

# Grouping FTICR-MS data with xcms

J. Bargsten

December 17, 2008

## 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)
```

```

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

```

## 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)

```

```

HAM004_641fE_14-11-07--Exp1.extracted:

```

```

> masslist <- xs4@peaks[c(1, 4, 7), "mz"]
> xs4@peaks[, "mz"] <- xs4@peaks[, "mz"] + 1e-05 * runif(1, 0,
+   0.4) * xs4@peaks[, "mz"] + 1e-04

```

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.

```

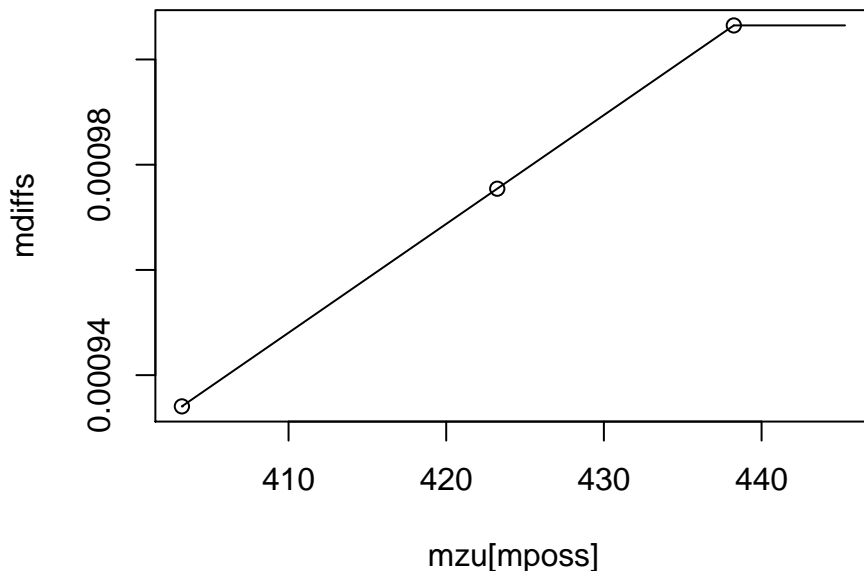
> xs4c <- calibrate(xs4, wishlist = masslist, method = "edgeshift",
+   mzabs = 1e-04, mzppm = 5, neighbours = 3, plotres = TRUE)

```

```

0.0009340734
0.0009754414
0.001006477

```



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 wishlist.

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: 0.0148 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

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

HAM004_641fE_14-11-07--Exp1.extracted	
402.3/-1	NA
403.2/-1	814693.1
405.1/-1	NA
409.2/-1	732119.9
410.1/-1	NA
413.3/-1	1018994.8
423.2/-1	435858.5
424.2/-1	NA
425.1/-1	NA
427.3/-1	1125644.3
HAM004_641fE_14-11-07--Exp2.extracted	
402.3/-1	NA
403.2/-1	1046733.1
405.1/-1	NA
409.2/-1	838774.7
410.1/-1	NA
413.3/-1	771793.6
423.2/-1	NA
424.2/-1	NA
425.1/-1	NA
427.3/-1	1007261.7
HAM004_641fE_14-11-07--Exp3.extracted	
402.3/-1	NA
403.2/-1	1063981.3
405.1/-1	NA
409.2/-1	802692.6
410.1/-1	NA
413.3/-1	842933.1
423.2/-1	411344.5
424.2/-1	NA
425.1/-1	NA
427.3/-1	901531.0
HAM004_641fE_14-11-07--Exp4.extracted	
402.3/-1	NA
403.2/-1	965765.8

405.1/-1	NA
409.2/-1	764068.8
410.1/-1	NA
413.3/-1	802110.5
423.2/-1	357822.2
424.2/-1	NA
425.1/-1	NA
427.3/-1	972574.2
HAM004_641fE_14-11-07--Exp5.extracted	
402.3/-1	NA
403.2/-1	1127939.6
405.1/-1	NA
409.2/-1	807672.1
410.1/-1	NA
413.3/-1	744445.3
423.2/-1	NA
424.2/-1	NA
425.1/-1	NA
427.3/-1	906765.5
HAM005_641fE_14-11-07--Exp1.extracted	
402.3/-1	690820.7
403.2/-1	813749.8
405.1/-1	493671.4
409.2/-1	NA
410.1/-1	460283.8
413.3/-1	NA
423.2/-1	NA
424.2/-1	446598.9
425.1/-1	768109.1
427.3/-1	532297.9
HAM005_641fE_14-11-07--Exp2.extracted	
402.3/-1	826545.2
403.2/-1	390026.7
405.1/-1	383127.7
409.2/-1	NA
410.1/-1	340133.0
413.3/-1	NA
423.2/-1	NA
424.2/-1	429362.0
425.1/-1	648666.9
427.3/-1	NA
HAM005_641fE_14-11-07--Exp3.extracted	

402.3/-1	825108.3
403.2/-1	369377.0
405.1/-1	520611.4
409.2/-1	NA
410.1/-1	387501.5
413.3/-1	NA
423.2/-1	NA
424.2/-1	411374.7
425.1/-1	758610.4
427.3/-1	NA
HAM005_641fE_14-11-07--Exp4.extracted	
402.3/-1	639126.8
403.2/-1	437076.3
405.1/-1	NA
409.2/-1	NA
410.1/-1	423966.5
413.3/-1	NA
423.2/-1	NA
424.2/-1	458619.0
425.1/-1	492720.7
427.3/-1	NA
HAM005_641fE_14-11-07--Exp5.extracted	
402.3/-1	916502.6
403.2/-1	NA
405.1/-1	461028.9
409.2/-1	NA
410.1/-1	NA
413.3/-1	NA
423.2/-1	NA
424.2/-1	429517.4
425.1/-1	578125.7
427.3/-1	NA

The results are suited for instance for heatmaps, etc.