

Package ‘scCertify’

September 10, 2026

Title Explainable Confidence Scoring for Single-Cell Annotations

Version 0.99.7

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, UCell

Suggests BiocStyle, SingleR, celldex, ggplot2, knitr, rmarkdown, scDbfFinder, 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

git_url <https://git.bioconductor.org/packages/scCertify>

git_branch devel

git_last_commit 4f082f1

git_last_commit_date 2026-08-21

Repository Bioconductor 3.24

Date/Publication 2026-09-09

Author Jaya Surya Doddetipalli [aut, cre] (ORCID:
<<https://orcid.org/0009-0002-8975-4425>>)

Maintainer Jaya Surya Doddetipalli <jayasuryadoddetipalli@gmail.com>

Contents

scCertify-package	2
calibrate_confidence	4
cell_certify	4
classify_confidence	6
entropy_score	6
explain_cell	7
explain_confidence	8
marker_score	9
match_labels	10
neighbor_score	11
Index	13

scCertify-package	<i>scCertify: Explainable Confidence Scoring for Single-Cell Annotations</i>
-------------------	--

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	<i>Calibrate Confidence Scores</i>
----------------------	------------------------------------

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	<i>Cell Annotation Confidence</i>
--------------	-----------------------------------

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
)
```

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.

Value

The input object with confidence scores and classifications 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")
)

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

sce <- cell_certify(
  sce,
  markers
)

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

```

classify_confidence	<i>Classify Confidence Levels</i>
---------------------	-----------------------------------

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
)
```

entropy_score	<i>Entropy-Based Uncertainty Score</i>
---------------	--

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	<i>Explain Confidence Attribution</i>
--------------------	---------------------------------------

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	<i>Marker Consistency Score</i>
--------------	---------------------------------

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"
  )
)
```

neighbor_score	<i>Neighbor Agreement Score</i>
----------------	---------------------------------

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](#)

entropy_score, [3](#), [6](#)

explain_cell, [3](#), [7](#)

explain_confidence, [3](#), [8](#)

marker_score, [3](#), [9](#)

match_labels, [3](#), [10](#)

neighbor_score, [3](#), [11](#)

scCertify (scCertify-package), [2](#)

scCertify-package, [2](#)