Package: woven (via r-universe)

November 17, 2026

Title Weighted Omics View Embedding via Nystrom for Incomplete Multi-Omics Data

Version 0.99.6

Description Supervised multi-omics integration for block-missing (``ragged'') data, where some subjects lack one or more whole omics modalities (assay blocks) rather than scattered individual values. WOVEN fits a shared latent space with SUMCOR (sum of correlations) multi-set canonical correlation analysis, a criterion that maximizes the summed pairwise correlation between modalities' latent scores, using only the anchor subjects observed in every modality; class labels enter through a label-augmented cross-covariance, and per-modality graph Laplacians regularize the fit. Every other subject is then scored from the modalities it has by a Nystrom extension, a way to place new points in an already fitted embedding (for WOVEN's linear embedding, multiplying the observed features by the learned projection matrices), so no missing block is imputed. The effective sample size (ESS) retention, the fraction of subjects a method can score, is therefore the fraction with at least one observed modality, rather than the complete-case fraction. woven_tune() chooses the regularization settings by cross-validated balanced error rate, and the fitted model is summarized by plots and a table of the top features per modality.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

URL https://github.com/NathanBresette/woven

BugReports https://github.com/NathanBresette/woven/issues

biocViews Software, StatisticalMethod, DimensionReduction, Classification, Clustering, GeneExpression, Transcriptomics, Proteomics, Metabolomics

Depends R (>= 4.5.0)

Imports Matrix, MatrixGenerics, MultiAssayExperiment, RANN, cluster, grDevices, methods, stats, SummarizedExperiment, utils

Suggests ggplot2, testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev zlib1g-dev

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

Date/Publication 2026-11-17 19:58:45 UTC

RemoteUrl https://github.com/bioc/woven

RemoteRef HEAD

RemoteSha 220139a1eaaa821e13917162827dbd2977e9e80b

Contents

woven-package . . . 3
$,wovenTune-method . . . 4
plot,woven,missing-method . . . 4
plot,wovenTune,missing-method . . . 6
show,woven-method . . . 6
show,wovenTune-method . . . 7
summary,woven-method . . . 8
woven . . . 9
woven-accessors . . . 12
woven-class . . . 13
woven_davies_bouldin . . . 15
woven_effect_bias . . . 15
woven_ess_retention . . . 16
woven_example . . . 17
woven_mcca_dual . . . 18
woven_metrics . . . 19
woven_nmi . . . 20
woven_nystrom_error . . . 21
woven_plot_coverage . . . 22
woven_plot_loadings . . . 23
woven_plot_variance . . . 24
woven_plot_vip . . . 25
woven_precompute . . . 26
woven_predict . . . 27
woven_rv . . . 28
woven_scores . . . 29
woven_silhouette . . . 30
woven_top_features . . . 31
woven_tune . . . 32
wovenTune-class . . . 33

Index **35**

woven-package *woven: supervised multi-omics integration for block-missing data*

Description

WOVEN (Weighted Omics View Embedding via Nystrom) learns a shared, supervised latent space across several omics modalities when some subjects are missing whole modalities. The model is fitted on the anchor subjects (those observed in every modality), and every other subject is then scored from the modalities it does have, without imputing the missing ones.

Main functions

- [woven()]: fit a model from a list of matrices or a MultiAssayExperiment.
- [woven_precompute()]: build the per-modality graphs once for repeated fits (cross-validation, tuning).
- [woven_scores()]: latent scores for new subjects.
- [woven_predict()]: class predictions for new subjects.
- [woven_metrics()]: silhouette, Davies-Bouldin, NMI and ESS retention of a fit.
- [woven-accessors]: [latent_scores()], [projection_matrices()], [anchor_indices()], [singular_values()].

Plots

`plot()` on a fit ([plot,woven,missing-method]) shows the latent space; [woven_plot_vip()], [woven_plot_loadings()] and [woven_plot_variance()] show feature importance, loadings and variance per dimension.

Getting started

See `vignette("woven-intro", package = "woven")`.

Author(s)

Maintainer: Nathan Bresette <nathanbresette04@gmail.com> (ORCID)

Authors:

- Ai-Ling Lin
- Jianlin Cheng

Other contributors:

- National Institutes of Health / National Institute on Aging (R56AG079586) [funder]

See Also

Useful links:

- https://github.com/NathanBresette/woven
- Report bugs at https://github.com/NathanBresette/woven/issues

$,wovenTune-method *Access parts of a woven_tune() result*

Description

`tune$best`, `tune$results`, `tune$fit`, `tune$folds` and `tune$chance` return the slots of the same name.

Usage

```
## S4 method for signature 'wovenTune'
x$name
```

Arguments

`x`	a `wovenTune` object
`name`	one of `best`, `results`, `fit`, `folds`, `chance`

Value

the requested slot

Examples

```
data(woven_example)
set.seed(1)
tune <- woven_tune(woven_example$X_missing, woven_example$Y,
    K = 3L, gamma_y = 1, ridge_w = c(1e-8, 1), screen_top = Inf,
    verbose = FALSE
)
tune$best
```

plot,woven,missing-method *Plot a WOVEN fit*

Description

One entry point for every WOVEN plot. The default draws the latent space: two latent dimensions with subjects coloured by class (the labels stored in the fit, unless `labels` is given), anchors (subjects observed in every modality, used to learn the projections) as filled points, and subjects scored from a subset of modalities as open points. The other types call the matching `woven_plot_*()` function.

Usage

```
## S4 method for signature 'woven,missing'
plot(
  x,
  y,
  type = c("scores", "features", "loadings", "coverage", "variance"),
  labels = NULL,
  dims = c(1L, 2L),
  highlight_anchors = TRUE,
  ...
)
```

Arguments

x	a woven object from [woven()]
y	unused; present for S4 'plot' signature compatibility
type	which plot: "scores" (default, the latent space), "features" (top features per modality, [woven_plot_vip()]), "loadings" ([woven_plot_loadings()]), "coverage" (which subjects have which modalities, [woven_plot_coverage()]), or "variance" ([woven_plot_variance()]).
labels	for type = "scores": a vector of length n to colour points by. Defaults to the class labels used to fit the model. A numeric vector with more than 10 distinct values (for example age) is drawn as a colour gradient.
dims	for type = "scores": integer vector of length 2, the latent dimensions to plot (default c(1, 2)). A fit with K = 1 is drawn as the distribution of dimension 1 per class.
highlight_anchors	for type = "scores": logical, draw anchors filled and other subjects open (default TRUE)
...	passed to the woven_plot_*() function for the other types, e.g. n_top or modality.

Value

a ggplot object, printed automatically when called at the top level (standard R plot-method behaviour).

Examples

```
data(woven_example)
fit <- woven(woven_example$X_missing,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
plot(fit)
plot(fit, type = "features", n_top = 8)
plot(fit, type = "coverage")
```

`plot,wovenTune,missing-method` *Plot woven_tune() results*

Description

Held-out balanced error rate for every setting tried: one panel per `screen_top` value, `ridge_w` across, `gamma_y` down, darker for lower error. The selected setting is outlined. When several `lambda` values were tried, the best over `lambda` is shown.

Usage

```
## S4 method for signature 'wovenTune,missing'
plot(x, y, ...)
```

Arguments

`x`	a `wovenTune` object from [woven_tune()]
`y`	unused
`...`	unused

Value

a ggplot object

Examples

```
data(woven_example)
set.seed(1)
tune <- woven_tune(woven_example$X_missing, woven_example$Y,
    K = 3L, gamma_y = c(1, 5), ridge_w = c(1e-8, 1), screen_top = Inf,
    verbose = FALSE
)
plot(tune)
```

`show,woven-method` *Show method for WOVEN fit*

Description

Called automatically when a `woven` object is printed or auto-printed at the console.

Usage

```
## S4 method for signature 'woven'
show(object)
```

Arguments

object	a woven object from [woven()]

Value

Invisibly returns NULL; called for its side effect.

Examples

```
set.seed(1)
n <- 20
K <- 2L
X1 <- matrix(rnorm(n * 5), n, 5)
X2 <- matrix(rnorm(n * 4), n, 4)
Y <- rep(1:2, each = n / 2)
miss <- matrix(FALSE, n, 2)
miss[c(15, 17, 19), 1] <- TRUE
miss[c(16, 18, 20), 2] <- TRUE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
anchor_idx <- which(rowSums(miss) == 0)
fit <- woven(list(X1, X2), Y = Y, anchor_idx = anchor_idx, K = K)
fit
```

---

show,wovenTune-method *Show a woven_tune() result*

Description

Show a woven_tune() result

Usage

```
## S4 method for signature 'wovenTune'
show(object)
```

Arguments

object	a wovenTune object from [woven_tune()]

Value

Invisibly returns NULL; called for its printed output.

Examples

```
data(woven_example)
set.seed(1)
tune <- woven_tune(woven_example$X_missing, woven_example$Y,
    K = 3L, gamma_y = c(1, 5), ridge_w = c(1e-8, 1), screen_top = Inf,
    verbose = FALSE
)
tune
```

---

summary,woven-method *Summarise a WOVEN fit*

Description

Prints a compact metrics table: silhouette, Davies-Bouldin, NMI, and ESS for the scored subjects, using the class labels stored in the fit unless `labels` is given.

Usage

```
## S4 method for signature 'woven'
summary(object, labels = NULL, ...)
```

Arguments

`object`	a woven object from [woven()]
`labels`	character, factor, or integer vector of length n. Defaults to the labels the model was fit with.
`...`	unused

Value

Invisibly returns a named numeric vector of metrics.

Examples

```
data(woven_example)
fit <- woven(woven_example$X_complete, Y = woven_example$Y, K = 3L)
summary(fit)
```

`woven` *Fit a supervised WOVEN model*

Description

Learns a shared supervised latent space across V omics modalities, handling block-missing data via anchor-restricted alignment and Nystrom projection. Labels Y are required – WOVEN is a supervised method (cf. DIABLO).

Usage

```
woven(
  X_list,
  Y,
  anchor_idx = NULL,
  K = 5L,
  lambdas = 0.1,
  gamma_y = 1,
  k_nn = 10L,
  ridge_w = 1e-08,
  screen_top = Inf,
  scale = TRUE,
  precomp = NULL,
  verbose = TRUE,
  plot = interactive()
)
```

Arguments

`X_list` the data, one block per modality (subjects in rows, features in columns), as either

- a named list of numeric matrices or data frames. When every block has row names (subject IDs), the blocks may cover different subjects in any order: woven matches them by ID and treats a subject absent from a block as missing that modality. Without row names, all blocks must have the same rows in the same order, with a subject missing a modality given an all-NA row; or
- a `MultiAssayExperiment`.

When a MultiAssayExperiment is supplied, each experiment becomes one modality, subjects are aligned to `colData(X_list)` via `sampleMap(X_list)`, and a subject not present in a given experiment gets an all-NA row for it automatically.

`Y` class labels for all subjects (factor, character or integer). Plots and tables list the classes in the order of `levels(as.factor(Y))`; pass a factor to choose the order. Either in subject order (the row order of the blocks, or of `colData(X_list)` for a MultiAssayExperiment) or a named vector whose names are the subject IDs, in which case order does not matter. A single string is instead treated as the

name of a colData column to use as labels, but only when X_list is a MultiAssayExperiment. Only anchor subjects' labels enter the supervised objective.

anchor_idx — integer vector – indices of fully-observed subjects (observed in all V modalities). Must have length >= K. If NULL (default), anchors are detected automatically as subjects with no block-missing modalities. A supplied anchor_idx is checked: indices must be distinct, within 1..n, and observed in every modality; offending subjects are named in the error.

K — integer – number of latent dimensions (default 5). Must be smaller than n and no larger than the number of anchors. Each modality's projection matrix has rank at most min(n_a, p_v), so a warning is given if K exceeds a modality's number of features.

lambdas — numeric scalar or length-V vector – Laplacian regularization strength per modality (default 0.1 for all)

gamma_y — numeric >= 0 – supervision strength. 0 = unsupervised CCA. Default 1.0 (equal weight to cross-modal alignment and label alignment). Tune via cross-validation on anchor set if labels are noisy.

k_nn — integer – k-nearest-neighbors for Laplacian graph (default 10).

ridge_w — numeric >= 0 – ridge shrinkage applied to the per-modality Gram matrix during projection-matrix (W) recovery (default 1e-8, a purely numerical-stability nudge with no regularizing effect). Increase (e.g. 0.1-5.0) to control capacity when p_v » n_a and the true signal is diffuse across many features – select it with [woven_tune()] (held-out classification error); silhouette is insensitive to this parameter even where it materially changes classification accuracy.

screen_top — integer or Inf – restricts W recovery to the top screen_top features per modality by ANOVA F-statistic against Y, computed on anchor subjects only (default Inf = no screening, uses all features). A genuine re-fit on the reduced feature set, not a post-hoc truncation. Complementary to ridge_w; both target overfitting when p_v is large relative to n_a. Select jointly with ridge_w via cross-validation, not independently (the two interact).

scale — logical – standardize each modality's features to zero mean and unit variance before fitting (default TRUE), so no modality dominates by measurement scale. Apply biological transforms (CLR/log/VST) upstream. The center and scale are stored and reapplied to new subjects in [woven_scores()]. When precomp is supplied it must have been built with the same scale value; a mismatch is an error.

precomp — optional output of [woven_precompute()] – pre-built Laplacians for the same X_list. Pass this when calling woven() multiple times on the same subjects (e.g. comparing hyperparameter settings) to avoid rebuilding the graph each time. Not usable for fits on a subset of subjects, such as cross-validation training folds. When supplied, k_nn is taken from precomp.

verbose — logical – report progress (default TRUE): a summary of the input, one line per step with timings, and a closing summary of coverage and top features. All output is via message().

plot — logical – draw the latent space when the fit finishes (default: in interactive sessions only). Needs ggplot2.

Details

All V use the same closed-form dual SUMCOR MCCA solver: a single eigendecomposition of a block matrix built from the per-modality Laplacians and the label kernel, with no iterative optimization. For V=2 this is equivalent to dual supervised CCA; for V>=3 it is the global optimum of the relaxed SUMCOR trace objective (see `woven_mcca_dual`).

Value

an object of S4 class `woven`. Key slots:

Z n x K matrix of consensus latent scores for ALL n subjects (anchors and block-missing). The primary output for downstream analysis.

W_list list of V projection matrices, each p_v x K

Z_anchors list of V anchor latent score matrices, each n_a x K

singular_values K supervised canonical correlations

anchor_idx indices of anchor (fully-observed) subjects

Y_levels integer codes of the classes used during fitting

Y_labels class names, `levels(as.factor(Y))`; for numeric `Y` these are the numbers as strings

K, V, n dimensions

lambdas, gamma_y hyperparameters

See `woven` for the complete slot list.

See Also

[woven_scores()], [woven_predict()], [woven_metrics()], [woven_precompute()], [woven-accessors]

Examples

```
set.seed(1)
n <- 60
p1 <- 20
p2 <- 15
K <- 2
Y <- rep(1:2, each = n / 2)
X1 <- matrix(rnorm(n * p1), n, p1)
X2 <- matrix(rnorm(n * p2), n, p2)
# 30% block missingness; enforce >= 1 view per subject
miss <- matrix(runif(n * 2) < 0.3, n, 2)
for (i in which(rowSums(miss) == 2)) miss[i, sample(2, 1)] <- FALSE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
# anchor_idx auto-detected from NA pattern -- no need to specify
fit <- woven(list(X1, X2), Y = Y, K = K)
dim(latent_scores(fit)) # 60 x 2 -- all subjects scored

# Most often each assay is its own table covering its own subjects.
# Give the tables row names (subject IDs) and woven matches them:
```

```
ids <- paste0("S", seq_len(n))
rna <- matrix(rnorm(50 * p1), 50, p1, dimnames = list(ids[1:50], NULL))
prot <- matrix(rnorm(45 * p2), 45, p2, dimnames = list(ids[16:60], NULL))
labels <- setNames(Y, ids)
fit_tab <- woven(list(RNA = rna, Protein = prot), Y = labels, K = K)

# Equivalently, from a MultiAssayExperiment: each experiment is one
# modality, and subjects missing an experiment need no special handling.
library(MultiAssayExperiment)
ids <- paste0("S", seq_len(n))
rownames(X1) <- ids
rownames(X2) <- ids
present1 <- ids[!is.na(X1[, 1])]
present2 <- ids[!is.na(X2[, 1])]
exps <- ExperimentList(list(
    Mod1 = t(X1[present1, , drop = FALSE]),
    Mod2 = t(X2[present2, , drop = FALSE])
))
cd <- DataFrame(row.names = ids, group = Y)
mae <- MultiAssayExperiment(exps, colData = cd)
fit_mae <- woven(mae, Y = "group", K = K)
dim(latent_scores(fit_mae)) # same 60 x 2 -- all subjects scored
```

---

woven-accessors *Accessors for a fitted WOVEN model*

Description

Extract the main results from a woven fit.

Usage

```
latent_scores(x)

## S4 method for signature 'woven'
latent_scores(x)

projection_matrices(x)

## S4 method for signature 'woven'
projection_matrices(x)

anchor_indices(x)

## S4 method for signature 'woven'
anchor_indices(x)

singular_values(x)
```

```
## S4 method for signature 'woven'
singular_values(x)
```

Arguments

x	a woven object.

Details

- latent_scores(): n x K matrix of consensus latent scores for every subject (anchors and block-missing subjects). Rows of subjects with no observed modality are NA. Row names come from the row names of the input matrices (or colData row names for a MultiAssayExperiment).
- projection_matrices(): named list of V projection matrices, each p_v x K (features by latent dimensions).
- anchor_indices(): integer row indices of the anchor subjects (observed in every modality) used to fit the model.
- singular_values(): length-K vector of supervised canonical correlations (eigenvalues of the dual problem), largest first.

Value

See Details.

Examples

```
data(woven_example)
fit <- woven(woven_example$X_complete,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
dim(latent_scores(fit))
names(projection_matrices(fit))
length(anchor_indices(fit))
singular_values(fit)
```

---

woven-class *S4 class for a fitted WOVEN model*

Description

The object returned by woven. Holds the consensus latent scores for every subject, the per-modality projection matrices, and the hyperparameters and bookkeeping needed by woven_scores, woven_predict, woven_metrics, and the woven_plot_* functions.

Slots

`Z` numeric matrix, n x K consensus latent scores for all n subjects (anchors and block-missing); rows with no observed modality are NA.

`W_list` list of V numeric matrices, each p_v x K, the per-modality projection matrices.

`Z_anchors` list of V numeric matrices, each n_a x K, the per-modality anchor latent scores before consensus averaging.

`singular_values` numeric vector of length K, the supervised canonical correlations from the dual eigendecomposition.

`anchor_idx` integer vector, indices (into the original n subjects) of the fully-observed anchor subjects.

`Y_anchor` integer class codes of the anchor subjects, in `anchor_idx` order.

`Y_levels` sorted unique integer class codes seen during fitting.

`Y_labels` character vector of class names, `levels(as.factor(Y))`, in code order (code 1 is `Y_labels[1]`). For numeric `Y` these are the numbers as strings, e.g. `c("1", "2", "3")`.

`K` integer, latent dimension.

`V` integer, number of modalities.

`n` integer, total number of subjects.

`mod_names` character vector of modality names; unnamed inputs get `"Modality 1"`, `"Modality 2"`, ...

`lambdas` numeric vector of length V, per-modality Laplacian regularization strength used at fit time.

`gamma_y` numeric, supervision strength used at fit time.

`k_nn` integer, k-NN graph size used at fit time.

`ridge_w` numeric, ridge strength used at fit time.

`screen_top` numeric (integer count or `Inf`), feature-screening depth used at fit time.

`scaled` logical, whether modalities were standardized before fitting.

`scale_center` list of V numeric vectors (or `NULL`), per-feature centers used for standardization, reapplied in `woven_scores`.

`scale_scale` list of V numeric vectors (or `NULL`), per-feature scales used for standardization, reapplied in `woven_scores`.

`fit_mcca` list, the raw output of `woven_mcca_dual` for advanced/diagnostic use.

`Y` factor of length n, the class label of every subject (names are the subject IDs when the data have row names). Used as the default labels by `plot`, `summary` and `woven_metrics`.

`observed` logical matrix, n x V: `TRUE` where a subject has data for a modality.

`class_means` list of V numeric matrices, each p_v x G: the mean of each (standardized) feature in each class, over the subjects observed in that modality. Used to colour features by the class in which they are highest.

See Also

[woven()] to fit a model; [woven-accessors] to read one.

woven_davies_bouldin *Davies-Bouldin index*

Description

Works on any n x K latent score matrix, so the same metric can be computed for WOVEN and for embeddings from other methods (for example DIABLO variates) when comparing them. [woven_metrics()] computes it directly from a fit.

Usage

```
woven_davies_bouldin(Z, labels)
```

Arguments

Z	numeric matrix n x K
labels	integer or factor of length n

Details

DB = (1/K) sum_i max_{j != i} (s_i + s_j) / d(c_i, c_j) where s_i = mean intra-cluster distance, d(c_i, c_j) = centroid distance.

Value

scalar >= 0, lower is better

Examples

```
set.seed(1)
Z <- matrix(rnorm(20 * 2), 20, 2)
labels <- rep(1:2, each = 10)
woven_davies_bouldin(Z, labels)
```

woven_effect_bias *Subgroup treatment-effect estimate bias*

Description

For simulation studies where the true treatment effect in each subgroup is known. Within each subgroup defined by labels, fits a linear model of a continuous outcome on a binary treatment indicator, adjusting for the latent scores Z, and compares the estimated treatment effect with the true effect. Subgroups with fewer than 5 subjects are skipped.

Usage

```
woven_effect_bias(Z, outcome, treatment, labels, true_effects)
```

Arguments

Z	numeric matrix n x K (latent scores; used as covariates)
outcome	numeric vector of length n (simulated continuous outcome)
treatment	integer/logical vector of length n (0/1 treatment indicator)
labels	integer or factor of length n (subgroup labels)
true_effects	numeric vector, true treatment effect per subgroup, in the order of `levels(as.factor(labels))`

Details

bias_g = |estimated_g - true_g| / |true_g| (relative) Returns mean bias across subgroups.

Value

scalar >= 0, lower is better

Examples

```
set.seed(1)
n <- 200
Z <- matrix(rnorm(n * 3), n, 3)
treatment <- rep(0:1, n / 2)
labels <- rep(1:2, each = n / 2)
true_eff <- c(0.5, 1.0)
outcome <- true_eff[labels] * treatment + Z[, 1] + rnorm(n, sd = 0.5)
woven_effect_bias(Z, outcome, treatment, labels, true_eff)
```

---

woven_ess_retention *Effective sample size (ESS) retention*

Description

The fraction of the cohort that a method can score: `n_used / n_total`. For WOVEN, every subject with at least one observed modality is scored. A complete-case method such as DIABLO scores only subjects observed in every modality, so its retention equals the anchor fraction.

Usage

```
woven_ess_retention(n_used, n_total)
```

Arguments

n_used	integer, number of subjects with a latent score
n_total	integer, total subjects in dataset

Value

scalar in [0, 1], higher is better

Examples

```
woven_ess_retention(n_used = 80, n_total = 100)
```

`woven_example` *Simulated three-modality example dataset*

Description

A small simulated dataset (90 subjects, three modalities, three classes) used in the help-page examples. All values are synthetic: three latent factors with class-specific means are mapped to each modality by random loadings, Gaussian noise is added, and each feature is standardized. In the block-missing version, 30 subjects each lose one whole modality, chosen at random (missing completely at random). For a real block-missing example see the package vignette, which uses the TCGA adrenocortical carcinoma data `miniACC` from MultiAssayExperiment.

Usage

```
woven_example
```

Format

A list with three components:

X_complete Named list of three numeric matrices with subjects in rows: `RNA` (90 x 25), `Methylation` (90 x 20) and `Proteomics` (90 x 15). No missing values.

X_missing The same three matrices with 30 subjects (33%) each missing one entire modality block (all-NA row). 60 subjects are observed in all three modalities.

Y Character vector of 90 class labels, `"CN"`, `"MCI"` and `"AD"` (30 each). The names are illustrative only; no clinical data were used.

Source

Simulated with `inst/scripts/make_woven_example.R` (`system.file("scripts", "make_woven_example.R", package = "woven")`), which reproduces `data/woven_example.rda` exactly. The data are released under the package license (MIT).

Examples

```
data(woven_example)
fit <- woven(woven_example$X_missing,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
fit # all 90 subjects are scored, including the 30 block-missing ones
```

`woven_mcca_dual` *Closed-form dual SUMCOR MCCA solver (low-level)*

Description

The solver that `woven` calls internally. Most users should call `woven()`, which adds input standardization, automatic anchor detection, MultiAssayExperiment input, latent scores for every subject, and an S4 result object. `woven_mcca_dual()` is exported for method developers who want the raw solver output (anchor scores, metric matrices, anchor Laplacians) without that wrapping.

Usage

```
woven_mcca_dual(
  X_list,
  anchor_idx,
  Y,
  K = 5L,
  lambdas = 0.1,
  gamma_y = 1,
  k_nn = 10L,
  La_list_precomp = NULL,
  ridge_w = 1e-08,
  screen_top = Inf,
  verbose = TRUE
)
```

Arguments

`X_list`	list of V matrices, each n x p_v. Rows that are entirely `NA` mark subjects missing that modality (block-missing).
`anchor_idx`	integer vector of subjects observed in every modality. Validated: indices must be distinct, within 1..n, and not block-missing in any modality.
`Y`	vector of length n of class labels. Only `Y[anchor_idx]` is used.
`K`	integer, number of latent dimensions. Each projection matrix has rank at most `min(n_a, p_v)`; a warning is given when `K` exceeds a modality's feature count.
`lambdas`	numeric scalar or length-V vector, Laplacian regularization.
`gamma_y`	numeric >= 0, label supervision strength.
`k_nn`	integer, k-NN graph size (ignored if `La_list_precomp` is supplied).
`La_list_precomp`	optional list of V pre-extracted n_a x n_a anchor Laplacians.
`ridge_w`	numeric >= 0, relative ridge strength in W recovery (`ridge_w * max(diag(Ka_v))` is added to the diagonal of the anchor Gram matrix `Ka_v`). The default 1e-8 only stabilizes the solve.

screen_top	integer or Inf; if finite, W recovery uses only the screen_top features per modality with the largest ANOVA F-statistic against Y on anchor subjects. Screened-out features get a zero row in W.
verbose	logical, print progress messages.

Details

The fit is a single eigendecomposition of a (V n_a) x (V n_a) block matrix, with no iterations, random restarts or local optima. SUMCOR ("sum of correlations") is the multi-set CCA criterion that maximizes the sum of pairwise correlations between the modalities' latent scores.

Value

a list with components W_list (V matrices, p_v x K), Za_list (V anchor score matrices, n_a x K), Xa_list (anchor data used), singular_values (length K), M_list, L_a_list, anchor_idx, K, V, lambdas, gamma_y, k_nn, n, p_v.

See Also

[woven()]

Examples

```
set.seed(1)
n <- 20
X1 <- matrix(rnorm(n * 5), n, 5)
X2 <- matrix(rnorm(n * 4), n, 4)
X3 <- matrix(rnorm(n * 3), n, 3)
Y <- rep(1:2, each = n / 2)
fit <- woven_mcca_dual(list(X1, X2, X3),
    anchor_idx = seq_len(14L), Y = Y,
    K = 2L, verbose = FALSE
)
fit$singular_values
```

---

woven_metrics *Convenience wrapper: compute core metrics directly from a woven fit*

Description

Computes silhouette width, Davies-Bouldin index, NMI and ESS retention on the subjects the fit scored, using latent_scores(fit) and the fit's subject count. NMI uses k-means, so call set.seed() first for reproducible values.

Usage

```
woven_metrics(fit, labels = NULL)
```

Arguments

fit	woven object from [woven()]
labels	integer, factor, or character vector of length n with subgroup labels for all subjects. Defaults to the labels the model was fit with.

Value

named numeric vector with elements Silhouette, Davies-Bouldin, NMI and ESS

See Also

[woven_silhouette()], [woven_davies_bouldin()], [woven_nmi()], [woven_ess_retention()]

Examples

```
set.seed(1)
n <- 40
K <- 2L
X1 <- matrix(rnorm(n * 8), n, 8)
X2 <- matrix(rnorm(n * 6), n, 6)
Y <- rep(1:2, each = n / 2)
miss <- matrix(runif(n * 2) < 0.3, n, 2)
for (i in which(rowSums(miss) == 2)) miss[i, sample(2, 1)] <- FALSE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
fit <- woven(list(X1, X2), Y = Y, K = K)
set.seed(1) # NMI uses k-means
woven_metrics(fit)
```

---

woven_nmi *Normalized mutual information between cluster assignments and true labels*

Description

Works on any n x K latent score matrix, so the same metric can be computed for WOVEN and for embeddings from other methods (for example DIABLO variates) when comparing them. [woven_metrics()] computes it directly from a fit.

Usage

```
woven_nmi(Z, labels, n_cl = NULL, n_start = 10L)
```

Arguments

Z	numeric matrix n x K
labels	integer or factor of length n (true labels)
n_cl	integer, number of clusters (default = number of unique labels)
n_start	integer, k-means random starts

Details

Clusters Z with k-means and computes the normalized mutual information between the cluster assignments and `labels`. k-means uses `n_start` random starts, so the result depends on R's random number generator: call `set.seed()` first for a reproducible value (the package itself never sets the seed). [woven_metrics()] and `summary()` call this function and inherit the same behaviour.

Value

scalar in [0, 1], higher is better

Examples

```
Z <- rbind(matrix(rnorm(40), 20, 2), matrix(rnorm(40, mean = 3), 20, 2))
labels <- rep(1:2, each = 20)
set.seed(1) # k-means is random; set the seed for a reproducible result
woven_nmi(Z, labels)
```

`woven_nystrom_error` *Leave-anchor-out Nystrom projection error*

Description

A diagnostic for how stable the projection of an unseen subject is. For each of `n_loo` randomly chosen anchor subjects, WOVEN is refitted without that subject (same hyperparameters as `fit`), the held-out subject is projected with the refitted projection matrices (the linear Nystrom out-of-sample extension used for block-missing subjects), and the Euclidean distance to its latent score in the full fit (`latent_scores(fit)`) is recorded. The function returns the mean distance. Small values mean that a subject's latent score does not depend much on whether it was in the training set.

Usage

```
woven_nystrom_error(fit, X_list, n_loo = NULL)
```

Arguments

`fit`	a woven object from [woven()]
`X_list`	the same list of modality matrices passed to [woven()], on the original (unscaled) feature scale
`n_loo`	integer, number of anchors to hold out, between 1 and the number of anchors (default `min(20, n_a)`)

Details

Because latent dimensions are identified only up to sign, each refitted dimension is sign-aligned to the full fit before the distance is taken. The anchors to hold out are drawn with `sample()`, so the result depends on R's random number generator: call `set.seed()` first for a reproducible value (the package itself never sets the seed).

Value

scalar >= 0, lower is better (mean distance per held-out anchor)

Examples

```
data(woven_example)
fit <- woven(woven_example$X_complete,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
set.seed(1) # held-out anchors are sampled at random
woven_nystrom_error(fit, woven_example$X_complete, n_loo = 5L)
```

---

woven_plot_coverage *Plot which subjects have which modalities*

Description

One bar per pattern of observed modalities (for example "RNA + Proteomics"), showing how many subjects have exactly that pattern, split by class. The subtitle compares the subjects a complete-case method can use (those observed in every modality, the anchors) with the subjects WOVEN scores. The class split shows whether the subjects a complete-case analysis would drop differ in their class mix.

Usage

```
woven_plot_coverage(fit, max_patterns = 12L, main = "Modality coverage")
```

Arguments

`fit`	woven object from [woven()]
`max_patterns`	integer: show at most this many bars (default 12). With many modalities there can be dozens of patterns; the least common are combined into one "Other patterns" bar.
`main`	character: plot title (default "Modality coverage")

Value

a ggplot object (printed automatically; add layers with +)

See Also

[woven()], [woven_ess_retention()]

Examples

```
data(woven_example)
fit <- woven(woven_example$X_missing,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
woven_plot_coverage(fit)
```

`woven_plot_loadings` *Plot feature loadings for one WOVEN latent dimension*

Description

For a given latent dimension, shows the top features by absolute loading for each modality (or a selected subset), colored by loading sign. Positive loadings are blue; negative loadings are red-orange. Equivalent to DIABLO's `plotLoadings()`.

Usage

```
woven_plot_loadings(
  fit,
  dim = 1L,
  n_top = 15L,
  feature_names = NULL,
  modality = NULL,
  main = NULL
)
```

Arguments

`fit`	woven object from [woven()]
`dim`	integer: which latent dimension to plot (1..K, default 1)
`n_top`	integer: number of top features per modality (default 15)
`feature_names`	optional list of V character vectors (one per modality). If a plain character vector is passed for a single-modality call, it is used for that modality. If NULL, uses rownames of W or "Feature_j".
`modality`	NULL, or one or more modality indices or names: plot only those modalities. If NULL (default), all V modalities are shown in faceted panels.
`main`	character: plot title. If NULL, a default is used.

Value

a ggplot object (printed automatically; add layers with +)

See Also

[woven_plot_vip()], [woven_plot_variance()]

Examples

```
set.seed(1)
n <- 60
p1 <- 30
p2 <- 20
K <- 3
Y <- rep(1:3, each = n / 3)
X1 <- matrix(rnorm(n * p1), n, p1)
X2 <- matrix(rnorm(n * p2), n, p2)
miss <- matrix(runif(n * 2) < 0.3, n, 2)
for (i in which(rowSums(miss) == 2)) miss[i, sample(2, 1)] <- FALSE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
fit <- woven(list(X1, X2), Y = Y, K = K)
woven_plot_loadings(fit, dim = 1L)
```

---

woven_plot_variance *Plot the strength of each WOVEN latent dimension*

Description

Bar chart of each latent dimension's share of the squared supervised canonical correlations (`singular_values(fit)^2`), with the cumulative share as a line. A dimension with a small share adds little shared, class-related signal, which helps when choosing `K`. The shares describe the cross-modal alignment, not the variance of the data.

Usage

```
woven_plot_variance(fit, main = "Strength of each latent dimension")
```

Arguments

`fit`	woven object from [woven()]
`main`	character: plot title

Value

a ggplot object (printed automatically; add layers with `+`)

See Also

[woven_plot_vip()], [woven_plot_loadings()]

Examples

```
data(woven_example)
fit <- woven(woven_example$X_missing,
    Y = woven_example$Y, K = 4L,
    verbose = FALSE
)
woven_plot_variance(fit)
```

`woven_plot_vip` *Plot the top features of a WOVEN fit by VIP score*

Description

Bar chart of the features with the highest Variable Importance in Projection (VIP) in each modality, one panel per modality. VIP combines a feature's loadings over all K latent dimensions, weighted by each dimension's share of the squared singular values (see [woven_top_features()] for the same ranking as a table). By default each bar is coloured by the class in which that feature has its highest mean, as in DIABLO's `plotLoadings(contrib = "max")`, so the plot shows both which features matter and which class they mark. A dashed line at VIP = 1 marks above-average importance.

Usage

```
woven_plot_vip(
  fit,
  modality = NULL,
  n_top = 15L,
  feature_names = NULL,
  color_by = c("class", "modality"),
  main = NULL
)
```

Arguments

`fit`	woven object from [woven()]
`modality`	`NULL` (default) for every modality, or one or more modality names or indices (1..V).
`n_top`	integer: number of top features per modality (default 15)
`feature_names`	optional character vector of feature labels, one per feature; only with a single `modality`. If NULL, uses the feature (column) names of the data.
`color_by`	`"class"` (default): colour each bar by the class with the highest mean for that feature. `"modality"`: one colour per modality.
`main`	character: plot title. If NULL, a default is generated.

Value

a ggplot object (printed automatically; add layers with +)

See Also

[woven_top_features()], [woven_plot_loadings()], [woven_plot_variance()]

Examples

```
data(woven_example)
fit <- woven(woven_example$X_missing,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
woven_plot_vip(fit)
woven_plot_vip(fit, modality = "RNA", n_top = 10)
```

---

woven_precompute *Pre-compute Laplacian graphs for reuse across multiple woven() calls*

Description

Builds k-NN RBF Laplacians for each modality from the observed data. Pass the result to `woven(..., precomp = precomp)` to avoid rebuilding the graph on every call when fitting the same subjects several times, for example to compare hyperparameter settings or run a sensitivity analysis. The graphs cover exactly the subjects in `X_list`, so they cannot be reused for fits on a subset of subjects (such as cross-validation training folds); fit those without `precomp`.

Usage

```
woven_precompute(X_list, k_nn = 10L, scale = TRUE)
```

Arguments

`X_list`	list of V matrices (n x p_v), or a `MultiAssayExperiment` (converted exactly as in [woven()], so the graphs line up with a later `woven(mae, ...)` call). Block-missing rows (all NA) are automatically excluded from the k-NN graph.
`k_nn`	integer, number of nearest neighbours (default 10)
`scale`	logical, standardize features to unit variance before building Laplacians (default TRUE); use the same value passed to [woven()].

Value

list of V sparse Laplacian matrices (n x n), one per modality. Pass directly to the `precomp` argument of [woven()]. The list records the `scale`, `k_nn` and `n` it was built with (as attributes `"woven_scale"`, `"woven_k_nn"`, `"woven_n"`), and [woven()] stops if its own `scale` or the number of subjects does not match.

See Also

[woven()]

Examples

```
set.seed(1)
n <- 40
K <- 2L
X1 <- matrix(rnorm(n * 8), n, 8)
X2 <- matrix(rnorm(n * 6), n, 6)
Y <- rep(1:2, each = n / 2)
miss <- matrix(runif(n * 2) < 0.3, n, 2)
for (i in which(rowSums(miss) == 2)) miss[i, sample(2, 1)] <- FALSE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
precomp <- woven_precompute(list(X1, X2), k_nn = 10L)
fit <- woven(list(X1, X2), Y = Y, K = K, precomp = precomp)
```

`woven_predict` *Predict class probabilities for new subjects*

Description

Projects new subjects into the WOVEN latent space with [woven_scores()] and returns soft class assignments from a nearest-centroid (or k-NN) classifier built on the anchor subjects' latent scores (computed by the same projection). Works the same way for complete and block-missing subjects.

Usage

```
woven_predict(fit, X_list_new, method = c("centroid", "knn"), k_pred = 5L)
```

Arguments

`fit`	woven object from [woven()]
`X_list_new`	new subjects, in any form accepted by [woven_scores()] (for example `list(RNA = new_rna)` to classify subjects that have RNA only).
`method`	`"centroid"` (default): inverse-squared-distance weights to the class centroids of the anchor scores. `"knn"`: k-NN vote among anchor subjects.
`k_pred`	integer – number of neighbors for knn method (default 5)

Value

data.frame with n_new rows and columns `predicted_class` (class name, as in `levels(as.factor(Y))`), `confidence` (probability of the predicted class), and one `p_<class>` column per class with soft probabilities. Subjects with no observed modality get uniform probabilities.

Examples

```
set.seed(1)
n <- 40
K <- 2L
X1 <- matrix(rnorm(n * 5), n, 5)
X2 <- matrix(rnorm(n * 4), n, 4)
Y <- rep(1:2, each = n / 2)
miss <- matrix(FALSE, n, 2)
miss[c(31, 33, 35), 1] <- TRUE
miss[c(32, 34, 36), 2] <- TRUE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
anchor_idx <- which(rowSums(miss) == 0)
fit <- woven(list(X1, X2), Y = Y, anchor_idx = anchor_idx, K = K)
pred <- woven_predict(fit, list(X1[1:5, ], X2[1:5, ]))
pred$predicted_class
```

---

woven_rv *RV coefficient between latent scores and ground-truth factor matrix*

Description

Works on any n x K latent score matrix, e.g. to compare an embedding with known factors in a simulation.

Usage

```
woven_rv(Z, Z_true)
```

Arguments

Z	numeric matrix n x K (inferred latent scores)
Z_true	numeric matrix n x K_true (e.g. ground-truth factor scores in a simulation)

Details

RV(X, Y) = trace(X X' Y Y') / sqrt(trace(X X' X X') * trace(Y Y' Y Y')) Measures similarity of two cross-product matrices; 1 = identical subspace.

Value

scalar in [0, 1], higher is better

Examples

```
set.seed(1)
Z <- matrix(rnorm(20 * 2), 20, 2)
Z_true <- matrix(rnorm(20 * 3), 20, 3)
woven_rv(Z, Z_true)
```

`woven_scores` *Extract latent scores for new subjects*

Description

Projects new subjects into the trained WOVEN latent space and returns an n_new x K score matrix: each observed modality's features (standardized with the training center and scale when the fit used `scale = TRUE`) are multiplied by that modality's projection matrix, and the results are averaged over the subject's observed modalities. This linear map is the Nystrom out-of-sample extension for WOVEN's linear embedding; no kernel evaluation against anchor subjects is needed.

Usage

```
woven_scores(fit, X_list_new)
```

Arguments

`fit`	woven object from [woven()]
`X_list_new`	the new subjects, as either • a list named with the training modality names, of matrices or data frames (subjects in rows). Blocks are matched by name, so their order does not matter and a modality may be left out (every new subject is then scored without it); blocks with row names are matched by subject ID; features are matched by column name when all training features are present; • an unnamed list of V matrices in training modality and feature order, with all-NA rows for subjects missing a modality; or • a `MultiAssayExperiment` whose experiments have the training modality names.

Details

For class predictions on new subjects, use [woven_predict()] instead.

Value

numeric matrix n_new x K of consensus latent scores, with rownames from `X_list_new[[1]]` and colnames `Dim1`, `Dim2`, etc. Subjects with no observed data in any view receive a row of NA.

See Also

[woven_predict()] for class predictions, [woven()] for model fitting

Examples

```
set.seed(1)
n <- 30
K <- 2L
X1 <- matrix(rnorm(n * 8), n, 8)
X2 <- matrix(rnorm(n * 6), n, 6)
Y <- rep(1:2, each = n / 2)
miss <- matrix(runif(n * 2) < 0.3, n, 2)
for (i in which(rowSums(miss) == 2)) miss[i, sample(2, 1)] <- FALSE
X1[miss[, 1], ] <- NA
X2[miss[, 2], ] <- NA
fit <- woven(list(X1, X2), Y = Y, K = K)
Z_new <- woven_scores(fit, list(X1[1:5, ], X2[1:5, ]))
dim(Z_new)
```

---

woven_silhouette *Average silhouette width*

Description

Works on any n x K latent score matrix, so the same metric can be computed for WOVEN and for embeddings from other methods (for example DIABLO variates) when comparing them. [woven_metrics()] computes it directly from a fit.

Usage

```
woven_silhouette(Z, labels)
```

Arguments

Z	numeric matrix n x K (latent scores)
labels	integer or factor of length n (subgroup labels)

Value

scalar in [-1, 1], higher is better

Examples

```
set.seed(1)
Z <- matrix(rnorm(20 * 2), 20, 2)
labels <- rep(1:2, each = 10)
woven_silhouette(Z, labels)
```

`woven_top_features` *Top features of a WOVEN fit*

Description

Ranks the features of each modality by their variable importance in projection (VIP) and returns them as a table, with the class in which each feature is highest. This is the tabular counterpart of `plot(fit, type = "features")`.

Usage

```
woven_top_features(fit, n = 10L, modality = NULL)
```

Arguments

`fit`	a woven object from [woven()]
`n`	integer: number of features per modality (default 10). Use `Inf` for all.
`modality`	`NULL` (default, all modalities), or one or more modality names or indices.

Details

VIP combines a feature's loadings over all latent dimensions, weighting each dimension by its share of the squared singular values, and is scaled so its mean square is 1 within a modality (so values above 1 are above average). It ranks features within a modality; it is not comparable across modalities with different numbers of features. Features removed by `screen_top` have zero loadings and a VIP of 0; because the mean square is taken over all of a modality's features, the features kept score higher when screening removes many of them.

Value

a data.frame with one row per feature and columns `modality`, `rank` (within modality), `feature`, `vip`, and `highest_in` (the class with the highest mean value of the feature, on the standardized scale the model was fit on).

See Also

[woven_plot_vip()], [woven_plot_loadings()]

Examples

```
data(woven_example)
fit <- woven(woven_example$X_missing,
    Y = woven_example$Y, K = 3L,
    verbose = FALSE
)
woven_top_features(fit, n = 5)
woven_top_features(fit, n = 3, modality = "RNA")
```

woven_tune *Tune woven() by cross-validation*

Description

Picks gamma_y, ridge_w, screen_top (and optionally lambdas) by the balanced error rate of held-out anchor subjects, then fits the model on all data with the best settings. This is the protocol of the WOVEN paper and is what lets the model control its capacity when a modality has many more features than there are anchors.

Usage

```
woven_tune(
  X_list,
  Y,
  K = 5L,
  gamma_y = c(0.5, 1, 5, 10),
  ridge_w = c(1e-08, 0.1, 0.5, 1, 5),
  screen_top = c(Inf, 500, 200, 50),
  lambdas = 0.1,
  k_nn = 10L,
  scale = TRUE,
  n_folds = 3L,
  n_repeats = 1L,
  refit = TRUE,
  verbose = TRUE,
  plot = interactive()
)
```

Arguments

X_list, Y	the data and labels, in any form accepted by [woven()].
K	integer, number of latent dimensions (default 5).
gamma_y	numeric vector of supervision strengths to try.
ridge_w	numeric vector of ridge strengths to try.
screen_top	vector of feature-screening depths to try (Inf means no screening). Values at or above every modality's feature count are the same as Inf and are skipped.
lambdas	numeric vector of Laplacian strengths to try (each used for all modalities; default 0.1 only, since results are usually insensitive to it).
k_nn, scale	as in [woven()].
n_folds	integer, number of cross-validation folds (default 3). It is reduced, with a message, if a class has fewer anchors than folds.
n_repeats	integer, number of times to repeat the cross-validation with new folds (default 1); the BER is averaged over repeats.

refit	logical, fit the full data with the best settings (default TRUE).
verbose	logical, report progress (default TRUE).
plot	logical, draw the tuning results when done (default: in interactive sessions only). Needs ggplot2.

Details

The anchor subjects are split into n_folds class-stratified folds. In each fold everything is refit on the other subjects, including standardization and the similarity graphs, so no held-out subject influences the model that predicts it. Held-out anchors are scored from their features with the fold's projection matrices and classified by linear discriminant analysis on the training anchors' scores (nearest class centroid if the within-class covariance is singular). The balanced error rate (BER) is the mean, over classes, of the fraction of that class misclassified; guessing gives 1 - 1/G.

Subjects missing a modality are kept in every training fold (they shape the similarity graphs and are scored like any other subject) but are not used for evaluation, because a complete-case comparison needs the same subjects for every setting. Fold assignment is random; call set.seed() first for reproducible results. The BER of the selected setting is optimistic, since it was the minimum over the grid; report a separate held-out estimate if you need an unbiased one.

Value

a wovenTune object. tune$best holds the selected settings, tune$results every setting's held-out BER, tune$fit the final model; plot(tune) shows the BER over the grid.

See Also

[woven()]

Examples

```
data(woven_example)
set.seed(1)
tune <- woven_tune(woven_example$X_missing, woven_example$Y,
    K = 3L,
    gamma_y = c(1, 5), ridge_w = c(1e-8, 1), screen_top = c(Inf, 10),
    verbose = FALSE
)
tune
tune$best
fit <- tune$fit
```

---

wovenTune-class *S4 class for a woven_tune() result*

Description

S4 class for a woven_tune() result

Slots

`best` named list: the selected `gamma_y`, `ridge_w`, `screen_top`, `lambda` and their held-out `ber`.

`results` data.frame, one row per setting tried, sorted by `ber`: the settings, the held-out balanced error rate `ber` and its standard deviation across folds `ber_sd`.

`fit` the `woven` fit on all data with the best settings, or `NULL` when `refit = FALSE`.

`folds` list of integer vectors (one per fold and repeat): the held-out subjects (row indices) of each fold.

`chance` numeric, the balanced error rate of guessing (`1 - 1/G` for G classes).

See Also

[woven_tune()]

Index

∗ **datasets**
    woven_example, 17
$,wovenTune-method, 4

anchor_indices *(*woven-accessors*)*, 12
anchor_indices,woven-method
        *(*woven-accessors*)*, 12

latent_scores *(*woven-accessors*)*, 12
latent_scores,woven-method
        *(*woven-accessors*)*, 12

plot,woven,missing-method, 4
plot,wovenTune,missing-method, 6
projection_matrices *(*woven-accessors*)*,
        12
projection_matrices,woven-method
        *(*woven-accessors*)*, 12

show,woven-method, 6
show,wovenTune-method, 7
singular_values *(*woven-accessors*)*, 12
singular_values,woven-method
        *(*woven-accessors*)*, 12
summary,woven-method, 8

woven, *6*, 9, *11–13*, *18*, *34*
woven-accessors, 12
woven-class, 13
woven-package, 3
woven_davies_bouldin, 15
woven_effect_bias, 15
woven_ess_retention, 16
woven_example, 17
woven_mcca_dual, *14*, 18
woven_metrics, *13, 14*, 19
woven_nmi, 20
woven_nystrom_error, 21
woven_plot_coverage, 22
woven_plot_loadings, 23
woven_plot_variance, 24
woven_plot_vip, 25
woven_precompute, 26
woven_predict, *13*, 27
woven_rv, 28
woven_scores, *13, 14*, 29
woven_silhouette, 30
woven_top_features, 31
woven_tune, 32
wovenTune-class, 33