

Analyse de données génomiques

Sélection de gènes

David Causeur
Institut Agro Rennes Angers
IRMAR UMR 6625 CNRS

23 mai, 2025

Objectifs

- ▶ **Acquérir des compétences d'analyse de données génomiques**
- ▶ Méthodes (normalisation, tests à l'effet du génome, contrôle des faux positifs)
- ▶ Démarches (comment choisir une méthode d'analyse ?)
- ▶ **Se familiariser avec R et Rstudio**
- ▶ Mettre en oeuvre (limma, DESeq2)
- ▶ Editer des rapports (Rmarkdown)


Modalités pédagogiques

- ▶ Apprentissage par mise en situation (tutoriel)
- ▶ Programme modulable selon questions

Environnement R et Rstudio

 : environnement pour l'analyse de données et la visualisation

- ▶ R est un logiciel libre
- ▶ Web : <https://www.r-project.org>
- ▶ 22 000 packages (Mai 2025) <https://cran.r-project.org/>
- ▶ Communautés R :
 - ▶ <https://www.r-bloggers.com/>,
 - ▶ <https://rladies.org/>,
 - ▶ <https://r-toulouse.netlify.app/>
 - ▶ <https://stackoverflow.com/questions/>
 - ▶ <https://r-graph-gallery.com/>

 : interface pour R (IDE : Integrated Development Environment)

- ▶ Web : <https://www.rstudio.com/>
- ▶ Environnement ergonomique pour l'analyse de données
- ▶ Outils pour l'édition de rapport d'études
 - ▶ Rmarkdown : <http://rmarkdown.rstudio.com/>
- ▶ Préparation de la session de travail
 - ▶ Créez un projet pour l'analyse des données RNA-seq

Plan

- ▶ 1 - Préparation des données RNA-seq
- ▶ 2 - Tests à l'échelle du génome
 - ▶ 2.1 - Normalisation des données
 - ▶ 2.2 - Modèles pour données de comptage
- ▶ 3 - Sélection de gènes
 - ▶ Contrôle des faux positifs
 - ▶ Procédures de sélection

Préparation des données

Importation des données dans la session de travail R

```
dta <- read.delim("./data/DataTest_foie.txt",  
                  header=TRUE,  
                  stringsAsFactors=FALSE)
```

dta - tableau de données d'expression :

```
head(dta[,1:8])
```

	F10_G_B	F11_M_H	F13_G_H	F14_M_H	F15_G_B	F16_M_B	F17_M_B	F18_G_B
CD69	124	149	210	339	161	352	216	96
GOLGB1	1918	2002	2052	2478	1639	2374	2083	1459
HCLS1	745	392	1250	817	901	813	694	788
RABL2A	29	61	44	55	80	54	33	53
SHANK3	113	148	185	203	186	181	176	116
GCC1	436	371	579	601	575	788	519	422

dta - dimensions :

```
dim(dta)
```

```
[1] 10708    15
```


Plan expérimental

Groupes expérimentaux (noms des colonnes) :

```
labs <- colnames(dta)
head(labs)
```

```
[1] "F10_G_B" "F11_M_H" "F13_G_H" "F14_M_H" "F15_G_B" "F16_M_B"
```

Plan 2x2 (Génotype x Régime) :

```
group <- substring(labs,first=5,last=7)
geno  <- substring(labs,first=5,last=5)
diet  <- substring(labs,first=7,last=7)
table(geno,diet,dnn=list("Génotype","Régime"))
```

	Régime	
Génotype	B	H
G	4	4
M	3	4

Association d'une Couleur à chaque groupe expérimental

```
colors <- as.factor(group)
levels(colors) <- wes_palette(4, name = "GrandBudapest2")
colors <- as.character(colors)
```

Format DGE (Differential Gene Expression, R package edgeR)

Création d'un objet DGE

```
dge <- DGEList(counts=dta,group=group)
names(dge)
```

```
[1] "counts" "samples"
```

Composante counts

```
head(dge$counts[,1:8],n=3)
```

	F10_G_B	F11_M_H	F13_G_H	F14_M_H	F15_G_B	F16_M_B	F17_M_B	F18_G_B
CD69	124	149	210	339	161	352	216	96
GOLGB1	1918	2002	2052	2478	1639	2374	2083	1459
HCLS1	745	392	1250	817	901	813	694	788

Composante samples

```
head(dge$samples,n=3)
```

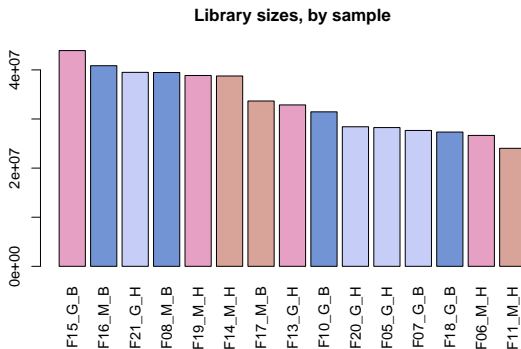
	group	lib.size	norm.factors
F10_G_B	G_B	31455867	1
F11_M_H	M_H	24027197	1
F13_G_H	G_H	32862911	1

Normalisation des données

Profondeur de séquençage

Profondeur de séquençage : nombre de *reads* séquencés par échantillon biologique

```
seq_depth <- colSums(dta)
barplot(seq_depth[order(seq_depth,decreasing=TRUE)],
        col=colors,main= "Library sizes, by sample", las=3)
```



Facteurs de correction pour la normalisation

Données RNA-seq : abondance relative de *reads* séquencés

Dans un échantillon : si un petit nombre de gènes monopolise un grand nombre de reads, l'expression des autres est sous-évaluée

Normalisation : corriger les profondeurs de séquençages pour garantir des niveaux moyens d'expression similaires entre échantillons pour les gènes les moins exprimés

TMM : trimmed mean of M-values (Robinson and Oshlack, 2010)

```
dge <- normLibSizes(dge,method="TMM")  
head(dge$samples[1:5,])
```

	group	lib.size	norm.factors
F10_G_B	G_B	31455867	0.9701708
F11_M_H	M_H	24027197	0.9494259
F13_G_H	G_H	32862911	1.0039536
F14_M_H	M_H	38757569	1.0349501
F15_G_B	G_B	43928923	0.9264295

Analyse différentielle

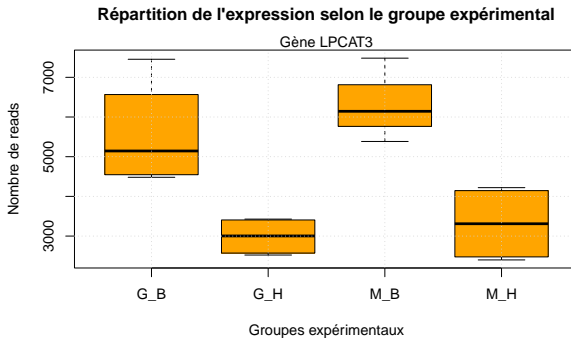
Sélection de gènes

Objectif : identifier les gènes dont l'expression moyenne dépend des conditions expérimentales (lignée, régime, ...)

Méthode : tests d'un même effet sur chaque gène

Exemple du gène LPCAT3

```
boxplot(dge$counts["LPCAT3",,drop=TRUE]~group,col="orange",  
        xlab="Groupes expérimentaux",ylab="Nombre de reads",  
        main="Répartition de l'expression selon le groupe expérimental")  
mtext("Gène LPCAT3")  
grid()
```

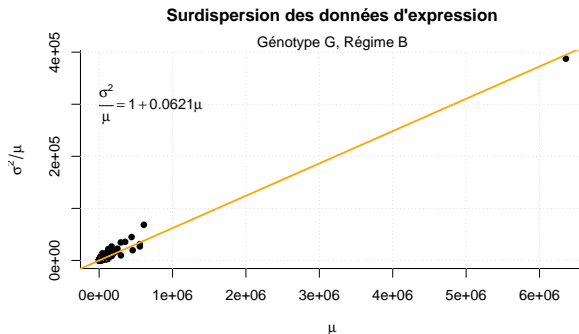


Surdispersion des données de comptage

Surdispersion : Relation entre variance et moyenne d'expression

$$\frac{\sigma^2}{\mu} = 1 + \alpha\mu > 1$$

Exemple dans le groupe G_B



Modèle de régression pour données surdispersées

Y : expression d'un gène (nombre de *reads* séquencés)

$$Y \sim \text{BN}(\mu; \alpha), \quad [\text{BN} : \text{Binomiale Négative}]$$

où

- ▶ $\mu \geq 0$: expression moyenne
- ▶ $\alpha > 0$: coefficient de surdispersion

Modèle linéaire généralisé (GLM)

$$\begin{aligned}\log \mu(x) &= \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p \\ \log \mu_{ij} &= \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} \\ &\vdots\end{aligned}$$

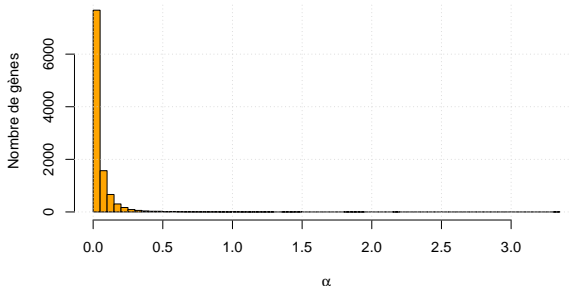
Sur-dispersion à l'échelle du génome

Estimation des coefficients de surdispersion pour chaque gène

```
design <- model.matrix(~diet+geno,data=data.frame(diet=factor(diet),
                                                    geno=factor(geno)))

dge <- estimateDisp(dge,design)
hist(dge$tagwise.dispersion,nclass=50,col="orange",
     main="Répartition des coefficients de surdispersion",
     xlab=expression(alpha),ylab="Nombre de gènes")
grid()
```

Répartition des coefficients de surdispersion



GLM à l'échelle du génome

Modèles : Pour le k ème gène, dans le régime i , $Y_i^{(k)} \sim \text{BN}(\mu^{(k)} + \alpha_i^{(k)}; \alpha^{(k)})$

$\alpha_2^{(k)}$: log-fold change (log-ratio des expressions moyennes par régime)

```
fit <- glmFit(dge, design)
```

Tests de l'effet régime : Pour le k ème gène, on teste $H_0^{(k)} : \alpha_2^{(k)} = 0$

```
tests <- glmLRT(fit, coef=2)
head(tests$table)
```

	logFC	logCPM	LR	PValue
CD69	0.16994359	2.5178451	1.32254298	0.25013617
GOLGB1	0.02711516	5.8955646	0.05175731	0.82003317
HCLS1	0.39504502	4.7751536	4.98180490	0.02561525
RABL2A	0.46434772	0.7887187	3.78873315	0.05159869
SHANK3	0.22036680	2.3772877	2.19008571	0.13890159
GCC1	0.01828934	4.0094943	0.02625765	0.87127267

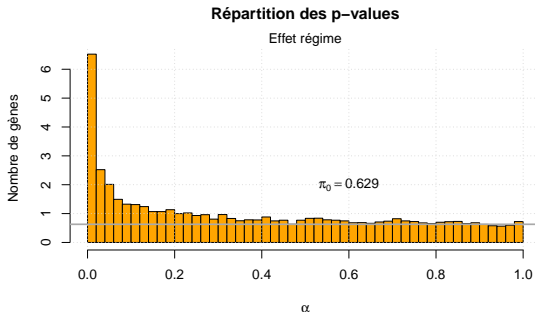
Si $H_0^{(k)}$ n'est pas vraie, on dit que le k ème gène est *differentially expressed* (DE)

Si $H_0^{(k)}$ est rejetée, on dit que le k ème gène est *positif*

Proportion de gènes différentiellement exprimés

```
pval <- tests$table$PValue  
pi0 <- pval.estimate.eta0(pval,diagnostic.plot=FALSE)
```

```
hist(pval,nclass=50,col="orange",proba=TRUE,  
     main="Répartition des p-values",  
     xlab=expression(alpha),ylab="Nombre de gènes")  
mtext("Effet régime")  
abline(h=pi0,lwd=2,col="darkgray")  
text(0.6,2,bquote(pi[0]==.(round(pi0,digits=3))))  
grid()
```



Identification des gènes positifs

Combien de gènes positifs ?

```
positives_LRT <- pval<=0.05  
select <- which(positives_LRT)  
P <- length(select)  
P
```

[1] 2173

Combien de gènes positifs si l'expression moyenne de tous les gènes était la même dans les deux régimes ?

Définition : un gène est **faux positif** si la procédure de test conclut que l'effet testé est significatif ($p\text{-value} \leq t$), alors que cet effet n'existe pas.

- ▶ Nombre de gènes positifs : P_t
- ▶ Nombre de gènes faux positifs : V_t (non-observable)
- ▶ Proportion de faux positifs : $FDP_t = \frac{V_t}{P_t}$ (non-observable)

Taux de faux positifs (False Discovery Rate) : $FDR_t = \mathbb{E}(FDP_t)$

Comment choisir le seuil t pour garantir $FDR_t \leq 0.05$?

Estimation du FDR

Supposons $m_0 \leq m$ gènes non DE dans $\mathcal{M}_0 = \left\{ k = 1, \dots, m, H_0^{(k)} \text{ est vraie} \right\}$.

Pour chacun de ces m_0 gènes, $\mathbb{P}(p_k \leq t) = t$

Comme $V_t = \# \{k \in \mathcal{M}_0, p_k \leq t\}$, $\mathbb{E}(V_t) = m_0 t$.

Estimation du FDR : $\widehat{FDR}_t = \frac{m_0 t}{\bar{P}_t} = \pi_0 \frac{mt}{\bar{P}_t}$

Exemple avec $t = 0.05$

```
m <- length(pval)
P <- sum(pval <= 0.05)
FDR <- pi0*m*0.05/P ; FDR
```

```
[1] 0.1550223
```


Méthode de Benjamini-Hochberg

Principe : le seuil t est le plus grand possible pour lequel $\widehat{\text{FDR}}_t \leq 0.05$

Sélection pour un contrôle du FDR au seuil de 0.05

```
fdr <- pi0*p.adjust(pval,method="BH")  
sum(fdr <= 0.05)
```

```
[1] 806
```

Seuil de décision

```
p_sort <- sort(pval)  
fdr <- pi0*m*p_sort/(1:m)  
seuil <- max(p_sort[fdr<=0.05]) ; seuil
```

```
[1] 0.00597943
```

Sélection multi-critères

Deux critères de sélection :

- ▶ p-value corrigée (BH) inférieure au seuil choisi pour le contrôle du FDR
- ▶ différence d'expression (log-fold change) suffisamment grande

```
res_tests <- decideTests.DGELRT(tests,adjust.method="BH",  
                                p.value=0.05,  
                                lfc=1)  
  
summary(res_tests)
```

```
      dietH  
Down      49  
NotSig 10637  
Up        22
```

Volcano plot

```
log_fc <- tests$table$logFC
plot(log_fc, -log10(pval), pch=16, xlab="log fold change",
      ylab="p-values (Manhattan)", main="Volcano plot", col="darkgray")
mtext("Effet régime")
abline(h=-log10(seuil), v=c(-1,1), lwd=2, col="orange")
points(log_fc[res_tests!=0], -log10(pval[res_tests!=0]), pch=16)
grid()
```

