Package 'DGP4LCF'

January 20, 2025

Type Package

Title Dependent Gaussian Processes for Longitudinal Correlated Factors

Version 1.0.0

Maintainer Jiachen Cai <jiachen.cai@mrc-bsu.cam.ac.uk>

Description Functionalities for analyzing high-dimensional and longitudinal biomarker data to facilitate precision medicine, using a joint model of Bayesian sparse factor analysis and dependent Gaussian processes. This paper illustrates the method in detail: J Cai, RJB Goudie, C Starr, BDM Tom (2023) <doi:10.48550/arXiv.2307.02781>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports GPFDA, Rcpp, factor.switching, mvtnorm, combinat, coda, corrplot, pheatmap, stats

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

VignetteBuilder knitr

NeedsCompilation yes

Author Jiachen Cai [aut, cre]

Repository CRAN

Date/Publication 2024-05-28 16:40:05 UTC

Contents

`factor_loading_heatmap` *Displaying significant factor loadings in the heatmap.*

Description

This function is used to visualize results of estimates of factor loadings (in heatmaps).

Usage

```
factor_loading_heatmap(factor_loading_matrix, heatmap_title)
```

Arguments

`factor_loading_matrix`	A matrix of dimension (p, k), which stores results for factor loadings.
`heatmap_title`	A character. Title for the heatmap.

Value

A heatmap presenting posterior median estimates of factor loadings.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
```

`factor_score_trajectory` *Plotting figures for factor score trajectory.*

Description

This function is used to visualize results of factor score trajectories.

Usage

```
factor_score_trajectory(
  factor_score_matrix,
  factor_index,
  person_index,
  trajectory_title,
  cex_main = 1
)
```

Arguments

`factor_score_matrix`	A matrix of dimension (q, k, n), used to store results for factor scores.
`factor_index`	A numeric scalar. Index of the factor of interest.
`person_index`	A numeric scalar. Index of the person of interest.
`trajectory_title`	A character. Title for the factor trajectory plot.
`cex_main`	A numeric scalar. Text size of the title.

Value

Trajectory of the designated person-factor.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
```

gibbs_after_mcem_algorithm *Generating posterior samples for parameters (other than DGP parameters) in the model and predicted gene expression for one chain.*

Description

Generating posterior samples for parameters (other than DGP parameters) in the model and predicted gene expression for one chain.

Usage

```
gibbs_after_mcem_algorithm(
  chain_index,
  mc_num,
  burnin,
  thin_step,
  pathname,
  pred_indicator = FALSE,
  pred_time_index = NULL,
  x,
  mcem_parameter_setup_result,
  mcem_algorithm_result,
  gibbs_after_mcem_diff_initials_result
)
```

Arguments

chain_index	A numeric scalar. Index of the chain.
mc_num	A numeric scalar. Number of iterations in the Gibbs sampler.
burnin	A numeric scalar. Number of iterations to be discarded as 'burn-in'.
thin_step	A numeric scalar. This function will only save every 'thin_step'th iteration results in the specified directory to reduce storage space needed. Note that this number can be different from that used in the function 'mcem_algorithm'.
pathname	A character. The directory where the saved Gibbs samplers are stored.
pred_indicator	A logical value. pred_indicator = TRUE denotes the need to predict gene expression at new time points. The default value is FALSE.
pred_time_index	Only needed if pred_indicator = TRUE. Index of the new time points in the full time vector.
x	A list of n elements. Each element is a matrix of dimension (p, q_i), storing the gene expression observed at q_i time points for the ith subject.
mcem_parameter_setup_result	A list of objects returned from the function 'mcem_parameter_setup'.

`mcem_algorithm_result`
: A list of objects returned from the function 'mcem_algorithm'.

`gibbs_after_mcem_diff_initials_result`
: A list of objects returned from the function 'gibbs_after_mcem_diff_initials'.

Details

This function corresponds to Algorithm 2: Step 1 in the main manuscript; therefore reader can consult the paper for more explanations.

Value

Posterior samples for parameters (other than DGP parameters) in the model and predicted gene expression for one chain.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

---

gibbs_after_mcem_combine_chains

Combining from all chains the posterior samples for parameters in the model and predicted gene expressions.

Description

Combining from all chains the posterior samples for parameters in the model and predicted gene expressions.

Usage

```
gibbs_after_mcem_combine_chains(tot_chain, gibbs_after_mcem_algorithm_result)
```

Arguments

`tot_chain`
: A numeric scalar. Total number of chains.

`gibbs_after_mcem_algorithm_result`
: A list of objects storing model constants. Should be the same as that input to the 'function gibbs_after_mcem_load_chains'.

Value

All saved posterior samples for parameters in the model and predicted gene expressions.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
```

`gibbs_after_mcem_diff_initials` *Generating different initials for multiple chains.*

Description

Generating different initials for multiple chains.

Usage

```
gibbs_after_mcem_diff_initials(
  ind_x = TRUE,
  tot_chain = 5,
  mcem_parameter_setup_result,
  mcem_algorithm_result
)
```

Arguments

`ind_x`	A logical value. ind_x = TRUE uses the model including the intercept term for subject-gene mean in after-MCEM-Gibbs sampler; otherwise uses the model without the intercept term.
`tot_chain`	A numeric scalar. Number of parallel chains.
`mcem_parameter_setup_result`	A list of objects returned from the function 'mcem_parameter_setup'.
`mcem_algorithm_result`	A list of objects returned from the function 'mcem_algorithm'.

Value

Different initials for multiple chains.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

`gibbs_after_mcem_load_chains` *Loading the saved posterior samples for parameters in the model and predicted gene expressions.*

Description

Loading the saved posterior samples for parameters in the model and predicted gene expressions.

Usage

```
gibbs_after_mcem_load_chains(chain_index, gibbs_after_mcem_algorithm_result)
```

Arguments

`chain_index`	A numeric scalar. Index of the chain.
`gibbs_after_mcem_algorithm_result`	A list of objects storing model constants.

Value

All saved posterior samples for parameters in the model and predicted gene expressions.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example",  package = "DGP4LCF")
```

`mcem_algorithm` *Monte Carlo Expectation Maximization (MCEM) algorithm to return the Maximum Likelihood Estimate (MLE) of DGP Parameters.*

Description

This function is used to return the MLE of DGP parameters.

Usage

```
mcem_algorithm(
  ind_x,
  ig_parameter = 10^-2,
  increasing_rate = 0.5,
  prob_conf_interval = 0.9,
  iter_count_num = 5,
  x,
```

```
  mcem_parameter_setup_result,
  ipt_x = FALSE,
  missing_list = NULL,
  missing_num = NULL
)
```

Arguments

ind_x	A logical value. ind_x = TRUE uses the model including the intercept term for subject-gene mean in within-MCEM-Gibbs sampler; otherwise uses the model without the intercept term.
ig_parameter	A numeric scalar. Hyper-parameters for the prior Inverse-Gamma distribution.
increasing_rate	A numeric scalar. Rate of increasing the sample size.
prob_conf_interval	A numeric scalar. The probability that the true change in the Q-function is larger than the lower bound.
iter_count_num	A numeric scalar. Maximum number of increasing the sample size; a larger number than this would end the algorithm.
x	A list of n elements. Each element is a matrix of dimension (p, q_i), storing the gene expression observed at q_i time points for the ith subject.
mcem_parameter_setup_result	A list of objects returned from the function 'mcem_parameter_setup'.
ipt_x	A logical value. ind_x = TRUE denotes the need to impute for NAs of gene expression. The default value is ind_x = FALSE.
missing_list	A list of n elements. Each element is a matrix of dimension (missing_num, 2): each row corresponds to the position of one NA that needs imputation; first and second columns denote the row and column indexes, respectively, of the NA in the corresponding person's matrix of gene expression.
missing_num	A vector of n elements. Each element corresponds to a single person's number of NAs that needs imputation.

Value

The MLE of DGP parameters.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

`mcem_cov_plot` *Visualizing cross-correlations among factors.*

Description

Visualizing cross-correlations among factors.

Usage

```
mcem_cov_plot(k, q, cov_input, title)
```

Arguments

`k`	A numeric scalar. Number of latent factors.
`q`	A numeric scalar. Number of time points in the covariance matrix of factors.
`cov_input`	A matrix of dimension (kq, kq). The covariance matrix of the vector obtained from vectorizing the matrix of latent factor scores.
`title`	A character. Title for the plot.

Value

Visualization of cross-correlations among factors.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

`mcem_parameter_setup` *Parameters' setup and initial value assignment for the Monte Carlo Expectation Maximization (MCEM) algorithm.*

Description

This function is used to create R objects storing parameters in the desired format, and assign initial values so that they are ready to use in the MCEM algorithm.

Usage

```
mcem_parameter_setup(
  p,
  k,
  n,
  q,
  ind_num = 10,
  burn_in_prop = 0.2,
  thin_step = 5,
  prior_sparsity = 0.1,
  em_num = 50,
  obs_time_num,
  obs_time_index,
  a_person,
  col_person_index,
  y_init,
  a_init,
  z_init,
  phi_init,
  a_full,
  train_index,
  x,
  model_dgp = TRUE
)
```

Arguments

p	A numeric scalar. Number of genes.
k	A numeric scalar. Number of latent factors.
n	A numeric scalar. Number of subjects.
q	A numeric scalar. Complete number of time points in the training data.
ind_num	A numeric scalar. Starting size of approximately independent samples for MCEM.
burn_in_prop	A numeric scalar. Proportion of burnin, which be used to calculate size of Monte Carlo samples needed in the Gibbs sampler. Must be the same as that in the function 'mcem_algorithm_irregular_time'.
thin_step	A numeric scalar. Thinning step, which be used to calculate size of Monte Carlo samples needed in the Gibbs sampler. Must be the same as that in the function 'mcem_algorithm_irregular_time'.
prior_sparsity	A numeric scalar. Prior expected proportion of genes involved within each pathway.
em_num	A numeric scalar. Maximum iterations of the expectation maximization (EM) algorithm allowed.
obs_time_num	A n-dimensional vector. One element represents one person's observed number of time points in the training data.
obs_time_index	A list of n elements. One element is a vector of observed time indexes for one person in the training data, sorted from early to late.

`a_person` A list of n elements. One element is a vector of observed time for one subject in the training data, sorted from early to late.

`col_person_index` A list of n elements. One element is a vector of column indexes for one subject in y_init.

`y_init` A matrix of dimension (k, sum(obs_time_num)). Initial values of the latent factor score. Can be obtained using BFRM software.

`a_init` A matrix of dimension (p, k). Initial values of the regression coefficients of factor loadings. Can be obtained using BFRM software.

`z_init` A matrix of dimension (p, k). Initials values of the binary variables of factor loadings. Can be obtained using BFRM software.

`phi_init` A p-dimensional column vector. Initials values of the variance for residuals when modeling gene expressions, corresponding to $\frac{1}{\phi^2}$ in the manuscript. Can be obtained using BFRM software.

`a_full` A numeric vector. Complete time observed, sorted from early to late.

`train_index` A q-dimensional column vector. Index of time points used in the training data.

`x` A list of n elements. Each element is a matrix of dimension (p, q_i), storing the gene expressions for the ith subject.

`model_dgp` A logical value. model_dgp = TRUE (default setting) uses the Dependent Gaussian Process to model latent factor trajectories, otherwise the Independent Gaussian Process is used.

Details

The following parameters are worth particular attention, and users should tune these parameters according to the specific data.

'burn_in_prop' and 'thin_step' co-control the number of Gibbs samples needed in order to generate approximately 'ind_num' independent samples. The ultimate purpose of tuning these two parameters is to generate high-quality posterior samples for latent factor scores. Therefore: if initials of the Gibbs sampler are not good, readers may need to increase 'burn_in_prop' to discard more burn-in samples; if high-correlation is a potential concern, 'thin_step' may need to be larger.

Value

A list of R objects required in the MCEM algorithm.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

numerics_summary_do_not_need_alignment *Numerical summary for important continuous variables that do not need alignment.*

Description

Numerical summary for important continuous variables that do not need alignment.

Usage

```
numerics_summary_do_not_need_alignment(
  burnin = 0,
  thin_step = 1,
  pred_x_truth_indicator = FALSE,
  pred_x_truth = NULL,
  gibbs_after_mcem_combine_chains_result
)
```

Arguments

burnin	A numeric scalar. The saved samples are already after burnin; therefore the default value for this parameter here is 0. Can discard further samples if needed.
thin_step	A numeric scalar. The saved samples are already after thinning; therefore the default value for this parameter here is 1. Can be further thinned if needed.
pred_x_truth_indicator	A logical value. pred_x_truth_indicator = TRUE means that truth of predicted gene expressions are available. The default value is FALSE.
pred_x_truth	Only needed if pred_x_truth_inidcator = TRUE. An array of dimension (n, p, num_time_test), storing true gene expressions in the testing data.
gibbs_after_mcem_combine_chains_result	A list of objects returned from the function 'gibbs_after_mcem_combine_chains'.

Details

This function corresponds to Algorithm 2: Steps 3 and 4 in the main manuscript; therefore reader can consult the paper for more explanations.

Value

Convergence assessment for important continuous variables that do not need alignment, and posterior summary for predicted gene expressions.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example",  package = "DGP4LCF")
```

numerics_summary_need_alignment *Numerical summary for factor loadings and factor scores, which need alignment.*

Description

Numerical summary for factor loadings and factor scores, which need alignment.

Usage

```
numerics_summary_need_alignment(
  burnin = 0,
  thin_step = 1,
  gibbs_after_mcem_combine_chains_result
)
```

Arguments

burnin	A numeric scalar. The saved samples are already after burnin; therefore the default value for this parameter here is 0. Can discard further samples if needed.
thin_step	A numeric scalar. The saved samples are already after thinning; therefore the default value for this parameter here is 1. Can be further thinned if needed.
gibbs_after_mcem_combine_chains_result	A list of objects returned from the function 'gibbs_after_mcem_combine_chains'.

Details

This function corresponds to Algorithm 2: Steps 2, 3 and 4 in the main manuscript; therefore reader can consult the paper for more explanations.

Value

Reordered posterior samples, convergence assessment, and summarized posterior results for factor loadings and factor scores.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

`sim_fcs_init` *Initials values.*

Description

Initial values provided by the two-step approach.

Usage

```
sim_fcs_init
```

Format

An object of class `list` of length 14.

`sim_fcs_results_irregular_6_8` *Results when people have irregularly observed time points (some 6 while others 8).*

Description

Results when people have irregularly observed time points (some 6 while others 8).

Usage

```
sim_fcs_results_irregular_6_8
```

Format

An object of class `list` of length 3.

`sim_fcs_results_regular_8` *Results when people are observed at common 8 time points.*

Description

Results when people are observed at common 8 time points.

Usage

```
sim_fcs_results_regular_8
```

Format

An object of class `list` of length 3.

sim_fcs_truth *Truth of simulated data.*

Description

Simulated data under the scenario where factors are correlated and have small variability (CS).

Usage

```
sim_fcs_truth
```

Format

An object of class `list` of length 19.

subject_specific_objects *Constructing subject-specific objects required for Gibbs sampler (for subjects with incomplete observations only).*

Description

Constructing subject-specific objects required for Gibbs sampler (for subjects with incomplete observations only).

Usage

```
subject_specific_objects(k, q, a_full, a_avail, cor_all)
```

Arguments

`k`	A numeric scalar. Number of latent factors.
`q`	A numeric scalar. Number of time points in the complete factor covariance matrix.
`a_full`	A q-dimensional numeric vector. Complete time sorted from early to late.
`a_avail`	A vector of time when gene expressions are available, sorted from early to late.
`cor_all`	A matrix of dimension (kq, kq). Correlation matrix of latent factor scores.

Details

This function is used to extract subject-specific factor covariance matrix from the complete factor covariance matrix, through constructing subject-specific indicator matrix, which indicates time indexes when gene expression are available.

Value

Subject-specific objects needed for Gibbs sampler.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

---

table_generator *Generating a table listing all possible combinations of the binary variables for one gene.*

Description

Generating a table listing all possible combinations of the binary variables for one gene.

Usage

```
table_generator(k)
```

Arguments

k	A numeric scalar. Number of latent factors.

Value

A table listing all possible combinations of the binary variables for one gene.

Examples

```
# See examples in vignette
vignette("bsfadgp_regular_data_example", package = "DGP4LCF")
vignette("bsfadgp_irregular_data_example", package = "DGP4LCF")
```

Index