

# 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*, *scipy*, *pandas*, *scikit-learn*, etc. For installing *PanoView* at your local computer, open your command prompt and type the following

```
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 (Spyder, PyCharm, Jupyter, etc. ) 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 Initialization and 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.

First, import PanoView module by `from PanoramicView import scPanoView`. Second, chose a job name ("Pollen" in this tutorial) and initialize PanoView by inputting the filename of the expression matrix, "PollenRaw".

(Note: you need to do the initialization whenever starting a new job)

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

You may check the raw expression values of the first three genes/cells from the input matrix. The raw expression value 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

Use `Pollen.RunPanoView( )` to identify single-cell clusters. It will show the progress of algorithm in percentage and display "RunPanoViw-Done" once the searching is finished.

```
[3] Pollen.RunPanoView( )
```

```
14%
```

```
20%
```

```
33%
```

```
:
```

```
74%
```

```
86%
```

```
RunPanoView-Done
```

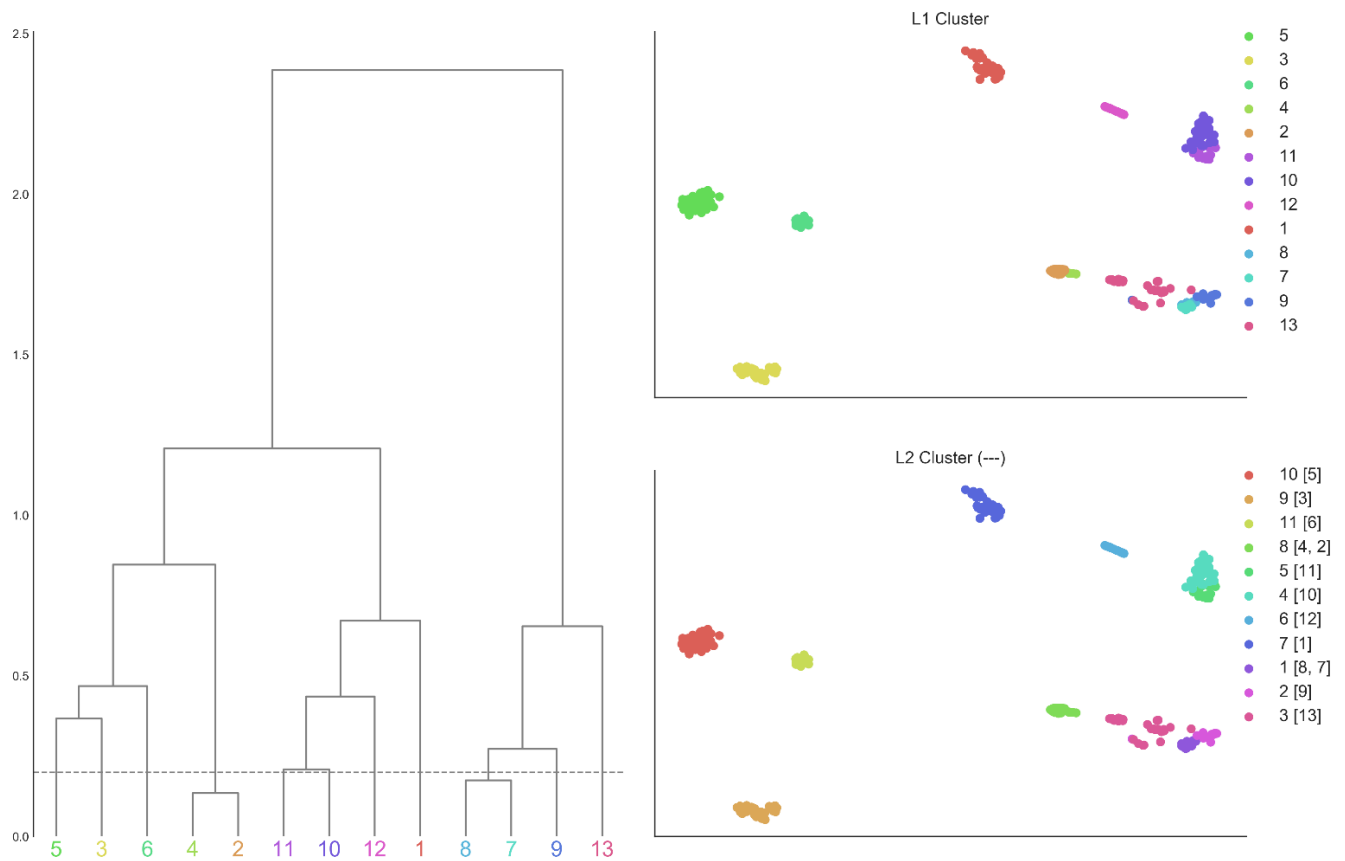
### 3.1.3 Output result

Use `Pollen.OutputPanoView()` to output the result of *PanoView* at your working directory. The result includes a table (“Cell\_Membership.csv”) storing clustering membership of cells and a figure (“PanoView\_output.png”) that consists of one cluster hierarchy and two TSNE plots.

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

```
Cell_Membership.csv
```

```
PanoView_output.png
```



As shown in the figure, *PanoView* identified 13 clusters (Level-1 clusters) and used hierarchical tree to show the similarity of identified clusters. One intuitive parameter in *PanoView* is the height (fclust\_dis = 0.2) of the hierarchical tree that would further merge nearby similar clusters. In this tutorial, the default value of 0.2 produced a total of 11 Level-2 clusters that merged cluster#2 and cluster#4 as cluster 8, and cluster#7 and cluster#8 as cluster 1.

#### 3.1.4 Visualization

#### 3.1.5 Variable Genes

### 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

```
VisClusterAnno(self, annotation)
```

## 4.5 VisGeneExp

### Arguments

```
VisGeneExp(self, genes)
```

## 4.6 RunVGs

### Arguments

```
RunVGs(self, clevel)
```

## 4.7 HeatMapVGs

### Arguments

```
HeatMapVGs(self, pval, number, fd, clevel,  
            genelist=None)
```

## 4.8 HeatMapGenes

Arguments
-----------

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

## 5. Others