Analysis of Fiedler et al. 2009 using `MALDIquant`

Sebastian Gibb*

May 20, 2014

Abstract

This vignette describes the analysis of the MALDI-TOF spectra described in Fiedler et˜al. (2009) 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 Dataset

In this vignette we use the dataset described in Fiedler et˜al. (2009). Please contact the authors directly if you want to use the dataset in your own analysis.

This dataset contains 480 MALDI-TOF mass spectra from blood sera of 60 patients and 60 healthy controls (each sample has four technical replicates).

It is divided in three set:

1. *Discovery Set A*: 20 patients with pancreatic cancer and 20 healthy patients from the University Hospital Leipzig.

2. *Discovery Set B*: 20 patients with pancreatic cancer and 20 healthy patients from the University Hospital Heidelberg.

3. *Discovery Set C*: 20 patients with pancreatic cancer and 20 healthy patients from the University Hospital Leipzig (half resolution).

Both discovery sets A and B were measured on the same target (batch). The validation set C was measured a few months later.

Please see Fiedler et˜al. (2009) for details.

3 Analysis

3.1 Setup

First 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), `sda` (Ahdesmaki et˜al., 2014) and `crossval` (Strimmer., 2014) directly from `CRAN`. To install this data package from `http://github.com/sgibb/MALDIquantExamples` you need the `devtools` (Wickham and Chang, 2014) package.

```
install.packages(c("MALDIquant", "MALDIquantForeign",
                   "sda", "crossval", "devtools"))
library("devtools")
install_github("sgibb/MALDIquantExamples")
```

Next we load the packages.

```
library("MALDIquant")
library("MALDIquantForeign")
library("sda")
library("crossval")

library("MALDIquantExamples")
```

3.2 Import Raw Data

We use the `getPathFiedler2009` function to get the correct file path to the spectra and the metadata file respectively.

```
## import the spectra
spectra <- import(getPathFiedler2009()["spectra"],
                  verbose=FALSE)

## import metadata
spectra.info <- read.table(getPathFiedler2009()["info"],
                           sep=",", header=TRUE)
```

Because of heavy batch effects between the two hospitals we consider only the data collected in the University Hospital Heidelberg.

```
isHeidelberg <- spectra.info$location == "heidelberg"

spectra <- spectra[isHeidelberg]
spectra.info <- spectra.info[isHeidelberg,]
```

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

3.3 Quality Control

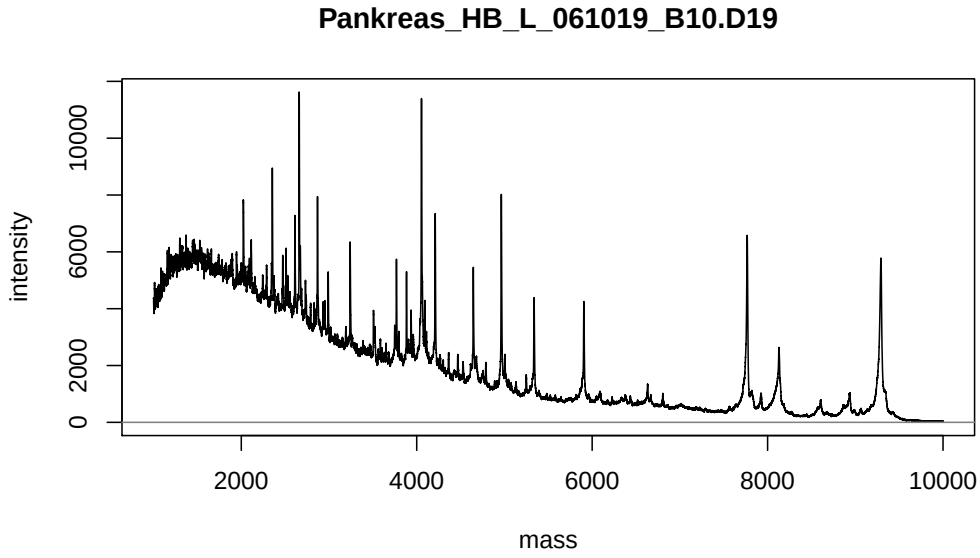
```
table(sapply(spectra, length))


42388
  160

any(sapply(spectra, isEmpty))

[1] FALSE

all(sapply(spectra, isRegular))

[1] TRUE
```

Subsequently we ensure that all spectra have the same mass range.

```
spectra <- trim(spectra)
```

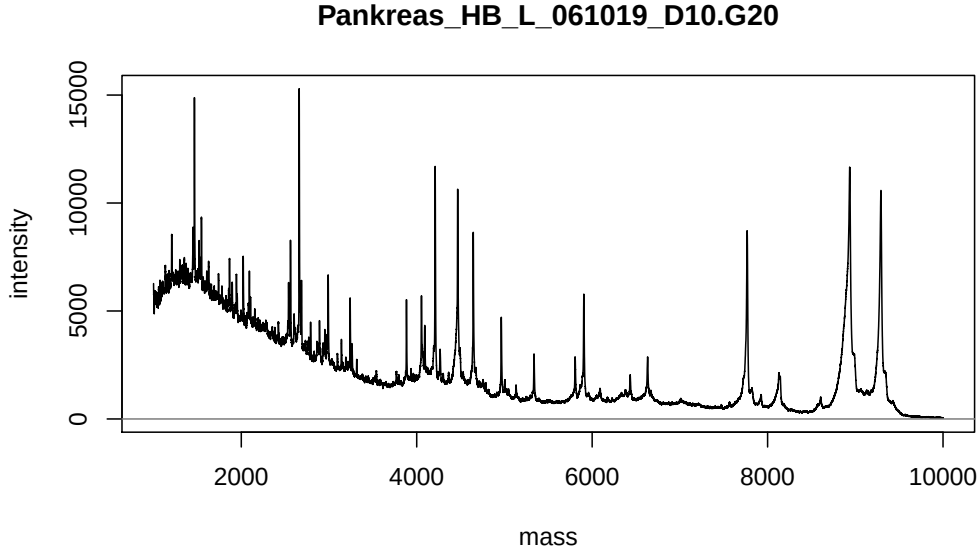
Finally we draw some plots and inspect the spectra visually.

```
idx <- sample(length(spectra), size=2)
plot(spectra[[idx[1]]])
```



DIquantForeign_uncompress/spectra_68085edb76f4/fiedler_et_al_2009/set B − discovery heidelberg/control/Pankreas_HB_L_

```
plot(spectra[[idx[2]]])
```



.DIquantForeign_uncompress/spectra_68085edb76f4/fiedler_et_al_2009/set B − discovery heidelberg/tumor/Pankreas_HB_L_(

3.4 Transformation and Smoothing

We apply the square root transformation to simplify graphical visualization and to overcome the potential dependency of the variance from the mean.

```
spectra <- transformIntensity(spectra, method="sqrt")
```

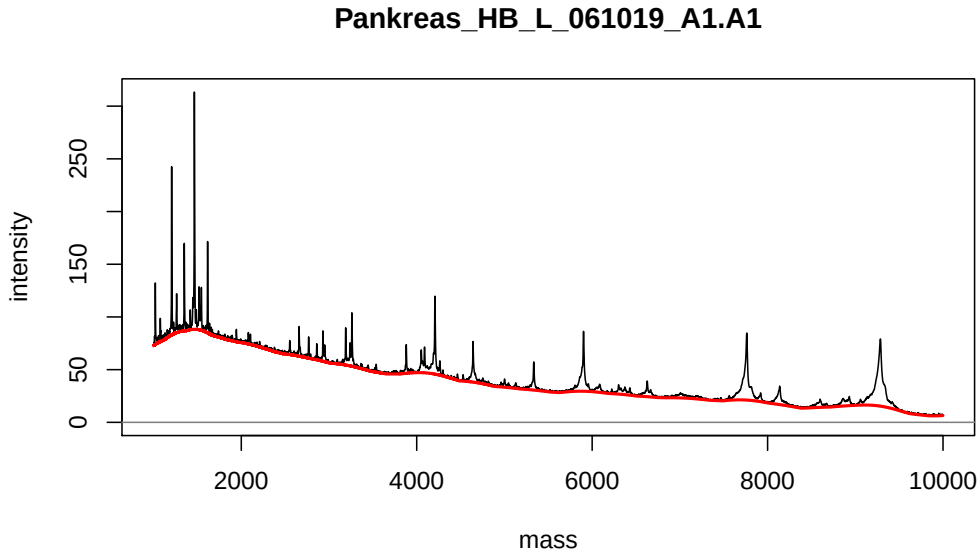
In the next step we use a 21 point *Savitzky-Golay*-Filter (Savitzky and Golay, 1964) to smooth the spectra.

```
spectra <- smoothIntensity(spectra, method="SavitzkyGolay",
                           halfWindowSize=10)
```

3.5 Baseline Correction

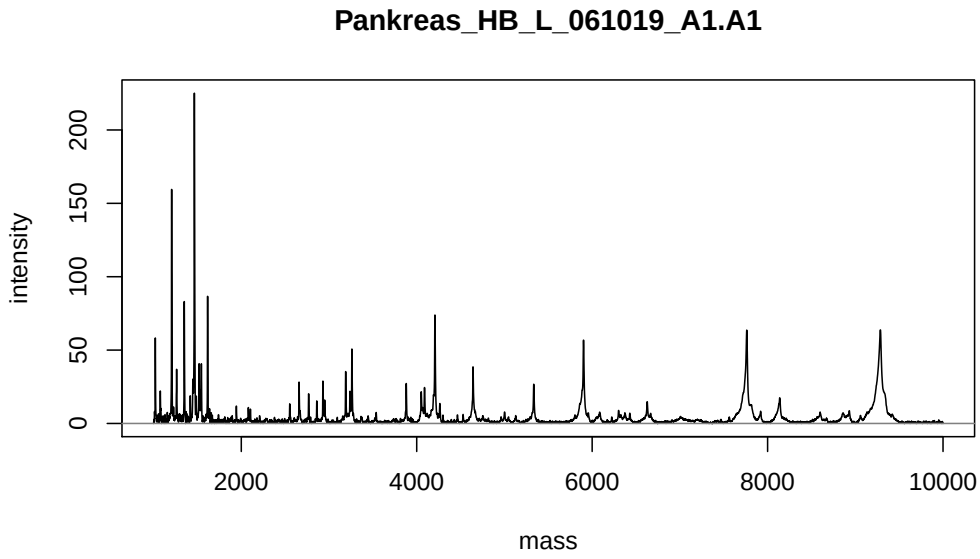
Matrix effects and chemical noise results in some background noise. That's why we have to apply a baseline correction. In this example we use the *SNIP* algorithm (Ryan et˜al., 1988) to correct the baseline.

```
baseline <- estimateBaseline(spectra[[1]], method="SNIP",
                             iterations=150)
plot(spectra[[1]])
lines(baseline, col="red", lwd=2)
```



LDIquantForeign_uncompress/spectra_68085edb76f4/fiedler_et_al_2009/set B − discovery heidelberg/control/Pankreas_HB_L

```
spectra <- removeBaseline(spectra, method="SNIP",
                          iterations=150)
plot(spectra[[1]])
```



LDIquantForeign_uncompress/spectra_68085edb76f4/fiedler_et_al_2009/set B − discovery heidelberg/control/Pankreas_HB_L

3.6 Intensity Calibration

We perform the *Total-Ion-Current*-calibration (TIC; often called normalization) to equalize the intensities across spectra.

```
spectra <- calibrateIntensity(spectra, method="TIC")
```

3.7 Alignment

Next we need to (re)calibrate the mass values. Our alignment procedure is a peak based warping algorithm. `MALDIquant` offers `alignSpectra` as a wrapper around more complicated functions. If you need a finer control or want to investigate the impact of different parameters please use `determineWarpingFunctions` instead (see `?determineWarpingFunctions` for details).

```
spectra <- alignSpectra(spectra)
```

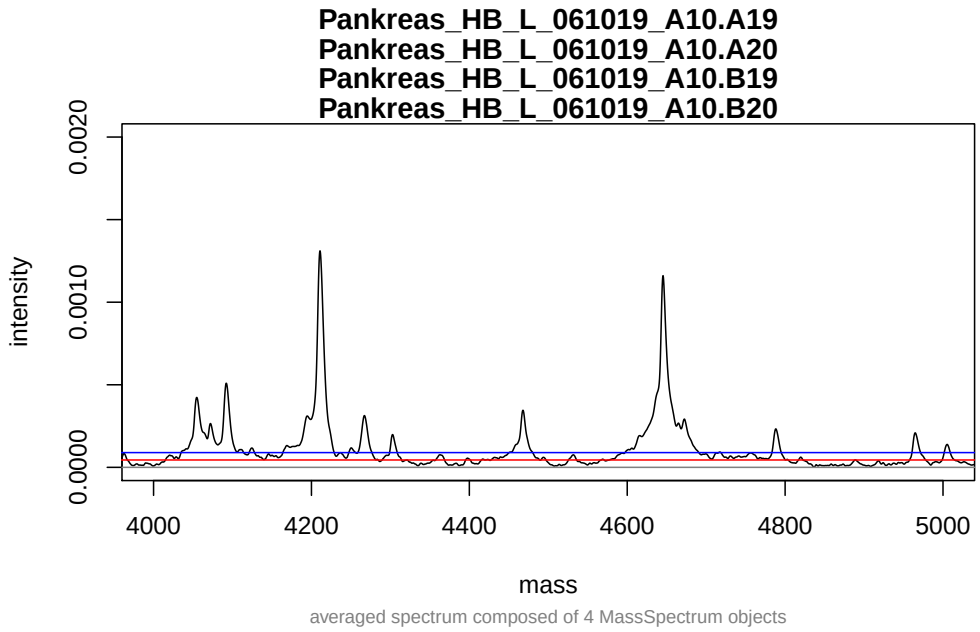
We average the technical replicates before we look for peaks and adjust our metadata table accordingly.

```
avgSpectra <-
  averageMassSpectra(spectra, labels=spectra.info$patientID)
avgSpectra.info <-
  spectra.info[!duplicated(spectra.info$patientID), ]
```

3.8 Peak Detection

The peak detection is the crucial feature reduction step. Before performing the peak detection we estimate the noise of some spectra to get a feeling for the *signal-to-noise ratio* (SNR).

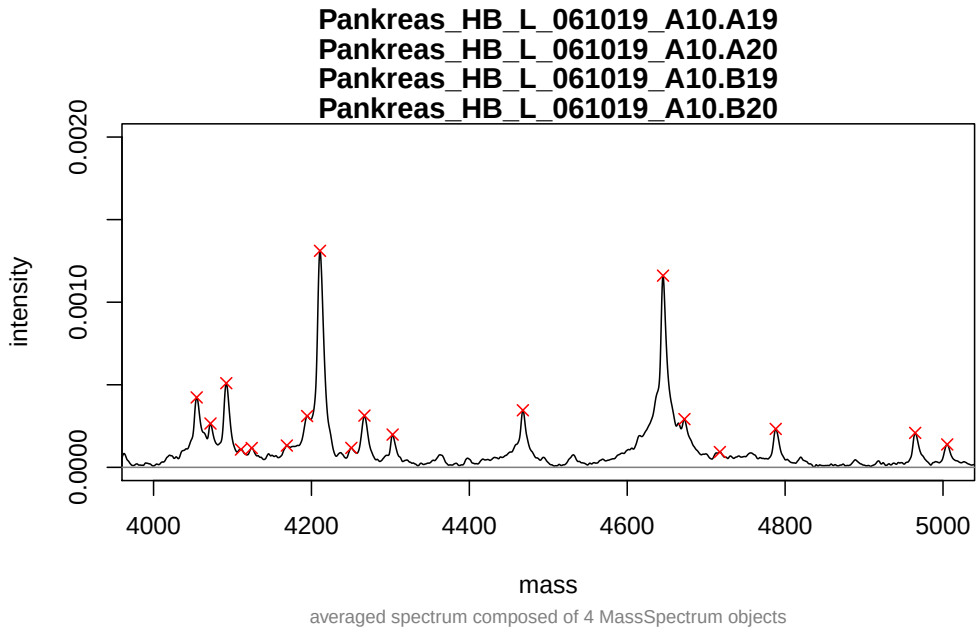
```
noise <- estimateNoise(avgSpectra[[1]])
plot(avgSpectra[[1]], xlim=c(4000, 5000), ylim=c(0, 0.002))
lines(noise, col="red")                    # SNR == 1
lines(noise[, 1], 2*noise[, 2], col="blue") # SNR == 2
```



In this case we decide to set a *SNR* of 2 (blue line).

```
peaks <- detectPeaks(avgSpectra, SNR=2, halfWindowSize=20)
```

```
plot(avgSpectra[[1]], xlim=c(4000, 5000), ylim=c(0, 0.002))
points(peaks[[1]], col="red", pch=4)
```



3.9 Post Processing

After the alignment the peak positions (mass) are very similar but not identical. The binning is needed to make similar peak mass values identical.

```
peaks <- binPeaks(peaks)
```

We choose a very low signal-to-noise ratio to keep as much features as possible. To remove some false positive peaks we remove peaks that appear in less than 50 % of all spectra in each group.

```
peaks <- filterPeaks(peaks, minFrequency=c(0.5, 0.5),
                     labels=avgSpectra.info$health,
                     mergeWhitelists=TRUE)
```

Finally we create the feature matrix and label the rows with the corresponding patient ID.

```
featureMatrix <- intensityMatrix(peaks, avgSpectra)
rownames(featureMatrix) <- avgSpectra.info$patientID
```

3.10 Diagonal Discriminant Analysis

We finish the `MALDIquant` preprocessing and use the *diagonal discriminant analysis* (DDA) function of `sda` (Ahdesmaki et˜al., 2014) to find the most important peaks.

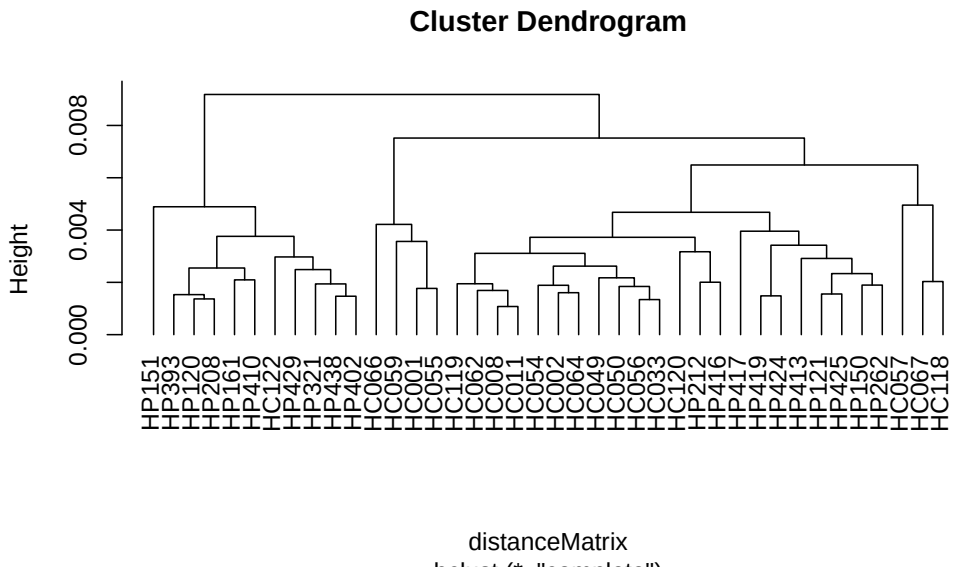
```
Xtrain <- featureMatrix
Ytrain <- avgSpectra.info$health
ddar <- sda.ranking(Xtrain=featureMatrix, L=Ytrain, fdr=FALSE,
                    diagonal=TRUE)

Computing t-scores (centroid vs. pooled mean) for feature ranking

Number of variables: 177
Number of observations: 40
Number of classes: 2

Estimating optimal shrinkage intensity lambda.freq (frequencies): 1
Estimating variances (pooled across classes)
Estimating optimal shrinkage intensity lambda.var (variance vector): 0.1072
```

	idx	score	t.cancer	t.control
8937.02377967774	169.00	89.24	9.45	-9.45
4467.82760195335	123.00	80.21	8.96	-8.96
8868.21035969657	168.00	79.76	8.93	-8.93
4494.72923193083	124.00	70.05	8.37	-8.37
8989.39357377299	170.00	65.53	8.09	-8.09
5864.40916105019	144.00	37.51	-6.12	6.12
5906.05239413598	145.00	34.49	-5.87	5.87
2022.76307818314	52.00	33.90	5.82	-5.82
5945.59928865137	146.00	33.34	-5.77	5.77
1866.06934418929	46.00	32.42	5.69	-5.69

3.11 Hierarchical Clustering

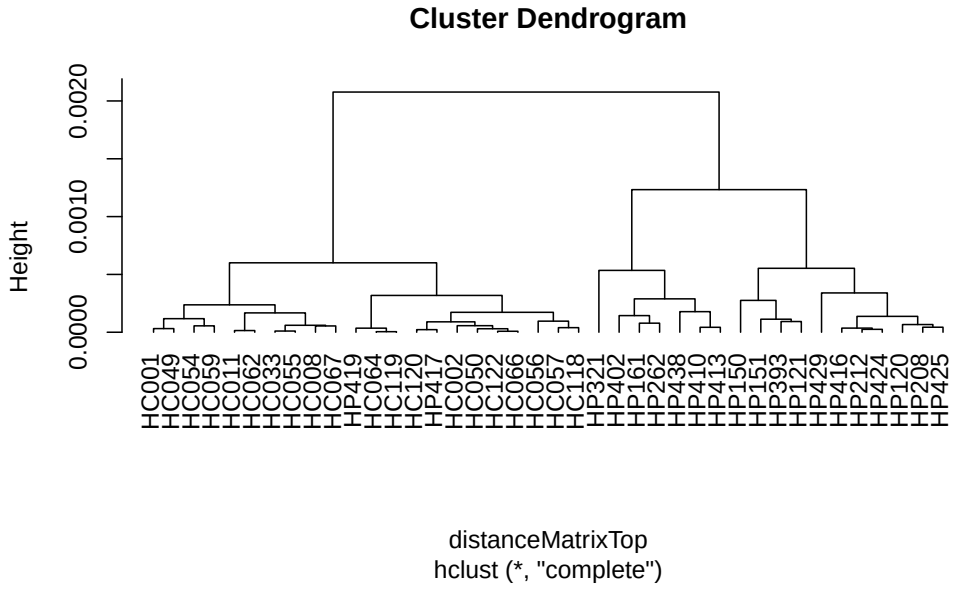
To visualize the results without any feature selection by *DDA* we apply a hierarchical cluster analysis based on the euclidean distance.

```
distanceMatrix <- dist(featureMatrix, method="euclidean")

hClust <- hclust(distanceMatrix, method="complete")

plot(hClust, hang=-1)
```



Next we use only the 2 top peaks selected in the *DDA* and we get a nearly perfect split between the cancer and control group.

```
top <- ddar[1:2, "idx"]

distanceMatrixTop <- dist(featureMatrix[, top],
                          method="euclidean")

hClustTop <- hclust(distanceMatrixTop, method="complete")

plot(hClustTop, hang=-1)
```



3.12 Cross Validation

Subsequently we use the `crossval` (Strimmer., 2014) package to perform a 10-fold cross validation of these two selected peaks.

```
# create a prediction function for the cross validation
predfun.dda <- function(Xtrain, Ytrain, Xtest, Ytest,
                        negative) {
  dda.fit <- sda(Xtrain, Ytrain, diagonal=TRUE, verbose=FALSE)
  ynew <- predict(dda.fit, Xtest, verbose=FALSE)$class
  return(confusionMatrix(Ytest, ynew, negative=negative))
}

# set seed to get reproducible results
set.seed(1234)

cv.out <- crossval(predfun.dda,
                   X=featureMatrix[, top],
                   Y=avgSpectra.info$health,
                   K=10, B=20,
```

```
                    negative="control",
                    verbose=FALSE)
diagnosticErrors(cv.out$stat)

   acc   sens   spec    ppv    npv    lor
0.9500 0.9000 1.0000 1.0000 0.9091    Inf
```

3.13 Summary

We found the peaks m/z 8937 and 4467 as important features for the discrimination between the cancer and control group.

4 Session Information

- R version 3.1.0 (2014-04-10), `x86_64-pc-linux-gnu`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: MALDIquant˜1.10.1, MALDIquantExamples˜0.1, MALDIquantForeign˜0.7, corpcor˜1.6.6, crossval˜1.0.1, entropy˜1.2.0, fdrtool˜1.2.12, knitr˜1.5, sda˜1.3.3, xtable˜1.7-3
- Loaded via a namespace (and not attached): XML˜3.98-1.1, base64enc˜0.1-1, digest˜0.6.4, downloader˜0.3, evaluate˜0.5.5, formatR˜0.10, highr˜0.3, readBrukerFlexData˜1.7, readMzXmlData˜2.7, stringr˜0.6.2, tools˜3.1.0

References

Ahdesmaki, M., Zuber, V., Gibb, S., and Strimmer, K. (2014). *sda: Shrinkage Discriminant Analysis and CAT Score Variable Selection*. R package version 1.3.2.

Fiedler, G.˜M., Leichtle, A.˜B., Kase, J., Baumann, S., Ceglarek, U., Felix, K., Conrad, T., Witzigmann, H., Weimann, A., Schütte, 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. *Clin Cancer Res*, 15(11):3812–3819.

Gibb, S. (2014). *MALDIquantForeign: Import/Export routines for MALDIquant*. R package version 0.6.2.

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.

Ryan, C., Clayton, E., Griffin, W., Sie, S., and Cousens, D. (1988). Snip, a statistics-sensitive background treatment for the quantitative analysis of pixe spectra in geoscience applications. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 34(3):396 – 402.

Savitzky, A. and Golay, M. J.˜E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639.

Strimmer., K. (2014). *crossval: Generic Functions for Cross Validation*. R package version 1.0.0.

Wickham, H. and Chang, W. (2014). *devtools: Tools to make developing R code easier*. R package version 1.5.