Package 'SelectBoost.FDA'

April 10, 2026

Title SelectBoost-Style Variable Selection for Functional Data Analysis

Date 2026-04-02

Version 0.5.0

Author Frederic Bertrand [cre, aut] (ORCID: <https://orcid.org/0000-0002-0837-8281>)

Maintainer Frederic Bertrand <frederic.bertrand@lecnam.net>

Description Implements 'SelectBoost'-style variable selection workflows for functional data analysis. The package provides FDA-native design and preprocessing objects for raw curves, spline-basis expansions, Functional principal component analysis scores, and scalar covariates; grouped stability-selection routines based on repeated subject-level subsampling; multiple selector backends including lasso, group lasso, and sparse-group lasso; FDA-aware grouping functions and calibration helpers for 'SelectBoost'; method-comparison utilities; a formula interface; simulation, benchmarking, and validation helpers with mapped ground truth; targeted sensitivity-study utilities and shipped benchmark summaries for mean 'F1' comparisons between FDA-aware and plain 'SelectBoost' workflows; small example datasets; and an optional adapter to the native stability-selection interface from the 'FDboost' package.

LazyData true

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 2.10)

Imports SelectBoost, stats

Suggests FDboost, glmnet, grpreg, knitr, pkgdown, rmarkdown, SGL, stabs, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website pkgdown

URL https://fbertran.github.io/SelectBoost.FDA/,
https://github.com/fbertran/SelectBoost.FDA

BugReports https://github.com/fbertran/SelectBoost.FDA/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2026-04-10 13:30:09 UTC

Contents

`apply_fda_preprocessor` *Apply an FDA Preprocessor*

Description

Applies a fitted preprocessor to new functional predictors and optional scalar covariates, returning an `fda_matrix` object compatible with the selection routines.

Usage

```
apply_fda_preprocessor(object, predictors, scalar_covariates = NULL, ...)
```

Arguments

`object`	A fitted `fda_preprocessor`.
`predictors`	New functional predictors.
`scalar_covariates`	Optional scalar covariates.
`...`	Not used.

Value

An object of class `fda_matrix`.

`as_functional_matrix` *Flatten Functional Predictors Into a Matrix*

Description

Accepts a standard numeric matrix/data frame or a named list of functional blocks. List inputs are column-bound while preserving the original block membership of each coefficient, which is later reused for grouped stability selection and FDA-aware SelectBoost grouping.

Usage

```
as_functional_matrix(x, center = FALSE, scale = FALSE)
```

Arguments

x	A numeric matrix/data frame, an fda_grid, an fda_basis, an fda_design, or a list of such objects. Each list element is treated as one functional block.
center, scale	Passed to base::scale() when either argument is TRUE.

Value

An object of class fda_matrix with elements x, blocks, and positions.

---

benchmark_selection_methods *Benchmark FDA Selection Methods on Shared Ground Truth*

Description

Runs compare_selection_methods() on a simulated dataset and evaluates the fitted objects against the mapped truth.

Usage

```
benchmark_selection_methods(
  data,
  methods = c("stability", "interval", "selectboost", "plain_selectboost"),
  levels = c("feature", "group"),
  stability_args = list(),
  interval_args = list(),
  selectboost_args = list(),
  plain_selectboost_args = list(),
  fdboost_model = NULL,
  fdboost_args = list(),
  keep_comparison = TRUE
)
```

Arguments

data	An object returned by simulate_fda_scenario().
methods	Methods passed to compare_selection_methods().
levels	Evaluation levels.
stability_args, interval_args, selectboost_args, plain_selectboost_args	Additional arguments passed to compare_selection_methods().
fdboost_model, fdboost_args	Optional FDboost inputs forwarded to compare_selection_methods().
keep_comparison	Should the fitted comparison object be stored?

Value

An object of class `fda_benchmark`.

Examples

```
sim <- simulate_fda_scenario(n = 24, grid_length = 16, seed = 1)
bench <- benchmark_selection_methods(
  sim,
  methods = c("selectboost", "plain_selectboost"),
  selectboost_args = list(B = 3, steps.seq = 0.5, c0lim = FALSE),
  plain_selectboost_args = list(B = 3, steps.seq = 0.5, c0lim = FALSE)
)
head(bench$metrics)
```

`calibrate_interval_width` *Calibrate Interval Widths*

Description

Runs interval stability selection over candidate interval widths.

Usage

```
calibrate_interval_width(
  design,
  widths,
  step = NULL,
  overlap = FALSE,
  selector = "lasso",
  keep_fits = FALSE,
  seed = NULL,
  ...
)
```

Arguments

`design`	An `fda_design` object.
`widths`	Candidate interval widths.
`step`	Optional step size. Defaults to `widths`.
`overlap`	Should the interval groups overlap?
`selector`	Base selector passed to `interval_stability_selection()`.
`keep_fits`	Should the fitted objects be stored in the result?
`seed`	Optional seed used to create deterministic per-grid seeds.
`...`	Additional arguments passed to `interval_stability_selection()`.

Value

An object of class `fda_calibration_grid`.

`calibrate_selectboost` *Calibrate SelectBoost c0 Values*

Description

Runs FDA-SelectBoost on a user-provided or suggested `c0` grid.

Usage

```
calibrate_selectboost(
  design,
  selector = "msgps",
  c0_grid = NULL,
  grid_method = c("quantile", "linear"),
  association_method = c("correlation", "neighborhood", "hybrid", "interval"),
  keep_fit = TRUE,
  ...
)
```

Arguments

`design`	An `fda_design` object.
`selector`	Base selector passed to `fit_selectboost()`.
`c0_grid`	Optional explicit `c0` grid.
`grid_method`	Rule used by `suggest_c0_grid()` when `c0_grid` is omitted.
`association_method`	Passed to `suggest_c0_grid()` and `fit_selectboost()`.
`keep_fit`	Should the fitted object be stored in the result?
`...`	Additional arguments passed to `fit_selectboost()`.

Value

An object of class `fda_calibration_grid`.

`calibrate_stability_selection` *Calibrate Stability-Selection Parameters*

Description

Runs grouped stability selection over a grid of subsampling fractions and cutoff values.

Usage

```
calibrate_stability_selection(
  design,
  selector = "group_lasso",
  sample_fraction_grid = c(0.5, 0.632, 0.75),
  cutoff_grid = c(0.6, 0.75, 0.9),
  keep_fits = FALSE,
  seed = NULL,
  ...
)
```

Arguments

`design`	An `fda_design` object.
`selector`	Base selector passed to `fit_stability()`.
`sample_fraction_grid`	Candidate subsampling fractions.
`cutoff_grid`	Candidate cutoff values.
`keep_fits`	Should the fitted objects be stored in the result?
`seed`	Optional seed used to create deterministic per-grid seeds.
`...`	Additional arguments passed to `fit_stability()`.

Value

An object of class `fda_calibration_grid`.

`compare_selection_methods` *Compare FDA Selection Methods*

Description

Runs multiple selection workflows on the same `fda_design` object and returns both the fitted objects and a comparison table.

Usage

```
compare_selection_methods(
  design,
  methods = c("stability", "interval", "selectboost"),
  stability_args = list(),
  interval_args = list(),
  selectboost_args = list(),
  plain_selectboost_args = list(),
  fdboost_model = NULL,
  fdboost_args = list()
)
```

Arguments

design	An fda_design object.
methods	Methods to run. Supported values are "stability", "interval", "selectboost", "plain_selectboost", and "fdboost".
stability_args, interval_args, selectboost_args, plain_selectboost_args	Named lists of arguments passed to the corresponding fitting functions.
fdboost_model	Optional fitted FDboost object used when methods includes "fdboost".
fdboost_args	Additional arguments passed to fdboost_stability_selection().

Value

An object of class fda_method_comparison.

Examples

```
sim <- simulate_fda_scenario(n = 24, grid_length = 16, seed = 1)
comparison <- compare_selection_methods(
  sim$design,
  methods = c("selectboost", "plain_selectboost"),
  selectboost_args = list(B = 3, steps.seq = 0.5, c0lim = FALSE),
  plain_selectboost_args = list(B = 3, steps.seq = 0.5, c0lim = FALSE)
)
summary(comparison)
```

---

evaluate_selection *Evaluate Selection Recovery Against Ground Truth*

Description

Computes support-recovery metrics for fitted selection objects against the truth generated by simulate_fda_scenario().

Usage

```
evaluate_selection(x, truth, level = c("feature", "group", "basis"), ...)
```

Arguments

x	A fitted selection object or an fda_method_comparison.
truth	Ground-truth object, typically the value returned by simulate_fda_scenario().
level	Evaluation level: "feature", "group", or "basis".
...	Additional arguments passed to the relevant method.

Value

A data frame with recovery metrics.

---

fda_basis *Basis-Expanded Functional Predictor*

Description

Constructor for a functional predictor represented by basis coefficients or FPCA scores.

Usage

```
fda_basis(
  coefficients,
  basis_type = c("generic", "spline", "wavelet", "fpca"),
  argvals = NULL,
  component_names = NULL,
  name = NULL,
  unit = NULL
)
```

Arguments

coefficients	Numeric matrix with one row per observation.
basis_type	Label describing the representation.
argvals	Optional labels or positions for basis functions/components.
component_names	Optional names for coefficient columns.
name	Optional predictor name.
unit	Optional unit for the basis domain.

Value

An object of class fda_basis.

fda_bspline *Spline-Basis Preprocessing Spec*

Description

Spline-Basis Preprocessing Spec

Usage

```
fda_bspline(
  df = 6L,
  degree = 3L,
  intercept = TRUE,
  center = FALSE,
  scale = FALSE
)
```

Arguments

df	Degrees of freedom used by splines::bs().
degree	Spline degree.
intercept	Should the spline basis include an intercept column?
center, scale	Logical flags controlling column-wise centering and scaling of the resulting coefficients.

Value

An object of class fda_preprocess_spec.

fda_design *Functional Design Object*

Description

Bundles the response, functional predictors, family, and a reversible feature map. This is the FDA-native entry point for the higher-level fitting functions.

Usage

```
fda_design(
  response = NULL,
  predictors,
  scalar_covariates = NULL,
  family = c("gaussian", "binomial"),
  id = NULL,
```

```
  center = FALSE,
  scale = FALSE,
  transforms = NULL,
  scalar_transform = NULL,
  preprocessor = NULL
)
```

Arguments

response	Response vector.
predictors	A single predictor or a named list of predictors. Elements can be fda_grid, fda_basis, fda_scalar, matrices, data frames, or numeric vectors.
scalar_covariates	Optional scalar covariates supplied separately from the functional predictors.
family	Model family.
id	Optional observation identifiers.
center, scale	Backward-compatible shortcuts for applying an identity transform with centering and scaling to the functional predictors.
transforms	Optional preprocessing specs for the functional predictors.
scalar_transform	Optional preprocessing specs for scalar covariates.
preprocessor	Optional fitted fda_preprocessor. When supplied, it is reused instead of fitting preprocessing from the current data.

Value

An object of class fda_design.

Examples

```
data("spectra_example", package = "SelectBoost.FDA")
idx <- 1:20
design <- fda_design(
  response = spectra_example$response[idx],
  predictors = list(
    signal = fda_grid(
      spectra_example$predictors$signal[idx, ],
      argvals = spectra_example$grid,
      name = "signal"
    ),
    nuisance = fda_grid(
      spectra_example$predictors$nuisance[idx, ],
      argvals = spectra_example$grid,
      name = "nuisance"
    )
  ),
  scalar_covariates = spectra_example$scalar_covariates[idx, ],
  scalar_transform = fda_standardize(),
  family = "gaussian"
```

```
)
summary(design)
```

`fda_design_formula` *Build an FDA Design from a Formula*

Description

Supports additive formulas of the form `y ~ signal + noise + age + batch`, where functional terms are supplied as matrices, `fda_grid`, or `fda_basis` objects in `data`, and scalar terms are expanded through `stats::model.matrix()`.

Usage

```
fda_design_formula(
  formula,
  data,
  family = c("gaussian", "binomial"),
  transforms = NULL,
  scalar_transform = NULL,
  preprocessor = NULL,
  center = FALSE,
  scale = FALSE
)
```

Arguments

`formula`	An additive formula with a single response.
`data`	A list or data frame containing the variables referenced in `formula`.
`family`, `transforms`, `scalar_transform`, `preprocessor`, `center`, `scale`	Passed to `fda_design()`.

Value

An object of class `fda_design`.

fda_fpca *FPCA Preprocessing Spec*

Description

FPCA Preprocessing Spec

Usage

```
fda_fpca(
  n_components = 3L,
  variance_explained = NULL,
  center = TRUE,
  scale = FALSE
)
```

Arguments

`n_components`	Number of principal components to retain.
`variance_explained`	Optional cumulative explained variance target in `(0, 1]`. When supplied, it overrides `n_components`.
`center, scale`	Passed to `stats::prcomp()`.

Value

An object of class `fda_preprocess_spec`.

fda_grid *Functional Predictor on a Common Grid*

Description

Constructor for one discretized functional predictor sampled on a common grid.

Usage

```
fda_grid(values, argvals = NULL, name = NULL, unit = NULL)
```

Arguments

`values`	Numeric matrix with one row per observation.
`argvals`	Optional vector of grid values. Defaults to `seq_len(ncol(values))`.
`name`	Optional predictor name.
`unit`	Optional unit for the grid axis.

Value

An object of class `fda_grid`.

`fda_identity` *Identity Preprocessing Spec*

Description

Identity Preprocessing Spec

Usage

```
fda_identity(center = FALSE, scale = FALSE)
```

Arguments

`center`, `scale`	Logical flags controlling column-wise centering and scaling of the transformed features.

Value

An object of class `fda_preprocess_spec`.

`fda_scalar` *Scalar Predictor Constructor*

Description

Wraps scalar covariates so they can participate in the same feature-mapping and preprocessing machinery as functional predictors.

Usage

```
fda_scalar(values, name = NULL, unit = NULL)
```

Arguments

`values`	Numeric vector or matrix with one row per observation.
`name`	Optional predictor name.
`unit`	Optional unit label.

Value

An object of class `fda_scalar`.

fda_standardize *Standardization Preprocessing Spec*

Description

Standardization Preprocessing Spec

Usage

```
fda_standardize(center = TRUE, scale = TRUE)
```

Arguments

center, scale	Logical flags controlling column-wise centering and scaling. Both default to TRUE.

Value

An object of class fda_preprocess_spec.

fdboost_stability_selection *Stability Selection for FDboost Fits*

Description

Thin adapter to the stabsel.FDboost() method. This is the native route when the model itself is already fitted with FDboost.

Usage

```
fdboost_stability_selection(model, ...)
```

Arguments

model	A fitted FDboost object.
...	Additional arguments forwarded to stabs::stabsel().

Value

A stabsel object.

fit_fda_preprocessor *Fit an FDA Preprocessor*

Description

Learns train/test-safe preprocessing transforms for functional predictors and optional scalar covariates. The fitted object can be reused to create compatible `fda_design` objects on new data.

Usage

```
fit_fda_preprocessor(
  predictors,
  scalar_covariates = NULL,
  transforms = NULL,
  scalar_transform = NULL
)
```

Arguments

`predictors`	One predictor or a named list of predictors.
`scalar_covariates`	Optional scalar covariates supplied as a vector, matrix/data frame, `fda_scalar`, or a named list.
`transforms`	Optional preprocessing specs for functional predictors.
`scalar_transform`	Optional preprocessing specs for scalar covariates.

Value

An object of class `fda_preprocessor`.

fit_selectboost *Fit SelectBoost from an FDA Design*

Description

Fit SelectBoost from an FDA Design

Usage

```
fit_selectboost(design, ...)
```

Arguments

`design`	An `fda_design` object.
`...`	Additional arguments forwarded to `selectboost_fda()`.

Value

An object inheriting from `selectboost_fda_result`.

Examples

```
sim <- simulate_fda_scenario(n = 24, grid_length = 16, seed = 1)
fit <- fit_selectboost(
  sim$design,
  mode = "fast",
  steps.seq = 0.5,
  c0lim = FALSE,
  B = 3
)
head(selection_map(fit, c0 = colnames(fit$feature_selection)[1]))
```

`fit_selectboost_formula` *Fit FDA-SelectBoost from a Formula*

Description

Fit FDA-SelectBoost from a Formula

Usage

```
fit_selectboost_formula(
  formula,
  data,
  family = c("gaussian", "binomial"),
  transforms = NULL,
  scalar_transform = NULL,
  preprocessor = NULL,
  center = FALSE,
  scale = FALSE,
  ...
)
```

Arguments

`formula`, `data`, `family`, `transforms`, `scalar_transform`, `preprocessor`, `center`, `scale`	Passed to `fda_design_formula()`.
`...`	Additional arguments passed to `fit_selectboost()`.

Value

An object inheriting from `selectboost_fda_result`.

fit_stability *Fit Grouped Stability Selection from an FDA Design*

Description

Fit Grouped Stability Selection from an FDA Design

Usage

```
fit_stability(design, ...)
```

Arguments

design	An fda_design object.
...	Additional arguments forwarded to stability_selection_fda().

Value

An object inheriting from fda_stability_selection.

Examples

```
sim <- simulate_fda_scenario(n = 24, grid_length = 16, seed = 1)
if (requireNamespace("glmnet", quietly = TRUE)) {
  fit <- fit_stability(
    sim$design,
    selector = "lasso",
    B = 4,
    cutoff = 0.4,
    seed = 1
  )
  head(selection_map(fit))
}
```

fit_stability_formula *Fit Stability Selection from a Formula*

Description

Fit Stability Selection from a Formula

Usage

```
fit_stability_formula(
  formula,
  data,
  family = c("gaussian", "binomial"),
  transforms = NULL,
  scalar_transform = NULL,
  preprocessor = NULL,
  center = FALSE,
  scale = FALSE,
  ...
)
```

Arguments

formula, data, family, transforms, scalar_transform, preprocessor, center, scale	Passed to fda_design_formula().
...	Additional arguments passed to fit_stability().

Value

An object inheriting from fda_stability_selection.

---

functional_association

Functional Association Matrix

Description

Computes or post-processes an absolute association matrix for discretized or basis-expanded functional predictors.

Usage

```
functional_association(
  x,
  association = NULL,
  method = c("correlation", "neighborhood", "hybrid", "interval"),
  within_blocks = TRUE,
  bandwidth = NULL,
  interval_groups = NULL,
  width = NULL,
  step = width,
  decay = 1
)
```

Arguments

x	Any input accepted by as_functional_matrix().
association	Optional square association matrix supplied by the user. When omitted, abs(stats::cor(X)) is used.
method	Association structure. "correlation" uses the absolute correlation matrix, "neighborhood" uses local positional similarity, "hybrid" multiplies correlation by a neighborhood kernel, and "interval" induces associations within interval groups.
within_blocks	Should cross-block associations be zeroed out?
bandwidth	Optional maximum within-block lag retained in the association matrix.
interval_groups	Optional interval grouping used when method = "interval".
width, step	Interval parameters used when method = "interval" and interval_groups is omitted.
decay	Positive exponent controlling the neighborhood kernel.

Value

A square absolute association matrix with unit diagonal.

---

functional_block_groups

Block-Level Groups for Functional Predictors

Description

Returns one group label per column, with each functional block defining a group.

Usage

```
functional_block_groups(x)
```

Arguments

x	Any input accepted by as_functional_matrix().

Value

An integer vector of group memberships.

functional_interval_groups

Interval Groups for Discretized Functional Predictors

Description

Creates non-overlapping interval groups within each functional block. This is useful when one wants region-level stability summaries instead of pointwise selection frequencies.

Usage

```
functional_interval_groups(x, width, step = width, overlap = FALSE)
```

Arguments

x	Any input accepted by as_functional_matrix().
width	Positive integer interval width within each block.
step	Step size between interval starts. Only non-overlapping intervals are supported by default.
overlap	Logical; should intervals be allowed to overlap? When TRUE, the result is returned as an overlapping group structure.

Value

Either an integer group vector with an interval_table attribute or an overlapping group structure of class fda_group_list.

interval_stability_selection

Interval Stability Selection

Description

Convenience wrapper around stability_selection_fda() that first creates non-overlapping interval groups within each functional block.

Usage

```
interval_stability_selection(
  x,
  y = NULL,
  width,
  step = width,
  overlap = FALSE,
  ...
)
```

Arguments

x	Any input accepted by as_functional_matrix(), or an fda_design object.
y	Response vector. Leave as NULL when x is an fda_design.
width	Positive interval width.
step	Step size between interval starts.
overlap	Logical; should the interval groups overlap?
...	Additional arguments forwarded to stability_selection_fda().

Value

An object of class fda_interval_stability_selection.

---

make_functional_grouping_function

FDA-Aware Grouping Function for SelectBoost

Description

Builds a closure that can be passed directly to group= in SelectBoost::fastboost() or SelectBoost::autoboost(). The returned grouping function respects functional block boundaries and can optionally restrict groups to local neighborhoods along the observation grid.

Usage

```
make_functional_grouping_function(
  x,
  association = NULL,
  method = c("threshold", "community"),
  association_method = c("correlation", "neighborhood", "hybrid", "interval"),
  within_blocks = TRUE,
  bandwidth = NULL,
  interval_groups = NULL,
  width = NULL,
  step = width,
  decay = 1
)
```

Arguments

x	Any input accepted by as_functional_matrix().
association	Optional square association matrix. When omitted, the correlation matrix supplied by SelectBoost is reused after applying the FDA-specific masks.
method	Grouping strategy. "threshold" wraps SelectBoost::group_func_1() and "community" wraps SelectBoost::group_func_2().

`association_method`
: Association structure passed to `functional_association()`.

`within_blocks`
: Should groups be restricted to features coming from the same functional block?

`bandwidth`
: Optional maximum within-block lag retained in groups.

`interval_groups`, `width`, `step`, `decay`
: Additional arguments passed to `functional_association()` when using region-aware associations.

Value

A function with signature `(absXcor, c0)` compatible with `SelectBoost`.

---

motion_example *Smooth Trajectory Functional Example*

Description

Simulated smooth trajectories used to demonstrate spline-basis and FPCA preprocessing from raw curves.

Usage

```
motion_example
```

Format

A list with four components:

grid Numeric vector of observation times.

response Numeric response vector.

predictors Named list of functional predictor matrices.

scalar_covariates Data frame with scalar covariates.

Source

Simulated for package examples.

plain_selectboost *Plain SelectBoost Baseline for Functional Predictors*

Description

Runs SelectBoost directly on the flattened predictor matrix without the FDA-specific grouping heuristics used by selectboost_fda(). This is useful as a benchmark against the FDA-aware variant.

Usage

```
plain_selectboost(
  x,
  y = NULL,
  mode = c("fast", "auto"),
  selector = "msgps",
  selector_fun = NULL,
  selector_args = list(),
  groups = NULL,
  family = c("gaussian", "binomial"),
  association = NULL,
  group_method = c("threshold", "community"),
  ...
)
```

Arguments

x	Any input accepted by as_functional_matrix(), or an fda_design object.
y	Response vector. Leave as NULL when x is an fda_design.
mode	"fast" for a fixed c0 grid or "auto" for the adaptive version.
selector	Base selector used inside SelectBoost. Choose from "msgps", "lasso", "group_lasso", "sparse_group_lasso", the backend-specific aliases "glmnet", "grpreg", "sgl", or provide a custom function.
selector_fun	Optional custom base selector. It must return a coefficient vector of length p.
selector_args	Optional named list of arguments forwarded to the base selector.
groups	Optional feature groups used by grouped base selectors such as "grpreg". Defaults to block-level groups for list inputs.
family	Model family passed to built-in selectors.
association	Optional absolute association matrix used directly by the raw SelectBoost grouping function.
group_method	Functional grouping backend: threshold-based or community-based.
...	Additional arguments passed to SelectBoost::fastboost() or SelectBoost::autoboost().

Value

An object of class plain_selectboost_result.

plot.fda_selection *Plot FDA Selection Results*

Description

Plots feature-, group-, interval-, and basis-level summaries derived from `selection_map()`. The available views depend on the fitted object:

Usage

```
## S3 method for class 'fda_stability_selection'
plot(
  x,
  type = c("feature", "group", "interval", "basis"),
  value = c("group", "mean", "max"),
  facet = c("none", "predictor"),
  palette = selection_palette(),
  show_legend = TRUE,
  legend_title = NULL,
  legend_n_ticks = 3L,
  legend_digits = 2L,
  legend_cex = 0.75,
  cutoff = x$cutoff,
  ...
)

## S3 method for class 'selectboost_fda_result'
plot(
  x,
  type = c("feature", "group", "basis"),
  value = c("max", "mean"),
  palette = selection_palette(),
  show_legend = TRUE,
  legend_title = NULL,
  legend_n_ticks = 3L,
  legend_digits = 2L,
  legend_cex = 0.75,
  c0 = NULL,
  ...
)
```

Arguments

x	An object returned by `stability_selection_fda()`, `interval_stability_selection()`, `fit_stability()`, `selectboost_fda()`, or `fit_selectboost()`.

type	Summary view to plot. Stability-selection fits support "feature", "group", "interval", and "basis". SelectBoost fits support "feature", "group", and "basis".
value	Quantity summarized in group, interval, and basis views. Stability-selection fits accept "group", "mean", and "max". SelectBoost fits accept "mean" and "max".
facet	Faceting mode for interval heatmaps. Currently only type = "interval" uses this argument.
palette	Vector of colors used for heatmaps.
show_legend	Logical; should heatmap views draw a legend?
legend_title	Optional legend title for heatmap views. By default an informative title is chosen from type and value.
legend_n_ticks	Approximate number of tick marks used in the heatmap legend.
legend_digits	Number of significant digits used for heatmap legend labels.
legend_cex	Character expansion used for heatmap legend text.
cutoff	Stability threshold. Only used for fda_stability_selection objects.
...	Additional graphical parameters passed to bar-plot-based views.
c0	Optional SelectBoost correlation threshold. When omitted, SelectBoost heatmaps are drawn across all available c0 values.

Details

- fda_stability_selection supports type = "feature", "group", "interval", and "basis".
- selectboost_fda_result supports type = "feature", "group", and "basis".

Heatmap-based views are used for interval summaries and for SelectBoost summaries over multiple c0 values. Bar-plot views are used otherwise.

Value

Invisibly returns the helper output used to build the plot.

See Also

selection_map()

---

run_selectboost_sensitivity_study

Run a Targeted Sensitivity Study for FDA-SelectBoost

Description

Repeats the FDA benchmark over a grid of simulation settings and a grid of FDA-aware SelectBoost settings. This is intended to answer the specific benchmark question of when selectboost_fda() improves on plain SelectBoost.

Usage

```
run_selectboost_sensitivity_study(
  n_rep = 10L,
  simulate_grid = expand.grid(scenario = c("localized_dense", "confounded_blocks"),
    confounding_strength = c(0.4, 0.9), active_region_scale = c(1, 0.7),
    local_correlation = c(0, 2), stringsAsFactors = FALSE),
  selectboost_grid = expand.grid(association_method = c("correlation", "neighborhood",
    "hybrid", "interval"), bandwidth = c(NA, 4, 8), stringsAsFactors = FALSE),
  simulate_args = list(),
  benchmark_args = list(),
  seed = NULL,
  keep_results = FALSE
)
```

Arguments

n_rep	Number of replications per setting combination.
simulate_grid	Data frame of simulation-setting combinations. Columns are merged into simulate_args and can include scenario, confounding_strength, active_region_scale, and local_correlation.
selectboost_grid	Data frame of selectboost_fda() setting combinations. Columns are merged into benchmark_args$selectboost_args and can include association_method, bandwidth, width, or step.
simulate_args	Named list forwarded to simulate_fda_scenario().
benchmark_args	Named list forwarded to benchmark_selection_methods(). When omitted, the study compares FDA-aware SelectBoost, plain SelectBoost, and grouped stability selection.
seed	Optional seed used to derive deterministic per-replication and per-setting seeds.
keep_results	Should the individual benchmark objects be returned?

Value

An object inheriting from fda_selectboost_sensitivity_study and fda_simulation_study.

Examples

```
grid <- data.frame(
  scenario = "confounded_blocks",
  confounding_strength = 0.9,
  active_region_scale = 0.7,
  local_correlation = 2,
  stringsAsFactors = FALSE
)
methods <- data.frame(
  association_method = c("correlation", "hybrid"),
  bandwidth = c(NA, 4),
  stringsAsFactors = FALSE
```

```
)
study <- run_selectboost_sensitivity_study(
  n_rep = 1,
  simulate_grid = grid,
  selectboost_grid = methods,
  simulate_args = list(n = 24, grid_length = 16),
  benchmark_args = list(
    methods = c("selectboost", "plain_selectboost"),
    levels = "feature",
    selectboost_args = list(B = 3, steps.seq = 0.5, c0lim = FALSE),
    plain_selectboost_args = list(B = 3, steps.seq = 0.5, c0lim = FALSE)
  ),
  seed = 1
)
summarise_benchmark_advantage(
  study,
  target = "selectboost",
  reference = "plain_selectboost",
  level = "feature"
)
```

`run_simulation_study` *Run a Repeated FDA Simulation Study*

Description

Repeats `simulate_fda_scenario()` and `benchmark_selection_methods()` over multiple replications and aggregates the resulting recovery metrics.

Usage

```
run_simulation_study(
  n_rep = 10L,
  simulate_args = list(),
  benchmark_args = list(),
  seed = NULL,
  keep_results = FALSE
)
```

Arguments

`n_rep`	Number of simulation replications.
`simulate_args`	Named list forwarded to `simulate_fda_scenario()`.
`benchmark_args`	Named list forwarded to `benchmark_selection_methods()`.
`seed`	Optional seed used to derive deterministic per-replication seeds.
`keep_results`	Should the individual benchmark objects be returned?

Value

An object of class `fda_simulation_study`.

`selectboost_fda` *FDA-Oriented SelectBoost Wrapper*

Description

Wraps `SelectBoost::fastboost()` or `SelectBoost::autoboost()` while adding FDA-specific structure through block-aware and region-aware grouping.

Usage

```
selectboost_fda(
  x,
  y = NULL,
  mode = c("fast", "auto"),
  selector = "msgps",
  selector_fun = NULL,
  selector_args = list(),
  groups = NULL,
  family = c("gaussian", "binomial"),
  association = NULL,
  group_method = c("threshold", "community"),
  association_method = c("correlation", "neighborhood", "hybrid", "interval"),
  within_blocks = TRUE,
  bandwidth = NULL,
  interval_groups = NULL,
  width = NULL,
  step = width,
  decay = 1,
  ...
)
```

Arguments

`x`	Any input accepted by `as_functional_matrix()`, or an `fda_design` object.
`y`	Response vector. Leave as `NULL` when `x` is an `fda_design`.
`mode`	`"fast"` for a fixed `c0` grid or `"auto"` for the adaptive version.
`selector`	Base selector used inside SelectBoost. Choose from `"msgps"`, `"lasso"`, `"group_lasso"`, `"sparse_group_lasso"`, the backend-specific aliases `"glmnet"`, `"grpreg"`, `"sgl"`, or provide a custom function.
`selector_fun`	Optional custom base selector. It must return a coefficient vector of length `p`.
`selector_args`	Optional named list of arguments forwarded to the base selector.

groups	Optional feature groups used by grouped base selectors such as "grpreg". Defaults to block-level groups for list inputs.
family	Model family passed to built-in selectors.
association	Optional custom association matrix used to define FDA-aware groups.
group_method	Functional grouping backend: threshold-based or community-based.
association_method	Association structure used to build FDA-aware groups.
within_blocks	Should SelectBoost groups stay within functional blocks?
bandwidth	Optional maximum within-block lag retained in groups.
interval_groups, width, step, decay	Additional arguments passed to make_functional_grouping_function().
...	Additional arguments passed to SelectBoost::fastboost() or SelectBoost::autoboost().

Value

An object of class selectboost_fda_result.

---

selected *Extract Selected Features or Groups*

Description

Returns the selected rows from selection_map() for stability-selection or SelectBoost fits.

Usage

```
selected(x, ...)
```

Arguments

x	A fitted selection object.
...	Additional arguments passed to the relevant method.

Value

A data frame.

selection_map *Feature-Level Selection Map*

Description

Returns a feature map augmented with selection summaries from a fit object.

Usage

```
selection_map(x, level = c("feature", "group", "basis"), ...)
```

Arguments

x	An fda_design, fda_stability_selection, or selectboost_fda_result object.
level	Summary level. "feature" returns one row per coefficient, "group" returns one row per stability/interval group, and "basis" returns one row per basis-expanded predictor.
...	Additional arguments passed to the relevant method.

Value

A data frame.

simulate_fda_scenario *Simulate an FDA Benchmark Scenario*

Description

Generates raw functional predictors, scalar covariates, a response, and the mapped ground truth for the transformed design matrix.

Usage

```
simulate_fda_scenario(
  n = 80L,
  grid_length = 60L,
  family = c("gaussian", "binomial"),
  representation = c("grid", "basis", "fpca"),
  transforms = NULL,
  basis_df = 7L,
  n_components = 5L,
  scenario = c("localized_dense", "distributed_smooth", "confounded_blocks"),
  confounding_strength = NULL,
  active_region_scale = 1,
```

```
  local_correlation = 0,
  include_scalar = TRUE,
  noise_sd = 0.4,
  seed = NULL
)
```

Arguments

n	Number of observations.
grid_length	Number of grid points per functional predictor.
family	Model family used to generate the response.
representation	Representation used when building the returned fda_design(): "grid" keeps the raw curves, "basis" applies a spline-basis transform, and "fpca" applies FPCA scores.
transforms	Optional transform list passed to fda_design(). When omitted, a sensible default is chosen from representation.
basis_df	Degrees of freedom used when representation = "basis".
n_components	Number of FPCA components used when representation = "fpca".
scenario	Benchmark scenario. "localized_dense" emphasizes narrow active regions under strong local correlation, "distributed_smooth" spreads the effect over broader smooth regions, and "confounded_blocks" adds stronger nuisance structure near the active block.
confounding_strength	Strength of cross-block confounding injected into the nuisance curve. Higher values make plain SelectBoost less able to separate true local signals from correlated nuisance structure.
active_region_scale	Positive multiplier applied to the width of the active regions. Values below 1 create narrower active regions.
local_correlation	Non-negative smoothing parameter applied to the simulated curves. Larger values increase local correlation along the grid.
include_scalar	Should scalar covariates be included in the design and truth object?
noise_sd	Observation noise level.
seed	Optional random seed.

Value

An object of class fda_simulation_data.

Examples

```
sim <- simulate_fda_scenario(n = 24, grid_length = 16, seed = 1)
sim
head(sim$truth$active_features)
```

spectra_example *Spectroscopy-Style Functional Example*

Description

Simulated dense spectra with one signal block, one nuisance block, and two scalar covariates. The response is continuous and depends on localized regions of the signal spectrum plus the scalar covariates.

Usage

```
spectra_example
```

Format

A list with four components:

grid Numeric vector of wavelength locations.

response Numeric response vector.

predictors Named list of functional predictor matrices.

scalar_covariates Data frame with scalar covariates.

Source

Simulated for package examples.

stability_selection_fda *Grouped Stability Selection for Functional Predictors*

Description

Repeatedly subsamples observations, refits a sparse base selector, and computes exact feature- and group-level selection frequencies. This is the generic FDA recipe for basis expansions, discretized curves, or FPCA scores.

Usage

```
stability_selection_fda(
  x,
  y = NULL,
  selector = "group_lasso",
  selector_fun = NULL,
  groups = NULL,
  family = c("gaussian", "binomial"),
```

```
  B = 100L,
  sample_fraction = 0.5,
  cutoff = 0.75,
  seed = NULL,
  keep_subsamples = FALSE,
  ...
)
```

Arguments

x	Any input accepted by as_functional_matrix(), or an fda_design object.
y	Response vector. Leave as NULL when x is an fda_design.
selector	Either "lasso", "group_lasso", "sparse_group_lasso", one of the backend-specific aliases ("glmnet", "grpreg", "sgl"), or a custom function.
selector_fun	Optional custom selector. It must accept X, y, groups, and family, and return either a coefficient vector or a logical selection vector of length p.
groups	Optional grouping structure. Defaults to block-level groups when x is supplied as a list, and otherwise to one group per feature.
family	Model family passed to the built-in selectors.
B	Number of subsampling replicates.
sample_fraction	Fraction of observations drawn without replacement in each subsample.
cutoff	Stability threshold used to define selected_features and selected_groups.
seed	Optional random seed.
keep_subsamples	Should the sampled row indices be returned?
...	Additional arguments forwarded to the built-in or custom selector.

Value

An object of class fda_stability_selection.

---

suggest_c0_grid *Suggest a c0 Grid for FDA-SelectBoost*

Description

Builds a data-driven c0 grid from an FDA-aware association matrix.

Usage

```
suggest_c0_grid(
  x,
  n = 5L,
  method = c("quantile", "linear"),
  association_method = c("correlation", "neighborhood", "hybrid", "interval"),
  within_blocks = TRUE,
  bandwidth = NULL,
  interval_groups = NULL,
  width = NULL,
  step = width,
  decay = 1
)
```

Arguments

x	Any input accepted by as_functional_matrix().
n	Number of grid values to return.
method	Grid construction rule: "quantile" or "linear".
association_method	Association structure passed to functional_association().
within_blocks, bandwidth, interval_groups, width, step, decay	Passed to functional_association().

Value

A decreasing numeric vector of c0 values.

---

summarise_benchmark_advantage *Summarize the Advantage of FDA-SelectBoost Over Baselines*

Description

Computes the per-scenario and per-level gain of a target method over one or more reference methods. This is intended to make the benchmark story explicit when comparing FDA-aware SelectBoost to existing baselines.

Usage

```
summarise_benchmark_advantage(
  x,
  target = "selectboost",
  reference = c("plain_selectboost", "stability"),
  level = c("feature", "group", "basis"),
  metric = "f1",
```

```
  optimize = c("max", "min"),
  select_c0 = c("best", "all")
)
```

Arguments

x	An fda_benchmark or fda_simulation_study object.
target	Method whose gain should be assessed.
reference	One or more baseline methods.
level	Evaluation level.
metric	Metric used both for best-c0 selection and for the reported gains.
optimize	Should larger or smaller values of metric be preferred?
select_c0	Keep all c0 rows or only the best one per method and replicate.

Value

A data frame.

---

summarise_benchmark_performance

Summarize Benchmark Performance by Method

Description

Collapses raw benchmark rows into method-level performance summaries, with an option to retain only the best c0 per method and replication.

Usage

```
summarise_benchmark_performance(
  x,
  level = c("feature", "group", "basis"),
  metric = "f1",
  optimize = c("max", "min"),
  select_c0 = c("best", "all")
)
```

Arguments

x	An fda_benchmark or fda_simulation_study object.
level	Evaluation level.
metric	Metric used to pick the best c0 when select_c0 = "best".
optimize	Should larger or smaller values of metric be preferred?
select_c0	Keep all c0 rows or only the best one per method and replicate.

Value

A data frame.

Index