

UC Berkeley

Berkeley Scientific Journal

Title

Tailoring AI for Biological Objectives: ProteinGuide and Machine Learning-Guided AAV Vector Design

Permalink

<https://escholarship.org/uc/item/7ns32705>

Journal

Berkeley Scientific Journal, 30(1)

ISSN

1097-0967

Authors

Arun, Samika
Tran, Catherine

Publication Date

2026-10-30

DOI

10.5070/BS3.69981

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NonCommercial-NoDerivatives License, available at <https://creativecommons.org/licenses/by-nc-nd/4.0/>

Tailoring AI for Biological Objectives: ProteinGuide and Machine Learning-Guided AAV Vector Design

INTERVIEW WITH PROFESSOR JENNIFER LISTGARTEN

BY SAMIKA ARUN & CATHERINE TRAN



Jennifer Listgarten is currently a Professor in the Department of Electrical Engineering and Computer Science, the Center for Computational Biology, and the Bioengineering program at the University of California, Berkeley, where she also holds the Jeffrey Huber and Angel Vossough Chancellor's Chair in Computational Biomedicine. She

is a member of the steering committee for the Berkeley AI Research (BAIR) Lab. Jennifer's research interests are broadly at the intersection of machine learning, applied statistics, molecular biology and science. Her current research is primarily in understanding how machine learning can be used to advance protein engineering. Academically, She completed two undergraduate degrees, one in Physics and one in Computer Science, from Queen's University in Canada and went on to complete her Ph.D. in the machine learning group in the Department of Computer Science at the University of Toronto, located in her hometown. Professionally, she worked at Microsoft Research from 2007 to 2017.

BSJ: In recent years, we have seen foundation models take off within protein design. What drew you to exploring guided protein generation as the next step in your research?

JL: The reality is your students pull you in different directions. My students pulled me in the direction of guided protein generation. Credit goes out to graduate students Hunter Nisonof, Junhao (Bear) Xiong, and Stephan Allenspach. Hunter was the ringleader of getting everyone into it, he thought there was an interesting and useful problem to be solved. And in fact, for various technical reasons, I was a bit suspicious of the area and we had some fun discussions arguing what the merits were for pursuing this. Ultimately, I think it was nice and useful to pursue. At the same time, I think some of my skepticism ultimately bore some fruit. That is how the best research happens. You look at the best research groups in the world, it is not a professor telling their students what to do. Some people may quibble with that but I think if you want creativity, passion, and spark, you need to find people that enjoy working together, that bring different skills to the table, and then let them do their thing and help them do their thing.

BSJ: What are generative models in protein engineering?

JL: Generative models, you may know of ChatGPT and so forth, are models that can generate text. These are analogies for protein engineering, where they can generate something that looks like text, proteins would generate something that looks like a sequence of amino

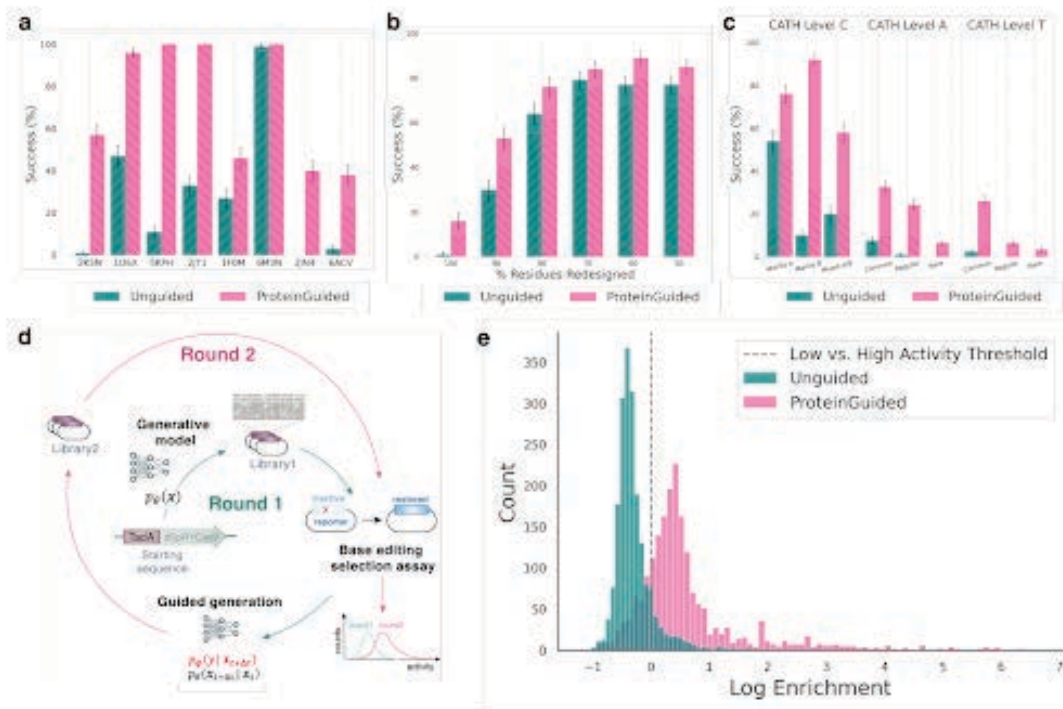


Figure 1: Experimental and computational validation of ProteinGuide to enhance protein design. Protein sequences designed using ProteinGuide (pink) are more often predicted to be stable and folded correctly, improving the success rate of designing functional enzyme sequences, and designing new protein backbones (a-c) compared to unguided designs (teal) across various protein structures. Its effectiveness is proven in a two-round experiment where an initial library is used to train ProteinGuide to generate an improved library (d-e).

parameters for a laboratory process that will stochastically synthesize a sequence.

Many models are trained and used in protein engineering, that ultimately produce sequences. For example, you might have some protein like green fluorescent protein (GFP) and want to engineer variants that hold structure and fluoresce at different frequencies. Within these models you would search for evolutionarily related sequences through a standard bioinformatics algorithm and build a generative model on those sequences. One example of such a model is you provide the 3D backbone structure of a protein and ask it to produce a probability distribution over sequences it can sample from. After having trained it on many pairs of structures, it can return sequences that fold up into a provided structure. This is called “inverse folding”, the reverse of the protein folding which predicts structure from sequence.

BSJ: What are the shortcomings of traditional protein engineering models and how does ProteinGuide differ from them?

JL: Ultimately, traditional models are not inherently useful for protein engineering. If you sample from the pan-protein

“...the point of ProteinGuide is to say people spend all this time and money building fairly general purpose models, and they are sometimes useful on their own, but mostly they are useful if you can tailor them to your problem.”

or foundation model, you get variations, but cannot optimize for properties. What if you are trying to make it more stable? Or improve stability at a higher pH. All you can do is sample from it. I like, sampled some proteins, and maybe if I made them in the lab, they might fold and be kind of stable. In protein engineering, you want to tailor it to have certain desired properties that may be related to therapeutics or environmental applications. And so what you would like to do is to condition on some property. So instead, unconditional generation or limited conditioning of the inverse folding model, you statistically condition on the backbone. You might know the desired property beforehand, but in general, it is not known upfront when you train that model. New data through wet lab experiments may prompt you to fold the protein structure back into one of these models under a new property condition. For example, after measuring the catalytic activity for an enzyme, you can retrain the model using that data to generate sequences with higher catalytic efficiency.

The idea of the ProteinGuide is instead of retraining the generative model, guiding the generation towards the property of interest. Almost all of these modern day generative models have an iterative way to sample where the final product is built up step-by-step. So in between each step, ProteinGuide evaluates information from the model’s sequence with the desired property in mind. For example, if the model is building a protein from three amino acids, the guide will look at the three and predict which next amino acid would produce a sequence with higher catalytic activity. The predictive model effectively modulates the probabilities of what sequence letter goes next.

And so really,

Our paper is about leveraging some of our earlier results combined with some new results that are theoretically and statistically correct

and practical at the same time. This is compared to a lot of modern day machine learning that are random shots in the dark that sometimes work but are hard to reason about.

BSJ: Regarding the application of protein guides, did you see any barriers or challenges with translating it from in silico demos to wet lab settings?

JL: There are always challenges when you develop a method, maybe you do some in silico and wet lab experiments where the practical bottleneck is going to hit. I always tell my students to have one eye in the sky, a machine learning project that does not depend on a wet lab. So that their assay is not working or it is taking another six months, you are not just wasting your PhD. And at the same time, always have one of those to keep yourself honest. You are working on a real problem that someone cares about because they are bothering to spend hours in the lab and money to do it. And there are always hiccups when you apply things, which are great in a way, because then you can think about what the important problems are.

BSJ: You presented protein guide as a framework to use in multiple model families, such as MLMs, diffusion models, autoregressive models. Do you anticipate these unified frameworks becoming the standard between AI foundation models and biological objectives?

JL: In this paper, we have formalized and proven some equivalences between training objectives of models that people have often thought of as quite different. So a mass language model and a diffusion model, and some broad set of some broad scope can be made, can be seen as exactly equivalent in their training objectives and their sample. Certainly, we are not the only people to have recognized this, but to the best of my knowledge, we are the first people to really write it cleanly, all of it, including the equivalence and sampling, not just in training. In addition to having a nice exposition of the equivalences, what is this bias? I do not think very many people have hammered that point home particularly well. So I think we have been learning as we do this project as well. People have been wanting to basically guide or condition or tailor certain models, like protein, MPN in particular, and maybe to a lesser extent, ESM three. They have been really wanting to bring their own experimental data into these models. And so until we put this together, I would argue that people knew how to do that, either in ad hoc ways and or in ways that require training a new generative model, which is potentially more complicated. I do not want to dismiss it, because it is still a useful method, but we offer a complementary perspective that has pros and cons potentially to each, but we have at least thrown this new toolkit out to the community so they can try

“Infusing AI or machine learning into this process helps search through this space more cleverly than you would by just simply making random mutations.”

this. And maybe there are places where retraining the general model is helpful and you want to do that. I do not think we have fleshed that out super well, but that we have some thinking about it that we have not yet written up.

BSJ: This next question is based on a paper that you co-authored, “Optimal trade-off control in machine learning-based library design, with application to adeno-associated virus (AAV) for gene therapy.” To provide some background, what are adeno associated virus variant libraries? How are they generated, and in what way are they utilized in gene therapy research?

JL: When you do directed evolution, without AI or with AI, you start with an initial parent protein. For example, it could be a green fluorescent protein as a canonical example. And then you decide, instead of fluorescing green, I want it to fluoresce yellow. The next step is to induce some set of mutations into it at random, so you are left with variants centered on the original one. You would only include one or two mutations, because if you have more than that the process won’t work. Depending on the particular engineering campaign and technology being used, you will be left with anywhere from hundreds to tens of millions of sequence variants. Then, you can screen them for your feature of interest. Going back to the green fluorescent protein example, if by accident some of these mutations make the protein fluoresce yellow, you can grab those variants with some kind of physical selection, and then repeat the process. This process is limited by the fact that you have to make small moves to move forward. Through this optimization, you’re going to quickly end up in a local optimum, which can be an issue.

There are still many open problems for AAV. For example, at the moment, they can typically only put it in gene therapy behind the

“It’s similar to a lottery ticket game; the more lottery tickets you have, the more likely you are to win.”

blood brain barrier, because otherwise antibodies will shut it down. To address this, you might want to make the AAV vector less antigenic, so that it doesn’t get attacked by the immune system. Additionally, AAV has its own way of targeting certain cell cells, but you might want it to target a different set of cells where you want to deliver the payload. Though there have been some engineering successes, there are still many open problems.

BSJ: How did your research use machine learning to address these challenges and improve the design of AAV variant libraries?

JL: Dave Schaffer’s lab at Berkeley is actively trying to re-engineer AAV vectors for various purposes. To do this, they start with a standard directed evolution campaign through inducing variations to get a library. This is similar to a library of books, but instead of books, it contains different sequences. The issue is that there are so many different ways you can induce this diversity into an initial parent sequence. They found out that a good way to induce diversity is through inserting a 21 nucleotide sequence into the capsid of the virus, which is the outside part that encounters everything in its environment. This 21 nucleotide insert will make the virus behave quite differently, so it introduces a lot of the phenotypic behavior of that AAV. The hope is that you can find an AAV that behaves the way you want—one that goes on to infect glial cells. The idea is that you want AAVs that are

all basically functional, but that differ slightly in how they behave, so that hopefully one of those differences gives you the outcome you're looking for. Maybe, just by chance, one of them ends up infecting glial cells instead of other cell types, which is exactly what you were hoping for, even if you don't yet know the mechanism behind it.

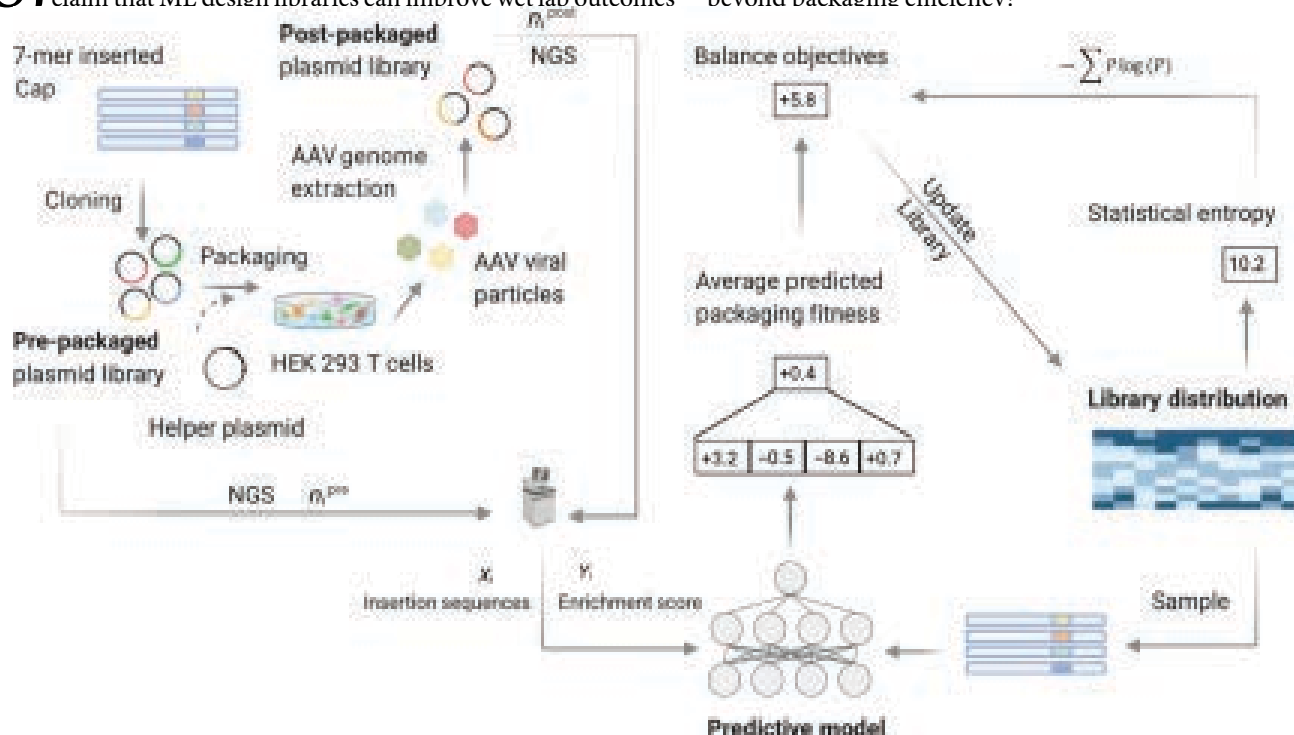
Before he met us, Dave's lab would randomly generate 21-nucleotide DNA sequences. So, essentially, they would make sequences using A, C, T, and G with equal probability at each position, repeated 21 times. In the lab, this means you're stochastically synthesizing these sequences at random, drawing from that uniform distribution. Then you perform a large screen and hope for the best. They've had many successes using this kind of library. However, the issue is that at the beginning of this process, half of the library falls flat on its face. It's like someone sold you lottery tickets that were photocopied in the barcodes and weren't valid. There is some basic functionality you need from an AAV. It has to be able to package its payload and maintain baseline functionality. It turns out that when you insert this 21-nucleotide sequence, even though it introduces many interesting and useful variations, roughly half of the resulting AAVs just completely fail to function. The challenge we set ourselves with Dave was to see if we could use machine learning to analyze his current libraries to understand which 21 nucleotide inserts produce functional versus non functional AAV. Our goal was to design a starting library that was as close to uniform as possible, subject to the fact that all the lottery tickets, or AAV variants, were valid. Of course, none of this is black and white – it is all probabilistic. Essentially, we created a new starting library that has just as much diversity and covers as much sequence space as previous libraries, however, it has more functional variants.

BSI: How did the experimental results of this paper support the claim that ML design libraries can improve wet lab outcomes

JL: What I described was a technique to make a new, general-purpose starting library for AAVs. Implicitly, I hadn't said it before, but now I'll say it explicitly — this library is meant to be useful for any downstream selection or engineering goal. Whether you want an AAV that targets glial cells or one that avoids them; whether you want it to be immunogenic or not, whatever the objective is, this should be a better starting library. That's because we didn't tailor it to any one outcome; we simply said to make it as uniform as possible, but still a basically functional AAV. The real test would be to take some downstream engineering goal that our library wasn't designed for, and show that our library, compared to the one previously used, gives you a much better chance of success — more “winning lottery tickets,” so to speak. After all, in drug development, you don't just want one winning lottery ticket.

Ultimately, you need many different properties like that. You always want to keep as many lottery tickets around, because they're going to keep getting filtered out as you try to get more and more of your property satisfied. Our final challenge was to do a series of steps to demonstrate why we chose this particular library based on these hyperparameters and design choices. But at the end of the day, the totally blind test was to use our library and the library Dave previously used to try to infect these brain cells in non-human primates, in collaboration with Tomasz Nowakowski at UCSF. It is a little tricky to describe how this was quantified, but ultimately, you could see that our library was getting more hits into those brain cells.

BSJ: How might this ML-guided library design approach be extended or adapted to other engineering perspectives beyond packaging efficiency?



compared to conventional libraries?

Figure 2: Experimental workflow for generating pre- and post-packaged AAV insertion library data for machine-learning guided design.

Sequencing counts (N_i) for each variant were used to train a regression model predicting packaging success, which was then inverted to design libraries that balance diversity with predicted packaging efficiency.

JL : We designed a library that had the highest expected or mean packaging efficiency according to our predictive model. This was subject to a diversity constraint, because going back to the prior example, you do not want the same lottery ticket over and over. When you're designing these libraries, you can turn a knob that balances the amount of diversity and how well in on average it scores in our predictive model for packaging efficiency. You can ask, do I want it to score on average as good as possible? Or do I just want the top 10% to score as well as possible? It also gets interesting when we only change a small part of the 21 nucleotide insert.

So it gets even more interesting. What if I'm actually diversifying the entire genome, and the variation isn't well-covered by typical read lengths, or the sequencing is just too expensive? Then you run into this issue where you only get scattered read coverage across the genome, and you don't automatically know how those fragments fit together. But what you actually want to determine is something like the packaging efficiency for an AAV virus. And instead, what you end up with are these short reads that only provide fragments of the full variants. So you could try using bioinformatics methods to essentially "stitch" them back together and hope that the reconstructed sequences represent real variants. And then you say, "Okay, that's a variant, and now I know its packaging efficiency." There are ways to do this, but it's definitely nontrivial.

BSJ : Where do you think the field of AI and ML in biology will be in the next few years?

JL : I would give credit to Frances Arnold, because many years ago—long before I started doing any of this—her group actually brought Gaussian process regression into directed evolution. You can look at papers with Phil Romero and Frances Arnold, who of course won the Nobel Prize for directed evolution. I think this might have been in the 90s or thereabouts. And I've always felt like they were probably the first group to really do something that looked like machine learning in this space. So her lab deserves credit not only for the work that led to the Nobel Prize, but also for thinking early about how AI could be used to guide directed evolution in different ways. Meanwhile, groups like the Baker lab were mostly using biophysics-based modeling approaches until relatively recently. It has definitely been a pretty big jump from then to now.

I think AI has definitely changed our lives—most of us use large language models daily now for small things. AI will remain useful, but I think the intensity will calm down. AlphaFold really was a true breakthrough, though. We will occasionally see moments like that, but I do not think we're going to see that level of success across every problem. I actually have a perspective piece that I think explains why AlphaFold was such a special case, and why we shouldn't expect that kind of phase transition everywhere. It's called "*The Perpetual Motion Machine of AI-generated Data and the Distraction of ChatGPT-as-scientist*."

"AI is going to keep infusing everything, including biology."

Overall, I'm optimistic.

But I think major leaps—the AlphaFold-scale leaps—will be rare. Progress will mostly be steady and incremental.

BSJ : How has your research translated to your personal life and growth, and have you changed any practices because of the things that you've learned?

JL : For me, what has shaped me most as a human being was when I transitioned from industry to be a faculty here and took on a mentoring role of PhD students. It is an incredible privilege to be around these amazing PhD students and watch them come in during formative years and find their way through their PhD. I think it is the human connections that are life changing and transformative. In academia, you always have this influx of students that are hungry for the next thing, and then they graduate, and you miss them, but ultimately it is good to constantly have fresh air in the system. I came to academia very late in life, and I came because I just wanted to do great research with good people. Then, I discovered that half the beauty is the relationship with your PhD students and your research group. That to me is kind of the metaphor for life. It is about doing the best you can do by somebody else when you also have your own needs and objectives. I think it is like a microcosm of living life at large.

REFERENCES

1. Xiong, J.; Nisonoff, H.; Lukarska, M.; Gaur, I.; Oltrogge, L. M.; Savage, D. F.; Listgarten, J. (2025). Guide your favorite Protein sequence generative model. [Preprint]. arXiv. doi.org/10.48550/arXiv.2505.04823
2. Zhu, D.; Brookes, D.H.; Busia, A.; Carnerio, A.; Fannjiang, C.; Popova, G.; Shine, D.; Donohue, K. C.; Lin, L. F.; Miller, Z. M.; Williams, E. R.; Chang, E. F.; Nowakowski, T. J.; Listgarten, J.; Scharffer, D. V. (2024). Optimal trade-off control in machine learning-based library design, with application to adeno-associated virus (AAV) for gene therapy. *Science Advances* 2024, 10(4), eadj3786. doi.org/10.1126/sciadv.adj3786

IMAGE REFERENCES

1. Figure 1: Xiong, J.; Nisonoff, H.; Lukarska, M.; Gaur, I.; Oltrogge, L. M.; Savage, D. F.; Listgarten, J. (2025). Guide your favorite Protein sequence generative model. [Preprint]. arXiv. doi.org/10.48550/arXiv.2505.04823
2. Figure 2: Zhu, D.; Brookes, D.H.; Busia, A.; Carnerio, A.; Fannjiang, C.; Popova, G.; Shine, D.; Donohue, K. C.; Lin, L. F.; Miller, Z. M.; Williams, E. R.; Chang, E. F.; Nowakowski, T. J.; Listgarten, J.; Scharffer, D. V. (2024). Optimal trade-off control in machine learning-based library design, with application to adeno-associated virus (AAV) for gene therapy. *Science Advances* 2024, 10(4), eadj3786. doi.org/10.1126/sciadv.adj3786