**Empower deep mutational scanning through machine learning**

Kefan Yang

PI: John Chaput

**Introduction**

Deep mutational scanning is a very practical methods for protein evolution.

However, there are some limitations: first, the library construction methods would have some bias that some of the library members might be ignored from experiment. Second, the number of library members has been limited to around 1000 because of the limited capacity of sequencing and library building.

Machine learning models would facilitate the protein engineering through deep mutational scanning by filling out the missing values with reasonable predictions, figuring out the patterns of existing data and making predictions of different substrates for future experiments.

Previously, machine learning algorithms have been successfully applied to protein engineering for directed evolution. According to Yang et. al [2], a variety of algorithms exists, but no single algorithm is optimal for all tasks. First, linear regression is commonly used as baseline predictors for more complex models. Second, Classification and regression trees, often used in ensemble methods like random forests and boosted trees, are a strong and efficient baseline, especially for small biological datasets. Third, Kernel methods, such as support vector machines, calculate the similarity between inputs. Forth, Gaussian process models, combine kernel methods with Bayesian learning to produce probabilistic predictions. It’s very suitable for a small dataset (<10^3) and successfully used to predict substrates for enzymatic reactions[4]. Last, Deep learning models, being able to extract high-level features from structured inputs, are well suited for large labeled datasets.

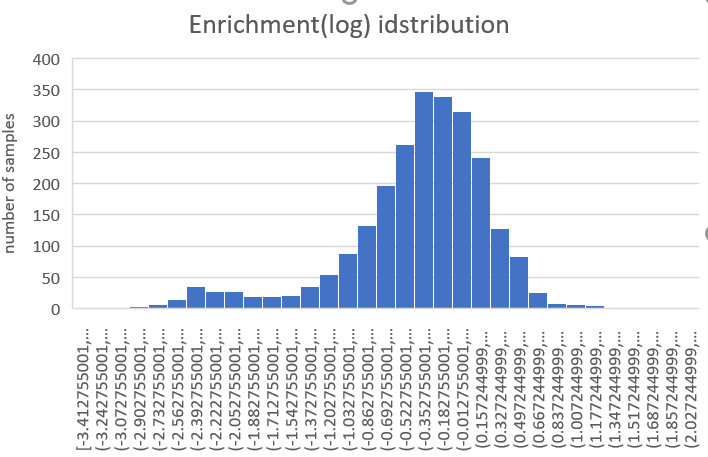
To have a better fitness and prediction, multiple models were tested in this experiment, and best models were used to predict the missing enrichment value.

**Results**

**Training and validation**

After deep mutational scanning experiment for TNA engineered polymerase, the variants were divided into two groups (“enriched” and “depleted”) for classification. The dataset contains 912 variants. To have enough confidences, variants that have no more than 10 redundancies will not be considered in the table. Thus there are 886 qualified samples and 26 missing samples.

The input features include position information using one-hot-encoding, T-scale descriptors of wild-type amino acids and variants amino acids information.

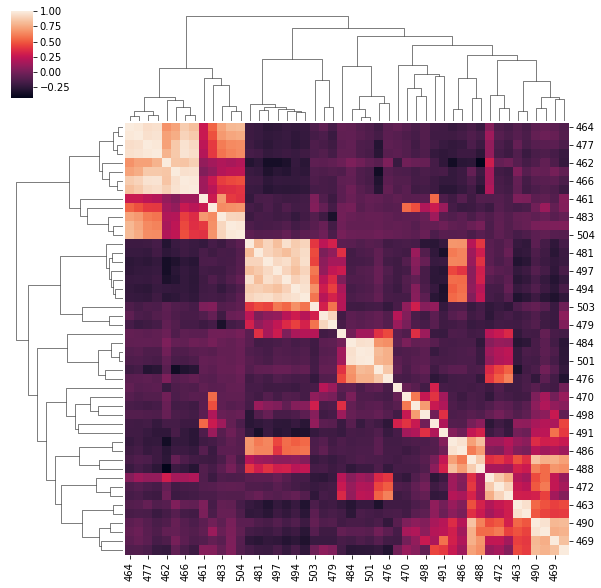
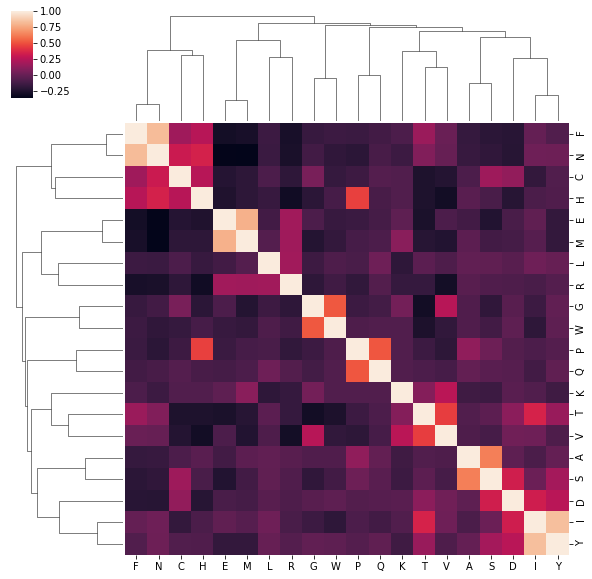
**Figure 1.** TNA dataset. A) deep mutional scanning table for classificaiton. Green spots are the missing data, blue spots are enriched group while white spots are depleted group. B) distribution of log enrichment data.

**Correlation cluster heatmaps**

Cluster heatmaps are commonly used in biology and related fields to reveal hierarchical clusters in data matrices. Heatmaps visualize a data matrix by drawing a rectangular grid corresponding to rows and columns in the matrix, and coloring the cells by their values in the data matrix.

To have a deeper understanding of the dataset, Pearson correlation coefficient between each column was calculated. The matrix was clustered and made into heatmaps.

Correlations of position were shown in **Figure 2a**. Some positions keep the same pace on the same single mutational walk. And there are about 4~5 clusters in total. Also correlations of amino acids were shown in **Figure 2b**. amino acids substitution correlation is unique and different from the well-known substitution matrix BLOSUM62. Probably because it’s the specific protein region and the dataset is comparably small.

**Figure 2.** Correlation shown by cluster heatmap. A) position based cluster heatmap. B) amino acids based cluster heatmap.

**Model performance**

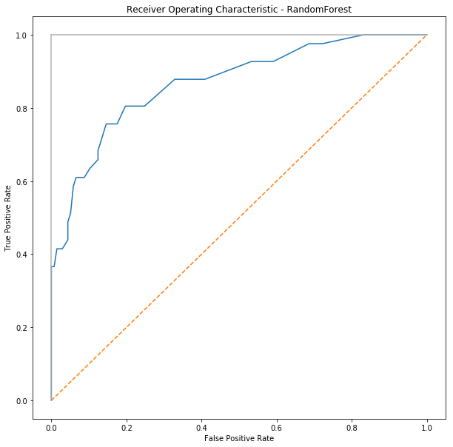
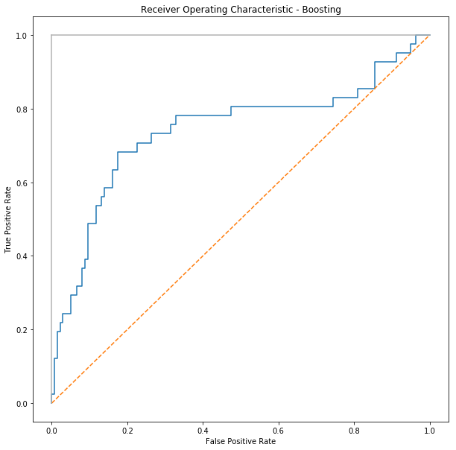
A variety of machine learning models were employed to predict missing samples and find out patterns for the deep mutational scanning experiment. Random forest (RF) classifer, Multilayer perceptron (MLP) classifier, boosting classifier and GassianNB classifier were trained by TNA dataset.

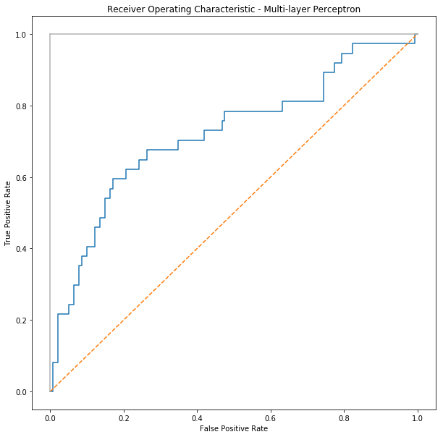
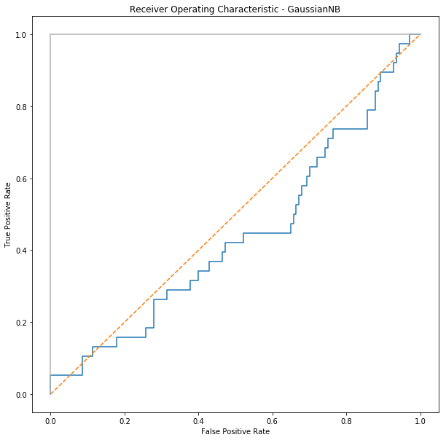
The performance of the methods was assessed by a 10- fold cross validation on the TNA dataset. The classifiers of RF and SVM yielded a similar overall accuracy (Q2) of the TNA dataset. Random forest outperformed in accuracy, F1-score and MCC to the other models. The better performance of the RF classifier was probably due to the imbalance of the two classes in the TNA dataset and may indicate that the RF algorithm was better at accommodating this imbalance than the rest of models. To further investigate the robustness of the classifiers, receiver operating characteristic (ROC) curves were plotted based on 10-fold cross validation tests on the TNA dataset (Figure 2). The values for the area under the curve for the RF, MLP, Boosting and GaussianNB classifiers were 0.85, 0.74, 0.80 and 0.39, respectively. These results indicate that the RF classifiers could be used to predict which mutations were beneficial and that the RF classifier was a better performer than the other models.

Also, the confusion matrix shows that random forest model performs best. It has very high precision and recall that the predictions are very faithful, shown in **Figure 4**.

**Table 1: Classification performance on the TNA dataset**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Accuracy (Q2) | F1-score | MCC |
| Random Forest | 0.85 (+/- 0.07) | 0.84 | 0.59879 |
| MLP | 0.75 (+/- 0.06) | 0.78 | 0.357795 |
| Boosting | 0.79 (+/- 0.06) | 0.82 | 0.455485 |
| GaussianNB | 0.41 (+/- 0.04) | 0.44 | -0.12006 |

**Figure 3.** Receiver operating characteristic curves for the prediction using the RF, MLP, GaussianNB and boosting methods. The curves were obtained from the 10-fold cross validation test on the TNA dataset.

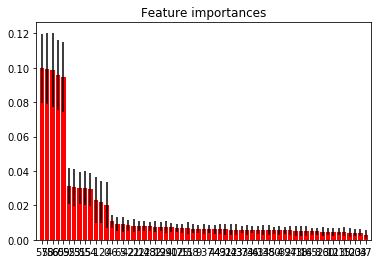
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a |  | Predicted | |  | b |  | Predicted | |
|  | RF | 0 | 1 |  |  | Boosting | 0 | 1 |
| Real | 0 | 137 | 4 |  | Real | 0 | 129 | 12 |
| 1 | 20 | 17 |  | 1 | 22 | 15 |
|  |  |  | |  |  |  |  |  |
| c |  | Predicted | |  | d |  | Predicted | |
|  | MLP | 0 | 1 |  |  | GaussianNB | 0 | 1 |
| Real | 0 | 126 | 15 |  | Real | 0 | 51 | 90 |
|  | 1 | 26 | 11 |  | 1 | 18 | 19 |

**Figure 4.** Confusion matrix of a) random forest, b) boosting, c) multilayer perceptron and d) GassianNB.

**Feature importance**

To figure out the importance of features training the model, we used forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importance of the forest, along with their inter-trees variability.

To our surprise, the amino acids descriptors contribute most to the random forest models whereas the one-hot encoding features of position have much less contributions.



**Figure 5.** Feature importance ranks. Top 10 features come from amino acids descriptor.

**Conclusion**

Machine learning models are helpful for deep mutational scanning experiment. It could give predictions with accuracy as high as 85% and F1 score of 0.84.

**Discussion**

Machine learning could help deep mutational scanning through several ways:

1. Fill in the missing data with models trained by existing data.
2. Enhance the library member size of DMS. Originally, each DMS library was limited to ~1000 members with the limitation of sequencing redundancies. With the help of machine learning. We could explore 20% more library members or more.
3. Find out patterns of deep mutational scanning.
4. Make some potential predictions for substrates that have not been tested through deep mutational scanning experiment.

**Methods**

**Feature Vector**

As a feature vector x for the protein, we used a precomputed feature vectors for amino acids. Specifically, we defined a feature vector of a variant by concatenating the feature vectors of amino acids. In this experiment T-scale descriptor was used for training.

**Classification evaluation**

1. **Accuracy**

Accuracy is the proportion of true results among the total number of cases examined.

Accuracy = (TP+TN)/(TP+FP+FN+TN)

Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed or No class imbalance.

1. **AUC**

AUC is the area under the ROC curve. AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes.

We have got the probabilities from our classifier. We can use various threshold values to plot our sensitivity(TPR) and (1-specificity)(FPR) on the cure and we will have a ROC curve.

Where True positive rate or TPR is just the proportion of trues we are capturing using our algorithm.

Sensitivty = TPR(True Positive Rate)= Recall = TP/(TP+FN)

and False positive rate or FPR is just the proportion of false we are capturing using our algorithm.

1- Specificity = FPR(False Positive Rate)= FP/(TN+FP)

AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values. So, for example, if you as a marketer want to find a list of users who will respond to a marketing campaign. AUC is a good metric to use since the predictions ranked by probability is the order in which you will create a list of users to send the marketing campaign.

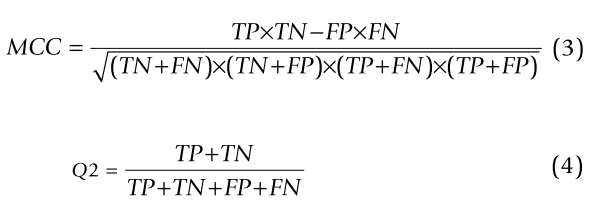
Another benefit of using AUC is that it is classification-threshold-invariant like log loss. It measures the quality of the model’s predictions irrespective of what classification threshold is chosen, unlike F1 score or accuracy which depend on the choice of threshold.

1. **F1 score**

The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall.

1. **Matthews correlation coefficient (MCC)**

the Matthews correlation coefficient (MCC). The Q2 and MCC were calculated as follows:



**Machine learning methods**

The Random forest is an ensemble machine learning methodology originated by Leo Breiman. The basic idea of ensemble learning is to boost the performance of a number of weak learners via a voting scheme, where a weak learner can be an individual decision tree, a single perceptron/sigmoid function, or other simple and fast classifier. Regarding RF, its hall- marks include bootstrap re-sampling, random feature selection, in-depth decision tree construction and out-of-bag error estimates.

**Reference:**

Tian, J., Wu, N., Chu, X., & Fan, Y. (2010). Predicting changes in protein thermostability brought about by single- or multi-site mutations. BMC Bioinformatics, 11.

Fowler, D. M. & Fields, S. Deep mutational scanning: a new style of protein science. Nat. Methods 11, 801–807 (2014).

Yang, Kevin K., Zachary Wu, and Frances H. Arnold. "Machine-learning-guided directed evolution for protein engineering." Nature methods 16.8 (2019): 687-694.

Nikoomanzar, Ali, Derek Vallejo, and John C. Chaput. "Elucidating the determinants of polymerase specificity by microfluidic-based deep mutational scanning." ACS synthetic biology (2019).

Mellor, Joseph, et al. "Semisupervised Gaussian process for automated enzyme search." ACS synthetic biology 5.6 (2016): 518-528.

Houlihan, Gillian, Sebastian Arangundy-Franklin, and Philipp Holliger. "Engineering and application of polymerases for synthetic genetics." *Current opinion in biotechnology* 48 (2017): 168-179.

Houlihan, Gillian, Sebastian Arangundy-Franklin, and Philipp Holliger. "Exploring the chemistry of genetic information storage and propagation through polymerase engineering." *Accounts of chemical research* 50.4 (2017): 1079-1087.

Anosova, Irina, et al. "The structural diversity of artificial genetic polymers." *Nucleic acids research* 44.3 (2015): 1007-1021.

Dunn, Matthew R., et al. "DNA polymerase-mediated synthesis of unbiased threose nucleic acid (TNA) polymers requires 7-deazaguanine to suppress G: G mispairing during TNA transcription." *Journal of the American Chemical Society* 137.12 (2015): 4014-4017.

Riesselman, A. J., Ingraham, J. B., & Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. Nature Methods, 15(10), 816–822.

Tian, F., Zhou, P., & Li, Z. (2007). T-scale as a novel vector of topological descriptors for amino acids and its application in QSARs of peptides. Journal of Molecular Structure, 830(1–3), 106–115.