Package ‘bgx’

Title  Bayesian Gene eXpression
Version  1.66.0
Author  Ernest Turro, Graeme Ambler, Anne-Mette K Hein
Maintainer  Ernest Turro <et341@cam.ac.uk>
Description  Bayesian integrated analysis of Affymetrix GeneChips
License  GPL-2
Depends  R (>= 2.0.1), Biobase, affy (>= 1.5.0), gcrma (>= 2.4.1)
Suggests  affydata, hgu95av2cdf
biocViews  Microarray, DifferentialExpression
Imports  Rcpp (>= 0.11.0)
LinkingTo  Rcpp
git_url  https://git.bioconductor.org/packages/bgx
git_branch  RELEASE_3_17
git_last_commit  21cf621
git_last_commit_date  2023-04-25
Date/Publication  2023-10-12

R topics documented:

  analysis.bgx  ..........................................................  2
  bgx  .................................................................  3
  mcmc.bgx  ............................................................  5
  readOutput.bgx  ......................................................  7
  saveAffinityPlot.bgx  ...............................................  7
  setupVars.bgx  ......................................................  8

Index  10
Description

Functions for plotting expression densities, differential expression densities, histogram of proportion of differentially expressed genes, etc.

Usage

```r
plotExpressionDensity(bgxOutput, gene=NULL, normalize=c("none","mean","loess"),...)
plotDEDensity(bgxOutput, gene=NULL, conditions=c(1,2), normalize=c("none","mean","loess"), normgenes=c(1:length(bgxOutput["geneNames"])), ...)
plotDEHistogram(bgxOutput, conditions=c(1,2), normalize=c("none","mean","loess"), normgenes=c(1:length(bgxOutput["geneNames"])), df=floor(1.8 * log10(length(bgxOutput["geneNames"]))))
rankByDE(bgxOutput, conditions=c(1,2),normalize=c("none","mean","loess"), normgenes=c(1:length(bgxOutput["geneNames"])), absolute=TRUE)
plotDiffRank(bgxOutput, conditions=c(1,2),normalize=c("none","mean","loess"), normgenes=c(1:length(bgxOutput["geneNames"])), ymax=NULL)
```

Arguments

- `bgxOutput`: A list obtained from running `readOutput.bgx` on a BGX output directory.
- `gene`: Which gene to analyse. This can either be an index or a name.
- `conditions`: Indices of conditions to compare.
- `normalize`: "none": do not normalise posterior distributions of mu. "mean": normalise by scaling posterior distributions of mu for conditions > 1 to have the same mean as the posterior distribution of mu for condition 1. "loess": same as "mean" but use loess normalisation.
- `normgenes`: Which genes to use for loess normalisation. By default, use all genes.
- `df`: Residual degrees of freedom. Decrease to 6 if the histogram fit goes haywire.
- `absolute`: Rank genes by absolute differential expression.
- `ymax`: Specify upper limit of y axis.
- `...`: Parameters to pass to density function (where applicable).

Details

- `plotExpressionDensity` plots gene expression distributions under various conditions for the specified gene.
- `plotDEDensity` plots the differential expression distribution between two conditions for a given gene.
- `plotDEHistogram` plots a histogram of differential expression between two conditions and estimates the number of up and down regulated differentially expressed genes.
- `rankByDE` ranks genes by differential expression and returns ordering and corresponding DE values in a matrix.
- `plotDiffRank` plots 2.5-97.5% confidence intervals for ranked differential expression estimates.
Value

None, except plotDERank, which returns a matrix of genes ranked by differential expression.

Author(s)

Ernest Turro

See Also

bgx, standalone.bgx, readOutput.bgx, plotExpressionDensity, plotDEDensity, plotDEHistogram

Description

'bgx' estimates Bayesian Gene eXpression (BGX) measures from an AffyBatch object.

'standalone.bgx' creates various files needed by the bgx standalone binary and places them in a directory. One of these files is 'infile.txt'. In order to run standalone BGX, compile it and run 'bgx <path_to_infile.txt>' from the command line.

Usage

bgx(aData, samplesets = NULL, genes = NULL, genesToWatch = NULL, burnin = 8192, iter = 16384, output = c("minimal","trace","all"), probeAff = TRUE, probecat_threshold = 100, adaptive = TRUE, rundir = ".")

standalone.bgx(aData, samplesets = NULL, genes = NULL, genesToWatch = NULL, burnin = 8192, iter = 16384, output = c("minimal","trace","all"), probeAff = TRUE, probecat_threshold = 100, adaptive = TRUE, batch_size = 50, optimalAR = 0.44, inputdir = "input")

Arguments

aData An AffyBatch object.
samplesets A numeric vector specifying which condition each array belongs to. E.g. if samplesets=c(2,2), then the first two replicates belong to one condition and the last two replicates belong to another condition. If NULL, each array is assumed to belong to a different condition. If the aData object contains information about the experiment design in its phenoData slot, this argument is not required.
genes A numeric vector specifying which genes to analyse. If NULL, all genes are analysed.
genestoWatch A numeric vector specifying which genes to monitor closely amongst those chosen to be analysed (see below for details).
burnin  Number of burn-in iterations.
iter    Number of post burn-in iterations.
output  One of "minimal", "trace" or "all". See below for details.
probeAff Stratify the mean (lambda) of the cross-hybridisation parameter (H) by categories according to probe-level sequence information.
probecat_threshold Minimum amount of probes per probe affinity category.
adaptive Adapt the variance of the proposals for Metropolis Hastings objects (that is: S, H, Lambda, Eta, Sigma and Mu).
batch_size Size of batches for calculating acceptance ratios and updating jumps.
optimalAR Optimal acceptance ratio.
rundir The directory in which to save the output runs.
inputdir The name of the directory in which to place the input files for the standalone binary.

details
- genesToWatch Specify the subset of genes for which thinned samples from the full posterior distributions of log(S+1) (x) and log(H+1) (y) are collected.
- output Output the following to disk:
  - "minimal" The gene expression measure (muave), thinned samples from the full posterior distributions of mu (mu.[1..c]), where 'c' is the number of conditions, the integrated autocorrelation time (IACT) and the Markov chain Monte Carlo Standard Error (MCSE) for each gene under each condition. Note that the IACT and MCSE are calculated from the thinned samples of mu.
  - "trace" The same as "minimal" plus thinned samples from the full posterior distributions of sigma2 (sigma2.[1..c]), lambda (lambda.[1..s]), eta2 (eta2), phi (phi) and tau2 (tau2), where 's' is the number of samples. If there are probes with unknown sequences, output a thinned trace of their categorisation.
  - "all" The same as "trace" plus acceptance ratios for S (sacc), H (hacc), mu (muacc), sigma (sigmaacc), eta (etaacc) and lambda (lambdasacc).

value
- 'bgx' returns an ExpressionSet object containing gene expression information for each gene under each condition (not each replicate).
- 'standalone.bgx' returns the path to the BGX input files.

note
The bgx() method and the bgx standalone binary create a directory in the working directory called 'run.x' (x:1,2,3,...), wherein files are placed for further detailed analysis.

author(s)
Ernest Turro
mcmc.bgx

References


Examples

```r
# This example requires the 'affydata' and 'hgu95av2cdf' packages
if(require(affydata) && require(hgu95av2cdf)) {
  data(Dilution)
  eset <- bgx(Dilution, samplesets=c(2,2), probeAff=FALSE, burnin=4096, iter=8192,
              genes=c(12500:12599), output="all", rundir=file.path(tempdir()))
}
```

mcmc.bgx

Internal wrapper function for calling the bgx C++ function.

Description

This internal function calls the bgx method in a loaded bgx shared object (bgx.so/bgx.dll)

Usage

```r
mcmc.bgx(pm, mm, samplesets, probesets, numberCategories, categories, unknownProbeSeqs, numberOfUnknowns, numberGenesToWatch, whichGenesToWatch, whichProbesToWatch, iter, burnin, adaptive, batch_size=50, optimalAR=0.44, output, samplenames = "unknown", subsample = ifelse(iter > 1024, iter/1024, 1), seed = 192492, rundir)
```

Arguments

- `pm`: Perfect Match probes
- `mm`: MisMatch probes
- `samplesets`: A numeric vector specifying which condition each array belongs to. E.g. if samplesets=c(2,2), then the first two replicates belong to one condition and the last two replicates belong to another condition. If NULL, each array is assumed to belong to a different condition.
- `probesets`: A numeric vector specifying how probes are grouped into probesets.
numberCategories
   Number of probe affinity categories.

categories
   A numeric vector specifying which category each probe belongs to.

unknownProbeSeqs
   A numeric vector specifying which probes lack sequence information.

numberOfUnknownProbeSeqs
   Number of probes lacking sequence information.

numberGenesToWatch
   How many genes to monitor closely.

whichGenesToWatch
   A numeric vector specifying which genes to monitor closely.

whichProbesToWatch
   The starting position for each probe in each gene to monitor closely.

iter
   Number of post burn-in iterations.

burnin
   Number of burn-in iterations.

adaptive
   Use adaptive MCMC for better mixing.

batch_size
   Batch size for adaptive MCMC.

optimalAR
   Optimal acceptance ratio.

output
   One of "minimal", "trace", "diagnostic" or "mcse".

samplenames
   Vector of names for each array.

subsample
   Subsampling interval.

seed
   Seed for PRNG.

rundir
   The directory in which to place the output run directories.

Details

See bgx for more details.

Value

The name of the output directory.

Note

You shouldn't call this function directly, but if you do, make sure the appropriate shared object is loaded.

Author(s)

Ernest Turro

See Also

bgx, standalone.bgx
readOutput.bgx

Read in the output from a BGX run.

Description

readOutput.bgx reads in output from BGX which can then be fed into BGX analysis functions.

Usage

readOutput.bgx(...)

Arguments

... Paths of BGX output directories. If you specify more than one path, then the runs will be combined such that each condition from each run is treated as different from all the others.

Details

See bgx for more details.

Value

A list containing data from the BGX output.

Author(s)

Ernest Turro

See Also

bgx, standalone.bgx, plotExpressionDensity, plotDEDensity, plotDEHistogram

saveAffinityPlot.bgx

Save a plot of affinity categorisation.

Description

This internal function saves a plot showing how probes were categorised into affinity categories.

Usage

saveAffinityPlot.bgx(originalAffinities, categories, dir, probecat_threshold)
**Arguments**

- `originalAffinities`: The affinities of the probes.
- `categories`: The categories of the probes.
- `dir`: Name of a directory in which to save the plot.
- `probecat_threshold`: The minimum number of probes per category that was used to categorise the probes.

**Author(s)**

Ernest Turro

**References**

See bgx

**See Also**

bgx

---

**setupVars.bgx**

*Initialise variables needed to run BGX simulation.*

**Description**

This internal function initialises several variables, which it returns in a list.

**Usage**

```r
setupVars.bgx(data, samplesets, genes, genesToWatch, probeAff, probecat_threshold, rounding_dec_places = 1)
```

**Arguments**

- `data`: An AffyBatch object.
- `samplesets`: A numeric vector specifying which condition each array belongs to. E.g. if `samplesets=c(2,2)`, then the first two replicates belong to one condition and the last two replicates belong to another condition. If NULL, each array is assumed to belong to a different condition.
- `genes`: A numeric vector specifying which genes to analyse. If NULL, all genes are analysed.
- `genesToWatch`: A numeric vector specifying which genes to monitor closely amongst those chosen to be analysed (see below for details).
- `probeAff`: Stratify the mean (lambda) for the cross-hybridisation parameter (H) by categories according to probe-level sequence information.
setupVars.bgx

probecat_threshold
Minimum amount of probes per probe affinity category.

rounding_dec_places
The initial probe categorisation is done by rounding affinities to the nearest
rounding_dec_places decimal places. 1 is a good value.

Value
A list:

pm       Perfect Match probes.
mm       MisMatch probes.
samplesets    A numeric vector specifying which condition each array belongs to. E.g. if
             samplesets=c(2,2), then the first two replicates belong to one condition and the
             last two replicates belong to another condition. If NULL, each array is assumed
             to belong to a different condition.
probesets A numeric vector specifying how probes are grouped into probesets.
numberOfCategories Number of probe affinity categories.
categories A numeric vector specifying which category each probe belongs to.
unknownProbeSeqs A numeric vector specifying which probes lack sequence information.
numberOfUnknownProbeSeqs Number of probes lacking sequence information.
genestoWatch A numeric vector specifying which genes to monitor closely.
firstProbeInEachGeneToWatch    The starting position for each probe in each gene to monitor closely.
numArrays   Number of arrays.

Note
This function shouldn’t be called directly.

Author(s)
Ernest Turro

References
See bgx

See Also
bgx
Index

* IO
  analysis.bgx, 2
  readOutput.bgx, 7
* internal
  mcmc.bgx, 5
  saveAffinityPlot.bgx, 7
  setupVars.bgx, 8
* manip
  bgx, 3
  analysis.bgx, 2
  bgx, 3, 6–9
  mcmc.bgx, 5
  plotDEDensity, 3, 7
  plotDEDensity (analysis.bgx), 2
  plotDEHistogram, 3, 7
  plotDEHistogram (analysis.bgx), 2
  plotDiffRank (analysis.bgx), 2
  plotExpressionDensity, 3, 7
  plotExpressionDensity (analysis.bgx), 2
  rankByDE (analysis.bgx), 2
  readOutput.bgx, 2, 3, 7
  saveAffinityPlot.bgx, 7
  setupVars.bgx, 8
  standalone.bgx, 3, 6, 7
  standalone.bgx (bgx), 3