

Package: scSimEval (via r-universe)

October 12, 2026

Type Package

Title Unified Benchmarking and Accuracy Evaluation for Single-Cell Multiomics Simulation Methods

Version 0.99.7

Description A simple and unified evaluation toolkit for benchmarking single-cell and multiomics data simulation methods. Evaluates how realistic simulated data are compared to real biological datasets across three multiomics experimental designs: (i) paired multiomics (simultaneous co-assay from identical cells, e.g. 10x Chromium Multiome, SHARE-seq, SNARE-seq), which unlocks the full Category 7 cross-modal coupling suite including FOSCTTM alignment scores and peak-to-gene regulatory linkage; (ii) unpaired multiomics (independent scRNA-seq and scATAC-seq profiled from separate cells of the same tissue), supported through per-modality unimodal evaluation (Categories 1-6, 8) and population-level cross-modal metrics such as co-expression module fidelity and network Jaccard similarity; and (iii) mosaic multiomics (partial co-measurement where only a subset of cells has both modalities, e.g. DOGMA-seq or mosaic pooling designs), handled by subsetting the paired cell fraction for coupling metrics and evaluating each modality independently otherwise. The toolkit quantifies simulation realism using 62 metrics organized into 8 easy-to-understand categories: (1) Distribution Differences: Statistical distances between real and simulated values (such as KS, Wasserstein, MAD, and RMSE), (2) Summary Properties: Gene correlations, zero counts, and dropout patterns, (3) Cell Structure and Clustering: How well distinct cell types and clusters are preserved (such as Silhouette width, ARI, and NMI), (4) Batch and Confounder Mixing: How well technical batch variations are captured or mixed, (5) Biological Signals and Marker Genes: Preservation of differentially expressed genes (DEGs) across Simpiper, SimBench, and Shaky Foundations frameworks, (6) Cell Lineage and Trajectories: Preservation of continuous developmental pathways (inferred directly from scRNA-seq

counts), (7) Multiomics Relationships: Connections between different layers (such as gene expression and chromatin accessibility) — fully active for paired data; population-level metrics available for unpaired and mosaic data, (8) Computational Resources: Speed (runtime in seconds) and memory usage (peak RAM in MiB). Includes non-linear low-dimensional cell embedding projections (UMAP, t-SNE, PCA) with quantitative cluster separability (Silhouette) and fidelity (ARI) metrics, automated score normalization, leaderboards, 600 DPI bubble plots, and an interactive Shiny app studio.

License MIT + file LICENSE

URL <https://kabilanbio.github.io/scSimEval/>,
<https://github.com/kabilanbio/scSimEval>

BugReports <https://github.com/kabilanbio/scSimEval/issues>

biocViews SingleCell, Transcriptomics, ATACSeq, QualityControl, DimensionReduction, DifferentialExpression, GeneExpression, Normalization, Software

Encoding UTF-8

LazyData false

Depends R (>= 4.5.0)

VignetteBuilder knitr

Imports stats, RColorBrewer, utils, Matrix, class, ggplot2, SingleCellExperiment, SummarizedExperiment, matrixStats

Suggests BiocStyle, testthat, knitr, rmarkdown, shiny, bslib, DT, writexl, openxlsx, zip, plotly, cowplot, ggrepel, patchwork, MLmetrics, overlapping, philentropy, provenance, fasano.franceschini.test, ks, waddR, emdist, MASS, randtests, caTools, edgeR, clue, mclust, cluster, clValid, clusterSim, fpc, CellMixS, kSamples, limma, RANN, bluster, Seurat, irlba, Rtsne, uwot

Config/roxygen2/version 8.1.0

Config/pak/sysreqs zlib1g-dev

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

Date/Publication 2026-10-12 17:54:42 UTC

RemoteUrl <https://github.com/BiocStaging/scSimEval>

RemoteRef HEAD

RemoteSha 0bb6c61086f3ed67561145b594b5bc36941ac0bfd

Contents

benchmark_resource_usage	5
------------------------------------	---

benchmark_simulation_suite	6
calc_accessibility_profile_concordance	7
calc_accessibility_sparsity_curve	8
calc_all_bivariate_metrics	9
calc_all_univariate_metrics	9
calc_ami	10
calc_ari	11
calc_atac_rna_coupling	12
calc_between_dataset_silhouette	12
calc_bhattacharyya	13
calc_calinski_harabasz	14
calc_cdi	14
calc_clustering_accuracy	16
calc_cms	16
calc_coexpression_module_fidelity	18
calc_completeness	19
calc_connectivity	19
calc_coregulation_fidelity	20
calc_cross_modal_generation	21
calc_cross_modality_correlation	21
calc_davies_bouldin	22
calc_deconvolution_accuracy	23
calc_delta_variance	24
calc_dunn	25
calc_ecdf_diffarea	25
calc_emd_2d	26
calc_excess_zero_weights	27
calc_fasano_franceschini	27
calc_fmi	28
calc_foscttm	29
calc_frechet_singlecell_distance	30
calc_generative_precision_recall	31
calc_homogeneity	32
calc_homogeneity_completeness_v_measure	32
calc_intraclass_correlation	33
calc_isi	34
calc_kde_test	35
calc_kinetic_noise_decomposition	35
calc_ks	36
calc_ldf	36
calc_ldf_diff	37
calc_likelihood_ratio_test	38
calc_local_structure_metric	39
calc_mad	40
calc_mae	40
calc_marginal_aic_bic	41
calc_mmd	42
calc_modality_alignment	43

calc_model_aic_bic	44
calc_neighborhood_purity	44
calc_network_jaccard	45
calc_nmi	46
calc_nn_mismatch	46
calc_outlier_proportion	47
calc_overlap	48
calc_peacock_2d	48
calc_peak_coaccessibility_fidelity	49
calc_pseudotime_correlation	50
calc_rmse	50
calc_runs_test	51
calc_seurat_mixing_metric	52
calc_signal_bd	53
calc_signal_dd	53
calc_signal_de	54
calc_signal_dp	54
calc_signal_dv	55
calc_silhouette	56
calc_tree_height_discrepancy	56
calc_v_measure	57
calc_variance_decomposition	58
calc_wasserstein_1d	59
calc_zero_probability_curve	59
compute_dataset_embeddings	60
compute_embedding_quality_metrics	61
compute_method_leaderboard	62
evaluate_accessibility_sparsity_curve	63
evaluate_batch_metrics	63
evaluate_clustering_metrics	65
evaluate_cross_batch_prediction	66
evaluate_cross_modal_prediction	67
evaluate_deg_fidelity	67
evaluate_epigenomic_annotation	69
evaluate_multiomics_accuracy	70
evaluate_multiple_datasets	72
evaluate_predictive_de_model	73
evaluate_simbench_signals	74
evaluate_simulation_accuracy	75
evaluate_trajectory_metrics	77
evaluate_zero_probability_curve	77
example_batch_info	78
example_cell_types	79
example_multiomics	79
example_scatac	80
example_scrna	81
extract_cell_properties	82
extract_dataset_summary	82

extract_feature_properties	83
fast_col_vars	84
fast_cross_distance_sq	84
fast_pca	85
fast_row_vars	85
fast_sparse_zero_fraction_cell	86
fast_sparse_zero_fraction_feature	86
infer_scrna_lineage_tree	87
infer_scrna_pseudotime	87
launch_scSimEval_app	88
plot_benchmark_bubble_matrix	89
plot_category_metric_bars	91
plot_dataset_embeddings	92
plot_distribution_qc	94
plot_evaluation_summary	95
plot_individual_metric_bar	97
plot_metric_boxplots	98
plot_metric_heatmap	99
plot_metric_mds	101
plot_metric_pca	102
plot_scalability_benchmark	104
subsample_dist_vector	105

Index**106**

benchmark_resource_usage

*Benchmark Execution Runtime and Memory Usage***Description**

Times an expression and measures change in memory allocation.

Usage

```
benchmark_resource_usage(expr)
```

Arguments

expr Expression or function to benchmark.

Value

A list with elapsed seconds and peak memory allocated (MB).

Examples

```
benchmark_resource_usage({ Sys.sleep(0.01); 1 + 1 })
```

benchmark_simulation_suite

Benchmark Simulation Methods Across Multiple Single-Cell Datasets

Description

A unified, high-performance, and beginner-friendly benchmarking runner designed for multi-dataset and multi-simulator studies (e.g., 16 datasets x 6 simulators). Automatically manages memory allocation with sparse matrix coercions, parallel execution across datasets, progress tracking, and produces both granular and cross-dataset aggregated benchmark tables ready for visualization.

Usage

```
benchmark_simulation_suite(
  datasets,
  simulators,
  compute_bivariate = FALSE,
  threads = 1,
  auto_sparse = TRUE,
  run_gc = TRUE,
  save_path = NULL,
  verbose = TRUE
)
```

Arguments

datasets	A named list of empirical reference datasets. Each item can be a single-cell count matrix (features x cells), a path to an RDS/CSV matrix file, or a list with ref, optional cell_types, and optional batch_info.
simulators	A named list of simulation methods. Each element must be a named list of simulated count matrices (or matrix file paths) corresponding to the datasets in datasets. Example: <code>list(Splatter = list(D1 = sim_splat_d1, D2 = sim_splat_d2), scDesign3 = ...)</code> .
compute_bivariate	Logical. Whether to compute 2D bivariate tests. Default FALSE for fast high-throughput execution.
threads	Number of threads / parallel workers. Default is 1.
auto_sparse	Logical. Automatically coerce dense count matrices to <code>dgCMatrix</code> to minimize RAM usage. Default TRUE.
run_gc	Logical. Run garbage collection (<code>gc()</code>) after each dataset evaluation to prevent memory accumulation on HPC nodes. Default TRUE.
save_path	Optional file path (ending in <code>.rds</code>) to directly persist the consolidated benchmark bundle for loading into Shiny Studio.
verbose	Logical. Print informative progress logging. Default TRUE.

Value

An object of class `scSimEval_multidataset_benchmark` containing:

- `granular_benchmark_table`: Tidy data.frame containing all evaluated metrics across every dataset and simulator.
- `benchmark_summary_table`: Tidy data.frame containing cross-dataset averaged fidelity scores for each simulator (ready for `plot_benchmark_bubble_matrix()`).
- `dataset_summary_table`: Summary of reference dataset dimensions, sparsity, and library properties.
- `methods`: Vector of evaluated simulator names.
- `datasets`: Vector of evaluated dataset names.
- `cross_dataset_stats`: Summary statistics (mean, SD, SE) for each metric across datasets.

Examples

```
data(example_scrna, package = "scSimEval")
refs <- list("CohortA" = example_scrna$ref)
sims <- list("Splatter" = list("CohortA" = example_scrna$sim))
res <- benchmark_simulation_suite(refs, sims, verbose = FALSE)
print(res)
```

calc_accessibility_profile_concordance

Evaluate Chromatin Accessibility Profile Concordance (DiTSim)

Description

Computes global and cell-type-stratified Pearson and Spearman correlations and Kullback-Leibler (KL) divergence between empirical reference and simulated single-cell chromatin accessibility profiles.

Usage

```
calc_accessibility_profile_concordance(
  ref_data,
  sim_data,
  cell_types = NULL,
  use_tfidf = TRUE
)
```

Arguments

<code>ref_data</code>	Reference count/accessibility matrix (peaks x cells).
<code>sim_data</code>	Simulated count/accessibility matrix (peaks x cells).
<code>cell_types</code>	Optional factor or vector of cell-type annotations for cells.
<code>use_tfidf</code>	Logical, whether to apply TF-IDF transformation prior to mean profile calculation (default TRUE).

Value

A list containing global PCC, global SCC, per-cell-type mean PCC/SCC, and mean accessibility KL divergence.

Examples

```
data(example_multiomics, package = "scSimEval")
r_rna <- example_multiomics$ref_multi$rna
r_atac <- example_multiomics$ref_multi$atac
calc_accessibility_profile_concordance(r_rna, r_atac)
```

calc_accessibility_sparsity_curve

Fit Chromatin Accessibility-Sparsity Polynomial Curve (simATAC)

Description

Fits a polynomial curve relating peak/bin mean accessibility to non-zero cell proportion (NZP / detection frequency) as modeled in the simATAC framework (Navidi et al., Genome Biology 2021):

$$\text{NZP} = c_0 + c_1 * \text{mean} + c_2 * \text{mean}^2.$$

Usage

```
calc_accessibility_sparsity_curve(data, poly_degree = 2)
```

Arguments

data	Count matrix (features/peaks x cells) or data.frame with 'mean' and 'nzp'.
poly_degree	Degree of polynomial (default: 2 for quadratic curve).

Value

A list with estimated coefficients (c0, c1, c2), R-squared, Spearman/Pearson correlations, and model fit summary.

Examples

```
data(example_scrna, package = "scSimEval")
calc_accessibility_sparsity_curve(example_scrna$ref)
```

`calc_all_bivariate_metrics`*Compute All Bivariate (2D) Distance & Accuracy Metrics for Joint Distributions*

Description

Compute All Bivariate (2D) Distance & Accuracy Metrics for Joint Distributions

Usage

```
calc_all_bivariate_metrics(ref_mat, sim_mat, metric_prefix = "", threads = 1)
```

Arguments

<code>ref_mat</code>	2-column matrix for reference.
<code>sim_mat</code>	2-column matrix for simulation.
<code>metric_prefix</code>	Optional prefix string for metric names.
<code>threads</code>	CPU threads.

Value

Named list of bivariate metrics.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)
sim <- matrix(stats::rnorm(100), 50, 2)
calc_all_bivariate_metrics(ref, sim)
```

`calc_all_univariate_metrics`*Compute All Univariate Distance & Accuracy Metrics for a Feature*

Description

Compute All Univariate Distance & Accuracy Metrics for a Feature

Usage

```
calc_all_univariate_metrics(ref, sim, metric_prefix = "")
```

Arguments

- ref Numeric vector of reference values.
- sim Numeric vector of simulated values.
- metric_prefix Optional prefix string for metric names.

Value

Named list of univariate distance metrics.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_all_univariate_metrics(ref, sim)
```

calc_ami	<i>Calculate Adjusted Mutual Information (AMI)</i>
----------	--

Description

Computes the Adjusted Mutual Information (AMI) between two clusterings or cell-type label assignments, correcting for chance expected mutual information under the hypergeometric model with fixed marginals (Vinh et al., JMLR 2010), as used in scMoMtF (Lan et al., PLOS Comput Biol 2024).

Usage

```
calc_ami(
  pred,
  truth,
  average_method = c("arithmetic", "max", "min", "geometric")
)
```

Arguments

- pred Vector of predicted cluster labels.
- truth Vector of ground truth cell type labels.
- average_method Method to normalize mutual information: "arithmetic" (default, scikit-learn standard), "max", "min", or "geometric".

Value

AMI score in [0, 1] (adjusted for chance).

References

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11, 2837-2854.

Lan, W., Ling, T., Chen, Q. et al. scMoMtF: An interpretable multitask learning framework for single-cell multi-omics data analysis. *PLOS Comput Biol* 20(12): e1012679 (2024).

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_ami(pred, truth)
```

calc_ari	<i>Calculate Adjusted Rand Index (ARI)</i>
----------	--

Description

Calculate Adjusted Rand Index (ARI)

Usage

```
calc_ari(pred, truth)
```

Arguments

pred	Predicted cluster labels.
truth	Ground truth cell type labels.

Value

ARI value between -1 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_ari(pred, truth)
```

calc_atac_rna_coupling

Chromatin Accessibility to RNA Regulatory Effect Coupling

Description

Evaluates whether simulated paired scATAC-seq and scRNA-seq profiles capture the true regulatory coupling where regional chromatin opening gates target gene transcription (scMultiSim; Li et al., Nat Methods 2023).

Usage

```
calc_atac_rna_coupling(atac_data, rna_data, linked_pairs = NULL)
```

Arguments

atac_data	Matrix of chromatin peak accessibilities (peaks x cells).
rna_data	Matrix of gene expression counts (genes x cells).
linked_pairs	Optional 2-column data.frame of known linked peak-gene pairs. If NULL, assumes 1-to-1 matching by row index.

Value

A list containing mean coupling correlation, positive coupling ratio, and mean R-squared.

Examples

```
data(example_multiomics, package = "scSimEval")
r_rna <- example_multiomics$ref_multi$rna
r_atac <- example_multiomics$ref_multi$atac
calc_atac_rna_coupling(r_rna, r_atac)
```

calc_between_dataset_silhouette

Calculate Between-Dataset Silhouette Width

Description

Integrated from countsimQC (Soneson & Robinson). Treats dataset identity (reference vs. simulation) as cluster labels to test if the two datasets separate into distinct clusters or remain well-mixed.

Usage

```
calc_between_dataset_silhouette(ref, sim, subsample_size = 300)
```

Arguments

ref Vector or matrix for reference dataset.
 sim Vector or matrix for simulated dataset.
 subsample_size Number of subsampled points to evaluate (default 300).

Value

A named list with global and local between-dataset silhouette widths.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_between_dataset_silhouette(ref, sim)
```

calc_bhattacharyya	<i>Calculate Bhattacharyya Distance (BH)</i>
--------------------	--

Description

Calculate Bhattacharyya Distance (BH)

Usage

```
calc_bhattacharyya(ref, sim, align = TRUE)
```

Arguments

ref Numeric vector of reference values.
 sim Numeric vector of simulated values.
 align Logical, whether to align lengths. Default TRUE.

Value

Statistical divergence between discrete probability measures.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_bhattacharyya(ref, sim)
```

calc_calinski_harabasz	<i>Calculate Calinski-Harabasz Index (CH)</i>
------------------------	---

Description

Calculate Calinski-Harabasz Index (CH)

Usage

```
calc_calinski_harabasz(data, cluster_labels)
```

Arguments

- data Matrix with features in rows, cells in columns.
- cluster_labels Cluster assignments.

Value

Calinski-Harabasz index.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 5))
dist_mat <- stats::dist(data)
calc_calinski_harabasz(data, cl)
```

calc_cdi	<i>Compute Clustering Deviation Index (CDI) for Single-Cell Clustering Evaluation</i>
----------	---

Description

Evaluates the deviation and goodness-of-fit of single-cell clustering labels directly from raw or normalized count matrices without requiring ground-truth cell-type labels (Fang et al., Genome Biology 2022; benchmarked in scDesign3, Song et al., Nat Biotechnol 2024). CDI calculates the penalized log-likelihood (CDI-AIC and CDI-BIC) of count data under a cluster-specific Poisson or Negative Binomial GLM with cell size factors.

Usage

```
calc_cdi(
  counts,
  cluster_labels,
  size_factors = NULL,
  model = c("poisson", "nb"),
  top_features = NULL
)
```

Arguments

counts	Matrix or data.frame of counts (genes x cells or cells x genes).
cluster_labels	Vector of cluster assignments for each cell.
size_factors	Optional numeric vector of cell-specific size factors. If NULL, computed as library size scaled by median library size.
model	Distribution model to use: "poisson" (fast, default) or "nb" (Negative Binomial).
top_features	Optional integer; if set, restricts CDI calculation to the top most variable features to accelerate computation on large matrices.

Value

A list containing:

cdi_aic	Normalized CDI score based on AIC penalty (lower indicates better clustering).
cdi_bic	Normalized CDI score based on BIC penalty (favors parsimonious main clusters).
loglik	Total log-likelihood under the cluster-specific count model.
deviance	Model deviance relative to saturated count model.
n_clusters	Number of clusters evaluated.
n_params	Number of free parameters in the clustering model.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 10))
dist_mat <- stats::dist(data)
calc_cdi(data, cl)
```

calc_clustering_accuracy	<i>Calculate Clustering Accuracy (ACC)</i>
--------------------------	--

Description

Integrated from scCluBench (Xu et al., AAAI 2026). Aligns predicted clusters to ground truth classes using the Hungarian bipartite matching algorithm, then computes the fraction of correctly mapped cells.

Usage

```
calc_clustering_accuracy(pred, truth)
```

Arguments

- pred Vector of predicted cluster labels.
- truth Vector of ground truth cell type labels.

Value

Clustering accuracy between 0 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_clustering_accuracy(pred, truth)
```

calc_cms	<i>Cell-Specific Mixing Score (CMS)</i>
----------	---

Description

Tests the hypothesis that group-specific distance distributions of k-nearest neighbor cells have the same underlying unspecified distribution. Computes cell-specific p-values via Anderson-Darling tests (if kSamples installed) or two-sample Kolmogorov-Smirnov / Kruskal-Wallis tests, with optional delegation to CellMixS::cms.

Usage

```
calc_cms(
  data = NULL,
  batch_info,
  k = NULL,
  n_pcs = 30,
  coords = NULL,
  cell_min = 4,
  unbalanced = FALSE
)
```

Arguments

data	Count or log-normalized expression matrix (features x cells) or NULL if coords provided.
batch_info	Factor or character vector of batch assignments for each cell.
k	Neighborhood size k for k-NN graph. Default is min(table(batch_info))/2 or 30.
n_pcs	Number of principal components to compute if coords is NULL. Default is 30.
coords	Optional precomputed embedding matrix (cells x dimensions, e.g. PCA or UMAP).
cell_min	Minimum number of cells per batch in neighborhood to perform test. Default is 4.
unbalanced	Logical, whether to set pure neighborhoods to NA. Default is FALSE.

Value

A named list:

cms_scores	Numeric vector of CMS p-values for each cell (higher means better mixed)
mean_cms	Mean CMS score across cells
median_cms	Median CMS score across cells

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
calc_cms(coords, batch)
```

`calc_coexpression_module_fidelity`*Evaluate Gene Co-Expression Module Fidelity (ESCO)*

Description

Automatically discovers or evaluates co-expression modules and measures the preservation of intra-module vs. inter-module correlation structure.

Usage

```
calc_coexpression_module_fidelity(  
  ref_mat,  
  sim_mat,  
  modules = NULL,  
  n_modules = 5,  
  top_genes = 200  
)
```

Arguments

<code>ref_mat</code>	Reference gene expression matrix (genes x cells).
<code>sim_mat</code>	Simulated gene expression matrix (genes x cells).
<code>modules</code>	Optional named list of gene modules. If NULL, auto-detected via hierarchical clustering.
<code>n_modules</code>	Number of modules to detect if modules is NULL (default 5).
<code>top_genes</code>	Number of top variable genes to consider for clustering (default 200).

Value

A list summarizing module correlation preservation, RMSE, and modularity ratio fidelity.

Examples

```
data(example_scrna, package = "scSimEval")  
calc_coexpression_module_fidelity(example_scrna$ref, example_scrna$sim)
```

calc_completeness	<i>Calculate Completeness Score</i>
-------------------	-------------------------------------

Description

Calculate Completeness Score

Usage

```
calc_completeness(pred, truth)
```

Arguments

pred	Vector of predicted cluster labels.
truth	Vector of ground truth cell type labels.

Value

Completeness score between 0 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))  
truth <- factor(rep(c("A", "B"), each = 20))  
calc_completeness(pred, truth)
```

calc_connectivity	<i>Calculate Cluster Connectivity</i>
-------------------	---------------------------------------

Description

Calculate Cluster Connectivity

Usage

```
calc_connectivity(dist_mat, cluster_labels)
```

Arguments

dist_mat	Distance matrix or numeric matrix.
cluster_labels	Cluster assignments.

Value

Connectivity metric.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 10))
dist_mat <- stats::dist(data)
calc_connectivity(dist_mat, cl)
```

calc_coregulation_fidelity

Multi-Omics Co-Regulation & Modularity Fidelity

Description

Evaluates the preservation of co-regulated gene/feature modules between reference and simulated multi-omics datasets (Monzo et al., 2025; Arzalluz-Luque et al., 2022).

Usage

```
calc_coregulation_fidelity(
  ref_data,
  sim_data,
  modules,
  method = c("pearson", "spearman")
)
```

Arguments

ref_data	Reference feature-by-cell matrix or data frame.
sim_data	Simulated feature-by-cell matrix or data frame.
modules	A list of character vectors representing feature clusters/modules, or a named vector/factor of module assignments per feature.
method	Correlation method: "pearson" (default) or "spearman".

Value

A list containing module correlation r, RMSE, MAE, and modularity fidelity.

Examples

```
data(example_scrna, package = "scSimEval")
modules <- list(Module_1 = rownames(example_scrna$ref)[seq_len(30)],
  Module_2 = rownames(example_scrna$ref)[31:60])
calc_coregulation_fidelity(example_scrna$ref, example_scrna$sim, modules)
```

calc_cross_modal_generation
<i>Cross-Modality In Silico Generation & Translation Fidelity</i>

Description

Evaluates the biological accuracy and reconstruction fidelity of cross-modality generation (e.g. predicting scATAC from scRNA or vice versa; Zhai et al., Genome Biology 2024). Computes cell-wise and feature-wise Pearson and Spearman correlations, cosine similarity, RMSE, and MAE between predicted and measured multi-omics profiles.

Usage

```
calc_cross_modal_generation(true_data, pred_data)
```

Arguments

- true_data Matrix or data.frame of measured features across cells (features x cells).
- pred_data Matrix or data.frame of in-silico generated / predicted features (features x cells).

Value

A list containing cell-wise and feature-wise correlation, cosine similarity, RMSE, and MAE.

Examples

```
data(example_multiomics, package = "scSimEval")
r_rna <- example_multiomics$ref_multi$rna
r_atac <- example_multiomics$ref_multi$atac
calc_cross_modal_generation(r_rna, r_atac)
```

calc_cross_modality_correlation
<i>Calculate Cross-Modality Correlation Fidelity</i>

Description

Evaluates whether cross-modality feature relationships (e.g. peak-to-gene links, promoter accessibility vs. gene expression, or mRNA vs. surface protein) are accurately captured by the simulation method compared to the empirical reference.

Usage

```
calc_cross_modality_correlation(
  ref_mod1,
  ref_mod2,
  sim_mod1,
  sim_mod2,
  feature_pairs = NULL,
  method = c("spearman", "pearson")
)
```

Arguments

ref_mod1	Reference matrix for Modality 1 (features x cells).
ref_mod2	Reference matrix for Modality 2 (features x cells).
sim_mod1	Simulated matrix for Modality 1.
sim_mod2	Simulated matrix for Modality 2.
feature_pairs	Optional 2-column data.frame/matrix of paired feature names/indices.
method	Correlation method: "spearman" (default) or "pearson".

Value

A named list of the 7 univariate accuracy metrics comparing the reference vs. simulated cross-modality correlation distributions.

Examples

```
data(example_multiomics, package = "scSimEval")
m <- example_multiomics
calc_cross_modality_correlation(m$ref_multi$rna, m$ref_multi$atac,
                                m$sim_multi$rna, m$sim_multi$atac)
```

calc_davies_bouldin	<i>Calculate Davies-Bouldin Index (DB)</i>
---------------------	--

Description

Calculate Davies-Bouldin Index (DB)

Usage

```
calc_davies_bouldin(data, cluster_labels)
```

Arguments

data	Matrix with features in rows, cells in columns.
cluster_labels	Cluster assignments.

Value

Davies-Bouldin index (lower is better).

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 5))
dist_mat <- stats::dist(data)
calc_davies_bouldin(data, cl)
```

calc_deconvolution_accuracy

Evaluate Cell-Type Deconvolution & Mixture Proportion Accuracy

Description

Evaluates the accuracy of cell-type mixture proportion estimation / deconvolution methods between empirical reference and simulated cell populations (Chen et al., Bioinformatics 2021). Can accept precomputed proportion matrices OR cell-type vectors (and optional batch labels) to calculate proportion preservation across batches.

Usage

```
calc_deconvolution_accuracy(
  true_proportions,
  estimated_proportions,
  batch_ref = NULL,
  batch_sim = NULL
)
```

Arguments

true_proportions	Vector, matrix, or data.frame of reference cell-type proportions, or factor/vector of reference cell types.
estimated_proportions	Vector, matrix, or data.frame of estimated / simulated cell-type proportions, or factor/vector of simulated cell types.
batch_ref	Optional batch labels for reference cells (if true_proportions is a cell-type vector).
batch_sim	Optional batch labels for simulated cells (if estimated_proportions is a cell-type vector).

Value

A list containing RMSE, MAE, Pearson correlation (r), Spearman correlation (rho), Jensen-Shannon Divergence (JSD), and Total Variation Distance (TVD).

Examples

```
data(example_scrna, package = "scSimEval")
calc_deconvolution_accuracy(example_scrna$ref, example_scrna$sim)
```

calc_delta_variance	<i>Calculate Delta Variance (Pseudoreplication Bias Metric)</i>
---------------------	---

Description

Measures the difference in gene expression variance between true biological replicates and cell-shuffled pseudoreplicates (Squair et al., Nature Communications 2021). Positive delta variance indicates that biological replicates have higher inter-individual variability than randomly pooled pseudoreplicates.

Usage

```
calc_delta_variance(
  counts,
  replicates,
  conditions,
  cell_types = NULL,
  n_perm = 3
)
```

Arguments

counts	Count matrix (features x cells).
replicates	Factor of biological replicate / batch assignments for each cell.
conditions	Factor of condition or cell-type labels.
cell_types	Optional cell type vector to evaluate per cell type or overall.
n_perm	Number of permutation shuffles to compute mean pseudoreplicate variance. Default is 3.

Value

A named list:

delta_variance	Numeric vector of gene-level delta variance values
mean_delta_variance	Mean delta variance across all genes
median_delta_variance	Median delta variance across all genes

Examples

```
data(example_scrna, package = "scSimEval")
calc_delta_variance(example_scrna$ref,
                     replicates = example_scrna$batch_info,
                     conditions = example_scrna$cell_types)
```

calc_dunn	<i>Calculate Dunn Index</i>
-----------	-----------------------------

Description

Calculate Dunn Index

Usage

```
calc_dunn(dist_mat, cluster_labels)
```

Arguments

`dist_mat` Distance matrix or numeric matrix.
`cluster_labels` Cluster assignments.

Value

Dunn index (ratio of smallest inter-cluster to largest intra-cluster distance).

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 10))
dist_mat <- stats::dist(data)
calc_dunn(dist_mat, cl)
```

calc_ecdf_diffarea	<i>Calculate Area Between Empirical Cumulative Distribution Functions (eCDFs)</i>
--------------------	---

Description

Integrated from countsimQC (Soneson & Robinson). Measures the normalized area between the eCDFs of reference and simulation across the shared support.

Usage

```
calc_ecdf_diffarea(ref, sim)
```

Arguments

ref Numeric vector of reference values.
 sim Numeric vector of simulated values.

Value

Normalized area between the two eCDFs.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_ecdf_diffarea(ref, sim)
```

 calc_emd_2d

Calculate 2D Earth Mover's Distance (2D EMD)

Description

Integrated from HelenaLC/simulation-comparison.

Computes 2D bivariate density via MASS::kde2d over a shared bounding box, then measures optimal transport distance via emdista::emd2d.

Usage

```
calc_emd_2d(ref_mat, sim_mat, n = 25)
```

Arguments

ref_mat 2-column numeric matrix for reference.
 sim_mat 2-column numeric matrix for simulation.
 n Grid resolution for 2D density estimation. Default is 25.

Value

Normalized 2D Earth Mover's Distance.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)
sim <- matrix(stats::rnorm(100), 50, 2)
calc_emd_2d(ref, sim)
```

`calc_excess_zero_weights`*Calculate Posterior Excess Zero Weights (zingeR & ZINB-WaVE)*

Description

Calculates the posterior probability of zero counts being technical dropouts (excess zeros) versus biological sampling zeros under a Negative Binomial model.

Usage

```
calc_excess_zero_weights(counts)
```

Arguments

counts Count matrix (genes x cells).

Value

A list with mean excess zero weight, gene-level weights, and estimated zero-inflation rate.

Examples

```
data(example_scrna, package = "scSimEval")
calc_excess_zero_weights(example_scrna$ref)
```

`calc_fasano_franceschini`*Calculate Fasano-Franceschini 2D Kolmogorov-Smirnov Test Statistic*

Description

Integrated from simpipe.

Usage

```
calc_fasano_franceschini(ref_mat, sim_mat, threads = 1)
```

Arguments

ref_mat 2-column numeric matrix for reference.
sim_mat 2-column numeric matrix for simulation.
threads CPU threads. Default 1.

Value

Estimated 2D KS statistic.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)
sim <- matrix(stats::rnorm(100), 50, 2)
calc_fasano_franceschini(ref, sim)
```

calc_fmi	<i>Calculate Fowlkes-Mallows Index (FMI)</i>
----------	--

Description

Integrated from scCluBench (Xu et al., AAAI 2026). Computes the geometric mean of pairwise cluster precision and recall.

Usage

```
calc_fmi(pred, truth)
```

Arguments

- pred Vector of predicted cluster labels.
- truth Vector of ground truth cell type labels.

Value

FMI score between 0 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_fmi(pred, truth)
```

calc_foscttm	<i>Fraction of Samples Closer Than The True Match (FOSCTTM)</i>
--------------	---

Description

Computes the Fraction of Samples Closer Than The True Match (FOSCTTM) to quantify cell alignment error across single-cell multi-omics modalities in a shared latent or integrated space (Zhai et al., Genome Biology 2024; Liu et al., Nat Biotechnol 2023). A score of 0 represents perfect alignment (where the true paired cell is the nearest neighbor), while 0.5 corresponds to random chance.

Usage

```
calc_foscttm(x, y, metric = c("euclidean", "cosine"))
```

Arguments

x	Coordinates matrix for Modality 1 (cells x dimensions).
y	Coordinates matrix for Modality 2 (cells x dimensions), where row i of y corresponds to the true match of row i of x.
metric	Distance metric to evaluate: "euclidean" (default) or "cosine".

Value

A list containing:

foscttm	Bidirectional mean FOSCTTM score across all cells (lower is better, 0 to 0.5).
foscttm_xy	Directional FOSCTTM from Modality 1 to Modality 2.
foscttm_yx	Directional FOSCTTM from Modality 2 to Modality 1.
match_at_1	Top-1 match rate (proportion of cells where the true match is rank 1).
match_at_5	Top-5 match rate (proportion of cells where the true match is in top 5).
cell_foscttm	Vector of bidirectional FOSCTTM scores for individual cells.

Examples

```
data(example_multiomics, package = "scSimEval")
r_rna <- example_multiomics$ref_multi$rna
r_atac <- example_multiomics$ref_multi$atac
calc_foscttm(r_rna, r_atac)
```

`calc_frechet_singlecell_distance`*Frechet Single-Cell Distance (FSD)*

Description

Computes the single-cell analogue of Frechet Inception Distance (FID) on low-dimensional PCA embeddings between reference and simulated cell populations.

Usage

```
calc_frechet_singlecell_distance(  
  ref_mat,  
  sim_mat,  
  cells_as_cols = TRUE,  
  n_pcs = 15,  
  max_features = 2000  
)
```

Arguments

<code>ref_mat</code>	Matrix of reference cells (cells x features or features x cells).
<code>sim_mat</code>	Matrix of simulated cells (cells x features or features x cells).
<code>cells_as_cols</code>	Logical, whether cells are columns (default TRUE).
<code>n_pcs</code>	Number of principal components to evaluate (default 15).
<code>max_features</code>	Number of top variable features to retain (default 2000).

Value

A list with Frechet distance (FSD), mean discrepancy, and covariance trace discrepancy.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)  
sim <- matrix(stats::rnorm(100), 50, 2)  
calc_frechet_singlecell_distance(ref, sim)
```

calc_generative_precision_recall

Generative Precision and Recall for Single-Cell Manifolds

Description

Evaluates the realism (Precision) and coverage/diversity (Recall) of generated single-cell distributions relative to empirical reference data using k-NN hyperspheres.

Usage

```
calc_generative_precision_recall(
  ref_mat,
  sim_mat,
  cells_as_cols = TRUE,
  k = 5,
  max_cells = 500,
  max_features = 1000
)
```

Arguments

ref_mat	Matrix of reference cells (cells x features or features x cells).
sim_mat	Matrix of simulated cells (cells x features or features x cells).
cells_as_cols	Logical, whether cells are columns (default TRUE).
k	Number of nearest neighbors to define the local manifold radius (default 5).
max_cells	Maximum number of cells to subsample (default 500).
max_features	Maximum number of features to retain for speed (default 1000).

Value

A list containing Generative Precision, Generative Recall, and Generative F1.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 10))
dist_mat <- stats::dist(data)
calc_generative_precision_recall(data, cl)
```

calc_homogeneity	<i>Calculate Homogeneity Score</i>
------------------	------------------------------------

Description

Calculate Homogeneity Score

Usage

```
calc_homogeneity(pred, truth)
```

Arguments

pred	Vector of predicted cluster labels.
truth	Vector of ground truth cell type labels.

Value

Homogeneity score between 0 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_homogeneity(pred, truth)
```

calc_homogeneity_completeness_v_measure	<i>Calculate Homogeneity, Completeness, and V-Measure</i>
---	---

Description

Integrated from scCluBench (Xu et al., AAAI 2026).

Usage

```
calc_homogeneity_completeness_v_measure(pred, truth, beta = 1)
```

Arguments

pred	Vector of predicted cluster labels.
truth	Vector of ground truth cell type labels.
beta	Weight of completeness vs. homogeneity (default 1).

Value

A named list containing homogeneity, completeness, and v_measure.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_homogeneity_completeness_v_measure(pred, truth)
```

calc_intraclass_correlation

Calculate Intraclass Correlation Coefficient (ICC) (hierarchicell & rescueSim)

Description

Computes the one-way random effects Intraclass Correlation Coefficient (ICC) per gene across donors or biological replicates to quantify hierarchical clustering.

Usage

```
calc_intraclass_correlation(counts, donor_labels, n_genes = 200)
```

Arguments

counts	Matrix (genes x cells) or SingleCellExperiment.
donor_labels	Vector or factor of donor / subject IDs.
n_genes	Number of top variable genes to evaluate (default 200).

Value

A list with median ICC, mean ICC, and vector of per-gene ICC values.

Examples

```
data(example_scrna, package = "scSimEval")
calc_intraclass_correlation(example_scrna$ref, example_scrna$sim)
```

calc_isi	<i>Inverse Simpson Index (ISI) for Batch Mixing</i>
----------	---

Description

Evaluates the diversity of batch labels within each cell's k-nearest neighborhood using the Inverse Simpson Index ($1 / \sum(p_b^2)$). Implements a pure-R, lightweight formulation of LISI with optional distance weighting.

Usage

```
calc_isi(coords, batch_info, k = 30, weighted = TRUE)
```

Arguments

coords	Matrix of cell coordinates (cells x dimensions).
batch_info	Factor or character vector of batch assignments.
k	Number of nearest neighbors. Default is 30.
weighted	Logical, whether to weight neighbor contributions by inverse distance. Default is TRUE.

Value

A named list:

isi_scores	Numeric vector of cell-level ISI values
mean_isi	Mean ISI score (ranges from 1 to number of batches; higher indicates better mixing)
median_isi	Median ISI score

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
calc_isi(coords, batch)
```

calc_kde_test	<i>Calculate 2D Bivariate Kernel Density Estimation (KDE) Test Statistic</i>
---------------	--

Description

Integrated from simpipe & SimBench.

Usage

```
calc_kde_test(ref_mat, sim_mat)
```

Arguments

ref_mat	2-column numeric matrix for reference.
sim_mat	2-column numeric matrix for simulation.

Value

z-statistic testing discrepancy between bivariate kernel density estimates.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)
sim <- matrix(stats::rnorm(100), 50, 2)
calc_kde_test(ref, sim)
```

calc_kinetic_noise_decomposition	<i>Kinetic Noise Decomposition (SymSim & Elowitz et al.)</i>
----------------------------------	--

Description

Decomposes gene expression variance / squared coefficient of variation (CV^2) into intrinsic transcriptional bursting noise and extrinsic cell-state noise.

Usage

```
calc_kinetic_noise_decomposition(counts, cell_states = NULL)
```

Arguments

counts	Matrix of expression counts (genes x cells) or SingleCellExperiment.
cell_states	Optional factor or vector of cell states / subpopulation clusters. If provided, noise is partitioned into within-state (intrinsic) and between-state (extrinsic) components. If NULL, intrinsic noise is estimated via Poisson shot-noise expectation ($1 / \text{mean}$).

Value

A list containing mean intrinsic noise, mean extrinsic noise, noise ratio, and gene-level vectors.

Examples

```
data(example_scrna, package = "scSimEval")
calc_kinetic_noise_decomposition(example_scrna$ref)
```

calc_ks	<i>Calculate Kolmogorov-Smirnov Distance (1D KS)</i>
---------	--

Description

Calculate Kolmogorov-Smirnov Distance (1D KS)

Usage

```
calc_ks(ref, sim)
```

Arguments

- ref Numeric vector of reference values.
- sim Numeric vector of simulated values.

Value

Maximum vertical distance between empirical CDFs (between 0 and 1).

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_ks(ref, sim)
```

calc_ldf	<i>Local Density Factor (LDF)</i>
----------	-----------------------------------

Description

Computes the Local Density Estimate (LDE) and Local Density Factor (LDF) using a Gaussian kernel over reachability distances in the k-NN neighborhood, adapted from Latecki et al. and CellMixS.

Usage

```
calc_ldf(coords, k = 15, h = 1, c = 1)
```

Arguments

coords	Matrix of coordinates / embeddings (cells x dimensions).
k	Number of nearest neighbors. Default is 15.
h	Bandwidth parameter for Gaussian kernel. Default is 1.
c	Scaling constant for comparison of LDE to neighboring observations. Default is 1.

Value

A named list:

lde	Local density estimate for each cell
ldf	Local density factor for each cell

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
calc_ldf(coords)
```

calc_ldf_diff	<i>Local Density Differences (ldfDiff)</i>
---------------	--

Description

Quantifies cell-specific changes in the Local Density Factor (LDF) before and after data integration or between reference and simulated datasets.

Usage

```
calc_ldf_diff(coords_pre, coords_post, k = 15, h = 1, c = 1)
```

Arguments

coords_pre	Matrix of coordinates before integration / reference (cells x dimensions).
coords_post	Matrix of coordinates after integration / simulated (cells x dimensions).
k	Number of nearest neighbors. Default is 15.
h	Bandwidth parameter for Gaussian kernel. Default is 1.
c	Scaling constant. Default is 1.

Value

A named list:

diff	Numeric vector of absolute LDF differences per cell
mean_ldf_diff	Mean absolute LDF difference (lower indicates better structure preservation)
median_ldf_diff	Median absolute LDF difference
ldf_pre	LDF in pre-integration / reference space
ldf_post	LDF in post-integration / simulated space

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
calc_ldf_diff(coords, coords)
```

calc_likelihood_ratio_test
<i>Perform Likelihood Ratio Test for Comparing Nested Single-Cell Simulation Models</i>

Description

Direct port and generalization of `scDesign3::perform_lrt` (Song et al., Nat Biotechnol 2024). Performs the likelihood ratio test to compare two nested simulation models (e.g., cell-type/covariate model vs intercept-only null model, or spline trajectory vs linear model).

Usage

```
calc_likelihood_ratio_test(
  alter_model,
  null_model,
  df_alter = NULL,
  df_null = NULL
)
```

Arguments

alter_model	Alternative model (more complex) or list of alternative models per gene, or numeric log-likelihoods.
null_model	Null model (simpler, strictly nested) or list of null models per gene, or numeric log-likelihoods.
df_alter	Degrees of freedom for alternative model (used if models are numeric log-likelihoods).
df_null	Degrees of freedom for null model (used if models are numeric log-likelihoods).

Value

A data.frame containing:

LogLik_alter	Log-likelihood under alternative model.
LogLik_null	Log-likelihood under null model.
df_alter	Degrees of freedom of alternative model.
df_null	Degrees of freedom of null model.
LR_statistic	Likelihood ratio statistic ($-2 * (LL_null - LL_alter)$).
delta_df	Difference in degrees of freedom ($df_alter - df_null$).
p_value	P-value from chi-squared test with $delta_df$ degrees of freedom.

Examples

```
calc_likelihood_ratio_test(stats::rnorm(50), stats::rnorm(50))
```

```
calc_local_structure_metric
```

Local Structure Preservation Metric

Description

Calculates the proportion of overlapping k-nearest neighbors between an original/reference embedding and an integrated/simulated embedding (adapted from Seurat LocalStruct and CellMixS locStructure).

Usage

```
calc_local_structure_metric(coords_pre, coords_post, k = 30)
```

Arguments

coords_pre	Matrix of coordinates before integration / reference (cells x dimensions).
coords_post	Matrix of coordinates after integration / simulated (cells x dimensions).
k	Number of nearest neighbors. Default is 30.

Value

A named list:

cell_overlaps	Numeric vector of overlap fractions per cell
mean_local_structure	Mean overlap fraction across cells (higher indicates better preservation)
median_local_structure	Median overlap fraction

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
calc_local_structure_metric(coords, coords)
```

calc_mad	<i>Calculate Median Absolute Deviation (MAD)</i>
----------	--

Description

Calculate Median Absolute Deviation (MAD)

Usage

```
calc_mad(ref, sim, align = TRUE)
```

Arguments

ref	Numeric vector of reference distribution.
sim	Numeric vector of simulated distribution.
align	Logical, whether to sort and quantile-align vectors. Default TRUE.

Value

Numeric MAD value.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_mad(ref, sim)
```

calc_mae	<i>Calculate Mean Absolute Error (MAE)</i>
----------	--

Description

Calculate Mean Absolute Error (MAE)

Usage

```
calc_mae(ref, sim, align = TRUE)
```

Arguments

ref	Numeric vector of reference values.
sim	Numeric vector of simulated values.
align	Logical, whether to sort and quantile-align vectors. Default TRUE.

Value

Mean absolute difference.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_mae(ref, sim)
```

calc_marginal_aic_bic *Marginal Model Goodness of Fit and Information Criteria for Single-Cell Simulators*

Description

Evaluates gene-wise and aggregate marginal goodness-of-fit (Poisson, Negative Binomial, or Gaussian) for simulated or fitted single-cell count matrices, computing total and mean AIC and BIC across genes (scDesign3; Song et al., 2024).

Usage

```
calc_marginal_aic_bic(
  counts,
  fitted_means,
  dispersions = NULL,
  distribution = c("poisson", "nb", "gaussian"),
  n_params_per_gene = NULL
)
```

Arguments

counts	Matrix or data.frame of observed or simulated counts (genes x cells).
fitted_means	Matrix or data.frame of model fitted means or expectations (genes x cells).
dispersions	Optional vector of gene-level dispersion parameters for Negative Binomial model. If NULL, estimated via method-of-moments.
distribution	Parametric distribution: "poisson", "nb" (Negative Binomial), or "gaussian".
n_params_per_gene	Number of estimated parameters per gene. If NULL, defaults to 1 for Poisson, 2 for NB, and 2 for Gaussian.

Value

A list containing:

total_loglik	Sum of marginal log-likelihoods across all genes and cells.
total_aic	Aggregate AIC across all genes.
total_bic	Aggregate BIC across all genes.
mean_gene_aic	Mean AIC per gene.
mean_gene_bic	Mean BIC per gene.
gene_loglik	Vector of log-likelihoods per gene.
gene_aic	Vector of AIC per gene.
gene_bic	Vector of BIC per gene.

Examples

```
calc_marginal_aic_bic(stats::rnorm(50), stats::rnorm(50))
```

calc_mmd

Maximum Mean Discrepancy (MMD) with Gaussian RBF Kernel

Description

Evaluates the distributional discrepancy between empirical reference and generated/simulated single-cell high-dimensional profiles or latent embeddings.

Usage

```
calc_mmd(
  ref_mat,
  sim_mat,
  cells_as_cols = TRUE,
  sigma = NULL,
  max_cells = 500,
  max_features = 1000
)
```

Arguments

ref_mat	Matrix of reference cells (cells x features or features x cells).
sim_mat	Matrix of simulated cells (cells x features or features x cells).
cells_as_cols	Logical, whether cells are columns. If TRUE (default for scRNA-seq), the matrix is transposed so rows represent cells.
sigma	Optional kernel bandwidth. If NULL, uses the median pairwise distance heuristic.
max_cells	Maximum number of cells to subsample for speed (default 500).
max_features	Maximum number of features to retain for speed (default 1000).

Value

A list with MMD, squared MMD, and the kernel bandwidth sigma used.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)
sim <- matrix(stats::rnorm(100), 50, 2)
calc_mmd(ref, sim)
```

calc_modality_alignment
<i>Modality Alignment & Omics Layer Mixing in Joint Latent Space</i>

Description

Quantifies how well different single-cell omics layers mix in a unified latent embedding space without sacrificing biological clustering (Zhai et al., Genome Biology 2024; Luecken et al., Nat Methods 2022).

Usage

```
calc_modality_alignment(embedding, modalities, cell_types = NULL, k = 15)
```

Arguments

embedding	Joint coordinates matrix (cells x latent_dimensions).
modalities	Factor or character vector indicating the modality of each cell (e.g. "RNA" vs "ATAC").
cell_types	Optional factor or character vector of cell type labels.
k	Number of nearest neighbors for neighborhood connectivity calculation (default 15).

Value

A list containing modality_asw, modality_mixing_score, and mean_cross_modality_neighbor_frac.

Examples

```
data(example_multiomics, package = "scSimEval")
r_rna <- example_multiomics$ref_multi$rna
r_atac <- example_multiomics$ref_multi$atac
calc_modality_alignment(rbind(t(r_rna[seq_len(10),]), t(r_atac[seq_len(10),])), rep(c("RNA", "ATAC"), each=ncol(
```

calc_model_aic_bic	Compute Model Information Criteria (AIC and BIC)
--------------------	--

Description

Evaluates the statistical trade-off between model fit and parameter complexity for single-cell generative models as benchmarked in scDesign3 (Song et al., Nat Biotechnol 2024).

Usage

```
calc_model_aic_bic(loglik, n_params, n_obs)
```

Arguments

loglik	Numeric vector or scalar of log-likelihood values.
n_params	Numeric vector or scalar of number of estimated parameters (degrees of freedom).
n_obs	Total number of independent observations (cells or cell-gene pairs).

Value

A named vector or data.frame containing loglik, n_params, n_obs, AIC, and BIC.

Examples

```
calc_model_aic_bic(-100, 5, 100)
```

calc_neighborhood_purity	Calculate Neighborhood Purity
--------------------------	-------------------------------

Description

Integrated from simpIC (Chugh et al., 2024; bluster::neighborPurity). For each cell, evaluates the proportion of its k-nearest neighbors that share the same cluster or cell-type identity.

Usage

```
calc_neighborhood_purity(data, cluster_labels, k = NULL, is_distance = FALSE)
```

Arguments

data	Matrix of coordinates (cells x dimensions) or expression matrix (features x cells) or dist matrix.
cluster_labels	Vector of cluster or cell-type labels.
k	Number of nearest neighbors (default 5% of cells).
is_distance	Logical, whether data is already a distance matrix. Default FALSE.

Value

A numeric vector of neighborhood purity scores (values from 0 to 1), with an attribute "mean_purity" containing the overall average.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 10))
dist_mat <- stats::dist(data)
calc_neighborhood_purity(data, cl)
```

calc_network_jaccard	<i>Jaccard Similarity of Predicted Regulatory Networks / Links</i>
----------------------	--

Description

Computes the Jaccard similarity index of top-k edge predictions between two networks or top correlated feature pairs between reference and simulated datasets.

Usage

```
calc_network_jaccard(edges1, edges2, k = NULL, directed = TRUE)
```

Arguments

edges1	Data frame or character vector representing the first network.
edges2	Data frame or character vector representing the second network.
k	Integer cutoff for top edge selection (optional).
directed	Logical, whether edges are directed (default TRUE) or undirected (FALSE).

Value

Jaccard similarity index in $[0, 1]$ ($|E1 \cap E2| / |E1 \cup E2|$).

Examples

```
calc_network_jaccard(data.frame(from = c("g1", "g2"), to = c("g2", "g3")),
  data.frame(from = c("g1", "g2"), to = c("g2", "g3")))
```

calc_nmi	<i>Calculate Normalized Mutual Information (NMI)</i>
----------	--

Description

Calculate Normalized Mutual Information (NMI)

Usage

```
calc_nmi(pred, truth)
```

Arguments

pred	Predicted cluster labels.
truth	Ground truth cell type labels.

Value

NMI value between 0 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_nmi(pred, truth)
```

calc_nn_mismatch	<i>Calculate Nearest-Neighbor Label Mismatch Proportion</i>
------------------	---

Description

Integrated from countsimQC (Soneson & Robinson). Evaluates whether the dataset label composition in k-NN neighborhoods departs significantly from the overall global dataset proportion using Chi-squared tests. Works for both 1D numeric vectors and 2D matrices.

Usage

```
calc_nn_mismatch(ref, sim, k = NULL, subsample_size = 300)
```

Arguments

ref	Vector or matrix for reference dataset.
sim	Vector or matrix for simulated dataset.
k	Number of nearest neighbors (default max(5, 0.05 * N)).
subsample_size	Number of subsampled points to test (default 300).

Value

Fraction of points with significant neighbor composition mismatch ($p \leq 0.05$).

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_nn_mismatch(ref, sim)
```

`calc_outlier_proportion`*Calculate Proportion of Outliers in a Numeric Vector*

Description

Uses the standard Tukey $1.5 * \text{IQR}$ criterion on quantiles, as used in `simpipes`.

Usage

```
calc_outlier_proportion(x)
```

Arguments

`x` Numeric vector.

Value

Outlier proportion (between 0 and 1).

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_outlier_proportion(ref)
```

calc_overlap	<i>Calculate Distribution Overlapping Index (OV)</i>
--------------	--

Description

Calculate Distribution Overlapping Index (OV)

Usage

```
calc_overlap(ref, sim)
```

Arguments

ref	Numeric vector of reference values.
sim	Numeric vector of simulated values.

Value

Area of overlap under probability densities (0 to 1).

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_overlap(ref, sim)
```

calc_peacock_2d	<i>Calculate Peacock 2D Kolmogorov-Smirnov Test Statistic</i>
-----------------	---

Description

Integrated from HelenaLC/simulation-comparison.

Usage

```
calc_peacock_2d(ref_mat, sim_mat)
```

Arguments

ref_mat	2-column numeric matrix for reference.
sim_mat	2-column numeric matrix for simulation.

Value

Peacock test statistic.

Examples

```
ref <- matrix(stats::rnorm(100), 50, 2)
sim <- matrix(stats::rnorm(100), 50, 2)
calc_peacock_2d(ref, sim)
```

calc_peak_coaccessibility_fidelity

Evaluate Single-Cell ATAC Peak Co-Accessibility Fidelity (SCRIP)

Description

Evaluates the preservation of chromatin peak-to-peak co-accessibility correlation matrices (cis-regulatory interactions) between reference and simulated scATAC-seq datasets.

Usage

```
calc_peak_coaccessibility_fidelity(
  ref_atac,
  sim_atac,
  top_n_peaks = 300,
  method = c("spearman", "pearson")
)
```

Arguments

ref_atac	Reference scATAC-seq matrix (peaks x cells).
sim_atac	Simulated scATAC-seq matrix (peaks x cells).
top_n_peaks	Number of highest variance peaks to evaluate (default 300).
method	Correlation method: "spearman" (default) or "pearson".

Value

A list containing RV coefficient, matrix correlation, Frobenius distance, and MAE.

Examples

```
data(example_scrna, package = "scSimEval")
calc_peak_coaccessibility_fidelity(example_scrna$ref, example_scrna$sim)
```

calc_pseudotime_correlation

Calculate Correlation of Geodesic Pseudotime Distances

Description

Evaluates whether the relative cell-to-cell ordering along a differentiation trajectory is preserved between reference and simulated data. Accepts either precomputed pseudotime vectors OR scRNA-seq expression matrices (which will be automatically inferred via PCA trajectory).

Usage

```
calc_pseudotime_correlation(
  ref_pseudotime,
  sim_pseudotime,
  method = c("spearman", "pearson")
)
```

Arguments

ref_pseudotime Numeric vector of reference pseudotimes OR reference count matrix (genes x cells).

sim_pseudotime Numeric vector of simulated pseudotimes OR simulated count matrix (genes x cells).

method Correlation method: "spearman" (default) or "pearson".

Value

Correlation coefficient between reference and simulated pseudotime trajectories.

Examples

```
data(example_scrna, package = "scSimEval")
calc_pseudotime_correlation(example_scrna$ref, example_scrna$sim)
```

calc_rmse

Calculate Root Mean Squared Error (RMSE)

Description

Calculate Root Mean Squared Error (RMSE)

Usage

```
calc_rmse(ref, sim, align = TRUE)
```

Arguments

ref	Numeric vector of reference values.
sim	Numeric vector of simulated values.
align	Logical, whether to sort and quantile-align vectors. Default TRUE.

Value

Quadratic error penalizing large discrepancies.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_rmse(ref, sim)
```

calc_runs_test	<i>Calculate Wald-Wolfowitz Runs Test on Pooled Distributions</i>
----------------	---

Description

Integrated from countsimQC (Soneson & Robinson). Tests whether values from the two datasets intermingle randomly when sorted. A significant left-sided result indicates that identical values cluster together into fewer runs than expected by chance, signaling distinct distributions.

Usage

```
calc_runs_test(ref, sim, alternative = c("left.sided", "two.sided"))
```

Arguments

ref	Numeric vector of reference values.
sim	Numeric vector of simulated values.
alternative	Alternative hypothesis: "left.sided" (default in countsimQC) or "two.sided".

Value

A named list with the runs test statistic and p-value.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_runs_test(ref, sim)
```

calc_seurat_mixing_metric
Seurat Mixing Metric

Description

Calculates the mixing metric originally proposed in Seurat (Stuart et al., Cell 2019) and adapted by CellMixS. For each cell, finds the rank of the `k_pos`-th neighbor from each batch in the sorted neighborhood, and computes the median rank across batches.

Usage

```
calc_seurat_mixing_metric(coords, batch_info, k = 300, k_pos = 5)
```

Arguments

<code>coords</code>	Matrix of cell coordinates (cells x dimensions).
<code>batch_info</code>	Factor or character vector of batch assignments.
<code>k</code>	Maximum neighborhood size to search. Default is 300.
<code>k_pos</code>	The rank position to extract per batch. Default is 5.

Value

A named list:

<code>mixing_metrics</code>	Numeric vector of median ranks per cell
<code>mean_mixing_metric</code>	Mean mixing rank (lower indicates better mixing)
<code>median_mixing_metric</code>	Median mixing rank

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
calc_seurat_mixing_metric(coords, batch)
```

calc_signal_bd	<i>Detect Bimodal Distribution (BD) Genes (Bimodal Separation Index)</i>
----------------	--

Description

Detect Bimodal Distribution (BD) Genes (Bimodal Separation Index)

Usage

```
calc_signal_bd(exprs_mat, cell_types)
```

Arguments

exprs_mat	Matrix (genes x cells).
cell_types	Factor of 2 cell types.

Value

Numeric vector of Bimodality Index values.

Examples

```
data(example_scrna, package = "scSimEval")  
calc_signal_bd(example_scrna$ref, example_scrna$cell_types)
```

calc_signal_dd	<i>Detect Differential Distribution (DD) Genes (Kolmogorov-Smirnov Test)</i>
----------------	--

Description

Detect Differential Distribution (DD) Genes (Kolmogorov-Smirnov Test)

Usage

```
calc_signal_dd(exprs_mat, cell_types)
```

Arguments

exprs_mat	Log-normalized matrix (genes x cells).
cell_types	Factor of 2 cell types.

Value

Vector of adjusted p-values.

Examples

```
data(example_scrna, package = "scSimEval")
calc_signal_dd(example_scrna$ref, example_scrna$cell_types)
```

calc_signal_de	<i>Detect Differential Expression (DE) Genes (Mean Shift via Limma or T-Test)</i>
----------------	---

Description

Detect Differential Expression (DE) Genes (Mean Shift via Limma or T-Test)

Usage

```
calc_signal_de(exprs_mat, cell_types, p_sig = 0.05)
```

Arguments

- exprs_mat Log-normalized matrix (genes x cells).
- cell_types Factor or binary vector of 2 cell types.
- p_sig Significance threshold (default 0.05).

Value

Vector of adjusted p-values.

Examples

```
data(example_scrna, package = "scSimEval")
calc_signal_de(example_scrna$ref, example_scrna$cell_types)
```

calc_signal_dp	<i>Detect Differential Proportion / Zero-Inflation (DP) Genes (Chisq Test)</i>
----------------	--

Description

Detect Differential Proportion / Zero-Inflation (DP) Genes (Chisq Test)

Usage

```
calc_signal_dp(exprs_mat, cell_types, threshold = 0)
```

Arguments

- exprs_mat Count or expression matrix (genes x cells).
- cell_types Factor of 2 cell types.
- threshold Value threshold for zero/detection status (default 0).

Value

Vector of adjusted p-values.

Examples

```
data(example_scrna, package = "scSimEval")
calc_signal_dp(example_scrna$ref, example_scrna$cell_types)
```

calc_signal_dv	<i>Detect Differential Variability (DV) Genes (Bartlett Test)</i>
----------------	---

Description

Detect Differential Variability (DV) Genes (Bartlett Test)

Usage

```
calc_signal_dv(exprs_mat, cell_types)
```

Arguments

- exprs_mat Log-normalized matrix (genes x cells).
- cell_types Factor of 2 cell types.

Value

Vector of adjusted p-values.

Examples

```
data(example_scrna, package = "scSimEval")
calc_signal_dv(example_scrna$ref, example_scrna$cell_types)
```

calc_silhouette	<i>Calculate Average Silhouette Width (ASW)</i>
-----------------	---

Description

Calculate Average Silhouette Width (ASW)

Usage

```
calc_silhouette(dist_mat, cluster_labels)
```

Arguments

dist_mat Distance matrix among cells or numeric matrix (features x cells).
cluster_labels Vector of cluster or cell-type labels.

Value

Mean silhouette width.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 10))
dist_mat <- stats::dist(data)
calc_silhouette(dist_mat, cl)
```

calc_tree_height_discrepancy	<i>Calculate Lineage Tree Branch Height Discrepancy (RMSE)</i>
------------------------------	--

Description

Computes the Root Mean Squared Error (RMSE) between sorted branch heights of hierarchical lineage trees. Accepts either precomputed hclust objects OR scRNA-seq count matrices + cell_types (which will be automatically inferred).

Usage

```
calc_tree_height_discrepancy(
  ref_tree,
  sim_tree,
  cell_types_ref = NULL,
  cell_types_sim = NULL
)
```

Arguments

- ref_tree Hierarchical clustering (hclust) object from reference data, OR reference count matrix.
- sim_tree Hierarchical clustering (hclust) object from simulated data, OR simulated count matrix.
- cell_types_ref Optional cell type labels for reference (if count matrix provided).
- cell_types_sim Optional cell type labels for simulation (if count matrix provided).

Value

Root mean squared error between branch heights.

Examples

```
data(example_scrna, package = "scSimEval")
calc_tree_height_discrepancy(example_scrna$ref, example_scrna$sim,
                             cell_types_ref = example_scrna$cell_types,
                             cell_types_sim = example_scrna$cell_types)
```

calc_v_measure	<i>Calculate V-Measure</i>
----------------	----------------------------

Description

Calculate V-Measure

Usage

```
calc_v_measure(pred, truth, beta = 1)
```

Arguments

- pred Vector of predicted cluster labels.
- truth Vector of ground truth cell type labels.
- beta Weight parameter (default 1).

Value

V-measure score between 0 and 1.

Examples

```
pred <- factor(rep(c("A", "B"), each = 20))
truth <- factor(rep(c("A", "B"), each = 20))
calc_v_measure(pred, truth)
```

calc_variance_decomposition

Variance Component Decomposition (BASiCS & muscat)

Description

Decomposes gene expression variance into biological cell-type heterogeneity, donor / biological replicate variation, and residual technical noise.

Usage

```
calc_variance_decomposition(
  counts,
  cell_metadata,
  donor_col = "donor",
  celltype_col = "cell_type",
  n_genes = 200
)
```

Arguments

counts	Matrix or data.frame (genes x cells) or SingleCellExperiment.
cell_metadata	Data.frame containing donor and cell-type columns.
donor_col	Character, column name for donor/replicate (default "donor").
celltype_col	Character, column name for cell identity (default "cell_type").
n_genes	Number of top variable genes to analyze (default 200).

Value

A list of mean percentage of variance explained by cell-type, donor, and residual noise.

Examples

```
data(example_scrna, package = "scSimEval")
md <- data.frame(donor = example_scrna$batch_info,
                 cell_type = example_scrna$cell_types)
calc_variance_decomposition(example_scrna$ref, md)
```

calc_wasserstein_1d *Calculate 1D Wasserstein Metric / Earth Mover's Distance (WS)*

Description

Integrated from HelenaLC/simulation-comparison.

Usage

```
calc_wasserstein_1d(ref, sim, p = 1)
```

Arguments

ref	Numeric vector of reference values.
sim	Numeric vector of simulated values.
p	Power of the Wasserstein metric (default 1 for standard earth mover's distance).

Value

The 1D Wasserstein distance between the two empirical distributions.

Examples

```
ref <- stats::rnorm(50)
sim <- stats::rnorm(50)
calc_wasserstein_1d(ref, sim)
```

calc_zero_probability_curve *Fit Zero-Probability Dropout Curve (Splatter & ZINB-WaVE)*

Description

Fits an empirical logistic dropout curve: $\text{logit}(P(Y=0)) = \beta_0 + \beta_1 * \log(\mu)$ to model the dropout relationship with mean expression.

Usage

```
calc_zero_probability_curve(counts)
```

Arguments

counts	Count matrix (genes x cells) or SingleCellExperiment.
--------	---

Value

A list containing intercept, slope, midpoint (inflection point), and R-squared.

Examples

```
data(example_scrna, package = "scSimEval")
calc_zero_probability_curve(example_scrna$ref)
```

```
compute_dataset_embeddings
```

Compute Low-Dimensional Embeddings for Reference and Simulated Single-Cell Datasets

Description

Projects empirical biological reference and simulated count matrices into low-dimensional representations (UMAP, t-SNE, or PCA) for side-by-side and multi-method topological benchmarking. Automatically performs library-size normalization, variance-based feature selection, principal component analysis, and optional unsupervised clustering.

Usage

```
compute_dataset_embeddings(
  reference,
  simulated,
  reduction = c("umap", "tsne", "pca"),
  n_pcs = 30,
  perplexity = 30,
  n_neighbors = 15,
  min_dist = 0.3,
  seed = 42,
  cell_types = NULL,
  batch = NULL
)
```

Arguments

reference	Biological reference count matrix (features x cells), dgCMatrx, SingleCellExperiment, or Seurat object.
simulated	A single simulated count matrix or a named list of simulated count matrices (e.g. list("Splatter" = sim1, "scDesign3" = sim2)).
reduction	Character string specifying the dimensionality reduction method: "umap" (default), "tsne", or "pca".
n_pcs	Integer specifying the number of principal components to calculate (default: 30).
perplexity	Numeric perplexity for t-SNE (default: 30; automatically adapted for small sample sizes).

n_neighbors	Integer number of nearest neighbors for UMAP (default: 15).
min_dist	Numeric minimum distance parameter for UMAP (default: 0.3).
seed	Random seed for reproducibility (default: 42).
cell_types	Optional factor or character vector of cell type annotations for reference cells (or named list if per-dataset).
batch	Optional factor or character vector of batch annotations.

Value

A tidy data.frame containing cell coordinates (Dim1, Dim2), Dataset name, Dataset_Type ("Reference" vs "Simulated"), Cell_Type, Cluster, Library_Size, Detected_Features, and Batch.

Examples

```
data(example_scrna, package = "scSimEval")
emb <- compute_dataset_embeddings(
  reference = example_scrna$ref,
  simulated = list("Splatter" = example_scrna$sim),
  reduction = "umap",
  cell_types = example_scrna$cell_types
)
head(emb)
```

```
compute_embedding_quality_metrics
```

Compute Quantitative Quality Metrics for Low-Dimensional Cell Embeddings

Description

Computes quantitative cluster separability (Silhouette width), unsupervised cluster recovery (Adjusted Rand Index), and distribution preservation metrics (Mean UMI, Mean detected features, sparsity percentage, and relative percentage error) between reference and simulated datasets.

Usage

```
compute_embedding_quality_metrics(
  embedding_data,
  metric_coords = c("Dim1", "Dim2")
)
```

Arguments

embedding_data	A data.frame produced by compute_dataset_embeddings .
metric_coords	Character vector of the 2 coordinate columns used to compute distances (default: c("Dim1", "Dim2")).

Value

A data.frame summarizing quantitative embedding quality metrics for each dataset.

Examples

```
data(example_scrna, package = "scSimEval")
emb <- compute_dataset_embeddings(
  reference = example_scrna$ref,
  simulated = list("Splatter" = example_scrna$sim),
  reduction = "umap",
  cell_types = example_scrna$cell_types
)
compute_embedding_quality_metrics(emb)
```

compute_method_leaderboard

Compute Simulator Method Performance Leaderboard

Description

Computes the overall performance leaderboard ranking simulation methods across all benchmark evaluation metrics. Metrics are first direction-inverted and min-max standardized into fidelity scores in [0, 1] (where 1.0 represents best observed performance). Simulators are then rank-ordered by mean overall fidelity score.

Usage

```
compute_method_leaderboard(benchmark_data)
```

Arguments

benchmark_data A benchmark data.frame (such as demo\$benchmark_summary_table or output from [evaluate_simulation_accuracy\(\)](#)) or a named list of benchmark result tables.

Value

A data.frame with columns:

- Overall_Rank: Integer rank (1 = top-performing simulator).
- Method: Simulator method name.
- Average_Fidelity: Formatted percentage string (e.g., "58.5%").
- Fidelity_Score: Numeric composite fidelity score in [0, 1] rounded to 4 decimals.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
lb <- compute_method_leaderboard(res)
```

`evaluate_accessibility_sparsity_curve`*Evaluate Accessibility-Sparsity Curve Concordance (simATAC)*

Description

Compares the non-linear relationship between peak mean accessibility and non-zero proportion (NZP) between reference and simulated scATAC-seq datasets.

Usage

```
evaluate_accessibility_sparsity_curve(ref_data, sim_data, poly_degree = 2)
```

Arguments

<code>ref_data</code>	Reference count matrix (peaks x cells).
<code>sim_data</code>	Simulated count matrix (peaks x cells).
<code>poly_degree</code>	Degree of polynomial (default: 2).

Value

A list of reference and simulated curve parameters, absolute discrepancies, and curve prediction RMSE.

Examples

```
data(example_scrna, package = "scSimEval")
evaluate_accessibility_sparsity_curve(example_scrna$ref, example_scrna$sim)
```

`evaluate_batch_metrics`*Calculate Batch Integration Metrics*

Description

Evaluates how realistically simulated technical batches mix or separate, computing kBET, LISI, Batch Silhouette, Shannon Entropy, PC Regression, CMS (Cell-Specific Mixing Score), Inverse Simpson Index (ISI), Seurat Mixing Metric, and optionally Local Density Differences (ldfDiff) and Local Structure Preservation.

Usage

```
evaluate_batch_metrics(
  data,
  batch_info,
  k = NULL,
  n_pcs = 30,
  pre_data = NULL,
  cell_types = NULL,
  verbose = FALSE
)
```

Arguments

<code>data</code>	Count or log-normalized expression matrix (features x cells).
<code>batch_info</code>	Factor or character vector of batch assignments for each cell.
<code>k</code>	Neighborhood size <code>k</code> for <code>k</code> -NN graph and mixing metrics. Default is <code>min(table(batch_info))/2</code> .
<code>n_pcs</code>	Number of principal components for embedding evaluation. Default is 30.
<code>pre_data</code>	Optional matrix of pre-integration / reference expression data (for <code>ldfDiff</code> and local structure).
<code>cell_types</code>	Optional factor or character vector of cell types for cross-batch evaluation.
<code>verbose</code>	Logical, whether to print progress.

Value

A named list of batch metrics:

<code>batch_silhouette</code>	Average silhouette width using batch labels (lower is better mixed)
<code>shannon_entropy</code>	Entropy of batch frequencies in local neighborhoods
<code>pcr_r2</code>	Total variance explained by batch in principal components (PCR)
<code>cms</code>	Mean Cell-specific mixing score (CellMixS)
<code>isi</code>	Mean Inverse Simpson Index for batch mixing (CellMixS)
<code>seurat_mixing_metric</code>	Mean Seurat mixing metric (Stuart et al. / CellMixS)
<code>ldf_diff</code>	Mean Local Density Factor difference (if <code>pre_data</code> supplied)
<code>local_structure</code>	Mean Local structure preservation overlap (if <code>pre_data</code> supplied)
<code>kbet_rejection</code>	kBET rejection rate (if kBET installed)
<code>lisi_batch</code>	Average batch LISI (if lisi installed)
<code>cross_batch_accuracy</code>	Mean cross-batch cell type transfer accuracy (if <code>cell_types</code> supplied)
<code>cross_batch_F1</code>	Mean cross-batch cell type transfer macro F1 (if <code>cell_types</code> supplied)

Examples

```
data(example_scrna, package = "scSimEval")
evaluate_batch_metrics(example_scrna$ref, example_scrna$batch_info,
  cell_types = example_scrna$cell_types)
```

```
evaluate_clustering_metrics
```

Full Clustering Performance Evaluation

Description

Evaluates both unsupervised cluster separation (ASW, Dunn, Connectivity, DB, CH, Neighborhood Purity, CDI) and supervised concordance against ground truth labels (Clustering Accuracy ACC, Hungarian F1/Precision/Recall, ARI, NMI, AMI, FMI, Homogeneity, Completeness, V-measure).

Usage

```
evaluate_clustering_metrics(
  data,
  cluster_info = NULL,
  pred_clusters = NULL,
  dist_mat = NULL,
  cell_types = NULL,
  ref_data = NULL
)
```

Arguments

<code>data</code>	Count or normalized matrix (features x cells).
<code>cluster_info</code>	Cluster or cell type labels.
<code>pred_clusters</code>	Optional predicted cluster labels (if different from ground truth). If NULL, k-means is automatically run.
<code>dist_mat</code>	Optional precomputed distance matrix.
<code>cell_types</code>	Optional alias for <code>cluster_info</code> .
<code>ref_data</code>	Optional reference data to compute reference clustering quality.

Value

A named list of all clustering metrics.

Examples

```
data <- matrix(stats::rnorm(200), 20, 10)
cl <- factor(rep(c("A", "B"), each = 5))
dist_mat <- stats::dist(data)
evaluate_clustering_metrics(data, cl)
```

 evaluate_cross_batch_prediction

Evaluate Cross-Batch Cell Classification Transfer Accuracy

Description

Evaluates whether simulated or integrated multi-batch single-cell data preserve cell-type identity across distinct experimental batches or donors. Trains a k-NN classifier on each individual batch and tests prediction accuracy and macro F1 score on all other remaining batches.

Usage

```
evaluate_cross_batch_prediction(coords, cell_types, batch_info, k = 5)
```

Arguments

coords	Embedding coordinates (cells x dimensions).
cell_types	Factor or character vector of ground-truth cell type labels.
batch_info	Factor or character vector of batch assignments.
k	Integer number of nearest neighbors for the k-NN classifier (default 5).

Value

A list containing:

mean_cross_batch_accuracy	Overall mean classification accuracy across all directed batch pairs
mean_cross_batch_F1	Overall mean macro-averaged F1 score across all directed batch pairs

Examples

```
coords <- matrix(stats::rnorm(100), 50, 2)
batch <- factor(rep(c("B1", "B2"), length.out = 50))
ct <- factor(rep(c("TypeA", "TypeB"), length.out = 50))
evaluate_cross_batch_prediction(coords, ct, batch)
```

evaluate_cross_modal_prediction
<i>Cross-Modal Cell-Type Label Transfer Accuracy</i>

Description

Trains a model using ONLY Modality 1 (e.g., scATAC-seq) and predicts cell-type identities defined in Modality 2 (e.g., scRNA-seq) to test cross-modal biological coherence.

Usage

```
evaluate_cross_modal_prediction(mod1_data, cell_types)
```

Arguments

mod1_data	Matrix for Modality 1 (features x cells).
cell_types	Ground truth cell-type vector from Modality 2.

Value

A list with cross-modal classification accuracy and macro F1 score.

Examples

```
data(example_multiomics, package = "scSimEval")
r_rna <- example_multiomics$ref_multi$rna
r_atac <- example_multiomics$ref_multi$atac
evaluate_cross_modal_prediction(r_rna, example_multiomics$cell_types)
```

evaluate_deg_fidelity	<i>Evaluate Differentially Expressed Gene (DEG) Fidelity</i>
-----------------------	--

Description

Evaluates the preservation of differentially expressed genes (DEGs) and biological signal between empirical reference and simulated datasets across three landmark single-cell benchmarking frameworks:

- **Simpipe (Duo et al., 2024):** True DEG ratio, Distribution score (p-value uniformity via Pearson Chi-Square goodness-of-fit test on remaining genes after DEG removal), and supervised machine learning classification (Accuracy, Precision, Recall, F1) using simulated DEGs to predict cell identity.
- **SimBench (Cao et al., 2021):** Symmetric Mean Absolute Percentage Error (SMAPE) on DEG proportions, SimBench DE fidelity score, log2 fold-change effect size Pearson and Spearman correlation, and top-N DEG Jaccard overlap.
- **Shaky Foundations (Crowell et al., 2023):** Group silhouette separation width, silhouette discrepancy, and Percent Variance Explained (PVE) by group assignment.

Usage

```

evaluate_deg_fidelity(
  ref_data,
  sim_data,
  ref_celltypes,
  sim_celltypes,
  group1 = NULL,
  group2 = NULL,
  fdr_cutoff = 0.05,
  logfc_cutoff = 0.5,
  top_n_de = 100,
  classifier = c("knn", "svm", "rf"),
  run_pvalue_uniformity = TRUE
)

```

Arguments

<code>ref_data</code>	Reference single-cell expression or count matrix (features x cells).
<code>sim_data</code>	Simulated single-cell expression or count matrix (features x cells).
<code>ref_celltypes</code>	Factor or character vector of cell types for reference cells.
<code>sim_celltypes</code>	Factor or character vector of cell types for simulated cells.
<code>group1</code>	Optional character string specifying group 1 name. Default NULL (auto-selects top abundant type).
<code>group2</code>	Optional character string specifying group 2 name. Default NULL (auto-selects second most abundant type).
<code>fdr_cutoff</code>	Adjusted p-value significance threshold. Default is 0.05.
<code>logfc_cutoff</code>	Absolute log2 fold-change cutoff. Default is 0.5.
<code>top_n_de</code>	Number of top ranked DEGs to evaluate for Jaccard overlap and classification. Default is 100.
<code>classifier</code>	Supervised classifier: "knn" (default), "svm", or "rf".
<code>run_pvalue_uniformity</code>	Logical, whether to run Simpipe's Chi-square p-value uniformity test. Default is TRUE.

Value

A list containing:

- `deg_summary_table`: Comprehensive data.frame uniting all 15 metrics across Simpipe, Sim-Bench, and Shaky Foundations.
- `deg_genes_ref`: Character vector of significant DEGs in reference.
- `deg_genes_sim`: Character vector of significant DEGs in simulation.
- `contrast`: String describing the evaluated group contrast.
- `logfc_ref`: Named numeric vector of reference log2 fold-changes.

- logfc_sim: Named numeric vector of simulated log2 fold-changes.
- ml_classification: Detailed classifier performance metrics.
- pvalue_uniformity: Chi-square goodness-of-fit test results.

Examples

```
data(example_scrna, package = "scSimEval")
evaluate_deg_fidelity(example_scrna$ref, example_scrna$sim,
                      ref_celltypes = example_scrna$cell_types,
                      sim_celltypes = example_scrna$cell_types)
```

```
evaluate_epigenomic_annotation
```

Evaluate Epigenomic Supervised Cell-Type Annotation (EpiAnno / SCAN-ATAC-Sim)

Description

Trains a supervised classification model on reference single-cell chromatin accessibility profiles (scATAC-seq / scCAS) and evaluates cell-type prediction / projection accuracy on simulated cells against ground-truth cell-type labels (Chen et al., Nat Mach Intell 2022).

Usage

```
evaluate_epigenomic_annotation(
  ref_data,
  sim_data,
  ref_celltypes,
  sim_celltypes,
  method = c("knn", "centroid"),
  n_pcs = 20,
  k = 5
)
```

Arguments

ref_data	Reference count/accessibility matrix (features x cells).
sim_data	Simulated count/accessibility matrix (features x cells).
ref_celltypes	Factor or character vector of cell types for reference cells.
sim_celltypes	Factor or character vector of true cell types for simulated cells.
method	Classification method: "knn" (k-nearest neighbors on PCA, default) or "centroid" (nearest centroid classifier).
n_pcs	Number of principal components for dimensionality reduction (default 20).
k	Number of nearest neighbors for k-NN (default 5).

Value

A list containing overall accuracy, balanced accuracy, macro F1, macro precision, macro recall, Cohen's kappa, and per-class performance metrics.

Examples

```
data(example_multiomics, package = "scSimEval")
evaluate_epigenomic_annotation(
  example_multiomics$ref_multi$atac,
  example_multiomics$sim_multi$atac,
  ref_celltypes = example_multiomics$cell_types,
  sim_celltypes = example_multiomics$cell_types
)
```

```
evaluate_multiomics_accuracy
```

Master Multiomics Simulation Benchmarking Pipeline

Description

Orchestrates comprehensive benchmarking across all 7 evaluation categories using strictly the user's provided inputs:

- `ref_multi`: List of reference omics matrices (e.g. `list(rna = ..., atac = ...)`).
- `sim_multi`: List of simulated omics matrices (e.g. `list(rna = ..., atac = ...)`).
- `cell_types`: Vector/factor of cell-type identities.
- `batch_info`: Vector/factor of batch or donor annotations.
- Computational resource usage: `cpu_time`, `memory_mb`, `system_time`.
- Trajectory features inferred directly from scRNA-seq counts.

Usage

```
evaluate_multiomics_accuracy(
  ref_multi,
  sim_multi,
  cell_types = NULL,
  batch_info = NULL,
  pairing = c("paired", "unpaired"),
  is_paired = NULL,
  memory_mb = NULL,
  elapsed_time = NULL,
  peak_memory_mb = NULL,
  feature_pairs = NULL,
  compute_bivariate = FALSE,
  threads = 1,
  verbose = TRUE
)
```

Arguments

ref_multi	Named list of reference matrices for Modality 1 and 2.
sim_multi	Named list of simulated matrices for Modality 1 and 2.
cell_types	Optional factor or vector of cell type labels for cells.
batch_info	Optional factor or vector of batch labels for cells.
pairing	Character string specifying whether the multiomics dataset is "paired" (simultaneous co-assay from the same individual cells, e.g., 10x Chromium Multiome, SHARE-seq, SNARE-seq) or "unpaired" (independent profiling of modalities from separate cells of the same biological tissue). Default is "paired". In "unpaired" mode, cell-level pairing metrics (FOSCTTM, Match@1, cross-modal generation, and direct peak-to-gene coupling) are skipped, while population-level cross-modal metrics (cross-modal label transfer, co-expression module fidelity, peak co-accessibility, and accessibility profile concordance) are computed alongside all unimodal metrics.
is_paired	Optional logical alias for pairing. If provided, TRUE corresponds to "paired" and FALSE to "unpaired".
memory_mb	Optional numeric value of peak memory allocation in MB (or peak_memory_mb).
elapsed_time	Optional numeric value of wall-clock elapsed time in seconds.
peak_memory_mb	Optional alias for memory_mb.
feature_pairs	Optional 2-column data.frame of linked feature pairs.
compute_bivariate	Logical, whether to compute 2D bivariate tests. Default FALSE for speed.
threads	Number of CPU threads. Default is 1.
verbose	Logical, whether to print execution progress. Default is TRUE.

Details**Multiomics Metric Measurement Capacity by Experimental Design & Metadata:**

- **Paired Multiomics (e.g. 10x Chromium Multiome, SHARE-seq, SNARE-seq):**
 - *Count Matrices Alone (RNA + ATAC)*: Measures approximately 48 metrics spanning both unimodal distribution layers (Categories I, II, VI, VIII across both modalities) plus Category VII Cross-Modal Coupling (FOSCTTM alignment score, Match@1 nearest neighbor accuracy, cross-modal generative prediction accuracy/F1, RV coefficient, and peak-to-gene regulatory correlation).
 - + *Cell Types*: Unlocks 15–20 additional metrics for cross-modal cell-type concordance and cell-type specific epigenetic promoter/enhancer activity.
 - + *Batch Info*: Unlocks 10–14 additional metrics evaluating technical confounder integration across both RNA and ATAC modalities.
 - + *Both (Full Metadata)*: Unlocks the complete multiomics suite of all 62 metrics across all 8 categories.
- **Unpaired Multiomics (independent scRNA and scATAC from separate cells):**

- *Count Matrices Alone (RNA + ATAC)*: Measures approximately 38–42 metrics spanning unimodal distributions for both layers plus population-level cross-modal metrics (co-expression module fidelity, Network Jaccard similarity, cross-modality correlation).
- + *Cell Types (Crucial for Unpaired)*: Unlocks 15–20 additional metrics for cluster-matched label transfer concordance and cell-type matched regulatory linkage.
- + *Batch Info*: Unlocks 7–12 additional metrics evaluating batch integration.

Value

A list containing detailed results and a unified tidy master summary table.

Examples

```
data(example_multiomics, package = "scSimEval")
evaluate_multiomics_accuracy(example_multiomics$ref_multi, example_multiomics$sim_multi)
```

```
evaluate_multiple_datasets
```

Evaluate Multiple Single-Cell Datasets Simultaneously

Description

Orchestrates simultaneous benchmarking across a collection of single-cell datasets, including paired multiomics (e.g., scRNA-seq and scATAC-seq from the same biological experiment) and unimodal datasets (e.g., standalone scRNA-seq or scATAC-seq). Produces a unified, consolidated summary table, comparative ranking matrix, and dataset overview.

Usage

```
evaluate_multiple_datasets(
  datasets,
  pair_by_prefix = TRUE,
  compute_bivariate = FALSE,
  threads = 1,
  verbose = TRUE
)
```

Arguments

- datasets** A named list of datasets to evaluate. Can be provided in two formats:
- **Format A (Grouped Multiomics & Unimodal)**: A list of dataset specifications, e.g.: `list("Method 1" = list(ref = list(rna = ..., atac = ...), sim = list(rna = ..., atac = ...)), "Method 3" = list(ref = ..., sim = ..., modality = "scATAC-seq"))`

- **Format B (Flat Named Datasets):** A list with named modality datasets, e.g.: `list("Method 1-scRNA-seq" = list(ref = ..., sim = ...), "Method 1-scATAC-seq" = list(ref = ..., sim = ...), "Method 3-scATAC-seq" = list(ref = ..., sim = ...), "Method 4-scRNA-seq" = list(ref = ..., sim = ...))`

`pair_by_prefix` Logical; if TRUE (default), datasets sharing a common root prefix (such as "Method 1-scRNA-seq" and "Method 1-scATAC-seq") are automatically paired as multi-omics for joint cross-modal evaluation. If FALSE, each dataset entry is evaluated independently.

`compute_bivariate` Logical; whether to compute 2D bivariate tests (default FALSE for speed).

`threads` Number of CPU threads for parallel metric computation. Default is 1.

`verbose` Logical; whether to print progress messages. Default is TRUE.

Value

An S3 object of class `scSimEval_consolidated` containing:

- `consolidated_summary_table`: Master tidy data.frame with columns Dataset, Data_Name, Modality, Category, Property, Metric, and Value.
- `consolidated_score_matrix`: Wide comparison matrix comparing all datasets across metrics.
- `dataset_overview`: Aggregate summary table with total metrics and mean scores per dataset.
- `dataset_results`: Named list containing detailed evaluation results for each individual dataset.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

evaluate_predictive_de_model

Predictive Cell Identity Classification Using DE / Top Variable Features

Description

Trains a classifier on simulated features (80% train / 20% test) to verify whether the simulated biological signals are sufficiently predictive of cell identities. If `de_features` is NULL, the top variable features are automatically selected.

Usage

```
evaluate_predictive_de_model(
  data,
  group,
  de_features = NULL,
  n_top = 50,
  method = c("knn", "rf", "svm")
)
```

Arguments

data	Matrix (features x cells).
group	Cell type labels.
de_features	Optional character vector of selected DE feature names.
n_top	Number of top features to auto-select if de_features is NULL (default 50).
method	Classifier: "knn" (default), "rf", or "svm".

Value

A list containing Accuracy, Macro Precision, Macro Recall, and Macro F1 score.

Examples

```
data(example_scrna, package = "scSimEval")
evaluate_predictive_de_model(example_scrna$ref, group = example_scrna$cell_types)
```

evaluate_simbench_signals

Evaluate the 5 SimBench Biological Signal Proportions

Description

Compares proportions of genes exhibiting DE, DV, DD, DP, and BD between empirical reference and simulated datasets.

Usage

```
evaluate_simbench_signals(
  ref_mat,
  sim_mat,
  ref_celltypes,
  sim_celltypes,
  p_sig = 0.05,
  bi_cutoff = 0.3
)
```

Arguments

ref_mat	Reference matrix.
sim_mat	Simulated matrix.
ref_celltypes	Reference cell type labels (subsets to top 2 abundant types).
sim_celltypes	Simulated cell type labels.
p_sig	Significance cutoff (default 0.05).
bi_cutoff	Bimodal index cutoff (default 0.3).

Value

A tidy data.frame comparing biological signal proportions.

Examples

```
data(example_scrna, package = "scSimEval")
evaluate_simbench_signals(example_scrna$ref, example_scrna$sim,
  ref_celltypes = example_scrna$cell_types,
  sim_celltypes = example_scrna$cell_types)
```

evaluate_simulation_accuracy
<i>Evaluate Comprehensive Simulation Accuracy (Unimodal Omics Layer)</i>

Description

Computes distribution distances, cellular and feature-level properties, zero-inflation curves, and manifold distances between reference and simulated datasets.

Usage

```
evaluate_simulation_accuracy(
  ref_data,
  sim_data,
  compute_bivariate = TRUE,
  threads = 1,
  memory_mb = NULL,
  elapsed_time = NULL,
  peak_memory_mb = NULL,
  verbose = TRUE
)
```

Arguments

<code>ref_data</code>	Reference count matrix (features x cells).
<code>sim_data</code>	Simulated count matrix (features x cells).
<code>compute_bivariate</code>	Logical, whether to compute 2D bivariate tests. Default is TRUE.
<code>threads</code>	Number of threads for parallel computation. Default is 1.
<code>memory_mb</code>	Optional numeric value of peak memory allocation in MB (or <code>peak_memory_mb</code>).
<code>elapsed_time</code>	Optional numeric value of wall-clock elapsed time in seconds.
<code>peak_memory_mb</code>	Optional alias for <code>memory_mb</code> .
<code>verbose</code>	Logical, whether to print execution messages. Default is TRUE.

Details

Metric Measurement Capacity by Data & Metadata Mode:

- **Count Matrix Alone (No Metadata):** Measures approximately 23 baseline metrics spanning Category I (Distributional Properties: KS, Wasserstein, MAD, MAE, RMSE, OV, Bhattacharyya, ECDF difference, Runs test, 2D Fasano-Franceschini, 2D Peacock, KDE test, MMD, Fréchet distance), Category II (Correlations & Zero-Inflation: BCV discrepancy, dropout curve midpoint/slope, mean-variance R^2 , cell/gene Pearson correlations), and Category VI (Trajectory & Lineage Dynamics: pseudotime correlation, tree height RMSE).
- **+ Cell Types (Optional):** Unlocks 12–15 additional metrics spanning Category III (Cellular Structure & Concordance: ARI, NMI, AMI, Silhouette width, Dunn, Davies-Bouldin, Calinski-Harabasz, V-measure, Neighborhood Purity, Generative Precision/Recall) and Category V (Biological Signal & Downstream Fidelity: SimBench DE/DP/DM/DB/DV signal fidelity, top DEG Jaccard overlap, Log2FC correlation, classifier accuracy and Macro-F1).
- **+ Batch Annotations (Optional):** Unlocks 7 additional metrics spanning Category IV (Batch Effects & Confounder Mixing: Batch Silhouette width, local neighborhood Shannon Entropy, PCR R^2 , CMS CellMixS, Inverse Simpson Index ISI, Seurat Mixing Metric, LDF difference).
- **Full Metadata (Count Matrix + Cell Types + Batch):** Unlocks the complete unimodal benchmarking suite of up to 55 metrics.

Value

A list containing tidy summary tables and detailed metric lists.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

`evaluate_trajectory_metrics`*Full Trajectory Accuracy Evaluation*

Description

Evaluates trajectory preservation using pseudotime correlation, branch height discrepancy, and univariate distribution distance metrics on pseudotime. If raw count matrices are provided, pseudotime and lineage trees are automatically inferred directly from the scRNA-seq data.

Usage

```
evaluate_trajectory_metrics(  
  ref_data,  
  sim_data,  
  cell_types_ref = NULL,  
  cell_types_sim = NULL  
)
```

Arguments

`ref_data` Reference count matrix (genes x cells) OR numeric vector of pseudotimes.
`sim_data` Simulated count matrix (genes x cells) OR numeric vector of pseudotimes.
`cell_types_ref` Optional vector of reference cell types (for lineage tree inference).
`cell_types_sim` Optional vector of simulated cell types (for lineage tree inference).

Value

A named list of trajectory accuracy metrics.

Examples

```
data(example_scrna, package = "scSimEval")  
evaluate_trajectory_metrics(example_scrna$ref, example_scrna$sim)
```

`evaluate_zero_probability_curve`*Evaluate Zero-Probability Dropout Curve Fidelity (Splatter & ZINB-WaVE)*

Description

Compares empirical dropout trends between reference and simulated datasets.

Usage

```
evaluate_zero_probability_curve(ref_counts, sim_counts)
```

Arguments

ref_counts Reference count matrix.
 sim_counts Simulated count matrix.

Value

Named numeric vector of curve parameter differences.

Examples

```
data(example_scrna, package = "scSimEval")
evaluate_zero_probability_curve(example_scrna$ref, example_scrna$sim)
```

example_batch_info	<i>Example Batch Annotations Vector</i>
--------------------	---

Description

A named factor vector containing technical batch annotations ("Batch1", "Batch2") for 80 single cells, matching the column names of [example_scrna](#) and [example_scatac](#). Useful as a reference template for formatting batch metadata inputs in `scSimEval`.

Usage

```
data(example_batch_info)
```

Format

A named factor with 80 elements across 2 levels: "Batch1" (40 cells) and "Batch2" (40 cells). Names correspond to individual cell barcodes ("Cell_01" to "Cell_80").

Source

Derived from `example_scrna$batch_info`.

Examples

```
data(example_batch_info)
head(example_batch_info)
table(example_batch_info)
```

example_cell_types	<i>Example Cell-Type Annotations Vector</i>
--------------------	---

Description

A named factor vector containing discrete cell-type labels ("TypeA", "TypeB") for 80 single cells, matching the column names of [example_scrna](#) and [example_scatac](#). Useful as a reference template for formatting cell-type metadata inputs in `scSimEval`.

Usage

```
data(example_cell_types)
```

Format

A named factor with 80 elements across 2 levels: "TypeA" (40 cells) and "TypeB" (40 cells). Names correspond to individual cell barcodes ("Cell_01" to "Cell_80").

Source

Derived from `example_scrna$cell_types`.

Examples

```
data(example_cell_types)
head(example_cell_types)
table(example_cell_types)
```

example_multiomics	<i>Synthetic Example Multiomics Benchmarking Dataset</i>
--------------------	--

Description

A unified single-cell multiomics dataset pairing scRNA-seq and scATAC-seq profiles across the same 80 cells, accompanied by cell-type labels, batch annotations, and logged computational scalability metrics.

Usage

```
data(example_multiomics)
```

Format

A list with 5 elements:

ref_multi Named list of reference matrices: rna (60 x 80) and atac (60 x 80).

sim_multi Named list of simulated matrices: rna (60 x 80) and atac (60 x 80).

cell_types A factor of cell-type identities ("TypeA", "TypeB") for the 80 cells.

batch_info A factor of batch annotations ("Batch1", "Batch2") for the 80 cells.

resource_stats A list of computational scalability metrics: cpu_time, memory_mb, system_time, elapsed_time.

Source

Synthetic multiomics benchmark data generated for testing scSimEval.

Examples

```
data(example_multiomics)
names(example_multiomics$ref_multi)
names(example_multiomics$sim_multi)
```

example_scatac

Synthetic Example scATAC-seq Benchmarking Dataset

Description

A synthetic single-cell ATAC sequencing peak accessibility dataset designed for testing, benchmarking, and demonstrating evaluation metrics in scSimEval. Contains paired empirical reference accessibility counts and simulated accessibility counts across 60 peaks and 80 cells matching the cells in example_scrna.

Usage

```
data(example_scatac)
```

Format

A list with 4 elements:

ref A numeric matrix of reference peak accessibility counts (60 peaks x 80 cells).

sim A numeric matrix of simulated peak accessibility counts (60 peaks x 80 cells).

cell_types A factor of cell-type identities ("TypeA", "TypeB") for the 80 cells.

batch_info A factor of batch annotations ("Batch1", "Batch2") for the 80 cells.

Source

Synthetic benchmark data generated via binomial sampling with cell-type specific accessibility shifts.

Examples

```
data(example_scatac)
dim(example_scatac$ref)
dim(example_scatac$sim)
table(example_scatac$batch_info)
```

example_scrna

*Synthetic Example scRNA-seq Benchmarking Dataset***Description**

A synthetic single-cell RNA sequencing dataset designed for testing, benchmarking, and demonstrating evaluation metrics in scSimEval. Contains paired empirical reference counts and simulated counts across 60 genes and 80 cells from two cell types and two batches.

Usage

```
data(example_scrna)
```

Format

A list with 4 elements:

ref A numeric matrix of reference count data (60 genes x 80 cells).

sim A numeric matrix of simulated count data (60 genes x 80 cells).

cell_types A factor of cell-type identities ("TypeA", "TypeB") for the 80 cells.

batch_info A factor of batch annotations ("Batch1", "Batch2") for the 80 cells.

Source

Synthetic benchmark data generated via negative binomial sampling with cell-type specific expression shifts.

Examples

```
data(example_scrna)
dim(example_scrna$ref)
dim(example_scrna$sim)
table(example_scrna$cell_types)
```

```
extract_cell_properties
```

Extract Comprehensive Cell-Level Properties

Description

Computes library size, log-library size, zero fraction / detection frequency, pairwise cell correlation (on top HVGs), cell k-NN hubness, pairwise PCA distance, and outlier proportions.

Usage

```
extract_cell_properties(
  data,
  max_cells_cor = 1000,
  n_top_hvgs = 500,
  verbose = FALSE
)
```

Arguments

<code>data</code>	Count matrix (features x cells).
<code>max_cells_cor</code>	Maximum number of cells for computing pairwise correlation (default 1000).
<code>n_top_hvgs</code>	Number of highly variable genes to use for correlation and PCA (default 500).
<code>verbose</code>	Logical, whether to print progress messages.

Value

A named list of cell-level summary vectors.

Examples

```
data(example_scrna, package = "scSimEval")
extract_cell_properties(example_scrna$ref)
```

```
extract_dataset_summary
```

Extract Dataset Dimensional Properties and Summary Statistics

Description

Computes key dimensional and biological properties from single-cell or multiomics count matrices, including cell counts, feature counts, sparsity percentage, biological group/cell type counts, batch counts, median library size, median detected features per cell, and mean expression level.

Usage

```
extract_dataset_summary(
  mat,
  role = "Reference",
  method_name = "Dataset",
  modality = "scRNA-seq",
  cell_types = NULL,
  batch_info = NULL
)
```

Arguments

<code>mat</code>	A count matrix (genes/features x cells), dgCMatrx, SingleCellExperiment, or Seurat object.
<code>role</code>	Character string describing the role of the dataset (e.g. "Biological Reference" or "Simulated"). Default is "Reference".
<code>method_name</code>	Character string indicating the method or dataset name (e.g. "Empirical Reference", "Splatter"). Default is "Dataset".
<code>modality</code>	Character string describing the molecular modality (e.g. "scRNA-seq", "scATAC-seq"). Default is "scRNA-seq".
<code>cell_types</code>	Optional factor or character vector of cell type annotations.
<code>batch_info</code>	Optional factor or character vector of batch annotations.

Value

A data.frame with 1 row summarizing the dataset properties.

Examples

```
data(example_scrna, package = "scSimEval")
extract_dataset_summary(example_scrna$ref, role = "Reference", method_name = "Empirical")
```

extract_feature_properties

Extract Comprehensive Feature-Level (Gene / Peak) Properties

Description

Computes mean abundance, standard deviation, variance, coefficient of variation (CV), feature dropout (zero fraction), dispersion, Biological Coefficient of Variation (BCV), gene-gene correlations, and outlier proportions.

Usage

```
extract_feature_properties(data, n_top_cor = 400, verbose = FALSE)
```

Arguments

- data Count matrix (features x cells).
- n_top_cor Number of top HVGs to use for gene-gene correlation matrix (default 400).
- verbose Logical, whether to print progress messages.

Value

A named list of feature-level summary vectors.

Examples

```
data(example_scrna, package = "scSimEval")
extract_feature_properties(example_scrna$ref)
```

fast_col_vars	<i>Fast Column-Wise Variances for Dense or Sparse Single-Cell Matrices</i>
---------------	--

Description

Fast Column-Wise Variances for Dense or Sparse Single-Cell Matrices

Usage

```
fast_col_vars(mat)
```

Arguments

- mat Matrix or dgCMatrix (features x cells).

Value

Numeric vector of column variances.

fast_cross_distance_sq	<i>Fast BLAS-Accelerated Cross-Distance Squared Matrix</i>
------------------------	--

Description

Computes $\|X_i - Y_j\|^2$ using BLAS matrix product: $\|x\|^2 + \|y\|^2 - 2 \langle x, y \rangle$.

Usage

```
fast_cross_distance_sq(X, Y)
```

Arguments

- X Matrix (m x p).
- Y Matrix (n x p).

Value

Matrix of squared Euclidean distances (m x n).

fast_pca	<i>Fast Truncated PCA on Top Highly Variable Features</i>
----------	---

Description

Fast Truncated PCA on Top Highly Variable Features

Usage

fast_pca(mat, n_pcs = 20, n_hvg = 2000, scale = FALSE)

Arguments

- mat Matrix or dgCMatrix (cells x features).
- n_pcs Number of principal components. Default 20.
- n_hvg Number of top highly variable genes to retain. Default 2000.
- scale Logical, whether to scale features. Default FALSE.

Value

Matrix of cell principal component embeddings (cells x n_pcs).

fast_row_vars	<i>Fast Row-Wise Variances for Dense or Sparse Single-Cell Matrices</i>
---------------	---

Description

Fast Row-Wise Variances for Dense or Sparse Single-Cell Matrices

Usage

fast_row_vars(mat)

Arguments

- mat Matrix or dgCMatrix (features x cells).

Value

Numeric vector of row variances.

fast_sparse_zero_fraction_cell
<i>Fast Zero Fraction per Cell (Column-wise)</i>

Description

Fast Zero Fraction per Cell (Column-wise)

Usage

fast_sparse_zero_fraction_cell(mat)

Arguments

mat Matrix or dgCMatrix (features x cells).

Value

Numeric vector of zero fraction per cell.

fast_sparse_zero_fraction_feature
<i>Fast Zero Fraction per Feature (Row-wise)</i>

Description

Fast Zero Fraction per Feature (Row-wise)

Usage

fast_sparse_zero_fraction_feature(mat)

Arguments

mat Matrix or dgCMatrix (features x cells).

Value

Numeric vector of zero fraction per feature.

`infer_scrna_lineage_tree`*Automatically Infer Lineage Tree (hclust) from scRNA-seq Counts & Cell Types*

Description

Builds a hierarchical clustering lineage tree between cell-type centroids from scRNA-seq expression data.

Usage

```
infer_scrna_lineage_tree(data, cell_types, method = "ward.D2")
```

Arguments

<code>data</code>	Expression or count matrix (genes x cells).
<code>cell_types</code>	Factor or vector of cell-type annotations for each cell.
<code>method</code>	Linkage method for hierarchical clustering (default "ward.D2").

Value

An hclust tree object representing the cell-type lineage tree.

Examples

```
data(example_scrna, package = "scSimEval")
infer_scrna_lineage_tree(example_scrna$ref, example_scrna$cell_types)
```

`infer_scrna_pseudotime`*Automatically Infer Pseudotime Trajectory from scRNA-seq Counts*

Description

Estimates cell differentiation pseudotime directly from single-cell expression counts using diffusion/principal curve projection along the first principal component of top highly variable genes.

Usage

```
infer_scrna_pseudotime(data, n_top = 500)
```

Arguments

<code>data</code>	Expression or count matrix (genes x cells).
<code>n_top</code>	Number of top variable genes to use (default 500).

Value

Numeric vector of pseudotime values in [0, 1] for each cell.

Examples

```
data(example_scrna, package = "scSimEval")
infer_scrna_pseudotime(example_scrna$ref)
```

launch_scSimEval_app *Launch Interactive scSimEval Benchmarking Studio*

Description

Launches the Shiny web application embedded in the **scSimEval** package. Provides an intuitive graphical interface to upload biological reference and simulated datasets (single-cell scRNA-seq, scATAC-seq, or paired multiomics), specify computational scalability metrics (elapsed time and peak memory), interactively inspect the flagship 62-measure comparative bubble matrix, explore diagnostic figures, and export high-resolution figures (600 DPI publication quality), multi-page PDF reports, Excel (.xlsx) workbooks, and complete results ZIP packages.

Usage

```
launch_scSimEval_app(
  port = NULL,
  host = "127.0.0.1",
  launch.browser = interactive(),
  maxRequestSize = 10 * 1024^3
)
```

Arguments

port	Optional port number for the local web server. Default is NULL (random open port).
host	Character string specifying the IP address to listen on. Defaults to "127.0.0.1".
launch.browser	Logical, whether to automatically launch the default web browser. Defaults to TRUE in interactive sessions.
maxRequestSize	Numeric value specifying the maximum allowable upload file size in bytes. Defaults to $10 * 1024^3$ (10 GB) to comfortably accommodate large single-cell multiomics matrices.

Value

Invisibly returns the Shiny app process object.

Examples

```
# Locate the embedded Shiny app directory bundled with the package
system.file("shiny", "scSimEvalApp", package = "scSimEval")

# Launch the interactive studio (interactive sessions only)
if (interactive()) {
  launch_scSimEval_app()
}
```

plot_benchmark_bubble_matrix

Plot Multi-Dimensional Benchmarking Bubble Matrix

Description

Produces a multi-dimensional benchmarking bubble matrix comparing simulation methods. Methods appear as rows; evaluation metrics appear as columns grouped under the eight canonical evaluation categories (I-VIII) displayed as colored header strips at the top.

Usage

```
plot_benchmark_bubble_matrix(
  data,
  method_classes = NULL,
  category_colors = NULL,
  metrics_order = NULL,
  methods_order = NULL,
  rank_methods = TRUE,
  compact_strips = TRUE,
  normalize_scores = TRUE,
  show_missing_dots = TRUE,
  bubble_size_range = c(2, 7.8),
  title = NULL,
  subtitle = NULL,
  base_size = 11,
  show_score_labels = FALSE
)
```

```
plot_bubble_matrix(
  data,
  method_classes = NULL,
  category_colors = NULL,
  metrics_order = NULL,
  methods_order = NULL,
  rank_methods = TRUE,
  compact_strips = TRUE,
```

```

normalize_scores = TRUE,
show_missing_dots = TRUE,
bubble_size_range = c(2, 7.8),
title = NULL,
subtitle = NULL,
base_size = 11,
show_score_labels = FALSE
)

```

Arguments

data	Input data. One of: • An scSimEval_consolidated object from evaluate_multiple_datasets(). • A named list of benchmark outputs from evaluate_simulation_accuracy(). • A tidy data.frame with columns: Method, Metric, Category, and Score (or Value).
method_classes	Optional named list mapping method names to row group labels. E.g. <code>list("scRNA-seq" = c("Splatter", "SymSim"), "Multiomics" = c("scDesign3", "dyngen"))</code> .
category_colors	Optional named character vector to override the canonical palette.
metrics_order	Optional character vector for explicit metric ordering (left to right).
methods_order	Optional character vector for explicit method ordering (top to bottom).
rank_methods	Logical. Automatically rank methods along the vertical axis by mean overall fidelity score. Default TRUE.
compact_strips	Logical. Use compact two-line facet strip titles for clean horizontal layout. Default TRUE.
normalize_scores	Logical. Normalize raw distances into [0, 1] fidelity scores. Default TRUE.
show_missing_dots	Logical. Show grey dots for missing metric combinations. Default TRUE.
bubble_size_range	Numeric vector of length 2: min and max bubble size. Default <code>c(2.0, 7.8)</code> .
title	Character. Main plot title. If NULL, dynamically derived from metric count.
subtitle	Character. Plot subtitle.
base_size	Numeric. Base font size. Default 11.
show_score_labels	Logical. Annotate bubbles with numeric score. Default FALSE.

Details

Bubble encoding:

- **Size:** Normalized fidelity score [0, 1] – larger = better performance.
- **Fill color:** Evaluation category (one of the eight canonical groups).
- **Circle (shape 21):** Regular performance (score < 0.96).

- **Square (shape 22)**: Top performer (score ≥ 0.96).
- **Small grey dot**: Metric not available/computed for that method.

Score normalization is direction-aware: distance/error metrics are inverted so that 1 always means best performance, while concordance/correlation metrics are used directly (higher = better).

Value

A ggplot object rendering the multi-dimensional bubble matrix.

Examples

```
# Tidy benchmark data frame of standardized scores
bm_df <- data.frame(
  Method = rep(c("Splat", "scDesign3", "SymSim", "dyngen"), each = 4),
  Category = rep(c("Accuracy", "Accuracy", "Cellular Structure", "Scalability"), 4),
  Metric = rep(c("KS Distance", "Wasserstein", "Silhouette ASW", "CPU Time"), 4),
  Score = c(0.8, 0.7, 0.9, 0.6, 0.5, 0.8, 0.6, 0.5, 0.8, 0.7, 0.6, 0.5, 0.7, 0.6, 0.9, 0.5)
)
p <- plot_benchmark_bubble_matrix(bm_df)
p

# Load benchmark summary across simulators
demo_file <- system.file("shiny/scSimEvalApp/data/demo_benchmark_data.rds", package = "scSimEval")
if (file.exists(demo_file)) {
  demo <- readRDS(demo_file)
  p <- plot_benchmark_bubble_matrix(
    data = demo$benchmark_summary_table,
    title = "Single-Cell Multiomics Simulation Benchmark",
    method_classes = list(
      "scRNA-seq" = c("Splatter", "SymSim"),
      "scATAC-seq" = c("simATAC", "SCRIP"),
      "Multiomics" = c("scDesign3", "dyngen")
    )
  )
  print(p)
}
```

plot_category_metric_bars

Plot Benchmark Metric Performance Barplots by Category

Description

Generates publication-ready faceted barplots comparing simulator methods across all individual evaluation metrics belonging to a specified benchmark category. Each metric is presented in its own sub-panel with exact numeric score labels, direction-awareness indicators (indicating whether higher or lower values are optimal), and standardized or original raw score formulation.

Usage

```
plot_category_metric_bars(
  benchmark_data,
  category,
  score_type = c("normalized", "raw"),
  palette = NULL,
  ncol = NULL,
  base_size = 11
)
```

Arguments

benchmark_data	A benchmark data.frame (e.g. demo\$benchmark_summary_table or output from <code>evaluate_simulation_accuracy()</code>) or a named list of benchmark result tables.
category	Character. The benchmark category to visualize (e.g., "(I) Distributional Properties" or "Distributional Properties").
score_type	Character. Either "normalized" (standardized [0, 1] fidelity score where 1.0 is optimal; default) or "raw" (unnormalized original metric values in native measurement units).
palette	Optional named character vector of simulator colors.
ncol	Integer. Optional number of columns in facet layout. Defaults to dynamic layout based on metric count.
base_size	Numeric. Base font size. Default 11.

Value

A ggplot object.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

plot_dataset_embeddings

Plot High-Dimensional Dataset Embeddings for Reference and Simulated Single-Cell Data

Description

Generates publication-ready comparative visualizations of empirical reference and simulated single-cell datasets across UMAP, t-SNE, or PCA coordinate spaces. Supports side-by-side comparison, comprehensive multi-method facet grids, and co-embedded overlays, with customizable color mappings and themes.

Usage

```
plot_dataset_embeddings(
  embedding_data,
  reduction = c("umap", "tsne", "pca"),
  layout = c("facet", "side_by_side", "overlay"),
  color_by = c("cell_type", "cluster", "library_size", "dataset", "batch"),
  selected_methods = NULL,
  pt_size = 0.8,
  alpha = 0.75,
  palette = "Set1",
  title = NULL,
  base_size = 12
)
```

Arguments

embedding_data	A data.frame produced by compute_dataset_embeddings .
reduction	Character string specifying reduction used: "umap", "tsne", or "pca".
layout	Character string specifying the comparison layout: "facet" Faceted grid showing Reference alongside all selected simulated datasets simultaneously (default). "side_by_side" Direct side-by-side comparison of Reference versus a single chosen simulator. "overlay" Overlaid single coordinate space showing Reference and Simulated cells together.
color_by	Feature used for coloring cells: "cell_type" (default), "cluster", "library_size", "dataset", or "batch".
selected_methods	Optional character vector of simulator names to display. If NULL, displays all available simulators.
pt_size	Numeric point size (default: 0.8).
alpha	Numeric transparency in [0, 1] (default: 0.75).
palette	Character string specifying discrete color palette from RColorBrewer (default: "Set1").
title	Optional title character string. If NULL, an informative default title is generated.
base_size	Base font size for ggplot2 rendering (default: 12).

Value

A ggplot2 object.

Examples

```
data(example_scrna, package = "scSimEval")
emb <- compute_dataset_embeddings(
  reference = example_scrna$ref,
```

```

simulated = list("Splatter" = example_scrna$sim),
reduction = "umap",
cell_types = example_scrna$cell_types
)
plot_dataset_embeddings(emb, reduction = "umap", layout = "facet")

```

plot_distribution_qc *Plot Single-Cell Summary Distribution Quality (Comparative Overlays)*

Description

Compares empirical reference and simulated datasets across canonical single-cell summary properties: library size, TMM, effective library size, mean expression, feature variance, scaled variance, fraction zero per cell, fraction zero per gene, cell-cell correlation, gene-gene correlation, and bivariate relationships (mean vs. variance, mean vs. zero fraction, library size vs. zero fraction, and biological signals). Designed after landmark benchmarking studies (countsimQC / Duo et al.) with a clean, human-crafted scientific layout featuring solid visible contour lines, semi-transparent color-filled density curves, and simple understated titles.

Usage

```

plot_distribution_qc(
  ref_data,
  sim_data,
  layout = c("comprehensive", "density"),
  properties = NULL,
  title = "Data properties",
  ref_name = "Original",
  sim_name = "Simulation",
  palette = NULL,
  cell_types = NULL,
  base_size = 9
)

```

Arguments

ref_data	Numeric count matrix for empirical reference (features x cells).
sim_data	Numeric count matrix for simulated data (features x cells), or a named list of simulated count matrices representing multiple simulators (e.g. <code>list("scDesign3" = m1, "Splatter" = m2)</code>).
layout	Character. Visualization layout: "comprehensive" (default 14-panel benchmarking grid matching canonical single-cell papers) or "density" (faceted density curves only).
properties	Optional character vector of specific properties to include if custom selection is desired.

title	Character. Main plot title. Default "Data properties". Set to NULL for no title.
ref_name	Character. Label for the empirical reference in the legend. Default "Original".
sim_name	Character. Label for the simulation if a single matrix is provided. Default "Simulation".
palette	Optional named character vector of colors. Default uses warm red for reference and classic steel blue for simulation.
cell_types	Optional factor or character vector of cell identities for computing biological signal proportions.
base_size	Numeric. Base font size. Default 9.

Details

Value

A ggplot or patchwork composite object containing the comparative panels.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
data(example_scrna)
p <- plot_distribution_qc(example_scrna$ref, example_scrna$sim)
if (requireNamespace("ggplot2", quietly = TRUE)) print(p)
```

plot_evaluation_summary

Plot Single-Cell Simulator Evaluation Summary

Description

Produces a 600 DPI horizontal bar matrix ranking single-cell simulators across the 8 canonical evaluation categories and overall composite performance.

Usage

```
plot_evaluation_summary(
  data,
  method_classes = NULL,
  category_colors = NULL,
  normalize_scores = TRUE,
  simulators_order = NULL,
  show_labels = TRUE,
  title = "Single-Cell Simulator Evaluation Summary",
  subtitle =
```

```

    "Average fidelity scores across 8 canonical evaluation categories and overall benchmark performance"
    base_size = 11
  )

plot_benchmark_summary_bars(
  data,
  method_classes = NULL,
  category_colors = NULL,
  normalize_scores = TRUE,
  simulators_order = NULL,
  show_labels = TRUE,
  title = "Single-Cell Simulator Evaluation Summary",
  subtitle =
    "Average fidelity scores across 8 canonical evaluation categories and overall benchmark performance"
  base_size = 11
)

plot_summary_bars(
  data,
  method_classes = NULL,
  category_colors = NULL,
  normalize_scores = TRUE,
  simulators_order = NULL,
  show_labels = TRUE,
  title = "Single-Cell Simulator Evaluation Summary",
  subtitle =
    "Average fidelity scores across 8 canonical evaluation categories and overall benchmark performance"
  base_size = 11
)

```

Arguments

<code>data</code>	A benchmark summary data frame, a list containing <code>benchmark_summary_table</code> , or the output of <code>evaluate_simulation_accuracy</code> .
<code>method_classes</code>	Optional named vector mapping method names to class/category labels (e.g., <code>c("Splatter" = "Class 1", "scDesign3" = "Class 2")</code>). If provided, rows are grouped into class panels.
<code>category_colors</code>	Optional named vector specifying custom colors for category panels and overall score.
<code>normalize_scores</code>	Logical indicating whether to normalize scores to [0, 1] (direction-aware where larger is better). Default is TRUE.
<code>simulators_order</code>	Optional vector specifying a custom ordering of simulators. Default is ranking by descending Overall Score.
<code>show_labels</code>	Logical indicating whether to display numerical score labels on/beside each bar. Default is TRUE.

title	Character plot title. Default is "Single-Cell Simulator Evaluation Summary".
subtitle	Character plot subtitle.
base_size	Numeric base font size (default 11).

Details

Value

A ggplot object representing the multi-panel evaluation summary.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)

demo <- readRDS(system.file("shiny/scSimEvalApp/data/demo_benchmark_data.rds", package = "scSimEval"))
p <- plot_evaluation_summary(demo$benchmark_summary_table)
print(p)

data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

plot_individual_metric_bar

Plot Individual Metric Performance Barplot

Description

Generates a publication-quality barplot comparing simulator methods on a single evaluation metric, with exact score labels, direction awareness, and performance ranking.

Usage

```
plot_individual_metric_bar(
  benchmark_data,
  metric,
  score_type = c("normalized", "raw"),
  palette = NULL,
  base_size = 12
)
```

Arguments

`benchmark_data` A benchmark data.frame or named list containing benchmark tables.

`metric` Character. The name of the metric to plot (e.g. "KS", "Wasserstein", "ARI").

`score_type` Character. Either "normalized" (standardized [0, 1] fidelity score; default) or "raw" (original unnormalized metric value).

`palette` Optional named character vector of simulator colors.

`base_size` Numeric. Base font size. Default 12.

Value

A ggplot object.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

`plot_metric_boxplots` *Plot Comparison Boxplots Across Simulators Plot Benchmark Metric Score Distributions Across Simulators*

Description

Generates 600 DPI boxplots with jittered points showing the distribution of evaluation metric scores across simulated single-cell and multiomics datasets. Supports standardized direction-aligned fidelity scores (where higher is universally superior, resolving metric polarity differences across distance and correlation measures) or original unnormalized raw values with optional direction separation or per-metric faceting.

Usage

```
plot_metric_boxplots(
  benchmark_data,
  categories = NULL,
  metrics = NULL,
  score_type = c("normalized", "raw"),
  separate_direction = FALSE,
  facet_by = c("category", "metric"),
  palette = NULL,
  ncol = 4,
  base_size = 11
)
```

Arguments

benchmark_data	A data frame (such as demo\$benchmark_summary_table or output from evaluate_simulation_accuracy) or a named list of benchmark result tables.
categories	Optional character vector of categories to include. Default NULL (all 8 categories).
metrics	Optional character vector of specific metrics to include. Default NULL (all).
score_type	Character. Score formulation: "normalized" (default; direction-inverted standardized fidelity scores in [0, 1] where higher is universally superior across all measures) or "raw" (unnormalized original metric values).
separate_direction	Logical. When score_type = "raw", whether to separate metrics by optimization direction (lower vs higher is better) into distinct sub-panels. Default FALSE.
facet_by	Character. Facet layout: "category" (default; 8 canonical category panels) or "metric" (independent panels for each individual metric).
palette	Optional named character vector of simulator colors.
ncol	Integer. Number of columns in facet wrap. Default 4.
base_size	Numeric. Base font size. Default 11.

Details

Value

A ggplot object.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

plot_metric_heatmap	<i>Plot Multi-Simulator Comparative Metric Heatmap Across Canonical Categories</i>
---------------------	--

Description

Produces a 600 DPI comparative heatmap displaying simulators on the x-axis and evaluated biological/computational measures on the y-axis, grouped by the 8 canonical evaluation categories. Displays the exact original raw evaluation score in text inside every cell, with cell fill colors scaled by relative fidelity (direction-aware, ensuring balanced visual contrast across all 62 measures without scale distortion from high-magnitude metrics like RAM or runtime).

Usage

```
plot_metric_heatmap(
  benchmark_data,
  category = NULL,
  metrics = NULL,
  scale_fill = c("relative", "raw"),
  top_n_properties = NULL,
  facet_by_category = TRUE,
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  base_size = 10,
  metric_name = NULL
)
```

Arguments

<code>benchmark_data</code>	A tidy benchmark summary table or a named list of benchmark tables.
<code>category</code>	Optional character vector to filter by canonical evaluation category. Default NULL (all 8 categories).
<code>metrics</code>	Optional character vector of specific metrics to include. Default NULL (all).
<code>scale_fill</code>	Character. Cell color fill scaling: "relative" (default; direction-aware min-max normalized fidelity in [0, 1] across methods per metric, ensuring balanced color contrast across all 62 measures while cell text displays exact raw numbers) or "raw" (direct numeric values).
<code>top_n_properties</code>	Optional integer. If specified, restricts to top N most variable metrics. Default NULL (all metrics).
<code>facet_by_category</code>	Logical. Whether to group rows into vertical category banner panels across all 8 categories. Default TRUE.
<code>cluster_rows</code>	Logical. Hierarchically cluster measures along y-axis. Default FALSE.
<code>cluster_cols</code>	Logical. Hierarchically cluster simulators along x-axis. Default FALSE.
<code>base_size</code>	Numeric. Base font size. Default 10.
<code>metric_name</code>	Optional legacy alias for metrics.

Details**Value**

A ggplot object.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

plot_metric_mds	<i>Multi-Dimensional Scaling (MDS) Ordination of Evaluation Metrics or Simulators</i>
-----------------	---

Description

Projects evaluation metric profiles or simulator performances into a 2D MDS space for benchmarking single-cell simulation methods.

Usage

```
plot_metric_mds(
  benchmark_data,
  ordination_by = c("simulators", "summaries"),
  by_category = FALSE,
  as_list = FALSE,
  category = NULL,
  exclude_categories = c("(VI) Trajectory & Lineage Dynamics",
    "(VIII) Computational Scalability"),
  metrics = NULL,
  normalize_scores = TRUE,
  palette = NULL,
  title = NULL,
  base_size = 11,
  ncol = 2
)
```

Arguments

benchmark_data	Benchmark summary data.frame, matrix, or list.
ordination_by	Character. Either "summaries" (default; ordinated points represent biological and computational summaries) or "simulators" (ordinated points represent candidate simulators).
by_category	Logical. If TRUE and ordination_by = "summaries", generates a multi-panel grid displaying an MDS ordination for each evaluation category separately (skipping categories with fewer than 3 measures). Default is FALSE.
as_list	Logical. If TRUE and by_category = TRUE, returns a named list of ggplot objects. Default FALSE.
category	Optional character vector to filter by evaluation category (e.g. "(I) Distributional Properties").
exclude_categories	Character vector of categories to exclude from the ordination. Default is c("(VI) Trajectory & Lineage Dynamics", "(VIII) Computational Scalability") because these two categories contain only two measures each, leaving the 6 canonical categories.

metrics	Optional character vector of specific metrics to include.
normalize_scores	Logical. If TRUE (default), scores are direction-aware normalized to [0, 1] so that metrics on disparate scales (e.g. MB vs distance) do not artificially distort Euclidean distance.
palette	Optional named character vector of colors. For ordination_by = "summaries", maps "gene", "cell", and "global".
title	Optional plot title.
base_size	Numeric. Base font size. Default 11.
ncol	Integer. Number of columns when by_category = TRUE. Default 2.

Details

When ordination_by = "summaries", each point represents an evaluated property/metric, colored by its biological level: "gene" (red), "cell" (blue), or "global" (green). If by_category = TRUE, a multi-panel dashboard is produced showing one MDS ordination per category (by default across the 6 canonical categories with >= 3 measures, excluding Trajectory and Scalability which each contain only 2 measures).

Value

A ggplot or patchwork object.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

plot_metric_pca	<i>Principal Component Analysis (PCA) Dashboard of Benchmark Metrics and Methods</i>
-----------------	--

Description

Generates a dual-panel PCA dashboard for evaluating single-cell simulation benchmark metrics and methods.

Usage

```
plot_metric_pca(
  benchmark_data,
  panel = c("both", "methods", "loadings"),
  by_category = FALSE,
  as_list = FALSE,
  category = NULL,
  exclude_categories = NULL,
```

```

    metrics = NULL,
    palette = NULL,
    top_n_loadings = 14,
    base_size = 11
  )

```

Arguments

benchmark_data	Benchmark summary data.frame, matrix, or list.
panel	Character. Which panel(s) to render: "both" (default; 2-panel stacked layout), "methods" (panel a only), or "loadings" (panel b only).
by_category	Logical. If TRUE, perform separate PCA ordinations per evaluation category. Default FALSE.
as_list	Logical. If TRUE and by_category = TRUE, returns a named list of ggplot objects. Default FALSE.
category	Optional character vector to filter by evaluation category.
exclude_categories	Optional character vector of categories to exclude (e.g. scalability).
metrics	Optional character vector of specific metrics to include.
palette	Optional named character vector of colors for simulators.
top_n_loadings	Integer or NULL. Maximum number of loading vectors to plot in panel b (ordered by vector magnitude in PC1-PC2 space). Default is 14 to prevent visual clutter; set to NULL or 62 to display all evaluated metrics.
base_size	Numeric. Base font size. Default 11.

Details

All metrics present in benchmark_data (e.g. all 62 canonical metrics across categories) are mathematically considered in the PCA decomposition. Panel "both" displays:

- **Panel a (Methods Ordination):** Simulators mapped into PC1 vs PC2 coordinates with solid black crosshairs, centroid markers, and white rounded badge labels with colored borders.
- **Panel b (PC Loadings Vectors):** Directional loading vectors (arrows) radiating from the origin (0, 0) to loading coordinates (w1, w2), colored by summary type ("gene" = red, "cell" = blue, "global" = green) with rounded badge labels. Pass top_n_loadings = NULL or 62 to render all metric loading vectors.

Value

A ggplot or patchwork object representing the ordination figure.

Examples

```

data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)

```

plot_scalability_benchmark
<i>Plot Multi-Dimensional Computational Scalability Benchmark</i>

Description

Evaluates computational resource efficiency across single-cell simulators using raw benchmarking measurements (elapsed real wall-clock time, total CPU time, peak resident Plot Multi-Dimensional Computational Scalability Benchmark

Usage

```
plot_scalability_benchmark(  
  benchmark_data,  
  type = c("composite", "runtime", "memory", "tradeoff", "cost", "throughput",  
    "efficiency"),  
  layout = c("4panel", "6panel"),  
  cell_count = 1000,  
  palette = NULL,  
  base_size = 11  
)
```

Arguments

benchmark_data	Benchmark summary table or named list containing Category VIII metrics.
type	Character. Visualization type: "composite" (multi-panel dashboard), "runtime" (elapsed wall-clock time barplot), "memory" (peak RAM barplot), "tradeoff" (runtime vs memory scatter/line curve), "cost" (combined resource footprint index), "throughput" (cells/sec), or "efficiency" (CPU/elapsed concurrency ratio if CPU time available). Default "composite".
layout	Character. For composite dashboards: "4panel" (runtime, memory, tradeoff, cost footprint) or "6panel" (adds throughput). Default "4panel".
cell_count	Integer. Number of cells processed in the benchmark, used to calculate simulation throughput (cells / second). Default 1000.
palette	Optional named character vector of simulator colors.
base_size	Numeric. Base font size. Default 11.

Details

Evaluates computational resource efficiency across single-cell simulators using raw benchmarking measurements (elapsed real wall-clock time in seconds and peak resident memory consumption in MiB).

Value

A ggplot or patchwork object.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(example_scrna$ref, example_scrna$sim)
```

subsample_dist_vector	<i>Subsample Large Vectors for Accelerated Distributional Distance Estimation</i>
-----------------------	---

Description

Subsample Large Vectors for Accelerated Distributional Distance Estimation

Usage

```
subsample_dist_vector(x, max_n = 3000)
```

Arguments

x	Numeric vector.
max_n	Maximum sample size (default 3000).

Value

Subsampled numeric vector if $\text{length}(x) > \text{max_n}$, else x.

Index

* datasets

- example_batch_info, [78](#)
- example_cell_types, [79](#)
- example_multiomics, [79](#)
- example_scatac, [80](#)
- example_scrna, [81](#)

- benchmark_resource_usage, [5](#)
- benchmark_simulation_suite, [6](#)

- calc_accessibility_profile_concordance, [7](#)
- calc_accessibility_sparsity_curve, [8](#)
- calc_all_bivariate_metrics, [9](#)
- calc_all_univariate_metrics, [9](#)
- calc_ami, [10](#)
- calc_ari, [11](#)
- calc_atac_rna_coupling, [12](#)
- calc_between_dataset_silhouette, [12](#)
- calc_bhattacharyya, [13](#)
- calc_calinski_harabasz, [14](#)
- calc_cdi, [14](#)
- calc_clustering_accuracy, [16](#)
- calc_cms, [16](#)
- calc_coexpression_module_fidelity, [18](#)
- calc_completeness, [19](#)
- calc_connectivity, [19](#)
- calc_coregulation_fidelity, [20](#)
- calc_cross_modal_generation, [21](#)
- calc_cross_modality_correlation, [21](#)
- calc_davies_bouldin, [22](#)
- calc_deconvolution_accuracy, [23](#)
- calc_delta_variance, [24](#)
- calc_dunn, [25](#)
- calc_ecdf_diffarea, [25](#)
- calc_emd_2d, [26](#)
- calc_excess_zero_weights, [27](#)
- calc_fasano_franceschini, [27](#)
- calc_fmi, [28](#)
- calc_foscttm, [29](#)

- calc_frechet_singlecell_distance, [30](#)
- calc_generative_precision_recall, [31](#)
- calc_homogeneity, [32](#)
- calc_homogeneity_completeness_v_measure, [32](#)
- calc_intraclass_correlation, [33](#)
- calc_isi, [34](#)
- calc_kde_test, [35](#)
- calc_kinetic_noise_decomposition, [35](#)
- calc_ks, [36](#)
- calc_ldf, [36](#)
- calc_ldf_diff, [37](#)
- calc_likelihood_ratio_test, [38](#)
- calc_local_structure_metric, [39](#)
- calc_mad, [40](#)
- calc_mae, [40](#)
- calc_marginal_aic_bic, [41](#)
- calc_mmd, [42](#)
- calc_modality_alignment, [43](#)
- calc_model_aic_bic, [44](#)
- calc_neighborhood_purity, [44](#)
- calc_network_jaccard, [45](#)
- calc_nmi, [46](#)
- calc_nn_mismatch, [46](#)
- calc_outlier_proportion, [47](#)
- calc_overlap, [48](#)
- calc_peacock_2d, [48](#)
- calc_peak_coaccessibility_fidelity, [49](#)
- calc_pseudotime_correlation, [50](#)
- calc_rmse, [50](#)
- calc_runs_test, [51](#)
- calc_seurat_mixing_metric, [52](#)
- calc_signal_bd, [53](#)
- calc_signal_dd, [53](#)
- calc_signal_de, [54](#)
- calc_signal_dp, [54](#)
- calc_signal_dv, [55](#)
- calc_silhouette, [56](#)
- calc_tree_height_discrepancy, [56](#)

calc_v_measure, 57
calc_variance_decomposition, 58
calc_wasserstein_1d, 59
calc_zero_probability_curve, 59
compute_dataset_embeddings, 60, 61, 93
compute_embedding_quality_metrics, 61
compute_method_leaderboard, 62

evaluate_accessibility_sparsity_curve, 63
evaluate_batch_metrics, 63
evaluate_clustering_metrics, 65
evaluate_cross_batch_prediction, 66
evaluate_cross_modal_prediction, 67
evaluate_deg_fidelity, 67
evaluate_epigenomic_annotation, 69
evaluate_multiomics_accuracy, 70
evaluate_multiple_datasets, 72, 90
evaluate_predictive_de_model, 73
evaluate_simbench_signals, 74
evaluate_simulation_accuracy, 62, 75, 90, 92, 96, 99
evaluate_trajectory_metrics, 77
evaluate_zero_probability_curve, 77
example_batch_info, 78
example_cell_types, 79
example_multiomics, 79
example_scatac, 78, 79, 80
example_scrna, 78, 79, 81
extract_cell_properties, 82
extract_dataset_summary, 82
extract_feature_properties, 83

fast_col_vars, 84
fast_cross_distance_sq, 84
fast_pca, 85
fast_row_vars, 85
fast_sparse_zero_fraction_cell, 86
fast_sparse_zero_fraction_feature, 86

infer_scrna_lineage_tree, 87
infer_scrna_pseudotime, 87

launch_scSimEval_app, 88

plot_benchmark_bubble_matrix, 89
plot_benchmark_summaryBars
 (plot_evaluation_summary), 95

plot_bubble_matrix
 (plot_benchmark_bubble_matrix), 89
plot_category_metric_bars, 91
plot_dataset_embeddings, 92
plot_distribution_qc, 94
plot_evaluation_summary, 95
plot_individual_metric_bar, 97
plot_metric_boxplots, 98
plot_metric_heatmap, 99
plot_metric_mds, 101
plot_metric_pca, 102
plot_scalability_benchmark, 104
plot_summary_bars
 (plot_evaluation_summary), 95

subsample_dist_vector, 105