CH4 bioinfo fct :

**read.csv(“*c:/path*”)** : lis le document

Wrapper function de read.table, avec les arg deja faits, ici le separateur est la virgule

**View(*dataframe)*** : fait un tableau reprenant la df

**str(*dataframe*)** : vient de **str**ucture, donne le type d’argument (num, chr, etc.) par colone

**dim(*dataframe*)**: donne la dimention du df:

Si on a 250 ligne et 20 colones alors dans la console : *>250 20*

**nrow(*dataframe*)** et **ncol(*dataframe*)** : donne le nombre de ligne et de colone respectivement

**head(*dataframe*)** et **tail(*dataframe*)** : donne les 6 premiers/dernieres lignes

**names(*dataframe*)** : donne le nom des colones <-> **colnames(*dataframe*)**

**rownames(*dataframe*)** : donne nom de lignes

**sum(*dataframe*)** : donne un resumé de chaque colone (nom, taille, type de char, si possible mode, moyenne etc.)

*df*[x,y] : x signifie le numero de la ligne et y de la colone

si il y a *df*[x1:x2, y] alors c’est de la ligne x1 a la ligne x2

si il y a *df*[ ,y] ou *df*[x, ] alors c’est toutes les lignes de y ou toutes les colones de x

si -x ou -y on exclus celui la : *df*[-1,] on prends toutes les colones et toutes les lignes sauf la premiere

pour choisir une colones tu peux utiliser le nom (ici on prends gene comme eg):

*df*[“gene”]

*df*[, “gene”]

*df*$gene

exporter le tout

write.csv(rna, file = "data\_output/my\_rnaseq.csv")

Factors:

Des arg charactere traité comme des integers

sex <- factor(c("male", "female", "female", "male", "female"))

R va donner des level (numero) a chaque variable

female va etre level 1 et male level 2 car f arrive avant m

levels(*var*) : donne le nom de chaque level

nlevels(*var*) : donne le nombre de level

pour reordoner les niveau (par eg male avant female):

sex <- factor(sex, **levels = c("male", "female")**)

changer en character:

as.character(sex)

(en num: as.nuleric(sex))

Renomer male en M et female en F:

levels(sex) <- **c("M", "F")**

Matrices:

Toutes les cellule doivent etre du meme type de var

m <- matrix(1:9, ncol = 3, nrow = 3)

m <- matrix(1:9, ncol = 3, nrow = 3, **byrow = TRUE**)

Signifie que au lieu de d’etre trié le long des colones c’est trie le long des lignes (regarde dans rstudio)

rnorm(n) : te donne n nombre selon une distribution, l’equart-type et la Moyenne peut etre change en utilisant sd = et mean =

paste() : concantenate strings together, use the separator=”/” to change what separates, if nothing then itll be space

dates: use lubridate from tidyverse

ymd(): YYYY-MM-DD

dmy(): DD-MM-YYYY

**list**: one dimension, every item can be of a different data type.

Pour une liste l:

l <- list(1:10, ## numeric

letters, ## character

installed.packages(), ## a matrix

cars, ## a data.frame

list(1, 2, 3)) ## a list

premiere ligne est de 1 a 10

2e est les lettres de l’alphabet

3e est le package installé (en matrice)

4 is a df

5 is a list (in a list)

To subset:

L[[1]] prends la premiere ligne de la premiere liste

L[1:2] prends le 2 premiere liste

GROUP\_BY()

1 rna %>%

2 group\_by(gene, infection, time) %>%

3 summarize(mean\_expression = mean(expression),

4 count = n()) %>%

5 print(n = 15)

1: pipe RNA

2: group par gene infection et temps 🡪 on aura des group du meme gene avec la mm infection au meme moment

3: donne l’expression Moyenne de chaque groupe

4: donne le nombre d’element par groupe

gene infection time mean\_expression count

*<chr>* *<chr>* *<dbl>* *<dbl>* *<int>*

1 AI504432 InfluenzaA 4 1104. 8

2 AI504432 InfluenzaA 8 1014 7

3 AI504432 NonInfected 0 1034. 7

Il y a 8 elements qui ont comme gene AI504432, infecté par l’influenza au temps 4 et la Moyenne d’expresion du groupe est de 1104