



RFdiffusion

Ana C. Chang-Gonzalez
Meiler Lab Rosetta Workshop
March 3, 2025

RFdiffusion

Article

De novo design of protein structure and function with RFdiffusion


<https://doi.org/10.1038/s41586-023-06415-8>


Received: 14 December 2022

Accepted: 7 July 2023

Published online: 11 July 2023

Open access

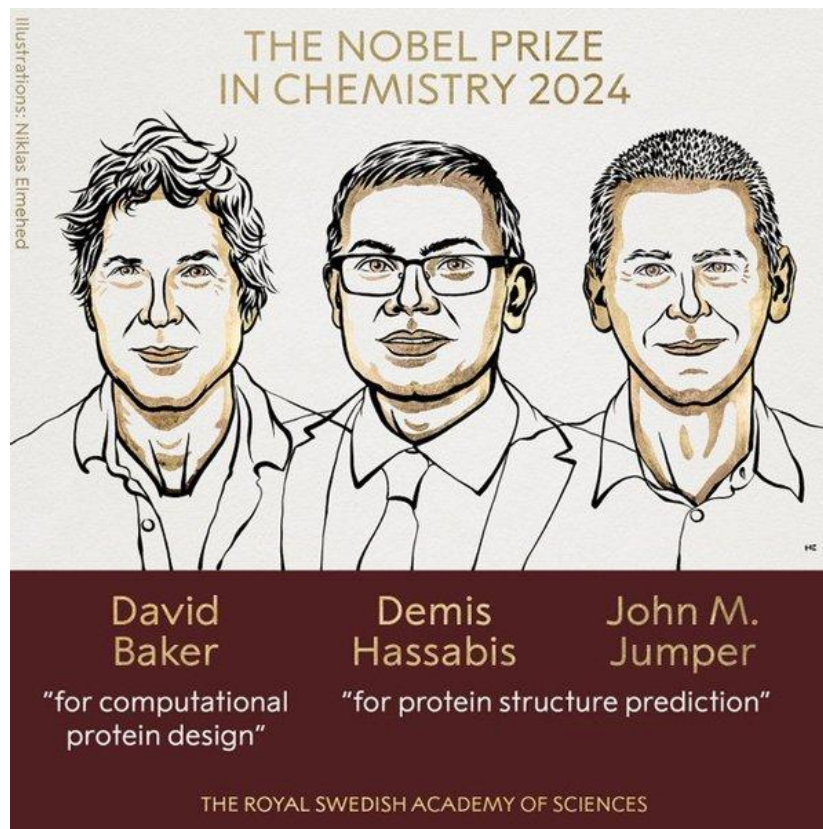
 Check for updates

Joseph L. Watson^{1,2,15}, David Juergens^{1,2,3,15}, Nathaniel R. Bennett^{1,2,3,15}, Brian L. Trippe^{2,4,5,15}, Jason Yim^{2,6,15}, Helen E. Eisenach^{1,2,15}, Woody Ahern^{1,2,7,15}, Andrew J. Borst^{1,2}, Robert J. Ragotte^{1,2}, Lukas F. Milles^{1,2}, Basile I. M. Wicky^{1,2}, Nikita Hanikel^{1,2}, Samuel J. Pellock^{1,2}, Alexis Courbet^{1,2,8}, William Sheffler^{1,2}, Jue Wang^{1,2}, Preetham Venkatesh^{1,2,9}, Isaac Sappington^{1,2,9}, Susana Vázquez Torres^{1,2,9}, Anna Lauko^{1,2,9}, Valentin De Bortoli⁸, Emile Mathieu¹⁰, Sergey Ovchinnikov^{11,12}, Regina Barzilay⁶, Tommi S. Jaakkola⁹, Frank DiMaio^{1,2}, Minkyung Baek¹³ & David Baker^{1,2,14} 

There has been considerable recent progress in designing new proteins using deep-learning methods^{1–9}. Despite this progress, a general deep-learning framework for protein design that enables solution of a wide range of design challenges, including de novo binder design and design of higher-order symmetric architectures, has yet to be described. Diffusion models^{10,11} have had considerable success in image and language generative modelling but limited success when applied to protein modelling, probably due to the complexity of protein backbone geometry and sequence–structure relationships. Here we show that by fine-tuning the RoseTTAFold structure prediction network on protein structure denoising tasks, we obtain a **generative model of protein backbones** that achieves outstanding performance on unconditional and topology-constrained protein monomer design, protein binder design, symmetric oligomer design, enzyme active site scaffolding and symmetric motif scaffolding for therapeutic

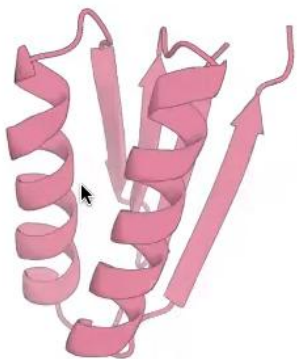
Why should we care about de novo protein design?

- Enable new functions
- Tune biochemical parameters
- Control protein function
- Create composite molecular machines, assemblies



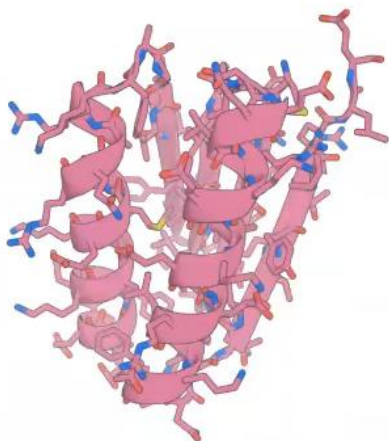
Backbone generation is a bottleneck in the protein design workflow

Backbone
Generation



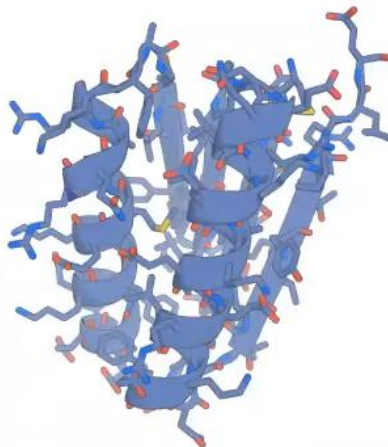
RFdiffusion

Sequence
Design



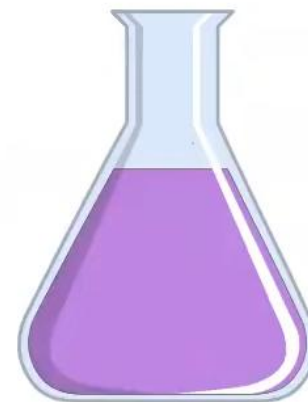
ProteinMPNN

Computational
Filtering



AlphaFold2
RoseTTAFold
Chai-1
ESMFold
OmegaFold
Boltz-1

Experimental
Characterisation



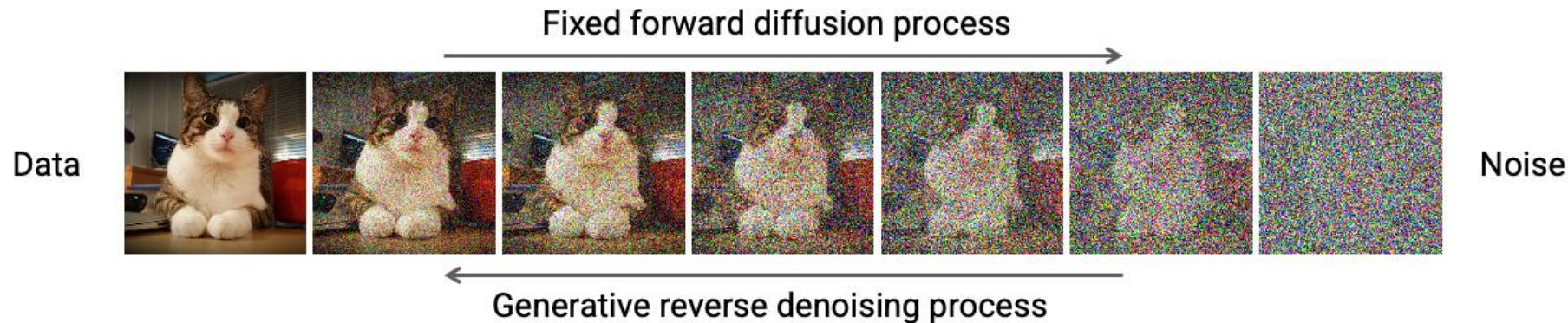
~ 1 week

*Figure from video "RFdiffusion"
by SBGrid Consortium.*

Outline

1. What is diffusion?
2. Introduction to RFdiffusion
3. How to use RFdiffusion

Diffusion models introduce random noise iteratively and learn to reverse the process during training

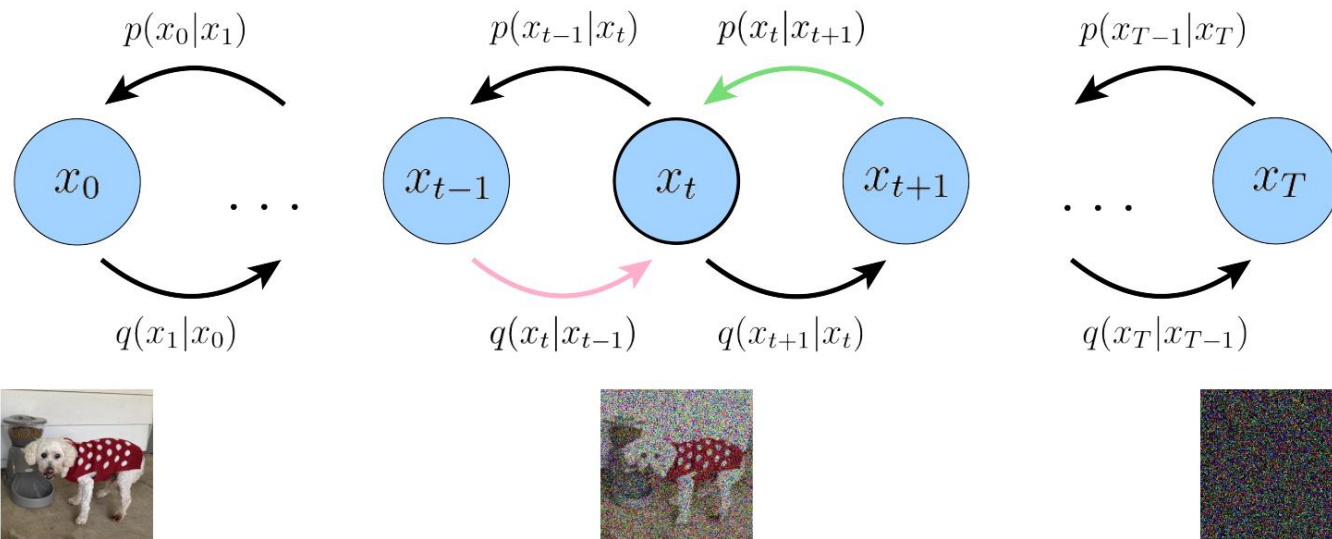


<https://cvpr2022-tutorial-diffusion-models.github.io/>

“Generate an image of a banana eating a hamburger while tap dancing.”



Denoising diffusion probabilistic models



C. Luo. arXiv. 2022

Outline

1. What is diffusion?
2. Introduction to RFdiffusion
3. How to use RFdiffusion

Diffusion occurs across translations and rotations

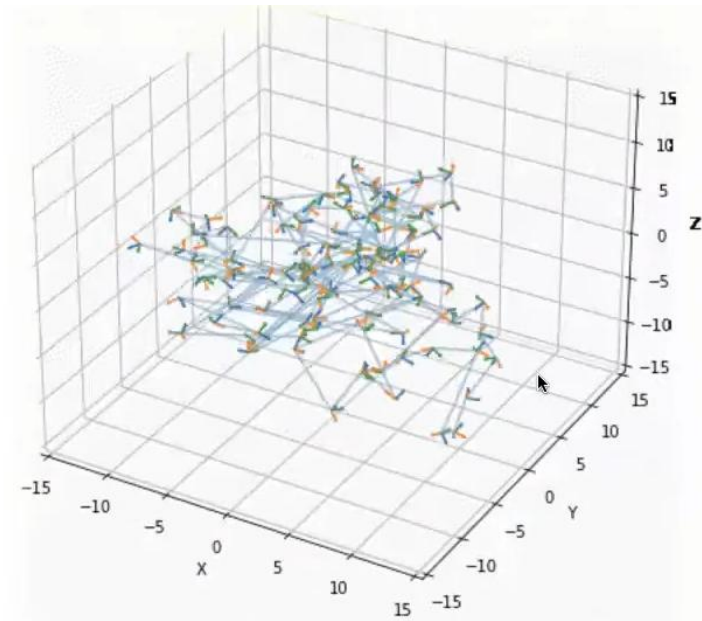
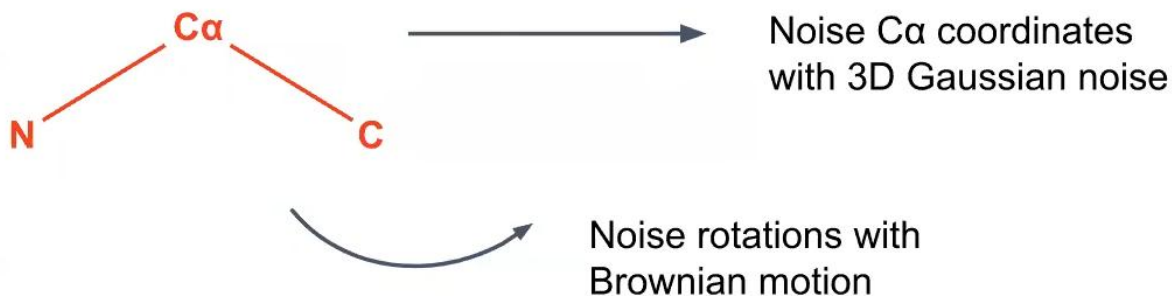
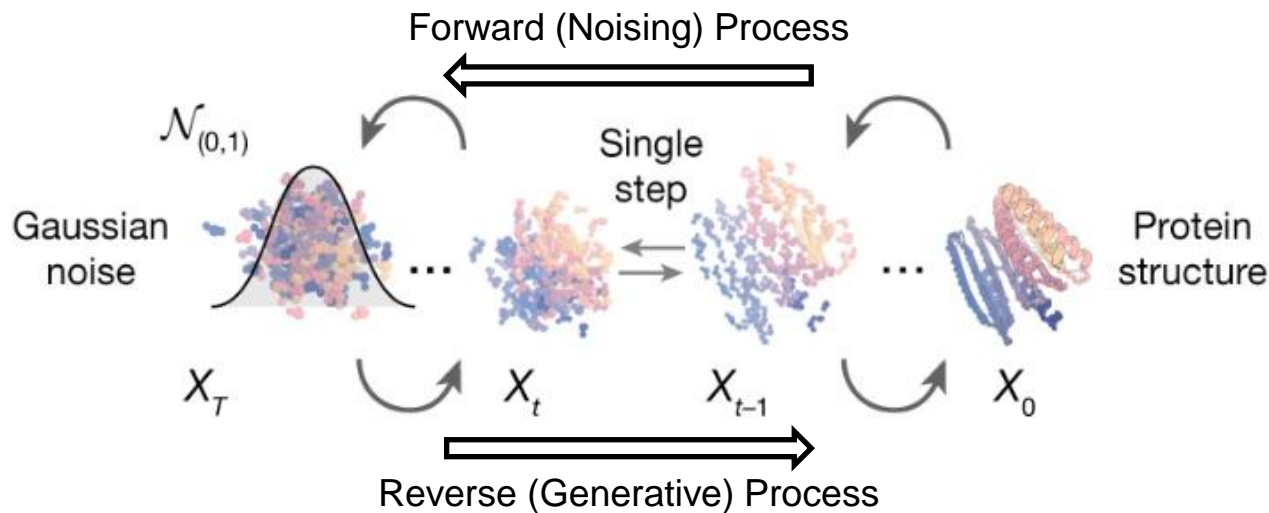
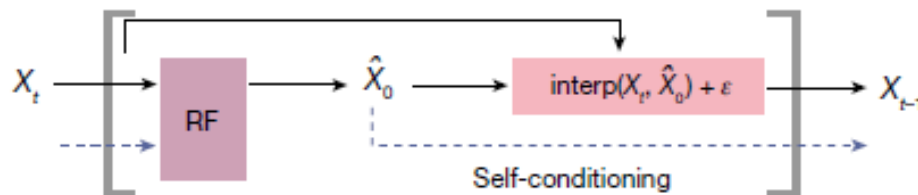


Figure from "RFDiffusion: Accurate protein design using structure prediction and diffusion generative models" video.

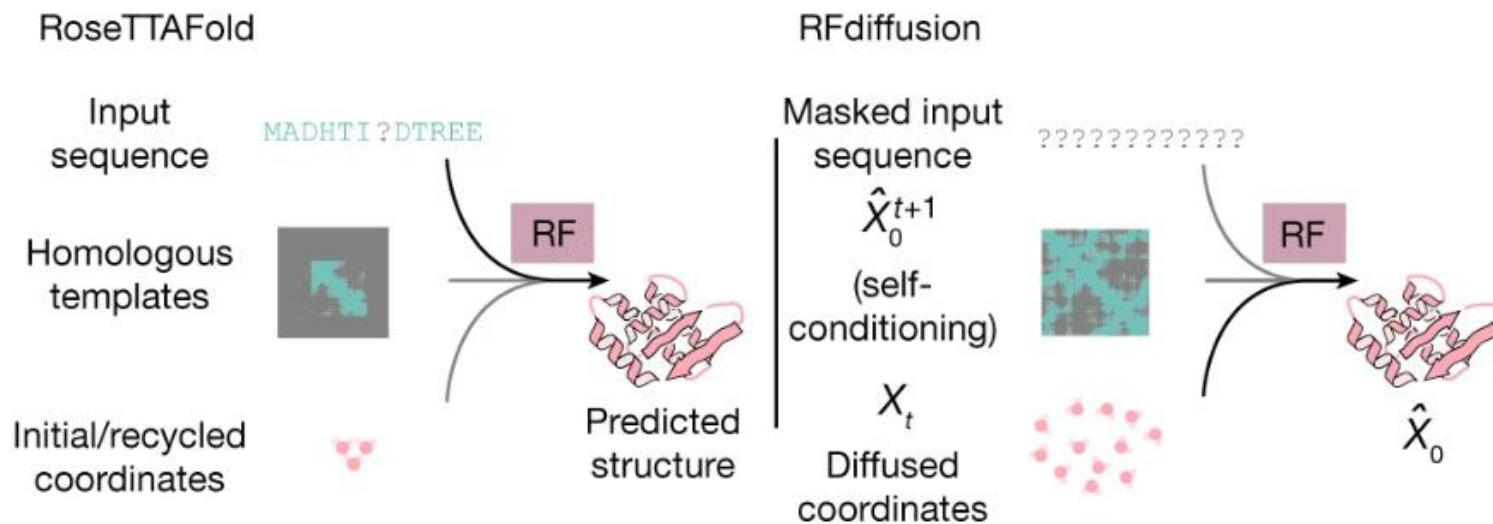
Utilizing diffusion models for use in protein design



Single RFdiffusion step

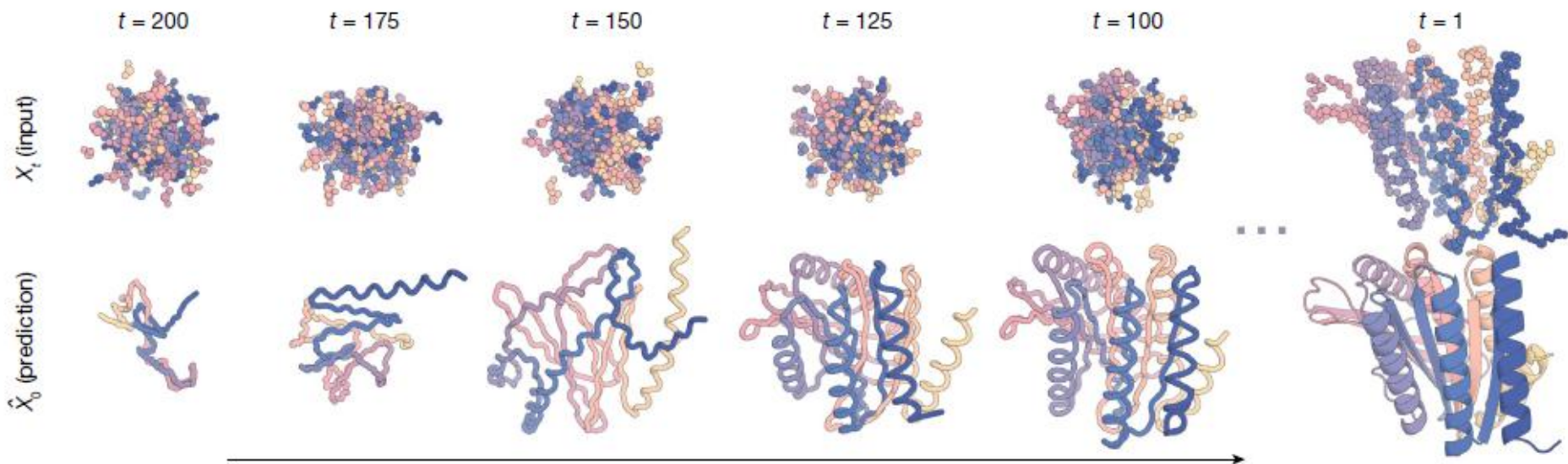


RoseTTAFold is used as the denoising framework



Watson et al. Nature. 2023.

Unconditional design trajectory for a 300-residue chain

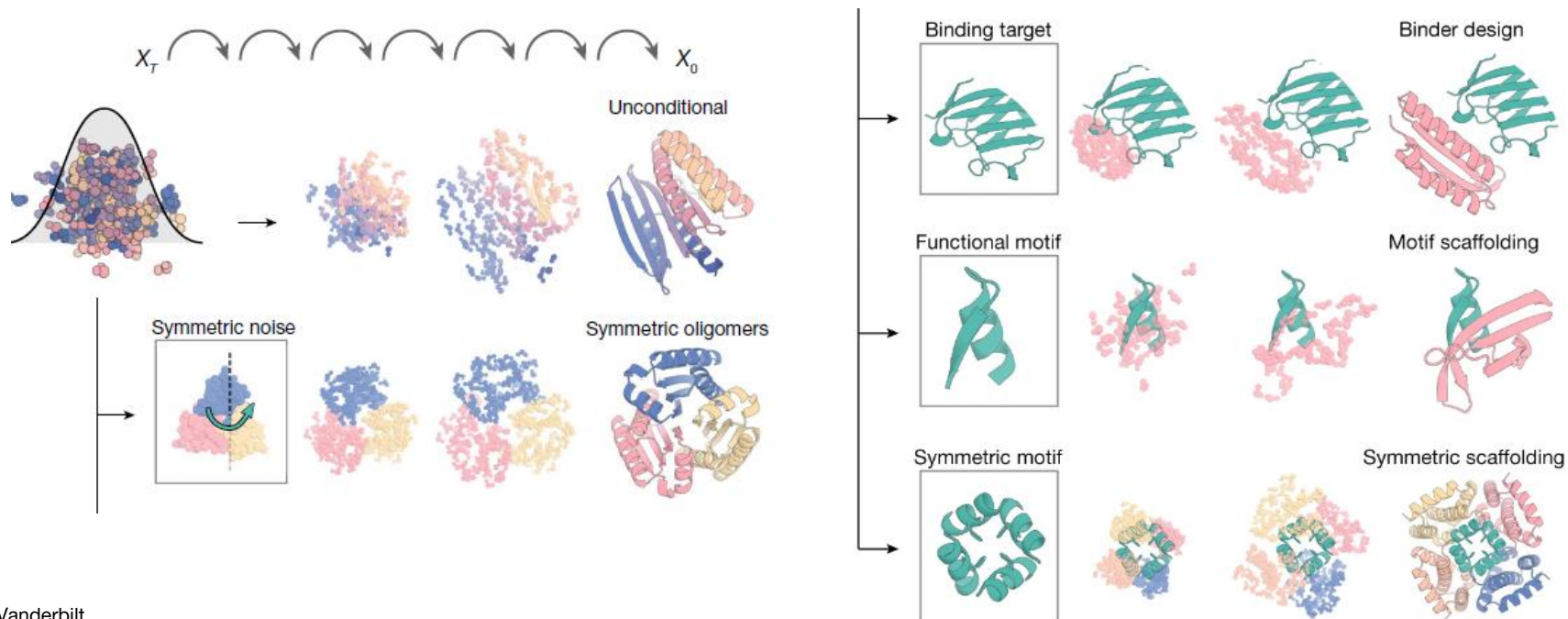


Watson et al. Nature. 2023.

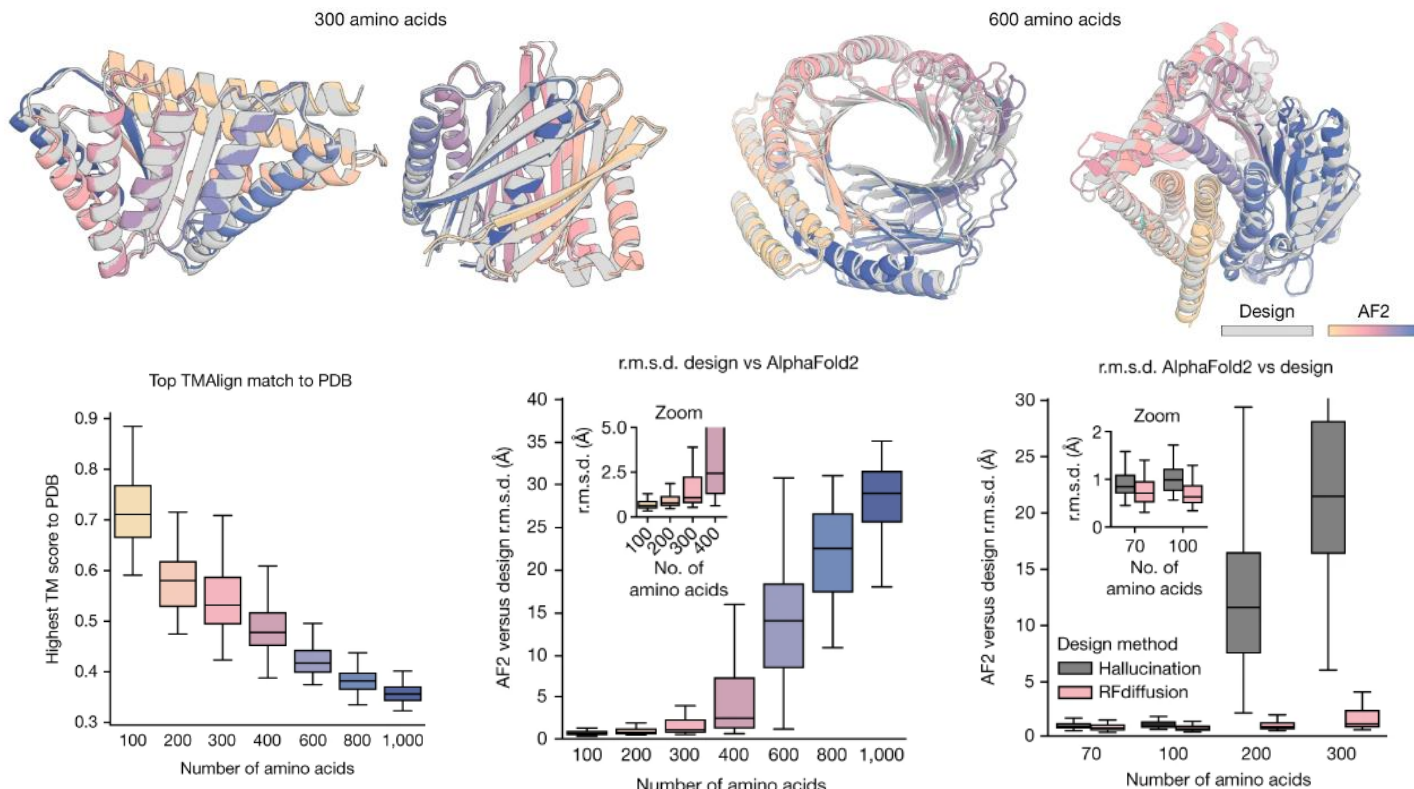
Broad applications of RFdiffusion

“Generate a protein that has 300 amino acids.”

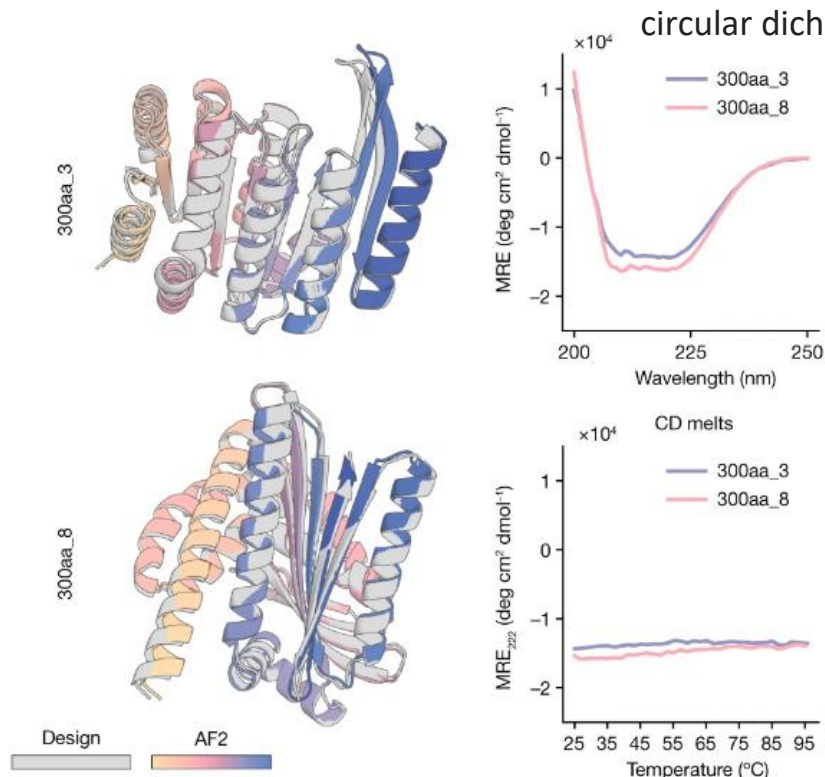
“Generate a protein that can bind to this target.”



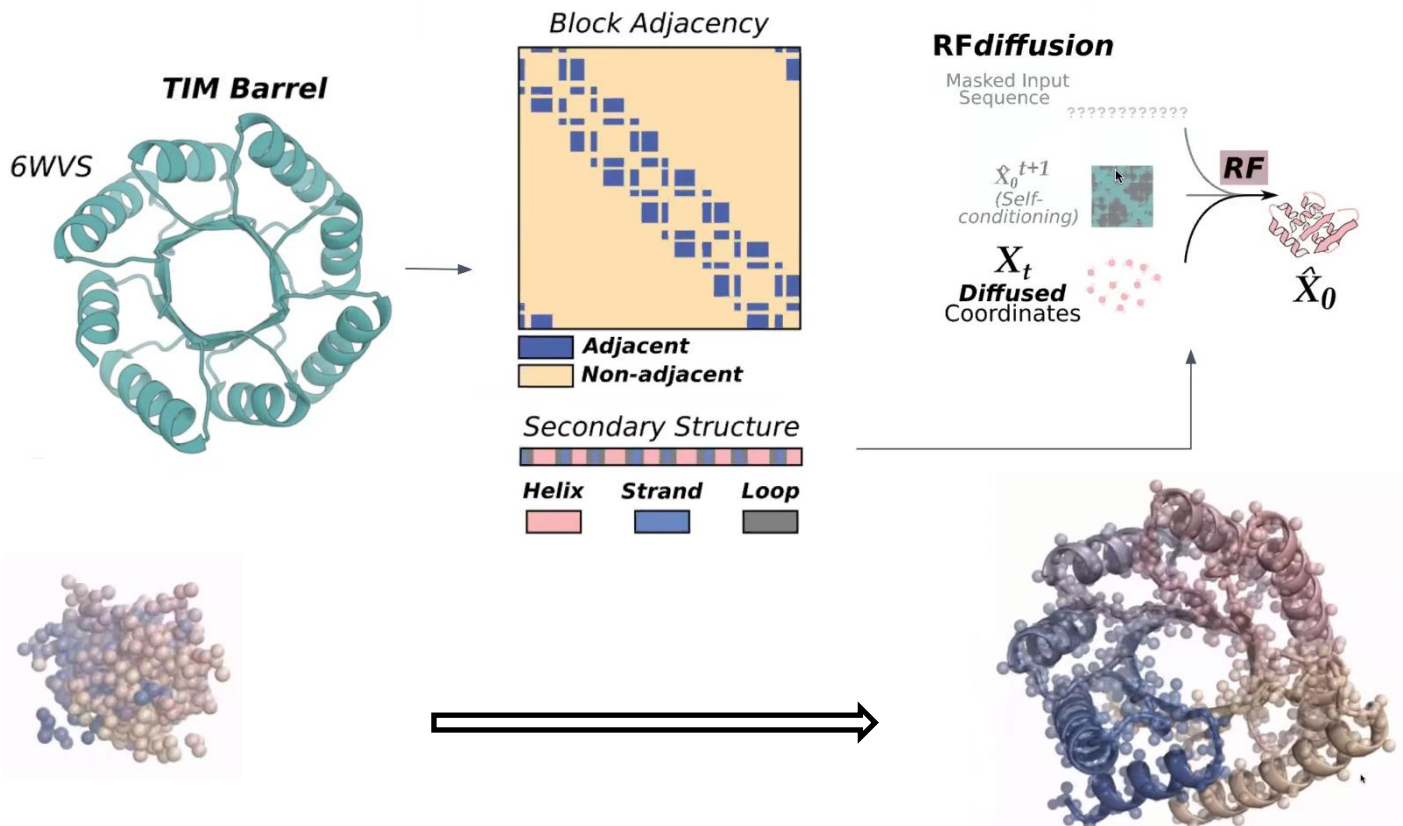
RFdiffusion produces diverse and new-to-nature proteins



Unconditional designs expressed as soluble monomers



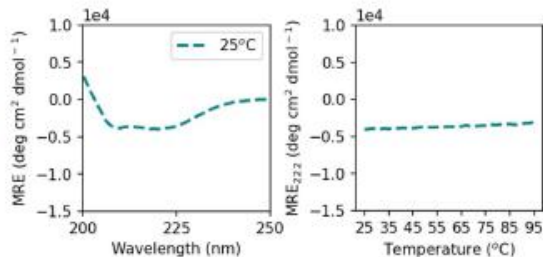
Condition designs to get a specific type of fold



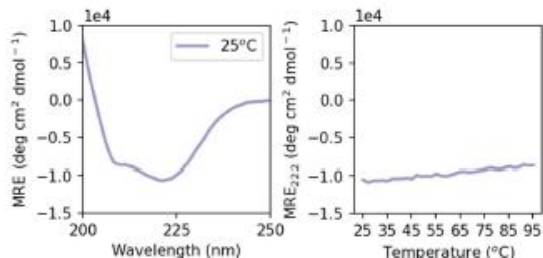
Watson et al. Nature. 2023.

Example TIM barrel-conditioned designs capture fold and are diverse

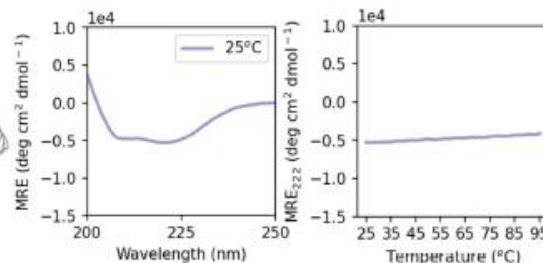
6WVS



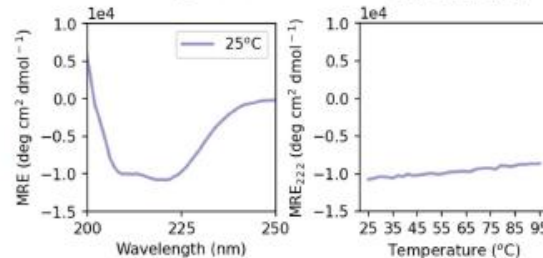
TIM_barrel_4



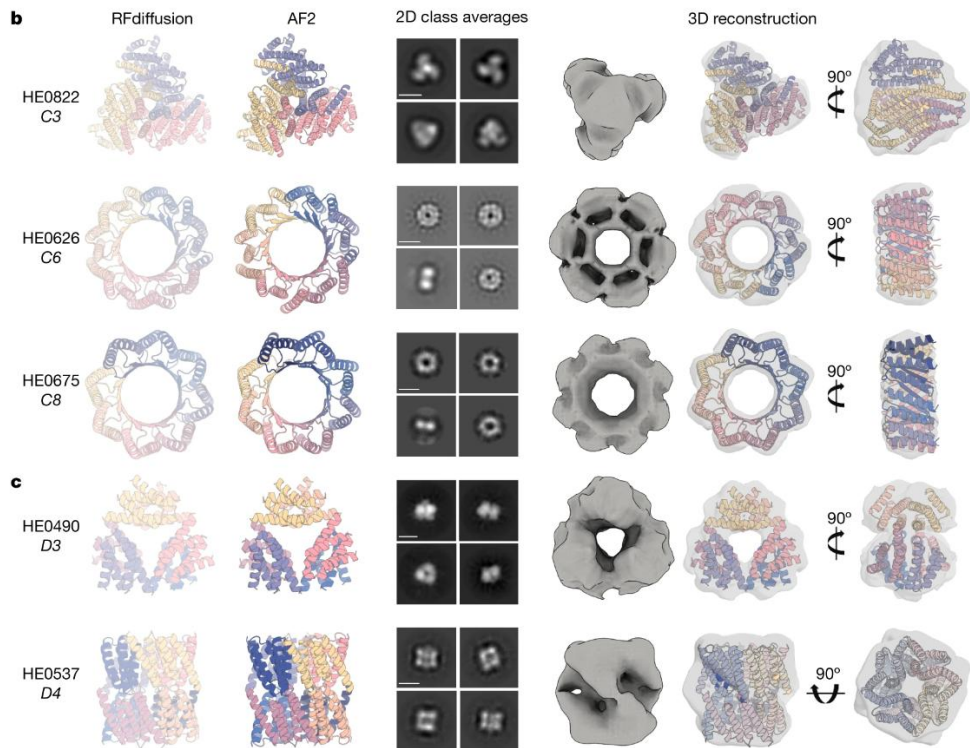
TIM_barrel_6



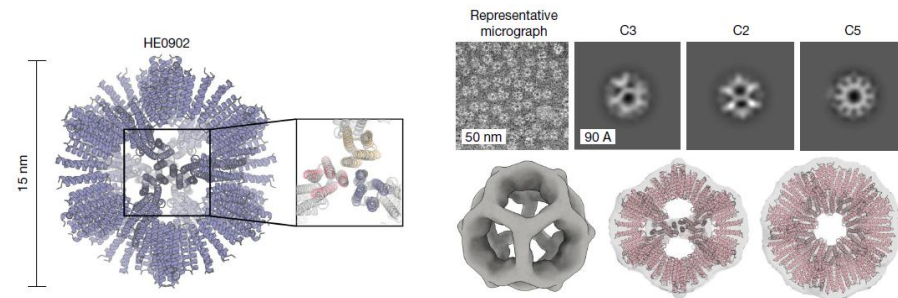
TIM_barrel_8



Symmetric oligomer design

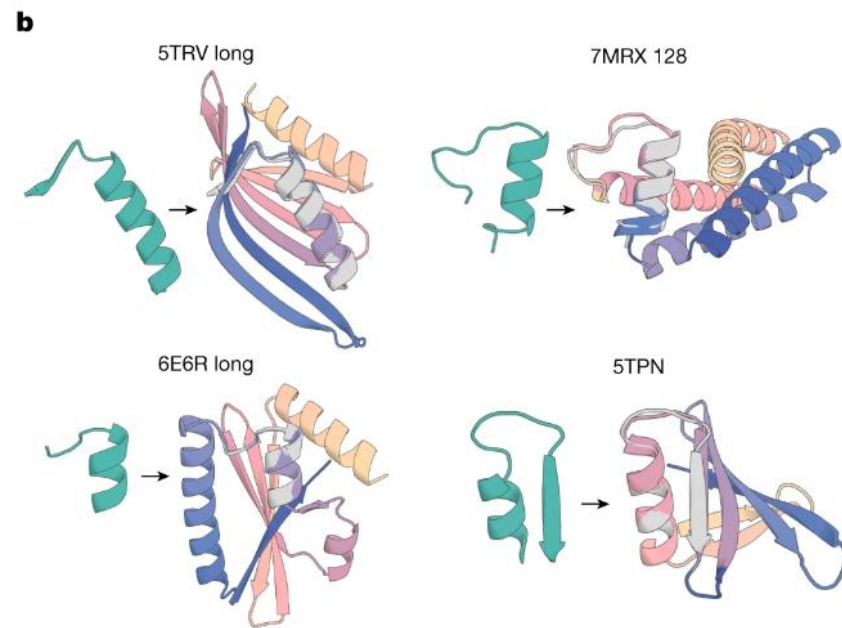
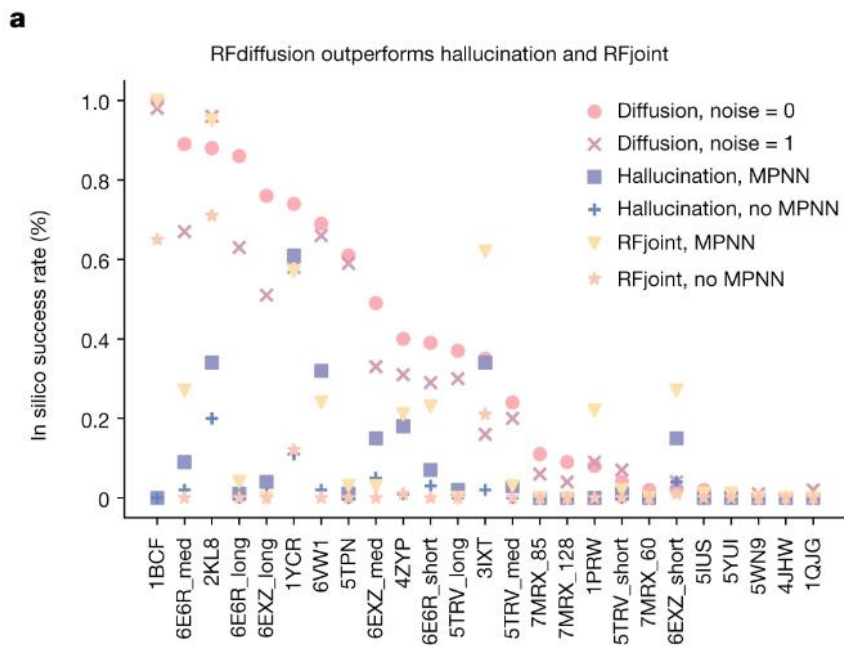


Design of icosahedral particle:

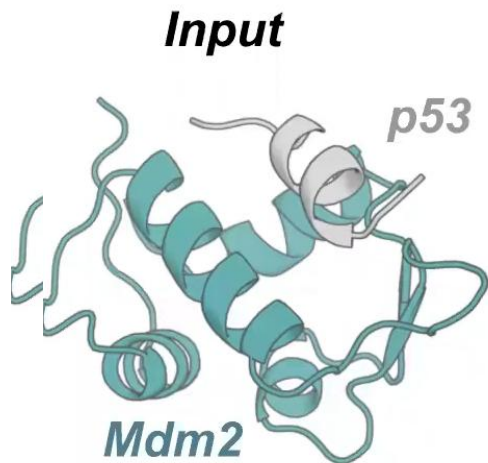


Generate proteins with given functional motifs

Given a functional motif, generate a protein which captures the structure of the input functional motif.

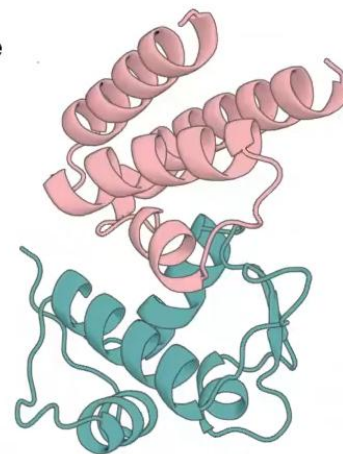
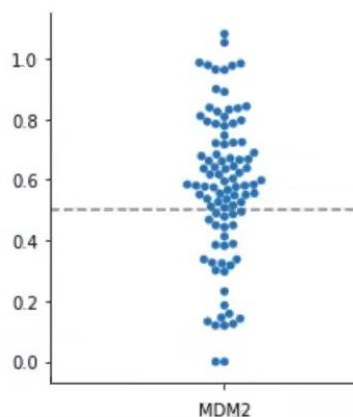


Designed binders show higher affinity for target than native binder

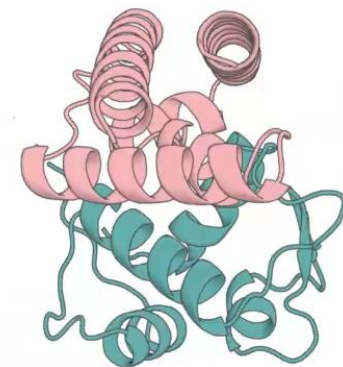


600nM affinity

> 50% experimental success rate



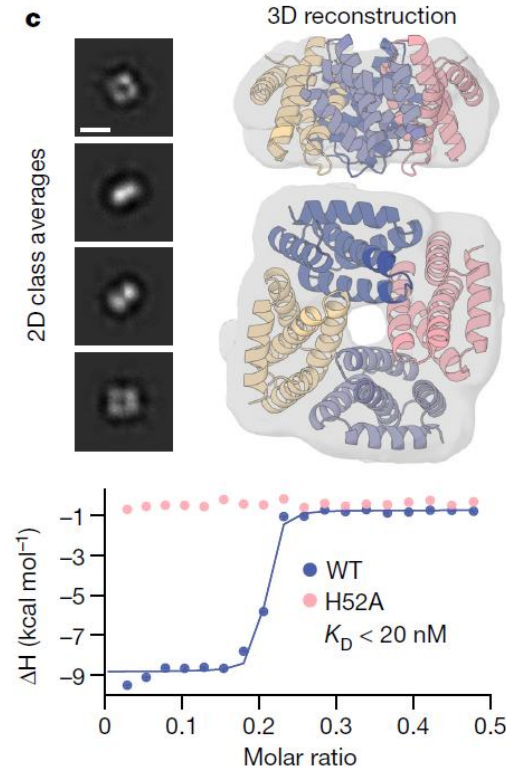
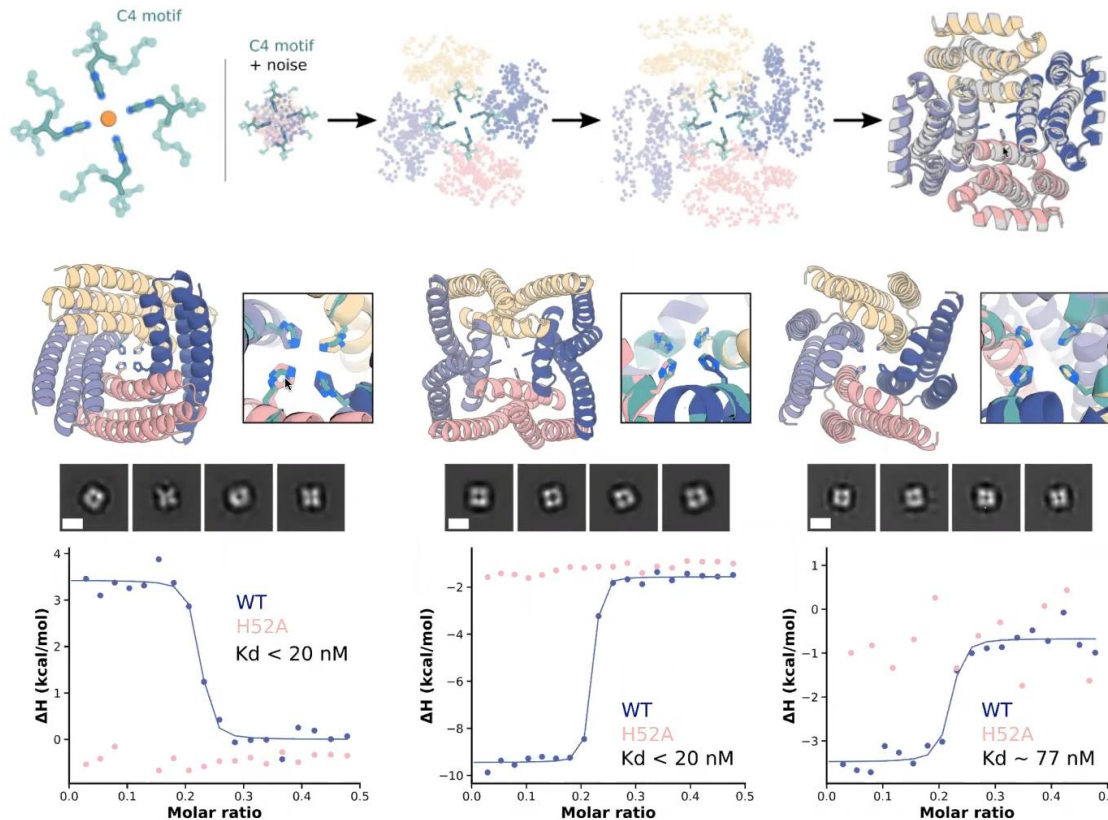
< 1nM



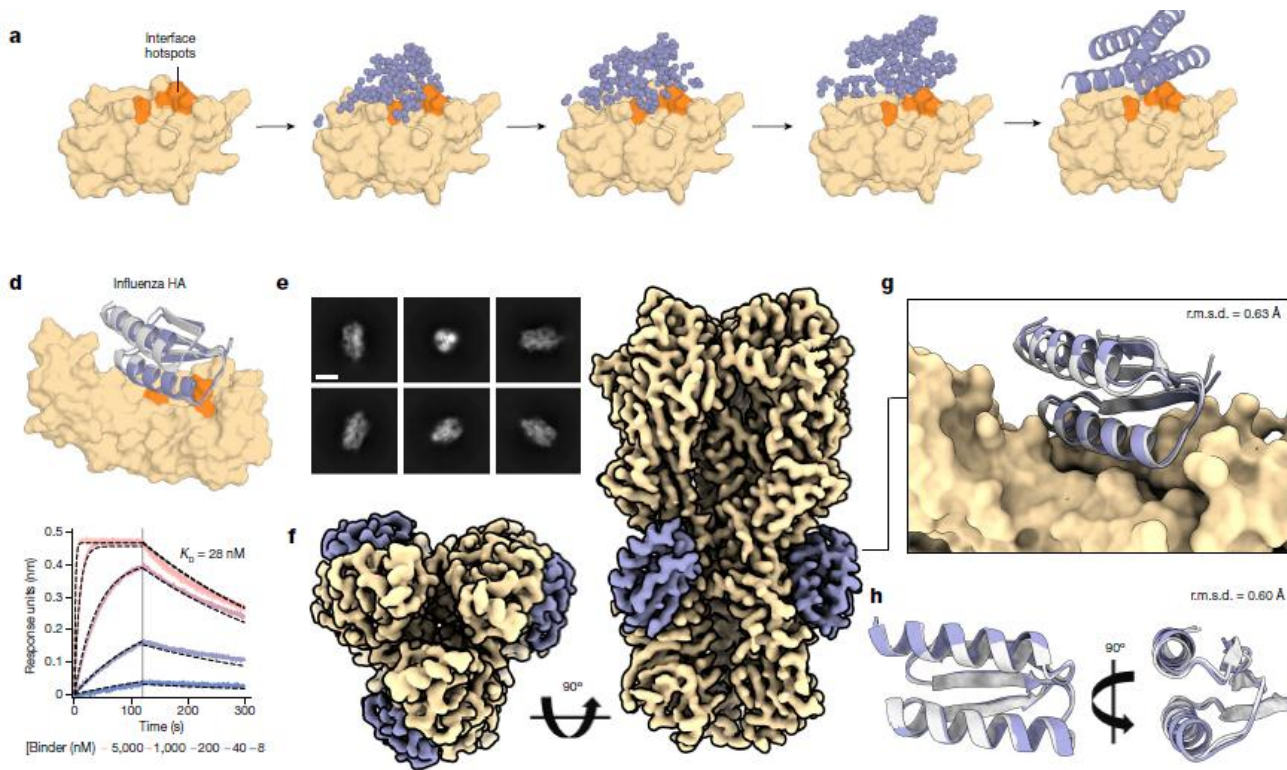
< 1nM

Figure from "RFDiffusion: Accurate protein design using structure prediction and diffusion generative models" video.

Symmetric metal-binding oligomers



De novo design of protein-binding proteins



Outline

1. What is diffusion?
2. Introduction to RFdiffusion
3. How to use RFdiffusion

How to use RFdiffusion?

Activate your conda environment!

- Example for unconditional monomer generation:

```
../RFdiffusion/scripts/run_inference.py \  
'contigmap.contigs=[150-150]' \  
inference.output_prefix=./output_monomer/monomer \  
inference.num_designs=5
```

Call the code: **~/rosetta_workshop/RFdiffusion/scripts/run_inference.py**

- Specify length of the protein with **contigmap.contigs**

Example: **'contigmap.contigs = [150-150]'**

- Specify the output

Example: **inference.output_prefix=path_to_output**

- Set the number of backbones you want to generate

Example: **inference.num_designs = 5**

Output of unconditional monomer generation

Output files are in **output_monomer** folder.

All outputs are poly-Glycine sequence.

- **PDB file** contains generated backbone
- **.trb file**, contains all metadata with the run, including input options
- **traj folder**, contains trajectory files (pdb)

Use PyMOL for visualization

```
pymol monomer_0.pdb
```

Motif scaffolding

- Example for generation conditioned on input motif:

```
../RFdiffusion/scripts/run_inference.py \  
'contigmap.contigs=[20-30/E422-440/40-50]' \  
inference.input_pdb=6UYG.pdb \  
inference.output_prefix=./output_scaffold/scaffold \  
inference.num_designs=1
```

- Specify motifs using **'contigmap.contigs'**

Example: **'contigmap.contigs=[20-30/E422-440/40-50]'**

- Motif is prefixed by letter followed by residue specification (**E422-440**)
- Build a backbone with 20-30 residues at the N-terminal of the motif, then 422-440 from chain E of the input, followed by 40-50 residues
- Specify input PDB with **'inference.input_pdb=path'**

Motif scaffolding with additional chains to avoid clashes

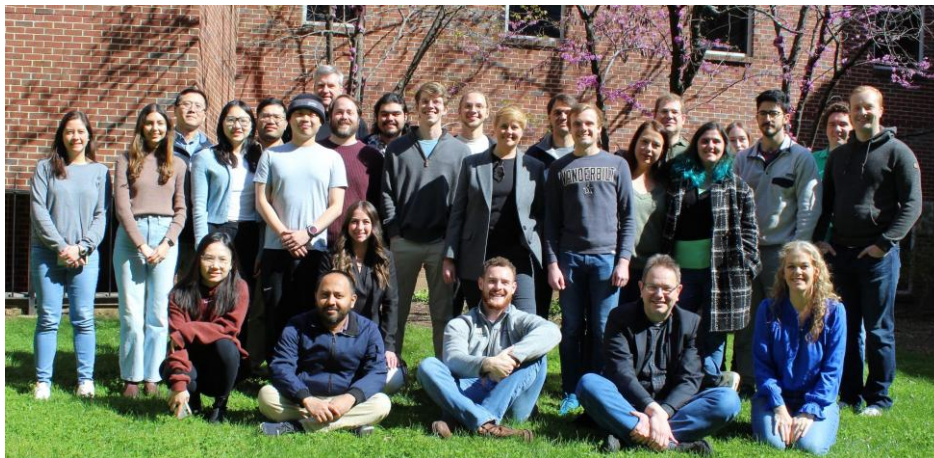
- Example where we can input structure from additional chains to avoid clashes
- Use **/0** followed by a space to symbolize a chain break

```
cd ../../
../RFdiffusion/scripts/run_inference.py \
'contigmap.contigs=[10-20/E422-440/40-50/0 H4-22/H29-71/H77-113/0
L2-29/L31-95/L97-109]' \
inference.input_pdb=6UYG.pdb \
inference.output_prefix=./output_scaffold/scaffold_Ab \
inference.num_designs=1
```

Heavy chain info

Light chain info

Thank You!



Dr. Ana C. Chang-Gonzalez
Postdoctoral Scholar

ana.chang-gonzalez@vanderbilt.edu
www.meilerlab.org

