Package 'iCARH'

October 13, 2022

Version 2.0.2.1

Date 2020-08-23

Title Integrative Conditional Autoregressive Horseshoe Model

Description Implements the integrative conditional autoregressive horseshoe model
discussed in Jendoubi, T., Ebbels, T.M. Integrative analy-
sis of time course metabolic data and biomarker discovery.
BMC Bioinformatics 21, 11 (2020) <doi:10.1186/s12859-019-3333-0>.
The model consists in three levels: Metabolic pathways level modeling interdependencies between variables via a conditional auto-regressive (CAR) component, integrative analysis level to identify potential associations between heterogeneous omic variables via a Horseshoe prior and experi-
mental
design level to capture experimental design conditions through a mixed-effects model.
The package also provides functions to simulate data from the model, construct pathway matrices, post process and plot model parameters.

Depends rstan, MASS, stats, ggplot2, glue

biocViews

Imports RCurl, KEGGgraph, igraph, reshape2, mc2d, abind, Matrix

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 3)

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Takoua Jendoubi [aut, cre],
Timothy M.D. Ebbels [aut]

Maintainer Takoua Jendoubi <t.jendoubi14@imperial.ac.uk>

Repository CRAN

Date/Publication 2020-08-27 07:50:07 UTC

R topics documented:

`iCARH.getBeta` *Return model parameters*

Description

Group of functions to return model parameters of interest

Usage

```
iCARH.getBeta(fit)

iCARH.getARCoeff(fit)

iCARH.getTreatmentEffect(fit)

iCARH.getPathwaysCoeff(fit, path.names = NULL)

iCARH.getDataImputation(fit)
```

Arguments

`fit`	Object returned by iCARH.model
`path.names`	pathway names

Value

the `iCARH.get[*]` functions return a an array with corresponding model parameters MCMC draws.

Functions

- `iCARH.getBeta`: Get beta parameter draws from all chains combined
- `iCARH.getARCoeff`: return theta coefficients
- `iCARH.getTreatmentEffect`: return alpha coefficients
- `iCARH.getPathwaysCoeff`: return phi coefficients
- `iCARH.getDataImputation`: return complete data (including imputed data)

Examples

```
data.sim = iCARH.simulate(4, 10, 14, 8, 2, path.probs=0.3, Zgroupeff=c(0,4),
beta.val=c(1,-1,0.5, -0.5))
XX = data.sim$XX
Y = data.sim$Y
Z = data.sim$Z
pathways = data.sim$pathways

rstan_options(auto_write = TRUE)
options(mc.cores = 2)
fit = iCARH.model(XX, Y, Z,groups=rep(c(0,1), each=5), pathways,
control = list(adapt_delta = 0.99, max_treedepth=10), iter = 2, chains = 2)
if(!is.null(fit$icarh))
iCARH.getBeta(fit)
```

`iCARH.getPathwaysMat` *Builds pathways adjacency matrices*

Description

Builds pathways adjacency matrices from specified KEGG identifiers. Returns a list of pathway adjacency matrices based on shortest paths.

Usage

```
iCARH.getPathwaysMat(keggid, org)
```

Arguments

`keggid`	KEGG identifiers as specified in KEGG. keggid is list that might contain multiple identifiers per metabolite.
`org`	organism

Value

list of pathway matrices based on shortest paths between two metabolites

Examples

```
keggid = list("C08363")
iCARH.getPathwaysMat(keggid, "rno")
gc()
keggid = list("Unk1", "C00350",c("C08363", "C01245"))
iCARH.getPathwaysMat(keggid, "rno")
gc()
```

iCARH.model *Runs the integrative CAR Horseshoe model*

Description

Infers treatment effects, association with heterogeneous omic variables, pathway perturbation among other parameters (e.g. time dependence). Regression coefficients (beta parameter) are initialized using a univariate regression ignoring time and metabolite dependence.

Usage

```
iCARH.model(
  X,
  Y = NULL,
  drug,
  groups = NULL,
  pathways,
  tau = 1.2,
  NA_value = -99999,
  init = T,
  ...
)
```

Arguments

`X`	the metabolomics time-course data with dimensions timepoints x observations x variables
`Y`	the additional omic time-course data with dimensions timepoints x observations x variables
`drug`	treatment effect. Could be either continuous (an administered drug or other external factor) or binary (cases vs controls). In the binary case the `groups` argument can be safely removed. NA values not allowed in `drug`. Dimensions are timepoints x observations
`groups`	grouping vector (binary). Use when `drug` is continuous.
`pathways`	pathway adjacency matrices as returned by iCARH.getPathwaysMat
`tau`	global sparsity parameter τ as in Jendoubi, T., & Ebbels, T. (2018)
`NA_value`	NA values are incompatible with stan. This is a wrapper to encode missing values in `X` and `Y`. NAs will be replaced by NA_value and will be inferred (only for X and Y data).
`init`	If `TRUE` use iCARH provided initialization function. Passed to Stan otherwise. Please see Stan manual on `init` possible values.
`...`	additional stan parameters

Value

stan object

Examples

```
data.sim = iCARH.simulate(4, 8, 10, 2, 2, path.probs=0.3, Zgroupeff=c(0,4),
beta.val=c(1,-1,0.5, -0.5))
XX = data.sim$XX
Y = data.sim$Y
Z = data.sim$Z
pathways = data.sim$pathways

rstan_options(auto_write = TRUE)
options(mc.cores = 2)
fit = iCARH.model(XX, Y, Z,groups=rep(c(0,1), each=4), pathways,
 control = list(adapt_delta = 0.99, max_treedepth=10), iter = 2, chains = 2)
```

`iCARH.params` *Summarize and return model parameters*

Description

Group of functions to summarize and return model parameters of interest

Usage

```
iCARH.params(
  fit,
  pars = c("theta", "alpha", "beta", "phi"),
  path.names = NULL,
  prob = 0.95,
  use_cache = TRUE,
  digits = 2,
  ...
)
```

Arguments

`fit`	Object returned by iCARH.model
`pars`	Parameters of interest ("theta","alpha","beta","phi"). All parameters by default.
`path.names`	Specify pathway names.
`prob`	Confidence level. Defaults to 0.95.
`use_cache`	passed to stan summary method.
`digits`	The number of significant digits for printing out the summary; defaults to 2. The effective sample size is always rounded to integers.
`...`	not used currently

Value

contain summaries for all chains. Included in the summaries are means, standard deviations (Est.Error), effective sample sizes (Eff.Sample), and split Rhats. Monte Carlo standard errors (MC.Error) are also reported.

Functions

- `iCARH.params`: Summary of model parameters

Examples

```
data.sim = iCARH.simulate(4, 10, 14, 8, 2, path.probs=0.3, Zgroupeff=c(0,4),
beta.val=c(1,-1,0.5, -0.5))
XX = data.sim$XX
Y = data.sim$Y
Z = data.sim$Z
pathways = data.sim$pathways

rstan_options(auto_write = TRUE)
options(mc.cores = 2)
fit = iCARH.model(XX, Y, Z, groups=rep(c(0,1), each=5), pathways,
control = list(adapt_delta = 0.99, max_treedepth=10), iter = 2, chains = 2)
if(!is.null(fit$icarh))
iCARH.params(fit)
```

`iCARH.plotBeta` *Postprocess and plot model parameters*

Description

Group of functions to postprocess and plot model parameters of interest, compute WAIC (Watanabe-Akaike Information Criterion) and MADs (Mean Absolute Deviation) for posterior predictive checks and check normality assumptions.

Usage

```
iCARH.plotBeta(fit, indx = TRUE, indy = TRUE)

iCARH.plotARCoeff(fit, indx = TRUE)

iCARH.plotTreatmentEffect(fit, indx = TRUE)

iCARH.plotPathwayPerturbation(fit, path.names, indpath = TRUE)

iCARH.plotDataImputation(fit, indx = T, indy = T, plotx = T, ploty = T, ...)
iCARH.checkRhats(fit)

iCARH.checkNormality(fit)

iCARH.waic(fit)

iCARH.mad(fit)
```

Arguments

`fit`	object returned by iCARH.model
`indx`	vector to specify X variables to plot. Selects all variables of X by default.
`indy`	vector to specify Y variables to plot. Selects all variables of Y by default.
`path.names`	pathway names
`indpath`	vector to specify pathways to plot. Selects all pathways by default.
`plotx`	plot X data imputation?
`ploty`	plot Y data imputation?
`...`	passed to ggplot2::geom_violin

Value

the `iCARH.plot[*]` functions return a ggplot graph object. `iCARH.checkNormality` returns the normalized data. `iCARH.waic` and `iCARH.mad` return corresponding waic (scalar) and mad (vector of $J * (J + 1)/2$) values. `iCARH.checkRhats` checks model convergence.

Functions

- `iCARH.plotBeta`: Plot boxplots of posterior densities of β coefficients.
- `iCARH.plotARCoeff`: Plot boxplots of posterior densities of theta (time effect) coefficients.
- `iCARH.plotTreatmentEffect`: Plot boxplots of posterior densities of treatment effect coefficients.
- `iCARH.plotPathwayPerturbation`: Plot posterior densities of pathway perturbation parameters
- `iCARH.plotDataImputation`: Plot imputed values
- `iCARH.checkRhats`: check model convergence and return Rhat coefficients
- `iCARH.checkNormality`: Check normality assumptions. Returns normalized data and performs quantile-quantile plot
- `iCARH.waic`: Compute Watanabe-Akaike Information Criterion (WAIC)
- `iCARH.mad`: Compute MADs (Mean Absolute Deviation) between true covariance matrix and inferred covariance matrix for posterior predictive checks

Examples

```
data.sim = iCARH.simulate(4, 10, 14, 8, 2, path.probs=0.3, Zgroupeff=c(0,4),
beta.val=c(1,-1,0.5, -0.5))
XX = data.sim$XX
Y = data.sim$Y
Z = data.sim$Z
pathways = data.sim$pathways

rstan_options(auto_write = TRUE)
options(mc.cores = 2)
fit = iCARH.model(XX, Y, Z, groups=rep(c(0,1), each=5), pathways,
control = list(adapt_delta = 0.99, max_treedepth=10), iter = 2, chains = 2)
if(!is.null(fit$icarh))
gplot = iCARH.plotBeta(fit, indx=1:3, indy=1:2)
```

`iCARH.simulate` *Simulates longitudinal data based on the iCARH model.*

Description

Simulates longitudinal data based on the iCARH model. Returns two types of datasets with relevant parameters (see below).

Usage

```
iCARH.simulate(
  Tp,
  N,
  J,
  P,
  K,
  path.names = NULL,
  path.probs = FALSE,
  pathway.perturb.ratio = 0.5,
  Ygroupeff = NULL,
  Zgroupeff = NULL,
  fe = 0,
  num.corr.y = 0,
  beta.val = NULL,
  sigma2 = 1,
  arz = 0.7,
  sdx = 0.01
)
```

Arguments

`Tp`	number of time points
`N`	number of samples (by default first N/2 controls and last N/2 cases)
`J`	number of metabolites
`P`	number of pathways (will probably change)
`K`	number of bacteria profiles (Y variables)
`path.names`	pathways to sample from as specified in KEGG. If not specified, path.probs will be considered.
`path.probs`	if TRUE, KEGG like density of pathways per metabolite is used to sample from. If scalar, path.probs is the expected ratio of metabolites in each pathway. Needs to be specified if path.names is not.
`pathway.perturb.ratio`	expected ratio of perturbed pathways
`Ygroupeff`	vector of 2xK variables (treatment effect on Y variables)
`Zgroupeff`	vector of 2 variables for treatment effect
`fe`	fixed effect
`num.corr.y`	number of correlated Y variables. The last num.corr.y will be highly correlated to the first num.corr.y variables
`beta.val`	beta values (regression coefficients) to sample from. Values will be randomly sampled if not specified.
`sigma2`	individual variance of metabolites
`arz`	autoregressive coefficient for treatment simulation
`sdx`	noise for autoregressive process, recommended value is 0.01

Value

list with the following objects :

`XX`	metabolomics data, X data
`Y`	additional omic data, Y data
`Z`	treatment
`beta`	effects of Y variables on X variables, column K+1 represents effect of treatment on X variables
`pathways`	pathway adjacency matrices
`path.perturb`	which pathways are perturbed?
`phi`	"spatial" dependence parameter, indicative of pathway perturbation
`arx`	autoregressive coefficients for X data
`ary`	autoregressive coefficients for Y data

Examples

```
 data.sim = iCARH.simulate(4, 8, 10, 2, 2, path.probs=0.3, Zgroupeff=c(0,4),
beta.val=c(1,-1,0.5, -0.5))
```

Index