Traitement de données FTIR et visualisation des données

Philippe STOCKER

2022-05-31

Preprocessing des données bruts : Bash et Python

Première étape, ranger les csv dans des dossier et attention partuculière aux nomx des fichiers sans espaces et caractères exotiques

Préconisation la nomenclature des fichier : **group__modality.csv**

Ici, on aura la nomenclature suivante :

```
--> processing
    --> ctrl_algue.csv
    --> ctrl_iris.csv
    --> ctrl_prox.csv
    --> pla_algue.csv
    --> pla_iris.csv
    --> pla_prox.csv
    --> outputs
    ---- --> output_ctrl_algue.csv
    ---- --> output_ctrl_iris.csv
    ---- --> output_ctrl_prox.csv
    ---- --> output_pla_algue.csv
    ---- --> output_pla_iris.csv
    ---- --> output_pla_prox.csv
```

Pour les fichiers bruts rangés dans les bons dossiers

Ranger comme suit :

```
--> processing
    --> ctrl
    ---- --> algue
    ---- --> iris
    ---- --> prox
    --> pla
    ---- --> algue
    ---- --> iris
    ---- --> prox
```

```bash
## Supprimer les deux premières lignes
for f in *.csv
do
    sed -i '1,2d' $f
done

## Fusionner les fichiers dans leurs dossiers respectifs
cat *csv name_files > group_modality.csv
```

```
with open('ctrl_algue.csv', 'r') as istr:
    with open('output_ctrl_algue.csv', 'w') as ostr:
        for line in istr:
            line = line.rstrip('\n') + ',control,algue'
            print(line, file=ostr)

with open('ctrl_iris.csv', 'r') as istr:
    with open('output_ctrl_iris.csv', 'w') as ostr:
        for line in istr:
            line = line.rstrip('\n') + ',control,iris'
            print(line, file=ostr)

with open('ctrl_prox.csv', 'r') as istr:
    with open('output_ctrl_prox.csv', 'w') as ostr:
        for line in istr:
            line = line.rstrip('\n') + ',control,prox'
            print(line, file=ostr)

with open('pla_algue.csv', 'r') as istr:
    with open('output_pla_algue.csv', 'w') as ostr:
        for line in istr:
            line = line.rstrip('\n') + ',pla,algue'
            print(line, file=ostr)

with open('pla_iris.csv', 'r') as istr:
    with open('output_pla_iris.csv', 'w') as ostr:
        for line in istr:
            line = line.rstrip('\n') + ',pla,iris'
            print(line, file=ostr)

with open('pla_prox.csv', 'r') as istr:
    with open('output_pla_prox.csv', 'w') as ostr:
        for line in istr:
            line = line.rstrip('\n') + ',pla,prox'
            print(line, file=ostr)
```

```
### à l'issue écrire dans la première ligne du dataset
for f in *.csv
do
    sed -i "1i\time,lambda,measure" $f
done
```

Import des librairies

```
library(ggplot2)
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag

## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

Import des datasets

```
dataset <- read.csv("dataset.csv")
df <- dataset
head(dataset)
```

```
##   wavenumber value   group modality
## 1       4000 92.09 control    algue
## 2       3999 92.09 control    algue
## 3       3998 92.10 control    algue
## 4       3997 92.09 control    algue
## 5       3996 92.08 control    algue
## 6       3995 92.06 control    algue
```

```
df$wavenumber <- as.factor(df$wavenumber)
head(df)
```

```
##   wavenumber value   group modality
## 1       4000 92.09 control    algue
## 2       3999 92.09 control    algue
## 3       3998 92.10 control    algue
## 4       3997 92.09 control    algue
## 5       3996 92.08 control    algue
## 6       3995 92.06 control    algue
```

```
## As Dataframe, because, RTFM
ftr <- data.frame(df)
head(ftr)
```

```
##   wavenumber value   group modality
## 1       4000 92.09 control    algue
## 2       3999 92.09 control    algue
## 3       3998 92.10 control    algue
## 4       3997 92.09 control    algue
## 5       3996 92.08 control    algue
## 6       3995 92.06 control    algue
```

Grouper l'échantillonnage

```
## moyenne par groupe des 5 mesures
grp <- group_by(ftr, group, modality, wavenumber)
mean <- summarise(grp, m = mean(value))
```

```
## `summarise()` has grouped output by 'group', 'modality'. You can override using
## the `.groups` argument.
```

Convertir en absorbance et Normaliser les données

```
mean$m <- (1/mean$m)
head(mean)
```

```
## # A tibble: 6 x 4
## # Groups:   group, modality [1]
##   group   modality wavenumber      m
##   <chr>   <chr>    <fct>       <dbl>
## 1 control algue    600        0.0142
## 2 control algue    601        0.0141
## 3 control algue    602        0.0141
## 4 control algue    603        0.0141
## 5 control algue    604        0.0141
## 6 control algue    605        0.0141
```

```
ftr <- mean
head
```

```
## function (x, ...)
## UseMethod("head")
## <bytecode: 0x563217dbe648>
## <environment: namespace:utils>
```

```
## Normalisation des données

# Define Min-Max normalization function
min_max_norm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

# Apply Min-Max
nrm <- as.data.frame(lapply(ftr[4], min_max_norm))
head(nrm)
```

```
##           m
## 1 0.1143503
## 2 0.1135399
## 3 0.1127922
## 4 0.1122607
## 5 0.1119909
## 6 0.1119332
```

```
group <- ftr$group
modality <- ftr$modality
wavenumber <- ftr$wavenumber
measure <- nrm$m

ftr <- data.frame(group, modality, wavenumber, measure)
ftr$wavenumber <- as.factor(ftr$wavenumber)

head(ftr)
```

```
##     group modality wavenumber   measure
## 1 control    algue        600 0.1143503
```

```
## 2 control    algue        601 0.1135399
## 3 control    algue        602 0.1127922
## 4 control    algue        603 0.1122607
## 5 control    algue        604 0.1119909
## 6 control    algue        605 0.1119332
```

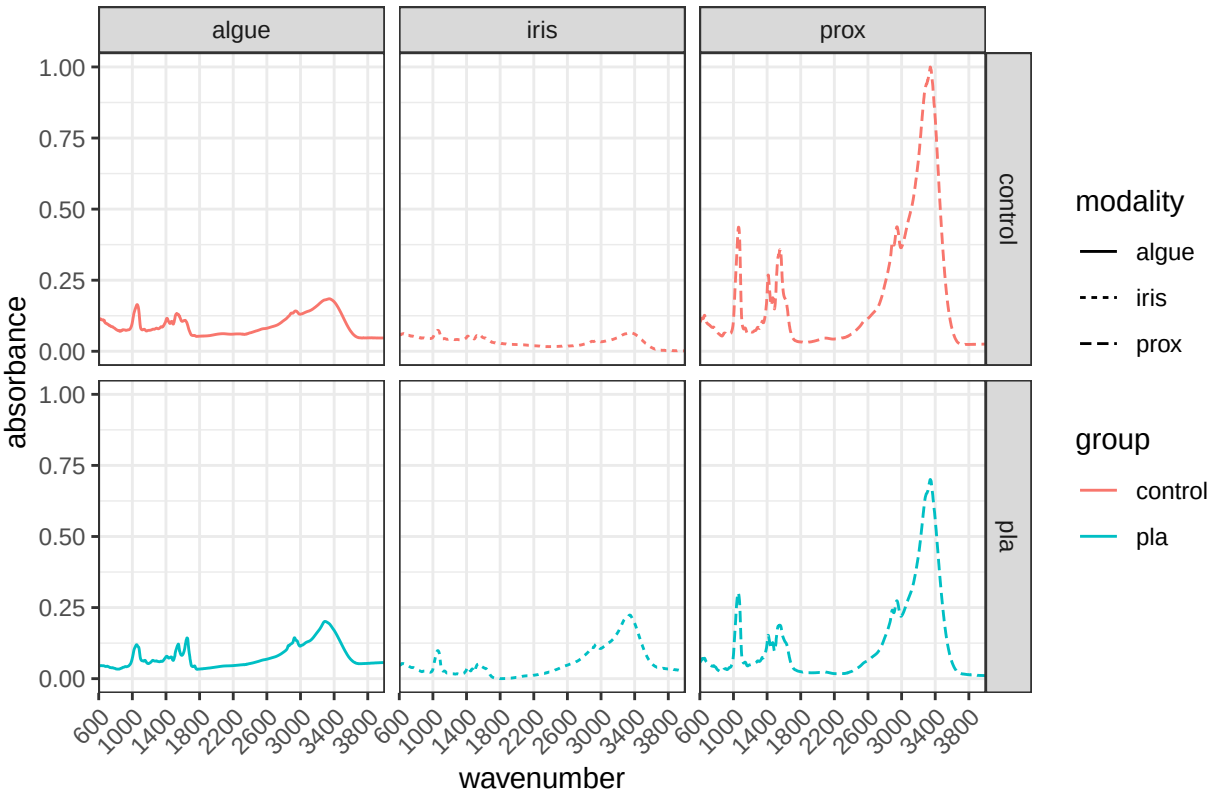
```
### Fin normalization
```

Visualisation des données

De 3 manières différentes, en séparant les groupes par pannels selon groupes et modalités.

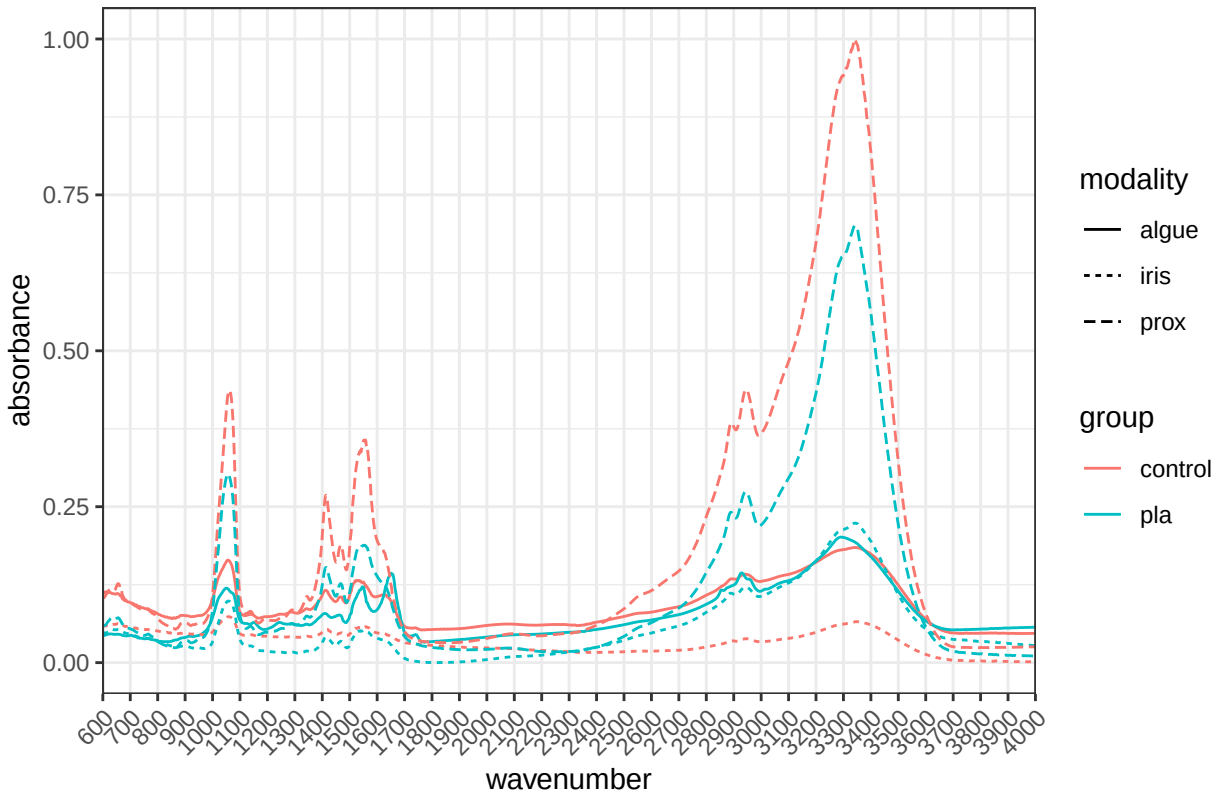
Premier Graphique : IR absorbance spectrums by group and modality

```
## Longueurs d'ondes d'intéret (tracer les Asymptotes)
#vertical.lines <- c(700, 1000, 3000)
## mettre dans le vecteur les longueurs d'ondes d'intéret.

ggplot(ftr, aes(x=wavenumber, y=measure, colour=group, group=interaction(group, modality))) +
  #geom_vline(xintercept = vertical.lines ,linetype = "dashed", color = "red", size=0.5) +
  geom_line(aes(linetype = modality)) + scale_x_discrete(breaks=seq(600, 4000, 400)) +
  labs(title = '', x = 'wavenumber', y = 'absorbance') +
  theme_bw() +
  theme(axis.text.x = element_text(angle=45, hjust = 1)) +
  facet_grid(group ~ modality)
```



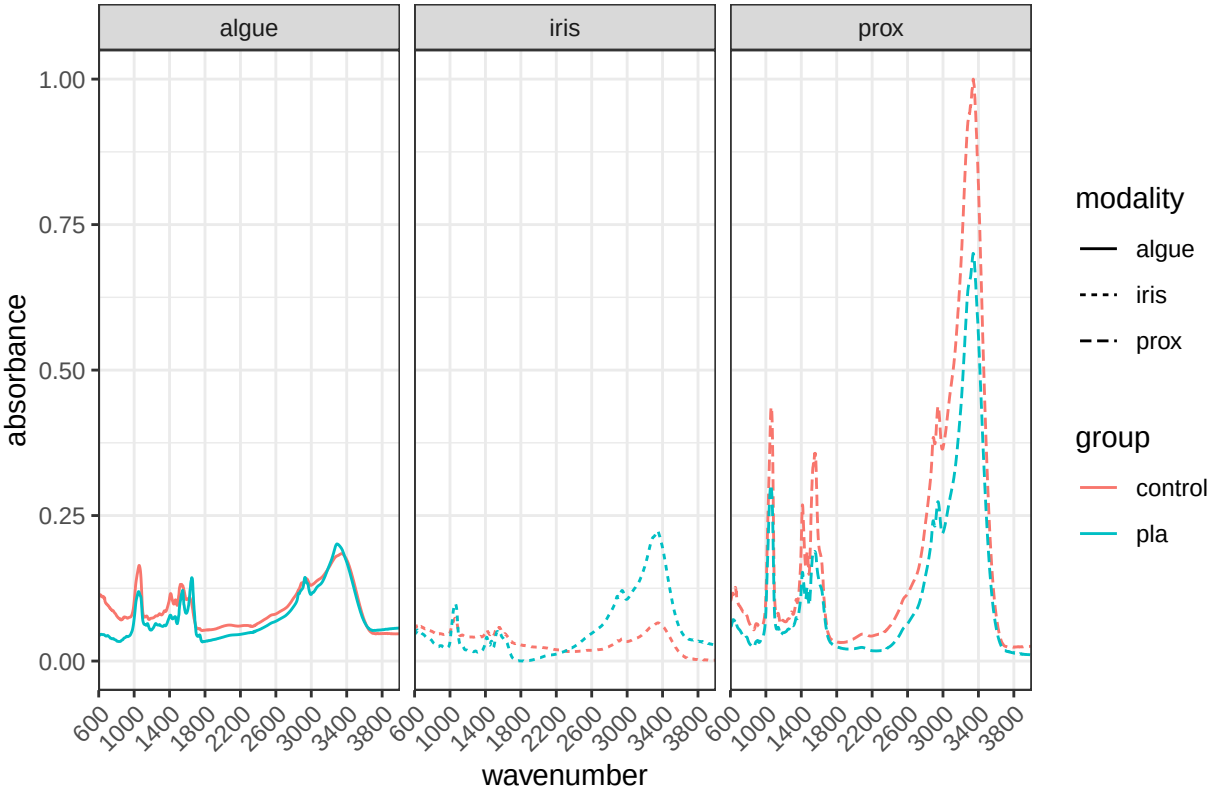
Deuxième Graphique : IR absorbance spectrums

```
## Longueurs d'ondes d'intéret (tracer les Asymptotes)
#vertical.lines <- c(700, 1000, 3000)
## mettre dans le vecteur les longueurs d'ondes d'intéret.

ggplot(ftr, aes(x=wavenumber, y=measure, colour=group, group=interaction(group, modality))) +
  #geom_vline(xintercept = vertical.lines ,linetype = "dashed", color = "red", size=0.5) +
  geom_line(aes(linetype = modality)) + scale_x_discrete(breaks=seq(600, 4000, 100)) +
  labs(title = '', x = 'wavenumber', y = 'absorbance') +
  theme_bw() +
  theme(axis.text.x = element_text(angle=45, hjust = 1))
```



Troisième Graphique : IR absorbance spectrums by groups

```
## Longueurs d'ondes d'intéret (tracer les Asymptotes)
#vertical.lines <- c(700, 1000, 3000)
## mettre dans le vecteur les longueurs d'ondes d'intéret.

ggplot(ftr, aes(x=wavenumber, y=measure, colour=group, group=interaction(group, modality))) +
  #geom_vline(xintercept = vertical.lines ,linetype = "dashed", color = "red", size=0.5) +
  geom_line(aes(linetype = modality)) + scale_x_discrete(breaks=seq(600, 4000, 400)) +
  labs(title = '', x = 'wavenumber', y = 'absorbance') +
  theme_bw() +
  theme(axis.text.x = element_text(angle=45, hjust = 1)) +
  facet_wrap(modality ~ .)
```


algue
iris
prox
absorbance
1.00
0.75
0.50
0.25
0.00
600
1000
1400
1800
2200
2600
3000
3400
3800
wavenumber
modality
algue
iris
prox
group
control
pla