Predicting Structures of Proteins and Other Biomolecules

CS/CME/BioE/Biophys/BMI 279

Oct. 10, 15, 17 and 29, 2024

Ron Dror

Note: Assignment 2 released

- Due October 31, but get started early
  - More work than assignment 1
- Kickstart / code tutorial on Friday

Outline

- Why predict protein structure?
- Can we use (pure) physics-based methods?
- Knowledge-based methods
- Approaches to protein structure prediction (i.e., what information can we leverage?)
  - Template-based (“homology”) modeling (e.g., Phyre2)
  - *Ab initio* modeling (e.g., Rosetta)
  - Multiple sequence alignments (coevolution)
- Deep learning methods for protein structure prediction
  - First-generation deep learning methods: learning inter-residue distances from multiple sequence alignments
  - Second-generation deep learning methods: learning the entire structure
  - Large language models
- Predicting structures of other biomolecules and complexes

Why predict protein structure?

Problem definition

- Given the amino acid sequence of a protein, predict its three-dimensional structure
- Each protein adopts many structures. We want the *average* structure, which is roughly what's measured experimentally.
  - This will depend on experimental conditions: for example, is the protein bound to a drug and/or other molecules (and which ones)?

SVYDAAAQLTADVKKDLRDSW
KVIGSDKKGNGVALMTTLFAD
NQETIGYFKRLGNVSQGMAND
KLRGHSITLMYALQNFIDQLD
NPDSLDLVCS…….

→

Why predict protein structures rather than determining them experimentally?

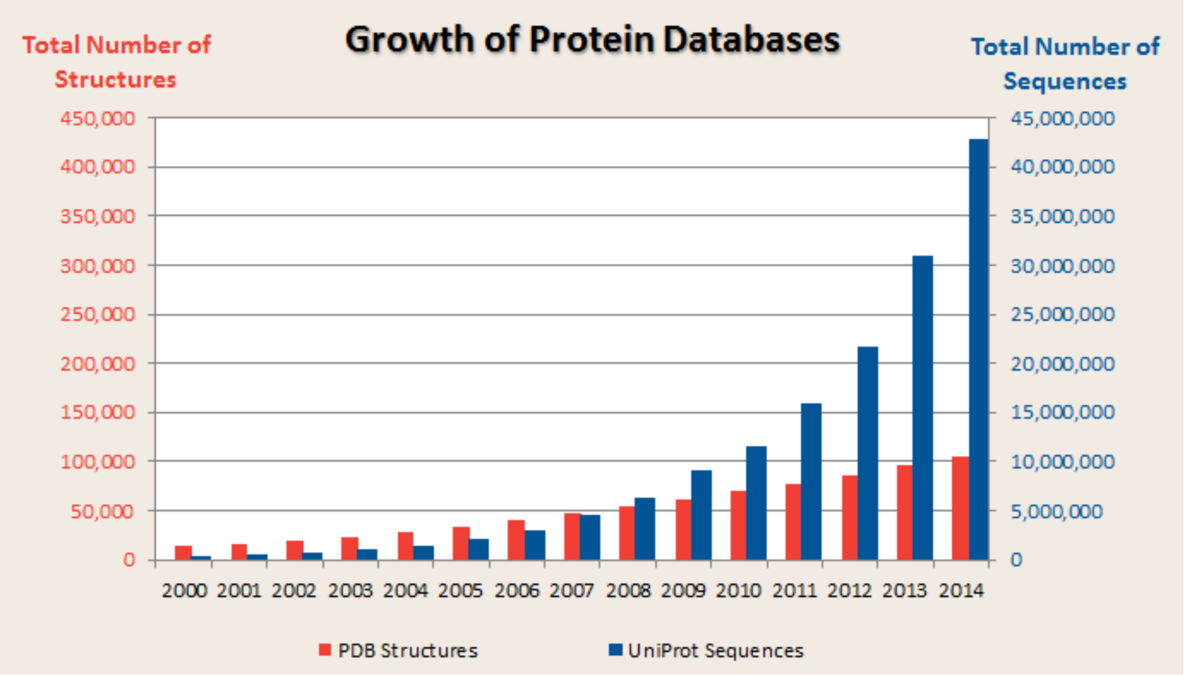
- Because predicting them computationally is (hopefully) cheaper and faster
- This answer is different from the answer to “why perform MD simulations?”
  - MD simulations are computationally expensive but often allow one to access information that simply can’t be observed with existing experimental methods

How are predicted structures used?

- Designing experiments
  - For example, what part of a protein should I cut out in order to make it easier to work with experimentally?
- Interpreting experimental data
  - For example, a computationally predicted approximate structure can help in determining an accurate structure experimentally, as we'll see later in this course
- Identifying the mechanism by which a protein functions
  - How does the protein work? If we think of a protein as a machine, knowing the structure of the machine can help us guess how the machine carries out its function (although we'd need much more information to be confident)
- Drug discovery
  - Computational screening of candidate drug compounds
  - Figuring out how to optimize a promising candidate compound
  - *Caveat:* Protein structure determination is rarely the hardest part of computational drug discovery, even when the structure must be determined experimentally

Why not just solve the structures experimentally?

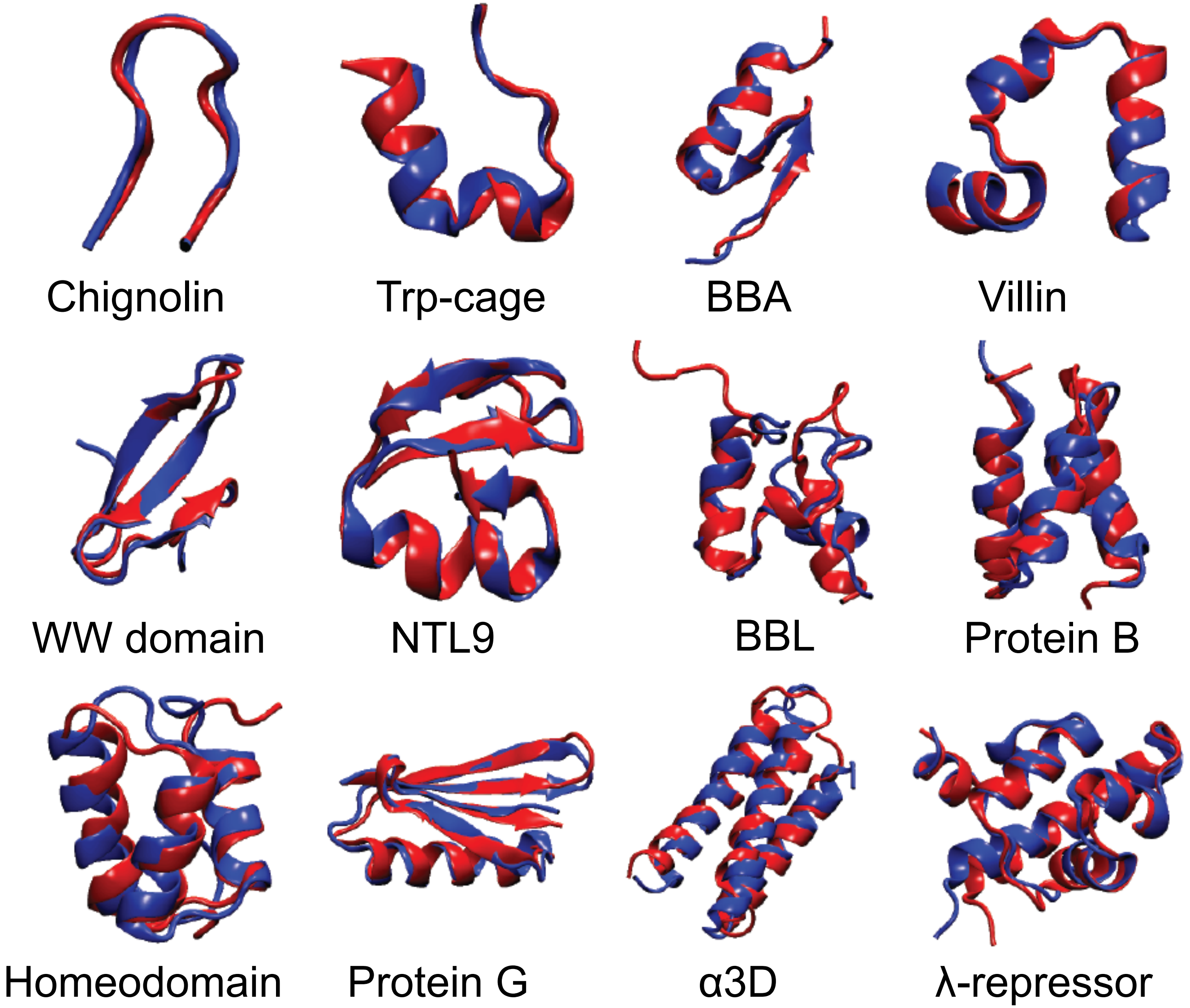
- Structures of certain proteins are very difficult to determine experimentally (although it's become much more tractable due to recent advances)
- Sequence determination far outpaces experimental structure determination
  - We already have far more sequences than experimental structures, and this gap will likely grow



http://www.dnastar.com/blog/wp-content/uploads/2015/08/ProteinDBGrowthBar3.png

Can we use (pure) physics-based methods?

Why not just simulate the folding process by molecular dynamics?



This is possible for some proteins.

Example: Simulation vs. experiment for 12 fast-folding proteins, up to 80 residues each

Lindorff-Larsen et al., *Science*, 2011

For most proteins, this doesn’t (yet) work

1. Folding timescales are usually much longer than simulation timescales.
2. Current molecular mechanics force fields aren’t always sufficiently accurate.
3. Disulfide bonds form during the real folding process. This is hard to mimic in simulation.

Simulating the folding *process* is important for understand how that process works (that is, how a protein gets from its unfolded state to its folded state—the original “protein folding problem”), but is *not* necessary to predict structure. Structure prediction is an easier problem (though still tough!).

Can we use (pure) physics-based methods?

- In principle, yes, but in practice, this is ***not*** the best way to predict protein structure
  - Instead, take advantage of the many known structures of proteins!

Knowledge-based methods

Basic idea behind knowledge-based (data-driven) methods

- The PDB contains about 200,000 experimentally determined protein structures
- Can we use that information to help us predict new structures?
- **Yes!**

We can also use the huge number of known, naturally occurring protein *sequences* (>250 million in the UniProt database, and >2 billion in the MGnify database)

http://www.duncanmalcolm.com/blog/startup-data-analytics-metric-

Questions for discussion

- If we want to predict the structure of protein X, how does knowing structures of other proteins help?
- If we want to predict the structure of protein X, how does knowing amino acid sequences of other proteins help (in particular, sequences of proteins whose structures we don’t know)?

Proteins with similar sequences tend to have similar structures

- Proteins with similar sequences tend to be homologs, meaning that they evolved from a common ancestor
- The fold of the protein (i.e., its overall structure) tends to be conserved during evolution
- This tendency is very strong. Even proteins with 15% sequence identity usually have similar structures.
  - **During evolution, sequence changes more quickly than structure**

For most human proteins, we can find a homolog with known structure

- As of 2017, 70% of human proteins—**and well over 90% of human drug targets**—had >30% sequence identity to a protein of known structure.
  - As of today, these numbers are even higher! They go up every year.
  - Structures with lower sequence identity (e.g., 20%) are still very informative
  - Most human proteins without a homolog of known structure probably don’t adopt a well-defined structure (“intrinsically disordered proteins”)

Somody et. al, Drug Discovery Today, 2017

What if we can’t identify a homolog in the PDB?

- We can still use information based on known structures
  - We can construct databases of observed structures of small fragments of a protein
  - We can use the PDB to build empirical, “knowledge-based” energy functions
- We can also extract substantial information from sequences of homologs whose structure has not been determined
  - Again, exploit the fact that proteins with similar sequence tend to have similar structure

Approaches to protein structure prediction (i.e., what information can we leverage?)

Why cover multiple approaches?

- Machine learning approaches (recent versions of AlphaFold, RoseTTAFold, etc.) now generally provide the best automated structure predictions.
- So why cover previous methods?
  - To better understand the sources of information on which more recent methods rely
  - To better understand topics covered later in the course (e.g., protein design)

Approaches to protein structure prediction

Template-based ("homology") modeling (e.g., Phyre2)

Template-based structure prediction: basic workflow

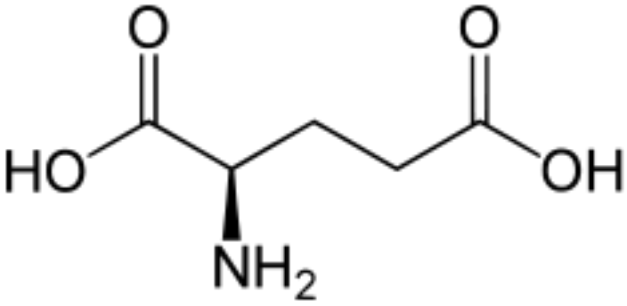
- User provides a *query* sequence with unknown structure
- Search the PDB for proteins with similar sequence and known structure. Pick the best match (the *template*).
- Build a model based on that template
  - One can also build a model based on multiple templates, where different templates are used for different parts of the protein.

What does it mean for two sequences to be similar?

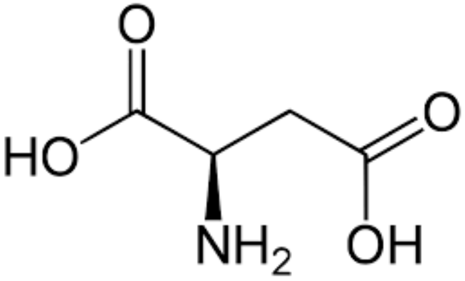
- Basic measure: count minimum number of amino acid residues one needs to change, add, or delete to get from one sequence to another
  - *Sequence identity*: amino acids that match exactly between the two sequences
  - Not trivial to compute for long sequences, but there are efficient dynamic programming algorithms to do so

What does it mean for two sequences to be similar?

- We can do better
  - Some amino acids are chemically similar to one another (example: glutamic acid and aspartic acid)
  - *Sequence similarity* is like sequence identity, but does not count changes between similar amino acids



Glutamic acid



Aspartic acid

What does it mean for two sequences to be similar?

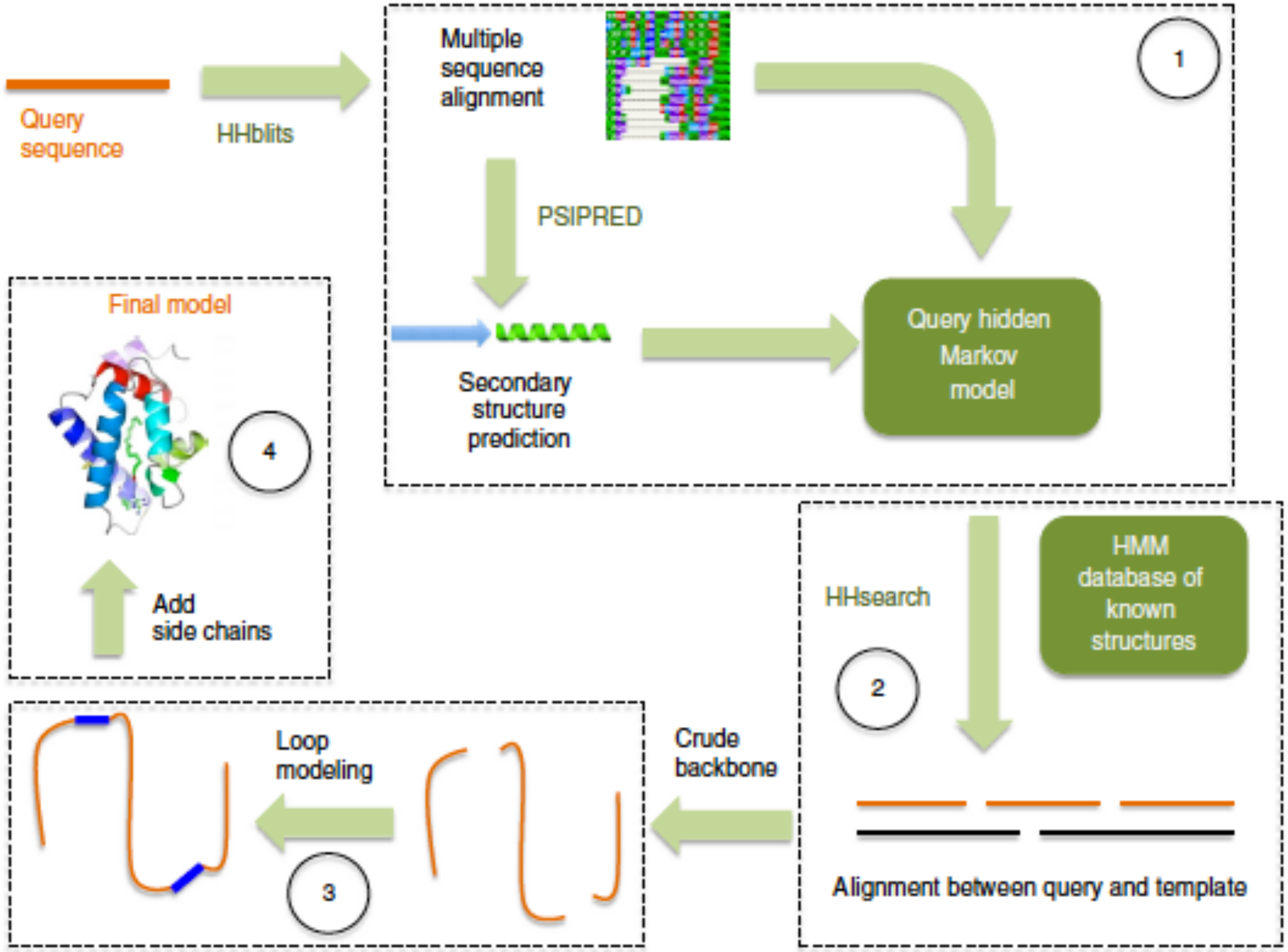
Optional material

- We can do even better
  - Once we’ve identified some homologs to a query sequence (i.e., similar sequences in the sequence database), we can create a *profile* describing the probability of mutation to each amino acid at each position
  - We can then use this profile to search for more homologs
  - Iterate between identification of homologs and profile construction
  - Measure similarity of two sequences by comparing their profiles
  - Often implemented using Hidden Markov Models (HMMs)
    - For example, the HHBlits software tool

We'll use the Phyre2 template-based modeling server as an example

- Try it out: http://www.sbg.bio.ic.ac.uk/phyre2/
- Why use Phyre2 as an example of template-based modeling?
  - Among the better automated structure prediction web-servers
  - Among the easiest to use
  - Approach is similar to that of other template-based modeling methods
  - Great name!

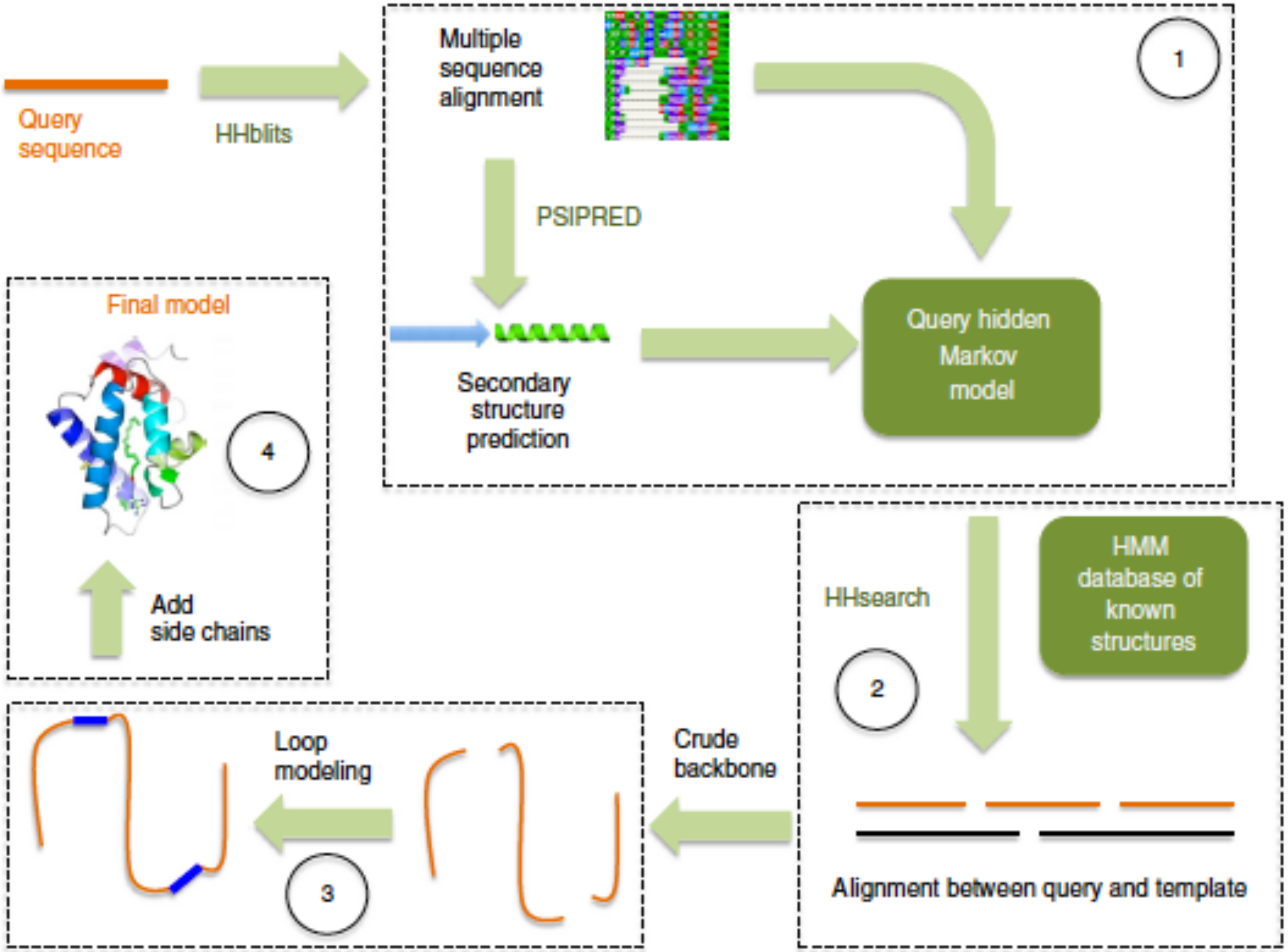
Phyre2 algorithmic pipeline


Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

LA Kelley et al., *Nature Protocols* 10:845 (2015)

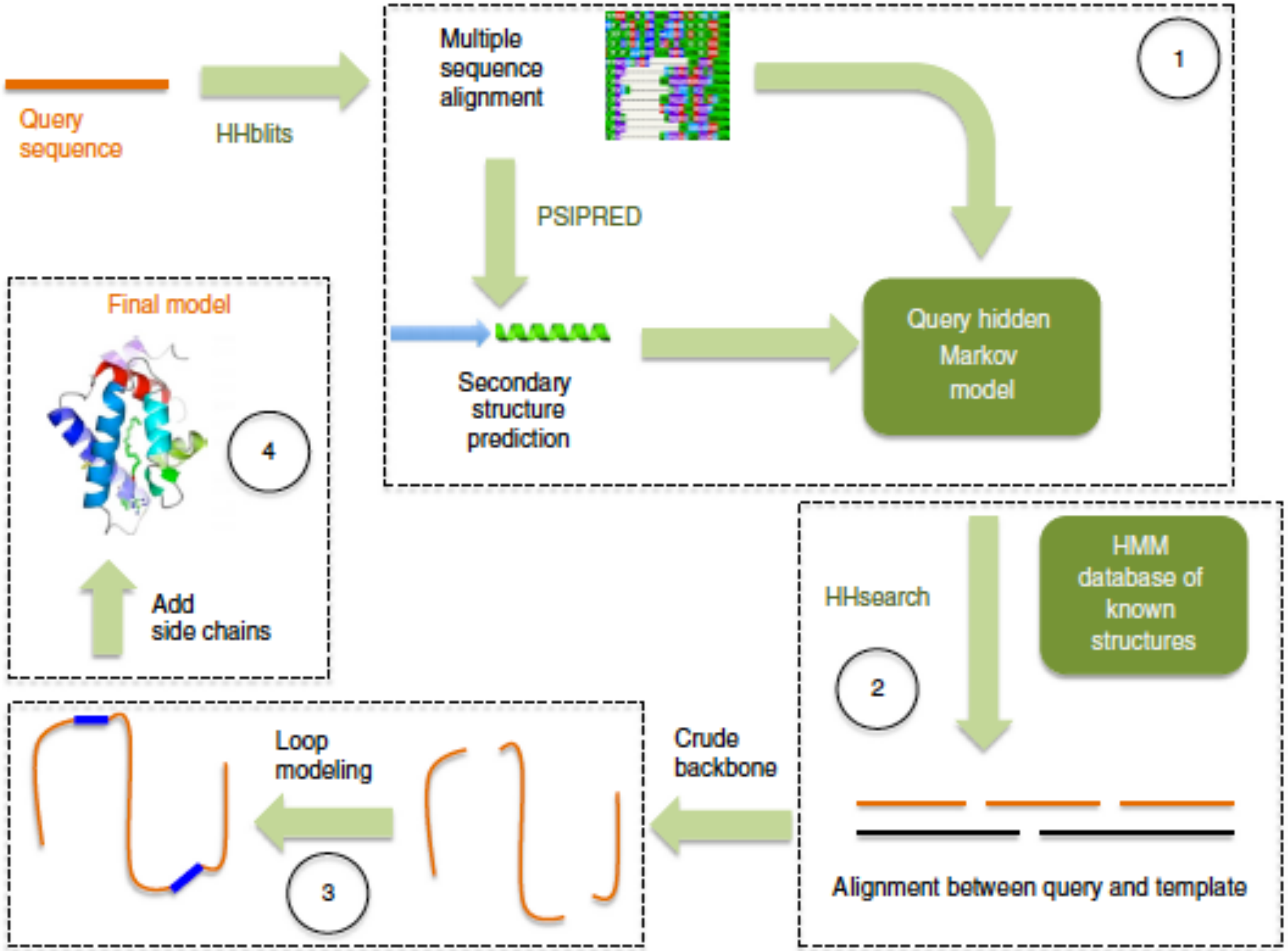
Phyre2 algorithmic pipeline

Identify similar sequences in protein sequence database



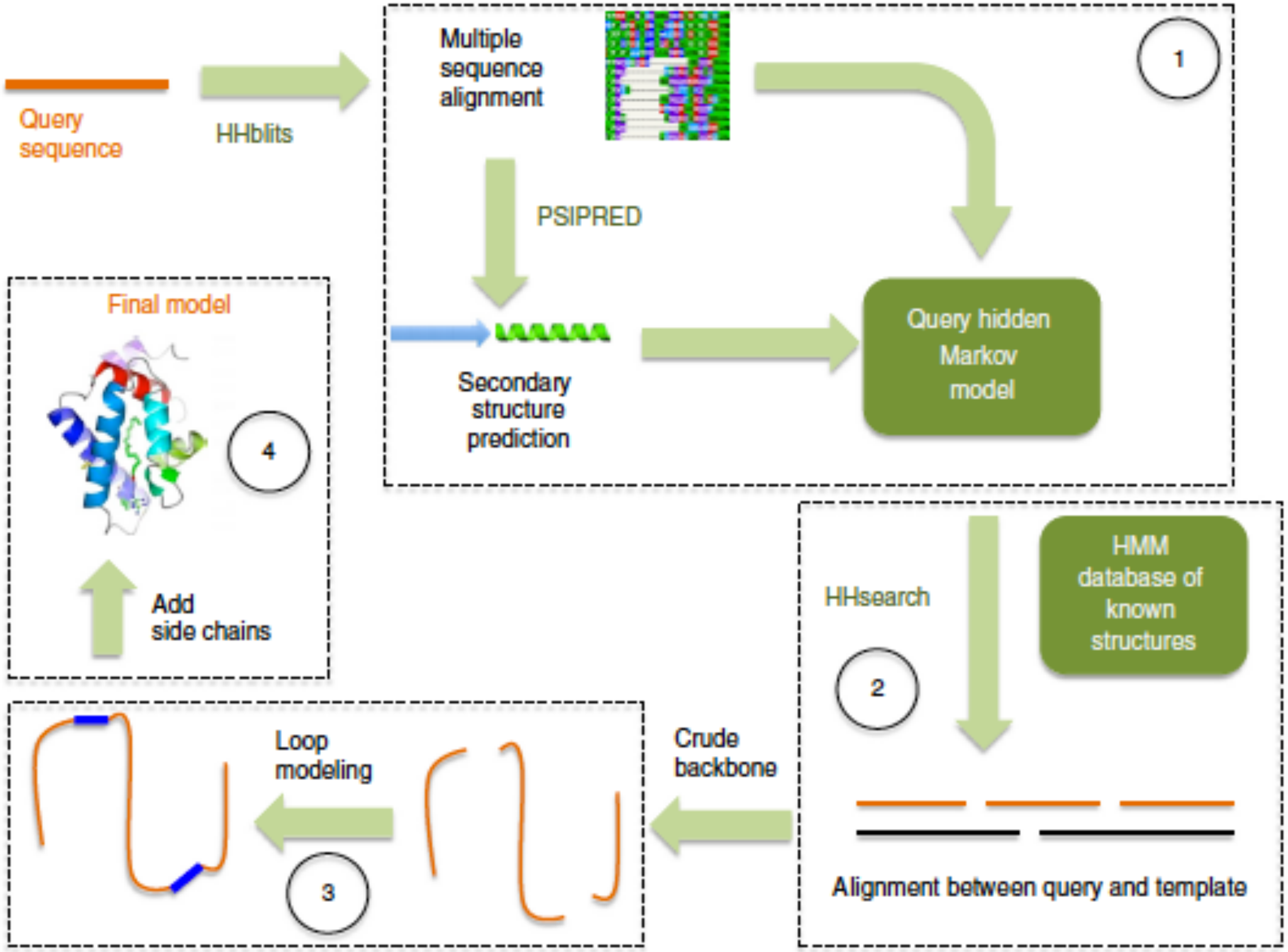
Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

Phyre2 algorithmic pipeline


Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Loop modeling
3
Crude backbone
Alignment between query and template

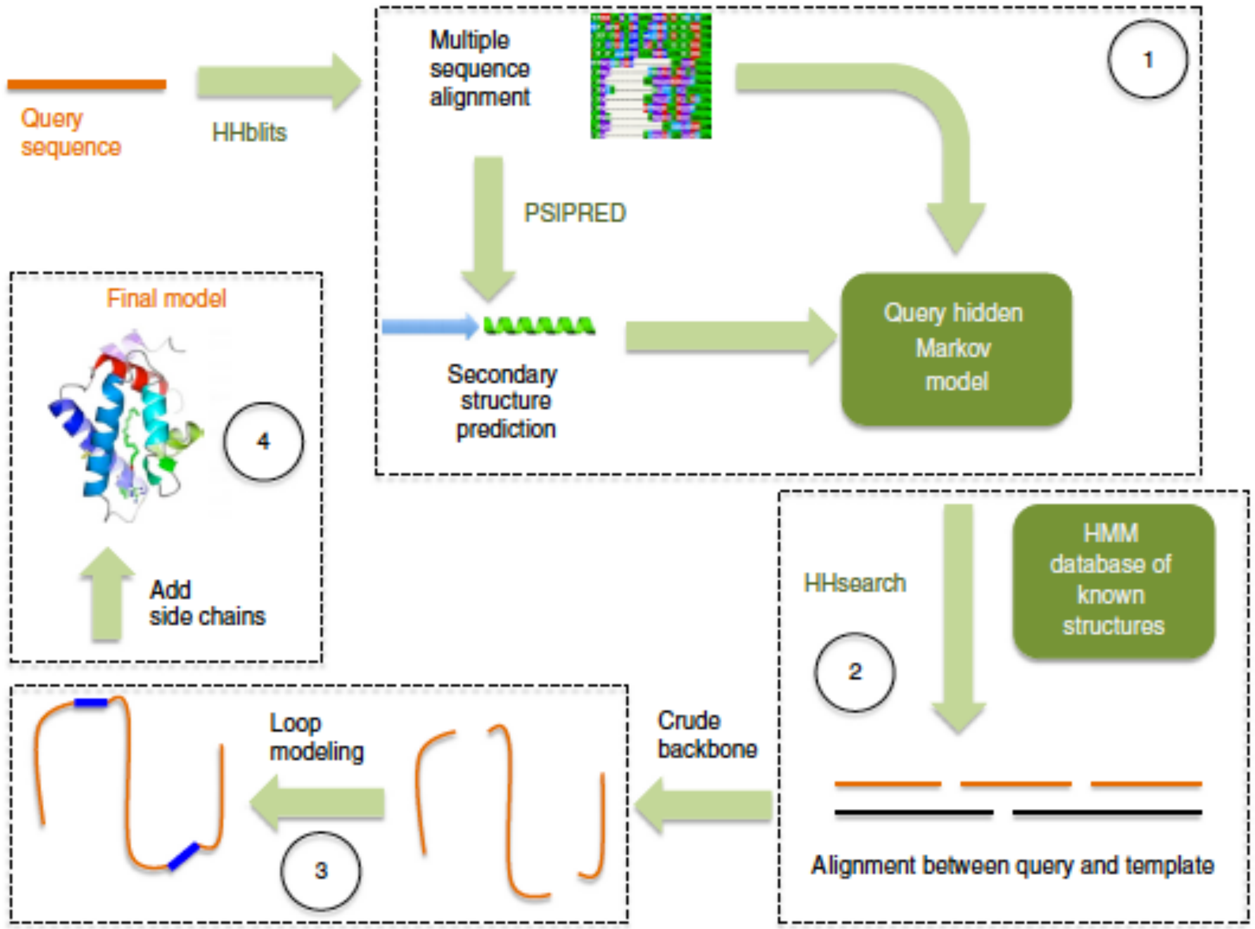
Choose a template structure by:
(1) comparing sequence profiles and
(2) predicting secondary structure for each residue in the query sequence and comparing to candidate template structures. Secondary structure (alpha helix, beta sheet, or neither) is predicted for segments of query sequence using a neural network trained on known structures.

Phyre2 algorithmic pipeline


Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

Compute optimal alignment of query sequence to template structure

Phyre2 algorithmic pipeline

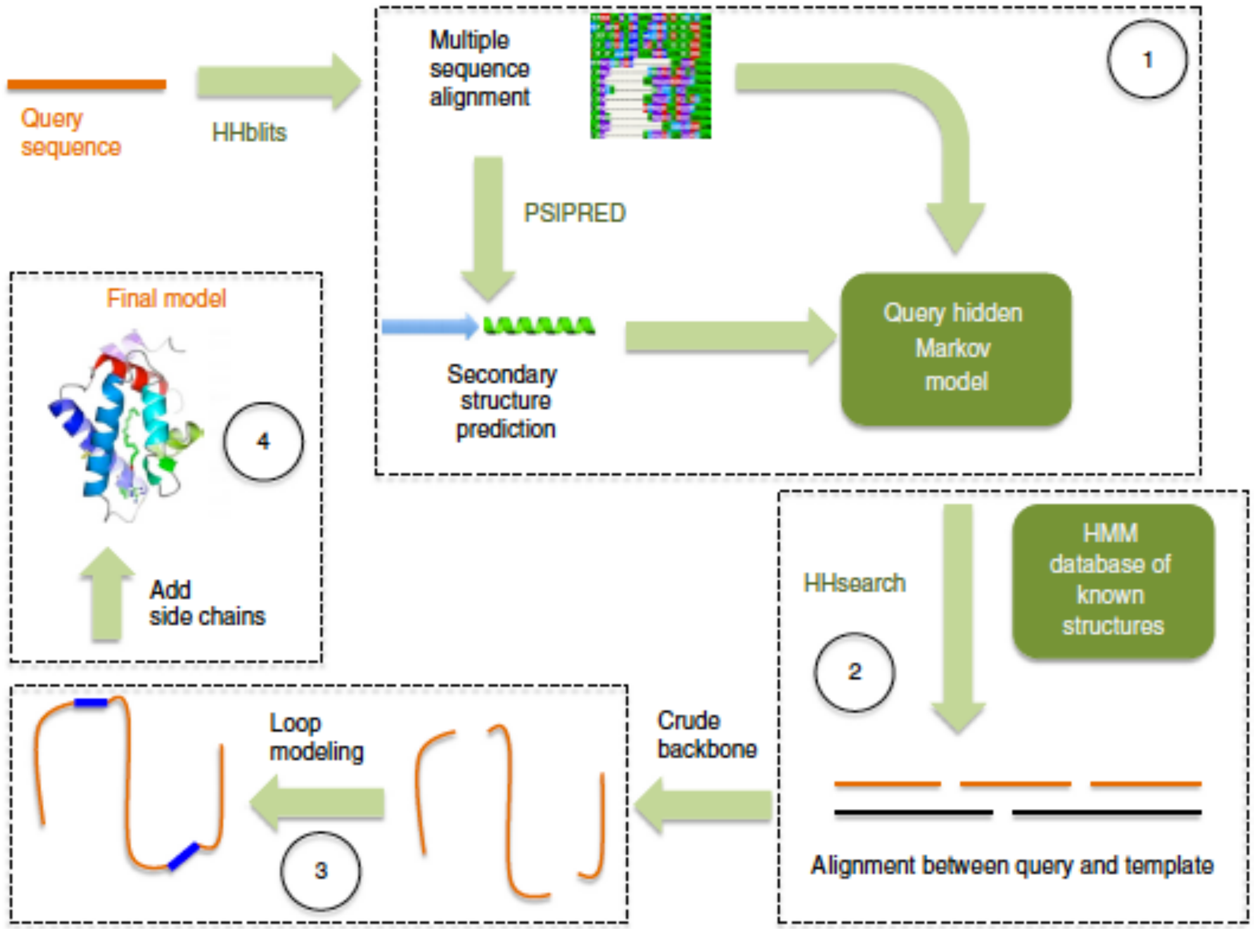

Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

Build a crude backbone model (no side chains) by simply superimposing corresponding amino acids.

Note: This model will be patched up and completed in subsequent steps, using local structural information from other structures in the PDB.

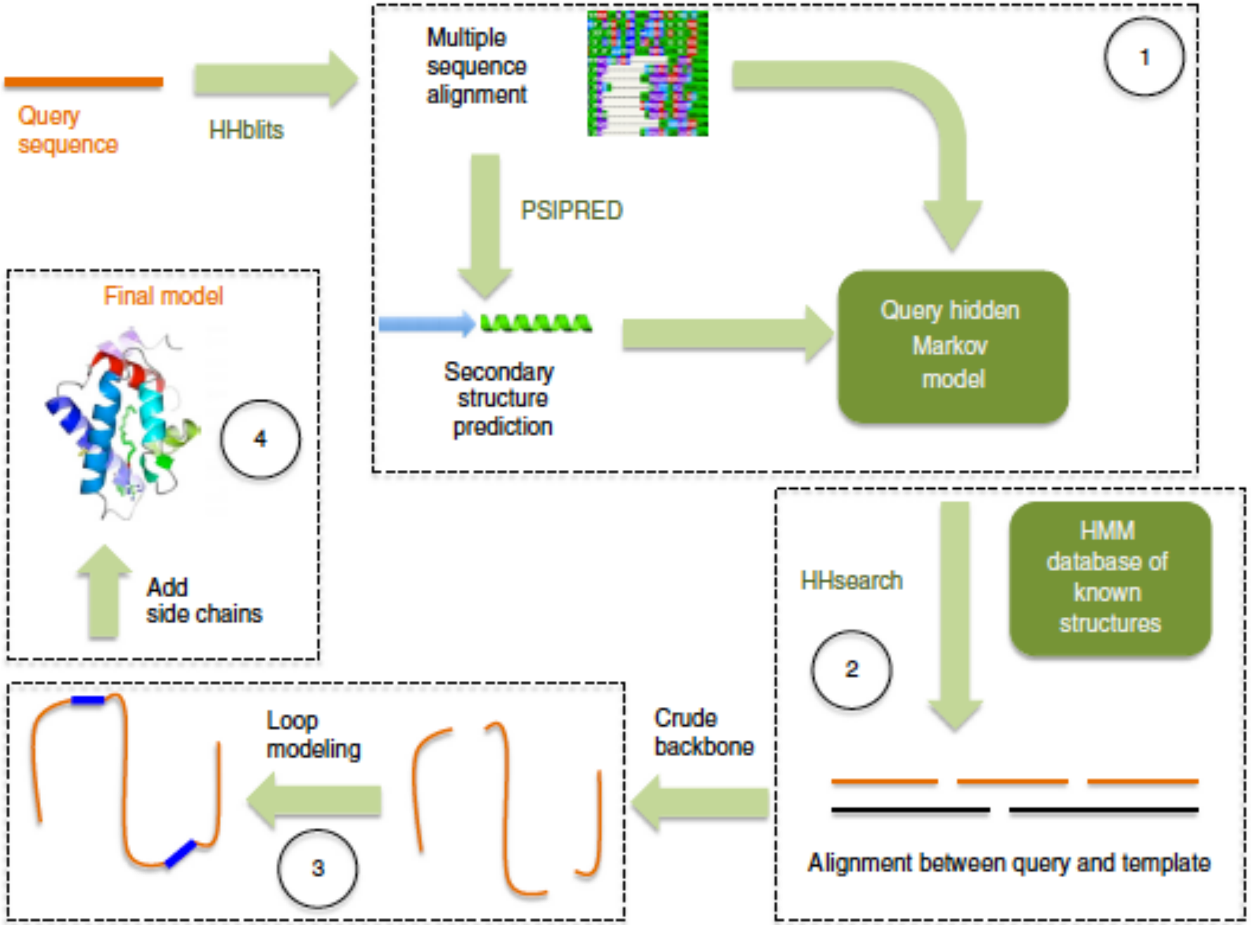
Optional material

Phyre2 algorithmic pipeline


Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

In the crude backbone models, some of the query residues will not be modeled, because they don’t have corresponding residues in the template (*insertions*). There will be some physical gaps in the modeled backbone, because some template residues don’t have corresponding query residues (*deletions*).

Phyre2 algorithmic pipeline

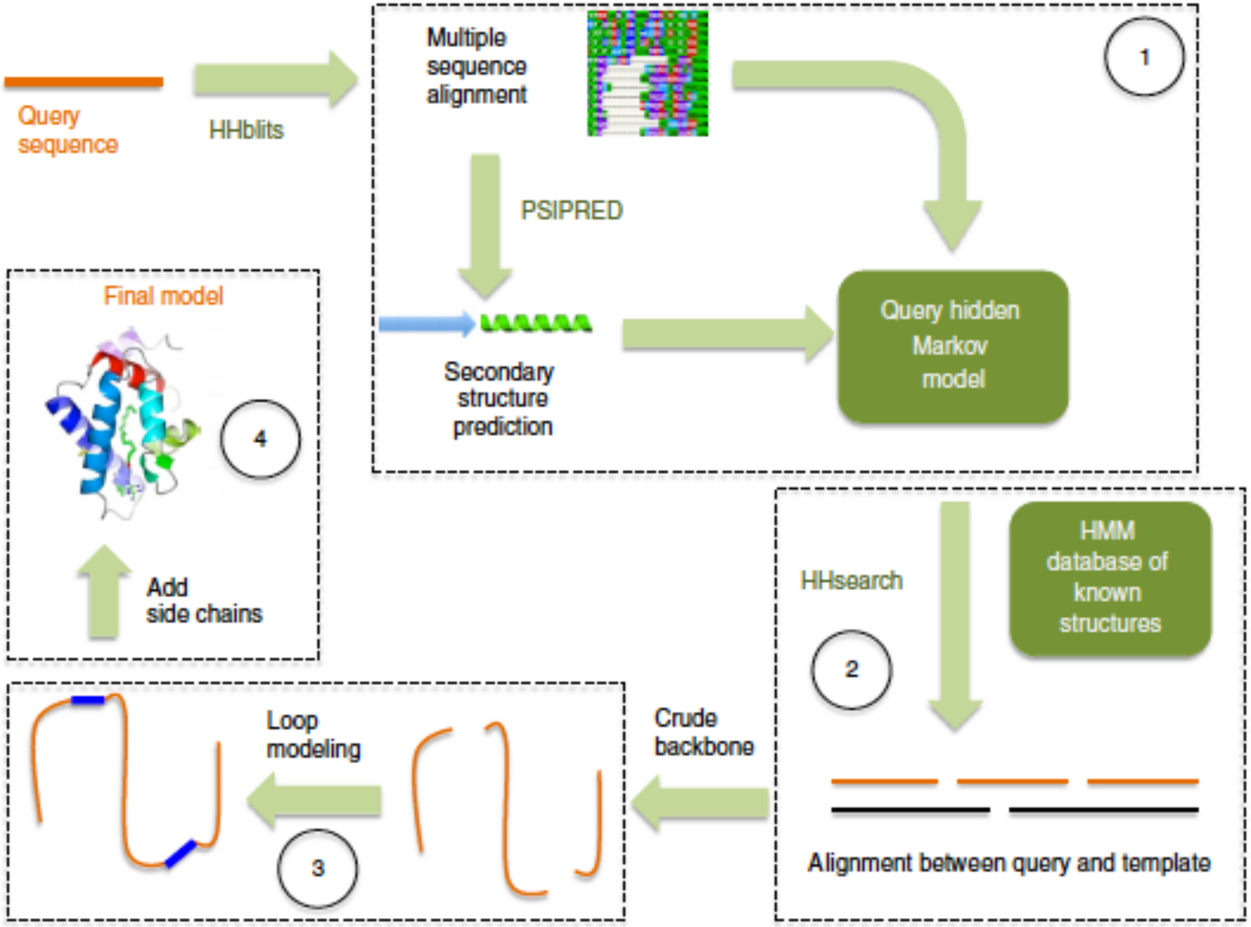

Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

Use *loop modeling* to patch up defects in the crude model due to insertions and deletions. For each insertion or deletion, search a large library of fragments (2–15 residues) of PDB structures for ones that match local sequence and fit the geometry best. Tweak backbone dihedrals within these fragments to make them fit better.

Optional material

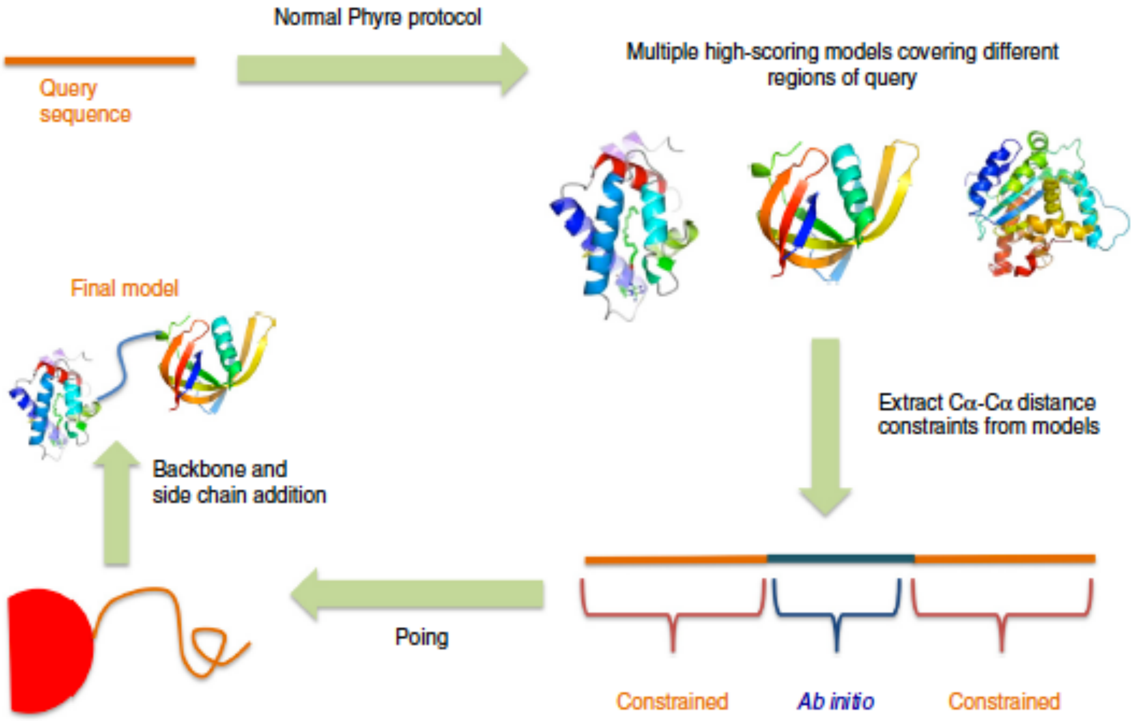
Phyre2 algorithmic pipeline

Add side chains. Use a database of commonly observed structures for each side chain (these structures are called *rotamers*). Search for combinations of rotamers that will avoid steric clashes (i.e., atoms ending up on top of one another).


Query sequence
HHblits
Multiple sequence alignment
1
PSIPRED
Secondary structure prediction
Query hidden Markov model
Final model
4
Add side chains
HHsearch
HMM database of known structures
2
Alignment between query and template
Crude backbone
Loop modeling
3

Modeling based on multiple templates

- In “intensive mode,” Phyre 2 will use multiple templates that cover (i.e., match well to) different parts of the query sequence.
  - Build a crude backbone model for each template
  - Extract distances between residues for “reliable” parts of each model
  - Perform a simplified protein folding simulation in which these distances are used as constraints. Additional constraints enforce predicted secondary structure
  - Fill in the side chains, as for single-template models


Normal Phyre protocol
Query sequence
Multiple high-scoring models covering different regions of query
Extract Cα-Cα distance constraints from models
Constrained
Ab initio
Constrained
Poing
Backbone and side chain addition
Final model

Poing: Synthesize from virtual ribosome. Springs for constraints. *Ab initio* modeling of missing regions. Backbone and side chain reconstruction.

LA Kelley et al., *Nature Protocols* 10:845 (2015)

Approaches to protein structure prediction

Ab initio modeling (e.g., Rosetta)

Ab initio structure prediction

- We define *ab initio* structure prediction methods as methods that do not exploit information about homologs of the query protein
- Also known as "*de novo* structure prediction"
- Many approaches proposed over time
- Probably the most successful is *fragment assembly*, as exemplified by the Rosetta software package

We'll use Rosetta as an example of *ab initio* structure prediction

- Software developed over the last 30 years by David Baker (U. Washington) and collaborators
- Software at: https://www.rosettacommons.org/software
- Structure prediction server: http://robetta.bakerlab.org/
- Why use Rosetta as an example?
  - Among the better *ab initio* modeling packages (for some years it was the best)
  - Approach is similar to that of many *ab initio* modeling packages
  - Rosetta provides a common framework that has become very popular for a wide range of molecular prediction and design tasks, especially protein design

Key ideas behind Rosetta

- Knowledge-based energy function
  - In fact, two of them:
    - The “Rosetta energy function,” which is coarse-grained (i.e., does not represent all atoms in the protein), is used in early stages of protein structure prediction
    - The “Rosetta all-atom energy function,” which depends on the position of every atom, is used in late stages
- A knowledge-based strategy for searching conformational space (i.e., the space of possible structures for a protein)
  - Fragment assembly forms the core of this method

Rosetta energy function

- At first this was the only energy function used by Rosetta (hence the name)
- Based on a simplified representation of protein structure:
  - Do not explicitly represent solvent (e.g., water)
  - Assume all bond lengths and bond angles are fixed
  - Represent the protein backbone using torsion angles (three per amino acid: Φ, Ψ, ω)
  - Represent side chain position using a single “centroid,” located at the side chain’s center of mass

Rosetta energy function

Optional material, of course!

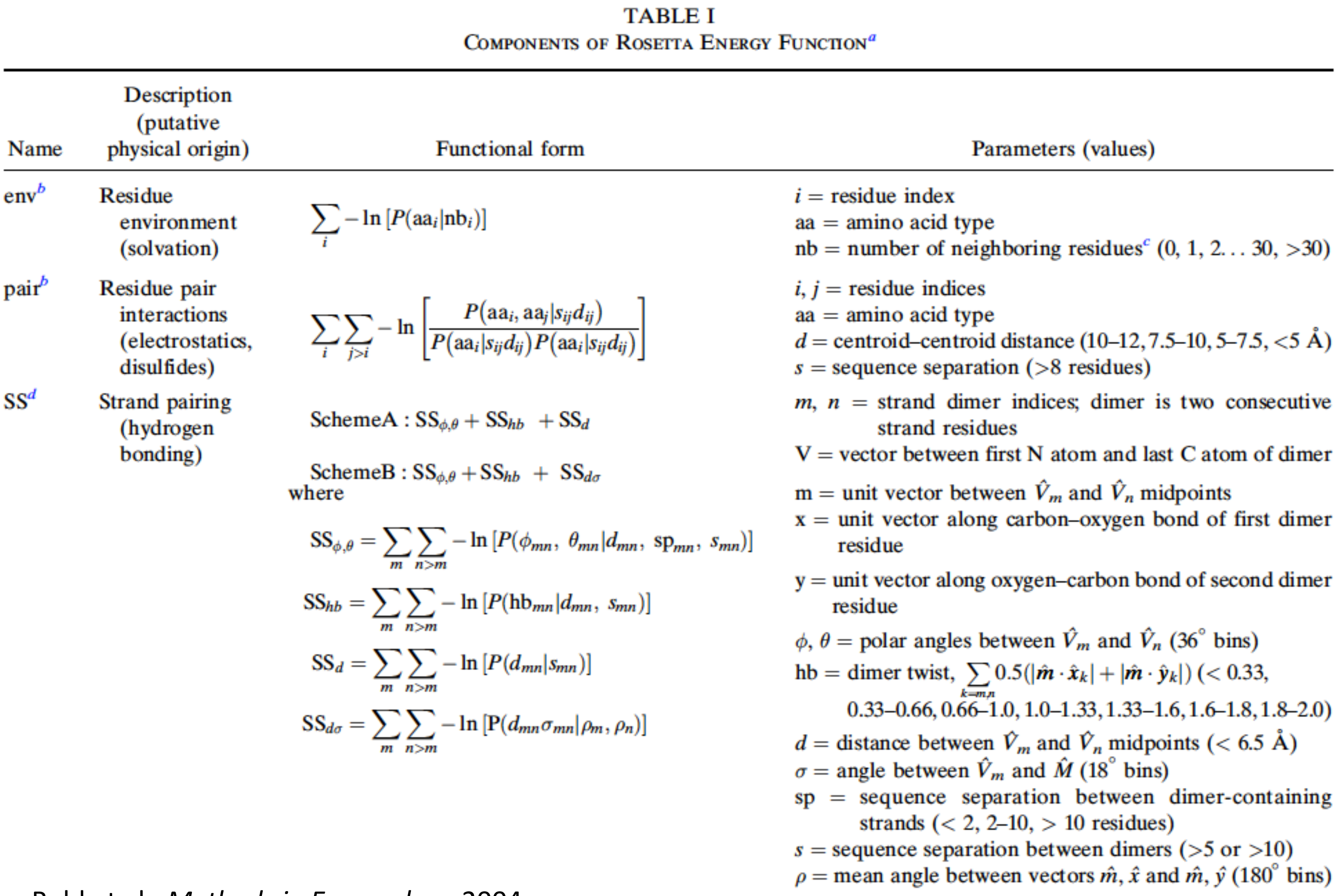


TABLE I
COMPONENTS OF ROSETTA ENERGY FUNCTION[a]

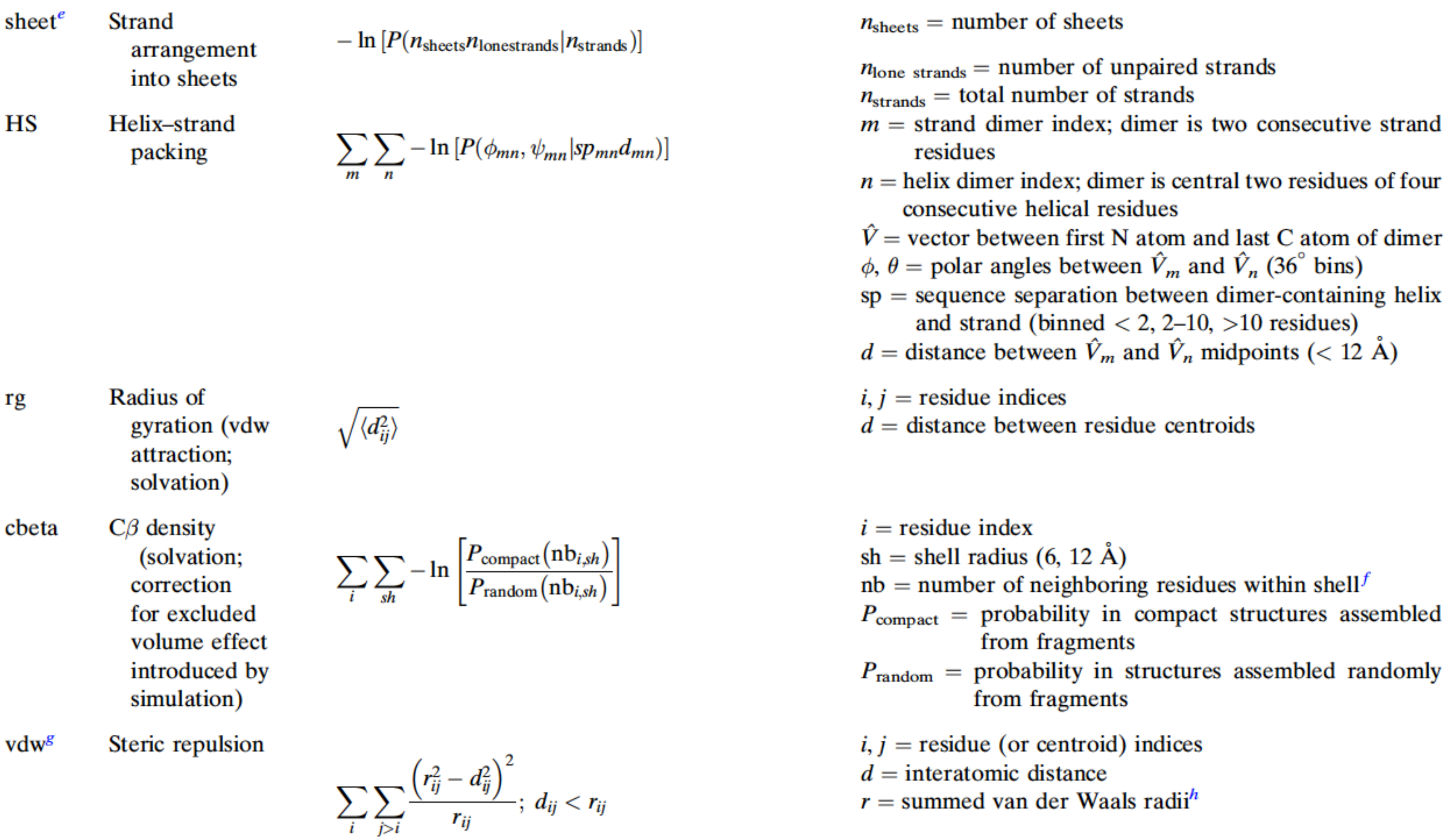
Name	Description (putative physical origin)	Functional form	Parameters (values)
env[b]	Residue environment (solvation)	$\sum_i -\ln\left[P(\mathrm{aa}_i \vert \mathrm{nb}_i)\right]$	i = residue index aa = amino acid type nb = number of neighboring residues[c] (0, 1, 2… 30, >30)
pair[b]	Residue pair interactions (electrostatics, disulfides)	$\sum_i \sum_{j>i} -\ln\left[\frac{P(\mathrm{aa}_i, \mathrm{aa}_j \vert s_{ij} d_{ij})}{P(\mathrm{aa}_i \vert s_{ij} d_{ij}) P(\mathrm{aa}_i \vert s_{ij} d_{ij})}\right]$	i, j = residue indices aa = amino acid type d = centroid–centroid distance (10–12, 7.5–10, 5–7.5, <5 Å) s = sequence separation (>8 residues)
SS[d]	Strand pairing (hydrogen bonding)	SchemeA : $\mathrm{SS}_{\phi,\theta} + \mathrm{SS}_{hb} + \mathrm{SS}_d$ SchemeB : $\mathrm{SS}_{\phi,\theta} + \mathrm{SS}_{hb} + \mathrm{SS}_{d\sigma}$ where $\mathrm{SS}_{\phi,\theta} = \sum_m \sum_{n>m} -\ln\left[P(\phi_{mn}, \theta_{mn} \vert d_{mn}, \mathrm{sp}_{mn}, s_{mn})\right]$ $\mathrm{SS}_{hb} = \sum_m \sum_{n>m} -\ln\left[P(\mathrm{hb}_{mn} \vert d_{mn}, s_{mn})\right]$ $\mathrm{SS}_d = \sum_m \sum_{n>m} -\ln\left[P(d_{mn} \vert s_{mn})\right]$ $\mathrm{SS}_{d\sigma} = \sum_m \sum_{n>m} -\ln\left[\mathrm{P}(d_{mn}\sigma_{mn} \vert \rho_m, \rho_n)\right]$	m, n = strand dimer indices; dimer is two consecutive strand residues V = vector between first N atom and last C atom of dimer m = unit vector between $\hat{V}_m$ and $\hat{V}_n$ midpoints x = unit vector along carbon–oxygen bond of first dimer residue y = unit vector along oxygen–carbon bond of second dimer residue ϕ, θ = polar angles between $\hat{V}_m$ and $\hat{V}_n$ (36° bins) hb = dimer twist, $\sum_{k=m,n} 0.5(\vert\hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{x}}_k\vert + \vert\hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{y}}_k\vert)$ (< 0.33, 0.33–0.66, 0.66–1.0, 1.0–1.33, 1.33–1.6, 1.6–1.8, 1.8–2.0) d = distance between $\hat{V}_m$ and $\hat{V}_n$ midpoints (< 6.5 Å) σ = angle between $\hat{V}_m$ and $\hat{M}$ (18° bins) sp = sequence separation between dimer-containing strands (< 2, 2–10, > 10 residues) s = sequence separation between dimers (>5 or >10) ρ = mean angle between vectors $\hat{m}, \hat{x}$ and $\hat{m}, \hat{y}$ (180° bins)

From Rohl et al., *Methods in Enzymology* 2004

You're not responsible for the details!

Rosetta energy function

Optional material, of course!



sheet[e]	Strand arrangement into sheets	$-\ln[P(n_{\text{sheets}} n_{\text{lonestrands}} \mid n_{\text{strands}})]$	n_{sheets} = number of sheets $n_{\text{lone strands}}$ = number of unpaired strands n_{strands} = total number of strands
HS	Helix–strand packing	$\sum_m \sum_n -\ln[P(\phi_{mn}, \psi_{mn} \mid sp_{mn} d_{mn})]$	m = strand dimer index; dimer is two consecutive strand residues n = helix dimer index; dimer is central two residues of four consecutive helical residues $\hat{V}$ = vector between first N atom and last C atom of dimer ϕ, θ = polar angles between $\hat{V}_m$ and $\hat{V}_n$ (36° bins) sp = sequence separation between dimer-containing helix and strand (binned < 2, 2–10, >10 residues) d = distance between $\hat{V}_m$ and $\hat{V}_n$ midpoints (< 12 Å)
rg	Radius of gyration (vdw attraction; solvation)	$\sqrt{\langle d_{ij}^2 \rangle}$	i, j = residue indices d = distance between residue centroids
cbeta	Cβ density (solvation; correction for excluded volume effect introduced by simulation)	$\sum_i \sum_{sh} -\ln\left[\frac{P_{\text{compact}}(\text{nb}_{i,sh})}{P_{\text{random}}(\text{nb}_{i,sh})}\right]$	i = residue index sh = shell radius (6, 12 Å) nb = number of neighboring residues within shell[f] P_{compact} = probability in compact structures assembled from fragments P_{random} = probability in structures assembled randomly from fragments
vdw[g]	Steric repulsion	$\sum_i \sum_{j>i} \frac{\left(r_{ij}^2 - d_{ij}^2\right)^2}{r_{ij}};\ d_{ij} < r_{ij}$	i, j = residue (or centroid) indices d = interatomic distance r = summed van der Waals radii[h]

From Rohl et al., *Methods in Enzymology* 2004

You're not responsible for the details!

This list of terms is incomplete, as more have been added

Updated version with more terms: Alford et al., *Journal of Chemical Theory and Computation, 2017*

Rosetta energy function: take-aways

- The (coarse-grained) Rosetta energy function is essentially entirely knowledge-based
  - Based on statistics computed from experimentally determined protein structures in the PDB
- Many of the terms are of the form –ln[*P*(*A*)] (that is, –$\log_e$[*P*(*A*)]), where *P*(*A*) is the probability of some conformational state *A*
  - This is essentially the free energy of *A*. Recall definition of free energy:

$$G_A = -k_B T \log_e \left(P(A) \right) \qquad P(A) = \exp\left({-G_A}/{k_B T} \right)$$

Rosetta all-atom energy function

- Still makes simplifying assumptions:
  - Do not explicitly represent solvent (e.g., water)
  - Assume all bond lengths and bond angles are fixed
- Functional forms are a hybrid between molecular mechanics force fields and the (coarse-grained) Rosetta energy function
  - Partly physics-based, partly knowledge-based

Are these potential energy functions or free energy functions?

- The molecular mechanics force fields discussed in previous lectures are potential energy functions
- One can also attempt to construct a free energy function, where the energy associated with a conformation is the free energy of the set of “similar” conformations (for some definition of “similar”)
- The Rosetta energy functions are approximate free energy functions (despite sometimes being referred to as potential energy functions)
  - This means that searching for the “minimum” energy is more valid (as a way to determine structure)
  - Nevertheless, typical protocol is to repeat the search process many times, cluster the results, and report the largest cluster as the solution. This rewards wider and deeper wells.

How does Rosetta search the conformational space?

- Challenge: considering every possible conformation would take far too long!
- Key ideas behind solution:
  - Use a Monte Carlo search strategy
  - Exploit the fact that each small piece (“fragment”) of the protein likely adopts a structure similar to that of a fragment from some previously determined protein structure

How does Rosetta search the conformational space?

- Two steps:
  - Coarse search: fragment assembly
  - Refinement
- Perform coarse search many times, and then perform refinement on each result

Coarse search: fragment assembly

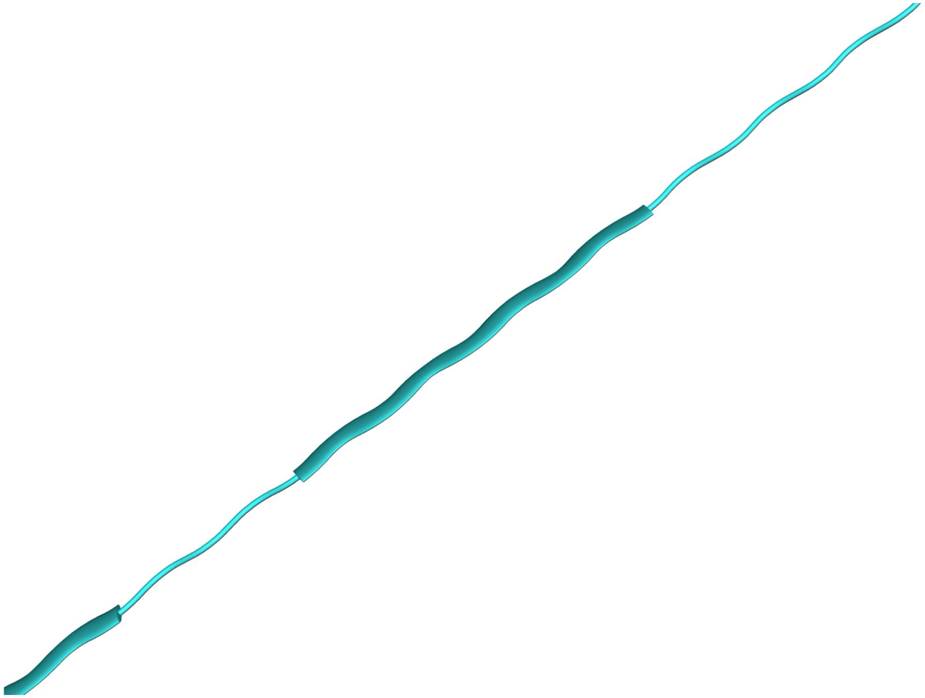
- Uses a large database of 3-residue and 9-residue fragments, taken from structures in the PDB
- Monte Carlo sampling algorithm proceeds as follows:
    1. Start with the protein in an extended conformation
    2. Randomly select a 3-residue or 9-residue section
    3. Find a fragment in the library whose sequence resembles it
    4. Consider a move in which the backbone dihedrals of the selected section are replaced by those of the fragment. Calculate the effect on the entire protein structure.
    5. Evaluate the Rosetta energy function before and after the move
    6. Use the Metropolis criterion to accept or reject the move
    7. Return to step 2
- The real search algorithm adds some bells and whistles

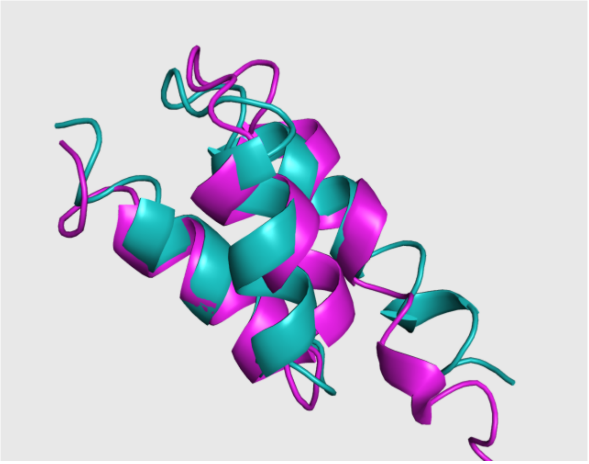
Refinement

- Refinement is performed using the Rosetta all-atom energy function, after building in side chains
- Refinement involves a combination of Monte Carlo moves and energy minimization
- The Monte Carlo moves are designed to perturb the structure much more gently than those used in the coarse search
  - Many still involve the use of fragments

Example: structure prediction by Rosetta

- Fragment assembly for a small protein





Final conformation from Rosetta fragment assembly

Experimentally determined structure

Note: This is not a full Rosetta structure prediction — just initial steps (doesn’t include refinement, multiple simulations, etc.)

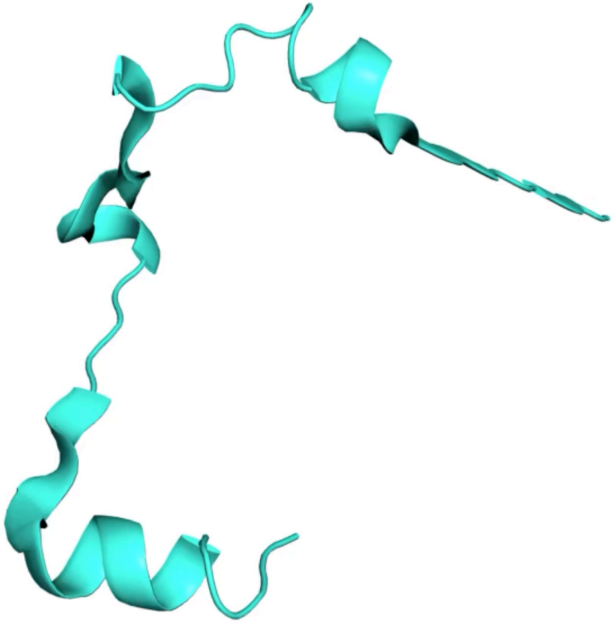
Hyun Soo Jeon

Example: structure prediction by Rosetta

- During Monte Carlo sampling, energy usually decreases but sometimes increases

Rosetta energy

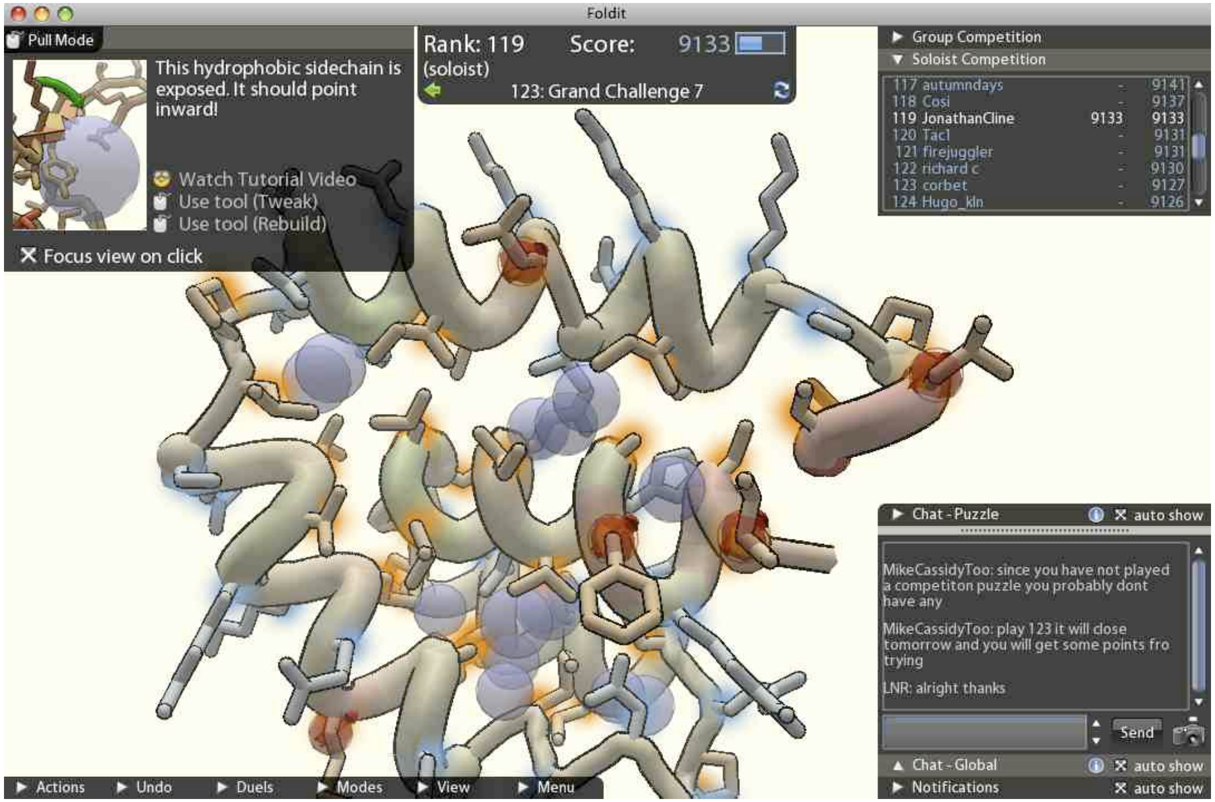
79.126



Hyun Soo Jeon

FoldIt: Protein-folding game

- https://fold.it/
- Basic idea: allow players to optimize the Rosetta all-atom energy function
  - Game score is negative of the energy (plus a constant)



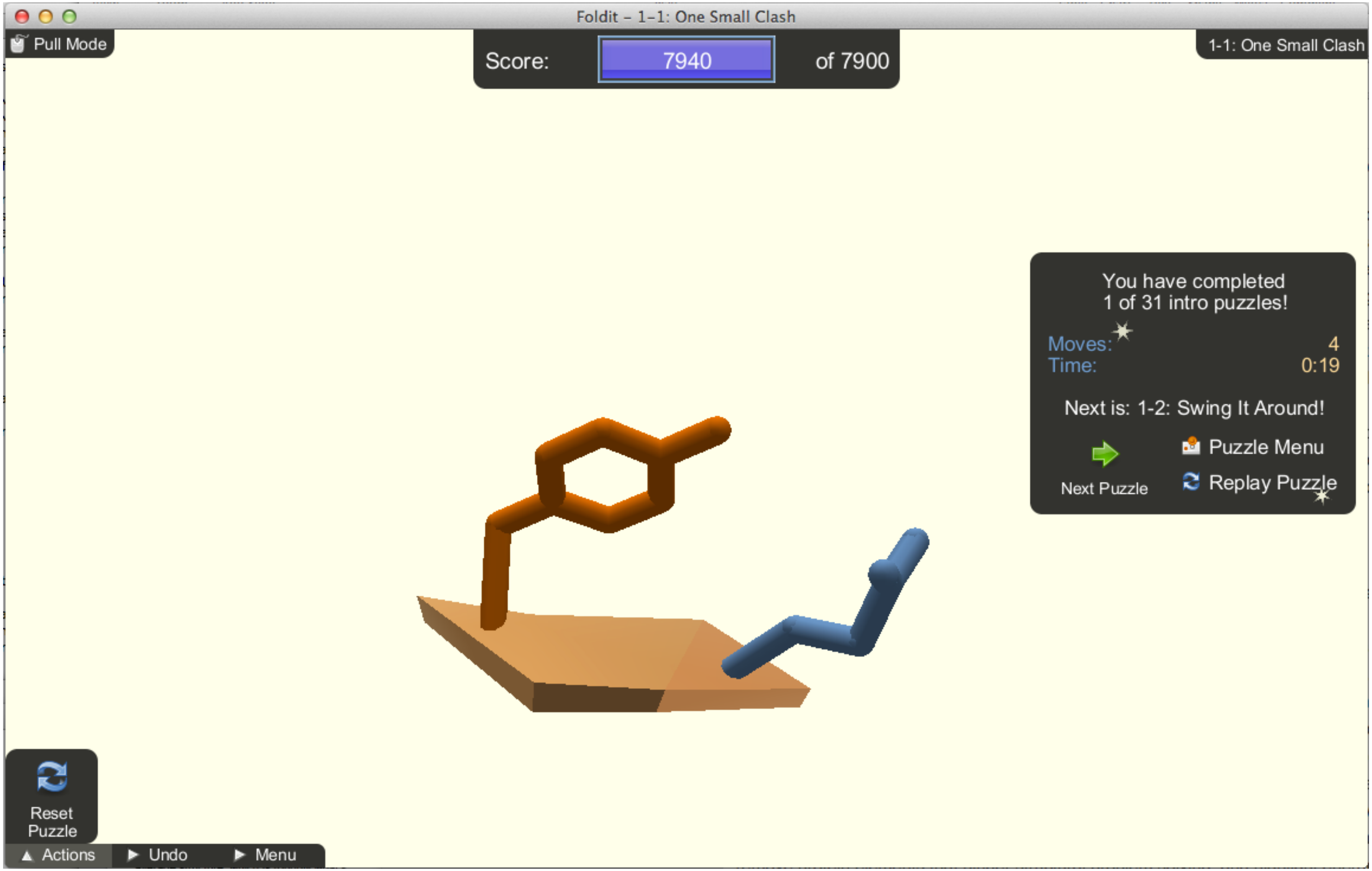
This hydrophobic sidechain is exposed. It should point inward!

117	autumndays	-	9141
118	Cosi	-	9137
119	JonathanCline	9133	9133
120	Tacl	-	9131
121	firejuggler	-	9131
122	richard c	-	9130
123	corbet	-	9127
124	Hugo_kln	-	9126

MikeCassidyToo: since you have not played a competiton puzzle you probably dont have any

MikeCassidyToo: play 123 it will close tomorrow and you will get some points fro trying

LNR: alright thanks



Score: 7940 of 7900

You have completed
1 of 31 intro puzzles!

Moves: 4
Time: 0:19

Next is: 1-2: Swing It Around!

Approaches to protein structure prediction

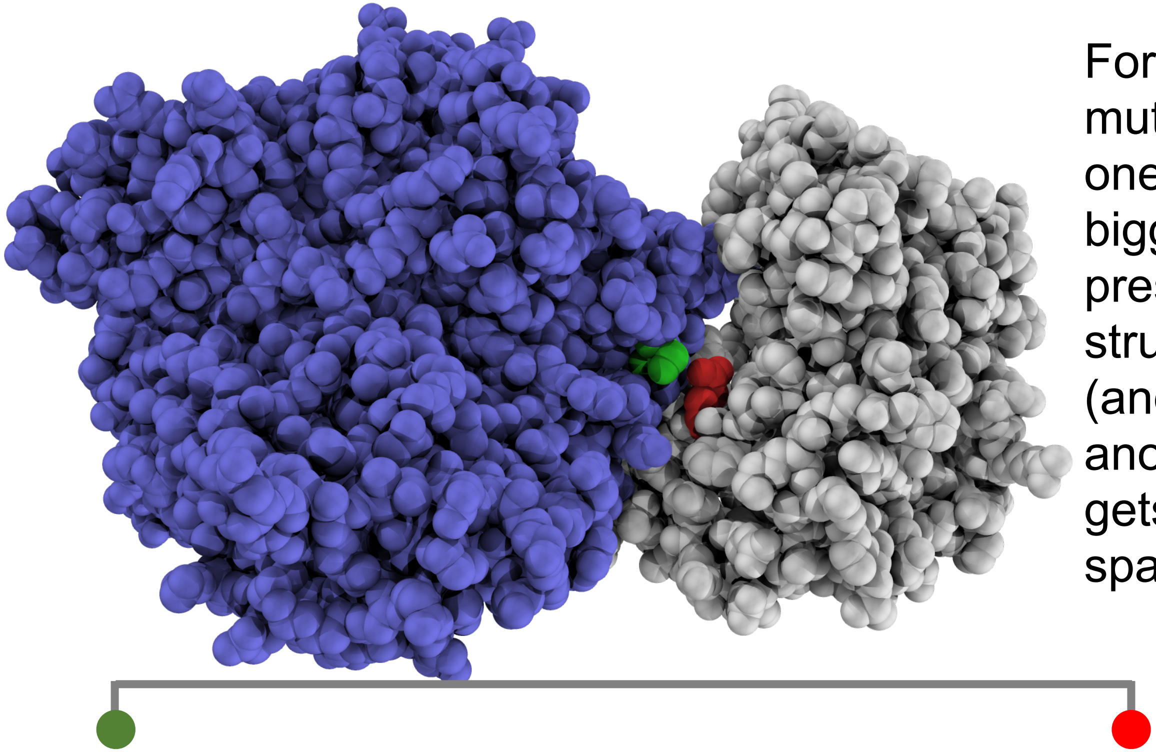
Multiple sequence alignments (coevolution)

We’ve discussed two approaches to protein structure prediction

- Template-based modeling (homology modeling)
  - Used when one can identify one or more likely homologs of known structure
- *Ab initio* structure prediction
  - Does not require any information on homologs of the query protein
  - State-of-the-art *ab initio* approaches still take advantage of available structural data (both to define an energy function and to efficiently search for its minimum)

What if we know sequences of many homologs, but don’t have structures for any of them?

Amino acids in direct physical contact tend to covary or “coevolve” across related proteins

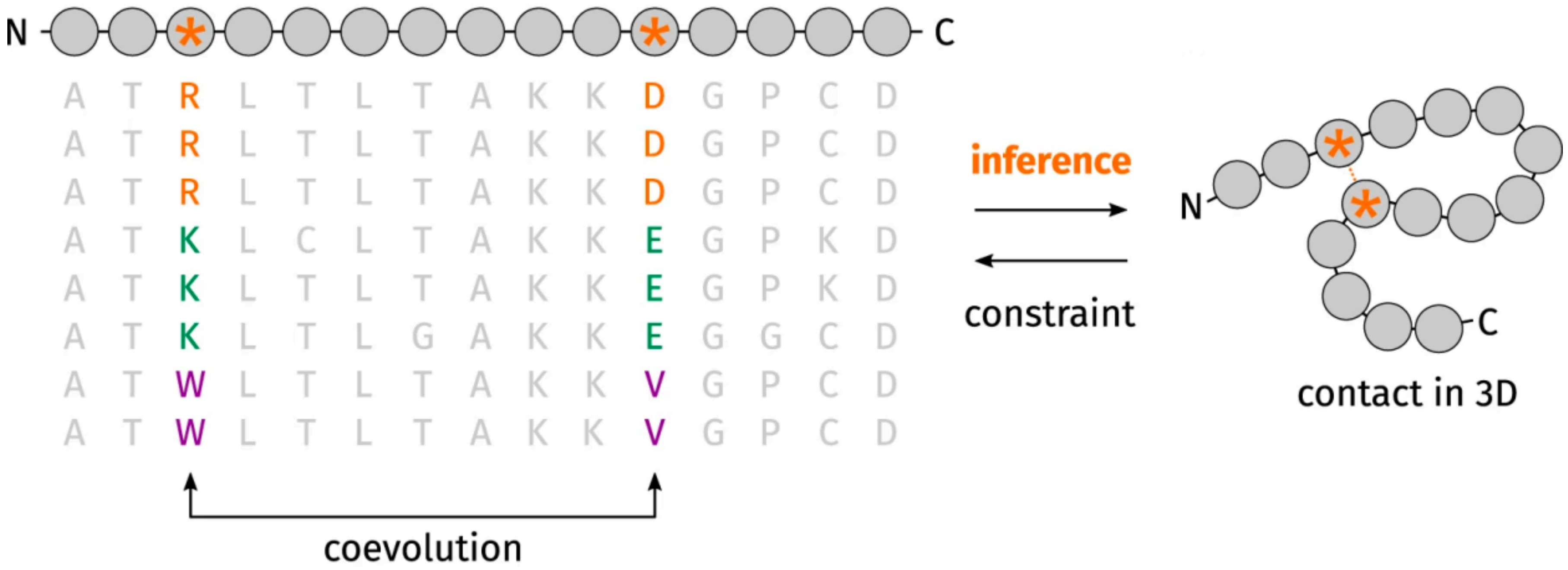


For example, a mutation that causes one amino acid to get bigger is more likely to preserve protein structure and function (and thus survive) if another amino acid gets smaller to make space

```
...GANPMHGRDQSGAVASLTSVA...
...GANPMHGRDQEGAVASLTSVA...
...GANPMHGRDEKGAVASLTSVG...
...GANPMHGRDSHGWLASCLSVA...
...GANPMNGRDVKGFVAAGASVA...
...GANPMHGRDRDGAVASLTSVA...
...GANPMHGRDQVGAVASLTSVA...
...GANPMHGRDOEGAVASLTSVA...
```

```
...VEDLMKEVVTYRHFMNASGG...
...VEALMARVLSYRHFMNASGG...
...VATVMKQVMTYRHYLRATGG...
...VARAMREIGKYAQVLKISRG...
...VPELMQDLTSYRHFMNASGG...
...ADHVLRRLSDFVPALLPLGG...
...FERARTALEAYAAPLRAMGG...
...VPEVMKKVMSYRHYLKATGG...
```

Amino acids in direct physical contact tend to covary or “coevolve” across related proteins



```
A T R L T L T A K K D G P C D
A T R L T L T A K K D G P C D
A T R L T L T A K K D G P C D
A T K L C L T A K K E G P K D
A T K L T L T A K K E G P K D
A T K L T L G A K K E G G C D
A T W L T L T A K K V G P C D
A T W L T L T A K K V G P C D
```

From: https://www.blopig.com/blog/2021/07/alphafold-2-is-here-whats-behind-the-structure-prediction-miracle/

How can we use this observation to predict protein structure?

- Given many sequences of related proteins (whose structure is assumed to be similar), look for amino acids that coevolve. They are probably close together.
- This idea has been around for several decades, but it only became practically useful after 2010, thanks to:
  - A dramatic increase in amount of sequence data available
  - Better computational methods

Protein 3D Structure Computed from Evolutionary Sequence Variation

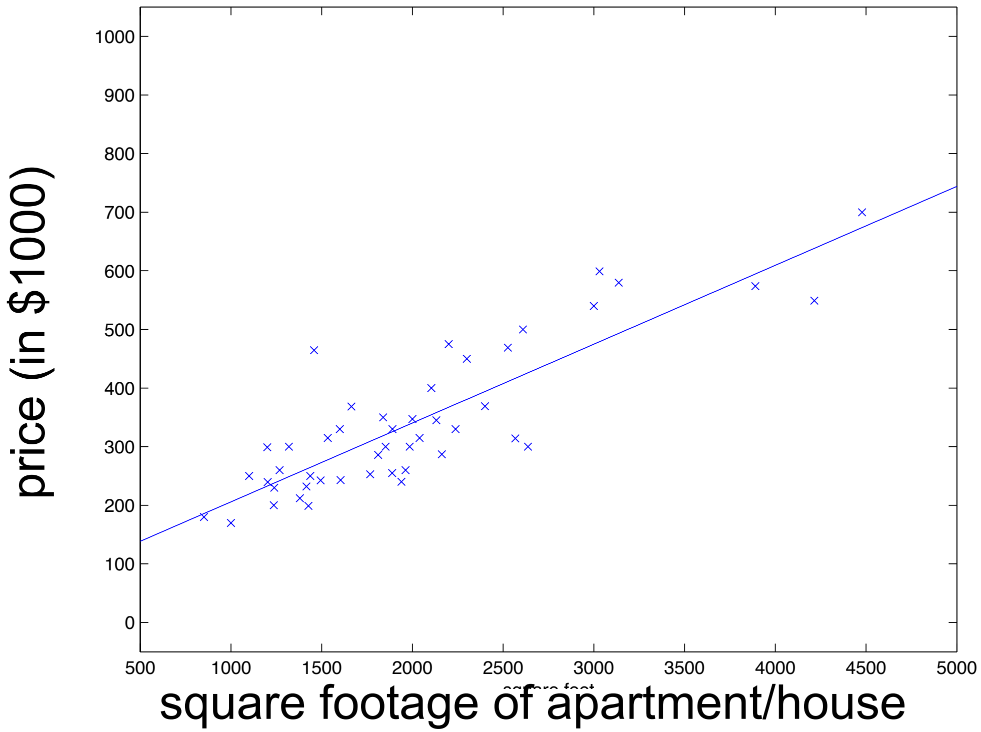
PLoS ONE, 2011

Debora S. Marks[1*☯], Lucy J. Colwell[2☯], Robert Sheridan[3], Thomas A. Hopf[1], Andrea Pagnani[4], Riccardo Zecchina[4,5], Chris Sander[3]

Deep learning methods for protein structure prediction

Machine learning

- Basic idea: given real-world data, find a mathematical relationship that allows you to predict one set of properties from another
- You’ve all done this: fit a straight line (regression)
  - This requires determining two parameters (slope and intercept)



Deep learning

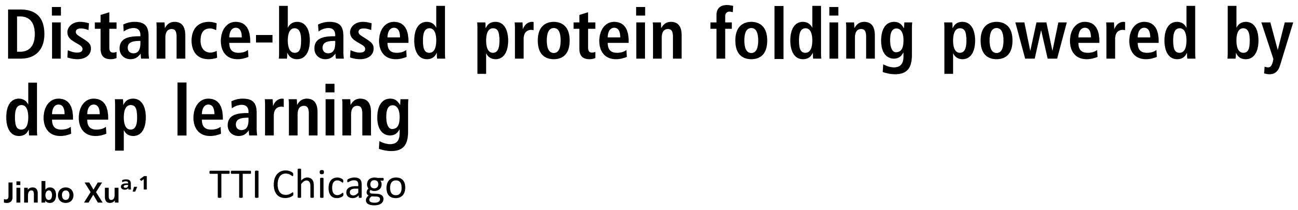
- One can determine much more complicated mathematical relationships
  - Modern neural networks usually have millions, billions, or trillions of parameters
- Over the last decade, researchers have discovered that very large (“deep”) neural networks can work very well, given:
  - Large amounts of data
  - Fitting the right types of mathematical functions in the right way
  - Substantial compute power

Deep learning methods for protein structure prediction

First-generation deep learning methods: learning inter-residue distances from multiple sequence alignments

Deep learning of inter-residue distances

- First generation of deep learning methods for protein structure prediction (including the original AlphaFold method and previous work by academic groups)

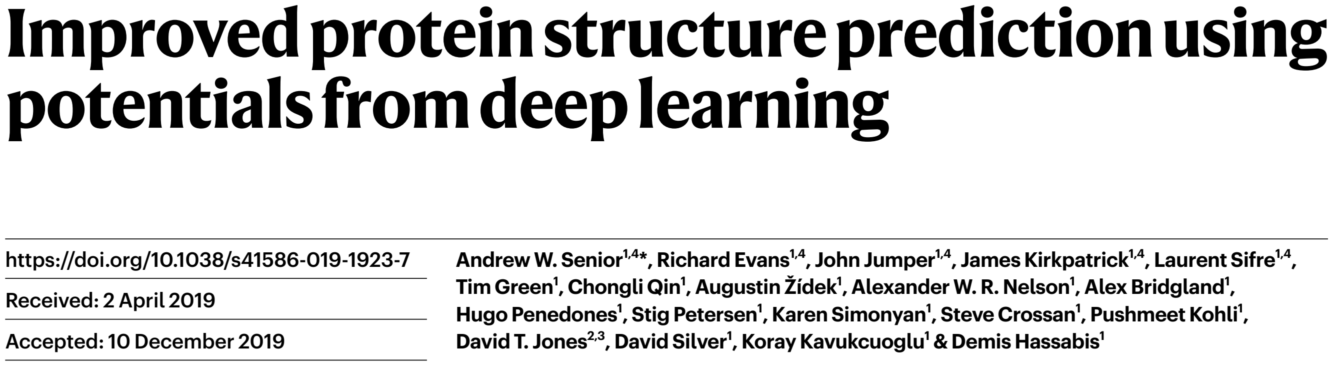


Distance-based protein folding powered by deep learning

Jinbo Xu[a,1]

TTI Chicago

PNAS, 2019



Improved protein structure prediction using potentials from deep learning

https://doi.org/10.1038/s41586-019-1923-7

Received: 2 April 2019

Accepted: 10 December 2019

Andrew W. Senior[1,4]*, Richard Evans[1,4], John Jumper[1,4], James Kirkpatrick[1,4], Laurent Sifre[1,4], Tim Green[1], Chongli Qin[1], Augustin Žídek[1], Alexander W. R. Nelson[1], Alex Bridgland[1], Hugo Penedones[1], Stig Petersen[1], Karen Simonyan[1], Steve Crossan[1], Pushmeet Kohli[1], David T. Jones[2,3], David Silver[1], Koray Kavukcuoglu[1] & Demis Hassabis[1]

Nature, 2020

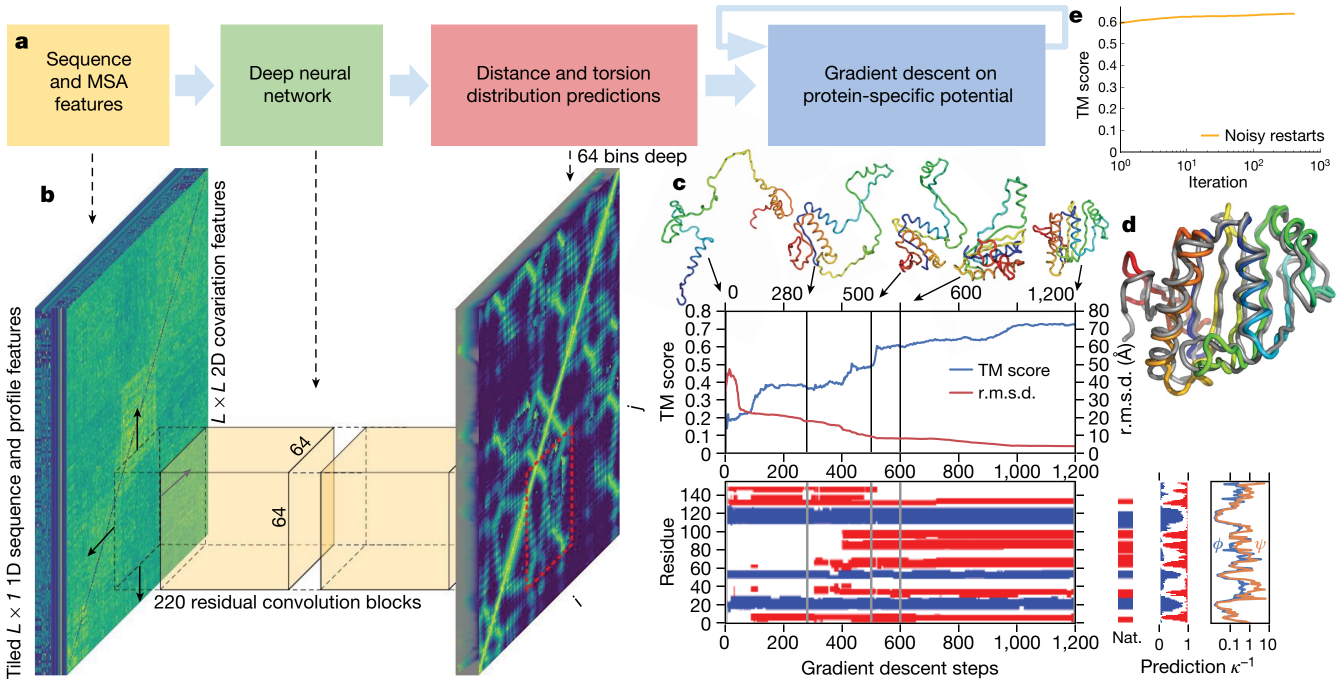
The original AlphaFold (DeepMind)

Deep learning of inter-residue distances

- Key input: multiple sequence alignments
- Key ideas
  - Predict the distance between each pair of amino acid residues, rather than just predicting whether or not each pair is in physical contact
  - Consider covariation not just between residues at two positions, but between entire blocks of adjacent residues
    - This allows one to pick out patterns associated with structural motifs (e.g., alpha helices)
  - Train deep neural networks rather than fitting simpler statistical models with fewer parameters

Deep learning of inter-residue distances

- Then search for a 3D structure that minimizes differences from the predicted distances
  - Certain terms of pre-existing energy functions (e.g., Rosetta all-atom energy function in the case of AlphaFold) are also incorporated at this step to ensure that local structural arrangements are physically reasonable.



Senior et al., *Nature* 2020

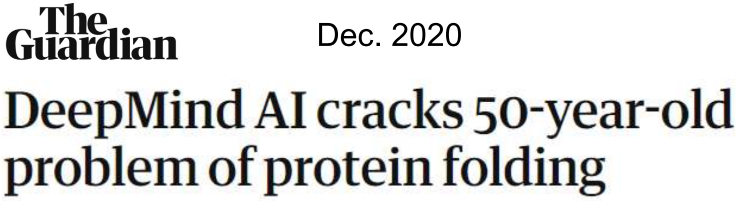
Deep learning of inter-residue distances

- These methods improved over the best previous structure predictions for proteins for which one can’t identify structural templates
- Limitations
  - Don’t incorporate any template information
    - No substantial improvement over template-based methods when templates are available
  - Incorporate only very limited information on local physics, and it’s not part of the machine learning
    - Limits prediction accuracy for side chains

Deep learning methods for protein structure prediction

Second-generation deep learning methods: learning the entire structure

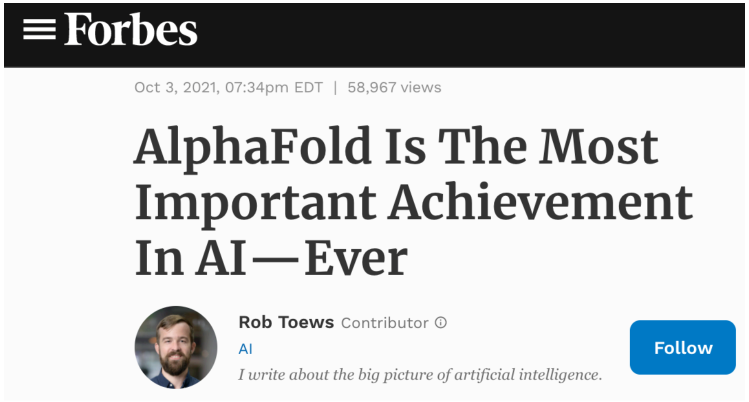
This “second generation” is what you’ve most likely heard about— lots of press coverage



The Guardian

DeepMind AI cracks 50-year-old problem of protein folding

Dec. 2020



Forbes

Oct 3, 2021, 07:34pm EDT | 58,967 views

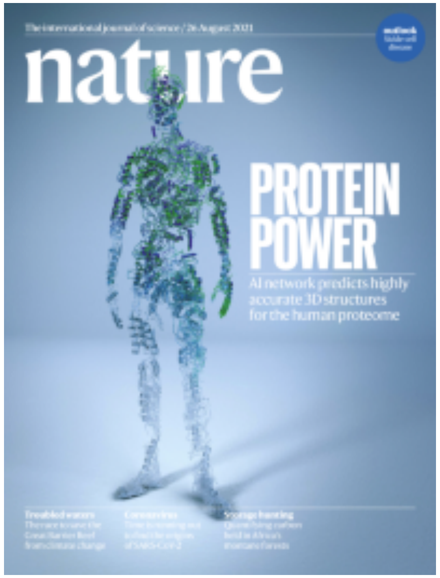
AlphaFold Is The Most Important Achievement In AI—Ever

Rob Toews Contributor

AI

I write about the big picture of artificial intelligence.

Journal covers from August 2021


The international journal of science / 26 August 2021
nature
PROTEIN POWER
AI network predicts highly accurate 3D structures for the human proteome

AlphaFold 2
(DeepMind)

SARS-CoV-2 is evolving fast. What will it do next? p. 844
Effects on surface water from hydraulic fracturing pp. 853 & 896
Baby bats babble like human babies p. 923
Science
$15
20 AUGUST 2021
sciencemag.org
AAAS
PREDICTING STRUCTURES
Deep learning accurately folds proteins p. 871

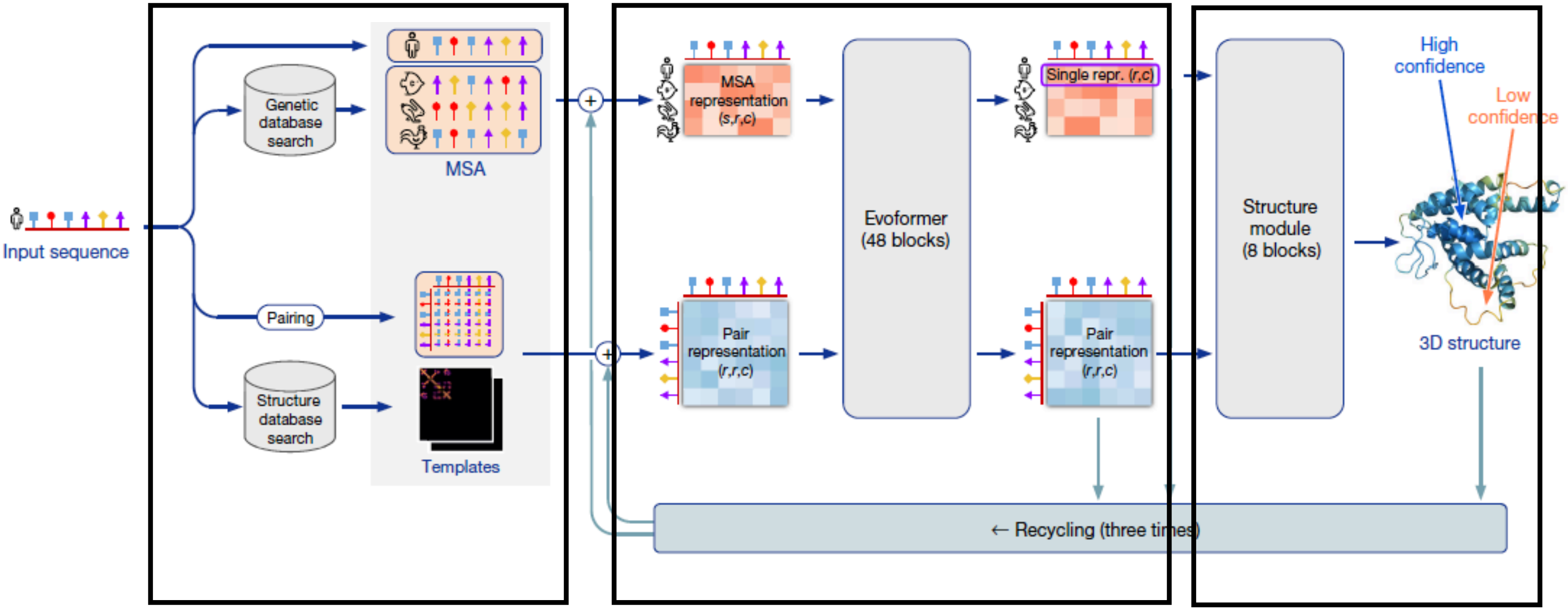
RoseTTAFold
(U. Washington
and academic
collaborators)

- Both AlphaFold 2 (AF2) and RoseTTAFold are deep learning methods for protein structure prediction, with similar architectures
- Note that:
  - AlphaFold 2 is completely different from the original AlphaFold, but both are officially named “AlphaFold”
  - RoseTTAFold structure prediction is very different from Rosetta *ab initio* structure prediction, although both are part of the Rosetta project
  - More recent versions (e.g., AlphaFold 3 and RoseTTAFold All-Atom) enable structure prediction for other types of biomolecules

Second generation: deep learning of entire structure

- Both AlphaFold 2 (AF2) and RoseTTAFold:
  - Take both multiple sequence alignments and templates as inputs (that is, sequences and structures of related proteins)
  - Learn favorability of local arrangements of amino acid residues and their constituent atoms (i.e., side-chain packing) from very large numbers of available protein structures
  - Learn how to combine these sources of information effectively

AF2 architecture


Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

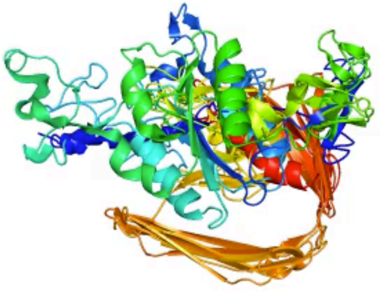
Identify sequences and structures of related proteins

Iteratively refine two representations of this information, one indexed by amino acid position and the other by *pairs* of positions. Each informs the other.

Add a third representation: position and orientation of each amino acid. Iteratively refine these, then predict conformation of each side chain

Jumper et al., *Nature*, 2021

AF2 in action



Recycling iteration 0, block 01
Secondary structure assigned from the final prediction

- AF2 doesn’t actually “see” most of these intermediate structures. They are guesses of what would have been predicted based on intermediate states (layers) of the network.
- Structure is initially “compressed” because all residues are initially superimposed

Also note …

- AF2 and RoseTTAFold are highly customized architectures, incorporating prior knowledge about proteins. This isn’t “generic” machine learning.
- These methods combine Cartesian coordinate (i.e., *x*, *y*, *z*) and torsional angle representations of proteins
- Beyond the machine learning, these methods involve:
  - Pre-processing: calling other software to select and align homologous sequences and to select templates
  - Post-processing: refine results with an existing molecular mechanics force field

Deep learning methods for protein structure prediction

Large language models

Optional material

Large language models for protein structure prediction

- Recent work from several groups (most notably, the ESM models from Meta)
  - Exploit large language models similar in spirit to Chat-GPT
- Approach:
  - Using hundreds of millions of naturally occurring protein sequences, train a model that can fill in missing residues
  - Add a “folding head” to this model, which takes the model’s learned representation and predicts structure from it.
    - The “folding head” of ESM2 is similar to AF2. ESM3 predicts structure more directly (but is trained on results of ESM2 and AF2).
    - No explicit multiple sequence alignment (MSA) or templates are necessary
- Typically less accurate than AF2, but works without explicitly identifying homologs, and learned representation can be used for other purposes.

Is the protein folding problem solved?

- The original “protein folding problem” was determining *how* a protein gets to its folded structure
- AlphaFold, RoseTTAFold, etc. tackle a different problem: protein structure prediction
  - They do this very well, and it’s a very important problem!
  - These methods don’t find a protein’s structure the way the protein does

Strengths of AlphaFold and RoseTTAFold

- Major improvement in accuracy over previous state of the art
- Work automatically for multi-domain proteins
  - No need to predict separate structures for each domain, then piece them together
- Also predict structures of multi-protein complexes, and more recent versions predict structures of other biomolecules and complexes

Limitations of these methods

- Predictions not perfect
  - E.g., side chain orientations often incorrect, and they’re important for drug discovery applications.
- Prediction of a structure under unspecified conditions
  - This is by design. In the CASP (Community Assessment of Structure Prediction) competition, competitors do not have access to the experimental structure *or* any information about the conditions under which its was solved.
    - For example, what is the acidity (pH)?
  - But when one uses an experimental structure, the conditions are known, and that information is important in applications.
    - Often experimental structures are available under different conditions, and those structures are different.

When should one determine a protein structure experimentally vs. predicting it computationally?

- Key factors to consider:
  - How long will experimental structure determination take for the protein/biomolecule of interest?
  - How do you want to use the structure?
    - How much accuracy do you need?
    - Are the conditions important?
- If you want to get a sense of the protein's overall shape, I'd use the predicted structure
- For most drug design applications, or for understanding complex functional mechanisms, I'd determine a structure experimentally if practical

Optional Reading: Terwilliger et al., AlphaFold predictions are valuable hypotheses and accelerate but do not replace experimental structure determination, *Nature Methods* (2023)

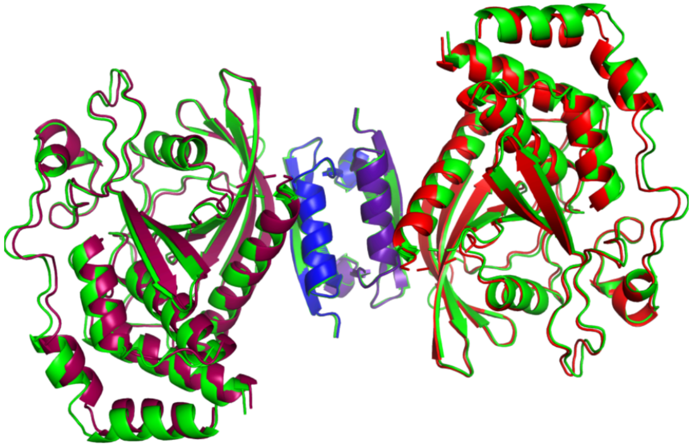
Protein structure prediction ≠ drug discovery

- Using the structure of a drug target to discovery and improve drugs is itself a challenging computational problem, even when high-resolution, experimentally determined structures are available

Predicting structures of other biomolecules and complexes

Predicting structures of protein-protein complexes

- AlphaFold 2 and RoseTTAFold—with little or no modification—are also good at predicting structures of multi-protein complexes, given the sequence of each component protein
  - Caveat: one needs to correctly specify which proteins form complexes and how many copies of each protein are in the complex



Experimentally determined structure (green) compared to AlphaFold-multimer prediction (other colors) for a complex comprising two copies of each of two proteins

From https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2.full.pdf

Predicting structures of RNA and DNA

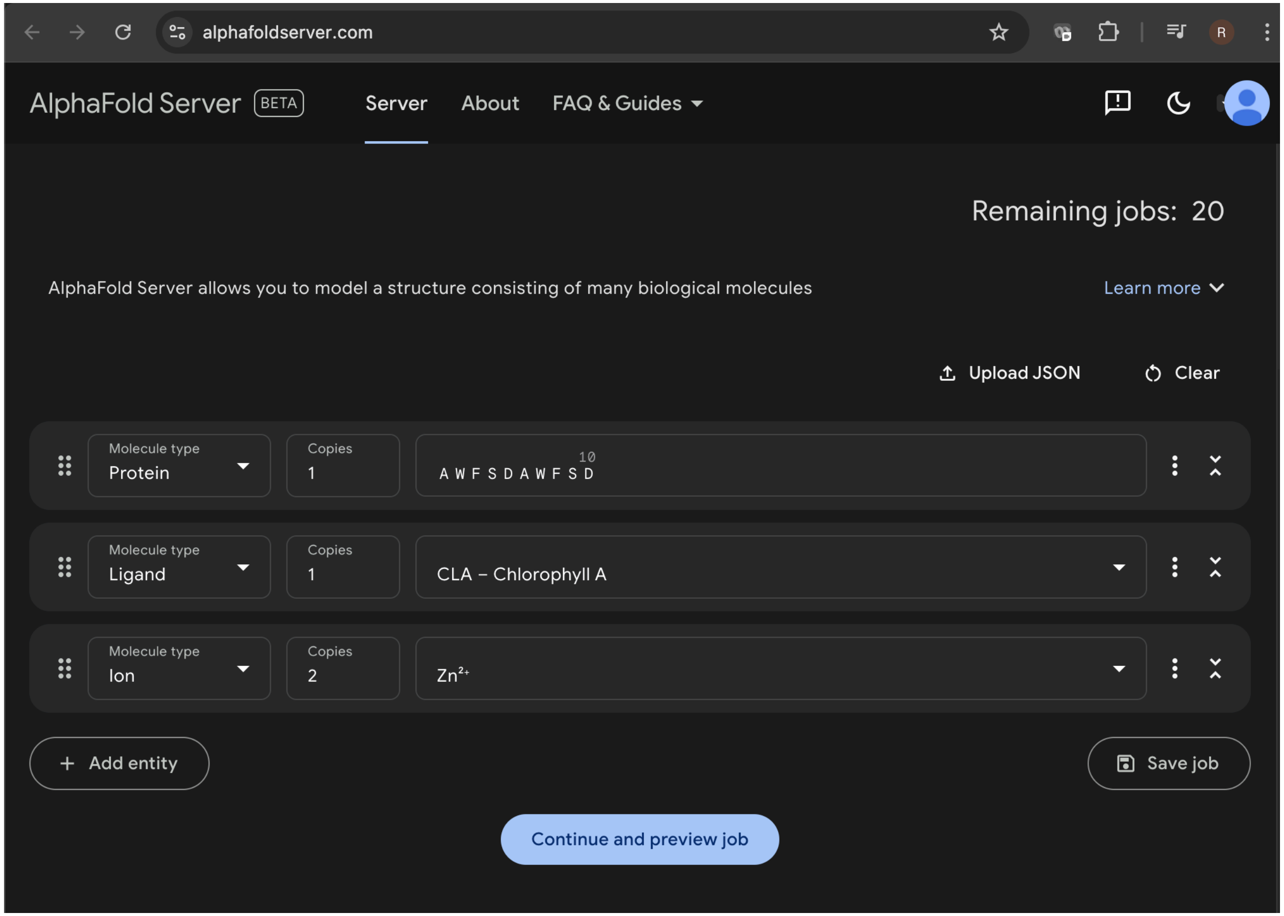
- The approaches used in AlphaFold 2 and RoseTTAFold can be generalized to predict structures of nucleic acids (RNA and DNA) as well as complexes of nucleic acids and proteins
  - Add additional “residue” types corresponding to nucleotides in RNA and DNA (rather than only amino acids)

A third generation of deep learning methods for structure prediction goes even further

- AlphaFold 3 and RoseTTAFold All-Atom allow structure prediction for complexes including:
  - Small molecule
  - Proteins with non-standard amino acids and covalently modified amino acids (e.g., phosphorylated amino acids)
  - Nucleic acids with non-standard nucleotides and covalently modified nucleotides (e.g., methylated DNA)

AlphaFold 3

- Try out the web server



Remaining jobs: 20

AlphaFold Server allows you to model a structure consisting of many biological molecules

Molecule type: Protein | Copies: 1 | 10 AWFSDAWFSD

Molecule type: Ligand | Copies: 1 | CLA – Chlorophyll A

Molecule type: Ion | Copies: 2 | Zn^{2+}

- Note: the server has limited functionality, and code has not been released (which is why we use AlphaFold 2 in the assignment)

How does AlphaFold 3 work?

- Similar approach to AlphaFold 2, but a few important changes, including:
  - Combine all-atom and residue-level representations of molecules
  - The last part of the predictor, which reconstructs the structure, uses “denoising diffusion” (an approach we’ll discuss later)

For more detail, see “The Illustrated AlphaFold,” by Stanford PhD students Elana Simon and Jake Silberg: https://elanapearl.github.io/blog/2024/the-illustrated-alphafold/