Classifying Protein Stability Changes upon Mutations using PRoBERTa

Presented by: Anowarul Kabir

Outlines

- Motivation
- Problem Formulation
- Embedding function
- Classification module
- Performance analysis

Motivation

- DNA mutations translate to changes to the amino acids that make up a protein molecule.
- Pathogenic mutations cause proteinopathies (disorders due to proteins).
- Mutations impact the ability of a protein (sequence/chain of amino acids) to fold into the expected, biologically-active structure.
- The impact of a mutation can be measured in the wet laboratory.
- **Could we predict it in silico?**

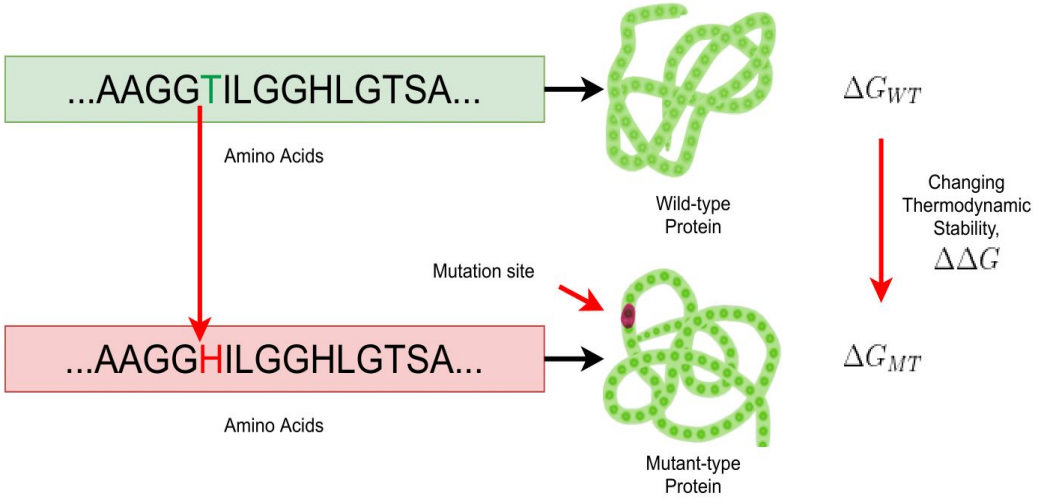

Fig: The schematic shows a single-point mutation, in which one amino acid is mutated, giving rise to a “mutant” version of the naturally-occurring version of a particular protein molecule.

Problem Formulation: Classifying the Mutations *in Silico*

- Wildtype and variant protein.
- Thermodynamic stability change (DeltaDeltaG or ddg).
- Stabilizing: if ddg>=0
- Destabilizing: if ddg<0
- A classification task: predicting mutation type given a wildtype and variant protein.

$$pred_{x_i} = g(E^w_{x_i}, E^m_{x_i}|\theta)$$

- Loss function: Cross-entropy loss.

$$\theta = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} out_{x_i} log(pred_{x_i}) + (1 - out_{x_i}) log(1 - pred_{x_i})$$

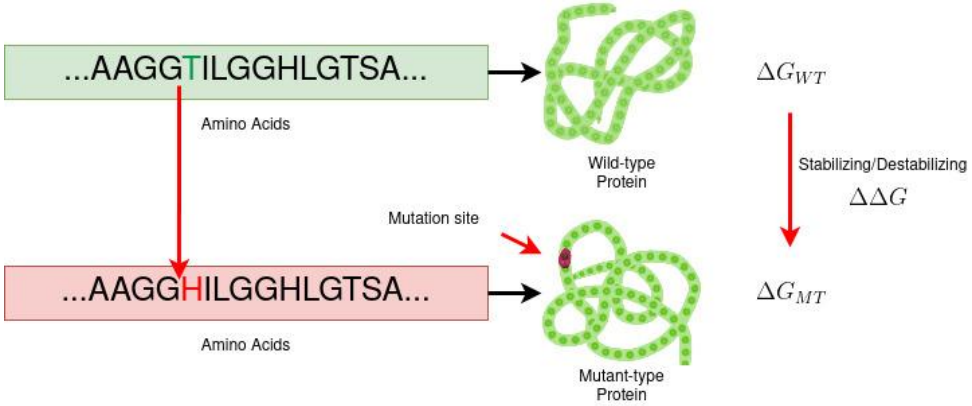

Fig: A schematic illustration of protein mutation type.

Embedding function

- Pretrained transformer model called PRoBERTa.
- Input: a tokenized protein sequence of size *k* tokens
- Output: the last layer features of size *(k, 768)*
- Sequence embedding:

$$E_{x_i} = Em(tok_1^{(i)}) + Em(tok_2^{(i)}) + ... + Em(tok_{k_i}^{(i)})$$

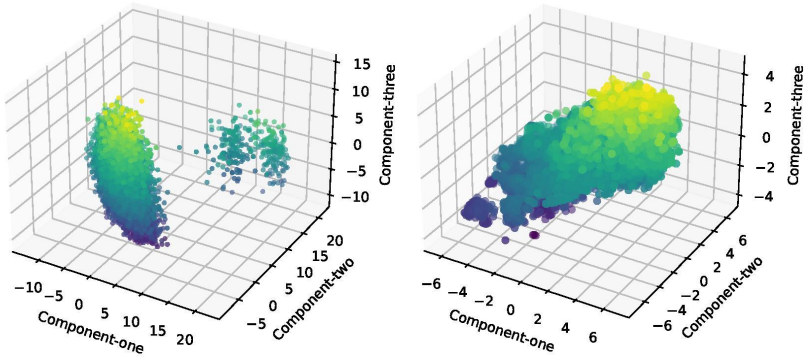

Fig: Vocabulary embedding of first three components using PCA (left) and T-SNE (right).

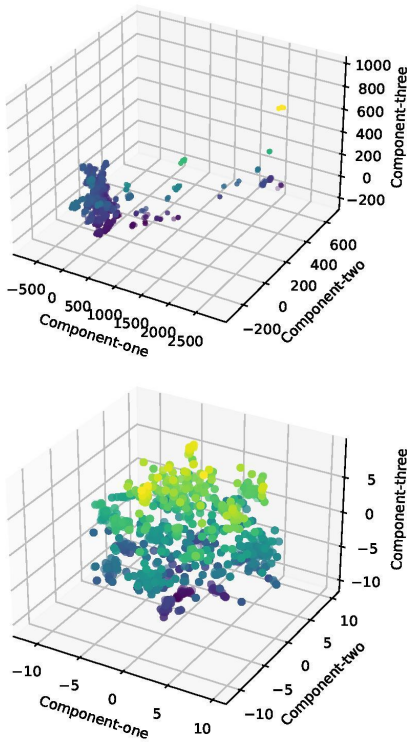

Fig: Protein sequence embedding of first three components using PCA (upper) and T-SNE (lower).

Classification module

Model-0

- No hidden layers
- 1,536 learable weights

Model-1

- 2 hidden layers of size (512, 64)
- 8,19,328 learable weights

Hyperparameters:

- Learning rate: 0.001, 0.0001, 0.00001
- Batch-size: 32, 64
- Class weights: [0.4, 0.6], [0.5, 0.5], [0.6, 0.4]

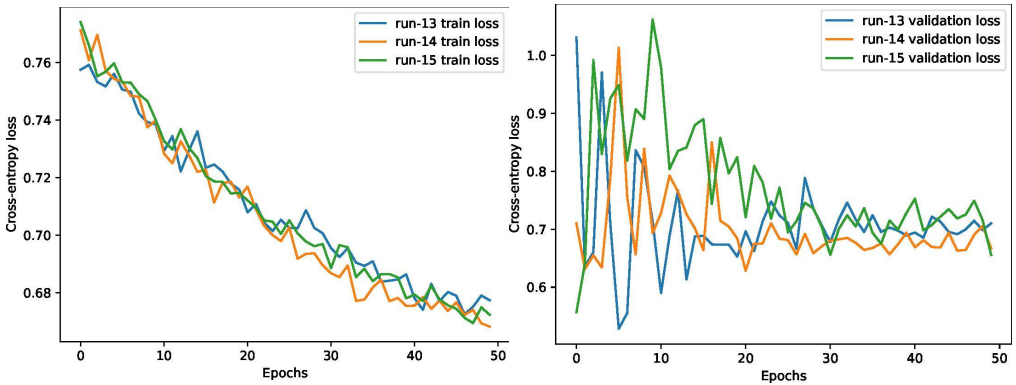

Fig: Best training and validation loss for run-13, 14 and 15 of Model-1.

Performance analysis

- The best performing model is run-15 of Model-1.
- The learned sequence features may not have enough information to solve the problem.
- The problem, predicting the mutation type based on very tiny distortion compared to the dynamic sequence length, poses a difficult problem.
- The number of features corresponding to variant amino acids compared with other amino acids is same.

	Hyperparameters				Performance analysis					
Run no	Learning rate	Batch-size	Epochs	Class weights	Confusion matrix[c]	Accuracy	Precision	Recall	F1-score	AUC
					Model-0					
1	0.001	32	50	0.6, 0.4	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
2	0.0001	32	50	0.6, 0.4	182, 0, 71, 0	0.719	0	0	0	0.5
3	0.001	32	50	0.5, 0.5	182, 0, 71, 0	0.719	0	0	0	0.5
4	0.0001	32	50	0.5, 0.5	182, 0, 71, 0	0.719	0	0	0	0.5
5	0.001	32	50	0.4, 0.6	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
6	0.0001	32	50	0.4, 0.6	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
7	0.001	64	50	0.6, 0.4	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
8	0.0001	64	50	0.6, 0.4	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
9	0.001	64	50	0.5, 0.5	182, 0, 71, 0	0.719	0	0	0	0.5
10	0.0001	64	50	0.5, 0.5	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
11	0.001	64	50	0.4, 0.6	182, 0, 71, 0	0.719	0	0	0	0.5
12	0.0001	64	50	0.4, 0.6	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
					Model-1					
1	0.001	32	50	0.6, 0.4	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
2	0.0001	32	50	0.6, 0.4	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
3	0.001	32	50	0.5, 0.5	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
4	0.0001	32	50	0.5, 0.5	182, 0, 71, 0	0.719	0	0	0	0.5
5	0.001	32	50	0.4, 0.6	182, 0, 71, 0	0.719	0	0	0	0.5
6	0.0001	32	50	0.4, 0.6	182, 0, 71, 0	0.719	0	0	0	0.5
7	0.001	64	50	0.6, 0.4	182, 0, 71, 0	0.719	0	0	0	0.5
8	0.0001	64	50	0.6, 0.4	182, 0, 71, 0	0.719	0	0	0	0.5
9	0.001	64	50	0.5, 0.5	0, 182, 0, 71	0.280	0.280	1.0	0.438	0.5
10	0.0001	64	50	0.5, 0.5	182, 0, 71, 0	0.719	0	0	0	0.5
11	0.001	64	50	0.4, 0.6	182, 0, 71, 0	0.719	0	0	0	0.5
12	0.0001	64	50	0.4, 0.6	15, 167, 4, 67	0.324	0.439	0.943	0.439	0.513
13	0.00001	64	50	0.6, 0.4	29, 153, 19, 52	0.320	0.253	0.732	0.376	0.445
14	0.00001	64	50	0.5, 0.5	57, 125, 24, 47	0.411	0.273	0.661	0.386	0.487
15	0.00001	64	50	0.4, 0.6	53, 129, 14, 57	0.434	0.306	0.802	0.443	0.547

Table: Performance comparison of all models with their hyperparameter settings.


thank
you