Protein KABOOM: Kermut-Aided Bayesian Optimization Of Mutations

Mads Herbert Kerrn and **Wouter Boomsma** *University of Copenhagen*
{make, wb}@di.ku.dk
Henry Moss *University of Cambridge*

Abstract

Machine learning has shown promise in enhancing protein engineering, particularly through guided selection in directed evolution. This study introduces a Bayesian optimization alternative to directed evolution. We propose using a Gaussian process with the recent Kermut kernel to guide the suggestion of protein variants, leveraging uncertainty estimates through Thompson sampling. Optimizing against established prediction tools for protein stability and solvent accessible surface area, we demonstrate that our Bayesian optimization framework consistently identifies superior protein variants compared to other methods, including traditional directed evolution, zero-shot models, and existing ML-guided directed evolution procedures.

Keywords: Protein Engineering, Gaussian Process, Bayesian Optimization

1 Introduction

Machine Learning has long been described as holding considerable promise to advance protein engineering, particularly in actively learning frameworks that leverage insights from past experiments to guide future ones. However, the feedback loop inherent in directed evolution — where the best candidate is selected in each iteration from variants generated by random mutagenesis — establishes a robust baseline even with the random introduction of mutations. Furthermore, pre-trained models increasingly demonstrate strong zero-shot performance, further enhancing non-adaptive approaches that do not update during experimental rounds. This raises a natural question: how much additional improvement can be achieved by iteratively refining a surrogate function? Additionally, how critical is the choice of regression algorithm, and are current models capable of providing uncertainty estimates that are genuinely effective in a Bayesian optimization context?

In this work, we propose KABOOM (Kermut-Aided Bayesian Optimization Of Mutants); a Bayesian Optimization (BO) framework for batched optimization of a wild type protein using Kermut as surrogate model. While many existing BO methods for protein engineering (e.g. LaMBO of Stanton et al. [2022a]) evaluate their acquisition functions in a local region of the latent space around a reference protein, KABOOM explores the space of variants within a number of mutations in a more exhaustive, greedy fashion. In this paper we

AI in Science
Conference 2025
EU2025DK

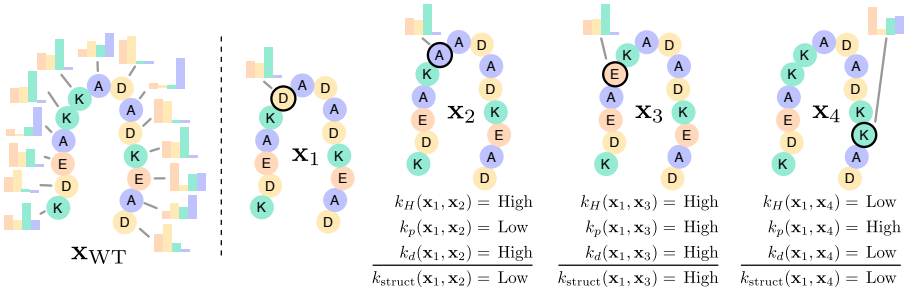

Figure 1: Figure from Groth et al. [2024]. Overview of Kermut's structure kernel. Using an inverse folding model, structure-conditioned amino acid distributions are computed for all sites in the reference protein. The structure kernel yields high covariances between two variants if the local environments are similar, if the mutation probabilities are similar, and if the mutates sites are physically close. Constructed examples of expected covariances between variant $\mathbf{x}_1$ and $\mathbf{x}_{2,3,4}$ are shown.

present KABOOM and demonstrate how it can handle large search spaces of protein variants. Moreover, we demonstrate that KABOOM achieves better solutions than existing baselines including directed evolution.

2 Methods

We develop a Bayesian optimization procedure using Kermut as surrogate model and a greedy variant of Thompson sampling as our acquisition function. See [Frazier, 2018] for a tutorial on Bayesian optimization.

2.1 Kermut

To predict protein variant effects, we rely on Gaussian process regression using the Kermut kernel [Groth et al., 2024]. For a comprehensive overview, see Rasmussen and Williams [2006], which this section is based on.

Kermut [Groth et al., 2024] is a kernel designed for predicting the effects of mutations on protein properties. The kernel is a weighted sum of two kernels that take protein sequences as inputs: A sequence kernel k_{seq} and a structure kernel k_{struct}.

$$k(\mathbf{x}, \mathbf{x}') = \pi k_{\mathrm{struct}}(\mathbf{x}, \mathbf{x}') + (1 - \pi) k_{\mathrm{seq}}(\mathbf{x}, \mathbf{x}'). \tag{1}$$

k_{seq} uses the RBF kernel on an embedding of the sequence obtained using a protein language model (ESM2; Lin et al. [2023]).

k_{struct} compares the local environment of the mutated residues using the structure of the wild type along with the distributions over amino acids given by an inverse folding model.

For two variants having mutations given by the sets M and M', k_{struct} is a double sum over the mutations, where k^1_{struct} operates on the single mutant variants:

$$k_{\text{struct}}(\mathbf{x}, \mathbf{x}') = \sum_{i \in M} \sum_{j \in M'} k^1_{\text{struct}}(\mathbf{x}^i, \mathbf{x}'^j) \tag{2}$$

k^1_{struct} compares the distributions at the mutated sites given by an inverse folding model (ProteinMPNN; Dauparas et al. [2022]) using three components. First, k_H is the Hellinger kernel [Michael et al., 2024], which compares the distributions at the mutated sites. Second, k_p uses the exponential kernel to compare the log probabilities of the amino acids to which the residues have been mutated. Lastly, k_d applies the exponential kernel to the euclidean distance between the mutated sites.

$$k^1_{\text{struct}}(\mathbf{x}, \mathbf{x}') = \lambda k_H(\mathbf{x}, \mathbf{x}') k_p(\mathbf{x}, \mathbf{x}') k_d(\mathbf{x}, \mathbf{x}'), \tag{3}$$

All parameters are optimized using gradient descent.

2.2 Handling the search space

The space of protein variants within a limited number of mutations from a wild type constitutes a vast combinatorial space. For a protein sequence of length L, an amino acid alphabet of 20 characters, and a mutation limit of 6, the total number of possible variants is given by $\binom{L}{6}(20-1)^6 + \binom{L}{5}(20-1)^5 + ... + \binom{L}{1}(20-1)^1$. For $L = 50$, this number exceeds 10^{14}.

One approach to explore this immense space is to use a differentiable model that operates on a continuous embedding of the protein sequences. However, if the model works directly in the discrete protein space, standard Bayesian optimization (BO) becomes computationally infeasible, as evaluating the acquisition function for all possible variants would be prohibitively expensive.

Our proposed solution employs a greedy approach to efficiently sample candidates for acquisition function evaluation. Initially, the acquisition function is evaluated for all single-mutant variants of the wild type, and the highest-scoring candidate is selected as the basis for generating second-order mutants. Subsequently, the acquisition function is evaluated on all second-order mutants derived from the best first-order mutant, and the process is repeated iteratively.

Once variants with six mutations are identified, we revisit and reevaluate the first five mutations in a random order to search for improved variants that incorporate the other mutations. This greedy approach significantly reduces computational requirements, as it only involves $L \cdot 20 \cdot (6 + 5)$ evaluations. For $L = 50$, this corresponds to approximately 10^4 evaluations, making it computationally feasible.

3 Results

Wet-lab experiments are often expensive and time-consuming, making computational proxies a practical alternative for benchmarking optimization methods.

The Poli library provides computational proxies for protein stability and solvent-accessible surface area (SASA), based on FoldX [Schymkowitz et al., 2005, Stanton et al., 2022b, González-Duque et al., 2024]. We use these proxies as oracle functions and maximize their values for a given wild type protein by applying substitution mutations. To prevent generating dysfunctional proteins that deviate significantly from the wild type, the search space is typically restricted to a maximum number of mutations. In our case, we limit the search space to six mutations. Additionally, as wet-lab experiments are often conducted in batches of size 96 [Tsuboyama et al., 2023], we adopt this as our batch size.

We optimize the stability and solvent accessible surface area (SASA) for two wild type proteins from the ProteinGym dataset: DNJA homolog and RfaH carboxyterminal domain.

Table 1: Best value found for four different computational protein engineering oracles averaged across ten repetitions (higher is better). Standard error is shown in parenthesis. KABOOM finds the most optimal protein variants across all four tasks.

	DNJA (Stability)	DNJA (SASA)	RfaH (Stability)	RfaH (SASA)
KABOOM	**34.9** (0.2)	**5374.0** (14.7)	**15.7** (0.2)	**4533.8** (8.6)
DE	32.8 (0.3)	5301.4 (13.5)	14.6 (0.2)	4439.7 (11.9)
LaMBO-2	26.9 (0.4)	5196.8 (17.9)	9.4 (0.3)	4227.8 (17.2)

Table 1 presents the results for KABOOM, Directed Evolution (DE), and LaMBO-2 across four protein engineering tasks. KABOOM outperforms both DE and LaMBO-2 consistently across all tasks, with DE achieving the second-best performance.

4 Conclusions

Our study investigates the application of Bayesian Optimization (BO) with Gaussian Process (GP) regression, leveraging the recent Kermut kernel, to enhance the efficiency of protein engineering. The primary comparison centers on the proposed BO framework, KABOOM, against LaMBO-2 and Directed Evolution (DE). KABOOM finds the best candidates among the tested methods.

The study illustrates the significant potential of Bayesian optimization in protein engineering, providing a robust alternative to traditional directed evolution methods. The results advocate for the broader application of BO in experimental design, particularly where budget constraints and the need for efficient exploration of large combinatorial spaces are critical factors.

References

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615): 49–56, Oct. 2022. doi: 10.1126/science.add2187.

P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

M. González-Duque, S. Bartels, and R. Michael. poli: a libary of discrete sequence objectives, Jan. 2024. URL `https://github.com/MachineLearningLifeScience/poli`.

P. M. Groth, M. H. Kerrn, L. Olsen, J. Salomon, and W. Boomsma. Kermut: Composite kernel regression for protein variant effects. *bioRxiv*, pages 2024–05, 2024.

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

R. Michael, S. Bartels, M. González-Duque, Y. Zainchkovskyy, J. Frellsen, S. Hauberg, and W. Boomsma. A Continuous Relaxation for Discrete Bayesian Optimization, 2024.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9.

J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, and L. Serrano. The FoldX web server: An online force field. *Nucleic Acids Research*, 33(suppl_2): W382–W388, July 2005. ISSN 0305-1048. doi: 10.1093/nar/gki387.

S. Stanton, W. Maddox, N. Gruver, P. Maffettone, E. Delaney, P. Greenside, and A. G. Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning*, pages 20459–20478. PMLR, 2022a.

S. Stanton, W. Maddox, N. Gruver, P. Maffettone, E. Delaney, P. Greenside, and A. G. Wilson. Accelerating Bayesian Optimization for Biological Sequence Design with Denoising Autoencoders. *arXiv:2203.12742 [cs, q-bio, stat]*, Mar. 2022b.

K. Tsuboyama, J. Dauparas, J. Chen, E. Laine, Y. Mohseni Behbahani, J. J. Weinstein, N. M. Mangan, S. Ovchinnikov, and G. J. Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023.