Comparative Analysis of Geneformer and Traditional Machine Learning Techniques in Predicting Perturbation Combination Efficacy on Cancer Cell Lines

An Empirical Evaluation Using the sciplex2 Dataset

Michal Krkoška[1]

Supervisor(s): Prof.dr.ir. Marcel J.T. Reinders[1]**, Niek Brouwer**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Michal Krkoška
Final project course: CSE3000 Research Project
Thesis committee: Prof.dr.ir. Marcel J.T. Reinders, Niek Brouwer, Dr. Merve Gürel

Abstract

Cancer poses a significant clinical, social, and economic burden, necessitating the development of effective treatments. Understanding how drugs interact with cancer cells and their downstream effects is critical for creating new therapies and overcoming drug resistance. This paper compares the predictive performance of the Geneformer model with traditional machine learning methods in predicting the response of cancer cells to perturbation combinations using the sciplex2 dataset.

The research involves preprocessing the sciplex2 dataset, training the models, and evaluating their performance in binary classification of cells as either treated or untreated, and the prediction of gene perturbation impacts. While traditional ML models demonstrated higher accuracy in binary classification tasks, Geneformer excelled in predicting the impact of gene perturbations due to its advanced architecture and extensive pre-training on single-cell transcriptomes.

Key findings reveal that highly expression-correlated gene pairs cause the largest shifts in cell classification, underscoring the importance of gene correlations in biological predictions. Geneformer showed a deeper understanding of gene network dynamics, achieving higher maximum Cosine Shifts compared to PCA embeddings and placing less emphasis on highly differentially expressed (HDE) Single Genes. Instead, it focused on HDE Gene Pairs, indicating its potential ability to capture complex downstream effects of gene perturbations.

This study highlights the potential of integrating advanced deep learning models like Geneformer into drug discovery, offering a pathway for more effective and targeted therapeutic interventions.

1 Introduction

Cancer poses the highest clinical, social, and economic burden in terms of cause-specific Disability-Adjusted Life Years (DALYs) among all human diseases [1]. Developing effective treatments is crucial to alleviate this burden. Understanding how drugs interact with cancer cells and their downstream effects is vital for creating new treatments and overcoming resistance to existing therapies.

The transcriptome is the set of all RNA transcripts, including coding and non-coding, in an individual or a population of cells [2]. It provides a comprehensive view of gene expression, revealing how genes are regulated and how their expression changes in response to different conditions.

Perturbations of such transcriptomes involve systematically altering the expression of specific genes to observe the resulting changes in cellular behavior. These perturbations help in understanding gene function and the interactions within gene networks [3]. In-silico analyses of these perturbations can identify potential therapeutic targets and understand disease mechanisms more effectively [4].

Examining gene perturbation combinations of cell transcriptomes is especially crucial for drug discovery. Unlike single-gene perturbations, combination analyses reveal synergistic effects and resistance mechanisms, aiding in the identification of effective drug combinations [5]. This approach helps to uncover intricate biological pathways and interactions that single-gene studies might miss. This research focuses purely on gene pair perturbations, providing a foundation for further studies into various perturbation combinations.

Machine learning (ML) models have been extensively applied to efficiently explore drug combinations from a vast array of approved and investigational chemical compounds [6]. However, the introduction of deep learning and transfer learning techniques presents a significant advancement in this domain.

The field of machine learning has recently experienced advancements in models that can be pretrained on large, generalized datasets and subsequently fine-tuned on smaller, task-specific data to enhance their predictive accuracy [7]. Geneformer is an application of the mentioned techniques in the flied of network biology. Pretrained on approximately 30 million single-cell transcriptomes, Geneformer has demonstrated refined predictive abilities in data-limited scenarios [7]. The model's capacity to identify candidate therapeutic targets through its attention-based, context-aware architecture signifies a potential leap forward in the area of drug efficacy prediction. Nevertheless, questions persist about how the Geneformer model compares against traditional machine learning methods when applied to complex, high-dimensional, and sparse datasets like sciplex2 [8].

This research paper aims to address the question of "How does the predictive performance of the Geneformer model compare to traditional machine learning methods, specifically Random Forest, Support Vector Machines, and Gradient Boosting Machines, in predicting the response of cancer cells to perturbation combinations using the sciplex2 dataset?" by evaluating the predictive performance of Geneformer in comparison to established ML techniques for predicting cancer cell responses to perturbation combinations. The evaluation is conducted in two parts. First, the models' abilities to classify cells as either treated or untreated are compared using metrics such as accuracy, precision, recall, F1 score, and AUC-ROC. Second, the models are assessed based on their ability to identify the relevance of perturbation combinations. This is achieved by examining changes in the models' embeddings, the types of identified perturbations, and shifts in classifying cells as treated instead of untreated.

The presented research contributes to the field of drug discovery by providing a detailed comparison of the predictive performance of both ML and Geneformer models, offering insights into their capabilities to handle complex, high-dimensional biological data. The evaluation of the models is based on their ability to predict responses to drug perturbations, thereby enhancing the understanding of drug effects on cancer cells using the sciplex2 dataset.

2 Methodology

This section provides a detailed description of the methods used to preprocess the sciplex2 dataset, train a selection of machine learning models, and evaluate their performance in predicting the response of cancer cells to different perturbation combinations.

2.1 Dataset Analysis

The sciplex2 dataset, which contains single-cell RNA sequencing (scRNA-seq) data, was utilized for this study. This dataset includes gene expression profiles for various cell lines affected by drugs at different doses. It is comprised of 35,653 genes and 20,964 A549 cancer cells.

Gene expression varies significantly across these cells, with a mean expression of 0.17, a median of 0.00, and a maximum of 5636.00. Detailed statistical properties of gene expression are provided in Appendix A (Table A1).

For our analysis, we focused on the subset of the dataset involving the drug "Dex" (Dexamethasone), known for inducing the expression of both CDK inhibitors (p21Cip1, p27Kip1) and cyclin-dependent kinases (CDK4, CDK6) [9], presenting a complex perturbation effect suitable for our research objectives. Statistical details and effects of "Dex" in the dataset are elaborated in Appendix A (Table A2).

We selected the dosage of 125.00 from the available dosages in μM (0.00, 0.25, 1.25, 2.50, 12.50, 25.00, 125.00, 250.00), as it represents the second highest dosage for "Dex". This dosage induces significant gene expression changes while being within a plausible range for clinical relevance, where doses up to 40.80μM are used to treat tumors [10]. The choice is justified by the balance between the sample size and the strength of gene expression shifts, as higher doses of Dexamethasone cause larger changes in gene expressions [11]. Different cell counts per dosage are visualized in Figure 1, while PCA and t-SNE were used to examine the high similarity of low dosages to untreted cells in Appendix A (Figures A4 and A5).

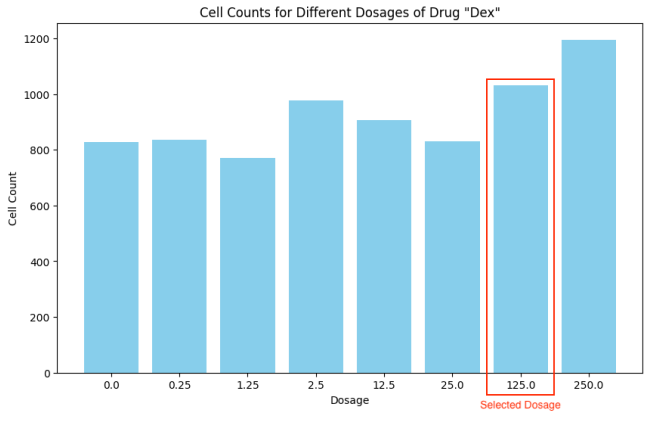

Figure 1: Counts of A549 cancer cells per different dosages [μM] of Dexamethasone in sciplex2 dataset.

2.2 Dataset Preparation and Preprocessing

The sciplex2 dataset underwent the following preprocessing steps to ensure readiness for model training and evaluation:

- **Normalization and Log-Transformation:** Gene expression values for each cell were normalized such that the total expression value across all genes in each cell summed to a target value (10,000). This normalization step helps to account for differences in sequencing depth across cells. Following normalization, a log-transformation was applied to stabilize variance and make the data more normally distributed, which is a common preprocessing step in single-cell RNA-seq data analysis [12].
- **Label Creation:** Binary labels were created to distinguish treated cells (cells treated with 125.00μM dose of "Dex") from untreated cells (cells treated with 0.00μM dose of any drug). Treated cells were labeled as 1, and untreated cells were labeled as 0.
- **Class Balancing:** Initially, there was a significant imbalance in the dataset, with 3,581 untreated cells and 1,030 treated cells. To address this, we downsampled the majority class (untreated cells) to match the number of treated cells. Specifically, we randomly selected 1,030 untreated cells from the original 3,581 to ensure both classes had an equal representation.
- **Dimensionality Reduction:** Principal Component Analysis (PCA) [13] was employed to reduce the dimensionality of the dataset. The high-dimensional gene expression data was projected onto 256 principal components. This reduction helps in capturing the most significant variation in the data while reducing the computational complexity for subsequent modeling steps. The number 256 was chosen to match the number of components used by the Geneformer model. Dimensionality reduction using PCA can have high impact on the classification performance [14], we examine the effects of PCA on the performance of ML methods in Appendix E.
- **Data Splitting:** The dataset was split into training (70%), validation (15%), and test (15%) sets. Stratified splitting was used to maintain the class balance in each subset.

Further details on the preprocessing steps are provided in Appendix A.

2.3 Selected Traditional ML Methods

Predicting the response of cancer cells to perturbation combinations necessitates ML techniques capable of handling high-dimensional, sparse data and capturing complex interactions. Based on a comprehensive comparison of state-of-the-art classification algorithms conducted by Zhang et al. (2017) [15], we selected Support Vector Machine (SVM), Gradient Boosting Machine (GBM), and Random Forest (RF).

Random Forests

Random Forests are ensemble learning methods that build multiple decision trees at training time and output the mode

of the classes (classification) or the mean prediction (regression) of the individual trees [4]. RFs are versatile and robust against overfitting, particularly effective with datasets containing many input variables, such as gene expression levels. They handle missing data well and provide strong baseline performance due to their ability to model complex nonlinear relationships with minimal parameter tuning [16].

Support Vector Machines

Support Vector Machines are powerful classifiers that find the hyperplane best separating different classes in a high-dimensional space [17]. SVMs manage high-dimensional data spaces effectively, crucial for gene expression data. Various kernels (linear, polynomial, radial basis function) can capture complex data relationships, making them ideal for modeling gene interactions and perturbation combinations [18].

Gradient Boosting Machines

Gradient Boosting Machines build models from individual weak learners in a sequential manner, each time focusing on the errors of the previous models to improve accuracy [19]. GBMs excel in handling different types of data, including categorical and continuous inputs, and are known for high predictive accuracy. They effectively manage overfitting through parameters like learning rate and number of trees. Their ability to model complex interaction effects between variables makes them suitable for biological data involving intricate gene expression relationships. XGBoost, a specific implementation of GBM, is particularly noted for its efficiency and performance [20].

Table 1 summarizes the expected training time and performance of the selected models [15]:

Table 1: Expected Model Performance

Model	Expected Train Time	Expected Performance
RF	Fast	Good
SVM	Moderate	Good
GBM	Slow	Best

2.4 Hyperparameter Optimization

Hyperparameter optimization is essential for enhancing the predictive performance of machine learning models by fine-tuning their configurations [21]. In this study, we used GridSearchCV for hyperparameter tuning of all ML models. GridSearchCV performs an exhaustive search over specified parameter grids and evaluates the performance using cross-validation [22]. For the Geneformer model, we used the provided hyperparameter optimization method as described in the original paper. Detailed information about the specific values tested and their rationale can be found in Appendix B.

2.5 Model Classification Performance Evaluation

To evaluate the classification performance of the Geneformer model and the selected ML methods in predicting the label of cancer cells, we employed several key metrics: accuracy [23], precision [24], recall [24], F1 Score [25], and AUC-ROC [26]. Additionally, we utilized the confusion matrix [27] to visualize the model performance. Appendix C contains for more information about the metrics.

The predictive performance was assessed on the test set, comparing the ability of each model to classify untreated versus treated cells.

2.6 Perturbation Algorithm

Geneformer employs an in-silico method for perturbing transcriptomes, allowing for the perturbation of one, two, or three genes simultaneously. These perturbations involve altering the rank value encoding of genes in a cell. Geneformer supports several types of perturbations: deletion (removing a gene from the rank value encoding), overexpression (moving a gene to the front), inhibition (shifting a gene to a lower quartile), and activation (shifting a gene to a higher quartile); all of which are described in detail in the original paper.

Our own perturbation algorithm was developed to allow for comparison of the selected ML methods to Geneformer. The algorithm is designed to perform the same types of in-silico perturbations as Geneformer, which ensures a fair comparison of the models' ability to predict the outcomes of perturbation combinations. We concentrated only on deletion and overexpression of genes, as these two perturbations are presumed to have the most notable effect on the cells [28].

The perturbation algorithm takes a list of genes and an action to be performed as input. The algorithm either deletes or overexpresses all genes in the list. When the action specified is 'delete', the algorithm sets the expression value of the gene to zero, effectively simulating the removal of the gene from the cell's transcriptome. This mimics a knockout experiment where the gene is completely inactive. When the action specified is 'overexpress', the algorithm sets the expression value of the gene to the maximum expression level observed in the dataset for that gene. This simulates overexpression, where the gene is expressed at abnormally high levels. This action is implemented by finding the maximum expression value for the gene across all samples and setting the gene's expression in the perturbed sample to this maximum value. Both actions are visualized in the Figure 2.

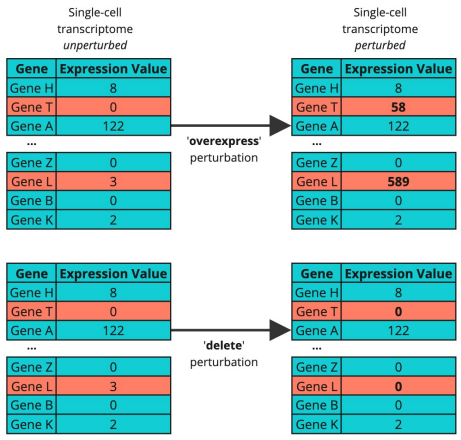

Figure 2: Example of perturbation combinations applied to the same cell for two genes. The left panel illustrates the overexpression perturbation. The right panel shows the deletion perturbation.

2.7 Differential Gene Expression

Differential Gene Expression (DGE) analysis identifies genes that are either significantly overexpressed or underexpressed between conditions in specific cell populations [29]. We used this method to identify the difference in the mean expression between treated and untreated cells for all individual genes in the sciplex2 dataset. They are referred to as highly differentially expressed (HDE) genes.

We utilized the Scanpy toolkit, applying t-test method to extract the top 500 HDE Single Genes.

Identification of Differentially Expressed Gene Pairs

To identify the most differentially expressed gene pairs, we generated all possible pairs from the top 200 HDE Single Genes identified by the t-test method. For each gene pair, we computed differential expression by calculating the absolute difference in the sum of expression values between treated and untreated cells. These pairs were then ranked by their differential expression values. The top 500 HDE Gene Pairs were selected for further analysis. The details about this procedure are examined in Appendix D.

2.8 Model Comparison for Gene Perturbation Combinations

The goal of this research is to compare the models by evaluating their responses to gene perturbation combinations using the sciplex2 dataset.

The Geneformer model offers methods to perform gene perturbations and evaluates their effects by reporting the "Shift_to_goal_end", which corresponds to the Cosine Shift in the embedding. Cosine Shift measures the change in the cosine similarity between the mean embeddings of cells before and after perturbation compared to the treated mean embedding. It is calculated as:

$$\text{Cosine Shift} = \cos(\theta_{\text{post}}) - \cos(\theta_{\text{pre}})$$

where $\cos(\theta_{\text{pre}})$ is the cosine similarity between the mean pre-perturbation embedding and the treated mean, and $\cos(\theta_{\text{post}})$ is the cosine similarity between the mean post-perturbation embedding and the treated mean.

The performance of each ML model can be assessed by calculating the percentage of cells that shifted from an untreated to a treated state after each perturbation. This enables the identification of the most efficient perturbation combinations for each ML model. The Shift Percentage is calculated using the following steps:

1. **Gene Perturbation**: Perturb genes based on predefined indices and actions (overexpression or deletion).
2. **Data Transformation**: Transform perturbed data using PCA.
3. **Prediction**: Use ML models to predict the treated state of the perturbed data.
4. **Shift Calculation**: Calculate the percentage shift from untreated to treated cells:

$$\text{Shift Percentage} = \left(\frac{\sum(y_{\text{unperturbed}} = 0 \wedge y_{\text{perturbed}} = 1)}{\sum(y_{\text{unperturbed}} = 0)} \right) \times 100$$

The performance of Geneformer and ML methods can be directly compared using several approaches. Firstly, we can use Cosine Shifts to observe the changes caused by a given set of perturbations in Geneformer's embedding space and the PCA-embedded space used to train the traditional ML methods. It is important to note that all selected ML models use the same PCA-embedded data, so the Cosine Shifts will be the same for each of the models. Secondly, we can compare Geneformer with each of the ML models using Cosine Shifts to identify the perturbation pairs that cause the largest change in embedding space for Geneformer. For each traditional ML model, the Shift Percentage metric can be used to identify the perturbation pairs that cause the highest shift of untreated cells being classified as treated. This allows us to compare the models directly based on the set of perturbation combinations they considered the most important. Lastly, we can compare the models based on the importance they place on the HDE genes. This involves reporting the percentage of the perturbations each model found to be the most important that come from either the top 500 HDE Single Genes or HDE Gene Pairs. Our hypothesis is that more primitive models will place high emphasis on the HDE genes as it only captures the difference in expression value between treated and untreated cells, ignoring the potential downstream effects that changing the expression of genes might cause. In contrast, the Geneformer model claims to understand the gene network dynamics and the downstream effects of perturbing genes. Therefore, we would expect that the Geneformer model will not have as many HDE genes among the top-ranking perturbations as the selected ML methods.

2.9 Experimental Setup

For our experiments, we utilized the DAIC cluster provided by TU Delft [30], ensuring sufficient computational resources for extensive model training and evaluation. We used fixed random seeds in all experiments to promote consistency and reproducibility of results, all can be found in Appendix L.

The Python library Scikit-Learn (Sklearn) [31] was employed for various tasks, including PCA embedding, implementing all machine learning models, and optimizing hyperparameters.

3 Results

This section presents the outcomes of our experiments in training and evaluating the performance of the selected machine learning models and Geneformer in predicting the response of cancer cells to different perturbation combinations. It encompasses the findings from the classification task, the analysis of selected perturbation combinations, and the comparative evaluation of model performance based on perturbation outcomes.

3.1 Classification Task

The first step in our model comparison involves fine-tuning the Geneformer model and training traditional ML models for a binary classification task, aiming to predict treatment labels of cells in the sciplex2 dataset. Followingly, hyperparameters of each model were optimized using the validation set.

Subsequently, we evaluated their performance on the test set. The performance metrics are detailed in Appendix F. Here is a summary of the results, ranked by Test Accuracy and Test F1 Score:

- **SVM:** Accuracy: 0.9126, F1 Score: 0.9143
- **GBM:** Accuracy: 0.9094, F1 Score: 0.9091
- **RF:** Accuracy: 0.8706, F1 Score: 0.8726
- **Geneformer:** Accuracy: 0.8544, F1 Score: 0.8530

3.2 Selected Perturbation Combinations

In the sciplex2 dataset, there are 35,653 different genes, leading to a vast number of possible perturbation pairs. The total number of possible gene pairs can be calculated using the formula for combinations:

$$\frac{n \cdot (n-1)}{2}$$

where n is the number of genes. For our dataset, this results in

$$\frac{35,653 \cdot 35,652}{2} = 635,550,378$$

gene pairs. Given that we perform two different perturbation types (overexpression and deletion), there are 1,271,100,756 different perturbations to consider. This number is too large to exhaustively test all perturbations within the constraints of our computational resources, necessitating a more selective approach.

Therefore, this paper focuses on a subset of the possible perturbations. Since selecting random perturbations might not yield the most insightful results, the focus was instead on identifying perturbations based on the correlation of gene expression values. In human cancer cell lines, expression-correlated genes often reveal molecular interaction networks, particularly in epithelial-like cancer cells [32]. Thus, we examined different types of gene pairs based on the correlation in their expression values.

We conducted an experiment where different perturbations were performed on all cells in the dataset, using a RF classifier to calculate the Shift Percentage for the following perturbation categories:

- **Random Gene Pair Perturbations:** 500 randomly selected gene pair perturbations.
- **Pair Perturbations with High Expression-Correlation:** 500 most expression-correlated gene pairs (correlation > 0.55).
- **Pair Perturbations with Low Expression-Correlation:** 500 least expression-correlated gene pairs (correlation < 0.000016).
- **Random Single Gene Perturbations:** 500 random single gene perturbations.
- **HDE Gene Pairs:** 500 genes pairs with the highest DGE.
- **HDE Single Genes:** 500 single genes with the highest DGE.

The results of these perturbations are illustrated in a box plot (Figure 3), which shows that the largest Shift Percentages occur for highly expression-correlated gene pairs.

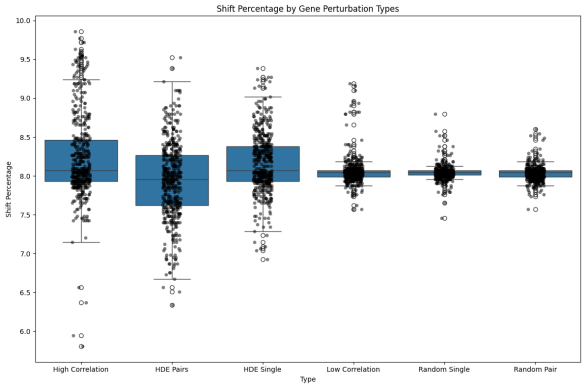

Figure 3: Shift Percentage by Gene Perturbation Types. The boxplot compares the Shift Percentages caused by different categories of gene perturbations.

To validate the hypothesis regarding the importance of high correlation in gene expression for gene perturbations, another experiment was performed using RF, where 100 random gene pairs from various correlation ranges were sampled and the resulting Shift Percentages were analyzed. The results indicated a clear trend: higher expression-correlation between gene pairs led to greater Shift Percentages in classification. This trend is visualized in Figures 4, which depict a steep increase in the shift for highly expression-correlated gene pairs, underscoring the strong relationship between the expression-correlation of gene pairs and the impact of the perturbation.

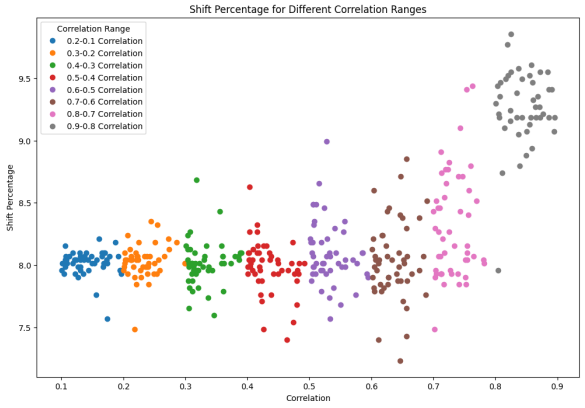

Figure 4: Shift Percentage for 100 randomly selected gene pairs for different correlation ranges between 0 and 0.9.

3.3 Model Comparison Based on Perturbation Combinations

An experiment was conducted to compare the performance of three selected ML methods and Geneformer in identifying perturbation combinations. The 2500 most expression-correlated gene pairs were selected, forming 5000 perturba-

tions (each gene pair creating one 'delete' and one 'overexpress' perturbation). Additionally, the top 250 Highly Differentially Expressed (HDE) Single Genes and the top 250 HDE Gene Pairs were included, forming 500 perturbations from each category. In total, 6000 unique perturbations were evaluated. Geneformer's built-in perturbation method was used to rank perturbations by the Cosine Shifts they caused in Geneformer's embedding space. Similarly, perturbations were ordered by the Cosine Shifts in the PCA embedding used for the sciplex2 data. The Shift Percentage of each perturbation was measured for each traditional ML model.

Cosine Shifts in PCA and Geneformer Embeddings

The comparison of Cosine Shifts caused by perturbations in the PCA and Geneformer embeddings is summarized in Table 2.

Statistic	Geneformer	PCA
Mean Shift	0.0012	0.0022
Median Shift	0.0005	0.0020
Max Shift	0.0824	0.0645

Table 2: Comparison of Cosine Shifts in Geneformer and PCA Embeddings.

The data reveals that the majority of Cosine Shifts for Geneformer were relatively small, as shown in Figure 5.

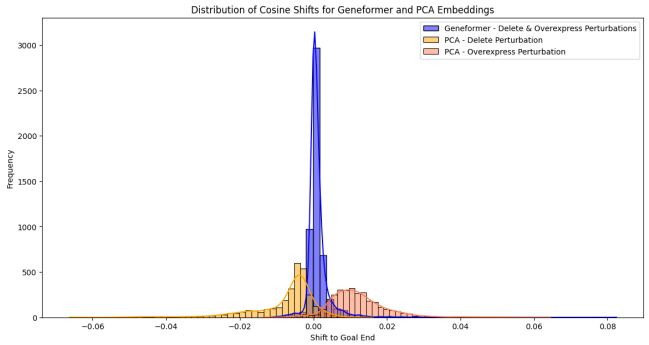

Figure 5: Distribution of Cosine Shifts for Geneformer and PCA Embeddings

In contrast, the PCA embedding resulted in a wider distribution of shifts, with larger median and mean shifts. However, Geneformer achieved a higher maximum Cosine Shift compared to PCA. Additionaly, PCA placed more importance on HDE gene perturbations. Among the top 100 Cosine Shifts, 84 originated from the most expression-correlated gene pairs for Geneformer, while PCA showed an almost equal split between highly expression-correlated gene and HDE perturbations, as illustrated in Figure 6.

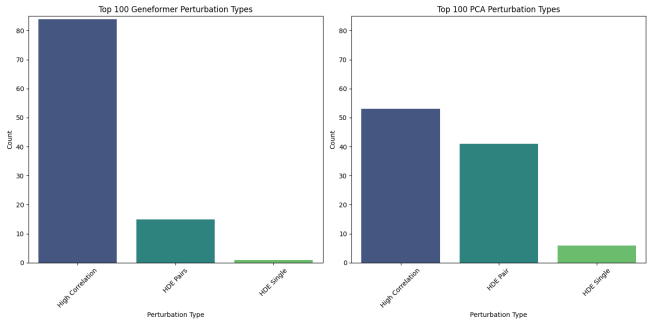

Figure 6: Top 100 Geneformer and PCA Perturbation Types

Perturbation Combinations in ML Models

Traditional ML methods were evaluated by their ability to identify perturbation combinations using Shift Percentage metric. The most significant shifts were found by GMB, followed by RF, and lastly SVM. GMB had the highest mean shift (8.0688) and maximum shift (9.8576), indicating its superior ability to identify significant perturbation combinations. RF showed a slightly lower mean shift (7.8155) and a maximum shift of 8.9361, while SVM had the lowest mean shift (6.7763) and maximum shift of 8.4613. More information can be found in Appendix I.

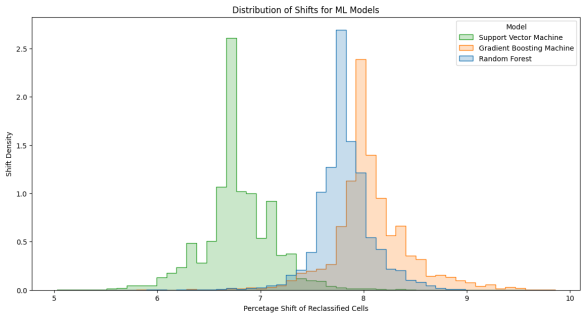

Figure 7: Distribution of Shift Percentages for traditional ML Models.

Perturbation Combinations in Geneformer and ML Models

Lastly, Geneformer was compared with the three selected ML models based on the top 100 perturbation combinations they considered most important. The models showed diverging behavior in assigning importance to the HDE genes. While all models found expression-correlated genes to be the most important, Geneformer placed little-to-no importance on HDE Single Genes, whereas traditional ML methods placed very little importance on HDE Gene Pairs as can be seen in Table 3.

	Geneformer	RF	GBM	SVM
HDE Pairs	0.15	0.00	0.01	0.01
HDE Single	0.01	0.12	0.15	0.13
High Correlation	0.84	0.88	0.84	0.86

Table 3: Importance of Perturbation Types for Geneformer and ML Models

The most significant difference between the models can be observed in the importance placed on the perturbation action type. Geneformer identified the vast majority of important perturbations as 'overexpress,' while traditional ML models predominantly highlighted 'delete' perturbations as seen in Figure 8.

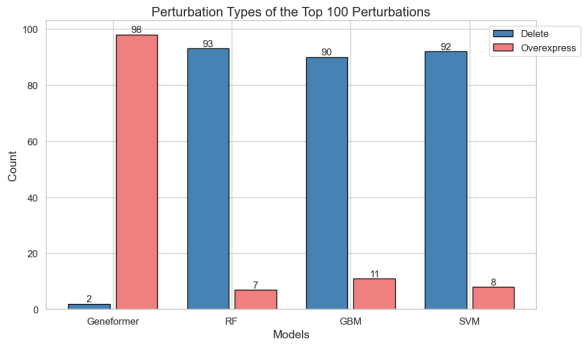

Figure 8: Counts of 'delete' and 'overexpress' perturbation types in the top 100 perturbations for each model.

Most Important Genes for Each Model

The top perturbations in Geneformer were dominated by high expression-correlated gene pairs. The most recurrent genes in these top pairs were:

- **KRT18** (Keratin 18)
- **KRT7** (Keratin 7)
- **SLC9A3R1**
- **RHOB** (Ras Homolog Family Member B)
- **GLIS3** (GLIS Family Zinc Finger 3)

The most frequently appearing genes for traditional ML models were:

- **MT-ND4** (NADH-ubiquinone oxidoreductase chain 4)
- **MT-RNR2** (Mitochondrially encoded 12S RNA)
- **MT-CO1**
- **MT-ATP6**
- **MT-ND5** (NADH-ubiquinone oxidoreductase chain 5)

Top 10 most important genes per model are reported in Appendix J.

4 Discussion

The results of our experiments provide valuable insights into the comparative performance of Geneformer and traditional machine learning models (GBM, SVM, and RF) on the sciplex2 dataset. Traditional ML methods outperformed Geneformer in the binary classification task of identifying the treated cell labels. This outcome was anticipated, given the efficacy of traditional ML models in well-defined classification tasks and the fact that Geneformer, pre-trained on a larger corpus of data, might introduce noise in such specific tasks. Among the models, SVM achieved the highest classification accuracy, followed by GBM, RF, and Geneformer, as detailed in our results section.

Our experiments revealed that highly expression-correlated gene pairs caused the largest shifts in classification across all models. This suggests that perturbations involving highly expression-correlated genes have a significant effect on cellular states, underlining their importance in predicting drug response. This finding is particularly valuable for further research on datasets where an exhaustive search for perturbations is not feasible. Identifying the most promising perturbation combinations based on gene correlation could optimize the search process, making it more efficient and effective.

Geneformer indicated a deeper understanding of gene network dynamics compared to traditional ML methods. It placed much less emphasis on HDE genes than PCA and achieved higher maximum Cosine Shifts. Unlike traditional ML methods, Geneformer focused more on HDE Gene Pairs rather than HDE Single Genes. This approach might reflect Geneformer's superior ability to understand the complex downstream effects of gene perturbation combinations, as supported by the findings of Theodoris et al. (2021) [7], which highlight the advantages of using complex, pre-trained models like Geneformer for biological predictions.

The most important genes identified by Geneformer and traditional ML methods differed significantly. Geneformer frequently identified genes involved in epithelial cell structure and function, such as Keratin 18 (KRT18) and Keratin 7 (KRT7), as well as genes involved in cellular signaling and regulation, such as RHOB and GLIS3. These genes are crucial in maintaining cellular integrity and signaling pathways, indicating Geneformer's capability to capture essential aspects of cellular dynamics.

In contrast, traditional ML methods highlighted mitochondrial genes as the most important, including MT-ND4, MT-RNR2, MT-CO1, MT-ATP6, and MT-ND5. These genes are involved in mitochondrial function and energy production, which are critical for cellular metabolism and survival. The focus on mitochondrial genes by traditional ML models might be due to their significant role in cellular energy balance, making them easily detectable in perturbation impact studies.

The observed trends align with the Geneformer paper, confirming the benefits of model complexity and pre-training in capturing intricate biological interactions. The higher shifts for highly expression-correlated gene pairs are consistent with the understanding that gene networks are interdependent and that perturbing one gene can have downstream effects on others.

5 Conclusions and Future Work

The primary research question addressed in this study was: "How does the predictive performance of the Geneformer model compare to traditional machine learning methods, specifically Random Forest, Support Vector Machines, and Gradient Boosting Machines, in predicting the response of cancer cells to perturbation combinations using the sciplex2 dataset?"

While Geneformer did not outperform the traditional models in classification accuracy, it demonstrated significant potential for understanding complex gene network dynamics. Our experiments showed that highly expression-correlated gene pairs caused the largest Percentage and Cosine Shifts across all models, indicating that perturbations involving these genes significantly influence cellular states. Geneformer's advanced architecture and extensive pre-training on single-cell transcriptomes allowed it to achieve higher maximum Cosine Shifts compared to PCA embeddings and placed less emphasis on highly differentially expressed (HDE) Single Genes, focusing instead on HDE Gene Pairs. This suggests that Geneformer has a superior capability in capturing the complex downstream effects of gene perturbations.

These findings are especially relevant for drug discovery and development. Accurate predictions of gene perturbation impacts are essential for identifying effective drug targets and combinations. Advanced models like Geneformer, which can better understand gene network dynamics, show potential for improving the accuracy of such predictions compared to traditional ML models.

Despite these promising results, our study had several limitations. Due to time and resource constraints, we did not perform an exhaustive search of all possible perturbation pairs. Instead, we selected and evaluated the 6000 most promising gene perturbations, which represent less than 0.001% of the possible 1,271,100,756 perturbations in the sciplex2 dataset. An exhaustive analysis would be necessary to draw conclusive results about Geneformer's ability to understand gene network dynamics and identify the most valuable perturbations. Additionally, our analysis was restricted to gene pairs. Evaluating all possible combinations of perturbations would provide a more comprehensive understanding of each model's ability to detect the downstream effects of different perturbations. However, this comprehensive analysis requires substantial computational resources, as the number of combinations increases exponentially with the number of genes. Furthermore, a bug in the Geneformer code related to in-silico perturbations caused a minor slow down in our research (see Appendix K for details).

To address these limitations, future work could explore several directions. Research could extend to perturbations involving three or more gene combinations, employing systematic approaches such as genetic algorithms to efficiently explore the solution space. Utilizing larger and more diverse datasets would validate the findings and improve the generalizability of the models. Applying the models to real-world drug discovery projects would evaluate their practical utility and potential for clinical applications.

In conclusion, this study demonstrates that while traditional ML models perform well in classification tasks, advanced models like Geneformer have a promising capability in predicting the impact of gene perturbations due to their ability to capture complex gene network dynamics. These findings underscore the potential for integrating advanced deep learning models in drug discovery and precision medicine, providing a pathway for more effective and targeted therapeutic interventions.

6 Responsible Research

Responsible research ensures integrity and applicability of scientific findings in the field of computational biology. This research adheres to ethical guidelines and reproducibility standards established by TU Delft.

6.1 Ethical Considerations

The research utilized publicly available dataset and did not involve any direct interactions with human subjects or use of personal data, mitigating the potential ethical concerns over privacy and consent. However, given the high risk topic of predicting cancer cell responses to perturbations, the findings could be applied in clinical settings and eventually influence treatment recommendations and clinical trials. The potential to inform drug development and treatment strategies carries significant ethical importance, highlighting the need for Reproducibility in this study, ensuring that this research contributes positively and ethically to patient care.

6.2 Reproducibility of Methods

To ensure the reproducibility of results, following measures were taken:

- **Transparent Methodology:** All experimental procedures, from data preprocessing and model training to evaluation metrics, are explicitly described in this paper. This includes detailing the hyperparameters used for each machine learning model and the criteria for data splitting and feature selection.
- **Open Source Tools and Frameworks:** All software and libraries used in the study are widely-recognized, open-source software tools, allowing for research replication. The specific versions and configurations used are provided in the Appendices.
- **Data Accessibility:** The sciplex2 dataset is publicly accessible, allowing other researchers to verify the findings and use the data for further research. Detailed descriptions of how the dataset was queried and manipulated are provided, ensuring that our work can be accurately reproduced.

6.3 Discussion on Reproducibility Challenges

Despite the best efforts to make this research reproducible, certain aspects still present challenges. Firstly, the stochastic nature of many machine learning algorithms can lead to variations in results when repeating the experiments. To address this, the research uses fixed seeds for random number generators and shares them in the Appendix L.

References

[1] Camilla Mattiuzzi and Giuseppe Lippi. Current cancer epidemiology. *J. Epidemiol. Glob. Health*, 9(4):217, 2019.

[2] Transcriptome - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Transcriptome. [Accessed 23-06-2024].

[3] Norbert Dojer, Anna Gambin, Andrzej Mizera, B. Wilczyński, and Jerzy Tiuryn. Applying dynamic bayesian networks to perturbed gene expression data. *BMC Bioinformatics*, 7:249 – 249, 2006.

[4] Random Forests - Machine Learning — link.springer.com. https://link.springer.com/article/10.1023/A:1010933404324. [Accessed 03-06-2024].

[5] Chen-Tsung Huang, C. Hsieh, Yun-Hsien Chung, Yen-Jen Oyang, Hsuan-Cheng Huang, and Hsueh-Fen Juan. Perturbational gene-expression signatures for combinatorial drug discovery. *iScience*, 15:291 – 306, 2019.

[6] Betül Güvenç Paltun, Samuel Kaski, and Hiroshi Mamitsuka. Machine learning approaches for drug combination therapies. *Briefings in Bioinformatics*, 22(6):bbab293, 08 2021.

[7] Theodoris et al. Transfer learning enables predictions in network biology. *Nature*, 680:616–624, 2023.

[8] Jos'e L. McFaline-Figueroa. Sample gsm4150377: Sciplex2 - a549 transcription modulators. National Center for Biotechnology Information, 2020. Accessed: 13 June 2024.

[9] Swati Srivastava, Shumaila Siddiqui, Samradhi Singh, Sangita Chowdhury, Vishal Upadhyay, Arppita Sethi, and Arun Kumar Trivedi. Dexamethasone induces cancer mitigation and irreversible senescence in lung cancer cells via damaging cortical actin and sustained hyperphosphorylation of prb. *Steroids*, 198:109269, 2023.

[10] X. Kostaras, F. Cusano, G. Kline, Wilson Roa, and Jacob C. Easaw. Use of dexamethasone in patients with high-grade glioma: a clinical practice guideline. *Current oncology*, 21 3:e493–503, 2014.

[11] Lei Xu, Hua Xia, Dongsheng Ni, Yanxia Hu, Jianing Liu, Y. Qin, Qin Zhou, Qiying Yi, and Yajun Xie. High-dose dexamethasone manipulates the tumor microenvironment and internal metabolic pathways in anti-tumor progression. *International Journal of Molecular Sciences*, 21, 2020.

[12] C. Ahlmann-Eltze and W. Huber. Comparison of transformations for single-cell rna-seq data. *Nature Methods*, 20:665 – 672, 2022.

[13] Sidharth Mishra, Uttam Sarkar, Subhash Taraphder, Sanjoy Datta, Devi Swain, Reshma Saikhom, Sasmita Panda, and Menalsh Laishram. Principal component analysis. *International Journal of Livestock Research*, page 1, 01 2017.

[14] Phauk Sokkhey and T. Okazaki. Hybrid machine learning algorithms for predicting academic performance. *International Journal of Advanced Computer Science and Applications*, 11, 2020.

[15] Chongsheng Zhang, Changchang Liu, Xiangliang Zhang, and G. Almpanidis. An up-to-date comparison of state-of-the-art classification algorithms. *Expert Syst. Appl.*, 82:128–150, 2017.

[16] Gene selection and classification of microarray data using random forest - BMC Bioinformatics — bmcbioinformatics.biomedcentral.com. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-3. [Accessed 03-06-2024].

[17] William Noble. What is a support vector machine? *Nature biotechnology*, 24:1565–7, 01 2007.

[18] Alain Rakotomamonjy, Isabelle Guyon, and André Elisseeff. Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3(7-8):1357–1370, 2003.

[19] Greedy function approximation: A gradient boosting machine. — doi.org. https://doi.org/10.1214/aos/1013203451. [Accessed 03-06-2024].

[20] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. pages 785–794, 08 2016.

[21] Philipp Probst, A. Boulesteix, and B. Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, 20:53:1–53:32, 2018.

[22] Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS — arxiv.org. https://arxiv.org/abs/1912.06059. [Accessed 03-06-2024].

[23] Accuracy and precision - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Accuracy_and_precision. [Accessed 23-06-2024].

[24] Precision and recall - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Precision_and_recall. [Accessed 23-06-2024].

[25] F-score - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/F-score. [Accessed 23-06-2024].

[26] Receiver operating characteristic - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Receiver_operating_characteristic. [Accessed 23-06-2024].

[27] Confusion matrix - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Confusion_matrix. [Accessed 23-06-2024].

[28] Katsunori Yoshikawa, T. Tanaka, Yoshihiro Ida, C. Furusawa, T. Hirasawa, and H. Shimizu. Comprehensive phenotypic analysis of single-gene deletion and overexpression strains of saccharomyces cerevisiae. *Yeast*, 28, 2011.

[29] Roy Francis. Differential gene expression — nbisweden.github.io. https://nbisweden.github.io/workshop-scRNAseq/labs/scanpy/scanpy_05_dge.html. [Accessed 03-06-2024].

[30] DAIC Documentation — daic.tudelft.nl. https://daic.tudelft.nl/. [Accessed 22-06-2024].

[31] scikit-learn: machine learning in Python &x2014; scikit-learn 1.5.0 documentation — scikit-learn.org. https://scikit-learn.org/stable/. [Accessed 23-06-2024].

[32] K. Kohn, Barry Zeeberg, W. Reinhold, and Y. Pommier. Gene expression correlations in human cancer cell lines define molecular interaction networks for epithelial phenotype. *PLoS ONE*, 9, 2014.

Appendix A: sciplex2 Information

Table A1: Gene Expression Statistics in the sciplex2 Dataset

Value	Statistic
Mean Expression	0.166518
Median Expression	0.000000
Standard Deviation	2.422994
Max Expression	5636.000000
Min Expression	0.000000
Total Genes	35653.000000
Total Cells	20964.000000
Genes Zero in All Cells	0.000000
Genes Non-Zero in All Cells	1.000000

Table A2: Summary Statistics of Observational Metadata for Dex Drug in the sciplex2 Dataset

Statistic	i	n.umi	Size_Factor	count genes expressed	ratio mt2non_mt
Count	7370.000000	7370.000000	7370.000000	7370.000000	7370.000000
Mean	11958.050882	5134.065672	1.271285	2428.673813	0.101970
Std	6929.145505	3613.007530	0.894645	1081.592091	0.055274
Min	1.000000	1000.000000	0.247618	508.000000	0.011734
25%	6167.250000	2820.000000	0.698282	1670.250000	0.065426
50%	11849.500000	4170.000000	1.032566	2230.000000	0.091272
75%	17574.750000	6299.750000	1.559929	2983.000000	0.125224
Max	24259.000000	65003.000000	16.095892	9360.000000	1.028486

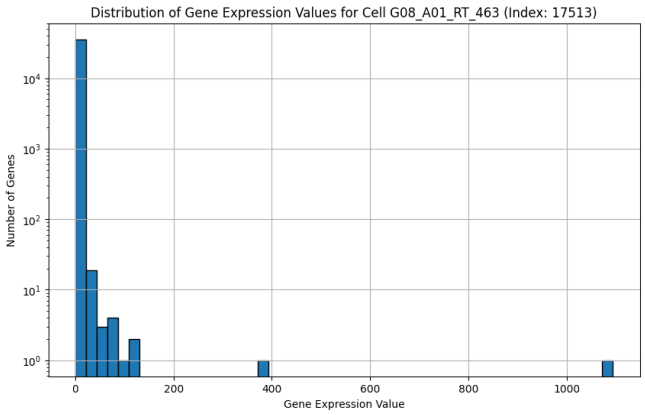

Figure A1: Gene expressions for three randomly sampled cells from the dataset (Example 1).

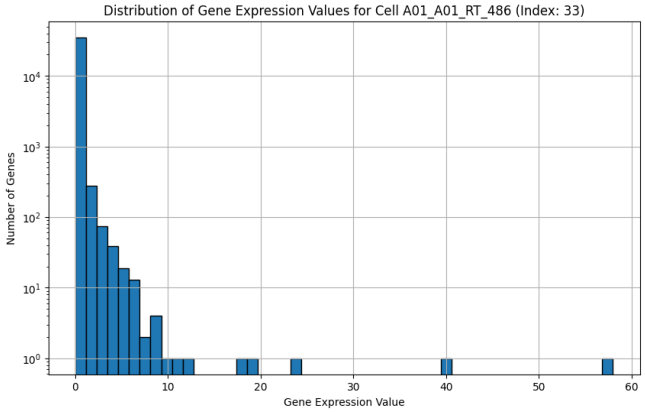

Figure A2: Gene expressions for three randomly sampled cells from the dataset (Example 2).

Figure A3: Gene expressions for three randomly sampled cells from the dataset (Example 3).

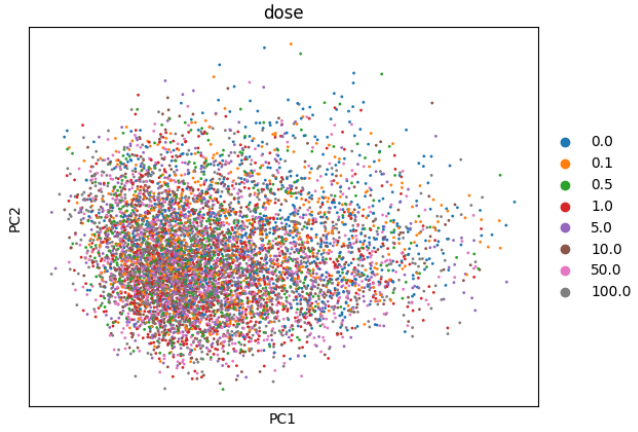

Figure A4: PCA plot showing the similarity of different Dex dosages based on gene expression data.

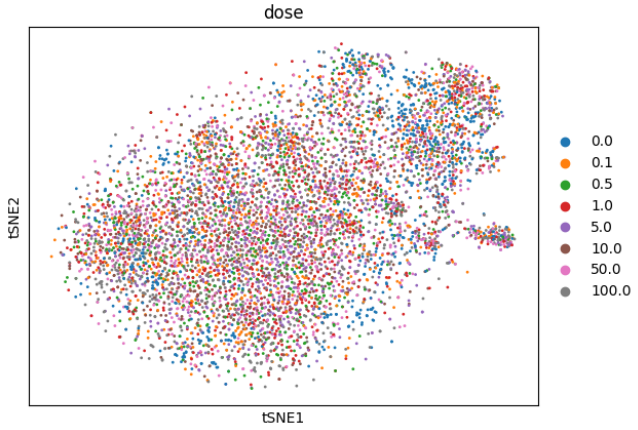

Figure A5: t-SNE plot illustrating the similarity of different Dex dosages based on gene expression data.

Appendix B: Hyperparameter Optimization

Random Forest

For the Random Forest model, the following hyperparameters were optimized: **n_estimators**, **max_depth**, **min_samples_split**, and **min_samples_leaf**.

- **n_estimators:** Tested values: [100, 200, 300].
- **max_depth:** Tested values: [None, 10, 20, 30].
- **min_samples_split:** Tested values: [2, 5, 10].
- **min_samples_leaf:** Tested values: [1, 2, 4].

Support Vector Machine

For the Support Vector Machine model, the following hyperparameters were optimized: **C**, **gamma**, and **kernel**.

Gradient Boosting Machine

- **C:** Tested values: [0.1, 1, 10, 100].
- **gamma:** Tested values: [1, 0.1, 0.01, 0.001].
- **kernel:** Tested values: ['linear', 'rbf'].

For the Gradient Boosting Machine model, the following hyperparameters were optimized: **n_estimators**, **learning_rate**, **max_depth**, **min_samples_split**, and **min_samples_leaf**.

- **n_estimators:** Tested values: [100, 200, 300].
- **learning_rate:** Tested values: [0.01, 0.1, 0.2].
- **max_depth:** Tested values: [3, 4, 5].
- **min_samples_split:** Tested values: [2, 5, 10].
- **min_samples_leaf:** Tested values: [1, 2, 4].

Performance Analysis

The performance of the RF model with and without hyperparameter optimization is compared in Table A3.

Metric	No Optimization	Optimized
Test Accuracy	0.8706	0.8738
Test Precision	0.8562	0.8616
Test Recall	0.8896	0.8896
Test F1 Score	0.8726	0.8754
Test AUC-ROC	0.9273	0.9383

Table A3: Performance metrics for Random Forest model with and without hyperparameter optimization.

Appendix C: Classification Evaluation Metrics

Confusion Matrix: A Confusion Matrix is a table used to evaluate the performance of a classification model by displaying the actual versus predicted classifications. It comprises four components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The matrix allows for the calculation of various performance metrics and provides insight into the types of classification errors the model is making.

Accuracy: Accuracy is a metric that measures the ratio of correctly predicted instances to the total instances in the dataset. It is calculated as the number of true positives and true negatives divided by the total number of instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy provides a straightforward measure of how often the classifier is correct.

Precision: Precision, also known as Positive Predictive Value, reflects the ratio of true positive predictions to the sum of true positive and false positive predictions. It is a measure of the accuracy of the positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP}$$

High precision indicates that the classifier has a low false positive rate.

Recall: Recall [24], also known as Sensitivity or True Positive Rate, indicates the ratio of true positive predictions to the sum of true positive and false negative predictions. It measures the ability of the classifier to identify all relevant instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

High recall indicates that the classifier has a low false negative rate.

F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is especially useful when the class distribution is imbalanced.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

This score ranges from 0 to 1, where a score of 1 indicates perfect precision and recall.

AUC-ROC: The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) represents the ability of the classifier to distinguish between classes. The ROC curve is a plot of the true positive rate (recall) against the false positive rate (1-specificity) at various threshold settings. The AUC is the area under this curve, with values ranging from 0.5 (no discrimination) to 1 (perfect discrimination). A higher AUC indicates better model performance.

Appendix D: HDE Gene Pairs Validation

To validate our approach for selecting HDE Gene Pairs, we compared the Shift Percentage caused by perturbing the top differentially expressed single genes and gene pairs. HDE Single Genes caused shifts proportional to their differential expression, while HDE Gene Pairs induced more nuanced shifts, highlighting possibly more complex gene interactions as can be seen in Figures A6 and A7.

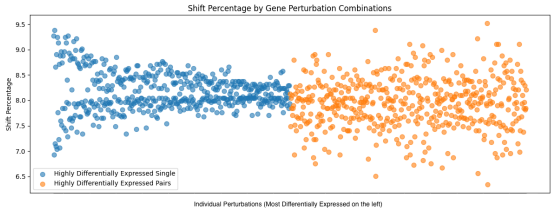

Figure A6: Each point represents the Shift Percentage caused by perturbing a specific single gene (blue) or gene pair (orange).

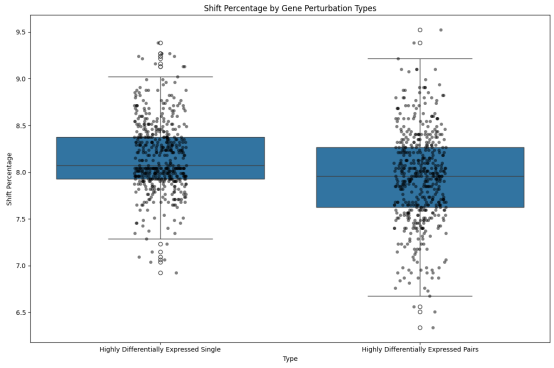

Figure A7: The boxplot displays the distribution of Shift Percentages for single gene and gene pair perturbations. Black dots represent perturbations.

Appendix E: Effects of PCA

To evaluate the impact of dimensionality reduction on ML methods, we compared the performance of Random Forest models trained on both original high-dimensional data with 35,653 features and data reduced to 256 dimensions using PCA. While the RF model trained on non-embedded data achieved marginally better classification performance, the PCA-embedded model significantly reduced training and hyperparameter optimization times.

Metric	No-Embedding	PCA
Test Accuracy	0.89	0.87
Test Precision	0.85	0.86
Test Recall	0.94	0.89
Test F1 Score	0.90	0.88
Test AUC-ROC	0.95	0.94
Embedding Time (s)	0	8
Training Time (s)	3	1
Hyperpar. time (s)	213	57

Table A4: Performance metrics for Random Forest models on non-embedded and PCA-embedded data.

In terms of permutation importance, the non-embedded model showed many genes with similar importance scores, whereas the PCA-embedded model highlighted a few genes with significantly higher importance, indicating enhanced interpretability. Figures A8 and A9 illustrate the permutation importance for both models.

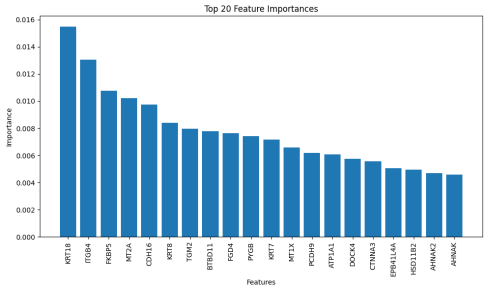

Figure A8: Permutation importance for Random Forest on non-embedded data.

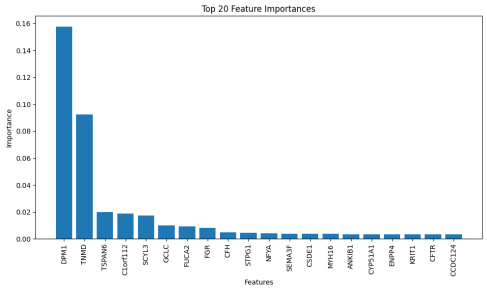

Figure A9: Permutation importance for Random Forest on PCA-embedded data.

Appendix F: Classification Performance

Gradient Boosting Machine (GBM)

- **Test Accuracy:** 0.9094
- **Test Precision:** 0.9091
- **Test Recall:** 0.9091
- **Test F1 Score:** 0.9091
- **Test AUC-ROC:** 0.9617

	Precision	Recall	F1 Score
Untreated	0.91	0.91	0.91
Treated	0.91	0.91	0.91
Accuracy		0.91	
Macro avg	0.91	0.91	0.91
Weighted avg	0.91	0.91	0.91

Table A5: Performance metrics for GBM.

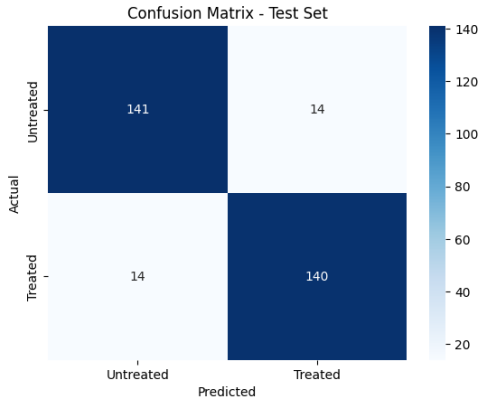

Figure A10: Confusion matrix for GBM.

Support Vector Machine (SVM)

- **Test Accuracy:** 0.9126
- **Test Precision:** 0.8944
- **Test Recall:** 0.9351
- **Test F1 Score:** 0.9143
- **Test AUC-ROC:** 0.9585

	Precision	Recall	F1 Score
Untreated	0.93	0.89	0.91
Treated	0.89	0.94	0.91
Accuracy		0.91	
Macro avg	0.91	0.91	0.91
Weighted avg	0.91	0.91	0.91

Table A6: Performance metrics for SVM.

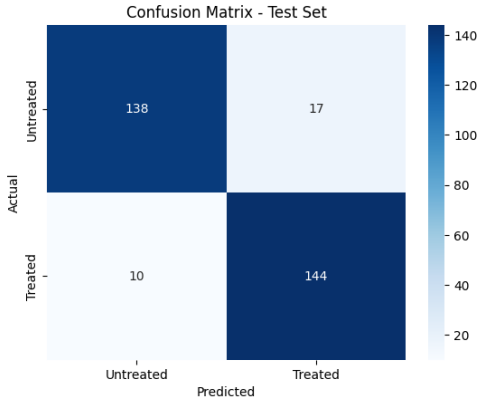

Figure A11: Confusion matrix for SVM.

Random Forest (RF)

- **Test Accuracy:** 0.8706
- **Test Precision:** 0.8562
- **Test Recall:** 0.8896
- **Test F1 Score:** 0.8726
- **Test AUC-ROC:** 0.9273

	Precision	Recall	F1 Score
Untreated	0.89	0.85	0.87
Treated	0.86	0.89	0.87
Accuracy		0.87	
Macro avg	0.87	0.87	0.87
Weighted avg	0.87	0.87	0.87

Table A7: Performance metrics for RF.

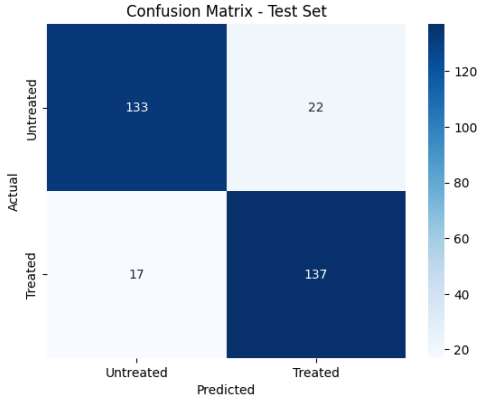

Figure A12: Confusion matrix for RF.

Geneformer

- **Test Accuracy:** 0.8544
- **Test Precision:** 0.8250
- **Test Recall:** 0.8307
- **Test F1 Score:** 0.8530
- **Test AUC-ROC:** 0.9237

	Precision	Recall	F1 Score
Untreated	0.83	0.83	0.83
Treated	0.83	0.83	0.83
Accuracy		0.83	
Macro avg	0.83	0.83	0.83
Weighted avg	0.83	0.83	0.83

Table A8: Performance metrics for Geneformer.

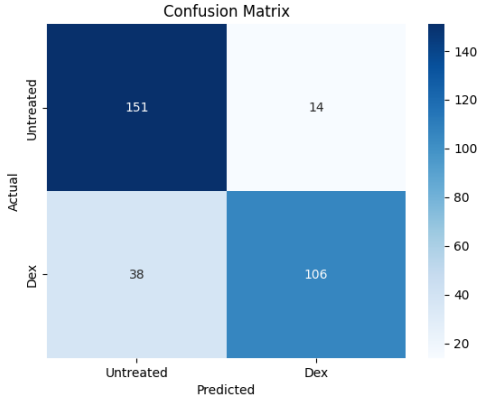

Figure A13: Confusion matrix for Geneformer.

Appendix G: Selected Perturbations

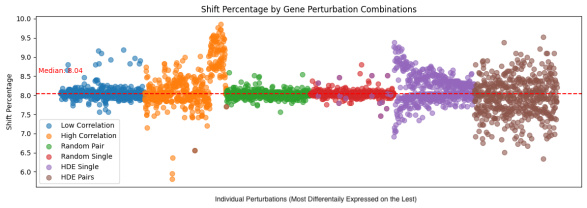

Figure A14: The plot shows the Shift Percentages caused by different gene perturbation types, with highly correlated gene pairs causing the largest shifts. The red dashed line indicates the median Shift Percentage for single gene perturbations.

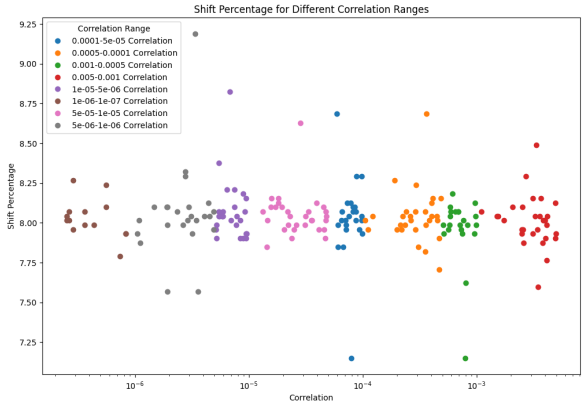

Figure A15: The plot shows the Shift Percentages for gene pairs with low correlations.

Appendix H: PCA vs Geneformer

Geneformer Cosine Shifts Statistical Data:

- Mean Shift: 0.0011603424179148444
- Median Shift: 0.00054156949039175
- Standard Deviation: 0.0033833955364798857
- Min Shift: -0.0114042460668756
- Max Shift: 0.0823748580189731

PCA Cosine Shifts Statistical Data:

- Mean Shift: 0.0022156844903766125
- Median Shift: 0.0020517334375
- Standard Deviation: 0.013044398082042132
- Min Shift: -0.06628146
- Max Shift: 0.06449759

Appendix I: ML Models Comparison

Model	Type	Count	Mean	Std	Min	25%	50%	75%	Max
GBM	HDE Pairs	500.0	7.923	0.447	6.507	7.651	7.959	8.217	9.383
GBM	HDE Single	500.0	8.189	0.515	6.925	7.819	8.056	8.601	9.383
GBM	High Correlation	5000.0	8.070	0.368	5.808	7.903	8.015	8.238	9.858
SVM	HDE Pairs	500.0	6.580	0.456	5.306	6.248	6.590	6.953	7.763
SVM	HDE Single	500.0	6.817	0.441	5.585	6.479	6.716	7.149	7.959
SVM	High Correlation	5000.0	6.782	0.338	5.027	6.646	6.758	6.953	8.461
RF	HDE Pairs	500.0	7.637	0.326	6.618	7.449	7.651	7.875	8.378
RF	HDE Single	500.0	7.886	0.341	6.814	7.624	7.819	8.126	8.936
RF	High Correlation	5000.0	7.820	0.261	5.976	7.679	7.791	7.931	8.936

Table A9: Comparison of ML models by their Shift Percentages based on perturbation types.

Model	Count	Mean	Std	Min	25%	50%	75%	Max
Gradient Boosting Machine	6000	8.0688	0.3790	5.8084	7.9028	8.0145	8.2379	9.8576
Support Vector Machine	6000	6.7763	0.3488	5.0265	6.6183	6.7579	6.9813	8.4613
Random Forest	6000	7.8155	0.2687	5.9760	7.6794	7.7911	7.9307	8.9361

Table A10: Statistical Summary of Shift Percentages for ML Models. GMB showed the highest mean shift, followed by RF and SVM, indicating its superior ability to identify significant perturbation combinations.

Comparison of Perturbation Types

The following figures present the box plots comparing the percentage shifts for different perturbation types for each of the ML models. These plots illustrate that highly correlated gene pairs generally caused the largest shifts across all models.

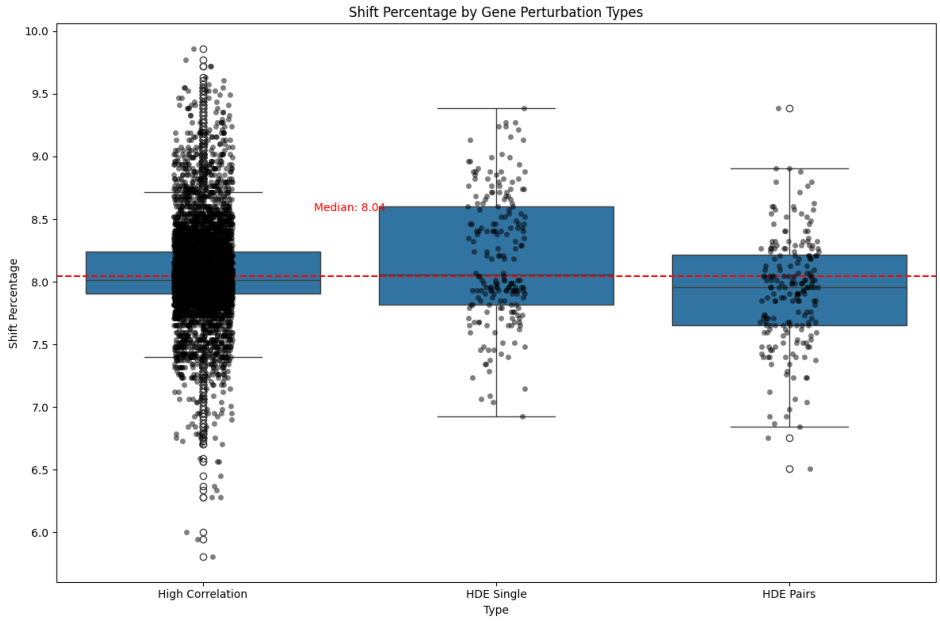

Figure A16: Boxplot of Shift Percentages for Different Perturbation Types - GBM.

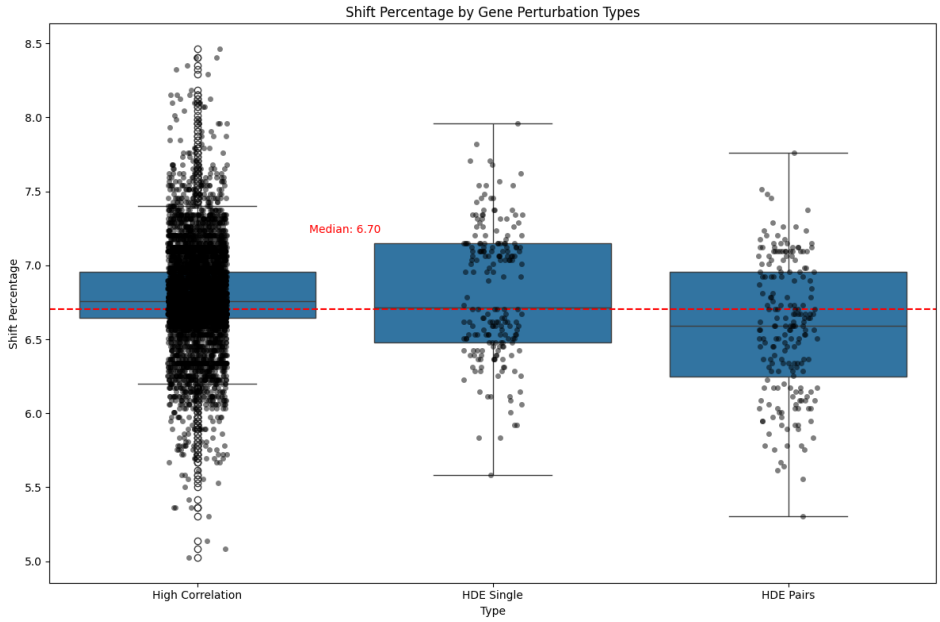

Figure A17: Boxplot of Shift Percentages for Different Perturbation Types - SVM.

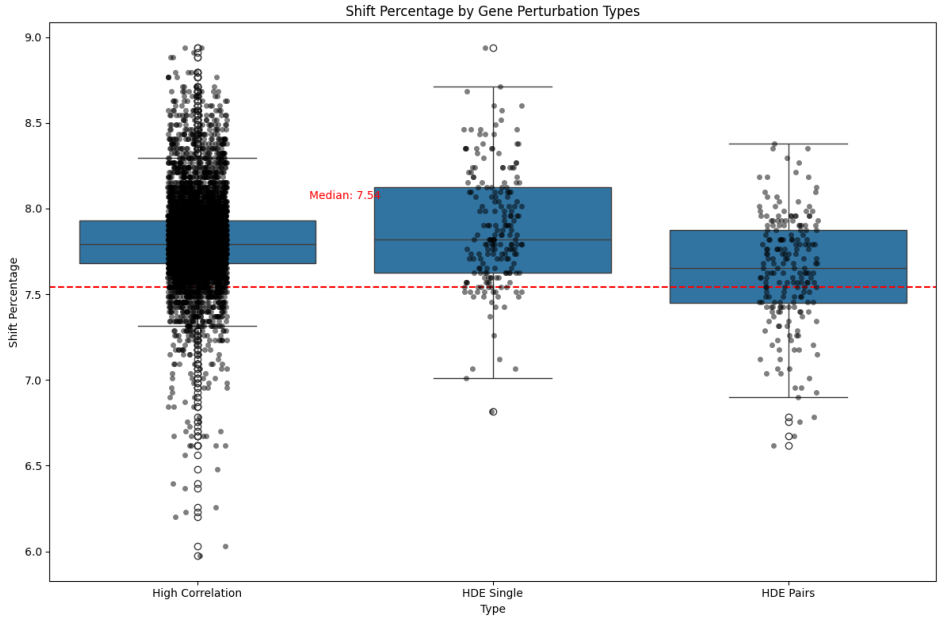

Figure A18: Boxplot of Shift Percentages for Different Perturbation Types - RF.

Appendix J: Best Performing Perturbation Combinations

The following tables list the top 10 perturbation combinations per model, their corresponding Shift Percentages or Cosine Shifts, and the type of perturbation.

Gene Perturbation	Shift Percentage	Type
MT-ND4 (Delete) + MT-RNR2 (Delete)	9.8576	High Correlation
MT-CO1 (Delete) + MT-ND4 (Delete)	9.7738	High Correlation
MT-ND4 (Delete) + MT-RNR1 (Delete)	9.7180	High Correlation
MT-ND4 (Delete) + MT-ATP6 (Delete)	9.7180	High Correlation
MT-CO1 (Delete) + MT-RNR2 (Delete)	9.6342	High Correlation
MT-ATP6 (Delete) + MT-RNR2 (Delete)	9.6063	High Correlation
MT-CYB (Delete) + MT-RNR2 (Delete)	9.5783	High Correlation
FTL (Delete) + MT-RNR2 (Delete)	9.5504	High Correlation
MT-CYB (Delete) + MT-ATP6 (Delete)	9.5504	High Correlation
MT-ND5 (Delete) + MT-ND4 (Delete)	9.5504	High Correlation

Table A11: Top Gene Combination Shift Percentages and Types for GBM

Gene Perturbation	Shift Percentage	Type
MT-ND4 (Delete) + MT-RNR2 (Delete)	8.4613	High Correlation
MT-ND4 (Delete) + MT-ATP6 (Delete)	8.4055	High Correlation
MT-ND5 (Delete) + MT-ND4 (Delete)	8.4055	High Correlation
MT-CO1 (Delete) + MT-ND4 (Delete)	8.3496	High Correlation
MT-ND2 (Delete) + MT-ND4 (Delete)	8.3217	High Correlation
MT-CYB (Delete) + MT-ND4 (Delete)	8.2938	High Correlation
MT-ND5 (Delete) + MT-RNR2 (Delete)	8.1821	High Correlation
MT-ND4 (Delete) + MT-RNR1 (Delete)	8.1541	High Correlation
MT-ND5 (Delete) + MT-CO1 (Delete)	8.1541	High Correlation
MT-ND4 (Delete) + MT-CO3 (Delete)	8.1541	High Correlation

Table A12: Top Gene Combination Shift Percentages and Types for SVM

Gene Perturbation	Shift Percentage	Type
KYNU (Delete) + MT-RNR2 (Delete)	8.9361	High Correlation
MT-ND4 (Delete) + MT-RNR2 (Delete)	8.9361	High Correlation
FKBP5 (Overexpress)	8.9361	HDE Single
MT-CYB (Delete) + MT-RNR2 (Delete)	8.9081	High Correlation
CNTN1 (Delete) + MT-RNR2 (Delete)	8.8802	High Correlation
MT-CO1 (Delete) + MT-RNR2 (Delete)	8.8802	High Correlation
MT-RNR2 (Delete) + MT-RNR1 (Delete)	8.7964	High Correlation
PDE10A (Delete) + KYNU (Delete)	8.7964	High Correlation
MT-ND4 (Delete) + MT-RNR1 (Delete)	8.7964	High Correlation
MT-ND5 (Delete) + MT-ND4 (Delete)	8.7964	High Correlation

Table A13: Top Gene Combination Shift Percentages and Types for RF

Gene Perturbation	Cosine Shift	Type
KRT18 (overexpress) + KRT8 (overexpress)	0.0824	High Correlation
KRT18 (overexpress) + S100A10 (overexpress)	0.0496	High Correlation
KRT18 (overexpress) + KRT7 (overexpress)	0.0460	High Correlation
SLC9A3R1 (overexpress) + RHOB (overexpress)	0.0433	High Correlation
KRT18 (overexpress) + CTNNA3 (overexpress)	0.0414	High Correlation
ALDOC (overexpress) + RHOB (overexpress)	0.0372	High Correlation
KRT18 (overexpress) + AKR1C2 (overexpress)	0.0342	High Correlation
DHRS2 (overexpress) + SLC9A3R1 (overexpress)	0.0288	High Correlation
SLC9A3R1 (overexpress) + CKB (overexpress)	0.0284	High Correlation
THSD4 (overexpress) + KRT7 (overexpress)	0.0280	HDE Pairs

Table A14: Top Gene Combination Cosine Shifts and Types for Geneformer

Appendix K: Bug in Geneformer Code

During the process of executing in-silico perturbations using the Geneformer model, we encountered a bug that caused a failure in the perturbation method. The following section details the identification, cause, and resolution of this bug.

Issue Description

The bug manifested when invoking Geneformer's in-silico perturbation method with specific parameters for perturbing gene pairs. The code snippet where the error occurred is shown below:

```
for idx, row in gene_pairs.iterrows():
    gene_1 = row['gene_1_id']
    perturb_type_1 = row['perturb_type_1']
    gene_2 = row['gene_2_id']
    perturb_type_2 = row['perturb_type_2']
    isp = InSilicoPerturber(
        perturb_type=perturb_type_1,
        perturb_rank_shift=None,
        genes_to_perturb=[gene_1, gene_2],
        combos=1,
        anchor_gene=None,
        model_type="CellClassifier",
        num_classes=2,
        emb_mode="cell",
        cell_emb_style="mean_pool",
        cell_states_to_model=cell_states_to_model,
        state_embs_dict=state_embs_dict,
        emb_layer=0,
        forward_batch_size=100,
        nproc=1
    )
```

The error produced was:

```
ValueError: Length of values (2) does not match length of index (1)
```

Cause of the Bug

The issue was found in the `isp_stats_to_goal_state` function in `in_silico_perturber_stats.py`. It occurred due to a mismatch between the values' length (2) and the index length (1) when creating a DataFrame, causing a `ValueError`.

Fixing the Bug

The bug was fixed by matching the values' length to the index length. The loop was restructured to aggregate results into a list of dictionaries before converting it to a DataFrame, ensuring consistent entry lengths.

Adjusted Geneformer Code

Below is the critical part of the corrected `isp_stats_to_goal_state` function in the `in_silico_perturber_stats.py` file:

```
results_list = []
for i in trange(cos_sims_df.shape[0]):
    token = cos_sims_df["Gene"][i]
    name = cos_sims_df["Gene_name"][i]
    ensembl_id = cos_sims_df["Ensembl_ID"][i]
    goal_end_cos_sim_megalist = result_dict[cell_states_to_model["goal_state"]]
                                            .get((token, "cell_emb"), [])
    mean_goal_end = np.mean(goal_end_cos_sim_megalist)
    pval_goal_end = ranksums(goal_end_random_megalist, goal_end_cos_sim_megalist).pvalue
    results_dict = {
        "Gene": token,
        "Gene_name": name,
        "Ensembl_ID": ensembl_id,
        "Shift_to_goal_end": mean_goal_end,
        "Goal_end_vs_random_pval": pval_goal_end
    }
    results_list.append(results_dict)
cos_sims_full_df = pd.DataFrame(results_list)
```

Appendix L: Reproducibility Details

In this section, we provide details about the seeds used for random functions to ensure reproducibility in our research. Consistently, we used seed 42 where possible, following the precedent set by the Geneformer paper authors.

Dataset Pre-processing

Downsampling Untreated Cells

During the dataset pre-processing stage, we downsampled the untreated cells in the data used for training the ML models and fine-tuning the Geneformer. We used a fixed seed of 42 to ensure the same distribution can be achieved each time when reproducing the experiments.

Data Splitting

When splitting the data into training (70%), validation (15%), and test (15%) sets, we used 'random_state=42' to make the splits reproducible. This was applied both during the initial split into training and temporary sets, and the subsequent split into validation and test sets.

ML Methods Code

Model Training

For training each of the ML models, we used the 'random_state' parameter to ensure reproducibility of the models' training. Specifically, we set 'random_state=42'.

Permutation Importance

When identifying permutation importance for each model, we used 'random_state=42' to ensure the reproducibility of the permutation importance calculations.

Geneformer

Geneformer Classifier

In the Geneformer code, we used seed 42 when creating the Geneformer classifier. This includes setting the seed in the training arguments to ensure consistent and reproducible training of the classifier.

Geneformer Training Process

The number of epochs was set to 10 when training Geneformer for the classification task.