

Package: woven (via r-universe)

September 18, 2026

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

Version 0.99.3

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 enters through a label-augmented cross-covariance (analogous to DIABLO), and projection matrices can optionally be ridge-regularized and univariately feature-screened for high-dimensional modalities. 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, BiocParallel, methods, stats

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

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 2026-09-18 19:02:00 UTC
RemoteUrl <https://github.com/BiocStaging/woven>
RemoteRef HEAD
RemoteSha 5f9ecfb94bc2ecf16a1ebd2810facc3d748b13ea

Contents

anchorIdx	2
plot,woven,missing-method	3
show,woven-method	4
singularValues	5
summary,woven-method	5
WList	6
woven	7
woven-class	9
woven_davies_bouldin	11
woven_effect_bias	11
woven_ess_retention	12
woven_example	13
woven_mcca_dual	13
woven_metrics	15
woven_nmi	16
woven_nystrom_error	17
woven_plot_loadings	17
woven_plot_variance	19
woven_plot_vip	20
woven_precompute	21
woven_predict	22
woven_rv	23
woven_scores	23
woven_silhouette	24
Z	25
Index	26

anchorIdx	<i>Indices of the anchor (fully-observed) subjects</i>
-----------	--

Description

Indices of the anchor (fully-observed) subjects

Usage

```
anchorIdx(x)

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

Arguments

x a *woven* object

Value

integer vector

Examples

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

plot,woven,missing-method

Plot the WOVEN latent space

Description

Plots the first two latent dimensions from $Z(\text{fit})$, 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

```
## S4 method for signature 'woven,missing'
plot(x, y, labels = NULL, dims = c(1L, 2L), highlight_anchors = TRUE, ...)
```

Arguments

x	a woven object from [woven()]
y	unused; present for S4 ‘plot’ signature compatibility
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

Value

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

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)
```

show,woven-method	<i>Show method for WOVEN fit</i>
-------------------	----------------------------------

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

```

singularValues

*Supervised canonical correlations***Description**

Supervised canonical correlations

Usage

```

singularValues(x)

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

```

Arguments

x a *woven* object

Value

numeric vector of length K

Examples

```

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

```

summary,woven-method

*Summarise a WOVEN fit***Description**

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

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.
...	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)
```

WList

Per-modality projection matrices

Description

Per-modality projection matrices

Usage

```
WList(x)

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

Arguments

x	a woven object
---	--------------------------------

Value

named list of V numeric matrices, each $p_v \times K$

Examples

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

Arguments

<code>X_list</code>	either a named list of V numeric matrices, each $n \times p_v$, with subjects missing an entire modality having that matrix row set to NA, or a <code>MultiAssayExperiment</code> . When a <code>MultiAssayExperiment</code> is supplied, each experiment becomes one modality, subjects are aligned to <code>colData(X_list)</code> via <code>sampleMap(X_list)</code> , and a subject not present in a given experiment gets an all-NA row for it automatically.
<code>Y</code>	integer or factor vector of length n – class labels for all subjects, in the same subject order as <code>X_list</code> (or as <code>colData(X_list)</code> when <code>X_list</code> is a <code>MultiAssayExperiment</code>). A single string is instead treated as the name of a <code>colData</code> column to use as labels, but only when <code>X_list</code> is a <code>MultiAssayExperiment</code> . Only anchor subjects' labels enter the supervised objective.
<code>anchor_idx</code>	integer vector – indices of fully-observed subjects (observed in all V modalities). Must have length $\geq K$. If NULL (default), anchors are detected automatically as subjects with no block-missing modalities.
<code>K</code>	integer – number of latent dimensions (default 5)
<code>lambdas</code>	numeric scalar or length- V vector – Laplacian regularization strength per modality (default 0.1 for all)

<code>gamma_y</code>	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.
<code>k_nn</code>	integer – k-nearest-neighbors for Laplacian graph (default 10).
<code>ridge_w</code>	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 $p_v \gg n_a$ and the true signal is diffuse across many features – select via cross-validation on held-out classification error, since silhouette is insensitive to this parameter even where it materially changes classification accuracy.
<code>screen_top</code>	integer or Inf – restricts W recovery to the top <code>screen_top</code> 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 <code>ridge_w</code> ; both target overfitting when p_v is large relative to n_a . Select jointly with <code>ridge_w</code> via cross-validation, not independently (the two interact).
<code>scale</code>	logical – standardize each modality’s features to zero mean and unit variance before fitting (default TRUE), matching DIABLO (<code>scale=TRUE</code>) and IntegrAO. Apply biological transforms (CLR/log/VST) upstream. The center and scale are stored and reapplied to new subjects in <code>[woven_scores()]</code> . Ignored when <code>precomp</code> is supplied.
<code>precomp</code>	optional output of <code>[woven_precompute()]</code> – pre-built Laplacians. Pass this when calling <code>woven()</code> multiple times on the same data (e.g. hyperparameter search, cross-validation) to avoid rebuilding the graph each time.
<code>verbose</code>	logical – print progress (default TRUE)

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 $V \geq 3$ 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 $n \times K$ 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 $p_v \times K$

Z_anchors list of V anchor latent score matrices, each $n_a \times K$

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 `woven` for the complete slot list.

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

# Equivalently, from a MultiAssayExperiment: each experiment is one
# modality, and subjects missing an experiment need no special handling.
if (requireNamespace("MultiAssayExperiment", quietly = TRUE)) {
  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(Z(fit_mae)) # same 60 x 2 -- all subjects scored
}
```

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, $n \times K$ 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 $p_v \times K$, the per-modality projection matrices.

`Z_anchors` list of V numeric matrices, each $n_a \times K$, 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` labels of the anchor subjects, in `anchor_idx` order.

`Y_levels` sorted unique class labels seen during fitting.

`Y_labels` optional original label names (e.g. "CN"), or NULL if Y was already numeric/integer.

`K` integer, latent dimension.

`V` integer, number of modalities.

`n` integer, total number of subjects.

`mod_names` optional character vector of modality names, or NULL if `X_list` was unnamed.

`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.

See Also

[\[woven\(\)\]](#)

woven_davies_bouldin *Davies-Bouldin index*

Description

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

Usage

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

Arguments

`Z` numeric matrix $n \times K$
`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

$\text{bias_g} = \text{lestimed_g} - \text{true_gl} / |\text{true_gl}|$ (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 $V \geq 2$ views via dual SUMCOR MCCA (closed-form)

Description

Single eigendecomposition of a $(V \times n_a) \times (V \times n_a)$ 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,
  ridge_w = 1e-08,
  screen_top = Inf,
  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
ridge_w	numeric >= 0, relative ridge strength for the W recovery step ($Ka_{reg} = Ka_v + ridge_w * \max(diag(Ka_v)) * I$). Default 1e-8 is a pure numerical-stability nudge (original behavior). Larger values shrink W_v toward zero, trading anchor fit for reduced variance when $p_v \gg n_a$ (v7 regularization; CV-selected like lambdas and gamma_y).
screen_top	integer or Inf, per-modality feature count for univariate (ANOVA F-score vs Y) supervised pre-screening BEFORE the W recovery step. Default Inf disables this (all features used, original behavior). The eigendecomposition (Za_list) never depends on features – only the Laplacians and label kernel – so screening only changes the $W_v = X_{a_v}^T Ka_v^{-1} Za_v$ recovery: $Ka_v = X_{a_v} X_{a_v}^T$ is built from the screened columns only, which is a genuine re-fit of the min-norm interpolation on a reduced feature set (not a post-hoc truncation of an already-complete solution). Screened-out features get an exact zero row in the returned W_v (same shape as unscreened, so all downstream code – <code>project_all()</code> , <code>woven_scores()</code> – is unaffected).
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 `Z(fit)` and the fit's subject count 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

<code>fit</code>	woven object from [woven()]
<code>labels</code>	integer, factor, or character vector of length <code>n</code> with subgroup labels for all subjects (same <code>Y</code> passed to [woven()])
<code>...</code>	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	<i>Normalized mutual information between cluster assignments and true labels</i>
-----------	--

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 $n \times K$
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	<i>Leave-anchor-out Nystrom projection error</i>
---------------------	--

Description

For each held-out anchor subject, refits WOVEN without it, projects via direct W scoring, and computes $\|Z_{\text{proj}} - Z_{\text{direct}}\|$. 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	<i>Plot feature loadings for one WOVEN latent dimension</i>
---------------------	---

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

<code>fit</code>	woven object from <code>[woven()]</code>
<code>dim</code>	integer: which latent dimension to plot (1..K, default 1)
<code>n_top</code>	integer: number of top features per modality (default 15)
<code>feature_names</code>	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".
<code>modality</code>	integer or NULL: if specified, plot only that modality. If NULL (default), all V modalities are shown in faceted panels.
<code>main</code>	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	<i>Plot variance explained per WOVEN latent dimension</i>
---------------------	---

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	<i>Plot VIP scores for a WOVEN modality</i>
----------------	---

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 row-names 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, scale = TRUE)
```

Arguments

<code>X_list</code>	list of V matrices (n x p_v). Block-missing rows (all NA) are automatically excluded from the k-NN graph.
<code>k_nn</code>	integer, number of nearest neighbours (default 10)
<code>scale</code>	logical, standardize features to unit variance before building Laplacians (default TRUE); use the same value passed to <code>[woven()]</code> .

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 (Nystrom).

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	<i>RV coefficient between latent scores and ground-truth factor matrix</i>
----------	--

Description

$RV(X, Y) = \text{trace}(X X' Y Y') / \sqrt{\text{trace}(X X' X X') * \text{trace}(Y Y' Y Y')}$ 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	<i>Extract latent scores for new subjects</i>
--------------	---

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 $n_{\text{new}} \times K$ 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 $n \times K$ (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)
```

Z	<i>Consensus latent scores</i>
---	--------------------------------

Description

Consensus latent scores

Usage

```
Z(x)
```

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

Arguments

x a [woven](#) object

Value

numeric matrix, n x K

Examples

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

Index

* datasets

- woven_example, [13](#)
- anchorIdx, [2](#)
- anchorIdx, woven-method (anchorIdx), [2](#)
- plot, woven, missing-method, [3](#)
- show, woven-method, [4](#)
- singularValues, [5](#)
- singularValues, woven-method (singularValues), [5](#)
- summary, woven-method, [5](#)
- WList, [6](#)
- WList, woven-method (WList), [6](#)
- woven, [3–6](#), [7](#), [8](#), [9](#), [23–25](#)
- woven-class, [9](#)
- woven_davies_bouldin, [11](#)
- woven_effect_bias, [11](#)
- woven_ess_retention, [12](#)
- woven_example, [13](#)
- woven_mcca_dual, [10](#), [13](#)
- woven_metrics, [9](#), [15](#)
- woven_nmi, [16](#)
- woven_nystrom_error, [17](#)
- woven_plot_loadings, [17](#)
- woven_plot_variance, [19](#)
- woven_plot_vip, [20](#)
- woven_precompute, [21](#)
- woven_predict, [9](#), [22](#), [23](#), [24](#)
- woven_rv, [23](#)
- woven_scores, [9](#), [10](#), [23](#)
- woven_silhouette, [24](#)
- Z, [25](#)
- Z, woven-method (Z), [25](#)