Protein design

CS 279 / BIOE 279 / BIOPHYS 279 / BMDS 245 / CME 279

Oct. 29 and Nov. 5, 2026

Ron Dror

No classes on Tuesday (Democracy Day)

Outline

- What is protein design, and why do it?
- Structure-based sequence design
  - Traditional approach: energy minimization
  - Newer approach: machine learning
- Structure design
- How well does protein design work?
- Complementary experimental methods
- Protein language models for protein design
- Examples of successful designs

What is protein design, and why do it?

Overall goal

- Design a protein to serve a particular purpose
  - Want a protein with a desired function or structure (or with multiple desired characteristics)
- In particular, choose an appropriate amino acid sequence

Sample applications

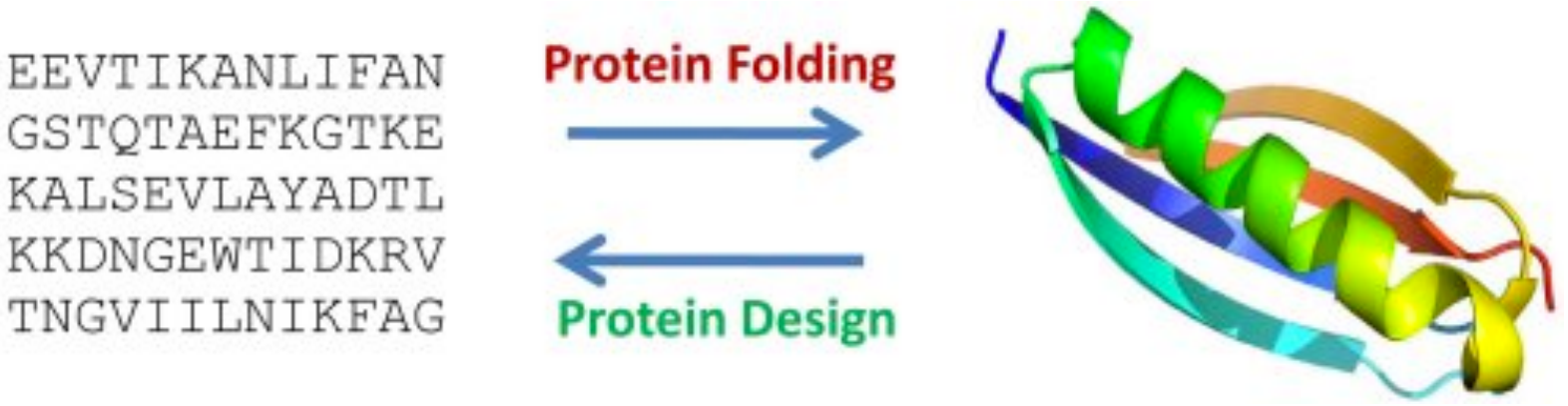
- Designing enzymes (proteins that catalyze chemical reactions)
  - Useful for production of industrial chemicals and drugs
  - Potential environmental applications: degrading toxins or producing biofuels
- Designing proteins that bind specifically to other proteins
  - Great potential for disease treatment (e.g., HIV, cancer, Alzheimer's)
  - Important special case: antibody design
- Designing sensors (proteins that bind to and detect the presence of small molecules—e.g., by lighting up or changing color)
  - Calcium sensors used to detect neuronal activity in imaging studies
  - Proteins that detect TNT or other explosives, for mine detection
- Making a more stable variant of an existing protein (to facilitate experimental investigation)

What is protein design?

- This is a rapidly evolving area, and even the definition of “protein design” is evolving
- I will focus on structure-based protein design, because that’s what’s consistently done computationally
  - This is what David Baker won the 2024 Nobel Prize (award for “computational protein design”)
  - I’ll more briefly cover experimental approaches and other computational approaches

Classical problem definition for “computational protein design”: structure-based sequence design

- Given the desired three-dimensional structure of a protein, design an amino acid sequence that will assume that structure.
  - Of course, a precise set of atomic coordinates would determine the sequence. Usually we start with an *approximate* desired structure.
  - The problem of designing an amino acid sequence that will adopt a desired backbone structure is also known as “structure-based sequence design” (or simply “sequence design,” or “inverse folding”)



http://www.riken.jp/zhangiru/images/sequence_protein.jpg

How do we choose the desired structure?

- Until recently, this was typically done with various *ad hoc* methods
- Lately, it's become more systematic, and is now sometimes called "structure design" (or "structure generation")
  - I will use the term "structure design"

Typical protein design workflow

1. Based on design goals (e.g., desired function), choose structural requirements (e.g., some small part of the protein must adopt a particular structure, with the rest holding that part in place) — Human expert judgement
2. **Structure design**: select a backbone structure that is compatible with structural requirements and that is "designable" (i.e., "achievable" by real proteins) — Computation (or Human expert judgement)
3. **Sequence design**: select an amino acid that will adopt (fold into) the desired backbone structure — Computation
4. Perform wet-lab experiments to check which designs work, and (optionally) to improve them — Wet-lab experiments

Structure-based sequence design

The “direct” approach (doesn’t work in practice!)

- Given a target structure, search over all possible protein sequences
- For each protein sequence, predict its structure, and compare to the target structure
- Choose the best match

Why doesn't the "direct" approach work?

- Computationally intractable
  - Huge number of sequences to consider
    - 20^N possible sequences with N amino acid residues
- May not be good enough!
  - Protein structure prediction remains imperfect, especially for proteins substantially different from naturally occurring ones
  - We want to maximize the probability of the desired structure (compared to all other possible folded and unfolded structures)
    - We could do this by sampling the full Boltzmann distribution for each candidate sequence (e.g., with very long MD simulations) … but that's very difficult for even one sequence!

Traditional approach: Energy minimization

This traditional approach has largely been replaced by the machine learning approaches I'll describe next, but I've included the traditional approach here because it helps understand why and when machine learning approaches work (and is often used by experts along with machine learning approaches)

Simplify the problem by making a few assumptions

1. Assume the backbone geometry is fixed
2. Assume each amino acid can only take on a finite number of geometries (*rotamers*)
3. Assume that what we want to do is to maximize the free energy drop from the completely unfolded state to the target geometry
   - In other words, simply ignore all the other possible folded structures that we want to avoid

The simplified problem

- At each position on the backbone, choose a rotamer (an amino acid type and a side-chain geometry) to minimize overall energy
  - Assume our energy function specifies a free energy. The Rosetta all-atom force field is the most common choice.
  - For each amino acid sequence, energy is measured relative to the unfolded state.
  - Assume that energy can be expressed as a sum of terms that depend on one or two rotamers each. This is the case for the Rosetta force fields (and for most other force fields).
- Thus, we wish to minimize total energy E_T, where

$$E_T = \sum_i \left[E_i(r_i) + \sum_{i \neq j} E_{ij}(r_i, r_j) \right]$$

Note that r_i specifies both the amino acid residue at position i and that residue's side-chain geometry

Optimization methods

- Today, this optimization (minimization) optimization is almost always solved using heuristic methods
  - Not guaranteed to find optimal solution, but much faster than exact methods
  - The most common choice is Metropolis Monte Carlo
    - Moves may be as simple as randomly choosing a position, then randomly choosing a new rotamer at that position
    - May decrease temperature over time (simulated annealing)

Optional: “Flexible backbone” design

- One of our key simplifying assumptions was that of a fixed backbone geometry.
- For many applications, protein design works better if you give the backbone some limited “wiggle room.”
- This requires optimizing simultaneously over rotamers and backbone geometry.
  - Often addressed through a Monte Carlo search procedure that alternates between local tweaks to backbone dihedrals and changes to side-chain rotamers

Optional: Negative design

- Another simplifying assumption was that we simply minimize the energy of the desired structure
  - We do not consider all other possible structures. It's possible that their energy ends up even lower.
- In negative design, we identify a few structures that we want to *avoid*, and we try to keep their energies high during the design process.
  - This can help, but we cannot explicitly avoid all possible incorrect structures without making the problem much more complicated. So the overall approach is still heuristic.
- One can also perform structure prediction for designed sequences as a way to filter out ones that fold to an undesired structure

Structure-based sequence design

Newer approach:
Machine learning

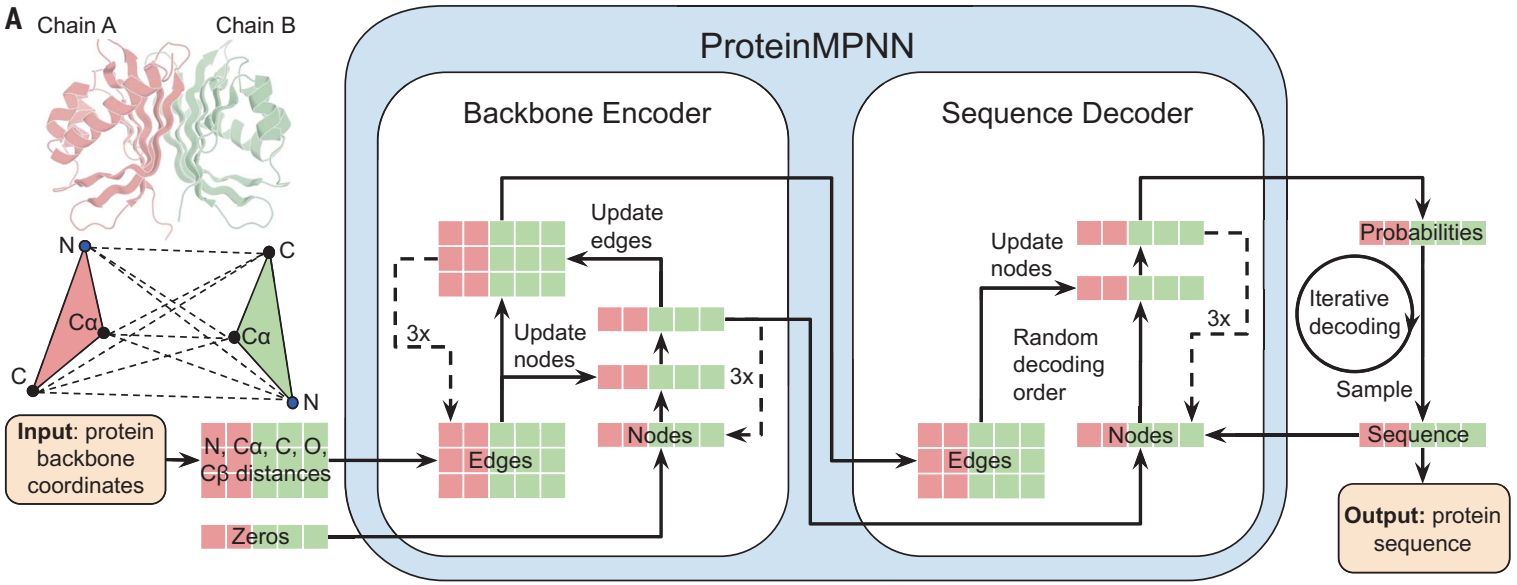
ProteinMPNN exemplifies the current dominant approach

- Many related recent methods, including LigandMPNN, SolubleMPNN, BoltzIF
- Basic idea: train a machine learning model (a deep neural network) to predict sequences of real proteins given their backbone structures
  - Importantly, add noise to the backbone coordinates when training
- ProteinMPNN goes through the sequence positions in a random order, predicting amino acid identities based on (1) backbone geometry and (2) amino acid identities already selected at other positions
  - ProteinMPNN can be run may times, generating a different sequence each time. Each of these is a candidate design.
  - For each sequence position, ProteinMPNN predicts the probability for each amino acid type.

Optional material

ProteinMPNN

- MPNN = message-passing neural network (operates on graphs)
- ProteinMPNN uses a graph to represent the backbone structure
  - That is, distances between atoms in the backbone



Dauparas …. Baker, *Science* 2023

Filtering results by structure prediction

- Like traditional sequence design methods, ProteinMPNN focuses on stabilizing the desired structure rather than destabilizing all other possible structures
- To help address this limitation, perform structure prediction on each designed sequence
  - Take each generated sequence and predict its structure using AF2, AF3, or a similar method
  - Select sequences whose predicted structure closely matches the target structure (by RMSD) and has high prediction confidence (according to metrics such as pLDDT, pTM, or pAE, which are returned by these prediction methods)

Structure design

Designing the backbone

- The first step of protein design is generally to select one or more target backbone structures.
- Traditionally, this was as much art as science
  - Real proteins can only adopt a limited set of backbone structures, but there isn't a great description of what that set is.
- Traditional methods to design backbone structure:
  - Use a a minor variation of an experimentally determined backbone structure
  - Use a fragment assembly program like Rosetta, selecting fragment combinations that fit some approximate desired structure
  - Assemble secondary structure elements by hand

Machine learning enables more systematic backbone design

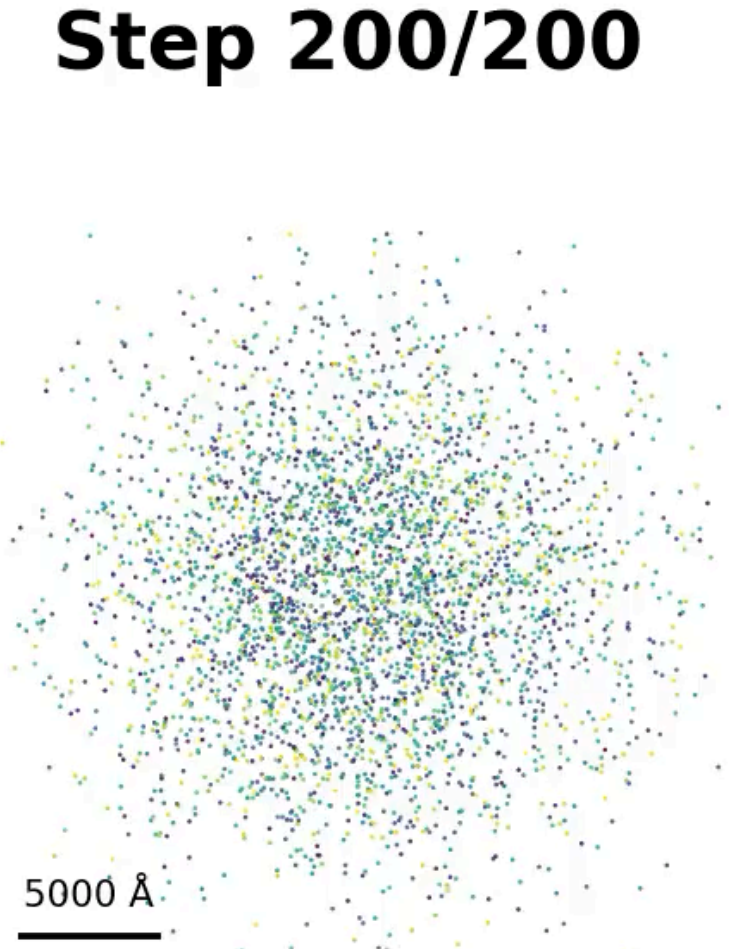
- A challenging problem
  - This isn't simple prediction. Instead, it requires generating backbones that satisfy criteria for a given design and that will be adopted by one or more actual amino amino acid sequences.
- RFDiffusion exemplifies the current dominant approach
  - It uses denoising diffusion (like AF3)

Watson … Baker, *Nature*, 2023

Article

De novo design of protein structure and function with RFdiffusion

Recall how AF3 uses denoising diffusion: it starts with the atoms in random positions and predicts how they should move to approach their correct positions, given information about the query protein



Movie by Jessica Karaguesian

RFDiffusion vs. AF3: Key differences

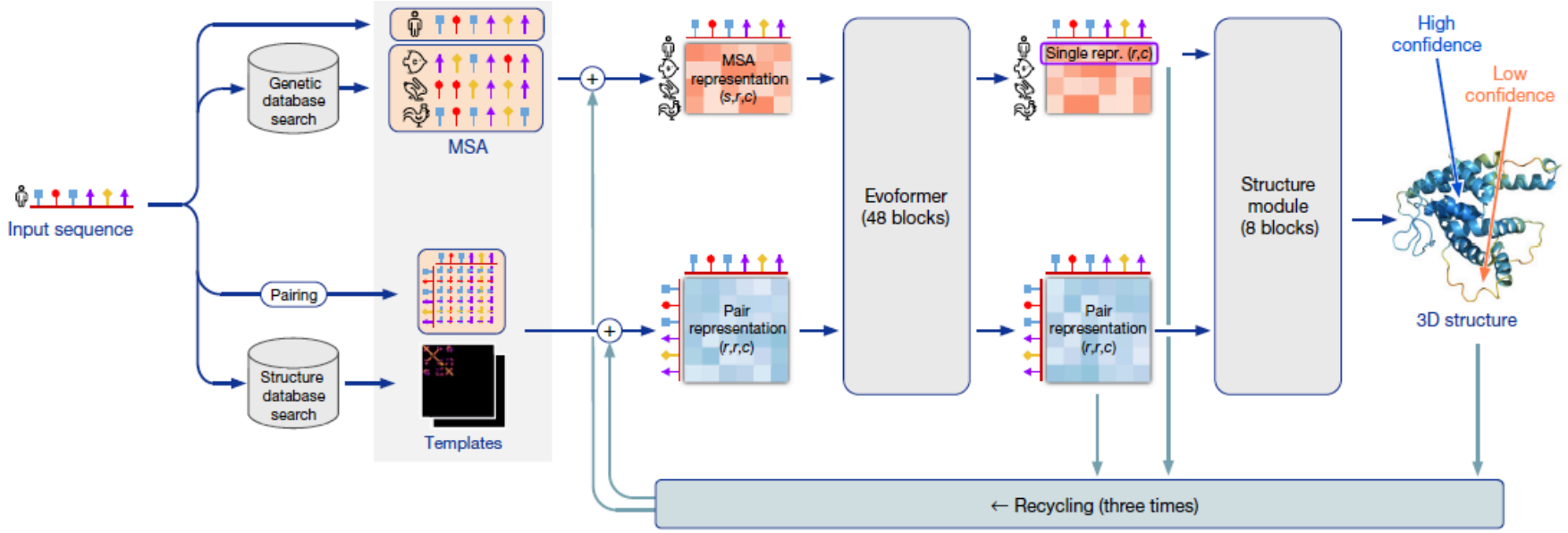
- In AF3, denoising diffusion is conditioned on information about the query protein (from sequence, MSA, and templates). In RFDiffusion, denoising diffusion is conditioned on requirements for the designed protein.
- In AF3, all atoms (other than hydrogens) are represented explicitly. In RFDiffusion, only protein backbone atoms are represented explicitly, because we don't yet know which side chains will be present.
  - In more recent variants (e.g., RFDiffusion 3 and BoltzGen), some or all side chain atoms are also represented and allowed to change identity during the diffusion process. These methods determine amino acid identities during diffusion but adjust them afterwards with ProteinMPNN or a similar method.

Note: RFDiffusion was published *before* AF3

RFDiffusion

Key insight: one can make this denoising process work much better by using a structure predictor as a component

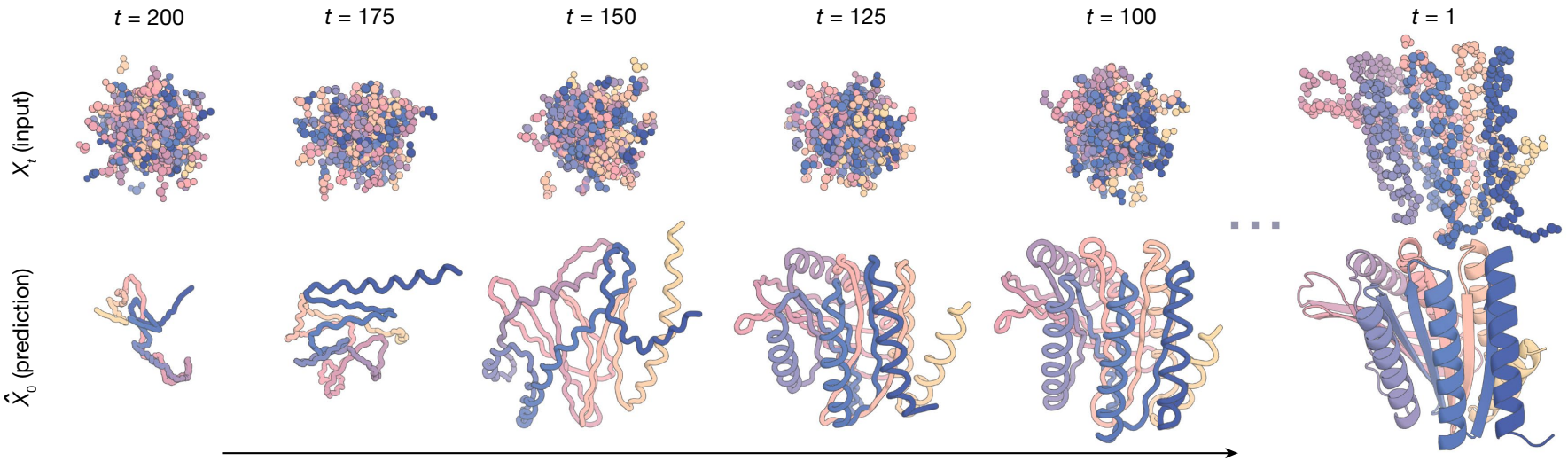
- RFDiffusion uses RoseTTAFold (similar to AF2) as a starting point, and tunes it for this application
- At each time step, RFDiffusion predicts the final structure but takes only a small step in that direction



AlphaFold 2 and recent versions of RoseTTAFold use “recycling,” meaning that they run the prediction process several times, feeding the output coordinates as inputs to the next prediction. This inspired the use of RoseTTAFold in RFDiffusion.

RFDiffusion

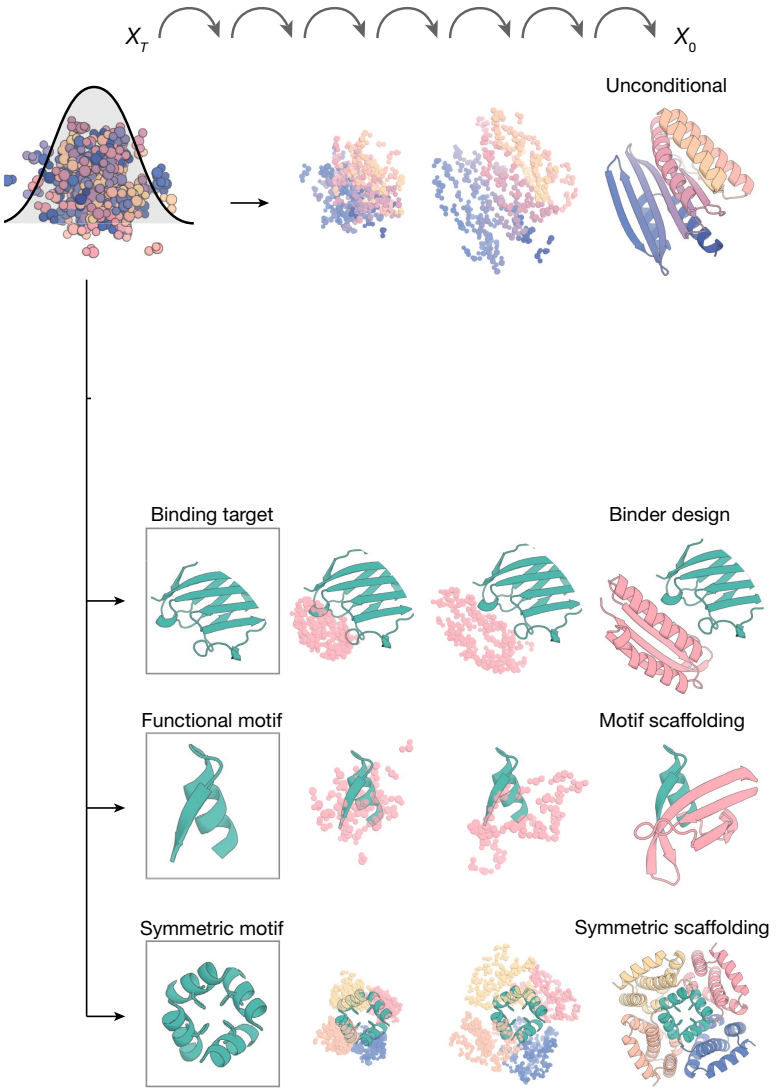
- At each time step, RFDiffusion predicts the final structure (using a tuned version of RosettaFold) but adjusts atom positions by only a small step in that direction



Watson et al.,
Nature, 2023

RFDiffusion

- In making its predictions, the denoising (generative) process can use information on properties of the protein to be generated. This approach, known as “conditioning,” allows one to generate designs with desired properties.
  - For example, desired local structure (functional motif), symmetry, or binding target



Watson et al., *Nature*, 2023

An alternative approach: hallucination

- For some design objectives, one can use a structure prediction method (e.g., AF2) to calculate a metric that predicts how good a given designed sequence is
- One can then predict how to tweak the designed sequence to best improve the metric
  - One uses back-propagation to compute the direction of steepest descent
  - Typically one will end up with a combination of amino acids at each position (which the AF2 input representation allows)
- Recently this approach has been used very successfully for design of proteins that bind other proteins (e.g, the BindCraft method)
  - BindCraft still uses a ProteinMPNN-like method to redesign sequence

How well does protein design work?

How well does protein design work?

- Very impressive recent successes!
- However, one should keep in mind that:
  - Successful protein design projects often involve making and experimentally testing tens of candidate proteins (or more) to find a good one
  - Projects and design strategies that fail generally aren't published
  - Protein design is not yet a matter of simply "turning the crank," although methods like ProteinMPNN, RFDiffusion, and BindCraft help automate it
- Evaluating/quantifying/comparing the effectiveness of protein design methodologies is difficult
  - Checking if a design "works" requires wet-lab experiments
    - Structure prediction and other computational predictions are useful as filters but definitely don't guarantee that a particular design will be successful
  - To compare methodologies, one would need to synthesize and test many designed sequences for each methodology, and do this for many design problems
  - Different protein design projects may have very different goals, so there isn't a universal metric for how "good" a given designed sequence is

Complementary experimental methods

Computational protein design is often combined with experimental protein engineering methods

- For example, computational designs can often be improved by directed evolution
  - Directed evolution involves introducing random mutations to proteins and picking out the best ones
  - Usually this is done in living cells, with the fittest cells (i.e., those containing the “best” version of the protein) selected by some measure
- This is particularly powerful when designing proteins for a desired function that can be easily measured in cells (e.g., binding affinity or enzyme activity)



Frances Arnold

2018 Nobel Prize “for the directed evolution of enzymes”