Package: scSimEval (via r-universe)

November 27, 2026

Type Package

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

Version 0.99.18

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 6 multiomics integration and cross-modal alignment suite (joint-embedding Batch ASW, Cell-Type ASW, Graph Connectivity, Integration ARI, and Integration NMI); (ii) unpaired multiomics (independent scRNA-seq and scATAC-seq profiled from separate cells of the same tissue), supported through per-modality unimodal evaluation (Categories 1-5, 7, 8) and joint-embedding integration fidelity; 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 evaluating the paired and unimodal cell fractions across all 8 canonical categories. The toolkit quantifies simulation realism using 56 evaluation measures organized into 8 canonical categories: (1) Distributional Properties: 10 univariate metrics (Wasserstein, KS, MAD, MAE, RMSE, Bhattacharyya, KDE-ISE, Runs statistic, NN mismatch, Inter-dataset silhouette) and 4 bivariate metrics (Fasano-Franceschini 2D KS, 2D EMD, 2D NN mismatch, 2D Inter-dataset silhouette) evaluated across 7 cell-level, 7 feature-level, and 4 bivariate property pairs, (2) Sparsity & Zero-Inflation Dynamics: logistic dropout curve midpoint/slope/R2 (scRNA-seq), quadratic accessibility-sparsity curve RMSE/R2 (scATAC-seq), and posterior excess zero weight discrepancy, (3) Cellular Manifold & Clustering Concordance: MMD, Generative Precision/Recall, Silhouette ASW, Clustering Deviation Index (CDI), Neighborhood Purity, ARI, and NMI (with

auxiliary Davies-Bouldin Index), (4) Technical Variation & Batch Mixing Dynamics: Batch Silhouette, PCR R2, CMS, Inverse Simpson Index (ISI), and Cross-Batch F1, (5) Biological Signal & Differential Fidelity: pseudobulk edgeR-QLF null P-value uniformity, accuracy, precision, recall, F1-score, and ROC-AUC across parallel scRNA-seq DE and scATAC-seq DA sub-categories, (6) Multiomics Integration & Alignment: joint-embedding Batch ASW, Cell-Type ASW, Graph Connectivity, Integration ARI, and Integration NMI, (7) Trajectory & Lineage Topology: HVG-filtered dynverse benchmark (Saelens et al., 2019) comprising HIM topology similarity, edge flip similarity, geodesic distance correlation, branch assignment F1, and Random Forest feature importance weighted correlation (wcor_features), (8) Computational Scalability & Efficiency: core-normalized execution runtime (core_seconds) and memory footprint per allocated core (memory_per_core_mib), with automated multi-core scaling and unified memory_mb resource benchmarking. 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, and publication visualizations. Interactive GUI exploration is available via the companion \code{scSimEvalStudio} package.

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, Alignment, Pathways, Network, Software, Clustering, GraphAndNetwork

Encoding UTF-8

LazyData false

Depends R (>= 4.3.0)

VignetteBuilder knitr

Imports methods, stats, utils, grDevices, grid, RColorBrewer, Matrix, class, ggplot2, SingleCellExperiment, SummarizedExperiment, matrixStats, BiocParallel, CDI, RANN

Suggests testthat (>= 3.0.0), knitr, rmarkdown, cowplot, ggrepel, patchwork, fasano.franceschini.test, ks, emdist, MASS, edgeR, clue, mclust, cluster, CellMixS, kSamples, limma, bluster, e1071, irlba, Rtsne, uwot, RcppParallel, RhpcBLASctl, data.table, igraph, ranger, scater, scuttle

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libglpk-dev make libicu-dev libpng-dev libxml2-dev libssl-dev zlib1g-dev

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

Date/Publication 2026-11-24 19:56:27 UTC

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

RemoteRef HEAD

RemoteSha 008664deeb28bee94ee776c78eedd507a3a3d6a8

Contents

benchmark_resource_usage . . . 4
benchmark_simulation_suite . . . 5
combine_benchmark_results . . . 6
compute_core_seconds . . . 8
compute_dataset_embeddings . . . 9
compute_embedding_quality_metrics . . . 10
compute_memory_per_core . . . 12
compute_method_leaderboard . . . 13
evaluate_accessibility_sparsity_curve . . . 14
evaluate_atac_da_fidelity . . . 15
evaluate_batch_metrics . . . 17
evaluate_clustering_metrics . . . 18
evaluate_deg_fidelity . . . 19
evaluate_multiomics_accuracy . . . 22
evaluate_multiomics_integration . . . 24
evaluate_multiple_datasets . . . 26
evaluate_simulation_accuracy . . . 27
evaluate_trajectory_metrics . . . 29
evaluate_zero_probability_curve . . . 31
example_multiomics . . . 32
example_rice_bud . . . 33
example_scatac . . . 34
example_scrna . . . 35
example_soybean_root . . . 36
normalize_bubble_scores . . . 37
plot_benchmark_bubble_matrix . . . 38
plot_category_metric_bars . . . 40
plot_dataset_embeddings . . . 42
plot_distribution_qc . . . 43
plot_evaluation_summary . . . 44
plot_individual_metric_bar . . . 47
plot_metric_boxplots . . . 48
plot_metric_heatmap . . . 49
plot_metric_mds . . . 51
plot_metric_pca . . . 53
plot_scalability_benchmark . . . 55

Index **58**

benchmark_resource_usage *Benchmark Execution Runtime and Memory Usage*

Description

Times an expression and measures change in memory allocation, computing both raw runtime/memory metrics and core-normalized Category VIII scalability indicators (`core_seconds` and `memory_per_core_mib`).

Usage

```
benchmark_resource_usage(expr, sim_cores = NULL)
```

Arguments

`expr`	Expression or function to benchmark.
`sim_cores`	Optional positive integer specifying CPU cores used during execution (default `1L`).

Value

A named list containing the evaluated expression result and resource metrics:

`result` Return value of `expr` (or `NULL` if evaluation errored).

`core_seconds` Total CPU core-seconds (`elapsed_seconds * cores`).

`core_hours` Total CPU core-hours (`core_seconds / 3600`).

`memory_per_core_mib` Peak R heap memory per allocated core in MiB.

`memory_per_core_gib` Peak R heap memory per allocated core in GiB.

`memory_per_core_mb` Peak R heap memory per allocated core in decimal MB.

`memory_per_core_gb` Peak R heap memory per allocated core in decimal GB.

`elapsed_seconds` Wall-clock elapsed runtime in seconds.

`elapsed_minutes` Wall-clock elapsed runtime in minutes.

`memory_mb` Peak R heap memory allocated in MiB.

`memory_mib` Alias of `memory_mb` in MiB.

`memory_mb_si` Peak R heap memory allocated in decimal MB.

`sim_cores` Normalized integer core count (included only when `sim_cores` is supplied).

References

Lannelongue, L., et al. (2021). Green Algorithms. *Advanced Science*, 8(12), 2100707. Holtgrewe, M., et al. (2015). *Bioinformatics*, 31(12), 1904–1912.

Examples

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

benchmark_simulation_suite
Benchmark Simulation Methods Across Multiple Single-Cell Datasets

Description

Unified benchmarking runner for multi-dataset and multi-simulator studies. Manages sparse matrix coercion, `SingleCellExperiment` conversion, parallel metric computation, and produces both granular and cross-dataset aggregated summary tables ready for visualization.

Usage

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

Arguments

`datasets`	Named list of empirical reference datasets (`SingleCellExperiment`, count matrix, file path, or list containing `ref`, `cell_types`, `batch_info`).
`simulators`	Named list of simulator outputs corresponding to `datasets`.
`compute_bivariate`	Logical; compute 2D bivariate tests. Default `FALSE`.
`threads`	Integer number of parallel workers. Default `1L`.
`auto_sparse`	Logical; coerce dense matrices to `dgCMatrix`. Default `TRUE`.
`run_gc`	Logical; run `gc()` between datasets. Default `TRUE`.
`save_path`	Optional `.rds` path to save the output bundle.
`verbose`	Logical; print progress logging. Default `TRUE`.

Value

An S3 object of class `c("scSimEval_multidataset_benchmark", "scSimEval_consolidated", "list")` containing:

granular_benchmark_table Tidy data.frame across all dataset-simulator evaluations.

benchmark_summary_table Cross-dataset averaged fidelity scores formatted for `plot_benchmark_bubble_matrix`.

cross_dataset_stats Mean, SD, SE, Min, Max, and `N_Datasets` for each simulator and metric.

dataset_summary Summary table of reference dimensions, sparsity, and cell-type counts.

dataset_summary_df Alias of `dataset_summary`.

methods Character vector of evaluated simulator names.

datasets Character vector of evaluated dataset names.

is_multidataset Logical `TRUE`.

n_datasets Integer count of datasets.

n_simulators Integer count of simulators.

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)
head(res$benchmark_summary_table, 8)
```

`combine_benchmark_results` *Combine and Append Multiple Single-Cell Benchmark Results*

Description

Harmonizes, standardizes, and consolidates multiple single-cell simulator evaluation datasets or saved benchmark sessions into a unified multi-dataset benchmark object.

Usage

```
combine_benchmark_results(
  ...,
  list_of_benchmarks = NULL,
  dataset_names = NULL,
  filter_gold_standard = TRUE,
  recompute_normalized = TRUE
)

append_benchmark_results(
  ...,
  list_of_benchmarks = NULL,
  dataset_names = NULL,
  filter_gold_standard = TRUE,
  recompute_normalized = TRUE
)
```

Arguments

`...`	One or more benchmark evaluation objects, tidy data frames, or file paths (ending in `.rds` or `.csv`).
`list_of_benchmarks`	Optional list of benchmark objects or data frames.
`dataset_names`	Optional character vector specifying custom names for each dataset being combined. If `NULL`, dataset names are automatically extracted from the input objects or generated cleanly (e.g., `"Dataset 1"`, `"Dataset 2"`).
`filter_gold_standard`	Logical. If `TRUE` (default), filters all combined datasets to the 48 curated evaluation metrics.
`recompute_normalized`	Logical. If `TRUE` (default), re-normalizes raw metric scores across all evaluated simulators and datasets using direction-aware scaling.

Details

When the same simulator (e.g., `"scDesign3"` or `"scMultiSim"`) is evaluated across multiple empirical datasets, `combine_benchmark_results` maintains each distinct dataset evaluation and provides cross-dataset aggregate fidelity scores across the 48 curated evaluation metrics. This enables rigorous comparative evaluation and bubble matrix visualization of average simulator performance across diverse biological contexts.

Value

An S3 object of class `c("scSimEval_multidataset_benchmark", "scSimEval_consolidated", "list")` containing:

- `benchmark_summary_table`: Tidy data frame of cross-dataset average fidelity scores per simulator and metric (ready for `plot_benchmark_bubble_matrix`).
- `granular_benchmark_table`: Complete combined data frame preserving individual dataset rows.
- `dataset_summary_table`: Summary table of all combined datasets and simulator counts.
- `dataset_summary_df`: Alias of `dataset_summary_table`.
- `cross_dataset_stats`: Data frame of mean, SD, SE, and sample count per simulator-metric pair.
- `methods`: Unique simulator names evaluated across all datasets.
- `datasets`: Vector of distinct dataset names.
- `is_multidataset`: Logical indicating whether more than one dataset is included.
- `n_datasets`: Integer number of datasets.
- `n_simulators`: Integer number of unique simulators.

Examples

```
res1 <- list(
  metrics_summary_table = data.frame(
    Category = "Distributional Properties",
    Property = "library_size",
    Metric = "Wasserstein",
    Value = 0.22
  )
)

res2 <- list(
  metrics_summary_table = data.frame(
    Category = "Distributional Properties",
    Property = "library_size",
    Metric = "Wasserstein",
    Value = 0.18
  )
)

comb <- combine_benchmark_results(
  res1, res2,
  dataset_names = c("Cohort A", "Cohort B"),
  filter_gold_standard = FALSE,
  recompute_normalized = FALSE
)

print(comb$datasets)
```

---

compute_core_seconds *Compute Core-Seconds Runtime Metric*

Description

Quantifies cumulative CPU work across all allocated processing cores (Elapsed Time (s) $\times$ N_{cores}; Lannelongue et al., 2021; Wratten et al., 2021; Grealey et al., 2022).

Usage

```
compute_core_seconds(elapsed_time, time_unit = "seconds", sim_cores = 1L)
```

Arguments

`elapsed_time`	Numeric elapsed wall-clock runtime.
`time_unit`	Time unit of `elapsed_time`: `"seconds"` (default) or `"minutes"`.
`sim_cores`	Positive integer specifying the number of CPU cores allocated (default `1L`).

Value

Numeric Core-Seconds (core · s).

References

Lannelongue, L., Grealey, J., & Inouye, M. (2021). Green Algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12), 2100707.

Wratten, L., Wilm, A., & Göke, J. (2021). Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nature Methods*, 18, 1161–1168.

Grealey, J., et al. (2022). The Carbon Footprint of Bioinformatics. *Molecular Biology and Evolution*, 39(3), msac034.

Examples

```
# Single-core run taking 45 seconds
compute_core_seconds(elapsed_time = 45, time_unit = "seconds", sim_cores = 1L)

# 8-core parallel run taking 2.5 minutes
compute_core_seconds(elapsed_time = 2.5, time_unit = "minutes", sim_cores = 8L)
```

---

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 SingleCellExperiment object or count matrix (features x cells).
simulated	A single simulated count matrix/SingleCellExperiment or a named list of them.
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` (with `check.names = FALSE`) containing one row per dataset with columns:

`Dataset` Dataset label (e.g., `"Reference"` or simulator name).

`Role` Dataset role (`"Reference"` or `"Simulated"`).

`Cells (N)` Number of cells in the embedding.

`Mean Silhouette` Mean silhouette width across cell-type labels rounded to 4 decimals (or `NA_real_`).

`ARI (Cluster Fidelity)` Adjusted Rand Index between clustering and cell-type labels rounded to 4 decimals (or `NA_real_`).

`Mean Library Size` Mean total counts per cell rounded to 1 decimal.

`Mean Detected Features` Mean detected features per cell rounded to 1 decimal.

`Library Size Diff (%)` Relative percent difference in mean library size vs. Reference (present when a `"Reference"` row exists).

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_memory_per_core` *Compute Peak RAM per Allocated CPU Core*

Description

Normalizes peak memory consumption by the number of allocated processor cores (Peak RAM$/N_{\text{cores}}$; Holtgrewe et al., 2015; Boisvert et al., 2012; Chen et al., 2022).

Usage

```
compute_memory_per_core(
  memory_val,
  memory_unit = "MiB",
  sim_cores = 1L,
  target_unit = c("MiB", "GiB", "MB", "GB")
)
```

Arguments

`memory_val`	Numeric peak memory consumption.
`memory_unit`	Memory unit of `memory_val`: `"MiB"` (default) or `"MB"`.
`sim_cores`	Positive integer specifying the number of CPU cores allocated (default `1L`).
`target_unit`	Target memory unit: `"MiB"` (MiB/core, default), `"GiB"` (GiB/core), `"MB"` (MB/core), or `"GB"` (GB/core).

Value

Numeric Peak RAM per allocated core in requested target unit.

References

Holtgrewe, M., Kuchenbecker, L., & Reinert, K. (2015). Methods for the detection and assembly of novel sequence in high-throughput sequencing data. *Bioinformatics*, 31(12), 1904–1912.

Boisvert, S., Raymond, F., Godzaridis, E., Laviolette, F., & Corbeil, J. (2012). Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biology*, 13(12), R122.

Chen, W., et al. (2022). STEPS 4.0: Fast and memory-efficient molecular simulations of neurons at the nanoscale. *Frontiers in Neuroinformatics*, 16, 883742.

Examples

```
# 4096 MiB peak RAM across 4 CPU cores (reported in MiB/core and GiB/core)
compute_memory_per_core(4096, memory_unit = "MiB", sim_cores = 4L, target_unit = "MiB")
compute_memory_per_core(4096, memory_unit = "MiB", sim_cores = 4L, target_unit = "GiB")
```

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

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim"), each = 4),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics"
  ), 4),
  Metric = rep(c("KS", "Wasserstein", "ARI", "CMS"), 4),
  Value = c(
    0.10, 0.16, 0.85, 0.88,
    0.14, 0.21, 0.79, 0.83,
    0.23, 0.34, 0.67, 0.72,
    0.31, 0.44, 0.56, 0.62
  ),
  stringsAsFactors = FALSE
)
```

```
lb <- compute_method_leaderboard(bm_df)
print(lb)
```

`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

`ref_data`	Reference count matrix (peaks x cells).
`sim_data`	Simulated count matrix (peaks x cells).
`poly_degree`	Degree of polynomial (default: 2).

Value

A named list quantifying concordance between reference and simulated accessibility-sparsity curves:

`curve_rmse` Root mean squared error between predicted NZP curves over a 100-point grid spanning the reference mean range.

`r2_discrepancy` Absolute difference between simulated and reference polynomial R^2 values.

`ref_r_squared` Polynomial R^2 of the reference dataset.

`sim_r_squared` Polynomial R^2 of the simulated dataset.

Examples

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

`evaluate_atac_da_fidelity` *Evaluate scATAC-seq Differential Accessibility (DA) Fidelity (Subcategory V-B)*

Description

Evaluates the preservation of Differentially Accessible Regions (DARs) between reference and simulated scATAC-seq datasets using biological replicate pseudobulks and edgeR-QLF (Squair et al., Nature Communications 2021; Teo et al., Nature Communications 2024 / Libra) across standard benchmarking metrics (Duo et al., Genome Biology 2024 / Simpipe).

Usage

```
evaluate_atac_da_fidelity(
  ref_data,
  sim_data,
  ref_celltypes,
  sim_celltypes,
  batch_ref = NULL,
  batch_sim = NULL,
  binarize = TRUE,
  fdr_cutoff = 0.05,
  logfc_cutoff = 0,
  min_cells = 3,
  min_reps = 2,
  threads = 1
)
```

Arguments

`ref_data`	Reference scATAC-seq count matrix (peaks x cells).
`sim_data`	Simulated scATAC-seq count matrix (peaks x cells).
`ref_celltypes`	Factor or character vector of reference cell types.
`sim_celltypes`	Factor or character vector of simulated cell types.
`batch_ref`	Factor or character vector of biological replicate / batch labels for reference cells.
`batch_sim`	Factor or character vector of biological replicate / batch labels for simulated cells.
`binarize`	Logical. If TRUE (default for scATAC-seq), binarizes single-cell counts prior to pseudobulk aggregation.
`fdr_cutoff`	Adjusted p-value significance threshold (default 0.05).
`logfc_cutoff`	Absolute log2 fold-change cutoff (default 0).
`min_cells`	Minimum number of cells per cell type (default 3).
`min_reps`	Minimum number of replicates per group (default 2).
`threads`	Number of CPU threads for parallel pairwise contrast evaluation (default 1).

Details

Category V is divided into two distinct sub-categories:

- **Sub-category V-A (scRNA-seq Differential Expression)**: Computes 6 metrics: `pval_dist_score`, `accuracy`, `precision`, `recall`, `f1_score`, and `auc`.
- **Sub-category V-B (scATAC-seq Differential Accessibility)**: Computes 6 metrics: `pval_dist_score`, `accuracy`, `precision`, `recall`, `f1_score`, and `auc`.

If `batch_ref` or `batch_sim` is NULL, the analysis is gracefully skipped, adhering to the strict rule that single-cell ATAC-seq biological signal evaluation requires genuine biological replicates to prevent pseudoreplication bias.

Value

A list containing:

`da_summary_table`
: Tidy summary data.frame with 6 scATAC-seq DA metrics: `pval_dist_score`, `accuracy`, `precision`, `recall`, `f1_score`, and `auc`

`pairwise_results`
: Detailed contrast-level data.frame with per-contrast metrics and confusion matrix counts

`pseudobulks_ref`
: List of reference pseudobulk matrices per cell type

`pseudobulks_sim`
: List of simulated pseudobulk matrices per cell type

`da_results_ref` List of edgeR-QLF DA results for reference

`da_results_sim` List of edgeR-QLF DA results for simulation

Examples

```
data(example_scatac, package = "scSimEval")
rep_id <- factor(rep(c("Rep1", "Rep2"), length.out = ncol(example_scatac$ref)))
res <- evaluate_atac_da_fidelity(
  ref_data = example_scatac$ref,
  sim_data = example_scatac$sim,
  ref_celltypes = example_scatac$cell_types,
  sim_celltypes = example_scatac$cell_types,
  batch_ref = rep_id,
  batch_sim = rep_id
)
res$da_summary_table
```

evaluate_batch_metrics *Calculate Batch Integration Metrics*

Description

Evaluates how realistically simulated technical batches mix or separate, computing the five canonical Category IV (Technical Variation & Batch Mixing Dynamics) metrics: Batch Silhouette, PC Regression R2, CMS (Cell-Specific Mixing Score), Inverse Simpson Index (ISI), and cross-batch cell type classification macro F1.

Usage

```
evaluate_batch_metrics(
  data,
  batch_info,
  k = NULL,
  n_pcs = 30,
  cell_types = NULL,
  threads = 1,
  verbose = FALSE
)
```

Arguments

data	Count or log-normalized expression matrix (features x cells).
batch_info	Factor or character vector of batch assignments for each cell.
k	Neighborhood size k for k-NN graph and mixing metrics. Default is min(table(batch_info))/2.
n_pcs	Number of principal components for embedding evaluation. Default is 30.
cell_types	Optional factor or character vector of cell types for cross-batch evaluation.
threads	Number of CPU threads (default 1).
verbose	Logical, whether to print progress.

Value

A named list of batch metrics:

batch_silhouette	Average silhouette width using batch labels (lower is better mixed)
pcr_r2	Total variance explained by batch in principal components (PCR)
cms	Mean Cell-specific mixing score (CellMixS)
isi	Mean Inverse Simpson Index for batch mixing (CellMixS)
cross_batch_F1	Mean cross-batch cell type transfer macro F1 (if cell_types 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, DB, Neighborhood Purity, CDI) and supervised concordance against ground truth labels (ARI, NMI).

Usage

```
evaluate_clustering_metrics(
  data,
  cluster_info = NULL,
  pred_clusters = NULL,
  dist_mat = NULL,
  cell_types = NULL,
  ref_data = NULL,
  ref_cell_types = NULL,
  batch_info = NULL,
  threads = 1L
)
```

Arguments

`data`	Count or normalized matrix (features x cells).
`cluster_info`	Cluster or cell type labels.
`pred_clusters`	Optional predicted cluster labels (if different from ground truth). If NULL, k-means is automatically run.
`dist_mat`	Optional precomputed distance matrix.
`cell_types`	Optional alias for cluster_info.
`ref_data`	Optional reference data to compute reference clustering quality.
`ref_cell_types`	Optional reference cell type labels (if ref_data has different cells than data).
`batch_info`	Optional batch labels per cell passed to `calc_cdi`.
`threads`	Integer number of parallel CPU threads passed to `calc_cdi` (default `1L`).

Value

A named list of clustering quality and cluster concordance metrics (keys present depend on inputs):

`silhouette_sim` Mean silhouette width across ground-truth labels in simulated data.

`silhouette` Alias of `silhouette_sim`.

`davies_bouldin` Davies-Bouldin index computed on simulated data.

`davies_bouldin_sim` Alias of `davies_bouldin`.

`neighborhood_purity` Mean k-nearest-neighbor label purity in simulated data.

`silhouette_ref` Mean silhouette width in reference data (present when `ref_data` is supplied).

`davies_bouldin_ref` Davies-Bouldin index in reference data (present when `ref_data` is supplied).

`cdi` Normalized CDI-AIC score from `calc_cdi` (when finite).

`CDI_AIC` Alias of `cdi`.

`CDI_BIC` Normalized CDI-BIC score from `calc_cdi` (when finite).

`ari` Adjusted Rand Index between `pred_clusters` (or k-means clusters) and `cluster_info`.

`ARI` Alias of `ari`.

`nmi` Normalized Mutual Information between `pred_clusters` and `cluster_info`.

`NMI` Alias of `nmi`.

Examples

```
data <- matrix(stats::rpois(200, lambda = 3), 20, 10)
cl <- factor(rep(c("A", "B"), each = 5))
dist_mat <- stats::dist(t(data))
evaluate_clustering_metrics(data, cl)
```

`evaluate_deg_fidelity` *Evaluate scRNA-seq Differential Expression (DE) Fidelity (Subcategory V-A)*

Description

Evaluates the preservation of Differentially Expressed Genes (DEGs) between reference and simulated scRNA-seq datasets using biological replicate pseudobulks and edgeR-QLF (Squair et al., Nature Communications 2021 / Libra) across standard benchmarking metrics (Duo et al., Genome Biology 2024 / Simpipe).

Usage

```
evaluate_deg_fidelity(
  ref_data,
  sim_data,
  ref_celltypes,
  sim_celltypes,
  batch_ref = NULL,
  batch_sim = NULL,
  fdr_cutoff = 0.05,
  logfc_cutoff = 0,
  classifier = c("svm", "knn", "rf"),
  min_cells = 3,
  min_reps = 2,
  threads = 1
)

evaluate_rna_de_fidelity(
  ref_data,
  sim_data,
  ref_celltypes,
  sim_celltypes,
  batch_ref = NULL,
  batch_sim = NULL,
  fdr_cutoff = 0.05,
  logfc_cutoff = 0,
  classifier = c("svm", "knn", "rf"),
  min_cells = 3,
  min_reps = 2,
  threads = 1
)
```

Arguments

ref_data	Reference scRNA-seq count matrix (genes x cells).
sim_data	Simulated scRNA-seq count matrix (genes x cells).
ref_celltypes	Factor or character vector of reference cell types.
sim_celltypes	Factor or character vector of simulated cell types.
batch_ref	Factor or character vector of biological replicate / batch labels for reference cells. Required.
batch_sim	Factor or character vector of biological replicate / batch labels for simulated cells. Required.
fdr_cutoff	Adjusted p-value significance threshold for DEGs (default 0.05).
logfc_cutoff	Absolute log2 fold-change cutoff (default 0).
classifier	Classifier algorithm for marker gene predictability: "svm" (default), "knn", or "rf".
min_cells	Minimum number of cells per cell type (default 3).

min_reps	Minimum number of replicates per group (default 2).
threads	Number of CPU threads for parallel pairwise contrast evaluation (default 1).

Details

Category V is divided into two distinct sub-categories:

- **Sub-category V-A (scRNA-seq Differential Expression)**: Computes 6 metrics: pval_dist_score, accuracy, precision, recall, f1_score, and auc.
- **Sub-category V-B (scATAC-seq Differential Accessibility)**: Computes 6 metrics: pval_dist_score, accuracy, precision, recall, f1_score, and auc.

If batch_ref or batch_sim is NULL, the analysis is gracefully skipped, adhering to the strict rule that single-cell biological signal evaluation requires genuine biological replicates to prevent false positive inflation and pseudoreplication bias.

Value

A list containing:

deg_summary_table
: Tidy summary data.frame with 6 scRNA-seq DE metrics: pval_dist_score, accuracy, precision, recall, f1_score, and auc

pairwise_results
: Detailed contrast-level data.frame with per-contrast metrics and confusion matrix counts

pseudobulks_ref
: List of reference pseudobulk matrices per cell type

pseudobulks_sim
: List of simulated pseudobulk matrices per cell type

de_results_ref
: List of edgeR-QLF DE results for reference

de_results_sim
: List of edgeR-QLF DE results for simulation

ml_classification
: Detailed classification results using simulated pseudobulk DEGs

Examples

```
data(example_scrna, package = "scSimEval")
rep_id <- factor(rep(c("Rep1", "Rep2"), length.out = ncol(example_scrna$ref)))
res <- evaluate_deg_fidelity(
  ref_data = example_scrna$ref,
  sim_data = example_scrna$sim,
  ref_celltypes = example_scrna$cell_types,
  sim_celltypes = example_scrna$cell_types,
  batch_ref = rep_id,
  batch_sim = rep_id
)
res$deg_summary_table
```

`evaluate_multiomics_accuracy` *Master Multiomics Simulation Benchmarking Pipeline*

Description

Orchestrates comprehensive benchmarking across all 8 evaluation categories using `SingleCellExperiment` inputs (either a multi-modal SCE containing an `altExp` for chromatin accessibility, or a named list of two SCE objects such as `list(RNA = rna_sce, ATAC = atac_sce)`). Cell-type and batch labels are read directly from `colData` via `cell_type_col` and `batch_col`. Users starting from `Seurat` objects should convert via `Seurat::as.SingleCellExperiment()`.

Usage

```
evaluate_multiomics_accuracy(
  ref_multi,
  sim_multi,
  cell_type_col = "cell_type",
  batch_col = "batch",
  pairing = c("paired", "unpaired"),
  memory_mb = NULL,
  elapsed_time = NULL,
  time_unit = c("seconds", "minutes"),
  memory_unit = c("MiB", "MB"),
  sim_cores = NULL,
  separate_resources = NULL,
  feature_pairs = NULL,
  compute_bivariate = TRUE,
  threads = 1L,
  trajectory_num_trees = 10000L,
  verbose = TRUE
)
```

Arguments

`ref_multi`	Reference multiomics `SingleCellExperiment` (with `altExp`) or named list of two SCE objects.
`sim_multi`	Simulated multiomics `SingleCellExperiment` (with `altExp`) or named list of two SCE objects.
`cell_type_col`	Column name in `colData` containing cell-type labels (default `"cell_type"`).
`batch_col`	Column name in `colData` containing batch/replicate labels (default `"batch"`).
`pairing`	Character string specifying `"paired"` (default) or `"unpaired"` assay structure.
`memory_mb`	Optional numeric peak memory allocation (in `memory_unit`).
`elapsed_time`	Optional numeric wall-clock runtime (in `time_unit`).
`time_unit`	Unit of `elapsed_time`: `"seconds"` (default) or `"minutes"`.

`memory_unit` Unit of `memory_mb`: `"MiB"` (default) or `"MB"`.

`sim_cores` Optional positive integer specifying CPU cores used by the simulator.

`separate_resources`
Optional named list (e.g., `list(rna = list(...), atac = list(...))`) giving per-modality `elapsed_time`, `memory_mb`, `time_unit`, `memory_unit`, and `sim_cores` when RNA and ATAC layers are simulated separately.

`feature_pairs` Optional 2-column data.frame of linked feature pairs.

`compute_bivariate`
Logical, whether to compute 2D bivariate tests. Default `TRUE`.

`threads` Integer number of CPU threads. Default is `1L`.

`trajectory_num_trees`
Number of Random Forest trees for Category VII `wcor_features` (default `10000L`).

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

Details

Multiomics Metric Measurement Capacity (56 Evaluation Measures Suite):

- **Paired Multiomics (56 Metrics):** Dual unimodal properties (Cat I–II), dual clustering (Cat III), dual batch mixing (Cat IV), dual biological signal (Cat V: 6 DE + 6 DA), cross-modal integration & alignment (Cat VI), scRNA trajectory (Cat VII), and scalability (Cat VIII).
- **Unpaired Multiomics (56 Metrics):** Evaluates both modalities independently across Cat I–V, VII, VIII plus population-level joint integration & alignment (Cat VI).

Value

A named list containing:

benchmark_summary_table Master tidy data frame with columns `Category`, `Property`, `Metric`, `Value`, `Modality` (and `Sim_Cores` if provided).

metrics_summary_table Alias of `benchmark_summary_table`.

summary_table Alias of `benchmark_summary_table`.

pairing Resolved pairing mode (`"paired"` or `"unpaired"`).

mod1_unimodal Full unimodal evaluation output for modality 1.

mod2_unimodal Full unimodal evaluation output for modality 2.

clustering List with elements `m1` and `m2` containing Category III results.

batch Named list of Category IV batch mixing results per modality.

signal Named list of Category V differential fidelity results per modality.

trajectory Category VII *dynverse* trajectory results inferred from the scRNA-seq layer.

cross_modality List storing Category VI integration output under `correlation`.

resource_usage Normalized Category VIII scalability list supporting both `"combined"` and `"separate"` modes.

Examples

```
data(example_multiomics, package = "scSimEval")
res <- evaluate_multiomics_accuracy(
  example_multiomics$ref_multi,
  example_multiomics$sim_multi,
  verbose = FALSE
)
head(res$benchmark_summary_table, 10)
```

`evaluate_multiomics_integration` *Evaluate Multiomics Integration & Cross-Modal Alignment Fidelity (Category VI)*

Description

Evaluates the multiomics integration and cross-modal alignment fidelity of simulated single-cell multiomics data across 5 gold-standard integration metrics extracted from key single-cell benchmarking suites (`scib`, `scMultiSim`, `scCross`, `Matilda`, and `scMoMtF`):

1. **Batch ASW (`batch_asw`):** Average Silhouette Width evaluating batch mixing within cell types (scib, scMultiSim).
2. **Cell-Type ASW (`cell_type_asw`):** Average Silhouette Width evaluating cell-type bio-conservation in the multiomics space (scCross, scib).
3. **Graph Connectivity (`graph_connectivity`):** Relative size of the largest connected component in the k-NN graph for each cell type (scCross, scMultiSim, scib).
4. **ARI (`ari`):** Adjusted Rand Index comparing joint multiomics clustering against ground-truth cell types (scib, scCross, scMultiSim, Matilda, scMoMtF).
5. **NMI (`nmi`):** Normalized Mutual Information quantifying information concordance between joint clustering and cell types (scib, scCross, scMultiSim, Matilda, scMoMtF).

Usage

```
evaluate_multiomics_integration(
  ref_multi = NULL,
  sim_multi,
  cell_types = NULL,
  batch_info = NULL,
  embed = NULL,
  reduction_name = NULL,
  n_pcs = 15,
  k = 30,
  threads = 1,
  pairing = NULL,
  verbose = TRUE
)
```

Arguments

ref_multi	Optional reference multiomics list of matrices (e.g. list(rna = ..., atac = ...)).
sim_multi	Simulated multiomics list of matrices or a matrix representing joint embedding coordinates.
cell_types	Optional cell-type vector or list/data.frame of annotations for each cell.
batch_info	Optional batch/donor/replicate vector or list/data.frame of annotations for each cell.
embed	Optional precomputed joint embedding matrix (cells x dimensions). If NULL, computed automatically from sim_multi.
reduction_name	Optional name of the precomputed dimensional reduction to extract from SingleCellExperiment objects.
n_pcs	Number of principal components per modality to retain for joint embedding (default 15).
k	Number of nearest neighbors for graph-based metrics (default 30).
threads	Number of CPU threads for parallel processing (default 1).
pairing	Optional pairing mode ("paired" or "unpaired"); auto-detected from cell barcodes if NULL.
verbose	Logical; whether to print progress messages (default TRUE).

Value

A list of class scSimEval_integration containing:

metrics	Named list of the 5 calculated integration metric values (batch_asw, cell_type_asw, graph_connectivity, ari, nmi).
integration_summary_table	A tidy data.frame containing the 5 metrics with columns Category, Property, Metric, Value, and Modality.
embedding	The joint multiomics embedding matrix used for evaluation.

Examples

```
data(example_multiomics, package = "scSimEval")
ct <- rep(c("A", "B"), each = 50)
res <- evaluate_multiomics_integration(
  sim_multi = example_multiomics$sim_multi,
  cell_types = ct
)
unlist(res[vapply(res, function(x) is.numeric(x) && length(x) == 1L, logical(1))])
```

`evaluate_multiple_datasets` *Evaluate Multiple Single-Cell Datasets Simultaneously*

Description

Orchestrates simultaneous benchmarking across a collection of single-cell `SingleCellExperiment` datasets, including paired/unpaired multiomics and unimodal scRNA-seq or scATAC-seq experiments.

Usage

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

Arguments

`datasets`	A named list of dataset specifications (`Format A` grouped multiomics/unimodal lists, or `Format B` flat modality-suffixed lists).
`pair_by_prefix`	Logical; if `TRUE` (default), datasets sharing a root prefix across RNA and ATAC entries are paired for multiomics evaluation.
`compute_bivariate`	Logical; whether to compute 2D bivariate tests (default `FALSE`).
`threads`	Integer number of CPU threads. Default is `1L`.
`verbose`	Logical; whether to print progress messages. Default is `TRUE`.

Value

An S3 object of class `c("scSimEval_consolidated", "list")` containing:

consolidated_summary_table Master tidy data.frame with columns `Dataset`, `Data_Name`, `Modality`, `Category`, `Property`, `Metric`, and `Value`.

consolidated_score_matrix Numeric matrix comparing datasets across all evaluated metric keys.

dataset_overview Data frame summarizing metric counts and mean KS, Wasserstein, and RMSE per dataset.

dataset_results Named list of detailed per-dataset evaluation outputs.

Examples

```
data(example_scrna, package = "scSimEval")
ds <- list("DemoCohort" = list(ref = example_scrna$ref, sim = example_scrna$sim))
res <- evaluate_multiple_datasets(ds, verbose = FALSE)
res$dataset_overview
```

`evaluate_simulation_accuracy` *Evaluate Comprehensive Simulation Accuracy (Unimodal SingleCellExperiment)*

Description

Computes distribution distances, cellular and feature-level properties, zero-inflation curves, clustering concordance, batch mixing, differential expression/accessibility fidelity, trajectory topology, and computational scalability between reference and simulated `SingleCellExperiment` objects. Users starting from `Seurat` objects should convert via `Seurat::as.SingleCellExperiment()`.

Usage

```
evaluate_simulation_accuracy(
  ref_sce,
  sim_sce,
  cell_type_col = "cell_type",
  batch_col = "batch",
  modality = NULL,
  compute_bivariate = TRUE,
  threads = 1L,
  memory_mb = NULL,
  elapsed_time = NULL,
  time_unit = c("seconds", "minutes"),
  memory_unit = c("MiB", "MB"),
  sim_cores = NULL,
  trajectory_num_trees = 10000L,
  include_trajectory = TRUE,
  include_downstream = TRUE,
  verbose = TRUE
)
```

Arguments

`ref_sce`	Reference `SingleCellExperiment` object.
`sim_sce`	Simulated `SingleCellExperiment` object.
`cell_type_col`	Column name in `colData` containing cell-type labels (default `"cell_type"`).
`batch_col`	Column name in `colData` containing batch/replicate labels (default `"batch"`).
`modality`	Optional modality hint (`"scRNA-seq"` or `"scATAC-seq"`); auto-detected if `NULL`.

`compute_bivariate`
: Logical, whether to compute 2D bivariate tests. Default is `TRUE`.

`threads`
: Integer number of parallel worker threads. Default is `1L`.

`memory_mb`
: Optional numeric peak memory allocation (in `memory_unit`).

`elapsed_time`
: Optional numeric wall-clock elapsed runtime (in `time_unit`).

`time_unit`
: Unit of `elapsed_time`: `"seconds"` (default) or `"minutes"`.

`memory_unit`
: Unit of `memory_mb`: `"MiB"` (default) or `"MB"`.

`sim_cores`
: Optional positive integer specifying CPU cores used by the simulator.

`trajectory_num_trees`
: Number of Random Forest trees for Category VII `wcor_features` (default `10000L`).

`include_trajectory`
: Logical; whether to evaluate Category VII trajectory metrics for scRNA-seq. Default `TRUE`.

`include_downstream`
: Logical; whether to evaluate Categories III, IV, and V downstream metrics. Default `TRUE`.

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

Details

Metric Measurement Capacity across 4 Analysis Options (56 Evaluation Measures Suite): Cell-type and batch annotations are retrieved directly from `colData(ref_sce)` and `colData(sim_sce)` using `cell_type_col` and `batch_col`.

- **1. scRNA-seq (44 Metrics):** Categories I, II, III, IV, V (DE), VII, and VIII.
- **2. scATAC-seq (38 Metrics):** Categories I, II, III, IV, V (DA), and VIII (trajectory skipped for ATAC).
- **3. Paired Multiomics (56 Metrics):** All 8 categories (I–VIII) via `evaluate_multiomics_accuracy`.
- **4. Unpaired Multiomics (56 Metrics):** All 8 categories (I–VIII) via `evaluate_multiomics_accuracy` on unpaired RNA and ATAC populations.

Value

A named list containing tidy summaries and detailed category results:

metrics_summary_table Data frame with columns `Category`, `Property`, `Metric`, `Value` (and `Sim_Cores` if provided).

cell_metrics Named list of per-property univariate metric lists at cell resolution.

feature_metrics Named list of per-property univariate metric lists at feature resolution.

bivariate_metrics Named list of 2D bivariate metric comparisons.

zero_probability_curve Category II scRNA-seq dropout curve results, or `NULL`.

accessibility_sparsity_curve Category II scATAC-seq sparsity curve results, or `NULL`.

excess_zero_weights List with `ref` and `sim` excess-zero probability weights.

generative_manifold List with `mmd` and `precision_recall` generative manifold scores.

ref_properties List containing `cell` and `feature` property vectors extracted from `ref_sce`.

sim_properties List containing `cell` and `feature` property vectors extracted from `sim_sce`.

batch_effects Category IV batch mixing metrics list, or `NULL`.

biological_signal Category V pseudobulk DE/DA fidelity results, or `NULL`.

trajectory Category VII *dynverse* trajectory evaluation results, or `NULL`.

resource_usage Normalized Category VIII scalability list (`core_seconds`, `memory_per_core_mib`, `memory_mb`, `elapsed_time`, `sim_cores`).

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_simulation_accuracy(
  example_scrna$ref, example_scrna$sim,
  include_trajectory = FALSE, verbose = FALSE
)
head(res$metrics_summary_table, 10)
```

---

evaluate_trajectory_metrics

Full Category VII Trajectory & Lineage Topology Evaluation (dynverse Workflow)

Description

Evaluates trajectory and lineage preservation using the standardized *dynverse* benchmarking framework (*Saelens et al., Nature Biotechnology, 2019*):

1. Standardized cell outlier QC (3 MADs), low-expression gene filtering ($\geq 5\%$ cells, mean > 0.02), and HVG selection (empirical mean-variance decomposition).
2. Shared HVG PCA projection and Docker-free Minimum Spanning Tree (MST) milestone trajectory inference with continuous cell-to-milestone geodesic distances.
3. Five complementary trajectory metrics: `him` (Hamming-Ipsen-Mikhailov similarity), `edge_flip` (minimum edge flip similarity), `geodesic_correlation`, `F1_branches`, and `wcor_features` (**ranger** Random Forest feature importance weighted correlation with `num_trees = 10000` and 1 percent `mtry`).

Usage

```
evaluate_trajectory_metrics(
  ref_data,
  sim_data,
  cell_types_ref = NULL,
  cell_types_sim = NULL,
  num_trees = 10000L,
  nmads = 3,
  min_cell_frac = 0.05,
```

```
  min_mean_expr = 0.02,
  fdr_threshold = 0.05,
  bio_var_threshold = 0.5,
  threads = 1L,
  seed = 42L,
  verbose = FALSE
)
```

Arguments

`ref_data`	Reference count matrix or `SingleCellExperiment` object.
`sim_data`	Simulated count matrix or `SingleCellExperiment` object.
`cell_types_ref`	Optional vector of reference cell types.
`cell_types_sim`	Optional vector of simulated cell types.
`num_trees`	Number of **ranger** trees per milestone for `wcor_features` (default `10000L`).
`nmads`	MAD cutoff for outlier cell QC (default `3`).
`min_cell_frac`	Minimum cell fraction per gene (default `0.05`).
`min_mean_expr`	Minimum average gene count (default `0.02`).
`fdr_threshold`	FDR threshold for HVG selection (default `0.05`).
`bio_var_threshold`	Biological variance threshold for HVG selection (default `0.5`).
`threads`	Number of CPU threads for **ranger** (default `1L`).
`seed`	Random seed (default `42L`).
`verbose`	Logical; print progress messages.

Value

A named list of Category VII *dynverse* trajectory accuracy metrics:

him Hamming-Ipsen-Mikhailov topology similarity (`[0, 1]`).

edge_flip Edge flip topology similarity (`[0, 1]`).

geodesic_correlation Spearman correlation of aligned geodesic distances (`[0, 1]`).

F1_branches Macro F1 overlap of MST branch assignments (`[0, 1]`).

wcor_features Reference-weighted Pearson correlation of Random Forest gene importances (`[0, 1]`).

cor_features Unweighted Pearson correlation of Random Forest gene importances.

n_hvgs Number of HVGs retained for trajectory evaluation.

ref_trajectory Inferred reference milestone trajectory object.

sim_trajectory Inferred simulated milestone trajectory object.

preprocessing_summary Cell and HVG filtering summary list.

Examples

```
data(example_scrna, package = "scSimEval")
res <- evaluate_trajectory_metrics(
  example_scrna$ref, example_scrna$sim,
  cell_types_ref = example_scrna$cell_types,
  cell_types_sim = example_scrna$cell_types,
  num_trees = 200L
)
unlist(res[c("him", "edge_flip", "geodesic_correlation",
             "F1_branches", "wcor_features", "n_hvgs")])
```

---

evaluate_zero_probability_curve

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

Description

Compares empirical logistic dropout trends between reference and simulated scRNA-seq 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 (`slope_error`, `midpoint_error`, `r2_discrepancy`).

Examples

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

`example_multiomics` *Example Paired Multiomics Benchmarking Dataset (Rice Bud)*

Description

A unified paired single-cell multiomics dataset combining `example_scrna` (100 HVGs) and `example_scatac` (100 HVPs) across the exact same 100 balanced rice bud cells, accompanied by shared cell-type labels, batch annotations, and empirical computational scalability metrics.

Usage

```
data(example_multiomics)
```

Format

A list with 5 elements:

ref_multi Named list of `SingleCellExperiment` objects: `rna` (100 HVGs x 100 cells) and `atac` (100 HVPs x 100 cells), complete with `colData` and precomputed `PCA` in `reducedDims`.

sim_multi Named list of simulated `SingleCellExperiment` objects: `rna` (100 x 100) and `atac` (100 x 100).

cell_types A named factor of shared cell-type identities across 8 Rice Bud cell types for the 100 paired cells.

batch_info A named factor of batch annotations (`Bud_21`, `Bud_22`) for the 100 paired cells.

resource_stats A list of empirical scalability metrics: `elapsed_time`, `memory_mb`, `sim_cores`.

Source

Derived from rice bud paired multiomics data (*Oryza sativa*, GEO accession GSE232863).

Examples

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

example_rice_bud *Real Paired Single-Cell Multiomics Benchmark Dataset (Rice Bud, SingleCellExperiment)*

Description

A real, pre-processed paired single-cell RNA-seq and ATAC-seq benchmark dataset of rice bud tissue (*Oryza sativa*, GEO accession GSE232863), standardized using Bioconductor `SingleCellExperiment` objects. Contains paired empirical reference profiles (top 100 HVGs x 100 balanced cells for RNA, top 100 HVPs x 100 identical paired cells for ATAC) covering all 8 biological cell types equally (12–13 cells per type) and 2 biological batches (`Bud_21`, `Bud_22`), alongside simulated outputs from 6 single-cell multiomics simulators (`scMultiSim`, `scDesign3`, `MOSim`, `scCross`, `scMoMtF`, and `Matilda`) and observed scalability benchmarks (runtime in seconds, peak RAM in MiB, and CPU cores).

Usage

```
data(example_rice_bud)
```

Format

A list with 4 elements:

metadata List containing dataset name, organism, tissue, pairing mode (`"paired"`), dimensions, and simulators.

reference List containing standardized `SingleCellExperiment` objects: `combined` (multi-modal `SingleCellExperiment` with RNA counts and `altExp(combined, "ATAC")` sharing identical paired cells and `colData`), `rna` (unimodal `SingleCellExperiment` for scRNA-seq), and `atac` (unimodal `SingleCellExperiment` for scATAC-seq).

simulated Named list of simulated outputs for 6 simulators: `scMultiSim`, `scDesign3`, `MOSim`, `scCross`, `scMoMtF`, and `Matilda`. Each simulator entry contains `combined`, `rna`, and `atac` `SingleCellExperiment` objects matching the reference structure.

scalability_benchmarks A data.frame with benchmarked `Method`, `Runtime_Seconds`, `PeakRAM_MiB`, and `Sim_Cores`.

Source

Gene Expression Omnibus (GEO) GSE232863. Preprocessed via standard rice tissue pipeline.

Examples

```
data(example_rice_bud)
example_rice_bud$reference$combined
dim(example_rice_bud$reference$rna)
dim(example_rice_bud$reference$atac)
names(example_rice_bud$simulated)
example_rice_bud$scalability_benchmarks
```

```
res <- evaluate_multiomics_accuracy(
  ref_multi = list(RNA = example_rice_bud$reference$rna,
                   ATAC = example_rice_bud$reference$atac),
  sim_multi = list(RNA = example_rice_bud$simulated$scDesign3$rna,
                   ATAC = example_rice_bud$simulated$scDesign3$atac),
  pairing = "paired",
  verbose = FALSE
)
head(res$benchmark_summary_table, 8)
```

`example_scatac` *Example scATAC-seq Benchmarking Dataset (Rice Bud)*

Description

A real, subsampled single-cell ATAC sequencing peak accessibility dataset from rice bud tissue (*Oryza sativa*) designed for testing, benchmarking, and demonstrating evaluation metrics in `scSimEval`. Contains empirical reference peak counts and `scDesign3`-simulated counts across the top 100 highly variable peaks (HVPs) and the same 100 balanced paired cells as `example_scrna`.

Usage

```
data(example_scatac)
```

Format

A list with 4 elements:

ref A `SingleCellExperiment` object containing reference peak accessibility counts (100 HVPs x 100 cells), cell types and batch annotations in `colData`, and precomputed `PCA` in `reducedDims`.

sim A `SingleCellExperiment` object containing simulated peak accessibility counts (100 HVPs x 100 cells), cell types and batch annotations in `colData`, and precomputed `PCA` in `reducedDims`.

cell_types A named factor of cell-type identities across 8 Rice Bud cell types for the 100 cells.

batch_info A named factor of batch annotations (`Bud_21`, `Bud_22`) for the 100 cells.

Source

Derived from rice bud paired multiomics data (*Oryza sativa*, GEO accession GSE232863).

Examples

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

`example_scrna` *Example scRNA-seq Benchmarking Dataset (Rice Bud)*

Description

A real, subsampled single-cell RNA sequencing dataset from rice bud tissue (*Oryza sativa*) designed for testing, benchmarking, and demonstrating evaluation metrics in `scSimEval`. Contains empirical reference counts and `scDesign3`-simulated counts across the top 100 highly variable genes (HVGs) and 100 balanced cells covering all 8 biological cell types and 2 batches.

Usage

```
data(example_scrna)
```

Format

A list with 4 elements:

ref A `SingleCellExperiment` object containing reference scRNA-seq counts (100 HVGs x 100 cells), cell types and batch annotations in `colData`, and precomputed `PCA` in `reducedDims`.

sim A `SingleCellExperiment` object containing simulated scRNA-seq counts (100 HVGs x 100 cells), cell types and batch annotations in `colData`, and precomputed `PCA` in `reducedDims`.

cell_types A named factor of cell-type identities across 8 Rice Bud cell types for the 100 cells.

batch_info A named factor of batch annotations (`Bud_21`, `Bud_22`) for the 100 cells.

Source

Derived from rice bud paired multiomics data (*Oryza sativa*, GEO accession GSE232863).

Examples

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

`example_soybean_root` *Real Unpaired Single-Cell Multiomics Benchmark Dataset (Soybean Root, SingleCellExperiment)*

Description

A real, pre-processed unpaired single-cell RNA-seq and ATAC-seq benchmark dataset of soybean root tissue (*Glycine max*), standardized using Bioconductor `SingleCellExperiment` objects. Contains independent empirical reference profiles (top 100 HVGs x 100 balanced RNA cells and top 100 HVPs x 100 independently sampled balanced ATAC cells) covering all 8 biological cell types (`Cortex`, `Endodermis`, `Epidermis`, `Pericycle`, `Phloem`, `Procambium`, `Root_hair`, `Xylem`; 12–13 cells per type), alongside simulated outputs from 3 unpaired multiomics simulators (`scMultiSim`, `scDesign3`, and `scCross`) and observed scalability benchmarks.

Usage

```
data(example_soybean_root)
```

Format

A list with 4 elements:

metadata List containing dataset name, organism, tissue, pairing mode (`"unpaired"`), dimensions, and simulators.

reference List containing standardized `SingleCellExperiment` objects: `combined` (multi-modal `SingleCellExperiment` with `colData$modality` distinguishing independent RNA and ATAC cells across `altExp(combined, "ATAC")`), `rna` (unimodal `SingleCellExperiment` for scRNA-seq), and `atac` (unimodal `SingleCellExperiment` for scATAC-seq).

simulated Named list of simulated outputs for 3 simulators: `scMultiSim`, `scDesign3`, and `scCross`. Each simulator entry contains `combined`, `rna`, and `atac` `SingleCellExperiment` objects matching the reference structure.

scalability_benchmarks A data.frame with benchmarked runtime, peak RAM, and CPU cores for RNA, ATAC, and combined execution.

Source

Preprocessed from soybean root single-cell atlas datasets.

Examples

```
data(example_soybean_root)
example_soybean_root$reference$combined
dim(example_soybean_root$reference$rna)
dim(example_soybean_root$reference$atac)
names(example_soybean_root$simulated)
example_soybean_root$scalability_benchmarks
```

```
res <- evaluate_multiomics_accuracy(
  ref_multi = list(RNA = example_soybean_root$reference$rna,
                   ATAC = example_soybean_root$reference$atac),
  sim_multi = list(RNA = example_soybean_root$simulated$scMultiSim$rna,
                   ATAC = example_soybean_root$simulated$scMultiSim$atac),
  pairing = "unpaired",
  verbose = FALSE
)
head(res$benchmark_summary_table, 8)
```

`normalize_bubble_scores` *Normalize Bubble Matrix Scores Across Benchmark Datasets and Simulators*

Description

Rescales metric raw scores to a 0-1 range where 1 is always best, taking into account whether the metric is higher-is-better or lower-is-better, naturally bounded, or unbounded.

Usage

```
normalize_bubble_scores(df)
```

Arguments

`df`	A data frame of benchmark results containing at least `Metric` and `Score_Raw` (or `Score`).

Value

The input data frame with `Score_Norm` populated in `[0, 1]` (higher is always better).

Examples

```
df <- data.frame(
  Method = rep(c("scDesign3", "Splatter", "SymSim"), each = 2),
  Metric = rep(c("KS", "ARI"), 3),
  Value  = c(0.12, 0.85, 0.28, 0.68, 0.41, 0.52),
  stringsAsFactors = FALSE
)
norm_df <- normalize_bubble_scores(df)
print(norm_df)
```

`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,
  assay_modality = NULL,
  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,
  show_category_bars = TRUE
)

plot_bubble_matrix(
  data,
  assay_modality = NULL,
  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,
```

```
  show_category_bars = TRUE
)
```

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).
assay_modality	Optional character scalar ("all", "RNA", or "ATAC") specifying the assay modality scope to dynamically format Category V metrics with the appropriate differential prefix ("DE/DA ", "DE ", or "DA "). If NULL, auto-detected from data.
method_classes	Optional named list mapping method names to row group labels.
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 c(2.0, 7.8).
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.
show_category_bars	Logical. Append a compact horizontal summary bar (normalized category mean score) at the right end of each category panel. Default TRUE.

Details

Bubble encoding:

- **Size**: Normalized fidelity score [0, 1] – larger = better performance.
- **Fill color**: Evaluation category (one of the eight canonical groups).
- **Circle (standard outline)**: Standard performance for that metric.

- **Bold outline circle (shape 21, stroke 1.6)**: Top-performing simulator for that specific metric.
- **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

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim"), each = 6),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
    "(II) Sparsity & Zero-Inflation Dynamics",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics",
    "(VIII) Computational Scalability & Efficiency"
  ), 4),
  Metric = rep(c("KS", "Wasserstein", "Zero_Prob_RMSE",
                 "Silhouette_ASW", "CMS", "Core_Seconds"), 4),
  Value = c(
    0.11, 0.18, 0.08, 0.82, 0.86, 42,
    0.15, 0.22, 0.11, 0.78, 0.81, 68,
    0.23, 0.34, 0.17, 0.65, 0.72, 18,
    0.31, 0.45, 0.24, 0.56, 0.64, 95
  ),
  stringsAsFactors = FALSE
)
p <- plot_benchmark_bubble_matrix(bm_df, base_size = 9.5) +
  ggplot2::theme(legend.box = "vertical")
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 evaluate_simulation_accuracy()) 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

```
bm_df <- data.frame(
  Method   = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim"), each = 4),
  Category = "(I) Distributional Properties",
  Metric   = rep(c("KS", "Wasserstein", "MAD", "RMSE"), 4),
  Value    = c(
    0.10, 0.16, 0.12, 0.19,
    0.15, 0.22, 0.16, 0.25,
    0.23, 0.34, 0.26, 0.36,
    0.31, 0.44, 0.35, 0.47
  ),
  stringsAsFactors = FALSE
)
p <- plot_category_metric_bars(
  bm_df,
  category = "(I) Distributional Properties",
  base_size = 9.5
)
print(p)
```

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 18 distributional properties: 7 univariate cell-level properties, 7 univariate feature-level properties, and 4 bivariate property pairs, either as empirical & joint density overlays or as quantile-quantile (Q-Q) concordance plots.

Usage

```
plot_distribution_qc(
  ref_data,
  sim_data,
  layout = c("density", "qq", "comprehensive"),
  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. list("scDesign3" = m1, "Splatter" = m2)).
layout	Character. Visualization layout: "qq" (18-panel quantile-quantile diagnostic plots comparing real vs. simulated quantiles on a common n = 1024 grid with KS, MAD, MAE, RMSE, and PCC annotations) or "density" / "comprehensive" (18-panel empirical univariate density curves and bivariate joint distributions across the exact same 18 properties).
properties	Optional character vector of specific properties or group keywords ("all", "cell", "feature", "bivariate", or a single property key) to include.
title	Character. Main plot title. Default "Data properties" (uses "Real vs. Simulated Q-Q Plots" for layout = "qq" and "Real vs. Simulated Empirical & Joint Density Plots" for density layout when left at default). 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 (retained for backward compatibility).
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")
p <- plot_distribution_qc(example_scrna$ref, example_scrna$sim, base_size = 8.5)
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 evaluation categories",
  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 evaluation categories",
  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 evaluation categories",
  base_size = 11
)
```

Arguments

`data`	A benchmark summary data frame, a list containing `benchmark_summary_table`, or the output of `evaluate_simulation_accuracy`.
`method_classes`	Optional named vector mapping method names to class/category labels (e.g., `c("Splatter" = "Class 1", "scDesign3" = "Class 2")`). If provided, rows are grouped into class panels.
`category_colors`	Optional named vector specifying custom colors for category panels and overall score.

normalize_scores
: Logical indicating whether to normalize scores to [0, 1] (direction-aware where larger is better). Default is TRUE.

simulators_order
: Optional vector specifying a custom ordering of simulators. Default is ranking by descending Overall Score.

show_labels
: 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

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim"), each = 6),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
    "(II) Sparsity & Zero-Inflation Dynamics",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics",
    "(V) Biological Signal & Differential Fidelity"
  ), 4),
  Metric = rep(c("KS", "Wasserstein", "Zero_Prob_RMSE",
                 "ARI", "CMS", "DE_F1"), 4),
  Value = c(
    0.10, 0.16, 0.07, 0.85, 0.88, 0.82,
    0.14, 0.21, 0.11, 0.79, 0.83, 0.77,
    0.23, 0.34, 0.17, 0.67, 0.72, 0.66,
    0.31, 0.44, 0.24, 0.56, 0.62, 0.55
  ),
  stringsAsFactors = FALSE
)
p <- plot_evaluation_summary(bm_df, base_size = 9.5)
print(p)
```

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

```
bm_df <- data.frame(
  Method   = c("scDesign3", "scMultiSim", "MOSim", "Splatter", "SymSim"),
  Category = "(I) Distributional Properties",
  Metric   = "KS",
  Value    = c(0.10, 0.14, 0.18, 0.23, 0.31),
  stringsAsFactors = FALSE
)
p <- plot_individual_metric_bar(bm_df, metric = "KS", base_size = 10)
print(p)
```

plot_metric_boxplots *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_accura 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

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim"), each = 6),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
    "(I) Distributional Properties",
    "(III) Cellular Manifold & Clustering Concordance",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics"
  ), 4),
  Metric = rep(c("KS", "Wasserstein", "RMSE", "ARI", "Silhouette_ASW", "CMS"), 4),
  Value = c(
    0.10, 0.16, 0.19, 0.85, 0.81, 0.88,
    0.15, 0.22, 0.25, 0.78, 0.75, 0.82,
    0.23, 0.34, 0.36, 0.67, 0.64, 0.72,
    0.31, 0.44, 0.47, 0.56, 0.54, 0.63
  ),
  stringsAsFactors = FALSE
)
p <- plot_metric_boxplots(bm_df, ncol = 3, base_size = 9.5)
print(p)
```

---

plot_metric_heatmap *Plot Multi-Simulator Comparative Metric Heatmap Across Canonical Categories*

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 standardized relative fidelity or original raw values inside every cell, with explicit optimization-direction arrows (up-arrow = higher is better, down-arrow = lower is better) across all 56 curated evaluation measures (14 Distributional, 5 Sparsity, 8 Cellular Manifold, 5 Batch Mixing, 12 Biological Signal DE/DA, 5 Multiomics Integration, 5 Trajectory, and 2 Scalability).

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

benchmark_data	A tidy benchmark summary table or a named list of benchmark tables.
category	Optional character vector to filter by canonical evaluation category. Default NULL (all 8 categories).
metrics	Optional character vector of specific metrics to include. Default NULL (all).
scale_fill	Character. Cell label & color mode: "relative" (default; cell text and tile fill both show per-metric direction-aware normalized fidelity in [0, 1]) or "raw" (cell text displays original unnormalized raw values in native units while tile shading still scales per metric by direction-aware fidelity in [0, 1] so high-magnitude metrics like runtime or RAM do not wash out color contrast across rows).
top_n_properties	Optional integer. If specified, restricts to top N most variable metrics. Default NULL (all metrics).
facet_by_category	Logical. Whether to group rows into vertical category banner panels across all 8 categories. Default TRUE.
cluster_rows	Logical. Hierarchically cluster measures along y-axis. Default FALSE.
cluster_cols	Logical. Hierarchically cluster simulators along x-axis. Default FALSE.
base_size	Numeric. Base font size. Default 10.
metric_name	Optional legacy alias for metrics.

Details

Value

A ggplot object.

Examples

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "MOSim", "Splatter", "SymSim"), each = 6),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
```

```
    "(II) Sparsity & Zero-Inflation Dynamics",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics",
    "(V) Biological Signal & Differential Fidelity"
  ), 5),
  Metric = rep(c("KS", "Wasserstein", "r2_discrepancy", "ari", "cms", "rna_de_f1"), 5),
  Value = c(
    0.10, 0.16, 0.07, 0.85, 0.88, 0.82,
    0.14, 0.21, 0.11, 0.79, 0.83, 0.77,
    0.18, 0.27, 0.13, 0.74, 0.78, 0.72,
    0.23, 0.34, 0.17, 0.67, 0.72, 0.66,
    0.31, 0.44, 0.24, 0.56, 0.62, 0.55
  ),
  stringsAsFactors = FALSE
)
p <- plot_metric_heatmap(bm_df, base_size = 9.5)
print(p)
```

`plot_metric_mds` *Multi-Dimensional Scaling (MDS) Ordination of Evaluation Metrics or Simulators*

Description

Projects all evaluated benchmark metrics (up to 56 metrics across 8 canonical categories) 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"),
  assay_modality = c("auto", "scRNA-seq", "scATAC-seq", "multiomics"),
  group_by = c("category", "summary_type"),
  by_category = FALSE,
  as_list = FALSE,
  category = NULL,
  exclude_categories = NULL,
  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 "simulators" (ordinated points represent candidate simulators) or "summaries" (ordinated points represent biological and computational evaluation metrics).
assay_modality	Character string specifying the molecular modality: "auto" (default; automatically detects from data attributes or metrics), "scRNA-seq", "scATAC-seq", or "multiomics". When "scATAC-seq", feature-level metrics are labeled "peak". When "scRNA-seq", feature-level metrics are labeled "gene". When "multiomics", feature-level metrics are labeled "features".
group_by	Character. Grouping variable for metric points when ordination_by = "summaries": "category" (default) groups points by evaluation category (I–VIII); "summary_type" groups by feature/cell/global level.
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	Optional character vector of categories to exclude from the ordination. Default is NULL (includes all 8 canonical categories / 56 curated metrics).
metrics	Optional character vector of specific metrics to include.
normalize_scores	Logical. If TRUE (default), scores are direction-aware normalized to [0, 1] before aggregation so that metrics on disparate scales (e.g. MiB vs distance) do not distort Euclidean distance.
palette	Optional named character vector of colors. For ordination_by = "summaries", maps categories or feature-level ("peak", "gene", or "features"), "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 metric annotated with its optimization-direction arrow (up-arrow = higher raw value better, down-arrow = lower raw value better), colored by canonical category (I-VIII) or biological level ("gene", "peak", "features", "cell", "global"). If by_category = TRUE, a multi-panel dashboard is produced showing one MDS ordination per category (automatically skipping categories with fewer than 3 measures).

Value

A ggplot or patchwork object.

Examples

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim", "MOSim"), each = 8),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
    "(I) Distributional Properties",
    "(II) Sparsity & Zero-Inflation Dynamics",
    "(III) Cellular Manifold & Clustering Concordance",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics",
    "(V) Biological Signal & Differential Fidelity"
  ), 5),
  Metric = rep(c("KS", "Wasserstein", "RMSE", "r2_discrepancy",
                 "ari", "silhouette_sim", "cms", "rna_de_f1"), 5),
  Value = c(
    0.10, 0.16, 0.22, 0.07, 0.85, 0.74, 0.88, 0.84,
    0.18, 0.20, 0.24, 0.15, 0.82, 0.81, 0.76, 0.79,
    0.21, 0.35, 0.31, 0.12, 0.64, 0.70, 0.80, 0.61,
    0.34, 0.42, 0.48, 0.26, 0.59, 0.52, 0.61, 0.68,
    0.15, 0.29, 0.26, 0.19, 0.75, 0.66, 0.70, 0.77
  ),
  stringsAsFactors = FALSE
)
p <- plot_metric_mds(bm_df, base_size = 9.5)
print(p)
```

---

plot_metric_pca *Principal Component Analysis (PCA) Dashboard of Benchmark Metrics and Methods*

Description

Generates a dual-panel PCA dashboard for evaluating single-cell simulation benchmark metrics and methods across all 56 curated evaluation measures.

Usage

```
plot_metric_pca(
  benchmark_data,
  panel = c("both", "methods", "loadings"),
  assay_modality = c("auto", "scRNA-seq", "scATAC-seq", "multiomics"),
  group_by = c("category", "summary_type"),
  by_category = FALSE,
  as_list = FALSE,
  category = NULL,
  exclude_categories = NULL,
  metrics = NULL,
  normalize_scores = TRUE,
  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).
assay_modality	Character string specifying the molecular modality: "auto" (default; automatically detects from data attributes or metrics), "scRNA-seq", "scATAC-seq", or "multiomics". When "scATAC-seq", feature-level metrics are labeled "peak". When "scRNA-seq", feature-level metrics are labeled "gene". When "multiomics", feature-level metrics are labeled "features".
group_by	Character. Grouping variable for loading vectors in panel b: "category" (default) groups vectors by evaluation category (I–VIII); "summary_type" groups by feature/cell/global level.
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 (default NULL, includes all 8 categories).
metrics	Optional character vector of specific metrics to include.
normalize_scores	Logical. If TRUE (default), direction-align and min-max normalize scores to [0, 1] before aggregation and PCA decomposition.
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 56 to display all evaluated metrics.
base_size	Numeric. Base font size. Default 11.

Details

All metrics present in benchmark_data (up to all 56 curated evaluation metrics across 8 canonical categories) are direction-normalized before PCA decomposition so higher values uniformly correspond to higher fidelity. 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 canonical category or summary type and annotated with optimization-direction arrows (up-arrow / down-arrow). Pass top_n_loadings = NULL or 56 to render all metric loading vectors.

Value

A ggplot or patchwork object representing the ordination figure.

Examples

```
bm_df <- data.frame(
  Method = rep(c("scDesign3", "scMultiSim", "Splatter", "SymSim", "MOSim"), each = 8),
  Category = rep(c(
    "(I) Distributional Properties", "(I) Distributional Properties",
    "(I) Distributional Properties",
    "(II) Sparsity & Zero-Inflation Dynamics",
    "(III) Cellular Manifold & Clustering Concordance",
    "(III) Cellular Manifold & Clustering Concordance",
    "(IV) Technical Variation & Batch Mixing Dynamics",
    "(V) Biological Signal & Differential Fidelity"
  ), 5),
  Metric = rep(c("KS", "Wasserstein", "RMSE", "r2_discrepancy",
                 "ari", "silhouette_sim", "cms", "rna_de_f1"), 5),
  Value = c(
    0.10, 0.16, 0.22, 0.07, 0.85, 0.74, 0.88, 0.84,
    0.18, 0.20, 0.24, 0.15, 0.82, 0.81, 0.76, 0.79,
    0.21, 0.35, 0.31, 0.12, 0.64, 0.70, 0.80, 0.61,
    0.34, 0.42, 0.48, 0.26, 0.59, 0.52, 0.61, 0.68,
    0.15, 0.29, 0.26, 0.19, 0.75, 0.66, 0.70, 0.77
  ),
  stringsAsFactors = FALSE
)
p <- plot_metric_pca(bm_df, base_size = 9.5)
print(p)
```

---

plot_scalability_benchmark

Plot Multi-Dimensional Computational Scalability Benchmark

Description

Evaluates computational resource efficiency across single-cell simulators using core-normalized Category VIII benchmarks: **Core-Seconds** (Elapsed Time (s) $\times$ N_{cores}; Lannelongue et al., 2021; Wratten et al., 2021; Grealey et al., 2022) and **Peak RAM per Core** (Peak RAM$/N_{\text{cores}}$; Holtgrewe et al., 2015; Boisvert et al., 2012; Chen et al., 2022), alongside resource tradeoff frontiers, cost footprints, and simulation throughput.

Usage

```
plot_scalability_benchmark(
  benchmark_data,
  type = c("composite", "runtime", "memory", "tradeoff", "cost", "throughput",
    "efficiency"),
  layout = c("4panel", "6panel"),
```

```
  modality = c("total", "rna", "atac"),
  cell_count = 1000,
  palette = NULL,
  base_size = 11,
  time_unit = c("core_seconds", "core_hours", "seconds", "minutes"),
  memory_unit = c("MiB/core", "GiB/core", "MB/core", "GB/core", "MiB", "MB")
)
```

Arguments

benchmark_data	Benchmark summary table or named list containing Category VIII metrics.
type	Character. Visualization type: "composite" (multi-panel dashboard), "runtime" (core-seconds CPU work barplot), "memory" (peak RAM per core barplot), "tradeoff" (core-seconds vs memory-per-core scatter 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".
modality	Character string specifying the modality layer to benchmark: "total" (default, combined runtime and memory), "rna" (scRNA-seq layer only), or "atac" (scATAC-seq layer only).
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.
time_unit	Character. Display unit for runtime: "core_seconds" (default), "core_hours", "seconds", or "minutes".
memory_unit	Character. Display unit for peak RAM per core: "MiB/core" (default), "GiB/core", "MB/core", or "GB/core".

Details

Value

A ggplot or patchwork object.

References

Lannelongue, L., et al. (2021). Green Algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12), 2100707.

Holtgrewe, M., et al. (2015). Methods for the detection and assembly of novel sequence. *Bioinformatics*, 31(12), 1904-1912.

Boisvert, S., et al. (2012). Ray Meta: scalable de novo metagenome assembly. *Genome Biology*, 13(12), R122.

Chen, W., et al. (2022). STEPS 4.0: Fast and memory-efficient simulations. *Frontiers in Neuroinformatics*, 16, 883742.

Examples

```
data(example_rice_bud, package = "scSimEval")
sb <- example_rice_bud$scalability_benchmarks
scal_df <- rbind(
  data.frame(
    Method    = sb$Method,
    Category  = "(VIII) Computational Scalability & Efficiency",
    Metric    = "Core_Seconds",
   Value    = mapply(compute_core_seconds, sb$Runtime_Seconds, sim_cores = sb$Sim_Cores),
    Sim_Cores = sb$Sim_Cores,
    stringsAsFactors = FALSE
  ),
  data.frame(
    Method    = sb$Method,
    Category  = "(VIII) Computational Scalability & Efficiency",
    Metric    = "Memory_Per_Core_MiB",
   Value    = mapply(compute_memory_per_core, sb$PeakRAM_MiB, sim_cores = sb$Sim_Cores),
    Sim_Cores = sb$Sim_Cores,
    stringsAsFactors = FALSE
  )
)
p <- plot_scalability_benchmark(scal_df, type = "composite", layout = "4panel", base_size = 9.5)
print(p)
```

Index

∗ **datasets**
  example_multiomics, 32
  example_rice_bud, 33
  example_scatac, 34
  example_scrna, 35
  example_soybean_root, 36

append_benchmark_results
  *(*combine_benchmark_results*)*, 6

benchmark_resource_usage, 4
benchmark_simulation_suite, 5

calc_cdi, *18, 19*
combine_benchmark_results, 6
compute_core_seconds, 8
compute_dataset_embeddings, 9, *11*, *42*
compute_embedding_quality_metrics, 10
compute_memory_per_core, 12
compute_method_leaderboard, 13

evaluate_accessibility_sparsity_curve, 14
evaluate_atac_da_fidelity, 15
evaluate_batch_metrics, 17
evaluate_clustering_metrics, 18
evaluate_deg_fidelity, 19
evaluate_multiomics_accuracy, 22, *28*
evaluate_multiomics_integration, 24
evaluate_multiple_datasets, 26, *39*
evaluate_rna_de_fidelity
  *(*evaluate_deg_fidelity*)*, 19
evaluate_simulation_accuracy, *13*, 27, *39*, *41*, *45*, *48*
evaluate_trajectory_metrics, 29
evaluate_zero_probability_curve, 31
example_multiomics, 32
example_rice_bud, 33
example_scatac, 34
example_scrna, 35
example_soybean_root, 36

normalize_bubble_scores, 37

plot_benchmark_bubble_matrix, *5*, *7*, 38
plot_benchmark_summary_bars
  *(*plot_evaluation_summary*)*, 44
plot_bubble_matrix
  *(*plot_benchmark_bubble_matrix*)*, 38
plot_category_metric_bars, 40
plot_dataset_embeddings, 42
plot_distribution_qc, 43
plot_evaluation_summary, 44
plot_individual_metric_bar, 47
plot_metric_boxplots, 48
plot_metric_heatmap, 49
plot_metric_mds, 51
plot_metric_pca, 53
plot_scalability_benchmark, 55
plot_summary_bars
  *(*plot_evaluation_summary*)*, 44