**SPmarker**

SPmarker is a machine learning based approach for identification of key markers involved in Arabidopsis root development from single-cell RNA-seqtool to identify marker genes

**Introduction**

An essential facet of the single-cell RNA sequencing (scRNA-seq) studies is to use marker genes that distinguish heterogeneous cell populations and dissect biological functions of each cell. In plants, the scRNA-seq focused mostly on the understood Arabidopsis root system. However, few suitable computational methodologies aid the identification of the novel marker genes in the root system. Here, we introduce SPmarker, a machine learning based method to identify the marker genes via identifying their feature importance from Random Forests (RF) model using the SHapley Additive exPlanations (SHAP) package (SHAP markers).

**Dependence and requirements**

SPmarker is developed in Python with modules and external tools.

Before running this pipeline, a dependency check should be performed first to make sure every dependency is correctly installed.

For information about installing the dependencies, please see below. The version numbers listed below represents the version this pipeline is developed with, and using the newest version is recommended.

**Requirements**

**Python** (v3.0 or more; Developed and tested with version 3.7.1)  
**pandas** (python package; Developed and tested with version 0.24.2)  
**numpy** (python package; Developed and tested with version 1.16.4)  
**itertools** (python package; Developed and tested with version 2.3)  
**sklearn** (python package; Developed and tested with version 0.22.2)  
**shap** (python package; Developed and tested with version 0.35.0)  
**Seurat** (R package; Developed and tested with version 3.0)

**Quick Start**

**Input data**

**1. Gene expression matrix file (.rds)**

**Note:**

1. The input is an object generated from the Seurat package.
2. If users have more than one datasets, they need to preprocess each individual dataset by following the processing step in the Seurat manual, and merge them to be the input data in the SPmarker.
3. The example file can be downloaded under the test\_input\_data folder (merged\_data\_before\_integration\_obj.rds).

**Example Run 1**

Three steps should be run to finish the whole process:

1. Step 1: Prepare data
2. Step 2: Train models
3. Step 3: Identify markers

Users can use **SPmarker\_pipeline.py** to run **Step 1, 2, 3**, but the parameters will be set as default.

* mkdir output\_dir working\_dir
* python SPmarker\_pipeline.py.py \  
  -d working\_dir/ -o output\_dir/ \  
  -SPmarker\_dir SPmarker/ \  
  -merged\_obj merged\_data\_before\_integration\_obj.rds \  
  -R\_p Path/to/Rscript \  
  -kmar\_fl SPmarker/test\_input\_data/known\_marker.txt

**Outputs:**

1. opt\_top\_20\_novel\_known\_marker.txt  
   The file that contains top 20 novel and known makrer genes.
2. opt\_top\_20\_novel\_marker.txt  
   The file that only contains top 20 novel marker genes.
3. opt\_top\_20\_summary\_marker\_composition.txt  
   The file that summarizes composition of the top 20 marker genes.

**Example Run 2**

Users can also run **Step 1, 2, 3** seperately.

**Step 1:**  
**Commands:**

* mkdir 01\_prepare\_data
* cd 01\_prepare\_data
* mkdir output\_dir working\_dir
* python Step1\_prepare\_data.py \  
  -d working\_dir/ -o output\_dir/ \  
  -sup\_dir SPmarker/sub\_dir \  
  -merged\_obj test\_input\_data/merged\_data\_before\_integration\_obj \  
  -R\_p Path/to/Rscript \  
  -ICI yes \  
  -itg yes \  
  -fl\_feat yes \

**Outputs:**

1. ICIn.csv  
   This file is generated from ICI index method that can assign cells with cell types.
2. merged\_data\_after\_integration\_mtx.csv  
   This file will be used in the Step 2.
3. var\_295\_feats.csv  
   This file contains 295 features selected from the Seurat package and that express variablely among all cells. The final marker genes will be selected from these 295 features.

**Step 2:**  
**Commands:**

* mkdir 02\_train\_model
* cd 02\_train\_model
* mkdir output\_dir working\_dir
* python Step2\_train\_models.py \  
  -d working\_dir/ -o output\_dir/ \  
  -ICI\_fl 01\_prepare\_data/output\_dir/ICIn.csv \  
  -exp\_fl 01\_prepare\_data/output\_dir/merged\_data\_after\_integration\_mtx.csv \  
  -feat\_fl 01\_prepare\_data/output\_dir/var\_295\_feats.csv \

**Outputs:**

1. opt\_exp\_indep\_test.csv  
   The independent testing expression dataset that is used in the Step 3.
2. opt\_meta\_indep\_test.csv  
   The independent testing meta dataset that is used in the Step 3.
3. rf\_model.pkl  
   The random forests model that is used in the Step 3.
4. opt\_train\_validation\_cell\_type\_number.txt  
   The summary of cell number in each cell type that is used in the training models.

**Step 3:**  
**Commands:**

* mkdir 03\_identify\_marker
* cd 03\_identify\_marker
* mkdir output\_dir working\_dir
* python Step3\_identify\_marker.py \  
  -d working\_dir/ -o output\_dir/ \  
  -m 02\_train\_model/output\_dir/rf\_model.pkl \  
  -exp\_fl 02\_train\_model/output\_dir/opt\_exp\_indep\_test.csv \  
  -meta\_fl 02\_train\_model/output\_dir/opt\_meta\_indep\_test.csv \  
  -kmar\_fl test\_input\_data/known\_marker.txt

**Outputs:**

1. opt\_top\_20\_novel\_known\_marker.txt  
   The file that contains top 20 novel and known makrer genes.
2. opt\_top\_20\_novel\_marker.txt  
   The file that only contains top 20 novel marker genes.
3. opt\_top\_20\_summary\_marker\_composition.txt  
   The file that summarizes composition of the top 20 marker genes.