OneStepBiweightAlgorithm

One-step Tukey's biweight

Description

Computes one-step Tukey's biweight on a vector. Note that this implementation follows the Affymetrix code, which is different from the Tukey's biweight computed by the affy package.

Usage

OneStepBiweightAlgorithm(x, c, epsilon)

Arguments

x vector of data
c tuning constant (see details)
epsilon fuzz value to avoid division by zero (see details)

Details

The details can be found in the given reference.

Value

A numeric value

Note

Based on Affymetrix MAS5 Statistical SDK source code http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip

Author(s)

Richard Kennedy (rkennedy@vcu.edu)

References

**SScore**

**Compute S-Score values**

**Description**

Computes the S-Score values for a pair of Affymetrix GeneChips.

**Usage**

```r
SScore(afbatch = stop("No CEL files specified"), classlabel = c(0,1), SF = NULL, 
      SDT = NULL, rm.outliers = TRUE, rm.mask = TRUE, rm.extra = NULL, 
      digits = NULL, verbose = FALSE, celfile.path = NULL, celfile.names = NULL)
```

**Arguments**

- `afbatch` An `AffyBatch` object
- `classlabel` A vector identifying the class for each column of the `AffyBatch` object
- `SF` a list of Scale Factor (SF) values for each GeneChip
- `SDT` a list of Standard Difference Threshold (SDT) values for each GeneChip
- `rm.outliers` should the spots marked as 'OUTLIERS' be excluded from S-Score calculation?
- `rm.mask` should the spots marked as 'MASKS' be excluded from S-Score calculation?
- `rm.extra` if TRUE, overrides what is in `rm.mask` and `rm.outliers`
- `digits` number of significant digits for S-Score values
- `verbose` logical value. If TRUE it provides more detail of the S-Score calculations.
- `celfile.path` character denoting the path for the *.CEL files corresponding to afbatch
- `celfile.names` optional character vector containing the names of the *.CEL files

**Details**

Computes S-Score values as described by Zhang et al. (2002). SScore provides a simpler interface for comparing only two classes of GeneChips, while SScoreBatch compares multiple pairs of chips.

The classlabel consists of a vector with one entry for each column of the `AffyBatch` object. Each entry consists of a 0 or a 1 to identify the class to which the chip for the corresponding column belongs. SScore will conduct a two-class test comparing all chips labeled 0 to all chips labeled 1. If classlabel is not specified, it defaults to a two-chip comparison, compatible with previous versions of SScore.

The SF and SDT factors are required for all calculations. If NULL, these values will be calculated according to the Affymetrix Statistical Algorithms Description Document. digits allows the specification of the number of significant digits for the S-Score values; if NULL, the maximum number of significant digits are retained.

**Value**

An `ExpressionSet` with S-Score values in the `exprs` slot.
**Note**

Based on C++ code by Li Zhang and Delphi code by Robnet Kerns

**Author(s)**

Richard Kennedy (rkennedy@vcu.edu)

**References**


**See Also**

SScoreBatch, computeSFandSDT, computeOutlier

**Examples**

```r
if (length(dir(pattern="*.cel$")) != 0) {

## Read in the *.CEL files
abatch <- ReadAffy()

## default calling method
SScores <- SScore(abatch)

## specifying SF and SDT (gives same results as above)
SfSdt <- computeSFandSDT(abatch)
SScores <- SScore(abatch,SF=SfSdt$SF,SDT=SfSdt$SDT)

## specifying outlier and masked values should be included in calculations
SScores <- SScore(abatch,rm.outliers=FALSE,rm.mask=FALSE)

## round results to 3 significant digits
SScores <- SScore(abatch,digits=3)

## show verbose output
SScores <- SScore(abatch,verbose=TRUE)
}
```

---

**SScoreBatch**

*Compute S-Score values*

**Description**

Computes the S-Score values for multiple pairs of Affymetrix GeneChips
SScoreBatch

Usage

SScoreBatch(afbatch = stop("No CEL files specified"), compare = stop("No list of comparisons given"),
SF = NULL, SDT = NULL, rm.outliers = TRUE, rm.mask = TRUE, rm.extra = NULL,
digits = NULL, verbose = FALSE, celfile.path = NULL, celfile.names = NULL)

Arguments

afbatch An AffyBatch object
compare A matrix describing which chips to compare
SF a list of Scale Factor (SF) values for each GeneChip
SDT a list of Standard Difference Threshold (SDT) values for each GeneChip
rm.outliers should the spots marked as 'OUTLIERS' be excluded from S-Score calculation?
rm.mask should the spots marked as 'MASKS' be excluded from S-Score calculation?
rm.extra if TRUE, overrides what is in rm.mask and rm.outliers
digits number of significant digits for S-Score values
verbose logical value. If TRUE it provides more detail of the S-Score calculations.
celfile.path character denoting the path for the *.CEL files corresponding to afbatch
celfile.names optional character vector containing the names of the *.CEL files

Details

Computes S-Score values as described by Zhang et al. (2002). SSscoreBatch allows comparison of multiple pairs of chips, while SScore provides a simpler interface when comparing only two GeneChips.

compare specifies how the pairwise comparisons are performed. It is an N x 2 matrix, where N is the number of pairwise comparisons; each row of the matrix contains index in the AffyBatch object for the chips to be compared. For example,

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

would do a comparison of chip 1 to chip 3, a comparison of chip 4 to chip 2, a comparison of chip 5 to chip 9, and so on. The columns in ExpressionSet correspond to the rows of compare, so that the results of the first comparison are in column 1, the results of the second comparison are in column 2, and so on.

The SF and SDT factors are required for all calculations. If NULL, these values will be calculated according to the Affymetrix Statistical Algorithms Description Document. probes. digits allows the specification of the number of significant digits for the S-Score values; if NULL, the maximum number of significant digits are retained.

Value

An ExpressionSet with S-Score values in the exprs slot.
computeAffxRawQ

Note

Based on C++ code by Li Zhang and Delphi code by Robnet Kerns

Author(s)

Richard Kennedy ⟨rkennedy@vcu.edu⟩

References


See Also

SScore, computeSFandSDT, computeOutlier

Examples

```r
if (length(dir(pattern=".cel$")) != 0) {

## Read in the *.CEL files
abatch <- ReadAffy()

## default calling method
SScores <- SScoreBatch(abatch)

## specifying SF and SDT (gives same results as above)
SfSdt <- computeSFandSDT(abatch)

## specifying outlier and masked values should be included in calculations
SScores <- SScoreBatch(abatch, SF=SfSdt$SF, SDT=SfSdt$SDT)
SScores <- SScoreBatch(abatch, rm.outliers=FALSE, rm.mask=FALSE)

## round results to 3 significant digits
SScores <- SScoreBatch(abatch, digits=3)

## show verbose output
SScores <- SScoreBatch(abatch, verbose=TRUE)
}
```

---

computeAffxRawQ  Computes RawQ using affxparser routines

Description

Computes the RawQ value of a single Affymetrix GeneChip
computeAffxSFandSDT

Compute SF and SDT values using affxparser routines

Description

Computes the scaling factor (SF) and statistical difference threshold (SDT) values of Affymetrix GeneChips, for use in calculating S-Score values.
Usage

computeAffxSFandSDT(afbatch, stdvs, pixels, TGT = 500, digits = NULL, verbose = FALSE, plot.histogram = FALSE)

Arguments

afbatch An AffyBatch object
stdvs matrix of standard deviations for probe intensities from *.CEL file
pixels matrix of number of pixels for probe intensities from *.CEL file
TGT the target intensity to which the arrays should be scaled
digits number of significant digits for SF and SDT values
verbose logical value. If TRUE it provides more detail of the SF and SDT calculations.
plot.histogram logical value. if TRUE it plots a histogram of intensities

Details

Calculates SF and SDT factors using the algorithms described in the Affymetrix Statistical Algorithms Description Document. The SF and SDT may be used in the calculation of S-Score values, or may be useful in their own right. One SF and SDT value is calculated for each GeneChip, which are arranged in the same order as the columns in the AffyBatch object.

Value

computeSFandSDT returns a list containing the following components:

SF SF values, one for each GeneChip
SDT SDT values, one for each GeneChip

Note

Based on Affymetrix MAS5 Statistical SDK source code http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip, as well as C++ code by Li Zhang and Delphi code by Robnet Kerns

Author(s)

Richard Kennedy (rkennedy@vcu.edu)

References


Examples

if (length(dir(pattern=".cel$")) != 0) {
  ## Read in the *.CEL files
  abatch <- ReadAffy()
  ## compute SF and SDT

## Description

Computes outlier and/or mask probes for a set of Affymetrix GeneChips that will be excluded from S-Score calculations.

## Usage

```r
computeOutlier(afbatch, rm.mask = TRUE, rm.outliers = TRUE, rm.extra = TRUE, celfile.path = NULL, celfile.names = NULL)
```

## Arguments

- `afbatch`: An `AffyBatch` object.
- `rm.mask`: should the spots marked as 'MASKS' be excluded from S-Score calculation? (default: `TRUE`)
- `rm.outliers`: should the spots marked as 'OUTLIERS' be excluded from S-Score calculation? (default: `TRUE`)
- `rm.extra`: if `TRUE`, overrides what is in `rm.mask` and `rm.outliers` (default: `FALSE`)
- `celfile.path`: character denoting the path for the *.CEL files corresponding to `afbatch` (default: `NULL`)
- `celfile.names`: optional character vector containing the names of the *.CEL files (default: `NULL`)

## Details

Computes the outlier and / or mask probes for an `AffyBatch` object. These are returned in matrix form, with one probe per row and one chip per column. The value of each location in the matrix will be `TRUE` if the corresponding probe is an outlier / masked value and `FALSE` if it is not. The options may be set to exclude only outlier values, only mask values, or both. The probes are be arranged in the same row order as the intensity values. Note that this function assumes the *.CEL files are still available in the directory given by `celfile.path` (or the current directory if `celfile.path` is not specified). The *.CEL files are given by `celfile.names`. If `celfile.names` is not specified, the sample names from the `AffyBatch` object will be used.

## Value

- A matrix containing the list of outliers / masked values for the given `AffyBatch` object.

## Note

Based on C++ code by Li Zhang and Delphi code by Robnet Kerns.
computeRawQ

**Author(s)**
Richard Kennedy ⟨rkennedy@vcu.edu⟩

**Examples**

```r
if (length(dir(pattern=".cel$")) != 0) {
  abatch <- ReadAffy()
  outlier <- computeOutlier(abatch)
}
```

---

**computeRawQ**  **Computes RawQ**

**Description**

Computes the RawQ value of a single Affymetrix GeneChip

**Usage**

```r
computeRawQ(fname, intensity, probe.index, probe.zoneID, bgCells, NumberZones, celfile.path = NULL)
```

**Arguments**

- `fname`  character string with the filename of the GeneChip
- `intensity`  vector of intensities for the GeneChip
- `probe.index`  vector of indices for each probe
- `probe.zoneID`  vector of zone ID numbers for each probe
- `bgCells`  number of background cells for the GeneChip
- `NumberZones`  number of zones on the GeneChip
- `celfile.path`  character denoting the path for the *.CEL files specified in `fname`

**Details**

Calculates RawQ using the algorithms described in the Affymetrix Statistical Algorithms Description Document for a single GeneChip. This is an internal function that will generally not be accessed directly.

**Value**

- the RawQ value for the given array

**Note**

Based on Affymetrix MAS5 Statistical SDK source code [http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip](http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip), as well as C++ code by Li Zhang and Delphi code by Robnet Kerns

**Author(s)**
Richard Kennedy ⟨rkennedy@vcu.edu⟩
References


computeSFandSDT  Compute SF and SDT values

Description

Computes the scaling factor (SF) and statistical difference threshold (SDT) values of Affymetrix GeneChips, for use in calculating S-Score values

Usage

computeSFandSDT(afbatch, TGT = 500, digits = NULL, verbose = FALSE, plot.histogram = FALSE, celfile.path = NULL)

Arguments

afbatch  An AffyBatch object
TGT  the target intensity to which the arrays should be scaled
digits  number of significant digits for SF and SDT values
verbose  logical value. If TRUE it provides more detail of the SF and SDT calculations.
plot.histogram  logical value. if TRUE it plots a histogram of intensities
celfile.path  character denoting the path for the *.CEL files corresponding to afbatch

Details

Calculates SF and SDT factors using the algorithms described in the Affymetrix Statistical Algorithms Description Document. The SF and SDT may be used in the calculation of S-Score values, or may be useful in their own right. One SF and SDT value is calculated for each GeneChip, which are arranged in the same order as the columns in the AffyBatch object.

Value

computeSFandSDT returns a list containing the following components:

SF  SF values, one for each GeneChip
SDT  SDT values, one for each GeneChip

Note

Based on Affymetrix MAS5 Statistical SDK source code http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip, as well as C++ code by Li Zhang and Delphi code by Robnet Kerns

Author(s)

Richard Kennedy (rkennedy@vcu.edu)
References


Examples

```r
if (length(dir(pattern=".cel$")) != 0) {
  ## Read in the *.CEL files
  abatch <- ReadAffy()

  ## compute SF and SDT
  SfSdt <- computeSFandSDT(abatch)

  ## show verbose output
  SfSdt <- computeSFandSDT(abatch, verbose=TRUE)

  ## plot PM and MM histograms for each *.CEL file
  SfSdt <- computeSFandSDT(abatch, plot.histogram=TRUE)
}
```

computeZoneIInfo  Compute zone background and noise

Description

Computes the background and noise for a given zone of a single Affymetrix GeneChip

Usage

```r
computeZoneIInfo(ZoneInfo, NumberBGCells)
```

Arguments

- **ZoneInfo**: vector of intensities in a given zone
- **NumberBGCells**: number of background cells for the GeneChip

Details

Calculates background and noise for a zone using the algorithms described in the Affymetrix Statistical Algorithms Description Document. This is an internal function that will generally not be accessed directly.

Value

`computeZoneIInfo` returns a list containing the following components:

- **background**: background value for the given zone
- **noise**: noise value for the given zone
computeZonenoise

Note

Based on Affymetrix MAS5 Statistical SDK source code http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip, as well as C++ code by Li Zhang and Delphi code by Robnet Kerns.

Author(s)

Richard Kennedy (rkennedy@vcu.edu)

References


computeZonenoise  Computes zone noise

Description

Computes the noise (average standard error) of the probe intensities for a single Affymetrix GeneChip.

Usage

computeZonenoise(index, intensity, stdv, npixels, bgCells)

Arguments

index  vector of indices for probes in the given zone
intensity  vector of intensities for the GeneChip
stdv  vector of standard deviations for the GeneChip
npixels  vector containing number of pixels for each probe of the GeneChip
bgCells  number of background cells on the GeneChip

Details

Calculates the noise (average standard error) of the probes in a given zone, using the algorithms described in the Affymetrix Statistical Algorithms Description Document, for a single GeneChip. This is an internal function that will generally not be accessed directly.

Value

the noise of the probes for the given array

Note

Based on Affymetrix MAS5 Statistical SDK source code http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip, as well as C++ code by Li Zhang and Delphi code by Robnet Kerns.
trimMean

Author(s)

Richard Kennedy (rkennedy@vcu.edu)

References


trimMean  Compute trimmed mean for a vector

Description

Computes the trimmed mean for a vector. Note that this implementation follows the Affymetrix code, which gives different results than the standard R function mean().

Usage

trimMean(vec, p1, p2)

Arguments

vec  vector of values
p1   lower percentage for trimming
p2   upper percentage for trimming

Details

The details can be found in the given reference.

Value

A numeric value

Note

Based on Affymetrix MAS5 Statistical SDK source code http://www.affymetrix.com/Auth/support/developer/stat_sdk/STAT_SDK_source.zip

Author(s)

Richard Kennedy (rkennedy@vcu.edu)

References

Index

∗Topic manip
  computeAffxRawQ, 5
  computeAffxSFandSDT, 6
  computeOutlier, 8
  computeRawQ, 9
  computeSFandSDT, 10
  computeZoneInfo, 11
  computeZonenoise, 12
  OneStepBiweightAlgorithm, 1
  SScore, 2
  SScoreBatch, 3

∗Topic univar
  trimMean, 13

computeAffxRawQ, 5
computeAffxSFandSDT, 6
computeOutlier, 3, 5, 8
computeRawQ, 9
computeSFandSDT, 3, 5, 10
computeZoneInfo, 11
computeZonenoise, 12

OneStepBiweightAlgorithm, 1

SScore, 2, 5
SScoreBatch, 3, 3

trimMean, 13