Package: IntegMultiReg (via r-universe)

October 16, 2026

Title Integrative Bayesian Multiple Regression for Multi-Platform Biomarkers

Version 0.1.3

Description A Bayesian framework that integrates several regression models to identify a parsimonious set of biomarkers shared across disparate data platforms (for example genomic, transcriptomic and proteomic assays). Subjects are partitioned into subgroups defined by their pattern of platform availability, so that no subject with partially missing platform data is excluded, and information is borrowed across subgroups through a Markov random field prior on the variable-selection indicators together with non-local (product moment) priors on the regression effects. The methodology was introduced for time-to-event outcomes by Chekouo, Stingo, Doecke and Do (2017) <doi:10.1111/biom.12587>; this package additionally supports continuous (Gaussian) and binary (probit) outcomes. Posterior inference is carried out by a Markov chain Monte Carlo sampler implemented in C for computational efficiency.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports graphics, grDevices, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

SystemRequirements GSL (>= 2.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.2

Author Sinian Zhang [aut], Jianfeng Wang [aut], Thierry Chekouo [aut, cre] (ORCID: <https://orcid.org/0000-0002-3594-9182>)

Maintainer Thierry Chekouo <tchekouo@umn.edu>

Config/pak/sysreqs libgsl0-dev

Repository https://cran.r-universe.dev

Date/Publication 2026-09-16 11:49:46 UTC

RemoteUrl https://github.com/cran/IntegMultiReg

RemoteRef HEAD

RemoteSha 94377eb6b938b445ffb9b7d9352f3ea1e8adc025

Contents

as.data.frame.imr_data . . . 2
coef.imr . . . 3
compare_imr . . . 4
confint.imr . . . 4
cv_imr . . . 5
imr . . . 6
imr_data . . . 10
imr_posterior_methods . . . 11
kircIMR . . . 11
plot.imr . . . 13
plot_subgroup_sizes . . . 15
plot_top_features . . . 16
posterior_draws . . . 18
posterior_summary . . . 20
predict.imr . . . 20
predict.imr_posterior . . . 22
print.imr . . . 23
simIMR . . . 24
summary.imr . . . 25
validate_imr . . . 26
validate_imr_data . . . 26

Index **27**

`as.data.frame.imr_data` *Convert IMR Data to an Availability Data Frame*

Description

Convert IMR Data to an Availability Data Frame

Usage

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

Arguments

x	An ‘"imr_data"‘ object.
row.names	Unused; present for compatibility with [as.data.frame()].
optional	Unused; present for compatibility with [as.data.frame()].
...	Unused.

Value

The subject-by-platform availability table, including bitstrings.

---

coef.imr *Marginal Posterior Inclusion Probabilities of an IMR Fit*

Description

Extracts the posterior mean variable-selection probabilities (the marginal posterior inclusion probabilities, mPIP) of a fitted model.

Usage

```
## S3 method for class 'imr'
coef(object, ...)

coef_imr(object, ...)
```

Arguments

object	A fitted object of class ‘"imr"‘.
...	Unused; present for S3 compatibility.

Value

A named list with one matrix per platform. Rows are the subgroups containing that platform (labelled by their availability bitstrings) and columns are the platform features.

See Also

[imr()]

compare_imr *Compare Fitted IMR Models*

Description

Creates a compact descriptive comparison of compatible IMR fits. The table deliberately does not treat raw log-posterior values as likelihood criteria; it reports model structure and the number of features passing a common mPIP threshold.

Usage

```
compare_imr(..., threshold = 0.5)
```

Arguments

`...`	Fitted ‘"imr"‘ objects, or one list of fitted objects.
`threshold`	Common mPIP threshold used to count selected features.

Value

A data frame with one row per fit.

confint.imr *Credible Intervals for an IMR Fit*

Description

Standard ‘confint()‘ interface to [posterior_summary()].

Usage

```
## S3 method for class 'imr'
confint(object, parm = c("all", "selection", "theta"), level = 0.95, ...)
```

Arguments

`object`	A fitted ‘"imr"‘ object.
`parm`	Either ‘"all"‘, ‘"selection"‘ or ‘"theta"‘.
`level`	Credible interval level.
`...`	Additional arguments passed to [posterior_summary()].

Value

A list of per-platform credible-interval tables, or a list with both selection and theta results when ‘parm = "all"‘.

cv_imr *Cross-Validated Predictive Accuracy of an IMR Fit*

Description

‘cv_imr()‘ evaluates predictive accuracy of a model fitted with [imr()] using repeated K-fold splits and Bayesian model averaging over the fitted MCMC samples. The sampler is not re-run inside each training fold; fit a separate object with ‘method = "BMS"‘ if a no-borrowing comparison is needed. The accuracy measure depends on the outcome type: the concordance index (C-index) for right-censored outcomes, the area under the ROC curve (AUC) for binary outcomes, and the mean squared error (MSE) for continuous outcomes.

Usage

```
cv_imr(
  object,
  k = 5,
  rounds = 2,
  method = NULL,
  max_models = 100,
  verbose = FALSE
)
```

Arguments

object	A fitted object of class ‘"imr"‘ returned by [imr()].
k	Integer number of cross-validation folds per round (default ‘5‘). Must be at least ‘2‘, and each availability subgroup must contain at least ‘k‘ subjects.
rounds	Integer number of independent cross-validation rounds to average over (default ‘2‘). Must be positive.
method	Optional compatibility argument. If supplied, it must match the method stored in ‘object‘; ‘cv_imr()‘ cannot turn an IMR fit into a BMS fit or vice versa.
max_models	Integer maximum number of selection models used for Bayesian model averaging (default ‘100‘). Must be positive.
verbose	Logical; if ‘TRUE‘, print the C routine’s cross-validation diagnostics. Defaults to ‘FALSE‘.

Value

A list with two numeric matrices, each of dimension ‘rounds‘ x ‘(n_subgroups + 1)‘, whose last column corresponds to all subjects pooled and whose remaining columns are named by the availability subgroup bitstrings:

total_cindex	Accuracy computed on the pooled cross-validated predictions within each availability subgroup (and overall).
subset_cindex	Accuracy computed fold-by-fold within each availability subgroup.

The accuracy metric (‘"C-index"‘, ‘"AUC"‘ or ‘"MSE"‘) is recorded in the ‘"metric"‘ attribute of the returned list.

See Also

[imr()], [predict.imr()]

Examples

```
data("simIMR", package = "IntegMultiReg")
fit <- imr(
  platform_data_list = simIMR$platforms, outcome = simIMR$outcome,
  cov = simIMR$covariates, type_outcome = "binary",
  nu = c(-4, -3, -4), sample_mcmc = c(200, 100), ssize = 5, seed = 1
)
cv <- cv_imr(fit, k = 5, rounds = 2)
cv$total_cindex
```

---

imr *Fit the Integrative Bayesian Multi-Platform Regression Model*

Description

‘imr()‘ fits the integrative multi-regression (IMR) model of Chekouo et al. (2017) by Markov chain Monte Carlo (MCMC). It identifies biomarkers associated with a time-to-event, binary or continuous outcome while borrowing information across all subjects, regardless of which platforms each subject has measured. Subjects are partitioned into availability subgroups defined by their pattern of platform availability; one regression is built per subgroup and information is shared across subgroups through a Markov random field (MRF) prior on the variable-selection indicators together with non-local priors on the regression coefficients. The sampler is implemented in C for efficiency.

The arguments are grouped by the orthogonal aspect of the analysis that each one controls: the *data* (‘platform_data_list‘, ‘outcome‘, ‘cov‘), the *likelihood* (‘type_outcome‘), the *model* (‘method‘), subgroup *filtering* (‘ssize‘), the *priors* (‘nu‘, ‘hh‘, ‘h0‘, ‘sig_alpha_psi‘, ‘thet_alph_bet‘), the *computation* (‘sample_mcmc‘, ‘seed‘) and the *output* (‘verbose‘).

Usage

```
imr(platform_data_list = NULL, ..., formula = NULL)

## Default S3 method:
imr(
  platform_data_list,
  outcome,
  cov = NULL,
  type_outcome = c("right.censored", "binary", "continuous"),
  method = c("IMR", "BMS"),
  ssize = 30,
```

```
  nu = rep(-3, length(platform_data_list)),
  hh = 0.087,
  h0 = 10000,
  sig_alpha_psi = c(0.001, 0.001),
  thet_alph_bet = c(40, 10),
  sample_mcmc = c(2000, 1000),
  seed = NULL,
  verbose = FALSE,
  survival_scale = c("log", "identity"),
  ...
)

## S3 method for class 'formula'
imr(
  platform_data_list,
  data,
  platforms,
  id = "id",
  type_outcome = c("right.censored", "binary", "continuous"),
  ...
)

## S3 method for class 'imr_data'
imr(platform_data_list, ...)
```

Arguments

`platform_data_list`
: A list of data frames, one per platform. Each data frame must contain an ‘id‘ column (the subject identifier, taken to be the first column); the remaining columns are finite numeric features measured on that platform. Subject identifiers must be unique within each data frame. The ‘id‘ column links subjects across platforms and to the outcome and covariate data.

`...`
: Additional fitting arguments passed from the formula or ‘imr_data‘ method to the default method. Unused arguments are rejected.

`formula`
: A model formula. This named argument is equivalent to passing the formula as the first argument. The model always includes an intercept: ‘0‘/‘-1‘ and ‘offset()‘ terms are rejected. The identifier column is excluded when expanding ‘.‘.

`outcome`
: A data frame containing an ‘id‘ column and the response. For ‘type_outcome = "right.censored"‘ it must also contain the event/censoring time and a censoring indicator (three columns in total). For ‘"binary"‘ the response must be coded 0/1 and for ‘"continuous"‘ it is a numeric response (two columns in total). The ‘id‘ column must be first and unique.

`cov`
: An optional data frame of clinical covariates including an ‘id‘ column followed by finite numeric covariates. Covariates are always included in every regression (they are not subject to selection). Defaults to ‘NULL‘ (no covariates).

type_outcome	Character string specifying the outcome type, one of ‘"right.censored"‘ (default), ‘"binary"‘ or ‘"continuous"‘.
method	Character string specifying the method, ‘"IMR"‘ (default) for the integrative model that shares information across subgroups via the MRF prior, or ‘"BMS"‘ for the non-integrative Bayesian multi-step variant that fits each subgroup independently (MRF interaction parameters set to zero).
ssize	Minimum availability subgroup size for a subgroup to be modelled. Subgroups with at most ‘ssize‘ subjects are dropped. Default is ‘30‘.
nu	A numeric vector of prior log-odds of inclusion, one value per platform, controlling the prior sparsity of the selected features. Defaults to ‘rep(-3, length(platform_data_list))‘.
hh	Scale of the non-local (product moment) prior on the slab regression effects (default ‘0.087‘).
h0	pMOM prior scale for the always-included intercept and clinical coefficients (default ‘10000‘), corresponding to tau_0 in the original paper. This is not the marginal prior variance.
sig_alpha_psi	Length-2 numeric vector with the shape and rate of the inverse-gamma prior on the response error variance (default ‘c(0.001, 0.001)‘). Ignored when ‘type_outcome = "binary"‘: a probit model uses a highly concentrated inverse-gamma prior with shape and rate ‘100000‘ to approximate unit residual variance for identifiability.
thet_alph_bet	Length-2 numeric vector with the shape and rate of the gamma prior on the MRF interaction parameters theta, which borrow information across subgroups (default ‘c(40, 10)‘).
sample_mcmc	An integer vector ‘c(n_retained, n_burnin)‘ giving the number of post-burn-in draws to retain and the number of burn-in iterations to discard. The sampler runs ‘n_retained + n_burnin‘ iterations in total, so the returned ‘log_posterior‘ has ‘sum(sample_mcmc)‘ entries. Default is ‘c(2000, 1000)‘.
seed	Optional integer seed for the sampler. If ‘NULL‘ (the default) a seed is drawn from the current R RNG state, so a run is reproducible whenever [set.seed()] is called beforehand or an explicit ‘seed‘ is passed.
verbose	Logical; if ‘TRUE‘, print the sampler's progress and diagnostics to the console. Defaults to ‘FALSE‘ (quiet).
survival_scale	Working response scale for right-censored outcomes. ‘"log"‘ (default) logs the supplied positive event/censoring times, as in the original AFT model. ‘"identity"‘ reproduces historical package analyses and does not fit a log-time AFT model. Ignored for other outcome types.
data	A data frame used with the formula interface.
platforms	A list of platform data frames used with the formula interface.
id	Name of the identifier column in ‘data‘ and ‘platforms‘.

Details

All feature and covariate data are standardized internally (mean 0, standard deviation 1); the centring and scaling factors are stored in the returned object so that [predict.imr()] can apply the same transformation to new subjects. For right-censored outcomes the latent log-survival times of censored subjects are imputed within the sampler; for binary outcomes a probit data-augmentation latent variable is sampled.

Value

An object of class '"imr"': a list with components

`gam_mean` List (one matrix per platform) of posterior mean variable-selection probabilities; rows index the subgroups containing the platform and columns index the platform features.

`theta_mean` List (one matrix per platform) of posterior mean MRF interaction parameters between subgroups.

`estimate_latent_y` List (one vector per subgroup) of posterior mean latent responses, useful for censored or binary data.

`log_posterior` Numeric vector of log-posterior values across all MCMC iterations (burn-in included).

`gam_sample` Post-burn-in MCMC samples of the selection indicators.

`theta_sample` Post-burn-in MCMC samples of the MRF interaction parameters (absent when 'method = "BMS"').

`list_hyperpara`, `data1`, `data2` Hyper-parameters and pre-processed data retained for prediction and cross-validation.

`call`, `type_outcome`, `n_platform`, `platform_names`, `feature_names`, `covariate_names`, `model_bitstrings`, `sample_size`, `model_platforms`, `platform_models`, `sample_mcmc`, `nu`, `method` Run metadata used by the print, summary, plot and predict methods.

References

Chekouo T, Stingo FC, Doecke JD, Do K-A (2017). "A Bayesian Integrative Approach for Multi-Platform Genomic Data: A Kidney Cancer Case Study." *Biometrics*, **73**(2), 615–624. doi:10.1111/biom.12587

See Also

[predict.imr()], [cv_imr()], [summary.imr()], [plot.imr()]

Examples

```
data("simIMR", package = "IntegMultiReg")
fit <- imr(
  platform_data_list = simIMR$platforms,
  outcome = simIMR$outcome,
  cov = simIMR$covariates,
  type_outcome = "binary",
  nu = c(-4, -3, -4),
  sample_mcmc = c(200, 100),
  ssize = 5,
  seed = 1
)
fit
```

imr_data *Validated Multi-Platform Data for IMR*

Description

‘imr_data()‘ creates a validated data object shared by model fitting and prediction. It standardizes the subject identifier to ‘id‘, records the observed-platform pattern for every subject and catches alignment problems before the MCMC sampler is called.

Usage

```
imr_data(
  platforms,
  outcome = NULL,
  covariates = NULL,
  type_outcome = NULL,
  id = "id"
)
```

Arguments

platforms	A non-empty named list of data frames, one per platform. Every data frame must contain the subject identifier and at least one finite numeric feature.
outcome	Optional outcome data frame. It is required when the object is passed to [imr()] and may be omitted for prediction data. For binary and continuous outcomes it contains the identifier and response; for right-censored outcomes it contains the identifier, time and status.
covariates	Optional data frame containing the identifier followed by clinical covariates.
type_outcome	Optional outcome type: ‘"binary"‘, ‘"continuous"‘ or ‘"right.censored"‘. It is required when ‘outcome‘ is supplied.
id	Name of the subject-identifier column in every supplied data frame.

Value

An object of class ‘"imr_data"‘ with components ‘platforms‘, ‘outcome‘, ‘covariates‘, ‘type_outcome‘, ‘id‘, ‘availability‘ and ‘subgroup_sizes‘, ‘n_platform_subjects‘ and ‘excluded_ids‘. Availability summaries include only subjects with all required outcome/covariate rows.

Examples

```
data("simIMR", package = "IntegMultiReg")
dat <- imr_data(
  platforms = simIMR$platforms,
  outcome = simIMR$outcome.binary,
  covariates = simIMR$covariates,
  type_outcome = "binary"
)
dat
```

imr_posterior_methods *Summarize Regression Posterior Draws*

Description

Summarize Regression Posterior Draws

Usage

```
## S3 method for class 'imr_posterior'
summary(object, level = 0.95, ...)

## S3 method for class 'imr_posterior'
confint(object, parm = "coefficients", level = 0.95, ...)

## S3 method for class 'imr_posterior'
coef(object, ...)

## S3 method for class 'imr_posterior'
print(x, ...)
```

Arguments

object, x	An ‘imr_posterior‘ object returned by [posterior_draws()].
level	Equal-tail credible level, between zero and one.
...	Unused.
parm	Currently only ‘"coefficients"‘.

Value

‘summary()‘ and ‘confint()‘ return coefficient tables by subgroup. ‘coef()‘ returns posterior mean coefficient vectors. ‘print()‘ returns its argument invisibly. Effects are on subgroup-standardized predictor scales. Intervals include zero-mass from inactive selection models.

kircIMR *Reduced TCGA-KIRC Multi-Platform Survival Example*

Description

A reduced real-data example derived from public UCSC Xena TCGA kidney renal clear cell carcinoma (KIRC) data, not from controlled-access TCGA/GDC files. The data set mirrors the multi-platform structure of Chekouo, Stingo, Doecke and Do (2017): mRNA expression, miRNA expression and DNA methylation are observed on overlapping but not identical sets of patients, with clinical covariates and right-censored survival outcome.

Usage

```
kircIMR
```

Format

A named list with the components:

platforms A list of three data frames, ‘mrna‘, ‘mirna‘ and ‘methylation‘. Each has an ‘id‘ column containing package-internal patient labels followed by screened molecular features.

covariates Clinical covariates with an ‘id‘ column: age at initial diagnosis, female sex indicator, pathologic stage and histologic grade.

outcome Alias of ‘outcome.survival‘, for convenience.

outcome.survival Data frame with ‘id‘, observed survival/follow-up time in days and event indicator ‘status‘ (1 = deceased, 0 = censored).

feature_screening Per-platform feature-ranking information and a short description of the screening procedure used to create the reduced panel.

platform_availability Patient-level logical matrix indicating which of the three platforms are available.

subgroup_sizes Counts of the non-empty availability subgroups induced by the reduced data.

model_subgroup_sizes The availability subgroups that would be modelled under the default real-data illustration threshold ‘ssize = 30‘.

paper_alignment Metadata describing how the reduced example maps to the Biometrics KIRC case study.

source Source URLs and reference metadata.

Details

The full TCGA source files are not distributed with the package. Instead, this object stores a small screened feature panel suitable for examples and vignette analyses. Patient identifiers are package-internal labels (‘KIRC001‘, ‘KIRC002‘, ...) rather than TCGA barcodes; no barcode mapping is distributed with the package. Users should not attempt participant re-identification or linkage to external resources. The generating script is provided in the source repository under ‘data-raw/make_kircIMR.R‘.

The source data are public UCSC Xena TCGA-KIRC sampleMap files: mRNA expression (‘HiSeqV2‘), miRNA expression (‘miRNA_HiSeq_gene‘), DNA methylation (‘HumanMethylation450‘) and the KIRC clinical matrix. Molecular features are screened with univariable Cox models after basic filtering, and missing molecular values in the retained panel are median-imputed within feature. The full Biometrics analysis used 776 mRNA, 91 miRNA and 729 DNA methylation features; this package example retains 50, 30 and 50 features, respectively, to keep examples lightweight. The package does not distribute controlled-access TCGA/GDC data or the full Biometrics/Wiley supplementary data files.

References

Chekouo T, Stingo FC, Doecke JD, Do K-A (2017). "A Bayesian Integrative Approach for Multi-Platform Genomic Data: A Kidney Cancer Case Study." *Biometrics*, **73**(2), 615–624. doi:10.1111/biom.12587

The Cancer Genome Atlas Research Network; National Cancer Institute Genomic Data Commons; UCSC Xena.

Examples

```
data("kircIMR", package = "IntegMultiReg")
sapply(kircIMR$platforms, dim)
kircIMR$subgroup_sizes
kircIMR$model_subgroup_sizes
```

`plot.imr` *Plot Method for IMR Fits*

Description

Visualizes a fitted ‘"imr"‘ object. Three plot types are available:

‘"selection"‘ Heatmap of the marginal posterior inclusion probabilities (mPIP), one panel per platform, with features on the horizontal axis, availability subgroups on the vertical axis and a small intensity legend showing that darker values are closer to 1.

‘"theta"‘ Heatmap of the posterior mean MRF interaction parameters between availability subgroups, one panel per platform.

‘"trace"‘ Trace plot of the log-posterior across MCMC iterations.

‘"theta_trace"‘ Trace plot for one retained MRF interaction parameter, selected by ‘platform‘ and ‘parameter‘.

‘"selection_trace"‘ Trace plot for one retained selection indicator, selected by ‘platform‘, ‘subgroup‘ and ‘feature‘.

See [plot_top_features()] and [plot_subgroup_sizes()] for two further ready made displays.

Usage

```
## S3 method for class 'imr'
plot(
  x,
  type = c("selection", "theta", "trace", "theta_trace", "selection_trace"),
  platform = NULL,
  base_cex = 1,
  cex_axis = NULL,
  cex_lab = NULL,
  cex_main = NULL,
  col = NULL,
```

```
  palette = NULL,
  legend = TRUE,
  legend_width = 0.28,
  mar = NULL,
  mgp = NULL,
  parameter = 1L,
  subgroup = 1L,
  feature = 1L,
  ...
)

plot_imr(x, ...)
```

Arguments

x	A fitted object of class ‘"imr"‘.
type	Character; one of ‘"selection"‘ (default), ‘"theta"‘, ‘"trace"‘, ‘"theta_trace"‘ or ‘"selection_trace"‘.
platform	Optional integer vector selecting which platforms to display for the ‘"selection"‘ and ‘"theta"‘ plots; defaults to all platforms.
base_cex	Overall text-size multiplier. The ‘cex_*‘ arguments default to values derived from this multiplier (default ‘1‘).
cex_axis	Axis-label size multiplier. If ‘NULL‘, a plot-specific default derived from ‘base_cex‘ is used.
cex_lab	Axis-title size multiplier. If ‘NULL‘, a plot-specific default derived from ‘base_cex‘ is used.
cex_main	Main-title size multiplier. If ‘NULL‘, a plot-specific default derived from ‘base_cex‘ is used.
col	Optional colours. For heatmaps this is the colour scale; for trace plots this is the line colour.
palette	Optional heatmap palette name used when ‘col = NULL‘. Selection plots default to ‘"grey"‘ to preserve the mPIP intensity scale; theta plots default to the muted grey-blue ‘"heatmap"‘ palette.
legend	Logical; for ‘"selection"‘ plots, should the mPIP intensity legend be drawn (default ‘TRUE‘)?
legend_width	Relative width of the intensity-legend panel for ‘"selection"‘ plots (default ‘0.28‘).
mar, mgp	Optional graphical margin and axis-title placement vectors passed to [graphics::par()] for finer layout control.
parameter	Positive integer selecting a theta-pair column for ‘type = "theta_trace"‘.
subgroup	Positive integer selecting a platform-specific subgroup row for ‘type = "selection_trace"‘.
feature	Positive integer selecting a feature column for ‘type = "selection_trace"‘.
...	Further graphical parameters passed to the underlying plotting functions.

Value

'NULL', invisibly; called for the side effect of producing a plot.

See Also

[imr()], [plot_top_features()], [plot_subgroup_sizes()]

`plot_subgroup_sizes` *Plot the Availability Subgroup Sizes of an IMR Fit*

Description

Draws a bar chart of the number of subjects in each modelled availability subgroup (the non-empty regions of the Venn diagram), giving a quick picture of how the sample is distributed across subgroups.

Usage

```
plot_subgroup_sizes(
  object,
  base_cex = 1,
  cex_axis = NULL,
  cex_lab = NULL,
  cex_main = NULL,
  cex_values = NULL,
  col = NULL,
  palette = "subgroup",
  show_values = TRUE,
  ylim = NULL,
  mar = NULL,
  mgp = NULL,
  ...
)
```

Arguments

`object`	A fitted object of class '"imr"' returned by [imr()].
`base_cex`	Overall text-size multiplier. The 'cex_*' arguments default to values derived from this multiplier (default '1').
`cex_axis`	Axis-label size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
`cex_lab`	Axis-title size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
`cex_main`	Main-title size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.

cex_values	Bar-value label size multiplier. If ‘NULL‘, a plot-specific default derived from ‘base_cex‘ is used.
col	Optional bar colours. If ‘NULL‘, colours are generated from ‘palette‘.
palette	Palette name used when ‘col = NULL‘; ‘"subgroup"‘ gives the package’s standard neutral grey-blue subgroup colours (default).
show_values	Logical; should sample sizes be printed above the bars (default ‘TRUE‘)?
ylim	Optional numeric vector of length two giving the vertical axis limits. If ‘NULL‘, limits are chosen from the subgroup sizes.
mar, mgp	Optional graphical margin and axis-title placement vectors passed to [graphics::par()] for finer layout control.
...	Further graphical parameters passed to [graphics::barplot()].

Value

Invisibly, the named integer vector of subgroup sizes.

See Also

[imr()], [plot.imr()], [plot_top_features()]

Examples

```
data("simIMR", package = "IntegMultiReg")
fit <- imr(
  platform_data_list = simIMR$platforms, outcome = simIMR$outcome,
  cov = simIMR$covariates, type_outcome = "binary",
  nu = c(-4, -3, -4), sample_mcmc = c(200, 100), ssize = 5, seed = 1
)
plot_subgroup_sizes(fit)
```

---

plot_top_features *Plot the Top Selected Features of an IMR Fit*

Description

Draws a horizontal bar chart of the features with the highest marginal posterior inclusion probability (mPIP). For each platform-feature pair, the plotted score is the highest mPIP attained among the availability subgroups containing that platform; the displayed bars are the largest such scores across all platforms. This complements the per-platform heatmap of ‘plot(fit, type = "selection")‘.

Usage

```
plot_top_features(
  object,
  top = 10,
  base_cex = 1,
  cex_names = NULL,
  cex_axis = NULL,
  cex_lab = NULL,
  cex_main = NULL,
  cex_legend = NULL,
  col = NULL,
  palette = "platform",
  reference = 0.5,
  show_source = TRUE,
  legend = TRUE,
  xlim = c(0, 1),
  mar = NULL,
  mgp = NULL,
  ...
)
```

Arguments

object	A fitted object of class '"imr"' returned by [imr()].
top	Integer; the number of highest-mPIP features to display (default '10').
base_cex	Overall text-size multiplier. The 'cex_*' arguments default to values derived from this multiplier (default '1').
cex_names	Feature-label size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
cex_axis	Axis-label size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
cex_lab	Axis-title size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
cex_main	Main-title size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
cex_legend	Legend text size multiplier. If 'NULL', a plot-specific default derived from 'base_cex' is used.
col	Optional bar colours. If 'NULL', colours are generated from 'palette'.
palette	Palette name used when 'col = NULL'; '"platform"' gives the package's standard muted platform colours (default).
reference	Optional vertical reference line; use 'NULL' to suppress it (default '0.5').
show_source	Logical; should each bar be annotated with the availability subgroup (bitstring) in which the feature attained its maximum mPIP (default 'TRUE')?
legend	Logical; should the platform legend be drawn (default 'TRUE')?

xlim	Numeric vector of length two giving the horizontal axis limits (default ‘c(0, 1)‘).
mar, mgp	Optional graphical margin and axis-title placement vectors passed to [graphics::par()] for finer layout control.
...	Further graphical parameters passed to [graphics::barplot()].

Value

Invisibly, a data frame of the displayed features with columns ‘platform‘, ‘feature‘, ‘mpip‘ and ‘subgroup‘, where ‘mpip‘ is the maximum subgroup mPIP for that platform-feature pair and ‘subgroup‘ is the subgroup where the maximum is attained. Rows are ordered by decreasing mPIP.

See Also

[imr()], [plot.imr()], [plot_subgroup_sizes()]

Examples

```
data("simIMR", package = "IntegMultiReg")
fit <- imr(
  platform_data_list = simIMR$platforms, outcome = simIMR$outcome,
  cov = simIMR$covariates, type_outcome = "binary",
  nu = c(-4, -3, -4), sample_mcmc = c(200, 100), ssize = 5, seed = 1
)
plot_top_features(fit, top = 8)
```

---

posterior_draws *Sample Regression Coefficients and Residual Variances*

Description

Augments the retained selection models with conditional coefficient and variance draws. The fitted sampler and cross-validation results are unchanged.

Usage

```
posterior_draws(
  object,
  draws = 1000L,
  burnin = 1000L,
  chains = 2L,
  conditional_draws = 200L,
  seed = 1L
)
```

Arguments

`object`	An ‘imr‘ fit with stored data and selection draws.
`draws`	Number of model-averaged draws to return (default ‘1000‘).
`burnin`	Conditional Gibbs burn-in iterations for each distinct subgroup selection model and each chain (default ‘1000‘).
`chains`	Number of conditional chains, at least two (default ‘2‘).
`conditional_draws`	Minimum retained iterations per conditional chain used for sampling and split R-hat (default ‘200‘). Increase with ‘burnin‘ if the reported conditional diagnostics are poor.
`seed`	Integer seed. The caller’s random-number state is restored.

Details

All active coefficients, including the always-included intercept and clinical effects, follow the original first-order pMOM prior. Inactive molecular coefficients are exactly zero. Molecular and clinical slopes are on the subgroup-standardized predictor scale; responses are not standardized.

Whole selection draws are resampled uniformly from ‘object$gam_sample‘, preserving their empirical joint distribution across subgroups. For each distinct subgroup model, a Gibbs sampler draws coefficients and variance conditional on the observed data. Binary and censored responses are augmented anew; posterior mean latent responses are not treated as observed data. The binary variance uses the same highly concentrated inverse-gamma prior as the fitted model (approximately, not exactly, one).

This is a two-stage posterior approximation: model weights inherit the original sampler’s Laplace approximation and finite-chain exploration. The conditional Gibbs draws do not make those weights exact. Classical split R-hat is reported for each conditional model and does not diagnose the original selection chain. Inspect both stages and increase simulation effort before interpreting intervals. Runtime grows with the number of distinct subgroup selection models, not just ‘draws‘.

Old survival fits without explicit response-scale metadata must be refitted. Log-time fits return coefficients for log time; identity-scale fits are retained only for historical compatibility.

Value

An ‘imr_posterior‘ object containing ‘beta‘ (one draws-by-coefficients matrix per subgroup), ‘variance‘, source ‘model_draw‘ indices, conditional ‘diagnostics‘, and the originating ‘fit‘. Coefficient column names distinguish clinical variables from platform features. Use ‘summary()‘, ‘confint()‘, ‘coef()‘ and ‘predict()‘ on this object.

References

Chekouo et al. (2017). doi:10.1111/biom.12587, Section 3.1 and Web Appendix C.

Examples

```
x <- data.frame(id = 1:40, marker = seq(-1, 1, length.out = 40))
y <- data.frame(id = x$id, y = 1 + x$marker + sin(x$id) / 3)
fit <- imr(list(assay = x), y, type_outcome = "continuous",
```

```
           h0 = 1, sample_mcmc = c(500, 250), seed = 1)
draws <- posterior_draws(fit, draws = 1000, burnin = 1000,
                         conditional_draws = 1000, seed = 2)
summary(draws)
```

`posterior_summary` *Posterior Uncertainty Summary for an IMR Fit*

Description

Summarizes retained MCMC draws for variable-selection indicators and MRF interaction parameters. The selection tables report posterior means, posterior standard deviations and equal-tail credible intervals for each platform-feature/subgroup indicator. The theta tables provide the same summaries for each pair of linked availability subgroups.

Usage

```
posterior_summary(object, ...)

## S3 method for class 'imr'
posterior_summary(object, level = 0.95, ...)
```

Arguments

`object`	A fitted ‘"imr"‘ object.
`...`	Unused; present for future methods.
`level`	Credible interval level between zero and one (default ‘0.95‘).

Value

An object of class ‘"posterior_summary.imr"‘ containing ‘selection‘ and ‘theta‘ tables.

`predict.imr` *Predict Outcomes for New Subjects*

Description

‘predict()‘ method for objects of class ‘"imr"‘ produced by [imr()]. New subjects are matched to the availability subgroup models learned during training, missing platforms are handled automatically, and the test features are standardized with the training set’s centring and scaling factors before Bayesian model averaging.

Usage

```
## S3 method for class 'imr'
predict(
  object,
  newdata,
  platform_names = NULL,
  covariates = NULL,
  max_models = 100,
  verbose = FALSE,
  ...
)

predict_imr(object, newdata, ...)
```

Arguments

object	A fitted object of class ‘"imr"‘ returned by [imr()].
newdata	A list of data frames with the new platform measurements, or an [imr_data()] object created for prediction. When an ‘imr_data‘ object is supplied, its covariates are used automatically. Each data frame must include ‘id‘ as the first column, followed by finite numeric feature columns matching the corresponding training platform.
platform_names	A character vector giving, for each element of ‘newdata‘, the index (‘"1"‘, ‘"2"‘, ...) of the corresponding training platform. Defaults to ‘NULL‘, meaning the elements of ‘newdata‘ are taken to be in the same order as the platforms supplied to [imr()].
covariates	An optional data frame of clinical covariates for the test subjects, including ‘id‘ as the first column. Required when the model was fitted with covariates and ignored with a warning when it was not. For a formula fit, supply the original predictor columns (including factors and variables used in transformations). The training terms, factor levels and contrasts are reused. The training identifier name is also accepted. Alternatively, an already encoded numeric model matrix may be supplied as a data frame with ‘id‘ and exactly the fitted covariate column names.
max_models	Integer; the maximum number of distinct selection models (gamma configurations) used for Bayesian model averaging. Default ‘100‘.
verbose	Logical; if ‘TRUE‘, print the C routine’s diagnostics. Defaults to ‘FALSE‘.
...	Unused; present for S3 compatibility.

Details

Predictions are only produced for subjects observed on at least one platform (and, when ‘covariates‘ is supplied, with covariate data). For ‘"binary"‘ outcomes the returned ‘predict‘ column is a probability obtained through the probit link (‘pnorm‘); for ‘"continuous"‘ and ‘"right.censored"‘ outcomes it is the predicted working response. For default log-time survival fits it is on the log-time scale; exponentiating gives a transformed point prediction, not a posterior mean survival time. Old and identity-scale fits retain their historical response scale.

Value

A named list with one data frame per active availability subgroup model. Each data frame has columns ‘id‘ (subject identifier) and ‘predict‘ (predicted value or probability at full numeric precision).

See Also

[imr()], [cv_imr()]

Examples

```
data("simIMR", package = "IntegMultiReg")
fit <- imr(
  platform_data_list = simIMR$platforms, outcome = simIMR$outcome,
  cov = simIMR$covariates, type_outcome = "binary",
  nu = c(-4, -3, -4), sample_mcmc = c(200, 100), ssize = 5, seed = 1
)
new_x <- simIMR$platforms[[1]][1:10, ]
new_z <- simIMR$platforms[[2]][1:7, ]
predict(fit, newdata = list(new_x, new_z), covariates = simIMR$covariates)
```

---

predict.imr_posterior *Predict Using Coefficient Posterior Draws*

Description

Predict Using Coefficient Posterior Draws

Usage

```
## S3 method for class 'imr_posterior'
predict(
  object,
  newdata,
  platform_names = NULL,
  covariates = NULL,
  type = c("mean", "response"),
  level = 0.95,
  seed = 1L,
  ...
)
```

Arguments

`object`	An ‘imr_posterior‘ object.
`newdata`, `platform_names`, `covariates`	As in [predict.imr()].
`type`	‘"mean"‘ returns uncertainty in the conditional response mean (event probability for binary data). ‘"response"‘ additionally generates new outcomes, including residual variability. Binary response intervals are discrete 0/1; use ‘"mean"‘ for event-probability intervals.
`level`	Equal-tail interval level.
`seed`	Integer simulation seed; the caller’s RNG state is restored.
`...`	Unused.

Details

Log-time survival fits return time-scale results: for ‘"mean"‘, each draw is exp(eta + variance/2); for ‘"response"‘, each draw is exp(eta + error). Censoring times for future observations are not generated. Identity-scale fits return their historical working scale. These results integrate conditional parameter uncertainty and may differ from the existing plug-in ‘predict.imr()‘ point predictions. Their model weights inherit the approximation described in [posterior_draws()]. For log-time fits, point predictions are medians of the simulated quantities. A time-scale posterior mean need not exist under an inverse-gamma variance mixture, so a sample mean is not reported.

Value

A list of data frames by subgroup, with ‘id‘, ‘predict‘, ‘lower‘ and ‘upper‘. The point prediction is the Monte Carlo mean except for log-time survival fits, where it is the median.

`print.imr` *Print Method for IMR Fits*

Description

Prints a compact overview of the fit, including a platform key (‘P1‘, ‘P2‘, ...) that decodes the availability-subgroup bitstrings. Optionally, it can also print the top-ranked features per platform, ranked by each feature’s maximum mPIP over availability subgroups containing that platform.

Usage

```
## S3 method for class 'imr'
print(x, threshold = 0.5, rank = FALSE, top = 5, ...)
```

Arguments

x	A fitted object of class '"imr"'.
threshold	Inclusion-probability threshold used to count selected features (default '0.5').
rank	Logical; if 'TRUE', print a short ranked feature table for each platform (default 'FALSE').
top	Integer; when 'rank = TRUE', the number of top-ranked features to show per platform (default '5').
...	Unused; present for S3 compatibility.

Value

'x', invisibly.

---

simIMR *Simulated Multi-Platform Example Data*

Description

A small simulated data set illustrating the integrative multi-regression (IMR) setting: three molecular platforms measured on overlapping but partially missing sets of subjects, clinical covariates available for everyone, and three outcome types (continuous, binary and right-censored) driven by the same latent signal.

Usage

```
simIMR
```

Format

A named list with the components:

platforms A list of three data frames, 'genomic' (20 features, subjects 1–240), 'proteomic' (10 features, subjects 1–180) and 'metabolomic' (8 features, subjects 121–300). Each has an 'id' column followed by the feature columns.

covariates A data frame of three clinical covariates ('age', 'sex', 'stage') with an 'id' column, available for all 300 subjects.

outcome The binary outcome data frame (alias of 'outcome.binary'), used as the canonical demonstration outcome.

outcome.binary Data frame with 'id' and a 0/1 response 'y'.

outcome.continuous Data frame with 'id' and a numeric response 'y'.

outcome.survival Data frame with 'id', observed 'time' and event indicator 'status' (1 = event, 0 = censored).

truth A list giving, per platform, the column indices of the features with non-zero effects used to generate the outcomes.

Details

The platforms are observed on the following subjects, which induces four non-empty availability subgroups (bitstrings over genomic/proteomic/ metabolomic): ‘011‘ (120 subjects), ‘111‘ (60), ‘101‘ (60) and ‘100‘ (60).

The generating script is provided in ‘data-raw/make_simIMR.R‘.

Examples

```
data("simIMR", package = "IntegMultiReg")
sapply(simIMR$platforms, dim)
str(simIMR$truth)
```

`summary.imr` *Summarize an IMR Fit*

Description

Produces a per-platform summary of the selected features (those whose marginal posterior inclusion probability exceeds ‘threshold‘ in at least one subgroup), ranked by their maximum inclusion probability.

Usage

```
## S3 method for class 'imr'
summary(object, threshold = 0.5, ...)

## S3 method for class 'summary.imr'
print(x, ...)

summary_imr(object, ...)
```

Arguments

`object`	A fitted object of class ‘"imr"‘.
`threshold`	Inclusion-probability threshold for selection (default ‘0.5‘).
`...`	Unused; present for S3 compatibility.
`x`	A ‘"summary.imr"‘ object.

Value

An object of class ‘"summary.imr"‘: a list with the run metadata and, for each platform, a data frame of selected features with their maximum mPIP and the subgroup achieving it.

validate_imr *Validate a Fitted IMR Model*

Description

Checks the public structure, dimensions and numerical ranges of a fitted IMR object. This is useful after loading a saved fit or before comparing fits.

Usage

```
validate_imr(object)
```

Arguments

object	A fitted ‘"imr"‘ object.

Value

‘TRUE‘, invisibly. Invalid objects fail with an informative error.

validate_imr_data *Validate an IMR Data Object*

Description

Rechecks the structure and subject alignment of an object created by [imr_data()]. Invalid objects fail with an informative error.

Usage

```
validate_imr_data(x)
```

Arguments

x	An ‘"imr_data"‘ object.

Value

‘TRUE‘, invisibly.

Index

∗ **datasets**
    kircIMR, 11
    simIMR, 24

as.data.frame.imr_data, 2

coef.imr, 3
coef.imr_posterior
        *(*imr_posterior_methods*)*, 11
coef_imr *(*coef.imr*)*, 3
compare_imr, 4
confint.imr, 4
confint.imr_posterior
        *(*imr_posterior_methods*)*, 11
cv_imr, 5

imr, 6
imr_data, 10
imr_posterior_methods, 11

kircIMR, 11

plot.imr, 13
plot_imr *(*plot.imr*)*, 13
plot_subgroup_sizes, 15
plot_top_features, 16
posterior_draws, 18
posterior_summary, 20
predict.imr, 20
predict.imr_posterior, 22
predict_imr *(*predict.imr*)*, 20
print.imr, 23
print.imr_posterior
        *(*imr_posterior_methods*)*, 11
print.summary.imr *(*summary.imr*)*, 25

simIMR, 24
summary.imr, 25
summary.imr_posterior
        *(*imr_posterior_methods*)*, 11
summary_imr *(*summary.imr*)*, 25

validate_imr, 26
validate_imr_data, 26