---

title: "MFAssignR (by Simeon K Schum, Lynn R Mazzoleni, Laura Brown) formula assignment markdown"

author: "This markdown is for iterative formula assignment (with recalibration) to mass lists from -ESI data (Amna Ijaz)"

date: "Thursday, September 03, 2020"

output:

  html_document:

    toc: yes

    df_print: kable

```{r Package, echo=FALSE, eval=FALSE}

#library(devtools)

#setwd("/Users/amnaijaz/Documents/PhD work/Research/R/MFAssignR")

#devtools::install(pkg = "MFAssignR_Nov 23, 2020")

##To install an older/newer version of a package available at CRAN

#install_version("ggplot2", version = "3.3.1", repos = "http://cran.us.r-project.org")

```

```{r Setup, echo=FALSE, message=FALSE, warning=FALSE, include=TRUE, paged.print=FALSE}

library(lemon)

library(MFAssignR)
```

library(dplyr)

library(ggplot2)

library(viridis)

knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, results = FALSE)

knit_print.data.frame <- lemon_print

```

### MFAssignR version: "MFAssignR_Nov 23, 2020"

### Mass list

```{r Directory and file name, warning=FALSE, message = FALSE, echo=TRUE}

setwd("/Users/amnaijaz/Documents/PhD work/Research/DOM:NOM_Biomass_PNNL_2020Feb01-08/21-T Data_ESI_Richland BB Smoke (BB0905)")

Data <- read.csv("BB_200k_mFT_Noise9.csv") %>% arrange(desc(abundance))

file.name <- "BB_200k_mFT_Noise9"

```

Formula assignment output

Number of peaks in the raw mass list

```{r Raw peak count, echo = FALSE, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print, fig.width=3}

count(Data)

```

Signal-to-noise estimation

```{r SN estimate, echo = TRUE, message=FALSE, warning=FALSE, paged.print=FALSE, results= TRUE}

Noise <- KMDNoise(Data, upper.y = 0.52, lower.y = 0.07, upper.x = NA, lower.x = NA)

```

```{r SN estimate plot, echo = FALSE, message=FALSE, warning=FALSE, results= TRUE, fig.width=5.5, fig.height=3}

plot <- Noise[["KMD"]]

plot

KMDN <-Noise[["Noise"]]

```

```{r SN plots, echo = TRUE, message=FALSE, warning=FALSE, paged.print=FALSE, results= TRUE, fig.width=9, fig.height=5, include=TRUE}

SNplot(Data, cut = 1.5*KMDN, mass = 301, window.x = 0.4, window.y = 5)

SNplot(Data, cut = 2.0*KMDN, mass = 301, window.x = 0.4, window.y = 5)

```

```{r KMDN value, echo = FALSE, message=FALSE, warning=FALSE, results=TRUE, render=lemon_print, fig.width=3}

noise <- as.data.frame(2.0*KMDN)

noise

```

Number of peaks above the noise threshold

```{r post-noise removal, echo = TRUE, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print, include=TRUE}

peaks <- Data %>% filter(abundance > 2.0*KMDN)

count(peaks)

```

```{r SN estimate plot 2, echo = FALSE, message=FALSE, warning=FALSE, paged.print=FALSE, results= TRUE, fig.width=5.5, fig.height=3}

Noise2 <- KMDNoise(peaks, upper.y = NA, lower.y = NA, upper.x = NA, lower.x = NA)

plot2 <- Noise2[["KMD"]]

plot2

```

Isotope prescreening

```{r Isotope filtration, echo = TRUE, message=FALSE, warning=FALSE, paged.print=FALSE}

Isotopes <- IsoFiltR(Data, SN = 2.0*KMDN, Carbrat = 60, Sulfrat = 30, Sulferr = 1.2, Carberr = 1.2)

```
```

```{r Isotope data frames, echo = FALSE, message=FALSE, warning=FALSE, paged.print=FALSE}

Mono <- Isotopes[["Mono"]]

Iso <- Isotopes[["Iso"]]

countmono <- count(Mono)

countiso <- count(Iso)

df0 <- data.frame(countmono, countiso) %>% rename(Monoisotopes = n, Polyisotopes = n.1)

```

Mono and polyisotope count

```{r Monoisotope count, echo = FALSE, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print, fig.width=3}

df0

```

Preliminary CHO assignments

```{r CHO assignments, fig.height=5, fig.width=7, message=FALSE, warning=FALSE, paged.print=FALSE, echo=TRUE}

Assign <- MFAssignCHO_RMD(Mono, Iso, ionMode = "neg", SN = 2.0*KMDN, ppm_err = 1.2 , H_Cmin = 0.3, H_Cmax = 2.5, O_Cmax = 2.0, DBEOmax = 20, DBEOmin = -13, NMScut = "on", DeNovo = 1000, highMW = 1200, lowMW = 100)
```

```

```{r preliminary assignment plotting, echo = FALSE, fig.height=3, fig.width=4.5, message=FALSE, warning=FALSE, paged.print=TRUE}

Unambig1.temp <- Assign[["Unambig"]]

Unambig1 <- Unambig1.temp %>% arrange(desc(abundance)) %>% distinct(formula, .keep_all = TRUE)

Ambig1 <- Assign[["Ambig"]]

Unassigned1 <- Assign[["None"]]

MSAssign_CHO <- Assign[["MSAssign"]]

Error_CHO <- Assign[["Error"]]

MSgroups_CHO <- Assign[["MSgroups"]]

VK_CHO <- Assign[["VK"]]

MSAssign_CHO

Error_CHO

MSgroups_CHO

VK_CHO
```

Number of preliminary CHO assignments

```{r preliminary unambiguous assignments, echo = FALSE, fig.height=5, fig.width=4, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print}

countunambig1.temp <- count(Unambig1.temp)

countunambig1 <- count(Unambig1)

countambig1 <- count(Ambig1)

countunassigned1 <- count(Unassigned1)

```

```{r Preliminary ambiguous assignments, echo = FALSE, fig.height=5, fig.width=4, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print}

df <- data.frame(countunambig1.temp,countunambig1, countambig1, countunassigned1) %>% rename("Preliminary unambiguous CHO MF" = n, "Preliminary unambiguous MF (no duplicates)" = n.1, "Preliminary ambiguous CHO MF" = n.2, "Preliminary unassigned CHO MF" = n.3)

df

```

Mass recalibration

```{r Recalibrant list, echo = FALSE, fig.height=5, fig.width=7, message=FALSE, warning=FALSE, paged.print=TRUE, include=TRUE, results=FALSE}

RecalList <- RecalList(df = Unambig1)

```

```{r Recalibration, echo = TRUE, message=FALSE, warning=FALSE, paged.print=TRUE, include=TRUE, results=TRUE, render=lemon_print}

Recalibration <- Recal(Unambig1, peaks = Mono, isopeaks = Iso, mode = "neg", mzRange = 10, SN = 2.0*KMDN, series1 = "O9_H_8", series2 = "O7_H_6", series3 = "O6_H_7", series4 = "O9_H_9", series5 = "O7_H_7", series6 = "O10_H_11", series7 = "O5_H_5", series8 = "O15_H_17", series9 = "O16_H_15", step_O = 3, step_H2 = 5)

```

```{r Recalibration plotting, echo = FALSE, message=FALSE, fig.height=3, fig.width=4.5, warning=FALSE, paged.print=TRUE, include=TRUE, results=FALSE}

Plot_Recalibration <- Recalibration[["Plot"]]

Plot_Recalibration

Mono2 <- Recalibration[["Mono"]]

Iso2 <- Recalibration[["Iso"]]

List <- Recalibration[["RecalList"]]

```

Final formula assignment (Round 1)

```{r Final assignments, message=FALSE, warning=FALSE, paged.print=FALSE, echo=TRUE}

Assign <- MFAssign_RMD(peaks = Mono2, isopeaks = Iso2, lowMW = 100, highMW = 1200, ionMode = "neg", SN = 2.0*KMDN, ppm_err = 0.5, iso_err = 0.5, H_Cmin = 0.3, O_Cmax = 2.0, H_Cmax = 2.5, DBEOmax = 20, DBEOmin = -13, HetCut = "off", NMScut = "on", DeNovo = 1000, SulfCheck = "on", Ambig = "off", MSMS = "off", N3corr = "on")

```

```{r Final assignment plotting, echo = FALSE, fig.height=3, fig.width=4.5, message=FALSE, warning=FALSE, paged.print=TRUE}

Unambig2 <- Assign[["Unambig"]]

Ambig2 <- Assign[["Ambig"]]

Unassigned2 <- Assign[["None"]]

```
MSAssign_MF <- Assign[["MSAssign"]]

Error_MF <- Assign[["Error"]]

MSgroups_MF <- Assign[["MSgroups"]]

VK_MF <- Assign[["VK"]]

KMD_MF <- Assign[["KMD"]]

MSgroupstwo_MF <- Assign[["MSgroupstwo"]]

DBEO_MF <- Assign[["DBEO"]]

VK2_MF <- Assign[["VK2"]]

MSAssign_MF

Error_MF

MSgroups_MF

VK_MF
```

Isotope prescreening and formula assignment (Round 2)

```{r Isotope filtration 2, echo = TRUE, message=FALSE, warning=FALSE, paged.print=FALSE}

Isotopes.2 <- IsoFiltR(Unassigned2, SN = 0*KMDN, Carbrat = 60, Sulfrat = 30, Sulferr = 0.5, Carberr = 0.5)

Mono3 <- Isotopes.2[["Mono"]]

Iso3 <- Isotopes.2[["Iso"]]

```

```{r Final assignments b, fig.height=5, fig.width=7, message=FALSE, warning=FALSE, paged.print=FALSE, echo=TRUE}

Assign.2 <- MFAssign_RMD(peaks = Mono3, isopeaks = Iso3, lowMW = 100, highMW = 1200, ionMode = "neg", SN = 0*KMDN, ppm_err = 0.5, iso_err = 0.5, H_Cmin = 0.3, O_Cmax = 2.0, H_Cmax = 2.5, DBEOmax = 20, DBEOmin = -13, HetCut = "off", NMScut = "on", DeNovo = 350, SulfCheck = "on", Ambig = "off", MSMS = "off", N3corr = "on", Nx = 3, Sx = 1, Px = 1)

```

```{r Final assignment plotting b, echo = FALSE, fig.height=3, fig.width=4.5, message=FALSE, warning=FALSE, paged.print=TRUE}

Unambig3 <- Assign.2[["Unambig"]]

Ambig3 <- Assign.2[["Ambig"]]

Unassigned3 <- Assign.2[["None"]]

MSAssign_MF <- Assign.2[["MSAssign"]]

Error_MF <- Assign.2[["Error"]]

MSgroups_MF <- Assign.2[["MSgroups"]]

VK_MF <- Assign.2[["VK"]]

KMD_MF <- Assign.2[["KMD"]]

MSgroupstwo_MF <- Assign.2[["MSgroupstwo"]]

DBEO_MF <- Assign.2[["DBEO"]]

VK2_MF <- Assign.2[["VK2"]]

MSAssign_MF

Error_MF

MSgroups_MF

VK_MF
```

```

# All final formula assignments

```{r Final unambiguous assignments, echo = TRUE, fig.height=5, fig.width=4, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print}

All.unambig.temp <- rbind(Unambig2, Unambig3)

All.unambig <- All.unambig.temp %>% arrange(desc(abundance)) %>% distinct(formula, .keep_all = TRUE)

All.ambig <- rbind(Ambig2, Ambig3)

```

```{r Final unambiguous assignments numbers, echo = FALSE, fig.height=5, fig.width=4, message=FALSE, warning=FALSE, paged.print=FALSE, results=TRUE, render=lemon_print}

countallunambig.temp <- count(All.unambig.temp)

countallunambig <- count(All.unambig)

countallambig <- count(All.ambig)

countallunassigned <- count(Unassigned3)



C13isotopes <- All.unambig %>% filter(C13_mass != 0)

countC13 <- count(C13isotopes)

C13isotopes2 <- All.unambig %>% filter(C13_mass2 != 0)

countC13two <- count(C13isotopes2)

S34isotopes <- All.unambig %>% filter(S34_mass != 0)
```
```

countS34 <- count(S34isotopes)

df2 <- data.frame(countallunambig.temp, countallunambig, countallambig, countallunassigned,countC13,countC13two, countS34) %>% rename("Monoisotopic MF (duplicates included)" = n, "Final monoisotopic MF (duplicates removed)" = n.1, "Ambiguous MF" = n.2, "Masses not assigned an MF" = n.3, "13C isotopologues" = n.4, "13C2 isotopologues" = n.5, "34S isotopologues" = n.6)

df2

All.unambig$group <- factor(All.unambig$group, levels = c("CHO", "CHNO", "CHOS", "CHNOS", "CH","CHN", "CHNS", "CHS", "CHP", "CHNOP", "CHOP", "CHNOSP"))

Unambigstat <- All.unambig %>% group_by(group) %>% summarize(total = n())

Unambigstat

```

### Final formula assignment spectrum

```{r Final assignment spectrum, echo = FALSE, fig.height=3, fig.width=5, message=FALSE, warning=FALSE, paged.print=TRUE, results=TRUE}

spectra <-ggplot2::ggplot() +

ggplot2::geom_segment(data=All.unambig, size=0.5,ggplot2::aes_string(x = "exp_mass", xend = "exp_mass", y = 0, yend = "abundance"), alpha = 0.9, color = "green")+

ggplot2::geom_segment(data=All.unambig, size=0.5,ggplot2::aes_string(x = "C13_mass", xend = "C13_mass", y = 0, yend = "C13_abund"), alpha = 0.9, color = "blue")+

```
ggplot2::geom_segment(data=All.unambig, size=0.5,ggplot2::aes_string(x = "C13_mass2", xend = "C13_mass2", y = 0, yend = "C13_abund2"), alpha = 0.9, color = "blue")+

ggplot2::geom_segment(data=All.unambig, size=0.5,ggplot2::aes_string(x = "S34_mass", xend = "S34_mass", y = 0, yend = "S34_abund"), alpha = 0.9, color = "blue")+

ggplot2::geom_segment(data=Unassigned3, size=0.5,ggplot2::aes_string(x = "exp_mass", xend = "exp_mass", y = 0, yend = "abundance"), alpha = 0.9, color = "red")+

coord_cartesian(xlim = c(150, 800)) +

ggplot2::labs(x = "m/z", y = "Abundance", title = "Assignment spectrum (duplicates removed)")+

theme_test() +

theme(axis.text=element_text(size=12,face="bold"),
axis.text.y=element_text(size=12,face="bold", angle = 0,
hjust=0.5,vjust=0.5),legend.title=element_text(face="bold", size = 9), axis.title = element_text(size=15,face="bold"),plot.title=element_text(size=15,face="bold"),
legend.text=element_text(face="bold",size=12),strip.text=element_text(face="bold",size=12),legend.position = "bottom",legend.justification="center",strip.background = element_blank())

spectra
```
```

Unidentified peaks

```
```{r Final unassigned, echo = FALSE, fig.height=3, fig.width=5, message=FALSE, warning=FALSE, paged.print=TRUE, results=TRUE}

Noise3 <- KMDNoise(Unassigned3, upper.y = NA, lower.y = NA, upper.x = NA, lower.x = NA)

plot3 <- Noise3[["KMD"]]

plot3
```

```

# Bulk properties of monoisotopic ions

```{r, results=TRUE, echo=FALSE, message = FALSE, fig.width=20, render=lemon_print}

average <- All.unambig %>% summarise(" " = "Average", "O/C" = mean(O_C), "H/C" = mean(H_C), "DBE" = mean(DBE), "#C"  = mean(C), "#O" = mean(O), "#N" = mean(N), "#S" = mean(S), "Absolute error" = mean(AE_ppm), "m/z" = mean(exp_mass))

sd <- All.unambig %>% summarise(" " = "Standard deviation", "O/C" = sd(O_C), "H/C"  = sd(H_C),"DBE" = sd(DBE), "#C"  = sd(C), "#O" = sd(O), "#N" = sd(N), "#S" = sd(S), "Absolute error" = sd(AE_ppm),"m/z" = sd(exp_mass))

table1 <- bind_rows(average, sd)

table1

```

Bulk properties by molecular groups from monoisotopic ions

```{r, results=TRUE, echo=FALSE, message = FALSE, fig.width=20, render=lemon_print}

```
average <- All.unambig %>% group_by(group) %>% summarise(" " = "Average", "O/C" = mean(O_C), "H/C" = mean(H_C), "DBE" = mean(DBE), "#C" = mean(C), "#O" = mean(O), "#N" = mean(N), "#S" = mean(S), "Absolute error" = mean(AE_ppm), "m/z" = mean(exp_mass))

sd <- All.unambig %>% group_by(group) %>% summarise(" " = "Standard deviation", "O/C" = sd(O_C), "H/C" = sd(H_C),"DBE" = sd(DBE), "#C" = sd(C), "#O" = sd(O), "#N" = sd(N), "#S" = sd(S), "Absolute error" = sd(AE_ppm),"m/z" = sd(exp_mass))

table1 <- bind_rows(average, sd) %>% arrange(group)

table1


```

```{r Write files, echo = TRUE, include=FALSE}

max.abundance <- max(Data$abundance)


All.unambig <- All.unambig %>% mutate(SNR = abundance/KMDN, n_abundance = abundance/max.abundance)

write.csv(All.unambig, paste0("MFOutput/", file.name, "_UMF.csv"), row.names=FALSE, na="")


All.ambig <- All.ambig %>% mutate(SNR = abundance/KMDN, n_abundance = abundance/max.abundance)

write.csv(All.ambig, paste0("MFOutput/", file.name, "_AMF.csv"), row.names=FALSE, na="")
```

```
Unassigned3 <- Unassigned3 %>% mutate(SNR = abundance/KMDN, n_abundance = abundance/max.abundance, tag = "Unassigned")

write.csv(Unassigned3, paste0("MFOutput/", file.name, "_NoMF.csv"), row.names=FALSE, na="")


all <- bind_rows(All.unambig,All.ambig,Unassigned3)

write.csv(all, paste0("MFOutput/", file.name, "_Full.csv"), row.names=FALSE)


```