Package: DIRECT (via r-universe)

October 9, 2024

Type Package

Title Bayesian Clustering of Multivariate Data Under the Dirichlet-Process Prior

Version 1.1.0

Date 2023-08-30

Author Audrey Qiuyan Fu, Steven Russell, Sarah J. Bray and Simon Tavare

Maintainer Audrey Q. Fu <audreyqyfu@gmail.com>

Description A Bayesian clustering method for replicated time series or replicated measurements from multiple experimental conditions, e.g., time-course gene expression data. It estimates the number of clusters directly from the data using a Dirichlet-process prior. See Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361. <doi:10.1214/13-AOAS650>.

License GPL (>= 2)

LazyLoad yes

LazyData yes

NeedsCompilation yes

Date/Publication 2023-09-07 22:10:02 UTC

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

RemoteUrl https://github.com/cran/DIRECT

RemoteRef HEAD

RemoteSha de9a9057fc960b2c5347ead9c2012b7daf17df38

Contents

DIRECT-package 2
DIRECT 3

Dirichlet 9
DPMCMC 10
MVNorm 14
outputData 15
plotClustersMean 16
plotClustersPCA 17
plotClustersSD 18
plotSimulation 19
relabel 20
resampleClusterProb 21
simuDataREM 23
summaryDIRECT 25
tc.data 27

Index **29**

`DIRECT-package` *Bayesian Clustering of Multivariate Data with the Dirichlet-Process Prior*

Description

This package implements the Bayesian clustering method in Fu et al. (2013). It also contains a simulation function to generate data under the random-effects mixture model presented in this paper, as well as summary and plotting functions to process MCMC samples and display the clustering results. Replicated time-course microarray gene expression data analyzed in this paper are in `tc.data`.

Details

This package three sets of functions.

- Functions `DIRECT` and others for clustering data. They estimate the number of clusters and infers the partition for multivariate data, e.g., replicated time-course microarray gene expression data. The clustering method involves a random-effects mixture model that decomposes the total variability in the data into within-cluster variability, variability across experimental conditions (e.g., time points), and variability in replicates (i.e., residual variability). The clustering method uses a Dirichlet-process prior to induce a distribution on the number of clusters as well as clustering. It uses Metropolis-Hastings Markov chain Monte Carlo for parameter estimation. To estimate the posterior allocation probability matrix while dealing with the label-switching problem, there is a two-step posterior inference procedure involving resampling and relabeling.
- Functions for processing MCMC samples and plotting the clustering results.
- Functions for simulating data under the random-effects mixture model.

See `DIRECT` for details on using the function for clustering.

See `summaryDIRECT`, which points to other related plotting functions, for details on how to process MCMC samples and display clustering results.

See simuDataREM, which points to other related functions, for simulating data under the random-effects mixture model.

Author(s)

Audrey Qiuyan Fu

Maintainer: Audrey Q. Fu <audreyqyfu@gmail.com>

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

DIRECT for the clustering method.

summaryDIRECT for processing MCMC estimates for clustering.

simuDataREM for simulating data under the mixture random-effects model.

Examples

```
## See examples in DIRECT and simuDataREM.
```

DIRECT *Bayesian Clustering with the Dirichlet-Process Prior*

Description

A Bayesian clustering method for multivariate data, e.g., replicated time-course microarray gene expression data. This method uses a mixture random-effects model that decomposes the total variability in the data into within-cluster variability, variability across experimental conditions (e.g., time points), and variability in replicates (i.e., residual variability). It also uses a Dirichlet-process prior to induce a distribution on the number of clusters as well as clustering. Metropolis-Hastings Markov chain Monte Carlo procedures are used for parameter estimation.

Usage

```
DIRECT (data, data.name = "Output",
        SKIP = 0, nTime, times = 1:nTime,
        c.curr,
        uWICluster = 1, uTSampling = 1, uResidual = 1,
        meanVec = rep(0, nTime), meanMT1 = 0, sdMT1 = 0.2,
        meanMTProc = 0, sdMTProc = 0.5, uSDT1 = 0.2, uSDProc = 1,
        shapeBetaProc = 0.5, rateBetaProc = 0.5,
        PAR.INIT = TRUE,
        sdWICluster.curr = 0.5, sdTSampling.curr = 0.5,
        sdResidual.curr = 0.5, alpha.curr = 0.01,
```

```
alpha.prior.shape = 0.01, alpha.prior.rate = 1,
WICluster.prop.sd = 0.2, TSampling.prop.sd = 0.2,
Residual.prop.sd = 0.2, alpha.prop.sd = 0.2,
nIter, burn.in, step.size, nRepeat = 1, nResample,
seed.value,
RNORM.METHOD = c("chol", "eigen", "svd"),
SAMPLE.C = c("FRBT", "Neal"),
PRIOR.MODEL = c("none", "OU", "BM", "BMdrift"),
ALPHA.METHOD = c("Gibbs", "MH"),
RELABEL.THRESHOLD = 0.01,
OUTPUT.CLUST.SIZE = FALSE, PRINT = FALSE)
```

Arguments

data	An $N \times JR$ matrix of continuous values, or a data frame containing such a matrix. N is the number if items, J the number of time points (or experimental conditions) and R the number of replicates. Each row contains values for Replicates 1 through R under Condition 1, values for Replicates 1 through R under Condition 2, and so on.
data.name	A character string used as the prefix of output files.
SKIP	Number of columns in data to be skipped when processing the data.
nTime	Number of time points (or experimental conditions).
times	An integer vector of length nTime, indicating times (or experimental conditions).
c.curr	An integer vector of length N, indicating initial cluster assignments for items 1 through N.
uWICluster	Upper bound of the uniform prior assigned to the standard deviation of within-cluster variability. The lower bound of the uniform prior is 0.
uTSampling	Upper bound of the uniform prior assigned to the standard deviation of variability due to sampling across time points (or experimental conditions). The lower bound of the uniform prior is 0.
uResidual	Upper bound of the uniform prior assigned to the standard deviation of residual variability (i.e., variability across replicates). The lower bound of the uniform prior is 0.
meanVec	Prior mean vector of length nTime. Required if PRIOR.MODEL="none".
meanMT1	Prior mean (scalar) of the mean at the first time point. Required if PRIOR.MODEL is one of the stochastic processes ("OU", "BM" and "BMdrift").
sdMT1	A positive scalar. Prior standard deviation (scalar) of the mean at the first time point. Required if PRIOR.MODEL is one of the stochastic processes ("OU", "BM" and "BMdrift").
meanMTProc	Prior mean (scalar) of the mean across time points. Required if PRIOR.MODEL is one of the stochastic processes ("OU", "BM" and "BMdrift"). Set to 0 if PRIOR.MODEL="BM".
sdMTProc	A positive scalar. Prior standard deviation (scalar) of the mean across time points. Required if PRIOR.MODEL is one of the stochastic processes ("OU", "BM" and "BMdrift").

uSDT1 The upper bound of the uniform prior assigned to the standard deviation at the first time point. The lower bound of the uniform prior is 0. Required if PRIOR.MODEL is one of the stochastic processes ("OU", "BM" and "BMdrift").

uSDProc The upper bound of the uniform prior assigned to the standard deviation across time points. The lower bound of the uniform prior is 0. Required if PRIOR.MODEL is one of the stochastic processes ("OU", "BM" and "BMdrift").

shapeBetaProc A positive scalar. The shape parameter in the beta prior for the mean-reverting rate in an Ornstein-Uhlenbeck process. Required if PRIOR.MODEL="OU".

rateBetaProc A positive scalar. The rate parameter in the beta prior for the mean-reverting rate in an Ornstein-Uhlenbeck process. Required if PRIOR.MODEL="OU".

PAR.INIT Logical value. Generate initial values for the standard deviations of the three types of variability if TRUE. Use the input values otherwise.

sdWICluster.curr
A positive scalar. Initial value of the standard deviation of the within-cluster variability. Ignored if PAR.INIT=TRUE.

sdTSampling.curr
A positive scalar. Initial value of the standard deviation of the variability across time points. Ignored if PAR.INIT=TRUE.

sdResidual.curr
A positive scalar. Initial value of the standard deviation of the residual variability (i.e., variability across replicates). Ignored if PAR.INIT=TRUE.

alpha.curr A positive scalar. Initial value of α, the concentration parameter of the Dirichlet-process prior.

alpha.prior.shape
A positive scalar. The shape parameter in the beta prior for α, the concentration parameter of the Dirichlet-process prior.

alpha.prior.rate
A positive scalar. The rate parameter in the beta prior for α, the concentration parameter of the Dirichlet-process prior.

WICluster.prop.sd
A positive scalar. The standard deviation in the proposal distribution (normal) for the standard deviation of the within-cluster variability.

TSampling.prop.sd
A positive scalar. The standard deviation in the proposal distribution (normal) for the standard deviation of the variability across time points.

Residual.prop.sd
A positive scalar. The standard deviation in the proposal distribution (normal) for the standard deviation of the residual variability (i.e., variability across replicates).

alpha.prop.sd A positive scalar. The standard deviation in the proposal distribution (normal) for α, the concentration parameter of the Dirichlet-process prior. Ignored if ALPHA.METHOD="Gibbs".

nIter The number of MCMC iterations.

burn.in A value in (0,1) indicating the percentage of the MCMC iterations to be used as burn-in and be discarded in posterior inference.

`step.size` An integer indicating the number of MCMC iterations to be skipped between two recorded MCMC samples.

`nRepeat` An integer ≥ 1 indicating the number of times to update the cluster memberships for all items. Useful only when `SAMPLE.C="Neal"`.

`nResample` An integer ≥ 1 indicating the number of resamples to draw to estimate the posterior mixing proportions.

`seed.value` A positive value used in random number generation.

`RNORM.METHOD` Method to compute the determinant of the covariance matrix in the calculation of the multivariate normal density. Required. Method choices are: "chol" for Choleski decomposition, "eigen" for eigenvalue decomposition, and "svd" for singular value decomposition.

`SAMPLE.C` Method to update cluster memberships. Required. Method choices are: "FRBT" for the Metropolis-Hastings sampler based on a discrete uniform proposal distribution developed in Fu, Russell, Bray and Tavare, and "Neal" for the Metropolis-Hastings sampler developed in Neal (2000).

`PRIOR.MODEL` Model to generate realizations of the mean vector of a mixture component. Required. Choices are: "none" for not assuming a stochastic process and using a zero vector, "OU" for an Ornstein-Uhlenbeck process (a.k.a. the mean-reverting process), "BM" for a Brown motion (without drift), and "BMdrift" for a Brownian motion with drift.

`ALPHA.METHOD` Method to update α, the concentration parameter of the Dirichlet-process prior. Required. Choices are: "Gibbs" for a Gibbs sampler developed in Escobar and West (1995), and "MH" for a Metropolis-Hastings sampler.

`RELABEL.THRESHOLD`
A positive scalar. Used to determine whether the optimization in the relabeling algorithm has converged.

`OUTPUT.CLUST.SIZE`
If TRUE, output cluster sizes in MCMC iterations into an external file *_mcmc_size.out.

`PRINT` If TRUE, print intermediate values during an MCMC run onto the screen. Used for debugging with small data sets.

Details

DIRECT is a mixture model-based clustering method. It consists of two major steps:

1. MCMC sampling. DIRECT generates MCMC samples of assignments to mixture components (number of components implicitly generated; written into external file *_mcmc_cs.out) and component-specific parameters (written into external file *_mcmc_pars.out), which include mean vectors and standard deviations of three types of variability.
2. Posterior inference, which further consists of two steps:
   (a) Resampling: DIRECT estimates posterior allocation probability matrix (written into external file *_mcmc_probs.out).
   (b) Relabeling: DIRECT deals with label-switching by estimating optimal labels of mixture components (written into external file *_mcmc_perms.out), implementing Algorithm 2 in Stephens (2000).

The arguments required to set up a DIRECT run can be divided into five categories:

1. Data-related, such as `data`, `times` and so on.
2. Initial values of parameters, including `c.curr`, `sdWICluster.curr`, `sdTSampling.curr`, `sdResidual.curr` and `alpha.curr`.
3. Values used to specify prior distributions, such as `uWICluster`, `meanMT1`, `rateBetaProc`, `alpha.prior.shape` and so on.
4. Standard deviation used in the proposal distributions for parameters of interest. A normal distribution whose mean is the current value and whose standard deviation is user-specified is used as the proposal. Reflection is used if the proposal is outside the range (e.g., (0,1)) for the parameter.
5. Miscellaneous arguments for MCMC configuration, for model choices and for output choices.

The user may set up multiple runs with different initial values or values in the prior distributions, and compare the clustering results to check whether the MCMC run has mixed well and whether the inference is sensitive to initial values or priors. If the data are informative enough, initial values and priors should lead to consistent clustering results.

Value

At least four files are generated during a DIRECT run and placed under the current working directory:

1. *_mcmc_cs.out: Generated from MCMC sampling. Each row contains an MCMC sample of assignments of items to mixture components, or cluster memberships if a component is defined as a cluster, as well as α, the concentration parameter in the Dirichlet-process prior.
2. *_mcmc_pars.out: Generated from MCMC sampling. Each row contains an MCMC sample of parameters specific to a mixture component. Multiple rows may come from the same MCMC iteration.
3. *_mcmc_probs.out: Generated from resampling in posterior inference. File contains a matrix of $HN \times K$, which is H posterior allocation probability matrices stacked up, each matrix of $N \times K$, where H is the number of recorded MCMC samples, N the number of items and K the inferred number of mixture components.
4. *_mcmc_perms.out: Generated from relabeling in posterior inference. Each row contains an inferred permutation (relabel) of labels of mixture components.

If argument `OUTPUT.CLUST.SIZE=TRUE`, the fifth file *_mcmc_size.out is also generated, which contains the cluster sizes of each recorded MCMC sample.

Note

`DIRECT` calls the following functions adapted or directly taken from other R packages: `dMVNorm`, `rMVNorm` and `rDirichlet`. See documentation of each function for more information.

Author(s)

Audrey Q. Fu

References

Escobar, M. D. and West, M. (1995) Bayesian density estimation and inference using mixtures. Journal of the American Statistical Association, 90: 577-588.

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

Stephens, M. (2000) Dealing with label switching in mixture models. Journal of the Royal Statistical Society, Series B, 62: 795-809.

Neal, R. M. (2000) Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics, 9: 249-265.

See Also

DPMCMC for the MCMC sampler under the Dirichlet-process prior.

resampleClusterProb for resampling of posterior allocation probability matrix in posterior inference.

relabel for relabeling in posterior inference.

summaryDIRECT for processing MCMC estimates for clustering.

simuDataREM for simulating data under the mixture random-effects model.

Examples

```
## Not run:
# Load replicated time-course gene expression data
# Use only first 50 genes for test run
data (tc.data)
data = tc.data[1:50,]
times = c(0,5,10,15,20,25,30,35,40,50,60,70,80,90,100,110,120,150)
nGene = nrow (data)
nTime=length (times)
SKIP = 2

# Initial values and MCMC specs
c.curr = rep (1, nGene)    # start with a single cluster
alpha.curr = 0.01

alpha.prior.shape = 1/nGene
alpha.prior.rate = 1

SAMPLE.C.METHOD="FRBT"    # method for sampling cluster memberships
PRIOR.MODEL = "OU"       # prior model for generating mean vector
ALPHA.METHOD = "MH"      # method for sampling concentration parameter
RELABEL.THRESHOLD=0.01   # stopping criterion used in relabeling algorithm

nIter=10
burn.in=0
step.size=1
nResample=2
seed.value = 12
```

```
data.name="tmp"          # prefix of output files

# Run DIRECT
# This is a short run that takes less than a minute
# All output files will be under current working directory
DIRECT (data=data, data.name=data.name, SKIP=SKIP, nTime=nTime, times=times,
    c.curr=c.curr, PAR.INIT=TRUE, alpha.curr=alpha.curr,
    alpha.prior.shape=alpha.prior.shape,
    alpha.prior.rate=alpha.prior.rate,
    nIter=nIter, burn.in=burn.in, step.size=step.size,
    nResample=nResample, seed.value=seed.value,
    RNORM.METHOD="svd", SAMPLE.C=SAMPLE.C.METHOD,
    PRIOR.MODEL=PRIOR.MODEL, ALPHA.METHOD=ALPHA.METHOD,
    RELABEL.THRESHOLD=RELABEL.THRESHOLD)

# Process MCMC samples from DIRECT
data.name="tmp"          # prefix of output files
tmp.summary = summaryDIRECT (data.name)

# Plot clustering results
#
# If the plots do not display well
# use pdf() to generate the plots in an external pdf
#
# Clustered mean profiles
plotClustersMean (data, tmp.summary, SKIP=2, times=times)
#
# To use pdf(), run the following lines in R
# > pdf ("plot_means.pdf")
# > plotClustersMean (data, tmp.summary, SKIP=2, times=times)
# > dev.off()
#
par (mfrow=c(1,1))
# Posterior estimates of standard deviations
# of three types of variability in each cluster
plotClustersSD (tmp.summary, nTime=18)
# PCA plot of the posterior allocation probability matrix
plotClustersPCA (data$GeneName, tmp.summary)

## End(Not run)
```

`Dirichlet` *The Dirichlet Distribution*

Description

Functions to compute the density of a Dirichlet distribution and to generate random realizations from such a distribution.

Usage

```
dDirichlet (x, alpha, log=FALSE)
rDirichlet (n, alpha)
```

Arguments

alpha	Shape parameter vector.
x	Vector of the same length as alpha.
n	Number of realizations (vectors) to generate.
log	Logical value. TRUE if computing the log density. Default is FALSE.

Value

rDirichlet returns a vector of the same length as alpha if n=1, or a matrix with each row being an independent realization otherwise.

Author(s)

Audrey Q. Fu coded dDirichlet.

The code for rDirichlet is taken from a similar function in R package gregmisc by Gregory R. Warnes. His code was based on code posted by Ben Bolker to R-News on 15 Dec 2000. See documentation in gregmisc for further information.

Examples

```
x <- rDirichlet (5, rep (0.5, 3))
dDirichlet (x[1, ], rep (0.5, 3))
```

DPMCMC *Dirichlet Process-Based Markov Chain Monte Carlo (MCMC) Sampler for Mixture Model-Based Clustering*

Description

The MCMC sampler for DIRECT. In each MCMC iteration, the function updates cluster memberships for all items, allowing for changes in the number of clusters (mixture components). This update implements a Metropolis-Hastings (MH) sampler developed in Fu et al. (2013), and an MH sampler developed in Neal (2000). It also updates parameters specific to each mixture components via MH sampling. Parameters of interest include the mean vector and standard deviations of the three types of variability. Additionally, it updates α, the concentration parameter in the Dirichlet-process prior, allowing for Gibbs (Escobar and West, 1995) and MH sampling.

Usage

```
DPMCMC(file.mcmc.cs, file.mcmc.pars, file.mcmc.probs, file.size,
       data, SKIP, nTime, times, c.curr, par.prior,
       PAR.INIT = FALSE,
       sdWICluster.curr = 0.5, sdTSampling.curr = 0.5,
       sdResidual.curr = 0.5, alpha.curr,
       alpha.prior.shape, alpha.prior.rate, sd.prop,
       nIter, burn.in, step.size, nRepeat = 1, nResample, seed.value,
       RNORM.METHOD = c("chol", "eigen", "svd"),
       SAMPLE.C = c("FRBT", "Neal"),
       PRIOR.MODEL = c("none", "OU", "BM", "BMdrift"),
       ALPHA.METHOD = c("Gibbs", "MH"),
       OUTPUT.CLUST.SIZE = FALSE, PRINT = FALSE)
```

Arguments

`file.mcmc.cs`	A character string in quotation marks indicating the output filename for cluster memberships and α.
`file.mcmc.pars`	A character string in quotation marks indicating the output filename for MCMC samples of parameters specific to mixture components.
`file.mcmc.probs`	A character string in quotation marks indicating the output filename for posterior allocation probability matrices from the resampling step.
`file.size`	A character string in quotation marks indicating the output filename for cluster sizes.
`data`	An $N \times JR$ matrix of continuous values, or a data frame containing such a matrix. N is the number if items, J the number of time points (or experimental conditions) and R the number of replicates. Each row contains values for Replicates 1 through R under Condition 1, values for Replicates 1 through R under Condition 2, and so on.
`SKIP`	Number of columns in `data` to be skipped when processing the data.
`nTime`	Number of time points (or experimental conditions).
`times`	An integer vector of length `nTime`, indicating times (or experimental conditions).
`c.curr`	An integer vector of length N, indicating initial cluster assignments for items 1 through N.
`par.prior`	A list that contains parameters of the prior distributions. It has the following format: `par.prior = list (uWICluster=???, uTSampling=???, uResidual=???, mean=???, meanMT1=???, sdMT1=???, meanMTProc=???, sdMTProc=???, uSDT1=???, uSDProc=???, shapeBetaProc=???, rateBetaProc=???)`. See `DIRECT` for possible values of the list components.
`PAR.INIT`	Logical value. Generate initial values for the standard deviations of the three types of variability if TRUE. Use the input values otherwise.
`sdWICluster.curr`	A positive scalar. Initial value of the standard deviation of the within-cluster variability. Ignored if `PAR.INIT=TRUE`.

`sdTSampling.curr`
: A positive scalar. Initial value of the standard deviation of the variability across time points. Ignored if `PAR.INIT=TRUE`.

`sdResidual.curr`
: A positive scalar. Initial value of the standard deviation of the residual variability (i.e., variability across replicates). Ignored if `PAR.INIT=TRUE`.

`alpha.curr`
: A positive scalar. Initial value of α, the concentration parameter of the Dirichlet-process prior.

`alpha.prior.shape`
: A positive scalar. The shape parameter in the beta prior for α, the concentration parameter of the Dirichlet-process prior.

`alpha.prior.rate`
: A positive scalar. The rate parameter in the beta prior for α, the concentration parameter of the Dirichlet-process prior.

`sd.prop`
: A list that contains standard deviations in the proposal distributions for some key parameters. It has the following format: `sd.prop=list(WICluster=???, TSampling=???, Residual=???, alpha=???)`. ??? needs to be filled in with positive values. See `DIRECT`.

`nIter`
: The number of MCMC iterations.

`burn.in`
: A value in (0,1) indicating the percentage of the MCMC iterations to be used as burn-in and be discarded in posterior inference.

`step.size`
: An integer indicating the number of MCMC iterations to be skipped between two recorded MCMC samples.

`nRepeat`
: An integer ≥ 1 indicating the number of times to update the cluster memberships for all items. Useful only when `SAMPLE.C="Neal"`.

`nResample`
: An integer ≥ 1 indicating the number of resamples to draw to estimate the posterior mixing proportions.

`seed.value`
: A positive value used in random number generation.

`RNORM.METHOD`
: Method to compute the determinant of the covariance matrix in the calculation of the multivariate normal density. Required. Method choices are: "chol" for Choleski decomposition, "eigen" for eigenvalue decomposition, and "svd" for singular value decomposition.

`SAMPLE.C`
: Method to update cluster memberships. Required. Method choices are: "FRBT" for the Metropolis-Hastings sampler based on a discrete uniform proposal distribution developed in Fu, Russell, Bray and Tavare, and "Neal" for the Metropolis-Hastings sampler developed in Neal (2000).

`PRIOR.MODEL`
: Model to generate realizations of the mean vector of a mixture component. Required. Choices are: "none" for not assuming a stochastic process and using a zero vector, "OU" for an Ornstein-Uhlenbeck process (a.k.a. the mean-reverting process), "BM" for a Brown motion (without drift), and "BMdrift" for a Brownian motion with drift.

`ALPHA.METHOD`
: Method to update α, the concentration parameter of the Dirichlet-process prior. Required. Choices are: "Gibbs" for a Gibbs sampler developed in Escobar and West (1995), and "MH" for a Metropolis-Hastings sampler.

OUTPUT.CLUST.SIZE
: If TRUE, output cluster sizes in MCMC iterations into an external file *_mcmc_size.out.

PRINT
: If TRUE, print intermediate values during an MCMC run onto the screen. Used for debugging for small data sets.

Details

The MCMC sampling step in DIRECT is accomplished with DPMCMC. DPMCMC generates MCMC samples of assignments to mixture components (number of components implicitly generated; written into external file *_mcmc_cs.out) and component-specific parameters (written into external file *_mcmc_pars.out), which include mean vectors and standard deviations of three types of variability.

Value

At least two files are generated by DPMCMC and placed under the current working directory:

1. *_mcmc_cs.out: Generated from MCMC sampling. Each row contains an MCMC sample of assignments of items to mixture components, or cluster memberships if a component is defined as a cluster, as well as α, the concentration parameter in the Dirichlet-process prior.
2. *_mcmc_pars.out: Generated from MCMC sampling. Each row contains an MCMC sample of parameters specific to a mixture component. Multiple rows may come from the same MCMC iteration.

If argument OUTPUT.CLUST.SIZE=TRUE, an additional file *_mcmc_size.out is also generated, which contains the cluster sizes of each recorded MCMC sample.

Author(s)

Audrey Q. Fu

References

Escobar, M. D. and West, M. (1995) Bayesian density estimation and inference using mixtures. Journal of the American Statistical Association, 90: 577-588.

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

Neal, R. M. (2000) Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics, 9: 249-265.

See Also

DIRECT, which calls DPMCMC.

Examples

```
## See example in DIRECT.
```

MVNorm *The Multivariate Normal Distribution*

Description

Functions to compute the density of a multivariate normal distribution and to generate random realizations from such a distribution.

Usage

```
dMVNorm (x, mean, sigma, log = FALSE)
rMVNorm (n, mean = rep(0, nrow(sigma)), sigma = diag(length(mean)),
         method=c("eigen", "svd", "chol"))
```

Arguments

x	Vector or matrix of quantiles. If x is a matrix, each row is taken to be a quantile.
n	Number of realizations.
mean	Mean vector, default is rep(0, length = ncol(x)).
sigma	Covariance matrix, default is diag(ncol(x)).
log	Logical; if TRUE, densities are log-transformed.
method	Matrix decomposition used to determine the matrix root of sigma, possible methods are eigenvalue decomposition ("eigen", default), singular value decomposition ("svd"), and Cholesky decomposition ("chol").

Value

rMVNorm returns a vector of the same length as mean if n=1, or a matrix with each row being an independent realization otherwise.

Author(s)

The code for both functions is taken from similar functions written by Friedrich Leisch and Fabian Scheipl in R package mvtnorm. Audrey Q. Fu modified dMVNorm to use a different method to compute the matrix determinants.

Examples

```
## Not run:
x <- rMVNorm (10, mean=rep(0,3), method="svd")
dMVNorm (x, mean=rep(0,3), log=TRUE)

## End(Not run)
```

outputData *Writing Simulation Parameters and Data to Files*

Description

Write simulation parameters and simulated data to files with user-specified filenames.

Usage

```
outputData(datafilename, parfilename, meanfilename,
    simudata, pars, nitem, ntime, nrep)
```

Arguments

datafilename	Name of text file containing simulated data.
parfilename	Name of text file containing simulation parameters, which include number of items, number of time points, number of replicates, true cluster-specific mean vectors, true standard deviations of three types of variability (random effects).
meanfilename	Name of text file containing sample means (averaged over replicates) of simulated data.
simudata	List produced by simuDataREM. Contains simulated data.
pars	Matrix of simulation parameters. Same object as pars.mtx in simuDataREM.
nitem	Number of items.
ntime	Number of time points.
nrep	Number of replicates.

Value

Three files are generated and placed under the current working directory or directories specified in filenames:

- Complete simulated data: Matrix of nitem by ntime*nrep+1. The first column contains the true cluster labels. In the rest of the columns, data are stored as Replicates 1 through nrep at Time 1, Replicates 1 through nrep at Time 2, ..., Replicates 1 through nrep at Time ntime.
- Simulated mean data: Matrix of nitem by ntime. Each row contains the sample means at Times 1 through ntime.
- Simulation parameters:
  - First row: nitem.
  - Second row: ntime.
  - Third row: nrep.
  - Rest of file: Matrix. Each row corresponds to a cluster, and contains cluster label, true mean vector of length ntime, standard deviations of within-cluster variability, variability across time points and residual variability.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

simuDataREM for simulating data.

plotSimulation for plotting simulated data.

DIRECT for clustering the data.

Examples

```
## See example for simuDataREM.
```

---

plotClustersMean *Plotting Clustered Mean Vectors*

Description

Function plotClustersMean produces a plot of multiple panels. Each panel displays for a inferred cluster the mean vectors of items allocated to this cluster, as well as the inferred cluster mean vector. See figures in Fu, Russell, Bray and Tavare.

Usage

```
plotClustersMean(data, data.summary,
    SKIP, nTime = length(times), times = 1:nTime, ...)
```

Arguments

data	An $N \times JR$ matrix of continuous values, or a data frame containing such a matrix. N is the number if items, J the number of time points (or experimental conditions) and R the number of replicates. Each row contains values for Replicates 1 through R under Condition 1, values for Replicates 1 through R under Condition 2, and so on.
data.summary	The list generated from summaryDIRECT that contains processed posterior estimates.
SKIP	Number of columns in data to be skipped when processing the data.
nTime	Number of time points (or experimental conditions).
times	An integer vector of length nTime, indicating times (or experimental conditions).
...	Additional arguments for plot.

Value

None.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

summaryDIRECT for processing MCMC estimates for clustering and generating the list `data.summary` used here.

plotClustersPCA, plotClustersSD, plotSimulation.

Examples

```
## See example in DIRECT.
```

---

plotClustersPCA *PCA Plot for Posterior Allocation Probability Matrix*

Description

Function `plotClustersPCA` generates a Principal Components Analysis (PCA) plot for the posterior mean estimate of allocation probability matrix. The first two principal components are used. See figures in Fu, Russell, Bray and Tavare.

Usage

```
plotClustersPCA(item.names, data.summary,
    PCA.label.adj = -0.01, ...)
```

Arguments

`item.names`	A vector of character strings, indicating how each item should be labeled in the PCA plot.
`data.summary`	The list generated from summaryDIRECT that contains processed posterior estimates.
`PCA.label.adj`	A scalar to be added to the coordinates of `item.names` for better display.
`...`	Additional arguments for `plot`.

Details

The PCA plot produced here displays the uncertainty in the inferred clustering. Each inferred cluster is shown with a distinct color. The closer two clusters are in the PCA plot, the higher the level of uncertainty in inferring these two clusters.

Value

None.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

summaryDIRECT for processing MCMC estimates for clustering and generating the list `data.summary` used here.

plotClustersMean, plotClustersSD, plotSimulation.

Examples

```
## See example in DIRECT.
```

---

plotClustersSD *Plotting Posterior Estimates of Cluster-Specific Random Effects*

Description

Function `plotClustersSD` displays in a single plot the posterior estimates of cluster-specific standard deviations of the three types of variability (random effects) under the DIRECT model. See figures in Fu et al. (2013).

Usage

```
plotClustersSD(data.summary, nTime, ...)
```

Arguments

`data.summary`	The list generated from summaryDIRECT that contains processed posterior estimates.
`nTime`	Number of time points (or experimental conditions).
`...`	Additional arguments for `plot`.

Value

None.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

summaryDIRECT for processing MCMC estimates for clustering and generating the list data.summary used here.

plotClustersPCA, plotClustersPCA, plotSimulation.

Examples

```
## See example in DIRECT.
```

---

plotSimulation *Plotting Data Simulated Under A Random-Effects Mixture Model*

Description

Function plotSimulation displays sample means of data simulated under a random-effects mixture model. Each plot corresponds to a cluster. May need to partition the plotting area to display all in one plot.

Usage

```
plotSimulation(simudata, times = 1:ntime, nsize,
    ntime = length(times), nrep, skip = 0, ...)
```

Arguments

simudata	List produced by simuDataREM. Contains simulated data.
times	Vector of length ntime indicating at which time points data are simulated.
nsize	An integer vector containing sizes of simulated clusters.
ntime	Number of time points.
nrep	Number of replicates.
skip	Not for use.
...	Addition arguments for plot.

Value

None.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

simuDataREM for simulating data.

outputData for writing simulated data and parameter values used in simulation into external files.

DIRECT for clustering the data.

Examples

```
## See example for simuDataREM.
```

---

relabel *A Relabel Algorithm*

Description

Function relabel implements Algorithm 2 in Matthew Stephens (2000) JRSSB for the posterior allocation probability matrix, minimizing the Kullback-Leibler distance. Step 2 in this algorithm is solved using the Hungarian (Munkres) algorithm to the assignment problem.

Usage

```
relabel(probs.mcmc, nIter, nItem, nClust,
    RELABEL.THRESHOLD, PRINT = 0, PACKAGE="DIRECT")
```

Arguments

probs.mcmc	A nItem*nIter-by-nClust matrix of samples of the posterior allocation probability matrix stored in file *_mcmc_probs.out generated by resampleClusterProb.
nIter	Number of stored MCMC samples.
nItem	Number of items.
nClust	Number of inferred clusters.
RELABEL.THRESHOLD	A positive scalar. Used to determine whether the optimization in the relabeling algorithm has converged.

PRINT	If TRUE, print intermediate values onto the screen. Used for debugging with small data sets.
PACKAGE	Not for use.

Value

Permuted labels for each store MCMC sample are written to file *_mcmc_perms.out, in which each row contains an inferred permutation (relabel) of labels of mixture components.

Note

This function calls a routine written in C, where implementation of Munkres algorithm is adapted from the C code by Dariush Lotfi (June 2008; web download).

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

Stephens, M. (2000) Dealing with label switching in mixture models. Journal of the Royal Statistical Society, Series B, 62: 795-809.

See Also

DIRECT for the complete method.

DPMCMC for the MCMC sampler under the Dirichlet-process prior.

resampleClusterProb for resampling of posterior allocation probability matrix in posterior inference.

Examples

```
## See example for DIRECT.
```

---

resampleClusterProb *Resampling to Estimate Posterior Allocation Probability Matrix*

Description

The resampling method as part of the posterior inference under DIRECT. It uses stored MCMC samples to generate realizations of the allocation probability matrix, and writes the realizations to a user-specified external file.

Usage

```
resampleClusterProb(file.out, ts, nitem, ntime, nrep,
    pars.mcmc, cs.mcmc, alpha.mcmc, nstart, nres)
```

Arguments

file.out	Name of file containing samples of posterior allocation probability matrix.
ts	A nitem-by-ntime-by-nrep array of data.
nitem	Number of items.
ntime	Number of time points.
nrep	Number of replicates.
pars.mcmc	A matrix or data frame of MCMC samples of mean vectors and random effects stored in file *_mcmc_pars.out, one of the output files from DPMCMC.
cs.mcmc	A matrix or data frame of MCMC samples of assignments of mixture components stored in file *_mcmc_cs.out, one of the output files from DPMCMC.
alpha.mcmc	A vector of MCMC samples of α, the concentration parameter in the Dirichlet-process prior, stored in the last column of file *_mcmc_cs.out, one of the output files from DPMCMC.
nstart	Starting from which recorded MCMC sample.
nres	How many times to draw resamples? Multiple samples are averaged.

Value

Samples of the allocation probability matrix are written to file *_mcmc_probs.out. This file contains a large matrix of $HN \times K$, which is H posterior allocation probability matrices stacked up, each individual matrix of $N \times K$, where H is the number of recorded MCMC samples, N the number of items and K the inferred number of mixture components.

Note

resampleClusterProb calls the following functions adapted or directly taken from existing R functions:

- dMVNorm is adapted from dmvnorm by Friedrich Leisch and Fabian Scheipl in package mvtnorm.
- rMVNorm is adapted from rmvnorm by Friedrich Leisch and Fabian Scheipl in package mvtnorm.
- rDirichlet is taken from rdirichlet by Gregory R. Warnes, Ben Bolker and Ian Wilson in package gregmisc.
- dDirichlet is based on ddirichlet by Gregory R. Warnes, Ben Bolker and Ian Wilson in package gregmisc.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

DIRECT for the complete method.

DPMCMC for the MCMC sampler under the Dirichlet-process prior.

relabel for relabeling in posterior inference.

Examples

```
## See example for DIRECT.
```

---

simuDataREM *Data Simulation Under the Random-Effects Mixture Model*

Description

Function simuDataREM simulates data under the Ornstein-Uhlenbeck (OU) (or Brownian Motion; BM) process-based random-effects mixture (REM) model.

Usage

```
simuDataREM(pars.mtx, dt, T, ntime, nrep, nsize, times,
    method = c("eigen", "svd", "chol"), model = c("OU", "BM"))
```

Arguments

pars.mtx	A $K \times 8$ matrix, where K is the number of clusters. Each row contains 8 parameters: standard deviation of within-cluster variability, of variability across time points, and of replicates, respectively; mean and standard deviation for the value at the first time point; the overall mean, standard deviation and mean-reverting rate of the OU process.
dt	Increment in times.
T	Maximum time.
ntime	Number of time points to simulate data for. Needs to be same as the length of vector times.
nrep	Number of replicates.
nsize	An integer vector containing sizes of simulated clusters.
times	Vector of length ntime indicating at which time points to simulate data.
method	Method to compute the determinant of the covariance matrix in the calculation of the multivariate normal density. Required. Method choices are: "chol" for Choleski decomposition, "eigen" for eigenvalue decomposition, and "svd" for singular value decomposition.

`model` Model to generate realizations of the mean vector of a mixture component. Required. Choices are: "OU" for an Ornstein-Uhlenbeck process (a.k.a. the mean-reverting process) and "BM" for a Brown motion (without drift).

Value

`means` A matrix of `ntime` columns. The number of rows is the same as that of `pars.mtx`, which is the number of clusters. Each row contains the true mean vector of the corresponding cluster.

`data` A matrix of N rows and `ntime*nrep+1` columns, where N is the sum of cluster sizes `nsize`. The first column contains the true cluster membership of the corresponding item. The rest of the columns in each row is formatted as follows: values for replicate 1 through `nrep` at time 1; values for replicate 1 through `nrep` at time 2, ...

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

`plotSimulation` for plotting simulated data.

`outputData` for writing simulated data and parameter values used in simulation into external files.

`DIRECT` for clustering the data.

Examples

```
## Not run:
# Simulate replicated time-course gene expression profiles
# from OU processes

# Simulation parameters
times = c(0,5,10,15,20,25,30,35,40,50,60,70,80,90,100,110,120,150)
ntime=length (times)
nrep=4

nclust = 6
npars = 8
pars.mtx = matrix (0, nrow=nclust, ncol=npars)
# late weak upregulation or downregulation
pars.mtx[1,] = c(0.05, 0.1, 0.5, 0, 0.16, 0.1, 0.4, 0.05)
# repression
pars.mtx[2,] = c(0.05, 0.1, 0.5, 1, 0.16, -1.0, 0.1, 0.05)
# early strong upregulation
pars.mtx[3,] = c(0.05, 0.5, 0.2, 0, 0.16, 2.5, 0.4, 0.15)
```

```
# strong repression
pars.mtx[4,] = c(0.05, 0.5, 0.2, 1, 0.16, -1.5, 0.4, 0.1)
# low upregulation
pars.mtx[5,] = c(0.05, 0.3, 0.3, -0.5, 0.16, 0.5, 0.2, 0.08)
# late strong upregulation
pars.mtx[6,] = c(0.05, 0.3, 0.3, -0.5, 0.16, 0.1, 1, 0.1)

nsize = rep(40, nclust)

# Generate data
simudata = simuDataREM (pars=pars.mtx, dt=1, T=150,
    ntime=ntime, nrep=nrep, nsize=nsize, times=times, method="svd", model="OU")

# Display simulated data
plotSimulation (simudata, times=times,
    nsize=nsize, nrep=nrep, lty=1, ylim=c(-4,4), type="l", col="black")

# Write simulation parameters and simulated data
# to external files
outputData (datafilename= "simu_test.dat", parfilename= "simu_test.par",
    meanfilename= "simu_test_mean.dat", simudata=simudata, pars=pars.mtx,
    nitem=sum(nsize), ntime=ntime, nrep=nrep)

## End(Not run)
```

---

summaryDIRECT *Processing Posterior Estimates for Clustering Under DIRECT*

Description

Function summaryDIRECT processes posterior estimates in the output files from DIRECT for clustering and parameter estimation.

Usage

```
summaryDIRECT(data.name, PERM.ADJUST = FALSE)
```

Arguments

data.name	A character string indicating the prefix of output files.
PERM.ADJUST	If TRUE, add 1 to labels of mixture components such that the labels start from 1 instead of 0.

Details

Output files from DIRECT include MCMC samples before relabeling and permuted labels of mixture components after relabeling. Function summaryDIRECT uses permuted labels stored in output file *_mcmc_perms.out to reorganize the MCMC samples stored in other output files *_mcmc_cs.out, *_mcmc_pars.out and *_mcmc_probs.out. It defines each mixture component as a cluster.

Value

A list with components:

`nitem` The number of items in the data.

`nclust` The number of inferred clusters.

`top.clust.alloc`
A vector of length `nitem`, each component being the maximum posterior probability of allocating the corresponding item to a cluster.

`cluster.sizes` Vector of cluster sizes.

`top.clust.labels`
An integer vector of labels of inferred clusters. The integers are not necessarily consecutive; that is, an inferred mixture component that is associated with items at small posterior allocation probabilities is dropped from the final list of cluster labels.

`top2allocations`
A data frame containing "first", the most likely allocation; "second", the second most likely allocation; "prob1", the posterior allocation probability associated with "first"; and "prob2", the posterior allocation probability associated with "second".

`post.alloc.probs`
A `nitem`-by-`nclust` matrix of mean posterior allocation probability matrix.

`post.clust.pars.mean`
A matrix of `nclust` rows. Each row, corresponding to an inferred cluster, contains the posterior mean estimates of cluster-specific parameters.

`post.clust.pars.median`
A matrix of `nclust` rows. Each row, corresponding to an inferred cluster, contains the posterior median estimates of cluster-specific parameters.

`misc` A list containing two components:

- `post.pars.mean`: Matrix each row of which contains the posterior mean estimates of parameters for a mixture component.
- `post.pars.median`: Matrix each row of which contains the posterior median estimates of parameters for a mixture component.

Author(s)

Audrey Q. Fu

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

See Also

`DIRECT` for what output files are produced.

`simuDataREM` for simulating data under the mixture random-effects model.

Examples

```
## See example in DIRECT.
```

`tc.data` *Time-Course Microarray Gene Expression Data*

Description

This data set contains quantile-normalized microarray gene expression measurements of 163 genes from four replicates at 18 time points. These data are part of the time-course experiment performed on Drosophila with a 5-min pulse of Notch activation (Housden et al. 2013). The experiment was carried out by Sarah Bray, Ben Housden, Alena Krejci and Bettina Fischer; see details in Housden et al. (2013).

Usage

```
data(tc.data)
```

Format

A data frame with 163 observations on the 74 variables. The first two variables are `GeneID` and `GeneName`.

Other variables are log2 fold change of treated cells over control cells for 4 biological replicates at 18 time points. They are organized as follows: values for replicates 1 through 4 at time 1; values for replicates 1 through 4 at time 2; and so on.

Details

The 18 time points are (in min):

0,5,10,15,20,25,30,35,40,50,60,70,80,90,100,110,120,150.

Microarray data have been cleaned and normalized. Missing values are imputed. See supplementary material for Fu, Russell, Bray and Tavare for detail on data pre-processing and missing value imputation.

References

Fu, A. Q., Russell, S., Bray, S. and Tavare, S. (2013) Bayesian clustering of replicated time-course gene expression data with weak signals. The Annals of Applied Statistics. 7(3) 1334-1361.

Housden, B. E., Fu, A. Q., Krejci, A., Bernard, F., Fischer, B., Tavare, S., Russell, S. and Bray, S. J. (2013) Transcriptional dynamics elicited by a short pulse of Notch activation involves feed-forward regulation by E(spl)/Hes genes. PLoS Genetics 9 1 e1003162.

Examples

```
## Not run:
# Compute mean profiles for genes
# and plot the means as a heatmap with the color scale on the side

library (fields) # to use function image.plot

data (tc.data)
times = c(0,5,10,15,20,25,30,35,40,50,60,70,80,90,100,110,120,150)

# Organize data into array of nGene-by-nTime-by-nRep
SKIP=2
nTime=length (times)
nGene = nrow (tc.data)
nRep = (ncol (tc.data) - SKIP) / nTime

ts = array (0, dim = c(nGene, nTime, nRep))
for (r in 1:nRep) {
    ts[,,r] = as.matrix (tc.data[,SKIP + (0:(nTime-1))*nRep + r])
}

# Compute mean profile for each gene
ts.mean = apply (ts, c(1,2), mean)

# Plot heatmap for mean profiles
image.plot (1:nGene, times, as.matrix(ts.mean),
    xlab="gene", ylab="time (min)",
    cex=1.5, cex.axis = 1.6, cex.lab = 1.6,
    legend.shrink=1, legend.width=2, col=topo.colors(8))

## End(Not run)
```

Index

∗ **cluster**
DIRECT, 3
∗ **datasets**
tc.data, 27
∗ **models**
DIRECT, 3
∗ **multivariate**
DIRECT, 3
∗ **package**
DIRECT-package, 2
∗ **ts**
DIRECT, 3

dDirichlet *(*Dirichlet*)*, 9
DIRECT, *2, 3*, 3, *10–13*, *16*, *20, 21*, *23–26*
direct *(*DIRECT*)*, 3
DIRECT-package, 2
Dirichlet, 9
dMVNorm, *7*
dMVNorm *(*MVNorm*)*, 14
DPMCMC, *8*, 10, *21–23*

MVNorm, 14

outputData, 15, *20*, *24*

plotClustersMean, 16, *18*
plotClustersPCA, *17*, 17, *19*
plotClustersSD, *17, 18*, 18
plotSimulation, *16–19*, 19, *24*

rDirichlet, *7*
rDirichlet *(*Dirichlet*)*, 9
relabel, *8*, 20, *23*
resampleClusterProb, *8*, *20, 21*, 21
rMVNorm, *7*
rMVNorm *(*MVNorm*)*, 14

simuDataREM, *3*, *8*, *15, 16*, *19, 20*, 23, *26*
summaryDIRECT, *2, 3*, *8*, *16–19*, 25

tc.data, 27