

Package ‘HRRI’

September 23, 2026

Type Package

Title Diagnostics for Soil-Plant-Microbial Redox Recovery

Version 1.0.6

Date 2026-09-12

Description Provides diagnostic functions for integrating longitudinal soil, plant and microbial observations during redox disturbance and recovery. Functions calculate stoichiometric potential oxygen demand, accessible electron capacity from explicitly supplied inventories and kinetic parameters, recovery signatures, fixed-reference domain scores, and exploratory multiblock scores with observation-coverage diagnostics. Memory is represented as a holobiont state accumulating from mineralogical, plant-acclimation and microbial-community legacies. An illustrative simulator produces closed Fe and Mn inventories alongside synthetic observations; its carbon, nitrogen, sulfur and oxygen budgets are not closed and its parameters are not calibrated to field rates. Simulation benchmarks assess agreement with a prescribed synthetic target and do not constitute empirical validation or parameter identification. Accuracy assessment is cluster-aware: intervals come from resampling whole trajectories, agreement is reported as Lin's concordance coefficient alongside correlation, and mean squared error is partitioned into bias, variance mismatch and lack of correlation. The measured quantities follow Sander, Hofstetter and Gorski (2015) <[doi:10.1021/acs.est.5b00006](https://doi.org/10.1021/acs.est.5b00006)> for mediated electrochemical determination of electron-accepting and electron-donating capacity, Kluepfel, Piepenbrock, Kappler and Sander (2014) <[doi:10.1038/ngeo2084](https://doi.org/10.1038/ngeo2084)> for regeneration of electron-accepting capacity across repeated anoxic periods, Thompson, Chadwick, Rancourt and Chorover (2006) <[doi:10.1016/j.gca.2005.12.005](https://doi.org/10.1016/j.gca.2005.12.005)> for the increase in iron-oxide crystallinity under redox oscillation, and Keiluweit, Wanzek, Kleber, Nico and Fendorf (2017) <[doi:10.1038/s41467-017-01406-6](https://doi.org/10.1038/s41467-017-01406-6)> for anaerobic microsites in otherwise aerobic soil. Agreement statistics follow Lin (1989) <[doi:10.2307/2532051](https://doi.org/10.2307/2532051)> and Kobayashi and Salam (2000) <[doi:10.2134/agronj2000.922345x](https://doi.org/10.2134/agronj2000.922345x)>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.3.0)

Imports ggplot2 (>= 3.4.0), graphics, grid, igraph (>= 1.5.0), rlang (>= 1.1.0), stats, utils, tidyr, tidyselect

Suggests ggtern (>= 3.4.0), knitr, patchwork (>= 1.2.0), psych, rmarkdown, testthat (>= 3.0.0), viridis

VignetteBuilder knitr

URL <https://github.com/mghotbi/HRRI>, <https://mghotbi.github.io/HRRI/>

BugReports <https://github.com/mghotbi/HRRI/issues>

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Mitra Ghotbi [aut, cre] (ORCID: <https://orcid.org/0000-0001-9185-9993>),
Marjan Ghotbi [ctb] (ORCID: <https://orcid.org/0000-0003-4655-6445>)

Maintainer Mitra Ghotbi <mitra.ghotbi@gmail.com>

Repository CRAN

Date/Publication 2026-09-23 04:30:02 UTC

Contents

attach_hrri_ids	3
benchmark_hrri	4
hrri_infer_architecture	5
HRRI_workflow_example	7
plot_hrri_benchmark	10
plot_rri_accuracy	11
plot_rri_properties	12
plot_rri_recovery_landscape	14
plot_rri_recovery_map	16
plot_rri_simulation_demo	18
plot_rri_state_space	19
plot_RRI_ternary	20
plot_rri_timeseries	22
plot_rri_validation	23
rri_accessible_capacity	24
rri_accuracy	25
rri_capacity_index	28
rri_compensation_index	29
rri_connectivity_score	30
rri_default_reservoirs	31
rri_domain_influence	32
rri_kinetics_score	33
rri_latent_correlation	34

rri_memory_index	35
rri_micro_functional_score	36
rri_o2_demand	37
rri_pipeline	40
rri_pipeline_st	42
rri_property_scores	45
rri_recovery_metrics	46
rri_reference_scores	49
rri_root_physio	50
rri_sensitivity	51
rri_simulation_demo	52
simulate_redox_holobiont	52
theme_ems	56

Index	57
--------------	-----------

attach_hrri_ids	<i>Attach design identifiers to a score table with explicit alignment checks</i>
-----------------	--

Description

Joins experimental design identifiers onto a pipeline score table. Alignment is established by a shared unique row_id, or by a complete shared observation key, or - only as a last resort and with an explicit warning - by preserved input row order. Matching row counts alone do not establish alignment, so the method actually used is recorded in the id_alignment attribute of the returned data frame.

Usage

```
attach_hrri_ids(scores, id, key = NULL)
```

Arguments

scores	Data frame of row-level scores, typically res\$row_scores from rri_pipeline or rri_pipeline_st .
id	Data frame of design identifiers, typically sim\$id from simulate_redox_holobiont .
key	Optional character vector naming the observation key columns to join on. NULL (default) selects a key automatically: row_id when present and unique in both inputs, otherwise the intersection of c("plot", "depth", "plant_id", "time") present in both inputs.

Value

scores with the non-conflicting columns of id attached. The id_alignment attribute is a list giving the method ("row_id", "observation_key" or "row_order"), the key columns used, and the number of rows matched. Shared identifier columns already present in scores are checked for conflicts rather than silently overwritten.

See Also

[rri_pipeline](#), [rri_pipeline_st](#)

Examples

```
scores <- data.frame(row_id = 1:4, RRI = c(0.4, 0.6, 0.5, 0.7))
ids <- data.frame(row_id = 1:4,
                  plot = c("P1", "P1", "P2", "P2"),
                  time = c(1, 2, 1, 2))
out <- attach_hrri_ids(scores, ids)
attr(out, "id_alignment")$method
head(out)
```

benchmark_hrri

Benchmark diagnostic agreement with a simulator-defined target

Description

Runs independent seeds and reports descriptive agreement, not held-out prediction, empirical validation or parameter identification.

Usage

```
benchmark_hrri(
  domains = c("soil", "plant", "micro"),
  missing = 0,
  noise = 0,
  n = 50L,
  seed_start = 1L,
  sim_args = NULL,
  verbose = TRUE,
  pipeline_args = list()
)

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

Arguments

domains	Nonempty subset of soil, plant and micro.
missing	Fraction in $[0, 1)$ of uniformly sampled cells removed (MCAR). This does not implement informative or MNAR missingness.
noise	Nonnegative Gaussian noise SD on each column's original scale. A common SD has different relative effects on differently scaled variables.
n	Positive integer number of independent simulations.

seed_start	First integer seed. The caller's RNG state is restored.
sim_args	Named simulator arguments, excluding seed.
verbose	Print progress messages.
pipeline_args	Named additional rri_pipeline arguments, excluding dat, soil, plant, micro and id; use this to justify orientations and weights.
x	An hrri_benchmark object.
...	Unused method arguments.

Details

Soil and plant synthetic observations and log1p gene abundances feed the pipeline. Latent architecture and microbial activity states are not scoring inputs. The latent target is still a prescribed simulator composite; it is not an independently measured recovery outcome. Replicated rows within seeds are dependent. Compare methods using held-out seeds and independent process outcomes in a separate validation design.

Value

An hrri_benchmark list with summary, seed_metrics, row_data, failures (seed and error), and settings including package/R versions. Summary RMSE and Bias compare the score directly with the chosen target. r_truth and rank_truth are pooled descriptive correlations; spread_association compares within-seed score and target SDs. None is interval coverage or predictive uncertainty. n_rows counts finite matched rows across all seeds.

See Also

rri_pipeline, simulate_redox_holobiont, plot_hrri_benchmark

Examples

```
b <- benchmark_hrri(domains = "soil", n = 2, missing = 0.1)
print(b)
```

hrri_infer_architecture

Fit a conditional capacity-recovery curve (legacy function name)

Description

Fits $y(t) = B * (1 - A * \exp(-r * t))$ after a specified disturbance peak. B is fixed from the observed baseline. A is a fractional recovery deficit and r is a trajectory recovery rate. They are NOT the accessibility alpha and reservoir exchange k in the accessible-capacity model.

Tabulates estimates and independently supplied targets. There is no default mapping from recovery-curve amplitude to accessibility or from recovery rate to reservoir exchange kinetics.

Usage

```

hrri_infer_architecture(
  Eh_stability,
  id = NULL,
  perturb_time = NULL,
  tau_unit = c("day", "hour", "week"),
  group_cols = c("plot", "depth"),
  fit_edc = FALSE,
  min_points = 3L,
  verbose = TRUE,
  control = list(),
  baseline_end = NULL
)

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

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

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

validate_architecture(arch, params = NULL)

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

```

Arguments

<code>Eh_stability</code>	Data frame containing EAC and optionally EDC.
<code>id</code>	Aligned identifiers containing time and grouping columns if absent.
<code>perturb_time</code>	Disturbance peak time; NULL detects the EAC minimum. Detection is descriptive and can select noise or the last observation.
<code>tau_unit</code>	Time unit: day, hour or week.
<code>group_cols</code>	Columns defining one trajectory. Include <code>plant_id</code> when plant-level trajectories are separate; duplicate times are rejected.
<code>fit_edc</code>	Also fit increasing EDC recovery toward its own baseline. This is inappropriate for a decreasing EDC trajectory; default FALSE.
<code>min_points</code>	Minimum distinct post-peak times (at least three). This is a computational threshold, not proof of identifiability.
<code>verbose</code>	Print fit status.
<code>control</code>	Named list passed to <code>stats::nls.control</code> .
<code>baseline_end</code>	Explicit end of the baseline window, strictly before the peak. NULL uses only the first observation time as a stated assumption.

x	An hrri_arch object.
...	Additional method arguments.
object	An hrri_arch object.
row.names, optional	Standard data-frame method arguments.
arch	An hrri_arch object with a keyed ground_truth data frame.
params	Explicit strings of the form estimated_column:true_column.

Details

A nonlinear fit converging does not establish structural or practical identifiability. Baseline uncertainty is not propagated into coefficient SEs. With $Cacc(t) = Q * \alpha * (1 - \exp(-k*t))$, Q and alpha cannot be separated using that curve alone. This function does not solve that inverse problem. Fit only a single recovery window. Nonmonotonic and multi-event trajectories require a different model and residual diagnostics.

Value

An hrri_arch object containing estimates, conditional fit status, fit objects and optional summaries of supplied latent columns. Legacy columns Q_eac, alpha_eac, k_eac and M_eac remain for compatibility. Prefer the aliases baseline_eac, deficit_fraction_eac, recovery_rate_eac and terminal_ratio_eac. M_eac is a terminal-to-baseline ratio, not a measure of causal memory. k_eac_h converts the trajectory rate to reciprocal hours, not exchange kinetics.

Data frame with finite_pair flags and descriptive aggregate statistics. The caller must establish that paired columns have the same definition and units.

See Also

rri_recovery_metrics, rri_accessible_capacity

Examples

```
tt <- 0:12
yy <- ifelse(tt <= 2, 10, 10 * (1 - 0.7 * exp(-0.3 * (tt - 3))))
dat <- data.frame(plot = "P1", depth = "D1", time = tt, EAC = yy)
a <- hrri_infer_architecture(dat, perturb_time = 3, baseline_end = 2,
                           verbose = FALSE)

a$estimates
```

HRRI_workflow_example *Worked example: a complete HRRI workflow*

Description

A documentation page, not a dataset: there is no object named HRRI_workflow_example to load, and data() will not find one. It demonstrates a complete HRRI workflow on a synthetic plant-soil-microbiome time-series dataset generated by [simulate_redox_holobiont\(\)](#).

Details

This example shows how to:

- Simulate a spatio-temporal holobiont redox dataset.
- Compute sample-level Redox Resilience Index scores.
- Inspect compositional Physio-Soil-Micro RRI allocation.
- Quantify perturbation-recovery metrics.
- Generate ternary and recovery-metric visualizations.

The simulated dataset contains plant physiological variables, rhizosphere oxygen-flux proxies, soil redox chemistry, hydrological variables, dissolved organic carbon, microbial abundance features, microbial redox trait proxies, and optional functional gene abundance or MetaT-style expression features.

The generated data are fully synthetic and are provided for examples, testing, benchmarking, teaching, and method development. They are not calibrated to any specific ecosystem. Users may replace these simulated inputs with their own external datasets, provided that rows are aligned across id, ROS_flux, Eh_stability, and micro_data.

See Also

`simulate_redox_holobiont()`, `rri_pipeline_st()`, `rri_recovery_metrics()`, `plot_RRI_ternary()`, `plot_rri_recovery_landscape()`

Examples

```
# Simulate a compact synthetic holobiont redox time series
sim <- simulate_redox_holobiont(
  n_plot = 2,
  n_depth = 1,
  n_plant = 2,
  n_time = 12,
  p_micro = 20,
  seed = 1
)

# Inspect the returned data layers
names(sim)
head(sim$id)
head(sim$ROS_flux)
head(sim$Eh_stability)
head(sim$micro_data)
head(sim$micro_traits)

# Combine microbial abundance, trait, and gene-level features
micro_features <- cbind(
  sim$micro_data,
  sim$micro_traits,
  log1p(sim$micro_gene_abundance)
)
```

```

# Compute HRRI scores
res <- suppressWarnings(rri_pipeline_st(
  ROS_flux = sim$ROS_flux,
  Eh_stability = sim$Eh_stability,
  micro_data = micro_features,
  id = sim$id,
  reducer = "per_domain",
  scaling = "pnorm"
))

# Sample-level RRI scores
head(res$row_scores)

# Compositional Physio-Soil-Micro allocation used for ternary plots
head(res$row_scores_comp)

# Ternary visualization requires res$row_scores_comp, not recovery metrics
# ggtern is not compatible with every ggplot2 release, and merely loading
# it breaks later ggplot2 output. plot_RRI_ternary() checks the ggplot2
# version before touching ggtern and errors cleanly if it cannot be used.
if (requireNamespace("viridis", quietly = TRUE)) {
  try(
    print(plot_RRI_ternary(
      res$row_scores_comp,
      point_size = 3,
      show_centroid = TRUE
    )),
    silent = TRUE
  )
}

# Quantify perturbation-recovery metrics from the RRI trajectory
rec <- rri_recovery_metrics(
  res = res,
  id = sim$id,
  time_col = "time",
  group_cols = c("plot", "depth", "plant_id"),
  perturb_start = 5,
  perturb_end = 7
)

head(rec)

# Recovery visualizations.
# plot_rri_recovery_map() takes the scored pipeline output plus identifiers;
# plot_rri_recovery_landscape() takes the metrics table.
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot_rri_recovery_map(
    res = res,
    id = sim$id,
    rec = rec,
    time_col = "time",

```

```

    group_cols = c("plot", "depth", "plant_id"),
    perturb_start = 5,
    perturb_end = 7
  )
}

if (requireNamespace("ggplot2", quietly = TRUE) &&
    requireNamespace("tidyr", quietly = TRUE) &&
    requireNamespace("tidyselect", quietly = TRUE)) {
  plot_rri_recovery_landscape(
    rec,
    metrics = c("depth_min_frac", "overshoot_frac", "I_norm",
                "k", "tau_lag", "t_half")
  )
}

```

plot_hrri_benchmark	<i>Plot descriptive benchmark agreement</i>
---------------------	---

Description

Plot descriptive benchmark agreement

Usage

```
plot_hrri_benchmark(bm, print = TRUE, colour = "#0072B2")
```

Arguments

bm	An hrri_benchmark object.
print	Whether to display the plot.
colour	Scatter colour.

Value

Invisibly, a two-panel patchwork object. Requires patchwork.

Examples

```

b <- benchmark_hrri(domains = "soil", n = 2, missing = 0.1)
plot_hrri_benchmark(b)

```

plot_rri_accuracy	<i>Four-panel diagnostic figure for an accuracy assessment</i>
-------------------	--

Description

Draws the figure that accompanies `rri_accuracy()`. Each panel answers a question a single correlation coefficient cannot: whether the score is calibrated, whether disagreement grows with level, how much precision the clustering costs, and which kind of error dominates.

Usage

```
plot_rri_accuracy(
  acc,
  panels = c("calibration", "agreement", "precision", "error"),
  score_label = "Score",
  target_label = "Reference target",
  point_alpha = 0.45,
  show_clusters = NULL,
  base_size = 11,
  ncol = 2
)
```

Arguments

<code>acc</code>	An object of class <code>rri_accuracy</code> from <code>rri_accuracy()</code> , created with <code>n_boot > 0</code> so that the resampled statistics are available.
<code>panels</code>	Character vector selecting panels, any of "calibration", "agreement", "precision" and "error". Defaults to all four.
<code>score_label</code> , <code>target_label</code>	Axis labels for the score and the reference target.
<code>point_alpha</code>	Opacity of the individual observations. Lower it when trajectories overplot.
<code>show_clusters</code>	Logical, or <code>NULL</code> to decide automatically. Colours observations by cluster. Set <code>FALSE</code> above roughly 20 clusters, where the colouring stops being informative.
<code>base_size</code>	Base font size passed to <code>theme_ems()</code> .
<code>ncol</code>	Number of columns in the assembled figure. Ignored when patchwork is unavailable.

Details

Panel A, calibration. Score against target, with the 1:1 line dashed and the fitted line solid. Perfect agreement puts the points on the dashed line; a solid line flatter than it means the score compresses the target's range, and one displaced from it means a systematic bias. Open points are cluster means, the level at which these units are independent.

Panel B, agreement. A Bland-Altman plot: the difference between score and target against their mean, with the mean difference and the limits of agreement. A scatter that fans out, or that slopes,

shows that disagreement depends on level, which a correlation coefficient cannot reveal. Because observations are clustered, the limits come from cluster means; row-level limits would be far too tight.

Panel C, precision. The bootstrap sampling distribution of r under row resampling and under trajectory resampling, with both intervals drawn beneath. The difference in width is the cost of treating repeated observations of one unit as independent observations of many. The permutation null, when computed, sits behind them for reference.

Panel D, error. Mean squared error split into squared bias, variance mismatch and lack of correlation. The three sum to the mean squared error exactly, so the panel is a partition rather than an approximation.

Colours follow the package's chemistry-derived palette: teal for redox, rust for iron, violet for manganese, ochre for cautionary annotation.

Value

If **patchwork** is installed, a single assembled patchwork object. Otherwise a named list of ggplot objects, so nothing is lost when the suggested package is absent.

See Also

[rri_accuracy\(\)](#) for the statistics the figure displays.

Examples

```
set.seed(1)
k <- 8; m <- 20
unit <- rnorm(k, 0, 0.30)
target <- unlist(lapply(unit, function(u) u + 0.5 + rnorm(m, 0, 0.05)))
score <- 0.75 * target + 0.10 + rnorm(k * m, 0, 0.06)
traj <- rep(seq_len(k), each = m)

acc <- rri_accuracy(score, target, cluster = traj,
                    n_boot = 200, n_perm = 200, seed = 1)
p <- plot_rri_accuracy(acc)

print(p)
```

plot_rri_properties *Radar Chart of Available HRRI Diagnostics*

Description

Displays available diagnostic summaries labelled Capacity, Connectivity, Kinetics and Memory. The composite RRI is not an axis: it is built from the plant, soil and microbial domains rather than from these four properties, so averaging across it is not defensible. These axes are operational descriptors returned by [rri_property_scores](#); they are not direct measurements or identified estimates of the theoretical mechanisms bearing the same names.

Usage

```
plot_rri_properties(
  props,
  rri_value = NULL,
  group_list = NULL,
  fill_alpha = 0.2,
  colours = c("#1A3A5C", "#E07B39", "#2E7D32", "#7B3294", "#B2182B"),
  show_values = TRUE,
  title = "HRRRI Diagnostic Profile",
  base_size = 13
)
```

Arguments

props	A list returned by rri_property_scores , or a named numeric vector with elements Capacity, Connectivity, Kinetics, Memory (values in $[0, 1]$).
rri_value	Optional numeric. Composite RRI, reported in the subtitle for reference. It is not plotted as an axis and does not enter the centre value, which is the mean of the resolved property axes. Defaults to props\$rri_summary if available.
group_list	Optional named list of property score vectors, one per group (e.g., per thaw stage or treatment). If supplied, multiple overlapping polygons are drawn, one per group.
fill_alpha	Numeric in $[0, 1]$. Polygon fill transparency.
colours	Character vector of polygon outline/fill colours, recycled across groups.
show_values	Logical. Annotate each axis tip with the numeric score.
title	Character. Plot title.
base_size	Numeric. Base font size.

Details

The chart uses Cartesian coordinates constructed with ggplot2; no external radar-chart package is required. Each available axis runs from 0 (centre) to 1 (rim). Polygon area has no quantitative meaning, and axes based on different transformations are not necessarily commensurable.

Axis meanings:

- **Capacity** — oxidative-oriented soil feature composite; not accessible capacity unless the caller calculates and supplies it explicitly.
- **Connectivity** — association or network-topology descriptor; not demonstrated electron transfer.
- **Kinetics** — cohort- or timescale-relative recovery-speed descriptor; not a mineral exchange rate.
- **Memory** — loop-area and persistent-displacement descriptor; not an identified causal memory state.

Value

A ggplot object.

Examples

```

sim <- simulate_redox_holobiont(
  n_plot = 3, n_depth = 2, n_plant = 3, n_time = 14,
  p_micro = 30, seed = 99
)

res <- rri_pipeline_st(
  ROS_flux      = sim$ROS_flux,
  Eh_stability  = sim$Eh_stability,
  micro_data    = sim$micro_data,
  id            = sim$id,
  reducer       = "per_domain",
  scaling       = "pnorm"
)

rec <- rri_recovery_metrics(
  res          = res,
  id           = sim$id,
  time_col     = "time",
  group_cols   = c("plot", "depth", "plant_id"),
  perturb_start = 5,
  perturb_end  = 8
)

props <- rri_property_scores(
  res      = res,
  rec      = rec,
  soil_df  = sim$Eh_stability,
  eac_col  = "EAC",
  edc_col  = "EDC",
  humic_col = "dissolved_organic_matter_redox"
)

plot_rri_properties(props)

```

plot_rri_recovery_landscape

Plot a recovery landscape from RRI perturbation-recovery metrics

Description

Visualises trajectory-level recovery metrics from `rri_recovery_metrics()`. Each row is one trajectory and each column one recovery signature. Cell colour encodes the within-column scaled magnitude; the printed number is always the unscaled value, so nothing is hidden by the scaling.

Usage

```
plot_rri_recovery_landscape(
```

```

    rec,
    group_cols = c("plot", "depth", "plant_id"),
    metrics = c("depth_min_frac", "overshoot_frac", "I_norm", "k", "tau_lag", "t_half"),
    order_by = "I_norm",
    orient = c("concern", "raw"),
    drop_empty = TRUE,
    base_size = 12
  )

```

Arguments

<code>rec</code>	A data frame returned by <code>rri_recovery_metrics()</code> .
<code>group_cols</code>	Character vector of columns identifying one trajectory.
<code>metrics</code>	Character vector of recovery metric columns to plot. Defaults to the columns returned by <code>rri_recovery_metrics()</code> . Legacy names (<code>A_norm</code> , <code>O_norm</code> , <code>tau_r</code>) are still labelled if supplied.
<code>order_by</code>	Character scalar. Metric used to order trajectories.
<code>orient</code>	Controls what darker colour means. "concern" (default) inverts metrics for which a <i>smaller</i> value is the more concerning outcome, so a dark cell always reads as "more concerning" across the whole panel. "raw" scales every column upward, meaning dark is high-valued regardless of interpretation. See Details.
<code>drop_empty</code>	Logical. Drop metric columns that are NA for every trajectory rather than drawing a blank column. <code>k</code> and <code>t_half</code> are NA unless a recovery rate was fitted, so an all-NA column is common and means "not estimable here", not "zero".
<code>base_size</code>	Numeric. Base font size.

Details

Why orientation matters. The recovery signatures do not share a polarity. A large `depth_min_frac` (deep decline), a large `tau_lag` (slow onset) and a large `t_half` (slow return) are all unfavourable, but a large `k` is a *fast* recovery rate and therefore favourable. Scaling every column upward and applying one colour ramp would make dark mean "bad" in some columns and "good" in others. With `orient = "concern"` the `k` column is inverted before scaling so the ramp is interpretable across the panel. `overshoot_frac` is treated as neutral and never inverted, because overshoot is not unambiguously favourable or unfavourable.

Scaling is min-max **within each column, within this cohort**. A dark cell means "high relative to the other trajectories in this run", not high in any absolute sense. Two datasets cannot be compared cell by cell.

If `rec` has no `trajectory_class` column, one is derived from `displaced_plateau_flag` and `incomplete_return_frac`. The derived labels describe the score trajectory only and identify no mechanism.

Value

A ggplot object.

See Also

`rri_recovery_metrics()`, `plot_rri_recovery_map()`

Examples

```
sim <- simulate_redox_holobiont(
  n_plot = 2,
  n_depth = 3,
  n_plant = 2,
  n_time = 12,
  p_micro = 20,
  seed = 109
)

res <- suppressWarnings(rri_pipeline_st(
  ROS_flux = sim$ROS_flux,
  Eh_stability = sim$Eh_stability,
  micro_data = sim$micro_data,
  id = sim$id,
  reducer = "per_domain",
  scaling = "pnorm"
))

rec <- rri_recovery_metrics(
  res = res,
  id = sim$id,
  time_col = "time",
  group_cols = c("plot", "depth", "plant_id"),
  perturb_start = 5,
  perturb_end = 7
)

plot_rri_recovery_landscape(
  rec,
  metrics = c("depth_min_frac", "overshoot_frac", "I_norm",
    "k", "tau_lag", "t_half")
)
```

`plot_rri_recovery_map` *Plot RRI Recovery Map*

Description

Visualises per-group RRI trajectories through baseline, perturbation and recovery phases as a tile-and-line map. Each row is one trajectory group; time proceeds along the x-axis; tile fill encodes RRI magnitude; vertical bands mark the perturbation window; and trajectory class is annotated on the right margin.

landscape shows cross-metric comparison per trajectory, while the recovery map shows temporal RRI dynamics per group.

Usage

```
plot_rri_recovery_map(
  res,
  id,
  rec = NULL,
  time_col = "time",
  group_cols = c("plot", "depth", "plant_id"),
  perturb_start = NULL,
  perturb_end = NULL,
  palette = "plasma",
  base_size = 11,
  max_groups = 40L
)
```

Arguments

<code>res</code>	An object returned by rri_pipeline_st .
<code>id</code>	A data frame of experimental identifiers (same rows as <code>res\$row_scores</code>), containing at minimum <code>time_col</code> and the columns in <code>group_cols</code> .
<code>rec</code>	Optional data frame from rri_recovery_metrics . If supplied, trajectory class annotations are added to the right margin.
<code>time_col</code>	Character. Name of the time column in <code>id</code> .
<code>group_cols</code>	Character vector. Columns in <code>id</code> defining trajectory groups (e.g., <code>c("plot", "depth", "plant_id")</code>).
<code>perturb_start</code>	Numeric. Start of perturbation phase (same units as <code>time_col</code>).
<code>perturb_end</code>	Numeric. End of perturbation phase.
<code>palette</code>	Character. Viridis palette option for RRI fill.
<code>base_size</code>	Numeric. Base font size.
<code>max_groups</code>	Integer. Maximum number of trajectory groups to display. Groups are sampled if the total exceeds this value.

Value

A ggplot object.

Examples

```
sim <- simulate_redox_holobiont(
  n_plot = 2, n_depth = 2, n_plant = 3, n_time = 14,
  p_micro = 20, seed = 101
)

res <- rri_pipeline_st(
  ROS_flux = sim$ROS_flux,
  Eh_stability = sim$Eh_stability,
  micro_data = sim$micro_data,
  id = sim$id,
```

```

    reducer      = "per_domain",
    scaling      = "pnorm"
  )

  rec <- rri_recovery_metrics(
    res          = res,
    id           = sim$id,
    time_col     = "time",
    group_cols   = c("plot", "depth", "plant_id"),
    perturb_start = 5,
    perturb_end  = 8
  )

  plot_rri_recovery_map(
    res          = res,
    id           = sim$id,
    rec          = rec,
    time_col     = "time",
    group_cols   = c("plot", "depth", "plant_id"),
    perturb_start = 5,
    perturb_end  = 8
  )

```

plot_rri_simulation_demo

Draw the reproducible simulation demonstration

Description

Draw the reproducible simulation demonstration

Usage

```

plot_rri_simulation_demo(
  demo,
  figure = c("observations", "coverage", "capacity")
)

```

Arguments

demo	Result of rri_simulation_demo.
figure	observations, coverage or capacity.

Value

Invisibly returns the plotted data table; draws on the active device. Uses base graphics, so no optional plotting package is required.

Examples

```
demo <- rri_simulation_demo(seed = 20260830L)
plot_rri_simulation_demo(demo, figure = "observations")
```

plot_rri_state_space *Plot domain-score space with correctly matched trajectory diagnostics*

Description

Plot domain-score space with correctly matched trajectory diagnostics

Usage

```
plot_rri_state_space(
  res,
  rec = NULL,
  x_property = c("Physio", "Connectivity", "Soil", "Micro"),
  y_property = c("Soil", "Micro", "Physio", "Kinetics"),
  colour_by = c("RRI", "Memory", "trajectory_class"),
  group_cols = c("plot", "depth", "plant_id"),
  base_size = 12
)
```

Arguments

res	RRI result with aligned identifiers in row_scores.
rec	Optional one-row-per-trajectory diagnostic table.
x_property, y_property	Domain scores, or group association/kinetics.
colour_by	RRI, Memory, or an explicitly supplied trajectory_class.
group_cols	Full key shared between row_scores and rec.
base_size	Plot font size.

Value

ggplot. Domain scores are not relabelled as mechanistic properties.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline(soil = sim$Eh_stability, plant = sim$ROS_flux,
  id = sim$id)
plot_rri_state_space(res, group_cols = c("plot", "depth", "plant_id"))
```

plot_RRI_ternary

Ternary Plot of Relative Domain Scores

Description

Creates a ternary diagram of the relative magnitudes of plant, soil and microbial domain scores after closure to a unit sum. These coordinates are display quantities, not fractions of causal buffering capacity. Points are filled according to the corresponding composite RRI value.

Usage

```
plot_RRI_ternary(
  ternary_df,
  point_size = 5,
  point_alpha = 0.9,
  palette = "plasma",
  show_subtitle = TRUE,
  show_centroid = TRUE,
  centroid_shape = 23,
  centroid_size = 1.4,
  tolerance = 1e-06,
  renormalize = FALSE,
  centroid_method = c("auto", "simplex_mean", "aitchison_mean")
)
```

Arguments

ternary_df	A data frame containing compositional columns Physio, Soil, Micro, and RRI.
point_size	Numeric; size of ternary points.
point_alpha	Numeric between 0 and 1 controlling point transparency.
palette	Character; viridis palette option.
show_subtitle	Logical; display system-level RRI mean in subtitle.
show_centroid	Logical; add compositional centroid marker.
centroid_shape	Numeric; ggplot2 shape for centroid marker.
centroid_size	Numeric multiplier for centroid size.
tolerance	Numeric; tolerance used for compositional closure checks.
renormalize	Logical; if TRUE, renormalises rows that do not sum to 1.
centroid_method	Character; one of "auto", "simplex_mean", or "aitchison_mean".

Details

Closure removes absolute score magnitude: rows with proportional domain scores occupy the same position even when their composite scores differ. Do not infer mechanistic allocation, causal contribution or substitution from this plot. If clr-transformed coordinates are attached as an attribute ("clr"), the centroid can be computed using the Aitchison mean. Otherwise, a simplex arithmetic mean is used.

Value

A ggtern object.

Examples

```
## ggtern cannot be used with ggplot2 >= 4.0.0, and loading it there breaks
## later ggplot output, so the example skips rather than errors.
if (utils::packageVersion("ggplot2") < "4.0.0" &&
    requireNamespace("ggtern", quietly = TRUE) &&
    requireNamespace("viridis", quietly = TRUE)) {
  sim <- simulate_redox_holobiont(
    n_plot = 2,
    n_depth = 2,
    n_plant = 2,
    n_time = 8,
    p_micro = 6,
    seed = 1234
  )

  # ---- Compute HRRI ----
  res <- rri_pipeline_st(
    ROS_flux = sim$ROS_flux,
    Eh_stability = sim$Eh_stability,
    micro_data = sim$micro_data,
    id = sim$id,
    reducer = "per_domain",
    scaling = "pnorm"
  )

  # ---- Extract compositional scores ----
  ternary_df <- res$row_scores_comp

  # ---- Plot ternary allocation ----
  p <- plot_RRI_ternary(
    ternary_df,
    point_size = 3,
    show_centroid = TRUE
  )
}
```

plot_rri_timeseries *Aligned time series with separate physical units*

Description

Shows Eh, EAC and the observation-derived score in separate panels sharing time. It does not place unlike units on a common axis.

Usage

```
plot_rri_timeseries(
  sim,
  res,
  plot_id = "P1",
  depth_id = "D1",
  plant_id = "Plant1",
  perturb_start = NULL,
  perturb_end = NULL,
  base_size = 9
)
```

Arguments

sim	Simulator output containing id and soil_data.
res	Pipeline output containing row_scores.
plot_id, depth_id, plant_id	Identifiers for a single trajectory.
perturb_start, perturb_end	Optional disturbance interval in input time units.
base_size	Base font size.

Value

A ggplot with three vertically aligned panels.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline(soil = sim$Eh_stability, plant = sim$ROS_flux,
  id = sim$id)
plot_rri_timeseries(sim, res)
```

plot_rri_validation	<i>Scatter of HRRI score against a simulator-defined target</i>
---------------------	---

Description

Plots mean RRI versus a declared synthetic target per aggregate row, annotated with descriptive Pearson r and direct score-target RMSE, not LOO error. No model is trained or held out by this plotting function.

Usage

```
plot_rri_validation(  
  pool_agg,  
  rri_col = "RRI",  
  truth_col = "truth",  
  colour_col = NULL,  
  label_col = NULL,  
  base_size = 9  
)
```

Arguments

pool_agg	Seed-level aggregate data frame with columns RRI and truth (one row per seed).
rri_col	Name of the HRRI column in pool_agg (default "RRI").
truth_col	Name of the latent truth column (default "truth").
colour_col	Optional column for point colour (e.g., "n_cycles").
label_col	Optional column for point labels.
base_size	Base font size.

Value

A ggplot object.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)  
res <- rri_pipeline(soil = sim$Eh_stability, plant = sim$ROS_flux)  
agg <- data.frame(RRI = res$row_scores$RRI, truth = sim$latent_truth)  
plot_rri_validation(agg)
```

rri_accessible_capacity

Event-window accessible reservoir capacities

Description

Computes $Q * \alpha * (1 - \exp(-k * \tau))$ for declared reservoirs. Q must be nonoverlapping electron-equivalent inventories with explicit reaction endpoints. EAC and EDC are returned separately. Their sum is an inventory descriptor, not electron flux; their difference is not an oxygen budget.

Usage

```
rri_accessible_capacity(
  soil_df,
  reservoirs,
  tau = 24,
  normalise = TRUE,
  return_components = FALSE
)
```

Arguments

soil_df	Numeric reservoir measurements.
reservoirs	Nonempty named list of Q_{col} , α , k and type (EAC/EDC). α in $[0, 1]$ and $k \geq 0$ may be scalars, row vectors or column names. Parameters must be specified for the relevant process and conditions; they are not identifiable separately from one accessible-capacity observation.
tau	Non-negative duration, scalar or row vector; units reciprocal to k .
normalise	Divide by the sum of observed inventories. This produces an accessible fraction, not absolute capacity or guaranteed comparability.
return_components	Include reservoir contribution summaries.

Details

Default reservoir parameters are illustrative scenario values only. Fe(II) oxidation rates must not be assigned as Fe(III) reduction constants. For capacity estimation use experimentally constrained process-specific rates.

Value

Capacities, observed subtotal, fraction and reservoir coverage. Missing reservoir types remain NA. Partial rows are labelled observed subtotals; absence is not zero. `ck_limited` is retained as NA because 0.30 is not a validated threshold. Negative inventories are treated as missing.

Examples

```
df <- data.frame(EAC = c(10, 20, 30), EDC = c(5, 8, 12))
res <- rri_accessible_capacity(
  df,
  reservoirs = list(
    bulk_EAC = list(Q_col = "EAC", alpha = 0.5, k = 0.2, type = "EAC"),
    bulk_EDC = list(Q_col = "EDC", alpha = 0.45, k = 0.15, type = "EDC")
  ),
  tau = 24
)
res$cacc
```

rri_accuracy	<i>Agreement between a score and a reference target, respecting clustering</i>
--------------	--

Description

Quantifies how closely a score tracks a reference target, using statistics appropriate to repeated observations of the same experimental units. A pooled row-wise correlation over a longitudinal panel overstates precision, because rows within a trajectory are not independent. This function reports the naive row-level result alongside the cluster-aware one, so the difference is visible rather than hidden.

Usage

```
rri_accuracy(
  score,
  target,
  cluster = NULL,
  score_col = "RRI",
  n_boot = 1000,
  n_perm = 1000,
  conf = 0.95,
  seed = NULL
)

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

Arguments

score	Numeric vector of scores, or an RRI object. If an RRI object, score_col is taken from its row_scores.
target	Numeric vector of reference values, the same length as score.

cluster	Vector identifying the independent experimental unit for each observation, typically one trajectory. Rows sharing a value are treated as dependent. If NULL, every row is treated as independent and the function warns, because that assumption is rarely correct for time series.
score_col	Column name used when score is an RRI object.
n_boot	Number of cluster bootstrap resamples for confidence intervals. Set to 0 to skip.
n_perm	Number of cluster permutations for the null test. Set to 0 to skip.
conf	Confidence level for intervals.
seed	Optional integer seed for reproducible resampling.
x	An rri_accuracy object.
...	Ignored.

Details

Why correlation is not agreement. Pearson's r is invariant to location and scale: a score equal to $2\times$ the target plus a constant correlates perfectly with it while agreeing with it nowhere. Lin's concordance correlation coefficient

$$\rho_c = \frac{2s_{xy}}{s_x^2 + s_y^2 + (\bar{x} - \bar{y})^2}$$

penalises departure from the 1:1 line and is reported alongside r . A large gap between the two means the score is well correlated but miscalibrated.

Why clustering matters. With m observations per unit and intra-cluster correlation ρ , the design effect is $1 + (m-1)\rho$ and the effective sample size is n/deff . For a 40-point trajectory with $\rho = 0.3$ this is roughly a twelvefold reduction, so a confidence interval computed from the row count is far too narrow. Intervals here come from a cluster bootstrap that resamples whole trajectories with replacement, which preserves the within-unit dependence.

Error decomposition. Mean squared error is split following Kobayashi and Salam (2000) into squared bias, a difference in variability, and a lack of correlation:

$$\text{MSE} = (\bar{x} - \bar{y})^2 + (s_x - s_y)^2 + 2s_x s_y (1 - r)$$

These say different things. Large squared bias means a systematic offset, correctable by recentring. Large variance mismatch means the score is too flat or too volatile. Large lack of correlation means the score does not track the target's pattern, and no rescaling will fix it.

What the permutation null asks. Whole trajectories are exchanged, so each unit keeps its own temporal shape and only the pairing between score and target is broken. This is deliberately the harder null. Permuting individual rows would destroy the shared event-driven shape that every trajectory has, making almost any score look significant; exchanging blocks retains that shape and asks whether the score tracks *this* unit's target beyond what the common disturbance imposes on all of them. A small p-value under this null is therefore informative, and a large one is not evidence that the score is uninformative about the disturbance itself.

What this does not establish. If the score and the target come from the same generative model, as they do for latent_truth from `simulate_redox_holobiont()`, high agreement is internal consistency and nothing more. It is not predictive accuracy, not out-of-sample error, and not evidence

of ecological validity. Those require a target constructed independently of the score, and replication at the level of independent experimental units.

Value

An object of class `rri_accuracy`: a list with elements `agreement`, `calibration`, `decomposition`, `dependence`, `null_test` and `notes`, plus `data` (the complete cases actually used), `draws` (the re-sampled and permuted statistics) and `conf`. The last three exist so that `plot_rri_accuracy()` can draw the sampling distributions without repeating the resampling. See Details.

Methods (by generic)

- `print(rri_accuracy)`: Compact console summary of an accuracy assessment.

References

Lin, L.I. (1989) A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, **45**, 255–268.

Kobayashi, K. & Salam, M.U. (2000) Comparing simulated and measured values using mean squared deviation and its components. *Agronomy Journal*, **92**, 345–352.

See Also

`plot_rri_accuracy()` for the four-panel diagnostic figure; `benchmark_hrri()` for repeated-seed benchmarking; `rri_domain_influence()` for which domain drives the score.

Examples

```
## Synthetic panel: 8 trajectories, 20 time points each.
## The score is deliberately miscalibrated so the r-vs-CCC gap is visible.
set.seed(1)
k <- 8; m <- 20
unit <- rnorm(k, 0, 0.30)
target <- unlist(lapply(unit, function(u) u + 0.5 + rnorm(m, 0, 0.05)))
score <- 0.75 * target + 0.10 + rnorm(k * m, 0, 0.06)
traj <- rep(seq_len(k), each = m)

acc <- rri_accuracy(score, target, cluster = traj,
                    n_boot = 200, n_perm = 200, seed = 1)
acc

## Correlation is high, concordance is not: the score compresses the target.
acc$agreement
acc$calibration

## The row count is not the sample size.
acc$dependence

## Against the simulator's own prescribed target.
sim <- simulate_redox_holobiont(
```

```

n_plot = 2, n_depth = 2, n_plant = 3, n_time = 30,
p_micro = 20, seed = 2026
)
res <- suppressWarnings(rri_pipeline_st(
  ROS_flux = sim$ROS_flux, Eh_stability = sim$Eh_stability,
  micro_data = sim$micro_data, id = sim$id
))
scored <- attach_hrri_ids(res$row_scores, sim$id)
tj <- interaction(scored$plot, scored$depth, scored$plant_id, drop = TRUE)
rri_accuracy(scored$RRI, sim$latent_truth, cluster = tj,
  n_boot = 500, n_perm = 500, seed = 1)

```

rri_capacity_index	<i>Oxidative-oriented soil feature composite</i>
--------------------	--

Description

Standardizes selected measured features and averages them with declared weights. EAC is positive and EDC is inverted after scaling. This is an oxidative-oriented descriptor, not accessible capacity, a redox potential, or a universal ranking of resilience. High EDC means greater reducing capacity.

Usage

```

rri_capacity_index(
  soil_df,
  eac_col = "EAC",
  edc_col = "EDC",
  reactive_fe_col = NULL,
  poorly_cryst_fe_col = NULL,
  humic_col = NULL,
  w_eac = 0.35,
  w_edc = 0.25,
  w_fe = 0.25,
  w_humic = 0.15,
  scaling = c("pnorm", "minmax", "reference"),
  ref_ranges = NULL
)

```

Arguments

soil_df	Numeric soil measurements.
eac_col, edc_col	Columns of electron-accepting/donating capacity.
reactive_fe_col, poorly_cryst_fe_col	Optional Fe-pool proxy columns.
humic_col	Optional organic redox proxy column.

w_eac, w_edc, w_fe, w_humic
Non-negative weights.

scaling pnorm, minmax, or reference; pnorm is not a calibrated probability.

ref_ranges Named increasing finite ranges in original measurement units; mandatory for every selected column when scaling=reference.

Value

capacity_score, EAC/(EAC+EDC) ratio, contributors and observed coverage. Coverage concerns selected available columns; overlapping pools must not be mistaken for independent evidence. Common assays and reference ranges are necessary but insufficient for cross-study comparability.

Examples

```
df <- data.frame(EAC = c(10, 20, 30), EDC = c(5, 8, 12))
rri_capacity_index(df)$capacity_score
```

rri_compensation_index

Cross-domain asynchrony diagnostic

Description

Association or variance cancellation, not evidence of causal buffering. Variance_ratio is $1 - \text{var}(\text{rowSums}(X))/\text{sum}(\text{var}(X_j))$; positive values indicate cancellation. mean_neg_cor is the negative mean correlation.

Usage

```
rri_compensation_index(
  res,
  per_group = FALSE,
  group_cols = NULL,
  id = NULL,
  method = c("mean_neg_cor", "variance_ratio"),
  scale_output = TRUE
)
```

Arguments

res RRI result.

per_group Compute by trajectory/group.

group_cols Group identifiers.

id Optional aligned identifiers.

method mean_neg_cor or variance_ratio.

scale_output For correlations maps $[-1, 1]$ to $[0, 1]$; for variance cancellation truncates negative values to zero. A correlation score of 0.5 means zero mean correlation, not moderate biological compensation.

Value

Diagnostic score, correlations and interpretation.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline(soil = sim$Eh_stability, plant = sim$ROS_flux)
rri_compensation_index(res)
```

rri_connectivity_score

Cross-domain association or graph-topology summary

Description

Retains the legacy function name. Correlation magnitude is association, not electron-transfer encounter probability or measured alpha. Network summaries concern unweighted topology, not biochemical connectivity.

Usage

```
rri_connectivity_score(
  res,
  method = c("cross_domain_magnitude", "network"),
  per_group = FALSE,
  group_cols = NULL
)
```

Arguments

res	RRI result; graph method uses meta\$graph.
method	cross_domain_magnitude or network.
per_group	Compute association by group.
group_cols	Required grouping columns when per_group=TRUE.

Value

Score, association coefficients and method/provenance information.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline(soil = sim$Eh_stability, plant = sim$ROS_flux)
rri_connectivity_score(res)
```

`rri_default_reservoirs`*Illustrative reservoir parameter template*

Description

Returns example values, not calibrated mineral-specific constants. The default uses the core bulk EAC and EDC columns. Supply phase-resolved column names only when those inventories are nonoverlapping and expressed in electron-equivalent units.

Usage

```
rri_default_reservoirs(  
  eac_ferrihydrite_col = NULL,  
  eac_goethite_col = NULL,  
  eac_structural_col = NULL,  
  edc_humic_fast_col = NULL,  
  edc_humic_slow_col = NULL,  
  edc_eac_col = "EAC",  
  edc_edc_col = "EDC"  
)
```

Arguments

`eac_ferrihydrite_col`, `eac_goethite_col`, `eac_structural_col`
EAC phase columns.

`edc_humic_fast_col`, `edc_humic_slow_col`
EDC fraction columns.

`edc_eac_col`, `edc_edc_col`
Bulk EAC and EDC fallback columns. Argument names are retained for backward compatibility.

Value

Named list for `rri_accessible_capacity`; k uses inverse hours.

Examples

```
rri_default_reservoirs()
```

rri_domain_influence *Realised influence of each domain on the composite score*

Description

A declared weight is not the same thing as realised influence. Two domains given equal weight contribute unequally to the composite whenever their scores differ in dispersion, in how strongly they covary with the other domains, or in how often they are missing. This function reports what each domain actually contributed, so that a claim such as "the index is driven by soil" can be checked rather than inferred from the pattern of a figure.

Usage

```
rri_domain_influence(
  res,
  domains = c("Physio", "Soil", "Micro"),
  rri_col = "RRI",
  weights = NULL
)
```

Arguments

res	An RRI object from <code>rri_pipeline()</code> or <code>rri_pipeline_st()</code> .
domains	Character vector of domain score columns. Defaults to <code>c("Physio", "Soil", "Micro")</code> .
rri_col	Name of the composite column. Default "RRI".
weights	Optional named numeric vector of the nominal weights used to build the composite. If NULL (default) the function tries <code>res\$effective_weights</code> , then <code>res\$meta\$weights</code> , and otherwise reports realised influence without a nominal comparison.

Details

How realised share is computed. For weights w_d and domain scores S_d , the composite is $R = \sum_d w_d S_d$. Because $\text{Var}(R) = \sum_d w_d \text{Cov}(S_d, R)$, the quantity

$$\phi_d = w_d \text{Cov}(S_d, R) / \text{Var}(R)$$

is an exact decomposition: the ϕ_d sum to one. Each domain's share therefore includes its own variance *and* its share of the covariance it has with the other domains. This is the appropriate attribution when domains are correlated, which they generally are under a shared forcing.

How to read ratio. ratio is realised share divided by nominal weight. A value near 1 means the domain influenced the composite about as much as intended. Values above roughly 1.3 or below roughly 0.7 indicate that the declared weights are not delivering the intended balance, usually for one of three reasons: the domain score is more (or less) dispersed than the others after scaling; it covaries strongly with the others, so it absorbs shared variance; or it is missing for many rows, so per-row weight renormalisation quietly redistributes its weight.

What this does not establish. A high realised share is a statement about the score, not about the ecosystem. It does not show that the domain is mechanistically more important, and it is not evidence that the composite is wrong. It shows only where the variance in this particular composite came from, for this cohort, under these weights and this scaling.

Value

A list with three elements.

`influence` One row per domain: mean, sd, n_missing, cor_with_rri, nominal_weight, realised_share and ratio.

`covariance` Pairwise correlations between domain scores. Strong cross-domain correlation means influence cannot be attributed cleanly to one domain.

`notes` Character vector of diagnostics worth acting on.

See Also

[rri_sensitivity\(\)](#) for the effect of alternative weight grids; [rri_compensation_index\(\)](#) for cross-domain asynchrony.

Examples

```
sim <- simulate_redox_holobiont(
  n_plot = 2, n_depth = 2, n_plant = 3, n_time = 40,
  p_micro = 25, seed = 2026
)

res <- suppressWarnings(rri_pipeline_st(
  ROS_flux = sim$ROS_flux,
  Eh_stability = sim$Eh_stability,
  micro_data = sim$micro_data,
  id = sim$id,
  reducer = "per_domain",
  scaling = "pnorm"
))

infl <- rri_domain_influence(res)
infl$influence
infl$notes
```

rri_kinetics_score	<i>Descriptive recovery speed score</i>
--------------------	---

Description

Combines response lag and one rate descriptor. By default k and $\log(2)/k$ are not counted as separate evidence. No causal exchange rate is inferred.

Usage

```
rri_kinetics_score(
  rec,
  forcing_window = NULL,
  lag_weight = 0.3,
  rate_weight = 0.7,
  halflife_weight = 0,
  invert_slow = TRUE
)
```

Arguments

rec Recovery metric data frame.

forcing_window Positive duration in the same units as recovery time. With a duration, speed is $kT/(1+kT)$, and lag score is $1/(1+\text{lag}/T)$. Without one, scores are cohort-relative min-max descriptions.

lag_weight, rate_weight, halflife_weight Non-negative weights.

invert_slow TRUE scores faster recovery higher; FALSE reverses all components.

Value

Input with kinetics_score, component coverage, heuristic class, and lag ratio.

Examples

```
rec <- data.frame(tau_lag = c(2, 4, 1), k_recovery = c(0.3, 0.1, 0.5))
rri_kinetics_score(rec)$kinetics_score
```

```
rri_latent_correlation
```

Correlation with a Simulator-Defined Target

Description

Computes a descriptive correlation between per-sample RRI and a prescribed simulator target. This is an internal simulation benchmark and is not predictive accuracy, empirical validation or recovery of a true latent state.

Usage

```
rri_latent_correlation(
  res,
  latent_truth,
  method = c("pearson", "spearman", "kendall")
)
```

Arguments

res	An object returned by <code>rri_pipeline_st()</code> .
latent_truth	Numeric simulator-target vector. The legacy argument name is retained for compatibility.
method	Correlation method. One of "pearson", "spearman", or "kendall".

Details

This function is designed for simulation benchmarking. In empirical datasets, no known latent state exists and this metric should not be used.

Value

A single numeric correlation coefficient.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline_st(sim$ROS_flux, sim$Eh_stability)
rri_latent_correlation(res, sim$latent_truth)
```

rri_memory_index	<i>Persistent-displacement and loop-area diagnostic</i>
------------------	---

Description

A descriptive composite, not proof of ecological memory. Slow relaxation, baseline drift and continuing forcing can also produce displacement.

Usage

```
rri_memory_index(
  rec,
  H_weight = 0.5,
  I_weight = 0.5,
  lag_weight = 0,
  normalise_inputs = FALSE
)
```

Arguments

rec	Recovery metric table.
H_weight, I_weight, lag_weight	Non-negative component weights.
normalise_inputs	FALSE uses bounded dimensionless H and I fractions; TRUE requests cohort-relative min-max scaling. Lag requires TRUE.

Value

Input with memory_index, memory_coverage and heuristic memory_class.

Examples

```
rec <- data.frame(H_hysteresis = c(0.1, 0.3, 0.05),
                  incomplete_return_frac = c(0.2, 0.4, 0.1))
rri_memory_index(rec)$memory_index
```

```
rri_micro_functional_score
```

Construct an explicitly weighted microbial guild contrast

Description

Summarises supplied guild measurements or proxies. The weights define a contrast, not a universal ordering of microbial resilience. Denitrification, sulfate reduction and methanogenesis may be beneficial or detrimental depending on the specified ecosystem function and disturbance.

Usage

```
rri_micro_functional_score(micro_traits, weights = NULL, scale = TRUE)
```

Arguments

micro_traits	Numeric data frame with comparable, justified guild scales. Gene abundance or expression does not by itself measure process rate.
weights	Named finite signed weights. Default legacy weights are illustrative only and trigger a warning; supply scientifically justified weights.
scale	If TRUE, scale finite contrasts within the supplied cohort to [0, 1]. Constants map to 0.5 and wholly unobserved rows remain NA.

Value

Numeric vector with coverage and raw_contrast attributes. The raw contrast is a signed weighted sum divided by the available absolute weight. Missing guilds are not zeros; changing availability changes the estimand.

See Also

rri_reference_scores

Examples

```
x <- data.frame(EET_reduction = c(0.2, 0.4, NA),
                methanogenesis = c(0.3, 0.1, NA))
rri_micro_functional_score(x, weights = c(EET_reduction = 1, methanogenesis = -1))
```

rri_o2_demand

*Stoichiometric O₂ Demand from Reduced-Pool Inventories***Description**

Computes the complete-oxidation O₂ demand for a given rhizosphere reduced-pool inventory and compares it to the specified O₂ stock on the same dry-soil mass basis:

$$O_2^{\text{demand}} = \sum_j n_j \cdot s_j$$

where n_j is the molar inventory (mmol kg⁻¹) of reduced species j and s_j is the stoichiometric O₂ coefficient for complete oxidation to the specified endpoint.

Stoichiometric coefficients follow the electron balance table of Ghotbi, Ghotbi, Mühling and Stukenbrock (Box 1 of the mechanistic review, submitted):

Reduced pool	Endpoint	O ₂ (mol mol ⁻¹)
Fe ²⁺	Fe(III) oxyhydroxide	0.25
Mn ²⁺	MnO ₂	0.50
HS ⁻	SO ₄ ²⁻	2.00
FeS (mackinawite)	Fe(III) + SO ₄ ²⁻	2.25
FeS ₂ (pyrite)	Fe(III) + 2SO ₄ ²⁻	3.75
NH ₄ ⁺	NO ₃ ⁻	2.00
CH ₄	CO ₂	2.00
Acetate equivalents	CO ₂	2.00

The *O₂ deficit ratio* (o2_deficit_ratio) is $O_2^{\text{demand}}/O_2^{\text{supply}}$: values > 1 indicate that the specified O₂ stock is smaller than the demand. This stock ratio does not determine recovery or account for continuing O₂ delivery.

Usage

```
rri_o2_demand(
  soil_df,
  fe2_col = NULL,
  mn2_col = NULL,
  hs_col = NULL,
  fes_col = NULL,
  fes2_col = NULL,
  nh4_col = NULL,
  ch4_col = NULL,
  acetate_col = NULL,
  ch4_unit = c("mmol_kg", "umol_kg"),
  acetate_basis = c("acetate", "carbon"),
  o2_supply_col = NULL,
```

```

    custom_coefs = NULL,
    bulk_density = NULL,
    theta_v = NULL,
    particle_density = 2.65,
    return_components = TRUE
  )

```

Arguments

<code>soil_df</code>	Data frame with soil chemistry (rows = samples).
<code>fe2_col</code>	Character or NULL. Column for Fe^{2+} (mmol kg^{-1}). Default stoichiometric coefficient: 0.25.
<code>mn2_col</code>	Character or NULL. Column for Mn^{2+} (mmol kg^{-1}). Coefficient: 0.50.
<code>hs_col</code>	Character or NULL. Column for HS^- (mmol kg^{-1}). Coefficient: 2.00.
<code>fes_col</code>	Character or NULL. Column for FeS/mackinawite (mmol kg^{-1}). Coefficient: 2.25.
<code>fes2_col</code>	Character or NULL. Column for FeS_2 /pyrite (mmol kg^{-1}). Coefficient: 3.75.
<code>nh4_col</code>	Character or NULL. Column for NH_4^+ (mmol kg^{-1}). Coefficient: 2.00.
<code>ch4_col</code>	Character or NULL. Column for CH_4 .
<code>acetate_col</code>	Character or NULL. Column for dissolved organic matter expressed as acetate or acetate-carbon equivalents.
<code>ch4_unit</code>	Character. Unit of <code>ch4_col</code> : "mmol_kg" (default) or "umol_kg".
<code>acetate_basis</code>	Character. "acetate" (default; 2 mol O_2 per mol acetate) or "carbon" (1 mol O_2 per mol acetate-C).
<code>o2_supply_col</code>	Character or NULL. Column for an explicitly defined O_2 inventory ($\text{mmol O}_2 \text{ kg}^{-1}$). If NULL, deficit ratio is not computed.
<code>custom_coefs</code>	Optional named numeric vector to override or extend the built-in stoichiometric coefficients. Names must match the argument names above (e.g., <code>c(fe2_col = 0.25)</code>). Useful for system-specific endpoint assumptions.
<code>bulk_density</code>	Numeric of length one or <code>nrow(soil_df)</code> . Soil bulk density (g cm^{-3}) for volumetric conversion of demand to $\text{mmol O}_2 \text{ L}^{-1}$ porewater. Set to NULL to skip volumetric conversion.
<code>theta_v</code>	Numeric of length one or <code>nrow(soil_df)</code> , or NULL. Volumetric water content (L water L^{-1} bulk soil). When omitted and <code>bulk_density</code> is supplied, saturated porosity is estimated as $1 - \text{bulk_density} / \text{particle_density}$.
<code>particle_density</code>	Numeric. Particle density in g cm^{-3} used only for the saturated-porosity estimate. Default 2.65.
<code>return_components</code>	Logical. If TRUE (default), return a per-species contribution matrix.

Details

Interpretation — the 26-fold contrast.

The mechanistic review (Ghotbi *et al.*, submitted) provides a worked example: a Fe-rich rhizosphere containing 50 mmol Fe(II) kg⁻¹, 5 mmol FeS kg⁻¹, 2 mmol Mn(II) kg⁻¹, 2 mmol NH₄⁺ kg⁻¹, 2 mmol acetate kg⁻¹, and 0.5 mmol CH₄ kg⁻¹ has a complete-oxidation O₂ ceiling of ≈ 34 mmol O₂ kg⁻¹, versus only ≈ 1.3 mmol O₂ kg⁻¹ in an illustrative initial pore-gas stock: about a 26-fold contrast. The assumed stock is not air-saturated porewater. Continuing atmospheric and root O₂ delivery can replenish it. Accessibility, reaction kinetics and transport determine realized demand during an event.

Pyrite stoichiometry.

Complete pyrite oxidation to sulfate and Fe(III) oxyhydroxide releases 4 mol H⁺ mol⁻¹ FeS₂ and consumes 3.75 mol O₂: $\text{FeS}_2 + 15/4 \text{ O}_2 + 7/2 \text{ H}_2\text{O} \rightarrow \text{Fe(OH)}_3 + 2\text{H}_2\text{SO}_4$. Partial oxidation to sulfur intermediates (S⁰, thiosulfate) requires fewer moles; adjust via custom_coefs.

pH coupling.

Under the applicable rate law, homogeneous abiotic Fe(II) oxidation increases ≈100-fold per unit pH rise (Stumm & Lee, 1961; Millero *et al.*, 1987). The stoichiometric demand computed here is for complete oxidation and is independent of pH, but actual O₂ consumption rates will be pH-modulated. This function reports a stoichiometric potential demand, not a thermodynamic limit or rate.

Value

A list:

o2_demand Numeric vector (mmol O₂ kg⁻¹ *drysoil*) of complete-oxidation O₂ demand per sample.

o2_deficit_ratio Per-sample $O_2^{\text{demand}}/O_2^{\text{supply}}$. Values > 1 indicate demand exceeds the specified stock. NA when o2_supply_col is absent.

o2_demand_vol Volumetric O₂ demand (mmol L⁻¹ porewater); NA if bulk_density is NULL.

n_species_used Integer. Number of reduced-pool columns found.

n_species_observed Integer vector. Number of species with a finite inventory in each row, so a low demand from sparse data is not mistaken for a low demand from a small inventory.

species_coverage Per-row fraction of the *requested* species that were observed. Columns named but absent from soil_df count against coverage.

interpretation Character. A one-line reminder that the ratio compares a demand to a stock, not to a delivery rate.

ch4_unit_used, acetate_basis_used The resolved values of ch4_unit and acetate_basis, recorded because both change the numbers returned.

components Data frame (one row per species) with: species, n_observed, stoich_coef, mean_inventory_mmol, mean_o2_contribution, fraction_total_demand. Returned only when return_components = TRUE.

stoich_table Data frame of the coefficients actually applied, one row per species found, including any custom overrides: species_arg, column_used, stoich_coef_O2, endpoint. Species not present in soil_df are absent.

References

- Ghotbi, M., Ghotbi, M., Mühling, K. H., & Stukenbrock, E. H. Rhizosphere redox recovery after hydrological disturbances: mechanisms across the soil–plant–microbiome continuum. *Submitted to Soil Biology & Biochemistry*.
- Stumm, W., & Lee, G. F. (1961). Oxygenation of ferrous iron. *Industrial & Engineering Chemistry*, **53**, 143–146. doi:10.1021/ie50614a030
- Millero, F. J., Sotolongo, S., & Izaguirre, M. (1987). The oxidation kinetics of Fe(II) in seawater. *Geochimica et Cosmochimica Acta*, **51**, 793–801. doi:10.1016/00167037(87)900937

See Also

[rri_accessible_capacity](#), [rri_capacity_index](#), [rri_root_physio](#)

Examples

```
## Reproduce the worked example from Box 1 of the mechanistic review
worked_example <- data.frame(
  Fe2 = 50.0, # mmol kg-1
  FeS = 5.0,
  Mn2 = 2.0,
  NH4 = 2.0,
  acetate = 2.0,
  CH4 = 0.5, # mmol kg-1; set ch4_unit = "umol_kg" for umol input
  O2_pw = 1.3 # assumed initial O2 stock, mmol/kg; NOT air-saturated porewater
)

demand <- rri_o2_demand(
  soil_df = worked_example,
  fe2_col = "Fe2",
  mn2_col = "Mn2",
  fes_col = "FeS",
  nh4_col = "NH4",
  acetate_col = "acetate",
  ch4_col = "CH4",
  o2_supply_col = "O2_pw",
  return_components = TRUE
)

demand$o2_demand # should be ~34 mmol O2 kg-1
demand$o2_deficit_ratio # should be ~26
demand$components
```

Description

Exploratory integration of available numeric domain observations. A larger score is not automatically greater resilience: justify feature orientation, the reference function and the observation window.

Usage

```
rri_pipeline(
  dat = NULL,
  soil = NULL,
  plant = NULL,
  micro = NULL,
  id = NULL,
  domain_weights = c(Physio = 0.4, Soil = 0.35, Micro = 0.25),
  ...
)
```

Arguments

<code>dat</code>	Optional wide data frame with canonical observation names.
<code>soil, plant, micro</code>	Optional numeric data frames with aligned rows. Supply these instead of <code>dat</code> to use custom measurement names or partial panels.
<code>id</code>	Optional identifier data frame in the same row order.
<code>domain_weights</code>	Named nonnegative weights for Physio, Soil and Micro. Available positive weights are renormalized per row; absent domains stay NA.
<code>...</code>	Arguments to <code>rri_pipeline_st</code> , excluding its domain inputs, identifiers and <code>w1/w2/w3</code> (use <code>domain_weights</code> instead).

Details

Known hidden simulator columns are excluded. This is a safeguard, not an automatic detector of every possible source of target leakage. Cohort-fitted PCA and scaling must not be interpreted as a trained predictor. Use `rri_reference_scores` for fixed, independently justified reference anchors. A reduced panel changes the estimand; compare panels through sensitivity analysis rather than treating their scores as interchangeable.

Value

An RRI object with `row_scores`, a `scores` alias, `effective_weights`, per-row `domain_coverage` and `n_domains`, and a `call_mode` field.

See Also

`rri_pipeline_st`, `rri_reference_scores`, `benchmark_hrri`

Examples

```
x <- data.frame(Eh = c(50, 100, 150, 200), pH = c(5, 5.5, 6, 6.5))
z <- rri_pipeline(soil = x, method_soil = "scale",
                 direction_anchor_soil = "Eh")
z$row_scores
```

rri_pipeline_st	<i>Exploratory domain-score integration (legacy interface)</i>
-----------------	--

Description

Integrates plant, soil and microbial latent scores. Direction is not biologically identifiable without justified anchors. All three domains may be incomplete. Available positive domain weights are renormalized per row. Use `rri_reference_scores` for externally anchored, fixed-reference comparisons.

Usage

```
rri_pipeline_st(
  ROS_flux = NULL,
  Eh_stability = NULL,
  micro_data = NULL,
  graph = NULL,
  id = NULL,
  time_col = NULL,
  group_cols = NULL,
  mode = c("snapshot", "rolling", "event"),
  window = 3,
  align = c("right", "center", "left"),
  event_col = NULL,
  baseline_label = "pre",
  recovery_labels = "recovery",
  alpha_micro = 0.5,
  method_phys = "pca",
  method_soil = "pca",
  method_micro = "pca",
  direction_phys = c("auto", "higher_is_better", "lower_is_better"),
  direction_soil = c("auto", "higher_is_better", "lower_is_better"),
  direction_micro = c("auto", "higher_is_better", "lower_is_better"),
  direction_anchor_phys = NULL,
  direction_anchor_soil = NULL,
  direction_anchor_micro = NULL,
  scale_by = NULL,
  network_agg = c("equation", "mean"),
  w1 = 0.4,
  w2 = 0.35,
  w3 = 0.25,
```

```

    add_coupling = FALSE,
    coupling_weight = 0,
    coupling_fun = c("geometric_mean", "agreement"),
    norm_method = NULL,
    reducer = c("per_domain", "mfa"),
    scaling = c("minmax_legacy", "pnorm"),
    comp_space = c("closure_legacy", "clr"),
    ref_stats = NULL,
    add_compensation = FALSE,
    compensation_weight = 0
)

```

Arguments

ROS_flux	Data frame of plant physiological variables (rows = samples).
Eh_stability	Data frame of soil redox chemistry variables (rows = samples).
micro_data	Optional data frame of microbial abundance or functional features.
graph	Optional igraph object or list of igraph objects representing microbial network structure.
id	Optional data frame describing experimental design (same number of rows as inputs).
time_col	Optional character. Name of time column in id.
group_cols	Optional character vector of grouping variables in id.
mode	Character. One of "snapshot", "rolling", or "event".
window	Integer ≥ 2 . Rolling window size (for mode = "rolling").
align	Character. Alignment rule for rolling window: "right", "center", or "left".
event_col	Optional character. Column in id identifying event phases.
baseline_label	Character. Label identifying baseline phase.
recovery_labels	Character vector identifying recovery phases.
alpha_micro	Numeric between 0 and 1 controlling blending of microbial abundance and network components.
method_phys	Character. Reduction method for plant block.
method_soil	Character. Reduction method for soil block.
method_micro	Character. Reduction method for microbial block.
direction_phys	Character. Orientation rule for plant latent dimension.
direction_soil	Character. Orientation rule for soil latent dimension.
direction_micro	Character. Orientation rule for microbial latent dimension.
direction_anchor_phys	Optional character. Anchor variable for plant orientation.
direction_anchor_soil	Optional character. Anchor variable for soil orientation.

<code>direction_anchor_micro</code>	Optional character. Anchor variable for microbial orientation.
<code>scale_by</code>	Optional character vector of grouping variables used for scaling.
<code>network_agg</code>	Character. Network aggregation method: "equation" or "mean".
<code>w1</code>	Numeric weight for plant domain.
<code>w2</code>	Numeric weight for soil domain.
<code>w3</code>	Numeric weight for microbial domain. Must sum with w1 and w2 to 1.
<code>add_coupling</code>	Logical. If TRUE, adds cross-domain coherence term.
<code>coupling_weight</code>	Numeric between 0 and 1 controlling weight of coupling term.
<code>coupling_fun</code>	Character. Coupling function: "geometric_mean" or "agreement".
<code>norm_method</code>	Optional character. If provided, overrides block-specific methods.
<code>reducer</code>	Character. Reduction strategy: "per_domain" or "mfa".
<code>scaling</code>	Character. Scaling rule: "minmax_legacy" or "pnorm".
<code>comp_space</code>	Character. Compositional projection method: "closure_legacy" or "clr".
<code>ref_stats</code>	Optional list of reference statistics used for scaling.
<code>add_compensation</code>	Logical. If TRUE, includes covariance-based compensation term.
<code>compensation_weight</code>	Numeric between 0 and 1 controlling compensation weight.

Details

MFA is disabled pending a validated implementation. Scaling statistics do not freeze PCA/FA loadings, so `ref_stats` is not a trained prediction model. The CLR round trip changes display coordinates only: inversion returns closure. Grouping does not imply within-group scaling; request `scale_by` explicitly. Missing data are median-imputed for exploratory reduction, not corrected for MNAR. Event scores are descriptive products of resistance and reference proximity; baseline/recovery label defaults must be matched to the supplied data.

Value

RRI object; identifiers accompany scores and rolling output retains original input order. Stochastic and advanced reducers need separate validation.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline_st(sim$ROS_flux, sim$Eh_stability, id = sim$id)
head(res$row_scores)
```

rri_property_scores	<i>Summarise supported diagnostics without fabricating missing properties</i>
---------------------	---

Description

Summarise supported diagnostics without fabricating missing properties

Usage

```
rri_property_scores(
  res,
  rec = NULL,
  soil_df = NULL,
  eac_col = "EAC",
  edc_col = "EDC",
  humic_col = NULL,
  connectivity_method = "cross_domain_magnitude",
  H_weight = 0.5,
  I_weight = 0.5,
  forcing_window = NULL
)
```

Arguments

res	RRI result.
rec	Optional recovery table.
soil_df	Optional soil capacity measurements.
eac_col, edc_col, humic_col	Capacity-related columns.
connectivity_method	Association or network summary.
H_weight, I_weight	Memory-diagnostic weights.
forcing_window	Optional timescale for the recovery speed score.

Value

Scores and a provenance table. Unavailable properties stay NA. Capacity here is an oxidative-oriented feature composite, Connectivity an association/topology descriptor, Kinetics a recovery-speed descriptor, and Memory a persistent-displacement descriptor. None proves the named mechanism.

Which inputs each property needs

Only Connectivity is derived from `res` alone. The other three require an additional argument, and are returned as NA with method "unavailable" when it is absent:

- **Capacity** needs `soil_df` containing the columns named by `eac_col` and `edc_col`. Passing only `res` and `rec` is the usual reason Capacity comes back NA.
- **Kinetics** and **Memory** need `rec`, the table returned by `rri_recovery_metrics()`.

A message names any missing input. NA here means "not supplied", never "measured and found to be zero".

Examples

```
sim <- simulate_redox_holobiont(
  n_plot = 2, n_depth = 2, n_plant = 2, n_time = 40, p_micro = 10,
  seed = 2026
)
res <- suppressWarnings(rri_pipeline_st(
  ROS_flux = sim$ROS_flux, Eh_stability = sim$Eh_stability,
  micro_data = sim$micro_data, id = sim$id
))
rec <- rri_recovery_metrics(
  res = res, id = sim$id, time_col = "time",
  group_cols = c("plot", "depth", "plant_id"),
  perturb_start = 12, perturb_end = 22
)

## All four properties available: soil_df supplies Capacity, rec supplies
## Kinetics and Memory.
full <- rri_property_scores(res, rec = rec, soil_df = sim$soil_data)
full$property_table

## Omitting soil_df leaves Capacity unavailable, and says so.
partial <- rri_property_scores(res, rec = rec)
partial$property_table
```

`rri_recovery_metrics` *Descriptive recovery metrics for a single disturbance*

Description

Summarises decline and return of a higher-is-better score. A score decline does not identify pathway truncation; a displaced plateau does not establish alternative electron routing. Hysteresis is only reported for a sufficiently closed, reversing forcing-response path. Temporal deficit asymmetry is a separate diagnostic. Analyse repeated events separately.

Usage

```
rri_recovery_metrics(
  res,
  id = NULL,
  time_col = "time",
  group_cols = NULL,
  perturb_start,
  perturb_end,
  rri_col = "RRI",
  forcing_col = NULL,
  min_pts = 3L,
  lag_threshold = 0.05,
  plateau_window = 3L,
  plateau_tol = 0.1
)
```

Arguments

<code>res</code>	RRI object or data frame.
<code>id</code>	Optional aligned identifiers; common columns must agree.
<code>time_col</code>	Numeric time column; time must be unique within each group.
<code>group_cols</code>	Columns identifying one longitudinal experimental unit.
<code>perturb_start, perturb_end</code>	Finite start and end of one disturbance.
<code>rri_col</code>	Numeric score column.
<code>forcing_col</code>	Optional measured external forcing column, not a response proxy.
<code>min_pts</code>	Minimum finite baseline and recovery observations.
<code>lag_threshold</code>	Fraction of observed decline defining recovery onset.
<code>plateau_window</code>	Number of final observations for plateau assessment.
<code>plateau_tol</code>	Fractional terminal displacement defining a plateau flag.

Value

One row per group, including diagnostic fit status and observation counts. `k` is a log-linear fit of positive baseline deficits after the observed minimum. It is a conditional trajectory descriptor, not a mechanistic exchange rate. Legacy `alt_routing` fields are retained as NA; use `displaced_plateau_flag`.

Why `k_recovery` and `t_half` are often NA

The rate fit is deliberately conservative. It uses only recovery observations that lie after the observed trough, strictly below the pre-event baseline, and before the first crossing back through that baseline, and it requires at least `min_pts` such points. Short recovery windows routinely leave fewer, in which case `fit_status` is `"insufficient_positive_deficits"` and both `k_recovery` and `t_half` are returned as NA rather than being fitted to two points.

This is missingness by design, not failure. A rate estimated from a handful of points spanning less than one recovery time constant is not informative, and reporting it would invite over-interpretation. Always read `k_recovery` together with `fit_status`, `n_fit` and `fit_r_squared`.

If most trajectories return NA, extend the observation window rather than lowering `min_pts`: as a rule of thumb the record should continue for at least two or three times the expected recovery time after `perturb_end`.

Examples

```
## A window long enough for the rate fit to succeed. The event occupies
## days 12-22 of a 40-day record, leaving 18 recovery observations.
sim <- simulate_redox_holobiont(
  n_plot = 2, n_depth = 2, n_plant = 2, n_time = 40,
  p_micro = 10, seed = 2026
)

res <- suppressWarnings(rri_pipeline_st(
  ROS_flux = sim$ROS_flux,
  Eh_stability = sim$Eh_stability,
  micro_data = sim$micro_data,
  id = sim$id,
  reducer = "per_domain",
  scaling = "pnorm"
))

rec <- rri_recovery_metrics(
  res = res, id = sim$id, time_col = "time",
  group_cols = c("plot", "depth", "plant_id"),
  perturb_start = 12, perturb_end = 22
)

## Inspect estimability before using any rate.
table(rec$fit_status)
rec[, c("plot", "depth", "plant_id", "depth_min_frac",
       "k_recovery", "n_fit", "fit_status")]

## Contrast: a short window leaves too few usable points and the rate
## is correctly withheld.
short <- simulate_redox_holobiont(
  n_plot = 1, n_depth = 1, n_plant = 2, n_time = 12,
  p_micro = 5, seed = 1
)
res_s <- suppressWarnings(rri_pipeline_st(
  ROS_flux = short$ROS_flux, Eh_stability = short$Eh_stability,
  micro_data = short$micro_data, id = short$id
))
rec_s <- rri_recovery_metrics(
  res = res_s, id = short$id, time_col = "time",
  group_cols = c("plot", "depth", "plant_id"),
  perturb_start = 5, perturb_end = 7
)
table(rec_s$fit_status)
```

rri_reference_scores *Score departures from an explicitly defined reference*

Description

An optional, transparent alternative to latent-axis scoring. Each feature score is $\max(0, 1 - \text{abs}(\text{value} - \text{target}) / \text{tolerance})$. The result measures proximity to the declared reference, not validated ecosystem functioning or a universal resilience scale.

Usage

```
rri_reference_scores(
  data,
  reference,
  id = NULL,
  domain_weights = c(Physio = 0.4, Soil = 0.35, Micro = 0.25),
  min_coverage = 0.5,
  na_policy = c("available", "complete")
)
```

Arguments

data	Numeric feature data frame; rows must be aligned with id.
reference	Data frame with feature, domain, target, tolerance, weight. Domains are Physio, Soil or Micro. Tolerance is a positive distance from target at which the feature score reaches zero. Reference rows for unmeasured features may be retained to report coverage against a common panel.
id	Optional aligned identifiers.
domain_weights	Named, non-negative domain weights.
min_coverage	Minimum weighted within-domain feature coverage.
na_policy	Use available domains or require every positive-weight domain.

Value

An RRI object with fixed-reference domain scores, feature scores, coverage, effective row-specific domain weights, and reference metadata.

Examples

```
dat <- data.frame(Eh = c(100, 200, 150), pH = c(5.5, 6.0, 5.8))
ref <- data.frame(feature = c("Eh", "pH"), domain = c("Soil", "Soil"),
  target = c(150, 5.8), tolerance = c(100, 0.5), weight = 1)
rri_reference_scores(dat, ref)$row_scores
```

rri_root_physio	<i>Exploratory root-trait composite</i>
-----------------	---

Description

A weighted standardized trait summary. Trait direction is context-dependent: greater ROL, porosity, aerenchyma or SRL is not universally better plant performance or greater oxygen delivery to every root region.

Usage

```
rri_root_physio(
  plant_df,
  biomass_col = NULL,
  length_col = NULL,
  rol_col = NULL,
  aerenchyma_col = NULL,
  porosity_col = NULL,
  srl_col = NULL,
  w_biomass = 0.2,
  w_length = 0.2,
  w_rol = 0.3,
  w_aerenchyma = 0.2,
  w_porosity = 0.05,
  w_srl = 0.05,
  directions = NULL,
  scaling = c("pnorm", "minmax")
)
```

Arguments

plant_df	Numeric root-trait measurements.
biomass_col, length_col	Columns for root biomass and length/density.
rol_col	Column for measured radial oxygen loss, with consistent units.
aerenchyma_col, porosity_col	Optional aeration trait columns; avoid double weighting correlated measures of the same anatomical attribute.
srl_col	Column for specific root length.
w_biomass, w_length, w_rol, w_aerenchyma, w_porosity, w_srl	Non-negative weights.
directions	Optional named numeric vector assigning +1 (larger maps to a larger score) or -1 (larger maps to a smaller score) to each selected measurement column. NULL uses +1 for backward compatibility and warns.
scaling	pnorm (normal-CDF scaling, not a calibrated probability) or minmax.

Value

root_physio_score and trait contributions; unobserved values stay NA.

Examples

```
df <- data.frame(ROL = c(0.5, 1.0, 1.5), biomass = c(10, 20, 30))
rri_root_physio(df, rol_col = "ROL", biomass_col = "biomass",
  directions = c(ROL = 1, biomass = 1))$root_physio_score
```

rri_sensitivity	<i>Sensitivity to domain aggregation weights</i>
-----------------	--

Description

Sensitivity to domain aggregation weights

Usage

```
rri_sensitivity(res, weight_grid = seq(0.2, 0.6, by = 0.1))
```

Arguments

res	RRI result.
weight_grid	Plant weights in (0,1), or a data frame/matrix with named Physio, Soil, Micro columns specifying complete alternative weights.

Value

Alternative normalized weights, finite-pair count and Spearman correlation. This conditions on the already computed features, reductions and missingness.

Examples

```
sim <- simulate_redox_holobiont(seed = 1)
res <- rri_pipeline(soil = sim$Eh_stability, plant = sim$ROS_flux)
rri_sensitivity(res)
```

rri_simulation_demo	<i>Reproducible observable-only HRRI demonstration</i>
---------------------	--

Description

Generates both forcing scenarios with the package simulator, calibrates illustrative targets using a separate baseline simulation, and compares full, 4-soil/3-plant/2-microbe, and single-domain panels with unchanged feature targets and tolerances. No hidden capacity, alpha, k, memory or latent_truth column enters the observed-feature index. This is a software demonstration, not a validation of latent-state recovery or ecological prediction.

Usage

```
rri_simulation_demo(seed = 20260830L)
```

Arguments

seed	Non-negative integer seed.
------	----------------------------

Value

Simulation objects, explicit reference specifications and all plotted tables. The capacity-horizon table is an internal equation check using known synthetic parameters; it is deliberately separate from observable scoring.

Examples

```
demo <- rri_simulation_demo(seed = 20260830L)
head(demo$scores)
```

simulate_redox_holobiont	<i>Simulate illustrative soil-plant-microbe redox trajectories</i>
--------------------------	--

Description

Synthetic daily trajectories with explicit Fe/Mn redistribution, plant indicators, gene abundance and transcript/count observation models. Only Fe and Mn inventories have closed-balance checks. C, N, S and oxygen budgets are not fully balanced. Model parameters are illustrative, not fitted. Gene abundance indicates potential; transcript counts are observations, not flux.

Usage

```
simulate_redox_holobiont(
  n_plot = 4,
  n_depth = 2,
  n_plant = 6,
  n_time = 30,
  p_micro = 60,
  seed = NULL,
  scenario = c("flood_drain", "drought_rewet"),
  n_cycles = 2L,
  disturbance_strength = 0.65,
  disturbance_center = NULL,
  disturbance_width = 0.08,
  seasonal_amp = 0.08,
  seasonal_phase = 0,
  history_strength = 0.55,
  rescue = c("none", "capacity", "connectivity", "kinetics"),
  event_tau_h = 24,
  sequencing_depth = 2e+05,
  metat_depth = 5e+05,
  decoupling = 0.25,
  zero_inflation = 0.2,
  MNAR_strength = 0.3,
  Eh_dropout_threshold = 100,
  micro_mean = 8,
  micro_slope = 3,
  micro_lambda_min = 1e-08,
  micro_lambda_max = 1e+06,
  stochastic_reassembly = TRUE,
  include_graph = FALSE,
  depth_labels = NULL
)
```

Arguments

<code>n_plot</code>	Positive integer. Number of plots (spatial replicates).
<code>n_depth</code>	Positive integer. Number of depth strata per plot.
<code>n_plant</code>	Positive integer. Number of plants per plot-depth unit. Must be ≥ 1 . Plants contribute to ROL and ROS signals.
<code>n_time</code>	Positive integer ≥ 4 . Number of daily time steps.
<code>p_micro</code>	Positive integer. Number of ASV-like microbial taxonomic features generated alongside functional gene data.
<code>seed</code>	Integer or NULL. Random seed passed to <code>set.seed</code> before simulation; NULL means no seeding (non-reproducible). All manuscript figures use explicit seeds. Defaults to NULL, so the function does not touch the random stream unless a seed is requested. When one is supplied, the RNG kind and <code>.Random.seed</code> are saved

	on entry and restored with <code>on.exit</code> , so the caller's stream is returned exactly as found, including when the function exits on an error.
scenario	Character; one of "flood_drain" (default) or "drought_rewet". Determines the shape of the hydrological forcing function and the sign of the dominant redox transition.
n_cycles	Positive integer. Number of forcing pulses.
disturbance_strength	Numeric in $[0, 1]$. Event severity. Controls peak WFPS, anaerobic volume fraction, and the amplitude of soil redox transitions.
disturbance_center	Numeric or NULL. Time step of the first disturbance event centre. Defaults to evenly spaced centres from 0.22 to 0.78 of <code>n_time</code> .
disturbance_width	Numeric in $(0, 1)$. Width of each Gaussian forcing pulse as a fraction of <code>n_time</code> .
seasonal_amp	Numeric ≥ 0 . Amplitude of additive seasonal forcing overlaid on the hydrological disturbance signal.
seasonal_phase	Numeric. Phase offset (radians) of the seasonal forcing.
history_strength	Numeric in $[0, 1]$. Scales the synthetic memory state at initialization and during disturbance. That state affects Fe crystallisation, accessibility and generated microbial descriptors; it is not a measured fraction of community carry-over.
rescue	Character; one of "none" (default), "capacity", "connectivity", or "kinetics". Simulates a targeted scenario modification (capacity at initialization; alpha/k throughout): "capacity" Fe(III) inventory replenishment. "connectivity" Increases <code>alpha_accept</code> and <code>alpha_donate</code> in selected Fe/Mn/N/S/C rate expressions and calculated accessible capacity. Effects on any recovery outcome must be evaluated, not assumed beneficial. "kinetics" Increases both calculated accessible-capacity rates. <code>k_accept</code> additionally gates crystalline-Fe reduction; <code>k_donate</code> has no direct process-rate gate. This intervention is not a general exchange-kinetics model.
event_tau_h	Positive numeric. Disturbance timescale τ (h) used internally by the accessible-capacity calculation; passed to <code>rri_accessible_capacity</code> .
sequencing_depth	Positive numeric. Mean library size for taxonomic count data (<code>micro_data</code> block), modelled as a negative-binomial process.
metat_depth	Positive numeric. Mean library size for metatranscript counts (<code>micro_metat_counts</code>), modelled separately from taxonomic counts with a higher biological variance.
decoupling	Numeric in $[0, 1]$. Cross-domain stochastic decoupling parameter for selected noise terms; other stochastic terms remain.
zero_inflation	Numeric in $[0, 1]$. Structural-zero probability for taxonomic count features (simulates taxa absent from some samples).
MNAR_strength	Numeric in $[0, 1]$. Maximum missing-not-at-random (MNAR) probability for Eh values under strongly reducing conditions.

Eh_dropout_threshold	Numeric. Eh (mV) below which the MNAR dropout probability begins to rise; is an artificial missingness design, not a platinum-electrode detection limit.
micro_mean, micro_slope, micro_lambda_min, micro_lambda_max	Backward-compatible parameters controlling mean and slope of the log-linear model for taxonomic count intensity. See legacy documentation.
stochastic_reassembly	Logical. If TRUE (default), adds additional stochastic variation; this is not an explicit succession model.
include_graph	Logical. If TRUE and igraph is installed, returns an igraph random graph object independent of the generated community in \$graph.
depth_labels	Character vector of length n_depth or NULL. Custom labels for depth strata. Defaults to "D1", "D2", ...

Details

alpha and k are prescribed internal state variables, not inferred parameters: they are computed from forcing, pore structure and memory at the start of each time step. Alpha multiplies selected Fe/Mn/N/S/C expressions as an accessibility factor, whereas k_accept directly gates only the model's crystalline-Fe reduction expression. A low alpha suppresses selected rates; restoring alpha changes selected rates, but does not guarantee functional recovery. k additionally gates the crystalline-Fe exchange rate. The latent_truth vector is a constructed index sharing ingredients with soil_data; omit those ingredients from observable-only benchmarks. history_pair denotes a shared random effect, not experimentally matched twins. Low n_time can leave no adequate baseline or recovery; inspect forcing and analyse each event separately. Mineral crystallisation is continuous in this implementation, not an event-only reoxidation ratchet. k does not explicitly decrease with crystallinity. ROL, ROS and several microbial features are synthetic descriptors, not calibrated physical fluxes. Do not infer quantitative field rates from their labels alone. Q_accept now counts one electron per crystalline Fe(III); the previous 0.22 factor mixed accessibility into inventory. DOC reducing equivalents remain an illustrative coefficient dependent on assumed carbon oxidation state.

Memory (M) is a holobiont state, not a mineralogical one. It accumulates from four sources with weights summing to 0.060 per step, scaled by history_strength: hydrological event load (0.020), the Fe-crystallinity ratchet (0.016), persistent plant acclimation measured as aerenchyma displacement above the naive baseline (0.012), and microbial community displacement (0.012). The microbial component is itself a state (micro_legacy) that accrues under sustained reduction and relaxes more slowly under oxic recovery, so community composition carries an asymmetric legacy. Memory decay remains keyed to Fe crystallinity because mineral ordering is the least reversible component and sets the floor on memory loss. The component series are returned in latent_state as micro_legacy and plant_legacy so the decomposition is auditable. These are illustrative weights, not calibrated rates.

Value

List of identifiers, data blocks, latent states, flux descriptors, balance checks, metadata and legacy views. The metadata records hidden columns, seed and RNG configuration. With seed supplied, the caller RNG is restored.

Examples

```
sim <- simulate_redox_holobiont(n_plot = 2, n_depth = 2, n_plant = 2,  
                               n_time = 20, seed = 42)  
nrow(sim$id) # 2 x 2 x 2 x 20 = 160 rows  
names(sim)   # top-level list elements
```

theme_ems

EMS plotting theme

Description

A simple ggplot theme used for HRRI visualizations.

Usage

```
theme_ems(base_size = 12)
```

Arguments

base_size Base font size

Value

A ggplot2 theme object

Examples

```
library(ggplot2)  
ggplot(data.frame(x = 1:3, y = 1:3), aes(x, y)) +  
  geom_point() + theme_ems()
```

Index

`as.data.frame.hrri_arch`
 (`hrri_infer_architecture`), 5
`attach_hrri_ids`, 3

`benchmark_hrri`, 4
`benchmark_hrri()`, 27

`hrri_infer_architecture`, 5
`HRRI_workflow_example`, 7

`on.exit`, 54

`plot_hrri_benchmark`, 10
`plot_rri_accuracy`, 11
`plot_rri_accuracy()`, 27
`plot_rri_properties`, 12
`plot_rri_recovery_landscape`, 14
`plot_rri_recovery_landscape()`, 8
`plot_rri_recovery_map`, 16
`plot_rri_recovery_map()`, 16
`plot_rri_simulation_demo`, 18
`plot_rri_state_space`, 19
`plot_RRI_ternary`, 20
`plot_RRI_ternary()`, 8
`plot_rri_timeseries`, 22
`plot_rri_validation`, 23
`print.hrri_arch`
 (`hrri_infer_architecture`), 5
`print.hrri_arch_validation`
 (`hrri_infer_architecture`), 5
`print.hrri_benchmark` (`benchmark_hrri`), 4
`print.rri_accuracy` (`rri_accuracy`), 25

`redoxrri_example`
 (`HRRI_workflow_example`), 7
`rri_accessible_capacity`, 24, 40, 54
`rri_accuracy`, 25
`rri_accuracy()`, 11, 12
`rri_capacity_index`, 28, 40
`rri_compensation_index`, 29
`rri_compensation_index()`, 33

`rri_connectivity_score`, 30
`rri_default_reservoirs`, 31
`rri_domain_influence`, 32
`rri_domain_influence()`, 27
`rri_kinetics_score`, 33
`rri_latent_correlation`, 34
`rri_memory_index`, 35
`rri_micro_functional_score`, 36
`rri_o2_demand`, 37
`rri_pipeline`, 3, 4, 40
`rri_pipeline()`, 32
`rri_pipeline_st`, 3, 4, 17, 42
`rri_pipeline_st()`, 8, 32
`rri_property_scores`, 12, 13, 45
`rri_recovery_metrics`, 17, 46
`rri_recovery_metrics()`, 8, 14–16, 46
`rri_reference_scores`, 49
`rri_root_physio`, 40, 50
`rri_sensitivity`, 51
`rri_sensitivity()`, 33
`rri_simulation_demo`, 52

`set.seed`, 53
`simulate_redox_holobiont`, 3, 52
`simulate_redox_holobiont()`, 7, 8, 26
`summary.hrri_arch`
 (`hrri_infer_architecture`), 5

`theme_ems`, 56
`theme_ems()`, 11

`validate_architecture`
 (`hrri_infer_architecture`), 5