Optimizing drug synergy prediction through categorical embeddings in Deep Neural Networks

Manuel Eduardo González Lastre[1], Raúl Guantes Navacerrada[2], Pablo González de Prado Salas[3]

[1]Departamento de Física Teórica de la Materia Condensada,
Condensed Matter Physics Center (IFIMAC), Universidad Autónoma de Madrid, 28049 Madrid, Spain
Email: manuele.gonzalez@estudiante.uam.es

[2]Departamento de Física de la Materia Condensada,
Condensed Matter Physics Center (IFIMAC), Universidad Autónoma de Madrid, 28049 Madrid, Spain
Email: raul.guantes@uam.es

[3]Foqum Analytics, C. de Agustín Durán, 24, 28028 Madrid, Spain

3 June 2022

Abstract

Cancer treatments often lose effectiveness as tumors develop resistance to single-agent therapies. This challenge can be addressed through combinatorial treatments, where multiple drugs are administered simultaneously. However, the combinatorial space of N different drugs at d doses results in N^d possible treatments, making it impractical to test each combination experimentally. To efficiently identify the most promising drug combinations, we propose a data-driven approach utilizing deep learning with categorical embeddings. Our method leverages neural networks to extract meaningful patterns from categorical variables, optimizing the prediction of drug synergy and enhancing treatment efficacy.

Keywords: drug synergy, drug discovery, chemoinformatics, synergy prediction, deep learning, categorical embeddings

1 Introduction

Personalized cancer treatments have proved to be a form of therapy with remarkable results over the last years [1]. However, single agent treatments lose effectiveness with time as tumors develop resistance to the administered drug [2]. It's for this reason that drug combinations have risen as an alternative therapy with the potential to provide a long-lasting benefit to cancer patients [3]. Other benefits of drug combinations are the synergistic effects that may arise when different drugs are applied to a certain type of tumor [4]. This increase in effectiveness could lower the needed dose of the agents, reducing side effects.

To evaluate the synergy of drug combinations, high-throughput preclinical screenings play an important role. They allow us to perform experiments in an undirected, parallelized manner, therefore they are an invaluable tool for massive screenings. Nevertheless, they have limitations, the combinatorial space generated by N drugs and d doses offers N^d possible treatments. The task of covering every single possibility on a screen is unrealistic, so computational approaches are needed in order to select potential candidates over all the other possibilities.

In this regard, Machine Learning has already proved to be a useful tool in drug discovery [5, 6, 7], as it can learn directly from the data without the need of an underlying model (data-driven discovery). These algorithms learn by extracting useful "representations" from the data [8], that will later be used to perform inference.

In this Master Thesis, we will study the prediction of drug synergy using classical Machine Learning algorithms and deep learning (neural networks). We

use the dataset provided in the AstraZeneca-Sanger Drug Combination Prediction DREAM Challenge to train and evaluate our models and benchmark it with the results from the competition. One of the novelties of our work is the use of deep learning for this challenge. Our neural network, Tabular Learner [9] is capable of discovering similarities between entities from the same categorical variable. This is thanks to its categorical embedding layers [10], that act like a funnel that forces the network to build accurate representations of our data in a space of lower dimension (see **Appendix 2: Categorical embeddings**). We train three well established models (Random Forest, XGBoost and SVR) and an experimental model (Tabular Learner) and evaluate the accuracy of the predictions using a custom metric appropriate for the nature of the problem [11]. After that, we create an ensemble model by averaging the predictions of all models. The accuracy of this ensemble is comparable with the one we would obtain by replicating the experiments [11] and it would rank us in the top 20 of the 160 teams that participated in the competition.

Lastly, we show that adding structural information about drug molecules improves the accuracy of all models. We compare two possible methods of feature engineering: plain MACCS vectors and the binary OR operation of them and determine the best method for each model.

2 Materials and Methods

2.1 The dataset

We used the dataset provided on the "AstraZeneca-Sanger Drug Combination Prediction DREAM Challenge" (www.dreamchallenges.org) [11] to train all of our models. The dataset consisted of 11,576 experiments from 910 drug pairs across 85 cancer cell lines. For each sample, the dataset provided drug compound names, cell line name, monotherapy features characterizing the Hill's curves of dose-response and a synergy score of the drug combination in the current cell line. The synergy score was calculated by performing a drug sensitivity screen in a 6x6 dose matrix and using the Loewe model [12].

The dataset also included additional information like the tissue of origin of the cell lines, pathway and function of drugs and, for approximately half, the SMILES code (Simplified Molecular Input Line Entry Specification) [13] which encodes the structure of the molecule and can be used to obtain molecular fingerprints of drugs.

2.2 Weighted Pearson Correlation metric

These high-throughput screenings have a large experimental variability, so typical regression metrics as mean-squared error are not suitable for evaluating the accuracy of our synergy predictions.

Instead, in the challenge [11] they focused on the correlation between predicted and experimental synergy of the drug combinations over the cell lines they were experimented on. For a single drug pair the Pearson correlation, ρ_i is obtained as followed:

$$\rho_i = \frac{\sum_{j=1}^{n_i} (x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{n} (x_j - \bar{x})^2}\sqrt{\sum_{j=1}^{n} (y_j - \bar{y})^2}}, \quad (1)$$

where x_j and y_j are the predicted and experimental values of the synergy and n_i is the number of cell lines corresponding to each combination.

The metric was then defined to be a Weighted Pearson Correlation average or WPC, where the weight of each drug combination was set to be $\sqrt{n_i - 1}$. The mean was done for the N tested drug combinations:

$$\rho_w = \frac{\sum_{i=1}^{N} \sqrt{n_i - 1}\rho_i}{\sum_{i=1}^{N} \sqrt{n_i - 1}}. \quad (2)$$

With this metric, we can properly evaluate our model's performance and have a lower and upper bound. The lower bound being 0, corresponding to the random model and the upper being the correlation obtained by replicating the experiment, which is on average 0.4 [11].

2.3 Machine Learning models

We have trained a wide variety of Machine Learning models, from classical ML to deep neural networks. Here I provide a brief description of them.

2.3.1 Decision Tree Regressor

Decision Tree is a Machine Learning algorithm originally created to perform classification tasks. The tree is constructed as follows: given a dataset, the model computes the optimal conditions to split the data into groups. Each split is designed to maximize "information gain". To turn this model into a regression model, we modify the ID3 algorithm [14] (the core

algorithm for building decision trees) to optimize not the information gain but the standard deviation.

2.3.2 Random Forest

Random Forest consists is an ensemble method where we average the prediction of many Decision Trees [15]. Each Decision Tree is trained in an overlapping sub-sample of the dataset (Bootstrapping), making it possible to train hundreds of different Decision Trees with a low amount of data.

The main advantage of Random Forest is that it is very resistant to noise. Each Decision Tree has a high variance with respect to its training data, but when we average all the predictions (Bootstrap Aggregation or Bagging) the errors of each tree are also averaged, reducing the variance of the model.

2.3.3 XGBoost

XGBoost [16] is very similar to Random Forest, but instead of trying to predict the target value, each tree tries to predict the error (residue) of the previous one. This is known as Gradient Boosting, and it has proven one of the most powerful tools in Machine Learning competitions on structured data [17], which is our case here.

2.3.4 Support Vector Regressor

SVR [18] is a model primarily used when dealing with high dimensional data. It represents the samples as points in a high dimensional space and tries to find the hyperplanes that best delimit the data into different groups. They are fast to train and explainable.

2.3.5 Tabular learner

All the other methods belong to what is called classical Machine Learning. In contrast, Tabular Learner [9] is a neural network, which are capable of recognizing more complex patterns in the data but require more training samples and are prone to overfitting.
This model uses categorical embeddings [10], a powerful concept brought from the Natural Language Processing field. An embedding is a layer which tries to extract meaningful information about the similarity among different items of the same category.
In our context, we want the network to check for similarities among drugs and cell lines and use them to our advantage (Fig. 1). Ideally, the network would translate the knowledge gained from one drug to another similar one. This could turn very useful as the amount of data for each drug is very low.

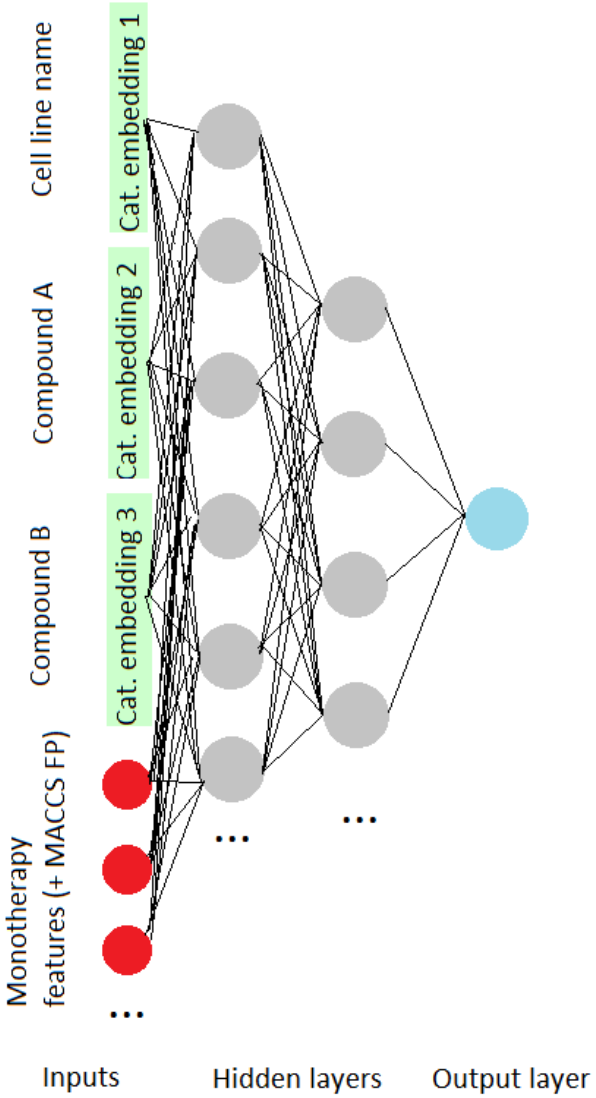

Figure 1: Tabular Learner architecture. In the first layer, the categorical variables go through the categorical embedding layer, meanwhile the numerical values are used without further transformations. Then these features are passed through the hidden layers, where a set of matrix multiplications and non-linear functions are applied. The final layer consists of one neuron that outputs the predicted synergy score of the network.

2.3.6 Ensemble models

An ensemble model consists of a combination of already existing models. This combination usually boosts performance, as different models tend to have different biases (assumptions made by the algorithm).

For example, Neural Networks are better at synthesizing different features than Random Forest or Gradient Boosting [19] so it's interesting to make an ensemble of them since they may capture different insights from the data.

2.4 Molecular Fingerprints

There have been previous studies where researchers used the chemical structure of the molecules to predict the sensitivity of a drug combination [20, 21]. Since our models can't work with plain text, we convert the SMILES into a binary vector of 166 MACCS fingerprints using the R package "rcdk" [22] which is something that our models can digest. Each element of these vectors is associated with a specific molecular pattern of interest in pharmacology.

3 Results

3.1 Preliminary study of the dataset

The first step of every Machine Learning project is the same, a preliminary exploratory data analysis or EDA. It let us gain insight about the features we will use to feed our models and if they need to be transformed in any way before. In this project, there are two kinds of variables: numerical and categorical. The numerical are the monotherapy features: IC_{50}, E_∞, H which are fitted from the dose-response curve using as reference the Hill equation (**Appendix 1: Hill dose-response curve**) and the maximum concentration used for each drug.

The categorical variables are the names of the cell lines and the drugs. The SMILES code could fall in this category but, since they are more problematic, we will leave them for later.
Lastly, the target variable, which is the synergy score calculated with the Loewe model. This means that we are dealing with a regression problem. When we plot the distribution of synergy score while subtracting outliers (samples with $||SC|| > 100$) the result is a gaussian like curve with mean 11.74 and standard deviation 24.71 (Fig. 2). The excessive kurtosis of the distribution is 1.00, so it is sharper than a normal distribution. Our data is tightly grouped around its mean, so even a model that just outputs the mean would have a low mean-squared error, even though it is not learning anything from training.

Also, we can't ignore the biological picture. These experiments have known problems of reproducibility, different instances of the same experiments can lead to different synergy scores. For these reasons, the DREAM Challenge 2015 proposed an alternative metric, the Weighted Pearson Correlation coefficient or WPC (see **Methods: Weighted Pearson Correlation metric**) that focuses more on correlation along different cell lines than in predicting a synergy score close to the target value appearing in the dataset.

Now we have a correct framing of our problem, so we continue exploring the features provided by the dataset. In Figure 12 we can make an idea about the numerical features of monotherapy. We observe that for the majority of drugs, the maximum concentration is below $20\mu M$. In the E_∞ histogram there is a valley from values from 0 to 90 and then a jump when it reaches the bin $E_\infty = 100$. This value corresponds to the drug being ineffective at any concentration at all (Eq. 3). The fact that there is a gap and then so many data points for this value could mean that **at the doses of the experiment** there was no observable effect of the drug, more than there is no effect at all.

Upon plotting the synergy score as a function of each monotherapy feature (Fig. 13), we realize that there is no obvious relationship between the synergy and the monotherapy features (at least in isolation). This means that, if a function of the synergy scores and such features exists, it will be non-linear and with mixed terms. This is one of the reasons why there are no successful analytical models for synergy.

The correlation map (Fig. 3) between monotherapy features reveals that most of the features are independent of the rest, so everything adds information and thus should be included in our models. There is a relatively high correlation between IC_{50} and the max concentration, but that is due to experimental design, since the researchers decided the maximum concentration that should be tested for a drug based on its IC_{50}.

Lastly, we explore the categorical variables. For that matter, we use notched box plots for cell line names (Fig. 14) and drug names (Fig. 15 and 16). There are cell lines that are more prone to synergy than other, for example the NCI-H1793 cell line corresponding to lung NSCLC adenocarcinoma has syn-

ergy above 25 with statistical relevance (Fig. 14). The same happens for drug names, for example MAP2K_3 is a drug that wherever is involved leads to synergy with statistical significance (Fig. 15).

These differences on the synergy distribution across drugs and cell lines justify adding them as features in our models. Since most ML models aren't able to process plain text, we will uniquely associate each item in a category with an integer. Now, our models can process categorical data. This is what we use for Random Forest, XGBoost and Support Vector Regressor, but Tabular Learner uses a more sophisticated concept to encode for categorical variables, called categorical embeddings (see **Methods: Tabular Learner**) which aim to look for relationships between items of a feature.

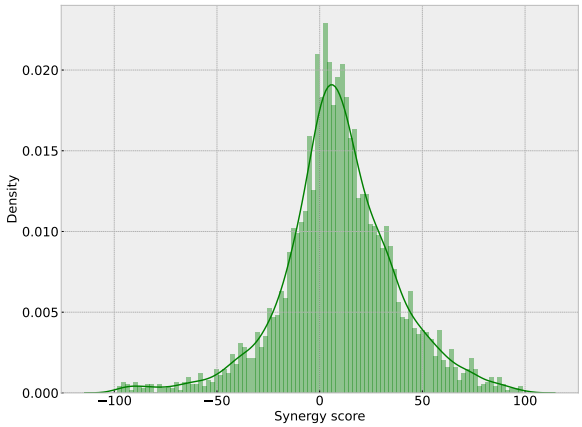

Figure 2: Synergy score distribution without outliers.

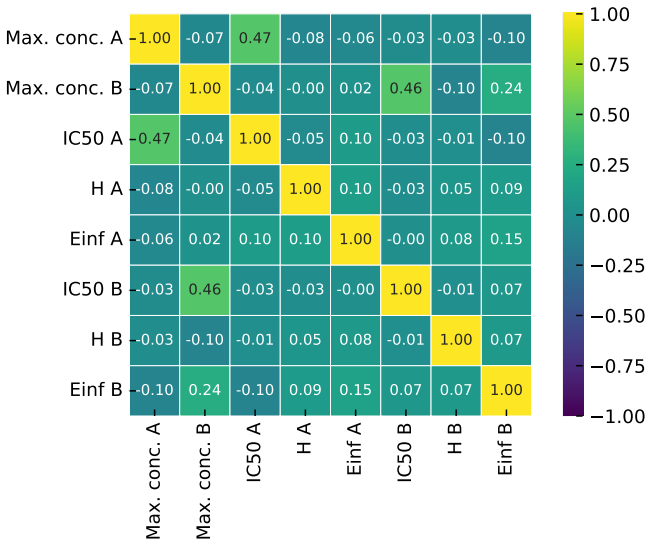

Figure 3: Pairwise Pearson correlation matrix of monotherapy features.

3.2 Prediction without molecular structure

In this section, we explore the prediction of our models without using information about the chemical structure of the drugs. We use cell line names, compound names and monotherapy features.

The first step is to clean the dataset. We select the dataset corresponding to Challenge 1 in the Dream Challenge, which is the one that contains combinations training data. This dataset contains 3879 samples that end up being 3439 when we filter out the experiments with bad quality (QA $\neq 1$).

Our first step towards a competitive result was to realize that the dataset was not commutative, i.e. for a sample with drug pair (A, B) in the dataset, there is not another one with the pair (B, A). This doesn't make any biological sense as we are putting drugs in a 6x6 dose matrix and transposing it doesn't change the synergy score. This detail, although obvious, went unnoticed by the top teams of the competition. Another benefit is the data augmentation, increasing the dataset size to 6878 samples.
For training, we use the original and permuted samples as if they were different experiments and for inference we average the prediction of both groups in order to have one predicted synergy score for drug pair and cell line.

The training of Random Forest, XGBoost and SVR is rather simple as they are very robust against hyperparameter choices. However, that's not the case for neural networks. The most important hyperparameters for neural networks are the learning rate, the number of layers and the neurons per layer.

Our choice of learning rate will depend on how deep (number of layers) and complex (number of parameters) we build our network. With biological data, specially with such a low number of features, we prefer shallow neural networks with as little parameters as possible. This is to avoid variance [23, 24], which is the modeling of the noise by the network (overfitting). We build a network with 2 hidden layers of 20 and 10 neurons, respectively. The learning rate was obtained applying the fast ai learning rate finder [25] which suggests an optimal learning rate of $LR = 0.001$ (Fig. 5).

Since the WPC score can't be used as loss function (we cannot compute the gradients from it), we

choose the Mean Squared Error loss as a proxy and Adam as optimizer. In Fig. 4 we see a training run until overfitting.

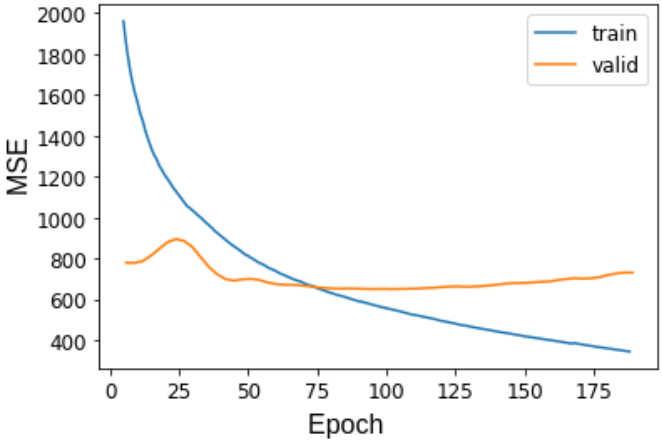

Figure 4: Mean Squared Error in each epoch in the training and test set. We appreciate some overfitting at the end of the training, but we have to remind ourselves that MSE is just a proxy of our real metric, the WPC score.

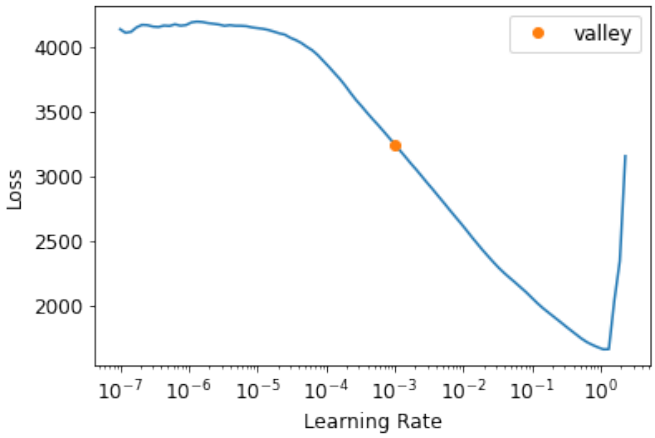

Figure 5: Fast ai learning rate finder on tabular learner.

We train our models with different methods and compute the results for each model in isolation and for the ensemble model, which is an average of the predictions of all models (Table 1). We realize that with commutativity, the performance of XGBoost and SVR greatly increases. It stays the same for Random Forest and worsens for Tabular Learner. However, we also observe that the training of Tabular Learner is really unstable, its performance greatly depends on random seed. We can (partially) alleviate this by using controlling training with an Earlystopping Callback (see: **Appendix 3: Custom Callbacks for Tabular Learner**). We will track the Pearson correlation of our predictions, check whenever it stops improving and stop training. This doesn't equal to using the Pearson coefficient to perform gradient descent, as it is impossible since it's not a loss function, but watching over for this metric combined with the fact that MSE is already a good proxy for the WPC score leads us to a more stable training and better results, as we see in the Table 1. In the official competition, they evaluated their models on a LeaderBoard set. After adding this subset to the evaluation, our performance increases, reaching $WPC = 0.32$ which would put is in the top 20 of the competition. This result is remarkably close to the score obtained by experimental replicates, which ranges from 0.4 to 0.5.

The WPC metric gives a statistic on how well the ensemble model predicted correlations over all cell lines. It is weighted to account for how many cell lines was the drug combination tried on. However, it's interesting to dissect the WPC to get a more accurate picture of the models' performance. For that, we rank the Pearson Correlation of each drug combination (Fig. 6) and observe that there are in fact 25 drugs where the model predicts the correlation backwards (negative values of Pearson), the predicted correlation is the inverse of the experimental. There are several possible explanations:

- Problems of replicability on these drug combinations.
- Too few cell lines for these drugs, thus giving raise to sampling problems.
- A deeper biological mechanism that we are not aware of.

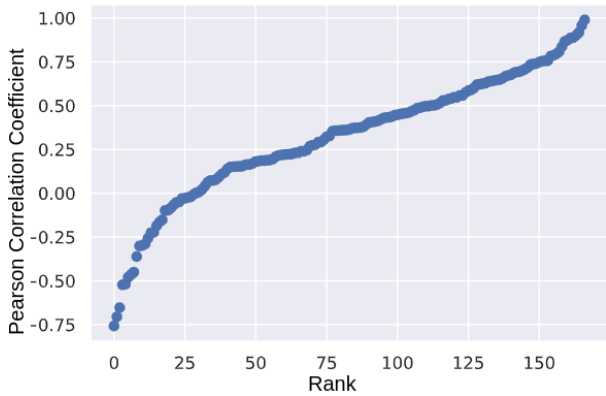

Figure 6: Person correlation of drug combinations ranked in ascending order.

To dive further on the sampling hypothesis, we plot the Pearson correlation distribution as a function of the number of cell lines (Fig. 7) and check

that the highest and lowest values of correlation correspond to the pairs experimented on the lowest number of cells, so sampling is indeed a big part of the problem. Lastly, we realize that, even though there are extreme values when the number of experimented cell lines is low, the median of the Pearson score is close to the WPC in all cases, so the number of cell lines modulates the standard deviation of the Pearson score distribution.

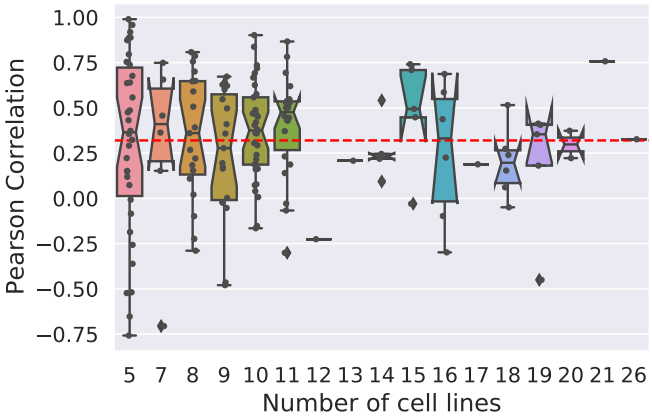

Figure 7: Notched box plots of synergy score distribution. The red dashed line represents the WPC = 0.32.

3.2.1 Training instability of Tabular Learner

Tabular Learner is by far the most complex model we used for inference. Its potential for uncovering complex representation comes at the price of a high variance. Even with our attempts to stabilize training through building a shallower network, fewer parameters, custom callbacks, etc. we have found that the end result is highly dependent on random initialization (Fig. 8).

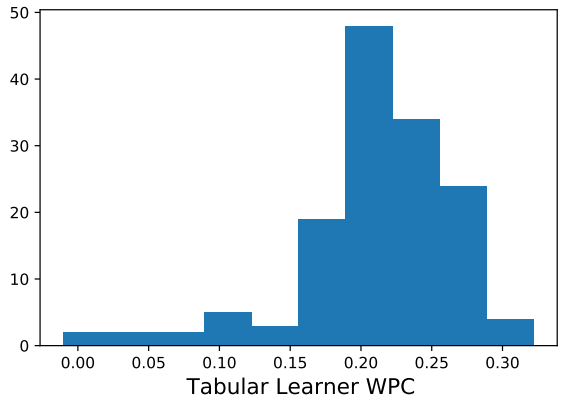

Figure 8: Histogram of performance of Tabular Learner with custom callbacks over 2000 samples. The mean is WPC = 0.21, however the extreme values go from 0 (basically a random model) to 0.32 which is by its own already close to experimental replicate accuracy.

3.3 Prediction including molecular structure

In this section, we decided to incorporate information about the drug's molecular structure to perform inference. The dataset provided the SMILES of the drugs for less than half the samples. However, as we discussed in **Methods: Molecular Fingerprints**, the majority of ML models can't process plain text as a human would do, we have to transform this feature prior to training the algorithms. For that, we rely on the R package "rcdk" [22] to convert this plain text feature into a 166 binary vector called MACCS fingerprints.

These vectors contain information about particular structural motifs, that the models will leverage to determine how synergistic a pair is. There is a lot of possible feature engineering that we could do to these vectors: performing a PCA, passing through an Autoencoder, etc. However, recent studies showed decent performance for similar tasks with simpler approaches [20], so we try 2 different methods and compare: leaving the features as they are and performing the binary OR operation on the vectors.

When we leave the vectors as they are, we give the models all the information available, without making any assumptions on the ideal representations to learn. This comes at the cost that the data is "harder to digest". In contrast, by performing the binary operation OR, we lose total information but

Models	Weighted Pearson Correlation score			
	Without commutativity	With commutativity	With commutativity and callbacks	With commutativity, callbacks and LB set
Random Forest	0.19	0.19	0.19	0.23
XGBoost	0.21	0.26	0.26	0.30
Tabular Learner	0.22	0.18	0.23	0.22
SVR	0.12	0.15	0.15	0.15
Ensemble	0.26	0.27	0.29	0.32

Table 1: Performance comparison with different data processing and training regimes. The ensemble model consists of the average of the predictions given by each model.

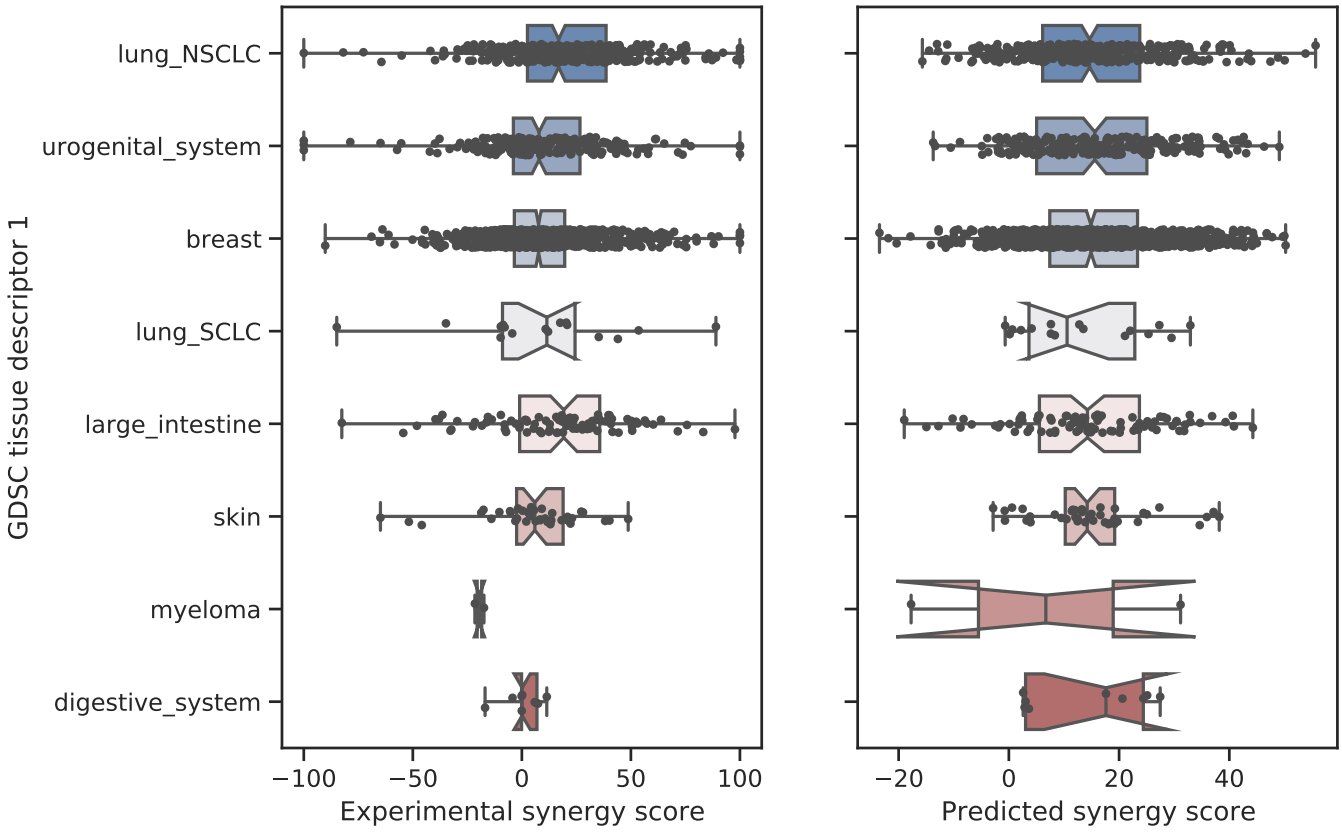

Figure 9: Experimental (left) and predicted (right) synergy score distribution over tissue of origin of the cell line.

Model	Weighted Pearson Correlation score		
	WPC without FP	WPC with FP	Binary Operation OR
Random Forest	0.25	0.26	0.29
Tabular Learner	0.21	0.23	0.22
XGBoost	0.26	0.28	0.26
SVR	0.17	0.24	0.22
Ensemble	0.31	0.33	0.32

Table 2: Performance comparison without and with molecular structure information.

create new features that may be more relevant for this particular problem [19].

Since the new dataset is half the size of the previous one, the performance that we obtained before isn't a good benchmark for the improvement seen after adding structural information, it's a "new" dataset after all. After evaluating our models on this dataset, we obtain a WPC score of 0.31 for the ensemble, which makes sense with previous results (Table: 2 column 1).

When we add the MACCS vectors, even without any processing at all, we see an improvement in all of our models (Table: 2 column 2). Although the increase is more modest for Random Forest (WPC +0.01) and XGBoost (WPC +0.02), the always lagging model, SVR, experiences a huge improvement in performance (WPC +0.07) that puts it before Tabular Learner, whose training we already demonstrated is really unstable. After ensembling we obtain $WPC = 0.33$, WPC + 0.02 compared to no structural data, which demonstrates that structure is an important predictor of drug synergy.

At last, we add the vector resulting from performing the OR operation over both MACCS vectors. The information that this vector contains is if a particular structural motif is present in at least one of the drug. After training and evaluating with these new features (Table: 2 column 3) we find that for Random Forest there is a big improvement, even when compared with just giving the plain vectors (WPC +0.03). But for XGBoost and SVR it's the opposite, both occur in a drop in performance of WPC -0.02 with respect to non-processed vectors. In the end, the ensemble suffers from a loss of WPC -0.01 when we process the vectors.

4 Discussion

If we were one of the participants of the challenge, our job would be already done. However, we can't lose focus on what is really the point of our work. We want to accurately predict the synergy score of drug pairs on cell lines. To correctly evaluate the models, we made use of a custom metric that focused on correlation more than on the exact value of the synergy score, which varies from experiment to experiment.
For the dataset without structural information we obtained a WPC score of 0.32, close to levels of experimental replicates, but when we plot the distributions of synergy score over the tissue of origin of cell line (Fig. 9), we realize that we lose statistical significance in the difference of synergy between some tissues of origin. For example, in the Menden et al. paper [11] they pointed out that lung (Non Small Cell Lung Cancer) cell lines had over twofold higher mean synergy than breast cell lines (t-test P = 7e-27), but our model doesn't capture that fact. The same happens for lung and urogenital cancer, lung's mean synergy is higher than urogenital's in experiments with P ¡0.005, but our model doesn't reflect such difference. The reason is that the domain of our predictions goes from -25 to 55 and the experimental data from -100 to 100. We are asymmetrically compressing the domain, which shifts means and kills the statistical differences.

5 Conclusions

To sum up our work, we have trained 4 different models (Random Forest, Tabular Learner, XGBoost and SVR) and ensembled them into a model that performs close to experimental replicability levels of accuracy (WPC = 0.32) (Table: 1). This was a hard task, both because of the nature of the problem (experimental variability) and the small size of the dataset for a machine learning competition. These results would put us in the top 20 of the 160 teams that competed in the DREAM challenge 2015, a series of competitions aimed at researchers in computational biology and bioinformatics all around the world.

Second of all, we demonstrated that information regarding the molecular structure of the drugs improved prediction, both the unprocessed MACCS vector and the resulting from binary operation OR (Table: 2). This serves as proof that structure has a role in how synergistic a drug pair is.

Lastly, we need to point out that there is a subset of 25 drug pairs where we didn't have any success in predicting correlation (Fig. 6). A possible explanation was provided based on the low number of cell lines where these pairs were experimented on (Fig. 7) but a further analysis needs to be addressed in further works.

6 Acknowledgments

We thank IFIMAC for its Master's Grants, AstraZeneca and the Sanger Institute for creating the dataset and releasing the dataset to the public, Sage Networks (https://sagebionetworks.org/) for hosting the competition and Google Colaboratory for providing the computing resources to train our models.

References

[1] Gambardella, V. *et al.* Personalized medicine: Recent progress in cancer therapy. *Cancers* **12** (2020). URL `https://www.mdpi.com/2072-6694/12/4/1009`.

[2] Holohan, C., Van Schaeybroeck, S., Longley, D. B. & Johnston, P. G. Cancer drug resistance: an evolving paradigm. *Nature Reviews Cancer* **13**, 714–726 (2013).

[3] W. Humphrey, R. *et al.* Opportunities and Challenges in the Development of Experimental Drug Combinations for Cancer. *JNCI: Journal of the National Cancer Institute* **103**, 1222–1226 (2011). URL `https://doi.org/10.1093/jnci/djr246`. `https://academic.oup.com/jnci/article-pdf/103/16/1222/7672885/djr246.pdf`.

[4] Bulusu, K. C. *et al.* Modelling of compound combination effects and applications to efficacy and toxicity: state-of-the-art, challenges and perspectives. *Drug discovery today* **21**, 225–238 (2016).

[5] Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications. *Drug discovery today* **20**, 318–331 (2015).

[6] Vamathevan, J. *et al.* Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery* **18**, 463–477 (2019).

[7] Zhang, L., Tan, J., Han, D. & Zhu, H. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today* **22**, 1680–1685 (2017).

[8] Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **35**, 1798–1828 (2013).

[9] Joseph, M. Pytorch tabular: A framework for deep learning with tabular data. *arXiv preprint arXiv:2104.13638* (2021).

[10] Dahouda, M. K. & Joe, I. A deep-learned embedding technique for categorical features encoding. *IEEE Access* **9**, 114381–114391 (2021).

[11] Menden, M. P. *et al.* Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen - nature communications. *Nature* (2019). URL `https://doi.org/10.1038/s41467-019-09799-2`.

[12] S., L. Effect of combinations : mathematical basis of problem. *Arch. Exp. Pathol. Pharmakol.* **114**, 313–326 (1926). URL `https://cir.nii.ac.jp/crid/1571980075578237056`.

[13] Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **28**, 31–36 (1988).

[14] Quinlan, J. R. Induction of decision trees. *Machine learning* **1**, 81–106 (1986).

[15] Breiman, L. *Machine Learning* **45**, 5–32 (2001).

[16] Chen, T. & Guestrin, C. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016). URL `http://dx.doi.org/10.1145/2939672.2939785`.

[17] Nielsen, D. *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?* Master's thesis, NTNU (2016).

[18] Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Statistics and computing* **14**, 199–222 (2004).

[19] Heaton, J. An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016*, 1–6 (2016).

[20] Malyutina, A. *et al.* Drug combination sensitivity scoring facilitates the discovery of synergistic and efficacious drug combinations in cancer. *PLoS computational biology* **15**, e1006752 (2019).

[21] Li, M. *et al.* Deepdsc: a deep learning method to predict drug sensitivity of cancer cell lines. *IEEE/ACM transactions on computational biology and bioinformatics* **18**, 575–582 (2019).

[22] Guha, R. Chemical informatics functionality in r. *Journal of Statistical Software* **18**, 1–16 (2007).

[23] Briscoe, E. & Feldman, J. Conceptual complexity and the bias/variance tradeoff. *Cognition* **118**, 2–16 (2011).

[24] Belkin, M., Hsu, D., Ma, S. & Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* **116**, 15849–15854 (2019).

[25] Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *CoRR* **abs/1803.09820** (2018). URL `http://arxiv.org/abs/1803.09820`. `1803.09820`.

[26] Fallahi-Sichani, M., Honarnejad, S., Heiser, L. M., Gray, J. W. & Sorger, P. K. Metrics other than potency reveal systematic variation in responses to cancer drugs. *Nature chemical biology* **9**, 708–714 (2013).

[27] Van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *Journal of machine learning research* **9** (2008).

Appendix 1: Hill dose-response curves

The drugs selected for the challenge can act as monotherapy treatment for tumors. The effectiveness follows a dose-response sigmoid curve that can be fitted to a Hill equation:

$$E(a) = 100 + \frac{E_\infty - 100}{1 + \left(\frac{I_{50}}{a}\right)^H}, \tag{3}$$

where a is the concentration used and E is the cell viability (rate of survival). The fitting parameters are IC_{50}, E_∞ and H (Fig. 10), which correspond to the concentration for 50% survival, the maximum kill and the slope of the curve respectively. These, along with the maximum concentration for each drug, are the monotherapy features provided in the dataset.

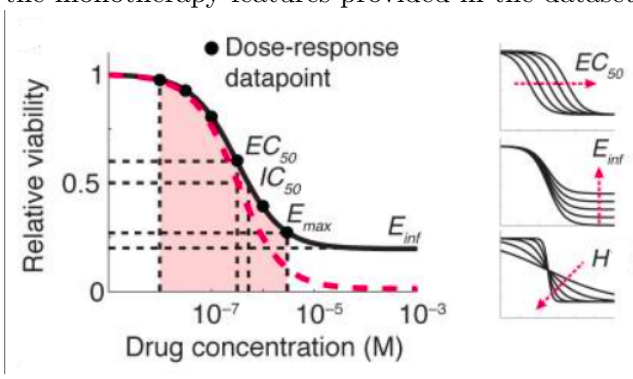

Figure 10: Sketch of Hill function. The sub-panels explain how the shape of the curve changes when we modify each fitting parameter. Extracted from [26]

Appendix 2: Categorical embeddings

One of the novelties of our work was the use of Tabular Learner, a neural network with categorical embeddings layers [10] to search for similarities between drug compounds and cell lines and exploit them to increase the accuracy of our predictions. The categorical embedding layer works like a funnel, the size of the output vector is always lower than the input, so it forces the network to look for common patterns in these categories. The output size of our embedding was decided by the rule of hand:

$$d_{out} = 1.6\, n^{0.56}, \tag{4}$$

where n is the cardinality of the feature. This leads to the embedding size marked by Table 3.

Feature	Cardinality	d_{out}
Cell line name	84	19
Compound A	69	17
Compound B	69	17

Table 3: Dimension of output vector of embedding for each feature.

Now these features are represented in a space of approximately dimension 20, but we can visualize them in a 2D map using a dimensionality reduction technique like t-SNE [27], a method that preserves relative distances from input vectors to output vectors. In competitions of other domains (sales prediction, housing market, etc.) after visualizing the embedding with t-SNE we would see clusters of categories which share some familiarity. However, when we plot our maps (Fig. 11) we don't see any meaningful structure.

Appendix 3: Custom Callbacks for Tabular Learner

To improve the performance and alleviate the training instability of the Tabular Learner, we decided to use Callbacks during the training. A Callback is an object that can perform actions during the process of training. In our case, we created an EarlyStopping Callback that stops training the model when the Pearson correlation stops increasing or does so slower than a set threshold.

This threshold is set by two parameters: min_delta, which is the minimum increase we consider and the patience, defined as the maximum number of epochs we wait for an increase to happen. Through trial and error, we determine that the optimal values for these parameters are min_delta=0.01 and patience=20.

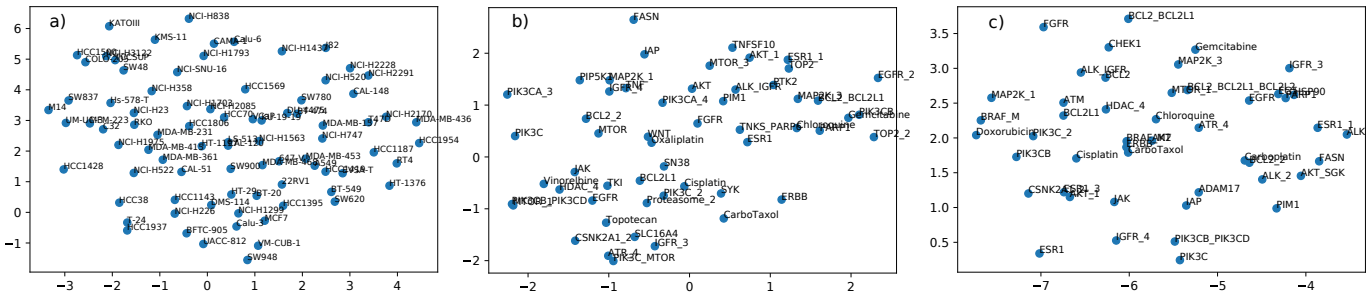

Figure 11: t-SNE visualization of a) cell line b) compound A and c) compound B embeddings extracted from the Tabular learning model

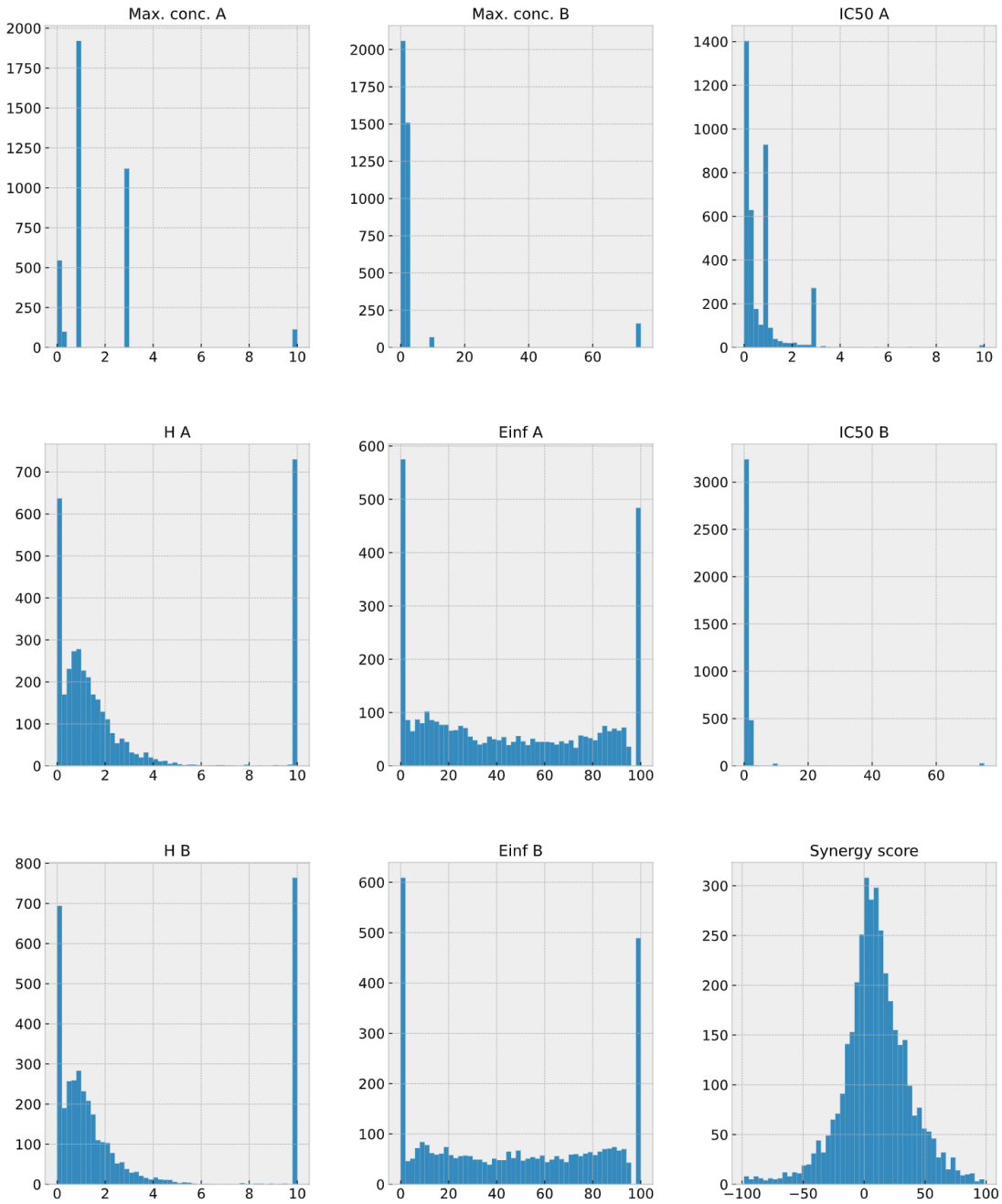

Figure 12: Histograms of monotherapy features and synergy score

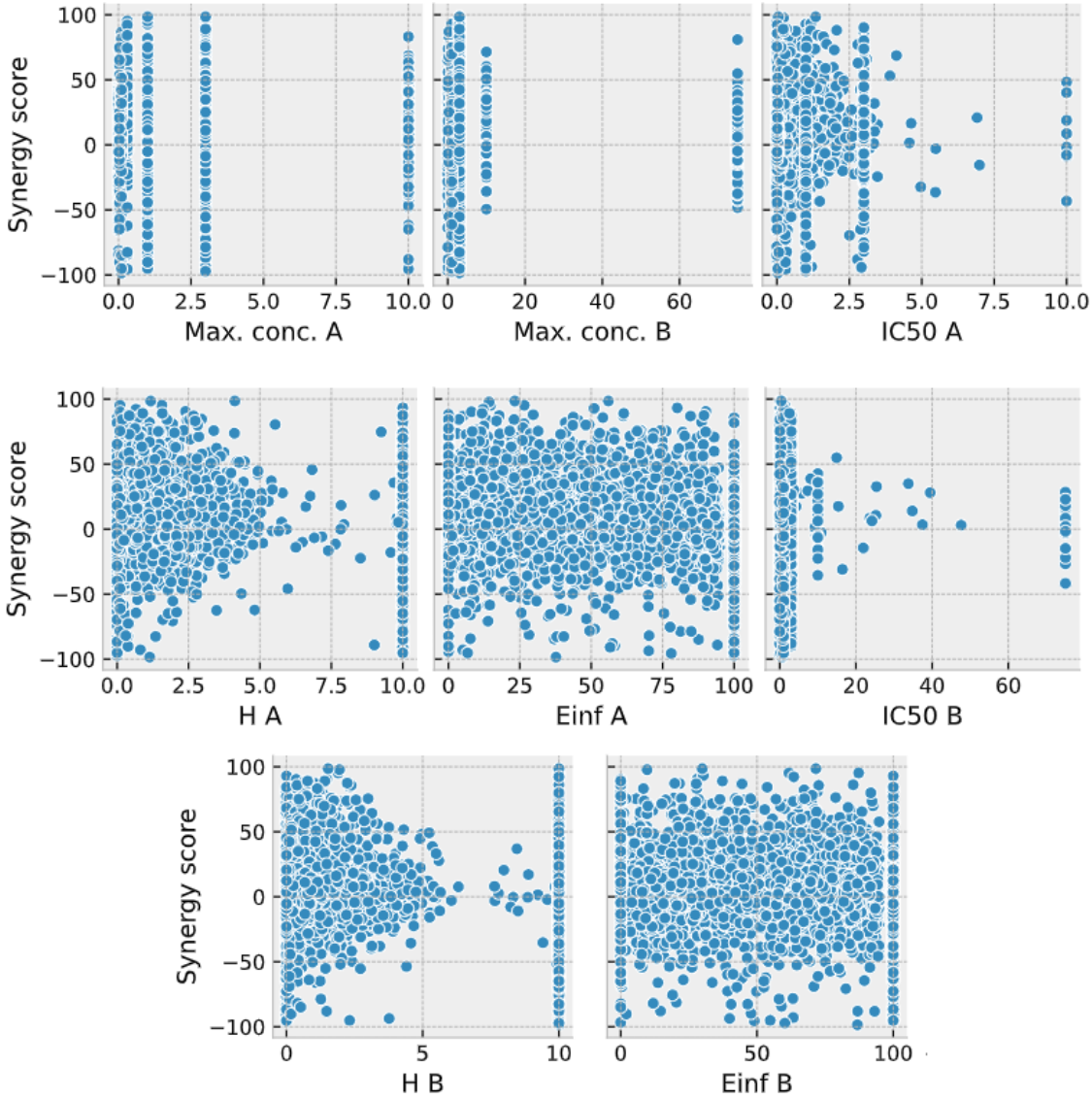

Figure 13: Synergy score as a function of monotherapy features. There is no obvious relationship between the synergy and each feature isolated, however, we can appreciate some funneling of the synergy as the slope of the curve H and the IC_{50} becomes larger.

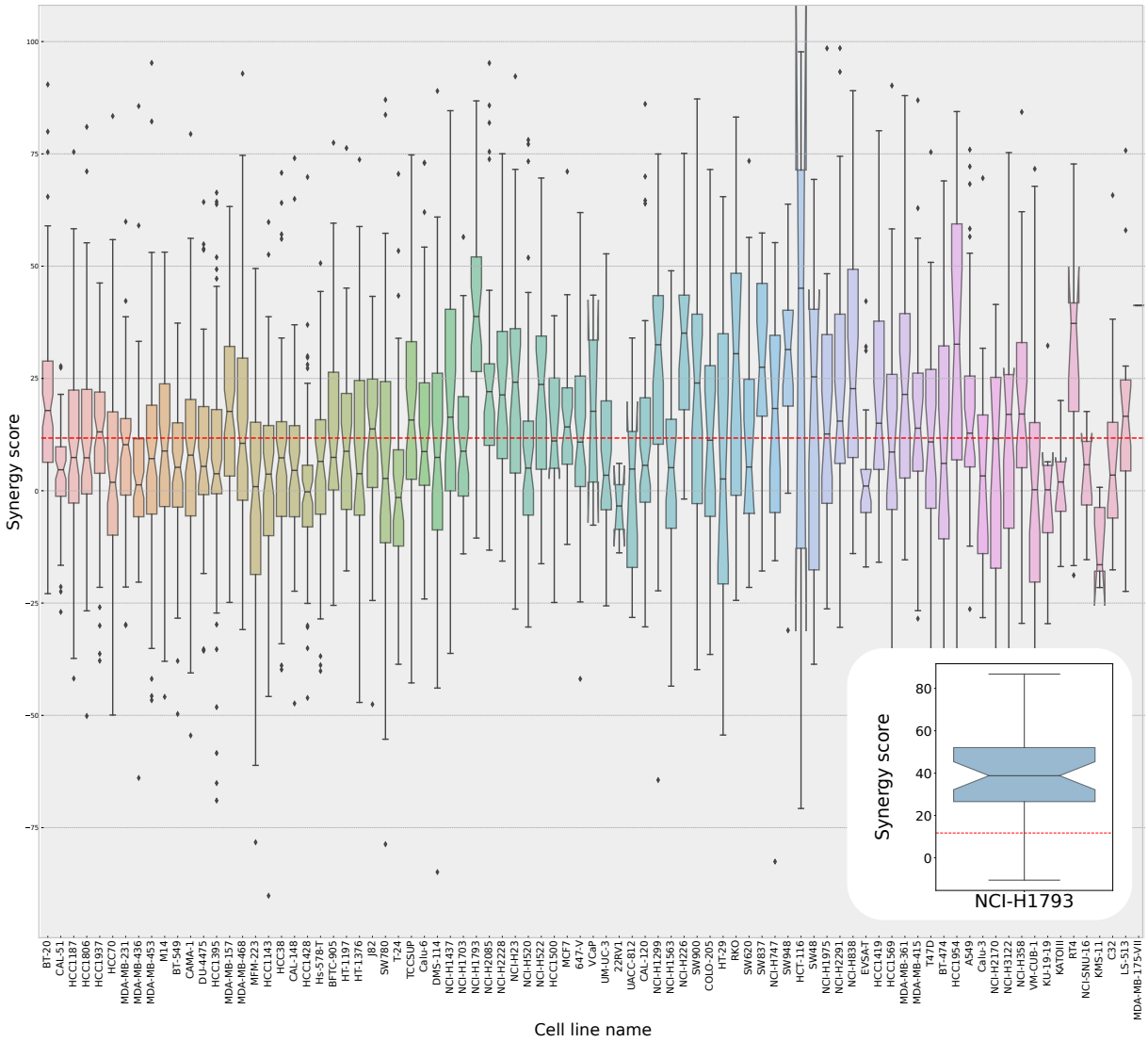

Figure 14: Synergy score as a function of cell line name. The dashed red line marks the mean synergy score.

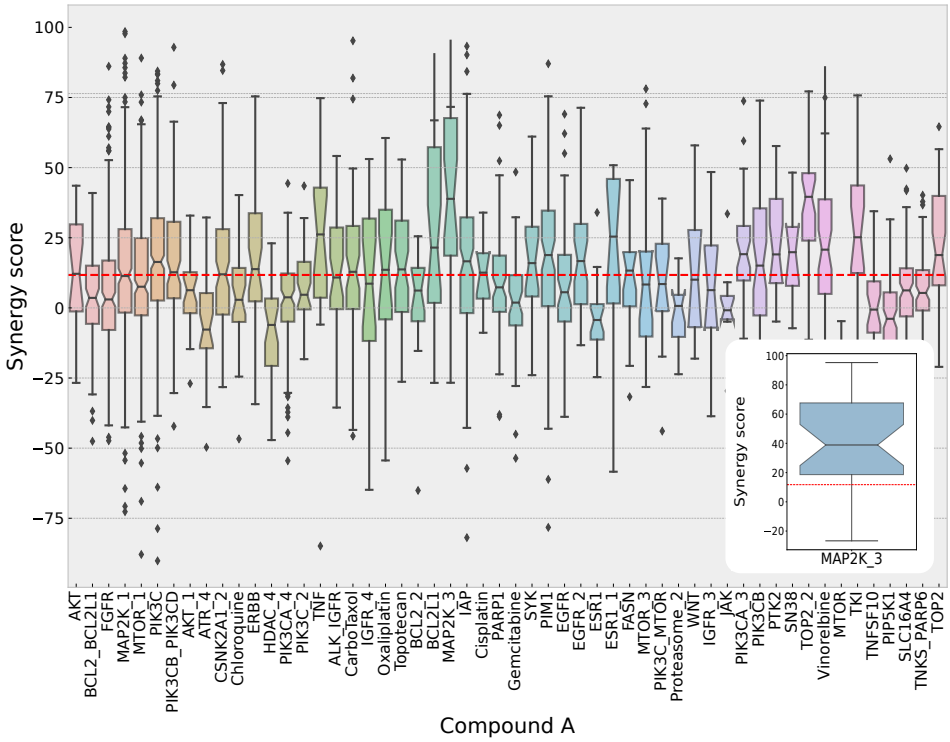

Figure 15: Synergy score as a function of compound A drug name. The dashed red line marks the mean synergy score.

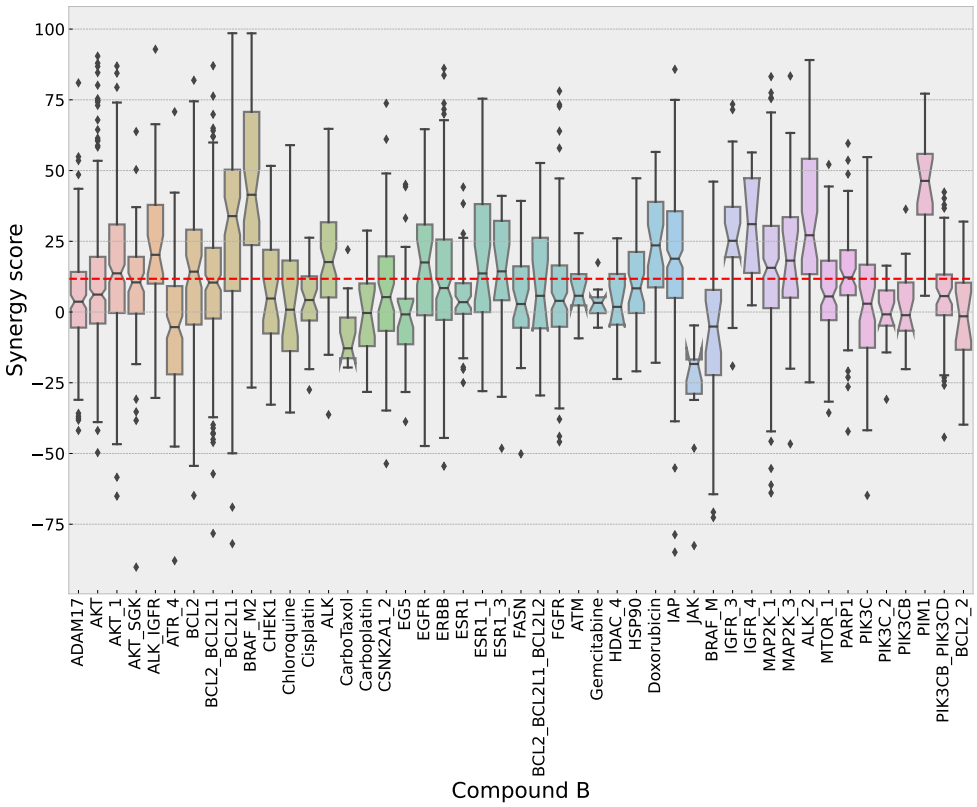

Figure 16: Synergy score as a function of compound B drug name. The dashed red line marks the mean synergy score.