

The calibration pipeline based on iterative internal calibration.

Witold E. Wolski, mail : `witek96@users.sourceforge.net`

May 12, 2005

Introduction

This vignette describes how to calibrate spectra using a adaptively determined list of calibration masses and internal calibration. This procedure is iterated with increasing mass accuracy. The performance of this pipeline was studied *e.g.* by Wolski et al. [8] or Chamrad et al. [2].

The calibration sequence consists of:

- We first determine the abundant masses using two shifted overlapping histograms, with a bandwidth of $0.6Da$, and calibrate them by comparison with a list of theoretical tryptic autolysis masses.
- Next we align the dataset to the calibration list using internal calibration with an *MME* window of $450ppm$.
- We remove the sinusoidal components of the *MME* by external calibration[3].
- Again we determine the abundant masses (bandwith `accur = 0.3Da`) and use them as calibration list. The internal calibration is performed with an *MME* of $200ppm$.
- Prior to the database search we remove the ubiquitous masses.

Loading of packages and data

The package `mscalib` depends on the packages `fields`, `XML` and `msbase`. The library `gstat` can be loaded because of the `bpy.color` scheme. To read the Bruker Daltonics `peaklist.xml` file we use the function `readBruker`¹.

```
library(gstat)
```

```
> rm(list = ls())
```

```
> library(msbase)
```

```
Loading required package: MASS
```

```
Loading required package: XML
```

```
[1] "test1"
```

¹This function depends on the `XML` package

```
> library(mscalib)
```

Loading required package: fields
fields is loaded use help(fields) for an overview of this library
Loading required package: spatial

To perform database searches against the Mascot search server 1.8.1 (www.matrixscience.com)
the package `msmascot` is required which can be downloaded from: r4proteomics.sourceforge.net.

```
library(msmascot)
```

To read files in `peaklist.xml` (Bruker Daltonics) format.

```
NPPG <- readBruker( new("Massvectorlist") , "brukersampled" )  
object <- readBruker( new("Massvectorlist") , "brukerppgdir" )  
print(length(object))
```

```
> data(samples)  
> data(NPPG)
```

We display the masses of the dataset using the function `plot` which is drawing a strip-chart.

```
> plot(samples)
```

First calibration using abundant masses

To obtain the abundant masses in the dataset we use the function `gamasses` and set the bandwidth `accur = 0.6`.

```
> abmasses <- gamasses(samples, accur = 0.6)  
> data(cal2, package = "mscalib")
```

The histogram shows the mass frequencies. The green vertical lines indicate the abundant masses.

```
> hist(samples, accur = 0.6)  
> lines(abmasses, type = "h", col = 3, lwd = 1)
```

We calibrate the abundant masses using the function `getintcalib` and `applycalib`. The function `getintcalib` compares the masses to theoretical masses of tryptic autolysis products (dataset `cal2`) to determine the *MME*-model. Function `applycalib` removes the *MME* to *compile* a new calibration list.

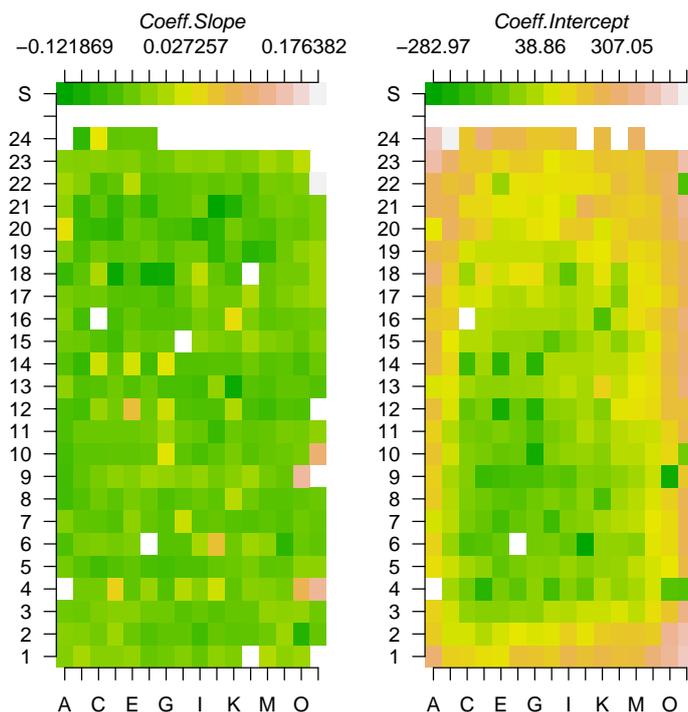
```
> calib <- getintcalib(abmasses, cal2, error = 300, ppm = TRUE)  
> abmasses <- applycalib(abmasses, calib)
```

We use the calibrated masses to determine the mass measurement error *MME* of all samples in the dataset (function `getintcalib`). We search for matching masses within a window of `error = 450ppm`.

```
> mmemod <- getintcalib(samples, abmasses, error = 450, ppm = T)
```

The image plot visualizes how the *MME*-model coefficients depend on the sample support position. While in case of the Slope coefficient we observe random variation, in case of the intercept coefficient we observe its systematic increase if getting closer to the sample support borders.

```
> par(mfrow = c(1, 2))
> image(mmemod, what = "Coeff.Slope", col = terrain.colors(100))
> image(mmemod, what = "Coeff.Intercept", col = terrain.colors(100))
```



We use the *MME*-models to calibrate the peak-lists (function `applycalib`).

```
> data.calib <- applycalib(samples, mmemod)
```

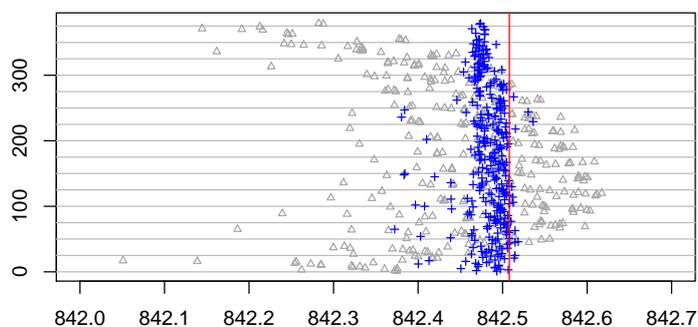
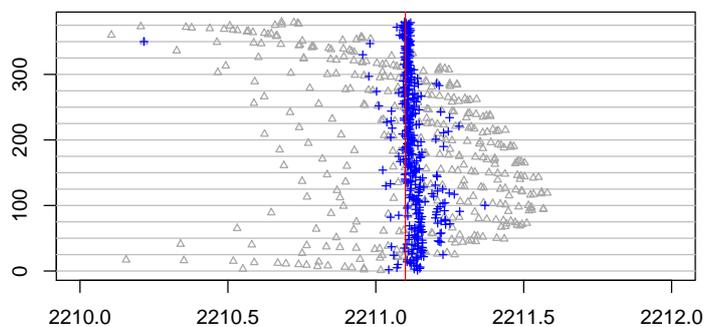
The strip-chart (function `plot`) allows to assess how well the calibration is. By the argument `xlim` we can select the mass range to be shown. In gray are the raw data and in blue are the calibrated masses. The red vertical lines are indicating the theoretical masses of tryptic autolysis products 842.508, 2211.100.

```
> par(mfrow = c(2, 1))
> mrang <- c(2210, 2212)
> par(mar = c(3, 3, 1, 1))
> plot(samples, xlim = mrang, main = "", pch = 2, col = "darkgray",
+       xlab = "", ylab = "", cex = 0.6)
> plot(data.calib, add = T, col = "blue", pch = 3, cex = 0.6)
> abline(v = cal2[, 1], col = 2)
> mrang <- c(842, 842.7)
> par(mar = c(3, 3, 1, 1))
> plot(samples, xlim = mrang, main = "", pch = 2, col = "darkgray",
```

```

+   xlab = "", ylab = "", cex = 0.6)
> plot(data.calib, add = T, col = "blue", pch = 3, cex = 0.6)
> abline(v = cal2[, 1], col = 2)

```



Validation of the MME-model

Chamrad et al.[2] suggest to check if the slope and intercept coefficient of the obtained *MME* is within the expected range. To do it the function `subset` can be used. We have observed that discarding the *MME*-model due to such criterions does not increase the identification rate. The code here is only to illustrate how the filtering can be performed using `mscalib`.

```

> mmemod2 <- subset(mmemod, Coeff.Intercept < 280 & Coeff.Intercept >
+   -280)
> mmemod2 <- subset(mmemod2, Coeff.Slope < 0.2 & Coeff.Slope >
+   -0.2)
> length(mmemod2)

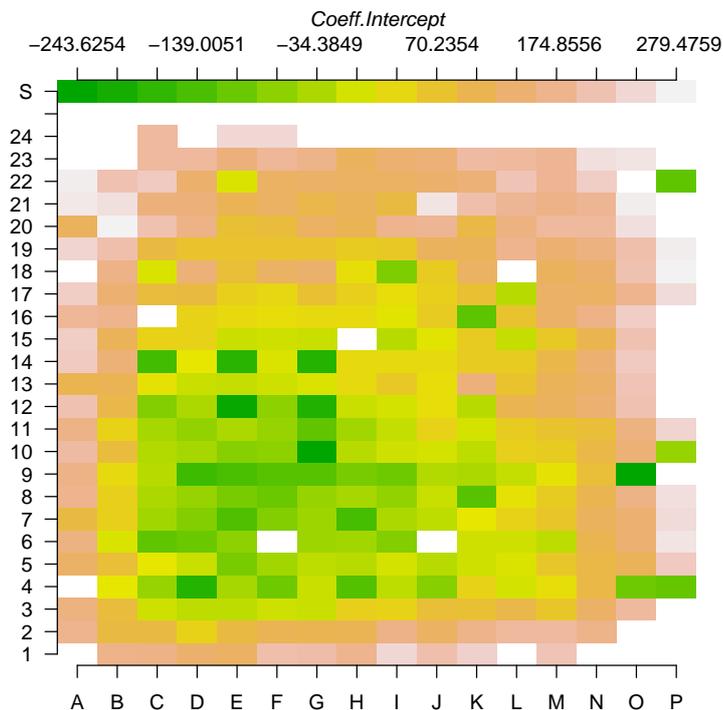
```

```
[1] 344
```

```

> image(mmemod2, what = "Coeff.Intercept", col = terrain.colors(100))

```



```
> rm(mmemod2)
```

Samuelsson et al. [6] suggest checking the obtained MME model against the peptide mass (PM) role. This can be easily performed on the calibrated peak-lists using the dissimilarity measure implemented in function `distance`.

```
> data.tmp <- applycalib(samples, mmemod)
> pr <- function(x) {
+   return(mean(distance(x[, 1, drop = T], 0)))
+ }
> hist(sapply(data.tmp, pr))
> mmemod2 <- mmemod[-which(sapply(data.calib, pr) > 0.22)]
> rm(mmemod2, data.tmp)
```

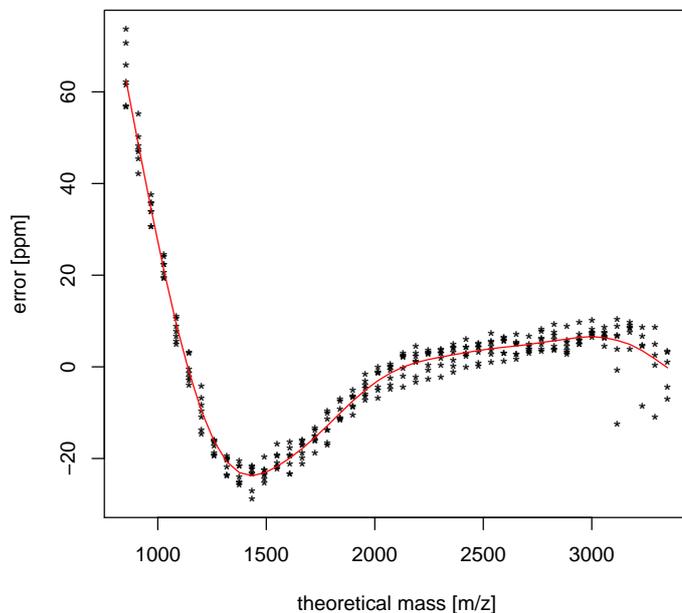
We do not going to use this filtering because we have not observed an increase of the identification rate due to it.

Removing higher order calibration errors

To remove higher order *MME*s we use external samples on which poly-(propylen glycol) (PPG) was measured [3]. Residuals of the experimental and matching theoretical ppg masses after removal of the intercept and slope *MME* are shown on the scatter plot below. We model this *MME* using internally the function `smooth.spline[1, 7]`.

```
> extcalib <- getextcalib(NPPG)
```

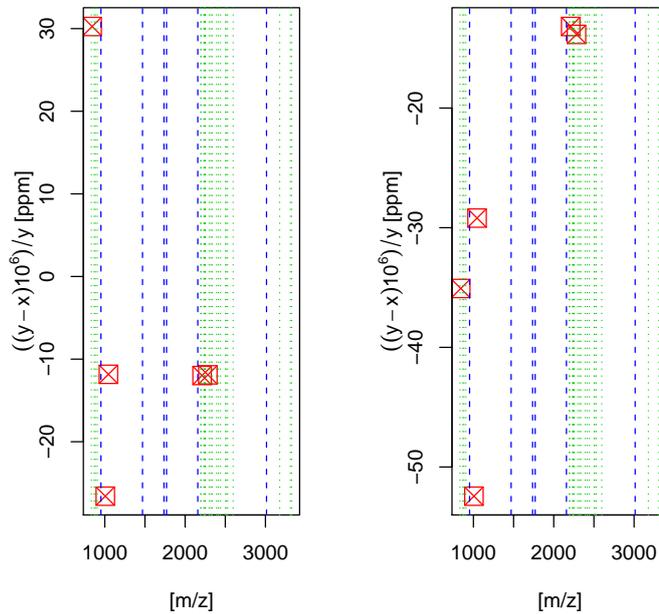
```
> plot(extcalib)
```



```
> abmasses <- gamasses(data.calib, accur = 0.4, abund = length(samples)/13)
> data.calib <- applycalib(data.calib, extcalib)
> abmasses2 <- gamasses(data.calib, accur = 0.4, abund = length(samples)/13)
```

To what extent this calibration increases the *MME* can be examined by determining the abundant masses and comparing them with theoretical masses of tryptic autolysis. The plot on the left is showing the residuals previous to external calibration, that one on the right the residuals after external calibration.

```
> par(mfrow = c(1, 2))
> plot(abmasses, cal2)
> plot(abmasses2, cal2)
```



Second calibration using abundant masses

We do a second calibration with a smaller mass range to search for matching internal calibration masses (200ppm).

```
> abmasses <- gamasses(data.calib, accur = 0.4, abund = length(samples)/13)
> calib <- getintcalib(abmasses, cal2, error = 200, ppm = TRUE)
> abmasses <- applycalib(abmasses, calib)
> mmemod <- getintcalib(data.calib, abmasses, error = 200, ppm = TRUE)
```

We apply the *MME*-models to correct for the *MME*.

```
> data.calib <- applycalib(data.calib, mmemod)
```

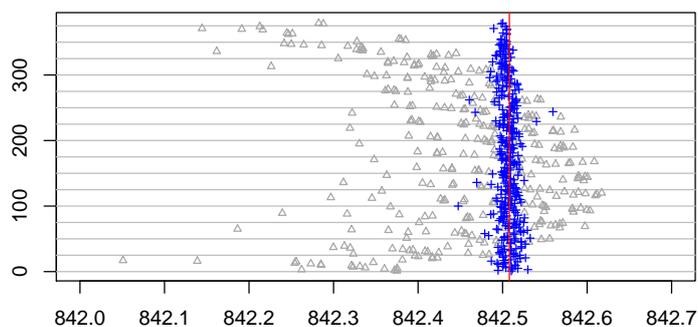
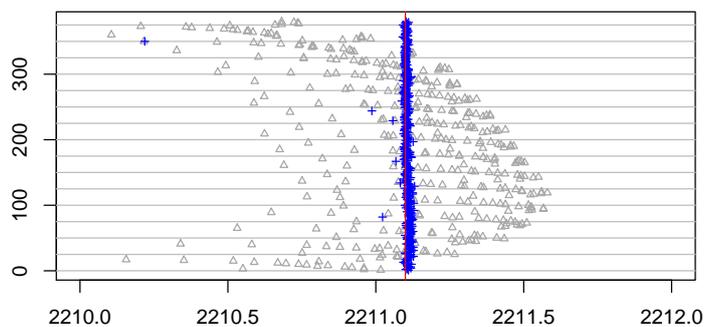
We use the strip-chart to visualize the *MME*. The gray triangles show the masses of the raw data, while the blue crosses show the masses after the second internal calibration.

```
> par(mfrow = c(2, 1))
> mrang <- c(2210, 2212)
> par(mar = c(3, 3, 1, 1))
> plot(samples, xlim = mrang, main = "", pch = 2, col = "darkgray",
+       xlab = "", ylab = "", cex = 0.6)
> plot(data.calib, add = T, col = "blue", pch = 3, cex = 0.6)
> abline(v = cal2[, 1], col = 2)
> mrang <- c(842, 842.7)
```

```

> par(mar = c(3, 3, 1, 1))
> plot(samples, xlim = mrange, main = "", pch = 2, col = "darkgray",
+       xlab = "", ylab = "", cex = 0.6)
> plot(data.calib, add = T, col = "blue", pch = 3, cex = 0.6)
> abline(v = cal2[, 1], col = 2)

```



Removal of ubiquitous masses

To increase the identification rate we remove masses that occur in more than 7.7% ($\text{abund} = \text{length}(\text{samples})/13$) of samples [2, 4]. To determine the abundant masses we use the function `gamasses`.

```

> abmasses <- gamasses(data.calib, accur = 0.25, abund = length(samples)/13)

```

We check which of them can be assigned with a significant Probability Based Mascot Score (*PBMS*) [5] to a, in the context of the experiment biologically significant, sequence database entry. We remove these masses from the filtering list. We configure the database search by choosing *e.g.* the database name and measurement accuracy.

(To execute the following code you need to have a running installation of the MASCOT search server 1.8.1 www.matrixscience.com and the package `ms-mascot` installed <http://r4proteomics.sourceforge.net>)

```

searchconfig <- getSearchconfig()
searchconfig$"DB" = "ARABI"

```

```

searchconfig$"TOL" = 0.2
searchconfig$"TOLU" <- "Da"
#search the abundant masses
res<-mascotSearch(abmasses,searchconfig,hits=5,research=TRUE)
summary(res)
tmp <- as.data.frame(res)
if(subset(tmp, research==1 & hitid==1)$score > 65)
{
  tmp <- getMatchedMass( res[[as.numeric(rownames(subset(tmp, research==1 & hitid==1)))]])
  plot(tmp)
  fmasses <- fsetdiff(abmasses,tmp)
}

```

Finally, we remove the ubiquitous masses from the dataset using the function `fsetdiff`.

```
data.cfilter <- fsetdiff(data.calib,abmasses,error=0.15,ppm=F)
```

The histogram shows in green the frequencies of masses in the filtered dataset. In black are the frequencies of the removed masses.

```
hist(data.calib,accur=0.3)
hist(data.cfilter,col=3,add=T,accur=0.3)
```

The sequence database search

The calibrated and filtered dataset is submitted for search.

```

searchconfig$"DB"="ARABI"
searchconfig$"TOL"=0.15
searchconfig$"TOLU"<-"Da"
searchres <- mascotSearch(data.cfilter,searchconfig,hits=5,minscore=55,research=TRUE,host=
save(searchres,file=paste(folderD,"searchres1cc.rda",sep=""))

```

The function `summary` creates a table. The table entries display the number of highest scoring DB hits (h1, h2, h3) with an significant *PBMS* (`sigscore=60`). The columns (s2, s3) show how many of the PL's resubmitted for search, after removal of the peaks matching a theoretical peptide mass in the first search s1, had a significant *PBMS*.

```

load(file=paste(folderD,"searchres1mst.rda",sep=""))
summary(searchres,sigscore=60)
tmp <- as.data.frame(searchres)
tmp <- subset(tmp, research==1 & hitid==1)$score
\begin{verbatim}

\begin{verbatim}
hist(tmp,breaks=seq(0,max(tmp)+1,by=1))
abline(v=60,col=2)

```

The histogram shows the distribution of the *PBMS* determined for the protein sequences ranked highest in the first search. They were retrieved from the `MascotResultList` object by the function `subset`. The red line is indicating the significance threshold of 60.

References

- [1] John M. Chambers and Trevor J. Hastie. *Statistical Models in S*. Chapman & Hall, London, 1992.
- [2] Daniel C. Chamrad, Gerhard Koerting, Johan Gobom, Herbert Thiele, Joachim Klose, Helmut E. Meyer, and Martin Blueggel. Interpretation of mass spectrometry data for high-throughput proteomics. *Anal Bioanal Chem*, 376(7):1014–22, Aug 2003.
- [3] J. Gobom, M. Mueller, V. Egelhofer, D. Theiss, H. Lehrach, and E. Nordhoff. A calibration method that simplifies and improves accurate determination of peptide molecular masses by maldi-tof-ms. *Analytical Chemistry*, 74(8):3915–3923, 2002.
- [4] Fredrik Levander, Thorsteinn Rognvaldsson, Jim Samuelsson, and Peter James. Automated methods for improved protein identification by peptide mass fingerprinting. *Proteomics*, 4(9):2594–601, Sep 2004.
- [5] D. N. Perkins, D. J. Pappin, D. M. Creasy, and J. S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, Dec 1999.
- [6] J. Samuelsson, D. Dalevi, F. Levander, and T. Rognvaldsson. Modular, scriptable, and automated analysis tools for High-Throughput peptide mass fingerprinting. *Bioinformatics*, Aug 5 2004.
- [7] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S. Fourth Edition*. Springer, 2002. ISBN 0-387-95457-0.
- [8] E. W. Wolski, T. Kreitler, J. Gobom, H. Lehrach, and K. Reinert. Ms calib. Poster at RECOMB 2003, 2003.