

Package ‘dawnn’

September 1, 2026

Title Differential Abundance with Neural Networks

Version 2.1.1

Description Detects regions of differential abundance in single-cell transcriptomic data by applying a pre-trained neural network model to the labels of each cell's nearest neighbours. Tests for both local and global differential abundance, controlling the false discovery rate with the Benjamini-Yekutieli procedure. The method is described in Hall and Castellano (2023) <[doi:10.1101/2023.05.05.539427](https://doi.org/10.1101/2023.05.05.539427)>.

License GPL (>= 3)

Encoding UTF-8

Imports stats, Seurat, reticulate, keras, utils, withr, tools

Depends R (>= 4.0.0)

Suggests rmarkdown, knitr, testthat (>= 3.1.7), callr, dplyr, pkgload, viridis

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://george-hall-ucl.github.io/dawnn/>,
<https://github.com/george-hall-ucl/dawnn>

BugReports <https://github.com/george-hall-ucl/dawnn/issues>

SystemRequirements Python (>= 3.9) with TensorFlow (>= 2.0), typically installed in a conda environment and selected via the 'tf_conda_env' argument of run_dawnn(). See the package documentation for setup instructions.

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation no

Author George Hall [aut, cre] (ORCID: <<https://orcid.org/0000-0002-4828-0668>>),
Sergi Castellano [aut] (ORCID: <<https://orcid.org/0000-0002-5819-4210>>),
University College London [cph]

Maintainer George Hall <george.hall@ucl.ac.uk>

Repository CRAN
Date/Publication 2026-09-01 08:40:09 UTC

Contents

download_model	2
run_dawnn	3
Index	5

download_model	<i>Download the neural network model used by Dawnn.</i>
----------------	---

Description

download_model() downloads the neural network model used by Dawnn, which is too large to be bundled with the package. This function must be used once before run_dawnn() can be executed. After this, the path to the model can be passed to this function.

Usage

```
download_model(  
  model_url = NULL,  
  model_file_path = NULL,  
  download_method = "auto",  
  download_timeout = 600  
)
```

Arguments

- model_url String url from which to download the model.
- model_file_path String path at which to save the downloaded model.
- download_method String download program to use (e.g. wget, curl etc).
- download_timeout Integer number of seconds before download times out (optional, default = 600).

Value

Message confirming the absolute path to the downloaded model.

Examples

```
## Not run:  
download_model()  
  
## End(Not run)
```

run_dawnn	<i>Identify which cells are in regions of differential abundance using Dawnn.</i>
-----------	---

Description

run_dawnn() is the main function used to run Dawnn. It takes a Seurat dataset and identifies which cells are in regions of differential abundance. Dawnn requires at least 1,001 cells.

Usage

```
run_dawnn(
  cells,
  label_names,
  label_pos_lfc,
  reduced_dim,
  n_dims = 10,
  nn_model = dawnn_default_model_file(),
  recalculate_graph = TRUE,
  alpha = 0.1,
  verbosity = 1,
  seed = 123,
  tf_conda_env = NULL
)
```

Arguments

cells	Seurat object containing the dataset.
label_names	String containing the name of the meta.data slot in cells containing the labels of each cell.
label_pos_lfc	String containing the name of the label associated with positive log-fold change.
reduced_dim	String containing the name of the dimensionality reduction to use.
n_dims	Integer number of dimensions to use if computing graph (optional, default 10).
nn_model	String containing the path to the model's .hdf5 file (optional, defaults to the location used by download_model()).
recalculate_graph	Boolean whether to recalculate the KNN graph. If FALSE, then the one stored in the cells object will be used (optional, default = TRUE).
alpha	Numeric target false discovery rate supplied to the Benjamini–Yekutieli procedure (optional, default 0.1, i.e. 10%).
verbosity	Integer how much output to print. 0: silent; 1: normal output; 2: display messages from predict() function.
seed	Integer random seed (optional, default 123).
tf_conda_env	Conda environment with TensorFlow installed, useful if it is unavailable in the current environment (optional, default NULL).

Value

Seurat dataset cells with added metadata: dawnn_scores (output of Dawnn's model for each cell); dawnn_lfc (estimated log2-fold change in the neighbourhood of each cell); dawnn_p_vals_lda and dawnn_p_vals_gda (p-values associated with the hypothesis tests for whether a cell is in a region of local or global differential abundance, respectively); dawnn_lda_verdict and dawnn_gda_verdict (Boolean output of Dawnn indicating whether it considers a cell to be in a region of local or global differential abundance, respectively).

Examples

```
## Not run:
# Only required options
run_dawnn(
  cells = dataset, label_names = "condition", label_pos_lfc = "Condition_1",
  nn_model = "my_model.h5", reduced_dim = "pca"
)
# All options
run_dawnn(
  cells = dataset, label_names = "condition", label_pos_lfc = "Condition_1",
  nn_model = "my_model.h5", reduced_dim = "pca", n_dims = 50,
  recalculate_graph = FALSE, alpha = 0.2, verbosity = 0, seed = 42,
  tf_conda_env = "my_tensorflow_env"
)

## End(Not run)
```

Index

`download_model`, [2](#)
`download_model()`, [3](#)
`run_dawnn`, [3](#)