MetMSLine Basics

WMB Edmands

November 12, 2015

The following illustrates the automated MetMSLine workflow with example data for those less familiar with R and the command line interface. R is primarily an interactive programming language, however a degree of automation can be acheived by the usage of wrapper functions, that is software functions that 'wrap' together several other software functions.

Getting started.

Install all of the latest package dependencies needed for proper/ complete functionality of MetMSLine. Copy and paste the following command into R and hit enter:

```
install.packages(c( 'ff', 'dynamicTreeCut', 'data.table', 'reshape2', 'foreach'))
```

This may take several minutes.

Read in example peak table and co-variates and pre-process the data. Run the following code in the same R session at the command line (i.e. copy and paste the lines and hit enter):

```
# read in table of synthetic MS1 metabolomic profiling data
peakTable <- system.file("extdata", "synthetic_MS_data.csv", package = "MetMSLine")
peakTable <- read.csv(peakTable, header=T, stringsAsFactors=F)


# load synthetic co-variates table in comma delimited csv file
coVariates <- system.file("extdata", "synthetic_coVar_data.csv", package = "MetMSLine")
coVariates <- read.csv(coVariates, header=T)
```

Next we need to establish the names of the samples, these observation names will be used to identify the correct column names in the peak table.

By entering the command colnames we can see all of the names of the columns in our peak table:

```
colnames(peakTable)
```

```
##   [1] "EIC"              "mzmed"            "rtmed"
##   [4] "name"             "fold"             "tstat"
##   [7] "pvalue"           "anova"            "mzmin"
##  [10] "mzmax"            "rtmin"            "rtmax"
##  [13] "npeaks"           "Blanks"           "QCs"
##  [16] "Samples"          "metlin"           "QC_1_1"
##  [19] "QC_2_2"           "QC_3_3"           "QC_4_4"
##  [22] "QC_5_5"           "QC_6_6"           "QC_7_7"
##  [25] "QC_8_8"           "QC_9_9"           "QC_10_10"
##  [28] "sample_90148_11"  "sample_20692_12"  "sample_105762_13"
##  [31] "sample_80259_14"  "QC_11_15"         "sample_75940_16"
##  [34] "sample_72889_17"  "sample_83233_18"  "sample_53839_19"
##  [37] "QC_12_20"         "sample_2492_21"   "sample_61966_22"
```

```
##  [40] "sample_84401_23"  "sample_16265_24"  "QC_13_25"
##  [43] "sample_65762_26"  "sample_69316_27"  "sample_88848_28"
##  [46] "sample_25450_29"  "QC_14_30"         "sample_63664_31"
##  [49] "sample_65099_32"  "sample_22679_33"  "sample_85614_34"
##  [52] "QC_15_35"         "sample_6115_36"   "sample_31433_37"
##  [55] "sample_63088_38"  "sample_44970_39"  "QC_16_40"
##  [58] "sample_68093_41"  "sample_46969_42"  "sample_81427_43"
##  [61] "sample_41979_44"  "QC_17_45"         "sample_35776_46"
##  [64] "sample_74037_47"  "sample_24059_48"  "sample_35644_49"
##  [67] "QC_18_50"         "sample_66742_51"  "sample_26783_52"
##  [70] "sample_59605_53"  "sample_26363_54"  "QC_19_55"
##  [73] "sample_105320_56" "sample_80224_57"  "sample_60050_58"
##  [76] "sample_44059_59"  "QC_20_60"         "sample_85303_61"
##  [79] "sample_107459_62" "sample_79428_63"  "sample_102682_64"
##  [82] "QC_10_65"         "sample_73599_66"  "sample_22777_67"
##  [85] "sample_1587_68"   "sample_68869_69"  "QC_11_70"
##  [88] "sample_10915_71"  "sample_99476_72"  "sample_14240_73"
##  [91] "sample_68158_74"  "QC_12_75"         "sample_44297_76"
##  [94] "sample_102649_77" "sample_43569_78"  "sample_110133_79"
##  [97] "QC_13_80"         "sample_82261_81"  "sample_91768_82"
## [100] "sample_75524_83"  "sample_41343_84"  "QC_14_85"
## [103] "sample_10078_86"  "sample_66518_87"  "sample_90107_88"
## [106] "sample_27336_89"  "QC_15_90"         "sample_111469_91"
## [109] "sample_89994_92"  "sample_91108_93"  "sample_48921_94"
## [112] "QC_16_95"         "sample_3918_96"   "sample_111304_97"
## [115] "sample_45723_98"  "sample_19738_99"  "QC_17_100"
## [118] "sample_18558_101" "sample_85913_102" "sample_21652_103"
## [121] "sample_59941_104" "QC_18_105"        "sample_36849_106"
## [124] "sample_20031_107" "sample_50189_108" "sample_43370_109"
## [127] "blank_1_110"      "blank_2_111"      "blank_3_112"
```

The synthetic peak table appears very similar to output from the XCMS peak picking software (specifically the **?xcms::diffreport** function).

We can see that there are many columns containing important information about each LC-MS variable such as the median mass-to-charge ("mzmed") and the median retention time ("rtmed") for example. Regardless of peak picking software (xcms or otherwise) a peak-picker output table in its most basic interpretation will almost always consist of rows of LC-MS variables and columns corresponding LC-MS variable information and sample peak intensity/ height information.

In order for MetMSLine to work it must know the column names which correspond to the observation/ sample peak intensity information. In certain cases such as lowess smoothing MetMSLine must also be supplied with the names of quality control samples also in order to distinguish these from other samples.

We can see that from column 18 to the last column there are observation columns. Within which we can see quality controls containing the character string "QC_", blanks (i.e. negative controls) containing the character string "blank_" and also samples containing the character string "sample_". These unique character strings can be then be used to quickly identify the correct column names to supply to MetMSLine within your R session, like so:

```
# observation names (i.e. sample names) by regular expression (?grep)
# all observation names
obsNames <- colnames(peakTable)[grep("QC_|sample_|blank_", colnames(peakTable))]
# print all the observation names
obsNames
```

```
##   [1] "QC_1_1"           "QC_2_2"           "QC_3_3"
##   [4] "QC_4_4"           "QC_5_5"           "QC_6_6"
##   [7] "QC_7_7"           "QC_8_8"           "QC_9_9"
##  [10] "QC_10_10"         "sample_90148_11"  "sample_20692_12"
##  [13] "sample_105762_13" "sample_80259_14"  "QC_11_15"
##  [16] "sample_75940_16"  "sample_72889_17"  "sample_83233_18"
##  [19] "sample_53839_19"  "QC_12_20"         "sample_2492_21"
##  [22] "sample_61966_22"  "sample_84401_23"  "sample_16265_24"
##  [25] "QC_13_25"         "sample_65762_26"  "sample_69316_27"
##  [28] "sample_88848_28"  "sample_25450_29"  "QC_14_30"
##  [31] "sample_63664_31"  "sample_65099_32"  "sample_22679_33"
##  [34] "sample_85614_34"  "QC_15_35"         "sample_6115_36"
##  [37] "sample_31433_37"  "sample_63088_38"  "sample_44970_39"
##  [40] "QC_16_40"         "sample_68093_41"  "sample_46969_42"
##  [43] "sample_81427_43"  "sample_41979_44"  "QC_17_45"
##  [46] "sample_35776_46"  "sample_74037_47"  "sample_24059_48"
##  [49] "sample_35644_49"  "QC_18_50"         "sample_66742_51"
##  [52] "sample_26783_52"  "sample_59605_53"  "sample_26363_54"
##  [55] "QC_19_55"         "sample_105320_56" "sample_80224_57"
##  [58] "sample_60050_58"  "sample_44059_59"  "QC_20_60"
##  [61] "sample_85303_61"  "sample_107459_62" "sample_79428_63"
##  [64] "sample_102682_64" "QC_10_65"         "sample_73599_66"
##  [67] "sample_22777_67"  "sample_1587_68"   "sample_68869_69"
##  [70] "QC_11_70"         "sample_10915_71"  "sample_99476_72"
##  [73] "sample_14240_73"  "sample_68158_74"  "QC_12_75"
##  [76] "sample_44297_76"  "sample_102649_77" "sample_43569_78"
##  [79] "sample_110133_79" "QC_13_80"         "sample_82261_81"
##  [82] "sample_91768_82"  "sample_75524_83"  "sample_41343_84"
##  [85] "QC_14_85"         "sample_10078_86"  "sample_66518_87"
##  [88] "sample_90107_88"  "sample_27336_89"  "QC_15_90"
##  [91] "sample_111469_91" "sample_89994_92"  "sample_91108_93"
##  [94] "sample_48921_94"  "QC_16_95"         "sample_3918_96"
##  [97] "sample_111304_97" "sample_45723_98"  "sample_19738_99"
## [100] "QC_17_100"        "sample_18558_101" "sample_85913_102"
## [103] "sample_21652_103" "sample_59941_104" "QC_18_105"
## [106] "sample_36849_106" "sample_20031_107" "sample_50189_108"
## [109] "sample_43370_109" "blank_1_110"      "blank_2_111"
## [112] "blank_3_112"
```

We can also identify the quality control samples and assign them to a new variable qcNames

```
qcNames <- colnames(peakTable)[grep("QC_", colnames(peakTable))]
# print the QC names
qcNames
```

```
##  [1] "QC_1_1"    "QC_2_2"    "QC_3_3"    "QC_4_4"    "QC_5_5"
##  [6] "QC_6_6"    "QC_7_7"    "QC_8_8"    "QC_9_9"    "QC_10_10"
## [11] "QC_11_15"  "QC_12_20"  "QC_13_25"  "QC_14_30"  "QC_15_35"
## [16] "QC_16_40"  "QC_17_45"  "QC_18_50"  "QC_19_55"  "QC_20_60"
## [21] "QC_10_65"  "QC_11_70"  "QC_12_75"  "QC_13_80"  "QC_14_85"
## [26] "QC_15_90"  "QC_16_95"  "QC_17_100" "QC_18_105"
```

Then the sample names (assigned to sampNames)

```
sampNames <- colnames(peakTable)[grep("sample_", colnames(peakTable))]
# print the sample names
sampNames
```

```
##  [1] "sample_90148_11"  "sample_20692_12"  "sample_105762_13"
##  [4] "sample_80259_14"  "sample_75940_16"  "sample_72889_17"
##  [7] "sample_83233_18"  "sample_53839_19"  "sample_2492_21"
## [10] "sample_61966_22"  "sample_84401_23"  "sample_16265_24"
## [13] "sample_65762_26"  "sample_69316_27"  "sample_88848_28"
## [16] "sample_25450_29"  "sample_63664_31"  "sample_65099_32"
## [19] "sample_22679_33"  "sample_85614_34"  "sample_6115_36"
## [22] "sample_31433_37"  "sample_63088_38"  "sample_44970_39"
## [25] "sample_68093_41"  "sample_46969_42"  "sample_81427_43"
## [28] "sample_41979_44"  "sample_35776_46"  "sample_74037_47"
## [31] "sample_24059_48"  "sample_35644_49"  "sample_66742_51"
## [34] "sample_26783_52"  "sample_59605_53"  "sample_26363_54"
## [37] "sample_105320_56" "sample_80224_57"  "sample_60050_58"
## [40] "sample_44059_59"  "sample_85303_61"  "sample_107459_62"
## [43] "sample_79428_63"  "sample_102682_64" "sample_73599_66"
## [46] "sample_22777_67"  "sample_1587_68"   "sample_68869_69"
## [49] "sample_10915_71"  "sample_99476_72"  "sample_14240_73"
## [52] "sample_68158_74"  "sample_44297_76"  "sample_102649_77"
## [55] "sample_43569_78"  "sample_110133_79" "sample_82261_81"
## [58] "sample_91768_82"  "sample_75524_83"  "sample_41343_84"
## [61] "sample_10078_86"  "sample_66518_87"  "sample_90107_88"
## [64] "sample_27336_89"  "sample_111469_91" "sample_89994_92"
## [67] "sample_91108_93"  "sample_48921_94"  "sample_3918_96"
## [70] "sample_111304_97" "sample_45723_98"  "sample_19738_99"
## [73] "sample_18558_101" "sample_85913_102" "sample_21652_103"
## [76] "sample_59941_104" "sample_36849_106" "sample_20031_107"
## [79] "sample_50189_108" "sample_43370_109"
```

Then finally the blank sample names (assigned to blankNames).

```
blankNames <- colnames(peakTable)[grep("blank_", colnames(peakTable))]
# print the blank names
blankNames
```

```
## [1] "blank_1_110" "blank_2_111" "blank_3_112"
```

Now that we have identified our observation column names we can now start the MetMSLine process by supplying the necessary arguments to the wrapper functions.

1. LC-MS peak table preprocessing.

```
# detect number of cores using parallel package
nCores <- parallel::detectCores()
# conduct LC-MS data preprocessing
preProc_peakTable <- preProc(peakTable, obsNames, sampNames, qcNames, blankNames,
                             cvThresh=30, nCores=nCores)
```

```
## zero filling with half the minimum non-zero value
##
## Median fold change normalization...
##
## applying LOESS fit (7-fold CV) and signal drift/ attenuation smoothing...
##
## Loading required package: foreach
## Starting SNOW cluster with 8 local sockets...
##
## 7-fold cross validaton loess fitting will be applied to 5000 LC-MS features. Please wait...
##
## 103 (2.06%) of the LC-MS variables contain one of more negative values following loess smoothing...
##
## A column of logicals "negVals" will be added to the returned table indicating these...
##
## Please examine and remove these potentially problematic features before proceeding with further analy
##
##
## removing 103 LC-MS features containing negative values following loess smoothing...
##
## performing blank substraction using...
## 80 samples
## 3 blanks
##
## 4897 (100%) of the LC-MS features were above the mean sample:blank fold change threshold of 2
##
## calculating coefficient of variation...
##
## 3719 (75.9%) features were below the CV% threshold of 30
##
## log transforming to the base 2.718...
```

The table returned from the preProc function contains three additional columns:

```
tail(colnames(preProc_peakTable), n=3)
```

```
## [1] "smoothSpanLoessFit" "meanFCsampBlank"    "coeffVar"
```

1. smoothSpanLoessFit: column contains the optimal loess smooth span parameter identified by 7-fold cross validation for each LC-MS variable.
2. meanFCsampBlank: the fold change value of samples:blanks.
3. coeffVar: the coefficient of variation (CV%) of the repeat pooled QC injections for each LC-MS variable.

Furthermore, the original peakTable has also been subset and LC-MS variables which were below the sample:blank fold change threshold (the default is >2 fold) or above the CV% threshold set (the default is 30%).

if you want to find out more about this wrapper function you can simply type ?MetMSLine::preProc in your R console and hit enter to see the help page.

2. PCA based automated outlier detection and cluster identification.