Package: MBCB (via r-universe)

October 23, 2026

Version 1.67.0

Date 2010-05-10

Title MBCB (Model-based Background Correction for Beadarray)

Description This package provides a model-based background correction method, which incorporates the negative control beads to pre-process Illumina BeadArray data.

Author Yang Xie <Yang.Xie@UTSouthwestern.edu>

Maintainer Bo Yao <Bo.Yao@UTSouthwestern.edu>

License GPL (>=2)

Depends R (>= 2.9.0), tcltk, tcltk2

Imports preprocessCore, stats, utils

URL https://qbrc.swmed.edu/

biocViews Microarray, Preprocessing

Config/pak/sysreqs tcl tk tk-dev tk-table

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:34:44 UTC

RemoteUrl https://github.com/bioc/MBCB

RemoteRef HEAD

RemoteSha e4027a7946e27ae293882eb257e0f732aed44935

Contents

bg.mcmc 2
bg.rma 3
mbcb.correct 4
mbcb.gui 6
mbcb.main 7
mbcb.parseFile 9
MBCBExpressionData 10
printMBCBOutput 11

Index **13**

bg.mcmc *MBCB - Bayesian Background Correction for Illumina Beadarray*

Description

This function provides the means of using only the MCMC (Bayesian) background correction method for the Illumina platform.

Usage

```
bg.mcmc(iter=500, burn=200)
```

Arguments

iter	The iteration count for the Baysian correction.
burn	The number of iterations to burn for the Bayesian correction.

Value

This function returns an array of alpha, mu, and sigma values representing the values computed during the mcmc trial.

Note

This function makes use of two global variables. It will expect obsbead and obsnc are both established prior to calling this function. Obviously, this is not ideal, but R's pass-by-value functionality hindered the ability to pass these matrices as parameters. Using global variables increases performance substantially.

Author(s)

Yang Xie < Yang.Xie@UTSouthwestern.edu>, Min Chen < min.chen@phd.mccombs.utexas.edu>, Jeff Allen < Jeffrey.Allen@UTSouthwestern.edu>

See Also

mbcb.main

Examples

```
data(MBCBExpressionData)
# Use of global variables is obviously not ideal, but with R's pass-by-value
#   setup, we quickly run out of memory without using them on such large
#   arrays

#all of the signals from sample #2
obsbead <<- expressionSignal[,2]
```

```
#the negative control values for this sample
obsnc <<- negativeControl[,2]

#compute the alpha, mu, and sigma values
bg.mcmc();
```

---

bg.rma *MBCB - Robust Multi-Array Average Background Correction for Illumina Beadarray*

Description

This function provides the means of using only the Robust Multi-Array Average background correction method for the Illumina platform.

Usage

```
bg.rma(pm, n.pts = 2^14)
```

Arguments

pm	The data to be background corrected.
n.pts	Pertains to the specificity or accuracy of the method. 2^14 is the default.

Value

A list with two values:

ex.rma: A vector containing the background-corrected values.

para: The statistical summary of the computation.

Note

This method does not make use of negative control beads.

Author(s)

Yang Xie < Yang.Xie@UTSouthwestern.edu>, Min Chen < min.chen@phd.mccombs.utexas.edu>, Jeff Allen < Jeffrey.Allen@UTSouthwestern.edu>

See Also

mbcb.main

Examples

```
data(MBCBExpressionData)

bg.rma(expressionSignal[,2]);
```

`mbcb.correct` *MBCB - Model-Based Background Correction for Illumina Beadarray*

Description

This function is used to background-correct the provided data using the selected correction methods. Normalization is *not* applied.

Usage

```
mbcb.correct (g,
  control,
  npBool=TRUE,
  rmaBool=FALSE,
  mleBool=FALSE,
  bayesBool=FALSE,
  gmleBool=FALSE,
  iter=500,
  burn=200,
  isRawBead=FALSE)
```

Arguments

`g`	The data representing the signal file.
`control`	The data representing the control file.
`npBool`	A boolean value representing the desire to compute the non-parametric background correction values. Set to true if you wish to compute Non-Parametric background correction.
`rmaBool`	A boolean value representing the desire to compute the RMA background correction values. Set to true if you wish to compute RMA background correction.
`mleBool`	A boolean value representing the desire to compute the MLE background correction values. Set to true if you wish to compute MLE background correction.
`bayesBool`	A boolean value representing the desire to compute the Bayes background correction values. Set to true if you wish to compute Bayes background correction.
`gmleBool`	A boolean value representing the desire to compute the GMLE background correction values. Set to true if you wish to compute GMLE background correction.
`iter`	The iteration count; only used in Bayesian correction.
`burn`	The number of iterations which will be burned; only used in Bayesian correction.

`isRawBead` A boolean value representing whether the input files are bead-level or bead-type. If the input is bead-level, set this value to True so that the raw bead-level values can be summarized to bead-type data.

Value

This function returns a complex list which can be grouped into two categories:
Background-corrected Values:

NP: The background corrected values of the Non-Parametric method (or an empty data.frame if this method was not used).

RMA: The background corrected values of the RMA method (or an empty data.frame if this method was not used).

MLE: The background corrected values of the MLE method (or an empty data.frame if this method was not used).

Bayes: The background corrected values of the Bayesian method (or an empty data.frame if this method was not used).

Average Values

AvgNP: A data.frame of statistics pertaining to the average of the NP normalization method (or an empty data.frame if this method was not used).

AvgRMA: A data.frame of statistics pertaining to the average of the RMA method (or an empty data.frame if this method was not used).

AvgMLE: A data.frame of statistics pertaining to the average of the MLE method (or an empty data.frame if this method was not used).

AvgBayes: A data.frame of statistics pertaining to the average of the Bayesian method (or an empty data.frame if this method was not used).

These values have not been normalized or log2 transformed. See `mbcb.main` for such functionality.

Author(s)

Yang Xie < `Yang.Xie@UTSouthwestern.edu`>, Min Chen < `min.chen@phd.mccombs.utexas.edu`>, Jeff Allen < `Jeffrey.Allen@UTSouthwestern.edu`>

See Also

`mbcb.main`

Examples

```
data(MBCBExpressionData)
mbcb.correct(expressionSignal, negativeControl);
```

mbcb.gui *MBCB - Model-Based Background Correction for Illumina Beadarray*

Description

This file provides the Graphical-User-Interface for the MBCB package.

Usage

```
mbcb.gui()
```

Value

This GUI will allow you to easily input data files and a negative-control file and will output a log2-transformed background-corrected file.

Note

The input files should be tab-delimited files in the following form:

signal.txt

```
    g.1 g.2 g.3 g.4
10181072_239_rc-S 160.3 776.4 135.8 407.7
10181072_290-S  138.1 219.8 122.1 142.4
...   ... ... ... ...
```

neg-con.txt

```
id g.1 g.2 g.3 g.4
50133 127 213.5 82 103
50315 232 295 143.5 156
... ... ... ... ...
```

Note that both of these examples come from a test involving four trials. Your data could use whatever number you'd like.

Author(s)

Yang Xie < Yang.Xie@UTSouthwestern.edu>, Min Chen < min.chen@phd.mccombs.utexas.edu>, Jeff Allen < Jeffrey.Allen@UTSouthwestern.edu>

See Also

mbcb.main

Examples

```
data(MBCBExpressionData)

#create files from the data provided in this package in the current directory
write.table(expressionSignal, 'signal.txt', sep="\t");
write.table(negativeControl, 'negative.control.txt', sep="\t");

#open the GUI; you can use the files just created as input for the signal and
#   negative control files.
mbcb.gui();
```

`mbcb.main` *MBCB - Model-Based Background Correction for Illumina Beadarray*

Description

This is the main function which incorporates all the others. This should be the most straightforward and autonomous function in the MBCB package.

Usage

```
mbcb.main (signal,
  control,
  npBool=TRUE,
  rmaBool=FALSE,
  mleBool=FALSE,
  bayesBool=FALSE,
  gmleBool=FALSE,
  paramEstFile="param-est",
  bgCorrectedFile="bgCorrected",
  iter=500,
  burn=200,
  normMethod="none",
  isRawBead=FALSE)
```

Arguments

`signal`	The data representing the signal file.
`control`	The data representing the control file.
`npBool`	A boolean value representing the desire to compute the non-parametric background correction values.
`rmaBool`	A boolean value representing the desire to compute the RMA background correction values.
`mleBool`	A boolean value representing the desire to compute the MLE background correction values.

bayesBool	A boolean value representing the desire to compute the Bayes background correction values.
gmleBool	A boolean value representing the desire to compute the GMLE background correction values.
paramEstFile	The *base* file name to which suffixes and a file extension will be appended (i.e. 'C:/output'). These files will store the parameter estimates of each background correction method selected.
bgCorrectedFile	The *base* file name to which suffixes and a file extension will be appended (i.e. 'C:/output'). These files will store the background corrected intensities
iter	The iteration count; only used in Bayesian correction.
burn	The number of iterations which will be burned; only used in Bayesian correction.
normMethod	The normalization method to be used. By default, none will be applied. The choices are: *none* - no normalization will be applied. *quant* - Quantile-Quantile normalization will be applied (requires the affy and affyio packages be present). *median* - Median or Global normalization will be applied.
isRawBead	A boolean value representing whether the input files are bead-level or bead-type. If the input is bead-level, set this value to True so that the raw bead-level values can be summarized to bead-type data.

Value

The function will compute and output log2-tranformed values for the desired background correction methods. Rather than returning this data as R objects, the output is written to files based on the names given in *paramEstFile* and *bgCorrectedFile*.

Note

You can use `mbcb.parseFile` to create the signal and control matrices from the given files.

Author(s)

Yang Xie < `Yang.Xie@UTSouthwestern.edu`>, Min Chen < `min.chen@phd.mccombs.utexas.edu`>, Jeff Allen < `Jeffrey.Allen@UTSouthwestern.edu`>

See Also

`mbcb.correct` `mbcb.parseFile`

Examples

```
data(MBCBExpressionData)
#Use of global variables is obviously not ideal, but with R's pass-by-value
#   setup, we quickly run out of memory without using them on such large
#   arrays
```

```
mbcb.main(expressionSignal, negativeControl);
```

`mbcb.parseFile` *MBCB - Model-Based Background Correction for Illumina Beadarray*

Description

This function is used to read the files provided into a format which will be usable by the MBCB package (data.frame).

Usage

```
mbcb.parseFile(sigFile, conFile, isRawBead = FALSE)
```

Arguments

`sigFile`	The file-name (character string) representing the signal file.
`conFile`	The file-name (character string) representing the control file.
`isRawBead`	A boolean value representing whether the input files are bead-level or bead-type. If the input is bead-level, set this value to True so that the raw bead-level values can be summarized to bead-type data.

Value

This function will return a list containing two data frames: *sig* and *con*.

sig: Represents the signal file.

con: Represents the negative control data.frame.

Note

The input files should be tab-delimited files in the following form:

signal.txt

```
    g.1 g.2 g.3 g.4
10181072_239_rc-S 160.3 776.4 135.8 407.7
10181072_290-S  138.1 219.8 122.1 142.4
...   ... ... ... ...
```

neg-con.txt

```
id g.1 g.2 g.3 g.4
50133 127 213.5 82 103
50315 232 295 143.5 156
... ... ... ... ...
```

Note that both of these examples come from a test involving four trials. Your data could use whatever number you'd like.

Author(s)

Yang Xie < Yang.Xie@UTSouthwestern.edu>, Min Chen < min.chen@phd.mccombs.utexas.edu>, Jeff Allen < Jeffrey.Allen@UTSouthwestern.edu>

See Also

mbcb.correct

Examples

```
data(MBCBExpressionData)

# Create files from the data provided in this package in the current
#     directory
# Obviously, this is the opposite of what the function does, but we need to
#     write sample files for the sake of the demonstration
# The signal and negative control files can be used by calling the above
#     command without the writing and reading of the data.
write.table(expressionSignal, 'signal.txt', sep="\t");
write.table(negativeControl, 'negative.control.txt', sep="\t");

#read in those files just created.
data <- mbcb.parseFile('signal.txt', 'negative.control.txt');
signal <- data$sig;
negCon <- data$con;
```

MBCBExpressionData *MBCB - Bayesian Background Correction for Illumina Beadarray*

Description

The MBCBExpressionData dataset consists of two dataframes, expressionSignal and negativeControl. See the vignette for more detail regarding these data.

Usage

```
data(MBCBExpressionData)
```

printMBCBOutput *MBCB - Model-Based Background Correction for Illumina Beadarray*

Description

This function is used to neatly output the values created by the other methods in the MBCB package.

Usage

```
printMBCBOutput(sig,
  average,
  rmaBool,
  npBool,
  mleBool,
  bayesBool,
  gmleBool,
  avgOutputFile,
  detailOutputFile)
```

Arguments

sig	A list corresponding to the backgrdoun-corrected signal values (as is generated by the mbcb.correct function). The list should have elements for all desired background correction methods named 'NP', 'RMA', 'Bayes', and/or 'MLE'.
average	A list containing the average values (as is generated by the mbcb.correct function). The list should have elements for all desired background correction methods named 'NP', 'RMA', 'Bayes', and/or 'MLE'.
npBool	A boolean value representing the desire to compute the non-parametric background correction values.
rmaBool	A boolean value representing the desire to compute the RMA background correction values.
mleBool	A boolean value representing the desire to compute the MLE background correction values.
bayesBool	A boolean value representing the desire to compute the Bayes background correction values.
gmleBool	A boolean value representing the desire to compute the GMLE background correction values.
avgOutputFile	The *base* file name to which suffixes and a file extension will be appended (i.e. 'C:/output'). These files will store the average values of each background correction method selected.
detailOutputFile	The *base* file name to which suffixes and a file extension will be appended (i.e. 'C:/output'). These files will store the background corrected intensities

Value

This function prints corresponding CSV files based on which background-correction methods were selected.

Author(s)

Yang Xie < Yang.Xie@UTSouthwestern.edu>, Min Chen < min.chen@phd.mccombs.utexas.edu>, Jeff Allen < Jeffrey.Allen@UTSouthwestern.edu>

See Also

mbcb.main

Index

∗ **datasets**
MBCBExpressionData, 10
∗ **models**
bg.mcmc, 2
bg.rma, 3
mbcb.correct, 4
mbcb.gui, 6
mbcb.main, 7
mbcb.parseFile, 9
printMBCBOutput, 11

bg.mcmc, 2
bg.rma, 3

expressionSignal *(*MBCBExpressionData*)*, 10

mbcb.correct, 4, *8*, *10*
mbcb.gui, 6
mbcb.main, *2*, *3*, *5*, *6*, 7, *12*
mbcb.parseFile, *8*, 9
MBCBExpressionData, 10

negativeControl *(*MBCBExpressionData*)*, 10

printMBCBOutput, 11