Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Runyuan Zhang Date

Machine Learning-based Prediction of the Three Drugs Combination Synergy

By

Runyuan Zhang

MSPH

Biostatistics and Bioinformatics Department

Zhaohui (Steve) Qin, PhD

Committee Chair

Erik C. Dreaden, PhD

Committee Member

Machine Learning-based Prediction of the Three Drugs Combination Synergy

By

Runyuan Zhang

B.S., Zhejiang University, 2019

Thesis Committee Chair: Zhaohui (Steve) Qin, PhD

An abstract of

A thesis submitted to the Faculty of the

Rollins School of Public Health of Emory University

in partial fulfillment of the requirements for the degree of

Master of Science in Public Health

in

Biostatistics and Bioinformatics Department

2021

Abstract

Machine Learning-based Prediction of the Three Drugs Combination Synergy

By **Runyuan Zhang**

Background: The development of novel drug combination therapy is one of the hot research areas to treat complex diseases. However, current drug combination research still using relatively inefficient experimental screening methods. Though some researchers have already developed in-silico tools for predicting drugs combination synergy, they only focused on two drugs combinations. In this study, we tried to predict three drugs combination synergy using machine learning approaches, and compared models' performance by different algorithms and datasets. It's the first study to explore predicting the synergy response of three-drug combinations.
Methods: In our datasets, we included three drugs combination synergy responses, drug dosages for 560 combination levels, gene expression and mutation data for 13 cell lines. We mainly used tree models for feature selection. We tried modern tree models, Support Vector Machine, and Deep Neural Network for model selection. We also explored whether include extra cell lineage information, gene expression and mutation data in the dataset would improve the model's performance. Furthermore, we developed a novel LightGBM-based vote model and compared its performance with other models.
Results: Adding extra cell lineage or gene expression / mutation data would improve the model's performance. The LightGBM model showed best performance among traditional models, while our novel vote model beat it and even achieved better results according to the cross-validation results.
Keywords: Drug combination synergy, machine learning, LightGBM, Vote model.

Machine Learning-based Prediction of the Three Drugs Combination Synergy

By

Runyuan Zhang

B.S., Zhejiang University, 2019

Thesis Committee Chair: Zhaohui (Steve) Qin, PhD

A thesis submitted to the Faculty of the

Rollins School of Public Health of Emory University

in partial fulfillment of the requirements for the degree of

Master of Science in Public Health

in

Biostatistics and Bioinformatics Department

2021

TABLE OF CONTENTS

1 Introduction

For complex diseases, such as cancer, multi-drug combination therapy is common to be implemented for the treatment [1]. Typically, the drug combination therapy can be synergetic, additive, and antagonistic [2]. Successful combination cancer therapy will lead to synergy results and reduce the drug toxicity, the likelihood of drug resistance, and improve the drug efficacy [3]. However, only a very small part of the drug combination will have a good synergistic effect, and undesirable drug combination may cause catastrophic consequences and endanger the patient's life [4]. In the past, most choices of drugs and their dosage for combination medications is empirically oriented, which makes it very difficult to discover new useful drug combinations in this field [5]. Identification of new drug combination requires the approvement of experiments, which are time and money consuming. Participants enrolled in immature trials will also face greater risks. Therefore, it's urgent to discover more efficient ways to find desirable drug combinations. With the development of the in-silico methods, especially the machine learning approaches, the next-generation sequencing, and the high-throughput drug screening [6], it's much easier for researchers to personalize treatment for patients and find clinically valuable drug combinations [7]. Precision medicine tailored for individuals is gradually becoming possible. Currently, some researchers already built models for drug combination synergy prediction, such as DeepSynergy [8], SynergyFinder [9], and DrugComb [10]. They combined the drug dosage and chemical information with whole genome sequencing, or identified

genetic networks, to predict the synergy results by machine learning approaches [11]. However, there existed limitations in the previous studies. Though in the treatment we may combine more than two type of drugs, previous studies only focused on two drug combination synergy prediction. They limited by the available dataset and most of them used the same dataset from Merck to build the model [12]. In addition, comparing with the single drug effect prediction, the accuracy of drug combination prediction is low. Furthermore, some of their cross-validation methods were too optimistic. They only tested new drug combinations on existing cell lines, rather than new cell lines, resulting in an unrealistic high accuracy rate. The main difficulties in modeling can be summarized in three parts. Firstly, the available data of drug combination with synergy response is very limited, which made it difficult to train a model with high accuracy. For individuals or single laboratories, the cost of using high-throughput screening or other methods to obtain large quantities of drug combination and synergy response data is too high to be acceptable. Secondly, the factors affecting the synergy of drugs have not yet been unified in the biological field, therefore, we do not have a standard to tell us what kind of information we need to collect for modeling. Thirdly, a large amount of biological information cannot be fully quantified. For example, although we can use gene expression or mutation data to represent cell lines, we cannot guarantee that the gene can provides all useful information for cell lines. When we convert these information, we may ignore some important parts.

In this study, we innovatively focused on three drug combination synergy prediction. To our best knowledge, we are the first to consider three drugs combination synergy

prediction. Since there not existed any publicly available three drugs combination datasets, we conducted the experiments and collected data in our own laboratory. We selected two popular anti-cancer drugs, Methotrexate and Vincristine, along with another drug not yet clinically approved, drug X. Total 560 combinations of the three drugs were experimented on 13 cell lines, and the corresponding synergy responses were calculated. We also collected gene expression and mutation data to represent the cell lines. To better determine which data source we should include, we compared model performances by microarray or RNA-Seq data. In order to test whether each part of the data is useful for modeling, including dosage data, gene expression data, mutation data, cell line types indicators, etc., we added them into the model one by one and compared the models' performances. In this part, we obtained our baseline models, advanced models, and final models, which included dosage data only, dosage data and cell type indicators (one-hot encoding for T cells, B cells, early T cell precursors), all data, respectively. We used K-Nearest Neighbors algorithms (KNN) for the baseline model and advanced model. For the final models, we also used several other machine learning algorithms to build models and do the model selection. In previous studies, many researchers found the XGBoost's superiority when they conducted classification tasks [13]. Its gradient boosting framework guaranteed the high accuracy. Recently, many similar algorithms appeared and even showed better performances than XGBoost, such as LightGBM and CatBoost. Therefore, we included all these models to do the model selection. The models included KNN, Support Vector Machine (SVM) with RBF kernel, Random Forest, CatBoost,

LightGBM, XGBoost, and Multi-layer Perceptron (MLP). KNN, SVM, Random Forest, and MLP were implemented through “scikit-learn” package in Python [14]. CatBoost, LightGBM, and XGBoost were conducted via their corresponding Python packages [15-17]. Considering we had too many features after we included gene expression or mutation information in our dataset, I used LightGBM to do the feature selection and only selected the top 60 features to avoid overfitting. Since only less than 5% of the combinations exhibit the synergy, the dataset is extremely imbalanced. We tried different thresholds to split the whole dataset into two, three, or five parts to compare the performances of the multi-class classifier. To avoid too optimistic accuracy, we applied “Leave One Cell Line Out” (LOCLO) cross validation in the training procedure, which means for each training cycle, one cell line’s data will be removed from the whole dataset and use for testing, so that the test cell line will be new cell line for the model. Furthermore, to better address the imbalanced dataset and the scarce data issues, we built both weighted models and unweighted models by each algorithm. The weighted model adopted weighted loss function. Moreover, we innovatively present LightVote, a LightGBM-based vote model, and compared its performances with other models. All models in our study were designed for classification, since we were interested in discovering specific drug combinations that showed synergy results. Furthermore, we tested our models on novel cell lines, which showed three drugs combination synergy prediction was challenging under the data and algorithms currently available.

2 Materials and methods

2.1 Datasets

2.1.1 Three drugs combination synergy experiments

In this study, we selected two commonly used anti-cancer drugs, Methotrexate and Vincristine, and another small molecule drug not yet on the market, drug X. We designed ten dose levels for Methotrexate, eight dose levels for Vincristine, and seven dose levels for drug X. Hence there are 560 drug combination levels in total in the experiments. For cell lines, we selected four T cell lines, seven B cell lines, and two Early T cell Precursors (ETP), as showed in Table 1. Synergy scores based on the experiments' responses were collected which vary from -100% to 100%. The negative and positive scores indicate antagonistic and synergetic results, respectively, whereas zero represent addictive results. We stratify the whole dataset into five groups based on synergy score levels: high antagonism - synergy score lower than or equal to -10%; low antagonism - synergy score between -10% and -5%; addition - synergy score between -5% and 5%; low synergy - synergy score between 5% and 10%; and high synergy - synergy score higher than 10%.

Table 1 – 13 Cell Lines in the Study

Cell Line	Lineage	Gene expression data		Gene mutation data
		Microarray	RNAseq	
CCRF-CEM	T	√		
MOLT-4	T	√		
JURKAT	T	√	√	
DND-41	T	√	√	√
NALM-6	B	√	√	√
KOPN-8	B	√	√	√
697	B	√	√	√
UOCB1	B			
REH	B	√	√	√
RS-411	B	√	√	√
RCH-ACV	B	√	√	√
PEER	ETP		√	√
LOUCY	ETP	√	√	√

2.1.2 Genomic features and Simple indicators for cell line types

We applied one-hot encoding methods to represent three cell lines types, T cell, B cell, and ETP [14]. RNA-seq data of the 13 cell lines are retrieved from CCLE“ (URL: https://portals.broadinstitute.org/ccle/data, file name: “CCLE_RNAseq_rsem_genes_tpm_20180929.txt.gz”) [18], and the Microarray data are retrieved from Sanger (URL: ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-

7.0/sanger1018_brainarray_ensemblgene_rma.txt.gz), mutation data are retrieved from Harmonizome (URL: https://maayanlab.cloud/Harmonizome/dataset/CCLE+Cell+Line+Gene+Mutation+Profiles) [19]. In order to achieve a fair comparison, we only used 17683 genes existed in both Microarray and RNA-Seq datasets.

2.2 Experiments

2.2.1 Feature selection

To avoid overfitting, we first applied the Light Gradient Boosting Machine (LightGBM) to select features. We chose the average gain of the feature when it is used in trees to get the feature importance [16], and selected the top sixty features for modeling.

2.2.2 Machine learning methods comparison

We tested three models in this study. A detailed performance comparison is conducted on these models.

Baseline model (model 1):

Only the dosage data and synergy scores were included in the baseline model since the dosage data for each cell line were the same, while the synergy scores were different.

$$y \sim x_1 + x_2 + x_3$$

where y is the synergy response, x_1 is the dose of Methotrexate, x_2 is the dose of

Vincristine, x_3 is the dose of drug X.

Advanced model (model 2):

For the advanced model, the second baseline model, we added the cell line types indicators into the KNN model.

$$y \sim x_1 + x_2 + x_3 + a$$

where a is the cell line types.

Full model (model 3):

The full models included the whole dataset with the feature selection procedure and were based on KNN and five other state-of-the-art machine learning methods.

$$y \sim x_1 + x_2 + x_3 + a + b$$

where b is the gene expression data by Microarray.

Considering our dataset was extremely imbalanced, we tried weighted and unweighted version for each model for the comparison. In weighted models, the loss function includes weights which are based on the distribution of the number of observations in each class. Furthermore, we also included Multi-layer Perceptron (MLP) for the comparison. All models' parameters were decided by grid search.

K-Nearest Neighbors (KNN): in order to fairly compare models' performances trained on different datasets, we kept KNN in our models. The number of neighbors were searched to achieve the best accuracy. The model was implemented via scikit-learn package in Python.

Support Vector Machine (SVM): considering we had numerous non-linear features in the study, we chose the RBF kernel for SVM modeling since RBF kernel had

advantages in non-linear problems with lots of parameters [20]. Because its essence was still a linear model, we utilized “selectkbest” package in Python for it to finish the feature selection procedure, rather than using LightGBM. In SVM, we used grid search for optimistic regularization parameter C and kernel coefficient gamma. The model was implemented via scikit-learn package in Python.

Random Forest (RF): different number of trees, max depths of trees, and criterions (‘gini’ and ‘entropy’) were taken into consideration when we conducted the grid search. The model was implemented via scikit-learn package in Python.

XGBoost: XGBoost is well known in data science competitions nowadays. It was developed by Tianqi Chen [15]. It contributed valuably novel ideas for traditional gradient boosting methods, such as newton boosting, extra parameters for randomization, etc. We utilized the XGBoost package it provided in Python to build models, and optimized its three parameters, including the number of trees, the max depths of trees, and the learning rate.

CatBoost: CatBoost is a novel open-source library for gradient boosting on decision trees. It was developed by Yandex and good at dealing with categorical features [17]. It executed fast due to symmetric trees and can order boosting to prevent from overfitting [17]. We built the CatBoost model by utilizing its CatBoost package in Python, and tried to optimize three parameters, iterations, depth, and learning rate.

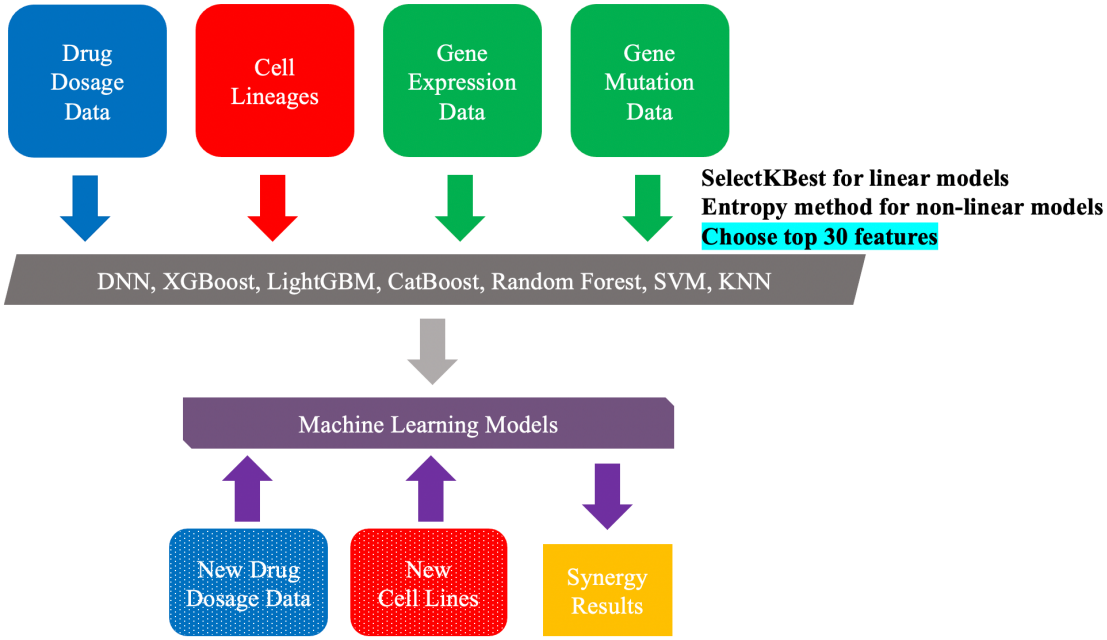
LightGBM: LightGBM is a novel gradient boosting framework based on trees developed by Microsoft [16]. It inherited most of the advantages of XGBoost and was further optimized, such as sparse optimization, early stopping, etc. Unlike other

boosting algorithms, the trees in LightGBM are grown leaf-wisely, rather than based on tree depths, because it believes leaf-wisely growth will decrease the loss more optimally than other methods. Another major difference in LightGBM is its tree learning algorithms. It innovatively chose to use histogram to conduct the tree learning procedure, rather than simply searching the best point for splitting [21]. In this way, its performance becomes more robust, more efficient, and the memory consumption requires much less comparing to XGBoost and CatBoost. In this study, the LightGBM was implemented via LightGBM package in Python [16], and we tried to optimize four parameters of it by grid search, the number of trees, the max depth of trees, its learning rate, and the number of leaves for each trees.

Multi-layer Perceptron (MLP): we also considered neural networks to build models, though our data may not be enough. We used scikit-learn package for modeling and considered different hidden layer sizes and initiate learning rate. For MLP, we only tried the unweighted models.

Figure 1 shows the general work flow for modeling.

Figure 1 – Flow Chart for the general models



2.2.3 Dataset comparison

Considering limited amount of data with extensive gene features, we built five weighted LightGBM models to compare their performances with various level of extra data included in the model. Dosage data and cell types information are considered in all models; whereas gene expression data by Microarray are included in the first model, gene expression data by RNA-Seq are included in the second model, gene mutation data are included in the third model, both gene expression data by Microarray and mutation data are included in the fourth model, and both gene expression data by RNA-Seq and mutation data are included in the final model.

2.2.4 Threshold comparison

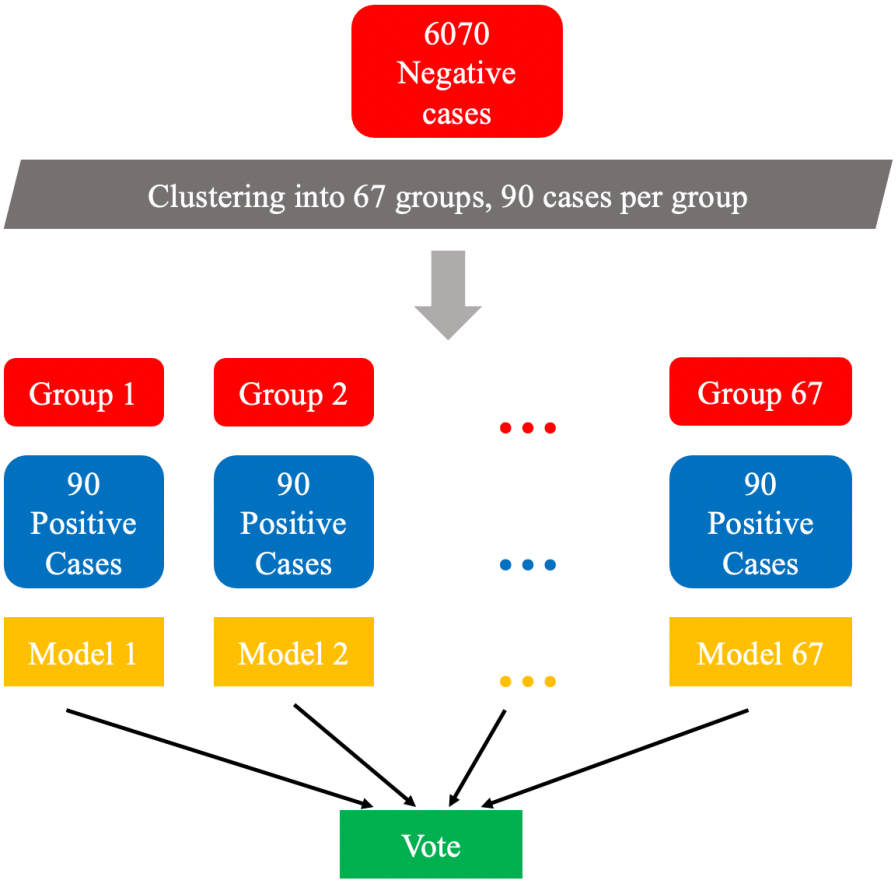
Since our dataset contains multi types and extremely imbalanced, we tried multiple

strategies to split the dataset and made it more balanced. We used weighted LightGBM models with different thresholds and different number of classes and compared models' performances: 5 classes with thresholds -5%, -1%, 1%, 5%; 5 classes with thresholds -10%, -5%, 5%, 10%; 3 classes with thresholds -5%, 5%; 3 classes with thresholds -1%, 1%; 3 classes with thresholds larger than 0 and less than 0.

2.2.4 A vote model for three drugs combination

In order to better prevent the overfitting, we built a vote model for the classification. First, we clustered negative data into appropriate number of parts so that the number of negative data in each part can be roughly equals to the number of positive data. Then we built weighted LightGBM model for each negative part with the whole positive dataset. We let all models tested the target novel cell lines with drug combinations and voted for the final results. Since the dataset for each model was small, we only selected the top 30 features to train the model. We believed this method can help overcome the overfitting and achieve better accuracy. The flow chart is shown in Figure 2.

Figure 2 – Vote Model



2.3 Leave one cell line out cross validation

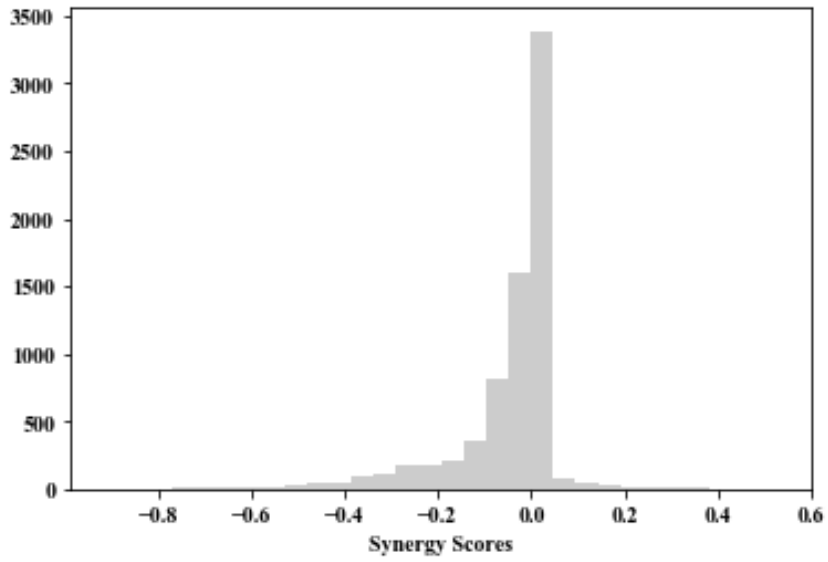
To avoid obtaining too optimistic cross validation results, we removed one cell line in the training procedure and used it for validation in every circle. We called it “Leave one cell line out” (LOCLO) cross validation, which can guarantee the cell line used for validation would never be “seen” by the model in the training procedure. In this study, LOCLO was much more reliable than the traditional cross validation, such as 5-Fold cross validation.

3 Results

3.1 Synergy scores and datasets

Based on the experiments results, total 7280 synergy scores (560 drug combinations with 13 cell lines) were calculated. The distribution of synergy scores was shown in Figure 3. There were 3573 synergy scores between -1% and 1%, and expanded to 5019 when the taken to -5% and 5%. Among them, most synergy scores (2965 synergy scores) equaled to 0, which showed additive results. 28.60% of the results showed antagonistic results, while only 2.46% showed synergy. Therefore, the dataset was extremely imbalanced under the thresholds we set.

Figure 3 Distribution of Synergy Scores



After we collected gene expression data by both Microarray and RNA-Seq, and mutation data, we found that not all 13 cell lines' related information were available. Only 11 of our 13 cell lines owned the gene expression data by Microarray. For gene expression data by RNA-Seq and mutation data, there were only ten and nine cell

lines available respectively, as showed in Table 1. Therefore, for the later parts, not all 13 cell lines would appear in the model's training or testing dataset.

3.2 Algorithms comparison

For the algorithms comparison, we used eight criteria to evaluate models, Accuracy (ACC), Balanced Accuracy (BACC), precision (PREC), sensitivity (SEN), specificity (SPE), F1 Score, kappa, and Matthews correlation coefficient (MCC). Since the tasks were multi-class classifications, for precision, sensitivity, specificity, and F1 score, we calculated both their micro and macro versions. 'Micro' means the related results are calculated globally by considering all cases at once, while 'macro' means that we calculate the results for each class first, then record the unweighted mean of them.
In this part, we adopted stratified 5-Fold cross validation to calculate the results since it's more convenient, and we believed it's enough for roughly comparing the models. Since the models were 5 classes classifier, the basic accuracy should be 20%. The results are shown in Table 2. Apparently, the baseline model's performance was not ideal. The weighted baseline model's accuracy was only 46.53%, and the unweighted model's accuracy was 73.04%. Their balanced accuracies were even worse and closed to 20%. Other evaluation indexes yielded to similar results, as showed in the Table A1 in the appendix. Therefore, it's inadequate to include the drug dosage data only in the model for prediction. To improve the models' prediction accuracies, we added the cell line types indicators into the advance models (Baseline model 2). Comparing with the baseline model 1, both weighted and unweighted KNN models' accuracy, balanced

accuracy, and other evaluation indexes were significantly improved. It was obvious that the extra cell lineages information had benefits for models. The advanced KNN models may distinguish T cell lines, B cell lines, and ETP. Intuitively, if models can recognize every unique cell line, their performances can be improved, again. The results confirmed the conjecture. After the gene expression data by Microarray were added in the training and testing dataset, the weighted and unweighted KNN models achieved 79.12% and 81.31% accuracy, respectively. Since the gene expression data provided too many features in the dataset, which may lead to models' overfitting, we utilized LightGBM to select top 60 features. Only these 60 features were included in the models. We also built six other models for model selection, SVM, XGBoost, LightGBM, CatBoost, Random Forest, and DNN. The results showed that the LightGBM achieved highest scores for both accuracy or balanced accuracy, though the weighted LightGBM didn't show its advantage to the unweighted version. Overall, tree models performed far better than others. Though LightGBM showed best performance, it not significantly exceeded other three tree models, especially compared with the XGBoost. The DNN's accuracy was lack of satisfactory because of the data amount limitation. The SVM's low accuracy may due to the high dimension caused by gene expression features.

Table 2 – Model Selection Results

		ACC	BACC	Dataset
DNN	unweighted	71.80%	23.75%	Drug Dosage + Cell Lineages + Each Cell Line's Gene Expression Data by Microarray
XGBoost	weighted	87.35%	61.60%	
	unweighted	87.40%	58.50%	
LightGBM	weighted	87.50%	62.17%	
	unweighted	88.13%	59.61%	
CatBoost	weighted	85.84%	61.59%	
	unweighted	86.69%	54.10%	
Random Forest	weighted	85.44%	47.56%	
	unweighted	86.27%	52.32%	
SVM (RBF)	weighted	43.25%	33.21%	
	unweighted	72.14%	24.99%	
KNN	weighted	79.12%	46.80%	
	unweighted	81.31%	39.41%	
Baseline 2 (KNN)	weighted	57.81%	33.75%	Drug Dosage + Cell Lineages
	unweighted	75.49%	29.57%	
Baseline 1 (KNN)	weighted	46.53%	29.99%	Drug Dosage Only
	unweighted	73.04%	24.71%	

After we roughly selected our best model, we chose to use and test the original weighted LightGBM model by LOCLO cross validation to obtain more reliable results since the weighted version of LightGBM was more reasonable for modeling for the imbalanced dataset, though there not existed significant difference between the performances of weighted LightGBM and unweighted LightGBM. It achieved high accuracy (99.29%) while its balanced accuracy was low (50%). By checking the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) details, we found the model barely detected any positive cases, the synergy results. Apparently, the model's performance was impacted by the extremely imbalanced data.

Therefore, we tried to solve the problem by innovatively building the LightGBM-based vote model, LightVote. The results were shown in the Table 3. Although the "cluster and vote" method slightly lower the accuracy of the model, the LightVote achieved significantly higher balanced accuracy than the normal weighted LightGBM model. The other evaluation criteria also exhibited similar results, as showed in the Table A2 in the appendix. As long as the evaluation stressed the balance of the data, the LightVote would outshine the normal weighted LightGBM model.

Table 3 – Weighted LightGBM vs LightVote

Model	Cell line for test	ACC	BACC
Weighted LightGBM	Nalm6	0.99	0.50
	KOPN8	0.98	0.54
	CCRFCEM	0.96	0.50
	697	0.96	0.50
	MOLT4	0.98	0.71
	REH	0.99	0.99
	RS411	1.00	0.50
	Jurkat	0.98	0.50
	DND41	0.99	0.50
	RCHACV	0.99	0.50
	LOUCY	1.00	1.00
	Mean	0.98	0.61
LightVote	Nalm6	0.98	0.74
	KOPN8	0.82	0.70
	CCRFCEM	0.87	0.68
	697	0.86	0.54
	MOLT4	0.75	0.87
	REH	0.86	0.86
	RS411	0.89	0.45
	Jurkat	0.93	0.72
	DND41	0.87	0.68
	RCHACV	0.71	0.73
	LOUCY	0.87	0.87
	Mean	0.86	0.71

3.3 Dataset comparison

To further determine whether we should use gene expression data by RNA-Seq or include gene mutation information in our model, we compared different datasets to train and evaluate the LightGBM model. For the feature selection, we still utilized LightGBM to select top 60 features by calculating the average gain of features [16]. In order to fairly compare each dataset, only eight cell lines, including DND-41, Nalm-6, KOPN-8, 697, REH, RS411, RCH-ACV, and Loucy, appeared in all Microarray, RNA-Seq, and mutation datasets were selected in this part. Based on the results showed in Table 4, the model trained by the dataset with RNA-Seq performed slightly better than the model trained by Microarray or Mutation, while extra mutation data added for existing gene expression data barely improved the model's performances.

Table 4 – Dataset Comparison Results

	ACC	BACC
Microarray	87.75%	58.38%
RNAseq	87.61%	62.49%
Mutation	87.79%	58.07%
Microarray + Mutation	87.81%	58.43%
RNAseq + Mutation	87.41%	61.60%

3.4 Thresholds comparison

We set ten groups of thresholds for better dealing with the imbalance issue. The dataset can be more balanced if we split fewer classes or set lower thresholds. 3 classes with thresholds ±0 or ±0.01, and 5 classes with thresholds ±0.01 and ±0.05

were more appropriate comparing to the former 5 classes with thresholds ±0.05 and ±0.1 for both the Microarray dataset of eleven cell lines and RNA-Seq dataset containing ten cell lines. We trained weighted LightGBM models based on these datasets of different thresholds. We employed leave one cell line out cross validation for evaluating models. The results were showed in Table 5. Overall, the results produced by LOCLO were worse than 5-Fold cross validation. The results highlighted by green color showed significantly better results than others, which were all produced by models trained by "more balanced" datasets. The more balanced the dataset we split, the better accuracy the model can achieve. As we split the dataset into three classes by thresholds ±0, the model achieved the best accuracy, 0.63, and best balanced accuracy, 0.63, which both the twice than the random guess accuracy, 0.33. It's also worth to point out that the accuracy and balanced accuracy were more consistent in LOCLO results than the 5-Fold cross validation, which suggested that using LOCLO was more reasonable than 5-Fold cross validation in this study.

Table 5 – Thresholds Comparison Results

Dataset	Thresholds	ACC	BACC
Microarray	3 classes ±0	0.63	0.63
Microarray	3 classes ±0.01	0.60	0.61
Microarray	3 classes ±0.05	0.69	0.50
Microarray	5 classes ±0.01 ±0.05	0.53	0.35
Microarray	5 classes ±0.05 ±0.1	0.66	0.32
RNAseq	3 classes ±0	0.65	0.60
RNAseq	3 classes ±0.01	0.62	0.54
RNAseq	3 classes ±0.05	0.47	0.32
RNAseq	5 classes ±0.01 ±0.05	0.68	0.49
RNAseq	5 classes ±0.05 ±0.1	0.59	0.32

4 Discussion

In this study, we came up with the new challenge of predicting three drug combination synergy. Unlike the two drugs combination, the three drugs combination prediction was more complex and may correlate with more influence factors. We not only completed the model selection by utilizing various machine learning approaches, but comparing five datasets containing different amount of data, ten groups of thresholds used for splitting dataset by the winning model as well.

Based on the model selection results, tree models in this study performed significantly better than others, it may due to the non-linear datasets. Trees models occupied a dominant position in non-linear problems. Within the four tree models, gradient boosting method showed its great superiority. In addition, the LightGBM algorithm

was the winner and beat XGBoost and CatBoost no matter how to evaluate the model. The leaf-wisely growing methods and the histogram-based learning procedure showed their absolute advantages, which eased the common overfitting issue in tree models and made models more robust. Considering our datasets were extremely imbalanced and high dimensioned, we introduced a novel Light-GBM based vote model, LightVote, which showed greater prediction ability comparing to other traditional models. Various evaluation criteria were used to determine whether models showed good performances in the testing part, including both the micro and macro aspects. Overall, we may simply focus on accuracy and balanced accuracy as the other evaluation criteria were consistent.

All models in this study required cell lines' gene expression data by Microarray to represent the genome features. There existed more than one thousand cell lines information in the CCLE dataset. Since the genome features and cell lines were one to one correspondent, the models can be utilized to predict drug combination synergy on every cell line in the dataset with the three specific drugs, drug X, Methotrexate and Vincristine. They were also available for predicting any two drugs combination of the three drugs by setting the third drug dose to zero.

Though the optimized LightVote showed great improvement on balanced accuracy comparing to traditional models, its performance was still not ideal. The main limitation in this study was the small dataset. Since there were no public three drugs combination synergy dataset available currently, we completed the related experiments and created the dataset by ourselves. We only focused on the same three

drugs with limited number of cell lines. Generally, only enough data can guarantee the machine learning models' performances. Therefore, all models faced difficulties to precisely predict new results. It's hard to obtain satisfied results only based on models without data. Besides, we found that adding mutation data can barely improved the model's performance as the dataset already contained gene expression data, no matter the Microarray or RNA-Seq data source. It may due to the high correlation between gene expression data and mutation data. It's also worth to point out that there may exist correlation within the gene expression data, which may be another reason why tree models dominated in this study – tree models would be less impacted by correlation data. In addition, our models were limited by the data amount, so adding more features in the dataset may not exhibit any improvement. In the future, creating a model targeting the correlation with modified loss function may be a breakthrough point.

We convinced that more researchers would attach importance to this field in the near future and contribute more synergy data of different drug combinations. With the increasing of the available data, we believed our models would explosively improving. In addition, by continuing exploration in biomedical research, the factors influencing drug combination synergy can be further discovered and confirmed. With more comprehensive dataset including the confirmed factors, the model can be improved significantly. Hopefully, we can accurately predict the synergy results as we applied combined therapy clinically. We were confident to improve the LightVote based on larger dataset and stronger algorithms, and convinced that the methods we

used in LightVote would be useful in other scenarios. As pioneers in the three drugs combination, though our models existed some limitation, they were useful tools for three drugs combination pre-screening, especially the LightVote.

Acknowledgment

Thank The Dreaden Lab for providing synergy response and dosage data. Thank Dr. Erik C. Dreaden and Mr. James Kelvin for giving suggestions about thresholds setting and explaining the biological significance of features.

Bibliography

[1] Jia J, Zhu F, Ma X, Cao Z, Cao ZW, Li Y, et al. Mechanisms of drug combinations: interaction and network perspectives. Nat Rev Drug Discov. 2009;8:111-28.

[2] Caesar LK, Cech NB. Synergy and antagonism in natural product extracts: when 1 + 1 does not equal 2. Nat Prod Rep. 2019;36:869-88.

[3] Yardley DA. Drug resistance and the role of combination chemotherapy in improving patient outcomes. Int J Breast Cancer. 2013;2013:137414.

[4] Gupta V, Datta P. Next-generation strategy for treating drug resistant bacteria: Antibiotic hybrids. Indian J Med Res. 2019;149:97-106.

[5] Tyers M, Wright GD. Drug combinations: a strategy to extend the life of antibiotics in the 21st century. Nat Rev Microbiol. 2019;17:141-55.

[6] Malyutina A, Majumder MM, Wang W, Pessia A, Heckman CA, Tang J. Drug

combination sensitivity scoring facilitates the discovery of synergistic and efficacious drug combinations in cancer. PLoS Comput Biol. 2019;15:e1006752.

[7] Tallarida RJ. Quantitative methods for assessing drug synergism. Genes Cancer. 2011;2:1003-8.

[8] Preuer K, Lewis RPI, Hochreiter S, Bender A, Bulusu KC, Klambauer G. DeepSynergy: predicting anti-cancer drug synergy with Deep Learning. Bioinformatics. 2018;34:1538-46.

[9] Ianevski A, He L, Aittokallio T, Tang J. SynergyFinder: a web application for analyzing drug combination dose-response matrix data. Bioinformatics. 2017;33:2413-5.

[10] Zagidullin B, Aldahdooh J, Zheng S, Wang W, Wang Y, Saad J, et al. DrugComb: an integrative cancer drug combination data portal. Nucleic Acids Res. 2019;47:W43-W51.

[11] Wang YY, Xu KJ, Song J, Zhao XM. Exploring drug combinations in genetic interaction network. BMC Bioinformatics. 2012;13 Suppl 7:S7.

[12] O'Neil J, Benita Y, Feldman I, Chenard M, Roberts B, Liu Y, et al. An Unbiased Oncology Compound Screen to Identify Novel Combination Strategies. Mol Cancer Ther. 2016;15:1155-62.

[13] Janizek JDC, S.; Lee, S. L. Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine. bioRxiv. 2018.

[14] Pedregosa FV, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:6.

[15] Chen TG, C. XGBoost: A Scalable Tree Boosting System. the 22nd ACM SIGKDD International Conference: ACM; 2016.

[16] Ke GMQF, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NIPS2017.

[17] Prokhorenkova LG, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: unbiased boosting with categorical features. NeurIPS2018.

[18] Ghandi M, Huang FW, Jane-Valbuena J, Kryukov GV, Lo CC, McDonald ER, 3rd, et al. Next-generation characterization of the Cancer Cell Line Encyclopedia. Nature. 2019;569:503-8.

[19] Rouillard AD, Gundersen GW, Fernandez NF, Wang Z, Monteiro CD, McDermott MG, et al. The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins. Database (Oxford). 2016;2016.

[20] Huang MW, Chen CW, Lin WC, Ke SW, Tsai CF. SVM and SVM Ensembles in Breast Cancer Prediction. PLoS One. 2017;12:e0161501.

[21] Zhang J, Mucs D, Norinder U, Svensson F. LightGBM: An Effective and Scalable Algorithm for Prediction of Chemical Toxicity-Application to the Tox21 and Mutagenicity Data Sets. J Chem Inf Model. 2019;59:4150-8.

Appendix

Table A1 - All evaluation results for the model selection

		ACC	BACC	PREC		SEN		SPE		F1 Score		kappa	MCC	Dataset
				micro	macro	micro	macro	micro	macro	micro	macro			
DNN	unweighted	71.80%	23.75%	71.80%	21.88%	71.80%	23.75%	92.95%	82.42%	71.80%	22.25%	13.99%	18.51%	Drug Dosage + Cell Lineages + Each Cell Line's Gene Expression Data by Microarray
XGBoost	weighted	87.35%	61.60%	87.35%	61.50%	87.35%	61.60%	96.84%	95.01%	87.35%	61.52%	72.04%	72.04%	
	unweighted	87.40%	58.50%	87.18%	62.46%	87.18%	58.50%	96.79%	94.60%	87.18%	60.22%	71.04%	71.09%	
LightGBM	weighted	87.50%	62.17%	87.50%	62.45%	87.50%	62.17%	96.88%	95.16%	87.50%	62.30%	72.45%	72.45%	
	unweighted	88.13%	59.61%	88.13%	64.09%	88.13%	59.61%	97.03%	94.84%	88.13%	61.53%	73.00%	73.10%	
CatBoost	weighted	85.84%	61.59%	85.84%	58.95%	85.84%	61.59%	96.46%	94.72%	85.84%	60.16%	69.19%	69.20%	
	unweighted	86.69%	54.10%	86.69%	61.90%	86.69%	54.10%	96.67%	93.75%	86.69%	57.04%	69.02%	69.29%	
Random Forest	weighted	85.44%	47.56%	85.44%	58.54%	85.44%	47.56%	96.36%	92.47%	85.44%	50.72%	64.76%	65.59%	
	unweighted	86.27%	52.32%	86.27%	61.97%	86.27%	52.32%	96.57%	93.31%	86.27%	55.74%	67.48%	67.96%	
SVM (RBF)	weighted	43.25%	33.21%	43.25%	28.44%	43.25%	33.21%	85.81%	83.04%	43.25%	25.96%	11.89%	13.42%	
	unweighted	72.14%	24.99%	72.14%	22.43%	72.14%	24.99%	93.04%	83.14%	72.14%	23.38%	17.72%	21.92%	
KNN	weighted	79.12%	46.80%	79.12%	45.68%	79.12%	46.80%	94.78%	92.14%	79.12%	46.22%	54.77%	54.79%	
	unweighted	81.31%	39.41%	81.31%	51.13%	81.31%	39.41%	95.33%	90.09%	81.31%	41.04%	53.30%	54.66%	
Baseline 2 (KNN)	weighted	57.81%	33.75%	57.81%	30.28%	57.81%	33.75%	89.45%	85.88%	57.81%	30.97%	22.62%	23.53%	Drug Dosage + Cell Lineages
	unweighted	75.49%	29.57%	75.49%	41.03%	75.49%	29.57%	93.87%	85.01%	75.49%	29.43%	30.35%	34.65%	
Baseline 1 (KNN)	weighted	46.53%	29.99%	46.53%	26.37%	46.53%	29.99%	86.63%	84.45%	46.53%	25.42%	13.93%	15.46%	Drug Dosage Only
	unweighted	73.04%	24.71%	73.04%	37.28%	73.04%	24.71%	93.26%	82.87%	73.04%	23.99%	17.93%	23.09%	

Table A2 – All evaluation results for Weighted LightGBM vs LightVote

Model	Cell line for test	TP	TN	FP	FN	ACC	BACC	PREC	RECALL	SPE	F1	kappa	MCC	AUC	PR
Weighted	Nalm6	0	556	0	4	0.992857	0.5	0	0	1	0	0	0	0.985162	0.430208
LightGBM	KOPN8	1	546	2	11	0.976786	0.539842	0.333333	0.083333	0.99635	0.133333	0.125841	0.158075	0.862682	0.253389
	CCRFCEM	0	537	0	23	0.958929	0.5	0	0	1	0	0	0	0.860416	0.148152
	697	0	536	3	21	0.957143	0.497217	0	0	0.994434	0	-0.00946	-0.01449	0.784654	0.143156
	MOLT4	6	545	1	8	0.983929	0.71337	0.857143	0.428571	0.998168	0.571429	0.564165	0.59967	0.976452	0.666422
	REH	0	556	4	0	0.992857	0.992857	0	0	0.992857	0	0	0	NA	NA
	RS411	0	558	0	2	0.996429	0.5	0	0	1	0	0	0	0.787634	0.012277
	Jurkat	0	551	1	8	0.983929	0.499094	0	0	0.998188	0	-0.00318	-0.00509	0.894022	0.164024
	DND41	0	557	1	2	0.994643	0.499104	0	0	0.998208	0	-0.00239	-0.00253	0.808244	0.014607
	RCHACV	0	556	0	4	0.992857	0.5	0	0	1	0	0	0	0.773606	0.033296
	LOUCY	0	559	1	0	0.998214	0.998214	0	0	0.998214	0	0	0	NA	NA
LightVote	Nalm6	2	546	10	2	0.978571	0.741007	0.166667	0.5	0.982014	0.25	0.241877	0.280315	0.997752	0.709524
	KOPN8	7	453	95	5	0.821429	0.704988	0.068627	0.583333	0.826642	0.122807	0.087829	0.153818	0.854015	0.125357
	CCRFCEM	11	478	59	12	0.873214	0.684196	0.157143	0.478261	0.89013	0.236559	0.186246	0.221062	0.736944	0.088591
	697	4	478	61	17	0.860714	0.538652	0.061538	0.190476	0.886827	0.093023	0.038521	0.045851	0.730939	0.092193
	MOLT4	14	408	138	0	0.753571	0.873626	0.092105	1	0.747253	0.168675	0.128788	0.262347	0.985348	0.504265
	REH	0	483	77	0	0.8625	0.8625	0	0	0.8625	0	0	0	NA	NA
	RS411	0	500	58	2	0.892857	0.448029	0	0	0.896057	0	-0.00695	-0.02035	0.62052	0.006775
	Jurkat	4	516	36	4	0.928571	0.717391	0.1	0.5	0.934783	0.166667	0.146341	0.200334	0.832654	0.081277
	DND41	1	485	73	1	0.867857	0.684588	0.013514	0.5	0.869176	0.026316	0.019496	0.065033	0.828853	0.016234
	RCHACV	3	397	159	1	0.714286	0.732014	0.018519	0.75	0.714029	0.036145	0.022517	0.086181	0.620953	0.013506
	LOUCY	0	487	73	0	0.869643	0.869643	0	0	0.869643	0	0	0	NA	NA