aussi pour une grande majorité des gènes différentiellement exprimés qu'il le sont pour les deux méthodes.

Enfin, nous sommes amenés à comparer directement les valeurs des p-valeurs obtenues en faisant le même tests sur les deux jeux de données normalisés, j'ai décidé de représenter le graphique en log, car il y beaucoup de p-valeurs proches de 0 comme le montre le paragraphe ci-dessus. Le R^2 ajusté entre les deux vecteurs de p-valeurs EdgeR et DEseq2 est de 0.8903:



On peut donc conclure que les méthodes de normalisation de DESeq2 et EdgeR, donnent des résultats très proches aussi bien en terme de distributions, que de nombre de gènes différentiellements exprimées, que de valeur de p-valeurs.

6.4 Aperçu du rôle de la normalisation sur les résultats obtenus

On s'est demandé en comparant les p-valeurs obtenus par différents test, si les résultats en terme de p-valeurs avec le même test mais avec des méthodes de normalisation différentes : par le sizefactor de DESeq2 ou par la taille de la librairie étaient très différents. On obtient les résultats suivant :

7 Comparaison des statistiques de tests sur l'ensemble des gènes filtrés

L'objectif de cette section est d'appliquer les statistiques de tests développés en première partie du stage afin d'identifier dans l'ensemble des gènes ceux

qui sont dit différentiellement exprimés par différents tests statistiques. Cela va nous permettre de comparer le résultat des différentes statistiques de test sur un même jeu de données. On aura souci de garder la comparaison avec le test employé par DESeq2. Les test que nous utilisons dans cette partie sont les suivants : Le t-test, le test exact de Fisher avec normalisation par la taille de la library, le test du maximum de vraisemblance avec loi de poisson et loi binomiale négative, le test de Wilcoxon et les résultats du test de Wald calculé par le package DESeq2. Pour obtenir des résultats proche, nous utiliserons pour une partie des résultats les facteurs de normalisation calculés par DESeq2.

Un des premier résultats est d'avoir réussi à retrouver les mêmes p-valers que celles trouvées par le package. En effet DESeq2 et EdgeR utilisent tous les deux un modèle linéaire généralisé avec des lois négatives binomiales afin d'estimer la valeur du log(FC). Le modèle employé par DEseq2 est le suivant :

$$\begin{aligned} K_{ij} &\sim NB(\mu_{ij}, \alpha_i) \\ \mu_{ij} &= s_j q_{ij} \\ \log(q_{ij}) &= x_j \beta_i \end{aligned}$$

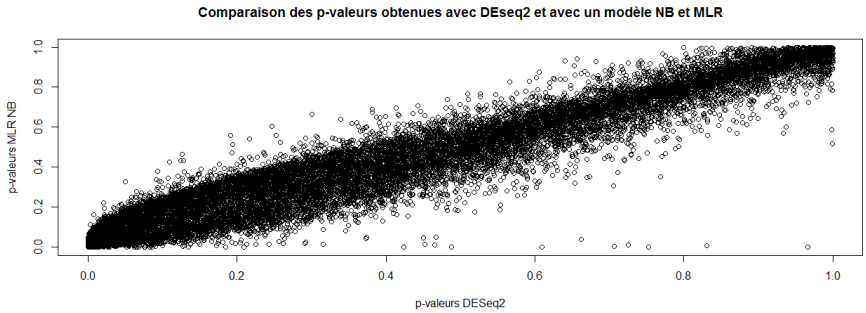
Avec K_{ij} pour le gène i et le répliqua j qui est modélisé par une distribution d'une variable aléatoire qui suit une loi négative binomiale avec μ_{ij} la moyenne et α_i le paramètre de dispersion est au niveau d'un gène. Leur modèle prend en compte un paramètre de normalisation pour chaque répliqua, s_i. Donc ils cherchent à estimer les paramètres β_i pour chaque gènes. La loi négative binomiale est une loi à deux paramètres r et p avec $0 <= p <= 1$, elle représente la probabilité d'obtenir r-1 succès et x échecs sur x+r-1 tentatives. la densité d'une loi binomiale négative est la suivante :

$$P(X = x) = \binom{x + r - 1}{r - 1} p^r (1 - p)^x \tag{30}$$

Les moments associés à cette loi sont :

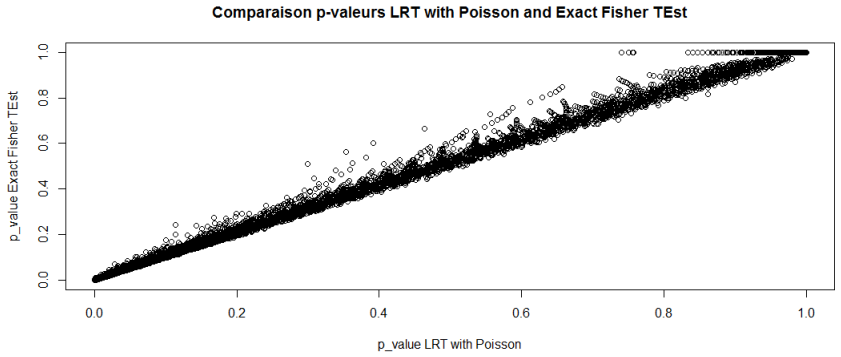
$$\begin{aligned} \mathrm{E}(X) &= \frac{r(1-p)}{p} \\ \mathrm{V}(X) &= \frac{r(1-p)}{p^2} \end{aligned}$$

Ainsi de notre côté nous avons reproduit ce modèle, et nous avons obtenu des résultats similaire en terme de p-valeurs. Le R^2 associé à ce graphique est de 0.9232.



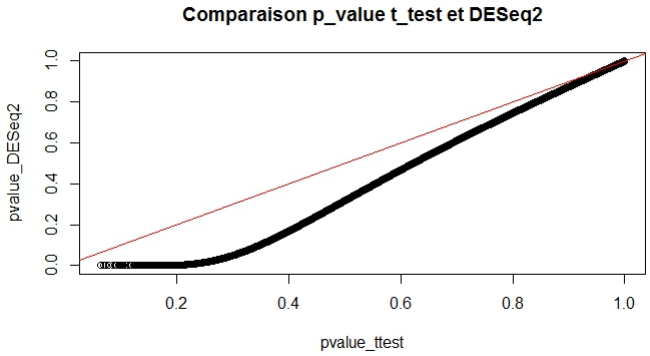
On en conclu que l'on retrouve bien les p-valeurs de DESeq2 en construisant un modèle linéaire généralisé avec *glm.nb* dans R. De plus on peut se demander si les coefficients obtenu sont similaires, ils ne devraient pas beaucoup différer entre les deux modèles. Nos résultats ne données pas quelque choses de bien car il y avait des outliers, après avoir enlevé 35 points aberrants, on obtient un R^2 entre les coefficients obtenus par DESeq2 et par le modèle glm.nb de R, de l'ordre de 0.9765, ce qui valide la similitude des méthodes que nous employons par rapports à celle de DESeq2.

Bullard et al [$reference$], on trouvé une relation entre les p-valeurs donné par le Fisher Exact Test et les p-valeurs donnée par le ratio du maximum de vraisemblance sous modèle de poisson que nous avons bien retrouvé après normalisation. En effet, on obtient un R^2 de 0.9941 entre les deux vecteurs de p-valeurs.



On obtient un résultat intéressant lorsque l'on compare les p-valeurs obtenues avec welch test et les p-valeurs obtenus avec DEseq2, on remarque que les p-valeurs obtenus avec le Welch Test sont globalement plus élevé, il y en a très peu en dessous de 0.05. On peut en conclure que le t-test est beaucoup plus conservatif que le test du maximum de vraisemblance. En effet on obtient le

graphique suivant sur lequel on observe clairement des p-valeurs plus faible pour la méthode employée par DESeq2.



Ces tests, nous ont permis de valider nos tests statistiques afin d'ensuite pouvoir tirer des conclusions sur nos données.

7.1 Une comparaison plus quantitative

Pour quantifier la différence entre les différents tests, on peut choisir de comparer les coefficients de corrélation entre les vecteurs de p-valeurs obtenu avec les 6 tests suivant : celui de Edger et DESeq2, le t-test, le test Exact de Fisher, le test du maximum de vraisemblance avec une loi de poisson et avec une loi négative binomiale.

Ainsi, on obtient la table suivante :

	Edger	DESeq2	FisherET	t-test	LRTP	LRTNB
Edger		0,917	0,065	0,603	0,712	0,837
DESeq2			0,043	0,657	0,772	0,923
FisherET				0,024	0,056	0,038
t-test					0,399	0,753
LRTP						0,695
LRTNB						

Ce tableau permet de comparer les p-values des différents test statistiques avec les mêmes données initiales de la fraction polysomiale. CElui-ci permet de confirmer à nouveau un certain nombre de résultats, notamment que les deux packages Edeger et DEseq2 utilisent des méthodes proche afin de caqlculer les p-values, ou du moins sur ce jeu de donnée, on obtient des p-valeurs très similaires. De plus, on remarque qu'avec une normalisation utilisant les facteurs

de normalisation de DEseq2, nous sommes très loin du résultat obtenu au-dessous concernant la similitude des p-valeurs obtenus par le test exact de Fisher et le test u maximum de vraisemblance avec la Loi de poisson.

7.2 Premières informations

Dans cette partie, nous exposons nos résultats concernant le nombre de gènes différentiellemet exprimés selon les tests et nous cherchons à identifier des gènes qui serait différentiellement exprimées à un niveau de risque 0.05 mais aussi 0.01 dans un contexte de différents test statistiques, j'ai choisis de croiser les p-valeurs obtenues avec le test du ratio du maximum de vraisemblance avec loi Négative Binomiale et les résultats de DESeq2.

Les premiers résultats que l'on peut donner sont résumés dans le tableau suivant avec le nombre de gènes différentiellemnt exprimés. Le nombre total de gène est de 18079. 0.05 et 0.01 désigne le niveau de risque que l'on se donne pour rejeter H0:

	0.05	0.01
Fisher Exact Test	9268	7183
T-test	473	53
Poisson LRT	6900	5005
NB LRT	1861	531
DESeq2	2286	889

7.3 Analyse de cross validation

J'ai choisi trois test et les p-valeurs de DEseq2, j'ai regarder leur intersection, afin de'identifier les gènes qui étaient différentiellement exprimés selon les 4 statistiques de test qui sont le test exact de Fisher, le LRT poisson et LRT NB. On en trouve 1169 qui le sont pour les trois tests.

On a ainsi pensé à faire un scoring pour chaque gène, en effet on s'est demandé si les 400 gènes qui avait un p-valeur en dessous de 0.05 vec le t-test, se retrouve tous dans les gènes qui ont une p-valeur en dessous de 0.05 avec le test du MLRP (Maximum Likelihood Ratio test avec une loi de Poisson). Ce n'est pas le cas, en effet on ne retrouve que 234 gènes qui ont une p-valeurs en dessous de 0.05 avec les 6 méthodes de tests.
Avec i fixé, soit $s_i = \sum_{j=1}^{6} p_{ij}$ Avec s_i la valeur du score pour le gène i, et p_{ij} prend ses valeurs entre 0 et 1, 0 si la p-valeurs pour le gène i et le test j est au dessus de 0.05, 1 sinon.
On peut présenter quelques résultats:
348 un score de 4, 872 5, 234 avec un score de 6 Il y a 1215 gène qui ont une p-valeurs en dessous de 0.05 pour les quatre statistiques fiables et cohérentes avec nos données.

8 Comment prendre en compte l'information présente dans les autres parties lib et 80s

8.1 Idée naïve : utiliser un test de Fisher-Snecdor

L'objectif de ce texte est de réfléchir à comment faire pour prendre en compte l'information contenue dans lib, 80s et pol pour statuer si les gènes sont différentiellement exprimés ou non après stimuli.

- Ma première idée serait de construire un modèle prenant en compte tous les coefficients qui aurait pour valeur le log de FC est de vérifier que tous ne sont pas égal à 0, et si au moins 1 est différents de 0 alors on peut statuer que le gène est différentiellement exprimé et calculer une p-valeurs à l'aide du test de Fisher-Snédécor qui permet de tester :

$$\log(\ u_{ij}) \ = \beta_i^{lib} + \beta_i^{pol} + \beta_i^{80s} \tag{31}$$

Rappel sur le Test de Fisher-Snédécor, celui-ci permet de tester l'hypothèse H0 : $Q\theta = 0$ avec Q un vecteur de taille $1 \times m$ et θ le vecteur des coefficient à tester dans notre cas les trois β donc de taille $m \times 1$ avec m le nombre de paramètres à tester.

La statistique de test est une généralisation du test de Student :

$$F = \frac{SSR_0 - SSR_1}{SSR_1} \times \frac{(n-k)}{q} \sim F_{q,n-k} \tag{32}$$

Pour obtenir SSR_0 on considère que H0 est vraie, c'est à dire que les $\log(FC)$ de lib, pol et 80s sont tous égaux à 0. Et on calcule ,$\log(\ u_{ij})$ - $\log(\widehat{u}_{ij})$ avec $\log(\widehat{u}_{ij})$ c'est l'estimateur du log de FC global entre pol, lib et 80s. On fait la même chose pour calculer SSR_1.

Dans l'équation au dessus n - k est le nombre de degrés de liberté, c'est à dire n le nombre d'observation, moins le nombre de paramètres que l'on souhaite estimer. q est le nombre de paramètre que l'on égalise à 0. Dans notre cas c'est 3 - 2 = 1.

Ce test n'est pas suffisant car on aimerait aussi pouvoir répondre à d'autres questions, telle que dans quelle fraction le gène est le plus différentiellement exprimé.

8.2 Le Modèle linéaire

En s'inspirant de l'article de Simon Anders sur des problématiques liées aux exon, on a décidé de construire un modèle linéaire qui nous permettrait de tester un grand nombre d'hypothèse en même temps. Pour tout i fixé, avec i pour le i^{eme} gène, le modèle est de la forme :

$$\log(\mu_{ijl}) = \delta_i + \beta_{il} + \theta_{i\rho(j)} + \lambda_{i\rho(j)l} \tag{33}$$

Avec μ_{ijl} la moyenne des counts pour un gène i, et un répliqua j avec j allant de 1 à 16. l va de 1 à 3 pour les trois fractions lib, pol, 80s. $\rho(j)$ est égal à 0 ou 1, si stimulé ou non. On applique un facteur de normalisation, fixe pour tout i et variant en fonction des répliquas, s_j, ainsi on a le modèle suivant:

$$\log(\mu_{ijl}) = \log(s_j) + \delta_i + \beta_{il} + \theta_{i\rho(j)} + \lambda_{i\rho(j)l} \tag{34}$$

- δ_i la quantité moyenne de counts pour un gène i
- β_{il} la quantité moyenne de counts pour chaque fractions lib,pol,80s
- $\theta_{i\rho(j)}$ le différentielle d'expression d'un gène au niveau des counts entre deux conditions et pour toutes les fractions.
- $\lambda_{i\rho(j)l}$ le différentielle d'expression pour chaque fractions

Dans notre cas particulier, le modèle est de la forme :

$$\begin{aligned}\log(\mu_{ijl}) = \log(s_j) + \delta_i + \beta_{ilib} + \beta_{ipol} + \beta_{i80s} + \theta_{i0} + \theta_{i1} + \\ \lambda_{i0lib} + \lambda_{i1lib} + \lambda_{i0pol} + \lambda_{i1pol} + \lambda_{i080s} + \lambda_{i180s}\end{aligned}$$

Cependant si on associe la matrice associée à ce modèle, on se rend compte qu'il y a beaucoup de colinéarité, c'est à dire que l'on a deux fois la même information, ce qui n'est pas intéressant pour un modèle linéaire. En effet dans notre cas, la matrice associée au modèle est la suivante:

$$\log(\mu_{ijl}) = \big(\delta_i, \lambda_{i0lib}, \lambda_{i1lib}, \lambda_{i0pol}, \lambda_{i1pol}, \lambda_{i080s}, \lambda_{i180s}, \beta_{ilib}, \beta_{ipol}, \beta_{i80s}, \theta_{i0}, \theta_{i1}\big) \times$$

		δ_i	λ_{i0lib}	λ_{i1lib}	λ_{i0pol}	λ_{i1pol}	λ_{i080s}	λ_{i180s}	β_{ilib}	β_{ipol}	β_{i80s}	θ_{i0}	θ_{i1}
lib	NS	1	1	0	0	0	0	0	1	0	0	1	0
	NS	1	1	0	0	0	0	0	1	0	0	1	0
	NS	1	1	0	0	0	0	0	1	0	0	1	0
	FSH	1	0	1	0	0	0	0	1	0	0	0	1
	FSH	1	0	1	0	0	0	0	1	0	0	0	1
pol	NS	1	0	0	1	0	0	0	0	1	0	1	0
	NS	1	0	0	1	0	0	0	0	1	0	1	0
	FSH	1	0	0	0	1	0	0	0	1	0	0	1
	FSH	1	0	0	0	1	0	0	0	1	0	0	1
	FSH	1	0	0	0	1	0	0	0	1	0	0	1
$80s$	NS	1	0	0	0	0	1	0	0	0	1	1	0
	NS	1	0	0	0	0	1	0	0	0	1	1	0
	NS	1	0	0	0	0	1	0	0	0	1	1	0
	FSH	1	0	0	0	0	0	1	0	0	1	0	1
	FSH	1	0	0	0	0	0	1	0	0	1	0	1
	FSH	1	0	0	0	0	0	1	0	0	1	0	1

On voit bien dans cette matrice que l'on peut choisir d'annuler toutes les situations non stimulés des fractions, mais aussi d'annuler une des fractions stimulés sinon on obtient le vecteurs associé à θ_{i1}. De plus on doit annuler, un des λ, on choisit λ_{i0lib}. Il existe des méthodes pour ne pas avoir à réduire à ce point la matrice des vecteurs explicatifs, en les modifiant quelque peu à l'aide de la déviance et de la moyenne. Cependant l'un des grands avantages du modèle linéaire est de permettre de faire des tests sans avoir besoin d'estimer la moyenne ou la variance, car pour utiliser le ratio du maximum de vraisemblance, il est simplement nécessaire d'obtenir le maximum de vraisemblance pour une loi Binomiale Négative. A la fin on obtient la matrice suivante simplifiée (répliquas non représentés) :

		Base Mean			*Fold Change*		
		δ_i	β_{ipol}	β_{i80s}	λ_{i1pol}	λ_{i180s}	θ_{i1}
Lib	NS	1	0	0	0	0	0
	FSH	1	0	0	0	0	1
Pol	NS	1	1	0	0	0	0
	FSH	1	1	0	1	0	1
$80s$	NS	1	0	1	0	0	0
	FSH	1	0	1	0	1	1

8.3 Les hypothèses à tester

Le modèle qui nous servira à tester nos hypothèses est le modèle épuré suivant:

$$\log(\mu_{ijl}) = \log(s_j) + \delta_i + \beta_{ipol} + \beta_{i80s} + \theta_{i1} + \lambda_{i1pol} + \lambda_{i180s} \qquad (35)$$

Nous utiliserons les résultats obtenus grâce à un test du ratio de maximum de vraisemblance avec une loi négative Binomiale. Choix de la loi négative

binomiale à expliciter plus haut: La litterature s'accorde sur le fait que l'on a une négative binomiale, montrer un histogramme des distribuition, permet plus de souplesse, la moyenne n'est pas forcement égale à la variance. Permet plus de variabilité.

On peut vouloir tester plusieurs hypothèses, la première est identique à ce que l'on a fait dans la partie sur les tests serait de tester,en supposant θ_{i1} nul pour les deux modèles sous HO et sous H1:

$$\begin{gathered}\text{H0 : } \lambda_{i0pol} = \lambda_{i1pol} \\ \text{H1 : ils sont différents}\end{gathered}$$

On a décidé de vérifier si l'on obtenait bien des résultats similaire pour la fraction polysomiale, aussi bien en terme de coefficients qu'en termes de p-valeurs à ce que l'on obtient avec DESeq2. Les résultats sont plutôt concluant, en utilisant toujours le facteur de normalisation obtenu avec DESeq2, on obtient des coefficients similaires sous l'hypothèse H1. Nous comparons les coefficients associés à λ_{i1pol} dans le modèle1 avec le Log2FC pour la fraction pol, cela pour chaque gène i. Pour l'étude des coefficents, j'ai utilisé le code suivant :

testrescoef ¡- function(Y,delta, Xpol, X80s,XpolFSH, X80sFSH, thetaFSH) model0 = glm.nb(Y offset(log(vectnorm))+ delta + Xpol + X80s + X80sFSH -1,link = "log") model1 = glm.nb(Y offset(log(vectnorm))+ delta + Xpol + X80s +XpolFSH + X80sFSH -1,link = "log") T ¡- lrtest(model0,model1) return(model1)

On peut aussi tester si le gène est globalement (pour toutes les fractions) differentiellement exprimé et calculer les p-valeurs associés pour chaque gène :

$$\begin{gathered}\text{H0 : } \theta_{i0} = \theta_{i1} \\ \text{H1 : ils sont différents}\end{gathered}$$

On peut vouloir savoir si le gène i est plus differentiellement exprimé dans la fraction pol que dans les autres fractions, on peut pour se faire utiliser un test de Wald avec la méthode suivante : Est-ce le gène j est differentiellement exprimé surtout dans la partie pol. Pour ce faire, on peut tester l'hypothèse nulle suivante :

$$H0 : 2 * |\lambda_{i1pol}| - (|\lambda_{i1lib}| + |\lambda_{i180s}|) < 0 \tag{36}$$

Ce qui donne si on pense à un test de Wald, la fonction R = (2,-1,-1), avec $\theta = (\lambda_{i1pol}, |\lambda_{i1lib}|, \lambda_{i180s})$ On peut de la même façon faire plusieurs variante en fonction des hypothèses H0 que l'on fixe sans forcément changer le modèle.
On peut tester cela en ne faisant qu'une petite modification au niveau des coefficients du modèle
Pour obtenir λ_{i1lib}, on peut utiliser le paramètre de $\theta_{i1} - \lambda_{i180s} - \lambda_{i1pol}$.