Single-Cell Cancer Biopsy Prediction with Transfer Learning

Danyang Shao (danyang823@gmail.com), Mica Haney (micahaney@my.unt.edu), Mani Dhakal (manidhakal@my.unt.edu)

1 Abstract

This project tackles the field of single-cell biopsy cancer identification. Currently there is not a lot of data available to train a machine learning model on, so any models trained with an end-to-end approach will not be robust or include many cancer varieties, particularly rare ones, and will possibly perform poorly as a result. This project tackles the problem with transfer learning, a method of fine-tuning that allows for the use of a large source dataset to build the initial representations of the domain data while a small target dataset tunes the model to the specific data for the task at hand. This has the potential to create a robust, well-performing model that can aid in the diagnosis of cancer types based on this biopsy technique, which could expand the usefulness of the biopsy itself. Promising results were achieved using a subset of the collected data and a dense model, indicating that future work will likely be fruitful.

2 Introduction

2.1 Cancer

Cancer is a complex group of diseases with some common attributes; they all happen when some normal cells grow uncontrollably. It is the second most cause for death in the U.S. The number of deaths is decreasing these days compared to those 10 years back but still the data is unacceptably high. According to the American Cancer Society, “a total of 1.9 million new cancer cases and 609,360 deaths are expected to occur in the US by the end of 2022, which is about 1,670 deaths a day” (*The cancer genome atlas program*.). Also, from the same American Cancer Society “there is a 32% drop in cancer death rate between 1991 and 2019 which translates to almost 3.5 million fewer cancer deaths during these years than what would have been expected if the death rate had not fallen.” Many factors played a role to bring these figures down of which early detection is one. A national priority to improve early diagnosis rates to 75% by 2028 was outlined in the National Health Service (NHS) long-term plan. Various diagnosis and testing procedures are prevalent nowadays among which biopsies is one.

Cancer classification using gene expression profiles have provided insight on the cause and the treatment of cancer. Recently, various machine learning approaches have been increasingly employed for classification and analysis of cancer disease. The gene expression data from RNA-seq technology aided the opportunity to distinguish healthy and diseased samples more accurately (Wang *et el.*, 2009). The transcriptome is the total number of transcripts and their quantity in a cell at a specific developmental stage or physiological condition (Wang *et el.*, 2009). The approach to disease studies is to quantify the changes in expression levels of each transcript during development and under different conditions. Conventional sequencing uses bulk RNA-Seq sequencing methods, which rely on the average gene expression of a cell population to reveal the presence as well as the amount of RNA in the cell sample during the measurement period.The bulk RNA-Seq method can identify differences between sample conditions (Hu *et al.*, 2016). However, there are limitations in the classification and early diagnosis of the disease.The test cells are often a mixture of diseased and healthy cells. Single cell RNA sequencing allows us to classify, characterize and distinguish each cell at the transcriptome level, which leads to identifying rare cell populations but functionally important.

2.2 Transfer Learning

As discussed in (Torrey and Shavlik, 2010; Brownlee, 2017; Seldon, 2021), transfer learning is a type of pre-training and fine-tuning where a model is pre-trained on an initial source dataset and then fine-tuned on a related but ultimately different target dataset. This process additionally involves freezing some layers of the model so that they don’t update during the fine-tuning stage and retain the weights learned during the pre-training.

(Seldon, 2021) discussed some of the situations where transfer learning is superior to end-to-end learning. Situations range from a scarcity of data for the task, reusing a pre-trained model, and faster learning. This project and the task it falls under are in the category of data scarcity. Since data for the specific task of predicting single-cell biopsy results is rare, there is not enough to train an end-to-end model on that is robust. However, pre-training on multi-cell biopsy data and transfer learning on the scarce single-cell biopsy data allows for a robust model to be trained.

3 Materials and Methods

3.1 Tools

Seurat (Cao *et al.*, 2022) is an integrated software package for single-cell data analysis developed by New York Genome Center, Satija Lab. Its functions include basic data analysis processes, such as quality control, cell screening, cell type identification, signature gene selection, differential expression analysis, and data visualization. It also includes advanced features, such as time-series single-cell data analysis and integration of single-cell data from different histologies (Cao *et al.*, 2022).

Pandas is a python library for data handling and data analytics (McKinney, 2010). After the first round of preprocessing, all further data loading and the second round of preprocessing is done using pandas.

Tensorflow is a python library designed for deep learning (Abadi, 2016). It supports a variety of model types and provides support for creating custom models from the ground-up. Tensorflow is the library used to build and train all models in this project.

The initial preprocessing of the data was done using UNT's TACC environment. The secondary data preprocessing and the model training were done using Google Colab.

3.2 Data

3.2.1 Databases

The cancer genome atlas (TCGA) was initiated by the National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI) in 2006 and contains clinical data, genomic variants, mRNA expression, methylation, and other data for various human cancers, including subtypes (*The cancer genome atlas program)*. Therefore, it is an essential source of data for cancer researchers. In our project, the muti-cell gene expression data is from TCGA, exploiting the R statistical package *GDC-client* (*Data Transfer Tool Release notes*)*,* the official data download tool provided by TCGA. We downloaded the 31 cancers from RNASeq gene expression data of interest into the source dataset and removed healthy samples. Types of cancer includes Acute Myeloid Leukemia (LAML), Adrenocortical Cancer (ACC), Bile Duct Cancer (CHOL), Bladder Cancer (BLCA), Breast Cancer (BRCA), Cervical Cancer (CESC), Colon Cancer (COAD), Glioblastoma (GBM), Head and Neck Cancer (HNSC), Kidney Chromophobe (KICH), Kidney Clear Cell Carcinoma (KIRC), Kidney Papillary Cell Carcinoma (KIRP), Large B-cell Lymphoma (DLBC), Liver Cancer (LIHC), Lower Grade Glioma (LGG), Lung Adenocarcinoma (LUAD), Lung Squamous Cell Carcinoma (LUSC), Melanoma (SKCM), Mesothelioma (MESO), Ocular melanomas (UVM), Ovarian Cancer (OV), Pancreatic Cancer (PAAD), Pheochromocytoma & Paraganglioma (PCPG), Prostate Cancer (PRAD), Rectal Cancer (READ), Sarcoma (SARC), Stomach Cancer (STAD), Testicular Cancer (TGCT), Thymoma (THYM), Thyroid Cancer (THCA), Uterine Carcinosarcoma (UCS).

The Gene Expression Omnibus (GEO) database is a gene expression database created and maintained by the National Center for Biotechnology Information NCBI (U.S. National Library of Medicine). It contains high-throughput gene expression data submitted by research institutions around the world. Our target dataset is Single-Cell RNA data of ovarian and colorectal cancers, downloaded from GEO using GEOquery (Davis and Meltzer, 2007). GEO accession number is GSE146026 and GSE132465.

3.2.2 TCGA Preprocessing

After the data was downloaded, *library (XML)* was applied for viewing patient information and generating clinical information forms. Then, in confidence that the gene id order of all files was consistent, the files were merged by column to get the expression matrix. A python script was used to create a dictionary with the sample name and TCGA sample number, and the sample name was converted into the TCGA sample number. Finally, duplicates and genes with 0 expressions were removed from the expression matrix. The 31 cancer data sets were aggregated to obtain 12,054 tumor samples from all cancers with 12,400 common genes.

3.2.3 GEO Preprocessing

For each cancer, barcodes.tsv.gz (total number of cells file), features.tsv.gz (concatenation of all cell expressed genes), and matrix.mtx.gz (coordinate file) was downloaded, and created a Seurat object. The number of genes in a cell cannot be less than 200, and the gene was expressed in at least three cells; otherwise, it was filtered out. Since low-quality cells or empty droplets usually have few genes (Mary Piper, 2020), we filtered cells with more than 2500 or less than 200 genes counted and filtered out cells with more than 5% of mitochondria. After removing unwanted cells from the dataset, the next step is normalizing the data. By default, we used the "*lognormalize*" global scaling normalization method, which normalized the gene expression value of each cell by the total expression value and multiplied it by a scaling factor (default 10,000). Finally, the result was log-transformed, and the normalized data was stored in a file.

3.2.4 Gene Preprocessing

For the multi-cell biopsy data, samples from healthy individuals were removed from the data since the single-cell biopsy data did not have samples from healthy cells. All preprocessing was done in a manner that maintained class balancing within but not between the multi-cell and single-cell datasets.

For the single-cell biopsy data, the colorectal cancer file was 5.72 GB, and could not be loaded into Google Colab’s memory all at once and required special processing before further preprocessing could be done. In order to transpose the file so that the samples were changed from columns to rows, the file was read in chunks and saved into smaller files such that each sample had 500 genes in each file. These files were then iteratively read, and new files were created such that each sample had one file and the genes were written to the file of the sample they belonged to. The file reading and writing was done such that the order of the genes between the files was maintained. Samples were then combined and written to .csv files in batches of 10 samples.

For both multi-cell and single-cell samples, each sample having all 0 values is removed from the sequence. Then, if the sequence was less than 8,000 values long, the sequence was concatenated with itself and a spacer value of 0 (e.g. [seq] + [0] + [seq]). This was repeated until the length of the sequence was over 8,000. The next operation was to divide the sequence into subsequences of a length of 200. If the last subsequence had fewer than 200 values it was discarded. This technique was inspired by (Mock *et al.*, 2021), which experimented with taking subsequences of viral genomes. Due to the massive sample count of the colorectal cancer data, 30,250 samples were processed before execution was ended and all other samples were discarded.

For the multi-cell biopsy data, the sequences were then sampled such that there were 1,000 batches, each with four samples from each cancer type. All other sequences were discarded.

For the ovarian cancer file, the file was in a .csv format and small enough to load and handle in one dataframe. The file was loaded once and sampled so that 10,317 batch files were saved, each containing 64 sequences of ovarian cancer. The colorectal cancer data was split into multiple files before sequence processing, resulting in 3,025 sequence files. These files were iterated over and sampled 64 sequences at a time. Each sequence sampling was combined with an ovarian cancer batch file until all batch files contained 128 sequence samples. The rest of the colorectal cancer sequences were discarded. For these single-cell batch files, the first 500 were used for training and the rest discarded.

Sample and batch discarding was done due to memory, processing power, and time concerns. Fully preprocessing the colorectal cancer file was large enough to take an exorbitant amount of time to accomplish, and given the decision to perform class balancing was unnecessary so the majority of this data was unneeded and unused. In order to train the models in a reasonable amount of time without crashing the runtime session, not all of the data could be used as not all of it would load into memory. Loading the batch files using a generator in order to significantly reduce memory and processing power demands was attempted, however this was not successfully implemented resulting in needing to load all training data into a Google Colab runtime at once, forcing large portions of the data to be discarded.

3.3 Model Architecture

Two model architectures were run, a fully-connected dense network and a LSTM network. Both maintained the same architecture shape of an input, three hidden layers of units 200, 100, and 50 respectively, and an output layer.

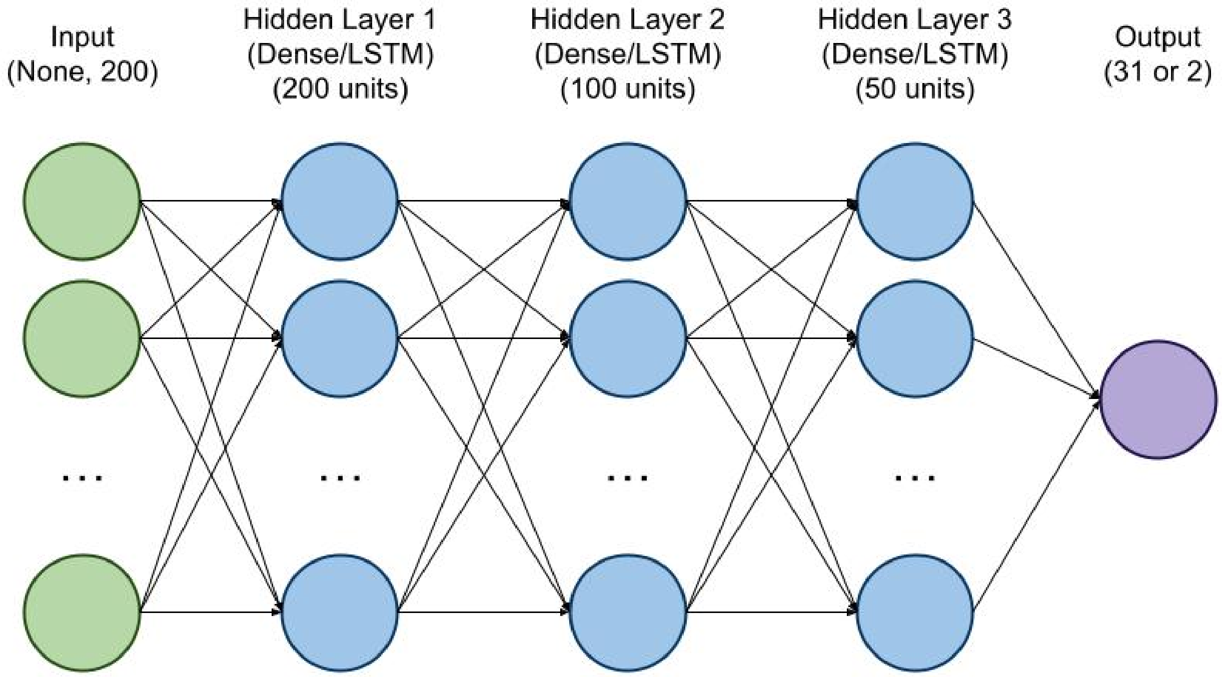

Figure 1: General model architecture, the structure of which is the same for both the dense and LSTM models.

The transfer model had the first two hidden layers frozen and the last hidden layer as well as the output layer unfrozen.

All models are compiled with sparse categorical cross entropy loss and an Adam optimizer.

4 Results

For each of the following sections, six model training runs were performed. Each architecture (dense/LSTM) had a baseline model trained end-to-end on the single-cell biopsy data, as source model trained end-to-end on the multi-cell biopsy data, and then a target model that fine-tuned the source model on the single-cell biopsy data.

4.1 Initial Results

The initial results are from training the models on 1,000 batch files of the multi-cell biopsy data and 500 batch files of the preprocessed single-cell biopsy data. The baseline model was trained for 20 epochs, and the source and target models were trained for 10 epochs each. The results are summarized in Table 1.

For both models, the baseline outperformed the transfer model. For the dense model this was only by a little for all scores but accuracy, where the baseline significantly outperformed the transfer model. The LSTM model, however, had the baseline significantly outperform the LSTM.

The dense model saw poor performance on the source model which drastically increased after transfer learning, indicating that this technique has potential to be effective at prediction. While for this data that means little, for cases with rare diseases that have little available data end-to-end training isn't possible and transfer learning is the only effective method of training a predictive model.

However the LSTM model did not share this improvement. Recall, F1, and accuracy all increased after transfer learning, but precision went to 0. Given the low baseline and source model scores as well as the lack of notable improvement and low scores of the target model, it is likely that the LSTM model needs more data to be effective. Increasing the training epochs may help as there was slow score improvement over training. Overall though, the reduction of precision to 0 and the poor performance when compared to the baseline indicate that time would be better spent looking at other architecture types.

Architecture	**Model**	**Precision**	**Recall**	**F1**	**Accuracy**
Dense	**Source**	0.000991	0.032258	0.001923	0.030726
Dense	**Target**	0.979746	0.978288	0.978567	0.471563
Dense	**Baseline**	**0.999846**	**0.999842**	**0.999844**	**0.999844**
LSTM	**Source**	0.020670	0.037464	0.010124	0.038629
LSTM	**Target**	0.000000	0.246953	0.500000	0.330614
LSTM	**Baseline**	0.787013	0.786902	0.786712	0.786719

Table 1: Summary of the initial results.

4.2 Secondary Results

The secondary results are from training the dense model on 10,000 batch files of the multi-cell biopsy data and 1,000 batch files of the preprocessed single-cell biopsy data. The LSTM model was not trained due to time constraints and the poor performance in the initial results. The baseline model was trained for 40 epochs, and the source and target models were trained for 20 epochs each. The results are summarized in Table 2.

The final baseline model is achieving near-perfect scores, and in regards to all metrics but accuracy the target model is behind by only the slimmest of margins. Again the accuracy score is oddly low, and without significant investigation it is hard to say why this is occurring.

Architecture	**Model**	**Precision**	**Recall**	**F1**	**Accuracy**
Dense	**Source**	0.001045	0.032258	0.002024	0.032395
Dense	**Target**	0.998819	0.998840	0.998828	0.503750
Dense	**Baseline**	**0.999921**	**0.999923**	**0.999922**	**0.999922**

Table 2: Summary of the secondary results of the dense model.

5 Discussion

The initial results showed a lackluster performance from the LSTM model. This could be due to a number of factors: training time, not enough data,etc. Given the decent scores of the baseline model it is clear that this architecture can be fit to the data with decent success, yet the transfer learning model failed rather spectacularly indicating that something about this setup is far from ideal for this task.

The initial results from the fully connected dense mode are far more encouraging. The baseline model scores very highly across the board, and for precision, recall, and F1 the final transfer learning model isn't very far behind, though there is an interesting and currently unexplained low accuracy. The secondary results with the dense model follow this pattern, only with higher scores across the board and a smaller absolute difference between the target and baseline scores.

While these scores are encouraging, the fact that there are only two classes for the target model is a concern. It is possible that the lack of diversity in the target dataset is inflating model scores due to high disparity in the distribution of data between the two classes. In the future one of the main focuses should be on collecting more single-cell RNA biopsy data of different cancer types in order to build a classifier with a larger output space. Additionally, all data from healthy individuals was removed for this project. Future work should include adding this data back in, so that the model works at diagnosing if cancer is present or not, instead of just which kind of cancer.

6 Conclusion

This project investigated the potential of transfer learning for building a predictive classifier for single-cell RNA cancer data. Two model architectures were tested, and initial and secondary results from the dense model are encouraging and indicate that this technique could be an excellent way to train models with the currently scarce single-cell biopsy data.

7 References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283).

American Cancer Society: Cancer Facts & Statistics. American Cancer Society | Cancer Facts & Statistics. (n.d.). Retrieved December 16, 2022, from https://cancerstatisticscenter.cancer.org/

Brownlee, J. (2017). A gentle introduction to transfer learning for deep learning. *Machine Learning Mastery*, *20*.

Cao, Y., Fu, L., Wu, J., Peng, Q., Nie, Q., Zhang, J., & Xie, X. (2022). Integrated Analysis of multimodal single-cell data with structural similarity. *Nucleic Acids Research*. https://doi.org/10.1093/nar/gkac781

Data Transfer Tool Release notes. GDC Docs. (n.d.). Retrieved December 16, 2022, from https://docs.gdc.cancer.gov/Data_Transfer_Tool/Release_Notes/DTT_Release_Notes/

Davis, S., & Meltzer, P. S. (2007). GEOquery: A bridge between the gene expression omnibus (GEO) and bioconductor. *Bioinformatics*, *23*(14), 1846–1847. https://doi.org/10.1093/bioinformatics/btm254

Hu, P., Zhang, W., Xin, H., & Deng, G. (2016). Single cell isolation and analysis. *Frontiers in cell and developmental biology*, *4*, 116.

Ilicic, T., Kim, J. K., Kolodziejczyk, A. A., Bagger, F. O., McCarthy, D. J., Marioni, J. C., & Teichmann, S. A. (2016). Classification of low quality cells from single-cell RNA-seq data. *Genome Biology*, *17*(1). https://doi.org/10.1186/s13059-016-0888-1

Mary Piper, L. P. (2020, February 24). *Single-cell RNA-seq: Quality control analysis*. Introduction to Single-cell RNA-seq - ARCHIVED. Retrieved December 16, 2022, from https://hbctraining.github.io/scRNA-seq/lessons/04_SC_quality_control.html

McKinney, W. (2010, June). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, No. 1, pp. 51-56).

Mock, F., Viehweger, A., Barth, E., & Marz, M. (2021). VIDHOP, viral host prediction with deep learning. *Bioinformatics*, *37*(3), 318-325.

Seldon. (2021, June 29). *Transfer learning for machine learning*. Seldon. Retrieved December 16, 2022, from https://www.seldon.io/transfer-learning#:~:text=Transfer%20learning%20helps%20developers%20take,models%20in%20an%20iterative%20way.

The cancer genome atlas program. National Cancer Institute. (n.d.). Retrieved December 16, 2022, from https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242-264). IGI global.

U.S. National Library of Medicine. (n.d.). *Home - Geo - NCBI*. National Center for Biotechnology Information. Retrieved December 16, 2022, from https://www.ncbi.nlm.nih.gov/geo/

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, *10*(1), 57-63.