Screening of pharamceuticals in wastewater samples from direct injection-HRMS single spectra files

Ingus Perkons

July 15, 2020

Load packages

```
require(readxl)

## Loading required package: readxl

require(dplyr,warn.conflicts = FALSE)

## Loading required package: dplyr

require(ggplot2)

## Loading required package: ggplot2

require(stringr)

## Loading required package: stringr

require(ggpmisc)

## Loading required package: ggpmisc

require(reshape2)

## Loading required package: reshape2
```

Set working directory to source file location *(In RStudio: Session -> Set Working Directory -> To Source File Location)*

```
#setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

Initialize MS1 and MS2 databases and define the name of the sample. The database file must be placed in “~Database/” folder! Do not edit the column names of the database.

```
Target.database <- read_xlsx(path =
"Database/Database_pharmaceuticals_environment.xlsx", sheet = 2)
Suspect.database <- read_xlsx(path =
"Database/Database_pharmaceuticals_environment.xlsx", sheet = 1)
```

Sample name must be typed manually.

```
Sample.name <- "Example_sample"
```

Check if the folders for Sample.name already exist. Folders will be created automatically in accordance to the given value in **Sample.name** variable

```
if (dir.exists("Samples/") == FALSE) {dir.create("Samples/")}
if (dir.exists(paste("Samples/",Sample.name,"/",sep="")) == FALSE)
{dir.create(paste("Samples/",Sample.name,"/",sep=""))}
if (dir.exists(paste("Samples/",Sample.name,"/MS1_spectra/",sep="")) ==
FALSE) {dir.create(paste("Samples/",Sample.name,"/MS1_spectra/",sep=""))}
if (dir.exists(paste("Samples/",Sample.name,"/MS2_spectra_suspect/",sep=""))
== FALSE)
{dir.create(paste("Samples/",Sample.name,"/MS2_spectra_suspect/",sep=""))}
if (dir.exists(paste("Samples/",Sample.name,"/MS2_spectra_suspect/",sep=""))
== FALSE)
{dir.create(paste("Samples/",Sample.name,"/MS2_spectra_target/",sep=""))}
if (dir.exists(paste("Samples/",Sample.name,"/Precursor lists/",sep="")) ==
FALSE) {dir.create(paste("Samples/",Sample.name,"/Precursor lists/",sep=""))}
if (dir.exists(paste("Samples/",Sample.name,"/Results/",sep="")) == FALSE)
{dir.create(paste("Samples/",Sample.name,"/Results/",sep=""))}
```

Activate custom-made functions

```
Spectra.find <- function(Spectra, Q1, Q2, Ratio, Mass.error.threshold) {
  colnames(Spectra) <- c("V1","V2")
  Error.1 <- Q1*(Mass.error.threshold/1000000)
  mz.1 <- Spectra %>% filter((V1 > Q1-Error.1) & (V1 < Q1+Error.1))
  Error.2 <- Q2*(Mass.error.threshold/1000000)
  mz.2 <- Spectra %>% filter((V1 > Q2-Error.2) & (V1 < Q2+Error.2))
  if (dim(mz.2)[1] == 0) {mz.2 <- data.frame(V1 = NA, V2 = NA)}
  if (dim(mz.1)[1] == 0) {
    mz.1 <- data.frame(V1 = NA, V2 = NA)
    mz.2 <- data.frame(V1 = NA, V2 = NA)
  }
  if (dim(mz.2)[1] > 1) {mz.2 <- (mz.2 %>% mutate(Error = abs(V1-Q2)) %>%
filter(Error == min(Error)))[,c(1:2)]}
  if (dim(mz.1)[1] > 1) {mz.1 <- (mz.1 %>% mutate(Error = abs(V1-Q1)) %>%
filter(Error == min(Error)))[,c(1:2)]}
  Ratio.exp <- (mz.2$V2 / mz.1$V2)*100
  if (is.na(mz.1$V1)==FALSE) {Error.1 <- (mz.1$V1-Q1)/Q1*1000000 } else
{Error.1 <- NA}
  if (is.na(mz.2$V1)==FALSE) {Error.2 <- (mz.2$V1-Q2)/Q2*1000000 } else
{Error.2 <- NA}
  Ratio.error <- (Ratio.exp-Ratio)/Ratio*100
  Ratio.error.abs <- Ratio.exp-Ratio
  return(c(as.numeric(round(mz.1$V2,0)),
           as.numeric(round(Error.1,3)),
           as.numeric(round(Error.2,3)),
           as.numeric(round(Ratio.exp,2)),
           as.numeric(round(Ratio.error,2)),
           as.numeric(round(Ratio.error.abs,2))))
}
```

```r
MS2.match <- function(Spectra, Fragment,Mass.error.threshold) {
  Error.1 <- Fragment*(Mass.error.threshold/1000000)
  mz.1 <- Spectra %>% filter((V1 > Fragment-Error.1) & (V1 < 
Fragment+Error.1))
  if (dim(mz.1)[1] == 0) {
    return(FALSE) } else {
      return(TRUE)
    }
}

Consolidate_MS2_results <- function(MSMS_sheet, Storage_sheet) {

  {MSMS_sheet <- MSMS_sheet %>%
    group_by(Name,Polarity,CE.level) %>%
    mutate(Fragments.found = sum(Status.found, na.rm = TRUE), 
Fragments.not.found = sum(Status.found == FALSE, na.rm = TRUE)) %>%
    select(Name,Polarity,CE.level,Fragments.found,Fragments.not.found) %>%
    unique(.) %>%
    group_by(Name,Polarity) %>%
    mutate(MS2.found = max(Fragments.found, na.rm = TRUE),
           MS2.fragments.in.db = (sum(Fragments.not.found, na.rm = 
TRUE)+sum(Fragments.found, na.rm = TRUE))/3,
           MS2.CE.level = (CE.level[Fragments.found == max(Fragments.found, 
na.rm = TRUE)])[1],
           Status = ifelse(MS2.found == 0, FALSE, TRUE)) %>%
    select(Name, Polarity, MS2.found,MS2.CE.level,MS2.fragments.in.db,Status) 
%>%
    unique(.)}

  Storage_sheet$MS2.found <- NA
  Storage_sheet$MS2.CE.level <- NA
  Storage_sheet$MS2.fragments.in.db <- NA
  Storage_sheet$MS2.Status <- NA


  for (i in 1:dim(Storage_sheet)[1]) {
    Storage_sheet$MS2.found[i] <- unname(unlist(MSMS_sheet %>%
                                                  filter(Name == 
Storage_sheet$Name[i]) %>%
                                                  filter(Polarity == 
Storage_sheet$Charge[i]))[3])
    Storage_sheet$MS2.CE.level[i] <- unname(unlist(MSMS_sheet %>%
                                                     filter(Name == 
Storage_sheet$Name[i]) %>%
                                                     filter(Polarity == 
Storage_sheet$Charge[i]))[4])
    Storage_sheet$MS2.fragments.in.db[i] <- unname(unlist(MSMS_sheet %>%
                                                            filter(Name == 
Storage_sheet$Name[i]) %>%
                                                            filter(Polarity
== Storage_sheet$Charge[i]))[5])
    Storage_sheet$MS2.Status[i] <- unname(unlist(MSMS_sheet %>%
                                                   filter(Name ==
Storage_sheet$Name[i]) %>%
                                                   filter(Polarity ==
Storage_sheet$Charge[i]))[6])

  }
  return(Storage_sheet)
}

MS2.screener <-
function(Precursor.list,MS2.peaklist,Sample.name,Type,Template,MS2.ref.spectr
a.type) {
  if (Type == "Target") { Type <-"MS2_spectra_target"}
  if (Type == "Suspect") { Type <-"MS2_spectra_suspect"}
  for (i in 1:dim(Precursor.list)[1]) {
    spectra.MS2 <-
read.csv(paste("Samples/",Sample.name,"/",Type,"/",Precursor.list$File.name[i
],sep = ""),sep = ",",header = T)
    colnames(spectra.MS2) <- c("V1","V2")
    if (MS2.ref.spectra.type == "Experimental") {
    MS2.fragments <- unlist(
      str_split(
        (MS2.peaklist[which(MS2.peaklist$Name %in% Precursor.list$Name[i]),]
%>%
           filter(Polarity == Precursor.list$Charge[i]))$`MS2 fingerprint
(experimental), m/z`[1],pattern = ";"))}
    if (MS2.ref.spectra.type == "In-silico") {
      MS2.fragments <- unlist(
        str_split(
          (MS2.peaklist[which(MS2.peaklist$Name %in%
Precursor.list$Name[i]),] %>%
             filter(Polarity == Precursor.list$Charge[i]))$`MS2 fingerprint
(predicted), m/z`[1],pattern = ";"))}


    for (g in 1:length(MS2.fragments)) {
      Template <- rbind(Template,
                        data.frame(Name =
as.character(Precursor.list$Name[i]),
                                   Polarity =
as.numeric(Precursor.list$Charge[i]),
                                   Fragment = as.numeric(MS2.fragments[g]),
                                   Status.found =
MS2.match(spectra.MS2,as.numeric(MS2.fragments[g]),Mass.error.threshold =
2.5),
                                   CE.level = 
as.character(Precursor.list$CE.level[i]),
                                   File.name = Precursor.list$File.name[i],
stringsAsFactors = FALSE))

    }}
  return(Template)
}
```

Sample measurement

Measure the sample to obtain two full MS spectra (in both polarities). Export the spectra as CSV files. Filename for negative mode: Sample.name-NEG.csv, filename for positive mode: Sample.name-POS.csv.

Store both spectra in the main directory under ~/Samples/**Sample_name**/MS1_spectra/. CSV files have to contain two columns (m/z values and m/z signal intensities):

Column 1 (m/z)	Column 2 (signal intensity, counts)
294.0068	121821
296.0063	29324

Target screening

Load full MS data files for the sample.

```
Spectra.list <- list(spectra.pos = 
read.csv(paste("Samples/",Sample.name,"/MS1_spectra/",Sample.name,"-
POS.csv",sep = ""),sep = ",",header = T),
                     spectra.neg = 
read.csv(paste("Samples/",Sample.name,"/MS1_spectra/",Sample.name,"-
NEG.csv",sep = ""),sep = ",",header = T))
```

Create data frame for target results.

```
Results.target <- data.frame(Name = as.character(NA),
                            MolecularFormula = as.character(NA),
                            Q1 = as.numeric(NA),
                            Ratio.theoretical = as.numeric(NA),
                            Charge = as.numeric(NA),
                            Intensity = as.numeric(NA),
                            Error.Q1 = as.numeric(NA),
                            Error.Q2 = as.numeric(NA),
                            Ratio.experimental = as.numeric(NA),
                            Ratio.error.rel = as.numeric(NA),
                            Ratio.error.abs = as.numeric(NA),
stringsAsFactors = FALSE)
```

Perform full MS target screening.

```
for (i in 1:dim(Target.database)[1]) {
if (Target.database$Polarity[i] == 1) {MS1.data <- Spectra.list$spectra.pos}
else {MS1.data <- Spectra.list$spectra.neg}
  Results.target <- rbind(Results.target,
c(as.character(Target.database$Name[i]),

as.character(Target.database$MolecularFormula[i]),
                                            as.numeric(Target.database$`Q1,
m/z`[i]),
                                            as.numeric(Target.database$`Ratio
Q2/Q1`[i]),

as.numeric(Target.database$Polarity[i]),

Spectra.find(MS1.data,Target.database$`Q1, m/z`[i],
                                                         Target.database$`Q2,
m/z`[i],

Target.database$`Ratio Q2/Q1`[i],
                                                         Mass.error.threshold
= 1.25)))
}
```

Suspect screening

Create data frame for suspect screening results.

```
Results.suspect <- data.frame(Name = as.character(NA),
                             MolecularFormula = as.character(NA),
                             Q1 = as.numeric(NA),
                             Ratio.theoretical = as.numeric(NA),
                             Charge = as.numeric(NA),
                             Intensity = as.numeric(NA),
                             Error.Q1 = as.numeric(NA),
                             Error.Q2 = as.numeric(NA),
                             Ratio.experimental = as.numeric(NA),
                             Ratio.error.rel = as.numeric(NA),
                             Ratio.error.abs = as.numeric(NA),
stringsAsFactors = FALSE)
```

Perform full MS suspect screening.

```
for (i in 1:dim(Suspect.database)[1]) {
  if (Suspect.database$Polarity[i] == 1) {MS1.data <-
Spectra.list$spectra.pos} else {MS1.data <- Spectra.list$spectra.neg}
  Results.suspect <- rbind(Results.suspect,
c(as.character(Suspect.database$Name[i]),

as.character(Suspect.database$MolecularFormula[i]),
                                            as.numeric(Suspect.database$`Q1,
m/z`[i]),

as.numeric(Suspect.database$`Ratio Q2/Q1`[i]),

as.numeric(Suspect.database$Polarity[i]),

Spectra.find(MS1.data,Suspect.database$`Q1, m/z`[i],

Suspect.database$`Q2, m/z`[i],

Suspect.database$`Ratio Q2/Q1`[i],
                                                         Mass.error.threshold
= 1.25)))
}
```

Filtering compliant matches for further MS2 analysis

Convert numbers to the correct format (numeric).

```
  Results.suspect[,3:11] <-
sapply(Results.suspect[,3:11,drop=FALSE],as.numeric)
  Results.target[,3:11] <-
sapply(Results.target[,3:11,drop=FALSE],as.numeric)
```

Define the number of MS2 spectra that will be acquired per compound. **Change** the value if the number of MS2 spectra per compound is not 3!

```
  No.MS2.spectra <- 3
```

Create a list of MS2 spectra filenames (CSV) for target screening.

```
Target.MSMS.list <- Results.target %>%
    mutate(Max.ratio.error = 20) %>%
    mutate(Max.ratio.error = ifelse(Ratio.theoretical > 50, 20,
                                    ifelse(Ratio.theoretical > 20, 25,
                                           ifelse(Ratio.theoretical > 10, 30,
50)))) %>%
    filter((Ratio.error.rel < Max.ratio.error) & (Ratio.error.rel > -
Max.ratio.error)) %>%
    filter((Error.Q1 < 1.25) & (Error.Q2 > -1.25)) %>%
    select(Name, Q1, Charge) %>%
    slice(rep(1:n(), each=No.MS2.spectra)) %>%
    mutate(File.name = paste(row_number(),"-",ifelse(Charge ==
1,"POS","NEG"),".csv", sep = ""))
```

Create a list of MS2 spectra filenames (CSV) for suspect screening.

```
Suspect.MSMS.list <- Results.suspect %>%
  mutate(Max.ratio.error = 20) %>%
#  mutate(Max.ratio.error = ifelse(Ratio.theoretical > 50, 20,
#' Delete "#" to make Q2/Q1 ratio thershold in accordance to 2002/657/EC
#                                  ifelse(Ratio.theoretical > 20, 25,
#' Delete "#" to make Q2/Q1 ratio thershold in accordance to 2002/657/EC
#                                         ifelse(Ratio.theoretical > 10, 30,
50)))) %>%   #' Delete "#" to make Q2/Q1 ratio thershold in accordance to
2002/657/EC
  filter((Ratio.error.rel < Max.ratio.error) & (Ratio.error.rel > -
Max.ratio.error)) %>%
  filter((Error.Q1 < 1.25) & (Error.Q2 > -1.25)) %>%
  select(Name, Q1, Charge) %>%
  slice(rep(1:n(), each=No.MS2.spectra)) %>%
  mutate(File.name = paste(row_number(),"-",ifelse(Charge ==
1,"POS","NEG"),".csv", sep = ""))
```

Assign collision energy (CE) levels for MS2 measurements.

If **No.MS2.spectra** == 1 then only one level is defined!

If **No.MS2.spectra** > 1 then set CE levels manually by writing them under **CE.levels** variable!

2 levels: CE.levels <- c(“1st CE level”, “2nd CE level”)

3 levels: CE.levels <- c(“1st CE level”, “2nd CE level”, “3rd CE level”)

4 levels: CE.levels <- c(“1st CE level”, “2nd CE level”, “3rd CE level”, “4th level”)

etc.

```
CE.levels <- c("1st CE level", "2nd CE level", "3rd CE level")

if (No.MS2.spectra == 1) {
  Target.MSMS.list$CE.level <- as.character("your CE level")
  Suspect.MSMS.list$CE.level <- as.character("your CE level")
} else {
  Target.MSMS.list$CE.level <- as.character(NA)
  Suspect.MSMS.list$CE.level <- as.character(NA)

  for (i in 1:dim(Target.MSMS.list)[1]) {
    Target.MSMS.list$CE.level[i] <- CE.levels[c(No.MS2.spectra,
seq(1:(No.MS2.spectra-1)))[(i%%No.MS2.spectra)+1]]
  }
  for (i in 1:dim(Suspect.MSMS.list)[1]) {
    Suspect.MSMS.list$CE.level[i] <- CE.levels[c(No.MS2.spectra,
seq(1:(No.MS2.spectra-1)))[(i%%No.MS2.spectra)+1]]
  }}
```

Check both MS2 filename lists:

```
head(Target.MSMS.list)
```

```
##            Name       Q1 Charge File.name     CE.level
## 1 Atorvastatin 559.2603      1 1-POS.csv 1st CE level
## 2 Atorvastatin 559.2603      1 2-POS.csv 2nd CE level
## 3 Atorvastatin 559.2603      1 3-POS.csv 3rd CE level
## 4   Diclofenac 294.0094     -1 4-NEG.csv 1st CE level
## 5   Diclofenac 294.0094     -1 5-NEG.csv 2nd CE level
## 6   Diclofenac 294.0094     -1 6-NEG.csv 3rd CE level
```

```
head(Suspect.MSMS.list)
```

```
##            Name       Q1 Charge File.name     CE.level
## 1   Alprazolam 309.0902      1 1-POS.csv 1st CE level
## 2   Alprazolam 309.0902      1 2-POS.csv 2nd CE level
## 3   Alprazolam 309.0902      1 3-POS.csv 3rd CE level
## 4 Atorvastatin 559.2603      1 4-POS.csv 1st CE level
## 5 Atorvastatin 559.2603      1 5-POS.csv 2nd CE level
## 6 Atorvastatin 559.2603      1 6-POS.csv 3rd CE level
```

Write precursor mass lists to a file

```
write.table(Target.MSMS.list,file = paste("Samples/",Sample.name,"/Precursor
lists/",Sample.name,"-Precursor_mass_list_target.xls",sep = ""),sep =
"\t",row.names = FALSE)
write.table(Suspect.MSMS.list,file = paste("Samples/",Sample.name,"/Precursor
lists/",Sample.name,"-Precursor_mass_list_suspect.xls",sep = ""),sep =
"\t",row.names = FALSE)
```

Sample measurement (MS2 analysis)

Measure the sample to obtain MS2 spectra that correspond to the generated lists: **Target.MSMS.list** and **Suspect.MSMS.list**

Each analyte has to be measured **No.MS2.spectra** times! In this example **No.MS2.spectra** is 3 and the measurements are done in three collision energy levels (low, medium and high).

Export the spectra as CSV files. Name the CSV files in accordance to generated lists (**Target.MSMS.list** and **Suspect.MSMS.list**).

Store target MS2 spectra in the main directory under
~/Samples/Sample_name/MS2_spectra_target/

Store suspect MS2 spectra in the main directory under ~/Samples/Sample_name/MS2_spectra_suspect/

CSV files of MS2 spectra have to contain two columns (m/z values and m/z signal intensities)

Column 1 (m/z)	Column 2 (signal intensity, counts)
294.0068	121821
296.0063	29324

MS2 data processing (target analysis; with experimental library spectra)

```
Results.target <- Consolidate_MS2_results(MS2.screener(Precursor.list = 
Target.MSMS.list,
                                                       MS2.peaklist = 
Target.database,
                                                       Sample.name = 
Sample.name,
                                                       Type = "Target",
                                                       MS2.ref.spectra.type = 
"Experimental",
                                                       Template = 
data.frame(Name = as.character(NA),

Polarity = as.numeric(NA),

Fragment = as.numeric(NA),

Status.found = as.logical(NA),

CE.level = as.character(NA),

File.name = NA, stringsAsFactors = FALSE)),Storage_sheet = Results.target)
```

Check target results:

```
Results.target %>%
  filter(MS2.Status == TRUE) %>%
  head(.)

##            Name MolecularFormula       Q1 Ratio.theoretical Charge
## 1 Atorvastatin      C33H35FN2O5 559.2603          36.38933      1
## 2   Diclofenac     C14H11Cl2NO2 294.0094          64.42621     -1
## 3    Ibuprofen         C13H18O2 205.1234          14.13700     -1
##   Intensity Error.Q1 Error.Q2 Ratio.experimental Ratio.error.rel
## 1     10000     0.55    -0.78              31.32          -13.93
## 2     10000     1.08    -0.44              59.11           -8.25
## 3     10000     0.91     0.98              12.98           -8.18
##   Ratio.error.abs MS2.found MS2.CE.level MS2.fragments.in.db MS2.Status
## 1           -5.07        3 3rd CE level                  3       TRUE
## 2           -5.32        2 1st CE level                  3       TRUE
## 3           -1.16        1 1st CE level                  1       TRUE
```

Write target results to a file.

```
write.table(Results.target[-1,],file =
paste("Samples/",Sample.name,"/Results/",Sample.name,"-
Target_results.xls",sep = ""),sep = "\t",row.names = FALSE)
```

MS2 data processing (suspect screeining; with experimental library spectra)

```
Results.suspect <- Consolidate_MS2_results(MS2.screener(Precursor.list =
Suspect.MSMS.list,
                                                       MS2.peaklist =
Suspect.database,
                                                       Sample.name =
Sample.name,
                                                       Type = "Suspect",
                                                       MS2.ref.spectra.type =
"Experimental",
                                                       Template =
data.frame(Name = as.character(NA),

Polarity = as.numeric(NA),

Fragment = as.numeric(NA),

Status.found = as.logical(NA),

CE.level = as.character(NA),

File.name = NA, stringsAsFactors = FALSE)),Storage_sheet = Results.suspect)
```

Check target results:

```
Results.suspect %>%
  filter(MS2.Status == TRUE) %>%
  head(.)
```

```
##                 Name MolecularFormula       Q1 Ratio.theoretical Charge
## 1        Alprazolam       C17H13ClN4 309.0902          32.20218      1
## 2       Venlafaxine        C17H27NO2 278.2115          18.45302      1
## 3        Diclofenac     C14H11Cl2NO2 294.0094          64.42621     -1
## 4 Meclofenamic acid     C14H11Cl2NO2 294.0094          64.42621     -1
## 5         Ibuprofen         C13H18O2 205.1234          14.13700     -1
##   Intensity Error.Q1 Error.Q2 Ratio.experimental Ratio.error.rel
## 1     10000    -0.64    -0.93              28.71          -10.84
## 2     10000     1.13    -0.68              16.12          -12.64
## 3     10000     1.08    -0.44              59.11           -8.25
## 4     10000     1.08    -0.44              59.11           -8.25
## 5     10000     0.91     0.98              12.98           -8.18
##   Ratio.error.abs MS2.found MS2.CE.level MS2.fragments.in.db MS2.Status
## 1           -3.49         4 3rd CE level                   4       TRUE
## 2           -2.33         5 1st CE level                   5       TRUE
## 3           -5.32         2 1st CE level                   3       TRUE
## 4           -5.32         1 2nd CE level                   2       TRUE
## 5           -1.16         1 1st CE level                   1       TRUE
```

Write target results to a file

```
write.table(Results.suspect[-1,],file =
paste("Samples/",Sample.name,"/Results/",Sample.name,"-
Suspect_results_exp_MS2.xls",sep = ""),sep = "\t",row.names = FALSE)
```

MS2 data processing (suspect screeining; with in-silico libray spectra)

```
Results.suspect.in.silico <-
Consolidate_MS2_results(MS2.screener(Precursor.list =  Suspect.MSMS.list,
                                                        MS2.peaklist =
Suspect.database,

                                                        Sample.name =

Sample.name,

                                                        Type = "Suspect",
                                                        MS2.ref.spectra.type
= "In-silico",

                                                        Template =

data.frame(Name = as.character(NA),

Polarity = as.numeric(NA),

Fragment = as.numeric(NA),

Status.found = as.logical(NA),

CE.level = as.character(NA),

File.name = NA, stringsAsFactors = FALSE)),Storage_sheet = Results.suspect)
```

Check target results:

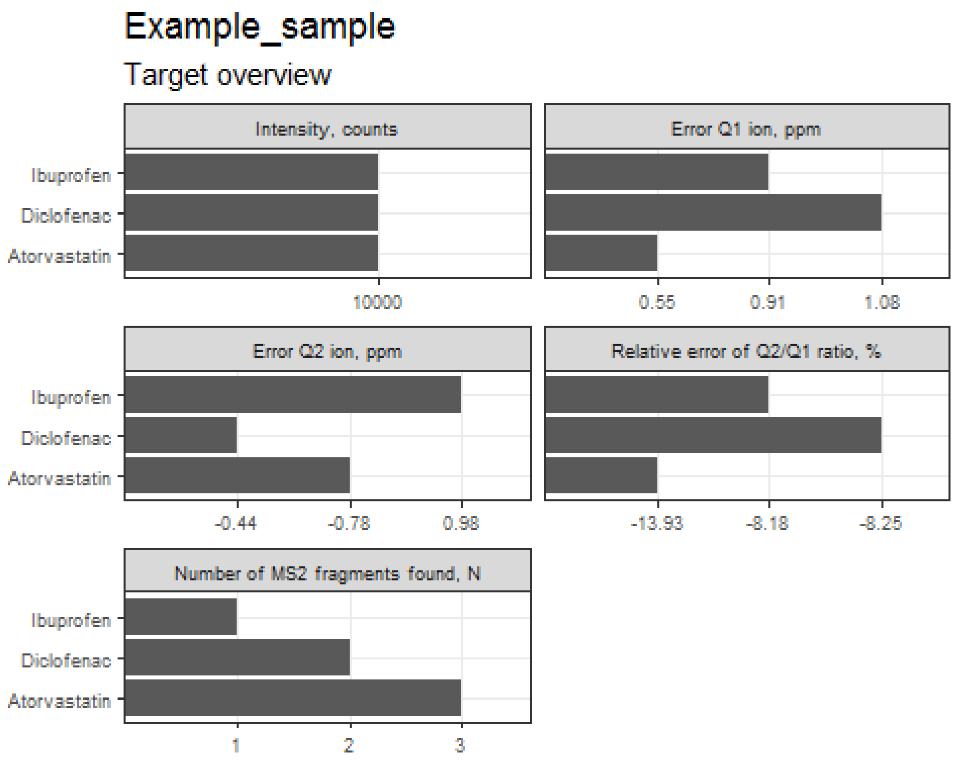
```
Results.suspect.in.silico %>%
  filter(MS2.Status == TRUE) %>%
  head(.)

##                Name MolecularFormula       Q1 Ratio.theoretical Charge
## 1       Venlafaxine        C17H27NO2 278.2115          18.45302      1
## 2        Diclofenac      C14H11Cl2NO2 294.0094          64.42621     -1
## 3 Meclofenamic acid     C14H11Cl2NO2 294.0094          64.42621     -1
##   Intensity Error.Q1 Error.Q2 Ratio.experimental Ratio.error.rel
## 1     10000     1.13    -0.68              16.12          -12.64
## 2     10000     1.08    -0.44              59.11           -8.25
## 3     10000     1.08    -0.44              59.11           -8.25
##   Ratio.error.abs MS2.found MS2.CE.level MS2.fragments.in.db MS2.Status
## 1           -2.33         2 1st CE level                   5       TRUE
## 2           -5.32         1 1st CE level                   5       TRUE
## 3           -5.32         1 1st CE level                   5       TRUE
```

Write target results to a file.

```
write.table(Results.suspect.in.silico[-1,],file =
paste("Samples/",Sample.name,"/Results/",Sample.name,"-Suspect_results_in-
silico_MS2.xls",sep = ""),sep = "\t",row.names = FALSE)
```

Visualize/preview target results

```
melt(Results.target %>%
  filter(MS2.Status == TRUE),id.vars = "Name", variable.name = "Variable")
%>%
  filter(Variable %in% c("Intensity", "Error.Q1",
"Error.Q2","Ratio.error.rel","MS2.found")) %>%
  mutate(Variable = factor(Variable, labels = c("Intensity, counts",
                                                "Error Q1 ion, ppm",
                                                "Error Q2 ion, ppm",
                                                "Relative error of Q2/Q1
ratio, %",
                                                "Number of MS2 fragments
found, N"))) %>%
  ggplot(., aes(x = Name, y = value))+
  geom_bar(stat = "identity")+
  coord_flip()+
  ggtitle(label = paste(Sample.name),subtitle = "Target overview")+
  theme_bw()+
  theme(strip.text = element_text(size = 7),
        axis.text = element_text(size = 7),
        axis.title = element_blank())+
  facet_wrap(~Variable, scales = "free_x",ncol = 2)
```

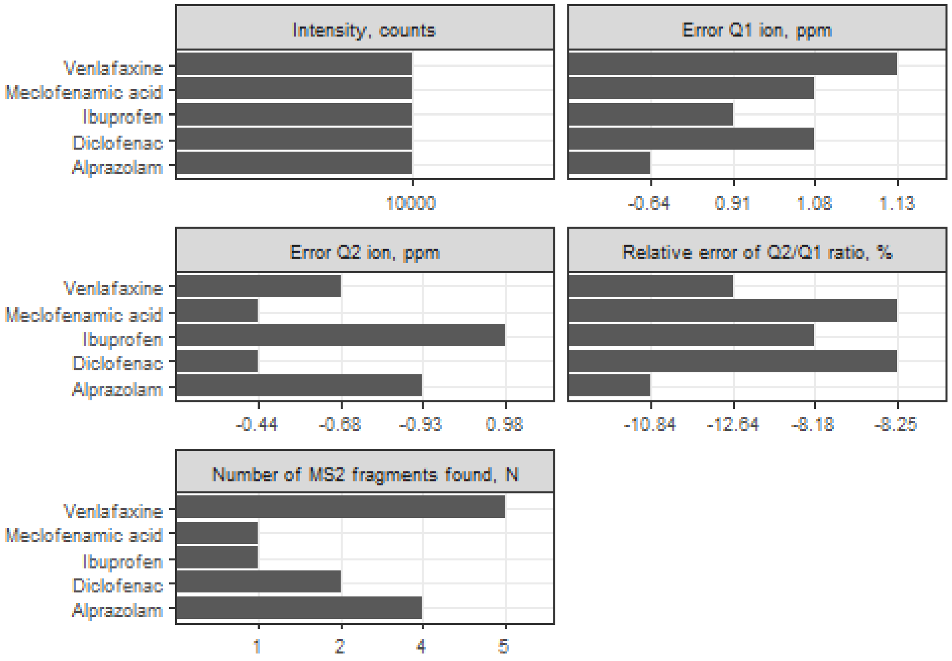


Visualize/preview suspect results

```
melt(Results.suspect %>%
       filter(MS2.Status == TRUE),id.vars = "Name", variable.name =
"Variable") %>%
  filter(Variable %in% c("Intensity", "Error.Q1",
"Error.Q2","Ratio.error.rel","MS2.found")) %>%
  mutate(Variable = factor(Variable, labels = c("Intensity, counts",
                                                "Error Q1 ion, ppm",
                                                "Error Q2 ion, ppm",
                                                "Relative error of Q2/Q1
ratio, %",
                                                "Number of MS2 fragments
found, N"))) %>%
  ggplot(., aes(x = Name, y = value))+
  geom_bar(stat = "identity")+
  coord_flip()+
  ggtitle(label = paste(Sample.name),subtitle = "Suspect overview")+
  theme_bw()+
  theme(strip.text = element_text(size = 7),
        axis.text = element_text(size = 7),
        axis.title = element_blank())+
  facet_wrap(~Variable, scales = "free_x",ncol = 2)
```

Example_sample

Suspect overview


Intensity, counts
Error Q1 ion, ppm
Error Q2 ion, ppm
Relative error of Q2/Q1 ratio, %
Number of MS2 fragments found, N
Venlafaxine
Meclofenamic acid
Ibuprofen
Diclofenac
Alprazolam