BGmix

October 25, 2011

`BGmix-package` *BGmix fits a variety of Bayesian hierarchical models for finding*

Description

BGmix uses a C++ routine to fit the chosen model via an MCMC algorithm. Files are written to a sub-directory in the working directory. The package includes R functions for reading the results into R, and several plotting functions and functions for estimating error rates.

Details

Package:	BGmix
Type:	Package
Version:	1.0
Date:	2007-02-01
License:	GPL

See Vignette for details of how to use this package (use openVignette()).

Author(s)

Alex Lewin and Natalia Bochkina

Maintainer: Alex Lewin <a.m.lewin@imperial.co.uk>

References

Lewin, A., Bochkina, N. and Richardson, S. (2007), Fully Bayesian mixture model for differential gene expression: simulations and model checks. http://www.bgx.org.uk/publications.html

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1,trace.pred=1)
```

```
## Basic plot of parameters
params <- ccParams(outdir)
plotBasic(params,ybar,ss)

## plots of FDR and related quantities
fdr <- calcFDR(params)
par(mfrow=c(1,2))
plotFDR(fdr)

## plots of Bayesian p-values
## for predictive checks of mixture prior
pred <- ccPred(outdir,q.trace=TRUE)
plotPredChecks(pred$pval.ybar.mix2,params$pc,probz=0.5)

## plots of predictive density superimposed on data
plotMixDensity(params,pred,ybar,ss)
```

`BGmix` *Fit the BGmix differential expression model.*

Description

This is the main function of the BGmix package. It calls the C++ code which performs the MCMC to fit the BGmix model.

Usage

```
BGmix(ybar, ss, nreps, neffects = 2, xx = matrix(c(1, 1, -0.5, 0.5),
ncol = 2, byrow = T), ntau = NULL, indtau = NULL, jstar = 1, niter =
10000, nburn = 10000, nthin = 10, seed = 12345, move.choice.bz = 4,
move.choice.aa = 1, move.choice.lam = 0, move.choice.tau = 1,
move.choice.eta = 1, trace.out = 1, trace.pred = 0, sig.aa = 0.1,
tau.eps = 50, lambda.up.init=1.5, lambda.down.init=1.5,
datafilename.ybar = NULL, xfilename = NULL, itfilename =
NULL, rundir=".")
```

Arguments

`ybar`	matrix no. genes x no. experimental conditions. Mean log gene expression for each gene in each condition.
`ss`	matrix no. genes x no. experimental conditions. Sample variance of log gene expression for each gene in each condition.
`nreps`	vector containing the number of replicate arrays in each experimental condition
`neffects`	number of effect parameters per gene (eg. 2 for unpaired differential expression)
`xx`	design matrix: no. effects x no. experimental conditions. See Vignette for specification of design matrix. Default is for unpaired differential expression.
`ntau`	number of variances per gene
`indtau`	label for each condition indicating which variance grouping that condition belongs to. See Vignette for more detail.
`jstar`	Label of the effect parameter which has the mixture prior. Labels start at 0, as in C++. If no parameter has a mixture prior, set jstar=-1.

`niter` no. MCMC iterations after burn-in. This must be at least 100 for the function to work (or else set to zero).

`nburn` no. MCMC iterations for burn-in. This must be at least 100 for the function to work (or else set to zero).

`nthin` thinning parameter for MCMC iterations

`seed` initial value for random seed

`move.choice.bz` indicates choice of mixture prior: 1 for point mass null + Uniform alternatives, 4 for point mass null + Gamma alternatives, 5 for small Normal null + Gamma alternatives

`move.choice.aa` if this is 1, hyperparameter a for gene variances is updated, if this is 0 it is fixed.

`move.choice.lam` if this is 1, hyperparameter lambda for mixture prior is updated, if this is 0 it is fixed.

`move.choice.tau` indicates choice of prior on gene variances: 1 for Inverse Gamma, 2 for log Normal.

`move.choice.eta` if this is 1, hyperparameter eta for mixture prior is updated, if this is 0 it is fixed.

`trace.out` if this is 1, output trace of model parameters, if this is 0, no output.

`trace.pred` if this is 1, output trace of predictive quantities, if this is 0, no output.

`sig.aa` step-size in random walk update for a (hyperparameter for gene variances distribution)

`tau.eps` Value of epsilon used in the small Normal null mixture component.

`lambda.up.init` init or fixed value of lambda+ (parameter of Gamma mixture component)

`lambda.down.init` init or fixed value of lambda- (parameter of Gamma mixture component)

`datafilename.ybar` character. Name describing the data set (by default this is taken from the name of the ybar argument).

`xfilename` character. Name describing the design matrix.

`itfilename` character. Name describing the indtau parameter.

`rundir` character. Path for saving output files. A new sub-directory is created in the `rundir` directory.

Details

The C++ code writes a count down on the screen, to give an indication of how long the code has to run. Output is written to a sub-directory of the working directory. This sub-directory is created automatically, and its name is printed by the C++ code to the screen.

Value

The output directory is returned (character).

Author(s)

Alex Lewin

References

Lewin, A., Bochkina, N. and Richardson, S. (2007), Fully Bayesian mixture model for differential gene expression: simulations and model checks. http://www.bgx.org.uk/publications.html

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
BGmix(ybar, ss, c(8,8), nburn=0, niter=1000, nthin=1)
```

EstimatePi0 *Proportion of the variables under the null hypothesis*

Description

Estimate of the proportion of the variables under the null hypothesis using tail posterior probabilities

Usage

```
EstimatePi0(tpp, pp0, plot = T)
```

Arguments

tpp	observed tail posterior probability
pp0	a vector of tail posterior probability under H0
plot	if True, estimated pi0 at different locations and the median estimate is plotted

Details

Use Storey (2002) approach to estimate pi0

Value

estimate of pi0 = proportion of non-differentially expressed genes

Author(s)

Natalia Bochkina

References

Bochkina N., Richardson S. (2007) Tail posterior probability for inference in pairwise and multiclass gene expression data. Biometrics (in press).

See Also

TailPP, FDRplotTailPP,histTailPP

Examples

```
 data(ybar, ss)
 nreps <- c(8,8)

## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
 outdir <- BGmix(ybar, ss, nreps, jstar=-1, nburn=0, niter=100, nthin=1)

 params <- ccParams(outdir)
 res <-  ccTrace(outdir)

 tpp.res <- TailPP(res, nreps, params, plots  = FALSE)
 pi0 <- EstimatePi0(tpp.res$tpp, tpp.res$pp0)
```

`FDRforTailPP` *FDR for tail posterior probability*

Description

Calculate the false discovery rate (FDR) for the tail posterior probability

Usage

```
FDRforTailPP(tpp, a1, a2 = NULL, n.rep1, n.rep2 = NULL, prec = 0.05, p.cut = 0.7
```

Arguments

`tpp`	vector of tail posterior probabilities
`a1`	posterior mean of the shape parameter of the inverse gamma distribution - prior for the variance in condition 1
`a2`	posterior mean of the shape parameter of the inverse gamma distribution - prior for the variance in condition 2
`n.rep1`	number of replicates in condition 1
`n.rep2`	number of replicates in condition 2
`prec`	precision of the estimate of the cumulative distribution function of tail posterior probability under H0 (at points 1 - k*prec, k =1,2,..)
`p.cut`	to save time, calculate FDR only for cutoffs on tail posterior probability > p.cut
`N`	simulation size for tail posterior probability under H0
`pp0`	a vector of simulated tail posterior probabilities under H0
`plot`	if True, the estimated pi0 at different locations and the median estimate is plotted

Value

`pi0`	estimate of pi0 - proportion of non-differentially expressed genes
`FDR`	estimate of FDR for all (distinct) cutoffs > p.cut

Author(s)

Natalia Bochkina

References

Bochkina N., Richardson S. (2007) Tail posterior probability for inference in pairwise and multiclass gene expression data. Biometrics.

See Also

TailPP, FDRplotTailPP,histTailPP,EstimatePi0

Examples

```
 data(ybar, ss)
 nreps <- c(8,8)

## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
 outdir <- BGmix(ybar, ss, nreps, jstar=-1, nburn=0, niter=100, nthin=1)

 params <- ccParams(outdir)
 res <-  ccTrace(outdir)

 tpp.res <- TailPP(res, nreps, params, plots  = FALSE)
 FDR.res = FDRforTailPP(tpp.res$tpp, a1 = params$maa[1],
a2 = params$maa[2], n.rep1=nreps[1], n.rep2=nreps[2], p.cut = 0.8)
```

FDRplotTailPP *Plot of FDR for tail posterior probability*

Description

Plots smoothed FDR vs tail posterior probability or vs the number of differentially expressed (DE) genes

Usage

```
FDRplotTailPP(tpp.res, nmax = sum(! is.na(tpp.res$FDR)), plot.TP = F)
```

Arguments

tpp.res	output of TailPP
nmax	maximum size of the list of DE genes
plot.TP	logical. If TRUE FDR is plotted, otherwise the number of false positives is plotted vs the number of differentially expressed genes

Author(s)

Natalia Bochkina

References

Bochkina N., Richardson S. (2007) Tail posterior probability for inference in pairwise and multiclass gene expression data. Biometrics.

See Also

TailPP, histTailPP, EstimatePi0

Examples

```
 data(ybar, ss)
 nreps <- c(8,8)

## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
 outdir <- BGmix(ybar, ss, nreps, jstar=-1, nburn=0, niter=100, nthin=1)

 params <- ccParams(outdir)
 res <-  ccTrace(outdir)

 tpp.res <- TailPP(res, nreps, params, plots  = FALSE)
 FDRplotTailPP(tpp.res, plot.TP = TRUE)
```

TailPP *Tail posterior probability for BGmix output.*

Description

For differential expression models with unstructured priors (no mixture prior), calculates tail posterior probabality and FDR, and plots a histogram. Uses whole posterior distributions of likelihood parameters (found by 'ccTrace') and posterior means of hyperparameters (found by 'ccParams').

Usage

```
TailPP(res, nreps, params, paired=F, alpha=0.05, N = 5000, prec=0.05, p.cut = 0.
```

Arguments

res	list object output from 'ccTrace'
nreps	vector length 2 containing the number of replicates in each condition
params	list object output from 'ccParams'
paired	logical. TRUE for paired design, FALSE for unpaired.
alpha	parameter of the tail posterior probability (1-alpha/2 quantile)

N	simulation size for tail posterior probability under H0
prec	parameter used when estimating CDF of tail posterior probability under H0
p.cut	calculate FDR only for cutoffs on tail posterior probability > p.cut
plots	logical. if TRUE, makes plots of the histogram of tail posterior probability with the null density and of FDR
plot.pi0	logical. if TRUE, diagnostic plot of the estimated pi0 at different locations and the median estimate

Value

tpp	vector of tail posterior probabilities with parameter alpha, one per gene
FDR	(smoothed) estimate of FDR for all (distinct) cutoffs > p.cut
pi0	estimated proportion of observations under the null
pp0	simulations under the null

Author(s)

Natalia Bochkina

References

Bochkina N., Richardson S. (2007) Tail posterior probability for inference in pairwise and multiclass gene expression data. Biometrics. http://www.blackwell-synergy.com/doi/abs/10.1111/j.1541-0420.2006.00807.x

See Also

FDRplotTailPP,histTailPP,EstimatePi0

Examples

```
 data(ybar, ss)
 nreps <- c(8,8)

## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
 outdir <- BGmix(ybar, ss, nreps, jstar=-1, nburn=0, niter=100, nthin=1)

 params <- ccParams(outdir)
 res <-  ccTrace(outdir)

 tpp.res <- TailPP(res, nreps, params, plots  = FALSE)
 histTailPP(tpp.res)
 FDRplotTailPP(tpp.res, plot.TP = TRUE)
```

`calcFDR` *Estimate the FDR (false discovery rate) and related quantities for*

Description

Given a threshold on the posterior probabilities, genes are declared as null or differentially expressed. For any given threshold, the FDR (false discovery rate) and FNR (false non-discovery rate) can be estimated using the posterior probabilities. Estimated numbers of false positives and false negatives are also output.

Usage

```
calcFDR(res, pcut = seq(0.01,0.5,0.01), true.z = NULL, q.print = F)
```

Arguments

`res`	list object output from ccParams (this includes the posterior classification probabilities)
`pcut`	scalar or vector of thresholds for which to estimate FDR etc.
`true.z`	vector of true classifications (if known, eg. for simulated data)
`q.print`	Print FDR etc. when pcut is a vector?

Details

If the true classification is known, it can be given as true.z, and the true FDR etc. for the threshold probability can be calculated.

Value

`fdr.est, fnr.est`
: scalars or vectors of estimated FDR, FNR

`fp.est, fn.est`
: scalars or vectors of estimated no. false positives, no. false negatives

`fdr.true, fnr.true`
: scalars or vectors of true FDR, FNR

`fp.true, fn.true`
: scalars or vectors of true no. false positives, no. false negatives

`npos, nneg`
: scalars or vectors of no. declared positives, no. declared negatives

`prob.class`
: posterior classification probabilites (from the 'res' object input to this function)

`true.z`
: argument to function is output

`pcut`
: argument to function is output

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
params <- ccParams(outdir)
fdr <- calcFDR(params)
```

`ccParams` *Read posterior means and classification probabilities from BGmix*

Description

Reads output files containing posterior means from BGmix AND reads posterior probabilities of each gene being classified in the null mixture component.

Usage

```
ccParams(filedir, q.beta = T, q.sig = T, q.z = T, quiet = T)
```

Arguments

`filedir`	character. The name of the output directory created by BGmix.
`q.beta`	logical. Read beta values?
`q.sig`	logical. Read gene variance parameters?
`q.z`	logical. Read z values?
`quiet`	logical. Parameter passed to 'scan'. (If false, 'scan' prints details of number of items read in.)

Value

`mbeta`	matrix no. genes x no. effects. Posterior means of gene effect parameters (usually gene means and log fold changes).
`msig2`	matrix no. genes x no. variances. Posterior means of gene variances.
`mbb`	vector of hyperparameters (b) for gene variances (posterior means).
`maa`	vector of hyperparameters (a) for gene variances (posterior means).
`mtau`	matrix no. genes x no. conditions. Posterior means of gene precisions.
`mwtc`	vector of posterior mean mixture weights
`mzg`	vector of posterior mean allocation for each gene
`meta`	vector of mixture parameters (eta)
`mlambda`	vector of mixture parameters (lambda)
`pc`	matrix no. genes x no. mixture components. Posterior probability for each gene of being classified into each mixture component.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
params <- ccParams(outdir)
```

---

ccPred *Read predictive quantities output from BGmix.*

Description

Reads predictive p-values from files output from BGmix. Also (optionally) reads posterior predictive distributions of data.

Usage

```
ccPred(filedir, q.partial = T, q.trace = F, quiet = T)
```

Arguments

filedir	character. The name of the output directory created by BGmix.
q.partial	logical. Read partial predictive p-values?
q.trace	logical. Read posterior predictive distributions of data?
quiet	logical. Parameter passed to 'scan'. (If false, 'scan' prints details of number of items read in.)

Value

pval.ss.post	matrices no. genes x no. conditions. Posterior predictive p-values for sum of squares for each gene in each condition.
pval.ss.mix	matrices no. genes x no. conditions. Mixed predictive p-values for sum of squares for each gene in each condition.
pval.ss.part	matrices no. genes x no. conditions. Partial predictive p-values for sum of squares for each gene in each condition.
pval.ybar.post	matrices no. genes x no. mixture components. Posterior predictive p-values for ybar for each gene in each mixture component.
pval.ybar.mix2	matrices no. genes x no. mixture components. Mixed predictive p-values for ybar for each gene in each mixture component.
pval.ybar.part	matrices no. genes x no. mixture components. Partial predictive p-values for ybar for each gene in each mixture component.
ybar.pred1	Posterior predictive distribution of ybar.
ybar.pred3	Mixed predictive distribution of ybar.
ss.pred1	Posterior predictive distribution of sums of squares.
ss.pred2	Mixed predictive distribution of sums of squares.

Note

Additional output: pval.ybar.mix1 and pval.ybar.mix3 are alternative versions of mixed predictive p-values (currently not used). Also, ybar.pred2 and ybar.pred4 are the corresponding alternative mixed predictive distributions for ybar.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
pred <- ccPred(outdir)
```

---

ccSummary *Read summary of BGmix output*

Description

Reads the summary.txt file output by BGmix, containing information about data sets used and model options. This function is called by ccParams, ccTrace and ccPred, therefore users will not in general need to call it directly.

Usage

```
ccSummary(filedir)
```

Arguments

filedir	character. The name of the output directory created by BGmix.

Value

A list of scalar values, as follows:

ngenes,nconds,neffects,ncomps,ntau
: nos. genes, conditions, effects, mixture components, gene variances

jstar
: label of effect with mixture prior (labels start at 0)

move.choice.bz,move.choice.cut,move.choice.aa, move.choice.eta,move.choice.lam,m
: model choice options (see BGmix help for details

lambda.up.init,lambda.down.init, eta.up.init,eta.down.init
: inital values for eta and lambda (parameters of mixture components)

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
summ <- ccSummary(outdir)
```

---

ccTrace *Read trace files from BGmix*

Description

Reads output files containing whole posterior distributions from BGmix. Also calls 'ccSummary', and outputs model options.

Usage

```
ccTrace(filedir, q.beta = T, q.sig = T, q.z = T, quiet = T)
```

Arguments

filedir	character. The name of the output directory created by BGmix.
q.beta	logical. Read beta values?
q.sig	logical. Read gene variances?
q.z	logical. Read z values?
quiet	logical. Parameter passed to 'scan'. (If false, 'scan' prints details of number of items read in.)

Value

summ	list object output by 'ccSummary'
eta	matrix (no. components -1) x no. MCMC samples. Posterior of mixture component parameters (eta).
lambda	matrix (no. components -1) x no. MCMC samples. Posterior of mixture component parameters (lambda).
aa	matrix no. MCMC samples x no. variances. Posterior of variance hyperparameters (a).
bb	matrix no. MCMC samples x no. variances. Posterior of variance hyperparameters (b).
wtc	matrix no. MCMC samples x no. mixture components. Posterior of mixture weights.
beta	matrix no. effects x no. genes x no. MCMC samples. Posterior of gene effects.
sig2	matrix no. variances x no. genes x no. MCMC samples. Posterior of gene variances.
zg	matrix no. MCMC samples x no. genes. Posterior of gene allocations.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
tr <- ccTrace(outdir)
```

---

histTailPP *Histogram plot for tail posterior probability*

Description

Plots a histogram of tail posterior probability with its density under the null hypothesis

Usage

```
histTailPP(tpp.res, bw=0.05, xlim=c(0,1),nc=10)
```

Arguments

tpp.res	output of TailPP
bw	bandwidth for kernel estimate of the null density
xlim	limits on the x axis
nc	number of bins of the histogram

Author(s)

Natalia Bochkina

References

Bochkina N., Richardson S. (2007) Tail posterior probability for inference in pairwise and multiclass gene expression data. Biometrics.

See Also

TailPP, FDRplotTailPP, EstimatePi0

Examples

```
data(ybar, ss)
 nreps <- c(8,8)

## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
 outdir <- BGmix(ybar, ss, nreps, jstar=-1, nburn=0, niter=100, nthin=1)
```

```
params <- ccParams(outdir)
res <-  ccTrace(outdir)

tpp.res <- TailPP(res, nreps, params, plots  = FALSE)
histTailPP(tpp.res, bw=0.04, xlim=c(0,1), nc=10)
```

`plotBasic` *Basic plots of BGmix parameters and data.*

Description

Plots gene effects and variances versus their corresponding data sufficient statistics (to show the effect of smoothing and shrinkage). Also plots "volcano plots": posterior probabilities of being classified in each mixture component versus the log fold change parameters.

Usage

```
plotBasic(res, ybar, ss, q.mean = T, q.diff = T, q.sig = T, q.volcano = T)
```

Arguments

`res`	list object output from 'ccParams'
`ybar`	ybar data (see `BGmix` help for details)
`ss`	ss data (see `BGmix` help for details)
`q.mean`	logical. Include mean plot?
`q.diff`	logical. Include log fold change plot?
`q.sig`	logical. Include variance plot?
`q.volcano`	logical. Include volcana plot (posterior classification v. fold change)?

Details

Note this plotting function is designed for model output from the unpaired differential expression design.

Value

No value is returned to R. Results from BGmix model are output to files.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
params <- ccParams(outdir)
plotBasic(params,ybar,ss)
```

plotCompare *Scatter plot with equal axes.*

Description

Plots a scatter plot of two variables with equal scales for the axes.

Usage

```
plotCompare(var1, var2, limi = 0, xlab = substitute(var1), ylab = substitute(var
```

Arguments

var1	data to plot (x co-ordinate)
var2	data to plot (y co-ordinate)
limi	limits of axes. If not specified, axes limits are determined from input data.
xlab	x-axis label
ylab	y-axis label
log	specifies if axes are on the log scale (as argument to 'par')
title	title of plot
...	other parameters input to plot

Value

Outputs the limits used in the plot (the input 'limi' argument if specified).

Author(s)

Alex Lewin

Examples

```
x <- runif(100)
y <- rbeta(100,0.5,0.5)
plotCompare(x,y)
```

plotFDR *Plot estimated FDR etc. for BGmix output.*

Description

Given a threshold on the posterior probabilities, genes are declared as null or differentially expressed. For any given threshold, the FDR (false discovery rate) and FNR (false non-discovery rate) can be estimated using the posterior probabilities. This function plots these quantities twice, once versus the threshold probabilities, and once versus the number of declared positives.

Usage

```
plotFDR(res, ylim = NULL, q.plotfnr = F, q.plotpcut = T, q.plotnpos = T, ...)
```

Arguments

res	list object output from 'calcFDR'
ylim	optional argument specifying limit for y-axis
q.plotfnr	Include FNR in plots?
q.plotpcut	Include the plot of error rates v. threshold on posterior probabilites?
q.plotnpos	Include the plot of error rates v. no. positives.
...	arguments passed to 'plot'

Value

No value is returned to R. Results from BGmix model are output to files.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
params <- ccParams(outdir)
fdr <- calcFDR(params)
par(mfrow=c(1,2))
plotFDR(fdr)
```

---

plotMixDensity *Plot predictive density of data.*

Description

Plot predictive density of data superimposed on histograms of observed data. Separate plots for ybar and sums of squares.

Usage

```
plotMixDensity(res, predres, ybar, ss)
```

Arguments

res	list object output from 'ccParams'
predres	list object output from 'ccPred' (need q.trace=T in 'ccPred')
ybar	ybar data (see BGmix help for details)
ss	ss data (see BGmix help for details)

Details

Note that this function is written for the unpaired differential expression design.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar,ss,c(8,8),niter=100,nburn=0,nthin=1,trace.pred=1)
pred <- ccPred(outdir,q.trace=TRUE)
params <- ccParams(outdir)
plotMixDensity(params,pred,ybar,ss)
```

---

plotPredChecks *Plots of predictive checks for mixture prior.*

Description

Histograms and q-q plots of predictive p-values for the mixture prior. Separate plots are given for each mixture component, using only genes with high posterior probability of being classified into the relevant component.

Usage

```
plotPredChecks(pvals, pc, probz = 0.8, label = "", breaks = 20)
```

Arguments

pvals	matrix of predictive p-values output by 'ccPred' (NB, not the whole list object, just the matrix of p-values)
pc	matrix of posterior classification probabilities output by 'ccParams' (NB, not the whole list object, just the matrix of probabilities)
probz	threshold on posterior probabilities for including genes in each mixture component plot
label	title used on histograms
breaks	argument input to histogram

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
params <- ccParams(outdir)
pred <- ccPred(outdir)
plotPredChecks(pred$pval.ybar.mix2,params$pc,probz=0.5)
```

`plotTrace` *Trace plots for BGmix output.*

Description

Trace plots are plotted for all scalar parameters. Optionally, traces are plotted for parameters indexed by genes, but for selected genes only.

Usage

```
plotTrace(res, q.beta = T, q.sig = T, q.z = T, ind.genes = (1:3))
```

Arguments

`res`	list object output from 'ccTrace'
`q.beta`	logical. Plot trace of beta (gene effect) parameters?
`q.sig`	logical. Plot trace of gene variances?
`q.z`	logical. Plot trace of gene allocation parameters?
`ind.genes`	indices of genes for which to plot gene parameters.

Author(s)

Alex Lewin

Examples

```
## Note this is a very short MCMC run!
## For good analysis need proper burn-in period.
data(ybar,ss)
outdir <- BGmix(ybar, ss, c(8,8), nburn=0, niter=100, nthin=1)
tr <- ccTrace(outdir)
plotTrace(tr)
plotTrace(tr,q.beta=TRUE,q.sig=FALSE,q.z=FALSE,ind.genes=1)
plotTrace(tr,q.beta=FALSE,q.sig=FALSE,q.z=TRUE,ind.genes=sample(1:1000,5))
```

`readBGX` *Reads output from BGX package, for input to BGmix.*

Description

Reads posterior mean parameters from BGX, and outputs objects suitable for input to BGmix.

Usage

```
readBGX(path)
```

Arguments

`path`	directory containing BGX output

Value

ybar	ybar object (see BGmix help for details)
ss	ss object (see BGmix help for details)
...	

Author(s)

Ernest Turro

Simulated gene expression data

Sample variance of log gene expression under two conditions

Description

Simulated gene expression data. 2500 genes under 2 experimental conditions, with 8 replicate arrays for each condition. The data is presented as mean and sum of squares of the log gene expression, in each condition. ss is the matrix containing the sample variances in each condition.

Usage

```
data(ss)
```

Format

matrix no. genes x no. experimental conditions

Simulated example data

Mean log gene expression under two conditions

Description

Simulated gene expression data. 2500 genes under 2 experimental conditions, with 8 replicate arrays for each condition. The data is presented as mean and sum of squares of the log gene expression, in each condition. ybar is the matrix containing the means in each condition.

Usage

```
data(ybar)
```

Format

matrix no. genes x no. experimental conditions

Index