

# LC-MS Peak Annotation and Identification with *CAMERA*

Carsten Kuhl, Ralf Tautenhahn and Steffen Neumann

October 24, 2023

## 1 Introduction

The R-package *CAMERA* is a **C**ollection of **A**lgorithms for **M**etabolite **p**rofile **A**nnotation. Its primary purpose is the annotation and evaluation of LC-MS data. It includes algorithms for annotation of isotope peaks, adducts and fragments in peak lists. Additional methods cluster mass signals that originate from a single metabolite, based on rules for mass differences and peak shape comparison [1]. To use the strength of already existing programs, *CAMERA* is designed to interact directly with processed peak data from the R-package *xcms* [2].

Based on this annotation results, the molecular composition can be calculated if the mass spectrometer has a high-enough accuracy for both the mass and the isotope pattern intensities in combination with the R-package *Rdisop*

## 2 Short Background

For soft ionisation methods such as LC/ESI-MS, different kind of ions besides the protonated molecular ion occurs. These are adducts (e.g.  $[M + K]^+$ ,  $[M + Na]^+$ ) and fragments (e.g.  $[M - C_3H_9N]^+$ ,  $[M + H - H_2O]^+$ ). Depending on the molecule having an intrinsic charge,  $[M]^+$  may be observed as well. In most cases a substance generates a bulk of different ions. There interpretation is time consuming, especially if substances co-elute. Therefore deconvolution, which separates the different substances and discovery of the ion species is necessary.

Solving these problems with *CAMERA* is demonstrated in the next chapters.

## 3 Processing with *CAMERA*

### 3.1 Preprocessing with *xcms*

CAMERA needs as input an *xcmsSet* object that is processed with your favorite parameters. For an example see below:

```
library(CAMERA)
#Single sample example
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file,method="centWave",ppm=30,peakwidth=c(5,10))

#Multiple sample
library(faahK0)
filepath <- system.file("cdf", package = "faahK0")
xsg <- group(faahko)
xsg <- fillPeaks(xsg)
```

After the preprocessing we create an CAMERA object, which is called *xsAnnotate* or in short *xsa*.

```
library(CAMERA)
xsa <- xsAnnotate(xs)
```

Depending on your analysis the upcoming workflows may differ at this point and we start with the description of the annotation workflow. Afterwards we demonstrate the wrapper functions and how to interpret the results.

### 3.2 Annotation Workflow

The CAMERA annotation procedure can be split into two parts: We want to answer the questions which peaks occur from the same molecule and secondly compute its exact mass and annotate the ion species. Therefore CAMERA annotation workflow contains following primary functions:

1. peak grouping after retention time (`groupFWHM`)
2. peak group verification with peakshape correlation (`groupCorr`)

Both methods separate peaks into different groups, which we define as "pseudospectra". Those pseudospectra can consists from one up to 100 ions, depending on the molecules amount and ionizability. Afterwards the exposure of the ion species can be performed with:

1. annotation of possible isotopes (`findIsotopes`, `findIsotopesWithValidation`)
2. annotation of adducts and calculating hypothetical masses for the group (`findAdducts`)

This workflow results in a data-frame similar to a *xcms* peak table, that can be easily stored in a comma separated table (Excel-readable).

The next section shows some practical examples.

### 3.2.1 Working with single sample

Let's come to the practical work. Here we have a single sample file either in positive or negative ionization mode. The *xcmsSet* was created as shown in section 3.1.

```
# Create an xsAnnotate object
an  <- xsAnnotate(xs)
# Group based on RT
anF <- groupFWHM(an, perfwHM = 0.6)
# Annotate isotopes
anI <- findIsotopes(anF, mzabs = 0.01)
# Verify grouping
anIC <- groupCorr(anI, cor_eic_th = 0.75)
#Annotate adducts
anFA <- findAdducts(anIC, polarity="positive")
anFA
```

In the above example, we create the pseudospectra according to the peak retention time information. The *perfwHM* parameter defines the window width, which is used for matching. Lower it for a smaller windows or set it to a higher value, if the retention time varies. This step generate 14 pseudospectra.

Afterwards we annotate isotopic peaks, with *mzabs* as allowed m/z error. In this example we find 33 isotope peaks, which is the number of  $[M + 1]$ ,  $[M + 2]$ , ... ions.  $[M + 1]$  is the first isotopic peak for the monoisotopic peak  $[M]$ . This isotope informations are useful in the next step, where for every peak in one pseudospectra a pairwise EIC correlation is performed. If the correlation value between two peaks is higher than the threshold *cor\_eic\_th* it will stay in the group, otherwise both are separated. If the peaks are annotated isotope ions, they will not be divided. This separates our 14 pseudospectra into 48.

After the second pseudogroup creating step we now finally do a complete annotation of adducts. Therefore, the *polarity* parameter must be set. The final output of the `xsAnnotate` object shows all important information. For further processing we export the results with:

```
peaklist <- getPeaklist(anFA)
write.csv(peaklist, file='xsannotated.csv')
```

where `file` is the output filename.

That's all for the simple sample approach. Please note that every method has additional parameters, that are not explicitly mentioned here. Also if your analysis doesn't need annotations, only a separation into groups, then simply stop after `groupCorr`. The grouping results are stored in the list `object@pspectra`, which saves the peak indices as elements for every group.

```
# anIC here is the result of groupCorr
peak.idx <- anIC@pspectra[[1]]
#print the indices of all peaks from pseudospectrum 1
cat(peak.idx)
```

### 3.2.2 Working with multiple samples

In this case we have a multiple samples experiment like replicates of one probe or a wildtype vs. mutant experiment. As in the previous example, we start with the already processed `xcmsSet`-object. Note: If you want an `diffreport` later on, make sure you run `fillPeaks` on your `xcmsSet` before. As test dataset we use here the `faahKO`. For more information about the dataset see <http://dx.doi.org/10.1021/bi0480335>. CAMERA contains different approaches with multiple sample analysis. Here we only show the most common way, for the other strategies see the `xsAnnotate` manpage, especially the parameter `sample`.

```
#Create an xsAnnotate object
xsa <- xsAnnotate(xsg)
#Group after RT value of the xcms grouped peak
xsaF <- groupFWHM(xsa, perfwhm=0.6)
#Verify grouping
xsaC <- groupCorr(xsaF)
#Annotate isotopes, could be done before groupCorr
xsaFI <- findIsotopes(xsaC)
#Annotate adducts
xsaFA <- findAdducts(xsaFI, polarity="positive")
```

```
#Get final peaktable and store on harddrive
write.csv(getPeaklist(xsaFA),file="result_CAMERA.csv")
```

Similar to the single sample experiment we group according to retention time, group according to EIC correlation and finally annotate isotopes and adducts. The main difference from our first example is the sample selection. The sample parameter sets either one specific sample (sample=x), a subset (sample=c(x:y)), or all samples (sample=c(1:nSamples)), where nSamples is the number of samples in the xcmsSet, as selection for the correlation analysis. Since the runtime increases with every sample, CAMERA includes a automatic selection method (sample=NA), where in groupFWHM a representative sample is chosen for every created pseudospectrum. The automatic selection is also the default value.

### 3.3 Interpretation of the Results

id	mz	rt	isotopes	adduct	pc
65	176.04	280.09			4
76	136.05	280.43	[14][M+1]1+		5
77	135.05	280.43	[14][M]1+		5
74	153.06	280.43		[M+H]+ 152.05437	5
75	175.04	280.43		[M+Na]+ 152.05437	5
73	197.02	280.76		[M+2Na-H]+ 152.05437	5
78	377.74	286.15			6
79	732.5	286.49			6
83	488.32	286.82		[M+Na]+ 465.33205	7
82	466.34	286.82		[M+H]+ 465.33205	7
...					

Table 1: Example of annotation result for one sample. Columns with intensity values are omitted. blue-line: annotated group 5, red-line: annotated group 7

Table 1 shows an cutout example of an annotation result. The columns with the intensity values are omitted and the rows are ordered by their rt values for better readability. The column *pc* shows the result of the peak correlation based annotation (independent of the annotations *iso* and *adduct*). Peaks with the same label are supposed to belong to the same spectrum. The column *adduct* shows the annotation hypotheses for the ions species. The value after the brackets is the estimated molecular mass.

The column *isotopes* contains the annotated isotopes for a monoisotopic peak. The values in the first square brackets denote the isotope-group-id(column *id*), the second is the isotope annotation and the number after the brackets is the charge of the isotope.

## 4 Wrapper functions and combination with diffreport

### 4.0.1 The function annotate

`annotate` is a wrapper function for the annotation workflow. It's similar to `annotateDiffreport`, but doesn't combine the results with those from the `diffreport`. With the parameter *pval\_list* a handmade preselection of pseudospectra is possible. A "quick" mode is also available, that runs only `groupFWHM` and `findIsotopes`. The normal mode runs `groupFWHM`, `findIsotopes`, `groupCorr` and `findAdducts` in order as mentioned. Every parameter of these functions also work with `annotate`. As a small example:

```
#A full annotation run
xsa <- annotate(xs, perfwhm=0.7, cor_eic_th=0.75,
ppm=10, polarity="positive")
#Generate result
peaklist <- getPeaklist(xsa)
#Save results
write.csv(peaklist,file="results.csv")
```

Similar to the previous example, the grouping is followed by an annotation. But in contrast we now have additional summaries respectively analysis functions. For a comparison and statistical analysis between different sample classes, *xcms* contains the `diffreport` function. CAMERA can use this method for a better representation.

```
#Run fillPeaks on xcmsSet
xsg.fill <- fillPeaks(xsg)
#Make a diffreport with CAMERA result
diffreport <- annotateDiffreport(xsg.fill)
#Save on harddrive
write.csv(diffreport, file="diffreport.csv")
```

The `annotateDiffreport` is a wrapper for the *xcms* `diffreport` function and combines the results from CAMERA. The resulting table has three different columns, see section 3.3. For a speed up it's possible to preselect

pseudospectra or make an automatic selection based on the `diffreport` result. As an example that selects only groups with a fold change higher than 4.

```
#Example 1 with creating list of interest from grouped xcmsSet
diffreport <- annotateDiffreport(xsg.fill, quick=TRUE)
#Save results
write.csv(diffreport, file="diffreport.csv")
#Look into the table and select interesting pseudospectra
#e.g. pseudospectra 10,11 and 30
psg_list <- c(10,11,30)
diffreport.annotated <- annotateDiffreport(xsg.fill, psg_list=psg_list,
polarity="positive")

#Example 2 with automatic selection
diffreport.annotated <- annotateDiffreport(xsg.fill, fc_th=4,
polarity="positive")
```

Both examples generate a data-frame, identical to the normal `diffreport` result, but now with three additional result columns from CAMERA. In example 1 we perform a quick-run, that means we only generate the `xsAnnotate` object and call `groupFWHM` and `findIsotopes`. From these results we pre-select 3 pseudospectra (10,11,30), taken from the column `pc`. In the next run, we run `annotateDiffreport` again with our list as parameter. An annotation will only be done for these three groups. In example 2 we perform an automatic preselection, where the `fc_th` parameter defines a threshold for selecting groups, which contains ions with a fold change higher than four. For other pseudospectra, no adduct annotation will be calculated. The fold change value is taken from the `diffreport` result. For other parameters see the manpage of `annotateDiffreport`.

## 4.1 Visualisation of the Results

For a graphical presentation of the annotation result CAMERA provides the function `plotEICs` to visualize the raw data and the function `plotPsSpectrum` to plot all peaks of a pseudospectrum. The next example show the use of both functions.

```
> library(CAMERA)
> file <- system.file('mzML/MM14.mzML', package = "CAMERA")
> xs <- xcmsSet(file, method="centWave",ppm=30, peakwidth=c(5,10))
> an <- xsAnnotate(xs)
> an <- groupFWHM(an)
```

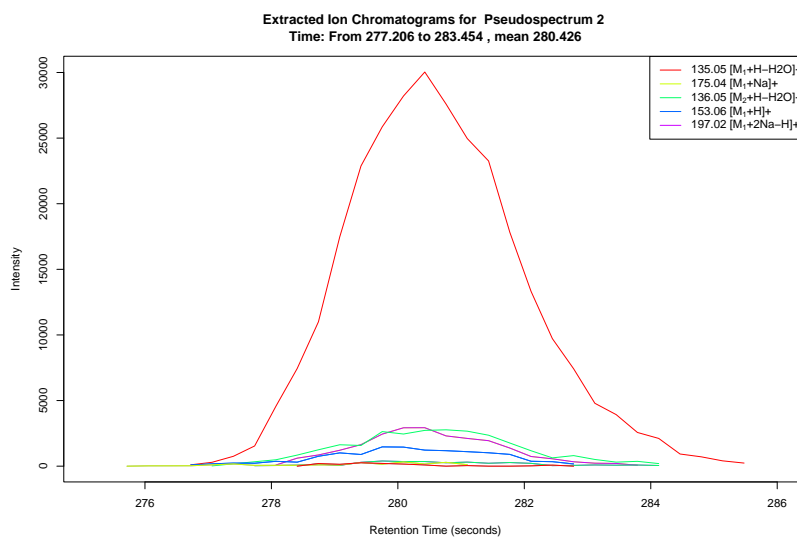


Figure 1: EICs.

```
> an <- findAdducts(an, polarity="positive")
> plotEICs(an, pspec=2, maxlabel=5)
```

Figure 1 displays the EICs of all peaks from one pseudospectrum. With this plot you can manual check if the grouping makes sense. In Figure 2 you see a typical  $m/z$  plot, with labelled, annotated peaks.

```
> plotPsSpectrum(an, pspec=2, maxlabel=5)
```

## 5 Function Overview

This section contains for every CAMERA function a small introduction with an example. See the manpages for further informations.

### 5.1 Function annotate

Wrapper function for the whole annotation workflow. Returns a `xsAnnotate` object. It handles also all functions parameters.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
xsa <- annotate(xs)
```



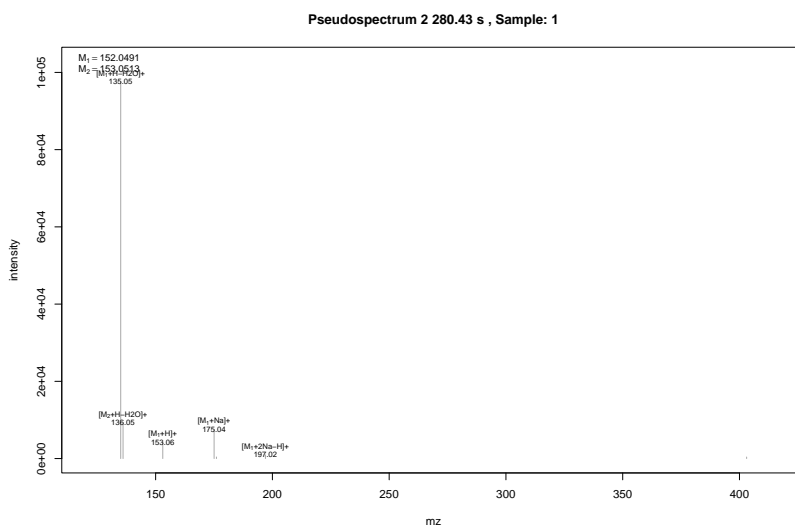


Figure 2: Spectrum.

## 5.2 Function annotateDiffreport

Wrapper function for the `xcms diffreport` and the `annotate` function. Returns a diffreport with the results from CAMERA's annotation progress. It handles also all functions parameters.

```
library(CAMERA)
library(faahKO)
xs.grp      <- group(faahko)
xs.fill     <- fillPeaks(xs.grp)
diffreport  <- annotateDiffreport(xs.fill)
write.csv(diffreport, file="...")
```

The combination of the diffreport and the CAMERA result can also be done without these functions. Therefore diffreports `sortpval` argument must set to FALSE. After the combination the sorting after the pvalue can be restored.

```
diffrep     <- diffreport(...)
xsa.peaklist <- getPeaklist(xsa)
diffrep.new <- cbind(diffrep, xsa.peaklist[, c("isotopes", "adduct",
"pcgroup")])
#Sort after pvalue
diffrep.new <- diffrep.new[order(diffrep.new[, "pvalue"]),]
```

### 5.3 Function findAdducts

After the peak grouping into pseudospectra with `groupCorr` or `groupFWHM` the resulting `xsAnnotate` can be processed with `findAdducts`. For every pseudospectra all possible adducts are calculated based on a provided rule table. As default CAMERA calculate its own table, which contains every possible combination from the standard ions H, Na, K, NH4 and Cl, depending on your ionization mode.

Additional CAMERA contains four precalculated rule tables: `primary_adducts_pos`, `primary_adducts_neg`, `extended_adducts_pos`, `extended_adducts_neg`. Those can be applied as shown in the example. For creating your own rule table, see 6.

```
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs   <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an   <- xsAnnotate(xs)
an   <- groupFWHM(an)
an   <- findIsotopes(an) # optional but recommended.
an   <- findAdducts(an, polarity="positive")

#With provided rule table
file <- system.file('rules/primary_adducts_pos.csv', package = "CAMERA")
rules <- read.csv(file)
an    <- findAdducts(an, polarity="positive", rules=rules)
```

### 5.4 Function findIsotopes

For a provided `xsAnnotate`, CAMERA can identify isotope peaks within every pseudospectra. The function `findIsotopes` takes as parameter *max-charge* and *maxiso* that controls the maximum number of the expected isotopes within one cluster and their charges.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs   <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an   <- xsAnnotate(xs)
an   <- groupFWHM(an)
an   <- findIsotopes(an)
```

### 5.5 Function findIsotopesWithValidation

For a provided `xsAnnotate`, CAMERA can identify isotope peaks within every pseudospectra. In particular, putative isotope clusters are validated

and deconvoluted based on database statistics of the KEGG database. The function `findIsotopesWithValidation` takes as parameter *maxcharge* and *maxiso* that controls the maximum number of the expected isotopes within one cluster and their charges.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopesWithValidation(an)
```

## 5.6 Function `findNeutralLoss`

A common strategy to identify interesting compounds is the screening after specific neutral losses. CAMERA adopts this strategy and provides with `findNeutralLoss` and `findNeutralLossSpecs` an interface for scanning every pseudospectrum for neutral losses. The difference between both methods is in the results. `findNeutralLossSpecs` returns an artificial `xcmsSet` containing the peaks, which have the neutral loss.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
xs.pseudo <- findNeutralLoss(an,mzdiff=18.01,mzabs=0.01)
#Searches for Peaks with water loss
xs.pseudo@peaks #show Hits
```

## 5.7 Function `findNeutralLossSpecs`

`findNeutralLossSpecs` returns a boolean vector for every pseudospectrum, where a hit is marked with `TRUE`.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
hits <- findNeutralLossSpecs(an,mzdiff=18.01,mzabs=0.01)
#Searches for pseudospectra with water loss
```

## 5.8 Function `getIsotopeCluster`

This method extracts the isotope annotation from a `xsAnnotate` object. The order of the resulting list correspond to those from the whole peaklist.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an)
isolist <- getIsotopeCluster(an)
isolist[[10]] #get IsotopeCluster 10
```

See the manpage for an example interaction with *Rdisop* to calculate the molecular composition.

## 5.9 Function `getPeaklist`

This function returns a peaklist containing all information from an `xsAnnotate` object. This peaklist can be directly saved as an csv file.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
xsa <- xsAnnotate(xs)
xsa <- groupFWHM(xsa)
xsa <- findIsotopes(xsa)
xsa <- findAdducts(xsa, polarity="positive")
peaklist <- getPeaklist(xsa)
write.csv(peaklist,file="...")
```

## 5.10 Function `getpspectra`

This function returns for a provided pseudospectrum index its peaktable and CAMERA's annotation information. This peaktable can be directly saved as an csv file. Note: The indexes for the isotopes, are those from the whole peaklist. See 5.9.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
```

```
xsa <- xsAnnotate(xs)
xsa <- groupFWHM(xsa)
psp.peaks <- getpspectra(xsa, 1)
psp.peaks
```

## 5.11 Function groupCorr

This function calculates the pearson correlation coefficient based on the peak shapes of every peak in the pseudospectrum to separate co-eluted substances. It's recommended to use groupFWHM before, otherwise the runtime is very long!!

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
xsa <- xsAnnotate(xs)
xsa <- groupFWHM(xsa)
xsa <- groupCorr(xsa)
```

## 5.12 Function groupFWHM

For grouping peaks into pseudospectra, this function uses the retention time information. Every peaks that falls into a defined window are considered as one group. The window is defined as a percentage of the peak FWHM around the RT\_med value.

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
xsa <- xsAnnotate(xsa)
xsa <- groupFWHM(xsa)
```

## 5.13 Function plotEICs

This method returns a batch plot including the extracted ion chromatograms to the current graphics device for a provided pseudospectrum.

```
#Plot all peak EICs of pseudospectrum 1
plotEICs(xsa,1)
```

## 5.14 Function plotPsSpectrum

This method plots the spectrum of a pseudospectrum, with labelling the most intense peaks.

```
#Plot the spectrum of pseudospectrum 1 and highlight the
#annotation and mz labels of the 10 strongest peaks
plotPsSpectrum(xsa,1,maxlabel=10)
```

## 6 Create rule table

As starting point for creating a specific rule table CAMERA provides four rule tables with primary adducts for positive and negative mode. The saving path can be found in R, see below.

```
file1 <- system.file('rules/primary_adducts_pos.csv', package = "CAMERA")
file2 <- system.file('rules/primary_adducts_neg.csv', package = "CAMERA")
file3 <- system.file('rules/extended_adducts_pos.csv', package = "CAMERA")
file4 <- system.file('rules/extended_adducts_neg.csv', package = "CAMERA")
```

Those files can be edited in every csv editor (e.g. Excel). The rule table has 7 columns. name: adduct name nmol: Number of molecules (xM) included in the molecule charge: charge of the molecule massdiff: mass difference without calculation of charge and nmol (CAMERA will do this automatically) oidscore: adduct index. Molecules with the same kations (anions) configuration and different nmol values have the same oidscore. For example [M+H] and [2M+H] quasi: Every annotation with belong to one molecule is called annotation group, for example [M+H] and [M+Na] where M means the same molecule. A annotation group must include at least one ion with quasi set to 1 for this adduct. If a annotation group only includes optional adducts (rule set to 0) then this group is excluded. To disable this reduction, set all rules to 1 or 0. ips: Rule score. If one peak can be explained with more than one annotation group, then only this group survive, with has the higher score (sum of all annotation). This decreases the number of false positive greatly, but the optional settings can differ in each machine.

After creation of your own rule set, use it as parameter rules in findAdducts see 5.3.

## References

- [1] Ralf Tautenhahn, Christoph Böttcher, Steffen Neumann : Annotation of LC/ESI-MS Mass Signals, BIRD 2007 Proc. of BIRD 2007 – 1st

International Conference on Bioinformatics Research and Development,  
2007. <http://www.springerlink.com/content/4731404001787974/>  
and <http://msbi.ipb-halle.de/~rtautenh/bird07.pdf>

- [2] Smith, C.; Want, E.; O'Maille, G.; Abagyan, R.; Siuzdak, G. *Anal Chem*  
**2006**, *78*, 779–787