A Computational Method for Predicting Cancer Drug Effectivity Using Machine Learning Techniques

Julia Christina Camacho
Jasper High School

Presented at the following competitions and symposiums in 2018 and 2019:

Dallas Regional Science and Engineering Fair (Honorable Mention in Computational Biology)
Plano District Science and Engineering Fair (1st Place in Computational Biology)
Jasper High School Science and Engineering Fair (Grand Prize in Life Sciences)
Texas Junior Science and Humanities Symposium (Invited Presenter)

Abstract:

Current cancer drug development processes that experimentally test all potential drugs on various cell lines through *in vitro* assays are tedious and inefficient. This project developed a computational method for predicting drug effectivity through IC50 values based on genomic and physicochemical characteristics of the cell lines and drugs.

Three machine learning methods were used: standard and Elastic Net linear regression, Stochastic Gradient Descent (SGD) linear regression, and deep neural networks with a mean squared error loss function and the Adam optimizer algorithm. Data from the GDSC, COSMIC, and ChEMBL databases were compiled and processed using Postgres, and genomic and drug features such as genetic mutation types and mechanisms of drug action were chosen and filtered. Exploratory data analysis was then conducted and demonstrated that the feature values cluster in patterns that differentiate the IC50 values, thus proving to be influential in prediction.

R-squared accuracy scores were calculated over four datasets: urinary tract cancer, kidney cancer, hematopoietic and lymphoid cancer, and a combined global set. The scores of the standard and Elastic Net linear regression were negative and did not converge; this was most likely due to the high sparsity and large size of the datasets. On the other hand, SGD regression and the neural network produced much more accurate predictions. On average, the neural network scores were higher and were more stable with less variation. Global models also displayed high accuracy, with scores of 0.80 and 0.95 for SGD regression and neural networks, respectively. Finally, the most significant predictive features were discovered to be the raw intensity scores of the drugs.

These computational models can be used to greatly speed up drug development processes by quickly eliminating ineffective combinations, thus considerably reducing the number of drug candidates to be physically tested in experimental assays. Finally, this method can be used to recognize overarching characteristics common to successful compounds, allowing for the creation of more effective drugs, and discover possible new targets for drugs outside the cancer types originally intended for them.

Acknowledgement of Major Assistance:

I conducted my research at home from October 2017 to February 2018, and worked by myself. A complete statement on outside assistance is included at the end of this paper.

Table of Contents

Introduction

This project developed an accurate computational method for the prediction of cancer drug effectivity based on the genomic characteristics of the cancer cell lines and the physicochemical properties of potent compounds. Three machine learning techniques were used: standard and regularized linear regression, Stochastic Gradient Descent (SGD) linear regression, and deep neural networks. These methods learn by minimizing prediction error in order to increase accuracy, and were chosen in order to recognize both linear and nonlinear possible relationships in the data.

Drug effectivity was measured by IC50 values, or the half-maximal inhibitory concentrations where drug response reaches an absolute cell viability inhibition of 50% (Aykul & Martinez-Hackert, 2016). IC50 values were chosen to measure drug effectivity because they provide a concise summary of drug suitability; the lower the IC50 value, the lower the concentration needed and thus the more potent the drug is.

Cancer drug development is a long and costly process; the average cost for a single drug is $648 million over 7.3 years (Prasad & Mailankody, 2017). This is due to the extensive experimental testing of potential drugs on various cell lines. Thus, the incorporation of alternative methods utilizing machine learning and analytical screening would be extremely helpful in narrowing fields of drug candidates before physical testing, therefore reducing both development cost and time.

Current computational methods use a variety of machine learning techniques to predict efficacy; however, there are few rigorous comparisons of the most promising methods, which was conducted in this project. Furthermore, by utilizing both genomic and physicochemical data, cancer patient heterogeneities can be better accounted for during the development of treatment. The ability to input varied values for genomic and chemical features enable accurate predictions of effectivity for different drug options, thus contributing to advancements in personalized medicine.

Materials, Methods, and Procedures

Three machine learning methods were utilized: standard and regularized (Elastic Net) linear regression, Stochastic Gradient Descent (SGD) linear regression, and deep neural networks. The deep neural networks (Keras Sequential Model) were constructed using Tensorflow, with the Adam optimizer and Mean Squared Error (MSE) loss function.

Data from three genomic and drug databases were aggregated and cleaned to form the datasets. Cleaning consisted of filtering out unwanted NaN (not-a-number) or missing data. The first database was GDSC (Genomics of Drug Sensitivity in Cancer), which provides IC50 and intensity data for each drug tested on the cell lines, such as the raw intensity values for each drug sample. Additionally, it contains cell line and drug information such as the cancer site/histology, drug name and associated synonyms, and the drug target and target pathway (Yang *et al.*, 2012). The second database used was COSMIC (Catalogue of Somatic Mutations in Cancer), which provides genomic features of the cancer cell lines such as the sequence mutation type (gain/loss), genetic mutation description, and DNA strand mutated (Forbes *et al.*, 2014). Finally, the third database used was the ChEMBL Drug Database, which provides the physicochemical properties of the drugs such as drug target, mechanisms of action, ingredients, and strengths (Gaulton *et al.*, 2016).

The programming language and libraries used to create the machine learning models were Python 3.6.3 with Pandas, Numpy, Matplotlib, Seaborn, scikit-learn, Tensorflow, and Keras libraries.

The scoring method used to evaluate the machine learning models was R-squared accuracy, which indicates the proportion of the variance of the predicted values in relation to the actual (expected) values. As the R^2 score grows closer to 1, this signifies that the model better fits the data. If it decreases toward 0, this signifies that the model fails to explain the data well. Equations 1, 2, and 3, shown below, shows the equations used to compute the score.

$$R^2 \equiv 1 - \frac{SS_{residual}}{SS_{total}}$$ where

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

This is the sum of the squared differences between expected and predicted values.

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

This is the sum of the squared differences between the expected values and the mean expected value.

Equations 1, 2, and 3.

The cancer types used to create datasets were urinary tract, kidney, and haematopoietic and lymphoid cancer. Below, Table 1 shows the number of samples and features in each dataset.

Cancer Type Datasets	**Total Number of Samples**	**Total Number of Features**
Urinary Tract	165,231	106
Kidney	447,913	110
Haematopoietic and Lymphoid	29,851	104
Global **(above datasets combined)**	670,171	110

Table 1.

Table 2, below, shows the total number of drugs and cell lines in the global dataset.

Total Number of Drugs	22
Total Number of Cell Lines	21

Table 2.

Main Procedural Steps

1) Data Cleaning, Compiling, and Preparation
   a) Assemble tables using the PostgreSQL 10 database & the SQLPro tool: inner join methods used.
      i) GDSC and ChEMBL databases matched by standard drug name.
      ii) GDSC and COSMIC databases matched by cosmic_id and sample_id.
   b) Read CSV files using SQL: import all data using read_csv()
   c) One Hot Encoding (OHE) - convert all values to continuous types using get_dummies()
   d) Standardize numerical values through scale()
   e) Create cross-validation training and testing datasets with cross_val_score()
2) Construct, Fit, and Run Linear Models
   a) Generate learning curves with 10-fold cross-validation: increments of data samples
   b) Repeat trials over cancer types
3) Construct, Fit, and Run SGD Model
   a) Generate learning curves with 10-fold cross-validation: increments of data samples
   b) Repeat trials over cancer types
4) Construct, Train, and Run Neural Network

a) Construction parameters, optimizer, loss functions: configure using compile()
b) Set Keras training parameters and number of epochs: train using fit()
c) Generate learning curves with 10-fold cross-validation: increments of data samples

Figure 1, shown below, provides a visual overview of the steps.

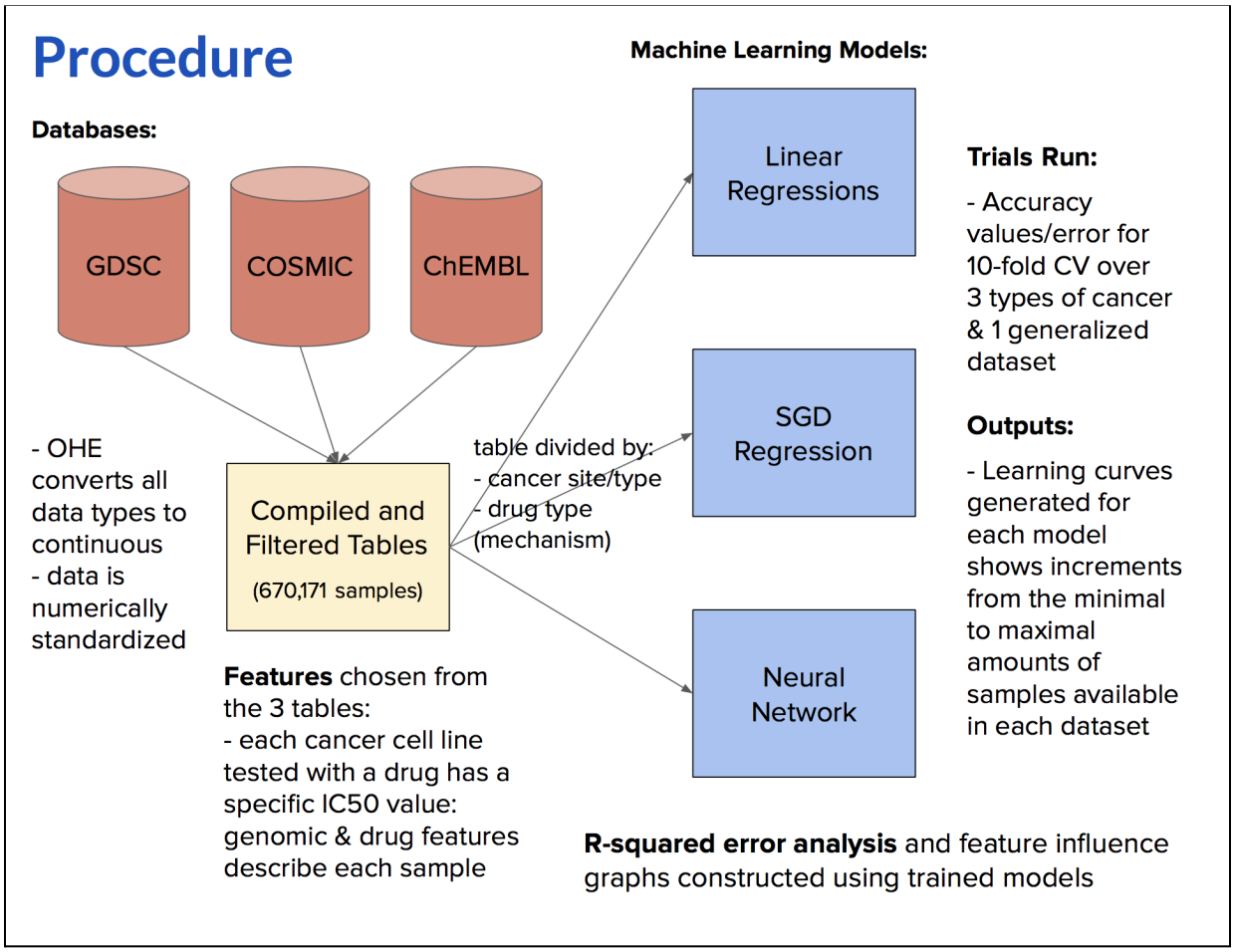

Figure 1.

In-Depth Procedure

1. Download raw data files from cancer cell line and cancer drug databases.
    1. Genomics of Drug Sensitivity in Cancer (GDSC) files:
        1. Cell_Lines_Details.xlsx (Annotated List of Cell Lines)
        2. Screened_Compounds.xlsx (Screened Compounds)
        3. (Recently Altered Chromosomal Segments (RACS) in Cell Lines)
        4. GSE68379_series_matrix.txt (DNA Methylation Data for Cell Lines)
        5. V17a_public_raw_data.xlsx (Compound Sensitivity Data for Cell Lines)
        6. V17a_fitted_dose_response.xlsx (Log (IC50) and AUC Values)
    2. COSMIC Database files:
        1. CosmicMutantExport.tsv.gz (COSMIC Mutation Data)

2. CosmicNCV.tsv.gz (Non-Coding Variants)
3. CosmicCompleteGeneExpression.tsv.gz (Gene Expression)
4. CosmicCompleteDifferentialMethylation.tsv.gz (Methylation Data)
5. CosmicStructExport.tsv.gz (Structural Genomic Rearrangements)

3. ChEMBL files:
    1. chembl_23.fa.gz (Fasta protein files)
    2. chembl_23_chemreps.txt.gz (Chemical representations)
    3. chembl_23_mysql.tar.gz (MySQL file loading)
4. STITCH files:
    1. General identification flatfiles
    2. 9606.protein_chemical.links.v5.0.tsv.gz (chemical-protein links)
    3. 9606.actions.v5.0.tsv.gz (interaction types of links)
5. Convert each file type to a .csv (comma-separated values)-type file in order to prepare the data to be uploaded to and read in the Python environment.

2. Read and upload data into Jupyter Notebook Python environment using the Anaconda Navigator program.
    1. Import necessary data processing and machine learning modules: pandas, numpy, matplotlib and tensorflow, keras.
    2. Save each read data file as an individual dataset using MySQL.
    3. Find overlapping samples / note and fill in missing data for each type of cancer (Breast Cancer, Large Intestine Cancer, Lung Small Cell Carcinoma)
        1. Use the COSMIC cell line names/IDs and drug names to compare samples in the datasets. Using the MySQL database, determine the overlapping samples and create a total dataset of all the features for the common samples of each cancer type.
        2. Divide the total datasets for each cancer type into training and testing datasets using an appropriate k-fold cross-validation outline.
3. Normalize data types
    1. Run One Hot Encoding in order to standardize the different types of data: continuous, binary, and categorical on a consistent numerical scale.
        1. Using Keras, define one_hot_labels = keras.utils.to_categorical(labels, num_classes=10). This will be used later on when plotting and learning from the data.
    2. Run a Min-Max data normalization method in order to fit the data on a level scale of 0 to 1.
4. Run linear regression models using numpy and matplotlib.
    1. Define x and y-axis arrays and append each line from the data file into the corresponding array.
    2. Use numpy to plot the arrays on a 2D graph.
    3. Predictions can be calculated using the line of best fit equation $\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_p x_p = w \bullet x = w^T x$.Use the derivative of the error minimizing equation $E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{N} (y_i - w^T x)^2$ in

order to calculate the line of best fit for the dataset. After calculating $\frac{d}{dx}\ E$and isolating w, individual weights for each feature can thus be determined by the equation$W = (x^{T}x)^{-1}x^{T}y$. (Use sklearn.linear_model.LinearRegression to implement this.)

4. Use the training data to fit the model, using model.fit(), and create learning curves. Repeat for each training dataset for each cancer type; add each additional type to the model and re-evaluate.
5. Use the testing data to evaluate the accuracy of the model (R-squared accuracy) using 10-fold cross-validation. Repeat for each testing dataset for each cancer type; add each additional type to the model and re-evaluate.
6. Repeat with the addition of a standardizing parameter using SGD linear regression. (Use sklearn.linear_model.SGDRegressor to implement this.)
7. Error and accuracy statistics for the testing datasets generated by cross-validation will be evaluated in data analysis.

5. Construct Neural Network (NN) model using Keras and Tensorflow.
    1. Create a sequential model and define the input shape for the first layer.
    2. Configure the learning process using a compile function. Set optimizer = "rmsprop" and loss = "mse." This sets up a mean squared error regression prediction process.
    3. Use the following hyperparameters: 3 hidden layers, 110 nodes per layer, 5 epochs per CV, and 10-fold cross-validation of each dataset.
6. Run the *training* dataset compiled by the cross-validation methods used above.
    1. Use Keras in the Jupyter environment to process the training datasets using the NN model, as described as follows:
        1. Make sure to take accuracy measurements (R-squared accuracy) at each interval in order to configure learning curves during data analysis.
        2. As the model learns, configure/adjust the weights by using the function model.fit(). Use labels = one_hot_labels.
        3. Repeat for each training dataset for each cancer type; add each additional type to the model and re-evaluate.
7. Run the *testing* dataset compiled by the cross-validation methods used above.
    1. Use Keras in the Jupyter environment to process the testing datasets using the NN model, as described as follows:
        1. Make sure to take accuracy measurements (R-squared accuracy) at each interval in order to configure learning curves during data analysis.
        2. Repeat for each testing dataset for each cancer type; add each additional type to the model and re-evaluate.
        3. Error and accuracy statistics for the testing datasets generated by cross-validation will be evaluated in data analysis.

Results (Data and Findings)

Results from exploratory dataset analysis showed that the features were distributed in ways that suggested promising predictive power. For instance, in the kidney dataset, drug maximal intensity values display a positive correlation with IC50 values. In the hematopoietic and lymphoid (H&L) dataset, drug mechanisms of action cluster in patterns that differentiate IC50 values. The features that cluster are thus more influential in prediction. The graphs are shown below in Figures 2 and 3.

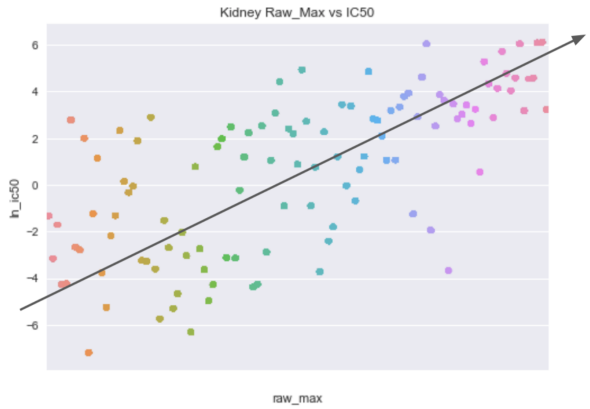

Figure 2.

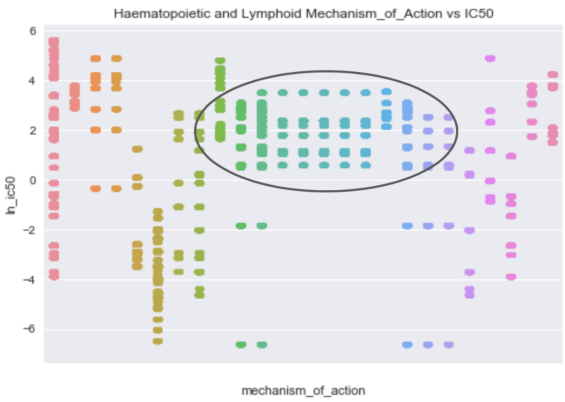

Figure 3.

Standard and Elastic Net linear regression models produced poor accuracy scores and failed to converge properly. Negative accuracy scores were produced, which signifies that the model is a worse predictor than a horizontal line. This is shown below in Figures 4 and 5.

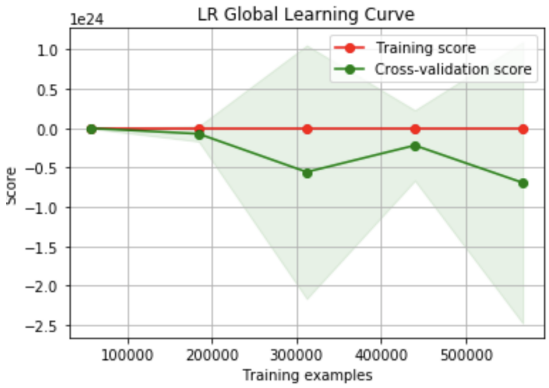

Figure 4.

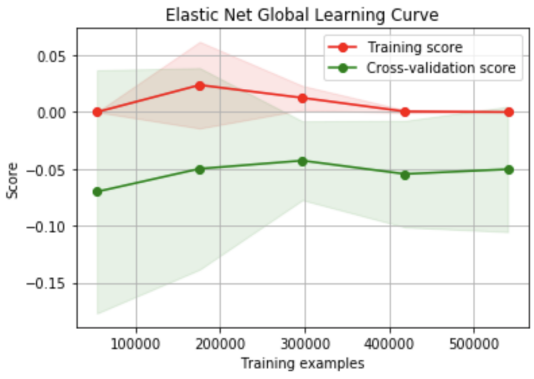

Figure 5.

The SGD Linear Regression model produced final rounded accuracy scores ranging from 0.62 to 0.95; however, there was consistently a large variance in CV scores. The urinary tract dataset produced the most accurate model. This is shown below in Figures 6, 7, and 8.

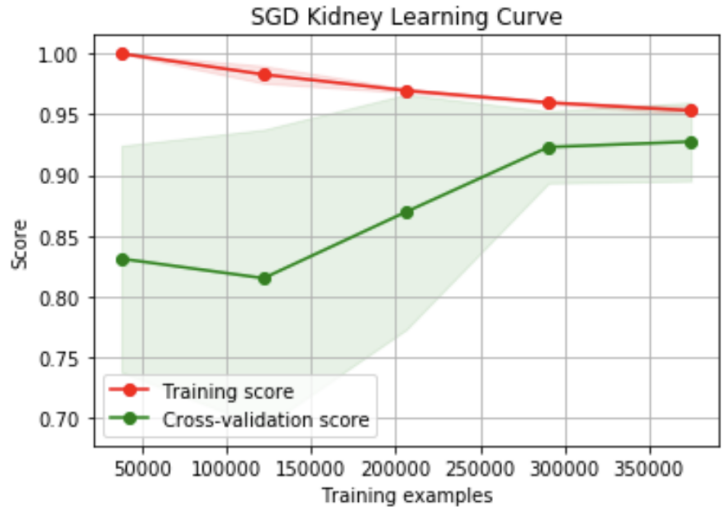

Figure 6.

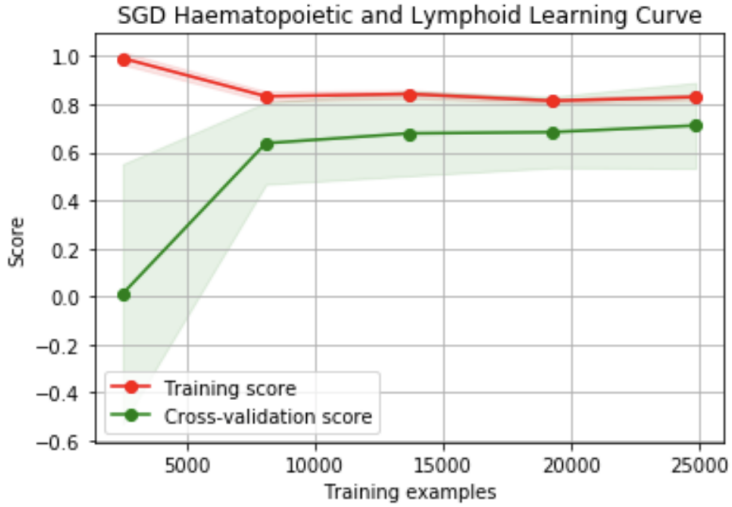

Figure 7.

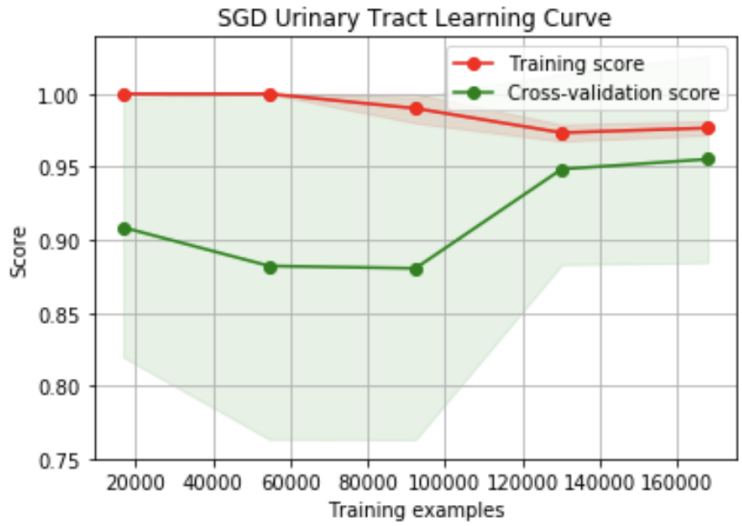

Figure 8.

The Neural Network model produced final rounded accuracy scores ranging from 0.80 to 0.99; variance in CV scores was consistent throughout the different datasets. The urinary tract dataset produced the most accurate model. This is shown below in Figures 9, 10, and 11.

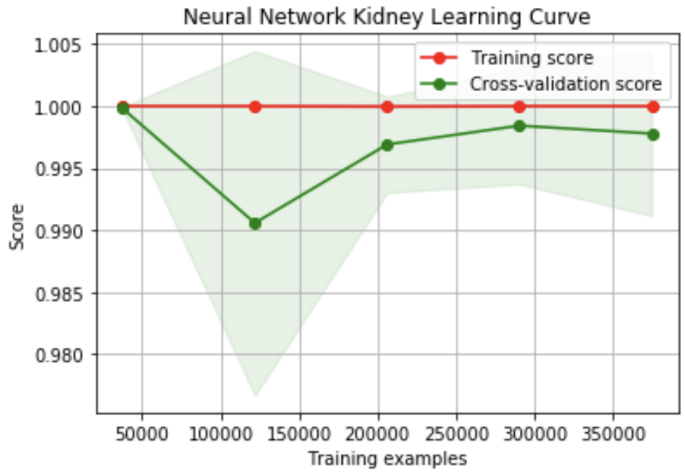

Figure 9.

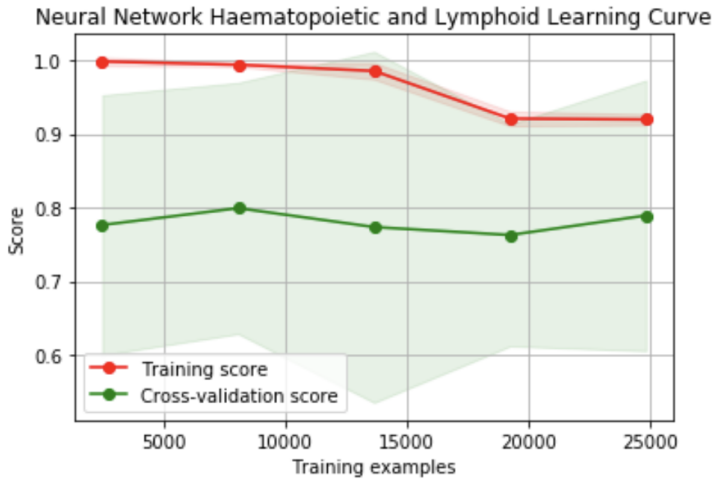

Figure 10.

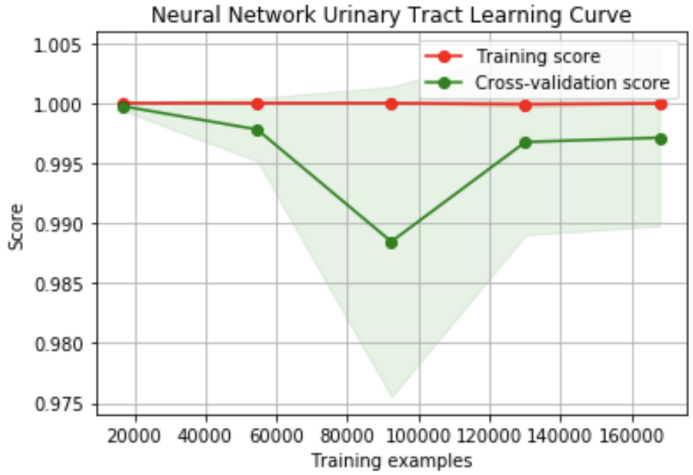

Figure 11.

For the global models (all cancer type datasets combined), convergence is achieved between the training and CV scores, which demonstrates accurate prediction ability. The final rounded 0.80 and 0.95 accuracy scores show that a global, multi-cancer model is functional and generalization can be achieved without sacrificing prediction accuracy. This is shown below in Figures 12, 13.

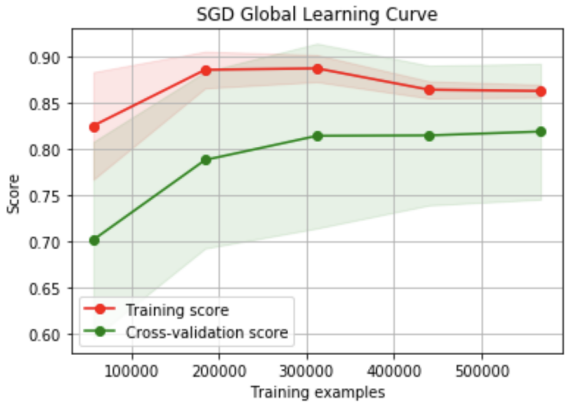

Figure 12.

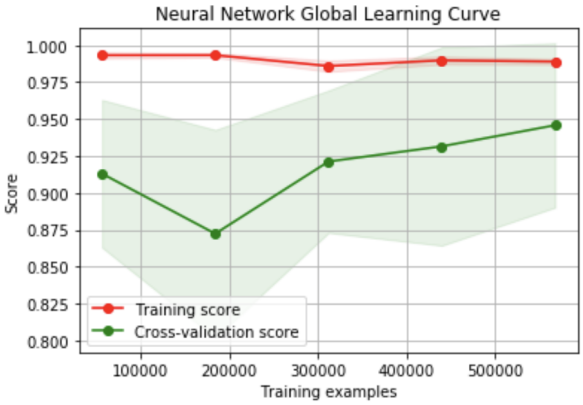

Figure 13.

On average, the neural network accuracy scores across the cancer types are higher than SGD scores (0.93 > 0.85). Neural networks produced a more stable set of accuracy scores; the average range is lower than the average range of SGD (0.23 < 0.32). Furthermore, the standard deviation in neural network scores across 10 trials is significantly lower than the variation of SGD scores (0.07 < 0.10). Below, Figures 14 and 15 show these comparisons between SGD regression and neural networks.

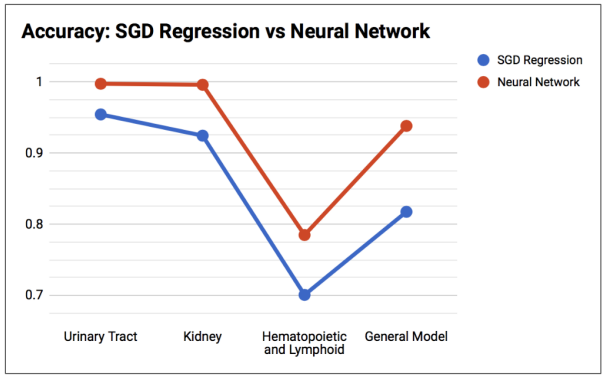

Figure 14.

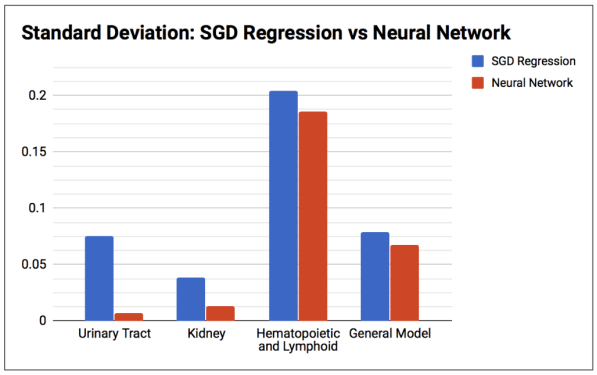

Figure 15.

The P-values obtained from statistical hypothesis testing are shown below in Table 3. The P-values from the T-tests conducted show that the accuracy differences between SGD and Neural Networks are more significant when larger datasets (Kidney and Global) are utilized in training.

	Urinary Tract	**Kidney**	**Haematopoietic & Lymphoid**	**Global**
P-Value (Neural networks vs. SGD)	0.1042752285	1.608x10^-4	0.348190281	1.80x10^-3

Table 3.

Feature coefficient value graphs display the most significant features, as shown below in Figure 16. The features most influential for prediction are indicated by their y-values (coefficient values), shown by the high spikes in the graphs.

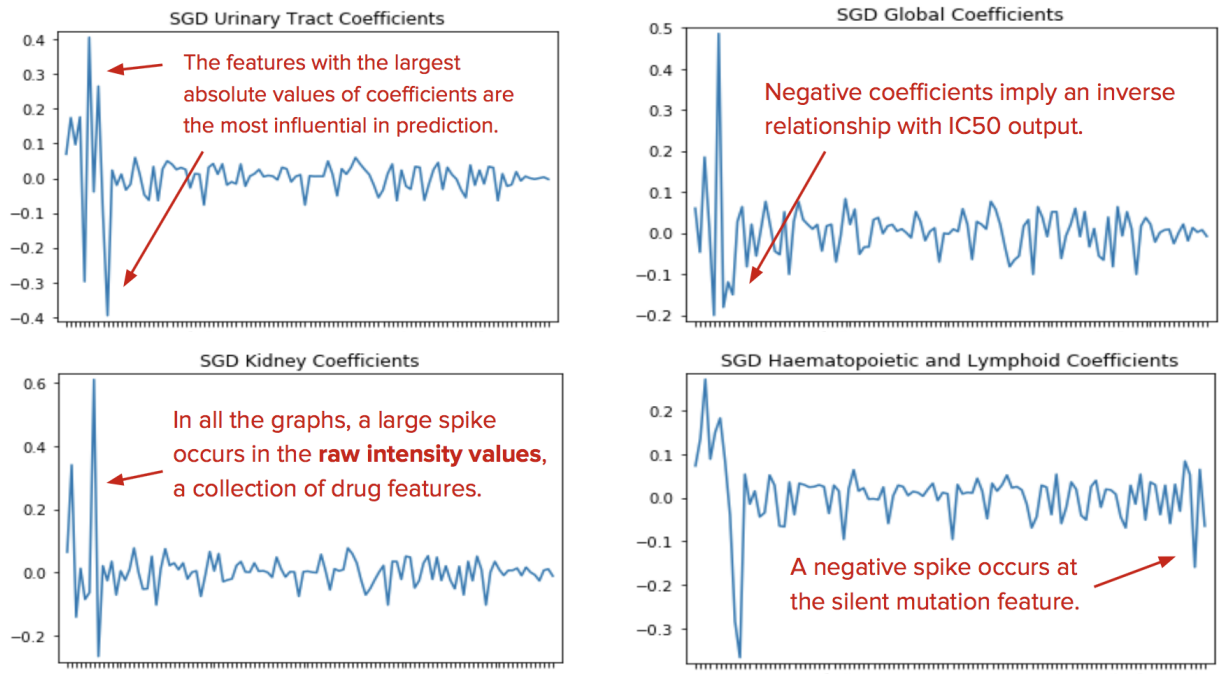

Figure 16.

Discussion and Conclusions

Several key conclusions can be drawn from the data. On average, the neural network accuracy scores across the cancer types are higher than SGD scores (0.93 > 0.85). Neural networks produced a more stable set of accuracy scores; the average range is lower than the average range of SGD (0.23 < 0.32). Furthermore, the standard deviation in neural network scores across 10 trials is significantly lower than the variation of SGD scores (0.07 < 0.10). The most accurate scores in both of the models (0.954 and 0.997) were produced by the urinary tract dataset. However, the H&L dataset produced significantly lower scores (0.701 and 0.785). This could be due to the smaller dataset size and thus the use of fewer samples for training.

Standard and Elastic Net linear models failed to converge and instead produced negative accuracy scores, which shows that they could not learn from the datasets and make accurate predictions due to the size (over 100,000 samples) and sparsity (values of zero were common due to One Hot Encoding) of the data matrices. SGD regression was a better fit than linear or Elastic Net models for the data because it is able to handle more samples (over 100,000) and works well with sparse matrices (relatively few nonzero values in each feature column).

A neural network model with 3 hidden layers, 110 nodes per layer, and 5 epochs per CV was found to produce the highest accuracy scores out of the models tested. Cross-validation was a useful technique in reducing overfitting potential; it provided even testing across the dataset. Furthermore, increasing the number of hidden layers or epochs did not significantly alter the accuracy scores produced for the neural network models; this could be because prediction accuracy levels off at 3 layers and 5 epochs.

Feature coefficient value graphs display the most significant features, as shown above in Figure 15. The features that correspond to the most prominent spikes in the graphs are the raw intensity values of the drugs and, in the haematopoietic and lymphoid graph, genetic mutation description (specifically, silent mutations).

Finally, the results show that a global, multi-cancer model is functional in prediction and generalization can be achieved without sacrificing accuracy. For SGD linear regression, the average score collected for the global dataset over 10-fold cross-validation is 0.817. For the deep neural network, the average score collected for the global dataset over 10-fold cross-validation is 0.938. For both models, the final cross-validation scores at the end of learning curve generation (progressing with incrementing amounts of samples) are close to the training scores at the same point, displaying convergence of the two curves. This would mean that a global model trained on multiple cancer type datasets, which would significantly increase the efficiency of model training and ease of overall use, can be utilized as it is still comparable in accuracy to the cancer-specific models.

Several factors may have contributed to experimental error. For instance, the datasets originally contained not a number (NaN) values, so the the dropna() function was used to filter out samples that contained any missing data. This may have led to decreased model accuracy as the removed samples may have skewed the dataset. Additionally, optimal parameters were not able to be achieved in the neural network due to time and computational constraints. This may have prevented the development of a model with optimal fit during training, thus decreasing accuracy scores during cross-validation tests.

Future improvements can also be made in order to improve prediction accuracy and efficiency. In addition to gene mutations, it is important to utilize a variety of types of genomic characteristics of cell lines, such as whole-protein and intra-protein changes, abnormal gene expression, and alternative transcription processes that lead to atypical gene splicing and protein isoforms. Additionally, incorporating certain known normal gene-drug interactions and specific pharmacogenomic characteristics would be helpful in predicting the effectivity of drugs in treating different variations of cancers and improving precision medicine efforts for patients. Finally, the use of feature selection methods in building and training the models can help increase their predictive power and accuracy by decreasing the amount of features needed and eliminating noise from unhelpful features.

A novel computational method for predicting cancer drug effectivity is a valuable alternative to physically testing many drug candidates through in vitro assays early in the development process, which is very tedious and inefficient. This multi-cancer prediction model can be used to identify drugs that can be effectively used for more purposes (such as new treatment for originally non-targeted cancers) based on the characteristics of the drugs. Furthermore, drug development procedures can be sped up greatly by utilizing data collected through machine learning to recognize characteristics common to compounds that are currently known to be effective for specific cancer types and formulate even more effective drugs that combine these properties. Finally, the ability to input different drug and genomic features and generate an accurate prediction of effectivity can be used in precision medicine, where personalized treatments are developed for each individual patient.

Overall, the use of these machine learning methods can greatly speed up drug development processes. They provide an initial stage that filters a large number of possibly potent drugs in order to eliminate ineffective combinations before starting the tedious process of conducting actual physical trials with cancer cell lines.

References

Aykul, S., & Martinez-Hackert, E. (2016). Determination of half-maximal inhibitory concentration using biosensor-based protein interaction analysis. *Analytical Biochemistry,508*, 97-103. doi:10.1016/j.ab.2016.06.025

Forbes, S. A., Beare, D., Gunasekaran, P., Leung, K., Bindal, N., Boutselakis, H., Ding, M., Bamford, S., Cole, C., Ward, S., Kok, C. Y., Jia, M., De, T., Teague, J. W., Stratton, M. R., McDermott, U., … Campbell, P. J. (2014). COSMIC: exploring the world's knowledge of somatic mutations in human cancer. *Nucleic acids research*, *43*(Database issue), D805-11.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magariños, M. P., Overington, J. P., Papadatos, G., Smit, I., … Leach, A. R. (2016). The ChEMBL database in 2017. *Nucleic acids research*, *45*(D1), D945-D954.

Prasad, V., & Mailankody, S. (2017). Research and Development Spending to Bring a Single Cancer Drug to Market and Revenues After Approval. *JAMA Internal Medicine,177*(11), 1569. doi:10.1001/jamainternmed.2017.3601

Yang, W., Soares, J., Greninger, P., Edelman, E. J., Lightfoot, H., Forbes, S., Bindal, N., Beare, D., Smith, J. A., Thompson, I. R., Ramaswamy, S., Futreal, P. A., Haber, D. A., Stratton, M. R., Benes, C., McDermott, U., … Garnett, M. J. (2012). Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research*, *41*(Database issue), D955-61.

Statement on Outside Assistance

Name: Julia Camacho
Title of Paper: A Computational Method for Predicting Cancer Drug Effectivity Using Machine Learning Techniques
What steps led you to formulate your hypothesis? (Where did you get the idea for your research?) Please be specific. After reading a book about the history of cancer research and modern drug development processes, I learned that *ex vivo* procedures to develop effective drugs are very slow and expensive. I decided to try to develop an *in silico* method in order to better improve the efficiency of this process.
Where did you conduct the major part of your work? (i.e. home, school, or other institutional setting, university lab, medical center, etc.) I conducted all of my work at home.
If you worked in an institutional setting, did you work on your project as part of a team or group? If so, how large was the team and who was on the team (students, adult researchers, etc.)? Describe your role on the team. I did not work in an institutional setting.
Describe what parts of the research you did on your own and what parts you received help (i.e. literature search, hypothesis, experimental design, use of special equipment, gathering data, evaluation of data, statistical analysis, conclusions, and preparation of written report (abstract and/or paper). I conducted a literature search and read all background papers on my own. I formulated my own hypothesis, and received help from a teacher in making sure my experimental design was not flawed. I gathered the data myself, and conducted evaluation of data, statistical analysis of results, and drawing of conclusions on my own. I wrote my abstract and paper myself, and my father helped proofread.
Is your research current or a continuation of previous research? If a continuation, please describe the current work and advancement(s) of this research in comparison to the prior work and results. My research has ended and was not a continuation of previous research.