Predicting cancer type from normalized single cell gene expressions using exchangeable neural networks

Elsa Bismuth
Columbia University & Ecole Polytechnique
eb3523@columbia.edu

Jonathan Matthews
Columbia University
jjm2270@columbia.edu

December 12, 2022

1 Introduction

Our research focused on recognizing cancer type from normalized single cell gene expression data. After which, we calculated average saliency maps to find the top ten highest contributing genes to different types of cancer. Some previous work has been done using neural networks to classify tumorous and normal cells and identify cancer types and subtypes.

In a previous paper by T. Ahn et al., "Deep Learning-based Identification of Cancer or Normal Tissue using Gene Expression Data" [1], they developed a model to discriminate between cancer and normal tissue data using various gene selection strategies. Using a deep neural network (DNN), they reached an accuracy of 0.979 in classifying cancerous and healthy data and proposed a method that can calculate a specific gene's contribution to an individual sample's cancer probability. Likewise, Mostavi, M., Chiu, YC., Huang, Y. et al. "Convolutional neural network models for cancer type prediction based on gene expression" [2] used several Convolutional Neural Network (CNN) models that take unstructured gene expression inputs to classify tumor and non-tumor samples into their designated cancer types or as healthy. They reached an accuracy of 0.95 among 34 classes (33 cancer and 1 normal) and identified biomarkers for several cancer types.

While most previous studies used bulk RNA-seq data for such tasks, we focused on single cell RNA-seq (scRNA-seq) data. We collected data from the Cancer Single

[1] T. Ahn et al., "Deep Learning-based Identification of Cancer or Normal Tissue using Gene Expression Data," 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2018, pp. 1748-1752, doi: 10.1109/BIBM.2018.8621108.

[2] Mostavi, M., Chiu, YC., Huang, Y. et al. Convolutional neural network models for cancer type prediction based on gene expression. BMC Med Genomics 13 (Suppl 5), 44 (2020). https://doi.org/10.1186/s12920-020-0677-2

Cell Expression Map database, which gives scRNA-seq gene expression matrices for tumor samples from many individuals, as well as the type of cancer each individual experienced. Although there exists common genes between each sample, each sample has differing single cells and number of cells. Thus, we were not able to utilize normal deep learning techniques. Instead, we used a specific type of neural network called an Exchangeable Neural Network (ENN) to learn from the data.

ENNs are made up of two parts: a CNN followed by a DNN. The CNN performs 1d convolutions on the data that reduces the width (number of genes) of each sample without changing the height (number of cells) of the original matrix. Then, an average function is applied to the output from the CNN to reduce the height of each sample matrix to one, essentially transforming each feature map into a feature vector. Finally, these feature vectors are fed into a DNN to perform multi-class predictions.

ENNs are also used in state-of-the-art models, as explained in the previous work of Arvaniti, E., Claassen, M. "Sensitive detection of rare disease-associated cell subsets via representation learning" [3]. This paper utilizes ENNs to process unordered multi-cell inputs for different prediction tasks. However, the model in this paper utilized bulk RNA-seq data, which does not preserve cell type proportion information like scRNA-seq data. Our data, on the other hand, utilizes scRNA-seq data and therefore retains information about the cell type proportion of each sample. This information is the driving power of our model. But, since our data and task are of a similar format to the one introduced in the paper, we used the same principles and general framework of ENNs to prepare and train our model.

2 Methods

As mentioned previously, the data was collected from the Cancer Single-Cell Expression Map database, which contains scRNA-seq gene expression matrices for many different tumor samples of different cancer types. We only selected data of which the cancer label appeared at least five times in the entire dataset. Thus, we selected approximately 114 different scRNA-seq gene expression matrices. After processing the data and performing manual feature selection (selecting only common genes between expression matrices), we found 5483 genes in common.

After performing class distribution analysis, we discovered the prevalence of each class in our data (**Fig. 1a**). As seen, the top three cancers in our dataset are Glioblastoma Multiforme, Colorectal Cancer, and Lung Adenocarcinoma, each with at least 15 occurrences. However, there was a large class imbalance in our data. So, we only selected data of which the cancer label appeared at least ten times (**Fig. 1b**). This way, the class distribution would be approximately equal, which helps reduce bias towards a certain class in our model.

It is important to note, that although we selected all data of which the cancer label appeared at least ten times, for data that appeared more than 15 times, we only selected 15 of these. Additionally, out of the 15 selected samples of GBM cancer, one had too little numeric data (orders of magnitude less than the other samples) to

[3] Arvaniti, E., Claassen, M. Sensitive detection of rare disease-associated cell subsets via representation learning. Nat Commun 8, 14825 (2017). https://doi.org/10.1038/ncomms14825

GBM	18
CRC	16
LUAD	15
AML	10
MPAL	10
PDAC	10
LUSC	7
NSCLC	7
UCEC	6
ATC	5
BCC	5
TNBC	5

(a) Initial distribution of cancer type labels.

AML	15
CRC	15
LUAD	15
PDAC	15
GBM	14
MPAL	10

(b) Final distribution of cancer type labels.

Figure 1: Cancer class distributions from scRNA-seq gene expression data, showing the number of samples that correspond to each cancer type. (a) The class distribution after the initial selection of data, showing a large class imbalance between the most occurring and the least occurring cancer types. (b) The class distribution after the final selection of data, where only classes with at least ten occurrences were chosen. This minimizes the class imbalance while still keeping a decent number of classes (six classes) for the multi-class model.

give effective results. Thus, we ended up with the class distribution seen in figure 1b.

After selecting the data, we were left with 84 samples (matrices). Then, we performed manual feature selection again and found 6865 genes in common, and discarded the other genes. These 6865 genes will be the features that are inputted into the ENN and whose contributions to cancer type prediction will be estimated.

After the final selection of data, we faced an issue where some of the sample matrices had values that could not be converted to numeric types, due to mislabeling by the open-source database administrators. As a result, we dropped all of the rows (individual cells) from each sample matrix where the values could not be converted to numeric types, since these values could not be imputed. This pre-processing only removed a minimal amount of data from our dataset, and each sample matrix was left with at least 600 individual cells each.

We then split the data into training, validation, and test sets. Due to the small size of our dataset, the validation and test sets needed to be big enough to prevent validation and test accuracy from surpassing training accuracy, which could indicate unreliable results. To accomplish this, we used a 50/25/25 train/validation/test split of the data. Additionally, due to the small dataset size, we utilized a batch size of one, effectively updating the weights of the model after each training sample was fed forward through the model.

From here, we built the ENN model framework using PyTorch (**Fig. 2**).[4] Our

[4]Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library [Conference paper]. Advances in Neural Information

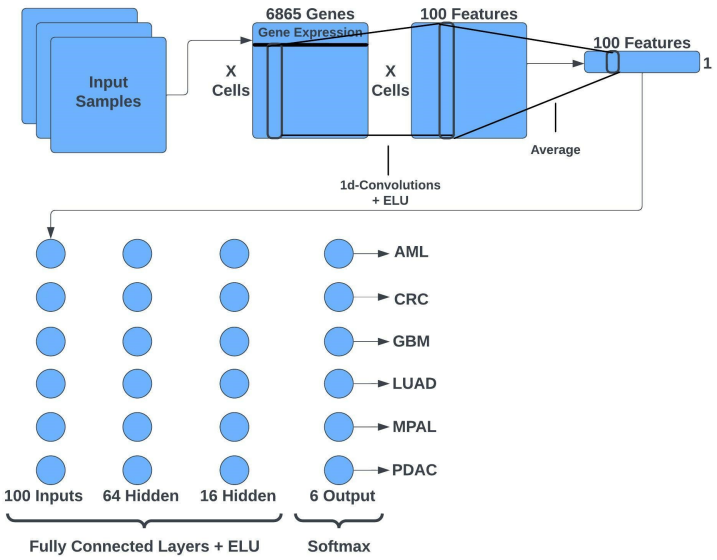


Figure 2: Architecture of the ENN. First, 1d-convolutions with ELU activation were applied to reduce 6865 genes to 100 features, while preserving the number of rows (individual cells) per sample. Then, the sample was averaged across each feature to reduce the sample to a feature vector with 100 features. The feature vectors were then fed into a DNN with two hidden layers of size 64 and 16, respectively, using ELU activation, and an output layer of size 6 using softmax activation. The six output cells correspond to the six classes present in our data.

model followed the ENN architecture described earlier in this paper, with a CNN, followed by averaging across the individual cells, followed by a DNN for classification. The CNN utilized one convolutional layer of 1d-convolutions, and an exponential linear unit (ELU) activation, to reduce the 6865 genes down to 100 features, while still maintaining the original number of individual cells per sample. After this, the samples were averaged across the individual cells to find the average value per feature, effectively reducing the data to a feature vector of size 100. The output was then fed into a DNN with four fully connected layers: one input layer of size 100, two hidden layers of size 64 and 16, respectively, and an output layer of size 6 (representing the six cancer types from our data). Each fully connected layer included ELU activation, except for the output layer, which used softmax activation for multi-class classification. Additionaly, we included a dropout of 0.3 on every layer (in both the CNN section and DNN section) except the output layer to add some regularization and generalizability to the model.

The model used the Adam optimizer to minimize validation loss. We used the early stopping technique with a patience of 5, based on the validation loss. We also used a maximum of 75 epochs to train and validate the model, followed by testing

Processing Systems 32, 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

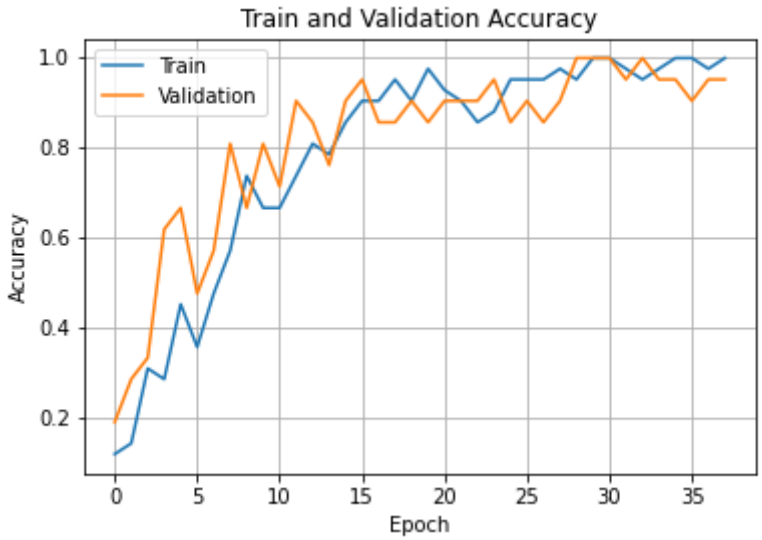


Figure 3: Tracking training and validation accuracy across epochs. After 37 epochs, the model reached a final training accuracy of 0.98 and a final validation accuracy of 0.95.

on unseen data using the model with the best validation loss.

3 Results

While training the model, we tracked training and validation accuracy across epochs (**Fig. 3**). Our model converged after 37 epochs, and the training and validation accuracies both reached an approximate asymptote. The final training accuracy of the model was 0.98 and the final validation accuracy was 0.95. After testing the model on the held out test set, the model achieved a test accuracy of 0.95.

To analyze the effectiveness of the model, and to verify the results, we compared the results to a baseline model we created. This baseline model utilized the same architecture as this model, with the differences being that the baseline model averaged the samples across individual cells before the CNN section of the architecture and utilized a dropout regularization of 0.2. The baseline model converged after 24 epochs, and achieved a final training accuracy of 0.90 and a final validation accuracy of 0.88. The testing accuracy of the baseline model was 0.88. Compared to the baseline, our ENN model performed better, doing about 7% better on unseen testing data.

4 Discussion

Our ENN model performs better than the baseline model due to the model architecture itself. Our model works by using the 1d-convolutions to extract commonality (similar gene expression values) across genes for each individual cell. Essentially, the 1d-convolutions encode data about which genes are expressed highly, along with their relative expression values, from each individual cell into 100 features per cell. Most importantly, the 1d-convolutions encode information about the cell type proportions across each sample. The cell type proportions of each sample are the driving power behind this model, since each cancer type will lead to a different proportion of cell types and subtypes in tumors. After reducing the data to these 100 features, averaging across the individual cells helps to quantify the cell type proportions across samples. Finally, the fully connected layers allow the model to learn patterns from these gene expression-based cell type proportions and classify these patterns as corresponding cancer types.

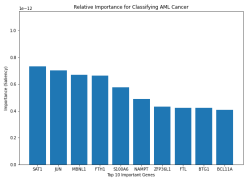
An interesting point to note is that using batch normalization in the CNN section caused the model to overfit. Normally batch normalization should not affect the results, and instead should only lead to faster convergence. However, with a batch size of 1, batch normalization leads to a high variance between samples. While this might help the model converge, it can cause the model to find patterns that are too specific to each sample, and thus cause overfitting.

To give further context into the relation between gene expression of individual cells from tumor samples and cancer type, we employed saliency maps within our model. We used the saliency maps to find the ten most impactful genes per cancer type, and compared their relative impactfulness (**Fig. 4**). We can see that for all except AML cancer, there is one gene that stands out amongst the others as the most impactful for detection. For AML cancer, however, the top four genes have approximately the same impactfulness for detection. These saliency maps show us that for our data, the following can be assumed: AML cancer is best detected with high expression of the SAT1, JUN, MBNL1, and FTH1 genes (**Fig. 4a**), CRC cancer is best detected with high expression of the CD74 gene (**Fig. 4b**), GBM cancer is best detected with high expression of the MT2A gene (**Fig. 4c**), LUAD cancer is best detected with high expression of the HSPA1A gene (**Fig. 4d**), MPAL cancer is best detected with high expression of the IFITM3 gene (**Fig. 4e**), and PDAC cancer is best detected with high expression of the IGFBP7 gene (**Fig. 4f**).

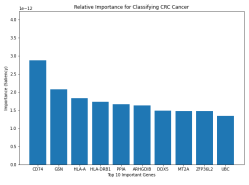
It is important to note that our dataset is too small to generalize this information about impactful genes for cancer detection; however, it can be used as preliminary results for further testing as scRNA-seq becomes more widespread.

The same sentiment goes for the ENN model proposed in this paper. While these results are a strong indicator that certain cancer types lead to patterns in cell type proporitions in tumor samples, they cannot be generalized on such a small dataset. Further experiments with more scRNA-seq data is needed before coming to a final conclusion on the relation of cell type proportion and cancer type.

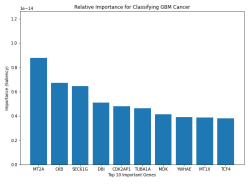
Additionally, a link to the Google Drive including our data, dataloader, baseline model, full ENN model, and figures is included in Appendix A. Note that you must be logged into your Columbia LionMail account to access the drive.



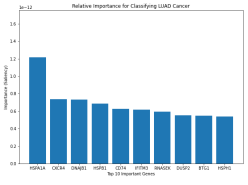
(a) Impactful genes for AML cancer prediction.



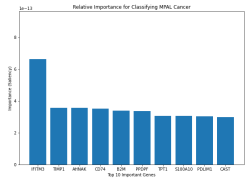
(b) Impactful genes for CRC cancer prediction.



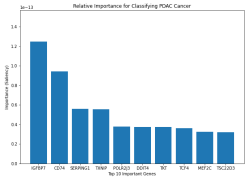
(c) Impactful genes for GBM cancer prediction.



(d) Impactful genes for LUAD cancer prediction.



(e) Impactful genes for MPAL cancer prediction.



(f) Impactful genes for PDAC cancer prediction.

Figure 4: Saliency maps showing relative impactfulness of ten most impactful genes per cancer type. (a) The top ten most impactful genes for detecting AML cancer, showing that the SAT1 gene is the most impactful, although not by a lot. (b) The top ten most impactful genes for detecting CRC cancer, showing that the CD74 gene is the most impactful. (c) The top ten most impactful genes for detecting GBM cancer, showing that the MT2A gene is the most impactful. (d) The top ten most impactful genes for detecting LUAD cancer, showing that the HSPA1A gene is the most impactful. (e) The top ten most impactful genes for detecting MPAL cancer, showing that the IFITM3 gene is the most impactful. (f) The top ten most impactful genes for detecting PDAC cancer, showing that the IGFBP7 gene is the most impactful.

References

[1] Arvaniti, E., Claassen, M. "Sensitive detection of rare disease-associated cell subsets via representation learning". Nat Commun 8, 14825 (2017). https://doi.org/10.1038/ncomms14825

[2] Li, Y., Kang, K., Krahn, J.M. et al. A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data. BMC Genomics 18, 508 (2017). https://doi.org/10.1186/s12864-017-3906-0

[3] T. Ahn et al., "Deep Learning-based Identification of Cancer or Normal Tissue using Gene Expression Data," 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2018, pp. 1748-1752, doi: 10.1109/BIBM.2018.8621108.

[4] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library [Conference paper]. Advances in Neural Information Processing Systems 32, 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Appendix A

https://drive.google.com/drive/folders/1PV-KOgox0XVEezSkvBP7QdKD1pG3ouno?usp=share_link