

Grouping FTICR-MS data with xcms

J. Bargsten

October 14, 2013

Introduction

This document describes how to use *xcms* for aligning multiple MS spectra against each other.

1 Prerequisites

Lots of Preprocessing has to be done before the data is ready for aligning. First of all *xcms* and *MassSpecWavelet* are needed for further processing.

```
> library(xcms)
> library(MassSpecWavelet)
```

This documentation uses raw mzdata files from *msdata* as example data set. Assuming that *msdata* is installed, we locate the path of the package and extract the datafiles.

```
> library(msdata)
> mzdatapath <- system.file("fticr", package = "msdata")
> mzdatafiles <- list.files(mzdatapath, recursive = TRUE, full.names = TRUE)
> cat("Starting xcmsDirect.Rnw")
```

Starting xcmsDirect.Rnw

The *xcmsSet*-Constructor parses the given files and applies peakpicking using the MassSpecWavelet algorithm, leading to a *xcmsSet* object with 2 sampleclasses, ham4 and ham5, and 5 samples, respectively.

```
> data.mean <- "data.mean"
> xs <- xcmsSet(
+   method="MSW",
+   files=mzdatafiles,
+   scales=c(1,4,9),
```

```
+      nearbyPeak=T,
+      verbose.columns = FALSE,
+      winSize.noise=500,
+      SNR.method="data.mean",
+      snthr=10
+ )
```

2 Calibration

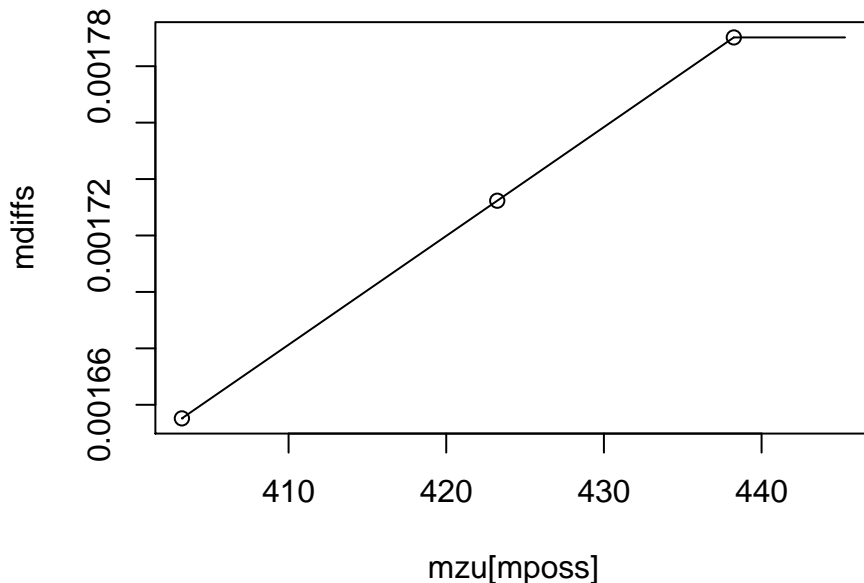
calibrate can be used to correct the m/z values in a `xcmsSet`. It needs a `xcmsSet` and a list of m/z value which should be found in the object. To show this on a example a sample of ham4 is created and discalibrated a bit after getting some m/z:

```
> xs4 <- xcmsSet(
+      method = "MSW",
+      files = mzdatafiles[1],
+      scales = c(1,4, 9),
+      nearbyPeak = T,
+      verbose.columns = FALSE,
+      winSize.noise = 500,
+      SNR.method = "data.mean",
+      snthr = 10)

> masslist <- xs4@peaks[c(1,4,7),"mz"]
> xs4@peaks[, "mz"] <- xs4@peaks[, "mz"] + 0.00001*runif(1,0,0.4)*xs4@peaks[, "mz"] + 0.
```

The `xcmsSet` now can be calibrated again with the m/z from the `masslist`. The plot shows the reference masses with the distances to the found ones and the regression-line.

```
> xs4c <- calibrate(xs4,
+      calibrants=masslist,
+      method="edgeshift",
+      mzabs=0.0001,
+      mzppm=5,
+      neighbours=3,
+      plotres=TRUE
+ )
```



The method "shift" adds a value to each m/z, "linear" does a regression and edgshift does a regression but uses a shift before the smallest and after the biggest m/z from the calibrants.

These steps are necessary to create a usable input for *mzClust*. However, if you have already stored the data in a *xcmsSet*, you can skip the steps above.

3 Aligning

Now we can align *xs* with *mzClust*. The result is a clone of *xs* enhanced by the result of *mzClust*. For a description of the arguments *mzClust* takes, see helppage of the function.

```
> xsg <- group(xs, method="mzClust")
```

```
1.14  6.82  17.05  21.59  27.27  31.82  37.50  38.64  42.05  47.73  53.41  54.55  61.36
```

```
> xsg
```

An "xcmsSet" object with 10 samples

Time range: -1--1 seconds (0-0 minutes)

Mass range: 400.1046-445.2931 m/z

Peaks: 88 (about 9 per sample)

Peak Groups: 18

Sample classes: ham4, ham5

Profile settings: method = bin
step = 0.1

Memory usage: 0.0177 MB

mzClust stores the grouping information like the standard *group* method of *xcms* suited for retrieval via *groups* and *groupidx*. An example is shown below.

```
> groups(xsg)[1:10,]
```

	mzmed	mzmin	mzmax	rtmed	rtmin	rtmax	npeaks	ham4	ham5
[1,]	402.2854	402.2851	402.2859	-1	-1	-1	5	0	5
[2,]	403.2365	403.2357	403.2367	-1	-1	-1	9	5	4
[3,]	405.1089	405.1087	405.1095	-1	-1	-1	4	0	4
[4,]	409.1844	409.1837	409.1845	-1	-1	-1	5	5	0
[5,]	410.1444	410.1440	410.1448	-1	-1	-1	4	0	4
[6,]	413.2672	413.2669	413.2677	-1	-1	-1	5	5	0
[7,]	423.2374	423.2363	423.2398	-1	-1	-1	3	3	0
[8,]	424.1611	424.1606	424.1615	-1	-1	-1	5	0	5
[9,]	425.1346	425.1344	425.1353	-1	-1	-1	5	0	5
[10,]	427.2681	427.2679	427.2681	-1	-1	-1	6	5	1

```
> peaks(xsg)[groupidx(xsg)[[1]]]
```

```
[1] 402.2851 402.2851 402.2851 402.2859 402.2859
```

4 Postprocessing

In most cases not all samples are in one group. This can be the origin of serious problems in code, which is based on e.g. *groupval*. *groupval* sets missing peaks to NA. The solution is *fillPeaks*. It changes all NA values to random noise based on the raw data file.

```
> groupval(xsg)[1,]
```

HAM004_641fE_14-11-07--Exp1.extracted	HAM004_641fE_14-11-07--Exp2.extracted
NA	NA
HAM004_641fE_14-11-07--Exp3.extracted	HAM004_641fE_14-11-07--Exp4.extracted
NA	NA
HAM004_641fE_14-11-07--Exp5.extracted	HAM005_641fE_14-11-07--Exp1.extracted
NA	37

```

HAM005_641fE_14-11-07--Exp2.extracted HAM005_641fE_14-11-07--Exp3.extracted
                                49                                60
HAM005_641fE_14-11-07--Exp4.extracted HAM005_641fE_14-11-07--Exp5.extracted
                                70                                80

```

```
> xsgf <- fillPeaks(xsg, method="MSW")
```

```

HAM004_641fE_14-11-07--Exp1.extracted HAM004_641fE_14-11-07--Exp2.extracted HAM004_641fE_14-11-07--Exp3.extracted

```

```
> groupval(xsgf, "medret", "into")[1:10,]
```

```

      HAM004_641fE_14-11-07--Exp1.extracted
402.3/-1      768754.0
403.2/-1      4735257.5
405.1/-1      761632.1
409.2/-1      4158404.5
410.1/-1      726003.9
413.3/-1      6099006.3
423.2/-1      2708391.1
424.2/-1      772516.1
425.1/-1      885238.7
427.3/-1      6302089.0
      HAM004_641fE_14-11-07--Exp2.extracted
402.3/-1      1230140.4
403.2/-1      6202417.6
405.1/-1      491944.3
409.2/-1      5004546.3
410.1/-1      532868.8
413.3/-1      4950641.7
423.2/-1      1801494.2
424.2/-1      521511.4
425.1/-1      948516.6
427.3/-1      5884065.2
      HAM004_641fE_14-11-07--Exp3.extracted
402.3/-1      810120.4
403.2/-1      6117414.1
405.1/-1      650391.6
409.2/-1      4403588.2
410.1/-1      1182671.8
413.3/-1      5517709.5
423.2/-1      2826896.2
424.2/-1      347349.0
425.1/-1      376380.5

```

427.3/-1	5354053.7
HAM004_641fE_14-11-07--Exp4.extracted	
402.3/-1	568660.8
403.2/-1	5328574.1
405.1/-1	950315.1
409.2/-1	4336554.2
410.1/-1	805050.9
413.3/-1	5008541.7
423.2/-1	2427532.8
424.2/-1	481549.3
425.1/-1	961748.7
427.3/-1	5654936.8
HAM004_641fE_14-11-07--Exp5.extracted	
402.3/-1	572090.2
403.2/-1	6429028.9
405.1/-1	1452332.4
409.2/-1	4580892.8
410.1/-1	1128403.7
413.3/-1	4856606.0
423.2/-1	1856867.6
424.2/-1	499037.6
425.1/-1	662997.4
427.3/-1	5248273.8
HAM005_641fE_14-11-07--Exp1.extracted	
402.3/-1	4095293
403.2/-1	4811391
405.1/-1	2982453
409.2/-1	1196232
410.1/-1	2872023
413.3/-1	1786533
423.2/-1	1064349
424.2/-1	2995850
425.1/-1	4431535
427.3/-1	3761371
HAM005_641fE_14-11-07--Exp2.extracted	
402.3/-1	4804762.5
403.2/-1	2581183.1
405.1/-1	2268984.5
409.2/-1	1210941.2
410.1/-1	2133219.4
413.3/-1	1061103.2
423.2/-1	688353.7

424.2/-1	2556865.3
425.1/-1	3821099.0
427.3/-1	1456574.4
HAM005_641fE_14-11-07--Exp3.extracted	
402.3/-1	4657726.8
403.2/-1	2727237.5
405.1/-1	2971705.2
409.2/-1	544048.4
410.1/-1	2466625.6
413.3/-1	892797.1
423.2/-1	897205.4
424.2/-1	2567877.2
425.1/-1	4246330.7
427.3/-1	1196915.1
HAM005_641fE_14-11-07--Exp4.extracted	
402.3/-1	3755889.7
403.2/-1	2496858.9
405.1/-1	2291624.1
409.2/-1	1346778.3
410.1/-1	2980996.6
413.3/-1	982491.3
423.2/-1	1209006.2
424.2/-1	2857624.1
425.1/-1	2977003.1
427.3/-1	1227957.6
HAM005_641fE_14-11-07--Exp5.extracted	
402.3/-1	5265972
403.2/-1	2165162
405.1/-1	3009065
409.2/-1	1187547
410.1/-1	2296774
413.3/-1	1027673
423.2/-1	1136440
424.2/-1	2892810
425.1/-1	3301529
427.3/-1	1682024

The results are suited for instance for heatmaps, etc.

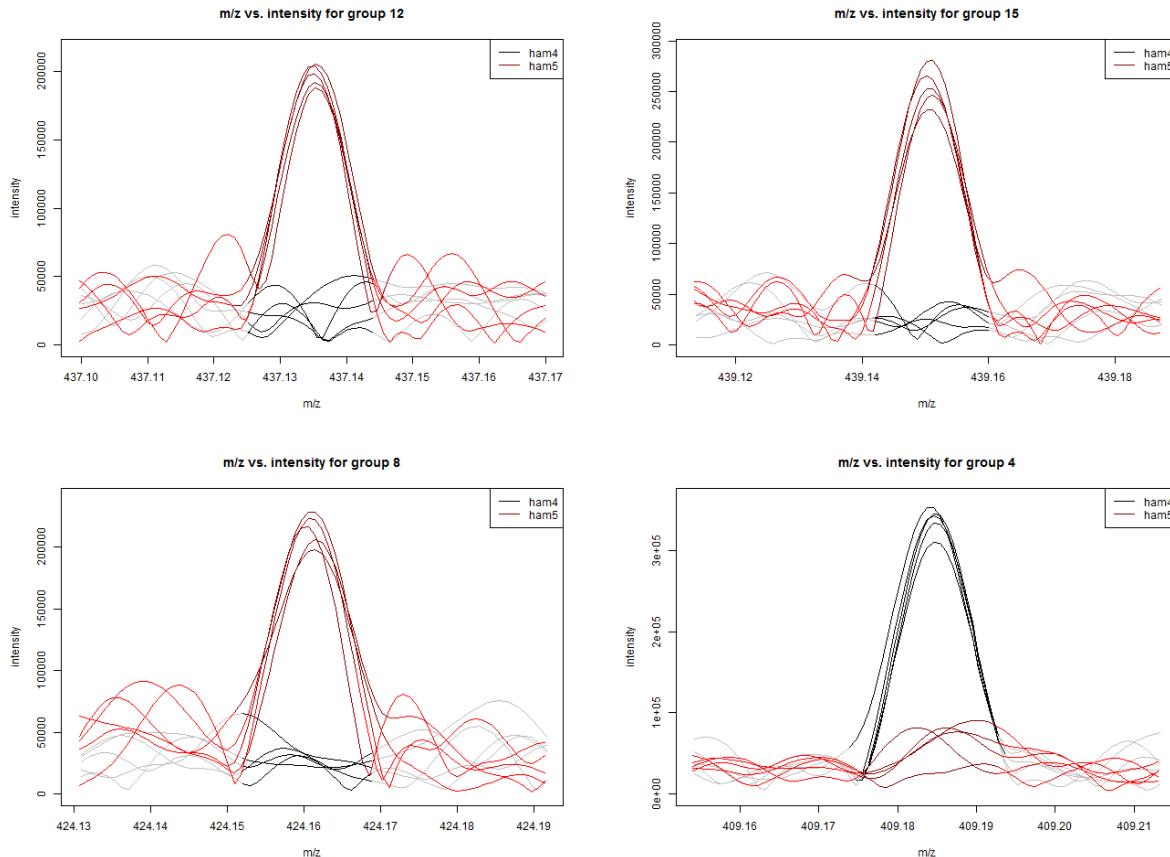


Figure 1: Auto-generated extracted spectra for the top three differentially regulated ions. Darkened lines indicate where the peaks were integrated for quantitation.

5 Analyzing and Visualizing Results

A report showing the most statistically significant differences in analyte intensities can be generated with the *diffreport* method. It will automatically show the superimposed peaks in the spectra for a given number of them, in this case 10. Several of those chromatograms are shown in Figure 1.

```
> reporttab <- diffreport(xsgf, "ham4", "ham5", "example", eicmax=4,
+                           h=480, w=640)
```

```
Processing data from sample:  1  2  3  4  5  6  7  8  9 10
```

```
group:  12 15  8  4
```

```
> reporttab[1:4,]
```


	name	fold	tstat	pvalue	mzmed	mzmin	mzmax	rtmed
1	M437T-1_1	5.144529	20.00921	4.258336e-08	437.1353	437.1353	437.1353	-1
2	M439T-1	6.877120	25.31711	4.886837e-08	439.1508	439.1503	439.1512	-1
3	M424T-1	5.290320	19.91077	8.868895e-08	424.1611	424.1606	424.1615	-1
4	M409T-1	4.098768	-16.86272	1.555750e-07	409.1844	409.1837	409.1845	-1
	rtmin	rtmax	npeaks	ham4	ham5	HAM004_641fE_14-11-07--Exp1.extracted		
1	-1	-1	4	0	4	665915.1		
2	-1	-1	5	0	5	595855.3		
3	-1	-1	5	0	5	772516.1		
4	-1	-1	5	5	0	4158404.5		
	HAM004_641fE_14-11-07--Exp2.extracted				HAM004_641fE_14-11-07--Exp3.extracted			
1					368064.3		303954.7	
2					645369.3		378272.6	
3					521511.4		347349.0	
4					5004546.3		4403588.2	
	HAM004_641fE_14-11-07--Exp4.extracted				HAM004_641fE_14-11-07--Exp5.extracted			
1					683672.7		511603.0	
2					320544.9		533959.4	
3					481549.3		499037.6	
4					4336554.2		4580892.8	
	HAM005_641fE_14-11-07--Exp1.extracted				HAM005_641fE_14-11-07--Exp2.extracted			
1					2619631		2432116	
2					3586827		3224767	
3					2995850		2556865	
4					1196232		1210941	
	HAM005_641fE_14-11-07--Exp3.extracted				HAM005_641fE_14-11-07--Exp4.extracted			
1					2470892.7		2826523	
2					3606573.9		3129954	
3					2567877.2		2857624	
4					544048.4		1346778	
	HAM005_641fE_14-11-07--Exp5.extracted							
1					2683008			
2					3465884			
3					2892810			
4					1187547			