Biomatter Assignment

Julija Maldutytė

Introduction

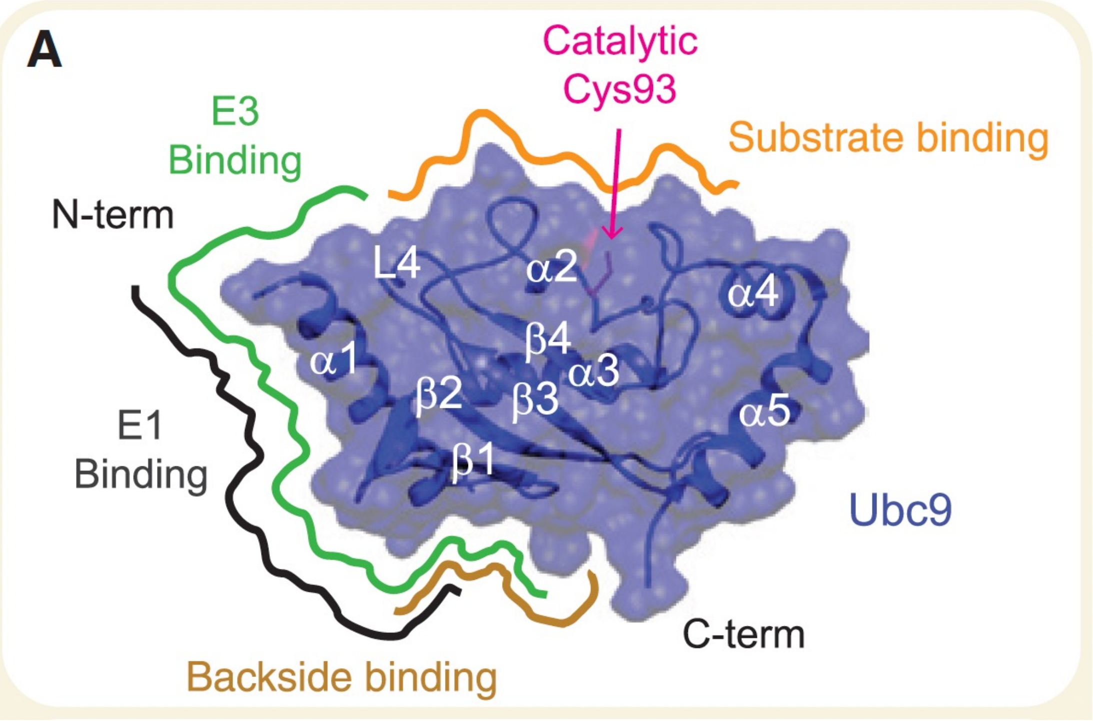
- **Target protein**: SUMO E2 conjugase Ube2i

- **Data to use for training**: 1) Biophysical features; 2) Numerical amino acid (aa) representations; 3) MSA

- **Objective**: using only 96 samples from Ube2i single-point mutant dataset, train/fit a model to predict fitness of unseen Ube2i sequences. Use Spearman's correlation to evaluate results.

Ube2i

- Active site: 93Cys
- Substrate binding: At the catalytic cleft, including 100Asp-101Lys
- E3 and E1 binding: N-terminus
- SUMO binding: backside, opposite the catalytic centre



From Pichler et al., 2017

Approach and choice of model

- Evaluate predictive power of biophysical features and aa rerpesentations separately. See what info can be extracted from MSA data to enhance the model.

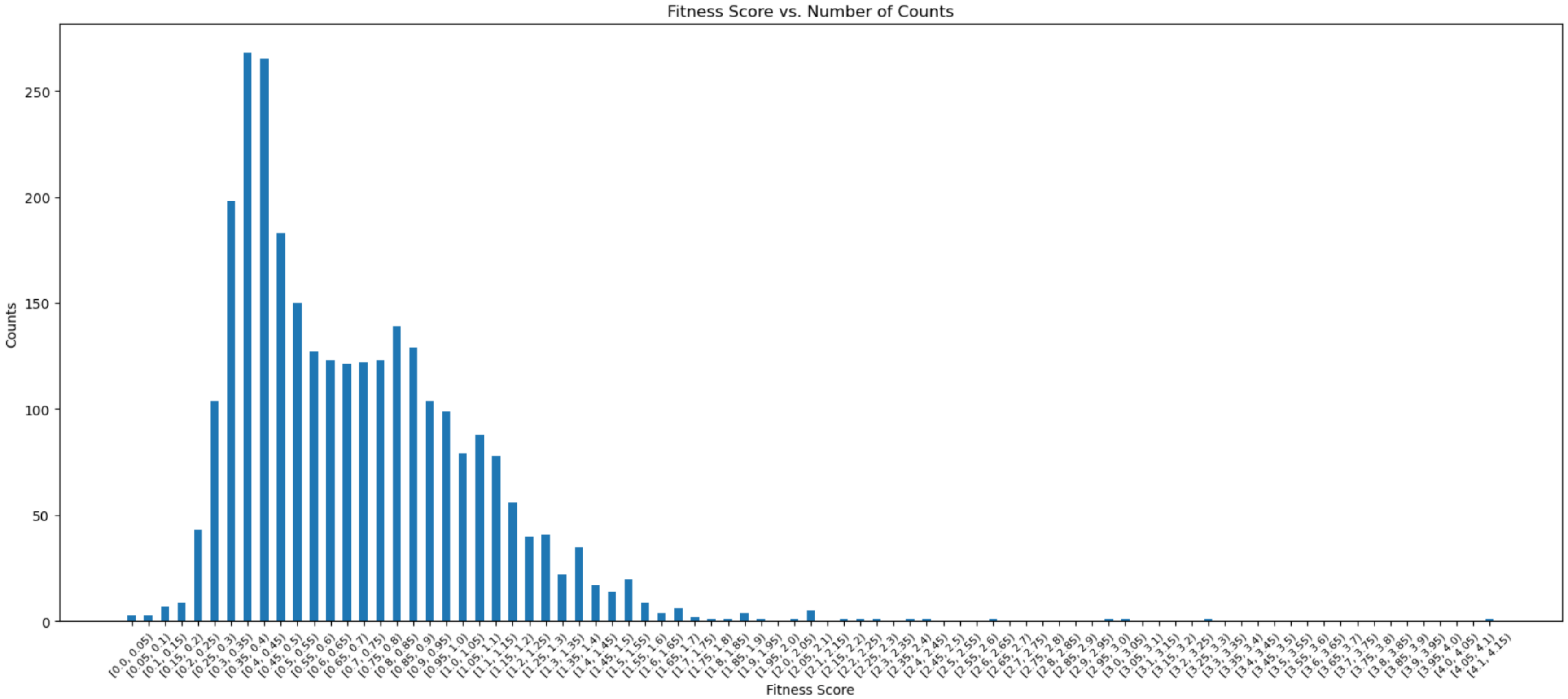
- According to Low-N protein engineering paper by Biswas et al. (2021), if aa representations are indeed semantically-rich, fairly simple models, such as linear regression variations, could be used to predict protein function. I therefore focused on using **ridge regression (RR)** and **random forest regression (RFR)** (for potentially better capturing non-linear relationships) from **scikit-learn** library.

- Even though a preliminary convolutional neural network (CNN) and support vector machine (SVM) were tested, these approaches was not further pursued due to poor initial performance and the low sample restraint (n=96) which does not suit deep learning approaches.

Initial data inspection. The target.

(Jupyter notebook: "1_biophys_feat_RFRselection")

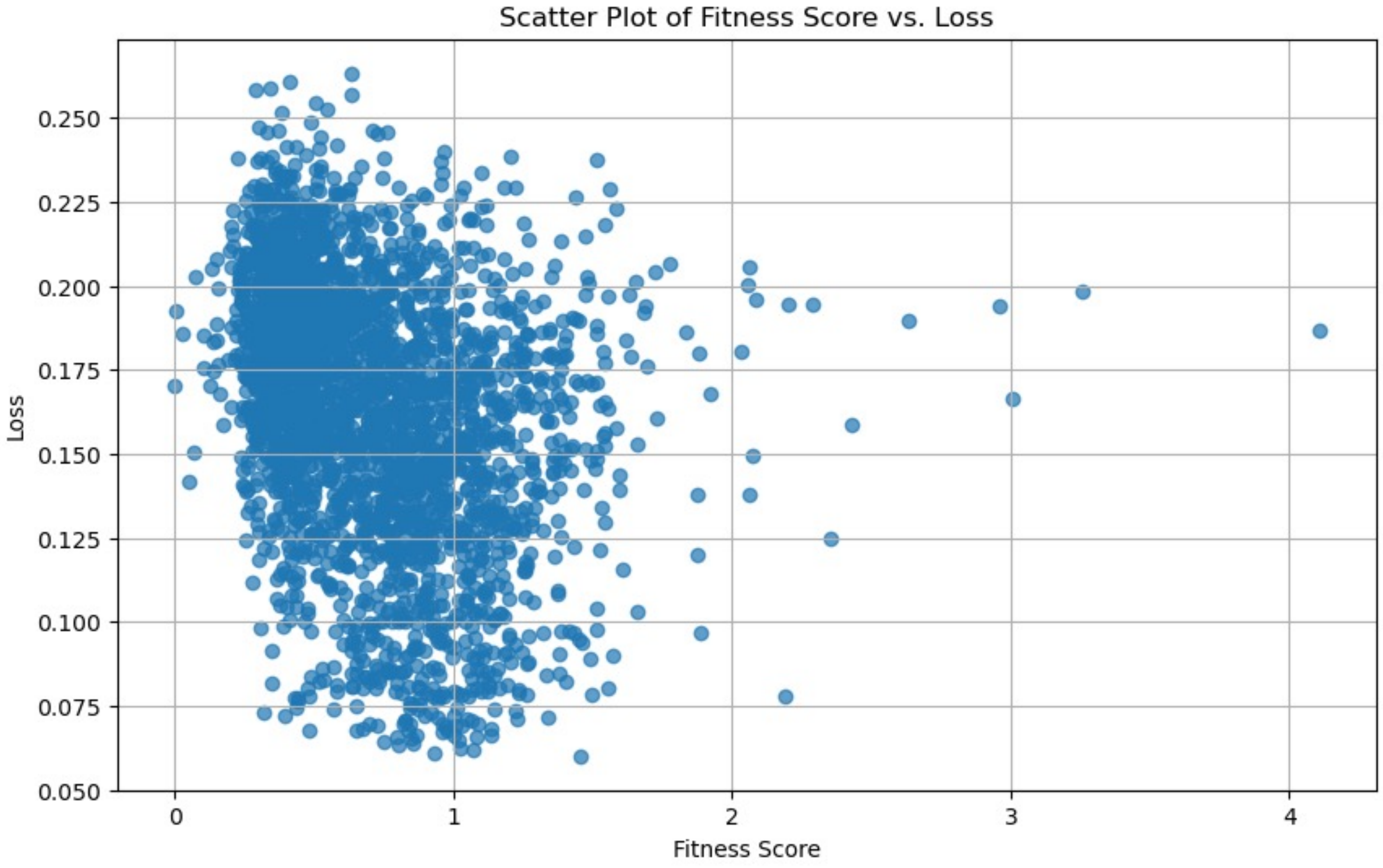
- Target (fitness) data distribution. Most values have 0.2-0.5 fitness. Binomial distribution.



- Data was further split into 8 quantile fitness bins for training. 96 samples were used for training from stratified sampling and the rest for model evaluation. I tried biased sampling to enrich for more high-end fitness score samples but then it skewed predictions of samples with <1 fitness (predicted as higher), which decreased Spearman's correlation, since those samples make up the majority of the data set.

Initial data inspection. The target.

- Correlation with loss values from trained PLM



```
Spearman correlation: -0.3906344323875879, p-value: 1.0947715485300501e-104
```

Low Spearman anti-correlation value suggests the model used in-house had fairly low predictive power which could be enhanced. |-0.3906| was used as a baseline for further improvement.

Biophysical features

(Jupyter notebook: “1_biophys_feat_RFRselection”)

- Feature selection: RFR and RR

	RFR	RFR + top 11 features from RFR	RR all features (standard scaling)
Spearman correlation	0.3899	0.4	0.3606
R^2	0.1159	0.12	0.05299

- From co-variance matrix, many features were seen to correlate. Therefore, top 11 using RFR were used further, which improved the model more.

	RR + top 11* + polynomial^2 (standard scaling)
Spearman correlation	0.39449
R^2	0.1117

*Interestingly, top features from RFR model worked better than top features from RR for an RR model (not shown).

→ RFR slightly outperformed RR when using top 11 biophysical features

Addition of MSA data (Jupyter notebook: “2_MSA”)

- PSSM (position-specific scoring matrix) was extracted from the provided MSA. 158aa*20 generated 3160 new features. Considering that only 20 features change at a time in each single-point mutant (i.e. 1aa), a mean of all features was taken for each sample and added as a single feature to the biophysical features.

- RR outperformed RFR when using PSSM matrix alone as features

	RR + top 11 + polynomial^2 + pssm mean (standard scaling)	**RFR + top11 features + pssm mean**
Spearman correlation	0.4247	0.42
R^2	0.125	0.139

→ Upon addition of pssm mean as a feature, RR started to outperform RFR which was the case with further added features, therefore this model was pursued further

Amino acid numerical representations

(Jupyter notebook: “3_aa_encodings”)

- Flattened in order to feed data into scikit-learn models (only takes 2D data)

- Very high number of features with high skewness and kurtosis (not shown) are generated when flattening (2854, 158*512). This needed to be reduced in order to achieve maximal predictive efficiency and be able to integrate with only 12 other features without outweighing them too much.

- Tested approaches: 512 latent variable (LV) reduction using **PCA**, UMAP and autoencoders (using PyTorch and Keras). Only PCA dimensionality reduction was able to improve model accuracy as compared to using full 80k flattened features (details not shown for simplicity).

Amino acid (aa) numerical representations

- Best performance was achieved when 512 LVs were reduced to 32 using PCA, with RR

	RR representations + polynomialˆ2 (MinMax scaling)	RFR representations + polynomial ˆ2
Spearman correlation	0.42	0.31
R^2	0.135	0.088

Residuals: highest variance around 0.7-0.8 predicted fitness (heavy underestimation of top performers)

Putting everything together

(Jupyter notebook: “4_Final_model”)

- Preprocessing steps for train and test data:

1) Use the following **biophysical features**: FractionCSP3, NumPolarRes, RosettaEnergy, NumApolarRes, MolWt, Hydropathy, Charge, Volume, ProtRDF_all_2.0_3.0, ProtRDF_all_2.0_4.0, NumRotatableBonds.

2) Extract PSMM from **MSA**. Take the mean of the resulting 3160 features to make it into a single feature

3) Flatten **aa numerical representations**, to from (n, 158, 512) format to (n*158, 512), use standard scaling and then PCA to reduce 512 LVs to 32. Reshape back to (n, 158, 512) and then flatten to (n, 158*32), generating 5056 features for each sample.

4) Concatenate the 3 data sources, square the features and use MinMax scaling

RR VS grid search-optimised RFR scores (no scaling for RFR):

Final score

	RR all features + polynomial^2 (MinMax scaling)	**RFR + polynomial ^2**
Spearman correlation	0.508	0.33
R^2	0.1738	0.1

NB: reducing aa representation LVs any further (to try to get it closer to 12 post-flattening) and putting them together with the other 12 features did not enhance model performance.

Summary of the model improvement

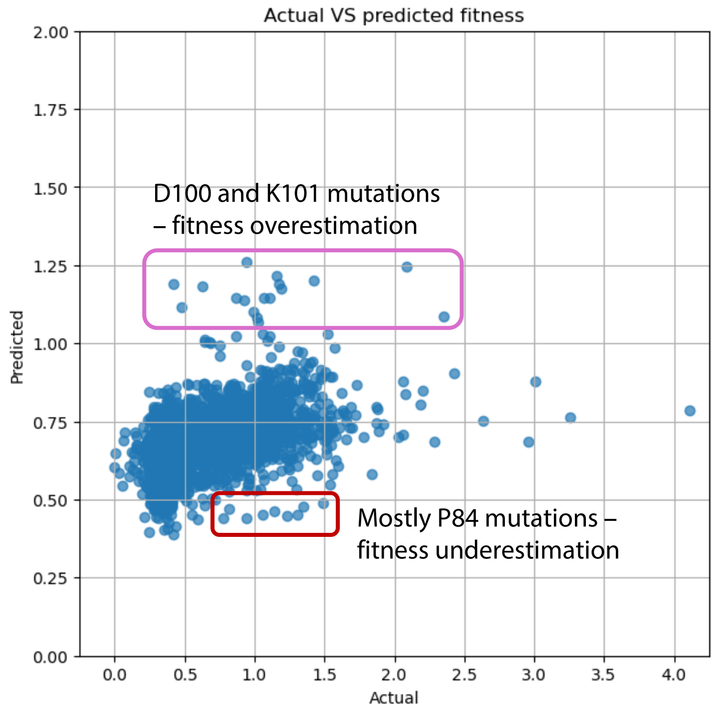
	Biophysical features-only				Biophysical features + PSSM mean	aa representations-only	All features
	RFR	**RFR + top 11 features from RFR**	**RR all features (standard scaling)**	**RR + top 11* + polynomial ˆ2 (standard scaling)**	**RR + top 11 + polynomial ˆ2 + pssm mean (standard scaling)**	**RR representations + polynomial ˆ2 (MinMax scaling)**	**RR all features + polynomial ˆ2 (MinMax scaling)**
Spearman's correlation	0.3899	0.4	0.3606	0.39449	0.4247	0.4248	0.508

Model improvement progression

Model limitations

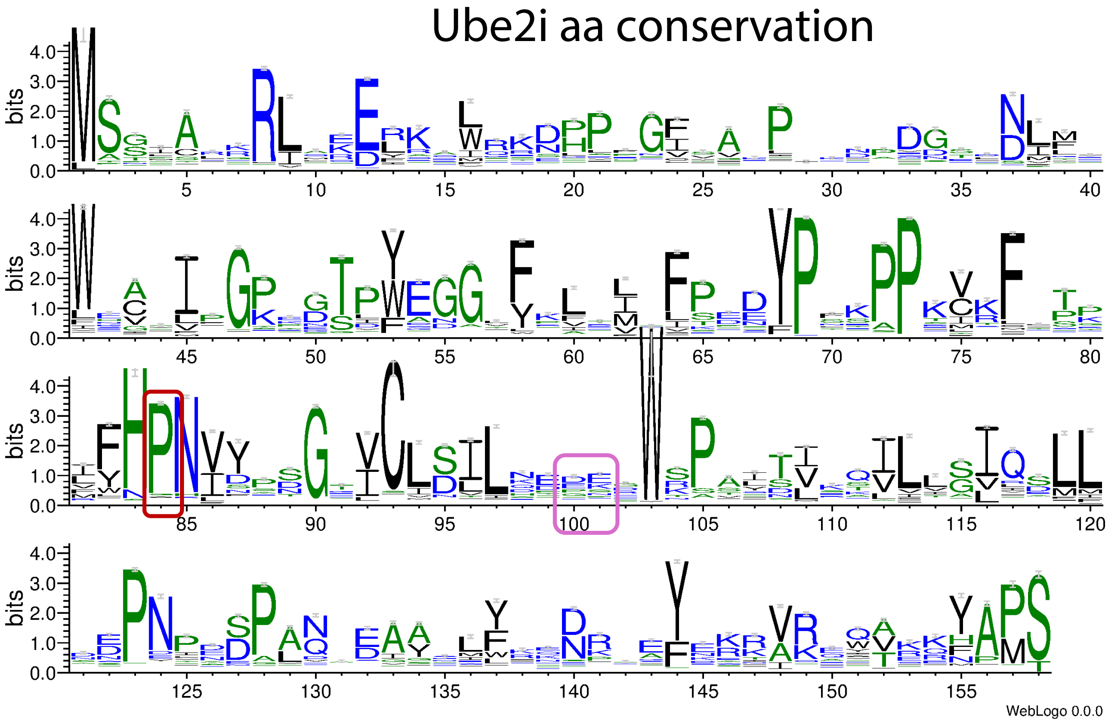
- The trained model was not able to accurately predict values of high fitness mutants (heavy underestimation).
- All mutants predicted to have fitness score >1 are at positions D100 and K101, known to be important for substrate binding (10.1021/bi026861x). They both have low conservation of ˜1/6 (see conservation plot below). Interestingly, D100S, D100I, D100M are all amongst the top 15 mutants with enhanced fitness. This implies that the model learned (or that amino acid encoding contain the information of) the importance of this position but not which amino acid substitutions enhance protein fitness.

Final model:



D100 and K101 mutations – fitness overestimation

Mostly P84 mutations – fitness underestimation


Ube2i aa conservation

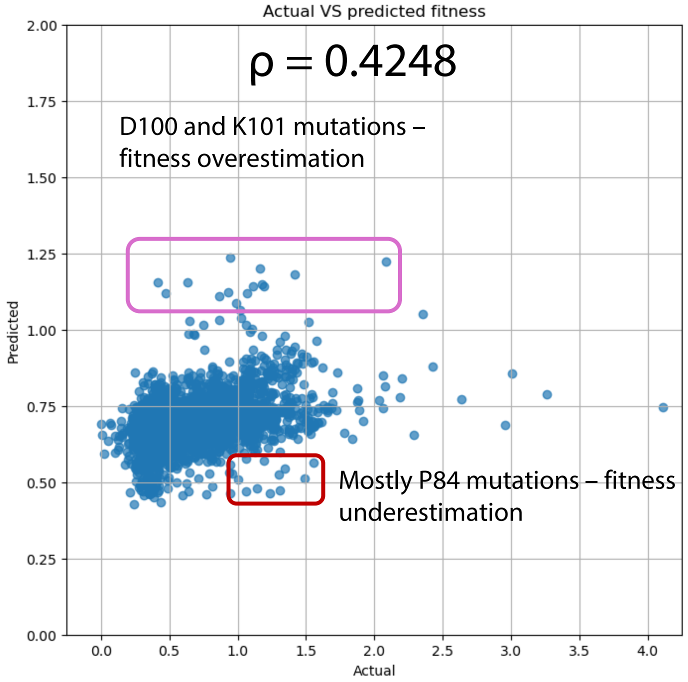
Generated using ConservFold, with which conservation scores were also mapped onto the structure https://colab.research.google.com/drive/1s7N6w2VEjadkJVS9bFzyOLaFZ3uImS0k

Some model insights

Comparing scatter plots where actual test set fitness is plotted against predicted, final model predictions unsurprisingly seem to be driven by aa representation-derived features (5k features VS 12 others):

Comparing an RR model trained on aa representations VS one trained one PSSM-derived features, it seems that aa representations contain a lot of evolutionary information, meaning they are likely derived from an MSA transformer-like PLM

Ridge regression trained only on aa representations

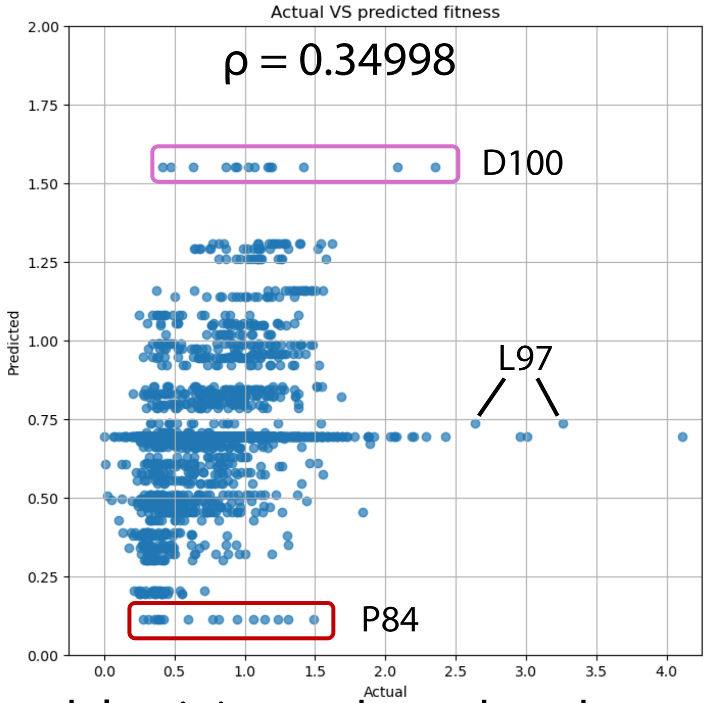


ρ = 0.4248

D100 and K101 mutations – fitness overestimation

Mostly P84 mutations – fitness underestimation

Ridge regression trained only on MSA PSSM-derived features (not mean!)



ρ = 0.34998

D100

L97

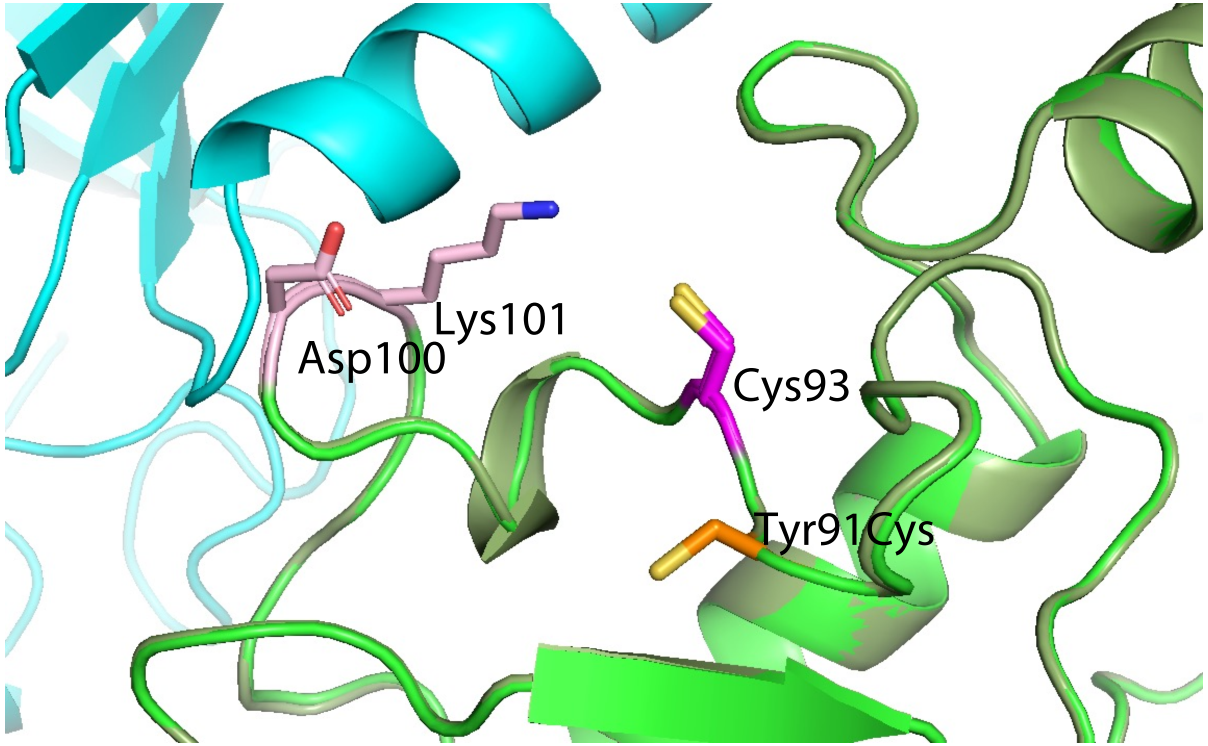
P84

Additionally, taking out "mean_pssm" feature from the final model training and test data does not change prediction of these points.

A closer look at top Ube2i performers

2 highest fitness mutants (T91C, L97C) have a cysteine (potentially catalytic) introduced at a new position. The region where WT catalytic Cys93 and these two mutant Cys lie is a flexible loop near the substrate binding cleft. Such new cysteine positioning likely provides a more favourable conformation for substrate accessibility or stabilizes the SUMO-E2-substrate interaction which normally requires the presence of E3 or cofactors.

In the figure where WT and T91C Ube2i are aligned, the only visible change is Leu94-Leu97 region folding up into a small a-helix in the T91C mutant. This region is known to be helical in complex with substrate so the T91C mutation likely contributes to Ube2i-substrate interaction stabilization. Further insights might be gained from modelling Ube2i, substrate, SUMO and E3 ligase and/or choosing a different substrate.



WT and T91C Ube2i co-folded (AlphaFold2) with substrate p53 and SUMO (not shown)

A closer look at top Ube2i performers

- A5H is at the Ube2i N-terminal region which is known to bind E1 and E3 enzymes in the pathway. The N-terminus of Ube2i has an extended +vely charged patch and so an extra +ve charge might enhance complex stability further.

- Other mutations in the flexible loop at Ube2i's catalytic site (such as E98S and L97T) likely provide favourable chemical moieties for enhanced catalytic activity.

- Rational design interventions could improve the model, e.g. avoid changing amino acids at positions D100 an K101

Potential further steps for better predictions

- Try other, more complex types of models for handling non-linearity, e.g. better-tuned SVM

- Ensemble of models: use a variety of different models and average them

- Larger size of training data increases model accuracy. E.g. using 50% of the given data, the model reaches R^2=0.358 and ρ=0.65. Further sampling optimization.

- Use a different PLM to encode amino acids, such as MSA Transformer or UniRep. Alternatively, PLM used to encode aas could be trained/tailored on a collection of enzymes in the ubiquitin/SUMO and similar pathways, in order to focus its learning on enzyme functionality in the context of interacting proteins.