Building Neural-Network Based Classifiers for Predicting Drug Responses of Single-Cell Expression Profiles of Tumor

Introduction

One of the reappearing issues in designing effective cancer treatments is Intra-Tumor Heterogeneity (ITH), which refers to variations in the genetic, phenotypic and molecular profiles within a single tumor. Solid tumors are composed of different types of cells like mesenchymal cells, endovascular cells and immune cells. The cancer cells within a single tumor as heterogeneous, and hence they are referred to as ITH. Their cell-to-cell variations are recognized in almost every type of cancer [1]. Exactly these variations make it hard to positively identify the tumor in the cell-lines. It arises from different factors like genetic mutations, epigenetic alterations, and also interaction with the tumor microenvironment [2]. Current cancer therapies treat the cancer as homogeneous, but this has a lot of impact on the treatment because it can lead to an expansion of treatment-resistant clones. The developed target drugs are working against single or multiple molecular signatures, but it is based on the diagnosis of mixed populations of cell-lines, that doesn't account for the cells' heterogeneity.

A crucial tool against this problem now is the single-cell RNA sequencing (scRNA-seq) that makes it possible to look at the transcriptional level of every single cell in each cell-line [3]. It analyses the RNA expression of single cells and provides a high resolution of the gene expression of the individual cell. It can provide a better understanding on how the gene expressions vary between the individual cells, and to explain how the cells react to different stimuli [4]. This way we could connect drug sensitivity with genomic features of the acquired single cells and overcome the complications caused by ITH.

My research was inspired by a paper that makes an attempt to create an estimation model of the drug response in a single-cell expression profile [5]. In their model they construct a Genomic Profiles of Drug Sensitivity (GPDS) ranked list using the Pearson correlation coefficient (PCC) and use it for evaluating how sensitive is a

cell towards a given drug. After they have their GPDS ranked list which serves as a list of predictive biomarkers for the importance of the gene expressions when evaluating whether the cell is sensitive or resistant towards the given drug, they use it on their model DREEP, to estimate precisely what will the drug response be for each cell.

I decided to find out whether there is a better alternative to constructing the GPDS ranked list, instead of calculating the Pearson correlation coefficient. I wanted to check if building a deep neural network model would yield better results for the predictive biomarkers of importance of the gene expression profiles. I trained the model on more than 600 cell-lines to predict their drug sensitivity. To evaluate the performance of my model, I tested it on the single-cell expression profiles of these cell-lines to see if I will get accurate predictions for the cell-line sensitivity towards the drug. The trained weights would serve me as a GPDS ranked list for predicting drug sensitivity of single-cell expression profiles of tumor.

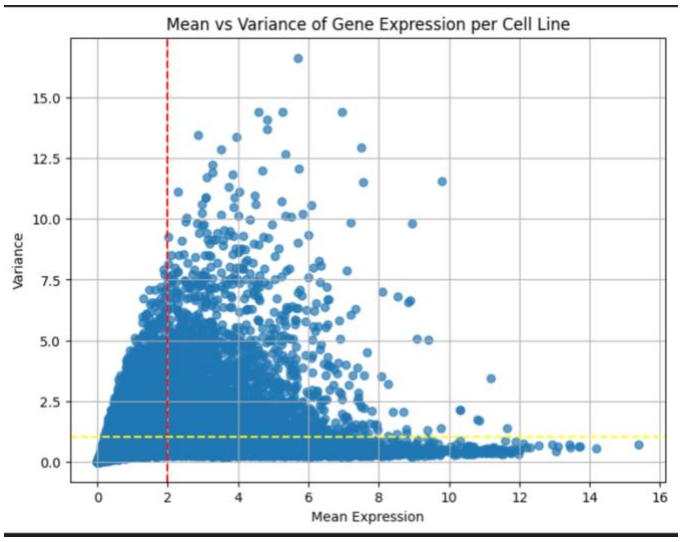

Fig.1

Methods

Data Processing

Before starting to build the classifier, I had to get all the required data, study it and preprocess it to be ready for training and evaluating. I downloaded all the data from the publicly accessible DepMap Portal (https://depmap.org/portal/). I used the Genomics of Drug Sensitivity in Cancer (GDSC2) dataset, where I have almost 800 cell lines and more than 30000 drug tests. It constructs a matrix where each element represents the sensitivity of that cell-line to that drug. I chose to work with GDSC2 instead of GDSC1 because the primer one is considered to be more accurate with the measurements. The other dataset I downloaded was the genetic expression profiles of the cell-lines, named “Expression Public 24Q4”. It contains more than 1600 cell-lines with almost 20000 genomic profiles.

First, I had to understand the data before starting to manipulate it. The GDSC2 dataset’s first column consists of the cell-lines’ ID from the DepMap portal. Later, when I had to use the cell-lines single cell expressions I had to check their ID based on their real name, but thankfully, DepMap portal made it easy for me with their search engine. The other columns respond to the drug that they used for measuring the drug potency on that cell-line and the concentration of drug they used. The second dataset with the gene expression profiles also has a first column for the cell-

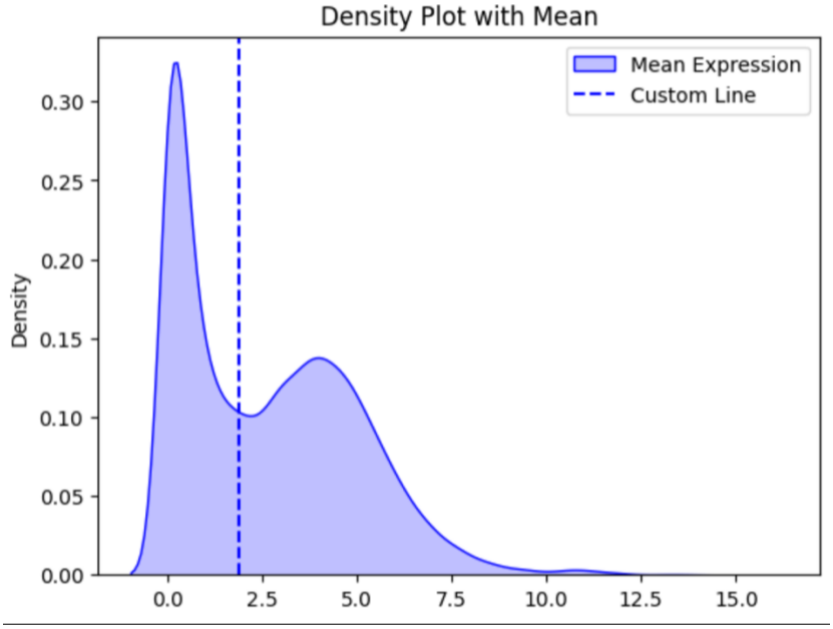

Fig.2

lines' ID, and the other columns are the gene expressions. The value in each cell of the matrix represents the correlation of the gene in that cell-line.

Secondly, I had to check for any missing values in the two datasets. The one with the gene expressions did not any missing values, while the GDSC2 had multiple, but I didn't have to deal with them at this point of the process because later I would use only one column of the dataset, i.e. only one drug with its value of concentration, so I decided to leave it like this for now.

Thirdly, I had to filter the datasets, so that I can have only the matching cell-lines in my remaining analysis. Essentially, I couldn't make predictions for the potency of a drug on a cell-line, for which I don't have its genomic profile, and vice versa, I cannot predict the potency on a cell-line for a drug, for which I don't have its actual value because I cannot check my results later on. Therefore, I left only the intersection of the two sets of cell-lines.

My next step was very crucial for my analysis because I had to find which genes from the genomic profiles' dataset are highly expressive and relevant and leave only them for my study. It is important to remove the poorly expressed genes because they introduce noise, and since they are not biologically relevant, this way we also reduce dimensionality and avoid the risk of overfitting. To find the highly expressed gene, I plotted the mean expression of the genes across all cell-lines against their variance where every dot represents a single gene (fig.1). By the figure it is clearly shown that the genes aren't generally too highly expressed, yet the least expressed must be cut off because of the reasons mentioned above. In order to

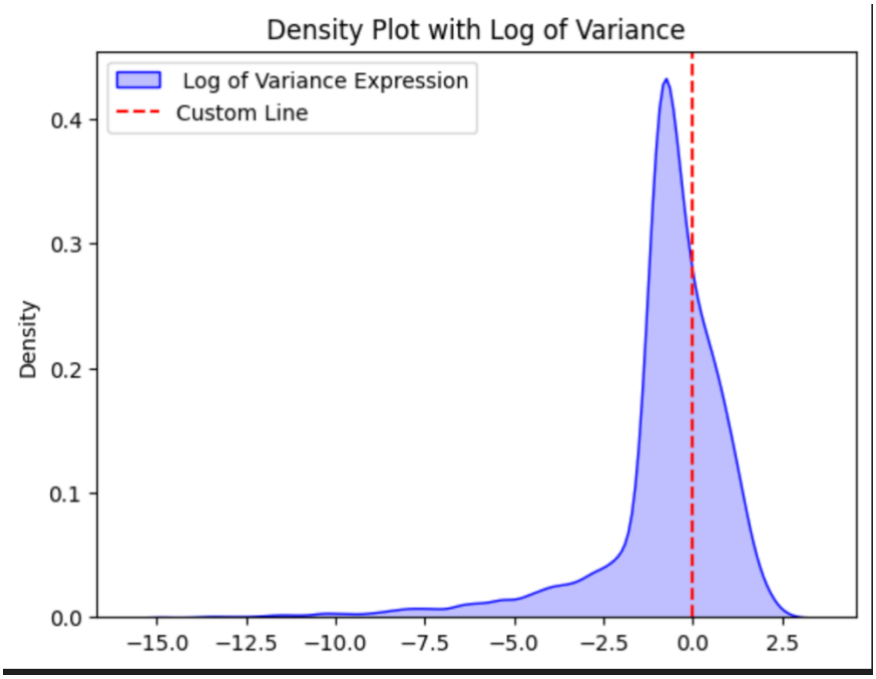

Fig. 3

decide where to put the red and the yellow line from the figure, that represent the cut we are going to make, I looked at the density plots of the mean and variance expressions of the genes and see where would be appropriate according to the distribution to cut the unnecessary genes. Figure 2 represents the density plot of the mean expression, and Figure 3 is the density plot of the log of variance. For the mean expression we can identify two distributions, so I decided to cut off one of them, meaning it will remove the genes with mean less than 1,9. I decided to plot the log of the variance because I couldn't extract as much information from its normal density plot as I did from the mean distribution. Therefore, I decide to cut the log distribution at 0, meaning 1 (since the base of my log is e) for the actual variance. On (fig.1) I have plotted the two cutting lines, and the formed upper right rectangle represents the genes that I am left with for my model. This thorough analysis was mandatory, so I can be sure that the genes on which I am going to train my model, are relevant for their cell-lines, and I don't overfeed my model with redundant data. In the end, I am left with 3072 genes for my further research. Before continuing to the next step of the exploratory analysis, I binned the data in 20 bins, which process is done to reduce noise and could also improve the model performance [6]. I did an equal frequency (quantile) binning where the logic is that each bin contains approximately the same number of values, and their values are scaled to equal the number of their bin. It is similar to ranking the values, but I do that using a fixed number of bins.

The last step from my data analysis is to choose on which drug I am going to predict the cell-lines' sensitivity. To do that, I decided to find which drug has the least missing values in my dataset, and Camptothecin had only one missing value, i.e. only one cell-line was not tested on it. Since it is only one, I decided it is better to remove this cell-line from the dataset because it wouldn't either way have too much weight on the training process.

Implementation of the DNN

Before explaining how I implemented the deep neural network, I want to summarize the data on which it is going to be trained on. I have the training data that consists of a matrix where the rows are the cell lines and the columns are the genes, and the value of each cell represents how expressed the gene is in that cell-line. The dimension of the matrix is (#cell-lines) x (#genes). For my target variable I have the column of the Camptothecin drug and all the cell-lines that are also in the other dataset, making its dimension (#cell-lines) x (1).

The goal of the deep neural network is to calculate predictive biomarkers for the importance of the gene expressions, i.e. the weights of the model will serve the purpose of biomarkers and using them to predict the drug potency of the cell-line based on its gene expression. The next step would be to check if these biomarkers are good enough to even predict the drug sensitivity based on the gene expression of the single cells in the cell-lines which is my primary goal.

First, I set a random seed at 42 to remove any unnecessary randomness in my model, so that I can make proper conclusions on my results.

Secondly, I split the data into training and testing parts, where only 5% of the whole dataset is going to be used for testing. I take such a small portion of the data, so I can make sure that the model will be trained on enough cell-lines, and by binning and removing the poorly expressed genes, I reduced the risk of overfitting the model.

Next, I scaled the training data, which is almost mandatory when one is using ReLU or tanh for activation functions. I convert the data to PyTorch tensors and define the regression model class with two hidden layers. I also implemented a grid search using Parameter Grid where I exhaustively checked for the best value of the hyperparameters: number of features in the two hidden layers, what activation function to use, the learning rate, the batch size and the number of epochs. I put a

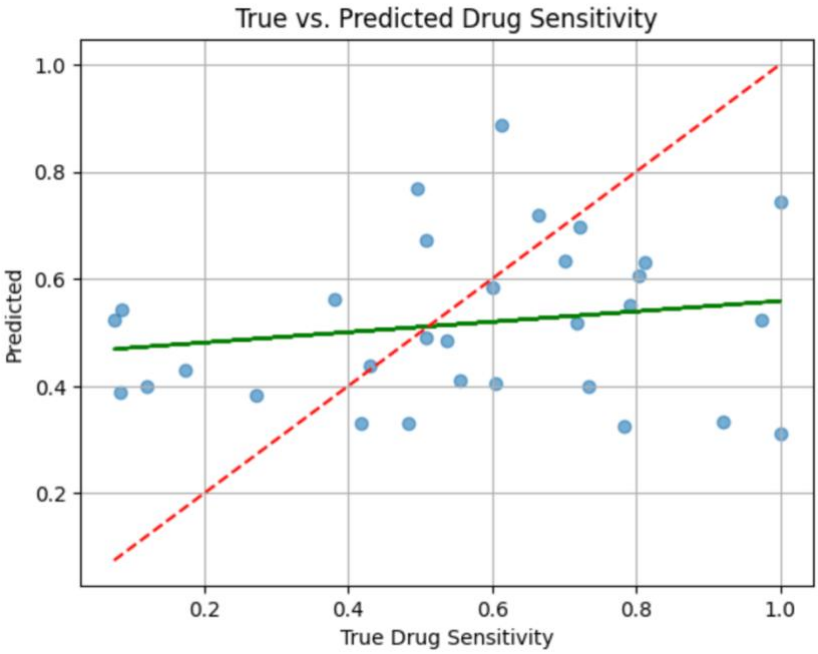

Fig. 4

check-up in the training function if the loss became NaN which would cause the stop of the training process due to recurrent issues later on with my evaluation procedure. For an optimizer I used Adam, and my parameters for accuracy of the model are the mean squared error, and the R squared score. In the end, I also print out the best hyper-parameters that my grid search has found.

Results

The purpose of the model is to come out with good, accurate predictive weights or biomarkers that would serve the purpose of a Genomic Profiles of Drug Sensitivity ranked list. My goal is to check if it can produce better results from the Pearson correlation coefficient, so the analysis of the DREEP model introduced in the paper mentioned in the Introduction can be strengthened. I train the model on 95% of the cell lines and test it on the remaining 5%.

From the best model that my grid search found, I got a great MSE score of 0.08 with $R^2 = -0.14$. Usually, when R^2 is less than 0, that means that the model is doing worse than randomly guessing, yet because the variance of the target variables is very little (0.09), it is possible to get a bad score on that metric. That is why it is better to focus on the MSE score which demonstrates that the model performs well. The best model uses tanh as an activation function, the batch size is 16, 100 number of epochs, 32 features in the first hidden layer, 16 features in the second hidden layer, and learning rate equal to 0.01.

I also decided to plot the true values against the predicted ones and draw the predicted line (the green line) and the line of perfect predictions (the dashed red line), which is considered as a good way to visualize how close are the predictions of the model to the reality (fig.4). It is visible that the green line is not close enough to the red line to safely assume that the model is doing a fantastic job, but it is on the right track to say the least. It is important to note here, that since the data is highly dimensional, it is impossible to visually present it, and the green line only serves the purpose to show how close are the predictions to the real values using a simple linear regression. Before jumping into any conclusions, it is important to look at the evaluation I did on the single-cell data that is my primary goal for this research.

Evaluation on Single-Cell Data

The evaluation part is the most important step from my research, since it will show me whether it is possible to engineer a model for constructing predictive biomarkers for gene expressions not only on the cell-lines the drug has been used, but on the cells within the cell-line for removing any probable problems caused by intra-tumor heterogeneity.

To evaluate the model, I got the single-cell data for all the cell-lines in my test set which were available to me. In the end, I evaluated my model with only four cell-lines, indeed: MKN45_STOMACH, NCIH1435_LUNG, SCC9_UPPER_AERODIGESTIVE_TRACT and SKMEL2_SKIN. The data was preserved in a special .h5ad file that I had to open using an appropriate library in python called scanpy that is designed for reading this file format, which is used for saving single-cell RNA-seq dataset. That dataset I divided into four different datasets and converted them into DataFrames, so I could more easily use them for my evaluation.

Another important step before going to the results of the evaluation, I had to find the common genes that I had for the single cells and cell-lines. Because of the unfortunate mismatch, that I didn't have the whole genetic expressions for the cell-lines, I had to remove the ones that were missing in both datasets. I filtered them, but that also meant that I trained my model on genes that were not anymore in the picture, so I had to retrain the whole model on the new, filtered genomic expressions dataset, and I was left with 2855 genes in common. The results shown before are after I used the final train dataset.

I saved the model and then loaded it again in evaluation mode to test it on the new unseen data that was actually "stored" in my test data, being the four cell-lines and their single cell expression profiles. The figures from 5 to 8 show the density distribution of the predictions from the single cells in their given cell-line.

All their distributions are very narrow and far from the target (the red line) which represents the actual value of the drug sensitivity for that cell line. These narrow distributions would propose that the predictions are confident without a lot of variances, yet completely wrong.

For the MKN45 cell-line the drug sensitivity is about 0.370, but the predictions spike at 0.390 (fig.5). Yet, a very small distribution is indeed present also at 0.370. Also compared to the other cell-lines densities and real values, the MKN45 cell-lines presents the best and closest prediction to the true value. For the NCIH1435 the model predicts something around 0.4, but the actual value is 0.9 (fig.6). The difference of 0.5 is too big and can safely say the model failed to predict the value accurately. The SCC9 cell line also is expected to have sensitivity of 0.4, but it is around 0.75 (fig.7). It is not as bad as the NCIH cell-line, but still too far from the truth. And lastly for the SKMEL2 cell-line the prediction is a bit below 0.4 with real value 1, and it is the cell-line with the most narrow density, looking almost like a Dirac delta function.

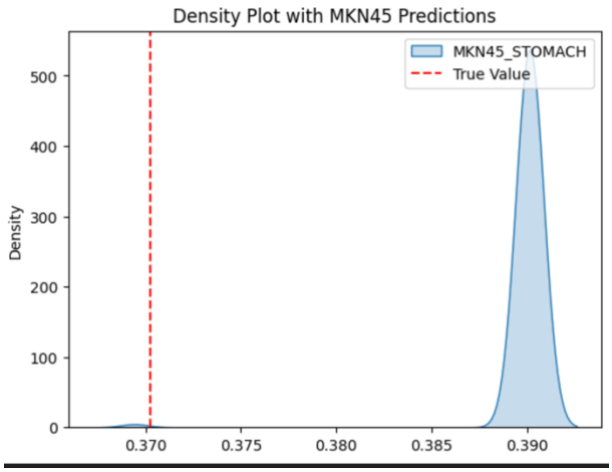

Fig. 5

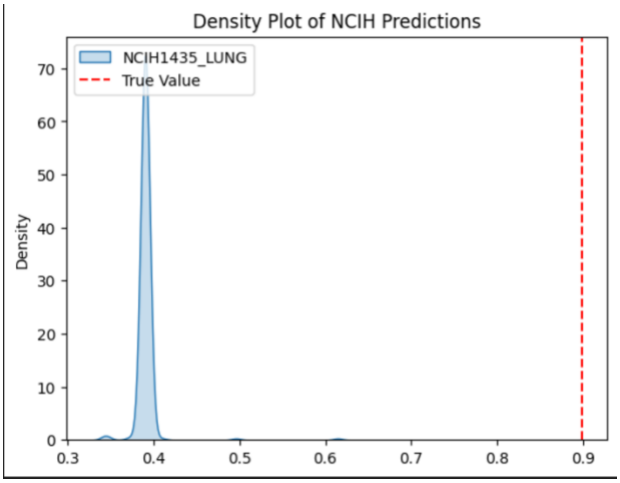

Fig. 6

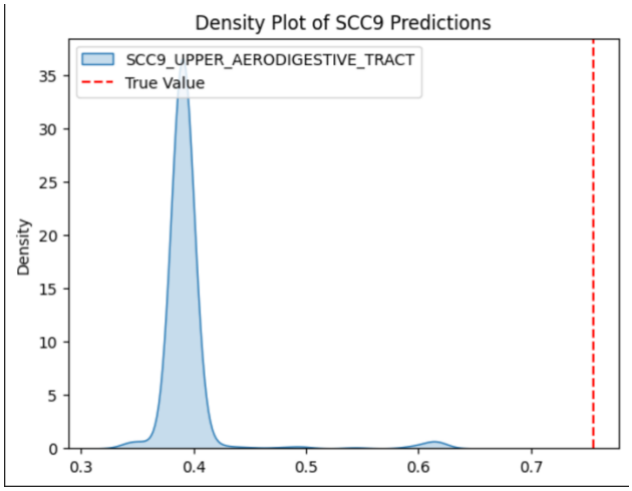

Fig. 7

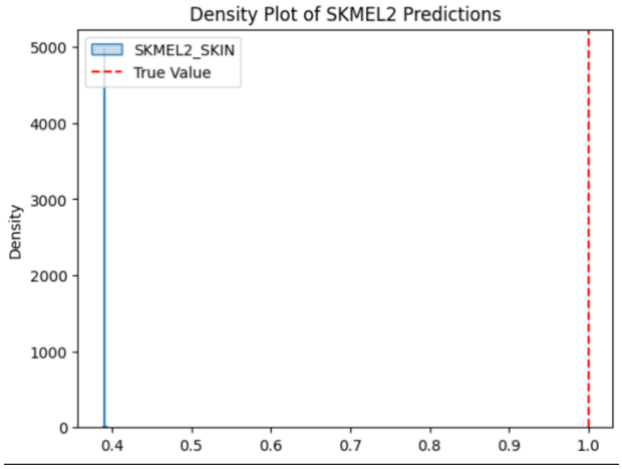

Fig. 8

Limitations

Before going to the conclusions, it is important to discuss the limitations of the project. The GDSC2 data is not enough by itself to train a whole model on which to draw any certain conclusions for the efficiency of the algorithm. Apparently, a bit over 600 cell-lines and almost 3000 gene profiles are not enough to train a model for extracting predictive biomarkers, if possible, at all. Yet, the great mean squared error, and the plot of real vs predicted values gives me hope that it is possible.

Another thing worth to mention is that the model was tested only on one drug. It is possible that Camptothecin is a drug towards which is hard in general to make conclusions about its potency on the cell-lines. Further research on other drugs is needed to make stronger conclusions.

Thirdly, the evaluation was done solely on four cell lines, due to lack of access to more single cell RNA sequencing data. If the model is evaluated on more cell-lines, it is possible to see more positive results.

Lastly, the deep neural network can also be made to looker for deeper connections by adding more hidden layers with more features, but for that purpose one needs also a lot of data storage and most importantly computational power to be able to train such a model.

Conclusion

Intra-tumor heterogeneity is one of the main obstacles nowadays for designing successful cancer treatments, and the most common step taken towards solving this problem is using single-cell RNA sequencing. I tried formulating a Genomic Profiles of Drug Sensitivity ranked list that would serve as predictive biomarkers for evaluating whether cells are sensitive or resistant towards a given drug. For this purpose, I built a deep neural network model, trained on more than 600 cell-lines and almost 3000 gene expressions in them.

In conclusion, despite the relatively strong performance of the deep neural network on the cell-line level, consisting of a low mean squared error and well-tuned hyperparameters, the model failed to generalize to single-cell predictions. This result suggests that while deep learning can identify some predictive structure in bulk RNA expression data, it doesn't manage to translate that knowledge into the single-cell level, possibly due to the noise, sparsity, and biological complexity inherent in scRNA-seq data.

However, the model also shows that constructing predictive models for drug sensitivity based on single-cell profiles is not entirely a dead end. The promising cell-line performance hints that with more training data, access to higher-resolution single-cell datasets, and improved preprocessing techniques for noise reduction and gene selection, better generalization may be possible.

Future work should explore different types of models for that purpose, which may capture relationships between genes more effectively. It is also essential to expand the training set to include other drugs as well and a more diverse range of cell types, as well as improve the evaluation pipeline using more single-cell samples.

Overall, while the current deep neural network model did not achieve satisfactory results for single-cell predictions, it offers a foundation upon which more refined approaches can be developed. With improvements in data, methodology, and computing resources, using AI to generate Genomic Profiles of Drug Sensitivity for single-cell data remains a compelling and feasible direction.

Bibliography

1. Sun, X.-X. & Yu, Q. (2015). Intra-tumor heterogeneity of cancer cells and its implications for cancer treatment. Acta Pharmacologica Sinica, 36, 1219–1227.
2. Dean, M., Fojo, T., & Bates, S. (2012). Breast cancer stem cells: Obstacles to therapy. Breast Cancer Research, 14(1), 202. https://doi.org/10.1186/bcr2310
3. Patel, A. P., Tirosh, I., Trombetta, J. J., Shalek, A. K., Gillespie, S. M., Wakimoto, H., ... & Bernstein, B. E. (2014). Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. Science, 344(6190), 1396–1401. https://doi.org/10.1126/science.1254257
4. Illumina. (n.d.). Ultra-Low Input Single-Cell RNA-Seq. Retrieved June 12, 2025, from Illumina website: “Single-cell sequencing is a powerful tool to reveal cellular heterogeneity and dynamic states such as differentiation, proliferation, and tumorigenesis.”
5. Pellecchia S, Viscido G, Franchini M, Gambardella G. Predicting drug response from single-cell expression profiles of tumours. https://pubmed.ncbi.nlm.nih.gov/38041118/
6. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.
7. The code for the project can be found at: link