A high-level programming language for generative protein design

Brian Hie, Salvatore Candido, Zeming Lin, Ori Kabeli,
Roshan Rao, Nikita Smetanin, Tom Sercu, Alexander Rives

Shuxian Zou
2023-01-12

Outlines

- Abstract
- Background & Motivation
- Methods
- Experiments
- Conclusions

Abstract

- **Top-down design of proteins**
  - Provide a language for user to specify desired properties of proteins
  - Properties include atomic coordinates, secondary structures, symmetry and multimerization
  - Modularity and programmability

- **An energy-based generative model**
  - The specified properties are complied into an energy function using ESMFold
  - The energy function is used to guide the search of protein sequences
  - Generality and controllability

Background & Motivation

- ***De novo* protein design**
  - Design novel amino acid sequences that encode proteins with desired properties

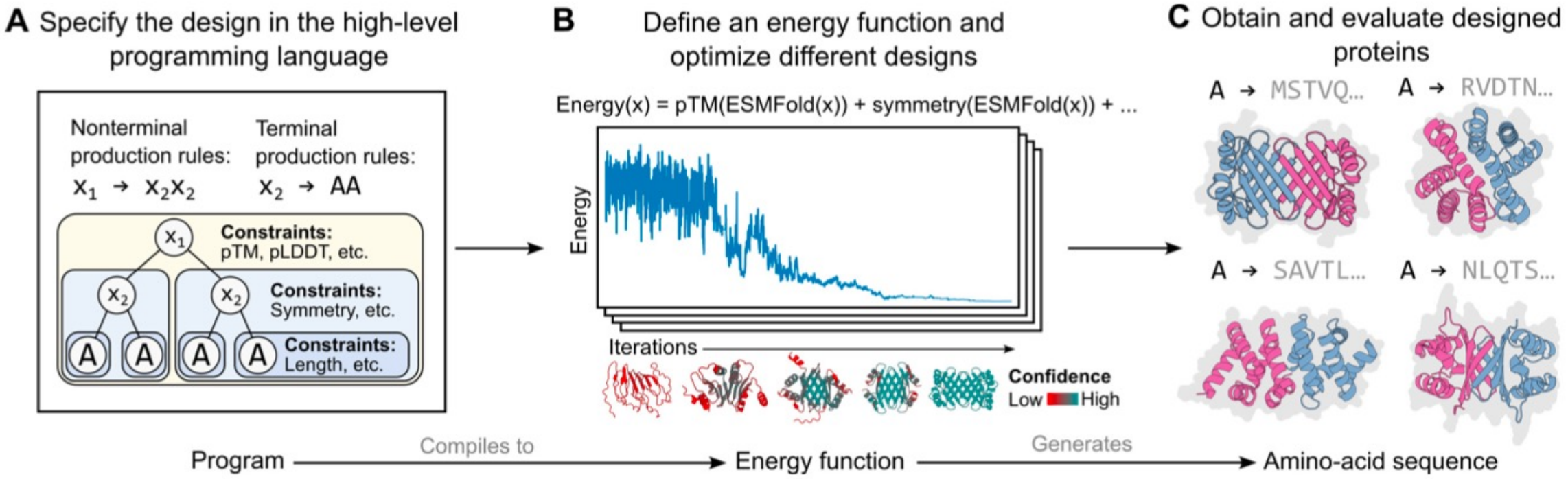
- **Motivation**
  - Previous methods use bottom-up design or top-down design with low combinatorial complexity
  - We propose a programming language for generative protein design, which allows a designer to specify intuitive, modular, and hierarchical programs

- **Challenge**
  - Proteins cannot be decomposed into easily recombinable parts because the local structure of the sequence is entangled in its global context

- **Idea**
  - Translate the high-level programs into low-level sequences and structures by a generative model

Overview of Methods

A protein design system equipped with high-level programming language and powered by a language model based protein structure prediction model

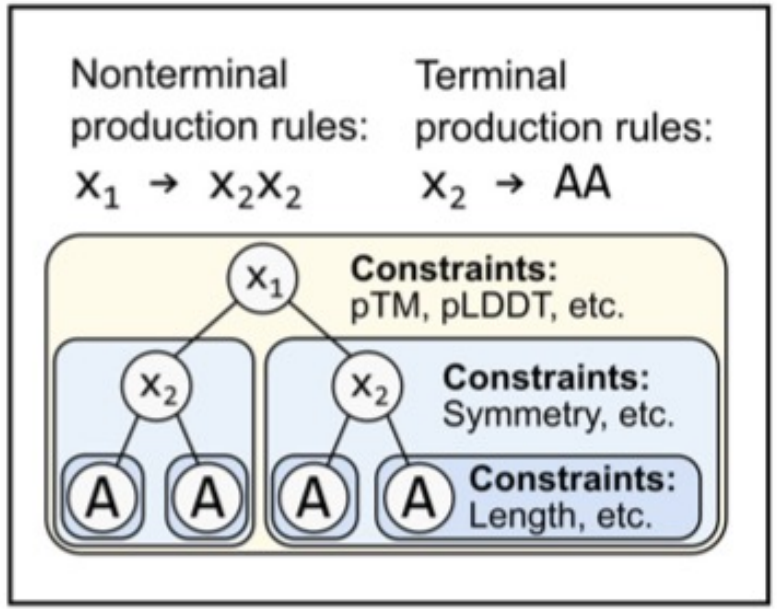


A High-level Programming Language for Protein Design

The language requires: (1) a syntax tree

- **Terminal symbols:** define a unique protein sequence
  - Denoted as A, B, C
- **Nonterminal symbols:** enable hierarchical organization
  - Denoted as x_i , x_1 is the special start symbol
  - Additional nonterminal symbols are used to define hierarchical complexity
- **Rules:** a nonterminal can produce any finite-length permutation of
  - higher-numbered nonterminals
  - terminals
  - mixed terminals and higher-numbered nonterminals

- A complete syntax tree is built by fully expanding the non-terminal x_1 into a set of terminals



$x_1 \rightarrow x_2x_3$ ✓
$x_2 \rightarrow x_1x_3$ ✗
$x_2 \rightarrow x_2x_3$ ✗
$x_1 \rightarrow AB$ ✓
$x_1 \rightarrow x_2B$ ✓
$x_1 \rightarrow Bx_2$ ✓

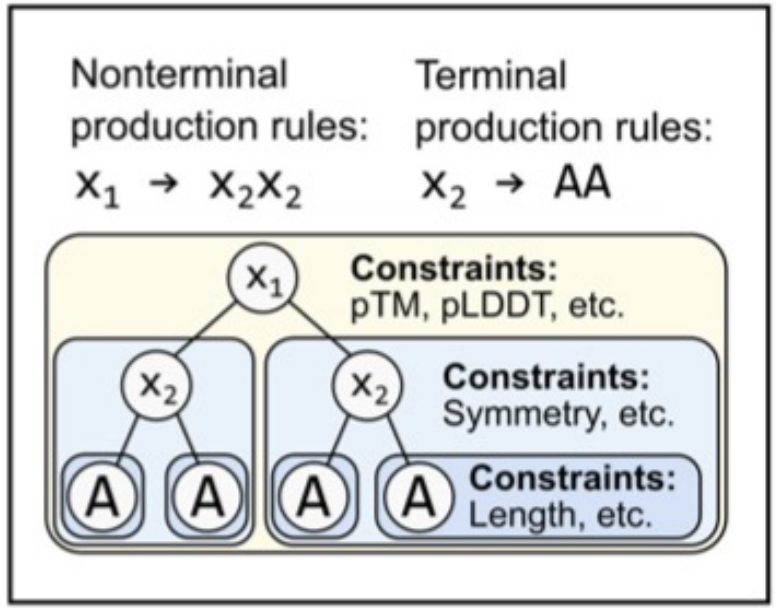
A High-level Programming Language for Protein Design

The language requires: (2) a set of constraints

- A single constraint is defined w.r.t a single node and all of its descendants in the syntax tree

- **A constraint is a function** that takes as input the (sub)tree and its corresponding (sub)sequence and (sub)structure, and outputs a number
  - $f_j(x_i)$: constraint j defined w.r.t node x_i
  - E.g., $f_j(x_1)$ takes the entire syntax tree, the full-length sequence, and full protein structure as input

- Can be **arbitrary and nondifferentiable**, can span a multiple scales of biological complexity



A program fully specified by a syntax tree and its constraints in the high-level language

Compilation of a Program into an Energy Function

- **The energy function:**

$$E(x) = \sum_i \sum_j a_j f_j(x_i)$$

  - $f_j(x_i) = 0$ if a constraint j is not applied to a given node
  - a_j: user-specified scalar

- **Top-down protein design → black-box optimization problem:**

$$\min_x E(x)$$

Simulated Annealing as Optimization Algorithm

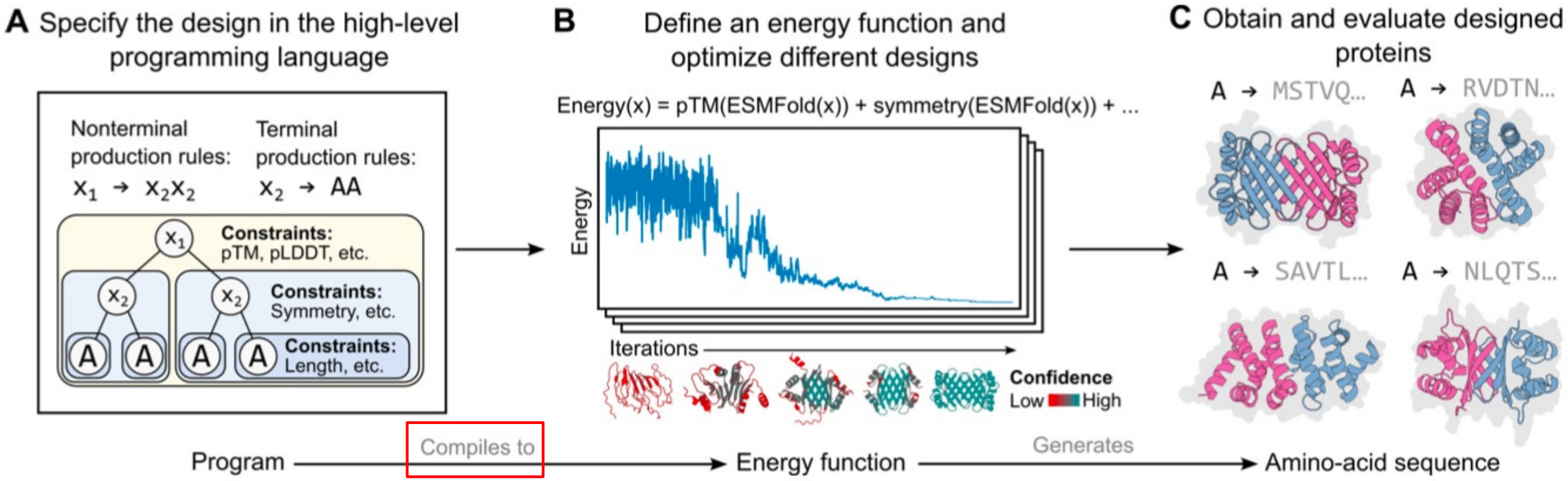
- Initialize the sequence state x^1 (one unique sequence per terminal node) with uniform amino acid probability to a given user-specified length
- Predict the structure of x^1 using ESMFold and compute the energy $E(x^1)$
- In iteration i, propose a mutation to the current protein state x^i and generate x^*:
  - Uniformly sample a terminal symbol
  - Sample a kind of mutation: substitution 60%, insertion 20%, deletion 20%
  - For substitution and insertion, uniformly sample an amino acid (except cysteine)
  - Uniformly sample a sequence position for the mutation
- Predict the structure of x^* using ESMFold and compute the energy $E(x^*)$
- Let $\Delta E = E(x^*) - E(x^i)$,
  - If $\Delta E < 0$, accept x^* as a new sequence state
  - Else, accept x^* with probability $e^{-\frac{\Delta E}{T_i}}$
- Return x^M and its predicted structure

Cooling schedule: $T_i = (\frac{T_{min}}{T_{max}})^{\frac{i}{M}}$

$T_{min} = 0.0001$, $T_{max} = 1$

M: user-specified number of annealing steps

Overview of Methods



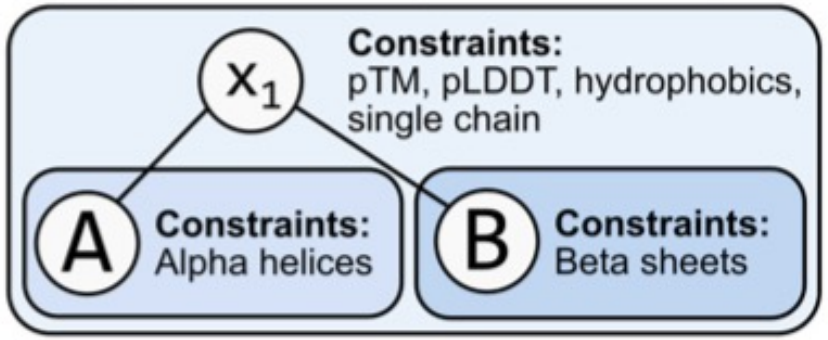
Constraint Implementation

Basis of constraint compilation: structure predicted by ESMFold

- Input the entire protein sequence to ESMFold and get all-atom structure predictions (atomic coordinates)
- 11 constraints in total

(1) Single chain constraint (part of structure prediction)

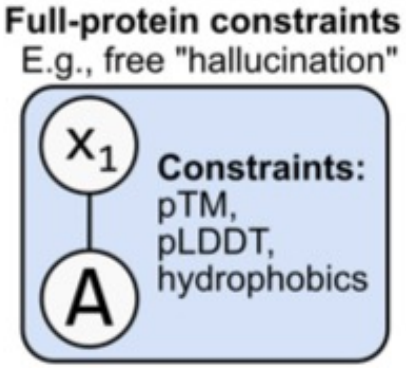
- By default, all terminal nodes correspond to separate chains without this constraint
- When this constraint is applied to a given node, it constrains all terminal symbols to be part of a single chain, according to the left-to-right order defined in the syntax tree.
- Enforced as part of the structure prediction, prior to the energy function compilation.



Constraint Implementation

(2) Structure prediction confidence (pTM and pLDDT)

- Prefer proteins with higher structure prediction confidence (more naturally plausible and designable)
- pTM: model's confidence on the overall structure prediction
- mean pLDDT: model's confidence in the backbone atomic coordinate predictions
- **energy = α(1-pTM) + β(1-pLDDT)** (α, β: user-specified weights)



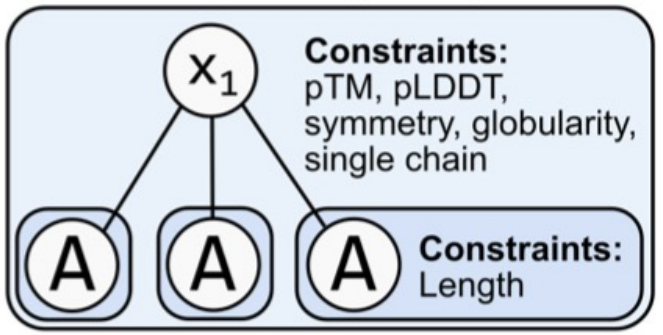
(3) Surface-exposed hydrophobics

- Prefer soluble and monomeric proteins (high hydrophobicity leads to protein aggregation and insolubility)
- Detect surface-exposed atoms: Shrake-Rupley "rolling probe" algorithm (*biotite*)
- **energy = # (surface exposed hydrophobic residues) / # (hydrophobic residues)**

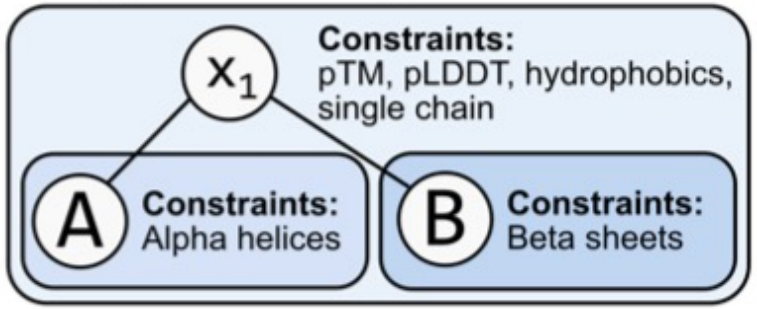
Constraint Implementation

(4) Globularity

- Prefer a protein chain to pack into a globular structure
- Compute the centroid c of a set of atomic coordinates
- **energy = variance**$(\{\boldsymbol{d}(\boldsymbol{a_i}, \boldsymbol{c}) | \boldsymbol{i} = \boldsymbol{1}, \ldots, \boldsymbol{n}\})$



E.g., mixed secondary structure domains



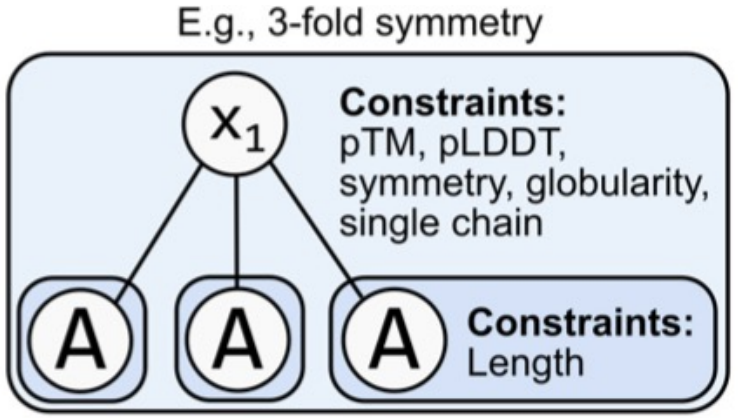
(5) Secondary structure

- Prefer proteins with user-defined secondary structure
- Annotate residue secondary structure: P-SEA algorithm (*biotite*)
- **energy = 1 – (# residues belongs to the desired secondary structure / # residues)**

Constraint Implementation

(6) Rotational symmetry

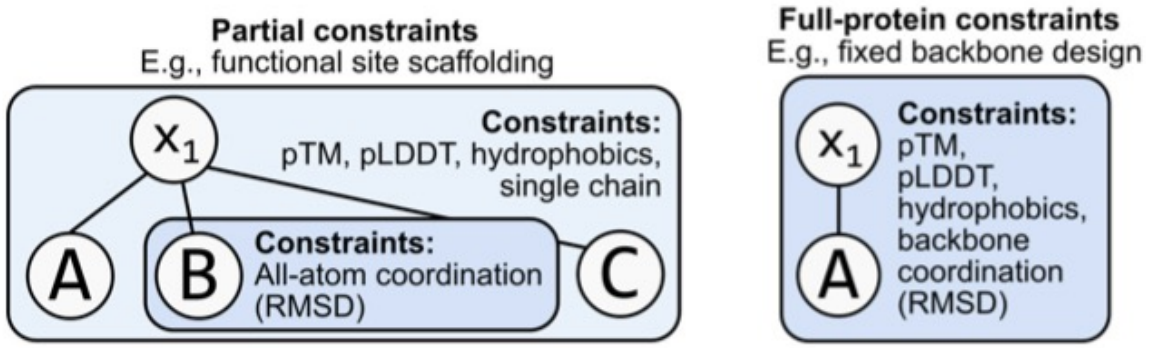
- Prefer ring-like structures (equally locate each substructures)
- Consider the centroids of the immediate children of the constraint's node
- E.g., $x_1 \rightarrow x_2x_3x_4$, compute centroids $c(x_2), c(x_3), c(x_4)$
- **energy = variance($\{\boldsymbol{d_{23}}, \boldsymbol{d_{34}}, \boldsymbol{d_{42}}\}$)** (adjacent distances)



(7) Globular symmetry

- Prefer globular symmetric structures (such as polyhedral)
- Consider the centroids of the immediate children of the constraint's node
- E.g., $x_1 \rightarrow x_2x_3x_4x_5$, compute centroids $c(x_2), c(x_3), c(x_4), c(x_5)$
- **energy = variance($\{\boldsymbol{d_{23}}, \boldsymbol{d_{24}}, \boldsymbol{d_{25}}, \boldsymbol{d_{34}}, \boldsymbol{d_{35}}, \boldsymbol{d_{45}}\}$)** (all pairwise distances)

Constraint Implementation



(8) All-atom coordination

- For functional site scaffolding
- Constrain (a portion of) the protein to match the structure of a known functional site in nature
- y_{native}: coordinates of a list of atoms from a native protein structure
- y_{design}: coordinates of all atoms in the corresponding (sub)tree
- constrained root mean square deviation (cRMSD):

$$\text{cRMSD}\left(y_{native}, y_{design}\right) = \min_{T} \frac{1}{n} \sum_{i=1}^{n} \left\| a_i(y_{native}) - T(a_i(y_{design})) \right\|^{1/2}$$

- distance-matrix RMSD (dRMSD):

$$\text{dRMSD}\left(y_{native}, y_{design}\right) = \left(\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij}(y_{native}) - d_{ij}(y_{design}))^2 \right)^{1/2}$$

- **energy** $= \boldsymbol{\alpha}\mathbf{cRMSD}\left(\boldsymbol{y_{native}}, \boldsymbol{y_{design}}\right) + \boldsymbol{\beta}\mathbf{dRMSD}\left(\boldsymbol{y_{native}}, \boldsymbol{y_{design}}\right)$ (α, β: user-specified weights)

T: structural transformation
a_i: coordinates of the ith atoms

(9) Backbone atom coordination

- For fixed backbone design, only constrain the backbone atoms of the protein structure (no side chains)

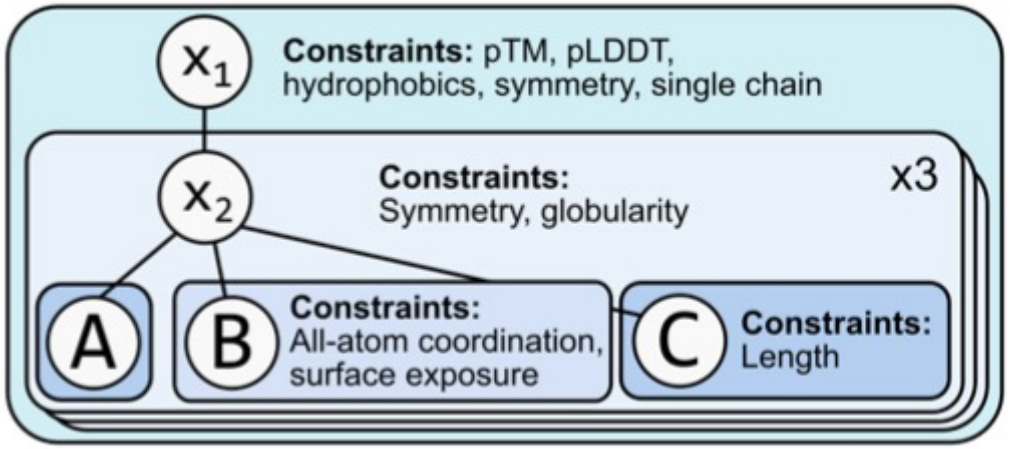
Constraint Implementation

(10) Surface exposure

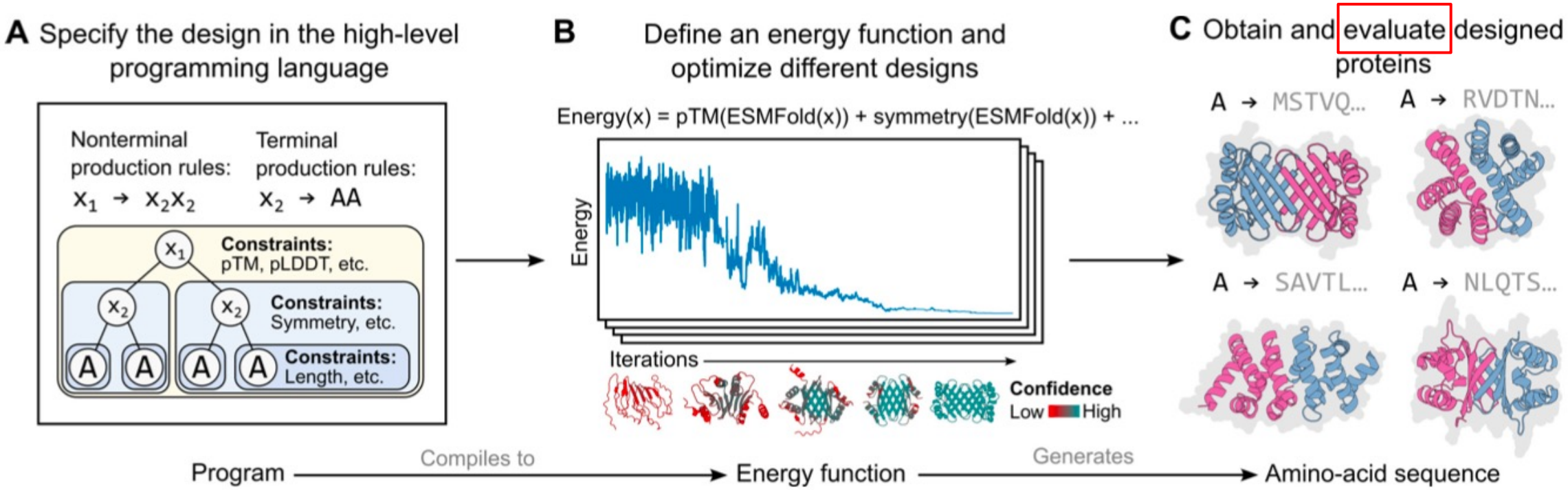
- Desire a given set of residues be exposed on the surface of the protein (e.g. a protein binding site)
- **energy = 1 - (# surface exposed atoms within the (sub)tree structure / # atoms within the (sub)tree structure)**

(11) Length

- Desire a user-specified number of length
- Hard-length constraint: disallow insertions and deletions



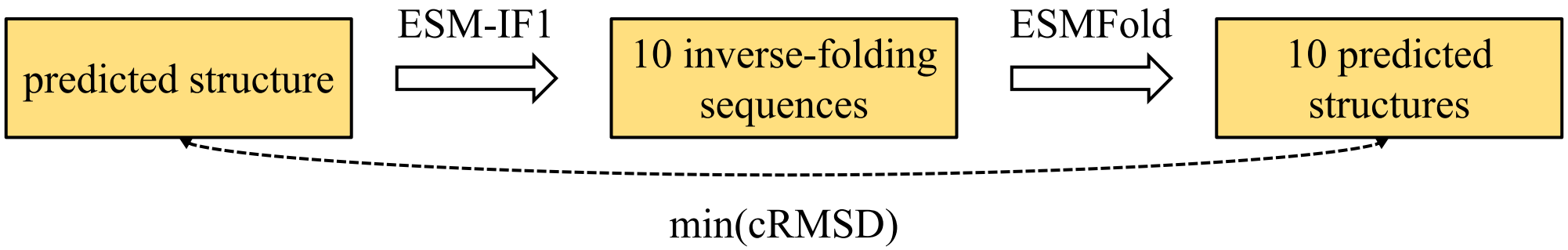
Overview of Methods



Designed Protein Evaluation Metrics

- **Structural novelty**
  - Search exhaustively over the PDB database to find the experimental structure with the highest TM-score to the designed structure

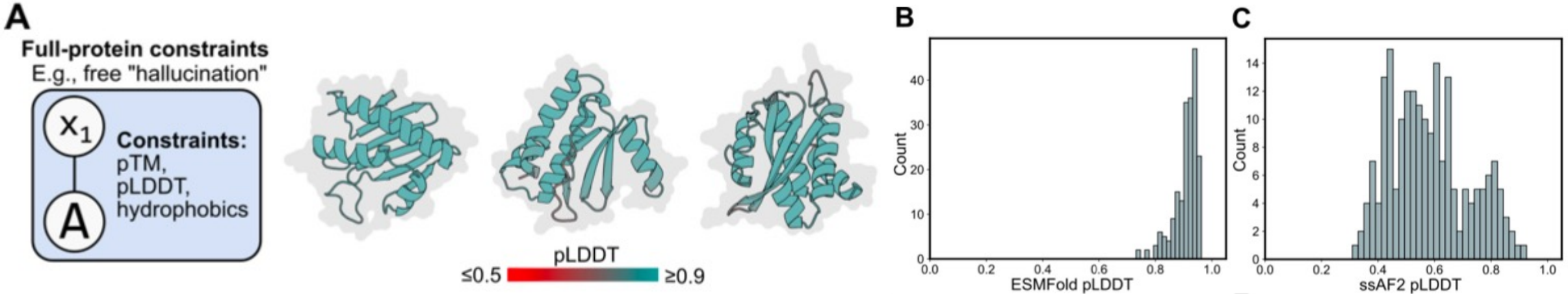
- **Inverse folding roundtrip experiments to access "designability" of a structure prediction**
  - Assumption: a designed protein backbone is "designable" if it can be recovered by roundtripping



Experiments – Full-protein constraints

Free hallucination

- Constraints: ((1-pTM) + (1-pLDDT) + hydrophobics) on the whole protein
- Ran simulated annealing over 30,000 iterations with $T_{max} = 1$ across **200 seeds**



Result: Able to generate high-confidence structures.

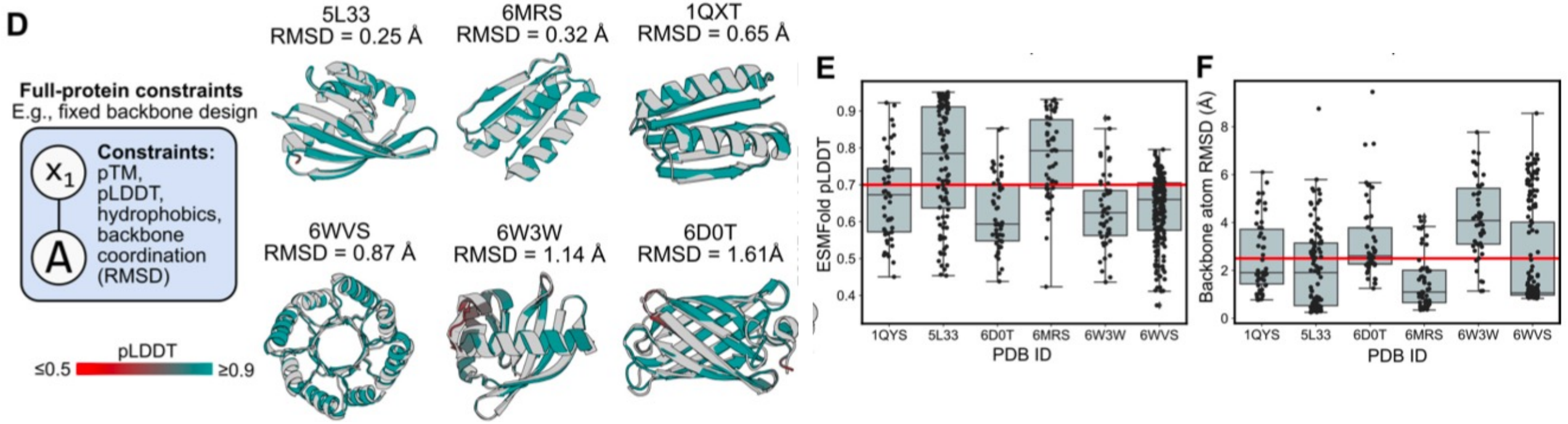
ESMFold
100% pLDDT > 0.7

AlphaFold2
22% pLDDT > 0.7

Experiments – Full-protein constraints

Fixed backbone design

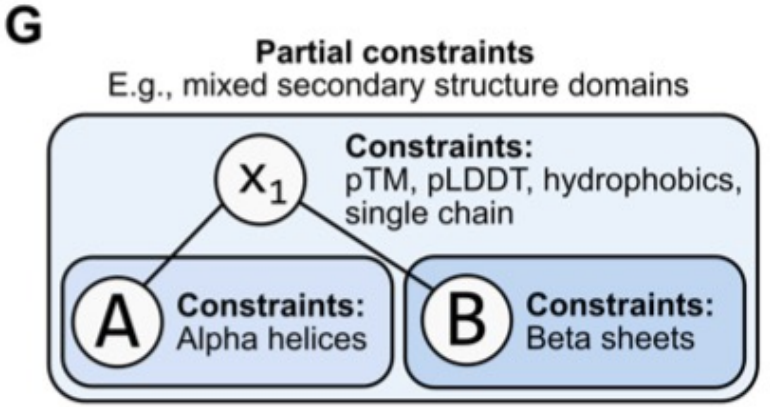
- Constraints: 2dRMSD + cRMSD + (1-pTM) + (1-pLDDT) + 0.5 hydrophobics
- Ran simulated annealing over 30,000 iterations with $T_{max} = 1$ across **at least 50 seeds** for each of the **six** de novo backbones.

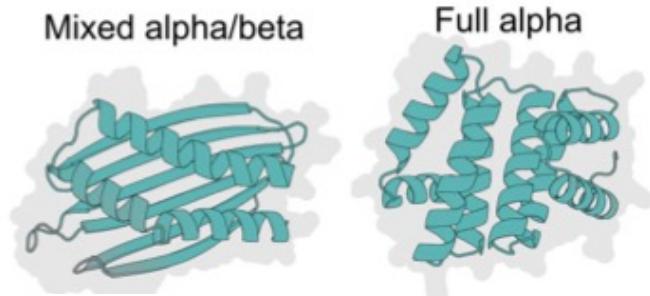


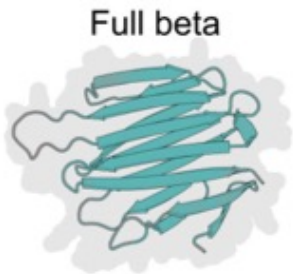
Experiments – Partial constraints

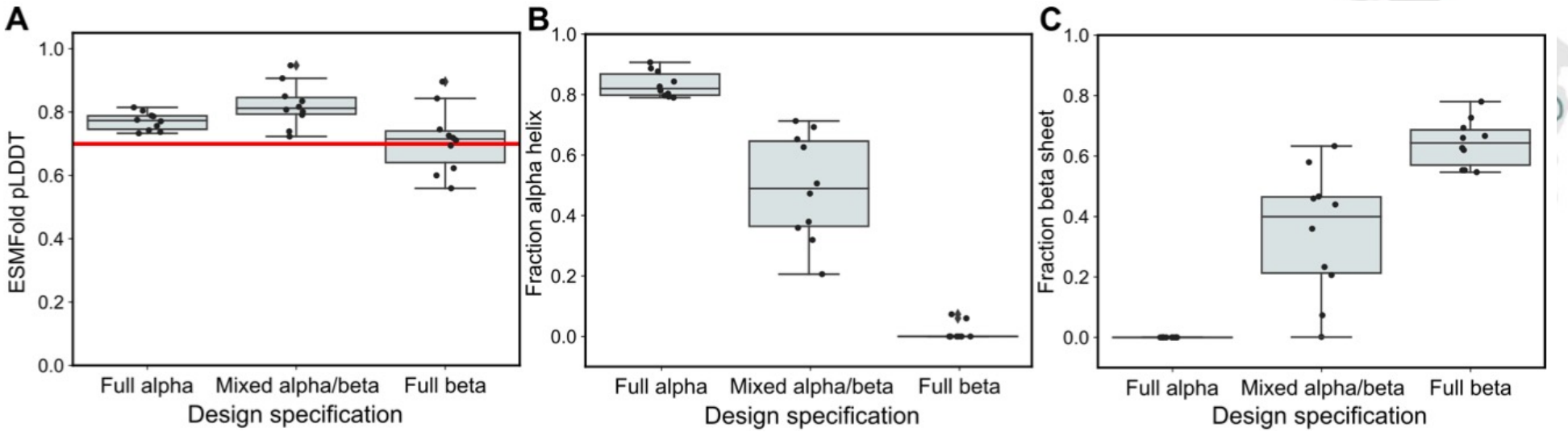
Secondary structure design

- Constraints: 10 secondary structure + (1-pTM) + (1-pLDDT) + hydrophobics
- Ran simulated annealing over 30,000 iterations $T_{max} = 1$ for 10 seeds for each of the three programs (all alpha, all beta, mixed)





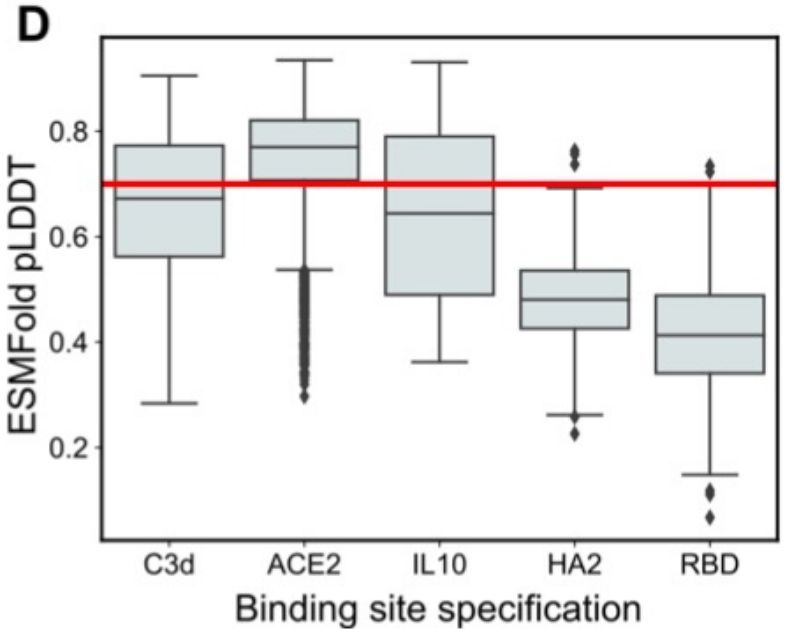


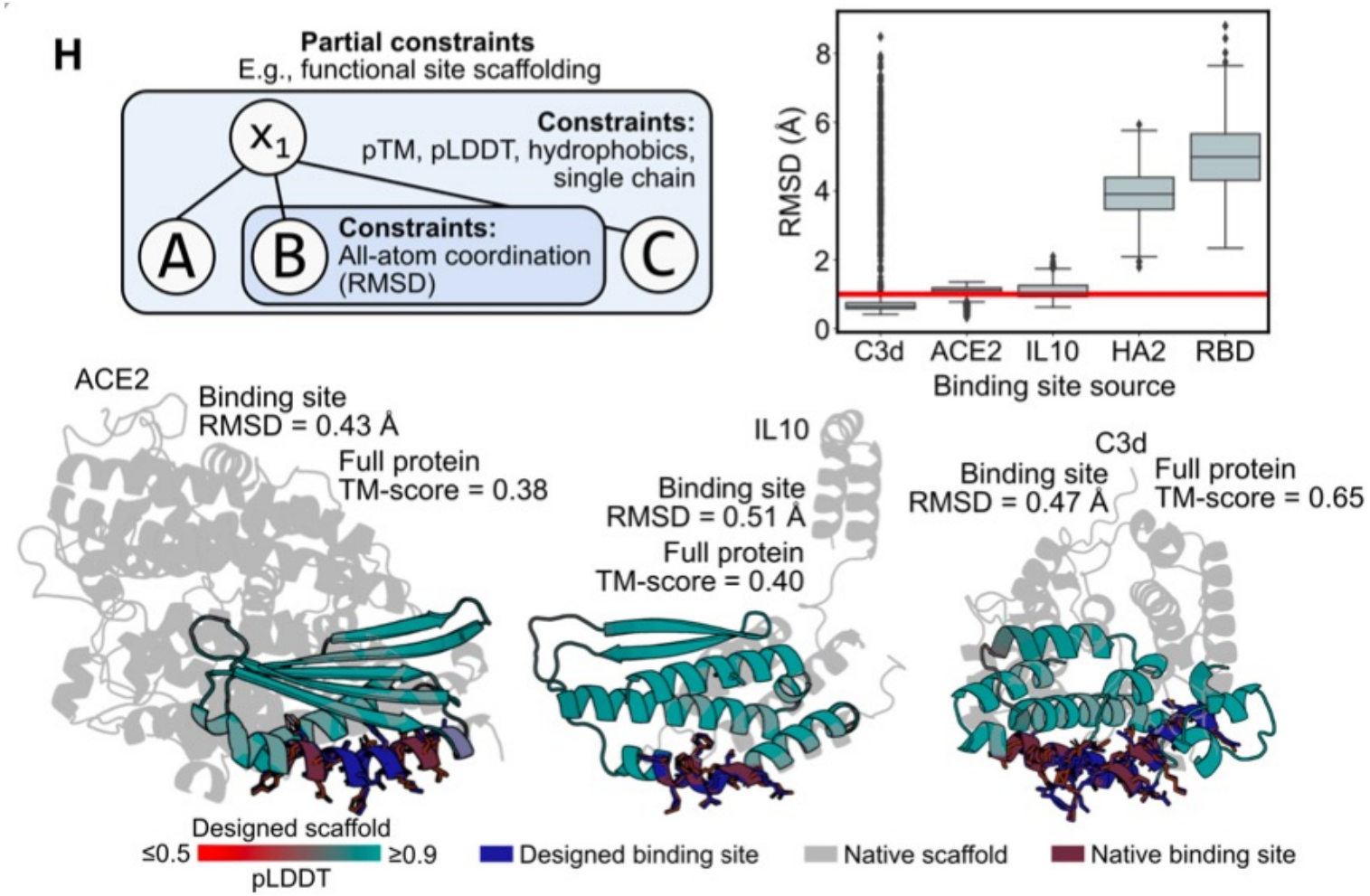


Experiments – Partial constraints

Single functional site scaffolding

- Ran simulated annealing over 30,000 iterations with $T_{max} = 1$ for **1,000 seeds** for each of the **five** binding sites.

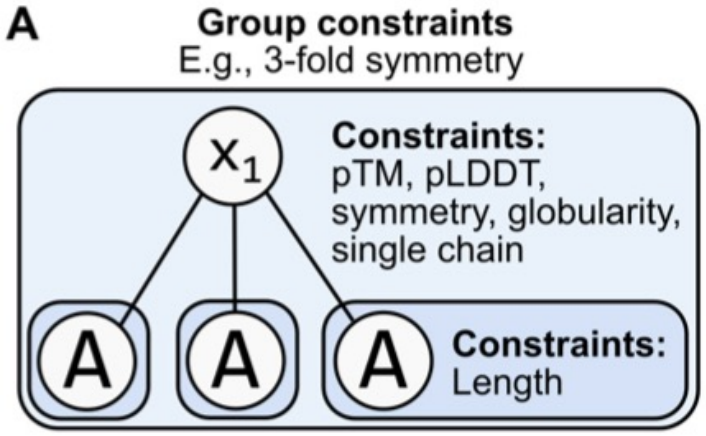




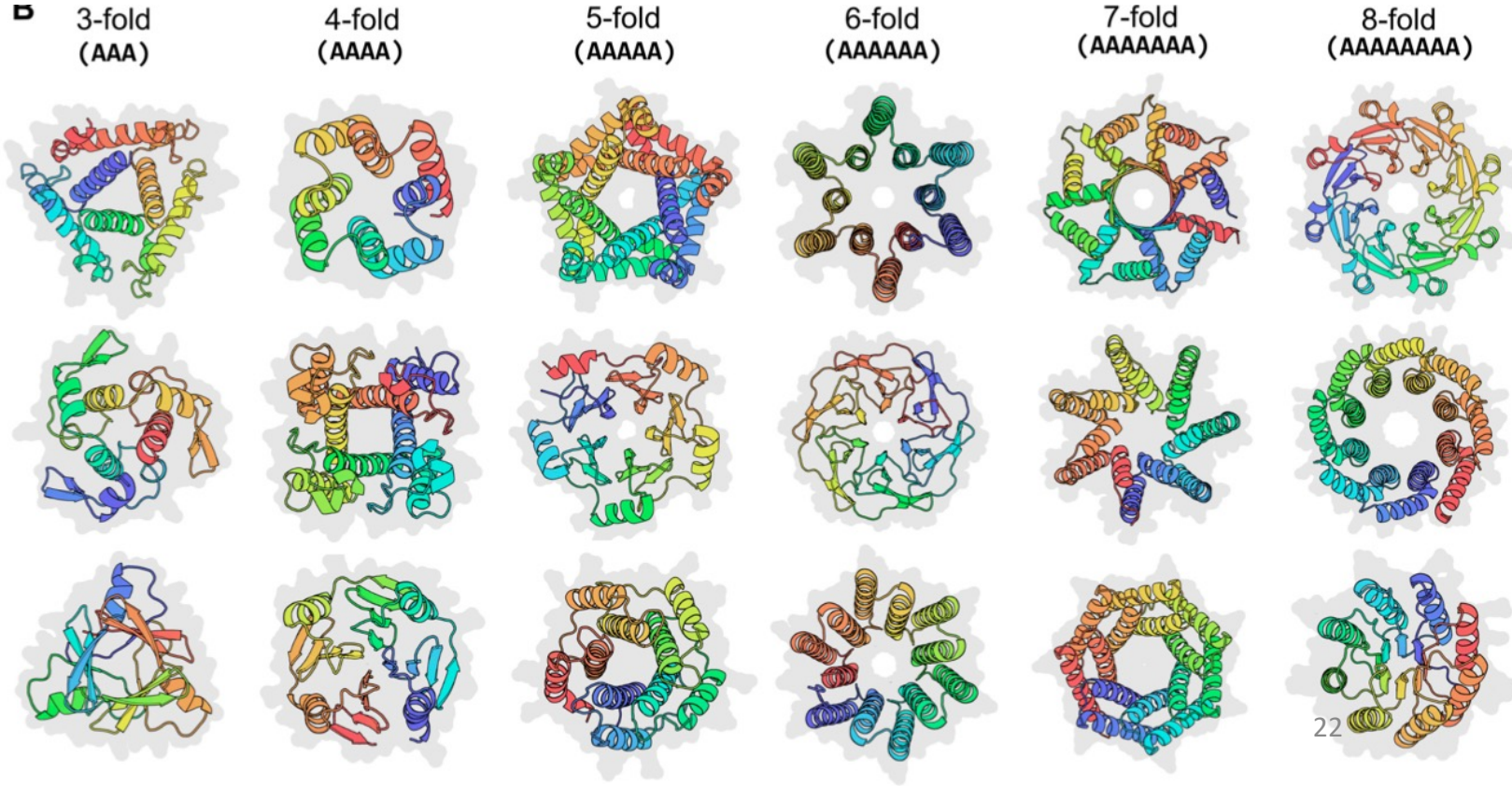
Experiments – Symmetric and multimeric group constraints

Symmetric protein design

- Constraints: rotational symmetry + (1-pTM) + (1-pLDDT) + hydrophobics, length, single chain
- Ran simulated annealing over 30,000 iterations with a starting temperature of 1 for 10 seeds for each of the six fold symmetries and each of the three length constraints



Nearest-PDB TM-scores range from 0.47 to 0.86, with a median TM-score of 0.64

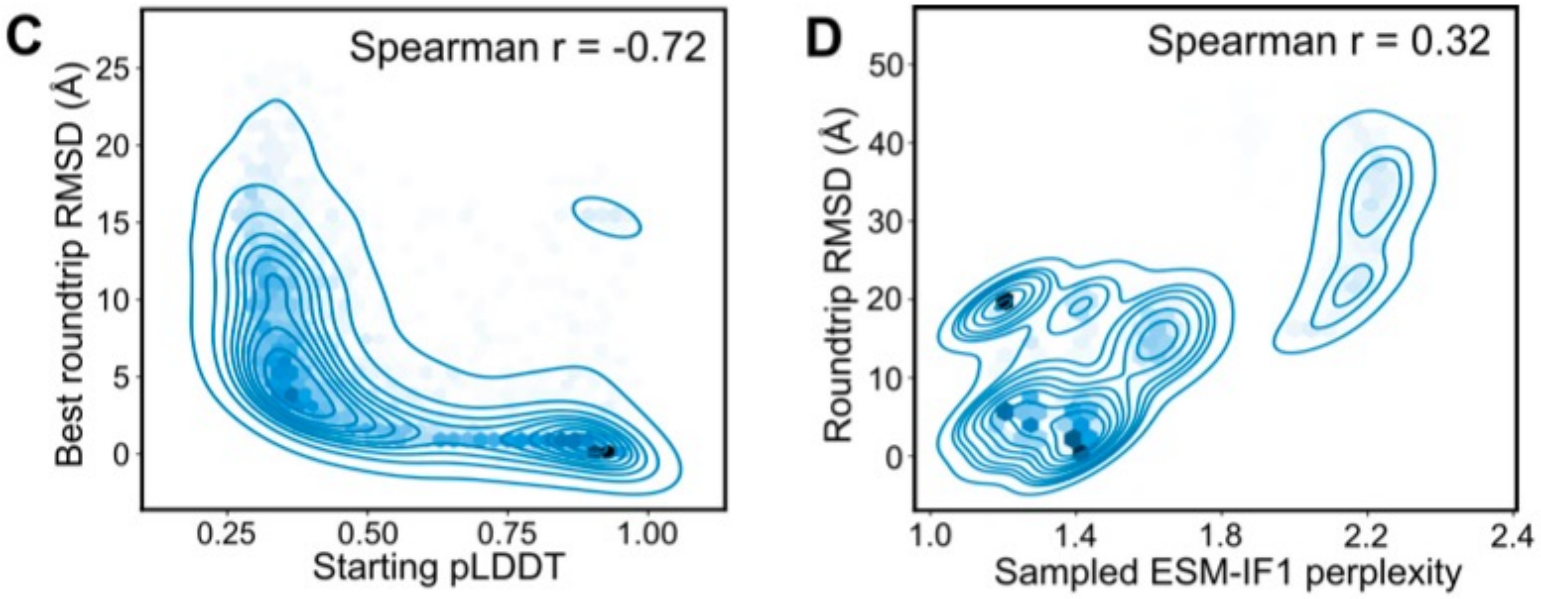


Experiments – Symmetric and multimeric group constraints

Symmetric protein design

- We observed that a more confident design is associated with roundtrip success (low roundtrip RMSD).
- We observed that a lower perplexity sequence is associated with roundtrip success.

1000 randomly sampled protein designs

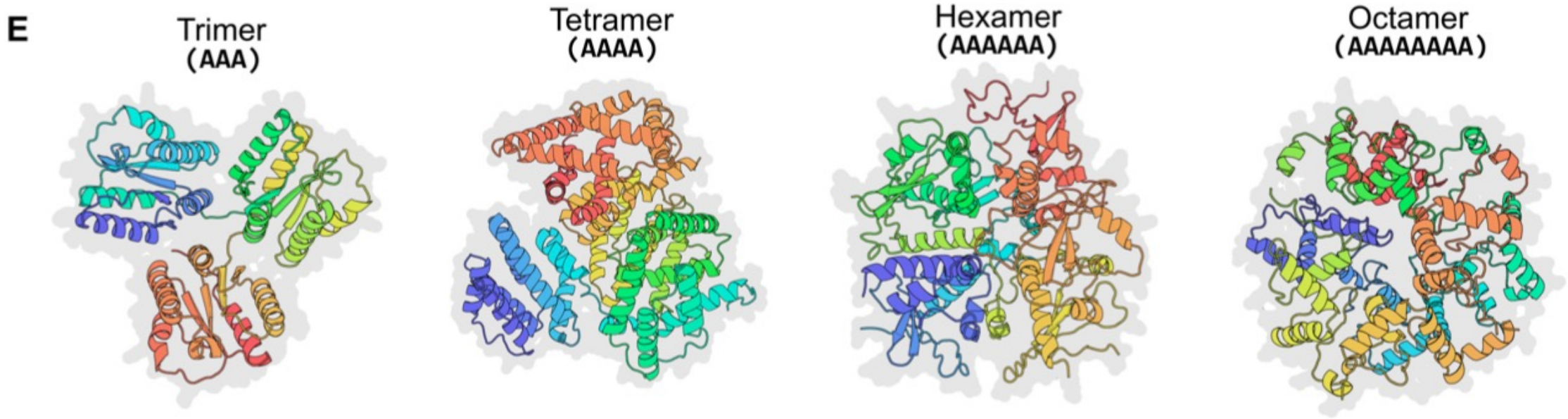


1000 randomly sampled inverse folding samples

Experiments – Symmetric and multimeric group constraints

Homo-oligomer design

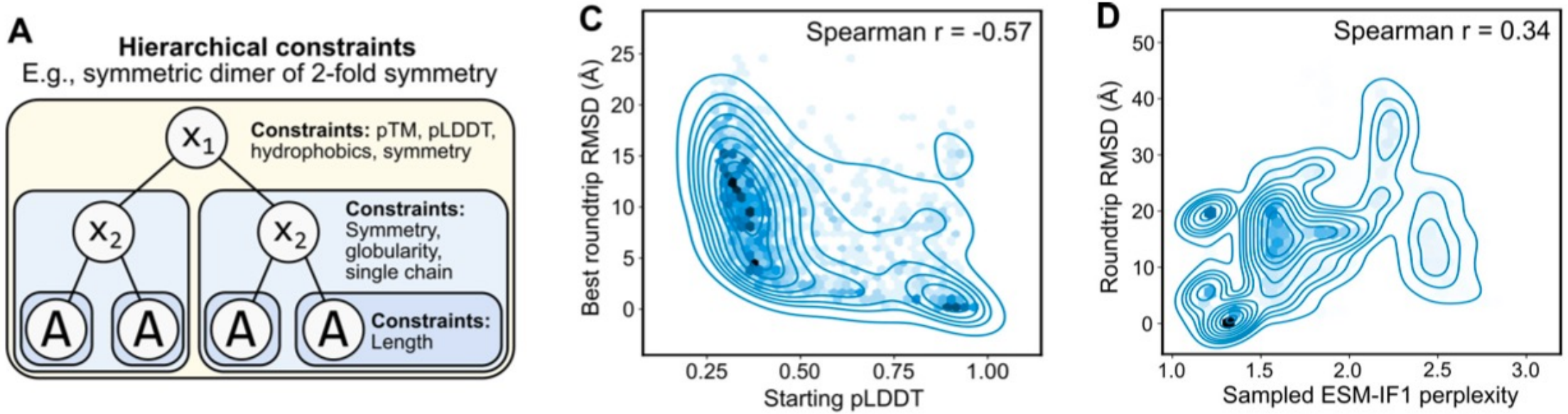
- Constraints: globular symmetry + (1-pTM) + (1-pLDDT) + hydrophobics + 0.1 globularity at each terminal symbol, length = 720 residues, ~~single chain~~
- Ran simulated annealing over 30,000 iterations with $T_{max} = 1$ for 10 seeds for each of oligomerization levels



Experiments – Hierarchical constraints

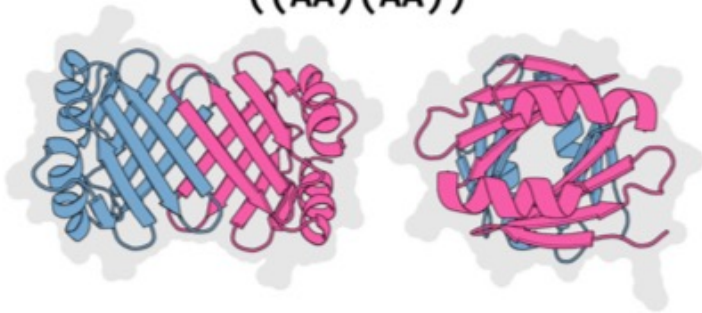
Two-level symmetry design → homo-oligomers

- Constraints: weights are all set to 1 for each constraint
- Dimer of 2-fold, 3-fold, 4-fold; Trimer of 2-fold, 3-fold, 4-fold; Tetramer of 2-fold, 3-fold, 4-fold
- Ran simulated annealing over 30,000 iterations with $T_{max} = 1$ for 10 seeds for each of these programs.

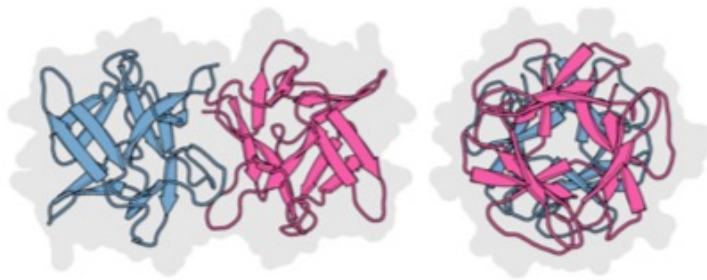


B

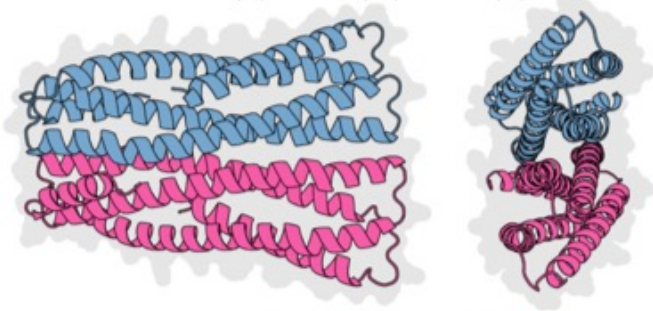
Dimer of 2-fold
((AA)(AA))



Dimer of 3-fold
((AAA)(AAA))

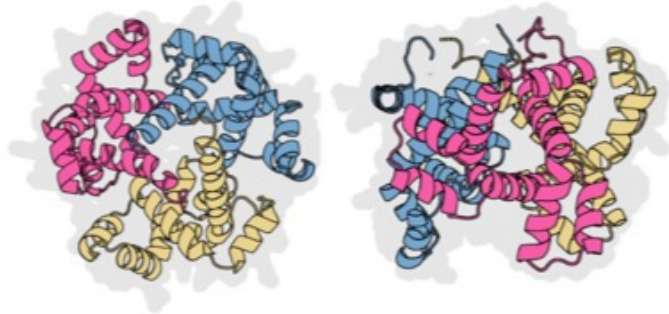


Dimer of 4-fold
((AAAA)(AAAA))



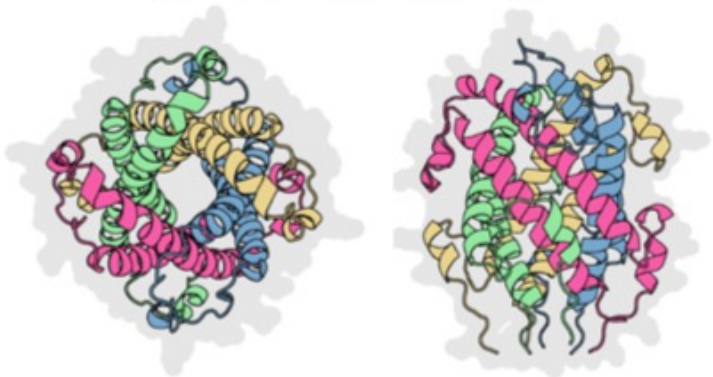
Trimer of 2-fold
((AA)(AA)(AA))

Trimer of 3-fold
((AAA)(AAA)(AAA))

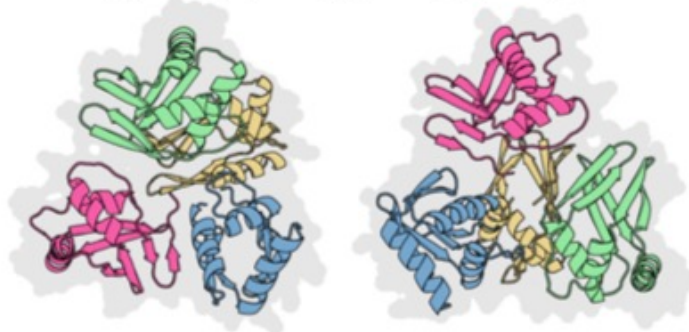


Trimer of 4-fold
((AAAA)(AAAA)(AAAA))

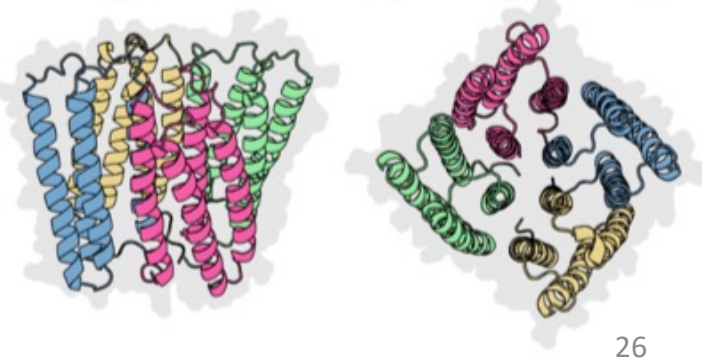
Tetramer of 2-fold
((AA)(AA)(AA)(AA))



Tetramer of 3-fold
((AAA)(AAA)(AAA)(AAA))



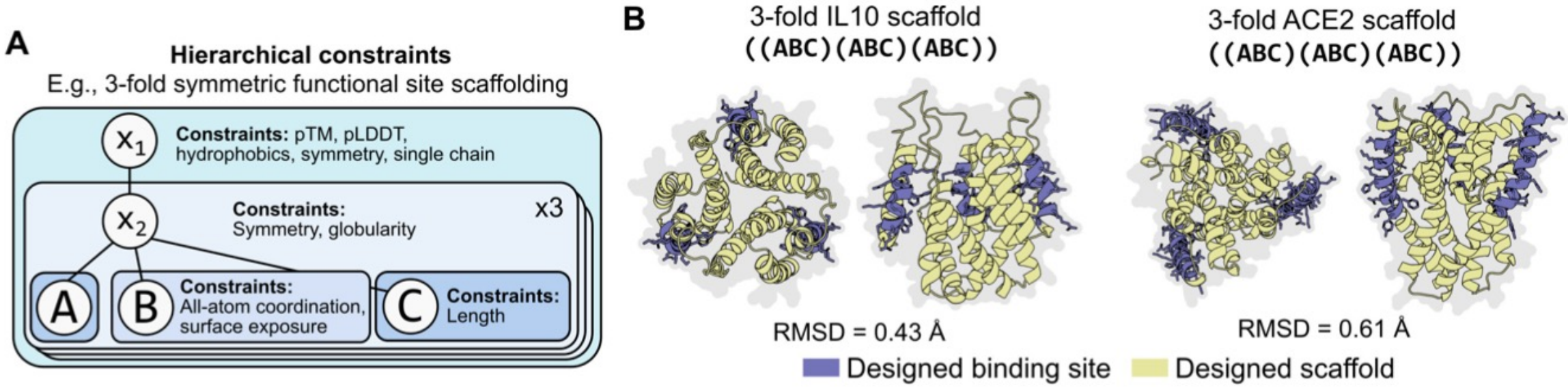
Tetramer of 4-fold
((AAAA)(AAAA)(AAAA)(AAAA))

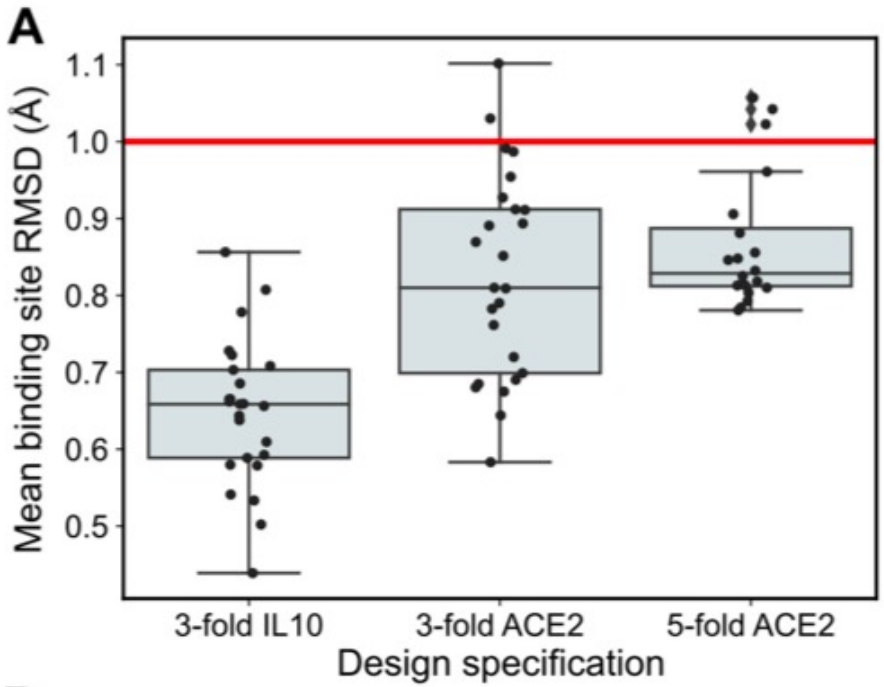


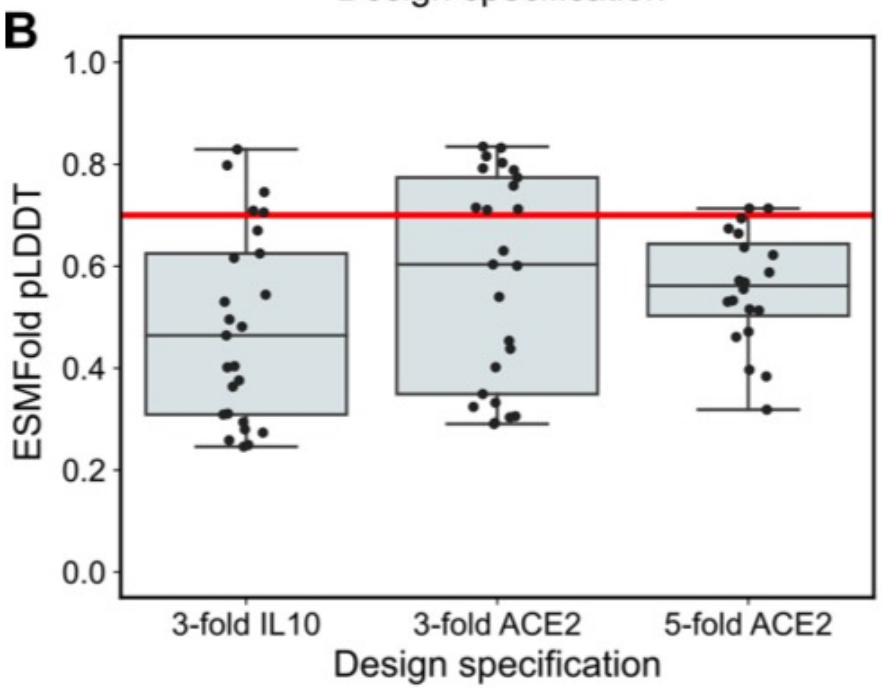
Experiments – Hierarchical constraints

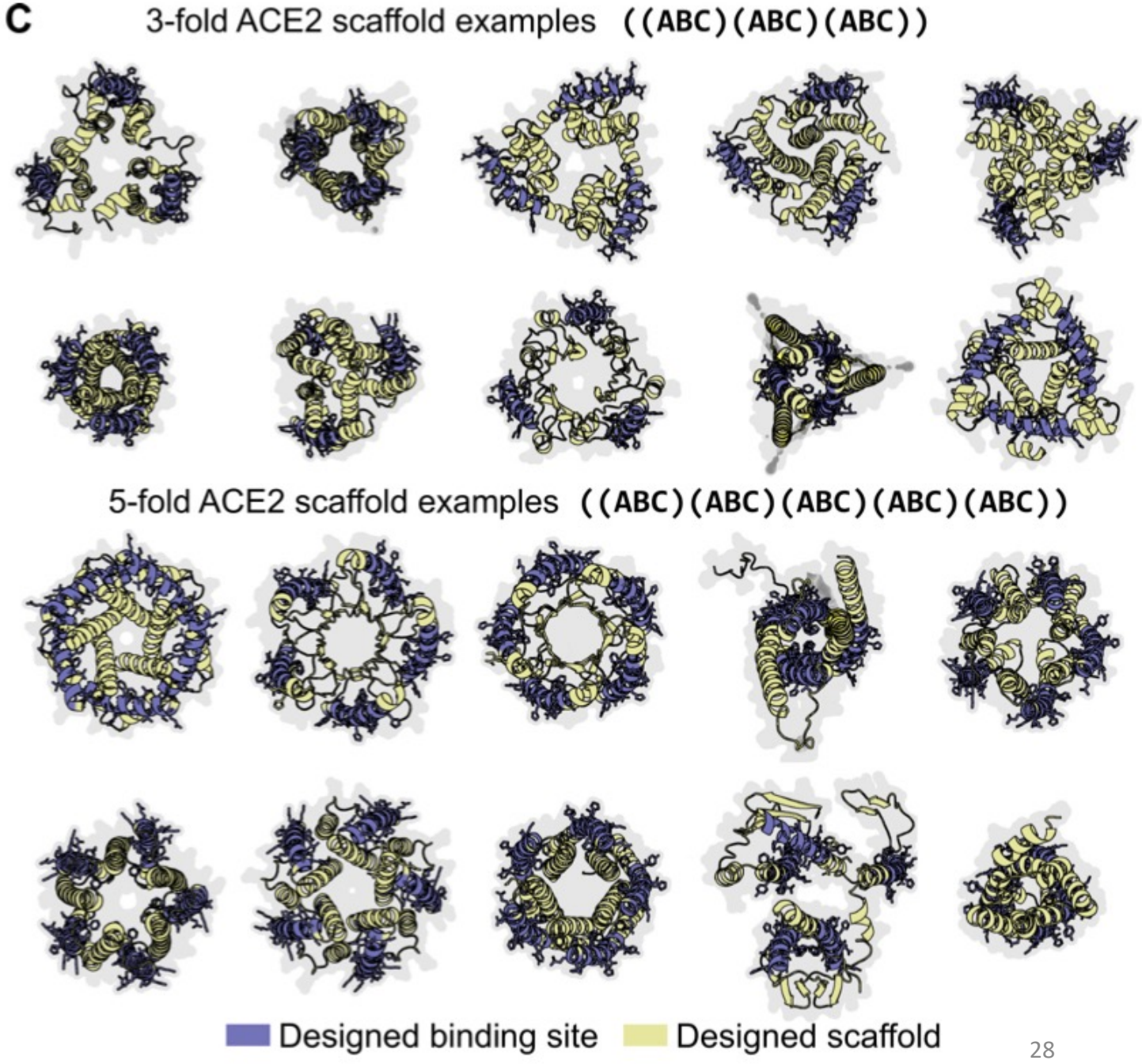
Symmetric functional cite scaffolding

- Constraints: weight 10 for cRMSD and dRMSD, 1 for others
- Ran simulated annealing over 30,000 iterations with a starting temperature of 1 over 20 seeds for the design of 3-fold scaffolds of the IL10 and ACE2 binding sites, as well as 20 seeds for the design of 5-fold scaffolds of the ACE2 binding site.




A
Mean binding site RMSD (Å)
1.1
1.0
0.9
0.8
0.7
0.6
0.5
3-fold IL10
3-fold ACE2
5-fold ACE2
Design specification

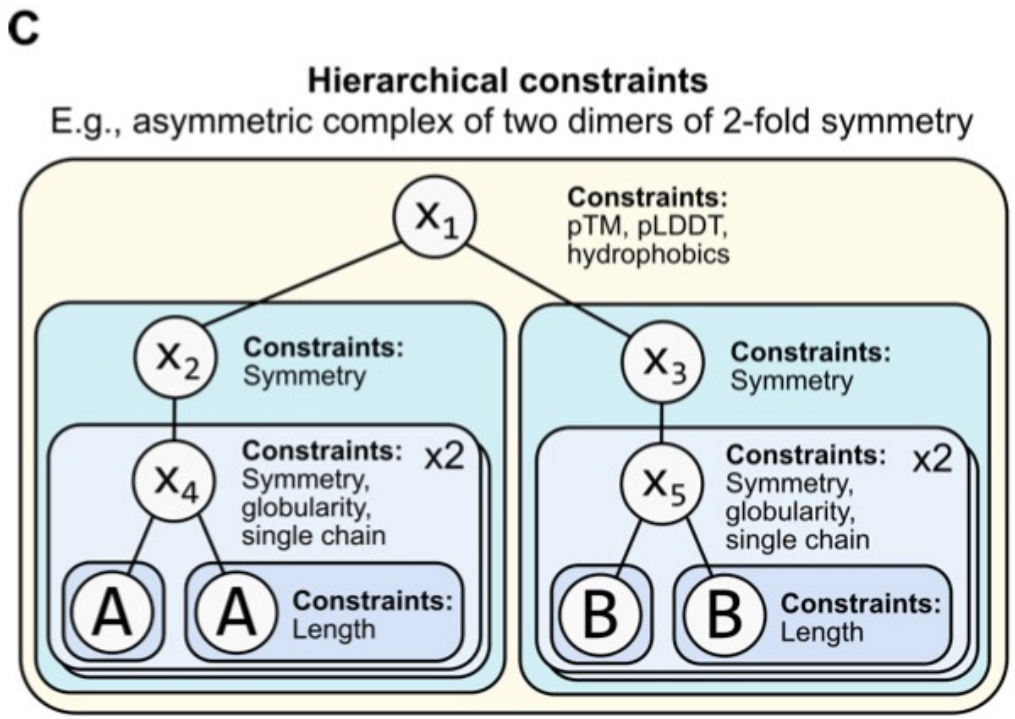

B
ESMFold pLDDT
1.0
0.8
0.6
0.4
0.2
0.0
3-fold IL10
3-fold ACE2
5-fold ACE2
Design specification

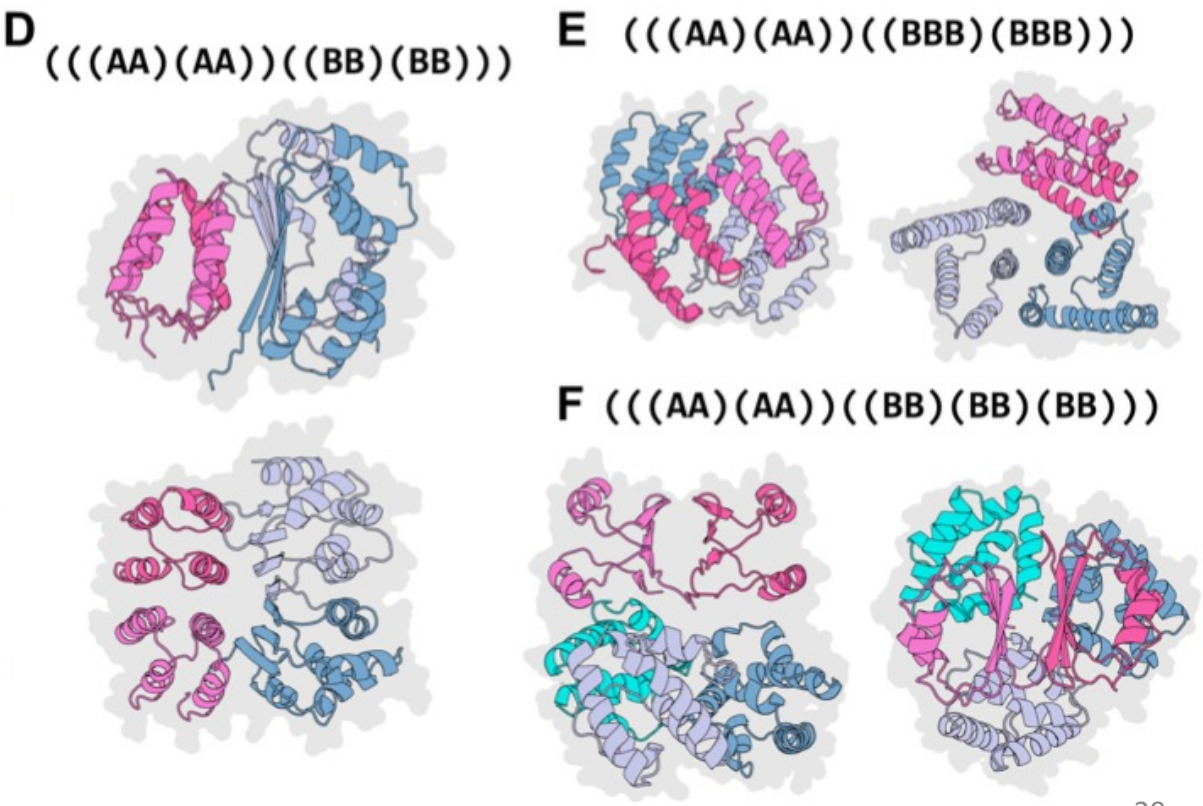

C
3-fold ACE2 scaffold examples ((ABC)(ABC)(ABC))
5-fold ACE2 scaffold examples ((ABC)(ABC)(ABC)(ABC)(ABC))
Designed binding site
Designed scaffold

Experiments – Hierarchical constraints

Hierarchical asymmetric symmetry design

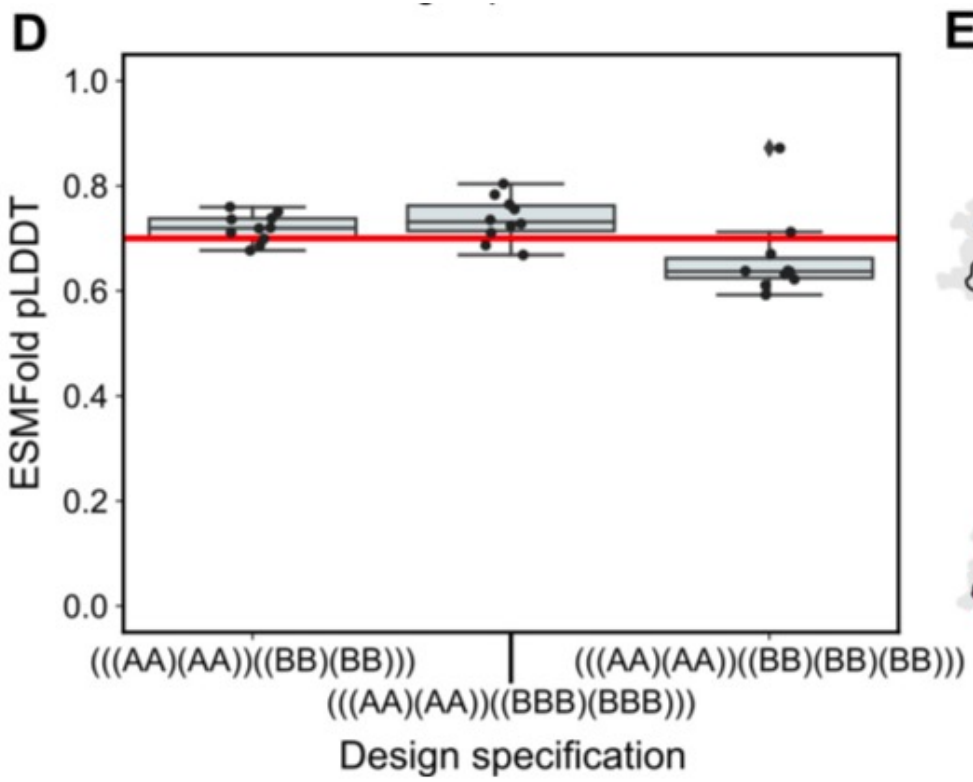
- Ran simulated annealing over 30,000 iterations with $T_{max} = 1$ over 10 seeds for each of the three programs.

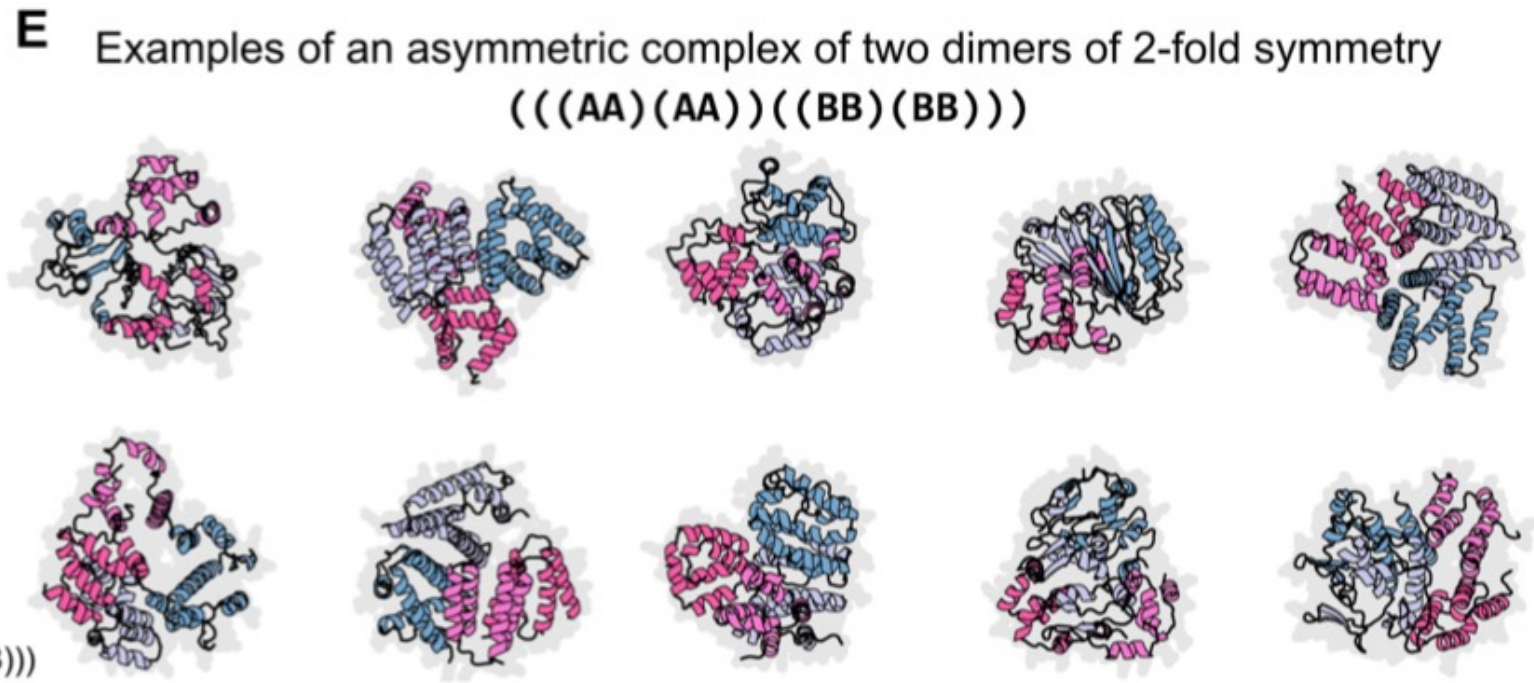




Experiments – Hierarchical constraints

Hierarchical asymmetric symmetry design





Conclusion

- Provide a high-level language for modular and programmable design of proteins
- Use ESMFold to convert constraints into an energy function which can be optimized by simulated annealing
- Demonstrate impressive protein design examples for programs with a wide range of complexity

Thanks!

Q & A