Project ID: CELL-415

Making Sense of Missense: Machine Learning Training On a Model for Missense Variants Data

Margaux Vasilescu, Bronx High School of Science

December 16, 2022

1 Introduction

In recent years, research scientists have been using computer machine learning programs to analyze missense variants in genetic data. Missense variants are a type of genetic variation where a single nucleotide change in the DNA sequence results in a change in the amino acid sequence of the protein that is produced. This can affect the protein's function and potentially lead to disease. Recently, science researchers at Columbia University have created a machine learning program called gMVP which is much more efficient than other computer programs or machine learning programs that analyze missense variants. (Zhang and et al ((2022))). gMVP is a supervised machine learning method for predicting functionally damaging missense variants. The functional consequence of missense variants depends on both the type of amino acid substitution and its protein context. gMVP uses a graph attention neural network to learn the representation of protein sequence and structure context and context- dependent impact of amino acid substitutions on protein function. The currently trained gMVP model that is highly efficient was trained on the following data (called "Original Data"):1) Collected likely pathogenic and benign missense variants from curated databases (HGMD, ClinVar, and UniProt) as training positives and negatives, respectively, excluding the variants with conflicting evidence in the databases: 2) to balance positive and negative sets, randomly selected rare missense variants observed in human population sequencing data DiscovEHR as additional negatives for training. (Zhang and et al ((2022))). Theoretically, other missense variants-related data that trains gMVP could make the gMVP model even more efficient.

As the gMVP paper itself points out (see Zhang and et al ((2022)) at 15), recently, a machine learning language model called Transformer has been applied on protein sequences and multi-sequence alignments (MSAs) to improve the performance of coevolution strength estimation and protein residue-residue contacts prediction. (see Rao and et al. ((2020)) Rao and et al. ((2021))) Created in 2017, Transformer is a type of neural network architecture that was introduced in 2017 by a team of researchers at Google. (see Vaswani and et al. ((2017))) It is a type of deep learning model that is capable of performing a wide range of natural language processing tasks, such as language translation, text summarization, and question answering. Unlike many other models, which process language one word at a time, the Transformer uses a technique called self-attention, which allows it to consider the entire input sequence simultaneously. This makes it much faster and more efficient than other models, and it has been shown to be very effective in many natural language processing tasks. The way scientists used Transformer for MSAs is they had the model interleave rows and columns attentions across the input sequences and is trained with a variant of the masked language modeling objective across many protein

families. This program called MSA Transformer surpasses current state-of-the-art unsupervised structure learning methods by a wide margin, with far greater parameter efficiency than prior state-of-the-art protein language models. (see Rao and et al. ((2020)) Rao and et al. ((2021)))

Using the output from the MSA Transformer to train the gMVP model could make the model be even more efficient than the Original Data that trained the gMVP model.(see Zhang and et al ((2022)) at 15) My research involved changing the gMVP machine learning program, by editing the program's Python language and adding more programming language so that gMVP could be trained by MSA Transformer data. I then compared the efficiency of the gMVP model trained with the MSA Transformer data with the gMVP model trained by the Original Data to see which model is more effective. My research shows that the MSA Transfer data makes the gMVP Model slightly more effective.

2 Background

a. Missense Varients

Missense variants are a type of genetic variation where a single nucleotide change in the DNA sequence results in a change in the amino acid sequence of the protein that is produced. This can affect the protein's function and potentially lead to disease. For example, a missense variant in the gene for a specific enzyme may result in the enzyme being less effective or not functioning at all, which can disrupt the biological processes that the enzyme is involved in and lead to a disease state. Missense variants can potentially lead to a wide range of diseases and conditions, depending on the specific protein that is affected and the severity of the change in amino acid sequence. Some examples of diseases and conditions that can be caused by missense variants include cystic fibrosis, sickle cell anemia, and certain types of cancer. (see Boettcher et al. ((2019)), Huang and et al ((2018)), Jin and et al. ((2017)), Satterstrom and et al., Kaplanis and et al. ((2019))) In some cases, missense variants may only have a minor effect on protein function and may not cause any noticeable health problems. In other cases, missense variants can be more severe and can lead to serious health problems. (see Boettcher et al. ((2019)), Huang and et al ((2018)), Jin and et al. ((2017)), Satterstrom and et al., Kaplanis and et al. ((2019)))

Some examples of conditions that can be caused by missense variants include:

- Cystic fibrosis: A missense variant in the gene that encodes the cystic fibrosis transmembrane conductance regulator (CFTR) protein can lead to the development of cystic fibrosis, a genetic disorder that affects the respiratory, digestive, and reproductive systems.

- Hemochromatosis: A missense variant in the HFE gene, which is involved in the regulation of iron metabolism, can cause hemochromatosis, a condition in which the body absorbs and stores too much iron.
- Sickle cell anemia: A missense variant in the HBB gene, which encodes the beta chain of hemoglobin, can result in the production of abnormal hemoglobin, leading to sickle cell anemia, a blood disorder in which red blood cells become stiff and shaped like crescents.
- Alzheimer's disease: Missense variants in the APP, PSEN1, and PSEN2 genes, which are involved in the production of amyloid beta, a protein that forms plaques in the brain, have been linked to an increased risk of developing Alzheimer's disease, a neurodegenerative disorder that affects memory and cognitive function.

(Boettcher et al. ((2019)), Huang and et al ((2018)), Jin and et al. ((2017)) Satterstrom and et al., Kaplanis and et al. ((2019))). These are just a few examples of the many different diseases and conditions that can be caused by missense variants.

b. Computer Programs and Machine Learning For DNA and Genetic Data Research.

In recent years, as computer machine learning and other computer programs have become popular amount science researchers, DNA and Genetic researchers have used various types of computer programs and machine learning for their research.

For example, since 2013, researchers have used Polyphen2. Polyphen2 is a computational tool used to predict the effects of mutations on proteins. It uses a combination of algorithms and manually curated data to predict whether a particular mutation is likely to have a damaging effect on the structure and function of a protein. Polyphen2 is often used in the field of genetics to help interpret the results of genetic tests, and to identify potential disease-causing mutations. It is also used in the study of evolutionary biology, to help understand how changes in protein sequences can lead to the development of new traits and functions. (see Adzhubei and et al. ((2013))).

Similarly, SIFT (Sorting Intolerant From Tolerant) has also been used since 2013. SIFT is a computational tool used to predict the functional effects of amino acid substitutions on proteins. It uses a combination of evolutionary conservation and structural information to determine whether a given amino acid substitution is likely to have a damaging effect on protein function. (See Carter and et al. ((2013))).

Since 2014, CADD has been used by researchers. CADD (Combined Annotation-Dependent Depletion) is a computational tool used to predict the functional effects of genetic variants on proteins. It uses a combination of machine learning algorithms and manually curated data to score the likely impact of a given variant on protein function. (see Kircher and et al. ((2014))).

Since 2016, REVEL has been used. Revel is a statistical method used in DNA research to

assess the accuracy of predicted amino acid sequences. It uses a probabilistic model to evaluate the compatibility of a given sequence with a multiple sequence alignment, taking into account the evolutionary relationships among the sequences in the alignment. (See Ioannidis and et al. ((2016))).

Also since 2016, researchers have employed M-CAP. M-CAP uses a gradient boosting tree classifier17, which learns a function of the input features as a linear combination of decision trees, each derived iteratively to correct previously misclassified elements. The model hyper-parameters were optimized with a systematic grid search (Online Methods). The authors of M-CAP argued in their paper that M-CAP did a better job analyzing." These widely used methods misclassify 26 to 38 percent of known pathogenic mutations, which could lead to missed diagnoses if the classifiers are trusted as definitive in a clinical setting. We developed M-CAP, a clinical pathogenicity classifier that outperforms existing methods at all thresholds and correctly dismisses 60 percent of rare, missense variants of uncertain significance in a typical genome at 95 precent sensitivity." See Jagadeesh and et al. ((2016))).

And there are several other tools that have even more recently been developed to analyze DNA and Genetics concerning missense variants, such as: Eigen (Ionita-Laza and et al. ((2016))), MVP (Qi and et al. ((2021))), PrimateAI (Sundaram ((2018))), MPC (Samocha and et al. ((2017))), and CCRs (Havrilla and et al. ((2019))).

These methods differ in several aspects, including the prediction features, how the features are represented in the model, the training data sets and how the model is trained. Sequence conservation or local protein structural properties are the main prediction features for early computational methods such as GERP20 and PolyPhen2. MPC and CCRs estimate sub-genic coding constraints from large human population sequencing data which provide additional information not captured by previous methods. PrimateAI learns protein context from sequences and local structural properties using deep representation learning. A number of studies have reported evidence that functionally damaging missense variants are clustered in 3-dimensional protein structures21-23.

c. gMVP Machine Learning.

Recently, in 2021, science researchers at Columbia University created a new machine learning program to better analyze missense variants.Zhang and et al ((2022)) This program, called gMVP (short for "generalized Minimal Variant Pools"), is a computational tool used in DNA research to identify common sequence motifs among a set of related sequences. It uses a probabilistic model to identify the most likely set of motifs that are shared among the sequences, and can be used to help identify functional elements within DNA sequences. gMVP can be useful

in a variety of DNA research applications, including the identification of regulatory elements and the study of evolutionary relationships among different DNA sequences. Zhang and et al ((2022)). gMVP works by analyzing the data in a multi-dimensional graph. gMVP does this because some studies have reported evidence that functionally damaging missense variants are clustered in 3-dimensional protein structures. (see Iqbal and et al. ((2020)), Hicks and et al. ((2019)), Sively and et al. ((2018))).

d. New Data, MSA Transformer, To Train the gMVP Model

As the gMVP paper itself points out (see Zhang and et al ((2022))), recently, a machine learning language model called Transformer has been applied on protein sequences and multi-sequence alignments (MSAs) to improve the performance of coevolution strength estimation and protein residue-residue contacts prediction. (see Rao and et al. ((2020)), Rives and et al. ((2021)), Rao and et al. ((2021))) Created in 2017, Transformer is a type of neural network architecture that was introduced in 2017 by a team of researchers at Google.(Vaswani and et al. ((2017))). It is a type of deep learning model that is capable of performing a wide range of natural language processing tasks, such as language translation, text summarization, and question answering. Unlike many other models, which process language one word at a time, the Transformer uses a technique called self-attention, which allows it to consider the entire input sequence simultaneously. This makes it much faster and more efficient than other models, and it has been shown to be very effective in many natural language processing tasks. The way scientists used Transformer for MSAs is they had the model interleave rows and columns attentions across the input sequences and is trained with a variant of the masked language modeling objective across many protein families. This program called MSA Transformer surpasses current state-of-the-art unsupervised structure learning methods by a wide margin, with far greater parameter efficiency than prior state-of-the-art protein language models. (see Rao and et al. ((2020)), Rives and et al. ((2021)), Rao and et al. ((2021)))

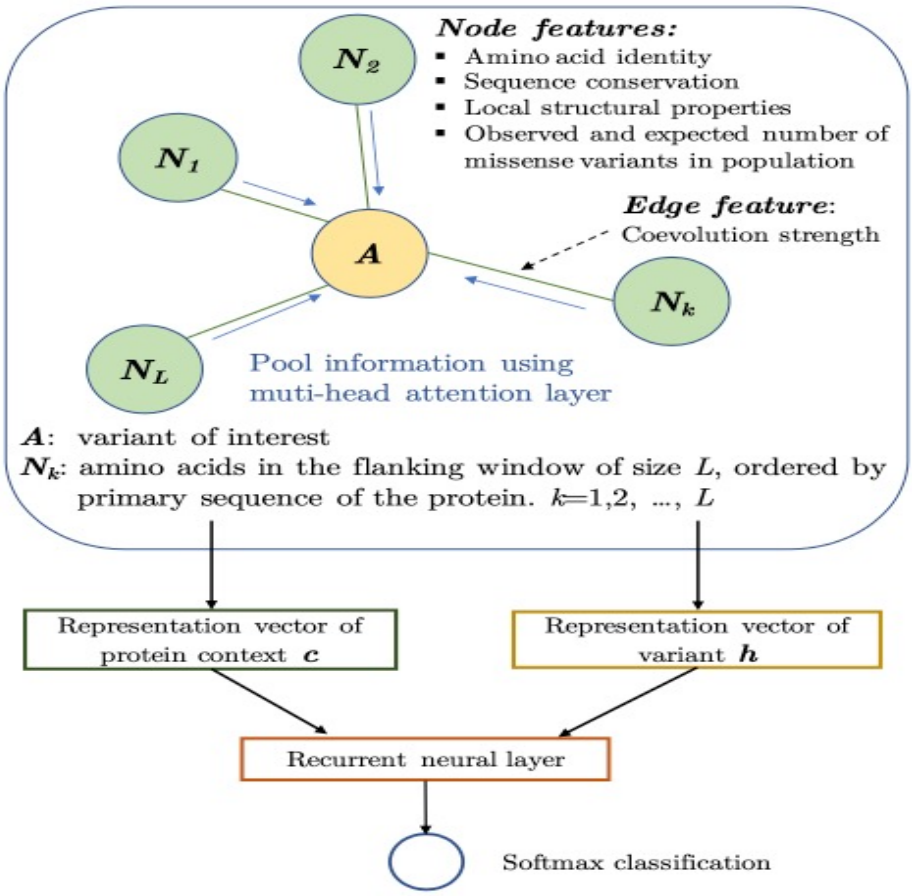
Theoretically, using the output from the MSA Transformer to train the gMVP model could make the model be even more efficient that the original data that trained the gMVP model.

3 Methodology And Data

For this research, I was given access to the gMVP program and also I was given access to the Original Data that the gMVP model was trained on and also new data, which is the output from the MSA Transformer program.

The way gMVP works as a supervised machine learning method is that it predicts functionally damaging missense variants. The consequence of missense variants is based on the

type of amino acid substitution and its protein context. Significantly, gMVP uses a graph attention neural network to learn representation of protein sequence and structure context and context-dependent impact of amino acid substitutions on protein function. See Zhang and et al ((2022))



Credit: diagram is from Zhang and et al ((2022))

The above diagram illustrates how gMVP works. A graph is created to represent a variant and its protein context defined as 128 amino acids that flank the amino acid of interest. The amino acid of interest is the node labeled "A" and the flanking amino acids the surrounding nodes labeled "N". The context nodes are connected with the center node but not each other. The edge feature is coevolution strength. Zhang and et al ((2022)) gMVP uses three 1-depth dense layers to encode the input features to latent representation vectors and used a multi-head attention layer to learn a context vector "c". It then uses a recurrent neural layer connected with a softmax layer to generate a prediction score from the context vector "c" and the representation vector "h" of variant. Zhang and et al ((2022))

The Original Data that was used to train the gMVP model came from three curated databases: HGMD (see Stenson and et al ((2003))), ClinVar (see Landrum ((2014))), and UniProt (see Mottaz ((2010))).

The positive training set in the Original training data used 22,607 variants from the ClinVar database (see Landrum ((2014))) under the Pathogenic and Likely-Pathogenic categories with review status of at least one star, 48,125 variants from the HGMD data based (see Stenson and et al ((2003))) under the disease mutation (DM) category, and 20,481 variants from UniProt (see Mottaz ((2010))) labeled as Disease-Causing. The negative training sets in the Original trianing data used 41,185 variants from ClinVar (see Landrum ((2014))) under the Benign and Likely-Benign categories, 33,387 variants from SwissVar at the UniProt database (see Mottaz ((2010))) labeled as Polymorphism. Zhang and et al ((2022)) at 16. Also, for estimating evolutional conservation, gMVP used data from two sources: (1) the homologous of the protein of interest against SwissProt database (Bateman and et al. ((2019))) with 3 iterations of search and then the built multiple sequence alignments (MSAs) with HHblits suite.(Remmert and et al. ((2012))) (2) the MSAs of 192 species downloaded from Ensemble website for each human protein sequence (Ensemble ((2022)).

3.1 Running the original gMVP Model

I set up and ran the gMVP program with the Original Data on a Linux workstation with 1 NVIDIA Titan RTX GPU. I had to create a new Anaconda environment (Anaconda ((2022)), install Python and the main machine learning program, PyTorch and various Python libriaries.(see Pytorch ((2022))). To go through 50 epochs, the program took an average of 3 to 4 minutes for each epoch. So altogether, it took about 175 minutes to go through all the data.

The gMVP Model's original training was tested to result in "functional readout data from deep mutational scan assays of four well-known disease risk genes, TP53 (see Kotler and et al. ((2018))), PTEN (see Mighll ((2018)), BRCA1 (see Findlay and et al ((2018)), and MSH2 (see Jia and et al. ((2021)), as benchmark data." (Zhang and et al ((2022)) at 7).

TP53 is the name of a gene that provides instructions for making a protein called tumor protein p53. This protein is involved in many processes in the body, including cell growth and division, DNA repair, and programmed cell death (apoptosis). Mutations in the TP53 gene can lead to the development of cancer and other health problems. For example, TP53 mutations are commonly found in a variety of cancers, including leukemias, lymphomas, and solid tumors such as breast, ovarian, and lung cancer. TP53 mutations can also cause Li-Fraumeni syndrome, a rare inherited disorder that increases the risk of developing several types of cancer.

PTEN (phosphatase and tensin homolog) is a protein that acts as a tumor suppressor by regulating the cell cycle and promoting apoptosis (programmed cell death). PTEN is commonly mutated or deleted in a variety of human cancers, including breast, prostate, and brain

cancer. Dysregulation of PTEN signaling has been linked to the development and progression of cancer, and PTEN-deficient tumors tend to be more aggressive and resistant to treatment. PTEN mutations are also associated with certain inherited cancer syndromes, such as Cowden syndrome and Bannayan-Riley-Ruvalcaba syndrome. In these syndromes, individuals have an increased risk of developing multiple types of cancer due to inherited PTEN mutations.

BRCA1 (breast cancer 1) is a protein that plays a critical role in DNA repair and the maintenance of genomic stability. BRCA1 is a tumor suppressor, meaning that it helps to prevent the development of cancer by regulating cell division and DNA repair. Mutations in the BRCA1 gene are associated with an increased risk of breast, ovarian, and several other types of cancer. BRCA1 mutations are particularly common in individuals with a family history of breast or ovarian cancer, and are often inherited in an autosomal dominant pattern. Women with BRCA1 mutations have a lifetime risk of breast cancer that can be as high as 80

MSH2 (mutS homolog 2) is a protein that is involved in DNA mismatch repair (MMR), a process that helps to correct errors that occur during DNA replication. MSH2 plays a critical role in maintaining the integrity of the genome by recognizing and repairing mismatched bases in DNA. Mutations in the MSH2 gene are associated with an increased risk of cancer, particularly colorectal cancer. Individuals with inherited MSH2 mutations have an increased risk of developing colorectal cancer, as well as other types of cancer, such as endometrial, ovarian, and stomach cancer. MSH2 mutations are also associated with an inherited cancer syndrome called hereditary nonpolyposis colorectal cancer (HNPCC), which is characterized by a predisposition to early onset colorectal cancer and a high risk of developing other types of cancer.

gMVP, as part of the model training, plots out two types of curves for each of the four diseases: AUROC curves and precision-recall curves.

AUC-ROC (area under the receiver operating characteristic curve) is a metric used to evaluate the performance of a binary classification model. The ROC curve is a graphical representation of the true positive rate and the false positive rate at different classification thresholds. The true positive rate (TPR) is the proportion of positive cases that are correctly identified by the model, also known as the sensitivity or recall. The false positive rate (FPR) is the proportion of negative cases that are incorrectly identified as positive by the model. The AUC-ROC is the area under the ROC curve, which is a measure of the model's ability to distinguish between positive and negative cases. A model with a high AUC-ROC value has a high true positive rate and a low false positive rate, indicating that it is able to accurately identify positive cases while minimizing the number of false positives. The ROC curve is a useful tool for comparing the performance of different models, as it provides a visual representation of

the trade-off between the true positive rate and the false positive rate. A model with a higher AUC-ROC value is generally considered to be a better model than one with a lower AUC-ROC value.

Precision-recall curves are a graphical representation of the trade-off between the precision and recall of a binary classification model. Precision is the proportion of positive predictions that are actually positive, while recall is the proportion of actual positive cases that are correctly identified by the model.

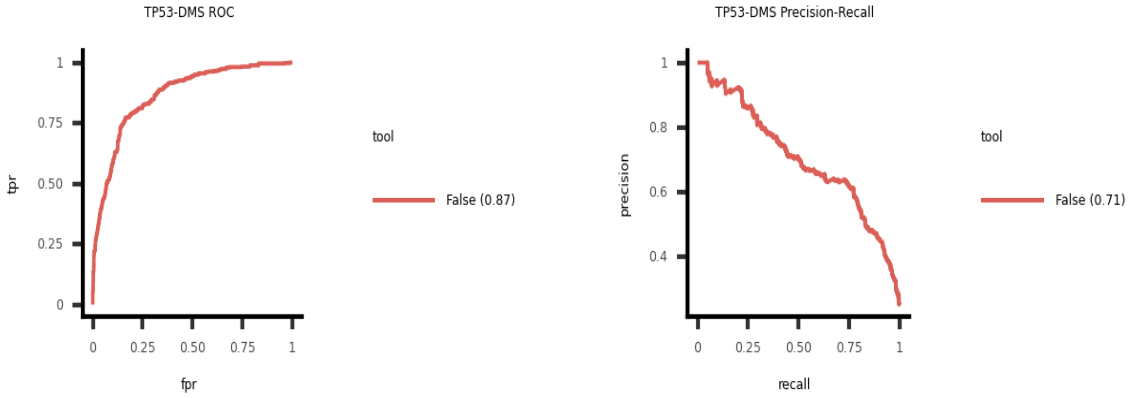
A precision-recall curve plots the precision on the y-axis and the recall on the x-axis at different classification thresholds. The curve is generated by varying the classification threshold and calculating the corresponding precision and recall values.

Precision-recall curves are useful when the goal is to identify all positive cases, even if it leads to a higher number of false positives. For example, in a medical setting, it may be more important to identify all cases of a particular disease, even if it leads to some false positives, rather than miss some cases and have a higher precision but lower recall.

The area under the precision-recall curve (AUPRC) is a measure of the model's performance, with a higher value indicating a better model. The AUPRC is often used when the positive class is rare or when the cost of false negatives is high.

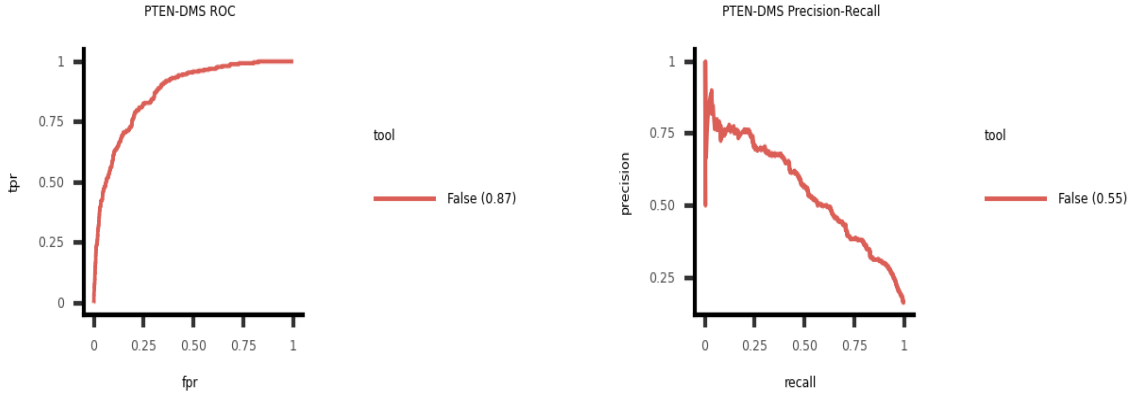
These are the AUROC plot outcomes from the model trained with the Original Data:

Results for TP53



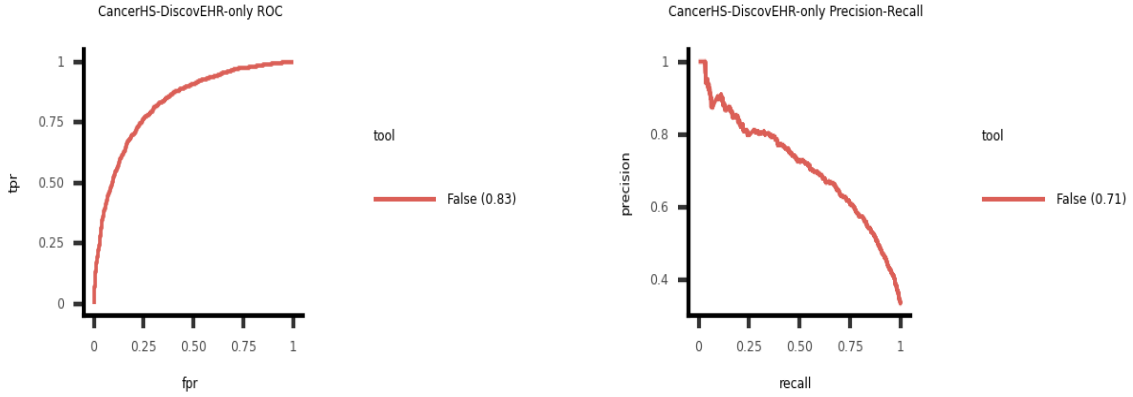
For TP53, my run of the Original Data on the gMVP model, resulted in a 0.87 ROC score and 0.71 Precision-Recall score.

Results for PTEN



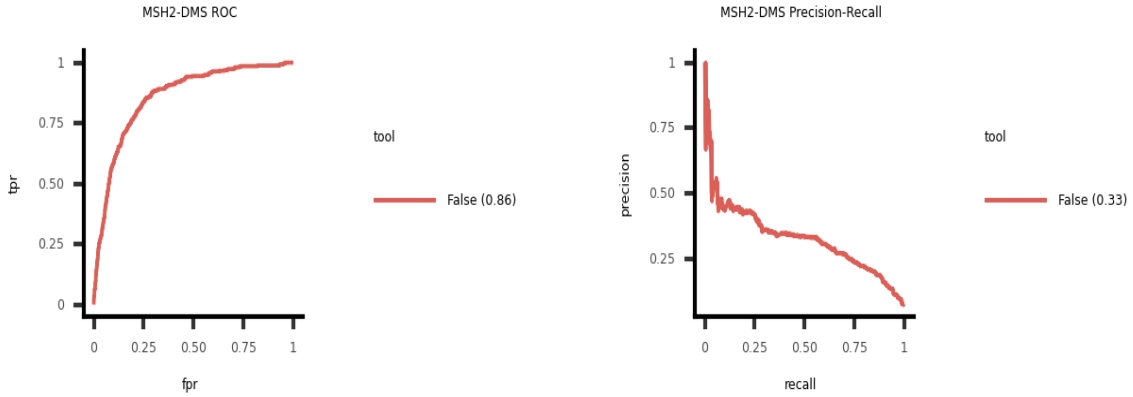
For PTEN, my run of the Original Data on the gMVP model, resulted in a 0.87 ROC score and 0.55 Precision-Recall score.

Results for BRCA1



For BRA1, my run of the Original Data on the gMVP model, resulted in a 0.83 ROC score and 0.71 Precision-Recall score.

Results for MSH2



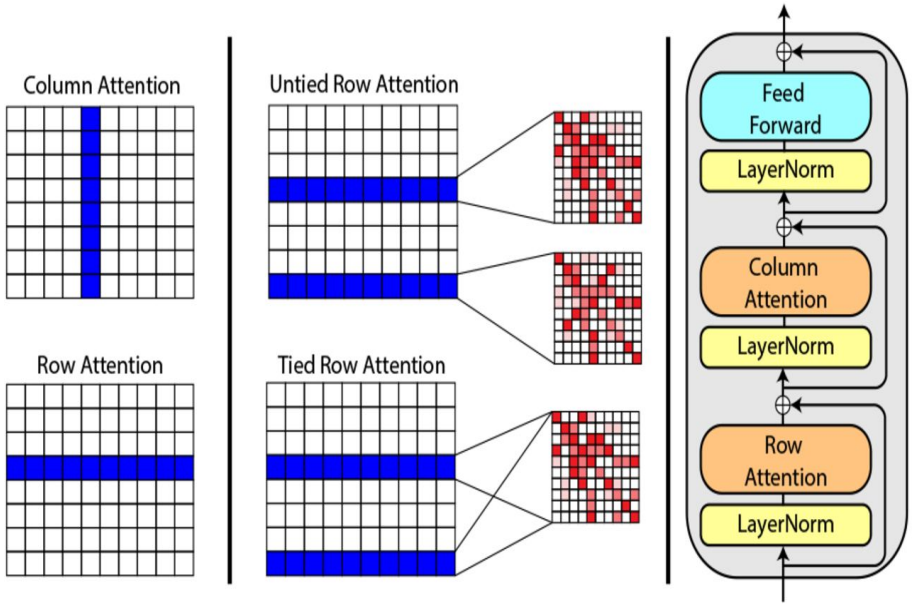
For MSH2, my run of the Original Data on the gMVP model, resulted in a 0.86 ROC score and 0.33 Precision-Recall score.

3.2 The MSA Transformer Data

The MSA Transformer language model can be used to extract embeddings from multi-sequence alignments (MSA). As explained by Rao et al:

> We introduce the MSA Transformer, a model operating on sets of aligned sequences. The input to the model is a multiple sequence alignment. The architecture interleaves attention across the rows and columns of the alignment as in axial attention (Ho et al., 2019). We propose a variant of axial attention which shares a single attention map across the rows. The model is trained using the masked language modeling objective. Self supervision is performed by training the model to reconstruct a corrupted MSA.

Rao and et al. ((2021)) MSA Transformer model the contact pattern among the proteins something called **"row attention"**: "Information about the contact pattern emerges directly in the tied row attention maps." Rao and et al. ((2021)). This diagram explains how the model finds the pattern of the protein row attention.



Credit: diagram is from Rao and et al. ((2021))

MSA Transformer, in addition to "row attention", then creates the contact data: "[W]e fit a sparse logistic regression to the model's row attention maps to identify heads that correspond with contacts." Rao and et al. ((2021)). Thus, the MSA Transformer creates two types of relevant data: `contacts.pt and row_attention.pt`

3.3 Editing the gMVP Model To Work with MSA Tranfomer data

To get MSA Transformer data to work with the gMSV Program, I had to create code language in Python to feed the data into the program. Below is the code:

```
search_path2 = self.feature2_dir
contacts_location = search_path2 + file + ".contacts.pt"
row_attention_locaton = search_path2 + file + ".row_attentions.pt"


if os.path.exists(contacts_location):
    contacts_file = torch.load(contacts_location)
    contacts_file = contacts_file[:, 0, :]
    row_attention = torch.load(row_attention_locaton)
    row_look = 4
    row_attention = row_attention[torch.arange(1, row_attention.shape[0] + 1) != row_look,
    row_attention = row_attention[0, :, :, 0, :].reshape(-1, row_attention.shape[-1])
    shape_no = row_attention.shape[1]
    row_attention = row_attention[:, 1:shape_no]
    combined_data = torch.cat((contacts_file, row_attention), 0)
    final_torch = torch.transpose(combined_data, 0, 1)
    what_file = "yes"
    return_data = final_torch
    feature_len2 = return_data.shape[0]
    final_feature2 = np.zeros([WIDTH * 2 + 1, return_data.shape[1]])
    first_one = return_data[var_start:var_end]
    second_one = return_data[start:end]
    return_data = return_data.detach().numpy()
    final_feature2[var_start:var_end] = return_data[start:end]
    second_data = final_feature2

 else:
    final_torch = torch.zeros(129, 235)
    what_file = "no"
    second_data = final_torch
```

As the MSA Transformer data was in two types of files: `contacts.pt and row_attention.pt`, the code above first uses the unique name of the Original Data file which will have the same unique name as the related `contacts.pt and row_attention.pt` files. In the above code, the term "file" is the variable for the unique name. Because not all the MSA Transformer data matched the data Original Data one for one, my code using an "if" statement first checks to see if there are files that match the name of the Original Data file. If the if-statement confirms those related files exists, i.e. `contacts.pt and row_attention.pt`, the code then has to reshape the tensors to fit training the model.The `contacts_file]` shape was a three dimensional matrix: 1, 429, 429. My new code changed that shape to two dimensions: 429, 429. The original shape of the `row_attention file` was a 5 dimensional matrix:[1, 12, 12, 430, 430]. That had to be changed to two dimensions of reshaped `row_attention file` to size [235, 129]. Then the code concatenates the two tensors that have the same two dimensional shape. Because I am using PyTorch, the machine learning language that deals with tensors, I am using PyTorch built-in

PyTorch function called "cat" to concatenate the two tensors that have the same shape. Then to match the shape of the Original Data, the program transposes the two dimensions so they are [129, 235]. Finally, the code takes the data and then just uses 64 side of of where the varient is in the data. That is why the first part of the shape is 129. 129 is 64 on one side of the variant plus 64 on the other side of the variant and 1 for the variant.

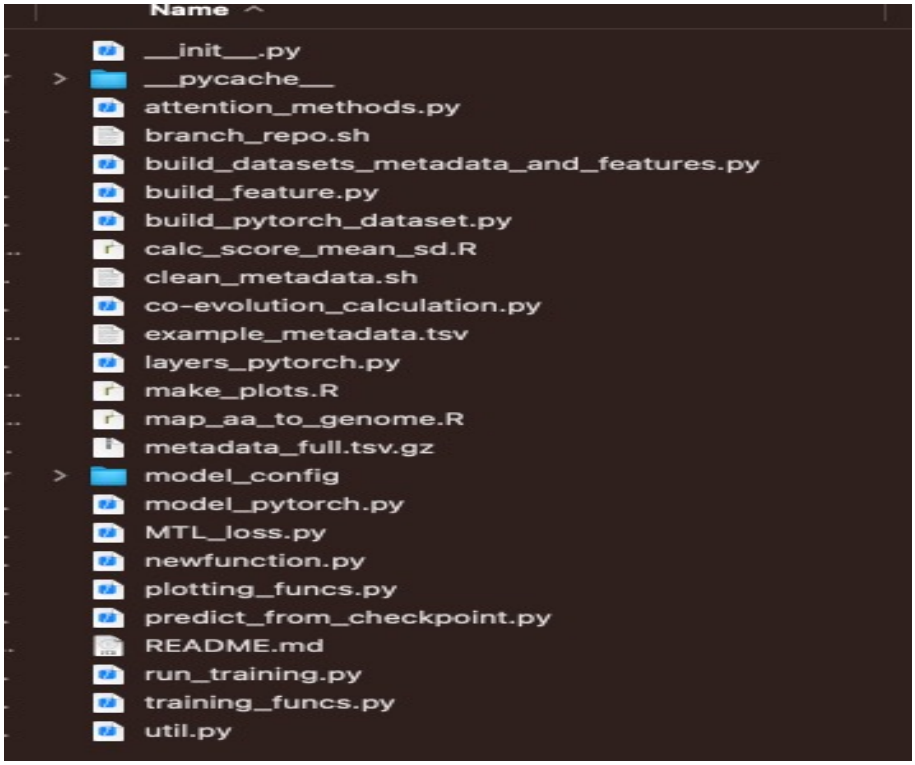
The "else" statement also plays a significant role in this program. As some of the Original Data files do not have related `contacts.pt and row_attention.pt`, the "else" statement creates a tensor matching the same shape of the final tensor in "if" but filled with zeros. This also has the shape: [129, 235]

Depending on if "If" or "else" is used, the `"second_data"` is what goes into the model.

Then into the model part of the gMVP program, this `"second_data"` is fed and becomes the new "pairwise" in the model.(The pairwise is distance between input vectors, or between columns of input matrices.) That data in the model is first renamed "feature2".

```
pairwise2 = feature2.type(torch.float32)
alt_aa = nn.functional.one_hot(alt_aa.type(torch.int64),20)
ref_aa = nn.functional.one_hot(ref_aa.type(torch.int64),20)
center = torch.cat([
ref_aa, alt_aa, evol[:, center_pos], feature[:, center_pos, 1:21],
    feature[:, center_pos, 221:231]
    ],
    dim=-1).type(torch.float32)
center = self.variant_encoder(center)
query = center[:, None]
context = self.neighbor_encoder(context)
key, value = context, context
if pairwise2 is not None:
    pairwise2 = self.pairwise2_encoder(pairwise2)
pairwise = pairwise2
context = self.mha((query, key, value), pairwise=pairwise, mask=mask)
context = self.dense_dropout(context)
        x = self.gru(center, context)
        x = self.dropout(x)
```

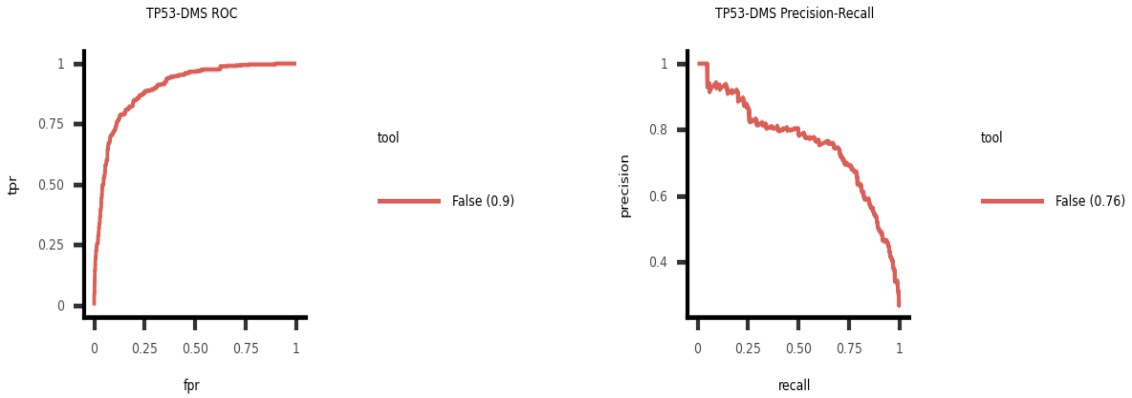
In addition to the above code, I had to go through all the different code files and change a lot of the code for the feature2 data to get through and work with the model code. This is what the gMVP code files look like which is somewhat complex:



Screenshot of gMVP code files taken by me

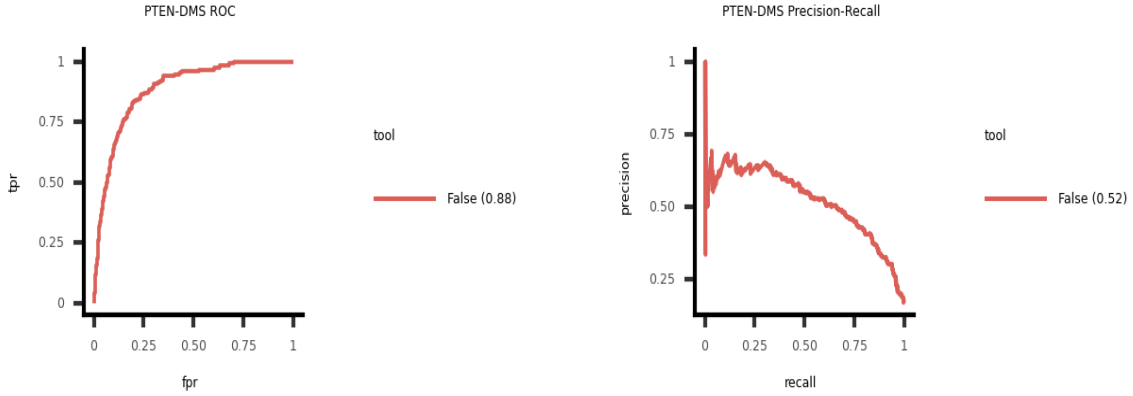
4 Results

When, gMVP model is run on the new data to train it, it again takes 3 to 5 minutes for each of the 50 epochs of training. After it runs, these are the plot outcomes from the new model trained with MSA Transformer: **Results for TP53**



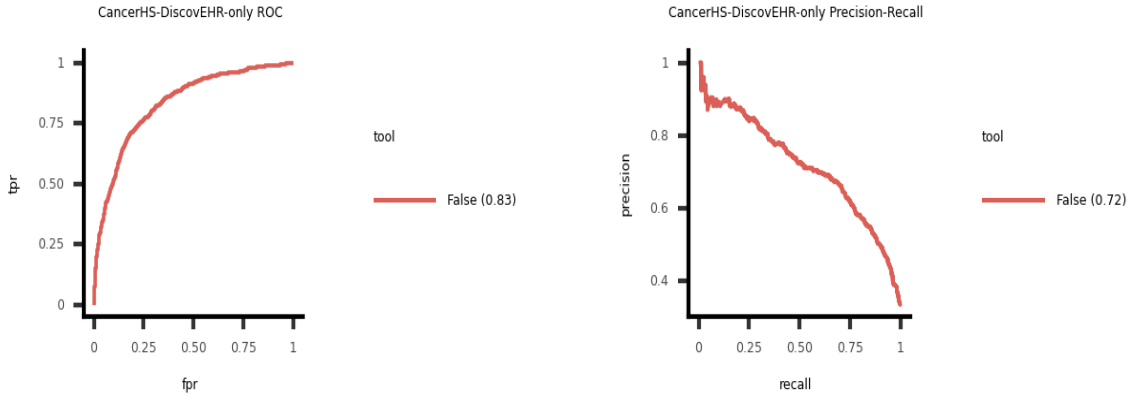
For TP53, my run of the MSA Transformer Data to train the gMVP model, resulted in a 0.9 ROC score and 0.76 Precision-Recall score.

Results for PTEN



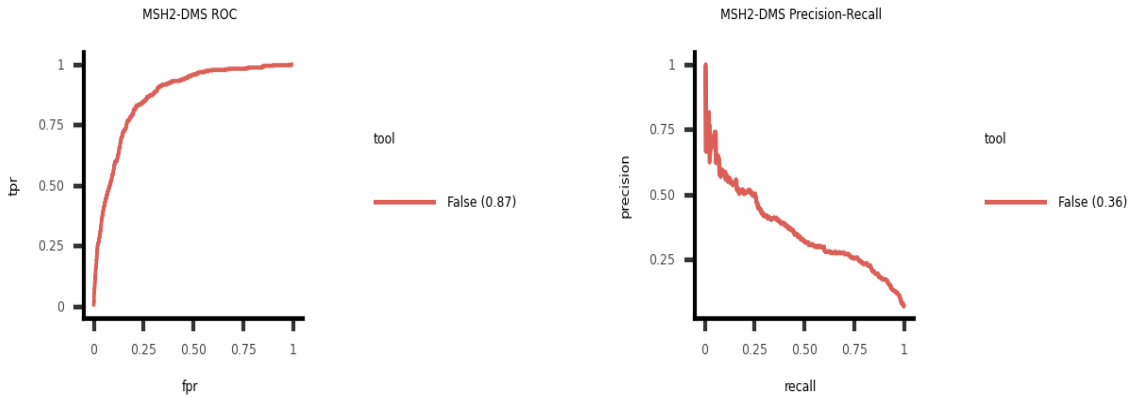
For PTEN, my run of the MSA Transformer Data to train the gMVP model, resulted in a 0.88 ROC score and 0.52 Precision-Recall score.

Results for BRCA1



For BRCA1, my run of the MSA Transformer Data to train the gMVP model, resulted in a 0.83 ROC score and 0.72 Precision-Recall score.

Results for MSH2



For MSH2, my run of the MSA Transformer Data to train the gMVP model, resulted in a 0.87 ROC score and 0.36 Precision-Recall score.

5 Discussion and Conclusion

Here is a chart comparing the ROC and PR results of the Original Data training the model and the MSA Transformer data training the model.

Training Data	TP53 ROC	TP53 PR	PTEN ROC	PTEN PR	BRCA1 ROC	BRCA1 PR	MSH2 ROC	MSH2 PR
Original	0.87	0.71	0.87	0.55	0.83	0.71	0.86	0.33
MSA Trans-former	0.9	0.76	0.88	0.52	0.83	0.72	0.87	0.36

The table above shows that the MSA Transformer data improved the ROC for 3 of the 4 types of diseases: TP53, PTEN and MSH2. For BRCA1, the ROC had the same ROC as the Orignal Data. As for Precision-Recall, the MSA Transformer data improved 3 of 4 types of diseases: TP53, BRCA1 and MSH2. However, the Precision-Recall score reduced for PTEN compared to the score for the Original Data.

One possible flaw in this research may require to again try this experiment by rerunning the Original Data. When I ran the Original Data to train the gMVP Model the ROC and PR scores were *lower* than the scores in the published gMVP article. For example, in the gMVP article, the ROC for TP53, is 0.88 while this research's results for the Original Data scored 0.87. (See Zhang and et al ((2022)) at 6). Also, in the gMVP article, the Precision-Recall scores are as follows: 0.85 (TP53), 0.78 (PTEN), 0.81 (BRCA1), and 0.39 (MSH2), respectively. (See Zhang and et al ((2022)) at 7). These results for the Original Data are higher than the results this research obtained using the Original Data.

This research shows that using MSA Transformer Data to train the gMVP Model improves the model compared to the Original Data that trained it. Possibly combining the Original Data and combining it with the MSA Transformer Data could make the gMVP Model work even better.

Acknowledgements

I would like to thank Dr. Vladimir Shapovalov, my advisor at Bronx Science, Dr. Yufeng Shen at Columbia University, Guojie Zhong a PhD student in Dr. Yufeng Shen's lab, and my supportive parents and brother.

References

I. Adzhubei and et al. Predicting functional effect of human missense mutations using polyphen-2. *Curr Protoc Hum Genet*, Chapter 7, Unit7:20, 2013. doi: https://doi.org/10.1002/0471142905.hg0720s76.

Anaconda. 2022. URL `https://www.anaconda.com`.

A. Bateman and et al. Uniprot: A universal hub of protein knowledge. *Protein Science*, 28: 32–32, 2019. doi: https://doi.org/10.1093/nar/gky1049.

S. Boettcher, P. G. Miller, R. Sharma, M. McConkey, M. Leventhal, A. V. Krivtsov, A. O. Giacomelli, W. Wong, J. Kim, S. Chao, K. J. Kurppa, X. Yang, K. Milenkowic, F. Piccioni, D. E. Root, F. G. Rücker, Y. Flamand, D. Neuberg, R. C. Lindsley, P. A. Jänne, W. C. Hahn, T. Jacks, H. Döhner, S. A. Armstrong, and B. L. Ebert. A dominant-negative effect drives selection of tp53 missense mutations in myeloid malignancies. *Science*, 2019. doi: https://doi.org/10.1002/0471142905.hg0720s76.

H. Carter and et al. Identifying mendelian disease genes with the variant effect scoring tool. *BMC Genomics*, 14 Suppl 3:53, 2013. doi: https://doi.org/10.1186/1471-2164-14-S3-S3.

Ensemble. Protein ensemble database. 2022. URL `https://proteinensemble.org`.

G. M. Findlay and et al. Accurate classification of brca1 variants with saturation genome editing. *Nature*, 562:217+–, 2018. doi: https://doi.org/10.1038/s41586-018-0461-z.

J. M. Havrilla and et al. A map of constrained coding regions in the human genome. *Nature Genetics*, 51:88–+, 2019. doi: https://doi.org/10.1038/s41588-018-0294-6.

M. Hicks and et al. Functional characterization of 3d protein structures informed by human genetic diversity. *Proceedings of the National Academy of Sciences of the United States of America*, 116:8960–8965, 2019. doi: https://doi.org/10.1073/pnas.1820813116.

K.-L. Huang and et al. Pathogenic germline variants in 10,389 adult cancers. *Cell*, 2018. doi: https://doi.org/10.1002/0471142905.hg0720s76.

N. M. Ioannidis and et al. An ensemble method for predicting the pathogenicity of rare missense variants. american journal of human genetics 99, 877-885 (2016). *Human Genetics*, 99:877–885, 2016. doi: https://doi.org/10.1016/j.ajhg.2016.08.016.

I. Ionita-Laza and et al. A spectral approach integrating functional genomic annotations for coding and noncoding variants. *Nature Genetics*, 48:214–220, 2016. doi: https://doi.org/10.1038/ng.3477.

S. Iqbal and et al. Comprehensive characterization of amino acid positions in protein structures reveals molecular effect of missense variants. *Proceedings of the National Academy of Sciences of the United States of America*, 117:28201–28211, 2020. doi: https://doi.org/10.1073/pnas.2002660117.

K. Jagadeesh and et al. M-cap eliminates a majority of variants of uncertain significance in clinical exomes at high sensitivity. *Nat Genet*, 48:1581–1586, 2016. doi: https://doi.org/10.1038/ng.3703.

X. Jia and et al. Massively parallel functional testing of msh2 missense variants conferring lynch syndrome risk. *The American Journal of Human Genetics*, 108:163–175, 2021. doi: https://doi.org/10.1016/j.ajhg.2020.12.003.

S. Jin and et al. Contribution of rare inherited and de novo variants in 2,871 congenital heart disease probands. *Nature Genetics*, 49:1593–+, 2017. doi: https://doi.org/10.1038/ng.3970.

J. Kaplanis and et al. Discovery and characterisation of 49 novel genetic disorders from analysing de novo mutations in 31,058 parent child trio exomes. *European Journal of Human Genetics*, 27:1046–1046, 2019. doi: https://doi.org/10.1038/ng.3970.

M. Kircher and et al. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, 46:310–+, 2014. doi: https://doi.org/10.1038/ng.2892.

E. Kotler and et al. A systematic p53 mutation library links differential functional impact to cancer mutation pattern and evolutionary conservation. *Molecular Cell*, 71:873–873, 2018. doi: https://doi.org/10.1016/j.molcel.2018.06.012.

M. J. Landrum. Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Research*, 42:D980–D985, 2014. doi: https://doi.org/10.1093/nar/gkt1113.

T. Mighll. A saturation mutagenesis approach to understanding pten lipid phosphatase activity and genotype- phenotype relationships. *American Journal of Human Genetics*, 102:943–955, 2018. doi: https://doi.org/10.1016/j.ajhg.2018.03.018.

A. Mottaz. Easy retrieval of single amino-acid polymorphisms and phenotype information using swissvar. *Bioinformatics*, 26:851–852, 2010. doi: https://doi.org/10.1093/bioinformatics/btq028.

Pytorch. 2022. URL `https://pytorch.org`.

H. Qi and et al. Mvp predicts the pathogenicity of missense variants by deep learning. *Nat Commun*, 12:520, 2021. doi: https://doi.org/10.1038/s41467-020-20847-0.

R. Rao and et al. Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020. doi: https://doi.org/10.1101/2020.12.15.422761.

R. Rao and et al. Msa transformer. *bioRxiv*, 2021.

M. Remmert and et al. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature Methods*, 9:173–175, 2012. doi: https://doi.org/10.1038/nmeth.1818.

A. Rives and et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118, 2021. doi: https://doi.org/10.1073/pnas.2016239118.

K. Samocha and et al. Regional missense constraint improves variant deleteriousness prediction. *bioRxiv*, page 148353, 2017. doi: https://doi.org/10.1101/148353.

F. K. Satterstrom and et al. Large-scale exome sequencing study implicates both developmental and functional changes in the neurobiology of autism. *Cell*, 180:568=584.

R. Sively and et al. Comprehensive analysis of constraint on the spatial distribution of missense variants in human protein structures. *American Journal of Human Genetics*, 102:415–426, 2018. doi: https://doi.org/10.1016/j.ajhg.2018.01.017.

P. Stenson and et al. Human gene mutation database (hgmd (r)): 2003 update. *Human Mutation*, 21:577=581, 2003. doi: https://doi.org/10.1002/humu.10212.

L. Sundaram. Predicting the clinical impact of human mutation with deep neural networks. *Nature Genetics*, 50:1161–+, 2018. doi: https://doi.org/10.1038/s41588-018-0167-z.

A. Vaswani and et al. Attention is all you need. *NIPS*, 30, 2017.

H. Zhang and et al. Predicting functional effect of missense variants using graph attention neural networks. *Nature Machine Intelligence*, 2022. doi: https://doi.org/10.1038/s42256-022-00561-w.