Package: scCertify (via r-universe)

November 25, 2026

Title Explainable Confidence Scoring for Single-Cell Annotations

Version 1.0.0

Description scCertify provides an explainable framework for assessing the confidence of cell-type annotations in single-cell RNA sequencing data. Confidence scores are computed using marker enrichment, neighborhood agreement, entropy-based uncertainty estimation, and ontology-aware label matching to support transparent interpretation of annotation quality.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Imports Seurat, SingleCellExperiment, SummarizedExperiment, FNN, entropy, Matrix, BiocParallel, UCell

Suggests BiocStyle, SingleR, celldex, ggplot2, knitr, rmarkdown, scDblFinder, SeuratObject, testthat (>= 3.0.0)

biocViews Software, SingleCell, RNASeq, GeneExpression, Sequencing

URL https://github.com/Jaya-Surya-dev/scCertify

BugReports https://github.com/Jaya-Surya-dev/scCertify/issues

VignetteBuilder knitr

Config/testthat/edition 3

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

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:56:50 UTC

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

RemoteRef RELEASE_3_24

RemoteSha 21cdca1f54587389c7dffa848ebd448f1fb1f6f5

Contents

scCertify-package . . . 2
calibrate_confidence . . . 4
cell_certify . . . 4
classify_confidence . . . 6
de_novo_signatures . . . 7
discovery_aware . . . 8
discovery_status . . . 10
entropy_score . . . 11
explain_cell . . . 12
explain_confidence . . . 13
marker_score . . . 14
match_labels . . . 16
negative_marker_score . . . 16
neighbor_score . . . 18

Index **20**

`scCertify-package` *scCertify: Explainable Confidence Scoring for Single-Cell Annotations*

Description

`scCertify` provides an explainable framework for evaluating the confidence of single-cell RNA sequencing (scRNA-seq) cell-type annotations.

The package combines multiple complementary sources of evidence, including marker gene enrichment, neighborhood agreement, entropy-based uncertainty, and doublet information, to produce calibrated confidence scores and interpretable confidence classes.

The package is designed to complement existing cell-type annotation methods rather than replace them. Predicted cell labels are supplied by the user and scCertify evaluates the confidence of those assignments.

Both `SingleCellExperiment` and `Seurat` objects are supported.

Details

The main workflow consists of:

1. obtaining predicted cell-type labels from an annotation method;
2. supplying marker genes for the predicted cell types;
3. evaluating marker consistency;
4. evaluating neighborhood agreement;
5. incorporating entropy-based uncertainty and doublet information;
6. combining these components into a calibrated confidence score; and
7. interpreting the resulting confidence using explanation functions.

Main functions

- `cell_certify` computes integrated confidence scores.
- `marker_score` evaluates marker consistency.
- `neighbor_score` evaluates neighborhood agreement.
- `entropy_score` estimates entropy-based uncertainty.
- `calibrate_confidence` calibrates confidence scores.
- `classify_confidence` assigns confidence classes.
- `explain_cell` summarizes confidence for an individual cell.
- `explain_confidence` identifies major confidence contributors.
- `match_labels` assists with label matching.

Supported data structures

`scCertify` supports both `SingleCellExperiment` and `Seurat` objects. `SingleCellExperiment` is the primary object class used in the package documentation and vignette, consistent with Bioconductor conventions. Seurat compatibility is provided for interoperability with existing single-cell analysis workflows.

Confidence framework

Confidence scores combine four sources of evidence:

- marker enrichment;
- neighborhood agreement;
- entropy-based uncertainty; and
- doublet information.

The resulting confidence score is classified into Low, Moderate, or High confidence according to user-defined thresholds.

Author(s)

Maintainer: Jaya Surya Doddetipalli <jayasuryadoddetipalli@gmail.com> (ORCID)

Authors:

- Jaya Surya Doddetipalli <jayasuryadoddetipalli@gmail.com> (ORCID)

See Also

Useful links:

- https://github.com/Jaya-Surya-dev/scCertify
- Report bugs at https://github.com/Jaya-Surya-dev/scCertify/issues

calibrate_confidence *Calibrate Confidence Scores*

Description

Converts raw confidence scores into calibrated percentile scores.

Usage

```
calibrate_confidence(scores)
```

Arguments

scores	Numeric vector

Value

Numeric vector

Examples

```
scores <- c(
  0.2,
  0.5,
  0.8
)

calibrate_confidence(
  scores
)
```

cell_certify *Cell Annotation Confidence*

Description

Computes confidence scores for cell-type annotations using marker enrichment, neighborhood agreement, entropy-based uncertainty, and doublet information.

Usage

```
cell_certify(
  object,
  markers,
  label_column = "predicted_label",
  moderate_threshold = 0.5,
  high_threshold = 0.8,
  run_discovery = FALSE,
  negative_markers = NULL
)
```

Arguments

object	A Seurat or SingleCellExperiment object.
markers	Named list of marker genes.
label_column	Metadata column containing predicted labels.
moderate_threshold	Lower threshold for the "Moderate" confidence class.
high_threshold	Lower threshold for the "High" confidence class.
run_discovery	Logical. Whether to run the optional discovery-aware annotation analysis. Defaults to FALSE.
negative_markers	Named list of negative marker genes, required when run_discovery = TRUE.

Value

The input object with confidence scores and classifications added to the metadata. When run_discovery = TRUE, discovery-aware scores and statuses are also added to the metadata.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(1000, lambda = 5),
  nrow = 50,
  ncol = 20,
  dimnames = list(
    paste0("Gene", seq_len(50)),
    paste0("Cell", seq_len(20))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)
```

```
SummarizedExperiment::colData(sce)$predicted_label <-
  rep(c("T cell", "B cell"), each = 10)

markers <- list(
  "T cell" = c("Gene1", "Gene2"),
  "B cell" = c("Gene3", "Gene4")
)

negative_markers <- list(
  "T cell" = c("Gene3", "Gene4"),
  "B cell" = c("Gene1", "Gene2")
)

SingleCellExperiment::reducedDim(sce, "PCA") <- prcomp(
  t(counts),
  rank. = 5
)$x

# Original workflow: discovery analysis is disabled by default.
sce <- cell_certify(
  sce,
  markers
)

head(
  SummarizedExperiment::colData(sce)$confidence_score
)

# Optional discovery-aware workflow.
sce_discovery <- cell_certify(
  sce,
  markers,
  run_discovery = TRUE,
  negative_markers = negative_markers
)

head(
  SummarizedExperiment::colData(sce_discovery)$discovery_status
)
```

`classify_confidence` *Classify Confidence Levels*

Description

Converts confidence scores into confidence classes.

Usage

```
classify_confidence(scores, moderate_threshold = 0.5, high_threshold = 0.8)
```

Arguments

scores	Numeric vector of confidence scores.
moderate_threshold	Lower threshold for the "Moderate" confidence class.
high_threshold	Lower threshold for the "High" confidence class.

Value

Character vector of confidence classes.

Examples

```
scores <- c(
  0.2,
  0.7,
  0.9
)

classify_confidence(
  scores
)

classify_confidence(
  scores,
  moderate_threshold = 0.4,
  high_threshold = 0.75
)
```

---

de_novo_signatures *Cluster-Enriched Gene Signatures*

Description

Identifies genes that are relatively enriched within each user-specified cluster.

Usage

```
de_novo_signatures(object, cluster_column, top_n = 10)
```

Arguments

object	A Seurat or SingleCellExperiment object.
cluster_column	Metadata column containing cluster identities.
top_n	Number of top enriched genes to return for each cluster.

Details

This function provides an exploratory, discovery-aware approach by identifying genes that are relatively enriched within existing clusters compared with the remaining cells.

The function does not perform clustering, differential expression testing, or biological cell-type annotation. The resulting cluster-enriched gene sets can be used to investigate heterogeneous, unusual, or potentially novel cell states.

Value

Named list containing the top cluster-enriched genes for each cluster.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(200, lambda = 5),
  nrow = 20,
  ncol = 10,
  dimnames = list(
    paste0("Gene", seq_len(20)),
    paste0("Cell", seq_len(10))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$cluster <- rep(
  c("Cluster1", "Cluster2"),
  each = 5
)

signatures <- de_novo_signatures(
  sce,
  cluster_column = "cluster",
  top_n = 5
)

signatures
```

`discovery_aware` *Discovery-Aware Annotation Analysis*

Description

Combines positive and contradictory marker evidence to provide a discovery-aware interpretation of predicted cell identities.

Usage

```
discovery_aware(
  object,
  markers,
  negative_markers,
  label_column = "predicted_label",
  positive_threshold = 0.5,
  negative_threshold = 0.5
)
```

Arguments

`object`	A Seurat or SingleCellExperiment object.
`markers`	Named list of positive marker genes.
`negative_markers`	Named list of negative marker genes.
`label_column`	Metadata column containing predicted labels.
`positive_threshold`	Threshold for strong positive evidence.
`negative_threshold`	Threshold for strong contradictory evidence.

Details

This function provides an optional exploratory layer for investigating heterogeneous or potentially unusual cell states. It does not modify the main scCertify confidence score.

The function combines positive marker evidence from `marker_score()` with contradictory evidence from `negative_marker_score()` and classifies each cell using `discovery_status()`.

Value

A data.frame containing positive marker scores, negative marker scores, and discovery-aware annotation status.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(200, lambda = 5),
  nrow = 20,
  ncol = 10,
  dimnames = list(
```

```
    paste0("Gene", seq_len(20)),
    paste0("Cell", seq_len(10))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$predicted_label <- rep(
  c("T cell", "B cell"),
  each = 5
)

markers <- list(
  "T cell" = c("Gene1", "Gene2"),
  "B cell" = c("Gene3", "Gene4")
)

negative_markers <- list(
  "T cell" = c("Gene3", "Gene4"),
  "B cell" = c("Gene1", "Gene2")
)

result <- discovery_aware(
  sce,
  markers = markers,
  negative_markers = negative_markers
)

head(result)
```

`discovery_status` *Discovery-Aware Annotation Status*

Description

Classifies cells according to positive marker evidence and contradictory negative marker evidence.

Usage

```
discovery_status(
  marker_score,
  negative_marker_score,
  positive_threshold = 0.5,
  negative_threshold = 0.5
)
```

Arguments

marker_score	Numeric vector of positive marker scores.
negative_marker_score	Numeric vector of negative marker scores.
positive_threshold	Threshold for strong positive evidence. Must be a single numeric value between 0 and 1.
negative_threshold	Threshold for strong contradictory evidence. Must be a single numeric value between 0 and 1.

Details

The function provides a simple exploratory interpretation of annotation evidence. It does not modify the main confidence score and does not replace biological validation.

Cells with strong positive evidence and weak contradictory evidence are classified as "Known". Cells with weak positive and weak contradictory evidence are classified as "Possible novel state". Cells with both strong positive and contradictory evidence are classified as "Possible transitional state". Remaining cells are classified as "Insufficient evidence".

Value

Character vector containing one discovery-aware status per cell.

Examples

```
positive_scores <- c(0.9, 0.2, 0.8, 0.3)
negative_scores <- c(0.1, 0.2, 0.9, 0.8)

discovery_status(
  marker_score = positive_scores,
  negative_marker_score = negative_scores
)
```

---

entropy_score *Entropy-Based Uncertainty Score*

Description

Calculates entropy from prediction score matrices.

Usage

```
entropy_score(score_matrix)
```

Arguments

score_matrix	Numeric matrix

Value

Numeric vector

Examples

```
mat <- matrix(
  runif(20),
  nrow = 5
)

entropy_score(
  mat
)
```

---

explain_cell *Explain Cell Annotation Confidence*

Description

Provides an interpretable explanation for the confidence score assigned to an individual cell.

Usage

```
explain_cell(object, cell_id)
```

Arguments

object	A Seurat or SingleCellExperiment object.
cell_id	Cell barcode or column name.

Value

A character vector containing the interpretation.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(200, lambda = 5),
  nrow = 20,
  ncol = 10,
  dimnames = list(
    paste0("Gene", seq_len(20)),
```

```
    paste0("Cell", seq_len(10))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$predicted_label <-
  rep(c("T cell", "B cell"), each = 5)

SummarizedExperiment::colData(sce)$marker_score <-
  runif(ncol(sce))

SummarizedExperiment::colData(sce)$neighbor_score <-
  runif(ncol(sce))

SummarizedExperiment::colData(sce)$entropy_norm <-
  runif(ncol(sce))

SummarizedExperiment::colData(sce)$confidence_score <-
  runif(ncol(sce))

explain_cell(
  sce,
  cell_id = "Cell1"
)
```

`explain_confidence` *Explain Confidence Attribution*

Description

Explains why a cell received its confidence score.

Usage

```
explain_confidence(object, cell_id)
```

Arguments

`object`	A Seurat or SingleCellExperiment object.
`cell_id`	Cell barcode.

Value

Character vector describing the factors contributing to the confidence score.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(200, lambda = 5),
  nrow = 20,
  ncol = 10,
  dimnames = list(
    paste0("Gene", seq_len(20)),
    paste0("Cell", seq_len(10))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$predicted_label <-
  rep(c("T cell", "B cell"), each = 5)

SummarizedExperiment::colData(sce)$marker_score <-
  runif(ncol(sce))

SummarizedExperiment::colData(sce)$neighbor_score <-
  runif(ncol(sce))

SummarizedExperiment::colData(sce)$entropy_norm <-
  runif(ncol(sce))

SummarizedExperiment::colData(sce)$doublet_score <-
  runif(ncol(sce))

explain_confidence(
  sce,
  cell_id = "Cell1"
)
```

---

marker_score *Marker Consistency Score*

Description

Uses UCell enrichment scoring for marker evaluation.

Usage

```
marker_score(object, markers, label_column = "predicted_label")
```

Arguments

object	A Seurat or SingleCellExperiment object.
markers	Named list of marker genes.
label_column	Metadata column containing predicted labels.

Value

Numeric vector of marker consistency scores.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(200, lambda = 5),
  nrow = 20,
  ncol = 10,
  dimnames = list(
    paste0("Gene", seq_len(20)),
    paste0("Cell", seq_len(10))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$predicted_label <-
  rep(c("T cell", "B cell"), each = 5)

markers <- list(
  "T cell" = c("Gene1", "Gene2"),
  "B cell" = c("Gene3", "Gene4")
)

scores <- marker_score(
  sce,
  markers,
  label_column = "predicted_label"
)

head(scores)
```

match_labels *Ontology Label Matching*

Description

Matches predicted labels to marker database labels.

Usage

```
match_labels(label, marker_names)
```

Arguments

label	Character label
marker_names	Character vector

Value

Character vector

Examples

```
match_labels(
  "T_cells",
  c(
    "T_cell",
    "B_cell"
  )
)
```

negative_marker_score *Negative Marker Consistency Score*

Description

Uses UCell enrichment scoring to evaluate expression of genes that are inconsistent with a predicted cell identity.

Usage

```
negative_marker_score(
  object,
  negative_markers,
  label_column = "predicted_label"
)
```

Arguments

`object`	A Seurat or SingleCellExperiment object.
`negative_markers`	Named list of negative marker genes.
`label_column`	Metadata column containing predicted labels.

Details

A higher score indicates stronger expression of negative markers and therefore greater evidence against the predicted annotation.

Negative markers provide contradictory molecular evidence for a predicted cell identity. For example, genes characteristic of B cells can be supplied as negative markers for cells predicted to be T cells.

The function returns one UCell score per cell. Scores are not incorporated into the main confidence score by default.

If a predicted label has no corresponding negative marker set, or none of the supplied negative marker genes are present in the object, a score of zero is returned for those cells. A zero score therefore indicates that no measurable negative-marker signal was obtained and should not be interpreted as biological evidence supporting the predicted identity.

Value

Numeric vector of negative marker scores.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(200, lambda = 5),
  nrow = 20,
  ncol = 10,
  dimnames = list(
    paste0("Gene", seq_len(20)),
    paste0("Cell", seq_len(10))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$predicted_label <- rep(
  c("T cell", "B cell"),
  each = 5
)
```

```
negative_markers <- list(
  "T cell" = c("Gene3", "Gene4"),
  "B cell" = c("Gene1", "Gene2")
)

scores <- negative_marker_score(
  sce,
  negative_markers = negative_markers
)

head(scores)
```

`neighbor_score` *Neighbor Agreement Score*

Description

Computes local neighborhood agreement scores.

Usage

```
neighbor_score(
  object,
  reduction = "pca",
  dims = NULL,
  k = 10,
  label_column = "predicted_label"
)
```

Arguments

`object`	A Seurat or SingleCellExperiment object.
`reduction`	Dimensionality reduction to use.
`dims`	Dimensions to include.
`k`	Number of nearest neighbors.
`label_column`	Metadata column containing predicted labels.

Value

Numeric vector of neighborhood agreement scores.

Examples

```
set.seed(123)

counts <- matrix(
  rpois(1000, lambda = 5),
  nrow = 50,
  ncol = 20,
  dimnames = list(
    paste0("Gene", seq_len(50)),
    paste0("Cell", seq_len(20))
  )
)

sce <- SingleCellExperiment::SingleCellExperiment(
  assays = list(
    logcounts = log1p(counts)
  )
)

SummarizedExperiment::colData(sce)$predicted_label <-
  rep(c("T cell", "B cell"), each = 10)

SingleCellExperiment::reducedDim(sce, "PCA") <- prcomp(
  t(counts),
  rank. = 5
)$x

scores <- neighbor_score(sce)

head(scores)
```

Index

∗ **package**
scCertify-package, 2

calibrate_confidence, *3*, 4
cell_certify, *3*, 4
classify_confidence, *3*, 6

de_novo_signatures, 7
discovery_aware, 8
discovery_status, 10

entropy_score, *3*, 11
explain_cell, *3*, 12
explain_confidence, *3*, 13

marker_score, *3*, 14
match_labels, *3*, 16

negative_marker_score, 16
neighbor_score, *3*, 18

scCertify *(*scCertify-package*)*, 2
scCertify-package, 2