Package: GPTCM (via r-universe)

June 4, 2026

Title Generalized Promotion Time Cure Model with Bayesian Shrinkage Priors

Version 2.0.0

Description Generalized promotion time cure model (GPTCM) via Bayesian hierarchical modeling for multiscale data integration (Zhao et al. (2025) <doi:10.48550/arXiv.2509.01001>). The Bayesian GPTCMs are applicable for both low- and high-dimensional data.

Maintainer Zhi Zhao <zhi.zhao@medisin.uio.no>

Copyright The code in src/arms.cpp is slightly modified based on the research paper implementation written by Wally Gilks.

URL https://github.com/ocbe-uio/GPTCM

BugReports https://github.com/ocbe-uio/GPTCM/issues

License GPL-3

VignetteBuilder knitr

Depends R (>= 4.1.0)

Encoding UTF-8

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, survival, riskRegression, ggplot2, ggridges, miCoPTCM, loo, mvnfast, Matrix, scales, utils, stats, graphics

Suggests knitr, survminer

NeedsCompilation yes

SystemRequirements C++17

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev

Repository https://ocbe-uio.r-universe.dev

Date/Publication 2026-06-04 23:05:30 UTC

RemoteUrl https://github.com/ocbe-uio/GPTCM

RemoteRef HEAD

RemoteSha 14488632e8bf031e8775d8a7a342c9c82c0cd67f

Contents

`getEstimator` *Extract the posterior estimate of parameters*

Description

Extract the posterior estimate of the parameters of a `GPTCM` class object.

Usage

```
getEstimator(object, estimator = "gamma", Pmax = 0, type = "marginal")
```

Arguments

`object`	an object of class `GPTCM`
`estimator`	the name of one estimator. Default is the latent indicator estimator "`gamma`". Other options are among "`c('beta', 'zeta', 'eta', 'xi', 'elpd', 'logP')`"
`Pmax`	threshold that truncate the estimator "`gamma`" or "`eta`". Default is `0`. If `Pmax=0.5` and `type="conditional"`, it gives median probability model betas
`type`	the type of output beta. Default is `marginal`, giving marginal beta estimation. If `type="conditional"`, it gives beta estimation conditional on gamma=1

Value

Return the estimator from an object of class `GPTCM`. It is a matrix or vector

References

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)

# run a Bayesian GPTCM model: GPTCM-Ber2
fit <- GPTCM(dat, nIter = 10, burnin = 0)

gamma.hat <- getEstimator(fit, estimator = "gamma")
```

GPTCM *Fit Bayesian GPTCM Models*

Description

This is the main function to fit the Bayesian GPTCMs (Zhao et al. 2025) with multiscale data for sparse identification of high-dimensional covariates. The core code for MCMC algorithm uses Rcpp (Eddelbuettel and François 2011) and RcppArmadillo (Eddelbuettel and Sanderson 2014)

Usage

```
GPTCM(
  dat,
  nIter = 500,
  burnin = 200,
  thin = 1,
  tick = 500,
  proportion.model = TRUE,
  dirichlet = TRUE,
  hyperpar = NULL,
  BVS = TRUE,
  threads = 1,
  kappaIGamma = FALSE,
  kappaSampler = "arms",
  gammaPrior = "bernoulli",
  gammaSampler = "MC3",
  etaPrior = "bernoulli",
  etaSampler = "MC3",
  w0IGamma = TRUE,
  initial = NULL,
  arms.list = NULL
)
```

Arguments

dat	input data as a list containing survival data sub-list survObj with two vectors (event and time), clinical variable matrix x0, cluster-specific covariates X, and proportions data matrix proportion
nIter	the number of iterations of the chain
burnin	number of iterations to discard at the start of the chain
thin	thinning MCMC intermediate results to be stored
tick	an integer used for printing the iteration index and some updated parameters every tick-th iteration. Default is 1
proportion.model	logical value; should the proportions be modeled or not. If (proportion.model = FALSE), the argument dirichlet will be invalid
dirichlet	logical value; should the proportions be modeled via the common (dirichlet = TRUE) or alternative (dirichlet = FALSE) parametrization of the Dirichlet regression model
hyperpar	a list of relevant hyperparameters
BVS	logical value for implementing Bayesian variable selection
threads	maximum threads used for parallelization. Default is 1
kappaIGamma	logical value for using inverse-gamma prior (TRUE) or gamma prior (FALSE) for Weibull's shape parameter shape parameter
kappaSampler	one of "arms", "slice" (slice not yet implemented)
gammaPrior	one of c("bernoulli", "MRF")
gammaSampler	one of c("mc3", "bandit")
etaPrior	one of c("bernoulli", "MRF")
etaSampler	one of c("mc3", "bandit")
w0IGamma	logical value; if FALSE, a common parameter is used for the intercept's prior variance and the coefficient's prior variance
initial	a list of initial values for parameters "kappa", "xi", "betas", and "zetas"
arms.list	a list of parameters for the ARMS method

Value

An object of a list including the following components:

- input - a list of all input parameters by the user
- output - a list of the all mcmc output estimates:
  - "xi" - a matrix with MCMC intermediate estimates of effects on clinical variables
  - "kappa" - a vector with MCMC intermediate estimates of the Weibull's shape parameter
  - "betas" - a matrix with MCMC intermediate estimates of effects on cluster-specific survival
  - "zetas" - a matrix with MCMC intermediate estimates of effects on cluster-specific proportions

  - "gammas" - a matrix with MCMC intermediate estimates of inclusion indicators of variables for cluster-specific survival
  - "gamma_acc_rate" - acceptance rate of the M-H sampling for gammas
  - "etas" - a matrix with MCMC intermediate estimates of inclusion indicators of variables for cluster-specific proportions
  - "eta_acc_rate" - acceptance rate of the M-H sampling for etas
  - "loglikelihood" - a matrix with MCMC intermediate estimates of individuals' likelihoods
  - "tauSq" - a vector with MCMC intermediate estimates of tauSq
  - "wSq" - a matrix with MCMC intermediate estimates of wSq
  - "vSq" - a matrix with MCMC intermediate estimates of vSq
  - "post" - a list with posterior means of "xi", "kappa", "betas", "zetas", "gammas", "etas"

- call - the matched call

References

Eddelbuettel D, Sanderson C (2014). *RcppArmadillo: Accelerating R with high-performance C++ linear algebra*. Computational Statistics and Data Analysis, 71, 1054–1063

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)

# run a Bayesian GPTCM model: GPTCM-Ber2
fit <- GPTCM(dat, nIter = 10, burnin = 0)
```

---

metropolis_sampler *Metropolis sampler for a target density*

Description

Random number generator via Metropolis-Hastings algorithm.

Usage

```
metropolis_sampler(
  initial_value,
  n = n,
  proposal_shape = 1,
  proposal_scale = 1,
  theta = 1,
  proportion = 0.5,
  mu = 1,
  kappas = 0.9,
  burnin = 0,
  lag = 1
)
```

Arguments

initial_value	initial values
n	number of draws
proposal_shape	Weibull's shape parameter in the proposal
proposal_scale	Weibull's scale parameter in the proposal
theta	cure rate parameter (log scale)
proportion	proportions data
mu	mean survival time
kappas	Weibull's true shape parameter
burnin	length of burn-in period
lag	discarding lag-1 values in the Metropolis step

Value

A dataframe consisting of the sampled values and acceptance rate

Examples

```
times <- metropolis_sampler(10, 5)
```

---

plotBrier *Plot curves of time-dependent Brier score*

Description

Predict time-dependent Brier scores based on different survival models

Usage

```
plotBrier(
  dat,
  datMCMC,
  dat.new = NULL,
  time.star = NULL,
  xlab = "Time",
  ylab = "Brier score",
  PTCM = TRUE,
  ...
)
```

Arguments

dat	input data as a list containing survival data sub-list survObj with two vectors (event and time), clinical variable matrix x0, cluster-specific covariates X, and proportions data matrix proportion
datMCMC	returned object from the main function GPTCM()
dat.new	input data for out-sample prediction, with the same format as dat
time.star	largest time for survival prediction
xlab	a title for the x axis
ylab	a title for the y axis
PTCM	logical value for adding survival prediction by the PTCM
...	other parameters

Value

A ggplot2::ggplot object. See ?ggplot2::ggplot for more details of the object.

References

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)

# run a Bayesian GPTCM model: GPTCM-Ber2
fit <- GPTCM(dat, nIter = 5, burnin = 0)
```

```
plotBrier(dat, datMCMC = fit, PTCM = FALSE)
```

---

plotCoeff *Plot posterior estimates of regression coefficients*

Description

create nice plots for estimated coefficients and 95

Usage

```
plotCoeff(
  dat,
  datMCMC,
  estimator = "beta",
  intercept = FALSE,
  bandwidth = NULL,
  xlim = NULL,
  xlab = NULL,
  label.y = NULL,
  first.coef = NULL,
  y.axis.size = 8,
  ...
)
```

Arguments

dat	input data as a list containing survival data sub-list survObj with two vectors (event and time), clinical variable matrix x0, cluster-specific covariates X, and proportions data matrix proportion
datMCMC	returned object from the main function GPTCM()
estimator	print estimators, one of c("beta", "zeta", "gamma", "eta")
intercept	logical value to print intercepts
bandwidth	a value of bandwidth used for the ridgeplot
xlim	numeric vectors of length 2, giving the x-coordinate range.
xlab	a title for the x axis
label.y	a title for the y axis
first.coef	number of the first variables. Default NULL for all variables
y.axis.size	text size in pts
...	others

Value

A ggplot2::ggplot object. See ?ggplot2::ggplot for more details of the object.

References

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)

# run a Bayesian GPTCM model: GPTCM-Ber2
fit <- GPTCM(dat, nIter = 10, burnin = 0)

plotCoeff(dat, datMCMC = fit, estimator = "beta")
```

---

plotMCMC *MCMC trace-plots*

Description

Trace-plots of regression coefficients over MCMC iterations

Usage

```
plotMCMC(dat, datMCMC, estimator = "xi")
```

Arguments

dat	input data as a list containing survival data sub-list survObj with two vectors (event and time), clinical variable matrix x0, cluster-specific covariates X, and proportions data matrix proportion
datMCMC	returned object from the main function GPTCM()
estimator	print estimators, one of c("beta", "zeta", "gamma", "eta")

Value

A ggplot2::ggplot object. See ?ggplot2::ggplot for more details of the object.

References

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)

# run a Bayesian GPTCM model: GPTCM-Ber2
fit <- GPTCM(dat, nIter = 10, burnin = 0)

plotMCMC(dat, datMCMC = fit, estimator = "xi")
```

`predict.GPTCM` *Prediction of survival probability*

Description

Compute predicted survival probability for a GPTCM

Usage

```
## S3 method for class 'GPTCM'
predict(object, dat, newdata = NULL, type = "survival", times = NULL, ...)
```

Arguments

`object`	the results of a `GPTCM` fit
`dat`	the dataset used in `GPTCM()`
`newdata`	optional new data at which to do predictions. If missing, the prediction will be based on the training data
`type`	the type of predicted value. Currently it is only valid with `'survival'`
`times`	evaluation time points for survival prediction. Default `NULL` for predicting all time points in the `newdata` set
`...`	for future methods

Value

A matrix object

References

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)

# run a Bayesian GPTCM model: GPTCM-Ber2
fit <- GPTCM(dat, nIter = 10, burnin = 0)

pred.survival <- predict(fit, dat, newdata = dat, times = c(1, 3, 5))
```

---

run_mcmc *Main function implemented in C++ for the MCMC loop*

Description

Main function implemented in C++ for the MCMC loop

Usage

```
run_mcmc(
  nIter,
  burnin,
  thin,
  tick,
  n,
  nsamp,
  ninit,
  convex,
  npoint,
  dirichlet,
  proportion_model,
  BVS,
  threads,
  gamma_prior,
  gamma_sampler,
  eta_prior,
  eta_sampler,
  initList,
```

```
  rangeList,
  hyperparList,
  datEvent,
  datTime,
  datX,
  datX0,
  datProportionConst
)
```

Arguments

nIter	number of MCMC iterations
burnin	length of MCMC burn-in period
thin	number of thinning
tick	an integer used for printing the iteration index
n	number of samples to draw
nsamp	how many samples to draw for generating each sample; only the last draw will be kept
ninit	number of initials as meshgrid values for envelop search
convex	adjustment for convexity (non-negative value, default 1.0)
npoint	maximum number of envelope points
dirichlet	not yet implemented
proportion_model	logical value for modeling the proportions data
BVS	logical value for implementing Bayesian variable selection
threads	maximum threads used for parallelization. Default is 1
gamma_prior	one of `c("bernoulli", "MRF")`
gamma_sampler	one of `c("mc3", "bandit")`
eta_prior	one of `c("bernoulli", "MRF")`
eta_sampler	one of `c("mc3", "bandit")`
initList	a list of initial values for parameters "kappa", "xi", "betas", and "zetas"
rangeList	a list of ranges of initial values for parameters "kappa", "xi", "betas", and "zetas"
hyperparList	a list of relevant hyperparameters
datEvent	a vector of survival status
datTime	a vector of survival times
datX	an array of cluster-specific covariates
datX0	a matrix of mandatory variables
datProportionConst	an array of cluster-specific proportions

`simData` *Simulate data*

Description

Simulate survival data based on a GPTCM or Cox model

Usage

```
simData(
  n = 200,
  p = 10,
  L = 3,
  Sigma = NULL,
  kappas = 2,
  proportion.model = "dirichlet",
  model = "GPTCM"
)
```

Arguments

`n`	number of subjects
`p`	number of covariates in each cluster
`L`	number of clusters
`Sigma`	NULL (for a default covariance matrix) or "independent" (i.e. Sigma=diag(p*L)) or a self-defined matrix
`kappas`	value of the Weibull's shape parameter
`proportion.model`	One of `c("alr", "cloglog", "log", "dirichlet")`
`model`	one of `c("GPTCM", "Cox")`

Value

An object of a list with 12 components

- "`survObj`" - a list including events and times
- "`accepted`" - a vector with acceptance rates to generate each time-to-event data point by Metropolis-Hastings algorithm.
- "`proportion.model`" - value to indicate the model type for simulating proportions data.
- "`proportion`" - a matrix with simulated proportions data.
- "`kappas`" - value of the Weibull's shape parameter.
- "`x0`" - a matrix with the data of clinical variables
- "`X`" - an array with cluster-specific covariates
- "`xi`" - effects of clinical variables

- "beta0" - intercepts related to cluster-specific-survival.
- "betas" - effects related to cluster-specific-survival.
- "zetas" - effects related to cluster-specific-proportions.
- "mrfG" - a graph corresponding to the precision matrix of cluster-specific covariates
- "mrfG2" - a graph corresponding to every second edge in "mrfG"

References

Zhao Z, Kızılaslan F, Wang S, Zucknick M (2025). *Generalized promotion time cure model: A new modeling framework to identify cell-type-specific genes and improve survival prognosis*. arXiv:2509.01001

Examples

```
# simulate data
set.seed(123)
n <- 200 # subjects
p <- 10 # variable selection predictors
L <- 3 # cell types
dat <- simData(n, p, L)
str(dat)
```

`target` *Target density*

Description

Predefined target density corresponding to the population survival function of GPTCM

Usage

```
target(x, theta, proportion, mu, kappas)
```

Arguments

x	value generated from the proposal distribution
theta	cure rate parameter (log scale)
proportion	proportions data
mu	mean survival time
kappas	Weibull's true shape parameter

Value

value of the targeted (improper) probability density function

Examples

```
time1 <- target(1.2, 0.1, c(0.2, 0.3, 0.5), c(0.2, 0.1, 0.4), 2)
```

Index