

(Re)découverte de R... en 1h45

<https://tinyurl.com/r-intro-ebai20>

Ecole de Bioinformatique AVIESAN-IFB – Roscoff – Octobre 2020

Hugo Varet – hugo.varet@pasteur.fr

Thomas Denecker – thomas.denecker@gmail.com

Jacques van Helden – jacques.van-helden@univ-amu.fr

aviesan

alliance nationale
pour les sciences de la vie et de la santé



CNRS UPMC

**Station Biologique
Roscoff**

R en quelques mots

Langage de programmation qui permet de :

1. manipuler des données : importer, transformer, exporter
2. faire des analyses statistiques plus ou moins complexes : description, exploration, modélisation...
3. créer des (jolies) figures

Disponible sur



Historique :

- 1993 : début du projet R
- 2000 : sortie de R 1.0.0
- 2020 : R 4.0.2

Avantages et inconvénients

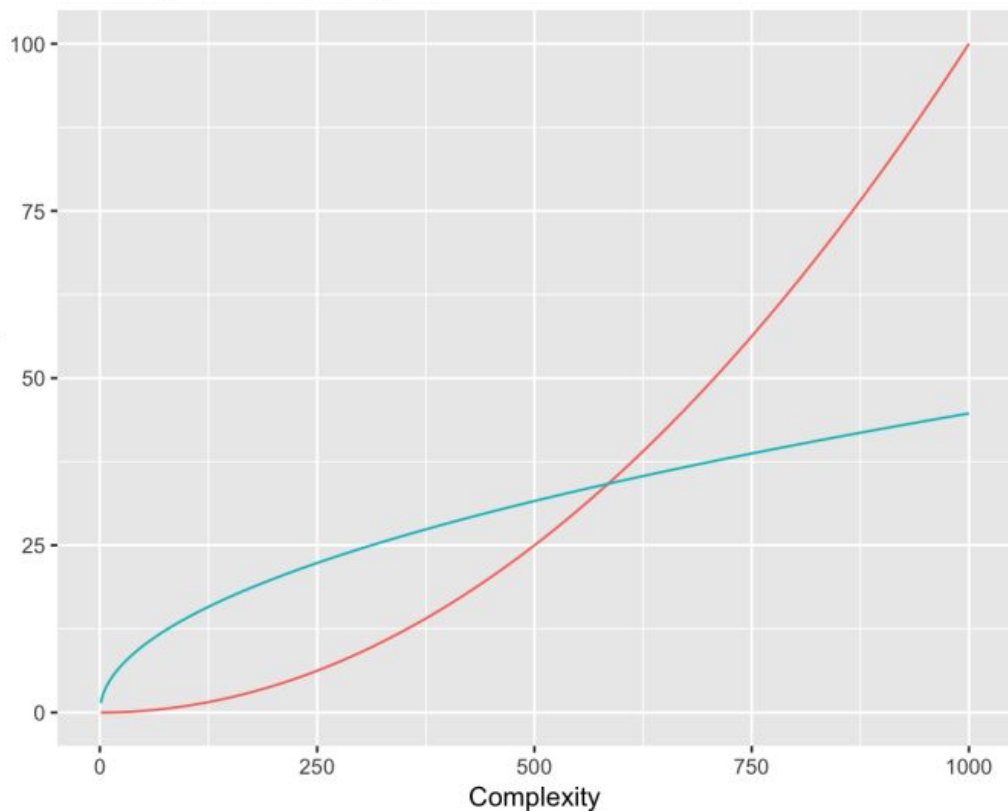
Avantages :

- Souplesse d'utilisation pour réaliser des analyses statistiques
- R est libre et gratuit, même s'il existe maintenant des versions payantes de RStudio (shiny et/ou server)
- Reproductibilité des analyses en écrivant/sauvegardant les commandes R dans des scripts
- Large communauté d'utilisateurs/aide en ligne
- Grand nombre de packages spécifiques

Inconvénients :

R vs Excel

Difficulty vs. Complexity



tool
— Excel
— R

Covid : le Royaume-Uni passe à côté de milliers de cas à cause... d'un fichier Excel arrivé à saturation

Les autorités sanitaires britanniques ont reconnu que près de 16.000 cas de coronavirus en Angleterre sont passés sous le radar au cours de la semaine écoulée à cause d'un problème dans le chargement des données.

[Lire plus tard](#) [Europe](#) [Partager](#) [Commenter](#)

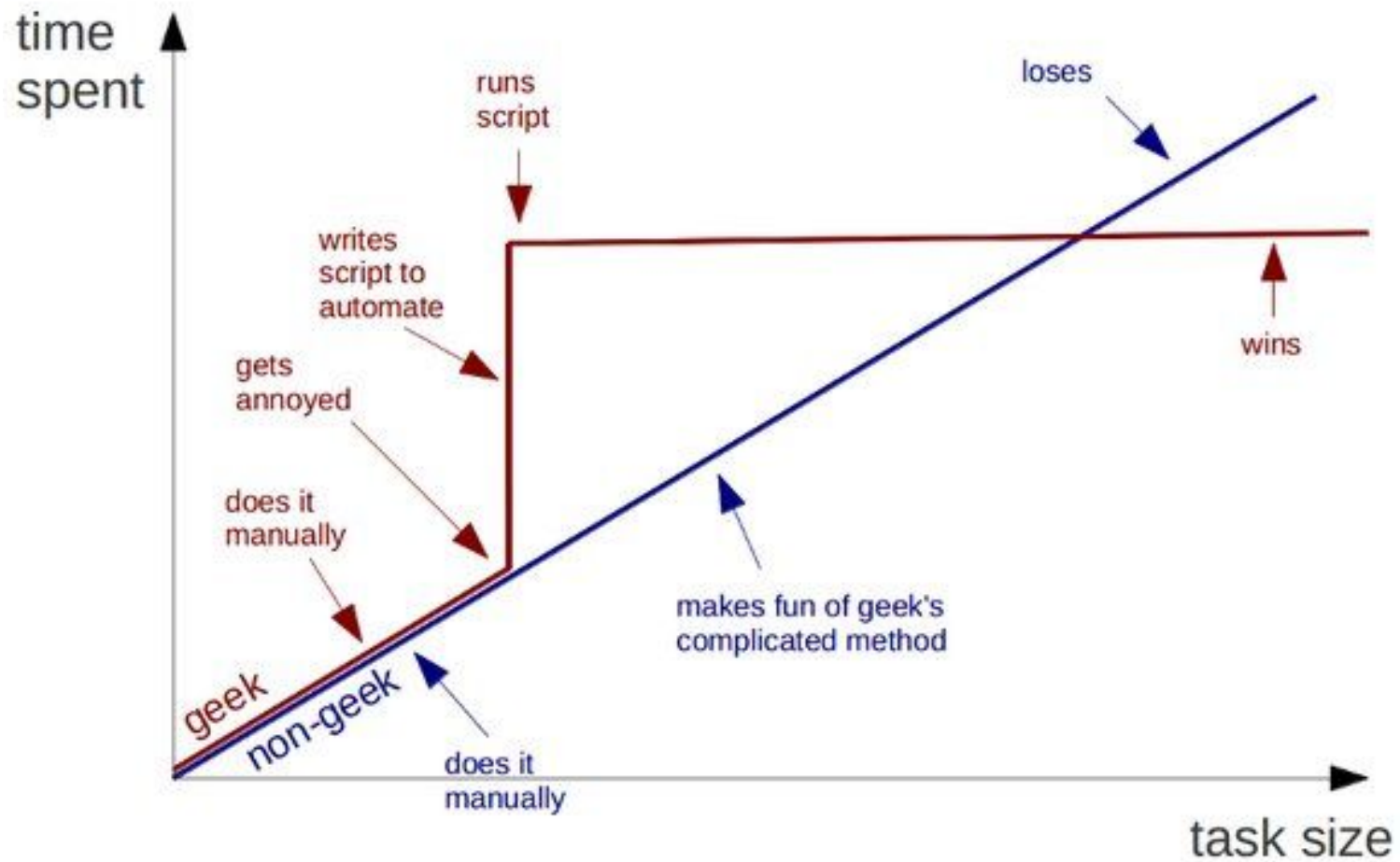


[Alexandre Counis, Les Echos, 5 oct. 2020](#)

Source: R-bloggers

Geeks and repetitive tasks

Geeks and repetitive tasks



R sait tout faire

Lire un tableau de données	<code>read.table()</code>
Fusionner deux tableaux	<code>merge()</code>
Filtrer des lignes	<code>data[data\$x > 10]</code>
Sélectionner des colonnes	<code>data[,c("x", "y")]</code>
Rechercher une chaîne de caractères	<code>grep()</code>
Calculer une moyenne	<code>mean()</code>
Réaliser une ACP	<code>prcomp()</code>
Additionner deux matrices	<code>mat1 + mat2</code>
Exporter un tableau de données	<code>write.table()</code>
Calculer une variance	<code>var()</code>
Régression linéaire	<code>lm()</code>
Tracer une courbe	<code>plot()</code>
Tester une hypothèse	<code>t.test()</code>
Dessiner un histogramme	<code>hist()</code>
Convertir des données	<code>as.matrix()</code>

Modes d'utilisation (liste non exhaustive)



Localement via le terminal



Localement via RStudio (utilisation classique)



Sur un serveur via le terminal et une connexion ssh



Sur un serveur via un navigateur web pour accéder à RStudio server

Fichiers à récupérer sur votre machine

A partir d'un navigateur Web, téléchargez et enregistrez **sur votre ordi** les fichiers de données:

- `expression.txt`: données d'expressions pour 4 échantillons
- `annotation.csv`: informations sur les gènes (id, name, chr, start, stop)

<https://tinyurl.com/r-exprs-txt>

<https://tinyurl.com/r-annot-csv>

Ouverture ou connexion à RStudio

2 alternatives :

1. Ouvrir RStudio sur votre propre ordinateur (si installé)
- 2. Vous connecter au serveur Web R de l'IFB :**

<https://rstudio.cluster.france-bioinformatique.fr>

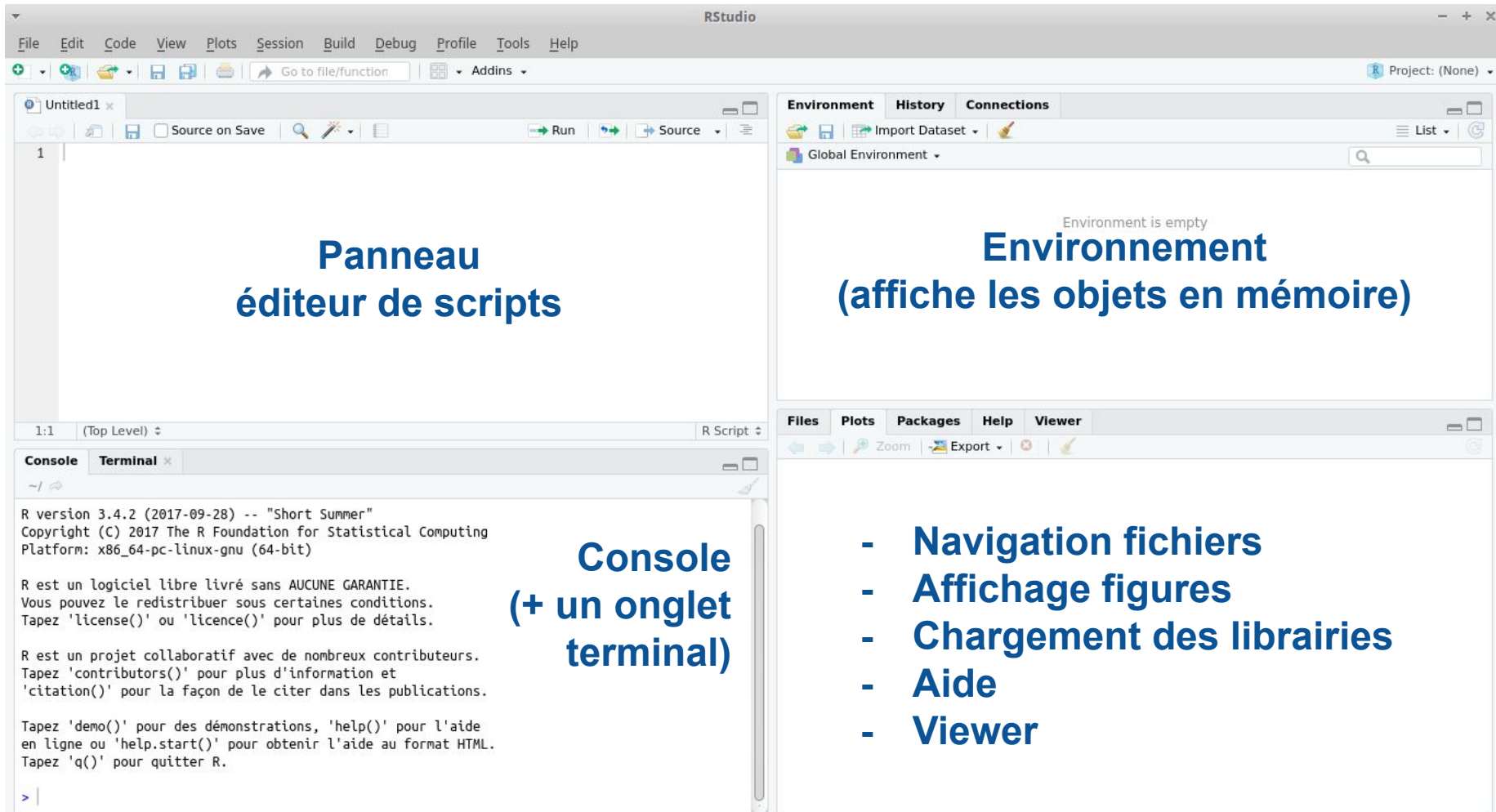


The image shows a sign-in form for RStudio. The form is titled "Sign in to RStudio" and contains the following elements:

- A horizontal line separating the title from the input fields.
- A label "Username:" followed by a text input field.
- A label "Password:" followed by a text input field.
- A checkbox labeled "Stay signed in".
- A blue "Sign In" button at the bottom.

RStudio

- Disponible depuis 2011
- Logiciel facilitant l'utilisation de R via 4 panneaux
- Chaque panneau présente plusieurs onglets (fonctionnalités complémentaires)



R sait tout faire : il compte

`2 + 3`

`4 * 5`

`6 / 4`

`1:10`

`8:-9`

`1,2`

`1.2`

Notion de variable/objet

```
a <- 2      ## Créer une variable nommée a et lui assigner une valeur
print(a)   ## Afficher la valeur de la variable a
a          ## Même résultat: si on évoque le nom de variable, R l'imprime
```

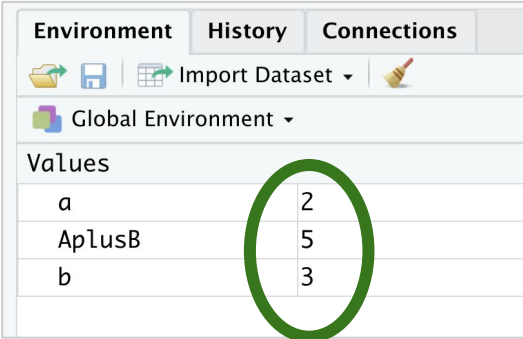
```
b <- 3      ## Assigner une valeur à une seconde variable
AplusB <- a + b ## Effectuer un calcul avec 2 variables
print(AplusB) ## Afficher le contenu de la variable AplusB
```

```
a <- 7      ## Changer la valeur de a
print(AplusB) ## Note: le contenu de AplusB n'est pas modifié
```

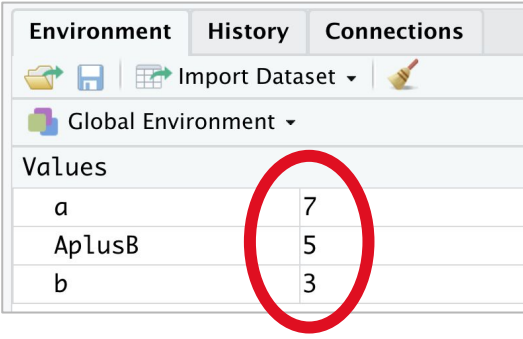
```
AplusB <- a + b ## On recalcule AplusB
print(AplusB)  ## La nouvelle valeur tient compte de la modification de a
```

```
vec1 <- c(1,10) ## Créer un vecteur
vec2 <- 1:10    ## Créer un vecteur contenant une séquence d'entiers de 1 à 10
vec2 + a       ## Somme d'un vecteur et d'un nombre
vec3 <- c("riri", "fifi", "loulou") ## Vecteur de chaînes de caractères
vec3 / b       ## Diviser des chaînes de caractères par un nombre
```

```
## Noms de variables interdits: TRUE, FALSE, T, F, c, t, pi, data, LETTERS, letters, ...
```



Environment	History	Connections
Global Environment	Import Dataset	
Values		
a	2	
AplusB	5	
b	3	



Environment	History	Connections
Global Environment	Import Dataset	
Values		
a	7	
AplusB	5	
b	3	

Cas pratique : données d'expression

Création d'un dossier "intro_R"

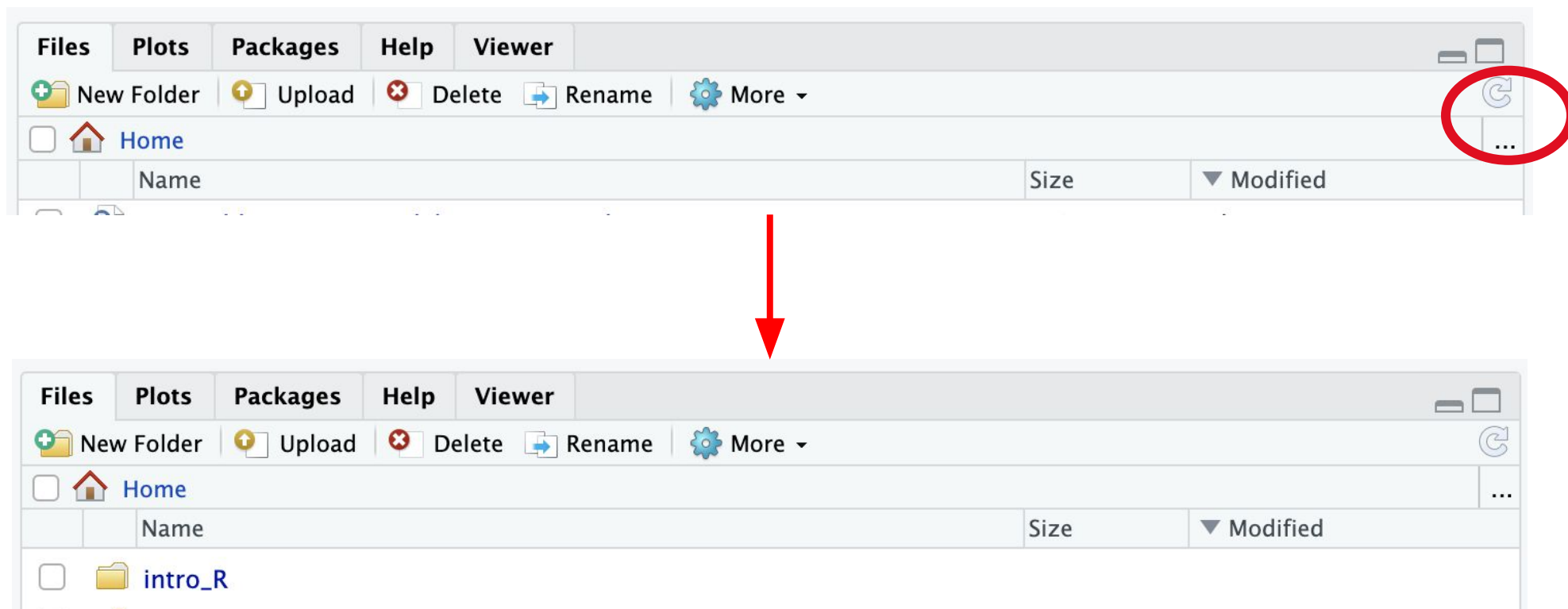
The screenshot shows the RStudio interface with the following components:

- Terminal:** Displays the R version 3.4.4 (2018-03-15) -- "Someone to Lean On" and copyright information. It also shows a message from the ABiMS support team regarding the update from version 3.3.2 to 3.4.4 on 16-04-2017.
- Environment:** Shows the Global Environment, which is currently empty.
- Files:** Shows the file explorer for the directory ~/intro_R. The 'New Folder' button is highlighted with a red circle and a red arrow pointing to it.

This close-up view of the Files pane shows the 'New Folder' button, which is circled in red. The button is located in the top-left corner of the Files pane, next to the 'Upload' button. The current directory is shown as 'Home > intro_R'.

Actualisation du dossier

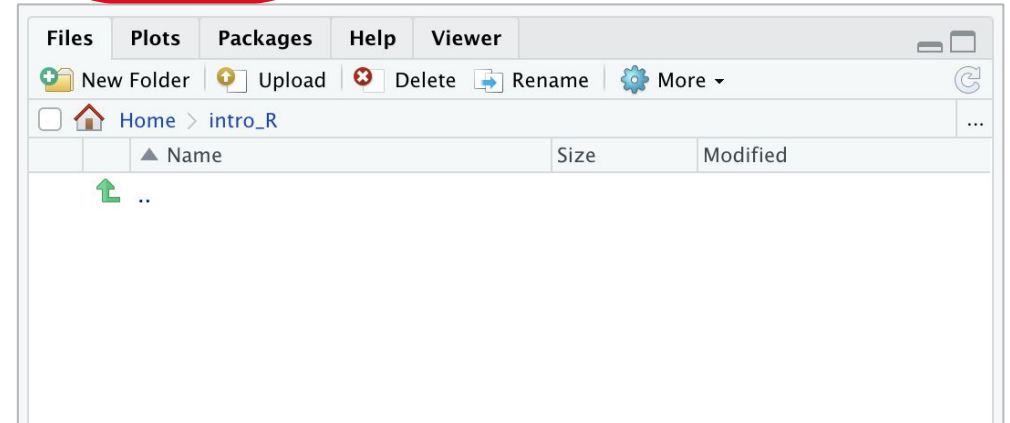
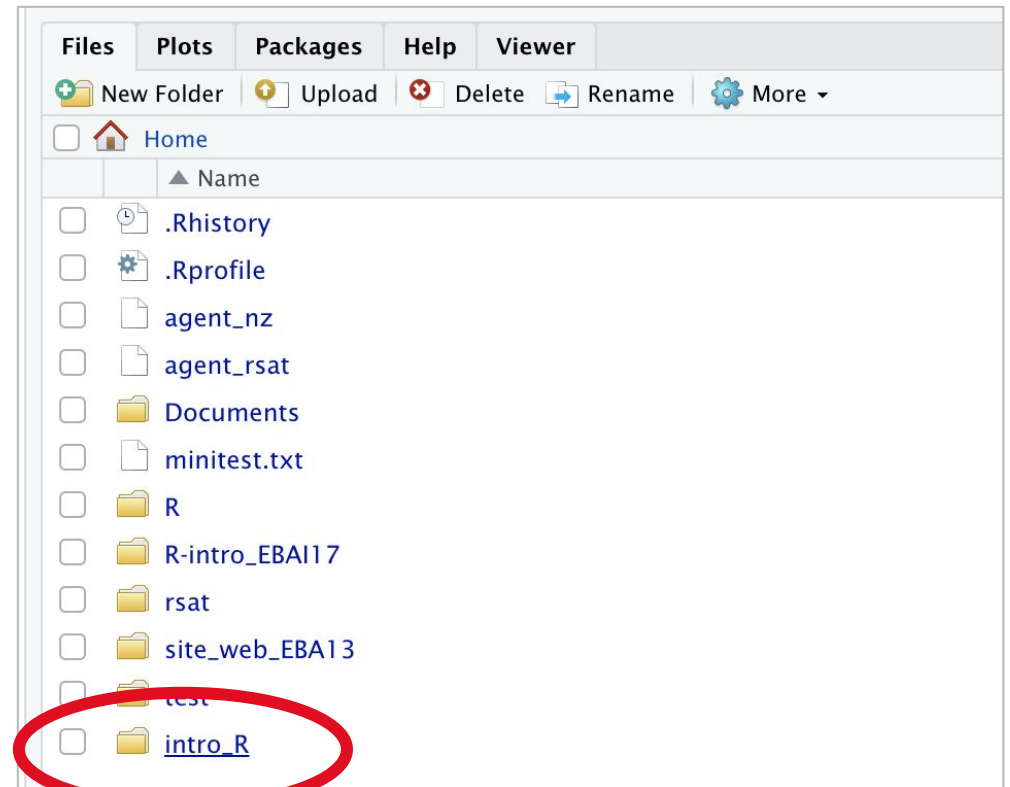
Dans certains cas, il faut actualiser le contenu du dossier pour pouvoir voir le nouveau sous-dossier. Vérifiez ensuite si `intro_R` apparaît bien dans le contenu de votre dossier principal.



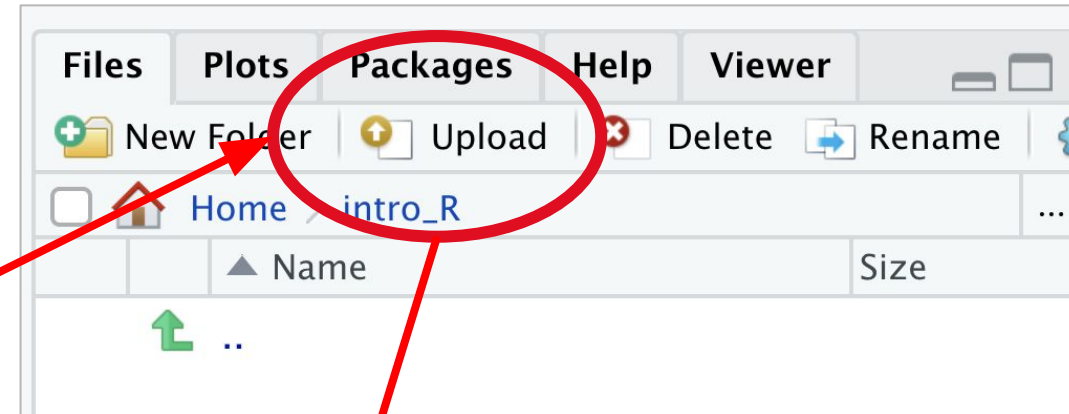
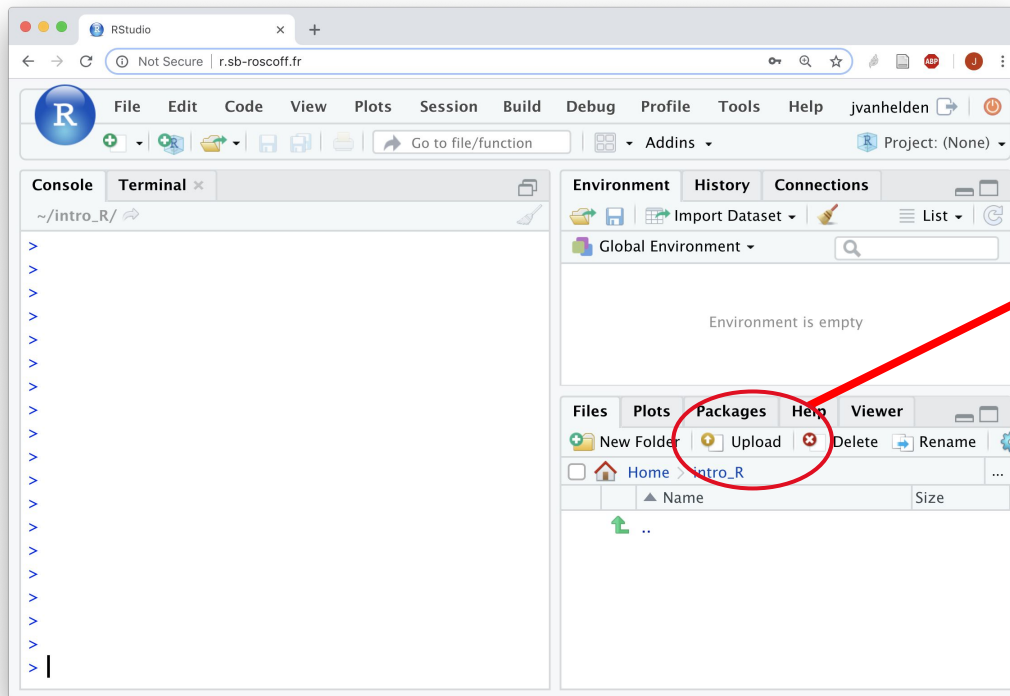
Déplacement dans le dossier “intro_R”

Double-cliquez sur le dossier “intro_R”, pour vous y déplacer.

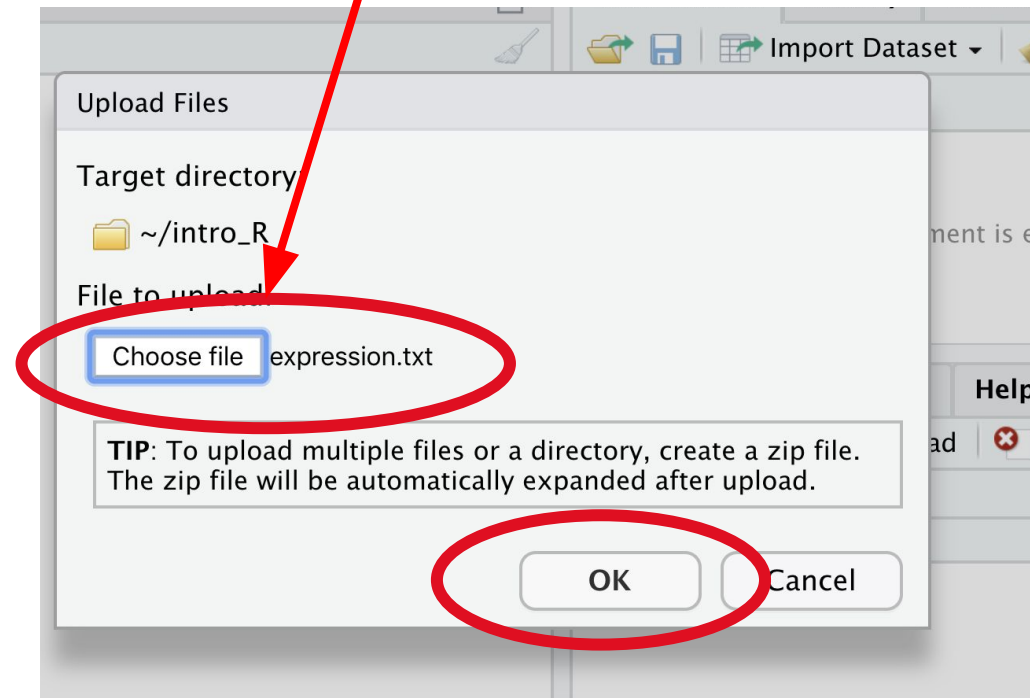
Puisque vous venez de créer le dossier il est vide (image du bas).



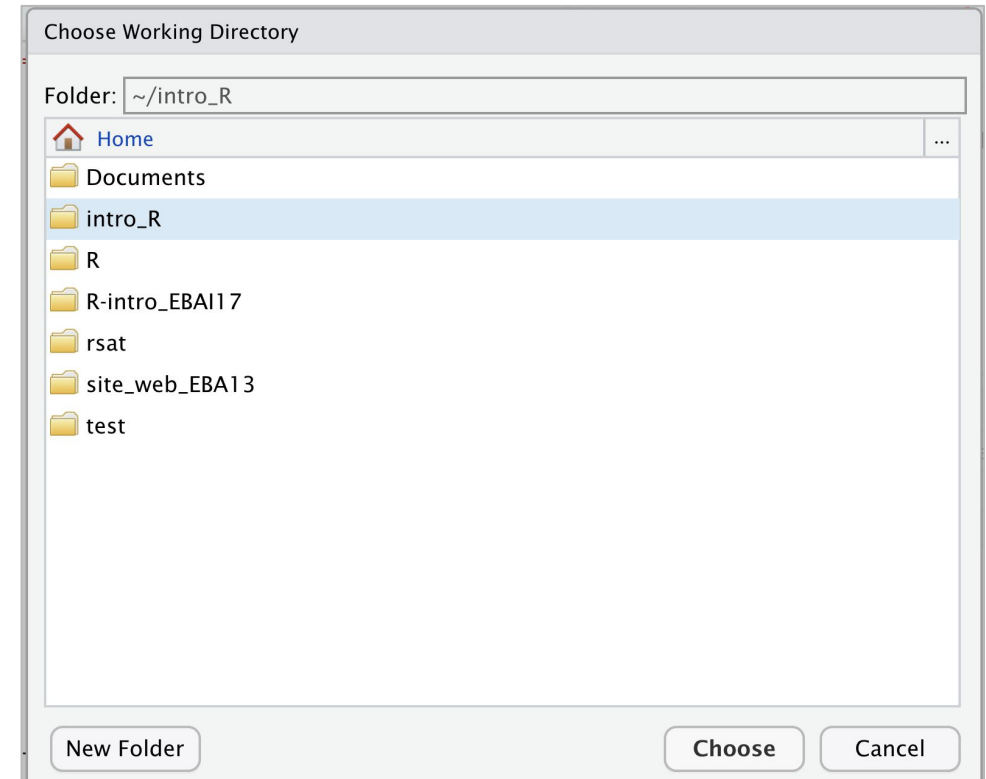
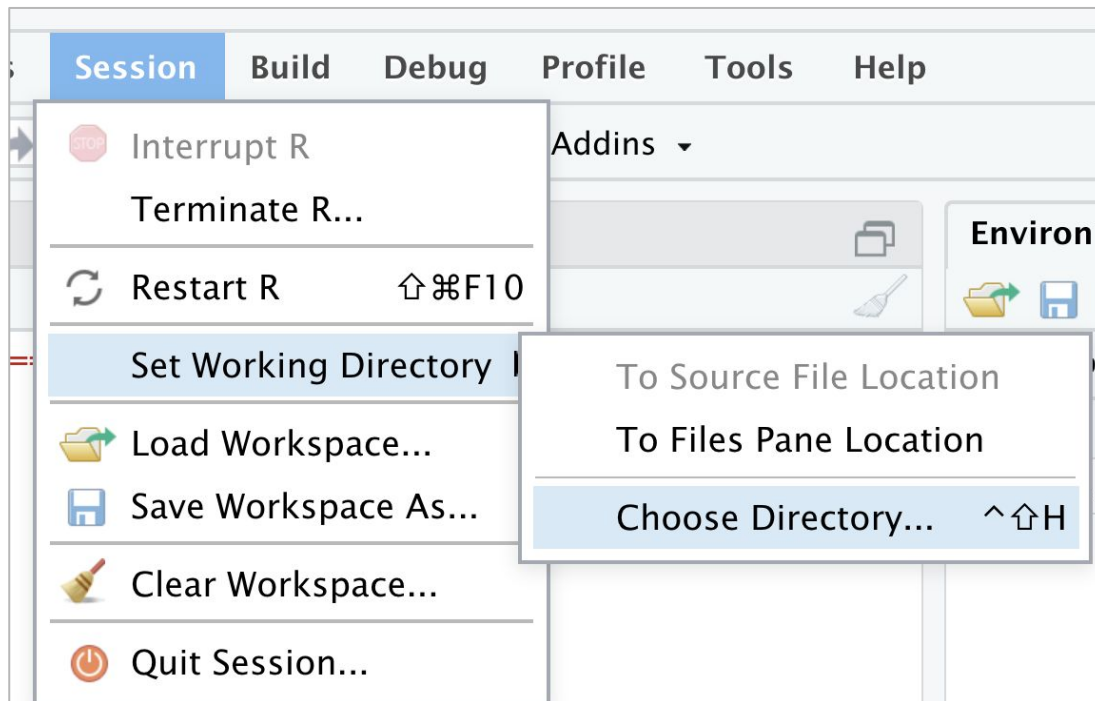
Téléversement (“upload”) des données



Au moyen du bouton “**Upload**”, téléversez les fichiers d’expression et d’annotation depuis votre ordinateur vers votre compte sur le serveur.



Définissez votre dossier espace de travail (workdirectory)



The “bash geek” way (V2, directement de votre home du cluster)

Alternative: dans le terminal du cluster, téléchargez et enregistrez dans votre home les fichiers de données:

- `expression.txt`: données d’expressions pour 4 échantillons
- `annotation.csv`: informations sur les gènes (id, name, chr, start, stop)

Ouvrez un connection ssh

```
ssh [votre_login]@core.cluster.france-bioinformatique.fr
```

Où suis-je ?

```
pwd
```

Créez un répertoire “intro_R”

```
mkdir -p ~/intro_R
```

Déplacez vous dans votre dossier

```
cd ~/intro_R
```

Où suis-je maintenant ?

```
pwd
```

Téléchargez les données

```
srun wget https://tinyurl.com/r-exprs-txt --output-document=expression.txt
```

```
srun wget https://tinyurl.com/r-annot-csv -O annotation.csv
```

Qu’y a-t-il ici ?

```
ls -l
```

The “R geek” way (V3, directement depuis Rstudio)

Définir une variable qui indique le chemin du dossier de travail (working directory).

```
work.dir <- "~/intro_R"
```

Note: R interprète le caractère “~” comme le “HOME” de Linux (cela marche aussi pour Windows!)

S’il n’existe pas encore, créer le dossier de travail.

(Commande Unix équivalente: "mkdir -p ~/intro_R")

```
dir.create(work.dir, recursive = TRUE, showWarnings = FALSE)
```

Où suis-je ? (Commande Unix équivalente: "pwd")

```
getwd()
```

Aller dans ce dossier de travail (Commande Unix équivalente: "cd ~/intro_R")

```
setwd(work.dir)
```

Et maintenant, où suis-je ?

```
getwd()
```

Qu'y a-t-il par ici ? (Commande Unix équivalente: "ls")

```
list.files()
```

```
dir() ## Un autre nom pour la même commande
```

Télécharger un fichier : the “geek” way (V3)

Nous avons montré ci-dessus comment télécharger des fichiers en utilisant l’interface graphique de RStudio.

Alternativement, on peut télécharger des fichiers au moyen de la commande R `download.file`.

Les deux commandes suivantes permettent de télécharger les fichiers utilisés pour les exercices.

```
download.file(url = "https://tinyurl.com/r-exprs-txt", destfile = "expression.txt")
```

```
download.file(url = "https://tinyurl.com/r-annot-csv", destfile = "annotation.csv")
```

Note : équivalent de la commande “`wget`” sous Unix.

Chargement des données (dans la mémoire de R)

Charger le contenu du fichier "expression.txt" dans une variable nommée "exprs".

```
exprs <- read.table(file = "expression.txt", header = TRUE, sep = "\t")
```

Accéder à l'aide d'une fonction

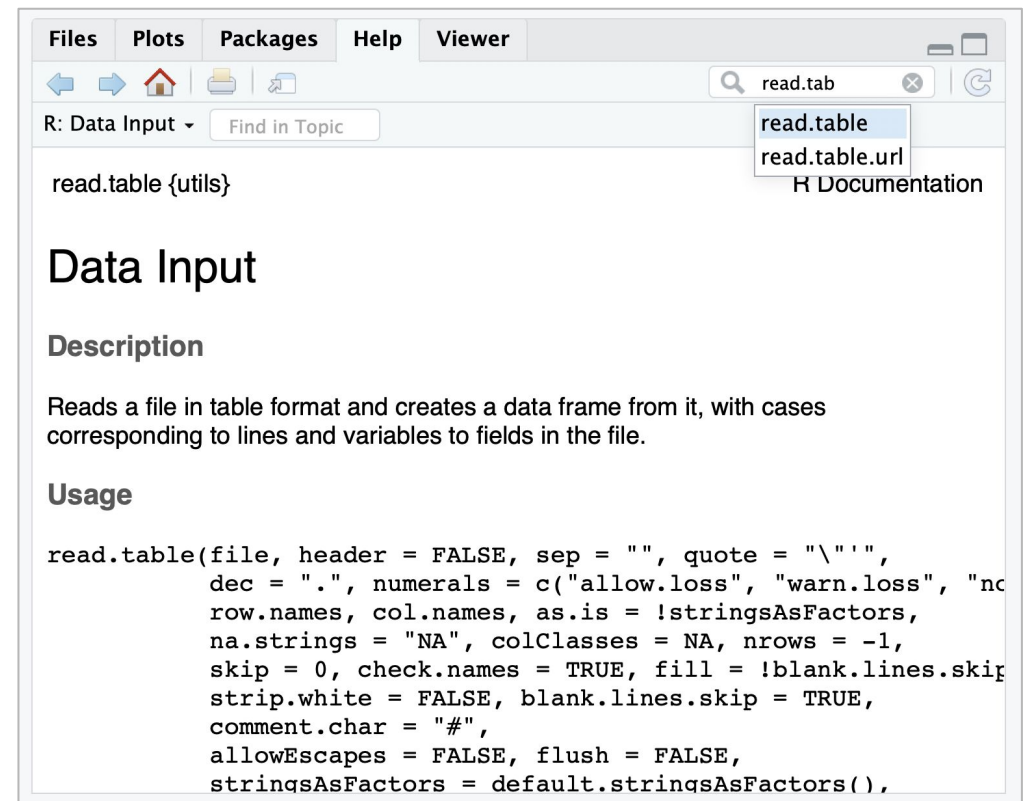
```
help(read.table)
```

Notation alternative

```
?read.table
```

Recherche interactive sous RStudio

- Sélectionner l'onglet "Help" du panneau inférieur droit.
- Taper "read.table" dans la boîte de recherche.



The screenshot shows the RStudio Help window for the `read.table` function. The window title is "R: Data Input" and the search bar contains "read.tab". The search results show "read.table" and "read.table.url" with a link to "R Documentation". The main content area displays the function signature: `read.table {utils}`. Below this, the section "Data Input" is followed by a "Description" which states: "Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file." The "Usage" section shows the full function signature with its arguments: `read.table(file, header = FALSE, sep = "", quote = "\"", dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"), row.names, col.names, as.is = !stringsAsFactors, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "#", allowEscapes = FALSE, flush = FALSE, stringsAsFactors = default.stringsAsFactors(), ...)`

Affichage de l'objet "exprs"

Imprimer toutes les valeurs.

```
print(exprs)
```

Affichage des premières lignes de l'objet

```
head(exprs)
```

Un peu plus de lignes

```
head(exprs, n = 20)
```

Caractéristiques d'un tableau de données

Dimensions

`ncol(exprs)` **## Nombre de colonnes**
`nrow(exprs)` **## Nombre de lignes**
`dim(exprs)` **## Dimensions**

Noms des lignes et colonnes

`colnames(exprs)`
`rownames(exprs)`

Résumé rapide des données par colonne

`summary(exprs)`

`str(exprs)`

Sélection de colonnes d'un tableau

Valeurs stockées dans la colonne nommée "WT1"

```
exprs$WT1
```

Notation alternative

```
exprs[ , "WT1"]
```

Sélection de plusieurs colonnes.

```
exprs[ , c("WT1", "WT2")]
```

Sélection de colonnes par leur indice

```
exprs[ , 2]
```

```
exprs[ , c(3, 2)]
```

Histogrammes

Histogramme des valeurs d'expression pour l'échantillon WT1

```
hist(exprs$WT1)
```

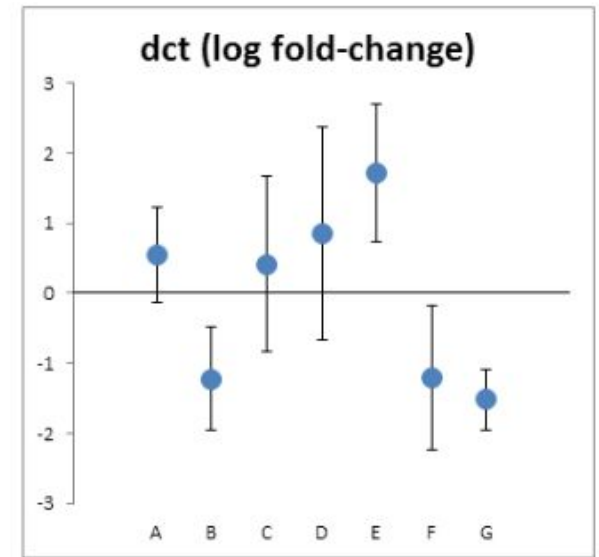
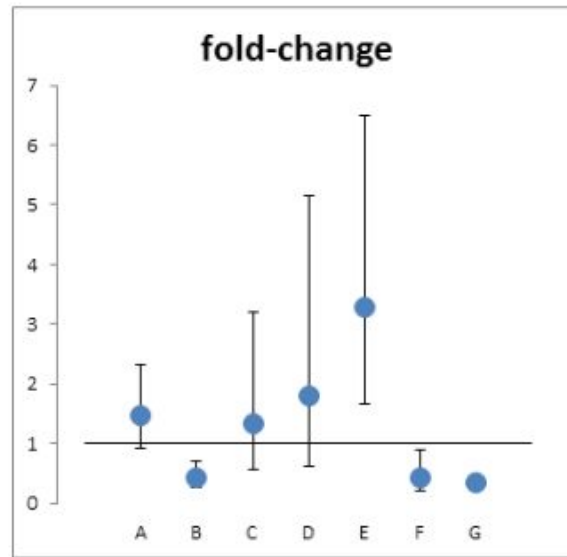
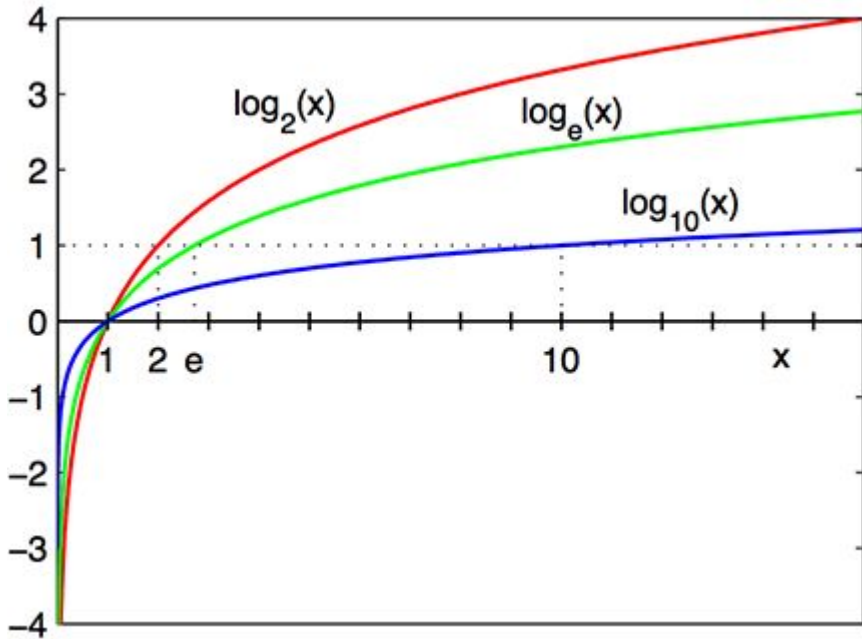
Histogramme du logarithme de ces valeurs

```
hist(log(exprs$WT1))
```

```
hist(log2(exprs$WT1))
```

```
hist(log10(exprs$WT1))
```

Logarithmes (pour rappel...)



Boxplots

Boite à moustache des valeurs d'expression pour l'échantillon WT1

```
boxplot(exprs$WT1)
```

Boite à moustache du logarithme de ces valeurs

```
boxplot(log(exprs$WT1))
```

```
boxplot(log2(exprs$WT1))
```

```
boxplot(log10(exprs$WT1))
```

Boite à moustache des valeurs d'expression pour les 4 échantillons

```
boxplot(exprs)
```

```
boxplot(exprs[,-1])
```

```
boxplot(log2(exprs[,-1]))
```

```
boxplot(exprs[,-1], log = "y")
```

Nuage de points

Expressions KO1 vs WT1

```
plot(x = log(exprs$WT1), y = log(exprs$KO1))
```

Personnalisation des paramètres graphiques

```
plot(x = log(exprs$WT1), ## données pour l'abscisse  
     y = log(exprs$KO1), ## données pour l'ordonnée  
     main = "Expression KO1 vs WT1", ## Titre principal  
     xlab = "WT1",      ## légende de l'axe X  
     ylab = "KO1",      ## légende de l'axe Y  
     pch = 16,          ## caractère pour marquer les points  
     las = 1,           ## écrire les échelles horizontalement  
     col = "red")       ## couleur des points  
grid()                 ## Ajout d'une grille  
abline(a = 0, b = 1)   ## Ajouter la droite X = Y (intercept a = 0, pente b = 1).
```


Sélection de lignes d'un tableau

Sélection des lignes 4 et 11 du tableau des expressions

```
exprs[c(4, 11), ]
```

Indices des lignes correspondant aux IDs ENSG00000253991 et ENSG00000099958

```
mygenes <- c("ENSG00000253991", "ENSG00000099958")
```

```
exprs$id %in% mygenes
```

```
which(exprs$id %in% mygenes)
```

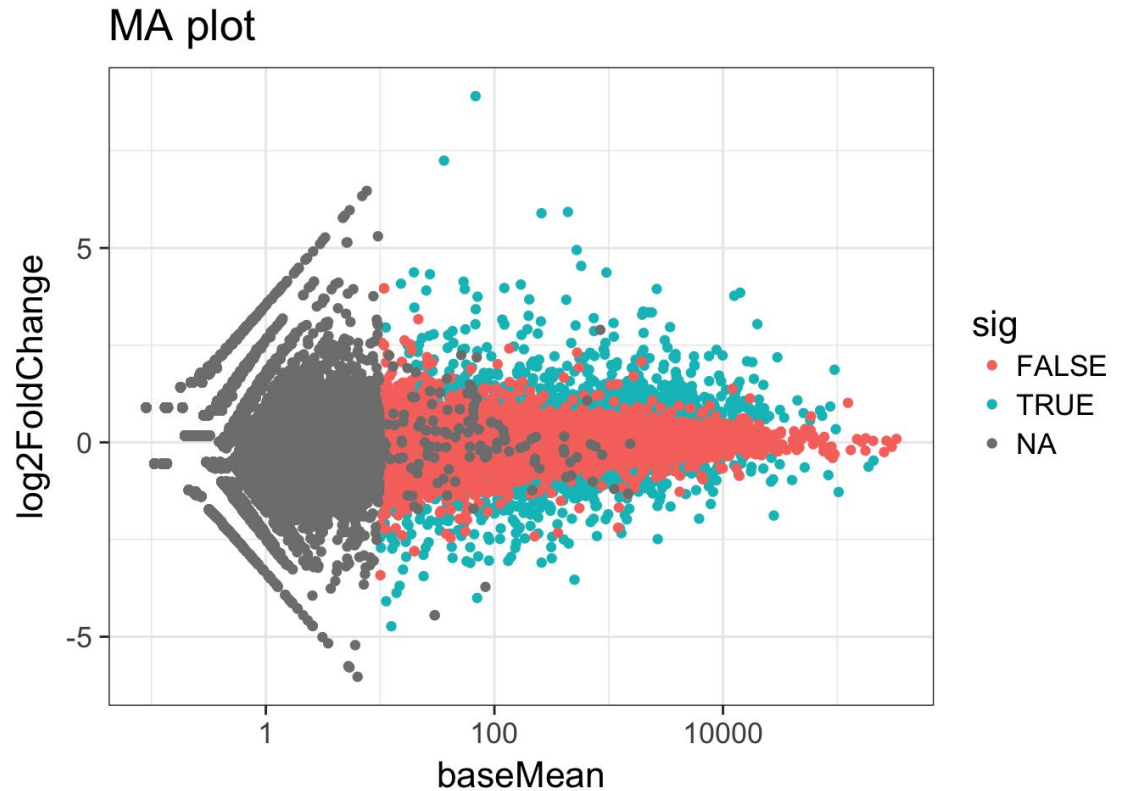
Afficher les lignes correspondantes

```
exprs[which(exprs$id %in% mygenes), ]
```

Analyse d'expression différentielle : MA-plot

```
> head(exprs, 10)
```

	id	WT1	WT2	K01	K02
1	ENSG00000034510	235960	94264	202381	91336
2	ENSG00000064201	116	71	64	56
3	ENSG00000065717	118	174	124	182
4	ENSG00000099958	450	655	301	472
5	ENSG00000104164	4736	5019	4845	4934
6	ENSG00000104783	9002	8623	7720	7142
7	ENSG00000105229	1295	2744	1113	2887
8	ENSG00000105723	3353	7449	3589	7202
9	ENSG00000116199	2044	4525	2604	4902
10	ENSG00000118939	7022	2526	6269	3068



Calculs sur les colonnes

Calcul de moyennes par ligne (`rowMeans``) pour un sous-ensemble donné des colonnes (WT1 et WT2).

```
rowMeans(exprs[ , c("WT1","WT2")])
```

Ajout de colonnes avec les expressions moyennes des WT et des KO

```
exprs$meanWT <- rowMeans(exprs[ , c("WT1","WT2")])  
exprs$meanKO <- rowMeans(exprs[ , c("KO1","KO2")])  
head(exprs) ## Check the result
```

Fold-change KO vs WT

```
exprs$FC <- exprs$meanKO / exprs$meanWT  
head(exprs) ## Check the result
```

Moyenne de tous les échantillons

```
exprs$mean <- rowMeans(exprs[ , c("WT1", "WT2", "KO1", "KO2")])
```

MA-plot : \log_2FC vs intensité

Le MA plot représente le lien entre différence d'expression et intensité moyenne.

M (magnitude) est le logarithme en base 2 du rapport d'expression ("log2 fold-change")

$$M = \log_2(FC) = \log_2(KO/WT) = \log_2(KO) - \log_2(WT)$$

```
exprs$M <- log2(exprs$FC)
```

A (average intensity) est la moyenne des logarithmes des valeurs d'expression.

$$A = \log_2(\text{moyenne d'échantillons})$$

```
exprs$A <- log2(exprs$mean)
```

On peut ensuite dessiner un nuage de points (en l'agrémentant un peu)

```
plot(x = exprs$A, y = exprs$M, main = "MA plot",  
     col = "blue", pch = 16, xlab = "A = intensity", ylab = "M = log2FC")  
grid(lty = "solid", col = "lightgray")  
abline(h = 0)
```

Appliquer une fonction sur les lignes/colonnes

Appliquer une fonction (moyenne, variance, ...) sur chaque **ligne** d'un tableau

```
mean.per.row <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 1, mean)
```

```
mean.per.row <- apply(exprs[ , c(2, 3, 4, 5)], 1, mean)
```

```
mean.per.row <- apply(exprs[ , -1 ], 1, mean)
```

```
mean.per.row <- apply(exprs[ , which(sapply(exprs, class) != "factor")], 1, mean)
```

```
var.per.row <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 1, var)
```

Appliquer une fonction (moyenne, variance, ...) sur chaque **colonne** d'un tableau

```
mean.per.col <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 2, mean)
```

```
var.per.col <- apply(exprs[ , c("WT1", "WT2", "KO1", "KO2")], 2, var)
```

Charger les annotations des gènes

```
read.table(file = "annotation.csv")  
read.table(file = "annotation.csv", sep = ";")  
read.table(file = "annotation.csv", sep = ";", header = TRUE)
```

```
annot <- read.table(file = "annotation.csv", sep = ";", header = TRUE)  
dim(annot)           ## Vérifier les dimensions  
head(annot)          ## Afficher quelques lignes
```

Combien de gènes par chromosome ?

```
table(annot$chr)
```

Question : combien de gènes sur le chromosome 8 ? Et sur le X ?

```
barplot(sort(table(annot$chr)), horiz = TRUE, las = 1,  
         col = "lightblue", xlab = "Number of genes")
```

Ma première bioinformatique intégrative

- 1ere étape : fusionner les tableaux d'expressions et d'annotations :

?merge

```
exprs.annot <- merge(exprs, annot, by = "id")  
head(exprs.annot)
```

- 2eme étape : sous-ensemble des lignes pour lesquelles chr vaut 8 :

```
exprs8 <- exprs.annot[which(exprs.annot$chr == 8), ]  
print(exprs8)
```

- Exporter exprs8 dans un fichier :

```
write.table(x = exprs8, file = "exprs8.txt", sep = "\t", row.names = FALSE, col.names = TRUE)
```

Take home messages

- Tout est faisable avec R
- **Définir et comprendre l'opération mathématique/statistique** avant de chercher la fonction R correspondante
- R est un langage :
 - plusieurs types et structures de données (out of scope)
 - énormément de commandes à découvrir (out of scope)
 - Google est votre ami
- Une infinité de :
 - ressources en ligne
 - tutoriels pour des analyses spécifiques (e.g. DESeq2 pour le RNA-Seq)
- Bonnes pratiques : <https://google.github.io/styleguide/Rguide.xml>

Serveur RStudio

<https://rstudio.cluster.france-bioinformatique.fr/>



Jupyter lab (inclut un serveur RStudio et plein d'autres choses)

<http://jupyterhub.cluster.france-bioinformatique.fr/>

NEW



Une question ? Un besoin ? Un problème ? **Contactez la communauté IFB**

<https://community.france-bioinformatique.fr/>



Ressources

Base R Cheat Sheet

Getting Help

Advanced R Cheat Sheet

Environments

RStudio IDE :: CHEAT SHEET

R Markdown :: CHEAT SHEET

What is R Markdown?

Workflow

render

Embed code with knitr syntax

Interactive Documents

Shiny

RStudio is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at rmarkdown.rstudio.com • rmarkdown 1.6 • Updated: 2016-02

R

<https://www.r-project.org/>

RStudio

<https://rstudio.com/>

R-bloggers

<https://www.r-bloggers.com/>

THINKR

<https://thinkr.fr/>

Rstudio Cheatsheets (un tas de thèmes):

<https://rstudio.com/resources/cheatsheets/>

