Package: woven (via r-universe)

June 24, 2026

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

Version 0.99.0

Description Supervised multi-omics integration for block-missing (``ragged'') data. WOVEN learns a shared latent space across V omics modalities using only fully-observed anchor subjects, then projects block-missing subjects via Nystrom extension without feature-level imputation. Supervision via label-augmented cross-covariance (analogous to DIABLO) with optional sparse projection matrices (PMD). Designed for comparative effectiveness research where intersection-only methods introduce selection bias.

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, MultipleComparison, GeneExpression, Transcriptomics, Proteomics, Metabolomics, DimensionReduction, Clustering, Classification, Sequencing

Depends R (>= 4.5.0)

Imports Matrix, MatrixGenerics, RANN, cluster, parallel, stats

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

VignetteBuilder knitr

Config/testthat/edition 3

Author Nathan Bresette [aut, cre] (<https://orcid.org/0009-0003-1554-6006>), Ai-Ling Lin [aut], Jianlin Cheng [aut]

Maintainer Nathan Bresette <nathanbresette04@gmail.com>

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

Date/Publication 2026-06-24 19:26:06 UTC

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

RemoteRef HEAD

RemoteSha 3a4f0cc5ab3070cb30eb20dd1ca2efc82553bef0

Contents

plot.woven . . . 2
print.woven . . . 3
summary.woven . . . 4
woven . . . 5
woven_davies_bouldin . . . 6
woven_effect_bias . . . 7
woven_ess_retention . . . 8
woven_example . . . 8
woven_mcca_dual . . . 9
woven_metrics . . . 10
woven_nmi . . . 11
woven_nystrom_error . . . 12
woven_plot_loadings . . . 13
woven_plot_variance . . . 14
woven_plot_vip . . . 15
woven_precompute . . . 16
woven_predict . . . 17
woven_rv . . . 18
woven_scores . . . 18
woven_silhouette . . . 19

Index **20**

`plot.woven` *Plot the WOVEN latent space*

Description

Plots the first two latent dimensions from `fit$Z`, colored by group label. Anchor subjects (complete cases used to learn W) are shown as filled circles; block-missing subjects projected via available views are shown as open triangles. Returns a ggplot object that can be further customized with + layers.

Usage

```
## S3 method for class 'woven'
plot(x, labels = NULL, dims = c(1L, 2L), highlight_anchors = TRUE, ...)
```

Arguments

x	a woven object from [woven()]
labels	integer or factor of length n for coloring points. If NULL, all points are plotted in a single color.
dims	integer vector of length 2: which latent dimensions to plot (default c(1, 2))
highlight_anchors	logical: distinguish anchors from projected subjects via point shape (default TRUE)
...	unused; present for S3 compatibility

Value

a ggplot object, invisibly. The plot is printed as a 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)
plot(fit, labels = Y)
```

---

print.woven *Print method for WOVEN fit*

Description

Print method for WOVEN fit

Usage

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

Arguments

x	a woven object from [woven()]
...	further arguments (unused)

Value

Invisibly returns the woven object x.

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)
print(fit)
```

---

summary.woven *Summarise a WOVEN fit*

Description

Prints a compact metrics table: silhouette, Davies-Bouldin, NMI, and ESS for the scored subjects. Requires class labels.

Usage

```
## S3 method for class 'woven'
summary(object, labels = NULL, ...)
```

Arguments

`object`	a woven object from [woven()]
`labels`	character, factor, or integer vector of length n.
`...`	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, labels = woven_example$Y)
```

`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 Nystrm 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,
  precomp = NULL,
  verbose = TRUE
)
```

Arguments

`X_list`	list of V numeric matrices, each n x p_v. Subjects missing an entire modality should have that matrix row set to NA.
`Y`	integer or factor vector of length n – class labels for all subjects. 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.
`K`	integer – number of latent dimensions (default 5)
`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). Ignored when `precomp` is supplied.
`precomp`	optional output of [woven_precompute()] – pre-built Laplacians. Pass this when calling `woven()` multiple times on the same data (e.g. hyperparameter search, cross-validation) to avoid rebuilding the graph each time.
`verbose`	logical – print progress (default TRUE)

Details

For V=2, uses the closed-form supervised CCA solver (fast, exact). For V>=3, uses the ALS solver with label-kernel supervision.

Value

object of class "woven" with:

$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 class label levels used during fitting

$K, $V, $n dimensions

$lambdas, $gamma_y hyperparameters

See Also

[woven_scores()], [woven_predict()], [woven_all_metrics()]

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(fit$Z) # 60 x 2 -- all subjects scored
```

`woven_davies_bouldin` *Davies-Bouldin index*

Description

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.

Usage

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

Arguments

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

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` *CER-specific: subgroup effect estimate bias*

Description

Fits a linear model of a continuous outcome on a binary treatment indicator, separately within each subgroup defined by ‘labels‘. Compares estimated treatment effect to the known true effect (from simulation ground truth).

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`	named numeric vector, true treatment effect per subgroup level

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 <- 60
Z <- matrix(rnorm(n * 3), n, 3)
outcome <- rnorm(n)
treatment <- rep(0:1, n / 2)
labels <- rep(1:2, each = n / 2)
true_eff <- c(0.5, 1.0)
woven_effect_bias(Z, outcome, treatment, labels, true_eff)
```

---

woven_ess_retention *Effective sample size retention*

Description

Effective sample size retention

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 (DIABLO structurally caps at overlap fraction)

Examples

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

---

woven_example *Example dataset for WOVEN*

Description

A small simulated three-modality dataset (90 subjects) for illustrating package functions. Parameters and label structure are inspired by typical ADNI multi-omics studies (CN / MCI / AD), but all values are synthetic.

Usage

```
woven_example
```

Format

A list with three components:

X_complete List of three matrices: RNA (90 x 25 genes), Methylation (90 x 20 CpGs), Proteomics (90 x 15 proteins). No missing values.

X_missing Same three matrices with ~33% of subjects missing one entire modality block (MCAR).

Y Character vector of 90 class labels: "CN", "MCI", "AD" (30 subjects each).

Examples

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

# Block-missing data — WOVEN retains all 90 subjects
fit_miss <- woven(woven_example$X_missing, Y = woven_example$Y, K = 3L)
woven_metrics(fit_miss, woven_example$Y)
```

---

woven_mcca_dual — *Fit supervised WOVEN for V >= 2 views via dual SUMCOR MCCA (closed-form)*

Description

Single eigendecomposition of a (V*n_a) x (V*n_a) block matrix. No iterations, no random restarts, no local optima. Unified solver for all V.

Usage

```
woven_mcca_dual(
  X_list,
  anchor_idx,
  Y,
  K = 5L,
  lambdas = 0.1,
  gamma_y = 1,
  k_nn = 10L,
  La_list_precomp = NULL,
  verbose = TRUE
)
```

Arguments

`X_list`	list of V matrices, each n x p_v (NA rows = block-missing)
`anchor_idx`	integer vector of fully-observed subject indices
`Y`	vector of length n, class labels (required)

`K`	integer, number of latent dimensions
`lambdas`	numeric scalar or length-V vector, Laplacian regularization
`gamma_y`	numeric >= 0, label supervision strength
`k_nn`	integer, k-NN for Laplacian (ignored if La_list_precomp supplied)
`La_list_precomp`	optional list of pre-extracted n_a x n_a anchor Laplacians
`verbose`	logical

Value

list with W_list, Za_list, Xa_list, singular_values, and metadata. Compatible with project_all() in benchmark_one_rep.R.

Examples

```
set.seed(1)
n <- 20
K <- 2L
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)
anchor_idx <- seq_len(14L)
fit <- woven_mcca_dual(list(X1, X2, X3), anchor_idx = anchor_idx, Y = Y, K = K)
length(fit$W_list)
```

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

Description

Calls [woven_all_metrics()] using `fit$Z` and `fit$n` so you do not need to extract them manually. Returns silhouette, Davies-Bouldin, NMI, and ESS retention as a named numeric vector.

Usage

```
woven_metrics(fit, labels, ...)
```

Arguments

`fit`	woven object from [woven()]
`labels`	integer, factor, or character vector of length n with subgroup labels for all subjects (same Y passed to [woven()])
`...`	additional arguments forwarded to [woven_all_metrics()]

Value

named numeric vector of metric values, printed as a tidy table

See Also

[woven_all_metrics()], [woven_silhouette()], [woven_nmi()]

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)
woven_metrics(fit, Y)
```

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

Description

Uses k-means on Z to get cluster assignments, then computes NMI. k-means run 10 times to reduce initialization variance.

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

Value

scalar in [0, 1], higher is better

Examples

```
set.seed(1)
Z <- matrix(rnorm(40 * 2), 40, 2)
labels <- rep(1:2, each = 20)
woven_nmi(Z, labels)
```

---

woven_nystrom_error *Leave-anchor-out Nystrm projection error*

Description

For each held-out anchor subject, refits WOVEN without it, projects via direct W scoring, and computes ||Z_proj - Z_direct||. Quantifies how well the projection generalizes across anchor subsets.

Usage

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

Arguments

fit	a woven object from [woven()]
X_list	list of complete (no block-missing) modality matrices, same structure as passed to [woven()]
n_loo	integer, number of anchors to hold out (default min(20, n_a))
sigma_proj	unused, kept for compatibility

Value

scalar >= 0, lower is better (mean Frobenius error per anchor)

Examples

```
data(woven_example)
fit <- woven(woven_example$X_complete, Y = woven_example$Y, K = 3L)
woven_nystrom_error(fit, woven_example$X_complete, n_loo = 5L)
```

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`	integer or NULL: if specified, plot only that modality. 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 variance explained per WOVEN latent dimension*

Description

Bar chart of the proportion of variance explained per latent dimension (proportional to squared singular values), overlaid with a cumulative variance curve on a secondary axis. Use this to choose K and to show how much shared multi-omics signal is captured in the leading dimensions.

Usage

```
woven_plot_variance(fit, main = "Variance Explained")
```

Arguments

`fit`	woven object from [woven()]
`main`	character: plot title (default "Variance Explained")

Value

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

See Also

[woven_plot_vip()], [woven_plot_loadings()]

Examples

```
set.seed(1)
n <- 60; p1 <- 30; p2 <- 20; K <- 4
Y <- rep(1:2, each = n / 2)
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_variance(fit)
```

woven_plot_vip — *Plot VIP scores for a WOVEN modality*

Description

Displays the top features by Variable Importance in Projection (VIP) score for one modality. VIP scores weight each feature's loading across all K latent dimensions by the variance explained per dimension, producing a single importance ranking analogous to DIABLO's contribution plot. A dashed reference line at VIP = 1 marks above-average importance.

Usage

```
woven_plot_vip(
  fit,
  modality = 1L,
  n_top = 20L,
  feature_names = NULL,
  main = NULL
)
```

Arguments

`fit`	woven object from [woven()]
`modality`	integer: which modality to plot (1..V, default 1)
`n_top`	integer: number of top features to display (default 20)
`feature_names`	optional character vector of length p_v with feature labels. If NULL, uses rownames of W_list[[modality]] or "Feature_j".
`main`	character: plot title. If NULL, a default is generated.

Value

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

See Also

[woven_plot_loadings()], [woven_plot_variance()], [woven_vip()]

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_vip(fit, modality = 1L)
```

---

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 – useful for hyperparameter search, cross-validation, or sensitivity analysis.

Usage

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

Arguments

`X_list`	list of V matrices (n x p_v). Block-missing rows (all NA) are automatically excluded from the k-NN graph.
`k_nn`	integer, number of nearest neighbours (default 10)

Value

list of V sparse Laplacian matrices, one per modality. Pass directly to the `precomp` argument of [woven()].

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 and returns soft class assignments using a nearest-centroid classifier in latent space. Works for complete subjects (direct projection) and block-missing subjects (Nystrm).

Usage

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

Arguments

`fit`	woven object from [woven()]
`X_list_new`	list of V matrices for new subjects (n_new x p_v each). Block-missing subjects should have their modality rows set to NA.
`method`	"centroid" (default) – nearest centroid in latent space. "knn" – k-NN vote using anchor subjects as the reference set.
`k_pred`	integer – number of neighbors for knn method (default 5)

Value

data.frame with n_new rows: $predicted_class integer predicted class label $confidence probability of predicted class (0-1) One column per class level with soft 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

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.

Usage

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

Arguments

`Z`	numeric matrix n x K (inferred latent scores)
`Z_true`	numeric matrix n x K_true (ground-truth factor scores from SUMO)

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. Uses direct linear projection (x %*% W_v) for each available modality, then averages across observed views.

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

Usage

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

Arguments

`fit`	woven object from `woven()`
`X_list_new`	list of V matrices (n_new x p_v). Set entire rows to NA for subjects missing that modality block. Every subject must have at least one non-missing view.

Value

Numeric matrix n_new x K of consensus latent scores. 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

```
# minimal example data (n=20, 14 anchors, 6 partial)
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)
# Rows 15-20: alternate missing view 1 or view 2 (never both)
miss <- matrix(FALSE, n, 2)
miss[c(15,17,19), 1] <- TRUE   # miss view 1
miss[c(16,18,20), 2] <- TRUE   # miss view 2
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)
dim(woven_scores(fit, list(X1,X2)))
```

---

woven_silhouette *Average silhouette width*

Description

Average silhouette width

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)
```

Index

∗ **datasets**
woven_example, 8

plot.woven, 2
print.woven, 3

summary.woven, 4

woven, 5, *18, 19*
woven_davies_bouldin, 6
woven_effect_bias, 7
woven_ess_retention, 8
woven_example, 8
woven_mcca_dual, 9
woven_metrics, 10
woven_nmi, 11
woven_nystrom_error, 12
woven_plot_loadings, 13
woven_plot_variance, 14
woven_plot_vip, 15
woven_precompute, 16
woven_predict, 17, *18, 19*
woven_rv, 18
woven_scores, 18
woven_silhouette, 19