CS-433 Machine Learning Project 2

Matthias Minder, Zora Oswald, Silvan Stettler

Abstract—**Single cell RNA sequencing (scRNA-seq) provides new data to gain insights into cellular functionalities. We compare several classifiers trained to predict whether a given cell is a stem cell based on scRNA-seq data, before finally selecting a best-performing classifier on independent data. The assessed classifiers were based on different machine learning techniques and constructed with different feature transformations. In particular, incorporation of gene interactions yielded an improved performance, presenting a promising technique for processing scRNA-seq data in general.**

Introduction

The recently developed method of single cell RNA sequencing (scRNA-seq) allows to measure the amount of RNA from a specific gene at the single cell resolution. This gives valuable insight into the properties and cellular function of single cells in a whole population of cells. High throughput methods allow to analyze thousands of cells simultaneously. scRNA-seq results in so-called read-count matrices (RCMs) which indicate the expression level of a given gene in a given cell.

The emergence of scRNA-seq has led to the discovery of many new cell populations based on their gene expression profile. The question thus naturally arises whether it would be possible to predict cell types using a machine learning approach. To this effect, classifiers can be trained on various publicly available data sets to infer cell types on a new data set [1]. Of special interest is the detection and *de novo* discovery of stem cells in tissues for which no stem cell population has been characterized. It is still unknown whether there exist specific biomarkers for multipotent stem cells allowing to identify them across tissues, which makes this problem attractive to solve.

However, several challenges are associated with working with scRNA-seq data. The three most predominant issues are the following: Differences in protocols and procedures give rise to batch effects, leading to substantial overall differences between datasets of different sources. Secondly, cell-type annotation is based on clusters of the dataset itself and thus depends on the analysis and interpretation of the data. This may lead to biases when basing analysis on the annotated clusters. Thirdly, the obtained data is perturbed by so called drop-out events, where the read-counts for a given gene is measured to be zero, even if there were a signal. The result is that RCMs are very sparse.

Additional issues arise from a machine learning perspective: Since the expression of different genes can be functionally related or controlled by the same transcriptional regulatory program, some features are highly correlated. The scRNA-seq data can thus be thought of as being on a sub-dimensional manifold. However, due to the noisy nature of the data, identifying highly correlated features is difficult. An accurate representation of this manifold is key for obtaining stable results that generalize well to other data sets in order to overcome batch-effects. Moreover, a good such representation also reduces the impact of drop-outs: If several genes are simultaneously expressed for a given biological process, and a subset of these aren't detected due to drop-outs, the other genes could correct this. Naturally, such an approach doesn't work when the drop-out rate is too high. Overall, this nature of scRNA-data suggests that the application of dimensionality reduction prior to method training will yield results that perform better on new, unrelated datasets.

Within this report, we train different stem cell classifiers on multiple published scRNA-seq datasets. The predictive performance is assessed using data from different sources in order to finally determine the best-performing classifier. The presented classifiers are based on different, well-established machine learning methods and were trained on data subjected to different data transformations and dimensionality reduction techniques.

Methods

A. Experimental Setup

For the entire analysis, 17 publicly available scRNA-seq datasets from mature *mus musculus* were used [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. These data sets were split into three sets, set A (eleven data sets), set B (three data sets) and set C (four data sets). Two types of classifiers were trained, which will be called basic and nested classifiers. The basic classifiers were trained on set A, and their predictive performance was assessed in terms of the area under the receiver operating characteristic curve (AUC) on the set B and C. The nested classifiers were trained as follows: Using the basic classifiers, predictions were generated for the data in set B. Then, the resulting predictions by all basic classifiers were used as training set for a second classifier, whose performance was assessed and compared with the basic classifiers on set C. By constructing the two-layer nested classifiers, we hoped to be able to obtain a final model which is less prone to batch effects. Different basic classifiers were trained on a total of eight data transformations described below, six of which are network-based.

Classifier type	Set A	Set B	Set C
Basic	Training	Testing	Testing
Nested	-	Training	Testing

Table I
USE OF DATASETS

B. Dataset Preparation

The data as well as the cell type annotations were taken from the original papers. The data was then processed to have only 13'587 genes remaining. Missing values were set to zero. Furthermore, the data was sampled to have exactly 50% clearly identified stem and 50% non-stem cells. For a complete explanation of the dataset preparation, refer to supplement -A. We normalized the expression values to transcripts per million before transforming the values with $ln(1+x)$, as is standard procedure for scRNA-seq data. All subsequent transformations and training steps were based on data in this format.

C. Gene Network Construction

Taking into account external information about the underlying structure is a way to reduce the influence of set specific batch effects. To this effect, we included a protein interaction network, which we then use as topology to process the expression signal. Protein interaction networks are representations of the physical and biochemical interaction between proteins in an organism. The protein interaction network was converted to a gene network by replacing proteins with the genes that encode them.

The protein interaction data that was used is available in the STRING database (www.string-db.org) [18]. The weight of the edges between individual proteins, which form the nodes, are a measure of how confidently an interaction can be judged to be true. Mapping of genes to their proteins was done using the Ensembl database for *mus musculus* (www.ensembl.org) [19]. For the creation of our network, only the proteins that could be associated to an expressing gene were kept. Self-loops that resulted from genes whose expression produces multiple proteins and isolated nodes (no link to another node) were removed, yielding a network with 20'330 genes forming nodes that are connected by 11'856'336 edges. It has to be mentioned that the gene network is constructed based on the entire dataset that was available from the sources mentioned above. Contrary to the analysed data, the scope of the network does not limit itself to the genes found in a particular type of cells, such as tissue.

The process of creating the gene network is schematically shown in Figure 1. In the case that the replacing of the proteins produces multiple edges between genes, a single edge with a weight equal to the sum of all the individual edge weights is created instead.

The original score s_{ij} of a link between nodes i and j was used to compute an equivalent distance, where high scores result in a short distance. Subsequently, new link weights W_{ij} were set using a Gaussian kernel.

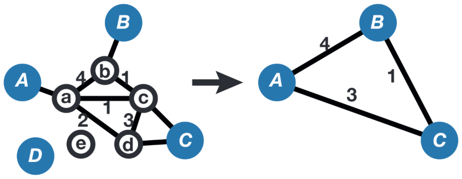

Figure 1. Schematic of network creation. The genes *A, B, C, D* replace the proteins which they express. Proteins that cannot be associated to a gene or vice versa are removed. Resulting self-loops, asi in the case of replacing protein c and d with gene *C* are also removed. Moreover, between gene *A* and *C* two edges of weight 1 and 2 respectively would be created. These edges are summarized in one edge of weight 3.

$$W_{ij} = \exp\{-\frac{1}{2}\frac{1}{(s_{ij}\sigma)^2}\} \tag{1}$$

D. Feature Creation through Graph Signal Processing

The scRNA-sec data can be transformed into a graph signal by assigning the count of expressed genes to the corresponding gene in the network, yielding a signal $\boldsymbol{x} \in \mathbb{R}^N$, where N is the total number of nodes. Genes whose expression was not present as a feature in the scRNA-sec data are assigned zero.

After mapping the data onto the resulting graph, graph signal processing (GSP) can be leveraged to find a lower-dimensional manifold that embeds the original data, based purely on the constructed network. The two methods that were applied take advantage of the fact that the eigendecomposition of the $N \times N$ graph Laplacian matrix $\boldsymbol{L}$ forms the equivalent of a Fourier basis on a graph. Given the eigendecomposition $\boldsymbol{L} = \boldsymbol{U\Lambda U}^T$, $\boldsymbol{U} = [\boldsymbol{u}_1 ... \boldsymbol{u}_N]$ are the N Fourier basis vectors and the N eigenvalues $\boldsymbol{\Lambda}$ represent the corresponding frequencies [20].

Graph sampling (GS) is a way to identify genes which are of particular importance to the network. In particular, it identifies nodes where signal energy is the most concentrated for frequencies in the range (f_{min}, f_{max}) [21]. The *local graph coherence* at node i of order k is defined as the square of the i-th entry of the k-th eigenvector u_{ik} of the Laplacian, which is a measure for how localized the first k Fourier modes are on node i [20]. By choosing the first K indices i that maximize $\Sigma_{k=f_{min}}^{f_{max}} u_{ik}^2$, the data can be reduced to the K genes whose read-count information has the most value based on their location in the network for the chosen frequency range. In particular, we applied the GS algorithm for only low frequencies (below $\frac{N}{2}$) and only high frequencies (above $\frac{N}{2}$).

The idea behind graph frequency sampling (GFS) is essentially to transform the data into the spectral domain and retain components above or below a certain frequency [22]. The Graph Fourier Transform (GFT) $\hat{\boldsymbol{x}}$ of a signal $\boldsymbol{x}$

and its inverse are given by

$$\hat{\boldsymbol{x}} = \boldsymbol{U}^T \boldsymbol{x} \tag{2}$$

$$\boldsymbol{x} = \boldsymbol{U} \hat{\boldsymbol{x}} \tag{3}$$

In the GFS algorithm, the graph signal is projected only onto the K eigenvectors associated to eigenvalues below or above a cut-off frequency, resulting in a K-dimensional signal $\hat{\boldsymbol{x}}_{\boldsymbol{GFS}}$. For example, choosing the K lowest frequencies,

$$\hat{\boldsymbol{x}}_{\boldsymbol{GFS}} = [\boldsymbol{u}_1 ... \boldsymbol{u}_K]^T \boldsymbol{x}. \tag{4}$$

We used both the K lowest and highest frequencies separately to transform the original data. GFS using high frequencies (HF) potentially mitigates issues due to batch effects that might appear as a low-frequency component in the graph signal. Similarly, the presence of drop-outs might manifest itself as a high-frequency component and thus be filtered when considering only low-frequency (LF) eigenvectors.

The GFT was also used to implement graph filtering with a simple low or pass rectangle filter in the graph spectral domain. After transforming the data into a graph signal as described previously, the GFT is computed (Equation 2). The components of the resulting spectrum that are above or below a cut-off frequency are set to zero. Subsequently, the inverse GFT (Equation 3) is applied to obtain a filtered version of the original signal. While this process does not reduce the dimensionality of the data, it could provide insight on validity of the network creation process and reduce noise.

E. Data Transformations

In total, 9 different data transformation methods were used and compared to a baseline model. The baseline model for each of the four classifiers is obtained by removing features with variance smaller than 0.1 from the raw data and then training the classifier in question on the remaining features. The data transformation methods that were tested are PCA, Graph sampling (LF and HF), Graph frequency sampling (LF and HF), Graph filtering (LF and HF), Graph filtering (LF and HF) followed by PCA, and a nested method that combines all the above. Graph filtering (LF and HF) do not reduce the dimensionality of the data. For PCA, the number of components was chosen to be $K = 500$ based on the explained variance proportion, which is around 50% for 500 components (see Figure S1). For the other dimensionality reduction methods that are based on the network, also 500 components were chosen for consistency.

F. Machine Learning Methods

Within this project, we applied four different types of classifiers to all our data transformations: L1-regularized logistic regression, random forest, neural networks and XGboost.

Logistic regression classifiers were trained using the scikit-learn package in python [23]. The regularization parameter was chosen to be the best in accuracy using the built in ten-fold cross-validation functionality using a grid-size of 40. The solution was found using stochastic average descent, with a tolerance of 0.005.

The random forest classifier was also trained using scikit-learn. For training, 5000 fully grown, unpruned trees were constructed. At every split, $\sqrt{n_{features}}$ were considered when looking for the best split.

The neural network consists of fully connected layers alternating with rectifier linear unit (ReLU) activation functions using PyTorch [24]. It classifies into two classes (non-stem or stem cell) and returns the probabilities using a softmax classifier. Six layers were used for the dimensionality reduced data containing 500 features, as described above. A hidden layer fc0 with 500 nodes was added at the front of the network for the data with full dimensionality. This allows us to compare the addition of a layer with 500 nodes to a PCA of the same number of dimensions. The architecture is shown in Table II below.

Layer	fc0	fc1	fc2	fc3	fc4	fc5	fc6
Nb Nodes	500	2500	1000	200	100	20	2

Table II
ARCHITECTURE OF THE NEURAL NETWORK. FC0 IS ONLY PRESENT FOR DATA WHOSE DIMENSIONS ARE NOT REDUCED.

The network was trained with stochastic gradient descent on the cross entropy loss in 40 epochs. The best learning rate and regularization constant were found by validating on 15% of the train set. During the training of the neural network, a learning weight decay was used to avoid oscillating across the optimum.

Training of XGBoost was done using the python package xgboost [25]. In order to prevent overfitting, the best classifier was selected based on the best accuracy on a validation set consisting of 20% of the training data. A learning rate of 0.2 and a maximum tree depth of three. For all other parameters, default values were used.

For the nested models, the same four classifiers were used. But instead of using the data directly, the predictions of all data transformations and methods were used as input.

RESULTS

The performance of each classifier was assessed using AUC. The AUC of the basic classifiers on set B are summarized in Figure 2 below.

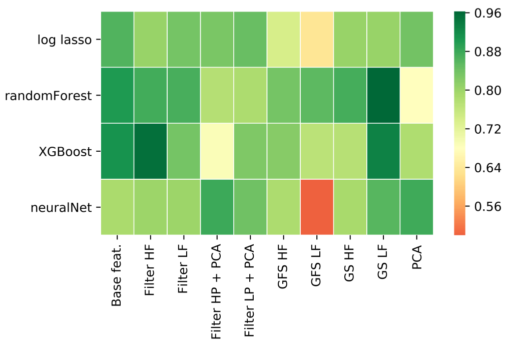

Figure 2. AUC scores on set B for the basic classifiers.

We observe that the impact of the choice of data transformation is significant. The best results were achieved using the network-based GS LF transformation with a maximum AUC of 0.963 for the random forest classifier. Moreover, HF filtering and GS LF with XGBoost perform well, with an AUC of 0.951 and 0.929. All other models perform worse than the baseline model trained with XGBoost and random forest with an AUC of 0.910 and 0.900 respectively. Notably, the neural network on HF followed by PCA performs quite well, with an AUC of 0.878 performing only slightly worse than the baseline model.

The following AUC scores were obtained on set C with all classifiers, including the nested classifier that was previously trained on the predictions of the basic classifiers on set B (Figure 2).

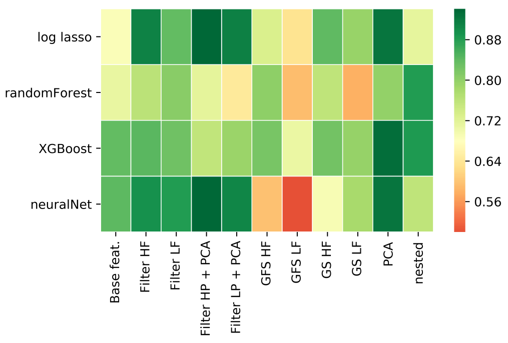

Figure 3. AUC scores on set C for all classifiers

It becomes immediately apparent that the nested classifiers did not perform better than their non-nested counterparts. Moreover, note that the best-performing method on set C, GS LF with random forest, performs very poorly with an AUC of 0.574, and is only barely better than just deciding by chance. Similarly, GS LF with XGBoost doesn't perform well either, with an AUC of 0.79. However, the neural network on HF followed by PCA has the best AUC score (0.941), followed by logistic lasso (0.940) on data with same transformation. This is followed by XGBoost, logistic lasso and neural networks trained on PCA transformed data (AUC of 0.934, 0.925, and 0.924). High-pass filtering before applying PCA thus slightly improves predictive performance of the classifier. Interestingly, dimensionality reduction using PCA outperformed the addition of a hidden layer of equal size in the case of the neural networks in both cases.

The results show that the neural network trained on HF filtered and PCA transformed data yields the most robust performance amongst all classifiers. Since robustness is of utmost importance, we thus determine it to be our final classifier. In order to understand how this final classifier performed on cells that are transitioning between stem- and non-stem cells, we generated predictions on the Joost dataset containing cells of the hair follicle. Supplementary figure S2 shows the predicted probabilities as box- and scatter plot for individual cells grouped by cell type, as well as the ordering of cell types according to stemness as done in the original paper. Our results clearly show that the predicted stemness probabilities are consistent with the results of the paper. The noise can at least to some extent be attributed to the somewhat inexact cell type annotation.

Conclusion

Our results show that different classifiers can have very different predictive performances on different test sets. We thus chose a final classifier that had a robust performance on both sets. The predicted probabilities were consistent with results of other methods. Moreover, a nested approach did not yield any improvement. Since set B, on which it was trained, contained much less different data sets due to a general lack of data, it is likely that the second level classifier overfitted and thus performed badly on set C.

Whereas the graph transformations didn't yield consistent improvements over the baseline model, high-pass filtering followed by PCA yielded the best generalizing classifier. This suggests that the filtering successfully reduced noise in the data, which could be promising for other applications as well.

As a follow-up to this report, it would be interesting to reconstruct the decision making process of the best-performing classifiers in order to backtrack the importance of individual genes in stemness decision. This may give valuable biological insights. Moreover, it could be of interest to take into account other networks, such as gene coexpression networks or neighborhood-based networks, in order to analyze whether they improve the classifier performance. A network that also yields consistently good predictions could be of interest for other applications, such as batch effect and drop-out correction[26]. Additionally, the application of existing batch-effect and drop-out correction techniques previous to classifier training may improve performance. Finally, since scRNA-seq gets increasingly popular, retraining the classifier when more data is available is of interest.

References

[1] P. C. Schwalie, P. Ordóñez-Morán, J. Huelsken, and B. Deplancke, "Cross-tissue identification of somatic stem and progenitor cells using a single-cell RNA-sequencing derived gene signature," *STEM CELLS*, vol. 35, no. 12, pp. 2390–2402, nov 2017. [Online]. Available: https://doi.org/10.1002/stem.2719

[2] J. N. Campbell, E. Z. Macosko, H. Fenselau, T. H. Pers, A. Lyubetskaya, D. Tenen, M. Goldman, A. M. J. Verstegen, J. M. Resch, S. A. McCarroll, E. D. Rosen, B. B. Lowell, and L. T. Tsai, "A molecular census of arcuate hypothalamus and median eminence cell types," *Nature Neuroscience*, vol. 20, no. 3, pp. 484–496, feb 2017. [Online]. Available: https://doi.org/10.1038/nn.4495

[3] R. Chen, X. Wu, L. Jiang, and Y. Zhang, "Single-cell RNA-seq reveals hypothalamic cell diversity," *Cell Reports*, vol. 18, no. 13, pp. 3227–3241, mar 2017. [Online]. Available: https://doi.org/10.1016/j.celrep.2017.03.004

[4] J. S. Dahlin, F. K. Hamey, B. Pijuan-Sala, M. Shepherd, W. W. Y. Lau, S. Nestorowa, C. Weinreb, S. Wolock, R. Hannah, E. Diamanti, D. G. Kent, B. Göttgens, and N. K. Wilson, "A single-cell hematopoietic landscape resolves 8 lineage trajectories and defects in kit mutant mice," *Blood*, vol. 131, no. 21, pp. e1–e11, mar 2018. [Online]. Available: https://doi.org/10.1182/blood-2017-12-821413

[5] B. W. Dulken, D. S. Leeman, S. C. Boutet, K. Hebestreit, and A. Brunet, "Single-cell transcriptomic analysis defines heterogeneity and transcriptional dynamics in the adult neural stem cell lineage," *Cell Reports*, vol. 18, no. 3, pp. 777–790, jan 2017. [Online]. Available: https://doi.org/10.1016/j.celrep.2016.12.060

[6] O. Gokce, G. M. Stanley, B. Treutlein, N. F. Neff, J. G. Camp, R. C. Malenka, P. E. Rothwell, M. V. Fuccillo, T. C. Südhof, and S. R. Quake, "Cellular taxonomy of the mouse striatum as revealed by single-cell RNA-seq," *Cell Reports*, vol. 16, no. 4, pp. 1126–1137, jul 2016. [Online]. Available: https://doi.org/10.1016/j.celrep.2016.06.059

[7] M. S. Kowalczyk, I. Tirosh, D. Heckl, T. N. Rao, A. Dixit, B. J. Haas, R. K. Schneider, A. J. Wagers, B. L. Ebert, and A. Regev, "Single-cell RNA-seq reveals changes in cell cycle and differentiation programs upon aging of hematopoietic stem cells," *Genome Research*, vol. 25, no. 12, pp. 1860–1872, oct 2015. [Online]. Available: https://doi.org/10.1101/gr.192237.115

[8] A. L. Haber, M. Biton, N. Rogel, R. H. Herbst, K. Shekhar, C. Smillie, G. Burgin, T. M. Delorey, M. R. Howitt, Y. Katz, I. Tirosh, S. Beyaz, D. Dionne, M. Zhang, R. Raychowdhury, W. S. Garrett, O. Rozenblatt-Rosen, H. N. Shi, O. Yilmaz, R. J. Xavier, and A. Regev, "A single-cell survey of the small intestinal epithelium," *Nature*, vol. 551, no. 7680, pp. 333–339, nov 2017. [Online]. Available: https://doi.org/10.1038/nature24489

[9] H. Hochgerner, A. Zeisel, P. Lönnerberg, and S. Linnarsson, "Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing," *Nature Neuroscience*, vol. 21, no. 2, pp. 290–299, jan 2018. [Online]. Available: https://doi.org/10.1038/s41593-017-0056-2

[10] L. B. Rodda, E. Lu, M. L. Bennett, C. L. Sokol, X. Wang, S. A. Luther, B. A. Barres, A. D. Luster, C. J. Ye, and J. G. Cyster, "Single-cell RNA sequencing of lymph node stromal cells reveals niche-associated heterogeneity," *Immunity*, vol. 48, no. 5, pp. 1014–1028.e6, may 2018. [Online]. Available: https://doi.org/10.1016/j.immuni.2018.04.006

[11] P. C. Schwalie, H. Dong, M. Zachara, J. Russeil, D. Alpern, N. Akchiche, C. Caprara, W. Sun, K.-U. Schlaudraff, G. Soldati, C. Wolfrum, and B. Deplancke, "A stromal cell population that inhibits adipogenesis in mammalian fat depots," *Nature*, vol. 559, no. 7712, pp. 103–108, jun 2018. [Online]. Available: https://doi.org/10.1038/s41586-018-0226-8

[12] S. Nestorowa, F. K. Hamey, B. P. Sala, E. Diamanti, M. Shepherd, E. Laurenti, N. K. Wilson, D. G. Kent, and B. Gottgens, "A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation," *Blood*, vol. 128, no. 8, pp. e20–e31, jun 2016. [Online]. Available: https://doi.org/10.1182/blood-2016-05-716480

[13] J. Park, R. Shrestha, C. Qiu, A. Kondo, S. Huang, M. Werth, M. Li, J. Barasch, and K. Suszták, "Single-cell transcriptomics of the mouse kidney reveals potential cellular targets of kidney disease," *Science*, vol. 360, no. 6390, pp. 758–763, apr 2018. [Online]. Available: https://doi.org/10.1126/science.aar2131

[14] P. T. Shah, J. A. Stratton, M. G. Stykel, S. Abbasi, S. Sharma, K. A. Mayr, K. Koblinger, P. J. Whelan, and J. Biernaskie, "Single-cell transcriptomics and fate mapping of ependymal cells reveals an absence of neural stem cell function," *Cell*, vol. 173, no. 4, pp. 1045–1057.e9, may 2018. [Online]. Available: https://doi.org/10.1016/j.cell.2018.03.063

[15] T. M. Consortium, "Single-cell transcriptomic characterization of 20 organs and tissues from individual mice creates a tabula muris," dec 2017. [Online]. Available: https://doi.org/10.1101/237446

[16] B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, D. Bertagnolli, J. Goldy, N. Shapovalova, S. Parry, C. Lee, K. Smith, A. Bernard, L. Madisen, S. M. Sunkin, M. Hawrylycz, C. Koch, and H. Zeng, "Adult mouse cortical cell taxonomy revealed by single cell transcriptomics," *Nature Neuroscience*, vol. 19, no. 2, pp. 335–346, jan 2016. [Online]. Available: https://doi.org/10.1038/nn.4216

[17] A. Zeisel, H. Hochgerner, P. Lönnerberg, A. Johnsson, F. Memic, J. van der Zwan, M. Häring, E. Braun, L. E. Borm, G. L. Manno, S. Codeluppi, A. Furlan, K. Lee, N. Skene, K. D. Harris, J. Hjerling-Leffler, E. Arenas, P. Ernfors, U. Marklund, and S. Linnarsson, "Molecular architecture of the mouse nervous system," *Cell*, vol. 174, no. 4, pp. 999–1014.e22, aug 2018. [Online]. Available: https://doi.org/10.1016/j.cell.2018.06.021

[18] D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, L. J. Jensen, and C. von Mering, "The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible," *Nucleic Acids Research*, vol. 45, no. D1, pp. D362–D368, oct 2016. [Online]. Available: https://doi.org/10.1093/nar/gkw937

[19] D. R. Zerbino, P. Achuthan, W. Akanni, M. R. Amode, D. Barrell, J. Bhai, K. Billis, C. Cummins, A. Gall, C. G. Girón, L. Gil, L. Gordon, L. Haggerty, E. Haskell, T. Hourlier, O. G. Izuogu, S. H. Janacek, T. Juettemann, J. K. To, M. R. Laird, I. Lavidas, Z. Liu, J. E. Loveland, T. Maurel, W. McLaren, B. Moore, J. Mudge, D. N. Murphy, V. Newman, M. Nuhn, D. Ogeh, C. K. Ong, A. Parker, M. Patricio, H. S. Riat, H. Schuilenburg, D. Sheppard, H. Sparrow, K. Taylor, A. Thormann, A. Vullo, B. Walts, A. Zadissa, A. Frankish, S. E. Hunt, M. Kostadima, N. Langridge, F. J. Martin, M. Muffato, E. Perry, M. Ruffier, D. M. Staines, S. J. Trevanion, B. L. Aken, F. Cunningham, A. Yates, and P. Flicek, "Ensembl 2018," *Nucleic Acids Research*, vol. 46, no. D1, pp. D754–D761, nov 2017. [Online]. Available: https://doi.org/10.1093/nar/gkx1098

[20] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446–475, mar 2018. [Online]. Available: https://doi.org/10.1016/j.acha.2016.05.005

[21] M. Menoret, N. Farrugia, B. Pasdeloup, and V. Gripon, "Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, nov 2017. [Online]. Available: https://doi.org/10.1109/globalsip.2017.8309033

[22] L. Rui, H. Nejati, and N.-M. Cheung, "Dimensionality reduction of brain imaging data using graph signal processing," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016. [Online]. Available: https://doi.org/10.1109/icip.2016.7532574

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[26] D. van Dijk, J. Nainys, R. Sharma, P. Kathail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Peer, "MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data," feb 2017. [Online]. Available: https://doi.org/10.1101/111591

[27] S. Joost, A. Zeisel, T. Jacob, X. Sun, G. L. Manno, P. Lönnerberg, S. Linnarsson, and M. Kasper, "Single-cell transcriptomics reveals that differentiation and spatial signatures shape epidermal and hair follicle heterogeneity," *Cell Systems*, vol. 3, no. 3, pp. 221–237.e9, sep 2016. [Online]. Available: https://doi.org/10.1016/j.cels.2016.08.010

SUPPLEMENT

A. Dataset Preparation

Cell type annotation was taken from the clustering done in the original papers. All three sets consist only of cell types which were determined as being clearly stem or non-stem in order to avoid introducing a bias into the model. Notably, progenitors, endothelial and fibroblast cells were excluded from the analysis. For the set A, the cells were down-sampled to be of equal proportions from the three different germ layers, and within each germ layer to be 50% stem and 50% non-stem. This was chosen in order to avoid introducing a tissue-specific bias in determining stem-ness.

Since different datasets measure the expression of different genes, one has to choose which genes to use for the analysis in order to be able to join different datasets. The genes retained for our analysis were selected as follows: For all eleven sets within our train set, we determined whether they contained stem cells, non-stem cells or both. Then, the genes common to all sets with stem-cells were combined with the genes present in all sets containing non-stem cells. In this way, genes that are only expressed in either stem or non-stem cells are retained while minimizing the number of genes with missing values. This resulted in 13'587 genes retained for further analysis.

For datasets in which a necessary gene isn't represented, we set the gene expression to zero. This is based on the heuristic that such genes are lowly expressed in these datasets, as otherwise they would have been sufficiently measured to be retained in the final dataset.

	Base feat.	Filter HF	Filter LF	Filter HP + PCA	Filter LP + PCA	GFS HF	GFS LF	GS HF	GS LF	PCA
log lasso	86.5	80.5	83.6	82.9	84.6	73.8	63.5	80.4	80.5	83.7
randomForest	90.1	87.6	87.1	77.6	78.4	83.5	85.6	87.3	96.3	68.0
xgboost	91.0	95.1	83.5	68.7	82.6	82.1	76.8	77.4	92.9	78.2
neural network	78.6	80.0	80.0	87.8	83.9	78.4	50.0	78.8	85.8	87.6

Table S1
AUC on set B, raw data in %

	Base feat.	Filter HF	Filter LF	Filter HP + PCA	Filter LP + PCA	GFS HF	GFS LF	GS HF	GS LF	PCA	nested
log lasso	68.7	91.0	83.8	94.0	91.2	73.1	63.4	84.0	79.5	92.5	71.3
randomForest	70.9	76.3	80.7	71.7	64.4	80.2	59.2	75.9	58.0	79.8	88.1
xgboost	83.8	84.4	82.8	75.6	79.2	82.1	70.8	82.5	79.7	93.4	88.4
neural network	84.1	89.4	88.2	94.1	90.8	59.6	50.0	69.2	78.2	92.4	75.8

Table S2
AUC on set C, raw data in %

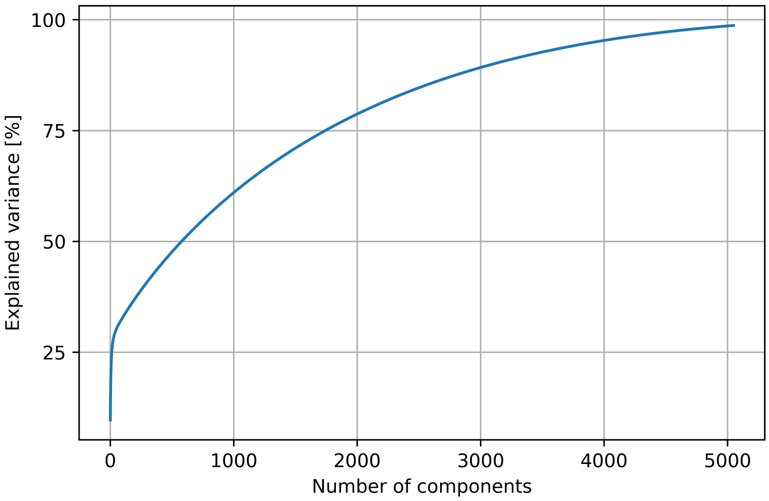

Figure S1. Explained variance as a function of the number of components chosen for PCA

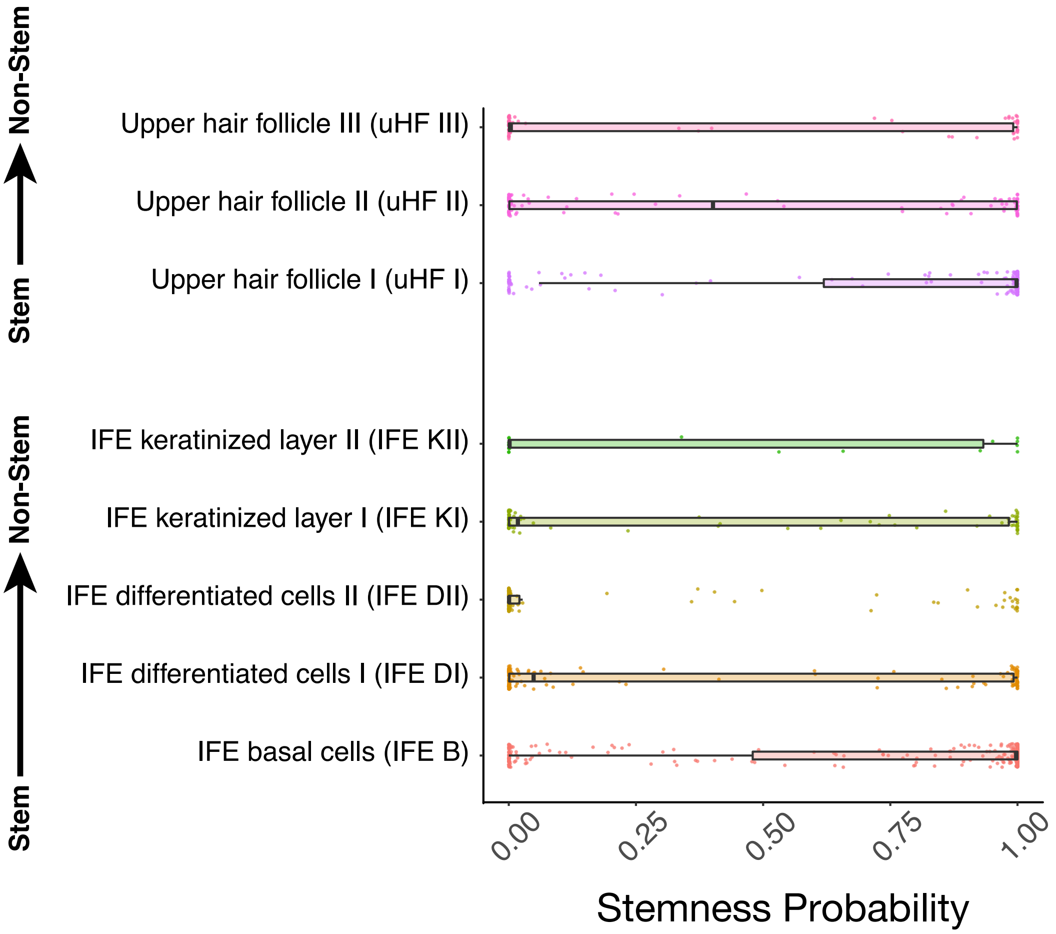

Figure S2. Predicted stemness probabilities as determined by best-performing classifier on subset of cell-types in Joost 2016 [27] data set. Arrows on the left side show the transition of stemness across cell-types, as determined in the original paper.