Package: BayGMST (via r-universe)

November 22, 2026

Type Package

Title Bayesian Reconstruction of Global Mean Surface Temperature

Version 0.1.0

Description Fits a Bayesian hierarchical AR(1)-driven state-space model relating annual global mean surface temperature (GMST) to a reduced paleoclimate proxy (computed from cached or user-specified paleoclimate multi-proxy network data), instrumental temperature observations, and radiative forcings (greenhouse gases, volcanic aerosols, and solar irradiance). Unobserved pre-instrumental temperatures are estimated jointly with the model parameters using 'Stan' via the 'instantiate' and 'cmdstanr' packages. Includes tools to reduce a multi-proxy network to a single representative proxy series (via principal component regression, partial least squares, LASSO, or supervised PCR), estimate the parameters in the reconstruction model, run k-fold cross-validation, provide uncertainty quantification on the reconstructed temperatures, and visualize reconstructions and posterior draws. The model and methods follow Barboza et al. (2014) <doi:10.1214/14-AOAS785>, Barboza et al. (2019) <doi:10.1007/s13253-019-00372-4>, and Wang (2020) <doi:10.25335/vb7s-rh49>.

License GPL (>= 3)

Encoding UTF-8

Language en-US

Depends R (>= 4.1)

Imports instantiate, posterior, ggplot2 (>= 3.5.0), dplyr, tidyr

Suggests cmdstanr, glmnet, spls, superpc, dr, patchwork, testthat (>= 3.0.0), knitr, rmarkdown

Additional_repositories https://stan-dev.r-universe.dev/

SystemRequirements CmdStan (>= 2.26.0) -- see the 'instantiate' and 'cmdstanr' packages for installation instructions (https://mc-stan.org/cmdstanr/articles/cmdstanr.html). BayGMST

cannot fit models without a working CmdStan installation; proxy-reduction and forcing-transform utilities do not require it.

URL https://paleopresto.github.io/BayGMST_R/, https://github.com/paleopresto/BayGMST_R

BugReports https://github.com/paleopresto/BayGMST_R/issues

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Tyler Bagwell [aut], Julien Emile-Geay [aut], Frederi Viens [aut], Nick McKay [aut, cre]

Maintainer Nick McKay <nick@nau.edu>

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

Date/Publication 2026-11-21 19:10:15 UTC

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

RemoteRef HEAD

RemoteSha 359dfa61b8e35b3cdfe9e9192e0d18c5b34121a7

Contents

as_baygmst_proxy . . . 3
cv_baygmst . . . 5
fit_baygmst . . . 7
plot.baygmst_fit . . . 9
plot_cv . . . 10
plot_posterior_densities . . . 10
plot_reconstruction . . . 11
plot_trace . . . 11
print.baygmst_fit . . . 12
reconstruct . . . 12
reduce_proxies . . . 13
summary.baygmst_fit . . . 15
transform_forcings . . . 16

Index **18**

as_baygmst_proxy *Coerce a reduced-proxy series into BayGMST's input format*

Description

Normalizes the many shapes a reduced/composited proxy series can arrive in – the output of `reduce_proxies()`, a plain `data.frame`, a bare vector, a time-by-ensemble matrix, or a composite object from another package (such as a `paleoComposite` from the in-development 'compositeR' package) – into a single, year-aware `"baygmst_proxy"` object that `fit_baygmst()` and `cv_baygmst()` accept directly and align by calendar year (rather than by position).

Usage

```
as_baygmst_proxy(x, ...)

## S3 method for class 'baygmst_proxy'
as_baygmst_proxy(x, ...)

## S3 method for class 'baygmst_rp'
as_baygmst_proxy(x, ...)

## S3 method for class 'numeric'
as_baygmst_proxy(x, years, ...)

## S3 method for class 'matrix'
as_baygmst_proxy(x, years, collapse = c("median", "mean"), ...)

## S3 method for class 'data.frame'
as_baygmst_proxy(x, ...)

## S3 method for class 'list'
as_baygmst_proxy(
  x,
  age_units = c("BP", "CE"),
  collapse = c("median", "mean"),
  ...
)
```

Arguments

`x`	The object to coerce. Supported inputs:
	`baygmst_rp` The output of `reduce_proxies()`; uses its `$composite` series.
	`data.frame` Must contain a year column (named `year` or `Year`) and exactly one other numeric column (e.g. the `example_proxy.csv` shipped in `inst/extdata`, with columns `Year, RP1`).
	`numeric` **vector** One value per element of `years`, which must also be supplied.

`matrix` **(or** `data.frame` **of ensemble columns)** Rows are time steps (matching `years`), columns are ensemble members; collapsed to a single series with `collapse`.

composite `list` Any list with elements `$ages` and `$composite` (rows = time steps of `$ages`, columns = ensemble members), the shape produced by compositing packages in the LiPD ecosystem (e.g. `compositeR::compositeEnsembles2()`). Ages are assumed to be in years BP (before present, i.e. before 1950 CE) unless `age_units = "CE"`. The ensemble is collapsed with `collapse`.

`...` Passed between methods.

`years` Numeric vector of calendar years (CE). Required when `x` carries no time axis of its own (bare vectors and matrices); ignored otherwise.

`collapse` How to collapse an ensemble (matrix or composite list) to a single series: `"median"` (default) or `"mean"`, taken across ensemble members within each time step, ignoring `NA`s. Time steps whose collapsed value is not finite (e.g. bins no record covers) are dropped with a message.

`age_units` For composite lists only: `"BP"` (default; ages are years before 1950 CE and are converted via `year = 1950 - age`) or `"CE"` (ages are already calendar years).

Value

An object of class `"baygmst_proxy"`: a `data.frame` with columns `year` (ascending) and `proxy`, plus attributes `source` (a short label for where the series came from) and `n_ensemble` (the number of ensemble members collapsed; `1L` for deterministic inputs).

Scale expectations

The BayGMST 'Stan' model regresses the proxy series on temperature with an estimated intercept and slope, so the proxy does not need to be pre-calibrated to temperature units. The (weakly informative) priors do, however, assume the series is on roughly unit scale: a standardized (z-score) composite or a temperature-anomaly-scale series is appropriate; a raw-unit series (per mil, mm, ...) with large numerical magnitude should be standardized first.

Uncertainty

In this version the ensemble dimension is collapsed before fitting, so compositing/ensemble uncertainty is not propagated into the posterior. A future release may add an ensemble-aware fitting pathway; the `"baygmst_proxy"` object records `n_ensemble` so existing code will not need to change shape when it does.

See Also

`fit_baygmst()`, `reduce_proxies()`

Examples

```
# from a bare vector
p <- as_baygmst_proxy(rnorm(100), years = 1901:2000)

# from a data.frame like inst/extdata/example_proxy.csv
```

```
df <- data.frame(Year = 1901:2000, RP1 = rnorm(100))
p <- as_baygmst_proxy(df)

# from a composite ensemble with ages in yr BP (compositeR-style)
comp <- list(
  ages = seq(950, 0, by = -10),               # 1000-1950 CE
  composite = matrix(rnorm(96 * 20), 96, 20)  # 20 ensemble members
)
p <- as_baygmst_proxy(comp)
head(p)
```

`cv_baygmst` *K-fold cross-validate the BayGMST model over the instrumental period*

Description

Partitions the instrumental (non-NA) years of `instrumental_T` into `nfold` contiguous, roughly equal blocks; for each block in turn, refits `fit_baygmst()`'s underlying model with that block's temperatures hidden (treated as missing/to-be-reconstructed), and reports how well the model recovers them. Ported from the fold loop in the original `cv_v0.1.R` script (see `system.file("legacy-scripts", package = "BayGMST")`).

Usage

```
cv_baygmst(
  proxy,
  instrumental_T,
  forcing_G,
  forcing_V,
  forcing_S,
  years,
  nfold = 3,
  chains = 4,
  parallel_chains = 1,
  iter_warmup = 500,
  iter_sampling = 1500,
  seed = NULL,
  ...
)
```

Arguments

`proxy`	The representative/reduced proxy series. Either a bare numeric vector with one value per year in `years` (aligned by position, no missing values), or any object accepted by `as_baygmst_proxy()` – the output of `reduce_proxies()`, a `data.frame` with year and value columns, or a composite object from another

package (e.g. a 'compositeR' `paleoComposite`) – in which case it is aligned to `years` by calendar year, with an informative error if any requested year is not covered.

`instrumental_T` Numeric vector, instrumental temperature, one value per year in `years`. `NA` for years without instrumental coverage (the pre-instrumental years the model reconstructs).

`forcing_G`, `forcing_V`, `forcing_S`
Numeric vectors of *already transformed* greenhouse-gas, volcanic, and solar forcing – see `transform_forcings()` – one value per year in `years`. No missing values are allowed.

`years` Integer vector of calendar years, defining the row order for all of the above.

`nfold` Number of folds. Default `3`, matching the original script.

`chains`, `parallel_chains`, `iter_warmup`, `iter_sampling`, `seed`
Passed to `cmdstanr`'s `$sample()` method. `iter_sampling` must be at least 1000 (the original script enforced the same minimum).

`...` Additional arguments passed on to `$sample()`.

Details

Two blocks present in the original script were **not** ported: a `Forcings.projections`/RCP-scenario block that was built but never referenced again (dead code, and it hardcoded a path to a specific contributor's machine), and a large commented-out "OLD" duplicate of the fold loop at the end of the file. See `NEWS.md`.

Value

An object of class `"baygmst_cv"`, a list with elements:

`folds` A single combined data.frame (one row per held-out year, across all folds) with columns `year`, `fold`, `T_true`, `T_mean`, `T_lo_inner`, `T_hi_inner`, `T_lo_outer`, `T_hi_outer`.

`r2` Named numeric vector, the posterior mean of the Stan-computed `r2_cv` generated quantity for each fold (fully Bayesian; this differs slightly from the plug-in point-estimate R-squared computed in the original R script – see Details in the package vignette).

`mse` As `r2`, for `mse_cv`.

`years`, `call` Echoed back.

CmdStan required

See `fit_baygmst()`.

See Also

`plot_cv()`

`fit_baygmst` *Fit the BayGMST Bayesian hierarchical AR(1) reconstruction model*

Description

Prepares the data list expected by the `baygmst` 'Stan' model and samples from it with **cmdstanr**, via **instantiate**. Ported from the data-preparation and `mod$sample()` block of the original `BayGMST_v1.0.R` script (see `system.file("legacy-scripts", package = "BayGMST")`), restructured to take its inputs as explicit arguments (rather than reading `config.yml` and hardcoded relative paths) and to return an object rather than leaving variables in the global environment.

Usage

```
fit_baygmst(
  proxy,
  instrumental_T,
  forcing_G,
  forcing_V,
  forcing_S,
  years,
  chains = 4,
  parallel_chains = 1,
  iter_warmup = 500,
  iter_sampling = 1500,
  seed = NULL,
  ...
)
```

Arguments

`proxy`	The representative/reduced proxy series. Either a bare numeric vector with one value per year in `years` (aligned by position, no missing values), or any object accepted by `as_baygmst_proxy()` – the output of `reduce_proxies()`, a `data.frame` with year and value columns, or a composite object from another package (e.g. a 'compositeR' `paleoComposite`) – in which case it is aligned to `years` by calendar year, with an informative error if any requested year is not covered.
`instrumental_T`	Numeric vector, instrumental temperature, one value per year in `years`. `NA` for years without instrumental coverage (the pre-instrumental years the model reconstructs).
`forcing_G`, `forcing_V`, `forcing_S`	Numeric vectors of *already transformed* greenhouse-gas, volcanic, and solar forcing – see `transform_forcings()` – one value per year in `years`. No missing values are allowed.
`years`	Integer vector of calendar years, defining the row order for all of the above.

chains, parallel_chains, iter_warmup, iter_sampling, seed
: Passed to cmdstanr's $sample() method. iter_sampling must be at least 1000 (the original script enforced the same minimum).

...
: Additional arguments passed on to $sample().

Value

An object of class "baygmst_fit", a list with elements:

fit
: The underlying CmdStanMCMC object (see cmdstanr::CmdStanMCMC).

years
: The years argument, echoed back.

idx_obs, idx_mis
: Integer indices (into years) of the instrumental and pre-instrumental years, respectively.

data
: The data list passed to Stan.

cmdstan_version
: The CmdStan version used, for provenance.

call
: The matched call.

Use reconstruct() to extract a tidy reconstruction data.frame, plot() or plot_reconstruction() / plot_trace() / plot_posterior_densities() to visualize it, and summary() for posterior parameter summaries.

CmdStan required

This function requires a working CmdStan installation (via **cmdstanr / instantiate**). Check availability first with instantiate::stan_cmdstan_exists().

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  set.seed(1)
  years <- 1:200
  forcing <- transform_forcings(
    G = 280 + cumsum(rgamma(200, 0.05, 1)),
    V = abs(rnorm(200, 0, 0.05)),
    S = 1361 + rnorm(200, 0, 0.3)
  )
  true_T <- cumsum(rnorm(200, 0, 0.05))
  instrumental_T <- c(rep(NA_real_, 150), true_T[151:200])
  proxy <- true_T + rnorm(200, 0, 0.1)

  fit <- fit_baygmst(
    proxy = proxy,
    instrumental_T = instrumental_T,
    forcing_G = forcing$G,
    forcing_V = forcing$V,
    forcing_S = forcing$S,
    years = years,
    chains = 1,
```

```
    iter_warmup = 200,
    iter_sampling = 1000
  )
  summary(fit)
}
```

`plot.baygmst_fit` *Plot a fitted BayGMST model as a single summary figure*

Description

Reproduces the combined figure of the original `BayGMST_v1.0.R` script (and the PReSto manuscript): the reconstruction from `plot_reconstruction()` on top, and below it three posterior-density panels grouping the proxy coefficient (`alpha1`), the forcing sensitivities (`betaG`, `betaV`, `betaS`), and the autoregressive coefficients (`phi_R`, `phi_T`). The individual pieces remain available through `plot_reconstruction()` and `plot_posterior_densities()`.

Usage

```
## S3 method for class 'baygmst_fit'
plot(
  x,
  title = "GMST Reconstruction using a Reduced Proxy",
  subtitle = NULL,
  ...
)
```

Arguments

`x`	A `"baygmst_fit"` object, as returned by `fit_baygmst()`.
`title`	Figure title. Default `"GMST Reconstruction using a Reduced Proxy"`.
`subtitle`	Figure subtitle. By default, states the instrumental period and the AR(1) model structure; pass `NULL` to omit it.
`...`	Unused; for compatibility with the `plot()` generic.

Details

Requires the 'patchwork' package.

Value

A 'patchwork' object (which is also a ggplot); save it with `ggplot2::ggsave()` if a file is wanted.

See Also

`plot_reconstruction()`, `plot_posterior_densities()`, `plot_trace()`

plot_cv *Plot k-fold cross-validation reconstructions*

Description

Ported from the p_ts_cv plotting block of the original cv_v0.1.R script.

Usage

```
plot_cv(cv)
```

Arguments

cv	A "baygmst_cv" object, as returned by cv_baygmst().

Value

A ggplot object showing each fold's held-out reconstruction against its true instrumental value, faceted by fold.

plot_posterior_densities *Posterior density plots for a fitted BayGMST model*

Description

Ported from the posterior-histogram block of the original BayGMST_v1.0.R script.

Usage

```
plot_posterior_densities(
  fit,
  parameters = c("alpha1", "betaG", "betaV", "betaS", "phi_R", "phi_T")
)
```

Arguments

fit	A "baygmst_fit" object, as returned by fit_baygmst().
parameters	Character vector of Stan parameter names to plot. Default c("alpha1", "betaG", "betaV", "betaS", "phi_R", "phi_T").

Value

A ggplot object overlaying the posterior densities of parameters.

plot_reconstruction *Plot a BayGMST reconstruction*

Description

Ported from the p_ts plotting block of the original BayGMST_v1.0.R script. Unlike the original script, this returns a ggplot object instead of calling ggsave() as a side effect – save it yourself with ggplot2::ggsave() if you want a file.

Usage

```
plot_reconstruction(fit, title = "GMST Reconstruction")
```

Arguments

fit	A "baygmst_fit" object, as returned by fit_baygmst().
title	Plot title. Default "GMST Reconstruction".

Value

A ggplot object.

plot_trace *Trace plots for a fitted BayGMST model*

Description

Ported from the trace-plot block of the original BayGMST_v1.0.R script.

Usage

```
plot_trace(
  fit,
  parameters = c("alpha1", "betaG", "betaV", "betaS", "phi_R", "phi_T")
)
```

Arguments

fit	A "baygmst_fit" object, as returned by fit_baygmst().
parameters	Character vector of Stan parameter names to plot. Default c("alpha1", "betaG", "betaV", "betaS", "phi_R", "phi_T").

Value

A ggplot object, faceted by parameter.

print.baygmst_fit *Print a fitted BayGMST model*

Description

Print a fitted BayGMST model

Usage

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

Arguments

x	A "baygmst_fit" object, as returned by fit_baygmst().
...	Unused; present for S3 consistency.

Value

x, invisibly.

reconstruct *Extract a tidy reconstruction from a fitted BayGMST model*

Description

Builds a tidy data.frame combining the observed instrumental temperatures with the posterior reconstruction of the pre-instrumental years, equivalent to the gmst_reconstruction_data.csv produced as a side effect by the original BayGMST_v1.0.R script – but returned as an object rather than written to disk. Save it yourself (write.csv(reconstruct(fit), "path/to/file.csv")) if you want a file.

Usage

```
reconstruct(fit, probs_inner = c(0.16, 0.84), probs_outer = c(0.025, 0.975))
```

Arguments

fit	A "baygmst_fit" object, as returned by fit_baygmst().
probs_inner, probs_outer	Numeric vectors of length 2, the lower/ upper quantile levels for the "inner" and "outer" credible bands. Defaults c(0.16, 0.84) (68%) and c(0.025, 0.975) (95%), matching the original script.

Value

A `data.frame` with one row per year in `fit$years`, and columns:

`year`	Calendar year.
`type`	`"instrumental"` or `"reconstruction"`.
`T_obs`	Observed instrumental temperature (`NA` for reconstructed years).
`T_mean`	Posterior mean (instrumental years: posterior predictive mean of the fitted value; reconstructed years: posterior mean of the latent temperature).
`T_lo_inner`, `T_hi_inner`	Inner credible band (`probs_inner`).
`T_lo_outer`, `T_hi_outer`	Outer credible band (`probs_outer`).

See Also

`plot_reconstruction()`

`reduce_proxies` *Reduce a multi-proxy network to a single representative proxy series*

Description

Collapses a matrix of paleoclimate proxy records into one (or a small set of) "representative proxy" series suitable for `fit_baygmst()`, following the segmented-composite approach of Barboza et al. (2019) as ported from `utils/PAGES2k_reducedProxy_UNSC.R` (see `system.file("legacy-scripts", package = "BayGMST")` for the original). The proxy network is broken into overlapping time segments of `chunk` years; within each segment, only proxies that already have data are used, and are regressed against instrumental temperature during `calib_years` using the chosen `method`. The final composite series averages across all segments that cover a given year.

Usage

```
reduce_proxies(
  proxy_matrix,
  years,
  temp_calib,
  calib_years,
  method = c("PCR", "LASSO", "sPCR", "SPLS", "SIR"),
  chunk = 250,
  target_adj_r2 = 0.7,
  max_na_frac = 0.05,
  n_components = 3,
  spls_eta = seq(0.1, 0.9, 0.1)
)
```

Arguments

proxy_matrix	Numeric matrix or data.frame of proxy values, one row per year (in the same order as years), one column per proxy record. Do not include a year column here – pass it separately via years.
years	Integer vector of calendar years, one per row of proxy_matrix.
temp_calib	Numeric vector of instrumental temperatures during the calibration period, one value per year in calib_years.
calib_years	Integer vector of calendar years used for calibration (must be a subset of years, and length(calib_years) == length(temp_calib)).
method	One of "PCR" (principal component regression, the default), "LASSO", "sPCR" (supervised PCR), "SPLS" (sparse partial least squares), or "SIR". "SIR" **is experimental** – see Details.
chunk	Segment length, in years. Default 250. Every segment (each of which extends from its own start year through to the *end* of the record, not just chunk years) must fully contain the calibration window, i.e. chunk must be small enough, relative to length(years), that even the last, shortest segment still spans all of calib_years – otherwise reduce_proxies() errors rather than silently comparing mismatched vectors (this mirrors an explicit check present in the original script).
target_adj_r2	For method = "PCR": the calibration-period adjusted R-squared at which to stop adding principal components (the smallest number of PCs reaching this threshold is used; if none reach it, the number maximizing adjusted R-squared is used). Default 0.70.
max_na_frac	Proxies missing more than this fraction of observations within a segment (evaluated both over the full segment and over its calibration window) are dropped from that segment. Default 0.05.
n_components	For method = "sPCR" only: number of supervised principal components to use for prediction. A single value (recycled across segments) or a vector with one value per segment. Default 3.
spls_eta	For method = "SPLS" only: grid of eta sparsity values passed to spls::cv.spls(). Default seq(0.1, 0.9, 0.1).

Value

An object of class "baygmst_rp", a list with components:

segments	A data.frame with columns year, RP1, ..., RPns (one column per time segment).
composite	A data.frame with columns year and RP1, the across-segment average – the single representative proxy series to pass to fit_baygmst() (which accepts the whole object directly, via as_baygmst_proxy()).
proxies	A list with one character vector per segment, naming the columns of proxy_matrix (or giving their indices, if it has no column names) that passed the missingness screen and were used in that segment. Skipped segments get character(0).
method	The method used, echoed back.

Method support

"PCR", "LASSO", and "SPLS" are fully adaptive (all tuning parameters are chosen by calibration-period cross-validation within each segment) and are considered supported. "sPCR" is supported but requires the caller to choose n_components (it is not cross-validated in the underlying **superpc** workflow, matching the original script).

"SIR" is a simplified, **unvalidated** reimplementation using a single sliced-inverse-regression direction from **dr**, with no per-segment variable pre-selection. It intentionally does *not* reproduce the original script's SIR branch, which called an undocumented edrSelec() step from the non-CRAN package 'edrGraphicalTools' that the original author flagged as never fully working ("NEED TO FIGURE OUT HOW TO INSTALL edrGraphicalTools!!"). config.yml in the original repository likewise documented SIR as "still under construction." Treat method = "SIR" results with corresponding caution.

Required packages

"LASSO" requires **glmnet**; "sPCR" requires **superpc**; "SPLS" requires **spls**; "SIR" requires **dr**. These are Suggests, not hard dependencies of BayGMST, and are only required for the method you actually call.

Reproducibility

"LASSO", "sPCR", and "SPLS" use randomized cross-validation internally. Call set.seed() before calling this function if you need reproducible output; reduce_proxies() itself never calls set.seed(), so it never perturbs the caller's random number stream as a side effect.

Examples

```
set.seed(1)
years <- 1:500
proxy_matrix <- cbind(
  p1 = c(rep(NA_real_, 50), cumsum(rnorm(450))),
  p2 = cumsum(rnorm(500))
)
calib_years <- 401:500
temp_calib <- cumsum(rnorm(100)) / 10
rp <- reduce_proxies(
  proxy_matrix, years, temp_calib, calib_years,
  method = "PCR", chunk = 250
)
head(rp$composite)
```

---

summary.baygmst_fit *Summarize a fitted BayGMST model*

Description

Reports posterior summaries for the structural parameters (matching the variable set used in the original BayGMST_v1.0.R script) together with simple instrumental-period fit performance statistics (RMSE, MAE, bias, correlation, and R-squared, both raw and linearly detrended).

Usage

```
## S3 method for class 'baygmst_fit'
summary(object, ...)
```

Arguments

object	A "baygmst_fit" object, as returned by fit_baygmst().
...	Unused; present for S3 consistency.

Value

An object of class "summary.baygmst_fit", a list with elements posterior (a data.frame of parameter posterior summaries), performance (a one-row data.frame of instrumental-period fit statistics), and convergence (a one-row data.frame with max_rhat and min_ess_bulk across the structural parameters; the print method warns when max_rhat exceeds 1.05).

---

transform_forcings *Transform raw radiative forcings for use with* fit_baygmst()

Description

Applies the forcing transformations used throughout the BayGMST pipeline: volcanic forcing is mapped to a negative saturating form, greenhouse-gas (CO2) forcing is log-transformed relative to a reference concentration, and solar forcing is centered on its own mean. Ported from the normalization block near the top of the original BayGMST_v1.0.R script (see system.file("legacy-scripts", package = "BayGMST")).

Usage

```
transform_forcings(G, V, S, co2_c0 = 280, co2_coef = 5.35, vol_coef = 25)
```

Arguments

G	Numeric vector of raw greenhouse-gas forcing (e.g. atmospheric CO2 concentration in ppm).
V	Numeric vector of raw volcanic forcing (e.g. aerosol optical depth). Same length as G.
S	Numeric vector of raw solar forcing (e.g. total solar irradiance). Same length as G.

co2_c0	Reference/baseline CO2 concentration used in the log transform. Default 280 (a common preindustrial reference, ppm).
co2_coef	Multiplier applied to the log CO2 ratio. Default 5.35.
vol_coef	Multiplier applied to the volcanic saturating transform. Default 25.

Details

The transforms applied are:

$$V' = -|\text{vol_coef}| \times (1 - e^{-V})$$

$$G' = \text{co2_coef} \times \log(G/\text{co2_c0})$$

$$S' = S - \bar{S}$$

Value

A list with three numeric vectors, G, V, and S – the transformed forcings, each the same length as the corresponding input.

Examples

```
fc <- transform_forcings(
  G = c(280, 300, 340, 400),
  V = c(0, 0.05, 0.02, 0.10),
  S = c(1360, 1361, 1360.5, 1361.2)
)
fc$G
```

Index

as_baygmst_proxy, 3
as_baygmst_proxy(), *5*, *7*, *14*

cv_baygmst, 5
cv_baygmst(), *3*, *10*

fit_baygmst, 7
fit_baygmst(), *3–6*, *9–14*, *16*

ggplot2::ggsave(), *9*

plot(), *8*, *9*
plot.baygmst_fit, 9
plot_cv, 10
plot_cv(), *6*
plot_posterior_densities, 10
plot_posterior_densities(), *8*, *9*
plot_reconstruction, 11
plot_reconstruction(), *8*, *9*, *13*
plot_trace, 11
plot_trace(), *8*, *9*
print.baygmst_fit, 12

reconstruct, 12
reconstruct(), *8*
reduce_proxies, 13
reduce_proxies(), *3–5*, *7*

summary.baygmst_fit, 15

transform_forcings, 16
transform_forcings(), *6*, *7*