

# Protein Design

## Part 3: *de novo* design

Anastassia Vorobieva



## Workshop Schedule

TIME	WEDNESDAY 28 MAY	THURSDAY 29 MAY	FRIDAY 20 MAY
<b>09:00 – 10:30</b>	Protein modelling and structure prediction: Intro to key concepts, from physics-based modelling to AI	De novo protein design: introduction, minimal sequence design, structure-based design principles	De novo design with AI models: RFDiffusion, ProteinMPNN, and ColabFold
<b>10:30 – 11:00</b>	Break and questions	Break and questions	Break and questions
<b>11:00 – 12:00</b>	Introduction to protein design: predicting the effect of mutations on protein stability	Structure-based de novo design: How to generate new structures? The chicken-and-egg problem	Practical session: De novo design of a SARS-CoV-2 RBD binder using RFDiffusion and ProteinMPNN
<b>12:00 – 13:30</b>	Lunch	Lunch	Lunch
<b>13:30 – 15:00</b>	Practical session: AlphaFold hands-on	Practical session: Parametric design of alpha-helical bundles	Practical session: Data analysis and group slide preparation
<b>15:15 – 17:45</b>	Practical session: In silico mutational scanning and $\Delta\Delta G$ calculations	Practical session: Sequence design for parametric bundles with PyRosetta	Practical session: Group presentations and results discussion



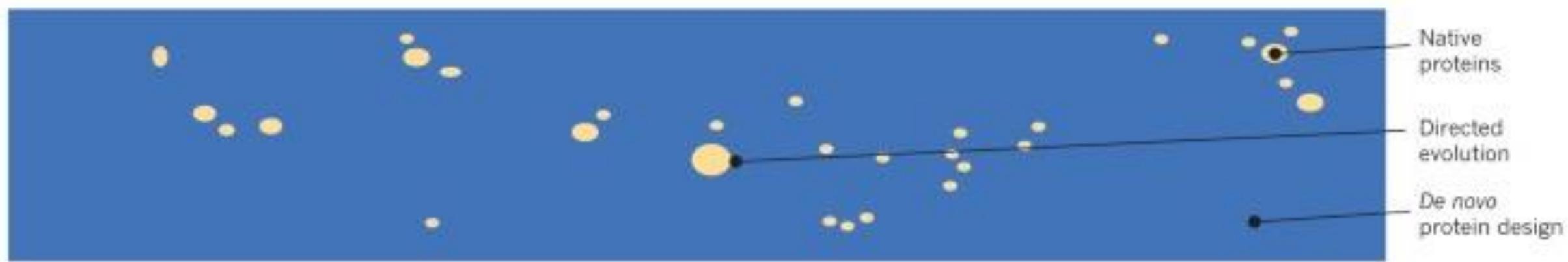
# *De novo* protein design

1. Introduction to protein design:
  - Definition and evolution in time
2. Minimal sequence design
3. Computational design
  - Principle
  - Pipeline
  - Designable backbones
  - Scoring and sampling
  - Parametric backbone generation
  - Fragment-based backbone generation

# Introduction - protein design

Generating new-to-nature protein structure and sequences.

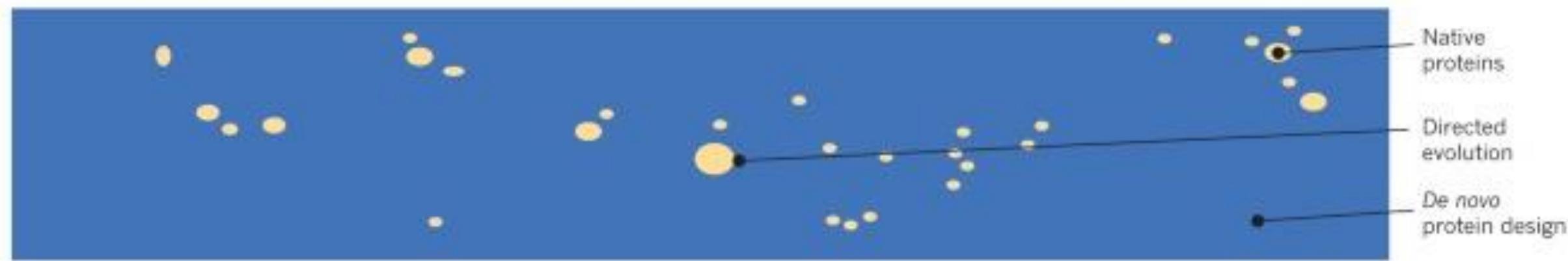
# *De novo* protein design – exploring the dark protein sequence space



Single helix of 30 residues:  $20^{30} = 10^{39}$  possible sequences

Which of these sequences encode foldable proteins?

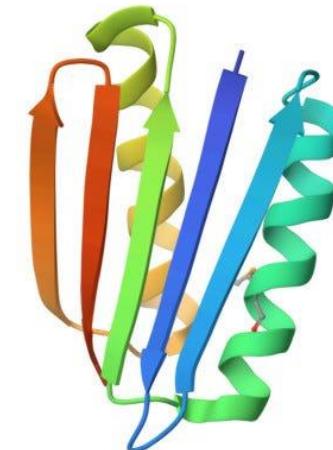
# *De novo* protein design – exploring the dark protein sequence space



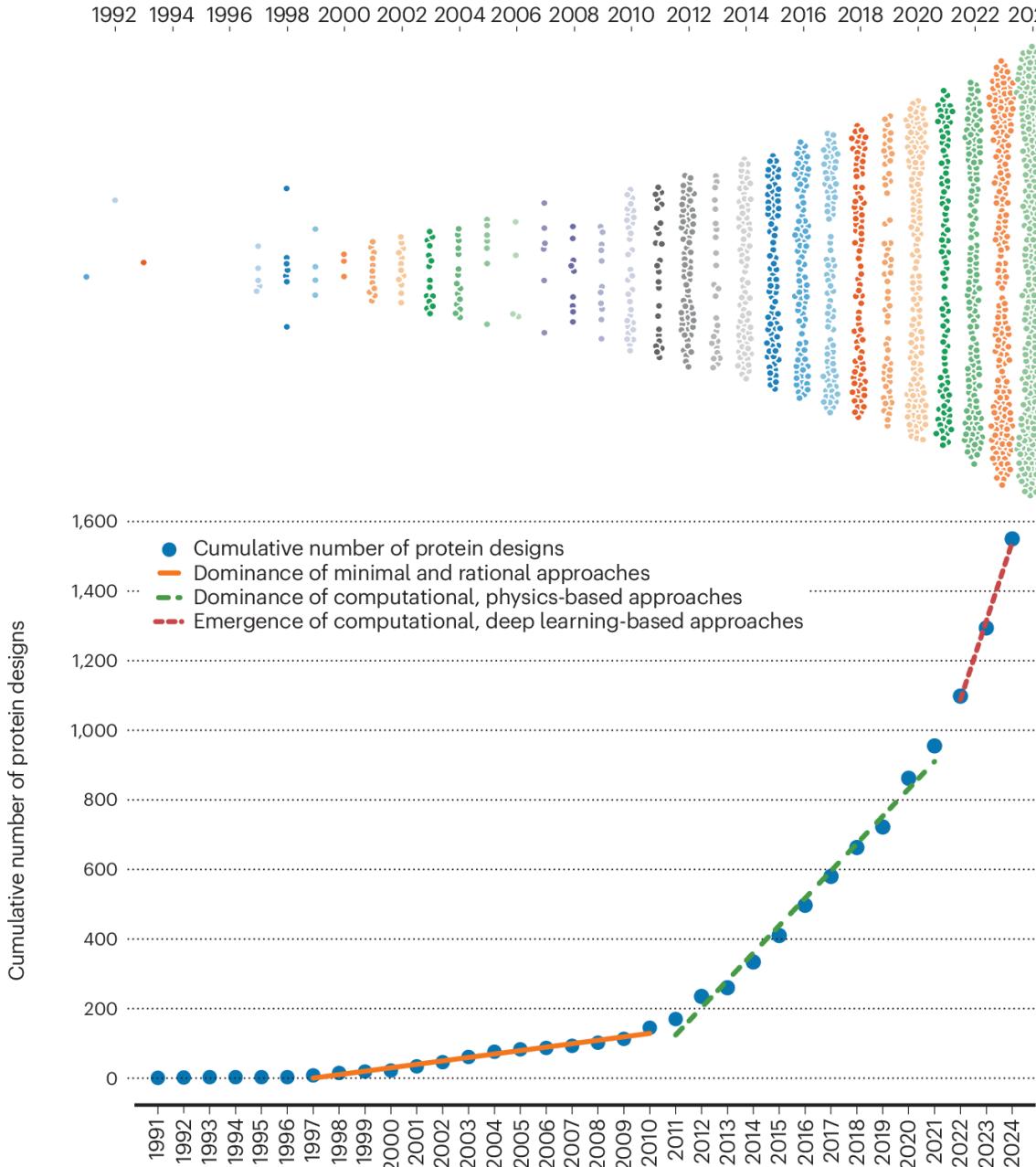
Single helix of 30 residues:  $20^{30} = 10^{39}$  possible sequences

Which of these sequences encode foldable proteins?

Top7 – first time that a new-to-nature fold was designed from scratch

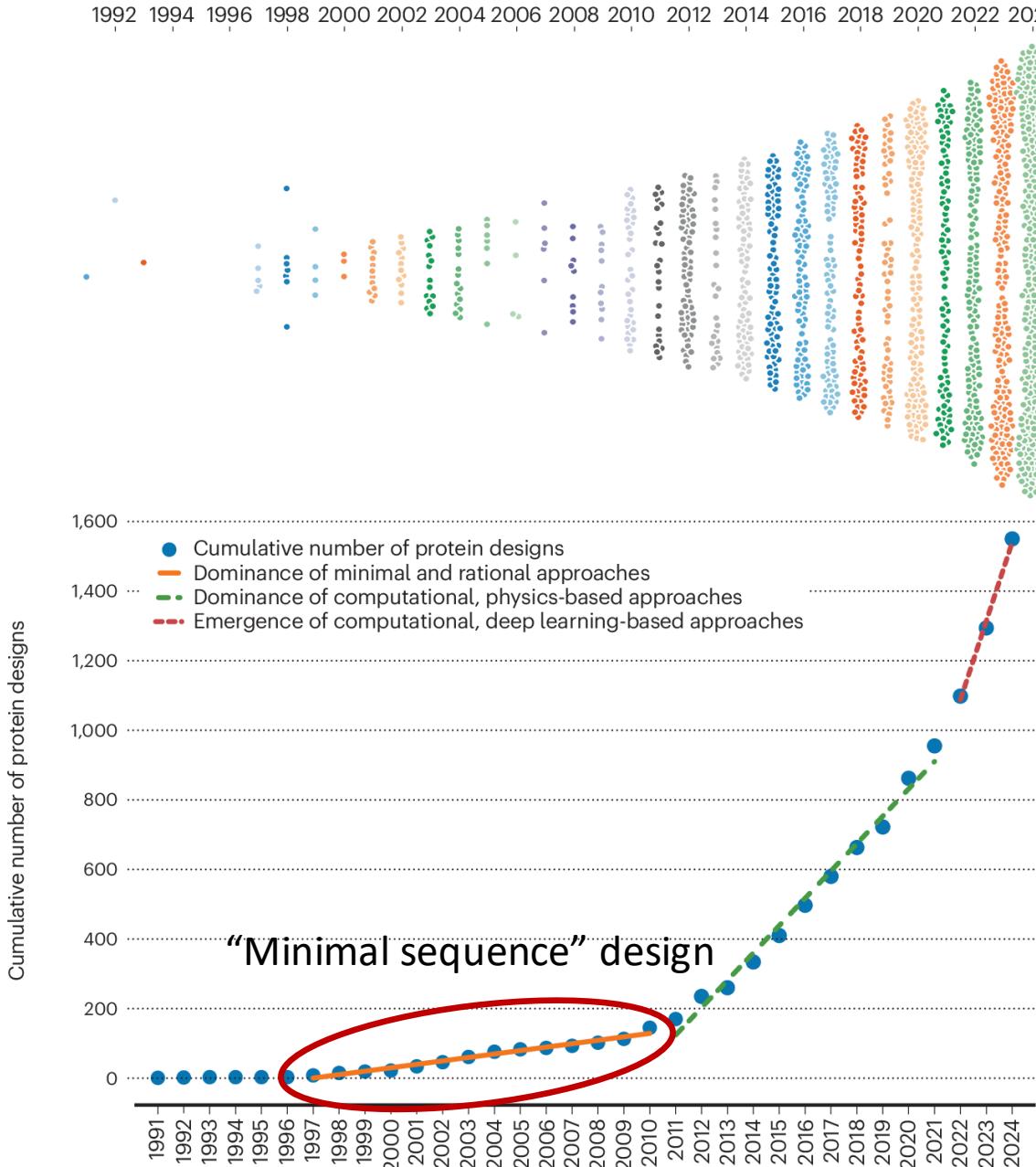


# Evolution of *de novo* protein design



# Minimal sequence design

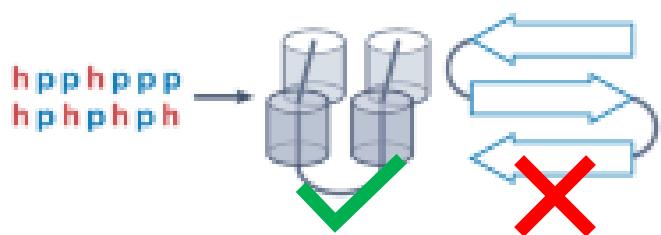
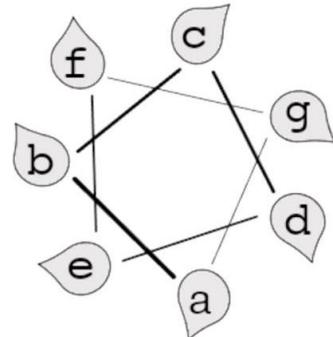
# Evolution of *de novo* protein design



# Minimal sequence design

TM helix 1 ...MPYIYLAI AIAAAEVVGT SALK...  
TM helix 2 ...LIPSVGTLVGYGASFYLLSLT...  
TM helix 3 ...YALWSGIGIVAI SLVGWILF...  
TM helix 4 ...LDLMKIVGLALIVAGVVILNL...

Pattern ...hhhhpGhGhhhphhGhhhph...  
Idealized ...LLL SGLGL LLL SLL GLL LLS...  
Position ...abcdefg abcdefg abcdefg...



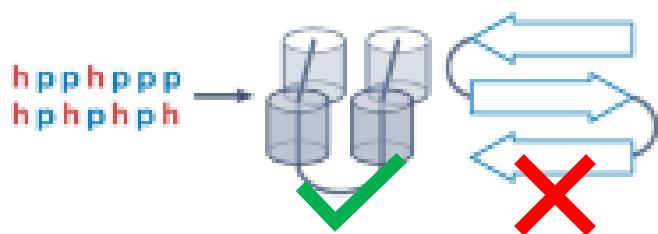
## Typical design:

1. Glycine
2. Proline
3. One hydrophobic amino acid  
(Leucine or Valine)
4. One polar amino acid (Serine)

# Minimal sequence design

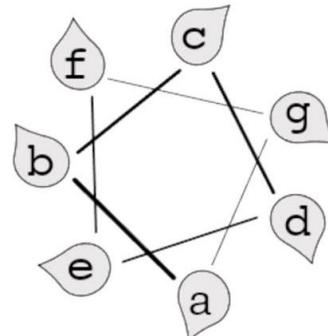
TM helix 1 ...MPYIYLAIAlAAEVVGTALK...  
TM helix 2 ...LIPSVGTLVGYGASFYLLSLT...  
TM helix 3 ...YALWSGIGIVAlAISLVGVWILF...  
TM helix 4 ...LDLMKIVGLALIVAGVVILNL...

Pattern ...hhhhhpGhGhhhphhGhhhp...  
Idealized ...LLLLSGLGLLLLSSLGLLLLS...  
Position ...abcdefgabcdefgabcdefg...

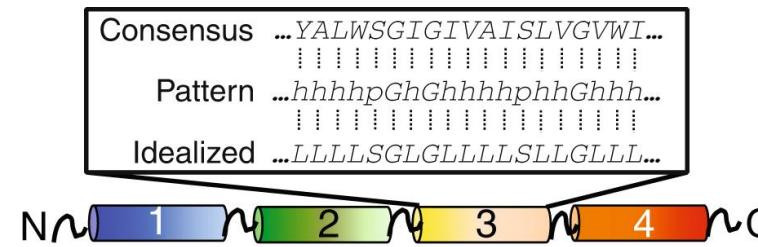


## Typical design:

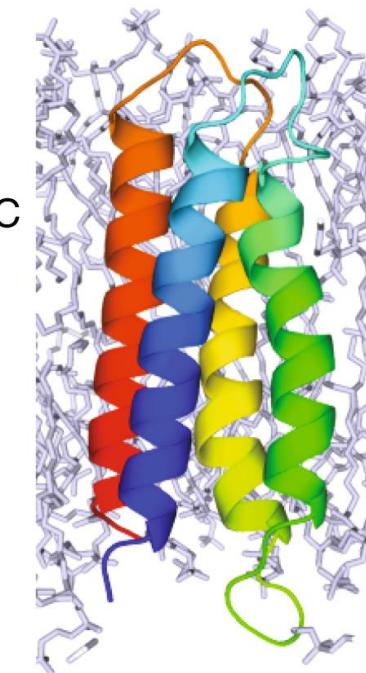
1. Glycine
2. Proline
3. One hydrophobic amino acid  
(Leucine or Valine)
4. One polar amino acid (Serine)



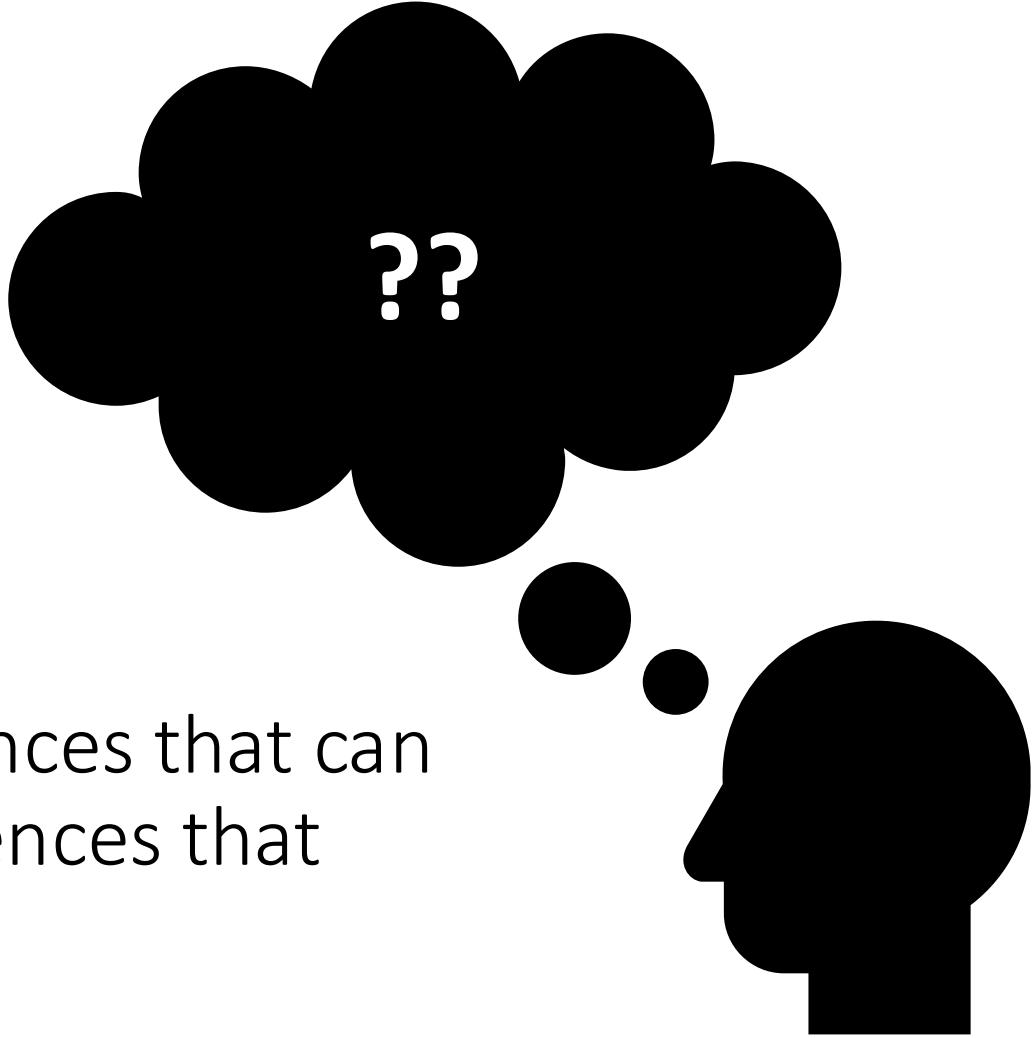
## Example: Minimal membrane protein



Membrane	Periplasm				Cytoplasm							
	S	G	E	E	G	S	S	G	E	E	G	S
M	L	L	S	S	L	L	L	L	L	L	L	L
L	G	L	L	S	G	L	S	G	L	L	S	G
L	S	L	L	S	G	L	L	S	L	L	S	L
G	L	L	L	S	L	S	G	L	L	S	L	L
L	S	G	L	L	G	L	L	S	G	L	L	S
V	L	L	L	S	W	S	S	L	L	S	G	S

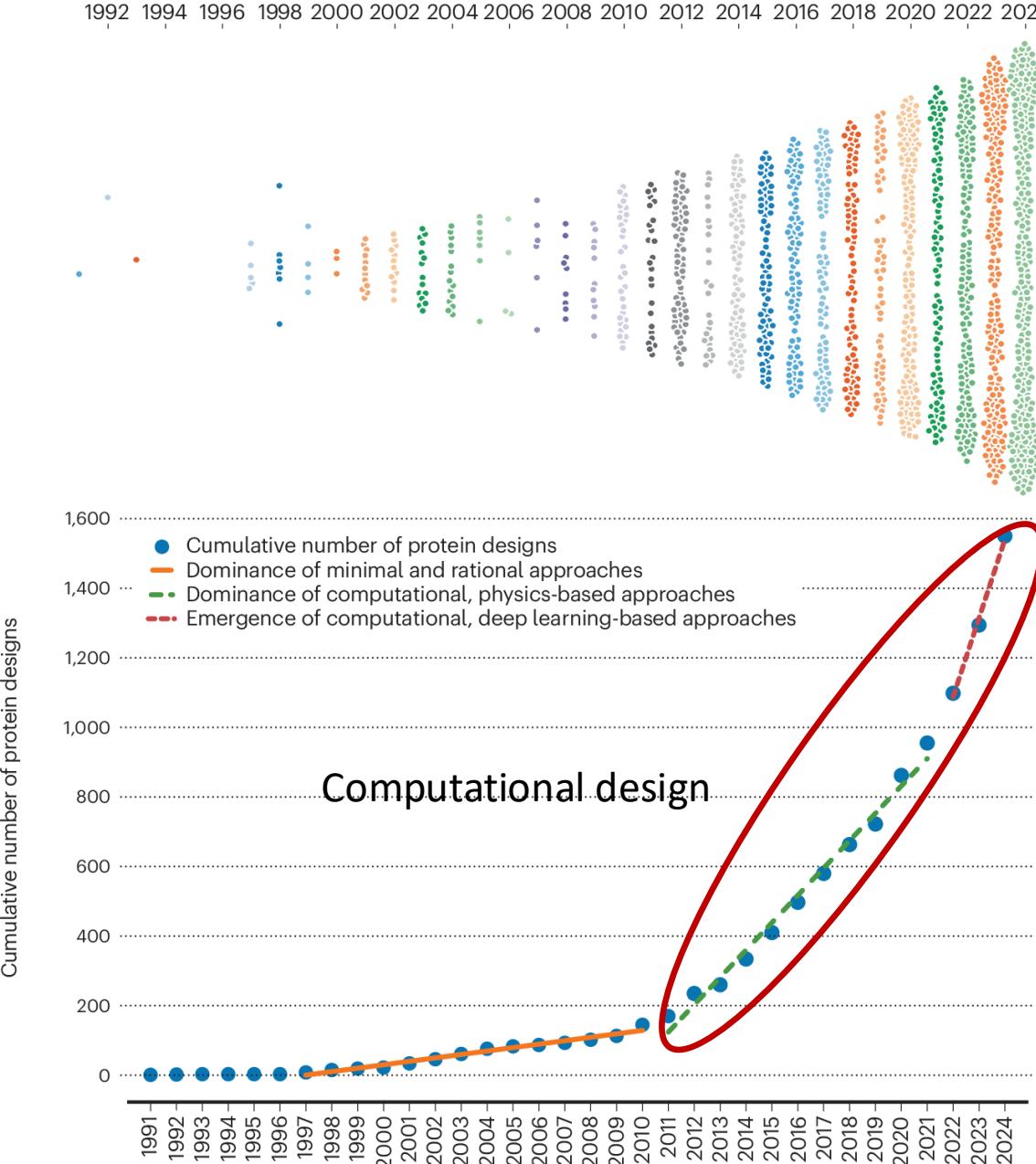


Minimal design generates sequences that can just fold? Can we generate sequences that have functions?



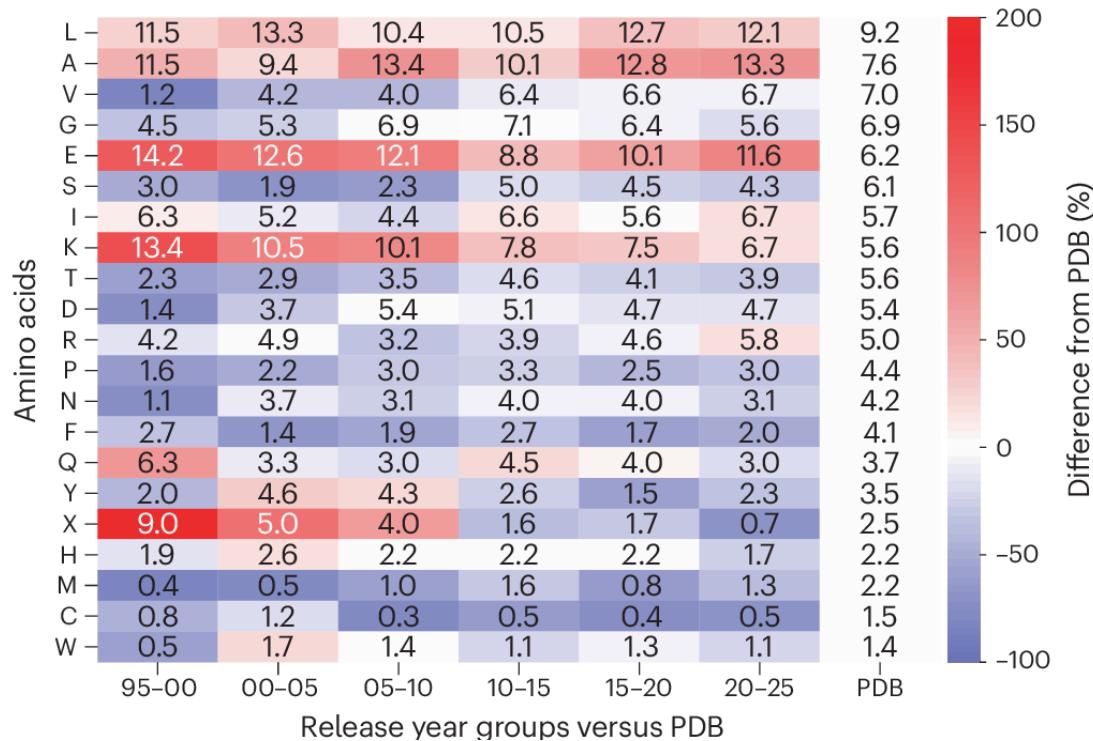
Computational design (structure-based)

# Evolution of *de novo* protein design

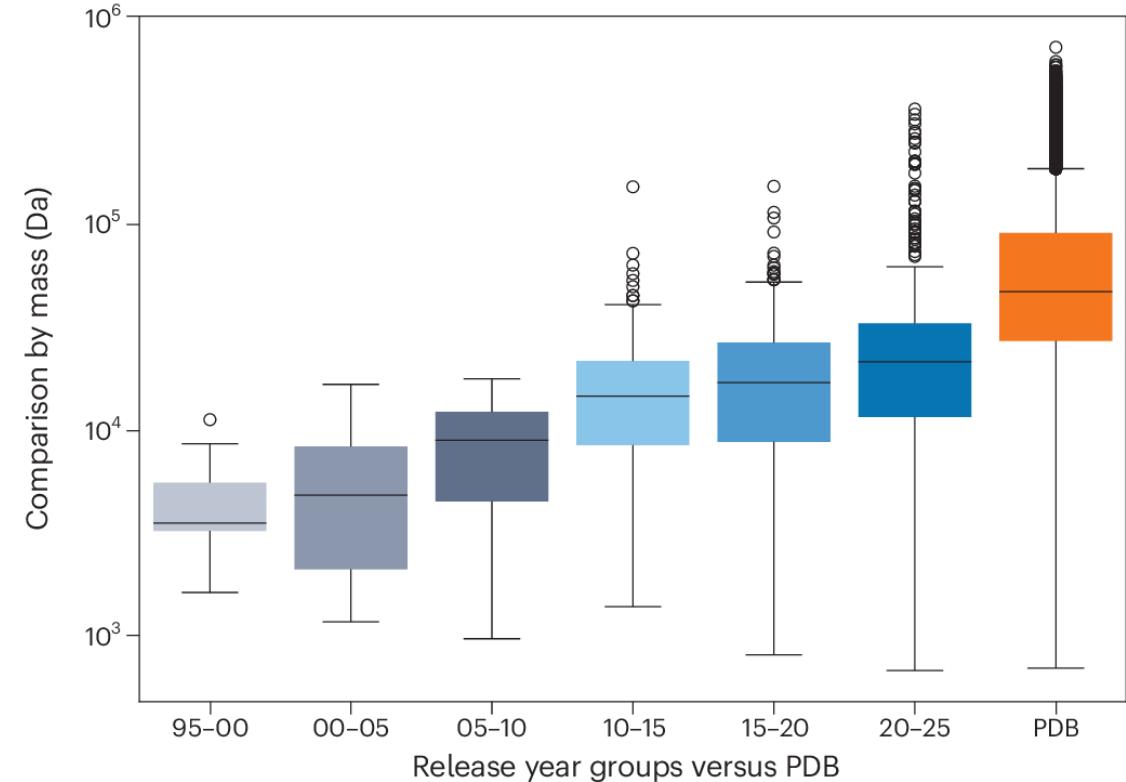


# Computational design enabled the generation of more diverse and longer sequences

**a Design amino acid percentages by year group versus PDB**



**c Comparison by mass (Da)**



The folded states of proteins are likely global energy minima for their sequences (C. Anfinsen)

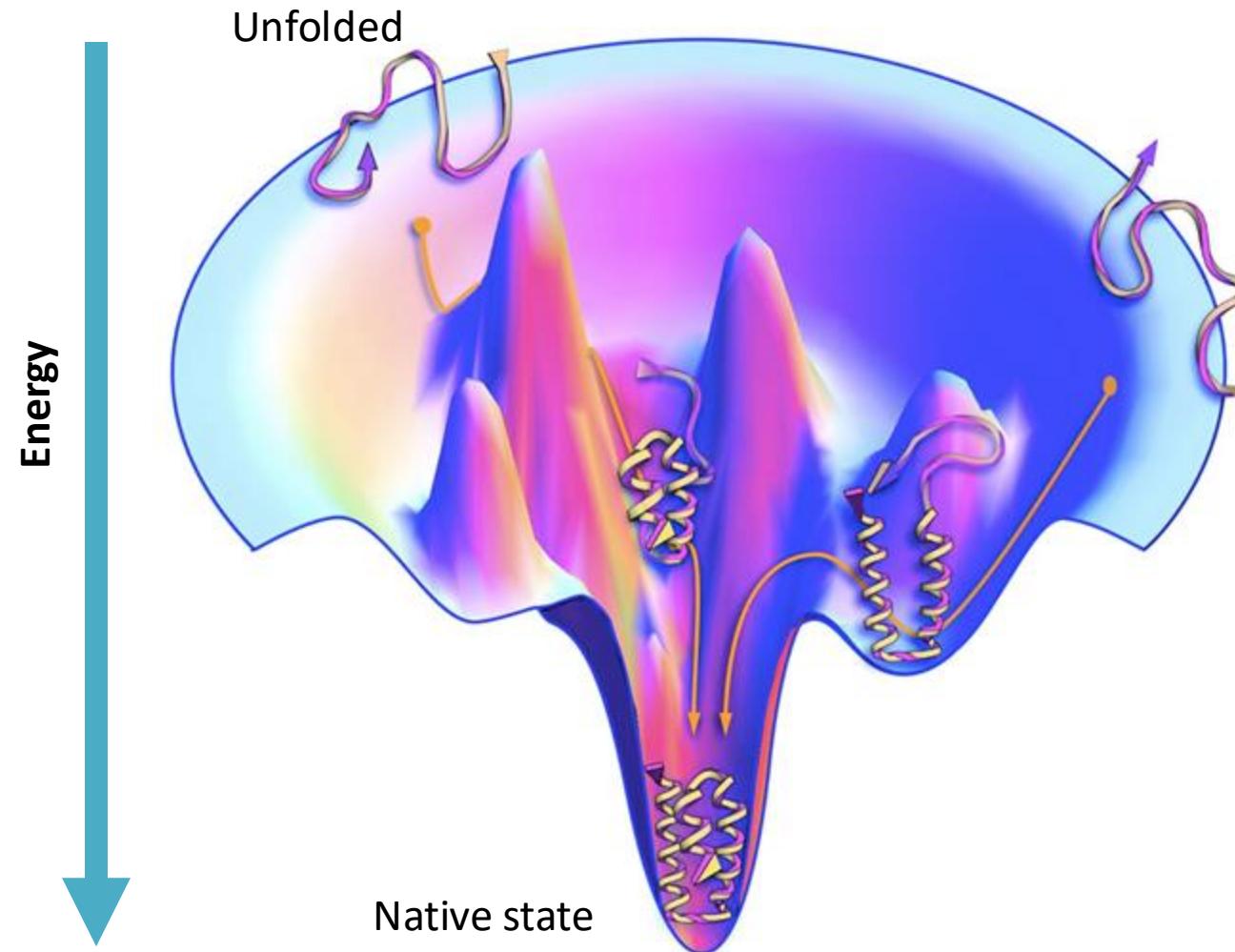


Image from: "The protein-folding problem, 50 years on." science 338, no. 6110 (2012): 1042-1046.

# Structure prediction(*ab initio*) and design

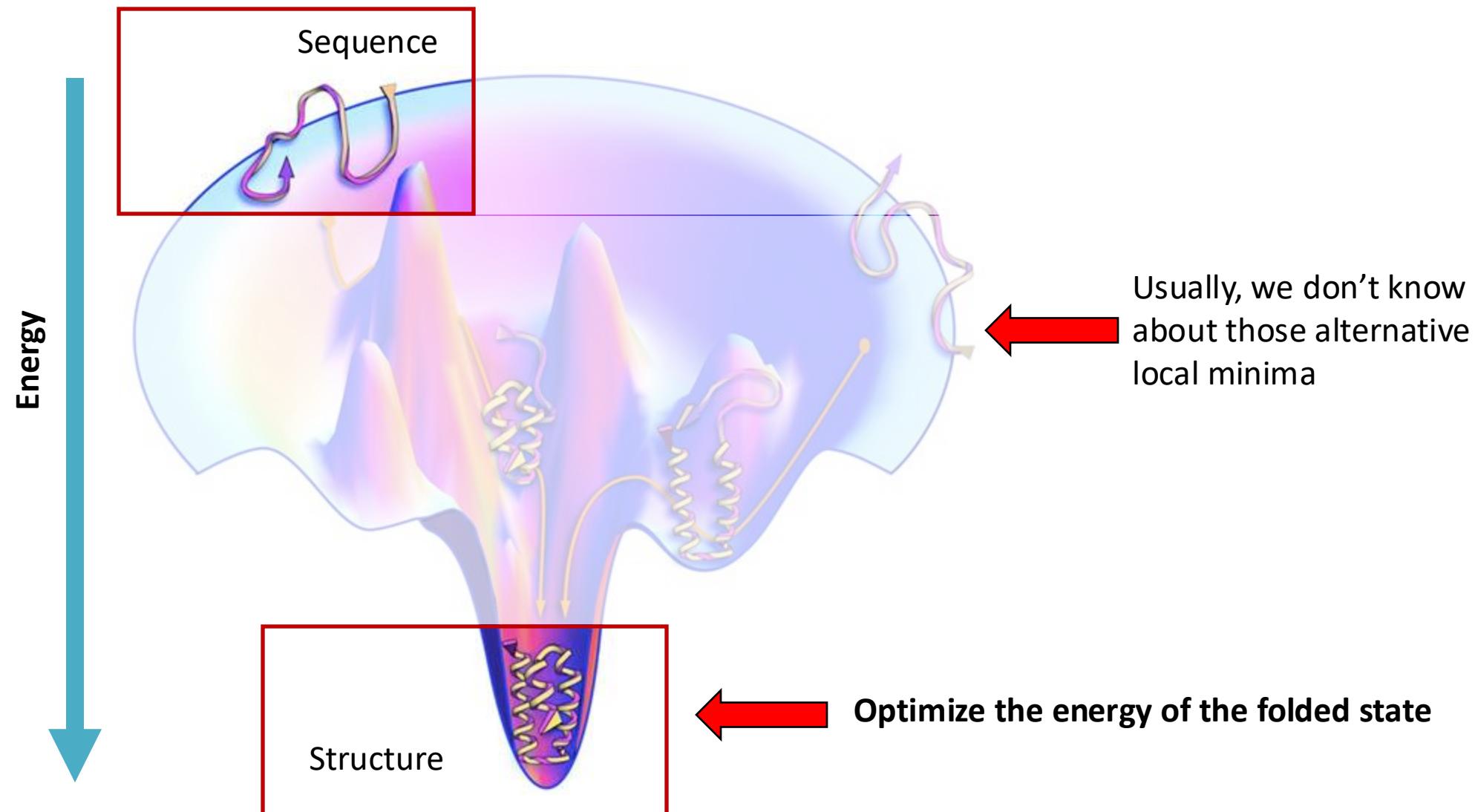
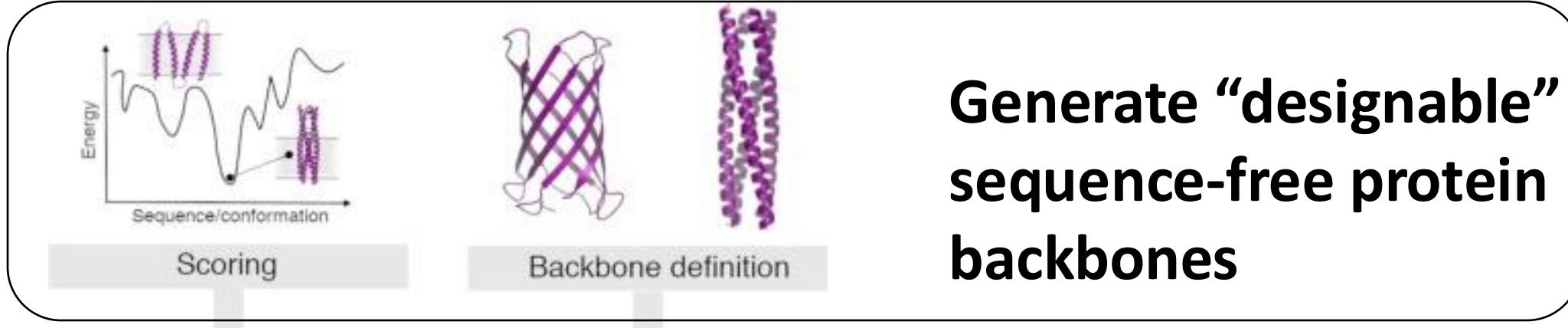


Image from: "The protein-folding problem, 50 years on." science 338, no. 6110 (2012): 1042-1046.

# *De novo* design pipeline

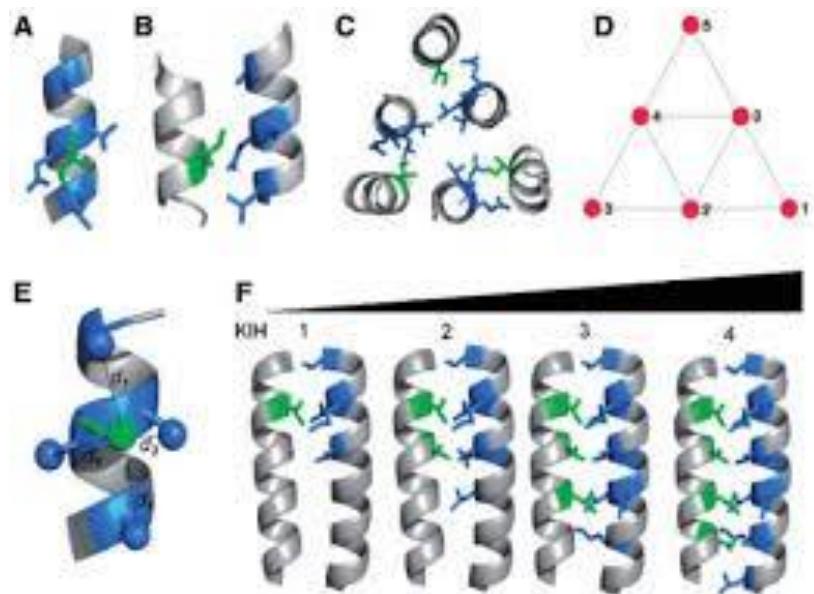
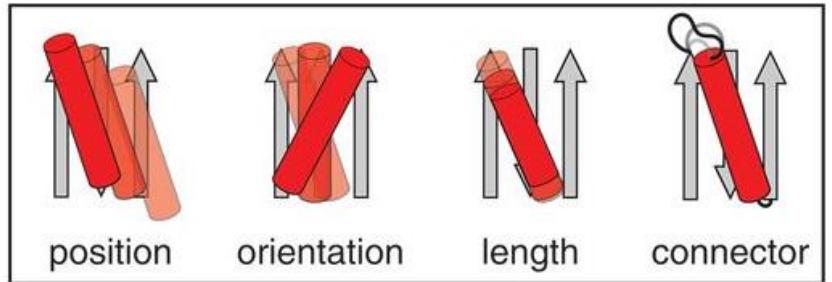
Backbone generation



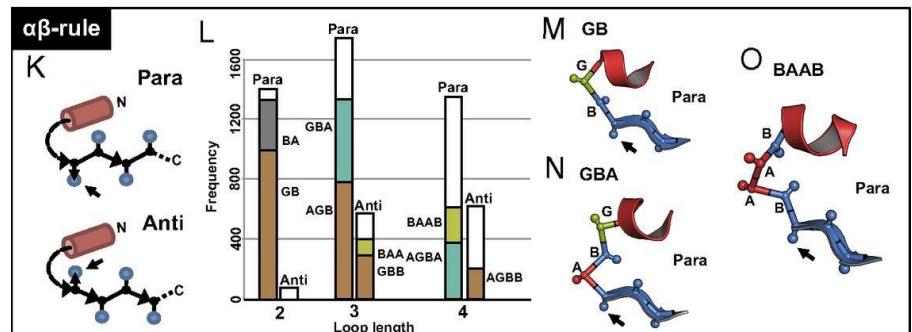
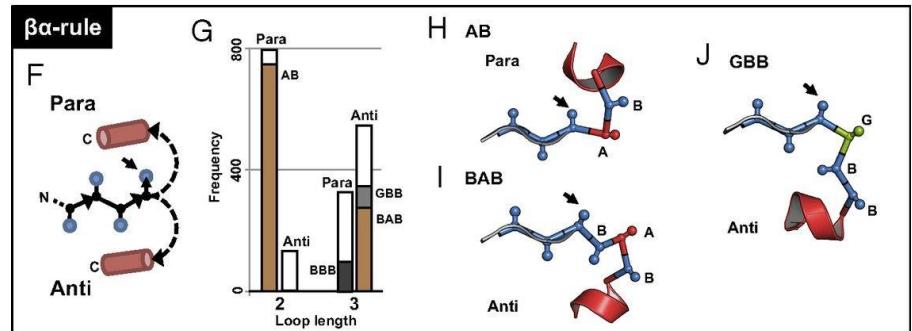
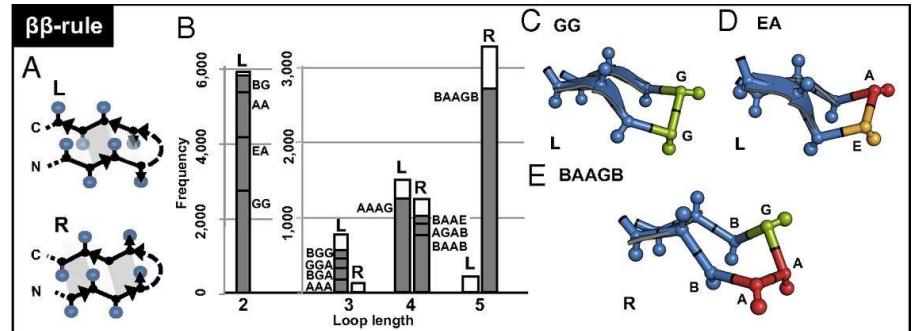
**Generate “designable” sequence-free protein backbones**

# Designable backbones: new structures incorporating native-like features

## 1. Side-chain packing: knob-into-hole

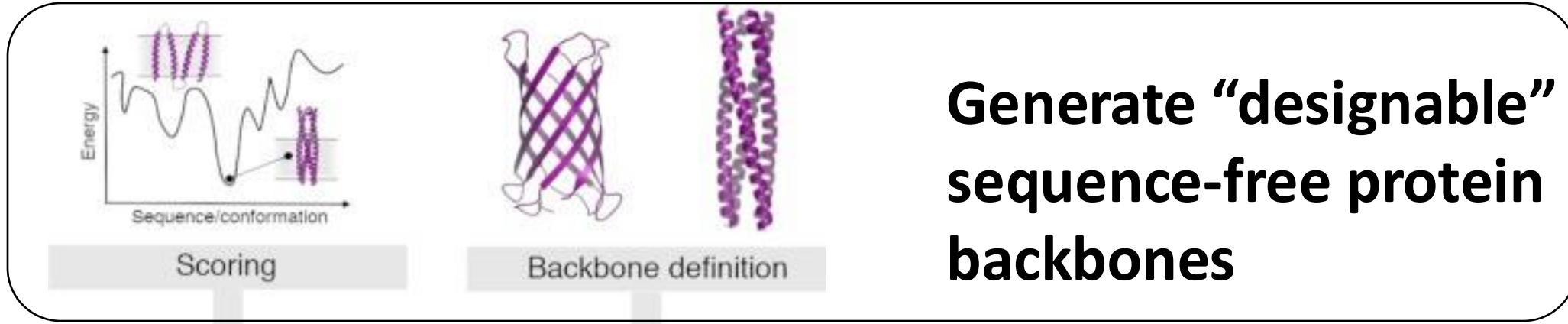


## 2. Loops with preferred geometries

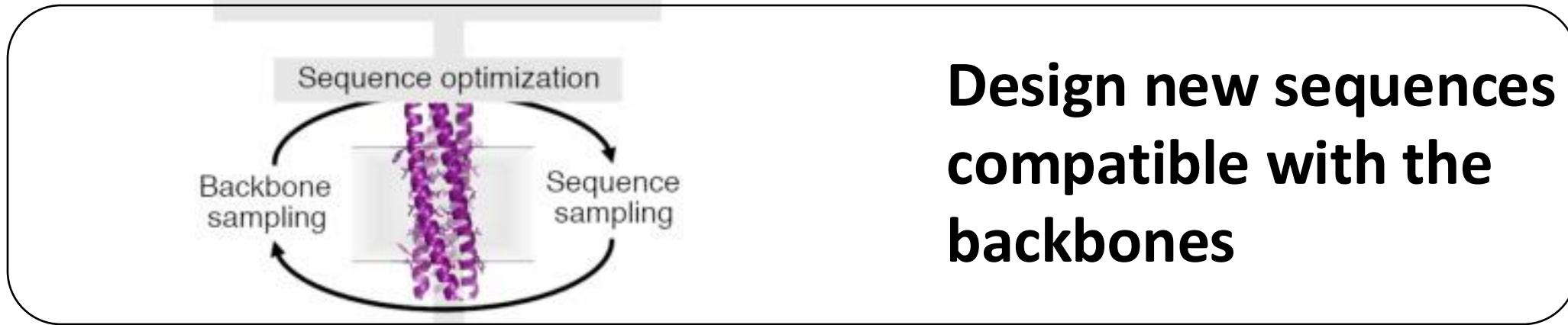


# *De novo* design pipeline

Backbone generation

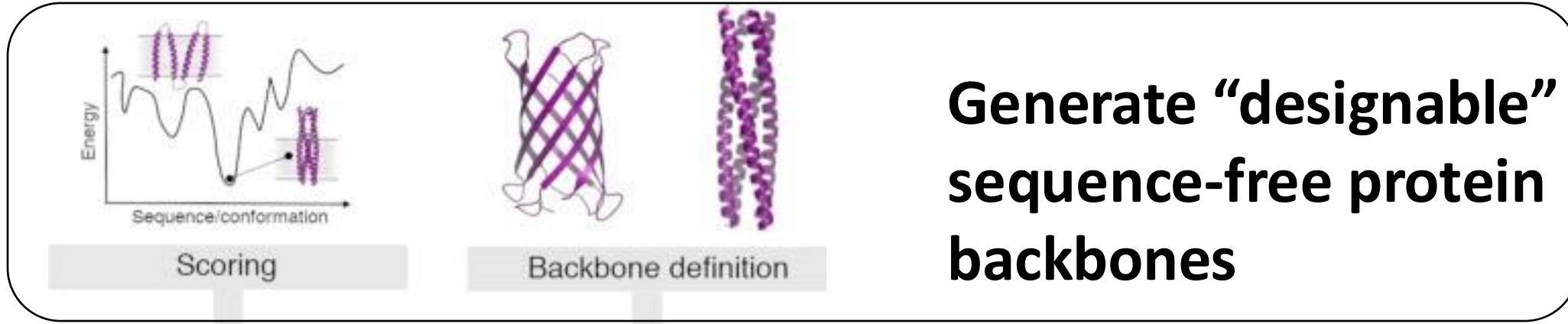


Sequence design

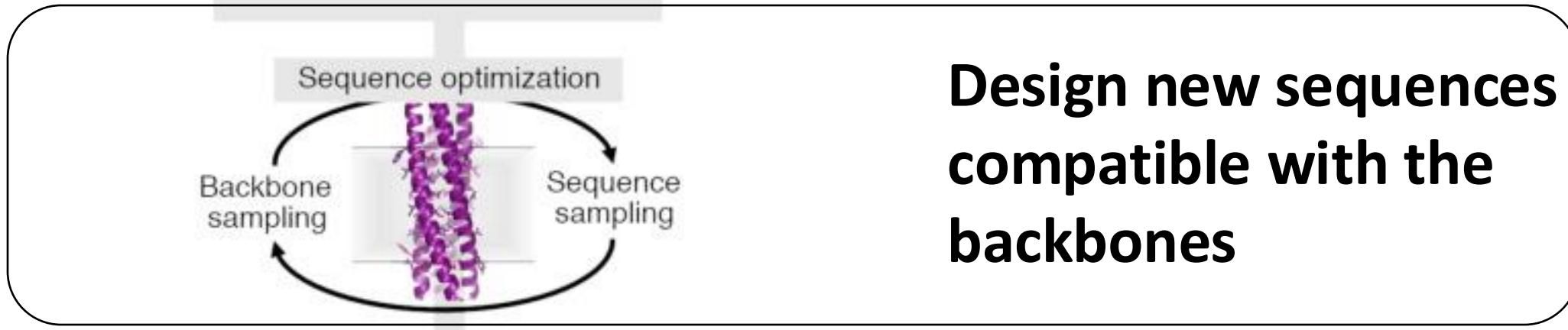


# *De novo* design pipeline

Backbone generation



Sequence design

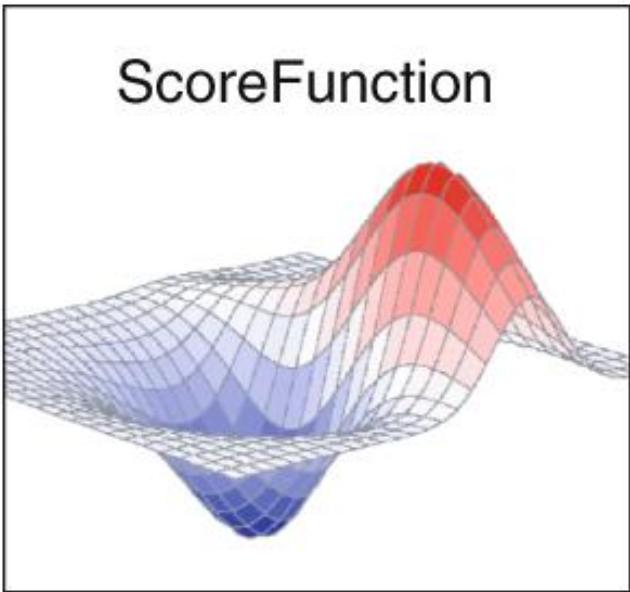


**Reminder: Two major components of physics based models?**

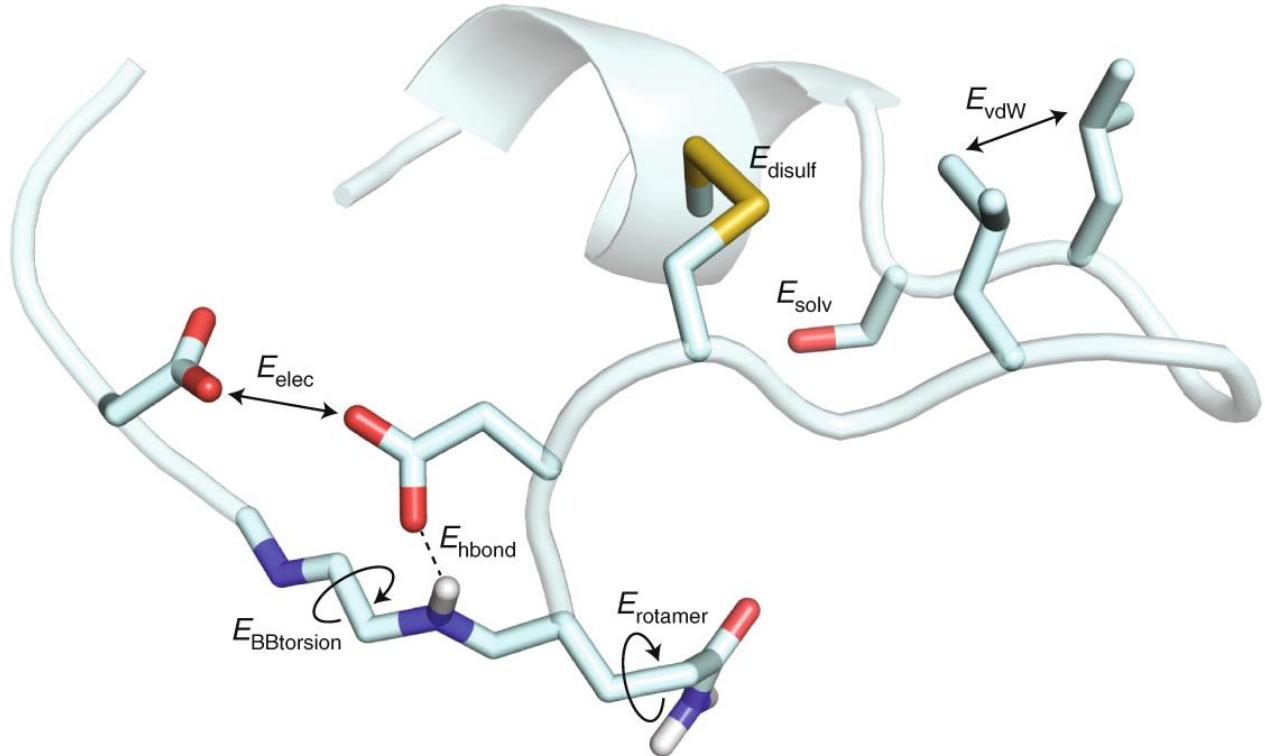
# Scoring – energy functions

weight

$$\Delta E_{\text{total}} = \sum_i w_i E_i(\Theta_i, \text{aa}_i)$$



- $E_{\text{vdW}}$  Lennard–Jones for attractive or repulsive interaction  
 $E_{\text{hbond}}$  Hydrogen bonding allows buried polar atoms  
 $E_{\text{elec}}$  Electrostatic interaction between charges  
 $E_{\text{disulf}}$  Disulfide bonds between cysteines



- $E_{\text{solv}}$  Implicit solvation model penalizes buried polar atoms  
 $E_{\text{BBtorsion}}$  Backbone torsion preferences from main-chain potential  
 $E_{\text{rotamer}}$  Side-chain torsion angles from rotamer library  
 $E_{\text{ref}}$  Unfolded state reference energy for design

# Sampling methods

Sampling residue type and rotamers

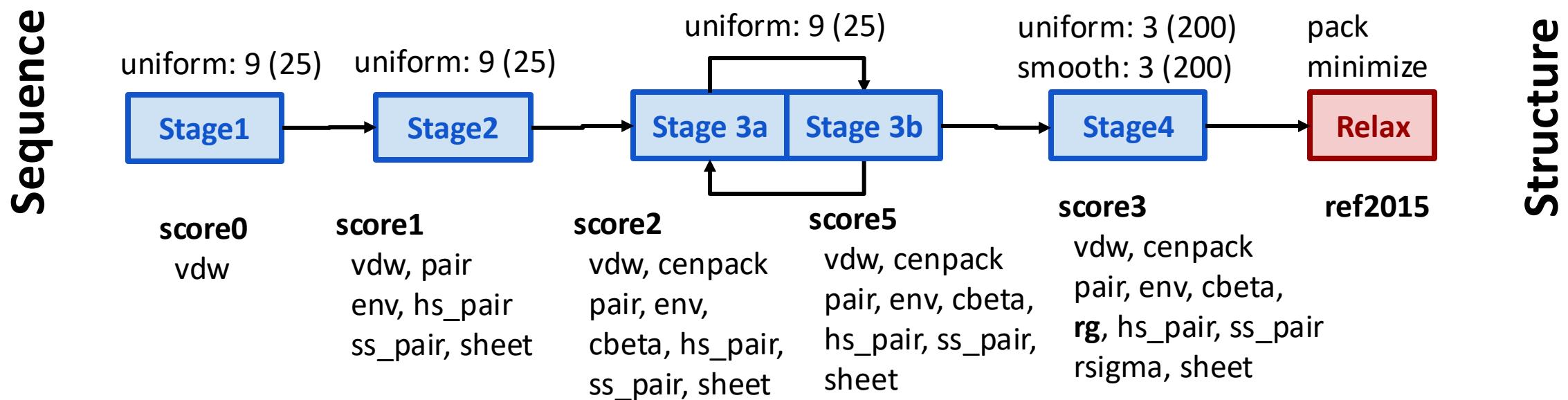
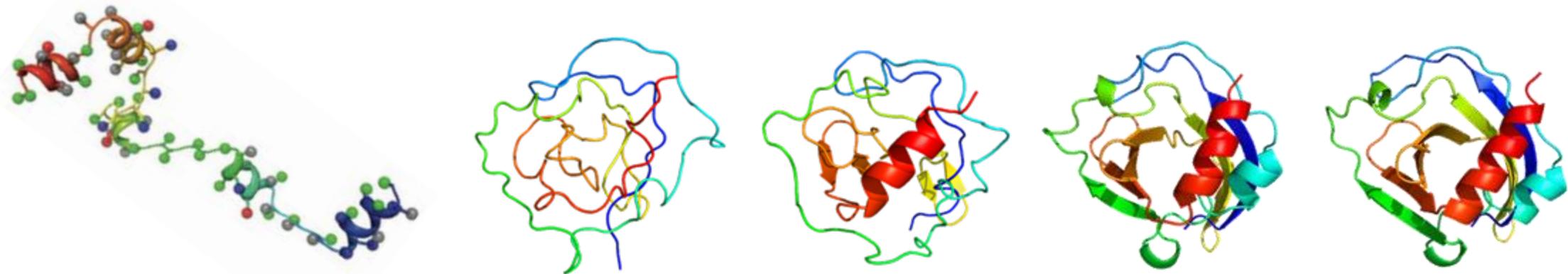


# Analogy: avalanche rescue search

Your search strategy will depend on your objectives and of the landscape



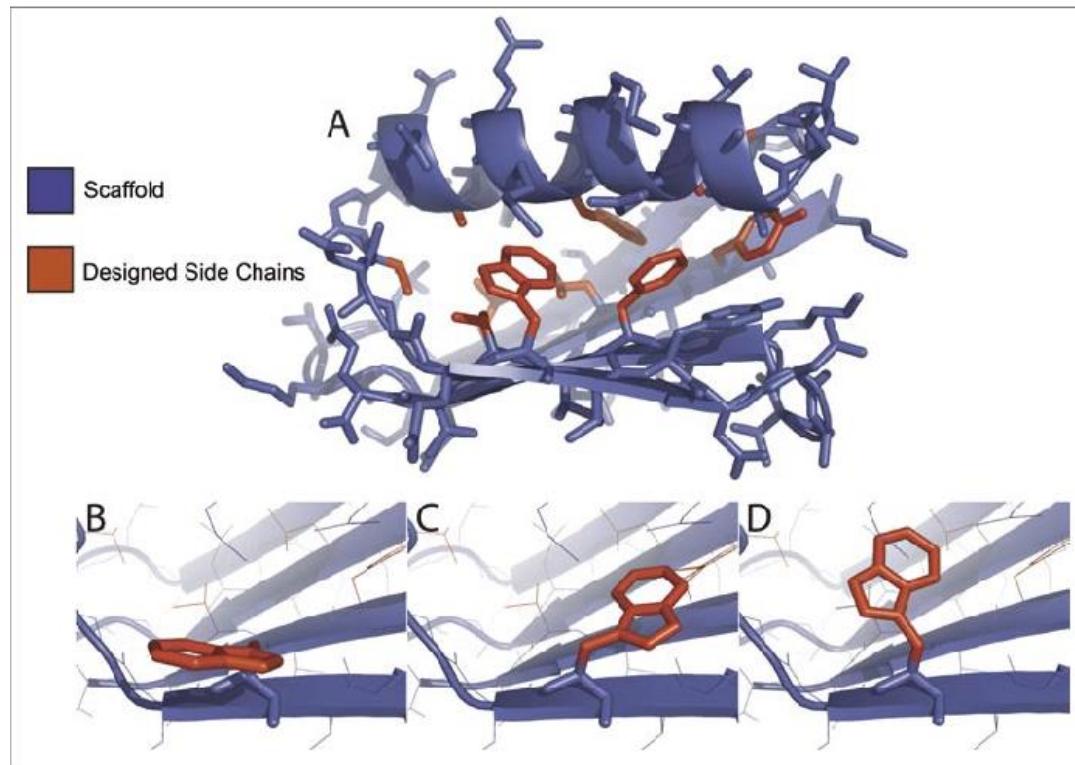
# Reminder: *Ab initio* structure prediction



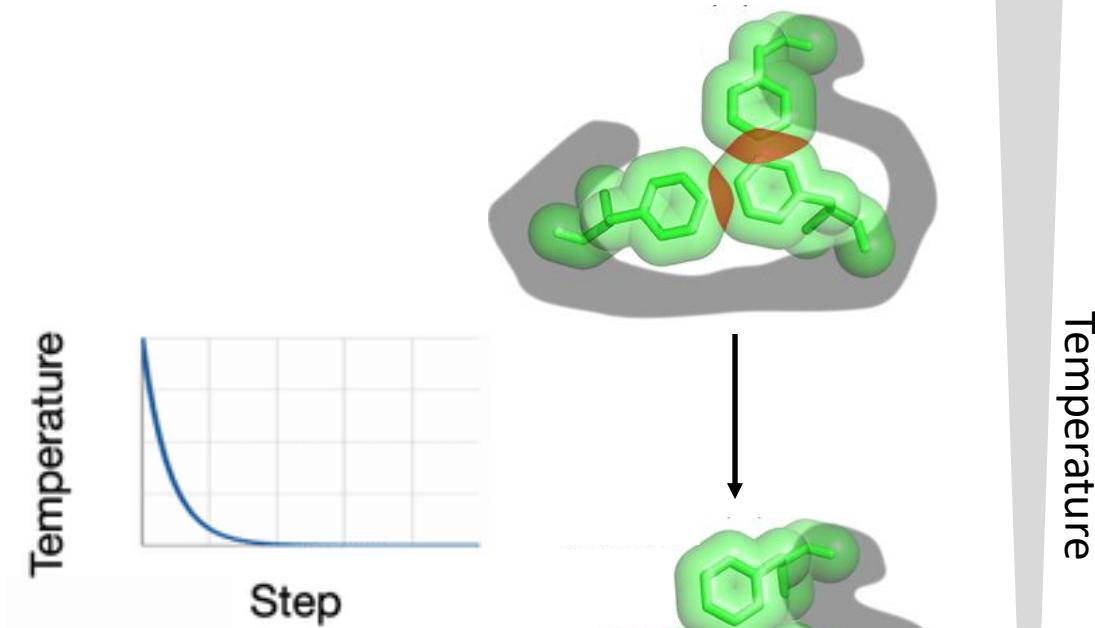
# Sampling methods

## Sampling residue type and rotamers

- Simulated annealer – Rosetta design



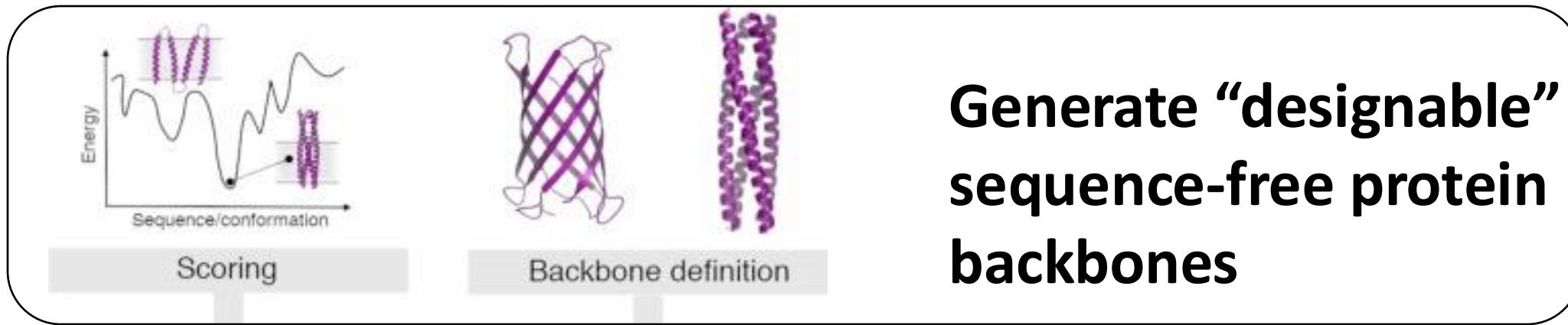
Low repulsive energy –  
many moves are allowed



High repulsive energy –  
less moves are allowed

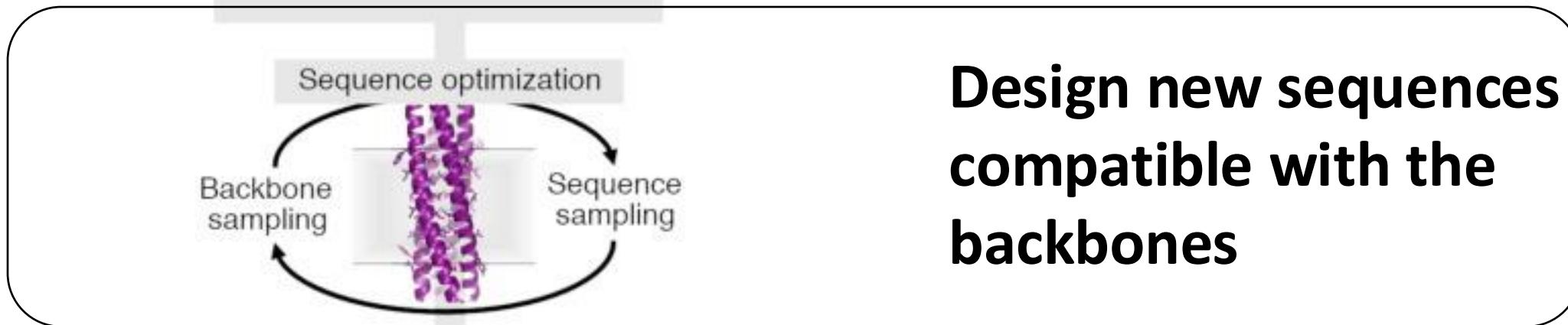
# *De novo* design pipeline

Backbone generation



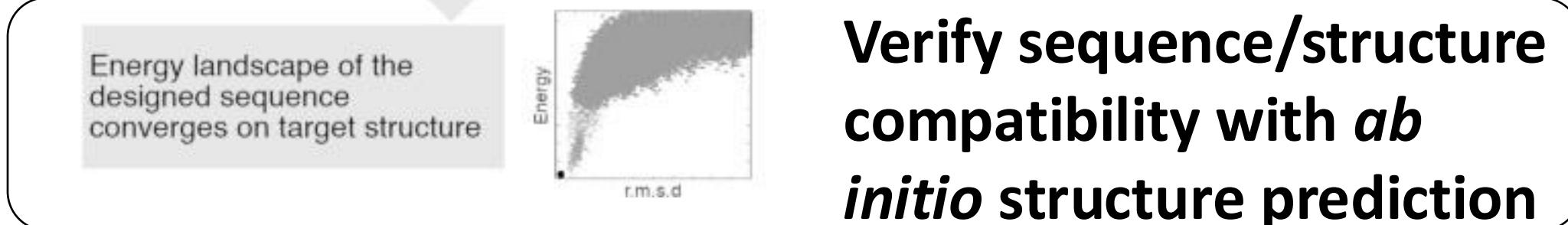
**Generate “designable” sequence-free protein backbones**

Sequence design



**Design new sequences compatible with the backbones**

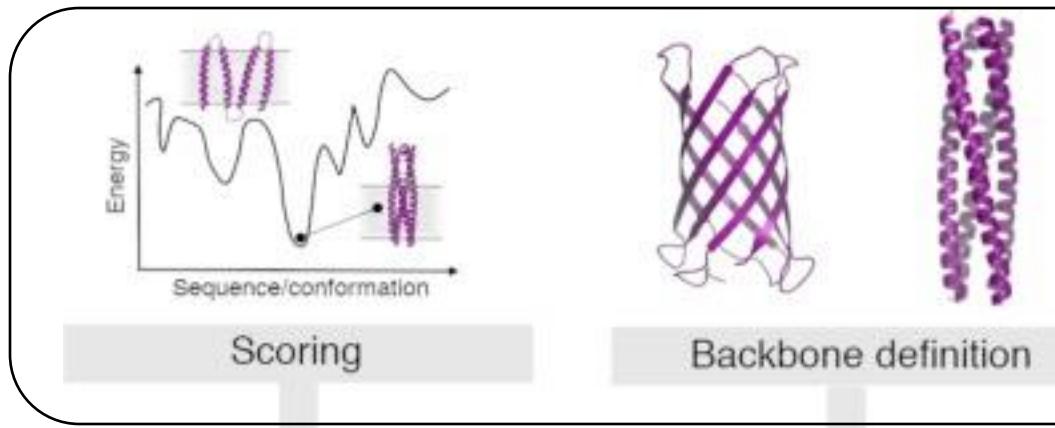
Validation



**Verify sequence/structure compatibility with *ab initio* structure prediction**

# *De novo* design pipeline

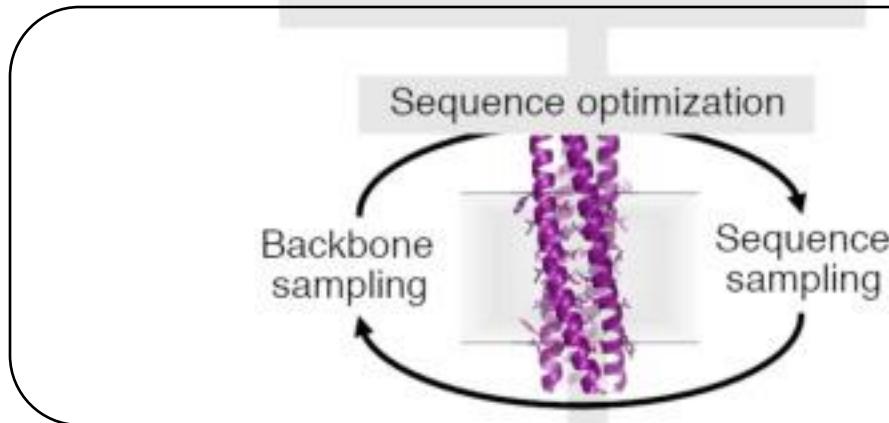
Backbone generation



**Generate “designable” sequence-free protein backbones**

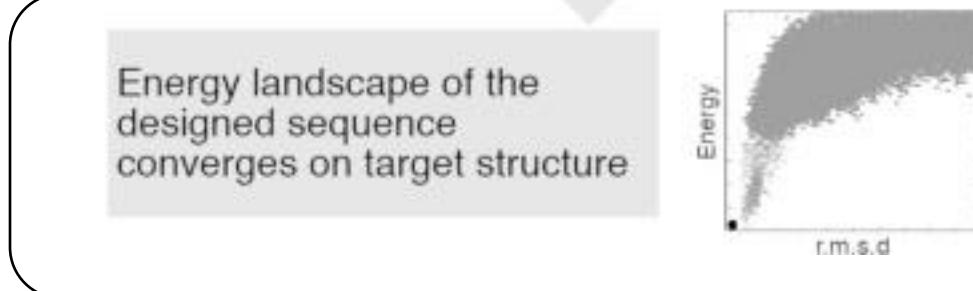


Sequence design



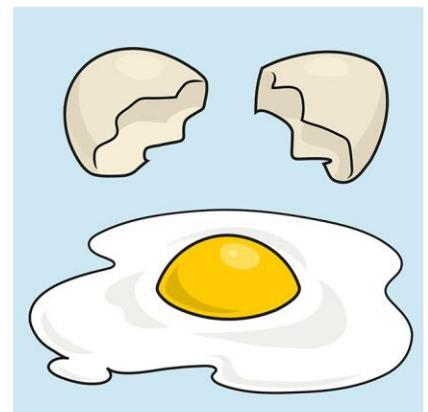
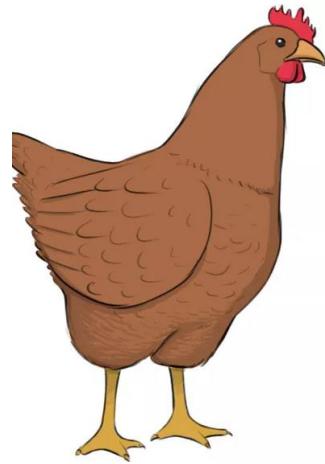
**Design new sequences compatible with the backbones**

Validation

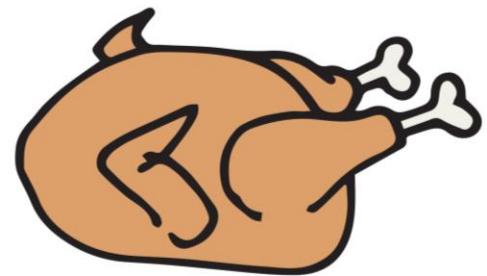


**Verify sequence/structure compatibility with *ab initio* structure prediction**

# *De novo* protein design: chicken/egg problem?

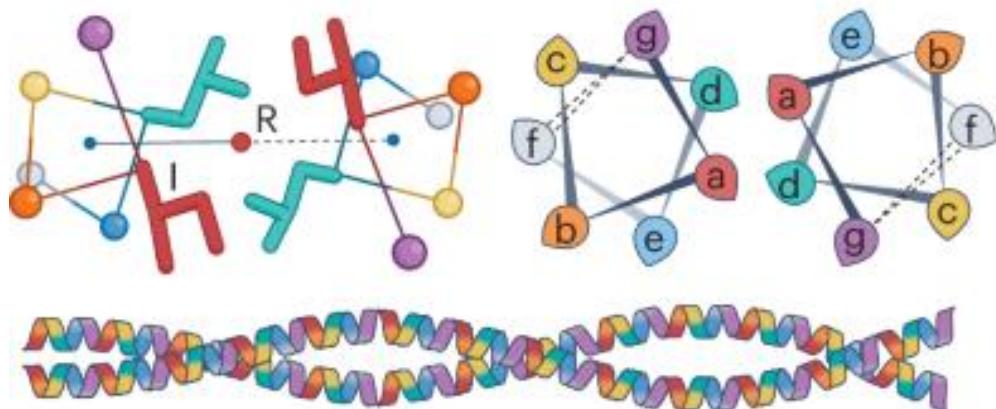


*De novo* protein design: chicken/egg problem?



# Parametric design of alpha-helical bundles

Knob-into-hole packing of side-chains



Crick coiled coil equations

For a helix in a coiled coil with supercoil axis along z, the Cartesian coordinates of the Calpha of residue  $t$  are given by:

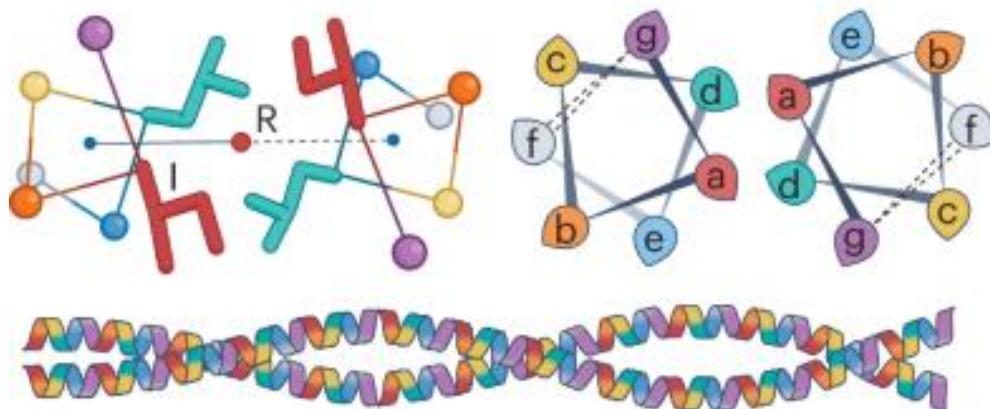
$$x = R_0 \cos(\omega_0 t + \varphi_0') + R_1 \cos(\omega_0 t + \varphi_0') \cos(\omega_1 t + \varphi_1) - R_1 \cos(\alpha) \sin(\omega_0 t + \varphi_0') \sin(\omega_1 t + \varphi_1)$$

$$y = R_0 \sin(\omega_0 t + \varphi_0') + R_1 \sin(\omega_0 t + \varphi_0') \cos(\omega_1 t + \varphi_1) + R_1 \cos(\alpha) \cos(\omega_0 t + \varphi_0') \sin(\omega_1 t + \varphi_1)$$

$$z = (\omega_0 R_0 / \tan(\alpha)) t - R_1 \sin(\alpha) \sin(\omega_1 t + \varphi_1) + \Delta z$$

# Parametric design of alpha-helical bundles

Knob-into-hole packing of side-chains

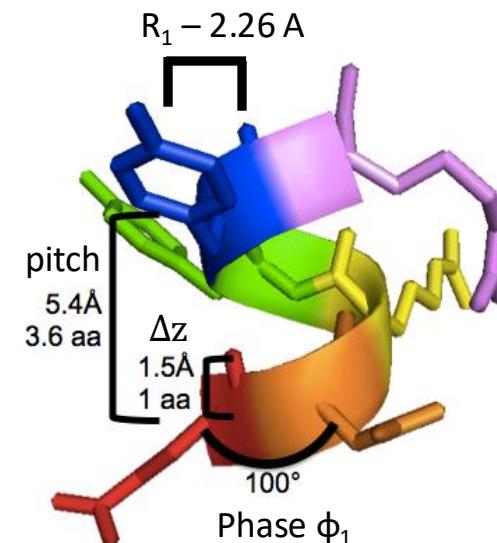


Crick coiled coil equations

For a helix in a coiled coil with supercoil axis along z, the Cartesian coordinates of the Calpha of residue  $t$  are given by:

$$x = R_0 \cos(\omega_0 t + \varphi_0) + R_1 \cos(\omega_0 t + \varphi_0) \cos(\omega_1 t + \varphi_1) - R_1 \cos(\alpha) \sin(\omega_0 t + \varphi_0) \sin(\omega_1 t + \varphi_1)$$
$$y = R_0 \sin(\omega_0 t + \varphi_0) + R_1 \sin(\omega_0 t + \varphi_0) \cos(\omega_1 t + \varphi_1) + R_1 \cos(\alpha) \cos(\omega_0 t + \varphi_0) \sin(\omega_1 t + \varphi_1)$$
$$z = (\omega_0 R_0 / \tan(\alpha)) t - R_1 \sin(\alpha) \sin(\omega_1 t + \varphi_1) + \Delta z$$

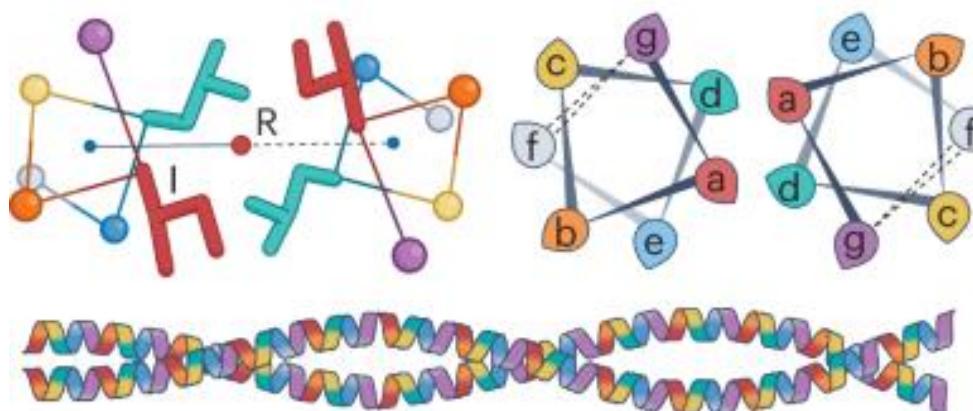
$\omega_1$  – helical twist



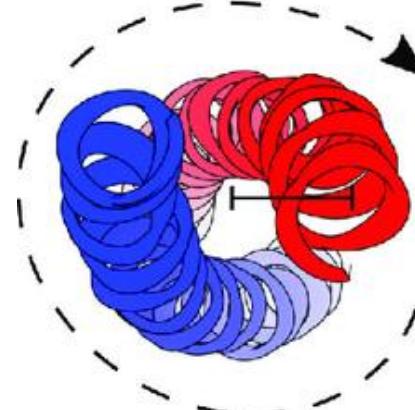
<https://doi.org/10.1016/j.jbc.2023.104579>

# Parametric design of alpha-helical bundles

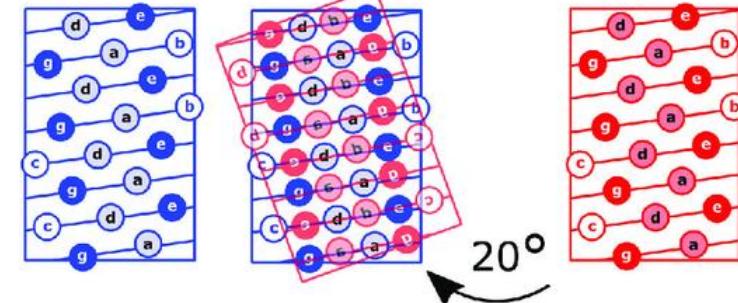
## Knob-into-hole packing of side-chains



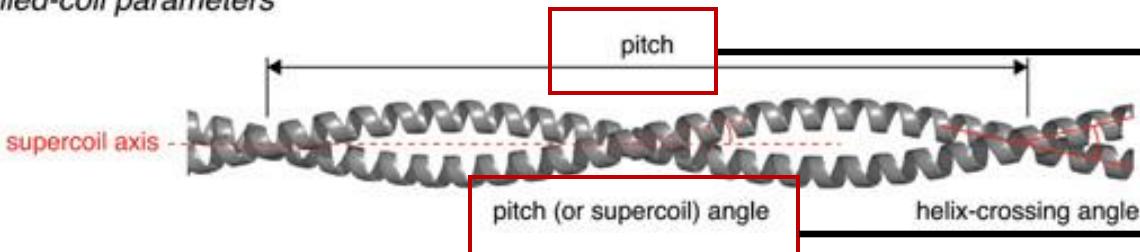
## Radius



## Knobs into holes packing



## Coiled-coil parameters



Number of residues to bring the helices back to sync.

Angle between helix and supercoil axis

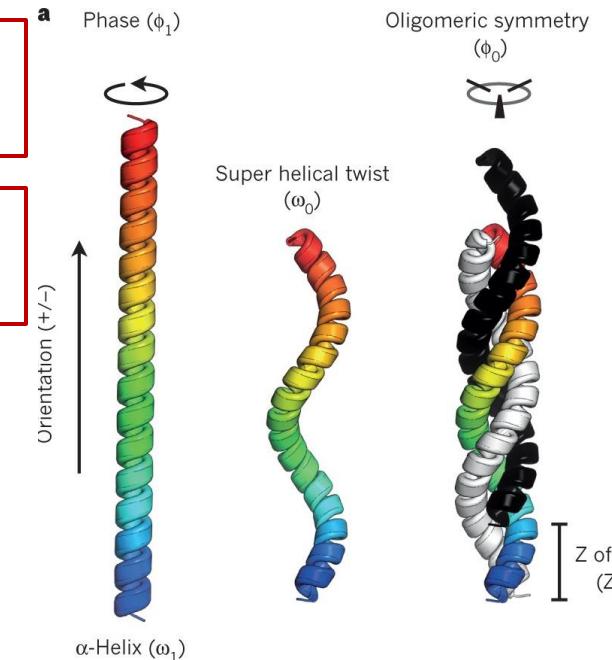
## Crick coiled coil equations

For a helix in a coiled coil with supercoil axis along z, the Cartesian coordinates of the Calpha of residue t are given by:

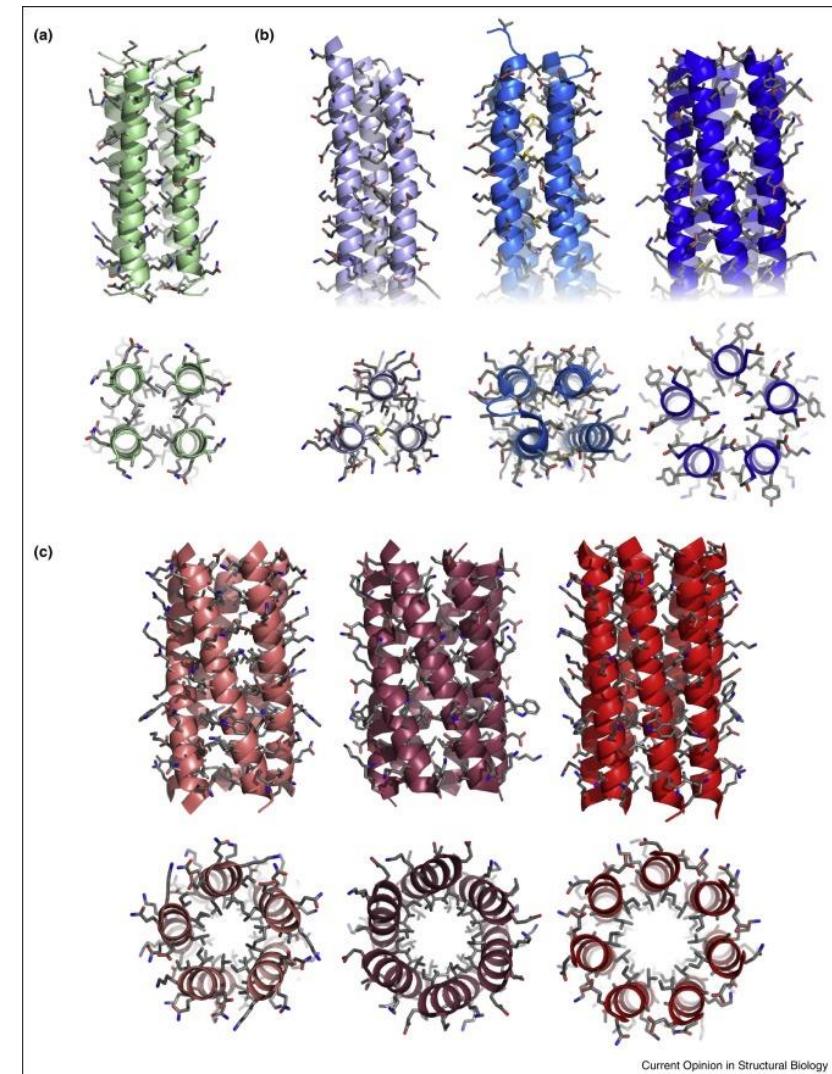
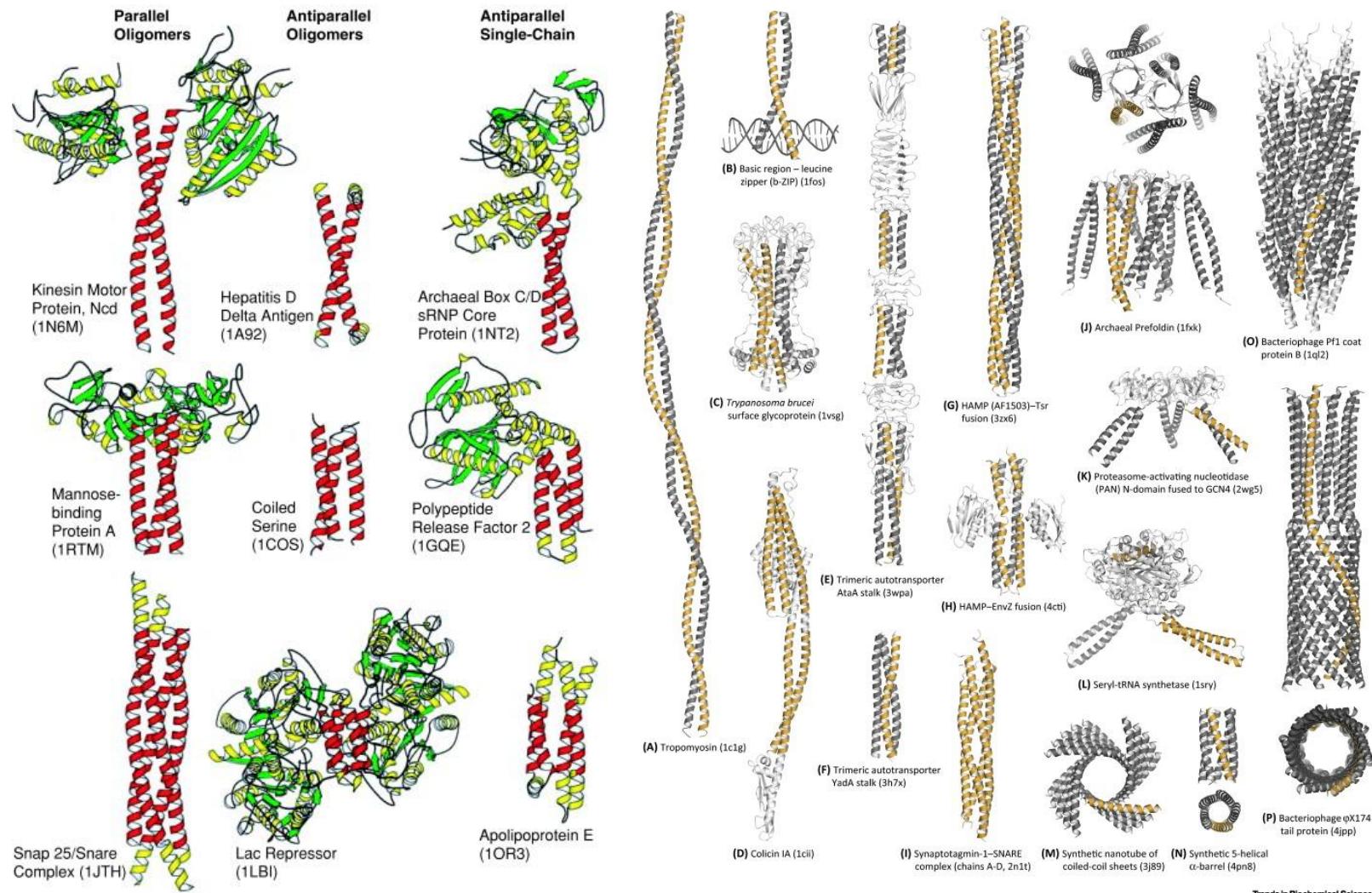
$$x = R_0 \cos(\omega_0 t + \phi_0') + R_1 \cos(\omega_0 t + \phi_0') \cos(\omega_1 t + \phi_1) - R_1 \cos(\alpha) \sin(\omega_0 t + \phi_0') \sin(\omega_1 t + \phi_1)$$

$$y = R_0 \sin(\omega_0 t + \phi_0') + R_1 \sin(\omega_0 t + \phi_0') \cos(\omega_1 t + \phi_1) + R_1 \cos(\alpha) \cos(\omega_0 t + \phi_0') \sin(\omega_1 t + \phi_1)$$

$$z = (\omega_0 R_0 / \tan(\alpha)) t - R_1 \sin(\alpha) \sin(\omega_1 t + \phi_1) + \Delta z$$



# Natural vs designed alpha-helical bundles



*De novo* designed coiled coils and bundles

Natural helical bundles and coiled-coil domains hidden in proteins

# Generating coiled coils with ideal parameters - CCCP

grigoryanlab.org/cccp/

The screenshot shows the CCCP (Coiled-coil Crick Parameterization) website. At the top, there's a red header bar with the text "CCCP (Coiled-coil Crick Parameterization)". Below it, a main content area has a red sidebar with the same title and a brief description: "A suite of tools for fitting Crick parameters<sup>1,2</sup> for coiled-coil structures and generating structures based on parameters. Contact gevorg.grigoryan at gmail dot com with bug reports and requests." It lists four tools: Structure Fitter, Structure Generator, Accommodation Index Analyzer, and Matlab/Octave source code. Below this, two references are cited: F. H. Crick, "The Fourier Transform of a Coiled Coil", *Acta Cryst.*, **6**: 685 (1953) and G. Grigoryan, W. F. DeGrado, "Probing Designability via a Generalized Model of Helical Bundle Geometry", *J. Mol. Biol.*, **405**(4): 1079-1100 (2011). At the bottom, there are links to "HOME | Research | Publications | People | Links | Contact @ Grigoryan Lab" and information about the Dartmouth College location and departments.

**CCCP (Coiled-coil Crick Parameterization)**  
This program generates coiled-coil structures based on specified Crick parameters<sup>[1-2]</sup>. Multiple structures can be generated that span a range of parameters. Contact gevorg.grigoryan at gmail dot com with bug reports and requests.

Number of chains:

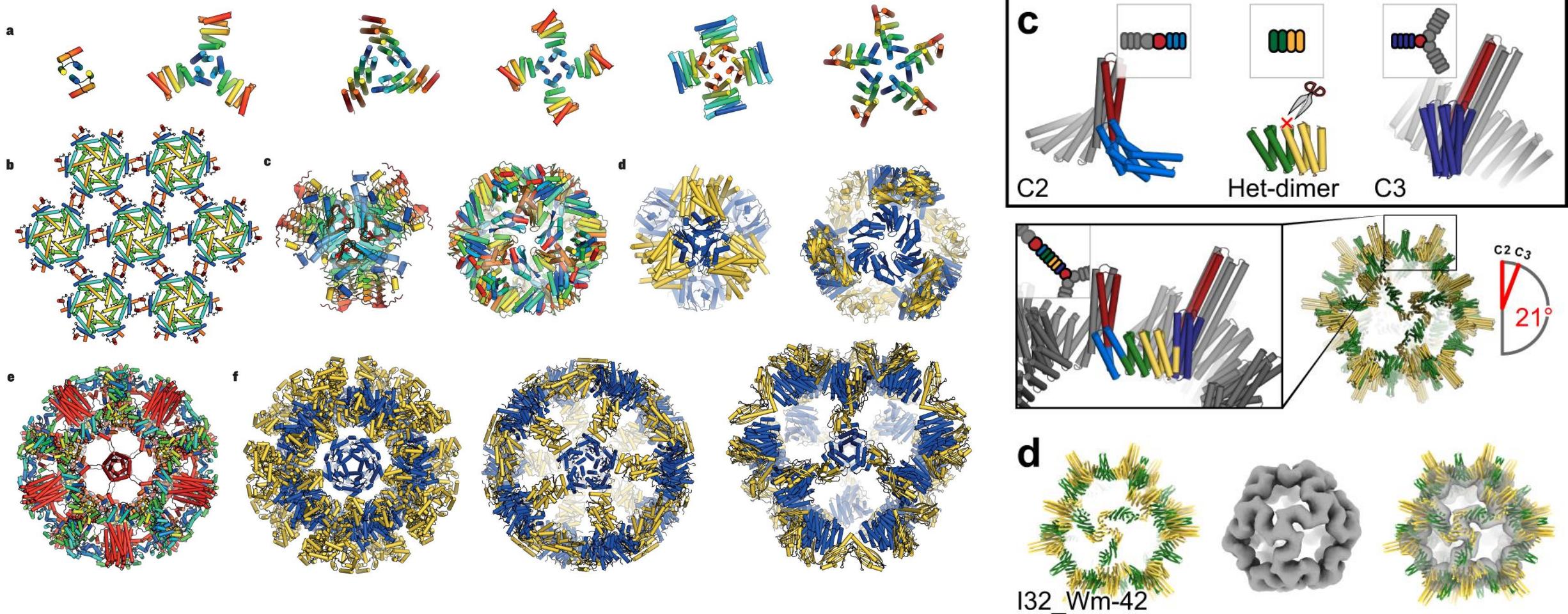
Chain length:

---

**Coiled-coil parameters**

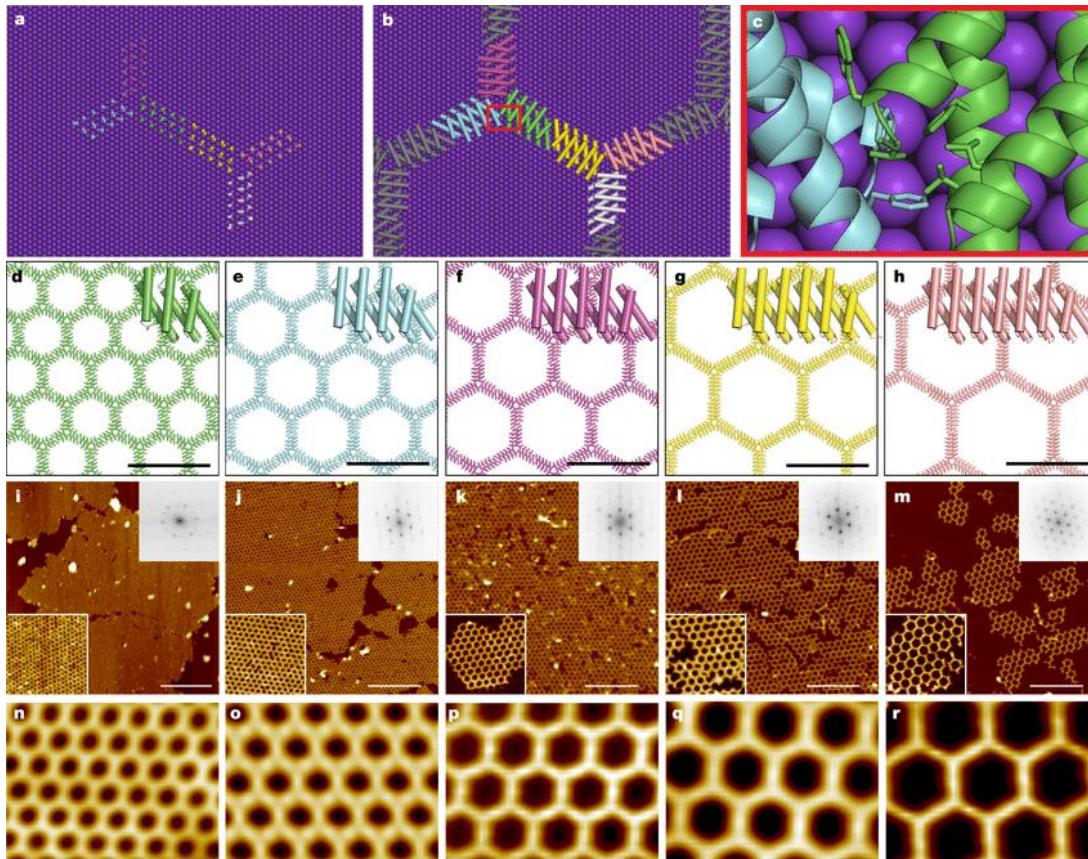
Parameter (symbol, unit)	Star	End	# Samples
Rise per residue ( $d$ , Å):	1.51	1.51	<input checked="" type="checkbox"/> constrain*
Superhelical radius ( $R_0$ , Å):	5.00	5.00	<input type="radio"/> adjust**
Superhelical frequency ( $\omega_0$ , °/aa):	-3.6	-3.6	<input checked="" type="radio"/> adjust**
(negative means left-handed superhelix)			<input type="checkbox"/> vary together
Pitch angle ( $\alpha$ , °):	-12.0	-12.0	<input type="radio"/> adjust**
$\alpha$ -helical radius ( $R_1$ , Å):	2.26	2.26	<input type="radio"/> adjust**
$\alpha$ -helical frequency ( $\omega_1$ , °/aa):	102.8	102.8	<input type="radio"/> adjust**
Symmetry (limits variable parameters):			
<input type="radio"/> C <sub>n</sub>			
<input type="radio"/> D <sub>n</sub>			
<input checked="" type="radio"/> Do not impose symmetry			
<input type="checkbox"/> Forcefield minimize final backbone (CA atoms held fixed)			
<input type="checkbox"/> Create a poly-alanine backbone, not a poly-glycine			

# Example 1: alpha-helical bundles as building units

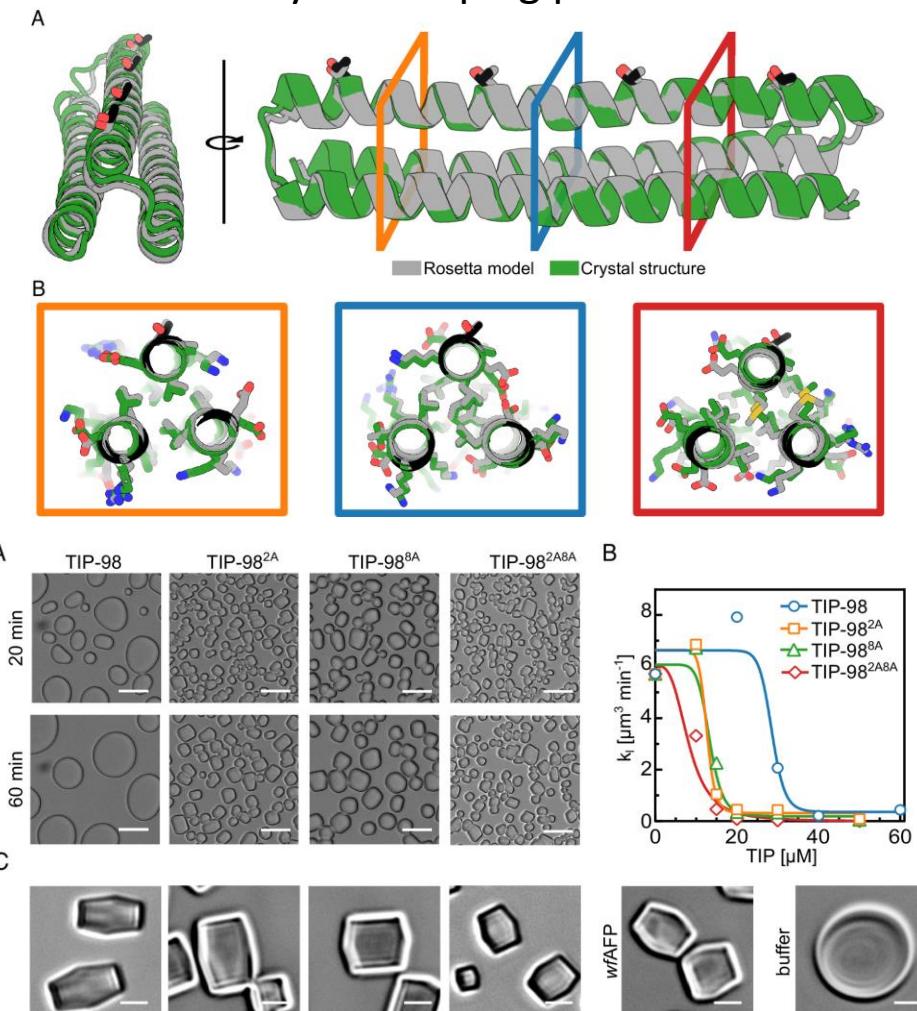


# Example 2: lattice-matching proteins

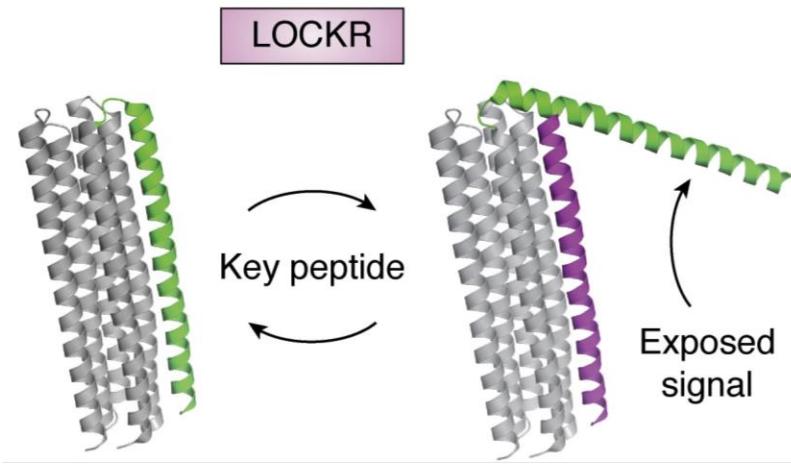
Mica-binding proteins and self-assembling material



Ice crystal-shaping proteins



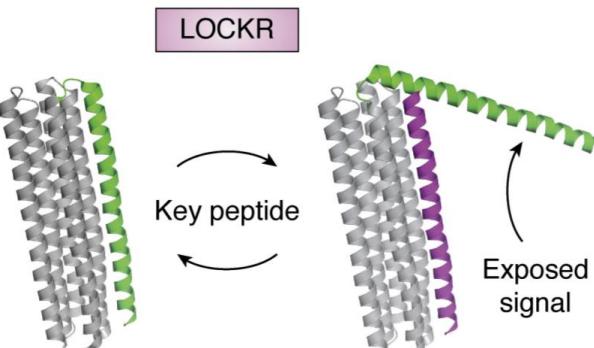
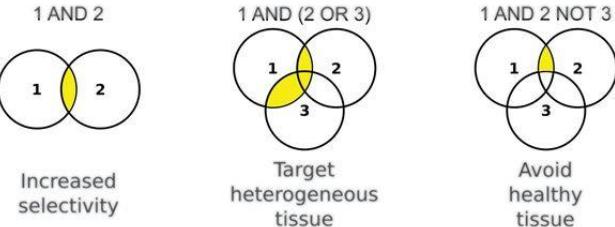
# Example 3: protein logic and switches



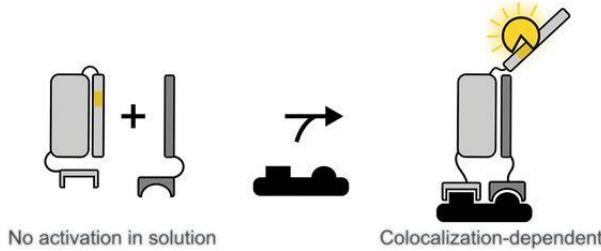
# Example 3: protein logic and switches

## Cell-surface signalling logic

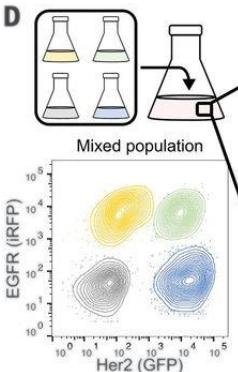
A



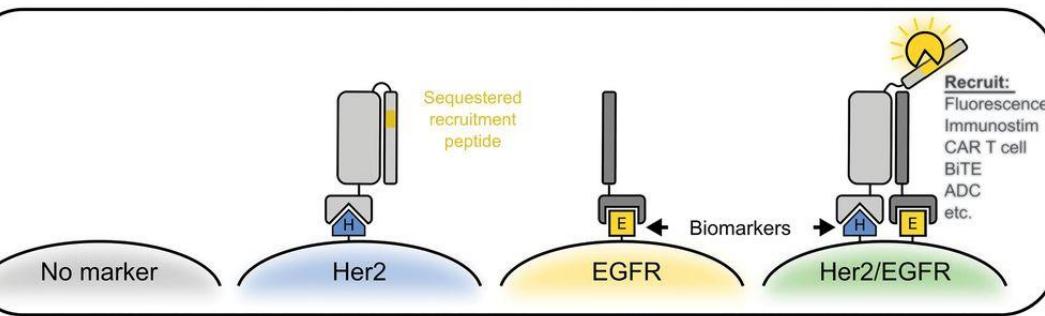
C



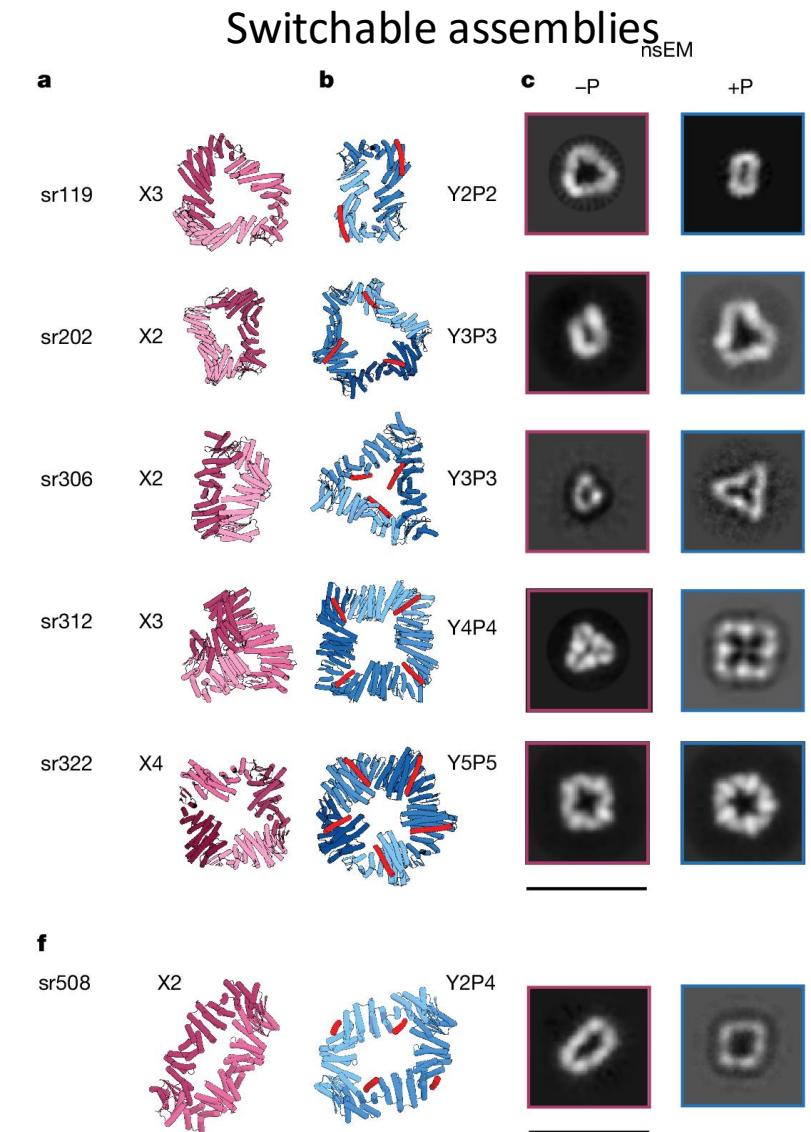
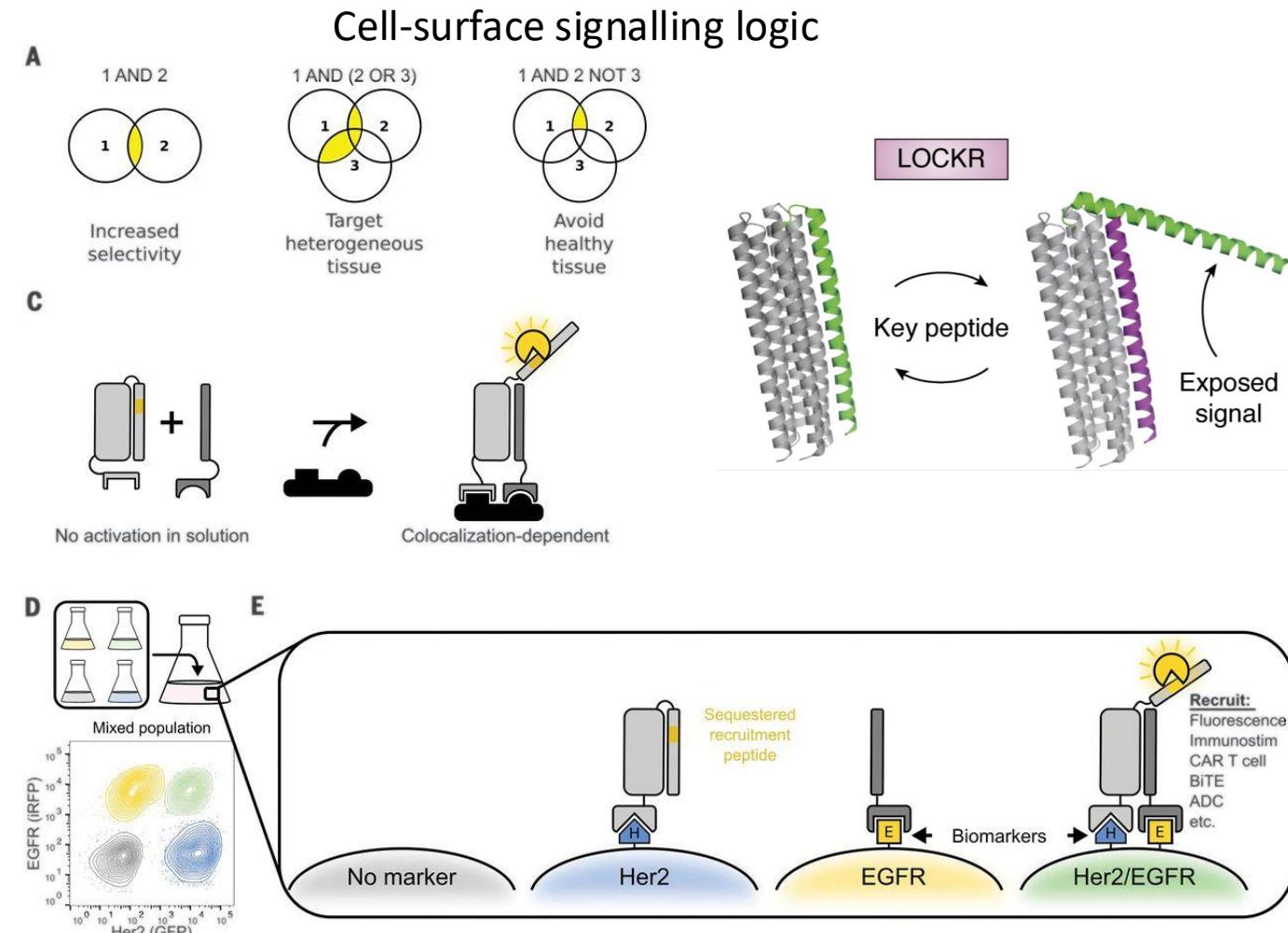
D



E



# Example 3: protein logic and switches

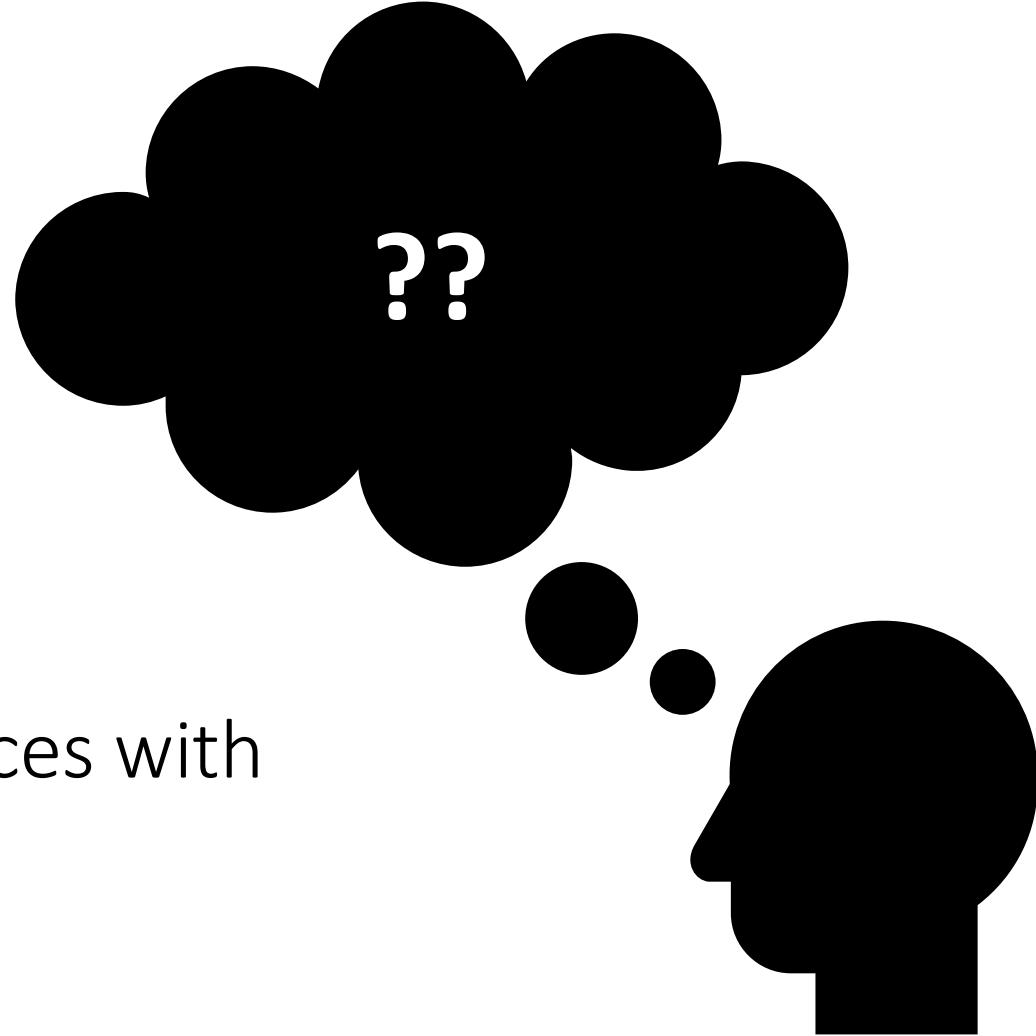


<https://doi.org/10.1038/s41586-021-03258-z>

<https://doi.org/10.1126/science.aba6527>

<https://doi.org/10.1038/s41586-024-07813-2>

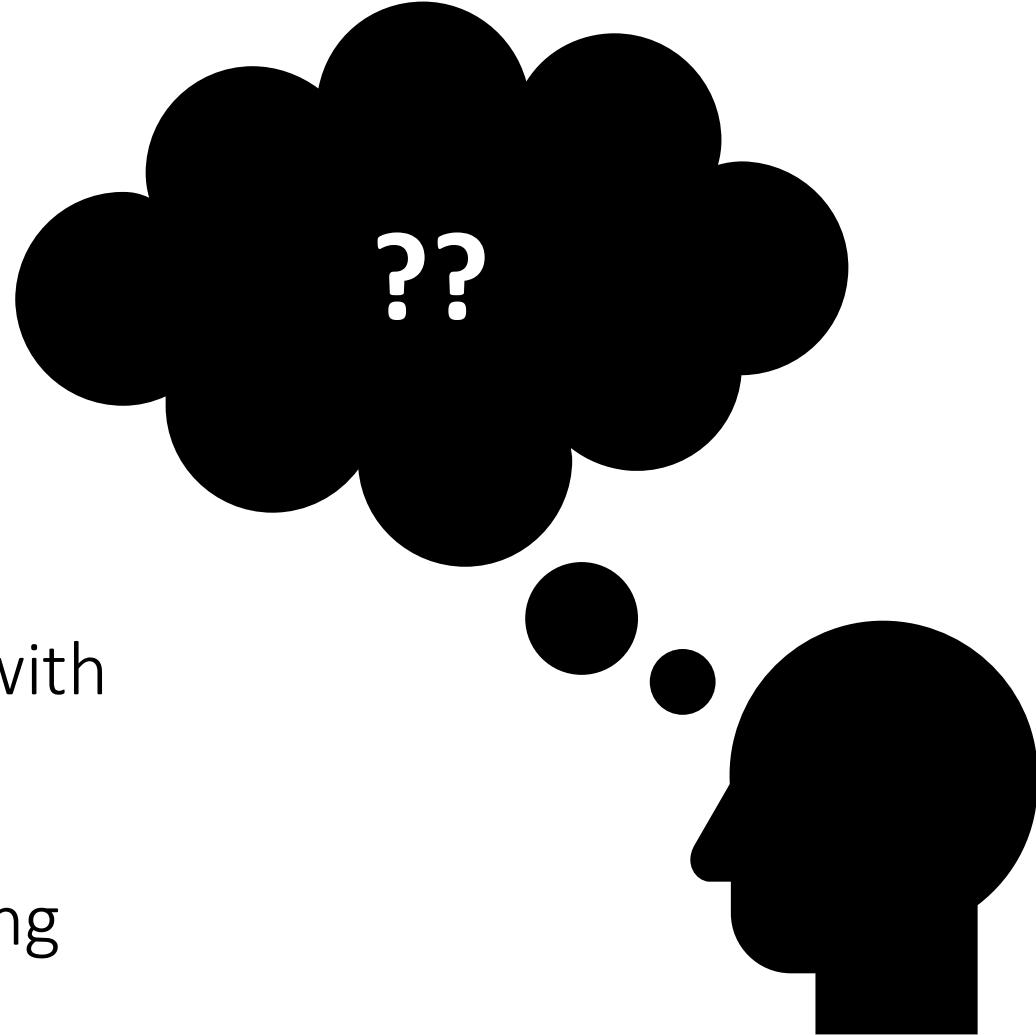
How to generate diverse sequences with  
parametric design?



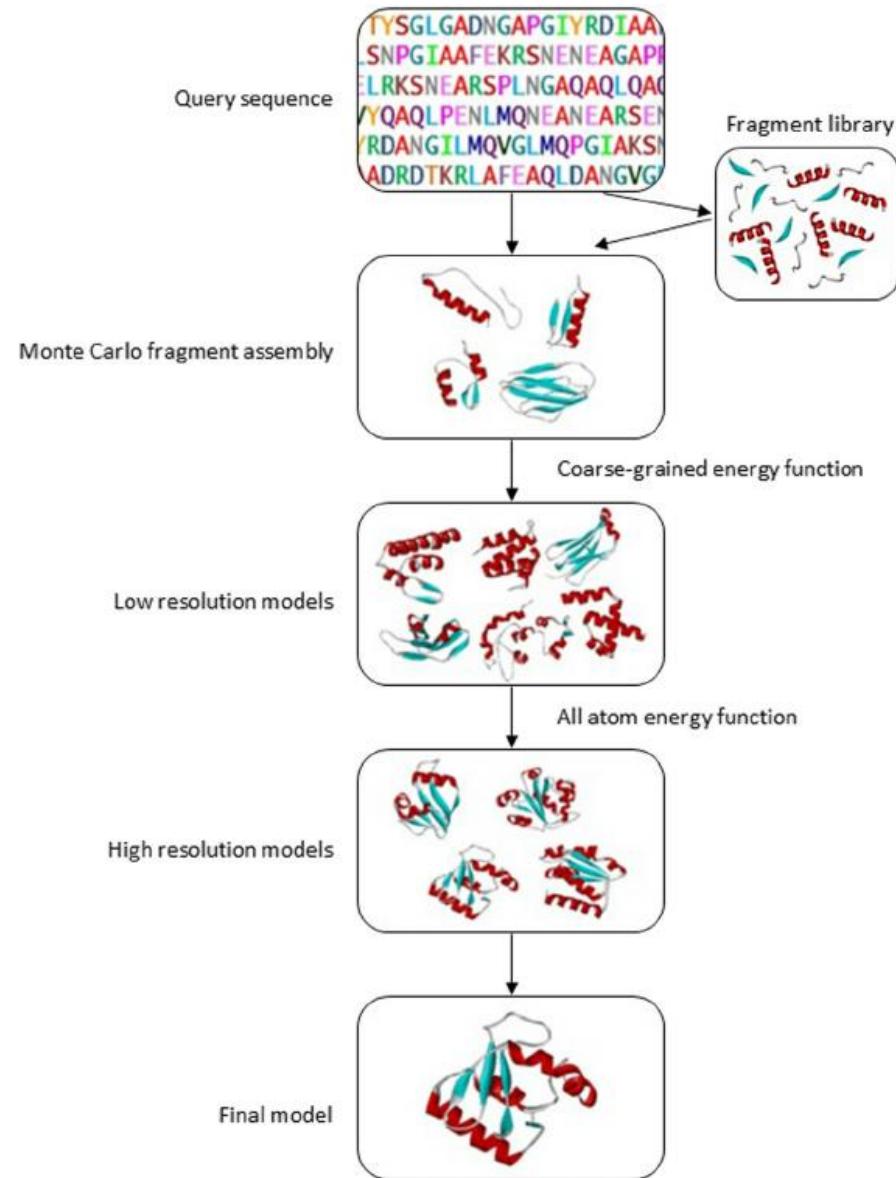
How to generate diverse sequences with parametric design?

→ Superhelical parameters fine-tuning

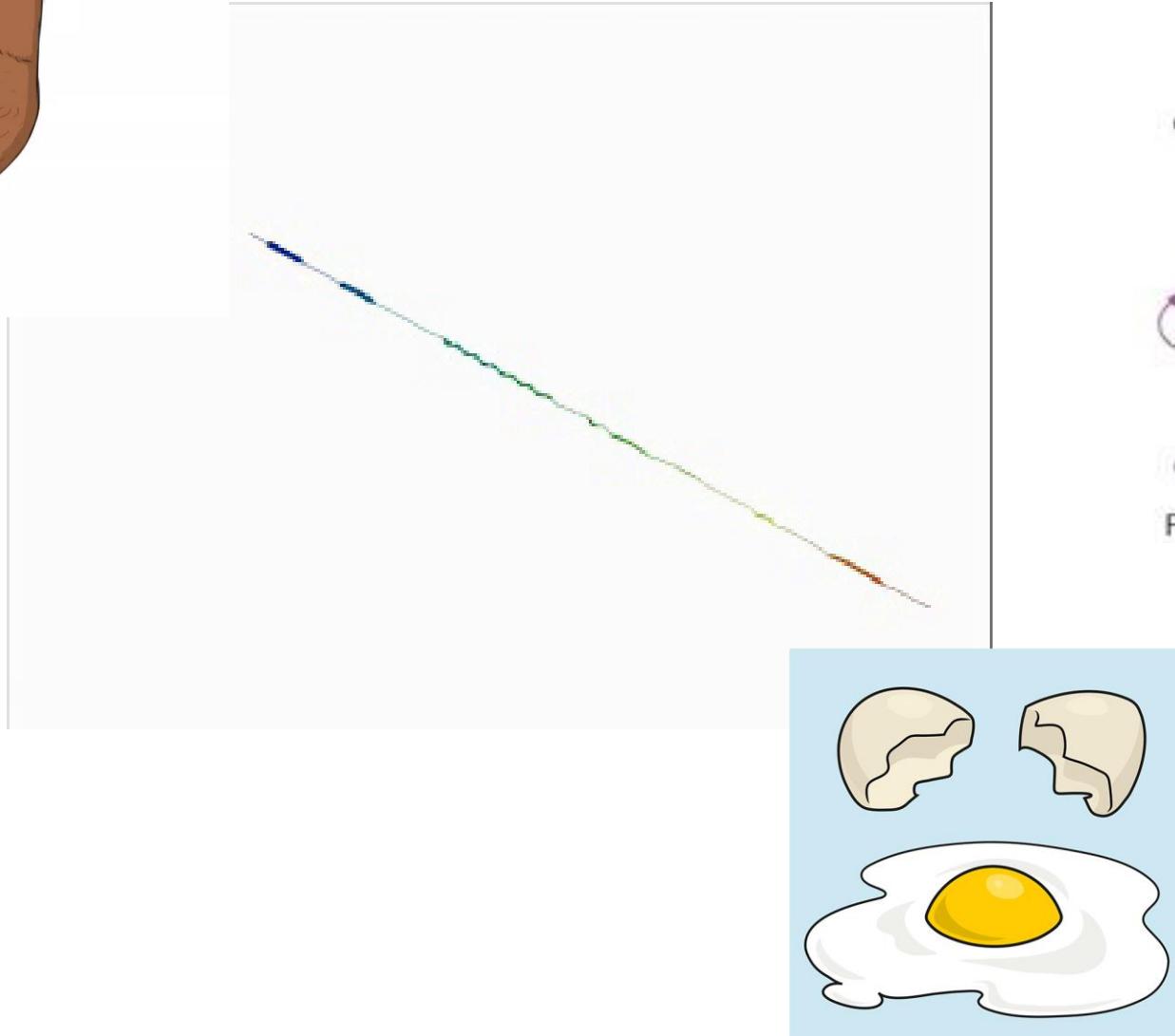
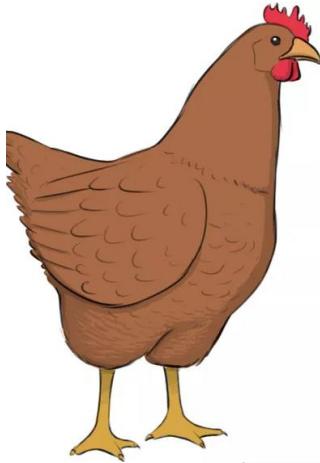
BUT ... parameteric designs mostly unsuccessful for beta-sheet proteins



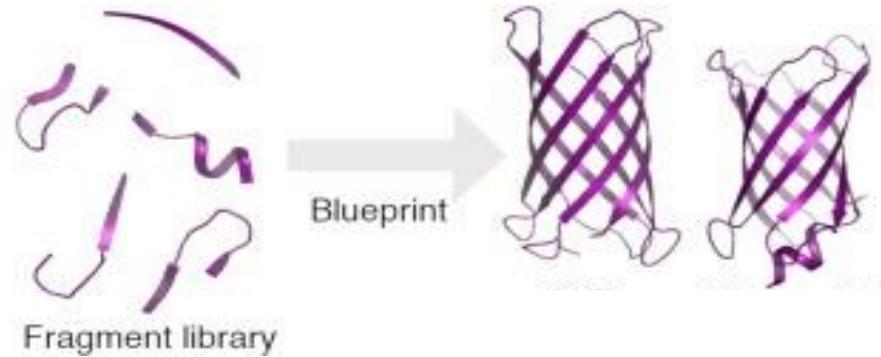
# “Stealing” concepts from structure prediction



# “Stealing” concepts from structure prediction

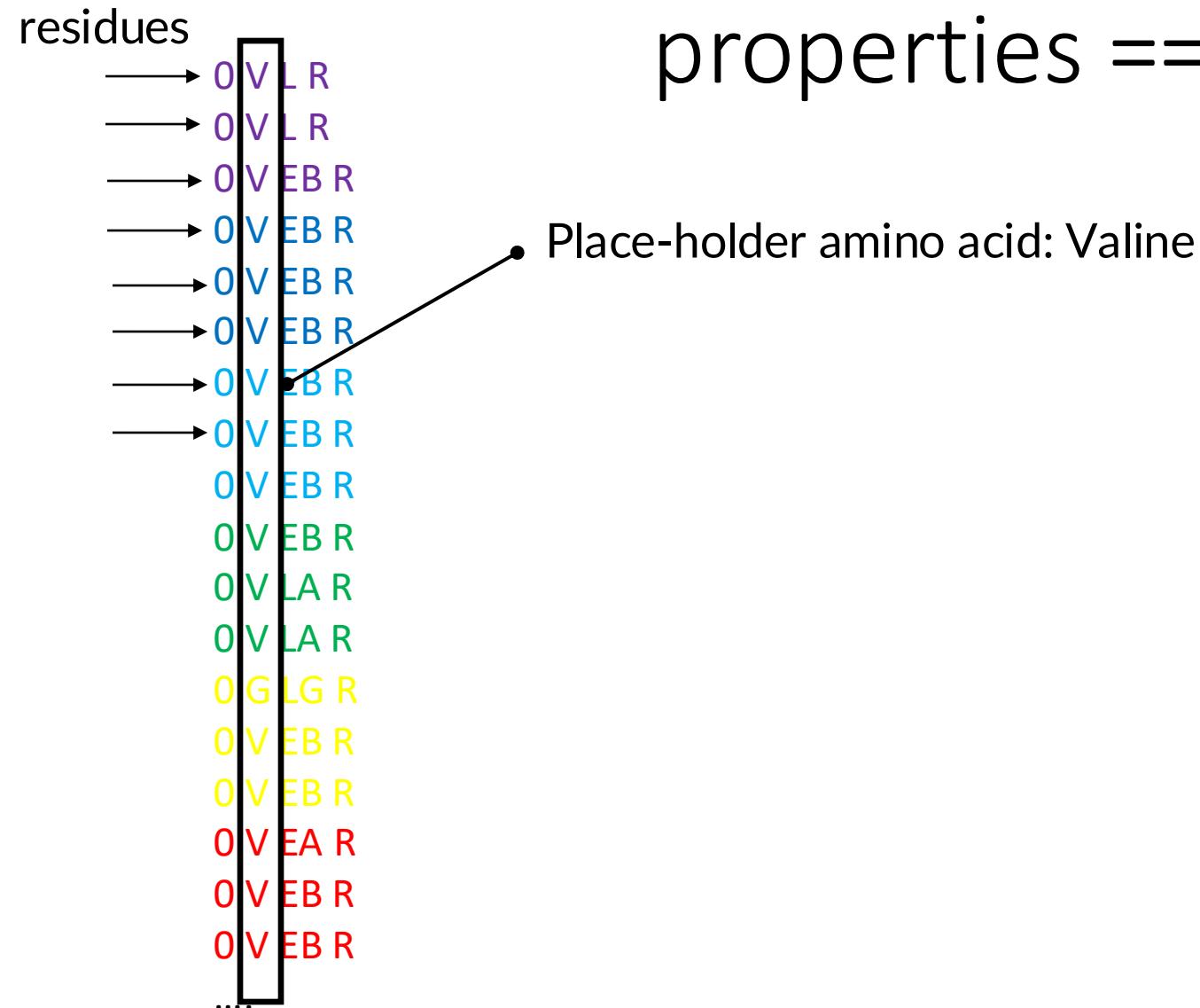


c Fragments-based backbone assembly



How can we pick fragments if  
the sequence is unknown?

# Picking fragments based on local structure properties == Blueprint



Place-holder amino acid: Valine

Blueprint + constraints

# Picking fragments based on local structure properties == Blueprint

residues

→ O V L R  
→ O V L R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V L A R  
→ O V L A R  
→ O G L G R  
→ O V E B R  
→ O V E B R  
→ O V E A R  
→ O V E B R  
→ O V E B R  
....

Place-holder amino acid: Valine

• Secondary structure:  
E = Beta-strand  
H = Alpha-helix  
L = Loop

Blueprint + constraints

# Picking fragments based on local structure properties == Blueprint

residues  
→ O V I R  
→ O V I R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V E B R  
→ O V I A R  
→ O V I A R  
→ O G L G R  
→ O V E B R  
→ O V E B R  
→ O V E A R  
→ O V E B R  
→ O V E B R  
....

Place-holder amino acid: Valine

Secondary structure:

E = Beta-strand

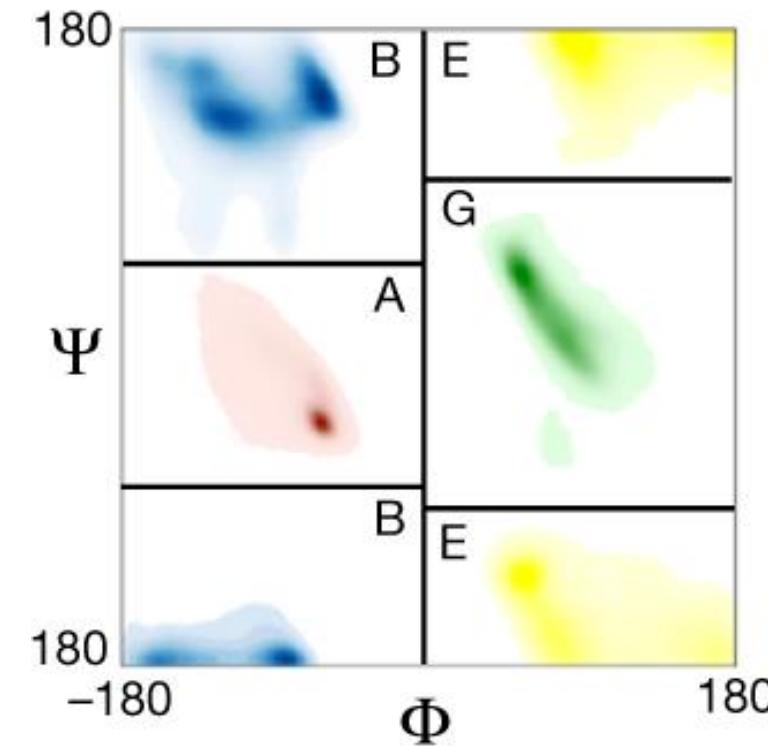
H = Alpha-helix

L = Loop

ABEGO type

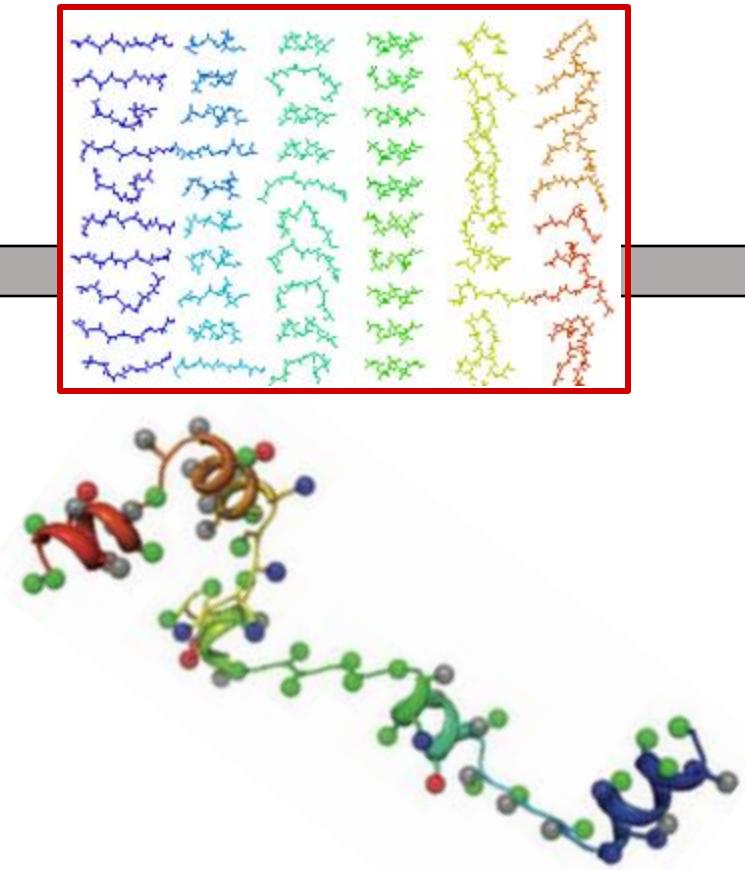
“O” = cis-peptide bond

Blueprint + constraints



# Picking fragments based on local structure properties == Blueprint

O V L R  
O V L R  
O V E B R  
O V E B R  
O V E B R  
O V E B R  
O V E B R  
O V E B R  
O V E B R  
O V E B R  
O V L A R  
O V L A R  
O G L G R  
O V E B R  
O V E B R  
O V E A R  
O V E B R  
O V E B R

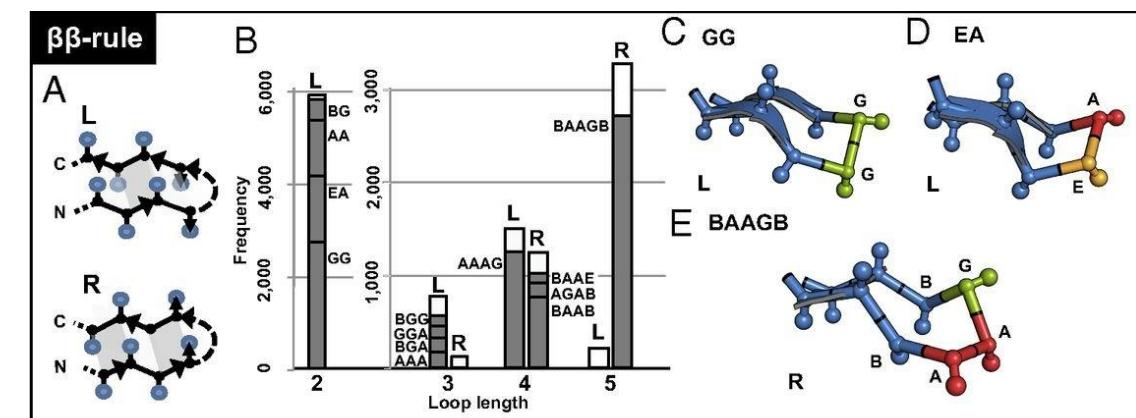
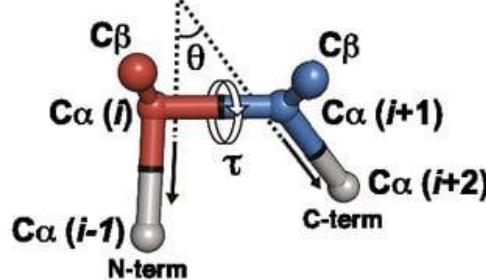
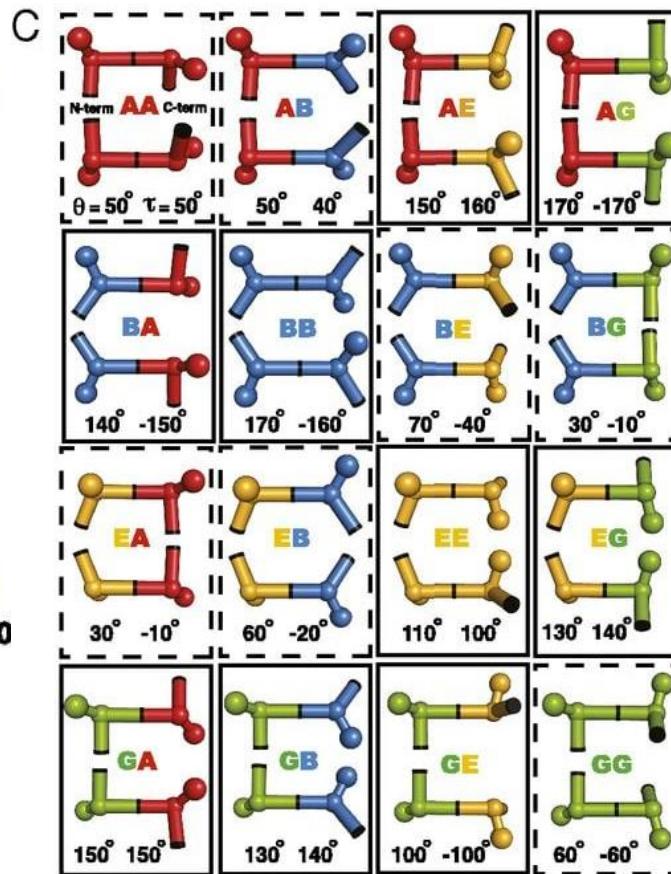
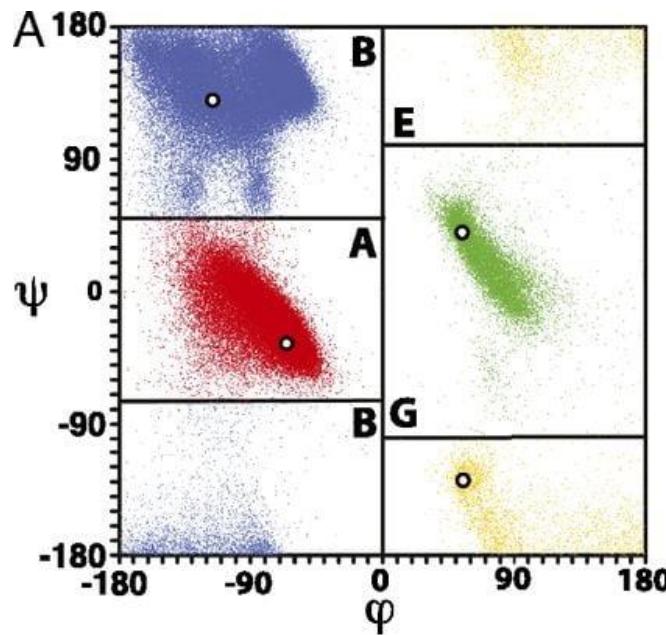


# Blueprint + constraints

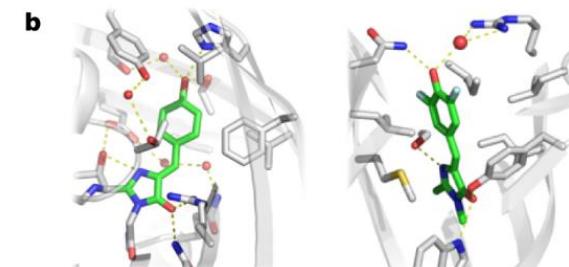
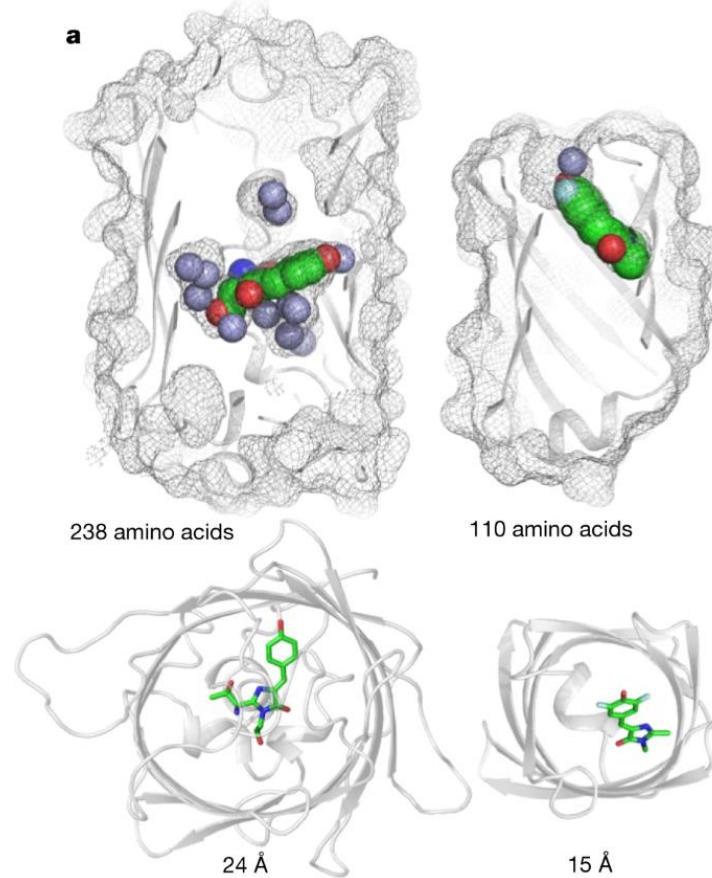
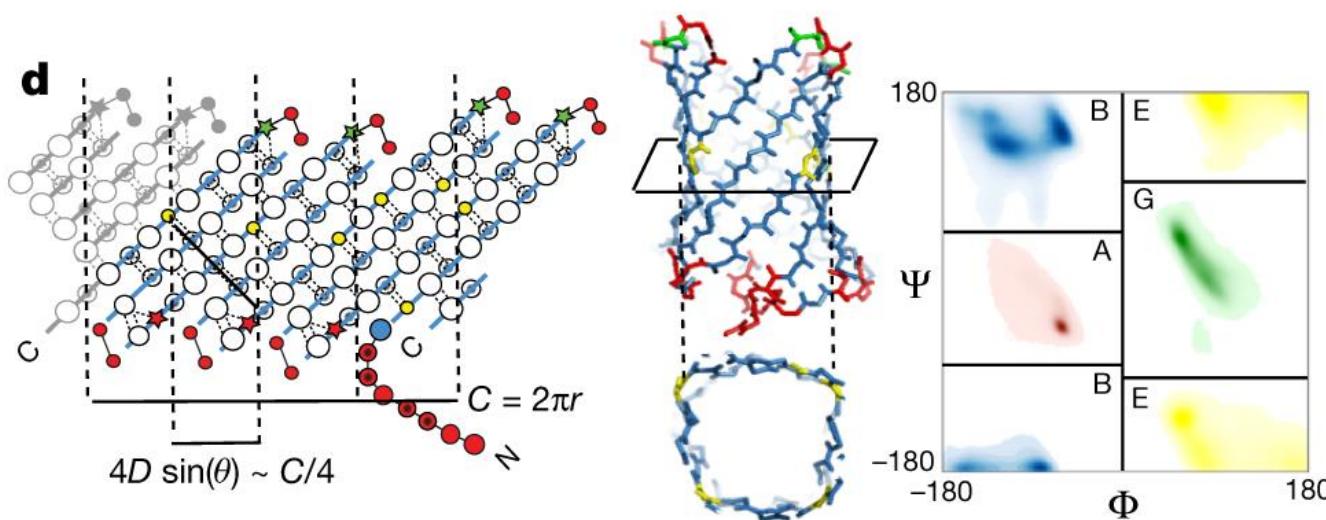


# Sequence design

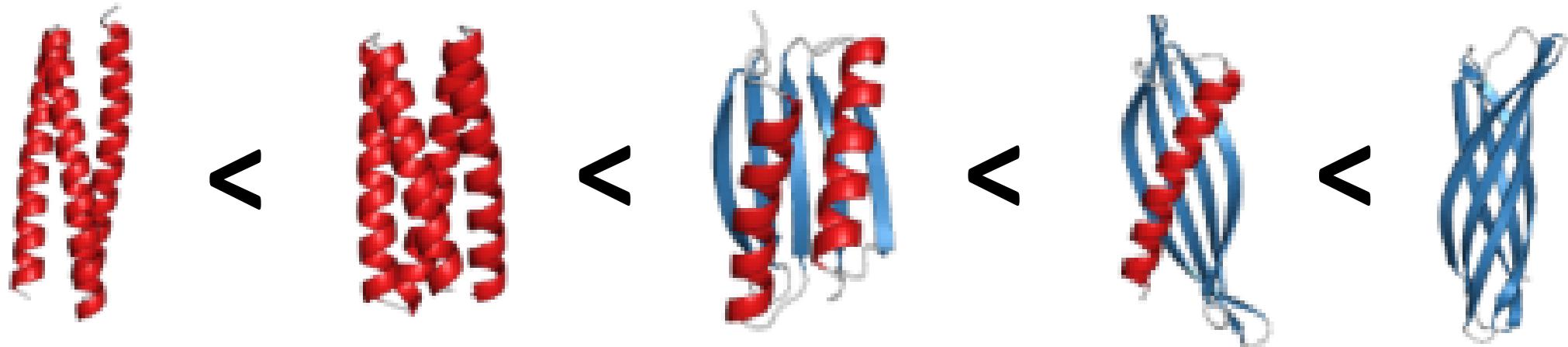
# Generating designable backbones



# Generating complex protein folds



# *De novo* design: remaining challenges

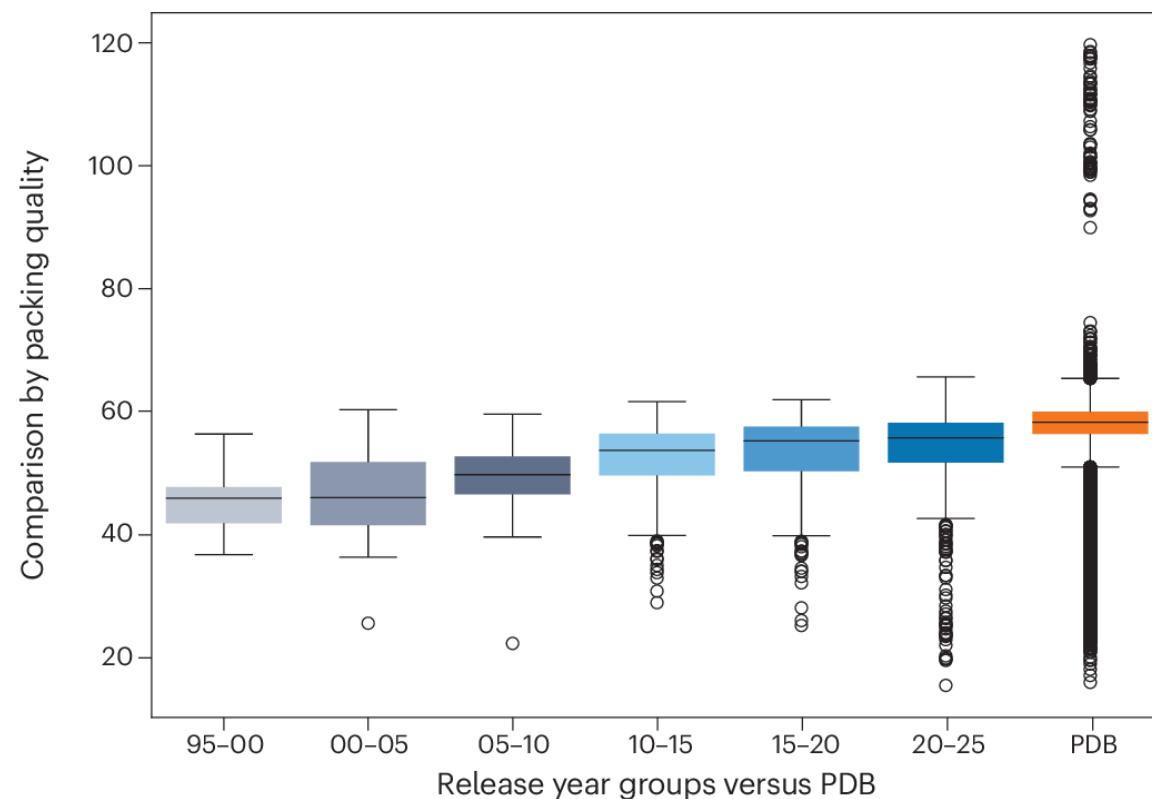


Increases with:

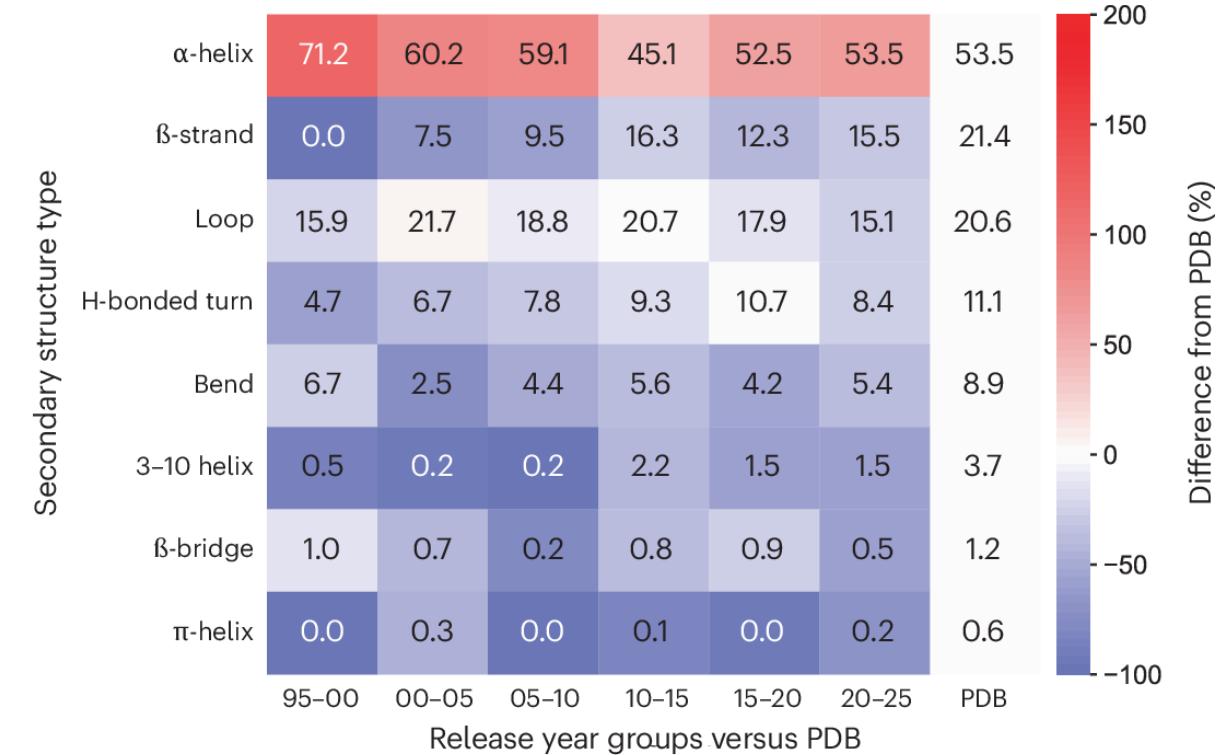
- Size
- Long-range interactions
- Beta-sheet content
- Membrane proteins require specific energy functions

... but remain pre-dominantly alpha-helical

**d Comparison by packing quality**



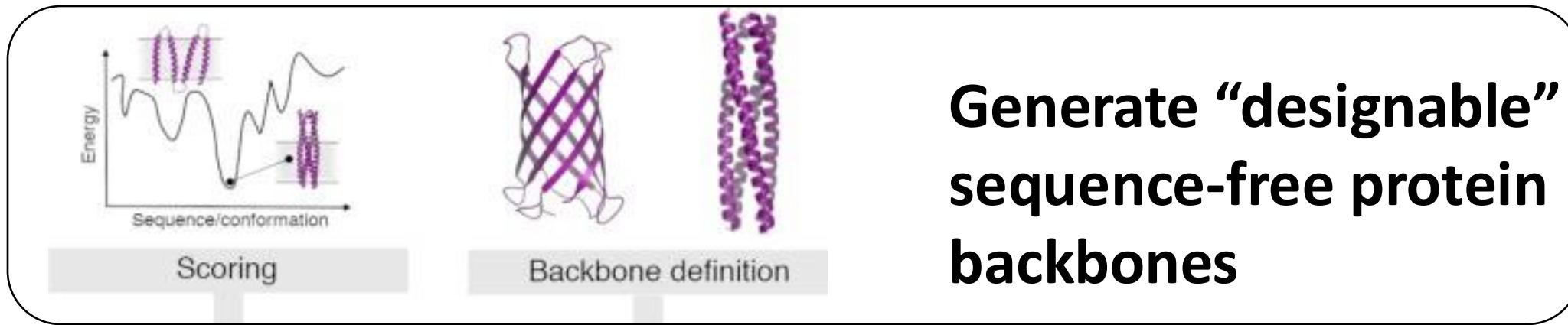
**b Design secondary structure percentages by year group versus PDB**



# AI-assisted design

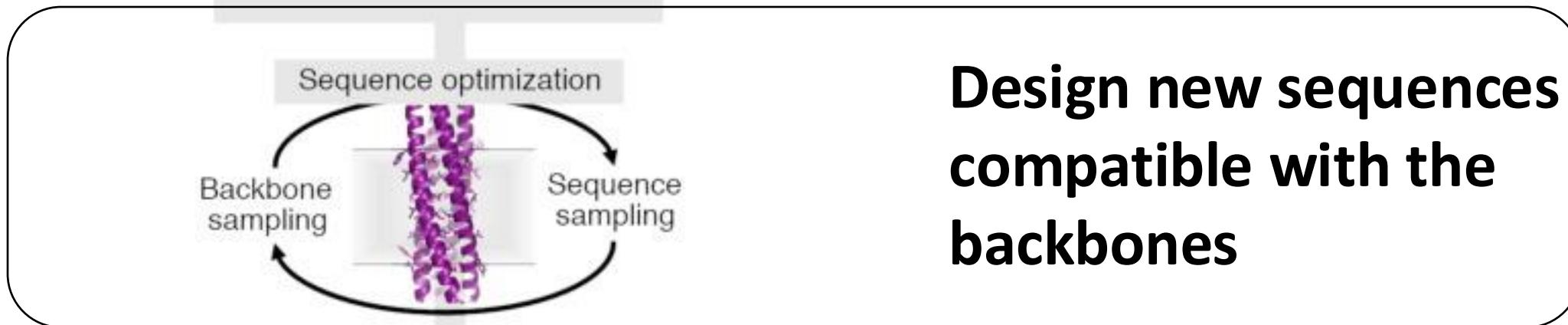
# *De novo* design pipeline

Backbone generation



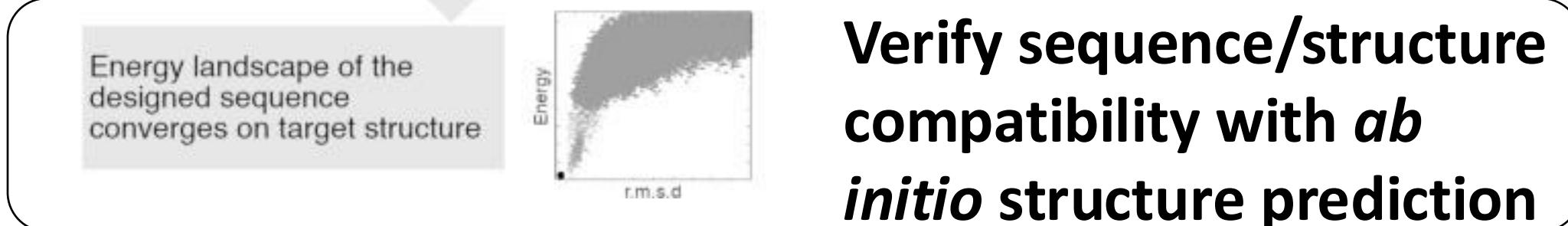
**Generate “designable” sequence-free protein backbones**

Sequence design



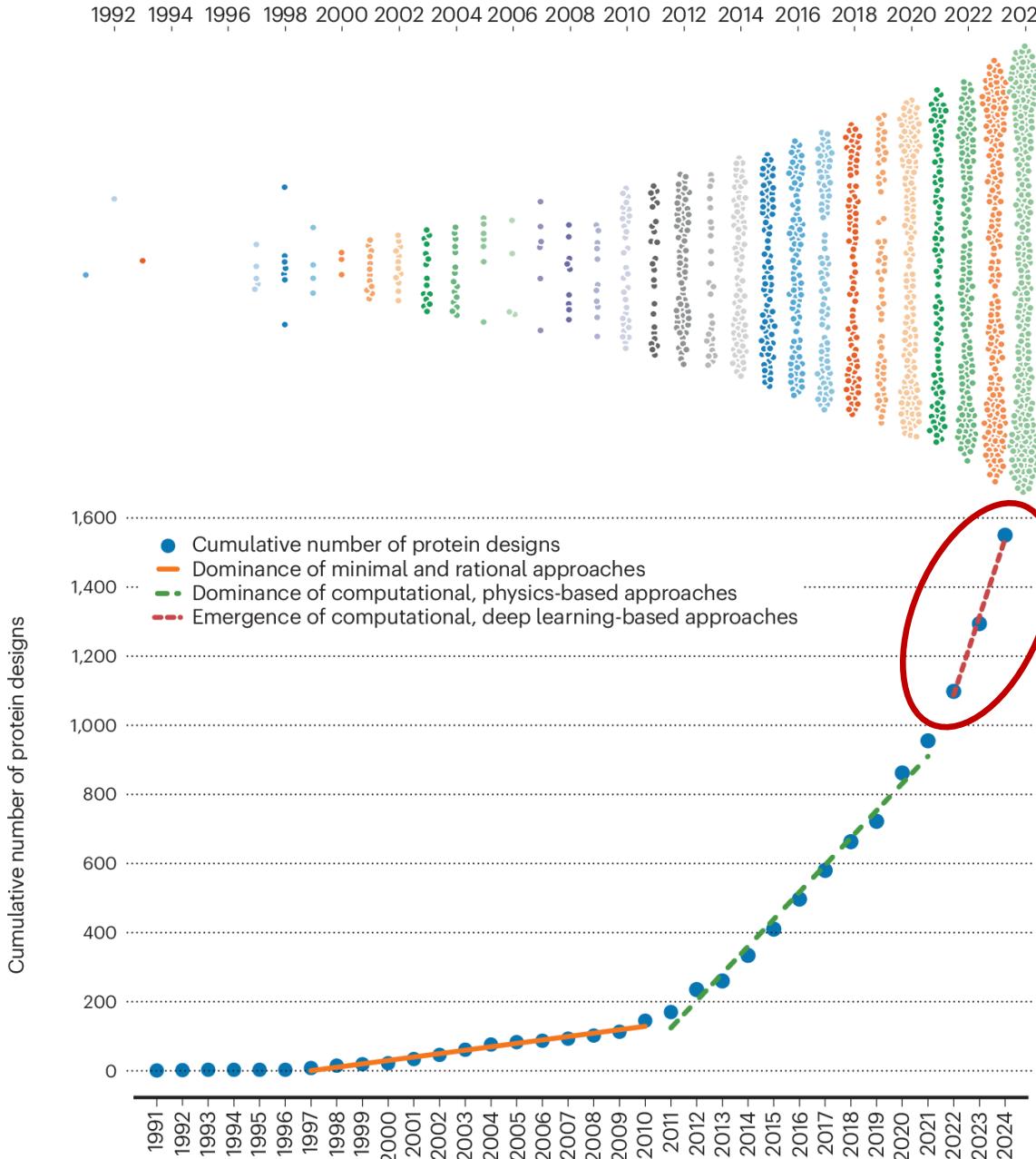
**Design new sequences compatible with the backbones**

Validation



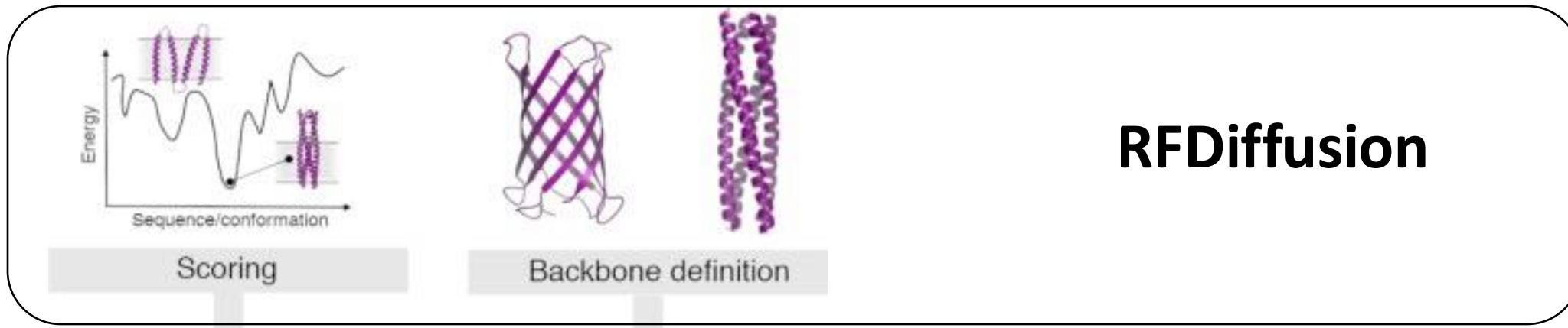
**Verify sequence/structure compatibility with *ab initio* structure prediction**

# Evolution of *de novo* protein design



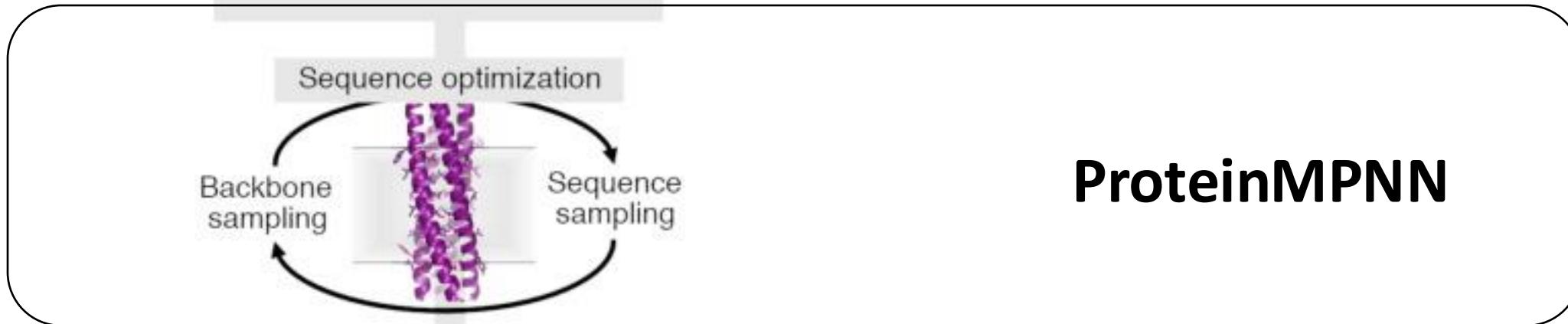
# AI-era *de novo* design pipeline

Backbone generation



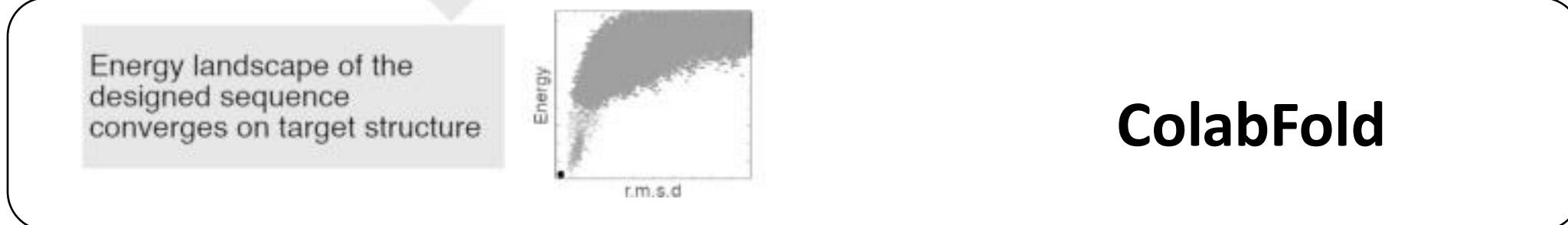
**RFDiffusion**

Sequence design



**ProteinMPNN**

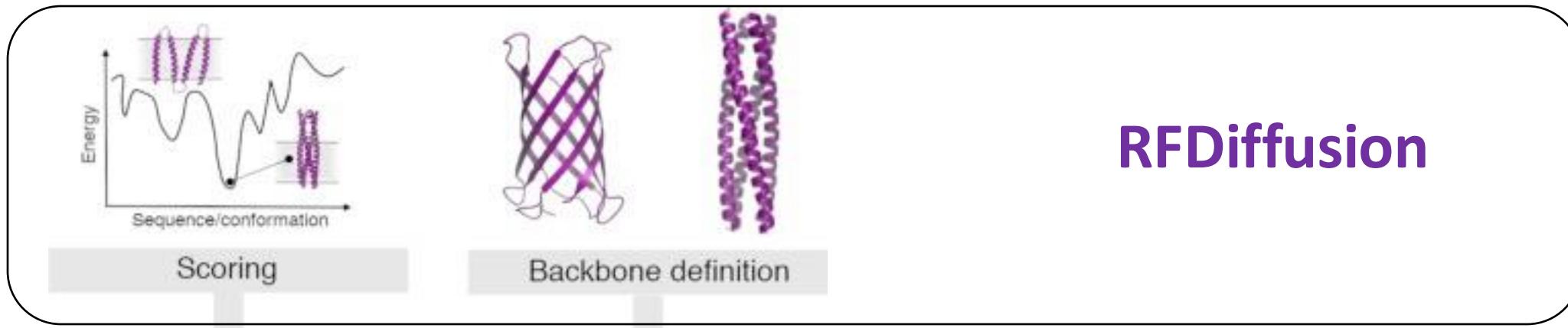
Validation



**ColabFold**

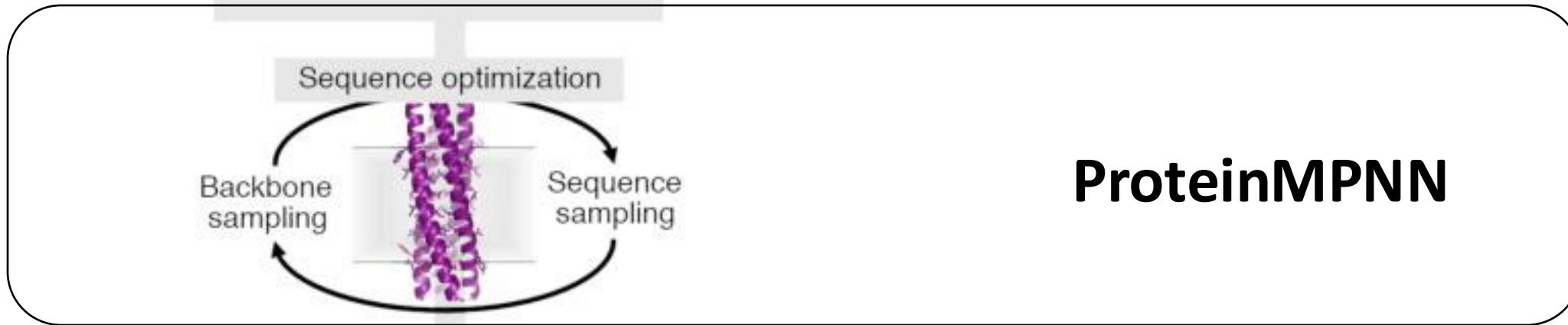
# AI-era *de novo* design pipeline

Backbone generation

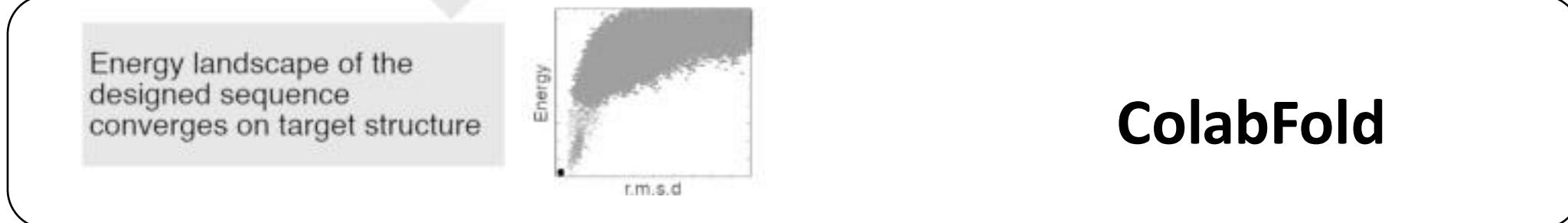


**RFDiffusion**

Sequence design

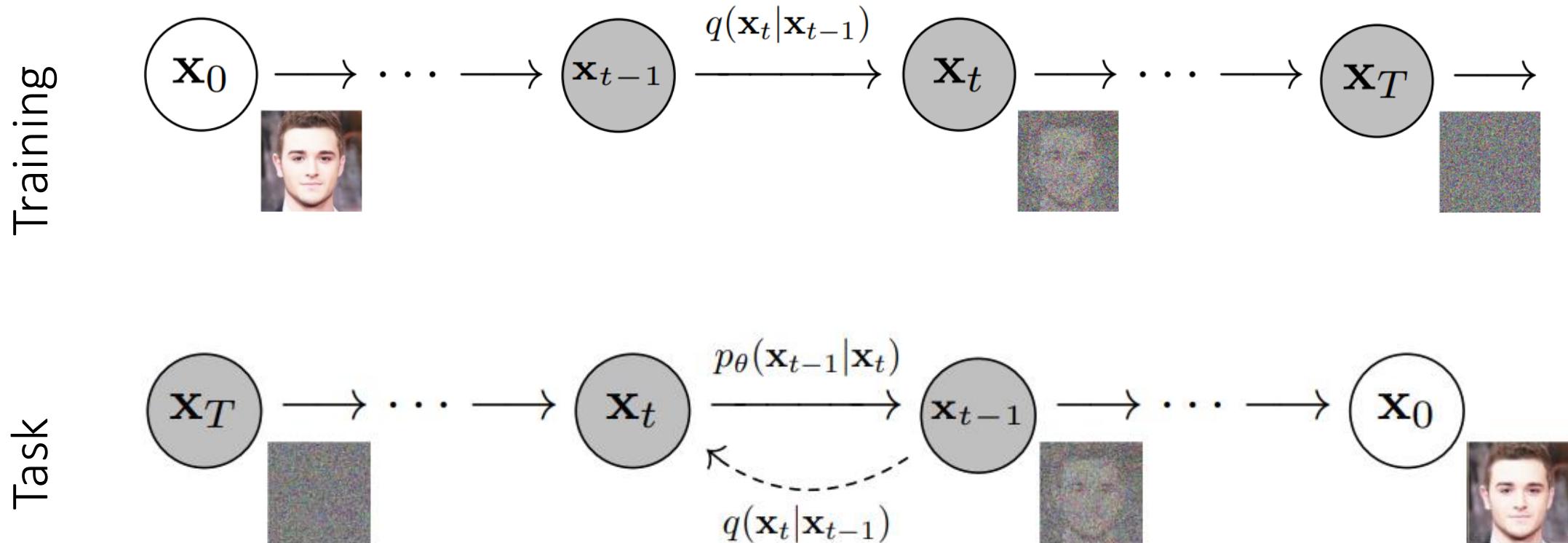


Validation



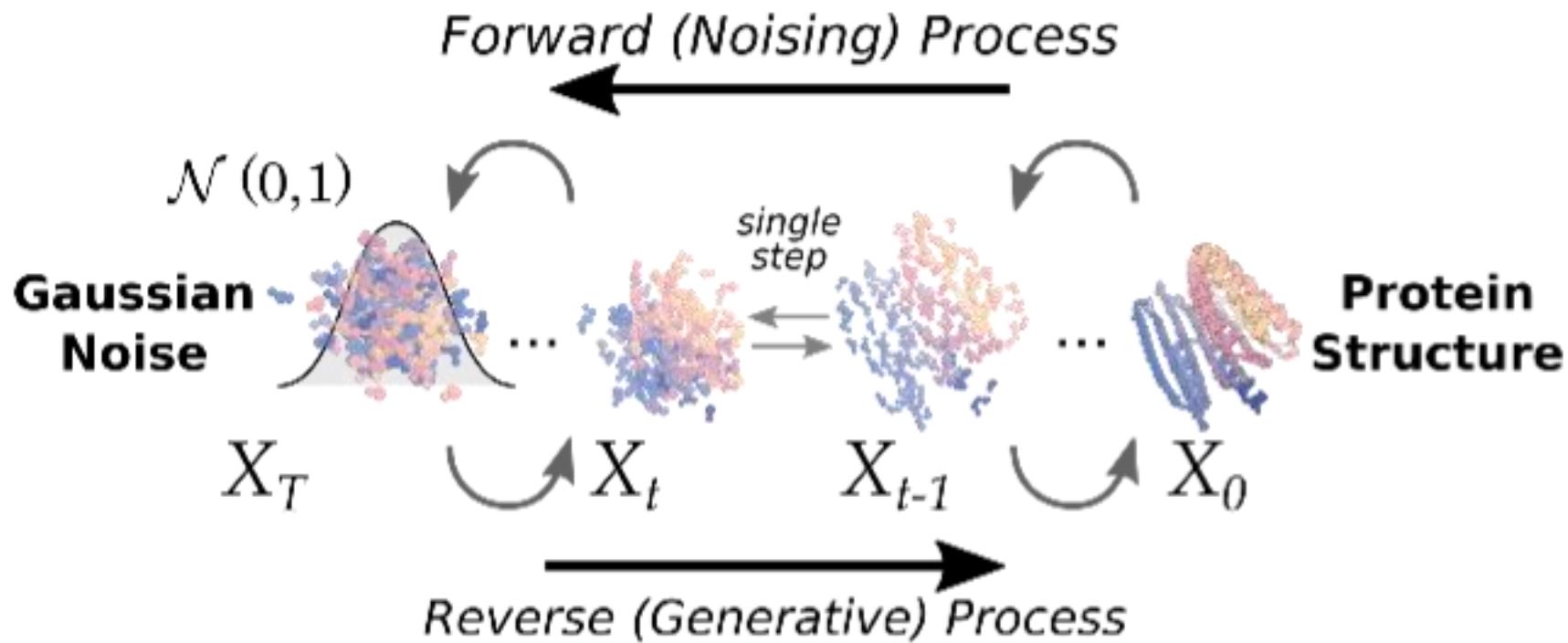
# RFDiffusion: protein structure generator inspired by computer vision

Learning to generate images from noise → Generative AI model??



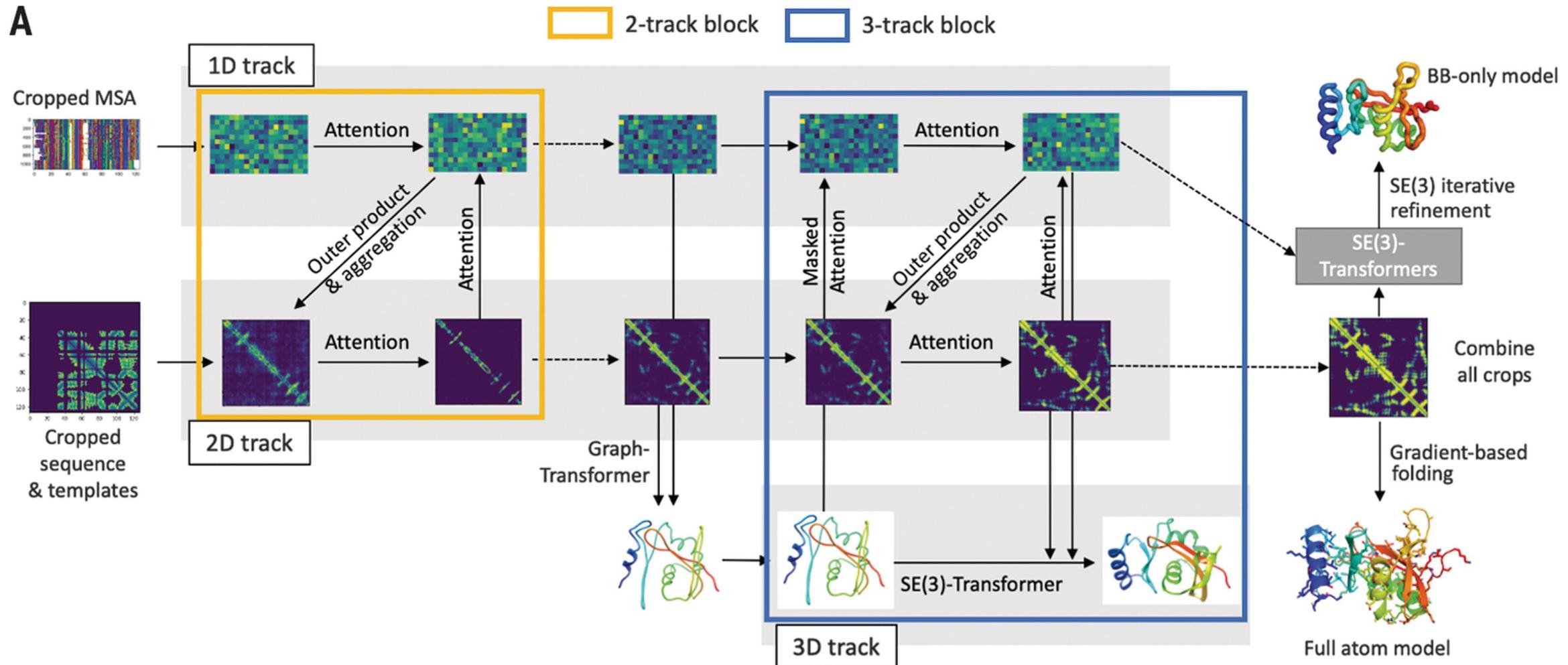
$RF_{diffusion}$ : Noising/denoising PDB structures

## Diffusion Model

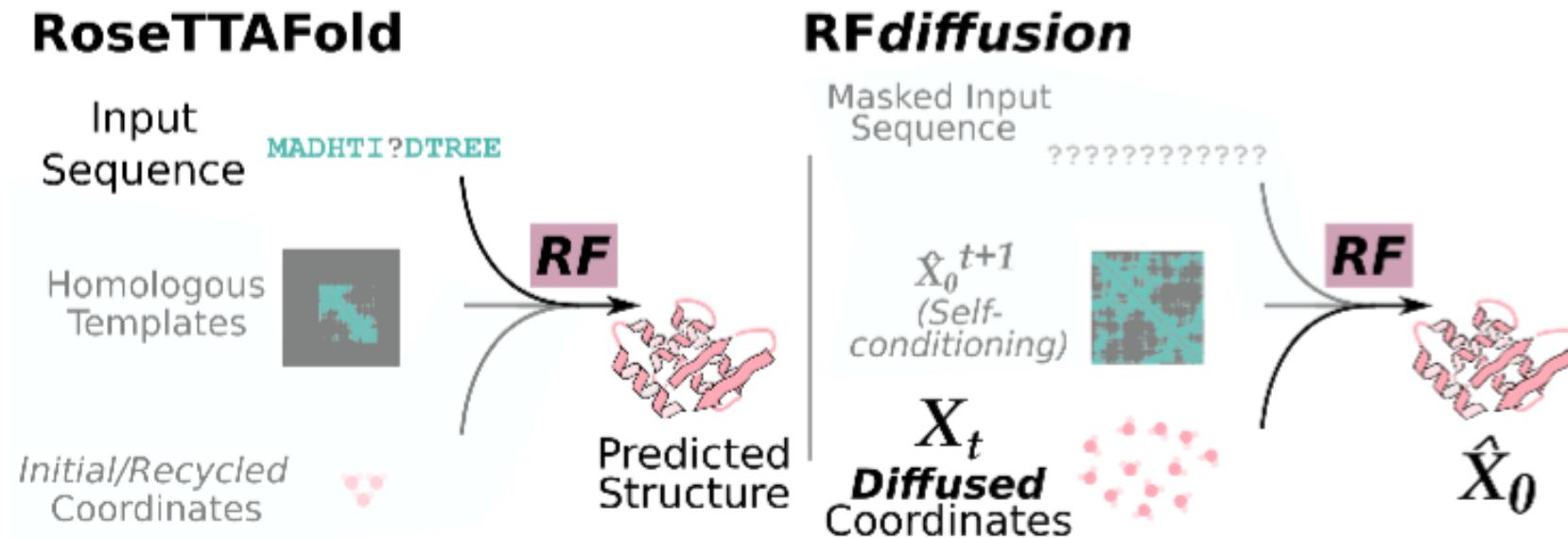


# RoseTTaFold: an open-source version of AlphaFold2

A

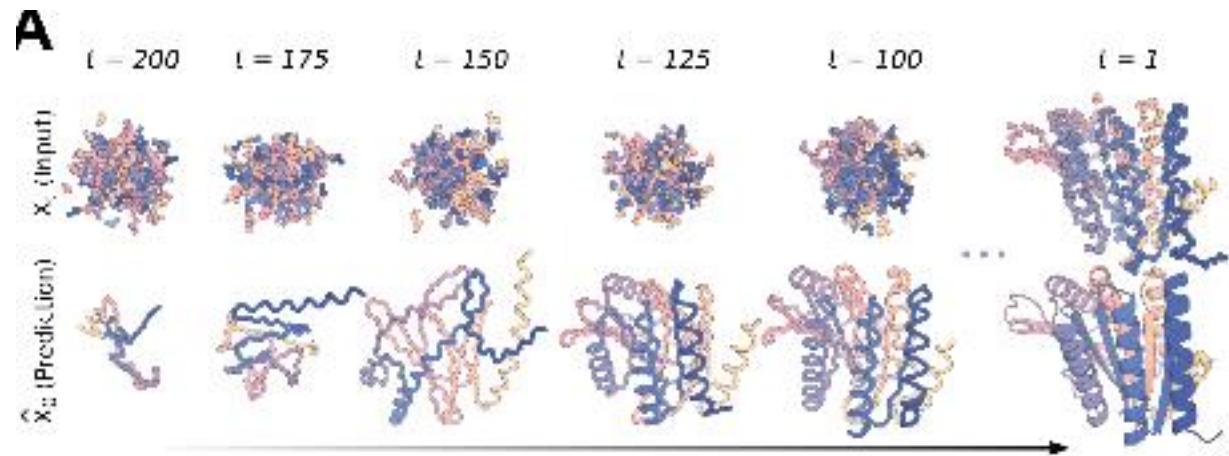
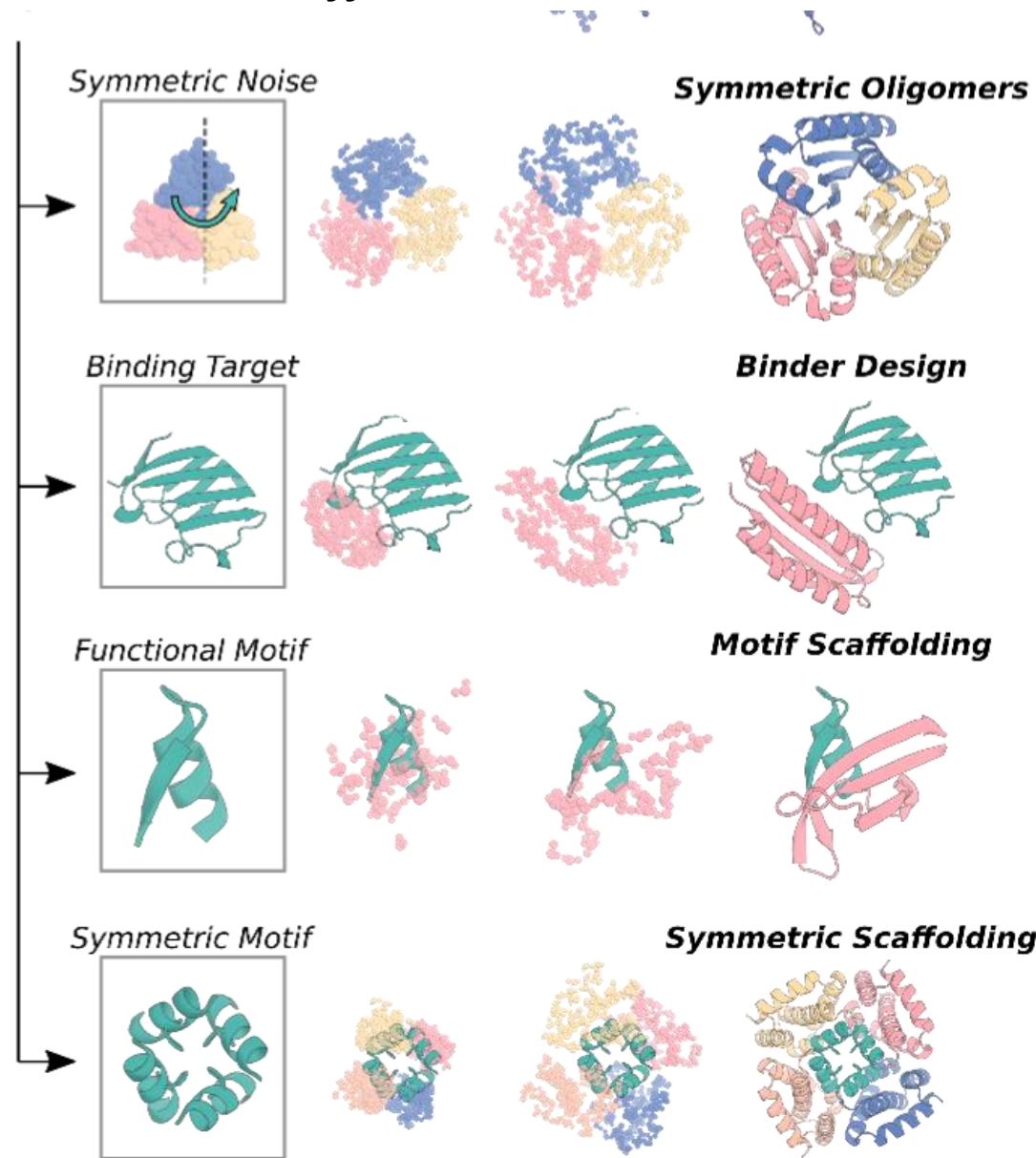


# $RF_{diffusion}$ : From structure prediction to structure generation

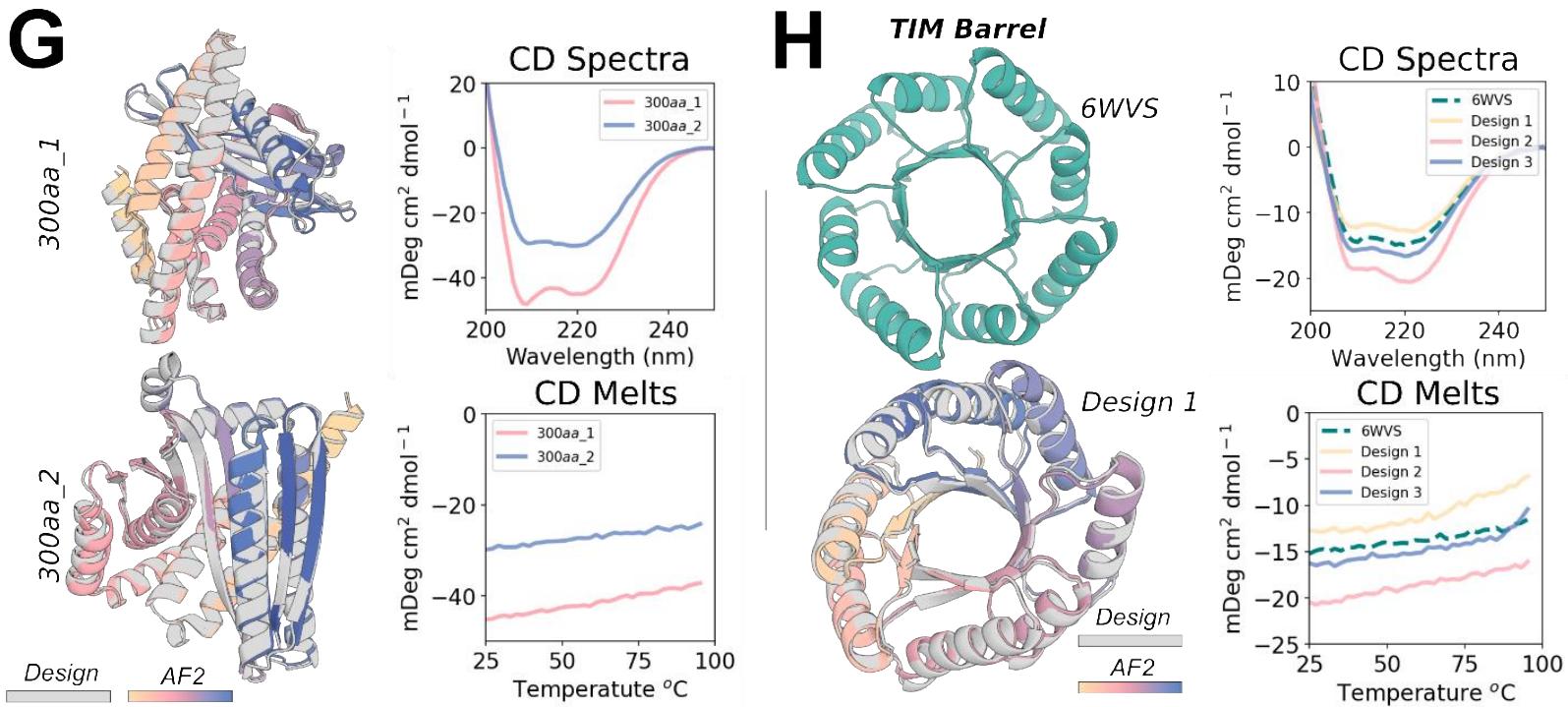


- Building on the pre-trained RoseTTAFold model (knows a lot about protein structure)
- Generates designable backbones

# $RF_{diffusion}$ : solving different protein design problems

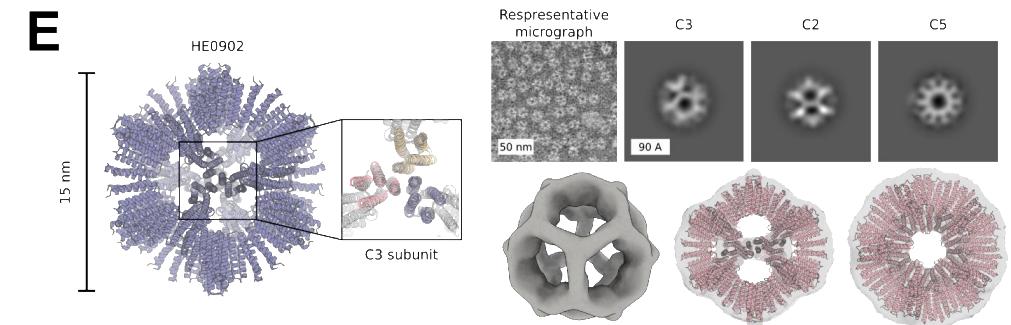
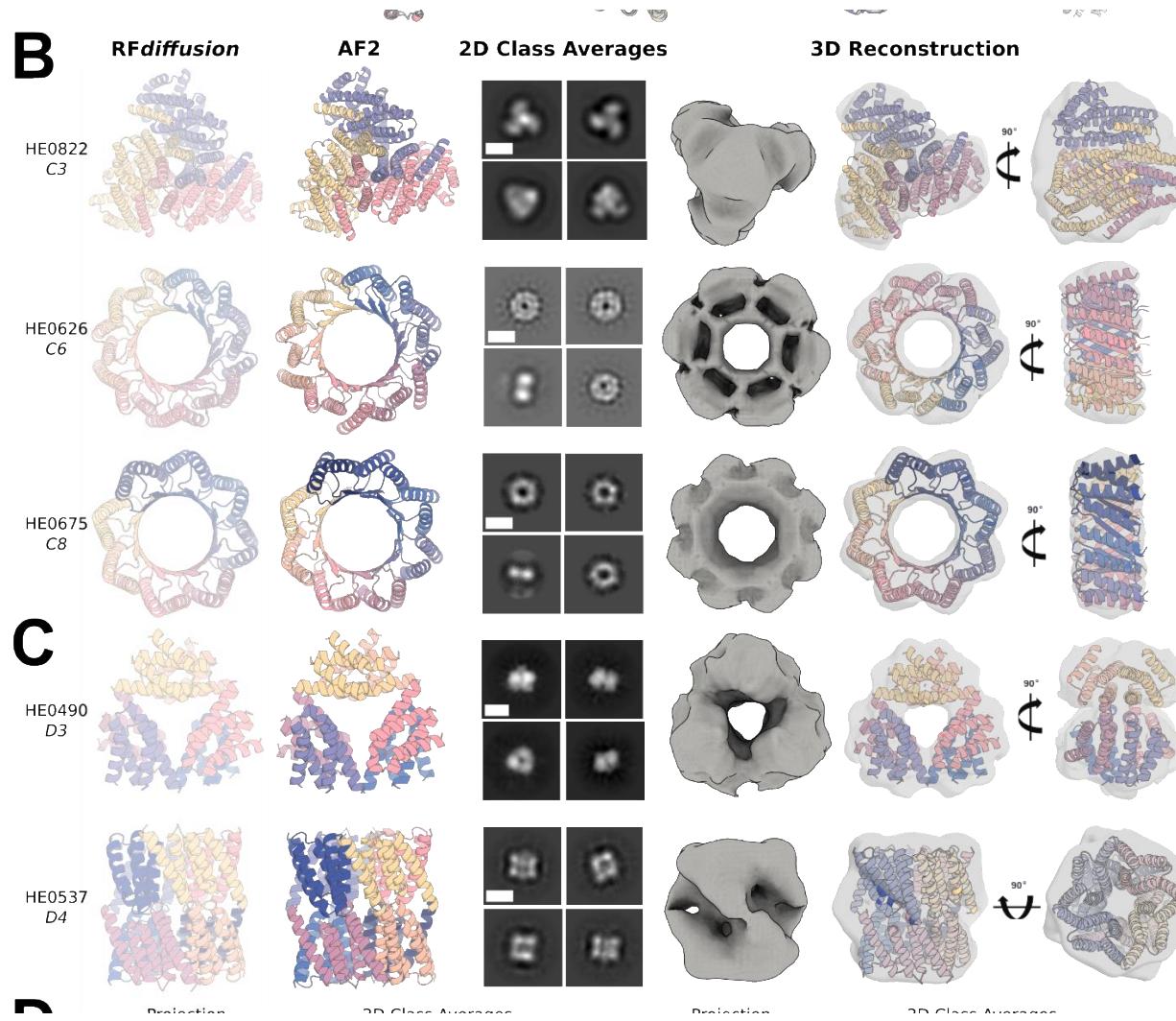


# Monomoeric structures



9/11 monomeric and folded!

# Symmetric self-assembles and nanocages

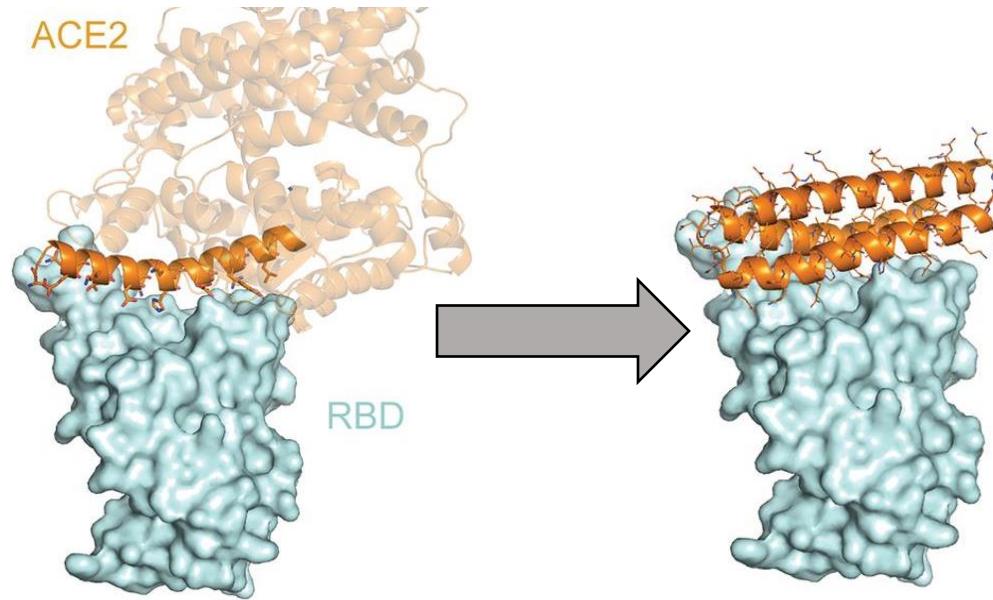


70/608 designs folded

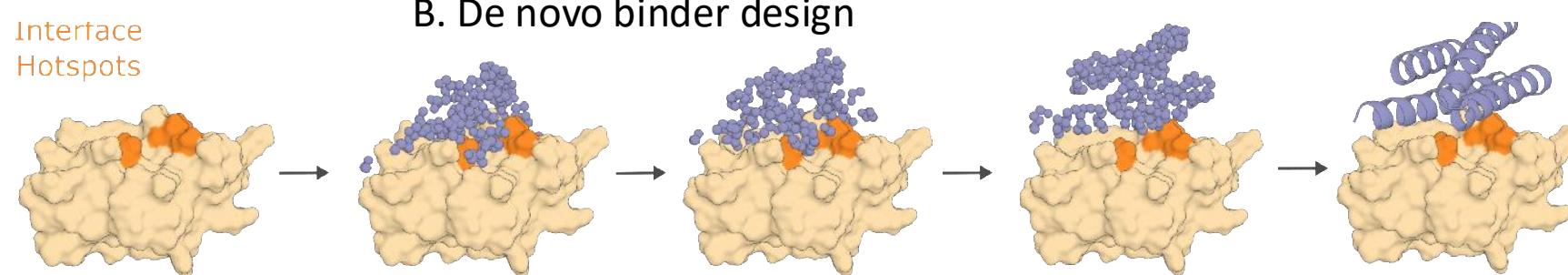
Dihedral, tetrahedral, ... symmetries

# De novo design of protein binders

A. Motif scaffolding



B. De novo binder design

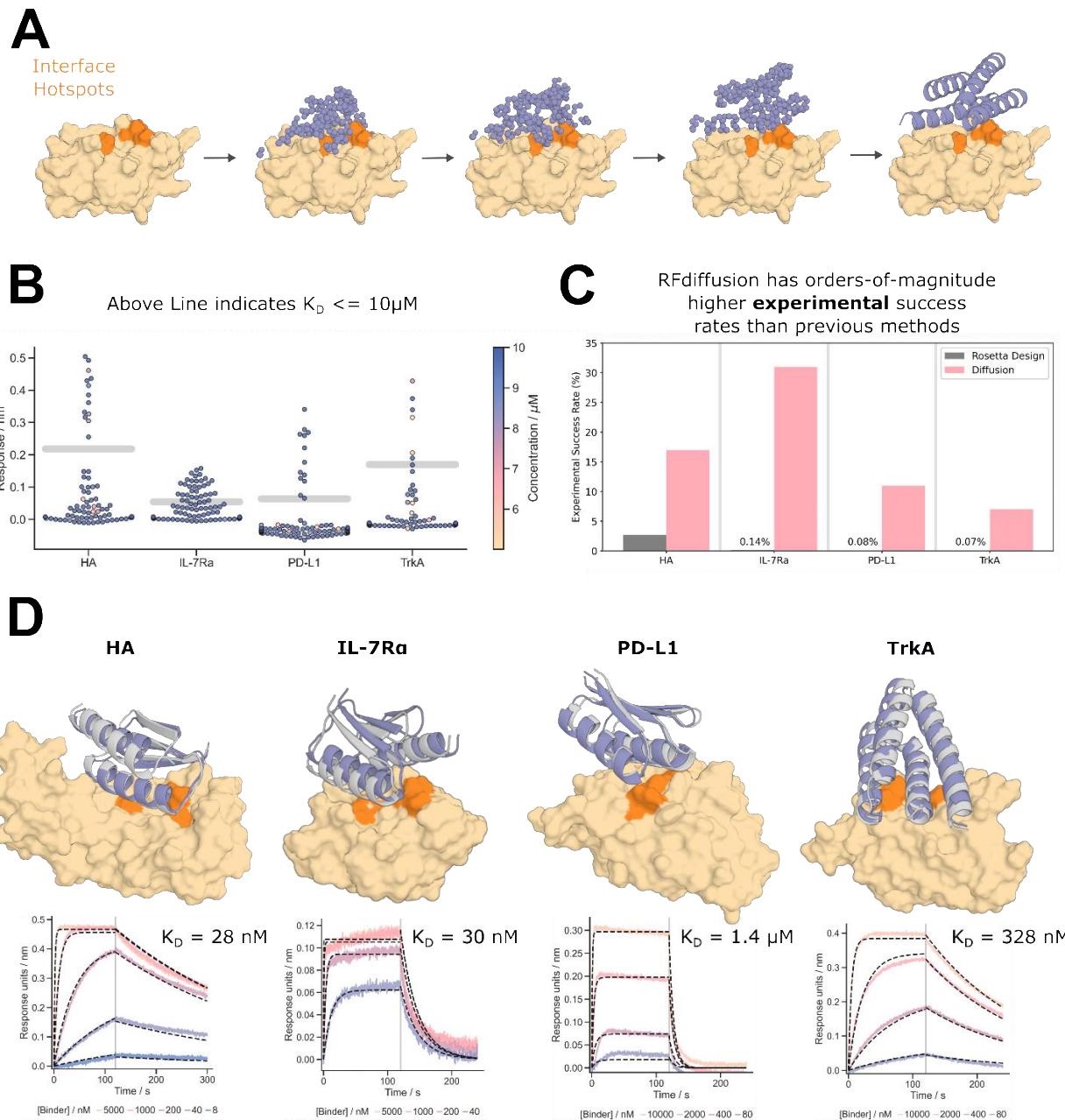


# *De novo* designed protein-protein interfaces

4 targets

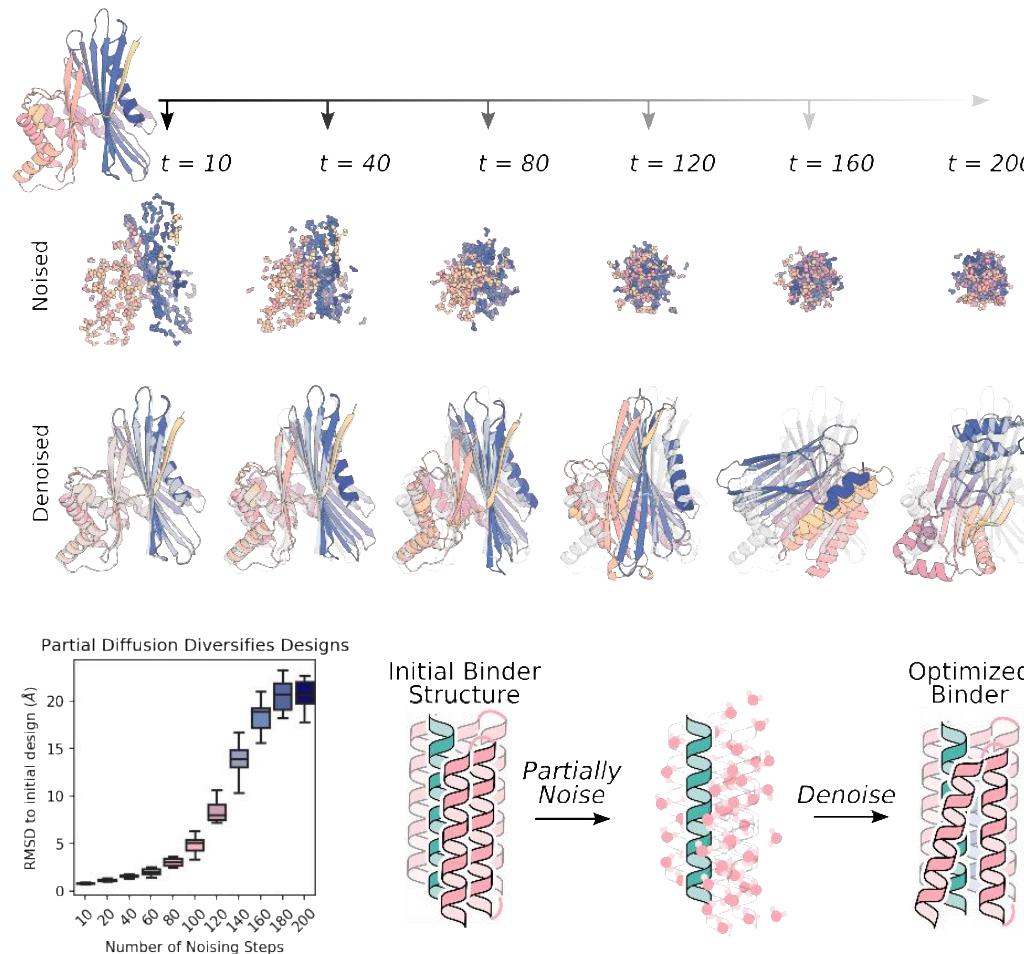
Binders obtained for all targets (95 designs tested per target)

~18% success rate (< 1% success rate before, required yeast display libraries).



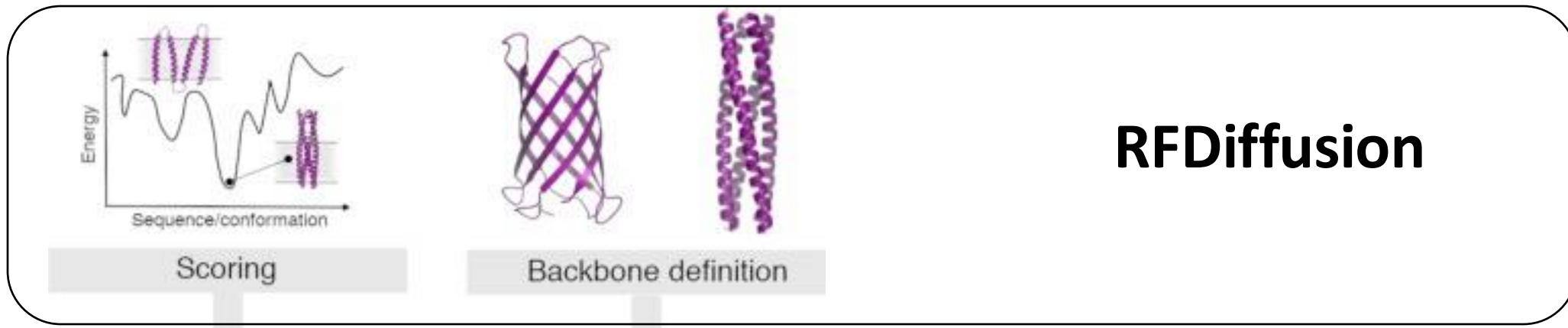
# Partial diffusion (refinement)

## A Partial noising and denoising using RFdiffusion

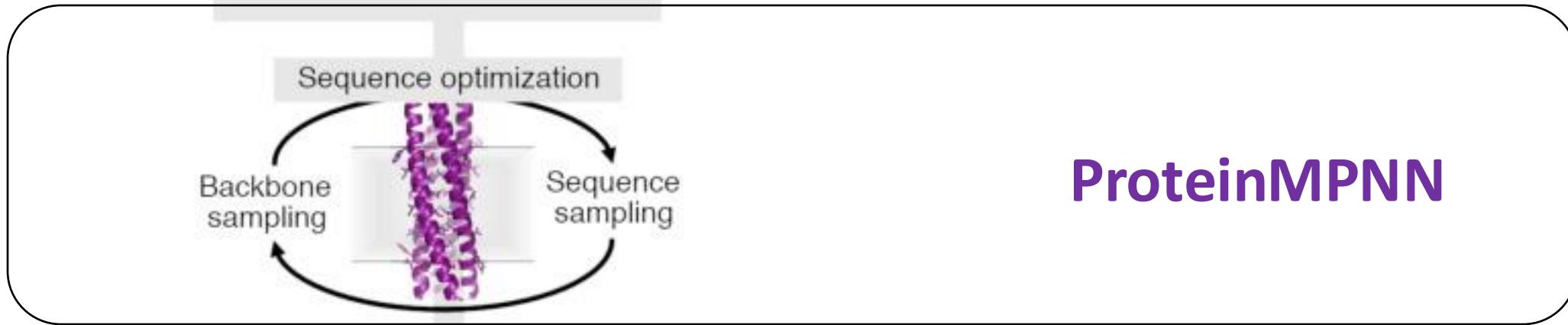


## 2. Sequence design with ProteinMPNN

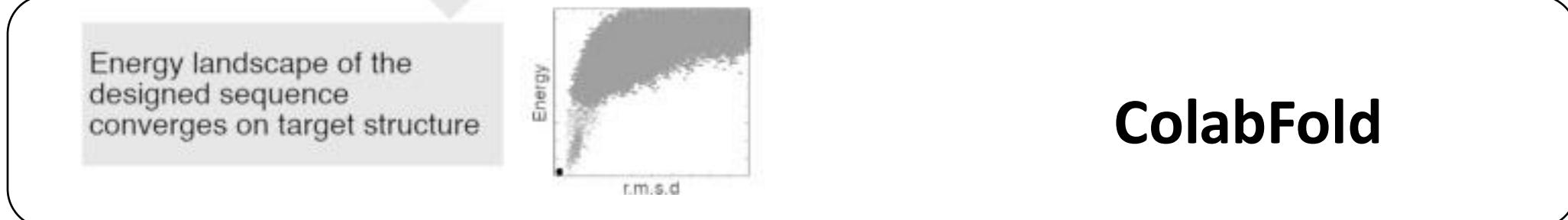
Backbone generation



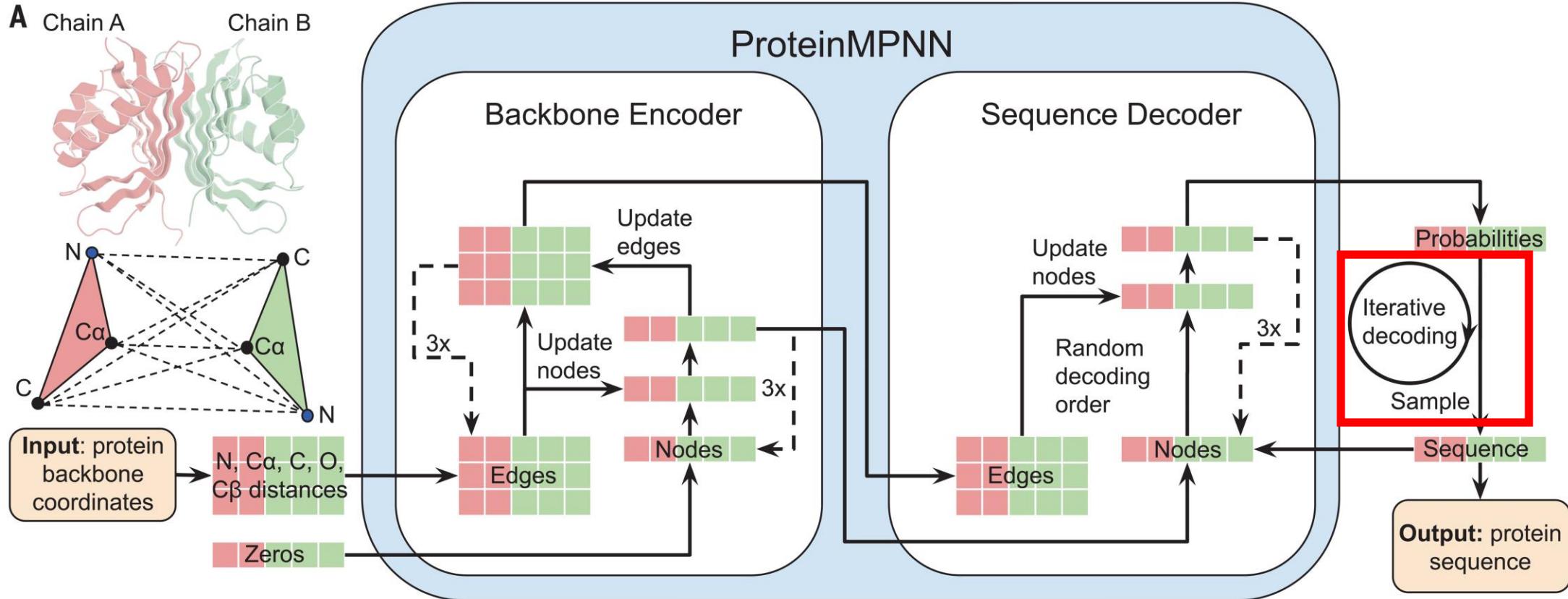
Sequence design



Validation



# ProteinMPNN: Graph neural network (GNN)



# ProteinMPNN: Graph neural network (GNN)

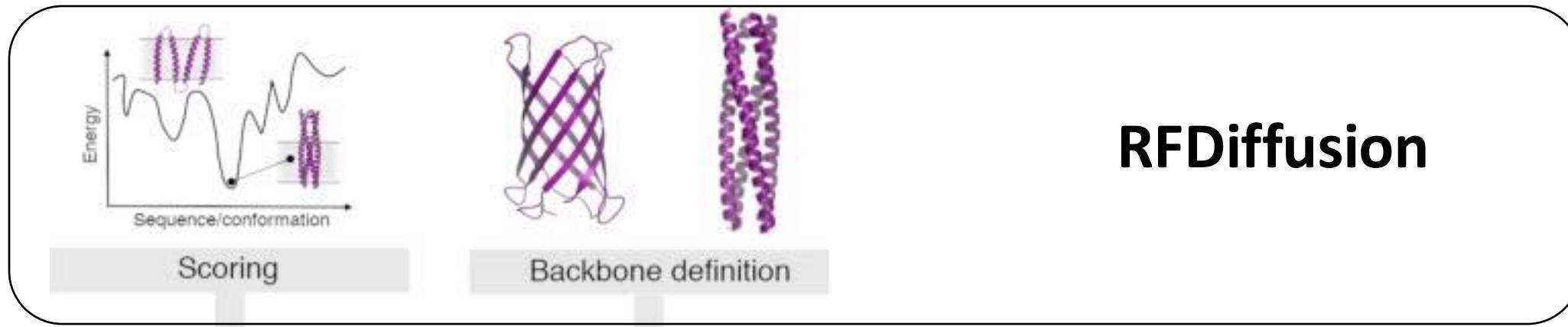
## **Improving protein expression, stability, and function with ProteinMPNN**

Kiera H. Sumida<sup>1,2</sup>, Reyes Núñez-Franco<sup>3</sup>, Indrek Kalvet<sup>2,4,5</sup>, Samuel J. Pellock<sup>2,4</sup>,  
Basile I. M. Wicky<sup>2,4</sup>, Lukas F. Milles<sup>2,4</sup>, Justas Dauparas<sup>2,4</sup>, Jue Wang<sup>2,4</sup>, Yakov Kipnis<sup>2,4,5</sup>,  
Noel Jameson<sup>1</sup>, Alex Kang<sup>2</sup>, Joshmyn De La Cruz<sup>2</sup>, Banumathi Sankaran<sup>6</sup>, Asim K. Bera<sup>2,4</sup>,  
Gonzalo Jiménez-Osés<sup>3,7</sup>, David Baker<sup>2,4,5\*</sup>

Improved myoglobin and TEV protease

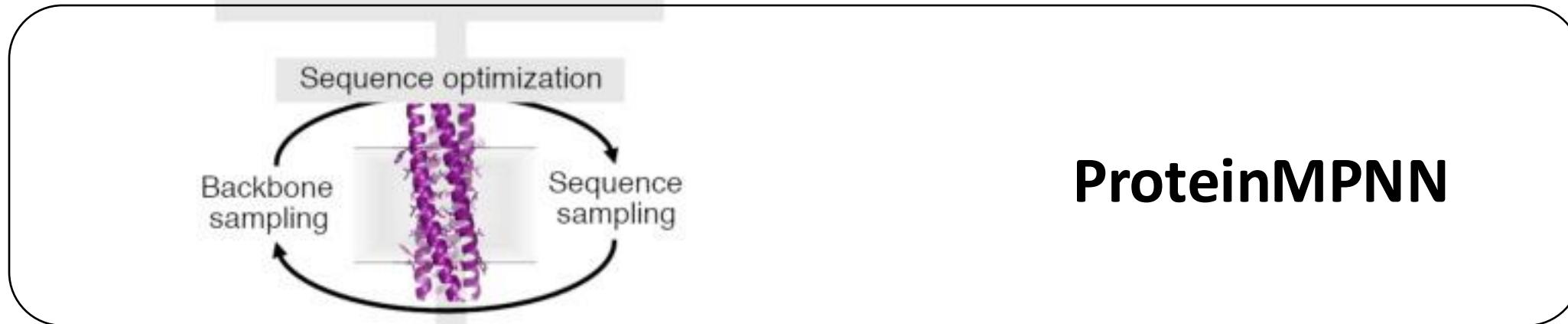
### 3. Prioritizing designs in silico with Colabfold

Backbone generation



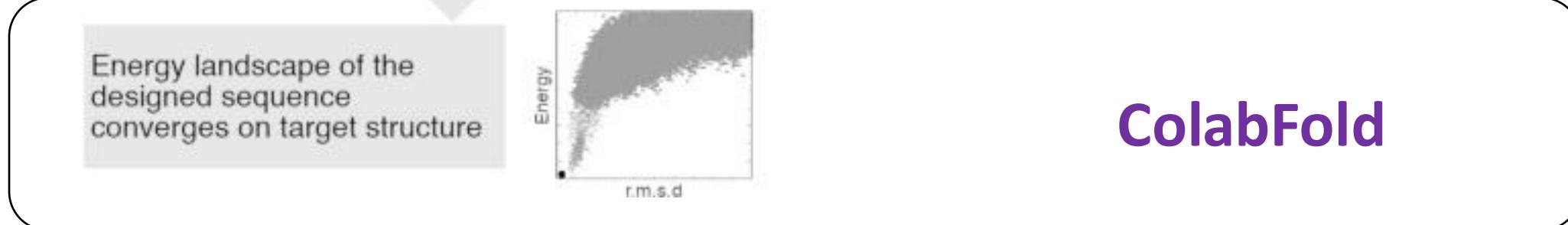
**RFDiffusion**

Sequence design



**ProteinMPNN**

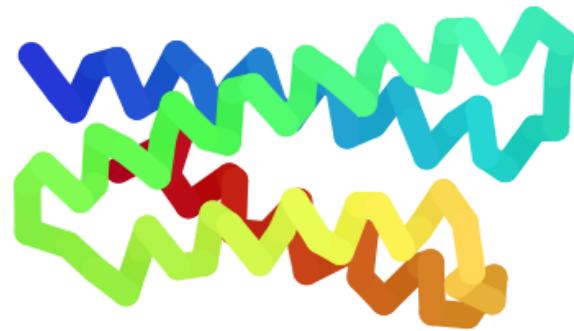
Validation



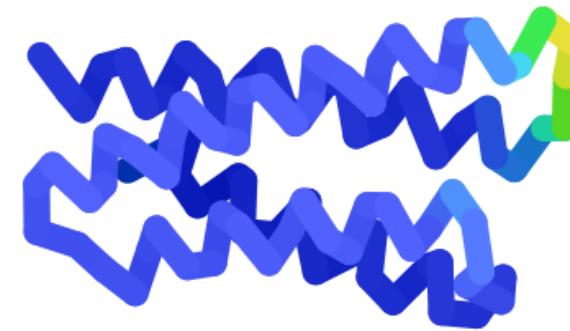
**ColabFold**

# Example from day1 – using AlphaFold in Colab

colored by N→C



colored by pLDDT



**Reminder:** Neoleukin-4 from Day 1

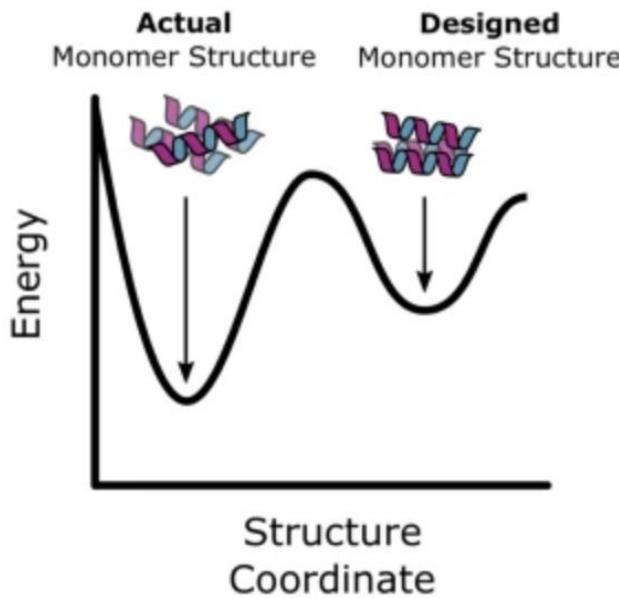
Msa\_mode is single\_sequence

# BUT binders can have several failure modes

Interface OK, but protein misfolds/aggregates

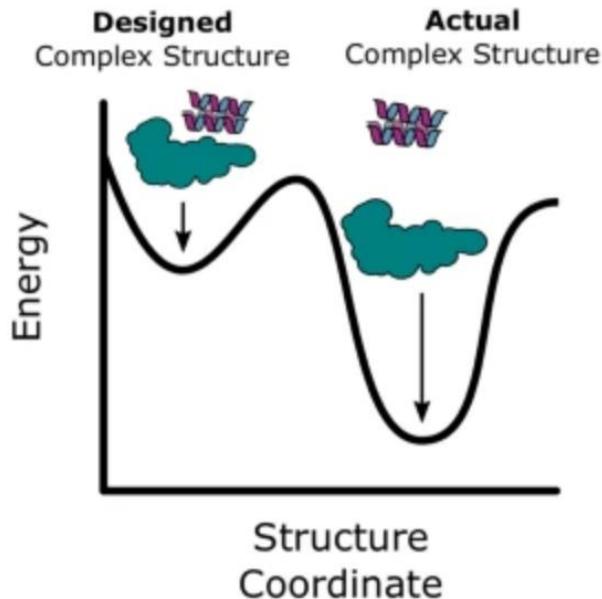
Protein folds but does not bind

## Type 1 Failure



AF2 prediction from  
single sequence

## Type 2 Failure

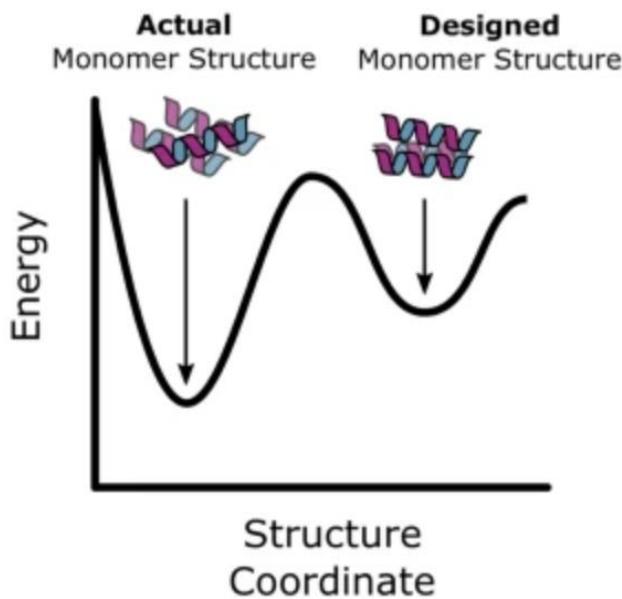


# BUT binders can have several failure modes

Interface OK, but protein misfolds/aggregates

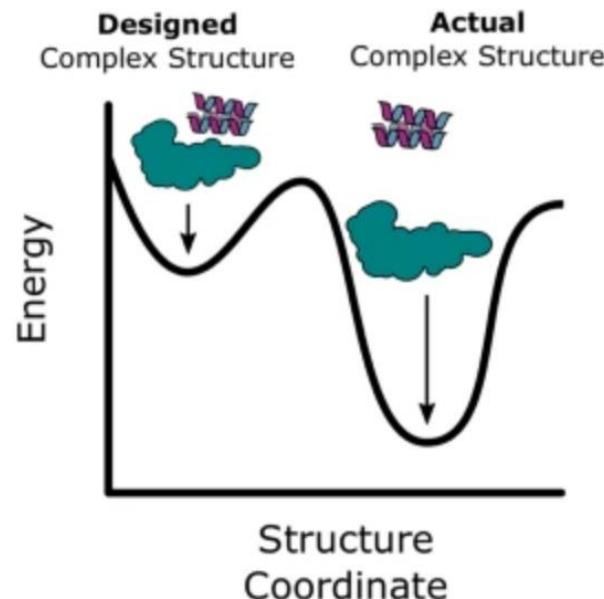
Protein folds but does not bind

## Type 1 Failure



AF2 prediction from single sequence

## Type 2 Failure



AF2 prediction from single sequence with initial guess

# AF2 with initial guess

Designed sequence

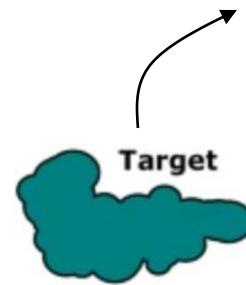
MGTF...TKLE

+

Target sequence

MKPRQ...SRHV

AF structure prediction  
Single sequence!



Target structure as  
template



Questions?

Thank you!

## Workshop Schedule

TIME	WEDNESDAY 28 MAY	THURSDAY 29 MAY	FRIDAY 20 MAY
<b>09:00 – 10:30</b>	Protein modelling and structure prediction: Intro to key concepts, from physics-based modelling to AI	De novo protein design: introduction, minimal sequence design, structure-based design principles	De novo design with AI models: RFDiffusion, ProteinMPNN, and ColabFold
<b>10:30 – 11:00</b>	Break and questions	Break and questions	Break and questions
<b>11:00 – 12:00</b>	Introduction to protein design: predicting the effect of mutations on protein stability	Structure-based de novo design: How to generate new structures? The chicken-and-egg problem	Practical session: De novo design of a SARS-CoV-2 RBD binder using RFDiffusion and ProteinMPNN
<b>12:00 – 13:30</b>	Lunch	Lunch	Lunch
<b>13:30 – 15:00</b>	Practical session: AlphaFold hands-on	Practical session: Parametric design of alpha-helical bundles	Practical session: Data analysis and group slide preparation
<b>15:15 – 17:45</b>	Practical session: In silico mutational scanning and $\Delta\Delta G$ calculations	Practical session: Sequence design for parametric bundles with PyRosetta	Practical session: Group presentations and results discussion

□

# Learning how to design alpha-helical bundles

