DoubletDetection

Release 2.5.2

Dec 21, 2020

Contents:

DoubletDetection is a Python3 package to detect doublets (technical errors) in single-cell RNA-seq count matrices.

CHAPTER 1

Installing DoubletDetection

```
git clone https://github.com/JonathanShor/DoubletDetection.git
cd DoubletDetection
pip3 install .
```

If you are using `pipenv` as your virtual environment, it may struggle installing from the setup.py due to our custom Phenograph requirement. If so, try the following in the cloned repo:

```
pipenv run pip3 install .
```

CHAPTER 2

Running DoubletDetection

To run basic doublet classification:

```
import doubletdetection
clf = doubletdetection.BoostClassifier()
# raw_counts is a cells by genes count matrix
labels = clf.fit(raw_counts).predict()
```

- `raw_counts` is a scRNA-seq count matrix (cells by genes), and is array-like
- `labels` is a 1-dimensional numpy ndarray with the value 1 representing a detected doublet, 0 a singlet, and `np.nan` an ambiguous cell.

The classifier works best when

- There are several cell types present in the data
- It is applied individually to each run in an aggregated count matrix

In `v2.5` we have added a new experimental clustering method (`scanpy`'s Louvain clustering) that is much faster than phenograph. We are still validating results from this new clustering. Please see the notebook below for an example of using this new feature.

See our jupyter notebook for an example on 8k PBMCs from 10x.

CHAPTER 3

Obtaining data

Data can be downloaded from the 10x website.

CHAPTER 4

Citations

bioRxiv submission and journal publication expected in the coming months. Please use the following for now:

Gayoso, Adam, & Shor, Jonathan. (2018, July 17). DoubletDetection (Version v2.4). Zenodo. http://doi.org/10.5281/zenodo.2678042

This project is licensed under the terms of the MIT license.

4.1 DoubletDetection

Doublet detection in single-cell RNA-seq data.

class doubletdetection.doubletdetection.**BoostClassifier**(*boost_rate=0.25*, *n_components=30*, *n_top_var_genes=10000*, *replace=False*, *use_phenograph=True*, *phenograph_parameters={'prune': True}*, *n_iters=25*, *normalizer=None*, *random_state=0*, *verbose=False*, *standard_scaling=False*)

Classifier for doublets in single-cell RNA-seq data.

Parameters

- **boost_rate** (*float, optional*) – Proportion of cell population size to produce as synthetic doublets.
- **n_components** (*int, optional*) – Number of principal components used for clustering.

- **n_top_var_genes** (*int, optional*) – Number of highest variance genes to use; other genes discarded. Will use all genes when zero.
- **replace** (*bool, optional*) – If False, a cell will be selected as a synthetic doublet's parent no more than once.
- **use_phenograph** (*bool, optional*) – Set to False to disable PhenoGraph clustering in exchange for louvain clustering implemented in scanpy. Defaults to True.
- **phenograph_parameters** (*dict, optional*) – Parameter dict to pass directly to PhenoGraph. Note that we change the PhenoGraph 'prune' default to True; you must specifically include 'prune': False here to change this. Only used when use_phenograph is True.
- **n_iters** (*int, optional*) – Number of fit operations from which to collect p-values. Defualt value is 25.
- **normalizer** (*(sp_sparse) -> ndarray*) – Method to normalize raw_counts. Defaults to normalize_counts, included in this package. Note: To use normalize_counts with its pseudocount parameter changed from the default 0.1 value to some positive float *new_var*, use: normalizer=lambda counts: doubletdetection.normalize_counts(counts, pseudocount=new_var)
- **random_state** (*int, optional*) – If provided, passed to PCA and used to seedrandom seed numpy's RNG. NOTE: PhenoGraph does not currently admit a random seed, and so this will not guarantee identical results across runs.
- **verbose** (*bool, optional*) – Set to False to silence all normal operation informational messages. Defaults to True.
- **standard_scaling** (*bool, optional*) – Set to True to enable standard scaling of normalized count matrix prior to PCA. Recommended when not using Phenograph. Defaults to False.

all_log_p_values_
Hypergeometric test natural log p-value per cell for cluster enrichment of synthetic doublets. Shape (n_iters, num_cells).

Type ndarray

all_p_values_
DEPRECATED. Exponentiated all_log_p_values. Due to rounding point errors, use of all_log_p_values recommended. Will be removed in v3.0.

Type ndarray

all_scores_
The fraction of a cell's cluster that is synthetic doublets. Shape (n_iters, num_cells).

Type ndarray

communities_
Cluster ID for corresponding cell. Shape (n_iters, num_cells).

Type ndarray

labels_
0 for singlet, 1 for detected doublet.

Type ndarray, ndims=1

parents_
Parent cells' indexes for each synthetic doublet. A list wrapping the results from each run.

Type list of sequences of int

suggested_score_cutoff_
Cutoff used to classify cells when n_iters == 1 (scores >= cutoff). Not produced when n_iters > 1.

Type float

synth_communities_
Cluster ID for corresponding synthetic doublet. Shape (n_iters, num_cells * boost_rate).

Type sequence of ints

top_var_genes_
Indices of the n_top_var_genes used. Not generated if n_top_var_genes <= 0.

Type ndarray

voting_average_
Fraction of iterations each cell is called a doublet.

Type ndarray

fit (*raw_counts*)
Fits the classifier on raw_counts.

Parameters **raw_counts** (*array-like*) – Count matrix, oriented cells by genes.

Sets: **all_scores_**, **all_p_values_**, **all_log_p_values_**, **communities_**, top_var_genes, parents, synth_communities

Returns The fitted classifier.

predict (*p_thresh=1e-07*, *voter_thresh=0.9*)
Produce doublet calls from fitted classifier

Parameters

- **p_thresh** (*float, optional*) – hypergeometric test p-value threshold that determines per iteration doublet calls
- **voter_thresh** (*float, optional*) – fraction of iterations a cell must be called a doublet

Sets: **labels_** and **voting_average_** if n_iters > 1. **labels_** and **suggested_score_cutoff_** if n_iters == 1.

Returns 0 for singlet, 1 for detected doublet

Return type **labels_** (ndarray, ndims=1)

doubletdetection.doubletdetection.**load_10x_h5** (*file*, *genome*)

Load count matrix in 10x H5 format Adapted from: https://support.10xgenomics.com/single-cell-gene-expression/software/ pipelines/latest/advanced/h5_matrices

Parameters

- **file** (*str*) – Path to H5 file
- **genome** (*str*) – genome, top level h5 group

Returns Raw count matrix. ndarray: Barcodes ndarray: Gene names

Return type ndarray

doubletdetection.doubletdetection.**load_mtx**(*file*)
Load count matrix in mtx format

Parameters **file** (*str*) – Path to mtx file

Returns Raw count matrix.

Return type ndarray

4.2 Plot

doubletdetection.plot.**convergence**(*clf*, *show=False*, *save=None*, *p_thresh=1e-07*, *voter_thresh=0.9*)
Produce a plot showing number of cells called doublet per iter

Parameters

- **clf** (*BoostClassifier object*) – Fitted classifier
- **show** (*bool, optional*) – If True, runs plt.show()
- **save** (*str, optional*) – filename for saved figure, figure not saved by default
- **p_thresh** (*float, optional*) – hypergeometric test p-value threshold that determines per iteration doublet calls
- **voter_thresh** (*float, optional*) – fraction of iterations a cell must be called a doublet

Returns matplotlib figure

doubletdetection.plot.**normalize_counts**(*raw_counts*, *pseudocount=0.1*)
Normalize count array. Default normalizer used by BoostClassifier.

Parameters

- **raw_counts** (*ndarray*) – count data
- **pseudocount** (*float, optional*) – Count to add prior to log transform.

Returns Normalized data.

Return type ndarray

doubletdetection.plot.**threshold**(*clf*, *show=False*, *save=None*, *log10=True*, *log_p_grid=None*, *voter_grid=None*, *v_step=2*, *p_step=5*)
Produce a plot showing number of cells called doublet across various thresholds

Parameters

- **clf** (*BoostClassifier object*) – Fitted classifier
- **show** (*bool, optional*) – If True, runs plt.show()
- **save** (*str, optional*) – If provided, the figure is saved to this filepath.
- **log10** (*bool, optional*) – Use log 10 if true, natural log if false.
- **log_p_grid** (*ndarray, optional*) – log p-value thresholds to use. Defaults to np.arange(-100, -1). log base decided by log10
- **voter_grid** (*ndarray, optional*) – Voting thresholds to use. Defaults to np.arange(0.3, 1.0, 0.05).

- **p_step** (*int, optional*) – number of xlabels to skip in plot
- **v_step** (*int, optional*) – number of ylabels to skip in plot

Returns matplotlib figure

doubletdetection.plot.**umap_plot**(*raw_counts*, *labels*, *n_components=30*, *show=False*, *save=None*, *normalizer=<function normalize_counts>*, *random_state=None*)

Produce a umap plot of the data with doublets in black.

Count matrix is normalized and dimension reduced before plotting.

Parameters

- **raw_counts** (*array-like*) – Count matrix, oriented cells by genes.
- **labels** (*ndarray*) – predicted doublets from predict method
- **n_components** (*int, optional*) – number of PCs to use prior to UMAP
- **show** (*bool, optional*) – If True, runs plt.show()
- **save** (*str, optional*) – filename for saved figure, figure not saved by default
- **normalizer** (*(ndarray) -> ndarray, optional*) – Method to normalize raw_counts. Defaults to normalize_counts, included in this package. Note: To use normalize_counts with its pseudocount parameter changed from the default 0.1 value to some positive float *new_var*, use: normalizer=lambda counts: doubletdetection.normalize_counts(counts, pseudocount=new_var)
- **random_state** (*int, optional*) – If provided, passed to PCA and UMAP

Returns matplotlib figure ndarray: umap reduction

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index