frma

April 20, 2011

barcode *Gene Expression Barcode*

Description

This function converts expression values produced via fRMA to a gene expression barcode.

Usage

```
barcode(object, platform=NULL, mu=NULL, tau=NULL, cutoff=6.5, output="binary")
```

Arguments

`object` a vector or matrix of expression values or an ExpressionSet or PLMset produced by frma

`platform` the platform of the input data. One of GPL96, GPL570, GPL1261. Required if object is a vector or matrix.

`mu` the mean of the unexpressed distribution. If NULL then precomputed values are used.

`tau` the standard deviation of the unexpressed distribution. If NULL then precomputed values are used.

`cutoff` the lod score cutoff used if output is binary.

`output` the desired values to be returned. Options are: p-value, z-score, lod, or binary.

Value

A matrix containing the type of output specified by the output parameter. The option *binary* creates a gene expression barcode where 1s denote expressed genes and 0s denote unexpressed genes. The option *p-value* returns the p-values for the expression values under the unexpressed distribution. The option *lod* returns the LOD scores for expression values under the unexpressed distribution. The option *z-score* returns the z-scores for the expression values under the unexpressed distribution.

Author(s)

Matthew N. McCall

Examples

```
library(frma)
library(frmaExampleData)
data(AffyBatchExample)
object <- frma(AffyBatchExample)
bc <- barcode(object)
```

`frma` *Frozen Robust Multi-Array Analysis*

Description

This function preprocesses an AffyBatch object using the fRMA method.

Usage

```
frma(object, background="rma", normalize="quantile",
    summarize="robust_weighted_average", input.vecs=list(normVec=NULL,
    probeVec=NULL, probeVarBetween=NULL, probeVarWithin=NULL,
    probesetSD=NULL), output.param=NULL, verbose=FALSE)
```

Arguments

`object` an AffyBatch

`background` type of background correction to perform: either "none" or "rma".

`normalize` type of normalization to perform: either "none" or "quantile".

`summarize` type of summarization to perform: one of "median_polish", "average", "median", "weighted_average", "robust_weighted_average", "batch".

`input.vecs` a list of vectors to be used in preprocessing. If necessary elements are NULL, the correct package with the default vectors is loaded if it has been installed. These packages are of the form: <platform>frmavecs.

`output.param` a vector of output elements to return. By default only the expression values are returned. Can contain any of "residuals", "weights", or "stderr"

`verbose` logical value. If TRUE then some messages are displayed while the function runs.

Value

The function returns a PLMset.

Author(s)

Matthew N. McCall

Examples

```
library(frmaExampleData)
data(AffyBatchExample)
object <- frma(AffyBatchExample)
```

`GNUSE` *GNUSE*

Description

Computes the generalized normalized unscaled standard error (a measure of microarray quality).

Usage

```
GNUSE(object, type=c("plot", "values", "stats", "density"), ...)
```

Arguments

`object`	a PLMset produced by frma
`type`	the desired output
`...`	additional graphical parameters for types plot or density

Value

If type is plot, boxplots of GNUSE values are displayed. If type is values, the GNUSE values are returned. If type is stats, the median, IQR, 95th, and 99th percentiles are reported. If type is density, a density plots of GNUSE values are displayed.

Author(s)

Matthew N. McCall

Examples

```
library(frma)
library(frmaExampleData)
data(AffyBatchExample)
object <- frma(AffyBatchExample, output.param=c("stderr"))
GNUSE(object, type="stats")
```

Index