

# frma

October 5, 2010

---

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 frmaExpression-Set 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, binary, or weight.

## 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. The option *weight* returns weights which roughly correspond to the probability of expression for each gene.

## 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 either an ExpressionSet or a frmaExpressionSet object depending on the elements of output.param. The assayData field of these objects contains the expression values (exprs) and optionally the standard errors (se.exprs). A frmaExpressionSet contains two further optional fields: the weights used in the summarization step (weights) and the residuals from fitting the probe-level model (residuals).

**Author(s)**

Matthew N. McCall

**Examples**

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

---

```
frmaExpressionSet-class
```

*Class to Contain and Describe High-Throughput Expression Level Assays preprocessed with fRMA*

---

**Description**

This is a class representation for fRMA-preprocessed expression data. `frmaExpressionSet` class is derived from `ExpressionSet`, and requires a matrix named `exprs` and optionally matrices named `se.exprs`, `weights`, and `residuals`.

**Extends**

Extends class `ExpressionSet`.

**Creating Objects**

```
new("frmaExpressionSet", phenoData = new("AnnotatedDataFrame"), featureData = new("AnnotatedDataFrame"), experimentData = new("MIAME"), annotation = character(0), exprs = new("matrix"), weights = new("matrix"), residuals = new("matrix"))
```

This creates a `frmaExpressionSet` with `assayData` implicitly created to contain `exprs`. The only required named argument is `exprs`. Two optional named matrices, `weights` and `residuals`, can be added to the object.

```
new("frmaExpressionSet", assayData = assayDataNew(exprs=new("matrix"), se.exprs=new("matrix"), phenoData = new("AnnotatedDataFrame"), featureData = new("AnnotatedDataFrame"), experimentData = new("MIAME"), annotation = character(0), weights = new("matrix"), residuals = new("matrix"))
```

This creates a `frmaExpressionSet` with `assayData` provided explicitly. In this form, the only required named argument is `assayData`. Two optional named matrices, `weights` and `residuals`, can be added to the object.

**Slots**

**weights:** weights used in the summarization step

**residuals:** residuals from fitting the probe-level model

**Inherited from ExpressionSet:**

**assayData:** Contains matrices with equal dimensions, and with column number equal to `nrow(phenoData)`. `assayData` must contain a matrix `exprs` with rows representing features and columns representing samples. It may also contain a matrix `se.exprs` containing standard errors.

**phenoData:** See `eSet`

**annotation:** See `eSet`

**featureData:** See `eSet`

**experimentData:** See `eSet`

**Methods**

Class-specific methods:

`se.exprs(frmaExpressionSet)` Access elements named `se.exprs` in the `AssayData-class` slot.

`weights(frmaExpressionSet)` Access elements named `weights`

`residuals(frmaExpressionSet)` Access elements named `residuals`

For derived methods (see `ExpressionSet`).

**See Also**

`eSet-class`, `ExpressionSet-class`, `frma`.

**Examples**

```
# create an instance of frmaExpressionSet
new("frmaExpressionSet")
```

# Index

**\*Topic classes**  
frmaExpressionSet-class, 3

**\*Topic manip**  
barcode, 1  
frma, 2

as.ExpressionSet  
(frmaExpressionSet-class),  
3

as.ExpressionSet, frmaExpressionSet-method  
(frmaExpressionSet-class),  
3

barcode, 1

class:frmaExpressionSet  
(frmaExpressionSet-class),  
3

frma, 2

frmaExpressionSet  
(frmaExpressionSet-class),  
3

frmaExpressionSet-class, 3

frmaExpressionSet-methods  
(frmaExpressionSet-class),  
3

initialize, frmaExpressionSet-method  
(frmaExpressionSet-class),  
3

residuals  
(frmaExpressionSet-class),  
3

residuals, frmaExpressionSet-method  
(frmaExpressionSet-class),  
3

se.exprs  
(frmaExpressionSet-class),  
3

se.exprs, ExpressionSet-method  
(frmaExpressionSet-class),  
3

se.exprs, frmaExpressionSet-method  
(frmaExpressionSet-class),  
3

weights  
(frmaExpressionSet-class),  
3

weights, frmaExpressionSet-method  
(frmaExpressionSet-class),  
3