Master Mathématiques et Applications
Spécialité Statistiques et Données du Vivant

Rapport de stage M1

Analyse différentielle des données omics

Présenté par *OUCHENE HAMZA*

Tuteurs de stage :
M.*Romain Yvinec*
Tutrice pédagogique :
Mme.*Hermine Biermé*

02 Mai 2017 — 31 Juillet 2017

Table des matières

Table des figures

1 Description de l'entreprise

1.1 L'INRA

Les missions de l'INRA

L'Institut National de la Recherche Agronomique (INRA) est le premier institut de recherche agronomique d'Europe. Elle mène de nombreuses recherches dans différents domaines. On peut résumer le travail de l'INRA en 4 mots : Explorer, Comprendre, Expérimenter et Anticiper.

L'une des actions de l'INRA est de produire et diffuser des connaissances afin de contribuer à la compréhension du monde et de ses évolutions, au développement et à la gestion des biens publics, au bien-être des citoyens ainsi qu'à l'innovation socio-économique. L'INRA utilise aussi ces connaissances pour dé- boucher sur des innovations qui bénéficieront aux autres entreprises (agricoles, industrielles, services).

Les activités de l'INRA ont un impact dans de nombreux domaines. Les chercheurs et techniciens de l'institut cherchent à nourrir la France et le monde, travaillent sur le réchauffement climatique, à réduire la dépendance aux pesticides et aux engrais, à la sauvegarde des espèces et également à conserver la biodiversité génétique des plantes et des animaux. Ils coopèrent aussi avec l'enseignement supérieur afin d'accueillir et former les futurs chercheurs.

Les différents impacts peuvent être classés en cinq catégories : économique, environnemental, politique, sanitaire, et territorial/social. Ils sont mesurés à long terme car il faut souvent plusieurs années (temps moyen : 20 ans) pour qu'une innovation soit réalisée, mise en application et ait des conséquences sérieuses sur la société.

L'impact économique consiste évidemment à mesurer les contributions des différents acteurs d'une innovation et de les comparer aux retombés économiques pour les bénéficiaires. L'impact environnemental consiste à mesurer les effets, en bien ou en mal, sur l'environnement des innovations (énergies, déchets, . . .).

Les pôles de recherche

Au centre INRA du Val de Loire où j'ai fait mon stage, il y a plusieurs pôles de recherche qui regroupent les activités du centre :

- Biologie animale intégrative et gestion durable des productions animales
- Biologie intégrative des arbres et des organismes associées
- Dynamique des sols et gestion de l'environnement
- Santé animale et santé publique

L'objectif du pôle Biologie animale intégrative et gestion durable des productions animales, en plus de la production de connaissances, est d'améliorer la durabilité des systèmes d'élevage dans les domaines économique, social et environnemental. Dans le domaine de l'économie, le but est d'assurer la compétitivité des filières, dans le domaine social, le but est de permettre la coexistence des petites et grandes exploitations et, dans le domaine environnemental, le but est de limiter l'usage des intrants (engrais, produits chimiques, hormones de synthèse, . . .) et gérer les paysages.

C'est pour garantir un avenir pour l'importante ressource naturelle que sont les forêts que le pôle Biologie intégrative des arbres et des organismes associées existe. On cherche à y approfondir les connaissances sur la diversité génétique des espèces forestières majeures afin d'optimiser la conservation de ces ressources et d'identifier de nouvelles espèces. On étudie aussi les insectes vivant dans ces forêts.

Le pôle Dynamique des sols et gestion de l'environnement est particulièrement important pour le développement durable. Les sols intervenant à la fois dans la production agricole et dans la protection de l'environnement. Deux approches complémentaires sont appliquées. D'une part, on inventorie et surveille la qualité des sols et, d'autre part, on étudie les déterministes du fonctionnement des sols. Tout cela ayant pour but de protéger les sols contre le tassement et l'érosion, d'améliorer l'alimentation en eau des cultures et de réduire les émissions de gaz à effet de serre par les sols.

Le pôle Santé animale et santé publique est dédié à l'étude et à la recherche sur les agents pathogènes animaux, plus spécifiquement de ceux qui sont communs à l'homme et aux animaux. Les recherches menées doivent contribuer à la maîtrise des infections des animaux d'élevage (maladies infectieuses et parasitaires) constituant un risque pour la santé publique (consommation d'aliments d'origine animale).

Pour finir, l'unité dont fait partie l'équipe au sein de laquelle j'ai travaillé (équipe BIOS) est l'unité de Physiologie de la Reproduction et des Comportements (PRC). C'est une unité de recherche qui mène des recherches fondamentales et appliquées sur la fonction de reproduction, sur les comportements émotionnels, sociaux et sexuels et sur les mécanismes d'adaptation des individus et des populations à des environnements changeants.

1.2 L'équipe BIOS

L'équipe BIOS (Biologie et Bio-informatique des Systèmes de Signalisation) regroupe des chercheurs ayant des compétences en mathématiques et en biologie pour étudier des données fournies par les expérimentateurs. On y développe et utilise des méthodes statistiques et des logiciels appropriés à cela.

Les objectifs de l'équipe sont de comprendre et modéliser les réseaux de signalisation induits par des hormones, comprendre l'action de ces hormones et tirer profit de ces connaissances pour développer des substituts à l'utilisation de ces hormones.

2 Problématique du stage

2.1 Notions biologique de base

2.1.1 L'ADN

L'acide désoxyribonucléique (ADN) est une molécule présente dans toutes les cellules vivantes et qui renferme l'ensemble des informations nécessaires au développement et au fonctionnement d'un organisme. Il porte l'information génétique (génotype) et constitue le génome des êtres vivants.
La structure standard de l'ADN est une double-hélice droite, composée de deux brins complémentaires se faisant face. Chaque brin d'ADN est constitué d'un enchaînement de nucléotides. On trouve quatre nucléotides différents dans l'ADN, notés A, G, C et T, du nom des bases azotées qui les composent. Les nucléotides trouvés dans un brin possèdent des nucléotides complémentaires dans l'autre brin avec lesquels ils peuvent interagir (A complémentaire avec T, G avec C). Le génotype est inscrit dans l'ordre dans lequel s'enchaînent les quatre nucléotides. La figure 3.1 donne une représentation de l'ADN.

2.1.2 L'ARN

L'ARN est une copie d'une région de l'un des brins de l'ADN. Les ARN produits peuvent avoir trois grands types de fonctions : ils peuvent être supports de l'information génétique d'un ou plusieurs gènes codant des protéines (on parle alors d'ARN messagers), ils peuvent adopter une structure secondaire et tertiaire stable et accomplir des fonctions catalytiques et ils peuvent enfin servir de guide ou de matrice pour des fonctions catalytiques accomplies par des facteurs protéiques.
Un ARN non codant (ou ARNnm pour ARN non-messager) est un ARN issu

de la transcription de l'ADN qui ne sera pas traduit en protéine. Il en existe différents types dont les ARNs régulateurs de l'expression des gènes.

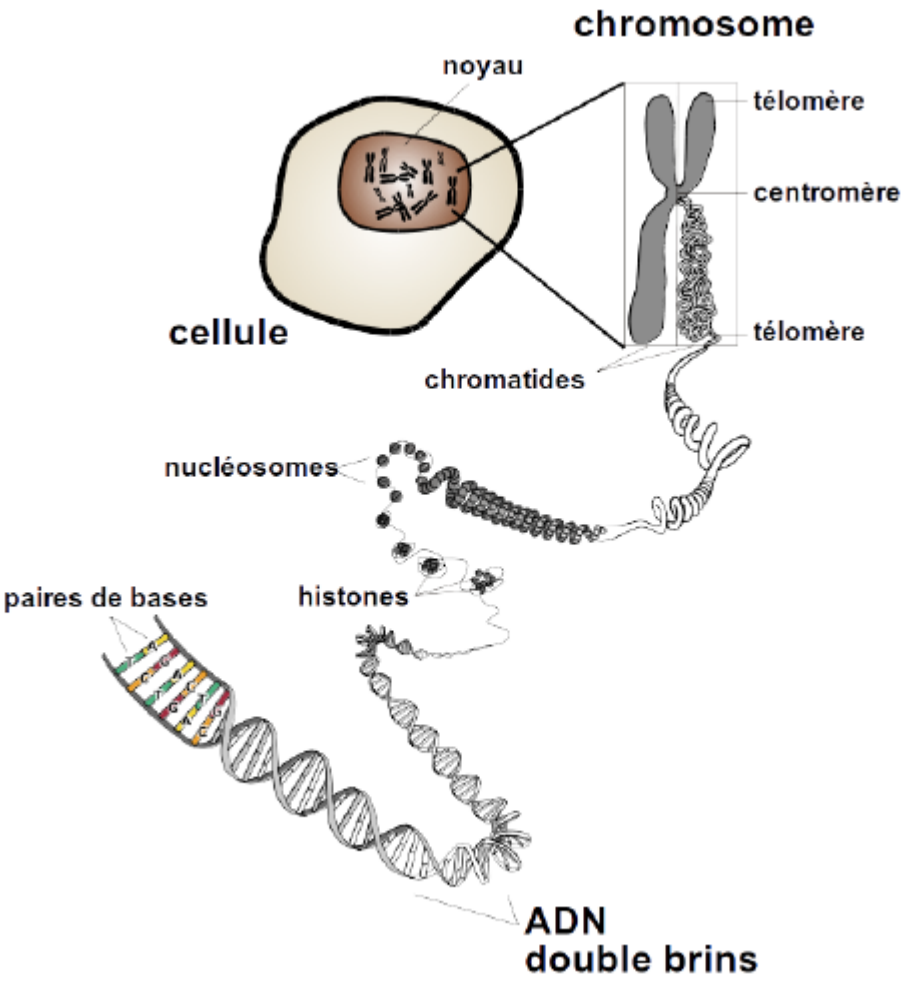

FIGURE 1 – De la cellule à l'ADN

2.1.3 Le Traductome

2.1.4 Expression de l'information par l'ADN

L'information génétique portée par l'ADN constitue le génotype d'un organisme qui s'exprime pour donner naissance à un phénotype, c'est-à-dire l'ensemble des caractères de cet organisme. Cette expression du génome se fait en plusieurs étapes, comme on peut le voir illustré dans la figure suivante :
- La transcription, qui consiste à copier des régions dites codantes de l'ADN en molécules d'ARN.
- La traduction, qui est un transfert d'information depuis l'ARN vers les protéines.
- L'activité des protéines, qui détermine l'activité des cellules, qui vont ensuite déterminer le fonctionnement des organes et de l'organisme.

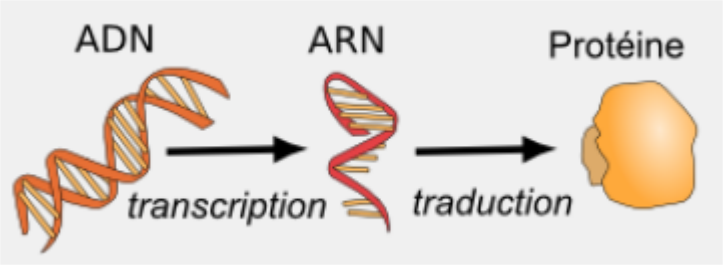

FIGURE 2 – Transcription, traduction

Le transcriptome est donc l'ensemble de toutes les molécules d'ARN produites dans une population de cellules. Il peut varier avec des conditions environnementales externes et reflète les gènes qui ont été exprimés de façon active à un temps donné. **Ainsi l'expression d'un gène est mesurée par la mesure quantitative de ses transcrits.**

2.1.5 Mesure du transcriptome technologie RNASeq

Pour mesurer le transcriptome, on dispose d'une technologie récente : le RNASeq. Il s'agit d'une technique de séquençage à haut débit qui mesure l'abondance de séquences d'ARN dans différentes cellules pour des milliers de gènes simultanément. Rappelons que le séquençage de l'ARN consiste à déterminer l'ordre d'enchaînement des nucléotides pour un fragment d'ARN donné. Le nombre de séquences lues (appelées aussi "reads") et alignées sur

une région d'intérêt (le gène par exemple) est considéré comme proportionnel au niveau d'expression de cette région d'intérêt. En effet, plus un gène s'exprime, plus grand est le nombre de ses transcrits.

2.1.6 Mesure du proteome technologie spectrométrie de masse

La spectrométrie de masse s'est imposée pour l'identification et la caractérisation des protéines. Cette technique permet de mesurer la masse d'une molécule suivant sont rapport masse sur charge (m/z). La fragmentation des molécules analysées, permet de recueillir des informations sur leur composition.
Une analyse protéomique se déroule en plusieurs étapes. La première étape consiste à extraire les protéines du milieu d'étude (cellules, liquides biologiques ou tissus...). Les protéines sont ensuite séparées le plus souvent par électrophorèse mono ou bidimensionnelle, une technique qui sépare les protéines suivant leur charge et/ou leur poids moléculaire sur un gel d'acrylamide. La coloration du gel permet de visualiser les protéines, et de découper les spots ou bandes d'intérêt. Les protéines présentes dans le morceau de gel subissent ensuite une hydrolyse enzymatique afin d'en extraire les peptides. Le mélange peptidique est alors analysé par spectrométrie de masse. Enfin, les données obtenues sont comparées avec des banques de données protéiques afin d'identifier les protéines présentes dans l'échantillon de départ

2.2 Problématique du stage

2.2.1 Les données

Les données fournies sont des données issues du séquençage haut débit. Il s'agit donc, pour chaque échantillon, une liste de séquençage des ARNm ou protéines produits (a l'échelle d'une cellule de ***Sértolli***) dans différentes conditions expérimentales a fin de identifier les gènes pour lesquels l'activité (quantité de produits géniques) à évolué, soit augmenté soit diminué.

2.2.2 La normalisation

Avant de pouvoir analyser notre tableau de comptage, il faut normaliser les données. Autrement dit, avant de pouvoir comparer ces données, il faut les rendre comparables entre elles. En effet, il existe plusieurs biais, notamment le biais de profondeur de séquençage(nombre de reads alignés dans l'échantillon) qui nous empêchent de passer directement à l'analyse statistique pour la recherche de gènes différentiellement exprimés.
Imaginons qu'il y ait deux fois plus de reads dans un échantillon (A) que dans un autre (B). Alors, pour un gène qui s'exprime de la même manière

dans les deux échantillons, le nombre de reads dans (A) sera approximativement égal à deux fois le nombre de reads dans (B) et les comptages ne seront pas directement comparables. Une des solutions que l'ont pourrait imaginer serait de diviser chaque comptage par le nombre total de reads dans l'échantillon, mais cette méthode est insuffisante car le nombre total de reads d'un échantillon est fortement influencé par quelques gènes dont le comptage est très élevé (la distribution du nombre de reads par gène est en général très asymétrique). La méthode qui sera utilisée sera celle développée par Anders et Hubers dans [1 BIBLIOGRAPHIE après] et nommée « **Relative Log Expression** ». Il s'agit d'utiliser, pour chaque échantillon j, un facteur d'échelle S_j qui sera utilisé en coefficient multiplicateur pour corriger le nombre de comptage dans l'échantillon. Ce facteur d'échelle est obtenu comme suit : - on calcule la moyenne géométrique Mi des comptages $K_{i,j}$ de chaque gène i au travers de tous les échantillons (notons N le nombre d'échantillons) :

$$M_i = (\prod_{j=1}^{N} K_{i,j})^{\frac{1}{N}}$$

On obtient ainsi un pseudo échantillon de référence $j^{'}$ composé de ces comptages moyens ;
- on calcule pour chaque comptage le ratio $\frac{comptage(i,j)}{comptage(i,j^{'})}$: c'est un ratio entre l'échantillon j et le nouvel échantillon de référence $j^{'}$;
- on calcule enfin, pour chaque échantillon j, la médiane S_j du ratio d'échantillons. Cette médiane sera notre facteur d'échelle par lequel tout les comptages seront multipliés pour rendre les différents échantillons comparables et corriger le biais de profondeur de séquençage.

Remarque : On utilise la moyenne géométrique qui est plus robuste que la moyenne arithmétique car moins sensible aux valeurs extrêmes. Elle donne donc une meilleure estimation de la tendance centrale des données.

2.2.3 L'analyse dite "différentielle"

Après avoir normalisé les données, on arrive à l'étape finale : la recherche de gènes différentiellement exprimés. Autrement dit on recherche les gènes qui ont un niveau d'expression significativement différent d'une condition à une autre. On teste donc, pour chaque gène, si son niveau moyen d'expression diffère d'une condition à une autre.

Le genre de résultats qu'on peut tirer avant et après l'analyse statistique

Avant l'analyse

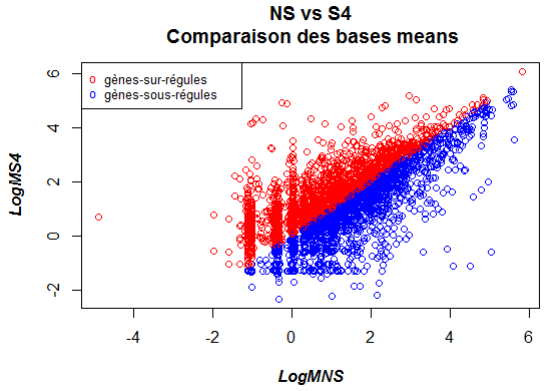

FIGURE 3 – Comparaison des moyennes de deux condition différents

Après l'analyse on identifier les gènes significativement différentiellement exprimés

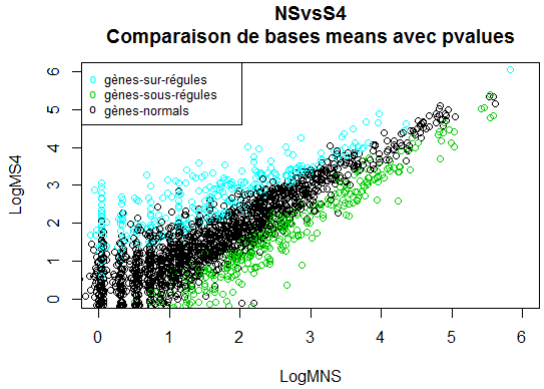

FIGURE 4 – Comparaison des moyennes de deux condition différents après l'analyse

2.2.4 La corrélation traductome versus proteome

Pour visualiser la corrélation traductome versus proteome il faut faire le match entre les deux jeux de données proteome et traductome et voir après quelles sont les gènes différentiellement exprimés a la fois au traductome et au proteome.

3 Travail réalisé

3.1 Première problématique :les gènes différentiellement exprimés

3.1.1 But et Outils

Il faut rappeler que le but de ce stage était de produire un script reproductible pour le traitement de données RNASeq et proteomique pour la recherche de gènes différentiellement exprimés et voir après est ce qu'il y a une corrélation entre ces deux dernières.
Le logiciel que j'ai utilisé est le logiciel statistique libre R avec son interface utilisateur graphique « Rstudio ».
J'ai en particulier utilisé le package DESeq2 , que l'on trouve, non pas sur le CRAN, mais sur un dépôt de packages R spécifiques à la bioinformatique, Bioconductor .

3.1.2 La qualité des données

Les données utilisées en premier lieu sont des données brutes de séquençage de la cellule de Sértolli. Il s'agit, pour chacun des échantillons récupérés, les reads obtenus.
Donc pour le jeux de données proteome il s'agit d'un tableau excel de 4874 lignes « représenter par des gèneID »et 9 colonnes, pour le jeux de données traductome il s'agit d'un tableau excel de 31000 lignes « représenter par des ensembles de gènes »et 16 colonnes.
Voici un extrait des deux jeux de données :

A	B	C	D	E	F	G	H	I	J
GeneID	R1NS	R2NS	R3NS	R1S90	R2S90	R3S90	R1S4	R2S4	R3S4
	Non Stimulé			Stimulé après 90 min			Stimulé après 4 h		
gène1	316	285	385	280	260	315	379	532	501
gène2	235	252	251	182	340	337	179	278	248
gène3	292	283	273	104	226	204	110	283	241
gène4	263	276	251	121	228	227	147	297	260
gène5	235	253	273	187	186	217	204	286	257
gène6	247	249	282	183	162	179	96	158	169
gène7	123	128	119	103	150	80	90	233	251
gène8	146	147	117	133	174	159	118	208	179
gène9	331	256	349	221	215	245	200	243	251
gène10	125	92	151	119	96	114	126	119	130
gène11	218	218	225	181	189	168	142	73	182
gène12	115	129	124	103	120	117	129	130	180
gène13	197	202	211	99	75	91	105	112	102
gène14	156	158	147	58	130	115	75	181	170
gène15	85	104	110	61	52	56	69	134	158
gène16	53	54	51	143	169	183	115	188	163
gène17	102	118	159	69	108	114	66	151	170
gène18	143	154	153	96	116	107	146	157	208
gène19	126	126	139	108	44	106	113	118	119
gène20	99	102	86	141	165	107	98	70	141
gène21	91	93	93	119	150	102	97	82	145
gène22	134	114	147	69	139	137	94	187	143

FIGURE 5 – Le jeux de données proteomique

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	ENS	NSLib1	NSLib2	NSLib4	FSHLib2	FSHLib4	NS80s1	NS80s2	NS80s4	FSH80s1	FSH80s2	FSH80s4	NSPol2	NSPol4	FSHPol1	FSHPol2	FSHPol4
2		Lib non stimulé			Lib stimulé		80s non stimulé			80s stimulé			Pol non stimulé		Pol stimulé		
3	ENSRNOG00000000001	16	21	23	19	21	7	6	5	8	6	3	2	3	6	5	2
4	ENSRNOG00000000007	105	152	147	99	123	125	104	78	95	118	83	54	63	134	62	59
5	ENSRNOG00000000008	111	203	156	150	150	155	107	100	94	140	94	83	36	106	80	53
6	ENSRNOG00000000009	61	112	72	85	70	57	54	46	50	80	53	28	10	48	48	24
7	ENSRNOG00000000010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	ENSRNOG00000000012	39	62	52	61	38	53	28	32	23	41	49	16	14	43	31	21
9	ENSRNOG00000000017	53	78	85	74	76	78	48	39	49	77	45	29	33	43	39	37
10	ENSRNOG00000000021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	ENSRNOG00000000024	140	220	186	196	158	188	136	107	109	161	154	65	46	141	65	69
12	ENSRNOG00000000033	404	587	516	569	469	333	192	159	203	312	202	147	62	203	160	112
13	ENSRNOG00000000034	125	196	157	134	122	83	59	56	67	110	39	26	28	58	47	40
14	ENSRNOG00000000035	112	152	152	162	127	133	86	109	79	147	71	74	29	97	61	55
15	ENSRNOG00000000036	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
16	ENSRNOG00000000040	75	99	74	76	81	46	48	31	44	68	28	20	15	43	31	30
17	ENSRNOG00000000041	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	ENSRNOG00000000042	869	1307	1115	1211	1047	885	664	589	651	986	630	417	198	688	471	366
19	ENSRNOG00000000043	88	136	115	155	137	62	44	43	54	77	29	51	16	83	34	36
20	ENSRNOG00000000044	137	199	175	208	166	156	94	99	103	130	126	105	146	181	155	188
21	ENSRNOG00000000047	136	233	177	179	159	119	108	92	102	202	143	77	40	125	50	78
22	ENSRNOG00000000048	639	991	829	854	759	608	453	545	365	594	485	411	463	572	499	656
23	ENSRNOG00000000053	297	404	358	353	320	285	180	191	164	253	150	110	76	180	137	98
24	ENSRNOG00000000054	74	126	104	114	98	44	39	20	27	40	36	23	5	35	31	23
25	ENSRNOG00000000055	54	107	83	99	88	70	41	42	44	52	44	30	31	53	42	37
26	ENSRNOG00000000060	320	473	419	395	360	247	139	126	167	277	143	125	66	206	95	120
27	ENSRNOG00000000062	356	481	380	425	387	219	180	160	214	371	244	170	163	638	570	598

FIGURE 6 – Le jeux de données traductome

Classification : Proteome

Classification Ascendante Hiérarchique (CAH). Il s'agit de regrouper les échantillons les plus proches afin de former des classes qui elles mêmes seront regroupées selon leur distance pour former de nouvelles classes, et ainsi de suite jusqu'à obtention d'un dendogramme comme présenté dans la figure suivante :

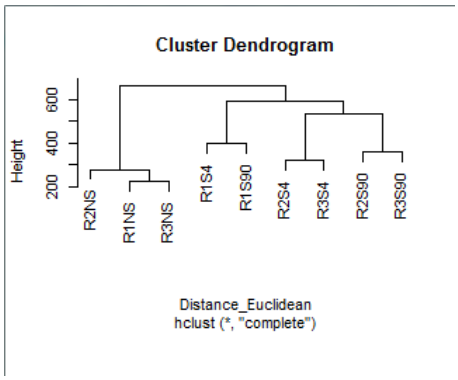

FIGURE 7 – Classification Hiérarchique Ascendante

On regroupe les classes selon une distance que l'on définit dans les paramètres de la fonction. Ici, il s'agit d'une distance euclidien. Ainsi, plus les classes sont « éloignées », moins elles sont corrélées. Le but est de vérifier que les échantillons que l'on suppose a priori être « proches »le sont effectivement.

Heatmap

Pour mieux visualiser cette classification, notamment lorsqu'il y a un plus grand nombre d'échantillons avec plusieurs facteurs de conditions, il est intéressant de générer une « heatmap »qui permet aussi de visualiser les ressemblances entre les échantillons.On obtient ainsi une « heatmap »dont on voit un exemple dans la figure en dessous. Plus la zone est foncée, plus la quantification est forte.

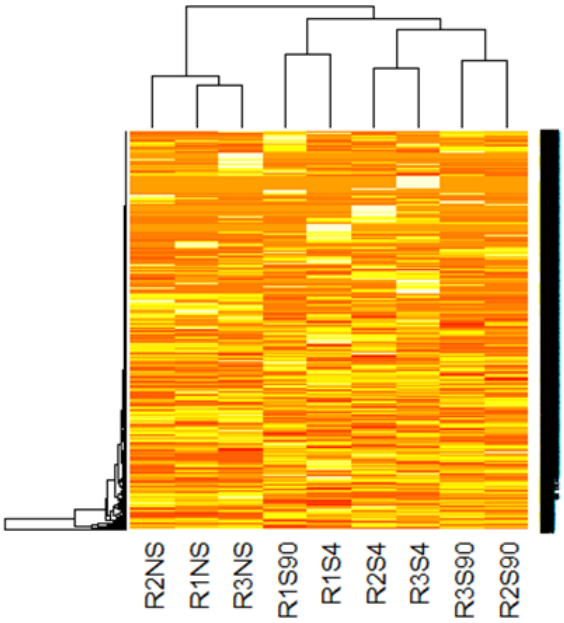

FIGURE 8 – Exemple de « heatmap »

3.1.3 Analyse statistique en utilisant DESeq2

— **Installation du package DESeq2** : normalisation et recherche de gènes différentiellement exprimés en se basant sur un modèle de distribution binomiale négative ;

— **Chargement de la table de comptage.**

— **Construction de l'objet DESeqDataSet (dds).** Un tel objet contient la table de comptage mais également un tableau décrivant le plan d'expérience (nommé coldata). Dans ce plan d'expérience, il y a au moins une variable de type facteur :
la condition. Il s'agit de la condition biologique qui peut prendre par exemple deux niveaux : « organisme traité », « organisme non traité ». Il y a également un facteur qui décrit le format de séquençage qui peut prendre deux niveaux : single-read quand on séquence l'ARN d'un seul coté, ou paired-end,quand on séquence des deux cotés.

— **Exploration des données.** Il s'agit d'explorer nos échantillons et d'observer les différences qui existent entre eux. On observe ainsi la proportion de comptages nuls par échantillons, la profondeur de séquençage (c'est à dire le nombre de reads) par échantillons, la distribution des comptages des gènes par échantillons, le tout pour pouvoir faire une première comparaison des échantillons entre eux. Pour mieux visualiser la distribution des comptages, on fait un histogramme des valeurs de comptages en utilisant leur log-fréquence (on aura ainsi en ordonnées les fréquences des « log(comptage + 1) » qui se trouvent en abscisses).

— **Filtrage.** Il s'agit de supprimer les gènes qui n'apportent aucun intérêt à l'analyse statistique : typiquement, les gènes aux comptages toujours nuls quelle que soit la condition. On peut également appliquer un filtrage arbitraire :Ce type de filtrage dépend du nombre de gènes et à l'appréciation du biologiste.

— **Normalisation et analyse différentielle.** La fonction DESeq() réalise la normalisation et l'analyse différentielle en une seule étape. Dans notre script, on a utilisé les valeurs par défaut des options test et fitType. Le traitement se décompose en deux étapes :
- la normalisation est effectuée par la méthode décrite dans la section 2.2.2 à la page 9. Les facteurs d'échelle résultant de la normalisation sont automatiquement inscrits dans la table coldata dans une colonne sizeFactor ; - l'analyse différentielle est basée sur une modélisation de

la distribution des gènes par une loi Binomiale Négative. L'estimation des paramètres de ce modèle se fait grâce à une estimation de la relation entre la variance et la moyenne des comptages. C'est à partir de ce modèle que les tests seront réalisés. Nous pouvons visualiser les résultats de la normalisation en représentant à nouveau la profondeur de séquençage et la distribution des comptages des données normalisées Les résultats de l'estimation des paramètres de dispersion, les détails sur les tests statistiques et p-values sont extraits grâce à des fonctions spécifiques au package (se reporter au script). Nous pouvons également visualiser l'estimation de la relation dispersion moyenne, comme dans la figure 9

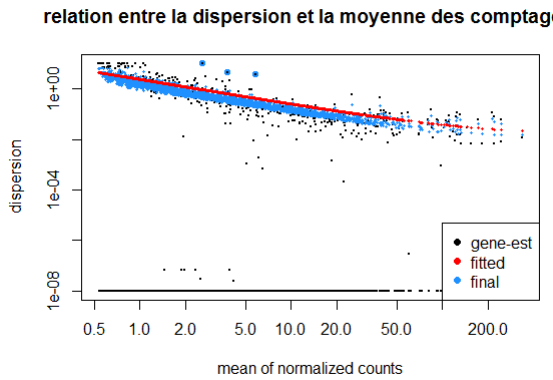

FIGURE 9 – Estimation de la relation entre dispersion et moyenne

— **Résultats de l'analyse**.
Les résultats de l'analyse sont accessibles via la

fonction results(dds) : pour chaque gène nous avons, entre autres, la statistique de test, la p-value, la p-value ajustée (par la méthode de ***Benjamini-Hochberg***).
Nous pouvons donc extraire une liste des gènes déclarés différentiellement exprimés à 1%, 2% ou 5% en les séléctionnant par rapport à leur p-value.

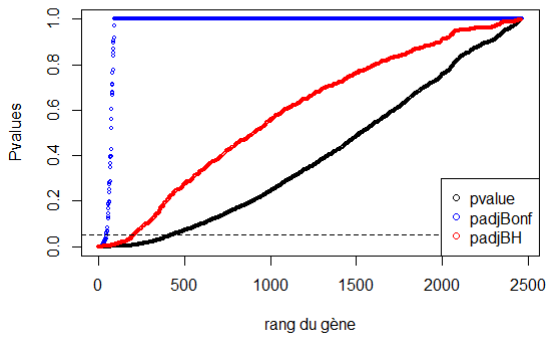

FIGURE 10 – Les pvalues cumulées

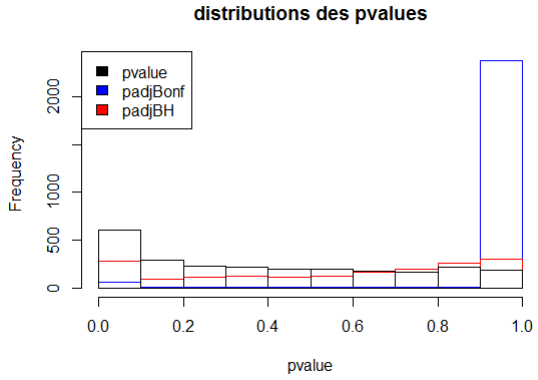

FIGURE 11 – La distribution des pvalues

3.1.4 Analyse statistique en utilisant GLM

le **modèle linéaire généralisé** est une généralisation souple de la régression linéaire. Le GLM généralise la régression linéaire en permettant au modèle linéaire d'être relié à la variable réponse via une **fonction lien** et en autorisant l'amplitude de la variance de chaque mesure d'être une fonction de sa valeur prévue.

Le GLM utilisé pour le traductome

$$\log(\mu_{ijl}) = \delta_i + \beta_{il} + \theta_{i\rho(j)} + \lambda_{i\rho(j)l} \tag{1}$$

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 :

$$\log(\mu_{ijl}) = \log(s_j) + \delta_i + \beta_{il} + \theta_{i\rho(j)} + \lambda_{i\rho(j)l} \tag{2}$$

— 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 :

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

	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_{NS1}	1	1	0	0	0	0	0	1	0	0	1	0
lib_{NS2}	1	1	0	0	0	0	0	1	0	0	1	0
lib_{NS3}	1	1	0	0	0	0	0	1	0	0	1	0
lib_{FSH1}	1	0	1	0	0	0	0	1	0	0	0	1
lib_{FSH2}	1	0	1	0	0	0	0	1	0	0	0	1
pol_{NS1}	1	0	0	1	0	0	0	0	1	0	1	0
pol_{NS2}	1	0	0	1	0	0	0	0	1	0	1	0
pol_{FSH1}	1	0	0	0	1	0	0	0	1	0	0	1
pol_{FSH2}	1	0	0	0	1	0	0	0	1	0	0	1
pol_{FSH3}	1	0	0	0	1	0	0	0	1	0	0	1
$80s_{NS1}$	1	0	0	0	0	1	0	0	0	1	1	0
$80s_{NS2}$	1	0	0	0	0	1	0	0	0	1	1	0
$80s_{NS3}$	1	0	0	0	0	1	0	0	0	1	1	0
$80s_{FSH1}$	1	0	0	0	0	0	1	0	0	1	0	1
$80s_{FSH2}$	1	0	0	0	0	0	1	0	0	1	0	1
$80s_{FSH3}$	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.

Les hypothèses à tester

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

$$\log(\mu_{ijl}) = \log(s_j) + \delta_i + \beta_{ipol} + \beta_{i80s} + \theta_{i1} + \lambda_{i1pol} + \lambda_{i180s} \qquad (3)$$

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 littérature s'accorde sur le fait que l'on a une négative binomiale, montrer un histogramme des distributions, 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 H_O et sous H_1 :

$$\begin{cases} \mathbf{H}_0 : \lambda_{i0pol} = \lambda_{i1pol} \\ \mathbf{H}_1 : \lambda_{i0pol} \neq \lambda_{i1pol} \end{cases}$$

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 $modle_1$ avec le Log2FC pour la fraction pol, cela pour chaque gène i. Pour l'étude des coefficients, 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 :

$$\begin{cases} \mathbf{H}_0 : \theta_{i0} = \theta_{i1} \\ \mathbf{H}_1 : \theta_{i0} \neq \theta_{i1} \end{cases}$$

Le GLM utilisé pour le proteome

$$\log(\mu_{ij}) = \delta_i + \beta_i + \theta_{i\rho(j)} \tag{4}$$

Avec μ_{ijl} la moyenne des counts pour un gène i, et un répliqua j avec j allant de 1 à 9. $\rho(j)$ 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 :

$$\log(\mu_{ijl}) = \log(s_j) + \delta_i + \beta_{il} + \theta_{i\rho(j)} \tag{5}$$

— δ_i la quantité moyenne de counts pour un gène i

— β_{il} la quantité moyenne de counts pour chaque condition NS,S90,S4

— $\theta_{i\rho(j)}$ le différentielle d'expression global d'un gène au niveau des counts pour toutes les conditions.

la matrice associée à ce modèle

$$\begin{array}{c} \\ NS_1 \\ NS_2 \\ NS_3 \\ S90_1 \\ S90_2 \\ S90_3 \\ S4_1 \\ S4_2 \\ S4_3 \end{array} \begin{array}{c} \begin{array}{ccc} \delta & \theta_{FSH} & \beta_{S4} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$$

Les hypothèses à tester

On a tester plusieurs hypothèses par exemple :

La première hypothèse pour tester NS vs FSH, en supposant

$$\beta_{s4} = 0 \tag{6}$$

pour les deux modèles sous HO et sous H1 :

$$\begin{cases} \mathbf{H}_0 : \theta_{FSH} = 0 \\ \mathbf{H}_1 : \theta_{FSH} \neq 0 \end{cases}$$

```
ratiotestR_NSvsFSH <- function(x1,x2,x3,y1,y2,y3,z1,z2,z3){
  Y = c(x1,x2,x3,y1,y2,y3,z1,z2,z3)
  X1 = c(0,0,0,1,1,1,1,1,1)
  X2 = c(1,1,1,1,1,1,1,1,1)
  model0 <- glm(Y ~ X2-1, family = poisson )
  model1 <- glm(Y ~ X2 + X1-1, family = poisson )
  A <- lrtest(model0, model1)
  p_value <- A$‘Pr(>Chisq)‘
  return(p_value[2])}
```

La deuxième hypothèse pour tester NS vs S4, en supposant
pour les deux modèles sous HO et sous H1 :

$$\begin{cases} \mathbf{H}_0 : \beta_{s4} = 0 \\ \mathbf{H}_1 : \beta_{s4} \neq 0 \end{cases}$$

```
k est le vecteur de normalisation récupérer par DESeq
k=c(0.9972781,0.9790043,1.0603067,1.0519516,1.2131734,1.2624532,
1.0717103,0.9694117,0.9276979)
ratiotestR_NSvsS4_norm <- function(x1,x2,x3,y1,y2,y3,z1,z2,z3,k){
  # k=c(k1,k2,k3,k4,k5,k6,k7,k8,k9)
  Y = c(x1,x2,x3,y1,y2,y3,z1,z2,z3)
  X1 = c(1,1,1,1,1,1,0,0,0)
  X2 = c(0,0,0,1,1,1,0,0,0)
  model0 <- glm(Y ~ offset(log(k)) + X1-1,family = poisson)
  model1 <- glm(Y ~ offset(log(k))+X1 + X2-1,family = poisson )
  A <- lrtest(model0, model1)
  p_value <- A$‘Pr(>Chisq)‘
  return(p_value[2])}
```

Algorithme d'estimation des paramètres du GLM

On note $l(\beta)$ la vraisemblance du modèle linéaire associée au paramètre β. On rappelle que le paramètre $\hat{\beta}$ obtenu par la méthode du maximum de vraisemblance vérifie $\nabla l(\hat{\beta}) = 0$ (où ∇ désigne la différentielle de la fonction de vraisemblance l).Lorsque l'équation $\nabla l(\hat{\beta}) = 0$ étant non-linéaire, on ne dispose d'aucune méthode pour déterminer algébriquement le paramètre $\hat{\beta}$ solution de cette équation. On a donc recours à une méthode numérique pour estimer ce paramètre, et plus précisément à l'algorithme des moindres carrés

repondérés itérativement (en anglais iteratively reweighted least squares). Cette méthode s'appuie sur l'algorithme de Newton-Raphson, qui permet de construire une suite convergeant vers la solution d'une équation :
-la première étape de cette méthode consiste à choisir un point de départ β^0
-pour tout entier k, on construit ensuite l'itération $k+1$ telle que

$$\beta^{k+1} = \beta^k + A^k \nabla l(\beta^k),$$

où $A^k = -(\nabla^2 l(\beta^k))^{-1}$
-on poursuit jusqu'à ce que la suite stagne, c'est à dire lorsque β^{k+1} est assez proche de β^k De cette manière, on obtient $\hat{\beta}$ qui maximise la vraisemblance.

3.1.5 Vérification des anciens résultats

D'après les résultats d'une plateforme qui a déjà effectue une analyse statistique sur les données proteomique, on a fait des tests sur les données normaliser et parmi ces tests :

Le test de Student-Fisher

On considère deux échantillons de même taille n, correspondant aux réalisations de 2 variables indépendantes **X** et **Y** de moyenne et de variance respectives μ_1, μ_2 et σ_1,σ_2.
On s'intéresse dans notre cas à la question suivante : peut on affirmer que la moyenne et la variance de **X** et **Y** sont égales à partir du calculs des moyenne et des variances empiriques ? Ceci revient à tester les hypothèses :

$$\begin{cases} \mathbf{H}_0 : \mu_1 = \mu_2 \quad et \quad \sigma_1 = \sigma_2 \\ \mathbf{H}_1 : \mu_1 \neq \mu_2 \quad et \quad \sigma_1 \neq \sigma_2 \end{cases}$$

Le test de Student-Fisher se déroule en deux étapes : on teste dans un premier temps l'égalité des variances à l'aide du test de Fisher-Snedecor. Si l'on accepte l'hypothèse d'égalité, on teste alors l'égalité des moyennes à l'aide du test de Student en supposant que $\sigma_1 = \sigma_2$.

Pour pouvoir mettre en place ce test sous **R**, nous aurons besoin de deux fonctions : *var.test* et *t.test* pour respectivement les tests de Fisher et Student. Ces dernières sont contenues dans le package **stats** chargé par défaut dans la plupart des distributions (*library(help="stats")* pour plus de détails.

La fonction *var.test* prend comme arguments deux vecteurs de même taille correspondant aux deux échantillons observés dont la variance est à tester. Il est également utile de renseigner les options suivantes :

- **ratio** : la valeurs du rapport à tester entre σ_1 et σ_2. Par défaut, vaut 1.
- **alternative** : une chaîne de caractère spécifiant le type d'hypothèse alternative à choisir parmi two.sided (par défaut) pour un test bilatéral, greater ou less.
- **conf.level** : renseigne le niveau de confiance pour l'intervalle de confiance affiché en sortie. Le niveau par défaut est 0.95.

Intéressons nous à présent à la mise en place du test de Student (dans l'éventualité ou le test précédant inciterait à conclure que les variances des deux échantillons sont égales). La fonction *t.test* prend comme argument les deux échantillons à tester. Les options sont essentiellement les mêmes que pour le test de Fisher. Il est cependant nécessaire de préciser que les variances sont supposées égales en donnant la valeur **TRUE** à la variable *var.equal*. Dans le cas contraire, c'est une variante du test de Student qui est utilisée.

Le Wald test

Le critère principal de ce test est d'examiner si la contrainte

$$g(\theta) = 0 \tag{7}$$

Est approximativement satisfait par $\hat{\theta_n}$,lorsque $g(\hat{\theta_n})$proche de 0.

D'après les propriétés asymptotiques la région critique est définie par :

$$\varsigma_n^w \geqslant \chi^2_{1-\alpha}(r) \tag{8}$$

avec

$$\varsigma_n^w = ng^t(\hat{\theta_n})(\frac{\partial g(\hat{\theta_n})}{\partial^t \theta} I(\hat{\theta_n})^{-1} \frac{\partial g^t(\hat{\theta_n})}{\partial \theta})^{-1} g(\hat{\theta_n}) \tag{9}$$

A un niveau asymptotique α cohérent
θ : le vecteur des paramétrés a estimer
I : la matrice d'information de fisher

```
fc<-function(y,g,v)
{
  invI<-solve(y)
  II<-t(v)%*%invI%*%v
  stat<-3*g^2*II^(-1)
  pv<-2*(1-pchisq(abs(stat),df=1))
  return(pv)
}


#statwaldNSvsS4<-rep(0,4873)
pvwaldNSvsS4<-rep(0,4873)

for(i in 1:4873)
{
  I<-matrix(data=rep(0,16), nrow = 4, ncol=4)
  invI<-matrix(data=rep(0,16), nrow = 4, ncol=4)

  if(var0[i]!=0 & var1[i]!=0){
    I[1,]=c(1/var0[i] , 0 , sum(NS[i,]-mu0[i])/3*(var0[i])^2 , 0 )
    I[2,]=c(0 , 1/var1[i] , 0 , sum(S4[i,]-mu1[i])/3*(var1[i])^2)
    I[3,]=c(sum(NS1[i,]-mu0[i])/3*(var0[i])^2 , 0, -1/(2*(var0[i]^2))
    +sum((NS[i,]-mu0[i])^2)/(3*var0[i]) , 0)
    I[4,]=c(0 , sum(S4[i,]-mu1[i])/3*(var1[i]^2) , 0 , -1/(2*(var1[i]^2))
    +sum((S4[i,]-mu1[i])^2)/(3*var1[i]))
    pvwaldNSvsS4[i]= fc(I,g[i],v)
  }
}
```

Après on a fait une comparaison entre les résultats de la plateforme et les résultats de DESeq et aussi entre les résultats de DESeq et GLM.

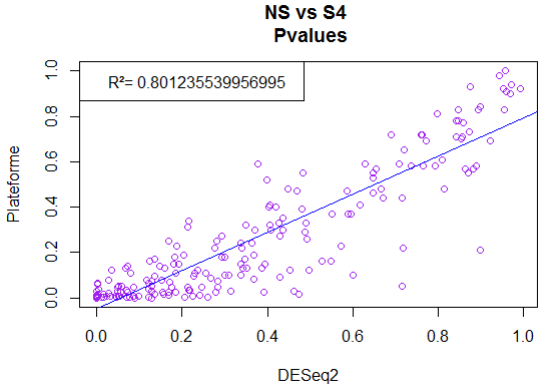

FIGURE 12 – La comparaison des pvalues de DESeq et plateforme

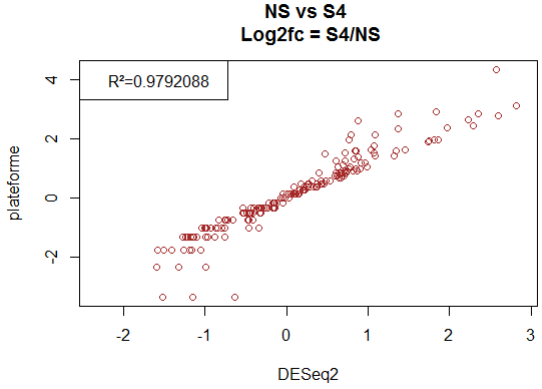

FIGURE 13 – La comparaison des LogFoldChange de DESeq et plateforme

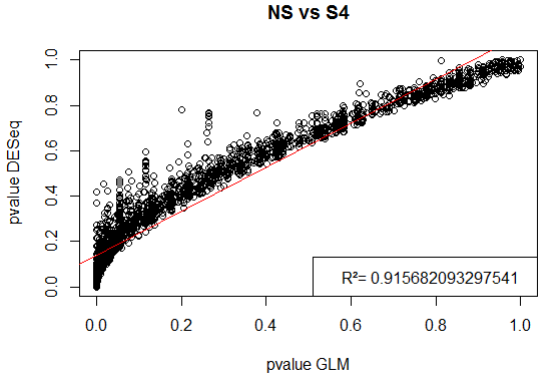

FIGURE 14 – La comparaison des pvalues de DESeq et GLM

NS vs S4

Tests	Nbr gènes diff exprimer
Fisher - Student	1196
Wald	937
DESeq	427
GLM	1095

FIGURE 15 – Le nombre des gènes différentiellement exprimées pour chaque test

3.2 Deuxième problématique :Corrélation traductome versus proteome

3.2.1 Analyse avec DESeq2

On a fait le matche entre les deux résultats de DESeq2 pour le proteome et le traductome, on passant par un tableau qui est le lien entre ces deux résultats et voici un extrait de ces trois tableaux :

A	
ENS	GeneID
ENSRNOG00000033195	A1cf
ENSRNOG00000028896	A2m
ENSRNOG00000007247	A2ml1
ENSRNOG00000005935	A3galt2
ENSRNOG00000042649	A4gnt
ENSRNOG00000000478	AA926063
ENSRNOG00000000967	Aacs
ENSRNOG00000013950	Aadac
ENSRNOG00000050494	Aadacl2
ENSRNOG00000026613	Aadacl3
ENSRNOG00000037030	Aadacl4
ENSRNOG00000018886	Aaed1
ENSRNOG00000008424	Aagab
ENSRNOG00000014399	Aamp
ENSRNOG00000020086	Aar2
ENSRNOG00000004708	Aard
ENSRNOG00000018404	Aars
ENSRNOG00000025808	Aars2
ENSRNOG00000020658	Aarsd1

FIGURE 16 – Le lien entre traductome et proteome

A	B	C	D	E	F	G	H
rowname	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	
ENSRNOG00000000001	7.73715406599618	0.0772078500884722	0.587554657595205	0.131405391975744	0.895454623471402	NA	
ENSRNOG00000000007	97.8452300035796	-0.172851010492722	0.286400431595742	-0.603529155070211	0.546156726198953	0.814080234174913	
ENSRNOG00000000008	105.543034600767	-0.15372670987158	0.192466967474011	-0.79871736895495	0.424454313118522	0.743226285236717	
ENSRNOG00000000009	50.5437793904442	0.475594809398706	0.297725585449741	1.59742673334668	0.110170617327132	0.512323997624352	
ENSRNOG00000000012	35.2650470440921	0.423830104468128	0.358625326660596	1.18181866410467	0.237277660461753	0.633591684194896	
ENSRNOG00000000017	53.0547159947461	-0.25140821354303	0.288054188773534	-0.872780967405703	0.382782475463504	0.722517542602401	
ENSRNOG00000000024	123.87178851668	0.0869894637206375	0.238585486098769	0.364605010736596	0.715406273948968	0.893917807129914	
ENSRNOG00000000033	245.470934962837	0.0330250385670839	0.167426877654914	0.197250519329115	0.84363149840564	0.946473947338203	
ENSRNOG00000000034	72.2663431344788	0.202806587620434	0.282385721069861	0.718189952565838	0.472640172228322	0.773100412620705	
ENSRNOG00000000035	94.9230795058553	-0.0988167264354931	0.224530109966118	-0.440104565264787	0.659861375603762	0.868483837539943	
ENSRNOG00000000040	44.9151105713208	0.368661124736904	0.313375179919903	1.1764209432002	0.239426706789549	0.635061943434872	
ENSRNOG00000000042	682.178097145845	0.143097133355488	0.0882884549678172	1.62079100158282	0.10506246899646	0.503655415745132	
ENSRNOG00000000043	62.4594611516539	0.0360223210487024	0.299527312553571	0.120263894272613	0.904274104672421	0.967697334981824	
ENSRNOG00000000044	157.887548030942	-0.17491680431761	0.326378010125147	-0.535933178373565	0.592004733151526	0.833836665884666	
ENSRNOG00000000047	115.873684182583	-0.0532137912798091	0.255003814024939	-0.208678413235831	0.834699292883043	0.942828967057461	
ENSRNOG00000000048	614.560710293974	-0.217566233199825	0.306170961304164	-0.710603749856227	0.477329818731576	0.775557648191639	
ENSRNOG00000000053	199.046530861135	-0.0258377425369616	0.148309886817891	-0.174214565807657	0.861696838723766	0.95309537670066	
ENSRNOG00000000054	42.5966265091611	0.4841598022627	0.332038778992131	1.45814234027819	0.14480131295177	0.553773719668984	
ENSRNOG00000000055	53.6459438551136	-0.0835884022168505	0.272799048131732	-0.30641016817803	0.759292367367727	0.912628907932114	
ENSRNOG00000000060	198.58735027009	-0.0331157101392884	0.207156898695163	-0.159858109229657	0.872992848056002	0.957104377538175	
ENSRNOG00000000062	360.312031958738	1.21731146032548	0.226735670846747	5.3688572943967	7.92370846813051e-08	5.83019865994653e-05	
ENSRNOG00000000064	6.90072733832987	0.354170620385396	0.556409397546053	0.636528825622651	0.524431798990737	NA	
ENSRNOG00000000065	263.864172192795	-0.0054291487980314	0.15852929461871	-0.0342469750533454	0.972680207815605	0.992058212680289	
ENSRNOG00000000066	171.105579038211	-0.306785324838556	0.19303837205283	-1.58924529654962	0.112005023638839	0.515822376126769	
ENSRNOG00000000068	329.647856222928	-0.139613956070815	0.418615277504311	-0.333513762094784	0.738746502882633	0.904619963553458	

FIGURE 17 – Le résultat de DESeq2 pour une fraction du traductome

A	B	C	D	E	F	G	H
GeneID	baseMean_NSvsS4	l2fc_NSvsS4	lfcSE_NSvsS4	stat_NSvsS4	pvalue_NSvsS4	padj_NSvsS4	
Stx5	1.34391192018831	0.853797348338245	0.901462691628388	0.947124441496252	0.343575371512526	NA	
Ptpn11	1.99728233724105	1.45695485812124	0.860000250155428	1.69413306316821	0.0902400008121082	0.315659522840754	
P4hb	35.4492899180613	-0.613062042161439	0.370685960177727	-1.65385827363816	0.098156312436229	0.332060716539583	
A2m	2.67824288300754	-0.788893778463532	0.807638964911249	-0.976790141062873	0.328673057909039	0.640143851094555	
Adprh	1.30308197338134	0.910052247570537	0.889632424858943	1.02295310078748	0.306330020892703	NA	
Aldoa	81.2613682766287	-0.172639854672993	0.208396651744966	-0.828419522230466	0.40743295432969	0.704583145949261	
Cpt1a	11.3678211892377	-0.848223984226485	0.535211197354243	-1.58483975750056	0.113002746743133	0.36264551202521	
Fasn	183.794970203734	-1.09598046641561	0.177439178530149	-6.17665430765841	6.54742159260168e-1	1.635920052208 62e-(	
Gstm1	130.422264575864	-0.346561412638776	0.203584849539357	-1.70229471113851	0.0887001357683146	0.313757288986419	
Lgals3	3.63660813274797	1.76591263539325	0.763019387804407	2.31437452785397	0.0206471868780421	0.117246525486025	
Lamp2	9.35551098835974	-0.324102509975172	0.504062402091903	-0.642980925834021	0.520236476700554	0.784361972298745	
NA	5.31753140822179	3.19217030722996	0.808416553452706	3.94867014238682	7.85865510764205e-0	0.0021953784815977	
Rab8a	1.10402244246416	-1.38987530457087	0.905842033124851	-1.53434622566173	0.12494449925311	NA	
Hibadh	21.2965648015461	0.593091345394194	0.366639096091398	1.6176434856973	0.105739451811581	0.349132257845653	
Cyp51	1.17125072221804	-0.545466181655395	0.902194223801561	-0.604599505588689	0.545445135261241	NA	
Dync1i2	4.44391777948184	1.81411732072041	0.746117908106336	2.43140836188302	0.0150402523905207	0.0974274127074842	
Fmr1	0.577012872623754	0.774081422305196	0.812229981872147	0.953032317916877	0.340573696443641	NA	
Scfd1	4.20355424868876	1.94670153722157	0.757276390580619	2.5706618632716	0.0101504380271698	0.0742808205419249	
Eif2b5	1.82559354770423	1.13990188838191	0.903580198544706	1.26153925265053	0.207114626996445	0.504517385260142	
NA	2.21814058218165	-0.284192980905974	0.833659172643217	-0.340898283413479	0.733180157137457	0.89311927569138	
NA	2.07767270309598	-0.166891352495922	0.824983085252376	-0.202296696113311	0.839684782701841	0.954263518462939	
Kyat1	1.23620658768211	-0.162738304281795	0.896689738196918	-0.181487862913467	0.855984658955906	NA	
NA	0.651809009722026	-0.337111236864828	0.883431153572815	-0.38159310490859	0.702763199398637	NA	

FIGURE 18 – Le résultat de DESeq2 pour une condition du proteome

Visualisation de la corrélation

Et après pour visualiser la corrélation on a fait une simple comparaison entre les **logfoldchange** du proteome et **logfoldchange** du traductome représenter par le graphique suivant :

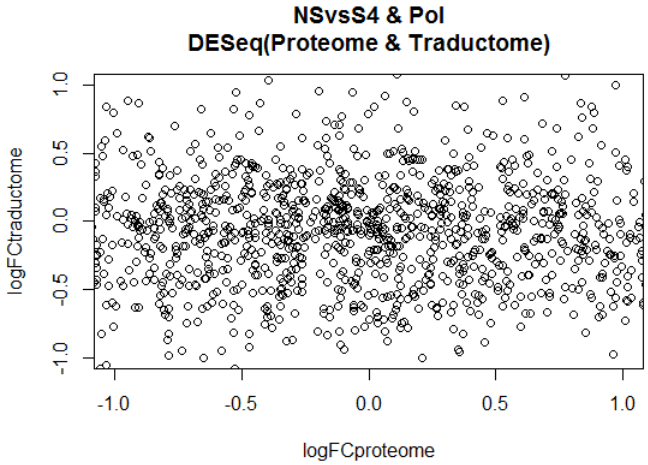

FIGURE 19 – Comparaison résultat de DESeq (Logfc)

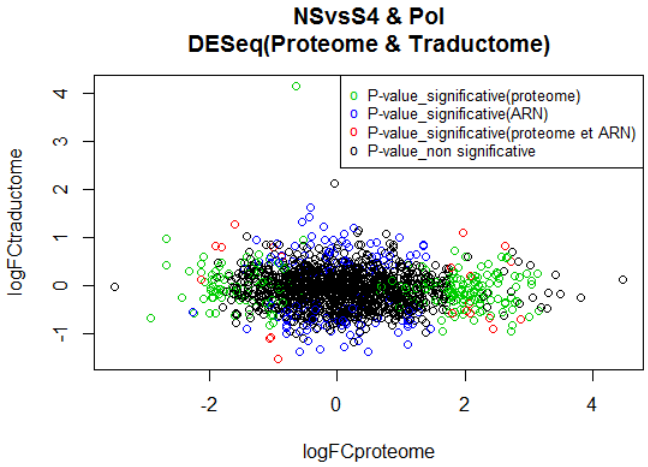

FIGURE 20 – Comparaison résultat de DESeq (Logfc Significatif)

3.2.2 Analyse avec GLM

Le modèle utilisé

$$\log(\mu_{ij}) = \delta_{prot} + \delta_{trad} + \beta_{80s} + \beta_{pol} + \lambda_{80s}^{FSH} + \lambda_{pol}^{FSH} + \lambda_{prot}^{FSH} + \lambda_{s4} + \theta_{FSH} \quad (10)$$

Matrice de codage disjonctif complet

	δ_{trad}	δ_{prot}	β_{80s}	β_{pol}	λ_{80s}^{FSH}	λ_{pol}^{FSH}	λ_{prot}^{FSH}	$\lambda s4$	θ_{FSH}
NS_{lib1}	1	0	0	0	0	0	0	0	0
NS_{lib2}	1	0	0	0	0	0	0	0	0
NS_{lib4}	1	0	0	0	0	0	0	0	0
NS_{80s1}	1	0	1	0	0	0	0	0	0
NS_{80s2}	1	0	1	0	0	0	0	0	0
NS_{80s4}	1	0	1	0	0	0	0	0	0
NS_{pol2}	1	0	0	1	0	0	0	0	0
NS_{pol4}	1	0	0	1	0	0	0	0	0
FSH_{lib2}	1	0	0	0	0	0	0	0	1
FSH_{lib4}	1	0	0	0	0	0	0	0	1
FSH_{80s1}	1	0	1	0	1	0	0	0	1
FSH_{80s2}	1	0	1	0	1	0	0	0	1
FSH_{80s4}	1	0	1	0	1	0	0	0	1
FSH_{pol1}	1	0	0	1	0	1	0	0	1
FSH_{pol2}	1	0	0	1	0	1	0	0	1
FSH_{pol4}	1	0	0	1	0	1	0	0	1
NS_{prot1}	0	1	0	0	0	0	0	0	0
NS_{prot2}	0	1	0	0	0	0	0	0	0
NS_{prot3}	0	1	0	0	0	0	0	0	0
$S90_1$	0	1	0	0	0	0	1	0	1
$S90_2$	0	1	0	0	0	0	1	0	1
$S90_3$	0	1	0	0	0	0	1	0	1
$S4_1$	0	1	0	0	0	0	1	1	1
$S4_2$	0	1	0	0	0	0	1	1	1
$S4_3$	0	1	0	0	0	0	1	1	1

Les hypothèses a tester

On veut répondre a deux questions :

— Est ce que un gène est globalement différentiellement exprimés en proteome et en traductome en même temps ?

— Est ce qu'il est plus ou moins différentiellement exprimés en proteome par rapport au traductome ?

Nous utiliserons les résultats obtenus grâce à un test du ratio de maximum de vraisemblance avec une loi poisson.

La première hypothèse pour répondre a la première question, en supposant

$$\lambda_{80s}^{FSH} = \lambda_{pol}^{FSH} = \lambda_{prot}^{FSH} = \lambda s4 = \beta_{pol} = \beta_{80s} = 0 \tag{11}$$

pour les deux modèles sous HO et sous H1 :

$$\begin{cases} \mathbf{H}_0 : \theta_{FSH} = 0 \\ \mathbf{H}_1 : \theta_{FSH} \neq 0 \end{cases}$$

Les résultats sont plutôt concluant, en utilisant toujours le facteur de normalisation$k$ obtenu avec DESeq2.

```
k1=coeffnormProt et k2=coeffnormARN
k1=c(0.99,0.97,1.06,1.051,1.21,1.26,1.071,0.96,0.92)
k2=c(1.38,1.98,1.70,1.76,1.55,1.20,0.87,0.79,0.84,1.33,0.85,0.65,0.36,
1,0.73,0.58)
k=c(k2,k1)
ratiotestR_NSvsFSH <- function(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,
x14,x15,x16,x17,x18,x19,x20,x21,x22,x23,x24,x25){
  Y=c(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,
  x19,x20,x21,x22,x23,x24,x25)

  model0 <- glm(Y ~ offset(k)+deltaprot+deltatrad-1, family = poisson )
  model1 <- glm(Y ~ offset(k)+deltaprot+deltatrad+tetaFSH-1, family = poisson )
  A <- lrtest(model0, model1)
```

La deuxième hypothèse pour répondre a la deuxième question, en supposant

$$\lambda_{80s}^{FSH} = \lambda_{pol}^{FSH} = \lambda_{prot}^{FSH} = \lambda s4 = \beta_{pol} = \beta_{80s} = 0 \tag{12}$$

pour les deux modèles sous HO et sous H1 :

$$\begin{cases} \mathbf{H}_0 : \lambda_{prot}^{FSH} = 0 \\ \mathbf{H}_1 : \lambda_{prot}^{FSH} \neq 0 \end{cases}$$

Les résultats sont plutôt concluant, en utilisant toujours le facteur de normalisation$k$ obtenu avec DESeq2.

```
k1=coeffnormProt et k2=coeffnormARN
k1=c(0.99,0.97,1.06,1.051,1.21,1.26,1.071,0.96,0.92)
k2=c(1.38,1.98,1.70,1.76,1.55,1.20,0.87,0.79,0.84,1.33,0.85,0.65,0.36,
1,0.73,0.58)
k=c(k2,k1)
ratiotestR_ProtvsARN <- function(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,
x12,x13,x14,x15,x16,x17,x18,x19,x20,x21,x22,x23,x24,x25){
  Y = c(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,
  x21,x22,x23,x24,x25)

model0 <- glm(Y ~ offset(k)+deltaprot+deltatrad+tetaFSH-1, family = poisson )
model1 <- glm(Y ~ offset(k)+deltaprot+deltatrad+tetaFSH+lambdaFSHprot-1,
family = poisson)
  A <- lrtest(model0, model1)
  p_value <- A$`Pr(>Chisq)`
  return(p_value[2])}
```

Les résultats de la corrélation

EN COURS !

4 Conclusion

La transformation des données brutes en tableau de comptage m'a appris qu'un statisticien doit pouvoir manipuler les données brutes qu'il doit traiter, quelle que soit leur nature. Enfin, la partie « normalisation et analyse des données » m'a permis de compléter mes connaissances en statistique et de progresser dans l'utilisation du logiciel RStudio. En travaillant avec des biologistes, j'ai pu comprendre qu'un script se devait d'être clair, reproductible et adapté au niveau et aux exigences de l'utilisateur. Cette expérience professionnelle m'a confortée dans mon choix de poursuivre mes études en statistique. Cette discipline débouche sur un grand choix de carrières professionnelles, permet de travailler dans une multitude de champs d'applications, avec des personnes aux parcours professionnels différents et c'est en cela, je pense, que réside la richesse du métier de statisticien.

5 Annexes

Script DESeq2

EN COURS !

Script GLM

EN COURS !

BIBLIOGRAPHIE

EN COURS !