LISH-MOA Challenge

Aysenur Yilmaz, *150114002, Marmara University CSE,* Mustafa A. Hakkoz, *150117509, Marmara University CSE*

***Abstract*— Laboratory for Innovation Science at Harvard (LISH) presents a Kaggle challenge [1] to improve current Mechanism of Action (MoA) prediction algorithms. MoAs are known patterns in large genomic databases and it's essential to match activities of molecules with high accuracy. These molecules will be used for modulating protein targets associated with diseases.**

Experiments are done by analyzing effects of drugs on human cells with new technologies and responses are saved as labels. The goal is a multiclass problem with 207 binary target attributes (possible MoA patterns) and each row can have multiple annotations. Datasets has 23814 rows in total and 876 features.

The main challenge is the very high number of labels and labels and low number of rows. Traditional Machine Learning algorithms such as SVM or Decision Tree-based models will most likely fail due to given specifics of the dataset. Neural Networks are more suitable for high numbered multiclass problems, but they also need huge amounts of data (maybe millions of rows). KNN and Naïve-Bayes may work as well.

Consequently, in this term project, advanced feature engineering and dimensionality reduction techniques such as PCA, Auto Encoders, GaussRank etc. along with interesting NN architectures will be examined (many of them mentioned in public notebooks of the challenge [2]).

***Index Terms*—Mechanism of Actions, MoA, Machine Learning, Neural Network, Kaggle, Literature Search.**

I. Introduction

For centuries, scientists produced drugs from natural herbals or traditional methods. They were put in to clinical use after trial-and-error kind of experiments before understanding the inner biological mechanisms. Today drug discovery process has changed with novel technologies. Scientists identify a protein target related with a disease and develop a molecule to modulate it. Biological activity of these molecules called as mechanism-of-action (MoA). First, a drug is injected to sample of human cells then responses are analyzed with algorithms and search for similarity to known patterns in large genomic databases.

Our dataset consists of these patterns: gene expression and cell viability data. After training on these data, we will try to predict binary MoA labels. For every sample, there could be multiple annotations for MoAs.

In the scope of this project, we aim to make a submission to official Kaggle competition [1] so if we successful, we could help to develop an algorithm to predict a compound's MoA given its signature so we can help scientists on drug discovery.

II. Literature Review

A mechanism of action (MoA) is a series of biochemical interactions that explain the physiological response of the organism to any stimulus at the molecular level. In pharmacology, MoA patterns are used to determine which drugs are useful for which diseases so, there are many industrial applications like TPMS of Anaxomics [3] and Contingent-AI of BioSymetrics [4]. They are mostly white-box models [5,6] constructed by domain experts' reviews, mathematical methods and hand-made decision trees (Fig. 1).

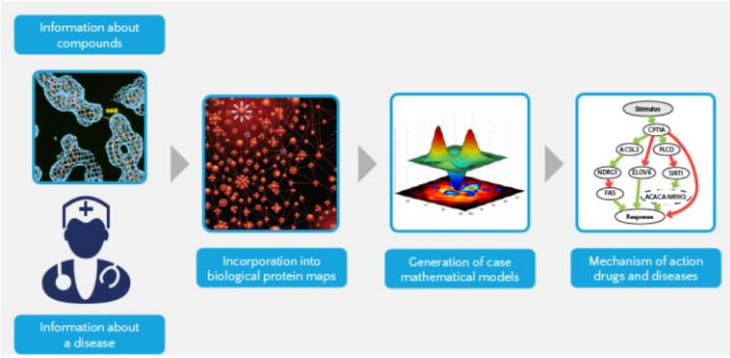


Fig. 1 A White-box model to explain MoAs. Industrial example of TPMS by Anaxomics [3].

Since traditional methods uses manual decision trees for MoA prediction, novel researches also use them as machine learning classifiers. These machine learning enhanced works use them as single classifiers or with ensemble techniques trained on morphological features [6]. Other machine learning approaches used for MoA prediction are SVMs [7], Random Forest [8].

Another novel approach is using Convolutional Neural Network directly on sample images [9]. Dimensionality reductions techniques such as Autoencoders and Principal Component Analysis also used widely in MoA frameworks [10] since high number of features by the nature of responses in cell viability and gene datasets.

The importance of MoA research is still viable and for some cases it's an urgent need. Predicting inhibitor and target protein

Aysenur Yilmaz is with the Marmara University, Goztepe Campus, 34722, Kadikoy, Istanbul, Turkey (e-mail: author@gmail.com).

Mustafa Abdullah Hakkoz is with the Marmara University, Goztepe Campus, 34722, Kadikoy, Istanbul, Turkey (e-mail: mustafa.hakkoz@gmail.com).

to the COVID-19 is crucial to protect human from the disease. Therefore, a protocol to identify anti-COVID-19 candidate based on computer-aided drug design is aimed [11].

III. Data Statistics

A. Dataset Information

There are totally three .csv files. The **train_drug.csv** file consists of 3982 rows and 2 columns. In **train_features.csv** is composed of 23814 rows and 876 columns. In **train_targets_scored.csv** consists of 23814 rows and 207 target columns, it contains binary output attributes for training data whether MoAs triggered by a specific sig_id. Also, this dataset is linked to train_features.csv dataset via sig_id attribute.

B. Feature Statistics

Here, **sig_id** is used for a unique id in the dataset which is also the primary key to link it to the train_targets_scored dataset.

The variables that contain **-c prefix** is used for representing cell viability information. There are almost a hundred different cell viabilities. Also, these variables have a mean nearly zero and a range of -10 and +5. There are some rows that contain negative cell viabilities which is not sensible. On the other hand, there are some variables that have a **prefix -g** which represents gene expression related information. There are almost 775 different gene expressions. These variables usually have a mean zero and take values between -10 and +10. **cp_dose** takes two different values D1&D2 which refers to low and high dosages, respectively. **cp_time** has three distinct integer values, namely 24, 28 and 72 hours. **cp_type** shows that if a sample has been treated with a compound or control perturbation (Table 1).

TABLE I
FEATURE STATISTICS

	sig id	cp type	cp time	cp dose	g-0	c-0
dtype	string	string	int	string	float	float
#unique	23814	2	3	2	14367	14421
count	23814	23814	23814	23814	23814	23814
mean	-	-	48.02	-	0.25	-0.35
std	-	-	19.40	-	1.42	1.75
min	-	-	24	-	-5.51	-10
25%	-	-	24	-	-0.47	-0.55
50%	-	-	48	-	-0.01	-0.01
75%	-	-	72	-	0.53	0.45
max	-	-	72	-	10	3.37

Statistics on first 4 features and 2 representative features with “g-“ and “c-“ prefixes.

IV. Exploratory Data Analysis

From the figures above (Fig. 2 and Fig. 3) both gene expression and cell viability attributes tend to follow a normal distribution which is centered around zero. Also, gene expression attributes have long tails for both sides that show presence of outliers. Moreover, cell viability attributes do not have a right skewed tail but left skewed which tells about negative cell viabilities rates. However, cell viability values cannot be less than zero because it is the percentage of live cells in an environment. One reason that could be caused by this fault is that some sort of transformation has been applied to the dataset and those values do not represent real values for cell viability rates.

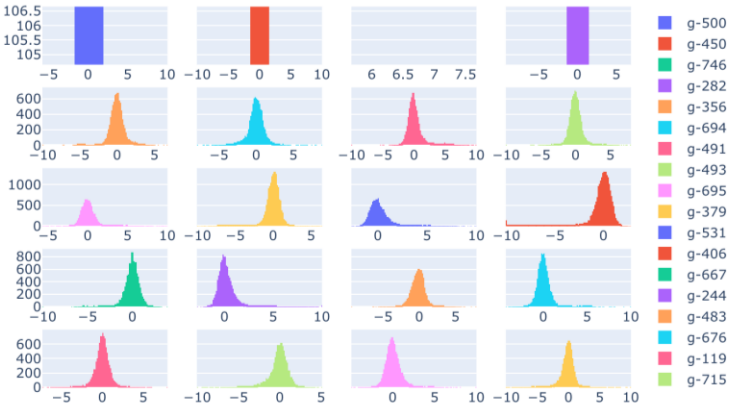


Fig. 2 Distribution of randomly selected gene expression variables.

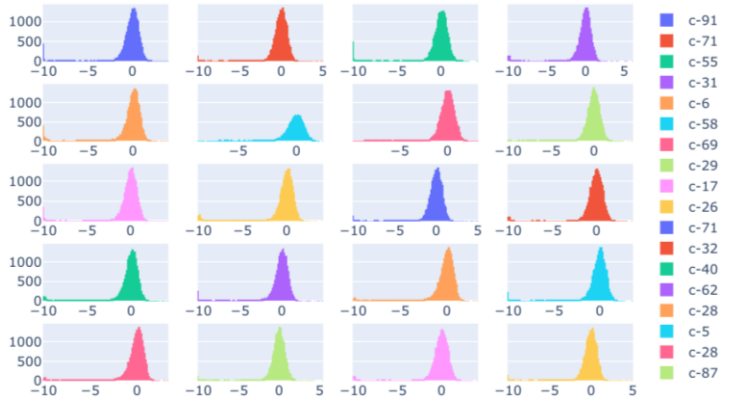


Fig. 3 Distribution for randomly selected cell variables.

For the correlation figure (Appendix 1), highly presence of red values in the plot that shows us cell viability attributes are correlated to each other. That could be helpful in dimensionality reduction. We will take care of this later.

When we examine the class distributions, the thing is that variables that we examine so far drug dosage, drug type and treatment duration have no difference in distribution (Fig. 4, 5 and 6). Hence, it is not expected that these attributes will play an important role while developing models. Also, records show us that ctrl_vehicle and cp_type do not have MoAs.

Moreover, the majority of samples have 0 or 1 MoAs and there are very few samples that have 2 or more MoAs. That is why models could struggle with samples that have multiple MoAs.

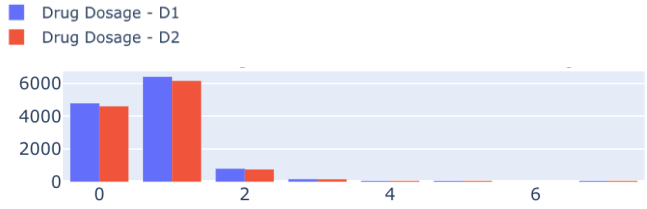


Fig. 4 Sum of drug actions with different dosages.

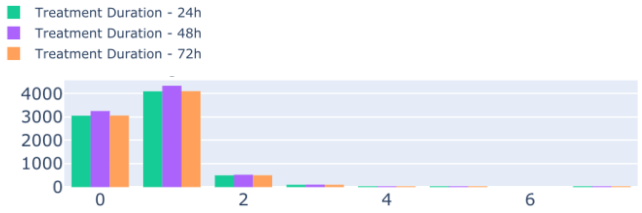


Fig. 5 Sum of drug actions with different treatment durations.

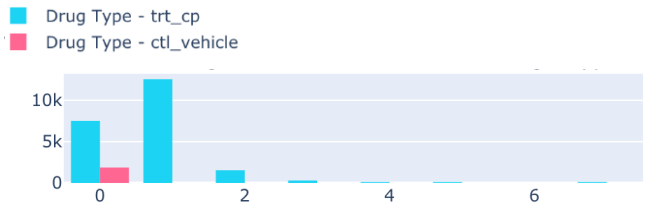


Fig. 6 Sum of drug actions with different dosage types.

This is a multilabel classification problem, there are 207 binary target attributes in the data. We are trying to predict whether a particular gene expression and cell viability sample corresponds to a specific MoAs. Also, one drug sample can have multiple MoAs. Below (Fig. 7) we are looking to find most and least triggered MoAs in the dataset.

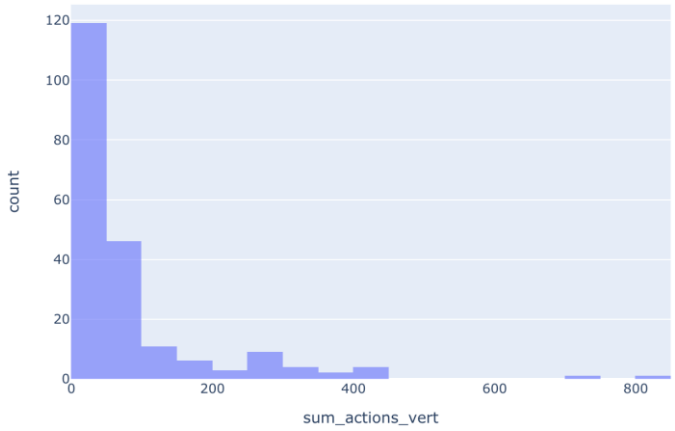


Fig. 7 Histogram of sum of actions across MoAs.

As seen the histogram is left skewed. Majority of MoAs have generally less than a hundred samples that are triggered which is quite low among 23k samples in our dataset.

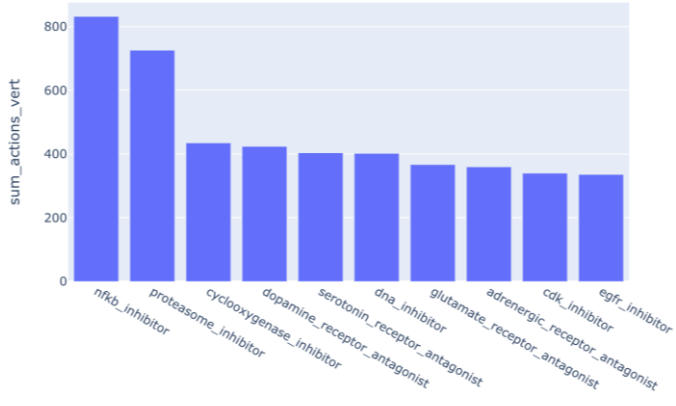


Fig. 8 Most common MoAs.

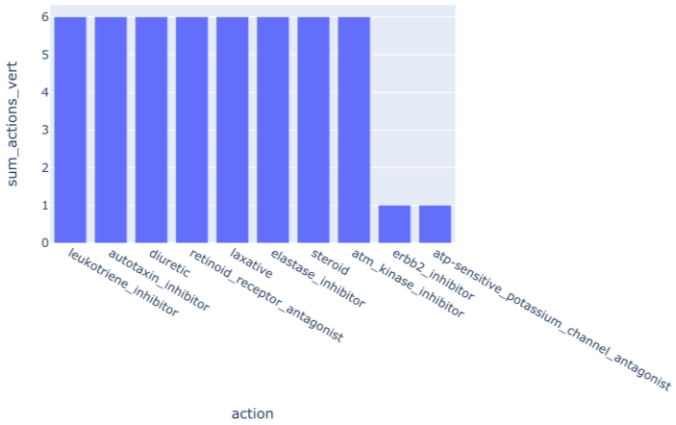


Fig. 9 Least common MoAs.

In Fig. 8 nfkb_inh and proteasome_inh are the most common MoAs. And In Fig 9, least common MoAs such as stereoid, elastese_inh have less than ten records.

Each MoA has been triggered at least once in the dataset. Models may struggle with low triggered MoAs.

Look at the correlation between MoAs (Appendix 2) that we try to define whether a drug is likely to trigger multiple MoAs that are correlated to each other. In Appendix 2, the plot shows that there is no correlation in MoAs, these MoAs are independent from each other. So, triggering of one MoA is not affected by any other MoA.

V. Methodology

Our pipeline starts by preprocessing which consist of feature engineering, dimensionality reduction and encoding. Gene features and cell features have skewed distributions and we convert them to normal distributions by using QuantileTransformer of sklearn [14] library (Fig 10).

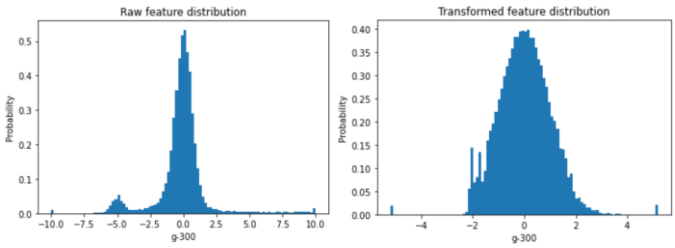


Fig. 10 Original distribution and transformed distribution of an example feature (g-300)

In feature engineering phase, PCA is used for constructing new features. Generally, PCA is used for dimensionality reduction purposes but we added PCA components to original dataset since the models we used (XGBoost and NN) are robust to overfitting. In the beginning, we had 772 gene features and 100 cell features. PCA gives 463 components for gene features and 60 for cell features. This numbers are determined by several experiments. After PCA phase, number of features increased 872 to 1399.

After that VarianceThreshod method [15] is used to eliminate features whose variance are below 0.9 threshold. This threshold also determined by examining correlation matrices (Appendix

1 and 2). As result, number of features are reduced to 1000 from 1399.

Lastly, we encoded two categorical features "cp_dose" and "cp_time" by using CountEncoder of scikit-learn-contrib library [16].

As first model XGBoostClassifier [17] is used with MultiOutputClassifier wrapper of scikit-learn [18]. Since we had 207 target features, this wrapper runs the model selected by 207 times for each feature. Additionally, out-of-prediction technique is implemented to increase our results. It uses cross-validations (5 folds) as training set and make predictions on test set takes average on them. In this way, prediction process behaves like bagging ensembling. As a folding technique MultilabelStratifiedKFold method of iterative-stratification library [19] which creates more suitable folds for multilable dataset.

In second experiment, a neural network with 3-layers is implemented. In each layer, WeightNormalization, BatchNormalization and Dropout is used (Fig 11).

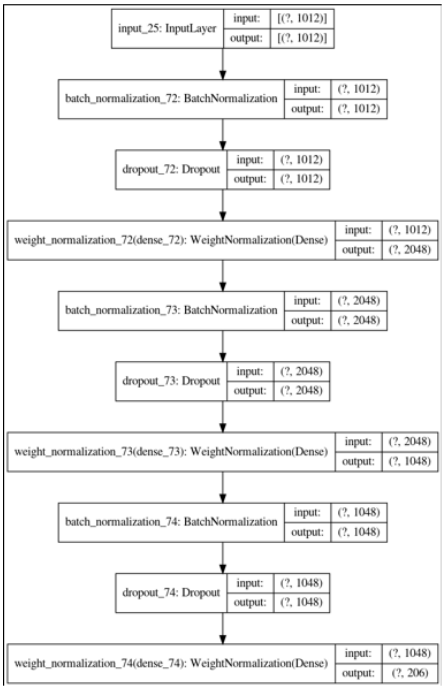


Fig. 11 Architecture of neural network.

The same preprocessing steps in addition with StandardScaler [20] and prediction methods defined above are used. Some parameters we selected are;

- droprout rates: 0.2, 0.5, 0.5
- epoch: 35
- batch size: 128
- ReduceLROnPlateau: 0.1 factor, 3 patience
- ModelCheckpoint: save_best_only
- Lookahead with Adam optimizer: 10 sync_period.
- Loss: binary_crossentropy.

VI. Experimental Results

As evaluation metric, average log loss is implemented.

$$\text{score} = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,m}\log(\hat{y}_{i,m}) + (1-y_{i,m})\log(1-\hat{y}_{i,m})\right]$$

where:

- **N** is the number of sig_id observations in the test data,
- **M** is the number of scored MoA targets,
- $\hat{y}_{i,m}$ is the predicted probability of a positive MoA response for a sig_id,
- $y_{i,m}$ is the ground truth, 1 for a positive response, 0 otherwise,
- **log()** is the natural logarithm.

OOF log loss for XGBoost is 0.024239 and NN is 0.015989. Experiments are run on Kaggle platform for coding the project. It provides 16 GB RAM and 41 hours GPU for a week.

VII. Revised project plan

Week 1 (Nov 16-22): Literature search and investigating approaches in notebooks
Week 2 (Nov 23-29): Literature search and investigating approaches in notebooks
Week 3 (Nov 30-Dec 6): Exploratory Data Analysis and Data Statistics
Week 4 (Dec 7-13): Midterms
Deadline (Dec 14): Midterm Report
Week 5 (Dec 14-20): Implementing Feature Engineering Techniques
Week 6 (Dec 21-27): Implementing Feature Engineering Techniques
Week 8 (Jan 4-10): Implementing NN models
Week 9 (Jan 11-17): Designing and Running Several Experiments
Week 10 (Jan 18-22): Preparing Final Report and Presentation
Deadline (Jan 22): Final Report and Presentation

VIII. Overview and Schedule

In the first week of the project, we did a comprehensive literature search about Mechanism of Actions. In this direction we found out several helpful notebooks on Kaggle platform [2]. Aysenur did exploratory data analysis and related statistics search. Since there are totally 207 binary target features in our test dataset, we decided to use Neural Networks and XGBoost in order to handle this issue.

Both Mustafa and Aysenur prepared a midterm report and evaluated our results on what we did.

In the feature engineering part Mustafa did dimensionality reduction by using PCA method and variance threshold used for feature elimination. For experiments part Aysenur used XGBoost and Mustafa used Neural Networks. Both experiments are run on Kaggle platform with GPU notebooks [21, 22]. After that Aysenur and Mustafa prepared the presentation.

IX. Project Accomplishments

Biggest challenge we faced are, how to handle high number of features (876) with very few samples (23814) by several feature reduction techniques. Our dataset also has too many labels (207) and it causes several problems on training and testing phases. We chose multioutput wrapper with classic models and multi output with NN.

In training phase, we learned out-of-prediction method to improve results and some tricks in Keras while building NN such as WeightNormalization, BatchNormalization, reducing learning rate on plateau (ReduceLROnPlateau), lookahead optimizer [23].

When we finish our project, Lish-MOA competition was already completed so our submissions aren't included on leaderboard but our results weren't very far from top (0.01599). Our scores on private dataset are 0.01657 for NN and 0.02339 for XGBoost.

X. Conclusion and Future Work

This project is built for an Introduction to Machine Learning course. We gain inspiration from several Kaggle notebooks as well. We implemented advanced preprocessing techniques specialized for high dimensional datasets and multilabel classification methods. Also, we learned how to increase our results model optimizations.

As future work, other feature reduction techniques like Auto Encoders and up-sampling techniques like MLSMOTE [24] can be implemented.

References

[1] Kaggle, Mechanisms of Action (MoA) Prediction, overview. [Online], Available: https://www.kaggle.com/c/lish-moa/overview (Date of Access 13 / 11 /2020)

[2] Kaggle, Mechanisms of Action (MoA) Prediction, notebooks. [Online], Available: https://www.kaggle.com/c/lish-moa/notebooks (Date of Access 13 / 11 /2020)

[3] Anaxomics, Mechanism of Action: From clinical evidence to molecular understanding, [Online], Available: http://www.anaxomics.com/mechanism-of-action.php (Date of Access 14 / 12 /2020)

[4] BioSymetrics, Mechanism of Action (MoA) Prediction Platform, [Online], Available: https://www.biosymetrics.com/products/moa-prediction/ (Date of Access 14 / 12 /2020)

[5] Jason H. Yang et al., "A White-Box Machine Learning Approach for Revealing Antibiotic Mechanisms of Action", Cell, Volume 177, Issue 6, 2019.

[6] Franklin J. Bauer et al., "High-accuracy prediction of mechanisms of action using structural alerts", Computational Toxicology, Volume 7, 2018, Pages 36-45, ISSN 2468-1113, https://doi.org/10.1016/j.comtox.2018.06.004

[7] Drakakis, Georgios et al. "Elucidating Compound Mechanism of Action and Predicting Cytotoxicity Using Machine Learning Approaches, Taking Prediction Confidence into Account." Current protocols in chemical biology vol. 11,3 (2019): e73. doi:10.1002/cpch.73

[8] Bakheet TM & Doig AJ Properties and identification of human protein drug targets. Bioinformatics 25, 451–457 (2009).

[9] Chang, Y. et al. "Cancer drug response profile scan (CDRscan): a deep learning model that predicts drug effectiveness from cancer genomic signature", Sci. Rep. 8, 8857 (2018).

[10] Warchal, Scott J., et al. "Evaluation of Machine Learning Classifiers to Predict Compound Mechanism of Action When Transferred across Distinct Cell Lines." SLAS DISCOVERY: Advancing the Science of Drug Discovery, vol. 24, no. 3, Mar. 2019, pp. 224–233, doi:10.1177/2472555218820805.

[11] Wang Q, Feng Y, Huang J, Wang T & Cheng G A novel framework for the identification of drug target proteins: combining stacked auto-encoders with a biased support vector machine. PLOS ONE 12, e0176486 (2017).

[12] Boopathi, Subramanian et al. "Novel 2019 coronavirus structure, mechanism of action, antiviral drug promises and rule out against its treatment." Journal of biomolecular structure & dynamics, 1-10. 30 Apr. 2020, doi:10.1080/07391102.2020.1758788

[13] Kaggle, ML Project: Lish MOA|EDA. [Online], Available: https://www.kaggle.com/hakkoz/ml-project-lish-moa-eda (Date of Access 15 / 12 /2020)

[14] Scikit-Learn, QuantileTransformer. [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html (Date of Access 05 / 02 /2021)

[15] Scikit-Learn, VarianceThreshold. [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html (Date of Access 05 / 02 /2021)

[16] Category Encoders, Count Encoder. [Online], Available: https://contrib.scikit-learn.org/category_encoders/count.html (Date of Access 05 / 02 /2021)

[17] XGBoost, Scikit-Learn API. [Online], Available: https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.sklearn (Date of Access 05 / 02 /2021)

[18] Scikit-Learn, MultiOutputClassifier. [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html (Date of Access 05 / 02 /2021)

[19] Git-hub, iterative-stratification. [Online], Available: https://github.com/trent-b/iterative-stratification (Date of Access 05 / 02 /2021)

[20] Scikit-Learn, StandardScaler. [Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (Date of Access 05 / 02 /2021)

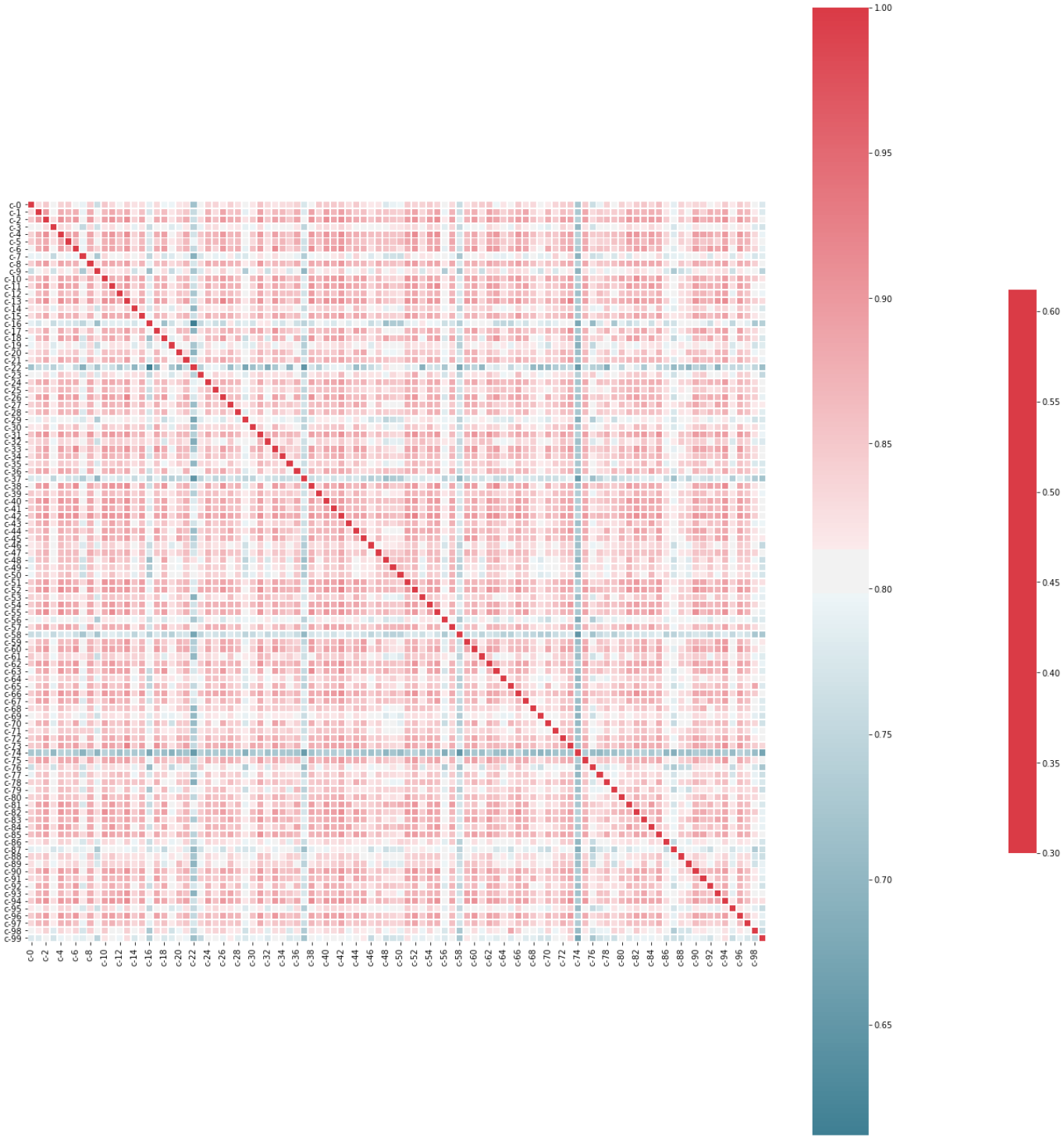
[21] Kaggle, ML Project: Lish MOA|XGBoost. [Online], Available: https://www.kaggle.com/hakkoz/ml-project-lish-moa-xgboost (Date of Access 05 / 02 /2021)

[22] Kaggle, ML Project: Lish MOA|NN. [Online], Available: https://www.kaggle.com/hakkoz/ml-project-lish-moa-nn (Date of Access 05 / 02 /2021)

[23] Zhang, Michael R., et al. "Lookahead optimizer: k steps forward, 1 step back." arXiv preprint arXiv:1907.08610 (2019).

[24] Git-hub, MLSMOTE. [Online], Available: https://github.com/niteshsukhwani/MLSMOTE (Date of Access 05 / 02 /2021)

Appendix 1: Correlation Between Gene Expression and Cell Viability Attributes.



Appendix 2: Correlation Between MoAs