

Package: leafwax (via r-universe)

June 16, 2026

Title Bayesian Inversion of Leaf Wax Hydrogen Isotopes to Precipitation

Version 0.2.7

Description Bayesian inversion of leaf wax hydrogen isotopes to reconstruct precipitation isotopes using hierarchical spatial models. Provides fourteen Bayesian models that vary in their use of spatial Gaussian processes and ancillary covariates (precipitation amount, plant functional type, C4 fraction). Models are pre-computed using 'Stan' and stored as posterior distributions, so prediction does not require 'Stan' to be installed. A 100-draw fixture ships with the package; full 1000-draw posteriors are downloaded from a versioned 'Zenodo' deposit on first use; see Bradley (2026) [<doi:10.5281/zenodo.20085465>](https://doi.org/10.5281/zenodo.20085465).

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 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 2026-06-08 02:34:58 UTC

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

RemoteRef HEAD

RemoteSha 085737c97e94a9d0af593d0b4acb8f7a9b63e24c

Contents

assess_claim	3
available_models	4
batch_predict	5
check_data_cache	7
clear_download_cache	8
compare_models	9
compute_vegetation_envelope	10
detect_change	12
detect_model_capabilities	13
download_model_data	14
estimate_temporal_autocorrelation	15
example_data	16
get_all_model_metadata	17
get_cache_dir	17
get_cache_info	18
get_data_path	19
get_data_url	19
get_model_info	20
get_model_parameters	20
invert_d2h	21
invert_d2H_ensemble	24
leafwax_config	25
leafwax_set_config	25
list_cached_models	26
list_model_names	27
list_models	27
load_posteriors	28
local_effective_slope	29
predict_d2h_precip	31
print.leafwax_posterior	33
select_best_model_from_flags	33
validate_inputs	34
validate_model_inputs	35

Index

36

assess_claim

Assess a paleoclimate claim against the leaf-wax taxonomy

Description

Walks the four-level taxonomy from manuscript Section 4.5.6 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 Section 4.5.3).
- 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

<code>record</code>	Data frame (or list) with at least <code>d2h_wax</code> and <code>age</code> columns of equal length. <code>d2h_wax_err</code> is optional; defaults to <code>claim\$sigma_analytical</code> per row.
<code>claim</code>	Named list specifying the claim. Required fields: <code>level</code> (integer 1-4, the level the user is asserting), <code>interval_baseline</code> (length-2 numeric c(min, max) age window), <code>interval_test</code> (length-2 numeric age window). Optional fields, used by higher levels: <code>sigma_analytical</code> (default 3), <code>rho_t</code> (default 0; from <code>estimate_temporal_autocorrelation()</code>), <code>beta_eff</code> (numeric scalar; 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_stationary, 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() (e.g., elevation, c4_fraction, pft_*, n_posterior_draws).

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	<i>Get available models</i>
------------------	-----------------------------

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`: Historical elevation/C4-interaction name; C4 effect only
- `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

Batch predict precipitation d2H for multiple sites

Description

Processes multiple sites with progress indicators and optional parallelization. Handles large datasets efficiently by processing in chunks.

Usage

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

Arguments

data	Data frame containing all measurements
model	Model name or "auto" for automatic selection
chunk_size	Number of sites to process at once (default 100)
parallel	Logical whether to use parallel processing
n_cores	Number of cores for parallel processing (NULL for auto)
progress	Logical whether to show progress bar
return_diagnostics	Logical whether to return diagnostic information
...	Additional arguments passed to predict_d2h_precip

Value

Data frame with predictions for all sites

Examples

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

  data(example_data)
  large_data <- example_data[rep(seq_len(nrow(example_data)), length.out = 12), ]
  row.names(large_data) <- NULL

  # Process in chunks
  results <- batch_predict(
    large_data,
    chunk_size = 6,
    progress = FALSE,
    verbose = FALSE
  )

  # Process with a specific model
  results <- batch_predict(
```

```
    large_data,  
    model = "baseline_sp",  
    chunk_size = 6,  
    progress = FALSE,  
    verbose = FALSE  
  )  
})
```

check_data_cache	<i>Check if model data exists in cache</i>
------------------	--

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	<i>Compare predictions across multiple models</i>
----------------	---

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

```
local({  
  old <- options(leafwax.suppress_preview_warning = TRUE)  
  on.exit(options(old))  
  
  data(example_data)  
  
  # Compare multiple models  
  comparison <- compare_models(  
    example_data,  
    models = c("baseline", "baseline_env", "baseline_sp"),  
    progress = FALSE  
  )  
  
  # Get all individual model results
```

```

all_results <- compare_models(
  example_data,
  models = c("baseline", "baseline_env"),
  return_all = TRUE,
  progress = FALSE
)
})

```

```
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 Level 2 claim taxonomy described in the accompanying manuscript Section 4.5.3: 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 <code>load_posteriors()</code>).
verbose	Logical, whether to emit progress messages.

Details

For each posterior draw the per-draw envelope is $\text{envelope} = \sum_k \beta_k * \text{delta_pft_k} + \sum_k \beta_{d2Hp_x_k} * \text{oipc_ref}$ where k indexes the four PFT classes (tree, shrub, grass, C4) and $\text{delta_pft_k} = \text{to}[k] - \text{from}[k]$. 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 $|\text{delta_wax}|$ 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

Section 4.5.3 of the accompanying manuscript defines the vegetation-only envelope and the constant-precipitation framing. Section 4.5.6 places it in the Level 2 claim taxonomy.

Examples

```
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
})

```

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% CI (4.5.3):

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 Section 4.5.3).

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. 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 $\Pr(\delta > \text{magnitude})$ against.

Details

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

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 Section 4.5.3).

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 v10 metadata

Description

Detect model capabilities from static v10 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 GitHub releases*

Description

Downloads model posterior draws and lookup tables from GitHub releases with progress tracking and integrity verification.

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

```
cache_dir <- file.path(tempdir(), "leafwax_download_example")
ok <- download_model_data(
  "baseline",
  version = "v1.0.1",
  cache_dir = cache_dir,
  verify = FALSE,
  verbose = FALSE
)
```

```
estimate_temporal_autocorrelation
```

Estimate lag-1 temporal autocorrelation

Description

Estimate the lag-1 autocorrelation ρ_t of a leaf-wax record's residuals after a flat-mean de-trend, ordering by age. This is the quantity that enters the within-record detection threshold from manuscript Section 4.5.3 ($\text{Var}(X_1 - X_2) = 2 \sigma^2 (1 - \rho_t)$).

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 $\text{resid}[-n]$ with $\text{resid}[-1]$ 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 ρ_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	<i>Get cache size information</i>
----------------	-----------------------------------

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	<i>Get path to data file</i>
---------------	------------------------------

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	<i>Get data download URLs</i>
--------------	-------------------------------

Description

Constructs download URLs for model data from GitHub releases.

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

```
# Get URLs for latest version
urls <- get_data_url("baseline_sp", "latest")

# Get URLs for specific version
urls <- get_data_url("baseline_sp", "v1.0.1")
```

`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

*Invert leaf wax d2H to precipitation d2H***Description**

Uses Bayesian posterior draws to invert leaf wax hydrogen isotope values to precipitation isotope values, accounting for all fitted model components including vegetation effects and spatial correlations where applicable.

Usage

```
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
)
```

```
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
)

```

Arguments

d2h_wax	Numeric vector of leaf wax d2H values (per mil)
d2h_wax_err	Numeric vector of measurement uncertainties (per mil)
longitude	Numeric vector of longitudes (decimal degrees)
latitude	Numeric vector of latitudes (decimal degrees)
elevation	Numeric vector of elevations (meters)
c4_percent	Numeric vector of C4 vegetation percentage (0-100)
pft_tree	Numeric vector of tree PFT fraction (0-1)
pft_shrub	Numeric vector of shrub PFT fraction (0-1)
pft_grass	Numeric vector of grass PFT fraction (0-1)
model_name	Character string specifying which model to use
n_draws	Integer number of posterior draws to use (NULL for all)
return_full	Logical whether to return full posterior draws or just summary
credible_level	Numeric credible interval level (default 0.9)
verbose	Logical whether to print progress messages
record_id	Character or numeric, optional record identifier. When supplied and constant across all input rows, all rows are treated as belonging to the same downcore series: the spatial Gaussian process is evaluated once per posterior draw at the shared site, so spatial draws are reused across the series rather than redrawn per row. The current implementation already shares spatial draws between identical (longitude, latitude) pairs; the record_id argument adds explicit validation that the caller intends within-record inference.
slope	Optional numeric override for the d2H_wax-d2H_precip slope. NULL (default) uses the model's site-specific slope, i.e., the local $\beta_{\delta^2 H_p}$ calibration slope including the spatial slope GP perturbation at the site. A single numeric replaces the slope with a fixed point estimate (broadcast across all posterior draws). A vector of length n_draws is used per draw. Use local_effective_slope() to build a defensible per-draw override from the calibration's site-specific posterior. When supplied, the override applies uniformly to every input row.
d2H_wax	Numeric vector of leaf wax d2H values (per mil)
d2H_wax_sd	Numeric vector of measurement uncertainties (per mil)
elevation_sd	Elevation uncertainty (not used, kept for compatibility)
c4_fraction	Numeric vector of C4 vegetation cover as a fraction in $[\theta, 1]$. The wrapper converts to the percent (0-100) scale used internally before standardisation.
c4_fraction_sd	C4 fraction uncertainty (not used, kept for compatibility)
n_posterior_draws	Integer number of posterior draws to use

Value

If `return_full` is `FALSE`, a data frame with columns:

<code>d2h_precip_mean</code>	Mean predicted precipitation d2H
<code>d2h_precip_median</code>	Median predicted precipitation d2H
<code>d2h_precip_sd</code>	Standard deviation of the posterior predictive interval
<code>d2h_precip_lower</code>	Lower bound of the credible interval
<code>d2h_precip_upper</code>	Upper bound of the credible interval
<code>prediction_interval_width</code>	Width of the credible interval (upper - lower).

The interval is the posterior predictive specified in manuscript supplement Section S4.1, Eq. 7: the wax-error draw combines analytical uncertainty with the model's posterior residual SD σ . For within-record change detection, the spatial GP intercept's contribution cancels in any contrast computed from the returned `posterior_draws` (manuscript Section 4.5.3); the same σ applies in both regimes.

If `return_full` is `TRUE`, a list with:

<code>summary</code>	The summary data frame described above
<code>posterior_draws</code>	Matrix of all posterior draws (<code>n_draws</code> x <code>n_locations</code>)
<code>model_info</code>	Information about the model used.

Examples

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

  # Simple inversion with base model
  results <- invert_d2h(
    d2h_wax = c(-150, -140, -130),
    d2h_wax_err = c(3, 3, 3),
    longitude = c(-120, -110, -100),
    latitude = c(40, 35, 30),
    elevation = c(1000, 1500, 500),
    model_name = "baseline",
    verbose = FALSE
  )

  # Inversion with spatial model
  results <- invert_d2h(
    d2h_wax = c(-150, -140, -130),
    d2h_wax_err = c(3, 3, 3),
    longitude = c(-120, -110, -100),
```

```

    latitude = c(40, 35, 30),
    elevation = c(1000, 1500, 500),
    model_name = "baseline_sp",
    return_full = TRUE,
    verbose = FALSE
  )
})

```

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 = c("full_sp", "full_interact_sp", "elevation_c4_interact_sp"),
  ensemble_method = c("equal", "all")
)

```

Arguments

<code>...</code>	Arguments passed to <code>invert_d2H</code> (e.g., <code>d2H_wax</code> , <code>d2H_wax_sd</code> , <code>longitude</code> , <code>latitude</code> , optional covariates).
<code>models</code>	Character vector of v10 model names to include in the ensemble. Defaults to three structurally distinct variants: <code>full_sp</code> (all covariates + spatial GP), <code>full_interact_sp</code> (full + elevation x C4 interaction + spatial GP), and <code>elevation_c4_interact_sp</code> (elevation x C4 interaction with spatial GP, no PFT).
<code>ensemble_method</code>	"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	<i>Get leafwax configuration</i>
----------------	----------------------------------

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	<i>Set leafwax configuration</i>
--------------------	----------------------------------

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	<i>List model names</i>
------------------	-------------------------

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	<i>List available models with details</i>
-------------	---

Description

Returns information about the v10 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	<i>Load posterior draws for a model</i>
-----------------	---

Description

Loads posterior draws for one of the 14 leafwax v10 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/` (1000 draws, development install only; excluded from the CRAN tarball).
2. **Cache** populated by `download_model_data()` 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 issues a warning whenever the preview tier is in use; downstream functions (`invert_d2H()`, `assess_claim()`, `detect_change()`) repeat the warning so it is visible at the call that actually matters.

For inference, run `download_model_data()` once to populate the cache and then call `load_posteriors()` again – the cache tier wins over the preview tier and no further downloads are needed.

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 returned vector is the raw posterior at the site; every draw the calibration produced is preserved without modification.

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, v10 model name (see <code>available_models()</code>). 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. NULL (default) uses the model slope. A single value broadcasts across all draws. A vector of length <code>n_draws</code> is used per draw.
n_draws	Integer, optional number of posterior draws to use (NULL uses all). Forwarded to <code>load_posteriors()</code> .
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).

Pass the returned vector to `invert_d2H(..., slope = ...)` to propagate it through the inversion.

Mechanistic reference values (e.g. the simple two-pool stationarity bound $\alpha = 1 + \text{epsilon_app}/1000 \sim 0.88$ under $\text{epsilon_app} \sim -120$ 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 (after override, if any).

Examples

```
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,
    verbose = FALSE)
})
```

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,
  model = "auto",
  n_draws = NULL,
  credible_level = 0.9,
  return_draws = FALSE,
  progress = TRUE,
  verbose = TRUE
)
```

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)
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

Value

A data frame with predictions (or list if `return_draws = TRUE`):

d2h_precip_mean Mean predicted precipitation d2H
d2h_precip_median Median predicted precipitation d2H
d2h_precip_sd Standard deviation of the posterior predictive interval
d2h_precip_lower Lower bound of the credible interval
d2h_precip_upper Upper bound of the credible interval
prediction_interval_width Width of the credible interval
model_used Name of model used for prediction

The interval is the posterior predictive specified in manuscript supplement Section S4.1, Eq. 7 (analytical uncertainty plus the model's posterior residual SD).

Examples

```

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

  # Using data frame input
  data(example_data)
  results <- predict_d2h_precip(example_data, verbose = FALSE)

  # Using individual vectors
  results <- predict_d2h_precip(
    d2h_wax = c(-150, -140, -130),
    longitude = c(-120, -110, -100),
    latitude = c(40, 35, 30),
    elevation = c(1000, 1500, 500),
    verbose = FALSE
  )

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

  # Get full posterior draws
  results <- predict_d2h_precip(
    example_data,
  
```

```
    return_draws = TRUE,  
    verbose = FALSE  
  )  
}
```

```
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 v10 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; shipped v10 posteriors do not contain 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 selected v10 model name

validate_inputs	<i>Validate input data for inversion</i>
-----------------	--

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, 16

assess_claim, 3

assess_claim(), 11, 28

available_models, 4

batch_predict, 5

check_data_cache, 7

clear_download_cache, 8

compare_models, 9

compute_vegetation_envelope, 10

compute_vegetation_envelope(), 3

detect_change, 12

detect_change(), 28

detect_model_capabilities, 13

download_model_data, 14

download_model_data(), 7, 28

estimate_temporal_autocorrelation, 15

example_data, 16

get_all_model_metadata, 17

get_cache_dir, 17

get_cache_dir(), 28

get_cache_info, 18

get_data_path, 19

get_data_url, 19

get_model_info, 20

get_model_parameters, 20

invert_d2H, 24

invert_d2H (invert_d2h), 21

invert_d2h, 21

invert_d2H(), 28

invert_d2H_ensemble, 24

leafwax_config, 25

leafwax_set_config, 25

list_cached_models, 26

list_model_names, 27

list_models, 27

load_posteriors, 28

load_posteriors(), 11

local_effective_slope, 29

predict_d2h_precip, 31

print.leafwax_posterior, 33

select_best_model_from_flags, 33

validate_inputs, 34

validate_model_inputs, 35