How machine learning is influencing protein engineering

Jennifer Listgarten

Berkeley EECS
ELECTRICAL ENGINEERING & COMPUTER SCIENCES

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

qb3
ucb·ucsc·ucsf

CCB | CENTER FOR COMPUTATIONAL BIOLOGY

Graduate Program in Bioengineering
UCSF
UC Berkeley

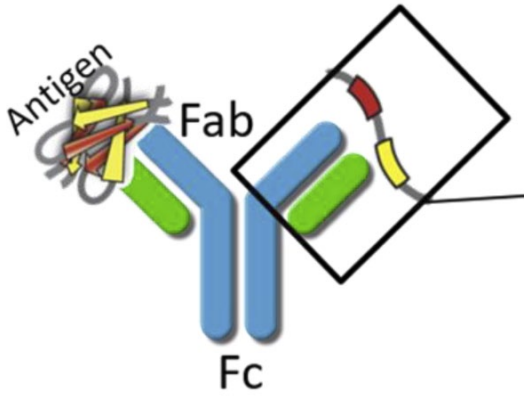
Talk outline

1. Intro: protein engineering + ML
2. ML-based design challenges
3. Conditioning for design

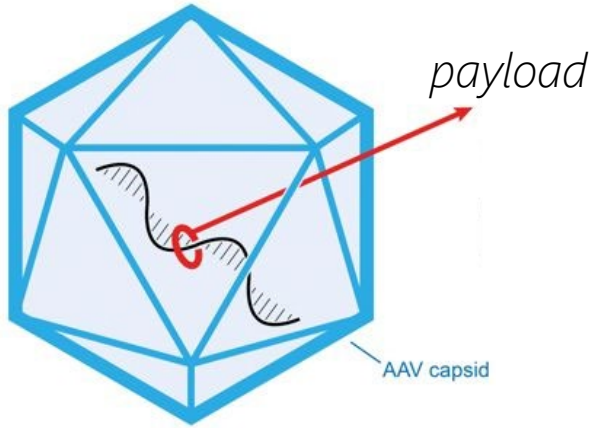
Talk outline

1. Intro: protein engineering + ML
2. ML-based design challenges
3. Conditioning for design

Protein engineering: therapeutics, environment, *etc.*

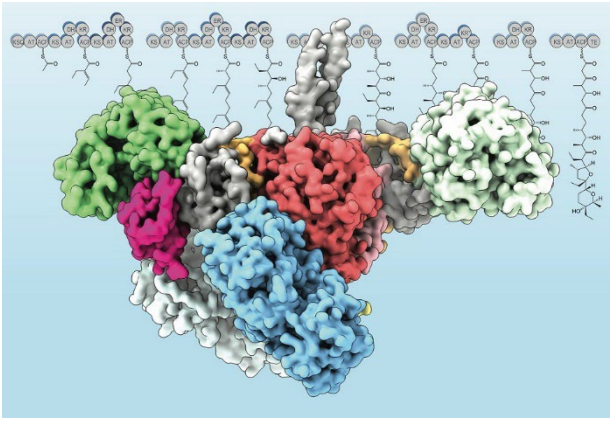


antibody therapeutics

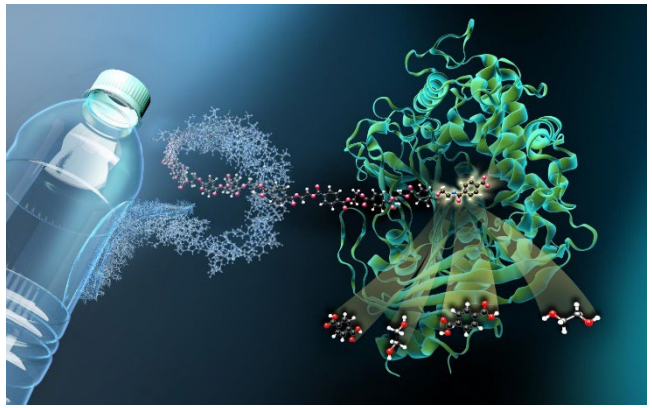


gene therapy virus delivery (AAV)

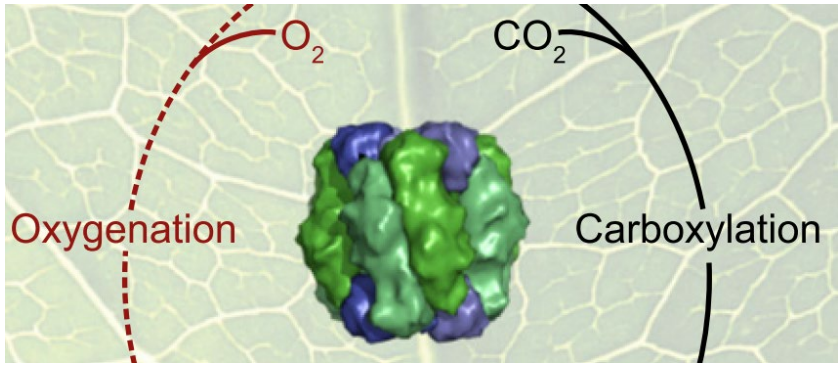
gene editing (CRISPR/Cas9)



antibiotics & biofuel production (PKS)



plastic recycling (PETase)



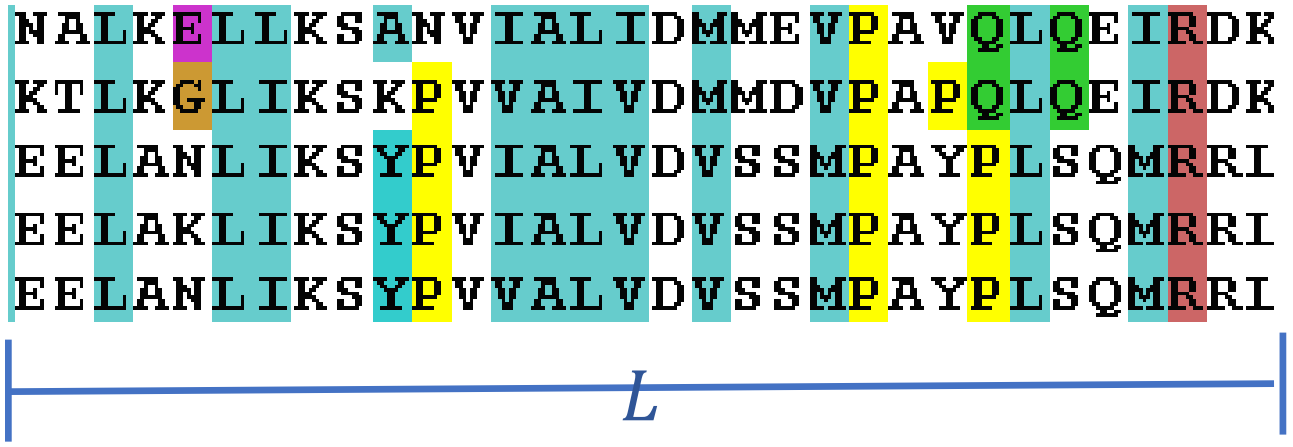
CO_2 biosequestration (RuBisCO)

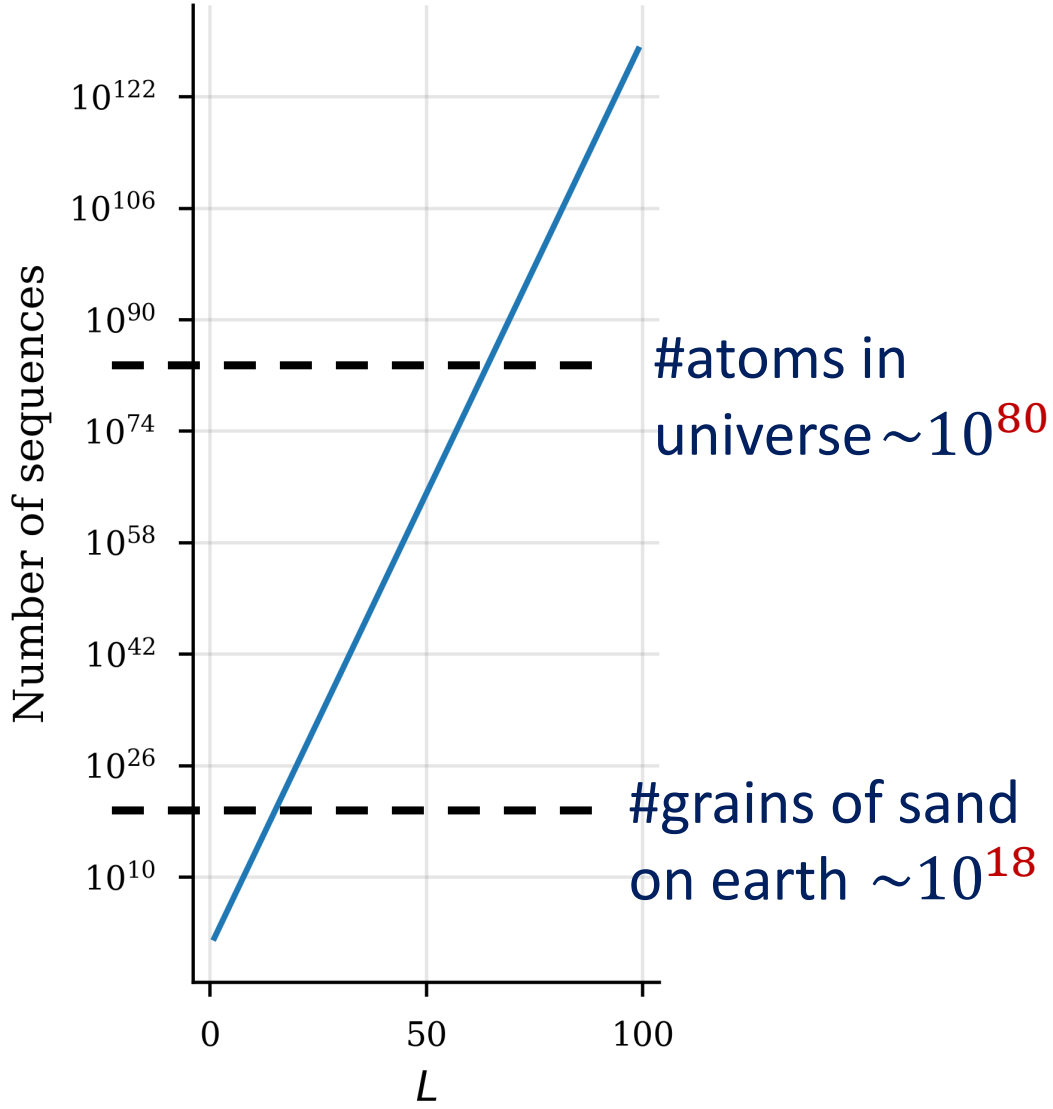
Fundamental difficulty: design space is nearly infinite

- Also highly rugged design space

$\Longrightarrow$ size scales as $\sim 20^L$

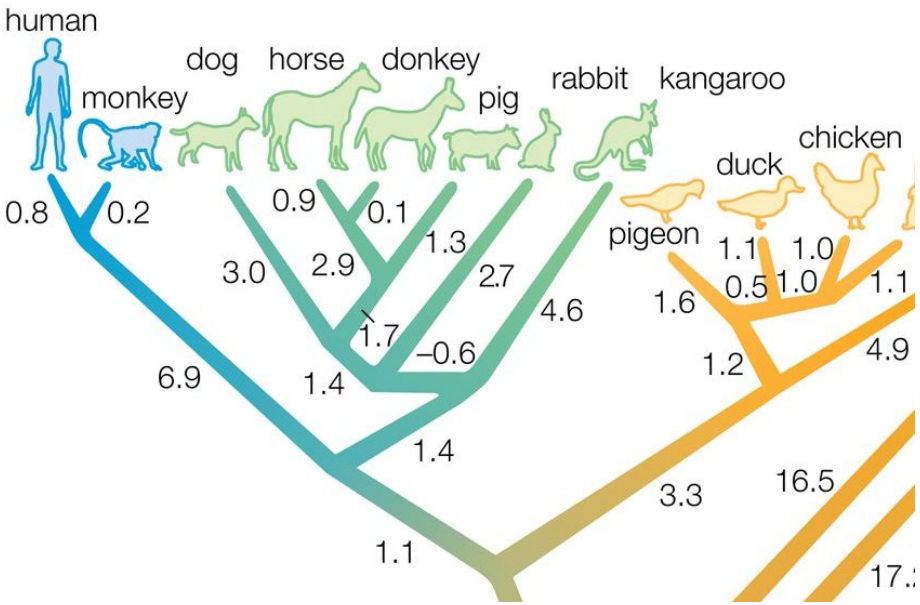
- Discrete search space (no gradients)



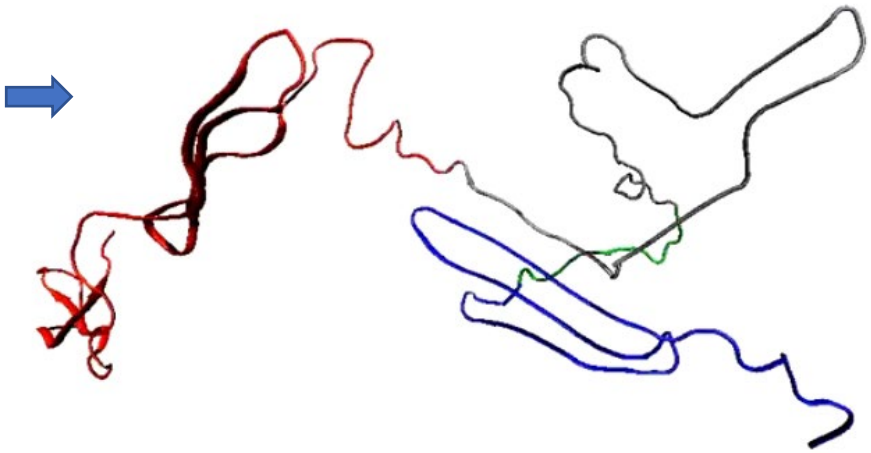


Successes in navigating this complex space

1. Nature: via evolution *over millions of years*.



MSKGEELFTGVVPILV
ELDGDVNGHKFSVSG
EGEGDATYGKLTLKFIC
TTGKLPVPWPTLVTTF
SYGVQCFSRYPDHMK
QHDFFKSAMPEGYVQ
ERTIFFKDDGNYKTRA
EVKFEGDTLVRIELKGI
DFKEDGNILGHKLEYN
YNSHNVYIMADKQKN
GIKVNFKIRHNIEDGSV
QLADYQQNTPIGDGPV
LLPDNHYLSTQSALSK
DPNEKRDHMVLLEFVT
AAGITHGMDELYK



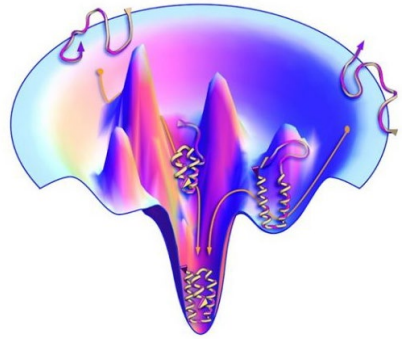
green fluorescent protein folding itself

Successes in navigating this complex space

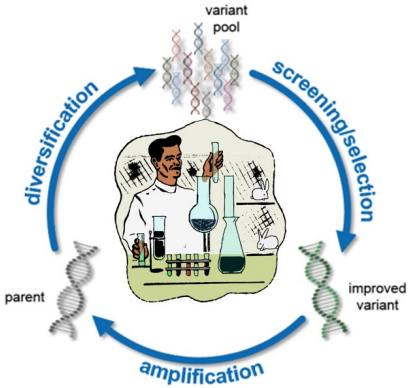
1. Nature: via evolution *over millions of years*.
2. Various protein engineering strategies.

Protein engineering strategies emerging

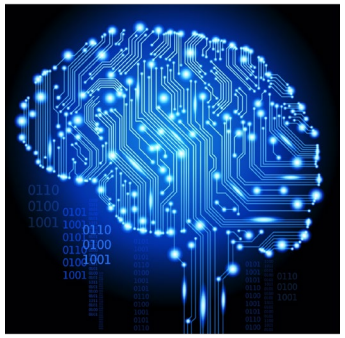
i. Computation ("data free"): **physics-based energy functions** (*e.g.*, Rosetta) to model **protein structure**, and protein binding.
~1997-2023'ish (almost R.I.P.) [2024 Nobel Prize]



ii. Wetlab: **directed evolution** to iteratively directly design property of interest.
~1993-present [2018 Nobel Prize]



iii. Machine learning (augmented): generative models; function prediction; structure prediction, etc. *~2018(?)-present*



Did AlphaFold2/3 "solve" protein engineering?

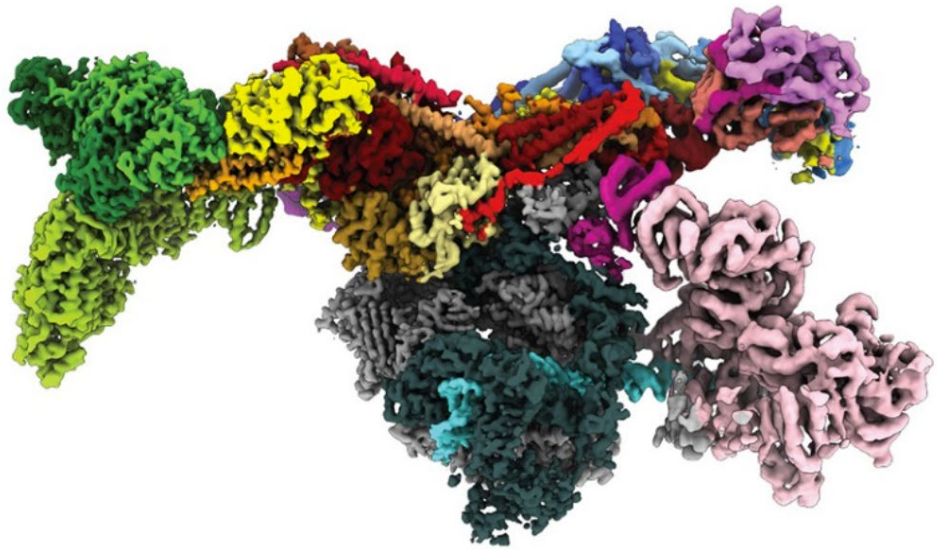
NEWS | 22 July 2021

DeepMind's AI predicts structures for a vast trove of proteins

AlphaFold neural network produced a 'totally transformative' database of more than 350,000 structures from *Homo sapiens* and 20 model organisms.

Ewen Callaway

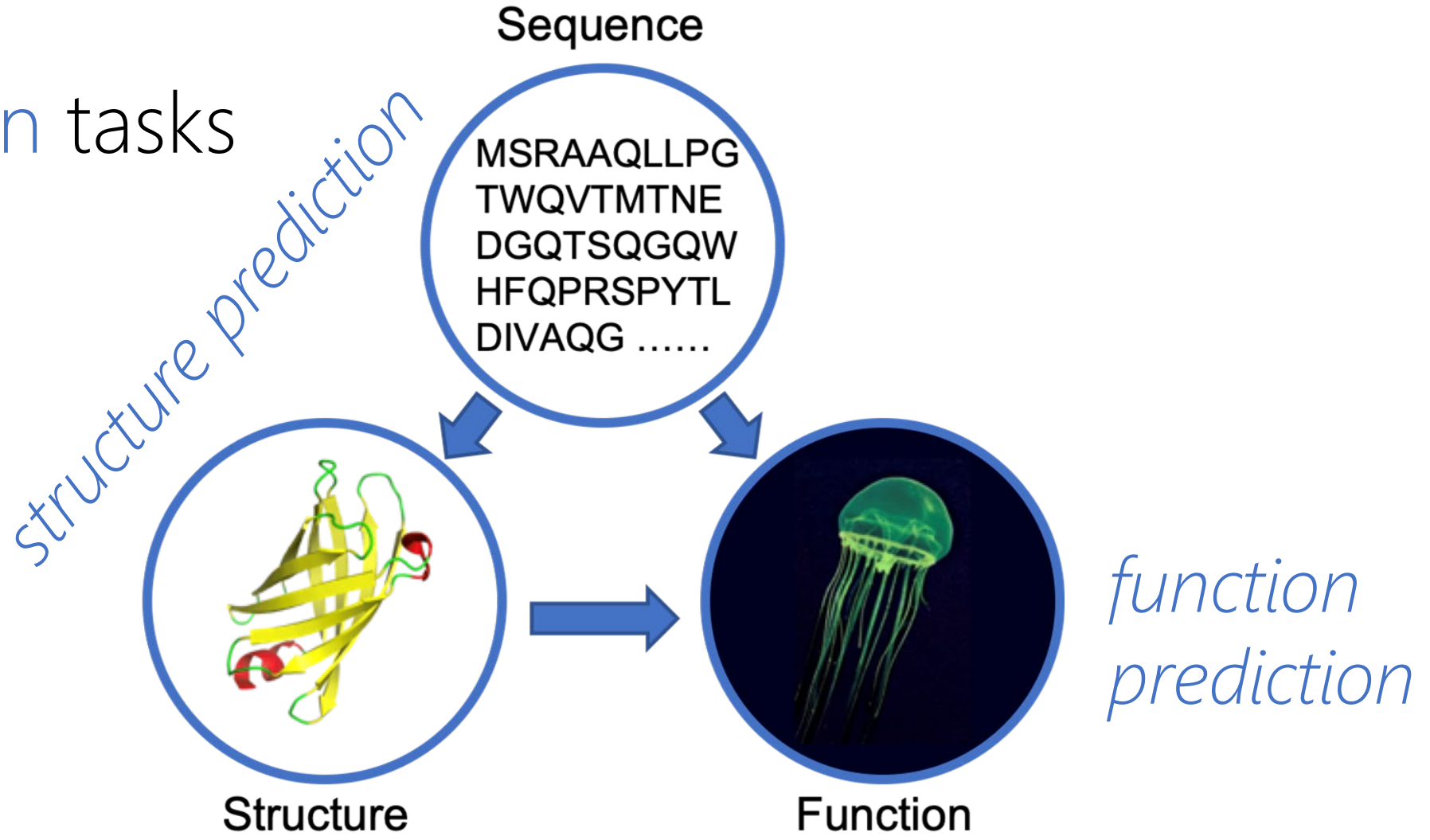
sequence→ structure



- No: don't typically know which protein structures we need.
- If did, would need: *structure→sequence*. (decent ML solutions exist).
- Bottleneck challenge: predict which proteinsjj have the function we desire—often extrapolatively.
- AlphaFold2 *was* a breakthrough, and is already useful.

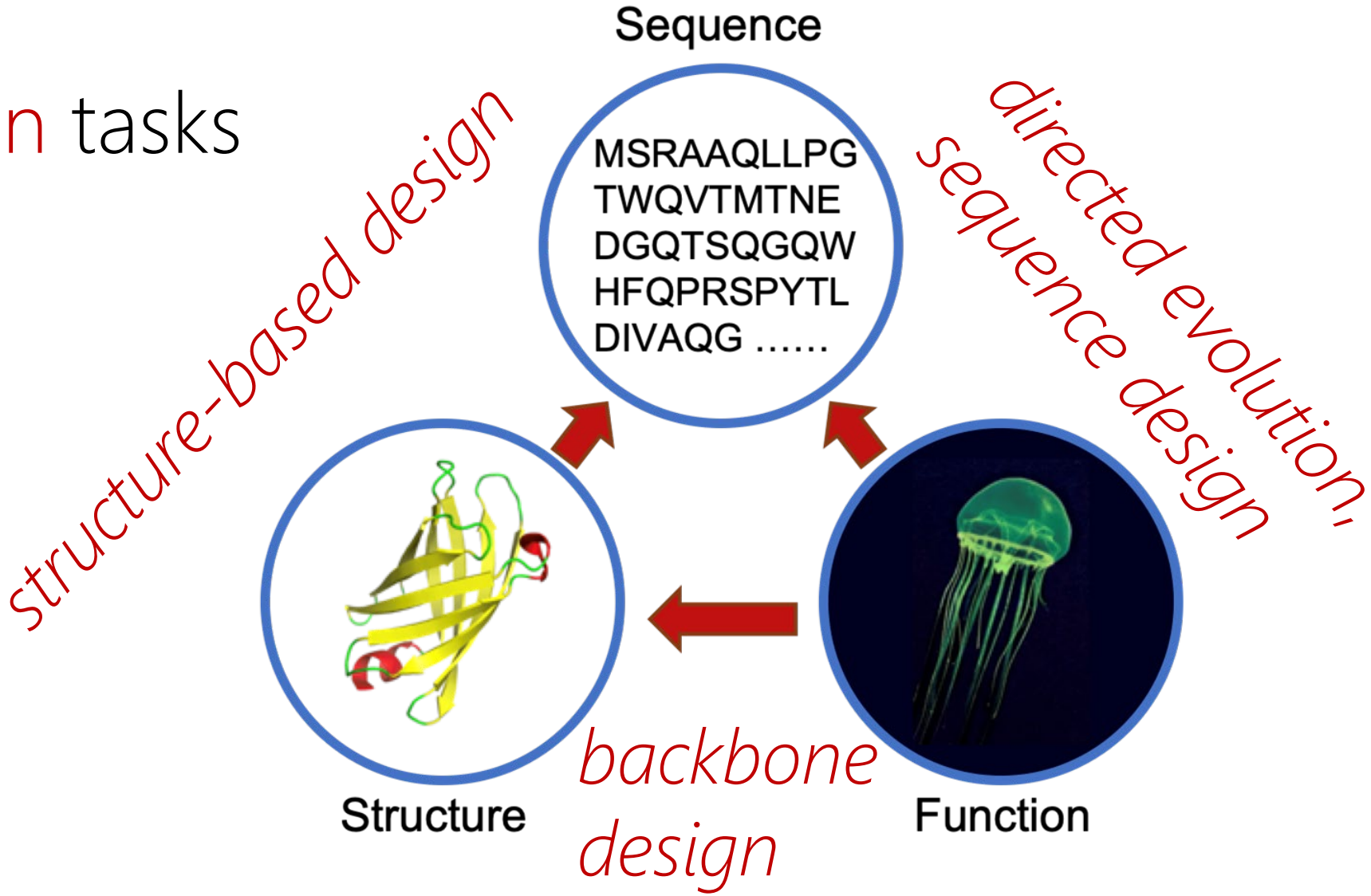
A suite of ML protein engineering problems

Prediction tasks



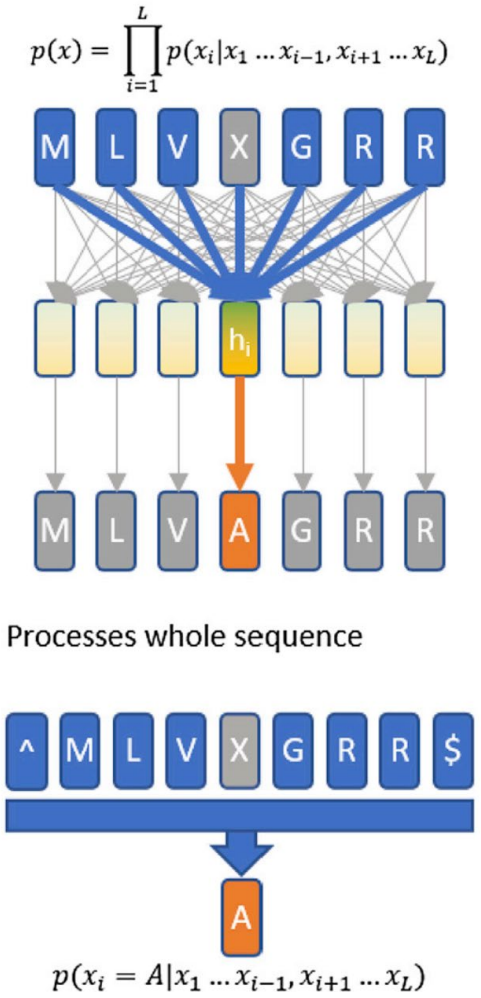
A suite of ML protein engineering problems

Design tasks



Some trends in ML + protein engineering

1. Representation learning: *un(self)supervised learning on* large-scale databases (millions of natural proteins, with *e.g.*, Transformers), or families.



- This is (approx.) *density estimation,* $p_\theta(sequence)$ through a bottleneck.

[Bepler *et al.*, *Cell Systems* 2021]

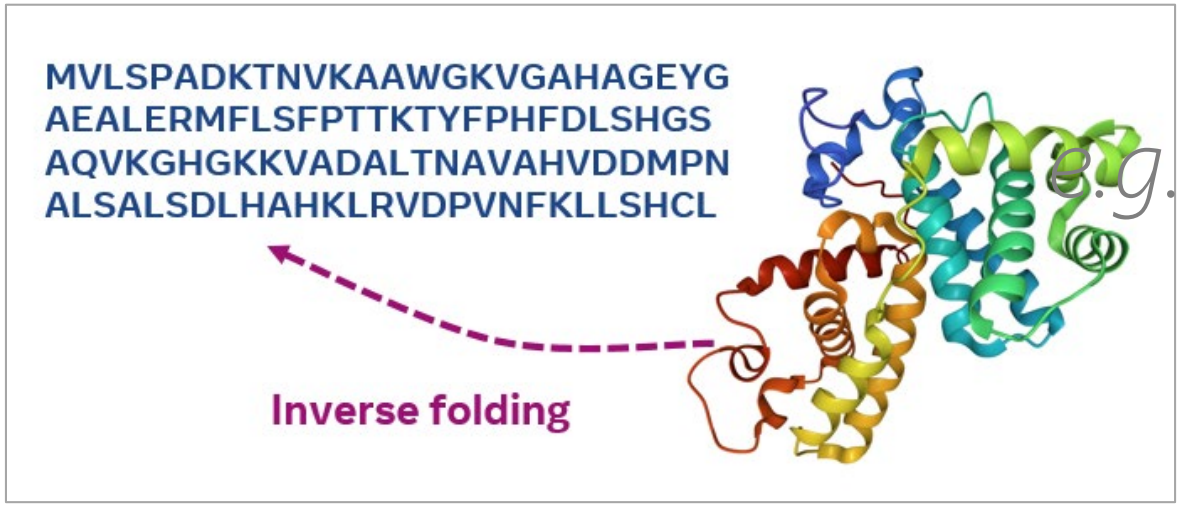
Some trends in ML + protein engineering

2. (Conditional) generative models for sequences.

This is (conditional) density estimation, $\boldsymbol{p_\theta(\text{sequence}|\mathbf{C})}$, (*e.g.* auto-regressive Transformer, Potts/VAE).

a) structure-conditioned,
   aka "inverse folding"

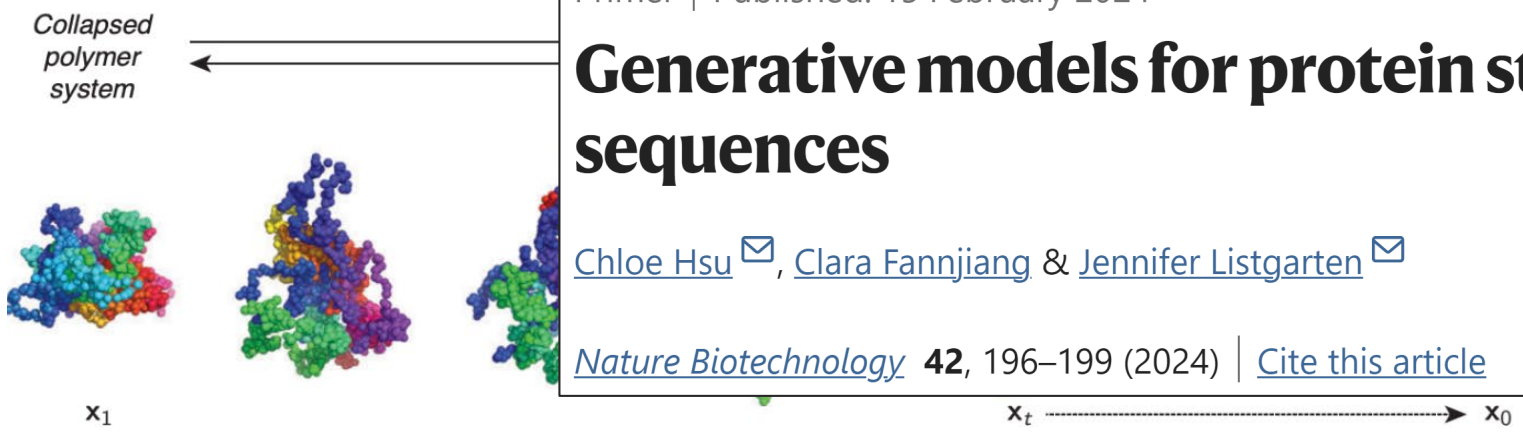
b) "control tag" conditioned,
   protein family



Some trends in ML + protein engineering

3. (Conditional) generative models for structure.

- This is (conditional) density estimation, $p_\theta(\mathbf{backbone}|\mathbf{F})$, (*e.g.* "Diffusion" models latest trend).
- Only as good as function prediction, $p(F|backbone)$.
- Paired with inverse-folding to get sequence



[Ingraham *et al. Nature* 20223

Primer | Published: 15 February 2024

Generative models for protein structures and sequences

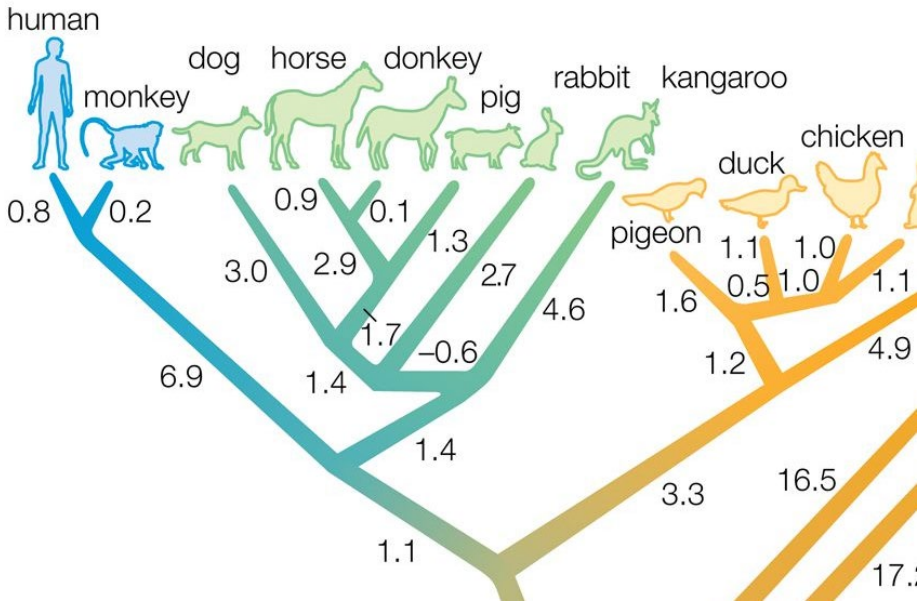
Chloe Hsu, Clara Fannjiang & Jennifer Listgarten

Nature Biotechnology **42**, 196–199 (2024) | Cite this article

Some trends in ML + protein engineering

4. ML to estimate function from sequence and/or function:

- *e.g.*, $p_\theta(F|sequence)$.
- Few or no labelled data.
- *Leverage evolutionary information**, or large unsupervised models on pan-proteomic database.

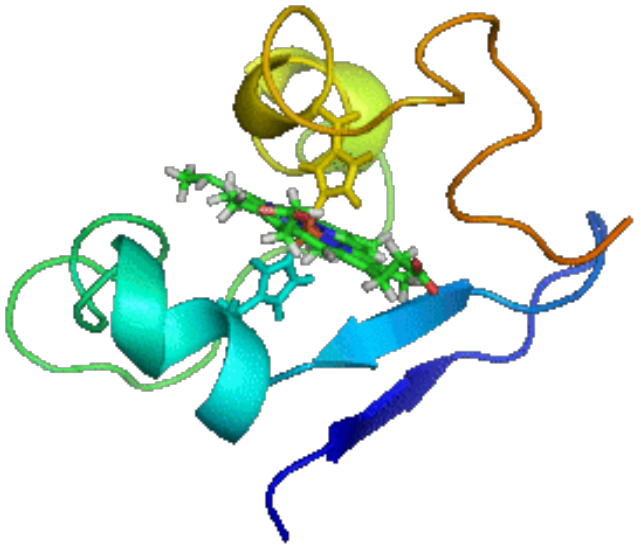


*key part of AlphaFold2/3

Some trends in ML + protein engineering

5. Structure prediction: filling the gaps left by AlphaFold2

- Orphan proteins (with *no/few homologs*).
- Protein-protein/DNA/RNA/small molecule binding.
- Protein dynamics and conformational distributions.



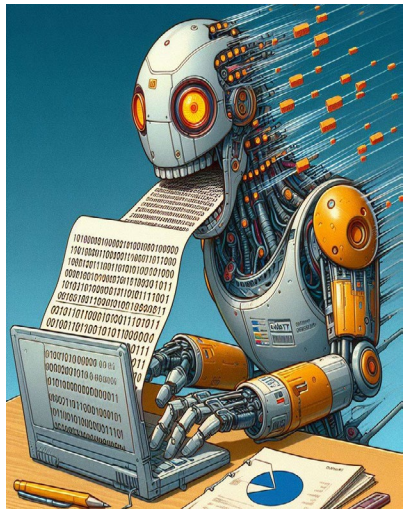
Unpacking some of the hype in AI+Science

Correspondence | Published: 25 January 2024

The perpetual motion machine of AI-generated data and the distraction of ChatGPT as a 'scientist'

Jennifer Listgarten

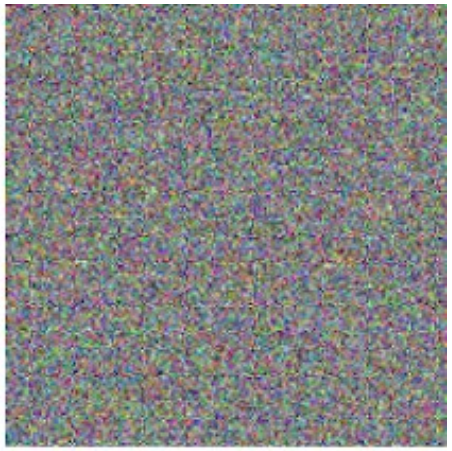
Nature Biotechnology (2024) | Cite this article

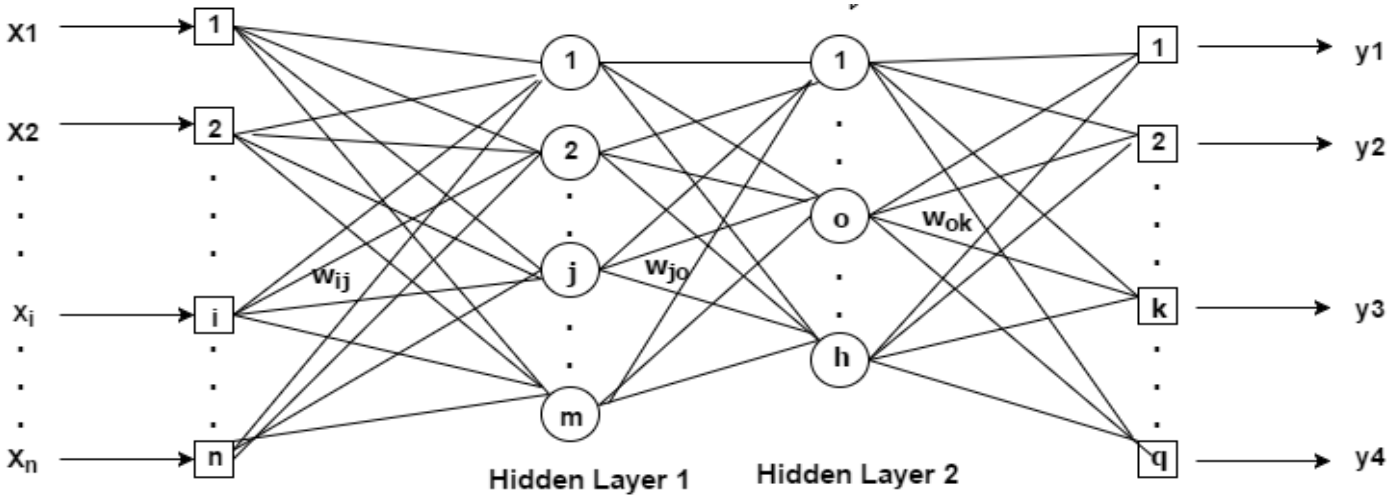


Talk outline

1. Intro: protein engineering + ML
2. ML-based design challenges
3. Conditioning for design

Analogy: can we trust “banana” design?

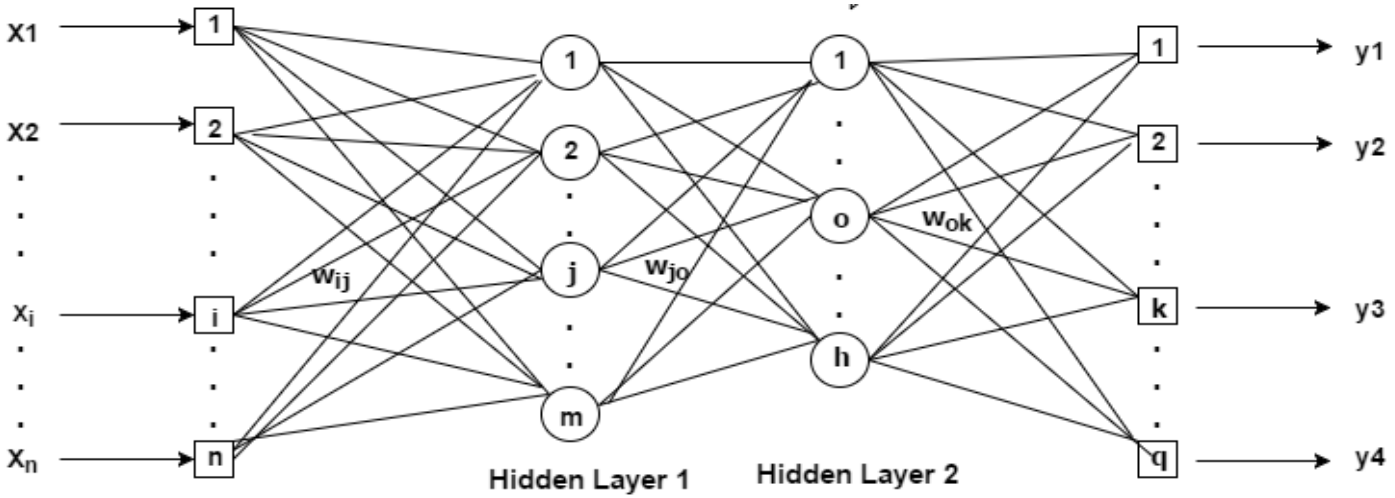




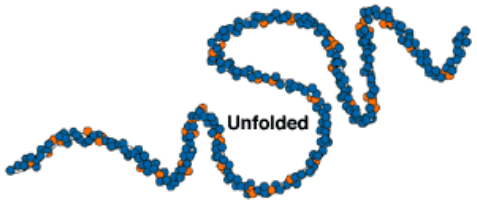
desired property

catalytic efficiency ↑

Naïve design yields abstract art ("pathology-finding").



desired property



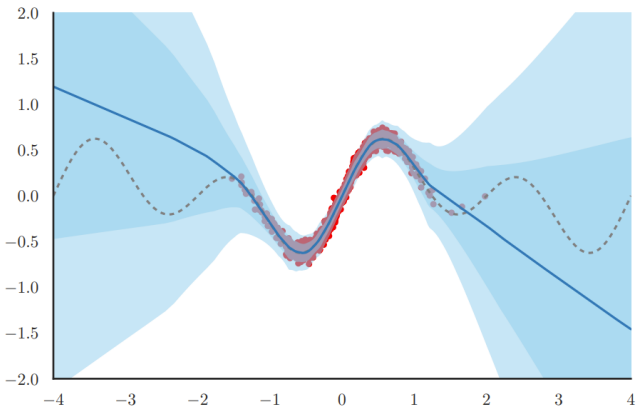
non-folding protein

catalytic efficiency ↑

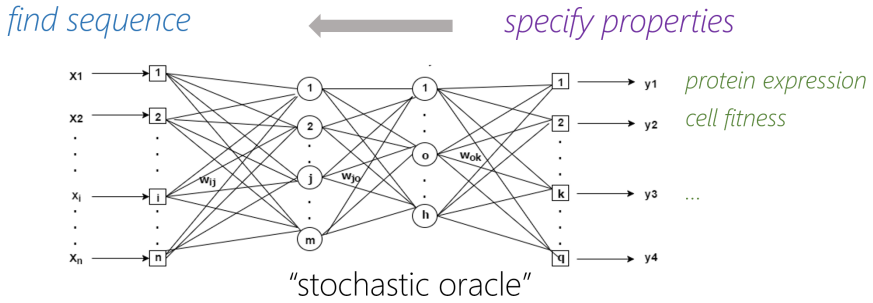
1. Brookes *et al ICLM* 2019 (CbAS)
2. Fannjiang *et al NeurIPS* 2020 (autofocus)

ML-based design challenges tackled in our group

1. A natural tension between leveraging the trained model for extrapolation, vs knowing that the model is not trustworthy in many areas of protein space (related to causality) [1,2].

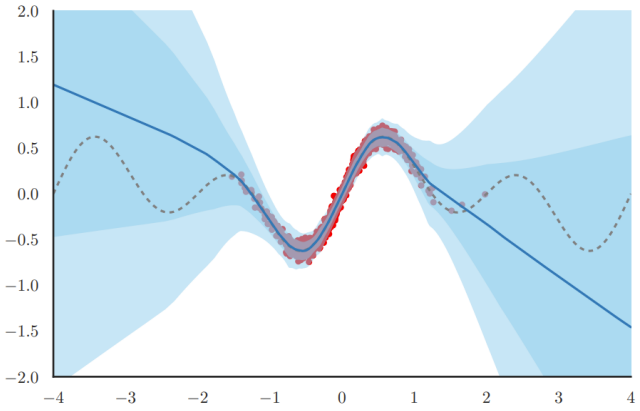


1. Brookes *et al* ICLM 2019 (CbAS)
2. Fannjiang *et al* NeurIPS 2020 (autofocus)

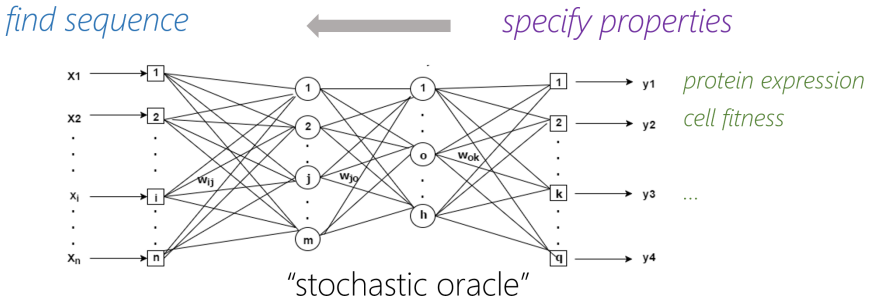


ML-based design challenges tackled in our group

1. A natural tension between leveraging the trained model for extrapolation, vs knowing that the model is not trustworthy in many areas of protein space (related to causality) [1,2].
2. Also related to estimation of *epistemic* uncertainty (whereas we typically think mostly of *aleotoric*) uncertainty [3, 4].

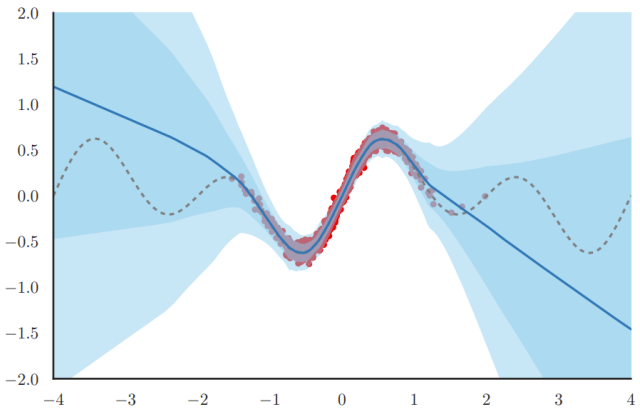


1. Brookes *et al* ICLM 2019 (CbAS)
2. Fannjiang *et al* NeurIPS 2020 (autofocus)
3. Nisinoff *et al* ACS Synth Bio 2023 (fv-BNN)
4. Fannjiang *et al* PNAS 2023(conformal)

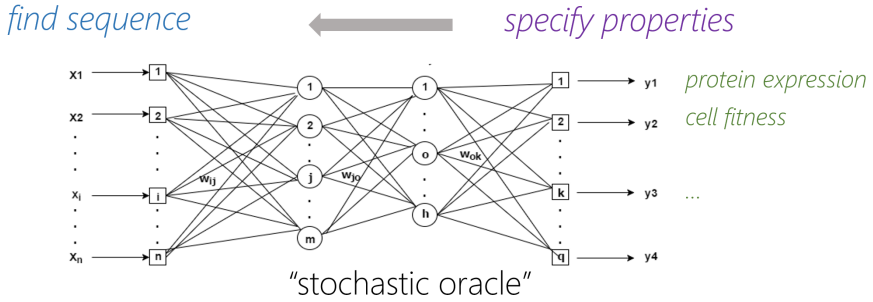


ML-based design challenges tackled in our group

1. A natural tension between leveraging the trained model for extrapolation, vs knowing that the model is not trustworthy in many areas of protein space (related to causality) [1,2].
2. Also related to estimation of *epistemic* uncertainty (whereas we typically think mostly of *aleotoric*) uncertainty [3,4].
3. Suitable protein inductive biases when using neural networks [3,5,6,7].



1. Brookes *et al* ICLM 2019 (CbAS)
2. Fannjiang *et al* NeurIPS 2020 (autofocus)
3. Nisinoff *et al* ACS Synth Bio 2023 (fv-BNN)
4. Fannjiang *et al* PNAS 2023(conformal)
5. Aghazadeh *et al* Nat. Comm. 2021
6. Brookes *et al* PNAS 2022
7. Hsu *et al* Nat. Biotech. 2022



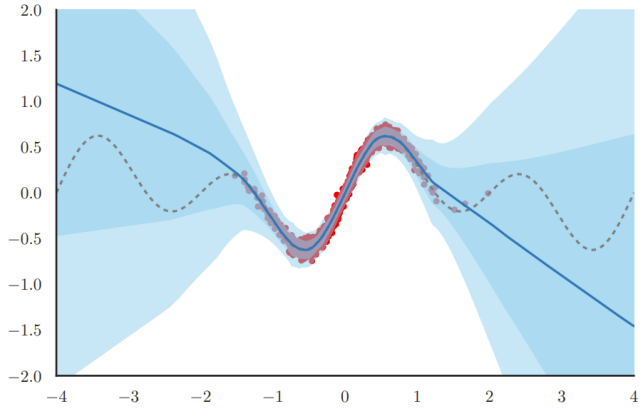
ML-based design challenges tackled in our group

1. A natural tension between leveraging the trained model for extrapolation, vs knowing that the model is not trustworthy in many areas of p
2. Also relate we typically th
3. Suitable p [3,5,6,7].
4. Design of distributions instead of individual sequences [1,2,8].

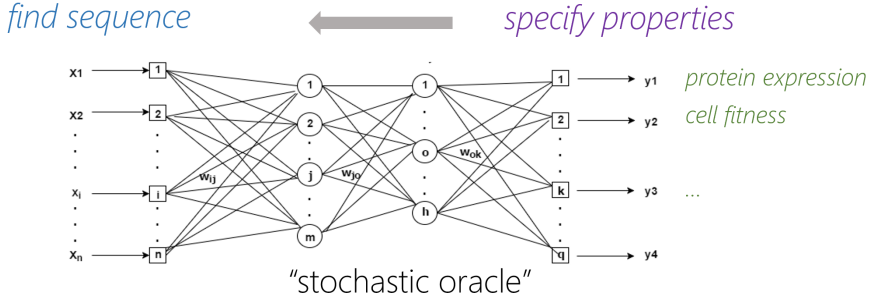
Is Novelty Predictable?

Clara Fannjiang and Jennifer Listgarten

Cold Spring Harb Perspect Biol ***2023***



1. Brookes *et al* ICLM 2019 (CbAS)
2. Fannjiang *et al* NeurIPS 2020 (autofocus)
3. Nisinoff *et al* ACS Synth Bio 2023 (fv-BNN)
4. Fannjiang *et al* PNAS 2023(conformal)
5. Aghazadeh *et al* Nat. Comm. 2021
6. Brookes *et al* PNAS 2022
7. Hsu *et al* Nat. Biotech. 2022
8. Zhu, Brookes *et al* Science Advances 2024



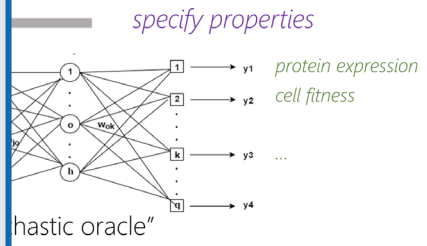
Conditioning by Adaptive Sampling for Robust Design

E θ)

m
x

Intimately related to Estimation of Distribution Algorithms (EDAs)

- Modern day "evolutionary" algorithms where "mutations", etc. replaced by generative model [Baluja & Caruana '95]
- ✓ CEM–rare event estimation [Rubinstein '99, '97]
- ✓ CMA-ES [Hansen *et al.* '03]
- ✓ Can be written as Expectation-Maximization [Brookes *et al.* 2019]
- ✓ Also more superficially to RL.



David Brookes

Brookes, Park & Listgarten *ICML* 2019

Conditioning by Adaptive Sampling for Robust Design

Two technical challenges:

1. $\boldsymbol{\theta}$ is in the expectation distribution.
2. MC estimates for rare events.

EM-like algorithm emerges

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \log \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \left[P(S|\mathbf{x}) \right],$$

$$\geq$$

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta}^{(t)})} \left[P(S|\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta}) \right]$$

Anneal and MC

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{M} P(S^{(t)}|\mathbf{x}_i^{(t)}) \log p(\mathbf{x}_i^{(t)}|\boldsymbol{\theta})$$

weights for MLE

Brookes, Park & Listgarten *ICML* 2019

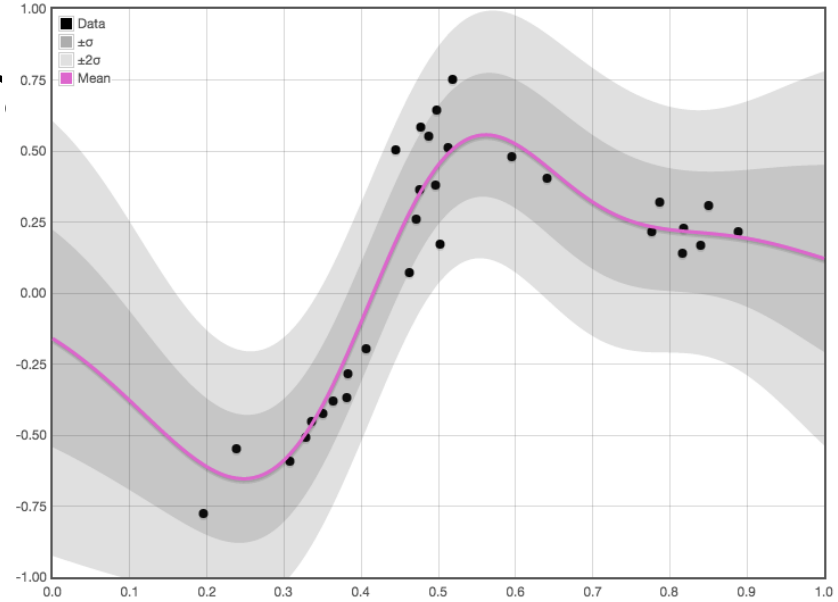
Conditioning by Adaptive Sampling for Robust Design

EM-like algorithm emerges

Assumes regression model is unbiased and has calibrated uncertainty estimates

Two

1. distribution.
2. MC estimates for r events.



$\geq$

$) \; [P(S|\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta})]$

Anneal and MC

$P(S^{(t)}|\mathbf{x}_i^{(t)}) \log p(\mathbf{x}_i^{(t)}|\boldsymbol{\theta})$

1

weights for MLE

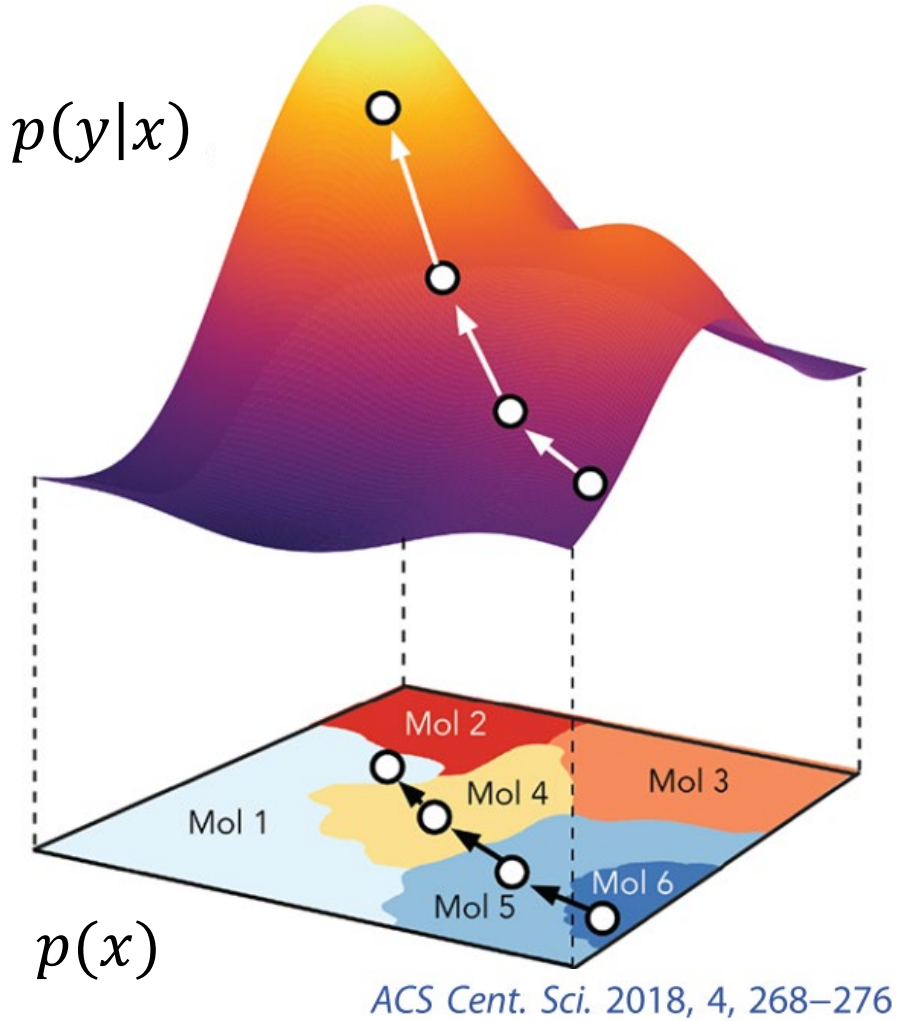
Brookes, Park & Listgarten ICML 2019

Conditioning by Adaptive Sampling for Robust Design (CbAS)

How to handle non-trustworthy predictive model in design problems

If have access data $\{x_i, y_i\}$ used to trair oracle, or prior "soft trust" information,

- then have prior knowledge about where $p(y|x)$ is likely to be accurate: near $\{x_i\}$, so estimate $p(x_i)$ from those data.

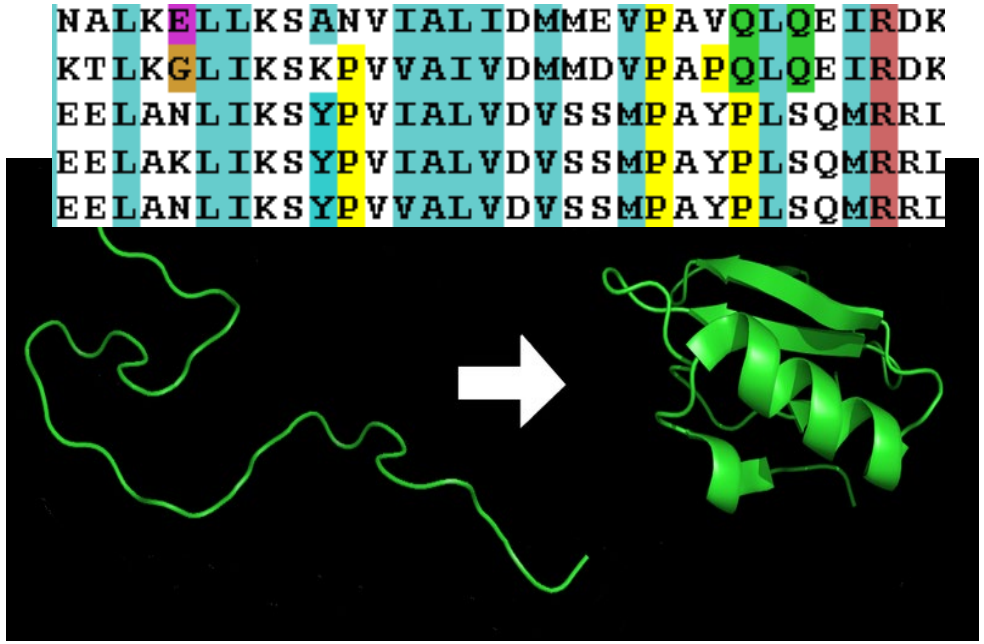


Brookes, Park & Listgarten ICML 2019

Conditioning by Adaptive Sampling for Robust Design (CbAS)

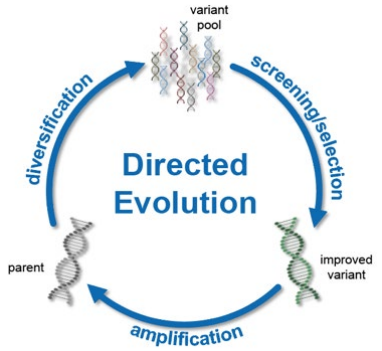
How to handle non-trustworthy predictive model in design problems

2. If don't have access to such data,
   - then leverage implicit domain knowledge, such as taking all proteins known to fold, to estimate $p(x_i)$.



Brookes, Park & Listgarten ICML 2019

Accounting for untrustworthy predictor



First approach (DbAS):

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \left[P(S|\mathbf{x}) \right],$$

$$\geq$$

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta}^{(t)})} \left[P(S|\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta}) \right]$$

Anneal and MC

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{M} P(S^{(t)}|\mathbf{x}_i^{(t)}) \log p(\mathbf{x}_i^{(t)}|\boldsymbol{\theta})$$

Updated approach (CbAS)

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} D_{KL} \left(p(\mathbf{x}|S, \boldsymbol{\theta}^{(0)}) || p(\mathbf{x}|\boldsymbol{\theta}) \right)$$

$$=$$

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta}^{(0)})} \left[P(S|\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta}) \right]$$

$$=$$

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta}^{(t)})} \left[\frac{p(\mathbf{x}|\boldsymbol{\theta}^{(0)})}{p(\mathbf{x}|\boldsymbol{\theta}^{(t)})} P(S|\mathbf{x}) \log p(\mathbf{x}|\boldsymbol{\theta}) \right]$$

Anneal and ...

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{M} \frac{p(\mathbf{x}_i^{(t)}|\boldsymbol{\theta}^{(0)})}{p(\mathbf{x}_i^{(t)}|\boldsymbol{\theta}^{(t)})} P(S^{(t)}|\mathbf{x}_i^{(t)}) \log p(\mathbf{x}_i^{(t)}|\boldsymbol{\theta})$$

Brookes, Park & Listgarten ICML 2019

Autofocused oracles for model-based design

- Previously, predictive model is fixed because we are not acquiring any new data.
- Should we consider changing the oracle as the optimization progresses, *even in a fixed data setting?*

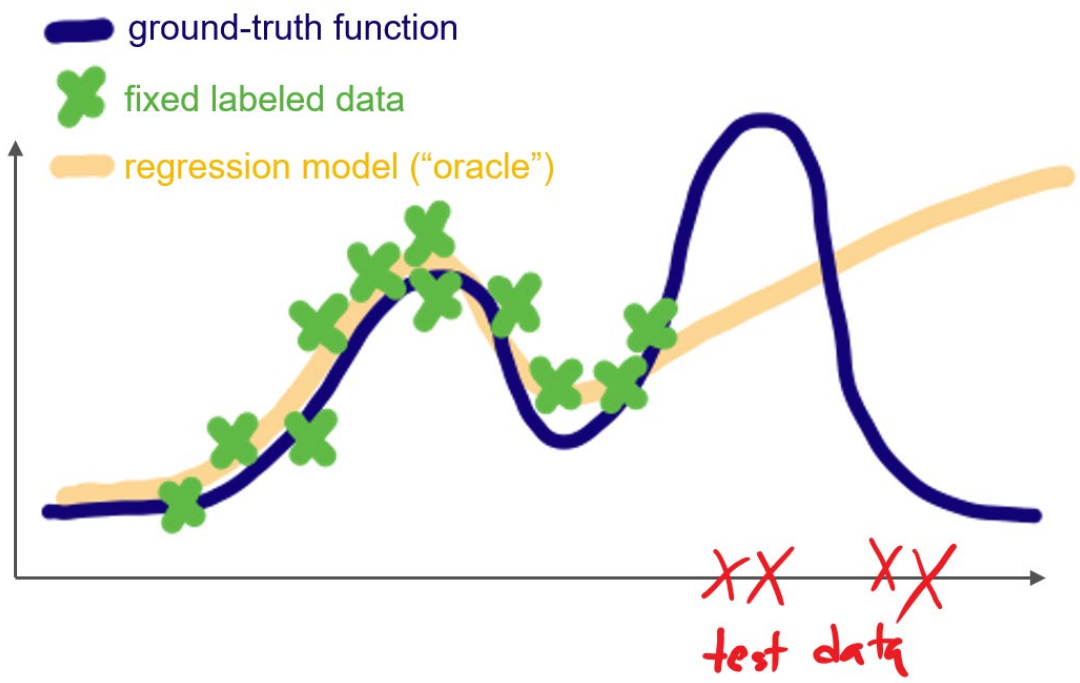
yes we should!

➢Related to accounting for domain shift (*e.g.*, IWERM).

Fannjiang & Listgarten *NeurIPS* 2020

Auto-focused oracles for model-based design

Show how updating the predictive model for function can help design, even when not collecting new data to train in.



ML-based design has "domain shift" as explore new regions of design space.

$$\beta^{(t)} = \arg\max_{\beta \in B} \frac{1}{n} \sum_{i=1}^{n} \frac{p_{\theta^{(t)}}(\mathbf{x}_i)}{p_0(\mathbf{x}_i)} \log p_\beta(y_i \mid \mathbf{x}_i).$$

Clara Fannjiang

Fannjiang & Listgarten, *NeurIPS* 2020

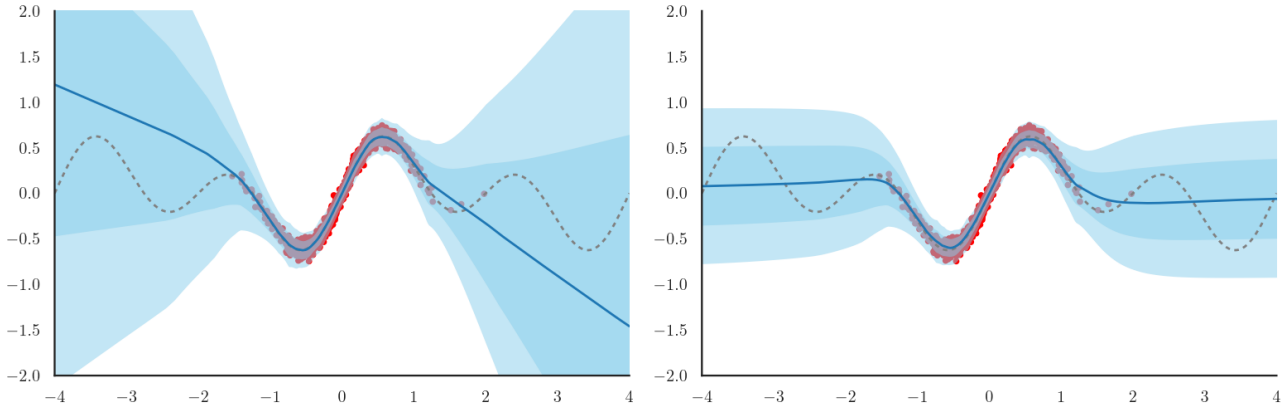
Augmenting Neural Networks with Priors on Functional Values

Coherent blending of function value prior information, such as biophysical models, to Bayesian Neural Networks (BNN).

Easy to implement, zero added cost.

$$\mu(\mathbf{x}) = \frac{\sigma^2_{\text{BNN}}(\mathbf{x})^{-1}\,\mu_{\text{BNN}}(\mathbf{x}) + \sigma^2_{\text{fv}}(\mathbf{x})^{-1}\mu_{\text{fv}}(\mathbf{x})}{\sigma^2_{\text{BNN}}(\mathbf{x})^{-1} + \sigma^2_{\text{fv}}(\mathbf{x})^{-1}},$$

$$\sigma^2(\mathbf{x}) = \left(\sigma^2_{\text{BNN}}(\mathbf{x})^{-1} + \sigma^2_{\text{fv}}(\mathbf{x})^{-1}\right)^{-1}.$$



regular BNN

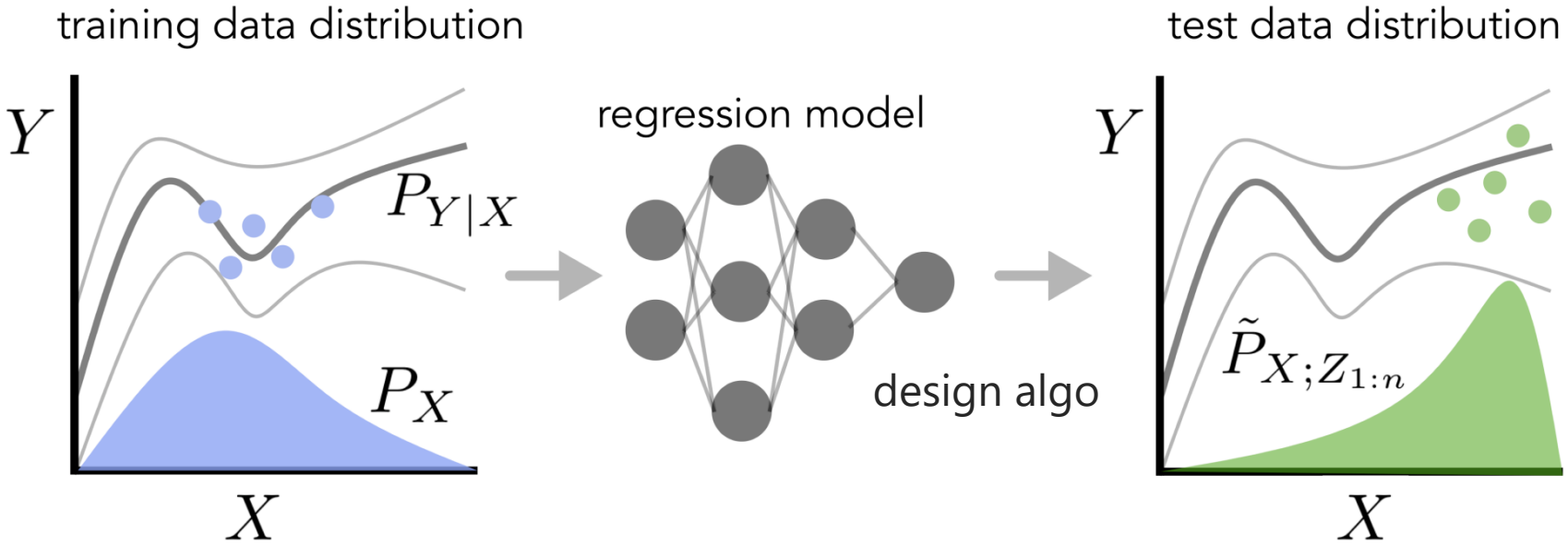
function-value augmented BNN

METHOD	LOG-LIKELIHOOD
NN	-8.33 ± 0.66
BNN	-5.73 ± 0.18
STACKING: BNN+NON-FUNCTIONAL PRIOR	-8.63 ± 0.33
STACKING: BNN+STABILITY PRIOR	-8.61 ± 0.34
fv-BNN (NON-FUNCTIONAL PRIOR)	-1.82 ± 0.00
fv-BNN (STABILITY PRIOR)	-1.53 ± 0.00

Hunter Nisonoff

Nisonoff, Wang, Listgarten, *ACS Synth Bio 2023*

Confidence sets for model-based design, with generalized *conformal prediction*



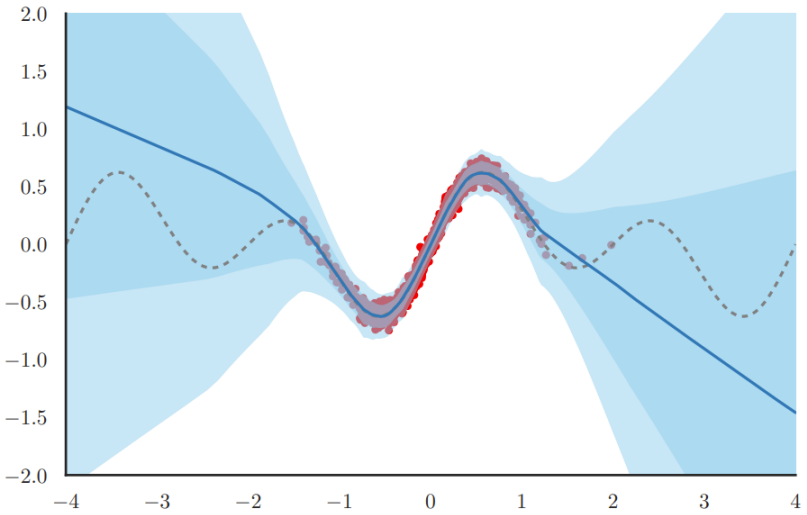
Design necessitates moving to regions of input space *far from training data*, where we trust the model's predictions the least.

[Conformal prediction for the design problem, Clara Fannjiang, *et al PNAS* 2022]

Standard *conformal prediction* gives finite sample guaranteed valid confidence sets (in expectation*).

- Under assumption of exchangeability of training and test data, obtain confidence sets on the labels,

$$\mathbb{P}\left(Y_{\text{test}} \in C(X_{\text{test}})\right) \geq 1 - \alpha$$



- Generalizations for different train and test distributions, but requires independence of train vs test
- Clara: generalize further to "design dependence" *(feedback covariate shift)*, allowing dependence.

Sketch of *conformal prediction* for design dependence

Intuition: include all candidate labels, *y*, such that (x_{test}, y), looks sufficiently similar to the **weighted** training data as quantified by a user-crafted score, S_i.

$$S_i(X_{\text{test}}, y) = |Y_i - \mu_y(X_i)|$$

μ_y : regression model trained on training + candidate test data points

scores of *n + 1* training + candidate test data points

score of candidate test data point

$$C_\alpha(X_{\text{test}}) = \left\{ y \in \mathbb{R} : S_{n+1}(X_{\text{test}}, y) \leq \text{QUANTILE}_{1-\alpha}\left(\sum_{i=1}^{n+1} w_i^y(X_{\text{test}})\, \delta_{S_i(X_{\text{test}}, y)} \right) \right\}$$

$$w_i^y(X_{\text{test}}) \propto v(X_i; Z_{-i} \cup \{(X_{\text{test}}, y)\}),\ i = 1, \ldots, n,$$

$$w_{n+1}^y(X_{\text{test}}) \propto v(X_{\text{test}}; Z_{1:n}),$$

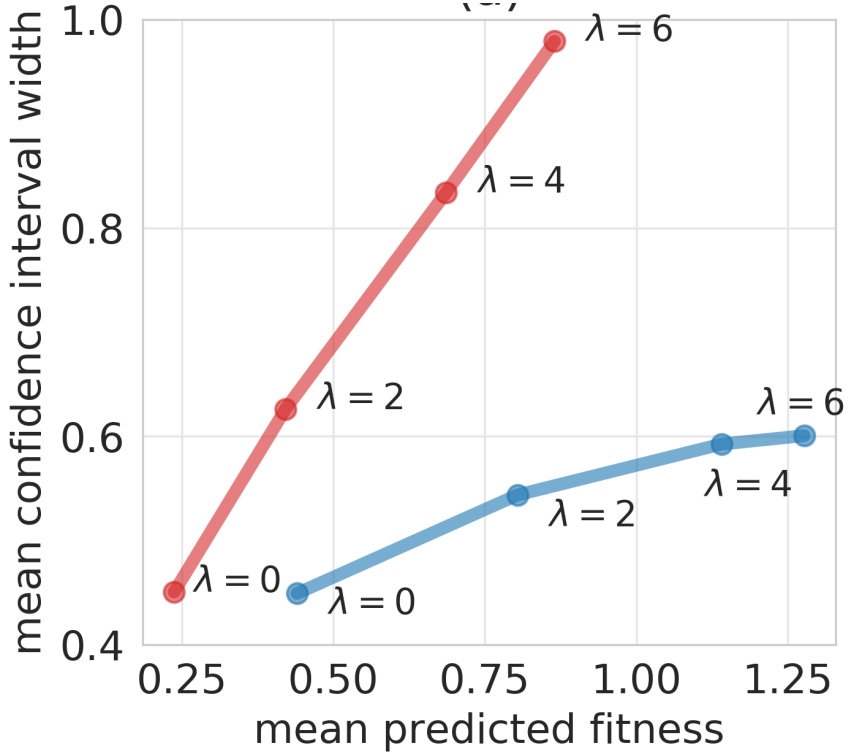
$$v(X; D) = \frac{\tilde{p}_{X;D}(X)}{p_X(X)}$$

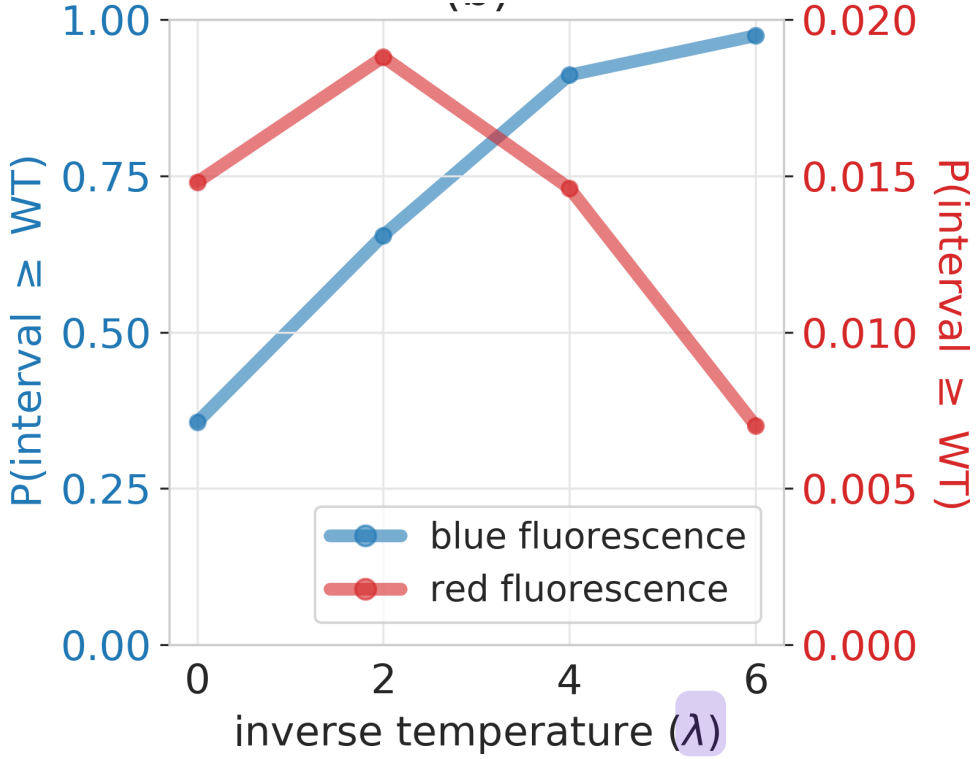
← distribution of designed inputs induced by training model on D

weights that take into account that the training and test data are
(i) **from different distributions** and
(ii) **statistically dependent**

Can guide hyperparameter choice (e.g. λ) of design algorithm

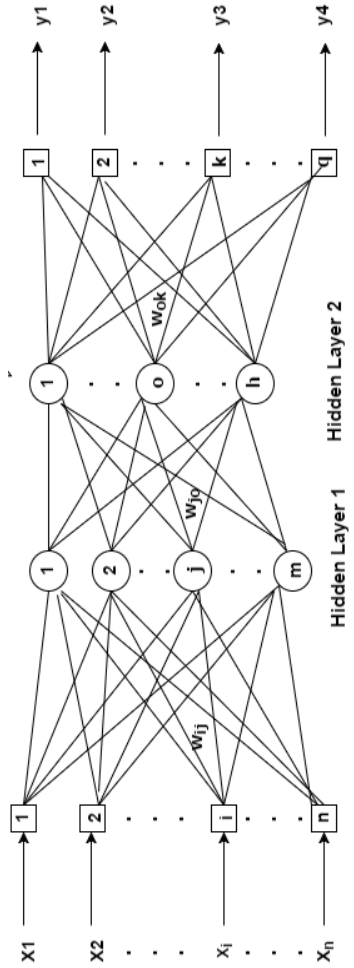
e.g., use confidence interval width to assess trade-off between entropy/diversity and expected predicted fitness

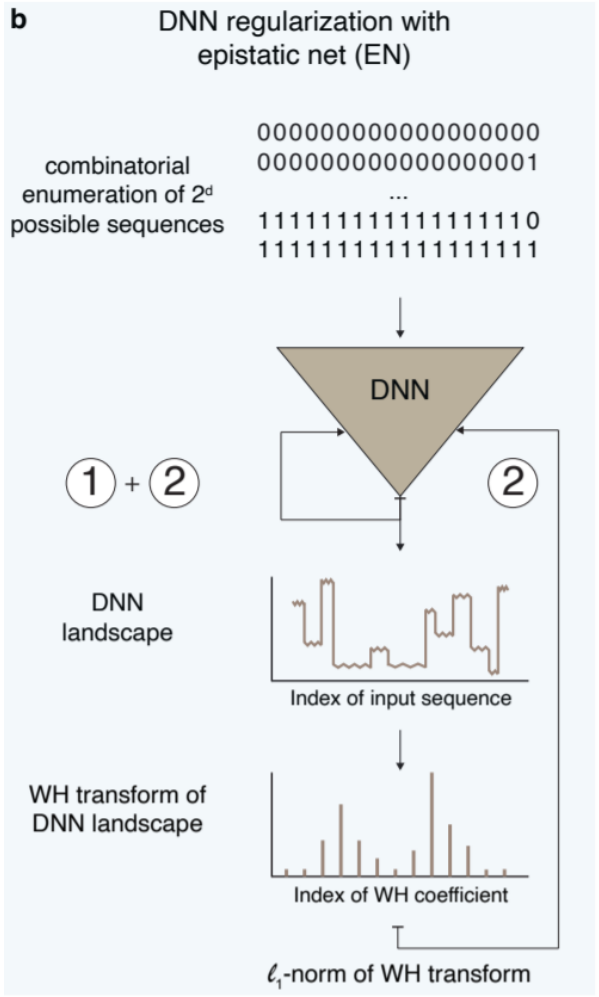




$$\tilde{p}_{X;Z_{1:n}}(X_{\text{test}}) \propto \exp(\lambda \cdot \mu_{Z_{1:n}}(X_{\text{test}}))$$

Sparse Epistatic Networks





- Inject suitable inductive biases for protein sequence functions.
- *i.e.* sparsity in "epistatic" terms (aka Walsh-Hadamard basis of features).

Aghazadeh*, Nisonoff* *et al*, Nat Comm 2020

Talk outline

1. Intro: protein engineering + ML
2. ML-based design challenges
3. Conditioning for design

How to condition for design?

- Suppose I have a generative model for protein sequences, $p(x)$.
- But I want to generate from a conditional generative model, $p(x|y)$, conditioned on structure.
- And I have access to either $\{(x_i, y_i)\}_i$ or a predictive model $p(y|x)$.

- $p(x)$ small molecule library (e.g. Enamine)
- $p(x|y)$, conditioned on desired chemical property (e.g, binding affinity)
- $p(y|x)$ predict binding affinity from molecule

How to condition for design?

Three ways to do this:

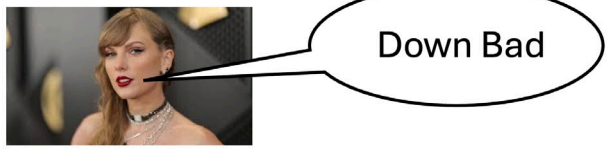
1. Start from scratch and directly train a conditional generative model.
2. Start with unconditional model, and "update it" using calls to the predictive model (e.g. CbAS [1-3], DPO [4]).
3. Freeze the unconditional model, and "guide" it at generation time (*e.g.*, diffusion models) [5-7].

1. Brookes, Park, Listgarten *ICLM* 2019
2. Fannjiang & Listgarten *NeurIPS* 2020
3. Brookes, Busia, *et al.* *GECCO* 2020
4. Rafailov *et al.* *NeurIPS* 2023
5. Sohl-Dickstein *et al.* *ICML* 2015
6. Dhariwal & Nichol *NeurIPS* 2021
7. Song *et al.* *ICLR* 2021

You are using Bayes rule!

For any modeling strategy, unless we bake in conditioning, we are using Bayes rule (even if we don't know it*).

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)}$$



*possibly approximately, such as in DPO, which could view as contrastive-based approximation to CbAS.

The beauty of classifier-guided diffusion

- Recall: diffusion/score models estimate $\nabla_x p_\theta(x)$.
- By pushing gradient through Bayes rule, we get rid of the normalizing constant

$$\begin{aligned}\nabla_{\mathbf{x}} \log p_t(\mathbf{x}|\mathbf{y}) &= \nabla_{\mathbf{x}} \log \frac{p_t(\mathbf{x}) p_t(\mathbf{y}|\mathbf{x})}{p_t(\mathbf{y})} \\ &= \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x}) - \nabla_{\mathbf{x}} \log p_t(\mathbf{y}) \\ &= \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x}).\end{aligned}$$

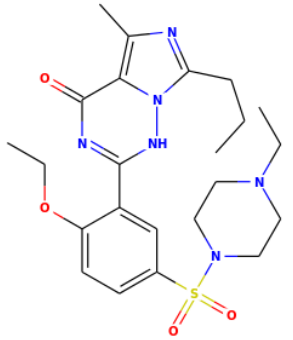
Unconditional model

Guidance for conditioning

J. Ingraham et al. Nature 623.7989 (2023): 1070-1078.

What about diffusion on discrete state spaces?

```
MYTWTGALITPCAAEESKLPINPLSNSLLRHH
YDTRCFDSTVTESDIRVEESIYQCCDLAPEEA
LTERLYIGGPLTNSKGQNCGYRRCRASGVLTT
SCGNTLTCYLKATAACRAAKLQDCTMLVNGDD
LVVICESAGTQEDAAALRAFTEAMTRYSAPPG
```



Challenge: for sequences, graphs, text, etc., $\nabla_x p_\theta(x)$ not useful.

Consequence I: Standard diffusion/score doesn't work.

Consequence II: Cannot use guidance, $\nabla_x p_\phi(y|x)$.

Some mitigating strategies:

a. Relax x into real-valued space and snap back.

b. Diffusion(flow) on multinomial space [Stark *et al. arXiv* 2024].

c. Continuous-time Markov processes [Campbell *et al.* 2022, 2024]

Unlocking Guidance for Discrete State-Space Diffusion and Flow Models

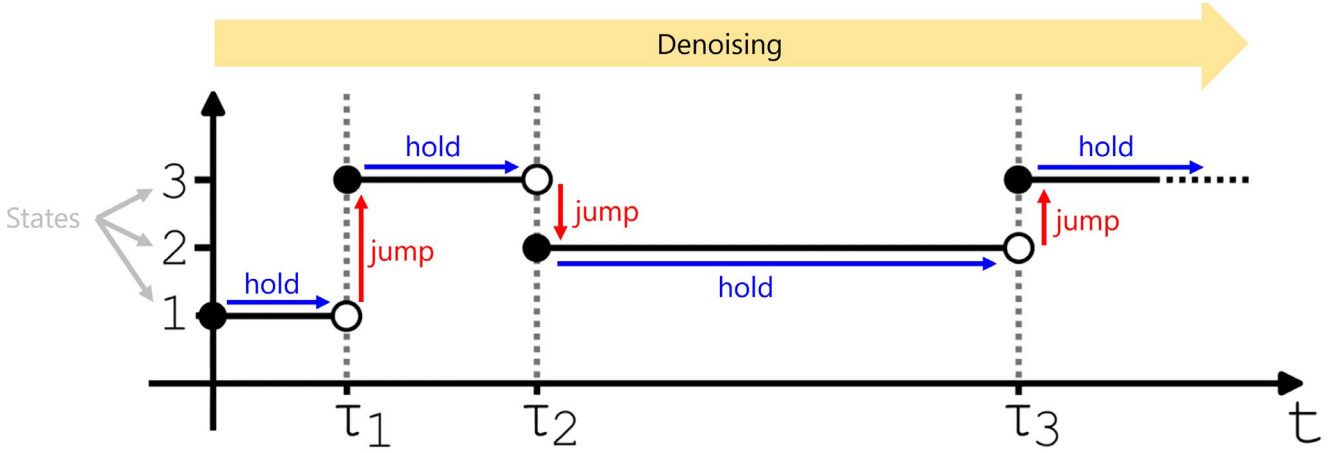
Hunter Nisonoff

Junhao (Bear) Xiong

Stephan Allenspach

CTMC enable not only diffusion, but also guidance, on both diffusion and flow models.

Nisonoff*, Xiong*, Allenspach*, Listgarten, *arXiv* 2024



$$p(x_{t+dt} = \tilde{x} | x_t = x) = \delta_{x,\tilde{x}} + R_t(x, \tilde{x})dt$$

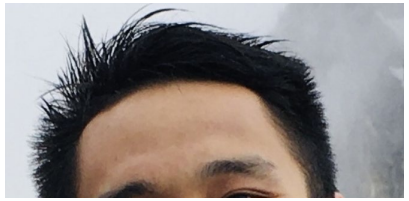
Campbell et al., NeurIPS (2022)

Campbell*, Yim* *et al.*, ICML (2024)

Unlocking Guidance for Discrete State-Space Diffusion and Flow Models

Hunter Nisonoff

Junhao (Bear) Xiong

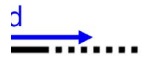


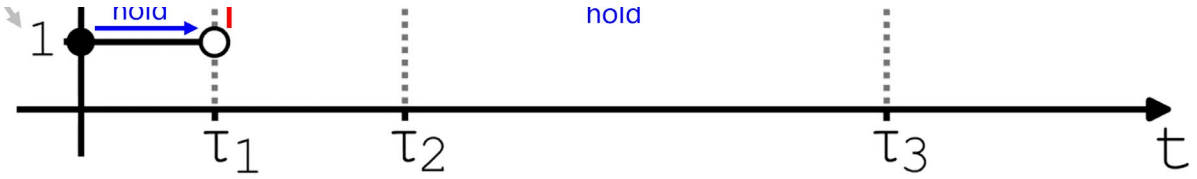
CTMC enable not only diffusion, but also guidance, on both diffusion and

Bayes' theorem:

$$p(\tilde{x}_{t+\Delta t}|x_t, y) = \frac{p(y|\tilde{x}_{t+\Delta t}, x_t)p(\tilde{x}_{t+\Delta t}|x_t)}{\sum_{x'_{t+\Delta t}} p(y|x'_{t+\Delta t}, x_t)p(x'_{t+\Delta t}|x_t)}$$

iv 2024





$$p(x_{t+dt} = \tilde{x}|x_t = x) = \delta_{x,\tilde{x}} + R_t(x, \tilde{x})dt$$

Campbell et al., NeurIPS (2022)

Campbell*, Yim* *et al.*, ICML (2024)

Stephan Allenspach

Unlocking Guidance for Discrete State-Space Diffusion and Flow Models

Continuous state-space

unconditional predictor

$$\nabla_{x_t} \log p^{(\gamma)}(x_t|y) = \nabla_{x_t} \log p(x_t) + \gamma \nabla_{x_t} \log p(y|x_t)$$

J. Sohl-Dickstein *et al.*, ICML (2015).
P. Dhariwal* and A. Nichol*, NeurIPS (2021).
Y. Song *et al.*, ICLR (2021).

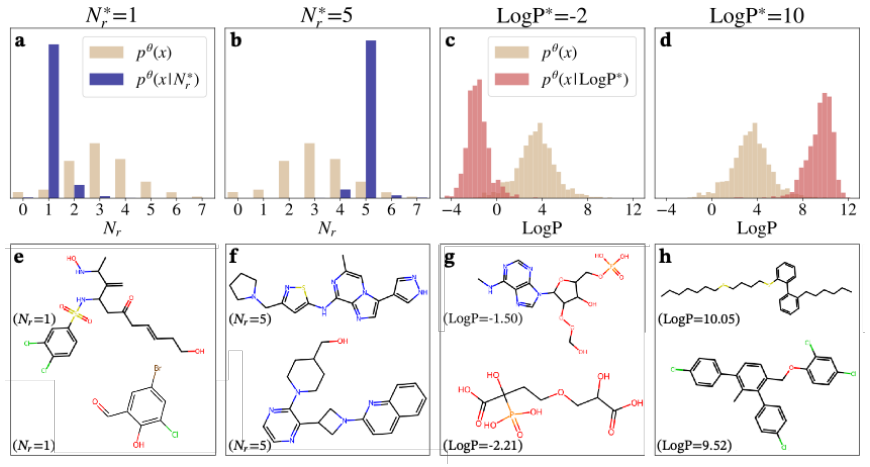
Discrete state-space

unconditional predictor

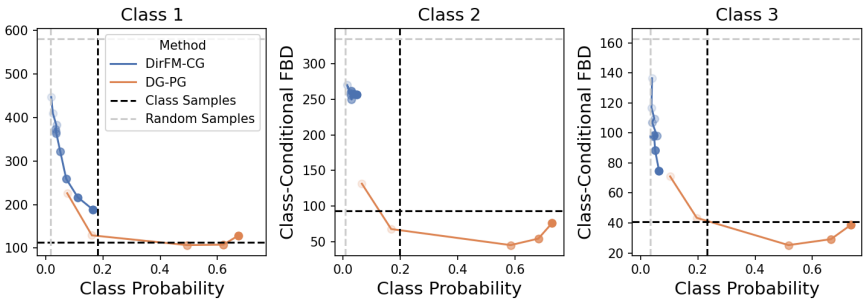
$$\log R_t^{(\gamma)}(x, \tilde{x}|y) = \log R_t(x, \tilde{x}) + \gamma \left(\log p(y|\tilde{x}, t) - \log p(y|x, t) \right)$$

Nisonoff*, Xiong*, Allenspach*, Listgarten, *arXiv* 2024

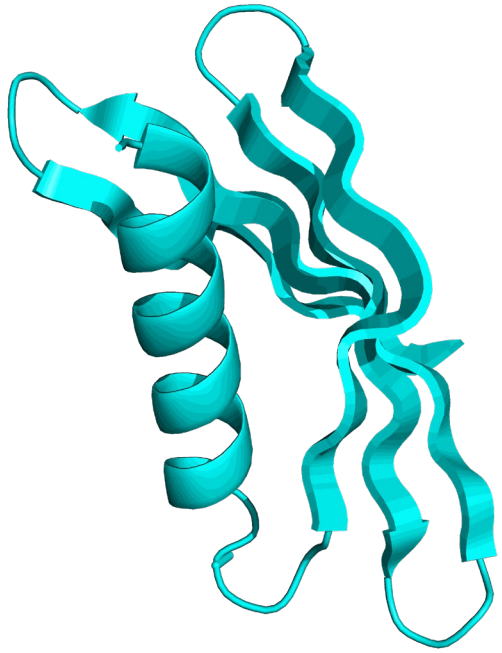
Application in the sciences



Small molecules



DNA sequences (enhancers)



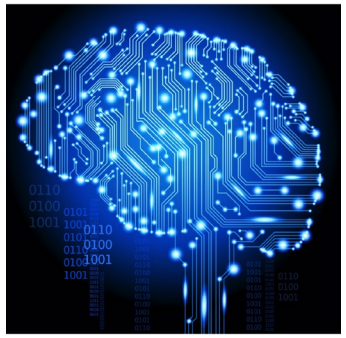
Amino acids (proteins)

stability-guided inverse folding

Nisonoff*, Xiong*, Allenspach*, Listgarten, *arXiv* 2024

The real deal: testing+developing our ideas with wetlab collaborators

- David Schaffer (UC Berkeley; AAV for gene therapy)
- David Savage (UC Berkeley; CRISPR-Cas9 system)
- Phil Romero (U Wisconsin; enzymes for plastic degradation)
- Secure and Robust Biosystems Design Group (LL National Labs, Columbia University, University of Maryland, University of Minnesota)
- Andrew Yang (UCSF; blood-brain barrier permeable proteins)



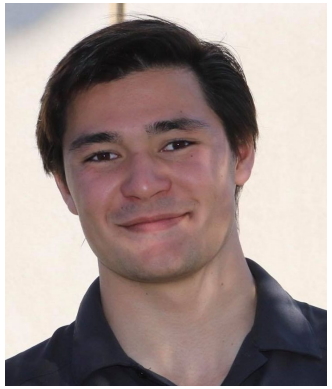
+

Parting thoughts: ML + protein engineering

1. Exciting times!
2. Are we close to ChatGPT4 for protein engineering? No.
3. AF2/3 super important, but doesn't solve design task.
4. Predicting function (generally) will remain difficult problem for a long time.
5. Generative models cool and powerful, but don't solve the need to understand/extrapolate on designed properties from predictive models.
6. Far less data than in text, vision—will need to be much more clever for the answers.

The group of people who make it happen

Stephan Allenspach, PhD (postdoc)



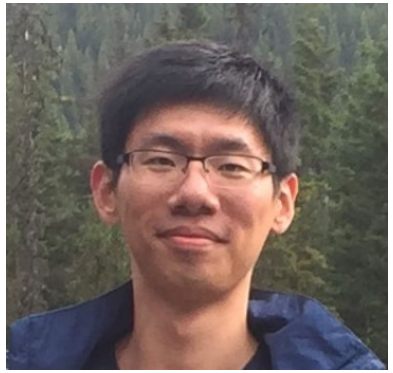
James Bowden (PhD student)

David Brookes, PhD (now at Dyno)

Akosua Busia, PhD (now at Dyno)

Clara Fannjiang, PhD (now at Genentech)

Chloe Hsu, PhD (now at startup)



Hanlun Jiang, PhD (postdoc)

Hunter Nisonoff (PhD student)

Junhao (Bear) Xiong (PhD student)

Come join us!

- PhD apps via EECS, BioE, CCB programs.
- Postdocs contact me.