ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

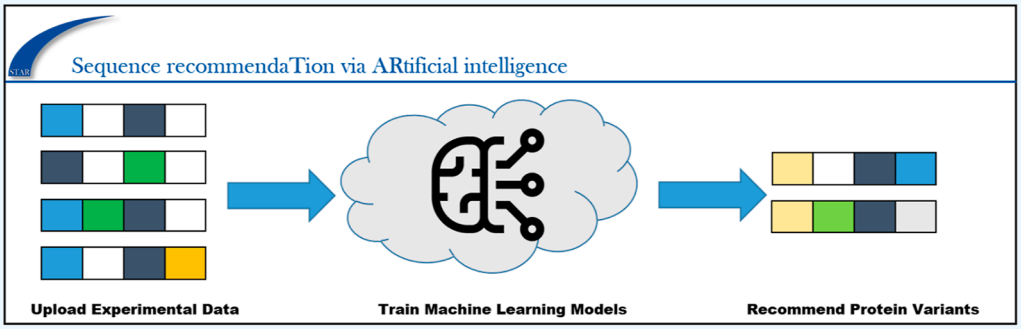
STAR: A Web Server for Assisting Directed Protein Evolution with Machine Learning

Likun Yang, Xiaoli Liang, Na Zhang,* and Lu Lu*

Cite This: *ACS Omega* 2023, 8, 44751−44756

Read Online

ACCESS | Metrics & More | Article Recommendations



ABSTRACT: Protein engineering has made significant contributions to industries such as agriculture, food, and pharmaceuticals. In recent years, directed evolution combined with artificial intelligence has emerged as a cutting-edge R&D approach. However, the application of machine learning techniques can be challenging for those without relevant experience and coding skills. To address this issue, we have developed a web-based protein sequence recommendation system: STAR (Sequence recommendaTion via ARtificial intelligence). Our system utilizes Bayesian optimization as its backbone and includes a filtering step using a regression model to enhance the success rate of recommended sequences. Additionally, we have incorporated an in silico-directed evolution approach to expand the exploration of the protein space. The Web site can be accessed at https://www.FindProteinStar.com/.

INTRODUCTION

Protein engineering is the process of designing and creating new variants with improved properties by manipulating their amino acid sequences. This technique has already yielded significant results in the fields of nanotechnology, agriculture, and medicine.[1,2] Protein-directed evolution is one of the most commonly used techniques in protein engineering;[3] it mimics natural evolution by iteratively introducing mutations and selecting for beneficial variations until the desired level of improvement is achieved. However, the vastness of the protein space and the scarcity of functional proteins pose significant challenges to directed evolution, resulting in potentially suboptimal outcomes.[2]

Machine learning integrated into directed evolution is a new paradigm for protein engineering.[1,2,4] This technique has shown great success in various applications. For instance, Romero et al.[5] successfully designed thermostable chimeric cytochrome P450 enzymes. In another study, Greenhalgh et al.[6] improved in vivo fatty alcohol production by engineering acyl-acyl carrier protein reductase. Wu et al.[4] utilized machine learning-guided directed evolution to identify higher fitness variants. It also should be mentioned the success of protein generative models, such as UniRep,[7] TAPE,[8] ProGen,[9] ProGPT2,[10] and ESM2.[11]

Bayesian optimization (BO) is a primary method used to address black-box function optimization problems. It has been shown to be a powerful tool in synthetic biology, including applications such as protein engineering[12−14] and biosynthetic pathway optimization.[15,16]

In this paper, we propose a web-based machine learning-assisted directed evolution platform: STAR (Sequence recommendaTion via ARtificial intelligence), which can be accessed at https://www.FindProteinStar.com/. There are some open-source frameworks, such as PyPEF,[17] BO-EVO,[14] and ftMLDE,[18] that aim to assist researchers in using machine learning for directed evolution. However, using such frameworks requires researchers to have certain coding skills, which can be a barrier to the widespread adoption of machine learning in the field. STAR aims to make the process more accessible to a wider range of researchers. Specifically, our work makes two contributions:

Received: August 4, 2023
Revised: October 10, 2023
Accepted: October 12, 2023
Published: November 14, 2023

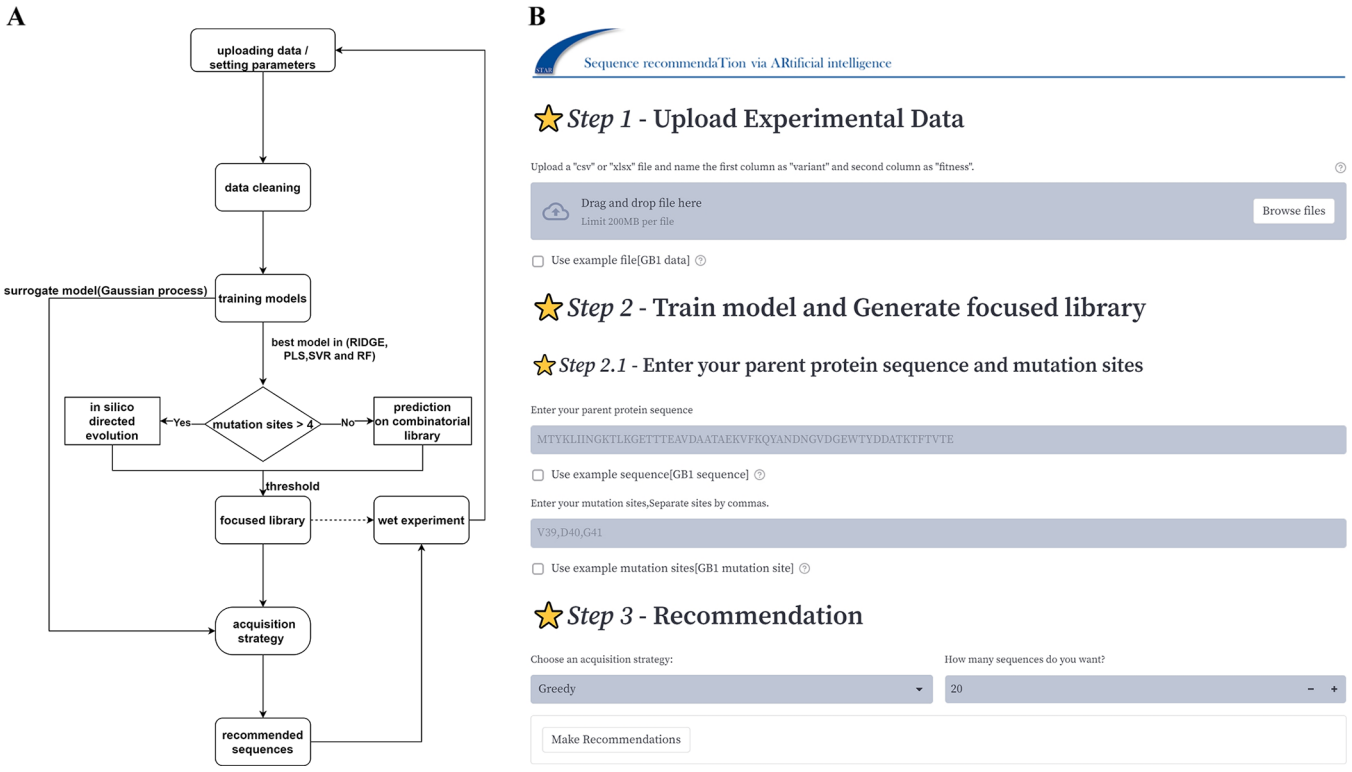


Figure 1. (A) Workflow of STAR, where the dashed line indicates that the user can directly use the sequences in the focused library. (B) Main interface of the web.

(1) A web-based machine learning-assisted protein-directed evolution platform, which allows researchers to use machine learning in protein engineering without any coding.
(2) Integrating in silico-directed evolution (iDE) into BO enables researchers to mutate more sites and explore a larger protein space.

METHODS

Bayesian Optimization. Bayesian optimization is an iterative technique widely used for optimizing expensive black-box objective functions.[19,20] Formally, it can be formulated as a maximization problem with an objective function f defined on a search space A:

$$\max_{x \in A} f(x)$$

In the first iteration, a surrogate function f^* is constructed by using initial observations to approximate f. Then, the next querying point(s) are selected using an acquisition function, and one iteration of Bayesian optimization is completed by updating the surrogate function f^* with the newly queried data. This process continues until predetermined criteria are met. In the context of machine learning-assisted protein engineering, the objective function f represents the relationship between sequence and fitness, which is measured through wet experiments. The surrogate function f^* is a probabilistic model trained on experimental data.

Workflow. As shown in Figure 1A, after uploading experimental data, we cleaned them to meet the machine learning standards. We then train two regression models: the first for initial filtering [from Lasso, Ridge, Support Vector Regression (SVR), and Random Forest (RF)] and the second, Gaussian process (GP), as a surrogate function for BO. To reduce search space, we create a "focused library" based on the number of mutation sites n: for $n \leq 4$ sites, a combinatorial library is constructed; for $n > 4$ sites, iDE is employed. We acquire recommended sequences using Greedy, Upper Confidence Bound (UCB), or Probability of Improvement (PI) acquisition strategies. The details of the process are explained in the following sections of this paper.

Data Cleaning. Data cleaning is a crucial step in the machine learning process as it addresses the presence of duplicates, missing data, and outliers, which can have a negative impact on the model's performance. To handle missing data, we remove it from the data set. For deduplication, the user can choose one of five strategies: select the first/last sequence, the sequence with the maximum/minimum value, or take the average value of duplications as the training value. We have decided not to handle outliers and leave this decision to the user's discretion, considering that outlier detection methods can present challenges when applied to protein-fitness data.

In the model training step, we employed Lasso, Ridge, SVR, and RF regression models as candidate models to predict protein fitness. To optimize the performance of each model, we utilized a grid search method to tune the hyperparameters and conducted 5-fold cross-validation to evaluate their performance. To obtain the uncertainty of the predictions for subsequent acquisition strategies, we selected the GP as the surrogate model for BO. We also applied hyperparameter tuning to the GP model to enhance its performance. Table 1 provides a summary of the parameter grid used for tuning the models.

In this study, we have implemented three different types of sequence representations, namely, one-hot encoding, phys-

https://doi.org/10.1021/acsomega.3c04832
ACS Omega 2023, 8, 44751−44756

Table 1. Different Regression Models and Their Respective Hyperparameter Search Spaces, with Five-Fold Cross-Validation Utilized to Identify the Optimum Value

regression method	parameter grid
Lasso	alpha: [0.01, 0.1, 1, 10, 100, 500, 1000, 5000, 10,000]
Ridge	alpha: [0.01, 0.1, 1, 10, 100, 500, 1000, 5000, 10,000]
SVR	C: [0.01, 0.1, 1, 10, 100, 500, 1000, 5000, 10,000]; gamma: [10, 100, 1000, 500, 1, 0.1, 0.01, 0.001, 0.0001]
RF	n_estimators: [100, 200, 300, 400, 500]; max_depth: [2, 3, 4, 5, 6, 7, 8, 9, 10]; max_features: [sqrt, log 2]
GP	alpha: [1×10^{-10}, 1×10^{-8}, 1×10^{-6}, 1×10^{-4}, 1×10^{-3}, 0.01, 0.1, 1, 10], kernel: [RBF, Matern, DotProduct]

icochemical encoding, and learned encodings. We use learned encodings as the default value and allow users to select from any of the three as the encoding method.

One-hot encoding is a straightforward approach for encoding categorical data and represents a protein sequence as a vector of binary values, where each element of the vector represents the presence or absence of a specific amino acid.

Additionally, protein sequences can also be represented by their physicochemical properties, such as hydrophobicity, mutability, and charge. Furthermore, higher-level properties, such as secondary and tertiary structures can also be incorporated into the embedding.[21] In this work, we have chosen the Georgiev[22] encoding technique, which is a physicochemical representation derived from the amino acid index database.[23] Wittmann et al.[18] have demonstrated that Georgiev encoding achieved similar performances as learned encoding on the protein G domain B1 (GB1) data set.

In recent years, transformer-based natural language pretrained models, such as BERT and GPT, have achieved remarkable success in natural language processing tasks.[24] These models have demonstrated the ability to capture complex and abstract features of text data, leading to improved performance on various *natural language processing* tasks. Inspired by the success of natural language pretrained models, researchers have begun to explore the application of similar pretraining techniques to protein sequences. For this study, we selected ESM-1v24 and ESM211 as the encoding strategies for protein sequences as they have demonstrated superior performance in predicting protein-related tasks.

Functional proteins are rare within the large space of possible sequences; thus, most variants in the sample space exhibit low or near-zero property values. To increase the efficiency of BO, a common approach is to first filter out less promising sequences using techniques such as classification,[6] outlier detection,[12] and zero-shot.[18] We employed regression models to create a reduced search space, termed a "focused library". Users can choose a certain threshold of predicted values to construct this focused library.

However, creating such a focused library poses a significant computational challenge as the number of possible sequences increases exponentially with the number of mutation sites. For example, when there are 5 mutation sites, the number of possible sequences can reach up to 3,200,000 (20^5). Therefore, we incorporated in silico-directed evolution into BO.

When the number of mutation sites is less than or equal to 4, we construct a combinatorial library by performing a full permutation of 20 amino acids at the mutation sites. When the number of mutation sites exceeds 4, we will use iDE methods to sample sequences. The iDE is essentially a Metropolis–Hastings Markov chain Monte Carlo algorithm, which is described and presented in these 2 papers.[17,25] After the initial variant is randomly selected and accepted, subsequent substitutions are determined by the Metropolis–Hastings criterion. The acceptance probability p is defined as follows:

$$p = \min\left(1, \frac{\exp(\Delta y)}{T}\right)$$

where Δy is the difference between the predicted fitness of the newly generated sequence and the last accepted one and T is the temperature, which controls the balance between exploration and exploitation of the searching space. After the acceptance probability p is determined, a random number between 0 and 1 is generated. If this number is less than or equal to the acceptance probability, the proposed mutation is accepted. Otherwise, it is rejected, and the previous variant is used as the starting point for the next iteration.

In addition, we have also implemented two additional features: (1) allowing users to provide a list of mutation positions and/or permitted amino acids for mutations, with all mutations being selected from this list and (2) restricting the number of mutations compared to the wild-type, also known as the trust radius.[25] This is done by setting the predicted fitness of the sequence that has more mutations than the input trust radius to negative infinity, thus forcing rejection of the proposal.

In this work, we implemented three different acquisition strategies, including Greedy, UCB, and PI.

$$\mathrm{UCB}(x) = \mu(x) + \beta\sigma(x)$$

$$\mathrm{Greedy}(x) = \mu(x)$$

$$\mathrm{PI}(x) = \begin{cases} \Theta(z), & \sigma(x) > 0 \\ 1, & \sigma(x) = 0 \text{ and } \gamma(x) > 0 \\ 0, & \sigma(x) = 0 \text{ and } \gamma(x) \le 0 \end{cases}$$

where $\mu(x)$ and $\sigma(x)$ are the surrogate model-predicted mean and uncertainty at point x and β is a constant number. $\gamma(x) = \mu(x) - f^* + \delta$; $z(x) = \frac{\gamma(x)}{\sigma(x)}$; Θ is the cumulative distribution function; f^* is the current maximum objective function value; and δ is a constant number.[26]

Web Usage and Implementation. The Web site offers a user-friendly interface (Figure 1B) and ease of use. Users can simply upload their training data, wild-type sequence, and intended mutation sites when using default parameters. Additionally, we provide a wide range of flexibility for users to adjust the output by modifying parameters.

Summary of adjustable parameters:

(1) Trust radius—a limit on the number of mutations compared to the wild-type sequence, with a default value of 12.

(2) Allowed mutation sites—the user can specify specific positions of the sequence that can be mutated; otherwise, the whole sequence is allowed by default.

(3) Allowed mutation amino acids—the user can specify a subset of the 20 standard amino acids that can be used for mutations; otherwise, all are allowed by default.

(4) Temperature (T)—controls the balance of exploration and exploitation in the Metropolis–Hastings algorithm, with a default value of 0.01.

https://doi.org/10.1021/acsomega.3c04832
ACS Omega 2023, 8, 44751−44756

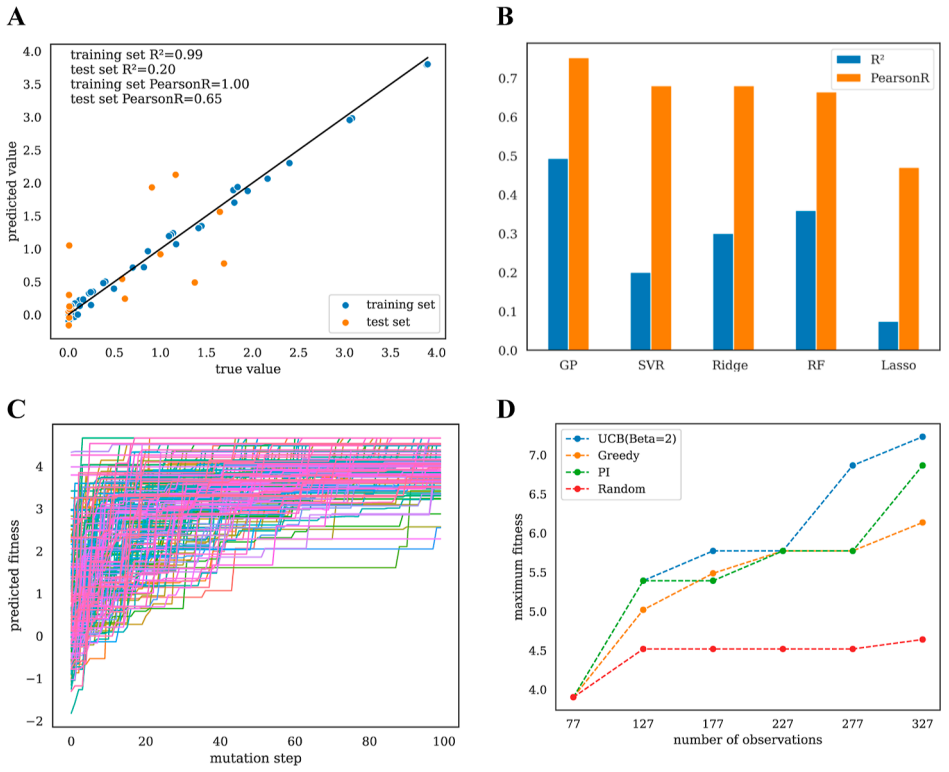


Figure 2. (A,B) Performance of the regression models on the initial batch of 77 samples. (A) SVR performance on the data set using ESM2 protein sequence encoding. The model achieved an *R*-squared value of 0.20 and a Pearson's r of 0.65 on a test set of 16 samples. The model's performance will be displayed on our Web site after training to assist users in selecting appropriate models for their specific needs. (B) Performance comparison of different machine learning models on the data set. GP achieved the highest scores (R^2 = 0.493, Pearson's r = 0.752), followed by RF (R^2 = 0.359, Pearson's r = 0.664). SVR, Ridge, and Lasso performed relatively poorly, with R^2 values ranging from 0.2 to 0.074 and Pearson's r values ranging from 0.47 to 0.68. (C) Example of iDE on the GB1 data set using a restricted set of mutation sites (V39, D40, G41, and V54). A total of 200 evolutionary paths were simulated with 100 trial mutations per path, using the trained SVR model (the same as in Figure 2A) as the energy function. This result serves as an illustrative example to demonstrate the efficacy of iDE, and it has led to higher predicted values. (D) Results of conducting BO on the GB1 data set. The *x*-axis shows the accumulated number of experimental points with a total of six rounds of evolution performed. In each round, an additional 50 samples were added to the batch size of 77 in the initial round. The *y*-axis displays the average maximum experimental value obtained over four simulations for each set of accumulated samples.

(5) Number of evolutionary trajectories—determines the number of independent evolutionary trajectories, with a default value of 150.
(6) Number of evolutionary steps per trajectory—determines the number of iterations within each trajectory, with a default value of 100.
(7) Threshold for generating the focused library—the user can specify a threshold for selecting sequences to form the focused library, with the default value being the top 10%.
(8) Acquisition strategy—the user can choose one from Greedy, UCB, and PI, with UCB as the default value.
(9) Encoding method—the user can choose one from one-hot, Georgiev, ESM-1v, and ESM2 as the encoding method for sequences, with ESM2 as the default value.

The frontend of the Web site was built using Streamlit, and the backend was written in Python language. For machine learning modeling, we utilized the Sklearn library.

RESULTS AND DISCUSSION

GB1 Example. The GB1 data set consists of 149,361 experimentally determined fitness measurements for 160,000 (i.e., 20^4) possible variants of the B1 domain of protein G. The fitness is determined by the protein's ability to bind to the fragment crystallizable domain of immunoglobulins. The data set was generated by Wu et al.[27] through saturation mutagenesis at four carefully chosen residue sites (V39, D40, G41, and V54). This data set has been used by many to demonstrate the feasibility of their machine learning approach in the protein design process.[4,12,14,18]

Our process was evaluated by using the GB1 database. Initially, 77 samples from the database were selected by performing single-residue single-site saturation mutagenesis at four positions based on the wild-type sequence. This approach ensures that all 20 amino acids occur at least once in the selected sites. The selection of the initial batch of samples can be random in principle, but we recommend a strategy that maximizes the amount of information obtained within the limited experimental budget. Subsequently, we selected SVR as the first-step filtering regression model. The acquisition strategies employed include UCB with a beta value of 2.0, PI, and Greedy, and for completeness, random sampling was also tested. We selected a relatively small batch size of 50 for a total of 5 rounds of iteration. It should be noted that the batch size and number of iterations can affect the final results, and users should make their selections based on their specific circumstances. Four independent tests were conducted, and the results are shown in Figure 2D. It can be observed that after a total of 327 (327/160,000 = 0.020%) experimental points were selected, using the UCB sampling strategy, we

https://doi.org/10.1021/acsomega.3c04832
ACS Omega 2023, 8, 44751−44756

identified sequences with a mean fitness value of 7.23. It should be noted that the maximum value in the entire data set is 8.76 and that the fitness values of the majority of sequences are close to 0. For additional experimental details, we refer readers to the Jupyter notebook available in our GitHub repository.

PhoQ Example. We further assessed the general applicability of STAR using the PhoQ[28] fitness landscapes. The PhoQ data set consists of 140,517 experimentally determined fitness measurements for 160,000 (i.e., 20^4) possible variants. The empirical fitness landscape reflects the interaction between PhoQ mutants and their substrate PhoP. Using the UCB sampling strategy, we conducted 5 rounds of iterations with a batch size of 300. In addition to the initial 70 samples, a total of 1570 samples were selected. In all 5 experiments, the optimal variant was successfully identified.

Despite the promising results demonstrated by our STAR system, we acknowledge that a comparative analysis of our model with existing models could bring additional validation. Such comparison, however, is challenging due to factors like initial values, batch size, and iteration rounds inherent to Bayesian optimization methodologies. We also note that similar comparisons are often not provided in the literature introducing Bayesian optimization.[12−14]

CONCLUSIONS

We have developed a web-based protein sequence recommendation platform, named "STAR", that assists users in protein design through the use of machine learning. With a user-friendly interface, the platform offers ease of use to its users. Additionally, it is the first web-based platform to integrate Bayesian optimization and in silico-directed evolution for protein sequence recommendation, and it can contribute to the advancement of the protein design community.

ASSOCIATED CONTENT

Data Availability Statement

Users can access the STAR application free of charge at https://www.findproteinStar.com/. The testing GB1 data set and a demonstrative Jupyter notebook are available at https://github.com/likun1212/findproteinStar. Additionally, to aid in the reproducibility of the results detailed in this paper, we have provided examples of the data used in this study on the Web site.

AUTHOR INFORMATION

Corresponding Authors

Na Zhang − *Asymchem Life Science (Tianjin) Co., Ltd, Tianjin 300457, P. R. China*; Email: zhangna@asymchem.com.cn

Lu Lu − *Asymchem Life Science (Tianjin) Co., Ltd, Tianjin 300457, P. R. China*; Email: lulu@asymchem.com.cn

Authors

Likun Yang − *Asymchem Life Science (Tianjin) Co., Ltd, Tianjin 300457, P. R. China*; orcid.org/0000-0001-6741-0766

Xiaoli Liang − *Asymchem Life Science (Tianjin) Co., Ltd, Tianjin 300457, P. R. China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.3c04832

Author Contributions

L.Y. contributed to the conception and design of the study, implemented the workflow, and drafted the manuscript. X.L. participated in the Web site construction. N.Z. and L.L. provided supervision for the project. All authors have given approval to the final version of the manuscript.

Funding

This project was financially supported fully by Asymchem Life Science (Tianjin) Co. Ltd., Tianjin, P.R. China.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We thank Jing Zhang and Wei Sun for their assistance with computational resources and the website domain name.

REFERENCES

(1) Siedhoff, N. E.; Schwaneberg, U.; Davari, M. D. Machine learning-assisted enzyme engineering. *Methods Enzymol.* **2020**, *643*, 281−315.

(2) Yang, K. K.; Wu, Z.; Arnold, F. H., Machine learning in protein engineering. **2018**, arXiv preprint arXiv:1811.10775.

(3) Arnold, F. H. Directed evolution: bringing new chemistry to life. *Angew. Chem., Int. Ed.* **2018**, *57*, 4143−4148.

(4) Wu, Z.; Kan, S. B. J.; Lewis, R. D.; Wittmann, B. J.; Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, *116*, 8852−8858.

(5) Romero, P. A.; Krause, A.; Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110*, E193−E201.

(6) Greenhalgh, J. C.; Fahlberg, S. A.; Pfleger, B. F.; Romero, P. A. Machine learning-guided acyl-ACP reductase engineering for improved in vivo fatty alcohol production. *Nat. Commun.* **2021**, *12*, 5825.

(7) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **2019**, *16*, 1315−1322.

(8) Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, P.; Canny, J.; Abbeel, P.; Song, Y. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems*; Curran Associates, 2019; Vol. *32*.

(9) Nijkamp, E.; Ruffolo, J.; Weinstein, E. N.; Naik, N.; Madani, A., Progen2: exploring the boundaries of protein language models. **2022**, arXiv:2206.13517. arXiv preprint.

(10) Ferruz, N.; Schmidt, S.; Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.* **2022**, *13*, 4348.

(11) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv* **2022**, *2022*, 500902.

(12) Cheng, L.; Yang, Z.; Liao, B.; Hsieh, C.; Zhang, S., ODBO: Bayesian Optimization with Search Space Prescreening for Directed Protein Evolution. **2022**, arXiv:2205.09548. arXiv preprint.

(13) Frisby, T. S.; Langmead, C. J. Bayesian optimization with evolutionary and structure-based regularization for directed protein evolution. *Algorithms Mol. Biol.* **2021**, *16* (1), 13.

(14) Hu, R.; Fu, L.; Chen, Y.; Chen, J.; Qiao, Y.; Si, T. Protein engineering via Bayesian optimization-guided evolutionary algorithm and robotic experiments. *Briefings Bioinf.* **2023**, *24* (1), bbac570.

(15) HamediRad, M.; Chao, R.; Weisberg, S.; Lian, J.; Sinha, S.; Zhao, H. Towards a fully automated algorithm driven platform for biosystems design. *Nat. Commun.* **2019**, *10* (1), 5150.

(16) Zhang, M.; Holowko, M. B.; Hayman Zumpe, H.; Ong, C. S. Machine learning guided batched design of a bacterial Ribosome Binding Site. *ACS Synth. Biol.* **2022**, *11* (7), 2314−2326.

(17) Siedhoff, N. E.; Illig, A.-M.; Schwaneberg, U.; Davari, M. D. PyPEF—an integrated framework for data-driven protein engineering. *J. Chem. Inf. Model.* **2021**, *61* (7), 3463−3476.
(18) Wittmann, B. J.; Yue, Y.; Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **2021**, *12* (11), 1026−1045.e7.
(19) Mockus, J. Application of Bayesian approach to numerical methods of global and stochastic optimization. *J. Global Optim.* **1994**, *4* (4), 347−365.
(20) Snoek, J.; Larochelle, H.; Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*; Curran Associates, 2012, Vol. *25*.
(21) Xu, Y.; Verma, D.; Sheridan, R. P.; Liaw, A.; Ma, J.; Marshall, N. M.; McIntosh, J.; Sherer, E. C.; Svetnik, V.; Johnston, J. M. Deep Dive into Machine Learning Models for Protein Engineering. *J. Chem. Inf. Model.* **2020**, *60*, 2773−2790.
(22) Georgiev, A. G. Interpretable numerical descriptors of amino acid space. *J. Comput. Biol.* **2009**, *16* (5), 703−723.
(23) Kawashima, S.; Pokarowski, P.; Pokarowska, M.; Kolinski, A.; Katayama, T.; Kanehisa, M. AAindex: amino acid index database, progress report 2008. *Nucleic Acids Res.* **2007**, *36*, D202−D205.
(24) Meier, J.; Rao, R.; Verkuil, R.; Liu, J.; Sercu, T.; Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*; Curran Associates, 2021; Vol. *34*, pp 29287−29303.
(25) Biswas, S.; Khimulya, G.; Alley, E. C.; Esvelt, K. M.; Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nat. Methods* **2021**, *18* (4), 389−396.
(26) Graff, D. E.; Shakhnovich, E. I.; Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.* **2021**, *12*, 7866−7881.
(27) Wu, N. C.; Dai, L.; Olson, C. A.; Lloyd-Smith, J. O.; Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *Elife* **2016**, *5*, No. e16965.
(28) Podgornaia, A. I.; Laub, M. T. Pervasive degeneracy and epistasis in a protein-protein interface. *Science* **2015**, *347* (6222), 673−677.

https://doi.org/10.1021/acsomega.3c04832
ACS Omega 2023, 8, 44751−44756