Analysis of Fiedler et al. 2009 using `MALDIquant`

Sebastian Gibb*

June 8, 2015

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 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), `sda` (Ahdesmäki and Strimmer, 2010) 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")
```

4 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.

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

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.

5.2 Quality Control

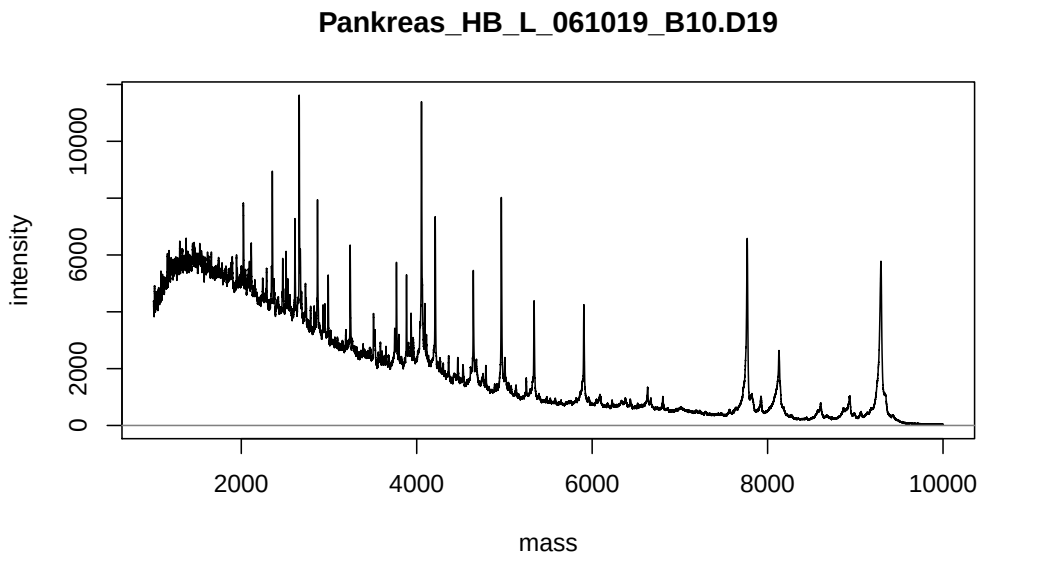
```
table(sapply(spectra, length))


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)
```

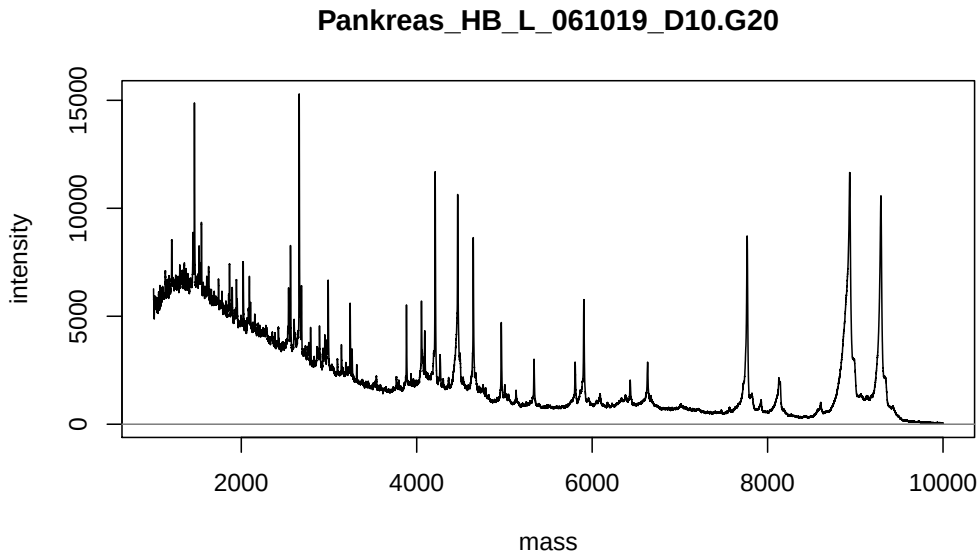
Finally we draw some plots and inspect the spectra visually.

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



DIquantForeign_uncompress/spectra_1b8b3fe51c8e/fiedler_et_al_2009/set B – discovery heidelberg/control/Pankreas_HB_L_

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



.DIquantForeign_uncompress/spectra_1b8b3fe51c8e/fiedler_et_al_2009/set B – discovery heidelberg/tumor/Pankreas_HB_L_(

5.3 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")
```

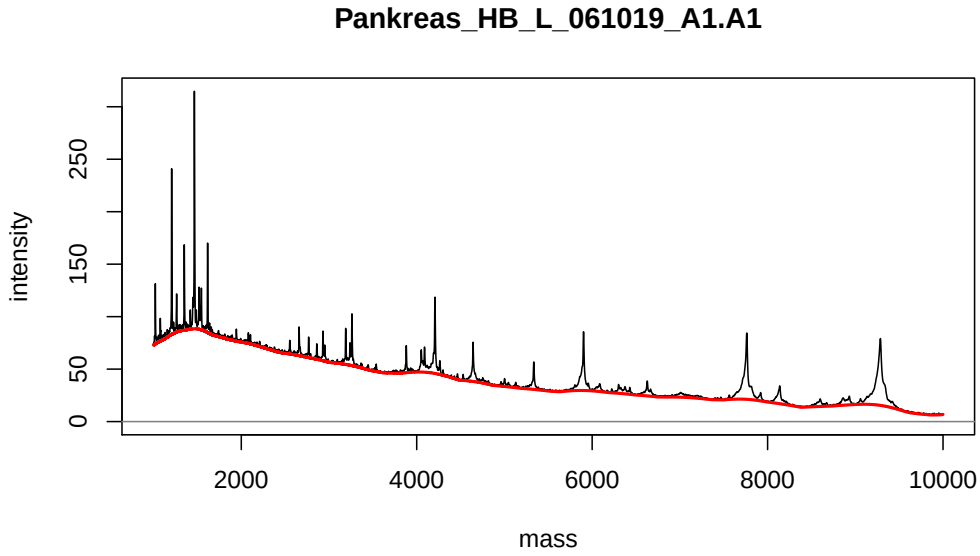
In the next step we use a 41 point *Savitzky-Golay*-Filter (Savitzky and Golay, 1964) to smooth the spectra.

```
spectra <- smoothIntensity(spectra, method="SavitzkyGolay",
                           halfWindowSize=20)
```

5.4 Baseline Correction

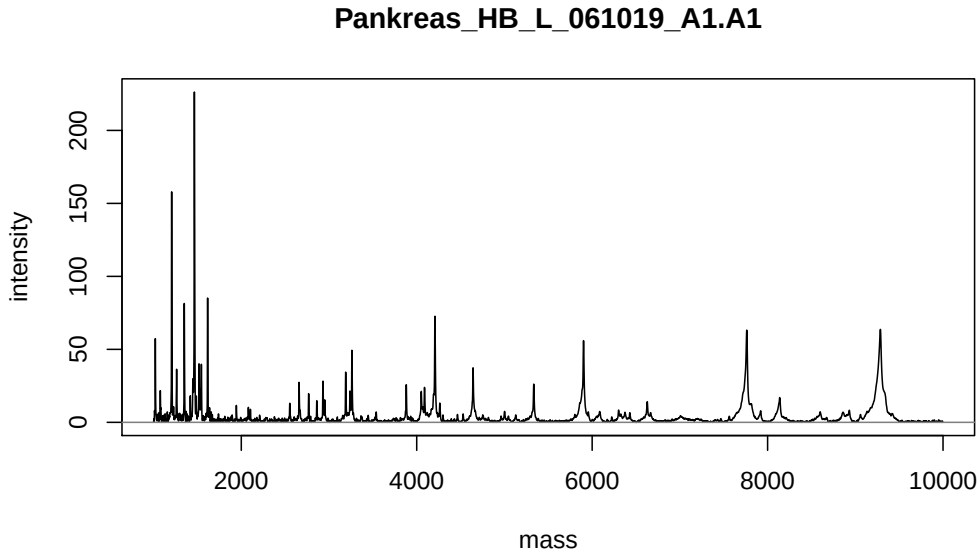
Matrix effects and chemical noise results in some background noise. That's why we have to apply a baseline correction. In this example we use the *SNIP* algorithm (Ryan et al., 1988) to correct the baseline.

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



LDIquantForeign_uncompress/spectra_1b8b3fe51c8e/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_1b8b3fe51c8e/fiedler_et_al_2009/set B − discovery heidelberg/control/Pankreas_HB_L

5.5 Intensity Calibration

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

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

5.6 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)
```

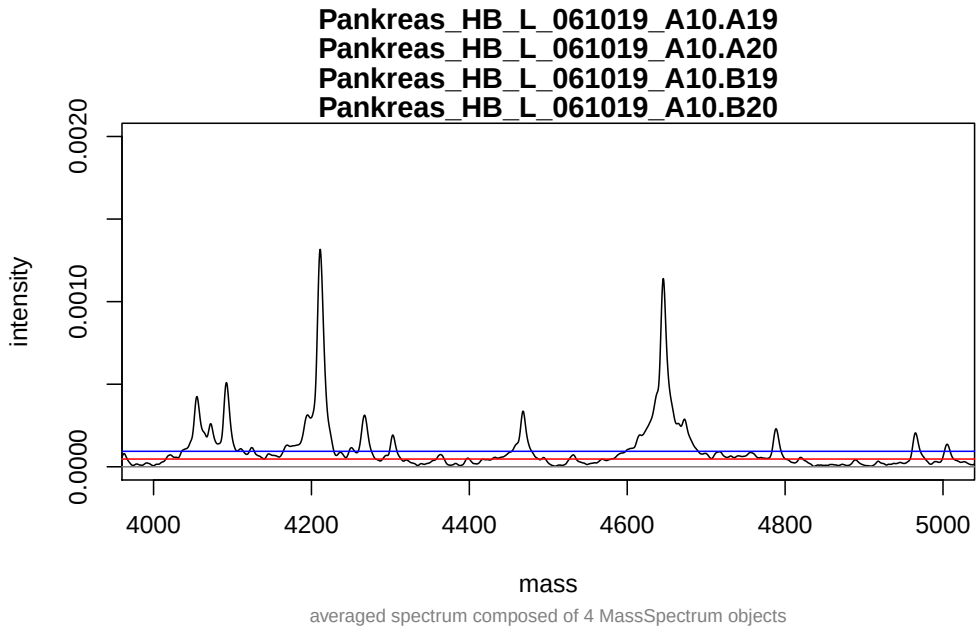
We average the technical replicates before we look for peaks and adjust our metadata table accordingly.

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

5.7 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).

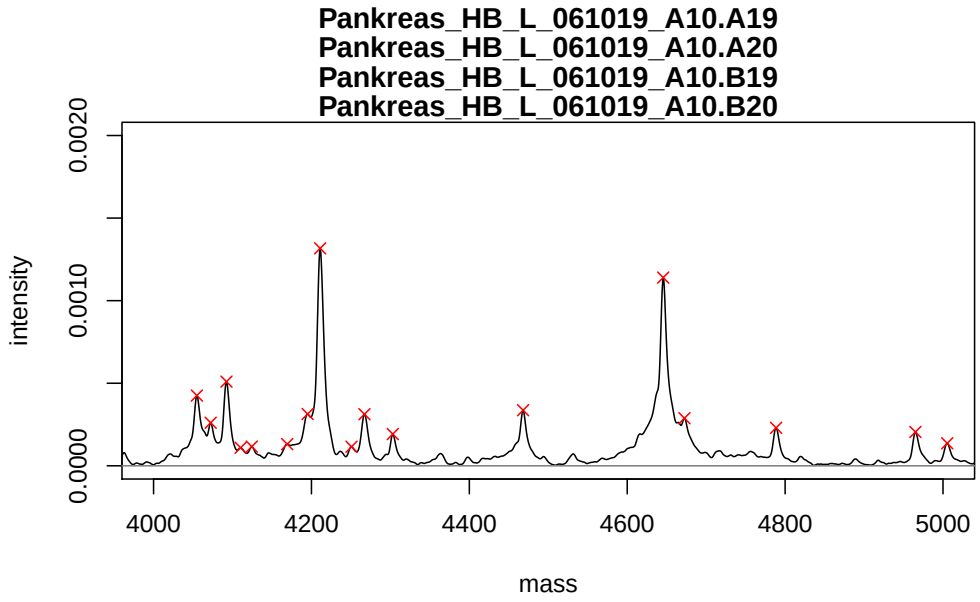
```
noise <- estimateNoise(avgSpectra[[1]])
plot(avgSpectra[[1]], xlim=c(4000, 5000), ylim=c(0, 0.002))
lines(noise, col="red")                        # SNR == 1
lines(noise[, 1], 2*noise[, 2], col="blue") # SNR == 2
```



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)
```



5.8 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
```

5.9 Diagonal Discriminant Analysis

We finish the `MALDIquant` preprocessing and use the *diagonal discriminant analysis* (DDA) function of `sda` (Ahdesmäki and Strimmer, 2010) to find the most important peaks.

```
library("sda")
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: 166
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): 0.107
```

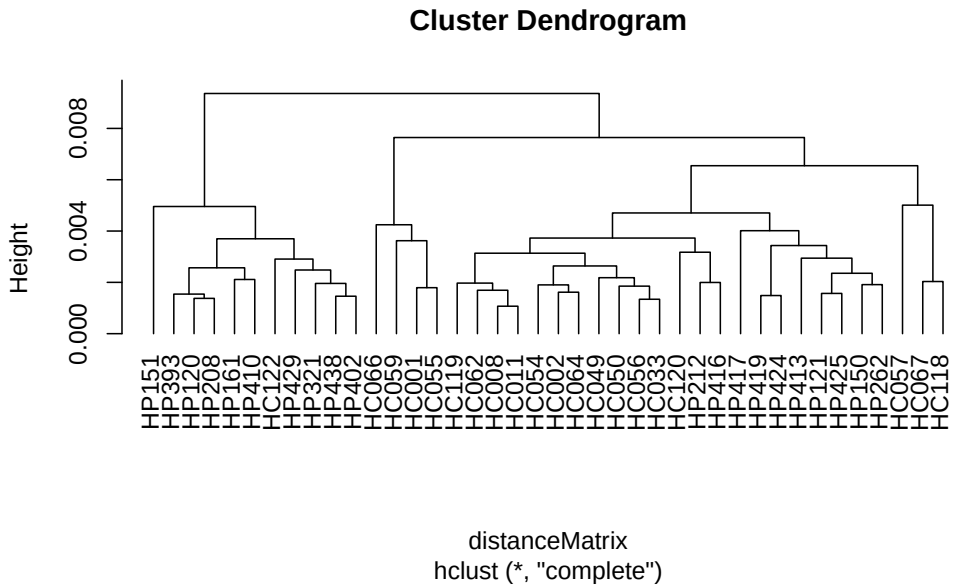
5.10 Hierarchical Clustering

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")
```

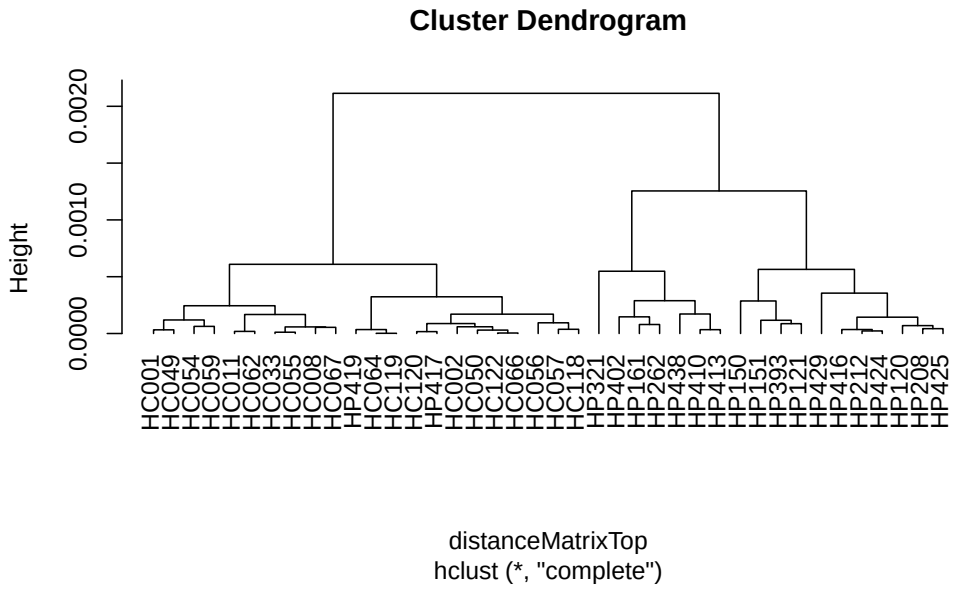
	idx	score	t.cancer	t.control
8936.97236585095	158.00	90.69	9.52	-9.52
4468.06600951353	116.00	80.80	8.99	-8.99
8868.2678310697	157.00	80.06	8.95	-8.95
4494.80267780907	117.00	67.00	8.19	-8.19
8989.20382965523	159.00	66.19	8.14	-8.14
5864.49053296298	135.00	37.56	-6.13	6.13
5906.17351903972	136.00	34.43	-5.87	5.87
2022.94475790442	49.00	33.30	5.77	-5.77
5945.5697657874	137.00	32.66	-5.71	5.71
1866.16591692443	44.00	32.12	5.67	-5.67

```
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)
```



5.11 Cross Validation

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

```
library("crossval")
# 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
0.9500000 0.9000000 1.0000000 1.0000000 0.9090909       Inf
```

5.12 Summary

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

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 1.11.15, 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

Ahdesmäki, M. and Strimmer, K. (2010). Feature selection in omics prediction problems using cat scores and false nondiscovery rate control. *The Annals of Applied Statistics*, 4(1):503–519.

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.

Ryan, C. G., Clayton, E., Griffin, W. L., Sie, S. H., and Cousens, D. R. (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*, 34:396–402.

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36:1627–1639.

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.