Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Structural Biology

Adaptive machine learning for protein engineering

Brian L. Hie[1,2] and Kevin K. Yang[3]

Abstract

Machine-learning models that learn from data to predict how protein sequence encodes function are emerging as a useful protein engineering tool. However, when using these models to suggest new protein designs, one must deal with the vast combinatorial complexity of protein sequences. Here, we review how to use a sequence-to-function machine-learning surrogate model to select sequences for experimental measurement. First, we discuss how to select sequences through a single round of machine-learning optimization. Then, we discuss sequential optimization, where the goal is to discover optimized sequences and improve the model across multiple rounds of training, optimization, and experimental measurement.

Addresses

[1] Department of Biochemistry, Stanford University School of Medicine, Stanford, CA, 94305, USA

[2] Stanford ChEM-H, Stanford University, Stanford, CA, 94305, USA

[3] Microsoft Research New England, Cambridge, MA, 02142, USA

Corresponding author: Yang, Kevin K. (yang.kevin@microsoft.com)

Current Opinion in Structural Biology 2022, **72**:145–152

This review comes from a themed issue on **Artificial Intelligence (AI) Methodologies in Structural Biology**

Edited by **Feixiong Cheng** and **Nurcan Tuncbag**

For complete overview of the section, please refer the article collection - Artificial Intelligence (AI) Methodologies in Structural Biology

Available online 9 December 2021

https://doi.org/10.1016/j.sbi.2021.11.002

Keywords

Machine learning, Protein engineering, Model-based optimization, Adaptive sampling, Bayesian optimization, Gaussian process.

Introduction

Protein engineering seeks to design or discover novel proteins with useful properties [1], but doing so is challenging because (i) evaluating a given design is costly (requiring a laboratory experiment to express and characterize each design) and (ii) the search space of all possible protein sequences is very large (with twenty natural amino acids, there are 20^{100} possible sequences of length 100, a search space larger than the number of atoms in the universe) [2].

To reduce the number of evaluations required, researchers can leverage machine learning to train a surrogate model that predicts the property of interest from the protein sequence. Because making predictions from a machine-learning model is less expensive and faster than conducting a wet-lab experiment, the surrogate model can reduce the overall experimental burden [2,3]. However, once a sequence-to-function model is available, a second consideration remains: how does one select new designs from the combinatorially large protein search space?

In this review, we address this question through what we call 'adaptive machine learning,' which covers two problem settings. First, we consider the problem of using a trained machine-learning surrogate model to select one or more optimized sequences for laboratory measurement. Second, we consider the problem of finding optimized sequences over sequential rounds of experimental measurement, model training, and property prediction. We describe key principles, highlight useful examples from the literature, and discuss areas that would benefit from future methodological improvements.

Overview and key principles

There are four important components to consider when doing adaptive learning for protein optimization: the property to optimize, the sequence-to-function predictor, the acquisition function that prioritizes designs, and the generative model that proposes the designs (Figure 1a–d).

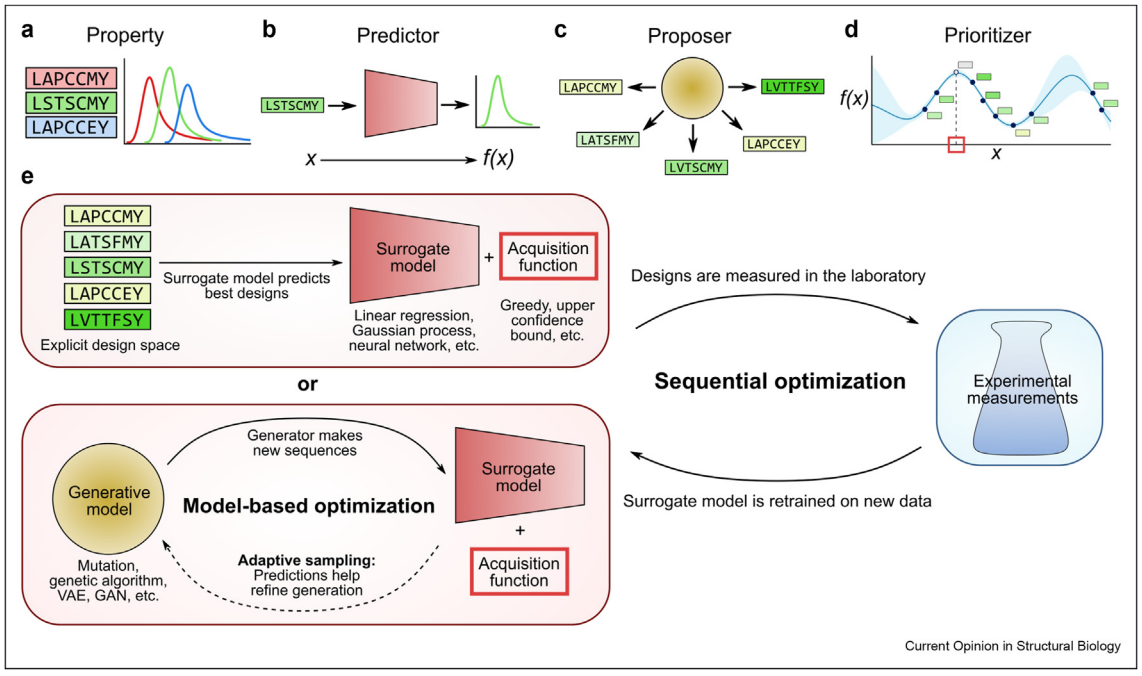
The property

The optimization property is the phenotype-of-interest (Figure 1a). For example, one could maximize the fluorescence intensity at a given excitation wavelength. Multiple properties may be considered, such as maximizing the enzymatic activity while minimizing the immunogenicity of a protein drug.

The predictor

The surrogate model takes in protein sequence information and predicts the value of the optimization property (Figure 1b). The surrogate model requires a number of design considerations, including how to represent the protein sequence and what kind of machine-learning algorithm to use. Typical sequence representations range from a simple binary encoding of

Figure 1



Overview of adaptive machine learning for protein engineering. Four key components are (**a**) the optimization property (such as enzymatic activity or protein fluorescence), (**b**) the surrogate model that predicts the property given a sequence (such as a linear regression model), (**c**) a generative model that proposes sequences (such as a variational autoencoders (VAE) neural network), and (**d**) an acquisition function that prioritizes sets of sequences to measure based on sequence information and predictions from the surrogate model (for example, greedily acquiring the top prediction according to the surrogate model, see Figure 2). (**e**) Sequences can be acquired from an explicit design space or sequences proposed by a generative model. A surrogate model is then evaluated on these sequences, and an acquisition function uses the output of the surrogate model to select a set of sequences for experimental measurement in the laboratory. Optionally, the surrogate model's predictions can guide the generative model toward better designs through adaptive sampling. The data from measuring newly acquired sequences can help guide future rounds of optimization, for example, by retraining the surrogate model or the generative model.

the raw sequence to more complex continuous neural encodings [4,5]. Model architectures also range in complexity from linear regression to deep neural networks and have been reviewed in-depth previously [2,5]. Often, it is also desirable for the surrogate model to quantify the uncertainty in its predictions to make reliable decisions. Popular models that provide a notion of uncertainty include Gaussian processes and model ensembles, which we review in Section Sequential optimization.

The prioritizer

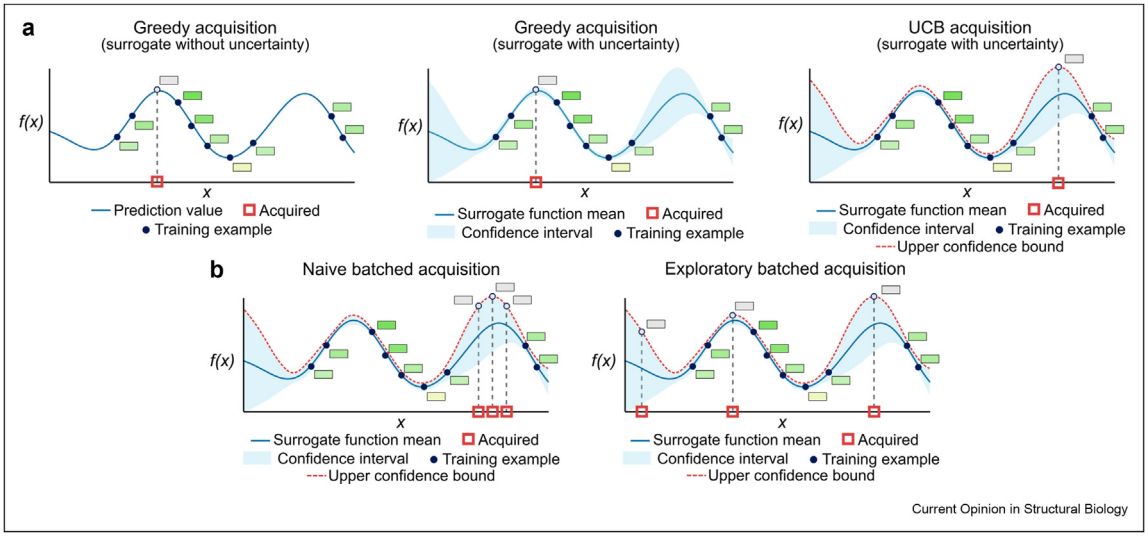
An acquisition function uses sequence information and the output of the surrogate model to prioritize designs for experimental measurements (Figure 1d). The simplest acquisition function greedily selects the top prediction (or the top few predictions) according to the surrogate model. Greedy acquisition is common in practice and can work well [3,6–10] but can also limit the search to narrower regions of the protein landscape (Figure 2a), which we discuss further in Section Sequential optimization.

Many standard acquisition functions are designed to select only a single example on each round. Often, however, obtaining many experimental measurements in parallel is desirable. While simply acquiring several of the top-ranked sequences is possible, this approach may be prone to acquiring many similar sequences (Figure 2b). Special methods for batched acquisition are therefore designed to encourage acquiring more diverse sequences [11–16], but this remains an open area of methodological development.

The proposer

Although evaluating the surrogate model is faster and cheaper than actually obtaining laboratory measurements, it is nevertheless impossible to evaluate the surrogate on all possible protein sequences. One approach is to explicitly define a design space of proteins. Example of explicit design spaces include traditional library designs for protein engineering, such as all single- or double-mutants, all the possible mutations at a small number of sites [6,7,17], or a recombination library defined by mixing and matching parts of homologous parent sequences [12,18–20].

Figure 2



Overview of acquisition functions. (**a**) The greedy acquisition function can work well but may limit the search to suboptimal regions of the sequence landscape, whereas incorporating uncertainty helps an acquisition function explore the landscape. Left: a greedy acquisition function acquires the sequence with the highest prediction value according to the surrogate model (blue line). Center: when the surrogate model also estimates the uncertainty of its predictions (shaded blue region), a greedy acquisition function still acquires the sequence with the highest prediction value without considering uncertainty. Right: in contrast to the greedy acquisition, upper confidence bound (UCB) acquisition acquires according to the surrogate model's prediction value and the estimated uncertainty, allowing the model to balance exploration of uncertain regimes and exploitation of high-confidence regimes. (**b**) Acquisition of multiple sequences within the same optimization round is called 'batched acquisition.' Left: in the batched setting, naively acquiring multiple sequences according to the prediction value or the UCB can lead to under-exploration by focusing on the local maxima of the acquisition function. Right: special batched acquisition functions can be designed to acquire a diverse set of examples. For example, UCB-based batched acquisition can reduce the uncertainty around an example once it has been selected [13].

Another approach is to use a generative model to implicitly define the design space by learning a probability distribution over sequences (Figure 1c). A generative model performs one or both of two fundamental tasks: (i) assigning every possible sequence a likelihood of being in the proposal distribution and (ii) generating examples of sequences from a proposal distribution. The simplest models assume this distribution can be modeled by considering sites independently or by relationships between pairs of sites [21,22]. More recently, researchers have used deep neural network generative models to learn more complex sequence distributions and propose sequences for evaluation [23]; relevant neural architectures include variational autoencoders (VAEs) [24–26], generative adversarial networks (GANs) [27,28], and autoregressive language models [29–32].

Model-based protein optimization

After obtaining a surrogate model, model-based optimization can prioritize sequences of interest by finding the sequence or sequences in the design space that optimize the acquisition function (Figure 1e). Notably, in this setting, we do not provide the surrogate model with new training data. The simplest approach is to first obtain a sequence design space (explicitly defined or from a generative model) and then to select sequences from this design space based on a surrogate model and a corresponding acquisition function (for example, picking the best predicted sequences in the design space) [6,12,40,10].

Adaptive sampling from a generative model

Rather than separate generation and evaluation steps, the surrogate model can also help shift the generative model so that it proposes more optimal sequences. For example, in adaptive sampling, protein sequences are sampled from a generative model, the outputs of a surrogate model are used to re-estimate the parameters of the generative model, and the process iterates until convergence [26]. For example, Brookes and Listgarten [26] use adaptive sampling based on a VAE generative model and a neural network surrogate model to optimize DNA sequences for protein expression abundance. Gupta and Zou [28] use adaptive sampling based on a GAN generative model and a recurrent neural network surrogate model to design antimicrobial peptides. To perform *de novo* protein design [41], Anishchenko et al. [37] use adaptive sampling based on a mutation-based generative model and a neural network surrogate model meant to identify sequences with valid folds (Table 1).

Table 1

Examples of adaptive learning for protein engineering. Acq. func.: Acquisition function. Seq. opt.: Sequential optimization, indicates studies that performed multiple rounds of variant selection and surrogate model training. In-vitro: Indicates studies that obtained in-vitro measurements of new protein sequences. UCB: upper confidence bound; VAE: variational autoencoder; GAN: generative adversarial network; RNN: recurrent neural network [29]; CNN: convolutional neural network [39].

Reference and date	Optimization property	Surrogate model	Acq. func.	Generative model/design space	Seq. opt.?	In-vitro?
Fox et al. (2007) [3]	Enzyme catalytic activity	Linear regression	Greedy	Sequence recombination	Yes	Yes
Romero et al. (2013) [12]	Protein thermostability	Gaussian process	UCB	Sequence recombination	Yes	Yes
Bedbrook et al. (2017) [33]	Protein localization	Gaussian process	UCB	Sequence recombination	Yes	Yes
Wu et al. (2019) [6]	Enzyme catalytic activity	Regressor ensemble	Greedy	Explicitly-defined design space	Yes	Yes
Brookes et al. (2019) [34]	Protein fluorescence	Neural network ensemble	Greedy	Neural network (VAE)	No	No
Kumar and Levine (2019) [35]	Protein fluorescence	Neural network ensemble	Greedy	Neural network (GAN)	No	No
Gupta and Zou (2020) [28]	Antimicrobial activity	Neural network (RNN)	Greedy	Neural network (GAN)	No	No
Liu et al. (2020) [36]	Antibody affinity	Neural network ensemble	Greedy	Activation maximization	No	Yes
Wittmann et al. (2020) [7]	Protein expression and binding	Regressor ensemble	Greedy	Explicitly-defined design space	Yes	No
Anishchenko et al. (2021) [37]	Valid folding	Neural network (CNN)	Greedy	Sequence mutation	No	Yes
Biswas et al. (2021) [8]	Protein fitness, fluorescence	Linear regression	Greedy	Sequence mutation	No	Yes
Bryant et al. (2021) [9]	Protein viability	Classifier ensemble	Greedy	Sequence mutation	No	Yes
Greenhalgh et al. (2021) [38]	Enzyme catalytic activity	Gaussian process	UCB	Sequence recombination	Yes	Yes

By default, adaptive sampling assumes a trustworthy surrogate model; however, in practice, these models are imperfect and can be prone to poor predictions in many regions of the protein space. To address this problem, adaptive sampling can avoid degeneracies in the surrogate model by constraining sampling to be close to the training distribution for the surrogate model [34,37]. In each iteration, it is also possible to retrain the surrogate model to avoid pathologies [42]. Sequence generation can also be improved by sampling from an ensemble of generative models [43].

A closely-related approach to model-based optimization uses genetic algorithms to heuristically balance both mutation and recombination to produce new sequences [44] by adaptively querying a surrogate model to preserve sequence designs [16]. Another approach to model-based optimization inverts the surrogate model by finding the elements from the generative model's distribution that are most likely to have a desirable value according to the surrogate model. An inverse of the surrogate model can be trained via an iterative procedure similar to adaptive sampling [35], or the inverse of a differentiable surrogate model can be computed using gradient-based methods [36,45,46].

Sequential optimization

In sequential optimization, the surrogate model has access to multiple rounds of experimental measurement, which provide new data that can be reincorporated into subsequent rounds of model training [47] (Figure 1e). To train the initial surrogate model, the first batch can consist of random samples from the design space [7] or sequences with known measurements (for example, from publicly available data) [8]. Sequential optimization is guided by an objective function that specifies the overall goal of the optimization procedure. The objective most relevant to protein engineering is to find the protein sequence that maximizes the optimization property (for example, to find the most fluorescent sequence).

Greedy acquisition across experimental rounds is the simplest implementation of sequential optimization and is used widely in practice [7] (Figure 2a). Going beyond greedy acquisition means tolerating more risk for potentially higher reward, which is often described as a tradeoff between 'exploitation' (equivalent to greedy acquisition) and 'exploration' [48–51], in which an algorithm acquires the proteins with high surrogate-model uncertainty to improve future predictions. Here, we focus predominately on Bayesian optimization as the main alternative to greedy acquisition. Different objectives, such as training the best overall model through active learning [47] or reformulating the objective as a reinforcement learning problem [52], are also possible, although less common in protein engineering applications.

Bayesian optimization

Bayesian optimization searches for a protein that maximizes the optimization property [50] by leveraging Bayesian uncertainty in the surrogate function to guide the exploration–exploitation tradeoff. Bayesian optimization relies on the acquisition function to systematically weigh the surrogate model's prediction with its associated uncertainty. The UCB acquisition function

adds the prediction value with a weighted uncertainty term that lets the user control the influence of uncertainty on the prediction, with a larger weight encouraging more exploration [49,53] (Figure 2a). UCB has good theoretical properties [50] and is used widely in practice [12,33,38,54] (Table 1). Other notable acquisition functions select an example that is predicted to, in expectation, have the largest improvement over the best example in the training set or a randomly drawn example from the training set [50,55,56]. Uncertainty can also help improve exploration in batched acquisition [11–15] (Figure 2b).

Gaussian process surrogate models

Gaussian process models are popular surrogate models for Bayesian optimization [57]. Gaussian process regression assumes that the optimization-property values of any set of sequences are distributed according to a multivariate Gaussian. Often, the mean of the distribution is used as the prediction value and the marginal variance can be used to estimate uncertainty. Probabilistic classification with Gaussian processes is also possible but requires approximation [58,57], and multitask Gaussian processes can be used to predict multiple optimization properties [59]. Because of their theoretical elegance, flexibility, and good performance in practice, Gaussian processes have seen wide use in protein engineering applications [12,33,38,54,60,61].

Gaussian processes are defined by a mean function and a kernel function that together specify how training examples influence predictions at unobserved sequences. The most widely used kernel functions [62] are defined on continuous inputs and can accommodate neural embeddings or binary sequence embeddings [4,33,54]. Kernels can also be defined on discrete inputs, including sequences [63–65].

The cost of exact inference with Gaussian processes grows cubically with the size of the training data, which may be prohibitive on extremely large protein sequence datasets. There is a wide literature on scaling Gaussian processes, including methods for exploiting sparsity in the structure of the training data or performing approximate inference, which applies to both continuous and discrete inputs [66,67].

Beyond Gaussian process regression

Bayesian optimization can also leverage more bespoke Bayesian models and algorithms for exact or approximate inference [68]. The wide interest in neural network models over the last decade has also led to increased interest in uncertainty prediction through Bayesian neural networks, in which the parameters of the network are themselves random variables with associated prior distributions, although efficient and accurate inference in these models can be challenging [69].

Probabilistic surrogate models can also be implemented by model ensembles, which train multiple sequence-to-function models on the same data and rely on variance in model predictions due to different model architectures or randomness in the training procedure to estimate uncertainty [36,70,71]. Ensembles are not Bayesian by default, so incorporating prior information into these models can be challenging [72,73].

Discussion

Protein engineering is a challenging task given its proven computational hardness [74] and general biological complexity. Here, we have reviewed how machine learning can help protein engineers deal with the immense complexity of the protein sequence landscape by proposing sequence designs and guiding a researcher across multiple rounds of laboratory experimentation.

While the techniques we review are a good representation of the current state-of-the-art, there is much room for methodological development. When performing Bayesian optimization, obtaining well-calibrated uncertainty estimates can be more difficult when the inputs are discrete or high-dimensional (or both).

Protein engineering also typically involves much fewer rounds (because experimental measurements are often resource-intensive) and much larger batches (for example, using multiplexed experimental designs) than is typically assumed in the theoretical literature for Bayesian optimization. Another consideration, particularly in scientific applications, is to design proteins without using a natural starting point or for properties that do not exist in natural proteins. Doing so with data-driven approaches may require additional modeling considerations beyond sequence data alone, such as biophysics, biochemistry, and protein structure.

Conflict of interest statement

Nothing declared.

Acknowledgements

The authors thank Nicholas Bhattacharya and Sam Sinai for helpful comments and discussion. B.L.H. acknowledges the support of the Stanford Science Fellows program.

References

Papers of particular interest, published within the period of review, have been highlighted as:

* of special interest
** of outstanding interest

1. Arnold FH: **Directed evolution: bringing new chemistry to life**. *Angew Chem Int Ed* 2018, **57**, https://doi.org/10.1002/anie.201708408.

2. Yang KK, Wu Z, Arnold FH: **Machine-learning-guided directed evolution for protein engineering**. *Nat Methods* 2019, **16**: 687–694, https://doi.org/10.1038/s41592-019-0496-6. arXiv: 1811.10775.

3. Fox RJ, Davis SC, Mundorff EC, Newman LM, Gavrilovic V, Ma SK, Chung LM, Ching C, Tam S, Muley S, Grate J, Gruber J, Whitman JC, Sheldon RA, Huisman GW: **Improving catalytic function by ProSAR-driven enzyme evolution**. *Nat Biotechnol* 2007, **25**:338–344, https://doi.org/10.1038/nbt1286.

4. Wittmann BJ, Johnston KE, Wu Z, Arnold FH: **Advances in machine learning for directed evolution**. *Curr Opin Struct Biol* 2021, **69**:11–18.

5. Frappier V, Keating AE: **Data-driven computational protein design**. *Curr Opin Struct Biol* 2021, **69**:63–69, https://doi.org/10.1016/j.sbi.2021.03.009.

6. Wu Z, Jennifer Kan SB, Lewis RD, Wittmann BJ, Arnold FH: **Machine learning-assisted directed protein evolution with combinatorial libraries**. *Proc Natl Acad Sci USA* 2019, **116**:8852–8858, https://doi.org/10.1073/pnas.1901979116.

7. * Wittmann BJ, Yue Y, Arnold FH: *Machine learning-assisted directed evolution navigates a combinatorial epistatic fitness landscape with minimal screening burden*. 2020, https://doi.org/10.1101/2020.12.04.408955.

This paper explores how different choices of sequence representations, surrogate models, and greedy (batched) acquisition functions affect the ability of machine-learning-guided directed evolution to acquire the optimal sequence over rounds of sequential optimization.

8. ** Biswas S, Khimulya G, Alley EC, Esvelt KM, Church GM: **Low-N protein engineering with data-efficient deep learning**. *Nat Methods* 2021, **18**, https://doi.org/10.1038/s41592-021-01100-y.

This paper demonstrates how neural sequence embeddings combined with linear-regression surrogate models can be used to greedily acquire sequences with desirable properties using a small amount of training data.

9. * Bryant DH, Bashir A, Sinai S, Jain NK, Ogden PJ, Riley PF, Church GM, Colwell LJ, Kelsic ED: **Deep diversification of an AAV capsid protein by machine learning**. *Nat Biotechnol* 2021, https://doi.org/10.1038/s41587-020-00793-4.

This paper evaluates the ability of a logistic regression model, a recurrent neural network, and a convolutional neural network to propose diverse, viable capsid proteins.

10. Singer JM, Novotney S, Strickland D, Haddox HK, Leiby N, Rocklin GJ, Chow CM, Roy A, Bera AK, Motta FC, Cao L, Strauch E-M, Chidyausiku TM, Ford A, Ho E, Mackenzie CO, Eramian H, DiMaio F, Grigoryan G, Vaughn M, Stewart LJ, Baker D, Klavins E: **Large-scale design and refinement of stable proteins using sequence-only models**. *bioRxiv* 2021, https://doi.org/10.1101/2021.03.12.435185.

11. Azimi J, Fern A, Fern XZ: **Batch Bayesian optimization via simulation matching**. *Adv Neural Inf Process Syst* 2010, **23**:109–117.

12. Romero PA, Krause A, Arnold FH: **Navigating the protein fitness landscape with Gaussian processes**. *Proc Natl Acad Sci USA* 2013, **110**, https://doi.org/10.1073/pnas.1215251110.

13. Desautels T, Krause A, Burdick JW: **Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization**. *J Mach Learn Res* 2014, **15**:4053–4103. URL, http://jmlr.org/papers/v15/desautels14a.html.

14. González J, Dai Z, Hennig P, Lawrence N: **Batch bayesian optimization via local penalization**. In *Proceedings of the 19th international conference on artificial intelligence and statistics, AISTATS*; 2016:2016.

15. Yang KK, Chen Y, Lee A, Yue Y: **Batched stochastic Bayesian optimization via combinatorial constraints design**. *Int Conf Artif Intell Stat* 2020, **22**:3410–3419.

16. * Sinai S, Slocum S, Wang R, Locane E, Whatley A, Kelsic ED: *AdaLead: A simple and robust adaptive greedy search algorithm for sequence design*. 2020. arXiv cs.LG (2010.02141).

This paper demonstrates how a simple genetic algorithm and greedy acquisition is competitive with other strategies for model-based optimization.

17. Wu NC, Dai L, Olson CA, Lloyd-Smith JO, Sun R: **Adaptation in protein fitness landscapes is facilitated by indirect paths**. *eLife* 2016, **5**, e16965, https://doi.org/10.7554/eLife.16965.

18. Voigt CA, Martinez C, Wang ZG, Mayo SL, Arnold FH: **Protein building blocks preserved by recombination**. *Nat Struct Biol* 2002, **9**, https://doi.org/10.1038/nsb805.

19. Otey CR, Landwehr M, Endelman JB, Hiraga K, Bloom JD, Arnold FH: **Structure-guided recombination creates an artificial family of cytochromes P450**. *PLoS Biol* 2006, **4**, https://doi.org/10.1371/journal.pbio.0040112.

20. Smith MA, Romero PA, Wu T, Brustad EM, Arnold FH: **Chimeragenesis of distantly-related proteins by noncontiguous recombination**. *Protein Sci* 2013, **22**:231–238, https://doi.org/10.1002/pro.2202.

21. Hopf TA, Ingraham JB, Poelwijk FJ, Schärfe CP, Springer M, Sander C, Marks DS: **Mutation effects predicted from sequence co-variation**. *Nat Biotechnol* 2017, **35**:128–135, https://doi.org/10.1038/nbt.3769.

22. Russ WP, Figliuzzi M, Stocker C, Barrat-Charlaix P, Socolich M, Kast P, Hilvert D, Monasson R, Cocco S, Weigt M, Ranganathan R: **An evolution-based model for designing chorismate mutase enzymes**. *Science* 2020, **369**:440–445, https://doi.org/10.1126/science.aba3304.

23. Wu Z, Johnston KE, Arnold FH, Yang KK: *Protein sequence design with deep generative models*. 2021. arXiv q-bio.QM (2104.04457).

24. Kingma DP, Welling M: *Auto-encoding variational bayes, 2nd international conference on learning representations*. 2014. arXiv: 1312.6114arXiv:arXiv:1312.6114vol. 10.

25. Riesselman AJ, Ingraham JB, Marks DS: **Deep generative models of genetic variation capture the effects of mutations**. *Nat Methods* 2018, **15**:816–822, https://doi.org/10.1038/s41592-018-0138-4.

26. Brookes DH, Listgarten J: *Design by adaptive sampling*. 2018. arXiv cs.LG (1810.03714).

27. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y: **Generative adversarial networks**. *Commun ACM* 2020, **63**:139–144, https://doi.org/10.1145/3422622.

28. * Gupta A, Zou J: **Feedback GAN for DNA optimizes protein functions**. *Nat Machine Intell* 2019, **1**:105–111, https://doi.org/10.1038/s42256-019-0017-4.

This paper uses adaptive sampling with a GAN generative model and a neural network surrogate model to design antimicrobial peptides.

29. Hochreiter S, Schmidhuber J: **Long short-term memory**. *Neural Comput* 1997, **9**:1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

30. Bepler T, Berger B: **Learning protein sequence embeddings using information from structure**. In *7th international conference on learning representations*. arXiv; 2019:1902. 08661.

31. * Shin J-E, Riesselman AJ, Kollasch AW, McMahon C, Simon E, Sander C, Manglik A, Kruse AC, Marks DS: **Protein design and variant prediction using autoregressive generative models**. *Nat Commun* 2021, **12**. Article number: 2403.

This paper uses an autoregressive language model as a sequence generator to proposes diverse camelid nanobody designs.

32. Madani A, McCann B, Naik N, Keskar NS, Anand N, Eguchi RR, Huang P-S, Socher R: **ProGen: language modeling for protein generation**. *bioRxiv* 2021, https://doi.org/10.1101/2020.03.07.982272.

33. Bedbrook CN, Yang KK, Rice AJ, Gradinaru V, Arnold FH: **Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization**. *PLoS Comput Biol* 2017, **13**, e1005786, https://doi.org/10.1371/journal.pcbi.1005786.

34. * Brookes DH, Park H, Listgarten J: **Conditioning by adaptive sampling for robust design**. *Int Conf Machine Learn* 2019, **36**:773–782.

This paper extends the design-by-adaptive-sampling framework to avoid pathologies in the surrogate model by constraining designs to be close to a prior distribution, which is applied to designing fluorescent proteins.

35. Kumar A, Levine S: **Model inversion networks for model-based optimization**. *Adv Neural Inf Process Syst* 2020, **33**.

36. * Liu G, Zeng H, Mueller J, Carter B, Wang Z, Schilz J, Horny G, Birnbaum ME, Ewert S, Gifford DK: **Antibody complementarity determining region design using high-capacity machine learning**. *Bioinformatics* 2020, **36**:2126–2133, https://doi.org/10.1093/bioinformatics/btz895.
This paper uses activation maximization, a gradient-based model inversion method, to design high-affinity antibodies.

37. * Anishchenko I, Pellock SJ, Chidyausiku TM, *et al.*: **De novo protein design by deep network hallucination**. *Nature* 2021, https://doi.org/10.1038/s41586-021-04184-w.
This paper uses an adaptive sampling approach to perform de-novo protein design based on a surrogate model that predicts the three-dimensional fold of a given sequence.

38. ** Greenhalgh JC, Fahlberg SA, Pfleger BF, *et al.*: **Machine learning-guided acyl-ACP reductase engineering for improved in vivo fatty alcohol production**. *Nat Commun* 2021, **12**:5825, https://doi.org/10.1038/s41467-021-25831-w.
This paper uses a Gaussian process regressor and UCB acquisition over multiple rounds of sequential optimization to engineer improved fatty acyl reductases.

39. Fukushima K: **Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**. *Biol Cybern* 1980, **36**:267–285, https://doi.org/10.1007/BF00344251.

40. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, Aguilera-Iparraguirre J, Hirzel TD, Adams RP, Aspuru-Guzik A: **Automatic chemical design using a data-driven continuous representation of molecules**. *ACS Cent Sci* 2018, **4**:268–276, https://doi.org/10.1021/acscentsci.7b00572.

41. Huang PS, Boyken SE, Baker D: *The coming of age of de novo protein design*. 2016, https://doi.org/10.1038/nature19946.

42. * Fannjiang C, Listgarten J: **Autofocused oracles for model-based design**. *Adv Neural Inf Process Syst* 2020, **33**.
This paper demonstrates that retraining a surrogate model during adaptive sampling can improve model-based optimization.

43. * Angermueller C, Belanger D, Gane A, Mariet Z, Dohan D, Murphy K, Colwell L, Sculley D: **Population-based black-box optimization for biological sequence design**. *Int Conf Machine Learn* 2020, **37**:324–334.
This paper improves the robustness and diversity of sequence designs by using an ensemble of generative models that can be re-weighted based on the output of the surrogate model over multiple rounds of adaptive sampling.

44. Hansen N: **The CMA evolution strategy: a comparing review**. In *Towards a new evolutionary computation*; 2006:75–102, https://doi.org/10.1007/11007937_4.

45. Linder J, Seelig G: *Fast differentiable DNA and protein sequence optimization for molecular design*, vol. 2005. arXiv cs.LG; 2020: 11275.

46. 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.e16, https://doi.org/10.1016/j.cels.2020.05.007.

47. Eisenstein M: **Active machine learning helps drug hunters tackle biology**. *Nat Biotechnol* 2020, **38**:512, https://doi.org/10.1038/s41587-020-0521-4.

48. Robbins H: **Some aspects of the sequential design of experiments**. *Bull Am Math Soc* 1952, **58**:527–535.

49. Auer P: **Using confidence bounds for exploitation-exploration trade-offs**. *J Mach Learn Res* 2003:397–422, https://doi.org/10.1162/153244303321897663.

50. Snoek J, Larochelle H, Adams RP: **Practical Bayesian optimization of machine learning algorithms**. *Adv Neural Inf Process Syst* 2012, **4**:2951–2959.

51. Sutton RS, Barto AG: *Reinforcement learning: an introduction*. MIT Press; 2018.

52. Angermueller C, Dohan D, Belanger D, Deshpande R, Murphy K, Colwell LJ: **Model-based reinforcement learning for biological sequence design**. *Int Conf Learn Rep* 2020.

53. Srinivas N, Krause A, Kakade S, Seeger M: **Gaussian process optimization in the bandit setting: No regret and experimental design**. *Int Conf Machine Learn* 2010, **27**, https://doi.org/10.1109/TIT.2011.2182033.

54. * Hie B, Bryson BD, Berger B: **Leveraging uncertainty in machine learning accelerates biological discovery and design**. *Cell Syst* 2020, **11**:461–477, https://doi.org/10.1016/j.cels.2020.09.007.
This paper uses a Gaussian process surrogate model with neural sequence embeddings to optimize for compound-kinase binding affinity and protein fluorescence while incorporating uncertainty into the acquisition function.

55. Frisby TS, Langmead CJ: **Fold family-regularized bayesian optimization for directed protein evolution**. In *Leibniz international proceedings in informatics*, vol. 172. LIPIcs; 2020, https://doi.org/10.4230/LIPIcs.WABI.2020.18.

56. Wilson JT, Hutter F, Deisenroth MP: **Maximizing acquisition functions for Bayesian optimization**. *Adv Neural Inf Process Syst* 2018, **31**:9884–9895.

57. Rasmussen CE, Williams CKI: *Gaussian processes for machine learning*. MIT Press; 2005.

58. Kuss M, Rasmussen CE: **Assessing approximations for Gaussian process classification**. *Adv Neural Inf Process Syst* 2006:699–706.

59. Bonilla EV, Chai KMA, Williams CK: **Multi-task Gaussian process prediction**. *Adv Neural Inf Process Syst* 2009, **20**:153–160.

60. * 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, https://doi.org/10.1038/s41592-019-0583-8.
This paper uses Gaussian process classifiers and regressors as surrogate models to design channelrhodopsins based on multiple optimization properties.

61. Voutilainen S, Heinonen M, Andberg M, Jokinen E, Maaheimo H, Pääkkönen J, Hakulinen N, Rouvinen J, Lähdesmäki H, Kaski S, Rousu J, Penttilä M, Koivula A: **Substrate specificity of 2-deoxy-D-ribose 5-phosphate aldolase (DERA) assessed by different protein engineering and machine learning methods**. *Appl Microbiol Biotechnol* 2020, **104**:10515–10529, https://doi.org/10.1007/s00253-020-10960-x.

62. Micchelli CA, Xu Y, Zhang H: **Universal kernels**. *J Mach Learn Res* 2006, **7**:2651–2667.

63. Oh C, Tomczak JM, Gavves E, Welling M: **Combinatorial Bayesian optimization using the graph cartesian product**. *Adv Neural Inf Process Syst* 2019, **33**.

64. Beck D, Cohn T: **Learning kernels over strings using Gaussian processes**. *Int Joint Conf Nat Lang Process* 2017, **2**:67–73.

65. Moss HB, Beck D, González J, Leslie DS, Rayson P: **BOSS: Bayesian optimization over string spaces**. *Neural Inform Process Syst* 2020, **34**.

66. Liu H, Ong YS, Shen X, Cai J: **When Gaussian process meets big data: a review of scalable GPs**. *IEEE Trans Neural Network Learn Syst* 2020, https://doi.org/10.1109/TNNLS.2019.2957109. arXiv:1807.01065.

67. Fortuin V, Dresdner G, Strathmann H, Rätsch G: *Scalable Gaussian processes on discrete domains*. 2019. arXiv stat.ML (1810.10368).

68. Koller D, Friedman N: *Probabilistic graphical models: principles and techniques*. MIT Press; 2009.

69. Neal RM: *Bayesian learning for neural networks*. Springer Science & Business Media; 2012.

70. Lakshminarayanan B, Pritzel A, Blundell C: **Simple and scalable predictive uncertainty estimation using deep ensembles**. *Adv Neural Inf Process Syst* 2017:6402–6413.

71. Zeng H, Gifford DK: **Quantification of uncertainty in peptide-MHC binding prediction improves high-affinity peptide selection for therapeutic design**. *Cell Syst* 2019, **9**:159–166, https://doi.org/10.1016/j.cels.2019.05.004.

72. Amini A, Schwarting W, Soleimany A, Rus D: **Deep evidential regression**. *Adv Neural Inf Process Syst* 2019, **33**:14927–14937.

73. Izmailov P, Vikram S, Hoffman MD, Wilson AG: *What are Bayesian neural network posteriors really like?*. 2021. arXiv cs.LG (2104.14421).

74. Pierce NA, Winfree E: **Protein design is NP-hard**. *Protein Eng* 2003, **15**:779–782, https://doi.org/10.1093/protein/15.10.779.