

# yaqcaffy

October 25, 2011

---

YAQCStats-class      *Class "YAQCStats"*

---

## Description

Holds Quality Control Data for a set of Affymetrix Arrays

## Objects from the Class

Objects can be created by calls of the form `yaqc` providing `AffyBatch` or `ExpressionSet` instances as arguments. `YAQCStats` is a subclass of `QCStats` and uses the `scale.factor`, `average.background`, `percent.present`, `arraytype` and `target` attributes of it's super-class.

## Slots

Class-specific slots:

`log`: Object of class "logical" that specifies if expression values are in log<sub>2</sub> form.

`average.noise`: Object of class "numeric". The average noise for the arrays.

`morespikes`: Object of class "matrix". More spiked in probes (e.g. `r2biob5`, `r2biob3`,...).

`gcos.probes`: Object of class "matrix". GAPDH and  $\beta$ -actin qc probes (e.g. `gapdh 3,5,M`,...) containing the GCOS values.

`bio.calls`: Object of class "matrix". BioB 5',3',M and BioC, BioC 5',3' present/absent/marginal calls .

`objectVersion` Character describing the version of the library used to generate the `YAQCStats` object.

`yaqcControlProbes` Object of class `YaqcControlProbes` that defines the different probes used for the quality control.

See also `QCStats` for slots inherited from super-class.

## Methods

Methods inherited from the super-class:

**target** signature(object = "YAQCStats"): Returns a numeric target value for MAS 5.0 normalization.

**avbg** signature(object = "YAQCStats"): Returns a vector of the average background levels for each array.

**minbg** signature(object = "YAQCStats"): Returns a vector of the minimum background levels for each array.

**percent.present** signature(object = "YAQCStats"): Returns a vector listing the percentage of probesets called present on each array.

**sfs** signature(object = "YAQCStats"): Returns a vector of scale factors for each array (as produced by the MAS 5.0 algorithm).

Class-specific methods:

**isLog** signature(object = "YAQCStats"): Returns a logical specifying if the expression intensities are in log2 from.

**moreSpikeInProbes** signature(object = "YAQCStats"): Returns a matrix of intensities for the internal spike probes.

**gcProbes** signature(object = "YAQCStats"): Returns a matrix of intensities for GAPDH and  $\beta$ -actin probes.

**avns** signature(object = "YAQCStats"): Returns a vector listing the average noise levels for each array.

**bioCalls** signature(object = "YAQCStats"): Returns a matrix of Present(P)/Marginal(M)/Absent(A) calls for the spike probes.

**arrays** signature(object = "YAQCStats"): Returns the names of the arrays in the YAQCStats instance.

**plot** signature(object = "YAQCStats"): visual representation of the qc metrics. (see [yaqc.plot](#) for more details).

**summary** signature(x = "YAQCStats", latex = "logical"): The outliers of the YAQCStats quality control metrics are summarized and returned as a data frame. If latex is set to TRUE (default), the data frame is returned as a latex table (requires the xtable package).

**show** signature(object = "YAQCStats"): displays the content of the object as a data frame.

**merge** signature(x = "YAQCStats", y = "YAQCStats"): merges two compatible YAQCStats objects, i.e. that have the same values for the log, target and arraytype slots.

**arrays** signature(object = "YAQCStats"): shows the array names of an YAQCStats objects.

**objectVersion** signature(object = "YAQCStats"): Returns the version of the yaqcaffy package as a character used to create the given object.

**getYqcControlProbes** signature(object = "YAQCStats"): Returns the [YaqcControlProbes](#) object that has been used to generate the current YAQCStats object.

## Author(s)

Laurent Gatto

**See Also**

[QCStats](#) from package [simpleaffy](#) and [YaqcControlProbes](#).

---

YaqcControlProbes-class

*Class "YaqcControlProbes"*

---

**Description**

The [YAQCStats](#) class stores the probes used for the quality control as a special class, namely the `YaqcControlProbes` class. This class encapsulated the probe names that are used to generate an [YAQCStats](#) object. Objects of this class are created with the [probeSelectionInterface](#) function. The hybridization, labelling and degradation probes are encapsulated in `YaqcBioProbes`, `YaqcSpkProbes` and `YaqcDegProbes` objects respectively. These can be retrieved with their respective accessors, as described below. Furthermore, an `info` function allows to retrieve or set a free text slot to describe the `YaqcControlProbes` object.

**Slots**

- `bio`: Object of class "YaqcBioProbes" encapsulating the 'bio' (BioB5, BioB3, BioBM, BioC5,...) hybridization probes.
- `spk`: Object of class "YaqcSpkProbes" encapsulating the labelling probes (dap5, dap3, dap3, phe5,...).
- `deg`: Object of class "YaqcDegProbes" encapsulating the degradation probes used to assess the 3'/5' ratio.
- `info`: Object of class "character" providing general information about the `YaqcControlProbes` object.

**Methods**

- bio** signature(object = "YaqcControlProbes"): returns the 'bio' (BioB5, BioB3, BioBM, BioC5,...) hybridization probes of the current object, as an `YaqcBioProbes` instance.
- bio** signature(object = "YaqcBioProbes"): returns the 'bio' (BioB5, BioB3, BioBM, BioC5,...) hybridization probes of the current object, as characters.
- spk** signature(object = "YaqcControlProbes"): returns the labelling probes (dap5, dap3, dap3, phe5,...) of the current object, as an `YaqcSpkProbes` instance.
- spk** signature(object = "YaqcSpkProbes"): returns the labelling probes (dap5, dap3, dap3, phe5,...) of the current object, as characters
- deg** signature(object = "YaqcControlProbes"): returns the degradation probes used to assess the 3'/5'ratio, as an `YaqcDegProbes` instance.
- deg** signature(object = "YaqcDegProbes"): returns the degradation probes used to assess the 3'/5'ratio, as characters.
- info** signature(object = "YaqcControlProbes"): returns the information slot of the current object.
- info<-** signature(object = "YaqcControlProbes"): sets the information slot of the current object.
- show** signature(object = "YaqcControlProbes"): shows the current object.

**Author(s)**

Laurent Gatto

**See Also**

[probeSelectionInterface](#) and [YAQCStats](#)

**Examples**

```
showClass("YaqcControlProbes")
showClass("YaqcBioProbes")
showClass("YaqcSpkProbes")
showClass("YaqcDegProbes")
```

---

getAllInt

*Get the summerized MAS5 values for a given spike probe*

---

**Description**

This function retrieves the expression intensities provided by the object of class "YAQCStats" for the probe which names are compatible with the given pattern and return their mean value.

**Usage**

```
getAllInt(YAQCStatsObject, pattern)
```

**Arguments**

```
YAQCStatsObject      an object of type "YAQCStats"
pattern               a pattern used to select the probe names to be used
```

**Value**

An object of type "numeric" in which the mean expression intensities of each array are given.

**Author(s)**

Laurent Gatto

**Examples**

```
## load a dataset
library(affydata)
data(Dilution)
## perform quality control
qc <- yaqc(Dilution)
## get intensities for the biob
## spikes probe sets
getAllInt(qc, "biob")
## or
getAllInt(qc, "b[3|5|m]")
```

---

getBioProbes	<i>Get the names of the Bio spike probes on the array</i>
--------------	---

---

**Description**

This function returns all the AFFX-Bio probes names that are located on the given GeneChip.

**Usage**

```
getBioProbes(object, onlyFirst)
```

**Arguments**

object	An object of type "AffyBatch" or "ExpressionSet"
onlyFirst	Boolean defining of only first or all instances found should be returned. Default is set to TRUE. Warnings are returned if more than one probe is found. The function stops with an error if no probe is found.

**Value**

An object of type "character" with the Affymetrix hybridation (bio) probe names for the given chip type.

**Author(s)**

Laurent Gatto

**See Also**

getSpikeProbes, getRatioProbes

**Examples**

```
## load a dataset
library(affydata)
data(Dilution)
getBioProbes(Dilution)
```

---

getOutliers	<i>Get outliers for the different YAQCStatsObject slots</i>
-------------	---

---

**Description**

This function retrieves the outliers for the different quality control metrics stored in a YAQCStatsObject. Outliers are defined as being outside of the mean +/- 2 stdev range or mean/2, mean\*1.5 for the scale factor.

**Usage**

```
getOutliers(YAQCStatsObject, slot)
```

## Arguments

YAQCStatsObject	an object of type "YAQCStats"
slot	an object of type <code>string</code> describing the slot for which the outliers should be retrieved (see details for possible slot strings)

## Details

The slot strings that can be used are:

**scale factor** "sfs"

**average background** "avbg"

**average noise** "avns"

**percentage present** "pp"

**$\beta$ -actin 3'/5' ratio** "actin"

**GAPDH 3'/5' ratio** "gapdh"

**internal bioB control** "biob"

**internal bioC control** "bioc"

**internal bioD control** "biod"

**Dap spike control** "dap"

**Thr spike control** "thr"

**Phe spike control** "phe"

**Lys spike control** "lys"

## Value

An object of type "numeric" giving the outliers names and values

## Author(s)

Laurent Gatto

## Examples

```
## load data
library(MAQCsubsetAFX)
data(refA)
## create the yaqc object
qobj <- yaqc(refA[, 3:5])
## get outliers for the scale factor
getOutliers(qobj, "sfs")
```

---

getQCRatios                    *Compute qc probe ratios using GCOS intensity values*

---

**Description**

This function computes the 3'/5' ratios of the GAPDh and  $\beta$ -actin qc probes using the GCOS intensity values.

**Usage**

```
getQCRatios(YAQCStatsObject)
```

**Arguments**

YAQCStatsObject  
                  an object of class YAQCStats

**Value**

An object of type "matrix" with two qc ratios per array.

**Author(s)**

Laurent Gatto

**See Also**

getRatioProbes

**Examples**

```
## load a dataset
library(affydata)
data(Dilution)
## create yaqc object
qobj <- yaqc(Dilution)
getQCRatios(qobj)
```

---

getRatioProbes                    *Get the names of degradation control probes on the array*

---

**Description**

This function returns the probes names used for degradation control that are located on the given GeneChip.

**Usage**

```
getRatioProbes(object, onlyFirst)
```

**Arguments**

object	An object of class "AffyBatch" or "ExpressionSet"
onlyFirst	Boolean defining of only first or all instances found should be returned. Default is set to TRUE. Warnings are returned if more than one probe is found. The function stops with an error if no probe is found.

**Value**

An object of type "character" with all the Affymetrix degradation control probe names.

**Author(s)**

Laurent Gatto

**See Also**

getSpikeProbes, getBioProbes

**Examples**

```
library(yaqcaffy)
## load a dataset
library(affydata)
data(Dilution)
getRatioProbes(Dilution)
```

---

getSpikeProbes	<i>Get the names of all spike probes on the array</i>
----------------	---

---

**Description**

This function returns all the spike probes (i.e. BioB-3', BioD-5', Lys-3, ...) that are located on the given GeneChip.

**Usage**

```
getSpikeProbes(object, onlyFirst)
```

**Arguments**

object	An object of type AffyBatch or ExpressionSet.
onlyFirst	Boolean defining of only first or all instances found should be returned. Default is set to TRUE. Warnings are returned if more than one probe is found. The function stops with an error if no probe is found.

**Value**

An object of class character containing all (hybridization and labelling) Affymetrix spike probe names.



**Author(s)**

Laurent Gatto

**See Also**

getBioProbes, getRatiosProbes

**Examples**

```
## load a dataset
library(affydata)
data(Dilution)
getSpikeProbes(Dilution)
```

---

`probeSelectionInterface`*Tcltk Interface to Generate an Instance of YaqcControlProbes for a*

---

**Description**

`probeSelectionInterface` starts a tcltk graphical user interface (GUI) that allows the user to choose the probes to be used for subsequent quality analyses with the `yaqcaffy` package. The probes are selected on basis of the features of a given set of Affymetrix Genechips provided as input. The list of probes can be pre-filtered to display only control probes (i.e starting by AFFX) or all probes on the Genechip can be shown.

**Usage**

```
probeSelectionInterface(object,
  returnVar="yaqcControlProbes",
  filter=TRUE)
```

**Arguments**

<code>object</code>	an object of class <code>AffyBatch</code> or <code>ExpressionSet</code> .
<code>returnVar</code>	a string defining the name of the variable the returned object will be saved as in the global environment. The default variable name is <code>'yaqcControlProbes'</code> . If such a variable name already exists, a warning will be issued and the user can cancel the function.
<code>filter</code>	logical value. If <code>'TRUE'</code> , the feature names of the input object are filtered out (see details). If <code>'FALSE'</code> , all features are listed for all control probes.

**Details**

Three tabs are displayed, one for the hybridization (bio) probes, labelling probes (dap, phe, thr and lys) and the degradation probes (actin and gapdh) respectively. If the user uses the 'Close' button, no return object is saved in the global environment. An object is saved as `returnVar` if the user presses 'Ok'. If such a variable name already exists, a warning will be issued and the user can close the interface and cancel the function.

If filtering is applied, the hybridization menus will list probes that match the given probe (BioB, BioC or BioD) and position (5, 3 or M). Similarly, only matching labelling probes (dap, phe, thr

and lys) and positions will be displayed. As the pattern for the degradation probes are less strict, all the 'AFFX' probes, except those already selected as hybridization and labelling probes, will be displayed in the drop-down menus.

### Value

Returns an object of class `YaqcBioProbes`.

### Author(s)

Laurent Gatto

### Examples

```
## Not run:
  library(affydata)
  data(Dilution)
  probeSelectionInterface(Dilution)

## End (Not run)
```

---

reprodPlot

*Plot human whole genome GeneChips reproductibility*

---

### Description

Compares Affymetrix Human Genome U133 Plus 2.0 Arrays to a subset of the MAQC arrays for a RNA reference.

### Usage

```
reprodPlot(userAffyBatchObject, ref,
           normalize=c("rma", "gcrma", "mas5", "none"),
           main="MAQC reference reproductibility",
           cex, ...)
```

### Arguments

<code>userAffyBatchObject</code>	a set of Human Genomue U133 Plus 2.0 arrays provided as an AffBatch object,
<code>ref</code>	a string ("refA", "refB", "refC", or "refD") defining the RNA reference to compare the userAffyBatchObject to,
<code>normalize</code>	a string defining the algorithm used for data normalization: <code>rma</code> (default) for RMA (as implemented in the <b>affy</b> library), <code>gcrma</code> for GCRMA (as implemented in the <code>gcrma</code> library), <code>mas5</code> for MAS5 (as implemented in the <b>affy</b> library) or 'none' for no normalization,
<code>main</code>	an overall title for the plot,
<code>cex</code>	size of text on the plot,
<code>...</code>	other arguments.

**Details**

The plot shows all the pairwise scatterplots (plotted with **graphics**'s `smoothScatter` function) with Pearson's correlation factor and MAplots (plotted with **affy**'s `ma.plot` function). The subset of the MAQC arrays are 1 randomly chosen .CEL file out of the 5 replicates for the 6 different test site.

**Value**

Outputs a graph on the available graphical device

**Author(s)**

Laurent Gatto

**Examples**

```
## Not run:
## loading data
library(MAQCsubsetAFX)
data(refB)
d<-refB[,1]
## testing the reproductibility against ref A
reprodPlot(d,"refA",normalize="rma")

## End(Not run)
```

---

yaqc.affy

*'YAQCStats' constructor*

---

**Description**

Generate YAQC metrics for Affymetrix data.

**Usage**

```
yaqc.affy(object,
myYaqcControlProbes=NULL,
alphas=NULL,
tgt=100,
tau=0.015,
logged,
verbose)
```

**Arguments**

`object` a object of type `AffyBatch` or `ExpressionSet`.  
`myYaqcControlProbes` an object of type `YaqcControlProbes`. If none is supplied (default behaviour), the control probes are selected automatically. See the `YaqcControlProbes` class for more details `probeSelectionInterface` to generate such an object.

alphas	a numeric of length 2 with the alpha1 and alpha2 values. Alpha1 and alpha2 are thresholds used to define if a given probe should be called present ( $p < \alpha_1$ ), marginal ( $\alpha_1 < p < \alpha_2$ ) or absent ( $\alpha_2 < p$ ), where $p$ is the p-value from the Wilcoxon Signed Rank test used in MAS5. The default is to get these values from simpleaffy's qcdef files (see simpleaffy's vignette for more details) or to use 0.04 and 0.06 as default values.
tgt	the target intensity to which the chips should be scaled (used to calculate the MAS5 intensity values).
tau	used for detection p-value.
logged	to be used with an ExpressionSet object, defining if the expression intensities are logged.
verbose	logical value. If 'TRUE', it writes out some messages indicating progress. If 'FALSE' nothing should be printed.

### Details

Affymetrix recommends a set of quality control metrics to check the quality of GeneChips expression arrays. This function applies the guidelines described in the Affymetrix Microarray Quality Control Consortium (MAQC) protocols to assess the success of the hybridization. See the package vignette for more details.

This function takes a raw (unnormalised) AffyBatch object or an ExpressionSet object. In the first case, it computes MAS5 intensity values, expression calls (see `call.exprs`) and other quality-related metrics to generate an YAQCStats. If an ExpressionSet object is provided, only the  $\beta$ -actin, GAPDH and internal control values are computed.

### Value

An YAQCStats object describing the input object

### Author(s)

Laurent Gatto

### Examples

```
## loading data
library(affydata)
data(Dilution)
## qc analysis
qobj <- yaqc(Dilution)
show(qobj)
```

---

yaqc.plot

*Plots a YAQCStats object*

---

### Description

S4 method to plot an YAQCStats object. `plot(object)` generates a visual summary of the various Affymetrix QC statistics.

**Usage**

```
yaqc.plot(YAQCStatsObject, which=c("all", "sfs", "avbg", "avns", "pp", "gapdh", "actin"))
```

**Arguments**

YAQCStatsObject	An object of class <code>YAQCStats</code> .
which	Which quality metrics should be plotted (all by default). See below for details.
...	Other arguments passed to the respective <code>plot</code> methods.

**Details**

The quality control metrics of the `YAQCStatsObject` are plotted in a serie of graphs with the recommended ranges.

The scale factors are represented through a dot chart and the upper and lower limits are defined with vertical red lines. The other qc metircs are shown using dot plots. For the upper row box plots (average background, average noise, percent present and  $\beta$ -actin and GAPDH ratios, the mean is represented by a dashed red line and the mean  $\pm$  2 stdev by red dotted lines. For the lower box plots, featuring the internal controls, grey boxes defines the mean (middle segment)  $\pm$  2 stdev.

Individual plots can also be generated with the `which` argument: 'sfs' for the scale factor, 'avbg' and 'avns' for the average background and noise, 'pp' for the percentage of present calls, 'gapdh' and 'actin' for the GAPDH and  $\beta$ -actin ratios, 'bio' for the hybridization controls and 'spikes' for the retro-transcription spiked controls.

If the `YAQCStatsObject` has been generated with an `Expression Set` objects, the scale factors, average noise and background and percent present can not be computed and the respective plots are removed from the final graph.

**Author(s)**

Laurent Gatto

**Examples**

```
## load data
library(affydata)
data(Dilution)
## create the yaqc object
## and plot it
qobj <- yaqc(Dilution)
plot(qobj)
```

# Index

## \*Topic **classes**

YaqcControlProbes-class, 3  
YAQCStats-class, 1

## \*Topic **hplot**

yaqc.plot, 12

## \*Topic **methods**

yaqc.affy, 11

## \*Topic **misc**

getAllInt, 4  
getBioProbes, 5  
getOutliers, 5  
getQCRatios, 7  
getRatioProbes, 7  
getSpikeProbes, 8  
probeSelectionInterface, 9  
reprodPlot, 10

arrays (YAQCStats-class), 1

arrays, YAQCStats-method  
(YAQCStats-class), 1

avbg, YAQCStats-method  
(YAQCStats-class), 1

avns (YAQCStats-class), 1

avns, YAQCStats-method  
(YAQCStats-class), 1

bio (YaqcControlProbes-class), 3

bio, YaqcBioProbes-method  
(YaqcControlProbes-class),  
3

bio, YaqcControlProbes-method  
(YaqcControlProbes-class),  
3

bio-methods  
(YaqcControlProbes-class),  
3

bioCalls (YAQCStats-class), 1

bioCalls, YAQCStats-method  
(YAQCStats-class), 1

class:YaqcBioProbes  
(YaqcControlProbes-class),  
3

class:YaqcControlProbes  
(YaqcControlProbes-class),  
3

class:YaqcDegProbes  
(YaqcControlProbes-class),  
3

class:YaqcSpkProbes  
(YaqcControlProbes-class),  
3

class:YAQCStats  
(YAQCStats-class), 1

deg (YaqcControlProbes-class), 3

deg, YaqcControlProbes-method  
(YaqcControlProbes-class),  
3

deg, YaqcDegProbes-method  
(YaqcControlProbes-class),  
3

deg-methods  
(YaqcControlProbes-class),  
3

gcosProbes (YAQCStats-class), 1

gcosProbes, YAQCStats-method  
(YAQCStats-class), 1

gcrma, 10

getAllInt, 4

getBioProbes, 5

getOutliers, 5

getQCRatios, 7

getRatioProbes, 7

getSpikeProbes, 8

getYaqcControlProbes  
(YAQCStats-class), 1

getYaqcControlProbes, YAQCStats-method  
(YAQCStats-class), 1

info (YaqcControlProbes-class), 3

info, YaqcControlProbes-method  
(YaqcControlProbes-class),  
3

info<- (YaqcControlProbes-class),  
3

- info<- , YaqcControlProbes-method  
(*YaqcControlProbes-class*),  
3
- isLog (*YAQCStats-class*), 1
- isLog, YAQCStats-method  
(*YAQCStats-class*), 1
- ma.plot, 11
- merge, YAQCStats, YAQCStats-method  
(*YAQCStats-class*), 1
- merge, YAQCStats-method  
(*YAQCStats-class*), 1
- method:yaqc (*yaqc.affy*), 11
- minbg, YAQCStats-method  
(*YAQCStats-class*), 1
- moreSpikeInProbes  
(*YAQCStats-class*), 1
- moreSpikeInProbes, YAQCStats-method  
(*YAQCStats-class*), 1
- objectVersion (*YAQCStats-class*), 1
- objectVersion, YAQCStats-method  
(*YAQCStats-class*), 1
- percent.present, YAQCStats-method  
(*YAQCStats-class*), 1
- plot, YAQCStats (*yaqc.plot*), 12
- plot, YAQCStats, missing-method  
(*yaqc.plot*), 12
- probeSelectionInterface, 3, 4, 9
- QCStats, 1, 3
- reprodPlot, 10
- sfs, YAQCStats-method  
(*YAQCStats-class*), 1
- show, YaqcBioProbes-method  
(*YaqcControlProbes-class*),  
3
- show, YaqcControlProbes-method  
(*YaqcControlProbes-class*),  
3
- show, YaqcDegProbes-method  
(*YaqcControlProbes-class*),  
3
- show, YaqcSpkProbes-method  
(*YaqcControlProbes-class*),  
3
- show, YAQCStats-method  
(*YAQCStats-class*), 1
- smoothScatter, 11
- spk (*YaqcControlProbes-class*), 3
- spk, YaqcControlProbes-method  
(*YaqcControlProbes-class*),  
3
- spk, YaqcSpkProbes-method  
(*YaqcControlProbes-class*),  
3
- spk-methods  
(*YaqcControlProbes-class*),  
3
- summary, YAQCStats-method  
(*YAQCStats-class*), 1
- target, YAQCStats-method  
(*YAQCStats-class*), 1
- yaqc, 1
- yaqc (*yaqc.affy*), 11
- yaqc, AffyBatch-method  
(*yaqc.affy*), 11
- yaqc, eSet-method (*yaqc.affy*), 11
- yaqc, ExpressionSet-method  
(*yaqc.affy*), 11
- yaqc-method (*yaqc.affy*), 11
- yaqc-methods (*yaqc.affy*), 11
- yaqc.affy, 11
- yaqc.plot, 2, 12
- yaqc.summary (*YAQCStats-class*), 1
- YaqcBioProbes, 10
- YaqcBioProbes-class  
(*YaqcControlProbes-class*),  
3
- YaqcControlProbes, 2, 3
- YaqcControlProbes-class, 3
- YaqcDegProbes-class  
(*YaqcControlProbes-class*),  
3
- YaqcSpkProbes-class  
(*YaqcControlProbes-class*),  
3
- YAQCStats, 2–4, 12, 13
- YAQCStats-class, 1