Advanced topics: Customizing arrayQualityMetrics reports and programmatic processing of the output

Audrey Kauffmann, Wolfgang Huber

April 16, 2015

Contents

1 Introduction

If you are new to this package, then please consult the vignette *Introduction: microarray quality assessment with arrayQualityMetrics*.

This vignette addresses advanced topics. It explains how to customize the report by selecting specific modules and sections, or by adding your own ones. Furthermore, we will see how to (programmatically) postprocess the results of the outlier detection, or how to adapt the detection criteria to your needs.

Terminology: In the documentation of this package, we refer to a *module* as a self-contained element of a report that investigates one particular quality metric. A module consists of a figure and an explanatory text. It may also contain a data structure (an object of class *outlierDetection*) that marks a subset of the arrays in the dataset as outliers according to the particular metric investigated in the module. We refer to a *section* as a collection of one or more modules that are thematically related.

For the following examples, let us load the needed packages and some data.

```
library("arrayQualityMetrics")
library("ALLMLL")
data("MLL.A")
```

2 Data preparation

Some of the computations that are needed for the modules are common between several modules, and thus we perform them once, beforehand. These functions are called `prepdata` and `prepaffy`, and we refer to their manual page for details. For example,

```
preparedData = prepdata(expressionset = MLL.A,
                        intgroup = c(),
                        do.logtransform = TRUE)
```

The arguments `intgroup` and `do.logtransform` are the same as for `arrayQualityMetrics`, but in `prepdata` they have no defaults, so we need to set them explicitely.

3 Module generating functions

The package contains a variety of functions that compute modules, and they are listed on a manual page which you can access by typing:

```
?aqm.boxplot
```

Here, let us create a report with only two quality metrics modules: boxplots and density plots.

```
bo = aqm.boxplot(preparedData)
de = aqm.density(preparedData)
qm = list("Boxplot" = bo, "Density" = de)
```

The objects `bo` and `de` are of class *aqmReportModule*; please consult the class' manual page for more information.

If you want to create your own modules, please have a look at the code for the various existing functions for this purpose, and adapt it. The function `aqm.pca` is a good place to start.

4 Outlier detection

Some of the modules perform outlier detection. For instance, in the default report produced by `arrayQualityMetrics`, the module headed *Boxplots* is followed by one headed *Outlier detection for Boxplots*. In the corresponding *aqmReportModule* object, this is reflected by a non-trivial value for the slot named `outliers`:

```
bo@outliers
```

```
## An object of class "outlierDetection"
## Slot "statistic":
##    JD-ALD009-v5-U133A.CEL    JD-ALD051-v5-U133A.CEL    JD-ALD052-v5-U133A.CEL
##                 0.0689350                 0.1044575                 0.1595225
##    JD-ALD057-v5-U133A.CEL    JD-ALD078-v5-U133A.CEL    JD-ALD180-v5-U133A.CEL
##                 0.0578675                 0.1180275                 0.2320050
## JD-ALD226-NA-v5-U133A.CEL    JD-ALD232-v5-U133A.CEL    JD-ALD237-v5-U133A.CEL
##                 0.2213400                 0.0916450                 0.1539975
```

```
##    JD-ALD244-v5-U133A.CEL    JD-ALD294-v5-U133A.CEL    JD-ALD380-v5-U133A.CEL
##                 0.1352750                 0.3778225                 0.0493600
##    JD-ALD381-v5-U133A.CEL    JD-ALD384-v5-U133A.CEL    JD-ALD385-v5-U133A.CEL
##                 0.0950700                 0.4442150                 0.1664475
##    JD-ALD420-v5-U133A.CEL    JD-ALD421-v5-U133A.CEL    JD-ALD431-v5-U133A.CEL
##                 0.1053325                 0.1261875                 0.1971625
##    JD-ALD433-v5-U133A.CEL    JD-ALD520-v5-U133A.CEL
##                 0.1528250                 0.0598975
##
## Slot "threshold":
## JD-ALD385-v5-U133A.CEL
##             0.3144763
##
## Slot "which":
## JD-ALD294-v5-U133A.CEL JD-ALD384-v5-U133A.CEL
##                     11                     14
##
## Slot "description":
## [1] "Kolmogorov-Smirnov statistic <i>K<sub>a</sub></i>"
## [2] "data-driven"
```

The slot named `statistic` contains, for each array, a single number based on which outlier detection is performed. For instance, in the case of `bo`, the slot `bo@outliers@statistic` is the Kolmogorov-Smirnov statistic for the comparison between each array's intensity distribution and the distribution of the pooled data. The slot `threshold` contains the threshold against which the valuess of `statistic` were compared. Arrays with a value of `statistic` greather than `threshold` are called outliers. Their indices are listed in the vector `which`. Finally, the slot `description` contains a textual description of the definition of `statistic` and indicates how the `threshold` was chosen.

The actual details of outlier detection are different for each report module, and are documented in the figure caption of the report module. For more information, please look at the code of the report module generating function of interest – for instance, at the first few lines of the `boxplot` function. The code there uses the helper functions `outliers` and `boxplotOutliers`, which are documented in their manual pages.

5 Rendering the report

A report is rendered by calling the function `aqm.writereport` with a list of *aqmReportModule* objects.

```
outdir = tempdir()
aqm.writereport(modules = qm, reporttitle = "My example", outdir = outdir,
                arrayTable = pData(MLL.A))
outdir
```

```
## [1] "E:\\biocbld\\bbs-3.1-bioc\\tmpdir\\RtmpC6uUss"
```

Point your browser to the `index.html` file in that directory.

Session Info

- R version 3.2.0 RC (2015-04-08 r68161), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, utils
- Other packages: ALLMLL 1.7.1, Biobase 2.28.0, BiocGenerics 0.14.0, affy 1.46.0, arrayQualityMetrics 3.24.0
- Loaded via a namespace (and not attached): AnnotationDbi 1.30.0, BeadDataPackR 1.20.0, BiocInstaller 1.18.1, BiocStyle 1.6.0, Biostrings 2.36.0, Cairo 1.5-6, DBI 0.3.1, Formula 1.2-1, GenomeInfoDb 1.4.0, GenomicRanges 1.20.0, Hmisc 3.15-0, IRanges 2.2.0, MASS 7.3-40, RColorBrewer 1.1-2, RJSONIO 1.3-0, RSQLite 1.0.0, Rcpp 0.11.5, S4Vectors 0.6.0, SVGAnnotation 0.93-1, XML 3.98-1.1, XVector 0.8.0, acepack 1.3-3.3, affyPLM 1.44.0, affyio 1.36.0, annotate 1.46.0, base64 1.1, beadarray 2.18.0, cluster 2.0.1, colorspace 1.2-6, digest 0.6.8, evaluate 0.6, foreign 0.8-63, formatR 1.1, gcrma 2.40.0, genefilter 1.50.0, ggplot2 1.0.1, grid 3.2.0, gridSVG 1.4-3, gtable 0.1.2, highr 0.4.1, hwriter 1.3.2, illuminaio 0.10.0, knitr 1.9, lattice 0.20-31, latticeExtra 0.6-26, limma 3.24.0, munsell 0.4.2, nnet 7.3-9, plyr 1.8.1, preprocessCore 1.30.0, proto 0.3-10, reshape2 1.4.1, rpart 4.1-9, scales 0.2.4, setRNG 2013.9-1, splines 3.2.0, stats4 3.2.0, stringr 0.6.2, survival 2.38-1, tools 3.2.0, vsn 3.36.0, xtable 1.7-4, zlibbioc 1.14.0