Evaluating Deep Learning Methods in Prediction of Patients with Pediatric Crohn's Disease or Ulcerative Colitis

Pramoda Karnati (pkarnati@mit.edu)
MIT Computer Science and Engineering, 77 Massachusetts Avenue
Cambridge, MA 02139 USA

Meghana Kamineni (kamineni@mit.edu)
MIT Computer Science and Engineering, 77 Massachusetts Avenue
Cambridge, MA 02139 USA

Abstract

Keywords: Neural networks; Logistic Regression; Transfer Learning; RNA-Seq Gene Expression data.

Deep learning methods have been used widely for many tasks and have gained popularity for their successes in achieving accuracy in the tasks at hand. RNA-Seq gene expression data proves to be a challenge for these models which depend on large and clean datasets. In this paper, we describe various deep learning approaches to analyze a RNA-Seq gene expression dataset and compare it to Logistic Regression techniques. We find that a multi-layer feed-forward artificial neural network achieves the highest accuracy in predicting disease subtypes of patients in the dataset. However, we recommend that further studies with more data should be performed in order to validate the findings in this study.

Introduction

Deep learning methods have been widely used to perform a number of tasks to aid the efforts of autonomous vehicles, accurate facial recognition, and most recently, biomedical applications. Artificial neural networks have been used to perform many tasks in biomedical sciences, the most widely known being medical imaging. Deep learning models recently are being used for predictive tasks, such as using RNA-Seq data to predict diseases. The task is widely different from traditional classification techniques and might not be as straight forward, as described by Urda et. al. in their paper describing applying these methods to RNA-Seq data. This paper aims to further evaluate deep learning models on RNA-Seq gene expression data by using a dataset of patients with Crohn's disease.

The rest of the paper is organized as follows: we first describe the background and motivation behind this prediction task. Then we describe the various approaches used to analyze this RNA-Seq data, followed by the results and discussion of the results. We conclude with what we hope can be done to further this area of research.

Background and Motivation

Inflammatory bowel diseases, which include ulcerative colitis and Crohn's disease, commonly affect both children and adults, and it is sometimes difficult to distinguish between them. Given the degree to which the two diseases differ, deciding which disease the patient has is a crucial step to the patient receiving treatment. Ulcerative colitis is characterized by inflammation that affects the entirety of the colon, starting from the rectum. Crohn's disease tends to involve discontinuous inflammation that begins near the terminal ileum and continues into the colon. While many treatments are similar for the two diseases, major distinctions persist due to the different complications that can arise from the two diseases. For instance, patients with Crohn's disease require surgery more often than those with ulcerative colitis, and 14% of children diagnosed with Crohn's require surgery within 5 years of diagnosis (Rosen et al. 2015).

We hypothesized that we could predict whether a patient has Crohn's disease or ulcerative colitis through analyzing their gene expression data. RNA seq data for patients with various disease phenotypes has been widely used in both classification and clustering tasks. The first step for all studies involving RNA seq data is normalization of the data, and one method includes total read-count normalization. After normalization, a common next step is to analyze differential gene expression and use it to build a model that can classify a new sample into one of the two categories. The current tools to measure this differentiation include DESeq/DESeq2 and edgeR, both of which heavily rely on statistical and regression-based models. Many of these methods tend to involve linear regressions, such as LASSO (Rapaport et al. 2013).

We wanted to see if we could apply deep learning methods to capture differentiation between gene expression data for patients with Crohn's disease and ulcerative colitis. Currently, research investigating the potential of deep learning in analyzing RNA gene expression has returned mixed results. One study found that LASSO, a regression-based model, tends to out-perfom deep learning methods, such as neural networks when predicting phenotypes based on RNA expression data. (Rapaport et al. 2013). Yet, we were curious to see if deep learning would have different outcomes on this dataset, particularly because there is not much differential expression between gene expression data in patients with Crohn's disease and those with ulcerative colitis. If there is less differential expression in our data-set, a typical linear regression may not perform as well, and we wanted to see if other methods could perform better by capturing non-linear relationships in the data.

We also wanted to build a model that could take into account the relationships between genes in our data set. One

study generated images based on the functional information of genes as well as the gene expression data of the patients. They trained a CNN model on these images, and this model performed better than their baseline (Lopez-Garcia et al. 2020). We found this approach intriguing and wanted to see if we could replicate similar results on our data-set. We are particularly interested in how we can map similarity of function between two genes to the distance between corresponding pixels in the output image. This model could be effective for our classification task since it makes sense that a group of similar genes may all be expressed significantly less or more. With its inherent detection of spatial information, a CNN would be better able to pick up on this gene group up-regulation or down-regulation than a standard neural network or regression-based model.

Methods

Data Description and Normalization

We used an RNA-Seq gene expression dataset from an expression atlas created by the European Bioinformatics Institute (Haberman et al. 2014). The data comprised of gene expression data from the ileum in 322 patients. The samples comprised of 218 patients with Crohn's disease, 62 patients with ulcerative colitis, and 42 patients with neither disease. The original dataset contains expression values for 65217 genes across the 322 patients.

We used total count normalization and the `pygmnormalize` python package to normalize the data.

Description	Number
Total number of patients	322
Patients with Crohn's disease	218
Patients with ulcerative colitis	62
Patients with neither disease	42
Total number of genes included	65217

Table 1: Dataset Description

t-SNE and PCA

The first thing we wanted to see was whether we could visualize the high dimensional RNA-Seq expression data by applying some dimensionality reduction techniques to the data. The two techniques best suited for this task are t-Distributed Stochastic Neighbouring Entities (t-SNE) and PCA. Each of these allow us to map the higher dimensional data to 2 or 3 dimensions to see whether we can understand any relationship between the genes or look at the distribution of variables across the three different classes of the patients (Crohn's disease, ulcerative colitis, and neither disease). The python package `sklearn` was used to implement the t-SNE and PCA embeddings.

Using all 65217 genes proved to be a challenge for the t-SNE and PCA algorithms as the sheer volume of the data was too massive to be handled. In order to reduce the number of genes visualized, genes that had no expression across the 322 patients were removed. We used the most differentially expressed genes in the dataset, calculated using a Z-score for the genes, as described in the next section.

Logistic Regression

We built a few logistic regression models to serve as a baseline performance metric to evaluate the performance of deep learning models. These models take normalized RNA seq data for a patient as an input and then output a label for the patient's predicted disease. We trained all models with 70% of the original data and held out 30% of the data for evaluation. For all models, training and evaluation involved 1,000-fold cross-validation.

The first model utilized l2 regularization, and the data set used for training and testing included all the genes available in the original data set.

The second model utilized l1 regularization instead, commonly known as LASSO, and the data used included all the genes in the original data set.

The third model utilized l2 regularization, and the data set used involved only the 82 most differentially expressed genes with respect to differentiating between the pediatric Crohn's and inflammatory bowel disease patients. We calculated the Z-scores for all genes, and then selected the top 82 most differentially expressed by setting a threshold for the Z-score as 1.00231. We calculated the Z-scores with the following formula, where Y_{crohns} is the expression data for patients with Crohn's diseases and Y_{ulcer} is the expression data for patients with ulcerative colitis.

$$\mu_{crohns} = \frac{sum(Y_{crohns})}{len(Y_{crohns})}$$

$$\mu_{ulcer} = \frac{sum(Y_{ulcer})}{len(Y_{ulcer})}$$

$$Z_{score} = \frac{\mu_{crohns} - \mu_{ulcer}}{\sqrt{(\frac{\sigma^2(Y_{crohns})}{len(Y_{crohns})} + \frac{\sigma^2(Y_{ulcer})}{len(Y_{ulcer}))}}}$$

We trained the fourth model on the same data set as in the third model, but this model used l1 regularization.

By narrowing down the data set to the most differentially expressed genes in the latter two models, we hoped to determine the effect of adding noise from less differentially expressed genes on the model. We tested two types of regularization to assess which type of regularization would lead to better results.

Neural Network

Deep learning models have been used to analyze high-throughput sequencing data to some success, as described by Urda et. al. in their work describing a "a first approximation on the use of deep learning for the analysis of RNA-Seq gene expression profiles data." In our paper, we wanted to perform

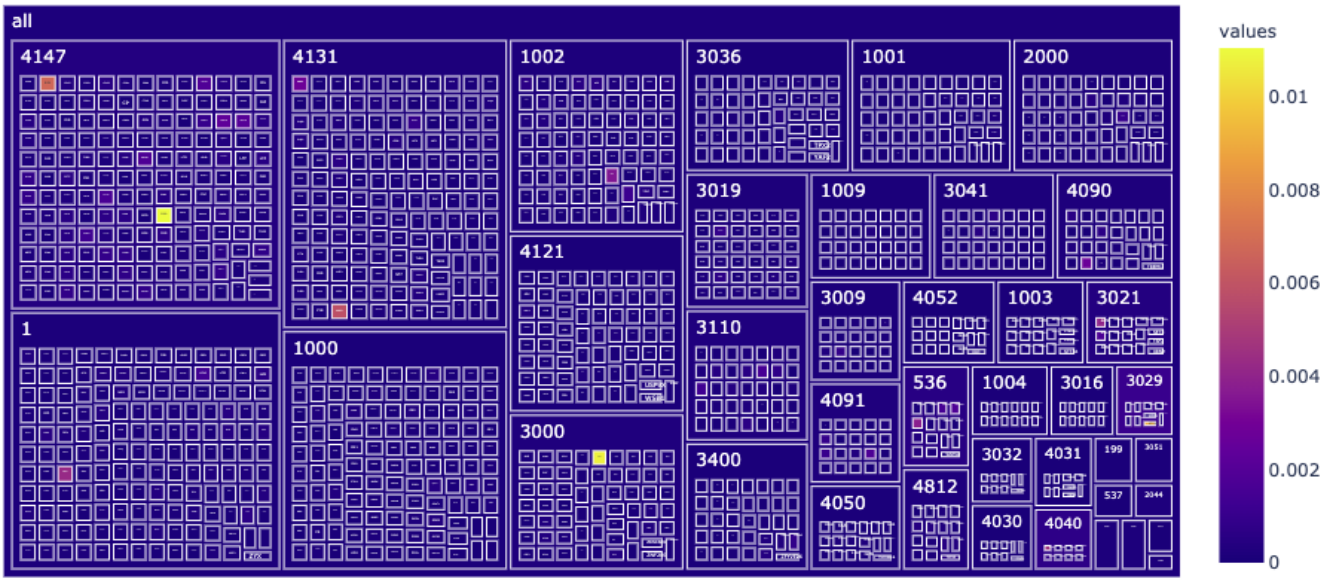

Figure 1: Example of an image of a patient's RNA-seq data generated using the Treemap function in Python's `plotly` package. The numbers and scale are included for the purposes of this paper. The numbers correspond to the functional hierarchy, while each box within the sector corresponds to a gene in that hierarchy. The box is colored according to its expression, with the scale shown to the side.

a similar approach by implementing what Urda et. al. define as *DeepNets*, a series of multi-layer feed-forward artificial neural network models to learn non-linearities between the input and output spaces.

Preparing the Data The RNA-Seq gene expression data was normalized before serving as input to the neural network. Genes that were not expressed across any of the patients were removed, and genes that were expressed an extremely small amount were removed as well.

The data was split into a training and test set through a $2:1$ ratio. 20% of the training data was used as validation data. Both the training loss and validation loss were monitored.

Model Architecture The paper describes the use of a multi-layer feed-forward neural network design to analyze the data. The model consists of multiple layers as well as regularization and dropout in order to avoid overfitting.

A subset of the series of network architectures described in the paper were tested on the data, as summarized in Table 2. The best parameters from this subset were chosen through a grid search of the parameter space.

The model was then tested on the test set to determine its accuracy on the test set. We also determined the AUC of the model and plotted the ROC curve.

Parameter	**Values tested**
Activation Function	{Rectifier, Tanh}
Number of hidden layers	{2, 3, 4}
Number of units per layer	[10, 200]
L1 regularization	[0.001, 0.01, 0.1]
L2 regularization	[0.001, 0.01, 0.1]
Dropout Ratio	[0.001, 0.01, 0.1]

Table 2: Subset of parameters used for the DeepNet architecture used to train on the RNA-seq data. The values listed are the tried values on the data.

CNN

We implemented a CNN model that takes images encoding gene expression data as an input and outputs the disease state of the patient: either Crohn's disease, ulcerative colitis, or healthy. This task involved creating images where the pixels represented gene expression data, and genes with similar functions mapped to spatially proximal pixels. We used hierarchical gene information from the KEGG database to construct a hierarchical data-frame, which we then mapped to an image using the treemap algorithm. We then trained a CNN model using these images.

Extracting Functional Gene Information In order to create images representing gene expression data, we used KEGG Brite hierarchical information as a way to place genes next to each other in the image. KEGG is a database that stores information about biological entities, including genes, and their relationships. KEGG Brite information classifies entities into functional annotation groups, such as "Genes and Proteins", and then further classifies entities by assign them a functional annotation subgroups, such as "Protein families: metabolism". Finally, each entity receives a functional annotation, such as "enzyme". Since we are

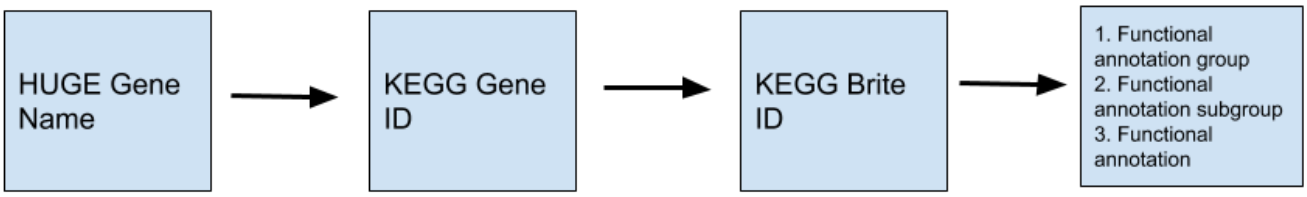

Figure 2: Mapping procedure from gene name to functional info. The original data-set includes HUGO gene names, and this mapping procedure allowed us to access information about the biological functions each gene is involved in from the KEGG database.

working with genes, we only used Brite hierarchical information for the "Genes and Proteins" functional annotation group. We extended the original gene expression dataframe of patient samples as columns and genes as rows, by adding BRITE hierarchial information - functional annotation groups, functional annotation subgroups, and functional annotations. In order to do so, we first mapped the HUGO genes names in our original data set to KEGG gene IDs using this file (http://rest.kegg.jp/list/hsa). We then mapped the KEGG gene IDs to the gene's Brite ID using this file (http://rest.kegg.jp/link/hsa/brite). Next, we made a request to the KEGG database to extract the BRITE information for these BRITE IDs using the KEGG API. This file (http://rest.kegg.jp/get/br:br08902) contains this hierarchy information. This mapping procedure only identified BRITE IDs and corresponding hierarchical information for 14118 genes out of the original 65217 genes. As 14118 gene expression data still provides a significant amount of information, we proceeded to our next step of generating images from this dataframe containing both gene expression data for each sample and hierarchical information for each gene.

Generating Images with Treemap After creating the hierarchical mapping from for each gene to the functional hierachies, we created 'images' for each patient. The python package `plotly` was used for this purpose. The package provides an easy interface through the object to map hiearchical data into a Treemap structure.

For each Treemap, the gene names were passed in as variables. The corresponding Kegg BRITE IDs were passed in as the parents, mapping to one of the 40 functional groups found in our dataset. The expression value of each gene was used to determine the shade of each of the pixels, which those expressed highly correspoding to lighter pixel shades. An example of a patient's treemap can be seen in Figure 1. First, we attempted to generate images for each patient using all 65217 genes. However, this was nearly impossible with our computational power. Therefore, we decided to again use only the most differentially expressed genes, resulting in about 1314 total genes per each patient.

Layer	**Description**
Convolutional	32 units, kernel shape = (3 * 3), relu activation
Max Pool	pool size = (2*2)
Convolutional	32 units, kernel shape = (3 * 3), relu activation
Max Pool	pool size = (2*2)
Convolutional	64 units, kernel shape = (3 * 3), relu activation
Max Pool	pool size = (2*2)
Flatten	N/A
Fully Connected	128 units, relu activation
Dropout	dropout rate TBD
Fully Connected	3 units, relu softmax

Table 3: Convolutional Neural Network Architecture

CNN Architecture We trained a CNN model to classify patients as having either Crohn's disease, ulcerative colitis, or no disease.

First, we processed the images and created our training and testing sets. The original images contained important spatial information, so no transformations were implemented. The pixels of the images were scaled from the original range of 0 to 255 to 0 to 1. We randomly sampled 80% of the 322 samples to create our training data set, and the remaining 20 % of the samples became the validation data set.

Next, we created the CNN architecture as shown in Table 3. This architecture includes 3 convolutional layers, each of which is followed by a max pooling layer. The first two layers have 32 kernels each, while the last layer has 64 kernels. The model then includes a flatten layer, a fully connected layer, a dropout layer, and a final fully connected layer. The first fully connected layer comprises 128 units. The final layer contains 3 output units that correspond to the three classes that the model is making predictions for - Crohn's disease, ulcerative colitis, or neither disease.

We then utilized grid search to determine the ideal hyperparameters for this model, in order to build the CNN with the highest accuracy on the validation data set. Grid search tested different values for the dropout rate and the learning rate of

the model, which are listed below.

$$\text{dropout rates} = [0.1, 0.2, 0.5, 0.8]$$

$$\text{learning rates} = [0.1, 0.01, 1e-3]$$

We trained the CNN model with every combination of the options for these two parameters. The training lasted 15 epochs, and we chose the model that achieved the maximum accuracy on the validation data set as the best model. After identifying which parameters the model included, we retrained the model with these parameters on the training set and saved the model.

Results and Discussion

t-SNE and PCA

We first used PCA to map the data to a lower dimension. We wanted to determine whether the three different classes of patients held any correlations based on the genes expressed. We chose to keep 3 components so that we could view the data in two- and three-dimensions. The corresponding graphs for each of the dimensions can be seen in Figures 3 and 4.

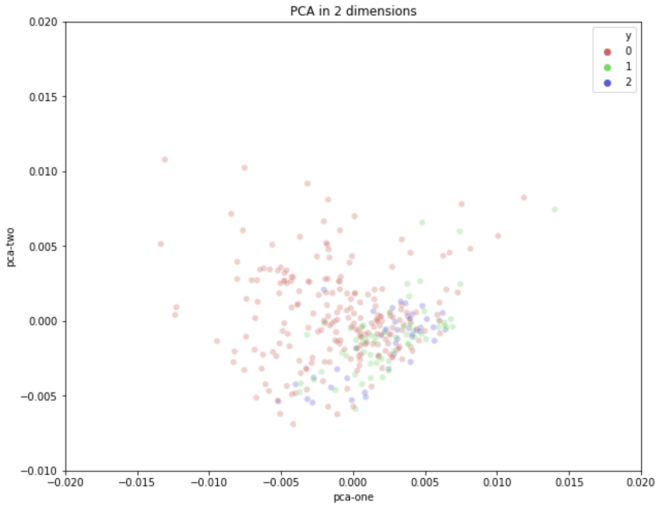

Figure 3: PCA in 2 dimensions.

We can see from these graphs that the data doesn't really seem to cluster well into the three classes. The two and three components hold some information about the three different groups, but not enough to set them apart.

We can see the same information from the t-SNE plot of the data as well, which can be seen in Figure 5. The dimensionality reduction techniques held some information about the relationships between the three groups of patients, but not enough to cluster them into their own distinct subgroups.

Logistic Regression

We determined the accuracy of the logistic regression models by finding the fraction of correct predictions made by the model. The results are shown in Table 3 below.

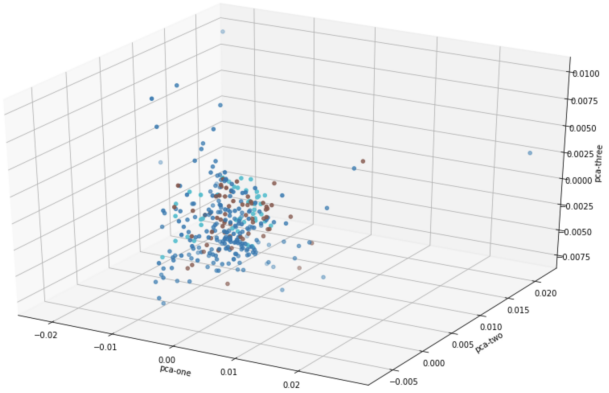

Figure 4: PCA in 3 dimensions.

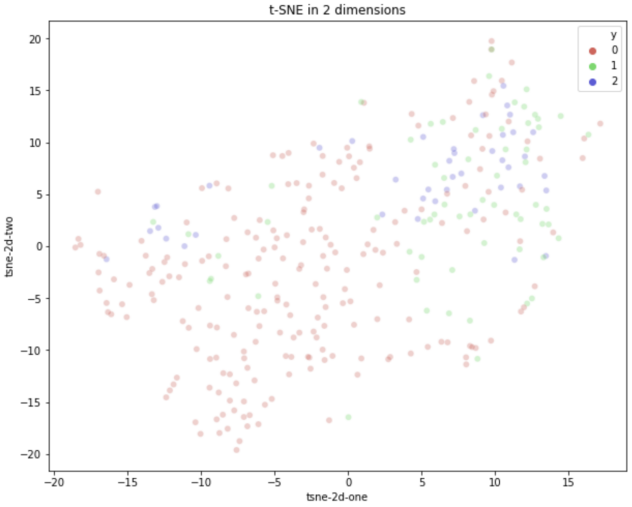

Figure 5: t-SNE in 2 dimensions.

Model Number	Regularization	Data Set	Accuracy
1	l2	all genes	0.604
2	l1	all genes	0.368
3	l2	82 most differentially expressed genes	0.608
4	l1	82 most differentially expressed genes	0.368

Table 4: Logistic Regression Model Accuracies

These results indicate that using l2 regularization leads to improved model performance. Using the 82 most differentially expressed genes out of 65217 genes does not seem to increase or decrease the logistic regression's accuracy on this data set. One reason for this observation may be that the other genes not in the top 82 genes do not add much information to the classification task. Therefore, omitting these other genes

did not change the accuracy of the model.

These results also indicate that the data does not comprise a particularly linear relationship between the gene expression data and differentiating between patients with Crohn's disease and ulcerative colitis. After running these experiments, we hypothesized that either gene expression data is not an appropriate metric to distinguish between these two disease or there may be a non-linear relationship between the gene expression data and the distinction between the two diseases. In order to investigate this hypothesis, we implemented two different deep learning models, a neural network and a CNN.

Neural Network

As mentioned in the methods section, grid search through the hyperparameters was performed to find the best model. We first attempted to search for the best combination of activation function, regularizer, regularization coefficient, and dropout rate. The corresponding validation losses are show in Figure 6.

			validation loss		
		lambda	0.001	0.01	0.1
reg type	dropout rate	activation			
l1	0.001	relu	0.508	0.574	1.235
		tanh	0.509	0.575	1.234
	0.01	relu	0.509	0.575	1.234
		tanh	0.508	0.574	1.234
	0.1	relu	0.509	0.575	1.235
		tanh	0.509	0.575	1.232
l2	0.001	relu	0.429	0.502	0.501
		tanh	0.414	0.501	0.501
	0.01	relu	0.424	0.501	0.501
		tanh	0.414	0.501	0.501
	0.1	relu	0.428	0.501	0.502
		tanh	0.415	0.501	0.501

Figure 6: Grid search of activation function, regularizer, regularization coefficient, and dropout rate with corresponding validation losses.

From this, we used the following best parameters: activation: tanh, reg type: l2, lambda: 0.001, dropout rate: 0.001. Then we wanted to search for the appropriate number of layers and hidden units, shown in Figure 7.

		validation loss
	lambda	0.001
num_layers	units	
2	10	0.436
	200	0.420
3	10	0.420
	200	0.420
4	10	0.419
	200	0.415

Figure 7: Grid search of number of layers and number of hidden units in each layer with corresponding validation losses.

Parameter	Values tested
Activation Function	{Tanh}
Number of hidden layers	{4}
Number of units per layer	[200]
L2 regularization	[0.001]
Dropout Ratio	[0.001]

Table 5: Best parameters

The best hyperparameters from the model are summarized in Table 5. We can see that the best activiation function was tanh for this task, and that a higher number of hidden units performed better on the dataset. We hypothesize that due to the high dimensionality of the data, higher layers with a higher number of units allowed the model to more accurately analyze the RNA-Seq data. Due to the potential non-linearity of the data, each additional layer was able to add more non-linearity to the model.

After training the best model on our training data, we achieved an accuracy of **84.735%** on the test set. Figure 8 shows the corresponding ROC Curve. Since this is a multi-class problem, we employ a one-vs-all approach to plot the ROC curve, assessing between no disease and disease.

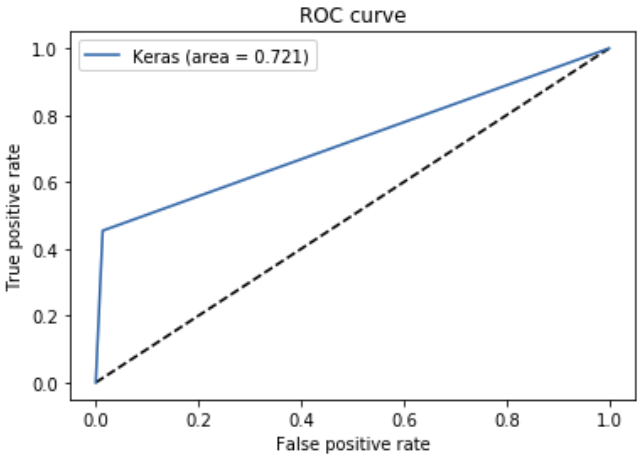

Figure 8: ROC Curve for the best model.

We hypothesize that the higher accuracy achieved from the DeepNet could be due to the nonlinearity of the data. When choosing for the genes that are differentially expressed, we find that there were not many, which could indicate a non-linear relationship in the data, proving as a possible explanation for why the neural network model was able to outperform the Logistic Regression model. We also theorize that with more data, we might have been able to achieve a higher AUC on the dataset.

CNN

The different models we trained with different hyper-parameters achieved the validation data set accuracies listed in Figure 9.

		Learning Rates		
Dropout Rates		0.001	0.01	0.1
	0.1	0.6744421721	0.6754564047	0.6713995934
	0.2	0.7180527449	0.6703854203	0.6754564047
	0.5	0.673427999	0.6764705777	0.6703854203
	0.8	0.6744421721	0.6707317233	0.6724137664

Figure 9: CNN Model Accuracies with Different Hyper-parameters

The model that achieved the best accuracy had a dropout rate of 0.1 and a learning rate of 1e-03. The increased accuracy with a dropout rate of 0.5 makes sense because having too high a dropout rate would lead to the model not taking into account enough of the training data to make predictions on new data. A small learning rate also makes sense as with a large learning rate, the model would be under-fitting on the training data.

None of the CNN models achieved better accuracy than the neural network model we created. This result is not what we expected, as we expect that by providing the model with functional information of the genes, the model would better be able to observe which biological functions may be suppressed or exaggerated in the patient samples. One explanation for this result could be the extra space in the CNN images that we generated that, although it was the same for all images, may have added noise to our model. In addition, the functional information of the genes that the images contained may have not been encoded specifically enough for it to impact the model's prediction. When creating the images, we placed all genes in a functional annotation subgroup near each other, but we did not have the information to determine which genes are more closely related within a given subgroup. The quality of the images generated might have also affected the accuracy of the model. We would recommend using a better performing Treemap generator to get a more usable image.

Future Work

In this section, we will attempt to discuss some potential future work to further studies in this area. First, we recommend that a better algorithm be used to select the genes of interest for any of the approaches described in this paper.

Regarding the logistic regression models that we created, we would like to implement similar models with different types and extent of regularization. Changing between l2 and l1 regularization drastically affected the results, so further varying regularization may also change the model's accuracy.

For the neural network model, an obvious suggestion would be to use more data for the model. The model was already able to predict disease subtypes with high accuracy, but due to the small number of samples available (322 patients), the model could have overfit to the data. The dataset also did not include enough samples of each of the three disease subtypes. Further, we think that better selection of genes of interest could provide higher accuracy for the data. Finally, we think that exploration of other model architectures for RNA-Seq gene expression data would be interesting to explore.

For the CNN we built, we would like to improve the model by training it on images that more specifically encode functional information of the genes in the data. We would also recommend using a better Treemap algorithm to generate the patient images. We think that the package used created images with a lot of noise, which might have significantly decreased the performance of the CNN.

We would also like to use all genes when creating the images, but due to time constraints of generating images, we only decided to use 1,314 genes to create the images.

For all the models, we would like to incorporate other features of these patients, in addition to the RNA seq data, to build a model that can better predict whether a patient will have Crohn's disease, ulcerative colitis, or neither. We are interested in this next step as there may not be enough information in the RNA seq data to make a good prediction about the patient's disease. The Z-scores for the genes in the data

set indicated that only one gene has significant differential expression between patients with Crohn's disease and those with ulcerative colitis, where a Z-score of 2 corresponds to a p-value less than 0.05. Without significant differential expression, it may be difficult to confidently predict the patient's disease, so incorporating other biological markers, such as vitals or past medications, that are relevant to the diseases in question could improve our models.

Acknowledgments

We would like to thank the entire course staff of 6.802/6.874 for providing us with great knowledge and support throughout the semester to understand the various applications of deep learning in the biological sciences. We would like to especially thank the TA staff, specifically Sachit Saksena, for his guidance and support throughout this project.

Contributions

Meghana normalized the RNA seq data and built the logistic regression models. She also generated the mappings between genes and their functional hierarchy information and created the hierarchical data-frame input for the tree-map algorithm. She also implemented the CNN models and performed grid search with different hyper-parameters to find the best CNN model for the data.

Pramoda performed the dimensionality reduction techniques on the data to visualize the high-dimensional gene expression data. She implemented the neural network models with various hyperparameters to train the best model on the data. She also generated the Treemap images for each patient's gene expression data to be used by the CNN.

References

B. Johnson and B. Shneiderman, "Treemaps: A space-filling approach to the visualization of hierarchical information," in Proc. Visualization '91 Conf, 1991, pp. 284–291.

Daniel Urda, J Montes-Torres, F Moreno, Leonardo Franco, and José Jerez. *Deep Learning to Analyze RNA-Seq Gene Expression Data.* pages 50–59, May 2017. ISBN 978-3-319-59146-9. doi: 10.1007/978-3-319-59147-6_5. 00001. https://core.ac.uk/download/pdf/132743527.pdf

Haberman, Y., Tickle, T. L., Dexheimer, P. J., Kim, M. O., Tang, D., Karns, R., et al. (2014). Pediatric Crohn disease patients exhibit specific ileal transcriptome and microbiome signature. J. Clin. Investig. 124, 3617–3633. doi: 10.1172/JCI75436 https://www.ebi.ac.uk/gxa/experiments/E-GEOD-57945/Experiment%20Design

Lopez-Garcıa G, Jerez JM, Franco L, Veredas FJ (2020) *Transfer learning with convolutional neural networks for cancer survival prediction using gene-expression data.* PLoS ONE 15(3): e0230536. https://doi.org/10.1371/journal. pone.0230536

Rapaport,F., Khanin,R., Liang,Y., Pirun,M., Krek,A., Zumbo,P., Mason,C.E., Socci,N.D. and Betel,D. (2013) *Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data.* Genome Biol., 14, R95. https://chagall.med.cornell.edu/RNASEQcourse/Intro2RNAseq.pdf

Rosen MJ, Dhawan A, Saeed SA. Inflammatory bowel disease in children and adolescents. JAMA Pediatr. 2015;169:1053–1060. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4702263/

Zhou, Bo Jin, Wenfei. (2020). *Visualization of Single Cell RNA-Seq Data Using t-SNE in R.* 10.1007/978-1-0716-0301-7_8.