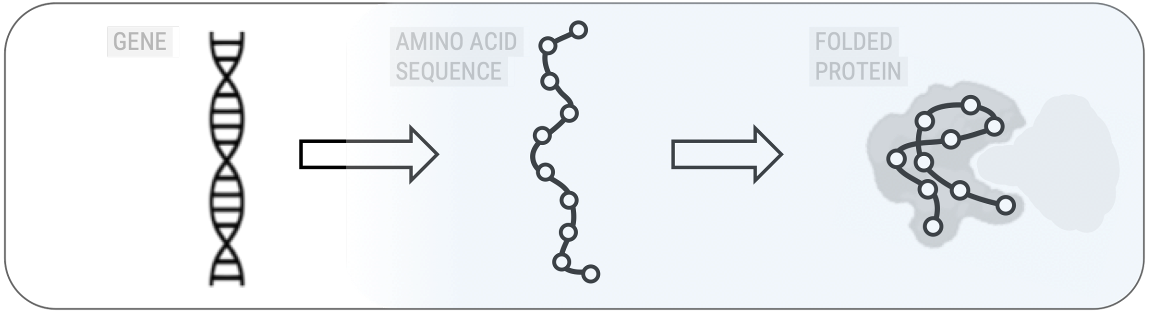
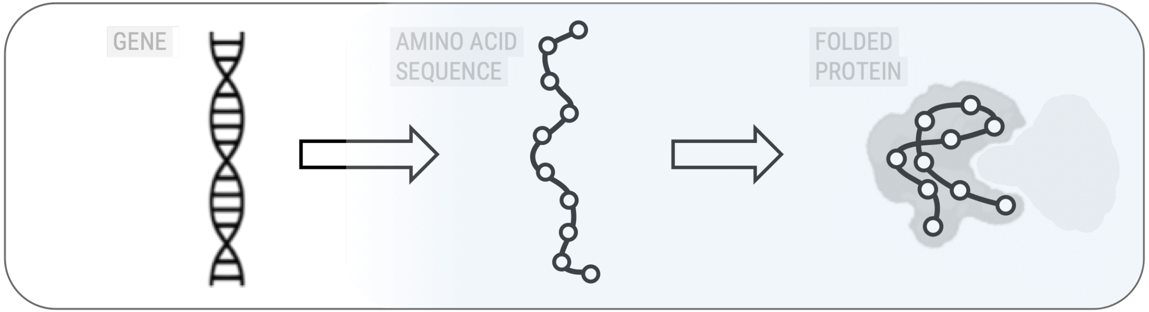
Introduction to Protein Design

11-06-2025

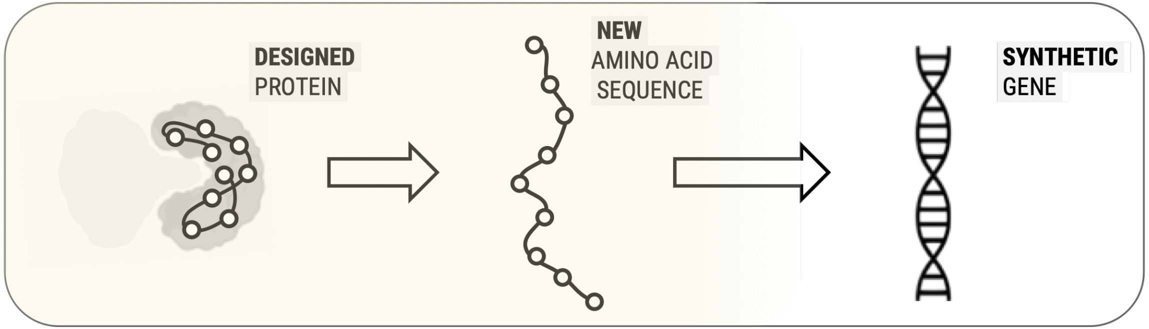
Rosetta was built for structure prediction...

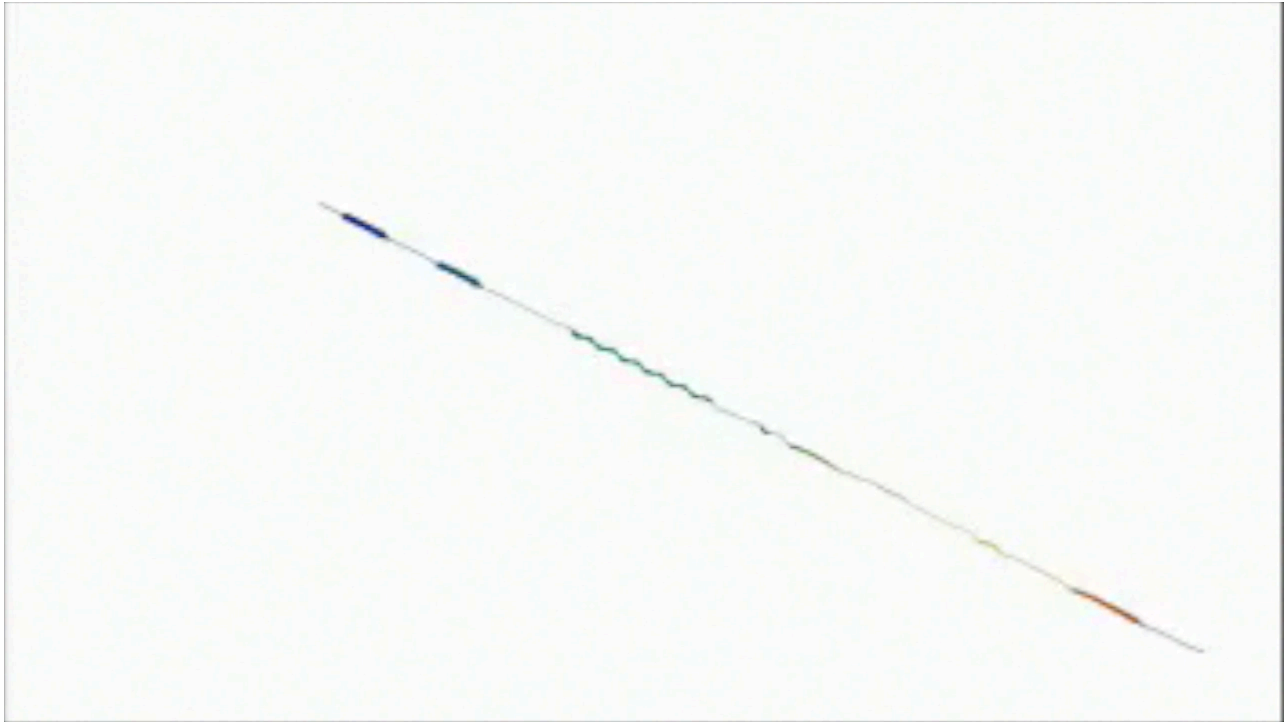


Rosetta was built for structure prediction...

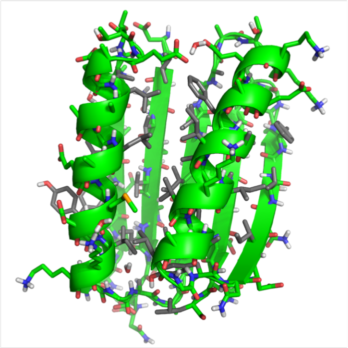


...but it can also be used for protein design

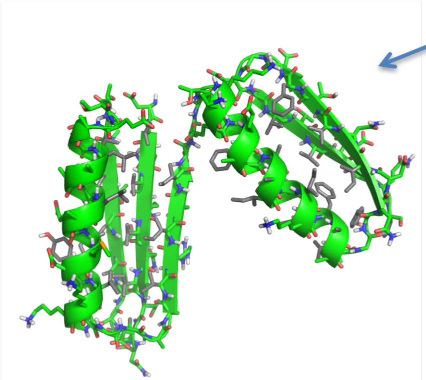




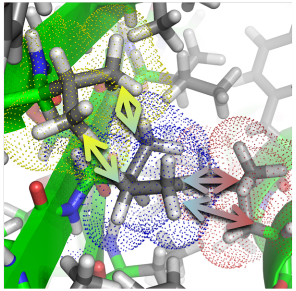
Broadly speaking, what does Rosetta *do*?



Keeps track of protein structure and kinematics



Provides algorithms for manipulating conformation and/or sequence



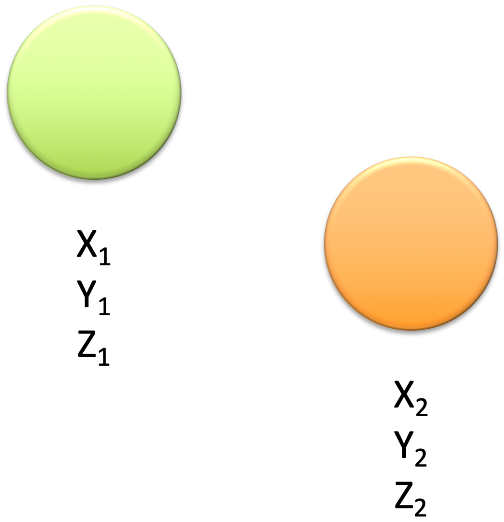
Scores favorability of current conformation and sequence

V. Mulligan

Basic Rosetta Terms

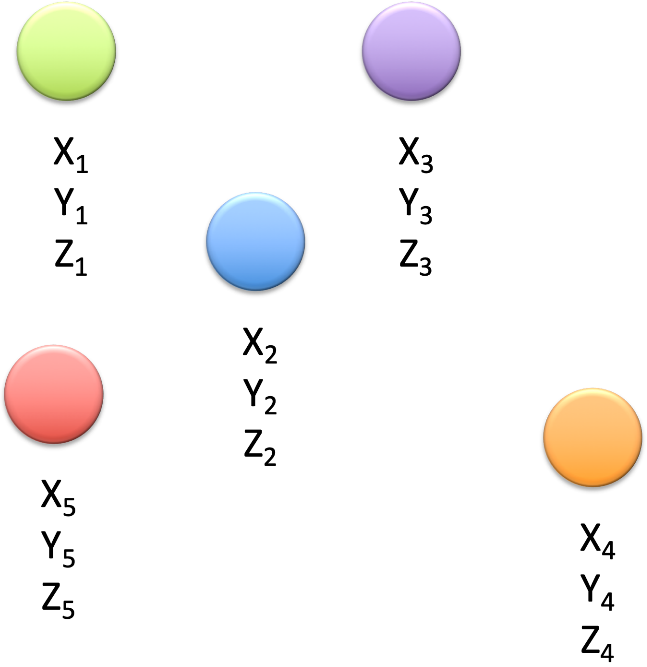
- Residue
- Mover
- Filter
- Pose
- Conformation Space

What is “conformation space”?



$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ X_2 \\ Y_2 \\ Z_2 \end{pmatrix}$$

What is “conformation space”?



$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ X_2 \\ Y_2 \\ Z_2 \\ X_3 \\ Y_3 \\ Z_3 \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

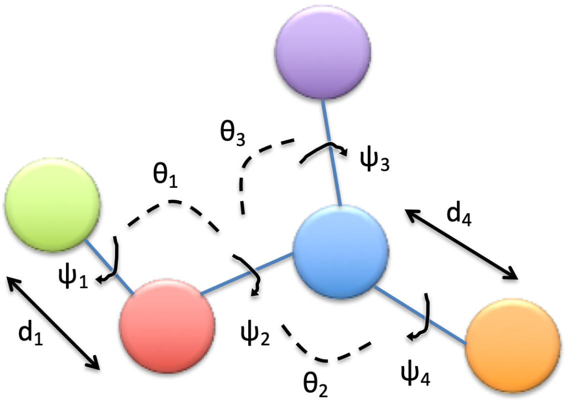
What is “conformation space”?

$$\vec{s}_1 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}, \quad \vec{s}_2 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

$$\left|\vec{s}_1 - \vec{s}_2\right| = \sqrt{\left(a_1 - b_1\right)^2 + \left(a_2 - b_2\right)^2 + \left(a_3 - b_3\right)^2 + \ldots}$$

Points in conformation space correspond to conformational states, and the distance between two points is a measure of how similar two states are. Given Cartesian coordinates, the length of the difference vector is the RMSD!

What is “conformation space”?

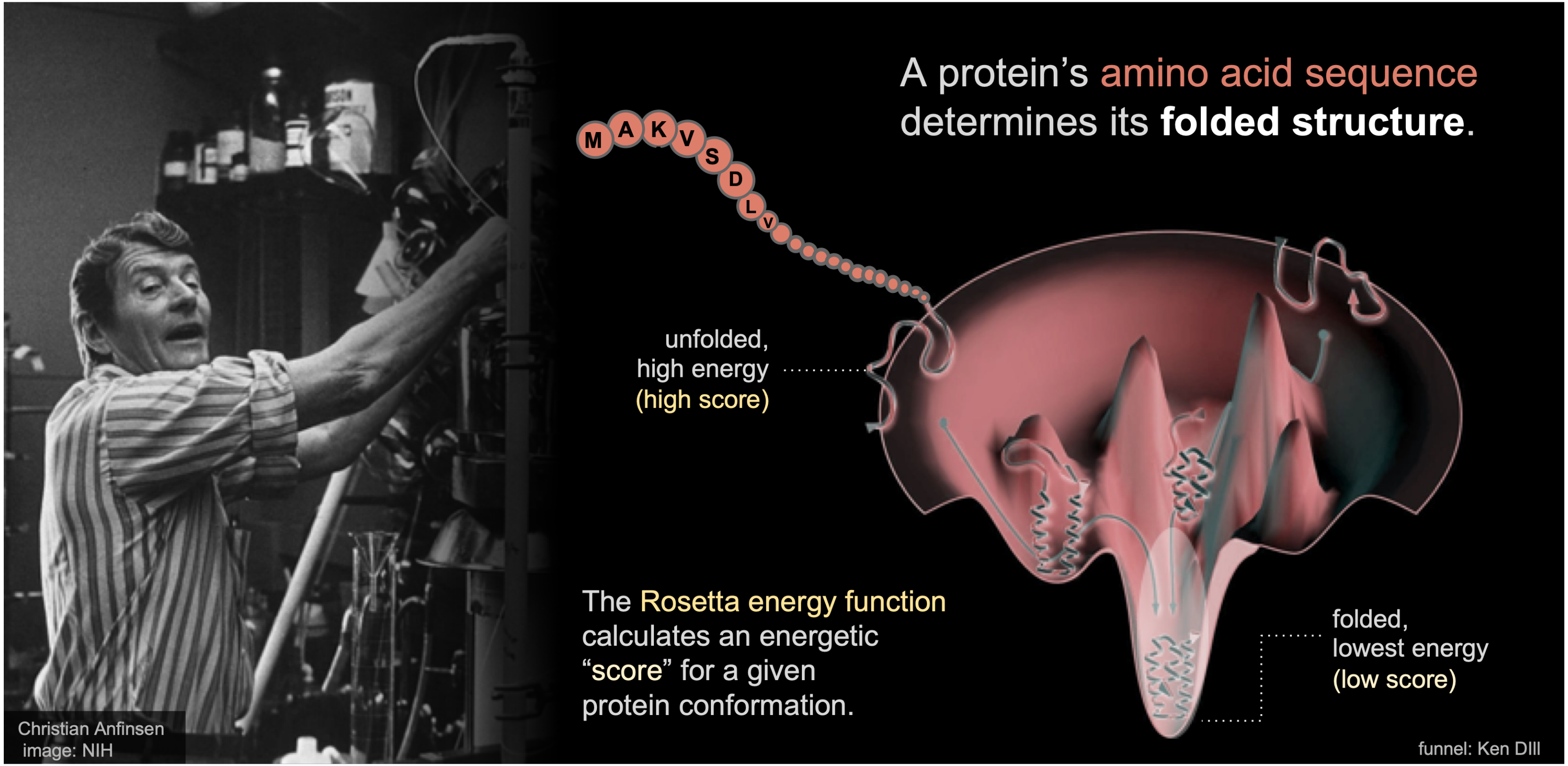


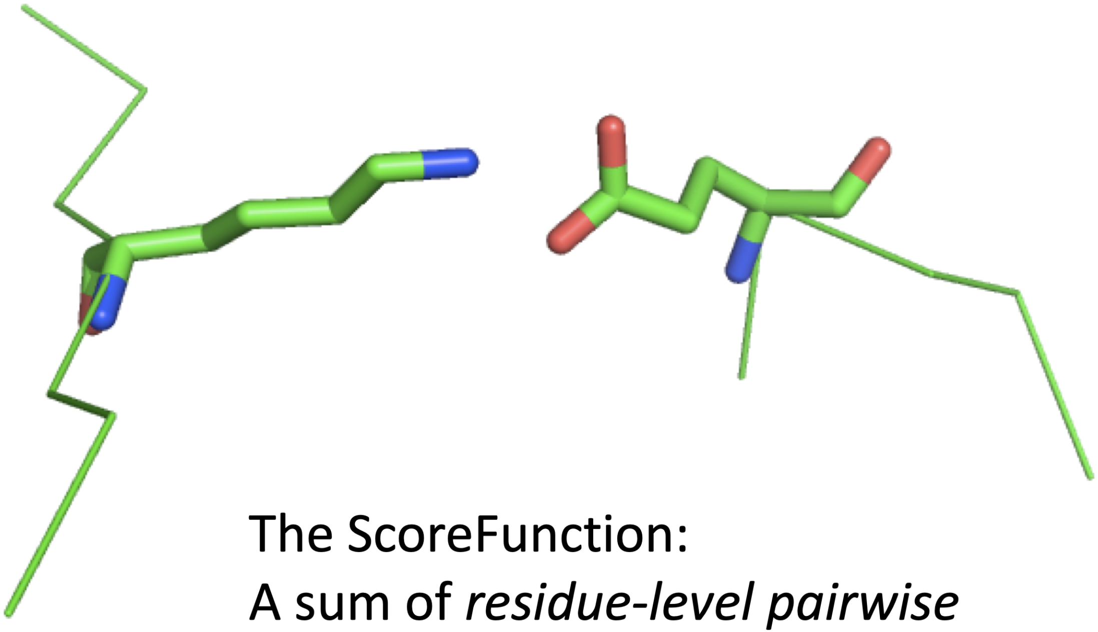
$$\begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \ldots \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \ldots \\ d_1 \\ d_2 \\ d_3 \\ \ldots \end{pmatrix}$$

Basic Rosetta Terms

- Residue
- Mover
- Filter
- Pose
- Conformation Space
- Score Function

Rosetta is tool for molecular modeling built on Anfinsen's Hypothesis

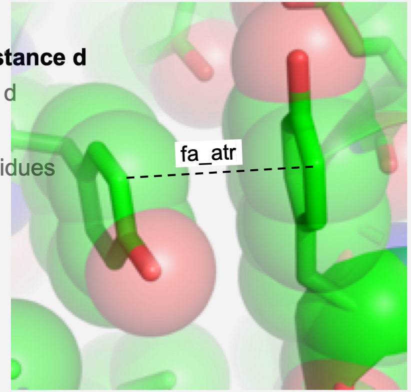


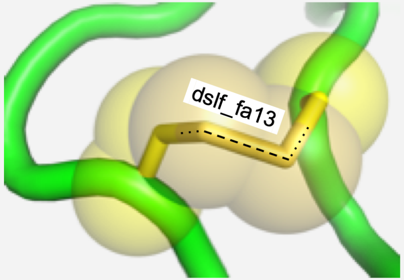


The ScoreFunction:
A sum of *residue-level pairwise decomposable* score terms.

Rosetta uses a knowledge-based, all-atom energy function

Term	Description
fa_atr	**attractive energy between two atoms on different residues separated by a distance d**
fa_rep	repulsive energy between two atoms on different residues separated by a distance d
fa_intra_rep	repulsive energy between two atoms on the same residue separated by a distance d
fa_sol	Gaussian exclusion implicit solvation energy between protein atoms in different residues
lk_ball_wtd	orientation-dependent solvation of polar atoms assuming ideal water geometry
fa_intra_sol	Gaussian exclusion implicit solvation energy between protein atoms in the same residue
fa_elec	energy of interaction between two nonbonded charged atoms separated by a distance d
hbond_lr_bb	energy of short-range hydrogen bonds
hbond_sr_bb	energy of long-range hydrogen bonds
hbond_bb_sc	energy of backbone–side-chain hydrogen bonds
hbond_sc	energy of side-chain–side-chain hydrogen bonds
dslf_fa13	**energy of disulfide bridges**
rama_prepro	probability of backbone ϕ, ψ angles given the amino acid type
p_aa_pp	probability of amino acid identity given backbone ϕ, ψ angles
fa_dun	probability that a chosen rotamer is native-like given backbone ϕ, ψ angles
pro_close	penalty for an open proline ring and proline ω bonding energy
yhh_planarity	sinusoidal penalty for nonplanar tyrosine χ3 dihedral angle
ref	reference energies for amino acid types
omega	backbone-dependent penalty for cis dihedrals that deviate from 0° and trans dihedrals that deviate from 180°





Alford *et al*, **The Rosetta all-atom energy function for macromolecular modeling and design.** *PNAS*. 2019.

RosettaScripts

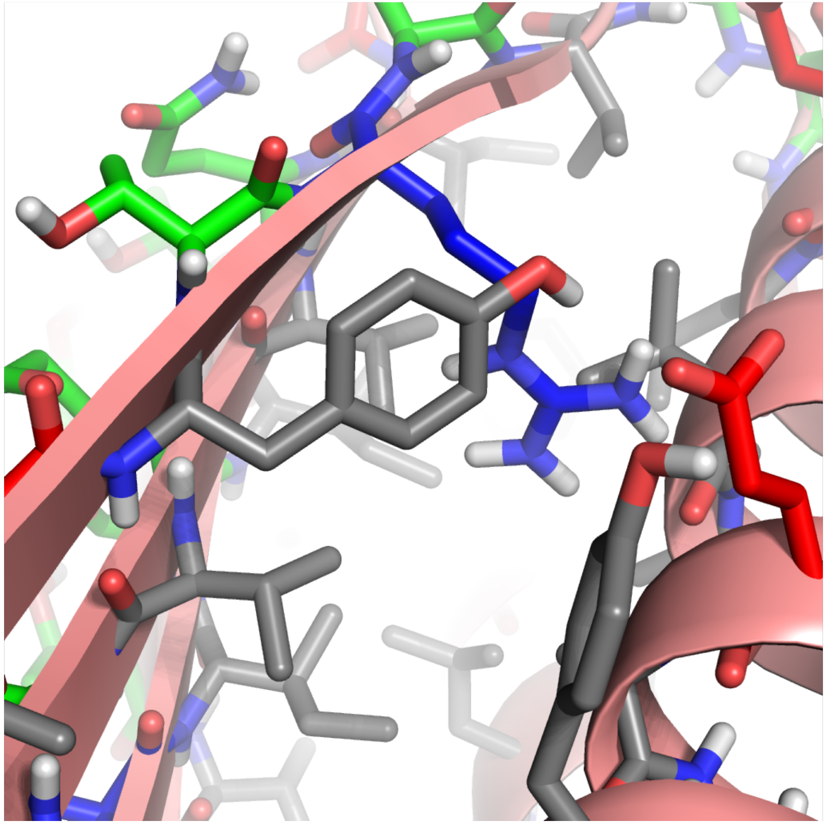
Layout of Rosetta Scripts

```
<ROSETTASCRIPTS>
    <SCOREFXNS>
    </SCOREFXNS>
    <RESIDUE_SELECTORS>
    </RESIDUE_SELECTORS>
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <FILTERS>
    </FILTERS>
    <MOVERS>
    </MOVERS>
    <PROTOCOLS>
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

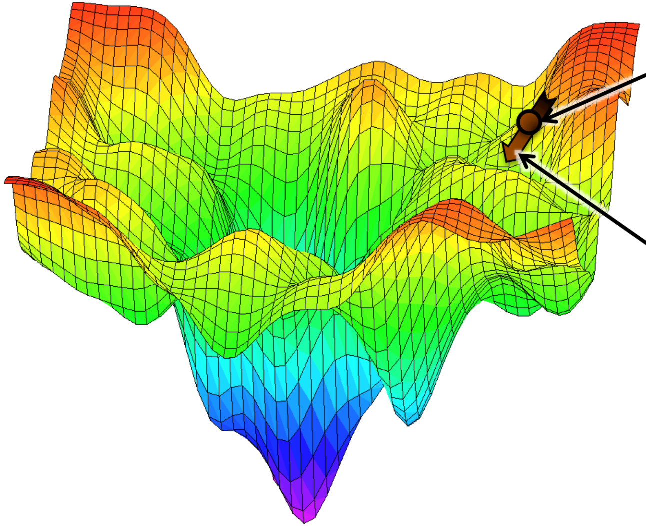
What is the “packer” and how does it work?

Rosetta B. Packer, Attorney

What is the “packer” and how does it work?



What is the "minimizer" and how does it work?



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

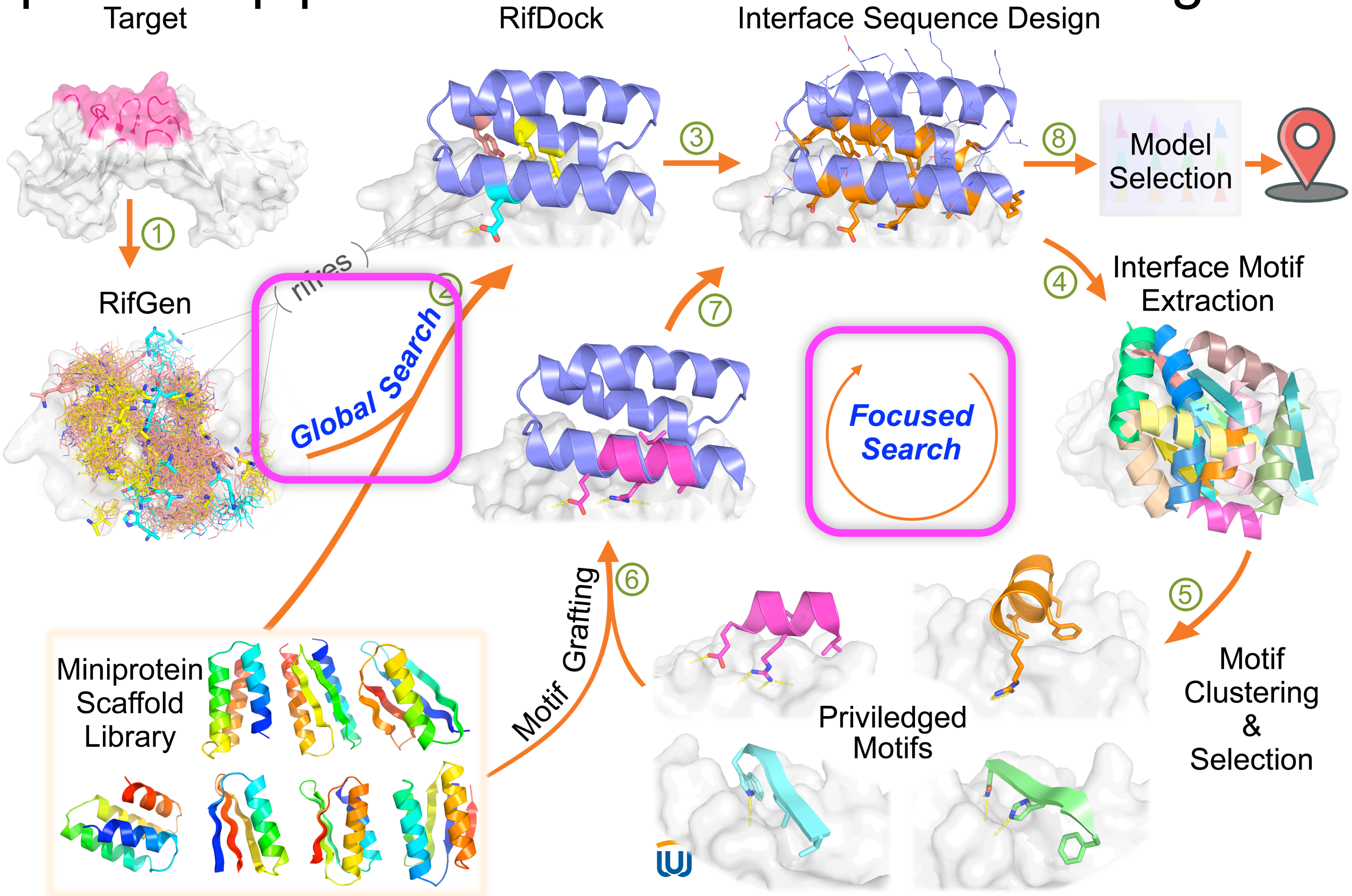
$$\begin{bmatrix} -\dfrac{\partial E}{\partial x_1} \\ -\dfrac{\partial E}{\partial x_2} \\ -\dfrac{\partial E}{\partial x_3} \\ \vdots \\ -\dfrac{\partial E}{\partial x_n} \end{bmatrix}$$

How are more complicated algorithms built?

FastRelax:

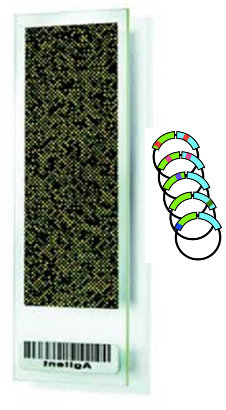
1. Set fa_rep to 2% of its normal value.
2. Call the packer.
3. Call the minimizer (low-stringency).
4. Set fa_rep to 25% of its normal value.
5. Call the packer.
6. Call the minimizer (low-stringency).
7. Set fa_rep to 55% of its normal value.
8. Call the packer.
9. Call the minimizer (low-stringency).
10. Set fa_rep to 100% of its normal value.
11. Call the packer, then the minimizer (high-stringency).
12. Call the minimizer (high-stringency).

Computation pipeline of the de novo binder design method

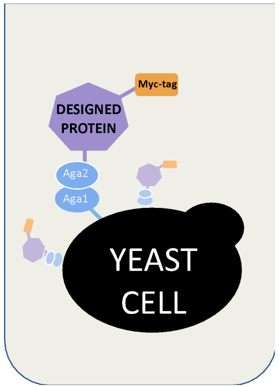


De novo Protein Design Experimental Pipeline

Gene Library Synthesis	Generate Yeast Surface Display Libraries	FACS and Next-Gen DNA Sequencing	Select Individual Designs for Verification



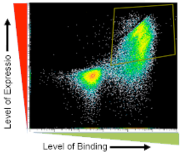
genes encoding the designed binder candidates

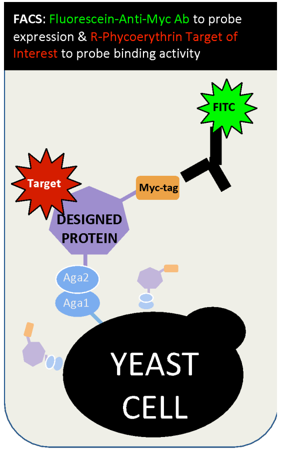


Limited Protease

Limited Heat

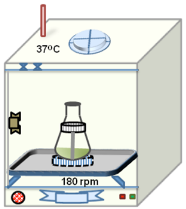
Transform yeast with plasmids encoding minibinder design library, and treat with limited protease and / or heat



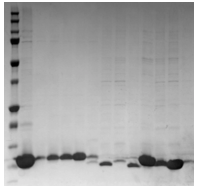


Identify gene sequences encoding functional designed minibinders

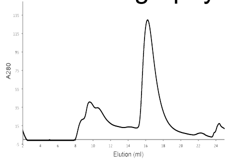
E. Coli expression



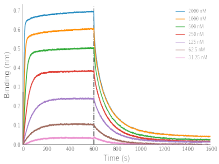
SDS-PAGE



Size-exclusion Chromatography



Octet binding



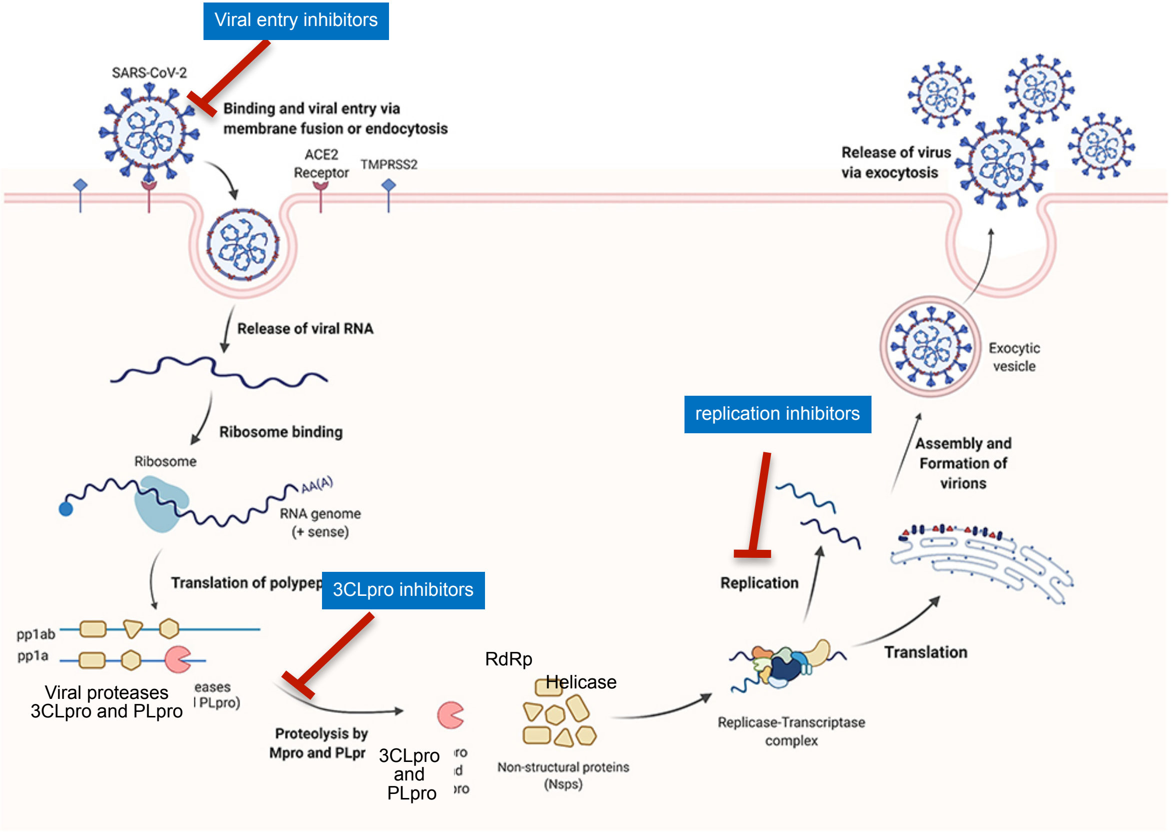
Individual clones expressing designed minibinders are used to verify function

Detailed pipeline of the de novo binder design method

1. **Target Selection** — Select the protein of interest and pick the targeting region
2. **RifGen** — Run rifgen to generate billions of favorable interacting RIF residues
3. **Patchdock** — Dock the miniprotein scaffolds to the selected region by low resolution shape matching
4. **RifDock** — Fine tune the rigid body orientation of Patchdock outputs using the grid-based refinement in RIF
5. **Predictor** — Pre-screen the RifDock outputs using a super fast Rosetta design protocol and ML based method
6. **Interface Design** — Use Rosetta to optimize the interface residues for high shape and chemical complementarity
7. **Motif Extraction** — Extract all the secondary structural motifs making good contacts with the target protein
8. **Motif Clustering** — Cluster the motifs and select privileged motifs based on per-position weighted binding energy
9. **Model Grafting** — Superimpose the scaffolds onto the privileged motifs
10. **Predictor** — Pre-screen the grafting outputs using a super fast Rosetta design protocol and ML based method
11. **Interface Design** — Use Rosetta to optimize the interface residues for high shape and chemical complementarity
12. **Model Selection** — Select good designs with favorable protein folding and protein interface metrics
13. **Oligo Array Synthesis** — Synthesis the oligonucleotides encoding the designs using the DNA printing technology
14. **Yeast Display Screening** — Select and enrich the binders through the high-throughput yeast display and cell sorting assay
15. **NextGen Sequencing** — Deep sequence the start and the selected yeast sorting pools to identify potential binders
16. **Binder Characterization** — Quickly check the expression in *E. coli*, solubility and binding of the potential binders
17. **SSM Library** — Substitute each residue with the other 19 AA to generate site saturation mutagenesis library
18. **Yeast Display Screening / NextGen Sequencing** — Screen the SSM library and identify beneficial mutations through NextGen Sequencing
19. **Combo Library** — Combine 5-15 beneficial mutations to generate the combinatorial library using degenerative codons
20. **Yeast Display Screening / Sanger Sequencing** — Screen the combo library using lower target concentration to identify the highest affinity binder
21. **Binder Characterization** — Thoroughly check the expression, solubility, stability, oligomerization and binding affinity of the optimize binder
22. **Structure Validation** — Obtain the high-resolution structure of the *de novo* binder to valid the design model
23. **Function Validation** — Characterize the potential functions of the *de novo* designed minibinder
24. **Protocol Optimization** — Learn the successes and failures of the whole process to optimize the computation pipeline

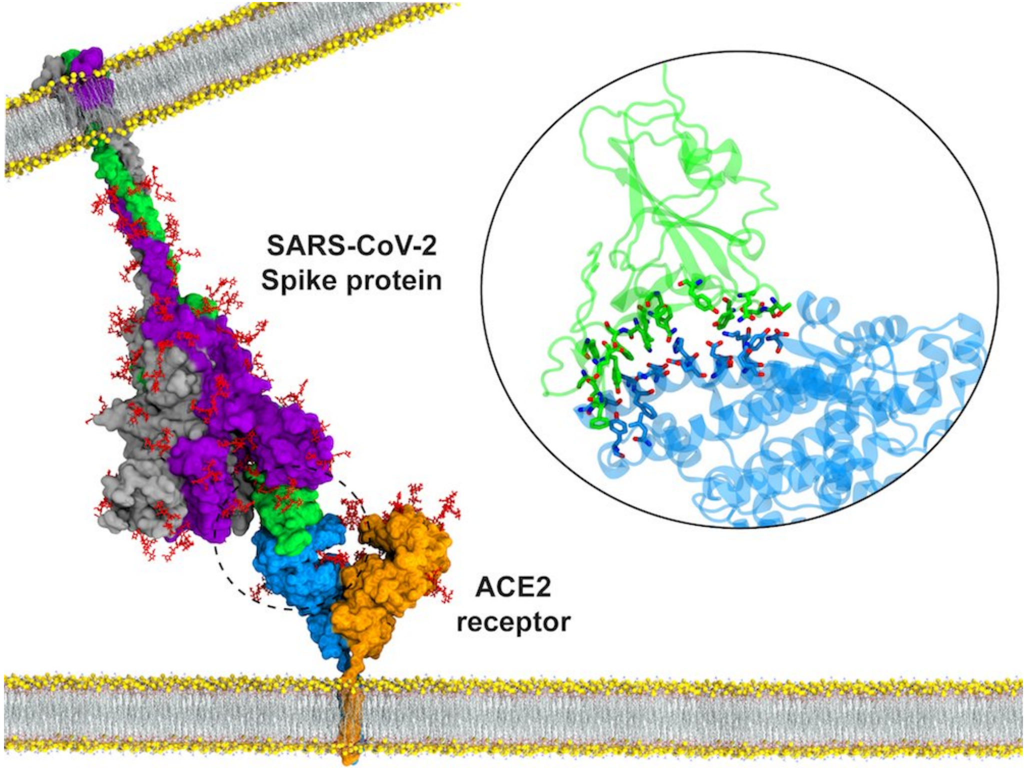
Slides Credit: Longxing Cao

Fighting SARS-CoV-2

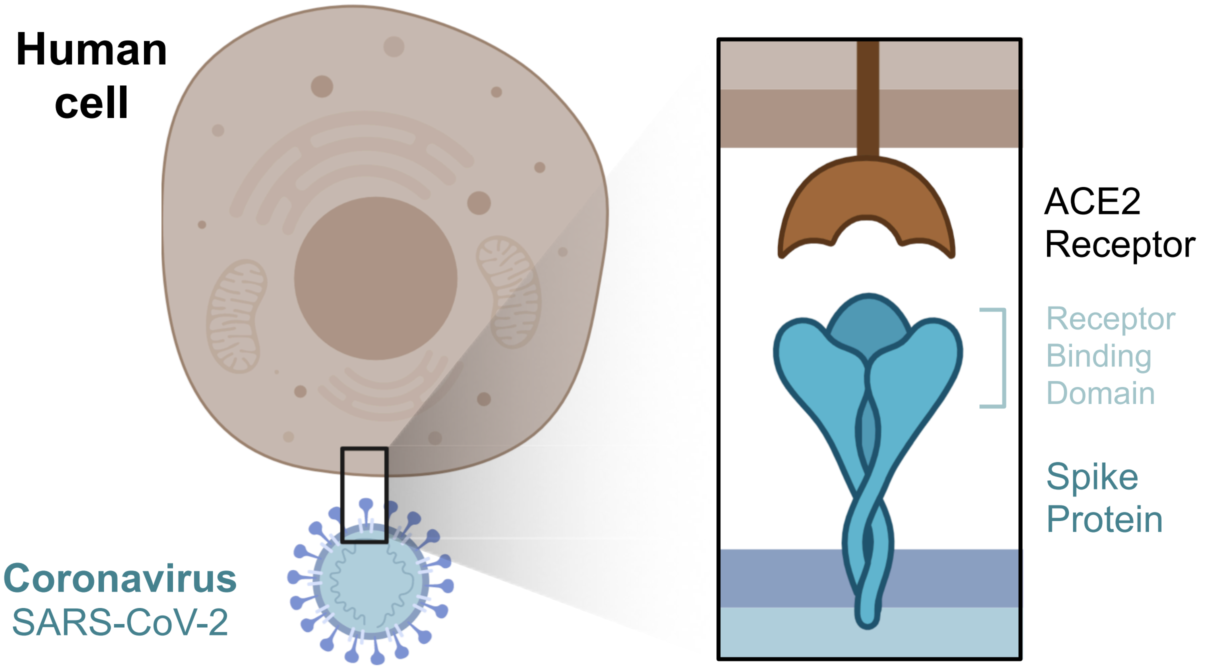


Alzyoud L, Ghattas MA, Atatreh N. Drug Des Devel Ther. 2022;16:2463-2478

Interaction between ACE2 receptor and SARS-CoV-2 spike protein



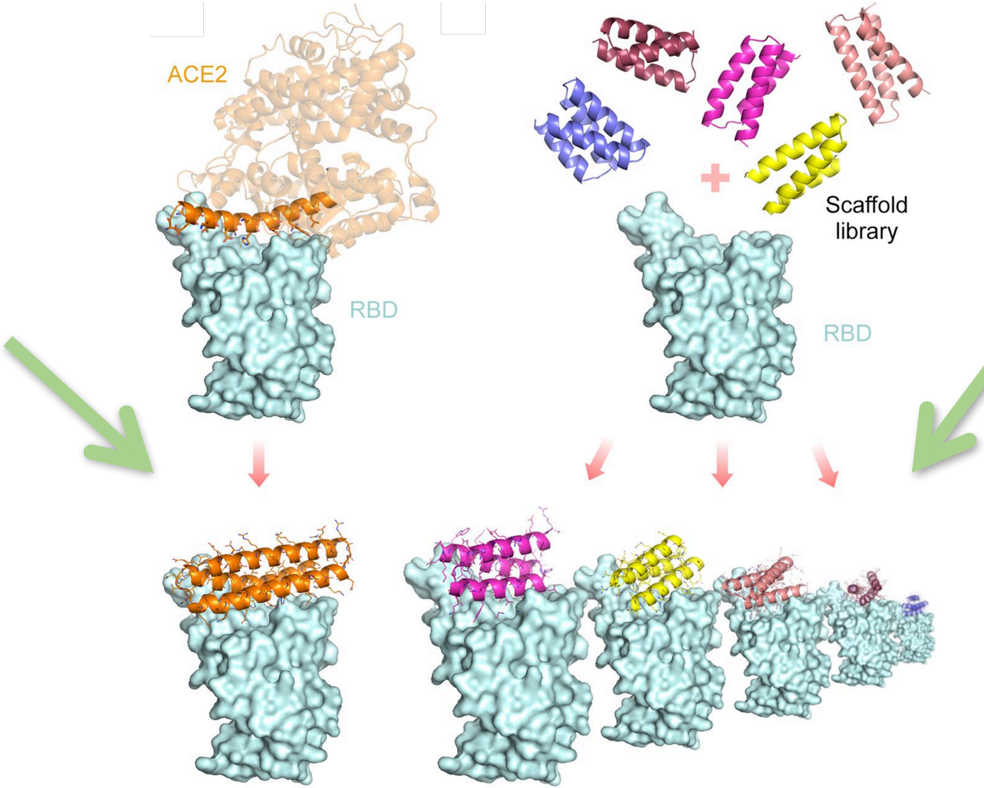
De novo design against COVID-19 -- design small proteins that disrupt viral infection



Computational Design of SARS-CoV-2 Miniprotein Inhibitors

Approach I

de novo proteins built from the ACE2 helix

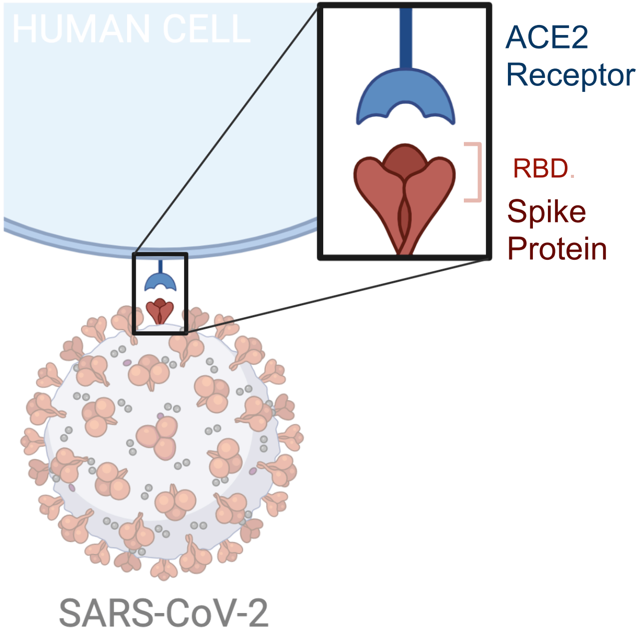


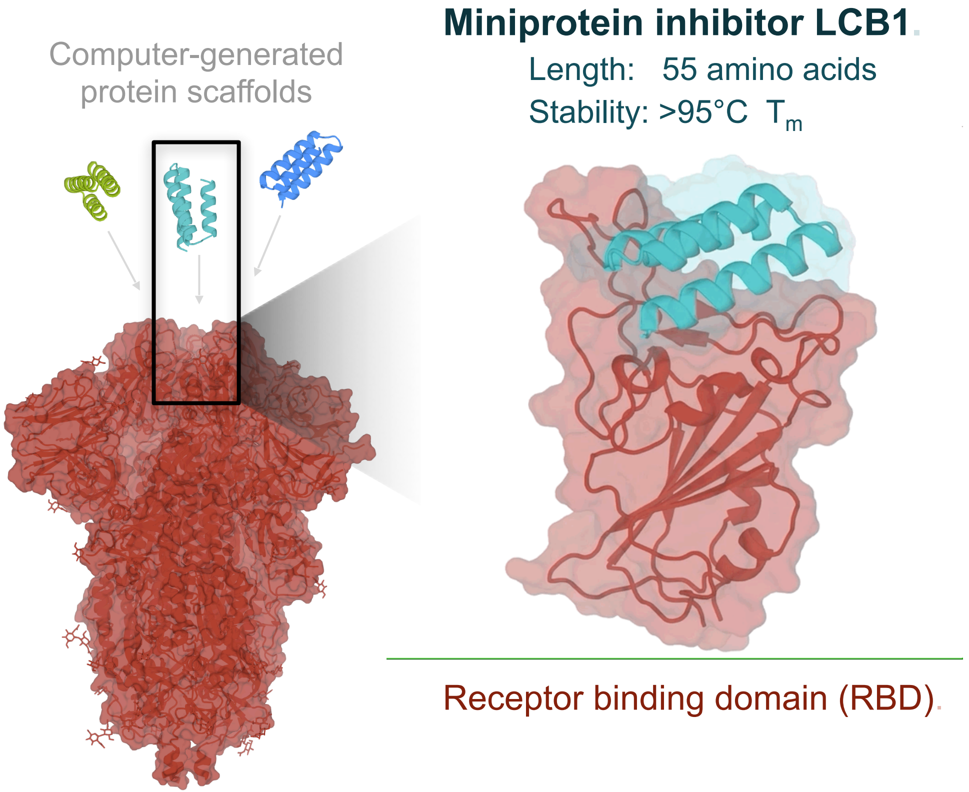
Approach II

de novo scaffolds docked to the ACE2 binding region

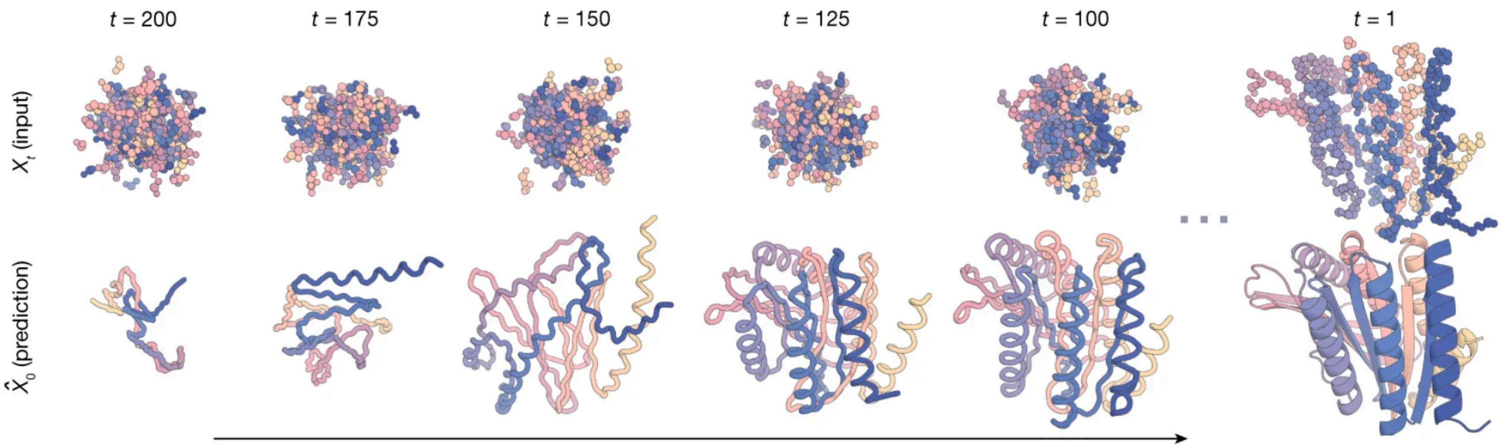
Cao *et al.*, *Science*, 2020.

Goal: Design small proteins that disrupt viral infection



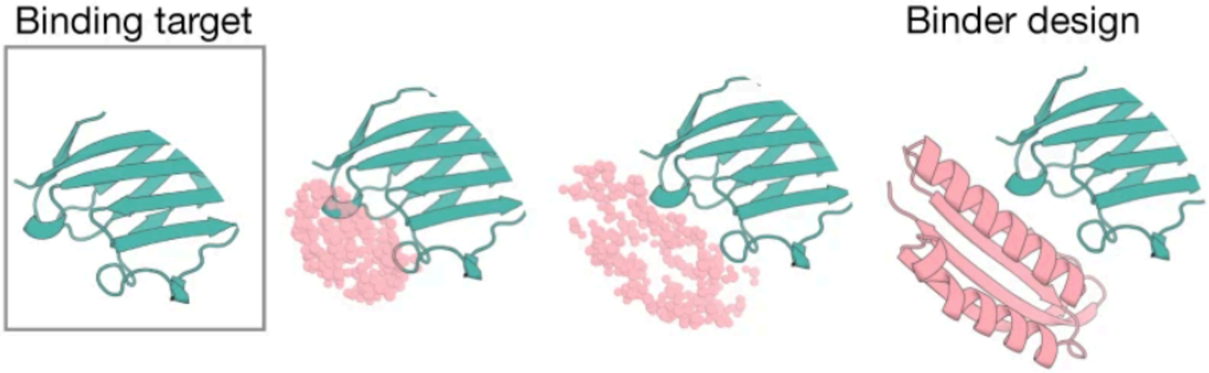


De novo binder design with RFdiffusion



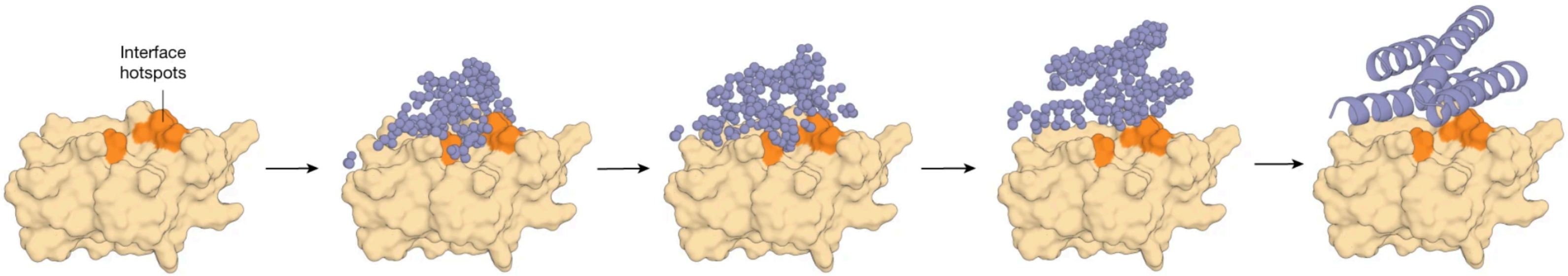
Watson, J., *Nature, 2023.*

De novo binder design with RFdiffusion



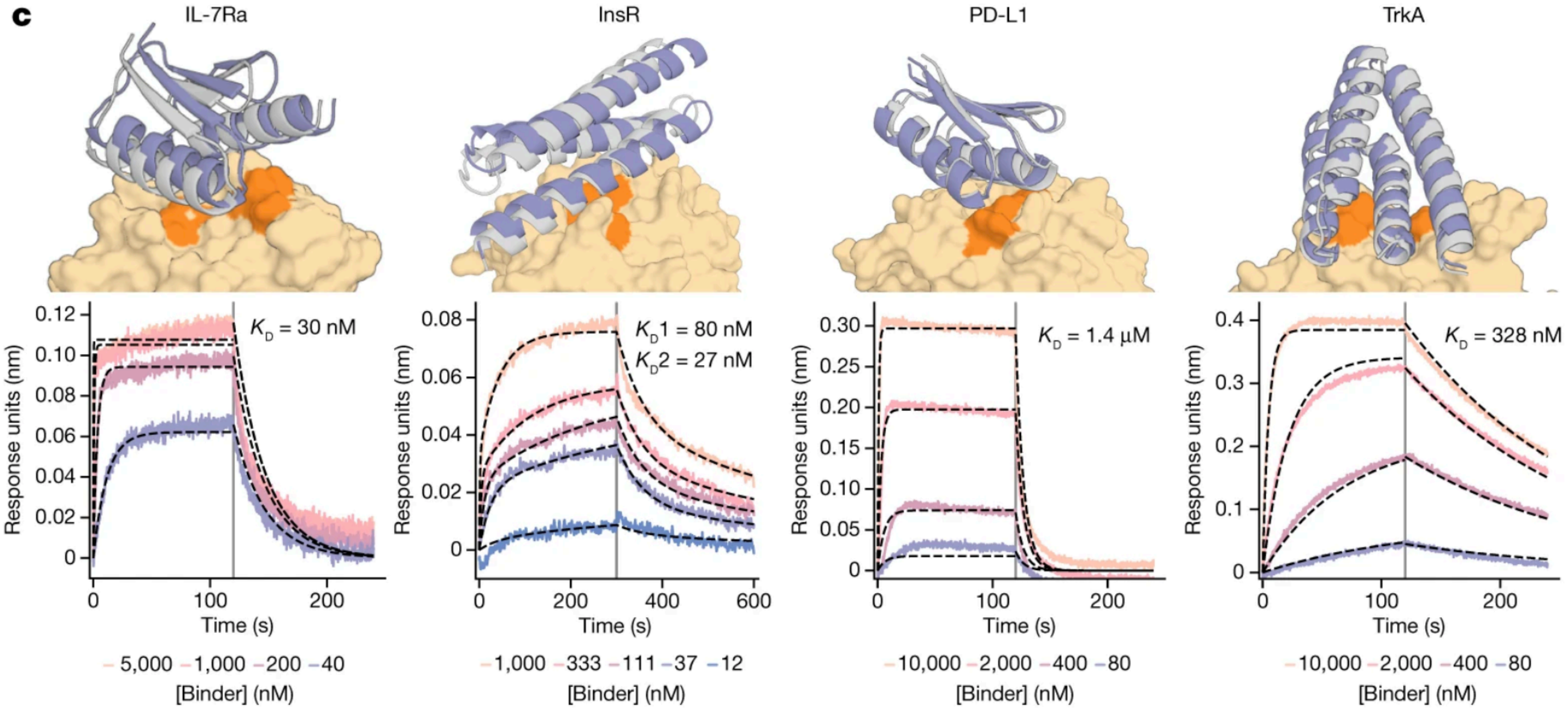
Watson, J., *Nature, 2023.*

De novo binder design with RFdiffusion



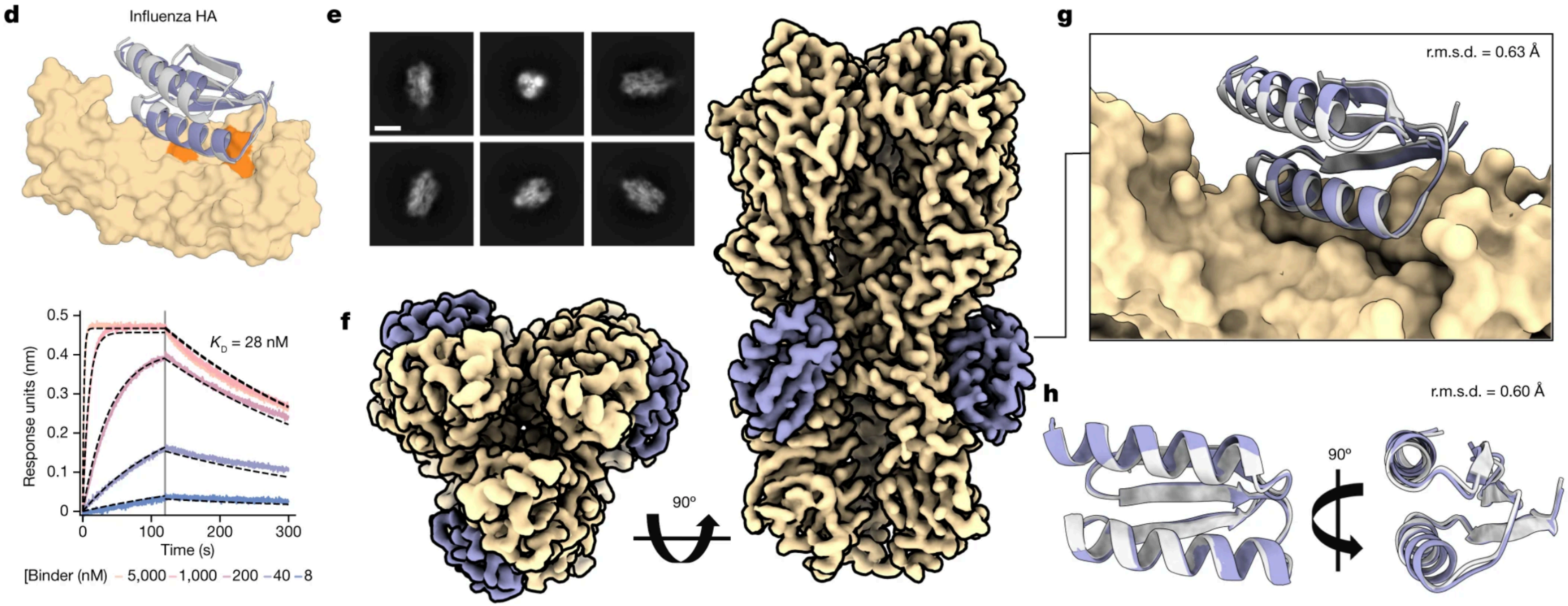
Watson, J., *Nature, 2023.*

De novo binder design with RFdiffusion



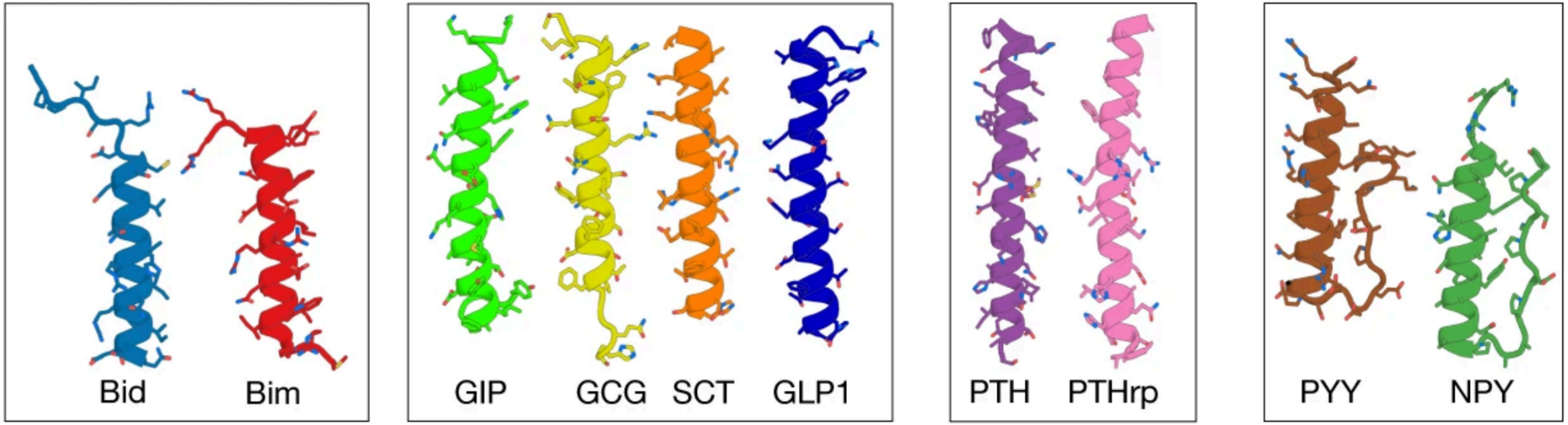
Watson, J., *Nature, 2023.*

De novo binder design with RFdiffusion



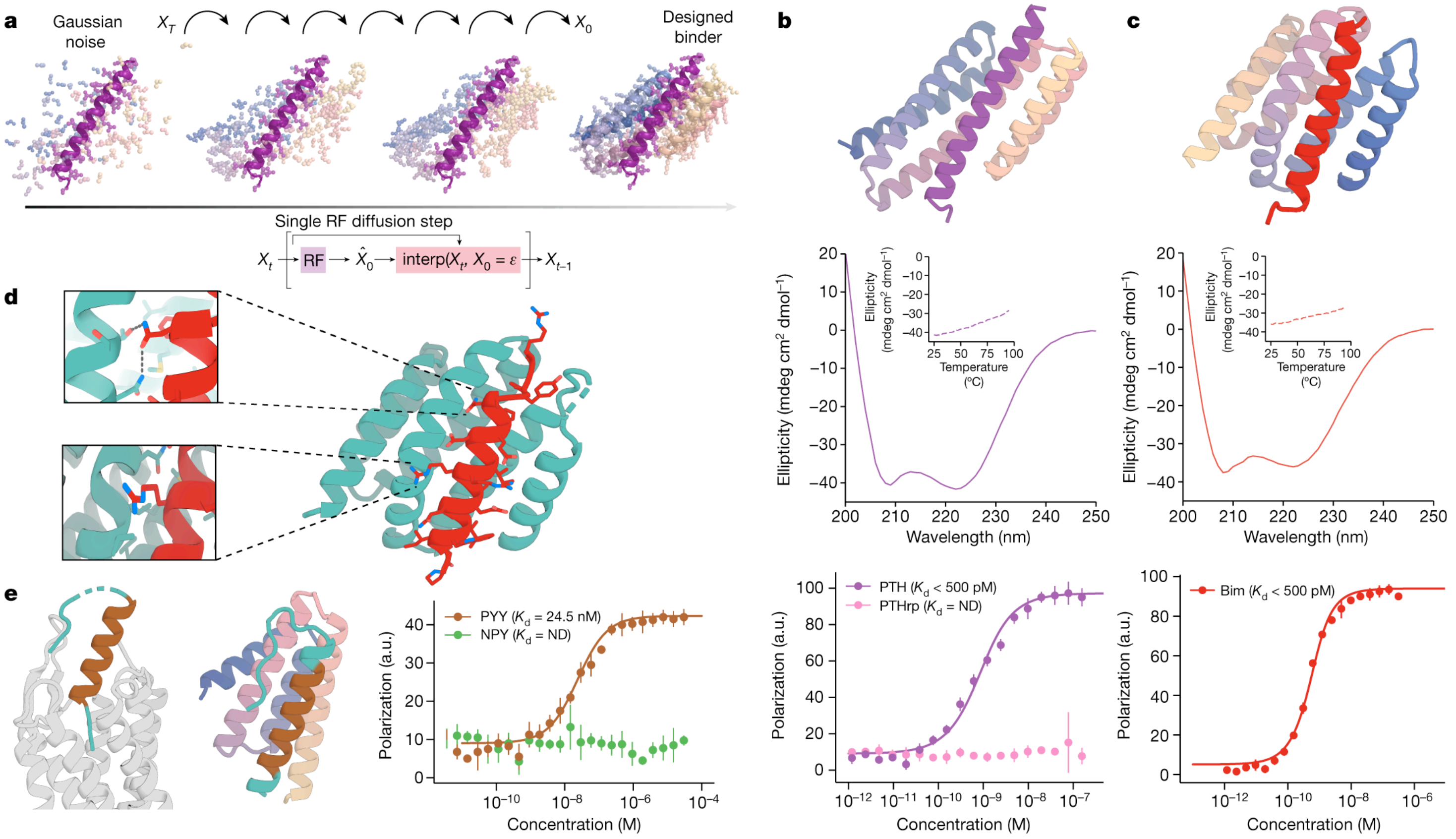
Watson, J., *Nature, 2023.*

De novo design of high-affinity binders of bioactive helical peptides



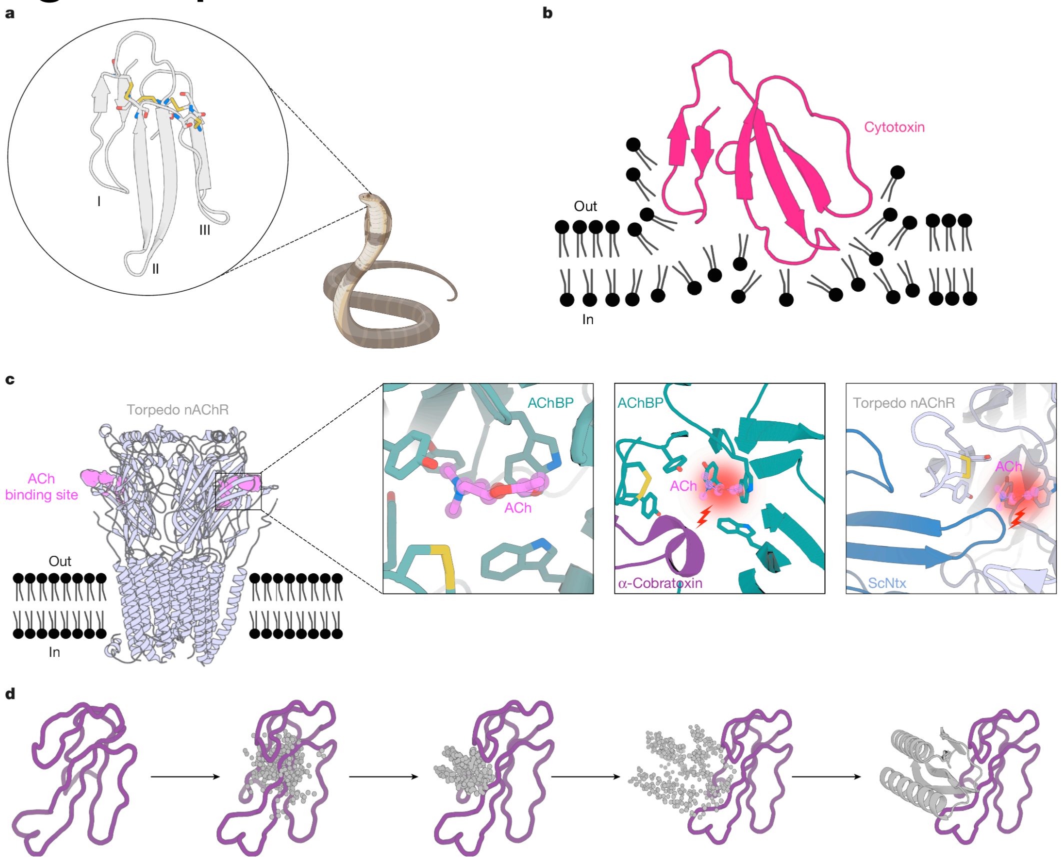
Torres, S., *Nature, 2023.*

De novo peptide binder design with RFdiffusion



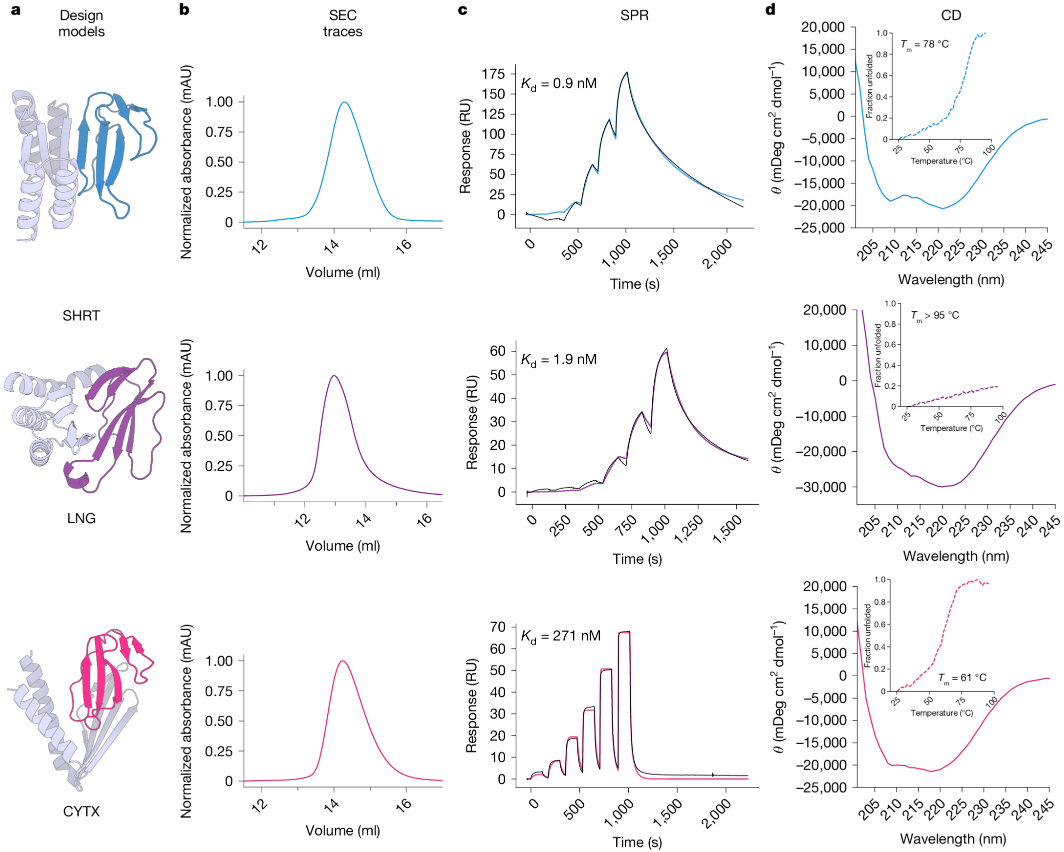
Torres, S., *Nature, 2023.*

De novo designed proteins neutralize lethal snake venom toxins



Susana, T., *Nature, 2025.*

De novo designed proteins neutralize lethal snake venom toxins



Susana, T., *Nature, 2025.*

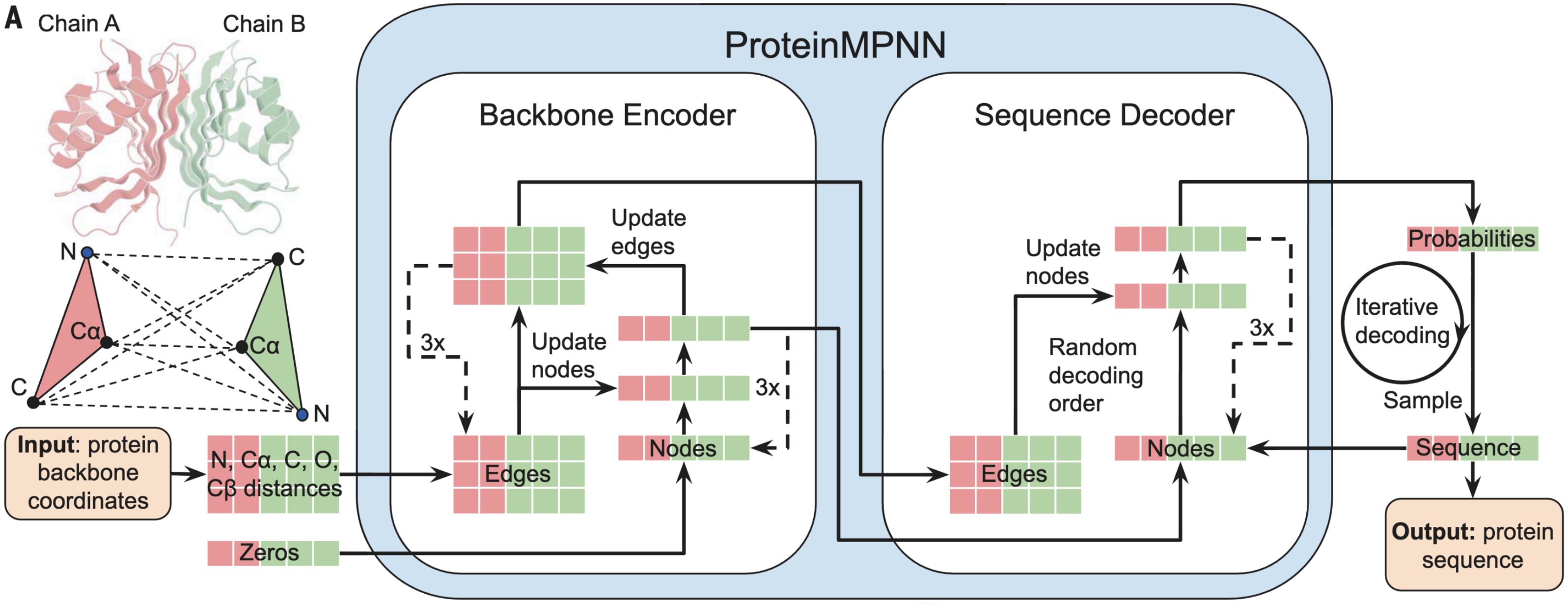
RESEARCH

PROTEIN DESIGN

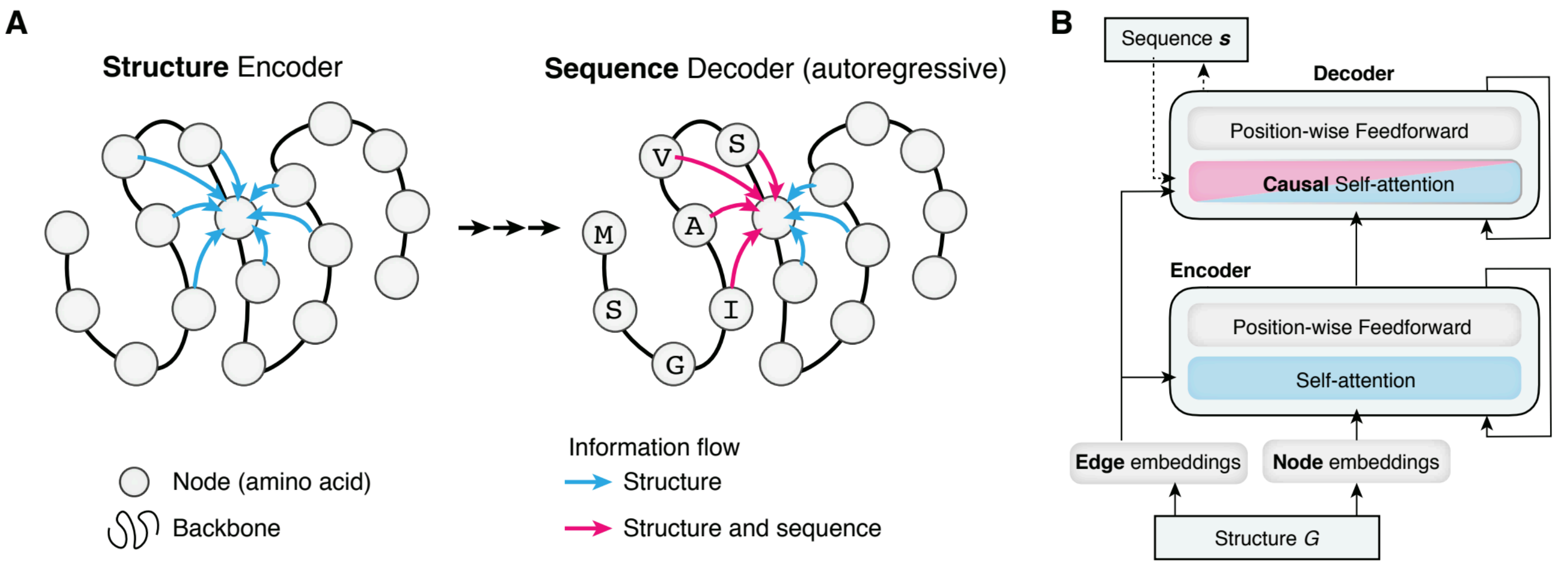
Robust deep learning–based protein sequence design using ProteinMPNN

J. Dauparas[1,2], I. Anishchenko[1,2], N. Bennett[1,2,3], H. Bai[1,2,4], R. J. Ragotte[1,2], L. F. Milles[1,2], B. I. M. Wicky[1,2], A. Courbet[1,2,4], R. J. de Haas[5], N. Bethel[1,2,4], P. J. Y. Leung[1,2,3], T. F. Huddy[1,2], S. Pellock[1,2], D. Tischer[1,2], F. Chan[1,2], B. Koepnick[1,2], H. Nguyen[1,2], A. Kang[1,2], B. Sankaran[6], A. K. Bera[1,2], N. P. King[1,2], D. Baker[1,2,4]*

ProteinMPNN architecture

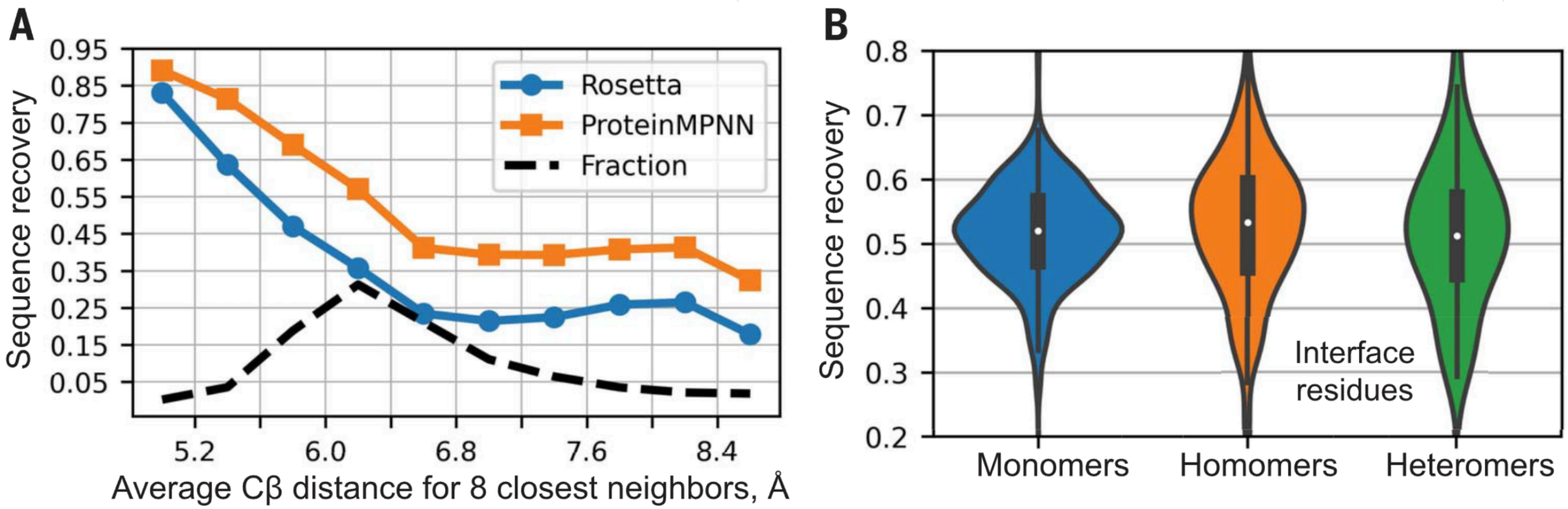


A graph-based, autoregressive model for protein sequences given 3D structures

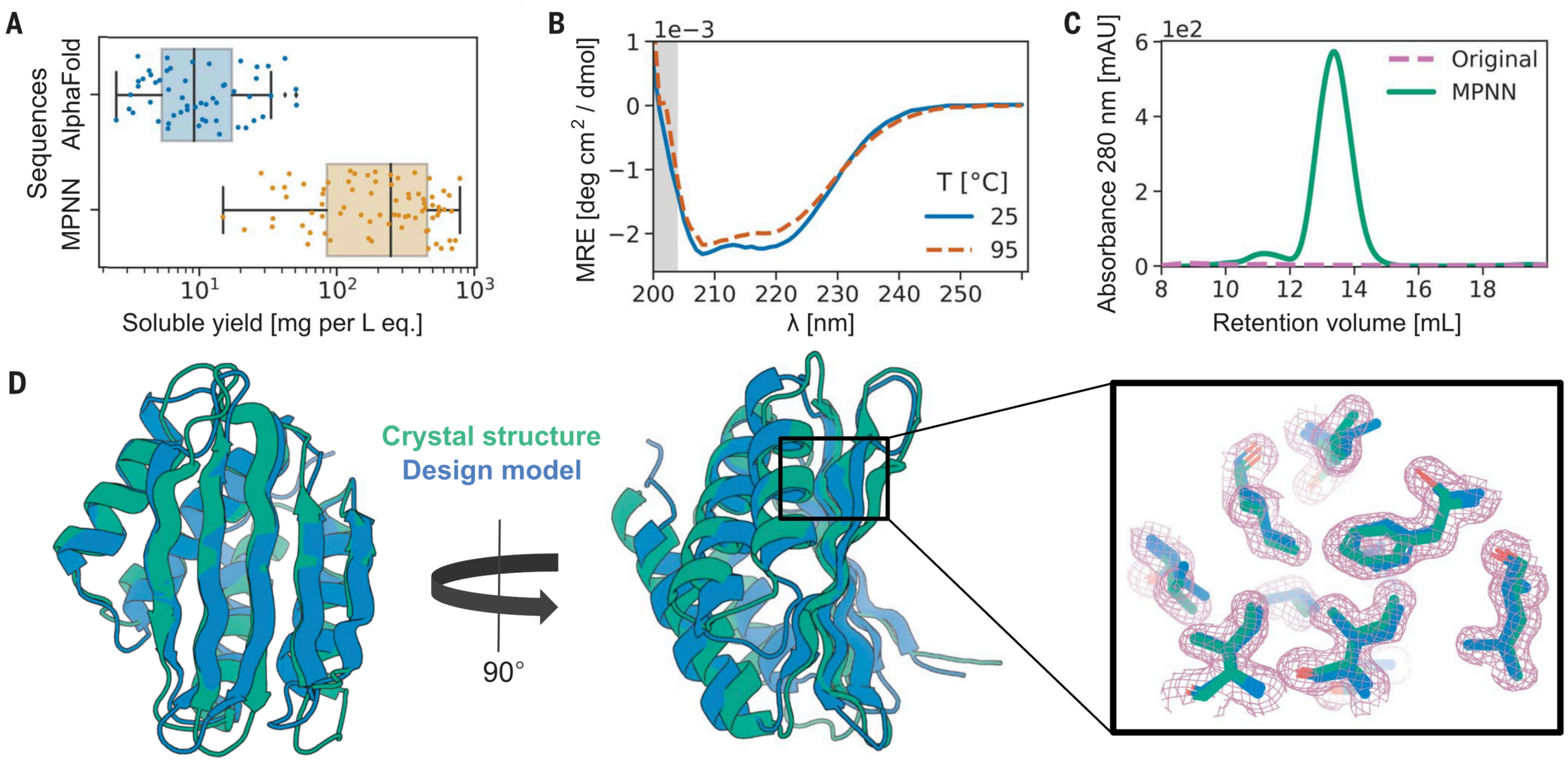


NIPS, 2019.

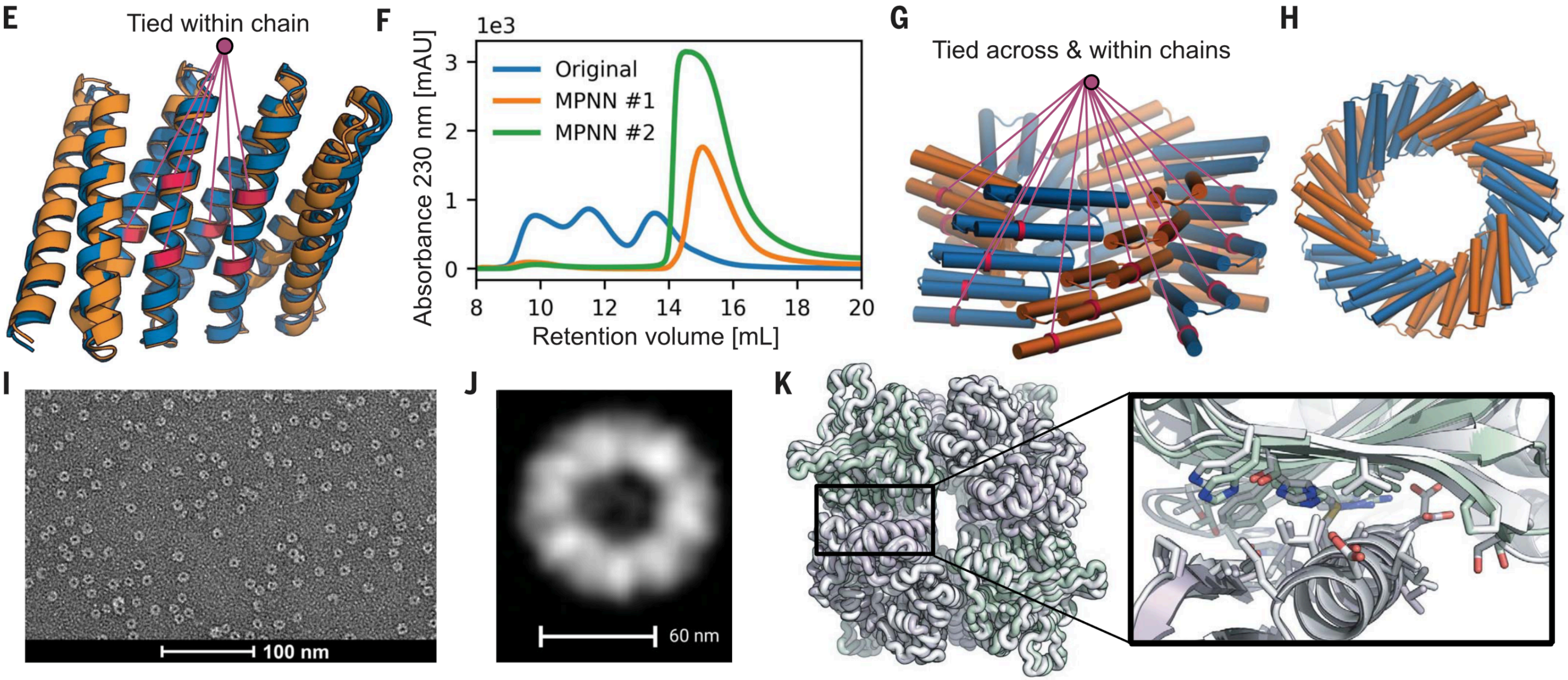
In silico evaluation of ProteinMPNN



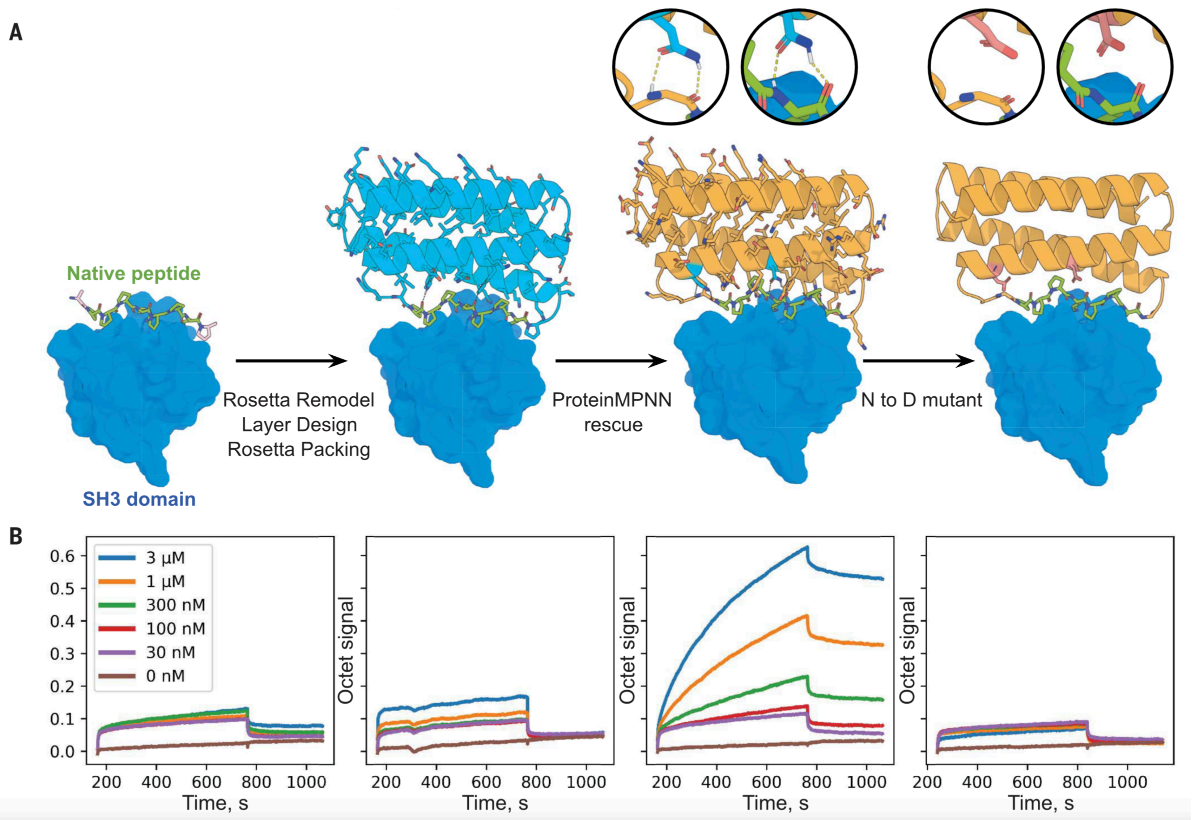
Structural characterization of ProteinMPNN designs



Structural characterization of ProteinMPNN designs



Design of protein function with ProteinMPNN



Highly accurate protein structure prediction with AlphaFold

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

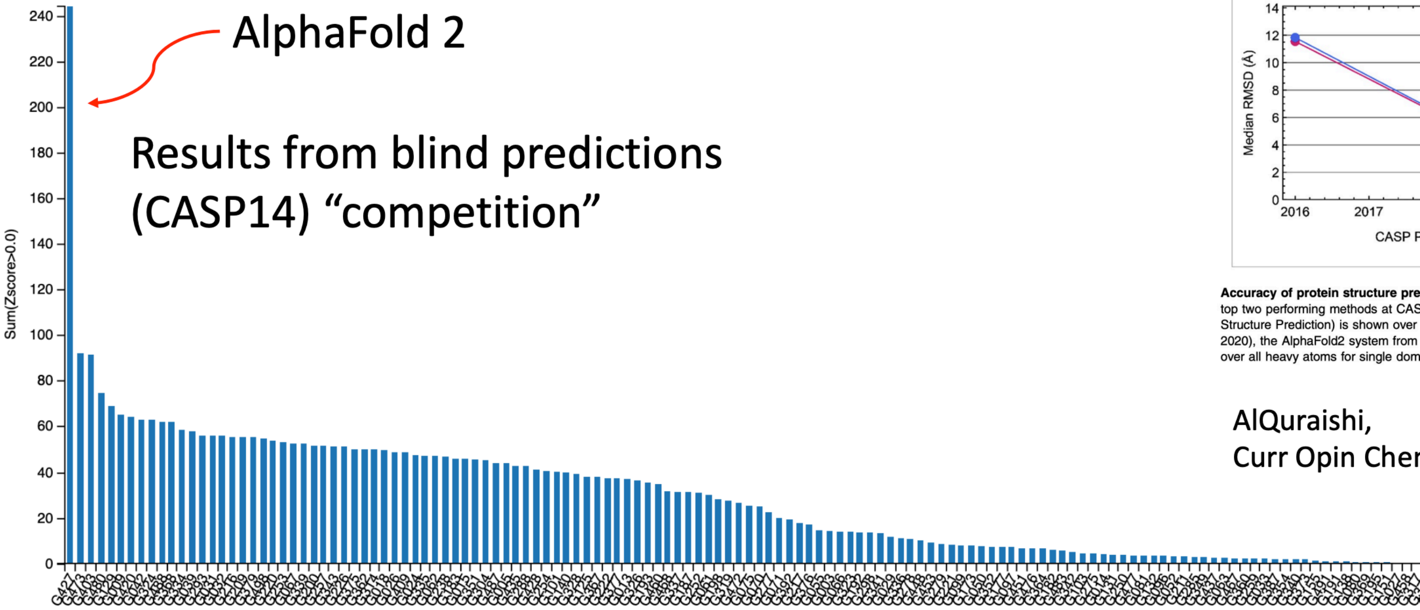
nature › articles › article

Article | Open access | Published: 15 July 2021

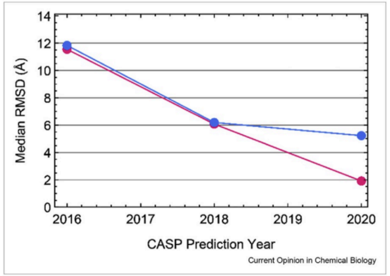
Highly accurate protein structure prediction with AlphaFold

John Jumper ✉, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli & Demis Hassabis ✉

AlphaFold2 performance in CASP14



Results from blind predictions (CASP14) “competition”



Accuracy of protein structure prediction. The median accuracy of the top two performing methods at CASP (Critical Assessment of protein Structure Prediction) is shown over the last four years. In CASP14 (late 2020), the AlphaFold2 system from DeepMind achieved ~2 Å accuracy over all heavy atoms for single domain apo proteins.

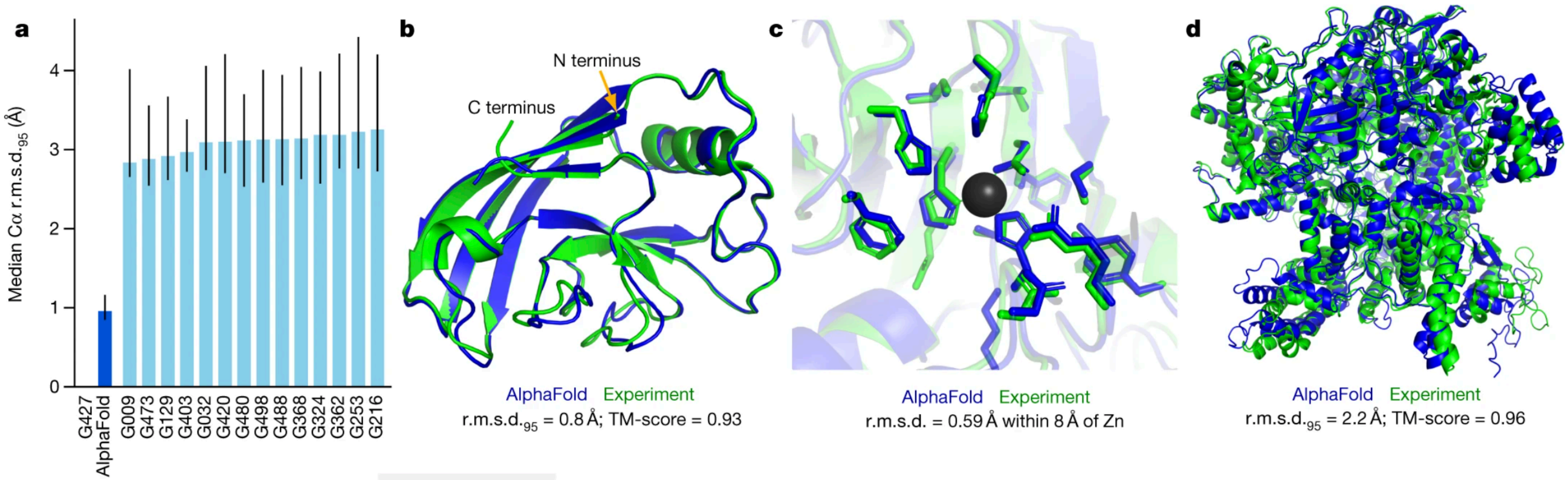
AlQuraishi,
Curr Opin Chem Biol, 2021

What is AlphaFold?

- A machine-learning-based model for predicting the 3D structure of proteins using only sequence as input.

- Trained on known sequences and structures from the Protein Data Bank, as well as large databases of protein sequences.

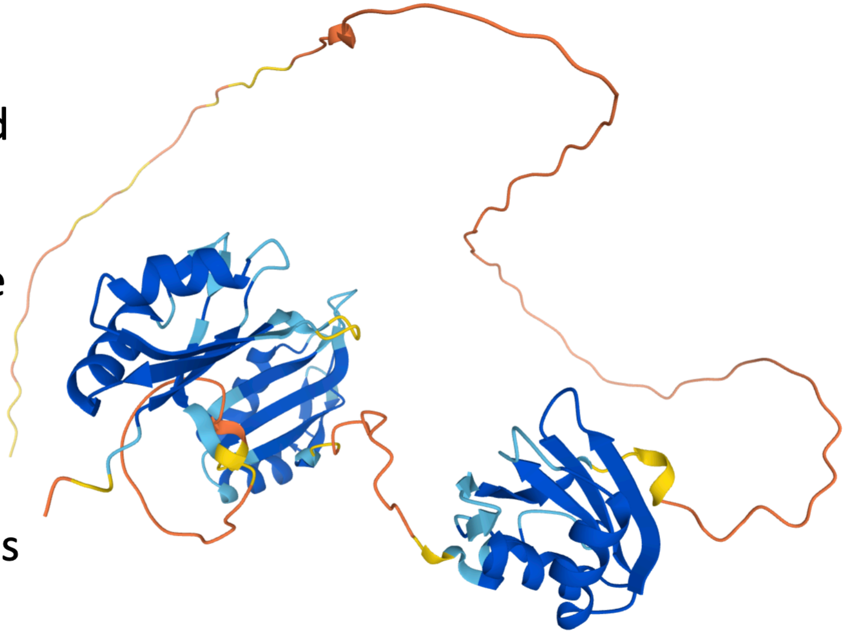
AlphaFold2 produces highly accurate structures



Example (TIA1)

Example (TIA1)

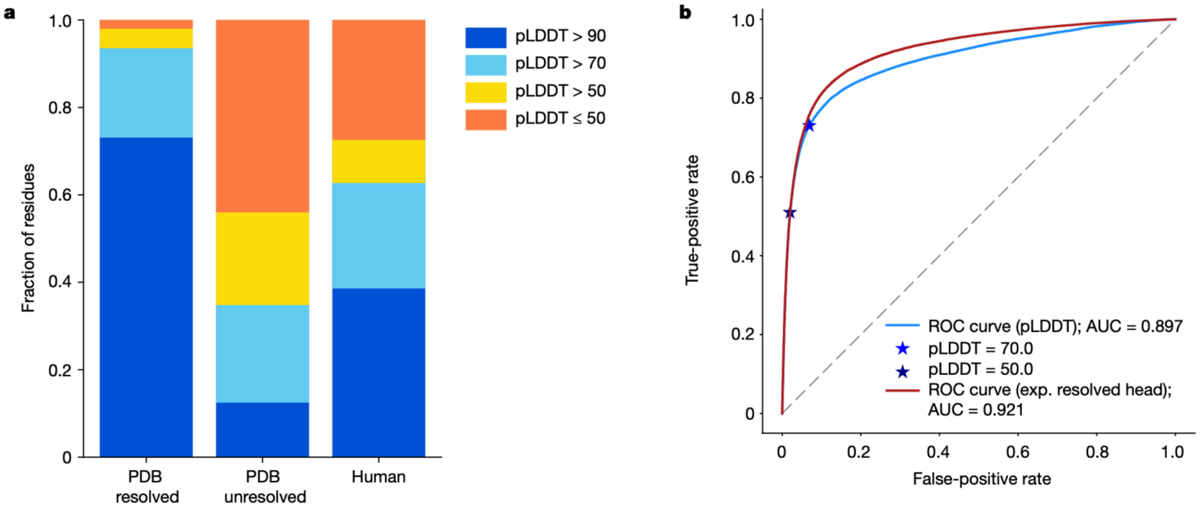
- AF2 was trained on monomeric proteins with structures resolved in the PDB
- It is not designed to predict flexibility or structures of flexible regions
- AF2, however, is pretty good at telling you when you should not trust the predictions
- When AF2 is unsure, the region is likely disordered*



* Tunyasuvunakool et al, Nature, 2021
* Akdel et al, bioRxiv, 2021

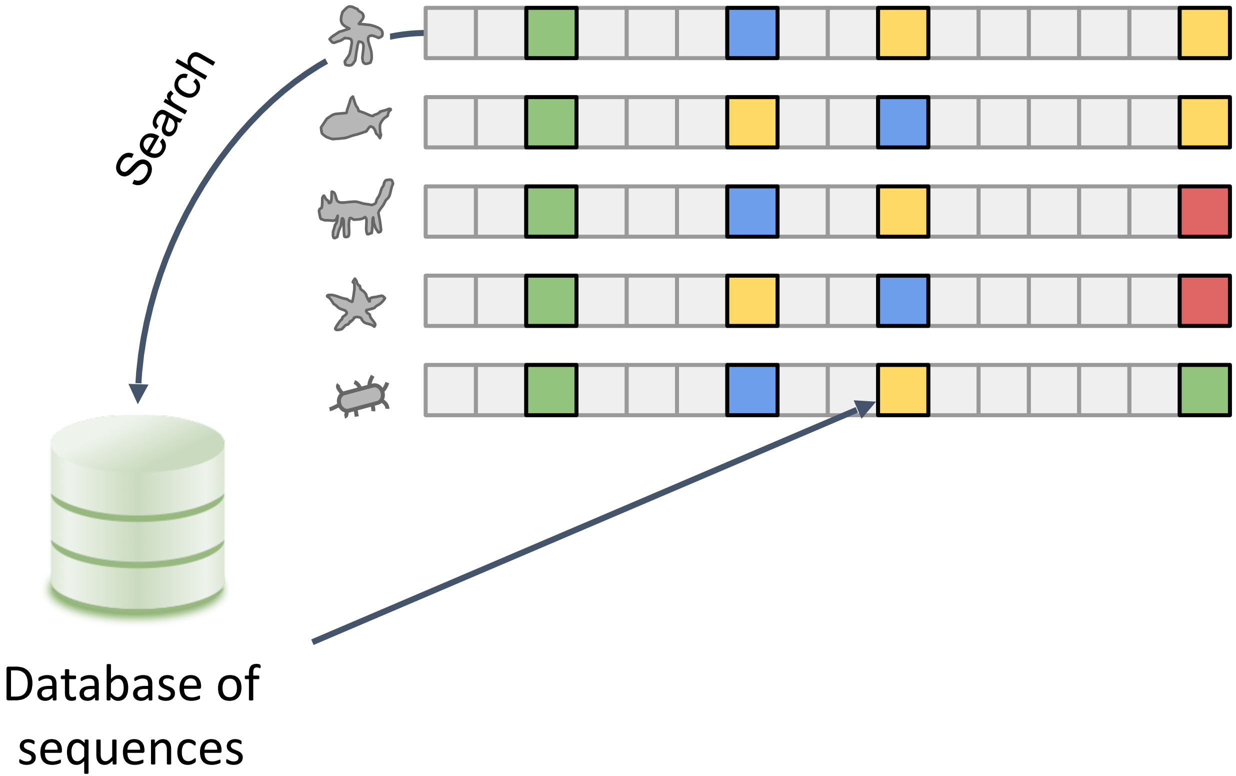
AF2 is a pretty good predictor of disorder

That is, when AlphaFold is unsure where the atoms should be, then Nature is too

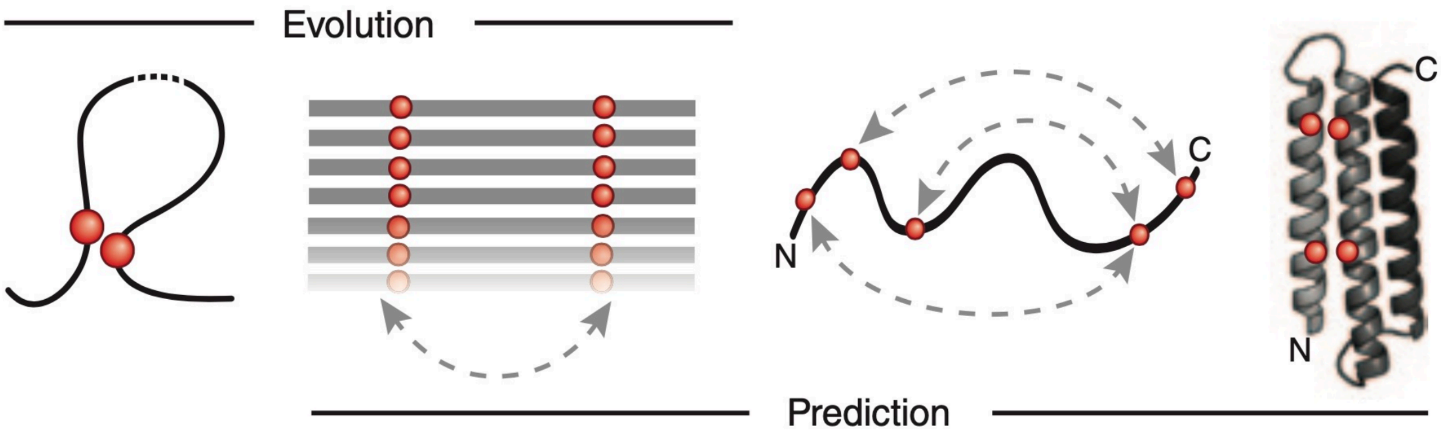


Tunyasuvunakool et al, Nature, 2021
Akdel et al, bioRxiv, 2021

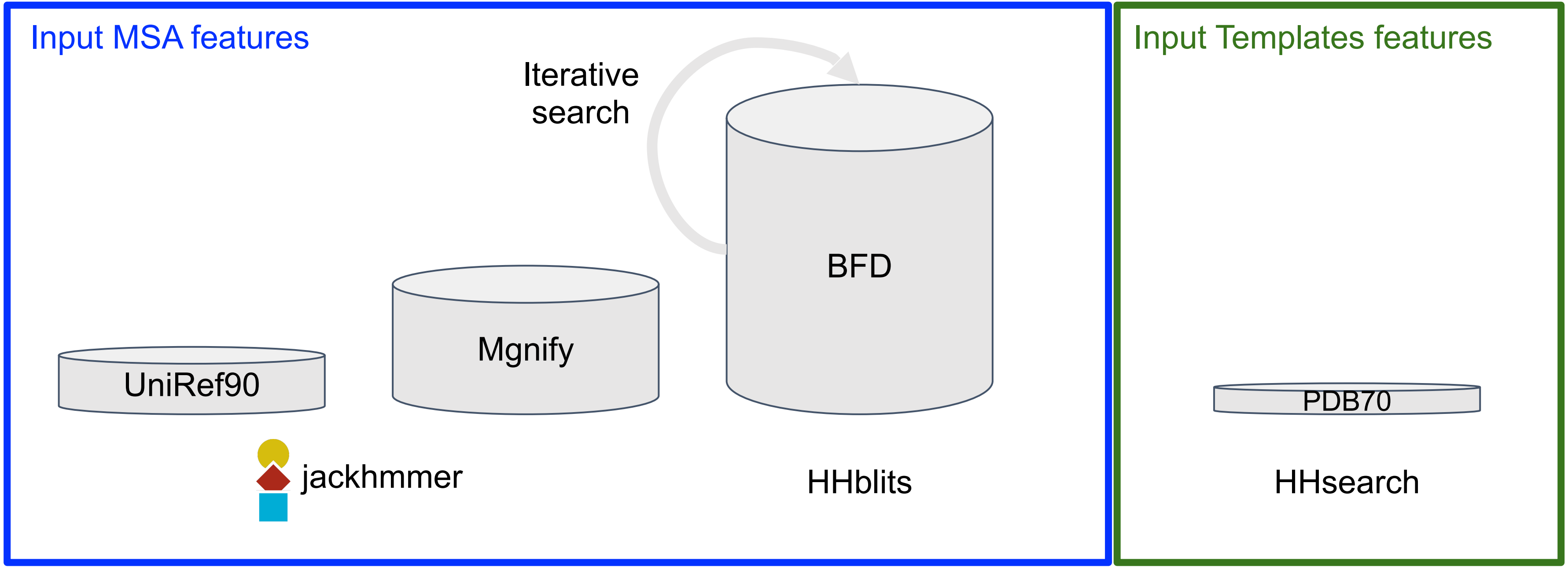
Co-evolution information from MSA



Multiple sequence alignment

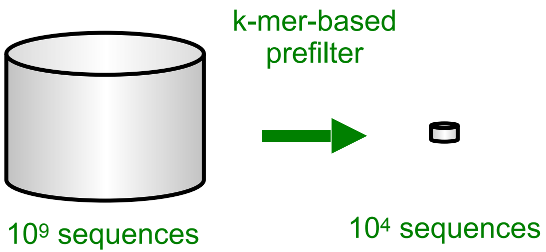


Input feature generation for AlphaFold2



Generation of input features can take **hours** for a single protein on multiple cores

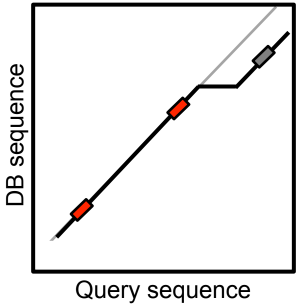
ColabFold uses MMseqs2 for fast MSA search



MMseqs2 key ideas

- Match *long* & *similar* *k*-mers

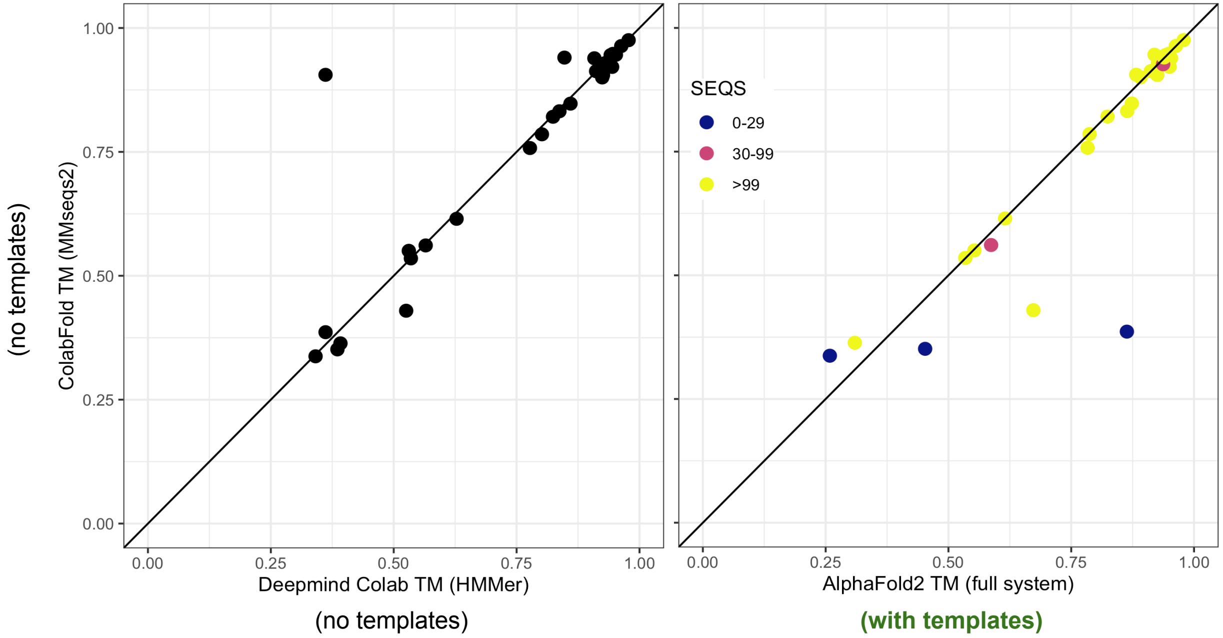
- Two *k*-mer matches  without gap in-between
- Sequence **profiles / iterative searches**



Steinegger and Söding, *Nature Biotechnol.*, 2017

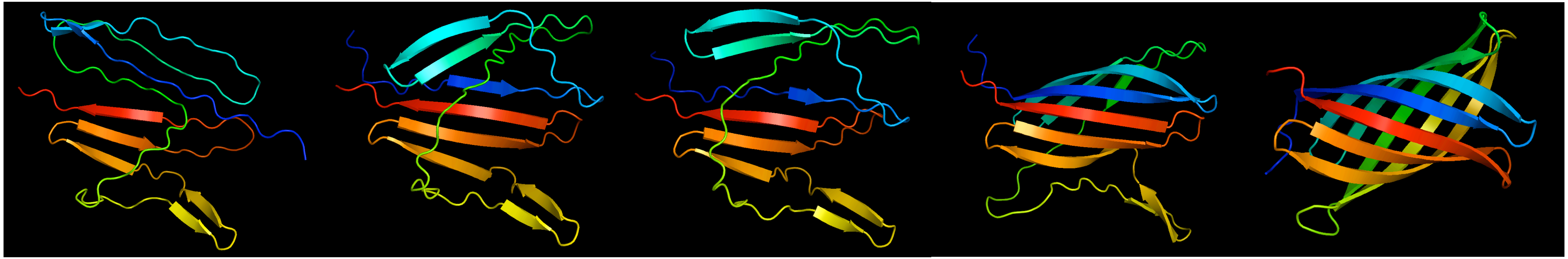
github.com/soedinglab/MMseqs2

ColabFold (MMseqs2) performs similar to the full Deepmind Colab and the AlphaFold2 system on CASP14-FM



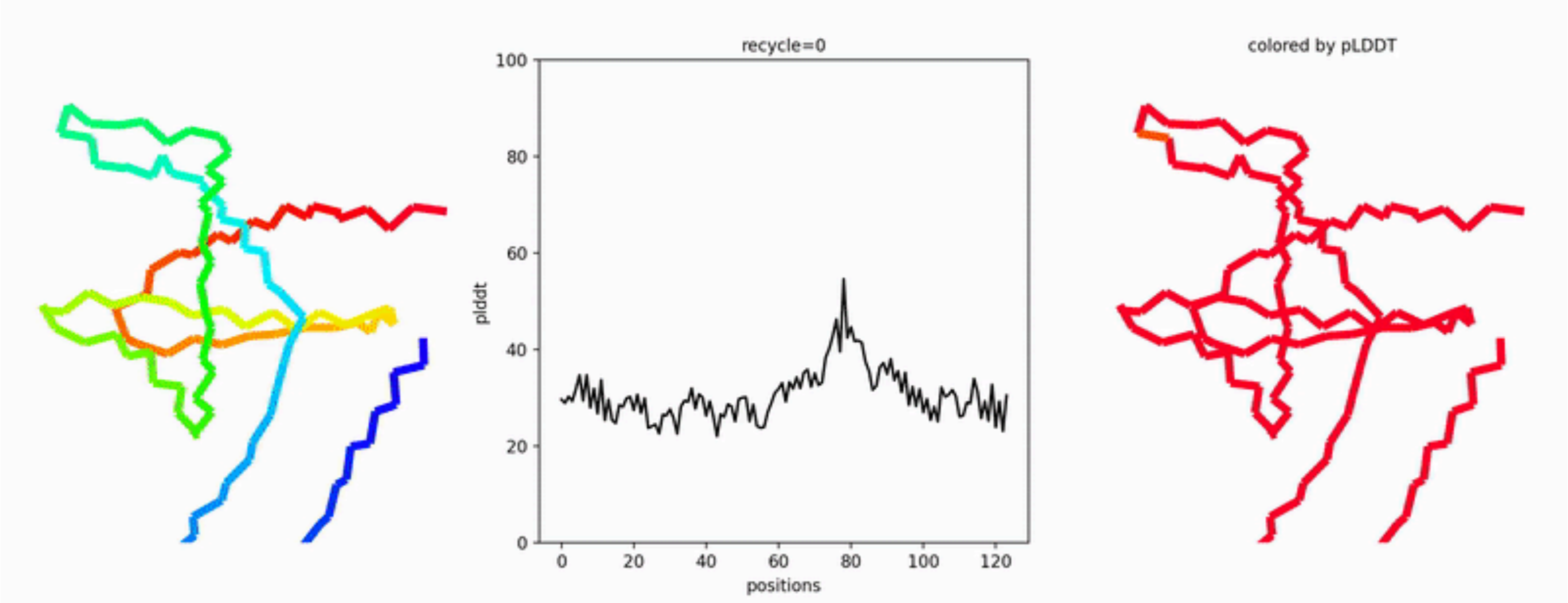
Generation of MSA for all 20 sequence take **<4 minutes** on one core

Prediction of designed transmembrane protein



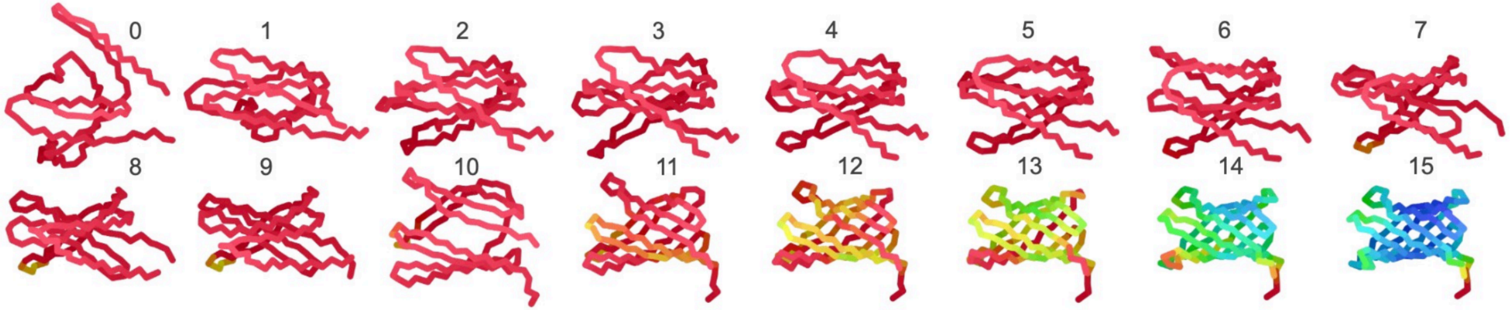
recycle 1 recycle 2 recycle 3 recycle 4 recycle 5

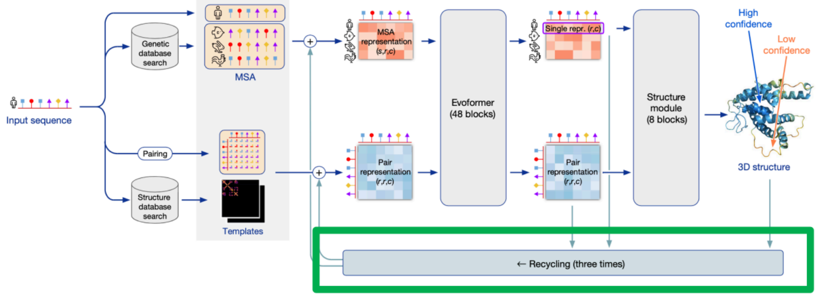
Another example: need up to 12 recycles to get the correct fold



Vorobieva, A.A., White, P., Liang, B., Horne, J.E., Bera, A.K., Chow, C.M., Gerben, S., Marx, S., Kang, A., Stiving, A.Q. and Harvey, S.R., 2021. De novo design of transmembrane β barrels. *Science*, *371*(6531).

Another example: need up to 12 recycles to get the correct fold

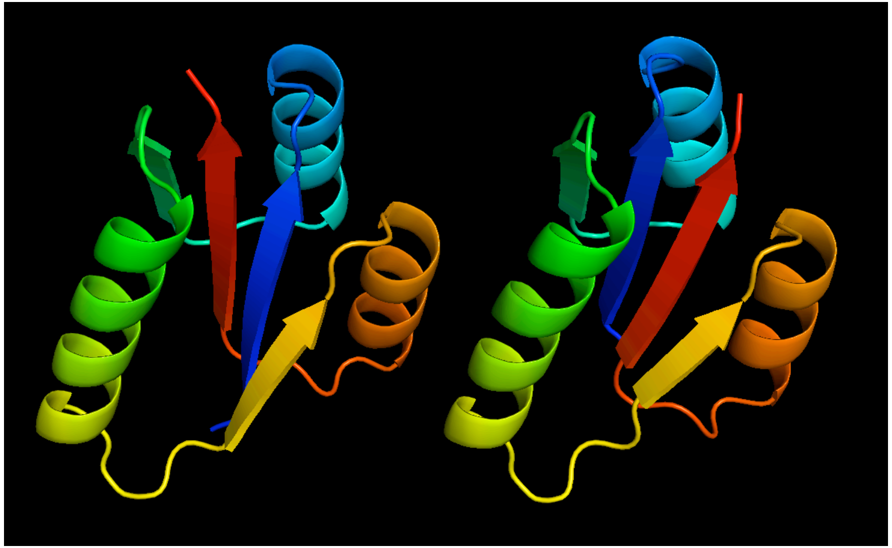




Mirdita et al, bioRxiv, 2021

Changing the seeds can give different results

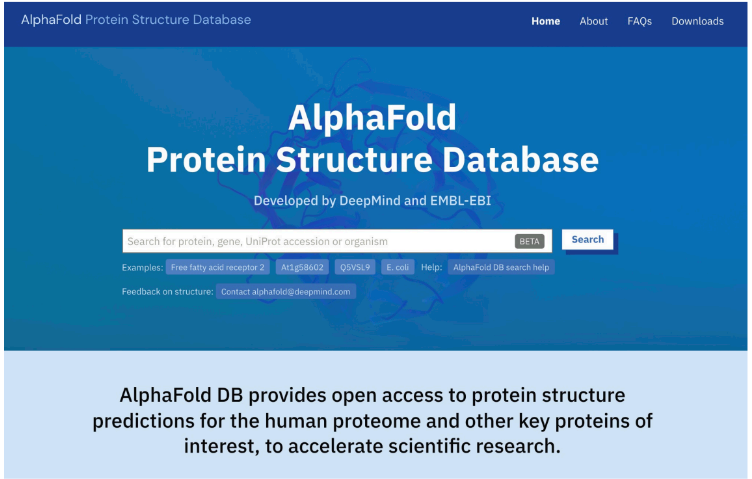
seed 0 seed 1

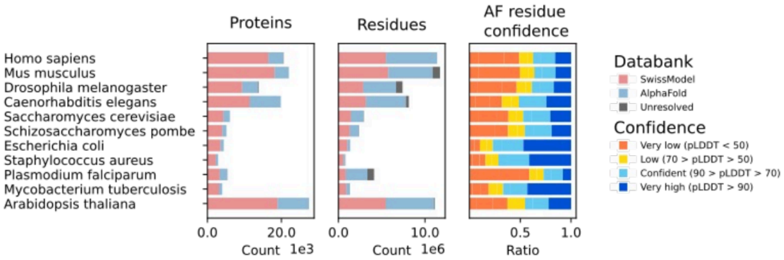


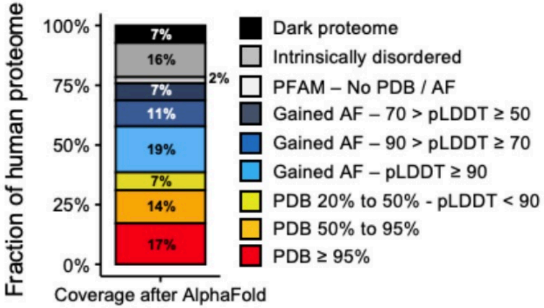
Protein with a knot, different seed value can change the outcome (knot or no knot)

Slide Credit: Sergey Ovchinnikov

Large scale application of AF2







Tunyasuvunakool et al, Nature, 2021
Varadi et al, Nucl Acid Res, 2021

Akdel et al, bioRxiv, 2021
Porta-Pardo, bioRxiv, 2021

Highly accurate protein structure prediction for the human proteome

nature > articles > article

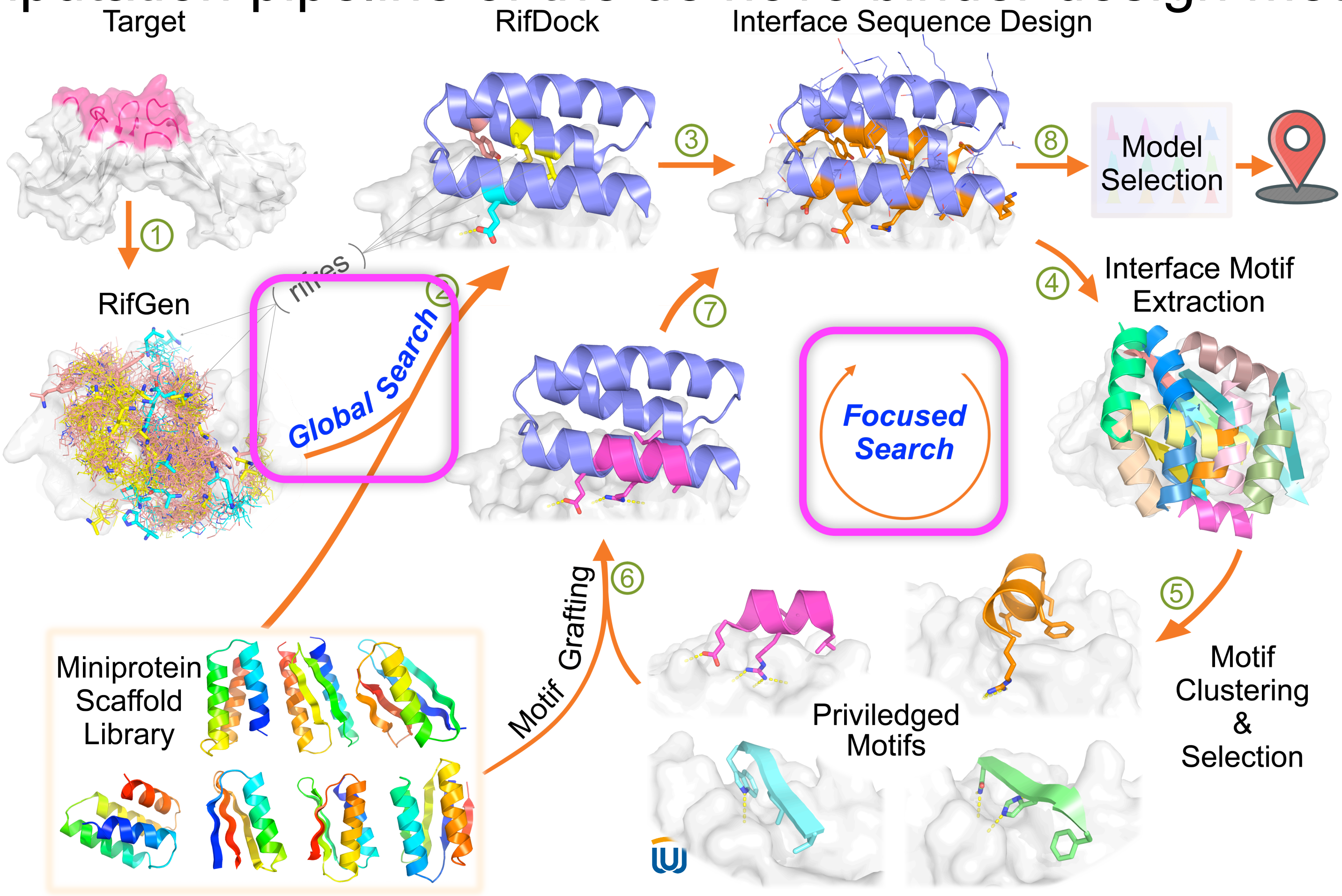
Article | Open access | Published: 22 July 2021

Highly accurate protein structure prediction for the human proteome

Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael Figurnov, Olaf Ronneberger, Russ Bates, Simon A. A. Kohl, Anna Potapenko, Andrew J. Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, ... Demis Hassabis + Show authors

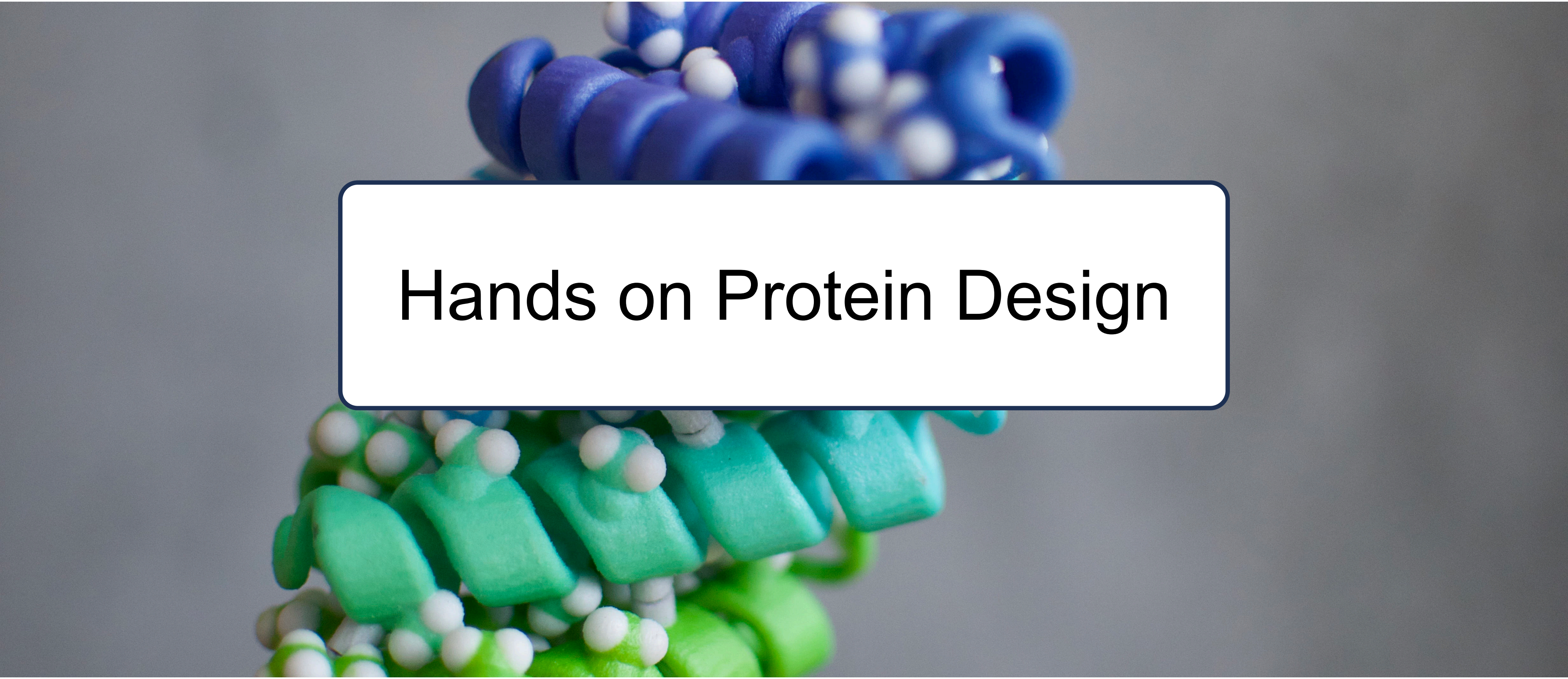
Nature **596**, 590–596 (2021) | Cite this article

Computation pipeline of the de novo binder design method



Protein binder design in the era of AI

RFdiffusion → **ProteinMPNN** → **AlphaFold2**



Hands on Protein Design

Bash

- **Bash**: Bash is a command-line interpreter. It translates commands you type into actions for the operating system (like macOS or Linux).
- **Prompt**: This is the symbol you see waiting for input (often a $ or %). It indicates the shell is ready to accept a command.
- **.bashrc**: a shell script that Bash executes every time a new interactive, non-login shell is started. It contains commands, functions, and configuration settings that you want to be run automatically.
- **Command Structure**: Most commands follow the pattern:

Command [Option/Flag] [Argument/File]

  - Options modify the command's behavior (e.g., -l for a long list format).
  - Arguments are the items the command acts upon (e.g., a file name or directory path).

Essential Navigation and File Commands

Command	Purpose	Example
pwd	Print Working Directory. Shows your current location.	pwd
ls	List directory contents.	ls -l (lists with detail)
cd	Change Directory.	cd .. (moves up one level); cd projects
mkdir	Make Directory. Creates a new folder.	mkdir new_results
touch	Creates an empty file.	touch readme.txt
cp	Copy files or directories.	cp fileA.txt /backup/
mv	Move or rename files/directories.	mv oldname.txt newname.txt
rm	Remove (delete) files. Use with caution!	rm junk.log
alias	create a shorthand for a longer command	alias b='cd ..'

Set up your environment for binder design first

Pull the latest version of tutorial repo:

cd /app/rfd_mpnn_af2_env && git pull origin main

cd *stands for Change Directory, which changes your current working directory in the terminal to your local Git repository.*

&& *is a conditional operator in shell scripting. It ensures that the command following it (git pull...) only executes if the preceding command (cd...) was successful (returned an exit code of 0).*

git pull *is a command that is actually a combination of two other Git commands.*

origin*: This is the default name for the remote repository you cloned from.*

main*: This is the name of the branch on the remote repository (origin) that you want to pull changes from and merge into your currently checked-out local branch.*

Slide Credit: wangchentong

Set up your environment for binder design first

There is a dot.

Copy the required target(sars2 spike protein) pdb files into the work dir:

```
cd /root/ && cp -r /app/rfd_mpnn_af2_env/input/ .
```

cp stands for copy. This command duplicates files or directories.

-r (or --recursive) is an essential option when copying directories. It tells the cp command to copy the directory specified as the source (/app/rfd_mpnn_af2_env/input/) and all its contents (subdirectories and files).

/app/rfd_mpnn_af2_env/input/: This is the source directory—the directory being copied.

. : This is the destination. In Linux, a single dot (.) is a shorthand reference for the current working directory. Since the first part of the command changed the working directory to /root/, the destination is /root/.

Set up your environment for binder design first

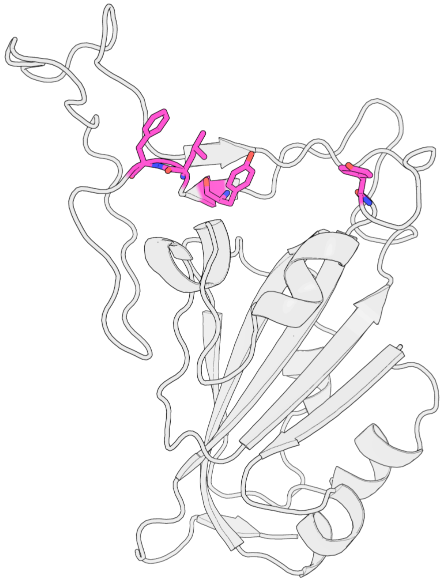
print the command instruction in /app/rfd_mpnn_af2_env/:

cat /app/rfd_mpnn_af2_env/run.sh

cat: *This stands for concatenate. It's a standard Unix utility that reads files sequentially and writes them to standard output (usually your terminal screen).*
/app/rfd_mpnn_af2_env/run.sh*: This is the path to the file whose contents you want to view.*

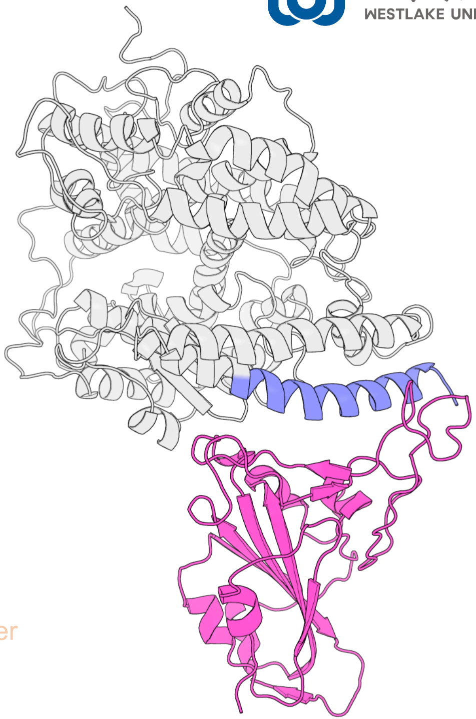
1.Scaffold generation : unconditional binder design(option 1)

python /app/RFdiffusion/scripts/run_inference.py
# This calls the Python interpreter to execute the main inference script for RFdiffusion.
inference.input_pdb=/app/rfd_mpnn_af2_env/input/Spike_glycoprotein.pdb:
# The design process will begin from the state defined in this PDB file, which appears to be the Spike glycoprotein (from SARS-CoV-2).
'contigmap.contigs=[B1-193/0 90-90]':
This is the design constraint defining the backbone segments.
B1-193，Target chain and residue index
/0 ，Chain gap(means the next comes another chain
90-90, Binder length(how many residues in binder)
'ppi.hotspot_res=[B120,B122,B123,B160,B172]':
# This is the key constraint for protein-protein interaction (PPI) design.
# It specifies a list of residue indices on chain B (the Spike glycoprotein) that are considered hotspot residues.
inference.ckpt_override_path=/app/RFdiffusion/models/Complex_base_ckpt.pt
# This explicitly sets the path to the model weights (checkpoint file).
# Complex_base_ckpt.pt confirms this is using the RFdiffusion model specifically trained for complex structure generation/binder design.
denoiser.noise_scale_ca=0 denoiser.noise_scale_frame=0
# These parameters control the amount of noise added during the reverse diffusion process.
# Setting both to 0 means no noise is added to the structure during the generation process.
inference.output_prefix=samples/uncondition:
# This specifies the path and filename prefix where the resulting PDB files will be saved



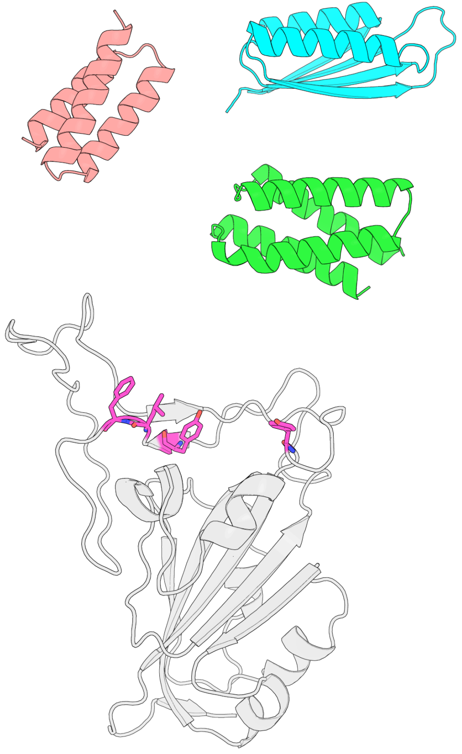
1.Scaffold generation : conditional binder design with ACE2 motif(option 2)

python /app/RFdiffusion/scripts/run_inference.py:
# Executes the main RFdiffusion inference script.
inference.input_pdb=/app/rfd_mpnn_af2_env/input/Spike_glycoprotein_complex.pdb:
# The starting structure for the diffusion process.
# This PDB file is the Spike glycoprotein already in a complex (with the ACE2 motif)
'contigmap.contigs=[B1-193/0 A19-42/60-60]':
# This is the core design constraint, defining the fixed and designed segments:
# B1-193: This segment of Chain B (the Spike glycoprotein) is fixed (not designed).
# /0: Chain gap.
# A19-42: This segment of Chain A (the ACE2 motif) is also fixed. This is the motif that the new binder must incorporate and correctly position.
# /60-60: This instructs the model to design a new connecting sequence of length 60 amino acids
inference.ckpt_override_path=/app/RFdiffusion/models/Base_ckpt.pt:
# Uses the general-purpose RFdiffusion model (Base_ckpt.pt), rather than the specialized Complex model.
denoiser.noise_scale_ca=0 denoiser.noise_scale_frame=0:
# setting these to 0 removes added noise, resulting in a more deterministic or direct sampling run guided strictly by the input structure and contigs.
inference.output_prefix=samples/ace2:
# The generated PDB files will be saved with the prefix samples/ace2, indicating that the results are the binder scaffolds designed around the ACE2 motif.



1.Scaffold generation : conditional binder design with scaffold library(option 3)

python /app/RFdiffusion/scripts/run_inference.py:
# Executes the main RFdiffusion inference script.
scaffoldguided.scaffoldguided=True:
# Activates the Scaffold-Guided mode.
scaffoldguided.scaffold_dir=/app/RFdiffusion/examples/ppi_scaffolds/:
# Specifies the path to the library of small, pre-folded, stable scaffolds (PDB files) that the model will sample from. The model tries to graft each of these scaffolds onto the target.
scaffoldguided.target_pdb=True:
# Confirms that a target PDB file is being provided.
scaffoldguided.target_path=/app/rfd_mpnn_af2_env/input/Spike_glycoprotein.pdb:
# Specifies the target protein (Spike glycoprotein) that the designed binder must interact with.
scaffoldguided.target_ss=/app/rfd_mpnn_af2_env/input/Spike_glycoprotein_ss.pt scaffoldguided.target_adj=/app/rfd_mpnn_af2_env/input/Spike_glycoprotein_adj.pt
# These arguments provide pre-calculated structural features (secondary structure, ss, and adjacency matrices, adj) for the target protein. Loading these pre-computed features speeds up the inference process.
'ppi.hotspot_res=[B120,B122,B123,B160,B172]':Retains the core constraint:
# The designed scaffold/binder must specifically interact with these designated hotspot residues on Chain B.
inference.ckpt_override_path=/app/RFdiffusion/models/Complex_Fold_base_ckpt.pt:
# Uses a highly specialized model checkpoint
denoiser.noise_scale_ca=0 denoiser.noise_scale_frame=0:
# setting noise scales to zero suggests a highly deterministic process
inference.output_prefix=samples/scaffold_guided:
# The output PDB files will be saved under this prefix.



2.Sequence design with ProteinMPNN

```
# Executes the main script
python /app/dl_binder_design/mpnn_fr/dl_interface_design.py \

# Input scaffold directory(generated by RFdiffusion in last step)
-pdbdir ./samples/ \

# Output pdb directory(binder scaffolds with added sequence and sidechain atoms)
-outpdbdir ./mpnn/ \

# How many cycles of ProteinMPNN+Fastrelax optimization(more cycles improve self-consistency between binder structure and sequence)
-relax_cycles 0

# Higher temperature generate more diverse sequence from single scaffold but more likely unfolded/unbind
-temperature 0.0001

# The number of sequence one scaffold generate
-seqs_per_struct 4
```

3.Alphafold2 Prediction

```
# Executes the main script
python /app/dl_binder_design/af2_initial_guess/predict.py \

# Design directory(generated by ProteinMPNN in last step)
-pdbdir ./mpnn/ \

# Output pdb directory (binder scaffolds with added sequence and sidechain atoms)
-outpdbdir ./predictions/ \

# Number cycles in alphafold2, more cycles mean more accurate prediction but slower
-recycle 3

# Turn on initial guess, use design models as a hint to alphafold2 for improved success rate
-no_initial_guess False
```

AlphaFold Confidence Scores

- pLDDT (Predicted Local Distance Difference Test).
- pLDDT is a per-residue confidence metric that estimates the local accuracy of the predicted structure.

pLDDT Score Range	Confidence Level	Meaning for the Structure	Color in Visualization
> 90	**Very High**	The residue is placed with extremely high accuracy; the side chain and backbone coordinates are reliable.	**Dark Blue**
70 – 90	**High**	The backbone is generally correct, but the side chain placement may have minor errors.	Cyan
50 – 70	**Low**	The backbone placement is poorly defined; this region may be flexible or exposed.	Yellow
< 50	**Very Low**	The region is likely **unstructured, intrinsically disordered, or highly flexible**. The coordinates are not reliable.	Red/Orange

AlphaFold Confidence Scores

- PAE (Predicted Aligned Error). PAE is a global confidence metric that estimates the error in the relative positions of two residues (i and j) after the entire predicted structure is optimally aligned on residue i.

PAE Score (Color)	Interpretation	Meaning for the Structure
Low Error (Dark Blue/Green)	High confidence in relative position.	The two residues are confidently placed relative to each other. This often means they belong to the **same rigid domain** or are part of a stable interface.
High Error (Light Yellow/White)	Low confidence in relative position.	The relative placement of the two residues is highly uncertain. This is typical for residues in **different domains connected by flexible linkers**, or in disordered regions.

Root Mean Square Deviation (RMSD)

- RMSD is a measure of the average distance between the corresponding atoms of two superimposed molecular structures.

Context	Application of RMSD
Benchmarking	RMSD is used to compare AlphaFold's model to experimentally validated structures in the Protein Data Bank (PDB). This is how the accuracy of AlphaFold's various versions (like AlphaFold 2) is definitively.
Structural Analysis	Researchers use RMSD to evaluate the prediction quality of specific regions, such as loop regions.

西湖大學 WESTLAKE UNIVERSITY

PyMOL, a widely used, powerful molecular visualization system