scASKapp is the GUI (graphical user interface) version of scASK with the App Designer technique, which can significantly simplify the analysis process and parameters selection. In this guide, we choose *Pollen* dataset (Wang et al., 2017) and *Data_Pollen* dataset (Park and Zhao, 2018) as sample data for demonstrating the full functionality of scASKapp. It is worth mentioning that scASKapp can handle data from different application fields, not just scRNA-seq data. The only requirement is that the input matrix named in_X should keep rows representing samples and columns representing attributes. The standard input matrix and input labels for scASK should have the format shown below.

Pollen Dataset	'A1BG'	'A2M'	'A2ML1'	'A2MP1'	'A4GALT'	'AAAS'	'AACS'	'AACSP1'	'AADAT'	···	Class
Sample1	0.30103	2.0086	0	0	0	0	0	1.30103	0.30103	···	10
Sample2	0	0	0	0	0	0	2.089905	0	1.041393	···	10
Sample3	0	0	0	0	0	0	0	0	1.544068	···	10
Sample4	0	0	0	0	0	2.523746	0	0	2.599883	···	9
Sample5	0	0	1.30103	0	0	0	2.584331	0	2.457882	···	9
Sample6	0	0	1.70757	0	0	2.523746	0.845098	0	1.78533	···	9
Sample7	0	0	0	0	0	0	3.202761	0	2.664642	···	9
Sample8	0	0	0	0	0	1.380211	0	0	1.869232	···	9
Sample9	0	0	0	0	1.041393	0	0.30103	0	0	···	9
Sample10	0	0	0.69897	0	0	2.093422	1	0	0	···	9
Sample11	0	0	0	0	0	0	1.819544	0	1.556303	···	9
Sample12	0	0	0	0	0	1.518514	1.278754	0	0	···	9
Sample13	0	0	0	0	0	0	0	0	0	···	10
Sample14	0	0	0	0	0	0	0	0	0	···	10
Sample15	0	0	0	0	0	1.770852	0	0	0	···	10
Sample16	0	1.380211	0	0	0.60206	1.544068	2.079181	0	0.60206	···	10
Sample17	2.39794	1.875061	0	0	0	1.041393	0	0	0	···	10
Sample18	1.568202	1.113943	0	0	0	2.255273	0	0	0	···	10
Sample19	0	0	0	0	0	2.481443	2.311754	0	1	···	10
Sample20	1.146128	0	0	0	0	0	0	0	2.568202	···	10
···	···	···	···	···	···	···	···	···	···	···	···
Sample249	0	0	0	0	0	0	0	0	0	···	4

true_labs

in_X

Wang et al., 2017 (5 example datasets)

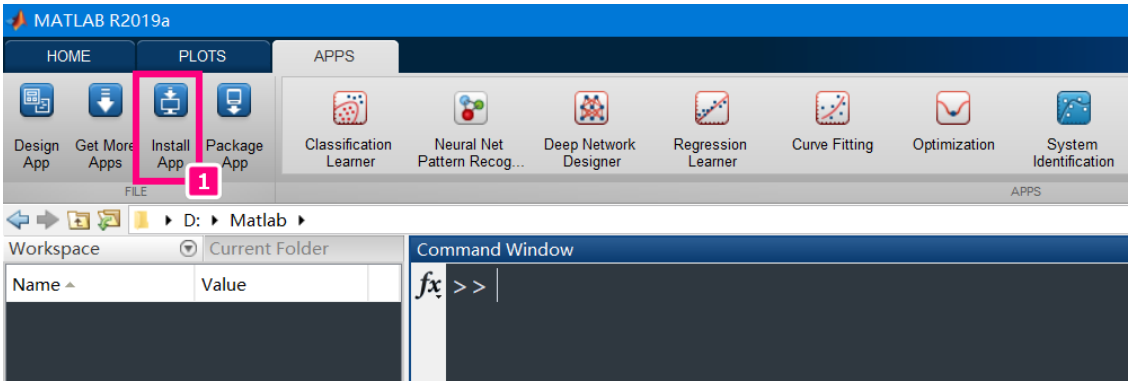
https://github.com/BatzoglouLabSU/SIMLR/tree/SIMLR/MATLAB/data

Park and Zhao, 2018 (9 example datasets)

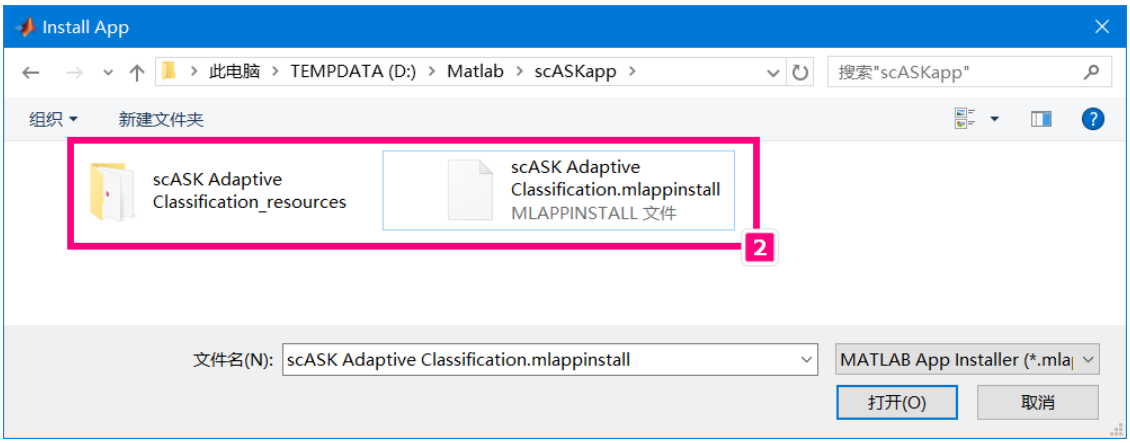
https://github.com/ishspsy/project/tree/master/MPSSC/Data

1. Installation

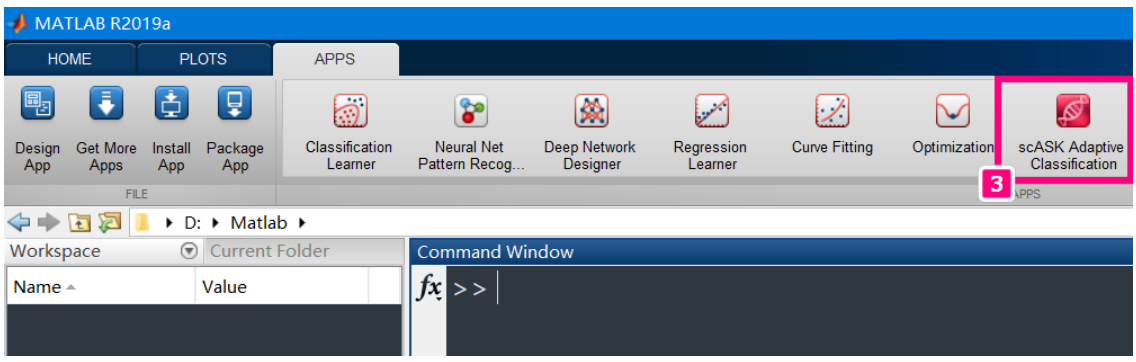
scASKapp 1.0 was developed by using Matlab 2019a, which is also the most suitable running environment for uninterrupted and error-free operating of the software. To install the scASKapp 1.0, you need to follow the installation steps in the order demonstrated below.



Screenshot 1: Click Install App button on the APPS tab.



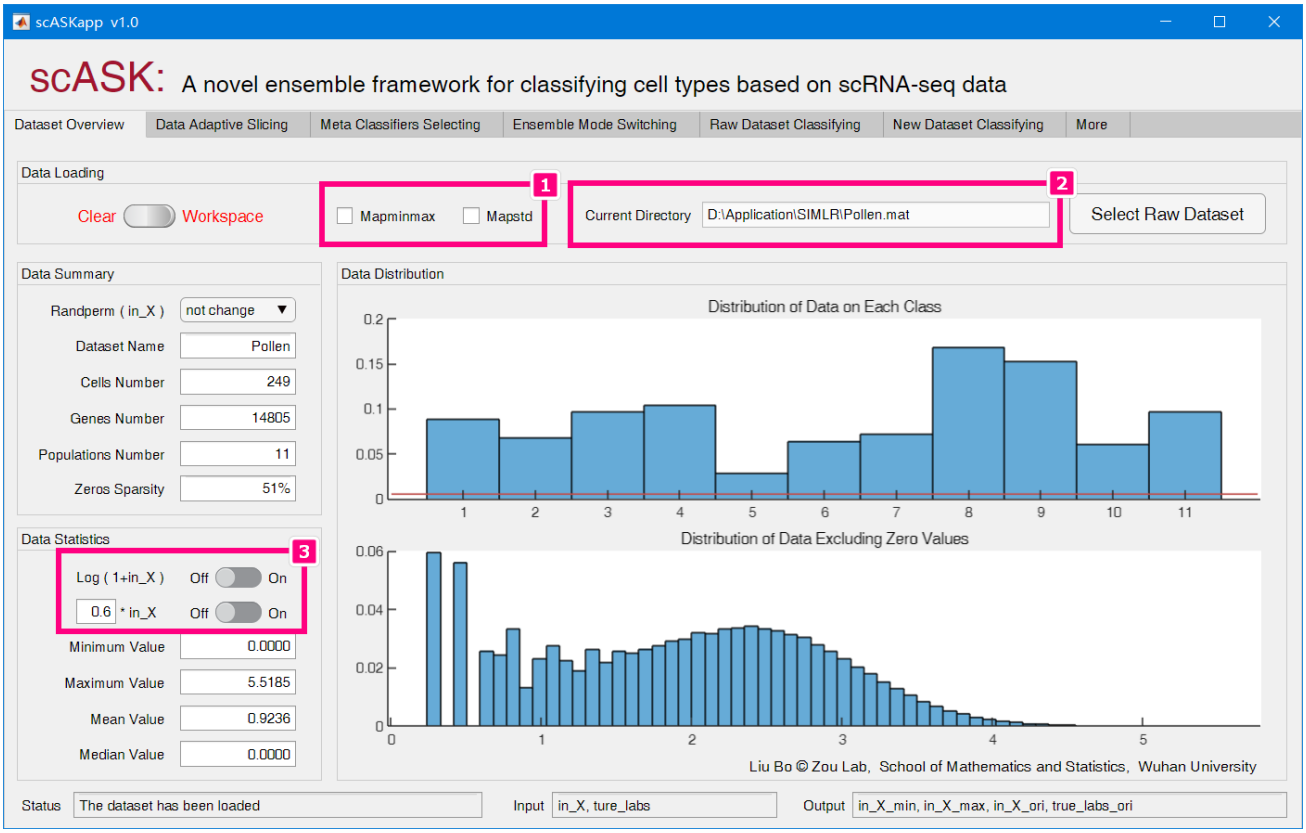
Screenshot 2: Select and open *scASK Adaptive Classification.mlappinstall* file.



Screenshot 3: Then the icon of scASKapp will appear in the toolstrip.

2. Dataset Overview

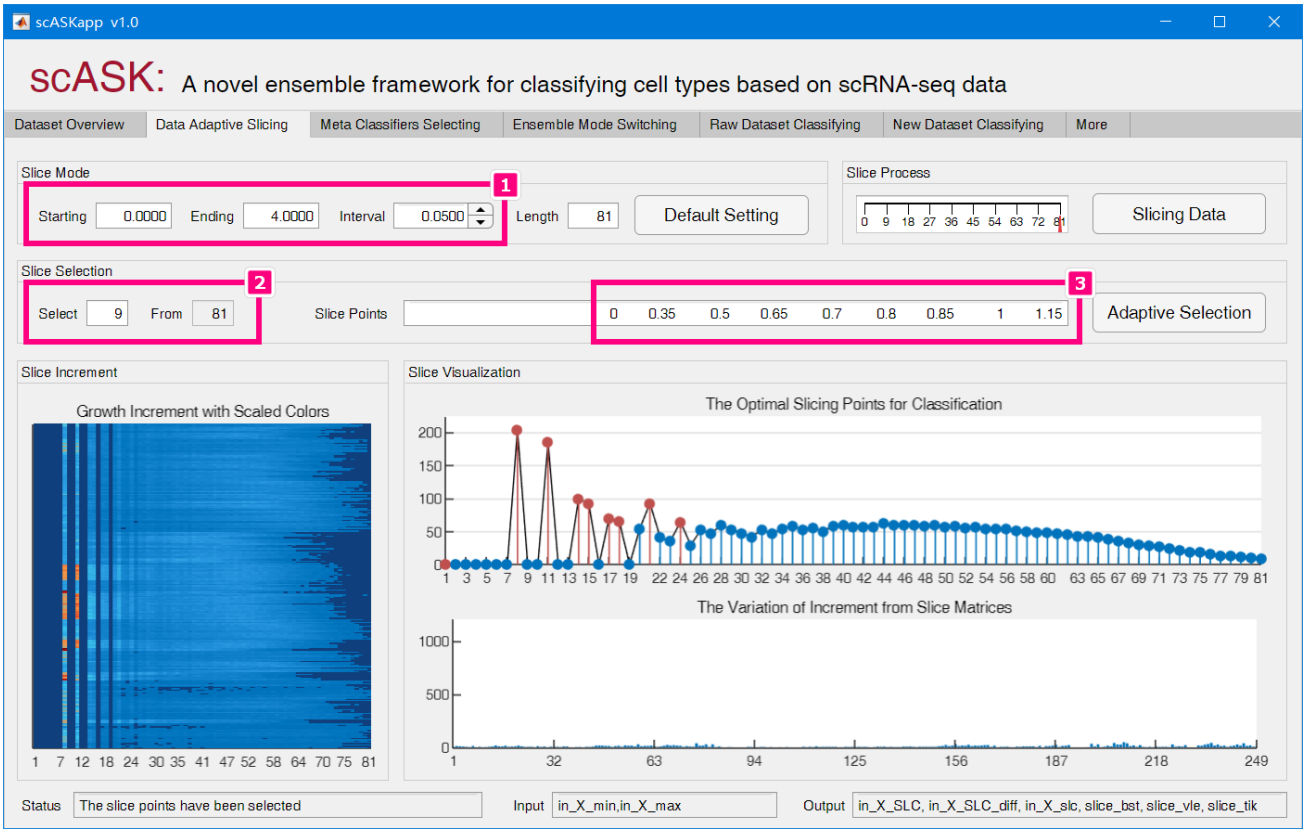
On the Dataset Overview tab of scASKapp 1.0, we should decide whether to do data normalization and feature scaling. This is a key step affecting the capture of classification information during subsequent progresses. In order to eliminate the influence of the difference in the range of values between attributes, the box Mapminmax or Mapstd should be checked. Meanwhile, the switches Log(1+in_X) and □*in_X could compress data to appropriate range for efficient slicing procedure.



Screenshot 4: The process of dataset overview for *Pollen* dataset.

3. Data Adaptive Slicing

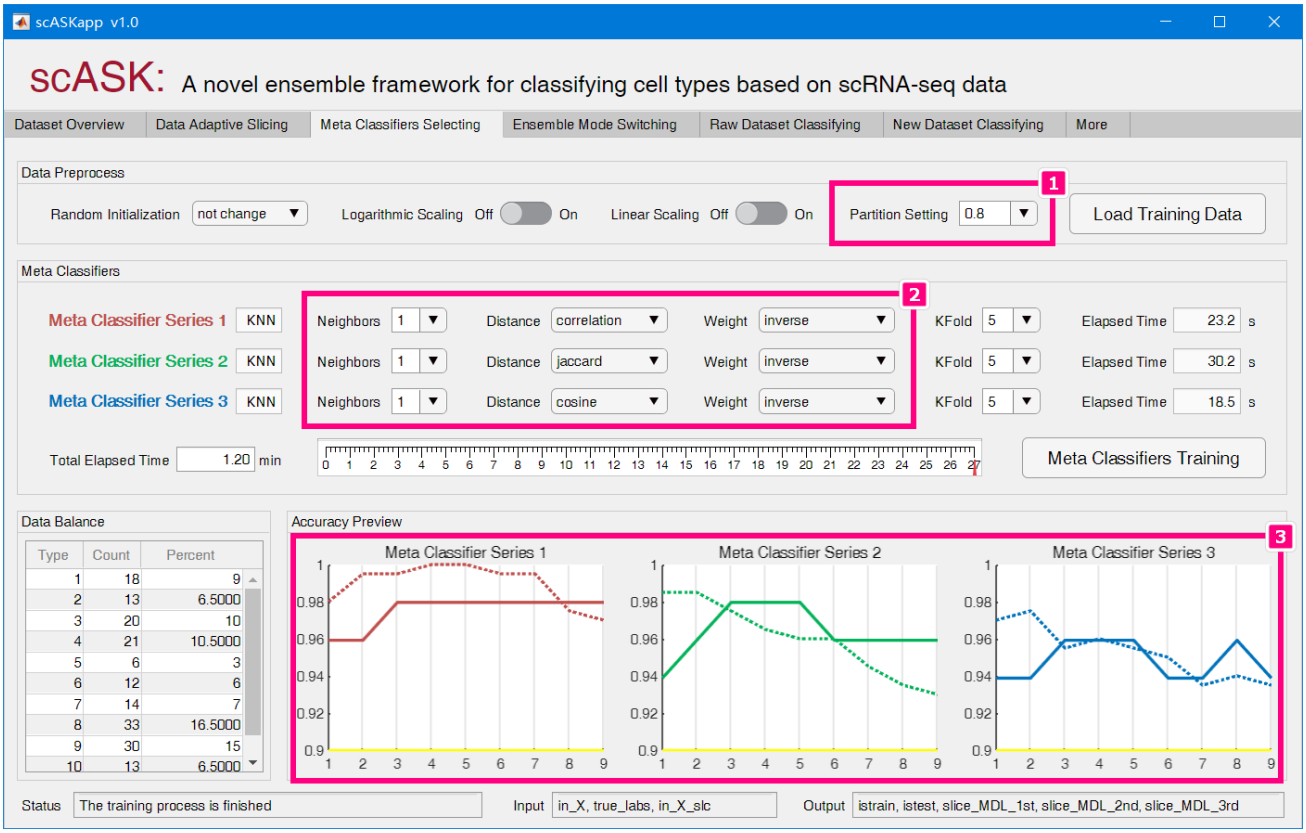
On the Data Adaptive Slicing tab of scASKapp 1.0, we will execute all of data adaptive slicing procedures for sufficiently extracting latent classification information from raw gene expression matrix. The core panel is Slice Mode, where Starting value, Ending value and Interval value can be setup separately. Some default settings are also provided, not optimal, but often works. Note that the slicing procedures will consume a lot of memory to store binary matrices, and those slice points on the peaks are what we really need to focus on.



Screenshot 5: The process of data adaptive slicing for *Pollen* dataset.

4. Meta Classifiers Selecting

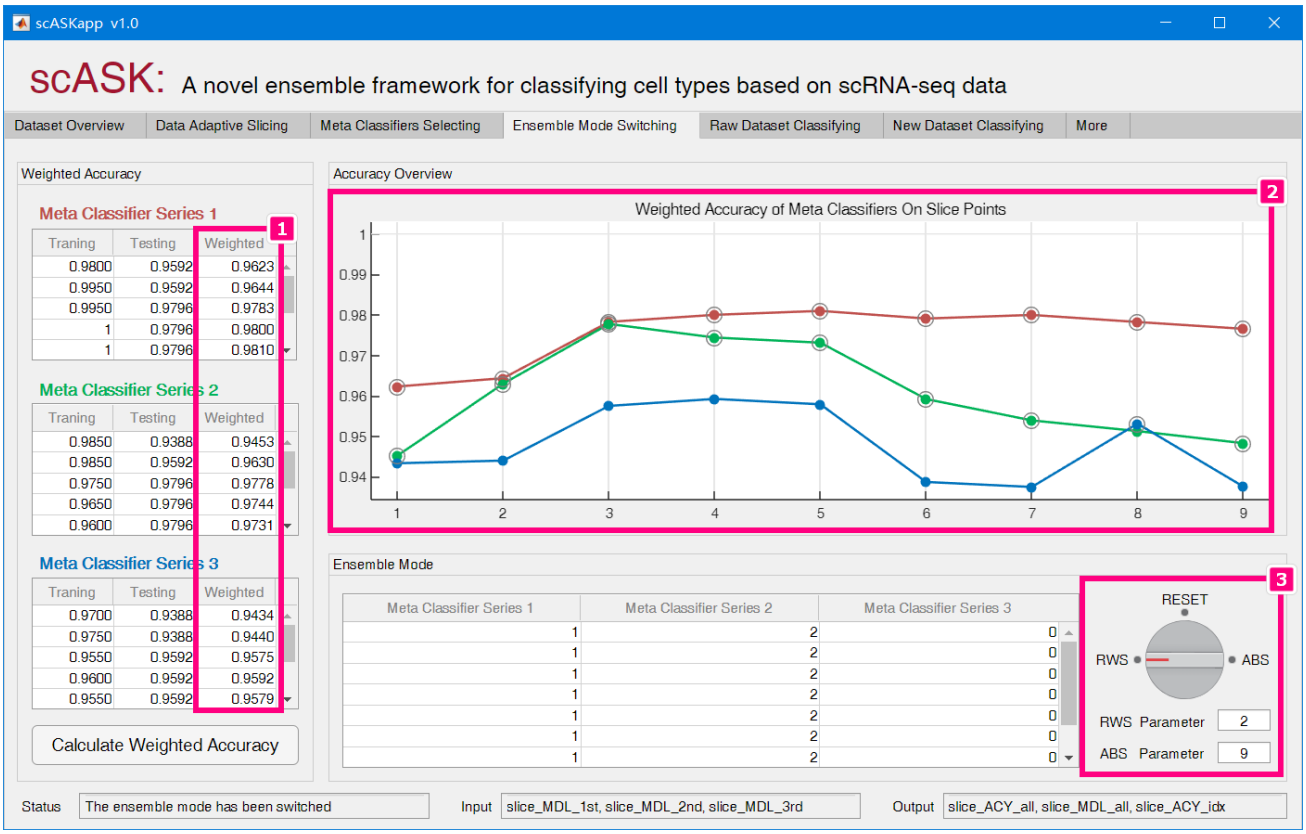
On the Meta Classifiers Selecting tab of scASKapp 1.0, we will adopt kNN as fundamental algorithm, and adopt Pearson's correlation coefficient, Jaccard similarity and Cosine similarity as the default distance measures. Later, three types of meta classifiers could be obtained on every slice points. The core function of this tab is selecting more appropriate combination of parameters for binary slice matrices, which guarantees that all trained meta classifiers are competitive enough for subsequent ensemble classification. The figures of training accuracy (dotted line) and testing accuracy (solid line) on every slice points will be output in real time for parameter optimization.



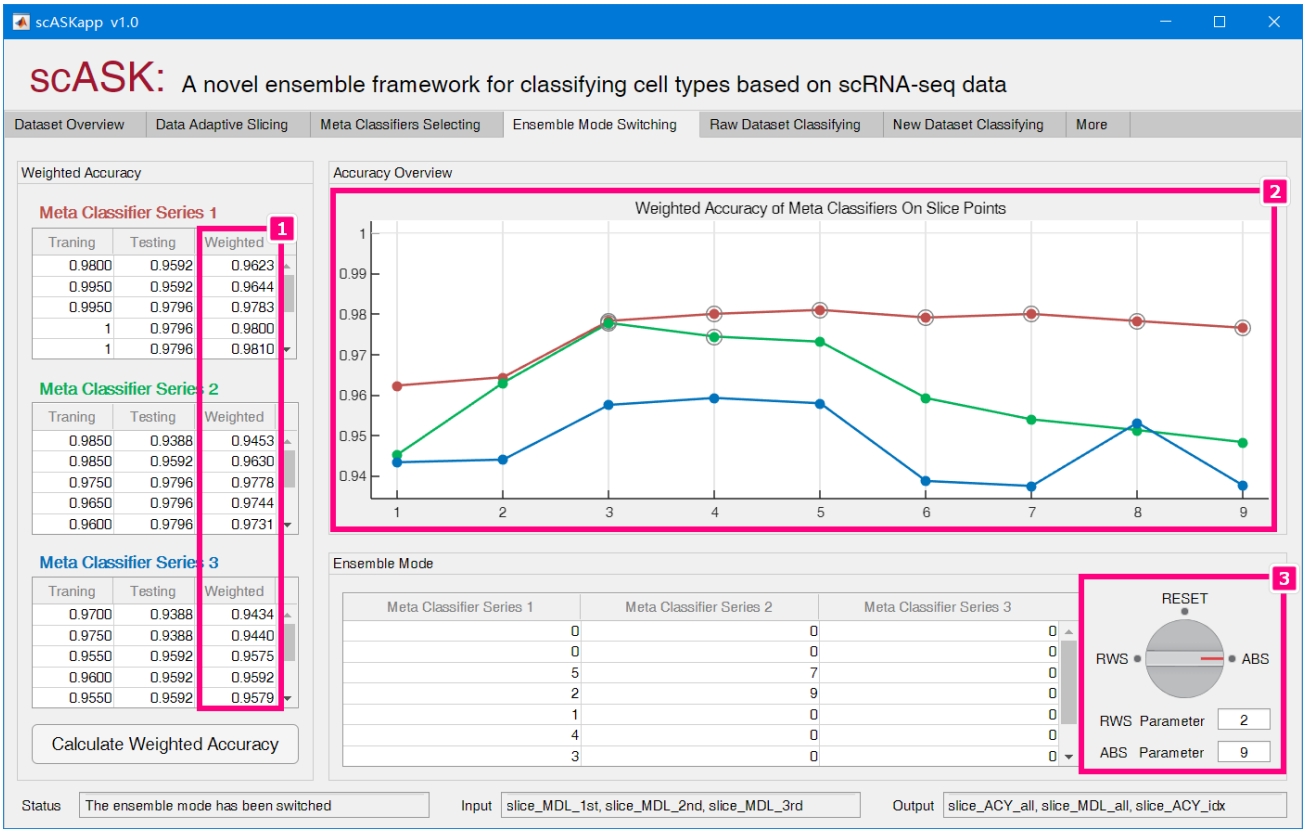
Screenshot 6: The process of meta classifiers selecting for *Pollen* dataset

5. Ensemble Mode Switching

On the Ensemble Mode Switching tab of scASKapp 1.0, we will sort all of meta classifiers with the weighted accuracy, and then integrate them via the switching strategy for final ensemble with two modes: the RWS mode and the ABS mode. Briefly, the RWS mode is a local optimal solution (selecting on each slice point), while the ABS mode is a global optimal solution (selecting on all slice points). For better accuracy and robustness, scASK switches ensemble strategy between two modes manually according to the practice and experience. Massive data experiments approve that scASK running with switching strategy can achieve better prediction for classifying cell types based on scRNA-seq data, which not only helps to confirm the results, but also enhances the reliability of the classification. The core function of this tab is switching meta classifiers for ensemble according to the weighted accuracy.



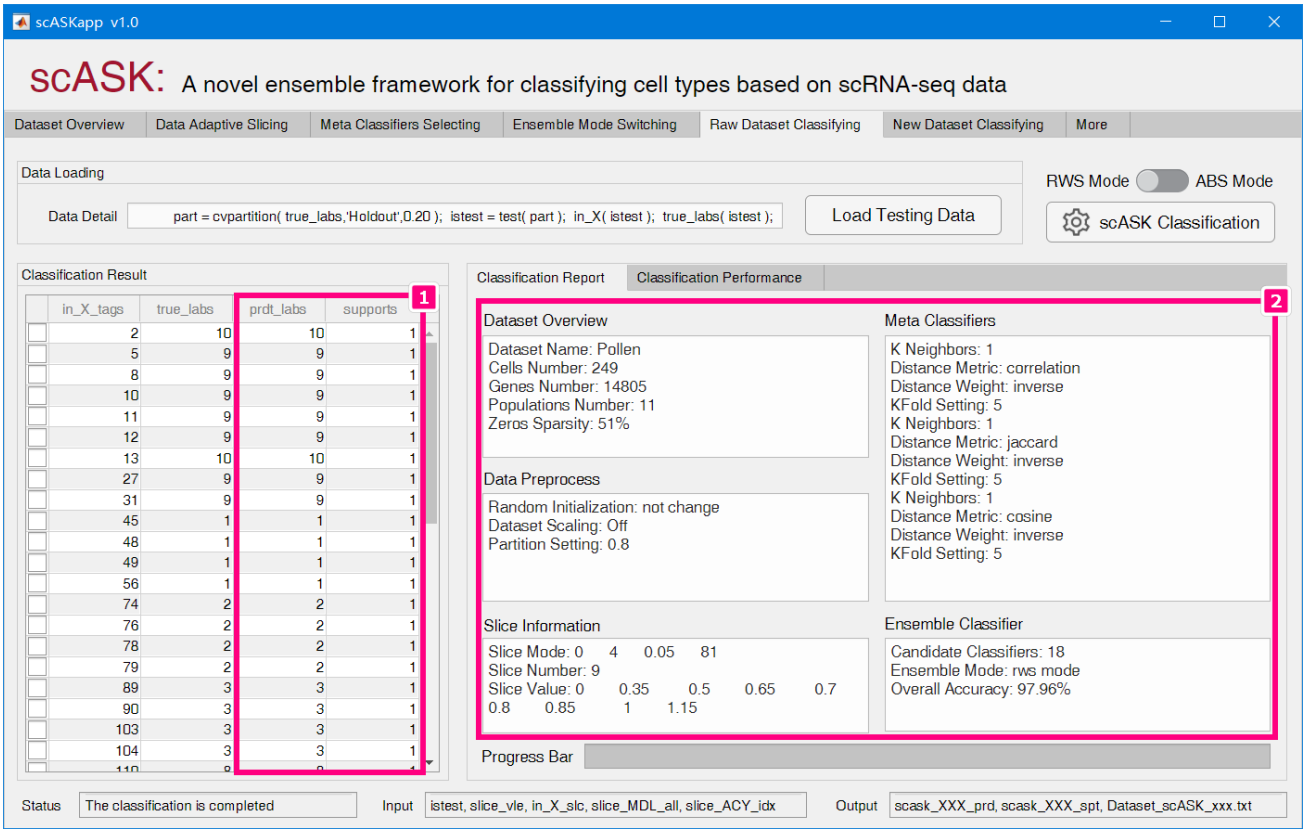
Screenshot 7: The process of ensemble mode switching for *Pollen* dataset (RWS mode)



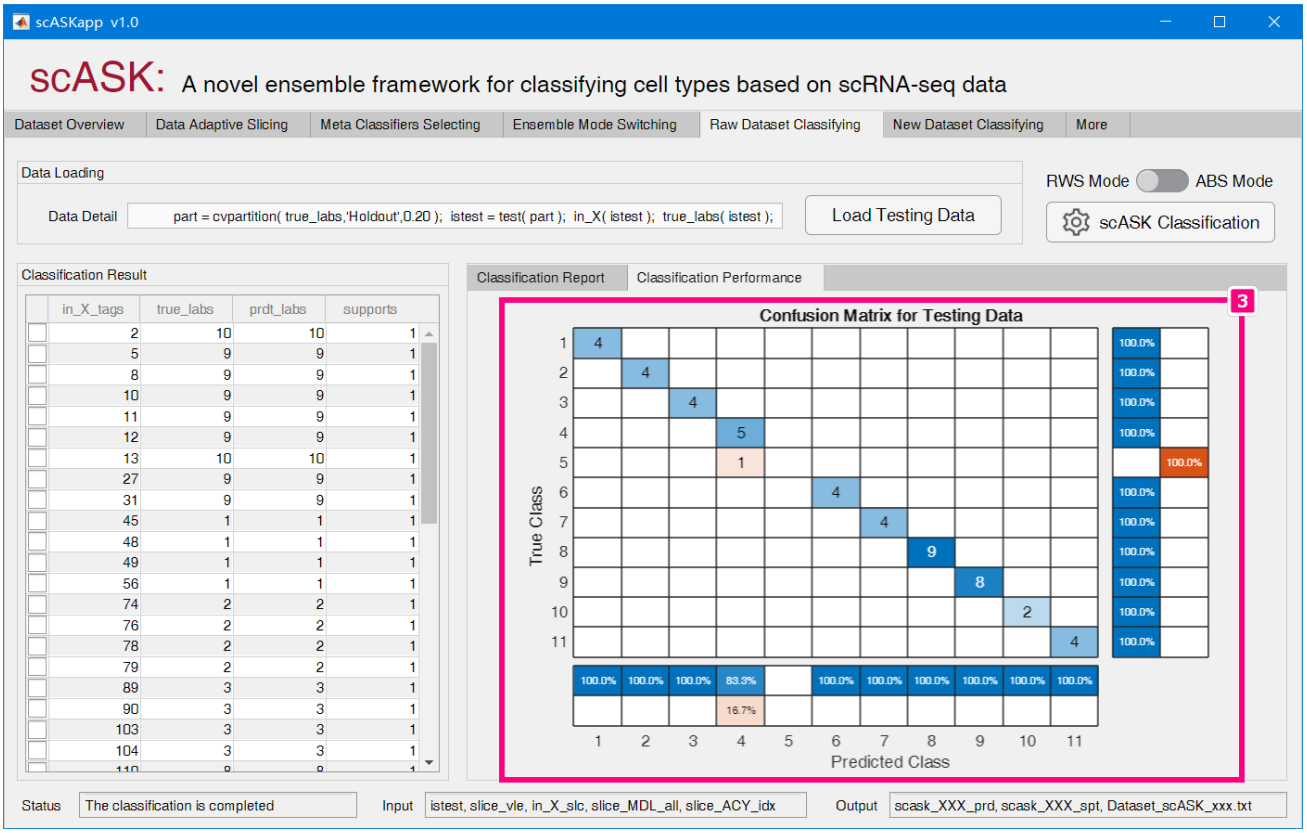
Screenshot 8: The process of ensemble mode switching for *Pollen* dataset (ABS mode)

6. Raw Dataset Classifying

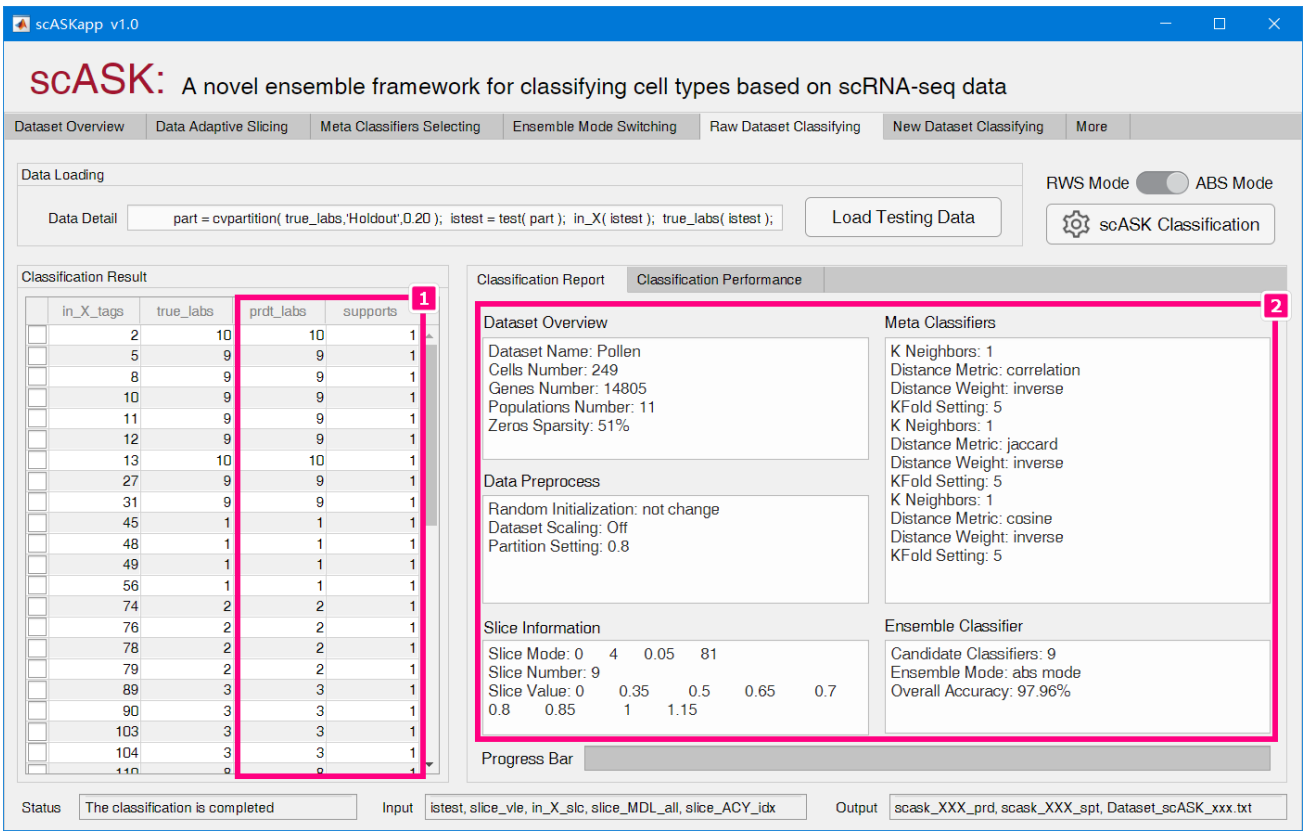
On the Raw Dataset Classifying tab of scASKapp 1.0, we will implement cell type classification using scASK with switching strategy (RWS mode and ABS mode) for *Pollen* dataset. More concretely, the support score of each sample will be computed for evaluating reliability of the final classification result, and the confusion matrix will be computed for evaluating the overall performance of the final ensemble classifier. The core function of this tab is generating classification report and evaluating classification performance.



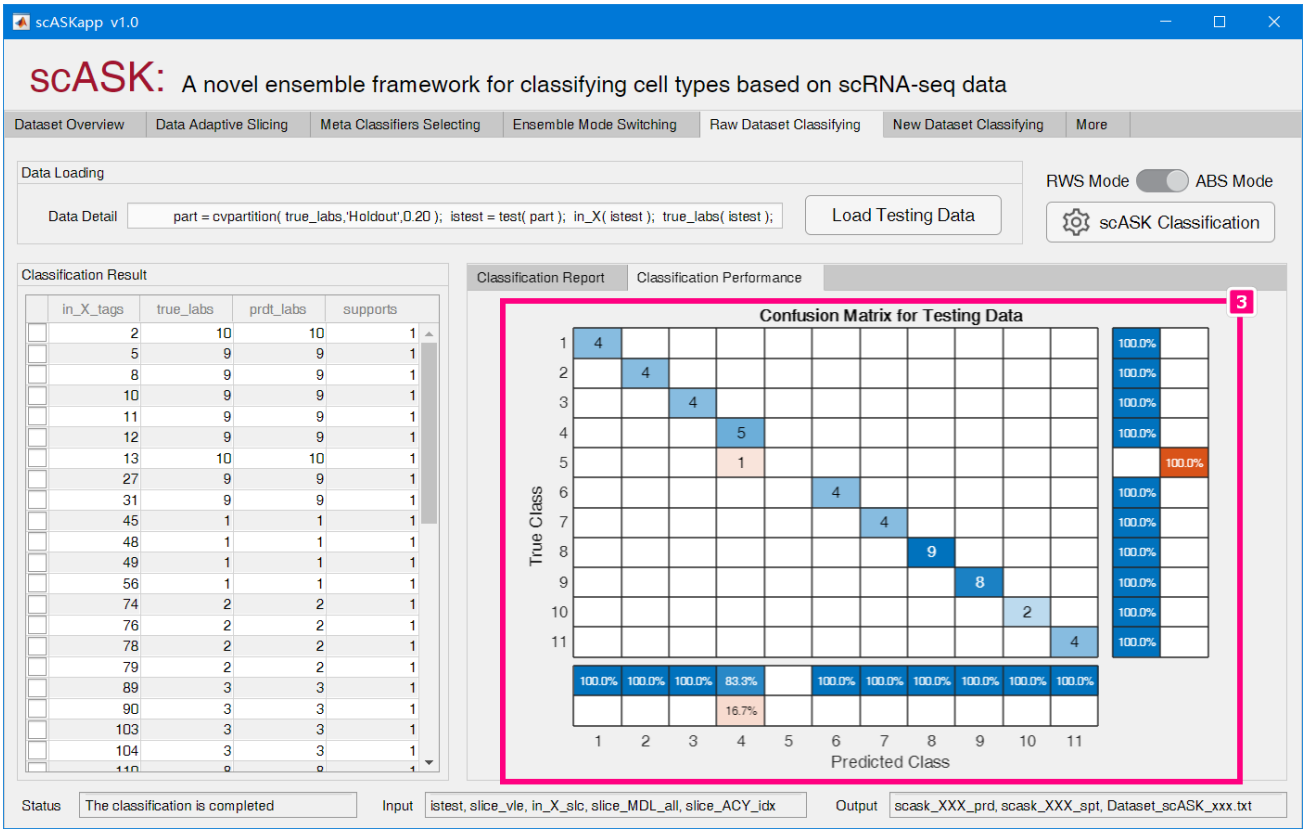
Screenshot 9: The classification report for *Pollen* dataset (RWS mode)



Screenshot 10: The classification performance for *Pollen* dataset (RWS mode)



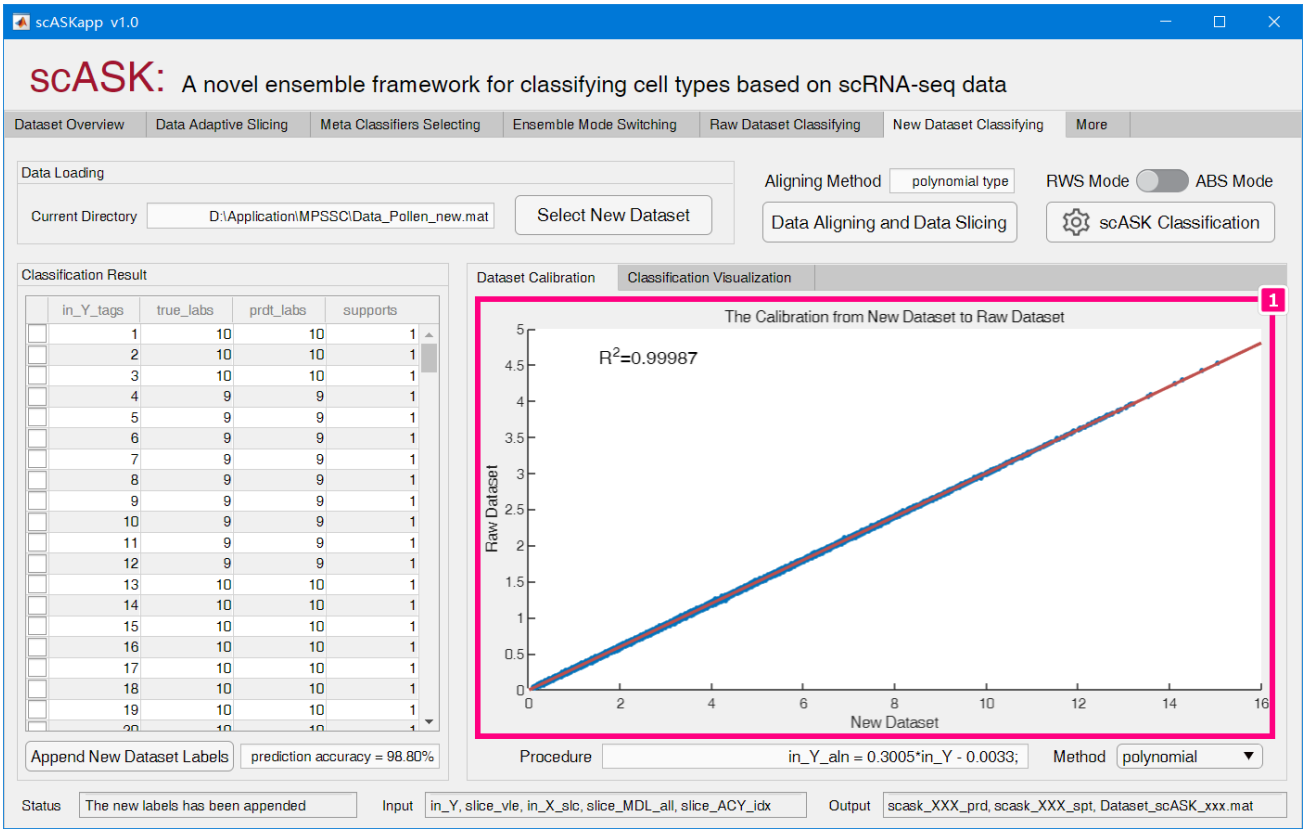
Screenshot 11: The classification report for *Pollen* dataset (ABS mode)



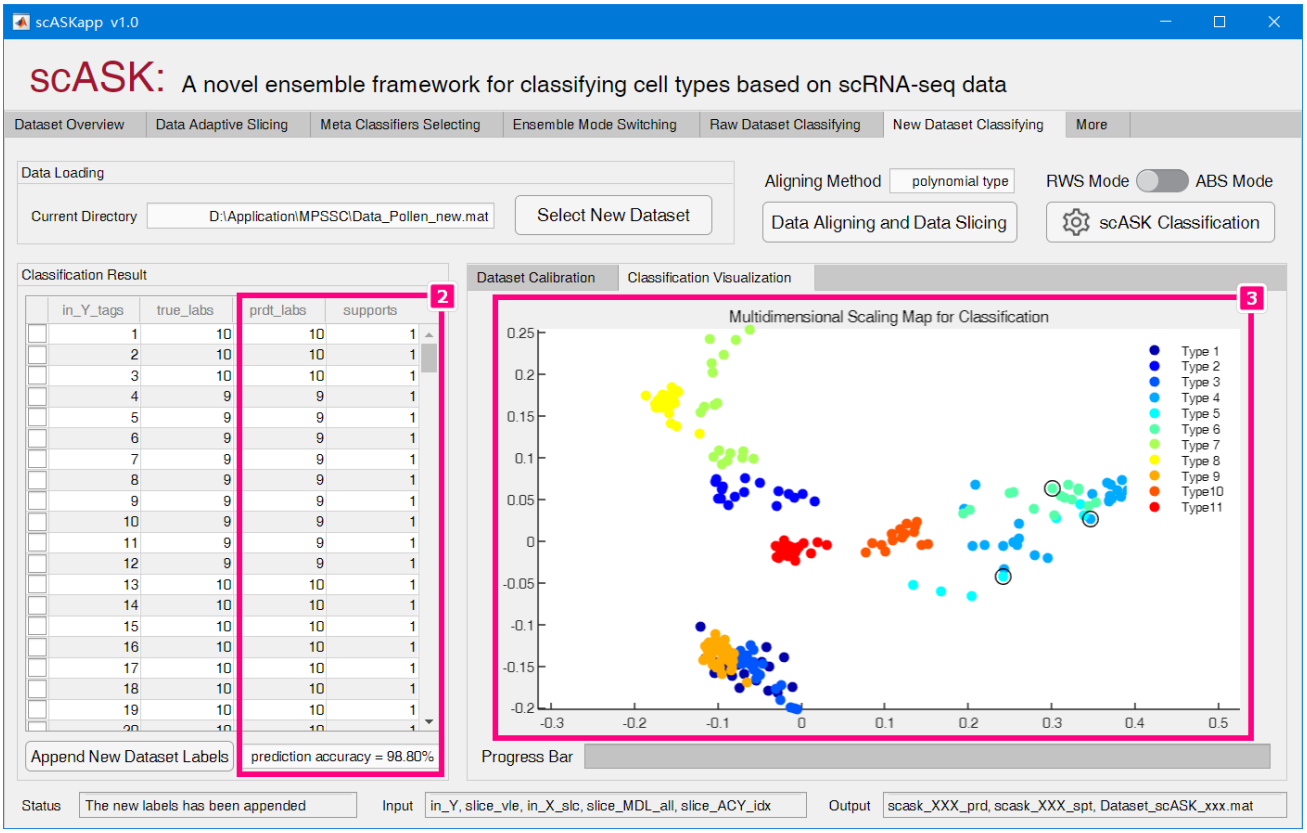
Screenshot 12: The classification performance for *Pollen* dataset (ABS mode)

7. New Dataset Classifying

On the New Dataset Classifying tab of scASKapp 1.0, we will implement cell type classification using scASK with switching strategy for *Data_Pollen* dataset. For the purpose of verification, we split the data and the labels from *Data_Pollen* dataset into two independent datasets as *Data_Pollen_new* and *Data_Pollen_new_labels*. Remarkably, with the aid of data alignment technique, scASK will jump out of comfort zone to deal with really challenging unlabeled dataset. Following similar analytical process, the support score of each sample will be computed for evaluating reliability of the classification result, and the multidimensional scaling map will be plotted for visualizing the classification result. The core function of this tab is the Dataset Calibration which gives scASK the ability to classify types across datasets.



Screenshot 13: The dataset calibration from *Data_Pollen* dataset to *Pollen* dataset



Screenshot 14: The classification result and visualization for *Data_Pollen* (RWS mode)

Screenshot 15: The classification result and visualization for *Data_Pollen* (ABS mode)