ML Challenge Final Project
CSCI 5461
Dr. Chad Meyers
5 May 2023

Team Members:
Evelyn Junaid (junai015)
Claire Bradley (bradl517)
Yueting Zhao (zhao1455)
Gus Shriver (shriv073)
Jinwei Zhou (zhou1909)

Abstract

As more interest has been placed onto investigating interactions between proteins, developing techniques to better predict how proteins interact has become even more important. Since actually performing assays to analyze protein interactions is costly and time consuming, computational prediction could be a way to avoid these challenges. We developed code for two computational prediction problems. Both are based on 200 test knockout genes interacting with 17,000 genes. The first problem was predicting gene GO terms based on a gene's 1x200 interaction profile. The second was also based on its interaction profile, this time predicting a gene's query profile. The method used for both problems was a neural network, which resulted in a Kaggle score of 0.52538 for the first and 0.01391 for the second. Having a positive score indicates that our code performed better than random in both problems, with the second one performing worse.

Introduction

Investigating the function of genes is important not only for the understanding of basic biology but also the discovery of the intricate relationship between genotype and phenotype. The applications of this information are varied, such as finding novel targets for cancer therapy and uncovering previously unidentified proteins involved in different cellular processes. The advances in CRISPR-Cas9 genome editing technology over the past decade have made it possible to efficiently screen the human genome and associate each gene with a phenotype. For example, by perturbing a single gene in a cell by site-specific mutations and comparing the resulting phenotype to that of a wild-type cell, we can better understand how the gene functions.

One application of these genome-wide screens is the identification of gene-gene interactions. As we know, biological systems are highly interconnected over many levels. Genes, or rather the proteins they encode, can interact in many ways; they may physically interact to form a functional protein complex, chemically modify each other in a signaling pathway, or just generally be involved in some similar cellular process such as sugar metabolism or cell adhesion. Gene-gene interactions can be analyzed through the generation of double-mutant organisms and measuring the resulting fitness of the cell. However, this process

is very time-consuming and costly; the human genome is made up of close to 25,000 genes, and the generation of double mutants for each combination of genes would be an impossible task with today's technology. Fortunately, through supervised machine learning, we can attempt to predict genetic interaction profiles, and the biological processes the genes are involved in, through computational methods.

This project aimed to use supervised machine learning approaches to investigate human genetic interaction profiles, composed of 181 genome-wide CRISPR-Cas9 screens across 17,804 single-knockout human cell lines. In challenge A, we used a classification approach to predict the Gene Ontology (GO) biological process annotations of 5,000 genes from the single-knockout human cell lines based on each gene's interaction profile. In challenge B, we used a regression approach to predict the query genetic interaction profile of 17,804 query genes based on the known library genetic interaction profile of 181 library genes across the ~17,804 single-knockout human cell lines. For both challenges, we used neural network and random forest approaches to make the respective predictions.

Methods

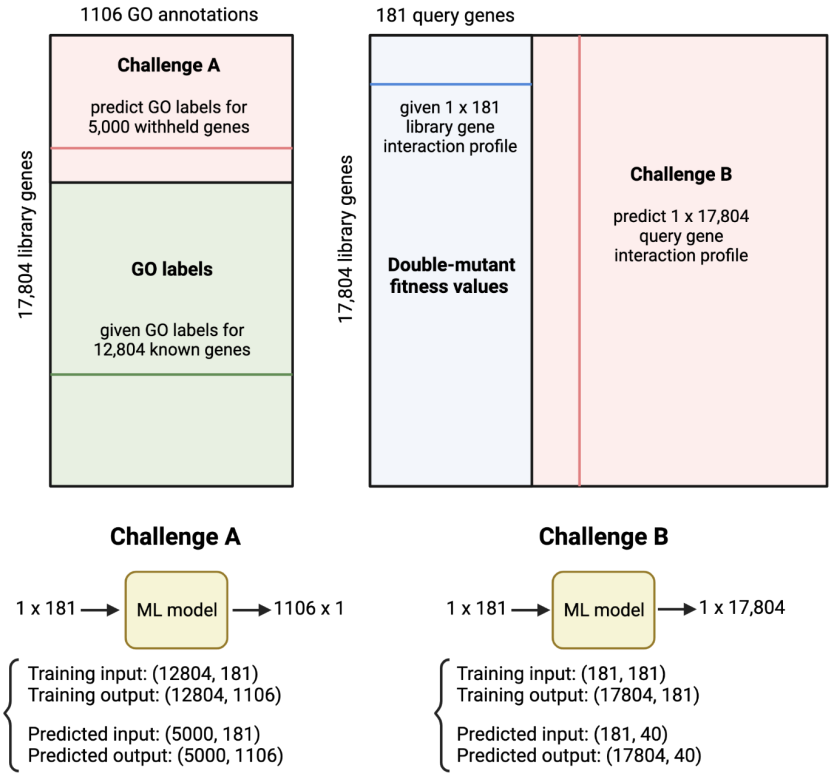

Figure 1. Workflow of ML challenges A and B. For challenge A, we trained on the 1 x 181 library gene interaction profile for the 12,804 genes with given GO labels in order to test GO label predictions on the 5,000 withheld genes. For challenge B, we trained on the 1 x 181 library gene interaction profile for the 181 query genes in order to test 1 x 17,804 query gene interaction profile predictions on 40 genes.

Challenge A

Classification

1. Neural Network

   The first 5000 genes were excluded from the gene interaction network and the remaining genes were used to train a Multi-Layer Perceptron (MLP) that can predict GO annotations, given the interaction profile of a gene. While training the MLP, the input data is fed forward through the network, with the output of each layer being the input to the next layer. During training the weights of the network are optimized so that the output of the MLP matches the target labels. The optimization is done using back propagation by computing the gradient of the error with respect to the weights and updating the weights to reduce the error. After the MLP is trained it can be used to predict the class labels of new data by passing the input through the network and obtaining the output of the final layer. The class with the highest probability is then chosen as the predicted class label. The solver algorithm used to optimize the weights of the network was Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), which is a quasi-Newton method that approximates the second-order optimization. This approximation allows for more efficient and accurate updates to the model weights than first-order approximations. A logistic activation function was used since this is a binary classification task. Four hidden layers with 400, 300, 200, and 400 neurons respectively, were used. A small hidden layer size was used to aid in the prevention of overfitting, improving the models generalization performance. In general, more complex and nonlinear problems require more hidden layers to capture the underlying patterns and relationships in the data. So, four layers may be a suitable choice. The model was able to train for a maximum of 2000 iterations, allowing the model to have a sufficient amount of time to reach a better solution.
   After the MLP was trained, the first 5000 genes' GO terms were predicted. A MinMaxScaler was used in order to ensure the predicted GO annotation values all fell between 0 and 1.

2. Random Forest

   Random Forest is an ensemble learning method that combines multiple decision trees to improve the accuracy of the model. It works by constructing a large number of decision trees and aggregating their predictions through a voting mechanism. Each decision tree is trained on a random subset of the input data and at each node of the tree a random subset of features is considered for splitting. To train the model, 100 decision trees were used in the random forest(100 is a common default value). The random seed was set to ensure reproducibility of results. Due to the size of the dataset, the processing time to train the model was inefficient and we were unable to fully train the model using this approach.

Challenge B

For challenge B, both models were trained using 181 gene interaction profiles as library genes and as query genes.

Regression

1. Neural Network

   A Multi-Layer Perceptron regression model was trained for challenge B. The optimization algorithm used was stochastic gradient descent (SGD) since it is suitable for large amounts of data. SGD works by updating the weights and biases of the neural network using a randomly selected subset of the training data at each iteration, rather than the entire dataset, allowing the algorithm to converge faster and use less memory than traditional gradient descent. At each iteration, the gradient of the loss function with respect to the weight and biases is computed using the randomly selected subset. Weights and Biases are updated using this gradient in addition to a learning rate which determines the step size for the update. Rectified Linear Unit (ReLU) was used as the activation function. ReLU is not only computationally efficient but also allows the network to learn non-linear relationships in data, making it a suitable choice for this challenge. Four hidden layers with 1000, 500, 500, and 500 neurons respectively, were used. A larger number of neurons were used for this model due to the nature of the training data. The training data has 181 features while the output matrix has 17,804 samples, indicating that the model needs to be able to capture a large number of patterns and relationships in the data. The tolerance for conversion was set to 1e-4, meaning that when the change in the loss of function between two consecutive iterations is less than or equal to 1e-4, the algorithm will declare it has converged to an optimal solution and stop running. If the loss of function between two iterations is smaller than 1e-4, it is unlikely that further iterations will result in significant improvement to the model's performance.

2. Random Forest

   A Random Forest regression model uses an ensemble of decision trees to make predictions about data. The model is trained by randomly selecting a subset of the training data and a subset of the features to create a decision tree at each iteration, ultimately creating a large number of decision trees. After this, the model can be used to make predictions on new data by running the new data through each decision tree to produce a final prediction. The regressor was created with 5 decision trees with ranges 120, 200, 300, 500, and 800. The maximum depth of these trees were chosen to be 5, 8, 15, and 25 respectively. The max features value was set to the square root of the total number of features for each decision tree for several reasons. It helps to reduce the correlation between trees in the forest, making the model more robust to noise in the dataset. It also aids in reducing overfitting of the model by forcing the focus to be on the most important features, improving the overall performance of the model. The next step

is to set up a GridSearchCV object which performs a search over the previously specified parameter grid to find the best combination of hyperparameters of the model. Two folds were chosen for cross-validation.

Once both of these models were trained, 40 query genes were run through the model to have their query profiles generated.

Results

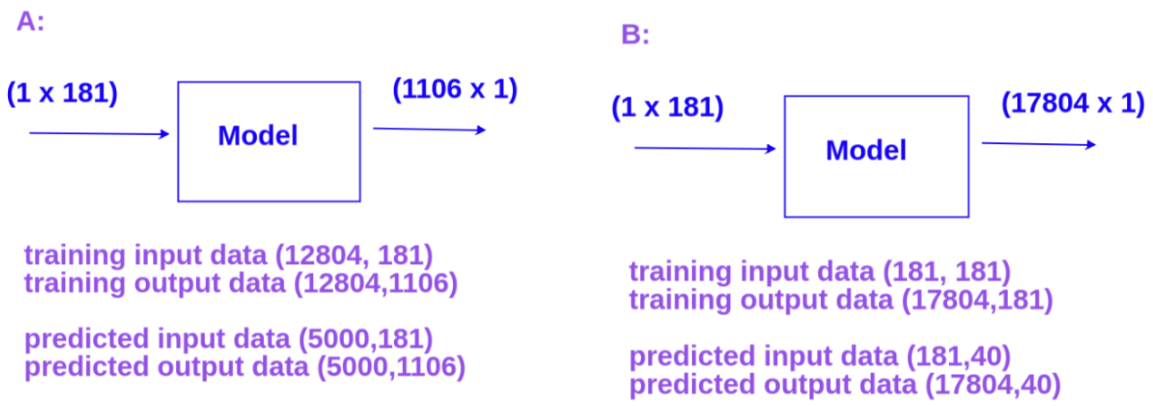

Figure 2. The flowchart of training. A) The input and output training data and predicted result for challenge A. B) The input and output training data and predicted result for challenge B.

The output predicted results were generated from trained ML models with input training data for both challenges (Figure 2). For challenge A, we got a [5000 x 1106] matrix of predicted results exhibiting the possibilities between the 1106 GO terms and 5000 genes. The range of scores is between 0~1, representing the possibility of if the expressed gene fits the GO annotation. For challenge B, we got a [17804 x 40] matrix of predicted results exhibiting the interactions between library and query genes. The range of scores is between -1~1. A negative score means a negative influence of the cell growth with the gene interaction, positive score means a positive influence of the cell growth with the gene interaction, and a 0 means no effect.

We tried both neural network and random forest models for challenge A and B. We then used Kaggle with totally random numbers to measure our models' predictions. The average Spearman correlation (from -1 to 1) on Kaggle serves as a simple reference of the performance. For both challenges, the prediction from the neural network approach showed a higher score. The scores were 0.52538 and 0.01391, for challenge A and B, respectively, indicating a nonrandom positive correlation between our prediction and the real data. The scores based on Kaggle score <0, indicating a negative correlation between our prediction and the real data. Random Forest was lower or even being negative, indicating that it is not a good machine learning model used in this condition for regression. Thus we would focus on the predictions from the neural network for following analysis.

For challenge A, we sorted the average possibility of the expressed genes fitting each GO annotation from ascending and descending orders. The GO annotations with the highest and lowest 5 possibilities are shown by Figure 3. For instance, GO.0006486 has the highest average possibility of the expressed genes fitting this Go annotation. The overall possibilities for all GO annotations are around 0.48-0.49.

For challenge B, we sorted the average interaction scores of genes from ascending and descending orders. The highest and lowest 5 scores and corresponding query genes are shown by Figure 4. For instance, gene 55 has the highest average positive interaction score, indicating that gene 55 usually positively influence cell growth during the interaction with library genes

In addition, we generated heat maps based on prediction results with the dendrogram of hierarchical clustering based on average linkage and Euclidean distance. Challenge A result (Figure 5) shows a clear strip pattern indicating clear clustering across GO annotations, while challenge B results (Figure 6) presents local clustering of several gene interactions.

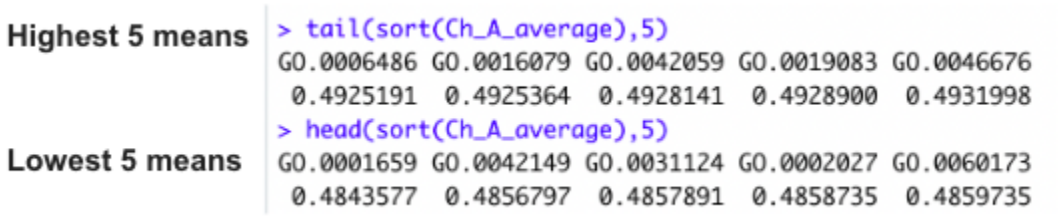

Figure 3. The GO annotations with the five highest and five lowest average probabilities of association with the genes tested in by Challenge A Neural Network model.

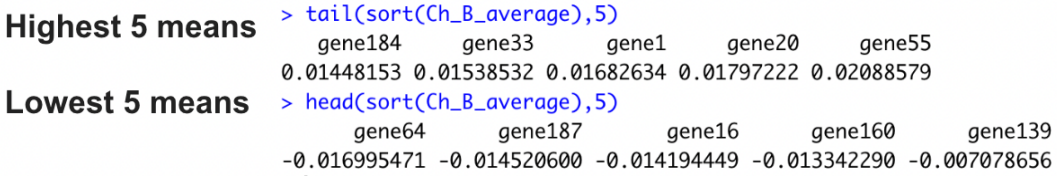

Figure 4. The five highest and five lowest average interaction scores with the query genes tested in by Challenge B Neural Network model.

In addition, we generated heat maps based on prediction results with the dendrogram of hierarchical clustering based on average linkage and Euclidean distance. Challenge A result (Figure 5) shows a clear strip pattern indicating clear clustering across GO annotations, while challenge B results (Figure 6) presents local clustering of several gene interactions.

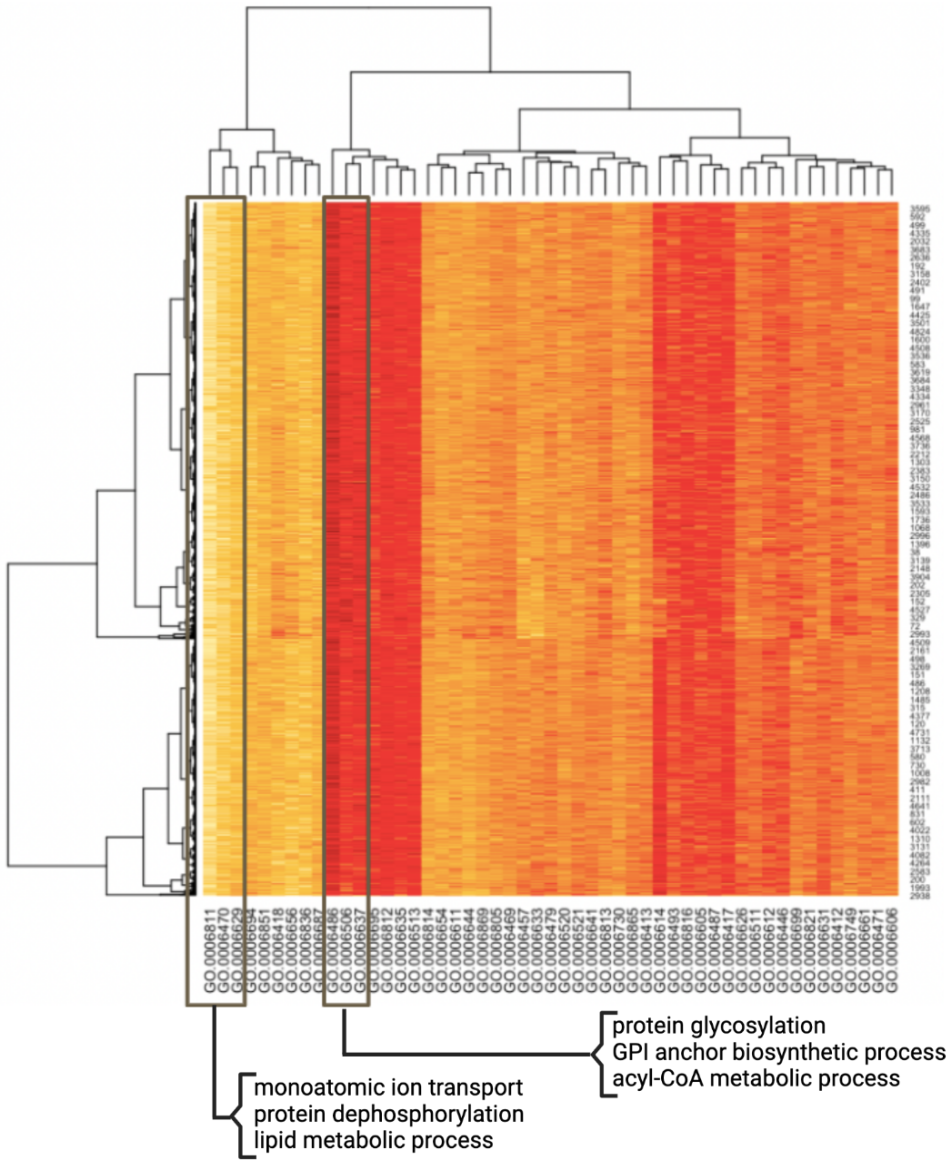

Figure 5. Representative dendrogram of hierarchical clustering based on average linkage and Euclidean distance, with GO labels on the x-axis and genes on the y-axis. A red color indicates a higher probability of the GO term associated with the gene, and a yellow color indicates a lower probability of the GO term associated with the gene. The two boxes indicate GO labels with strong low (left box) and high (right) probabilities for association across all genes.

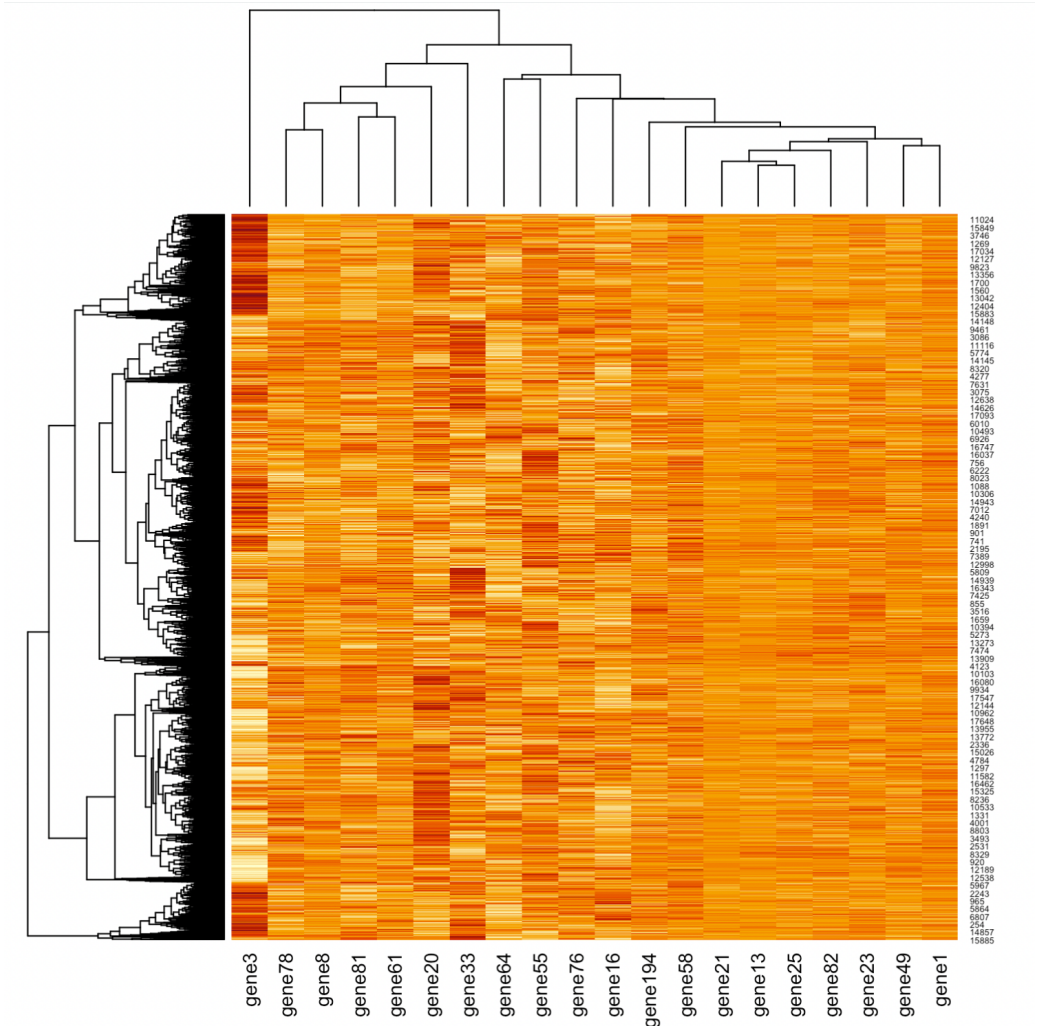

Figure 6. Representative dendrogram of hierarchical clustering based on average linkage and Euclidean distance, with query genes on the x-axis and library genes on the y-axis. A red color indicates a more positive interaction between library and query genes, and a yellow color indicates a more negative interaction between library and query genes. Global clusters were exhibited on first several query genes, such as gene 3.

Discussion

During the exploration of ML models, we mainly explored random forest and neural networks. We finally chose a neural network for its high efficiency and high accuracy. We also attempted to use kNN, but it would take hours or even days to run, so we didn't dig deeper with this model. Neural network model had a better performance in both classification (challenge A) and regression (challenge B) conditions may due to its flexibility of parameters such as the various activation functions and hidden layers. Thus we could try out a better combination of the parameters to improve the performance within this model. Based on the neural network model, we then analyzed the biological findings with predicted results.

For challenge A, the highest average probability of the expressed genes being associated with a GO term was protein glycosylation (GO.0006486) with an average probability of 0.4925191. This GO term, which reflects the process of removing an acetyl group from an amino acid, has 12,564 total annotations. The lowest average probability of the expressed genes being associated with a GO term was temperature homeostasis (GO.0001659) with an average probability of 0.4843577. This GO term, which reflects the homeostatic process in which an organism modulates its internal body temperature, has 266 total annotations.

We found a similar trend of GO annotation predictions across all 5,000 genes in the test set. After clustering, we found GO labels with either a strong high or low probability of being associated with all genes. In particular, protein glycosylation (GO:0006486), GPI anchor biosynthetic processes (GO:0006506), and acyl-CoA metabolic processes (GO:0006637) were shown to have the highest probabilities of association across all genes. Protein glycosylation is a process of modifying proteins that results in the addition of a carbohydrate to an amino acid, such as adding glycan chains to proteins. GPI anchor biosynthetic processes involve pathways leading to formation of glycosylphosphatidylinositol (GPI) anchors that attach membrane proteins to the lipid bilayer of the plasma membrane. Lastly, acyl-CoA metabolic processes involve pathways involving the derivative of coenzyme A in which the sulfhydryl group is in a thioester linkage with an acyl group. There are 21,172 total annotations for these three GO terms.

Conversely, monoatomic ion transport (GO:0006811), protein dephosphorylation (GO:0006470), and lipid metabolic processes (GO:0006629) were shown to have the lowest probabilities of association across all genes. Monoatomic ion transport includes the movement of an ion between cells or across some intracellular membrane through a transporter or pore. Protein dephosphorylation is the process of removing one or more phosphate groups from a protein residue. Lipid metabolic processes are pathways involving lipids, which are compounds that are soluble in an organic solvent like alcohol but insoluble in an aqueous solvent like water. There are 209,145 total annotations for these three GO terms, which is significantly more than the three GO terms showing a higher probability of association with the tested genes.

For challenge B, we found that the neural network method performed better than a random prediction, indicating that our results have some amount of biological significance. Since the Kaggle score was very low, our method would not be effective in producing highly accurate predictions of gene interactions. The local clustering on challenge B heatmap indicates the similarity between genes which can be used to predict their roles on interaction within the clusters. However, the interpretation may be limited by the separated local clusters, which may be due to the prediction results which didn't reflect reality accurately. The score being so low could be for several reasons. Our method could be ineffective for this type of prediction question. Another possibility is that it's basically impossible to produce an accurate prediction based on the given date. Looking at different machine learning approaches such as SVM to go about this problem could result in improved accuracy, and could also show that such a prediction is not feasible.

References

1. Geneontology. http://geneontology.org/
2. Kaggle. https://www.kaggle.com/competitions/csci-5461-s23-human-gene-function-prediction-B