Title: Predictive Modeling of Single-Cell Response to Small Molecule Perturbations Using Integrative Bioinformatics Approaches

Name: Brandon Tong, Chun Hei (Jacky) Yiu
Email: brandonidas@gmail.com, jackyyiu0810@gmail.com

December 17, 2023

1 Introduction

For our project, we will participate in a Kaggle competition (https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/overview). The competition looks into the intricacy of cellular responses to chemical perturbations, which necessitates advanced predictive models, especially in drug discovery and therapeutic interventions. This research proposal aims to develop a robust predictive model leveraging bioinformatics to understand the alterations in gene expression within various cell types in response to small molecule interventions. Our approach integrates biological knowledge, focusing on single-cell data and employing innovative model designs for accuracy, robustness, and interpretability. The dynamism of cellular functions and their responses to external stimuli, small molecules, remains a paramount concern within bioinformatics and pharmacology. Current methods, including auto-encoders like Dr.VAE [12], scGEN [9], and ChemCPA [6], show promise but suffer due to limited benchmarking datasets and poor generalization across diverse cell types.

This project proposes a systematic approach to predict cellular responses, considering the multifaceted nature of biological systems and the nuanced impact of chemical perturbations. By incorporating single-cell sequencing data and utilizing advanced analytical models, we aim to address the gaps in predictive accuracy and applicability in real-world biological systems.

The competition supplied us with a dataset encompassing differential expression data for 18,211 genes, based on the interactions between six types of human peripheral blood mononuclear cells (PBMCs) and 144 chemical compounds. Our objective is to devise a model capable of predicting gene differential expression in response to specific PBMCs and chemical combinations. Comprehensive information about the dataset and the methodology employed in data generation is detailed in the Methods section.

2 Results

2.1 Feature Engineering: Enhancing Data with Cheminformatics Techniques

The initial phase of our project focused on the critical task of feature engineering. Our analysis identified that the primary relevant columns in our dataset were the cell types and SMILES data. SMILES, an acronym for Simplified Molecular Input Line Entry System, is a widely-used text representation of chemical structures in cheminformatics. We aimed to employ advanced cheminformatics methods to extract the maximum possible information from the SMILES data, thereby enriching our dataset.

2.1.1 Morgan Fingerprint: A Compact Molecular Representation

We utilized the RDkit cheminformatics toolkit [2] to transform the SMILES data of each entry into a 2048-bit binary vector known as the Morgan Fingerprint. This technique is a well-established method in cheminformatics for representing molecular structures in a concise binary format. It greatly aids in the comparison and analysis of chemical compounds.

The essence of Morgan Fingerprinting lies in its ability to capture the local structural features surrounding each atom within a molecule. This means that distinct molecules yield unique fingerprints, while chemically similar molecules exhibit more analogous fingerprints. This approach provides a far more detailed and informative representation of chemical information compared to simpler methods like one-hot encoding.

2.1.2 Molecular Descriptor Extraction: Unlocking Chemical and Physical Insights

In addition to generating Morgan Fingerprints, we extracted a suite of 210 numerical molecular descriptors using the RDkit package. These descriptors encompass a range of chemical and physical properties, including molecular weight, the LogP (Octanol-Water Partition Coefficient), the Wiener Index, and others.

This extraction of chemical and physical properties from SMILES data is a pivotal step for developing sophisticated prediction models in the realms of cheminformatics and drug discovery. While SMILES offers a straightforward, textual depiction of molecular structures, it lacks the numerical detail required for machine learning applications. Molecular descriptors effectively convert the structural data encoded in SMILES into a numerical format, providing our predictive models with a rich, quantifiable dataset to work with.

2.1.3 Principal Component Analysis

In total, we have 18211 target values representing the differential expression of 18211 genes. We decided to use PCA to reduce the number of columns. We decided to reduce the target to 150 columns, with a cumulative explained variance of 97%.

We also decided to use PCA on the 2048 + 210 drug features that we just generated. We reduced the drug features to 10 PCs, which have a cumulative explained variance of 99%.

2.1.4 Cell Type Encoding

For cell type encoding, we use simple one-hot encoding for the 6-cell type. We thought of a hierarchical representation based on cell lineage but instead realized that the lineage of each cell played a significant part in its gene expression. This is probably something a neural network could learn, so we opted for one-hot encoding.

2.2 Baseline Boosting Model

In our initial approach, we selected CatBoost as our foundational model. Recognized for its effectiveness in handling categorical data (hence the 'Cat') and employing gradient boosting techniques ('Boost'), CatBoost operates on gradient-boosted decision trees. This choice was strategic, providing a solid reference point for evaluating our custom-developed model.

For this model, we did not use any feature engineering. We just used cell type and drug name as our categorical data. We did use the 150 PC on the expression data as the target columns to shorten training time. The Kaggle competition provided us with a test input called id_map.csv, which consists of cell types and drug pairs to be used as a test input. We do not know the actual target value of the testing data, we just use our model on the provided test input and submit our prediction to the competition website and it automatically calculates the Mean Row-wise Root Mean Squared Error (MRRMSE) of our prediction.

Our Catboost model got a score of 0.772 on the public test and 1.018 on the private test, while the score is not ideal (for MRRMSE, the lower the score the better), it was an anticipated outcome considering that it served as our baseline model.

2.3 Neural Network Model

Next, in our pursuit to enhance prediction accuracy, we ventured into developing a neural network model. For training this network, we implemented a standard 80/20 split for train-validation on our training data. ReLU activation functions were utilized to counteract the issue of vanishing gradients.

Once the training phase was completed using our engineered features introduced above, we deployed this neural network model on the test data and submitted our results to the competition's website. The performance of our neural network model markedly surpassed that of our CatBoost baseline, achieving a score of 0.656 in the public test and 0.888 in the private test.

2.4 Variational Auto-Encoder

In the final phase of our project, we endeavored to construct a Variational Autoencoder (VAE) model. Initially, this approach showed promise; however, we soon faced several challenges that ultimately hindered the successful development of the model. A critical issue we encountered was the phenomenon of exploding gradients, which could be attributed, in part, to the high dimensionality of the data. This complexity likely exacerbated the instability in gradient calculations.

To address this, we implemented various mitigation strategies. These included the normalization of activations in both batch and layer levels, which is intended to stabilize the learning process by maintaining the mean and variance of the layers' inputs. We also employed gradient clipping, a technique where gradient values are capped within a specific range to prevent excessively large updates that could destabilize the neural network.

Further, we experimented with reducing the batch size and scaling down the model size. Unfortunately, these measures did not yield the desired results. Another significant constraint was the computational resources; as we neared the completion of our project, we exhausted the free compute resources provided by the competition, which limited our ability to train and refine the VAE model extensively.

2.5 Final Result

The competition concluded on November 30, 2023. We submitted the best results from our neural network model, which demonstrated a notable level of accuracy, achieving a Mean Row-wise Root Mean Squared Error (MRRMSE) score of 0.656 in the public test and 0.888 in the private test. To put these results into perspective, the competition's winning entry achieved scores of 0.523 in the public test and 0.729 in the private test.

The fact that even the first-place score was above 0.5 in the public test illustrates the inherent complexity and challenge of the problem we were trying to solve. This observation underscores the intricate nature of the task at hand, involving predicting interactions between various cell types and drugs, a task that evidently pushes the boundaries of current machine learning capabilities.

Figure 1 presents a histogram comparing the scores of our different models with the first-place winner in the competition. This comparison clearly demonstrates the relative performance of our model. Note there is no data for VAE, as the model failed to predict meaningful results.

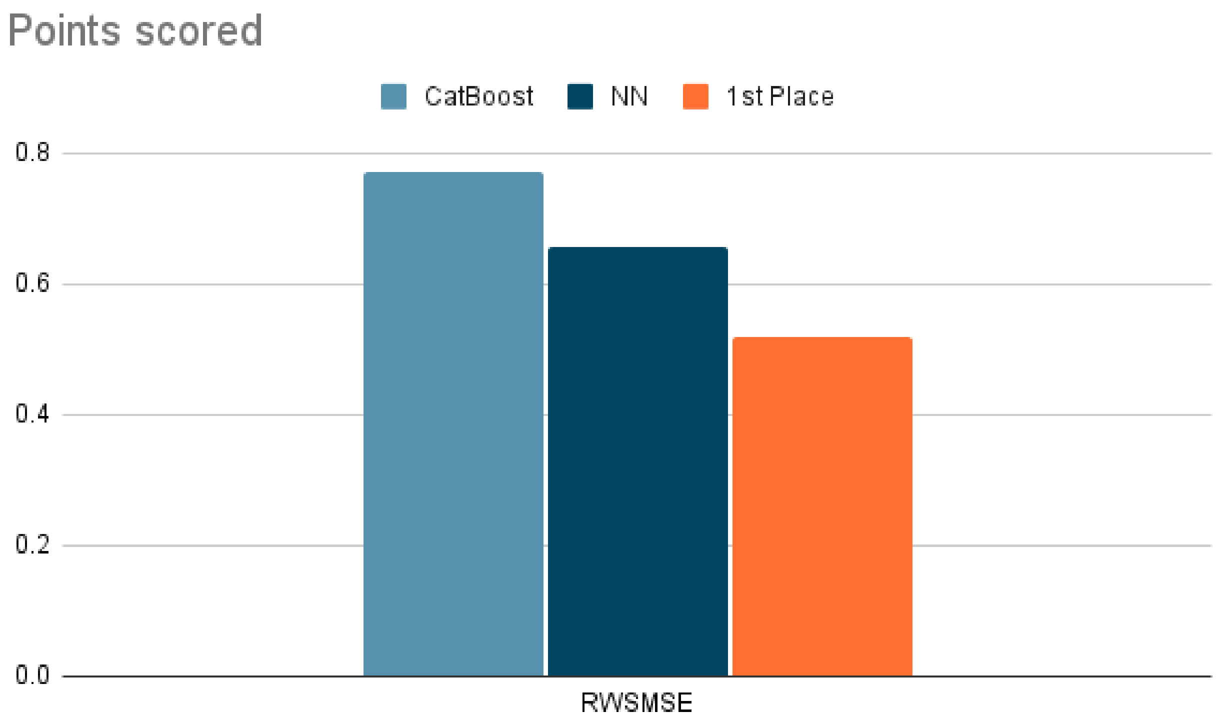


Figure 1: Histogram comparing the scores of our different models with the first-place winner

3 Methods

3.1 Data Generation

For this competition, we obtained a unique dataset involving the perturbation of individual cells in human peripheral blood mononuclear cells (PBMCs). A total of 144 compounds were carefully chosen, and their effects on single-cell gene expression profiles were measured 24 hours after treatment.

The process involved thawing and plating PBMCs from donors onto 96-well plates. Two columns were assigned for positive controls (dabrafenib and belinostat), one column for a negative control (DMSO), which served as the solvent for the compounds. The remaining wells on the plate were designated for each of the 72 compounds. The complete dataset comprised two different compound plates per donor, resulting in a total of six plates. The illustration of the experiment setup is shown in figure 2.

It's essential to note that each well-contained PBMCs, encompassing various cell types such as T cells, B cells, NK cells, and Myeloid cells like Macrophages and Monocytes. Using gene expression data measured in single-cell RNA sequencing (scRNA), computational methods were employed to assign each cell to a specific cell type.

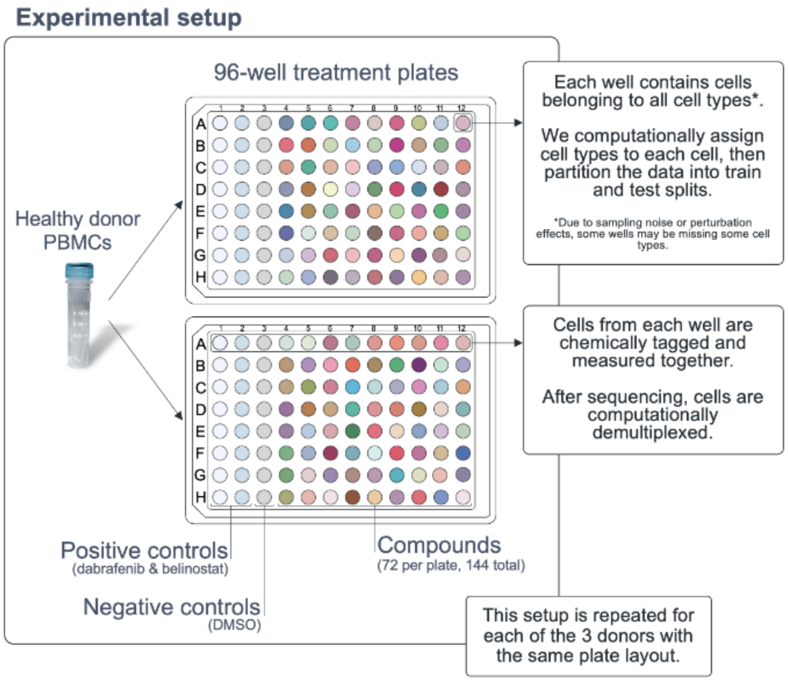

Experimental setup
96-well treatment plates
1
2
3
4
5
6
7
8
9
10
11
12
A
B
C
D
E
F
G
H
Healthy donor PBMCs
Each well contains cells belonging to all cell types*.
We computationally assign cell types to each cell, then partition the data into train and test splits.
*Due to sampling noise or perturbation effects, some wells may be missing some cell types.
Cells from each well are chemically tagged and measured together.
After sequencing, cells are computationally demultiplexed.
Positive controls
(dabrafenib & belinostat)
Negative controls
(DMSO)
Compounds
(72 per plate, 144 total)
This setup is repeated for each of the 3 donors with the same plate layout.

Figure 2: Diagram illustrating the experimental process.

3.2 Differential Expression Calculation

The main goal of this competition is to model differential expression (DE) to evaluate how experimental perturbations affect gene expression levels. In this dataset, we focus on the transcriptional responses of 18,211 genes. DE is estimated by averaging the raw gene expression counts for each cell type in a sample, a method referred to as pseudo-bulking in single-cell research. This involves summing the raw counts for all cells of the same type in each well of the experiment.

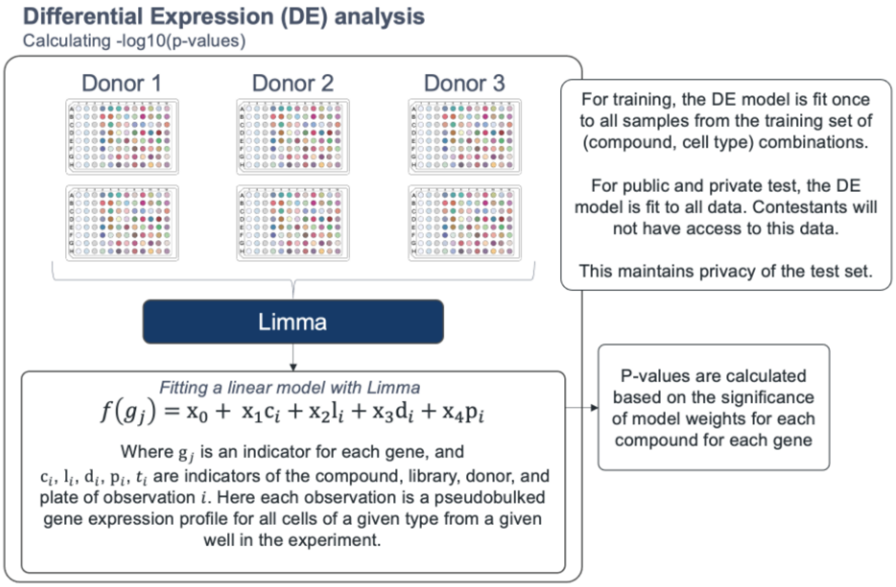
To achieve this, the Limma package (a Bioconductor package for linear models) is used to fit a linear model to the pseudo-bulked RNA counts data. This model incorporates technical covariates like library (row), plate, and donor, in addition to the experimental covariate, which is the compound. Figure 3 illustrates the differential expression calculation process.

The output of this model is an estimated fold-change in gene expression and a multiple-testing corrected p-value that a given gene's expression is dependent on the compound experimental variable.

3.3 Data Splits

Our task is to predict differential expression values for Myeloid and B cells for a majority of compounds. We train our model on data from all 144 compounds in T cells (CD4+, CD8+, regulatory) and NK cells and 10% of compounds in Myeloid and B cells. This mirrors a scientific context where one might want to make predictions about new cell types while taking only 1/10th of the measurements in that cell type.

The distribution of train/test split across cell types is shown in Figure 4.


Differential Expression (DE) analysis
Calculating -log10(p-values)
Donor 1
Donor 2
Donor 3
Limma

For training, the DE model is fit once to all samples from the training set of (compound, cell type) combinations.

For public and private test, the DE model is fit to all data. Contestants will not have access to this data.

This maintains privacy of the test set.

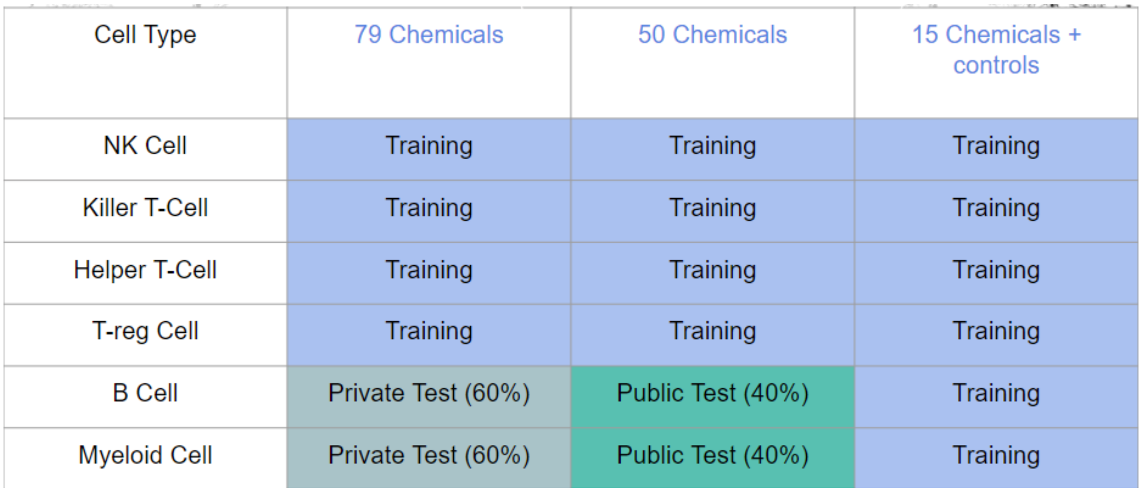
Fitting a linear model with Limma

$$f(g_j) = x_0 + x_1 c_i + x_2 l_i + x_3 d_i + x_4 p_i$$

Where g_j is an indicator for each gene, and c_i, l_i, d_i, p_i, t_i are indicators of the compound, library, donor, and plate of observation i. Here each observation is a pseudobulked gene expression profile for all cells of a given type from a given well in the experiment.

P-values are calculated based on the significance of model weights for each compound for each gene

Figure 3: Diagram illustrating the calculation of Differential Expression (DE).



Cell Type	79 Chemicals	50 Chemicals	15 Chemicals + controls
NK Cell	Training	Training	Training
Killer T-Cell	Training	Training	Training
Helper T-Cell	Training	Training	Training
T-reg Cell	Training	Training	Training
B Cell	Private Test (60%)	Public Test (40%)	Training
Myeloid Cell	Private Test (60%)	Public Test (40%)	Training

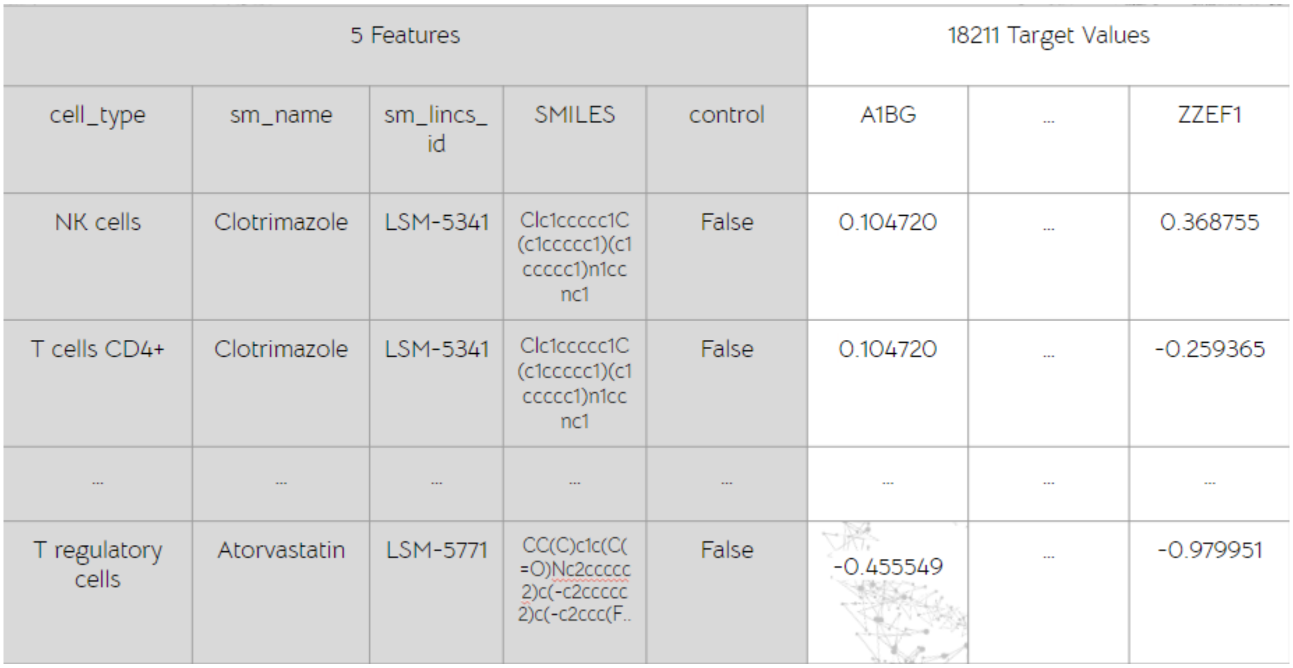
Figure 4: Diagram illustrating the Train-Test split of the data provided).

3.4 Data Field Descriptions

Our dataset, named `de_train.parquet`, contains differential expression data in a dense array format, with key columns as follows:

- **Genes (Total 18,211)**: This column lists genes (e.g., A1BG, ZZEF1) with their differential expression values, calculated as $-\log_{10}(\text{p-value}) \times \text{sign(LFC)}$, where LFC represents log-fold change in expression.
- **Cell Type**: Denotes the cell type based on RNA expression.
- **sm_name**: The primary name for compounds, standardized by LINCS.
- **sm_lincs_id**: Global LINCS ID for each compound.
- **SMILES**: Simplified molecular-input line-entry system representations of compounds.
- **Control**: Boolean value indicating control instances.

A visualization of our data is shown in Figure 5.



5 Features					18211 Target Values		
cell_type	sm_name	sm_lincs_id	SMILES	control	A1BG	...	ZZEF1
NK cells	Clotrimazole	LSM-5341	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1	False	0.104720	...	0.368755
T cells CD4+	Clotrimazole	LSM-5341	Clc1ccccc1C(c1ccccc1)(c1ccccc1)n1ccnc1	False	0.104720	...	-0.259365
...	...	...	...	...	...	...	...
T regulatory cells	Atorvastatin	LSM-5771	CC(C)c1c(C(=O)Nc2ccccc2)c(-c2ccccc2)c(-c2ccc(F..	False	-0.455549	...	-0.979951

Figure 5: Table illustrating our training data.

3.5 Feature Engineering

We use chemo-informatics methods like extracting Morgan fingerprints and molecular descriptors from SMILES data. We also utilize Principal Component Analysis (PCA) for dimension reduction.

3.5.1 Morgan Fingerprint

Morgan fingerprints, or extended-connectivity fingerprints (ECFPs), encode molecular structures into a binary format for cheminformatics analysis [8]. This method, based on atom connectivity, offers a compact representation of molecules [4].

For this project, we transformed SMILES data into Morgan fingerprints through these steps:

1. **Molecular Parsing**: Convert SMILES strings into molecular graphs, with atoms and bonds as nodes and edges.
2. **Feature Extraction**: Iteratively analyze each atom's local environment, generating a numerical value for it.
3. **Hashing and Vectorization**: Hash these values into a 2048-bit vector, indicating the presence or absence of molecular substructures.
4. **Fingerprint Generation**: The final bit vector, representing the molecular structure, is used for data analysis.

This 2048-bit vectorization offers a more detailed molecular representation than One-Hot Encoding, capturing substructures and atom connectivity, which reflects the molecule's 3D shape and functional groups.

We use the RDKit package [2] to convert SMILES into Morgan fingerprints. RDKit is a collection of cheminformatics and machine-learning software written in C++ and Python.

3.5.2 Molecular Descriptor

Molecular descriptors are numerical values that capture various chemical properties and structural characteristics of a molecule [10]. These descriptors play a crucial role in cheminformatics and computational chemistry, providing a quantitative basis for understanding and predicting the behaviour and biological activities of chemical compounds.

Here are some examples of molecular descriptors we extracted:

- **Molecular Weight**: The total mass of all atoms in the molecule.
- **LogP** (Octanol-Water Partition Coefficient): A measure of the molecule's hydrophobicity, indicating how it partitions between a hydrophobic (octanol) and a hydrophilic (water) phase.
- **Wiener Index**: A value based on the distances between all pairs of vertices (atoms) in the molecular graph, reflecting molecular size and shape.
- **Balaban Index**: A topological index that takes into account both the distance and connectivity of the molecular graph.
- **Atom Counts**: The number of specific types of atoms (e.g., carbon, nitrogen, oxygen) in the molecule.
- **Functional Group Counts**: The number of occurrences of specific functional groups (e.g., hydroxyl, carbonyl, amine groups).

We extracted all 210 available molecular descriptors from the chemical structure using the RDkit package.

3.5.3 Principal Component Analysis

PCA, or Principal Component Analysis, is a statistical technique used to simplify complex data sets. It transforms the data into a set of orthogonal (uncorrelated) variables known as principal components [11]. These components capture the most significant variance in the data, allowing for a reduced-dimension representation while retaining essential information.

3.6 CatBoost

The core of CatBoost's effectiveness lies in its innovative approach to gradient boosting [1]. It utilizes oblivious decision trees, which ensures that each level of the tree splits on the same feature, leading to more symmetrical and balanced trees. This structure not only improves the model's performance but also enhances its interpretability.

Furthermore, CatBoost incorporates a unique algorithm for processing categorical variables. It employs an ordered boosting method that reduces overfitting and improves the model's generalizability. Additionally, the model uses a permutation-driven approach for calculating feature importance, allowing for more accurate and insightful analysis of the driving factors behind the predictions.

For our specific application, we fine-tuned the CatBoost model parameters, including learning rate, depth of trees, and the number of trees, to optimize its performance on the given dataset. The choice of loss function was guided by the nature of our prediction task, ensuring that the model's focus aligns with our predictive goals.

Overall, the adoption of the CatBoost model in our research offers a robust and efficient approach to handling complex datasets with a mix of numerical and categorical variables, ultimately contributing to more accurate and reliable predictive outcomes.

3.7 Neural Network Model

Neural networks, a cornerstone of modern artificial intelligence, have significantly impacted the field of bioinformatics, revolutionizing how biological data is analyzed and interpreted. At their core, neural networks are computational models inspired by the human brain's structure and function. They consist of interconnected nodes or neurons that work in unison to perform complex tasks. These networks are capable of learning from data through a process involving adjusting the weights of connections based on input-output examples. This adaptability makes them particularly suited for handling the vast and complex datasets typical in bioinformatics, such as genomic sequences, protein structures, and cellular images. [13]

We used a basic neural network with three layers to create a regressor on data. We speculate that neural networks can make inter-expression connections that other methods cannot. However, we also note that there are more specialized models for capturing high-dimensional interactions that may perform better.

3.8 Variational Auto-Encoder (VAE)

Variational Autoencoders (VAEs) are a class of generative models that excel in learning complex data distributions. Structurally, a VAE comprises two main components: an encoder and a decoder. The encoder transforms input data into a set of parameters in a latent space, typically capturing the mean and variance. This latent space represents a compressed, encoded version of the input data, capturing its key features. The decoder then reconstructs the input data from this latent space, aiming to generate outputs that are as close as possible to the original inputs. The beauty of VAEs lies in their ability to handle complex, high-dimensional data, making them ideal for tasks like image generation, style transfer, and more.[7]

The loss function of a VAE is a combination of two terms: a reconstruction loss and a regularization term. The reconstruction loss ensures that the decoded samples match the original inputs, typically measured using a distance metric like the mean squared error for continuous data or cross-entropy for categorical data. The regularization term, often represented as the Kullback-Leibler (KL) divergence, encourages the learned distribution in the latent space to approximate a prior distribution, usually a standard normal distribution. This balance between reconstruction and regularization allows VAEs to not only learn efficient encodings of data but also to generate new samples from the learned distribution.

The reason we speculate that a VAE may be suitable for this competition is that drug molecules often have complex, high-dimensional structures, and understanding their interactions with biological systems is a challenge in bioinformatics. VAEs, with their advanced inference and learning capabilities, can effectively model these complex structures. By capturing the intricate relationships in the latent space, VAEs can predict how different drug molecules may influence gene expression patterns.

3.9 Model Evaluation

We use the Mean Row-wise Root Mean Squared Error to evaluate the model, computed as follows:

$$MRRMSE = \frac{1}{R}\sum_{i=1}^{R}(\frac{1}{n}\sum_{j=1}^{n}(y_{ij} - \hat{y}_{ij})^2)^{\frac{1}{2}}$$

where R is the number of scored rows, and y_{ij} and $\hat{y}_{ij}$ are the actual and predicted values, respectively, for row $i$and column j, and n is the number of columns.

Explanation of terms:

Row-wise Error Calculation: Gene expression data typically consist of measurements across many genes (rows) for different conditions or samples (columns). Each gene's expression profile can be highly variable and unique. The MRRMSE calculates the error row-wise, focusing on the accuracy of predictions for each gene across different conditions or samples. This is crucial because it ensures that the model's performance is assessed based on its ability to capture the variability and patterns specific to each gene.

Root Mean Squared Error Component: The use of the Root Mean Squared Error (RMSE) within the MRRMSE accounts for the magnitude of the errors in the predictions. RMSE penalizes larger errors more severely, which is important in gene expression analysis where significant deviations from the actual expression levels can have profound biological implications.

4 Discussion

4.1 Winner: Language Models

The winner of this competition noted that "gradient boosting models, MLP, and 2D CNN .... did not work so well. I finally selected LSTM, GRU, and 1-d CNN architectures as they performed better on the validation sets. Below I show a rough implementation of the GRU mode".

Which is notable because despite the high dimensional nature of the problem supposedly being suitable for variational autoencoders, the winner of the competition used components more suited to machine translation.

"LSTM" stands for Long Short Term Memory and GRU standards for Gated Recurrent Unit, both of which are models from the natural language processing domain. Further 1D convulsions are sequences in essence. These models are known for their ability to handle sequential data, capturing long-term dependencies and nuances in the data sequence, which seems to have been more critical for this specific challenge.

We speculate that perhaps learning distributions is somewhat meaningless in respect to the problem of single cell perturbations. Perhaps the "context" of gene expressions speaks a language of its own.

4.2 Lack of Computational Resources

We were hindered by 10 hour per month limit of the use of cloud resources. So for students without access to GPUs for this competition, we were not able to scale up a model to the quality we required.

4.3 Future Work

4.3.1 Bayesian Flow Networks

Bayesian Flow Networks (BFNs) are a novel generative model where parameters of independent distributions, adjusted using Bayesian inference from noisy data, feed into a neural network to produce an interdependent distribution. This iterative process, beginning with a basic prior and continuously updating the distributions, mirrors the reverse mechanism of diffusion models but is inherently simpler as it omits the forward process. [5]

One of the authors has managed to train on SMILES (Simplified Molecular Input Line Entry System) data to predict logP values (a measure of the lipophilicity of a molecule). The code is available here.

The authors feel this is a suitable model for single cell perturbation data given distributions can be learned between high dimensional data and extrapolated to make predictions.

4.3.2 Transformers

Given the winner's success with NLP approaches, a reasonable extension would using more advanced NLP approaches - namely the transformer models with multi-head attention mechanisms [14]. Variations also exist such the LongFormer which can even further increase the context from which the model can draw from. [3]

5 Conclusion

This study embarked on an ambitious journey to predict single-cell responses to small molecule perturbations using integrative bioinformatics approaches. Despite the challenges and limitations, some trides were made in understanding and predicting cellular behavior in response to chemical compounds.

The use of cheminformatics techniques and machine learning models, including CatBoost and neural networks, proved some what capable in analyzing the intricate interactions between various cell types and drugs. The utilization of Morgan Fingerprints and molecular descriptors provided a deeper insight into the chemical structures of the compounds, augmenting predictive capabilities of our models. Additionally, the application of PCA for dimensionality reduction effectively streamlined our dataset, ensuring more focused analyses.

Our neural network model demonstrated remarkable performance improvements over the baseline CatBoost model. However, it's worth noting that the complexity of predicting cellular responses still poses a significant challenge, as evidenced by the scores achieved by even the competition's winning entries.

The challenges encountered with the Variational Autoencoder model underscore the difficulties in modeling such complex biological systems. The computational constraints further highlighted the need for more resources and innovative approaches in this domain.

Findings from the winner of the competition winner, suggest that models adept at handling sequential data, such as LSTM, GRU, and 1D CNNs, might be more suitable for this type of bioinformatics problem. This insight opens up new avenues for exploring NLP-based approaches, like transformers, in future research.

In conclusion, this project identified key areas for future exploration, such as the potential of Bayesian Flow Networks and advanced NLP techniques like transformers. The journey through this complex landscape of single-cell response prediction has challenging.

References

[1] CatBoost, . URL https://catboost.ai/en/docs/.

[2] RDKit: Open-source cheminformatics, . URL https://www.rdkit.org.

[3] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020.

[4] A. Capecchi, D. Probst, and J.-L. Reymond. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. 12(1):43. ISSN 1758-2946. doi: 10.1186/s13321-020-00445-4. URL https://doi.org/10.1186/s13321-020-00445-4.

[5] A. Graves, R. K. Srivastava, T. Atkinson, and F. Gomez. Bayesian flow networks, 2023.

[6] L. Hetzel, S. Böhm, N. Kilbertus, S. Günnemann, M. Lotfollahi, and F. Theis. Predicting cellular responses to novel drug perturbations at a single-cell resolution. URL http://arxiv.org/abs/2204.13545.

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[8] Laksh. A practical introduction to the use of molecular fingerprints in drug discovery. URL https://towardsdatascience.com/a-practical-introduction-to-the-use-of-molecular-fingerprints-in-drug-di

[9] M. Lotfollahi, F. A. Wolf, and F. J. Theis. scGen predicts single-cell perturbation responses. 16(8):715–721. ISSN 1548-7105. doi: 10.1038/s41592-019-0494-8. URL https://www.nature.com/articles/s41592-019-0494-8. Number: 8 Publisher: Nature Publishing Group.

[10] H. Moriwaki, Y.-S. Tian, N. Kawashita, and T. Takagi. Mordred: a molecular descriptor calculator. 10(1):4. ISSN 1758-2946. doi: 10.1186/s13321-018-0258-y. URL https://doi.org/10.1186/s13321-018-0258-y.

[11] K. Pearson. LIII. on lines and planes of closest fit to systems of points in space. 2 (11):559–572. ISSN 1941-5982. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/14786440109462720.

[12] L. Rampášek, D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg. Dr.VAE: improving drug response prediction via modeling of drug perturbation effects. 35(19):3743–3751. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz158. URL https://doi.org/10.1093/bioinformatics/btz158.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.