Package ‘ivd’

May 8, 2026

Type Package

Title Individual Variance Detection

Version 1.0.0

Maintainer Philippe Rast <rast.ph@gmail.com>

Description Fit mixed-effects location scale models with spike-and-slab priors on the location random effects to identify units with unusual residual variances. The method is described in detail in Carmo, Williams and Rast (2025) <https://osf.io/sh6ne>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports nimble (>= 1.1.0), coda (>= 0.19.4), ggplot2, patchwork, future, future.apply, utils, rstan, ggrepel

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

BugReports https://github.com/consistentlyBetter/ivd/issues

Depends R (>= 3.0)

NeedsCompilation no

Author Philippe Rast [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-3630-6629>), Marwin Carmo [aut]

Repository CRAN

Date/Publication 2025-12-10 21:30:06 UTC

Contents

.require_suggest 2
build_ivd_model 2
codaplot 3
ivd 4

plot.ivd 6
run_MCMC_compiled_model 7
saeb 9
summary.ivd 9

Index **11**

`.require_suggest` *Helper to check for suggested package*

Description

Helper to check for suggested package

Usage

```
.require_suggest(pkg, feature)
```

Arguments

`pkg`	requested package
`feature`	requested feature

Author(s)

philippe

`build_ivd_model` *Build and compile NIMBLE model and MCMC once This function is exported for use in 'future' and is not meant to be called by user.*

Description

Build and compile NIMBLE model and MCMC once This function is exported for use in 'future' and is not meant to be called by user.

Usage

```
build_ivd_model(code, constants, dummy_data, dummy_inits, useWAIC = TRUE)
```

Arguments

`code`	Nimble code
`constants`	Constants
`dummy_data`	Data
`dummy_inits`	inits
`useWAIC`	Defaults to TRUE. Nimble argument

Value

A named `list` with two elements:

- `cmodel`: The compiled NIMBLE model object produced by `compileNimble()`.
- `cmcmc`: The compiled NIMBLE MCMC object, created using `buildMCMC()` and `compileNimble()`, configured to monitor the model parameters (including WAIC monitors if `useWAIC = TRUE`).

The function is intended for internal use (e.g., within parallel workers) and is not meant to be called directly by end users.

Examples

```
## Not run:
library(nimble)
# Generic nimble example
code <- nimbleCode({
  mu ~ dnorm(0, 1)
  x  ~ dnorm(mu, 1)
})

constants   <- list()
dummy_data  <- list(x = 0)
dummy_inits <- list(mu = 0)

out <- build_ivd_model(
  code        = code,
  constants   = constants,
  dummy_data  = dummy_data,
  dummy_inits = dummy_inits,
  useWAIC     = FALSE
)

str(out)

## End(Not run)
```

---

codaplot *Traceplot from the coda package*

Description

For more plots see coda

Usage

```
codaplot(obj, parameters = NULL, type = "traceplot", askNewPage = TRUE)
```

Arguments

`obj`	ivd object
`parameters`	Provide parameters of interest using names from the summary() output (e.g., "Intc", "scl_Intc", "sd_Intc", "R\[scl_Intc, Intc\]", "pip\[Intc, 5\]"). Defaults to NULL (plots all parameters).
`type`	Coda plot. Defaults to 'traceplot'. See coda for more options such as 'acfplot', 'densplot' etc.
`askNewPage`	Should user be prompted for next plot. Defaults to `TRUE`

Value

Specified coda plot

Author(s)

Philippe Rast

---

ivd — *Main function to set up and run parallel MCMC using nimble and future.* `ivd` *computes a mixed effects location and scale model with Spike and Slab regularization on the scale random effects.*

Description

Main function to set up and run parallel MCMC using nimble and future. `ivd` computes a mixed effects location and scale model with Spike and Slab regularization on the scale random effects.

Usage

```
ivd(
  location_formula,
  scale_formula,
  data,
  niter,
  nburnin = NULL,
  WAIC = TRUE,
  workers = 4,
  n_eff = "local",
  ss_prior_p = 0.5,
  ...
)
```

Arguments

`location_formula`	A formula for the location model
`scale_formula`	A formula for the scale model
`data`	Data frame in long format for analysis
`niter`	Total number of MCMC iterations after burnin
`nburnin`	Number of burnin iterations, defaults to the same as niter
`WAIC`	Compute WAIC, defaults to 'TRUE'
`workers`	Number of parallel R processes – doubles as 'chains' argument
`n_eff`	Use stan::monitor function or built local: 'stan' vs. 'local'
`ss_prior_p`	Prior inclusion probability. Defaults to '.5'.
`...`	Currently not used

Value

An object of class `"ivd"` (and `"list"`), which contains the results from fitting a mixed-effects location-scale model with Spike-and-Slab regularization using NIMBLE and parallel MCMC sampling.

The returned object is a named list with the following components:

- `samples`: An `mcmc.list` object containing posterior samples for all monitored parameters across all chains.
- `logLik_array`: A 3D array of pointwise log-likelihood values with dimensions `iterations` × `chains` × `N`.
- `rhat_values`: Vector of split-$\hat{R}$ convergence diagnostics (Vehtari et al., 2021).
- `n_eff`: Vector of effective sample sizes, either computed internally ("local") or via `rstan::monitor()` ("stan").
- `nimble_constants`: List of model constants used by the underlying NIMBLE model (e.g., number of groups, number of parameters).
- `X_location_names`, `Z_location_names`: Names of fixed and random effects in the location submodel.
- `X_scale`, `Z_scale`: Matrices used for the scale submodel's fixed and random effects.
- `Y`: Data frame with the response vector and group identifiers.
- `workers`: Number of parallel chains used.
- `...`: Additional elements created internally and used for downstream S3 methods (`print()`, `summary()`, etc.).

The object is designed to support S3 methods for printing, summarizing, and extracting results from the `ivd` model.

Examples

```
out <- ivd(location_formula = math_proficiency ~ 1 + (1 | school_id),
   scale_formula =  ~ 1 + (1 | school_id),
   data = saeb,
   niter = 1000,
   nburnin = 2000,
   WAIC = TRUE,
   workers = 1) ## Workers = 1 for CRAN server - not ideal for individual use
## Posterior inclusion probability plot (PIP)
plot(out, type = "pip")
## PIP vs. Within-cluster SD
plot(out, type =  "funnel")
## Diagnostic plots based on coda plots:
library(coda)
codaplot(out, parameters =  "Intc")
codaplot(out, parameters =  "R[scl_Intc, Intc]")
```

`plot.ivd` *Plot method for ivd objects*

Description

Plot method for ivd objects

Usage

```
## S3 method for class 'ivd'
plot(
  x,
  type = "pip",
  pip_level = 0.75,
  variable = NULL,
  label_points = TRUE,
  ...
)
```

Arguments

`x`	An object of type `ivd`.
`type`	Defaults to 'pip', other options are 'funnel' and 'outcome'.
`pip_level`	Defines a value for the posterior inclusion probability. Defaults to 0.75.
`variable`	Name of a specific variable. Defaults to `NULL`
`label_points`	Should points above the pip threshold be labelled? Defaults to `TRUE`.
`...`	Controls ggrepel aruments.

Value

Invisibly returns a `ggplot` object corresponding to the selected plot type. The primary purpose of this method is the side effect of displaying the plot.

The exact plot depends on the value of `type`:

- `"pip"` — Posterior inclusion probability plot for random scale effects.
- `"funnel"` — Funnel plot showing the relation between within-cluster standard deviation (`tau`) and posterior inclusion probabilities.
- `"outcome"` — Outcome plot relating cluster means (`mu`), posterior inclusion probability, and within-cluster SD.

When `label_points = TRUE`, labels for clusters exceeding the `pip_level` threshold are added using **ggrepel** (if available).

Author(s)

Philippe Rast

---

run_MCMC_compiled_model

Run MCMC on an already compiled model Exposed but internal function for future()

Description

Run MCMC on an already compiled model Exposed but internal function for future()

Usage

```
run_MCMC_compiled_model(
  compiled,
  seed,
  new_data,
  new_inits,
  niter,
  nburnin,
  useWAIC = TRUE,
  ...
)
```

Arguments

`compiled`	Compiled nimble model
`seed`	Seed, set by future
`new_data`	Data
`new_inits`	inits

`niter`	Sampling iteratons
`nburnin`	Number of burnin iterations
`useWAIC`	Defaults to TRUE
`...`	Placeholder for nimble arguments

Value

The output produced by `nimble::runMCMC()` when applied to a compiled NIMBLE MCMC object. The returned value depends on the `useWAIC` argument:

- If `useWAIC = TRUE`, a named `list` containing:
  - `samples`: A matrix of posterior draws (iterations × parameters).
  - `WAIC`: The WAIC value computed by NIMBLE.
  - `...`: Additional elements returned by `runMCMC()` when WAIC is enabled.
- If `useWAIC = FALSE`, a numeric matrix containing the posterior samples (iterations × parameters) with no additional elements.

This function is intended for internal use (e.g., within `future` workers) and is not meant to be called directly by end users.

Examples

```
## Not run:
library(nimble)
# Generic nimble example
code <- nimbleCode({
  mu ~ dnorm(0, 1)
  x  ~ dnorm(mu, 1)
})

constants   <- list()
dummy_data  <- list(x = 0)
dummy_inits <- list(mu = 0)

out <- build_ivd_model(
  code        = code,
  constants   = constants,
  dummy_data  = dummy_data,
  dummy_inits = dummy_inits,
  useWAIC     = FALSE
)

str(out)

## End(Not run)
```

saeb — *Basic Education Evaluation System (Saeb) -2021*

Description

The Basic Education Evaluation System (Saeb) is a series of large-scale external assessments conducted by Inep (National Institute for Educational Studies and Research) to diagnose the state of basic education in Brazil and identify factors that may affect student performance. This dataset contains the standardized math scores of 12th graders from 160 randomly selected schools in Rio de Janeiro who took the 2021 exam.

Usage

```
saeb
```

Format

A data frame with 11386 student's observations including 5 variables:

school_id A unique identifier for each school in the dataset.

public A binary variable indicating the type of school. It takes a value of `1` if the school is public and `0` if the school is private.

student_ses A numerical variable representing the socioeconomic status (SES) of the students.

math_proficiency A numerical variable representing the math proficiency level of the students, standardized with a mean of 0 and a standard deviation of 1.

location A numerical variable indicating the geographical location of the school. It takes a value of 1 for urban schools and 2 for rural schools.

Source

https://web.archive.org/web/20250202015037/https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/saeb/resultados

summary.ivd — *Summary of posterior samples*

Description

Summarize ivd object

Usage

```
## S3 method for class 'ivd'
summary(object, digits = 3, pip = "all", ...)
```

Arguments

`object`	ivd object
`digits`	Integer (Default: 2, optional). Number of digits to round to when printing.
`pip`	Print pip and model parameters ('all'); Only pip ('pip'), or only model parameeters 9('model'). Defaults to 'all'
`...`	Not used

Value

summary.ivd object

Author(s)

Philippe Rast

Index

∗ **datasets**
    saeb, 9
.require_suggest, 2

build_ivd_model, 2

codaplot, 3

ivd, 4

plot.ivd, 6

run_MCMC_compiled_model, 7

saeb, 9
summary.ivd, 9