Single-cell identity definition using random forests and recursive feature elimination (scRFE)

Madeline Park[1], Sevahn Vorperian[2], Sheng Wang[3, 4], Angela Oliveira Pisco[1*]

1 Chan Zuckerberg Biohub, San Francisco, California, USA
2 Department of Chemical Engineering, Stanford University, Stanford, California, USA
3 Department of Bioengineering, Stanford University, Stanford, California, USA
4 Department of Genetics, Stanford University, Stanford, California, USA

* angela.pisco@czbiohub.org

Abstract

Single cell RNA sequencing (scRNA-seq) enables detailed examination of a cell's underlying regulatory networks and the molecular factors contributing to its identity. We developed scRFE (single-cell identity definition using random forests and recursive feature elimination, pronounced 'surf') with the goal of easily generating interpretable gene lists that can accurately distinguish observations (single-cells) by their features (genes) given a class of interest. scRFE is an algorithm implemented as a Python package that combines the classical random forest method with recursive feature elimination and cross validation to find the features necessary and sufficient to classify cells in a single-cell RNA-seq dataset by ranking feature importance. The package is compatible with Scanpy, enabling a seamless integration into any single-cell data analysis workflow that aims at identifying minimal transcriptional programs relevant to describing metadata features of the dataset. We applied scRFE to the Tabula Muris Senis and reproduced commonly known aging patterns and transcription factor reprogramming protocols, highlighting the biological value of scRFE's learned features.

Author summary

scRFE is a Python package that combines the classical random forest algorithm with recursive feature elimination and cross validation to find the features necessary and sufficient to classify cells in a single-cell RNA-seq dataset by ranking feature importance. scRFE was designed to enable straightforward integration as a part of any single-cell data analysis workflow that aims at identifying minimal transcriptional programs relevant to describing metadata features of the dataset.

Introduction

Single-cell RNA sequencing (scRNA-seq) facilitates newfound insights into the long withstanding challenge of relating genotype to phenotype by resolving the tissue composition of an organism at single-cell resolution [1]. Moreover, scRNA-seq enables the delineation of the constituents contributing to the homeostasis of the organism by understanding the widely varying functional heterogeneity of every cell [2].

Tremendous efforts have been put into developing computational methods for cell type classification. Cell types have historically been characterized by morphology and

functional assays, but one of the main goals of single-cell transcriptomics is the ability to define cell identities and their underlying regulatory networks. As an example, transcription factors (TFs) have the ability to mediate cell fate conversions [3], which could greatly address the need for specific cell types in regenerative medicine and research [5, 8]. A panoply of biological experiments has been performed to further validate and understand the role that transcription factors play in determining cell subpopulations, demonstrating that it is possible to transform the fate of a cell with a unique group of TFs [6, 7]. Furthermore, several computational methods exist for finding particular sets of TFs [8]. However, the models often rely on an exceptionally well characterized target cell type population and a 'background population' [8]. They also face the challenge of having noisy data and being sensitive to technical differences in experimentation methods, causing them to return a large list of TFs to be considered, hampering their usability by bench scientists [8].

Here we present scRFE (single-cell identity definition using random forests and recursive feature elimination, pronounced 'surf'), an algorithm and Python package that combines the classical random forest algorithm with recursive feature elimination and cross validation to find the features necessary and sufficient to classify cells in a single-cell RNA-seq dataset by ranking feature importance. A random forest is an ensemble of uncorrelated decision trees, where for a given split in a given tree, a random subset of the feature space is considered [11]. Random forests are ideal for single-cell data given their effective performance on sparse, high dimensional data with collinear features and straightforward interpretability [11].

The goal of scRFE is to produce interpretable gene lists that accurately distinguish cells between all labels within a specific class of interest by ranking feature importance. The out-of-the-box compatibility of scRFE with Scanpy [9] enables seamless integration as a part of any single-cell data analysis workflow that aims at identifying minimal sets of genes relevant to describing metadata features of the dataset. Thus, scRFE has potential applications not only in finding transcription factors suitable for reprogramming analysis, but more generally in identifying gene importances for a given subpopulation of cells.

Results

scRFE Overview

scRFE (Fig. 1) takes as input an AnnData object and a class of interest. Next, the algorithm iterates over the set of labels (specific values) in the class of interest (category to split observations by). scRFE utilizes a random forest with recursive feature elimination and cross validation to identify each feature's importance for classifying the input observations. Downsampling is performed separately for each label in the class of interest, ensuring that the groups are balanced at each iteration. In this work, each observation is a single-cell, each feature is a gene, and the data is the expression level of the genes in the entire feature space. In order to learn the features to discriminate a given cell type from the others in the dataset, scRFE was built as a one versus all classifier. Recursive feature elimination was used to avoid high bias in the learned forest and to address multicollinearity [26]. In this way, scRFE learned the features for a given label in the class of interest and returned the mean decrease in Gini score for the top learned features. This is a numerical measure of the feature's importance in the entire model by measuring node purity [16]. Recursive feature elimination removes the bottom 20 % of features with the lowest mean decrease in Gini score at each iteration. We incorporated k-fold cross validation (default = 5), to minimize variance and reduce the risk of overfitting [12]. scRFE returns the Gini scores ranking the top features for each label in the class of interest. These selected features are then considered the ones

necessary and sufficient to describe the class of interest due to the implementation of variable elimination and cross validation.

scRFE identifies concordance between FACS and Droplet techniques in Tabula Muris Senis

We have applied scRFE to the Tabula Muris Senis (TMS) dataset [13]. TMS is a comprehensive atlas of cells from 23 tissues over the murine lifespan. The dataset's deep characterization of phenotype and physiology allowed us to explore TF reprogramming protocols [14], as well as aging signatures [15], on the scale of hundreds of thousands of cells.

We started by investigating the results from the top performing genes for both the 10X/droplet and Smart-Seq2/FACS datasets from TMS, indicating that scRFE's performance is often not greatly affected by such technical aspects of experimentation (Fig. 2, Supplementary Table 1). Looking at tissue specific results (Supplementary Table 2), we saw that scRFE run on the mammary gland specific FACS and droplet objects with age as the class of interest outputs $\geq$73% of shared genes for the 3 month mice (Fig. 2a). scRFE run on the lung specific FACS and droplet objects with age as the class of interest outputs $\geq$76% of shared features for the 18 month old mice (Fig. 2b). This finding remained true when scRFE was run on cell types (Supplementary Table 3), as the FACS and droplet liver specific objects shared $\geq$84% of selected features for hepatocyte cells (Fig. 2c). However, perfect overlap between the two methods is still not achieved due to the inherent experimental differences between the 10X/droplet and Smart-Seq2/FACS methods, with the Smart-Seq2/FACS based approach generating full-length cDNAs with a greater ability to detect rare cells, compared to the 10X/droplet method, an end-to-end solution with high cell capture efficiency [20]. A future interesting approach could be combining the independent datasets to create a more robust characterization of the class of interest.

scRFE learns biologically consistent transcriptional programs

Specific sets of transcription factors have the potential to determine a cell's fate [3]. Our method finds concordance with known marker genes for different cell ages, and ontologies coming from a variety of tissues. Not only was scRFE able to recapitulate sets of genes for identifying TFs (Table 1), but the numerical feature rankings also followed existing literature. The features with high importance had often already been experimentally validated and/or computationally derived as shown by the colored genes in Fig. 3 [3, 17–19].

Top feature importances reveal gene ontology patterns by age

The aging phenotype is associated with notable transcriptional changes [27], and we used scRFE to learn the top genes associated with each age. Analysis on the global FACS dataset subsetted for TFs as features and with observations split by age revealed that 574 genes were shared between the 3m, 18m, and 24m labels (Supplementary Table 4). When looking at age specific genes, we found that Ppp1c is the gene with the highest mean decrease in Gini score that is specific to the 24m mice. Ppp1cb has been found to be associated with aging and mortality [28]. The second highest ranked gene specific to only the 24m label is Sfpq. Previous literature states that Sfpq has been linked to diseases such as familial amyotrophic lateral sclerosis (ALS) [4]. Additionally, mutations in Sfpq or the dysregulation of the Sfpq-dependent energy metabolism may be connected to sarcopenia, or what is commonly known as biological aging [10].

We then performed GSEA analysis [24, 25] and found that the selected features exclusive for 3m are associated with regular biological processes, such as amino-acid metabolism and cell adhesion (Figs 4a). When looking at either 18m (Fig. 4b) or 24m (Fig. 4c) selected features, excluding the 3m selected features, we observed a dramatic switch towards a disease signature. For the 18m and 24m labels, the top KEGG pathways are associated with both infections and neurological diseases, reiterating scRFE's ability to recover biologically relevant signatures in the form of minimal transcriptional sets.

Conclusion

scRFE is a robust and reproducible solution to the problem of identifying cell type marker genes and biological transcriptional programs in single-cell RNA sequencing. The code is available from GitHub (https://github.com/czbiohub/scRFE) and it has been included in PyPI to facilitate its integration with current workflows.

When aplied to the TMS dataset, scRFE is capable of identifying concordance between droplet and FACS techniques. scRFE also identifies biologically consistent transcription factor reprogramming methods. This highlights its robustness as a method to contribute towards identifying reprogramming protocols and potentially to find novel ones.

scRFE has applications of finding new conclusions while mapping genotypes to phenotypes, especially regarding aging processes. When applied to aging datasets, scRFE produces results that match commonly known aging patterns, highlighting developmental pathways in the younger organisms and moving towards canonical aging and disease related pathways in the older organisms. Future work may include utilizing scRFE to discover genes and pathways biologically inherent to young versus old organisms, furthermore helping to understand aging.

Acknowledgments

We thank Jingyi Wei, Jim Karkanias, and Steve R Quake for thoughtful discussions. S.K.V. is supported by a NSF Graduate Research Fellowship, the Benchmark Stanford Graduate Fellowship, and the Stanford ChEM-H Chemistry Biology Interface (CBI) training program. This work was funded by the Chan Zuckerberg Biohub.

Author's contributions

M.P. designed and implemented the method, packaged the code for PyPI, performed the computational analysis and co-wrote the manuscript. S.V. designed and implemented the method and co-wrote the manuscript. S.W. packaged the code for PyPI. A.O.P. conceptualized the study, designed and implemented the method, performed the computational analysis and co-wrote the manuscript. All authors read the final manuscript.

Methods

Implementation

scRFE is built in Python using methods from scikit-learn [21] and Scanpy [9] libraries. We chose to use Python over R given its compatibility with Scanpy/AnnData objects. For R users, we recommend using the varselRF package [22]; example of applications to

single-cell datasets in the Tabula Muris [14]. scRFE's direct compatibility with Scanpy makes it easy to integrate as a part of any single-cell data analysis workflow interested in identifying minimal sets of genes relevant to describing metadata features of the dataset.

Input

scRFE expects as input *adata*, an h5ad/AnnData object. The $classOfInterest$ must be one of the metadata variables present in the *adata.obs* of the AnnData object. scRFE iterates over the set of labels in this class and splits the data in two groups, label A (current label) and label B(rest). To have a more faithful representation of the data while having balanced A and B groups, scRFE downsamples the observations in B to match the number of observations in A. While doing so, it maintains the representativeness of each other label in *rest* (please refer to GitHub for the detailed implementation).

Data Preprocessing

scRFE removes all cells with missing assignments in the class of interest (e.g. label being None or NaN) before iterating over the set of labels in the class of interest.

How to run scRFE

scRFE receives *adata*, an AnnData object (.h5ad), as input. The data is used as it is in the $.X$ slot; we did not notice significant differences between using raw or normalized data, but the user should evaluate this on their own dataset. scRFE also receives as input $classOfInterest$, a column of the *.obs* slot of adata that is the category to split observations by.

scRFE begins by removing observations that had a missing (NaN) annotation for the class of interest from the inputted AnnData object. A statement is printed, noting that this action is happening.

Next, scRFE splits the data for one versus all classification. For every unique label in the specified $classOfInterest$, scRFE iterates through each observation and distinguishes between observations that match the given label of interest in the class, versus observations with other labels. scRFE downsamples to the smallest category, meaning it randomly selected the same number of observations that the $classofInterest$ had from the rest of the observations, allowing for a balanced one versus rest comparison.

scRFE sets the X value of the random forest to $adata.X$, the matrix with observations and variables. The y value of the random forest is the labels created from the previous step. Then, scRFE uses sklearn to create a random forest (clf) and recursive feature eliminator ($selector$). The random forest splits observations (cells) based on what features (genes) are expressed until all observations are classified. Feature importance is calculated using the mean decrease in Gini score, a numerical measurement of node impurity [16].

At each iteration, the selector removes the bottom percent (default = 20%) of features with the lowest mean decrease in Gini importances to avoid high bias and to address multicollinearity [26]. The selector also incorporates k-fold cross validation (default = 5) to minimize variance and reduce the risk of overfitting [12]. Thus, the data is split randomly into 5 folds, where at each iteration, 4 folds are trained. This combination of recursive feature elimination and cross validation allows for not only a robust model to be learned, but an elegant and concise model with only the necessary and sufficient features to describe the data. For the most part, we used scRFE's default parameters. The default number of trees ($nEstimators$) is set to 1,000, the default step size ($Step$) to 0.2, and the default k-fold cross validation metric (Cv) to 5. All of these parameters are tunable and should be adjusted given a particular dataset (Table 2).

Once the top features and corresponding mean decrease in Gini scores are found, scRFE puts them in a sorted pandas DataFrame. This process happens for each unique label within $classOfInterest$. The output is a pandas DataFrame with the top selected features and their Gini scores for every label in the $classOfInterest$.

Datasets and Gene Lists

Tabula Muris Senis was downloaded from FigShare (processed official FACS and droplet AnnData objects found at https://figshare.com/projects/Tabula_Muris_Senis/64982). We ran scRFE on the TMS AnnData objects using the normalized data. Users may choose to remove cells or genes that do not meet a minimum requirement, or to use raw data. We did not notice substantial differences when using raw or normalized data. The list of GO mouse transcription factors list was defined as the 1,140 genes annotated by the Gene Ontology term 'DNA binding transcription factor activity', downloaded from the Mouse Genome Informatics database (http://www.informatics.jax.org/mgihome/GO/project.shtml).

Running scRFE with different values for classOfInterest

We ran scRFE on the Tabula Muris Senis processed official FACS and droplet AnnData objects on a variety of combinations of tissues, cell types, and ages. We used scRFE's default parameters and sweeped the number of trees (scRFE argument nEstimators) to assess model performance. We observed how the results changed with different values of nEstimators to determine 1,000 to be our default parameter (Fig 5).

A detailed list of what analysis we ran can be found in the Supplementary Tables.

gProfiler and GSEA analysis

We ran scRFE on the global FACS dataset with age as the class of interest. For the ages 3m, 18m, and 24m, we took the lists of features selected and inputted them to gProfiler [23] (Supplementary Table 5). This returned a dataframe highlighting significant pathways (sorted by p-value) for different gene ontologies based on the inputted gene lists (Supplementary Table 6). The intersection of the lists of features selected was analyzed using the GSEA MGSig Database [24, 25] and the results are summarized in Fig.4.

References

1. Hwang, B., Lee, J. H. Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. EXP MOL MED. 2018; 50(8), 96.

2. Saliba, A.E., Westermann, A.J., Gorski, S. A. Vogel, J. Single-cell RNA-seq: advances and future challenges. Nucleic Acids Res. 2014; 42(14, 18), 8845-8860.

3. Takahashi, K. Yamanaka, S. A decade of transcription factor-mediated reprogramming to pluripotency. Nat Rev Mol Cell Biol. 2016; 17, 183–193.

4. Thomas-Jinu, S., Gordon, P. M., Fielding, T., et al. Non-nuclear Pool of Splicing Factor SFPQ Regulates Axonal Transcripts Required for Normal Motor Development. Neuron. 2017; 94(2), 322–336.

5. Xu J., Du Y. Deng H. Direct lineage reprogramming: strategies, mechanisms, and applications. Cell Stem Cell. 2015; 16(2), 119-134.

6. Zhu, S., Wang, H. Ding, S. Reprogramming fibroblasts toward cardiomyocytes, neural stem cells and hepatocytes by cell activation and signaling-directed lineage conversion. Nat Protoc. 2015; 10, 959–973.

7. Okawa, S., del Sol, A. A general computational approach to predicting synergistic transcriptional cores that determine cell subpopulation identities. Nucleic Acids Res. 2019; 47(7) 3333–3343.

8. Guerrero-Ramirez, G. I., Valdez-Cordoba, C. M., Islas-Cisneros, J. F., Trevino, V. Computational approaches for predicting key transcription factors in targeted cell reprogramming (Review). Molecular medicine reports. 2018; 18(2), 1225–1237.

9. Wolf, F., Angerer, P. Theis, F. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018; 19, 15.

10. Hosokawa, M., Takeuchi, A., Tanihata, J., Iida, K., Takeda, S., Hagiwara, M. Loss of RNA-Binding Protein Sfpq Causes Long-Gene Transcriptopathy in Skeletal Muscle and Severe Muscle Mass Reduction with Metabolic Myopathy. iScience. 2019; 13, 229–242

11. Gareth, J., Witten, D., Hastie, T., Tibshirani, R. An introduction to statistical learning. Springer. 2013; 112.

12. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Stanford CS Dept. 1995.

13. Almanzar, N., Antony, J., Baghel, A.S. et al. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. Nature. 2020; 583, 590–595.

14. Schaum, N., Karkanias, J., Neff, N.F. et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature. 2018; 562, 367–372.

15. Kimmel, J. C., Penland, L., Rubinstein, N. D., Hendrickson, D. G., Kelley, D. R., Rosenthal, A. Z. Murine single-cell RNA-seq reveals cell-identity- and tissue-specific trajectories of aging. Genome research. 2019; 29(12), 2088–2103.

16. Muchai, E., Odongo, L. Comparison of crisp and fuzzy classification trees using gini index impurity measure on simulated data. ESJ. 2014; 10, 1857 – 7881.

17. Yang, N., Zuchero, J., Ahlenius, H. et al. Generation of oligodendroglial cells by direct lineage conversion. Nat Biotechnol. 2013; 31, 434–439.

18. Davis, R.L., Weintraub, H., Lassar, A.B. Expression of a single transfected cDNA converts fibroblasts to myoblasts. Cell. 1987; 51(6), 987-1000

19. Ring, K.L., Tong, L.M., Balestra, M.E., et al. Direct reprogramming of mouse and human fibroblasts into multipotent neural stem cells with a single factor. Cell Stem Cell. 2012; 11(1), 100-9

20. Seel, P., Lum, J., Chen, J., Ginhoux, F. A Single-Cell Sequencing Guide for Immunologists Front. Immunol. 2018; 9, 2425.

21. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. Scikit-learn: Machine Learning in Python. J MACH LEARN RES. 2014; 42(14, 18): 8845-8860.

22. Diaz-Uriarte, R. GeneSrF and varSelRF: a web-based tool and R package for gene selection and classification using random forest. BMC Bioinformatics. 2007; 8, 328.

23. Reimand, J., Kull, M., Peterson, H., Hansen, J., Vilo, J. g:Profiler–a web-based toolset for functional profiling of gene lists from large-scale experiments. Nucleic Acids Res. 2007; 35, W193-W200.

24. Subramanian, A., Tamayo, P., Mootha, V.K., et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. PNAS. 2005; 102(43),15545-15550.

25. Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., Mesirov, J.P. Molecular signatures database (MSigDB) 3.0 Bioinformatics. 2011; 27(12), 1739-40

26. Baptiste, G., Michel, B., Saint-Pierre, P. Correlation and Variable Importance in Random Forests. Statistics and Computing. 2016; 27(3), 659–678.

27. Schaum, N., Lehallier, B., Hahn, O. et al. Ageing hallmarks exhibit organ-specific temporal signatures. Nature. 2020; 583, 596–602.

28. Bult, C.J., Blake, J.A., Smith, C.L., Kadin, J.A., Richardson, J.E. Mouse Genome Database (MGD) Nucleic Acids Res. 2019; 8(47), D801-D806.

Figure Legends

Fig. 1: scRFE Overview.

Given an input AnnData object and specified value for 'classOfInterest,' or the class to split the data by, scRFE performed one vs all classification. scRFE downsampled to the smallest category to provide an equal sized comparison for every label within the classOfInterest. For each label, scRFE built a random forest with recursive feature elimination, where the bottom percent of features were removed at each iteration. scRFE also performed k-fold cross validation. scRFE outputted a pandas DataFrame with columns corresponding to each label within the classOfInterest, storing the selected features and their respective mean decrease in Gini importances.

Fig. 2: scRFE finds consistency between FACS and Droplet Methods.

a) scRFE ran with default parameters on the mammary gland specific objects for FACS and Droplet with classOfInterest = 'age.' Here, we selected the results for 3m. Majority of the selected genes for both datasets were shared. The two results overlapped $\geq 73\%$ of the time with 157 shared genes, 56 droplet specific genes, and 57 FACS specific genes.
b) scRFE run with default parameters on the lung specific objects for FACS and Droplet with classOfInterest = 'age.' Here, we selected the results for 18m. Majority of the selected genes for both datasets were shared. The two results overlapped $\geq 76\%$ of the time with 163 shared genes, 50 droplet specific genes, and 51 FACS specific genes.
c) scRFE run with default parameters on the liver specific objects for FACS and Droplet with classOfInterest = 'cell_ontology_class.' Here, we selected the results for hepatocytes. Majority of the selected genes for both datasets were shared. The two results overlapped $\geq 84\%$ of the time with 360 shared genes, 60 droplet specific genes, and 65 FACS specific genes.

Fig. 3: scRFE learns biologically consistent transcriptional programs

a) Sox10, Olig2, Zfp536 are necessary to reprogram cells into oligodendrocytes [17]. scRFE found all three transcription factors in the top ten selected features as shown in green.
b) 'Synergistic identity core' of transcription factors for pericyte cells in the murine liver includes Lmcd1, Foxf2, Tbx2, Nr1h3, and Nr2f1 [7]. scRFE found four out of these five TFs in the top six selected features as shown in green.
c) 'Synergistic identity core' of transcription factors for NK cells in the murine liver includes Tbx21, Runx3, Gata3, Ikzf3, and Tcf7 [7]. scRFE found three of these in the top 22 selected features as shown in green.

Fig. 4: KEGG pathway analysis for scRFE age results.

a) Statistically significant KEGG pathways for the scRFE selected features for 3m excluding 18m and 24m.
b) Statistically significant KEGG pathways for the scRFE selected features for 18m excluding 3m.
c) Statistically significant KEGG pathways for the scRFE selected features for 24m excluding 3m.

Fig. 5: scRFE run on global FACS object with classOfInterest = age and different values for nEstimators. 18m results.

Table Legends

Table 1

Examples of scRFE identifying biologically consistent transcription programs.

Table 2

scRFE parameters with default values and descriptions.

Supplementary Tables

Note: scRFE default parameters were used unless stated otherwise.

Supplementary Table 1

FACS and droplet tissue and age specific objects subsetted for TFs as features with classOfInterest = cell type.

Supplementary Table 2

FACS and droplet tissue specific objects subsetted for TFs as features with classOfInterest = age. nEstimators = 5000.

Supplementary Table 3

FACS and droplet tissue specific objects subsetted for TFs as features with classOfInterest = cell type.

Supplementary Table 4

FACS and droplet global objects subsetted for TFs as features with classOfInterest = age. nEstimators = 50, 100, 500, 1000, 2000, 5000.

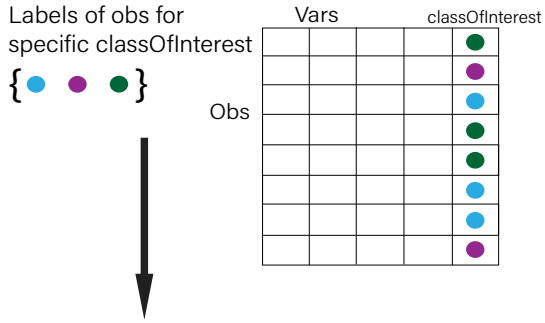
Supplementary Table 5

gProfiler Results for Supplementary Table 2 lists.

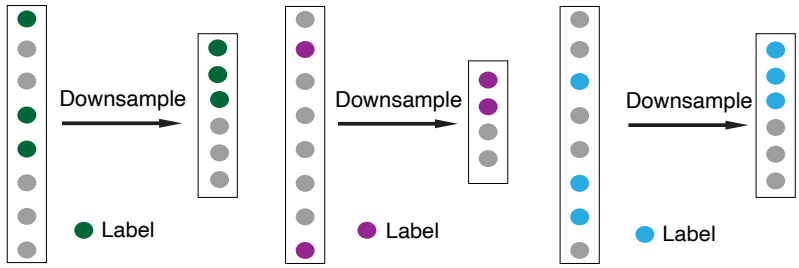
Supplementary Table 6

FACS global object with classOfInterest = age.

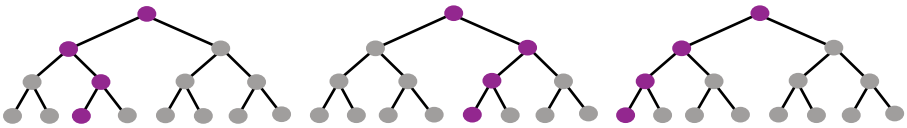
Read in AnnData Object



One vs All Classification For classOfInterest



For Each Label:



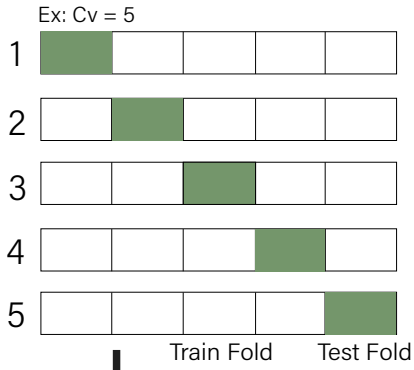
Random Forest

Ex: nEstimators = 3

Recursive Feature Elimination

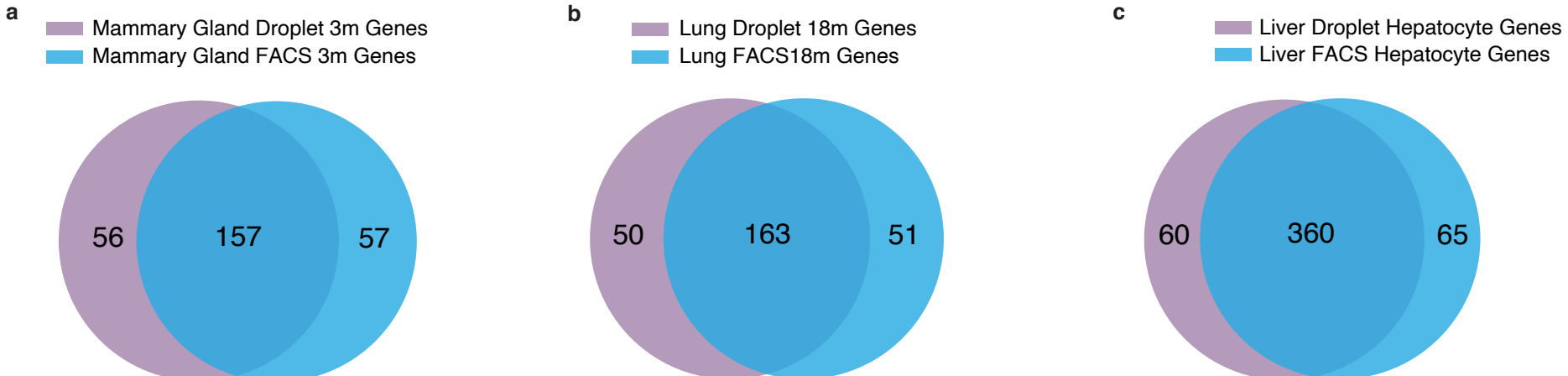
At each iteration, remove the bottom % of features with the lowest mean decrease in Gini scores

K-Fold Cross Validation

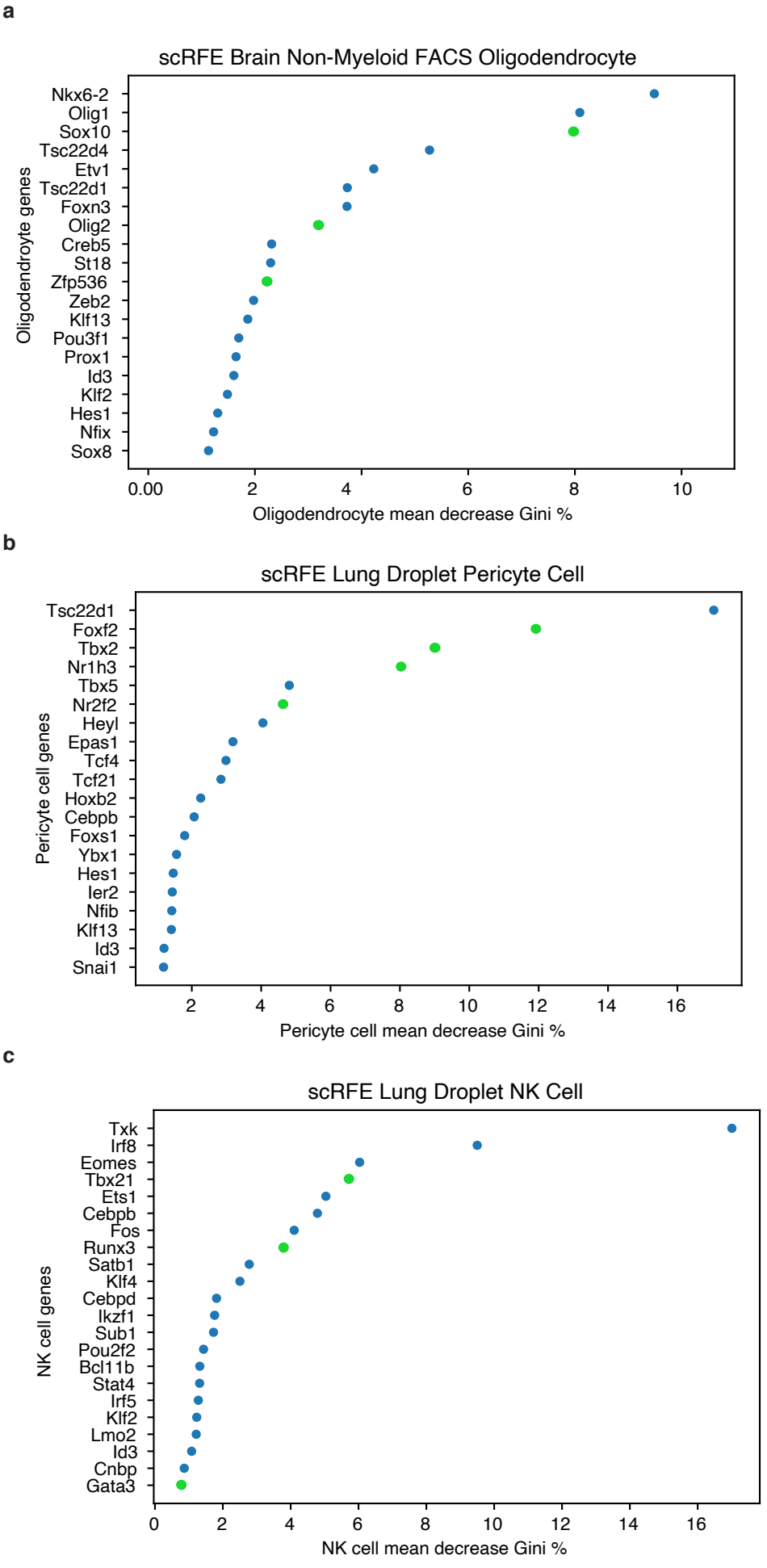


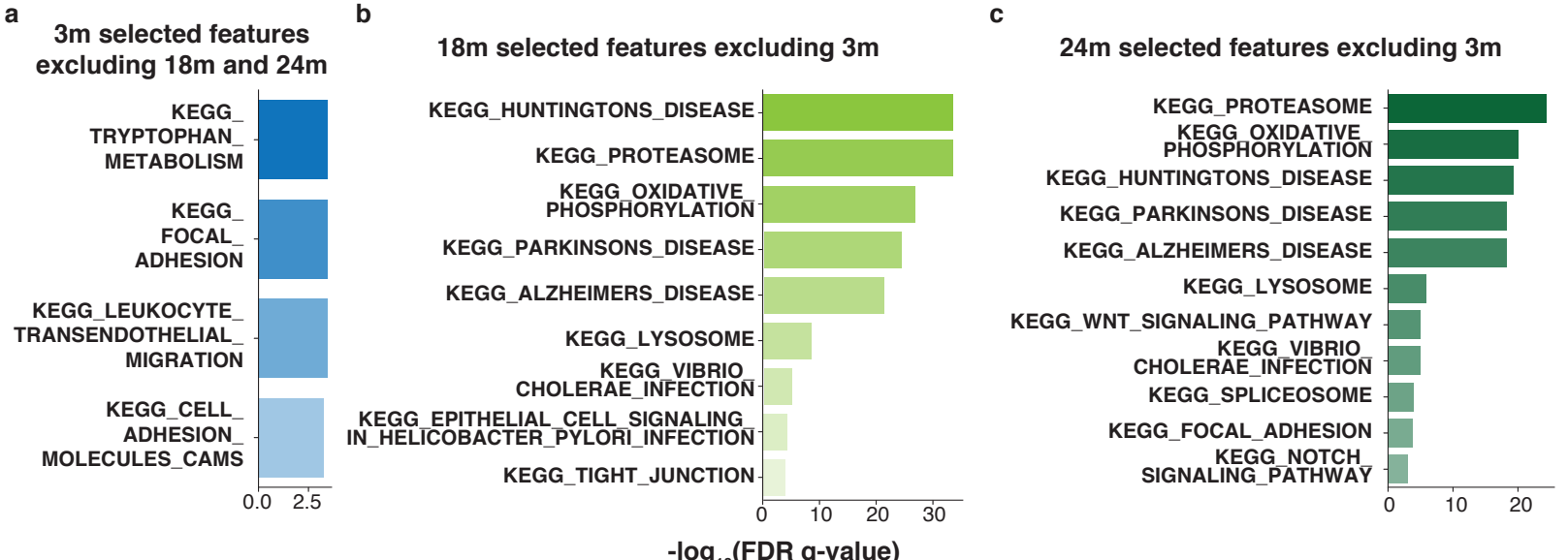
Return Results DataFrame

Label 1	Label 1 Gini	Label 2	Label 2 Gini	Label 3	Label 3 Gini


a
Mammary Gland Droplet 3m Genes
Mammary Gland FACS 3m Genes
56
157
57
b
Lung Droplet 18m Genes
Lung FACS18m Genes
50
163
51
c
Liver Droplet Hepatocyte Genes
Liver FACS Hepatocyte Genes
60
360
65

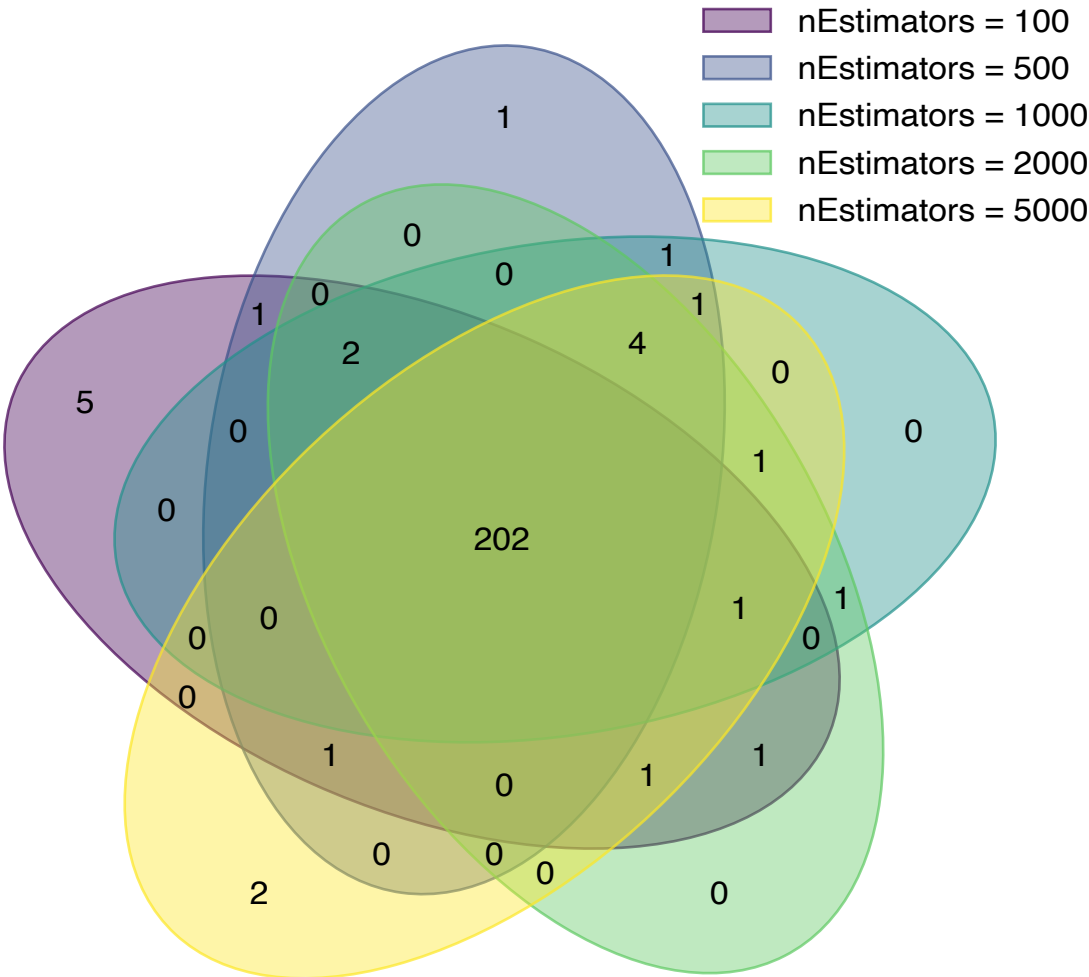
Cell Type	Transcription Factors *Bolded are ones scRFE found in top 25 selected features	Reference
Pericyte Cells	Lmcd1, **Foxf2, Tbx2, Nr1h3, Nr2f1**	Okawa, S., et al. Nucleic Acids Res 2019.
NK Cells	**Tbx21, Runx3, Gata3,** Iksf3, Tcf7	Okawa, S., et al. Nucleic Acids Res 2019.
Oligodendrocytes	**Sox10, Olig2, Zfp536**	Yang, N., et al. Nat Biotechnol 2013.
Skeletal Muscle Cells	**Myod1**	Davis, et al. Cell 1987.
Neuronal Stem Cells	**Sox2**	Ring, et al. Cell Stem Cell 2012.


a
scRFE Brain Non-Myeloid FACS Oligodendrocyte
Oligodendroyte genes
Nkx6-2
Olig1
Sox10
Tsc22d4
Etv1
Tsc22d1
Foxn3
Olig2
Creb5
St18
Zfp536
Zeb2
Klf13
Pou3f1
Prox1
Id3
Klf2
Hes1
Nfix
Sox8
0.00
2
4
6
8
10
Oligodendrocyte mean decrease Gini %
b
scRFE Lung Droplet Pericyte Cell
Pericyte cell genes
Tsc22d1
Foxf2
Tbx2
Nr1h3
Tbx5
Nr2f2
Heyl
Epas1
Tcf4
Tcf21
Hoxb2
Cebpb
Foxs1
Ybx1
Hes1
Ier2
Nfib
Klf13
Id3
Snai1
2
4
6
8
10
12
14
16
Pericyte cell mean decrease Gini %
c
scRFE Lung Droplet NK Cell
NK cell genes
Txk
Irf8
Eomes
Tbx21
Ets1
Cebpb
Fos
Runx3
Satb1
Klf4
Cebpd
Ikzf1
Sub1
Pou2f2
Bcl11b
Stat4
Irf5
Klf2
Lmo2
Id3
Cnbp
Gata3
0
2
4
6
8
10
12
14
16
NK cell mean decrease Gini %


a
3m selected features excluding 18m and 24m
KEGG_ TRYPTOPHAN_ METABOLISM
KEGG_ FOCAL_ ADHESION
KEGG_LEUKOCYTE_ TRANSENDOTHELIAL_ MIGRATION
KEGG_CELL_ ADHESION_ MOLECULES_CAMS
0.0
2.5
b
18m selected features excluding 3m
KEGG_HUNTINGTONS_DISEASE
KEGG_PROTEASOME
KEGG_OXIDATIVE_ PHOSPHORYLATION
KEGG_PARKINSONS_DISEASE
KEGG_ALZHEIMERS_DISEASE
KEGG_LYSOSOME
KEGG_VIBRIO_ CHOLERAE_INFECTION
KEGG_EPITHELIAL_CELL_SIGNALING_ IN_HELICOBACTER_PYLORI_INFECTION
KEGG_TIGHT_JUNCTION
0
10
20
30
-log10(FDR q-value)
c
24m selected features excluding 3m
KEGG_PROTEASOME
KEGG_OXIDATIVE_ PHOSPHORYLATION
KEGG_HUNTINGTONS_DISEASE
KEGG_PARKINSONS_DISEASE
KEGG_ALZHEIMERS_DISEASE
KEGG_LYSOSOME
KEGG_WNT_SIGNALING_PATHWAY
KEGG_VIBRIO_ CHOLERAE_INFECTION
KEGG_SPLICEOSOME
KEGG_FOCAL_ADHESION
KEGG_NOTCH_ SIGNALING_PATHWAY
0
10
20

scRFE parameter	Default	Description
adata	anndata object	.h5ad input file
classOfInterest	str	class to classify obs by
nEstimators	int, 1000	# of trees in the forest
randomState	int, 0	controls random number being used
min_cells	int, 15	min number of cells in a given class to downsample
keep_small_categories	bool, True	whether to keep classes with small #observations
nJobs	int, -1	# jobs to run in parallel
oobScore	bool, True	whether to use out-of-bag samples
Step	float, 0.2	fraction of features to remove at each iteration
Cv	int, 5	determines k-fold cv strategy
verbosity	bool, True	whether to print statements

Detailed descriptions of our parameters can be found at *scrfe.readthedocs.io/en/latest/*


nEstimators = 100
nEstimators = 500
nEstimators = 1000
nEstimators = 2000
nEstimators = 5000
1
0
0
1
0
1
1
2
4
0
5
0
0
1
0
202
1
1
0
0
0
0
1
1
0
1
0
0
0
2
0