1 Pre-processing of LC/MS metabolomics data: FAAH

Lecturers: Prof. dr. Antoine van Kampen (Amsterdam University Medical Centers, AMC)
Eric Wever (Amsterdam University Medical Centers, AMC)
Dr. Mia Pras-Raves (Amsterdam University Medical Centers, AMC)
Dr. Adrie Dane (Amsterdam University Medical Centers, AMC)

1.1 Background

For this tutorial about the data analysis of LC-MS metabolomics data we will use data that was obtained by Saghaltelian and co-workers [1]. In their study they were interested in identifying substrates for the fatty acid amide hydrolase (FAAH) enzyme, which is an integral membrane hydrolase with a single N-terminal transmembrane domain [2]. In vivo, FAAH is the principal catabolic (breakdown) enzyme for a class of bioactive lipids called the fatty acid amides (FAAs; Figure 1). Members of the FAAs include:

- Anandamide (N-arachidonoylethanolamine), an endocannabinoid
- Other N-acylethanolamines, such as N-oleoylethanolamine and N-palmitoylethanolamine
- The sleep-inducing lipid oleamide
- The N-acyltaurines, which are agonists of the transient receptor potential (TRP) family of calcium channels.

To identify substrate(s) for FAAH *in vivo* they compared FAAH(+/+) wild type (WT) mice with FAAH(-/-) knockout (KO) mice in which the FAAH is inactivated. They used untargeted LC-MS to measure metabolites in brain and spinal cord. The data is obtained in positive ionization mode and collected in centroid mode.

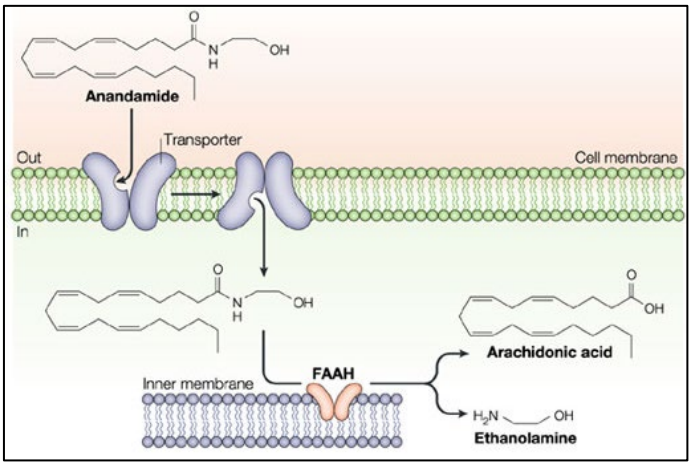

Figure 1. From Piomelli D. (2003) The molecular logic of endocannabinoid signalling. Nature Reviews

Neuroscience, 4(11), 873-84. FAAH is part of endomembrane structures.

Installation instruction for xcms

We will use the **R xcms package** for the analysis of the LC-MS data. Large part of the text below was copied from the xcms tutorial by Colin Smith [3]. Further details about xcms can be found in [4,5,6].

- Install R version 4.0.0 or higher on your computer. Install RStudio.
- Make sure the run R as administrator, otherwise you don't have permissions for installation.
- Carefully check if you do not get error messages during installation.

To install xcms and the FAAH data run the following commands from within R (note: if you are asked to update packages press 'a' (all)):

```
if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")
BiocManager::install("xcms")
BiocManager::install("faahKO")
BiocManager::install("multtest")
```

If you´re using an older version of R, you may need to change to an older version of Bioconductor. See https://www.bioconductor.org/about/release-announcements/ for details, but in principle you should then follow this procedure:

- On local computer desktops of the UvA/AMC with an old version of R first use: .libPaths("C:/Scratch")
- For older versions of R (e.g., R3.5.1) you should use install.packages("biocLite") and biocLite("xcms"). Subsequently, use biocLite("faahKO"), library(xcms), and library(faahKO).
- If the list.files command (see exercises below) doesn't show any files then manually download faahKO as zip file from http://www.bioconductor.org/packages//2.7/data/experiment/html/faahKO.html and, subsequently, from within Rstudio use Tools >> Install packages >> Install from Package archive file (zip) and select the downloaded zip file.

Now you should be able to load (activate) the packages:

```
library(xcms)
library(faahKO)
library(multtest)
```

The data

To get a first impression of the data, I have included three figures. In Figure 2 you see the total ion-current chromatograms (TIC) of the different wild-type (WT) and knock-out (KO) samples. In Figure 3 you see one example mass-spectrum, and in Figure 4 you see of 3D (mz, retention time, and ion current) plot of the data obtained for WT mice number 19. You should be able

to understand what these figures represent. If not, have another look at the lecture material. These plots were not made with xcms but with mzMine (http://mzmine.github.io), which is another software package for doing LC-MS data analysis.

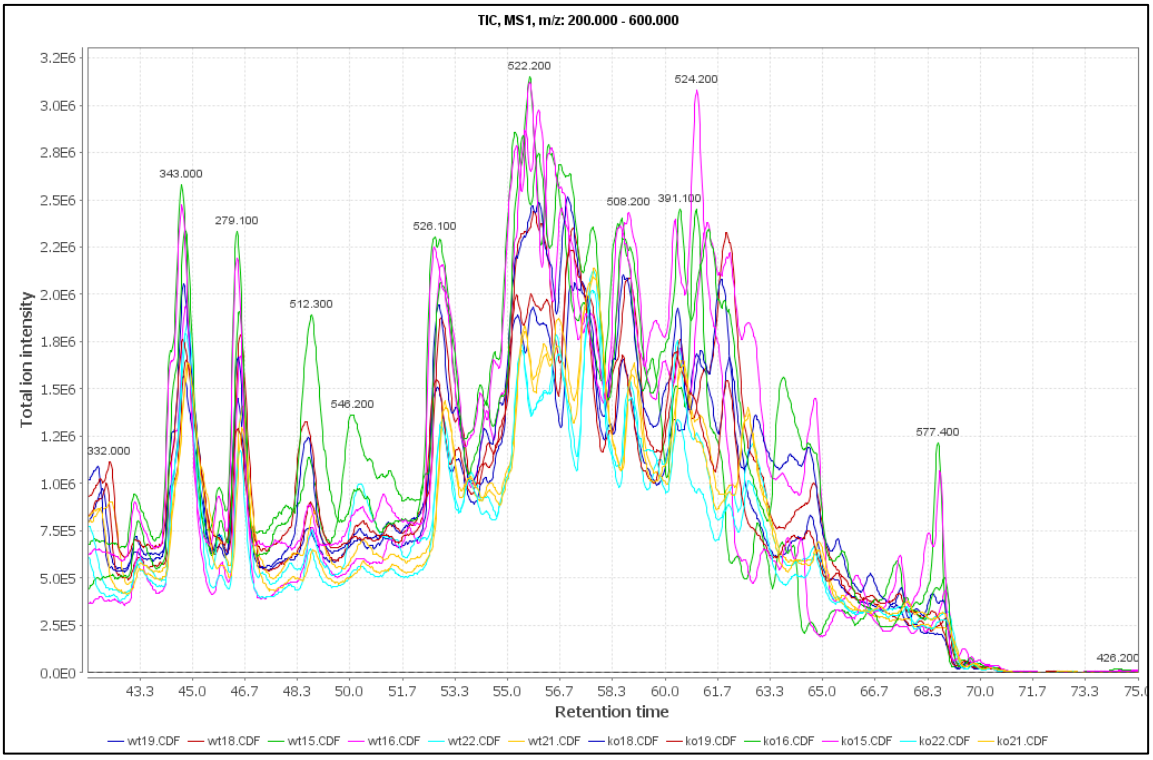

Figure 2. Total Ion Current (TIC) chromatograms (retention time versus intensity) of the WT and KO mice. The peak labels indicate the underlying mass peak with the highest intensity. See Figure 3.

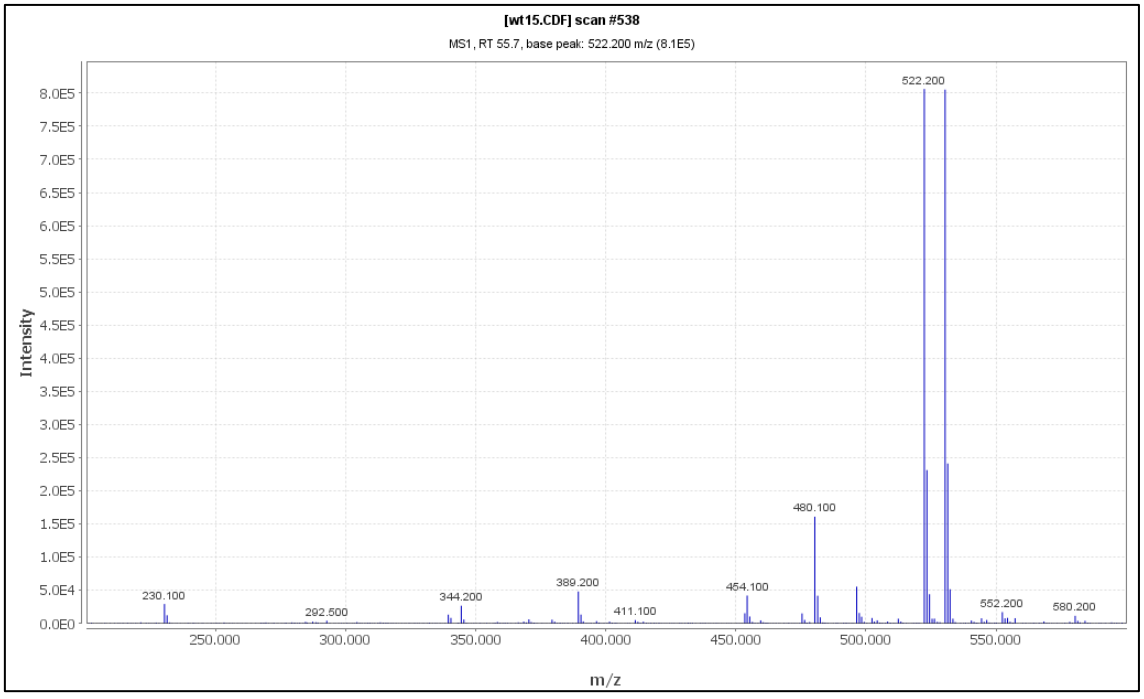

Figure 3. Mass spectrum (m/z versus intensity) of one wild-type mouse. The mass peak at 522.2 is a base peak that corresponds to one of the peaks in Figure 2.

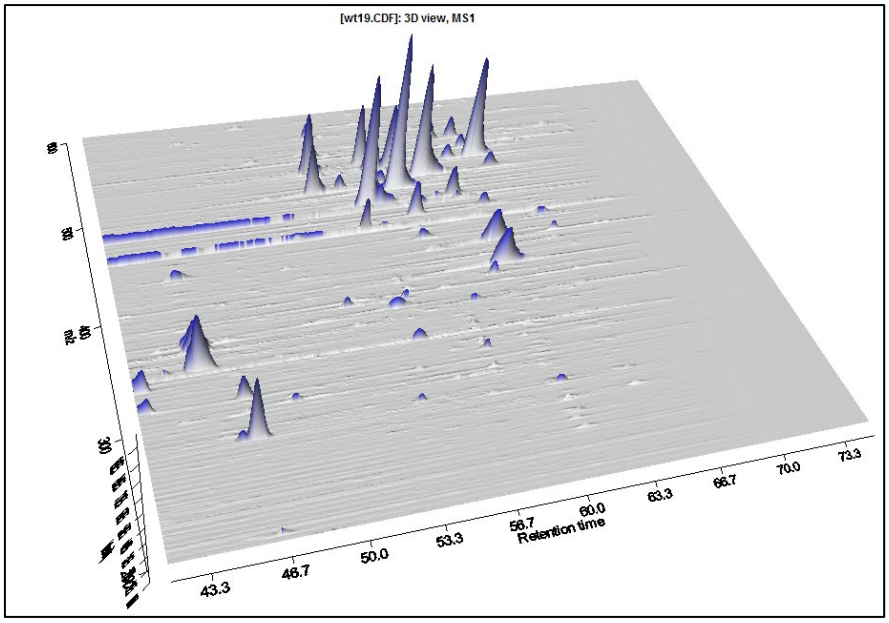

Figure 4. 3D LC-MS plot of WT mice number 19. One axis denotes the retention time. The second axis denotes the m/z values. The height of the peaks (intensities) are correlated to the concentration of the corresponding metabolites.

1.2 Aim

In this tutorial we will identify the peaks (compounds) in 6 WT mice and 6 KO mice. We will pre-process the data to make a comparison between the samples possible and, subsequently, we will compare these two groups of samples to identify the peaks (=metabolites=substrates) that are different between WT and KO.

You will start with the raw LC-MS data of 12 samples. The end result of the metabolomics pre-processing pipeline is a peak table (in an excel file)

1.3 Analysis with XCMS

This document describes how to use xcms to pre-process LC/MS data for relative quantitation and statistical analysis. It gives examples of how visualization can be used throughout the process and to display final results. An overview of the pre-processing/analysis methodology, along with the function names associated with each step, is shown in Figure 5.

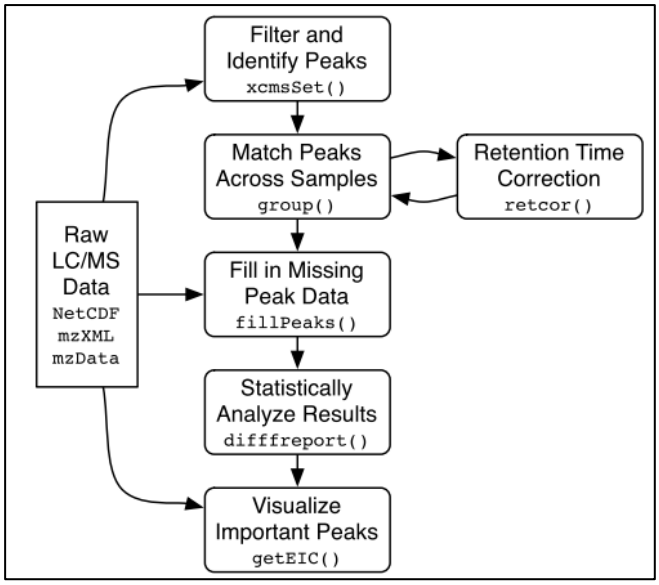

Figure 5: Flow chart showing a high-level overview of the pre-processing/analysis methodology employed by xcms. Function/method names corresponding to each step are also given. In this tutorial we will not use 'getEIC()'. NetCDF, mzXML, and mzData are different data formats that can be used with xcms. Our data is in the 'NetCDF' format.

1.4 Data pre-processing

1.4.1 Loading the LC-MS data

Start the program Rstudio (or R).

Question 1. Loading and visualizing the data

a. Use xcms to determine how many samples are present in the current FAAH datasets. How many WT samples? How many KO samples? Note: the R code is given below in purple.

b. How many mass spectra are in this data set? What was the duration of the measurement?

c. To have a better impression of the data you can use xcms (and, if you want, standard R functions) to visualize chromatograms and mass spectra (of the binned profile data in this example). Make of plot of the Total Ion Current (TIC). What does this plot show? Make a plot of the mass spectrum that was obtained in scan 601. What is the intensity of the highest peak in the spectrum? At what time (in seconds) was this scan measured? Plot a chromatogram based on m/z=496.2. How do you interpret this plot and how is this different from the TIC plot?

The R code to obtain the answer for question (a):
Note that the hash ('#') is used to denote comments in R. You can safely copy/paste comments into R: they will be neglected. The '>' sign denotes the 'R' prompt. Thus, don't type the first '>' in the commands below.

```
#set directory path to data files and list files
> cdfpath <- system.file("cdf",package="faahKO")
> print(cdfpath) #at this location is your data installed
> list.files(cdfpath,recursive=TRUE)
```

The R code to obtain the answer for question (b):

```
#First initialize (=activate) all xcms functions required for the metabolomics data analysis
>library(xcms)

#determine all data files that need to be analysed and store them in a variable (= 'cdffiles')
> cdffiles <- list.files(cdfpath,recursive=TRUE, full.names=TRUE)

#Note that 'library' and 'list.files' are both R commands. To obtain more information about
#e.g., list.files just type ?list.files at the R prompt.

#show what is contained in cdffiles
> cdffiles
> cdffiles[1]

#The next file reads in the first raw data file
#note: if profstep=0 then no profile matrix is generated
> rawdata <-xcmsRaw(cdffiles[1],profstep=0.1,profmethod="bin")
> rawdata
```

Note: if you want help about an xcms function (e.g., xcmsRaw) then type the following at the R prompt:

```
> ?xcmsRaw
Or
> help(xcmsRaw)
```

The R code to obtain the answer for question (c):

```
#Note: the raw lc-ms data is internally stored in a variable that is called 'profile' which can be accessed
through rawdata@env$profile. You can have a closer look at this data by using the commands below.

#show which data parts are stored in the object 'rawdata'. An object can be seen as a container that
stores several other variables or objects.
> str(rawdata,max.level=2)

#one of the objects within the object rawdata is 'env'. 'env' itself is a compound variable that contains
three sub-variables that contain the actual raw data
> ls(rawdata@env)

#to show part of the raw data (e.g., binned profile) use
> head(rawdata@env$profile)
```

#compare the dimensions of this profile with the output of xcmsRaw
```
> dim(rawdata@env$profile)
> rawdata
```

#plot the total ion current and explain what you see.
```
> plotTIC(rawdata)
```

#visualize the mass trace that was measured in the 601th scan (time point) and explain what you see
```
> plotScan(rawdata,scan=601,mzrange=c(200,600))
```

#visualize the chromatogram based on m/z = 496.2 and explain what you see
```
> plotChrom(rawdata, fitgauss = FALSE, mzrange=c(496.2, 496.2))
```

1.4.2 Peak identification

The class of objects used for pre-processing analyte data from multiple LC/MS files is xcmsSet. The class xcmsSet can be seen as a container that stores data and results, and provides the methods for dealing with them. It stores peak lists and provides methods for grouping and aligning those peaks. To create an xcmsSet object from a set of data files, use the xcmsSet() constructor function. It handles peak detection and generation of the xcmsSet object.

The peak identification method that we will use is 'matched filtration'. This methods combines noise reduction and peak identification in one algorithm.

Question 2. Peak identification

a. Use matched filtration for peak detection. How many peaks are identified in each sample?

b. What are the mass and time ranges of the data set? How many peaks are identified in total? What do these peaks represent? Did the method detect the same number of peaks in every sample? Note: you can neglect the Peak Groups, and profile settings for the time being

The R code to obtain the answer for question (a).

Note that xcmsSet is configured to use 'matched filtration' for peak detection. This function uses many parameters that influence the outcome of peak detection. Below are the default values that we will use in this exercise. Don't worry if you don't understand the meaning of all these parameters but you should be able to understand several of them. Try to give a brief description for each of the following parameters.

sigma=fwhm/2.3548
```
> fwhm<-30;
> sigma<-fwhm/2.3548
```

maximum number of peaks per extracted ion-chromatogram

```
> max <-5;

# signal to noise cutoff
> snthresh<-10;

# step size to use for profile generation
> step <-0.1

#number of steps to merge prior to matched filtration
> steps<-2

# minimum difference in m/z for peaks with overlapping retention times
> mzdiff<-0.8-steps*step

#"positive"  "negative"
> polarity <-NULL;
> profmethod <-"bin"

#Now you can use all of these parameters to identify peaks in the LC-MS data:
> xset <-xcmsSet(cdffiles,
                 profmethod = profmethod,
                 polarity = polarity,
                 fwhm=fwhm,
                 sigma=sigma,
                 max=max,
                 snthresh=snthresh,
                 step=step,
                 steps=steps,
                 mzdiff=mzdiff,
                 sleep=0,
                 method="matchedFilter")
```

Next you have to write a few lines of R code to retrieve the number of peaks per sample. See if you understand this R code:

```
#Number of peaks per sample
for(i in c(1:12)) {
  p=subset(peaks(xset), peaks(xset)[,"sample"] == i)
  cat("sample ",i,": ",dim(p)[1],"\n")
}
```

The code to obtain the answer for question (b)

```
> xset
```

1.4.3 Matching peaks across samples

After peak identification, peaks representing the same analyte across samples must be placed into groups to allow comparison (e.g., statistical analysis) of the compounds. This is accomplished with the 'group' method, which returns a new xcmsSet object with the additional group information. The grouping algorithm processes the peak

lists in order of increasing mass and will regularly output the mass it is currently working on.

Question 3. Matching peaks across samples

a. Use the group method to identify peaks representing the same analyte across samples. How many peak groups can be obtained for this data? What does this mean?

b. Have a closer look at the peak groups that are generated. How is the first peak group defined in terms of m/z range, rt-range, and number of samples in which this peak group occurs.

The R code to obtain the answer for question (a)

```
> xset<-group(xset)
> xset
```

#Note: if you provide the following additional argument to the function group then you get graphical output showing the wild type and KO peaks (black and red dots) that were grouped together

```
> xset<-group(xset, sleep=1)
```

The R code to obtain the answer for question (b)

#have a look at the first 5 peaks that were identified (or use peaks(xset) to show them all)

```
> head(peaks(xset),5)
```

#have a look at the first 5 groups that were identified (or use groups(xset) to show them all)

```
> head(groups(xset),5)
```

1.4.4 Retention time correction

After matching peaks into groups, xcms can use those groups to identify and correct drifts in retention time from run to run. The aligned peaks can then be used for a second pass of peak grouping which will be more accurate than the first. The whole process can be repeated in an iterative fashion, although we will only demonstrate a single pass of retention time alignment here.

Not all peak groups will be helpful for identifying retention time drifts. Some groups may be missing peaks from a large fraction of samples and thus provide an incomplete picture of the drift at that time point. Still others may contain multiple peaks from the same sample, which is a sign of improper grouping. xcms ignores those groups by only considering **"well-behaved" peak groups** which are missing at most one sample and have at most one extra peak. These values can be changed with the 'missing' and 'extra' arguments. For each of those well-behaved groups, the algorithm calculates a median retention time and, for every sample, a deviation from that median.

Retention time correction is performed by the retcor method, which returns an

xcmsSet object with corrected retention times. Because it changes the retention times of all peaks, it is important to store the new object under a new variable name. That will allow you to backtrack and repeat retention time correction if necessary.
After retention time correction, the initial peak grouping becomes invalid and is discarded. Therefore, the resulting object needs to be regrouped.

Question 4. Retention time correction

a. Perform a retention time correction and new peak groups. How many retention time correction groups were used? What does this mean?

b. The retention time correction produces a graph. Explain this figure (you can neglect the lower panel of this figure).

c. Perform a second iteration of grouping. How many peaks groups are identified (after retention time correction)? Why do we have fewer groups compared to the grouping prior to retention time correction?

The code to obtain this answer:

```
> xset2 <-retcor(xset, missing=1, extra=1, smooth="loess", family=
"symmetric", plottype = "mdevden")

#If you inspect the content of xset2 then you will see the raw and corrected retention times
> str(xset2)

#note: next a regroup is necessary. Note that the inclusiveness is decreased by using the bw
argument.
> xset2 <-group(xset2,bw=10)
```

1.4.5 Filling in missing peak data

After the second pass of peak grouping, there will still be peak groups which are missing peaks from some of the samples. That can occur because peaks were missed during peak identification or because an analyte was not present in a sample. In any case, those missing data points can be filled in by rereading the raw data files and integrating them in the regions of the missing peaks. That is performed using the fillPeaks method, which returns an xcmsSet object with the filled in peak data. While running, it outputs the name of the sample it is currently processing.

Question 5. Filling in missing peak data

a. Fill in the missing peaks.

b. How many peaks were missing?

c. Find the missing peaks in the peak table. What do you notice?

The code to obtain this answer:

```
> xset3 <-fillPeaks(xset2)
> xset3
```

You can inspect the first few lines of the peak table:

```
> head(peaks(xset3))
```

You can also specifically identify the peaks that were added to the peak table

```
> head(peaks(xset3)[which(is.na(peaks(xset3)[,'sn'])),])
```

1.4.6 Analysing and visualizing results

We have now identified all peaks for all samples, aligned the chromatograms to allow comparison of the samples, and have added missing peaks. Now we are ready to compare the WT and KO mice to find the most statistically significant differences in analyte intensities. This is done with the diffreport method. It will automatically generate extracted ion chromatograms for a given number of them (in this case 10).

Question 6. Identify metabolite differences between WT and KO mice

a. Determine which metabolites have different concentrations between WT and KO mice: open the file report.tsv in Excel. Give a description of the information that is in this file.

b. Have a look at the EICs and boxplots that are produced. What is an EIC? What is a boxplot? What do you notice from the EIC's?

c. Can you confirm that the peak of the first EIC corresponds to a N-acyl ethanolamine. What do you conclude from this?

The R code to obtain the answer to this question:

#Note: **replace the directory for fb with your own directory** where you want to store the results (e.g., C:\\Users\\AvK\\Desktop).

```
>fb="C:\\Users\\AvK\\Desktop"

>setwd(fb) #set working directory

>reporttab<-diffreport(xset3, filebase=paste(fb, "\\report",sep=""),
"WT", "KO", 10, metlin=0.15)

> write.table(reporttab, file="plist.txt", na="", quote=FALSE,
row.names=FALSE, dec=".",sep = "\t")
```

The 'diffreport' function produces a tab-delimited output file (report.tsv) that can be opened in Excel. In this example, it also produces also 10 EIC's and 10 boxplots of the most significant metabolites. These are automatically placed in two subdirectories: report_box and report_eic.

The Excel file contains the links to the Metlin database for all identified peaks. If these links do not work anymore then go to http://metlin.scripps.edu/ and try to 'Sign Up' for an account. This only takes 2 minutes to do. Once you have an account, you can login and search Metlin. Just select a few masses

from the Excel file and search Metlin manually. All data was acquired in positive ion mode and the neutral mass is calculated assuming all ions are M+H.

Alternatively, try identifying the mass using the Human Metabolome Database at http://www.hmdb.ca/. Use menu option Search >> MS search. Input mass 299.19, Ionization Neutral and Molecular Weight Tolerance 0.1 Da or use the mass for the positive ion given in the mzmed column: 300.19, Ionization Positive. Towards the end of the list of possible annotations you'll find Palmitoylethanolamine.

1.5 References

[1] Saghatelian et al (2004) Assignment of endogenous substrates to enzymes by global metabolite profiling, Biochemistry, 43, 14332.

[2] http://en.wikipedia.org/wiki/FAAH

[3] Colin Smith (2004) LC/MS Preprocessing and Analysis with xcms.

[4] C.A. Smith, E.J. Want, G.C. Tong, R. Abagyan, and G. Siuzdak (2006) XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. Analytical Chemistry, 78 (3), 779.

[5] xcms website: http://metlin.scripps.edu/xcms/

[6] xcms documentation: http://www.bioconductor.org/packages/release/bioc/html/xcms.html