

single-cell Panoramic View clustering (PanoView)

Manual

Contents

- 1. Introduction
- 2. Installation
- 3. Tutorial
 - 3.1 Input data
 - 3.2 Generate clusters
 - 3.3 Output results
- 4. Functions
 - 4.1 RunPanoView
 - 4.2 OutputPanoView
 - 4.3 VisCluster
 - 4.4 VisClusterAnno
 - 4.5 VisGeneExp
 - 4.6 RunVGs
 - 4.7 HeatMapVGs
 - 4.8 HeatMapGenes

1. Introduction

2. Installation

PanoView is a python module that uses other common python libraries such as *numpy*, *pandas*, *scikit-learn*, etc. For installing *PanoView* at your local computer, open your command prompt and type the following line

```
pip install git+https://github.com/mhu10/scPanoView.git#egg=scPanoView
```

It will install all the required python libraries for executing *PanoView*. Please make sure that *Git* is probably installed or go to <https://git-scm.com/> for the installation.

To test the PanoView, open the python interpreter or your preferred IDE and type the following

```
from PanoramicView import scPanoView
```

There should not be any error message popping out.

3. Tutorial

3.1 Pollen data

3.1.1 Input expression matrix

The format of the expression matrix should be comma-separated values (csv). The rows are the genes and the columns are the cells. For demonstration, we use expression data (i.e. "PollenRaw.csv") from [Pollen, etc, Nature Biotechnology 2014] as the input matrix. Please make sure that "PollenRaw.csv" is in your python working directory.

```
from PanoramicView import scPanoView  
[1] Pollen = scPanoView.PanoView( "PollenRaw" )
```

You may check the raw expression values of the first three genes from the input matrix. The raw expression values is stored at `Pollen.raw_exp`

```
[2] Pollen.raw_exp.iloc[:3,:3]
```

| Gene_Symbol | Hi_2338_1 | Hi_2338_2 | Hi_2338_3 |
|-------------|-----------|-----------|-----------|
| A1BG | 9.08 | 0.00 | 0.00 |
| A1BG-AS1 | 0.00 | 0.00 | 3.47 |
| A1CF | 0.00 | 0.05 | 0.00 |

3.1.2 Generate clusters

You may use *PanoView*'s default setup to do the clustering. It will display the progress of searching in percentage.

```
[3] Pollen.RunPanoView( )  
14%  
20%  
33%  
:  
74%  
86%  
RunPanoView-Done
```

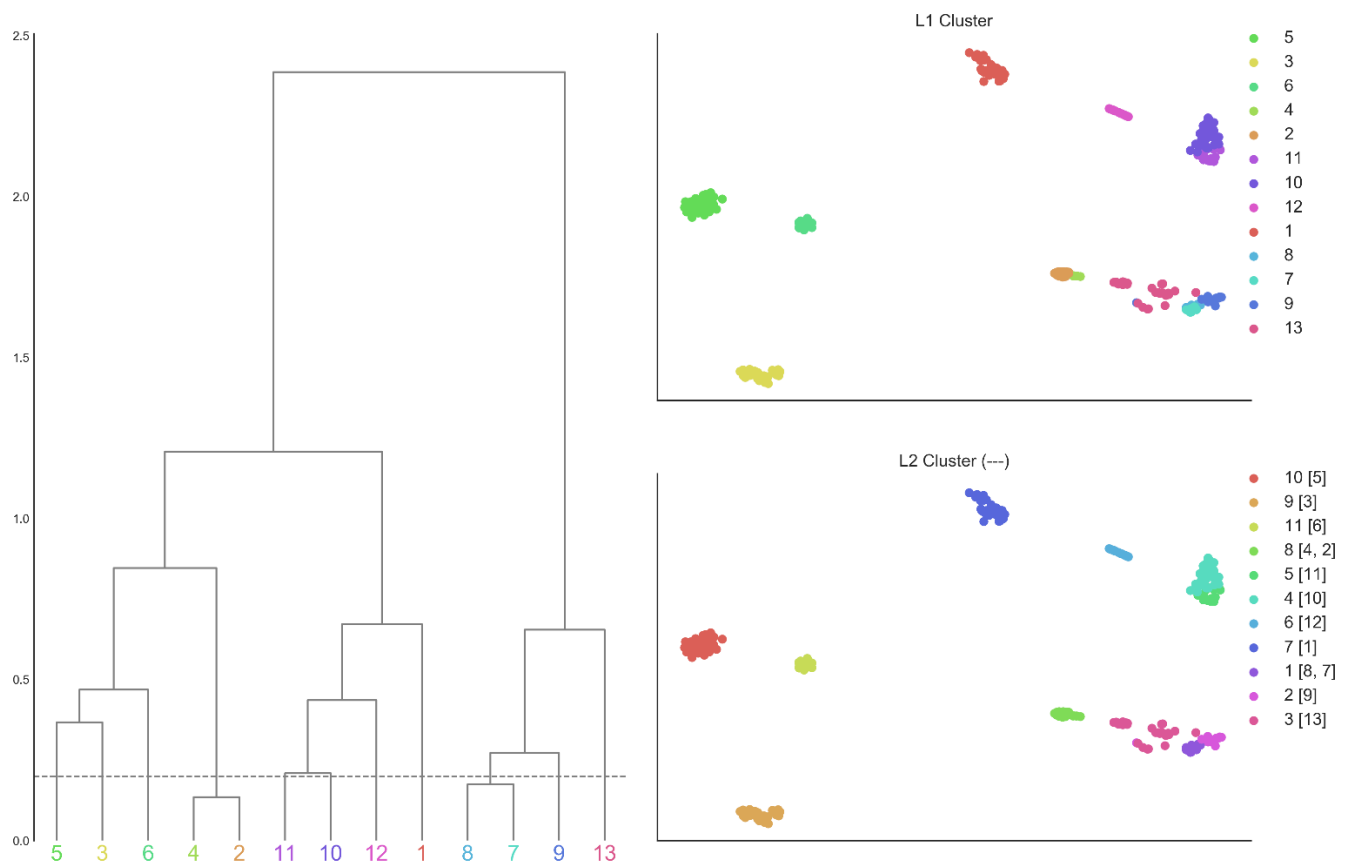
3.1.3 Output results

PanoView will generate two files at your working directory, “Cell_Membership.csv” and “PanoView_output.png”.

```
[4] Pollen.OutputPanoView()
```

Cell_Membership.csv

PanoView_output.png



The clustering result is stored at `Pollen.cell_membership`

```
[5] Pollen.cell_membership.iloc[:5,:]
```

| | Cell_ID | L1Level | L2Level |
|---|-----------|---------|---------|
| 0 | Hi_2338_1 | 6 | 11 |
| 1 | Hi_2338_2 | 6 | 11 |
| 2 | Hi_2338_3 | 6 | 11 |
| 3 | Hi_2338_4 | 6 | 11 |
| 4 | Hi_2338_5 | 6 | 11 |

3.2 Synthetic data

4. **Functions**

4.1 RunPanoView

Arguments

```
RunPanoView(self, Normal=True, Log2=True,  
            GeneLow='default', Zscore='default')
```

4.2 OutputPanoView

Arguments

```
OutputPanoView(self, clust_merge='default',  
               metric_dis='default',  
               fclust_dis='default',  
               init='default', n_PCs='default')
```

4.3 VisCluster

Arguments

```
VisCluster(self, clevel, cnumber)
```

4.4 VisClusterAnno

| Arguments |
|---|
| <code>VisClusterAnno(self, annotation)</code> |

4.5 VisGeneExp

| Arguments |
|--------------------------------------|
| <code>VisGeneExp(self, genes)</code> |

4.6 RunVGs

| Arguments |
|-----------------------------------|
| <code>RunVGs(self, clevel)</code> |

4.7 HeatMapVGs

| Arguments |
|--|
| <code>HeatMapVGs(self, pval, number, fd, clevel, genelist=None)</code> |

4.8 HeatMapGenes

| Arguments |
|---|
| <code>HeatMapGenes(self, clevel, genelist)</code> |

5. Others