Prediction of Cell Line Sensitivity to Cancer Drugs

Peyton Greenside and Winston Haynes

1. Background

1.a. Project Goal

The goal of our project is to find a machine learning model that will predict how sensitive a given cell line will be to a certain drug. A common problem in biological research that extends into clinical research is how therapeutic treatments vary across tissue type, disease state, and other variations across biological cells. For example, one drug might be very successful at inhibiting the growth of a certain type of breast cancer tumor. However, there are many different types of breast cancer that are differentiated by mutation status and tissue type and these may not respond equally to the same drug. If a particular drug works in one type of breast cancer, it would be highly useful to know if that drug would perform similarly in other types of breast cancer. Researchers and clinicians would both want to know what characteristics of a cell line may enable or prevent a therapeutic from working in a particular tissue type.

1.b. Data

In order to address this question, we are using a data set from the Cancer Cell Line Encyclopedia. This data set provides the sensitivity of 432 different human cancer cell lines to 24 different drugs. For each cell line there is comprehensive information detailing the base expression level for each gene in that cell line, copy number variations known for that cell line, as well as known oncogenic mutations. This data set is part of a larger effort to conduct a detailed genomic characterization of human cancers and make the data publicly available. When these three feature data sets are aggregated, we have 24 unique drugs on which to build a prediction model for 432 cell lines with 40,492 features.

Table 1.1: Data properties

	Data type	**# points**
Measurments	Copy number variants	23124
	Gene expression	15071
	Oncogene mutations	1667
Effect	Sensitivity to anti-cancer drugs	24

2. Dimensionality Reduction

The CCLE data exists in the m»n space, where the number of measurements is significantly larger than the number of samples. We realized early in the project that the two order of magnitude difference necessitated careful prevention of model over-fitting. Further, we wanted to enable lower-dimensional visualization of the data to identify meaningful cluster patterns.

2.a. Principal Component Analysis

Principal Component Analysis (PCA) is a commonly utilized algorithm for reducing the dimensionality by selecting orthogonal combinations of data points which minimize the variance of the underlying dataset.
PCA was implemented using the prcomp function in R. PCA reduced the data to a 432 dimensional feature space (equivalent to the number of samples). As part of the tuning process, we needed to identify the optimal number of principle components to use in our analysis.

2.b. Stochastic Neighbor Embedding

Stochastic Neighbor Embedding (SNE) is a relatively new algorithm for dimensionality reduction, which is noted for maintaining local structure while also revealing the global structure of the data set. SNE calculates a probability distribution that any two points are in the neighborhood of each other, which is then utilized to form a lower dimensional representation. In particular, we utilized implementations of t-distributed SNE (tSNE).
tSNE calculations can be computationally intensive for high dimensionality datasets. The first implementation we utilized ran on quad-core server with 16GB of RAM for over a week of constant resource consumption without producing any results. Fortunately, we were able to use the Barnes-Hut SNE algorithm which completed calculations within one minute.

2.c. Visualizing Dimensionality Reduction

We want to use our reduced dimensionality to visualize the data in a two-dimensional space. In particular, we would like a 2D visualization which exhibits clustering of similar outcomes measures to gauge how well sensitive and insensitive cancer cell lines can be distinguished. To test this, we plotted our first and second ranked principal components on the x and y axis. Then, we generated 24 graphs (one for each drug) where we colored each point according to its sensitivity to the drug.
We have displayed two cases in Figure 2.1. L685458 (left) exhibits the strongest clustering patterns of the 24 drugs. Erlotinib (right) is more typical of our results, with clustering patterns bordering on random noise. Unfortunately, even in the case of L685458, we do not see any cases where the two-dimensions alone provide enough information to segregate the data into meaningful clusters.

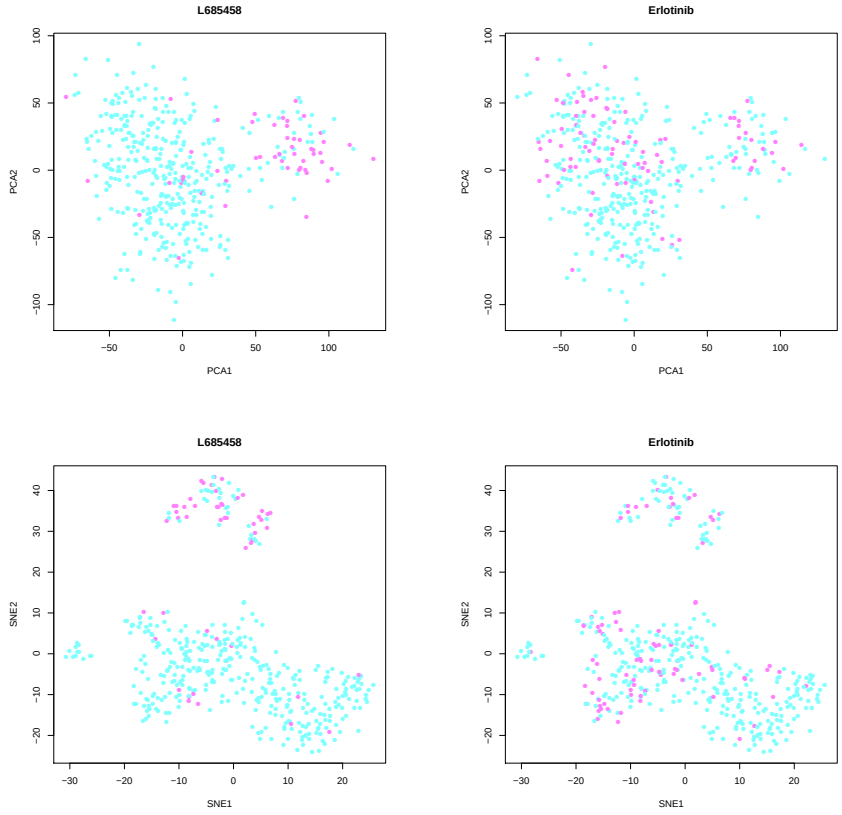


Figure 2.1: **2D visualization results.** The first and second columns are L685458 and Erlotinib, respectively. Purple= sensitive to drug, Blue= insensitive to drug. **(Top) PCA component clusters.** The x-axis and y-axis represent the first and second principal components. **(Bottom) SNE component clusters.** x and y axis represent the first and second components of the SNE dimensionality reduction.

2.d. Optimal Dimensionality Reduction

With PCA, it is necessary to select the optimal number of principal components to include in calculations. tSNE requires tuning of both the perplexity (a measure controlling the number of nearest neighbors) and the ratio of points to be used as landmarks. In order to determine the optimal dimensionality reduction methodology for our data, we examined a broad range of parameterizations of both PCA and tSNE. We looked at the performance of the different parameterizations using simple classification (K nearest neighbors) and regression (linear models) algorithms. Figure 2.2 visualizes the results for each drug.

3. Classification Methods

Instead of using continuous sensitivity data as the response, we divided the drug/cell line pairs into sensitive and not-sensitive classes using a sensitivity value threshold of 1. We randomly held out ∼10% of the data for testing and examined the effect of each algorithm on our classification accuracy.

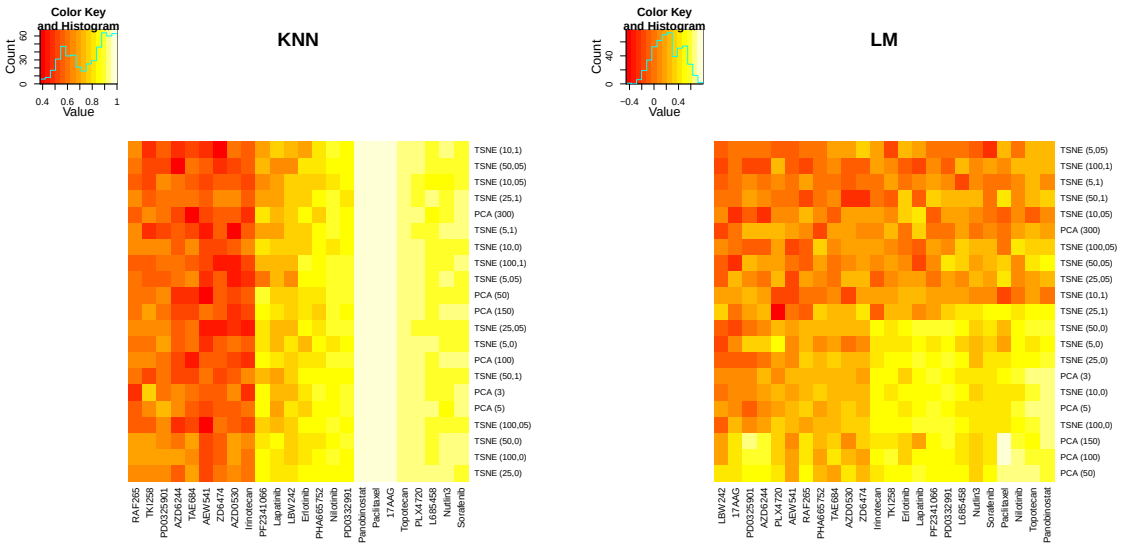


Figure 2.2: **Optimal dimensionality reduction** Columns are drugs and rows are parameterization of PCA (# principle components) and tSNE (perplexity, landmark ratio). Lighter coloring indicates a higher percent accuracy and correlation for discrete and continuous data, respectively. Rows sorted top to bottom are from worst to best performance.

3.a. K-Nearest Neighbors

As a first, and most simple, classification algorithm, we analyzed our data using the K-Nearest Neighbors (KNN) implementation from the R package FNN. KNN has a convenient intuition for follow up to dimensionality reduction as, in the ideal case, we are forming clusters which will bring similar data labels to a similar geometric location.

3.b. Support Vector Machines

Maintaining a desire to work with classifiers that have intuitive interpretations, we examined the CCLE data using support vector machines (SVMs). We hoped that the hyperplane decision boundaries formed by support vectors would nicely capture some of the visual cluster patterns. We used the SVM implementation from the R package e1071.

3.c. Naive Bayes

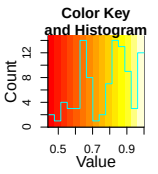
We implemented Naive Bayes also using the R package e1071 . While the independence assumptions in our data may not necessarily hold true, we found that this simple method was comparable in its performance to the other classifiers.

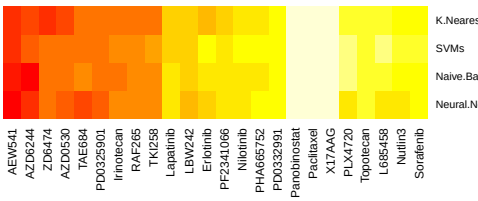
3.d. Neural Networks

We used the R package nnet to implement a discrete neural network. For this implementation we used the softmax version of the nnet package to use maximum conditional fitting. This binary implementation is a special case of the multinomial implementation of softmax. We scaled the input data such that each column has mean 0 and standard deviation 1.

3.e. Comparison

Results are shown in Table 3.1. While the SVMs performed slightly better than all other algorithms, the accuracy of the various approaches were surprisingly similar. Even at the single drug level, classifiers perform with similar levels of accuracy, where some drugs are easily classified and others confound every methodology we implemented.





Approach	Percent Correctly Classified
K-nearest neighbors	0.776
Support vector machines	0.808
Naive Bayes	0.788
Neural networks	0.775

Table 3.1: Comparison of classification algorithms. Color represents the percent accuracy of the predictions.

4. Regression Methods

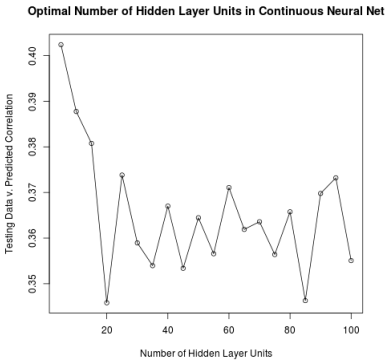
We found that despite implementing and tuning four classification methods, we seemed to reach a limit in our predictive power. As a result, we then tried to see if we could predict sensitivity values on a continuous scale. This implementation is likely more biologically useful, because in reality therapeutics are often designed to have certain dose response curves and the most strongly inhibiting state is not always the most useful to predict. We decided to assess the success of our regression algorithms by looking at the correlation between the real and predicted values. We implemented two continuous methods, a continuous version of neural networks and linear models.

4.a. Linear Models

As a simple regression model, we experimented with implementing linear models using the lm function in R. Due to their simplicity, we also utilized the linear models in determining the optimal dimensionality reduction approach.

4.b. Neural Networks

We used the nnet package in R, which uses one hidden layer. We scaled each column of our input data to fit a distribution with mean 0 and standard deviation 1. The parameter we adjusted the most to find the optimal prediction was the size of the single layer or how many units were included in the prediction. We found that a smaller layer size was more successful at prediction, but the margin between that improvement and the other implementations was minimal. As a result, we used in our final model a hidden layer with 5 units.

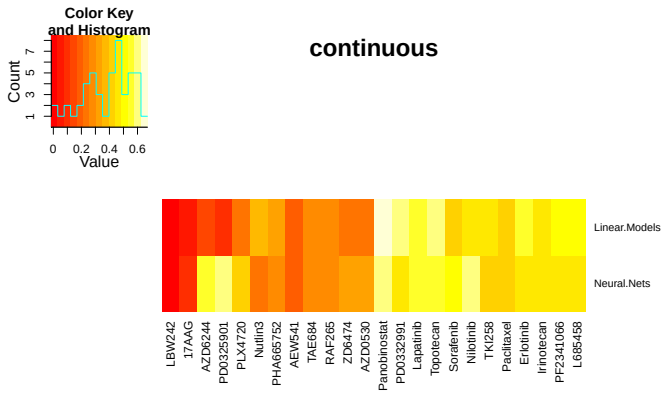


4.c. Comparison

Results are shown in Table 4.1. The neural networks slightly outperformed the linear model implementation. As with the classification algorithms, certain drugs seem to be either inherently easier or more challenging to predict.

5. Next Steps

One aspect of our model that may interfere with successful prediction is the decision to interpret missing data as sensitivity values of 0. The way our model is designed, this implies that missing values become cases where there is absolutely no growth inhibition. These values should either be transformed to be closer to the boundary between sensitive and non-sensitive states or dealt with in a more sophisticated way. We also



Approach	Avg. Correlation
Linear Models	0.365
Neural Networks	0.402

Table 4.1: Comparison of regression algorithms. Color represents the correlation of predicted with the actual data.

discovered that some drugs were more easily predictable than others. From just looking at how the sensitive and non-sensitive cell lines separate when plotted with the first two PCA components, we can already begin to see that some data are more easily separable. These drugs also often had more successful prediction rates for the models we implemented. Thus, we can focus follow-up experimental efforts on studying the predictive ability of our model on these well-characterized drugs.

6. Conclusions

The challenge of predicting cell line sensitivity to drugs has been tried many times on many similar data sets with varying features. No one has been able to find a good prediction model. We experimented with 6 different models, attempting to frame the problem both as a discrete and a continuous model. We found that the majority of methods had comparable predictive power. The accuracy of these models may help to gauge which therapeutics can be looked at more closely in given cell lines, but are by no means sufficient to address this clinical and biological research question. The predictive power of our model, as well as other similar therapeutic efficacy prediction problems, are also quite limited by what we know about genomics at this stage. We have three types of input data - oncogenic mutations, gene expression, and copy number variation - but there are many more parameters characterizing cells, and particularly cancer cells, that may be more strongly related to a therapeutic's inhibiting potential. As with most current biological models, we can work well with what data we have, but imminent biological advances will likely improve this prediction problem. However, the models we have implemented provide a success rate substantially above random that provide useful context to interpreting classes of therapeutics and gauging the general behavior of each therapeutic in a given cellular context. Predicting therapeutic efficacy is an important biological and clinical research challenge, and our results take a strong first step toward a practical research model for cancer cell sensitivity.

7. Implementation notes

A majority of data analysis was performed using the R programming language, due to the high availability of machine learning packages. SNE calculations utilized a python interface on top of a binary implementation of the Barnes-Hut-SNE algorithm.

8. Acknowledgements

We would like to thank David Knowles, who motivated our investigation of the CCLE dataset and provided us with the processed data for analysis.