Two-state protein design using deep learning

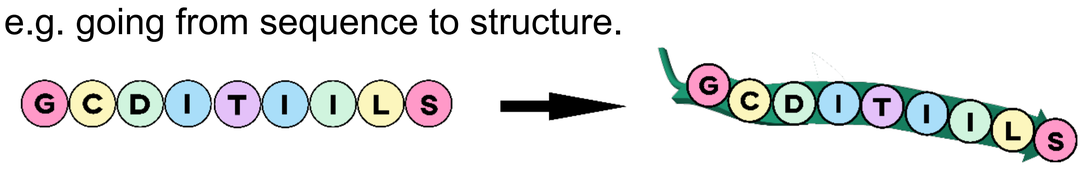
Sofia Andersson, Lund University
Centre for Molecular Protein Science
LUND UNIVERSITY

Motivation & Research goals

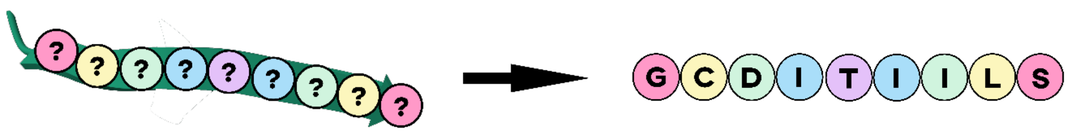
Most interesting protein functions are associated with proteins being able to exist in more than one structural state. While designing proteins that exist in one state is a mostly solved problem, designing multi-state proteins remains challenging and serendipitous. The current state-of-the-art for designing proteins is a combination of diffusion model RFDiffusion and sequence design model ProteinMPNN, with high experimental success rates. This project combines *de novo* backbone design of two states with a novel method of sequence design using amino acid probabilities and simulated annealing. Two sequences designed using this method have been expressed experimentally and display a temperature dependence for switching between states.

Protein design

AlphaFold revolutionized the protein structure prediction problem, e.g. going from sequence to structure.

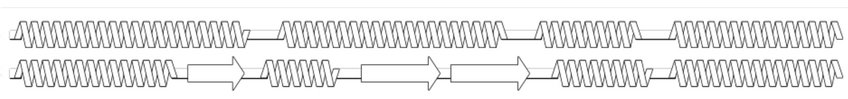


Protein design is doing the opposite; going from a desired structure to a sequence that results in it.



Protein switches

Protein switches are proteins that have the same amino acid sequence but take on different structures in some conditions. Usually only one section of the protein changes, as seen below.

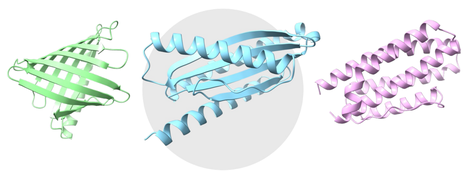


Sequence model

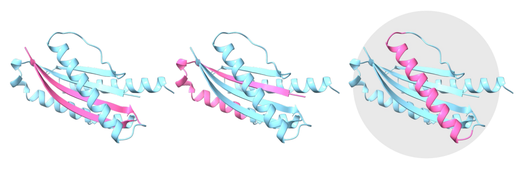
ProteinMPNN takes a protein backbone and returns a sequence based on a probability distribution of every amino acid acid a_i at each position i. These probabilities are dependent on each other. A sequence s can be represented as $s = (a_1, a_2, \dots, a_L)$.

Backbone design

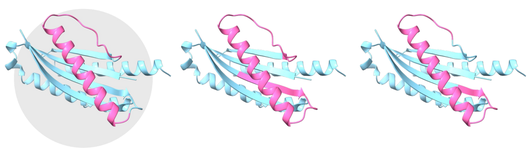
1. Generating backbone templates with RFDiffusion



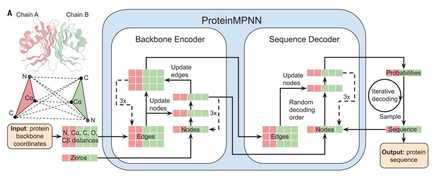
2. Trialling segment candidates for structural redesign using RFDiffusion



3. Sampling variations of candidate segment



4. Two-state ProteinMPNN sequence design



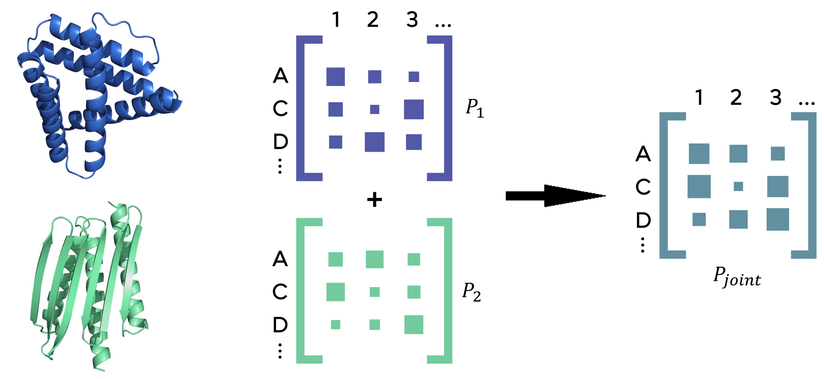
Diffusion-based model **RFDiffusion** is used to design the backbones. At each step, structural difference to the template is optimized.

Sequence design

Sequence design can be thought of as an optimization problem. Consider two probability distributions P_1 and P_2, representing the likelihood of each amino acid at each position based on the two desired protein structures. The goal is to find a joint probability distribution P_{joint} that combines the information from P_1 and P_2 to create a switch:

$$P_{joint} = \alpha P_1(\bar{a}_i|\text{structure 1}) + (1-\alpha)P_2(\bar{a}_i|\text{structure 2})$$

where α is a parameter determining the weight assigned to each structure.

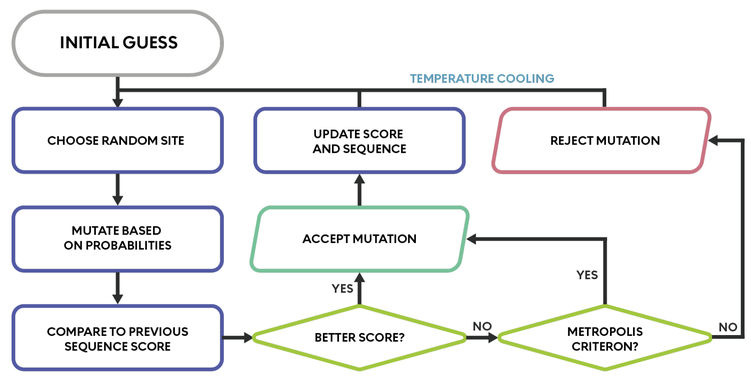


Exploring the sequence space

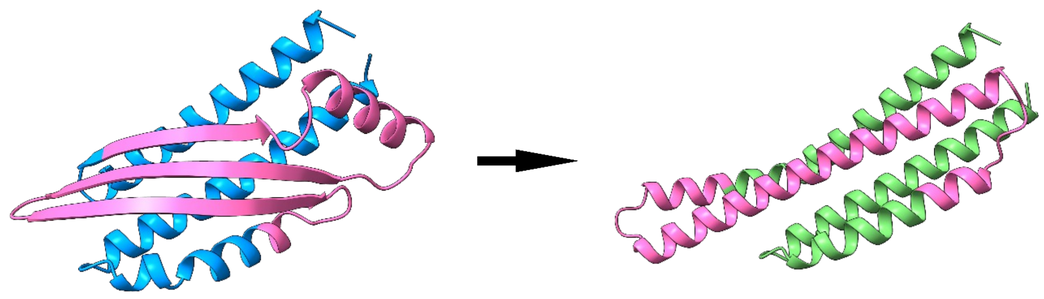
To take an initial guess at the joint sequence, s_{in}, you can take the most likely amino acid at each position for P_{joint}:

$$s_{in} = \sum_{i=1}^{L} \text{argmax}(\alpha P_1(\bar{a}_i|\text{structure 1}) + (1-\alpha)P_2(\bar{a}_i|\text{structure 2}))$$

This may not be the optimal solution. The sequence space needs to be explored for such a sensitive system. This is done through **simulated annealing**. The quality of the sequence is scored by a combination of different structural metrics.



Results



This method designed two sequences for this *de novo* protein switch with quite high **AlphaFold2** scores for both states (>70). Both sequences have been expressed experimentally and show strong indication of being temperature-dependent switches.

References

1. J. Dauparas *et al.*, “Robust deep learning–based protein sequence design using ProteinMPNN,” *Science*, vol. 378, no. 6615, pp. 49–56, Oct. 2022, doi: 10.1126/science.add2187.
2. J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, no. 7873, pp. 583–589, Jul. 2021, doi: 10.1038/s41586-021-03819-2.
3. J. P. Watson *et al.*, “De novo design of protein structure and function with RFdiffusion,” *Nature*, vol. 620, no. 7976, pp. 1089–1100, Jul. 2023, doi: 10.1038/s41586-023-06415-8.
4. Flaticon.com *(for some graphics resources)*

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM