Package 'MRCCC'

November 24, 2026

Type Package

Title Bayesian Mendelian Randomization for Causal Cell-Cell Communication

Version 1.0.0

Description Implements MR-CCC, a Bayesian Mendelian randomization framework for inferring causal ligand-receptor signalling between cell types from population-scale single-cell expression and genotype data. For each ligand-receptor-pathway triplet, cis-eQTL genotypes instrument ligand expression in the sender cell type and receptor expression in the receiver cell type. A working model with a ligand-by-receptor interaction is then fitted to downstream pathway activity by a blocked Gibbs sampler. A spike-and-slab prior on the causal block yields a posterior inclusion probability that serves as the communication score. The retained draws support credible intervals, sign probabilities, the receptor level at which the ligand effect reverses sign, and Bayesian false discovery rate control across many triplets. Inputs are taken from SummarizedExperiment objects of donor-level expression and genotype dosages, and cis-eQTL instruments can be selected within the package.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.6.0)

Imports methods, stats, utils, Rcpp, SummarizedExperiment, S4Vectors, GenomicRanges, IRanges, BiocParallel, ggplot2 (>= 3.4.0), withr (>= 2.5.0)

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle, SingleCellExperiment, scuttle, VariantAnnotation

VignetteBuilder knitr

biocViews Software, StatisticalMethod, Bayesian, GeneExpression, SingleCell, Genetics, SystemsBiology, GeneRegulation

URL https://github.com/bitansa/MRCCC

BugReports https://github.com/bitansa/MRCCC/issues

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

NeedsCompilation yes

Config/roxygen2/version 8.1.1

Collate 'AllClasses.R' 'AllGenerics.R' 'MRCCC-package.R' 'MRCCCData.R' 'mrccc.R' 'MRCCCFit-methods.R' 'RcppExports.R' 'bayesFDR.R' 'selectInstruments.R' 'buildTriplet.R' 'credibleIntervals.R' 'hyperparameters.R' 'internal.R' 'mrcccScreen.R' 'onek1kExample.R' 'plotEffectCurve.R' 'signProbabilities.R' 'signReversal.R' 'simulateMRCCC.R'

git_url https://git.bioconductor.org/packages/MRCCC

git_branch RELEASE_3_24

git_last_commit bba60b0

git_last_commit_date 2026-11-23

Repository Bioconductor 3.24

Date/Publication 2026-11-23

Author Bitan Sarkar [aut, cre] (ORCID: <https://orcid.org/0009-0000-5178-5934>), Yang Ni [aut] (ORCID: <https://orcid.org/0000-0003-0636-2363>), National Institutes of Health [fnd] (Grants R01GM148974 and R01GM163238)

Maintainer Bitan Sarkar <bitansarkar010899@gmail.com>

Contents

MRCCC-package . . . 3
bayesFDR . . . 4
bfdrSelect . . . 5
buildTriplet . . . 6
credibleIntervals . . . 8
donors . . . 9
mrccc . . . 10
MRCCCData . . . 13
MRCCCData-class . . . 15
MRCCCFit-class . . . 16
mrcccHyperparameters . . . 17
mrcccScreen . . . 18
MRCCCSummary-class . . . 19
onek1kExample . . . 20
pip . . . 21
plot,MRCCCFit,missing-method . . . 24
plotEffectCurve . . . 25
screenTable . . . 26
selectInstruments . . . 26
signProbabilities . . . 28
signReversal . . . 29
simulateMRCCC . . . 30
summary,MRCCCFit-method . . . 32

Index **34**

MRCCC-package *MRCCC: Bayesian Mendelian randomization for causal cell-cell communication*

Description

MRCCC implements MR-CCC, a Bayesian Mendelian randomization framework for inferring causal ligand-receptor signalling between cell types from population-scale single-cell expression and genotype data. For each ligand-receptor-pathway triplet, cis-eQTL genotypes of the same donors instrument ligand expression in the sender cell type and receptor expression in the receiver cell type. A working model with a ligand-by-receptor interaction is then fitted to downstream pathway activity by a blocked Gibbs sampler. A spike-and-slab prior on the causal block yields a posterior inclusion probability that serves as the communication score.

Workflow

`buildTriplet()` Assemble one triplet from donor-level `SummarizedExperiment` objects for the sender and receiver cell types and a genotype `RangedSummarizedExperiment`, selecting cis-eQTL instruments with `selectInstruments()`. `MRCCCData()` builds the same object from aligned vectors and matrices.

`mrccc()` Fit the model. Returns an MRCCCFit object.

MRCCCFit-accessors `pip()`, `effectSizes()`, `convergence()`, `instrumentStrength()`, `firstStageEffects()`, `posteriorMeans()`, `posteriorDraws()`, `inputScale()` and `fitSettings()`.

Posterior summaries `credibleIntervals()`, `signProbabilities()`, `signReversal()` and `plotEffectCurve()`.

`mrcccScreen()`, `screenTable()` Fit many triplets, in parallel through BiocParallel, and summarise them.

`bayesFDR()`, `bfdrSelect()` Bayesian false discovery rate control across many triplets.

`onek1kExample()` A real-data example from the OneK1K cohort.

`simulateMRCCC()` Simulated data in the layout of real data.

A worked analysis, on simulated and real data, is in `vignette("MRCCC", package = "MRCCC")`.

Author(s)

Maintainer: Bitan Sarkar <bitansarkar010899@gmail.com> (ORCID)

Authors:

- Bitan Sarkar <bitansarkar010899@gmail.com> (ORCID)
- Yang Ni (ORCID)

Other contributors:

- National Institutes of Health (Grants R01GM148974 and R01GM163238) [funder]

References

Sarkar B, Ni Y (2026). MR-CCC: Bayesian Mendelian Randomization for Causal Cell-Cell Communication. Bioinformatics (in press). arXiv:2604.23917.

See Also

Useful links:

- https://github.com/bitansa/MRCCC
- Report bugs at https://github.com/bitansa/MRCCC/issues

`bayesFDR` *Bayesian false discovery rate of nested PIP-ranked sets*

Description

For a vector of posterior inclusion probabilities from many ligand-receptor-pathway triplets, computes the Bayesian false discovery rate (bFDR) of every nested set obtained by declaring the top-k triplets by PIP, for $k = 1, \ldots, m$.

Usage

```
bayesFDR(pip)
```

Arguments

`pip`	Posterior inclusion probabilities, one per triplet: a numeric vector in $[0, 1]$, or a list of MRCCCFit objects. Missing values are not allowed.

Details

For a declared set S, the posterior expected number of false discoveries is $\sum_{j \in S}(1 - \mathrm{PIP}_j)$, so the Bayesian false discovery rate is

$$\mathrm{bFDR}(S) = |S|^{-1} \sum_{j \in S}(1 - \mathrm{PIP}_j).$$

Ranking the triplets by decreasing PIP and accumulating gives, for each possible cut-off, the bFDR incurred by declaring everything at or above it. Because the PIPs are posterior probabilities, this controls the expected proportion of false discoveries without a separate multiplicity correction. The value returned for triplet j is the bFDR of the smallest nested set that contains j. It is non-decreasing in the rank of j. Ties in PIP are broken by input order.

Value

A numeric vector of the same length as `pip`, in the order of the input, giving the bFDR of the nested set that ends at each triplet.

References

Newton MA, Noueiry A, Sarkar D, Ahlquist P (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. Biostatistics 5(2):155-176.

See Also

`bfdrSelect()`, `mrccc()`, `pip()`

Examples

```
pip <- c(0.9, 0.8, 0.5)
bayesFDR(pip)
# Unsorted input is handled. Results are returned in input order.
bayesFDR(c(0.5, 0.9, 0.8))
```

`bfdrSelect` *Largest PIP-ranked set with Bayesian FDR at most alpha*

Description

Selects the largest nested set of triplets, ranked by decreasing posterior inclusion probability, whose Bayesian false discovery rate does not exceed `alpha`. The rule is monotone: if the top-k set qualifies, so does every smaller top-k' set. If no set qualifies, nothing is selected.

Usage

```
bfdrSelect(pip, alpha)
```

Arguments

`pip`	Posterior inclusion probabilities, one per triplet: a numeric vector in $[0, 1]$, or a list of MRCCCFit objects.
`alpha`	Target Bayesian false discovery rate in $(0, 1)$.

Value

A logical vector of the same length as `pip`, in input order, that is `TRUE` for the selected triplets.

See Also

`bayesFDR()`

Examples

```
pip <- c(0.9, 0.8, 0.5, 0.2)
bayesFDR(pip)
bfdrSelect(pip, alpha = 0.2)
bfdrSelect(pip, alpha = 0.05)
```

buildTriplet *Build an MR-CCC triplet from SummarizedExperiment data*

Description

Assembles the MRCCCData object for one ligand-receptor-pathway triplet from donor-level (pseudo-bulk) expression of a sender and a receiver cell type and from donor genotypes. Donors are matched across the three objects by column name, covariates are taken from `colData`, the pathway activity score is computed from the receiver's expression of the pathway genes, and, unless supplied, the cis-eQTL instruments of the ligand and the receptor are chosen with `selectInstruments()`.

Usage

```
buildTriplet(
  sender,
  receiver,
  genotypes,
  ligand,
  receptor,
  pathway,
  covariates,
  ligandSNPs = NULL,
  receptorSNPs = NULL,
  assay = 1L,
  genotypeAssay = 1L,
  pathwayMethod = c("mean", "pc1"),
  pathwayName = NA_character_,
  senderName = NA_character_,
  receiverName = NA_character_,
  ...
)
```

Arguments

`sender`	A `SummarizedExperiment` (a `RangedSummarizedExperiment` when instruments are to be selected) of donor-level expression in the sender cell type, genes in rows and donors in columns.
`receiver`	A `SummarizedExperiment` of donor-level expression in the receiver cell type, laid out as `sender`.
`genotypes`	A `SummarizedExperiment` of allele dosages, SNPs in rows (with identifiers as row names) and donors in columns. A `RangedSummarizedExperiment` with SNP positions is required when instruments are to be selected. Dosages count copies of the alternative allele (0, 1 or 2, or an imputed value in between). Missing values are imputed by the SNP's mean dosage.
`ligand`	Name of the ligand gene (a row name of `sender`).
`receptor`	Name of the receptor gene (a row name of `receiver`).
`pathway`	The pathway activity, given in one of three ways. It can be a character vector of pathway genes, scored in `receiver` according to `pathwayMethod`. It can be the name of a numeric column of `colData(receiver)` that holds a precomputed

score, for example from AUCell, UCell or GSVA. It can also be a numeric vector of such a score named by donor. A single name that is both a gene of `receiver` and a `colData` column is read as a gene.

`covariates` Character vector naming donor covariates in `colData(receiver)` (or, failing that, `colData(sender)`), or a numeric matrix or data frame with donors as row names. Factors and character columns are expanded to indicator columns. At least one covariate is required. The same covariates adjust the instrument screen and the model.

`ligandSNPs`, `receptorSNPs`
Optional instruments: character vectors of SNP identifiers (row names of `genotypes`), or the `DataFrames` returned by `selectInstruments()`. When `NULL`, `selectInstruments()` is called.

`assay` Assay of `sender` and `receiver` holding the normalised expression: a name or an index.

`genotypeAssay` Assay of `genotypes` holding the dosages.

`pathwayMethod` How a character `pathway` is scored: `"mean"`, the mean expression of the pathway genes, or `"pc1"`, their first principal component (genes centred and scaled), oriented so that higher scores mean higher mean expression.

`pathwayName`, `senderName`, `receiverName`
Optional labels used in printed output.

`...` Further arguments passed to `selectInstruments()`, such as `window`, `maxSNPs` or `r2Max`.

Details

Only donors present in `sender`, `receiver` and `genotypes` are used, and a message reports how many were dropped. The instrument screens use the same donors and covariates as the model, so the analysis is one-sample. The selection tables are kept in the `metadata` slot of the result under `ligandInstruments` and `receptorInstruments`.

Value

An MRCCCData object.

See Also

`selectInstruments()`, `MRCCCData()`, `mrccc()`, `simulateMRCCC()`

Examples

```
sim <- simulateMRCCC(n = 120, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"),
                    pathwayName = "Simulated pathway")
dat
```

`credibleIntervals` *Credible intervals for the MR-CCC effect sizes*

Description

Computes equal-tailed posterior credible intervals for β_X, β_{XZ} and β_Z from the pooled retained draws of an MR-CCC fit.

Usage

```
credibleIntervals(x, ...)

## S4 method for signature 'MRCCCFit'
credibleIntervals(x, level = 0.95, standardized = TRUE, included = FALSE, ...)
```

Arguments

`x`	An MRCCCFit object returned by `mrccc()`.
`...`	Not used.
`level`	Credible level in $(0, 1)$. The default 0.95 gives the 2.5 and 97.5 percent posterior quantiles.
`standardized`	Logical. If `TRUE` (the default) the intervals are on the standardised scale (per standard deviation of ligand and receptor expression, in standard deviations of pathway activity). If `FALSE`, they are on the centred input scale.
`included`	Logical. If `FALSE` (the default) the intervals use all pooled draws. If `TRUE`, they use only the draws with $\gamma = 1$, which gives the intervals conditional on the causal block being included, as reported for the discoveries in the paper.

Details

The prior on (β_X, β_{XZ}) is a spike-and-slab mixture, so the marginal posterior of each of these coefficients mixes draws concentrated near zero ($\gamma = 0$) with slab draws ($\gamma = 1$). An interval that covers zero therefore reflects the posterior weight of the spike as much as the uncertainty within the slab. See `signProbabilities()` for a decomposition. With `included = TRUE` the spike draws are left out, so the intervals describe the size of the effects given that the ligand acts on the pathway.

Value

A data frame with columns `parameter` (`beta_X`, `beta_XZ`, `beta_Z`), `lower`, `median` and `upper`. With `included = TRUE` and no draw with $\gamma = 1$, the bounds are `NA`.

See Also

`mrccc()`, `effectSizes()`, `signProbabilities()`

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
credibleIntervals(fit)
credibleIntervals(fit, level = 0.9, standardized = FALSE)
credibleIntervals(fit, included = TRUE)
```

---

donors *Accessors for MRCCCData objects*

Description

Extract the components of an MRCCCData object. All components are aligned by donor.

Usage

```
donors(x)

ligandExpression(x)

receptorExpression(x)

pathwayActivity(x)

ligandGenotypes(x)

receptorGenotypes(x)

covariates(x)

tripletLabels(x)

## S4 method for signature 'MRCCCData'
donors(x)

## S4 method for signature 'MRCCCData'
ligandExpression(x)

## S4 method for signature 'MRCCCData'
receptorExpression(x)

## S4 method for signature 'MRCCCData'
pathwayActivity(x)

## S4 method for signature 'MRCCCData'
ligandGenotypes(x)
```

```
## S4 method for signature 'MRCCCData'
receptorGenotypes(x)

## S4 method for signature 'MRCCCData'
covariates(x)

## S4 method for signature 'MRCCCData'
tripletLabels(x)
```

Arguments

x	An MRCCCData object (or, for `tripletLabels()`, an MRCCCFit object).

Value

`donors()` returns the donor identifiers. `ligandExpression()`, `receptorExpression()` and `pathwayActivity()` return named numeric vectors. `ligandGenotypes()` and `receptorGenotypes()` return donors-by-SNPs dosage matrices. `covariates()` returns the donors-by-covariates matrix. `tripletLabels()` returns the named character vector of labels.

See Also

`MRCCCData()`, `buildTriplet()`

Examples

```
sim <- simulateMRCCC(n = 120, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
head(donors(dat))
colnames(ligandGenotypes(dat))
head(covariates(dat))
tripletLabels(dat)
```

---

mrccc *Fit the MR-CCC model to one ligand-receptor-pathway triplet*

Description

Fits the Bayesian Mendelian randomization model for causal cell-cell communication to one triplet stored in an MRCCCData object. The cis-eQTL genotypes instrument ligand expression in the sender cell type and receptor expression in the receiver cell type. The model asks whether the genetically predicted ligand level, modulated by the genetically predicted receptor level, shifts pathway activity in the receiver. Several Gibbs chains are run and pooled, and the posterior inclusion probability (PIP) is returned with its Monte Carlo standard error and convergence diagnostics.

Usage

```
mrccc(x, ...)

## S4 method for signature 'MRCCCData'
mrccc(
  x,
  nIter = 20000L,
  burnIn = 2000L,
  thin = 1L,
  nChains = 4L,
  initScale = 1,
  pipThreshold = 0.5,
  hyper = mrcccHyperparameters(),
  seed = NULL,
  verbose = FALSE,
  BPPARAM = BiocParallel::SerialParam(),
  ...
)
```

Arguments

x	An MRCCCData object, usually from `buildTriplet()`.
...	Not used.
nIter	Total number of Gibbs iterations per chain.
burnIn	Number of initial iterations discarded per chain. Must be smaller than `nIter`.
thin	Thinning interval: every `thin`-th post-burn-in iteration is retained.
nChains	Number of independent chains, pooled for all posterior summaries. With two or more chains the Gelman-Rubin statistic is reported and the chains start from dispersed values. Pooling `C` chains of `nIter` iterations gives roughly the Monte Carlo precision of one chain of `C * (nIter - burnIn)` retained draws. Each chain repeats the burn-in, which is the only extra cost.
initScale	Standard deviation of the normal distribution from which β_X and β_{XZ} are initialised, on the centred input scale, when `nChains > 1`. It must then be positive, so that the chains start from different values. The default 1 gives standard normal starting values. With a single chain both effects start at zero.
pipThreshold	Threshold in $(0, 1)$ that defines a discovery (`PIP > pipThreshold`). The default of 0.5 corresponds to the median probability model. Used for the borderline warning and by summary().
hyper	Prior hyperparameters, as returned by `mrcccHyperparameters()`.
seed	Optional integer. If supplied, the fit is reproducible: one seed per chain is derived from it, each chain runs under its own seed wherever it is evaluated, and the caller's random number state is restored afterwards. With the default `NULL`, the per-chain seeds are drawn from the current random number stream, so calling `set.seed()` beforehand also makes the fit reproducible.
verbose	Logical. If `TRUE`, each chain prints a progress line every 1000 iterations.
BPPARAM	A BiocParallel::BiocParallelParam object that controls how the chains are run. The default `BiocParallel::SerialParam()` runs them one after another. For example, `BiocParallel::MulticoreParam(4)` runs four chains at once. Because each chain has its own seed, the result does not depend on the back end.

Details

Model. For donor i, with $X^* = E[X \mid G, V]$ and $Z^* = E[Z \mid H, V]$ the genetically predicted ligand and receptor levels, the outcome model is

$$Y = \mu + \beta_X X^* + \beta_Z Z^* + \beta_{XZ} X^* Z^* + \alpha_Y^\top V + \tilde{\varepsilon}_Y,$$

where X is ligand expression, Z receptor expression, Y pathway activity, G and H the instrument dosages and V the covariates. The causal block (β_X, β_{XZ}) has a spike-and-slab prior with inclusion indicator $\gamma \in \{0, 1\}$. Under $\gamma = 0$ the prior variance of the block is multiplied by `nu1` (10^{-4} by default), so the coefficients are concentrated tightly at zero. The PIP $P(\gamma = 1 \mid \text{data})$ is the communication score. The receptor-modulated ligand effect is $\beta_X + \beta_{XZ} z$, which changes sign at $\tau = -\beta_X / \beta_{XZ}$. See `signReversal()`.

Centring and scale. Neither first stage carries an intercept, so the expression, pathway, genotype and covariate data are all centred internally. Effects are reported on the centred input scale and on a standardised scale (per standard deviation of X and Z, in standard deviations of Y). See `effectSizes()`.

Priors. The first-stage blocks and the covariate block carry Zellner g-priors on their designs. The second-stage blocks are scaled by a fixed diagonal computed once from the least-squares first stage, so every Gibbs update is an exact conjugate step. Unless overridden in `hyper`, every scale g is $\min(n, 100)$. See `mrcccHyperparameters()`.

Instrument strength. The first-stage partial F statistic of each instrument set, after the covariates, is reported by `instrumentStrength()`. A message is issued when either F is below 10. MR-CCC does not require F > 10, but PIPs are less decisive with weak instruments and longer chains may be needed.

Chains, Monte Carlo error and convergence. Odd-numbered chains start with $\gamma = 1$ and even-numbered chains with $\gamma = 0$. The Monte Carlo standard error (MCSE) of the PIP is estimated by non-overlapping batch means within each chain and combined across chains. It is `NA` when the indicator never changes value in any chain, so that the PIP is exactly 0 or 1. A warning is issued when $|\text{PIP} - t| < 2\,\text{MCSE}$ for threshold t, because the discovery status is then not resolved at the current chain length, and when any Gelman-Rubin $\hat{R}$ exceeds 1.01. $\hat{R}$ is reported for β_X, β_{XZ}, γ and the joint log-likelihood. It requires at least two chains. The default chain length is a starting point. Reported analyses should use chains long enough that neither warning is issued.

Which R-hat. $\hat{R}$ is the classic Gelman-Rubin potential scale reduction factor (Gelman and Rubin, 1992), without the correction for sampling variability, which is the statistic defined and reported in the paper. The threshold 1.01 is the one recommended for the rank-normalised split $\hat{R}$ of Vehtari et al. (2021). For the classic statistic, the conventional threshold is 1.1, so the check here is stricter than usual and errs on the side of longer chains.

Settings used in the paper. The real-data analysis ran four chains of 100,000 iterations per triplet with a burn-in of 2,000. The simulation studies ran one chain per fit, of 20,000 iterations when the instruments are strong (scenarios S1 to S3) and of 400,000 iterations under weak or otherwise misspecified instruments (scenarios S4 to S7), with a burn-in of 2,000. A comparison with MR-CCC should use chains of these lengths, or longer, for the corresponding setting.

Value

An MRCCCFit object. See MRCCCFit-accessors.

References

Sarkar B, Ni Y (2026). MR-CCC: Bayesian Mendelian Randomization for Causal Cell-Cell Communication. Bioinformatics (in press). arXiv:2604.23917.

Gelman A, Rubin DB (1992). Inference from iterative simulation using multiple sequences. Statistical Science 7(4):457-472.

Vehtari A, Gelman A, Simpson D, Carpenter B, Burkner P-C (2021). Rank-normalization, folding, and localization: an improved R-hat for assessing convergence of MCMC. Bayesian Analysis 16(2):667-718.

See Also

buildTriplet(), MRCCCFit-accessors, credibleIntervals(), signProbabilities(), signReversal(), plotEffectCurve(), mrcccScreen(), bayesFDR()

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
fit
pip(fit)
effectSizes(fit)
```

MRCCCData *Create an MRCCCData object from aligned donor-level data*

Description

Constructs the input object for one ligand-receptor-pathway triplet from donor-level vectors and matrices. Donors are matched by name rather than by position, so components supplied in a different donor order are aligned automatically and components that refer to different donors are rejected. Most users will create the object from SummarizedExperiment data with buildTriplet() instead.

Usage

```
MRCCCData(
  ligandExpression,
  receptorExpression,
  pathwayActivity,
  ligandGenotypes,
  receptorGenotypes,
  covariates,
  labels = NULL,
  metadata = list()
)
```

Arguments

ligandExpression

Numeric vector (or one-column matrix), one value per donor: ligand expression in the sender cell type, typically normalised pseudo-bulk expression.

`receptorExpression`
: Numeric vector: receptor expression in the receiver cell type.

`pathwayActivity`
: Numeric vector: activity score of the downstream pathway in the receiver cell type.

`ligandGenotypes`
: Numeric matrix or data frame, donors in rows and SNPs in columns: allele dosages (0, 1 or 2 copies of the alternative allele, or an imputed dosage between 0 and 2) of the cis-eQTL SNPs that instrument the ligand. See `selectInstruments()`.

`receptorGenotypes`
: Numeric matrix or data frame, donors by SNPs: allele dosages of the cis-eQTL SNPs that instrument the receptor.

`covariates`
: Numeric matrix or data frame, donors in rows: donor covariates such as sex, age and genotype principal components. Factor, character and logical columns of a data frame are expanded to indicator columns (treatment contrasts). At least one covariate is required.

`labels`
: Optional named character vector or list with any of the elements `ligand`, `receptor`, `pathway`, `sender` and `receiver`, used in printed output.

`metadata`
: Optional list of additional information, read back with `S4Vectors::metadata()`.

Details

Donor matching. If every component carries donor names (names of the vectors, row names of the matrices), the donors of `ligandExpression` define the order and every other component is reordered to match. A component that lacks any of these donors is an error. If no component carries donor names, all must have the same length and are taken to be in the same order, and donors are named `donor1`, `donor2`, and so on. Supplying names for some components but not others is an error.

Units. Genotypes are individual-level allele dosages of the same donors whose expression is analysed. MR-CCC is a one-sample method and does not use external eQTL summary statistics. Expression and pathway activity may be on any scale (for example log-normalised pseudo-bulk expression). `mrccc()` centres all inputs and reports effects on both the centred input scale and a standardised scale.

Value

An MRCCCData object.

See Also

`buildTriplet()`, `mrccc()`, MRCCCData-accessors

Examples

```
set.seed(1)
n <- 50
ids <- sprintf("D%02d", seq_len(n))
G <- matrix(rbinom(n * 3, 2, 0.3), n, 3,
            dimnames = list(ids, c("rs1", "rs2", "rs3")))
H <- matrix(rbinom(n * 2, 2, 0.4), n, 2,
            dimnames = list(ids, c("rs4", "rs5")))
cov <- data.frame(sex = sample(c("female", "male"), n, TRUE),
```

```
                  age = round(runif(n, 20, 80)), row.names = ids)
x <- setNames(drop(G %*% c(0.5, 0.3, 0.2)) + rnorm(n), ids)
z <- setNames(drop(H %*% c(0.4, 0.4)) + rnorm(n), ids)
y <- setNames(0.3 * x + 0.2 * x * z + rnorm(n), ids)
dat <- MRCCCData(ligandExpression = x, receptorExpression = z,
                 pathwayActivity = y, ligandGenotypes = G,
                 receptorGenotypes = H, covariates = cov,
                 labels = c(ligand = "LIG1", receptor = "REC1"))
dat
```

`MRCCCData-class` *The MRCCCData class: one ligand-receptor-pathway triplet*

Description

An `MRCCCData` object holds everything MR-CCC needs for one ligand-receptor-pathway triplet, aligned by donor: ligand expression in the sender cell type, receptor expression and pathway activity in the receiver cell type, the cis-eQTL genotypes that instrument the ligand and the receptor, and the donor covariates. Objects are normally created from `SummarizedExperiment` data with `buildTriplet()`, or from aligned vectors and matrices with the constructor `MRCCCData()`. The validity method checks that every component refers to the same donors in the same order, that genotypes are allele dosages in [0, 2], and that no value is missing.

Usage

```
## S4 method for signature 'MRCCCData'
show(object)
```

Arguments

`object`	An MRCCCData object.

Value

`show()` prints a description of the triplet and returns `object` invisibly.

Slots

`ligandExpression` Named numeric vector, one value per donor: ligand expression in the sender cell type.

`receptorExpression` Named numeric vector: receptor expression in the receiver cell type.

`pathwayActivity` Named numeric vector: pathway activity score in the receiver cell type.

`ligandGenotypes` Numeric matrix, donors by SNPs: allele dosages (0, 1 or 2 copies of the alternative allele, or an imputed value in between) of the cis-eQTL SNPs that instrument the ligand.

`receptorGenotypes` Numeric matrix, donors by SNPs: allele dosages of the cis-eQTL SNPs that instrument the receptor.

`covariates` Numeric matrix, donors by covariates: donor-level covariates (for example sex, age and genotype principal components), with factors already expanded to indicator columns.

`labels` Named character vector with elements `ligand`, `receptor`, `pathway`, `sender` and `receiver` (any may be `NA`), used in printed output.

`metadata` List of additional information, inherited from S4Vectors::Annotated and read with `S4Vectors::metadata()`. `buildTriplet()` records there the instrument-selection tables (`ligandInstruments`, `receptorInstruments`), the pathway genes (`pathwayGenes`) and how the pathway was scored (`pathwayMethod`).

See Also

`MRCCCData()`, `buildTriplet()`, `mrccc()`

`MRCCCFit-class` *The MRCCCFit class: a fitted MR-CCC model*

Description

An `MRCCCFit` object is returned by `mrccc()`. It stores the pooled posterior draws of the causal effects and the inclusion indicator, the posterior inclusion probability (PIP) with its Monte Carlo error, the convergence diagnostics, the first-stage instrument strength, and the settings of the run. Use the accessors rather than the slots: `pip()`, `pipMCSE()`, `effectSizes()`, `credibleIntervals()`, `signProbabilities()`, `signReversal()`, `convergence()`, `instrumentStrength()`, `firstStageEffects()`, `posteriorMeans()`, `posteriorDraws()`, `inputScale()`, `fitSettings()` and `plotEffectCurve()`.

Usage

```
## S4 method for signature 'MRCCCFit'
show(object)
```

Arguments

`object`	An MRCCCFit object.

Value

`show()` prints a short summary of the fit and returns `object` invisibly.

Slots

`pip` Posterior inclusion probability, pooled over chains.

`pipMCSE` Monte Carlo standard error of `pip`.

`pipESS` Effective sample size of the pooled indicator chain.

`draws` Data frame of retained draws, one row per draw, with columns `chain`, `beta_X`, `beta_XZ`, `beta_Z`, `gamma`, `mu` (intercept) and `loglik` (joint log-likelihood of the three equations), the coefficients on the centred input scale. Read with `posteriorDraws()`.

`scale` List with the standard deviations (`sdX`, `sdZ`, `sdY`) and means (`meanX`, `meanZ`, `meanY`) of ligand expression, receptor expression and pathway activity, and the observed receptor range in standard deviations from its mean (`zRangeStd`). Read with `inputScale()`.

`diagnostics` Data frame with columns `parameter` (`beta_X`, `beta_XZ`, `gamma`, `loglik`) and `rhat`, the Gelman-Rubin statistic (`NA` for a single chain). Read with `convergence()`.

`instruments` Data frame with one row per exposure and columns `exposure` (`ligand`, `receptor`), `nInstruments`, `F` (first-stage partial F statistic after the covariates) and `weak` (`F < 10`). Read with `instrumentStrength()`.

`firstStage` Data frame with one row per instrument and columns `exposure`, `snp`, `mean`, `lower` and `upper`: the posterior mean and 95% credible interval of the instrument's effect on expression per copy of the alternative allele. Read with `firstStageEffects()`.

`posteriorMeans` List of the posterior means of the remaining parameters: `covariateEffects` (data frame with columns `covariate`, `ligand`, `receptor` and `pathway`), `residualVariances` (named vector for the ligand, receptor and pathway equations), `inclusionProbabilityPrior` (ρ) and `intercept` (μ). Read with `posteriorMeans()`.

`settings` List of the run settings: `nIter`, `burnIn`, `thin`, `nChains`, `nKeep` (retained draws per chain), `pipThreshold`, `hyper` (the hyperparameters), `g` (the g-prior scale used), `initScale` and `priorScale` (the fixed second-stage prior scale `d_X`, `d_XZ`, `d_Z`). Read with `fitSettings()`.

`labels` Labels copied from the MRCCCData object.

See Also

`mrccc()`

`mrcccHyperparameters` *Hyperparameters of the MR-CCC prior*

Description

Constructs and validates the list of prior hyperparameters used by `mrccc()`. The defaults are those used throughout the MR-CCC analyses and are weakly informative for inputs on a scale of roughly unit variance: with several hundred donors the likelihood then dominates the inverse-gamma and beta priors. The inverse-gamma priors are not invariant to the units of `X`, `Z` and `Y`, so inputs on very different scales should be standardised before fitting.

Usage

```
mrcccHyperparameters(
  aSigma = 3,
  bSigma = 2,
  aRho = 3,
  bRho = 1,
  nu1 = 1e-04,
  g = NULL,
  ridge = 1e-08
)
```

Arguments

`aSigma`, `bSigma`	Shape and scale of the inverse-gamma prior on the three residual variances $\sigma^2_X, \sigma^2_Z, \sigma^2_Y$. Both must be positive.
`aRho`, `bRho`	Shape parameters of the beta prior on the prior inclusion probability ρ. Both must be positive.
`nu1`	Spike variance multiplier in $(0, 1)$. The spike component of the prior on (β_X, β_{XZ}) has the slab covariance scaled by `nu1`, so small values enforce a near-zero spike.

g	Common g-prior scale applied to every coefficient block (first-stage sender effects, first-stage receiver effects, covariate effects, receptor main effect, and the causal block). If NULL (the default), mrccc() sets $g = \min(n, 100)$, where n is the number of donors. A positive number overrides this rule.
ridge	Positive diagonal ridge added before every matrix inversion for numerical stability.

Details

Each first-stage block θ with observed design D (the genotype matrices G, H or the covariates V) has the Zellner g-prior $\theta \sim N(0, g\,\sigma^2(D^\top D)^{-1})$. The second-stage blocks (β_X, β_{XZ}) and β_Z use the same scale g but against a fixed diagonal computed once from the least-squares first stage, because their regressors X^* and Z^* are themselves parameters of the model. See the *Priors* section of mrccc(). The default $g = \min(n, 100)$ follows the unit-information convention for small samples and caps the prior dispersion for large ones. The spike-and-slab prior on the causal block uses scale g in the slab and $\nu_1 g$ in the spike.

Value

A named list of class mrcccHyperparameters with elements aSigma, bSigma, aRho, bRho, nu1, g and ridge.

See Also

mrccc()

Examples

```
hyper <- mrcccHyperparameters()
str(hyper)

# A wider spike (less strict shrinkage under gamma = 0) and a fixed
# g-prior scale of 50, which narrows the slab when there are more than
# 50 donors
mrcccHyperparameters(nu1 = 1e-3, g = 50)
```

---

mrcccScreen *Fit MR-CCC to many triplets*

Description

Fits mrccc() to every triplet in a list and returns the fits, ready for Bayesian false discovery rate control with bayesFDR(), bfdrSelect() or screenTable(). Triplets are fitted in parallel through BiocParallel, and the chains within each triplet are run one after another.

Usage

```
mrcccScreen(triplets, ..., seed = NULL, BPPARAM = BiocParallel::SerialParam())
```

Arguments

triplets	A list of MRCCCData objects, usually named. Each is typically made with buildTriplet().
...	Further arguments passed to mrccc(), such as nIter, burnIn, nChains or hyper.
seed	Optional integer. If supplied, one seed per triplet is derived from it, so the screen is reproducible whatever the back end.
BPPARAM	A BiocParallel::BiocParallelParam object that controls how the triplets are distributed. The default BiocParallel::SerialParam() fits them one after another.

Value

A list of MRCCCFit objects with the names of triplets. When triplets is unnamed, the names are built from the ligand and receptor labels.

See Also

screenTable(), bayesFDR(), bfdrSelect(), mrccc()

Examples

```
sims <- lapply(c(1, 2), function(s) simulateMRCCC(n = 200, seed = s))
triplets <- lapply(sims, function(sim) {
    buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                 ligand = "LIG1", receptor = "REC1",
                 pathway = sim$pathwayGenes,
                 covariates = c("sex", "age", "PC1", "PC2"))
})
names(triplets) <- c("replicate1", "replicate2")
fits <- mrcccScreen(triplets, nIter = 4000, burnIn = 1000, seed = 1)
screenTable(fits)
```

MRCCCSummary-class *The MRCCCSummary class: a printed summary of an MR-CCC fit*

Description

Returned by summary(). Printing it shows the posterior inclusion probability with its Monte Carlo error, the standardised effects with credible intervals, the instrument strength and the convergence diagnostics. as.data.frame() returns the table of effects.

Usage

```
## S4 method for signature 'MRCCCSummary'
show(object)

## S3 method for class 'MRCCCSummary'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`object`	An MRCCCSummary object.
`x`	An MRCCCSummary object.
`row.names, optional`	Not used.
`...`	Not used.

Value

`show()` prints the summary and returns `object` invisibly.

`as.data.frame()` returns the table of effects.

Slots

`table` Data frame with one row per effect and columns `parameter` (`beta_X`, `beta_XZ`, `beta_Z`), `mean`, `lower` and `upper` (posterior mean and credible interval on the standardised scale) and `rhat`. The `rhat` of `beta_Z` is `NA` because R-hat is computed for `beta_X`, `beta_XZ`, `gamma` and the log-likelihood only.

`diagnostics` Data frame of Gelman-Rubin statistics, as returned by `convergence()`.

`instruments` Data frame of first-stage instrument strength, as returned by `instrumentStrength()`.

`info` List with `title` (the triplet labels), `nChains`, `nKeep`, `pip`, `pipMCSE`, `pipESS`, `pipThreshold` and `level` (the credible level).

See Also

summary,MRCCCFit-method

---

onek1kExample *Real-data example: OneK1K NK cells to monocytes*

Description

Loads a small subset of the OneK1K cohort (Yazar et al., 2022) that ships with the package, for demonstrating MR-CCC on real data. It covers the NK cells (sender) to monocytes (receiver) axis analysed in the MR-CCC paper, restricted to the ligand-receptor pairs that CellPhoneDB classifies as "Signaling by Interleukin".

Usage

```
onek1kExample()
```

Details

Provenance. The data were derived from the files deposited on Zenodo for the MR-CCC paper (doi:10.5281/zenodo.19675075) by the script `inst/scripts/make-onek1k-example.R`, which can be found with `system.file("scripts", "make-onek1k-example.R", package = "MRCCC")`. Donors pass the quality control of the paper. Expression is donor-level (pseudo-bulk) counts divided by a library-size factor, as in the paper. Genotypes are allele dosages of the SNPs within 200 kb of the transcription start site of each ligand and receptor gene that are carried by more than 5% of

donors. The receiver `colData` holds the donor covariates used in the paper (`sex`, `ageBin` for age in seven groups coded 1 to 7, namely under 30, five 10-year bins and 80 or over, and the genotype principal components `PC1`, `PC2` and `PC3`) and two activity scores of the Reactome interleukin gene set: `interleukinPC1`, the first principal component of the genes (centred, scaled, and oriented so that higher scores mean higher mean expression), and `interleukinMean`, their mean expression.

Value

A list with elements

`sender` `RangedSummarizedExperiment` of NK-cell expression of the ligand genes (assay `normcounts`), donors in columns.

`receiver` `RangedSummarizedExperiment` of monocyte expression of the receptor genes (assay `normcounts`), with the covariates and pathway scores in `colData`.

`genotypes` `RangedSummarizedExperiment` of allele dosages (assay `dosage`), SNPs in rows.

`triplets` Data frame of the ligand-receptor-pathway triplets with instruments for both genes, with columns `ligand`, `receptor` and `pathway`.

References

Yazar S, Alquicira-Hernandez J, Wing K, et al. (2022). Single-cell eQTL mapping identifies cell type-specific genetic control of autoimmune disease. Science 376(6589):eabf3041.

Sarkar B, Ni Y (2026). MR-CCC: Bayesian Mendelian Randomization for Causal Cell-Cell Communication. Bioinformatics (in press). arXiv:2604.23917.

See Also

`buildTriplet()`, `mrcccScreen()`

Examples

```
ex <- onek1kExample()
ex$receiver
ex$triplets
```

`pip` *Accessors for MRCCCFit objects*

Description

Extract results from a fitted MR-CCC model.

Usage

```
pip(x)

pipMCSE(x)

effectSizes(x, ...)
```

```
convergence(x)

instrumentStrength(x)

posteriorDraws(x, ...)

firstStageEffects(x)

posteriorMeans(x)

inputScale(x)

fitSettings(x)

## S4 method for signature 'MRCCCFit'
pip(x)

## S4 method for signature 'MRCCCFit'
pipMCSE(x)

## S4 method for signature 'MRCCCFit'
effectSizes(x, standardized = TRUE, ...)

## S4 method for signature 'MRCCCFit'
convergence(x)

## S4 method for signature 'MRCCCFit'
instrumentStrength(x)

## S4 method for signature 'MRCCCFit'
posteriorDraws(x, standardized = FALSE, ...)

## S4 method for signature 'MRCCCFit'
tripletLabels(x)

## S4 method for signature 'MRCCCFit'
firstStageEffects(x)

## S4 method for signature 'MRCCCFit'
posteriorMeans(x)

## S4 method for signature 'MRCCCFit'
inputScale(x)

## S4 method for signature 'MRCCCFit'
fitSettings(x)
```

Arguments

x	An MRCCCFit object returned by `mrccc()`.
...	Not used.
standardized	Logical. If `TRUE` (the default for `effectSizes()`) coefficients are on the stan-

dardised scale (per standard deviation of ligand and receptor expression, in standard deviations of pathway activity). If FALSE they are on the centred input scale.

Value

- pip(): the posterior inclusion probability, a number in [0, 1].
- pipMCSE(): a named numeric vector with the Monte Carlo standard error (mcse) and effective sample size (ess) of the PIP.
- effectSizes(): a data frame with columns parameter and mean, the posterior mean. The parameters are beta_X, the ligand effect at the mean receptor level, beta_XZ, the ligand-by-receptor interaction, and beta_Z, the receptor main effect.
- convergence(): a data frame of Gelman-Rubin statistics (rhat) for beta_X, beta_XZ, gamma and the joint log-likelihood. The values are NA for a single chain.
- instrumentStrength(): a data frame with one row per exposure (ligand, receptor) and columns nInstruments, F (first-stage partial F statistic after the covariates) and weak (F < 10).
- posteriorDraws(): a data frame of the pooled retained draws with columns chain, beta_X, beta_XZ, beta_Z, gamma, mu and loglik. With standardized = TRUE the three coefficients are rescaled.
- firstStageEffects(): a data frame with one row per instrument and columns exposure (ligand or receptor), snp, mean, lower and upper: the posterior mean and 95% credible interval of the effect of one copy of the alternative allele on expression.
- posteriorMeans(): a list with the posterior means of the remaining parameters: covariateEffects (data frame with columns covariate, ligand, receptor and pathway), residualVariances, inclusionProbabilityPrior and intercept.
- inputScale(): a list with the standard deviations (sdX, sdZ, sdY) and means (meanX, meanZ, meanY) of ligand expression, receptor expression and pathway activity, and the observed receptor range in standard deviations from its mean (zRangeStd).
- fitSettings(): a list of the run settings (nIter, burnIn, thin, nChains, nKeep, pipThreshold, hyper, g, initScale and priorScale).
- tripletLabels(): the labels of the triplet.

See Also

mrccc(), credibleIntervals(), signProbabilities(), signReversal(), plotEffectCurve()

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
pip(fit)
pipMCSE(fit)
effectSizes(fit)
convergence(fit)
instrumentStrength(fit)
head(posteriorDraws(fit))
```

```
firstStageEffects(fit)
posteriorMeans(fit)$residualVariances
fitSettings(fit)$nKeep
```

`plot,MRCCCFit,missing-method`
Plot an MR-CCC fit

Description

Plots the receptor-modulated ligand effect with its credible band. See `plotEffectCurve()`.

Usage

```
## S4 method for signature 'MRCCCFit,missing'
plot(x, y, ...)
```

Arguments

`x`	An MRCCCFit object.
`y`	Not used.
`...`	Further arguments passed to `plotEffectCurve()`.

Value

A `ggplot` object.

See Also

`plotEffectCurve()`

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
plot(fit)
```

`plotEffectCurve` *Plot the receptor-modulated ligand effect with a credible band*

Description

Draws the posterior mean of the receptor-modulated ligand effect $\beta_X + \beta_{XZ} z$ over the observed range of receptor expression, with a pointwise credible band from paired posterior draws.

Usage

```
plotEffectCurve(x, ...)

## S4 method for signature 'MRCCCFit'
plotEffectCurve(x, level = 0.95, nDraws = 2000L, standardized = TRUE, ...)
```

Arguments

`x`	An MRCCCFit object returned by `mrccc()`.
`...`	Not used.
`level`	Credible level in $(0, 1)$ of the pointwise band.
`nDraws`	Maximum number of paired draws used for the band. A systematic subsample of the pooled draws of this size is taken.
`standardized`	Logical. If `TRUE` (the default) the effect is on the standardised scale against receptor expression in standard deviations from its mean. If `FALSE`, it is on the centred input scale.

Details

At each receptor level z the band is the equal-tailed interval of $\beta_X + \beta_{XZ} z$ across paired draws, which accounts for the posterior correlation of the two coefficients. The draws include both spike and slab iterations. A dashed line marks an effect of zero. Where the band crosses it, the direction of the ligand effect is uncertain.

Value

A `ggplot` object.

See Also

`mrccc()`, `signReversal()`

Examples

```
sim <- simulateMRCCC(n = 200, betaX = 0.3, betaXZ = -0.3, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
plotEffectCurve(fit)
```

screenTable *Summarise a screen of many triplets*

Description

Collects the posterior inclusion probability, its Monte Carlo error and the largest Gelman-Rubin statistic of every fit, and applies Bayesian false discovery rate control across the triplets.

Usage

```
screenTable(fits, alpha = 0.1)
```

Arguments

fits	A list of MRCCCFit objects, as returned by mrcccScreen().
alpha	Target Bayesian false discovery rate for the selected set.

Value

A data frame with one row per fit, ordered by decreasing PIP, and columns triplet (the name of the fit), ligand, receptor, pathway, pip, pipMCSE, maxRhat (the largest Gelman-Rubin statistic, NA for a single chain), bFDR (the Bayesian false discovery rate of the PIP-ranked set ending at that triplet) and selected (the largest set with Bayesian FDR at most alpha).

See Also

mrcccScreen(), bayesFDR(), bfdrSelect()

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fits <- mrcccScreen(list(example = dat), nIter = 4000, burnIn = 1000,
                    seed = 1)
screenTable(fits, alpha = 0.1)
```

selectInstruments *Select cis-eQTL instruments for one gene*

Description

Selects the cis-eQTL SNPs that serve as instrumental variables for the expression of one gene (a ligand in the sender cell type or a receptor in the receiver cell type), following the procedure of the MR-CCC paper: SNPs in a window around the gene's promoter are screened in-sample by a covariate-adjusted regression of the gene's donor-level expression on allele dosage, ranked by p-value, and retained greedily subject to a linkage-disequilibrium (LD) bound until at most maxSNPs are kept.

Usage

```
selectInstruments(
  expression,
  gene,
  genotypes,
  covariates = NULL,
  assay = 1L,
  genotypeAssay = 1L,
  window = 2e+05,
  maxSNPs = 10L,
  r2Max = 0.8,
  minCarrierFreq = 0.05,
  minCandidates = 1L
)
```

Arguments

`expression`	A `RangedSummarizedExperiment` of donor-level (pseudo-bulk) expression for one cell type, genes in rows and donors in columns. Its `rowRanges` must give the genomic location of `gene`.
`gene`	Name of the gene (a row name of `expression`).
`genotypes`	A `RangedSummarizedExperiment` of genotypes, SNPs in rows and donors in columns, with SNP identifiers as row names and SNP positions as `rowRanges`. The assay holds allele dosages: the number of copies of the alternative allele (0, 1 or 2), or an imputed dosage in between. Missing values are allowed.
`covariates`	Donor covariates to adjust the screen for: either a character vector naming columns of `colData(expression)`, or a numeric matrix or data frame with donors as row names. `NULL` adjusts for an intercept only. Use the same covariates as in the model.
`assay`	Assay of `expression` to use: a name or an index.
`genotypeAssay`	Assay of `genotypes` to use: a name or an index.
`window`	Half-width in base pairs of the window around the transcription start site in which candidate SNPs are sought. The default 200 kb is the paper's setting.
`maxSNPs`	Maximum number of instruments retained.
`r2Max`	Maximum squared dosage correlation (LD r^2) between two retained instruments. A value of 1 disables clumping.
`minCarrierFreq`	SNPs carried (dosage of at least 0.5) by a fraction of donors no greater than this are removed before screening.
`minCandidates`	Minimum number of SNPs with a valid screening p-value required. With fewer, no instruments are returned.

Details

Only donors present in both `expression` and `genotypes` (and with covariates, when supplied as a matrix) are used. The steps are:

1. *Window.* Candidate SNPs lie within `window` base pairs upstream or downstream of the transcription start site of `gene`, as given by `IRanges::promoters()` applied to its range.
2. *Carrier filter.* SNPs carried by at most `minCarrierFreq` of the donors with a called genotype are removed. Remaining missing dosages are imputed by the SNP's mean dosage.

3. *Screen.* For each candidate the gene's expression is regressed on the dosage and the covariates (with an intercept), and the two-sided p-value of the dosage coefficient is computed. SNPs whose dosage is constant or collinear with the covariates are skipped.
4. *LD clumping.* SNPs are visited in order of increasing p-value. A SNP is kept if its squared dosage correlation with every SNP already kept is at most `r2Max`, until `maxSNPs` are kept. The strongest SNP is always kept.

No p-value threshold is applied: the strength of the selected set is assessed in the model through the first-stage F statistic reported by `instrumentStrength()`. Because the screen uses the same donors as the model, MR-CCC is a one-sample analysis.

Value

A `DataFrame` with one row per selected SNP, in the order of selection, and columns `snp` (identifier), `estimate` (screening coefficient per copy of the alternative allele) and `pvalue`. Its `metadata()` records the gene, the window, the number of donors, the number of SNPs in the window and the number screened. When fewer than `minCandidates` SNPs can be screened, a zero-row `DataFrame` is returned with a message.

References

Sarkar B, Ni Y (2026). MR-CCC: Bayesian Mendelian Randomization for Causal Cell-Cell Communication. Bioinformatics (in press). arXiv:2604.23917.

See Also

`buildTriplet()`, which calls this function for the ligand and the receptor.

Examples

```
sim <- simulateMRCCC(n = 120, seed = 1)
sel <- selectInstruments(sim$sender, "LIG1", sim$genotypes,
                         covariates = c("sex", "age", "PC1", "PC2"))
sel
S4Vectors::metadata(sel)
```

---

signProbabilities *Posterior sign probabilities of the interaction effect*

Description

Computes the posterior probability that the ligand-by-receptor interaction β_{XZ} is positive and that it is negative, both marginally and conditionally on the causal block being included ($\gamma = 1$).

Usage

```
signProbabilities(x, ...)

## S4 method for signature 'MRCCCFit'
signProbabilities(x, ...)
```

Arguments

x	An MRCCCFit object returned by `mrccc()`.
...	Not used.

Details

Under the spike-and-slab prior, the marginal posterior of β_{XZ} mixes draws taken while $\gamma = 0$, which are concentrated near zero and carry no directional information, with slab draws taken while $\gamma = 1$.

- The **marginal** probabilities `pPositive` and `pNegative` use every retained draw. They combine the evidence that there is an effect with the evidence about its direction: $P(\beta_{XZ} > 0) = \text{PIP} \cdot P(\beta_{XZ} > 0 \mid \gamma = 1) + (1 - \text{PIP}) \cdot P(\beta_{XZ} > 0 \mid \gamma = 0)$, and the last factor is close to one half.
- The **conditional** probabilities `pPositiveIncluded` and `pNegativeIncluded` use only the draws with $\gamma = 1$. They describe the direction of the interaction given that the ligand acts on the pathway. They are `NA` when no draw has $\gamma = 1$.

Value

A one-row data frame with columns `pip`, `pPositive`, `pNegative`, `pPositiveIncluded`, `pNegativeIncluded` and `nIncluded` (the number of pooled draws with $\gamma = 1$).

See Also

`mrccc()`, `credibleIntervals()`, `signReversal()`

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
signProbabilities(fit)
```

---

signReversal *Receptor level at which the ligand effect reverses sign*

Description

Computes the posterior distribution of the sign-reversal threshold $\tau = -\beta_X / \beta_{XZ}$, the receptor expression level at which the receptor-modulated ligand effect $\beta_X + \beta_{XZ} z$ changes sign, from paired posterior draws.

Usage

```
signReversal(x, ...)

## S4 method for signature 'MRCCCFit'
signReversal(x, level = 0.95, ...)
```

Arguments

x	An MRCCCFit object returned by mrccc().
...	Not used.
level	Credible level in $(0, 1)$ for the reported interval.

Details

τ is a ratio of two correlated parameters, so its posterior is computed from paired draws of (β_X, β_{XZ}). Draws whose standardised interaction is negligible, $|\beta_{XZ}|\,\mathrm{sd}(X)\,\mathrm{sd}(Z)/\mathrm{sd}(Y) \le 10^{-8}$, are dropped, since τ diverges there. All remaining draws are used, including those from the spike, so for a fit with a PIP well below one the threshold is best read together with the PIP.

Scale. Receptor expression is centred before fitting, so τ is reported in standard deviations of receptor expression relative to its mean over donors: $\tau = 0$ is a donor with average receptor expression and $\tau = 1$ a donor one standard deviation above average. The observed receptor range, on the same scale, is returned as zMin and zMax, and pInRange is the posterior probability that τ lies inside it, which determines whether the reversal is realised among the donors. The posterior of a ratio is heavy-tailed, so the median and quantiles are reported rather than a mean. Note that the interaction acts on the genetically predicted receptor level, whereas the range refers to observed receptor expression.

Value

A one-row data frame with columns tauMedian, tauLower, tauUpper (equal-tailed credible bounds at level), level, nDraws (paired draws used), zMin, zMax and pInRange. The numeric summaries are NA when no usable draw is available.

See Also

mrccc(), plotEffectCurve(), signProbabilities()

Examples

```
sim <- simulateMRCCC(n = 200, betaX = 0.3, betaXZ = -0.3, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
signReversal(fit)
```

---

simulateMRCCC *Simulate expression and genotype data for one triplet*

Description

Generates donor-level pseudo-bulk expression for a sender and a receiver cell type and donor genotypes, stored as RangedSummarizedExperiment objects, under the MR-CCC data-generating process with an unmeasured confounder. The output has the layout of real data and can be passed to selectInstruments() and buildTriplet().

Usage

```
simulateMRCCC(
  n = 300L,
  nInstruments = 4L,
  betaX = 0.3,
  betaXZ = 0.3,
  betaZ = 0.5,
  instrumentEffect = 0.8,
  confounding = 0.7,
  nPathwayGenes = 8L,
  seed = NULL
)
```

Arguments

`n`	Number of donors.
`nInstruments`	Number of causal cis-eQTL SNPs for the ligand and for the receptor.
`betaX`	True ligand effect on the pathway at the mean receptor level.
`betaXZ`	True ligand-by-receptor interaction effect.
`betaZ`	True receptor main effect on the pathway.
`instrumentEffect`	Effect on expression of one standard deviation of each causal SNP's dosage (instrument strength).
`confounding`	Loading of an unmeasured confounder on ligand expression, receptor expression and pathway activity. Zero removes confounding.
`nPathwayGenes`	Number of genes in the downstream pathway.
`seed`	Optional integer. If supplied, the data are simulated under this seed and the caller's random number state is restored afterwards.

Details

Donors have covariates `sex`, `age` and two genotype principal components `PC1` and `PC2`, stored in the `colData` of both expression objects. Genotypes are allele dosages (0, 1 or 2 copies of the alternative allele). Around each of the ligand gene `LIG1` (chromosome 1, plus strand) and the receptor gene `REC1` (chromosome 2, minus strand) the genotype object contains `nInstruments` causal SNPs, one strongly linked proxy for each causal SNP, six SNPs with no effect, one rare SNP and, outside the 200 kb window, five distant SNPs. With U the confounder and $\tilde{G}, \tilde{H}$ the standardised causal dosages,

$$X = \tilde{G}\pi + V\alpha + cU + \varepsilon_X, \quad Z = \tilde{H}\pi + V\alpha + cU + \varepsilon_Z,$$

$$Y = \beta_X X + \beta_Z Z + \beta_{XZ} XZ + V\alpha + cU + \varepsilon_Y,$$

where every element of π is `instrumentEffect`, every element of α is 0.3, c is `confounding` and the errors are standard normal. The sender expresses `LIG1` at $5 + X$. The receiver expresses `REC1` at $4 + Z$ and each pathway gene at $6 + Y$ plus independent noise with standard deviation 0.5, so the mean expression of the pathway genes tracks Y. Both objects also contain unrelated genes. Setting `betaX = betaXZ = 0` gives a null triplet.

Value

A list with elements

`sender` `RangedSummarizedExperiment` of sender expression (assay `logcounts`).

`receiver` `RangedSummarizedExperiment` of receiver expression (assay `logcounts`).

`genotypes` `RangedSummarizedExperiment` of allele dosages (assay `dosage`), SNPs in rows.

`pathwayGenes` Names of the pathway genes.

`truth` List with the generating values `betaX`, `betaXZ`, `betaZ`, the causal SNP identifiers `ligandCausal` and `receptorCausal`, and `tauStd`, the true sign-reversal threshold $-\beta_X/\beta_{XZ}$ in standard deviations of receptor expression from its mean (`NA` when `betaXZ` is zero).

See Also

`buildTriplet()`, `selectInstruments()`, `mrccc()`

Examples

```
sim <- simulateMRCCC(n = 120, seed = 1)
sim$sender
sim$genotypes
sim$truth$betaXZ
```

`summary,MRCCCFit-method`

Summarise an MR-CCC fit

Description

Collects the posterior inclusion probability with its Monte Carlo error, the standardised posterior means and credible intervals of the three effect sizes, the instrument strength and the convergence diagnostics.

Usage

```
## S4 method for signature 'MRCCCFit'
summary(object, level = 0.95, ...)
```

Arguments

`object`	An MRCCCFit object returned by `mrccc()`.
`level`	Credible level of the reported intervals.
`...`	Not used.

Value

An MRCCCSummary object, which prints the summary. `as.data.frame()` extracts its table, with one row per effect (`beta_X`, `beta_XZ`, `beta_Z`) and columns `mean`, `lower` and `upper` on the standardised scale, and `rhat`.

See Also

mrccc(), credibleIntervals()

Examples

```
sim <- simulateMRCCC(n = 200, seed = 1)
dat <- buildTriplet(sim$sender, sim$receiver, sim$genotypes,
                    ligand = "LIG1", receptor = "REC1",
                    pathway = sim$pathwayGenes,
                    covariates = c("sex", "age", "PC1", "PC2"))
fit <- mrccc(dat, nIter = 4000, burnIn = 1000, seed = 1)
s <- summary(fit)
s
as.data.frame(s)
```

Index

∗ **internal**
MRCCC-package, 3

as.data.frame.MRCCCSummary
(MRCCCSummary-class), 19

bayesFDR, 4
bayesFDR(), *3*, *5*, *13*, *18, 19*, *26*
bfdrSelect, 5
bfdrSelect(), *3, 4*, *18, 19*, *26*
BiocParallel::BiocParallelParam, *11*, *19*
BiocParallel::SerialParam(), *11*, *19*
buildTriplet, 6
buildTriplet(), *3*, *10, 11*, *13–16*, *19*, *21*, *28*, *30*, *32*

convergence (pip), 21
convergence(), *3*, *16*, *20*
convergence,MRCCCFit-method (pip), 21
covariates (donors), 9
covariates,MRCCCData-method (donors), 9
credibleIntervals, 8
credibleIntervals(), *3*, *13*, *16*, *23*, *29*, *33*
credibleIntervals,MRCCCFit-method
(credibleIntervals), 8

donors, 9
donors,MRCCCData-method (donors), 9

effectSizes (pip), 21
effectSizes(), *3*, *8*, *12*, *16*
effectSizes,MRCCCFit-method (pip), 21

firstStageEffects (pip), 21
firstStageEffects(), *3*, *16, 17*
firstStageEffects,MRCCCFit-method
(pip), 21
fitSettings (pip), 21
fitSettings(), *3*, *16, 17*
fitSettings,MRCCCFit-method (pip), 21

inputScale (pip), 21
inputScale(), *3*, *16*
inputScale,MRCCCFit-method (pip), 21
instrumentStrength (pip), 21
instrumentStrength(), *3*, *12*, *16, 17*, *20*, *28*
instrumentStrength,MRCCCFit-method
(pip), 21
IRanges::promoters(), *27*

ligandExpression (donors), 9
ligandExpression,MRCCCData-method
(donors), 9
ligandGenotypes (donors), 9
ligandGenotypes,MRCCCData-method
(donors), 9

MRCCC (MRCCC-package), 3
mrccc, 10
mrccc(), *3, 4*, *7, 8*, *14*, *16–19*, *22, 23*, *25*, *29, 30*, *32, 33*
mrccc,MRCCCData-method (mrccc), 10
MRCCC-package, 3
MRCCCData, *6, 7*, *9–11*, 13, *14, 15*, *17*, *19*
MRCCCData(), *3*, *7*, *10*, *15, 16*
MRCCCData-accessors, *14*
MRCCCData-accessors (donors), 9
MRCCCData-class, 15
MRCCCFit, *3–5*, *8*, *10*, *12*, *16*, *19*, *22*, *24–26*, *29, 30*, *32*
MRCCCFit-accessors, *3*, *12, 13*
MRCCCFit-accessors (pip), 21
MRCCCFit-class, 16
mrcccHyperparameters, 17
mrcccHyperparameters(), *11, 12*
mrcccScreen, 18
mrcccScreen(), *3*, *13*, *21*, *26*
MRCCCSummary, *20*, *32*
MRCCCSummary-class, 19

onek1kExample, 20
onek1kExample(), *3*

pathwayActivity (donors), 9
pathwayActivity,MRCCCData-method
(donors), 9
pip, 21
pip(), *3, 4*, *16*
pip,MRCCCFit-method (pip), 21

pipMCSE *(pip)*, 21
pipMCSE(), *16*
pipMCSE,MRCCCFit-method *(pip)*, 21
plot,MRCCCFit,missing-method, 24
plotEffectCurve, 25
plotEffectCurve(), *3*, *13*, *16*, *23, 24*, *30*
plotEffectCurve,MRCCCFit-method
(plotEffectCurve), 25
posteriorDraws *(pip)*, 21
posteriorDraws(), *3*, *16*
posteriorDraws,MRCCCFit-method *(pip)*, 21
posteriorMeans *(pip)*, 21
posteriorMeans(), *3*, *16, 17*
posteriorMeans,MRCCCFit-method *(pip)*, 21

receptorExpression *(donors)*, 9
receptorExpression,MRCCCData-method
(donors), 9
receptorGenotypes *(donors)*, 9
receptorGenotypes,MRCCCData-method
(donors), 9

S4Vectors::Annotated, *16*
S4Vectors::metadata(), *14*, *16*
screenTable, 26
screenTable(), *3*, *18, 19*
selectInstruments, 26
selectInstruments(), *3*, *6, 7*, *14*, *30*, *32*
set.seed(), *11*
show,MRCCCData-method
(MRCCCData-class), 15
show,MRCCCFit-method *(MRCCCFit-class)*,
16
show,MRCCCSummary-method
(MRCCCSummary-class), 19
signProbabilities, 28
signProbabilities(), *3*, *8*, *13*, *16*, *23*, *30*
signProbabilities,MRCCCFit-method
(signProbabilities), 28
signReversal, 29
signReversal(), *3*, *12, 13*, *16*, *23*, *25*, *29*
signReversal,MRCCCFit-method
(signReversal), 29
simulateMRCCC, 30
simulateMRCCC(), *3*, *7*
summary(), *11*, *19*
summary,MRCCCFit-method, *20*, 32

tripletLabels *(donors)*, 9
tripletLabels,MRCCCData-method
(donors), 9
tripletLabels,MRCCCFit-method *(pip)*, 21