Novozymes Enzyme Stability Prediction
Kaggle Competition

https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction

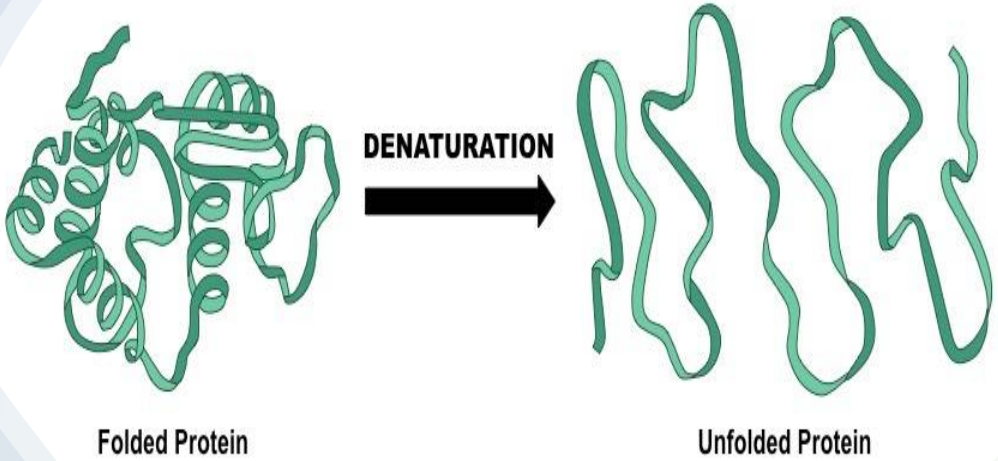
Team members

Amudha Giridharan, Chaithrashree Sadashivappa, Sai Shiva Bhaskar Emani, Paul Merica, Soundarya Alwa

Competition Overview

The Context

- Enzymes are proteins that act as catalysts in the chemical reactions of living organisms
- Many enzymes are only marginally stable, which limits their performance under harsh application conditions.
- They may unfold and denature in harsh conditions like temperature beyond their melting threshold
- Understanding and accurately predicting protein stability is a fundamental problem in biotechnology.
- Its applications include enzyme engineering for addressing the world's challenges in sustainability, carbon neutrality and more.
- Improvements to enzyme stability could lower costs and increase the speed scientists can iterate on concepts.
- More and more protein structures are being solved however, accurate prediction of protein thermal stability remains a great challenge

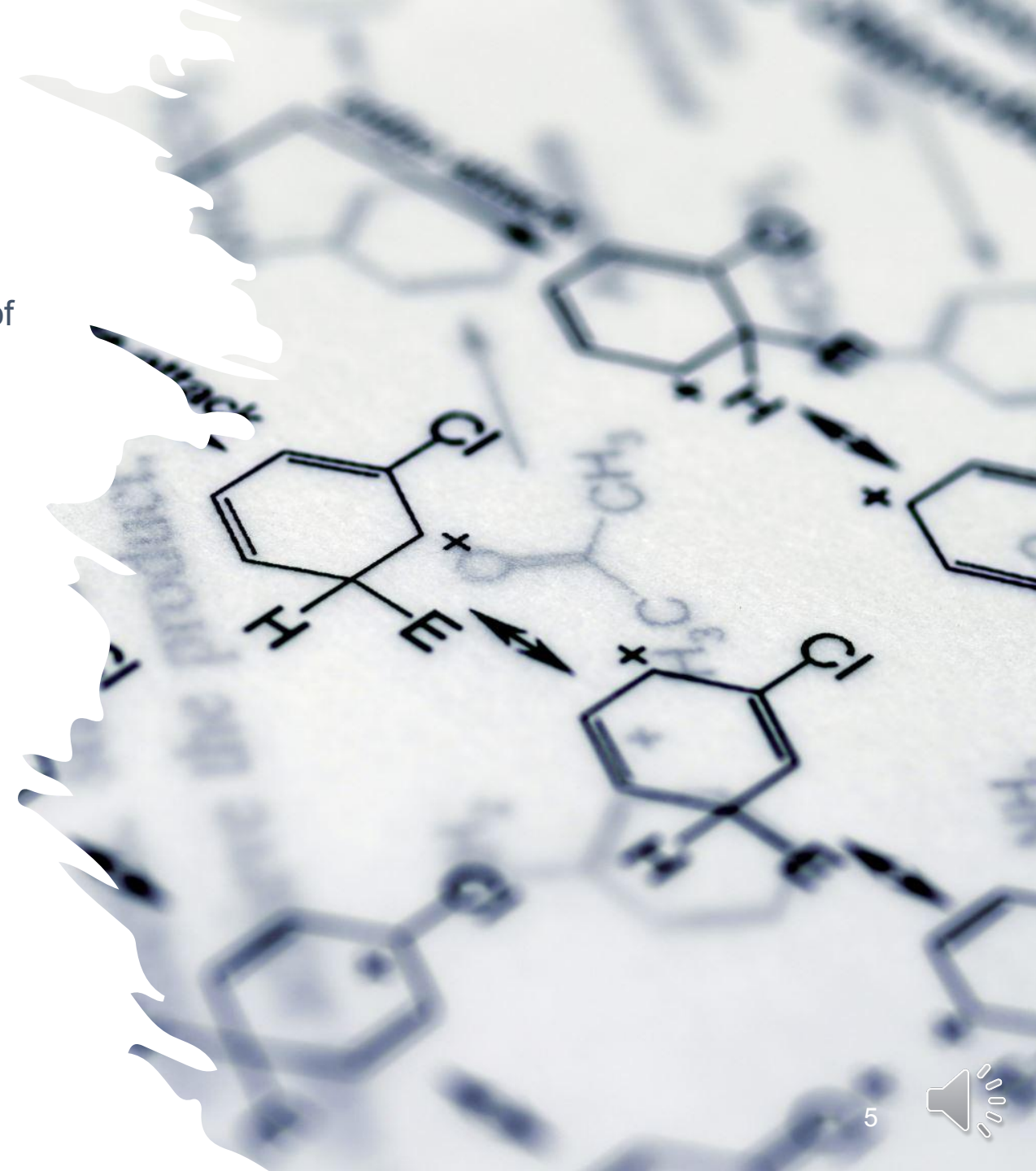


Terminologies

- tm – The protein melting point (TM) is defined as the temperature at which the protein denatures.
- dtm – delta melting temperatures between two enzymes
- ddg - the change in energy between the folded and unfolded states (ΔGfolding) when a point mutation is present.
- blosum - BLOSUM (BLOcks SUbstitution Matrix) matrix is a substitution matrix used for sequence alignment of proteins
- b factor - B-factor aka Debye–Waller factor, temperature factor, or atomic displacement parameter, is used in protein crystallography to describe the attenuation of X-ray or neutron scattering caused by thermal motion.
- Spearman's correlation - a nonparametric measure of rank correlation (statistical dependence between the rankings of two variables). It assesses how well the relationship between two variables can be described using a monotonic function.
- PDB – Protein Data Bank. The PDB format provides for description and annotation of protein and nucleic acid structures including atomic coordinates, secondary structure assignments, as well as atomic connectivity.
- wildtype - A term used to describe a enzyme (in this competition context) when it is found in its natural, non-mutated (unchanged) form
- mutant – protein with an amino acid change from its wildtype (in the context of this competition)

The Goal

- The goal of this competition is to predict the thermostability of enzyme variants.
- The variants should be then ranked based on this predicted thermostability (aka melting temperature - tm), assigning greater ranks to more stable variants
- Submissions are evaluated on the Spearman's correlation coefficient between the ground truth and the predictions.
- This gives us the flexibility to either predict the melting temperature itself, or the predict the delta of melting temperature between the variants (dtm)



Data Sets & EDA

Training Data

Competition host provided 3 data sets for training the model:

- training.csv - contains natural sequences, as well as engineered sequences with mutations upon the natural sequences, pH at experiment and the respective melting temperature.
- train_updates_20220929.csv - corrected rows to resolve data discrepancy in train data set
- External data sources were allowed in this competition: We used related data from microbiologist Jin Yuan (https://github.com/JinyuanSun/mutation-stability-data)

seq_id	protein_sequence	pH	data_source	tm
0	AAAAKAAALALLGEAPEVVDIWLPAGWRQPFRVFRLERKGDGVL...	7	doi.org/10.1038/s41592-020-0801-4	75.7
1	AAADGEPLHNEEERAGAGQVGRSLPQESEEQRTGSRPRRRRDLG...	7	doi.org/10.1038/s41592-020-0801-4	50.5
2	AAAFSTPRATSYRILSSAGSGSTRADAPQVRRLHTTRDLLAKDYYA...	7	doi.org/10.1038/s41592-020-0801-4	40.5
3	AAASGLRTAIPAQPLRHLLQPAPRPCLRPFGLLSVRAGSARRSGLL...	7	doi.org/10.1038/s41592-020-0801-4	47.2
4	AAATKSGPRRQSQGASVRTFTPFYFLVEPVDTLSVRGSSVILNCSA...	7	doi.org/10.1038/s41592-020-0801-4	49.5
5	AACFWRRTVIPKPPFRGISTTSARSTVMPAWVIDKYGKNEVLRFT...	7	doi.org/10.1038/s41592-020-0801-4	48.4
6	AACFWRRTVIPKPPFRGISTTSARSTVMPAWVIDKYGKNEVLRFT...	7	doi.org/10.1038/s41592-020-0801-4	45.7

MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA	<---Wildtype
AKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA	<-----Example of Single point mutation
MKGASKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA	
MKGMAKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA	
MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKAEGRIGA	
MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGAIGA	
MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIAA	

```
[1] "alanine - lysine - glycine - methionine - serine - lysine - methionine - proline - glutamine - phenylalanine - asparagine
 - leucine - arginine - tryptophan - proline - arginine - glutamic acid - valine - leucine - aspartic acid - leucine - valine
 - arginine - lysine - valine - alanine - glutamic acid - glutamic acid - asparagine - glycine - arginine - serine - valine -
 asparagine - serine - glutamic acid - isoleucine - tyrosine - glutamine - arginine - valine - methionine - glutamic acid - se
rine - phenylalanine - lysine - lysine - glutamic acid - glycine - arginine - isoleucine - glycine - alanine"
```

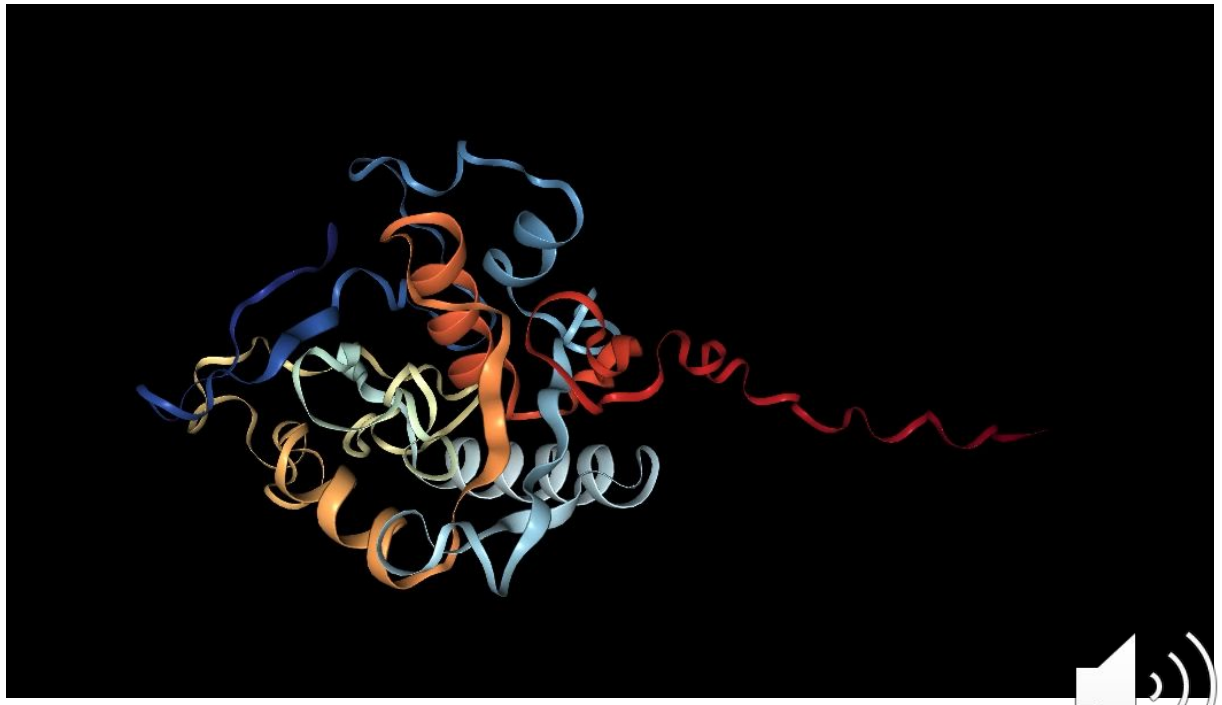
Test Data

- test.csv – contains a single wild type, 77 deletion mutations of this wild type and 2,335 substitution mutations of the same wild type
- wildtype_structure_prediction_af2.pdb - the 3-dimensional structure of the test enzyme, as predicted by AlphaFold

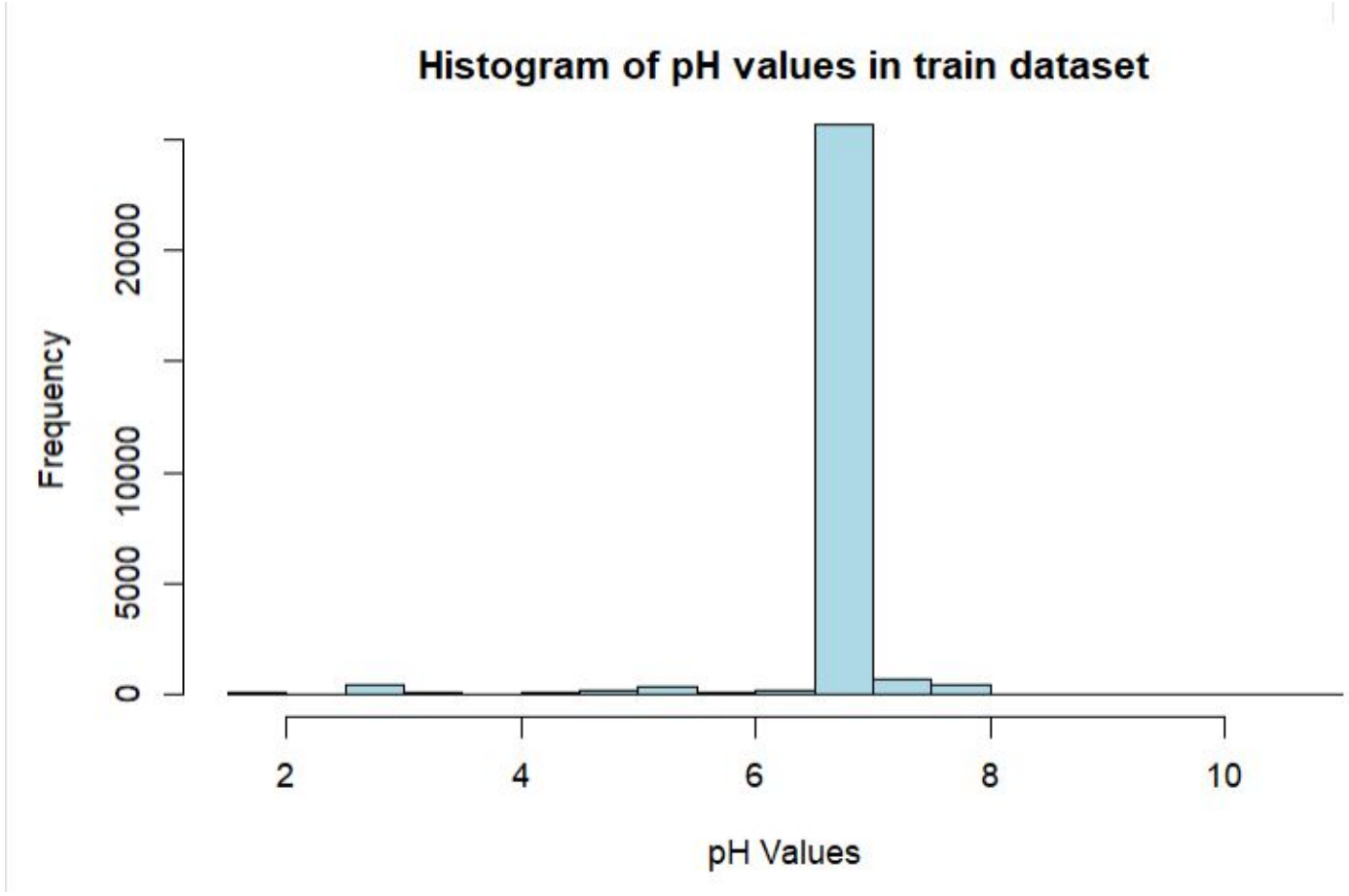
test data structure

seq_id	protein_se...	pH	data_source
31390	VPVNPEPDATSVENV AEKTGSGDSQSDPIK ADLEVKGQSALPFDV DCWAILCKGAPNVLQ RVNEKTKNSNRDRSG ANKGPFKDPQKWGIK ALPPKNPSWS...	8	Novozymes
31391	VPVNPEPDATSVENV AKKTGSGDSQSDPIK ADLEVKGQSALPFDV DCWAILCKGAPNVLQ RVNEKTKNSNRDRSG ANKGPFKDPQKWGIK ALPPKNPSWS...	8	Novozymes
31392	VPVNPEPDATSVENV AKTGSGDSQSDPIKA DLEVKGQSALPFDVD CWAILCKGAPNVLQR VNEKTKNSNRDRSGA NKGPFKDPQKWGIKA LPPKNPSWSA...	8	Novozymes

3D view of the test wild type protein structure

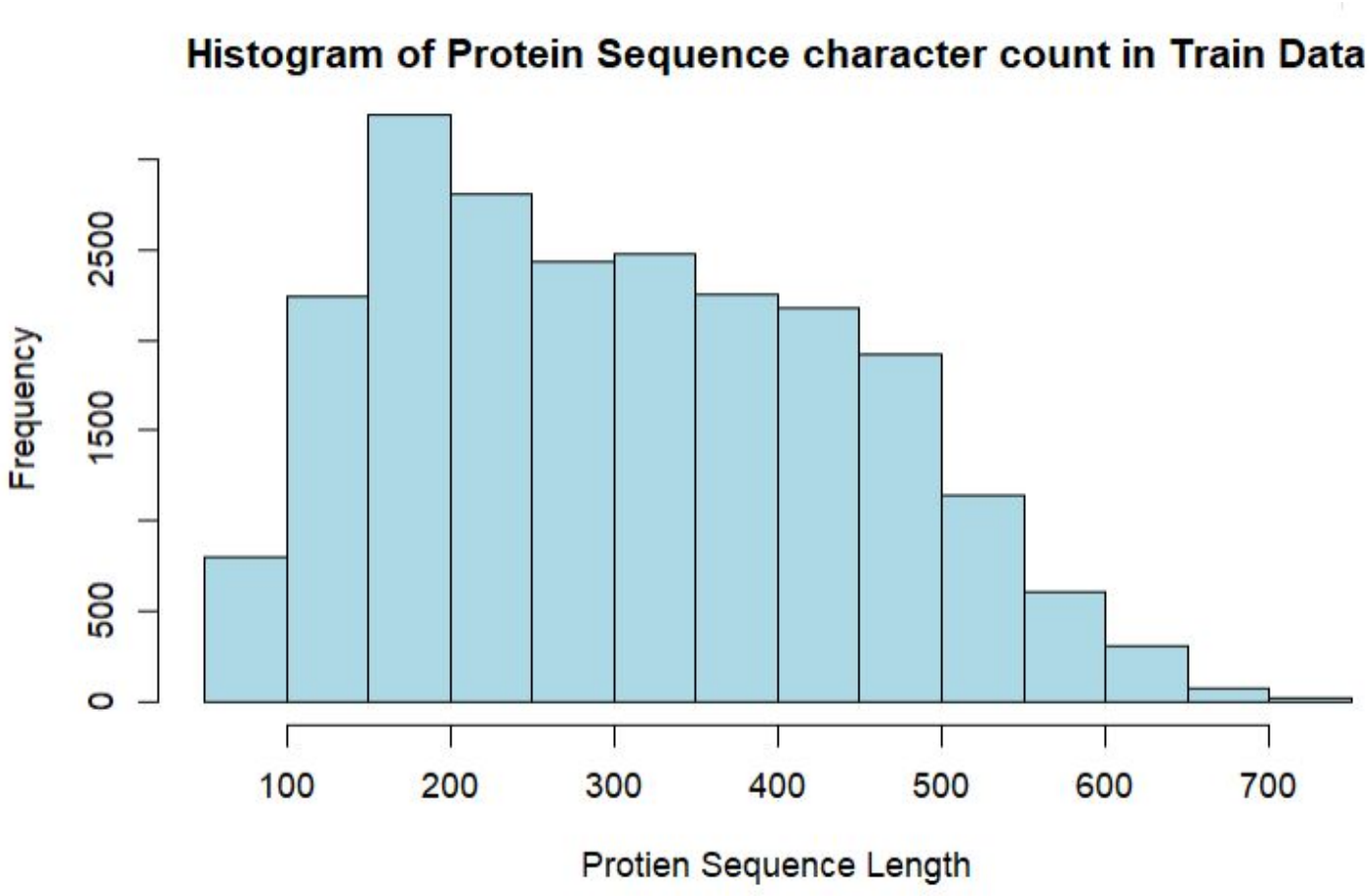


EDA on Train and Test Data



- The pH values in the Train data have a mean of 6.8 and a median of 7
- The pH values in the Test data is 8 for all the records

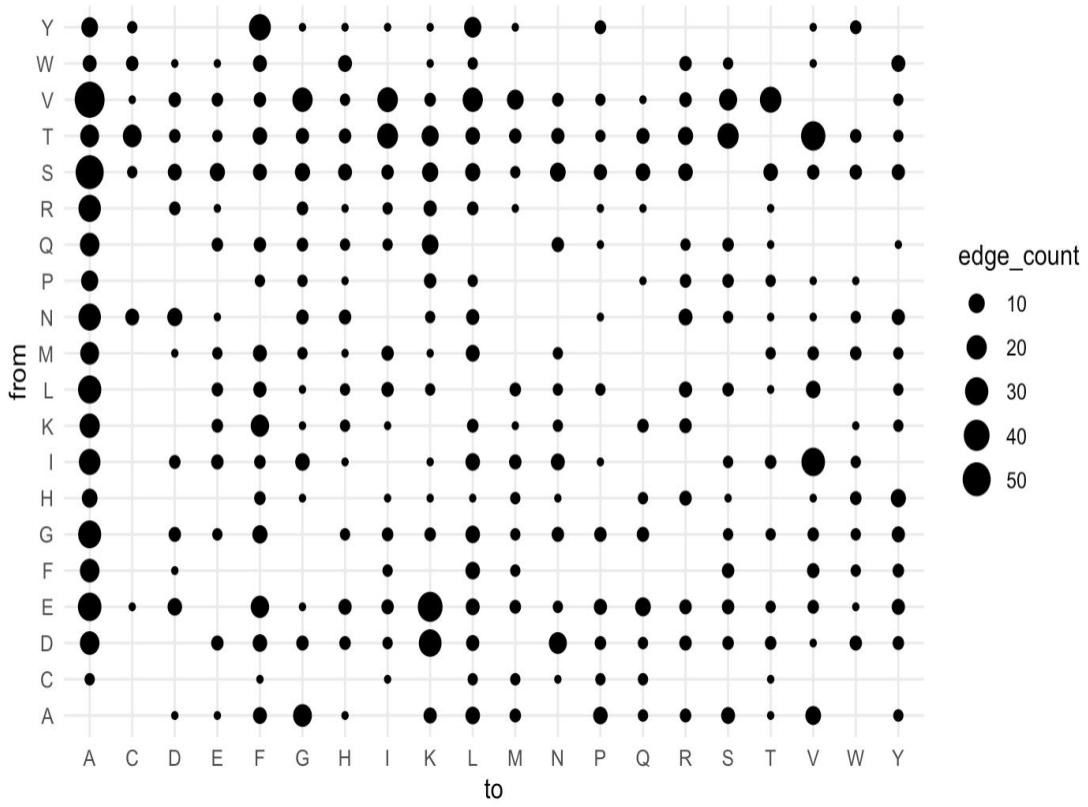
EDA on Train and Test Data



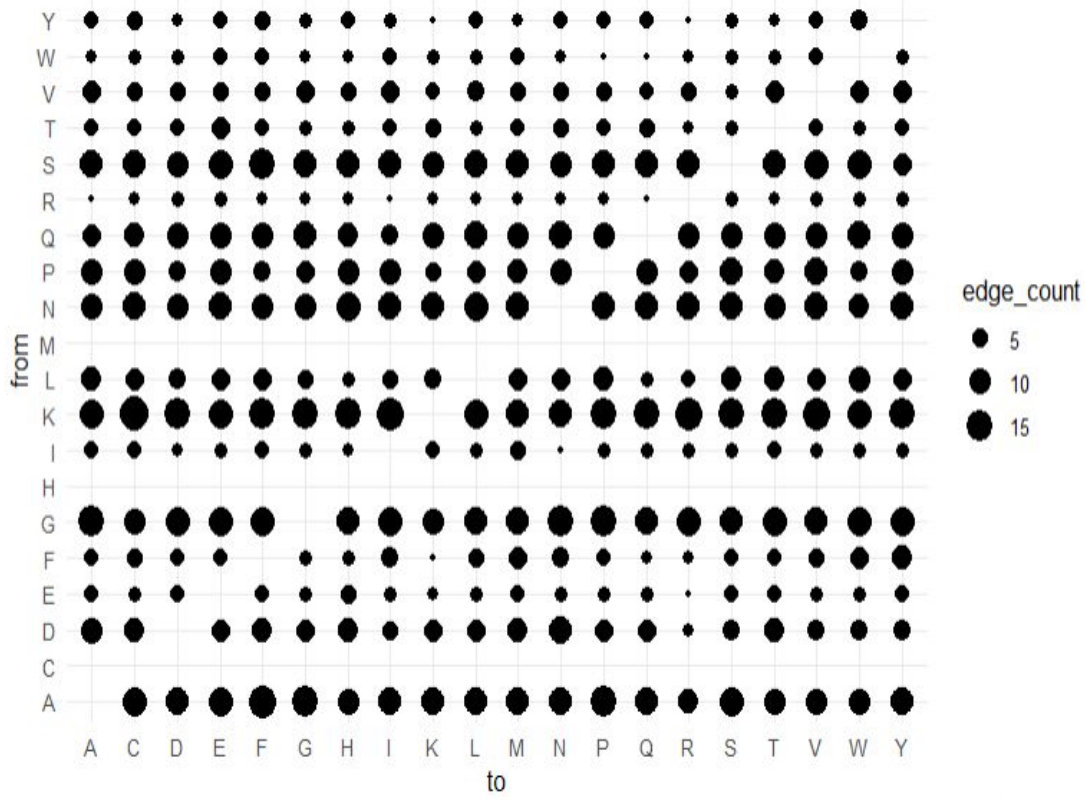
- The character count for the Protein Sequence in the Train data is spread across with a "range from 53 to 747".
- **"97%"** of the Protein Sequence in the Test data has character length of 221 and the remaining **"3%"** is 220

Train & Test Data – Node graph

Train Data



Test Data



Initial Data Analysis & Challenges identified

Train data deficiency

- Train data had single point substitution mutations only
- Test data had both single point substitution and deletion mutations
- The model might not be able to predict tm for deletion mutation as there was no training data available for this scenario

Training data correctness

- Train data was a compilation from multiple data sources
- Similar protein sequences at the same pH had different tm, based on the data source in the train data
- Test data was from a single data source, and wasn't referenced in the training data source
- The model might not be able to predict the tm correctly for the new data source

Test data singularity

- Entire test data set was based on single wild type
- Train data did not contain granular level observations to match with test set

Feature Engineering

Feature Engineering – Amino acid weightages

Amino acid weightage within a protein sequence

- The team did a POC to find if bioseq R package can help with creating amino acid weightage for the sequences, and the results were affirmative.
- The proportion of each amino acid was calculated for all the sequences using Seq_stat_prop function Bioseq package from R
- With this weightage calculated, we can tie back the change in tm to the variation in weightage between mutations

Sample:

The below protein sequence is hard to model, but the numeric weightages in screenshot enables us to model the data right. We then merge this information with the train data.

```
> seq_stat_prop(aa('MNQSVSSLPEKDIQYQLHPYTNARLHQELGPLIIERGEGIYVYDDQGKGYIEAMAGLWSAALGFSNQRLIKAAEQQFNTLPFYHLFSHKSHRPSIELAEKLIAMAPVPMSKVFFTN
SGSEANDTVVKMVWYLNNALGKPAKKKFISRVNGYHGITVASASLTGLPGNQRGFDLPLPGFLHVGCPHHYRFALAGESEEHFADRLAVELEQKILAEGPETIAAFIGEPLMGAGGVIVPPRTYWEKIQKVCRKYD
ILVIADEVICGFGRTGQMFGSQTFGIQPDIMVLSKQLSSSYQPIAAILINAPVFEGIADQSQALGALGHGFTGSGHPVATAVALENLKIIEEESLVEHAAQMGQLLRSGLQHFIDHPLVGEIRGCGLIAAVELVGD
RVSKAPYQALGTLGRYMAGRAQEHGMITRAMGDAVAFCPPLIVNEQEVGMIVERFARALDDTTQWVG
+ '))
[[1]]
          A           C           G           T           W           S           M           K           R           Y
0.101098901 0.010989011 0.101098901 0.035164835 0.008791209 0.054945055 0.028571429 0.039560440 0.041758242 0.030769231
          B           D           H           V           N           E           F           I           L           P
0.000000000 0.032967033 0.035164835 0.065934066 0.028571429 0.068131868 0.043956044 0.070329670 0.094505495 0.052747253
          Q
0.054945055
```

	A	C	G	T	W	S	M	K	R	Y
IQYQLHPYTNARLHQELGPLIIERGEGIYVY...	0.1010989	0.010989011	0.1010989	0.03516484	0.008791209	0.05494505	0.02857143	0.03956044	0.04175824	0.03076923
IQYQLHPYTNARLHQELGPLIIERGEGIYVY...	0.0989011	0.010989011	0.1010989	0.03516484	0.008791209	0.05494505	0.02857143	0.04175824	0.04175824	0.03076923
IQYQLHPYTNARLHQELGPLIIERGEGIYVY...	0.0989011	0.010989011	0.1010989	0.03516484	0.008791209	0.05494505	0.02637363	0.03956044	0.04175824	0.03076923
IQYQLHPYTNARLHQELGPLIIERGEGIYVY...	0.0989011	0.010989011	0.1010989	0.03516484	0.008791209	0.05494505	0.02637363	0.03956044	0.04175824	0.03296703
IQYQLHPYTNARLHQELGPLIIERGEGIYVY...	0.0989011	0.010989011	0.1010989	0.03516484	0.008791209	0.05274725	0.02857143	0.03956044	0.04175824	0.03076923
IQYQLHPYTNARLHQELGPLIIERGEGIYVY...	0.0989011	0.010989011	0.1010989	0.03516484	0.008791209	0.05494505	0.02857143	0.03956044	0.04175824	0.03076923

Feature Engineering – Protein sequence clustering

Protein sequence clustering to identify potential mutations

- Again, we did POC for methods to identify the potential mutations in training data, and we identified that seq_nchar and seq_cluster function from bioseq R package can help with this.
- We computed the aminoacid count in each of the sequence using seq_nchar function.
- This helped us to group potential mutations together. Substitution mutations would have same character size as of the wild type and deletion mutation will have one-character lesser size.
- We then used seq_cluster function to compute consensus and assign representative numbers for clusters.

Sample:

The below is representative sample of amino acid character count and cluster sequencing.

Filter

e	pH	data_source	tm	charcnt	clusternum	wildtype	A	C
RWPREVLDLVRKVAEENGRSVNSEIYQRV...	7.5	10.1038/nsb0894-518	58.0	53	1		0.05660377	0.0000000
TIVSDGKPQTDNDTGMISYKDANGNKQ...	7.0	doi.org/10.1038/s41592-020-0801-4	62.9	53	2		0.01886792	0.0000000
RWPREVLDLVRKVAEENGRSVNSEIYQR...	7.5	10.1038/nsb0894-518	58.7	53	1		0.05660377	0.0000000
TTFAIESTVELSFEEAMTPEVIEKYNAKTT...	7.0	doi.org/10.1038/s41592-020-0801-4	37.5	53	3		0.07547170	0.0000000
RWPREVLDLVRKVAEENGRSVNSEIYQR...	7.5	10.1038/nsb0894-518	58.1	53	1		0.05660377	0.0000000
RWPREVLDLVRKVAEENGRSVNSEIYQRV...	7.5	10.1038/nsb0894-518	59.2	53	1		0.05660377	0.0000000
.RWPREVLDLVRKVAEENGRSVNSEIYQR...	7.5	10.1038/nsb0894-518	57.5	53	1		0.05660377	0.0000000
RWPREVLDLVRKVAEENGRSVNSEIYQR...	7.5	10.1038/nsb0894-518	59.6	53	1		0.05660377	0.0000000
RWPREVLDLVRKVAEENGRSVNSEIYQRV...	7.5	10.1038/nsb0894-518	55.5	53	1		0.05660377	0.0000000
.RWPREVLDLVRKVAEENGRSVNSEIYQR...	7.5	10.1038/nsb0894-518	74.1	53	1		0.05660377	0.0000000

Showing 1 to 10 of 22,507 entries, 29 total columns

```
> table(train_filtered$charcnt,train_filtered$clusternum)

    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
53 52  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
63  1  1  1  1  1  1 21  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
66  1  1  1  1  2 20  1  1  1 53  1  1  1  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
67  1  1  1  1  1  1 45  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
71  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0
84  1  1  1  1  1  1  1  1  1  1  1 55  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
85  1  1  1  1  1  4  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0
87  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0
89  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  0
90  1  1  1  1  1  1  1  1  1  1  1 13  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0
```

Feature Engineering – Wild type consensus

Wild type for the mutation clusters

- With the clustered protein sequences, we used the seq_consensus to find the wild type.
- The POC we did on various data sets from test and train data confirmed the reliability of this function.
- For each amino acid character count and cluster combination, we used the consensus technique to identify the wild type, only if the total mutations in that category is substantial.
- Trivial clusters with protein sequences less than the threshold (set to 20) were filtered out.
- We now had the wild type mapped to each protein sequence for our further analysis

Sample:

The below is representative sample of consensus derivation for wild type identification.

```
> seq_consensus(aa(train_filtered[(train_filtered$charcnt == 53 & train_filtered$clusternum == 1), ]$protein_sequence))
AA vector of 1 sequences
>   MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA
> train_filtered[(train_filtered$charcnt == 53 & train_filtered$clusternum == 1), ]$protein_sequence
 [1] "AKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MAGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
 [3] "MKAMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGASKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
 [5] "MKGMAKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSAMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
 [7] "MKGMSKAPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMAQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
 [9] "MKGMSKMLQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPAFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[11] "MKGMSKMPQANLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFALRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[13] "MKGMSKMPQFNARWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLAWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[15] "MKGMSKMPQFNLRAPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWAREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[17] "MKGMSKMPQFNLRWPAEVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPRAVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[19] "MKGMSKMPQFNLRWPREALDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPREVADLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[21] "MKGMSKMPQFNLRWPREVLALVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPREVLDAVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[23] "MKGMSKMPQFNLRWPREVLDLVAKVAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPREVLDLVRAVAEENGRSVNSEIYQRVMESFKKEGRIGA"
[25] "MKGMSKMPQFNLRWPREVLDLVRKAAEENGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPREVLDLVRKVAAENGRSVNSEIYQRVMESFKKEGRIGA"
[27] "MKGMSKMPQFNLRWPREVLDLVRKVAEANGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPREVLDLVRKVAEEAGRSVNSEIYQRVMESFKKEGRIGA"
[29] "MKGMSKMPQFNLRWPREVLDLVRKVAEEAGRSVNSEIYQRVMESFKKEGRIGA" "MKGMSKMPQFNLRWPREVLDLVRKVAEENARSVNSEIYQRVMESFKKEGRIGA"
```

Feature Engineering – Mutation position and type

Position and type of mutation

- With the wild type identified, we then added the information of residue position of mutation and type of mutation (substitution/deletion/wildtype) and appended that to the training set.
- This information further helped to enhance the model accuracy

Sample:

The below is corresponding position and mutation information from the previous slide.

```
> head(train_filtered[ , c("protein_sequence","wildtype","charcnt","clusternum" ,"type", "resid","wt","mut")])
                                    protein_sequence                                    wildtype charcnt clusternum type resid wt mut
1 AKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA      53          1  SUB     1  M   A
2 MAGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA      53          1  SUB     2  K   A
3 MKAMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA      53          1  SUB     3  G   A
4 MKGASKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA      53          1  SUB     4  M   A
5 MKGMAKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA      53          1  SUB     5  S   A
6 MKGMSAMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA MKGMSKMPQFNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA      53          1  SUB     6  K   A
```

The POC notebook with the Bioseq package functions for feature engineering was published by our team in Kaggle
Link is https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction/discussion/371747

Test data set enhancement

B factor

- The test wild type PDB file provided was leveraged to deduce the b-factor and was used to enhance the test dataset
- The B-factor describes the displacement of the atomic positions from an average (mean) value (mean-square displacement). Higher flexibility results in larger displacements and, eventually, lower electron density, leading to lesser stability
- Therefore, B factor is proven to be inversely proportional to delta thermal stability for both deletion and substitution mutations

Blosum score

- Leveraged the Blosum matrix to find the alignment score for the substitutions in the test data set
- Blosum alignment score is proven to be directly proportional to thermal stability

Modelling Methods & Analysis

Modelling Methods

This led us down 3 ways to model the data:

- Model directly on Tm (thermal stability) and predict form weightage/mutation data. This would only use (Kaggle training data which is filtered down to 1634 rows after our groupings)
- Model on dTm (the difference in thermal stability of the mutation string from the wildtype string) . This also allows us to some of the external data from Micro-Biology Researcher - Jin (2632 rows) . No PH in this model .
- Model on normalized (mean of 0, standard deviation of 1) ddG and dTm by grouped wildtypes (ddG and dTm are usually inversely correlated, but this already corrected in this external dataset) . This allows us to compare these measures directly and train on them. This method allows us to use all of the external data from this Micro-Biology Researcher (6642 rows) . No pH in this model.

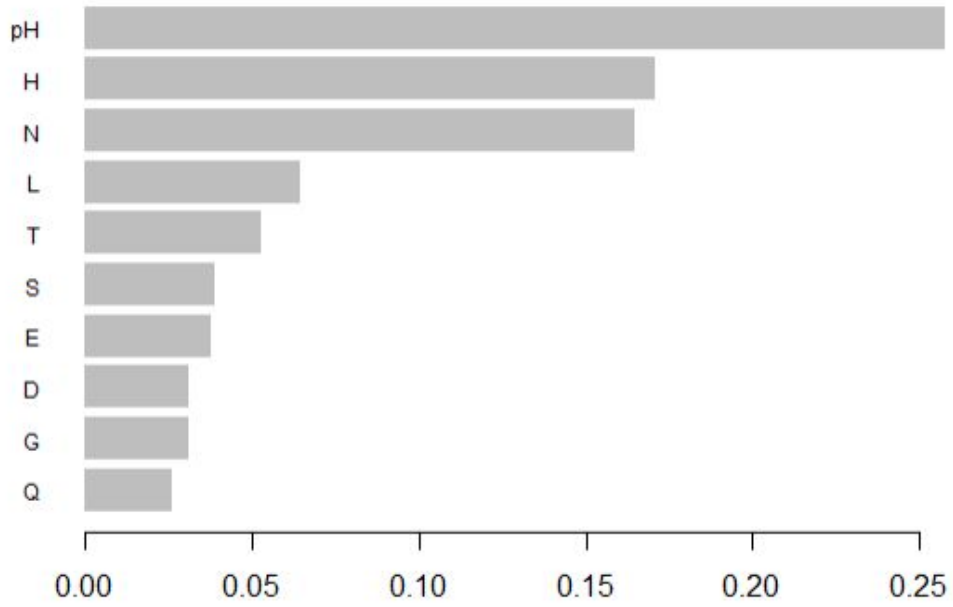
Modelling methods Comparison

Analysis Finding - XGboost was better for both Tm Prediction and dTm prediction. While Random Forest was better on the normalized prediction method.

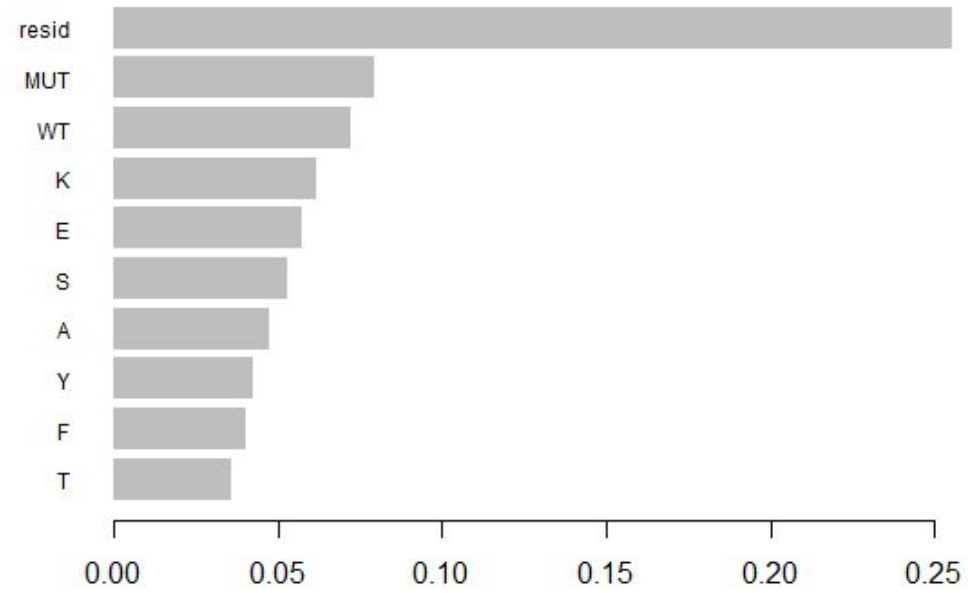
Validation Set MSE	XGB	Random Forest	CTREE	RPART	TREE	GBM
Tm Prediction	33.9	38.7	41.9	45	44.8	38.9
dTm Prediction	20.5	23.7	23	23.4	23.3	24.36
Normalized Prediction	.83	.79	.92	.92	1.02	.92

Importance Graphs for the Two XGB models

XGB TM



XGB Raw dTM



HyperParameter Tuning for Final models

XGB TM

```
#Setting up train for XGB training
xgb_train = xgb.DMatrix(data=(data.matrix(traingrp[,-2])), label=(traingrp[,2] ))
xgb_test = xgb.DMatrix(data=(data.matrix(testxgb[,-2])), label=(testxgb[,2] ))

# Using Cross Validation for best parameter
xgbcv = xgb.cv(data = xgb_train, nfold =25, nrounds =1000, early_stopping_rounds = 40)

opt_iterations = xgbcv$best_iteration

xgb <- xgboost(data = xgb_train, max.depth=5,nrounds=opt_iterations)
```

```
[48]    train-rmse:3.320600+0.067863    test-rmse:5.808892+0.824579
[49]    train-rmse:3.302793+0.066572    test-rmse:5.815543+0.826385
[50]    train-rmse:3.285932+0.061033    test-rmse:5.823334+0.828303
[51]    train-rmse:3.272147+0.061716    test-rmse:5.828117+0.830344
[52]    train-rmse:3.255452+0.061722    test-rmse:5.833975+0.830214
[53]    train-rmse:3.239568+0.061754    test-rmse:5.835998+0.829984
[54]    train-rmse:3.225454+0.062774    test-rmse:5.839196+0.830854
[55]    train-rmse:3.211521+0.060765    test-rmse:5.841874+0.828613
[56]    train-rmse:3.199090+0.063970    test-rmse:5.847947+0.825641
Stopping. Best iteration:
[16]    train-rmse:4.170187+0.089069    test-rmse:5.591269+0.754519
```

XGB Raw dTM

```
xgb_train = xgb.DMatrix(data=(data.matrix(dTm_mastertable[,-1])), label=(dTm_mastertable[,1] ))
xgb_test = xgb.DMatrix(data=(data.matrix(testxgb[,-1])), label=(testxgb[,1] ))

xgbcv_dtm = xgb.cv(data = xgb_train, nfold =25, nrounds =1000, early_stopping_rounds = 40)

xgbcv_dtm$best_iteration

opt_iterations_dtm = xgbcv_dtm$best_iteration

xgb_dtm  <- xgboost(data = xgb_train, max.depth=5,nrounds=opt_iterations_dtm)
```

```
[65]    train-rmse:2.615596+0.046799    test-rmse:4.634141+0.660299
[66]    train-rmse:2.601689+0.045859    test-rmse:4.640272+0.670981
[67]    train-rmse:2.586849+0.049265    test-rmse:4.644494+0.677599
[68]    train-rmse:2.576731+0.049178    test-rmse:4.647567+0.684768
Stopping. Best iteration:
[28]    train-rmse:3.246016+0.050777    test-rmse:4.485755+0.607431
```

RF normalized Prediction

```
mtry <- tuneRF(Ultra_dataset2[-1],Ultra_dataset2$dtm, ntreeTry=500,
               stepFactor=1.5,improve=0.01, trace=TRUE, plot=TRUE)
best.m <- mtry[mtry[, 2] == min(mtry[, 2]), 1]
print(mtry)
print(best.m)

rfo_dtm <- randomForest(dtm ~., data = Ultra_dataset2, mtry = best.m)
```

```
mtry = 8  OOB error = 0.685954
Searching left ...
mtry = 6        OOB error = 0.6923058
-0.009259808 0.01
Searching right ...
mtry = 12       OOB error = 0.6694303
0.02408864 0.01
mtry = 18       OOB error = 0.6643963
0.007519806 0.01
   mtry  OOBError
6     6 0.6923058
8     8 0.6859540
12   12 0.6694303
18   18 0.6643963
[1] 18
```

Handling Deletions & Ensemble modelling

- There was no deletions in the training data, so for deletion data we used the b_factor & blosum matrix to rank them since the model wouldn't be able to pick up on this.
- We realized that the blosum matrix and b_factor were decently predictive after testing it out on the deletions in Kaggle submissions.
- Thus, we decided to make our final models an equally weighted ensemble model of the average our models + B_factor rankings + Blosum matrix rankings.

Kaggle Results From Our Models

Our TM model performed the best on the test data and put at around the middle of the pack in the leaderboard (rankings change every day)

Challenges, Achievements & Future Plans

Challenges

- Extreme amount of bioinformatics knowledge is needed in order to understand terminologies and correlations. The team had to do lot of research on bioinformatics to get up-to speed
- Due to the limited information in the given train data and test data, training data alone wasn’t sufficient to predict test observations tm
- The competition encourages external data utilization and part of this is the use of ESM and PDB files.
- We weren't able to incorporate ESM and external PDB files into our models as this requires a lot of industry knowledge into the protein data structures or additional deep dive analysis(most submissions have been working on this for multiple months).
- Also, ESM is a python package, and we couldn’t find any pretrained protein R language model which we could leverage to extract similar features for a protein sequence.
- Complex feature engineering were required for accurate prediction

Achievements

- While most of the Kaggle experts sided Python, we attempted the implementation with R
- The team did a POC with R bioseq package and the notebook was published in Kaggle https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction/discussion/371747
- We leveraged external data from Blosum matrix and also used the b-factor from PDB file to support model performance
- GitHub was used and the code base is available under https://github.com/amudhagiridharan/Novozome-Kaggle-competition

Possible Enhancements

- Using Amino acids properties data from external source, we are planning to calculate the delta values for all the physical and chemical properties of a wildtype letter with mutated letter and using this as a new feature
- Using embeddings from ESM model to get new features.
- Using CIF and PDB files to get the x, y, z co-ordinates of the corresponding wildtype and mutated element in each sequences to calculate the location id as a new feature for before and after mutation.

Conclusion

Conclusion

- Our team attempted to use R for solving this problem when compared to majority of the competitors used Python.
- We felt that Python had more socialized machine learning packages when compared to R and we had to do a lot of POCs to come up with a sustainable approach.
- The team was not familiar with bioinformatics before this competition, and we learned a lot during this process, competing with bioinformatics industry masters.
- Despite all these, we were still able to build a relatively high-scoring model that got us to the middle of the leaderboards, within a couple of weeks.
- We plan to improve our model by incorporating additional predictors and ranking methods to help build our score to reach an even higher

References

https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction

https://en.wikipedia.org/wiki/BLOSUM

http://thegrantlab.org/bio3d/articles/online/pdb_vignette/Bio3D_pdb.html

https://proteinstructures.com/structure/protein-databank/

https://github.com/JinyuanSun/mutation-stability-data

https://www.kaggle.com/code/pjt222/sequence-as-graph