Package: leafwax (via r-universe)

October 15, 2026

Title Bayesian Inversion of Leaf Wax Hydrogen Isotopes to Precipitation

Version 0.4.0

Author Alexander S. Bradley [aut, cre] (<https://orcid.org/0000-0002-4044-2802>)

Maintainer Alexander S. Bradley <abradley@wustl.edu>

Description Full Bayesian inversion of leaf wax hydrogen isotopes to reconstruct precipitation isotopes from hierarchical calibration posteriors. The likelihood is integrated over paired calibration draws and an explicit proper reconstruction prior, without dividing by sampled slopes. Multi-sample records jointly reweight shared calibration draws. A 100-draw fixture is included only for code-path verification; complete posterior deposits are required for inference and are distributed in the companion public data repository.

License MIT + file LICENSE

URL https://github.com/bradleylab/leafwax, https://bradleylab.github.io/leafwax/

BugReports https://github.com/bradleylab/leafwax/issues

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports digest, jsonlite, stats, utils

Suggests rappdirs, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 3.5.0)

LazyData true

LazyDataCompression xz

Config/testthat/edition 3

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

Date/Publication 2026-08-11 01:59:50 UTC

RemoteUrl https://github.com/bradleylab/leafwax

RemoteRef HEAD

RemoteSha 3147c052e94c70f0d98dc7c739a60698e220abf7

Contents

assess_claim . . . 3
available_models . . . 4
batch_predict . . . 5
bayesian_linear_inverse . . . 7
bayesian_record_inverse . . . 8
check_data_cache . . . 9
clear_download_cache . . . 10
compare_models . . . 11
compute_vegetation_envelope . . . 12
d2h_prior_normal . . . 15
d2h_prior_truncated_normal . . . 15
d2h_prior_uniform . . . 16
detect_change . . . 16
detect_model_capabilities . . . 18
download_model_data . . . 18
estimate_temporal_autocorrelation . . . 19
example_data . . . 20
get_all_model_metadata . . . 21
get_cache_dir . . . 22
get_cache_info . . . 22
get_data_path . . . 23
get_data_url . . . 24
get_model_info . . . 24
get_model_parameters . . . 25
invert_d2H . . . 25
invert_d2H_ensemble . . . 28
leafwax_config . . . 28
leafwax_set_config . . . 29
list_cached_models . . . 30
list_model_names . . . 31
list_models . . . 31
load_posteriors . . . 32
local_effective_slope . . . 33
predict_d2h_precip . . . 35
print.leafwax_posterior . . . 37
select_best_model_from_flags . . . 38
validate_inputs . . . 38
validate_model_inputs . . . 39

Index **41**

`assess_claim` *Assess a paleoclimate claim against the leaf-wax taxonomy*

Description

Walks the package's four-level taxonomy and reports the highest level a claim survives at. The taxonomy is:

- Level 1: a leaf-wax delta-2-H change occurred between two intervals. Defensible when the change exceeds analytical uncertainty.
- Level 2: the wax change is consistent with a directional hydroclimate change. Requires (i) sediment-source change ruled out by independent evidence (`sediment_source_ruled_out`), AND (ii) depositional artifact ruled out by independent evidence (`depositional_artifact_ruled_out`), AND EITHER (a) named corroborating evidence against vegetation reorganization via `corroborating_proxies` (the original path), OR (b) demonstration that the observed wax shift exceeds the vegetation-only envelope computed from a user-supplied PFT-change scenario (`level2_vegetation_path`; see `compute_vegetation_envelope()` and manuscript Supplementary Section S8.2).
- Level 3: the wax change implies a quantitative delta-2-H_precip magnitude. Requires a defended local effective slope and explicit uncertainty propagation through the inversion. When `reconstruction` is NULL the function calls `invert_d2H()` itself.
- Level 4: the magnitude is uniquely attributable to precipitation isotope change rather than to vegetation, source-water seasonality, or evapotranspirative enrichment. Requires independent stationarity evidence for each non- precipitation control over the interval.

Usage

```
assess_claim(
  record,
  claim,
  reconstruction = NULL,
  longitude = NULL,
  latitude = NULL,
  model_name = "baseline_sp",
  ...
)
```

Arguments

`record`	Data frame (or list) with at least `d2h_wax` and `age` columns of equal length. `d2h_wax_err` is optional; defaults to `claim$sigma_analytical` per row.
`claim`	Named list specifying the claim. Required fields: `level` (integer 1-4, the level the user is asserting), `interval_baseline` (length-2 numeric c(min, max) age window), `interval_test` (length-2 numeric age window). Optional fields, used by higher levels: `sigma_analytical` (default 3), `rho_t` (default 0; from `estimate_temporal_autocorrelation()`), `beta_eff` (numeric scalar in per

mil wax per per mil precipitation; required at Level 3+), `confidence` (default 0.95), `magnitude_precip` (numeric, the precip-space magnitude the user asserts; required at Level 3+), `sediment_source_ruled_out`, `depositional_artifact_ruled_out` (each a list with `value` (TRUE) and a non-empty `evidence` string; BOTH required at Level 2+ regardless of which Level 2 path is used), `corroborating_proxies` (list, used at Level 2 path (a); the test is non-empty + named), `level2_vegetation_path` (list, used at Level 2 path (b); must contain `vegetation_scenario` = `list(from, to)` with named numeric vectors over `{tree, shrub, grass, C4}` and an optional `evidence` string. The claim must also supply `oipc_ref` (numeric scalar, calibration-period d2H_precip at the site, per mil) at the top level. An optional `level2_vegetation_path$model_name` selects the calibration model used for the envelope; default `"full_interact_sp"`.), `vegetation_stationary`, `seasonal_source_stationar` `evapotranspirative_stationary` (each a list with `value` (TRUE) and a non-empty `evidence` string; required at Level 4).

`reconstruction` Optional output of `invert_d2H(..., return_full = TRUE)` on the record. When NULL and the claim's level is 3 or 4, the function runs the inversion itself.

`longitude, latitude`
Site coordinates, used only when `reconstruction` is NULL.

`model_name` Model to use when running the inversion (default "baseline_sp").

`...` Additional args forwarded to `invert_d2H()`. For an internally constructed Level 3+ reconstruction these must include an explicit `prior`, positive `n_inverse_samples`, and `seed`.

Details

Use this when a colleague claims that a downcore record shows a specific d2H_precip shift and you want a structured check of which levels of the taxonomy that claim actually clears.

Value

A list with elements:

- `highest_level` - integer in 0:4. 0 means even Level 1 did not clear.
- `levels` - data frame, one row per level, with columns `level`, `passed` (logical), `summary` (one-line reason).
- `details` - per-level lists of computed quantities (e.g., delta_wax, threshold, p_exceed, missing fields).
- `claim` - the (validated) claim object.

`available_models` *Get available models*

Description

Returns a list of all available calibration models for leaf wax hydrogen isotope inversion. Models vary in their complexity and data requirements.

Usage

```
available_models()
```

Value

Character vector of available model names. Models include:

- `baseline`: Basic OIPC model without spatial effects
- `baseline_sp`: Basic model with spatial Gaussian process
- `baseline_env`: Includes precipitation-amount effects
- `baseline_env_sp`: Precipitation-amount effects with spatial GP
- `baseline_veg`: Includes vegetation interaction effects
- `baseline_veg_sp`: Vegetation interactions with spatial GP
- `c4_only_sp`: C4 vegetation effects only (spatial)
- `elevation_only_sp`: Historical elevation-context variant (spatial)
- `elevation_c4_sp`: Historical elevation-context + C4 variant
- `elevation_c4_interact_sp`: C4 effect + OIPC x C4 interaction (spatial; elevation spline fitted, not consumed by inversion)
- `full`: Precipitation amount + vegetation interactions without spatial component
- `full_sp`: Precipitation amount + vegetation interactions with spatial component
- `full_interact`: Precipitation amount + vegetation interactions
- `full_interact_sp`: Full interaction model with spatial GP

Examples

```
# List all available models
models <- available_models()
print(models)

# Get details for a specific model
model_info <- get_model_parameters("baseline_sp")
print(model_info$description)
```

---

batch_predict *Jointly invert multiple observations from one record*

Description

Sends all rows through one joint Bayesian inversion so they coherently reweight the shared calibration draws. Chunked and parallel processing are deliberately disabled because splitting a record changes that joint target.

Usage

```
batch_predict(
  data,
  model = "auto",
  chunk_size = 100,
  parallel = FALSE,
  n_cores = NULL,
  progress = TRUE,
  return_diagnostics = FALSE,
  ...
)
```

Arguments

data	Data frame containing observations from one same-site record. A record_id column is required when there is more than one row.
model	Model name or "auto" for automatic selection
chunk_size	Retained for compatibility and recorded in diagnostics; it does not split the joint inversion.
parallel	Must be FALSE; parallel chunks would change the target.
n_cores	Retained for compatibility and diagnostics.
progress	Logical whether to show progress bar
return_diagnostics	Logical whether to return diagnostic information
...	Additional arguments passed to predict_d2h_precip

Value

A leafwax_inverse object for the jointly inverted rows.

Examples

```
## Not run:
local({
  old <- options(leafwax.suppress_preview_warning = TRUE)
  on.exit(options(old))

  data(example_data)
  prior <- d2h_prior_normal(mean = -70, sd = 30)
  large_data <- example_data[rep(seq_len(nrow(example_data)), length.out = 12), ]
  row.names(large_data) <- NULL
  large_data$longitude <- large_data$longitude[[1]]
  large_data$latitude <- large_data$latitude[[1]]
  large_data$record_id <- "example_record"

  # Process in chunks
  results <- batch_predict(
    large_data,
```

```
    chunk_size = 6,
    progress = FALSE,
    prior = prior,
    verbose = FALSE
  )

  # Process with a specific model
  results <- batch_predict(
    large_data,
    model = "baseline_sp",
    chunk_size = 6,
    progress = FALSE,
    prior = prior,
    verbose = FALSE
  )
})

## End(Not run)
```

`bayesian_linear_inverse` *Bayesian inversion of a linear calibration posterior*

Description

Integrates a proper prior with a normal likelihood for paired posterior draws of the intercept, slope, and residual standard deviation. This avoids unstable division by slopes near zero and preserves calibration-draw pairing.

Usage

```
bayesian_linear_inverse(
  y,
  intercept,
  slope,
  residual_sd,
  analytical_sd = 0,
  prior,
  credible_level = 0.9,
  grid_size = 2001L,
  integration_tolerance = 0.001,
  tail_mass_tolerance = 1e-08,
  max_refinements = 4L,
  n_samples = 0L,
  seed = NULL
)
```

Arguments

`y`	Observed response value.
`intercept`, `slope`, `residual_sd`	Paired calibration posterior draws.
`analytical_sd`	Known response-space analytical standard deviation.
`prior`	A proper prior created by one of the `d2h_prior_*()` functions.
`credible_level`	Probability in the central credible interval.
`grid_size`	Odd number of grid points used for numerical integration.
`integration_tolerance`	Maximum permitted change in reported summaries between nested numerical grids, in precipitation-isotope units.
`tail_mass_tolerance`	Maximum permitted posterior mass in either outer grid cell for an unbounded prior.
`max_refinements`	Maximum domain-expansion attempts.
`n_samples`	Number of optional posterior samples to return.
`seed`	Required integer seed when `n_samples` is positive.

Value

A `leafwax_inverse_grid` object containing the normalized posterior, summaries, numerical and identifiability diagnostics, and optional samples.

`bayesian_record_inverse` *Joint Bayesian inversion of a multi-sample record*

Description

Each row has its own latent precipitation-isotope value and proper prior, while all rows share and jointly reweight the paired calibration draws.

Usage

```
bayesian_record_inverse(
  y,
  intercept,
  slope,
  residual_sd,
  analytical_sd = 0,
  prior,
  credible_level = 0.9,
  grid_size = 2001L,
```

```
  integration_tolerance = 0.001,
  tail_mass_tolerance = 1e-08,
  max_refinements = 4L,
  n_samples,
  seed
)
```

Arguments

`y`	Numeric vector of observed response values.
`intercept`, `slope`	Matrices with calibration draws in rows and observations in columns.
`residual_sd`	Positive vector with one value per calibration draw.
`analytical_sd`	Non-negative scalar or one value per observation.
`prior`	A `leafwax_d2h_prior`, applied to every observation, or a list containing one such prior per observation.
`credible_level`	Probability in the central credible interval.
`grid_size`	Odd number of grid points used for numerical integration.
`integration_tolerance`	Maximum permitted change in reported summaries between nested numerical grids, in precipitation-isotope units.
`tail_mass_tolerance`	Maximum permitted posterior mass in either outer grid cell for an unbounded prior.
`max_refinements`	Maximum domain-expansion attempts.
`n_samples`	Number of joint posterior samples. Set to zero for a deterministic summary without samples.
`seed`	Required integer seed when `n_samples` is positive.

Value

A `leafwax_inverse` object with row summaries, diagnostics, marginal grids, joint samples, and sampled calibration-draw identifiers.

`check_data_cache` *Check if model data exists in cache*

Description

Checks whether the heavy posterior file for a model is present in the user cache populated by `download_model_data()`.

Usage

```
check_data_cache(
  model_name,
  data_type = c("minimal", "standard", "full"),
  verbose = FALSE
)
```

Arguments

model_name	Character string specifying the model name.
data_type	Retained for API compatibility. The v0.2 download layout ships a single posterior file per model (posteriors/<model>_posterior.rds) so all values check the same path; the argument is accepted but otherwise ignored.
verbose	Logical, whether to print status messages.

Value

Logical indicating whether the cached posterior file exists.

Examples

```
local({
  old <- options(leafwax.cache_dir = file.path(tempdir(), "leafwax_cache"))
  on.exit(options(old))

  exists <- check_data_cache("baseline_sp", verbose = FALSE)
})
```

---

clear_download_cache *Clear download cache*

Description

Removes downloaded model data from the local cache.

Usage

```
clear_download_cache(
  model_name = NULL,
  type = c("all", "posteriors"),
  confirm = TRUE
)
```

Arguments

model_name	Model name to clear (NULL for all)
type	Type of data to clear: "all" or "posteriors"
confirm	Whether to ask for confirmation

Value

Invisible NULL

Examples

```
local({
  old <- options(leafwax.cache_dir = file.path(tempdir(), "leafwax_cache"))
  on.exit({
    unlink(getOption("leafwax.cache_dir"), recursive = TRUE, force = TRUE)
    options(old)
  })

  post_dir <- file.path(getOption("leafwax.cache_dir"), "posteriors")
  dir.create(post_dir, recursive = TRUE, showWarnings = FALSE)
  file.create(file.path(post_dir, "baseline_sp_posterior.rds"))

  # Clear cache for a specific model without prompting
  suppressMessages(clear_download_cache("baseline_sp", confirm = FALSE))
})
```

---

compare_models *Compare predictions across multiple models*

Description

Runs predictions using multiple models and compares results.

Usage

```
compare_models(
  data,
  models = NULL,
  summary_fun = mean,
  return_all = FALSE,
  progress = TRUE,
  ...
)
```

Arguments

data	Data frame with measurements
models	Character vector of model names to compare
summary_fun	Function to summarize across models (default is mean)
return_all	Logical whether to return all model results
progress	Logical whether to show progress
...	Additional arguments passed to predict_d2h_precip

Value

Data frame with ensemble predictions or list of all results

Examples

```
## Not run:
local({
  old <- options(leafwax.suppress_preview_warning = TRUE)
  on.exit(options(old))

  data(example_data)
  prior <- d2h_prior_normal(mean = -70, sd = 30)

  # Compare multiple models
  comparison <- compare_models(
    example_data,
    models = c("baseline", "baseline_sp"),
    prior = prior,
    progress = FALSE
  )

  # Get all individual model results
  all_results <- compare_models(
    example_data,
    models = c("baseline", "baseline_sp"),
    prior = prior,
    return_all = TRUE,
    progress = FALSE
  )
})

## End(Not run)
```

---

compute_vegetation_envelope

Vegetation-only envelope for a paleo wax-isotope record

Description

Computes the posterior-propagated wax-shift envelope expected under a user-supplied PFT-change scenario, holding `d2H_precip` constant. This is the magnitude path of the package's Level 2 claim taxonomy. The vegetation-only envelope is described in the accompanying manuscript's Supplementary Note 8 (Section S8.2): the calibration's PFT main-effect and PFT-by-$\delta^2 H_p$ interaction coefficients are combined across all posterior draws to bound how much wax change vegetation reorganization alone can produce at the site, with no contribution from precipitation-isotope change.

Usage

```
compute_vegetation_envelope(
  oipc_ref,
  from,
  to,
  model_name = "full_interact_sp",
  n_draws = NULL,
  verbose = TRUE
)
```

Arguments

`oipc_ref`	Numeric scalar, the calibration-period `d2H_precip` at the site (per mil). Typically extracted from the OIPC raster (Bowen and Wilkinson 2002) at the site coordinates; the package does not bundle the raster. Held constant across the envelope by construction.
`from`	Named numeric vector of fractional PFT cover at the baseline interval. Names must be exactly `tree`, `shrub`, `grass`, `C4` (case-sensitive). Values in [0, 1] with `tree + shrub + grass <= 1`; `C4` is an independent fraction.
`to`	Named numeric vector of fractional PFT cover at the test interval. Same name and value constraints as `from`.
`model_name`	Character, name of the calibration model to use. Must contain all eight PFT coefficients (PFT main effects and PFT-by-$\delta^2 H_p$ interactions). Default `"full_interact_sp"` is the only shipped model that satisfies this requirement.
`n_draws`	Integer, number of posterior draws to use. `NULL` (default) uses all available draws. Subsampling is deterministic (stratified, via `load_posteriors()`).
`verbose`	Logical, whether to emit progress messages.

Details

For each posterior draw the per-draw envelope is `envelope = sum_k beta_k * delta_pft_k + sum_k beta_d2Hp_x_k * oi` where `k` indexes the four PFT classes (`tree`, `shrub`, `grass`, `C4`) and `delta_pft_k = to[k] - from[k]`. The $\beta_{\delta^2 H_p} \times PFT$ interaction terms are interactions with the precipitation-isotope calibration slope. The returned `envelope_p975_abs = quantile(abs(envelope_draws), 0.975)` is the absolute upper bound used by `assess_claim()` path (b) to test whether an observed `|delta_wax|` exceeds what the supplied PFT scenario alone can produce.

Passing path (b) rejects the vegetation-only null for the supplied scenario at the site. It does not identify the hydroclimate mechanism, quantify the precipitation-isotope change, or address sediment-source change, depositional artifact, compound-source mixing, age-model errors, evapotranspirative regime change, or seasonality shifts (which `assess_claim()` gates with separate fields). The calibration coefficients are derived from spatial variation across sites; applying them to within-record temporal vegetation change assumes the same response holds through time at one location.

Value

List with

- `envelope_draws` - numeric vector of per-draw signed envelopes (per mil).
- `envelope_median` - posterior median of `envelope_draws`.
- `envelope_p975_abs` - `quantile(abs(envelope_draws), 0.975)`, the absolute upper bound used by `assess_claim()` path (b).
- `oipc_ref` - the `oipc_ref` value used (echoed for traceability).
- `delta_pft` - named numeric vector `to - from`.
- `n_draws_used` - integer, number of draws contributing to `envelope_draws`.
- `model_name` - the model name used.
- `details` - list with `coefs_summary` (posterior median of each of the eight coefficients).

Manuscript reference

Supplementary Note 8 (Section S8.2) of the accompanying manuscript defines the vegetation-only envelope and the constant-precipitation framing. The four-level claim taxonomy is an additional package workflow.

Examples

```
## Not run:
local({
  old <- options(leafwax.suppress_preview_warning = TRUE)
  on.exit(options(old))

  # Hypothetical: a 30 percentage-point woody-to-grass transition at a
  # site where the OIPC raster lookup gave d2H_precip approximately -60 per mil.
  env <- compute_vegetation_envelope(
    oipc_ref = -60,
    from = c(tree = 0.4, shrub = 0.3, grass = 0.2, C4 = 0.05),
    to   = c(tree = 0.1, shrub = 0.2, grass = 0.5, C4 = 0.20),
    model_name = "full_interact_sp",
    n_draws = 100,
    verbose = FALSE
  )
  vegetation_bound <- env$envelope_p975_abs
})

## End(Not run)
```

d2h_prior_normal *Normal prior for precipitation-isotope inversion*

Description

Normal prior for precipitation-isotope inversion

Usage

```
d2h_prior_normal(mean, sd, units = "permil VSMOW")
```

Arguments

mean	Prior mean in precipitation-isotope units.
sd	Positive prior standard deviation.
units	Label for the isotope units.

Value

A proper prior specification for Bayesian inversion.

d2h_prior_truncated_normal *Truncated-normal prior for precipitation-isotope inversion*

Description

Truncated-normal prior for precipitation-isotope inversion

Usage

```
d2h_prior_truncated_normal(mean, sd, lower, upper, units = "permil VSMOW")
```

Arguments

mean	Mean of the untruncated normal distribution.
sd	Positive standard deviation of the untruncated normal distribution.
lower, upper	Finite prior bounds with lower < upper.
units	Label for the isotope units.

Value

A proper prior specification for Bayesian inversion.

d2h_prior_uniform *Uniform prior for precipitation-isotope inversion*

Description

Uniform prior for precipitation-isotope inversion

Usage

```
d2h_prior_uniform(lower, upper, units = "permil VSMOW")
```

Arguments

lower, upper	Finite prior bounds with lower < upper.
units	Label for the isotope units.

Value

A proper prior specification for Bayesian inversion.

detect_change *Within-record d2H_precip change detection*

Description

Given a downcore invert_d2H() reconstruction posterior with full draws (return_full = TRUE), report (a) the posterior probability that the difference in mean d2H_precip between two stratigraphic intervals exceeds user-supplied magnitudes, and (b) the within- record 95\ Section S8.1.1:

Usage

```
detect_change(
  reconstruction,
  age,
  baseline_interval,
  test_intervals = NULL,
  sigma_residual,
  sigma_analytical = 3,
  rho_t = NULL,
  beta_eff,
  confidence = 0.95,
  magnitudes = NULL
)
```

Arguments

reconstruction	Output of invert_d2H(..., return_full = TRUE) on a downcore series. Must contain a posterior_draws matrix of shape n_iter x n_samples.
age	Numeric vector, length n_samples, of sample ages matching the reconstruction columns.
baseline_interval	Length-2 numeric c(min, max) defining the baseline window in age units.
test_intervals	Either a length-2 numeric vector for a single test window, or a named list of length-2 numerics for multiple windows. NULL skips the per-interval probability table and returns only the threshold.
sigma_residual	Numeric, required, the model's posterior residual SD on the leaf-wax per-mil scale (sigma, approximately 16 per mil for the spatial models; see Supplementary Section S8.1.1).
sigma_analytical	Numeric, the analytical uncertainty on d2H_wax measurements in per mil (default 3).
rho_t	Numeric, lag-1 temporal autocorrelation. Use estimate_temporal_autocorrelation() to compute. Defaults to 0 (independent samples) with a message.
beta_eff	Numeric, the local effective slope in per mil wax per per mil precipitation. Use local_effective_slope() for a point estimate (e.g., its median).
confidence	Numeric in (0, 1), the confidence level for the detection threshold. Default 0.95.
magnitudes	Optional numeric vector of magnitudes (per mil) to evaluate posterior Pr(|delta| > magnitude) against.

Details

$$\text{threshold}_{precip} = \frac{z_{\alpha/2}\sqrt{2\sigma^2_{residual}(1-\rho_t) + 2\sigma^2_{analytical}}}{\beta_{\text{eff}}}$$

The threshold is the smallest difference in d2H_precip between two independent samples that can be distinguished from within-record noise at the chosen confidence level. The lag-1 autocorrelation rho_t enters only the residual term, because analytical measurement error is independent between samples by construction. The spatial GP intercept contributes a constant to every sample in the record and cancels in the contrast; the same sigma_residual from the spatial calibration applies (manuscript Supplementary Section S8.1.1).

Value

A list with elements:

- threshold - the detection threshold on d2H_precip at the requested confidence level.
- formula - the components used: z, rho_t, sigma_residual, sigma_analytical, beta_eff.
- intervals - a data frame with one row per test interval reporting the posterior median and CI of delta and (if magnitudes supplied) the posterior probability of exceeding each magnitude.

detect_model_capabilities *Detect model capabilities from static metadata*

Description

Detect model capabilities from static metadata

Usage

```
detect_model_capabilities(model_name)
```

Arguments

model_name	Name of the model

Value

List of capability flags

download_model_data *Download model data from the configured public release*

Description

Downloads model posterior draws from the release configured in `inst/extdata/data_urls.json`. Downloads are accepted only when their byte size and SHA-256 checksum match the configured release manifest.

Usage

```
download_model_data(
  model_name,
  version = "latest",
  data_type = c("posteriors"),
  cache_dir = NULL,
  overwrite = FALSE,
  verify = TRUE,
  verbose = TRUE
)
```

Arguments

model_name	Character string specifying the model name
version	Version tag to download (default "latest")
data_type	Type of data to download (only "posteriors" is currently supported)
cache_dir	Directory to save files (default uses get_cache_dir())
overwrite	Logical whether to overwrite existing files
verify	Logical whether to verify file integrity with checksums
verbose	Logical whether to show progress messages

Value

Logical indicating success

Examples

```
## Not run:
cache_dir <- file.path(tempdir(), "leafwax_download_example")
ok <- download_model_data(
  "baseline",
  version = "latest",
  cache_dir = cache_dir,
  verify = TRUE,
  verbose = FALSE
)
\dontshow{unlink(cache_dir, recursive = TRUE, force = TRUE)}

## End(Not run)
```

---

estimate_temporal_autocorrelation

Estimate lag-1 temporal autocorrelation

Description

Estimate the lag-1 autocorrelation rho_t of a leaf-wax record's residuals after a flat-mean detrend, ordering by age. This is the quantity that enters the within-record detection threshold described in manuscript Supplementary Section S8.1.1 (Var(X1 - X2) = 2 sigma^2 (1 - rho_t)).

Usage

```
estimate_temporal_autocorrelation(
  d2h_wax,
  age,
  method = c("ar1", "lomb_scargle")
)
```

Arguments

d2h_wax	Numeric vector of leaf-wax delta-2-H measurements (per mil).
age	Numeric vector of sample ages, same length as d2h_wax.
method	One of "ar1" or "lomb_scargle".

Details

Two methods are supported:

- "ar1" (default): Pearson correlation of resid[-n] with resid[-1] after age-ordering. For irregularly sampled series this is an approximation; see "lomb_scargle" for an alternative.
- "lomb_scargle": not yet implemented. Returns an error pointing the user at "ar1" until the spectral implementation lands. The plan is to estimate rho_t from the dominant timescale of a Lomb-Scargle periodogram on the irregularly sampled series.

Value

Numeric scalar in [-1, 1], or NA_real_ when the residuals are constant (e.g., n < 3 finite samples).

Examples

```
set.seed(1)
n <- 200
rho <- 0.7
e  <- numeric(n); e[1] <- rnorm(1, 0, 5)
for (k in 2:n) e[k] <- rho * e[k-1] + rnorm(1, 0, 5 * sqrt(1 - rho^2))
ag <- seq(0, 10000, length.out = n)
estimate_temporal_autocorrelation(-150 + e, ag)
```

---

example_data *Example leaf wax hydrogen isotope data*

Description

A 10-row data frame of synthetic leaf wax hydrogen isotope measurements bundled for demonstration and testing.

Usage

```
example_data
```

Format

A data frame with 10 rows and 10 variables:

site_id Character, site identifier

longitude Numeric, longitude in decimal degrees (-180 to 180)

latitude Numeric, latitude in decimal degrees (-90 to 90)

elevation Numeric, elevation in meters above sea level

d2h_wax Numeric, leaf wax hydrogen isotope value in per mil VSMOW

d2h_wax_sd Numeric, analytical uncertainty in per mil

c4_fraction Numeric, C4 vegetation fraction (0-1)

pft_tree Numeric, tree plant functional type fraction (0-1)

pft_shrub Numeric, shrub plant functional type fraction (0-1)

pft_grass Numeric, grass plant functional type fraction (0-1)

Source

Synthetic values designed to span the calibration range.

Examples

```
data(example_data)
head(example_data)
```

---

get_all_model_metadata

Get all model metadata

Description

Returns a list of all 14 models with their descriptions and properties. The models are based on the spatial leafwax hierarchical Bayesian models.

Usage

```
get_all_model_metadata()
```

Value

Named list of available models with metadata

get_cache_dir *Get leafwax data cache directory*

Description

Returns the path to the local cache directory for leafwax model data. Uses rappdirs for platform-specific paths or a user-specified directory.

Usage

```
get_cache_dir(create = TRUE)
```

Arguments

create	Logical, whether to create the directory if it doesn't exist

Value

Character string with the cache directory path

Examples

```
local({
  old <- options(leafwax.cache_dir = file.path(tempdir(), "leafwax_cache"))
  on.exit(options(old))

  cache_dir <- get_cache_dir(create = FALSE)
  dir.exists(cache_dir)
})
```

get_cache_info *Get cache size information*

Description

Reports the disk space used by cached model data.

Usage

```
get_cache_info(by_model = FALSE, by_type = FALSE)
```

Arguments

by_model	Logical, whether to break down by model
by_type	Logical, whether to break down by data type

Value

Data frame with cache size information

Examples

```
local({
  old <- options(leafwax.cache_dir = file.path(tempdir(), "leafwax_cache"))
  on.exit(options(old))

  # Get total cache size
  cache_info <- get_cache_info()

  # Get size by model and type
  cache_info <- get_cache_info(by_model = TRUE, by_type = TRUE)
})
```

---

get_data_path *Get path to data file*

Description

Returns the full path to a data file based on the data source.

Usage

```
get_data_path(filename, data_source = "auto")
```

Arguments

filename	Name of the file
data_source	Source of data: "package", "cache", or "download"

Value

Character string with the file path

get_data_url *Get data download URLs*

Description

Constructs download URLs for the configured model-data release.

Usage

```
get_data_url(model_name, version = "latest", data_type = c("posteriors"))
```

Arguments

model_name	Character string specifying the model name
version	Version tag (e.g., "v1.0.0" or "latest")
data_type	Type of data (only "posteriors" is currently supported)

Value

List of download URLs and filenames

Examples

```
## Not run:
urls <- get_data_url("baseline_sp", "latest")

## End(Not run)
```

get_model_info *Get model info*

Description

Get model info

Usage

```
get_model_info(model_name)
```

Arguments

model_name	Name of the model

Value

List with model metadata

get_model_parameters *Get model parameters*

Description

Returns which parameters each model expects and provides metadata about model capabilities for validation.

Usage

```
get_model_parameters(model_name)
```

Arguments

model_name	Name of the model

Value

List with model parameters and capabilities

invert_d2H *Bayesian inversion of leaf-wax d2H to precipitation d2H*

Description

Performs a likelihood-times-prior inversion over paired calibration posterior draws. No division by a sampled slope, clipping, or post-hoc draw removal is used. Multiple rows jointly reweight the shared calibration draw.

Usage

```
invert_d2H(
  d2H_wax,
  d2H_wax_sd = NULL,
  longitude,
  latitude,
  elevation = NULL,
  elevation_sd = 100,
  c4_fraction = NULL,
  c4_fraction_sd = 10,
  pft_tree = NULL,
  pft_shrub = NULL,
  pft_grass = NULL,
  model_name = "baseline",
  n_posterior_draws = NULL,
  return_full = FALSE,
```

```
  credible_level = 0.9,
  verbose = TRUE,
  record_id = NULL,
  slope = NULL,
  prior = NULL,
  n_inverse_samples = 0L,
  seed = NULL,
  grid_size = 2001L,
  integration_tolerance = 0.001,
  tail_mass_tolerance = 1e-08,
  draw_stability_tolerance = 2
)

invert_d2h(
  d2h_wax,
  d2h_wax_err = NULL,
  longitude,
  latitude,
  elevation = NULL,
  c4_percent = NULL,
  pft_tree = NULL,
  pft_shrub = NULL,
  pft_grass = NULL,
  model_name = "baseline",
  n_draws = NULL,
  return_full = FALSE,
  credible_level = 0.9,
  verbose = TRUE,
  record_id = NULL,
  slope = NULL,
  prior = NULL,
  n_inverse_samples = 0L,
  seed = NULL,
  grid_size = 2001L,
  integration_tolerance = 0.001,
  tail_mass_tolerance = 1e-08,
  draw_stability_tolerance = 2
)
```

Arguments

d2H_wax	Numeric vector of leaf wax d2H values (per mil)
d2H_wax_sd	Numeric vector of measurement uncertainties (per mil)
longitude, latitude	Numeric site coordinates in decimal degrees.
elevation	Optional elevations retained in the output but not consumed by the currently supported inversion designs.
elevation_sd	Elevation uncertainty (not used, kept for compatibility)

`c4_fraction` Numeric vector of C4 vegetation cover as a fraction in `[0, 1]`. The wrapper converts to the percent (0-100) scale used internally before standardisation.

`c4_fraction_sd` C4 fraction uncertainty (not used, kept for compatibility)

`pft_tree`, `pft_shrub`, `pft_grass`
Unsupported PFT inputs; supplying any currently fails closed.

`model_name` One of `baseline`, `baseline_sp`, or `c4_only_sp`.

`n_posterior_draws`
Integer number of posterior draws to use

`return_full` Whether to return joint posterior samples.

`credible_level` Central credible interval probability.

`verbose` Whether to report loading and diagnostic status.

`record_id` Identifier for a single shared-site record. Optional for a one-row inversion and required when `length(d2h_wax) > 1`.

`slope` Optional scalar or paired-draw override in per mil wax per per mil precipitation. It is converted to the fitted model's standardized coefficient internally. Zero and negative values are retained.

`prior` Required proper precipitation-isotope prior from a `d2h_prior_*()` constructor, or one prior per row.

`n_inverse_samples`
Number of joint inverse-posterior samples. Required to be positive when `return_full = TRUE`; otherwise it must be zero.

`seed` Explicit integer seed required for inverse-posterior samples.

`grid_size` Numerical integration grid size.

`integration_tolerance`
Maximum permitted nested-grid summary change.

`tail_mass_tolerance`
Maximum permitted unbounded-prior edge mass.

`draw_stability_tolerance`
Maximum permitted summary change when the paired calibration draw bank is reduced to a deterministic nested half.

`d2h_wax` Numeric vector of observed leaf-wax isotope values in per mil.

`d2h_wax_err` Non-negative analytical standard deviations in per mil.

`c4_percent` C4 cover on a 0–100 scale, required by `c4_only_sp`.

`n_draws` Optional deterministic thinning count for paired calibration posterior draws.

Value

A `leafwax_inverse` object. Its `status` is `"inconclusive"` if numerical or saved-draw stability tolerances fail. Posterior samples are present only when explicitly requested with a seed.

invert_d2H_ensemble *Ensemble predictions across multiple models*

Description

Runs invert_d2H against each model in models and combines the per-draw reconstructions per site, preserving the per-site dimension. Useful for downstream uncertainty estimates that should span structural model uncertainty rather than condition on one calibration variant.

Usage

```
invert_d2H_ensemble(..., models = NULL, ensemble_method = c("equal", "all"))
```

Arguments

...	Arguments passed to invert_d2H (e.g., d2H_wax, d2H_wax_sd, longitude, latitude, optional covariates).
models	Explicit character vector of compatible Bayesian-inversion model names. There is no scientific default ensemble.
ensemble_method	"equal" (default) pools per-draw reconstructions per site across models with equal weighting and returns a per-site posterior. "all" returns the per-model results without pooling.

Value

If ensemble_method = "equal", a list with: posterior_draws (an n_draws x n_sites matrix of pooled per-site, per-draw reconstructions), ensemble_summary (a data frame with one row per site: mean, median, sd, ci_90_lower/ci_90_upper, ci_95_lower/ci_95_upper, n_models_used), model_results (the per-model output from invert_d2H() with return_full = TRUE), models_used (the models actually pooled), and ensemble_method. If ensemble_method = "all", only model_results and ensemble_method are returned.

leafwax_config *Get leafwax configuration*

Description

Returns current configuration options for the leafwax package.

Usage

```
leafwax_config(option = NULL)
```

Arguments

option	Specific option to retrieve (NULL for all)

Value

List of options or single option value

Examples

```
# Get all configuration options
cfg <- leafwax_config()

# Get specific option
auto_download <- leafwax_config("auto_download")
```

---

leafwax_set_config *Set leafwax configuration*

Description

Sets configuration options for the leafwax package.

Usage

```
leafwax_set_config(..., persist = TRUE)
```

Arguments

...	Named arguments for options to set
persist	Logical, whether to show code to make changes permanent

Value

Invisible NULL

Examples

```
# Enable auto-download
old <- options()
leafwax_set_config(auto_download = TRUE, persist = FALSE)
options(old)

# Set multiple options
old <- options()
leafwax_set_config(
  auto_download = TRUE,
  cache_dir = "~/my_leafwax_cache",
  verbose = FALSE,
  persist = FALSE
```

```
)
options(old)
```

`list_cached_models` *List available models in cache*

Description

Lists all models that have been downloaded to the local cache.

Usage

```
list_cached_models(data_type = NULL, verbose = TRUE)
```

Arguments

`data_type`	Filter by data type (NULL for any)
`verbose`	Logical, whether to print detailed information

Value

Character vector of available model names

Examples

```
local({
  old <- options(leafwax.cache_dir = file.path(tempdir(), "leafwax_cache"))
  on.exit(options(old))

  # List all cached models
  models <- list_cached_models(verbose = FALSE)

  # List models with full data
  models_full <- list_cached_models(data_type = "full", verbose = FALSE)
})
```

list_model_names *List model names*

Description

Returns the names of every model the package can resolve. Prefers the heavy posteriors directory when present (development install), otherwise falls back to the lightweight posteriors directory that ships with every install. The user cache is intentionally not enumerated here so the answer is stable regardless of what has been downloaded.

Usage

```
list_model_names()
```

Value

Character vector of model names. Empty if neither directory contains posterior files.

list_models *List available models with details*

Description

Returns information about the calibration models available in the leafwax package, including which covariates each model uses.

Usage

```
list_models(check_data = TRUE, verbose = TRUE)
```

Arguments

check_data	Logical, whether to check if model data is available
verbose	Logical, whether to print formatted output

Value

Data frame with model information

Examples

```
# List all models
models <- list_models(verbose = FALSE)
models_head <- head(models)
```

`load_posteriors` *Load posterior draws for a model*

Description

Loads posterior draws for one of the 14 leafwax calibration models. The function searches three tiers in order:

Usage

```
load_posteriors(model_name, n_draws = NULL, verbose = TRUE)
```

Arguments

`model_name`	Character string specifying the model name.
`n_draws`	Integer number of posterior draws to use, or `NULL` for all available. Requesting more draws than are present silently returns whatever is available (e.g. all 100 from the preview tier).
`verbose`	Logical indicating whether to print loading info.

Details

1. **Heavy** posteriors at `inst/extdata/posteriors/` (complete retained draws in a validated development checkout; excluded from the tarball).
2. **Cache** populated by a validated public data release under `get_cache_dir()`.
3. **Preview** posteriors at `inst/extdata/posteriors_light/`. These are a 100-draw stratified subsample shipped with every install so examples and tests run offline. They are intended as a fixture for code-path verification, **not** for inference: tail probabilities and 95% credible intervals are noisy at this sample size. The package warns when this tier is loaded, and inferential functions fail closed.

Use `download_model_data()` once to populate the verified cache before inferential inversion from an installed package.

Value

A `leafwax_posterior` object: a list with `draws`, `metadata` (including `metadata$tier`, one of "heavy", "cache", "light"), optional `spatial`, and accessor closures.

Examples

```
local({
  old <- options(leafwax.suppress_preview_warning = TRUE)
  on.exit(options(old))

  # Load a model (preview tier on a fresh install)
  model <- load_posteriors("baseline", verbose = FALSE)
```

```
  # Spatial model with limited draws
  model_fast <- load_posteriors("baseline_sp", n_draws = 50, verbose = FALSE)
})
```

`local_effective_slope` *Local effective slope at a paleo-reconstruction site*

Description

Returns a per-draw vector of the local $\beta_{\delta^2 H_p}$ calibration slope at a single site, combining the global precipitation-isotope slope with the spatial slope GP prediction at that site. The fitted coefficient is converted from standard deviations of wax per standard deviation of precipitation to per mil wax per per mil precipitation. Every posterior draw is preserved; only its units change.

Usage

```
local_effective_slope(
  longitude,
  latitude,
  model_name,
  override = NULL,
  n_draws = NULL,
  verbose = FALSE
)
```

Arguments

`longitude`	Numeric, single longitude in decimal degrees.
`latitude`	Numeric, single latitude in decimal degrees.
`model_name`	Character, calibration model name (see `available_models()`). Must be a spatial model (`*_sp`) for the site-specific slope to differ from the global mean; non-spatial models return the global posterior unchanged.
`override`	Optional numeric in per mil wax per per mil precipitation. NULL (default) uses the model slope. A single value broadcasts across all draws. A vector of length `n_draws` is used per draw.
`n_draws`	Integer, optional number of posterior draws to use (`NULL` uses all). Forwarded to `load_posteriors()`.
`verbose`	Logical, whether to print progress messages.

Details

Two modes:

- Default: returns the model's per-draw slope at the site.
- Override (single value or per-draw vector) replaces the model slope with a defended local value (e.g., from independent evidence about source-water seasonality, leaf-water enrichment, or vegetation).

The Bayesian inversion propagates its paired local-slope draws internally. Pass this vector as `slope = ...` only when deliberately overriding that internal calculation with a separately defended slope posterior.

Mechanistic reference values (e.g. the simple two-pool stationarity bound `alpha = 1 + epsilon_app/1000` ~ 0.88 under `epsilon_app ~= -120 permil`; Sessions 2005) are documented for interpretation but are never applied to the returned draws. The frequency of draws above any chosen reference is computable directly from the returned vector (`mean(slope > 0.88)`) and carries scientific information about how often the calibration implicates non-stationarity at the site.

Value

Numeric vector of length `n_draws`, the per-draw effective slope at the site in per mil wax per per mil precipitation (after override, if any).

Examples

```
## Not run:
local({
  old <- options(leafwax.suppress_preview_warning = TRUE)
  on.exit(options(old))

  # St. Louis with the baseline_sp model
  s <- local_effective_slope(
    longitude = -90, latitude = 38,
    model_name = "baseline_sp",
    n_draws = 200
  )
  slope_summary <- summary(s)

  # How often does the calibration imply a slope above the simple-model
  # stationarity bound at this site?
  fraction_above_bound <- mean(s > 0.88)

  # Override with a defended local slope
  s_fixed <- local_effective_slope(
    longitude = -90, latitude = 38,
    model_name = "baseline_sp",
    override = 0.55
  )

  # Pass through to the inversion. The slope vector and the
  # inversion's posterior must use the same n_draws: pair
  # local_effective_slope(..., n_draws = N) with
  # invert_d2H(..., n_posterior_draws = N, slope = s), or pass a
  # single point estimate (e.g., median(s)).
  result <- invert_d2H(d2H_wax = -180, d2H_wax_sd = 3,
                       longitude = -90, latitude = 38,
                       model_name = "baseline_sp",
                       n_posterior_draws = 200,
                       slope = s,
                       prior = d2h_prior_normal(-70, 30),
                       verbose = FALSE)
```

```
})

## End(Not run)
```

`predict_d2h_precip` *Predict precipitation d2H from leaf wax d2H*

Description

Main user-facing function for inverting leaf wax hydrogen isotopes to precipitation isotopes. Automatically selects appropriate model based on available data and returns results in a tidy format.

Usage

```
predict_d2h_precip(
  data = NULL,
  d2h_wax = NULL,
  longitude = NULL,
  latitude = NULL,
  d2h_wax_err = NULL,
  elevation = NULL,
  c4_fraction = NULL,
  pft_tree = NULL,
  pft_shrub = NULL,
  pft_grass = NULL,
  record_id = NULL,
  model = "auto",
  n_draws = NULL,
  credible_level = 0.9,
  return_draws = FALSE,
  progress = TRUE,
  verbose = TRUE,
  prior = NULL,
  n_inverse_samples = 0L,
  seed = NULL,
  grid_size = 2001L,
  integration_tolerance = 0.001,
  tail_mass_tolerance = 1e-08,
  draw_stability_tolerance = 2
)
```

Arguments

`data`	Data frame containing measurements, or NULL to use individual vectors
`d2h_wax`	Numeric vector of leaf wax d2H values (per mil)
`longitude`	Numeric vector of longitudes (decimal degrees)
`latitude`	Numeric vector of latitudes (decimal degrees)

d2h_wax_err	Numeric vector of measurement uncertainties (optional)
elevation	Numeric vector of elevations in meters (optional)
c4_fraction	Numeric vector of C4 vegetation fraction 0-1 (optional)
pft_tree	Numeric vector of tree PFT fraction (optional)
pft_shrub	Numeric vector of shrub PFT fraction (optional)
pft_grass	Numeric vector of grass PFT fraction (optional)
record_id	Optional identifier for one same-site record; required for multi-row input.
model	Character string specifying model, or "auto" for automatic selection
n_draws	Integer number of posterior draws (NULL for all)
credible_level	Numeric credible interval level (default 0.9)
return_draws	Logical whether to return full posterior draws
progress	Logical whether to show progress bar for batch processing
verbose	Logical whether to print status messages
prior	Required proper precipitation-isotope prior.
n_inverse_samples	Number of joint posterior samples when return_draws = TRUE.
seed	Explicit seed required when posterior samples are requested.
grid_size, integration_tolerance, tail_mass_tolerance	Numerical integration controls passed to invert_d2H().
draw_stability_tolerance	Saved-draw stability tolerance in per mil.

Value

A leafwax_inverse object. Its summary data frame contains the posterior median, central interval, and supporting moments for each row; diagnostics and method metadata are always returned. Joint posterior_draws are included only when return_draws = TRUE with an explicit sample count and seed.

The interval is the posterior predictive specified in manuscript Supplementary Note 8 (Section S8.1; analytical uncertainty plus the model's posterior residual SD).

Examples

```
## Not run:
local({
  old <- options(leafwax.suppress_preview_warning = TRUE)
  on.exit(options(old))

  # Using data frame input
  data(example_data)
  prior <- d2h_prior_normal(mean = -70, sd = 30)
  results <- predict_d2h_precip(
    example_data, prior = prior, verbose = FALSE
  )
```

```
  # Using individual vectors
  results <- predict_d2h_precip(
    d2h_wax = c(-150, -140, -130),
    longitude = rep(-90, 3),
    latitude = rep(38, 3),
    record_id = "example_record",
    elevation = c(1000, 1500, 500), prior = prior,
    verbose = FALSE
  )

  # Specify model explicitly
  results <- predict_d2h_precip(
    example_data,
    model = "baseline_sp", prior = prior,
    verbose = FALSE
  )

  # Get full posterior draws
  results <- predict_d2h_precip(
    example_data,
    prior = prior, return_draws = TRUE,
    n_inverse_samples = 1000, seed = 20260801,
    verbose = FALSE
  )
})

## End(Not run)
```

`print.leafwax_posterior` *Print method for leafwax_posterior*

Description

Print method for leafwax_posterior

Usage

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

Arguments

`x`	A leafwax_posterior object
`...`	Additional arguments

Value

The input `leafwax_posterior` object `x`, invisibly. Called for its side effect of printing a one-line model summary to the console.

select_best_model_from_flags *Select best model based on available data*

Description

Automatically selects the most appropriate model name from the 14 shipped variants given which covariates the user has available. Spatial-aware models are preferred when `prefer_spatial = TRUE`.

Usage

```
select_best_model_from_flags(
  has_elevation = FALSE,
  has_c4 = FALSE,
  has_pft = FALSE,
  prefer_spatial = TRUE,
  verbose = FALSE
)
```

Arguments

`has_elevation`	Logical, whether elevation data is available. Accepted for compatibility; the reconstruction interface does not consume fitted elevation coefficients, so elevation alone does not change the selected model.
`has_c4`	Logical, whether C4 vegetation data is available
`has_pft`	Logical, whether PFT data is available
`prefer_spatial`	Logical, whether to prefer spatial models
`verbose`	Logical, whether to print selection reasoning

Value

Character string with the selected model name

validate_inputs *Validate input data for inversion*

Description

Checks that input data meets requirements for the specified model and returns cleaned, validated data.

Usage

```
validate_inputs(
  d2h_wax,
  longitude,
  latitude,
  d2h_wax_err = NULL,
  elevation = NULL,
  c4_fraction = NULL,
  pft_tree = NULL,
  pft_shrub = NULL,
  pft_grass = NULL,
  model_name = "baseline"
)
```

Arguments

d2h_wax	Leaf wax d2H values
longitude	Longitude values
latitude	Latitude values
d2h_wax_err	Measurement uncertainties
elevation	Elevation values
c4_fraction	C4 vegetation fraction
pft_tree	Tree PFT fraction
pft_shrub	Shrub PFT fraction
pft_grass	Grass PFT fraction
model_name	Name of model to use

Value

List of validated inputs

---

validate_model_inputs *Validate inputs for a specific model*

Description

Checks that all required predictors are provided and warns about unused ones.

Usage

```
validate_model_inputs(
  model_name,
  d2h_wax,
  longitude,
  latitude,
  elevation = NULL,
  c4_percent = NULL,
  pft_tree = NULL,
  pft_shrub = NULL,
  pft_grass = NULL,
  verbose = TRUE
)
```

Arguments

model_name	Name of the model
d2h_wax	Leaf wax d2H values
longitude	Longitude values
latitude	Latitude values
elevation	Elevation values (optional)
c4_percent	C4 percentage values (optional)
pft_tree	Tree PFT fraction (optional)
pft_shrub	Shrub PFT fraction (optional)
pft_grass	Grass PFT fraction (optional)
verbose	Whether to print validation messages

Value

List with validation results

Index

∗ **datasets**
example_data, 20

assess_claim, 3
assess_claim(), *13, 14*
available_models, 4

batch_predict, 5
bayesian_linear_inverse, 7
bayesian_record_inverse, 8

check_data_cache, 9
clear_download_cache, 10
compare_models, 11
compute_vegetation_envelope, 12
compute_vegetation_envelope(), *3*

d2h_prior_normal, 15
d2h_prior_truncated_normal, 15
d2h_prior_uniform, 16
detect_change, 16
detect_model_capabilities, 18
download_model_data, 18
download_model_data(), *9*, *32*

estimate_temporal_autocorrelation, 19
example_data, 20

get_all_model_metadata, 21
get_cache_dir, 22
get_cache_dir(), *32*
get_cache_info, 22
get_data_path, 23
get_data_url, 24
get_model_info, 24
get_model_parameters, 25

invert_d2H, 25, *28*
invert_d2h *(*invert_d2H*)*, 25
invert_d2H(), *36*
invert_d2H_ensemble, 28

leafwax_config, 28
leafwax_set_config, 29
list_cached_models, 30
list_model_names, 31
list_models, 31
load_posteriors, 32
load_posteriors(), *13*
local_effective_slope, 33

predict_d2h_precip, 35
print.leafwax_posterior, 37

select_best_model_from_flags, 38

validate_inputs, 38
validate_model_inputs, 39