Berkeley
UNIVERSITY OF CALIFORNIA

Predicting Drug Response in Cancer Patients

Project Deliverable 2: Midterm Report

Rebecca Sarto Basso, Emma Besier, India Bergeland
Nikhil Yerasi, Oskar Radermecker, Claire Dubin

Data-X
Spring 2019

Contents

1. Introduction

1.1 Objective

With advances in genome sequencing technology, patient genomic data is becoming readily accessible to physicians and is more frequently integrated into treatment plans, a practice known as personalized medicine. A significant challenge in this developing field is choosing an effective treatment for a given patient based on their genomic information.

So far in this project, multiple methods of feature selection (PCA, Lasso, RFE, RF) were used to identify important markers in mutation data that affect drug therapy response. Machine learning methods (SVM, logistic regression, multi-layer perceptron, random forest) were then applied to train classifiers for predicting the sensitivity of drugs in cancerous cell lines.

Our end goal is to develop a web application that makes this prediction tool available to physicians. Users can type their patient's mutation data directly into the application, and will be provided with a list of drugs that the patient will likely be sensitive to. As of now, there are no tools on the market that perform such a task.

1.2 Context

The multitude of drugs developed to treat cancer has produced large data sets measuring their effect on various cell types. Isolates of tumor cells have long been used to test drugs in early stages of development. Researchers use the growth and death rates of cell lines as a standardized measure of a drug's efficacy against specific tumor types. Two prominent datasets with this information are the Genomics of Drug Sensitivity in Cancer (GDSC) and Cancer Cell Line Encyclopedia (CCLE).

While basing models on gene expression data generally results in higher accuracy in predicting drug response, this information is less commonly available for patients than mutation data. Mutation data is easier to collect because it involves only genome sequencing commonly performed on cancer patients, while gene expression data involves further analysis of RNA levels within a cell.

Translating knowledge from the laboratory to clinical settings is a major challenge due to the complexity of patient data. Patients may have multiple mutations or other factors that affect drug efficacy, which are not elucidated by experiments on single cell populations. Computational analysis of drug response across cell lines bridges this gap by giving a more holistic view of the drug's interactions in an environment similar to the human body.

2. Literature Review

A literature review was conducted to identify and summarize findings related to gene expression and drug prediction.

Suggested Strategies Azuaje (Azuaje, 2016) describes four essential steps in the development of computational models for predicting drug response. First, given the nature of genomic data having many features relative to sample size, data must be selected and pre-processed using computer-driven selection, normalization, and filtering.

Next, the researcher should choose and train a machine learning model. As Azuaje describes, the most effective techniques when approaching a problem such as this are often ridge regression, elastic net, support vector machines, nearest neighbors, and various combinations of the four.

After completing model training, it is important to implement multiple tests on independent data. Given the nature of the project, it is ideal to test on unseen data sets from multiple laboratories, and to validate accuracy with cross-validation techniques. In the final step, new training-test iterations are used to further refine the model parameters.

Relevant Literature Some studies have achieved promising results based on the above strategies. In a recent NCI-DREAM challenge, the scientific community was given genomic profiling information for 53 cell lines, and the drug response data for 35 of these 53 cell lines. Forty-four sets of predictions were returned, with the highest accuracy being 78%. (Costello, 2014).

As discussed in the article, there are significant challenges in performing a classification task such as this. Ideally, genomic data sets would have characterized drug sensitivities across a large cohort of patients, but this data is extremely time-intensive to generate and is limited in scope of the number of drugs that can be tested. Thus, it is common for genomic data sets to have an extremely high dimensionality, but very small sample size. As evident in the article, this is an issue that leads to less-than optimal test accuracies.

Assessment While none of these studies were based on mutation data, we can gain insight from the above literature. We face similar dimensionality issues in our data set, and will likely have to spend a significant amount of time finding an appropriate means of feature selection. We benefit from the fact that our data set is new, and will be added to on an annual-basis by the GDSC. This means that we can potentially achieve higher test accuracies as we update our model with additional cell lines in years to come.

3. Materials and Methods

3.1 Dataset Description and Data Cleaning

This work involves the use of two different datasets.

3.1.1 GDSC

The Genomics of Drug Sensitivity in Cancer (GDSC) (Benes et al., 2012) provides the drug response of 1065 cell lines and 251 drugs. Figure 3.1a provides a screenshot of a pandas dataframe created while exploring the dataset.

After basic preprocessing, we have a dataframe where each row is composed of a cell line, a drug, and a response (see Figure 3.1b). Cell lines are identified by a unique "Cosmic ID" and drug responses are quantified in two ways:

DRUG_NAME	Number of tests	Number of unique cell populations
Avagacestat	1934	1043
UNC0638	1930	1038
AKT inhibitor VIII	1913	998
JQ1	1881	1040
CHIR-99021	1879	1040

(a) Most investigated drugs of the GDSC dataset.

	COSMIC_ID	DRUG_NAME	LN_IC50	AUC
61224	683665	Avagacestat	3.430306	0.964239
61225	683667	Avagacestat	5.111190	0.985744
61226	684052	Avagacestat	4.197360	0.984087
61227	684055	Avagacestat	4.754255	0.981344
61228	684057	Avagacestat	5.166850	0.982060

(b) Processed GDSC dataset: each row corresponds to a cell line with its unique Cosmic Id, drug, and response.

Figure 3.1: Exploratory data analysis and data cleaning on the GDSC dataset.

1. The **IC-50** measures the concentration of a drug that reduces a response to 50 percent of its maximum. The higher the concentration of drug needed to kill the cell line, the more resistant that cell line is to the drug.
2. The **AUC** or area under the dose-response curve involves fitting the dose-response curve, where the response can be assessed by via the metabolic activity of the cells. The smaller the AUC, the less responsive and the more resistant that cell line is to the drug. Thus, small AUC values indicate a higher concentration of the drug is needed to kill the cell line.

We initially focused on the more standard IC-50 value instead of the AUC. We also focused our analysis on drugs with the largest number of tests and cell lines associated with them. The above results present the analysis on *AKT inhibitor VIII*.

3.1.2 CCLE

The other dataset comes from the Cancer Cell Line Encyclopedia (CCLE). It contains the mutations of 1457 cell lines for 84,434 different genes. After basic preprocessing, we have a dataframe where each row corresponds to a cell line and a 0-1 vector which represents the presence or lack of a mutation in that cell population. Cell lines are identified by a unique "CCLE Name". This is represented in Figure 3.2.

Description	PLCH2_mut	UBE4B_mut	ADGRB2_mut	ZSCAN20_mut	SZT2_mut	MOB3C_mut	ZFYVE9_mut	ST6GALNAC3_mut	TCHH_mut	HRI
127399_SOFT_TISSUE	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
22RV1_PROSTATE	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
A204_SOFT_TISSUE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
A253_SALIVARY_GLAND	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
A427_LUNG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Figure 3.2: Screenshot of the CCLE dataset loaded into Python.

3.1.3 Merging the Data

A conversion matrix provided by GDSC was used to link the "Cosmic ID" with the "CCLE Name" associated with each cell line. All outer values were dropped, leading to a loss of around 60% of all cell lines.

From here, we looked at the distribution of IC-50 values (Figure 3.3) and categorized the drug sensitivity into three classes: "resistant", "medium" and "sensitive".

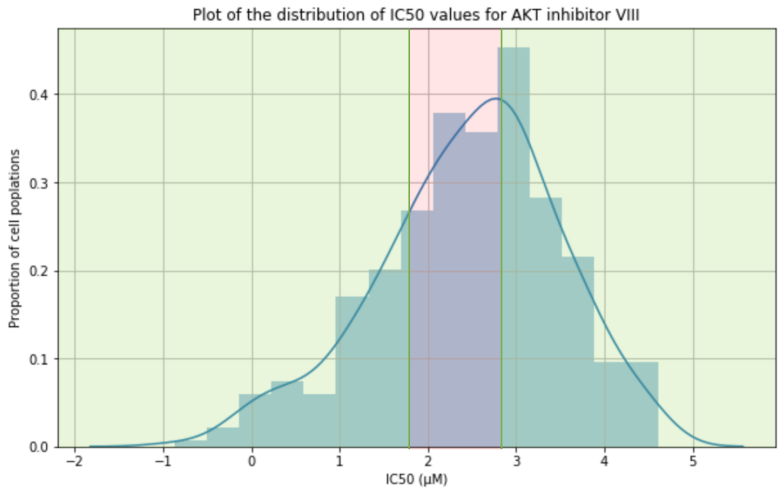

Figure 3.3: Distribution of IC-50 scores and thresholds. The cell lines located in the red zone were discarded; the ones in the green zones were classified as either "resistant" or "sensitive".

Similar to results found in literature (Dong et al., 2015), everything above the mean + 0.5 × the standard deviation (high IC-50) is classified as "resistant" whereas everything below the mean - 0.5 × the standard deviation is classified as "sensitive" (low IC-50). Everything that is in the middle is discarded, and we are left with 218 cell lines. This is the final matrix that was inputted in our feature reduction methods and model training. A screenshot of the final dataframe is displayed in Figure 3.4.

	PLCH2_mut	UBE4B_mut	ADGRB2_mut		DGCR2_del	CASP8AP2_del	SCO2_del	Response
22RV1_PROSTATE	1.0	0.0	0.0		0.0	0.0	0.0	sensitive
A673_BONE	0.0	1.0	0.0		0.0	0.0	0.0	sensitive
ALLSIL_HAEMATOPOIETIC_AND_LYMPHOID_TISSUE	0.0	0.0	0.0	•••	0.0	0.0	1.0	sensitive
CORL23_LUNG	0.0	0.0	0.0		0.0	0.0	0.0	sensitive
DOV13_OVARY	0.0	1.0	0.0		0.0	0.0	0.0	resistant

Figure 3.4: Screenshot of the final, merged matrix.

Note: We did not impute the missing data, as most missing values were due to merging of both datasets and cannot be inferred. Having to drop these values will be further discussed in the limitations section chapter 4.

The final baseline accuracy is 52%.

3.2 Feature Reduction

As described above, our data set has a significant amount of features relative to sample size. The group implemented and investigated several feature reduction and selection methods to decrease the number of features. To avoid data leakage, we computed all supervised feature selection methods on the training set first. The same transformations were then applied on the test set, without looking at the corresponding labels.

First, a rough thresholding of the variance for each feature was performed. All the features with a variance lower than 0.1 were discarded. This allowed us to simplify our training and test sets. Genetic mutations which were present in only a small subset of our cell population were discarded. This method decreased the number of features from 64,000+ to ∼450.

Second, more complex feature selection and feature reduction methods were implemented. Below we provide a brief description of each of these methods.

1. **Principal Component Analysis (PCA):** PCA is an unsupervised feature reduction method which projects the data in a new space given by its principal components. It reduces the dimensionality by keeping the combination of features which lead to the smallest drop in overall variance. The resulting features are combinations of the initial features. The first principal components representing a large enough part of the total variance can be kept, while the others are discarded. Because we end with combinations of our features, this method cannot be used to link individual genetic mutations with a drug response. PCA is the only method which was also applied on the original training set, without the initial variance-based feature selection.
2. **Random Forest-Based Feature Selection:** This method involves training a random forest model on the data and looking at the average depth of each feature in each tree of the forest. Because each individual tree is built on a random subset of all the features (i.e the mutations), discriminant features will tend to be used earlier in the splitting process compared to others. They will then end closer to the root of the tree.
3. **LASSO:** This feature selection method is based on L1 regularization. It constrains the complexity of the model by adding a weighted sum of the absolute value of all the model's parameters to the loss function. This forces the model to shrink, and eventually disregard, the less important features.
4. **Recursive Feature Elimination (RFE):** This feature selection method fits the data with a model in an iterative fashion and removes the least informative features after each iteration.

3.3 Baseline Models

After reducing the number of features using the techniques described above, statistical methods were applied on the data to predict the sensitivity to a given drug.

Model Description

1. **Logistic Regression (LR)** is a statistical model that uses a logistic function to model a binary output ("resistant"/"sensitive" in our case).
2. **Random Forest (RF)** is an ensemble learning method which operates by building a large number of decision trees, a weak classifier with a tendency to overfit.
3. **Support Vector Machine (SVM)** is a statistical model which aims at defining a hyperplane capable of separating the data in a high dimension in the feature space.
4. **Multilayer Perceptron (MLP)** is the only deep learning neural network used in this project so far. MLP are composed of input, output and hidden layers where most of the computation is done.

4. Model Performances & Discussion

4.1 Performance Evaluation

Feature Selection	SVM	Logistic Regression	MLP	Random Forest
PCA	52.73±4.93	54.07±5.73	55.97±3.5	53.12±6.93
RFE	59.55±0.91	56.36±3.34	61.36±3.21	55±3.34
LASSO	55.91±4.68	59.55±4.64	56.82±4.31	56.36±5.82
RF	60.91±7.39	59.55±3.91	**63.18±4.17**	56.36±4.64
No Feature Selection	63.18±5.06	60.45±3.96	61.36±5.18	56.82±3.21

Table 4.1: Average Accuracies from 5-Fold Cross Validation

Amongst all the feature selection and model combinations, the best performing tied using Random Forest for feature selection and MLP as well as no feature selection and SVM. Models implemented with no feature selection consistently performed better than other feature selection methods indicating our feature selection methods could be improved upon. The deep learning neural network model, MLP, performed on average better than other models. Overall, our accuracies are not ideal so we are focusing on improving the model performances.

4.2 Improving Model Performance

After reviewing our model performances, we have decided to take the following next steps:

Improve Drug Selection So far, we have only trained our models on AKT Inhibitor III, a drug with a high number of cell lines. An alternative (and potentially better) way to choose a drug for this task could be to select therapies that have a proven association between patient outcome and genetic mutations. In a recent publication, (Kim et al., 2019) used this same dataset to identify mutations associated with drug response; their findings can inform our decision in selecting a small sample of high potential candidate drugs with a proven association to genetic mutations. Eventually, best models will be tested on all drugs available.

Increase Number of Cell Lines Preprocessing of the data leads to a drop of 80% of the total number of cell lines (from 1065 to 218). This is due to two reasons: (i) a substantial loss during the matching process with the mutation table and (ii) a second major loss when dividing the data into the "resistant" and "sensitive" categories. For (i), the matching process could be improved by using external tools. For instance, the Depmap Portal aims to map cell lines for different data types, including mutations and drug response data, to enable scientists to run joint analyses. Alternatively, we could use an older (and more comprehensive) version of the GDSC dataset which has already been tested in (Kim et al., 2019). To address (ii), we could switch from modeling a classification to a continuous regression, which has proven to yield higher accuracies.

Improve Feature Selection Selecting mutations based on variance is not optimal for our application, as it eliminates potentially relevant rare mutations that may have a strong correlation with the drug response. Feature

selection could be improved in various ways: (i) by using a list of 310 mutations frequently associated to cancer from GDSC's website, (ii) by using a list of the relevant copy number alterations frequently altered in cancer, also from GDSC's website, (iii) by selecting the features with the highest point biserial correlation (used to measure the relationship between a binary and a continuous vector), and (iv) by focusing on those mutations which are commonly tested in the clinic. This last solution follows our goal to make this tool usable by doctors for patients.

Select Better Models We implemented various statistical models, but the literature review identified more promising methods yet to be tried. Our next step is to apply Elastic Net, a machine learning model that was used by many contestants in the NCI-Dream Challenge, (Jang, 2014).

5. User Interface

Aside from continuing to work on feature reduction and increasing our accuracy, we will be building out the user interface. The application requirements will include reading in new patient data, using our trained model to predict drug responses, displaying top drug matches succinctly and clearly, and providing next steps for the user.

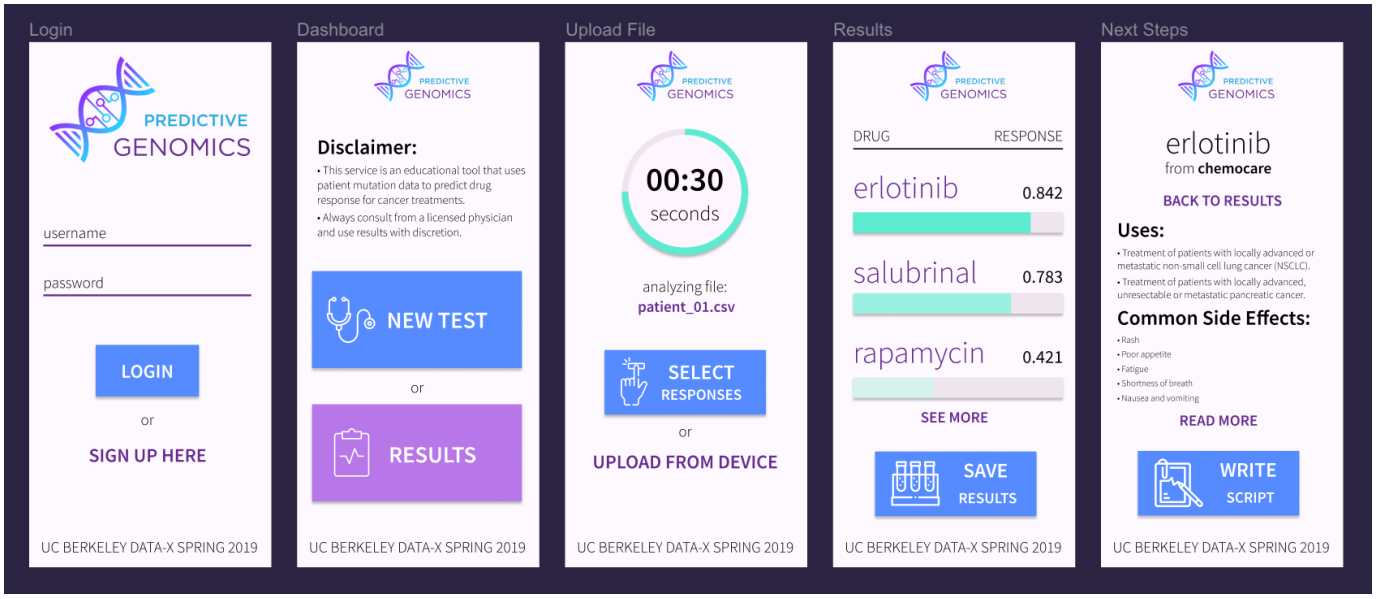

Figure 5.1: Screenshot of Wireframe in Figma

The UI/UX should be simple and clear (see Figure 5.1), providing actionable insights for treatment methods and a consistent design. The application is geared towards healthcare providers and assumes a basic understanding of the background context by the physician.

While this initial version of our wireframe does not represent the final design of our interface, many of the design principles will remain, such as a consistent color scheme to differentiate interactive content such as buttons and text fields from static content like labels and images.

1. Users begin by authenticating or signing up for an account. Our Firebase backend will handle both this and storage of patient data and model results. Since privacy is a paramount in working with health data, storing patient responses may present a challenge.

2. Users can then choose between predicting drug responses for new patient data or viewing results from prior sessions, organized by patient.
3. Once authenticated, users will either upload a .csv file containing their patient's mutation data for analysis. Automatic uploads prevent the risk of human error and allows for input of patient data with high dimensionality.
4. Once uploaded, the .csv is parsed into a DataFrame and processed by our pre-trained model, which selects features to analyze from the new data and returns recommended drugs. In our wireframe, we will display the top three results; each entry includes the drug name as well as a predicted response – a float from 0 to 1. The user has the option to view more results or save the current test. We expect the save screen to allow users to input a name for the session as well as general notes and recommendations for future consideration.
5. We include information about the drugs chosen for the patient as well, much of which we can scrape from reputable and complete sources like Chemocare. Ideally, the app would allow users to locate the nearest and most affordable source of the chosen drug, perhaps even letting licensed physicians write prescriptions.

Deployment

We plan to build the application for the web using React Native to ensure cross-compatibility between Android and iOS platforms. We'll integrate Firebase support to allow for user authentication, and data storage.

This will present a massive undertaking but once we finish the front-end, we will deploy using Heroku, ZEIT, or a similar service.

We will first create a server to listen to web traffic and run functions when a request is addressed to it. To do this, we will use the Flask framework for Python to listen to POST requests, or data carried in JSON objects. The pickle library allows us to load our trained model for drug response predictions onto our web server.

6. Next Steps

To complete our goals within the given time frame of the class, our next steps will include:

1. Switch to a continuous measure of drug sensitivity to preserve data, and begin testing other drugs.
2. Find a more suitable machine learning model (such as Elastic Net), and explore other options for feature reduction.
3. Create user interface to make our model accessible to physicians interested in analyzing their patients' genomic data.

Bibliography

Azuaje, F. (2016). Computational models for predicting drug responses in cancer research. *Briefings in Bioinformatics 18*(5), 820–829.

Benes, C., D. A. Haber, D. Beare, E. J. Edelman, H. Lightfoot, I. R. Thompson, J. A. Smith, J. Soares, M. R. Stratton, N. Bindal, P. A. Futreal, P. Greninger, S. Forbes, S. Ramaswamy, W. Yang, U. McDermott, and M. J. Garnett (2012, 11). Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research 41*(D1), D955–D961.

Costello, James, e. a. (2014). A community effort to assess and improve drug sensitivity prediction algorithms. *Nature*.

Dong, Z., N. Zhang, C. Li, H. Wang, Y. Fang, J. Wang, and X. Zheng (2015, Jun). Anticancer drug sensitivity prediction in cell lines from baseline gene expression through recursive feature selection. *BMC Cancer 15*(1), 489.

Jang, Sock, e. a. (2014). Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data. *Pacific Symposium on Biocomputing*.

Kim, Y.-A., R. Sarto Basso, D. Wojtowicz, D. S. Hochbaum, F. Vandin, and T. M. Przytycka (2019). Identifying drug sensitivity subnetworks with netphlix. *bioRxiv*.