New Phytologist Supporting Information

Article title: Identification of new marker genes from plant single-cell RNA-seq data using interpretable machine learning methods

Authors: Haidong Yan, Jiyoung Lee, Qi Song, Qi Li, John Schiefelbein, Bingyu Zhao, Song Li

Article acceptance date: Feb-06-2022

Methods S1. Additional materials and methods for this study.

Explanation of the machine learning methods used in this study

The machine learning methods for selecting marker genes were based on distinct underlying computational models. In this section, we will briefly summarize and explain the mechanisms of each type of model with non-technical language for readers who are not familiar with the technical details of these machine learning models. We also illustrated these methods using **Figure S1C**. Classification of two cell types was used as our example to explain the general principles of these machine learning methods. More details and exact parameters used in our analyses are provided as a separate section titled "Technical details for the machine learning methods used in this study" in this document.

KNN is a baseline machine learning model to assign cell types (**Figure S1C**, KNN with K=4). For each cell, the gene expression similarities between this cell and each individual other cells were calculated based on all genes that were used in the analysis. The K most similar cells were then selected and the cell types of these K nearest neighboring cells were used to assign cell type of the original cell by majority vote. Specifically, for one cell, if majority of the neighboring cells are of type A, then this cell is assigned as type A. Because all genes were used to determine similarity between cell types, we did not use KNN to select marker genes. KNN is fast and simple, which makes it a first choice of machine learning classifier in many cases when computation resource is limited.

PCA is an unsupervised method to group cells (**Figure S1C**, PCA). For all cells, each gene's expression level was treated as a feature. These cells were projected into a multi-dimensional space where first and second PCA dimensions were typically plotted to demonstrate the grouping of the cells. The dimensions of PCA were based on decreasing variations explained by the data where the first dimension had most variation in the data followed the second dimension. The contribution of each gene to each of the PCA dimension can be extracted by a loading factor and higher absolute loading represented higher impact on the PCA dimension. Therefore, the loadings were used to rank genes and select marker genes.

SVM is based on an optimization method that can find best-separating-hyperplane between different cell clusters (**Figure S1C**, SVM). In SVM, each gene was represented by one dimension. If we had only two genes in all cell types, we could generate a x-y scatter plot with each dot represents one cell and x axis is the expression of gene 1 and y axis is the expression of gene 2. The SVM method, when applied in this hypothetical situation of 2 genes, is essentially a method to find a line that best separate two cell types in this two-dimensional space. In single cell data, we typically have a few thousands of dimensions and the separating line between two cell types becomes a "hyperplane". The weight of each gene explained how much each gene is contributing to the separating hyperplane and genes with higher weights would be selected as marker genes. This way of classification also partly explains why SVM markers do not have good correlations because these genes are important in determining the boundaries between clusters of cells. The highly correlated genes with specific cell types are more similar to the expression in the centers of each cell clusters.

RF is an ensemble, tree-based methods for classification (**Figure S1C**, RF). The basic component of a RF model is a decision tree. To decide whether a cell is type A or type B, a decision tree would evaluate every single gene to decide a threshold in the expression level for best separating cells into these two classes. For example, in figure S1C, gene G1 was used to make the first decision. After the first step, the cells were split into two clusters based on the single gene expression and associated threshold that best separate two cell types. Each sub cluster was then divided based on the second-best gene (G2). The decision tree could grow to a very large tree with many genes used to make this decision until a stopping criterion was met. RF is a method based on a large number of decision trees and trained on bootstrapped input data. The consensus of all the decision trees was used as a trained model to make predictions. **SHAP** is one of the latest approaches to select features in decision trees that help to improve the interpretability of the tree. The idea of SHAP is, for every cell, the method would calculate how much each gene contributes to the prediction of the cell type. The SHAP value was calculated by summing up the loss of prediction power if the marker gene were excluded from the model and a permutation of all possible combinations of other genes used in the prediction. This is challenging to evaluate explicitly but a computational approximation was applied in our manuscript based on a published Python package (Pedregosa et al., 2011).

BNN is a baseline neural network where three fully connected neural network layers were used (**Figure S1C**, BNN). At each neuron, gene expression from all genes were used as inputs, and an output of this neuron was calculated based on linear regression followed by an activation function (ReLU in our model). At the first layer of the network, 586 neurons were used, thus the gene expression from >20,000 genes were converted into 586 neurons at the first layer. These were converted into a second layer of the neural network by a similar process and then converted into a third layer of the neural network. The final output layer will have two neurons if there were two cell types or multiple neurons based on the number of cell types to be predicted. This neural network model was "trained" using labeled cell types from input data and the weights on each neuron were determined by an algorithm called back propagation. In this training process, the changes in the "goodness of fit" to the labeled data were used to update the weights on the neurons at each layers sequentially in a reversed order.

Triplet NN and **Contrastive NN** are two more complexed neural network architectures as compared to **BNN** (**Figure S1C**, TRINN and CTNN). In BNN, the gene expression from each cell was used to decide whether one cell is in type A or type B. The TRINN and CTNN did not directly assign cell types. In contrast, these two networks were used to learn the similarity between different cells. This is considered as manifold learning in general. The advantage of manifold learning as compared to the BNN was that manifold learning can help to identify rare cell types because the model was trained to learn "distance", not cell identity. For rare cell types, it is challenging to find enough training data to train classifiers to learn the signatures of different cells. In CTNN, the models were trained by input data where cells from the same cell types would have higher weight if they were predicted to be more similar than cells from different cell types. In TRINN, the models were trained such that each training were evaluated using three cells, two from the same cell type and one from a different cell type. The model parameters were optimized such that cells from same cell types should be classified as similar and simultaneously, cells from different cell types had to be distant from the two cells of the same cell type. In contrast to CTNN, TRINN encourages the model to distinguish different cells while maintaining similarity between similar cells.

Technical details for the machine learning methods used in this study

KNN. KNN is a commonly used simple classifier that does not have explicit training process. KNN first computed a distance between the new input vector and every feature vector in the training dataset. Then the top K nearest neighbors were used for new prediction. In the last step, class label of the new input vector was determined by majority vote among the K nearest neighbors. The hyperparameter for KNN is K, the number of top nearest neighbors. K was set to be 50 in our analysis. We also set weights as 'uniform' that means all points in each neighborhood were weighted equally and set p to be 2 that means Euclidean distance. All other hyperparameters were set as default.

RF. RF is an ensemble tree-based machine learning approach. For each decision tree, a subset of training examples was randomly sampled as inputs and a subset of features were randomly sampled to split each tree node. The final class label was determined by majority vote. Number of trees (N) for RF was set as 50 in our analysis.

SVM. SVM is a machine learning classifier that maximizes the margin between different classes in a high dimensional space transformed by a kernel function. Depending on the kernel function, SVM can be a linear classifier (linear kernel) or a non-linear classifier (e.g., Gaussian kernel). To be able to extract interpretable feature weights, linear kernel was used in our analysis to train SVM classifier.

Baseline NN. Baseline NN refers to a basic type of neural network that uses densely connected layer as input layer and hidden layers. The output layer has number of neurons equal to number of cell types (ten cell types). Architecture of the base NN is demonstrated in **Figure S23B**. Briefly, input layer has number of neurons equal to number of genes used for classification and three hidden layers were used, of which each has 586, 256, and 100 neurons. The last layer is an output layer to which a softmax is applied to ensure output scores are summed to 1.

Triplet NN. Triplet NN is the implementation of Siamese neural network with triplet loss function. The use of triplet loss function was discussed in a published study (Alavi et al., 2018). Briefly, Siamese DNN consists of two subnetworks which had identical architecture and weights. The two neural networks connected to the same distance layer which computed a vector of distance between the last two hidden layers in the two subnetworks. The last two hidden layers were lower dimensional embeddings of original feature vectors. Architecture of Siamese NN is demonstrated in **Figure S23.** In this work, number of neurons in input layer was equal to number of genes used

for classification (29,929). Numbers of neurons used in three hidden layers were 586, 256, and 100. In training dataset, each scRNA-seq expression profile was an "anchor" that can be paired with positive example and negative example. Positive examples were those labeled with the same cell type with anchor and negative examples were those with different cell type. For each anchor, it would be paired with a positive example and a negative example, which formed a group of triplets. Then for each group of triplets, anchor-positive and anchor-negative pairs would be respectively fed into triplet NN. Based on the discussion in (Schroff et al., 2015) and (Alavi et al., 2018), the loss function of triplet NN can be written as:

$$L(D)max\left\{,\left(\sum_{i=1}^{T}\left(D_{a,p}^{i}\right)^{2}-\left(D_{a,n}^{i}\right)^{2}+m\right)\right\}$$

Where T is the number of groups of triplets. $D_{a,p}^{i}$ is the Euclidean distance between anchor and positive samples and $D_{a,n}^{i}$ is the Euclidean distance between anchor and negative samples. m is a hyperparameter that represents the margin between $\left(D_{a,p}^{i}\right)^{2}$ and $\left(D_{a,n}^{i}\right)^{2}$.

To ensure that triplet NN can be effectively trained, the groups of triplets need to include anchor-positive pairs with large distances and anchor-negative pairs with small distances. These are the hard training examples that enforce the model to learn effectively. As discussed in Alavi's study (2018), batch hard loss function was used to generate hard training examples. In each iteration of optimization, *M* cell types which had *K* cells in each were sampled to generate a mini-batch. In this mini-batch, losses of hard training examples were selected and summed up as final loss value for the mini-batch. A slight modification of batch hard loss function was made in this study to include more training samples in each mini-batch. Instead of using one pair of hardest anchor-positive and anchor-negative respectively for each anchor, top *k* pairs of hardest pairs were selected for each anchor. The batch hard loss function therefore can be written as:

$$L'(D)=\left\{0,\sum_{i=1}^{M}\sum_{j=1}^{K}\left[topmax(k,P_{j}^{i})-\ topmin(k,N_{j}^{i})+m\right]\right\}$$

Where P_{j}^{i} is the set of distances between *j*th cell from *i*th cell type and all other cells in *i*th cell type (anchor-positive pairs) and N_{j}^{i} is the set of distances between *j*th cell from *i*th cell type and all other cells not from *i*th cell type (anchor-negative pairs). $topmax(k,P_{j}^{i})$ selects the top k pairs with largest distances in P_{j}^{i} and sums the selected distances. $topmin(k,N_{j}^{i})$ selects the top k pairs with smallest distances in N_{j}^{i} and sums the selected distances. This gives k pairs of anchor-positive

sample pairs and k pairs of anchor-negative sample pairs for each anchor. In our analysis k was set as 10.

Contrastive NN. Contrastive NN is an implementation of Siamese neural network with contrastive loss functio (Alavi et al., 2018). In our work, contrastive NN was constructed using the same neural network architecture as triplet NN (**Figure S15**). The difference here was that contrastive NN uses paired samples which pair the cell assigned with same/different cell types. The idea was to penalize large distances between samples of same cell type and small distances between samples of different cell types. The loss function of Contrastive NN can be written as:

$$L(Y, D) = \sum_{i=1}^{P} \left(Y^i\right) \frac{1}{2} (D)^2 + \left(1 - Y^i\right) \frac{1}{2} (\{0, m - D\})^2$$

Where P represents number of pairs of training samples. $Y^i = 1$ if two samples in the ith pair are assigned with same cell type and $Y^i = 0$ if not. D is the Euclidean distance between the two samples in each pair, computed using the last hidden layers of the two sub-networks. m is a hyperparameter that represents the margin between two samples assigned with different cell types, usually set to 1.

Model evaluation

For the KNN, SVM, RF and baseline NN, the sub-training datasets were used to train the models that can directly predict cell type label. The trained models were then used to predict cell type labels for the independent testing datasets. For triplet NN, and contrastive NN, the sub-training datasets were used to trained models that predict neural embeddings of the original feature vectors of in training dataset. For each new input vector from testing dataset, the trained models were first used to predict a neural embedding and this embedding was compared to all neural embeddings of the training dataset. The final cell type label was determined by majority vote of m nearest neighbors. Here we set $m = 50$.

To further evaluate the performance of these seven models, we calculated parameters such as numbers of true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The sensitivity (SE), accuracy (AC), specificity (SP), precision (PR), geometric mean (GM) of SE and SP, and Matthews Correlation Coefficient (MCC) were used to evaluate these models. SE, SP, AC, PR, GM, MCC, and F1 were defined as follows:

SE = TP / (TP + FN);

SP = TN / (FP + TN);

PR = TP / (TP + FP);

GM = $\sqrt{SE \times SP}$;

AC = (TP + TN) / (TP + TN + FP + FN);

MCC = $\frac{(TP \times TN - FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

F1 = 2 × (PR × SE / (PR + SE))

Tukey's honestly significant difference test was used as a conservative statistical test to find significant differences in all pairwise comparisons and to control for family wise error rate (Abdi and Williams, 2010). The Mean average precision (MAP) was also used as an evaluation metric for all classification approaches. The MAP works on ranked lists (e.g. a list of nearest neighbor cells in a retrieval database) by calculating the precision at exact-match cutoffs in the list, and then taking the mean of these. We followed the MAP calculation in Alavi's study (Alavi et al., 2018).

Identification of marker genes based on machine learning approaches

For the five-publication datasets, we first selected the top 20 percent of highly variable genes (5,986 genes) using the Seurat package and then identified SHAP and SVMM markers from these 5,986 genes. The TreeExplainer in SHAP package was used to calculate feature importance in the RF model (Lundberg et al., 2020). Briefly, this package calculated marginal contribution of each feature of a given observation from all model combinations. For a target feature, each combination contains one model with and another without this feature, and the marginal contribution can be calculated based on the difference yielding between these two models. Due to its local interpretability that each observation can get its own set of SHAP values, we can calculate these values of each gene under each cell type. The higher SHAP value suggests higher contribution of the feature to the classification. The novel marker genes assumed to have higher SHAP values than other genes. The implementation of the SVM model is based on libsvm (Chang and Lin, 2011). The absolute size of the coefficient relative to the other ones gave an indication of how important the feature was for the separation. We assumed the absolute coefficient values represent feature importance. The multiclass support of the SVM was handled according to a one-vs-one scheme. The attributes coefficients had the shape: (number_of_cell_type * (number_of_cell_type -1) / 2, number of features). To identify the feature importance of each gene on cell type, we calculated the average absolute coefficient values from all pairs for a specific cell type for each feature. Each

feature had one coefficient of each cell type. The cell type with the highest coefficient was assigned to the feature.

Identification of correlation marker genes

Pearson correlation analysis was conducted between the cell expression of known marker genes and other genes (Benesty et al., 2009). Each of the unknown markers had a correlation score for each cell type. We ranked the ten cell types for each marker based on the correlation score. An unknown marker was assigned to a cell type where this marker achieved the highest correlation score in this cell type.

Cell clustering

The integrated dataset with 25,618 cells and 25,092 genes was used for clustering analysis. The top 30 aligned correlated components were used as input for UMAP dimension reduction and clustering analysis. Clusters were identified using Seurat FindClusters function with default settings. The DoHeatmap and DotPlot function in the Seurat was used to visualize expression patterns of the novel marker genes for the identified clusters.

Method for SHAP markers consisting of the top 20 markers in each of ten cell types were used. The same methods were used to select 200 markers in SVMM and CORR. To select the top 20 BULR and KNOW markers of each cell type, we ranked them based on their expression specificity. To calculate the marker specificity for a specific cell type, we generated a cell vector by labeling cells under this cell type to 1, and all the other cells to 0. The cells under the specific cell type were defined based on the ICI method. The Pearson correlation analysis was used to calculate the correlation rate between the marker expression and the cell vector developed before. The higher absolute correlation rate means higher marker specificity. The top 180 BULR markers in the cell types were selected except for the Protophloem where no marker was detected. In the KNOW markers, the top 161 of them were selected since no Meri_xylem marker and only one Protoxylem marker were found. The 232 ICIM markers were all used in the comparison (**Table S4**).

Assign cell identity

In figure 2H, expression values for each marker genes were normalized across all cells and clusters using AverageExpression function in the Seurat package (Satija et al., 2015). For each of

the marker genes, the normalized expressions were ranked from high to low (from 1 to N with N equals the number of clusters, N=17 in our dataset). Finally, the average rankings of all marker genes were used to determine the cluster identity.

Classify cells into different developmental stages using machine learning methods

We used our published single cell data (Ryu et al., 2019), and root hair, non-root hair and lateral root caps cells were extracted. The selected 2,932 cells have been differentiated into nine sub-populations at different development stages (LateralRootCap, Differ_LateralRootCap, Early_Differ_NonHairEpiderm, Differ_NonHairEpiderm, NonHairEpiderm, RootHairEpiderm, Late_Differ_RootHairEpiderm, Mid_Differ_RootHairEpiderm, Early_Differ_RootHairEpiderm) in a previous analysis. This dataset was divided into independent, training, and validation datasets, and five-fold cross validation was conducted described above. The RF and SVM models were trained, and predictions were made on trichoblast and atrichoblast cells (11,904) extracted from the other four datasets (ICI score > 0.5) as well as 1,970 'positive' WER cells labeled by expressed WER-AT gene. Next, we integrated the 2,932 training cells and the 13,874 prediction cells using Seurat (v3.1) multicanonical correlation analysis with top 50 aligned correlated components as input for UMAP dimension reduction.

Compare overlapping ratio with rice markers among six marker types

A recent rice scRNA-seq study (Liu et al., 2021) had listed a number of candidate marker genes. There were three cell types (Cortex, Endodermis, Trichoblast) in that study that exactly matched three of ten cell types used in our study. To compare the ratio of cells where genes are detected in these cell types among the six different marker types, we identified rice orthologs based on information from Phytozome (v12.1) (Goodstein et al., 2012). The previous top 20 markers based on ICI score over 0.9 of each marker type were compared. The marker type with less than three rice orthologs was not considered. To compare the frequency of markers overlapping with the rice set, all the six types of markers corresponding to rice orthologs were used to overlap with these rice candidates. We randomly picked rice genes with the same orthologs number for each marker type. This step was repeated for 100 times to calculate an average overlapping ratio. We performed an exact binomial test (Wagner-Menghin, 2014) by setting this overlapping ratio as hypothesized

probability of success, number of overlapping markers as number of successes, number of rice candidate markers as number of trials, and alternative as 'greater'.

Methods for literature search and additional wet-bench experimental support for newly identified markers.

To identify supporting evidence of cell type specificity of new marker genes that are identified in our work, we performed literature search for all SHAP markers (200). We also searched SVMM and CORR markers (20 for each type) for trichoblast cell types. Because it is difficult to perform complete literature search automatically, this analysis is done as a demonstration that many newly identified markers have support from published, non-high throughput experiments. We used the following procedure for each marker genes used in this search. We first search the TAIR website using the ATxGxxxxx ID to extract all literature related to this gene ID. We then exclude the publications where more than 20 gene ids were associated with a single publication. This is to avoid finding evidence that were generated by high throughput methods such as RNAseq. For the rest of associated publications for each gene, we check the full text for evidence of promoter GFP/YFP/GUS reporters. The literature search results are provided in supplementary table S10. We found that in 11 cases, the newly identified SHAP markers had published reporter genes. In 2 cases, the published reporter genes are not in agreement with the specificity determined by SHAP/scRNA-seq data. For the rest 187 genes, there are no published evidence based on promoter-reporter genes. Here is a list of genes that are supported by published literatures.

1. Trichoblast: AT2G21045 (HAC1). SHAP selected this gene as trichoblast marker. Promoter-GFP shows this gene is expressed in epidermal layers, and in mature trichoblast which is overlapping with WER-GFP expressed mature trichoblast. Fischer et al., Journal of Experimental Botany, 2021. DOI: 10.1093/jxb/eraa465

2. Atrichoblast: At2g41800 (DUF642). SHAP and CORR selected this gene as atrichoblast (non-hair epidermal cells) marker. Figure 1 and 4 in this paper shows promoter-GFP fusion of this gene is expressed in epidermal cells. Salazar-Iribe et al., Plant Science, 2016. DOI: 10.1016/j.plantsci.2016.10.007

3. Endodermis: AT2G37180 (PIP2;2). SHAP and SVMM selected this gene as an endodermis marker. Figure 4 of this publication shows highly specific endodermis expression of promoter-GUS reporter of this gene. Javot et al., The Plant Cell, 2003. DOI: 10.1105/tpc.008888

4. Endodermis: AT3G22600 (ATXLP12). SHAP/SVMM/CORR selected this gene as an endodermis marker. Figure 5 in this paper shows strong endodermis expression of this marker with

some expression in pericycle. Kobayashi et al., Plant and Cell Physiology, 2011. DOI: 10.1093/pcp/pcr060

5. Protoxylem: AT4G04460 (PASPA3). SHAP selected this gene as protoxylem marker. Figure 5C of this paper shows this gene expressed in differentiating protoxylem. Fendrych et al. Current Biology, 2014. DOI: https://doi.org/10.1016/j.cub.2014.03.025.

6. Protoxylem: AT2G40320 (TBL33). SHAP selected this gene as protoxylem marker. Figure 1B of this paper shows this gene expressed in protoxylem. Yuan et al. PloS One. 2016. DOI: https://doi.org/10.1371/journal.pone.0146460.

7. Protophloem: AT4G29920 (SMXL4). SHAP/CORR selected this gene as protophloem marker. Figure 1 and 2 of this paper shows that this gene promoter is active protophloem differentiation. Wallner et al., Current Biology, 2017. DOI: https://doi.org/10.1016/j.cub.2017.03.014.

8. Phloem: AT5G02600 (NAKR1). SHAP/CORR selected this gene as phloem_CC marker. Figure 3F of this paper shows that the promoter-GUS of this gene is expressed in phloem. Shibuta and Abe. Plant Cell Physiology, 2017. DOI: https://doi.org/10.1093/pcp/pcx133.

9. Phloem: AT3G21190 (MSR1). SHAP selected this gene as protophloem marker. Figure 4 of this paper shows that the promoter-GUS of this gene is expressed in Phloem. Wang et al. The Plant Journal, 2012. DOI: https://doi.org/10.1111/tpj.12019.

10. Cortex: AT2G25810 (TIP4;1) SHAP selected this gene as cortex marker. Figure 1 of this paper shows that the TIP4;1 expressed in both epidermal and cortex. Gattolin et al., BMC Plant Biology, 2009. DOI: 10.1186/1471-2229-9-133

11. Cortex: AT2G45960 (PIP1;2). SHAP selected this gene as cortex marker. Figure 1B of this paper shows that the promoter-GUS of this gene is expressed in cortex, endodermis and stele. Postaire et al., Plant Physiology, 2009. DOI: 10.1104/pp.109.145326

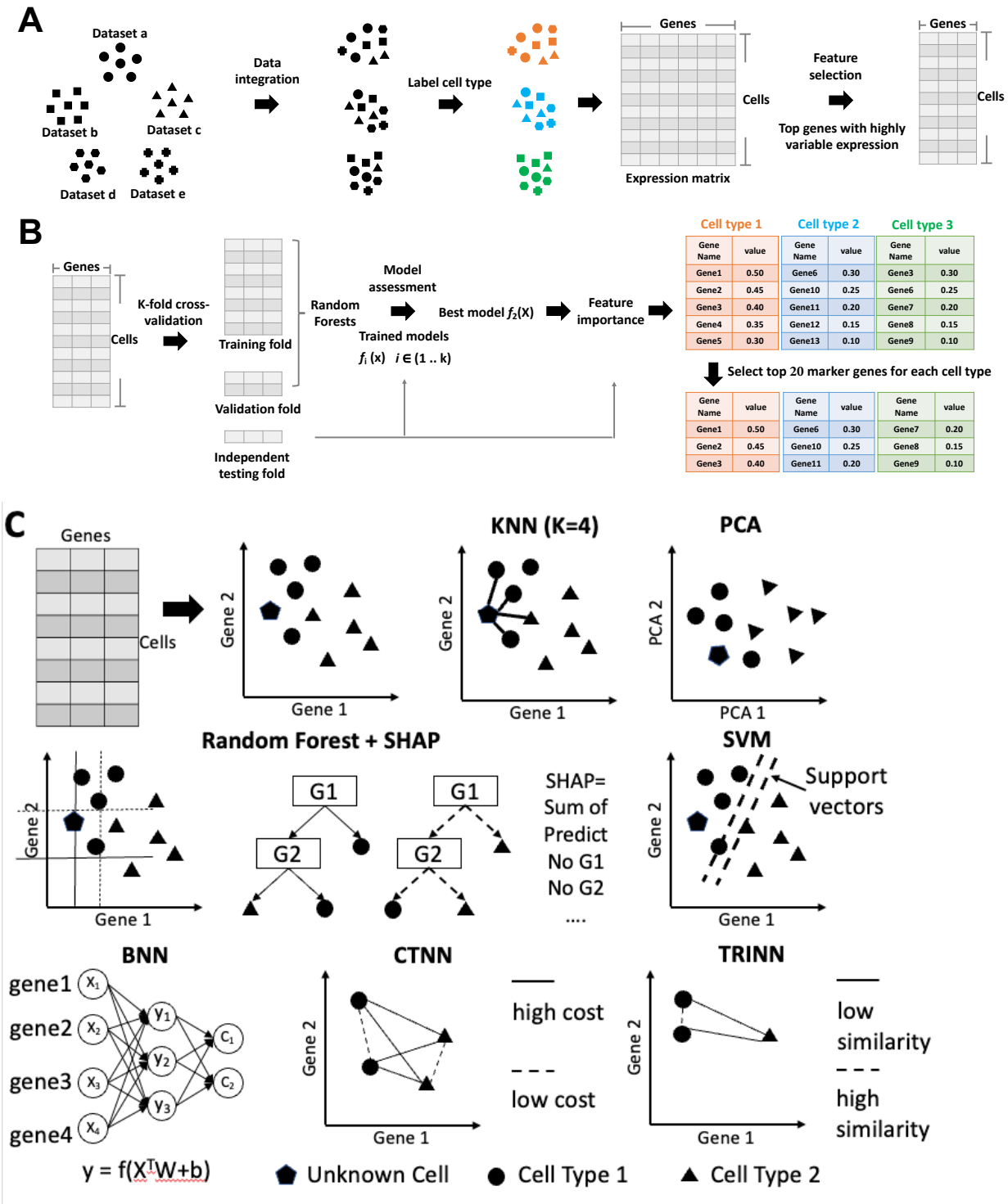


Figure S1 Summary of the SPmarker. A. Data processing pipeline. The different datasets are integrated together. After labeling the cells, a gene by cell expression matrix is built. The top genes with highly variable expression are selected to build a new expression matrix. **B. Model training and identification of SHAP marker genes**. The

integrated expression matrix was divided into the training dataset (90%) and the independent testing dataset (10%). The independent testing was used to evaluate the prediction performance a $f_i(x)$ model trained with the training dataset. The best model ($f_2(x)$ in this case) was selected to identify the feature importance using the SHAP method. The top SHAP marker genes were selected from each cell type such that each cell type having its own marker genes that are not shared with others. **C.** Explanations of different machine learning methods used in this study.

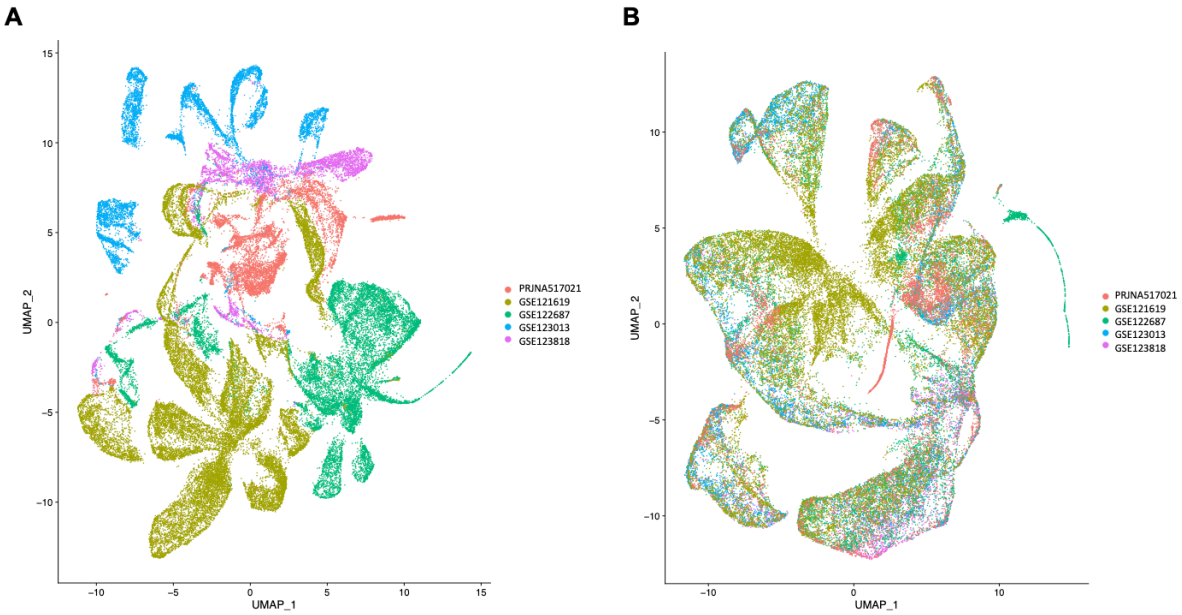


Figure S2 Integration of five datasets using the canonical correlation function in the Seurat package. A. Data before integration. B. Data after integration.

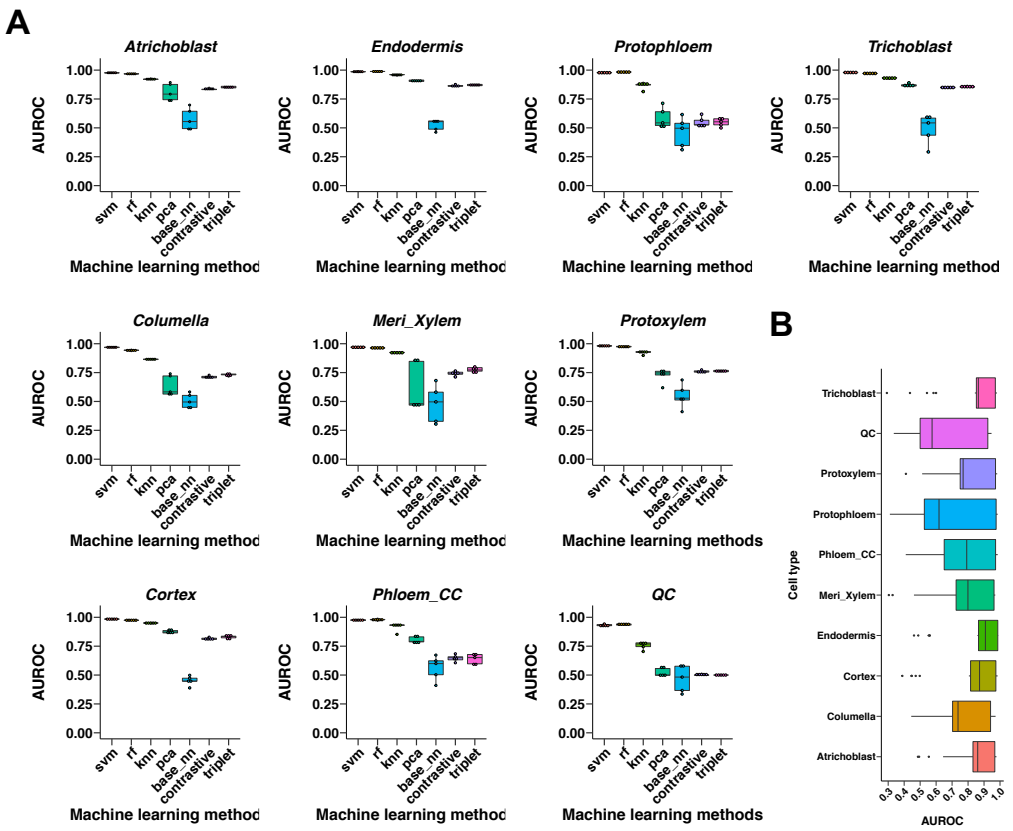


Figure S3 Classification performance (AUROC) of ten root cell types of *Arabidopsis*. A. comparison of seven machine learning models on cell type classification. In these boxplots, the mid-horizontal line represents the median and dots represent data points. B. comparison of classification performance of all the ten cell types. Dots represent outliers. AUROC means Area Under the Receiver Operating Characteristics. Number of cells used in this figure is the same as shown in **Figure 1A**.

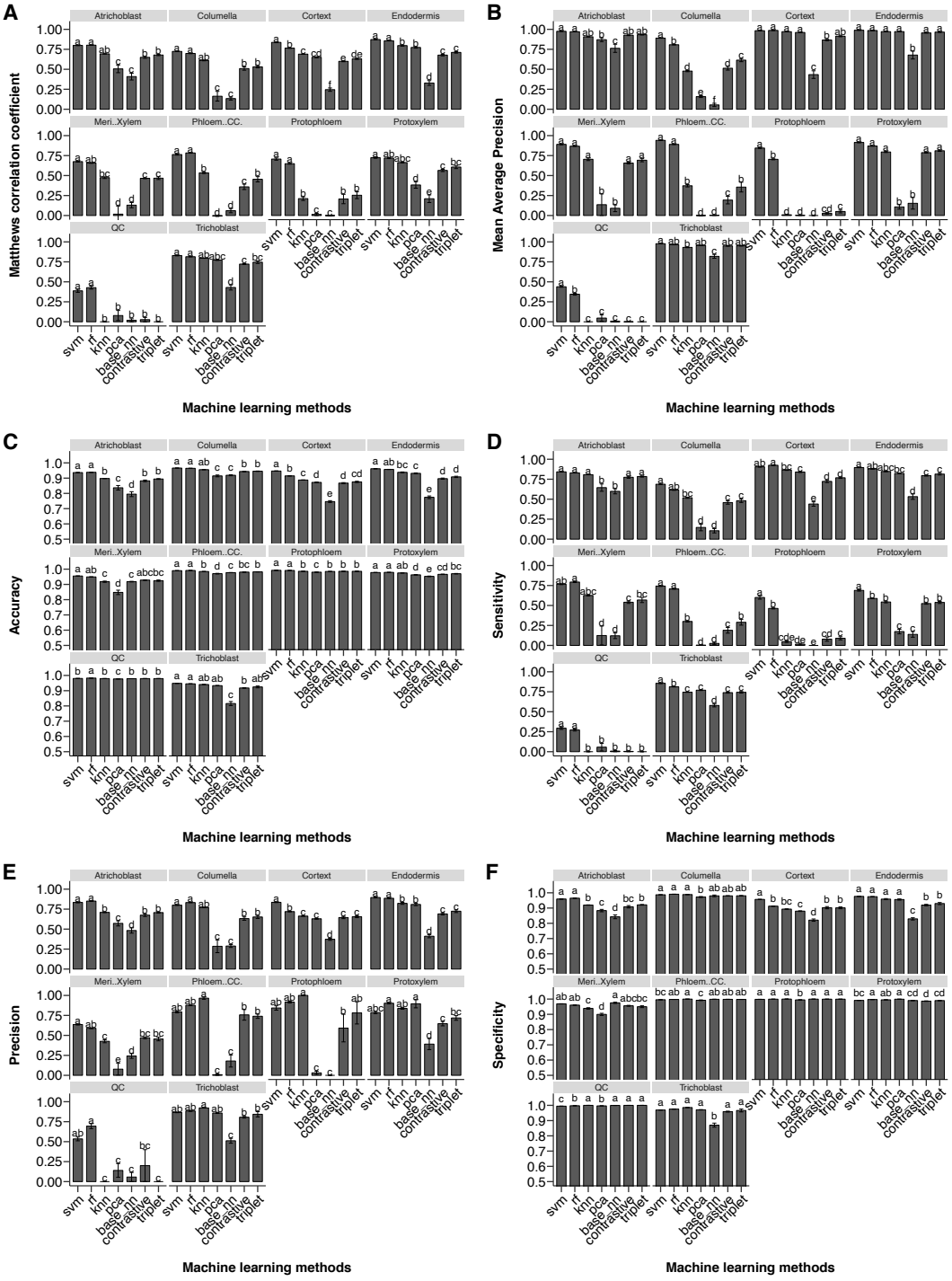


Figure S4 Performance comparison among seven machine learning methods based on six different evaluation scores. A. Matthew's correlation coefficient. B. Mean average precision. C. Accuracy. D. Sensitivity. E. Precision. F. Specificity. All pair wise comparisons are statistically significant as indicated by different letters (a, b, c, d, and e). Error bars represent +/- SE.

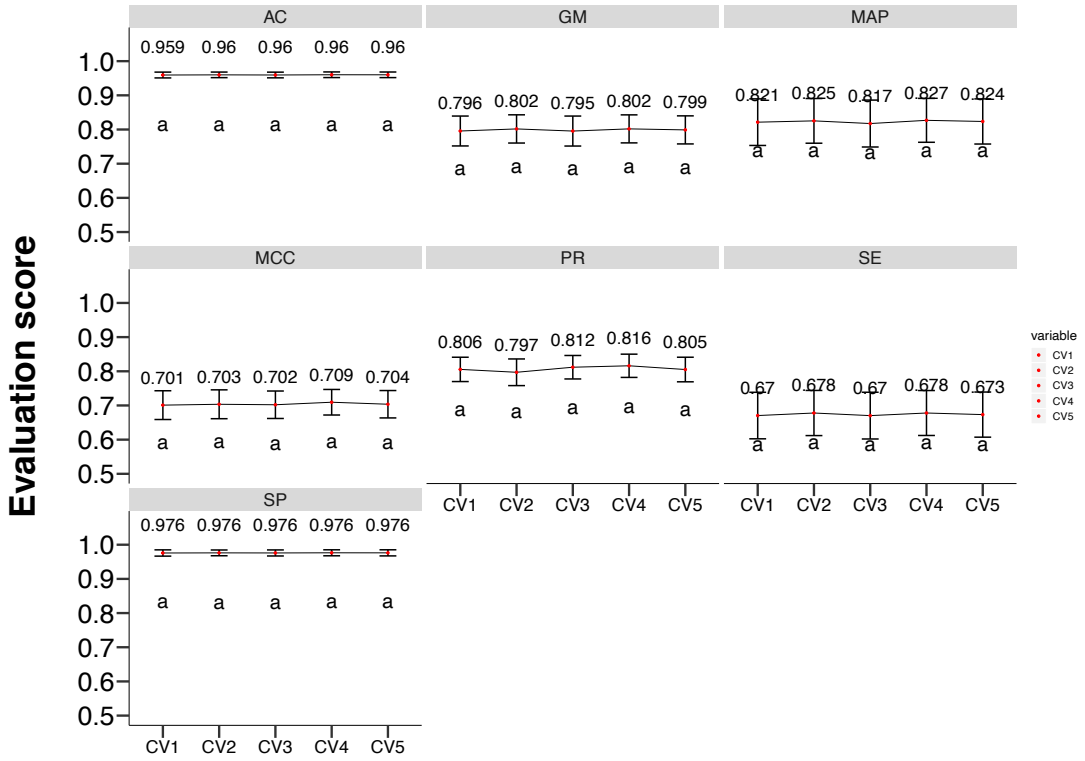
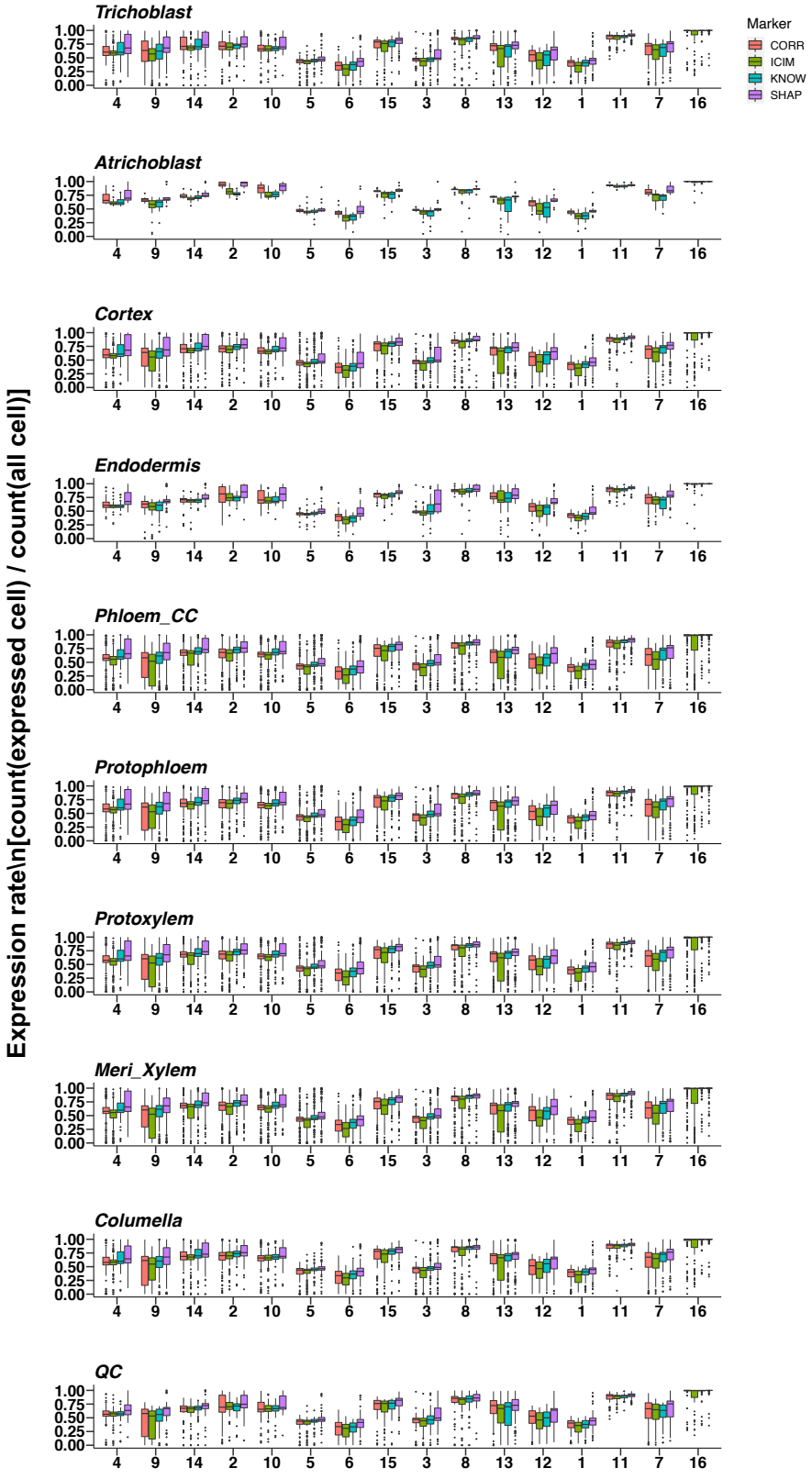


Figure S5 Performance comparisons among five cross validation random forest models. CV1 to CV5 suggest models obtained from the five-fold cross validation. The error bar suggests evaluation score variations of the ten cell types. The evaluation scores include sensitivity (SE), accuracy (AC), specificity (SP), precision (PR), geometric mean (GM), matthews correlation coefficient (MCC), and mean average precision (MAP). All pair wise comparisons are not statistically significant, as represented by the same letter a.


Trichoblast
Atrichoblast
Cortex
Endodermis
Phloem_CC
Protophloem
Protoxylem
Meri_Xylem
Columella
QC
Marker
CORR
ICIM
KNOW
SHAP
Expression rate\n[count(expressed cell) / count(all cell)]
1.00
0.75
0.50
0.25
0.00
4 9 14 2 10 5 6 15 3 8 13 12 1 11 7 16

1

Figure S6 Comparisons of proportion of expressed cells among the SHAP, CORR, ICIM, and KNOW markers across all the clusters. In these boxplots, the mid-horizontal line represents the median and dots represent data outliers.

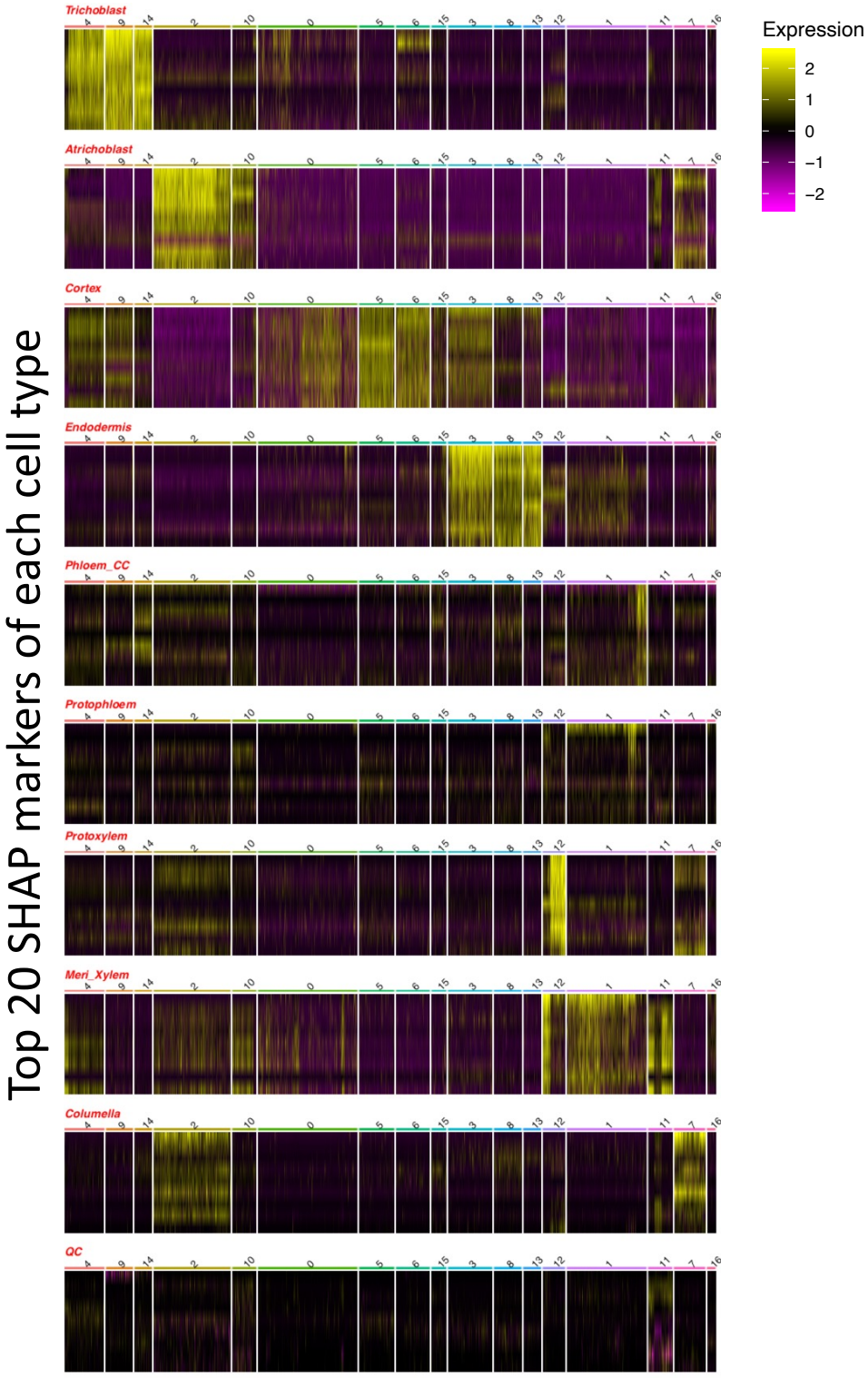


Figure S7 Heat map of top 20 SHAP markers from each cell type across all the 17 cell clusters.

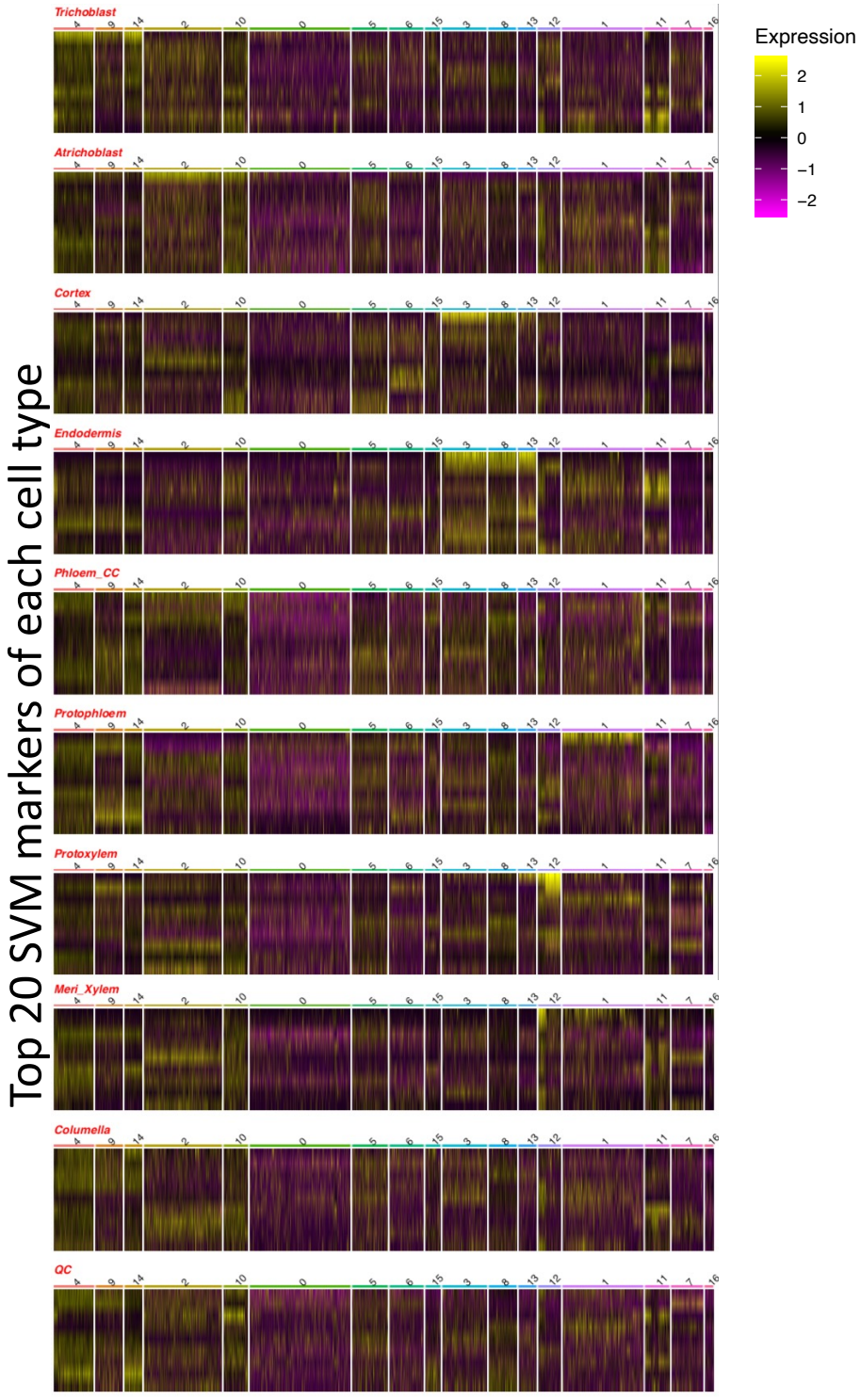


Figure S8 Heat map of top 20 SVMM markers from each cell type across all the 17 cell clusters.

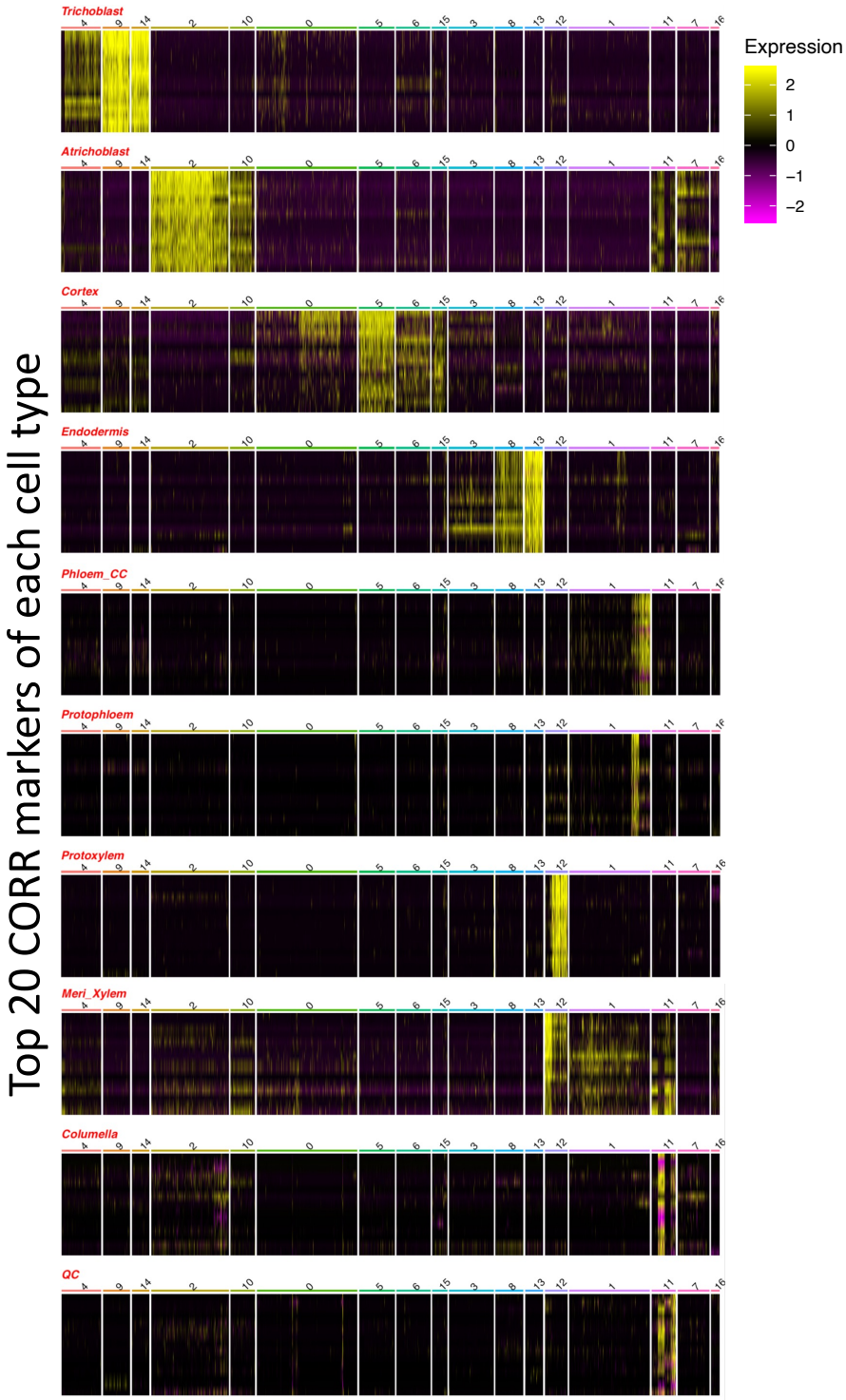


Figure S9 Heat map of top 20 CORR markers from each cell type across all the 17 cell clusters.

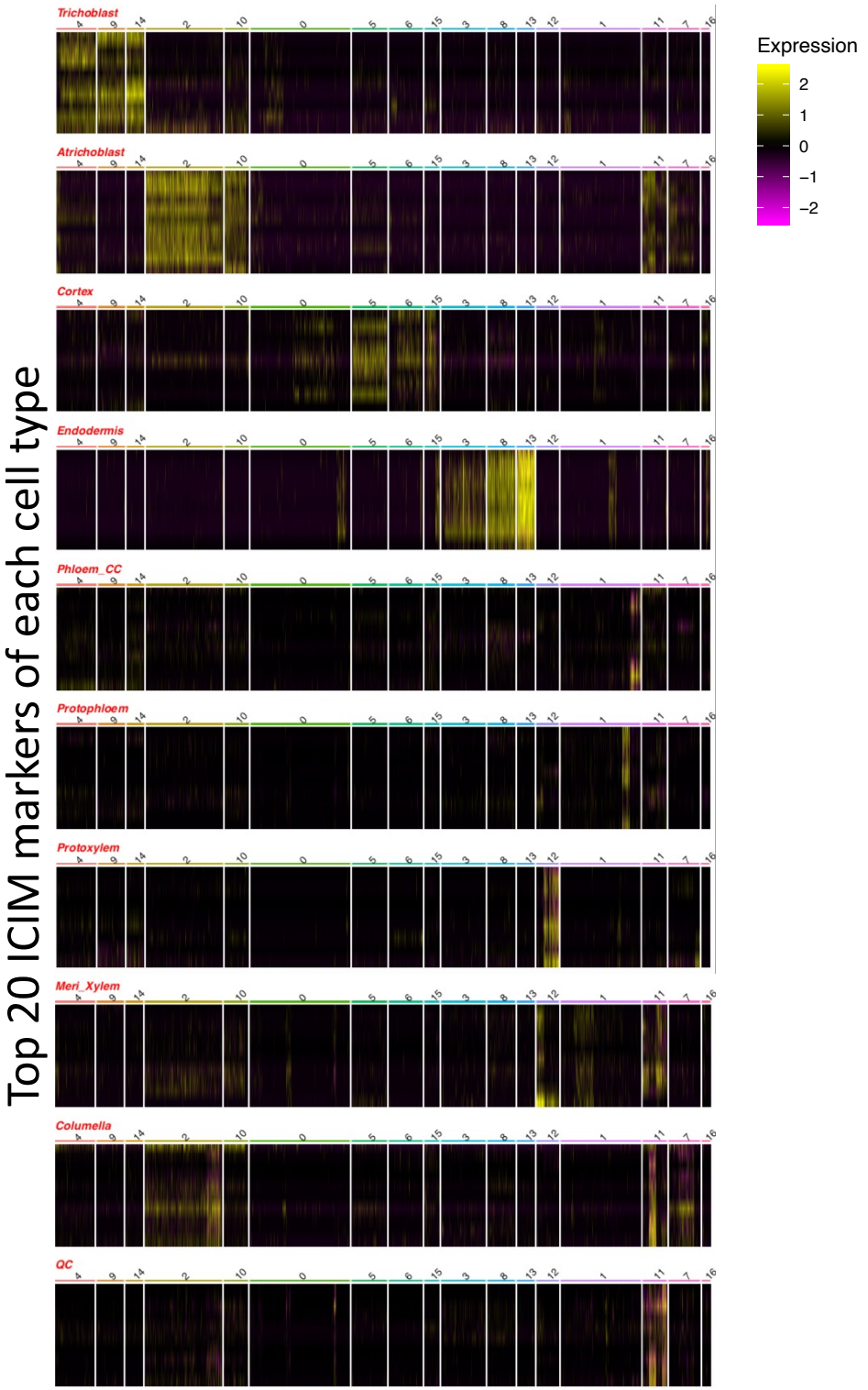


Figure S10 Heat map of all ICIM markers from each cell type across all the 17 cell clusters.

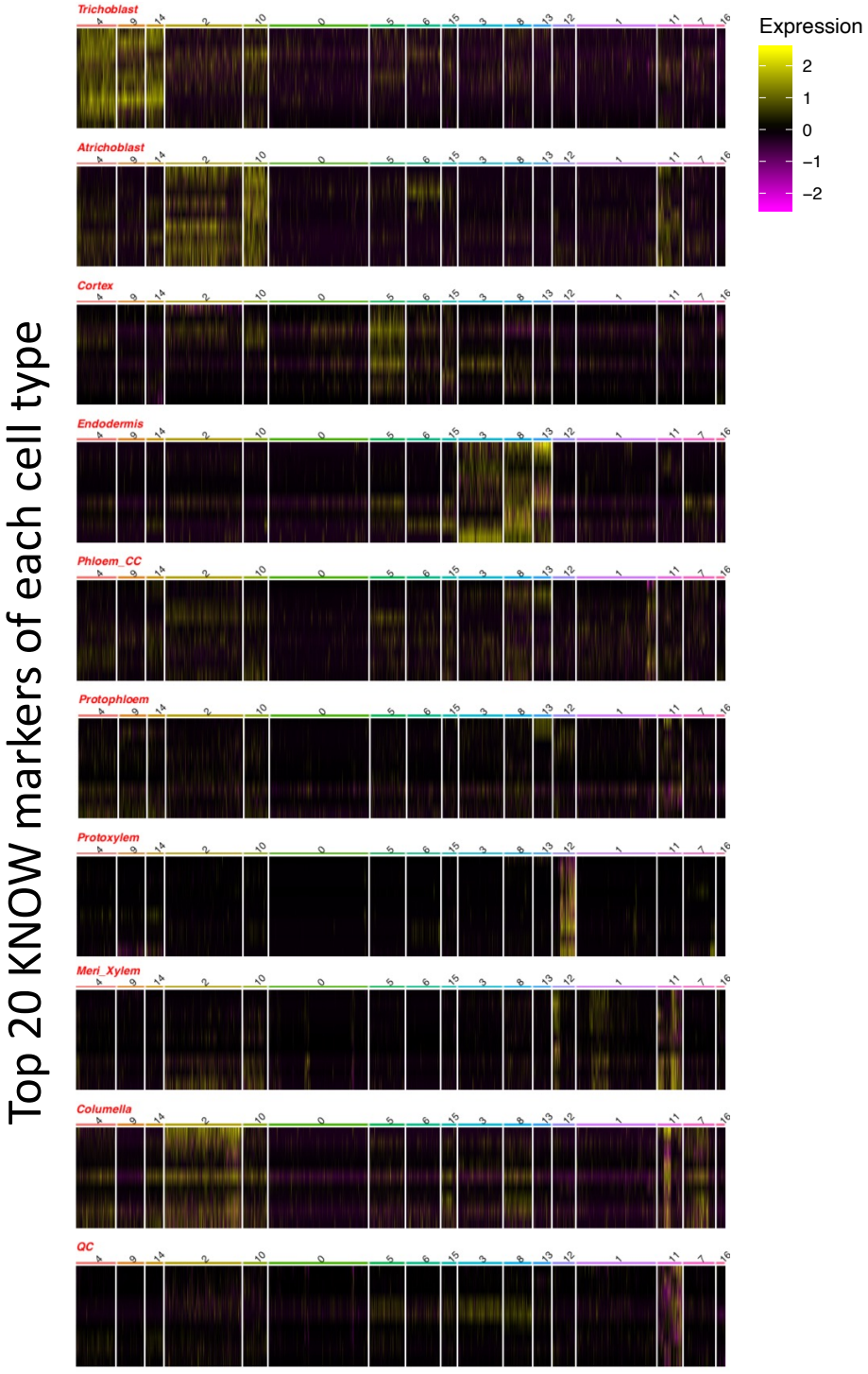


Figure S11 Heat map of top 20 KNOW markers randomly selected from each cell type across all the 17 cell clusters

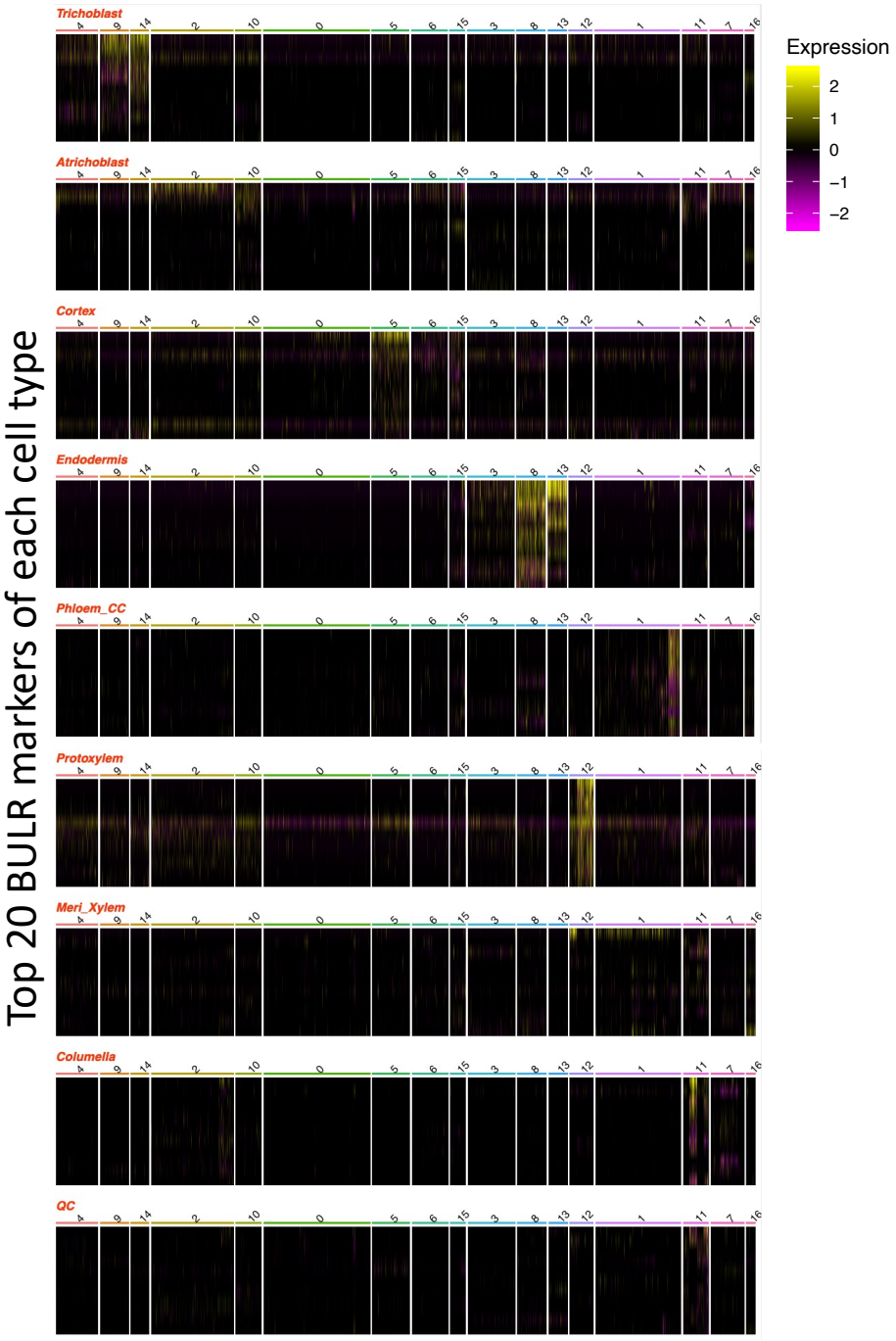


Figure S12 Heat map of top 20 BULR markers from each cell type across all the 17 cell clusters.

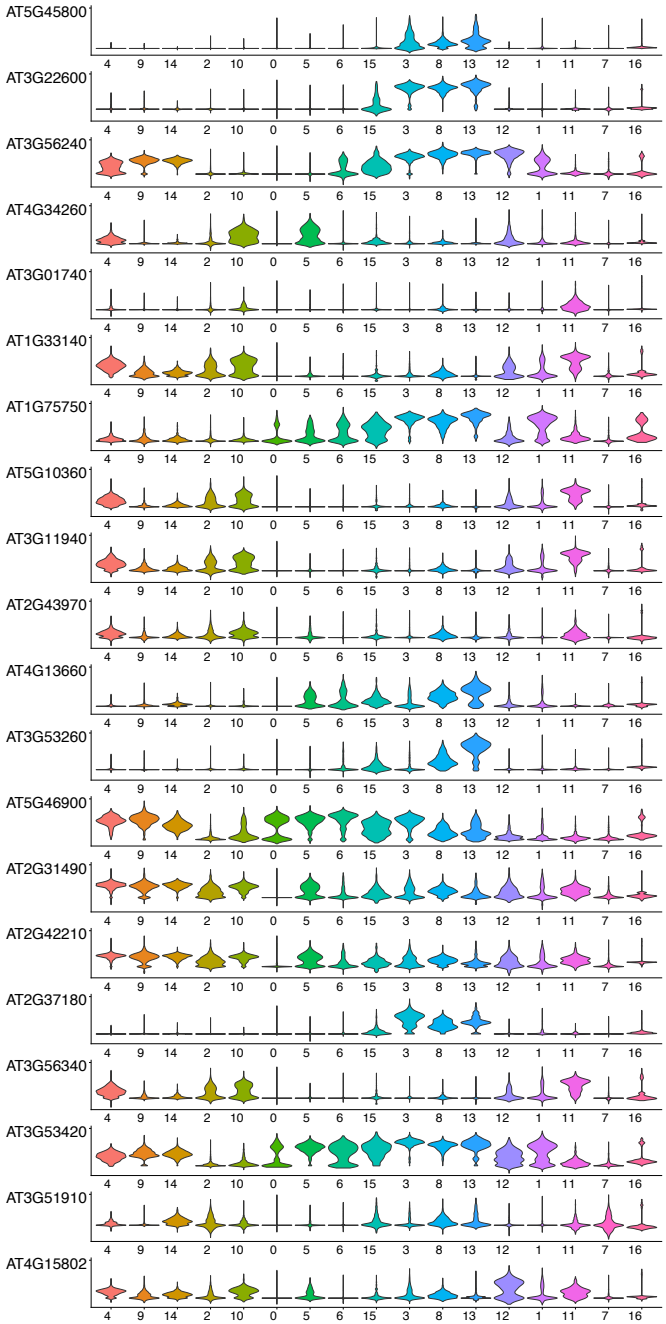


Figure S13 Violin plots of top 20 SVMM markers of endodermis cell type. Violin plots only show the distribution of the data.

1
2

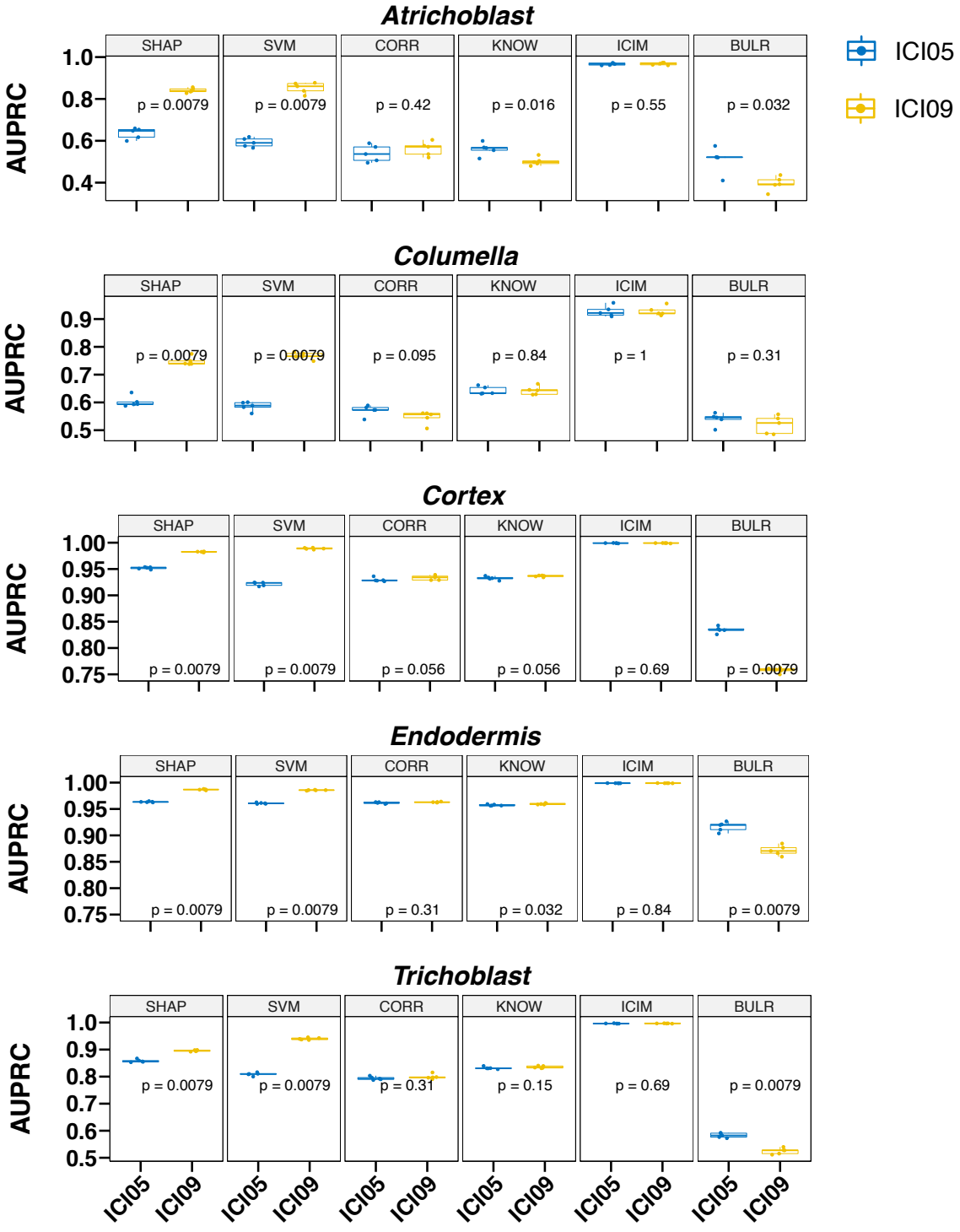


Figure S14 Comparison of classification performance based on ICI labeling method between 0.5 and 0.9 thresholds in five cell types. p value < 0.05 indicates significant differences between ICI05 and ICI09 groups. In these boxplots, the mid-horizontal line represents the median and dots represent data points.

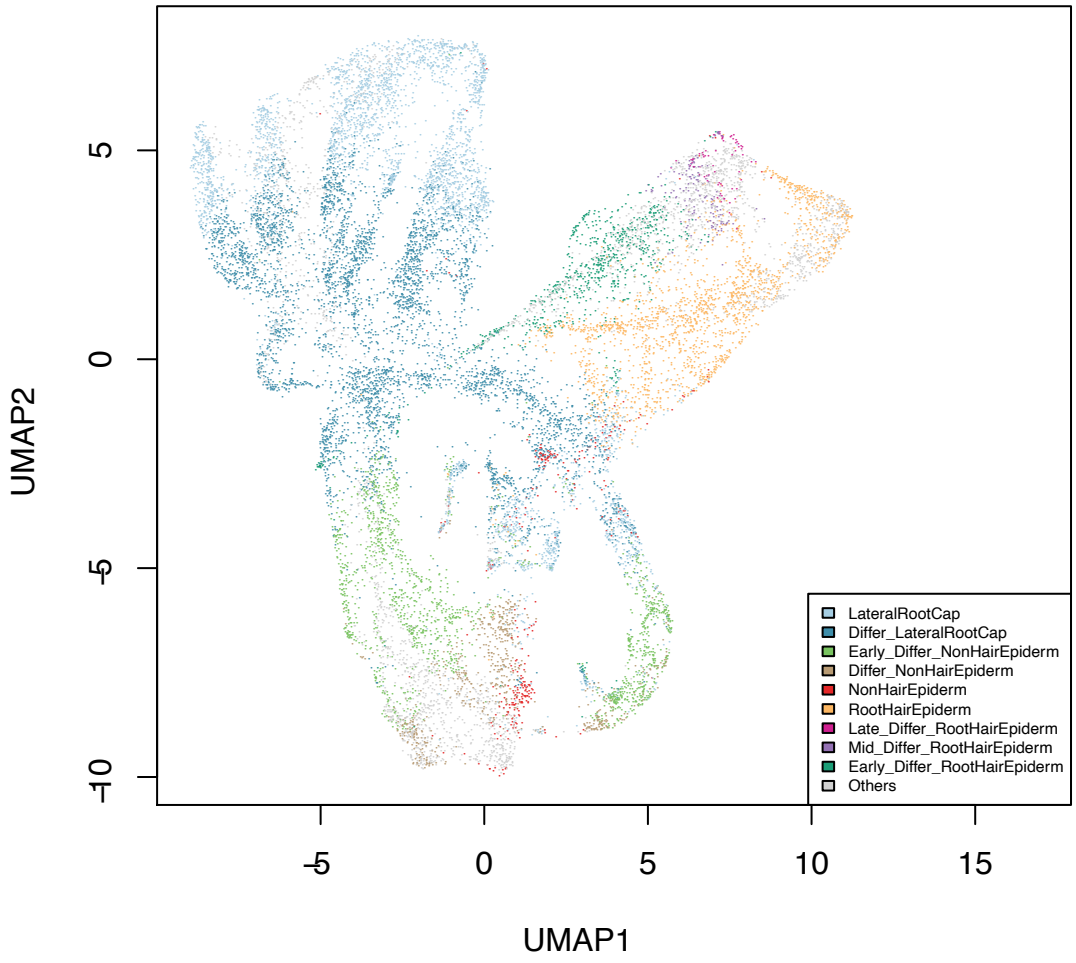
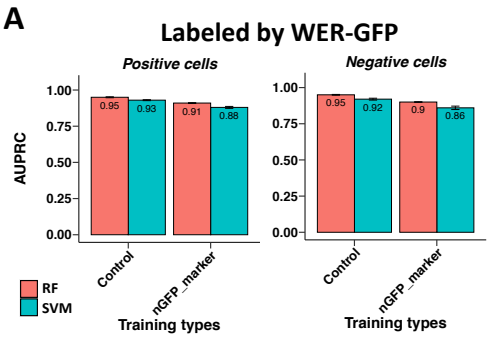


Figure S15 A UMAP for SVM predicted cell types. Note that a group of cells predicted as non-hair epidermal cells by SHAP random forest model are predicted as lateral root cap cells. 'Others' label indicates cells from Ryu's study (2019).



B

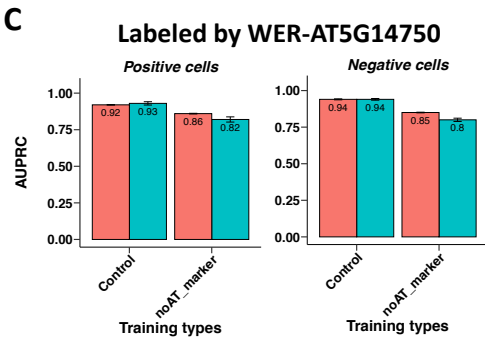
Labeled by WER-GFP

Ranking (SHAP)	SHAP marker	Correlation coefficient (GFP)	Ranking (Correlation)
1	AT5G14750	0.385	1
2	AT5G12940	0.283	3
3	AT5G62340	0.276	6
4	AT1G54000	0.226	29
5	AT2G42570	0.197	78

Ranking (SHAP)	SHAP marker	Correlation coefficient (GFP)	Ranking (Correlation)
1	AT5G14750	0.385	1
55	AT1G68560	0.292	2
2	AT5G12940	0.283	3
44	AT3G14530	0.281	4
175	AT5G60520	0.280	5

Ranking (SVM)	SVM marker	Correlation coefficient (GFP)	Ranking (Correlation)
1	AT5G14750	0.385	1
2	AT2G14890	0.178	138
3	AT5G49280	0.165	216
4	AT1G12090	0.038	8,228
5	AT4G22590	0.056	5,511

Ranking (SVM)	SVM marker	Correlation coefficient (GFP)	Ranking (Correlation)
1	AT5G14750	0.385	1
1751	AT1G68560	0.292	2
139	AT5G12940	0.283	3
1824	AT3G14530	0.281	4
6825	AT5G60520	0.280	5



D

Labeled by WER-AT5G14750

Ranking (SHAP)	SHAP marker	Correlation coefficient (AT5G14750)	Ranking (Correlation)
1	AT2G14890	0.221	114
2	AT1G28290	0.296	11
3	AT5G48900	0.281	16
4	AT4G38740	0.234	76
5	AT3G15950	0.221	112
.....	.....	.....	.....
19	GFP	0.385	1
.....	.....	.....	.....

Ranking (SHAP)	SHAP marker	Correlation coefficient (AT5G14750)	Ranking (Correlation)
19	GFP	0.385	1
284	AT5G60520	0.339	2
16	AT5G12940	0.336	3
44	AT5G62340	0.328	4
60	AT3G14530	0.321	5

Ranking (SVM)	SVM marker	Correlation coefficient (AT5G14750)	Ranking (Correlation)
1	AT2G01540	-0.019	20,567
2	AT3G61070	0.086	3,734
3	AT4G14420	0.170	428
4	AT1G27970	0.073	4,936
5	AT5G65925	0.144	918
.....	.....	.....	.....
334	GFP	0.385	1
.....	.....	.....	.....

Ranking (SVM)	SVM marker	Correlation coefficient (AT5G14750)	Ranking (Correlation)
334	GFP	0.385	1
9967	AT5G60520	0.339	2
323	AT5G12940	0.336	3
333	AT5G62340	0.328	4
4756	AT3G14530	0.321	5

Figure S16. Classification and predicted top markers. A. Comparison of classification performance on GFP-labeled WER cells (positive cells) and none GFP-labeled WER cells (negative cells) between using all genes (control) and genes without GFP marker (nGFP_marker) for both RF and SVM models. B. Ranking of best SHAP and SVMM markers to predict WER-GFP positive cells (left two tables). Ranking of genes with top correlation values with the GFP markers (right two tables). C-D show the similar legends as A-B except cells were labeled by AT5G14750. AUPRC means Area Under Precision-Recall Curve. Error bars represent +/- SE.

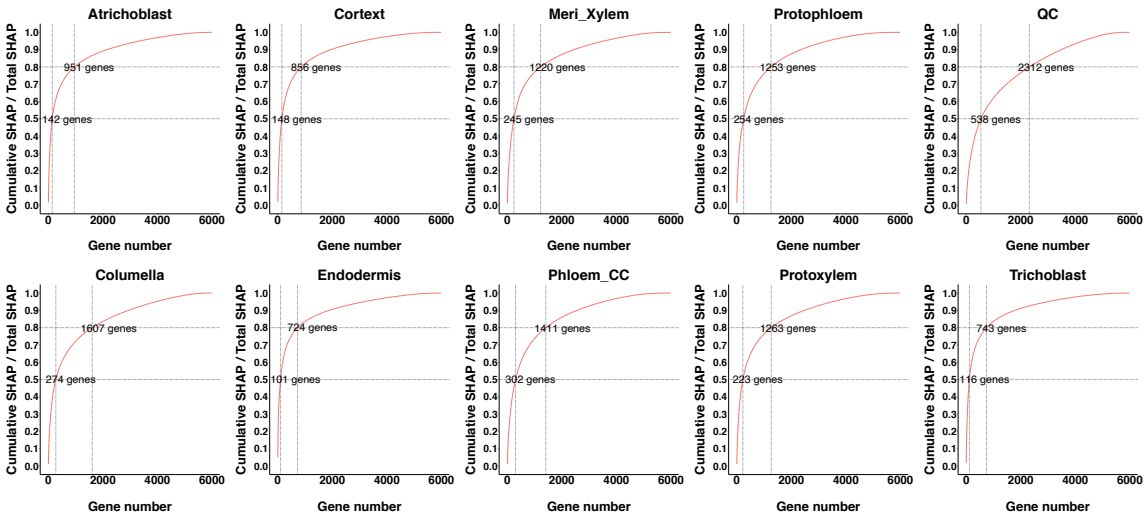


Figure S17. Cumulative SHAP values for all SHAP markers for each of the ten cell types.

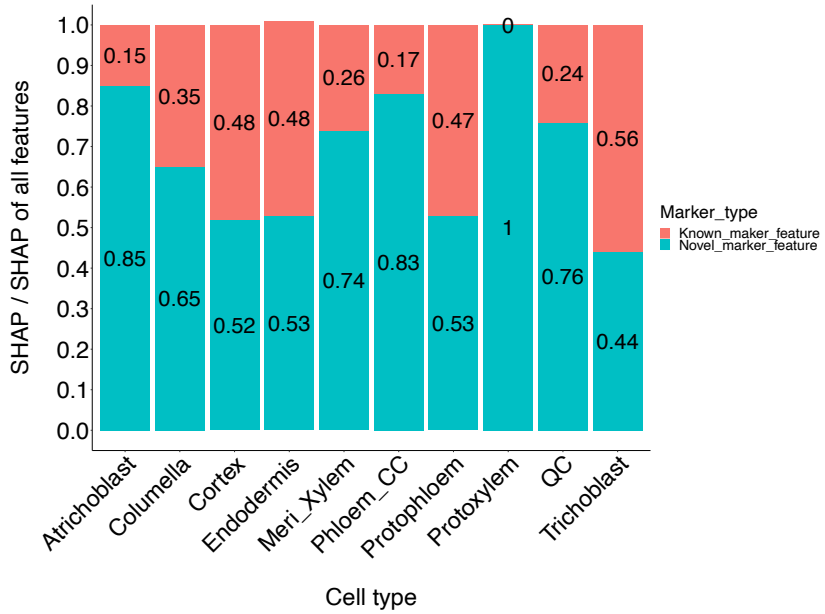


Figure S18 Comparison of proportion of cumulative SHAP values from the SHAP to and from the known markers in the top 20 features with the highest SHAP value in each cell type.

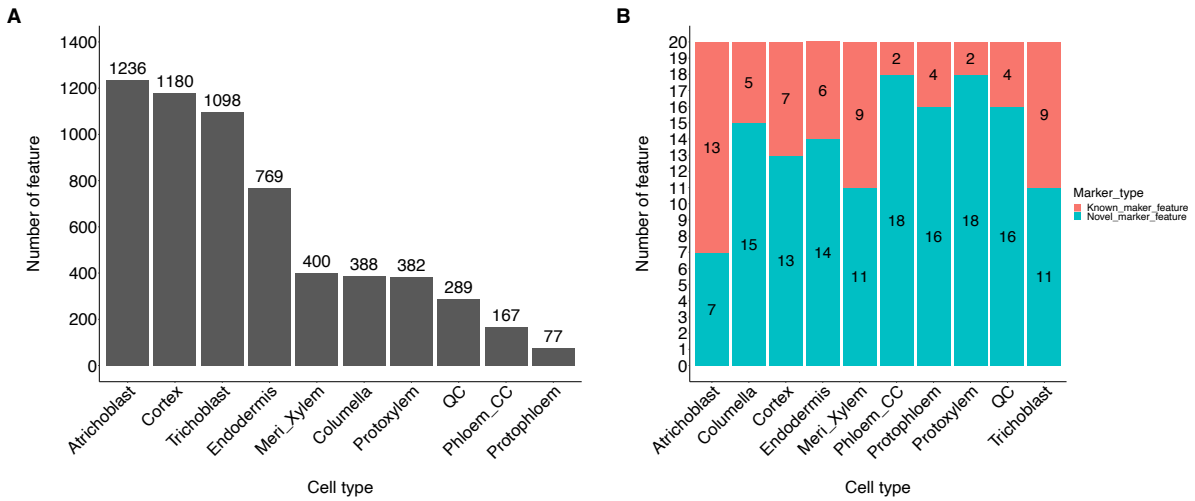


Figure S19 Summary of the SVMM markers. The SVMM genes were determined based on the feature importance estimated from the absolute coefficient values in the SVM model. Each gene has a coefficient for each of the ten cell types, and each gene is assigned to the cell type with the highest coefficient. The number of unique marker genes and novel marker genes determined by SVM are similar to that determined by SHAP. For example, most unique SVMM genes were assigned to atrichoblast (1,236), cortex (1,180) and vascular tissue and QC cells have lower numbers of SVMM markers (A). However, some cell types, such as trichoblast, have approximately twice as many SVMM markers as SHAP markers (1,098 SVMM vs 555 SHAP). In the top 20 genes with the highest coefficient of each cell type, on average, 69.5% of SVMM are novel marker genes (B).

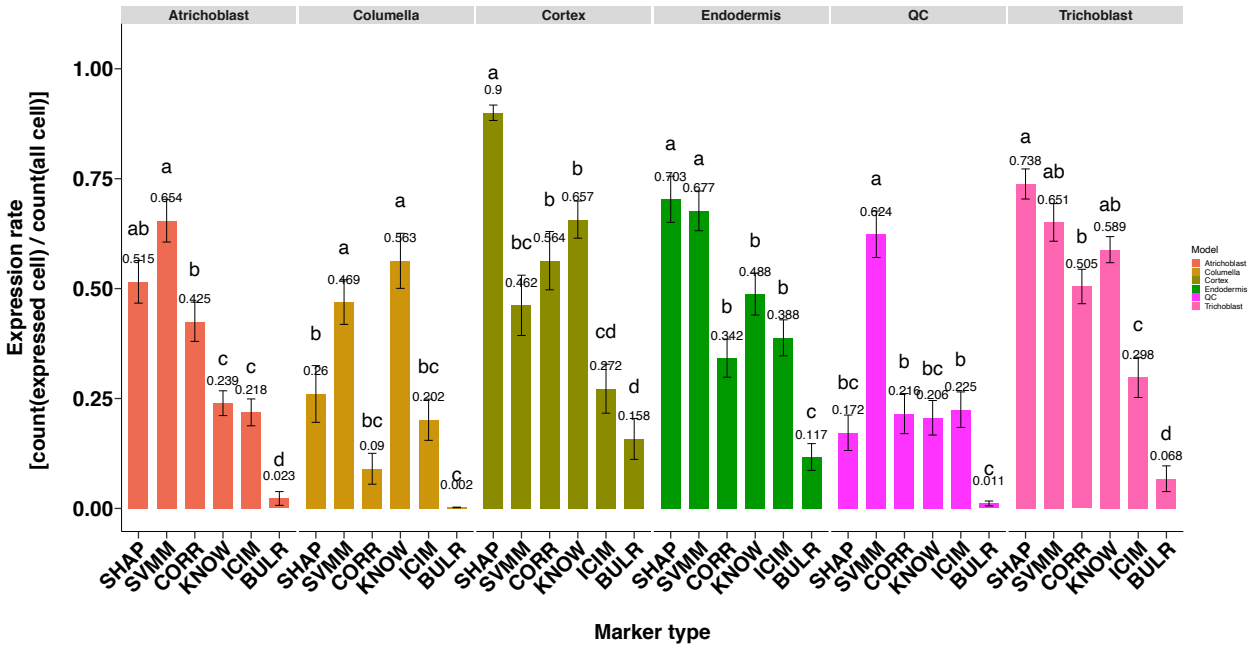


Figure S20 Expression rate of SHAP marker genes in (Shahan et al., 2020). All pair wise comparisons are statistically significant as indicated by different letters (a, b, c, and d). If two bars have the same letter, then they are not significantly different from each other. Error bars represent +/- SE.

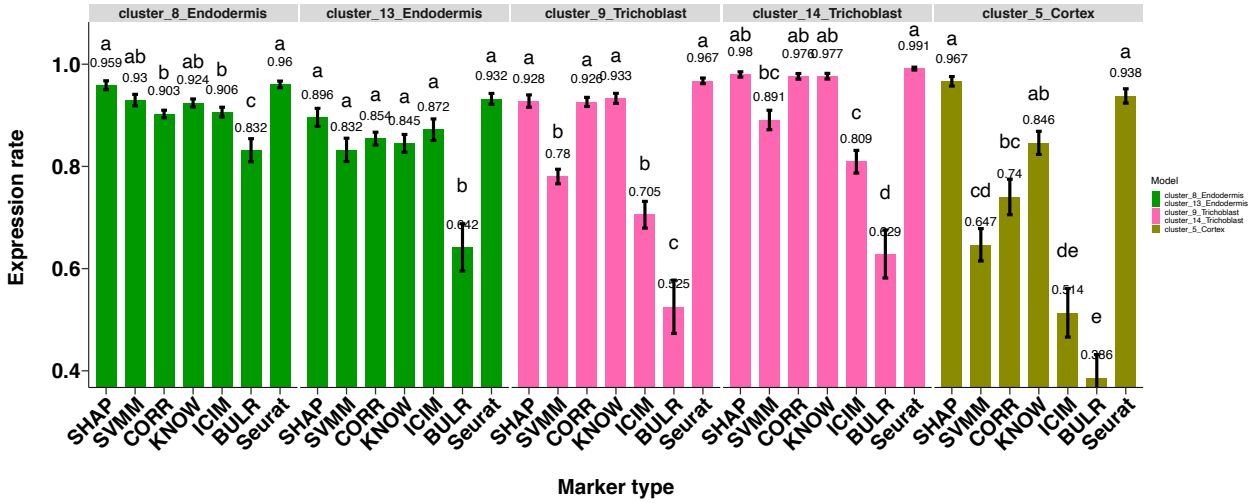


Figure S21. Expression rate comparison for five clusters between different marker types. Error bars represent +/- SE. All pair wise comparisons are statistically significant as indicated by different letters (a, b, c, d, and e).

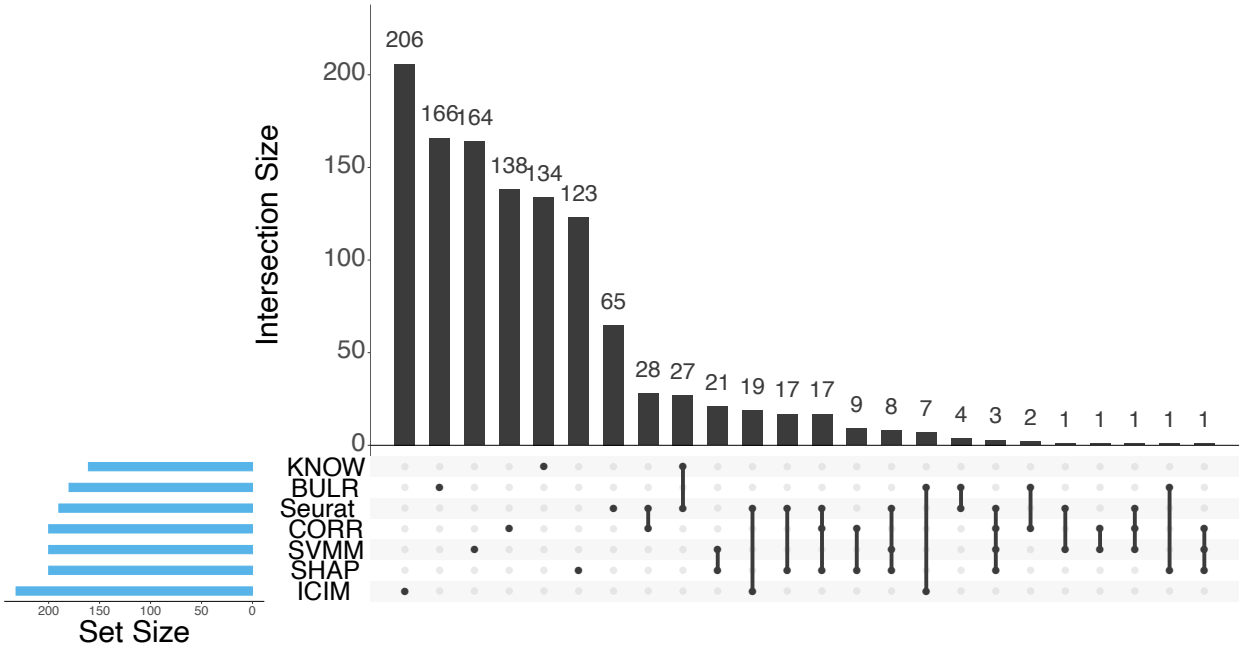


Figure S22. Number of marker genes uniquely and commonly identified by different methods Intersection size means gene count of different marker types. The dots under the bars mean the genes specifically exist in the relative marker type. The line connected between two or more dots under the bars mean genes exist in two or more marker types. If two or more marker types do not have connection, it means these groups do not have shared genes.

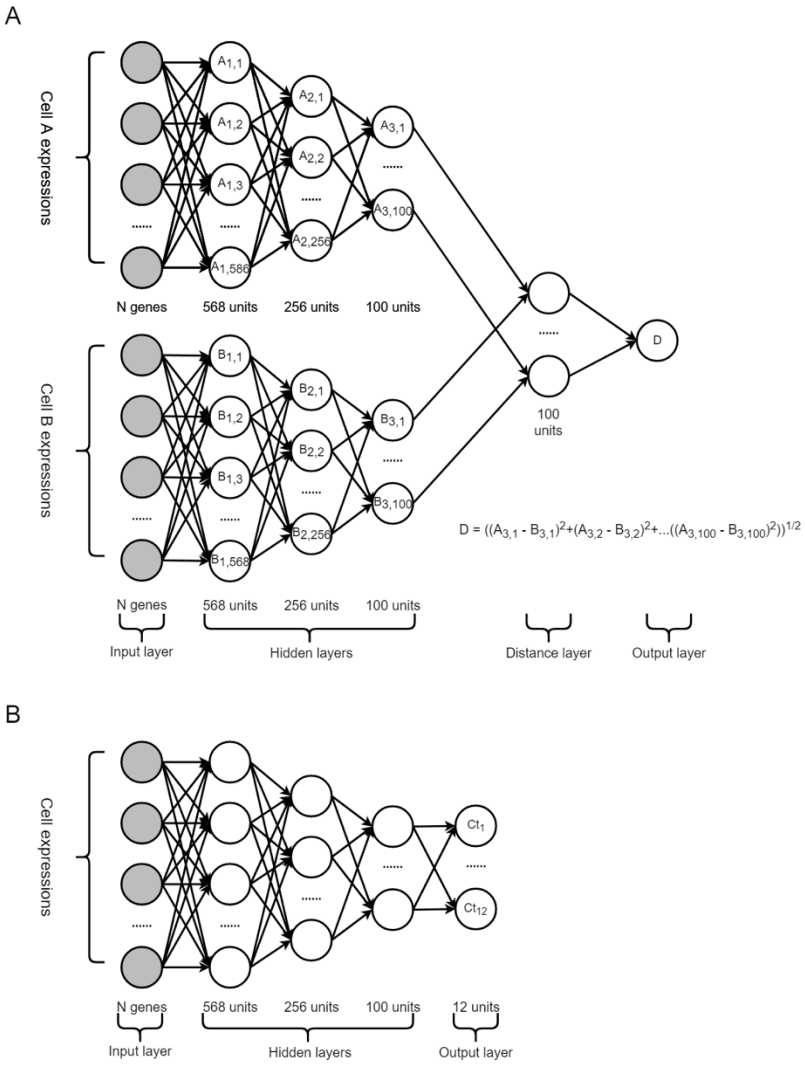


Figure S23 Schematic demonstration of architecture for each type of neural network. A. Architecture of Siamese NN, which was used for both triplet NN and contrastive NN. The distance layer computes a vector of distance between the last two hidden layers A3 and B3. This distance was then used in the objective function of triplet NN and contrastive NN to train cell type classifier. B. Architecture of multi-task NN. Ct represents a cell type (10 cell types were used in total for classification)

Reference

Abdi, H., and Williams, L.J. (2010). Tukey's honestly significant difference (HSD) test. Encyclopedia of Research Design. Thousand Oaks, CA: Sage, 1-5.

Alavi, A., Ruffalo, M., Parvangada, A., Huang, Z., and Bar-Joseph, Z. (2018). A web server for comparative analysis of single-cell RNA-seq data. Nature communications *9*, 1-11.

Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In Noise reduction in speech processing (Springer), Editors: Jacob Benesty, Jingdong Chen, Yiteng Huang and Israel Cohen. pp. 1-4.

Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) *2*, 1-27.

Goodstein, D.M., Shu, S., Howson, R., Neupane, R., Hayes, R.D., Fazo, J., Mitros, T., Dirks, W., Hellsten, U., and Putnam, N. (2012). Phytozome: a comparative platform for green plant genomics. Nucleic acids research *40*, D1178-D1186.

Liu, Q., Liang, Z., Feng, D., Jiang, S., Wang, Y., Du, Z., Li, R., Hu, G., Zhang, P., and Ma, Y. (2021). Transcriptional landscape of rice roots at the single-cell resolution. Molecular Plant *14*, 384-394.

Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature machine intelligence *2*, 2522-5839.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research *12*, 2825-2830.

Ryu, K.H., Huang, L., Kang, H.M., and Schiefelbein, J. (2019). Single-cell RNA sequencing resolves molecular relationships among individual plant cells. Plant physiology *179*, 1444-1456.

Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., and Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. Nature biotechnology *33*, 495-502.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815-823.

Shahan, R., Hsu, C.-W., Nolan, T.M., Cole, B.J., Taylor, I.W., Vlot, A.H.C., Benfey, P.N., and Ohler, U. (2020). A single cell Arabidopsis root atlas reveals developmental trajectories in wild type and cell identity mutants. bioRxiv. DOI: 10.1101/2020.06.29.178863

Wagner-Menghin, M.M. (2014). Binomial test. Wiley StatsRef: Statistics Reference Online.