Automated labeling scRNA-seq cluster data using VAE and Random Forest

Project Group 17

Emily Hang, Landon Menuey, Jiwoo (Jenna) Kim, Caleb Connors

Abstract

The exponentially increasing amount of RNA sequencing data following the development of RNA sequencing technology provides novel knowledge about high-resolution profiling of cell transcriptomes. However, one of the largest challenges with RNA-seq data is properly labeling data for analysis. Labeling the cell clusters of single-cell RNA-seq data (scRNA) typically requires manual labor but this process is time-consuming and requires the labeler to have a comprehensive background in biology. Many machine learning methods recently developed specialize in automated label generation for single-cell data types. Here, we implemented a machine learning method that utilizes a variational autoencoder (VAE) and random forest model to predict cell types (B, CD4 T, CD8 T, DC, Mono, NK, other, and other T) in bone marrow aspirate concentrate (BMAC). Our model predicts BMAC cell type with 92% accuracy.

Introduction

The rise of RNA sequencing has opened avenues for thousands of new research studies, from discovering novel transcripts to understanding differential gene expression across cell types. Since the first scRNA sequencing experiment in 2009, scRNA sequencing has given researchers the ability to understand the transcriptome of individual cells at a high level.

One of the common steps in scRNA seq processing is the generation of UMAPs and cell clusters within said UMAPs. A UMAP (Uniform Manifold Approximation and Projection for dimensional reduction) visualizes high-dimensional data–such as gene expression–in a 2D space. Although common programs for scRNA seq processing have been built (e.g. Seurat in R or ScanPy in Python), these programs still require manual labor. In essence, programs identify the optimal number of clusters for a dataset, but researchers must label each cluster.

The Basics of Single Cell Analysis with Bioconductor states “It is more difficult to determine what biological state is represented by each of those clusters. Doing so requires us to bridge the gap between the current dataset and prior biological knowledge, and the latter is not always available in a consistent and quantitative manner… with most practitioners possessing an ‘I’ll know it when I see it’ intuition that is not amenable to computational analysis. As such, interpretation of scRNA-seq data is often manual and a common bottleneck in the analysis workflow.”

Our goal with this project was to improve automated label generation by leveraging automated labeling programs and our experience with scRNA-seq data. The model we implemented consists of a VAE and a random forest model. We had an average precision and recall of 0.89, with our highest precision and recall being in identifying B-cells (1.00, 0.98) and monocytes (0.99, 1).

Relevant work:

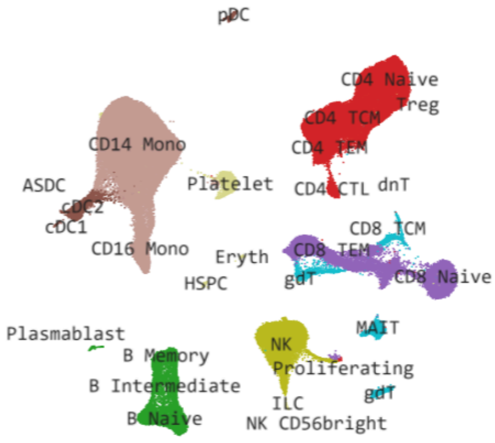
Many recent machine learning models are based on three approaches used for cell type annotations: marker gene database-based approach, correlation-based approach, and supervised classification-based approach. The general workflow of supervised learning for annotation tasks requires having labeled training data and the model is trained to predict the labels based on the selected features.

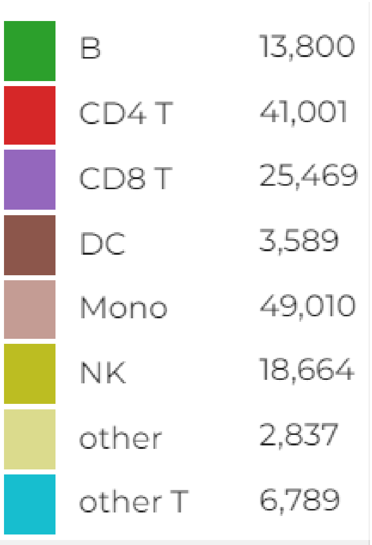
Three examples of recent ML models are scPred, SingleCellNet, and ACTINN. scPred mainly uses the classification model of Support Vector Machine (SVM) to train the classifiers for each cell but is flexible to other classification algorithms such as random forest and gradient descent (Alquicira-Hernandez et al., 2019). SingleCellNet uses the classification model of Random Forest to robustly quantitatively measure the similarity of a query single-cell RNA-seq data to the accumulated reference data across different databases (Tan & Cahan, 2019). ACTINN is an artificial neural network-based model that is implemented using TensorFlow (Ma & Pellegrini, 2019). The model network has 3 hidden layers and ReLU is used for the activation function and softmax function for the output layer. These three models showcase that various classification and machine-learning techniques can be used for automated labeling tasks. The prediction accuracy for ACTINN is 76%, SingleCellNet is 57%, and scPred is over 90% on highly specific cell types. However, recently, a deep generative artificial neural network model called variational autoencoder (VAEs), is more commonly used for scRNA-seq data annotation. VAE is optimized for image feature selection. Feature selection is traditionally performed by creating a list of features and respective metrics. However, the downside of this is that it is biased towards features that are more easily quantifiable. VAEs can reduce biases like this as an unsupervised model that doesn’t require prior knowledge. Some previous VAE models that were developed tackle this problem by controlling for uninformative features by using the hypersensitivity VAE model (Ternes et al. 2022).

Our approach to making an automated cell labeling model is similar to previous models that utilize VAEs and artificial neural networks. We used similar layer compositions with fully connected layers with ReLU as activation layers. However, our approach is different from previous models as we are doing an additional dimension reduction before the VAE and recreate the type of processing seen in scRNA-seq for better encoding results, as well as utilizing a random forest after encoding the data to predict cell type.

Data:

The BMAC data was downloaded from the New York Genome Center, as part of their integrated analysis of multimodal single-cell data: https://atlas.fredhutch.org/nygc/multimodal-pbmc/. We downloaded the full (raw) dataset of 161,159 cells. This dataset has three levels of labeling (L1, L2, and L3) along with information on the gene expressions across cells. L1 is the broadest level and has 8 categories: B, CD4 T, CD8 T, DC, Mono, NK, other, and other T. L3 is the more specific, with 57 categories. We focused on L1 for our model and aimed to be able to classify cells across 8 categories accurately. For pre-processing, we scaled the data and ran both PCA and UMAP on the data to reduce dimensionality. The data is visualized here with a UMAP:





🟩	B	13,800
🟥	CD4 T	41,001
🟪	CD8 T	25,469
🟫	DC	3,589
🟫	Mono	49,010
🟨	NK	18,664
🟨	other	2,837
🟦	other T	6,789

Methods:

Additional research on previously done relevant work along with the feedback from Prof. Luo made it clear that we needed to shift our goals slightly to mainly focus on utilizing our knowledge and experience of variational autoencoders to develop our model instead of pursuing our original plan of trying out scripts of different models such as scPred, SingleCellNet, and ACTINN. Although these models use machine learning techniques and have been published with high accuracy, more updated machine learning methods such as generative artificial neural networks were available and the scripts were written for specific datasets the researchers had in mind. VAEs are optimized for high dimensional data such as scRNA seq cluster data, so it is currently widely used for scRNA data. After manipulating the dataset by pulling out the cell expressions manually, we determined that it was best to develop our own VAE model from scratch that is catered to our dataset with the previously mentioned models in mind.

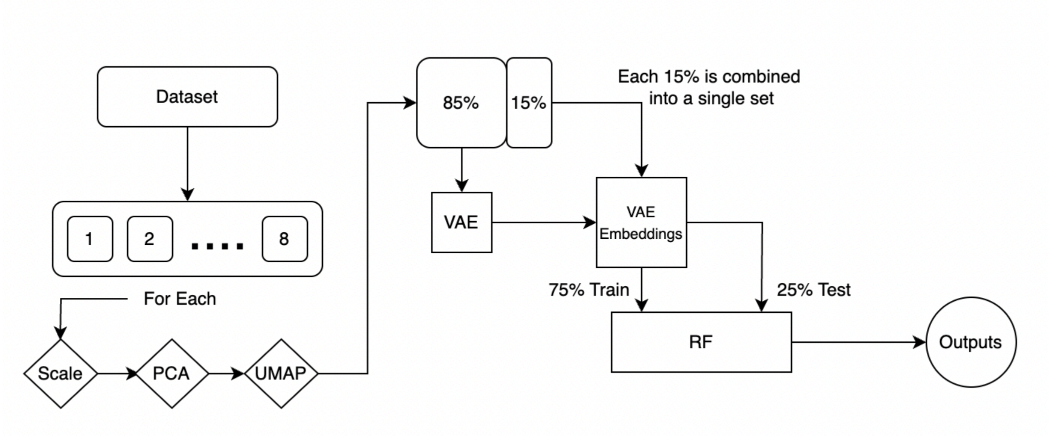


Figure 1. The Model Workflow

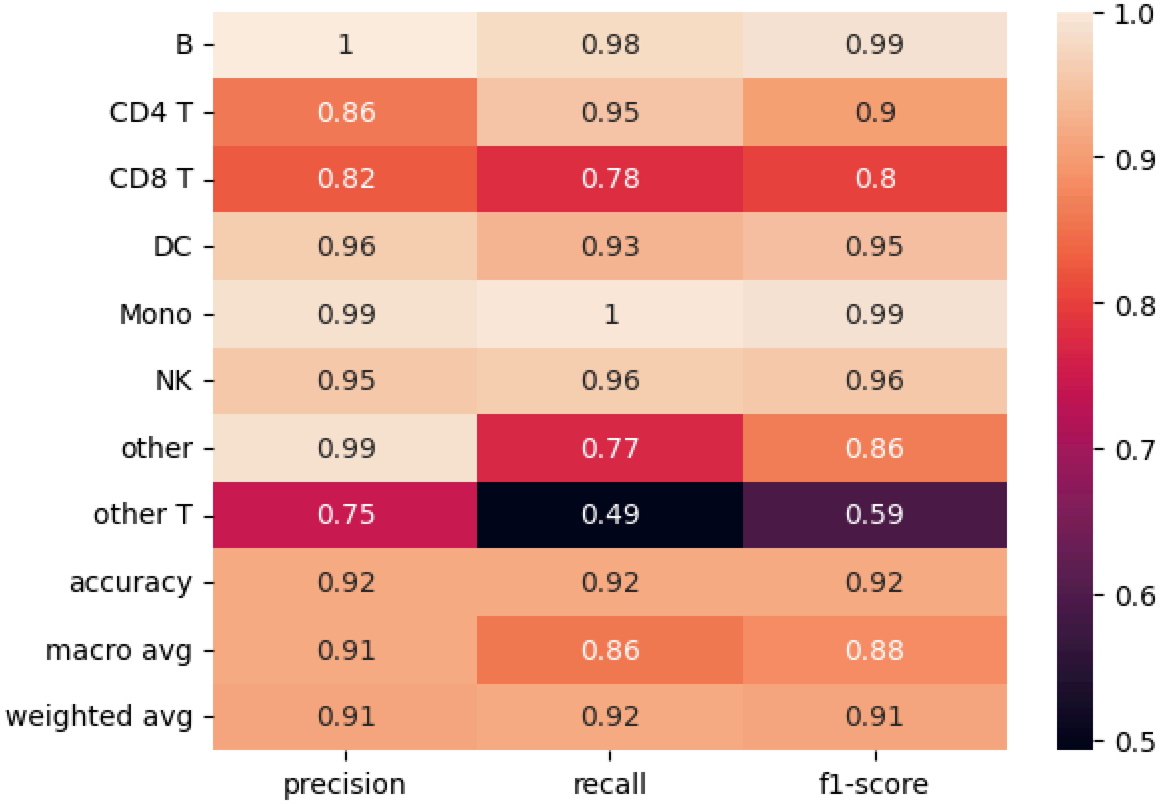
For each 8th of the data, we divide the dataset into 85% and 15% split. For the data pre-processing, we ran a standard scaler on the data, then PCA and UMAP to reduce the very large dimensions of the data (a little over 22000 points per cell) before handing the data off to the VAE. The 15% from each dataset are combined to create a separate train and test set for the random forest separate from the VAE fitting data.

VAE combines the features of encoders and variational Bayesian methods. The encoder of VAE compresses and plots each data point on the distribution in the latent space. The decoder takes the sampled information from the latent space distribution as input and produces a prediction by decompressing the sampled information. The encoder of our VAE consists of three fully connected dense layers with ReLU activation functions with 256 and 128 as the number of neurons of hidden layers. Each of these layers gradually reduces the dimensionality of the high dimensional scRNA-seq data input. The last layer of the encoder outputs two vectors, mean and log variance, which sample the latent space points. The decoder also consists of three fully connected dense layers with ReLU activation functions. Each of these layers gradually increases the dimensionality back to the original input space. The last layer of the decoder outputs the same sized vector as the input layer and uses the sigmoid function as an activation function to normalize the values to fit the [0, 1] interval that represents the images. We used this model on 85% of the data and split the data into train and test data. Then, we took the remaining 15% of the data from 8 datasets, combined them, and generated latent representations using the encoder of our VAE model to further reduce the dimensionality of the data. Our model compiles with an Adam optimizer and categorical focal cross-entropy loss as a loss function since we are classifying cell types that could benefit from being assigned different weights. Then, we put 75% of the data that was not used in VAE training or testing (15% of each of the 8 dataset partitions) in latent space into a random forest model to fit and the rest of it to predict to get the model performance data.

Experiments

There were several features that we tried in the model that did not make the final cut. Firstly, we started with a mean squared error as the loss function. It worked fine, but we quickly realized that MSE was not for categorical data, and thus the loss function was switched to categorical focal cross-entropy to account for this. Since we were using categorical focal cross-entropy as the loss function, we thought that we needed to one hot encode the data as stated on the keras website. This wound up not being the case for our labels, as the loss function worked perfectly without the labels, and they wound up creating issues downstream when we switched from running one CSV at a time as a proof of concept of the VAE and random forest combination to a batch train of all the CSVs. This batch training was a necessity due to the size of the data. Even when given over 500gb of ram in PACE, attempting to pd.concatenate() on the 8 CSVs crashed the kernel due to memory usage. The data was simply too large in a non-sparse format to handle all at once in a data frame.

We also initially did not have a UMAP step, but it was added as it was a typical part of scRNA seq processing in the hopes that it would improve our accuracy, which it did. Later attempts to remove the UMAP and make the VAE more complex were met with negatives on the loss function and no accuracy at all, so the UMAP proved itself very important in this task. Pandas was also replaced with modin.pandas in order to improve read times and take advantage of the multi-threaded computational resources we had. We also removed an early cutoff point when the model was not reducing its loss after several generations. It worked well on one CSV, but with multiple the VAE loss often increased as it had more and more data to deal with and so the model would get cut off early and not learn much. Different hyperparameters were tried, although with a limited amount of trials due to the long read time. Components such as PCA and UMAP components, number of random forest estimators, learning rate, layers in the VAE, and latent dimensions were all tried. Further hyperparameter tuning here would be valuable, but we simply ran out of time. The best solution we had performed very admirably, with an overall accuracy of .92 which is at or above the other packages listed earlier. In larger, more clearly defined clusters such as B cells and monocytes, it was nearly flawless. In cell labels that were slightly more irregular, such as the CD8T and CD4T labels containing proliferating cells that are not near their main clusters, or the “other” grouping that contained a wider range of clusters, the performance was slightly worse. The worst by far was “Other T”, which not only has several small subclusters all under one banner but is also mixed right around the CD8T cluster (which unsurprisingly was the second worst performing label). Ironically, adding more specificity to the labels (such as l2) would probably improve performance in these specific clusters due to the wide range of biological states included in one group.



Conclusion and Discussion

One current limitation of our model is that it was only trained on BMAC data, and it cannot transfer to other cell types. This limitation can be overcome by additional training on specific datasets of interest. Specific models could then be called upon depending on the cell type of the sample in question. Additionally, the accuracy in a few of the clusters ranges from suboptimal to unhelpful. Creating a custom set of labels that blends l1 and l2 could create a much better labeling schema for this task. From experience, BMAC samples often end up with mid-teen cluster counts. Keeping the broader labels for some of the cell types but breaking down the other categories into their more specific cell types would almost certainly improve accuracy in the categories that are the worst.

Our current model can accurately categorize the L1 level category of the dataset except for a few of the more mixed clusters. In future works, we plan to work on the model and the dataset to both remove dataset barriers discussed above that are preventing the model from learning well, as well as improve the way the model learns from the whole dataset, not just one cell at a time. Adding the ability to compare the cell against its neighbors' expression could help present a more aggregated cell-type prediction score, where one cell outlier in the middle of a cluster would be “corrected” by studying nearby cells. We could improve the sensitivity of the model by changing the model architecture and parameters. The model is yet to be hyperparameter-tuned, which could unlock more potential. We can also use different learning mechanics for latent space by using a method such as Disentangled Representational Learning (DRL) (Wang et al. 2022) could be used to encourage the latent space to learn more of the underlying factors hidden in the data more independently.

Code Availability

Link to CoLab notebook can be found here:

https://colab.research.google.com/drive/1FQ7MtrWn6hwJflzSGaLMfQB-GUEs1wa_?usp=sharing

Works Cited

Integrated Analysis Of Multimodal Single-Cell Data. (n.d.). New York Genomic Center. https://atlas.fredhutch.org/nygc/multimodal-pbmc/

Ma, F., & Pellegrini, M. (2019). ACTINN: automated identification of cell types in single cell RNA sequencing. Bioinformatics, 36(2), 533–538. https://doi.org/10.1093/bioinformatics/btz592

Muraro MJ, Dharmadhikari G, Grün D, Groen N, Dielen T, Jansen E, van Gurp L, Engelse MA, Carlotti F, de Koning EJ, van Oudenaarden A. A Single-Cell Transcriptome Atlas of the Human Pancreas. Cell Syst. 2016 Oct 26;3(4):385-394.e3. doi: 10.1016/j.cels.2016.09.002. Epub 2016 Sep 29. PMID: 27693023; PMCID: PMC5092539.

Tan, Y., & Cahan, P. (2019). SingleCellNet: a computational tool to classify single cell RNA-SEQ data across platforms and across species. Cell Systems, 9(2), 207-213.e2. https://doi.org/10.1016/j.cels.2019.06.004

Ternes, L., Dane, M., Gross, S. M., Labrie, M., Mills, G. B., Gray, J. W., Heiser, L. M., & Chang, Y. H. (2022). A multi-encoder variational autoencoder controls multiple transformational features in single-cell image analysis. *Communications Biology*, *5*(1). https://doi.org/10.1038/s42003-022-03218-x

Wang, X., Chen, H., Tang, S., Wu, Z., & Zhu, W. (2022). Disentangled representation learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2211.11695