

Package ‘miRcomp’

March 7, 2025

Version 1.36.0

Date 2017-03-04

Title Tools to assess and compare miRNA expression estimation methods

Description Based on a large miRNA dilution study, this package provides tools to read in the raw amplification data and use these data to assess the performance of methods that estimate expression from the amplification curves.

Author Matthew N. McCall <mccallm@gmail.com>, Lauren Kemperman <lkemperm@u.rochester.edu>

Maintainer Matthew N. McCall <mccallm@gmail.com>

Depends R (>= 3.2), Biobase (>= 2.22.0), miRcompData

Imports utils, methods, graphics, KernSmooth, stats

VignetteBuilder knitr

Suggests BiocStyle, knitr, rmarkdown, RUnit, BiocGenerics, shiny

License GPL-3 | file LICENSE

biocViews Software, qPCR, Preprocessing, QualityControl

NeedsCompilation no

git_url <https://git.bioconductor.org/packages/miRcomp>

git_branch RELEASE_3_20

git_last_commit 8942391

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-03-06

Contents

accuracy	2
completeFeatures	3
expressionComp	4
lifetech	5
limitOfDetection	5
miRcompShinyApp	6
precision	7

qpcRb4	8
qpcRb5	8
qpcRdefault	9
qpcR15	9
qpcRlinexp	10
qualityAssessment	10
titrationResponse	11
Index	13

accuracy

Assess the accuracy of the expression estimates

Description

This function assesses the signal detect slope (a measure of accuracy) for each feature.

Usage

```
accuracy(object1, qcThreshold1, object2=NULL, qcThreshold2=NULL,
commonFeatures=TRUE, bins=3, label1=NULL, label2=NULL)
```

Arguments

object1	a list containing two elements: ct (the expression estimates) and qc (quality scores)
qcThreshold1	a numeric threshold corresponding to object1\$qc below which values are considered low quality.
object2	an optional second list of the same format as object1, used to compare two methods.
qcThreshold2	a numeric threshold corresponding to object2\$qc below which values are considered low quality.
commonFeatures	if TRUE and object2 is non-NULL, only high quality non-NA features in common between both objects are used.
bins	the number of bins to divide the data into.
label1	optional label corresponding to object 1 to be used in plotting.
label2	optional label corresponding to object 2 to be used in plotting.

Value

A plot of signal detect slopes stratified by difference in pure sample expression is produced, and a summary (median and MAD) of the signal detect slopes in each bin is returned.

Author(s)

Matthew N. McCall

Examples

```
data(lifetech)
accuracy(object1=lifetech, qcThreshold1=1.25)
data(qpcRdefault)
accuracy(object1=lifetech, qcThreshold1=1.25,
         object2=qpcRdefault, qcThreshold2=0.99)
```

completeFeatures

Determine the Number of Completely Observed Features

Description

This function determines the number of features that are good quality and non-NA across all samples using a given quality threshold.

Usage

```
completeFeatures(object1, qcThreshold1, object2=NULL, qcThreshold2=NULL,
                 label1=NULL, label2=NULL)
```

Arguments

object1	a list containing two elements: ct (the expression estimates) and qc (quality scores)
qcThreshold1	a numeric threshold corresponding to object1\$qc below which values are considered low quality.
object2	an optional second list of the same format as object1, used to compare two methods.
qcThreshold2	a numeric threshold corresponding to object2\$qc below which values are considered low quality.
label1	optional label corresponding to object 1 to be used in plotting.
label2	optional label corresponding to object 2 to be used in plotting.

Value

The function generates a table of the number of complete, partial, and absent features.

Author(s)

Matthew N. McCall

Examples

```
data(lifetech)
completeFeatures(object1=lifetech, qcThreshold1=1.25)
data(qpcRdefault)
completeFeatures(object1=lifetech, qcThreshold1=1.25,
                 object2=qpcRdefault, qcThreshold2=0.99)
```

 expressionComp

Compare expression estimates between two methods

Description

This function allows one to easily compare the expression estimates generated by two different algorithms. Feature / sample combinations that differ above a given threshold can be easily highlighted.

Usage

```
expressionComp(object1, qcThreshold1,
               object2, qcThreshold2,
               label1=NULL, label2=NULL,
               diffThreshold=2,
               plotType=c("scatterplot", "MAplot"))
```

Arguments

object1	a list containing two elements: ct (the expression estimates) and qc (quality scores)
qcThreshold1	a numeric threshold corresponding to object1\$qc below which values are considered low quality.
object2	a list containing two elements: ct (the expression estimates) and qc (quality scores)
qcThreshold2	a numeric threshold corresponding to object2\$qc below which values are considered low quality.
label1	an optional label for plotting on the x-axis
label2	an optional label for plotting on the y-axis
diffThreshold	feature / sample combinations for which the expression estimates differ by more than this value are returned by the function and highlighted in the plot.
plotType	scatterplot or MA-plot

Value

This function plots the expression estimates produced by two different methods. Feature / sample combinations for which the expression estimates differ by more than diffThreshold are returned by the function and highlighted in the plot.

Author(s)

Matthew N. McCall

Examples

```
data(lifetech)
data(qpcRdefault)
tmp <- expressionComp(object1=lifetech, qcThreshold1=1.25,
                      object2=qpcRdefault, qcThreshold2=0.99, plotType="scatter")
```

`lifetech`*The processed data generated using the LifeTech software.*

Description

These data are also used to demonstrate the functionality of the miRcomp package.

Usage

```
data(lifetech)
```

Format

A list with two elements.

```
ct  a miRNA x sample matrix of expression estimates
qc  a miRNA x sample matrix of quality scores (here AmpScores)
```

Examples

```
data(lifetech)
```

`limitOfDetection`*Assess the limit of detection of a given method*

Description

This function assesses the limit of detection in one of two ways: (1) the distribution of expression estimates stratified by the proportion of poor quality values within replicates, (2) the average vs expected expression for the two most diluted sample types.

Usage

```
limitOfDetection(object, qcThreshold, plotType=c("boxplot", "scatterplot", "MAplot"))
```

Arguments

<code>object</code>	a list containing two elements: <code>ct</code> (the expression estimates) and <code>qc</code> (quality scores)
<code>qcThreshold</code>	a numeric threshold corresponding to <code>object\$qc</code> below which values are considered low quality.
<code>plotType</code>	the desired output type – boxplot is option (1); scatterplot is option (2), an MA-plot is option (3).

Value

This function assesses the limit of detection in several ways. If `plotType` is `boxplot`, then boxplots of expression estimates stratified by the proportion of poor quality values within replicates is displayed. The function also outputs a list with the values plotted in each box of the boxplot. If `plotType` is `scatterplot`, then the average within replicate expression vs expected expression (based on pure sample expression) is displayed for the 0.1/0.1 dilution and 0.01/0.01 dilution. If `plotType` is `MAplot`, then the difference in expression (average within replicate expression - expected expression) is displayed for the 0.1/0.1 dilution and 0.01/0.01 dilution. For both `plotTypes`, `scatterplot` and `MAplot`, the function outputs a matrix containing estimates of the limit of detection for four different tolerances. Specifically, the two columns correspond to the two dilutions (0.1/0.1 and 0.01/0.01) and rows correspond to the median difference between the observed and expected values. The values in the matrix are the expected expression values such that the median absolute difference of all larger expected expression values is approximately equal to the given tolerance.

Author(s)

Matthew N. McCall

Examples

```
data(lifetech)
tmp <- limitOfDetection(object=lifetech, qcThreshold=1.25)
data(qpcRdefault)
limitOfDetection(object=qpcRdefault, qcThreshold=0.99, plotType="scatter")
```

miRcompShinyApp

Launch a Shiny app to use the miRcomp package interactively

Description

This function launches the miRcomp Shiny app locally.

Usage

```
miRcompShinyApp()
```

Author(s)

Matthew N. McCall and Lauren Kemperman

Examples

```
## miRcompShinyApp()
```


qpcRb4

The processed data generated using the 4 parameter sigmoidal method from the qpcR software.

Description

These data are also used to demonstrate the functionality of the miRcomp package.

Usage

```
data(qpcRb4)
```

Format

A list with two elements.

```
ct  a miRNA x sample matrix of expression estimates
qc  a miRNA x sample matrix of quality scores (here R-squared values)
```

Examples

```
data(qpcRb4)
```

qpcRb5

The processed data generated using the 5 parameter sigmoidal method from the qpcR software.

Description

These data are also used to demonstrate the functionality of the miRcomp package.

Usage

```
data(qpcRb5)
```

Format

A list with two elements.

```
ct  a miRNA x sample matrix of expression estimates
qc  a miRNA x sample matrix of quality scores (here R-squared values)
```

Examples

```
data(qpcRb5)
```

qpcRdefault	<i>The processed data generated using the default method (4 parameter log-logistic) implemented in the qpcR software package.</i>
-------------	---

Description

These data are also used to demonstrate the functionality of the miRcomp package.

Usage

```
data(qpcRdefault)
```

Format

A list with two elements.

```
ct  a miRNA x sample matrix of expression estimates
qc  a miRNA x sample matrix of quality scores (here R-squared values)
```

Examples

```
data(qpcRdefault)
```

qpcR15	<i>The processed data generated using the 5 parameter log-logistic method from the qpcR software.</i>
--------	---

Description

These data are also used to demonstrate the functionality of the miRcomp package.

Usage

```
data(qpcR15)
```

Format

A list with two elements.

```
ct  a miRNA x sample matrix of expression estimates
qc  a miRNA x sample matrix of quality scores (here R-squared values)
```

Examples

```
data(qpcR15)
```

qpcRlinexp	<i>The processed data generated using the linear-exponential method implemented in the qpcR software package.</i>
------------	---

Description

These data are also used to demonstrate the functionality of the miRcomp package.

Usage

```
data(qpcRlinexp)
```

Format

A list with two elements.

```
ct  a miRNA x sample matrix of expression estimates
qc  a miRNA x sample matrix of quality scores (here R-squared values)
```

Examples

```
data(qpcRlinexp)
```

qualityAssessment	<i>Quality Assessment for MicroRNA Expression Estimates</i>
-------------------	---

Description

This function produces several figures to examine quality scores.

Usage

```
qualityAssessment(object1, object2=NULL, cloglog1=FALSE, cloglog2=FALSE,
                  na.rm=FALSE, plotType=c("scatterplot", "boxplot"),
                  label1=NULL, label2=NULL)
```

Arguments

object1	a list containing two elements: ct (the expression estimates) and qc (quality scores)
object2	an optional second list of the same format as object1, used to compare two methods.
cloglog1	if TRUE, the $-\log(-\log(\text{object1}\$qc))$ is plotted. This is useful to visualize certain quality scores such as R-squared.
cloglog2	if TRUE, the $-\log(-\log(\text{object2}\$qc))$ is plotted. This is useful to visualize certain quality scores such as R-squared.
na.rm	if TRUE, quality scores corresponding to NA expression estimates are removed from analysis. This is only used if plotType=="boxplot".
plotType	type of plot to return. Options are a scatterplot or a boxplot.
label1	optional label corresponding to object 1 to be used in plotting.
label2	optional label corresponding to object 2 to be used in plotting.

Value

The function generates a plot of the desired type.

Author(s)

Matthew N. McCall

Examples

```
data(lifetech)
qualityAssessment(lifetech, plotType="boxplot")

data(qpcRdefault)
qualityAssessment(object1=lifetech,object2=qpcRdefault,cloglog2=TRUE)
```

titrationResponse	<i>Assess monotone signal across titration</i>
-------------------	--

Description

This function determines

Usage

```
titrationResponse(object1, qcThreshold1, object2=NULL, qcThreshold2=NULL,
                  commonFeatures=TRUE, label1=NULL, label2=NULL)
```

Arguments

object1	a list containing two elements: ct (the expression estimates) and qc (quality scores)
qcThreshold1	a numeric threshold corresponding to object1\$qc below which values are considered low quality.
object2	an optional second list of the same format as object1, used to compare two methods.
qcThreshold2	a numeric threshold corresponding to object2\$qc below which values are considered low quality.
commonFeatures	if TRUE and object2 is non-NULL, only high quality non-NA features in common between both objects are used.
label1	optional label corresponding to object 1 to be used in plotting.
label2	optional label corresponding to object 2 to be used in plotting.

Value

A table listing the number of features showing a monotone titration response and a figure showing the proportion of features showing a monotone titration response vs the difference in pure sample expression.

Author(s)

Matthew N. McCall

Examples

```
data(lifetech)
titrationResponse(object1=lifetech, qcThreshold1=1.25)
data(qpcRdefault)
titrationResponse(object1=lifetech, qcThreshold1=1.25,
  object2=qpcRdefault, qcThreshold2=0.99)
```

Index

* datasets

- lifetech, [5](#)
- qpcRb4, [8](#)
- qpcRb5, [8](#)
- qpcRdefault, [9](#)
- qpcRl5, [9](#)
- qpcRlinexp, [10](#)

* manip

- accuracy, [2](#)
- completeFeatures, [3](#)
- expressionComp, [4](#)
- limitOfDetection, [5](#)
- miRcompShinyApp, [6](#)
- precision, [7](#)
- qualityAssessment, [10](#)
- titrationResponse, [11](#)

[accuracy, 2](#)

[completeFeatures, 3](#)

[expressionComp, 4](#)

[lifetech, 5](#)

[limitOfDetection, 5](#)

[miRcompShinyApp, 6](#)

[precision, 7](#)

[qpcRb4, 8](#)

[qpcRb5, 8](#)

[qpcRdefault, 9](#)

[qpcRl5, 9](#)

[qpcRlinexp, 10](#)

[qualityAssessment, 10](#)

[titrationResponse, 11](#)