Package 'sclValid'

September 12, 2026

Title Ensemble Validation and Ranking of Clustering Methods

Version 0.1.0

Description Provides tools for clustering, validation, and ranking of single-cell RNA sequencing data. The package implements multiple clustering approaches, a collection of internal, external, and stability-based validation measures, and rank aggregation for comparing clustering solutions. User-defined clustering functions can also be incorporated into the validation framework. Methods are based in part on Visser and Datta (2025) <doi:10.1002/sim.70331>.

License GPL (>= 3)

Encoding UTF-8

Config/roxygen2/version 8.1.0

Depends R (>= 3.5)

LazyData true

Imports cccd, cli, clue, igraph, mclust, mnormt, pcaMethods, RaceID, RankAggreg, Rcpp, Rtsne, S4Vectors, SingleCellExperiment, SummarizedExperiment

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Owen Visser [aut, cre],
Justina Žurauskienė [ctb] (pcaReduce compatibility code adapted from the pcaReduce package),
Christopher Yau [ctb] (pcaReduce compatibility code adapted from the pcaReduce package)

Maintainer Owen Visser <owviss@gmail.com>

Repository CRAN

Date/Publication 2026-09-12 09:40:08 UTC

Contents

`check_input` *Check Input Data*

Description

Checks whether an input matrix and corresponding label vector are suitable for use with sclValid. The matrix must be numeric, finite, and contain no missing values. Rows represent features and columns represent observations. The label vector must contain exactly one label for each column of the matrix.

Usage

```
check_input(x, labels)
```

Arguments

`x`	A numeric matrix with features in rows and observations in columns.
`labels`	A vector or factor containing one label for each column of `x`.

Value

Invisibly returns `TRUE` if all input checks are passed.

Examples

```
check_input(example_data, example_labels)
```

example_data *Example Single-Cell Expression Data*

Description

A reduced example expression matrix derived from the single-cell RNA-seq dataset reported by Biase et al. The original dataset contains cell identities established in that study and is available from the NCBI Gene Expression Omnibus under accession GSE57249.

Usage

```
example_data
```

Format

A numeric matrix with 1000 rows and 49 columns.

Details

For use in sclValid examples, 1000 rows were randomly sampled from the original expression matrix while retaining all 49 observations. Rows represent features and columns represent cells.

Source

Biase et al. single-cell RNA-seq dataset, NCBI Gene Expression Omnibus, accession GSE57249.

example_labels *Example Cell Labels*

Description

Cell identity labels corresponding to the 49 columns of `example_data`. The labels are derived from the cell identities established in the Biase et al. study associated with GEO accession GSE57249.

Usage

```
example_labels
```

Format

A character vector of length 49.

Details

For use in sclValid, the original cell groups were relabeled as `"A"`, `"B"`, and `"C"`.

Source

Biase et al. single-cell RNA-seq dataset, NCBI Gene Expression Omnibus, accession GSE57249.

`make_sce` *Create a SingleCellExperiment Object*

Description

Creates a `SingleCellExperiment` object from a numeric feature matrix and a corresponding vector of observation labels. The input matrix is stored as the `"data"` assay and the supplied labels are stored in `colData`.

Usage

```
make_sce(x, labels)
```

Arguments

`x`	A numeric matrix with features in rows and observations in columns.
`labels`	A vector or factor containing one label for each column of `x`.

Value

A `SingleCellExperiment` object containing the input matrix and observation labels.

Examples

```
sce <- make_sce(example_data, example_labels)
sce
```

`run_custom_clustering` *Run a User-Defined Clustering Method*

Description

Applies a user-supplied clustering function to the `"data"` assay of a `SingleCellExperiment` object. The clustering function must return one cluster assignment for each observation in the dataset.

Usage

```
run_custom_clustering(sce, fun, name, ...)
```

Arguments

sce	A SingleCellExperiment object containing a "data" assay.
fun	A clustering function. The first argument supplied to fun is the numeric data matrix with features in rows and observations in columns. The function must return one cluster assignment per observation.
name	Character string specifying the name of the clustering solution.
...	Additional arguments passed to fun.

Details

The clustering assignments are stored in colData using the name supplied by name. Information required to rerun the clustering method is retained internally so that stability-based validation measures can reproduce the clustering on perturbed datasets.

Value

The input SingleCellExperiment object with the custom clustering assignments added to colData.

Examples

```
sce <- make_sce(example_data, example_labels)

my_kmeans <- function(data, centers) {
  stats::kmeans(t(data), centers = centers)$cluster
}

sce <- run_custom_clustering(
  sce,
  fun = my_kmeans,
  name = "MyKmeans_cs3",
  centers = 3
)

SummarizedExperiment::colData(sce)
```

---

run_pcareduce *Run pcaReduce Clustering*

Description

Applies the pcaReduce clustering procedure to the data assay of a SingleCellExperiment object for one or more requested numbers of clusters. Cluster assignments are added to the object's colData.

Usage

```
run_pcareduce(sce, cluster_sizes, method = "M", nbt = 100, verbose = TRUE)
```

Arguments

sce	A SingleCellExperiment object containing a "data" assay.
cluster_sizes	An integer vector specifying the numbers of clusters to fit.
method	Character string specifying the pcaReduce merging method. Must be either "M" or "S".
nbt	Number of pcaReduce iterations to perform. Defaults to 100.
verbose	Logical indicating whether detailed progress messages should be displayed during clustering‘.

Details

The pcaReduce method may be run using either the "M" or "S" merging procedure. For a cluster size k, resulting columns are named "pcaReduceM_cs{k}" or "pcaReduceS_cs{k}", depending on the selected method.

Value

The input SingleCellExperiment object with additional pcaReduce cluster-assignment columns in colData.

References

Žurauskienė J, Yau C. pcaReduce: Hierarchical clustering of single cell transcriptional profiles. BMC Bioinformatics. 2016;17:140.

Examples

```
sce <- make_sce(example_data, example_labels)
sce <- run_pcareduce(
  sce,
  cluster_sizes = 2:4,
  method = "M"
)
SummarizedExperiment::colData(sce)
```

---

run_raceid *Run RaceID Clustering*

Description

Applies RaceID clustering to the data assay of a SingleCellExperiment object for one or more requested numbers of clusters. Cluster assignments are added to the object's colData.

Usage

```
run_raceid(sce, cluster_sizes, verbose = TRUE)
```

Arguments

sce	A SingleCellExperiment object containing a "data" assay.
cluster_sizes	An integer vector specifying the numbers of clusters to fit.
verbose	Logical indicating whether detailed progress messages should be displayed during clustering‘.

Details

If feature or observation names are absent from the input matrix, temporary names are generated internally because RaceID requires them.

For a cluster size k, the resulting column is named "RaceID_cs{k}".

Value

The input SingleCellExperiment object with additional RaceID cluster-assignment columns in colData.

References

Grün D, Lyubimova A, Kester L, et al. Single-cell messenger RNA sequencing reveals rare intestinal cell types. Nature. 2015;525:251–255.

Examples

```
sce <- make_sce(example_data, example_labels)
sce <- run_raceid(sce, cluster_sizes = 2:4)
SummarizedExperiment::colData(sce)
```

---

run_rank_aggregation *Rank Clustering Solutions*

Description

Aggregates scaled clustering validation measures to produce an overall ranking of clustering solutions.

Usage

```
run_rank_aggregation(sce, measures = NULL, importance = NULL, seed = 0)
```

Arguments

sce	A SingleCellExperiment containing scaled validation results produced by scale_validation().
measures	Optional character vector specifying validation measures to include in the aggregation. If NULL, all validation measures available in the scaled validation table are used.
importance	Optional numeric vector specifying the relative importance of the selected validation measures. If NULL, all measures receive equal importance.
seed	Integer random seed supplied to RankAggreg.

Details

Rank aggregation is performed in two stages. First, clustering solutions are compared separately within each cluster size, and the highest-ranked solution is retained for each size. The retained solutions are then aggregated across cluster sizes to produce the final ranking.

Validation scores are obtained from S4Vectors::metadata(sce)$validation_scaled. All scores are assumed to have been scaled so that larger values indicate better clustering performance.

Value

The input SingleCellExperiment with the final aggregated ranking stored in metadata(sce)$ranking.

References

Pihur V, Datta S, Datta S. Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. Bioinformatics. 2007;23(13):1607-1615.

Pihur V, Datta S, Datta S. RankAggreg, an R package for weighted rank aggregation. BMC Bioinformatics. 2009;10:1-10.

Examples

```
sce <- make_sce(example_data, example_labels)

sce <- run_raceid(
  sce,
  cluster_sizes = 2:3
)

sce <- run_validation(
  sce,
  measures = c("ARI", "BHI", "DI", "IGP", "SW"),
  verbose = FALSE,
  progress = FALSE
)

sce <- scale_validation(sce)

sce <- run_rank_aggregation(sce)
```

```
S4Vectors::metadata(sce)$ranking
```

`run_reduction_clustering` *Run Clustering on PCA and t-SNE Representations*

Description

Performs dimensionality reduction followed by one or more clustering procedures on the `"data"` assay of a `SingleCellExperiment` object. Principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) representations are computed only when requested and are reused across clustering procedures to avoid repeated dimensionality reduction and distance calculations.

Usage

```
run_reduction_clustering(
  sce,
  reductions = c("PCA", "TSNE"),
  methods = c("kmeans", "hierarchical", "louvain", "leiden"),
  cluster_sizes = 2:4,
  knn = c(10, 25, 40),
  resolution = seq(0.1, 0.9, 0.1),
  pca_dims = 50,
  verbose = TRUE
)
```

Arguments

`sce`	A `SingleCellExperiment` object containing a `"data"` assay.
`reductions`	Character vector specifying the dimensionality reductions to use. Supported values are `"PCA"` and `"TSNE"`.
`methods`	Character vector specifying the clustering methods to apply. Supported values are `"kmeans"`, `"hierarchical"`, `"louvain"`, and `"leiden"`.
`cluster_sizes`	An integer vector specifying the numbers of clusters used for K-means and hierarchical clustering.
`knn`	An integer vector specifying the numbers of nearest neighbors used to construct graphs for Louvain and Leiden clustering.
`resolution`	A numeric vector specifying resolution values for Louvain and Leiden clustering.
`pca_dims`	Maximum number of principal components to retain. The actual number is limited by the dimensions of the input data. Defaults to `50`.
`verbose`	Logical indicating whether detailed progress messages should be displayed during clustering‘.

Details

Supported clustering methods include K-means, hierarchical clustering, Louvain community detection, and Leiden community detection. Cluster assignments are added to the object's `colData`.

K-means and hierarchical clustering results are named according to the dimensionality reduction and requested number of clusters. For example, `"PCA_Kmeans_cs3"` contains PCA-based K-means assignments for three clusters.

Louvain and Leiden results include both the nearest-neighbor and resolution parameters in their column names, for example `"PCA_Louvain_knn10_res0.5"`.

When t-SNE is requested, PCA distances are first calculated and used as the input distance representation for t-SNE.

Value

The input `SingleCellExperiment` object with additional cluster-assignment columns in `colData`.

Examples

```
sce <- make_sce(example_data, example_labels)

sce <- run_reduction_clustering(
  sce,
  reductions = c("PCA", "TSNE"),
  methods = c("kmeans", "hierarchical"),
  cluster_sizes = 2:4
)

SummarizedExperiment::colData(sce)
```

`run_validation` *Calculate Clustering Validation Measures*

Description

Calculates internal, external, and stability-based validation measures for clustering solutions stored in the `colData` of a `SingleCellExperiment`.

Usage

```
run_validation(
  sce,
  measures = c("AD", "ADM", "APN", "ARI", "BHI", "BSI", "CN", "DI", "IGP", "SW"),
  clusterings = NULL,
  index = NULL,
  nn_k = 50,
  connectivity_h = 5,
  verbose = FALSE,
```

```
  progress = TRUE
)
```

Arguments

sce	A SingleCellExperiment containing a "data" assay and one or more clustering-assignment columns in colData.
measures	Character vector specifying validation measures to calculate. Supported values are "AD", "ADM", "APN", "ARI", "BHI", "BSI", "CN", "DI", "IGP", and "SW".
clusterings	Optional character vector specifying particular clustering columns in colData to validate. If NULL, all recognized clustering solutions are used.
index	Optional list specifying observations removed for stability calculations. If NULL, leave-one-observation-out removal is used.
nn_k	Number of nearest neighbors to calculate when a requested validation measure requires nearest-neighbor information. Defaults to 50.
connectivity_h	Number of nearest neighbors used by the connectivity measure. Defaults to 5.
verbose	Logical indicating whether detailed progress messages should be displayed during validation. Defaults to FALSE.
progress	Logical indicating whether a progress bar should be displayed while reduced clustering solutions are generated for stability-based validation measures. Defaults to TRUE.

Details

Clustering solutions are detected automatically from columns produced by the built-in clustering functions in sclValid, as well as from user-defined clustering methods registered with run_custom_clustering(). This allows clustering methods not implemented directly in sclValid to be evaluated using the same validation framework. A subset of clustering solutions may instead be selected with clusterings.

Shared quantities such as pairwise distances and nearest-neighbor lists are calculated once and reused across validation measures. When stability measures are requested, reduced datasets are reclustered internally and the resulting reduced clustering solutions are used for comparison with the full-data solutions.

Validation results are stored in S4Vectors::metadata(sce)$validation, with one row per clustering solution and one column per requested validation measure.

ARI, BHI, and BSI require known observation labels stored in the "label" column of colData. If complete labels are not available, these measures are skipped while validation measures that do not require known labels may still be calculated.

AD, ADM, APN, and BSI require reduced clustering solutions and therefore trigger reclustering of reduced versions of the dataset.

User-defined clustering methods may be registered with run_custom_clustering(). The supplied clustering function should accept the expression matrix as its first argument, with features in rows and observations in columns, and must return one cluster assignment per observation. Additional arguments may be supplied through run_custom_clustering().

A general custom clustering workflow is:

```
my_clustering <- function(data, ...) {
  clusters <- clustering_method(data, ...)
  clusters
}

sce <- run_custom_clustering(
  sce,
  fun = my_clustering,
  name = "MyClustering"
)

sce <- run_validation(sce)
```

The custom function and its supplied arguments are retained internally so that stability-based validation measures can rerun the same clustering procedure on reduced datasets.

By default, `run_validation()` displays a progress bar during reduced reclustering while suppressing detailed clustering messages. Set `progress = FALSE` to disable the progress bar, or `verbose = TRUE` to display additional status messages. The progress bar is only displayed for stability-based validation measures that require reduced clustering solutions.

Value

The input `SingleCellExperiment` with validation results added to `metadata(sce)$validation`.

Examples

```
sce <- make_sce(example_data, example_labels)
sce <- run_raceid(sce, cluster_sizes = 2:3)

sce <- run_validation(
  sce,
  measures = c("ARI", "BHI", "DI", "IGP", "SW")
)

S4Vectors::metadata(sce)$validation
```

---

scale_validation *Scale Clustering Validation Measures*

Description

Scales clustering validation measures to the interval from zero to one so that larger values consistently represent better clustering performance.

Usage

```
scale_validation(sce)
```

Arguments

sce	A SingleCellExperiment containing validation results produced by run_validation().

Details

Validation measures for which smaller values indicate better performance are reversed after scaling. Before scaling, numeric values are bounded to the interval from -1e6 to 1e6.

Raw validation scores stored in metadata(sce)$validation are not modified. The scaled results are stored separately in metadata(sce)$validation_scaled.

The measures DI, IGP, SW, ARI, BSI, and BHI are treated as measures for which larger values indicate better performance.

AD, ADM, APN, and CN are treated as measures for which smaller values indicate better performance and are reversed after scaling.

Value

The input SingleCellExperiment with scaled validation results added to metadata(sce)$validation_scaled.

Examples

```
sce <- make_sce(example_data, example_labels)
sce <- run_raceid(sce, cluster_sizes = 2:3)
sce <- run_validation(sce, measures = c("ARI", "DI", "IGP", "SW"))
sce <- scale_validation(sce)

S4Vectors::metadata(sce)$validation_scaled
```

---

validation_apn *Calculate Average Proportion of Non-Overlap*

Description

Calculates an adapted Average Proportion of Non-Overlap (APN) stability measure by comparing cluster memberships between full and perturbed clustering solutions.

Usage

```
validation_apn(clusters, reduced_clusters, index)
```

Arguments

clusters	A vector containing cluster assignments obtained from the full dataset.
reduced_clusters	A list of clustering-assignment vectors obtained from the reduced datasets.
index	A list specifying the observations removed for each reduced clustering solution.

Details

The implementation is adapted from the clustering stability measures described by Datta and Datta.

Value

A numeric average proportion of non-overlap. Lower values indicate more stable clustering.

References

Adapted from: Datta S, Datta S. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics. 2003;19(4):459-466.

Examples

```
clusters <- c(1, 1, 2, 2)

reduced_clusters <- list(
  c(1, 2, 2),
  c(1, 2, 2),
  c(1, 1, 2),
  c(1, 1, 2)
)

index <- as.list(1:4)

validation_apn(
  clusters,
  reduced_clusters,
  index
)
```

---

validation_ari *Calculate Adjusted Rand Index*

Description

Calculates the adjusted Rand index between a clustering solution and known observation labels. The adjusted Rand index measures agreement between two partitions while correcting for agreement expected by chance.

Usage

```
validation_ari(clusters, labels)
```

Arguments

clusters	A vector containing one cluster assignment per observation.
labels	A vector containing the known label for each observation.

Value

A numeric adjusted Rand index.

References

Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985;2:193-218.

Examples

```
clusters <- c(1, 1, 2, 2)
labels <- c("A", "A", "B", "B")
validation_ari(clusters, labels)
```

`validation_average_distance`
Calculate Average Distance

Description

Calculates an adapted Average Distance (AD) stability measure by comparing clustering solutions obtained from the full dataset with corresponding solutions obtained after perturbing the dataset.

Usage

```
validation_average_distance(clusters, reduced_clusters, distmat, index)
```

Arguments

`clusters`	A vector containing cluster assignments obtained from the full dataset.
`reduced_clusters`	A list of clustering-assignment vectors obtained from the reduced datasets.
`distmat`	A numeric matrix containing pairwise distances between observations in the full dataset.
`index`	A list specifying the observations removed for each reduced clustering solution.

Details

The implementation is adapted from the clustering stability measures described by Datta and Datta.

Value

A numeric average-distance stability score. Lower values indicate greater agreement between full and perturbed clustering solutions.

References

Adapted from: Datta S, Datta S. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics. 2003;19(4):459-466.

Examples

```
clusters <- c(1, 1, 2, 2)
x <- matrix(
  c(
    0, 0, 3, 3,
    0, 1, 3, 4
  ),
  nrow = 2,
  byrow = TRUE
)

reduced_clusters <- list(
  c(1, 2, 2),
  c(1, 2, 2),
  c(1, 1, 2),
  c(1, 1, 2)
)

index <- as.list(1:4)
distmat <- as.matrix(stats::dist(t(x)))

validation_average_distance(
  clusters,
  reduced_clusters,
  distmat,
  index
)
```

---

validation_average_distance_means

Calculate Average Distance Between Means

Description

Calculates an adapted Average Distance Between Means (ADM) stability measure based on distances between cluster centroids in full and perturbed clustering solutions.

Usage

```
validation_average_distance_means(clusters, reduced_clusters, data, index)
```

Arguments

clusters	A vector containing cluster assignments obtained from the full dataset.
reduced_clusters	A list of clustering-assignment vectors obtained from the reduced datasets.
data	A numeric matrix with features in rows and observations in columns.
index	A list specifying the observations removed for each reduced clustering solution.

Details

The implementation is adapted from the clustering stability measures described by Datta and Datta.

Value

A numeric average-distance-between-means stability score. Lower values indicate greater stability in cluster locations.

References

Adapted from: Datta S, Datta S. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics. 2003;19(4):459-466.

Examples

```
clusters <- c(1, 1, 2, 2)
x <- matrix(
  c(
    0, 0, 3, 3,
    0, 1, 3, 4
  ),
  nrow = 2,
  byrow = TRUE
)

reduced_clusters <- list(
  c(1, 2, 2),
  c(1, 2, 2),
  c(1, 1, 2),
  c(1, 1, 2)
)

index <- as.list(1:4)

validation_average_distance_means(
  clusters,
  reduced_clusters,
  x,
  index
)
```

validation_bhi *Calculate Biological Homogeneity Index*

Description

Calculates the Biological Homogeneity Index (BHI), which measures homogeneity of known biological labels within clusters.

Usage

```
validation_bhi(clusters, labels)
```

Arguments

clusters	A vector containing one cluster assignment per observation.
labels	A vector containing the known label for each observation.

Value

A numeric biological homogeneity index between zero and one, with larger values indicating greater biological homogeneity.

References

Pihur V, Brock GN, Datta S. Cluster validation for microarray data: an appraisal. In: Advances in Multivariate Statistical Methods. World Scientific; 2009:79-94.

Examples

```
clusters <- c(1, 1, 2, 2)
labels <- c("A", "A", "B", "B")
validation_bhi(clusters, labels)
```

validation_bsi *Calculate Biological Stability Index*

Description

Calculates an adapted Biological Stability Index (BSI), which evaluates the stability of clustering assignments within known biological groups across dataset perturbations.

Usage

```
validation_bsi(clusters, labels, reduced_clusters)
```

Arguments

clusters	A vector containing cluster assignments obtained from the full dataset.
labels	A vector containing the known label for each observation.
reduced_clusters	A list of clustering-assignment vectors obtained after removing observations and reclustering the reduced datasets.

Details

The implementation is adapted from the biological stability measure described by Datta and Datta.

Value

A numeric biological stability index between zero and one, with larger values indicating greater stability with respect to known labels.

References

Adapted from: Datta S, Datta S. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC Bioinformatics. 2006;7:1-9.

Examples

```
clusters <- c(1, 1, 2, 2)
labels <- c("A", "A", "B", "B")

reduced_clusters <- list(
  c(1, 2, 2),
  c(1, 2, 2),
  c(1, 1, 2),
  c(1, 1, 2)
)

validation_bsi(
  clusters = clusters,
  labels = labels,
  reduced_clusters = reduced_clusters
)
```

---

validation_connectivity

Calculate Connectivity

Description

Calculates the Connectivity (CN) measure for a clustering solution using a nearest-neighbor representation of the data. The measure penalizes nearby observations that are assigned to different clusters.

Usage

```
validation_connectivity(clusters, nn, h = 5)
```

Arguments

clusters	A vector containing one cluster assignment per observation.
nn	A list containing nearest-neighbor indices for each observation.
h	Number of nearest neighbors to consider. Defaults to 5.

Value

A numeric connectivity score. Lower values indicate better agreement between cluster assignments and local neighborhood structure.

References

Pihur V, Brock GN, Datta S. Cluster validation for microarray data: an appraisal. In: Advances in Multivariate Statistical Methods. World Scientific; 2009:79-94.

Examples

```
clusters <- c(1, 1, 2, 2)
x <- matrix(
  c(
    0, 0, 3, 3,
    0, 1, 3, 4
  ),
  nrow = 2,
  byrow = TRUE
)
nn <- list(c(2, 3),  c(1, 3),  c(4, 2),  c(3, 2))
validation_connectivity(clusters, nn, h = 2)
```

---

validation_dunn *Calculate the Dunn Index*

Description

Calculates the Dunn Index (DI), defined as the minimum inter-cluster distance divided by the maximum intra-cluster distance.

Usage

```
validation_dunn(clusters, distmat)
```

Arguments

clusters	A vector containing one cluster assignment per observation.
distmat	A numeric matrix containing pairwise distances between observations.

Value

A numeric Dunn index. Larger values indicate greater separation between clusters relative to within-cluster dispersion.

References

Dunn JC. Well-separated clusters and optimal fuzzy partitions. Journal of Cybernetics. 1974;4(1):95-104.

Pihur V, Datta S, Datta S. Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. Bioinformatics. 2007;23(13):1607-1615.

Examples

```
clusters <- c(1, 1, 2, 2)
x <- matrix(
  c(
    0, 0, 3, 3,
    0, 1, 3, 4
  ),
  nrow = 2,
  byrow = TRUE
)
distmat <- as.matrix(stats::dist(t(x)))
validation_dunn(clusters, distmat)
```

---

validation_igp *Calculate In-Group Proportion*

Description

Calculates the In-Group Proportion (IGP), which summarizes the proportion of observations whose nearest neighbor belongs to the same cluster.

Usage

```
validation_igp(clusters, nn)
```

Arguments

clusters	A vector containing one cluster assignment per observation.
nn	A list containing nearest-neighbor indices for each observation.

Details

The implementation follows the overall IGP construction used by Pihur and colleagues.

Value

A numeric in-group proportion, with larger values indicating stronger agreement between clusters and local neighborhood structure.

References

Kapp AV, Tibshirani R. Are clusters found in one dataset present in another dataset? Biostatistics. 2007;8(1):9-31.

Pihur V, Brock GN, Datta S. Cluster validation for microarray data: an appraisal. In: Advances in Multivariate Statistical Methods. World Scientific; 2009:79-94.

Examples

```
clusters <- c(1, 1, 2, 2)
x <- matrix(
  c(
    0, 0, 3, 3,
    0, 1, 3, 4
  ),
  nrow = 2,
  byrow = TRUE
)
nn <- list(c(2, 3),  c(1, 3),  c(4, 2),  c(3, 2))
validation_igp(clusters, nn)
```

---

validation_silhouette *Calculate Average Silhouette Width*

Description

Calculates the average Silhouette Width (SW) for a clustering solution. For each observation, the silhouette value compares the average distance to observations in its own cluster with the average distance to the nearest alternative cluster.

Usage

```
validation_silhouette(clusters, distmat)
```

Arguments

clusters	A vector containing one cluster assignment per observation.
distmat	A numeric matrix containing pairwise distances between observations.

Value

A numeric average silhouette width. Values closer to one indicate better-separated and more compact clusters.

References

Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987;20:53-65.

Examples

```
clusters <- c(1, 1, 2, 2)
x <- matrix(
  c(
    0, 0, 3, 3,
    0, 1, 3, 4
  ),
  nrow = 2,
  byrow = TRUE
)
distmat <- as.matrix(stats::dist(t(x)))
validation_silhouette(clusters, distmat)
```

Index