Advances in Machine Learning for Directed Evolution

Bruce J. Wittmann[a], Kadina E. Johnston[a], Zachary Wu[b,c], and Frances H. Arnold[a,b]

Addresses:
[a]Division of Biology and Biological Engineering,
[b]Division of Chemistry and Chemical Engineering,
California Institute of Technology, MC 210-41
1200 E. California Boulevard, Pasadena, CA 91125 (USA)

[c]Present Address: Google DeepMind, 6 Pancras Square, Kings Cross, London, N1C 4AG (UK)

Corresponding author: Arnold, Frances H. (frances@cheme.caltech.edu)

Author contact: Bruce J. Wittmann (bwittman@caltech.edu); Kadina E. Johnston (kjohnston@caltech.edu); Zachary Wu (zacharywu@gmail.com)

Keywords: Directed evolution, machine learning, protein engineering, unsupervised learning, self-supervised learning, semi-supervised learning

Abstract

Machine learning (ML) can expedite directed evolution by allowing researchers to move expensive experimental screens *in silico*. Gathering sequence-function data for training ML models, however, can still be costly. In contrast, raw protein sequence data is widely available. Recent advances in ML approaches use protein sequences to augment limited sequence-function data for directed evolution. We highlight contributions in a growing effort to use sequences to reduce or eliminate the amount of sequence-function data needed for effective *in silico* screening. We also highlight approaches that use ML models trained on sequences to generate new functional sequence diversity, focusing on strategies that use these generative models to efficiently explore vast regions of protein space.

Navigating the Protein Fitness Landscape: Building a Map with Machine Learning

Enzymes provide solutions to life's most challenging chemical problems. The ability of enzymes to catalyze chemical reactions efficiently and selectively makes them useful not only to their host organisms, but also for myriad applications that humans have devised. As green, cheap, efficient catalysts, enzymes have been taken up by industries ranging from pharmaceuticals to consumer products, materials, food, and fuels, and their importance is expected to continue to grow [1–3].

Enzymes and many other proteins useful to humans often must function in non-native environments (non-aqueous solutions, high temperatures, in the presence of surfactants, etc.) that eliminate or reduce the activity of the natural protein. Additionally, although enzymes exhibit remarkable selectivity, they typically have a limited substrate scope, which often means that a new enzyme must be optimized for new target reactions or applications by engineering its amino acid sequence [4,5].

A protein's sequence encodes its function ("fitness"), and the relationship between them is often conceptualized as a surface in high-dimensional space called the protein fitness landscape [6,7]. New proteins are developed by searching this landscape, commonly with a process of directed evolution [7]. Directed evolution proceeds by subjecting a protein having at least a small amount of the desired function to iterative rounds of mutagenesis and screening, using the best variant in each round as the starting point for the next until the functional goal is achieved (Figure 1A). Despite its success, directed evolution relies on extensive laboratory characterization, a bottleneck for the development of many engineered proteins where screening more than a few hundred or thousand variants can be highly resource-intensive.

To reduce the experimental burden of directed evolution, protein engineers are increasingly turning to *in silico* strategies for screening, particularly machine learning (ML). When applied to directed evolution, ML has thus far largely been cast as a supervised problem; that is, given a set of protein sequences with associated labels (e.g., catalytic activity, stability, etc.), the task is to learn a function that can predict the label of previously unseen sequences (Figure 1B). Using this function, large numbers of proteins can be evaluated computationally during each cycle of evolution, enabling much greater exploration of the protein fitness landscape than could be accomplished with laboratory screening alone.

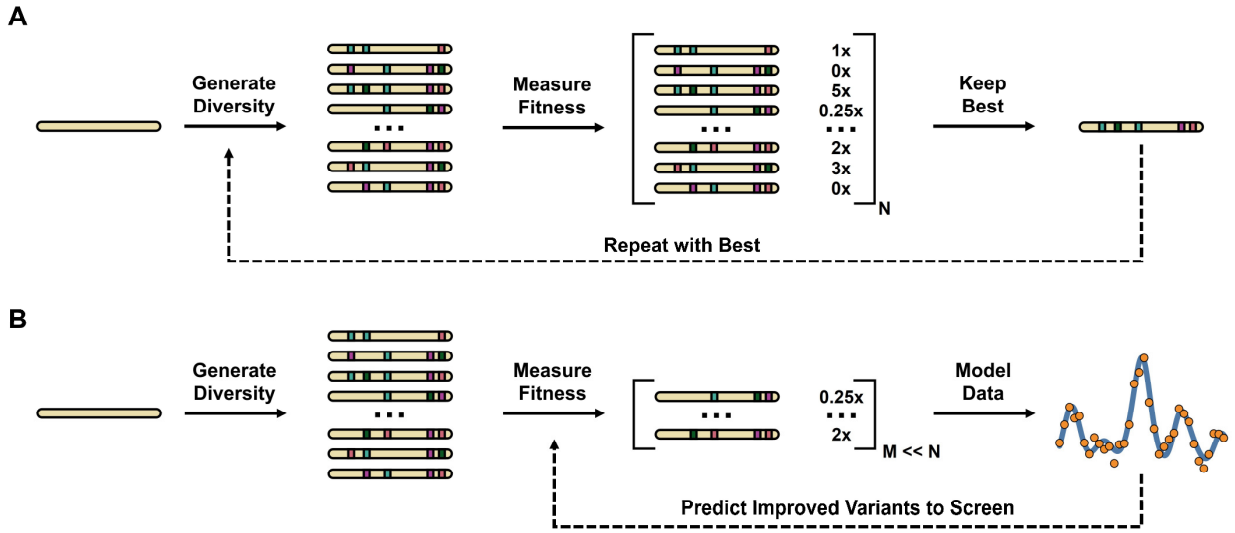

Figure 1. Example workflows of (A) traditional directed evolution and (B) supervised machine learning for directed evolution. Both workflows begin by identifying a protein with activity for a target function. Once the starting point is identified, diversity is introduced by mutagenesis and resulting variants are screened for function. (A) In traditional directed evolution, many variants are screened and the best variant is then fixed as the parent for the next round of mutagenesis/screening. (B) When applying supervised machine learning to directed evolution, fewer variants are screened. Using the resulting sequence-function data, a function is fit that relates protein sequence to protein fitness (e.g., for f(x) = y, "x" is the protein sequence and "y" is the protein fitness). This function can be used to predict the fitnesses of variants not experimentally evaluated or to propose a new set of variants to screen in the next round of evolution.

There are excellent examples of applications of supervised ML to directed evolution, and they have been covered in other reviews [8,9]. Reviews covering the applications of ML to the broader field of protein engineering [10,11] and strategies for applying ML to protein engineering have also been published [8,11,12]. The goal of this Current Opinion is not to survey all applications of ML to directed evolution nor to serve as a manual for applying ML to directed evolution, but instead to point out key recent developments and trends in ML for directed evolution. In particular, we focus on ways researchers are leveraging unsupervised learning strategies—strategies that learn from unlabeled protein sequences—to overcome the challenges associated with collecting large protein sequence-function datasets. We begin by discussing notable contributions toward using protein sequences to reduce or eliminate the amount of labeled training data needed in supervised ML. We then highlight works that demonstrate how models trained only on unlabeled data can be used to generate new sequence diversity with desired properties as well as to navigate extremely large protein fitness landscapes. We aim to make this accessible to a protein engineering audience and so avoid extensive explanation of the model architectures, algorithms, and learning strategies underpinning the examples presented.

The Cost of Labeling Proteins and How Unsupervised Pretraining Can Help

Although ML models perform best when trained with a large amount of high-quality data, gathering labeled protein data can be challenging. Indeed, except for the few protein properties either amenable to high-throughput screening or well represented in sequence-function databases, curation of such a dataset can require significant experimental resources [10,13–15]. A longstanding optimization strategy for guiding expensive data collection is active learning. In this approach, a researcher iteratively trains a model on a small amount of labeled data, then uses that model to identify new datapoints to collect which would be informative and improve model performance. Gaussian processes, which model their own uncertainty, are among the most popular models for this approach, and have been used, for instance, in the directed evolution of more thermostable cytochromes P450 and channelrhodopsin variants for optogenetics applications [16,17].

More recently, researchers have focused on augmenting small labeled datasets with information extracted from large unlabeled datasets, a strategy generally known as semi-supervised learning. When applied to protein engineering, semi-supervised learning consists of an unsupervised learning phase—often referred to as “unsupervised pretraining” or “self-supervised pretraining” due to the specific model training procedures typically employed—followed by a supervised learning phase [18]. Drastic reductions in sequencing costs have led to a deluge of unlabeled sequence data, and hundreds of millions of protein sequences are now stored in online databases [10,19–21]. Unsupervised pretraining works on the assumption that every sequenced protein follows some set of biophysical and evolutionary rules that allow that protein to be produced and carry out a biological function. By training models, which are often adapted from natural language processing (NLP) [22], on unlabeled protein sequences, the sequence constraints that result from these rules can be learned (Figure 2A) [23–29].

After training, an unsupervised model can be repurposed to generate continuous vector representations of proteins known as “embeddings”, which can be used for protein encoding (Figure 2B). A protein encoding is a vector representation of a protein sequence required for use by ML algorithms. The simplest encodings result in a sparse representation of sequence space, providing limited information about the relationships between sequences and so making learning more challenging [8,12]. Protein embeddings from unsupervised models capture information learned during pretraining and define the relationships between proteins within the context of learned sequence constraints: similar sequences will be found closer together in embedding space and so can, for instance, be inferred to have similar properties by a downstream supervised model. In this way, learned protein embeddings allow information contained in unlabeled sequences to be passed to a downstream supervised task (Figure 2C–D), in principle reducing the amount of labeled data needed compared to less informative encoding strategies [30].

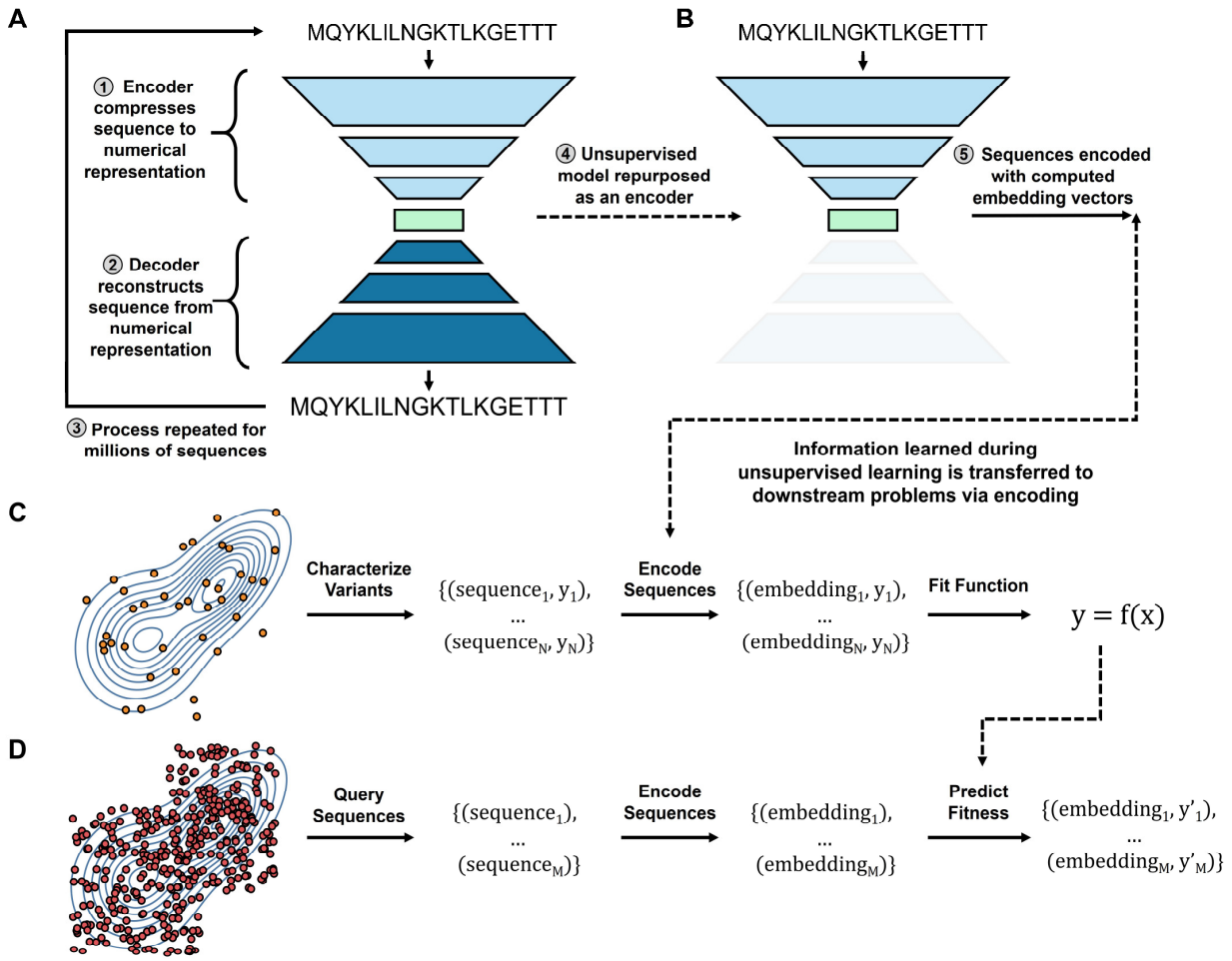

Figure 2. An example semi-supervised learning workflow illustrated using an autoencoder as the unsupervised model. (A) In this example, during the unsupervised stage, an autoencoder is trained to compress ("encode") protein sequences to a numerical representation and then use that representation to reconstruct ("decode") the sequences. The compression during encoding creates an information bottleneck (the central green layer in the figure) that forces the model to extract the most relevant features of protein sequences; the more informative the extracted features, the greater the model's ability to reconstruct sequences. (B) Once the unsupervised model is trained, the protein sequence encoder may be repurposed by removing the decoder module and taking the bottleneck ("embedding") layer as an encoding. This encoding transfers information learned during unsupervised training to a supervised process, in principle decreasing the amount of required labeled data. (C) During supervised training, an additional "top" model is trained to relate the encoded sequences to their characterized fitness values. The parameters defining the encoder can either be frozen (i.e., the encoder is not modified during supervised training) or further fine-tuned (i.e., the encoder is further trained along with the top model for the specific supervised task) during supervised training. (D) As more sequences are drawn from the fitness landscape, they are first encoded by the encoder, then passed into the learned function to predict the fitness of previously unseen protein variants.

Many models for unsupervised learning with protein sequences—sometimes complemented by other information such as phylogeny or structure—have been developed, and all have shown success when applied to downstream supervised tasks [23,25,27–29,31–36]. The application by Biswas *et al.* is particularly interesting, as it highlights how unsupervised pretraining can assist supervised learning in the extremely low-N case where models are trained on tens of variants [37]. In previous work, the authors had developed a fully unsupervised model, UniRep, by training on ~24 million sequences from the UniRef50 database [19,28]. Models like UniRep that have been trained on global databases like UniRef50 learn a general representation of protein sequence constraints across many protein families. Because the goal of

the authors was to engineer specific proteins, they further trained ("fine-tuned") UniRep on sequences homologous to an engineering target to refine the model for the desired protein family. Then, they gathered labeled data for 24 or 96 mutants for that engineering target and trained a supervised model for fitness prediction using the fine-tuned UniRep model to encode sequences. The supervised model was used in a fully *in silico* directed evolution study to identify improved variants multiple mutations away from the initial sequence. Interestingly, the authors present evidence that unsupervised pretraining served to discourage their supervised model from predicting that "unnatural" sequences (sequences significantly different from those observed during unsupervised pretraining) would be improved in fitness, suggesting that the constraints learned during pretraining are passed to downstream tasks. Assuming this case study's success generalizes, semi-supervised learning could guide researchers away from exploring sequences that are not similar to those in existing databases. Such a conservative search would likely yield fewer non-functional proteins, but it may also sacrifice the identification of mutations beneficial to target activity that are underrepresented in related proteins.

There is still much to be explored for semi-supervised learning in protein engineering. For instance, unsupervised model architectures used for pretraining have thus far been adapted primarily from NLP. While there is evidence to suggest larger NLP models trained on more diverse sequences can improve engineering outcomes [23,30], there is also evidence that much smaller models with learning objectives more tailored for proteins can achieve competitive predictive performance in downstream supervised tasks [38]. It is also not always clear when semi-supervised strategies will be superior to fully supervised ones. Shanehsazzadeh *et al.*, for example, recently showed that, when larger amounts of labeled data are available, significantly smaller models trained in a fully supervised manner can be competitive with and sometimes superior to state-of-the-art semi-supervised strategies, suggesting semi-supervised learning may be most helpful in the low-N setting [39]. Amidst the growing concern in the NLP community about the monetary and energy costs of training large language models [40], further development of smaller unsupervised models and identification of situations in which semi-supervised learning is beneficial are important areas for future research.

Finally, it is also worth noting that, given the beyond-astronomical size of possible protein space, ML for directed evolution will always be performed in a comparatively low-N setting and will never be able to fully enumerate the space of possible proteins—some degree of iteration is required. With this consideration, the question of how to combine unsupervised pretraining approaches with active learning becomes important. A strategy recently described by Hie *et al.* that combines Gaussian processes with learned protein embeddings is one possible approach, as are a number of nascent algorithms for optimization in large combinatorial spaces [41–50]. In all, distinguishing the best unsupervised model architectures and iteration strategies will require extensive benchmarking against datasets collected for different protein engineering tasks, such as those provided by Rao *et al.* [35].

Functional Classification without Labeled Data

Because mutations frequently lead to loss of function, the ability to avoid non-functional variants *a priori* would waste fewer screening resources and significantly improve the efficiency of directed evolution. Among the more interesting applications of unsupervised learning is zero-shot prediction, where fully unsupervised models are used to predict whether a protein functions without any further supervised training on labeled data [32,51,52]. Typically, this is accomplished using a generative model, which is a model trained on unlabeled protein sequence data that learns a representation of the distribution of allowed protein sequences (Figure 3A). Such models are used to query the likelihood that a new protein sequence was generated from the learned distribution of underlying sequences (Figure 3B). If this sequence is highly likely to belong to the learned distribution, then it is more likely to be a functional protein, and vice versa. In many ways, this approach is similar to the long-standing strategy of scoring protein mutants based on evolutionary conservation such as the use of BLOSUM matrices. ML models are more capable of capturing the higher order epistatic interactions believed to pervade protein evolution, however, and so are expected to be more

effective. Indeed, Riesselman *et al.* showed that nonlinear latent variable models trained on multiple sequence alignments (MSAs) were typically more effective than site-independent or pairwise evolutionary conservation methods at predicting the effects of missense mutations on 42 different proteins [51].

The quantity, quality, and distribution of sequence data used to train a sequence-based zero-shot predictor will determine how accurately that model learns to represent the true distribution of functional sequences. As a result, there is often a tradeoff between the number of training sequences and their quality or relatedness to the engineering target. For instance, while training a zero-shot predictor on MSAs allows a generative model to learn a rich representation of sequences closely related to an engineering target, if there are few sequences homologous to the target, then the learned distribution may be too narrow or sparse to be used reliably for zero-shot prediction. Indeed, DeepSequence, used by Riesselman *et al.,* struggled when applied to proteins for which few homologous sequences could be found. Because models trained on global databases can learn a more general representation of protein sequences, they may be more effective in such cases. Madani *et al.*, for example, demonstrated that a large NLP model trained on hundreds of millions of sequences from diverse families could be used as a zero-shot predictor without needing to collect protein sequences closely related to the target [32]. Of course, all of these studies assume that the target fitness of a directed evolution experiment correlates well with evolutionarily optimized fitness, but this will not always be the case (Figure 3A).

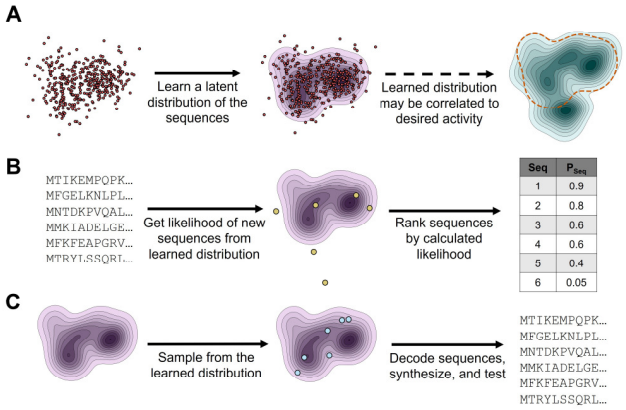

Figure 3. An illustration of the use of generative models for zero-shot prediction and sequence generation. (A) Generative models learn a representation of the distribution of allowed protein sequences from those used to train them. This distribution can correlate with the fitness landscape (green) for a desired activity, but the two distributions may not necessarily overlap. (B) When generative models are used for zero-shot prediction, it is assumed that the learned distribution correlates well with the distribution of target activity. When used in this capacity, the model is used to find the likelihood for new sequences. Sequences with high likelihood can be prioritized for screening under the assumption that a higher likelihood corresponds to higher fitness or at least higher probability of maintaining some degree of function. (C) When generative models are used for sequence generation, new sequences are drawn from the learned distribution. Sequences with higher likelihood are more likely to be drawn, and so the drawn sequences tend to be functionally similar to those used for training.

Generative Modeling for New Functional Sequence Diversity and Exploration of Vast Fitness Landscapes

Generative models can also be used to propose libraries of new, functional proteins. When used in this capacity, sequences are randomly drawn from the underlying sequence distribution learned during training to generate new candidate sequences (Figure 3C). The proteins produced will thus be representative of the learned distribution and so tend to be functionally similar to those used for model training [32,53–58]. Notably, unlike the computational cost of explicitly predicting and comparing the fitnesses of candidate proteins, the computational cost of generating candidate proteins is independent of the size of protein space considered for engineering; generative modeling thus provides an efficient strategy for identifying fit protein

variants from an extremely large pool of candidates. This concept is highlighted in the work of Repecka *et al.*, who first trained a generative model on over 16,000 malate dehydrogenase (MDH) sequences and then used it to propose new sequence diversity. From this new diversity, a functional MDH variant 106 mutations away from the closest training sequence was identified [58]. The full space of 106-mutation proteins contains ~10^{138} variants, and so could never be fully explored computationally (i.e., using a predictive model to predict the fitness of all variants) or experimentally. By drawing from a generative model instead, however, a functional variant could be identified with tractable computational cost.

From the perspective of directed evolution, an ideal use of generative models would be to identify *improved* variants among vast numbers of possibilities. Unfortunately, because the learned distribution of sequences does not explicitly model the degree to which a protein might be fit—only a sense of similarity to sequences on which the model was trained—there is no expectation that a drawn sequence will be improved in fitness. However, recently proposed strategies that couple a predictive model—which can identify fit variants, but requires potentially expensive prediction of the fitness of all candidates—with a generative model—which can propose functional variants to test from large pools of candidates—combine the strengths of both, and potentially enable optimization over vast protein fitness landscapes without extensive computational characterization [42,48–50,59]. Though the details vary, the high-level approach of such methods is to first use the generative model to propose a set of sequences for the predictive model to evaluate. Those sequences with the highest predicted fitness are then used to update the generative model (and potentially the predictive model) toward proposing higher-fitness variants. By repeating this cycle, the generative model proposes increasingly fit proteins, thus optimizing protein fitness. So far, such strategies are primarily theoretical and will need to be thoroughly validated by laboratory experimentation, though there are some examples of successful application to engineering biological systems. Linder *et al.*, for instance, developed a new approach to increase levels of both functionality and sequence diversity, demonstrating increased fitness for polyadenylation sequences and GFP variants in recent work [59].

Conclusion and Outlook

By moving expensive experimental screens *in silico*, ML greatly expands our ability to explore protein sequence space. While ML has so far been cast mainly as a supervised problem when applied to directed evolution, there has been significant expansion in unsupervised ML strategies as well. These unsupervised approaches can be used to limit or eliminate required experimental characterization of proteins, assist with navigation of combinatorial sequence space, and generate new protein sequence diversity, all of which can improve the efficiency of directed evolution campaigns. Yet, ML for directed evolution is still a relatively young field with much room for continued advancement. In particular, continued decreases in the cost and time of gene synthesis and sequencing as well as increases in computational power will make the laboratory application of ML methods more feasible and enable expansion of both sequence and sequence-function databases. As data availability grows, continued and improved collaboration between ML scientists and protein engineers will prove critical to developing experimentally tractable ML strategies that advance the field and drive more widespread adoption of the technology.

Acknowledgements: This work was supported by the Amgen Chem-Bio-Engineering Award (CBEA), the NSF Division of Chemical, Bioengineering, Environmental and Transport Systems (1937902), the Camille and Henry Dreyfus Foundation (ML-20-194), and the Caltech Carver Mead New Adventure Seed Fund.

Declaration of Interest: The authors declare no conflict of interest.

References

1. BCC Research Staff: *Global Markets for Enzymes in Industrial Applications*. BCC Research LLC; 2018. https://www.bccresearch.com/market-research/biotechnology/global-markets-for-enzymes-in-industrial-applications.html.

2. Bornscheuer UT, Huisman GW, Kazlauskas RJ, Lutz S, Moore JC, Robins K: **Engineering the third wave of biocatalysis**. *Nature* 2012, **485**:185–194.
3. Blamey JM, Fischer F, Meyer H-P, Sarmiento F, Zinn M: **Enzymatic biocatalysis in chemical transformations: A promising and emerging field in green chemistry practice**. In *Biotechnology of Microbial Enzymes*. Edited by Brahmachari G. Elsevier; 2017:347–403.
4. Rosenthal K, Lütz S: **Recent developments and challenges of biocatalytic processes in the pharmaceutical industry**. *Curr Opin Green Sustain Chem* 2018, **11**:58–64.
5. Devine PN, Howard RM, Kumar R, Thompson MP, Truppo MD, Turner NJ: **Extending the application of biocatalysis to meet the challenges of drug development**. *Nat Rev Chem* 2018, **2**:409–421.
6. Smith JM: **Natural selection and the concept of a protein space**. *Nature* 1970, **225**:563–564.
7. Romero PA, Arnold FH: **Exploring protein fitness landscapes by directed evolution**. *Nat Rev Mol Cell Biol* 2009, **10**:866–876.
8. Yang KK, Wu Z, Arnold FH: **Machine-learning-guided directed evolution for protein engineering**. *Nat Methods* 2019, **16**:687–694.
9. Li G, Dong Y, Reetz MT: **Can machine learning revolutionize directed evolution of selective enzymes?** *Adv Synth Catal* 2019, **361**:2377–2386.
10. Mazurenko S, Prokop Z, Damborsky J: **Machine learning in enzyme engineering**. *ACS Catal* 2020, **10**:1210–1223.
11. Siedhoff NE, Schwaneberg U, Davari MD: **Machine learning-assisted enzyme engineering**. In *Methods in Enzymology*. Edited by Tawfik DS. Elsevier Inc.; 2020:281–315.
12. Xu Y, Verma D, Sheridan RP, Liaw A, Ma J, Marshall NM, McIntosh J, Sherer EC, Svetnik V, Johnston JM: **A deep dive into machine learning models for protein engineering**. *J Chem Inf Model* 2020, **60**:2773–2790.
13. Jeske L, Placzek S, Schomburg I, Chang A, Schomburg D: **BRENDA in 2019: A European ELIXIR core data resource**. *Nucleic Acids Res* 2019, **47**:D542–D549.
14. Wang CY, Chang PM, Ary ML, Allen BD, Chica RA, Mayo SL, Olafson BD: **ProtaBank: A repository for protein design and engineering data**. *Protein Sci* 2018, **27**:1113–1124.
15. Fowler DM, Fields S: **Deep mutational scanning: a new style of protein science**. *Nat Methods* 2014, **11**:801–807.
16. Romero PA, Krause A, Arnold FH: **Navigating the protein fitness landscape with Gaussian processes**. *Proc Natl Acad Sci* 2013, **110**:E193–E201.
17. Bedbrook CN, Yang KK, Robinson JE, Mackey ED, Gradinaru V, Arnold FH: **Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics**. *Nat Methods* 2019, **16**:1176–1184.
18. Mao HH: **A survey on self-supervised pre-training for sequential transfer learning in neural networks**. *arXiv* 2020, arXiv:2007.00800.
19. The UniProt Consortium: **UniProt: a worldwide hub of protein knowledge**. *Nucleic Acids Res* 2019, **47**:D506–D515.
20. El-Gebali S, Mistry J, Bateman A, Eddy SR, Luciani A, Potter SC, Qureshi M, Richardson LJ, Salazar GA, Smart A, *et al.*: **The Pfam protein families database in 2019**. *Nucleic Acids Res* 2019, **47**:D427–D432.
21. Mitchell AL, Almeida A, Beracochea M, Boland M, Burgin J, Cochrane G, Crusoe MR, Kale V, Potter SC, Richardson LJ, *et al.*: **MGnify: the microbiome analysis resource in 2020**. *Nucleic Acids Res* 2020, **48**:D570–D578.
22. Young T, Hazarika D, Poria S, Cambria E: **Recent Trends in Deep Learning Based Natural Language Processing**. *IEEE Comput Intell Mag* 2018, **13**:55–75.
23. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Guo D, Ott M, Zitnick CL, Ma J, Fergus R: **Biological structure and function emerge from scaling unsupervised learning to 250 million protein**

sequences. *bioRxiv* 2020, doi:https://doi.org/10.1101/622803.

* This paper provides evidence that larger natural language processing (NLP) models trained on more diverse protein sequences can achieve improved performance when applied to downstream supervised protein engineering tasks. The authors make their developed language models available for use by others.

24. Vig J, Madani A, Varshney LR, Xiong C, Socher R, Rajani NF: **BERTology meets biology: interpreting attention in protein language models**. *arXiv* 2020, arXiv:2006.15222.
25. Melidis DP, Malone B, Nejdl W: **dom2vec: Assessable domain embeddings and their use for protein prediction tasks**. *bioRxiv* 2020, doi:https://doi.org/10.1101/2020.03.17.995498.
26. Ding X, Zou Z, Brooks III CL: **Deciphering protein evolution and fitness landscapes with latent space models**. *Nat Commun* 2019, doi:https://doi.org/10.1038/s41467-019-13633-0.
27. Elnaggar A, Heinzinger M, Dallago C, Rihawi G, Wang Y, Jones L, Gibbs T, Feher T, Angerer C, Steinegger M, *et al.*: **ProtTrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing**. *bioRxiv* 2020, doi:https://doi.org/10.1101/2020.07.12.199554.
28. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM: **Unified rational protein engineering with sequence-based deep representation learning**. *Nat Methods* 2019, **16**:1315–1322.
29. Asgari E, Mofrad MRK: **Continuous distributed representation of biological sequences for deep proteomics and genomics**. *PLoS One* 2015, doi:https://doi.org/10.1371/journal.pone.0141287.
30. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, *et al.*: **Language models are few-shot learners**. In *Advances in Neural Information Processing Systems.* Edited by Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H; 2020.
31. Kimothi D, Soni A, Biyani P, Hogan JM: **Distributed representations for biological sequence analysis**. *arXiv* 2016, arXiv:1608.05949.
32. Madani A, McCann B, Naik N, Keskar NS, Anand N, Eguchi RR, Huang PS, Socher R: **ProGen: Language modeling for protein generation**. *arXiv* 2020, arXiv:2004.03497.

* This paper demonstrates that a large NLP model can be used for zero-shot identification of functional proteins without the need to collect similar sequences to the protein of interest. This is evidence that large NLP models could possibly overcome some of the limitations of zero-shot predictors trained on alignments, which require that the protein of interest have many closely related proteins to be effective. In addition to protein sequence, the NLP model in this paper was also trained to predict protein properties such as molecular function, cellular location, etc., a strategy that can enable conditional protein generation.

33. Min S, Park S, Kim S, Choi H-S, Yoon S: **Pre-training of deep bidirectional protein sequence representations with structural information**. *arXiv* 2019, arXiv:1912.05625.
34. Yang KK, Wu Z, Bedbrook CN, Arnold FH: **Learned protein embeddings for machine learning**. *Bioinformatics* 2018, **34**:2642–2648.
35. Rao R, Bhattacharya N, Thomas N, Duan Y, Chen P, Canny J, Abbeel P, Song YS: **Evaluating protein transfer learning with TAPE**. In *Advances in Neural Information Processing Systems*. Edited by Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R Curran Associates, Inc.; 2019:9689–9701.

** The authors of this work provide a number of datasets and scripts that can be used to benchmark the applicability of unsupervised models to various downstream protein engineering tasks. Benchmarks for a number of different models are provided, and the models themselves are made available for use by others. Extensive benchmarking of new models is critical to understanding their applicability, merits, and shortcomings, and will be crucial for confident and efficient improvement of unsupervised pretraining models and strategies.

36. Bepler T, Berger B: **Learning protein sequence embeddings using information from structure**. In *International Conference on Learning Representations*. 2019.
37. Biswas S, Khimulya G, Alley EC, Esvelt KM, Church GM: **Low-N protein engineering with data-efficient deep learning**. *bioRxiv* 2020, doi:https://doi.org/10.1101/2020.01.23.917682.

** This work highlights the potential of unsupervised pretraining to reduce the amount of labeled datapoints needed in subsequent supervised tasks. By encoding proteins using embeddings derived from a purely unsupervised model, the authors were able to engineer improved variants of a green fluorescent protein and beta-lactamase using only 24 or 96 labeled datapoints. Significantly fewer improved variants were identified when proteins were encoded with alternate strategies to these embeddings.

38. Lu AX, Zhang H, Ghassemi M, Moses A: **Self-supervised contrastive learning of protein representations by mutual information maximization**. *bioRxiv* 2020, doi:https://doi.org/10.1101/2020.09.04.283929.

* This work shows that more biologically inspired, non-NLP models could be a valuable avenue of future research for unsupervised pretraining. The authors found that a model trained to distinguish between random protein fragments and subsequent protein fragments from the same protein could achieve similar performance to NLP models when applied to downstream engineering tasks, despite having 2%–10% the number of model parameters.

39. Shanehsazzadeh A, Belanger D, Dohan D: **Is transfer learning necessary for protein landscape prediction?** *arXiv* 2020, arXiv:2011.03443
40. Strubell E, Ganesh A, McCallum A: **Energy and policy considerations for deep learning in NLP**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics; 2019:3645–3650.
41. Sinai S, Wang R, Whatley A, Slocum S, Locane E, Kelsic ED: **AdaLead: A simple and robust adaptive greedy search algorithm for sequence design**. *arXiv* 2020, arXiv:2010.02141.
42. Gane A, Belanger D, Dohan D, Angermueller C, Deshpande R, Vora S, Chapelle O, Alipanahi B, Colwell L: **A comparison of generative models for sequence design**. *Machine Learning in Computational Biology Workshop* 2019.
43. Angermueller C, Dohan D, Belanger D, Deshpande R, Murphy K, Colwell L: **Model-based reinforcement learning for biological sequence design**. In *International Conference on Learning Representations*. 2020.
44. Yang KK, Chen Y, Lee A, Yue Y: **Batched stochastic bayesian optimization via combinatorial constraints design**. In *Proceedings of Machine Learning Research.* Edited by Chaudhuri K, Sugiyama M PMLR; 2019:3410–3419.
45. Kumar A, Levine S: **Model inversion networks for model-based optimization**. *arXiv* 2019, arXiv:1912.13464.
46. Myers V, Greenside P: **A hierarchical approach to scaling batch active search over structured data**. *arXiv* 2020, arXiv:2007.10263.

47. Hie B, Bryson BD, Berger B: **Leveraging uncertainty in machine learning accelerates biological discovery and design**. *Cell Syst* 2020, **11**:461-477.

* The vastness of protein space necessitates iterative exploration, even when assisted with machine learning. This paper describes a simple and elegant strategy for combining Gaussian processes—a popular model class for iterative navigation of complex spaces—with embeddings derived from unsupervised models.

48. Fannjiang C, Listgarten J: **Autofocused oracles for model-based design**. *arXiv* 2020, arXiv:2006.08052.
49. Brookes DH, Listgarten J: **Design by adaptive sampling**. *arXiv* 2018, arXiv:1810.03714.
50. Brookes DH, Park H, Listgarten J: **Conditioning by adaptive sampling for robust design**. In *Proceedings of the 36th International Conference on Machine Learning*. Edited by Chaudhuri K, Salakhutdinov R PMLR; 2019:773–782.
51. Riesselman AJ, Ingraham JB, Marks DS: **Deep generative models of genetic variation capture the effects of mutations**. *Nat Methods* 2018, **15**:816–822.

* This work demonstrates how information learned from multiple sequence alignments (MSAs) alone can be used to predict protein fitness. The ability to filter out non-functional variants *a priori* can significantly improve the efficiency of protein engineering by avoiding wasting valuable screening resources. The authors provide code that, given an MSA for a protein of interest, can train a model to predict the fitness of new variants.

52. Riesselman A, Shin J-E, Kollasch A, McMahon C, Simon E, Sander C, Manglik A, Kruse A, Marks D: **Accelerating protein design using autoregressive generative models**. *bioRxiv* 2019, doi:https://doi.org/10.1101/757252.
53. Wu Z, Yang KK, Liszka M, Lee A, Batzilla A, Wernick D, Weiner DP, Arnold FH: **Signal peptides generated by attention-based neural networks**. *ACS Synth Biol* 2020, **9**:2154–2161.
54. Greener JG, Moffat L, Jones DT: **Design of metalloproteins and novel protein folds using variational autoencoders**. *Sci Rep* 2018, doi:https://doi.org/10.1038/s41598-018-34533-1.
55. Hawkins-Hooker A, Depardieu F, Baur S, Couairon G, Chen A, Bikard D: **Generating functional protein variants with variational autoencoders**. *bioRxiv* 2020, doi:https://doi.org/10.1101/2020.04.07.029264.
56. Das P, Wadhawan K, Chang O, Sercu T, Dos Santos C, Riemer M, Chenthamarakshan V, Padhi I, Mojsilovic A: **PepCVAE: Semi-supervised targeted design of antimicrobial peptide sequences**. *arXiv* 2018, arXiv:1810.07743.
57. Amimeur T, Shaver JM, Ketchem RR, Taylor JA, Clark RH, Smith J, Van Citters D, Siska CC, Smidt P, Sprague M, *et al.*: **Designing feature-controlled humanoid antibody discovery libraries using generative adversarial networks**. *bioRxiv* 2020, doi:https://doi.org/10.1101/2020.04.12.024844.
58. Repecka D, Jauniskis V, Karpus L, Rembeza E, Zrimec J, Poviloniene S, Rokaitis I, Laurynenas A, Abuajwa W, Savolainen O, *et al.*: **Expanding functional protein sequence space using generative adversarial networks**. *bioRxiv* 2019, doi:https://doi.org/10.1101/789719.

* This is one of the few early sequence-generation studies in which predicted sequences were synthesized and tested for a desired activity. The authors trained a generative model on malate dehydrogenase sequences and then used it to generate new, soluble and functional ones. Some of the generated sequences even showed wild-type-level activity, although they did not outperform wild type.

59. Linder J, Bogard N, Rosenberg AB, Seelig G: **A generative neural network for maximizing fitness and diversity of synthetic DNA and protein sequences**. *Cell Syst* 2020, **11**:49-62.

* The authors develop a new generative model approach designed to optimize for both sequence diversity (through a contrastive training procedure) and fitness (through a regressive fitness prediction component), while further guiding generation by staying within the model's confident regions. Additionally, the authors perform experimental validation with a variety of biological applications, providing a strong case for applying generative models to optimize biomolecules.