Mass Spectrometry Imaging using `MALDIquant`

Sebastian Gibb*

June 7, 2015

Abstract

This vignette describes the analysis of Mass Spectrometry Imaging data using `MALDIquant`

Contents

*mail@sebastiangibb.de

1 Foreword

MALDIquant is free and open source software for the R (R Core Team, 2014) environment and under active development. If you use it, please support the project by citing it in publications:

> Gibb, S. and Strimmer, K. (2012). MALDIquant: a versatile R package for the analysis of mass spectrometry data. *Bioinformatics*, 28(17):2270–2271

If you have any questions, bugs, or suggestions do not hesitate to contact me (mail@sebastiangibb.de).
Please visit http://strimmerlab.org/software/maldiquant/.

2 Other vignettes

Please have a look at our other vignettes on https://github.com/sgibb/MALDIquantExamples:

- MALDIquant Introduction — a general introduction how to analyze mass spectrometry data using MALDIquant.
- MALDIquantForeign Introduction — a general introduction how to import/export data using MALDIquantForeign.
- Analysis of Fiedler et al. 2009 — a guidance to analyse the serum profile MALDI-TOF data described in Fiedler et al. (2009).
- Bacterial Species Determination — a guidance to determine different species based on their MALDI-TOF spectra.
- Mass Spectrometry Imaging — a guidance how to analyse mass spectrometry imaging data using MALDIquant.

3 Setup

Before any analysis we need to install the necessary packages (you can skip this part if you have already done this). You can install MALDIquant (Gibb

and Strimmer, 2012), MALDIquantForeign (Gibb, 2014) directly from CRAN. To install this data package from https://github.com/sgibb/MALDIquantExamples you need the drat (Eddelbuettel, 2015) package and add our repository to your installation sources. Afterwards you could install the latest version of MALDIquantExamples via install.packages/update.packages:

```
install.packages("drat")

## add this to your .Rprofile to make the change permanent
drat::addRepo("sgibb")

## install MALDIquantExamples package and all its dependencies
install.packages("MALDIquantExamples")

## to update to the latest version
## (if you have installed MALDIquantExamples before)
update.packages()
```

4 Dataset

The dataset we use in this vignette was kindly provided by Dr. Adrien Nyakas (adrien.nyakas@dcb.unibe.ch; http://dx.doi.org/10.6084/m9.figshare.735961). It contains 2222 MALDI-TOF spectra (coordinates: (29, 61) to (101, 98)) of a mouse kidney.

5 Analysis

First we have to load the packages.

```
## the main MALDIquant package
library("MALDIquant")
## the import/export routines for MALDIquant
library("MALDIquantForeign")

## example data
library("MALDIquantExamples")
```

5.1 Import Raw Data

Next we use the `getPathNyakas2013` function to get the correct file path of our example data and import them into `R`.

```
## import the spectra
spectra <- import(getPathNyakas2013(), verbose=FALSE)
```

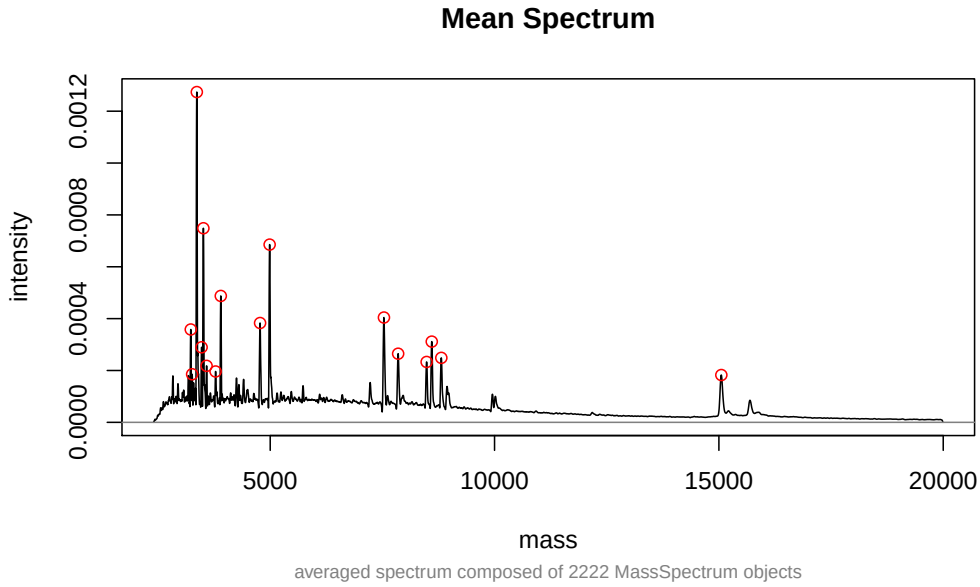
We do a basic quality control and test whether all spectra contain the same number of data points and are not empty.

5.2 Preprocessing

The complete preprocessing is very similar to the default worflow for mass spectrometry data. Please find a more detailed description in the vignette MALDIquant Introduction.

After a basic preprocessing of all spectra we produce a mean spectrum and run a peak detection on it to find regions of interest.

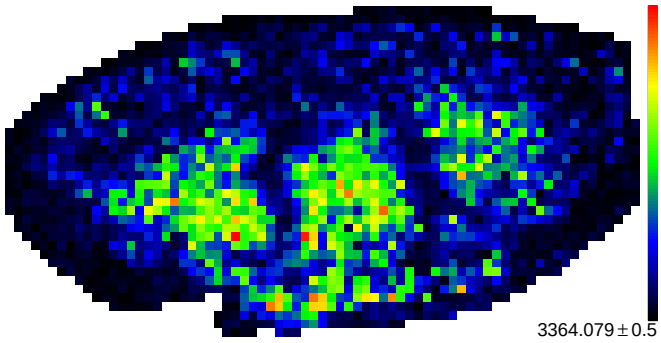
```
meanSpectrum <- averageMassSpectra(spectra)

roi <- detectPeaks(meanSpectrum, SNR=4,
                   halfWindowSize=10)

plot(meanSpectrum, main="Mean Spectrum")
points(roi, col="red")
```



5.4 Plotting Slices

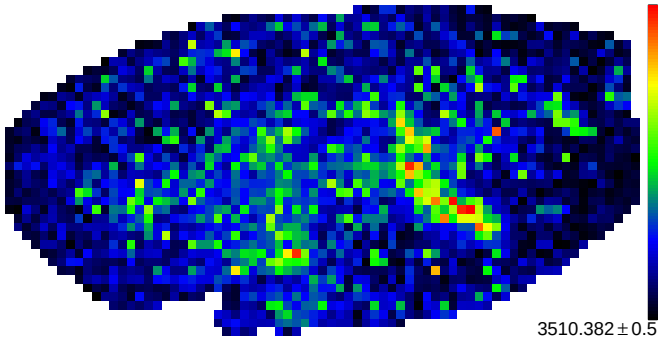
We want to plot a mass spectrometry image slice around the highest peak in our mean spectrum.

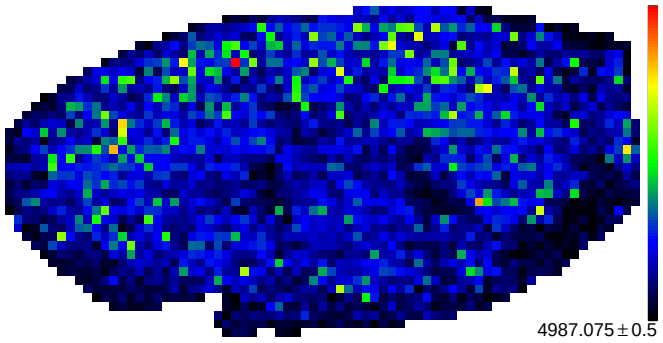
```
## find order of peak intensities
o <- order(intensity(roi), decreasing=TRUE)

## plot MSI slice for the highest one
plotMsiSlice(spectra, center=mass(roi)[o[1]], tolerance=0.5)
```



We could plot multiple slices as well.

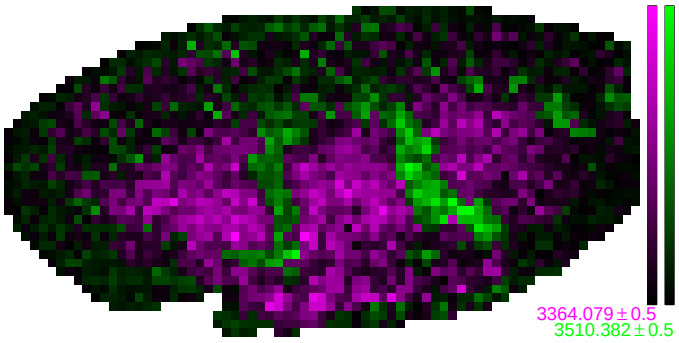
```
plotMsiSlice(spectra, center=mass(roi)[o[2:3]], tolerance=0.5)
```





Another possibility would be to combine these regions of interest in one plot.

```
plotMsiSlice(spectra, center=mass(roi)[o[1:2]], tolerance=0.5,
             combine=TRUE,
             colRamp=list(colorRamp(c("#000000", "#FF00FF")),
                          colorRamp(c("#000000", "#00FF00"))))
```



5.5 Working with slices/coordinates

Sometimes the slices should be processed further. For this purposes `msiSlices` generates an array with the dimensions x coordinates, y coordinates and center mass.

```
slices <- msiSlices(spectra, center=mass(roi), tolerance=0.5)
attributes(slices)

$dim
 x  y  z
73 38 16

$center
 [1]  3229.643  3259.476  3364.079  3470.337  3510.382  3581.936  3781.966
 [8]  3898.438  4773.936  4987.075  7534.798  7851.204  8486.532  8600.549
[15]  8813.489 15054.208

$tolerance
[1] 0.5
```

```
$method
[1] "sum"
```

Via the `coordinates` method we get the pixel coordinates of our spectra. Use the argument `adjust` to set the minimal values to 1.

```
head(coordinates(spectra))
```

```
      x  y
[1,] 29 75
[2,] 29 76
[3,] 29 77
[4,] 29 78
[5,] 29 79
[6,] 29 80
```

```
head(coordinates(spectra, adjust=TRUE))
```

```
     x  y
[1,] 1 15
[2,] 1 16
[3,] 1 17
[4,] 1 18
[5,] 1 19
[6,] 1 20
```

5.6 Clustering

While we could highlight some mass values in our slices we sometimes want to do some clustering to get information about the spatial order.

Therefore we build a peak intensity matrix first.

```
peaks <- detectPeaks(spectra, SNR=3,
                     halfWindowSize=10)
peaks <- binPeaks(peaks)
intMatrix <- intensityMatrix(peaks, spectra)
```

Subsequently we run a `kmeans` clustering with 3 centers. We choose 3 centers because the kidney is divided in 2 main anatomical parts, the renal cortex (the outer part) and the renal medulla (the inner part, containing the renal pyramids).

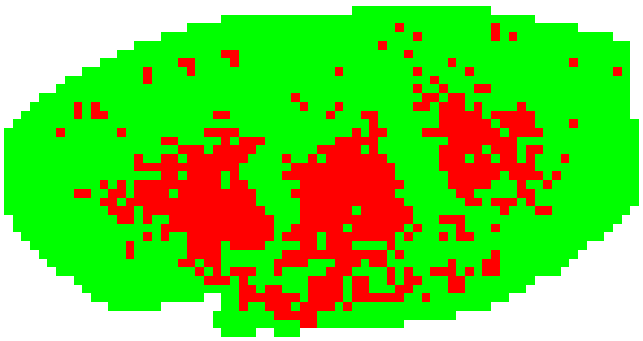
```
km <- kmeans(intMatrix, centers=2)
```

For visualisation we create a new matrix and replace each coordinate by its cluster number.

```
coord <- coordinates(spectra, adjust=TRUE)
maxPixels <- apply(coord, MARGIN=2, FUN=max)
m <- matrix(NA, nrow=maxPixels["x"], ncol=maxPixels["y"])
m[coord] <- km$cluster
```

In the following step we use the `plotMsiSlice` function again to plot our cluster matrix. Now we use the argument `scale=FALSE` to avoid the scaling to values between 0 and 1. Also we provide an own `colRamp` function that returns red or green for the clusters 1, and 2 respectively (must generated the same matrix output as `graphics::colorRamp`).

```
rgbCluster <- function(x) {
  col <- matrix(c(255, 0, 0,
                  0, 255, 0), nrow=2, byrow=TRUE)
  col[x, ]
}
plotMsiSlice(m, colRamp=rgbCluster, scale=FALSE)
```



Please note that the base `kmeans` doesn't respect any spatial information.

5.7 Summary

While the default Mass Spectrometry Imaging workflow is very similar to the default profile spectra workflow (and could be found in a detailed discussion in the other `MALDIquant` vignettes) we demonstrate typical Mass Spectrometry Imaging functions like plotting slices and clustering data.

6 Session Information

- R version 3.2.0 (2015-04-16), `x86_64-pc-linux-gnu`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: MALDIquant 1.11.14, MALDIquantExamples 0.4, MALDIquantForeign 0.9.11, corpcor 1.6.7, crossval 1.0.2, entropy 1.2.1, fdrtool 1.2.14, knitr 1.10.5, pvclust 1.3-2, sda 1.3.6, xtable 1.7-4

- Loaded via a namespace (and not attached): XML 3.98-1.2, base64enc 0.1-2, digest 0.6.8, downloader 0.3, evaluate 0.7, formatR 1.2, highr 0.5, magrittr 1.5, parallel 3.2.0, readBrukerFlexData 1.8.2, readMzXmlData 2.8, stringi 0.4-1, stringr 1.0.0, tools 3.2.0

References

Eddelbuettel, D. (2015). *drat: Drat R Archive Template*. R package version 0.0.4.

Fiedler, G. M., Leichtle, A. B., Kase, J., Baumann, S., Ceglarek, U., Felix, K., Conrad, T., Witzigmann, H., Weimann, A., Schtte, C., Hauss, J., Büchler, M., and Thiery, J. (2009). Serum peptidome profiling revealed platelet factor 4 as a potential discriminating peptide associated with pancreatic cancer. *Clinical Cancer Research*, 15:3812–3819.

Gibb, S. (2014). *MALDIquantForeign: Import/Export routines for MALDIquant*. R package version 0.7.

Gibb, S. and Strimmer, K. (2012). MALDIquant: a versatile R package for the analysis of mass spectrometry data. *Bioinformatics*, 28(17):2270–2271.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.