Predicting GEx from L1000 Genes and Beyond for scRNA-seq Data

Newman Cheng
Department of Computer Science
Columbia University
New York, NY 10027
nc2893@columbia.edu

Abhishek Iyer
Department of Biological Sciences
Columbia University
New York, NY 10027
ahi2112@columbia.edu

Sachin Kadyan
Department of Computer Science
Columbia University
New York, NY 10027
sk4835@columbia.edu

Abstract

The LINCS L1000 consortium introduced a novel approach for profiling gene expression with only a limited number of fully mapped genes. Their idea was to infer the gene expression of approximately 12k genes by only sequencing 978 genes called the landmark genes. With single cell RNA-seq data, the same intuition can be applied. Single cell data is more sensitive and may perform better to predict gene expression for different cell types. Our approach uses a deep neural network model to predict gene expression. We first aim to evaluate the performance of the model using the landmark genes from the LINCS consortium. The DNN model outperforms the accuracy of the inference algorithm developed by the LINCS consortium. Using group lasso for feature selection, we identify a new subset of genes based on scRNAseq data and compare its performance with the landmark genes from L1000. We compared the performance of the new 910 genes after feature selection and obtained performance comparable to using the L1000 genes.

1 Introduction

RNA sequencing and in particular sc-RNA sequencing has provided tremendous insights in characterizing the genetic basis of human diseases. With the advent of sc-RNA seq, novel cell types have been characterized based on genetic characteristics rather than histopathological assays alone. This has led to insights in understanding development, cell proliferation and understanding human diseases such as cancer. One of the major challenges in RNA-sequencing is to sequence the RNA at a sufficient depth to accurately predict expression values. Sequencing all the genes in the genome can be time consuming and costly for obtaining accurate results. The LINCS consortium tried to mitigate the problem by coming up with an elegant solution. They identified that the expression of genes are correlated and can be determined by a small subset of genes they termed as the landmark genes. Using a data driven approach they were able to infer the expression of 12,000 genes from just 978 genes. This approach further led to the development of the L1000 assay, where only the landmark genes are sequenced with greater throughput. This helped establish a consortia where rapid high throughput assays could be performed with various perturbations to gain further insights to disease progression and treatment. With the advent of scRNA-seq where throughput for single cells could be lower, this approach could be beneficial. Also, due to the increased sensitivity of scRNA-seq we

Machine Learning for Functional Genomics (COMS 4762 Spring 2021), New York, United States.

expect a deep learning model would do better in terms of inferring gene expression values. Add to that the advantage of having cell type information in tissues would help in improving prediction for various tissues and cell types. The biological question we are tackling aims to find a subset of genes that when fully sequenced can be used to predict all other gene expressions with high accuracy. We first aim to establish a deep learning method that outperforms the inference algorithm using scRNA seq data and the 978 landmark genes. Using the performance as a benchmark, we aim to find LXXX genes that can be used to predict expression profiles for scRNA-seq data. Our study shows that using a simple group lasso feature selection method on the first layer of the neural network helps us obtaing a new subset of genes that outperforms the results obtained using the L1000 genes.

1.1 Prior Work

To the best of our knowledge there have been no reports on establishing a method to perform feature selection on scRNA seq data to establish a new set of "landmark" genes that could predict the gene expression of the entire dataset. Our inspiration comes particularly from the study by the Broad institute on developing the L1000 assay and their implementation of the inference algorithm. The L1000 genes were established using a data driven approach. They performed repeated k-means clustering to identify robust gene clusters. Based on the clusters they selected the genes that could explain the variance the most. The genes were then excluded and clustering was performed again to identify a new set of genes. The cycle was continued till they could not find genes that clustered together with high significance. The authors had sampled 12000 microarray studies They have also later developed a deep learning model on the same L1000 genes and L1000 assay that slightly outperforms the inference algorithm. While they tried to implement a deep learning approach, there was no implementation of feature selection to identify if the L1000 genes were the best genes to characterize the entire gene space or if a new subset could perform better. It is also surprising that this approach has not been translated to scRNAseq data. scRNA seq is much more sensitive compared to microarray used by the L1000 assay. It also has the added advantage of sequencing individual cells hence the information obtained is more robust. Another advantage is that a single scRNA seq experiment can yield a deep matrix with number of samples equal to the number of cells sequenced. This is highly advantageous considering that using a few scRNA seq datasets we can obtain a lot more samples compared to the 12000 analysed by the L1000 consortium.

1.2 Technical Motivations

One of the challenges with scRNA seq data is sparsity of the data. There are a lot of zeros for most of the features creating a class imbalance. In turn, this affects the training model pushing the model to predict more 0s during training. We took inspiration from DeepImpute and adopted a weighted MSE and group lasso for our neural network approach. DeepImpute uses a weighted mean square error loss function to mitigate the 0 value class imbalance problem. The general idea is to give higher weights to non-zero values. This is one way of mitigating class imbalance. This ensures that high confidence values are more deterministic of the accuracy. It also prevents over-penalizing of genes with low expression values. In DeepImpute, their loss function is defined as such.

$$loss = target * (input - target)^2$$

However, given the prevalence of 0's in our dataset, we deemed that this loss function would not work since it would not count any loss if target is 0. The zero would wipe out the entire loss and is the most case for all of our data. Thus, we defined a loss function, similar to DeepImpute, but factoring in targets of zeroes.

$$loss = (target + 1) * (input - target)^2$$

1.3 Group Lasso

In order to accomplish gene selection, we used group lasso to identify which genes were most important in our deep neural network's predictions. Group lasso is a regularization technique used to reduce high dimensional data sets like ours by sending beta coefficients to zero. While traditional lasso regression takes the absolute value of the beta coefficient, group lasso uses the sum of squares of coefficients belonging to the same group to penalize. By using group lasso, we can effectively perform feature selection as group lasso yields the most important feature or feature groups.

2 Methods

2.1 Data Collection & Exploration

The data we use comes from PanglaoDB for human tissue samples from two different tissues types: colon and prostate. Each individual tissue further contains different cell types. With this data, we trained a baseline model using linear regression and an advanced model using deep neural networks. The linear regression model aims to tell us whether the predictions we make with the deep neural network are actually more informative. At a high level, to train our neural network we will first use the L1000 genes as features to see if there is an improvement before training the network on all genes to do feature selection. To validate the features (genes) selected by our model we will also compare it with a model trained with the L1000 landmark genes as the benchmark.

To collect the data, we first structured the data to allow for conducting experiments similar to and using L1000 data. Since we aim to find landmark genes universal across different cell types, we collected single cell data for two tissue types with 12 cell types total. To minimize bias in the data across all cell types, we collected a similarly equal number of cells for each cell type, within a few hundred cells difference.

To explore the data, we used a Python package called scprep which handles single cell data with ease and allows for sparse visualizations. Given the sparsity of raw scRNA-seq data, we decided to do preprocessing in the form of gene filtering. With gene filtering, we only kept expressed genes with at least three reads in at least five cells. All non expressed genes were ignored and not considered. Cells with library size less than 1000 was also removed from analysis. Furthermore, we normalized the total number of reads for each cell and did a log transformation on the data. The normalization helped to remove any outliers and anomalies in the data that would potentially affect the performance of the model.

Our final dataset consisted of 15933 cells and 8960 genes. The 15933 cells, belonged to 12 different cell types: Basal Cells, Cholangiocytes, Enterocytes, Epithelial Cells, Goblet Cells, T Cells, B Cells, Dendritic Cells, Luminal Epithelial Cells, Serotonergic Neurons, Ductal Cells, and Unknown. To feed the data into our baseline models, we created PyTorch data generators that would yield a batch of results with multiple cell types from different experiments. Specifically, the data generator would yield a tuple containing the gene set and its respective target values. In order to make predictions based on different "landmark" sets, we had to remove the landmark genes from the final dataset as it would affect prediction results. To clarify, the L1000 genes removed from the final dataset are used to predict all other expression, but a full dataset is required to do feature selection since we do not know which genes are most important for prediction. Thus, our final dataset with all but landmark genes intact consisted of 15933 cells and 8260 genes. Expanding, We used the same methods to feed the data into our model. We have two different datasets because we do not know which genes are good predictors of other expression in the entire set. The condensed dataset allows us to test the performance of the DNN model against the linear regression model, while the full dataset allows us to do feature selection for our own "landmark" genes.

2.2 Model Training

2.2.1 Linear Regression

In regards to the ML model, we decided to start out with a linear regression model to try and predict gene expressions using the L1000 gene set. Our first experiment aimed to create a baseline with the 978 landmark genes from L1000 which we used to test future approaches' performance against. The model takes in a matrix of shape [N x 978] where N refers to the number of cells in the batch and 978 refers to the number of landmark genes. It then outputs a matrix of shape [N x 8206], with N being the number of cells in the batch and 8206 being the number of genes to predict expression for excluding the landmark.

To train the linear regression model for our baseline, we used the entire data set with 15933 cells with corresponding gene expressions. We split the data at a 80/20 training and validation split resulting in 12746 training examples and 3187 validation examples. In this case, to train our linear regression model, we trained the model with a batch size of 1, resulting in a input shape of [1x978] to an output shape of [1x8206]. To prevent over fitting of the model, we used early stopping to avoid the model from learning the training data too well. Finally, to evaluate our model, we used Mean Squared Error Loss to give us insight into how far off the predicted values were from the true target values.

2.2.2 Deep Neural Network

We chose to pursue a deep neural network approach for predicting gene expressions from a smaller subset of genes. In our proposal, we wanted to use transformers or convolutional neural networks but further research showed that these architectures did not fit the context of our problem. In training all of our DNN's we used a manual seed to ensure reproducibility of results. To construct our DNN, we used a series of Linear layers followed by a dropout layer and ReLU activation before a final Linear layer was added to make predictions. The first DNN model we trained was using the L1000 landmark genes to predict gene expressions for the remaining genes, excluding landmark. The inputs and outputs for the model were similarly [Nx978] and [Nx8206] respectively, with N being the batch size. We chose to use an Adam optimizer with a learning rate of $1e^{-2}$ and trained the network for 100 epochs. Similar to how we trained our linear regression model, we implemented early stopping, with a patience of 5, to ensure the model would not over fit to the training data.

Furthermore, in order to find our own "landmark" genes that would be able to predict all other gene expressions, we trained the model by feeding in the entire gene set with a goal of predicting the entire set. Thus, our network's inputs and outputs were the same, [Nx8960] since we do not remove any genes as features or targets. We chose this approach because through training, the network would eventually be able to identify which genes were most impactful into predicting the remaining gene expressions. For this model specifically, we implemented a weighted MSE loss in order to account for the imbalance in zeros within the data set. In addition to the weighted MSE loss, we added a group lasso term. The group lasso term was applied to the first layer of the neural network as a form of feature selection. Since group lasso was only applied to the first layer, after the model converged, we were able to look at the layer's weights to tell us which genes were most impactful in making predictions. The maximum and minimum of each gene's layer's weight was taken and ranked against all other genes. As a result, we took the top five hundred and bottom five hundred genes to form our "1000" landmark gene set. There is undoubtedly overlap between the top and bottom five hundred genes and as a result we ended up with 910 genes.

Finally, to test the prediction quality of our own landmark genes, we used the same model for retrieving a baseline accuracy with the L1000 genes. However, rather than a [Nx978] layer input, we had a [Nx910] input due to the smaller amount of genes we extracted. Our output size was also different as we had a different set of genes to predict, resulting in a [Nx8050] batch.

3 Results

3.1 Data Exploration

Through the use of gene filtering, we were able to condense the gene set by around 80%. At first glance, the significant reduction in genes may be alarming but given that the genes are un-expressive across different tissue types and experiments, it is unlikely that they are expressive in other cell types. Furthermore, we decided to utilize visualizations of our data in order to retrieve a better representation.

3.1.1 Principal Component Analysis

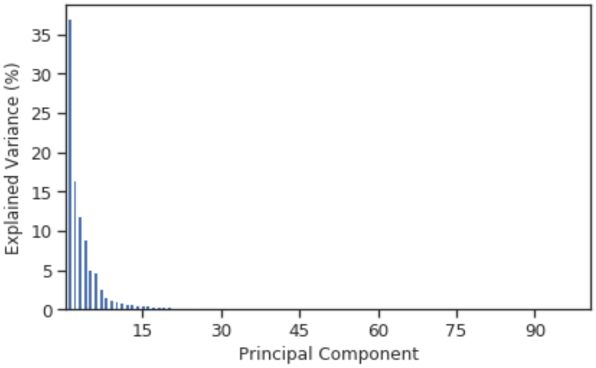

Figure 1: Colon Experiment PCA

We ran PCA to get a better understanding of correlation between features in the data, in this case between genes themselves. In the above figure, it seems that through looking at PCA, there is a high correlation in the genes indicating that a subset of genes could explain the entire experiment. However, this analysis is representative of only colon and prostate cell types, not across other cell types and is not able to be generalized.

3.1.2 Number of Cells versus Library Size

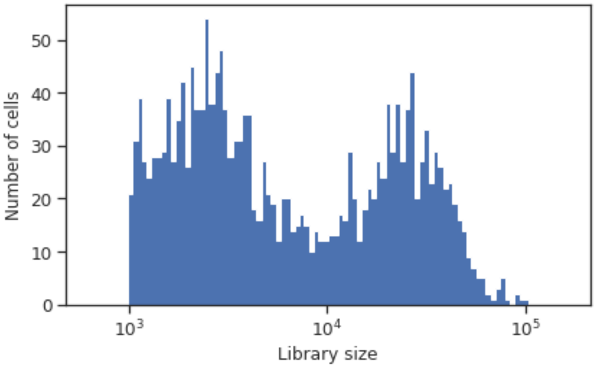

Figure 2: Single Colon Experiment Library Size

We decided to plot the number of cells in relation to the library size to understand the reliability of the data. The library size refers to the total number of reads for every cell and the more reads there are, the better capture of the RNA inside the cell. Hence, with more reads, the reliability of the data increases. From the figure above, we see that all the cells have at least 1000 reads indicating that the data is of good quality and the cells have been sequenced to sufficient depth. However, the distribution also shows that the library size varies a lot amongst the cells. To mitigate this problem we normalized the data as mentioned in the methods.

3.1.3 t-SNE Plot for Colon Cells

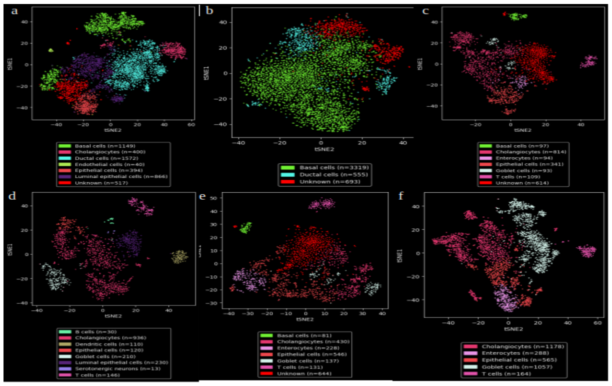

Figure 3: t-SNE Cell Clustering

t-SNE plots were generated for the data to observe how the cells cluster. We did it for individual experiments and observed a minimum of 10 clusters for a single experiment. Based on the cell type markers we expect atleast 13 different known cell types but many more unknown clusters were present. We tried creating a t-SNE plot for all cells combined, but we could not get distinct clusters. All analysis was performed using the alona tool from PanglaoDB. Information of cell type markers were obtained from PanglaoDB. Fig.3 represents a sample t-SNE plot for the individual experiments. a,b are prostate tissues and c-f are colon experiments.

3.2 Model Performance

There has prior work yet to be done on this specific task, so we first trained a basic linear regression model to compare our deep neural network's performance against. With the 978 landmark genes to train our base linear regression and DNN, we can proceed to train the DNN on the entire data set to extract a smaller gene set that can be used to predict. As a result, the landmark genes only serve the purpose of comparing initial performances between the two approaches and confirming that the DNN is suitable for gene set extraction.

3.2.1 Linear Regression

With a basic linear regression model without any data normalization, we were able to achieve a MSE of 1.87, not an impressive standard to begin with. However, these results were improved upon through the use of normalizing the data as discussed above. Through data normalization, the model was able to achieve a MSE of 0.0684, an increase in model performance. The predicted values were closer to the true target values with normalized data and as a result we proceeded to only use normalized data in training our deep neural networks. Naturally, we also expect to see an increase in performance with more advanced models like deep neural networks. We fed in the same inputs to the model, by taking in the 978 landmark genes and seeing how well the deep neural network predicted gene expressions for the remaining 8206 genes.

3.2.2 DNN

The DNN's baseline performance for predicting gene expressions using the L1000 genes saw a MSE of 0.0387 on the validation set. The baseline DNN showed that it would be an informative model to use in trying to extract a feature set since it performs better than a simple linear regression model. Not only is the model more complex allowing it to learn different representations and groupings, larger and deeper models typically perform better.

The DNN's performance for predicting gene expression using our own landmark gene set saw a MSE of 0.0490 on the validation set. While the model's performance using our own landmark gene set did not achieve a MSE lower than the model trained using the L1000 genes, these results are comparable. A MSE difference of 0.01 does not entirely mean that our selected genes perform worse, but perhaps may be attributed to different gradient descent initializations, random seeds, or other factors. However, in comparing the DNN prediction's using our landmark set, we find that it still performs better than a simple linear regression model.

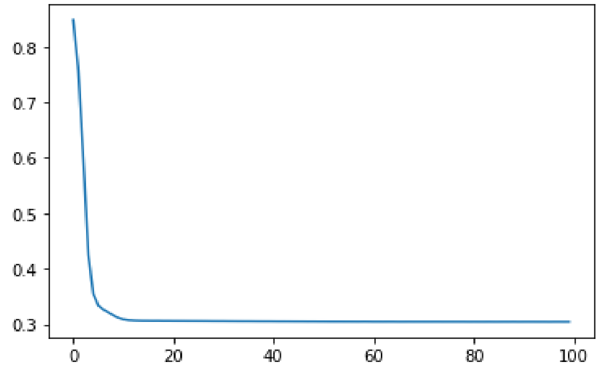

Figure 4: DNN Training Curve (100 epochs)

3.3 Selected Genes

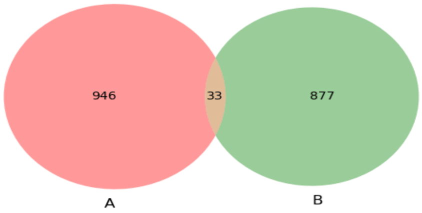

Figure 5: Comparison between L1000 and our F910 set (100 epochs)

With our DNN, upon feature selection using group lasso we obtained 910 genes from our entire dataset that could be used to predict other gene expressions henceforth referred to as F910. From Fig. 5 it is visible that in comparison to the L1000 genes, our F910 is quite different. We found that only 33 of the genes in our extracted landmark set were in the original L1000, constituting less than 5% of genes. It is interesting to see such a small overlap in terms of the "selected" genes between L1000 and our approach and yet obtain comparable performance. To investigate further we did a basic GO enrichment to check if similar biological processes were enriched. From the analysis it appears that F910 is enriched for genes involved in ECM organization and cytoskeletal structure. In comparison, the L1000 genes are enriched for response and regulation to stimulus and metabolic processes respectively. It would be interesting to further investigate why this is the case and if this is a result of the sparsity of scRNAseq data. As cytoskeletal genes and ECM genes have stable and high gene expression across cell types. It is possible that these 33 genes are the most important in predicting gene expression for our cell types in our dataset, thus explaining their appearance in our extracted gene set.

3.3.1 Colab Notebooks

Linear Regression Notebook:

`https://colab.research.google.com/drive/1qBEaCwLwKn1iryjS8aJFwri2m02HBvHl?usp=sharing`

Deep Neural Network L1000 + Finding L910 Notebook:

`https://colab.research.google.com/drive/1xs8Akv2RnKjsrSMn4-R-OptS3yjMYJcX?usp=sharing`

Deep Neural Network L910 Notebook:

`https://colab.research.google.com/drive/1f967fM1rbmAqfqAjm-2kSsQRDIrS_weW?usp=sharing`

4 Discussion

The F910 genes that we found did not perform better than the L1000 landmark genes in what we had originally hoped for. While the MSE was not significantly different when predicting using the L1000 versus our 910 "landmark" genes, we believe that they may be limited in scope. One of the major limitations of our study was that we weren't able to delineate between cell types and avoid biases that could have risen due to over representation of a particular type. We tried performing t-SNE to cluster cells however once we expanded the dataset we were unable to generate tSNE clusters with confidence. Another major limitation is the use of scRNA seq data. scRNA seq is highly sensitive however it requires a great deal of preprocessing and weight optimization which we feel was required to a greater extent before we trained our model. Lastly we were able to work with only two tissue types and given the abundance of scRNA seq data available, we believe with increased training we will be able to achieve a much lower MSE with both L1000 genes and F910 genes. Despite the limitations we were able to reduce the geneset from 978 to 901 without significant loss in performance. Another strategy we hope to implement is a guided training approach where we use the L1000 as a baseline and then build on it by adding and removing genes to the dataset. Our work shows that this is promising for future work to pursue to find a similar set of landmark genes like the L1000. As mentioned earlier PanglaoDB itself is a vast repository of scRNA seq data. If further modeling could be done using systems with higher computational power we believe that this approach could make significant progress. If such a set of landmark genes can be found, it would significantly help the community by allowing for a reliable way to predict gene expression without imputation or full sequencing. Despite the dramatic decrease in cost, sequencing is still a very cost and time intensive process. Reducing the number of genes required to be sequenced can allow for more high throughput assays to be performed with different perturbagens and deletion constructs to further understand biology.

4.1 Model Selection

In our initial approach, we wanted to train transformers as our additional approach but after further research we realized that our data was not sequence based. Since transformers are traditionally used in NLP or sequence prediction tasks, it was not suitable for our use case where we aim to predict gene expressions from a smaller subset of genes. As the order of genes in the smaller subset does not affect how gene expressions will be predicted, a transformer based approach was not suitable.

4.2 Challenges

One of the biggest challenges we faced was finding a good metric for evaluating the models. Initially, we took an approach that would round the class predictions to the nearest integer and compare that against the target, but found that it was not representative of our model's performance. Given that many of the expression values were zero, any predictions between -0.49 and 0.49 were rounded directly to zero, giving us an imperfect representation of how our model was performing. We eventually decided to use a weighted MSE loss to tell us how far off we were from the true target values, with heavier weight going towards target values of zero. As mentioned previously another key challenge is handling scRNAseq data. Unlike bulk RNAseq, scRNAseq, required careful handling and a lot of preprocessing and normalization steps. While we did employ some basic normalization techniques, we found it difficult to work with the sparsity of the data.

4.3 Next Steps

To further this work, the next step would be to create larger data sets that encompass a wider range of cell and tissue types. We were only able to find a subset of 910 genes that were able to predict gene expression for 2 different tissue types. It is hard to generalize whether this will be applicable at a wider scale to all other tissues without future work. Another limitation is that the dataset has been tested on "healthy cells". The gene expression of cells vary dramatically as perturbagens are added or deletion strains are constructed. It would be interesting to see if we can maintain predictive power in perturbed tissue samples as well. In part, this is due to the size of our data set which further work will be needed to improve the quality of such models.

References

[1] Arisdakessian, C., Poirion, O., Yunits, B. et al. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol 20, 211 (2019). https://doi.org/10.1186/s13059-019-1837-6

[2] Franzen, Oscar. "ScRNA-Seq Data." PanglaoDB, panglaodb.se/

[3] Krishnaswamy, Smita. "Scprep Python Package." Scprep Read the Docs, scprep.readthedocs.io/en/stable/

[4] Subramanian, Aravind, et al. “A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles.” Cell, Cell Press, 30 Nov. 2017, www.sciencedirect.com/science/article/pii/S0092867417313090.

[5] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, Xiaohui Xie, Gene expression inference with deep learning, Bioinformatics, Volume 32, Issue 12, 15 June 2016, Pages 1832–1839, https://doi.org/10.1093/bioinformatics/btw074