A short tutorial on using *metaX* for high-throughput mass spectrometry-based metabolomic data analysis

Bo Wen

March 22, 2016

Contents

1 Introduction

The *metaX* package provides a integrated pipeline for mass spectrometry-based metabolomic data analysis. It includes the stages peak detection, data preprocessing, normalization, missing value imputation, univariate statistical analysis, multivariate statistical analysis such as PCA and PLS-DA, metabolite identification, pathway analysis, power analysis, feature selection and modeling, data quality assessment and HTML-based report generation. This document describes how to use the function included in the R package *metaX*.

2 Example data

We are going to use a dataset from the reference [1]. This data can be accessed through the *faahKO* package. The samples in this data set can be divided into two groups (group knockout or KO, group wild type or WT) which each group includes six samples.

3 Using metaX

3.1 Data import and parameter setting

The first step in the *metaX* pipeline is the definition of a sample list file, that provides the file names (sample), batch number (batch), sample class (class) and the sample injection order (order). An example sample list file is shown below:

```
sampleListFile <- system.file("extdata/faahKO_sampleList.txt",
                              package = "metaX")
samList <- read.delim(sampleListFile)
print(samList)
```

```
##    sample batch class order
## 1    ko15     1    KO     1
## 2    ko16     1    KO     2
## 3    ko18     1    KO     3
## 4    ko19     1    KO     4
## 5    ko21     1    KO     5
## 6    ko22     1    KO     6
## 7    wt15     1    WT     7
## 8    wt16     1    WT     8
## 9    wt18     1    WT     9
## 10   wt19     1    WT    10
## 11   wt21     1    WT    11
## 12   wt22     1    WT    12
```

Please note that if the sample list file contains quality control (QC) sample, the value in the column of class must be "NA". Except the sample list file, the user also needs to provide the MS data (mzML, mzXML or CDF format) or a peak list file which generated by *XCMS*, MZmine [2] or other software which can be used for peak picking. If the user provides MS data, *metaX* uses the *XCMS* to perform peak picking. In this situation, the MS data must be placed in two subdirectories of a single folder like below:

```
list.files(system.file("cdf", package = "faahKO"),
           recursive = TRUE,full.names = TRUE)
```

```
##  [1] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/KO/ko15.CDF"
##  [2] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/KO/ko16.CDF"
##  [3] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/KO/ko18.CDF"
##  [4] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/KO/ko19.CDF"
##  [5] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/KO/ko21.CDF"
##  [6] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/KO/ko22.CDF"
##  [7] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/WT/wt15.CDF"
##  [8] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/WT/wt16.CDF"
##  [9] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/WT/wt18.CDF"
```

```
## [10] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/WT/wt19.CDF"
## [11] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/WT/wt21.CDF"
## [12] "/home/biocbuild/bbs-3.2-bioc/R/library/faahKO/cdf/WT/wt22.CDF"
```

In the *metaX* package, it uses a metaXpara-class object to manage the file path information and other parameters for data processing. We can set the input files path like below:

```
## create a metaXpara-class object
#library("metaX")
para <- new("metaXpara")
## set the MS data path
dir.case(para) <- system.file("cdf/KO", package = "faahKO")
dir.ctrl(para) <- system.file("cdf/WT", package = "faahKO")

## set the sample list file path
sampleListFile(para) <- sampleListFile
```

Usually, the user also needs to set several other parameters for data analysis:

1. Peak picking. If the user wants to use *metaX* to do the peak picking, several parameters related to peak picking must be set.

   ```
   ## set parameters for peak picking
   xcmsSet.peakwidth(para) <- c(20,50)
   xcmsSet.snthresh(para) <- 10
   xcmsSet.prefilter(para) <- c(3,100)
   xcmsSet.noise(para) <- 0
   xcmsSet.nSlaves(para) <- 4
   ```

   For the complete parameters, please see the help page of metaXpara-class.
2. Missing value imputation. Missing values is a common phenomenon in a typical quantitative metabolomics dataset. There are several methods provided by *metaX* to process the missing value. Currently, we implemented a variety of methods which enable users to automatically perform missing value imputation by Probabilistic PCA (PPCA), Bayesian PCA (BPCA), k nearest-neighbor (KNN) missForest and Singular Value Decomposition Imputation (SVDImpute).

   ```
   ## bpca, svdImpute, knn, rf
   missValueImputeMethod(para) <- "knn"
   ```

3. normalization. Currently, we implemented several methods to perform data normalization, such as the QC-RLSC, sum, VSN, probabilistic quotient normalization (PQN), quantiles and robust quantiles.

   ```
   ## bpca, svdImpute, knn, rf
   missValueImputeMethod(para) <- "knn"
   ```

4. set the comparison groups. We can use the following method to set the comparison groups:

   ```
   ## set the comparison groups
   ratioPairs(para) <- "KO:WT"
   ```

If multiple comparison groups must be set in a single analysis, the user can set the "para@ratioPairs" like "A:B;C:B;D:B", each comparison group is separated by semicolon.

5. output parameters. The user can set the output directory and the prefix of the output files as below:

```
## set the output parameters
outdir(para) <- "test"
prefix(para) <- "metaX"
```

3.2 Run metaX

The function metaXpipe automates the whole data analysis process.

```
plsdaPara <- new("plsDAPara")
res <- metaXpipe(para = para,plsdaPara = plsdaPara,
                 cvFilter = 0.2, remveOutlier = TRUE)
```

After the analysis has completed, the file "index.html" in the output directory can be opened in a web browser to access report generated.

Session information

All software and respective versions used to produce this document are listed below.

- R version 3.2.4 Revised (2016-03-16 r70336), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, utils
- Other packages: Biobase 2.30.0, BiocGenerics 0.16.1, ProtGenerics 1.2.1, Rcpp 0.12.3, SSPA 2.10.0, VennDiagram 1.6.16, faahKO 1.10.0, futile.logger 1.4.1, lattice 0.20-33, limma 3.26.9, metaX 1.0.3, mzR 2.4.1, pROC 1.8, qvalue 2.2.2, xcms 1.46.0
- Loaded via a namespace (and not attached): BBmisc 1.9, BiocInstaller 1.20.1, BiocStyle 1.8.0, CAMERA 1.26.0, DBI 0.3.1, DiffCorr 0.4.1, DiscriMiner 0.1-29, Formula 1.2-1, Hmisc 3.17-2, MASS 7.3-45, Matrix 1.2-4, MatrixModels 0.4-1, Nozzle.R1 1.1-1, R6 2.1.2, RBGL 1.46.0, RColorBrewer 1.1-2, RCurl 1.95-4.8, SparseM 1.7, acepack 1.3-3.3, affy 1.48.0, affyio 1.40.0, ape 3.4, assertthat 0.1, backports 1.0.2, bitops 1.0-6, boot 1.3-18, bootstrap 2015.2, car 2.1-1, caret 6.0-64, checkmate 1.7.3, chron 2.3-47, cluster 2.0.3, codetools 0.2-14, colorspace 1.2-6, corpcor 1.6.8, data.table 1.9.6, doParallel 1.0.10, dplyr 0.4.3, ellipse 0.3-8, evaluate 0.8.3, fdrtool 1.2.15, foreach 1.4.3, foreign 0.8-66, formatR 1.3, futile.options 1.0.0, ggplot2 2.1.0, graph 1.48.0, gridExtra 2.2.1, gtable 0.2.0, highr 0.5.1, igraph 1.0.1, impute 1.44.0, iterators 1.0.8, itertools 0.1-3, knitr 1.12.3, lambda.r 1.1.7, latticeExtra 0.6-28, lme4 1.1-11, magrittr 1.5, mgcv 1.8-12, minqa 1.2.4, missForest 1.4, mixOmics 5.2.0, multtest 2.26.0,

munsell 0.4.3, nlme 3.1-126, nloptr 1.0.4, nnet 7.3-12, pbkrtest 0.4-6, pcaMethods 1.60.0, pheatmap 1.0.8, pls 2.5-0, plyr 1.8.3, preprocessCore 1.32.0, quantreg 5.21, randomForest 4.6-12, reshape2 1.4.1, rgl 0.95.1441, rpart 4.1-10, scales 0.4.0, scatterplot3d 0.3-36, splines 3.2.4, stats4 3.2.4, stringi 1.0-1, stringr 1.0.0, survival 2.38-3, tools 3.2.4, vsn 3.38.0, zlibbioc 1.16.0

References

[1] Alan Saghatelian, Sunia A Trauger, Elizabeth J Want, Edward G Hawkins, Gary Siuzdak, and Benjamin F Cravatt. Assignment of endogenous substrates to enzymes by global metabolite profiling. *Biochemistry*, 43(45):14332–14339, 2004.

[2] Mikko Katajamaa, Jarkko Miettinen, and Matej Orešič. Mzmine: toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics*, 22(5):634–636, 2006.