When AI meets life sciences
当人工智能遇上生命科学

2025 清华大学 AI创新实践课程

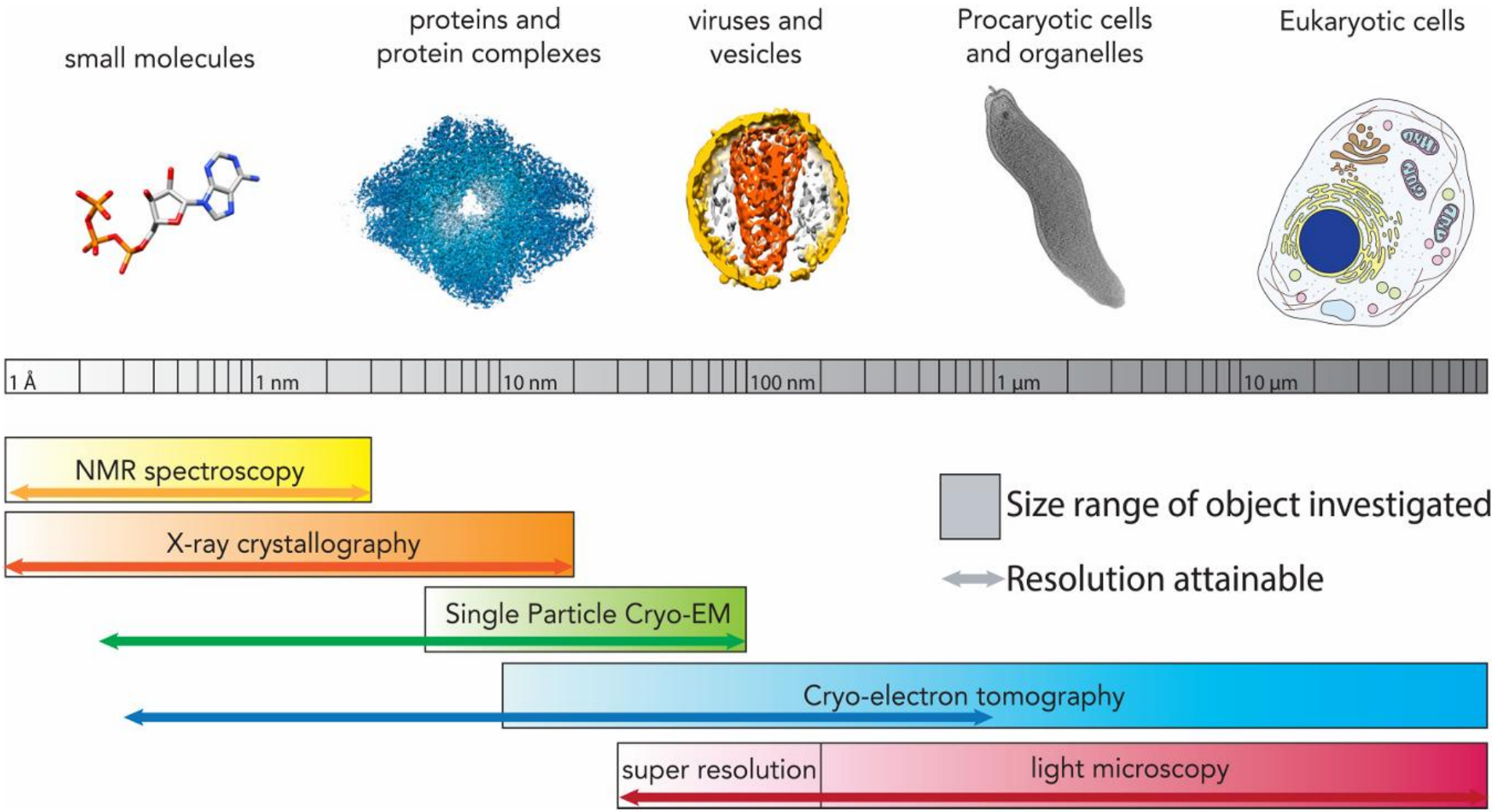
张强锋，清华大学

Dec 1, 2025

Outline: AI-based protein structure prediction—AF2 and others

- The grand challenge
- Protein structure prediction in the past fifty years
- The deep learning shift: pre-AF2
- AF2 deep dive
- Structure prediction landscape post-AF2
  - RosettaFold & RF2
  - AF3
  - Protein language models (pLMs)
- Protein design
- Other AI structural biology
  - Simulation
  - Docking
  - Combine with experiments
- Structure-based drug discovery

Recap: structure biology

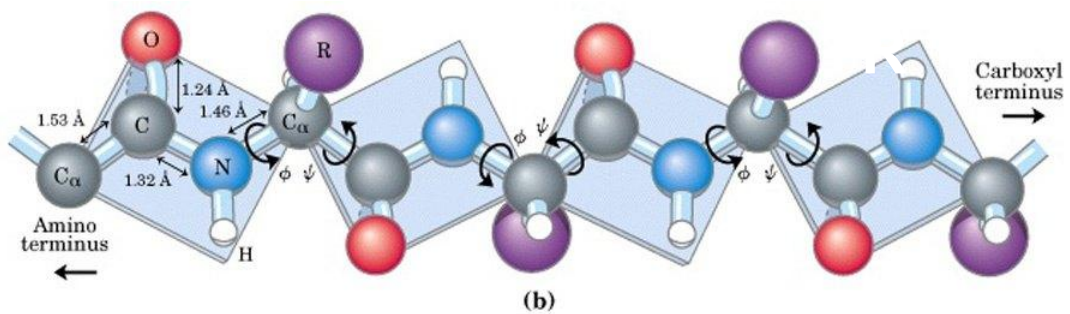


Recap: basis of protein structure

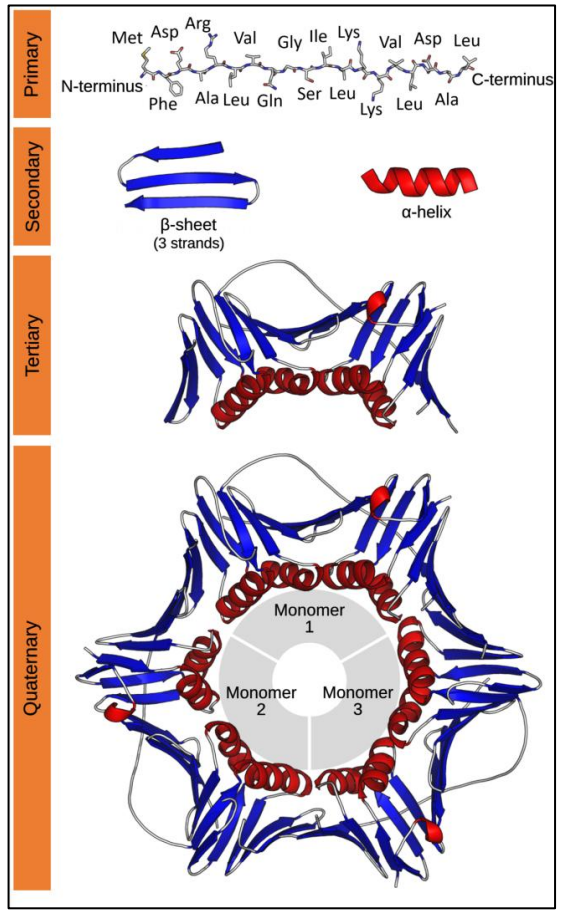
- Protein: Polymer chain formed by linking 20 different amino acids in various combinations.

. . . ASNDCEQGHILKMFPSTWYV. . .

Side chain



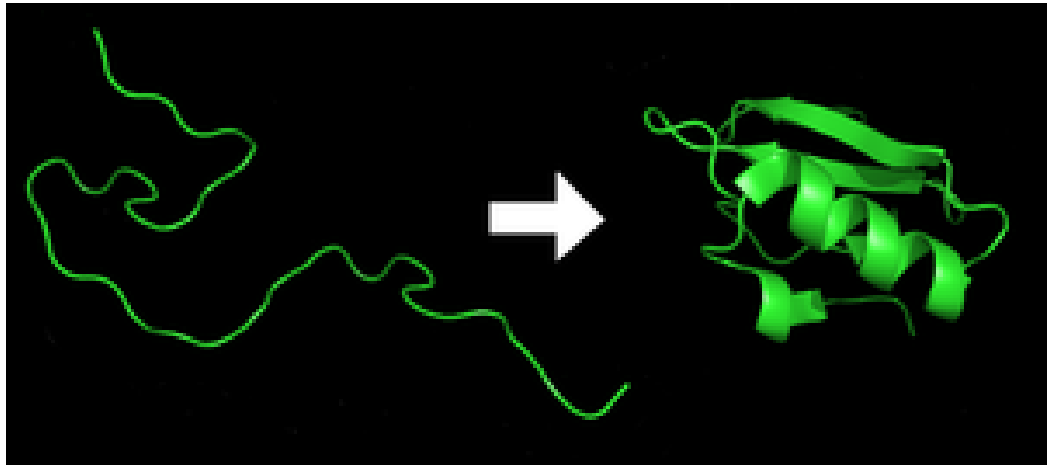
dihedral angle



https://www.quora.com/

Recap: Protein folding

- Physical process by which a polypeptide folds into its characteristic and functional three-dimensional structure

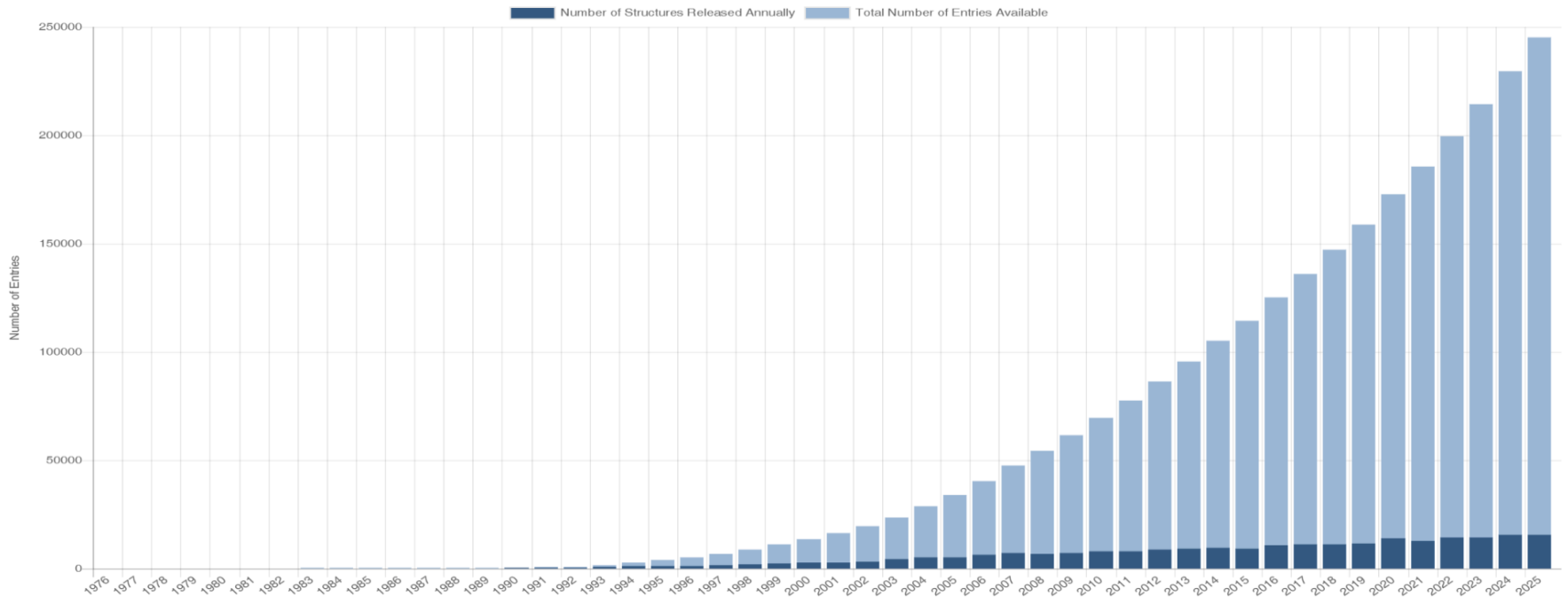


Folded proteins

Hydrophobic core -> side chain packing stabilizes the folded state

Charged or polar side chains occupies the solvent-exposed surface -> interact with surrounding water

Recap: Protein Data Bank (PDB)



Molecular Type	X-ray	EM	NMR	Integrative	Multiple methods	Neutron	Other	Total
Protein (only)	176,968	20,748	12,721	343	224	84	32	211,120
Protein/Oligosaccharide	10,297	3,419	34	8	11	1	0	13,770
Protein/NA	9,020	6,028	287	24	7	0	0	15,366
Nucleic acid (only)	3,101	210	1,557	3	15	3	1	4,890
Other	174	14	33	3	0	0	0	224
Oligosaccharide (only)	11	0	6	0	1	0	4	22
Total	199,571	30,419	14,638	381	258	88	37	245,392

Recap: PDB file format

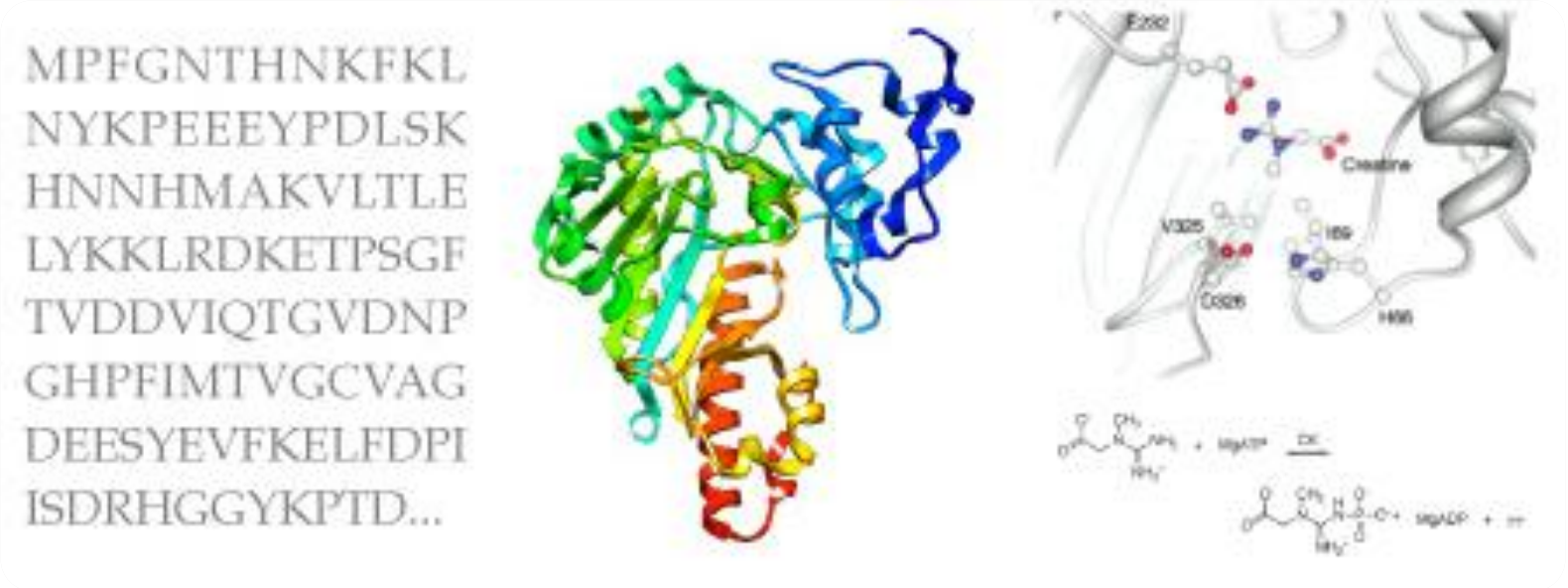
- http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html

Examples

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
MODEL        1
ATOM      1  N   ALA A   1      11.104   6.134  -6.504  1.00  0.00           N
ATOM      2  CA  ALA A   1      11.639   6.071  -5.147  1.00  0.00           C
...
...
...
ATOM    293 1HG  GLU A   18    -14.861  -4.847   0.361  1.00  0.00           H
ATOM    294 2HG  GLU A   18    -13.518  -3.769   0.084  1.00  0.00           H
TER     295      GLU A   18
ENDMDL
MODEL        2
ATOM    296  N   ALA  A   1      10.883   6.779  -6.464  1.00  0.00           N
ATOM    297  CA  ALA  A   1      11.451   6.531  -5.142  1.00  0.00           C
...
...
ATOM    588 1HG  GLU A   18    -13.363  -4.163  -2.372  1.00  0.00           H
ATOM    589 2HG  GLU A   18    -12.634  -3.023  -3.475  1.00  0.00           H
TER     590      GLU A   18
```

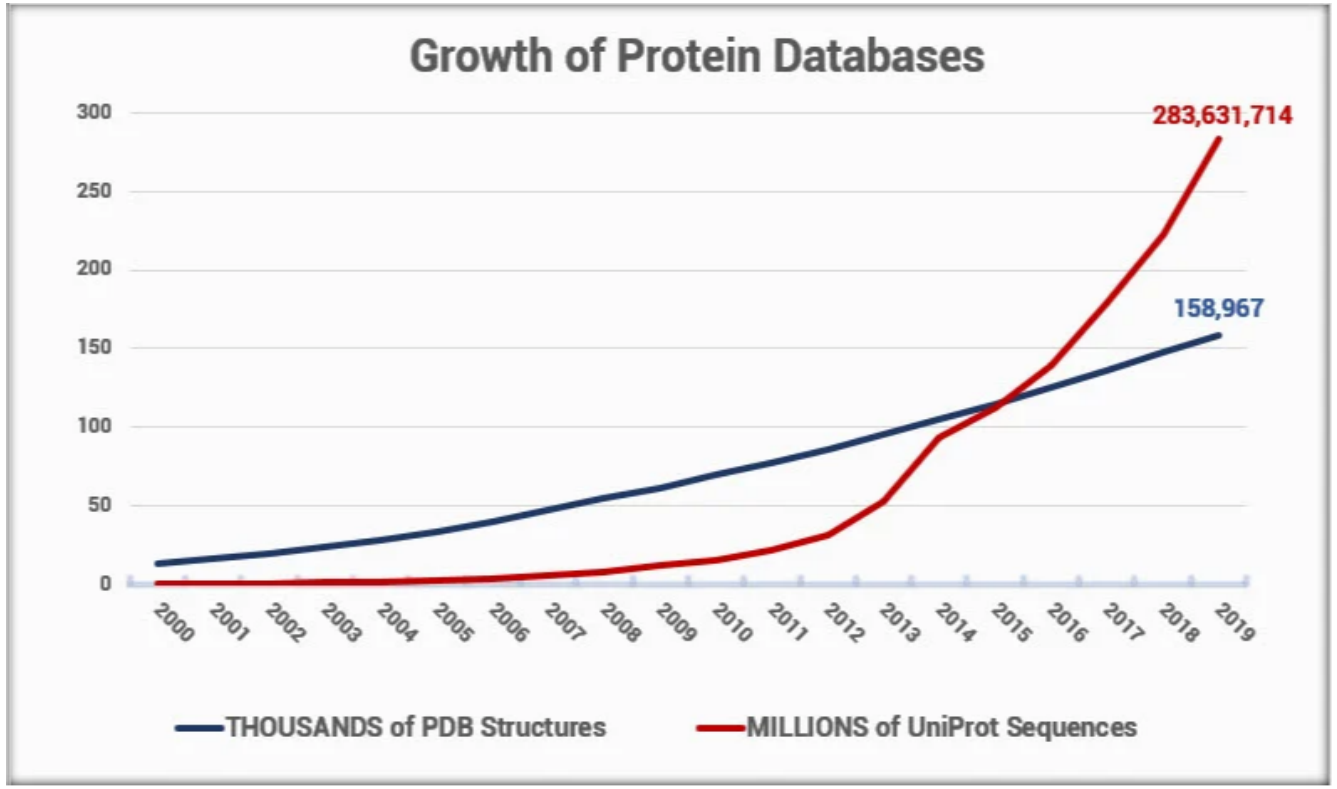
Structure prediction

- **Problem definition**: Given the sequence of a protein/DNA/RNA, predict its three-dimensional structure.
- **Application**:
  - Predicting the function of a biomolecule
  - Drug development



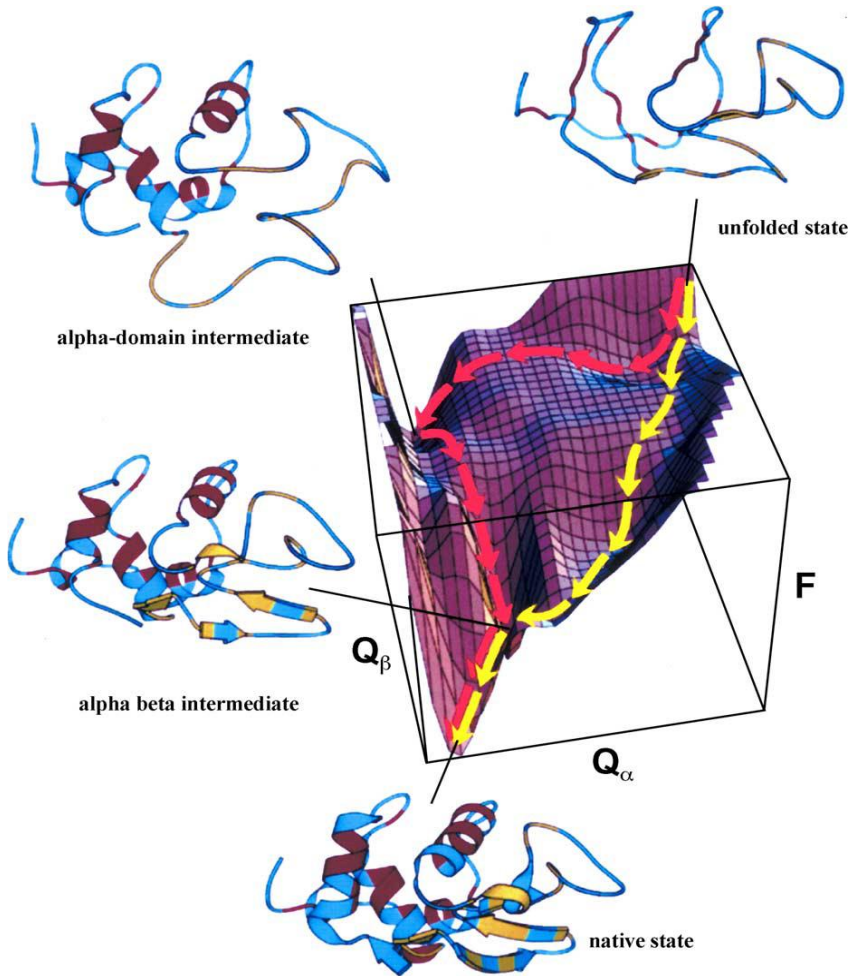
Why structure prediction matters?

- Some structures are very difficult to solve experimentally
- Sequence determination far outpaces experimental structure determination



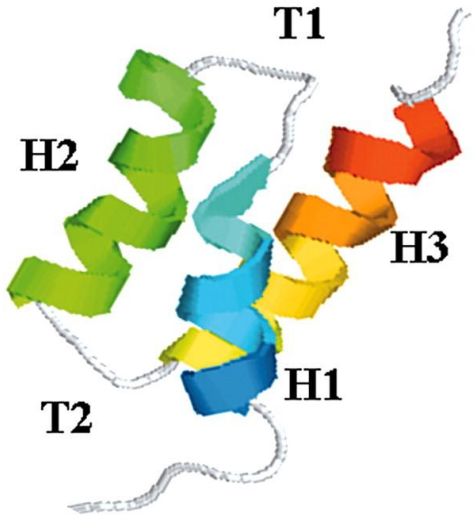
Notice that the scale of the red numbers is 1000x the scale of the blue numbers.

Levinthal paradox



- There are vastly too many different possible conformations for a protein to fold by a random search.
- Consider just for the peptide backbone, there are 3 conformations per amino acid in the unfolded state, For a 100 a.a. protein we have 3^100 conformations.
- If the chain can sample 10^12 conformations/sec, it takes 5 x 10^35 sec (2 x 10^28 years)
- Conclusion: Protein folding is not random, must have pathways.

Protein main chain secondary structure prediction



Main chain organization
(secondary structure)



helix

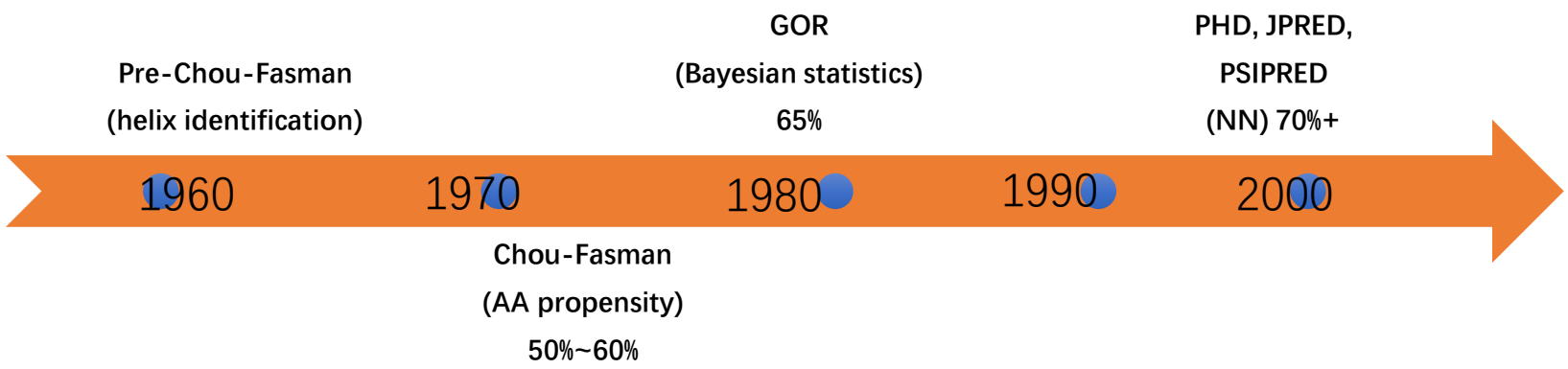


β-sheet

Random coil

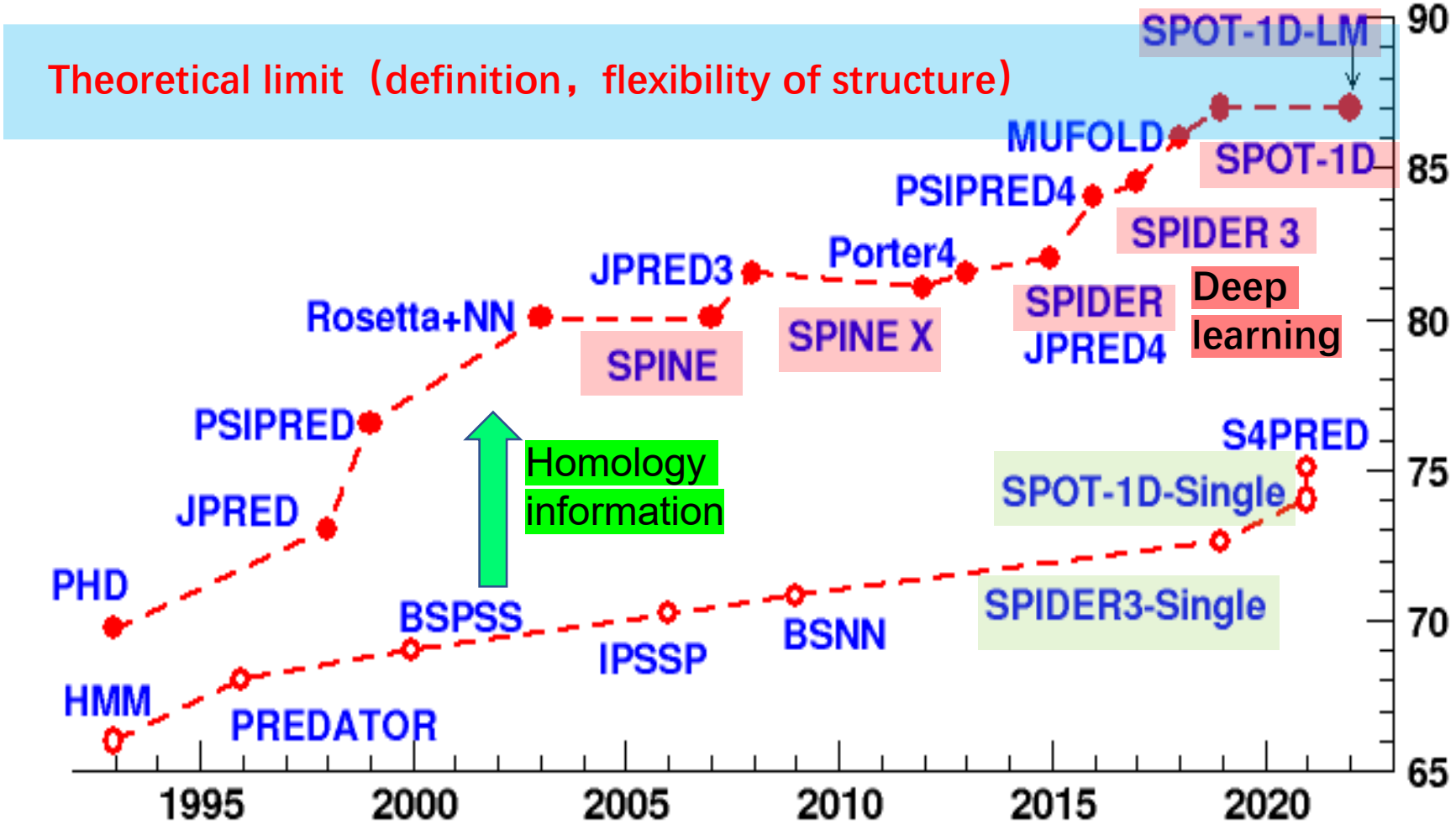
GHWIATRGQLIREAYEDYRHFSSECPFIP → CEEEEECHHHHHHHHHHHHCCCHHCCCCCC

Statistical learning

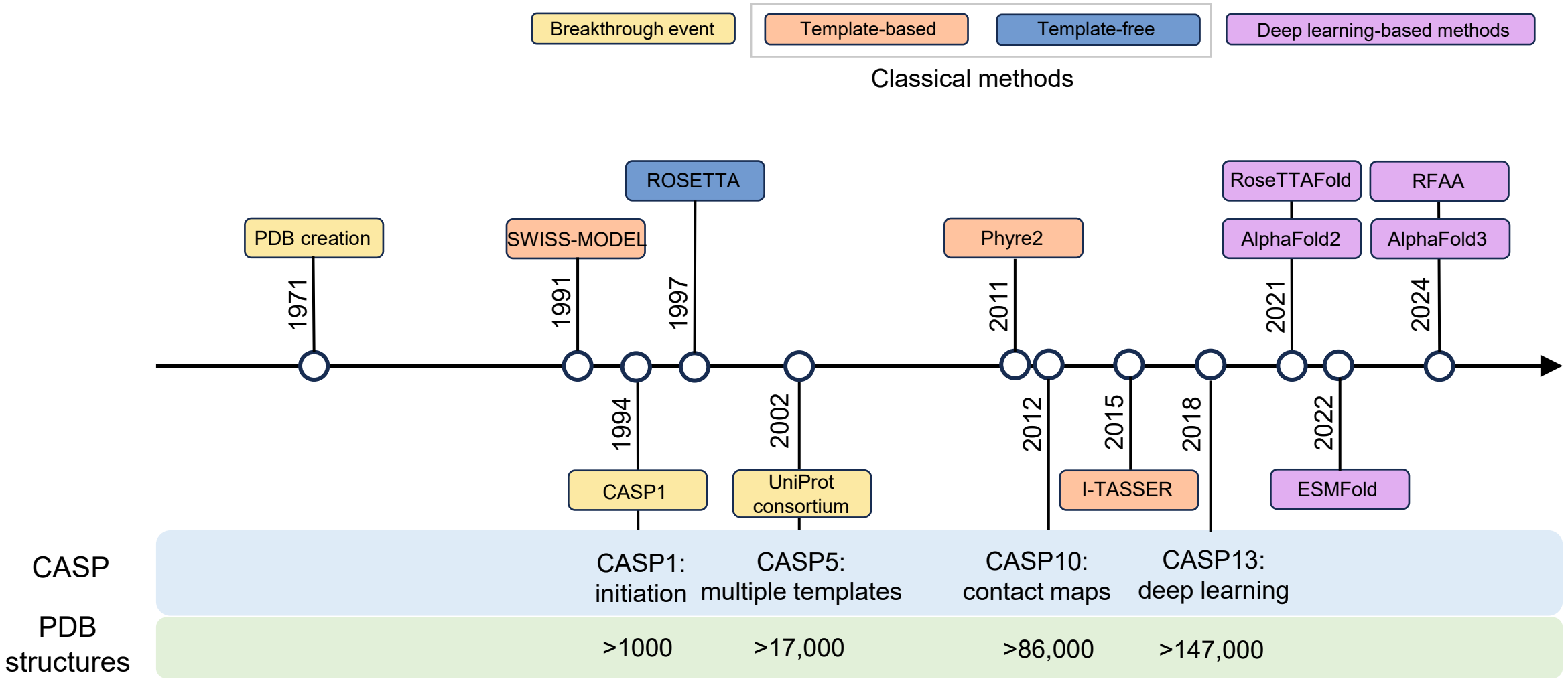


- Based on analyzing frequency of amino acids in different secondary structures
  - – A, E, L, and M strong predictors of alpha helices
  - – P and G are predictors in the break of a helix
- Table of predictive values created for alpha helices, beta sheets, and loops
- Structure with greatest overall prediction value greater than 1 used to determine the structure

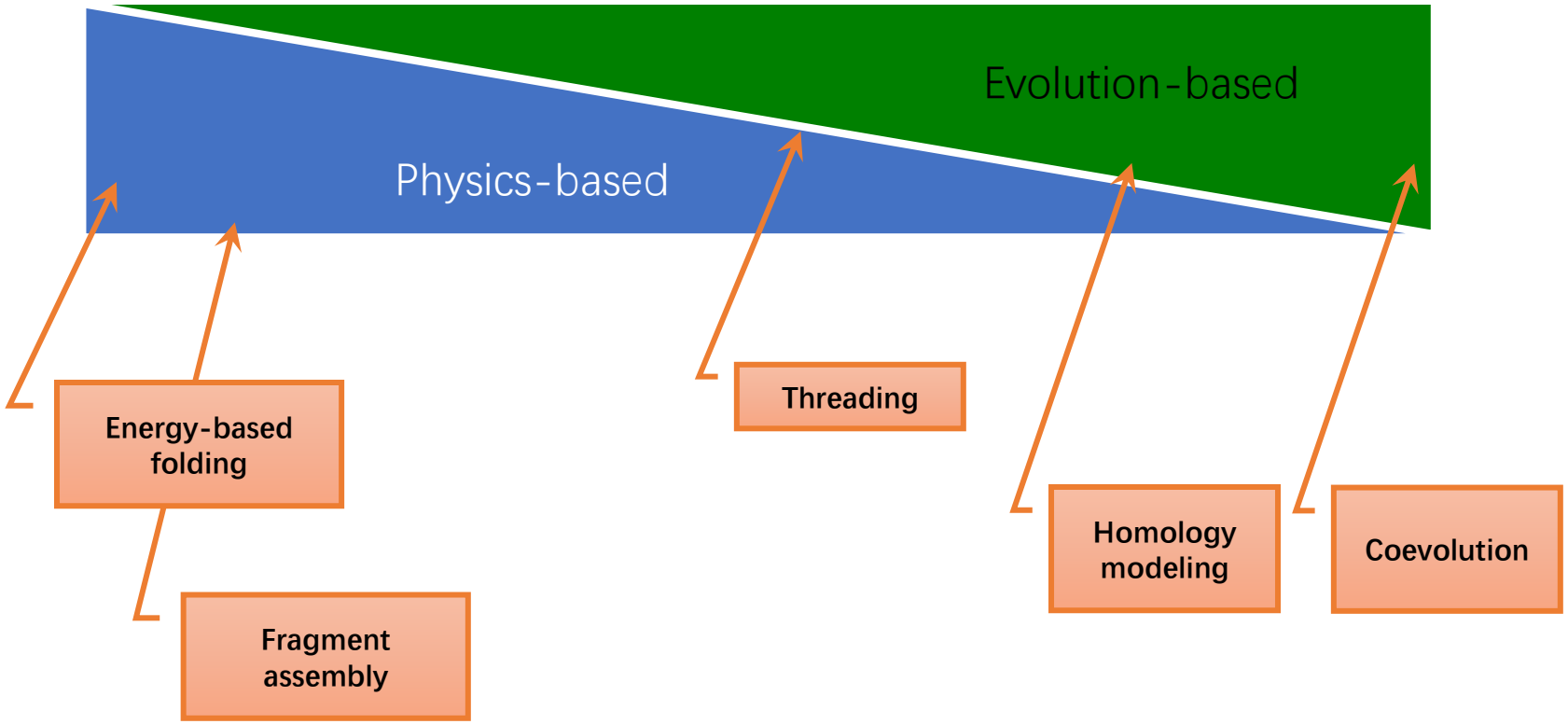
Protein main chain secondary structure prediction



Tertiary structure prediction timeline

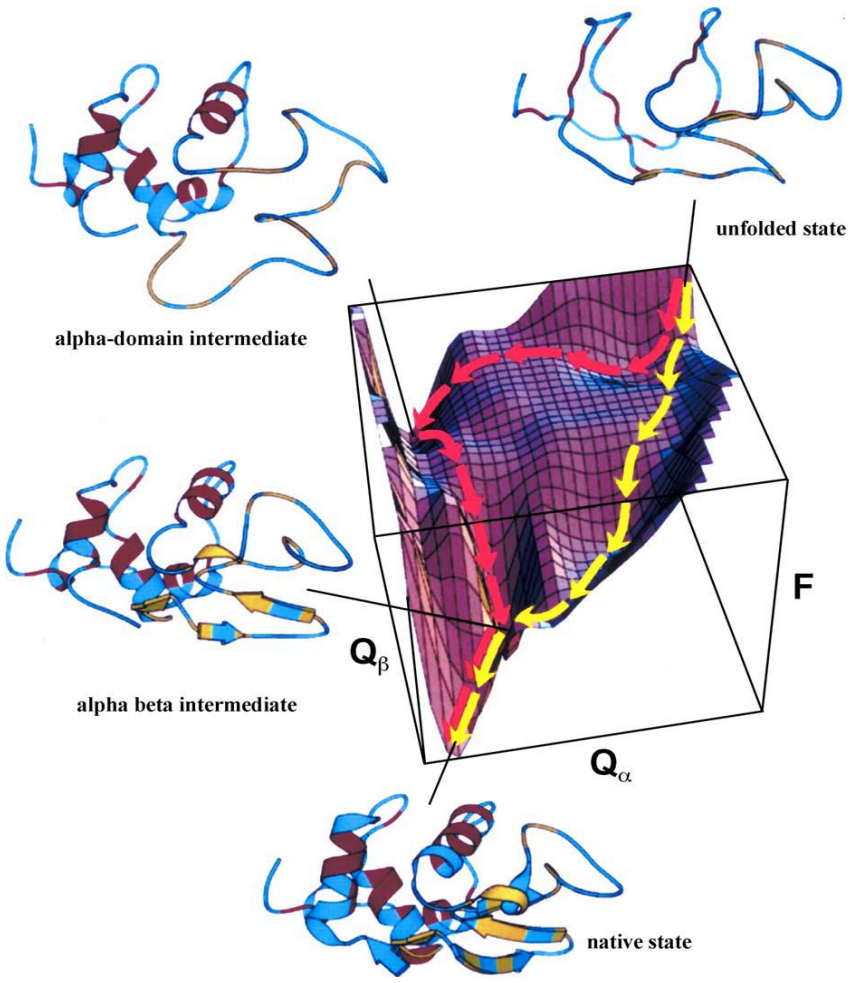


Classical tertiary structure prediction methods



Note: These classic structure prediction methods are often overlap, leading to hybrid approaches in modern computational biology.

Protein folding: force fields



$$V(\mathbf{r}) = \overset{(1)}{\sum_{\text{bonds}} K_b(b-b_0)^2} + \overset{(2)}{\sum_{\text{angles}} K_\theta(\theta-\theta_0)^2} + \overset{(3)}{\sum_{\text{dihedrals}} K_\chi(1+\cos(n\chi-\delta))} + \overset{(4)}{\sum_{\text{nonbonded-pairs},i,j}} \left[\frac{q_i q_j}{4\pi e_0 r_{ij}} - \varepsilon_{ij}\left\{\left(\frac{R_{\min ij}}{r_{ij}}\right)^{\overset{(5)}{12}} - 2\left(\frac{R_{\min ij}}{r_{ij}}\right)^{6}\right\}\right]$$

Energy dependencies on:

1. Bond length
2. Bond valence angle
3. Bond dihedral angle
4. Non-bonded electrostatic interactions
5. Non-bonded van-der Waals interactions

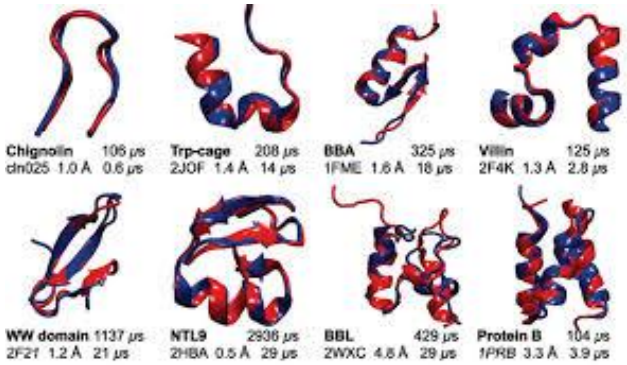
Amber、CHARMM、GROMOS、OPLS、…

From wall street to protein folding

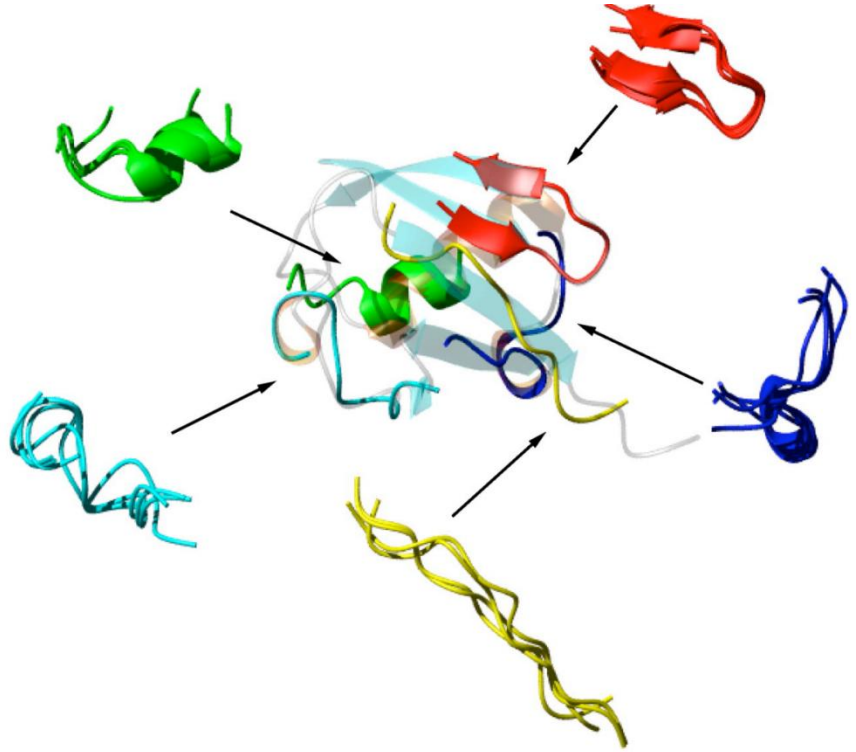








Fragment-assembly method: Rosetta



Key Idea: Rosetta searches structure space to find fragments in proper structures to assemble the whole structure

Fragment-assembly method: Rosetta

Given: protein sequence P

for each window of length 9 in P assemble a set of structure fragments

M = initial structure model of P (fully extended conformation); S = score(M)

while stopping criteria not met

randomly select a fixed width “window” of amino acids from P

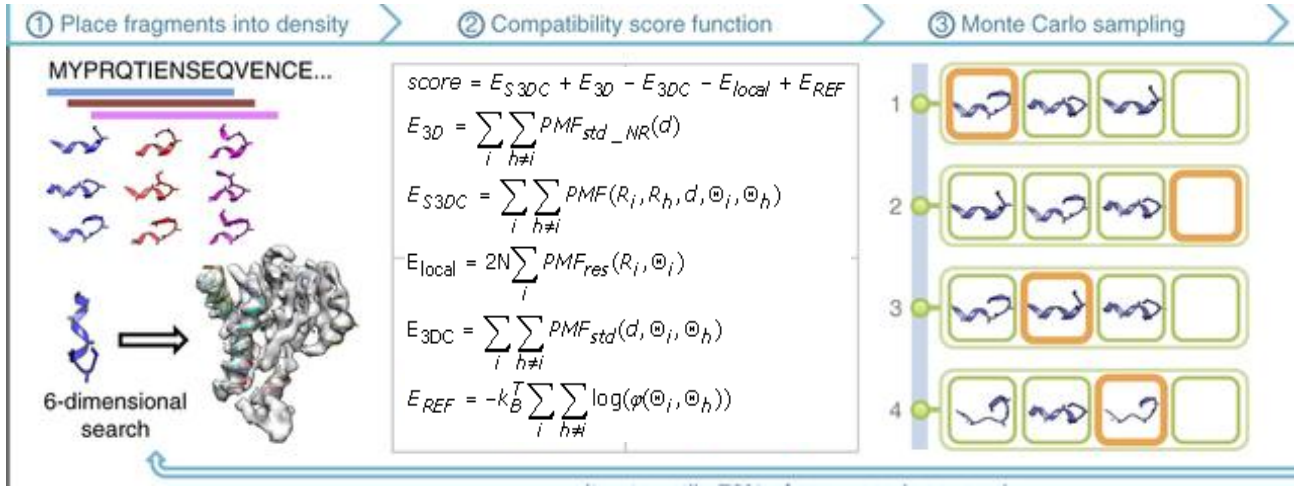
randomly select a fragment from the list for this window

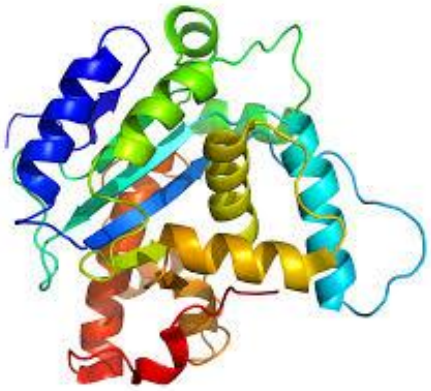
M’ = M with torsion angles in window replaced by angles from fragment; S’ = score(M')

if Metropolis criterion(S, S') satisfied

M = M’; S = S'

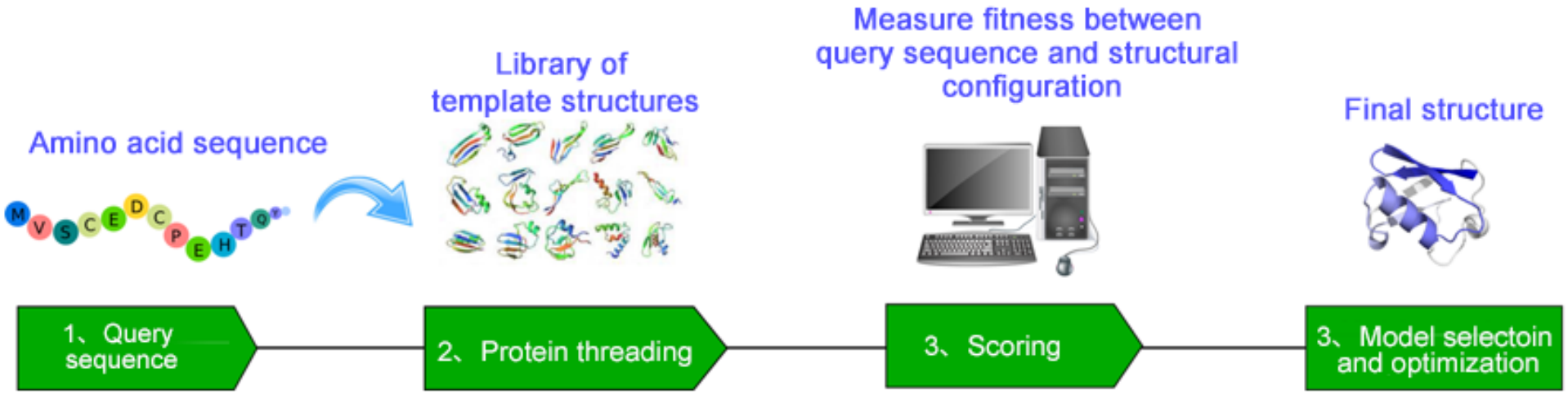
Return: predicted structure M



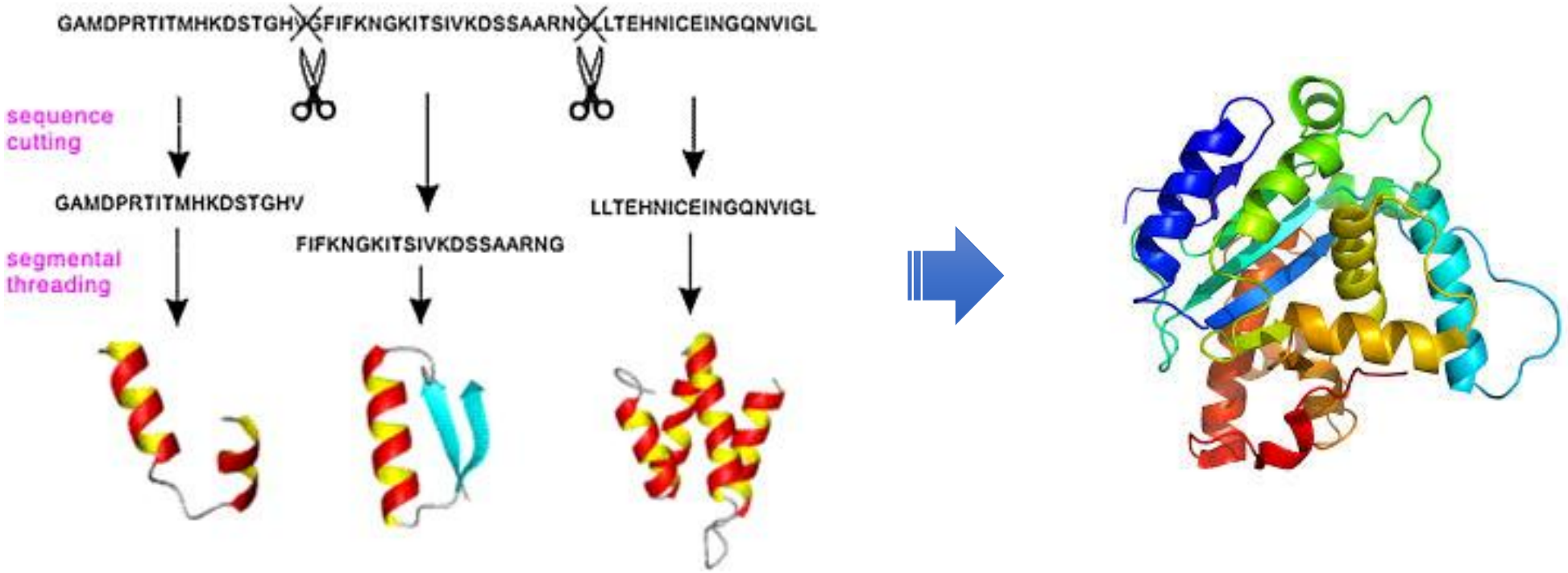


Threading

- **Threading**, also known as **fold recognition**, is a method of protein modeling which is used to model those proteins which have the same fold as proteins of known structures, but do not have homologous proteins with known structure.
- **Assumption**: the number of different folds in nature is fairly small.
- Structure template database: SCOP, CATH, …



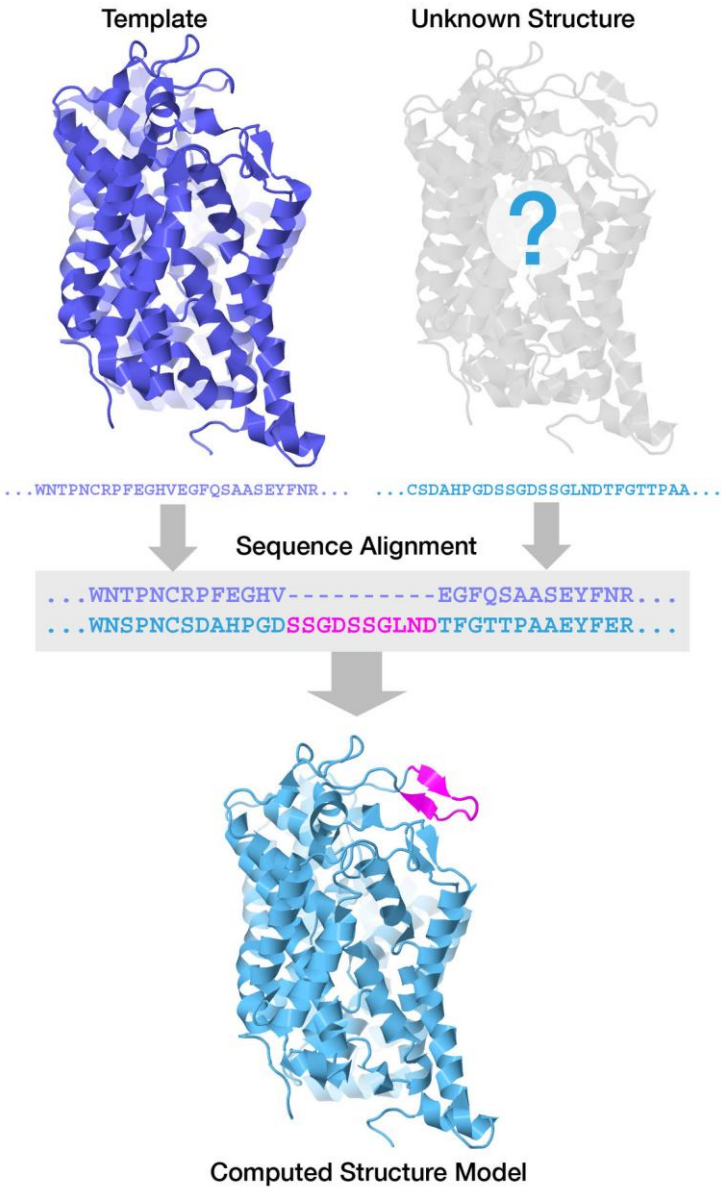
Threading framework



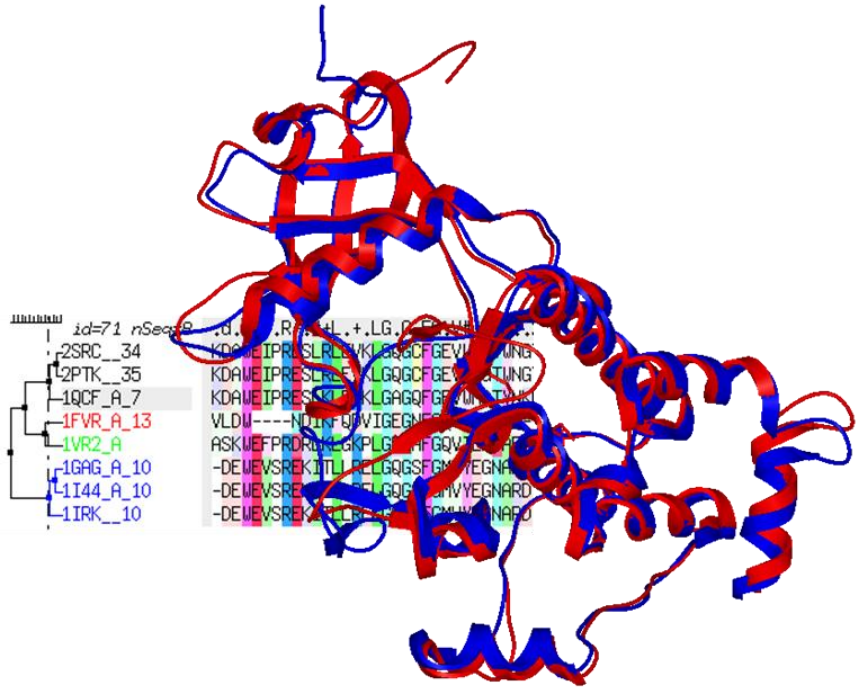
Query sequence => Search in library of folds (fold recognition) => Scoring and selection => Model (main chain) construction => Side chain packing => model refinement

Homology modeling

- **Homology modeling** refers to constructing an atomic-resolution model of the "**target**" protein from its amino acid sequence and an experimental three-dimensional structure of a related homologous protein (the "**template**").
- **Assumption**: protein tertiary structure is better conserved than amino acid sequence
- Sequence alignment algorithms: PSI-BLAST, HMM, Smith-Waterman algorithm



Homology modeling: key steps



Template identification

- Homology sequence search
- Template selection

Alignment

- often coupled with template selection
- Alignment correction

Model building

- Backbone only (how to deal with insertions and deletions ?)
- Single or multiple template approaches

Model refinement

- Loop modeling (ab initio)
- Sidechain packing (rotamer library)
- Possible molecular dynamics

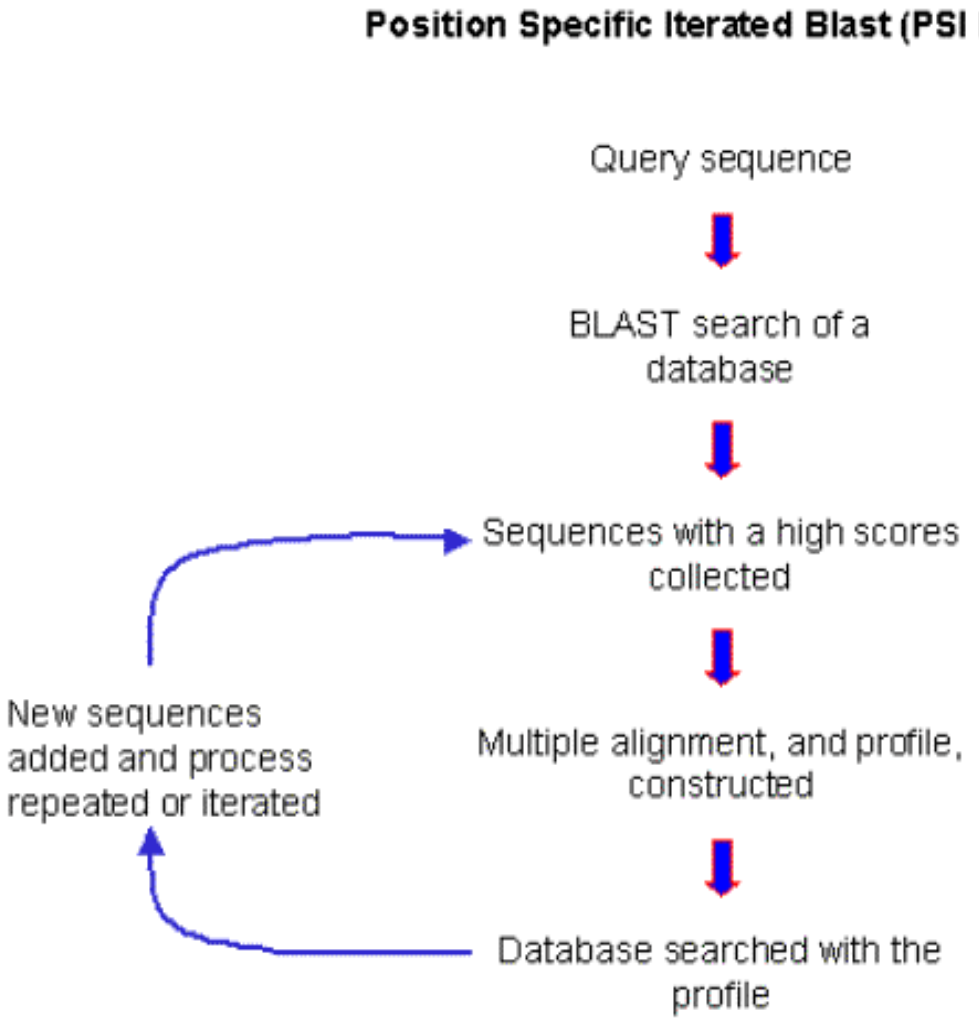
Model evaluation

- Quality score

Homology modeling: template recognition

- search the related proteins sequence(templates) to the target sequence in any structural database of proteins
  - Blast, Psi-Blast, Guided by secondary structure (fold recognition methods)
- Use biological information
- Functional annotation in databases
- Active site/motifs
- **Selecting the best template is crucial!**

PSI-BLAST



PSI-BLAST Profile

```
  ACSDSASSSVNYMYRLVMNSPRSTVLMNSPQN...

A 12132931112123133131232132123132 1...
C 29232312312321332132121212312 1...
D 12312313231232321212213213213213...
E 12131321321321321321231212121313...
F 21321321231212313132321321233211...
G 13212332132132123333343123123213...
H 12321341322132212123141212131222...
I 11321313231421213213212131321232...
K 12321312312323233211322313213232...
L 31321321212212332323222132321213...
M 11132123312293132132131321212313...
N 13213322239321231232123123321222...
P 13223332311313123213333333213913...
Q 31212231213212312313211113321231...
R 22213132132131313233322213313133...
S 12119199923131313332213132313123...
T 33231222213312231213232212221321...
V 13313222192213132121331233123121...
W 33121323122122331232122313213231...
Y 11212312122929121321231213213231...
```

Homology modeling: loop modeling and refinement

- Knowledge based:
  - Searches PDB for fragments that match the sequence to be modelled (Honig, Levitt, Holm, Baker etc.).
- Energy based:
  - Uses an energy function to evaluate the quality of the loop and minimizes this function by Monte Carlo (sampling) or molecular dynamics (MD) techniques.
- Rosetta approach:
  - Find fragments (10 per amino acid) with the same sequence and secondary structure profile as the query sequence.Combine them using a Monte Carlo scheme to build the loop.

- Energy minimization
- Molecular dynamics
  - *Big errors like atom clashes can be removed, but force fields are not perfect and small errors will also be introduced – keep minimization to a minimum or matters will only get worse.*

Comparison of classical structure prediction methods

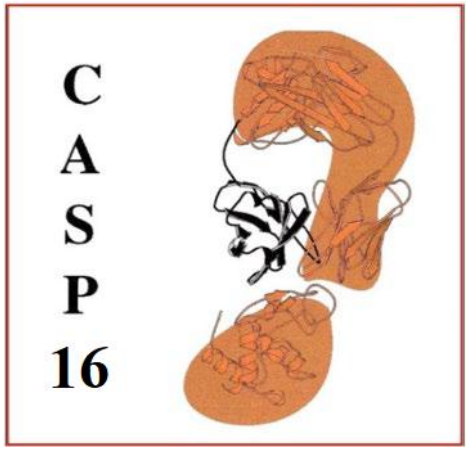
Methodology	Mechanism	Computational Cost	Typical Application	Accuracy Dependence
Ab initio modeling	Energy minimization	Extremely high	Small proteins, novel folds, absence of templates	Accuracy of forcefield and sampling efficiency
Fragment assembly	Sampling and energy minimization	High	Small to large proteins, novel folds, absence of templates	Accuracy of forcefield and sampling efficiency
Threading	Fold recognition	Moderate to high	Target with low sequence identity but similar topology	Size of fold library and optimized scoring function
Homology modeling	Template matching	Moderate	Target with high sequence identity (>30%) to known structure	Quality of template and alignment

Practice

- Build a structural model for protein HNRNPK.
  - Which method will you use? Why?
  - Does the structural model tell you anything about the protein?

CASP: Critical Assessment of Structure Prediction

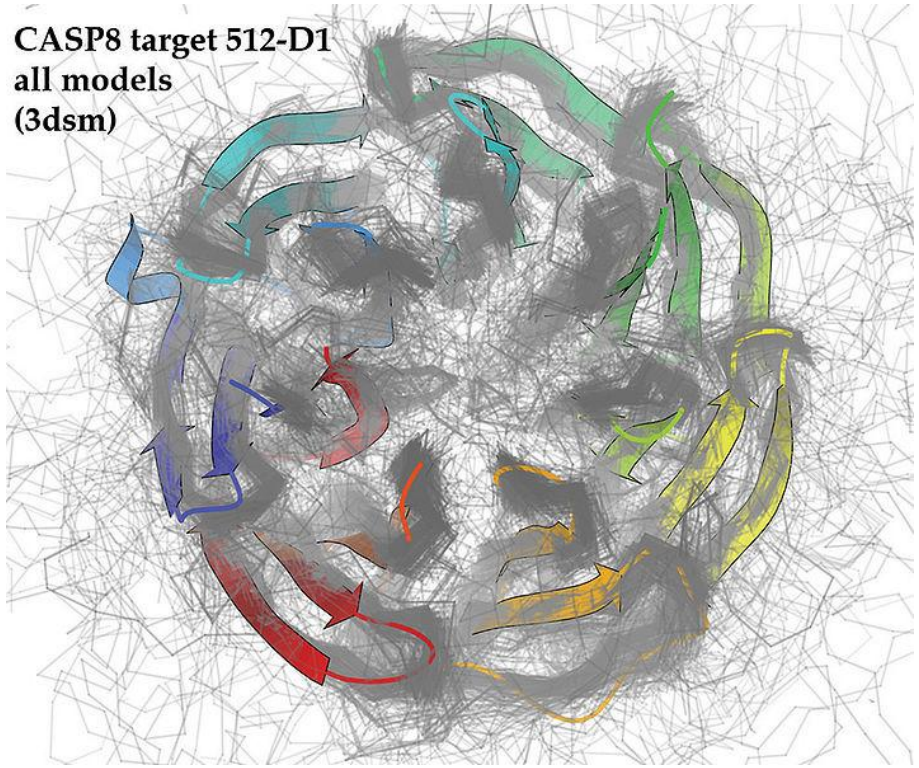
- Structural biologists provide structures prior to their publication for blind prediction by participants.
- Participants are given the primary sequences; their submitted 3D structures are then assessed by comparison to the correct structures.
- “World cup” of structure prediction.

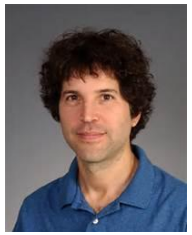


MSKGEELFTGVVPILVELDGDVNGHKFSVSGEGEGDATYGKLT
LKFICTTGKLPVPWPTLVTTFTYGVQCFSRYPDHMKRHDFFKS
AMPEGYVQERTIFFKDDGNYKTRAEVKFEGDTLVNRIELKGIDF
KEDGNILGHKLEYNYNSHNVYIMADKQKNGIKVNFKIRHNIED
GSVQLADHYQQNTPIGDGPVLLPDNHYLSTQSALSKDPNEKR
DHMVLLEFVTAAGITHGMDELYK



CASP: Critical Assessment of Structure Prediction





David Baker
@UWash

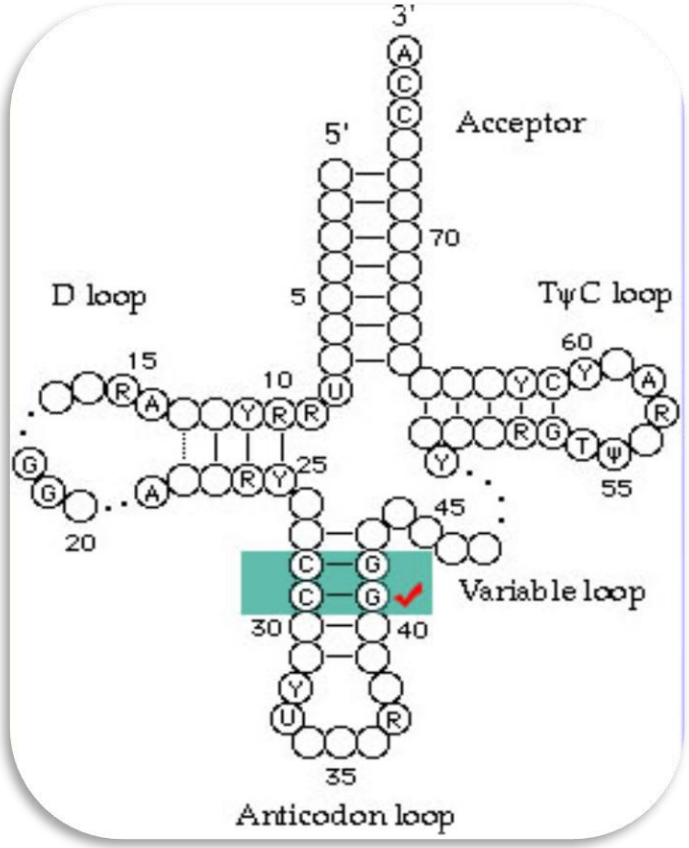


Yang Zhang
@UMich


WHAT IS PAST
IS PROLOGUE

The very success of tRNA structure prediction





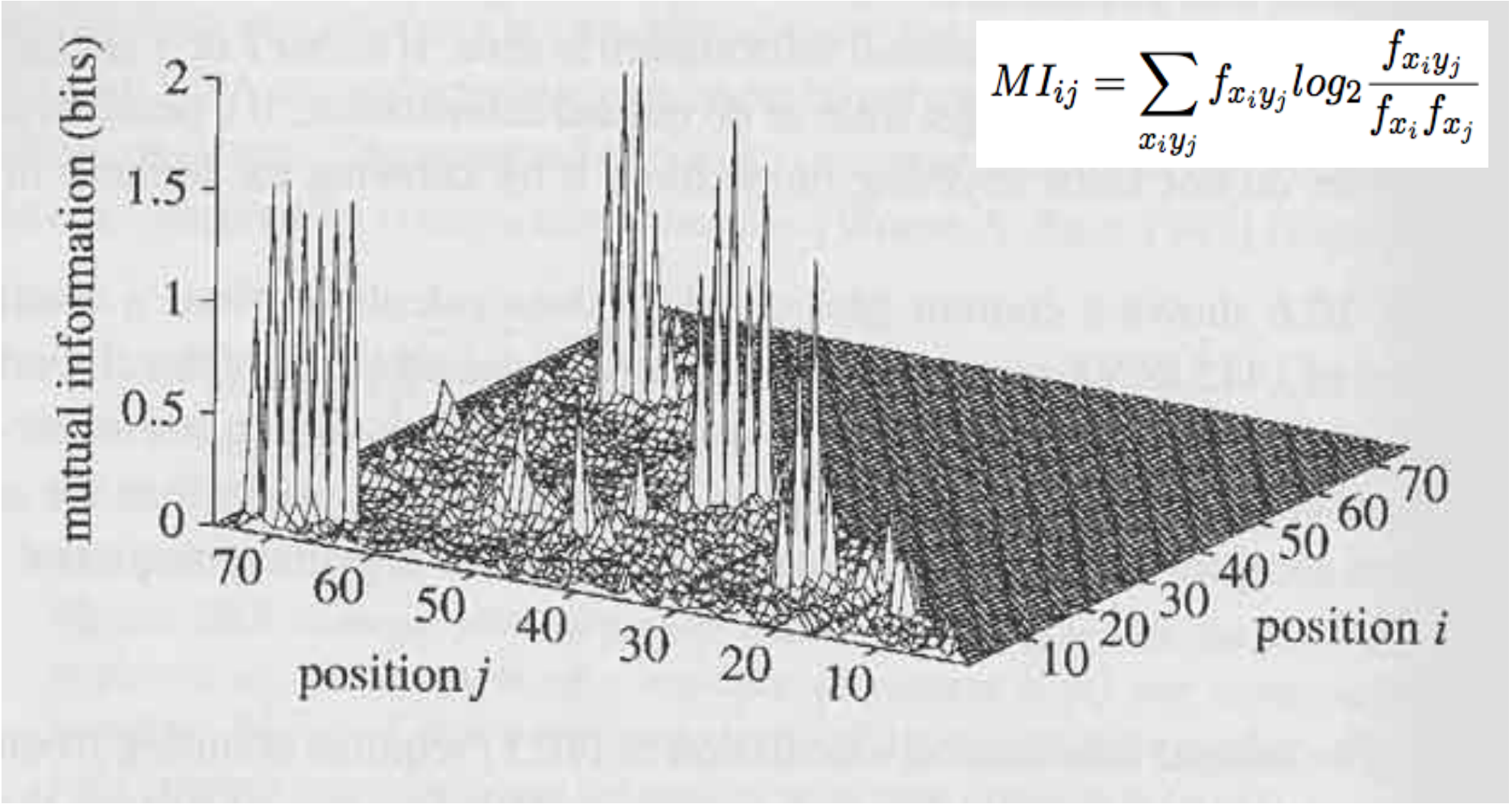
- Very basic assumption: structures are more conserved than sequences
- Example: tRNA
- Measure of covariance
  - Mutual information (desirable for large datasets)
    - Most common measurement
    - Used in CM (Covariance Model) for structure prediction
  - Covariance score (better for small datasets)

Covariance in multiple alignment

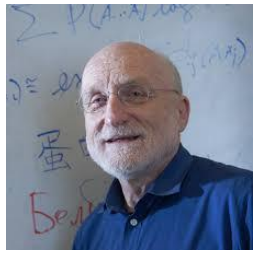


Covariance in multiple alignment

$$MI_{ij} = \sum_{x_i y_j} f_{x_i y_j} log_2 \frac{f_{x_i y_j}}{f_{x_i} f_{x_j}}$$

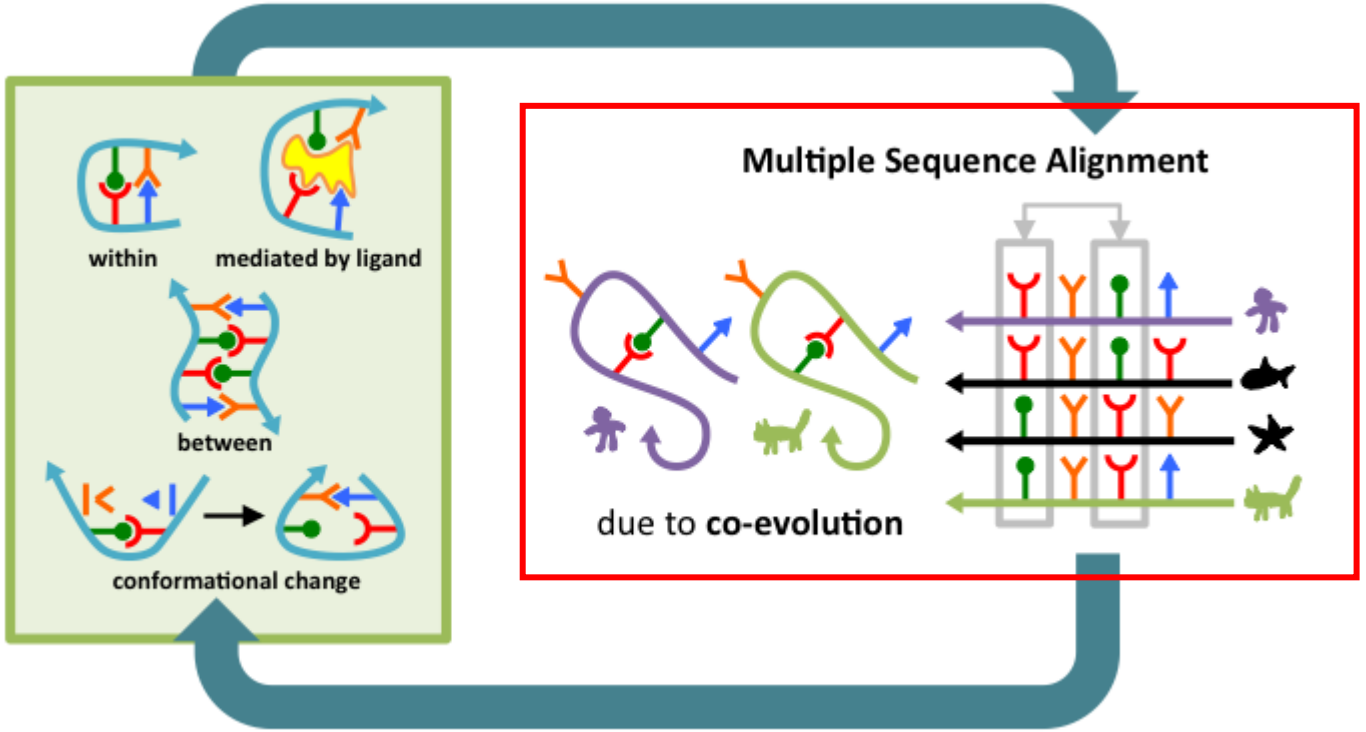


Co-evolution patterns and structures





Important **contacts** in proteins, are evolutionarily conserved and encoded in a

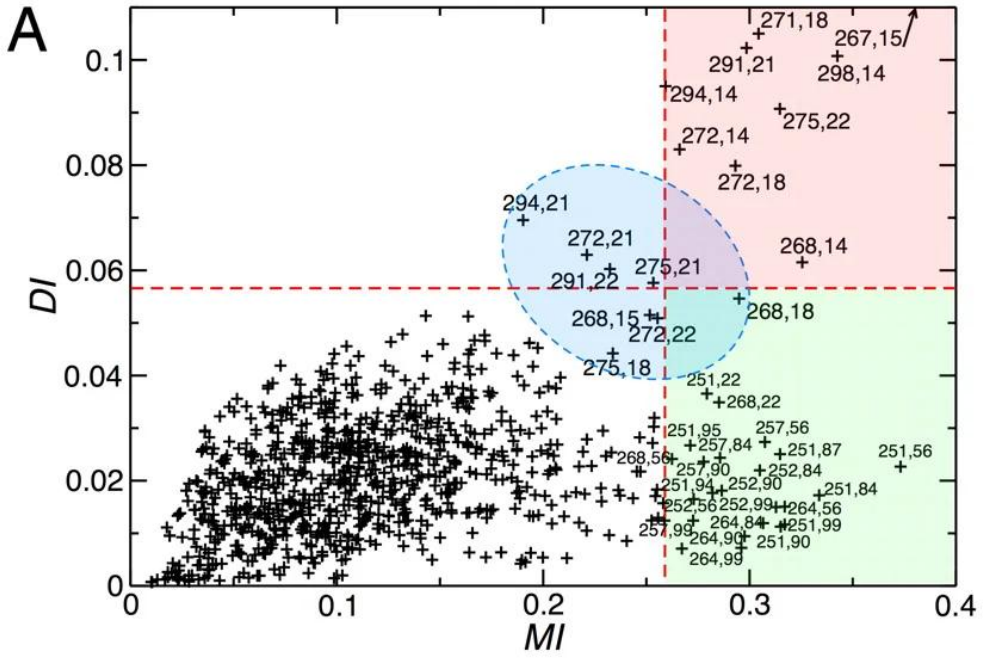


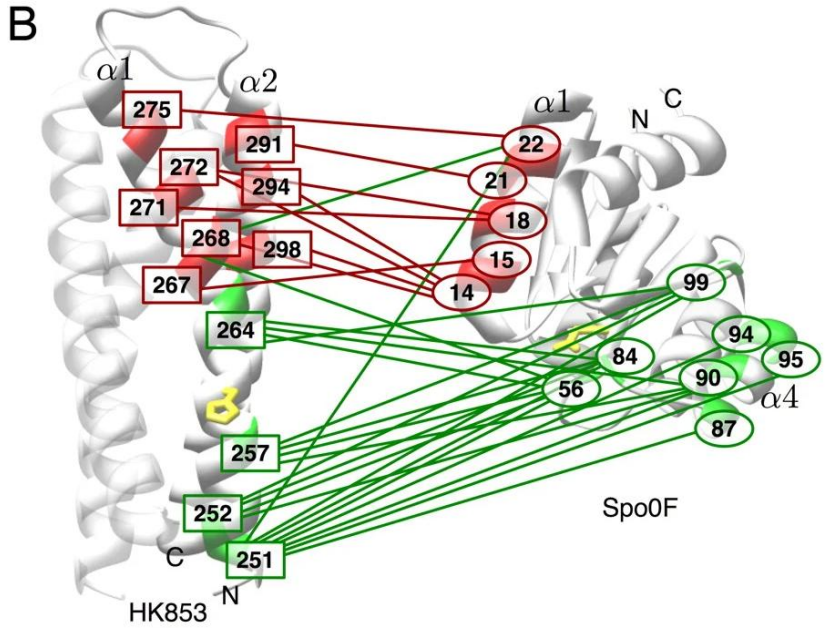
We can extract co-evolution information for contact prediction from MSA

By measuring **coevolution**, we can infer important contacts in proteins!

Direct and indirect interactions







Potts model

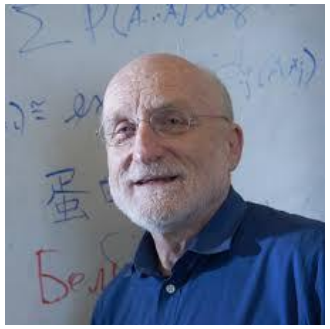
$$P(A_1, \cdots, A_N) = \frac{1}{Z} \exp\left\{ -\sum_{i<j} e_{ij}(A_i, A_j) + \sum_i h_i(A_i) \right\}$$

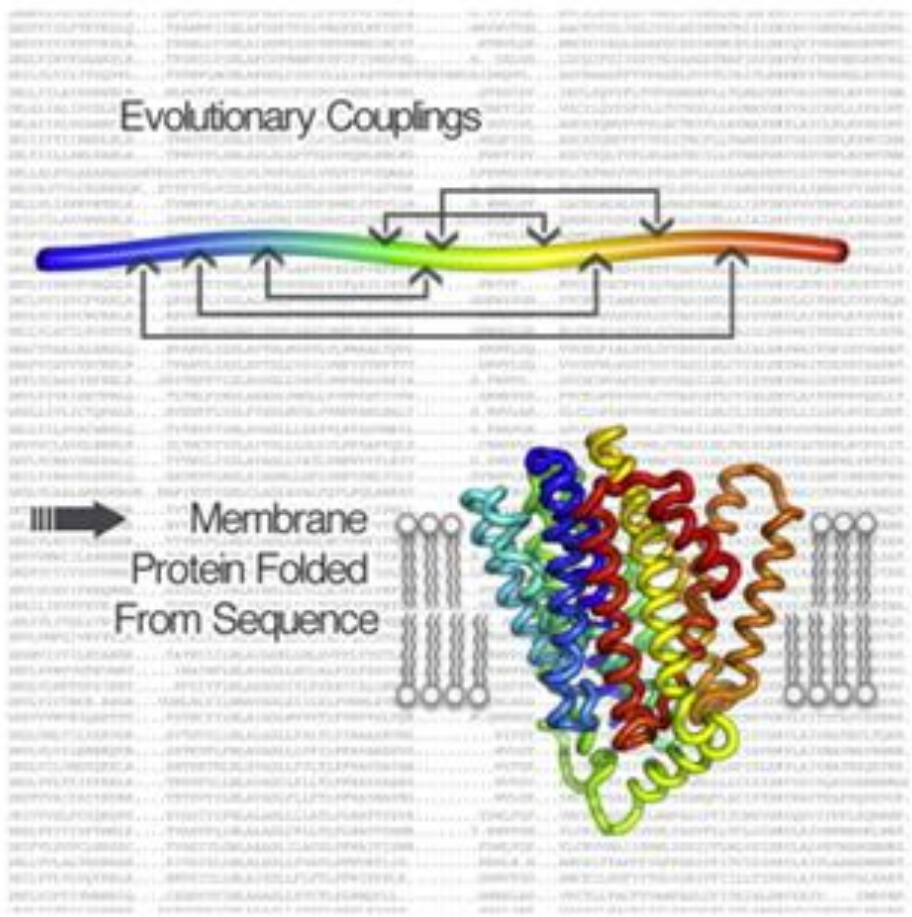
EM algorithm

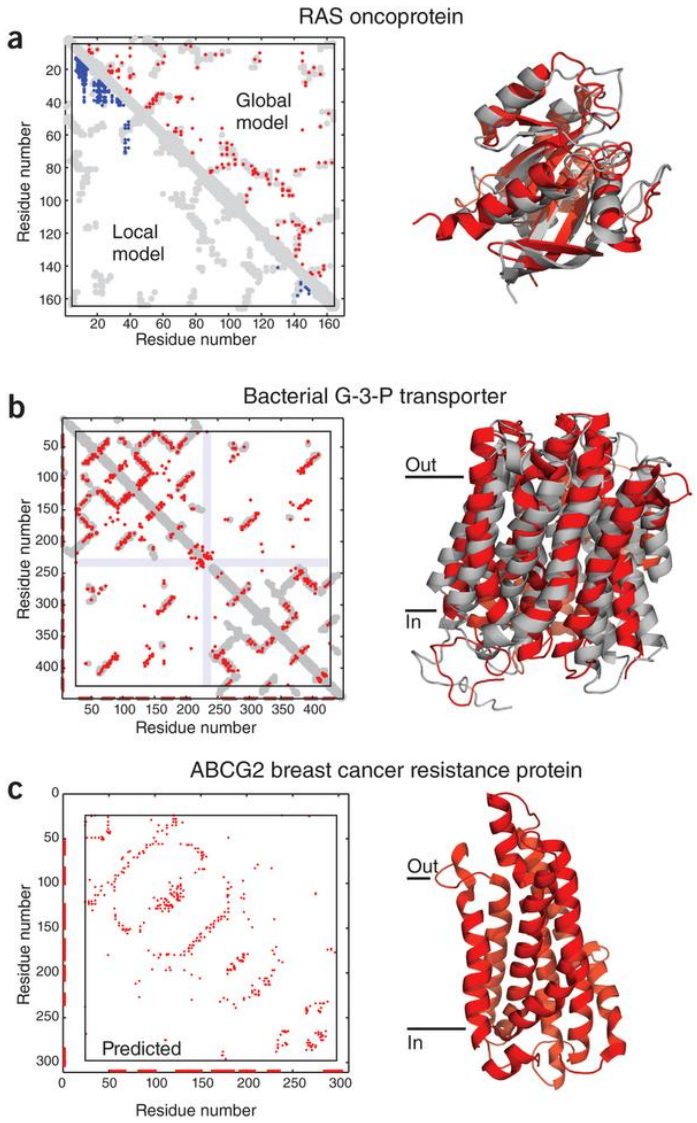
$$e_{ij}^{(\text{new})}(A_i, A_j) = e_{ij}^{(\text{old})}(A_i, A_j) + \Delta[f_{ij}(A_i, A_j) - P_{ij}(A_i, A_j)]$$

PNAS, Vol. 106, No. 1, Pages 67-72.

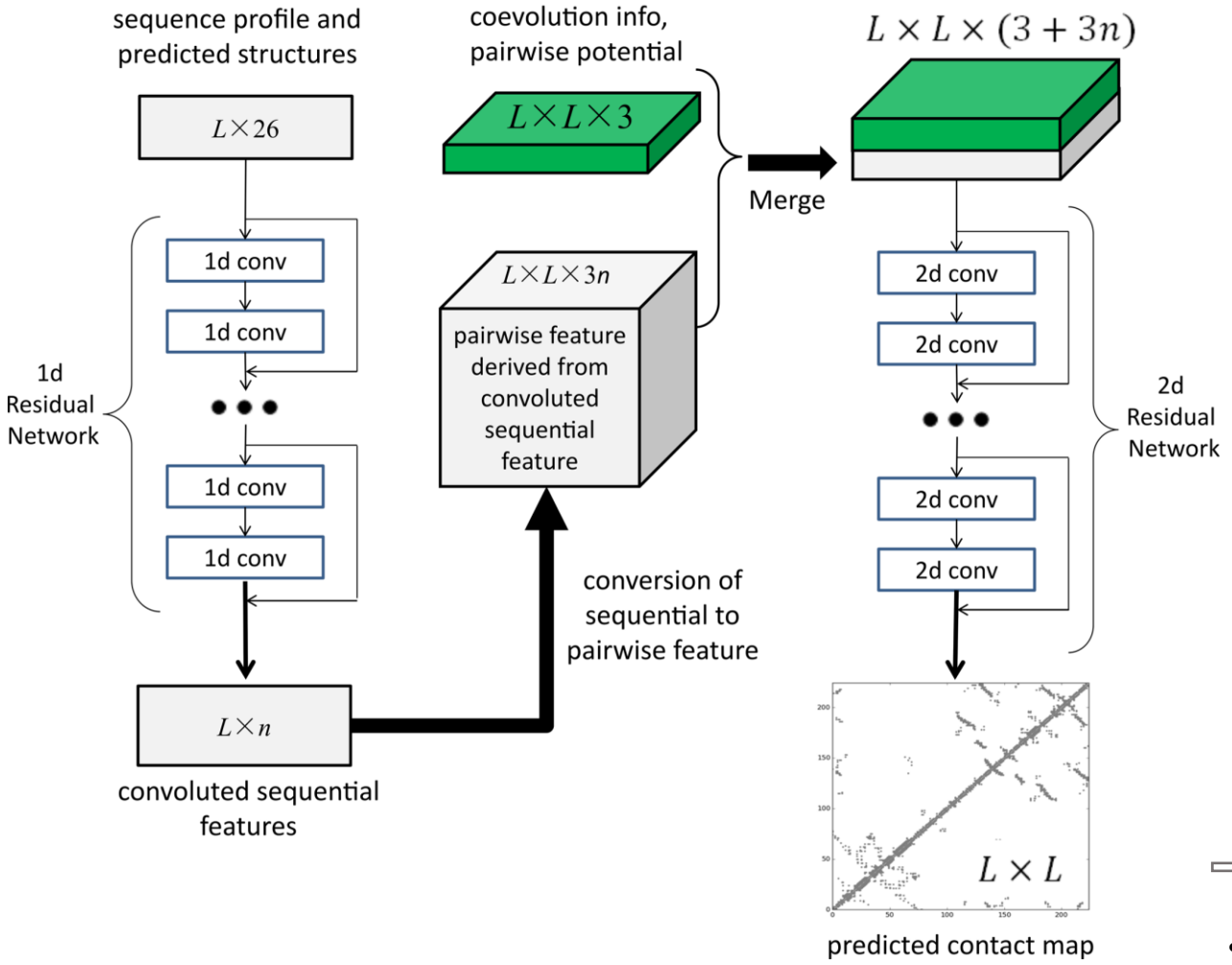
For membrane protein structure prediction







The first attempt of deep learning





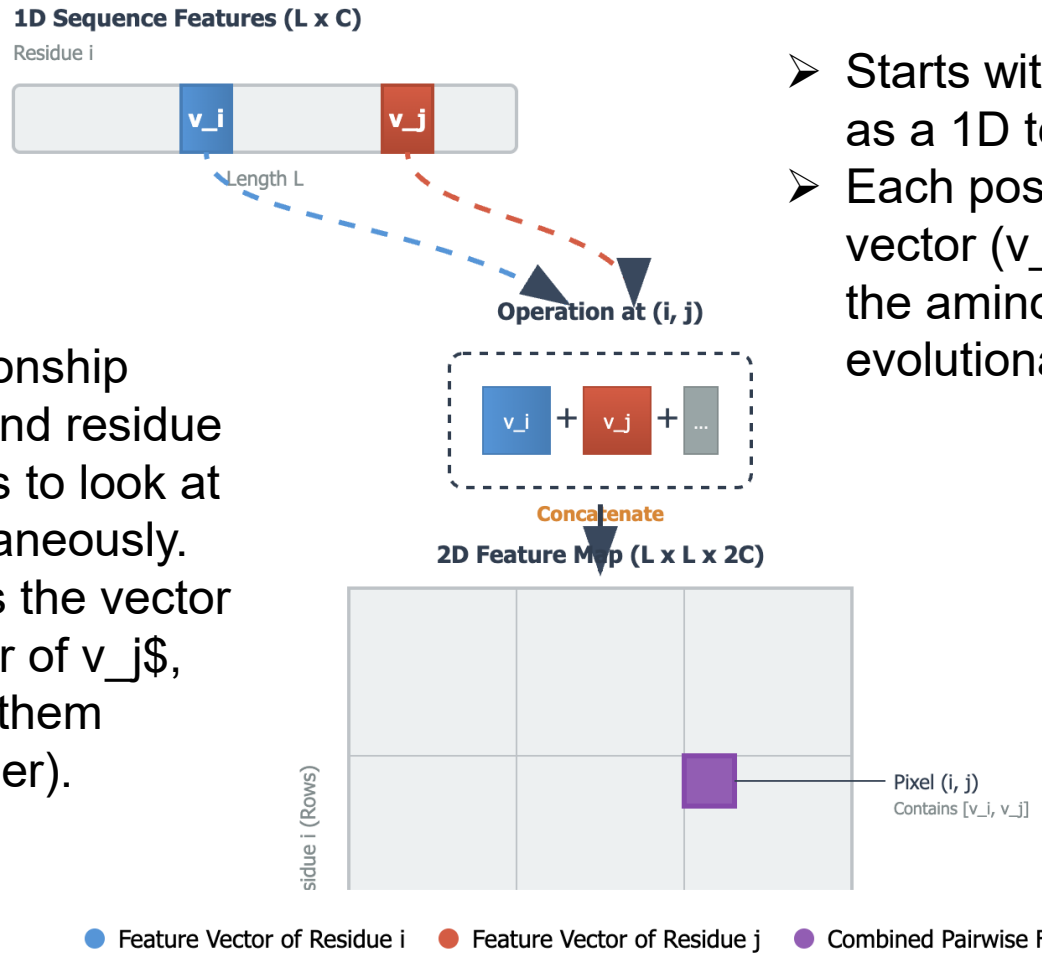
Group Name	Rank	F1	Precision	Recall
RaptorX-Contact	1	12.386	55.831	7.029
MetaPSICOV	2	10.919	51.307	6.155
iFold_1	3	10.917	50.916	6.153
MULTICOM-CONSTRUCT	4	10.847	50.449	6.121
Pcons-net	5	10.810	49.536	6.113
RBO-Epsilon	6	10.736	48.601	6.081
FALCON_COLORS	7	10.387	47.253	5.880
Yang-Server	8	10.186	46.460	5.763
Deepfold-Contact	9	10.003	46.442	5.644
PconsC31	10	9.734	45.728	5.483
IGBteam	11	9.428	45.596	5.288
MULTICOM-CLUSTER	12	9.114	42.661	5.138
naive	13	9.016	42.009	5.085
raghavagps	14	9.005	40.269	5.110
Shen-Group	15	8.984	40.938	5.085
AkbAR	16	8.901	41.417	5.021
MULTICOM-NOVEL	17	8.235	36.630	4.680
Zhang_Contact	18	8.111	38.259	4.570
PLCT	19	7.690	35.133	4.356

⟹ **Folding in CNS or Rosetta**

- ResNet
- Dilated Convolution
- DCA output

Improve DCA estimation with 2D patterns

Outer Concatenation: from individual residues to pairs



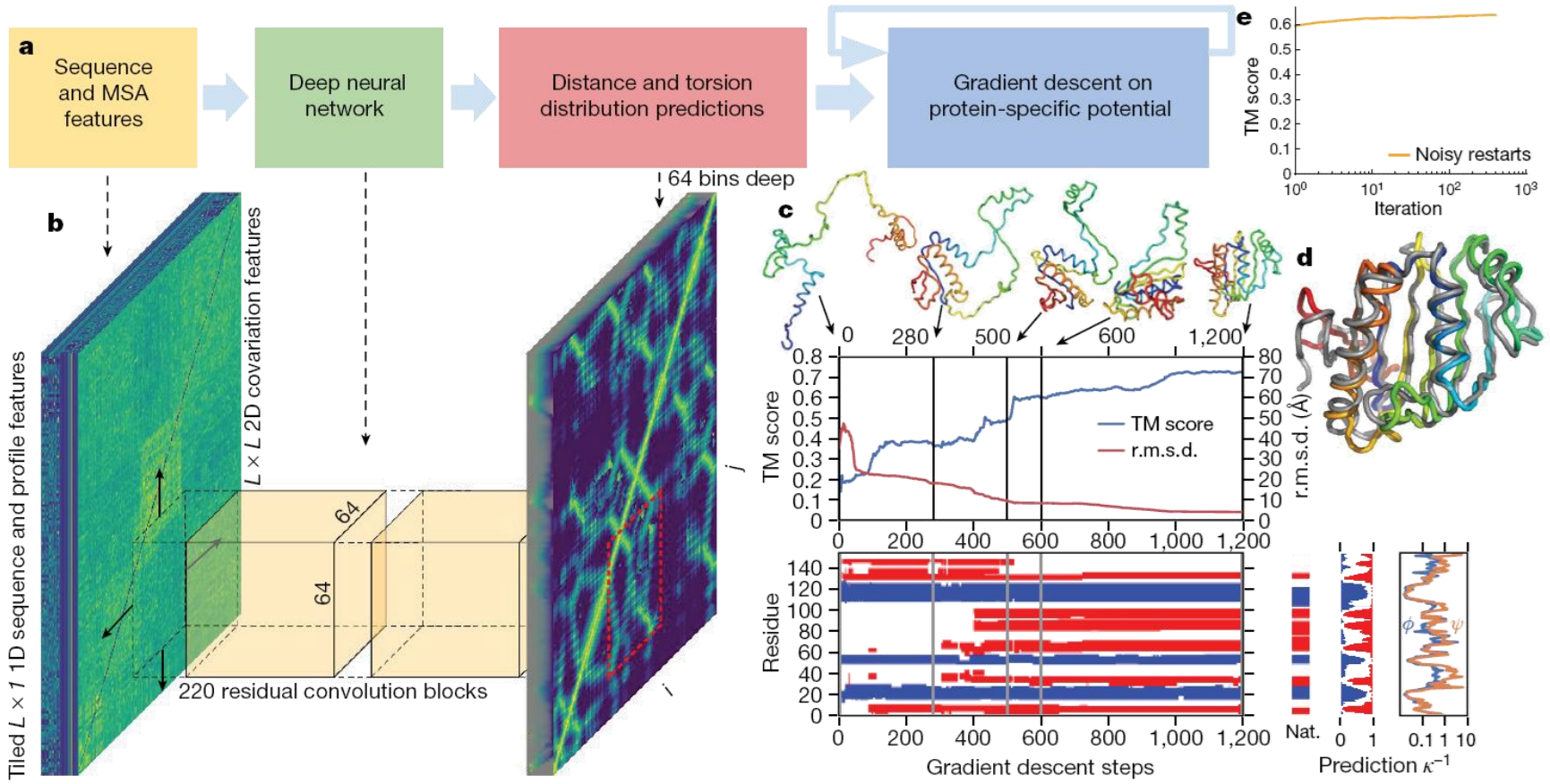
- Starts with the protein sequence represented as a 1D tensor.
- Each position in the sequence has a feature vector (v_i). These features usually include the amino acid type, secondary structure, and evolutionary information (PSSM).

- To predict the relationship between residue *i* and residue *j*, the network needs to look at both of them simultaneously.
- The operation takes the vector of v_i and the vector of v_j$, and **concatenates** them (stacks them together).

- The result is a 2D Feature Map (an image-like grid of size $L \times L$).
- The specific pixel at coordinate (i, j) now contains the combined information of both residues
- This 2D map is then fed into 2D Networks (e.g., ResNets) to predict contact probability

AlphaFold architecture

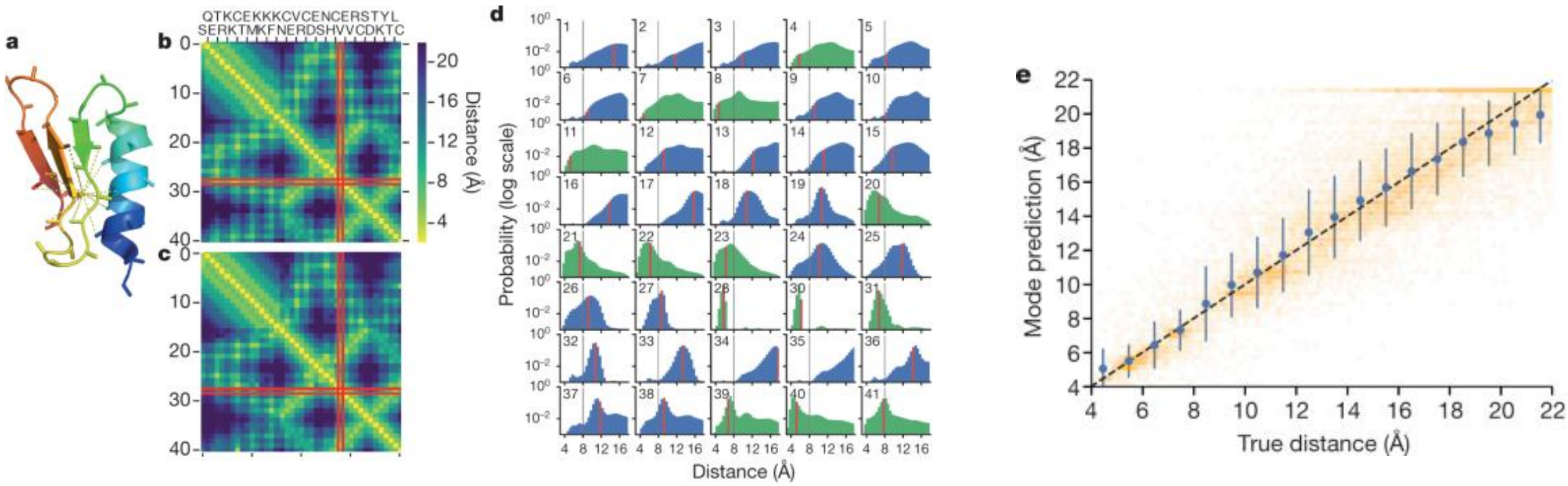




- From contact map to distance map
- Folding with Gradient Descent (L-BFGS)
- Torsion angle prediction
- Improved AI engineering (more complex network, better MSA, model ensemble)

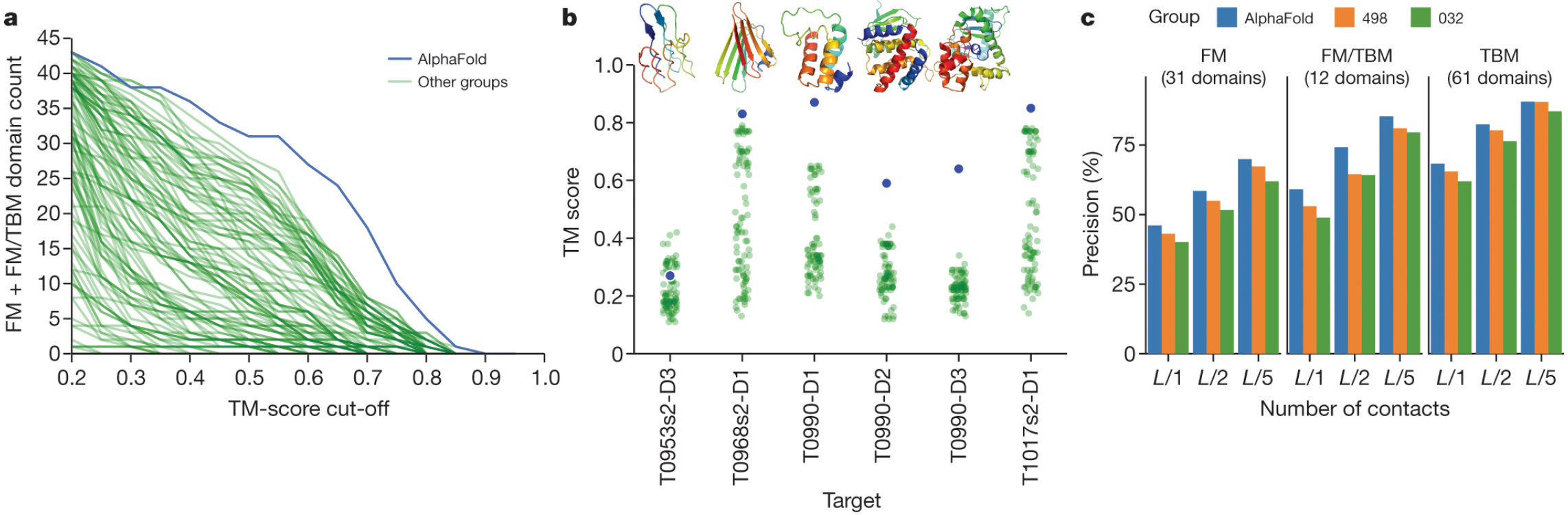
Nature, 2020

Predicted distance distributions compared with true distances

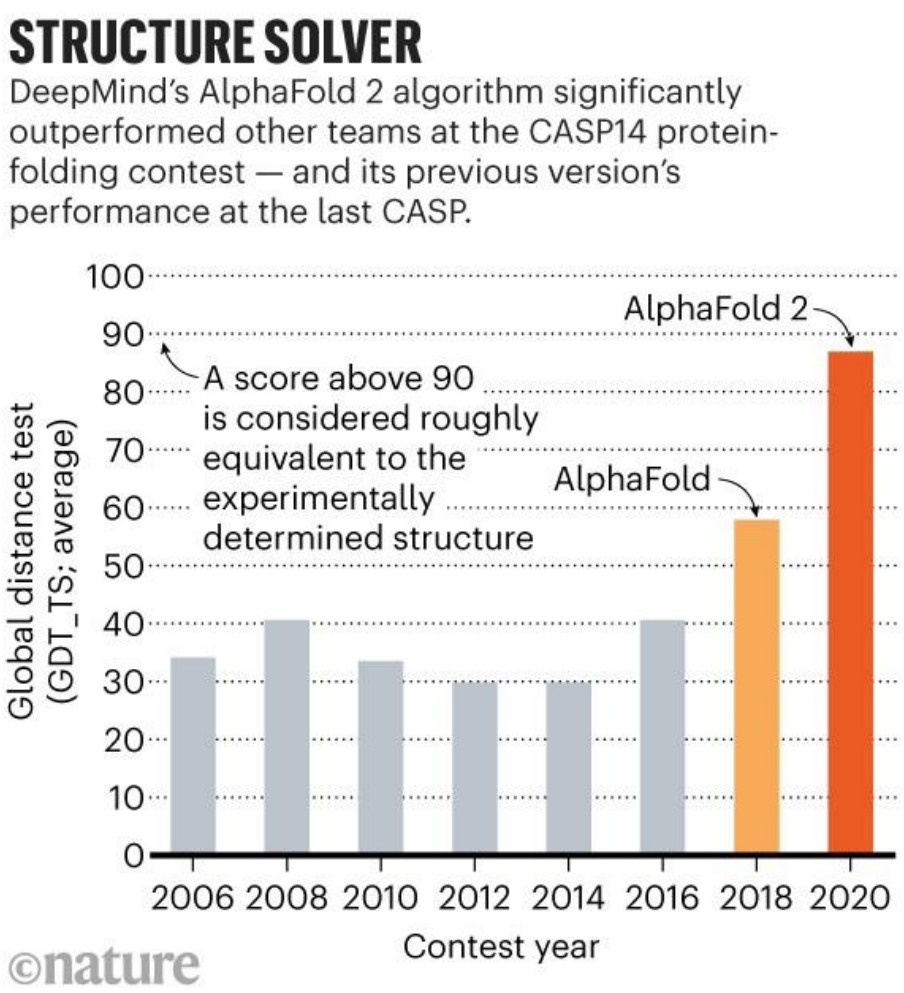


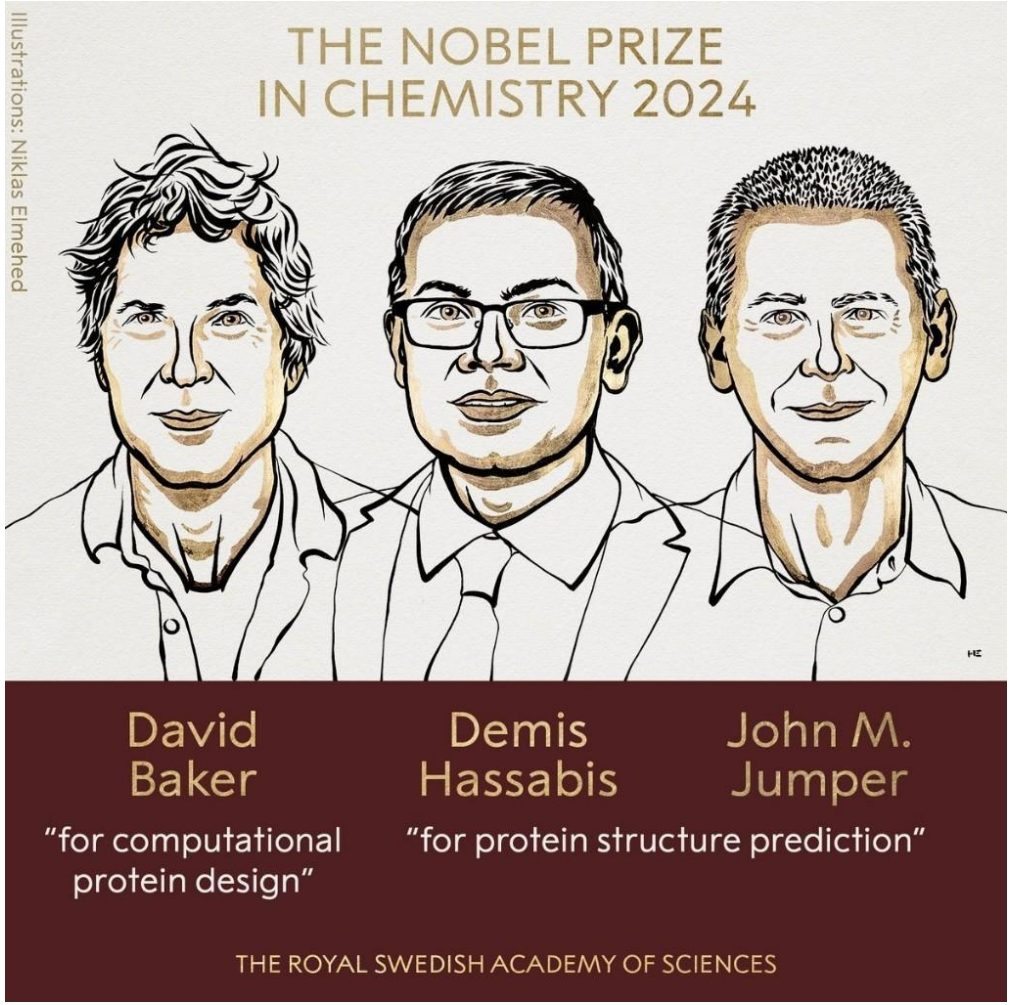
AF1 accurately predicts residue distance

The performance of AlphaFold in CASP13



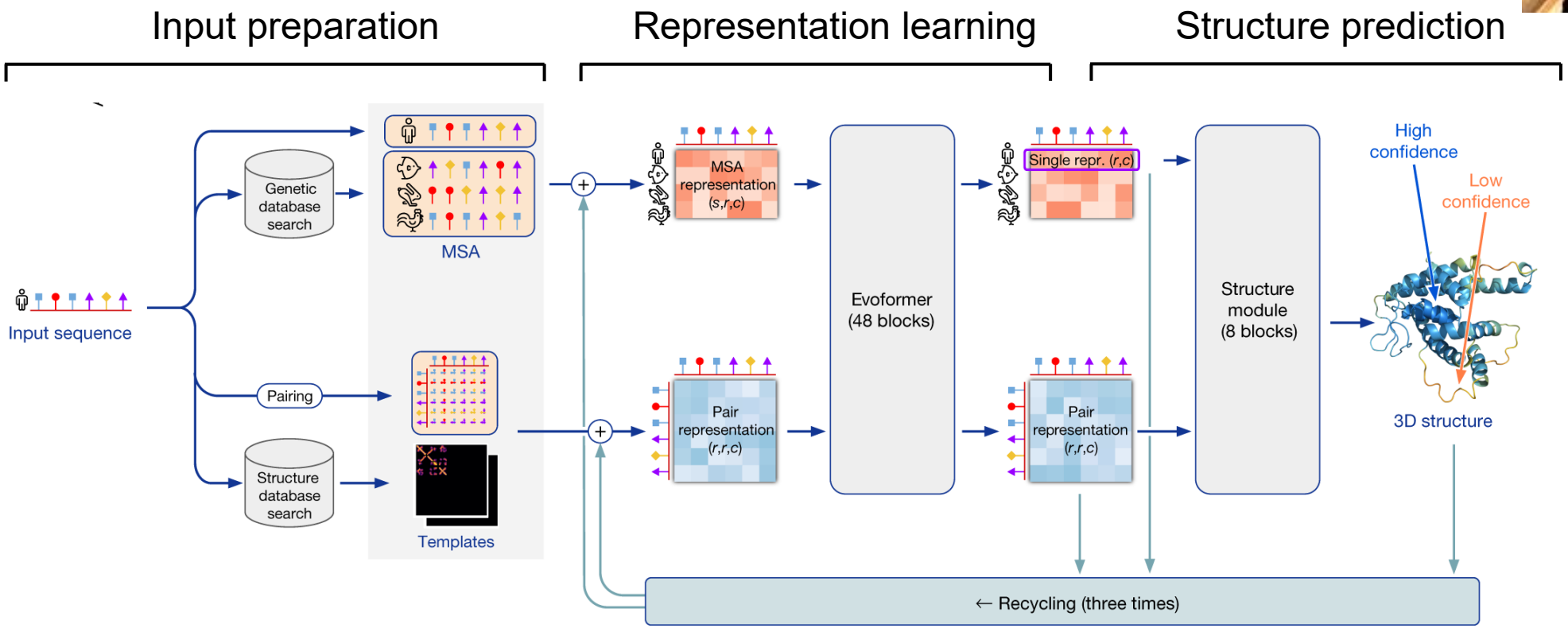
The game-changer: AlphaFold2





Overview of AlphaFold2: end-to-end



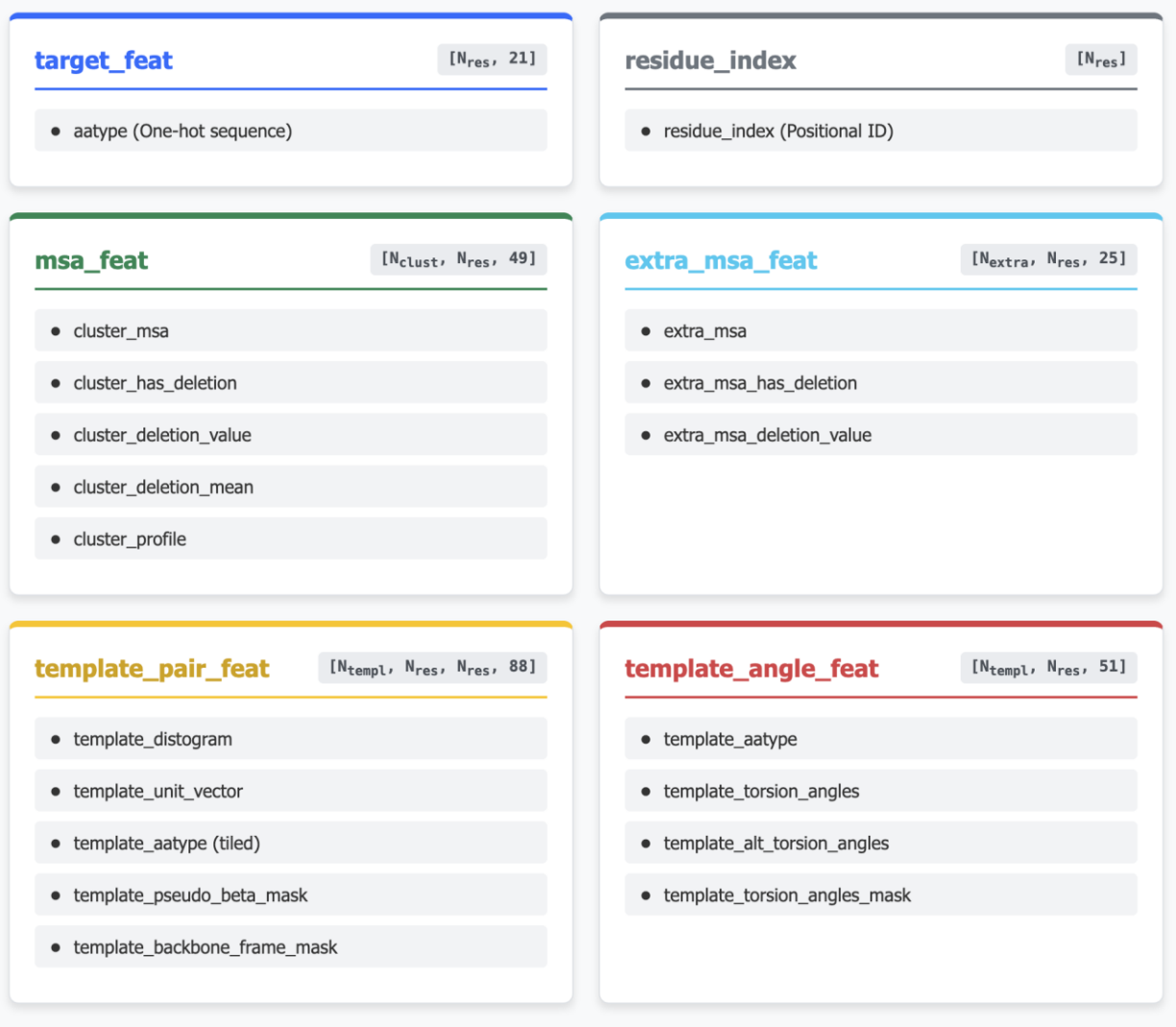


- End-to-end (structure module)
- From CNN to transformer/attention (evoformer)
  - Exchange information between MSA representation and pair representation
  - Triangle inequalities as natural constraints
- Other AI engineering (recycling)

Nature, 2021

Input preparation

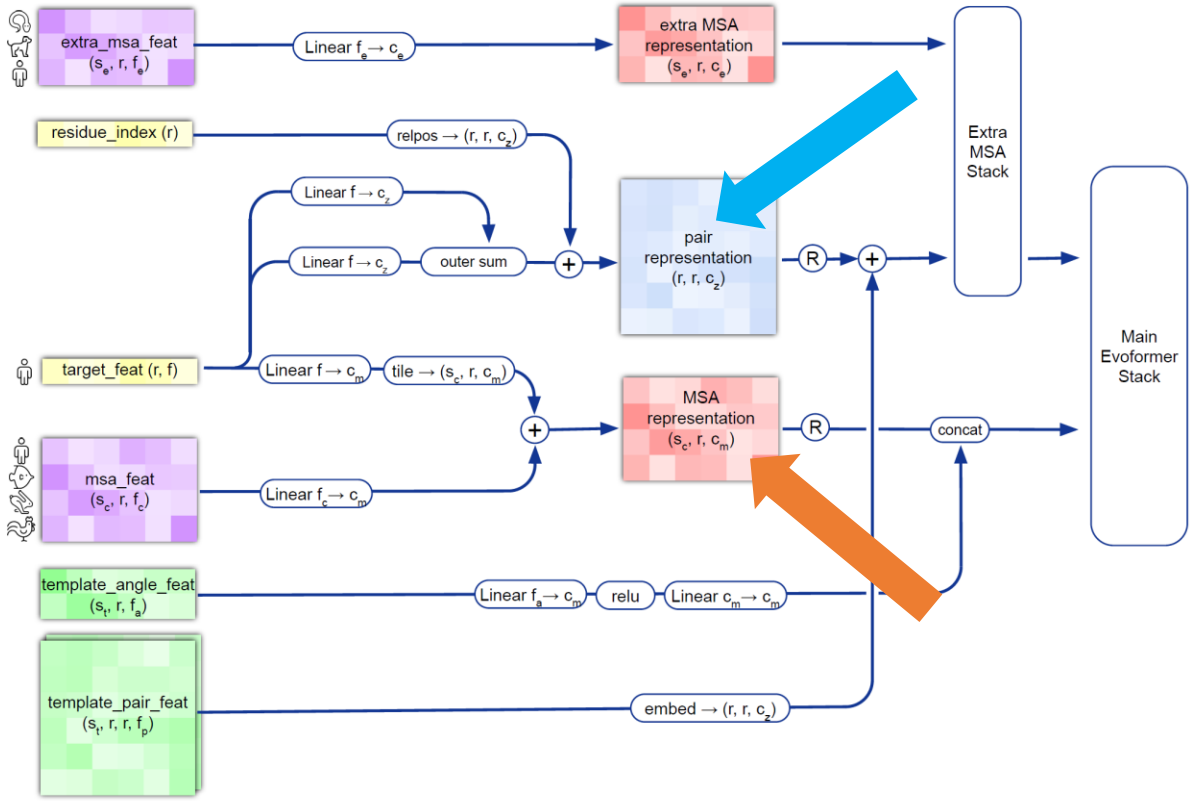
- **target_feat:** Represents the target amino acid sequence.
- **residue_index:** The positional index of each residue in the original sequence.
- **msa_feat:** Represents the Multiple Sequence Alignment (MSA) cluster centers and their statistics.
- **extra_msa_feat:** Represents additional MSA sequences that were not selected as cluster centers.
- **template_pair_feat:** Pairwise features derived from structural templates.
- **template_angle_feat:** Torsion angle features derived from structural templates.



Input embedding

This step is crucial, serving as the bridge that transforms raw biological features into high-dimensional representations for the network. AlphaFold 2 utilizes a dual-track architecture to simultaneously maintain and update two states:

- **MSA Representation (m_{si})**
- Contains evolutionary information from the Multiple Sequence Alignment (MSA).
- **Dimensions:** $s \in {1 \dots N_{seq}}$ (sequences), $i \in {1 \dots N_{res}}$ (residues), with channel size $c_m = 256$.
- **Pair Representation (${z_{ij}}$)**
- Contains relationship data between residue pairs (such as distances and contacts).
- **Dimensions:** $i, j \in {1 \dots N_{res}}$ (residue pairs), with channel size $c_z = 128$.



Input embedding

Algorithm 3 Embeddings for initial representations

def InputEmbedder($\{\mathbf{f}_i^{\text{target_feat}}\}, \{f_i^{\text{residue_index}}\}, \{\mathbf{f}_{s_c i}^{\text{msa_feat}}\}$) :

1: $\mathbf{a}_i, \mathbf{b}_i = \text{Linear}(\mathbf{f}_i^{\text{target_feat}})$ $\quad \mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^{c_z}, c_z = 128$

2: $\mathbf{z}_{ij} = \mathbf{a}_i + \mathbf{b}_j$ $\quad \mathbf{z}_{ij} \in \mathbb{R}^{c_z}, c_z = 128$

3: $\{\mathbf{z}_{ij}\}$ += relpos($\{f_i^{\text{residue_index}}\}$)

4: $\mathbf{m}_{s_c i} = \text{Linear}(\mathbf{f}_{s_c i}^{\text{msa_feat}}) + \text{Linear}(\mathbf{f}_i^{\text{target_feat}})$ $\quad \mathbf{m}_{s_c i} \in \mathbb{R}^{c_m}, c_m = 256$

5: **return** $\{\mathbf{m}_{s_c i}\}, \{\mathbf{z}_{ij}\}$

Input:

- **target_feat (f_i):** The raw features of the target protein sequence (e.g., residue type).
- **residue_index (f_i):** The position/index of each residue in the sequence.
- **msa_feat (f_sci):** The raw features from the Multiple Sequence Alignment (MSA).

Procedure:

- **Line 1: Linear Projection for Pair Initialization**

The algorithm takes the target sequence features (**f_i**) and projects them linearly into two separate vectors, **a_i** and **b_i**. These will represent the "rows" and "columns" of the pair matrix.

- **Line 2: Create Pair Representation (z_ij)**

The initial Pair representation (**z_ij**) is created by adding vector **a** at position *i* to vector **b** at position *j*. This essentially creates a grid where every residue is compared to every other residue.

Input embedding

Algorithm 3 Embeddings for initial representations

def InputEmbedder($\{\mathbf{f}_i^{\text{target_feat}}\}, \{f_i^{\text{residue_index}}\}, \{\mathbf{f}_{s_c i}^{\text{msa_feat}}\}$) :

1: $\mathbf{a}_i, \mathbf{b}_i = \text{Linear}(\mathbf{f}_i^{\text{target_feat}})$ $\quad \mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^{c_z}, c_z = 128$

2: $\mathbf{z}_{ij} = \mathbf{a}_i + \mathbf{b}_j$ $\quad \mathbf{z}_{ij} \in \mathbb{R}^{c_z}, c_z = 128$

3: $\{\mathbf{z}_{ij}\}$ += relpos($\{f_i^{\text{residue_index}}\}$)

4: $\mathbf{m}_{s_c i} = \text{Linear}(\mathbf{f}_{s_c i}^{\text{msa_feat}}) + \text{Linear}(\mathbf{f}_i^{\text{target_feat}})$ $\quad \mathbf{m}_{s_c i} \in \mathbb{R}^{c_m}, c_m = 256$

5: **return** $\{\mathbf{m}_{s_c i}\}, \{\mathbf{z}_{ij}\}$

Procedure:

- **Line 3: Add Relative Positional Encoding**

The algorithm calculates the relative distance between residues using the **residue_index**. This relative position encoding (relpos) is added to the Pair representation (z_ij). This tells the network how far apart residues are in the primary sequence.

- **Line 4: Create MSA Representation (m_sci)**

The MSA representation (**m_sci**) is initialized. It takes the raw MSA features (**f_sci**) and projects them linearly. It also takes the target sequence features (**f_i**), projects them, and adds them to the MSA features. This ensures the MSA representation contains information about both the evolutionary history and the specific target sequence.

- **Line 5: Return**

The algorithm outputs the initialized MSA representation (**m_sci**) and the initialized Pair representation (**z_ij**).

Input embedding

Algorithm 3 Embeddings for initial representations

def InputEmbedder($\{\mathbf{f}_i^{\text{target_feat}}\}, \{f_i^{\text{residue_index}}\}, \{\mathbf{f}_{s_c i}^{\text{msa_feat}}\}$) :

1: $\mathbf{a}_i, \mathbf{b}_i = \text{Linear}(\mathbf{f}_i^{\text{target_feat}})$ $\quad \mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^{c_z}, c_z = 128$

2: $\mathbf{z}_{ij} = \mathbf{a}_i + \mathbf{b}_j$ $\quad \mathbf{z}_{ij} \in \mathbb{R}^{c_z}, c_z = 128$

3: $\{\mathbf{z}_{ij}\}$ += relpos($\{f_i^{\text{residue_index}}\}$)

4: $\mathbf{m}_{s_c i} = \text{Linear}(\mathbf{f}_{s_c i}^{\text{msa_feat}}) + \text{Linear}(\mathbf{f}_i^{\text{target_feat}})$ $\quad \mathbf{m}_{s_c i} \in \mathbb{R}^{c_m}, c_m = 256$

5: **return** $\{\mathbf{m}_{s_c i}\}, \{\mathbf{z}_{ij}\}$

Output:

- **{m_sci}**: The initialized MSA representation (ready for the Evoformer).
- **{z_ij}**: The initialized Pair representation (ready for the Evoformer).

Input embedding

- **Concept:** It replaces absolute position (e.g., "residue #100") with relative distance
- **Mechanism:**
  - **Calculate Difference:** Compute $i - j$.
  - **Clipping:** Distances are capped at ± 32.
  - **Encoding:** This clipped distance is one-hot encoded and projected into a 128-dimension vector.
- **Benefits:**
  - **Invariance:** The features remain the same regardless of its position.
  - **Generalization:** It handles long sequences without quality loss.
  - **Local Focus:** It forces the model to prioritize immediate sequence neighbors first, leaving long-range 3D interactions for later layers.

Algorithm 4 Relative position encoding

def $\text{relpos}(\{f_i^{\text{residue_index}}\}, \mathbf{v}_{\text{bins}} = [-32, -31, \ldots, 32])$:

1: $d_{ij} = f_i^{\text{residue_index}} - f_j^{\text{residue_index}}$ $\quad d_{ij} \in \mathbb{Z}$

2: $\mathbf{p}_{ij} = \text{Linear}(\text{one_hot}(d_{ij}, \mathbf{v}_{\text{bins}}))$ $\quad \mathbf{p}_{ij} \in \mathbb{R}^{c_z}$

3: **return** $\{\mathbf{p}_{ij}\}$

Algorithm 5 One-hot encoding with nearest bin

def $\text{one_hot}(x, \mathbf{v}_{\text{bins}})$: $\quad x \in \mathbb{R}, \quad \mathbf{v}_{\text{bins}} \in \mathbb{R}^{N_{\text{bins}}}$

1: $\mathbf{p} = \mathbf{0}$ $\quad \mathbf{p} \in \mathbb{R}^{N_{\text{bins}}}$

2: $b = \arg\min(|x - \mathbf{v}_{\text{bins}}|)$

3: $p_b = 1$

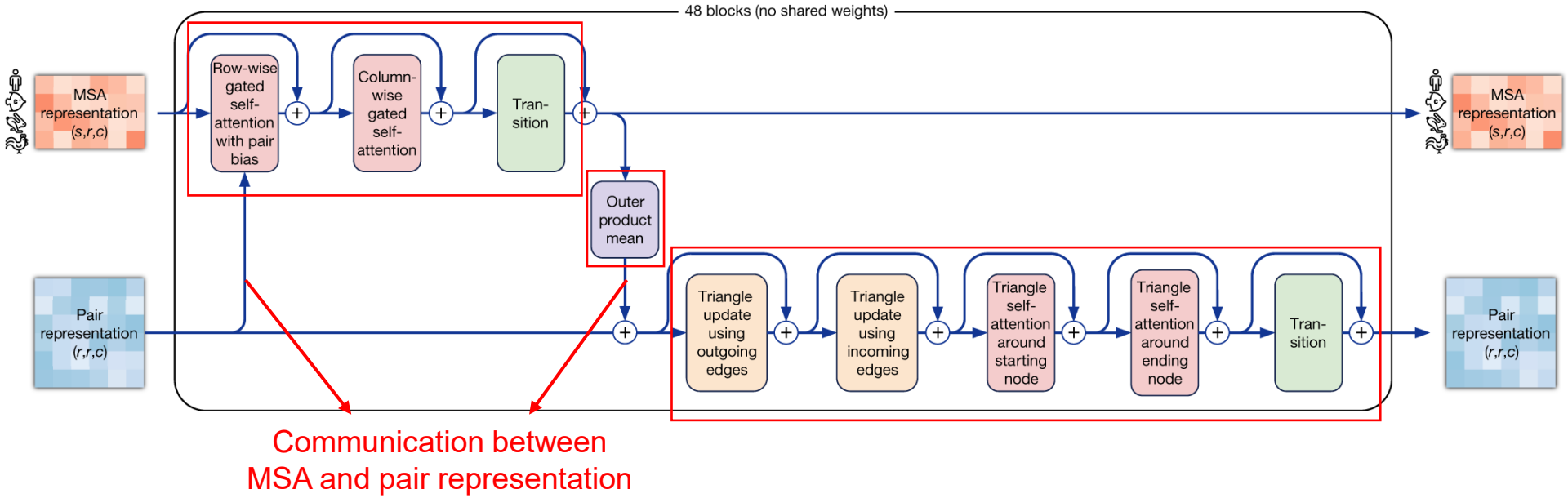
4: **return** $\mathbf{p}$

	j=1	j=2	j=3	j=4	j=5
i=1	0	-1	-2	-3	-4
i=2	1	0	-1	-2	-3
i=3	2	1	0	-1	-2
i=4	3	2	1	0	-1
i=5	4	3	2	1	0

Evoformer module

Evoformer consists of 3 major steps:

1. Exchange information between MSA (evolution) and pair (structure) representations: Updates MSA representations with axial attention and to inform pair representations. Updates pair representations using outer product mean block, and to inform MSA representations.
2. Applies triangle inequalities to the updated pair representations to enforce consistency.



Evoformer module consists of 48 identical blocks that take MSA (evolution) and pair (structure) representations as input and produce their refined versions as output.

Evoformer module

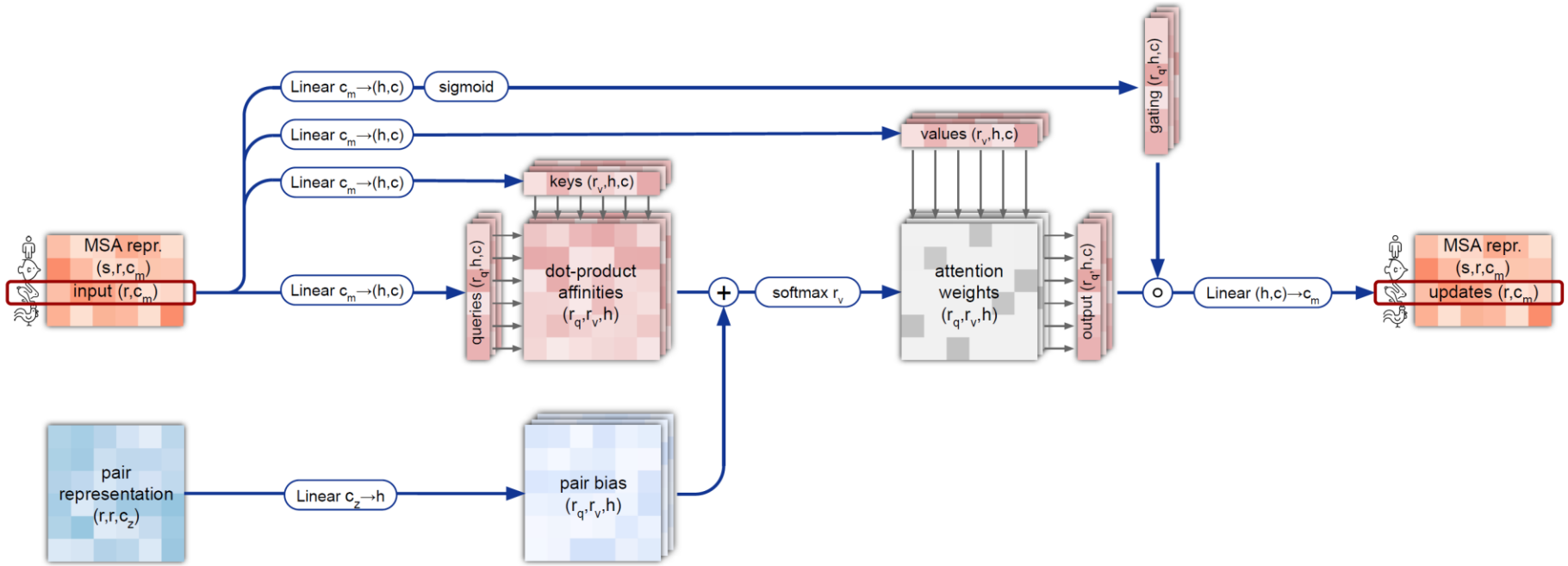
•**Concept:** The core engine of the network. It uses a "two-tower" architecture to process evolutionary data (MSA) and spatial data (Pair).
•**Mechanism:**
1.MSA Stack: Updates sequence information using **Row Attention** (which incorporates Pair information) and **Column Attention**.
2.Communication: Information flows from the MSA stack to the Pair stack via an **Outer Product Mean**.
3.Pair Stack: Updates spatial relationships using **Triangle Multiplicative** updates and **Triangle Attention** to strictly enforce 3D geometric constraints.
•**Outcome:** It outputs highly processed MSA and Pair representations, plus a "single" representation (derived from the first row of the MSA) to be used by the structure module.

Algorithm 6 Evoformer stack

def EvoformerStack($\{\mathbf{m}_{si}\}, \{\mathbf{z}_{ij}\}, N_{\text{block}} = 48, c_s = 384$) :

for all $l \in [1, \ldots, N_{\text{block}}]$ **do**
 # MSA stack
$\{\mathbf{m}_{si}\}$ += $\text{DropoutRowwise}_{0.15}(\text{MSARowAttentionWithPairBias}(\{\mathbf{m}_{si}\}, \{\mathbf{z}_{ij}\}))$
$\{\mathbf{m}_{si}\}$ += $\text{MSAColumnAttention}(\{\mathbf{m}_{si}\})$
$\{\mathbf{m}_{si}\}$ += $\text{MSATransition}(\{\mathbf{m}_{si}\})$
 # Communication
$\{\mathbf{z}_{ij}\}$ += $\text{OuterProductMean}(\{\mathbf{m}_{si}\})$
 # Pair stack
$\{\mathbf{z}_{ij}\}$ += $\text{DropoutRowwise}_{0.25}(\text{TriangleMultiplicationOutgoing}(\{\mathbf{z}_{ij}\}))$
$\{\mathbf{z}_{ij}\}$ += $\text{DropoutRowwise}_{0.25}(\text{TriangleMultiplicationIncoming}(\{\mathbf{z}_{ij}\}))$
$\{\mathbf{z}_{ij}\}$ += $\text{DropoutRowwise}_{0.25}(\text{TriangleAttentionStartingNode}(\{\mathbf{z}_{ij}\}))$
$\{\mathbf{z}_{ij}\}$ += $\text{DropoutColumnwise}_{0.25}(\text{TriangleAttentionEndingNode}(\{\mathbf{z}_{ij}\}))$
$\{\mathbf{z}_{ij}\}$ += $\text{PairTransition}(\{\mathbf{z}_{ij}\})$
end for
 # Extract the single representation
$\mathbf{s}_i = \text{Linear}(\mathbf{m}_{1i})$ $\mathbf{s}_i \in \mathbb{R}^{c_s}$
return $\{\mathbf{m}_{si}\}, \{\mathbf{z}_{ij}\}, \{\mathbf{s}_i\}$

Row-wise self-attention



Supplementary Figure 2 | MSA row-wise gated self-attention with pair bias. Dimensions: s: sequences, r: residues, c: channels, h: heads.

- **Row-wise attention:** exchange information within sequence
- **"With pair bias":** if pair representation suggest that two amino acid residues interact in 3D, row-wise self-attention also makes use of this information to reflect this in the updated MSA embedding.
- **Gating mechanism**: control how much information will be kepted

Row-wise self-attention

Algorithm 7 MSA row-wise gated self-attention with pair bias

def $\text{MSARowAttentionWithPairBias}(\{\mathbf{m}_{si}\}, \{\mathbf{z}_{ij}\}, c = 32, N_{\text{head}} = 8)$:

 # Input projections

$\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

$\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h = \text{LinearNoBias}(\mathbf{m}_{si})$ $\qquad \mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h \in \mathbb{R}^c,\ h \in \{1, \ldots, N_{\text{head}}\}$

$b_{ij}^h = \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$

$\mathbf{g}_{si}^h = \text{sigmoid}\left(\text{Linear}(\mathbf{m}_{si})\right)$ $\qquad \mathbf{g}_{si}^h \in \mathbb{R}^c$

 # Attention

$a_{sij}^h = \text{softmax}_j\left(\frac{1}{\sqrt{c}}\, \mathbf{q}_{si}^{h\top} \mathbf{k}_{sj}^h + b_{ij}^h\right)$

$\mathbf{o}_{si}^h = \mathbf{g}_{si}^h \odot \sum_j a_{sij}^h \mathbf{v}_{sj}^h$

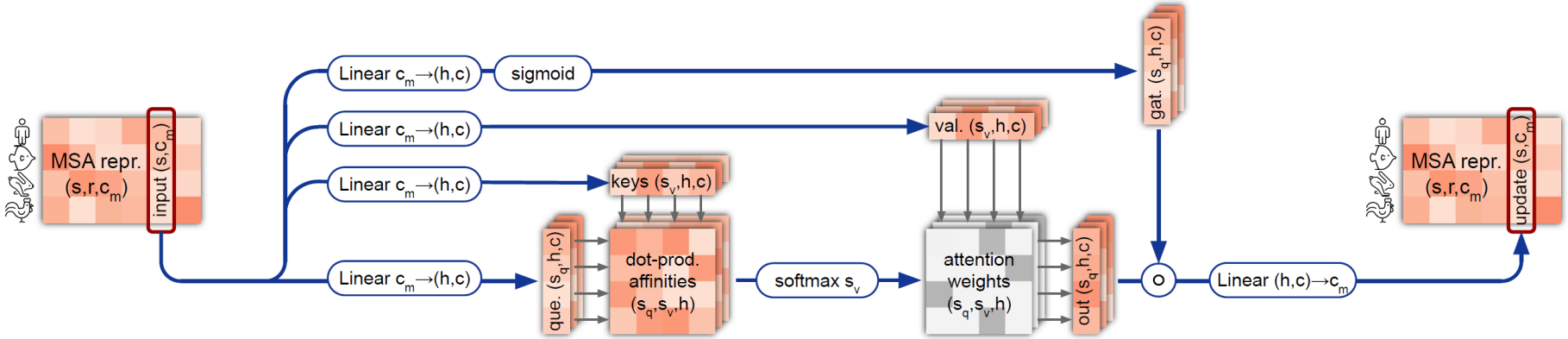
 # Output projection

$\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}_{si}^h)\right)$ $\qquad \tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

return $\{\tilde{\mathbf{m}}_{si}\}$

This algorithm updates the **MSA Representation (m_{si})** by performing self-attention along the horizontal axis (the sequence dimension). For every individual sequence s in the MSA, it calculates how residue i relates to every other residue j within that same sequence, updating the features of i based on these relationships.

Column-wise self-attention



Supplementary Figure 3 | MSA column-wise gated self-attention. Dimensions: s: sequences, r: residues, c: channels, h: heads.

- **Column-wise attention:** exchange information between sequences within an alignment column

Column-wise self-attention

Algorithm 8 MSA column-wise gated self-attention

def $\text{MSAColumnAttention}(\{\mathbf{m}_{si}\}, c = 32, N_{\text{head}} = 8)$:

 # Input projections

$\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

$\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h = \text{LinearNoBias}(\mathbf{m}_{si})$ $\quad \mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h \in \mathbb{R}^c,\ h \in \{1, \ldots, N_{\text{head}}\}$

$\mathbf{g}_{si}^h = \text{sigmoid}\left(\text{Linear}(\mathbf{m}_{si})\right)$ $\quad \mathbf{g}_{si}^h \in \mathbb{R}^c$

 # Attention

$a_{sti}^h = \text{softmax}_t\left(\frac{1}{\sqrt{c}}\, \mathbf{q}_{si}^{h\top} \mathbf{k}_{ti}^h\right)$

$\mathbf{o}_{si}^h = \mathbf{g}_{si}^h \odot \sum_t a_{sti}^h \mathbf{v}_{st}^h$

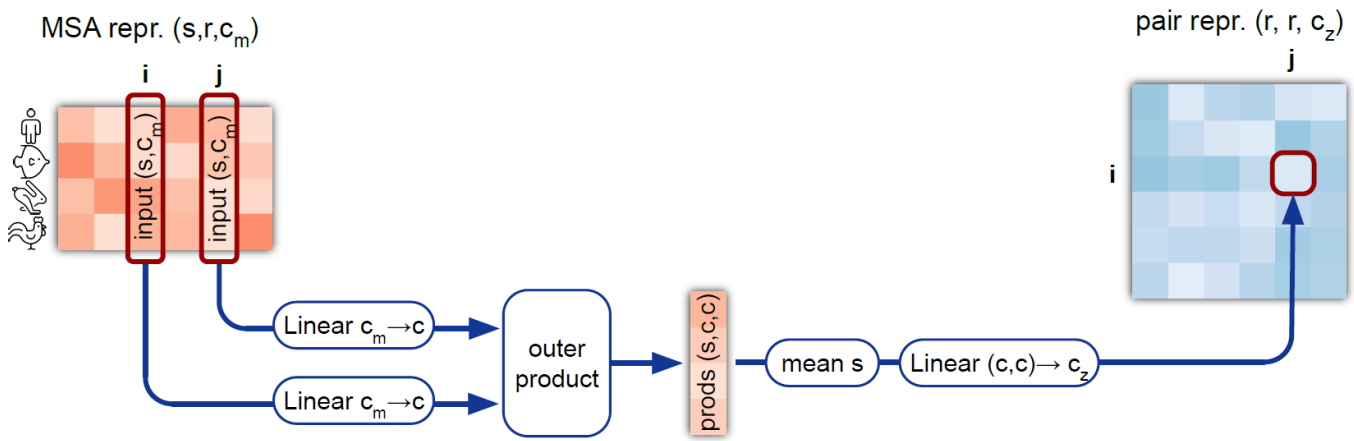
 # Output projection

$\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}_{si}^h)\right)$ $\quad \tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

return $\{\tilde{\mathbf{m}}_{si}\}$

- **Computing attention scores** between the rows of the MSA representation to determines which sequences are most relevant to each other, effectively filtering for evolutionary "clues".
- However, it does not directly calculate the “relevance”, which is done later
- It does not need pair-bias.

Outer product mean



Supplementary Figure 5 | Outer product mean. Dimensions: s: sequences, r: residues, c: channels.

- Mathematically, this block calculates an **empirical covariance matrix** between residue i and residue j across the evolutionary history.
- **Per-Sequence Interaction:** For every sequence s, the **Outer Product** creates a matrix that records every possible combination of features between position i and position j.
- The goal is to bridge the gap between **1D sequence data** and **2D spatial maps**.

Outer product mean

Algorithm 10 Outer product mean

def $\text{OuterProductMean}(\{\mathbf{m}_{si}\}, c = 32)$:

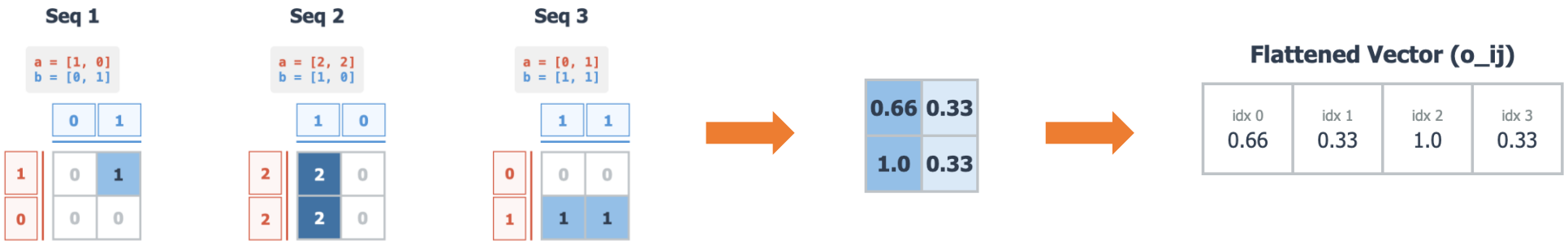
1: $\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

2: $\mathbf{a}_{si}, \mathbf{b}_{si} = \text{Linear}(\mathbf{m}_{si})$ $\quad \mathbf{a}_{si}, \mathbf{b}_{si} \in \mathbb{R}^{c}$

3: $\mathbf{o}_{ij} = \text{flatten}\left(\text{mean}_s(\mathbf{a}_{si} \otimes \mathbf{b}_{sj})\right)$ $\quad \mathbf{o}_{ij} \in \mathbb{R}^{c \cdot c}$

4: $\mathbf{z}_{ij} = \text{Linear}(\mathbf{o}_{ij})$ $\quad \mathbf{z}_{ij} \in \mathbb{R}^{c_z}$

5: **return** $\{\mathbf{z}_{ij}\}$

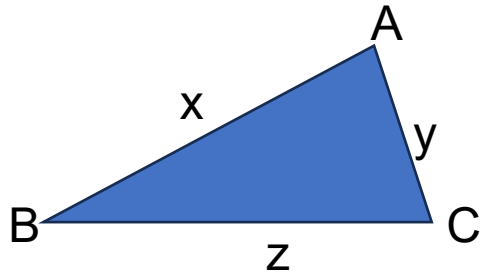


INDIVIDUAL OUTER PRODUCTS MEAN OVER SEQUENCES FLATTEN

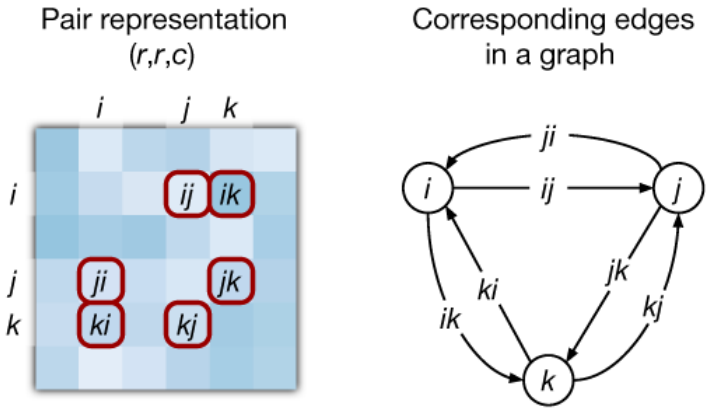
Triangle inequalities as natural constraints for pair distances

- Triangle inequality theorem: in △ABC, if AB = x, BC = z, CA = y, then

$$x + y > z,\ x + z > y,\ y + z > x$$

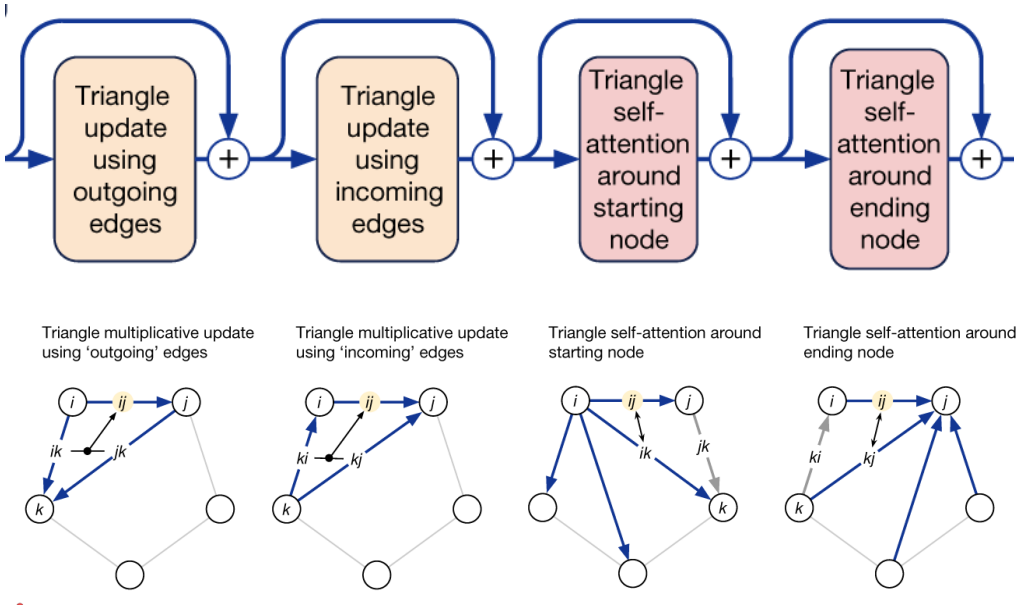


The pair representation interpreted as directed edges in a graph

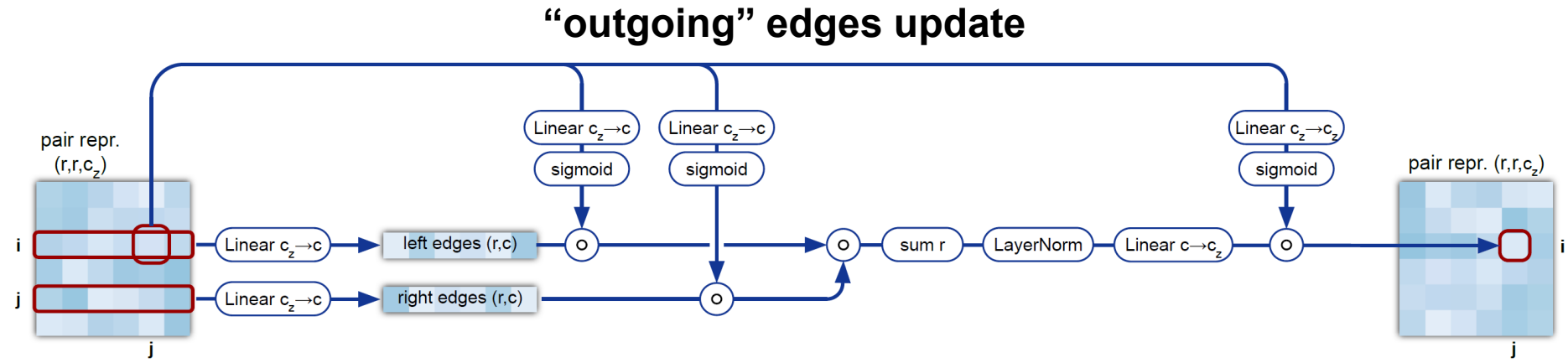


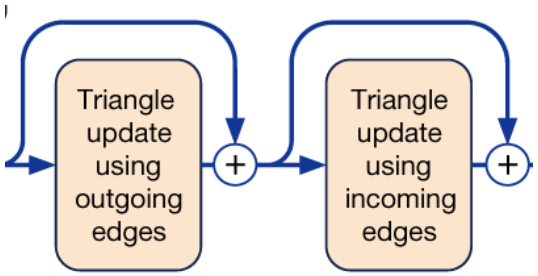
Circle: residue
Directed edges: entries in the pair representation

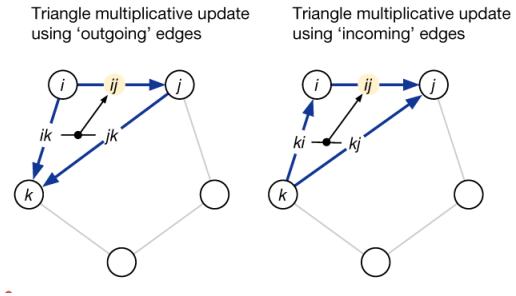
Triangle multiplicative update and triangle self-attention



Triangular multiplicative update







Algorithm 11 Triangular multiplicative update using "outgoing" edges

def TriangleMultiplicationOutgoing($\{\mathbf{z}_{ij}\}, c = 128$) :

1: $\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$
2: $\mathbf{a}_{ij}, \mathbf{b}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right) \odot \text{Linear}(\mathbf{z}_{ij})$ $\quad \mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^{c}$
3: $\mathbf{g}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right)$ $\quad \mathbf{g}_{ij} \in \mathbb{R}^{c_z}$
4: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \text{Linear}(\text{LayerNorm}(\sum_k \mathbf{a}_{ik} \odot \mathbf{b}_{jk}))$ $\quad \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{c_z}$
5: **return** $\{\tilde{\mathbf{z}}_{ij}\}$

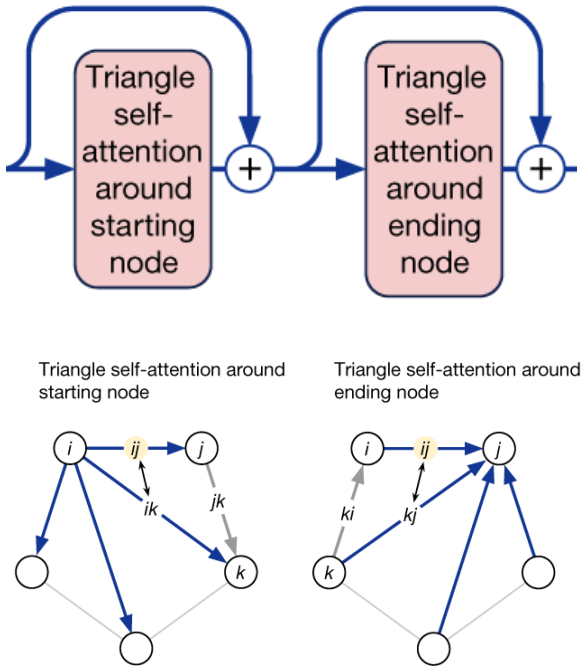
Algorithm 12 Triangular multiplicative update using "incoming" edges

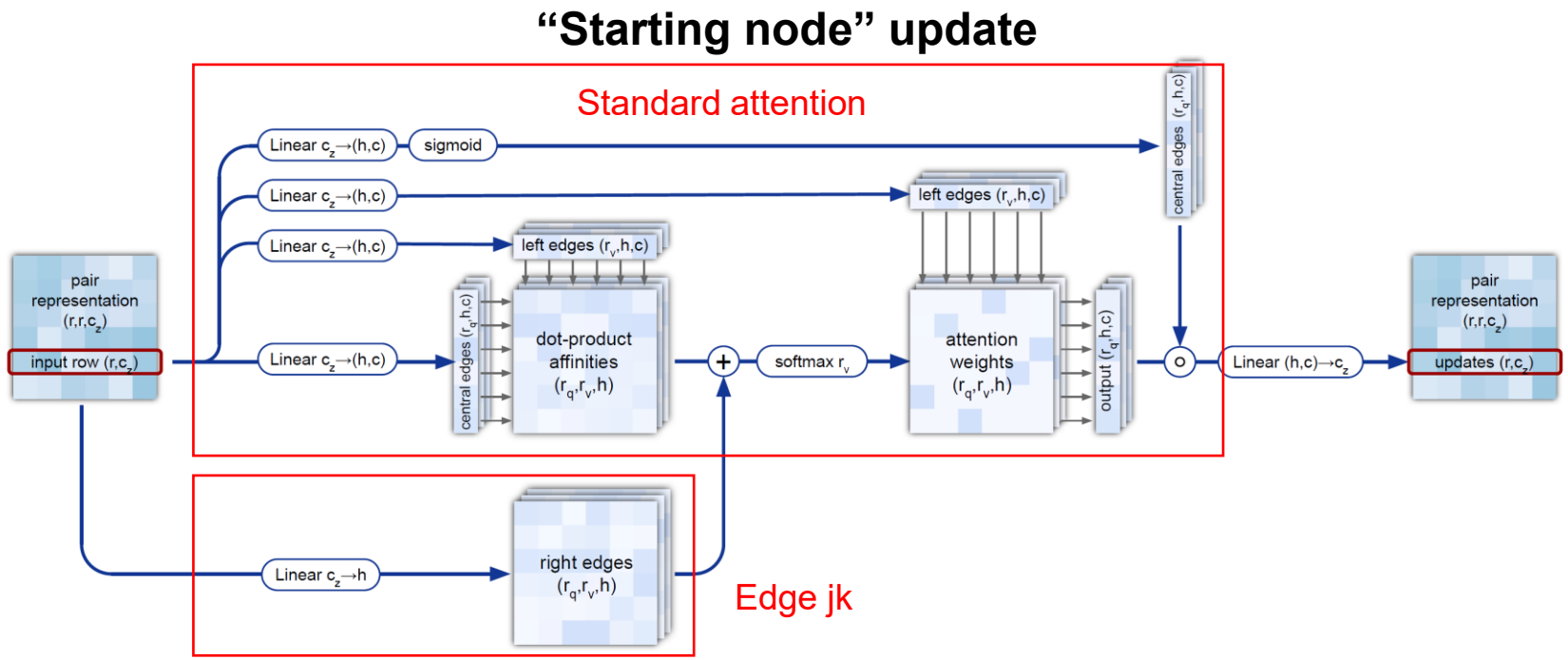
def TriangleMultiplicationIncoming($\{\mathbf{z}_{ij}\}, c = 128$) :

1: $\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$
2: $\mathbf{a}_{ij}, \mathbf{b}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right) \odot \text{Linear}(\mathbf{z}_{ij})$ $\quad \mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^{c}$
3: $\mathbf{g}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right)$ $\quad \mathbf{g}_{ij} \in \mathbb{R}^{c_z}$
4: $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \text{Linear}(\text{LayerNorm}(\sum_k \mathbf{a}_{ki} \odot \mathbf{b}_{kj}))$ $\quad \tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{c_z}$
5: **return** $\{\tilde{\mathbf{z}}_{ij}\}$

Triangular self-attention

- The **“starting node”** updates the edge ij with values from all edges that share the same starting node i.
- The **“ending node”** updates the edge ij with values from all edges that share the same ending node j.

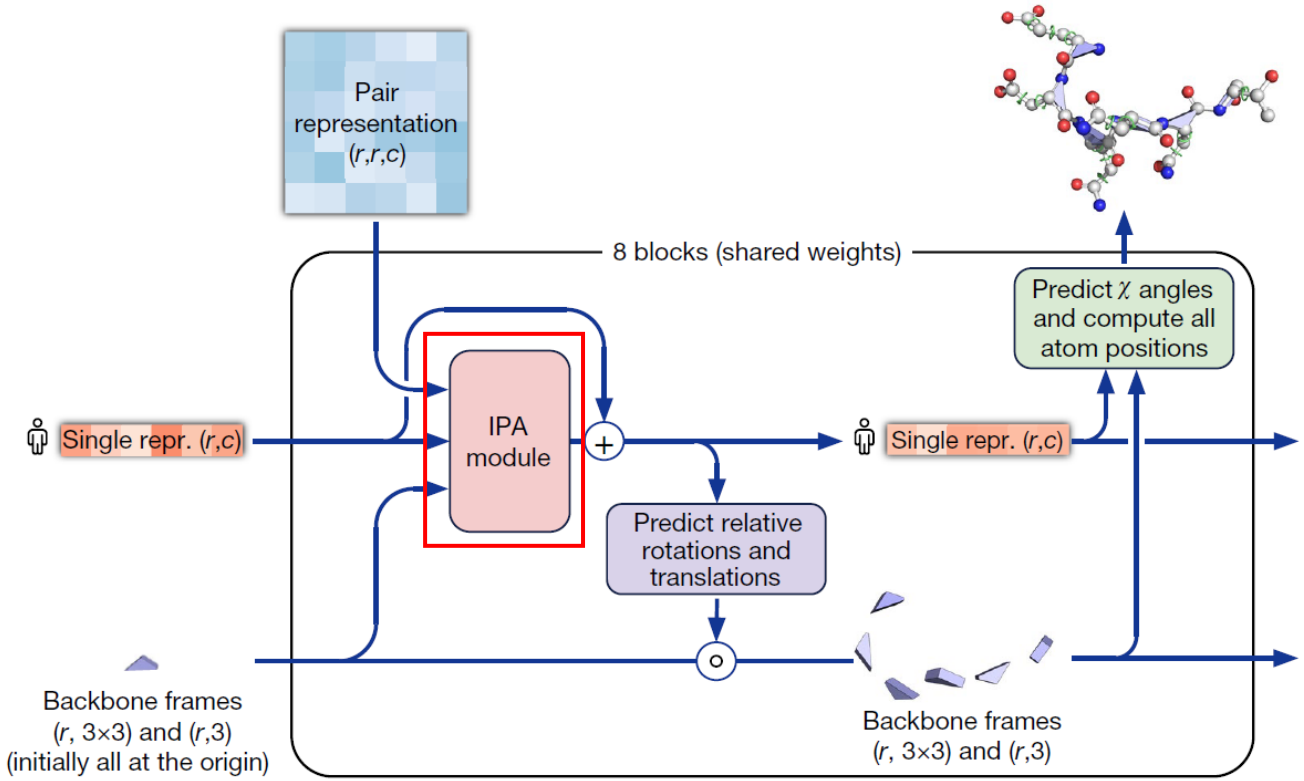




The decision whether edge ij will receive an update from edge ik is not only determined by their query-key similarity, but also modulated by the information b_{jk} derived from the third edge jk of this triangle.

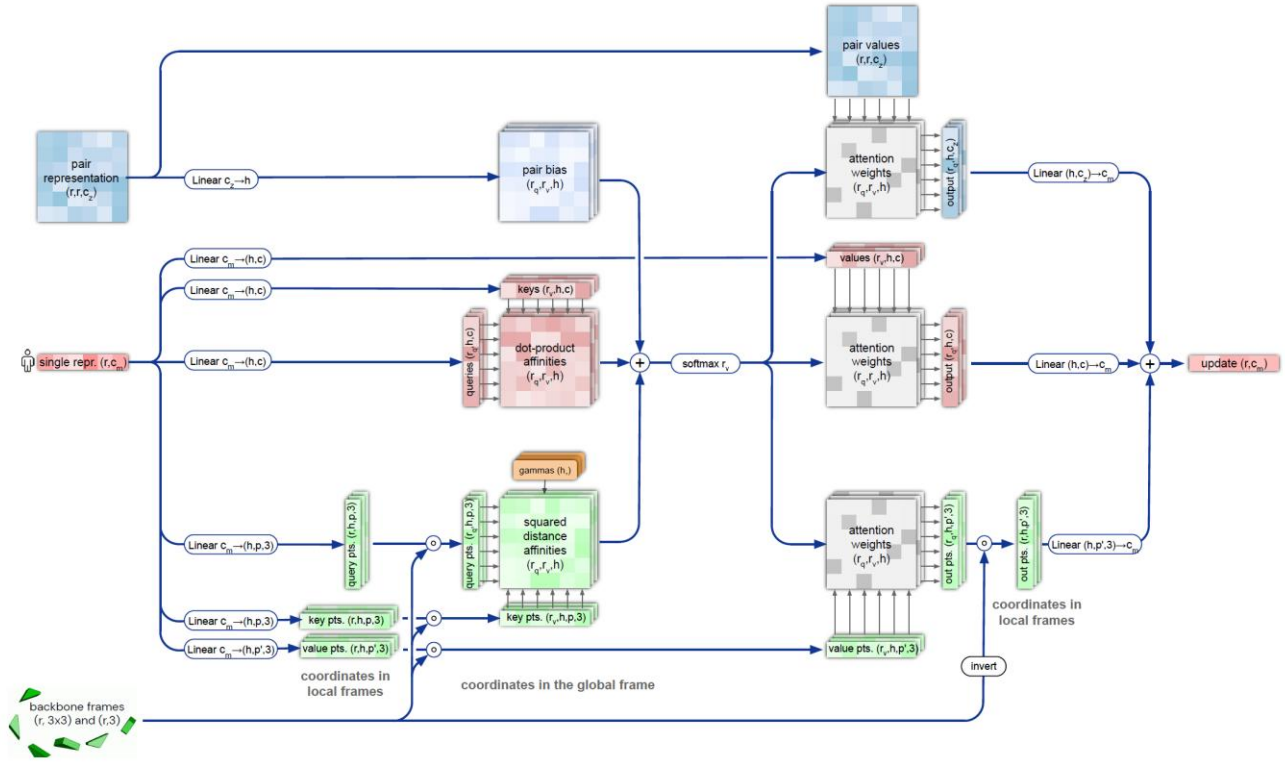
Structure module

- **IPA module**: Invariant Point Attention (IPA) is a form of attention that acts on a set of frames and is **invariant under global Euclidean transformations T_{global}** on said frames.
- The IPA module aggregating together the information from sequence embedding (extracted from MSA embedding), pair representation and backbone frames, and updates the embedding of a sequence.



Invariant Point Attention Module

- The IPA augments each of the usual attention queries, keys and values with 3D points that are produced **in the local frame of each residue** such that the final value is invariant to global rotations and translations

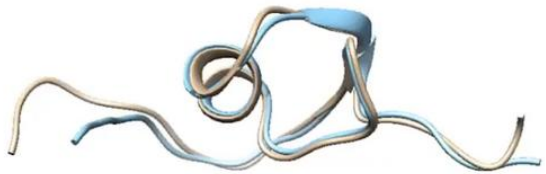


Supplementary Figure 8 | Invariant Point Attention Module. (**top, blue arrays**) modulation by the pair representation. (**middle, red arrays**) standard attention on abstract features. (**bottom, green arrays**) Invariant point attention. Dimensions: r: residues, c: channels, h: heads, p: points.

Structure evaluation metrics

Global accuracy

First align predicted structure with reference



RMSD↓

$$d = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2}$$

TM-score↑

$$\text{TM score} = \max\left[\frac{1}{L_{target}}\sum_{i}^{L_{common}} \frac{1}{1+\left(\frac{d_i}{d_0(L_{target})}\right)^2}\right]$$

L_{target}是目标蛋白的长度， L_{common}是模板蛋白和目标蛋白公共残基数量， d_i是第i个残基在两种结构中的距离。其中：

$$d_i(L_{target}) = 1.24\sqrt[3]{L_{target} - 15} - 1.8$$

是归一化因子。TM-score的分数范围为0-1，随机两蛋白的TM-score一般小于0.2，基本相同的两结构TM-score一般大于0.5

Local accuracy

LDDT↑

度量在真实结构中空间距离<15 Å且残基序号差异≥r的哪些残基对在真实结构和预测结构中距离差异小于t所占的比例，t的取值范围为0.5, 1, 2, 4 Å。公式如下：

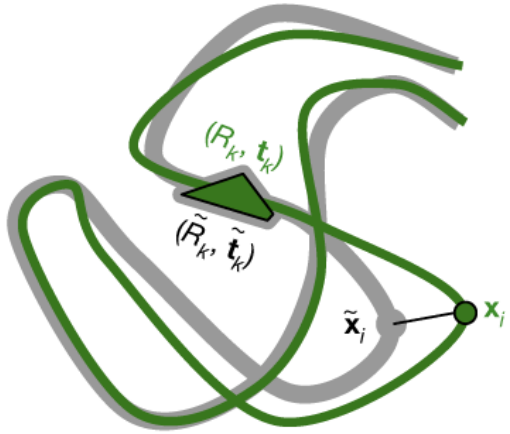
$$\text{LDDT} = \frac{100}{4L}\sum_{t\in\{0.5,1,2,4\}}\sum_{i=1}^{L}\frac{\sum_{j,|i-j|\geq r, D_{ij}<15} 1(|D_{ij}-d_{ij}|<t)}{\sum_{j,|i-j|\geq r, D_{ij}<15} 1}$$

plddt in AlphaFold2: confidence evaluation

Loss function used in AlphaFold2

$$\mathcal{L} = \begin{cases} 0.5\mathcal{L}_{\text{FAPE}} + 0.5\mathcal{L}_{\text{aux}} + 0.3\mathcal{L}_{\text{dist}} + 2.0\mathcal{L}_{\text{msa}} + 0.01\mathcal{L}_{\text{conf}} & \text{training} \\ 0.5\mathcal{L}_{\text{FAPE}} + 0.5\mathcal{L}_{\text{aux}} + 0.3\mathcal{L}_{\text{dist}} + 2.0\mathcal{L}_{\text{msa}} + 0.01\mathcal{L}_{\text{conf}} + 0.01\mathcal{L}_{\text{exp resolved}} + 1.0\mathcal{L}_{\text{viol}} & \text{fine-tuning} \end{cases}$$

Frame aligned point error (FAPE) calculation



Green, predicted structure; grey, true structure; $(R_k, \mathbf{t}_k)$, frames; $\mathbf{x}_i$, atom positions.

Algorithm 28 Compute the Frame aligned point error

def computeFAPE$(\{T_i\}, \{\vec{\mathbf{x}}_j\}, \{T_i^{\text{true}}\}, \{\vec{\mathbf{x}}_j^{\text{true}}\}, Z = 10\text{Å}, d_{\text{clamp}} = 10\text{Å}, \epsilon = 10^{-4}\text{Å}^2)$:

$T_i, T_i^{\text{true}} \in (\mathbb{R}^{3\times3}, \mathbb{R}^3)$

$\vec{\mathbf{x}}_j, \vec{\mathbf{x}}_j^{\text{true}} \in \mathbb{R}^3,$

$i \in \{1, ..., N_{\text{frames}}\}, j \in \{1, ..., N_{\text{atoms}}\}$

1: $\vec{\mathbf{x}}_{ij} = T_i^{-1} \circ \vec{\mathbf{x}}_j$ Predicted atom position $\overrightarrow{x_j}$ relative to frame Ti $\quad \vec{\mathbf{x}}_{ij} \in \mathbb{R}^3$

2: $\vec{\mathbf{x}}_{ij}^{\text{true}} = T_i^{\text{true}-1} \circ \vec{\mathbf{x}}_j^{\text{true}}$ True atom position $\overrightarrow{x_j}$ relative to frame Ti $\quad \vec{\mathbf{x}}_{ij}^{\text{true}} \in \mathbb{R}^3$

3: $d_{ij} = \sqrt{\|\vec{\mathbf{x}}_{ij} - \vec{\mathbf{x}}_{ij}^{\text{true}}\|^2 + \epsilon}$ $\quad d_{ij} \in \mathbb{R}$

4: $\mathcal{L}_{\text{FAPE}} = \frac{1}{Z}\,\text{mean}_{i,j}(\text{minimum}(d_{\text{clamp}}, d_{ij}))$

5: **return** $\mathcal{L}_{\text{FAPE}}$

Other details in AlphaFold2: MSA block deletion

Why we need it:

- **Generalization:** Forces the model to learn folding principles rather than relying on dense statistical correlations.
- **Robustness:** Simulates "shallow" MSAs, training the model to work with scarce evolutionary data.
- **Regularization:** Prevents overfitting to specific alignment biases.

Solutions:

- **Method:** Deletes large, contiguous blocks (30% size, 5 times) from the MSA during training.
- **Targeting:** Because MSAs are sorted by similarity, this removes entire evolutionary clusters/sub-families.
- **Dynamic:** The model sees different subsets of data every time.

Algorithm 1 MSA block deletion

def MSABlockDeletion(**msa**) :

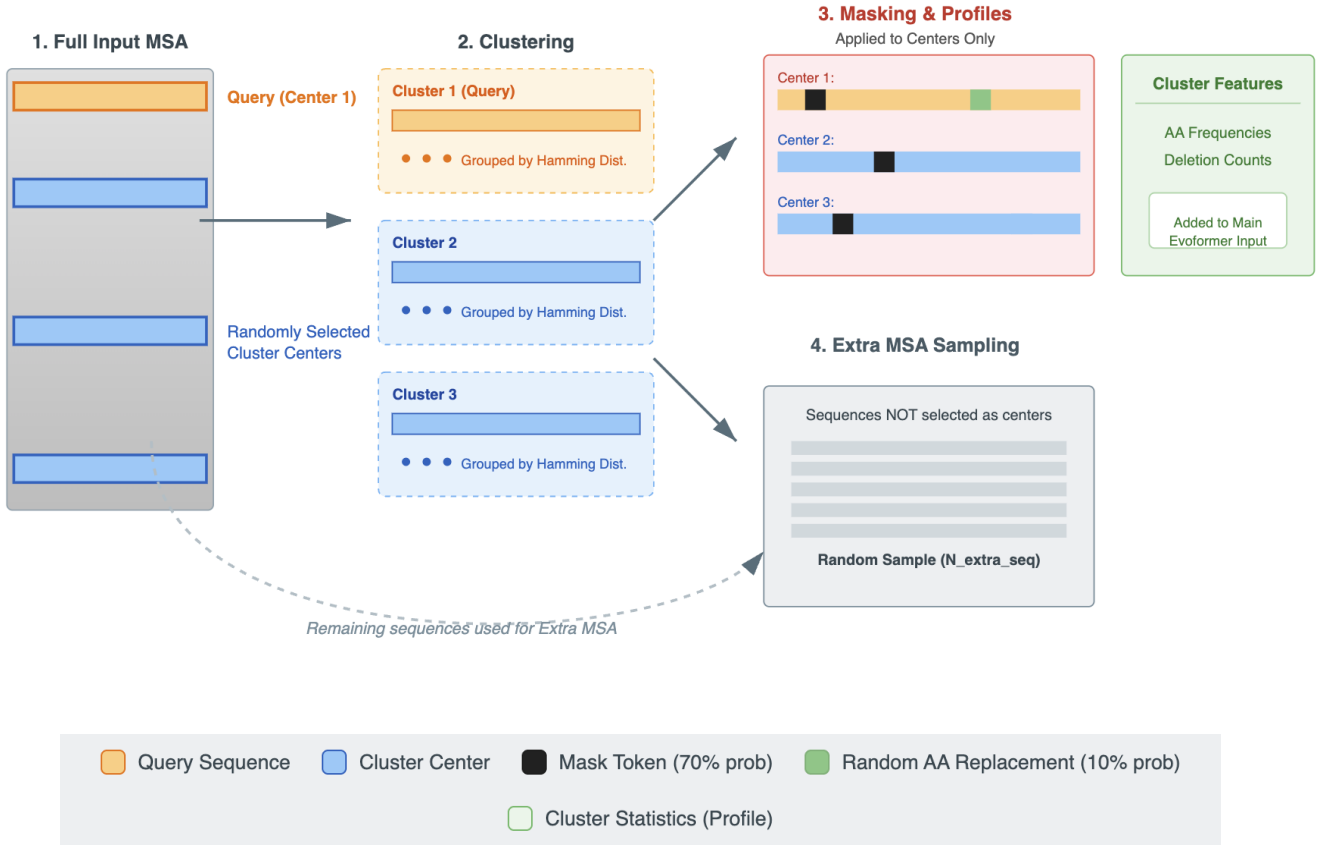
block_size $= \lfloor 0.3 \cdot N_{\text{all_seq}} \rfloor$
to_delete $= \{\}$
for all $j \in [1, \ldots, 5]$ **do**
    block_start $\leftarrow$ uniform$(1, N_{\text{all_seq}})$
    to_delete $\leftarrow$ to_delete $\cup$ [block_start, ..., block_start + block_size -1]
end for
keep $\leftarrow [1, \ldots, N_{\text{all_seq}}] \setminus$ to_delete
msa $\leftarrow$ **msa**$_{\text{keep}}$
return **msa**

Other details in AlphaFold2: MSA clustering

Why we need it: Processing all MSA sequences is too computationally expensive (N_{seq}^2).

Solution:

- **Clustering:** Sequences are grouped by similarity. Random "cluster centers" are chosen to represent the groups.
- **Aggregation:** Data from the unchosen sequences isn't lost; it is averaged into statistical "mini-profiles" attached to the cluster centers.
- **Masking:** Cluster centers are randomly masked during training to force the model to learn evolutionary patterns.
- **Extra Sequences:** A random sample of raw sequences is added separately to ensure diversity is maintained.



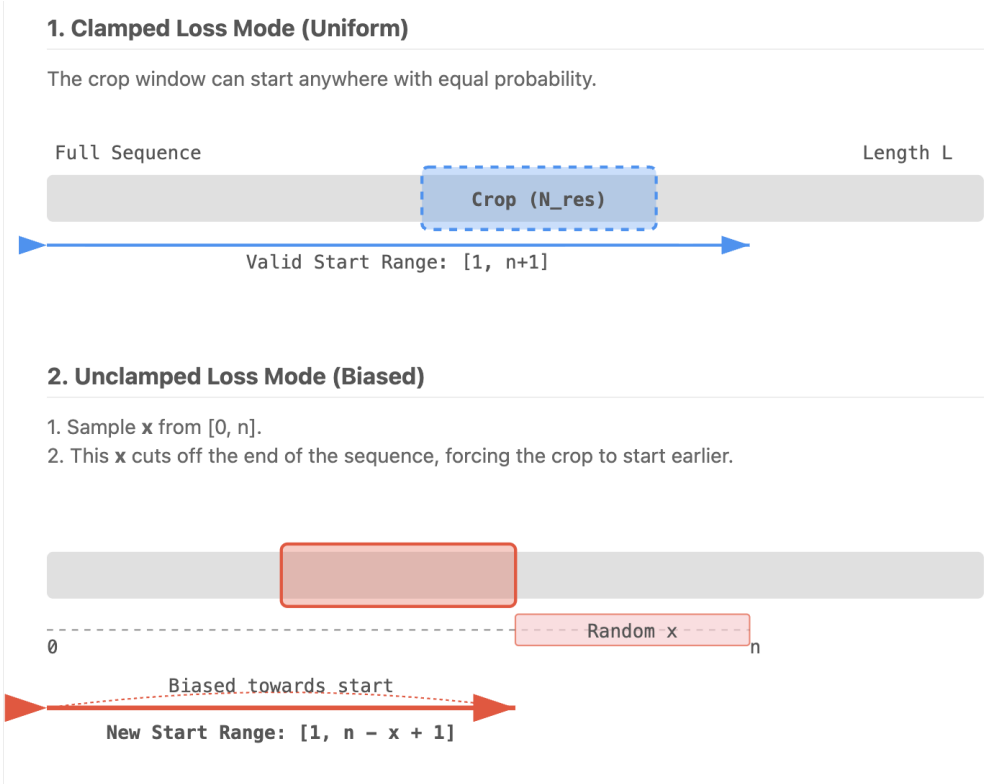
Other details in AlphaFold2: residue cropping

Why we need it:

- Memory Efficiency: prevents memory overflow on GPUs by reducing long protein sequences (which grow quadratically in complexity) to a fixed, manageable size.
- Data Augmentation: forces the model to learn local structures robustly rather than overfitting to specific global positions.

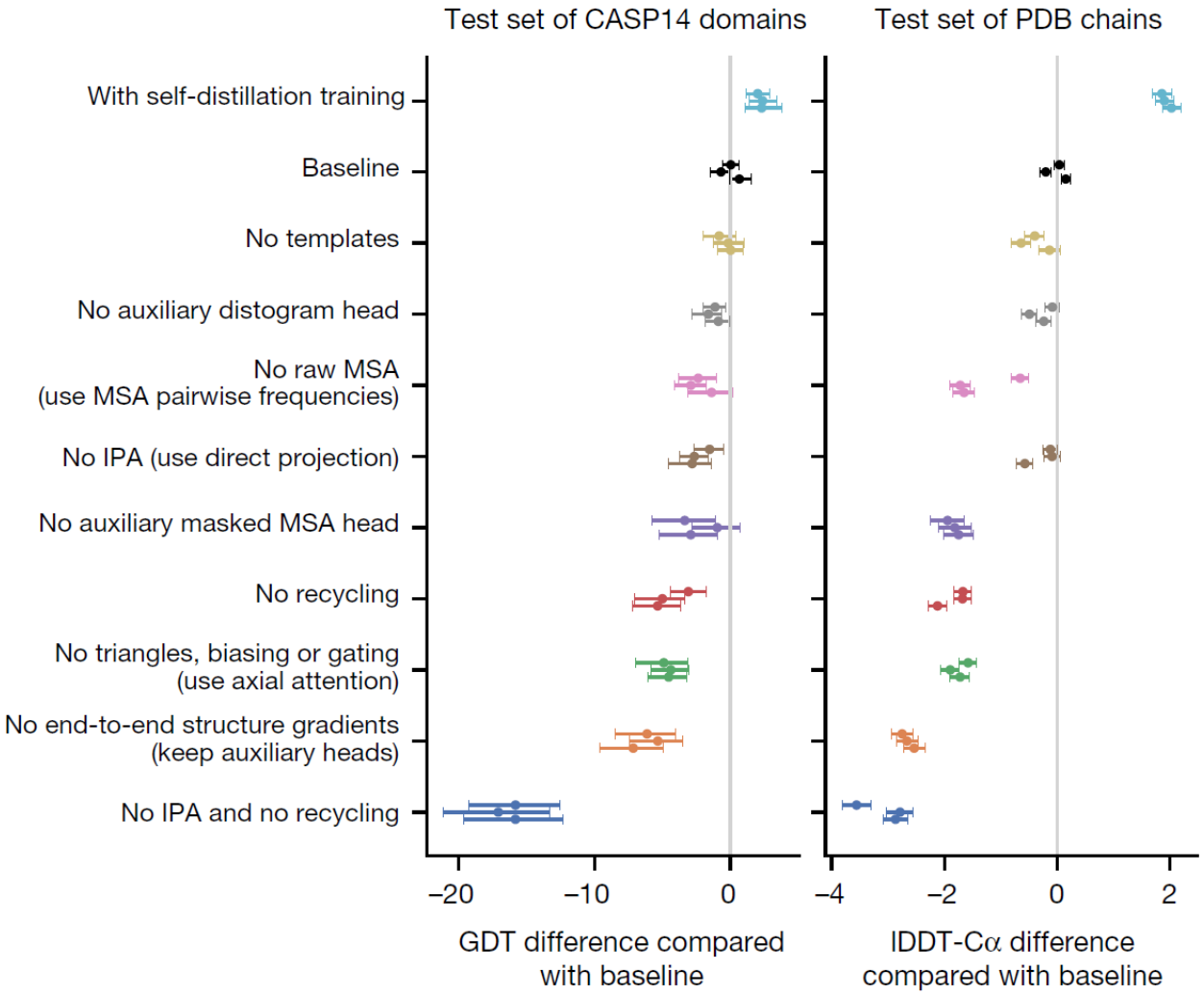
Solution:

- Clamped Loss (Uniform): The crop window is chosen completely randomly; every part of the sequence has an equal chance of being selected.
- Unclamped Loss (Biased): The crop window is chosen using a two-step process that statistically favors the beginning (N-terminus) of the protein sequence.



Other details in AlphaFold2

1. Self-distillation
2. Recycling mechanism

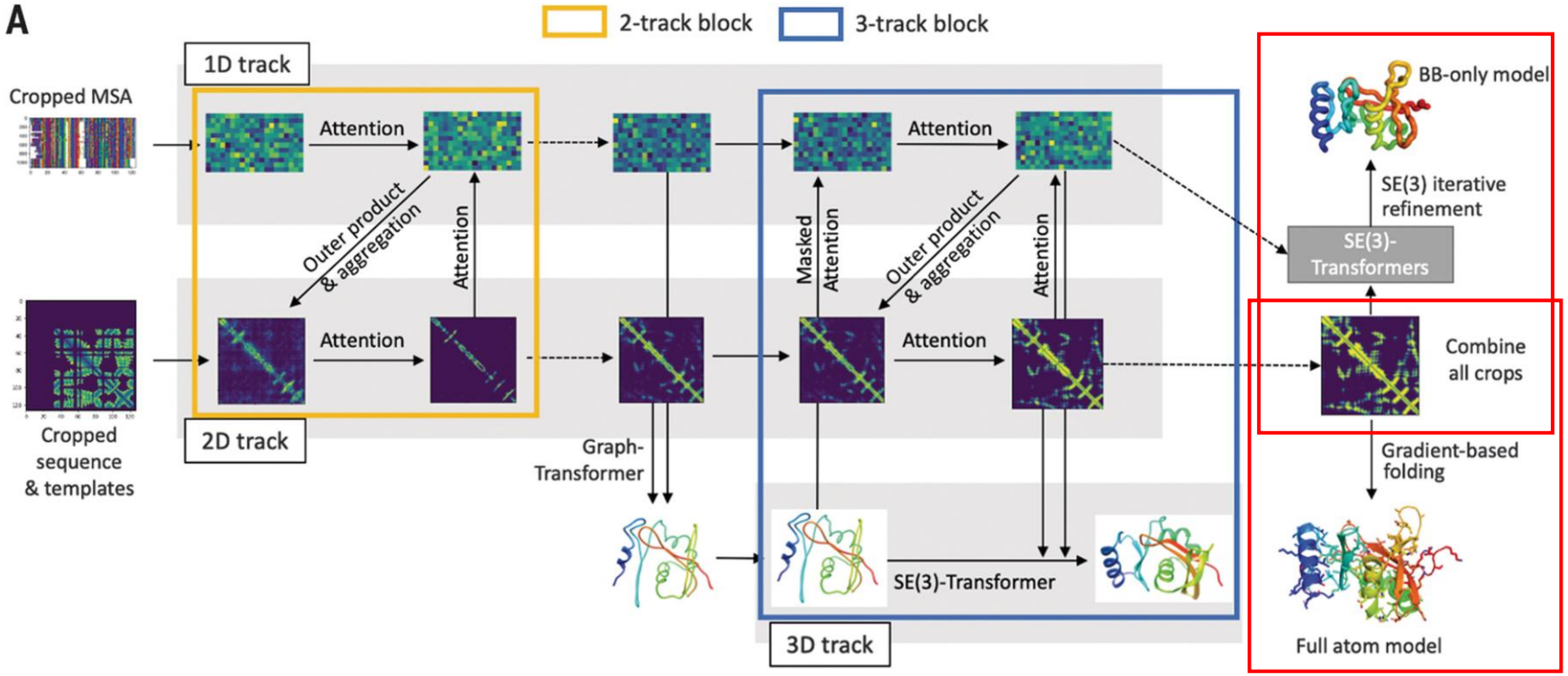


Structure prediction landscape post AF2

- **Optimization, accessibility, and industrialization**
  - From "is it possible?" to "how can we make this usable?"
  - **RoseTTAFold/2 (2021, 2024)**, ColabFold (2021), OpenFold (2022), SimpleFold (2025), etc…
- **From monomers to systems**
  - **AlphaFold-Multimer (2021)**, AF2Complex (2022), etc…
- **Breaking the static barrier (dynamic & conformational ensembles)**
  - **AF-Cluster (2022)**, AlphaFlow (2024), etc…
- **The era of protein language models (pLMs)**
  - OmegaFold (2022), RGN2 (2022), **ESMFold (2022)**, etc…
- **The "All-Atom" revolution**:
  - RoseTTAFold All-Atom (RFAA) (2024), **AF3 (2024)**, etc…
- **Other molecules**: **RNA…**
  - ARES (2021), RhoFold (2024), etc…

RoseTTAFold with a three-track network

- **MSA Track**: Processes evolutionary information from multiple sequence alignments
- **Pair Track**: Processes pairwise relationships between residues
- **Structure Track**: Processes 3D structural information (only in the 3-track network)

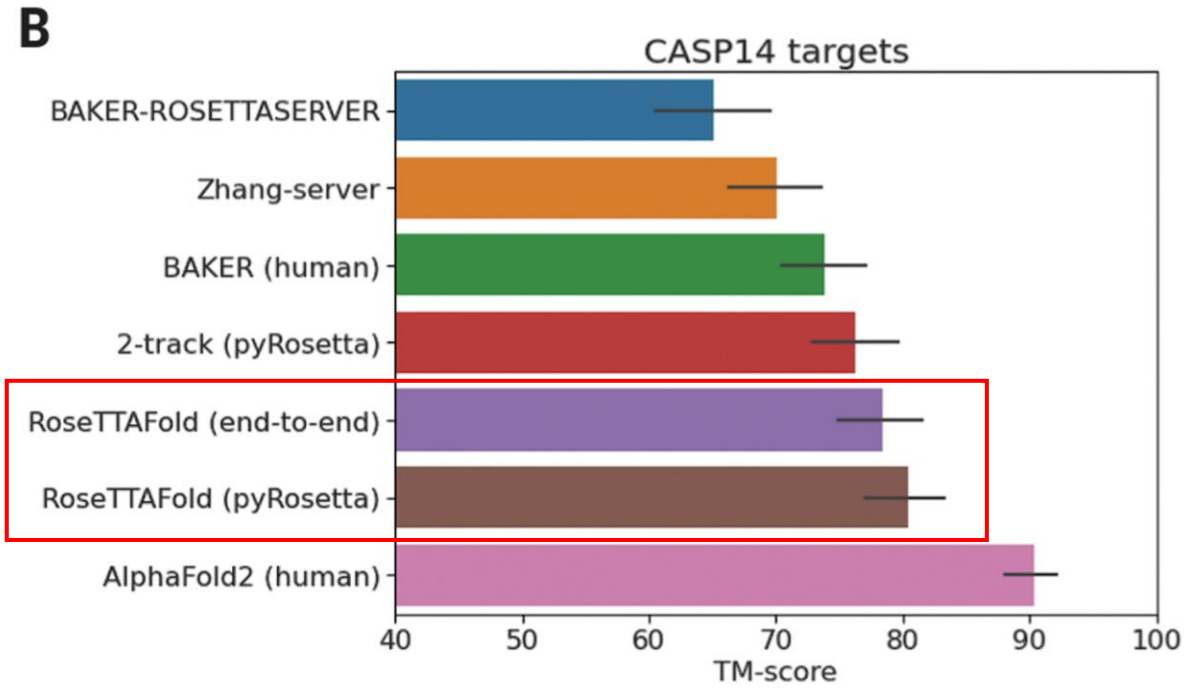


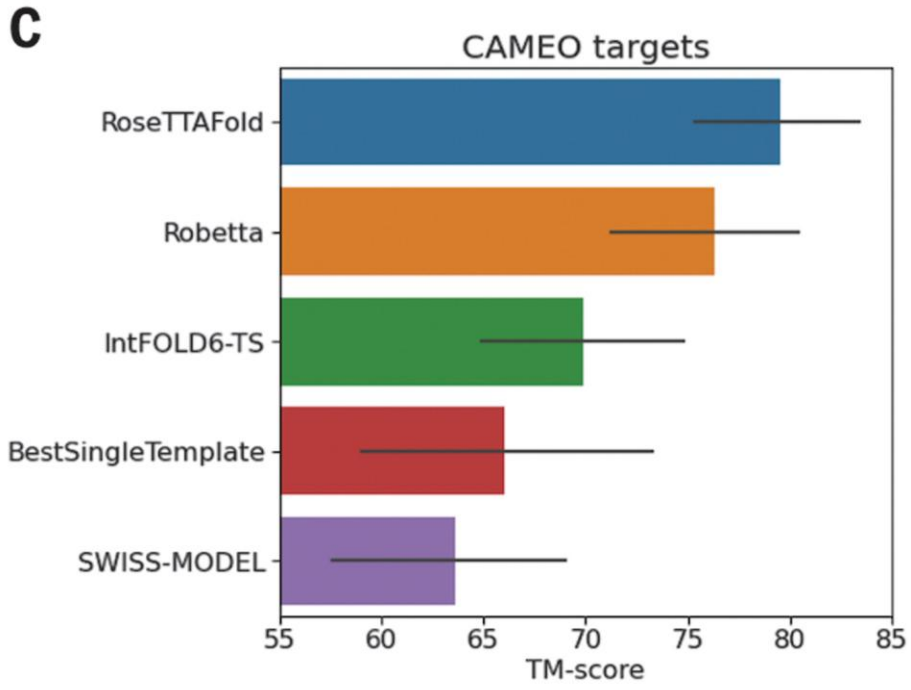
RoseTTAFold (end-to-end): the averaged 1D and 2D features are fed into a final SE(3)-equivariant layer to generate **backbone coordinates**

RoseTTAFold (pyRosetta): the predicted residue-residue distance and orientation distributions are fed into pyRosetta to generate all-atom models

Science, 2021

RoseTTAFold with a three-track network





Science, 2021

Comparison between RoseTTAFold and AlphaFold

Architecture:

1. the inclusion of a third "3D structure" track in the RF basic block
2. the use of biaxial attention in the 2D pair track (RF) versus triangle multiplication and triangle attention (AF2)
3. the use of the SE3-equivariant transformer in RF rather than Invariant Point Attention (IPA) in AF2 for computing structural updates
4. the overall depth of the network, with 8+4 2-track/3-track layers in RF versus 4+48 full-msa/clustered-msa evoformer layers in AF2

Training strategy:

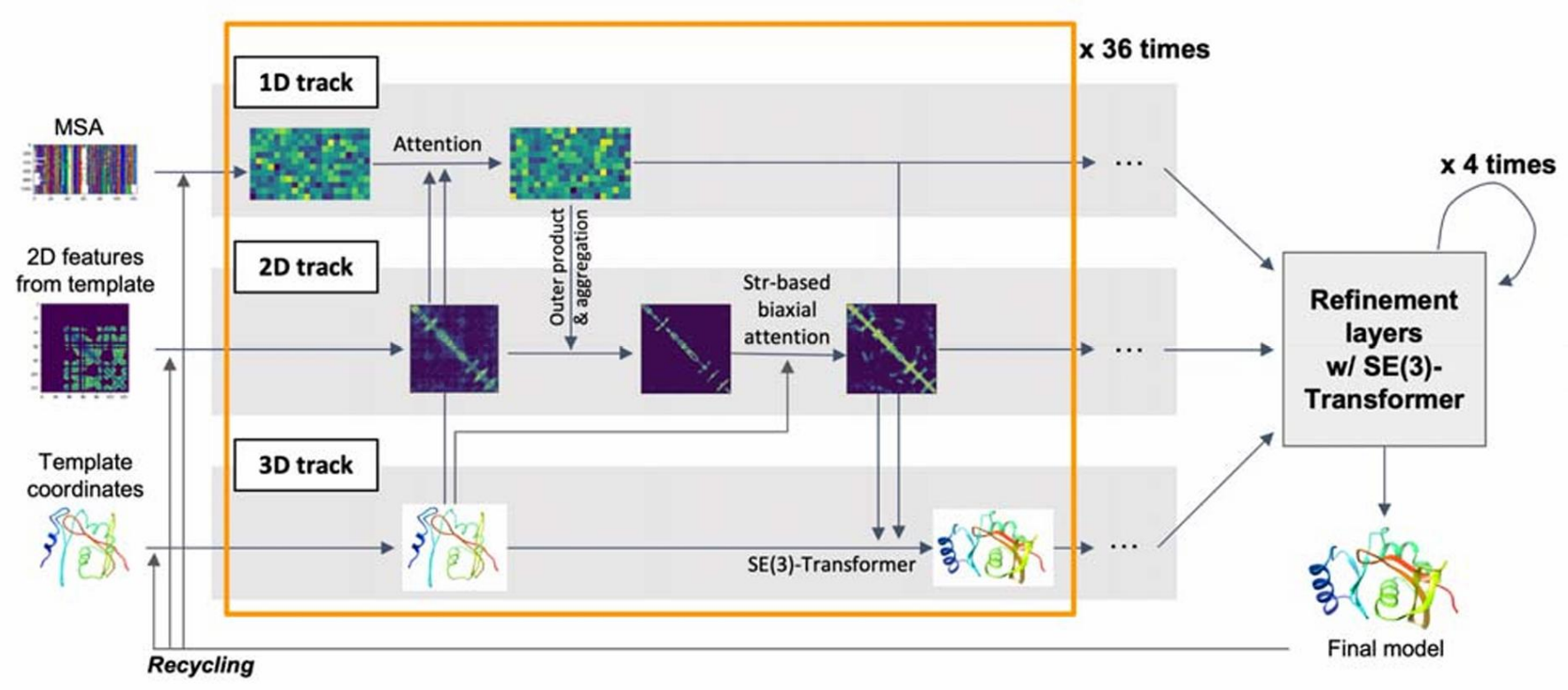
1. the idea of "recycling“ in AF2
2. the use of frame-aligned point error (FAPE) as the structure loss function in AF2
3. the use of distillation data in AF2

Science, 2021

RoseTTAFold2 with a three-track network

- **Aim**: combine the best features of both models
- **Main updates**:

1. Incorporating the concepts of Frame-aligned point error, recycling during training, and the use of a distillation set from AlphaFold2
2. More computationally efficient structure-biased attention as opposed to triangle attention
3. Side chain structure prediction



bioRxiv, 2023

ColabFold: making protein folding accessible to all

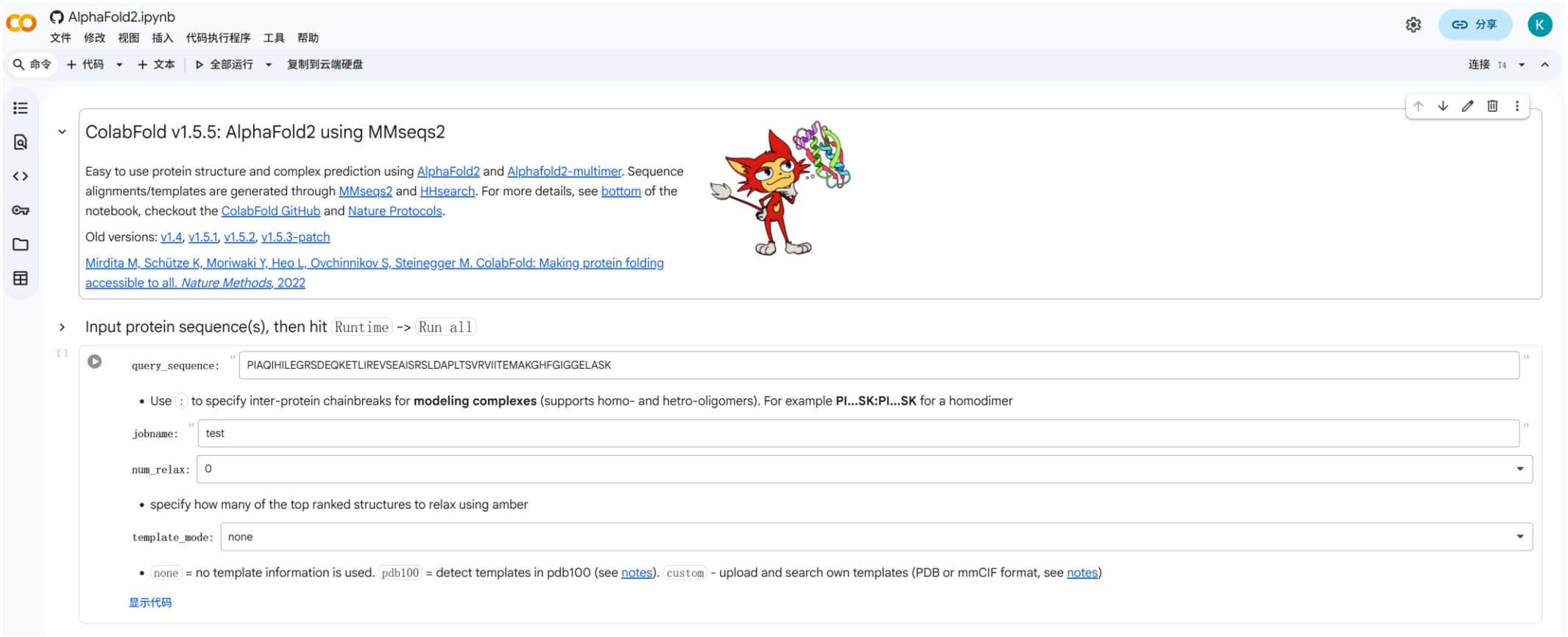
Making Protein folding accessible to all via Google Colab!

Notebooks	monomers	complexes	mmseqs2	jackhmmer	templates
AlphaFold2_mmseqs2	Yes	Yes	Yes	No	Yes
AlphaFold2_batch	Yes	Yes	Yes	No	Yes
AlphaFold2 (from Deepmind)	Yes	Yes	No	Yes	No
relax_amber (relax input structure)					
ESMFold	Yes	Maybe	No	No	No
BETA (in development) notebooks					
RoseTTAFold2	Yes	Yes	Yes	No	WIP
Boltz	Yes	Yes	Yes	No	No
BioEmu	Yes	No	Yes	No	No
OmegaFold	Yes	Maybe	No	No	No
AlphaFold2_advanced_v2 (new experimental notebook)	Yes	Yes	Yes	No	Yes

https://github.com/sokrypton/ColabFold

Nat. Methods, 2022

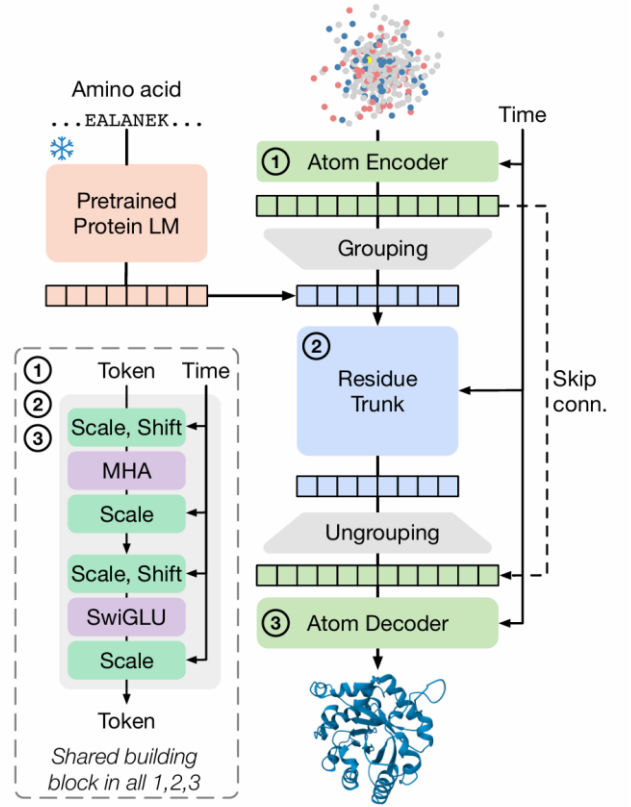
ColabFold: making protein folding accessible to all



Nat. Methods, 2022

SimpleFold: Folding proteins is simpler than you think

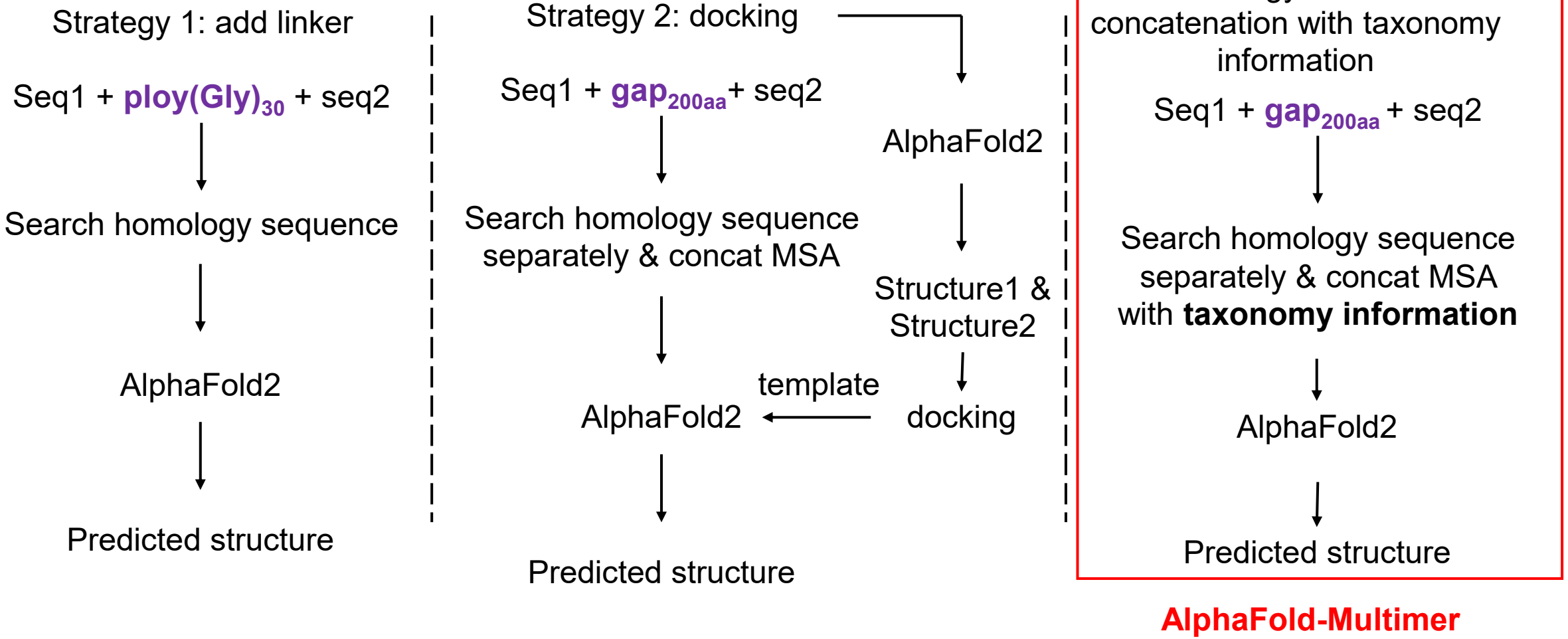
- Built on general-purpose standard Transformer block with adaptive layers.
- Atom encoder, residue trunk, and atom decoder all share the same general-purposed building block. Circumvents the need for pair representations or triangular updates.



Type	Model	TM-score ↑	GDT-TS ↑	LDDT ↑	LDDT-C_α ↑	RMSD ↓
	CAMEO22					
MSA-based	RoseTTAFold (Baek et al., 2021)	0.780 / 0.860	0.715 / 0.775	0.575 / 0.605	0.798 / 0.827	5.721 / 2.864
	AlphaFlow (Jing et al., 2024a)	0.840 / 0.927	0.808 / 0.853	0.741 / 0.798	0.855 / 0.893	3.846 / 2.122
	AlphaFold2 (Jumper et al., 2021)	0.863 / 0.942	0.844 / 0.903	0.816 / 0.856	0.893 / 0.923	3.578 / 1.857
	RoseTTAFold2 (Baek et al., 2023)	0.864 / 0.947	0.845 / 0.904	0.727 / 0.767	0.893 / 0.926	3.571 / 1.707
PLM-based	ESM3 (Hayes et al., 2025)	0.746 / 0.840	0.694 / 0.758	–	–	–
	ESMDiff (Lu et al., 2024a)	0.754 / 0.847	0.701 / 0.760	–	–	–
	EigenFold (Jing et al., 2023)	0.750 / 0.840	0.710 / 0.790	–	–	–
	OmegaFold (Wu et al., 2022)	0.805 / 0.899	0.767 / 0.844	0.746 / 0.815	0.829 / 0.892	5.294 / 2.622
	ESMFlow (Jing et al., 2024a)	0.818 / 0.893	0.774 / 0.832	0.696 / 0.745	0.827 / 0.867	4.528 / 2.693
	ESMFold (Lin et al., 2023b)	0.853 / 0.933	0.826 / 0.875	0.792 / 0.834	0.871 / 0.906	3.973 / 2.019
Ours	SimpleFold-100M	0.803 / 0.878	0.746 / 0.787	0.721 / 0.752	0.822 / 0.852	4.897 / 2.855
	SimpleFold-360M	0.826 / 0.905	0.782 / 0.841	0.773 / 0.803	0.844 / 0.878	4.775 / 2.681
	SimpleFold-700M	0.829 / 0.915	0.788 / 0.845	0.775 / 0.809	0.850 / 0.886	4.557 / 2.423
	SimpleFold-1.1B	0.833 / 0.924	0.793 / 0.851	0.776 / 0.807	0.850 / 0.883	4.350 / 2.334
	SimpleFold-1.6B	0.835 / 0.916	0.799 / 0.864	0.782 / 0.816	0.853 / 0.889	4.397 / 2.187
	SimpleFold-3B	0.837 / 0.916	0.802 / 0.867	0.773 / 0.802	0.852 / 0.884	4.225 / 2.175
	CASP14					
MSA-based	RoseTTAFold (Baek et al., 2021)	0.654 / 0.678	0.562 / 0.572	0.464 / 0.456	0.705 / 0.723	9.676 / 6.420
	AlphaFlow (Jing et al., 2024a)	0.740 / 0.812	0.661 / 0.711	0.632 / 0.662	0.767 / 0.799	7.091 / 3.949
	RoseTTAFold2 (Baek et al., 2023)	0.802 / 0.881	0.740 / 0.824	0.638 / 0.669	0.824 / 0.869	6.744 / 3.292
	AlphaFold2 (Jumper et al., 2021)	0.845 / 0.907	0.783 / 0.855	0.778 / 0.817	0.856 / 0.897	5.027 / 3.015
PLM-based	ESMDiff (Lu et al., 2024a)	0.521 / 0.499	0.447 / 0.430	–	–	–
	ESM3 (Hayes et al., 2025)	0.534 / 0.567	0.459 / 0.488	–	–	–
	EigenFold (Jing et al., 2023)	0.590 / 0.637	0.539 / 0.575	–	–	–
	ESMFlow (Jing et al., 2024a)	0.627 / 0.679	0.539 / 0.544	0.525 / 0.539	0.669 / 0.730	10.503 / 6.974
	OmegaFold (Wu et al., 2022)	0.693 / 0.773	0.625 / 0.723	0.627 / 0.726	0.715 / 0.824	9.845 / 4.042
	ESMFold (Lin et al., 2023b)	0.701 / 0.792	0.622 / 0.711	0.637 / 0.705	0.725 / 0.802	8.679 / 4.016
Ours	SimpleFold-100M	0.611 / 0.628	0.513 / 0.544	0.537 / 0.549	0.659 / 0.685	11.157 / 8.976
	SimpleFold-360M	0.674 / 0.758	0.585 / 0.654	0.617 / 0.657	0.703 / 0.762	9.382 / 4.828
	SimpleFold-700M	0.680 / 0.767	0.591 / 0.668	0.630 / 0.674	0.714 / 0.763	9.289 / 4.431
	SimpleFold-1.1B	0.697 / 0.796	0.607 / 0.668	0.640 / 0.676	0.723 / 0.758	9.249 / 4.462
	SimpleFold-1.6B	0.712 / 0.801	0.630 / 0.709	0.660 / 0.699	0.741 / 0.798	8.424 / 4.722
	SimpleFold-3B	0.720 / 0.792	0.639 / 0.703	0.666 / 0.709	0.747 / 0.829	7.732 / 3.923

arXiv, 2025

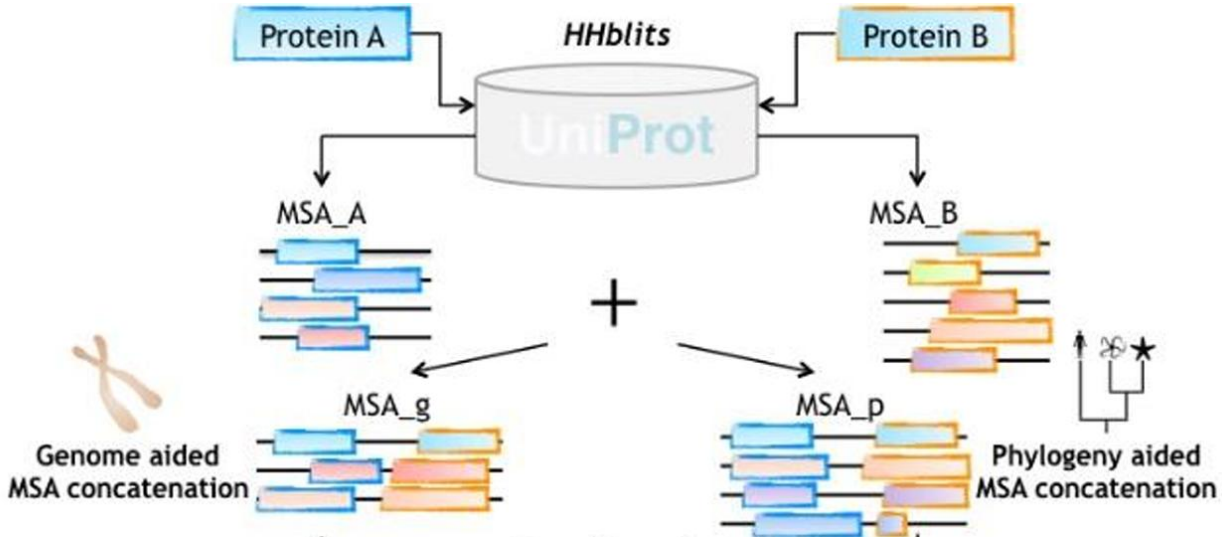
Enhanced MSA for complex structure prediction



AlphaFold-Multimer

AlphaFold-Multimer for complex structure prediction

Cross-chain genetics

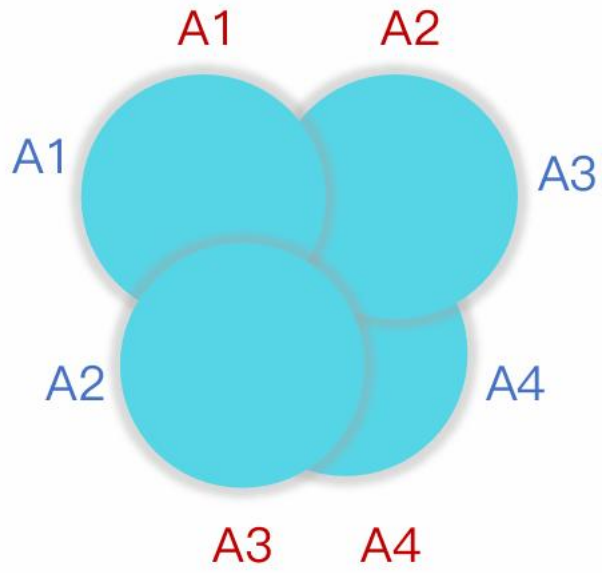


bioRxiv, 2018

Pair sequences using the UniProt species annotation and resolve ambiguities following Zhou et al..

- **Prokaryotic**: smallest genetic distance
- **Eukaryotic**: rank by similarity to their respective target sequence, and concatenate pairs of the same rank.

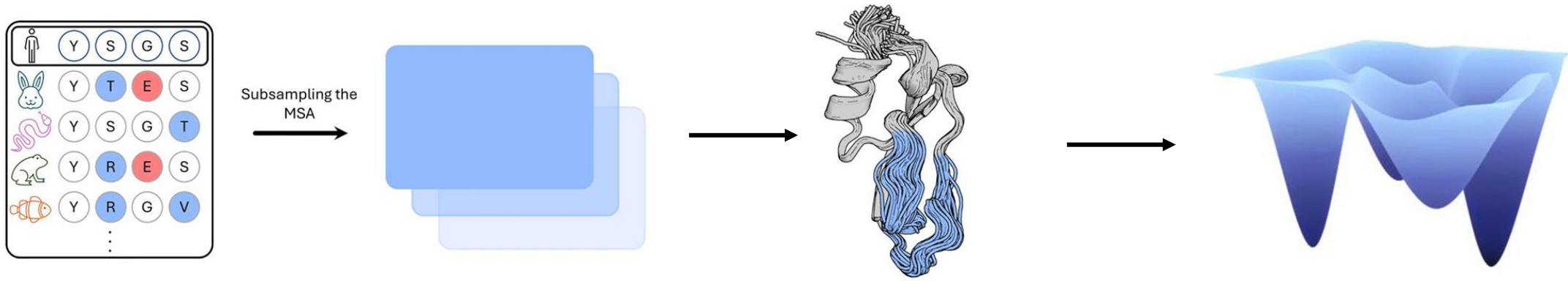
Multi-Chain permutation alignment



When a protein of a given sequence appears multiple times in the complex, the mapping between the predicted and the ground-truth coordinates is arbitrary.
Employ a simple **heuristic** that greedily tries to find a good permutation.

Protein dynamics and conformational ensembles prediction

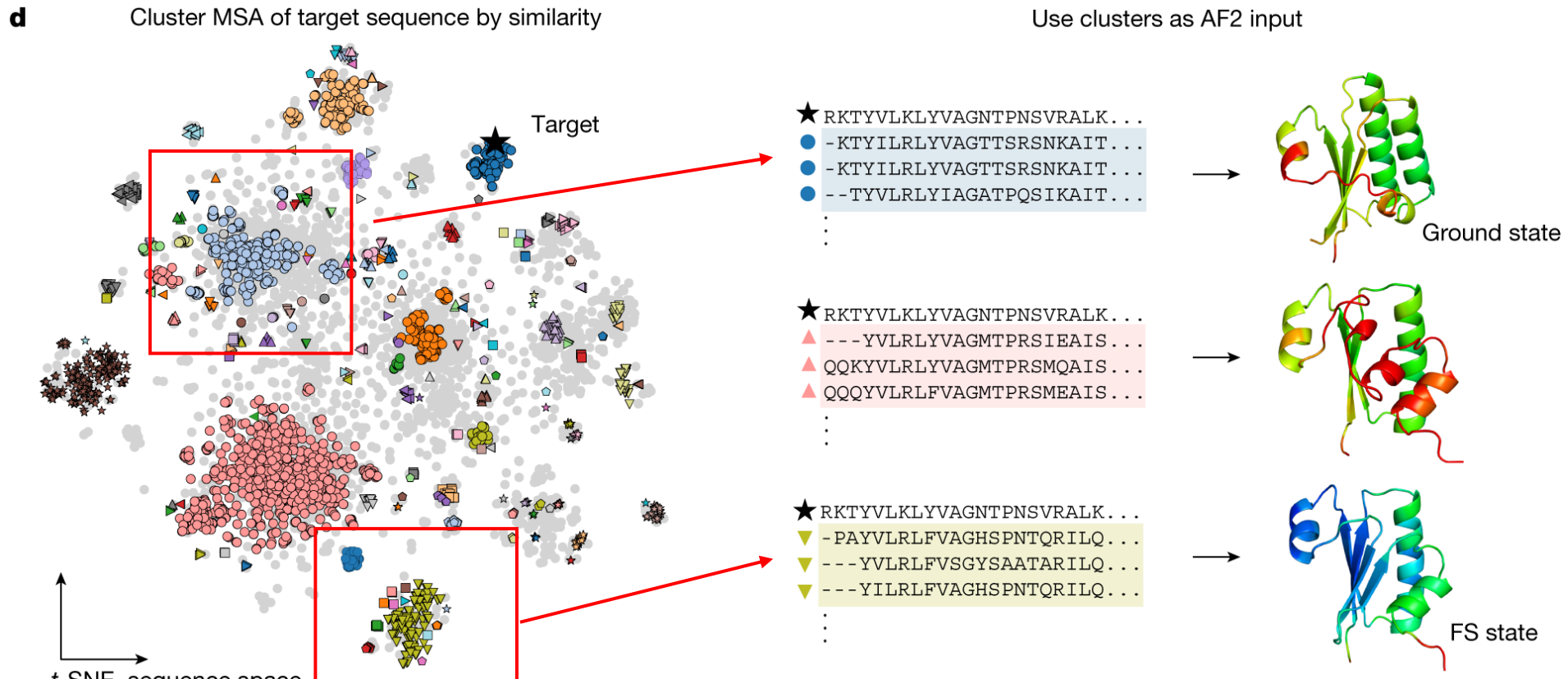
- Solution: MSA clustering and subsampling



From which paper?
Seems not enough.

Protein dynamics and conformational ensembles prediction

AF-Cluster: clustering a multiple-sequence alignment by sequence similarity enables AlphaFold2 to sample alternative states of known metamorphic proteins with high confidence.



Nature, 2024

Protein language model for coevolution information modeling

- LMs are powerful in text understanding & generation
  - ChatGPT, Llama, Copilot, BERT…
  - Statistical characteristics of texts can be modeled via pre-training

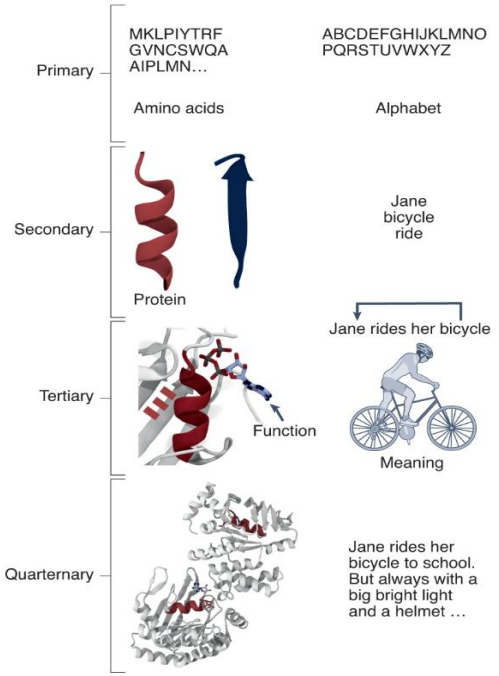
- Pre-training PLMs on vast protein sequences
  - Model coevolutionary information, capture the interdependencies between residues
  - Generative design

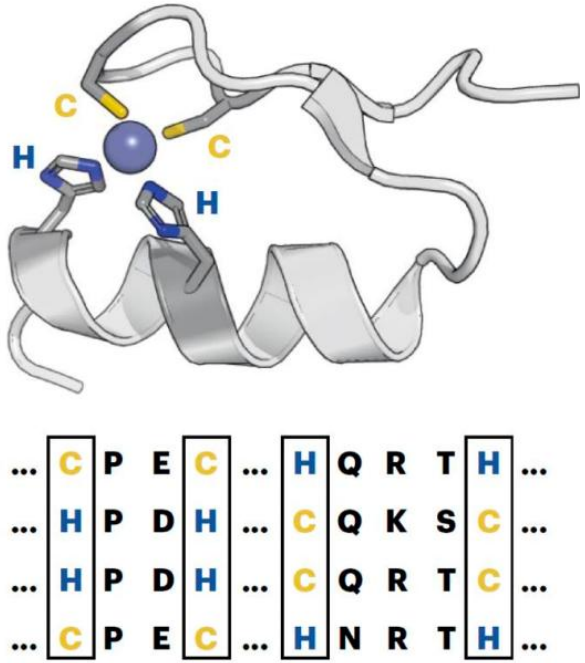
Sequence

Secondary structure

Tertiary structure

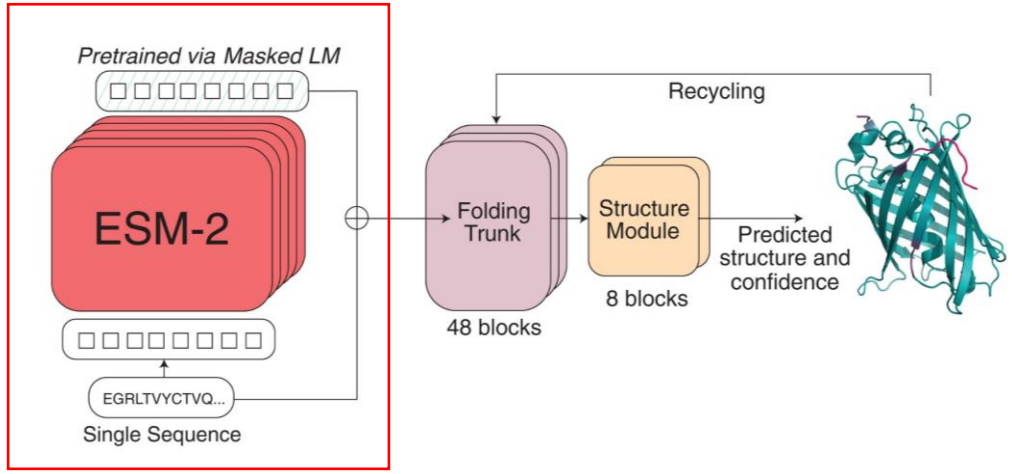
Quaternary structure

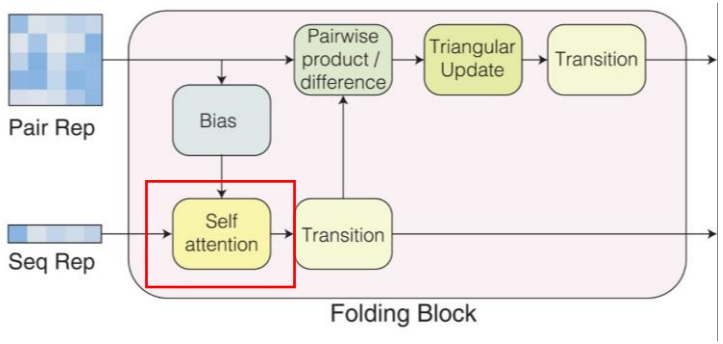




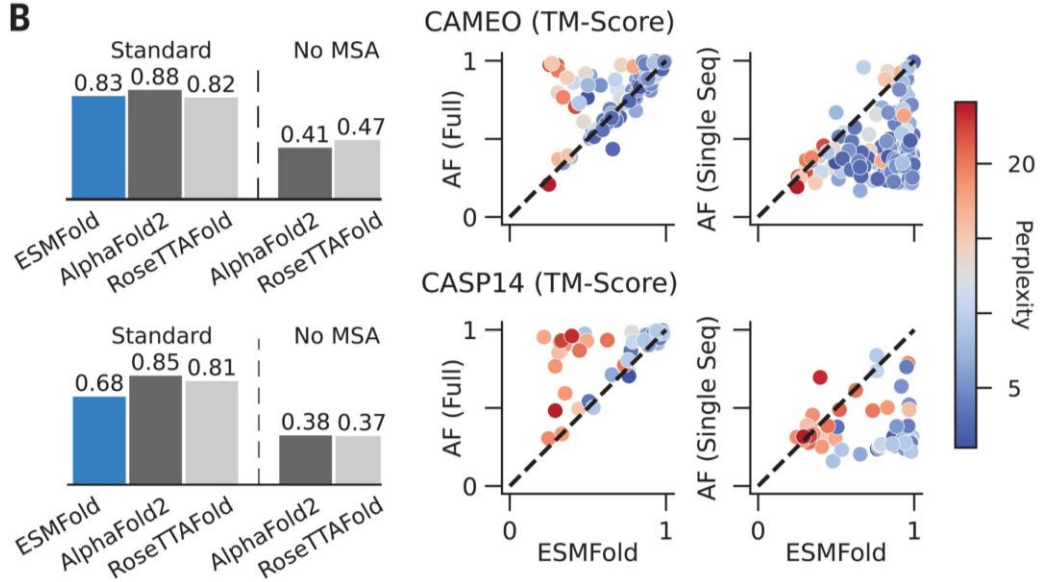
ESMFold enables fast and accurate single sequence prediction with a pre-trained protein language model

Extract coevolutionary information





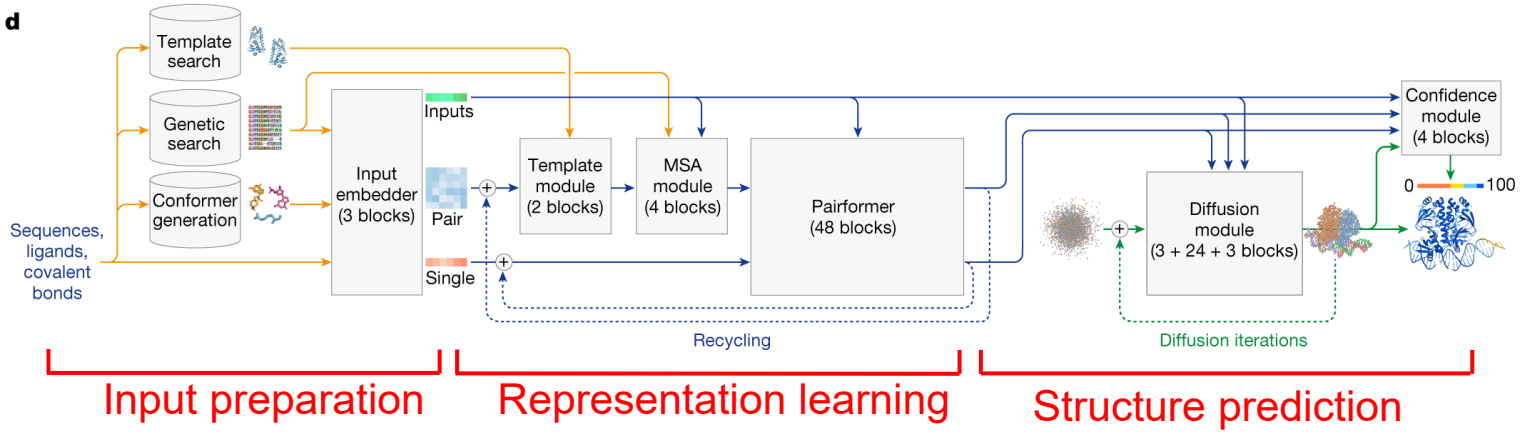
In AlphaFold2: row- and column-wise attention



Science, 2023

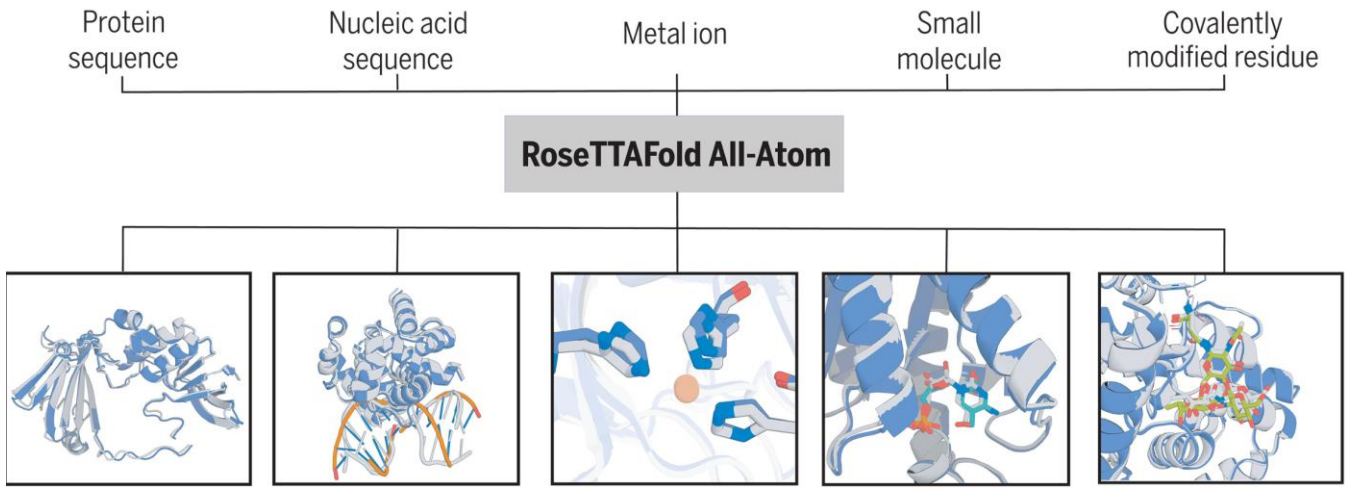
Structure prediction for all biomolecular types

- Predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues
- Example: AlphaFold3, RFAA, …



AlphaFold3

Nature, 2024

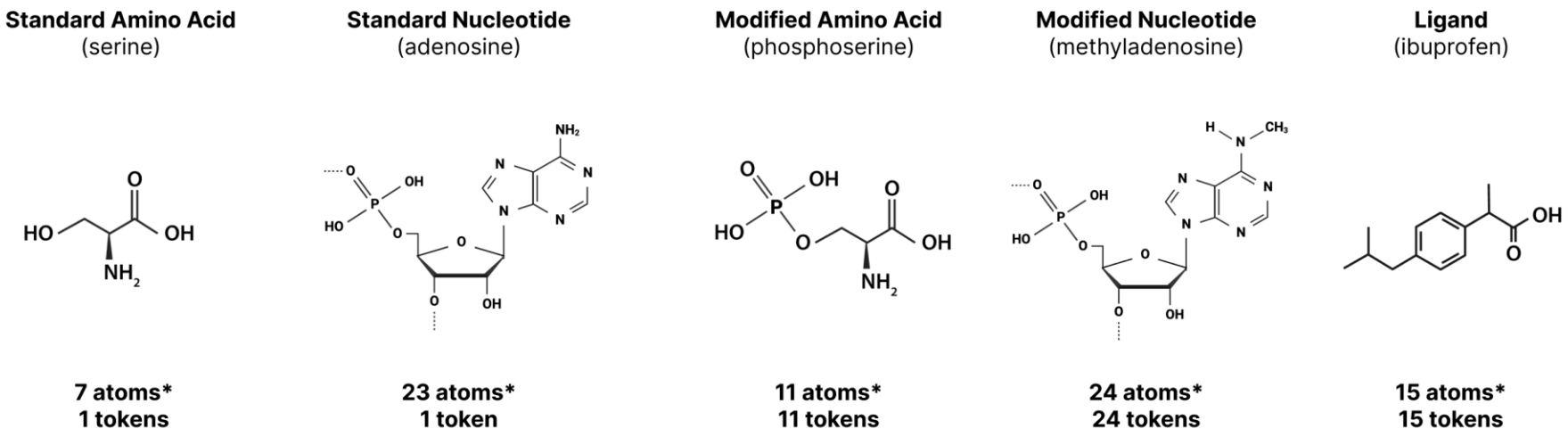


RFAA

AlphaFold3 for multiple input type structure prediction

- **Single representation:** represents all the tokens in the complex structure
- **Pair representation**: represents the relationships between all pairs of amino acids / atoms
- Each of these can be represented at an **atom-level** and **a token-level.**
  - Standard amino acid: 1 token (as per AF2)
  - Standard nucleotide: 1 token
  - Non-standard amino acids or nucleotides (methylated nucleotide, amino acid with post-translational modification, etc.): 1 token per atom
  - Other molecules: 1 token per atom

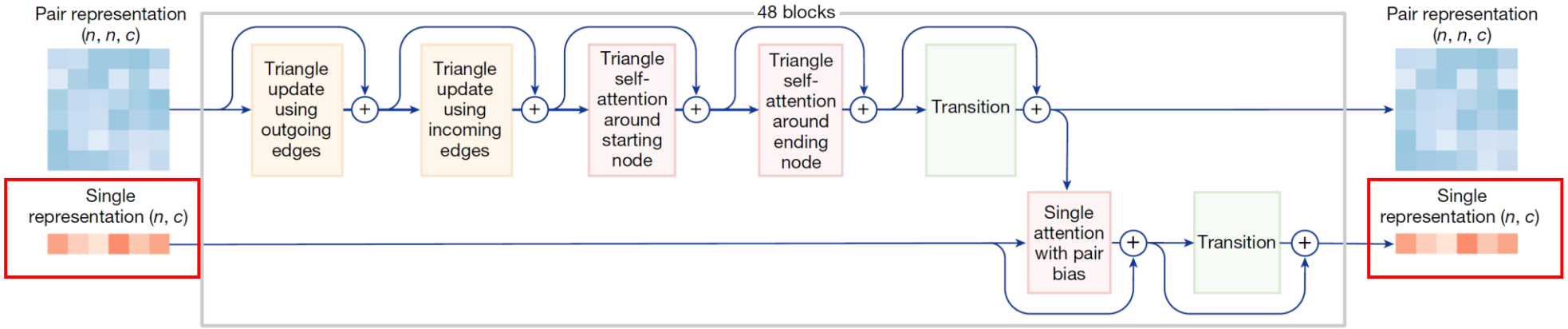
Universal tokenization approach



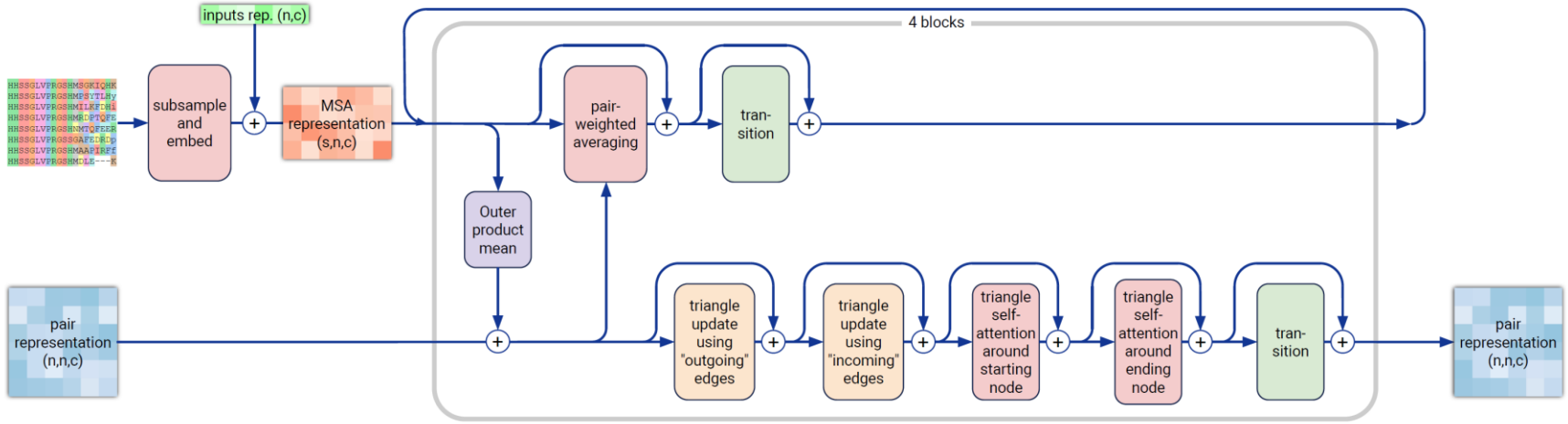
*atoms=heavy atoms

AlphaFold3 for multiple input type structure prediction

- Operate on the single representation, the MSA representation is not retained
- MSA is processed with a smaller and simpler MSA embedding block



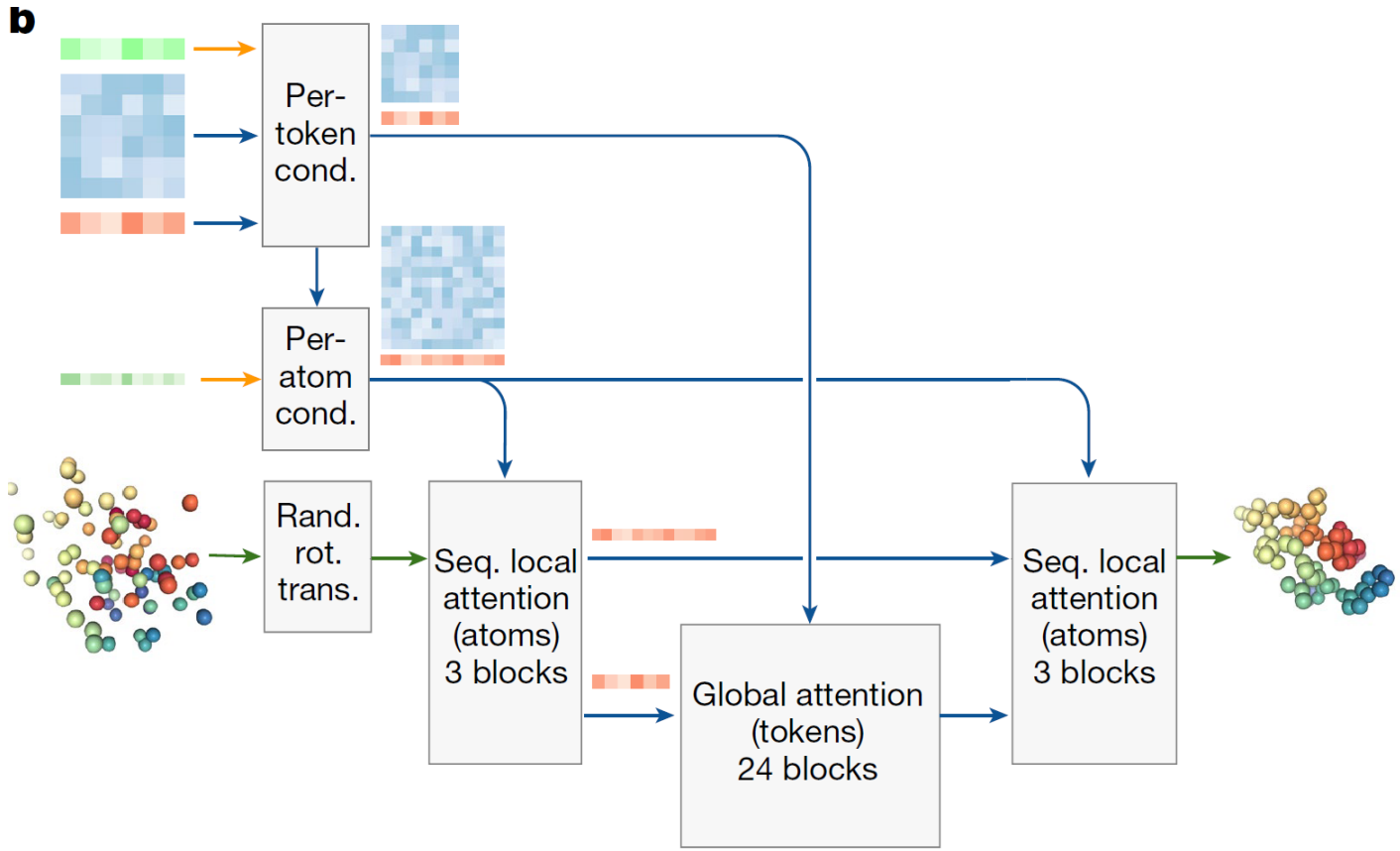
Pairformer architecture



MSA module

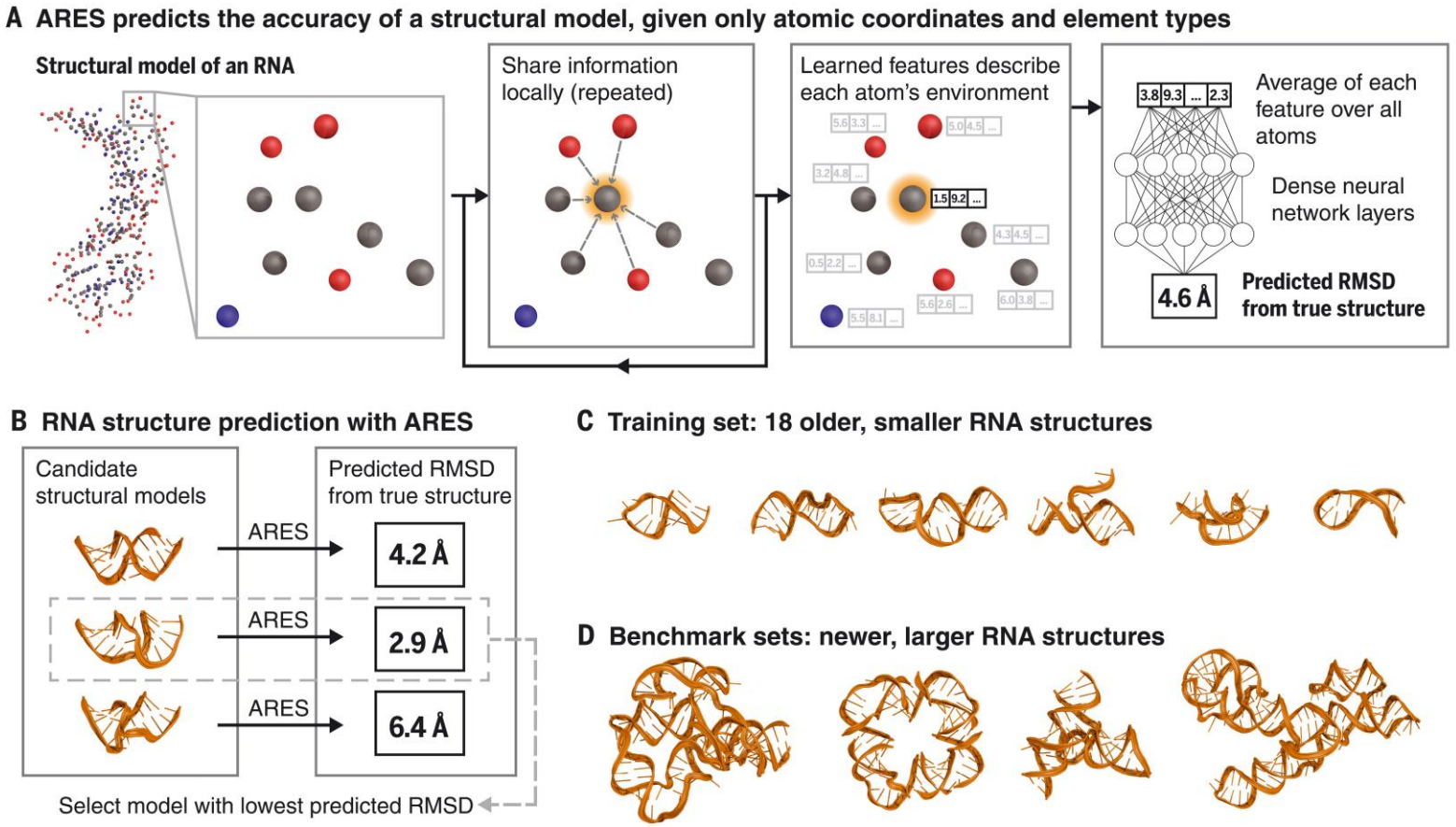
AlphaFold3 for multiple input type structure prediction

A novel **diffusion module** operates directly on raw atom coordinates, and on a coarse abstract token representation, without rotational frames or any equivariant processing.



RNA structure prediction

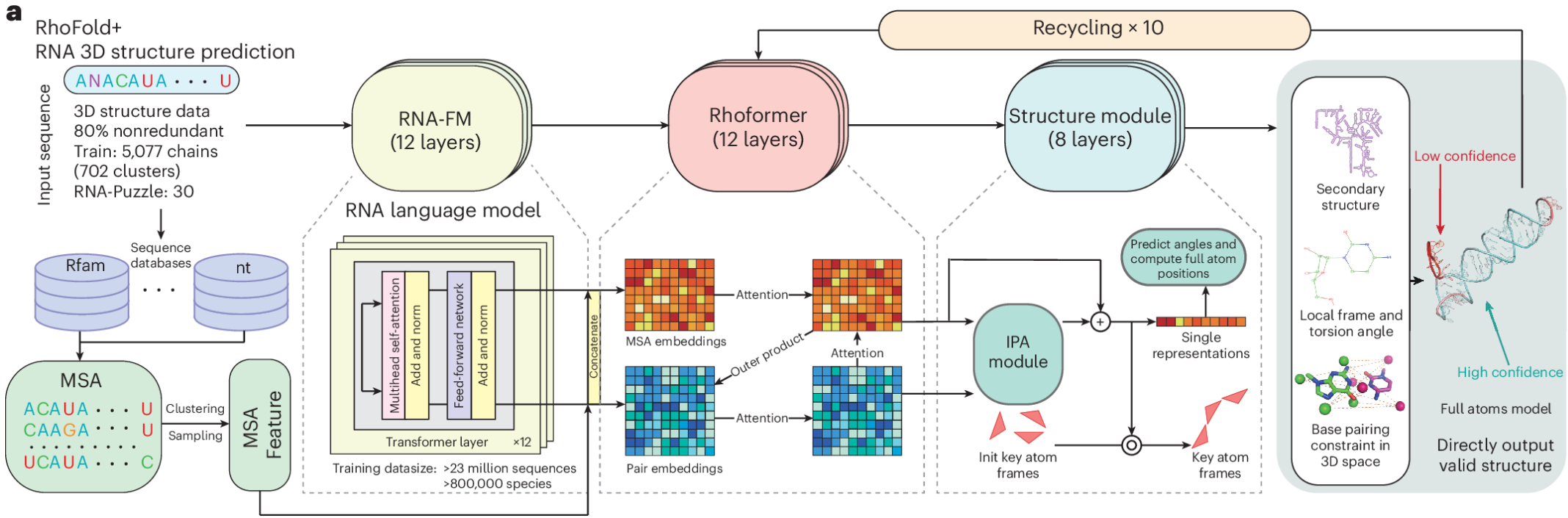
- Example: ARES



Science, 2021

RNA structure prediction

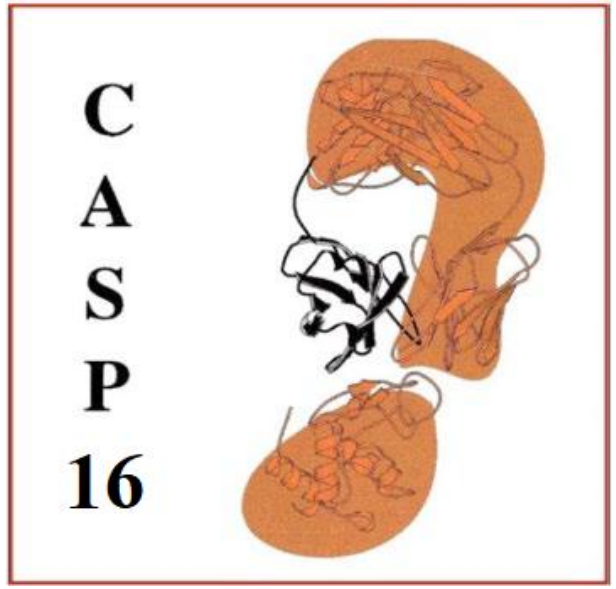
- Example: RhoFold, …



Nat Methods, 2024

The future of structure prediction: opportunities and challenges coexist

- Complex structure prediction
- Protein dynamics and conformational ensembles prediction
- Sensitivity to MSA quality, performs poorly on proteins with few or no known evolutionary relatives (orphan proteins)
- Ligand, nucleic acid, ion binding prediction, modifications
- Structures of proteins with point mutations
- Structures of disordered regions of proteins



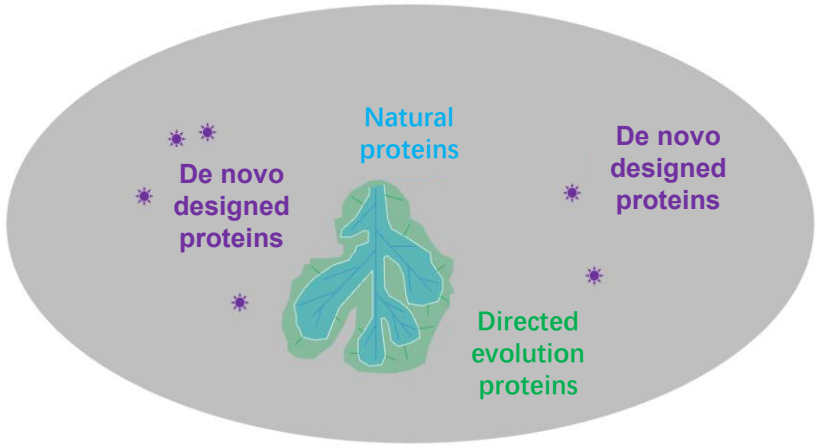
Protein design

- **Problem definition**: designing protein/NA sequences with desired structures and functions.
- **Approaches**:
  - Inverse folding (design sequence with given backbone structure)
  - *De novo* design (design structure and sequence)

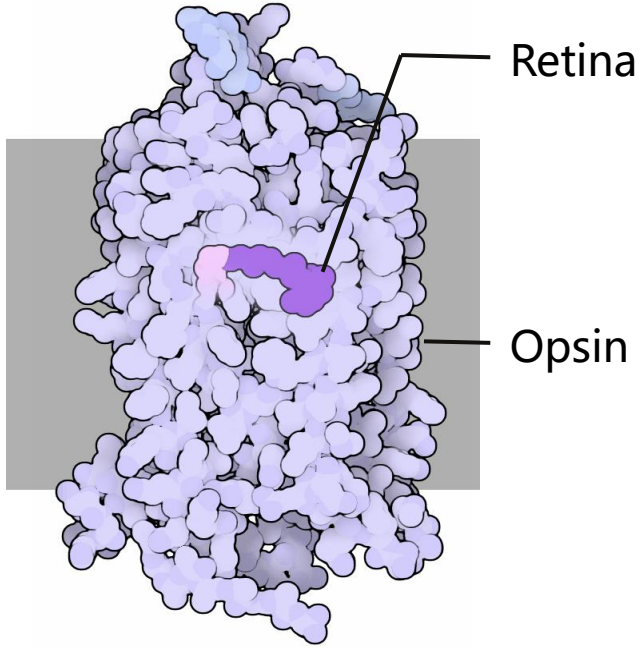
Protein universe

- **The Problem:** Nature evolved proteins for *survival*, not for our specific modern needs (e.g., curing cancer, breaking down plastic, capturing carbon).
- **The Math:** Explain the combinatorial explosion.
  - A small protein has 100 amino acids. 20^{100} possibilities.
  - There are more possible protein sequences than atoms in the universe. **Nature has sampled only a tiny droplet of this ocean.**
- **The Goal:** Protein design is about exploring the parts of that ocean nature never touched

Diagram of the protein universe

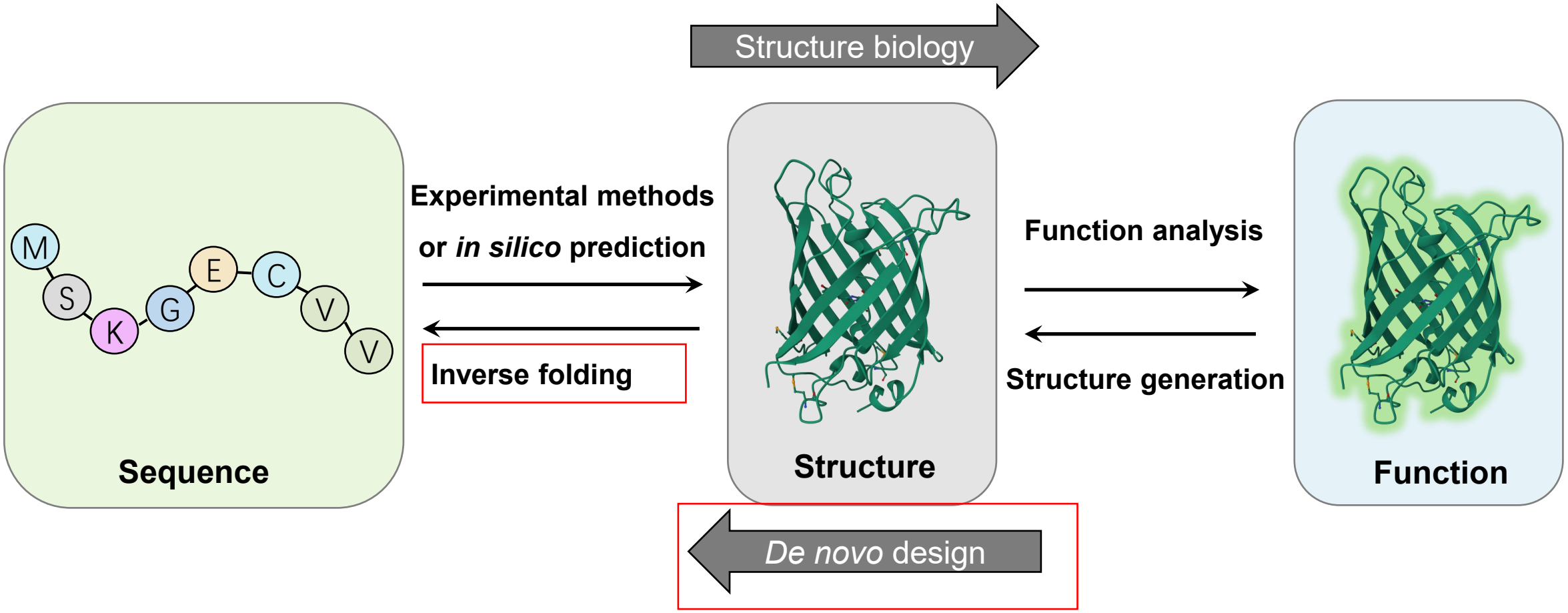


- **Natural proteins**: a small portion of the protein universe
- **Directed evolution proteins**: extend around the natural protein area
- **De novo designed proteins**: able to explore the giant protein universe



Rhodopsin

Sequence-structure-function relationship



Phase I: Emergence & Rational design (1970s – 1990s)

- **Core Characteristics:**
  - Manual design based on intuition and physicochemical rules.
  - Heavy reliance on site-directed mutagenesis.
- **Key Achievements:**
  - **Helix Bundles (The "De Novo" Start):**
    - William DeGrado utilized hydrophobic interactions and helix propensity.
    - **Design Logic:** Simple intuition—Hydrophobic residues inside (core), hydrophilic outside (solvent).
  - **Zinc Fingers (Functional Design):**
    - Carl Pabo redesigned DNA-binding domains to target specific sequences.
    - A classic example of early functional re-design.
- **Limitations:**
  - Restricted to **Re-designing** natural backbones or simple secondary structures.
  - **Lack of computational tools:** Relied on manual trial-and-error with a very low success rate.

Rational design



(a) $(LKKLLKL)_2$ Leu-Lys-Lys-Leu-Leu-Lys-Leu-Leu-Lys-Lys-Leu-Leu-Lys-Leu

α_1A Gly-Glu-Leu-Glu-Glu-Leu-Leu-Lys-Lys-Leu-Leu-Glu-Glu-Leu-Lys-Gly

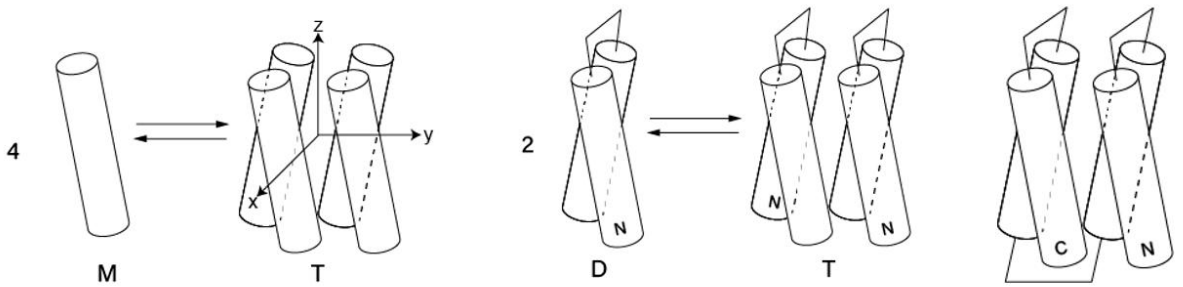
α_1B Gly-Glu-Leu-Glu-Glu-Leu-Leu-Lys-Lys-Leu-Lys-Glu-Leu-Leu-Lys-Gly

Loop Pro-Arg-Arg

α_2B α_1B–Loop–α_1B

α_4B α_1B–Loop–α_1B–Loop–α_1B–Loop–α_1B

(b)



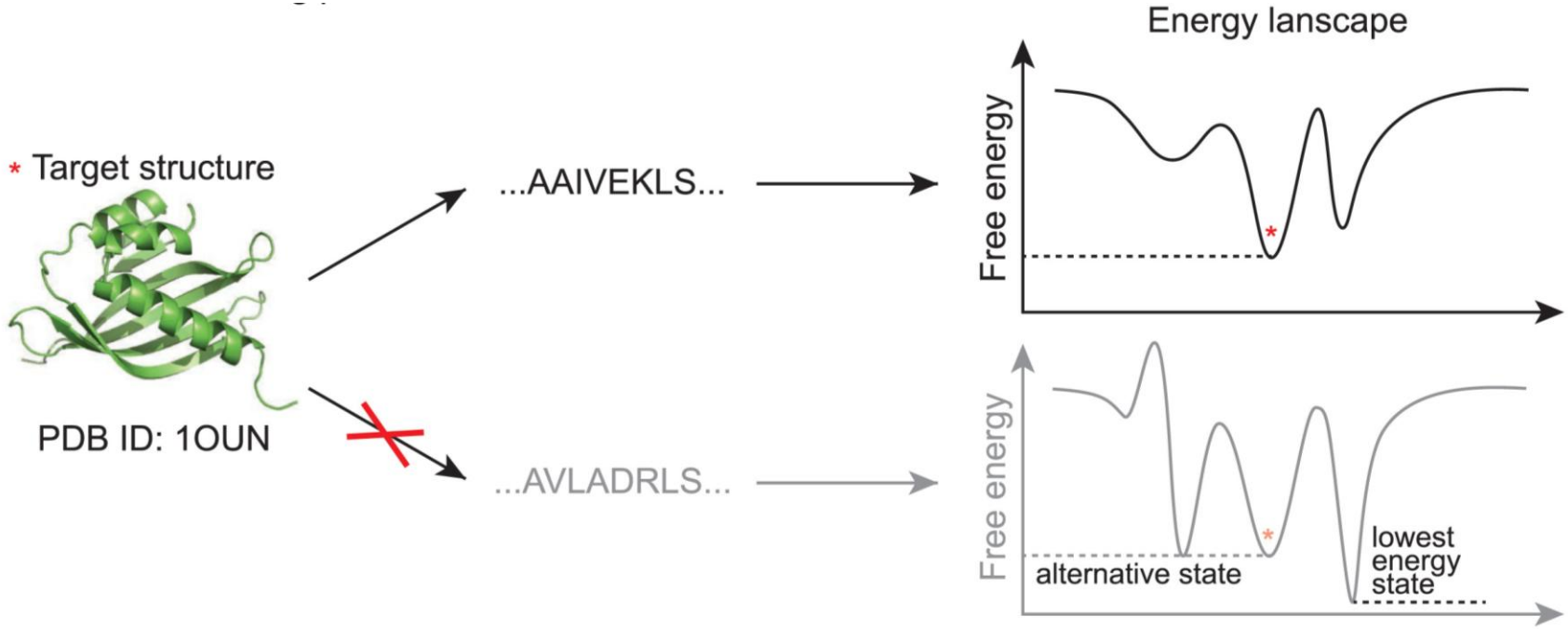
The design of an α-helice bundle. A. Sequences of peptides and proteins designed by DeGrado and his colleagues are shown.**B.** Fragments capable of folding into an antiparallel helical bundle is displayed. Next, a loop sequence is inserted between two helices to create a two-helix bundle. Then, three loops are inserted between four helices to form the full-length helical bundle.

Phase II: The Computational Revolution (1990s – 2020)

- **Core Characteristics:**
  - Establishment of Energy Functions.
  - The birth of **Rosetta**.
  - Breakthroughs in *De Novo* (from scratch) design.
- **The Rosetta Era (David Baker Lab):** Rosetta became the cornerstone software.
  - **Hybrid Energy Function:** Combined **Physical terms** (Van der Waals, electrostatics, H-bonds) with **Statistical terms** (derived from PDB data).
  - **Algorithms:** Used **Monte Carlo** simulations and **Simulated Annealing** to search vast conformational spaces for the lowest energy state.
- **Key Innovation: Rotamer Packing:**
  - Introduced "Rotamer Libraries" (Side-chain conformation libraries).
  - Solved the complex problem of optimizing side-chain packing onto a fixed backbone.

Inverse folding

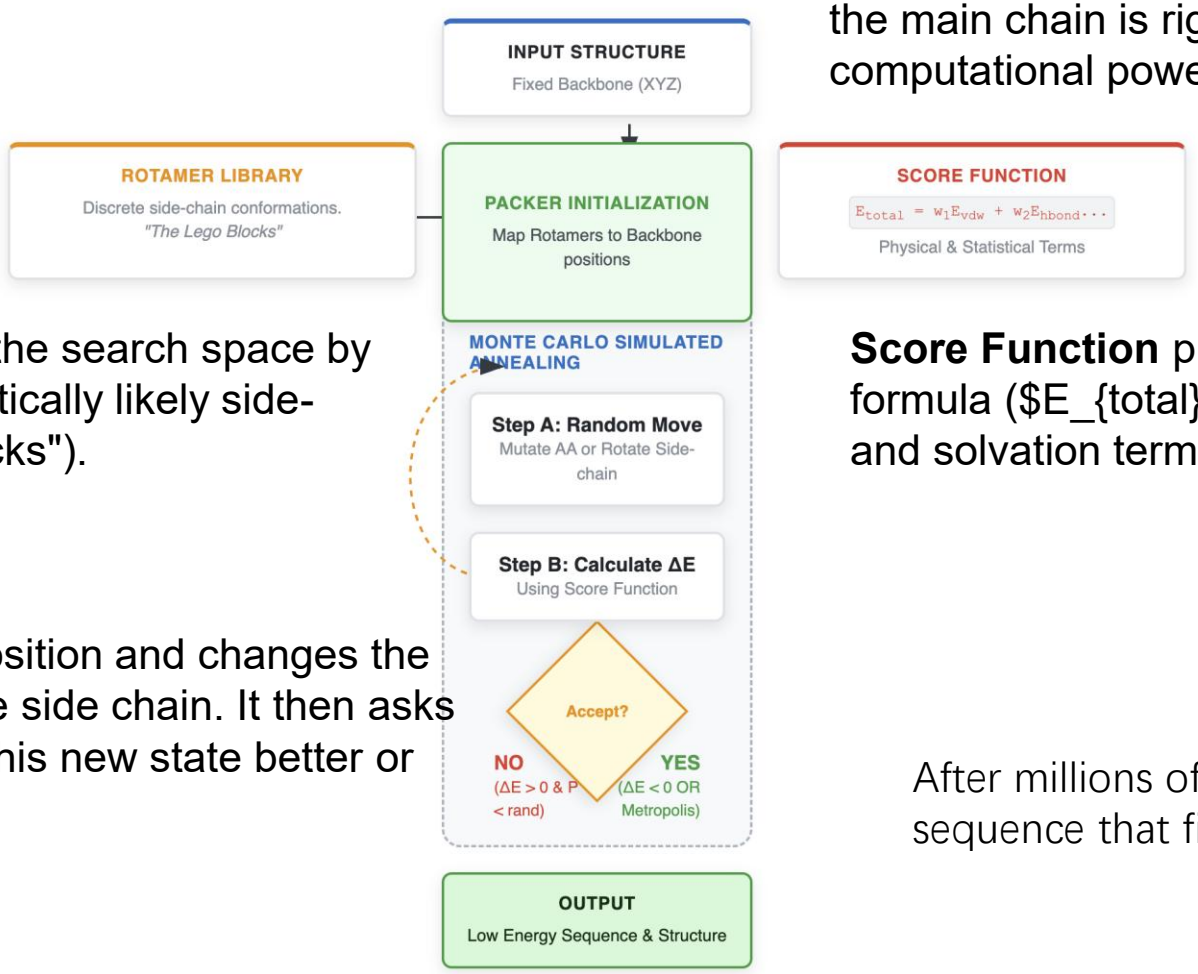
- Problem definition: designing a sequence that has the target structure as the unique and accessible lowest energy conformation.
- Strategies:
  - Straightforward idea: predict structure and select sequence with lowest energy ➡ Computation intensive
  - Deep learning methods



Rosetta fixed backbone design



The process begins with the **Fixed Backbone**. Early Rosetta assumes the main chain is rigid to save computational power.



Rotamer Library limits the search space by providing discrete, statistically likely side-chain angles ("Lego blocks").

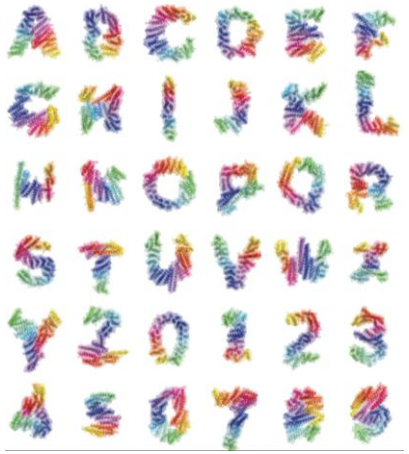
Score Function provides the mathematical formula (E_{total}) containing VDW, H-bonds, and solvation terms to evaluate every move.

The algorithm picks a position and changes the amino acid or rotates the side chain. It then asks the Score Function: "Is this new state better or worse?"

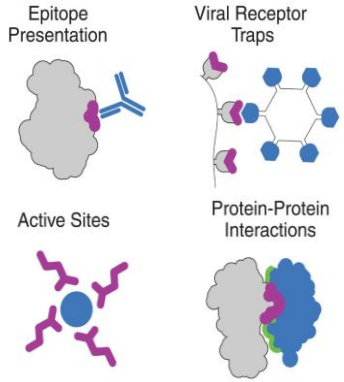
After millions of cycles, the system converges on a sequence that fits the backbone with the lowest energy

De novo protein design

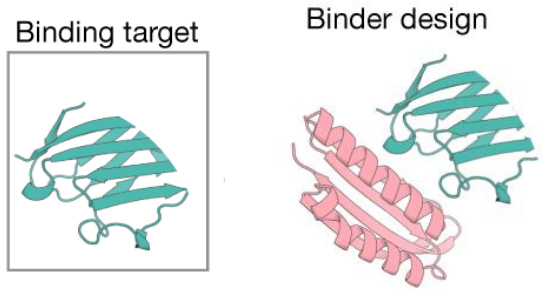
- Question not answered by inverse folding: how to obtain novel backbone structure?
- *De novo* structure design:



design novel topology



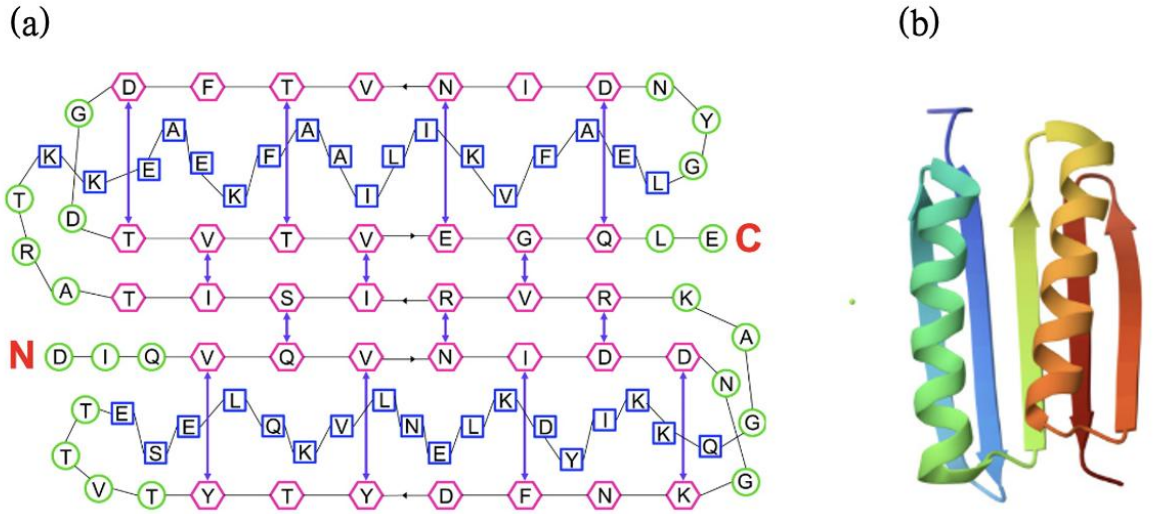
design proteins with desired functional sites



design proteins binding to desired target

Success with Rosetta





A. The sequence and secondary structure of Top7. N and C denote the N-terminus and C-terminus of the protein, respectively. Blue, pink, and green represent α-helices, β-strands, and loop regions, respectively;

B. The three-dimensional structure of Top7.

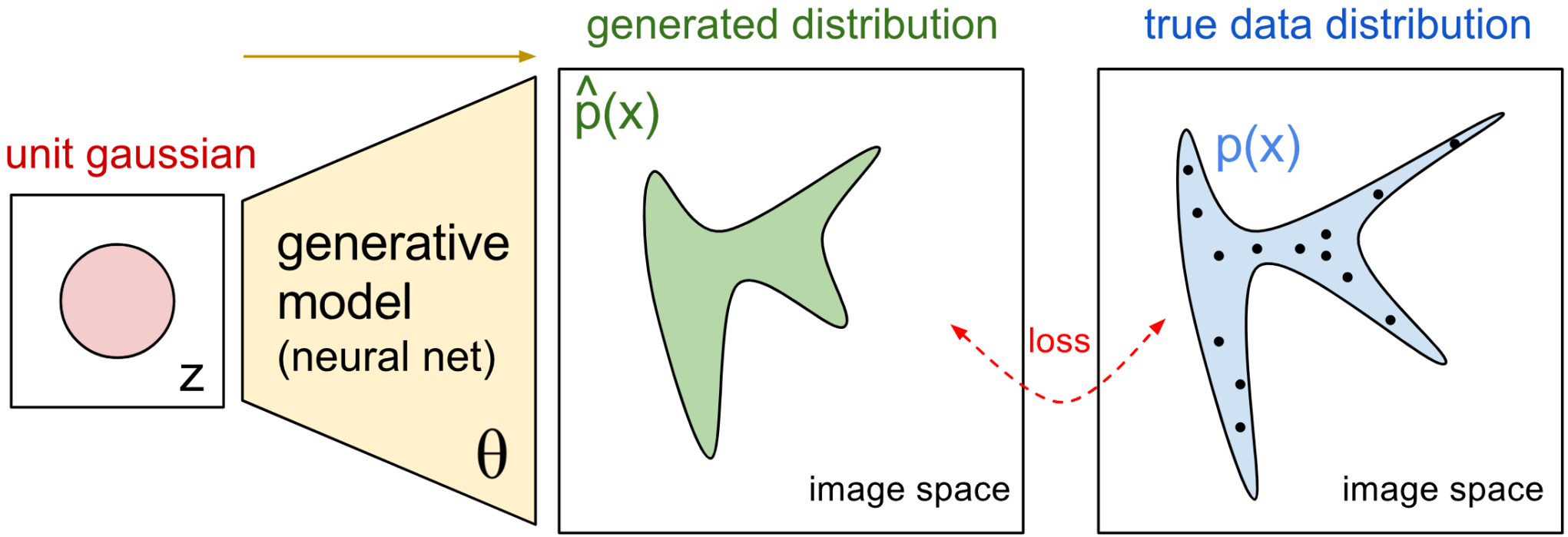
Phase III: The AI & Deep Learning Revolution (2020 – Present)

- **Core Characteristics:**
  - Paradigm shift triggered by **AlphaFold**.
  - Rise of **Generative AI** and **Diffusion Models**.
  - *Context:* The most drastic transformation in history. AI changed the rules of the game.
- **Leveraging Structure Prediction (2020-2021):**
  - **AlphaFold2 & RoseTTAFold:** Solved the protein structure prediction problem.
  - **"Hallucination" Design:** Inverting these networks to perform "Hallucination"—generating sequences that the network predicts as stable structures with high confidence.
- **The Leap in Sequence Design: ProteinMPNN (2022):**
  - **Function:** Given a backbone structure, it generates a folding sequence in seconds.
  - **Impact:** Orders of magnitude faster than Rosetta. Crucially, it significantly improves **solubility** and **expression rates** in wet lab experiments.
- **The Backbone Revolution: Diffusion Models (2023-2024)**
  - **Tools: RFdiffusion**, Chroma
  - **Principle:** Adapted from image generation (DALL-E, Midjourney). The model starts with noise and progressively "denoises" it to form a new protein backbone.
  - **Capabilities**: Can generate highly feasible backbones from vague prompts (e.g., "bind to this target" or "have this symmetry")

Recap: generative modeling

"What I cannot create, I do not understand." — Richard Feynman

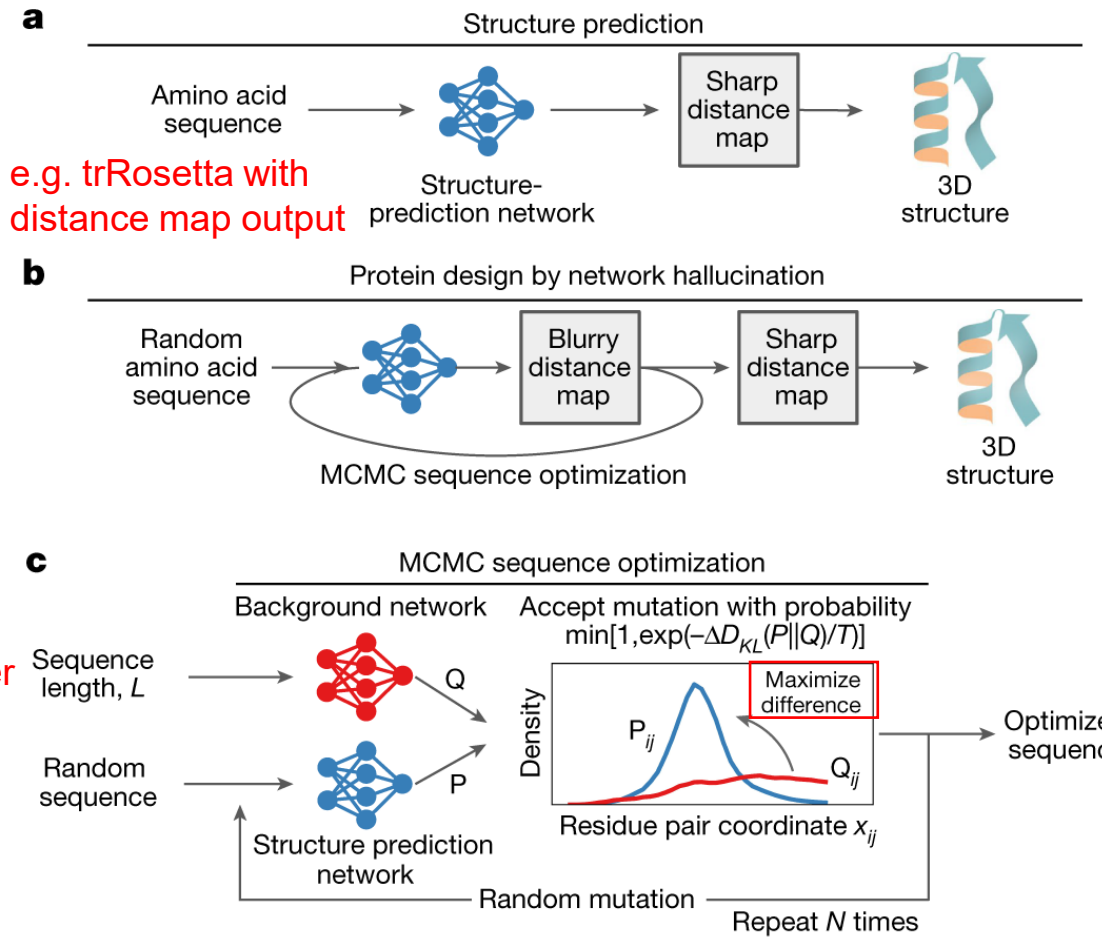
- **Generative modeling** are designed to learn the underlying distribution of data and generate new data samples that resemble the original data.

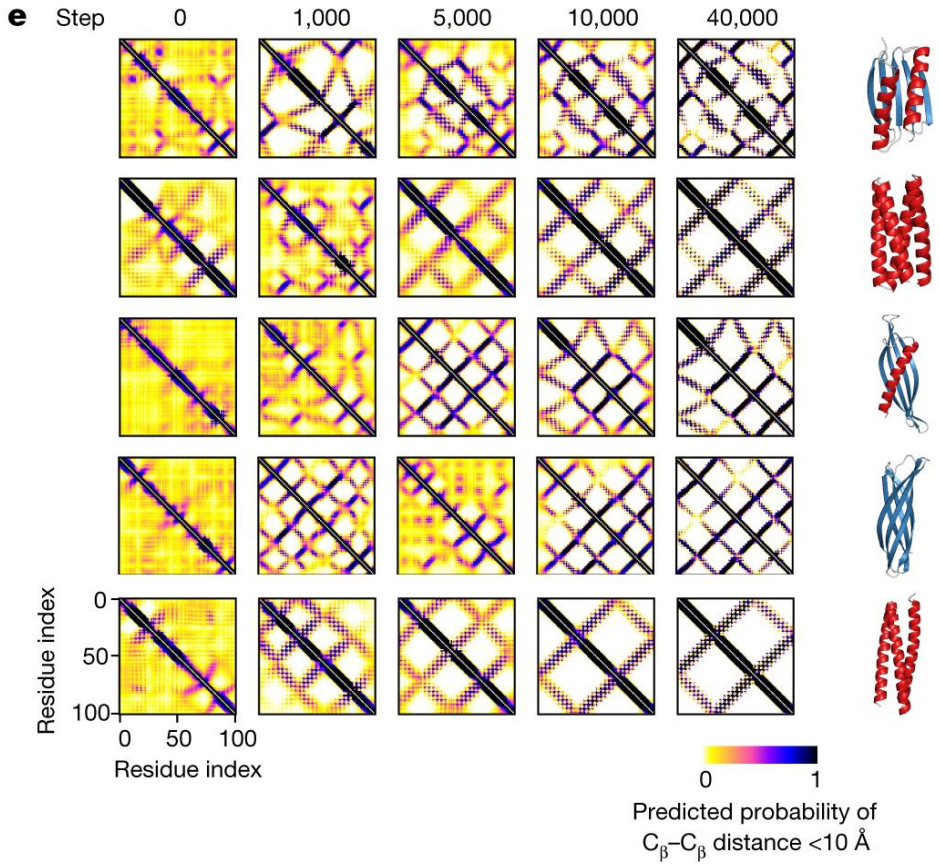


Protein hallucination



- **Hallucination**: the network "dreamed up" or "hallucinated" based on the generalized principles it learned from the data.



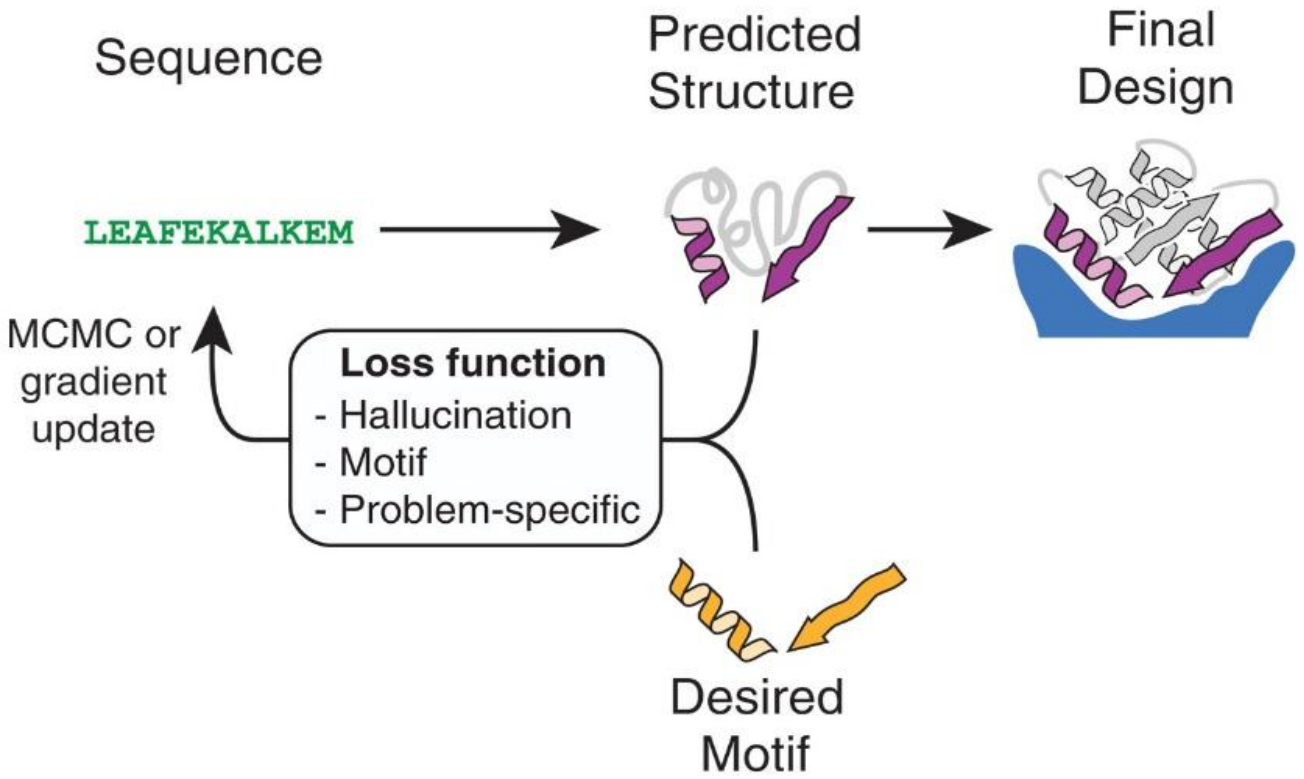


Nature, 2021

Protein hallucination with functional sites constraint



- Find sequences predicted to fold into stable structures that contain the functional site

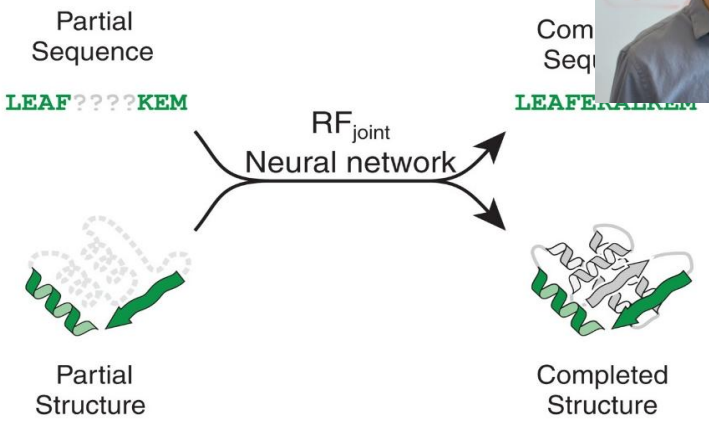


Science, 2022

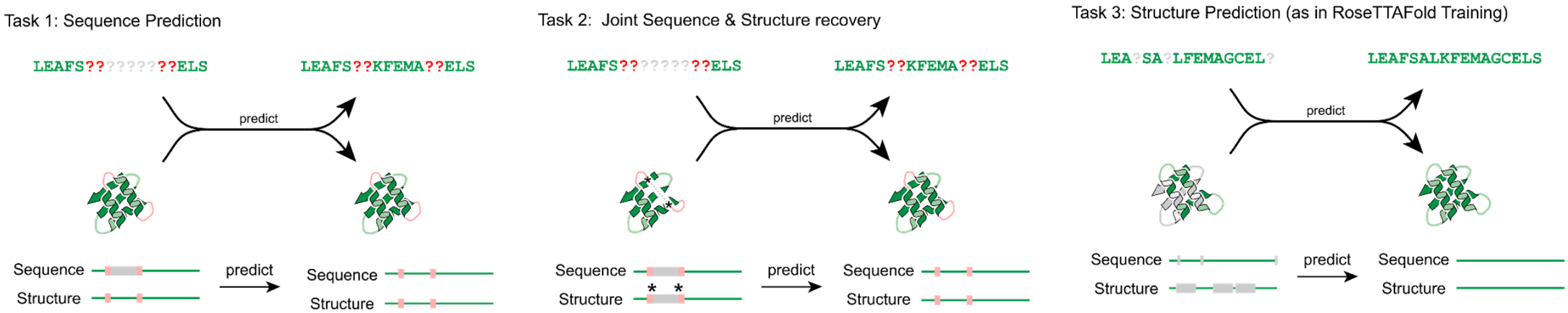
Protein inpainting



- Modify RosettaFold network, retrained a structure prediction network RF_{joint} to recover the sequence and full structure of a protein given only the functional site.

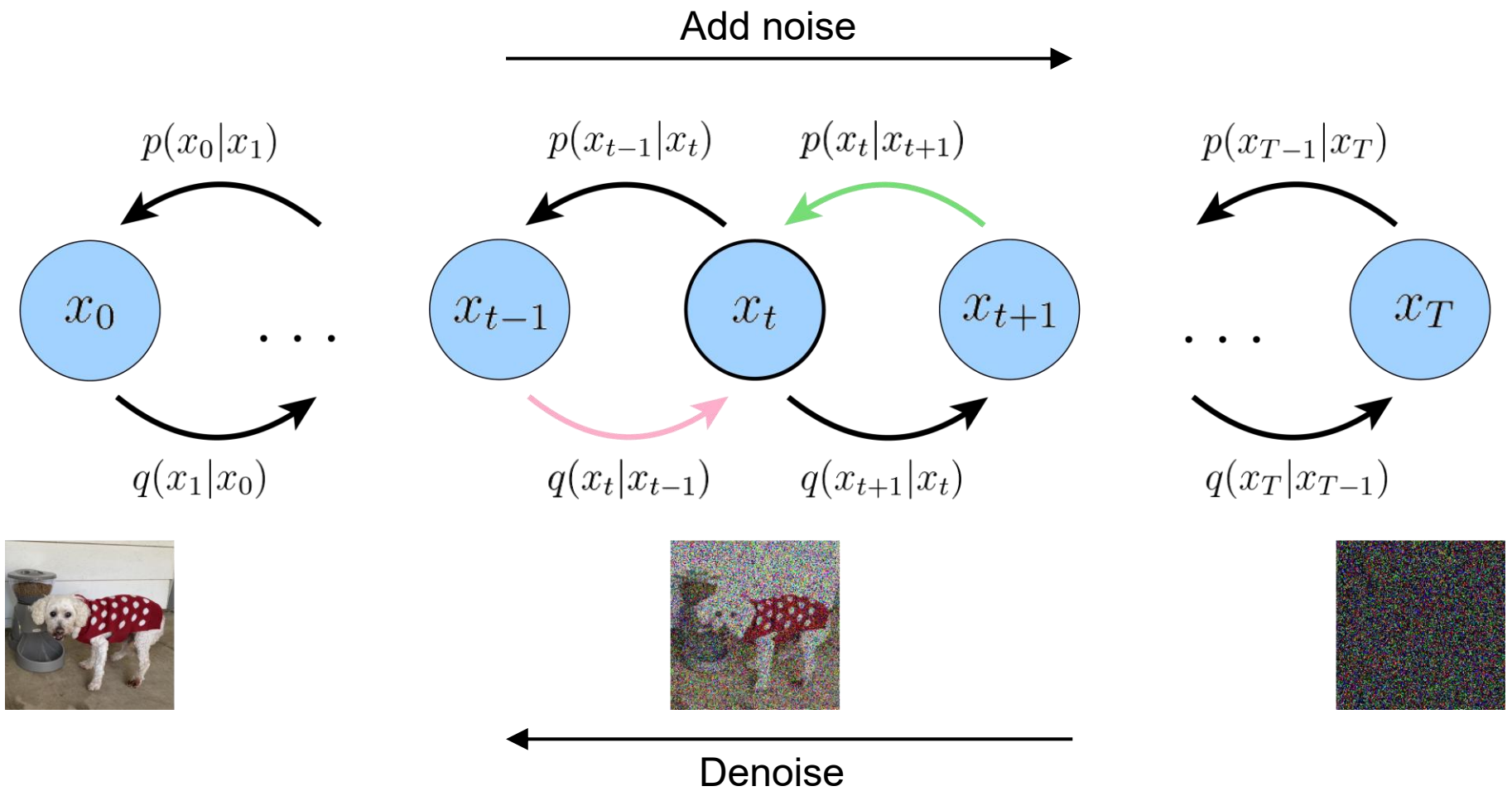


Three tasks used to train RF_{joint}



Science, 2022

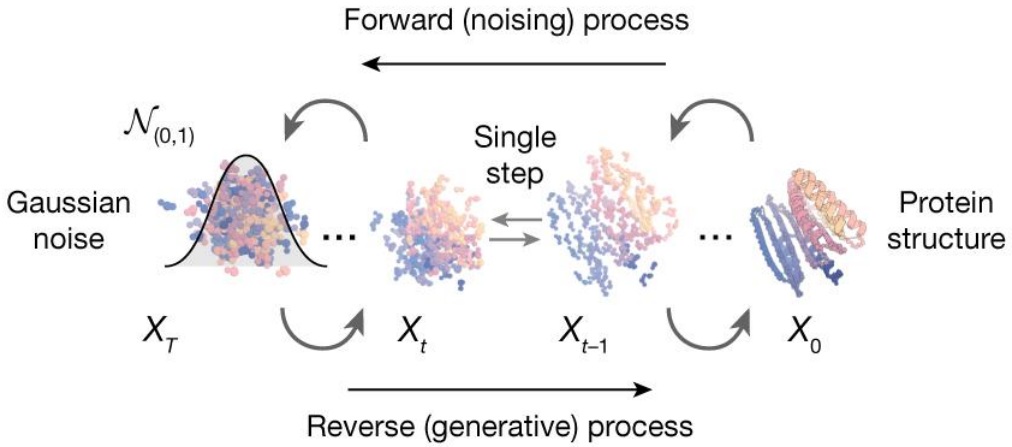
Recap: diffusion models



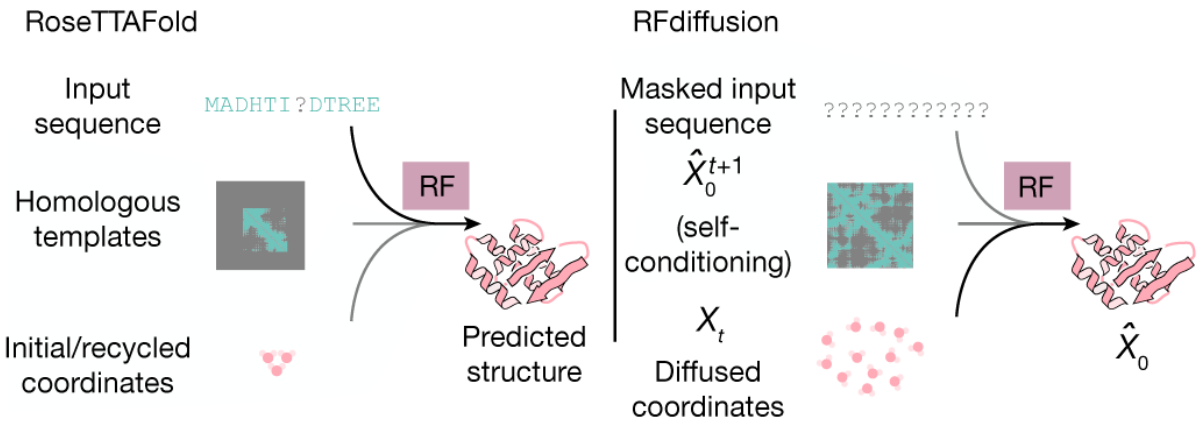
Protein design using RFdiffusion

- **Aim**: generate novel protein backbone structure

Diffusion model for protein structure modeling

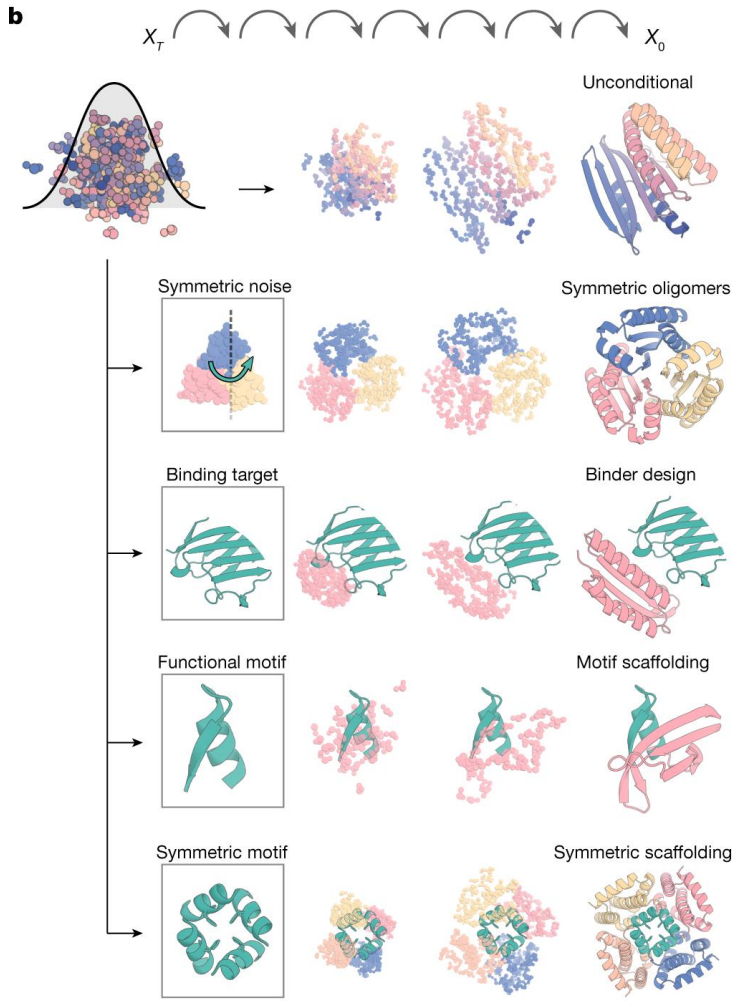


Similarity between structure prediction and structure design



Nature 2023

Rfdiffusion is broadly applicable for protein design



Unconditional design

Symmetry condition

Binding target condition

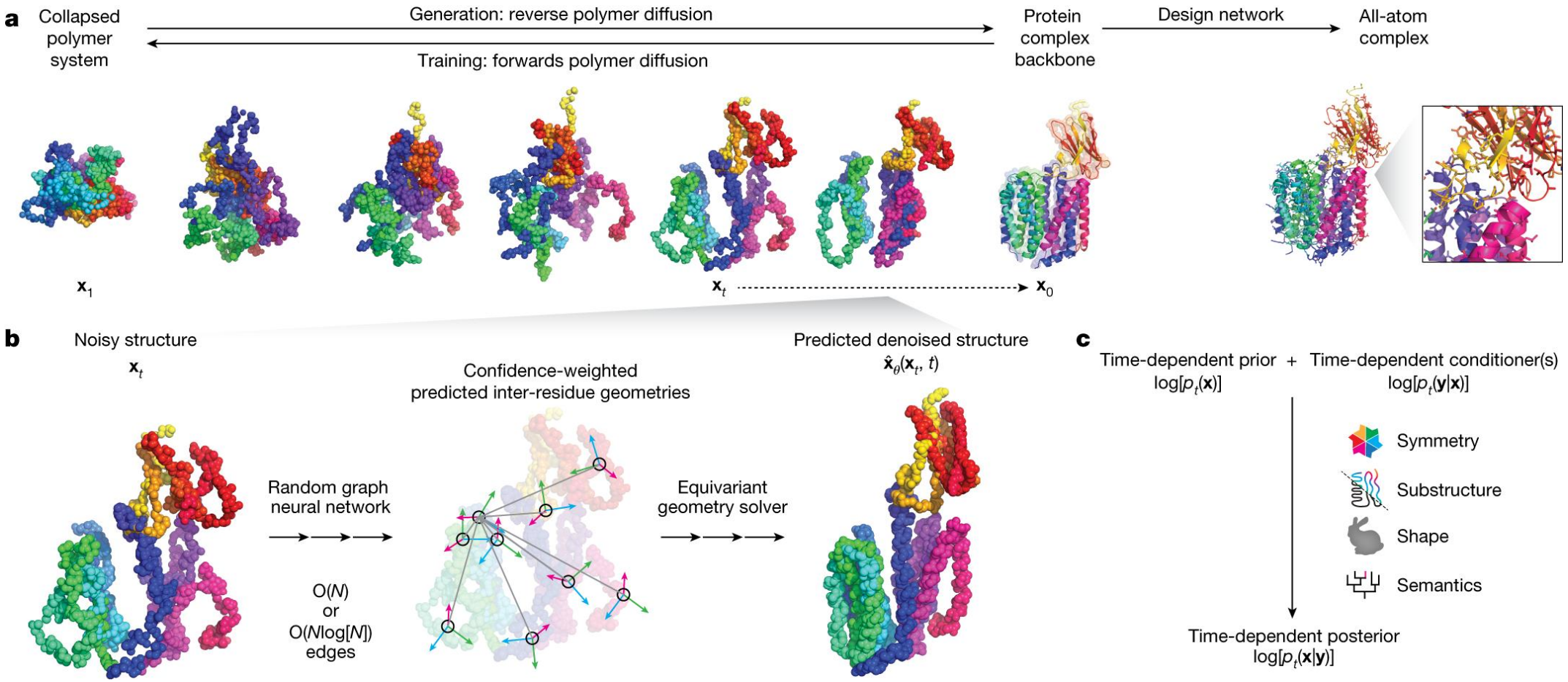
Functional motif condition

Symmetric motif condition

ProteinMPNN could be used for further sequence generation

Nature 2023

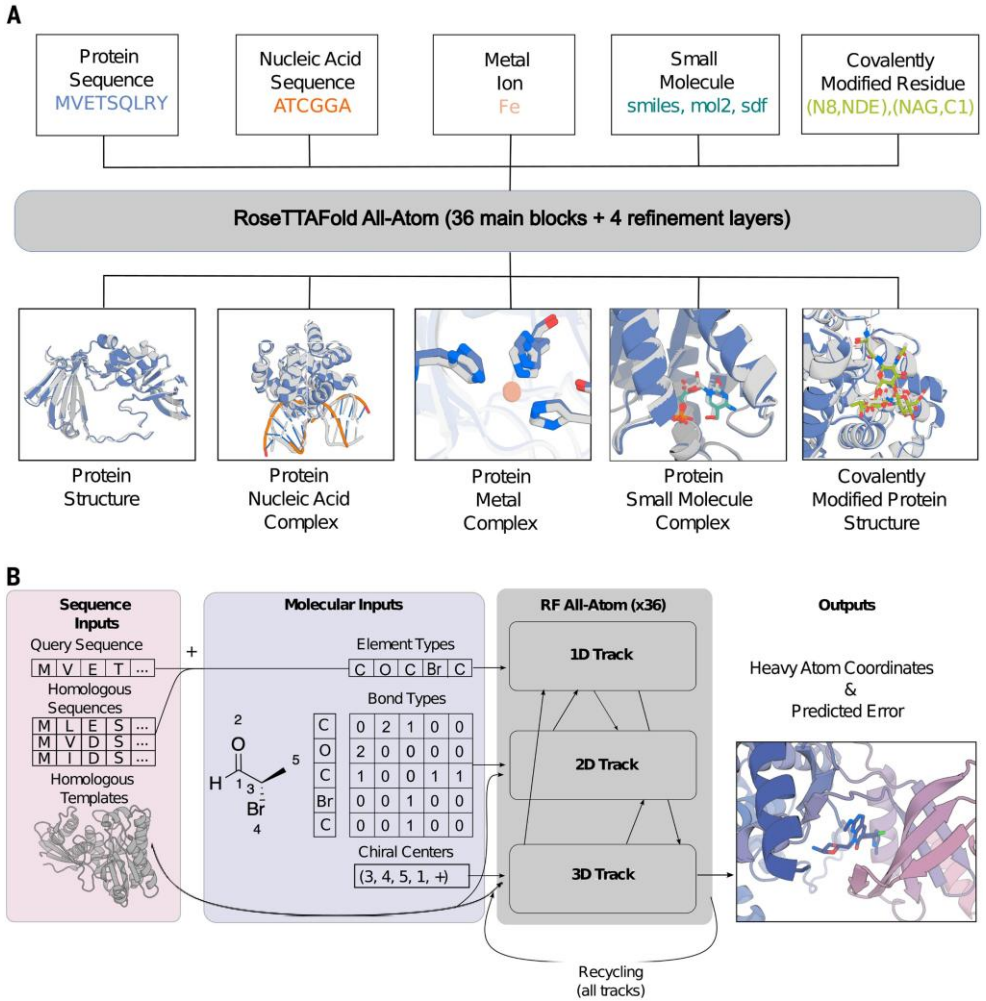
Chroma: illuminating protein space with a programmable generative model



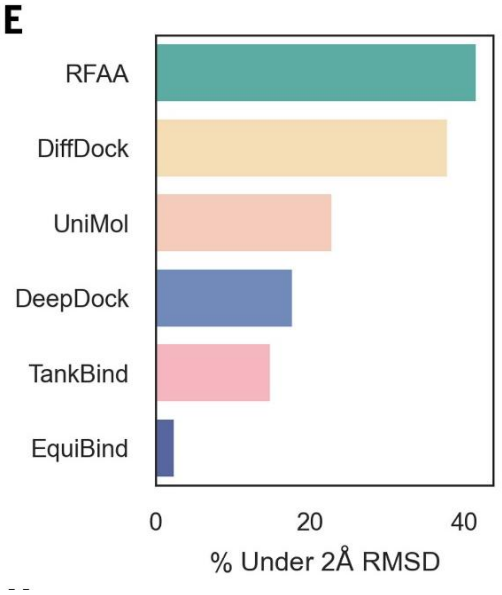
Nature, 2023

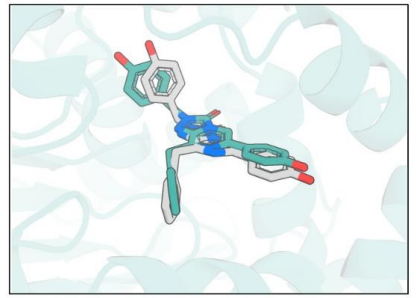
Generalized biomolecular modeling and design with RFAA

Unified structure prediction architecture

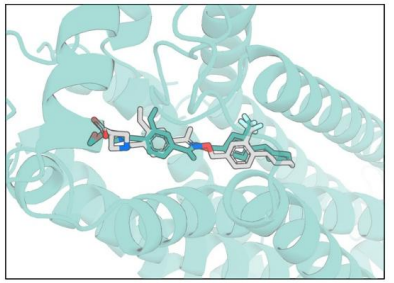


Protein-small molecular structure prediction





Closest Protein Seq in Training: 23%
Closest Ligand In Training: 0.46
Ligand RMSD: 1.20

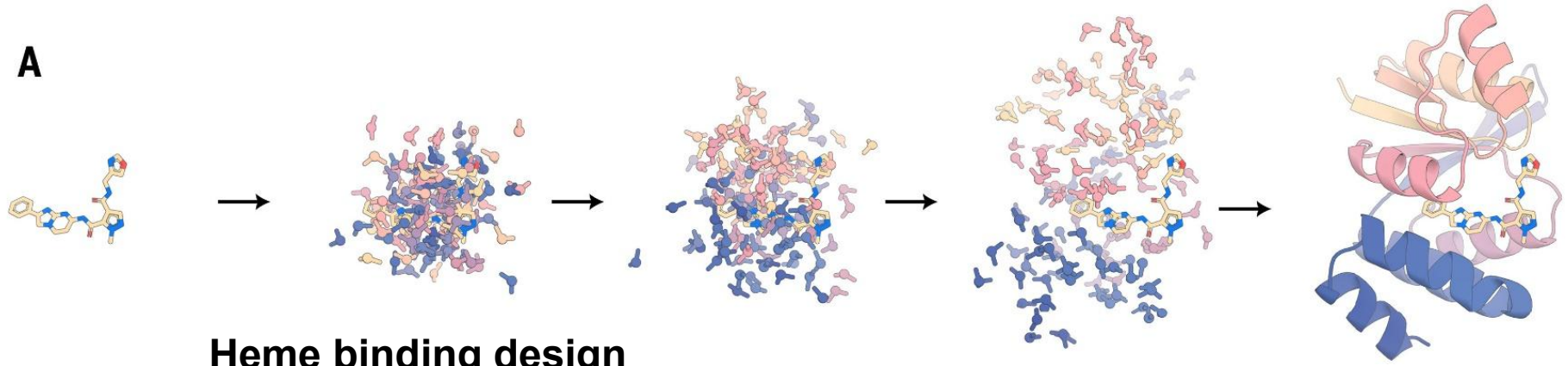


Closest Protein Seq in Training: 31%
Closest Ligand In Training: 0.48
Ligand RMSD: 1.31

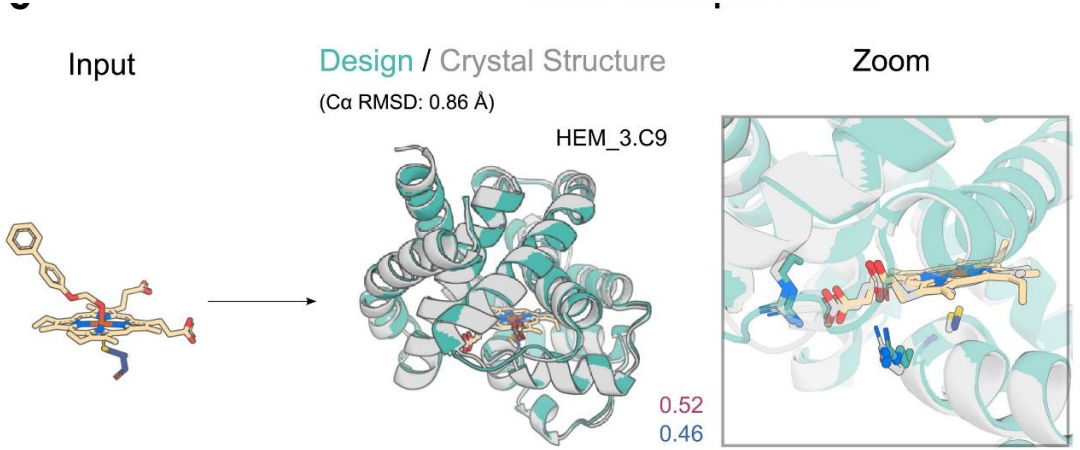
Science 2024

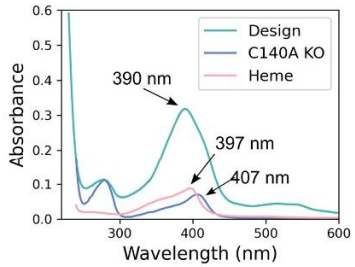
Generalized biomolecular modeling and design with RFAA

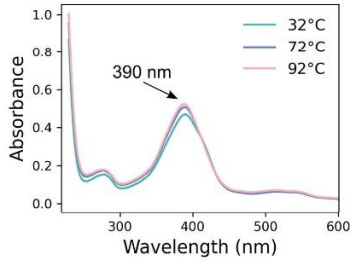
De novo small-molecule binder design



Heme binding design







The designed protein matches the expected spectra for penta-coordinated heme and that mutating cysteine to alanine abolishes binding

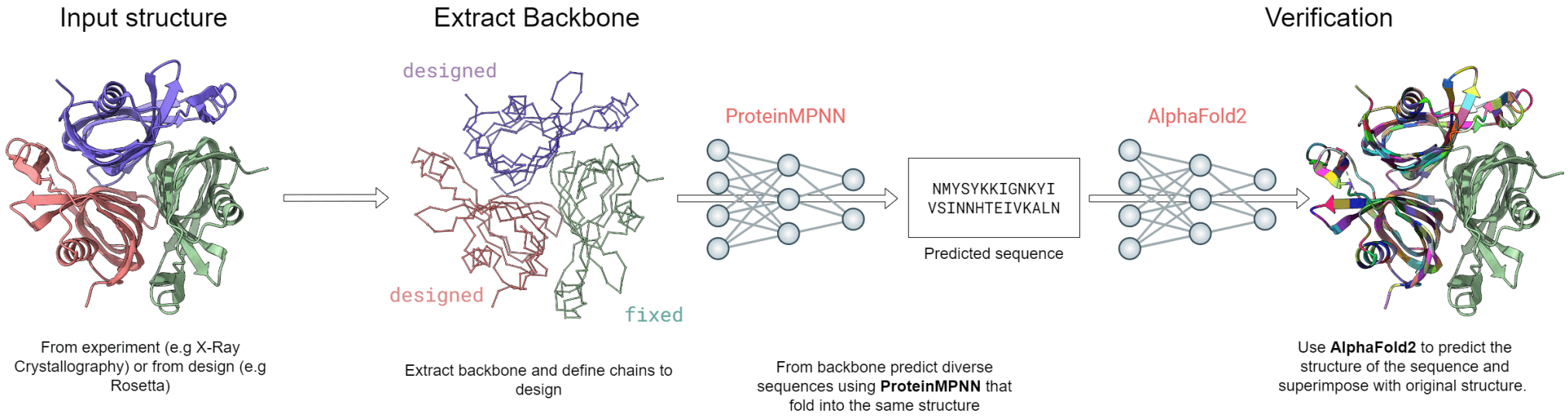
The designed protein retains heme binding at temperatures up to 90°C

Protein inverse folding

- **Problem definition**: given a target protein backbone structure, it designs amino acid sequences that are predicted to fold into that structure, prioritizing stability and desired functions.
- **Application**: redesign proteins with improved
  - Protein expression
  - Stability
  - Function
- **Possible workflow**:

Category	Key features	Limitation
Physics-based optimization	Uses simulations to find sequences that minimize a physics-like energy score	Very slow and limited by imperfect energy models
Knowledge-based statistical methods	Uses statistical patterns from known protein structures to pick compatible sequences	Copies existing patterns; not good for creating novel designs
Evolution-based methods	Designs sequences that look like natural proteins by using evolutionary family data	Requires known protein relatives; fails for completely new folds
Hybrid/expert-guided Methods	Combines tools and manual expert input to refine designs	Not scalable; slow and dependent on human expertise

ProteinMPNN for protein inverse folding

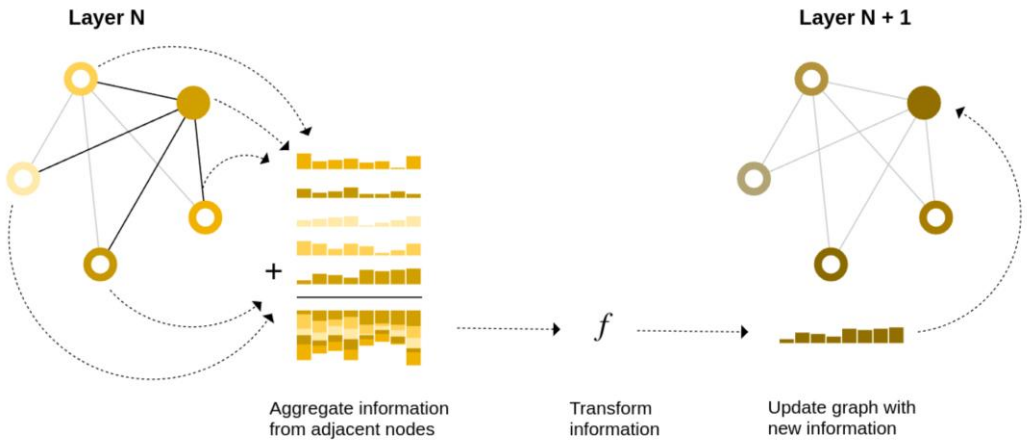


ProteinMPNN workflow

- **MPNN** (message passing network):

For each node (filled circle):

1. Gather neighboring node embeddings (messages).
2. Learn a function to update node embeddings from the gathered messages by aggregating and refining them.



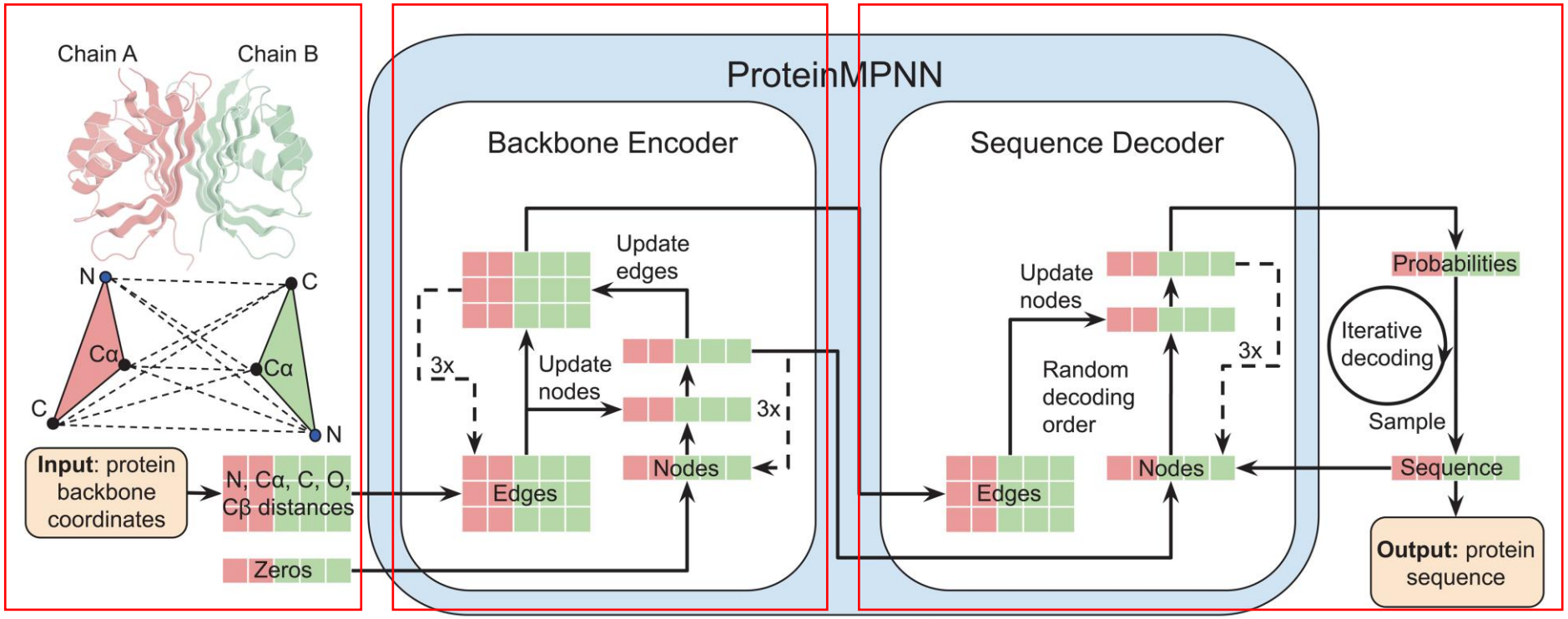
ProteinMPNN for protein inverse folding

- ProteinMPNN replacing slow physics-based simulations with a fast, robust deep learning model that directly predicts optimal sequences for any given backbone structure.

1. Input preparation: calculate geometry features from backbone atoms

2. Update node and edge features using a message-passing neural network

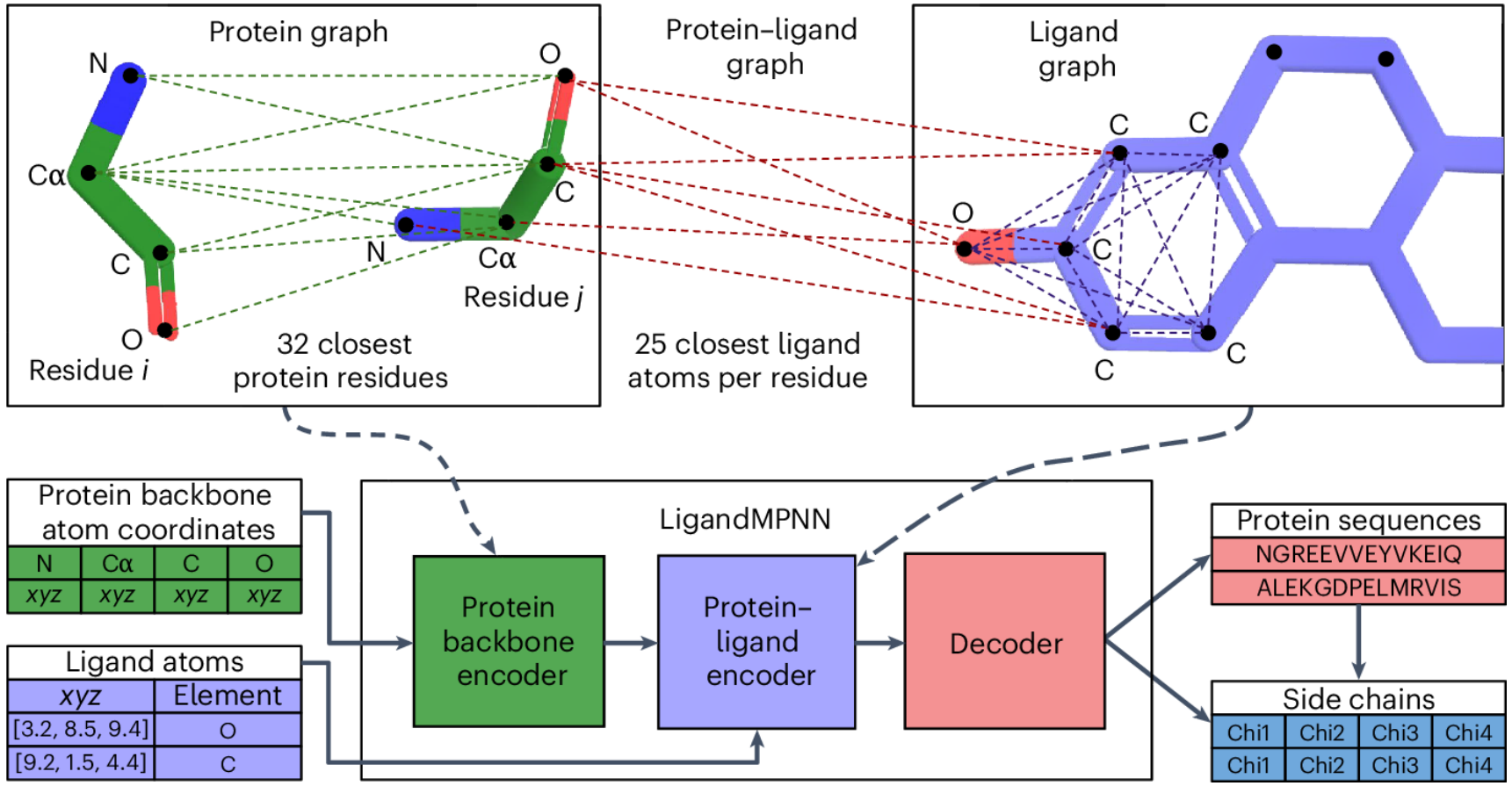
3. Iteratively decoding to generate protein sequence



Science, 2022

LigandMPNN incorporates ligand context for protein inverse folding

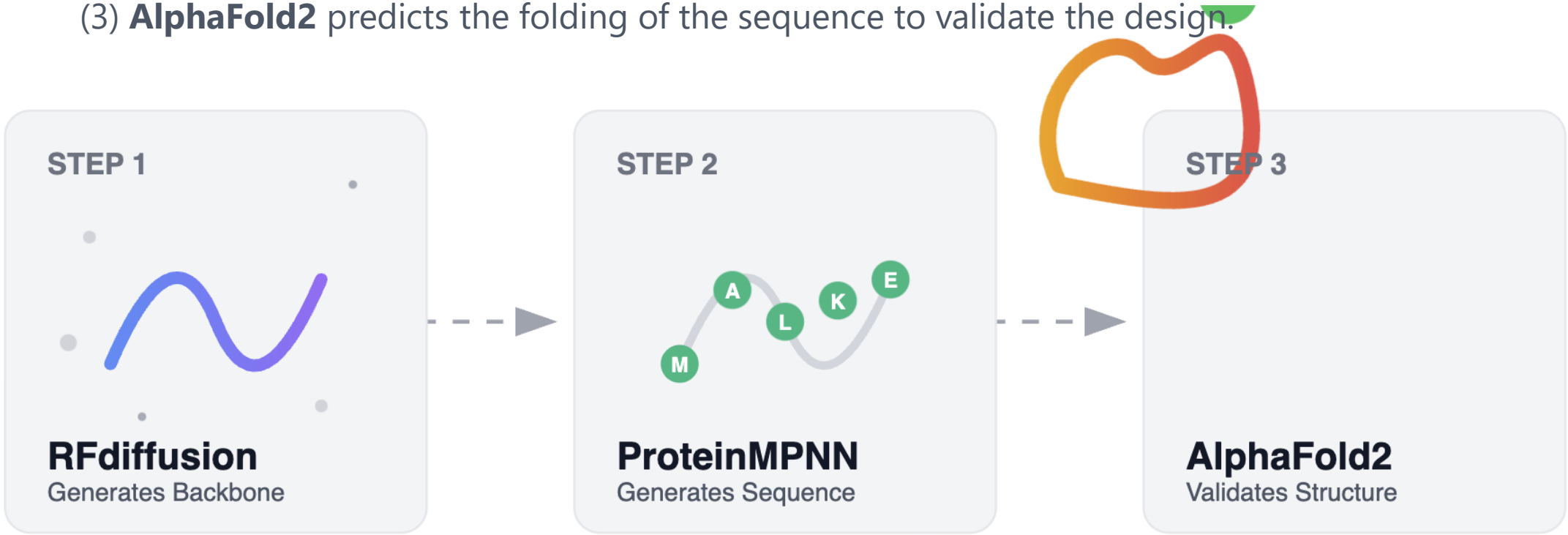
- Feature: introduces a protein-ligand encoder to update geometry features embedded in protein-ligand graph



Nat Methods, 2025

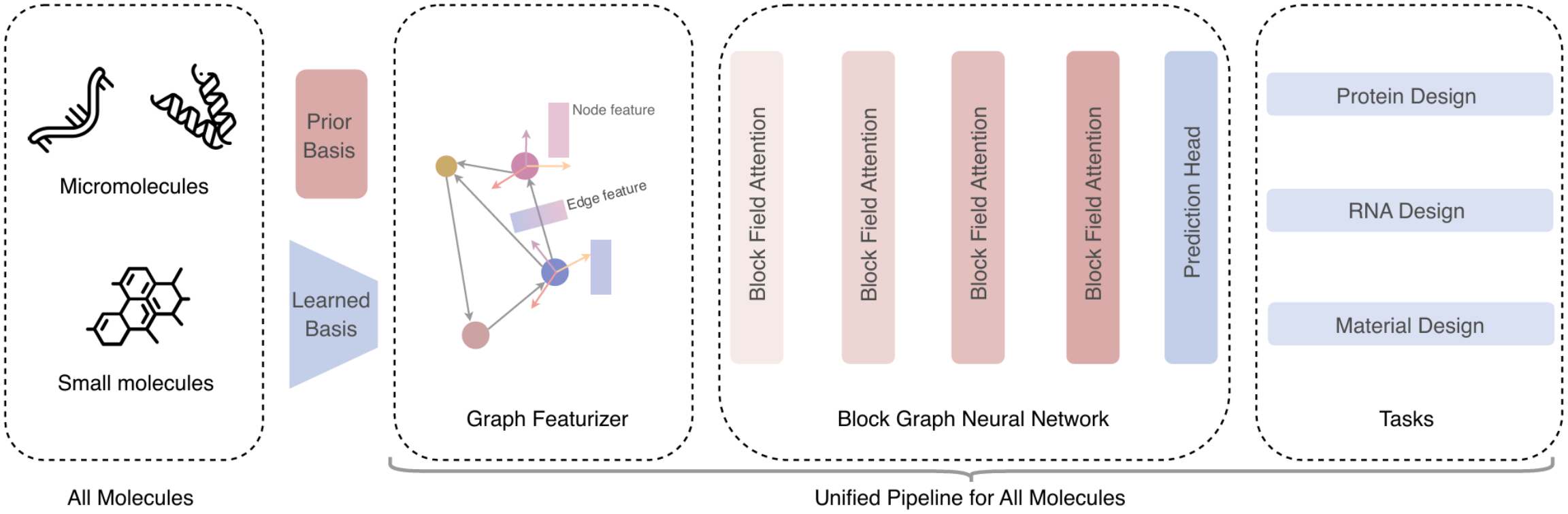
The modern workflow

- **The Generative Protein Design Pipeline.**
  (1) **RFdiffusion** generates a novel backbone structure from noise.
  (2) **ProteinMPNN** designs a sequence that fits the backbone.
  (3) **AlphaFold2** predicts the folding of the sequence to validate the design.



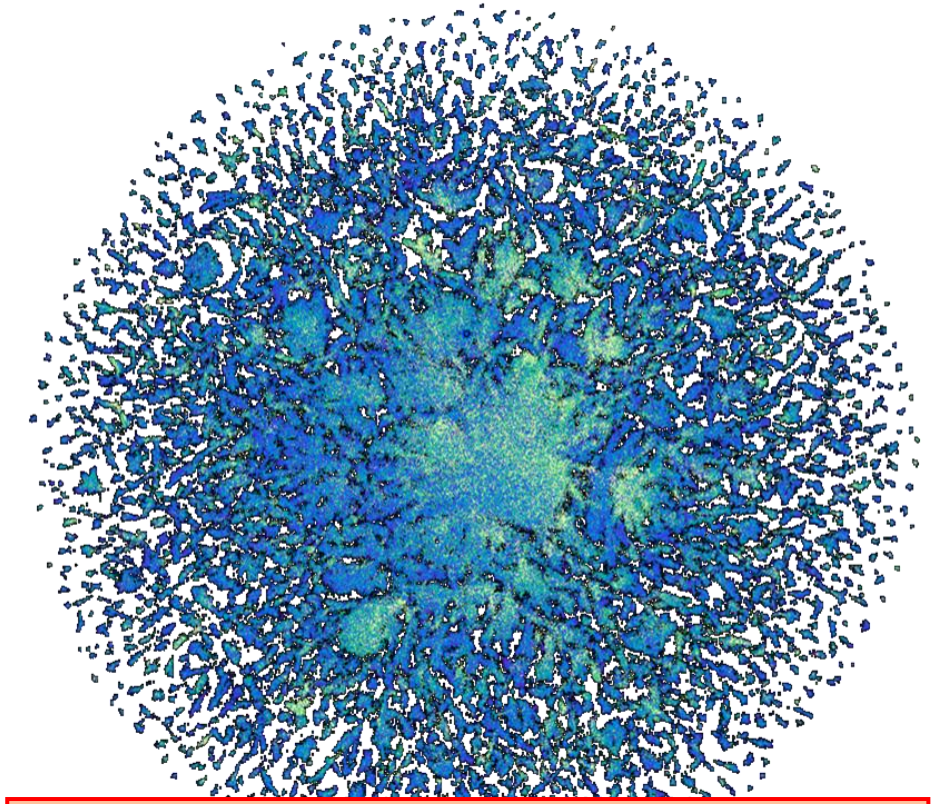
Unified molecule inverse folding

- Extend to all biomolecules including protein, RNA and small molecules



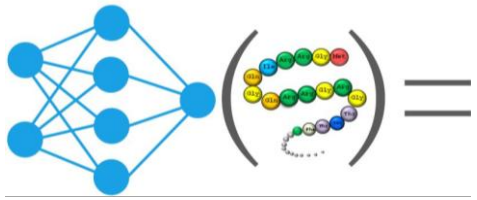
arXiv, 2024

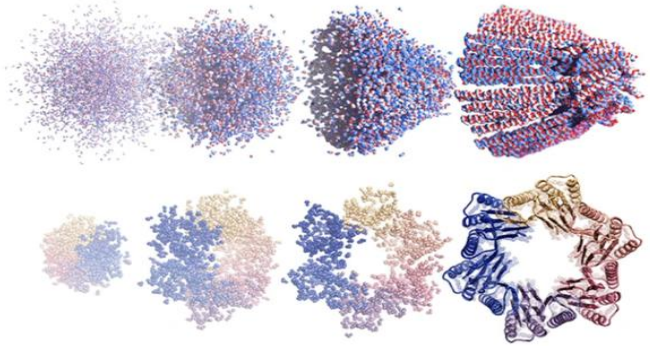
Non-structure-based protein design



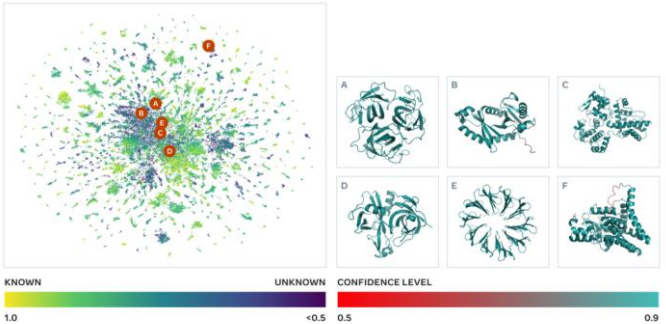
Protein sequence space

Data-driven / Semantic-based Methods

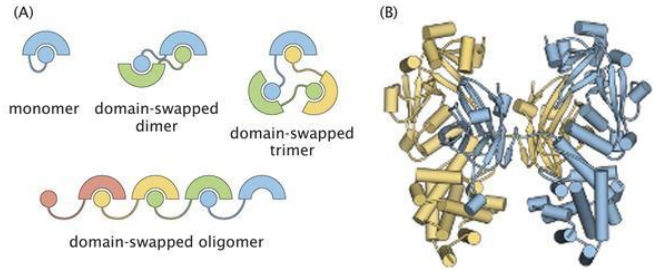




Protein design

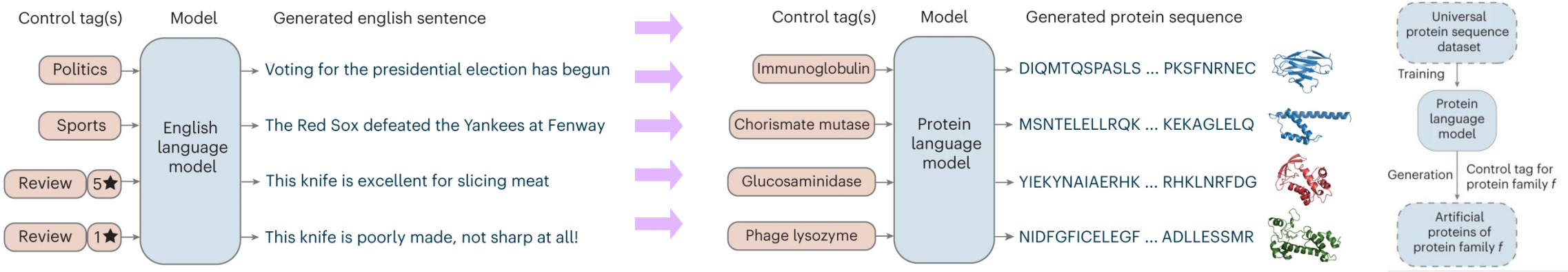


Protein Evolution

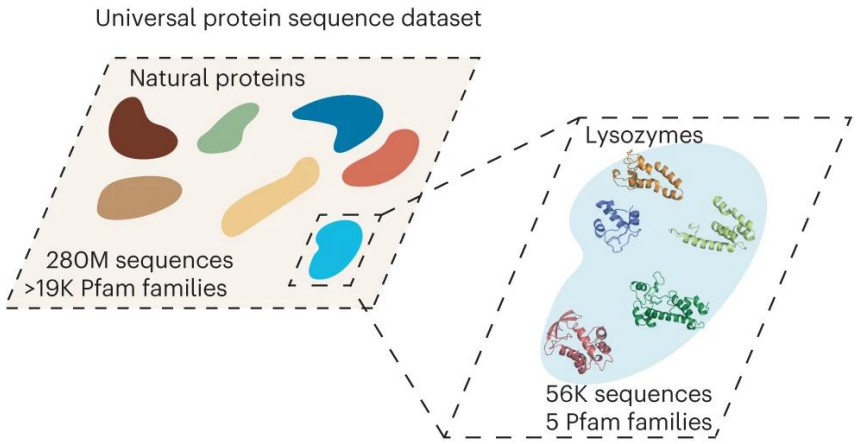


Function based generation

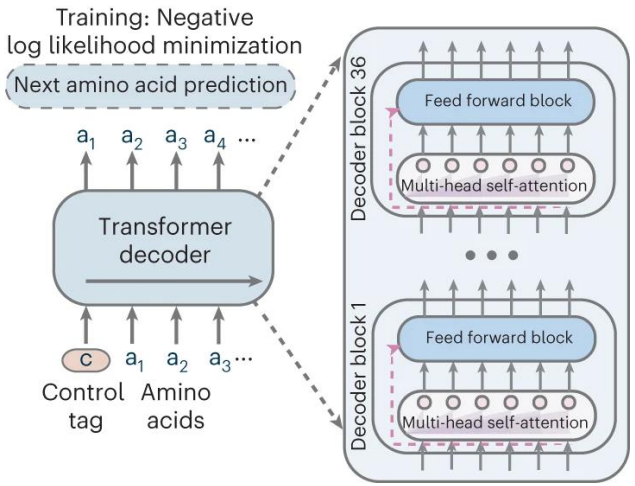
ProGen: protein language model trained on Pfam database



Datasets:



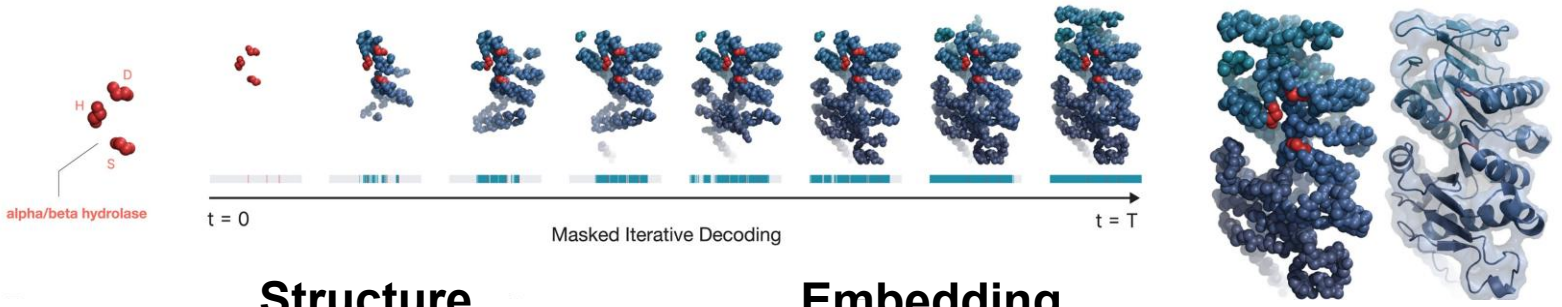
Model architecture:



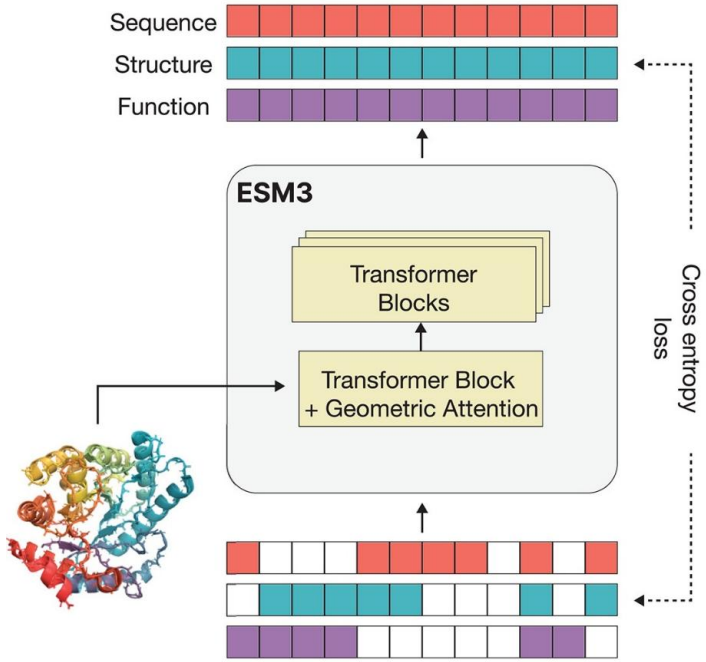
(*Nature Biotechnology*, 2023)

ESM3: simulating evolution with a protein LLM

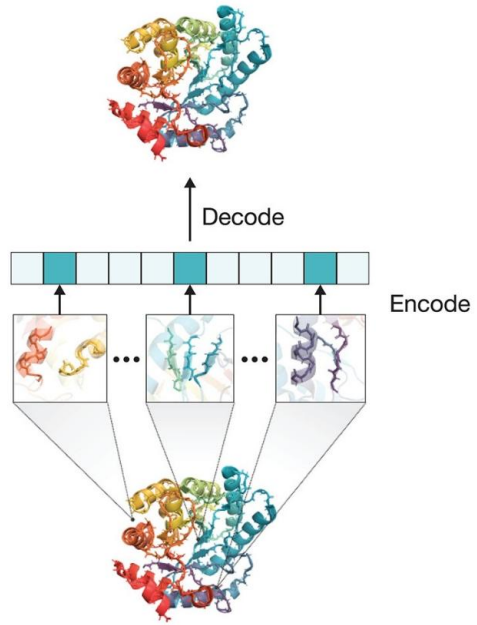
Inference:



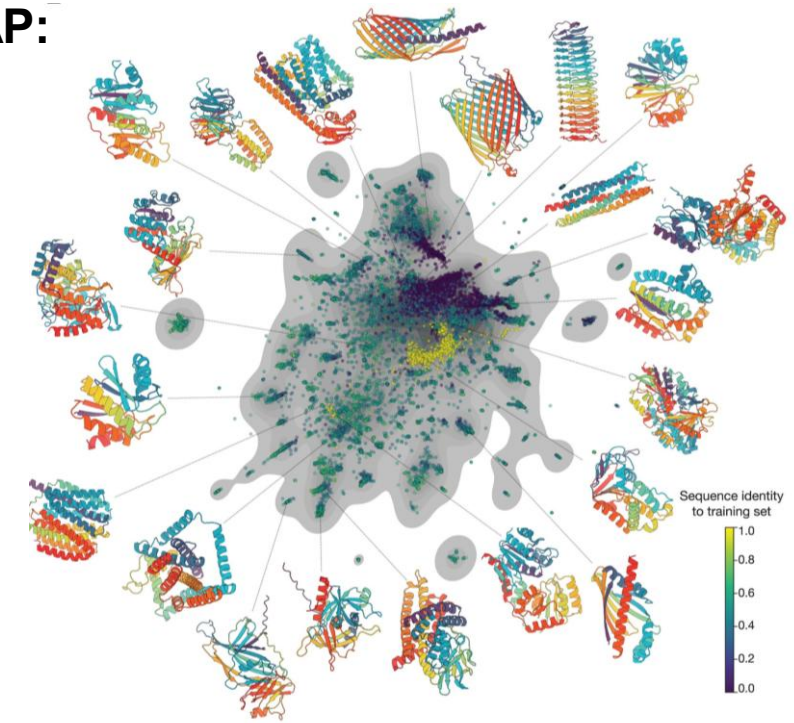
Training:



Structure tokenization:

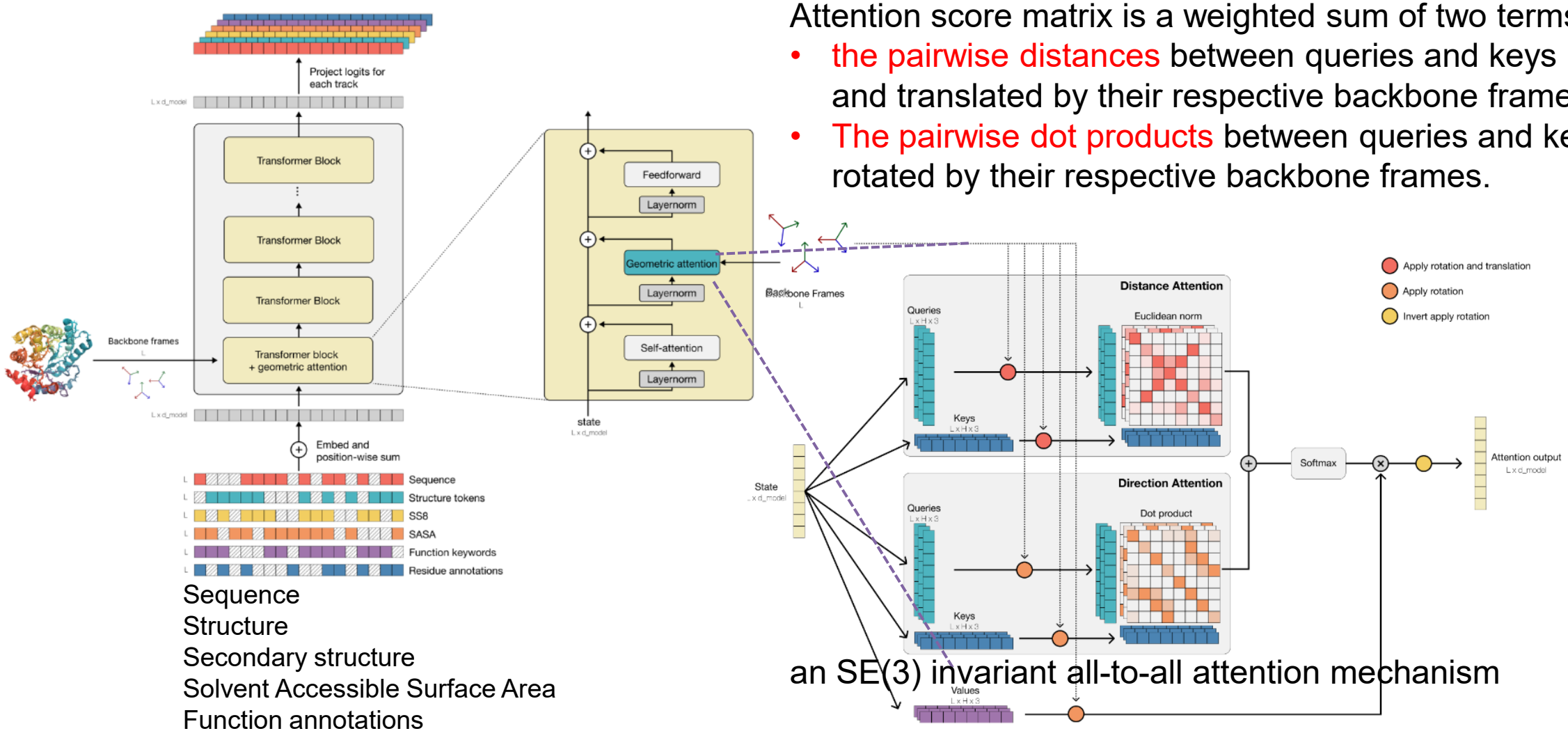


Embedding UMAP:



(*Science*, 2025)

ESM3 architecture



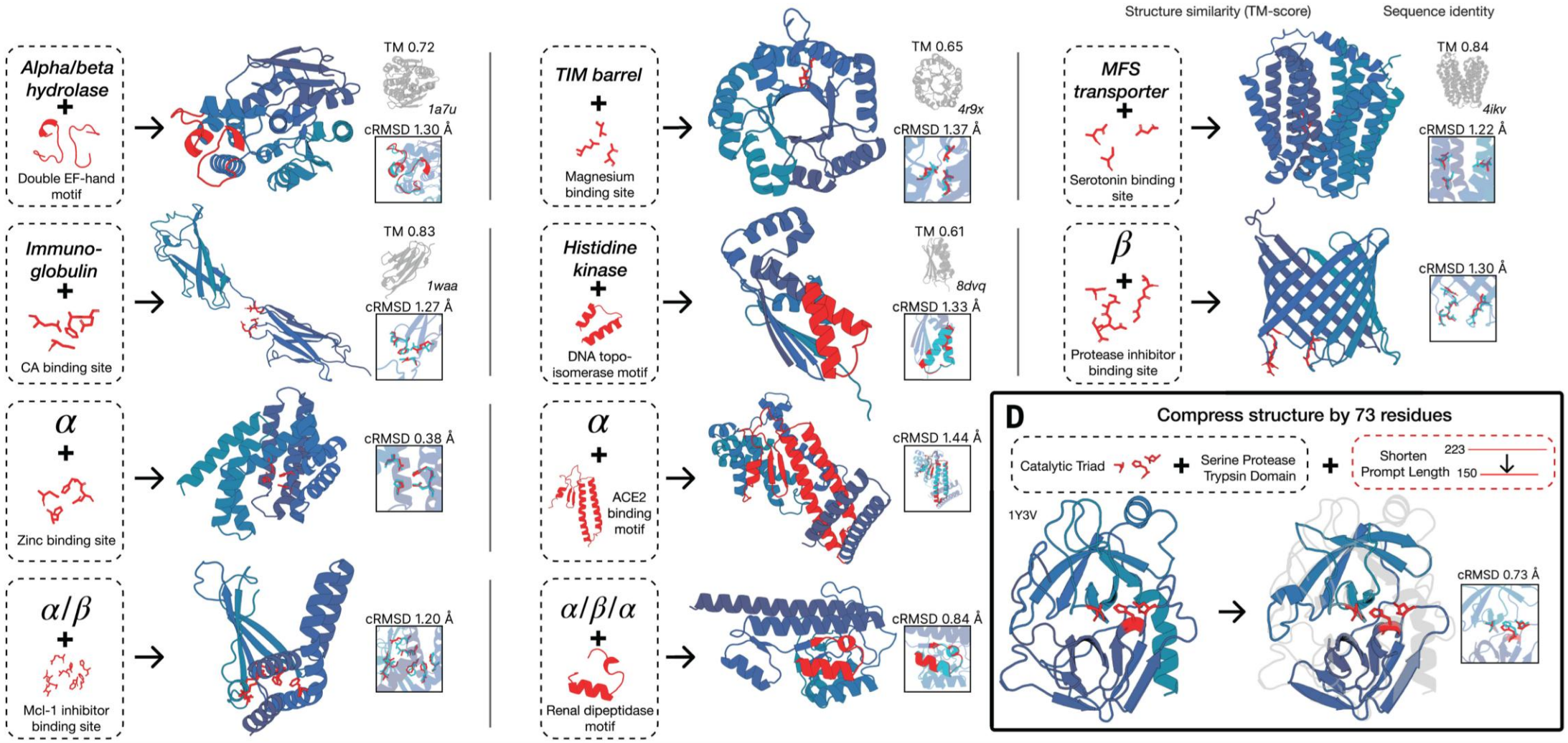
Attention score matrix is a weighted sum of two terms:

- the pairwise distances between queries and keys rotated and translated by their respective backbone frames
- The pairwise dot products between queries and keys rotated by their respective backbone frames.

an SE(3) invariant all-to-all attention mechanism

Sequence
Structure
Secondary structure
Solvent Accessible Surface Area
Function annotations
Residue annotations

ESM3 Designs whole proteins with prompt of physicochemical property or atomic-level motifs

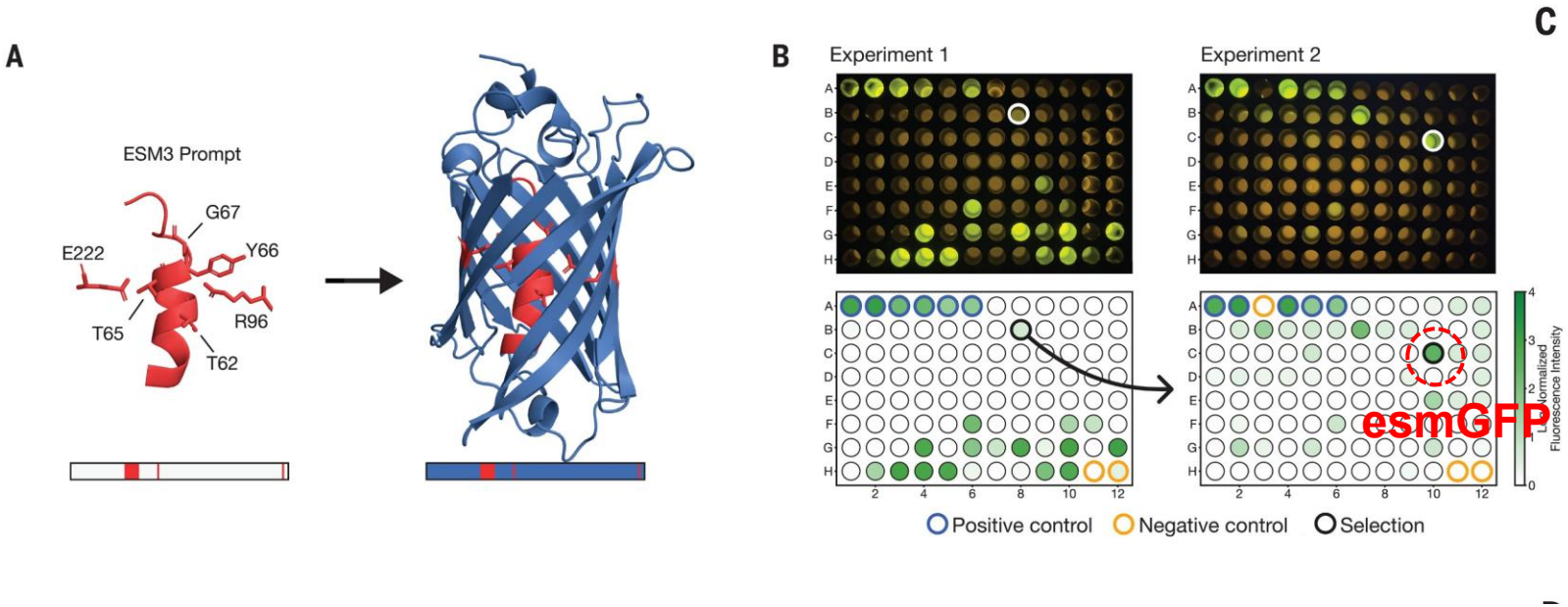


ESM3 Generates a functional fluorescent protein

ESM3 generate a fluorescent protein condition on residues required for forming and catalyzing the chromophore reaction

The fluorescent protein selected from 88 designed proteins is functional

esmGFP exhibits fluorescence intensity similar to common GFPs.



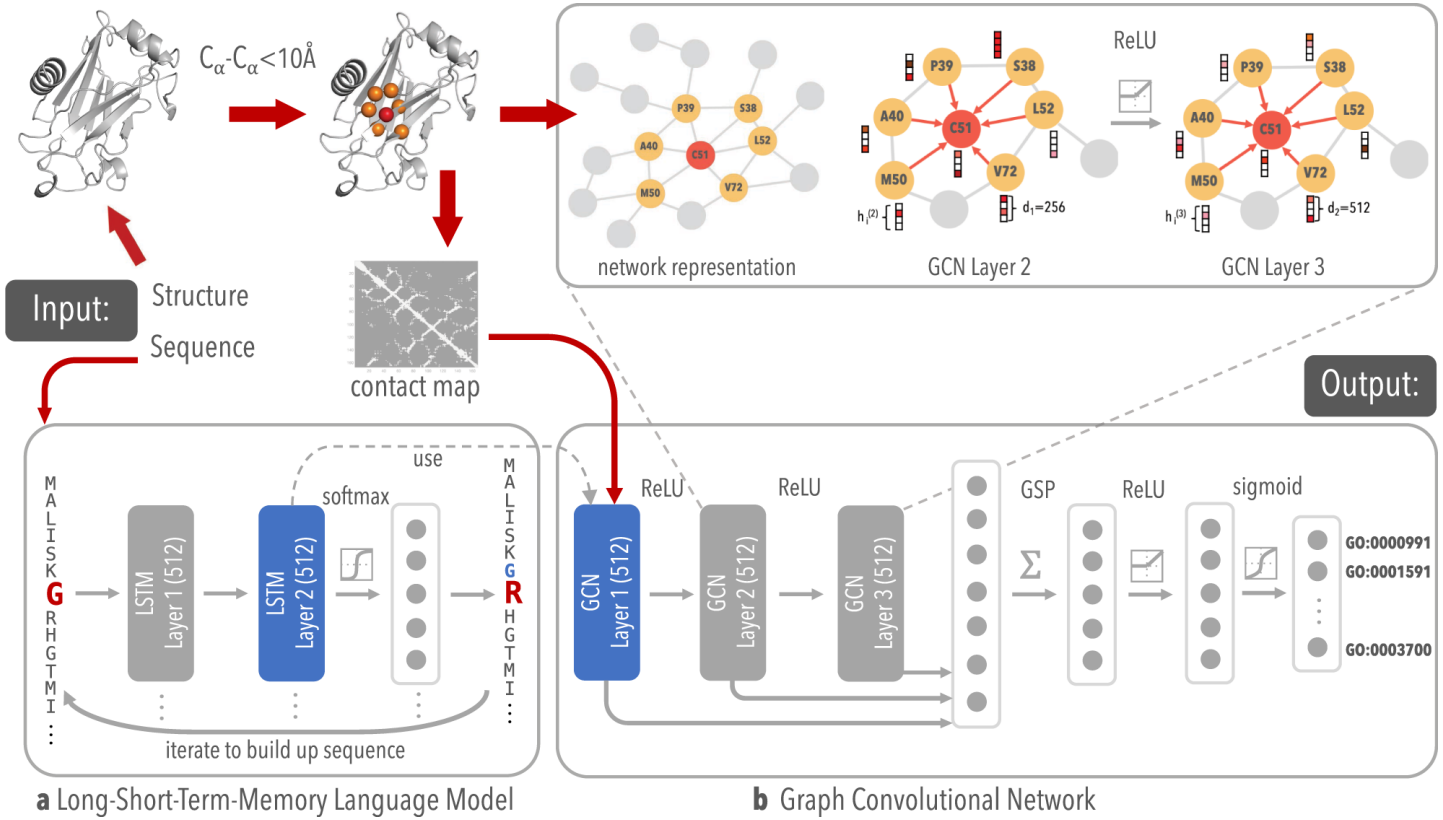
What do we do once we have a structure?

AI structural biology studies beyond structure prediction:

- **Structure-based sequence design**: generating novel protein sequences that are predicted to fold into a specific target structure and exhibit desired properties.
- **Structure-based function prediction**: using protein 3D structure to computationally infer biological function.
- **Molecular simulation**: simulate atomic-level motions and interactions of biomolecules over time.
- **Molecular docking**: computational prediction of optimal binding modes between two molecules, such as a drug and its protein target.
- **AIDD** (AI-aided drug design): applying artificial intelligence to accelerate and optimize various stages of drug discovery, from target identification to molecule optimization.

Structure-based function prediction

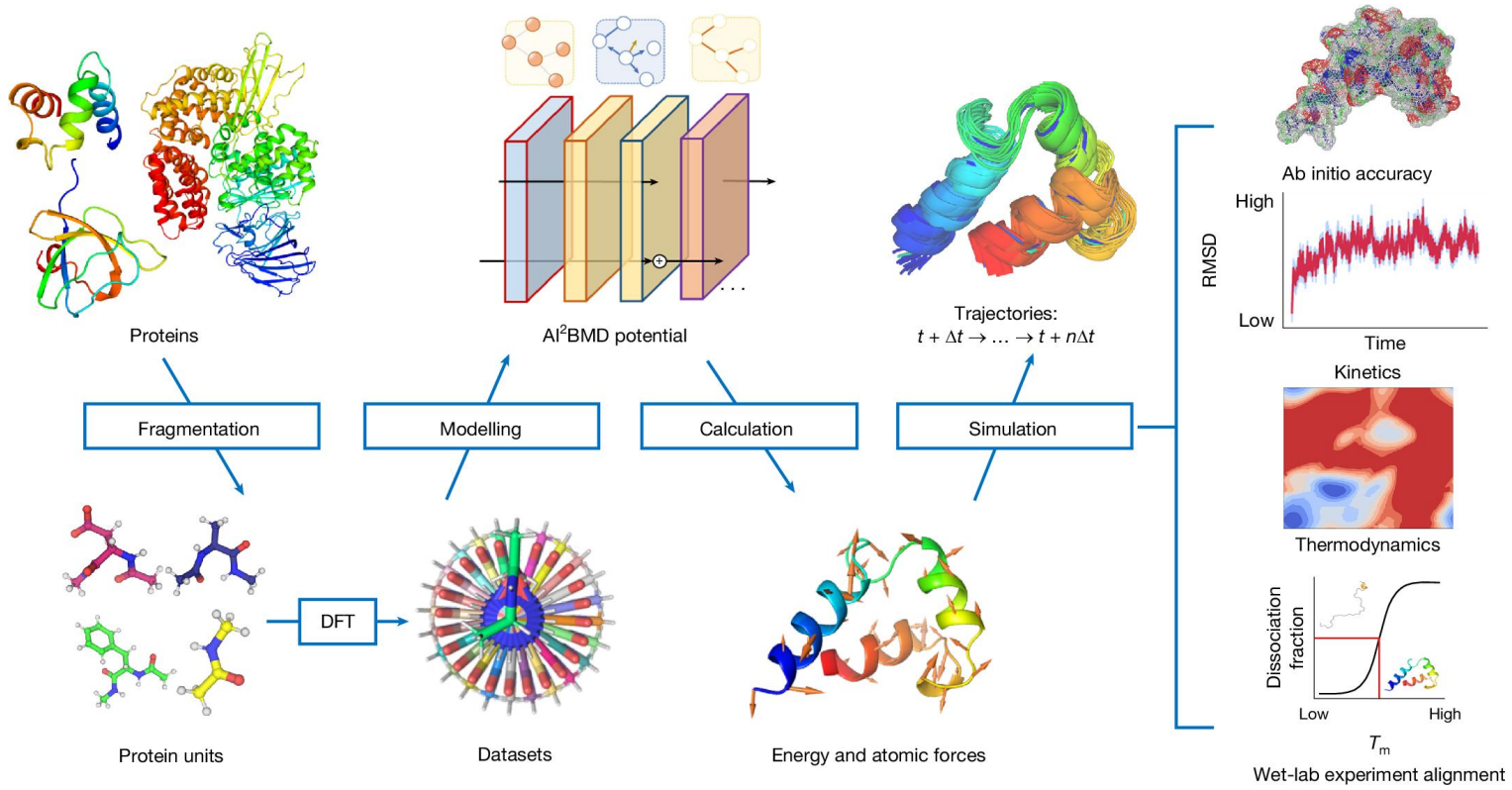
- **Core principle:** the 3D structure of a protein, especially its active sites, binding pockets, and surface features, provides direct clues about its molecular function (e.g., catalysis, ligand binding, protein-protein interactions), which sequence alone may not fully reveal.
- **Example**: DeepFRI take protein structures (and sequences) as input and directly predict GO terms or EC numbers by learning from the Protein Data Bank.



Nat Commun, 2021

AI²BMD: AI for molecular dynamics simulation

- Classical molecular dynamics simulation is fast but lacks chemical accuracy. Quantum chemistry methods such as density functional theory (DFT) can reach chemical accuracy but cannot scale to support large biomolecules.
- AI²BMD uses a protein fragmentation scheme and a **machine learning force field** to achieve generalizable ab initio accuracy for proteins comprising more than 10,000 atoms.



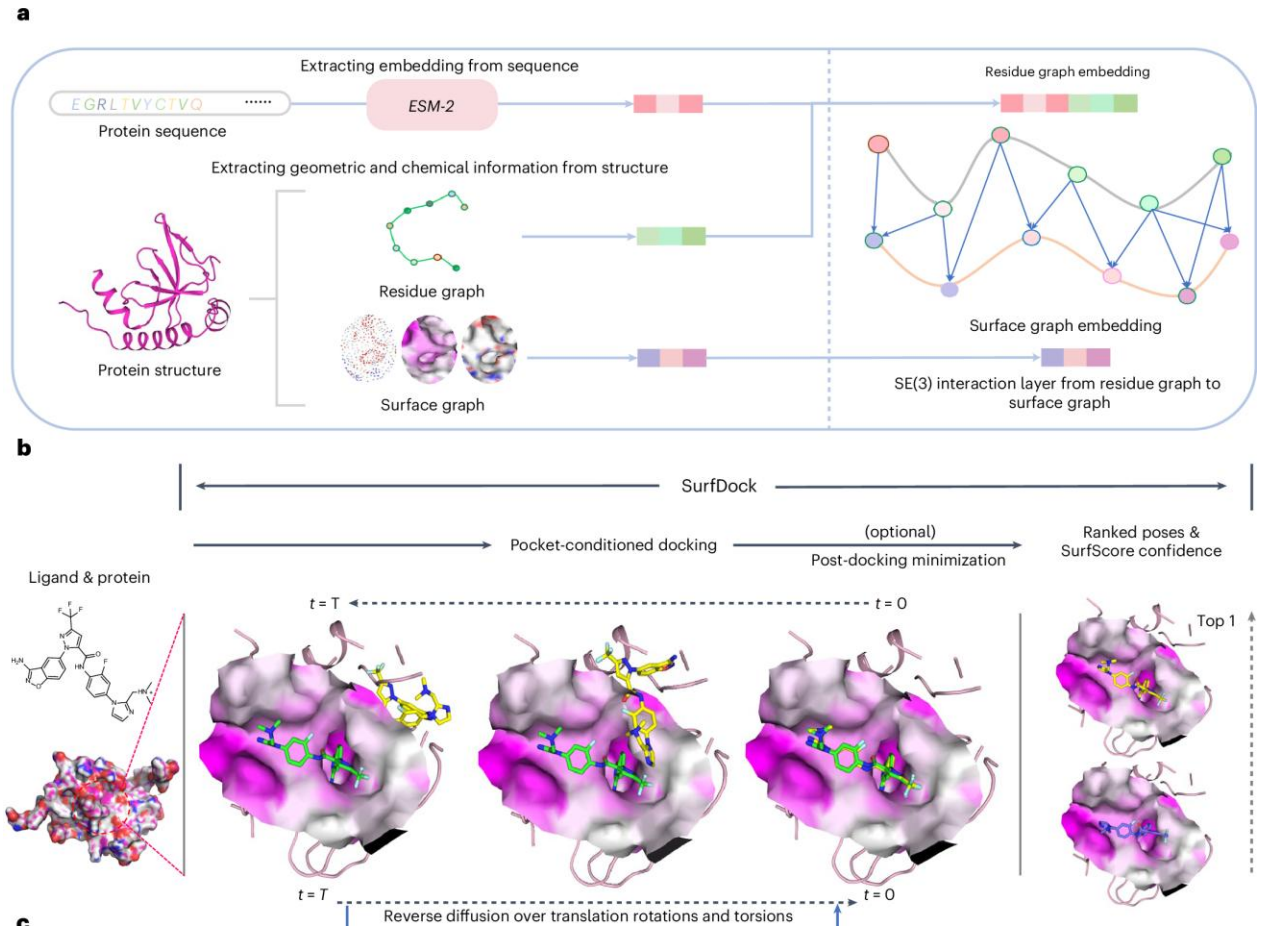
Nature, 2024

AI for ligand docking

- Strategy 1: complex structure prediction
- Strategy 2: tailored model (e.g. SurfDock, DiffDock)

Map embeddings from protein sequence and residue graph onto surface graph

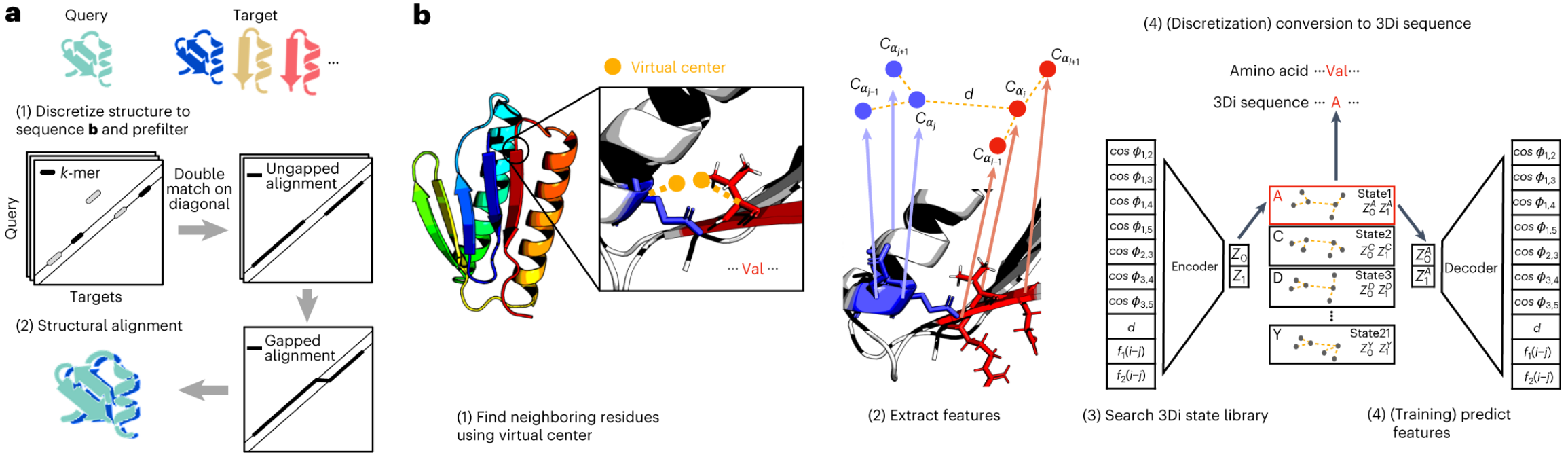
Geometric diffusion network for ligand pose refinement



Nat Methods, 2024

Foldseek: fast and accurate protein structure search

- Structure is more conserved than sequence.
- Sequence search is four to five orders of magnitude faster than structure search using TM-align.
- **Key idea**: discretize structure to sequence using VQ-VAE.



Nat Biotech, 2024

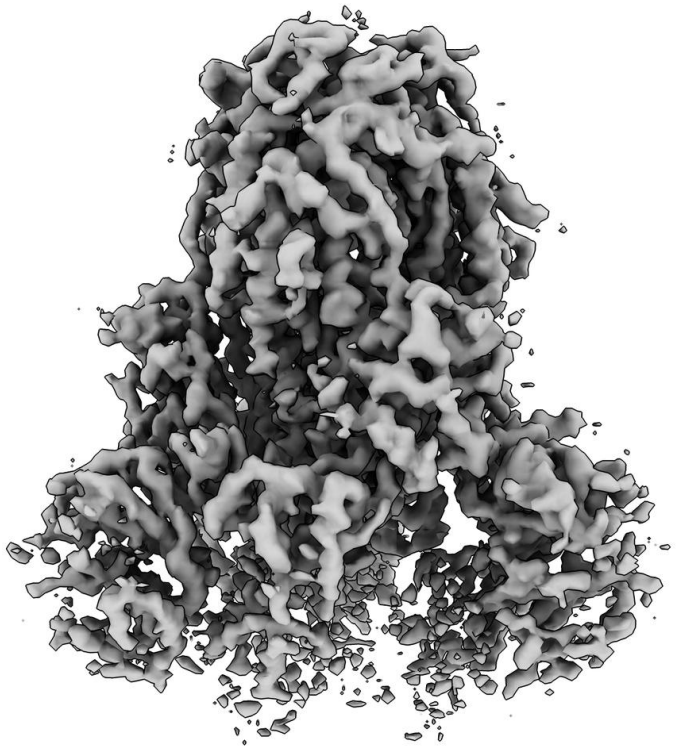
Combining AI with experimental data

Experimental data could provide:

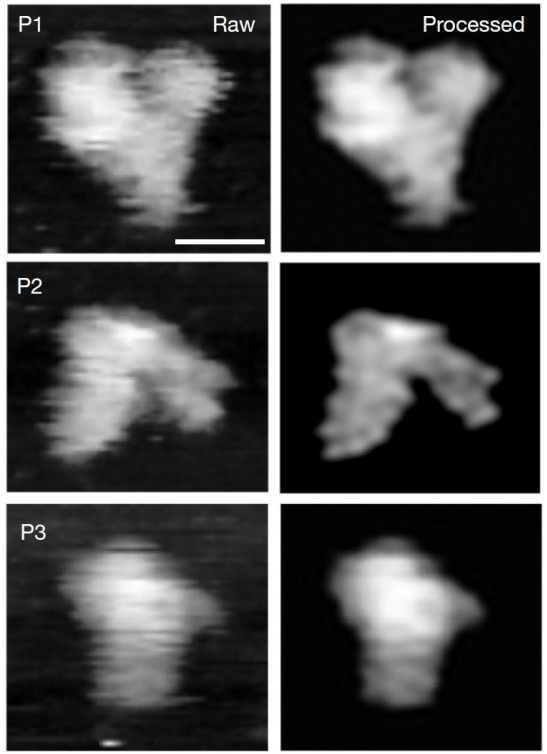
- In situ structure information
- Molecular dynamic information

Main challenge: low resolution

Cryo-EM/ET density map



AFM image

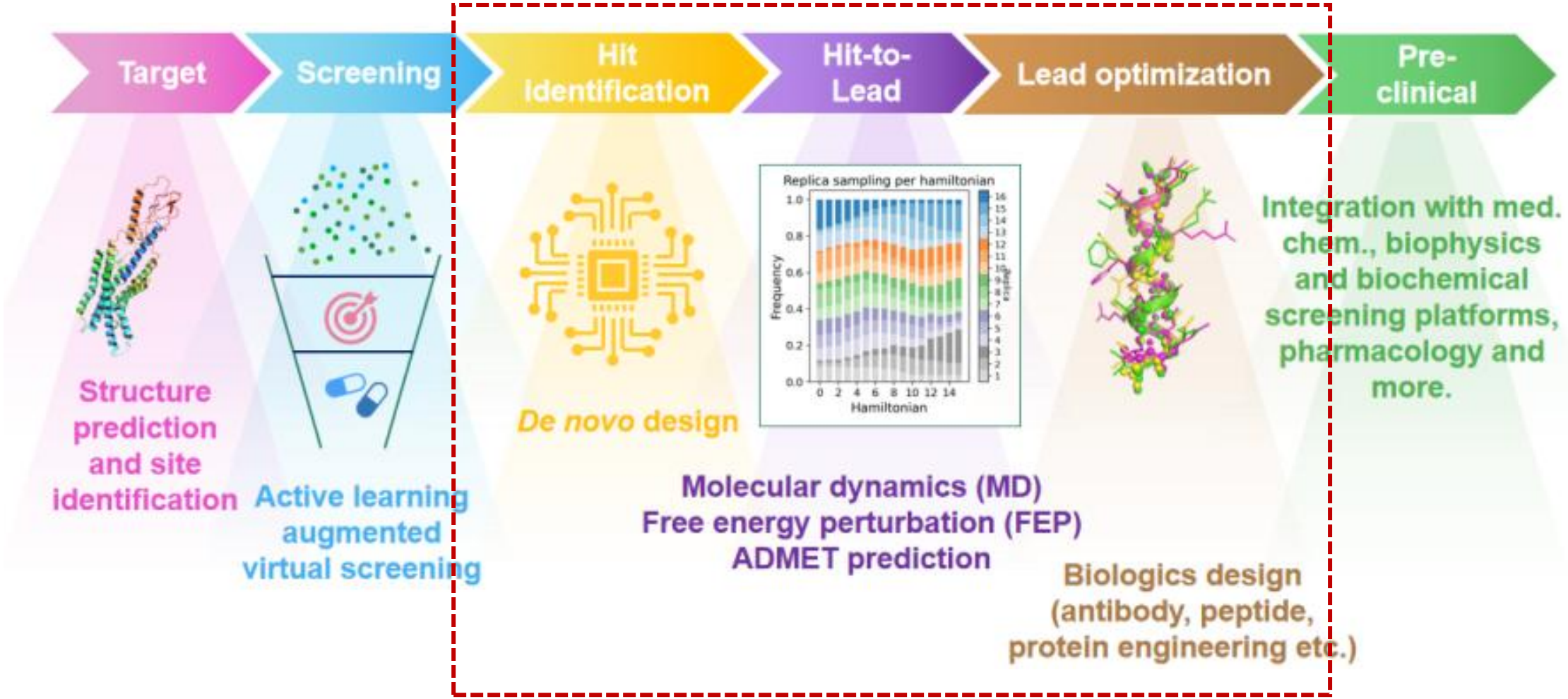


Structure-based drug discovery

- **Problem definition**: design or screen small-molecule ligands that bind with high affinity and specificity to pre-determined biomolecular targets.

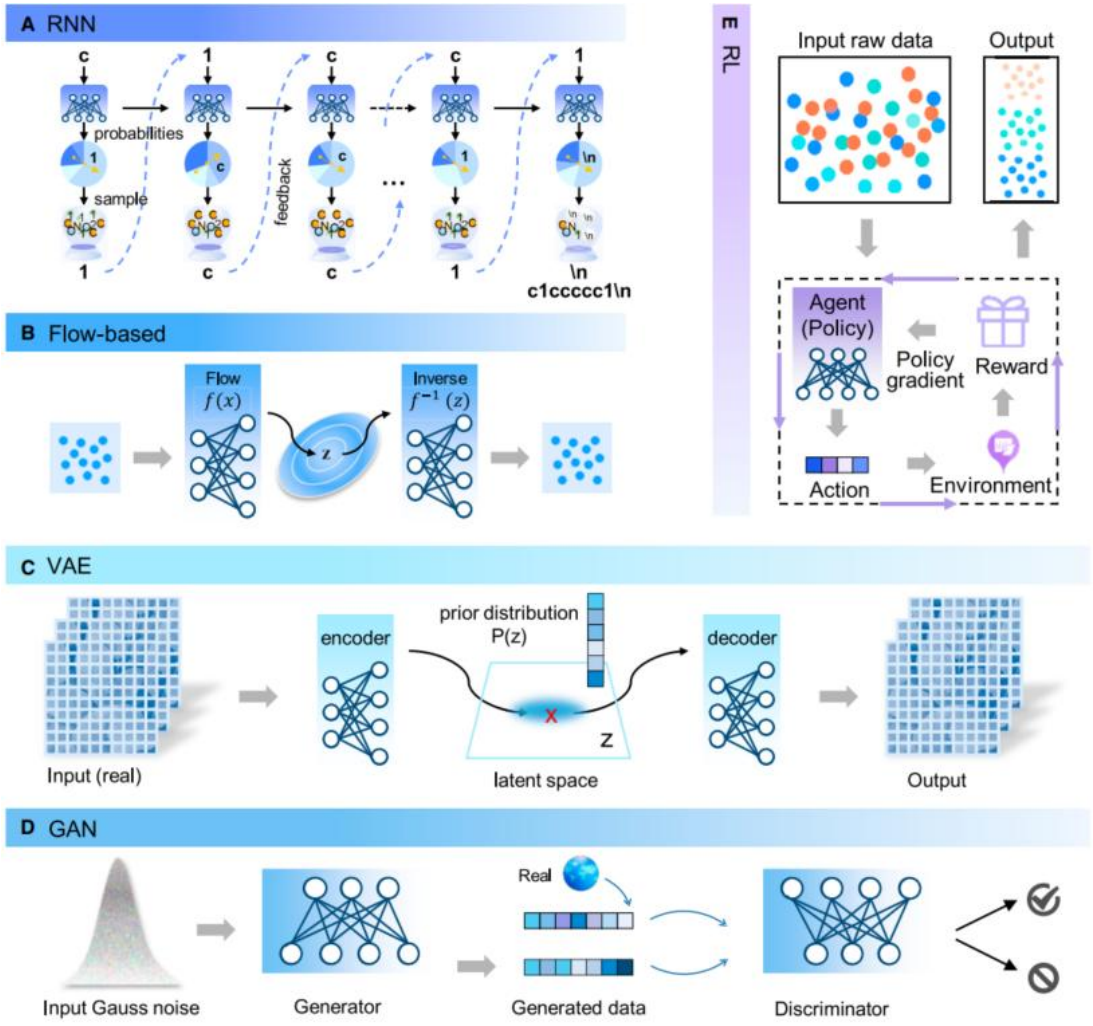
- **Approaches**:
  - Virtual screening using molecular docking
  - Molecular dynamics simulation
  - Complex structure prediction
  - **Structure-based drug design (SBDD)**

Structure-based Drug Design with AI

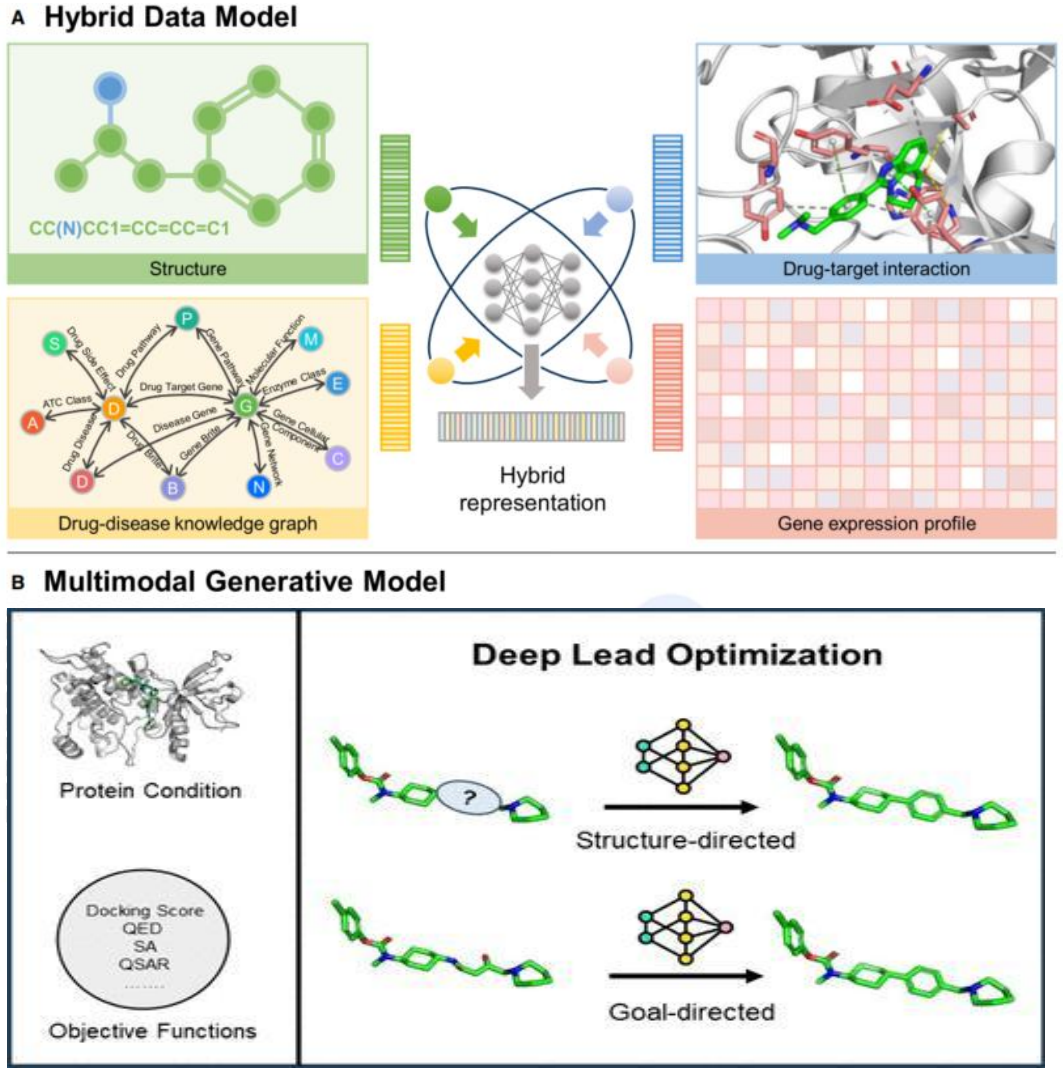


Structure-based Drug Design with AI

Generator with structure conditions

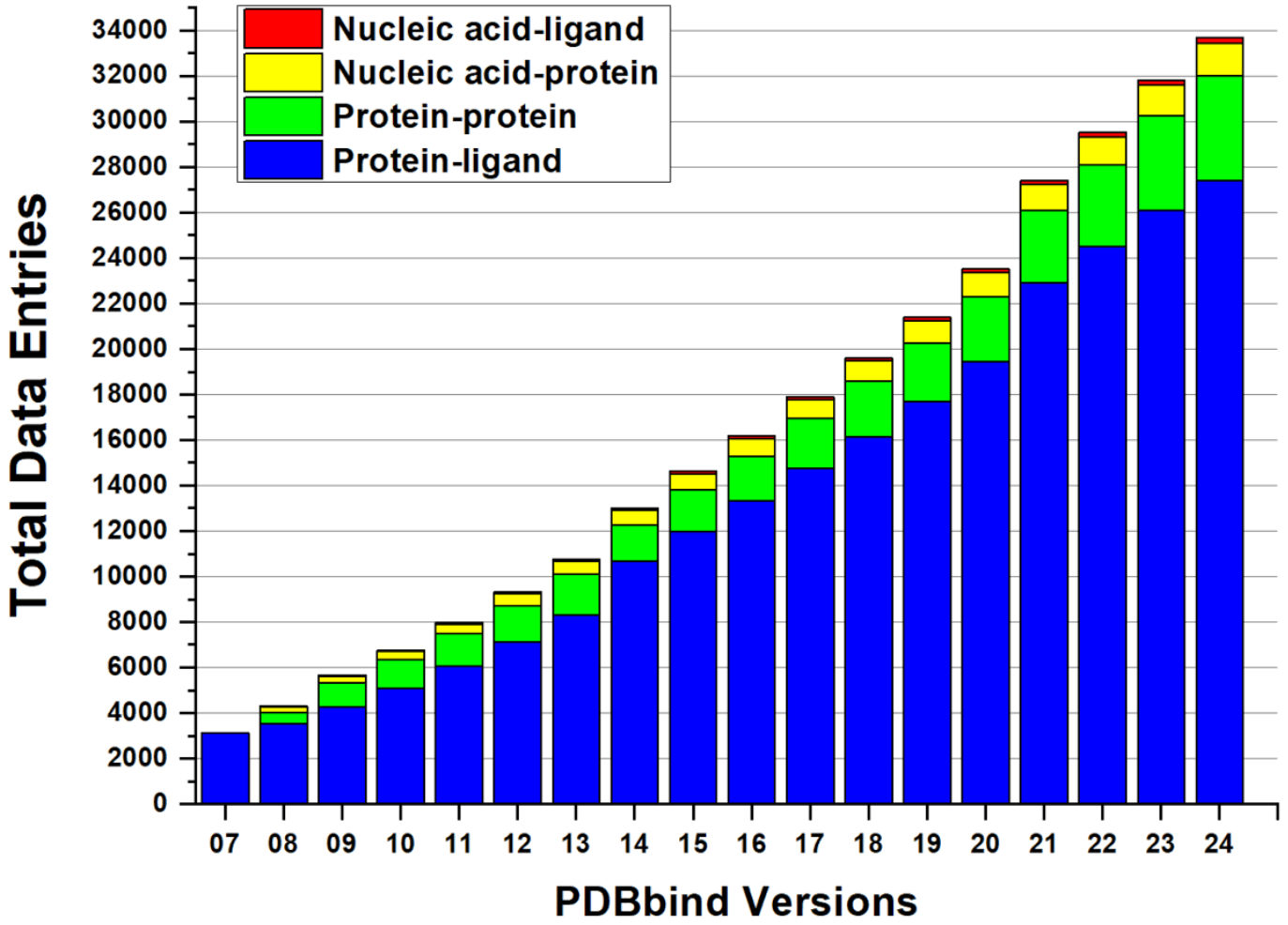


Hybrid structure data



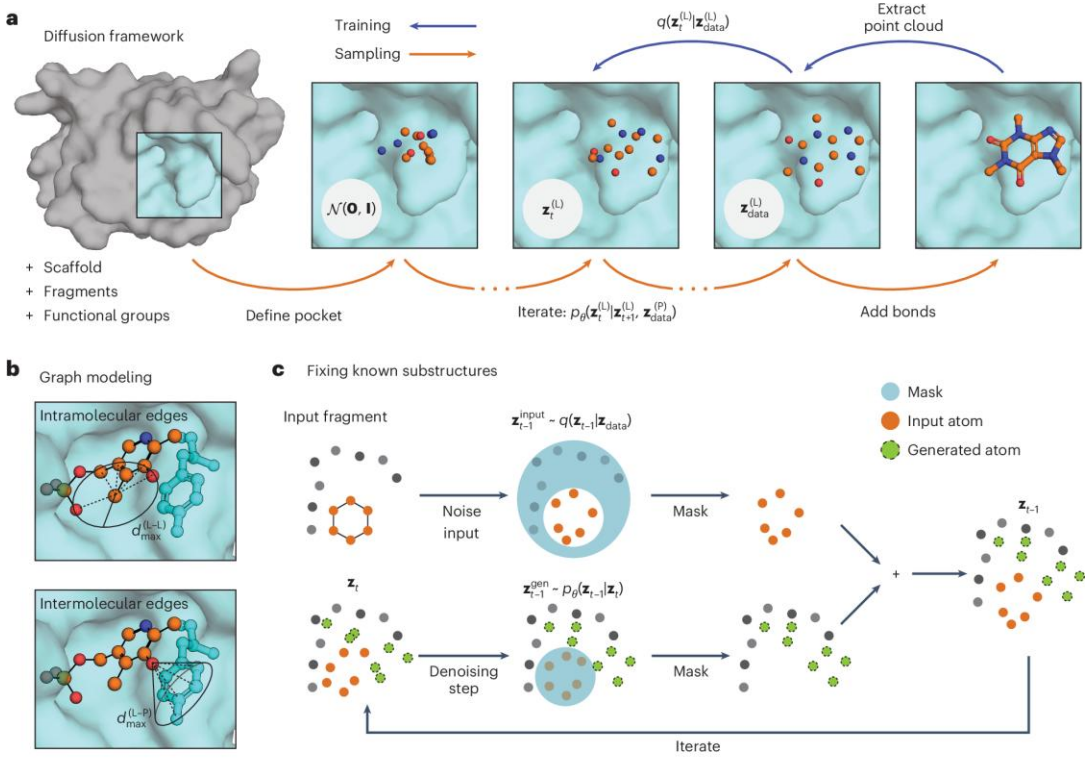
PDBbind: collect experimentally measured binding data for the biomolecular complexes throughout PDB





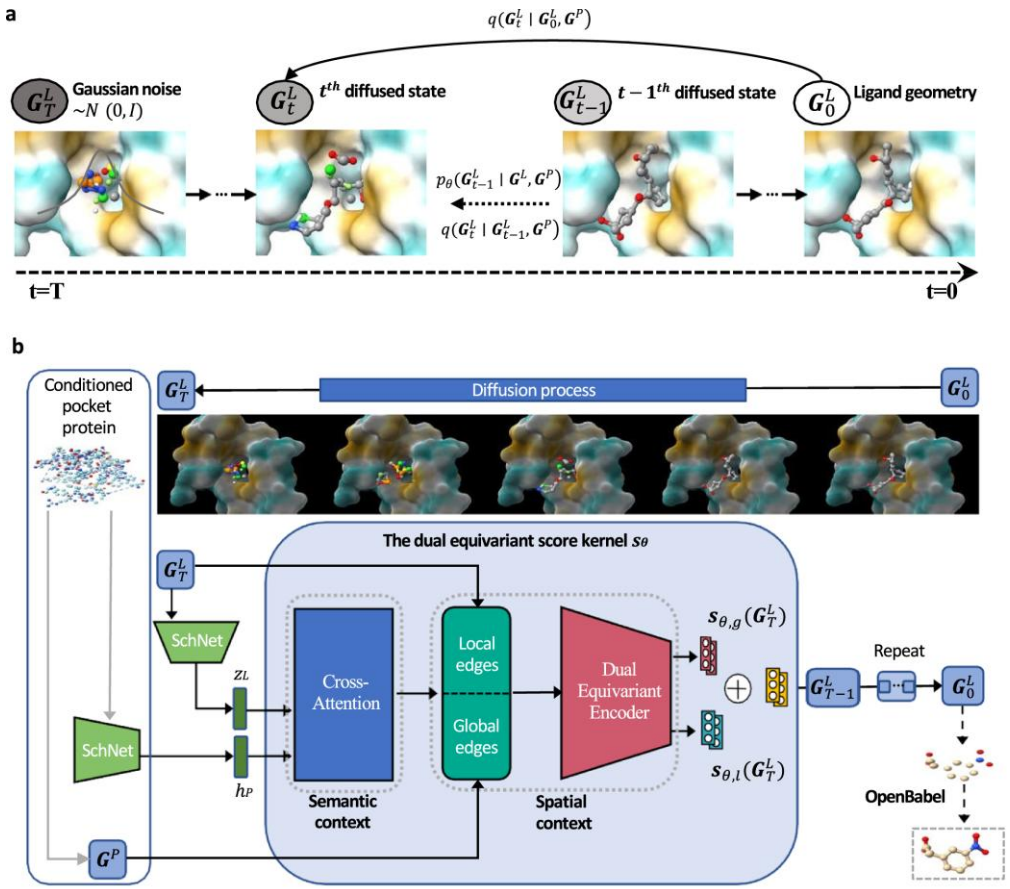
Structure-based drug design with diffusion models

DiffSBDD



Nat Comput Sci, 2024

PMDM

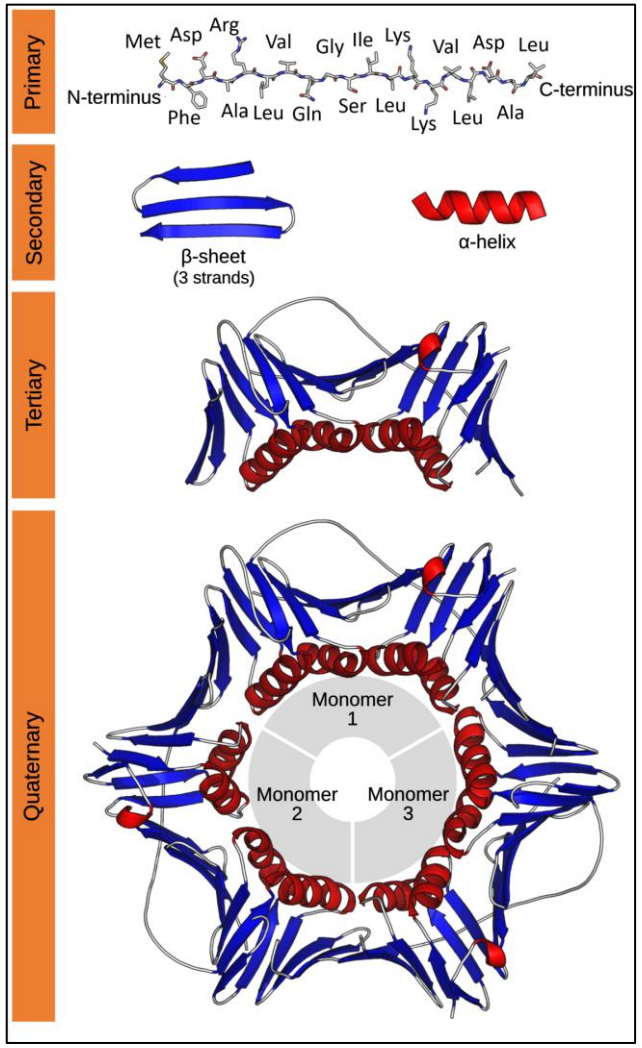


Nat Commun, 2024

Q&A

Recap: Hierarchical organization of biomolecule structure

Proteins



Nucleic Acids

