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

Audrey Kauffmann, Wolfgang Huber

April 30, 2024

Contents

1 Introduction ................................................. 1
2 Data preparation ........................................... 2
3 Module generating functions ............................... 2
4 Outlier detection ........................................... 3
5 Rendering the report ....................................... 4

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,

```r
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 explicitly.

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:

```r
?aqm.boxplot
```

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

```r
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.
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`:

```r
bo@outliers

## An object of class "outlierDetection"
## Slot "statistic":
## JD-ALD009-v5-U133A.CEL  JD-ALD051-v5-U133A.CEL
## 0.0710425 0.0996975
## JD-ALD052-v5-U133A.CEL  JD-ALD057-v5-U133A.CEL
## 0.1589775 0.0573950
## JD-ALD078-v5-U133A.CEL  JD-ALD180-v5-U133A.CEL
## 0.1178625 0.2296400
## JD-ALD226-NA-v5-U133A.CEL  JD-ALD232-v5-U133A.CEL
## 0.2187325 0.0905925
## JD-ALD237-v5-U133A.CEL  JD-ALD244-v5-U133A.CEL
## 0.1493675 0.1296925
## JD-ALD294-v5-U133A.CEL  JD-ALD380-v5-U133A.CEL
## 0.3760475 0.0509450
## JD-ALD381-v5-U133A.CEL  JD-ALD384-v5-U133A.CEL
## 0.0913800 0.4417800
## JD-ALD385-v5-U133A.CEL  JD-ALD420-v5-U133A.CEL
## 0.1653850 0.1042625
## JD-ALD421-v5-U133A.CEL  JD-ALD431-v5-U133A.CEL
## 0.1252625 0.1935600
## JD-ALD433-v5-U133A.CEL  JD-ALD520-v5-U133A.CEL
## 0.1494125 0.0590450
##
## Slot "threshold":
## JD-ALD385-v5-U133A.CEL
## 0.3122019
##
## Slot "which":
## JD-ALD294-v5-U133A.CEL  JD-ALD384-v5-U133A.CEL
## 11 14
##
## Slot "description":
## [1] "Kolmogorov-Smirnov statistic \( K_a \)"
## [2] "data-driven"
```
Advanced topics

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 values of statistic were compared. Arrays with a value of statistic greater 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 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.

```r
outdir = tempdir()
aqm.writereport(modules = qm, reporttitle = "My example", outdir = outdir,
                 arrayTable = pData(MLL.A))
outdir
# [1] "/tmp/Rtmpvbw4v6"
```

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

Session Info

- R version 4.4.0 beta (2024-04-15 r86425), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=en_US.UTF-8, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Time zone: America/New_York
- TZcode source: system (glibc)
- Running under: Ubuntu 22.04.4 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.19-bioc/R/lib/libRblas.so
Advanced topics

- LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: affy 1.82.0, ALLMLL 1.43.0, arrayQualityMetrics 3.60.0, Biobase 2.64.0, BiocGenerics 0.50.0
- Loaded via a namespace (and not attached): affyio 1.74.0, affyPLM 1.80.0, annotate 1.82.0, AnnotationDbi 1.66.0, askpass 1.2.0, backports 1.4.1, base64 2.0.1, base64enc 0.1-3, beadarray 2.54.0, BeadDataPackR 1.56.0, BiocManager 1.30.22, BiocStyle 2.32.0, Biostrings 2.72.0, bit 4.0.5, bit64 4.0.5, blob 1.2.4, cachem 1.0.8, checkmate 2.3.1, cli 3.6.2, cluster 2.1.6, codetools 0.2-20, colorspace 2.1-0, compiler 4.4.0, crayon 1.5.2, data.table 1.15.4, DBI 1.2.2, deldir 2.0-4, digest 0.6.35, dplyr 1.1.4, evaluate 0.23, fansi 1.0.6, fastmap 1.1.1, foreign 0.8-86, Formula 1.2-5, gcrma 2.76.0, genefilter 1.86.0, generics 0.1.3, GenomInfoDb 1.40.0, GenomInfoDbData 1.2.12, GenomicRanges 1.56.0, ggplot2 3.5.1, glue 1.7.0, grid 4.4.0, gridExtra 2.3, gridSVG 1.7-5, gtable 0.3.5, hexbin 1.28.3, highr 0.10, Hmisc 5.1-2, htmlTable 2.4.2, htmltools 0.5.8.1, htmlwidgets 1.6.4, httr 1.4.7, hwriter 1.3.2.1, illuminaio 0.46.0, interp 1.1-6, lIRanges 2.38.0, jpeg 0.1-10, jsonlite 1.8.8, KEGGREST 1.44.0, knitr 1.46, lattice 0.22-6, latticeExtra 0.6-30, lifecycle 1.0.4, limma 3.60.0, magrittr 2.0.3, Matrix 1.7-0, MatrixGenerics 1.16.0, matrixStats 1.3.0, memoise 2.0.1, munsell 0.5.1, nnet 7.3-19, openssl 2.1.2, pillar 1.9.0, pkgconfig 2.0.3, plyr 1.8.9, png 0.1-8, preprocessCore 1.66.0, R 6.2.5.1, RColorBrewer 1.1-3, Rcpp 1.0.12, reshape2 1.4.4, rlang 1.1.3, rmarkdown 2.26, rpart 4.1.23, RSQLite 2.3.6, rstudioapi 0.16.0, S4Vectors 0.42.0, scales 1.3.0, setRNG 2024.2-1, splines 4.4.0, stats 4.4.0, stringi 1.8.3, stringr 1.5.1, survival 3.6-4, svglite 2.1.3, systemfonts 1.0.6, tibble 3.2.1, tidyselect 1.2.1, tools 4.4.0, UCSC.utils 1.0.0, utf8 1.2.4, vctrs 0.6.5, vsn 3.72.0, xfun 0.43, XML 3.99-0.16.1, xtable 1.8-4, XVector 0.44.0, yaml 2.3.8, zlibbioc 1.50.0