MDA

Lingfei Wang

Oct 06, 2020

CONTENTS:

CHAPTER

ONE

MDA

MDA (Maximal Discriminating Axes) is a dimensional reduction method to contrast different groups of samples. MDA assumes a normally distributed noise and uses classification training error in Linear Discriminant Analyses (LDA) to estimate the similarity between groups of samples. MDA has been used to compare cell groups in single-cell RNA sequencing.

MDA is a Python3 library and provides examples in Jupyter notebooks. You can read more about the MDA method from manuscript (See *References*).

1.1 Installation

MDA can be installed with pip: `python -m pip install git+https://github.com/lingfeiwang/mda.git`.

1.2 Documentation

Documentations are available as html and pdf.

1.3 Examples

You can find an examples with simulated data in the 'examples' folder.

1.4 Contact

Please raise an issue on github .

1.5 References

- TBA

CHAPTER

TWO

ALL API

mda.**draw_triangle**(*prob*, *group*, *group_names*, *colors=array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])*, *figsize=2*, *fs=12*, ***ka*)

Draws triangular plot from LDA probabilities between 3 groups.

Parameters

- **prob** (*numpy.ndarray(shape=(n_cell,3))*) – Probability of each cell being assigned to each group using coordinates on MDA.
- **group** (*numpy.ndarray(shape=(n_cell,))*) – Group ID of each cell. Values must be 0,1,2, matching prob.shape[1].
- **group_names** (*List of str*) – Names of groups.
- **colors** (*numpy.ndarray(shape=(3,3))*) – Colors in [r,g,b] format for each group (as rows). 0<=r,g,b<=1
- **figsize** (*float*) – Figure size (each dimension)
- **fs** (*float*) – Font size
- **ka** (*dict*) – Keyword arguments passed to seaborn.kdeplot

Returns Figure drawn

Return type matplotlib.pyplot.figure

mda.**mda**(*data*, *group*, *datatype='raw'*, ***ka*)

Computes the Maximal Discriminating Axes (MDA) between cell groups in scRNA-seq.

Parameters

- **data** (*numpy.ndarray(shape=(n_gene,n_cell))*) – Gene expression matrix. Can be raw read counts or log(CPM+1).
- **group** (*numpy.ndarray(shape=(n_cell,))*) – Group ID of each cell. Each group must have at least 2 cells. Values must cover 0 to n_group-1.
- **datatype** (*str*) – Type of data.
  - raw: Raw read counts
  - lcpm: Log(CPM+1). Natural log.
- **ka** (*dict*) – Keyword arguments passed to sklearn.discriminant_analysis.LinearDiscriminantAnalysis

Returns

- **loc** (*numpy.ndarray(shape=(n_cell,n_group-1))*) – Coordinates of each cell on the (n_group-1) dimensional MDA.

- **prob** (*numpy.ndarray(shape=(n_cell,n_group))*) – Probability of each cell being assigned to each group using coordinates on MDA.

mda.**sphere2tri**(*d*)

Converts 3-dimensional probabilities to two-dimensional coordinates within unit triangle.

Uses stereographic projection from (0,0,-1), (0,-1,0), (-1,0,0), and then average.

Parameters **d** (*numpy.ndarray(shape=(3,n_cell))*) – Input 3-dimensional probabilities

Returns Two-dimensional coordinates within unit triangle at [0,0],[1,0],[0.5,sqrt(3)/2]

Return type numpy.ndarray(shape=(2,n_cell))

mda.**sphere2tri1**(*d*)

Converts 3-dimensional sqrt probabilities to two-dimensional coordinates within unit triangle.

Uses stereographic projection from (0,0,-1).

Parameters **d** (*numpy.ndarray(shape=(3,n_cell))*) – Input 3-dimensional probabilities

Returns Two-dimensional coordinates within unit triangle at [0,0],[1,0],[0.5,sqrt(3)/2]

Return type numpy.ndarray(shape=(2,n_cell))

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

m

INDEX