

Predicting Structures of Proteins and Other Biomolecules

CS 279 / BIOE 279 / BIOPHYS 279 / BMDS 245 / CME
279

Oct. 6, 8 and 13, 2026

Ron Dror

Reminder: Assignment 1

- If you haven't already tested the software setup you plan to use, please do so today

Assignment 2 released

- Assignment 2 is on the course website
 - Due October 29 at 1 pm
- Assignment 2 kickstarts:
 - Part 1: Monday, October 19th, 11 AM - 12 PM.
 - Part 2: Tuesday, October 20th, 11 AM - 12 PM.
 - They will both be on Zoom and they will both be recorded.
- We recommend that you use the LTS computers!

Outline

- Why predict biomolecular structure (e.g., 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
 - Multiple sequence alignments (coevolution)
 - *Ab initio* modeling (e.g., Rosetta)
- Deep learning (AI) methods for protein structure prediction
 - First generation: learning inter-residue distances from multiple sequence alignments
 - Second generation: learning the entire structure
 - Third generation: generalization to other biomolecules
- Large language models for proteins
- Strengths and limitations

Why predict biomolecular structure
(e.g., protein structure)?

Problem definition: protein structure prediction

- 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 other molecules (and which ones)?

SVYDAAAQLTADVKKDLRDSW
KVIGSDKKGNVALMTTLFAD
NQETIGYFKRLGNVSQGMAND
KLRGHSITLMYALQNFIDQLD
NPDSL DLVCS.....



Why predict protein structures rather than determining them experimentally?

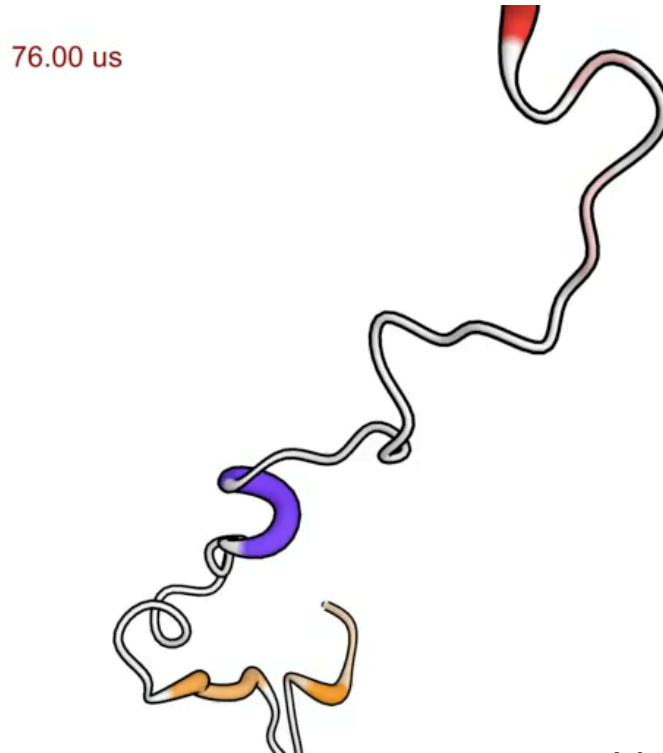
- Because predicting them computationally is (hopefully) cheaper and faster
- This answer is different from the answer to “why perform MD simulations?”
 - MD simulations are computationally expensive but often allow one to access information that simply can't be observed with existing experimental methods

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

Can we use (pure) physics-based
methods?

A molecular dynamics simulation of a protein folding spontaneously



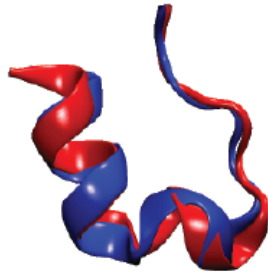
Lindorff-Larsen et al., *Science* 2011

- In this simulation, a protein folds to a structure matching that determined experimentally. The simulation used no information about the experimentally determined structure.

Why not just simulate the folding process by molecular dynamics?



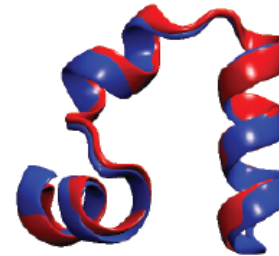
Chignolin



Trp-cage



BBA

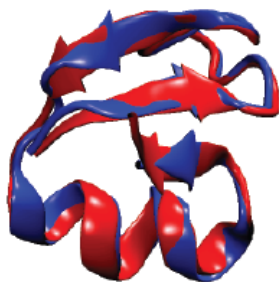


Villin

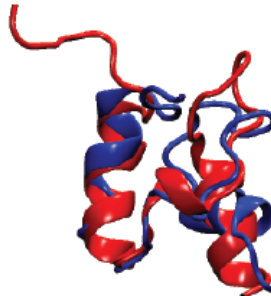
This is possible for some proteins.



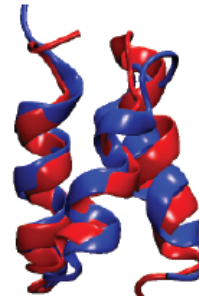
WW domain



NTL9

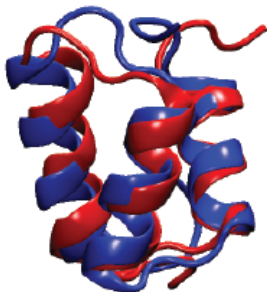


BBL



Protein B

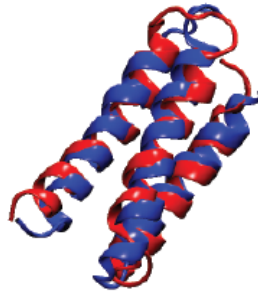
Example:
Simulation vs.
experiment for 12 fast-folding proteins, up to 80 residues each



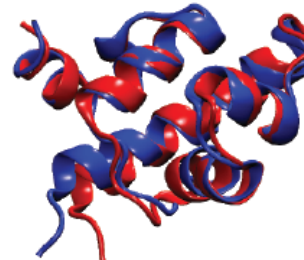
Homeodomain



Protein G



α 3D



λ -repressor

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, and of the even larger number of known sequences of naturally occurring proteins!

Knowledge-based methods

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

- The PDB contains over 250,000 experimentally determined protein structures
- Can we use that information to help us predict new structures?
- **Yes!**

Me
WANTS
THE
DATA



We can also use the huge number of known, naturally occurring protein *sequences* (~200 million in the UniProt database, and billions in the MGnify database)

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 naturally occurring 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”)

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

- We can still use information based on known structures
 - For example, 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?

- Recent versions of AlphaFold and similar machine learning methods 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

Template-based structure prediction: basic workflow

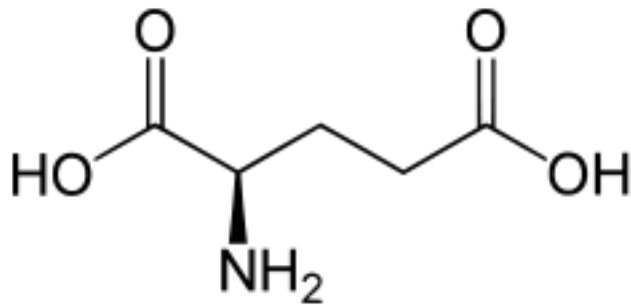
- User provides a *query* sequence with unknown structure
- Search the PDB for proteins with similar sequence and known structure. Pick the best match (the *template*).
- Build a model based on that template
 - One can also build a model based on multiple templates, where different templates are used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

What does it mean for two sequences to be similar?

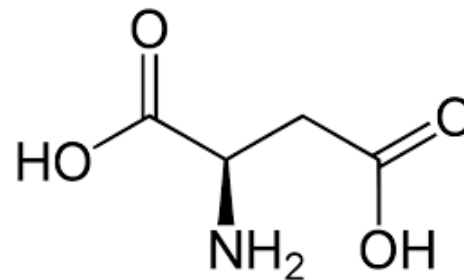
- Basic measure: count minimum number of amino acid residues one needs to change, add, or delete to get from one sequence to another
 - *Sequence identity*: amino acids that match exactly between the two sequences
 - Not trivial to compute for long sequences, but there are efficient dynamic programming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

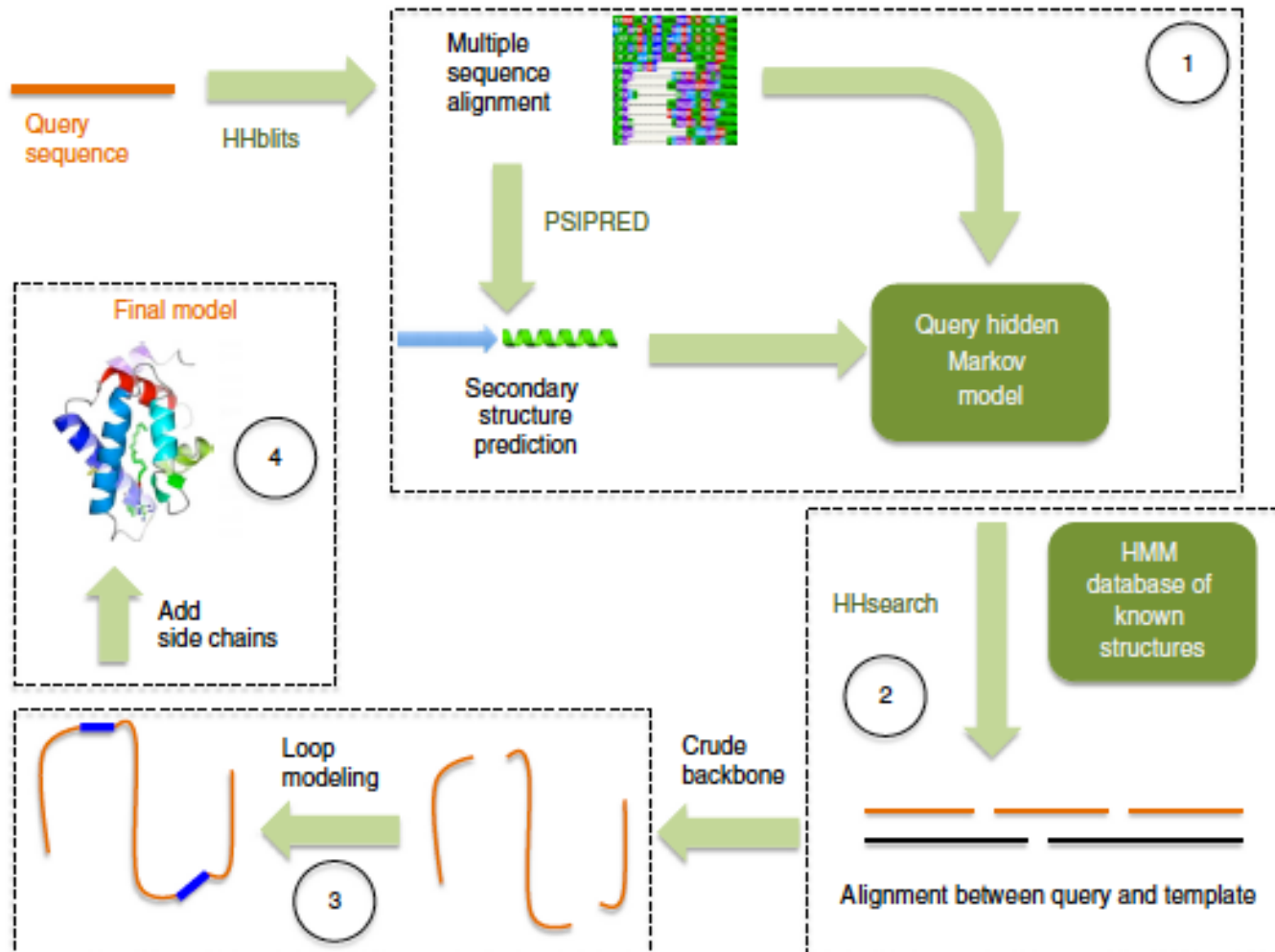
Optional material

- 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

Example of template-based modeling: Phyre2 server

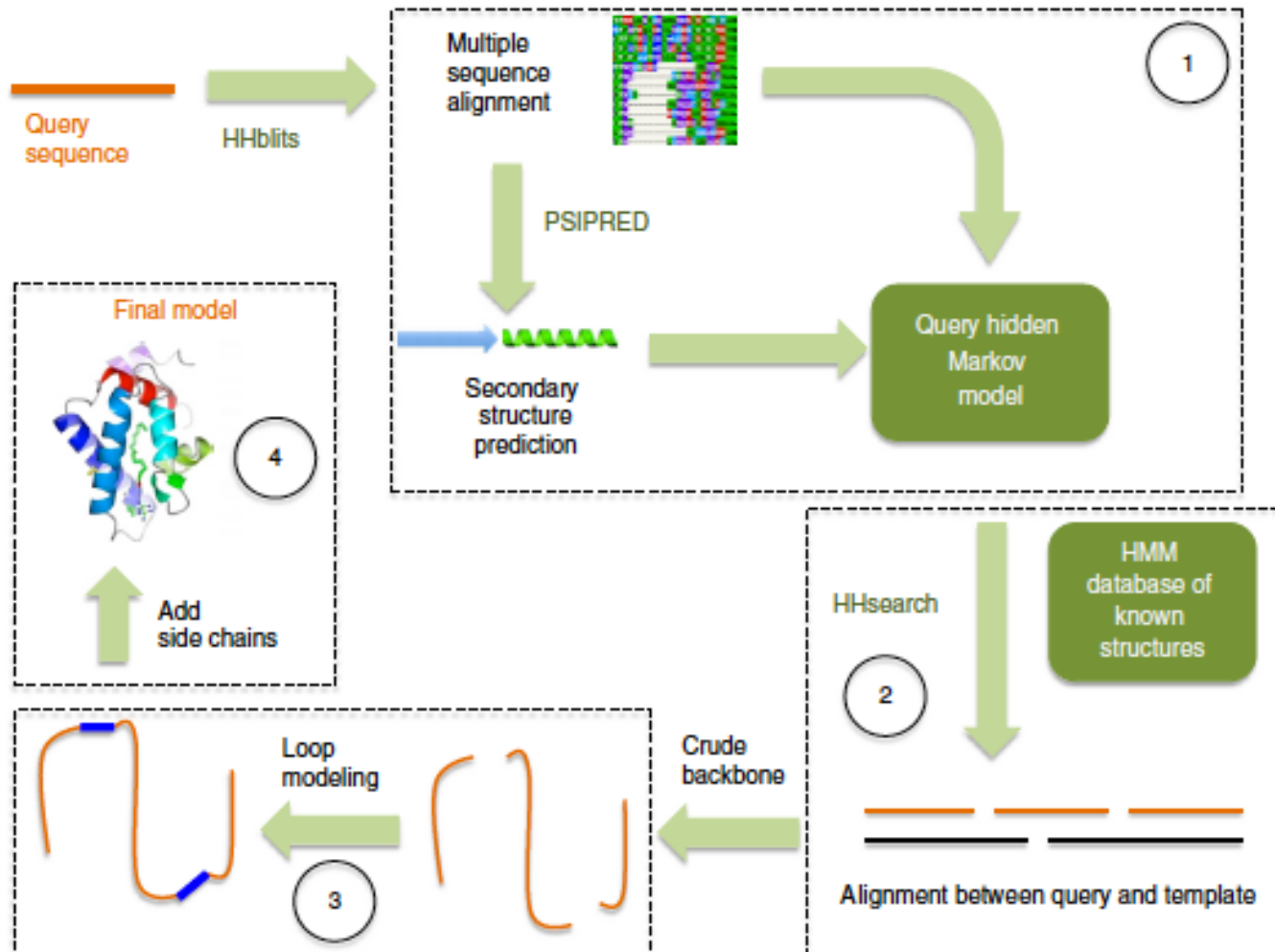
- Try it out: <http://www.sbg.bio.ic.ac.uk/phyre2/>
- Why use Phyre2 as an example of template-based modeling?
 - Among the better automated structure prediction web-servers
 - Among the easiest to use
 - Approach is similar to that of other template-based modeling methods
 - Great name!

Phyre2 algorithmic pipeline



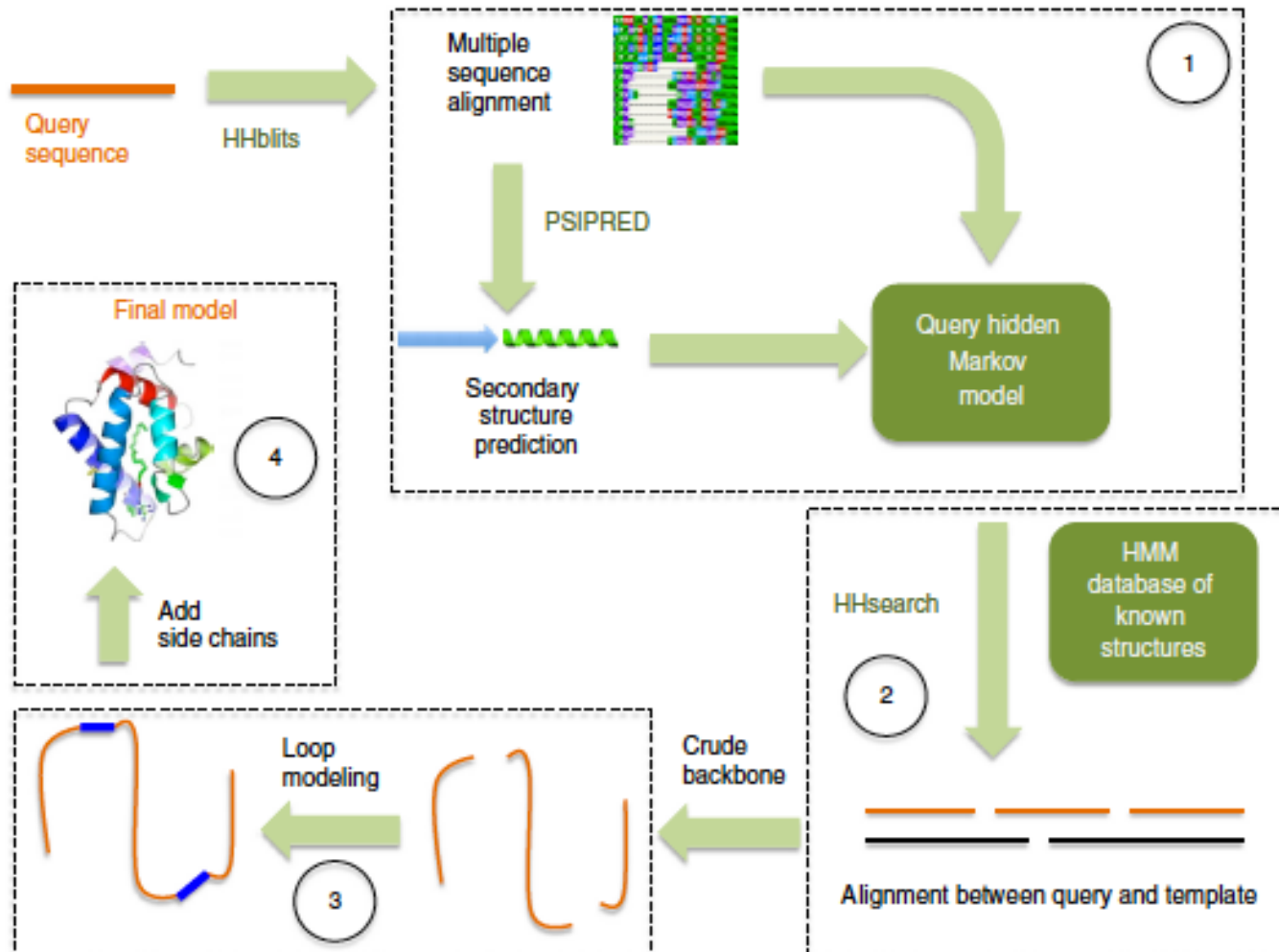
Phyre2 algorithmic pipeline

Identify similar sequences in protein sequence database

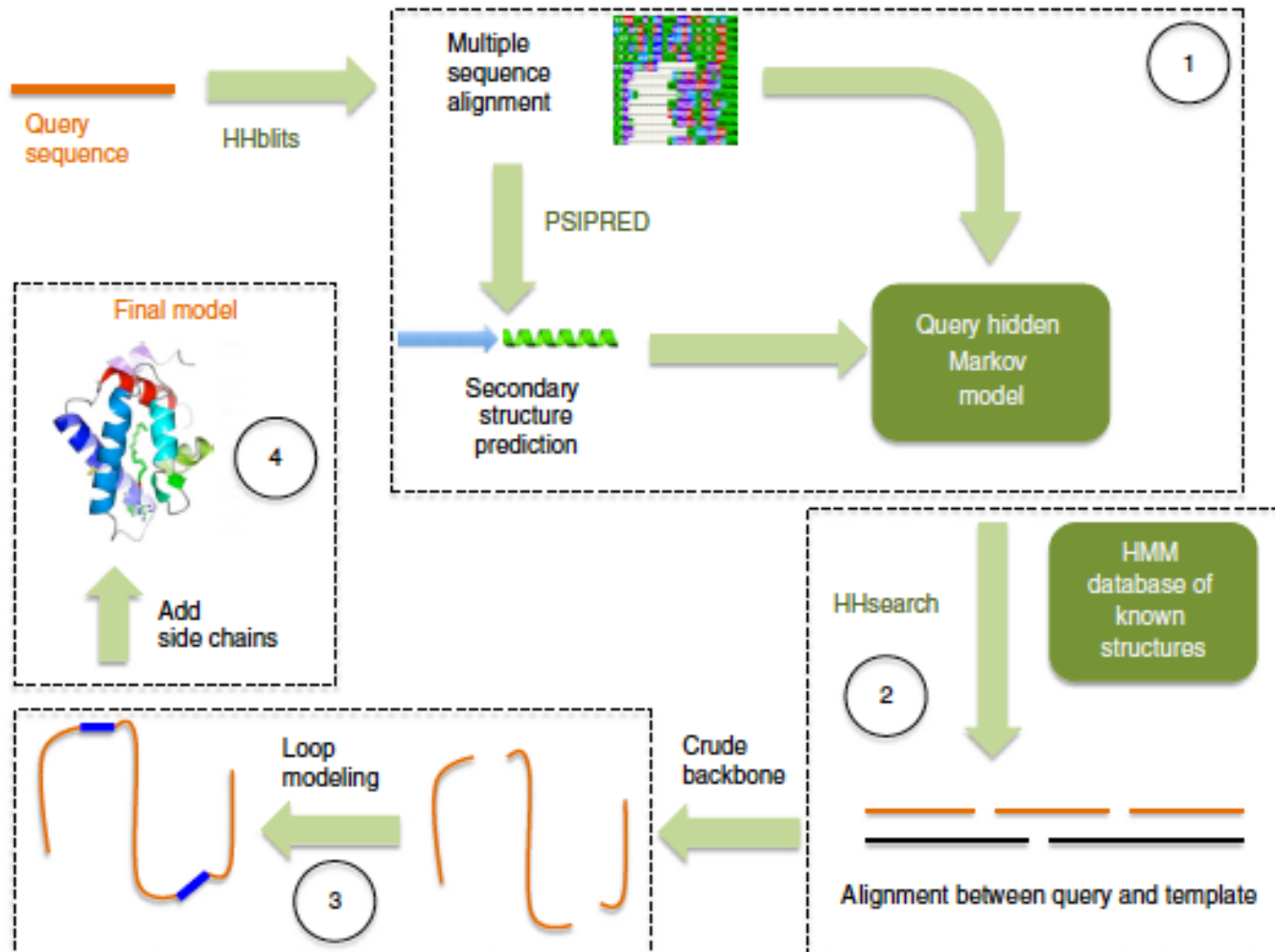


Phyre2 algorithmic pipeline

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.

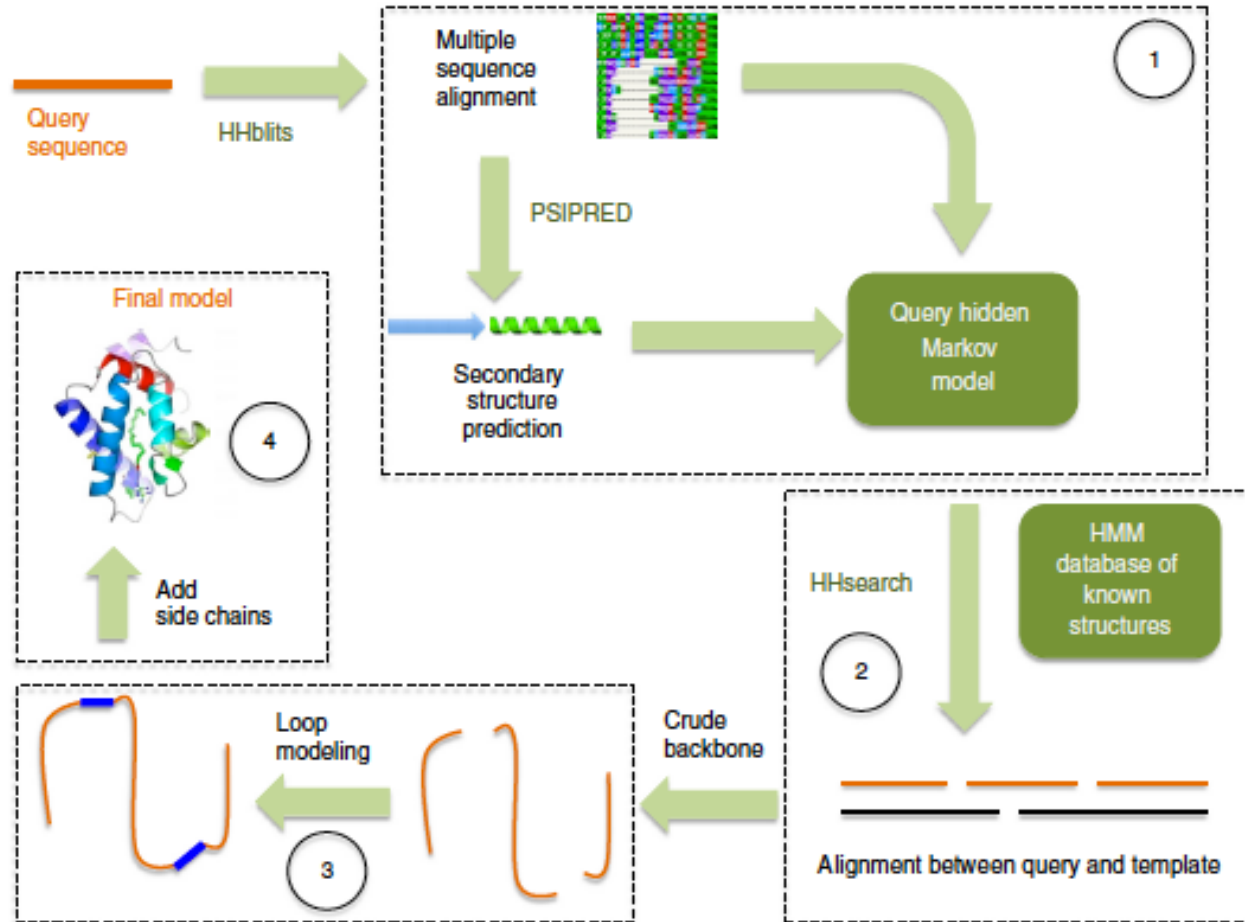


Phyre2 algorithmic pipeline



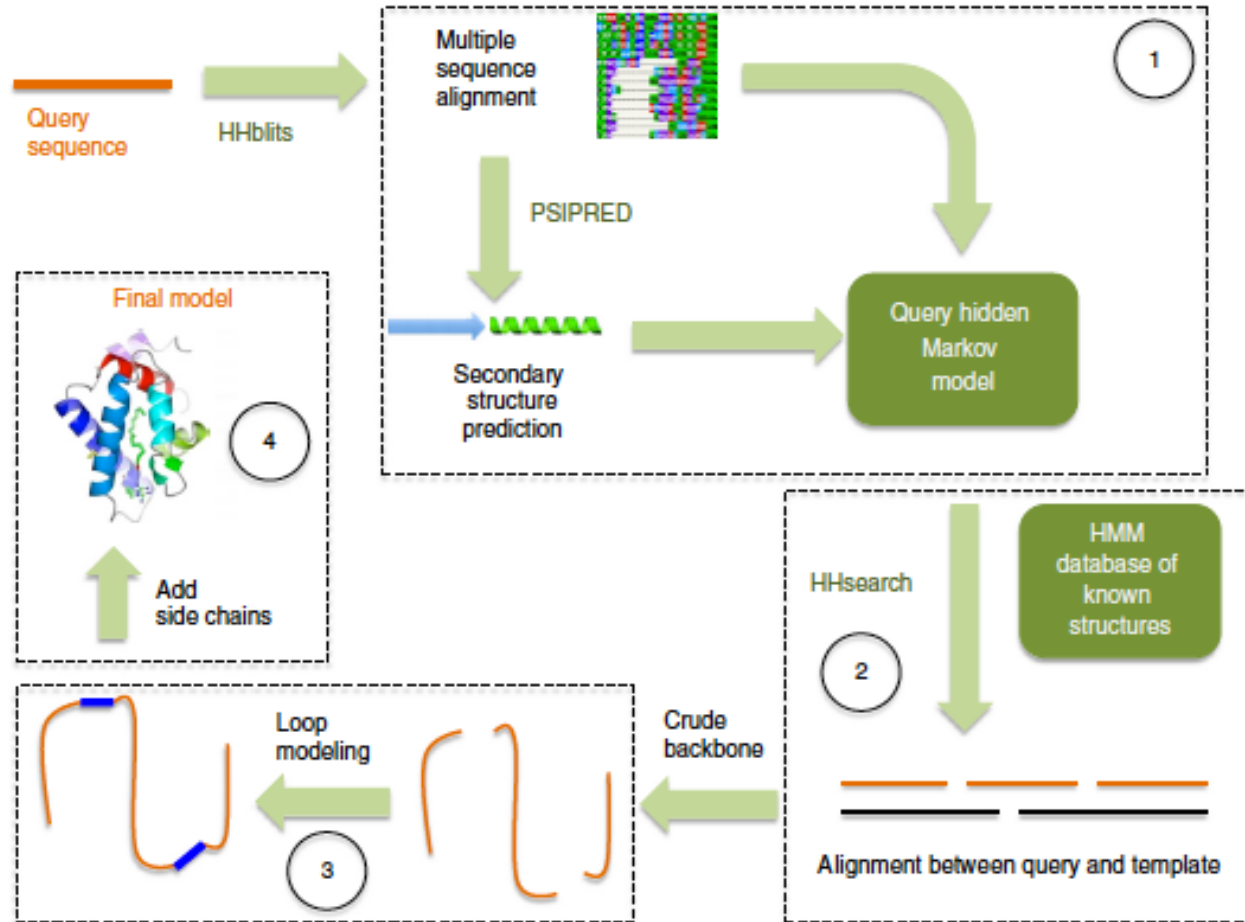
Compute optimal alignment of query sequence to template structure

Phyre2 algorithmic pipeline



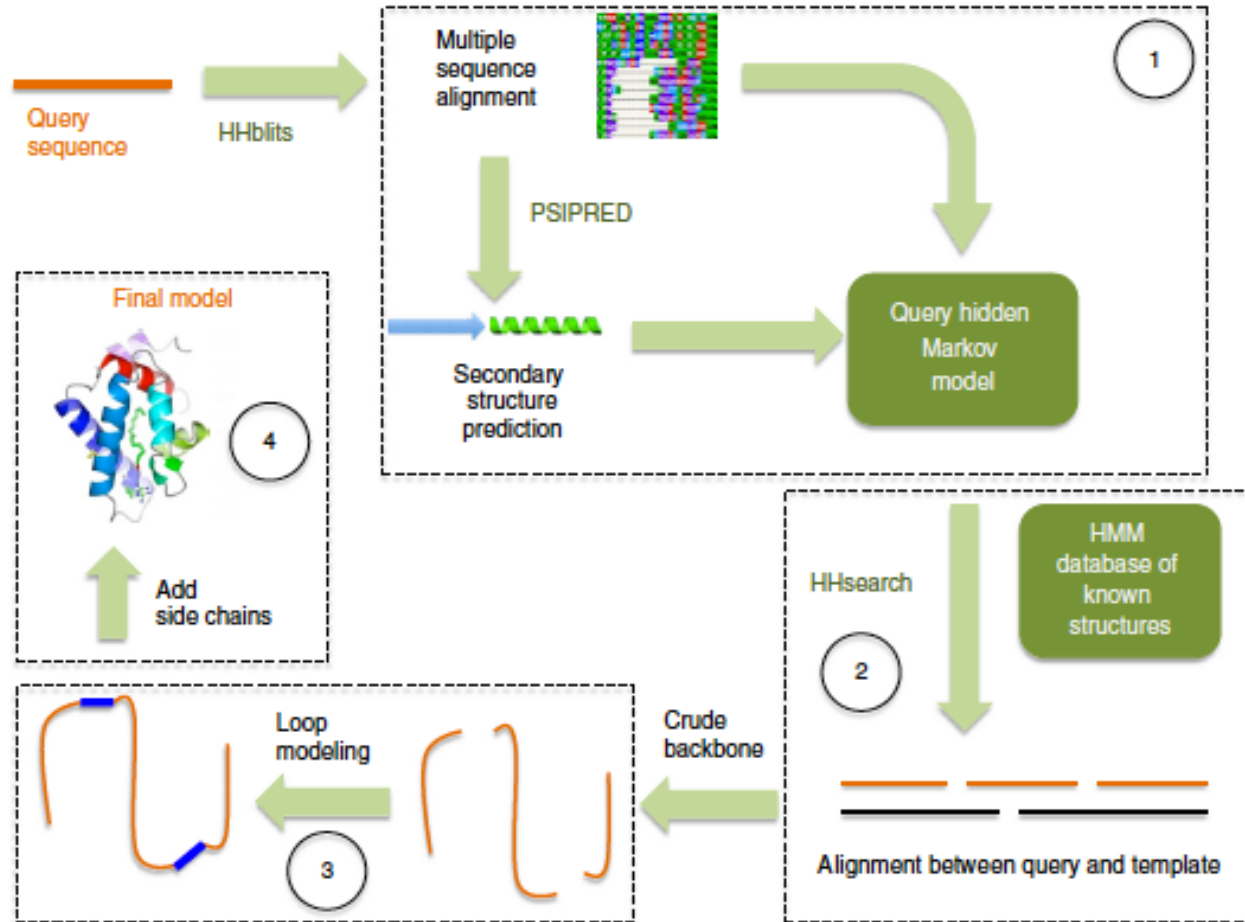
Build a crude backbone model (no side chains) by simply superimposing corresponding amino acids.

Phyre2 algorithmic pipeline



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*).

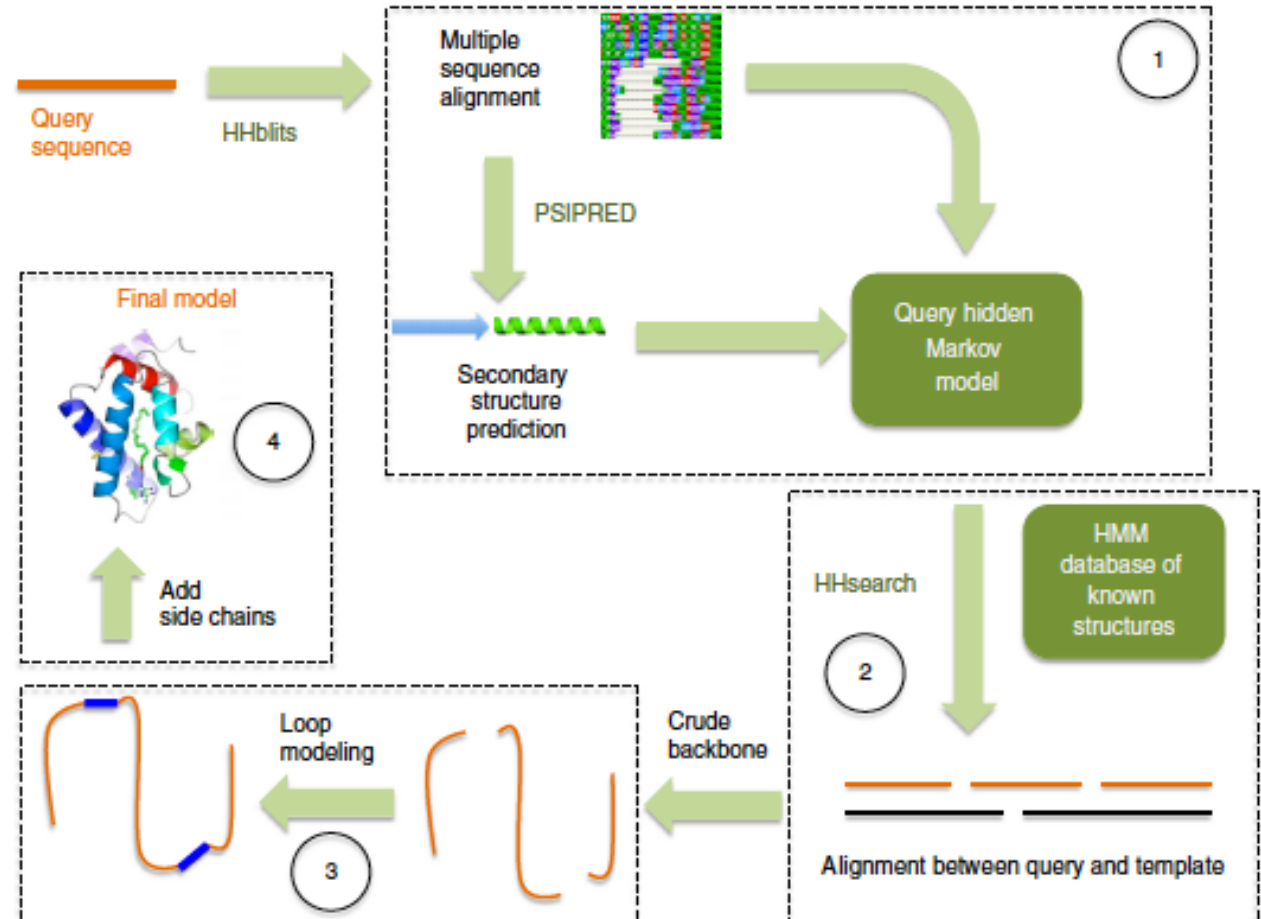
Phyre2 algorithmic pipeline



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.

Phyre2 algorithmic pipeline

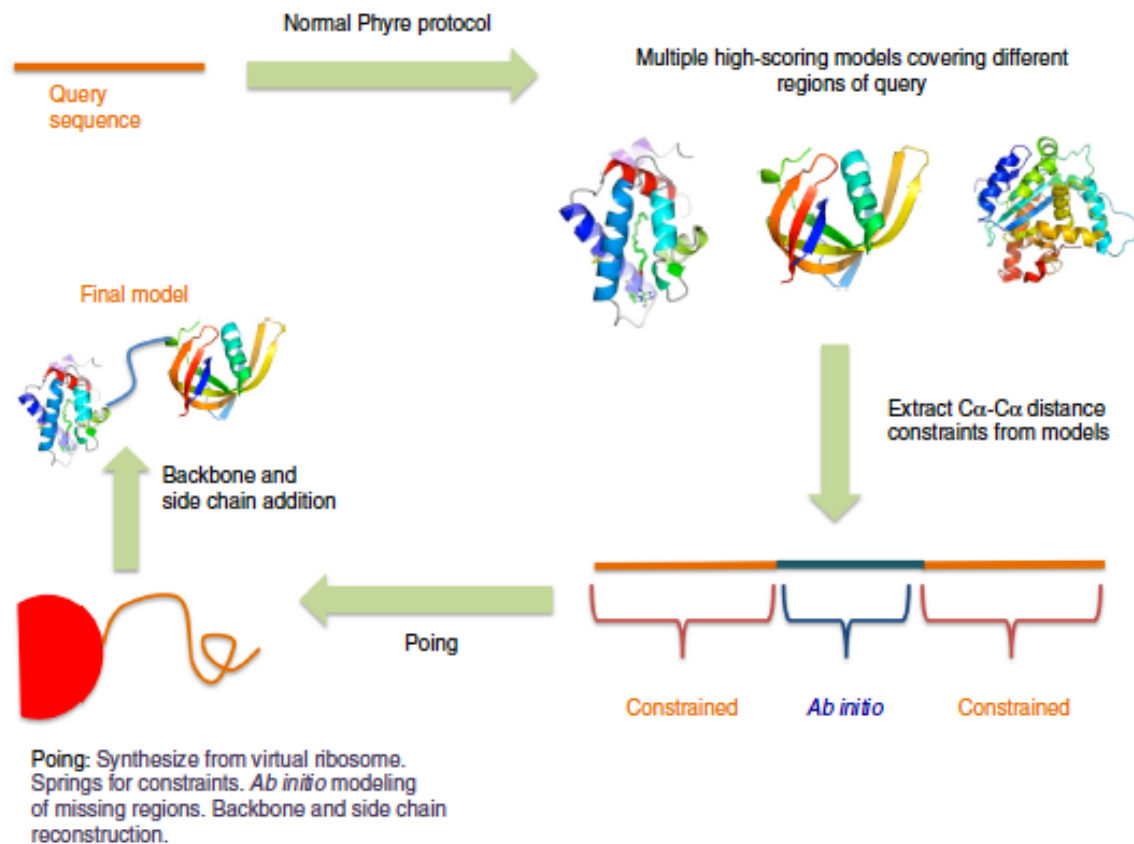
Add side chains. Use a database of commonly observed structures for each side chain (these structures are called *rotamers*). Search for combinations of rotamers that will avoid steric clashes (i.e., atoms ending up on top of one another).



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



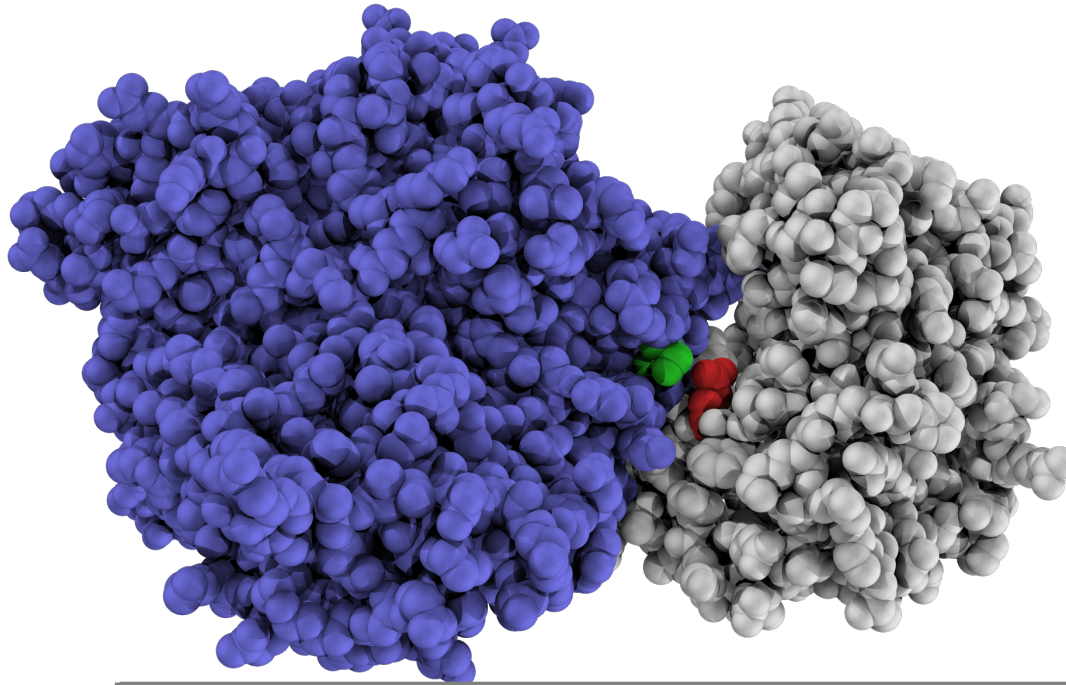
Approaches to protein structure prediction

**Multiple sequence alignments
(coevolution)**

What if we know sequences of many homologs of the query protein, but don't have structures for any of them?

- Because so many sequences of naturally occurring proteins are now available, we can almost always identify proteins that share sequence similarity with the query protein, at least if it's a naturally occurring protein

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

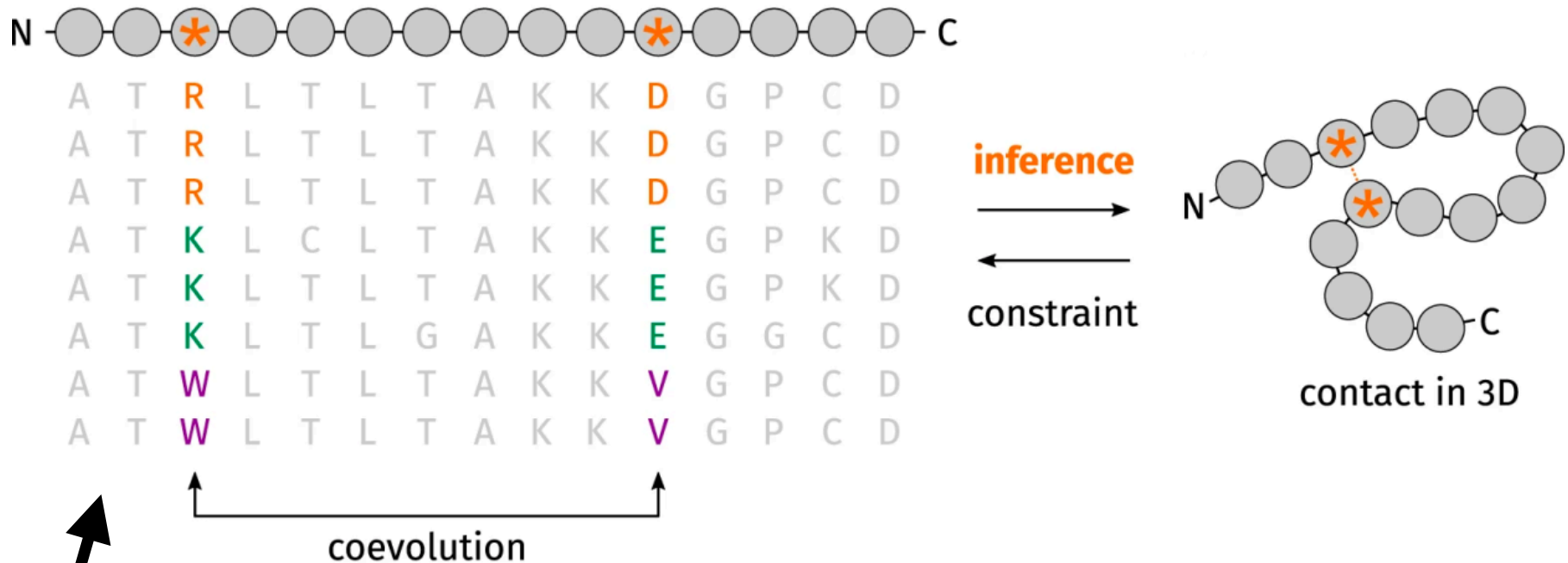


For example, a mutation that causes one amino acid to get bigger is more likely to preserve protein structure and function (and thus survival of the organism) if another amino acid gets smaller to make space

... GANPMHGRDQ**S**GAVASLTSVA ...
... GANPMHGRDQ**E**GAVASLTSVA ...
... GANPMHGRDE**K**GAVASLTSVG ...
... GANPMHGRDS**H**GWLASCLSVA ...
... GANPMNGRDV**K**GFVAAGASVA ...
... GANPMHGRDR**D**GAVASLTSVA ...
... GANPMHGRDQ**V**GAVASLTSVA ...
... GANPMHGRDO**E**GAVASLTSVA ...

... VEDLMK**E**VV TYRHFMNASGG ...
... VEALMA**R**VLSYRHFMNASGG ...
... VATVMK**Q**VMTYRH YLRATGG ...
... VARAMR**E**IGKYAQVLKISR G ...
... VPELMQ**D**LTSYRHFMNASGG ...
... ADHVLR**R**LSDFVPALLPLGG ...
... FERART**A**LEAYAAPLRAMGG ...
... VPEVMK**K**VMSYRH YLKATGG ...

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



- This is a “Multiple Sequence Alignment” (MSA)
 - Each sequence is the amino acid sequence of a different protein
 - These are related proteins from many species

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 many 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*9}, Lucy J. Colwell²⁹, Robert Sheridan³, Thomas A. Hopf¹, Andrea Pagnani⁴, Riccardo Zecchina^{4,5}, Chris Sander³

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 (even their sequences)
- Also known as “*de novo* structure prediction”
- Many approaches proposed over time

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>
- Why use Rosetta as an example?
 - Among the better *ab initio* modeling packages (for 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:
 - 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
 - Assume all bond lengths and bond angles are fixed
 - Do not explicitly represent solvent (e.g., water)

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)	$\sum_i -\ln [P(aa_i nb_i)]$	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)	$\sum_i \sum_{j>i} -\ln \left[\frac{P(aa_i, aa_j s_{ij}d_{ij})}{P(aa_i s_{ij}d_{ij})P(aa_j s_{ij}d_{ij})} \right]$	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 : $SS_{\phi,\theta} + SS_{hb} + SS_d$ SchemeB : $SS_{\phi,\theta} + SS_{hb} + SS_{d\sigma}$ where $SS_{\phi,\theta} = \sum_m \sum_{n>m} -\ln [P(\phi_{mn}, \theta_{mn} d_{mn}, sp_{mn}, s_{mn})]$ $SS_{hb} = \sum_m \sum_{n>m} -\ln [P(hb_{mn} d_{mn}, s_{mn})]$ $SS_d = \sum_m \sum_{n>m} -\ln [P(d_{mn} s_{mn})]$ $SS_{d\sigma} = \sum_m \sum_{n>m} -\ln [P(d_{mn}\sigma_{mn} \rho_m, \rho_n)]$	m, n = strand dimer indices; dimer is two consecutive strand residues V = vector between first N atom and last C atom of dimer $\hat{m}$ = unit vector between $\hat{V}_m$ and $\hat{V}_n$ midpoints $\hat{x}$ = unit vector along carbon–oxygen bond of first dimer residue $\hat{y}$ = 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, $\sum_{k=m,n} 0.5(\hat{m} \cdot \hat{x}_k + \hat{m} \cdot \hat{y}_k)$ (< 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) 7

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	$-\ln [P(n_{\text{sheets}}n_{\text{lonestands}} n_{\text{strands}})]$	n_{sheets} = number of sheets $n_{\text{lonestands}}$ = number of unpaired strands n_{strands} = total number of strands
HS	Helix-strand packing	$\sum_m \sum_n -\ln [P(\phi_{mn}, \psi_{mn} sp_{mn} d_{mn})]$	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	$\sum_i \sum_{j>i} \frac{(r_{ij}^2 - d_{ij}^2)^2}{r_{ij}}; d_{ij} < r_{ij}$	i, j = residue (or centroid) indices d = interatomic distance r = summed van der Waals radii ^h

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

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:

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

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 conformations similar to it (for some definition of “similar”)
- The Rosetta energy functions are approximate free 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

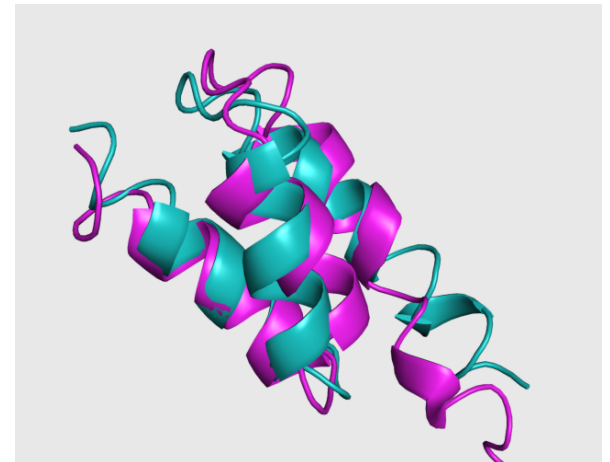
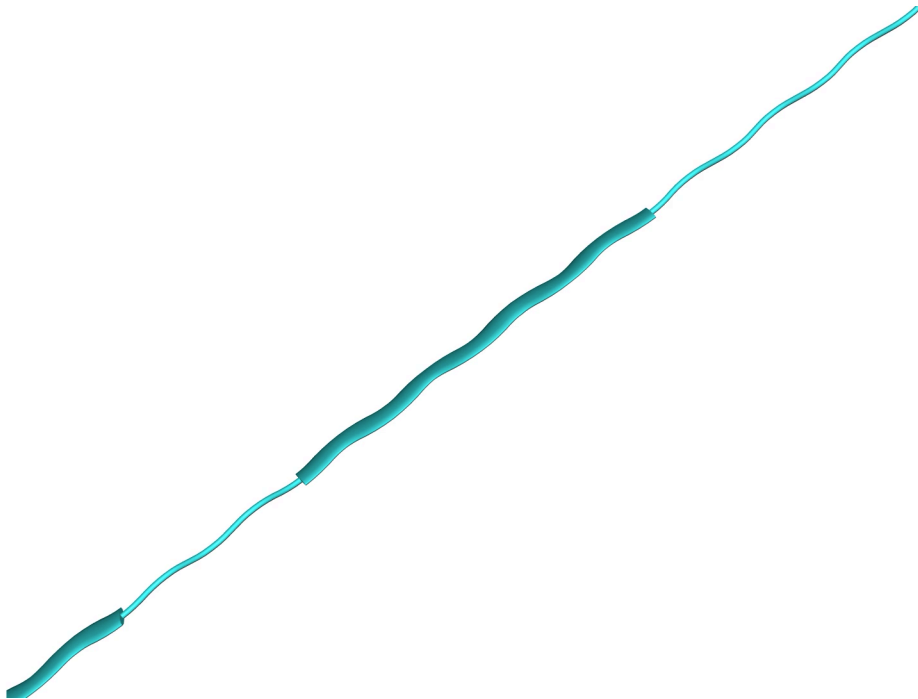
- Uses a large database of 3-residue and 9-residue fragments, taken from structures in the PDB
- Monte Carlo sampling algorithm proceeds as follows:
 1. Start with the protein in an extended conformation
 2. Randomly select a 3-residue or 9-residue section
 3. Find a fragment in the library whose sequence 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

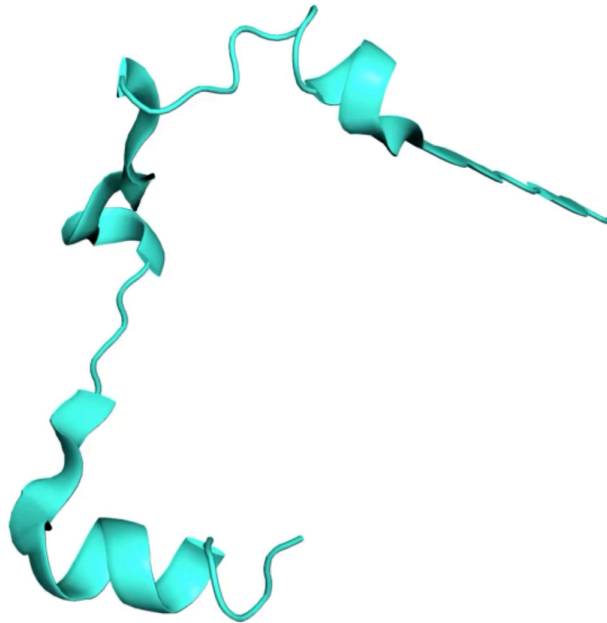
Note: This is not a full Rosetta structure prediction — just initial steps (doesn't include refinement, multiple simulations, etc.)

Example: structure prediction by Rosetta

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

Rosetta energy

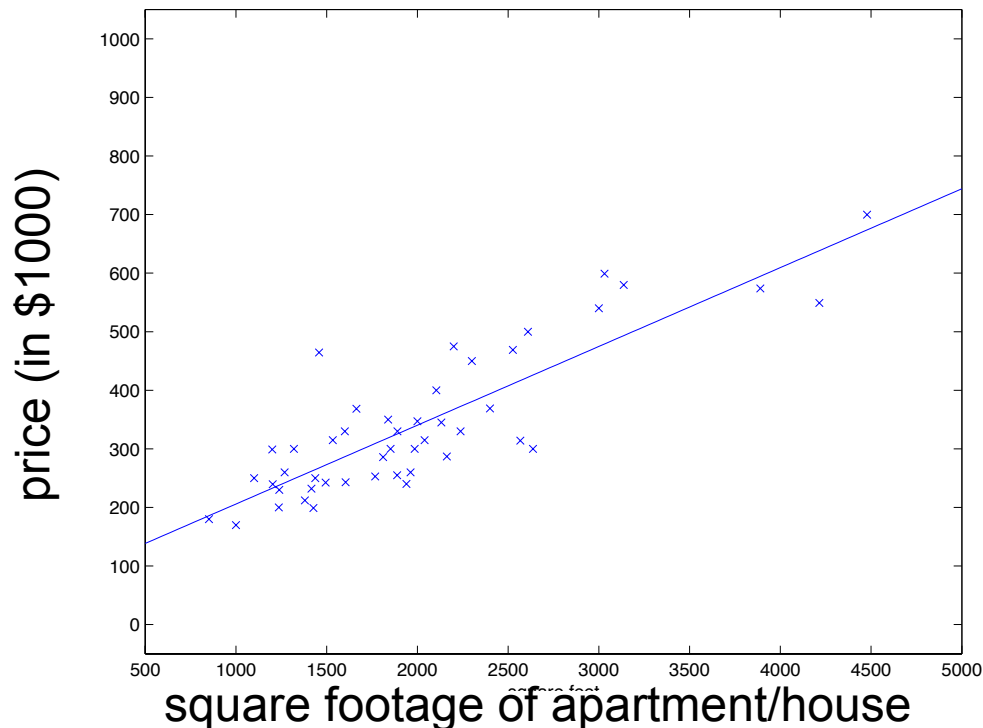
79.126



Deep learning (AI) 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 ~15 years, 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
- This sparked the current AI boom

Deep learning methods for protein structure prediction

First generation: 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

PNAS, 2019

Jinbo Xu^{a,1} TTI Chicago

Improved protein structure prediction using potentials from deep learning

Nature, 2020

<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 Green¹, Chongli Qin¹, Augustin Židek¹, Alexander W. R. Nelson¹, Alex Bridgland¹, Hugo Penedones¹, Stig Petersen¹, Karen Simonyan¹, Steve Crossan¹, Pushmeet Kohli¹, David T. Jones^{2,3}, David Silver¹, Koray Kavukcuoglu¹ & Demis Hassabis¹

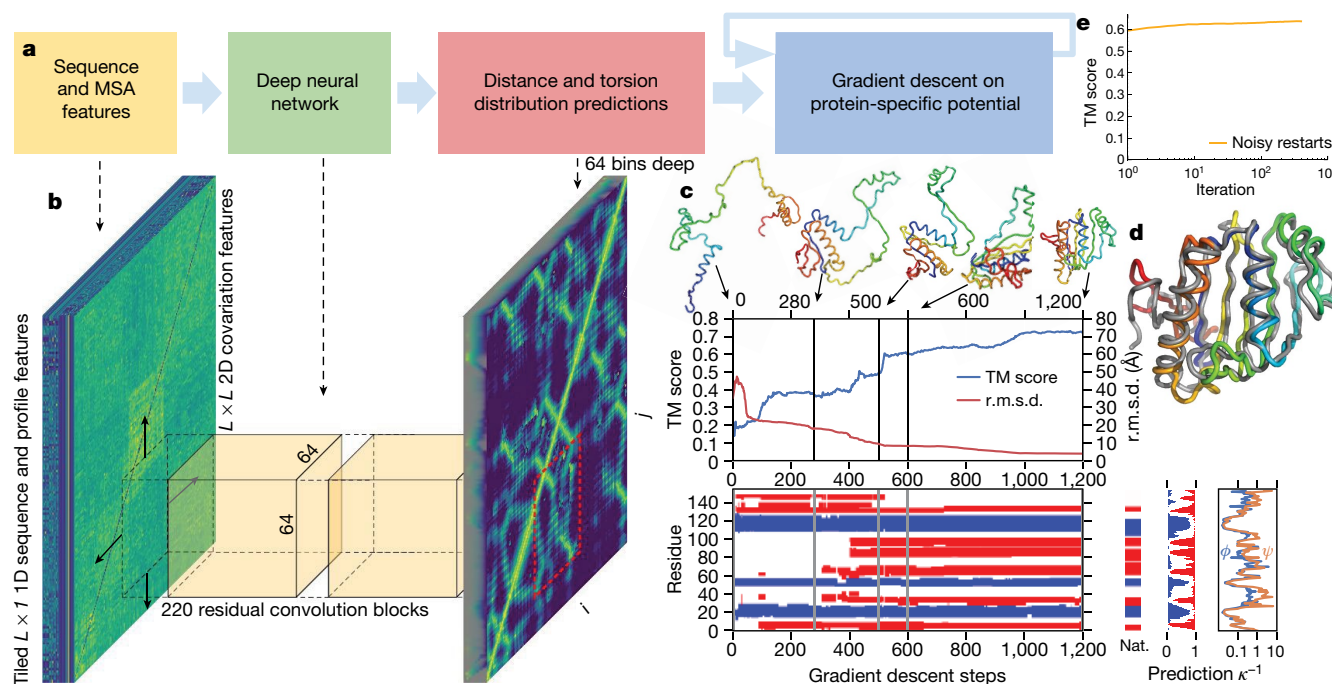
The original AlphaFold (DeepMind)
I'll call this "AlphaFold 1"

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
 - Pre-existing energy functions (e.g., Rosetta all-atom energy function in the case of AlphaFold 1) are also incorporated at this step to ensure that local structural arrangements are physically reasonable.



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
 - Generally no 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:
learning the entire structure
(end-to-end learning)**

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

**The
Guardian**

Dec. 2020

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

≡ **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.

Follow

Journal covers from August 2021



AlphaFold 2
(DeepMind)



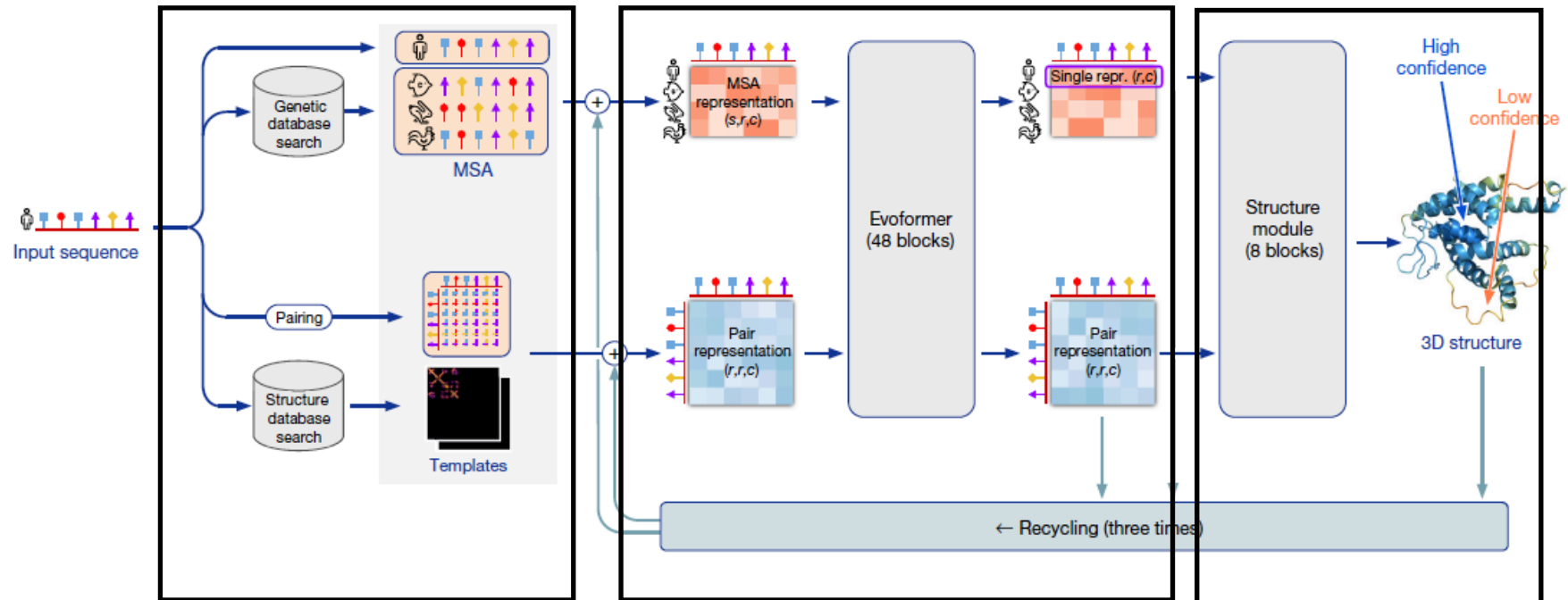
RoseTTAFold
(U. Washington
and academic
collaborators)

- 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: learning the entire structure (end-to-end learning)

- 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



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. Finally, determine conformation of each side chain by predicting its dihedral angles.

AF2 in action



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

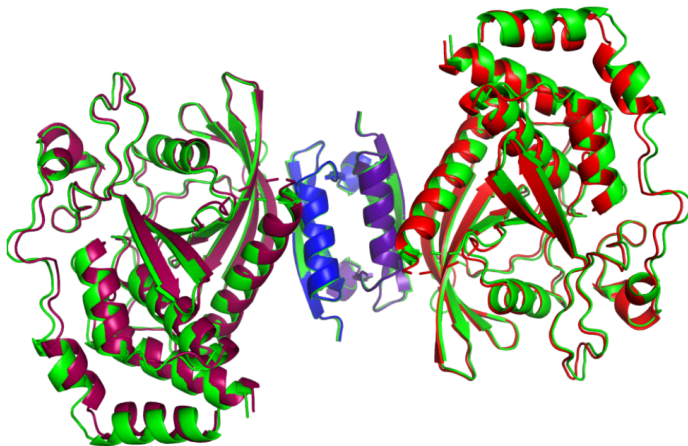
- AF2 doesn't actually "see" most of these intermediate structures. They are guesses of what would have been predicted based on intermediate states (layers) of the network.
- Structure is initially "compressed" because all residues are initially superimposed

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 (AF2 performs a short molecular dynamics simulation with an AMBER force field)

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

Deep learning methods for protein structure prediction

**Third generation:
generalization to other biomolecules**

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)

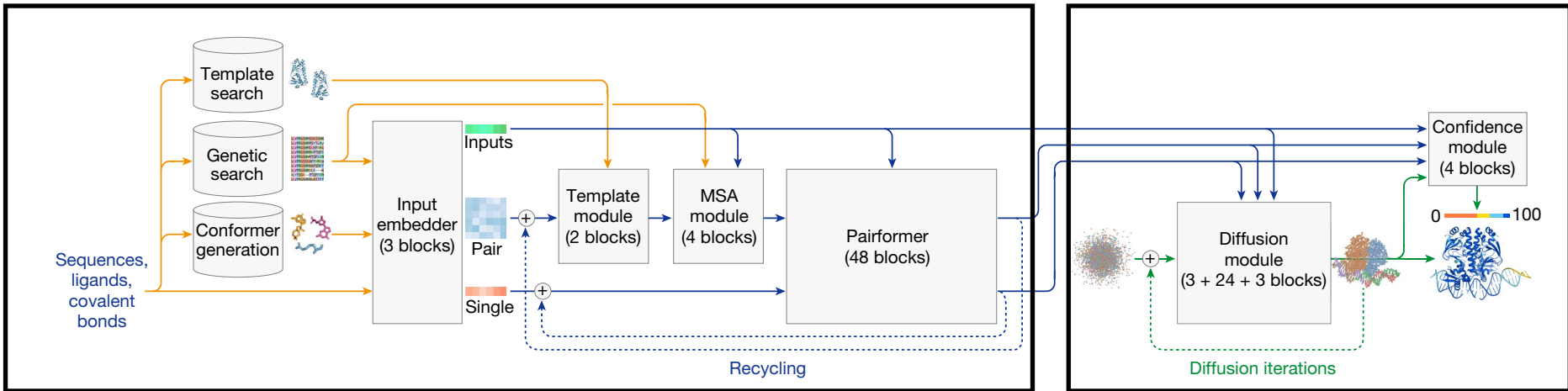
A third generation of deep learning methods for structure prediction goes even further

- AlphaFold 3 (and similar software packages) allow structure prediction for biomolecules and complexes including:
 - Small molecules (e.g., most drugs)
 - Ions (e.g., Na^+ , Cl^- , K^+ , Ca^{2+})
 - 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)

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”

AlphaFold 3 architecture



Similar to AlphaFold 2, with generalizations to handle not only residues but also individual atoms connected by arbitrary covalent bonds

Denoising diffusion module to predict final 3D structure

Abramson et al., *Nature* 630:493–500 (2024)

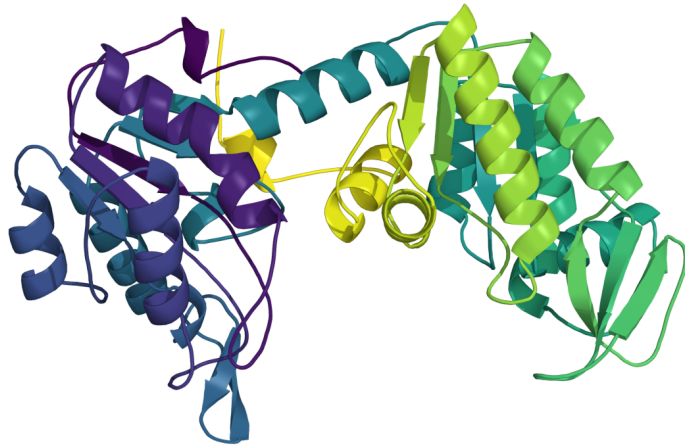
Denoising diffusion

- This is the same machine learning approach used by image generators like DALL-E
 - Originated in Surya Ganguli's lab, Stanford Applied Physics Dept.



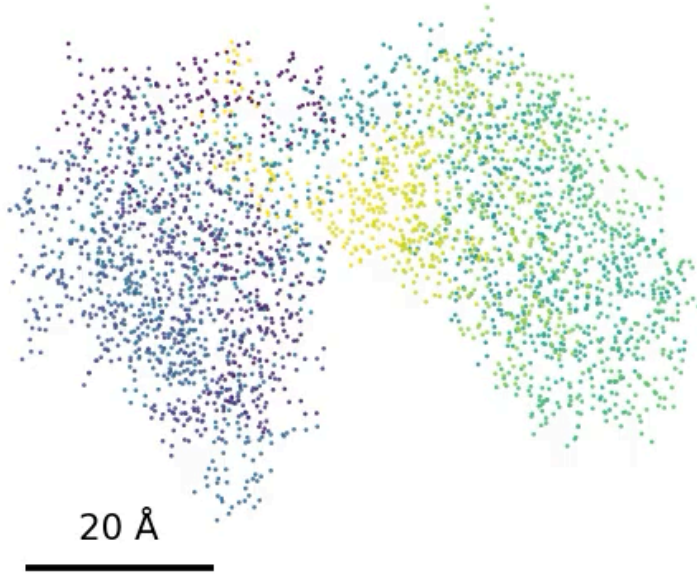
W.D. Heaven, "This avocado armchair could be the future of AI", *Technology Review*, 2021

Adding noise to an experimentally determined structure



20 Å

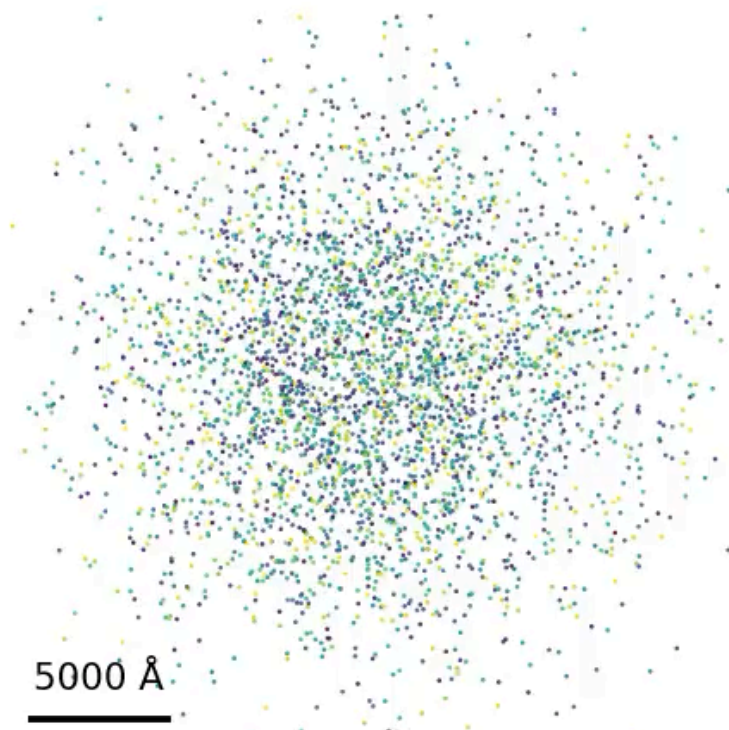
Adding noise to an experimentally determined structure



- Add small random numbers to the x , y , and z coordinates of each atom, over and over
- We end up with completely random coordinates for each atom
- Do this more for many biomolecules
- Train a machine learning method that predicts the position of each atom at the *previous* step given information about the biomolecule

Now start with a random cloud of atoms and predict the correct atom positions, given information about the biomolecule

Step 200/200



Now start with a random cloud of atoms and predict the correct atom positions, given information about the biomolecule



- This process is called “denoising diffusion”

Multiple similar software packages

- Fully open-source: OpenFold 3, Boltz, Protenix
- Servers: AlphaFold Server, Chai, HelixFold

Deep learning methods for protein structure prediction

Large language models

Large language models for protein structure prediction

- Recent work from several groups (most notably, the ESM models from Meta/Evolutionary Scale)
 - Exploit large language models similar in spirit to ChatGPT or Claude
- 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/AF3, but works without explicitly identifying homologs, and learned representation is useful for other purposes.

Strengths and limitations

Is the protein folding problem solved?

- The original “protein folding problem” was determining *how* a protein gets to its folded structure
- AlphaFold and similar methods 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 similar methods

- 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
- Produce measures of local confidence in their predictions

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 biomolecular structure experimentally vs. predicting it computationally?

- Key factors to consider:
 - How long will experimental structure determination take for the biomolecule (or complex) of interest?
 - How do you want to use the structure?
 - How much accuracy do you need?
 - Are the conditions important?
- To get a sense of a protein's overall shape, compare structures of many proteins, or consider how many different molecules bind to a protein, 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 $\neq$ 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