Machine Learning and Design Optimization for Molecular Biology and Beyond

Interview with Dr. Jennifer Listgarten

BY BRYAN HSU, NATASHA RAUT, KAITLYN WANG, AND ELETTRA PREOSTI

Jennifer Listgarten is a professor in the Department of Electrical Engineering and Computer Science and a principal investigator in the Center for Computational Biology at the University of California, Berkeley. She is also a member of the steering committee for the Berkeley AI Research (BAIR) Lab, and a Chan Zuckerberg investigator. In this interview, we discuss her work in the intersection of machine learning, applied statistics, and molecular biology.

BSJ: You have a very diverse background ranging from machine learning to applied statistics to molecular biology. Can you tell us how you came to start working on design optimization?

JL: It was very unplanned, actually. I had been working in the fields of statistical genetics and CRISPR guide design for some time, so I wanted to look for something really crazy and different. That summer, a graduate student intern and I wondered if we could predict the codon usage in actual organisms with modern day machine learning. That was totally crazy and not necessarily useful, but I thought it might shed some interesting biological insights. Is codon usage predictable, and if so, what enables you to predict it? Is it just the organism or also the type of gene?

From there, we moved to codon optimization using more sophisticated modeling techniques and ideally ingesting more data to make use of those techniques. I approached my colleague, John Dunwich, and we started working on this very concrete problem. I came up with a ridiculous idea: what if I just think about finding sequences of amino acids or nucleotides that will do what I want them to do in a general way? Of course, I was aware that there were decades' worth of research done to answer this question in terms of biophysics based modeling. David Baker's lab at the University of Washington, for example, built energy based models. But, I thought that we should use machine learning. I talked to a lot of people, convinced some students to work on this, and now, I think this is my favorite research area that I have ever worked in.

BSJ: Can you provide a general overview of how machine learning methods such as neural networks are applied to successfully optimize small molecule drug discovery?

JL: The general way to think about this is that machine learning methods can be used to build an *in silico* predictive model for measuring things. Measuring quantities in a lab can oftentimes be tricky and require creativity because you cannot always exactly measure what you want. Typically, a proxy is first used to scale things. Then, the correlation between the proxy and the quantity which we want to measure to scale must be understood. But, what if we can have a predictive model to reduce the number of measurements needed? Maybe instead of having to take a thousand measurements, we can get away with taking fifty or a hundred measurements at a particular location and time during the experimental process. This would be a tremendous saving in many senses of the word.

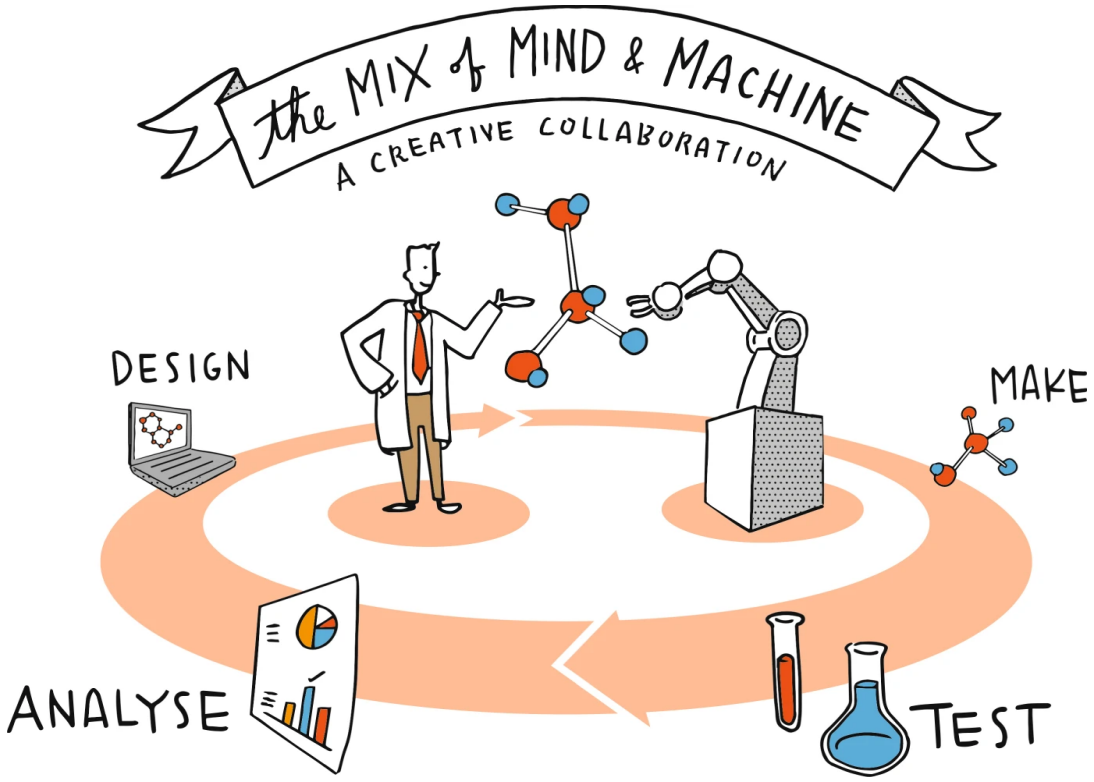

Figure 1: Integrating mind and machine in drug discovery. *While machines and machine learning models are capable of making and testing designs, they do not yet have the capability to create these designs or derive meaningful conclusions from extensive data analysis. However, as the fields of computational biology and chemistry continue to progress, the collaboration between mind and machine may drastically change scientific research as we know it.*[2]

BSJ: Do you think that there will come a point in time in which machines can fully take over the analysis and design processes?

JL: The general answer is no. I think our work is unlike natural language processing, computer vision, and speech, where the benchmarks of machine learning have been blown away by deep neural networks. What distinguishes these three areas from computational biology and chemistry is that it is easy to obtain data in these areas. For example, you can trivially take a gazillion images or snippets of speech from people. You can also have an ordinary human annotate this data since most of us are born with brains that can comprehend and make sense of it. Therefore, getting the labels required for machine learning is really easy. However, you cannot do this in chemistry and biology. You have to spend a lot of time and money in the lab and use your ingenuity to measure the quantities you care about. Even then, it is an indirect measurement. So, the data problem itself is inherently much trickier. For this reason, I think there is no way we are going to replace domain experts.

"Machine learning methods can be used to build an *in silico* predictive model for measuring things."

The question becomes: how can we synergistically interact with each other? For example, as a machine-learning person, I must decide which data an experimenter should grab in order to help me build a good machine learning model. The machine learning model would in turn make more useful predictions. On the other hand, an experimenter might have considerations about how difficult it is to measure one quantity compared to another that is additional to what the machine learning model indicates.

Overall, I think that there are so many difficult, complex problems that it will take a very long time, if ever, before humans are out of the loop.

BSJ: Some of your past work focused on developing algorithms in order to predict off-target activities for the end-to-end design of CRISPR guide RNAs. Why is optimizing guide RNAs important for CRISPR-Cas9?

JL: In CRISPR-Cas9, the Cas9 enzyme resembles a Pac-Man. The "Pac-Man" comes in, pulls apart a double strand of DNA, and makes a cut. After which, a native machinery attempts to fix the cut. However, since the native machinery is not very good in fixing the cut, it actually disables the gene. But, if you can deliver the "Pac-

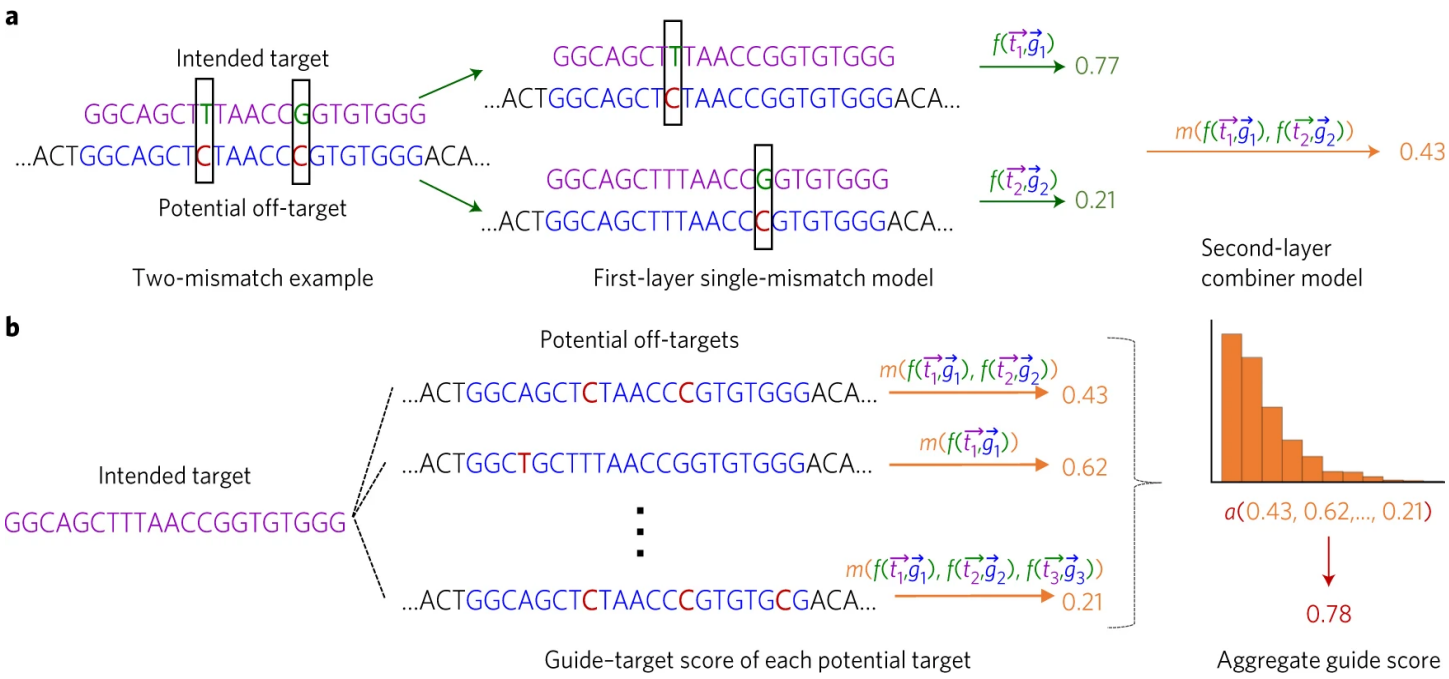

Figure 2. Schematic of Elevation off-target predictive modelling. a. *A visual walkthrough of how Elevation would score a pair of gRNA target sequences with two potential off-target mismatches. The sequences are first separated into two cases. Then, they are scored by the first-layer model, which deals specifically with single mismatches. Elevation evaluates using the Spearman correlation, which weights each gRNA-target pair by a monotonic function of its measured activity in the cell. Next, the second-layer model combines the two scores. In neural networks, a layer is a container that transforms a weighted input with non-linear functions before passing the output to another layer for further evaluation.* **b.** *A closer look at the second-layer aggregation process. The model statistically computes an input distribution of all the single mismatch scores and derives the final score accordingly.*[3]

Man" to the right part of the gene, it is more likely to get messed up without repairing itself. That is how you get a gene knockout. So, the question becomes: how do I deliver the "Pac-Man" to the right part of the gene?

This is where the guide RNA comes in. The guide RNA attaches itself to the "Pac-Man" and brings it to a specific part of the genome. It does so on the basis of complementarity between the guide RNA and the human genome since the guide RNA cannot latch onto anything other than the unique sequence to which it is complementary. But, if the target sequence is not unique or has certain thermodynamic properties, the guide RNA may end up attaching itself to other parts of the genome. Thus, if you are trying to conduct a gene knockout experiment, and you design the wrong guide RNA, you might draw the wrong biological conclusions since you have actually knocked out other genes as well.

BSJ: Can you briefly describe the Elevation model, and how it overcomes the limitations of current prediction models?

JL: There are two things that you care about when it comes to Elevation. The first is that if I am using a guide RNA (gRNA) to knock out a target gene, I want to know what other genes I have knocked out. In order to do this *in silico*, I need a model that, given a guide RNA and part of the genome, gives us the probability that we have accidentally knocked out a gene in that part of the genome. Then, I would need to run the model along the genome at every position that I am worried about. The second important thing is the aggregate of all the probability scores. With what I have told you so far, the model will return three billion numbers, each of which is the probability of an accidental knock out. No biologist, when considering one guide RNA, wants to look at three billion numbers. So, how can we summarize these numbers in some meaningful way?

The way we solved this problem is by training the model on viability data so that we can measure a sort of aggregate effect. To do this, we target a non-essential gene such that if we were successful in knocking the gene out, the organism would still survive. This means that if we choose a bad guide RNA, and it knocks out other genes by accident, it is going to kill the organism. The organism's survival rate gives us an aggregate indirect measurement of how much of a target there is. So, now there are some larger number of predictions from the first layer of the model, although not quite three billion. These predictions then get fed into the aggregate model, which has its own supervised label in a wet lab. This was our crazy compound approach.

However, a big challenge was the very limited data that was available at the time we developed this model. Because there was so little data, I could not just throw deep neural networks at the problem. We basically had to create new approaches to deal with this problem based on standard, simple models.

"That is why it is so beautiful; it is not very obvious until you see it. I think those are often the nicest kinds of results."

BSJ: What are the universal benefits of creating a cloud-based service for end-to-end guide RNA design?

JL: To be a successful researcher in computational biology, you typically need to make tools that people can immediately use. Now, I do not know if our CRISPR tool is such a tool, or if it was and has been superseded. Modern-day molecular biology is so heavily dependent on elements of data science and machine learning, but sometimes people who have the skills to develop them do not have the time or the bandwidth. You cannot reinvent the wheel constantly, right? Science progresses more rapidly when researchers build on top of existing tools.

Thus, we released the GitHub source code for our tool, which makes it reproducible and robust. But the core code with the machine learning modeling should be pretty accessible.

BSJ: We also read your recent work about Estimation of Distribution Algorithms (EDAs). What are the core steps of an EDA, and what are EDAs used for?

JL: So, I am not from the mainstream community that works on EDAs, and I think some of them would quibble with my viewpoint. I would say that EDAs are an optimization method in which you do not need to have access to the gradient (rate of change) of the function you are optimizing. They seem to be very widely used in a number of science and engineering disciplines where optimization is an important factor.

First, I have to decide up front what distribution to use that would represent the function, where to start that distribution, and how to move from there. I am not going to follow the gradient. Instead, I am going to draw samples from the distribution and evaluate each sample under *f(x)*. Then, another ingredient is reweighting the samples based on their performance under *f(x)*. We want to throw away the bad points and train a new distribution just with the good points. Finally, the whole process is repeated. It is a lot like directed evolution, but in a computer. I must have a parametric form of the search pathway and a weighting function that will tell me how to modulate evaluations under *f(x)*. Given those two things, everything else follows. Essentially, we are re-estimating the distribution with maximum likelihood estimation.

I have to say it was super cool because we had not heard about EDAs before this project. David and I were trying to tackle this protein engineering problem, and he reinvented this thing that was essentially an EDA, except more rigorously defined. Then, looking at it, we realized we are trying to do directed evolution in silico starting from first principles, which blew my mind.

BSJ: Could you describe the connection between EDAs and Expectation Maximization, and why this connection is important?

JL: It is a very technical connection rather than an intuitive one. The machine learning community usually uses EM to fit data to a model. To illustrate this: if we had some points, we would fit the mean and covariance of a mixture of Gaussian (normal) distributions to those points. In contrast, that is not the fundamental problem for EDAs. The EDA problem is how to find the *x* that maximizes *f(x)*. I am not fitting the function to *x*; I am trying to find a maximum. So, they sound like very different problems, but for technical reasons, you can actually create an analogy that connects them. That is why it is so beautiful; it is not very obvious until you see it. I think those are often the nicest kinds of results.

BSJ: How can the connection be used in research on design optimization?

JL: To be honest, it is not clear to me how to leverage our insight. We thought the connection was so beautiful, and we wanted to write it up and share it with the community to see if they might find it interesting and be able to make use of it. However, we did not spend the time to demonstrate how that connection allows people to do things they could not have otherwise done. That remains to be seen.

BSJ: Since you also have extensive experience in industry, how do you think the field of therapeutics in relation to industry has been impacted by computational protein engineering?

JL: My time in industry was at Microsoft Research, which in my instance was basically like being in academia. Ironically, one of the reasons I moved to academia was so that I could work with companies that cared about drug design. Biotechnology companies doing diagnostics or therapeutics have been trying to use machine learning, but I am not sure that there have ever been any runaway success stories there. Maybe it has been helpful; computation comes in everywhere. A lot of technologies require sequencing to

$$\begin{aligned}\hat{\theta} &\equiv \underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{p(z|\theta)}[f(z)] \\ &= \underset{\theta}{\operatorname{argmax}}\ \log\mathbb{E}_{p(z|\theta)}[f(z)] \\ &= \underset{\theta}{\operatorname{argmax}}\ \mathcal{L}_{EDA}(\theta)\end{aligned}$$

Figure 3: The connection between EDAs and EM presented as a mathematical argument where $\mathbf{f(z)}$ ***is the black-box function to be optimized,*** $\mathbf{p(z|\theta)}$ ***represents the search model, and*** $\mathbf{L_{EDA}(\theta)}$ ***can be thought of as an EDA equivalent to the log marginal likelihood in EM without any observed data*** $\mathbf{x}$.[4]

assess what is happening and sequencing results require a lot of computational biology. But, can you develop a new COVID vaccine using machine learning? I do not think we have seen that kind of thing. However, I actually do think that we are on the cusp of starting to see where machine learning might contribute in groundbreaking ways, which is of course why I am working in this area.

There are companies whose whole goal and premise of existence is the combination of high throughput machine learning and high throughput biology to really move the dial. Then, there are a whole bunch of places that use some machine learning on the side. Maybe they save some money on experiments or maybe they get to a better point than they would have otherwise. However, I do think we are starting to see a lot more sophistication in communication between the machine learning and biology spheres, including in industry. The next 5 to 10 years are going to be really interesting in terms of what happens and where it happens. I hope that it happens in protein engineering and small molecule design.

BSJ: How do you hope your research in particular will impact the future of drug design?

JL: There are many hybrid groups out there that are computationally focused, but very application driven. These groups make things happen and get results, but they are typically more consumers of machine learning methods. That is really valuable. It is sort of the equivalent of translational research in biology, right? You need those people there, making sure it works.

I sit in the electrical engineering computer science department in the AI group, which has some of the best AI students in the whole country. I have had some students who are really cross-disciplinary with very rigorous technical expertise find me, so my group is one of the very few that is trying to think things through very cleanly from first principles or more abstract concepts. People like our two most recent papers, for example, because we carefully painted a really clear picture of the problem. I think that is what is missing from a lot of computational biology. Sometimes when I give talks, I say, "You know what? I am not even going to show you our results from our paper. If you want to see them, you can see them. What I want to convince you of is how to think about this problem." That might sound silly, but I think that is actually really important because how you think about it dictates how you find specific solutions with particular collaborators.

When you are thinking in a coherent, fundamental way, you are more likely to arrive at an engineering solution that works. We are creating a more rigorous scaffolding on which other researchers can think about the specifics with respect to certain domains. We are also working very collaboratively with people on the translational side of things. Doing both foundation and application is beautiful because there can be a very nice interplay between them.

REFERENCES

1. *Headshot*: Jennifer Listgarten [Photograph]. Retrieved from http://www.jennifer.listgarten.com
2. Schneider, P., Walters, W. P., Plowright, A. T., Sieroka, N., Listgarten, J., Goodnow, R. A., Fisher, J., Jansen, J. M., Duca, J. S., Rush, T. S., Zentgraf, M., Hill, J. E., Krutoholow, E., Kohler, M., Blaney, J., Funatsu, K., Luebkemann, C., & Schneider, G. (2020). Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, *19*(5), 353–364. https://doi.org/10.1038/s41573-019-0050-3
3. Listgarten, J., Weinstein, M., Kleinstiver, B. P., Sousa, A. A., Joung, J. K., Crawford, J., Gao, K., Hoang, L., Elibol, M., Doench, J. G., & Fusi, N. (2018). Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. *Nature Biomedical Engineering*, *2*(1), 38–47. https://doi.org/10.1038/s41551-017-0178-6
4. Brookes, D., Busia, A., Fannjiang, C., Murphy, K., & Listgarten, J. (2020). A view of estimation of distribution algorithms through the lens of expectation-maximization. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 189–190. https://doi.org/10.1145/3377929.3389938