Package 'palimpsestr'

May 9, 2026

Type Package

Title Probabilistic Decomposition of Archaeological Palimpsests

Version 0.10.0

Description Probabilistic framework for the analysis of archaeological palimpsests based on the Stratigraphic Entanglement Field (SEF). Integrates spatial proximity, stratigraphic depth, chronological overlap, and cultural similarity to estimate latent depositional phases via diagonal Gaussian mixture Expectation-Maximisation (EM). Provides the Stratigraphic Entanglement Index (SEI), Excavation Stratigraphic Energy (ESE), and Palimpsest Dissolution Index (PDI) for quantifying depositional coherence, detecting intrusive finds, and measuring palimpsest formation. Includes simulation, diagnostics, phase-count selection, publication-quality plots, and Geographic Information System (GIS) export via 'sf'. Methods are described in Cocca (2026) <https://github.com/enzococca/palimpsestr>.

URL https://github.com/enzococca/palimpsestr

BugReports https://github.com/enzococca/palimpsestr/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports stats, utils, graphics, grDevices

Suggests testthat (>= 3.0.0), knitr, rmarkdown, sf, ggplot2, viridis, ggrepel, plotly, DBI, RSQLite, RPostgres

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Enzo Cocca [aut, cre] (ORCID: <https://orcid.org/0000-0002-8096-2810>)

Maintainer Enzo Cocca <enzo.ccc@gmail.com>

Repository CRAN

Date/Publication 2026-04-09 09:00:21 UTC

Contents

`adjusted_rand_index` *Adjusted Rand Index*

Description

Compares estimated phase assignments against known true labels using the Adjusted Rand Index (Hubert and Arabie, 1985). Values near 1 indicate perfect agreement; values near 0 indicate random agreement.

Usage

```
adjusted_rand_index(object, true_labels)
```

Arguments

`object`	A `sef_fit` object, or an integer vector of predicted labels.
`true_labels`	Integer vector of known phase assignments.

Value

A single numeric value in $[-1, 1]$.

See Also

`confusion_matrix`, `fit_sef`

Other validation: `bootstrap_sef()`, `confusion_matrix()`, `cv_sef()`, `optimize_weights()`

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1, mixing = 0.05)
fit <- fit_sef(x, k = 3, seed = 1)
adjusted_rand_index(fit, x$true_phase)
```

`archaeo_sim` *Simulate an archaeological palimpsest dataset*

Description

Generates a synthetic excavation dataset with known latent phases, controlled spatial clustering, and configurable inter-phase mixing.

Usage

```
archaeo_sim(n = 150, k = 3, seed = NULL, mixing = 0.08)
```

Arguments

`n`	Number of observations (finds).
`k`	Number of latent depositional phases.
`seed`	Optional random seed for reproducibility.
`mixing`	Proportion of observations to perturb spatially and taphonomically, simulating post-depositional disturbance (0–1).

Value

A data.frame with columns: `id`, `x`, `y`, `z`, `context`, `date_min`, `date_max`, `class`, `taf_score`, `true_phase`.

See Also

`fit_sef` for fitting the SEF model to the output.

Examples

```
easy <- archaeo_sim(n = 100, k = 3, mixing = 0.05, seed = 1)
table(easy$true_phase)

hard <- archaeo_sim(n = 200, k = 4, mixing = 0.50, seed = 1)
table(hard$true_phase)
```

as_phase_table *Extract phase probability table*

Description

Returns a data.frame combining dominant phase assignments, membership probabilities, entropy, local SEI, and energy.

Usage

```
as_phase_table(object)
```

Arguments

object	A sef_fit object.

Value

A data.frame with one row per find.

See Also

predict_phase, phase_diagnostic_table

Other diagnostics: detect_intrusions(), ese(), pdi(), phase_diagnostic_table(), predict_phase()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
head(as_phase_table(fit))
```

as_plotly *Convert a ggplot to an interactive plotly widget*

Description

Wraps ggplotly with archaeological tooltips showing find ID, context, phase probability, dating, class, entropy, and energy.

Usage

```
as_plotly(gg, tooltip = "text", ...)
```

Arguments

gg	A ggplot object produced by any gg_* function.
tooltip	Character vector of aesthetics to show. Defaults to "text" which displays the enriched archaeological tooltip.
...	Additional arguments passed to ggplotly.

Value

A plotly htmlwidget object.

See Also

gg_phasefield, gg_entropy, gg_energy, gg_intrusions

Other plotting: gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1)
fit <- fit_sef(x, k = 3)
if (requireNamespace("ggplot2", quietly = TRUE) &&
    requireNamespace("plotly", quietly = TRUE)) {
  p <- as_plotly(gg_phasefield(fit))
}
```

---

as_sf_links *Export high-SEI links as an sf LINESTRING layer*

Description

Extracts the strongest pairwise SEI connections as sf line geometries.

Usage

```
as_sf_links(object, quantile_threshold = 0.9, crs = NA_integer_)
```

Arguments

object	A sef_fit object.
quantile_threshold	Quantile for retaining strongest links (default: 0.9).
crs	CRS for the output geometry.

Value

An sf object with columns from, to, sei, and geometry.

See Also

as_sf_phase, sei_matrix

Other GIS: as_sf_phase()

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  x <- archaeo_sim(n = 60, k = 2, seed = 1)
  fit <- fit_sef(x, k = 2)
  links <- as_sf_links(fit)
}
```

---

as_sf_phase *Convert a fitted model to an sf point layer*

Description

Creates an sf point object with phase assignments and diagnostics for use in QGIS or spatial analysis.

Usage

```
as_sf_phase(object, crs = NA_integer_, dims = c("XY", "XYZ"))
```

Arguments

object	A sef_fit object.
crs	CRS passed to st_as_sf.
dims	Either "XY" or "XYZ".

Value

An sf object.

See Also

as_sf_links, phase_diagnostic_table

Other GIS: as_sf_links()

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  x <- archaeo_sim(n = 60, k = 2, seed = 1)
  fit <- fit_sef(x, k = 2)
  pts <- as_sf_phase(fit)
}
```

`bootstrap_sef` *Bootstrap confidence intervals for SEF diagnostics*

Description

Resamples the data with replacement, refits the model, and computes the distribution of key statistics (PDI, mean entropy, mean energy, ARI if true labels provided). Returns percentile confidence intervals.

Usage

```
bootstrap_sef(
  object,
  n_boot = 100,
  conf = 0.95,
  true_labels = NULL,
  verbose = TRUE
)
```

Arguments

`object`	A `sef_fit` object (used to extract fitting parameters).
`n_boot`	Number of bootstrap replicates (default: 100).
`conf`	Confidence level (default: 0.95).
`true_labels`	Optional integer vector of true phase labels for ARI.
`verbose`	Print progress (default: TRUE).

Value

A data.frame with columns `statistic`, `estimate`, `lower`, `upper`, `se`.

See Also

`fit_sef`, `pdi`, `adjusted_rand_index`

Other validation: `adjusted_rand_index()`, `confusion_matrix()`, `cv_sef()`, `optimize_weights()`

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2, seed = 1)
bootstrap_sef(fit, n_boot = 20)
```

`compare_k` *Compare multiple candidate phase counts*

Description

Fits the SEF model for each value of K and returns a summary table with BIC, PDI, entropy, energy, and other diagnostics.

Usage

```
compare_k(data, k_values = 2:6, ...)
```

Arguments

`data`	Input data.frame.
`k_values`	Integer vector of candidate phase counts.
`...`	Additional arguments passed to `fit_sef`.

Value

A data.frame with one row per K value.

See Also

`fit_sef`, `gg_compare_k`

Other fitting: `fit_sef()`, `reorder_phases()`, `sef_summary()`

Examples

```
x <- archaeo_sim(n = 100, k = 3, seed = 1)
ck <- compare_k(x, k_values = 2:4)
print(ck)
```

confusion_matrix *Confusion matrix between estimated and true phases*

Description

Cross-tabulates estimated phase assignments against known true labels. Phases are reordered to maximise diagonal agreement (Hungarian matching).

Usage

```
confusion_matrix(object, true_labels)
```

Arguments

object	A sef_fit object, or an integer vector of predicted labels.
true_labels	Integer vector of known phase assignments.

Value

A matrix with estimated phases as rows and true phases as columns.

See Also

adjusted_rand_index

Other validation: adjusted_rand_index(), bootstrap_sef(), cv_sef(), optimize_weights()

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1, mixing = 0.05)
fit <- fit_sef(x, k = 3, seed = 1)
confusion_matrix(fit, x$true_phase)
```

cv_sef *K-fold cross-validation for SEF model*

Description

Splits the data into folds, fits on training folds, and evaluates log-likelihood on the held-out fold. Useful for comparing different K values or weight configurations.

Usage

```
cv_sef(data, k_values = 2:6, n_folds = 5, seed = 1, ...)
```

Arguments

data	A data.frame with archaeological find data.
k_values	Integer vector of candidate phase counts.
n_folds	Number of cross-validation folds (default: 5).
seed	Random seed for fold assignment.
...	Additional arguments passed to fit_sef.

Value

A data.frame with columns k, fold, train_loglik, test_loglik, train_pdi.

See Also

compare_k, fit_sef

Other validation: adjusted_rand_index(), bootstrap_sef(), confusion_matrix(), optimize_weights()

Examples

```
x <- archaeo_sim(n = 100, k = 3, seed = 1)
cv <- cv_sef(x, k_values = 2:4, n_folds = 3)
# Mean test log-likelihood per K
aggregate(test_loglik ~ k, data = cv, FUN = mean)
```

---

demo_compressed *Demo dataset: compressed palimpsest*

Description

Simulated dataset with 250 artefacts, 4 phases, and 50% mixing. Represents a heavily disturbed deposit where phases are difficult to separate.

Usage

```
demo_compressed
```

Format

A data.frame with 250 rows and 10 columns. See demo_easy for column descriptions.

Examples

```
data(demo_compressed)

ck <- compare_k(demo_compressed, k_values = 2:6)
print(ck)
```

demo_easy *Demo dataset: well-separated phases*

Description

Simulated dataset with 150 artefacts, 3 phases, and 5% mixing. Represents a site where occupation phases are clearly distinguishable.

Usage

```
demo_easy
```

Format

A data.frame with 150 rows and 10 columns:

id Artefact identifier

x, y Planimetric coordinates (metres)

z Depth (metres)

context Stratigraphic unit label

date_min, date_max Chronological interval (CE)

class Cultural class (ceramic, lithic, bone, metal)

taf_score Taphonomic disturbance score (0-1)

true_phase Known phase assignment (for validation)

Examples

```
data(demo_easy)
fit <- fit_sef(demo_easy, k = 3)
plot_phasefield(fit)
```

demo_moderate *Demo dataset: moderate palimpsest*

Description

Simulated dataset with 200 artefacts, 3 phases, and 30% mixing. Represents a site with significant but resolvable depositional mixing.

Usage

```
demo_moderate
```

Format

A data.frame with 200 rows and 10 columns. See demo_easy for column descriptions.

Examples

```
data(demo_moderate)
fit <- fit_sef(demo_moderate, k = 3, tafonomy = "taf_score", context = "context")
summary(fit)
```

---

detect_intrusions *Detect potentially intrusive observations*

Description

Combines rescaled entropy, energy, and inverse local SEI into a composite intrusion probability score.

Usage

```
detect_intrusions(object)
```

Arguments

object	A sef_fit object.

Value

A data.frame with columns id and intrusion_prob.

See Also

gg_intrusions, fit_sef

Other diagnostics: as_phase_table(), ese(), pdi(), phase_diagnostic_table(), predict_phase()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
di <- detect_intrusions(fit)
head(di[order(di$intrusion_prob, decreasing = TRUE), ])
```

ese *Compute Excavation Stratigraphic Energy*

Description

Measures local depositional disruption for each find by summing weighted dissimilarities with neighbours.

Usage

```
ese(
  data,
  coords = c("x", "y", "z"),
  chrono = c("date_min", "date_max"),
  class_col = "class",
  beta = c(1, 1, 1, 1),
  neighbourhood = NULL
)
```

Arguments

data	Input data.frame.
coords	Character vector of coordinate column names.
chrono	Character vector with minimum and maximum dating columns.
class_col	Class column name.
beta	Numeric vector of length 4: weights for spatial, vertical, temporal, and class mismatch.
neighbourhood	Maximum XY distance for neighbour inclusion. When NULL, all observations contribute.

Value

A numeric vector of local energy values.

See Also

fit_sef, gg_energy

Other diagnostics: as_phase_table(), detect_intrusions(), pdi(), phase_diagnostic_table(), predict_phase()

Examples

```
x <- archaeo_sim(n = 30, k = 2, seed = 1)
e <- ese(x)
summary(e)
```

`export_results` *Export all results to files*

Description

Writes phase assignments, intrusion scores, US summary, and model summary to CSV files in the specified directory.

Usage

```
export_results(object, dir, format = "csv", prefix = "palimpsestr")
```

Arguments

`object`	A `sef_fit` object.
`dir`	Output directory (created if it does not exist). Must be specified explicitly; consider using `tempdir()` for temporary output.
`format`	Export format: `"csv"` (default).
`prefix`	File name prefix (default: `"palimpsestr"`).

Value

Invisibly returns a character vector of written file paths.

See Also

`us_summary_table`, `as_phase_table`

Other export: `phase_transition_matrix()`, `us_summary_table()`

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2, context = "context")
export_results(fit, dir = tempdir())
```

`fit_sef` *Fit the Stratigraphic Entanglement Field model*

Description

Estimates latent depositional phases from spatial, stratigraphic, chronological, and cultural evidence using diagonal Gaussian mixture EM.

Usage

```
fit_sef(
  data,
  coords = c("x", "y", "z"),
  chrono = c("date_min", "date_max"),
  class = "class",
  tafonomy = NULL,
  context = NULL,
  harris = NULL,
  k = 3,
  weights = c(ws = 1, wz = 1, wt = 1, wc = 1),
  seed = 1,
  em_iter = 100,
  em_tol = 1e-05,
  n_init = 1
)
```

Arguments

`data`	A data.frame with archaeological find data.
`coords`	Character vector of length 3 naming the x, y, z coordinate columns.
`chrono`	Character vector of length 2 naming the min and max dating columns.
`class`	Character scalar naming the material class column.
`tafonomy`	Optional column name for taphonomic disturbance scores (0–1).
`context`	Optional column name for stratigraphic unit labels.
`harris`	Optional $n \times n$ matrix of pairwise stratigraphic penalties.
`k`	Integer number of phases to estimate.
`weights`	Named numeric vector with components `ws`, `wz`, `wt`, `wc`.
`seed`	Random seed for reproducibility.
`em_iter`	Maximum number of EM iterations (default: 100).
`em_tol`	Convergence tolerance on the log-likelihood.
`n_init`	Number of random initialisations. The run with the highest log-likelihood is retained (default: 1).

Value

An S3 object of class `sef_fit`.

See Also

`archaeo_sim`, `compare_k`, `pdi`, `detect_intrusions`

Other fitting: `compare_k()`, `reorder_phases()`, `sef_summary()`

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
print(fit)


x <- archaeo_sim(n = 150, k = 3, seed = 42)
fit <- fit_sef(x, k = 3, tafonomy = "taf_score", context = "context")
summary(fit)
```

---

gg_bootstrap *Bootstrap confidence interval plot*

Description

Visualises bootstrap confidence intervals for key SEF diagnostics (PDI, entropy, energy, log-likelihood, and optionally ARI).

Usage

```
gg_bootstrap(bs_result)
```

Arguments

`bs_result`	Data.frame returned by `bootstrap_sef`.

Value

A ggplot object.

See Also

`bootstrap_sef`

Other plotting: `as_plotly()`, `gg_compare_k()`, `gg_confusion()`, `gg_convergence()`, `gg_cv()`, `gg_energy()`, `gg_entropy()`, `gg_intrusions()`, `gg_map()`, `gg_phase_profile()`, `gg_phasefield()`, `gg_weights()`, `plot_energy()`, `plot_entropy()`, `plot_phasefield()`, `plot_sei_profile()`

gg_compare_k *Phase count selection diagnostics*

Description

Three-panel plot showing BIC, PDI, and Mean Entropy for different K values.

Usage

```
gg_compare_k(ck)
```

Arguments

ck	Data.frame from compare_k().

Value

A ggplot object.

See Also

compare_k

Other plotting: as_plotly(), gg_bootstrap(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

gg_confusion *Confusion matrix heatmap*

Description

Plots a heatmap of the confusion matrix between estimated and known true phase assignments.

Usage

```
gg_confusion(object, true_labels)
```

Arguments

object	A sef_fit object.
true_labels	Integer vector of known true phase assignments.

Value

A ggplot object.

See Also

confusion_matrix, adjusted_rand_index

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  x <- archaeo_sim(n = 80, k = 3, seed = 1, mixing = 0.05)
  fit <- fit_sef(x, k = 3, seed = 1)
  gg_confusion(fit, x$true_phase)
}
```

---

gg_convergence *EM convergence trace*

Description

Plots the log-likelihood at each EM iteration to verify convergence.

Usage

```
gg_convergence(object)
```

Arguments

object	A sef_fit object.

Value

A ggplot object.

See Also

fit_sef

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  x <- archaeo_sim(n = 80, k = 3, seed = 1)
  fit <- fit_sef(x, k = 3)
  gg_convergence(fit)
}
```

gg_cv *Cross-validation diagnostic plot*

Description

Plots mean test and training log-likelihood across K values from `cv_sef` output.

Usage

```
gg_cv(cv_result)
```

Arguments

`cv_result`	Data.frame returned by `cv_sef`.

Value

A ggplot object.

See Also

`cv_sef`, `gg_compare_k`

Other plotting: `as_plotly()`, `gg_bootstrap()`, `gg_compare_k()`, `gg_confusion()`, `gg_convergence()`, `gg_energy()`, `gg_entropy()`, `gg_intrusions()`, `gg_map()`, `gg_phase_profile()`, `gg_phasefield()`, `gg_weights()`, `plot_energy()`, `plot_entropy()`, `plot_phasefield()`, `plot_sei_profile()`

gg_energy *Excavation Stratigraphic Energy map*

Description

Shows ESE values across space. High energy indicates zones where neighbouring artefacts are dissimilar (depositional disruption).

Usage

```
gg_energy(object, xlabel = "Easting (m)", ylabel = "Northing (m)")
```

Arguments

`object`	A `sef_fit` object.
`xlabel`, `ylabel`	Axis labels.

Value

A ggplot object.

See Also

gg_entropy, as_plotly for interactive version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

---

gg_entropy *Spatial entropy map*

Description

Shows Shannon entropy of phase probabilities across the excavation area. High entropy = uncertain phase assignment (possible palimpsest zone).

Usage

```
gg_entropy(object, xlabel = "Easting (m)", ylabel = "Northing (m)")
```

Arguments

object	A sef_fit object.
xlabel, ylabel	Axis labels.

Value

A ggplot object.

See Also

plot_entropy for base R version, as_plotly for interactive version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

gg_intrusions *Intrusion probability map*

Description

Maps the probability that each find is an intrusion (displaced or redeposited). Top suspects are circled and labelled.

Usage

```
gg_intrusions(
  object,
  top_n = 5,
  xlabel = "Easting (m)",
  ylabel = "Northing (m)"
)
```

Arguments

object	A sef_fit object.
top_n	Number of top intrusions to label.
xlabel, ylabel	Axis labels.

Value

A ggplot object.

See Also

detect_intrusions, as_plotly for interactive version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

gg_map *Overlay SEF results on excavation plan geometries*

Description

Displays SEF analysis results directly on the excavation plan. Polygons represent stratigraphic units; points represent individual finds.

Usage

```
gg_map(
  object,
  geometries,
  layer = c("phase", "entropy", "energy", "intrusion"),
  show_labels = TRUE,
  show_points = TRUE,
  xlabel = NULL,
  ylabel = NULL
)
```

Arguments

object	A sef_fit object.
geometries	An sf object with excavation plan polygons. Must contain a us column matching context values in the data.
layer	What to display: "phase", "entropy", "energy", or "intrusion".
show_labels	Show US labels on polygons.
show_points	Overlay individual finds as points.
xlabel, ylabel	Axis labels (default: NULL = use CRS units).

Value

A ggplot object.

See Also

load_geometries, as_sf_phase

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

---

gg_phasefield *Phase assignment map*

Description

Plots artefact positions coloured by dominant phase, with point size proportional to assignment confidence.

Usage

```
gg_phasefield(object, xlabel = "Easting (m)", ylabel = "Northing (m)")
```

Arguments

object	A sef_fit object.
xlabel, ylabel	Axis labels (default: "Easting (m)" / "Northing (m)").

Value

A ggplot object.

See Also

plot_phasefield for base R version, as_plotly for interactive version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

---

gg_phase_profile *Vertical phase profile*

Description

Plots finds along the depth (z) axis, coloured by phase assignment, to visualise the stratigraphic ordering of phases.

Usage

```
gg_phase_profile(object)
```

Arguments

object	A sef_fit object.

Value

A ggplot object.

See Also

phase_transition_matrix

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield(), plot_sei_profile()

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  x <- archaeo_sim(n = 80, k = 3, seed = 1)
  fit <- fit_sef(x, k = 3)
  gg_phase_profile(fit)
}
```

`gg_weights` *Weight sensitivity heatmap*

Description

Plots the mean test log-likelihood across weight configurations from `optimize_weights` output.

Usage

```
gg_weights(opt_result)
```

Arguments

`opt_result`	List returned by `optimize_weights`.

Value

A ggplot object.

See Also

`optimize_weights`

Other plotting: `as_plotly()`, `gg_bootstrap()`, `gg_compare_k()`, `gg_confusion()`, `gg_convergence()`, `gg_cv()`, `gg_energy()`, `gg_entropy()`, `gg_intrusions()`, `gg_map()`, `gg_phase_profile()`, `gg_phasefield()`, `plot_energy()`, `plot_entropy()`, `plot_phasefield()`, `plot_sei_profile()`

`harris_from_contexts` *Generate stratigraphic penalty matrix from context depth ordering*

Description

Infers vertical ordering between stratigraphic units from the mean depth of finds in each context, and builds a penalty matrix that discourages finds from different vertical zones being assigned to the same phase.

Usage

```
harris_from_contexts(
  data,
  z_col = "z",
  context_col = "context",
  penalty_scale = 0.5
)
```

Arguments

data	A data.frame with find data.
z_col	Name of the depth column.
context_col	Name of the context column.
penalty_scale	Penalty magnitude for cross-context assignments.

Value

An $n \times n$ symmetric penalty matrix.

See Also

fit_sef, read_harris

Other harris: read_harris(), validate_phases_harris()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
H <- harris_from_contexts(x, z_col = "z", context_col = "context")
dim(H)
```

---

load_geometries *Load excavation geometries from file or database*

Description

Reads stratigraphic unit polygons from Shapefile, GeoPackage, GeoJSON, or a PostGIS database for use with gg_map.

Usage

```
load_geometries(
  source,
  layer = NULL,
  query = NULL,
  us_column = "us",
  crs = NULL
)
```

Arguments

source	File path or DBIConnection.
layer	Layer name for multi-layer sources.
query	SQL query for database connections.
us_column	Column containing US/context identifiers.
crs	Target CRS for reprojection (optional).

Value

An sf object with a us column.

See Also

gg_map, read_db

Other data-import: read_db()

---

local_sei *Compute local SEI values*

Description

Aggregates the SEI matrix by row, yielding a per-observation measure of total depositional coherence with all other finds.

Usage

```
local_sei(sei_mat)
```

Arguments

sei_mat	A symmetric SEI matrix from sei_matrix.

Value

A numeric vector of length nrow(sei_mat).

See Also

sei_matrix

Other SEI: sei_matrix(), sei_sparse()

Examples

```
x <- archaeo_sim(n = 30, k = 2, seed = 1)
S <- sei_matrix(x)
lsei <- local_sei(S)
summary(lsei)
```

`optimize_weights` *Estimate optimal SEI weights via cross-validation*

Description

Tests a grid of weight configurations and selects the one that maximises the mean held-out log-likelihood across folds. This provides a data-driven alternative to manual weight specification.

Usage

```
optimize_weights(
  data,
  k,
  weight_grid = NULL,
  n_folds = 3,
  seed = 1,
  verbose = TRUE,
  ...
)
```

Arguments

`data`	A data.frame with archaeological find data.
`k`	Number of phases.
`weight_grid`	A data.frame with columns `ws`, `wz`, `wt`, `wc`. If `NULL`, a default grid is used.
`n_folds`	Number of cross-validation folds (default: 3).
`seed`	Random seed.
`verbose`	Print progress (default: TRUE).
`...`	Additional arguments passed to `fit_sef`.

Value

A list with components:

best_weights Named numeric vector of optimal weights.

results Data.frame with all tested configurations and their scores.

See Also

`fit_sef`, `cv_sef`

Other validation: `adjusted_rand_index()`, `bootstrap_sef()`, `confusion_matrix()`, `cv_sef()`

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1)
opt <- optimize_weights(x, k = 3, n_folds = 3)
opt$best_weights
```

---

pdi *Compute Palimpsest Dissolution Index*

Description

Measures global phase separability as $1 - \bar{H} / \log(K)$. Values close to 1 indicate well-separated phases; values near 0 indicate a compressed palimpsest.

Usage

```
pdi(object)
```

Arguments

`object`	A `sef_fit` object.

Value

A single numeric value between 0 and 1.

See Also

`fit_sef`, `compare_k`

Other diagnostics: `as_phase_table()`, `detect_intrusions()`, `ese()`, `phase_diagnostic_table()`, `predict_phase()`

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
pdi(fit)
```

phase_diagnostic_table

Return a compact diagnostic table

Description

Combines the input data with dominant phase, phase probabilities, entropy, local SEI, and energy in a single data.frame.

Usage

```
phase_diagnostic_table(object)
```

Arguments

object	A sef_fit object.

Value

A data.frame with all input columns plus diagnostics.

See Also

as_phase_table, as_sf_phase

Other diagnostics: as_phase_table(), detect_intrusions(), ese(), pdi(), predict_phase()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
pdt <- phase_diagnostic_table(fit)
names(pdt)
```

phase_transition_matrix

Phase vertical transition matrix

Description

Computes how often finds from phase i are found directly above finds from phase j in the vertical sequence, revealing the stratigraphic ordering of phases.

Usage

```
phase_transition_matrix(object)
```

Arguments

object	A sef_fit object.

Value

A $K \times K$ matrix where entry [i,j] counts transitions from phase i (above) to phase j (below).

See Also

us_summary_table

Other export: export_results(), us_summary_table()

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1)
fit <- fit_sef(x, k = 3)
phase_transition_matrix(fit)
```

---

plot_energy *Plot local Excavation Stratigraphic Energy (base R)*

Description

Plot local Excavation Stratigraphic Energy (base R)

Usage

```
plot_energy(object)
```

Arguments

object	A sef_fit object.

Value

Invisibly returns the object.

See Also

gg_energy for the ggplot2 version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_entropy(), plot_phasefield(), plot_sei_profile()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
plot_energy(fit)
```

plot_entropy *Plot entropy across space (base R)*

Description

Plot entropy across space (base R)

Usage

```
plot_entropy(object)
```

Arguments

object	A sef_fit object.

Value

Invisibly returns the object.

See Also

gg_entropy for the ggplot2/plotly version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_phasefield(), plot_sei_profile()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
plot_entropy(fit)
```

plot_phasefield *Plot dominant phase assignment (base R)*

Description

Plot dominant phase assignment (base R)

Usage

```
plot_phasefield(object)
```

Arguments

object	A sef_fit object.

Value

Invisibly returns the object.

See Also

gg_phasefield for the ggplot2/plotly version.

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_sei_profile()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
plot_phasefield(fit)
```

---

plot_sei_profile *Plot ordered SEI profile (base R)*

Description

Plot ordered SEI profile (base R)

Usage

```
plot_sei_profile(object)
```

Arguments

object	A sef_fit object.

Value

Invisibly returns the object.

See Also

sei_matrix, local_sei

Other plotting: as_plotly(), gg_bootstrap(), gg_compare_k(), gg_confusion(), gg_convergence(), gg_cv(), gg_energy(), gg_entropy(), gg_intrusions(), gg_map(), gg_phase_profile(), gg_phasefield(), gg_weights(), plot_energy(), plot_entropy(), plot_phasefield()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
plot_sei_profile(fit)
```

predict_phase *Predict phase probabilities*

Description

Convenience alias for as_phase_table.

Usage

```
predict_phase(object)
```

Arguments

object	A sef_fit object.

Value

A data.frame with probabilities and diagnostics.

See Also

as_phase_table

Other diagnostics: as_phase_table(), detect_intrusions(), ese(), pdi(), phase_diagnostic_table()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
head(predict_phase(fit))
```

read_db *Read archaeological data from a SQL database*

Description

Loads find data from any DBI-compatible database and maps columns to the format expected by fit_sef.

Usage

```
read_db(
  conn,
  query = NULL,
  table = NULL,
  col_id = "id",
  col_x = "x",
  col_y = "y",
  col_z = "z",
  col_context = "context",
  col_date_min = "date_min",
  col_date_max = "date_max",
  col_class = "class",
  col_taf = NULL,
  schema = "custom",
  sito = NULL
)
```

Arguments

conn	A DBIConnection object.
query	SQL query returning data with the required columns.
table	Table name (alternative to query).
col_id, col_x, col_y, col_z, col_context, col_date_min, col_date_max, col_class, col_taf	Column name mappings.
schema	Use "pyarchinit" for automatic PyArchInit mapping.
sito	Site filter for PyArchInit schema.

Value

A data.frame ready for fit_sef.

See Also

fit_sef, load_geometries

Other data-import: load_geometries()

---

read_harris *Read Harris Matrix from CSV edge list*

Description

Reads a CSV file with columns from, to, and optionally weight, and converts it to an $n \times n$ penalty matrix aligned with the find-level data.

Usage

```
read_harris(file, contexts, default_weight = 1)
```

Arguments

file	Path to CSV with columns from, to, and optionally weight.
contexts	Character vector of context labels for each find (length = number of finds).
default_weight	Weight for edges without an explicit weight.

Value

An $n \times n$ penalty matrix.

See Also

harris_from_contexts, fit_sef

Other harris: harris_from_contexts(), validate_phases_harris()

---

reorder_phases *Reorder phases by mean depth*

Description

Relabels phases so that phase 1 corresponds to the deepest (oldest) stratum and phase K to the shallowest (most recent). This ensures consistent, interpretable phase numbering across different runs.

Usage

```
reorder_phases(object)
```

Arguments

object	A sef_fit object.

Value

A sef_fit object with reordered phases.

See Also

fit_sef

Other fitting: compare_k(), fit_sef(), sef_summary()

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1)
fit <- fit_sef(x, k = 3)
fit <- reorder_phases(fit)
```

report_sef *Generate a textual interpretive report for a SEF model*

Description

Produces a structured Markdown report covering phase composition, intrusion detection, stratigraphic unit purity, and recommendations. Available in English and Italian.

Usage

```
report_sef(object, lang = c("en", "it"), file = NULL)
```

Arguments

object	A sef_fit object.
lang	Language: "en" (English) or "it" (Italian).
file	Optional file path to save the report.

Value

Character string with the report text (invisibly).

See Also

fit_sef, sef_summary

Examples

```
x <- archaeo_sim(n = 100, k = 3, seed = 1)
fit <- fit_sef(x, k = 3)
report_sef(fit, lang = "en")
```

sef_summary *Compact summary for a fitted SEF model*

Description

Returns a named list with global diagnostics and phase counts.

Usage

```
sef_summary(object)
```

Arguments

object	A sef_fit object.

Value

A named list.

See Also

fit_sef, pdi

Other fitting: compare_k(), fit_sef(), reorder_phases()

Examples

```
x <- archaeo_sim(n = 60, k = 2, seed = 1)
fit <- fit_sef(x, k = 2)
sef_summary(fit)
```

---

sei_matrix *Compute the Stratigraphic Entanglement Index matrix*

Description

Builds an $n \times n$ symmetric matrix quantifying pairwise depositional coherence from spatial, vertical, temporal, and cultural evidence.

Usage

```
sei_matrix(
  data,
  coords = c("x", "y", "z"),
  chrono = c("date_min", "date_max"),
  class_col = "class",
  weights = c(ws = 1, wz = 1, wt = 1, wc = 1),
  eps = 1e-09,
  z_floor = 0.25,
  max_dist = NULL
)
```

Arguments

data	Input data.frame.
coords	Character vector of coordinate column names (x, y, z).
chrono	Character vector with minimum and maximum dating columns.
class_col	Class column name.
weights	Named numeric vector with components ws, wz, wt, wc. Each component is normalised to $[0, 1]$ before weighting, so the weights represent relative importance.
eps	Small value to avoid division by zero in spatial distance.
z_floor	Minimum vertical denominator.

`max_dist` Maximum spatial distance for pair inclusion. When `NULL` (default), all pairs are computed. For large datasets (n > 2000), setting this to a reasonable neighbourhood radius dramatically reduces memory and computation time. The result is a sparse-like matrix with zeros for distant pairs.

Value

A symmetric numeric matrix with zero diagonal.

Note

SEI values are normalised within each dataset. Absolute SEI values are **not directly comparable** across different excavations or datasets of different sizes. Use SEI for within-dataset ranking and relative comparisons only.

See Also

`local_sei`, `fit_sef`

Other SEI: `local_sei()`, `sei_sparse()`

Examples

```
x <- archaeo_sim(n = 30, k = 2, seed = 1)
S <- sei_matrix(x)
dim(S)
```

---

sei_sparse *Compute SEI matrix with automatic sparsification*

Description

Wrapper around `sei_matrix` that automatically sets `max_dist` when the dataset exceeds a size threshold, using the 25th percentile of pairwise distances as the cutoff.

Usage

```
sei_sparse(
  data,
  coords = c("x", "y", "z"),
  chrono = c("date_min", "date_max"),
  class_col = "class",
  weights = c(ws = 1, wz = 1, wt = 1, wc = 1),
  eps = 1e-09,
  z_floor = 0.25,
  n_threshold = 1000
)
```

Arguments

data	Input data.frame.
coords	Character vector of coordinate column names (x, y, z).
chrono	Character vector with minimum and maximum dating columns.
class_col	Class column name.
weights	Named numeric vector with components ws, wz, wt, wc. Each component is normalised to $[0, 1]$ before weighting, so the weights represent relative importance.
eps	Small value to avoid division by zero in spatial distance.
z_floor	Minimum vertical denominator.
n_threshold	Datasets larger than this trigger sparse mode (default: 1000).

Value

A numeric matrix (same as sei_matrix).

See Also

sei_matrix

Other SEI: local_sei(), sei_matrix()

Examples

```
x <- archaeo_sim(n = 50, k = 2, seed = 1)
S <- sei_sparse(x)
```

---

summary.sef_fit *Summarise a fitted SEF model*

Description

Summarise a fitted SEF model

Usage

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

Arguments

object	A sef_fit object.
...	Ignored.

Value

A named list.

us_summary_table *Summary table per stratigraphic unit*

Description

Aggregates finds by context (US), reporting the dominant phase, purity (proportion of finds in dominant phase), mean entropy, mean energy, and intrusion count.

Usage

```
us_summary_table(object)
```

Arguments

object	A sef_fit object.

Value

A data.frame with one row per stratigraphic unit.

See Also

export_results, phase_diagnostic_table

Other export: export_results(), phase_transition_matrix()

Examples

```
x <- archaeo_sim(n = 80, k = 3, seed = 1)
fit <- fit_sef(x, k = 3, context = "context")
us_summary_table(fit)
```

validate_phases_harris *Validate phase assignments against stratigraphic ordering*

Description

Checks whether the estimated phases follow the expected vertical ordering within each stratigraphic unit pair.

Usage

```
validate_phases_harris(object)
```

Arguments

object	A sef_fit object.

Value

A data.frame with one row per context pair, indicating whether the dominant phase ordering is consistent with depth.

See Also

harris_from_contexts, us_summary_table

Other harris: harris_from_contexts(), read_harris()

Examples

```
x <- archaeo_sim(n = 60, k = 3, seed = 1)
fit <- fit_sef(x, k = 3, context = "context")
validate_phases_harris(fit)
```

---

villa_romana *Simulated Roman Villa excavation dataset*

Description

A realistic archaeological dataset representing 300 finds from a multi-period Roman villa with 4 occupation phases: Republican (2nd–1st c. BCE), Early Imperial (1st–2nd c. CE), Late Imperial (3rd–4th c. CE), and Late Antique (5th–6th c. CE).

Usage

```
villa_romana
```

Format

A data.frame with 300 rows and 10 variables:

id Unique find identifier (VR_0001 to VR_0300)

x Easting coordinate (metres)

y Northing coordinate (metres)

z Depth below datum (metres)

context Stratigraphic unit label (US_101 to US_404)

date_min Start of chronological interval (BCE as negative, CE as positive)

date_max End of chronological interval

class Material class (ceramic types, bone, metal, glass, etc.)

taf_score Taphonomic disturbance score (0 = pristine, 1 = fully disturbed)

true_phase Known depositional phase (1–4, for validation)

Details

The dataset includes realistic post-depositional disturbances: bioturbation (8\ stratigraphic intrusions), and residual pottery (3\ in younger contexts).

Source

Simulated data; see `data-raw/site_villa_romana.R`.

Examples

```
data(villa_romana)
str(villa_romana)
table(villa_romana$true_phase)
```

Index