Package 'ArrayBin'

February 19, 2015

Version 0.2

Date 2013-02-01

Title Binarization of numeric data arrays

Author Ed Curry

Maintainer Ed Curry <e.curry@imperial.ac.uk>

Depends R (>= 2.15.0), SAGx

Description Fast adaptive binarization for numeric data arrays, particularly designed for high-throughput biological datasets. Includes options to filter out rows of the array with insufficient magnitude or variation (based on gap statistic).

License GPL (>= 2)

URL http://www.r-project.org,

http://www1.imperial.ac.uk/medicine/people/e.curry/

NeedsCompilation no

Repository CRAN

Date/Publication 2013-06-21 11:34:43

R topics documented:

binarize.array *Fast Adaptive Binarization*

Description

Performs fast adaptive binarization of numeric arrays, providing options for filtering rows with insufficient variation

Usage

```
binarize.array(x,min.filter=NA,var.filter=0,fc.filter=0,
na.filter = FALSE,log.base=NA,use.gap=FALSE)
```

Arguments

x	Numeric data input array used to generate binary output array. Each row of the array represents a different variable.
min.filter	Minimum-value filter: rows of x with no value greater than min.filter will have all values set to 0.
var.filter	Variation filter: the proportion of lowest-variance rows of x to have all values set to 0.
fc.filter	Fold-change filter: rows of x with maximum fold-change less than fc.filter will have all values set to 0.
na.filter	NA filter: all rows of x with _any_ NAs will have all values set to 0. NB: even with na.filter=FALSE any NA values will be passed through with output value NA.
log.base	Base of logarithm to use for calculating fold-changes in rows of x. Unless log.base=NA input data x is assumed to be log-transformed.
use.gap	Boolean indicating whether to use gap statistic to identify rows insufficiently converted to binary representation. If TRUE, execution will be _much_ slower.

Details

Implementation of an adaptive method for binarizing gene expression data on a per-probe basis and demonstrate the superior effectiveness of our method when compared with other, commonly used approaches. This adaptive binarization method can be applied to DNA methylation microarray data, which has implications for cross-platform integration, and can reduce batch effects in the data.

Value

Binarized representation of x. That is, a numeric array of same dimensions as input x, containing values 0 (representing a 'low' value of corresponding variable) and 1 (respresenting a 'high' value of the corresponding variable).

Author(s)

Ed Curry <e.curry@imperial.ac.uk>

Examples

```
## create a numeric array
x.cont <- array(runif(60),dim=c(10,6))
## Not run: x.cont

## find binary representation of array
x.bin <- binarize.array(x.cont)
## Not run: x.bin

## use gap statistic to filter insufficiently variable rows
x.gap <- binarize.array(x.cont,use.gap=TRUE)
## Not run: x.gap
```

---

clusterDisc *Fast Adaptive Binarization - internal*

Description

Performs fast adaptive binarization of numeric arrays

Usage

```
clusterDisc(x,use.gap)
```

Arguments

x	Numeric data input vector used to generate binary output
use.gap	Boolean indicating whether to use gap statistic to infer whether or not the data can be sufficiently converted to a binary representation.

Details

Function called by `binarize.array`

Value

Binarized representation of `x`. That is, a numeric vector of the same length as input `x`, containing values `0` (representing a 'low' value) and `1` (respresenting a 'high' value).

Author(s)

Ed Curry <e.curry@imperial.ac.uk>

mskmeans *Maximally-Separated K-Means*

Description

Performs k-means clustering with initialization of centroids to partition data points around the data points with greatest magnitude difference

Usage

```
mskmeans(data,k=2)
```

Arguments

data	Numeric data input vector used to generate binary output
k	Number of clusters

Details

Function called by `binarize.array`. Calculates k-means (default k=2 gives binarization) classification around maximally-separated data points

Value

Discretized representation of `data`. For k=2, that is a numeric vector of the same length as input `data`, containing values `0` (representing a 'low' value) and `1` (respresenting a 'high' value).

Author(s)

Ed Curry <e.curry@imperial.ac.uk>

Index