Package ‘bgx’
April 25, 2017

Title     Bayesian Gene eXpression
Version   1.42.0
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), germa (>= 2.4.1)
Suggests  affydata, hgu95av2cdf
biocViews Microarray, DifferentialExpression
NeedsCompilation yes

R topics documented:

   analysis.bgx ....................................................... 1
   bgx ............................................................... 3
   readOutput.bgx .................................................. 5

Index

   analysis.bgx    Analyse BGX output.

Description

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

Usage

   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

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

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`
Fully Bayesian integrated approach to the analysis of Affymetrix GeneChip data

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

```r
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

References


Examples

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

**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 a different 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
Index

*Topic IO
  analysis.bgx, 1
  readOutput.bgx, 5
*Topic manip
  bgx, 3

analysis.bgx, 1

bgx, 2, 3, 5

plotDEDensity, 2, 5
plotDEDensity(analysis.bgx), 1
plotDEHistogram, 2, 5
plotDEHistogram(analysis.bgx), 1
plotDiffRank(analysis.bgx), 1
plotExpressionDensity, 2, 5
plotExpressionDensity(analysis.bgx), 1

rankByDE(analysis.bgx), 1
readOutput.bgx, 2, 5

standalone.bgx, 2, 5
standalone.bgx(bgx), 3