Package 'dyebias'

November 11, 2026

Title The GASSCO method for correcting for slide-dependent gene-specific dye bias

Version 1.72.0

Date 2 March 2016

Author Philip Lijnzaad and Thanasis Margaritis

Description Many two-colour hybridizations suffer from a dye bias that is both gene-specific and slide-specific. The former depends on the content of the nucleotide used for labeling; the latter depends on the labeling percentage. The slide-dependency was hitherto not recognized, and made addressing the artefact impossible. Given a reasonable number of dye-swapped pairs of hybridizations, or of same vs. same hybridizations, both the gene- and slide-biases can be estimated and corrected using the GASSCO method (Margaritis et al., Mol. Sys. Biol. 5:266 (2009), doi:10.1038/msb.2009.21)

Maintainer Philip Lijnzaad <plijnzaad@gmail.com>

License GPL-3

Depends R (>= 1.4.1), marray, Biobase

Suggests limma, convert, GEOquery, dyebiasexamples, methods

URL http://www.holstegelab.nl/publications/margaritis_lijnzaad

biocViews Microarray, TwoChannel, QualityControl, Preprocessing

git_url https://git.bioconductor.org/packages/dyebias

git_branch RELEASE_3_23

git_last_commit 07a87ca

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

dyebias-package 2
dyebias.application.subset 3
dyebias.apply.correction 4
dyebias.boxplot 8
dyebias.estimate.iGSDBs 10
dyebias.monotonicity 13
dyebias.monotonicityplot 15
dyebias.rgplot 17
dyebias.trendplot 20

Index **23**

`dyebias-package` *The GASSCO method for correcting for slide-dependent gene-specific dye bias*

Description

Many two-colour hybridizations suffer from a dye bias that is both gene-specific and slide-specific. The former depends on the content of the nucleotide used for labeling; the latter depends on the labeling percentage. The slide-dependency was hitherto not recognized, and made addressing the artefact impossible. Given a reasonable number of dye-swapped pairs of hybridizations, or of same vs. same hybridizations, both the gene- and slide-biases can be estimated and corrected using the GASSCO method (Margaritis et al., Mol. Sys. Biol. 5:266 (2009), doi:10.1038/msb.2009.21)

Details

Package:	dyebias
Type:	Package
Version:	1.7.1
Date:	26 May 2010
Licence:	GPL-3

Author(s)

Philip Lijnzaad and Thanasis Margaritis
Philip Lijnzaad <p.lijnzaad@umcutrecht.nl> (Maintainer).

See Also

`dyebias.application.subset`, `dyebias.apply.correction`, `dyebias.boxplot`, `dyebias.estimate.iGSDBs`, `dyebias.monotonicity`, `dyebias.monotonicityplot`, `dyebias.rgplot`, `dyebias.trendplot`.

dyebias.application.subset *Return a subset of reporters that can be dye bias-corrected reasonably well*

Description

Convenience function returning a subset of reporters that can be expected to be corrected reasonably well. Often, the logical AND of this set and that of `maW(data.norm) == 1.0` is used. The resulting subset is passed as the `application.subset`-argument to `dyebias.apply.correction`.

Usage

```
dyebias.application.subset(data.raw=NULL, min.SNR=1.5,
                           use.background=FALSE, maxA=15)
```

Arguments

`data.raw`	A `marrayRaw` object whose normalized data is to be dye bias-corrected.
`min.SNR`	The minimum signal to noise ratio to require. It is loosely defined here as the foreground over the background signal. The background signal may not be real; see below.
`use.background`	Logical indicating whether or not to use the background signals `maRb(data.raw)` and `maGb(data.raw)`. If the `data.raw` object does not have them, specify `use.background=FALSE`. This will use the smallest foreground of all reporters instead of the real backgrounds.
`maxA`	The maximum signal that is still allowed.

Details

This routine requires an `marrayRaw` object since only that contains the background intensities. If you only have normalized data, use something like

```
bg <- matrix(0.5, nrow=maNspots(data.norm), ncol=maNsamples(data.norm))
data.raw <- new("marrayRaw", maRf=maR(data.norm), maGf=maG(data.norm),
                             maRb=bg, maGb=bg, maW=maW(data.norm))
```

Value

A matrix of logicals with the same dimensions as those of `maRf{data.raw}` is returned.

Author(s)

Philip Lijnzaad `p.lijnzaad@umcutrecht.nl`

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009) Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

See Also

dyebias.apply.correction

Examples

```
## First load data and estimate the iGSDBs
## (see dyebias.estimate.iGSDBs)



### choose the estimators and which spots to correct:
estimator.subset <- dyebias.umcu.proper.estimators(maInfo(maGnames(data.norm)))

### choose which genes to dye bias correct. Typically, this is based
### both on flagged spots and intensity
application.subset <- maW(data.norm) == 1 &
        dyebias.application.subset(data.raw=data.raw, use.background=TRUE)

summary(application.subset)
```

---

dyebias.apply.correction

Perform dye bias correction using the GASSCO method

Description

Corrects the gene- and slide specific dye bias in a data set, using the GASSCO method by Margaritis et al.

Usage

```
dyebias.apply.correction(data.norm, iGSDBs, estimator.subset=TRUE,
                         application.subset=TRUE, dyebias.percentile=5,
                         minmaxA.perc=25, minA.abs=NULL, maxA.abs=NULL,
                         verbose=FALSE)
```

Arguments

`data.norm` A `marrayNorm` object containing the data whose dye bias should be corrected. This object must be a complete `marrayNorm` object. In particular, `maLabels(maGnames(data.norm))` should be set and indicate the identities of the spots. Spots with the same ID should contain the same oligo or cDNA sequence, and will receive the same dye bias correction.

`iGSDBs` A data frame with the intrinsic gene specific dye bias per reporter (i.e., oligo or cDNA). The data frame would typically have come from a call to `dyebias.estimate.iGSDBs`, but this is not necessary; other estimates can also be used.

The data frame must have (at least) the following columns:

- reporterIdThe name of the reporter. This must match the IDs in `maLabels(maGnames(data.norm))`
- dyebias An estimate of the dye bias
- A
  The average expression value A of this reporter. ($A = (log_2(R)+log_2(G))/2 = (log_2(Cy5)+log_2(Cy3))/2$). The A-value is used to base exclusions on. If you don't have it, you can use any value (but realize that the `minmaxA.perc`, `minA.abs`, `maxA.abs` arguments are still applied).

The order of the rows in this data frame is irrelevant. There must be no rows with duplicate `reporterId` in this frame.

For any reporter in `data.norm` that is not in the `iGSDBs` data frame, an iSGDB of 0.00 is used, i.e. data from such reporters is not dye bias-corrected.

`estimator.subset`
An index indicating which reporters are fit to be used as estimators of the slide bias. This set of reporters is used throughout the whole data set. Reporters that are typically excluded are those corresponding to parasitic DNA elements or mitochondrial genes.

`application.subset`
An index indicating which values must be dye bias-corrected. It should be either a vector with as many values as spots, or a matrix of the same dimensions as `maM(data.norm)`. In former case, the selected spots on all slides with be dye bias-corrected; in the latter, selected spots on selected slides will corrected.

Often it is prudent not to dye bias-correct measurements that are close to the detection limit or close to signal saturation. A convenience function for this is provided; see `dyebias.application.subset`.

`dyebias.percentile`
The slide bias estimation uses a small subset of reporters having the strongest green or red iGSDB, as specified by this percentile. The default should suffice in practically all cases.

`minmaxA.perc` To obtain a robust estimate of the slide bias, the range of the average expression A is trimmed by `minmaxA.perc` percent on both sides; only reporters lying inside this trimmed range are considered as estimators of the slide bias. The default value is 25, meaning that only probes with an average expression within the interquartile range are considered as estimator genes (from these, the top

dyebias.percentile red- and green-biased are then actually used). The default value should suffice in practically all cases.

minA.abs — If specified, reporters with an average expression (A) lower than this value are never considered as estimators of the slide bias. If not specified, reporters with an A-percentile < minmaxA.perc are not considered.

maxA.abs — If specified, reporters with an average expression (A) greater than this are never considered as estimators of the slide bias. If not specified, reporters with an A-percentile < 100-minmaxA.perc are not considered.

verbose — Logical speficying whether to be verbose or not

Details

This function corrects the gene-specific dye bias of two-colour microarrays with the GASSCO method. This method is general, robust and fast, and is based on the observation that the total bias per gene is the product of a slide-specific factor (strongly related to the labeling percentage) and an intrinsic gene-specific factor (iGSDB), which is strongly related to the probe sequence.

The slide bias is estimated from the total bias of the dyebias.percentile percentage of reporters having the strongest iGSDB. The iGSDBs can be estimated with dyebias.estimate.iGSDBs.

If the signal of certain oligos is too weak, or in contrast, tends to be saturated, they are no good estimator of the slide bias. Therefore, only reporters with an average expression level A that is not too extreme are allowed to be slide bias estimators. (This is the reason for the A-column in the iGSDBs data frame).

Full control over which reporters to allow as slide bias estimators is given by the arguments minmaxA.perc, minA.abs, and maxA.abs; see there for details. To not exclude any reporter (e.g., when A is not available and therefore artificially set), you can use minA.abs= -Inf and maxA.abs = Inf.

For further details concerning the method, see the dyebias vignette and the publication. If your research benefits from using this package, we kindly request that you cite this work.

Value

The data returned is a list wit the following elements

data.corrected — A marrayNorm object of the same 'shape' as the input data.norm, but with corrected M values.

estimators — Another list, containing the details of the reporters that were used to obtain an estimate of the slide bias. The contents of the estimators list are:

- green.idsThe IDs of the reporters having the strongest green effect.
- green.cutoffAll reporters in green.ids have an iGSDB below this value.
- green.subsetAn index into the reporters having the strongest green effect.
- green.iGSDBsThe corresponding iGSDBs
- red.idsThe IDs of the reporters having the strongest red effect.
- red.cutoffAll reporters in green.ids have an iGSDB above this value.
- red.subsetAn index into the reporters having the strongest red effect.
- red.iGSDBsThe corresponding iGSDBs

`summary` A data frame summarizing the correction process per slide. It consist of the following columns:

- slideThe slide number
- fileWhich file it came from
- green.correctionThe slide bias based on only the green bias of this slide
- red.correctionThe slide bias based on only the red bias of this slide
- avg.correctionThe total correction factor of this slide. This is in fact the slide bias
- var.ratioThe ratio of the variance of M after and before the correction. The smaller this number, the smaller the variance of M around the mean has become, providing a measure of the success of the dye bias correction. Only data points that were in the `application.subset` are considered.
- reduction.percAs `var.ratio`, but expressed as a percentage. The larger this value, the greater the correction.
- p.valueThe p-value for the signficance of the reduction in variance (F-test; H_0: variances before and after correction are identical)

`data.uncorrected`
The uncorrected input `marrayNorm`, for convenience

Note

Note that the input data should be normalized, and that the dye swaps should **not** have been swapped back (if needed, this can of course be done afterwards).

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009). Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

See Also

`dyebias.estimate.iGSDBs`, `dyebias.application.subset`, `dyebias.rgplot`, `dyebias.maplot`, `dyebias.boxplot`, `dyebias.trendplot`

Examples

```
## First load data and estimate the iGSDBs
## (see dyebias.estimate.iGSDBs)



### choose the estimators and which spots to correct:
estimator.subset <- dyebias.umcu.proper.estimators(maInfo(maGnames(data.norm)))
```

```
  ### choose which genes to dye bias correct:
  application.subset <- (maW(data.norm) == 1 &
              dyebias.application.subset(data.raw=data.raw, use.background=TRUE))

  ### do the correction:
  correction <- dyebias.apply.correction(data.norm=data.norm,
                                         iGSDBs = iGSDBs.estimated,
                                         estimator.subset=estimator.subset,
                                         application.subset = application.subset,
                                         verbose=FALSE)

  ## Not run:
     edit(correction$summary)

## End(Not run)

  ## give overview:
  correction$summary[,c("slide", "file", "avg.correction", "reduction.perc", "p.value")]

  ## and summary:
  summary(as.numeric(correction$summary[, "reduction.perc"]))
```

`dyebias.boxplot` *Creates boxplots of the reporters with the strongest dye bias*

Description

The aim of this routine is to show the magnitude of the dye bias across the data set, as well as the extent to which the GASSCO method could get rid of it. Typically, two boxplots would be shown, one before, one after dye bias correction. For esthetic reasons, the boxplots are usually ordered by the overal slide bias of the uncorrected data set. See also Margaritis et al. (2009), Fig. 1 and 3.

Usage

```
dyebias.boxplot(data, iGSDBs, dyebias.percentile=5,
                application.subset=TRUE, order, output=NULL,
                ylim=c(-4,4), ...)
```

Arguments

`data`	The `marrayNorm` object to boxplot.
`iGSDBs`	A data frame with intrinsic gene-specific dye biases, the same as that used in `dyebias.apply.correction`, probably returned by `dyebias.estimate.iGSDBs`; see there for documentation.
`dyebias.percentile`	The percentile of intrinsic gene specific dye biases (iGSDBs) for which to highlight the reporters.

`application.subset`
: The set of reporters that was eligible for dye bias correction; same argument as for `dyebias.apply.correction`.

`order`
: If `order==FALSE`, no ordering of slides prior to boxplotting takes place. If `order==NULL`, the slides are sorted by increasing slide bias prior to boxplotting. This is typically done for data that is not yet dye bias corrected. This order is also returned as a value. If an `order!=NULL`, the slides are put this order before boxplotting. This is typically done for a dye bias-corrected data set, using the order of the uncorrected set. (See also Fig. 3 in the paper).

`output`
: Specifies the output. If `NULL`, the existing output device is used; if `output` is one of `"X11"`, `"windows"`, `"quartz"`, a new X11 (Unix)/windows (Windows)/quartz (Mac) device is created. If `output` is a string ending in one of `".pdf"`, `".png"`, `".eps"`, `".ps"` is given, a file of that name and type is created and closed afterwards.

`ylim`
: As for `boxplot()`

`...`
: Other arguments (such as `main`, etc.) are passed on to `boxplot()`.

Value

The order obtained, for use in a later call to this same function.

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009) Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

See Also

`dyebias.estimate.iGSDBs`, `dyebias.apply.correction`, `dyebias.rgplot`, `dyebias.maplot`, `dyebias.trendplot`

Examples

```
ylim <- c(-1, 1)

layout(matrix(1:2, nrow=1,ncol=2))

order <- dyebias.boxplot(data=data.norm,
                      iGSDBs=iGSDBs.estimated,  # from e.g. dyebias.estimate.iGSDBs
                      order=NULL,               # i.e., order by increasing slide bias
                      output=NULL,
                      main="before correction",
                      ylim=ylim)
```

```
order <- dyebias.boxplot(data=correction$data.corrected, # from dyebias.apply.correction
                         iGSDBs=iGSDBs.estimated,
                         order=order,              # order by the original slide bias
                         output=NULL,
                         main="after correction",
                         ylim=ylim
                         )
```

`dyebias.estimate.iGSDBs` *Estimate intrinsic gene specific dye biases (part of the GASSCO method)*

Description

Obtain estimates for the instrinsic gene-specific dye bias (iGSDB) using a set of normalized data, as part of the GASSCO method.

Usage

```
dyebias.estimate.iGSDBs(data.norm, is.balanced=TRUE, reference="ref",
                        verbose=FALSE)
```

Arguments

`data.norm`	A `marrayNorm` object containing the data for estimating the dye bias. This object is supposed to be complete. In particular, `maLabels(maGnames(data.norm))` must be set and must indicate the identities of the reporter sequence (i.e., oligo or cDNA sequence) of each spot. This helps identify replicate spots, which are averaged as part of the estimation. If the data is unbalanced (so `is.balanced` is `FALSE`), `maInfo(maTargets(data.norm))` is also required, and should contain at least two attributes: `Cy5` and `Cy3`. Both should indicate the factor value for the respective channel.
`is.balanced`	The use of this argument is discouraged, since designs should generally be balanced. The values other than `TRUE` will become illegal in the future. Logical indicating whether the data set represents a balanced design (which is by far the most common case). A design is balanced if all factor values are present an equal number of times in both the forward and reverse dye orientations. A self-self design is by definition balanced (even if the number of slides is uneven). If `is.balanced` is `TRUE`, the iGSDB estimate is obtained by simply averaging, per reporter, all M values (and the value of the `reference` argument is ignored). If `is.balanced==FALSE`, the design is inferred from the `reference` argument, and subsequently the `limma` package is used to model the dye effect. This is typically done for an unbalanced data set, but there is no harm in setting

is.balanced=FALSE for a design that by itself is already balanced. If there are no missing values in the data, the results of using the simple average and the limma procedure are identical (although LIMMA takes longer to compute the iGSDBs). If the data set contains many missing data points (NA's), the limma estimates differ slightly from the simple averaged estimates (although it is not clear which ones are better).

reference — If the design contains a single common reference, the reference argument should be this common reference (which may not be empty). If the design contains multiple common references, reference should be a vector listing all the common references, and the name of the factor value that is not the common reference should have its own common reference as a prefix. E.g., if two mutant strains mutA and mutB were assayed, each against a separate common reference ref1 and ref2, the reference-argument would be c("ref1", "ref2"), and the Cy3 and Cy5 attributes of maInfo(maTargets(data.norm)) would be values from "ref1:mutA", "ref2:mutA", "ref1:mutB", "ref2:mutB". The colon is not important, but the prefix is, as it allows the association of each sample with its 'own' common reference.

verbose — Logical, indicating wether or not to be verbose.

Details

This function implements the first step of the GASSCO method: estimating the so-called intrinsic gene specific dye biases, or briefly iGSDB. They can be estimated from a (preferably large) data set containing either self-self experiments, or dye-swapped slides.

The assumption underlying this approach is that with self-selfs, or with pairs of dye swaps, the only effect that can lead to systematic changes between Cy5 and Cy3, is in fact the dye effect.

There are two cases to distinguish, the balanced case, and the unbalanced case. In the balanced case, the iGSDB estimate is simply the average M (where $M = log_2(R/G) = log_2(Cy5/Cy3)$) over all slides. A set of slides is balanced if all factor values are present in as many dye-swapped as non-dye-swapped slides. A set of self-self slides is in fact a degenerate form of this, and is therefore also balanced.

In the unbalanced case, one could omit slides until the data set is balanced. However, this is wasteful as we can use linear modelling to obtain estimates. We use the limma package for this (Smyth, 2005). The only unbalanced designs currently supported are a common reference design, and a set of common reference designs.

There are no weights or subset argument to this function; the estimation is done for all reporters found. If there are replicate spots, they are averaged prior to the estimation (the reason being that we are not interested in p-values for the estimate)

Having obtained the iGSDB estimates, the corrections can be applied to either to the hybridizations given by the data.norm argument, or to a different set of slides that is thought to have very similar iGSDBs. Applying the corrections is done with dyebias.apply.correction.

Value

A data frame is returned with as many rows as there are reporters (replicate spots have been averaged), and the following columns:

reporterId	The name of the reporter
dyebias	The intrinsic gene-specific dye bias (iGSDB) of this reporter
A	The average expression level of this reporter in the given data set
p.value	The *p*-value for the dyebias (H_0: dyebias = 0). All *p*-value are set to NA if they were not estimated (i.e., if limma was not run because is.balanced was TRUE)

This data frame is typically used as input to dyebias.apply.correction.

Note

Note that the input data should be normalized, and that the dye swaps should **not** have been swapped back. After all, we're interested in the difference of Cy5 over Cy3, **not** the difference of experiment over reference.

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009) Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

Dudoit, S. and Yang, Y.H. (2002) Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In: Parmigiani, G., Garrett, E.S. , Irizarry, R.A., and Zeger, S.L. (eds.) *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Smyth, G.K. (2005) Limma: linear models for microarray data. In: Gentleman, R., Carey, V., Dudoit, S., Irizarry, R. and Huber, W. (eds). *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, Springer, New York.

See Also

dyebias.apply.correction

Examples

```
iGSDBs.estimated <- dyebias.estimate.iGSDBs(data.norm,
                                            is.balanced=TRUE,
                                            verbose=FALSE)
summary(iGSDBs.estimated)

## Not run:
  hist(iGSDBs.estimated$dyebias, breaks=50)

## End(Not run)
```

`dyebias.monotonicity` *Calculate the degree of monotonicity of the dye bias across the slides. DEPRACATED*

Description

If you order genes by their iGSDB, and hybridizations by slide bias, the graphs of each gene should form a 'fan' out of the origin (see also dyebias.trendplot). This function gives measure of the extent to which this is true.

This function has been depracated, as it is of limited use and takes too long to compute.

Usage

```
dyebias.monotonicity(data,
                     iGSDBs, dyebias.percentile = 5,
                     order = NULL)
```

Arguments

`data`	The `marrayNorm` to determine the degree of monotonicity for
`iGSDBs`	A data frame with intrinsic gene-specific dye biases, the same as that used in `dyebias.apply.correction`, probably returned by `dyebias.estimate.iGSDBs`; see there for documentation.
`dyebias.percentile`	The percentile of intrinsic gene specific dye biases (iGSDBs) for which to highlight the reporters. Default should suffice in almost all cases.
`order`	If `order==NULL`, the slides are sorted by increasing slide bias prior to boxplotting. This is typically done for data that is not yet dye bias corrected. (This order is also returned in the object returned). If `order!=NULL`, the slides are put into this order first. This is typically done for a dye bias-corrected data set, using the order of the uncorrected set.

Details

The total dye bias appears to be the product of iGSDB and slide bias. In other words, it is monotonous (always increasing or always decreasing), both with respect to the intrinsic gene specific dye bias and with respect to the slide bias. This function orders genes by their iGSDB and the slides by slide bias. Subsequently a linear regression of each gene is done, with x being the slide bias rank, (not the slide bias itself), and y being the M. The slopes of each linear regression line should be an increasing array of values, representing the 'fan' of lines. The degree to which this array is increasing is tested using the Mann-Kendall test, and is returned. In the case of uncorrected data, tau is generally larger than 0.3. After correction, tau should be close to zero.

Value

A `dyebias.monotonicity` uses `cor.test`, which returns `htest` object. To this list an extra element, `order`, is added, which indicates the ordering of the data set by slide bias. The degree of monotonicity is indicated by the `estimate` element; its signficance by the `p.value` element.

Note

This function takes very long to compute, since it calculates regressions for each gene.

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009). Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi:10.1038/msb.2009.21.

See Also

dyebias.trendplot, dyebias.monotonicityplot

Examples

```
## Not run:
   options(stringsAsFactors = FALSE)

   library(dyebias)
   library(dyebiasexamples)
   data(data.raw)
   data(data.norm)

   ### obtain estimate for the iGSDBs:
   iGSDBs.estimated <- dyebias.estimate.iGSDBs(data.norm,
                                               is.balanced=TRUE,
                                               verbose=FALSE)

   ### choose the estimators and which spots to correct:
   estimator.subset <- dyebias.umcu.proper.estimators(maInfo(maGnames(data.norm)))

   application.subset <- maW(data.norm) == 1 &
                dyebias.application.subset(data.raw=data.raw, use.background=TRUE)

   ### do the correction:
   correction <- dyebias.apply.correction(data.norm=data.norm,
                                          iGSDBs = iGSDBs.estimated,
                                          estimator.subset=estimator.subset,
                                          application.subset = application.subset,
                                          verbose=FALSE)

cat("monotonicity before correction")
monotonicity <- dyebias.monotonicity(data=data.norm,
                                     iGSDBs=iGSDBs.estimated,
                                     order=NULL)
monotonicity

cat("monotonicity after correction")
```

```
  dyebias.monotonicity(data=correction$data.corrected,
                                    iGSDBs=iGSDBs.estimated,
                                    order= monotonicity$order)

## End(Not run)
```

`dyebias.monotonicityplot` *Show the degree of monotonicity of the dye bias across the slides. DEPRACATED*

Description

If you order genes by their iGSDB, and hybridizations by slide bias, the graphs of each gene should form a 'fan' out of the origin (see also dyebias.trendplot). This function plots the regression slope of each gene, ordered by iGSDB and slide bias. If the uncorrected total dye bias is indeed monotonous, an increasing trend should be visible.

This function has been depracated, as it is of limited use and takes too long to compute.

Usage

```
dyebias.monotonicityplot(data,
                         iGSDBs,
                         dyebias.percentile = 5,
                         order = NULL,
                         output = NULL,
                         pch = 19, cex = 0.3, cex.lab = 1.4,
                         ylim = c(-0.2, 0.2),
                         xlab = "rank", ylab = "slope",
                         sub = NULL,
                         ...)
```

Arguments

`data`	The `marrayNorm` to determine the degree of monotonicity for
`iGSDBs`	A data frame with intrinsic gene-specific dye biases, the same as that used in `dyebias.apply.correction`, probably returned by `dyebias.estimate.iGSDBs`; see there for documentation.
`dyebias.percentile`	The percentile of intrinsic gene specific dye biases (iGSDBs) for which to highlight the reporters. Default should suffice in almost all cases.
`order`	If `order==NULL`, the slides are sorted by increasing slide bias prior to boxplotting. This is typically done for data that is not yet dye bias corrected. (This order is also returned in the object returned). If `order!=NULL`, the slides are put into this order first. This is typically done for a dye bias-corrected data set, using the order of the uncorrected set.

`output` Specifies the output. If `NULL`, the existing output device is used; if `output` is one of `"X11"`, `"windows"`, `"quartz"`, a new X11 (Unix)/windows (Windows)/quartz (Mac) device is created. If `output` is a string ending in one of `".pdf"`, `".png"`, `".eps"`, `".ps"` is given, a file of that name and type is created and closed afterwards.

`pch`, `cex`, `cex.lab`, `ylim`, `xlab`, `ylab`
As for `plot()`

`sub` The subtitle. If `sub==NULL`, a string giving the tau and p-value of the Mann-Kendall test is used; otherwise as for `plot()`

`...` Other arguments are passed on to `plot()`

Details

The total dye bias appears to be the product of iGSDB and slide bias. In other words, it is monotonous (always increasing or always decreasing), both with respect to the intrinsic gene specific dye bias and with respect to the slide bias. This function orders genes by their iGSDB and the slides by slide bias. Subsequently a linear regression of each gene is done, with x being the slide bias rank (not the slide bias itself), and y being the M. The slopes of each linear regression line should be an increasing array of values, representing the 'fan' of lines. The array of slopes is plotted (versus the rank). Generally, a clear trend is visible for uncorrected hybridizations, and the trend has disappeared after dye bias correction.

Value

The order of the slide bias is returned, for use in plotting the behaviour of the regression slopes in the corrected data set.

Note

This function takes very long to compute, since it calculates regressions for each gene.

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009). Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

See Also

`dyebias.monotonicity`, `dyebias.trendplot`

Examples

```
## Not run:
   options(stringsAsFactors = FALSE)

   library(dyebias)
   library(dyebiasexamples)
   data(data.raw)
   data(data.norm)

   ### obtain estimate for the iGSDBs:
   iGSDBs.estimated <- dyebias.estimate.iGSDBs(data.norm,
                                               is.balanced=TRUE,
                                               verbose=FALSE)

   ### choose the estimators and which spots to correct:
   estimator.subset <- dyebias.umcu.proper.estimators(maInfo(maGnames(data.norm)))

   application.subset <- maW(data.norm) == 1 &
                dyebias.application.subset(data.raw=data.raw, use.background=TRUE)

   ### do the correction:
   correction <- dyebias.apply.correction(data.norm=data.norm,
                                          iGSDBs = iGSDBs.estimated,
                                          estimator.subset=estimator.subset,
                                          application.subset = application.subset,
                                          verbose=FALSE)

layout(matrix(1:2, nrow=1,ncol=2))

order <- dyebias.monotonicityplot(data=data.norm,
                      iGSDBs=iGSDBs.estimated, # from e.g. dyebias.estimate.iGSDBs
                      order=NULL,              # i.e., order by increasing slide bias
                      output=NULL,
                      main="before correction"
                      )

order <- dyebias.monotonicityplot(data=correction$data.corrected,
                      iGSDBs=iGSDBs.estimated,
                      order=order,             # order by the original slide bias
                      output=NULL,
                      main="after correction"
                     )

## End(Not run)
```

---

dyebias.rgplot *Produce scatterplots of the hybridization, with strongest dye biases highlighted.*

Description

Plots the $log_2(R)$ vs. $log_2(G)$ (or alternatively M vs. A) signal of one slide, highlighting the reporters with the strongest red and green dye bias. Two lines indicate two-fold change. See also Margaritis et al. (2009), Fig. 1

Usage

```
dyebias.rgplot(data, slide, iGSDBs, dyebias.percentile=5,
               application.subset=TRUE, output=NULL, xlim =
               c(log2(50),log2(50000)), ylim = c(log2(50),log2(50000)),
               xticks = c(100,1000,10000,10000), yticks =
               c(100,1000,10000,10000), pch = 19, cex = 0.3, cex.lab =
               1.4, ...)

dyebias.maplot(data, slide, iGSDBs, dyebias.percentile=5,
               application.subset=TRUE, output=NULL, xlim = c(6,16),
               ylim = c(-2,2), pch = 19, cex = 0.3, cex.lab = 1.4, ...)
```

Arguments

`data`	The `marrayNorm` object to plot one slide of.
`slide`	The index of the slide to plot; must be > 1, and < `maNsamples(data)`
`iGSDBs`	A data frame with intrinsic gene-specific dye biases, the same as that used in `dyebias.apply.correction`, probably returned by `dyebias.estimate.iGSDBs`; see there for documentation.
`dyebias.percentile`	The percentile of intrinsic gene specific dye biases (iGSDBs) for which to highlight the reporters.
`application.subset`	The set of reporters that was eligible for dye bias correction; same argument as for `dyebias.apply.correction`.
`output`	Specifies the output. If `NULL`, the existing output device is used; if `output` is one of `"X11"`, `"windows"`, `"quartz"`, a new X11 (Unix)/windows (Windows)/quartz (Mac) device is created. If `output` is a string ending in one of `".pdf"`, `".png"`, `".eps"`, `".ps"` is given, a file of that name and type is created and closed afterwards.
`xlim`, `ylim`, `xticks`, `yticks`, `pch`, `cex`, `cex.lab`	Graphical parameters; see `par()`
`...`	Other arguments (such as `main` etc.) are passed on to `plot()`.

Value

None.

Note

The highlighted spots are *all spots* with an iGSDB that lies in the top- or bottom- `dyebias.percentile` of iGSDBS. That is, not just the estimator genes are highlighted.

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009). Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

See Also

dyebias.estimate.iGSDBs, dyebias.apply.correction, dyebias.rgplot, dyebias.maplot, dyebias.boxplot, dyebias.trendplot

Examples

```
## show both an RG-plot and an MA-plot of the uncorrected data and the
## corrected data next to each other.

slide <- 3                              # or any other other, of course

layout(matrix(1:4, nrow=2,ncol=2, byrow=TRUE))

dyebias.rgplot(data=data.norm,
               slide=slide,
               iGSDBs=iGSDBs.estimated,   # from dyebias.estimate.iGSDBs
               main=sprintf("RG-plot, uncorrected, slide %d", slide),
               output=NULL)

dyebias.rgplot(data=correction$data.corrected,
               slide=slide,
               iGSDBs=iGSDBs.estimated,
               main=sprintf("RG-plot, corrected, slide %d", slide),
               output=NULL)


dyebias.maplot(data=data.norm,
               slide=slide,
               iGSDBs=iGSDBs.estimated,
               main=sprintf("MA-plot, uncorrected, slide %d",slide),
               output=NULL)

dyebias.maplot(data=correction$data.corrected,
               slide=slide,
               iGSDBs=iGSDBs.estimated,
               main=sprintf("MA-plot, corrected, slide %d",slide),
               output=NULL)
```

`dyebias.trendplot` *Creates a trend-plot of all reporters, binned by dye bias, with all slides ordered by slide bias.*

Description

The aim of this routine is to show the monotonicity of the total dye bias in the (uncorrected) data set. This is to judge whether the total dye bias of one reporter in one hybridization indeed behaves as the product of an intrinsic gene specific dye bias (iGSDB) and a slide specific factor (the slide bias), which is at the heart of the GASSCO method.

Showing the total dye bias of all reporters is too overwhelming, therefore the medians of the total dye bias after binning by intrinsic gene specific dye bias (as given in `dyebias$dyebias`) are plotted.

Usage

```
dyebias.trendplot(data, iGSDBs, dyebias.percentile=5,
                  application.subset=TRUE, n.bins=20, order,
                  output=NULL, ylim=c(-1,1), cex=0.3, lty=1, lwd=1,
                  type="median", main="dye bias trend plot",
                  xlab="slide bias rank", ylab="M", sub=NULL, ...)
```

Arguments

`data`	The `marrayNorm` to trendplot.
`iGSDBs`	A data frame with intrinsic gene-specific dye biases, the same as that used in `dyebias.apply.correction`, probably returned by `dyebias.estimate.iGSDBs`; see there for documentation.
`dyebias.percentile`	The percentile of intrinsic gene specific dye biases (iGSDBs) for which to highlight the reporters. Default should suffice in almost all cases.
`application.subset`	The set of reporters that was eligible for dye bias correction; same argument as for `dyebias.apply.correction`.
`n.bins`	The number of bins into which to classify the reporters, based on their intrinsic gene-specific dye bias. The median of each bin is plotted.
`type`	What to print for each bin and hybridization. Valid values are:

- medianPlot the bin-median: the median of all genes in that bin and hybridization
- meanPlot the mean of all genes in that bin and hybridization
- genePlot the single gene that has the lowest squared distance to the bin-median over *all* hybridizations
- median.genePlot the single gene that has the median squared distance to the bin-medians over *all* hybridizations
- worst.genePlot the single gene that correlates worst with the bin-median over *all* hybridizations

`order` If `order==NULL`, the slides are sorted by increasing slide bias prior to boxplotting. This is typically done for data that is not yet dye bias corrected. This order is also returned as a value. If `order!=NULL`, the slides are put into this order before trendplotting. This is typically done for a dye bias-corrected data set, using the order of the uncorrected set.

`output` Specifies the output. If `NULL`, the existing output device is used; if `output` is one of `"X11"`, `"windows"`, `"quartz"`, a new X11 (Unix)/windows (Windows)/quartz (Mac) device is created. If `output` is a string ending in one of `".pdf"`, `".png"`, `".eps"`, `".ps"` is given, a file of that name and type is created and closed afterwards.

`ylim`, `lty`, `lwd`, `main`, `sub`, `cex`, `xlab`, `ylab` As for `matplot()`

`...` Other arguments are passed on to `matplot()`.

Value

The order obtained, for use in a later call to this same function.

Author(s)

Philip Lijnzaad <p.lijnzaad@umcutrecht.nl>

References

Margaritis, T., Lijnzaad, P., van Leenen, D., Bouwmeester, D., Kemmeren, P., van Hooff, S.R and Holstege, F.C.P. (2009). Adaptable gene-specific dye bias correction for two-channel DNA microarrays. *Molecular Systems Biology*, 5:266, 2009. doi: 10.1038/msb.2009.21.

See Also

`dyebias.estimate.iGSDBs`, `dyebias.apply.correction`, `dyebias.rgplot`, `dyebias.maplot`, `dyebias.monotonicity` `dyebias.monotonicityplot`

Examples

```
## show trend plots of uncorrected and corrected next to each other:
ylim <- c(-0.6, 0.6)

layout(matrix(1:2, nrow=1,ncol=2))

order <- dyebias.trendplot(data=data.norm,
                      iGSDBs=iGSDBs.estimated, # from e.g. dyebias.estimate.iGSDBs
                      order=NULL,              # i.e., order by increasing slide bias
                      output=NULL,
                      main="before correction",
                      ylim=ylim
                      )
```

```
order <- dyebias.trendplot(data=correction$data.corrected, # from dyebias.apply.correction
                        iGSDBs=iGSDBs.estimated,
                        order=order,             # order by the original slide bias
                        output=NULL,
                        main="after correction",
                        ylim=ylim
                       )
```

Index

∗ **hplot**
  dyebias.boxplot, 8
  dyebias.monotonicityplot, 15
  dyebias.rgplot, 17
  dyebias.trendplot, 20
∗ **htest**
  dyebias.monotonicity, 13
∗ **misc**
  dyebias.application.subset, 3
  dyebias.apply.correction, 4
  dyebias.boxplot, 8
  dyebias.estimate.iGSDBs, 10
  dyebias.monotonicity, 13
  dyebias.monotonicityplot, 15
  dyebias.rgplot, 17
  dyebias.trendplot, 20
∗ **package**
  dyebias-package, 2

dyebias *(*dyebias-package*)*, 2
dyebias-package, 2
dyebias.application.subset, *2*, 3, *5*, *7*
dyebias.apply.correction, *2–4*, 4, *8, 9*, *11–13*, *15*, *18–21*
dyebias.boxplot, *2*, *7*, 8, *19*
dyebias.estimate.iGSDBs, *2*, *5–9*, 10, *13*, *15*, *18–21*
dyebias.maplot, *7*, *9*, *19*, *21*
dyebias.maplot *(*dyebias.rgplot*)*, 17
dyebias.monotonicity, *2*, 13, *16*, *21*
dyebias.monotonicityplot, *2*, *14*, 15, *21*
dyebias.rgplot, *2*, *7*, *9*, 17, *19*, *21*
dyebias.trendplot, *2*, *7*, *9*, *14*, *16*, *19*, 20