Protein structure prediction (Part I)

CS/CME/BioE/Biophys/BMI 279

Oct. 7, 2021

Ron Dror

Outline

- Why predict protein structure?
- Can we use (pure) physics-based methods?
- Knowledge-based methods
- Two major approaches to protein structure prediction
  - Template-based ("homology") modeling (e.g., Phyre2)
  - *Ab initio* modeling (e.g., Rosetta)
- An additional approach: analysis of multiple sequences (coevolution)
- Structure prediction games

I will include additional material about recent progress in protein structure prediction in a subsequent lecture

Why predict protein structure?

Problem definition

- Given the amino acid sequence of a protein, predict its three-dimensional structure
- Proteins sample many structures. We want the *average* structure, which is roughly what's measured experimentally.

Average structure = the one that the proteins spends most of it time in

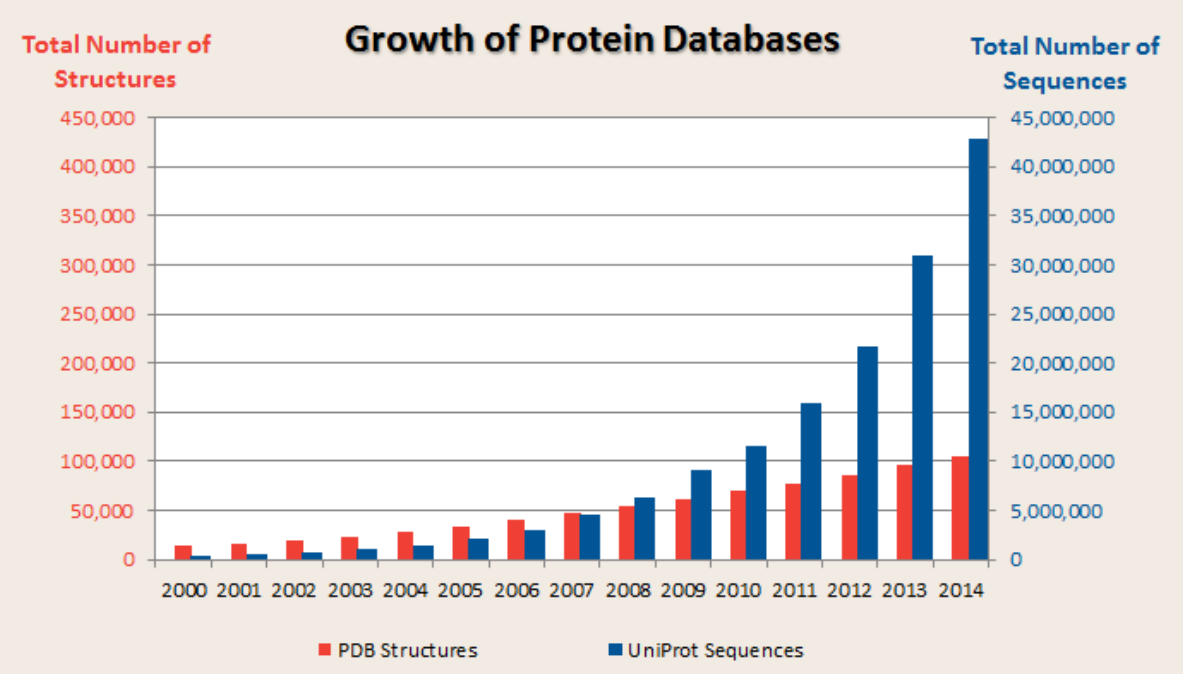
SVYDAAAQLTADVKKDLRDSW
KVIGSDKKGNGVALMTTLFAD
NQETIGYFKRLGNVSQGMAND
KLRGHSITLMYALQNFIDQLD
NPDSLDLVCS.......

How are predicted structures used?

- Drug development
  - Computational screening of candidate drug compounds
  - Figuring out how to optimize a promising candidate compound
  - Figuring out which binding site to target
- Identifying the mechanism by which a protein functions
  - How do genetic mutations alter that function (e.g., cause disease)?
  - How one might alter that protein’s function (e.g., with a drug)?
- 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

Why not just solve the structures experimentally?

- Some structures are very difficult to solve experimentally
  - Sometimes many labs work for decades to solve the structure of one protein
- Sequence determination far outpaces experimental structure determination
  - We already have far more sequences than experimental structures, and this gap will likely grow

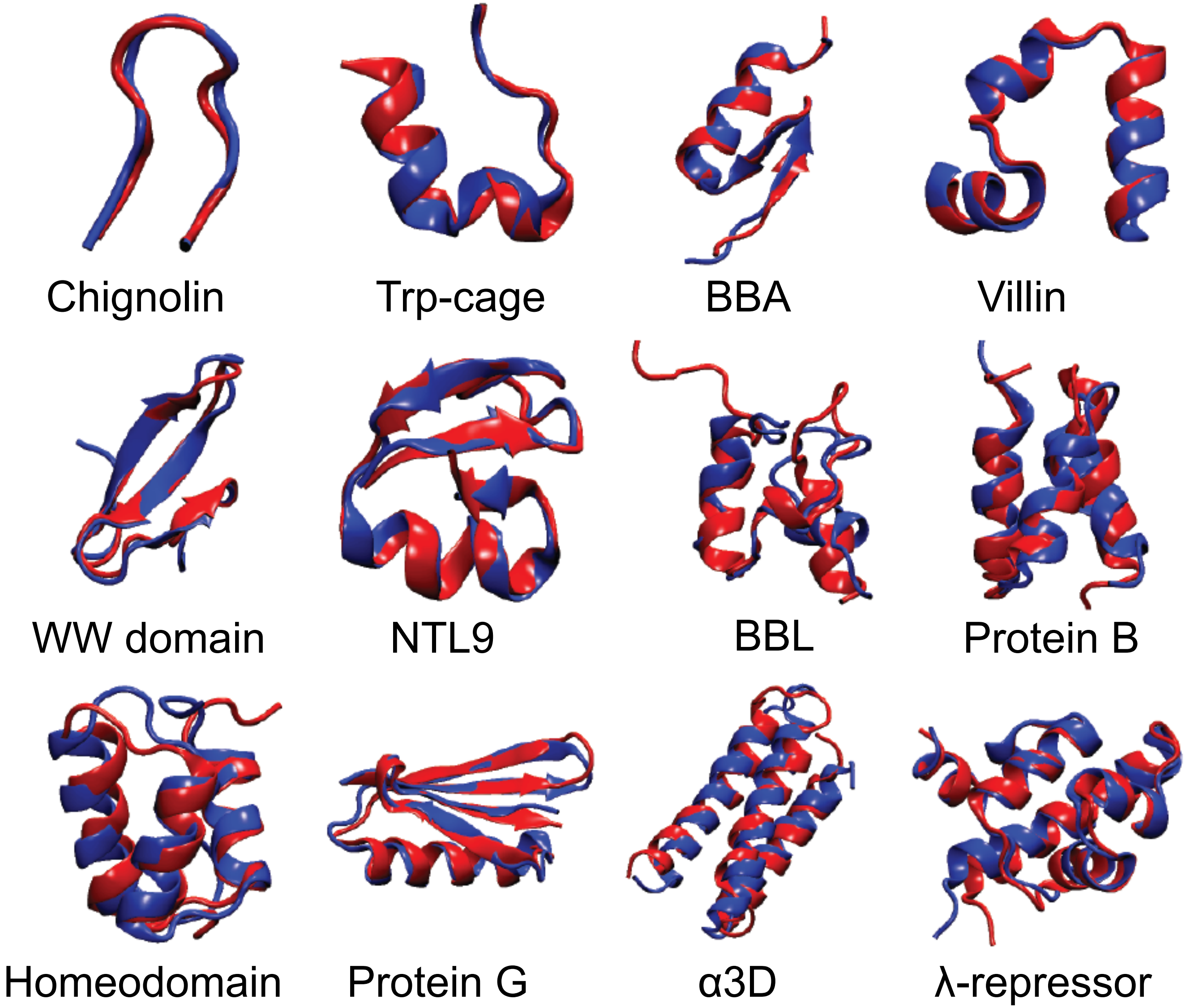


Experimentally derived structures are more accurate but difficult to solve than computationally derived ones.

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 folding is important for understand how the folding 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. Although many people refer to structure prediction as “the protein folding problem,” structure prediction is an easier problem (easier, but still tough!).

Can we find simpler physics-based rules that predict protein structure?

- For example, look at patterns of hydrophobic, hydrophilic, or charged amino acids?
- People have tried for a long time without much success

Knowledge-based methods

Most methods used in practice utilize knowledge based methods.

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

- We have about 150,000 experimentally determined protein structures Can find in PDB (protein data bank)
- Can we use that information to help us predict new structures?
- **Yes!**

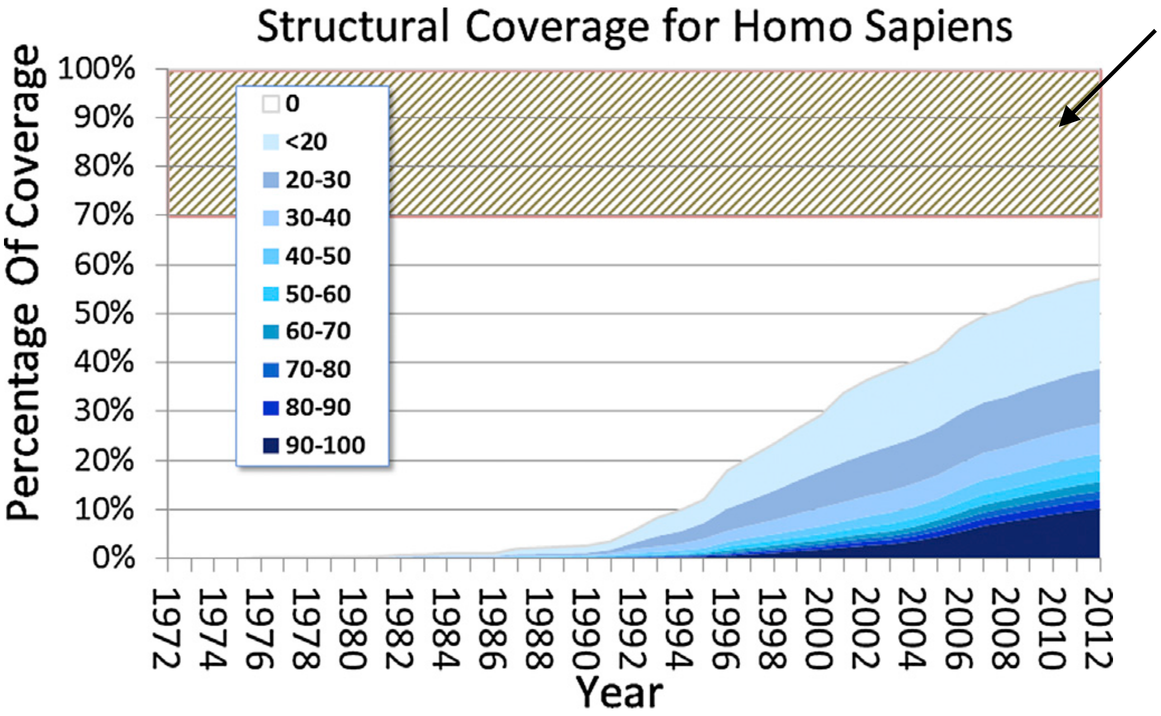
We can also use the >50 million protein *sequences* in the UniProt database

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

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. Sequence identity refers to each amino acid residue
  - **During evolution, sequence changes more quickly than structure**
- Also, there only appear to be 1,000–10,000 naturally occurring protein folds

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



Unstructured
(disordered)
amino acids

The plot shows the fraction of amino acids in human proteins that can be mapped to similar sequences in PDB structures. Different colors indicate % sequence identity.

As this graph stops at 2012, percentages are likely even higher in 2021

Schwede, *Structure* 2013

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

Two major approaches to protein structure prediction

Two major 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 homologs
  - Even *ab initio* approaches usually take advantage of available structural data, but in more subtle ways

Two major approaches to protein structure prediction

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

Template-based structure prediction: basic workflow

Query sequence = seq of protein whose structure you are interested in

- 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.

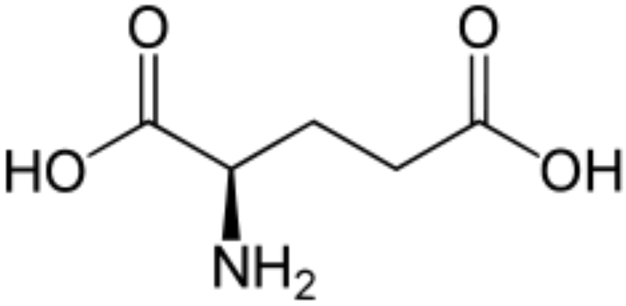
What does it mean for two sequences to be similar?

- 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

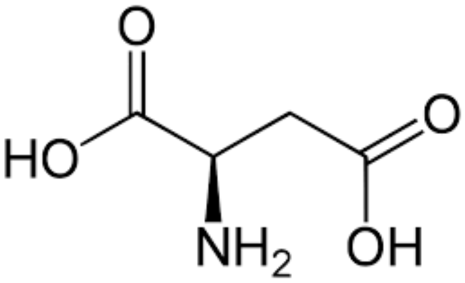
What does it mean for two sequences to be similar?

- 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

Similar in terms of chemical properties (ie. Acidic, basic, nonpolar, aromaticity)



Glutamic acid



Aspartic acid

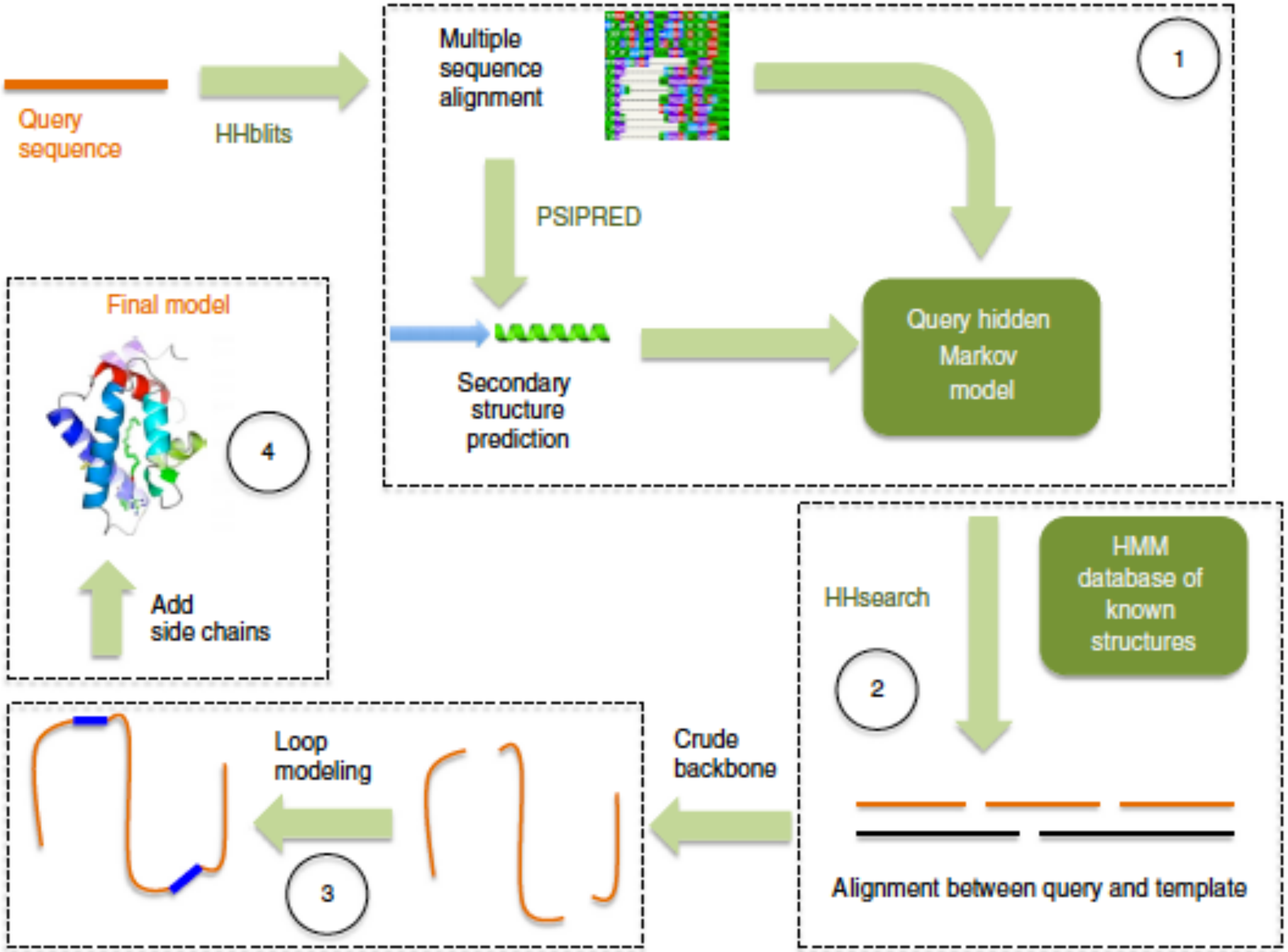
What does it mean for two sequences to be similar?

- 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
    - You are not responsible for knowing about HMMs

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 servers
  - Among the most widely used, and arguably the easiest to use
  - Approach is similar to that of other template-based modeling methods
  - Great name!

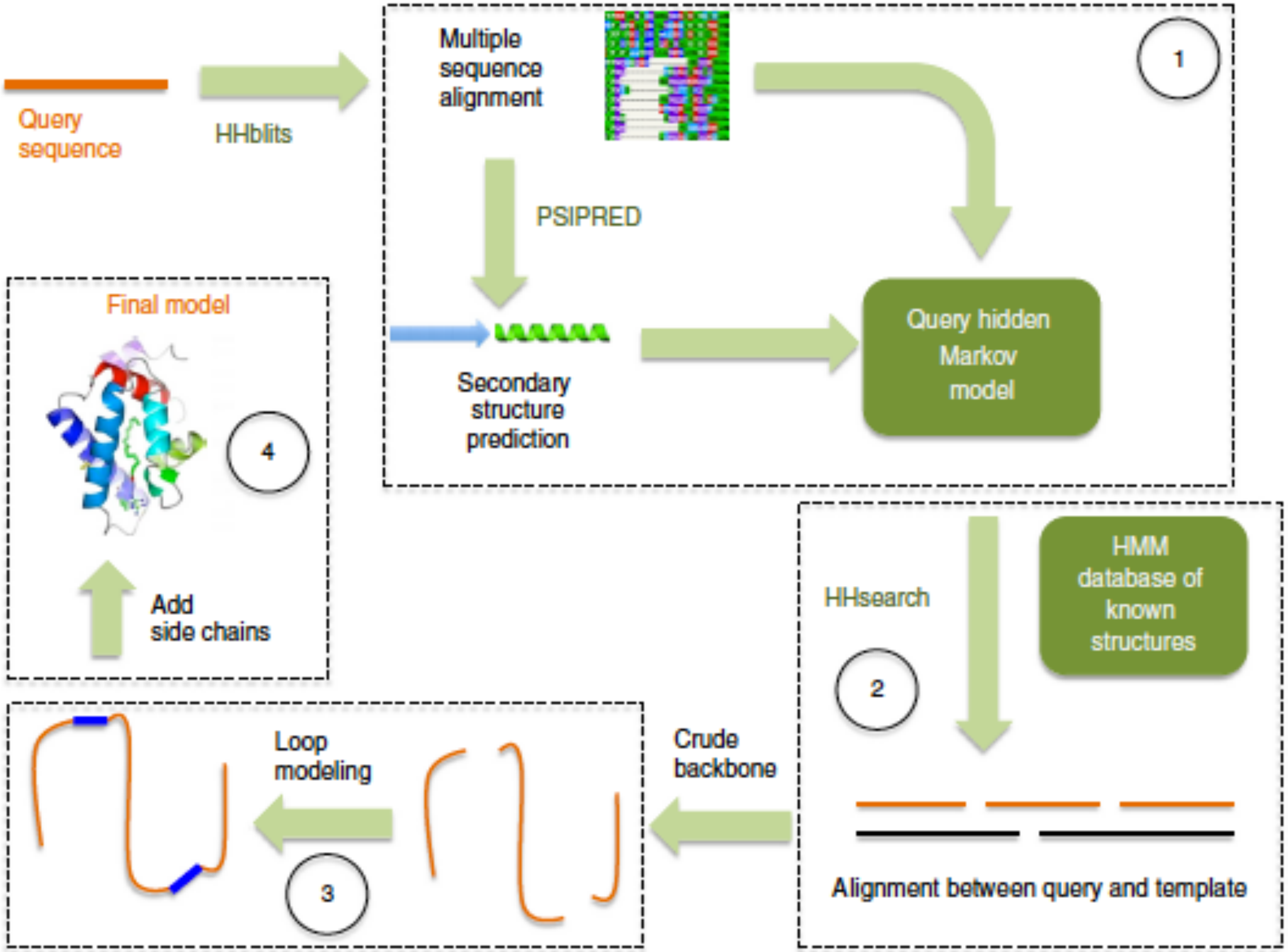
Phyre2 algorithmic pipeline



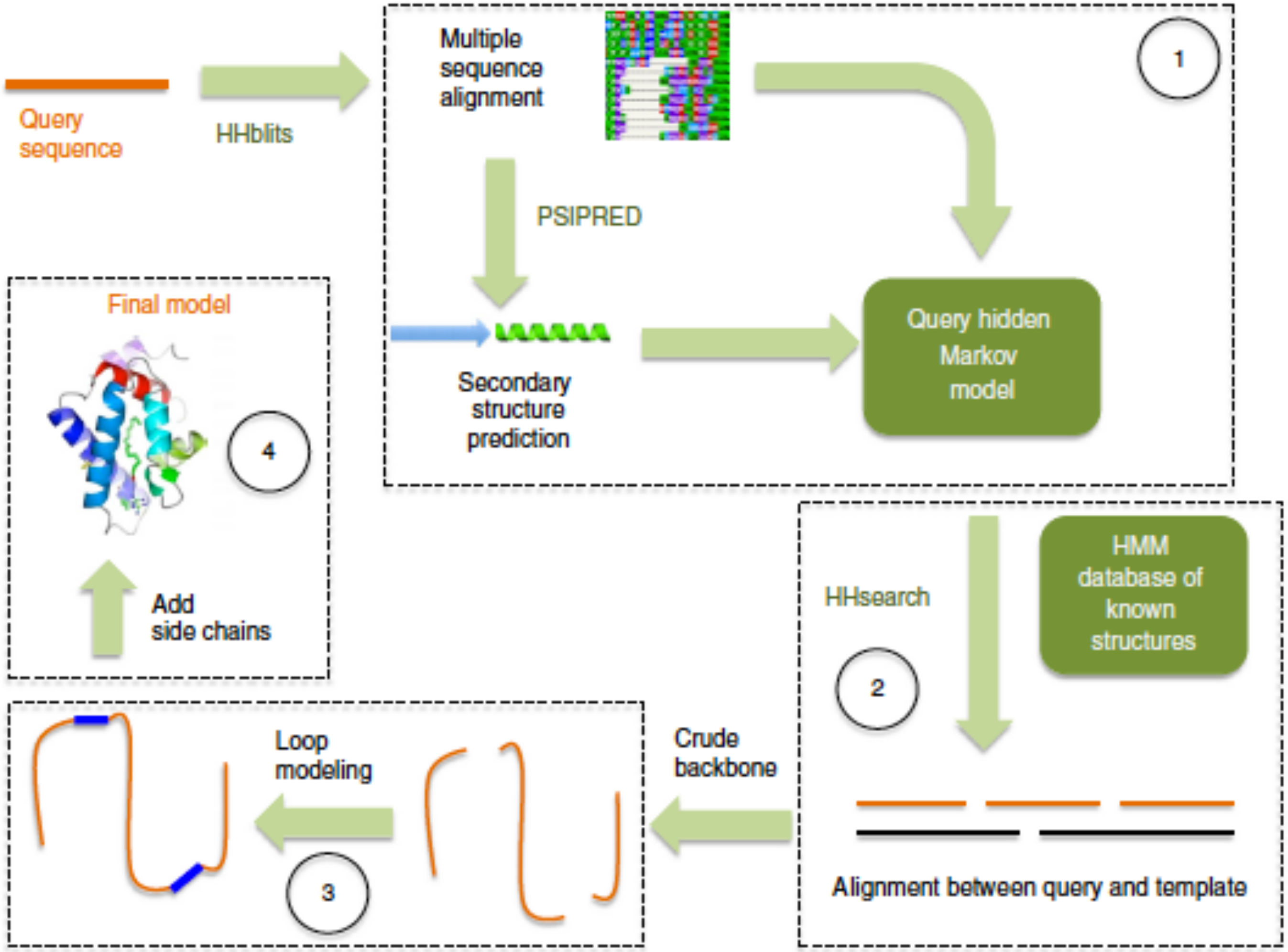
LA Kelley et al., *Nature Protocols* 10:845 (2015)

Phyre2 algorithmic pipeline

Identify similar sequences in
protein sequence database

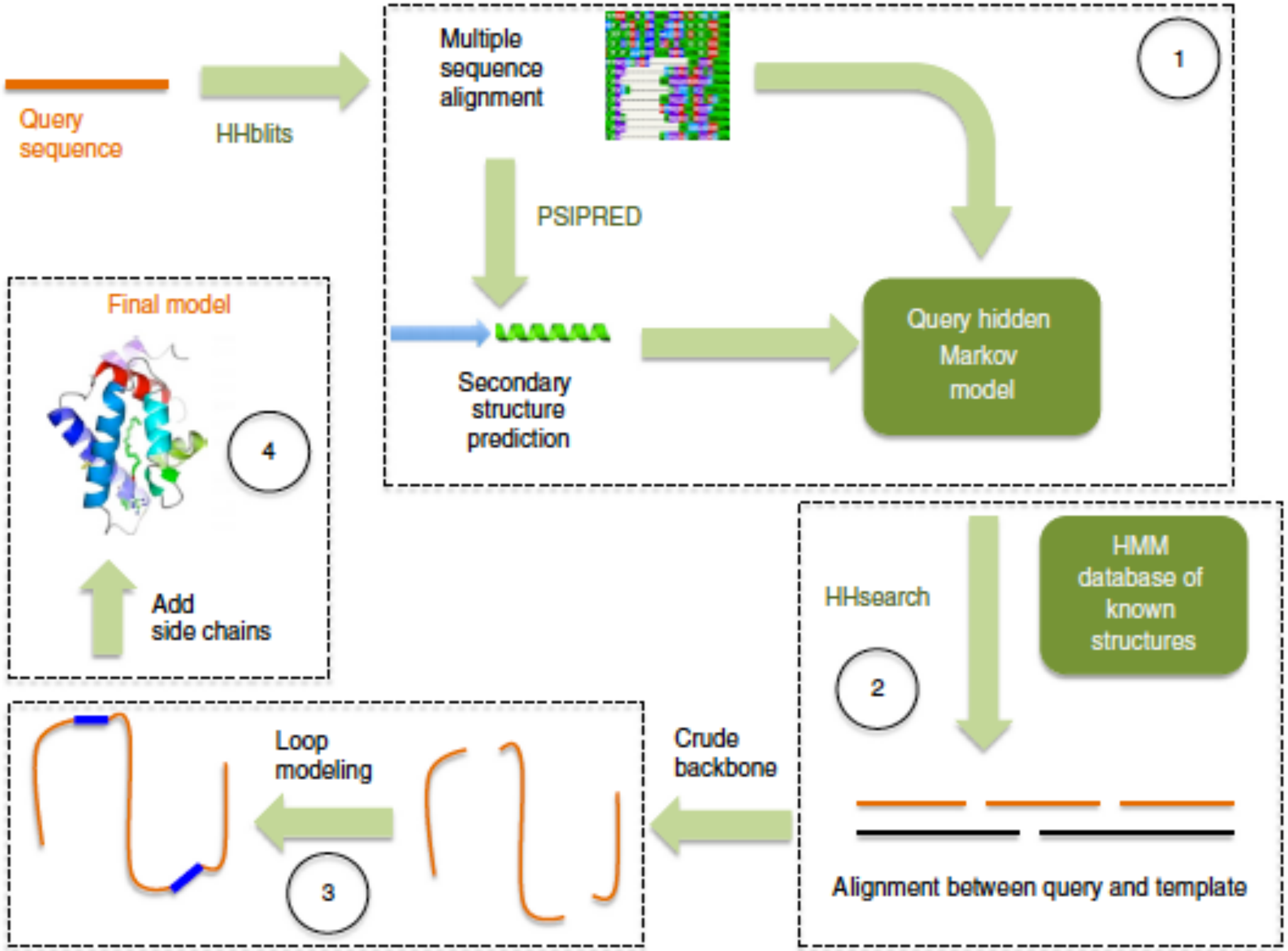


Phyre2 algorithmic pipeline



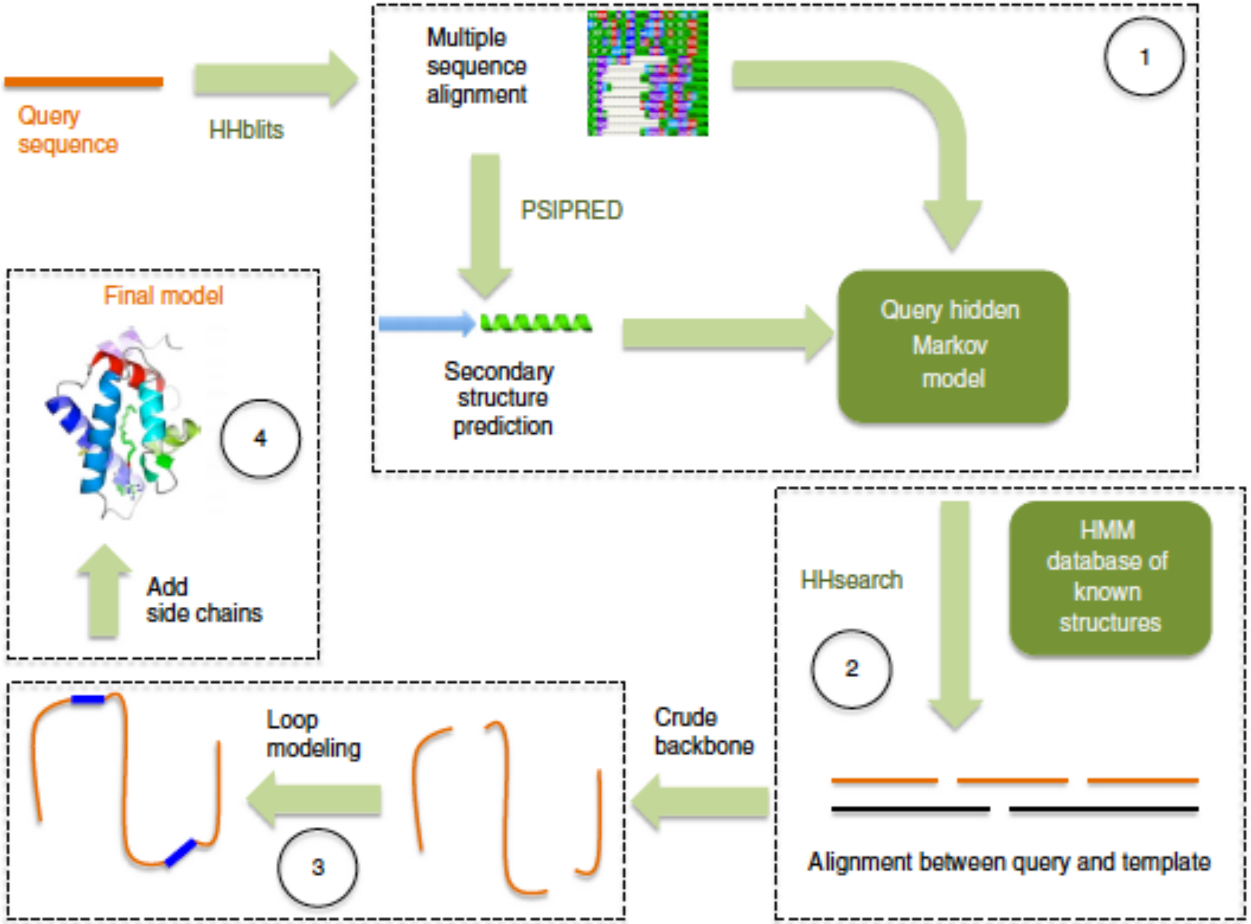
Choose a template structure by:
(1) comparing sequence profiles and
(2) predicting secondary structure for each residue in the query sequence and comparing to candidate template structures. Secondary structure (alpha helix, beta sheet, or neither) is predicted for segments of query sequence using a neural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

Phyre2 algorithmic pipeline



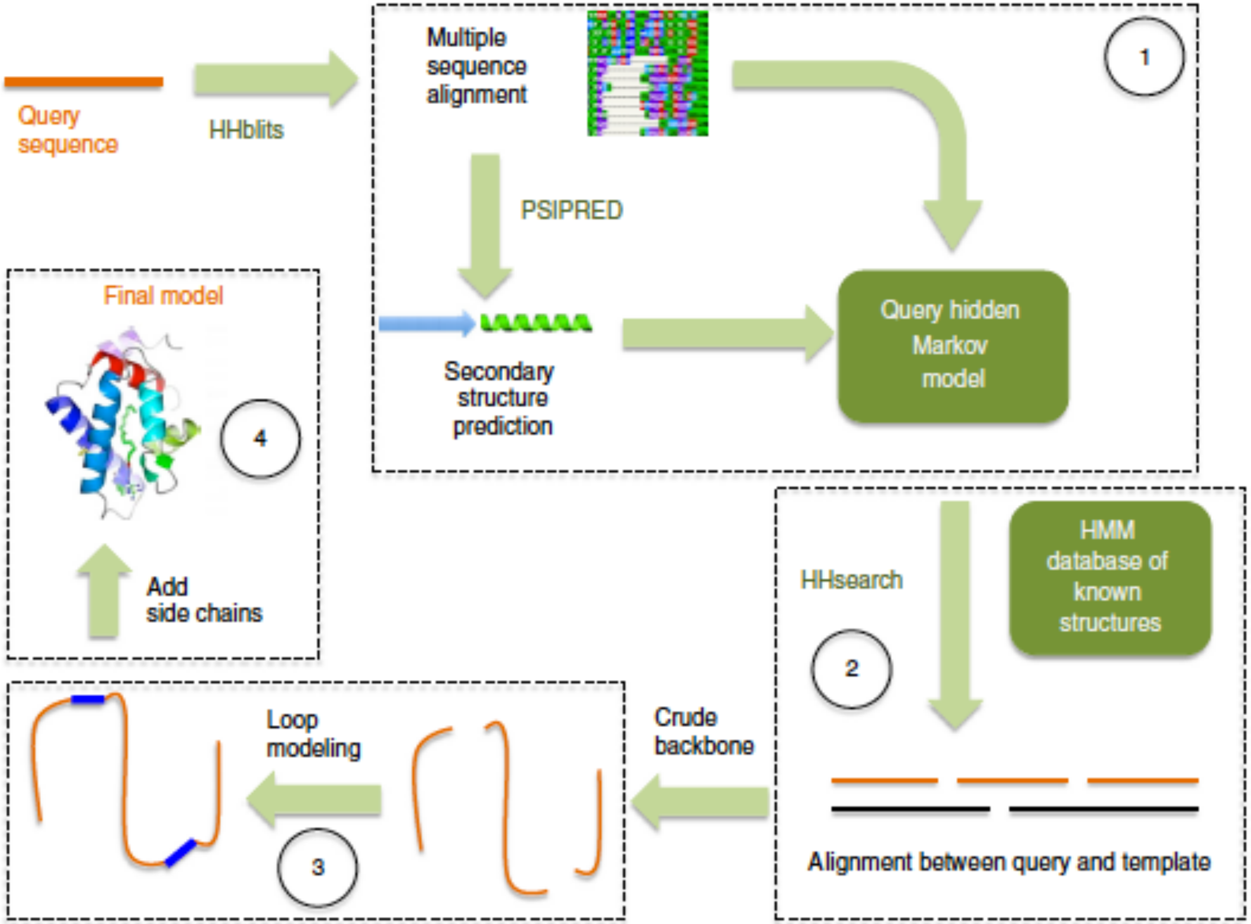
Compute optimal alignment of query sequence to template structure

Phyre2 algorithmic pipeline



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

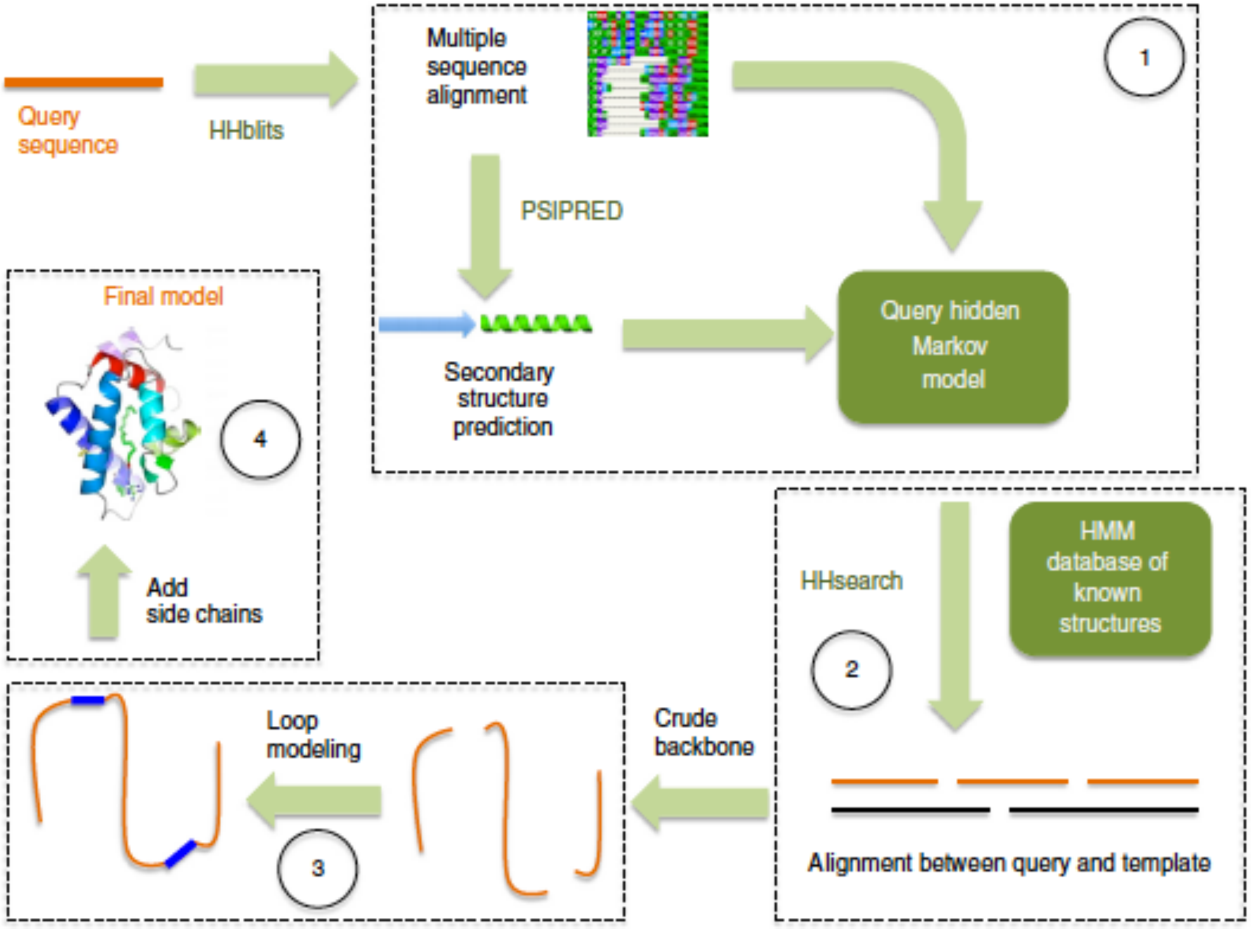
Phyre2 algorithmic pipeline



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.

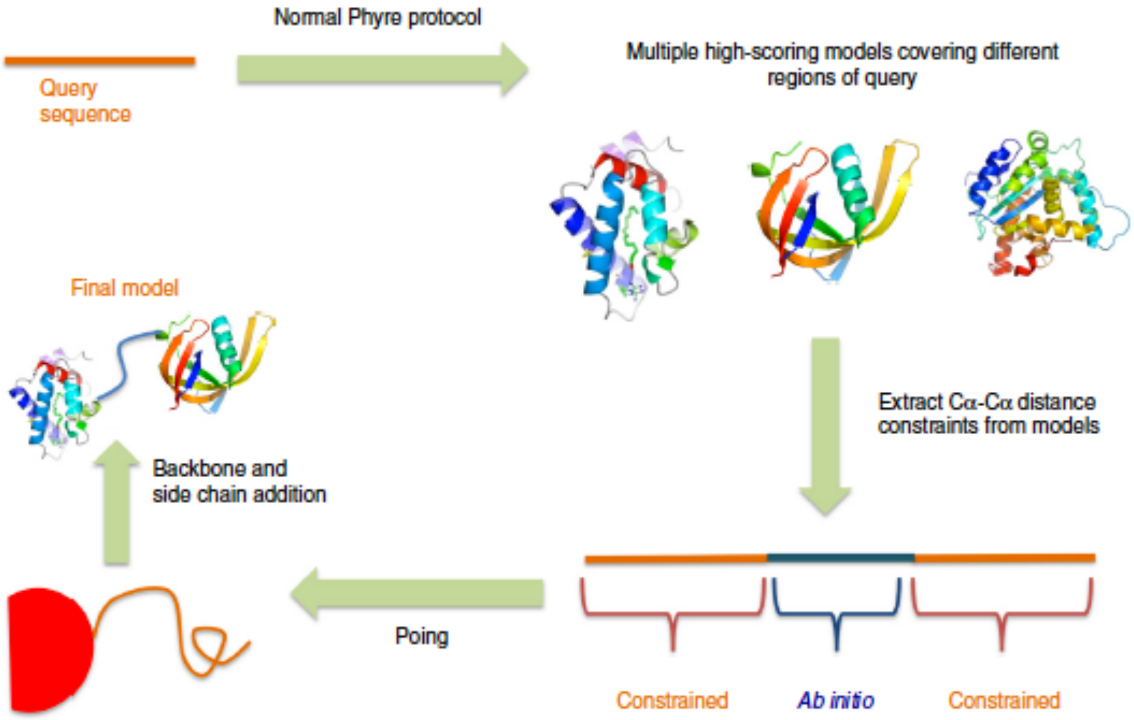
Phyre2 algorithmic pipeline

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



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

You’re not responsible for this

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

Two major approaches to protein structure prediction

Ab initio modeling (e.g., Rosetta)

Two major 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 homologs
  - Even *ab initio* approaches usually take advantage of available structural data, but in more subtle ways

Ab initio structure prediction

- 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 20–25 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, such as protein design and RNA structure prediction

Key ideas behind Rosetta

- Knowledge-based energy function
  - In fact, two of them:

    Gives an approximation of side chains

    - 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

Called "knowledge-based" because these strategies are informed by structures in PDB and are not necessarily homologs of the protein you are predicting

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
    - Centroid position determined by averaging over all structures of that side chain in the PDB

Rosetta energy function

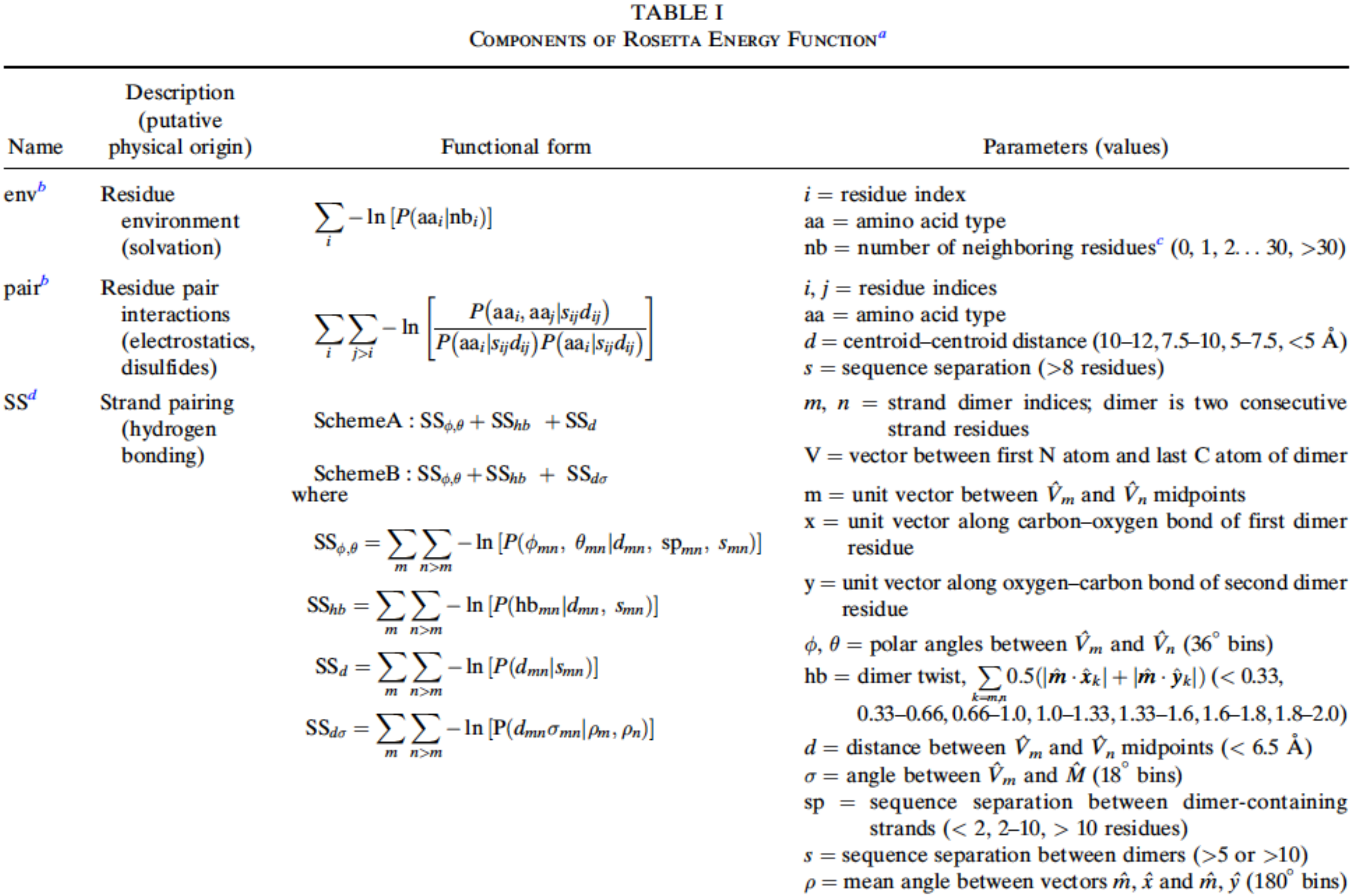


TABLE I

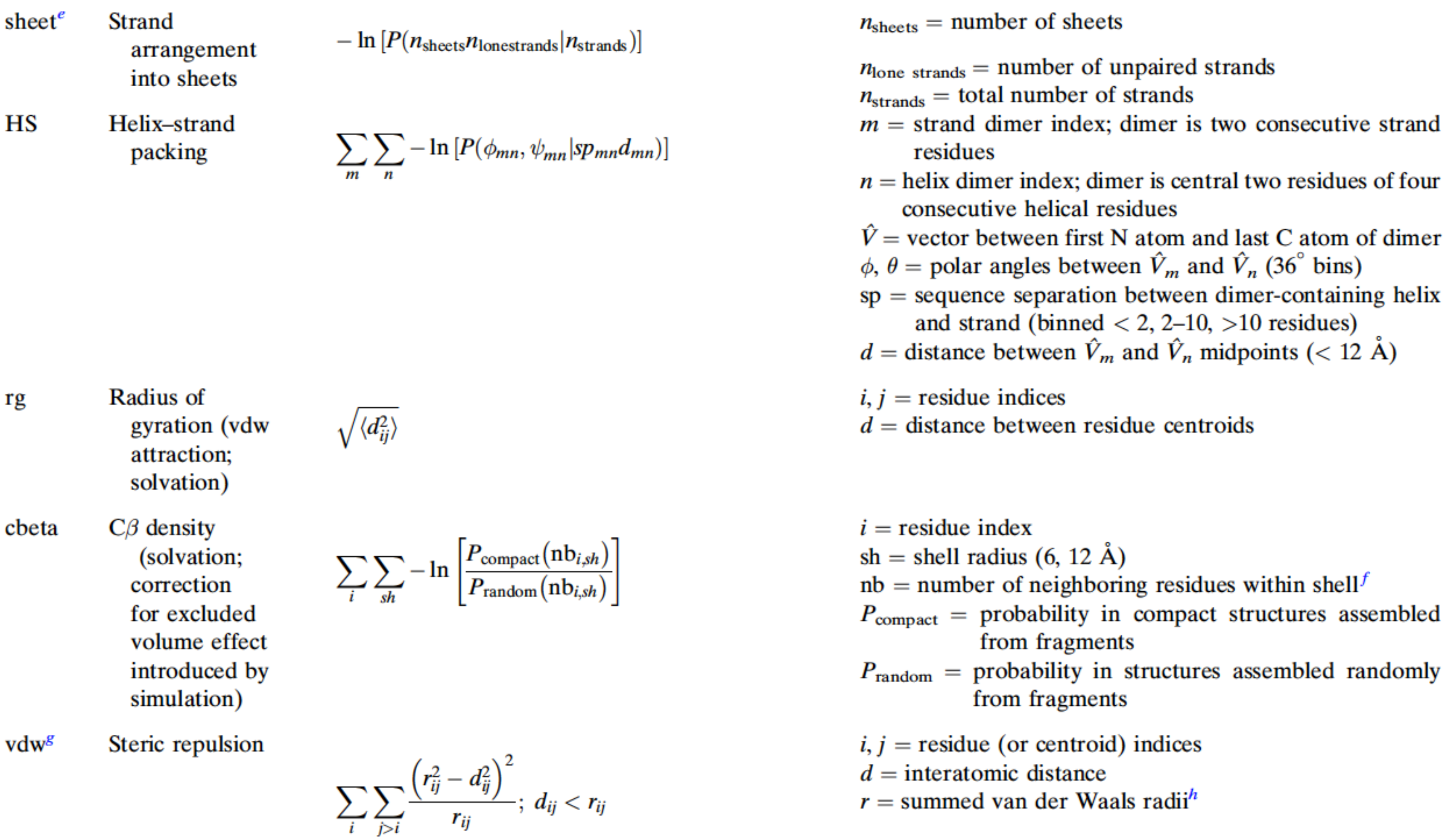
COMPONENTS OF ROSETTA ENERGY FUNCTION[a]

Name	Description (putative physical origin)	Functional form	Parameters (values)
env[b]	Residue environment (solvation)	$\sum_i -\ln[P(\mathrm{aa}_i \mid \mathrm{nb}_i)]$	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)	$\sum_i \sum_{j>i} -\ln\left[\frac{P(\mathrm{aa}_i, \mathrm{aa}_j \mid s_{ij}d_{ij})}{P(\mathrm{aa}_i \mid s_{ij}d_{ij})P(\mathrm{aa}_i \mid s_{ij}d_{ij})}\right]$	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 $\mathrm{SS}_{\phi,\theta} = \sum_m \sum_{n>m} -\ln[P(\phi_{mn}, \theta_{mn} \mid d_{mn}, \mathrm{sp}_{mn}, s_{mn})]$ $\mathrm{SS}_{hb} = \sum_m \sum_{n>m} -\ln[P(\mathrm{hb}_{mn} \mid d_{mn}, s_{mn})]$ $\mathrm{SS}_d = \sum_m \sum_{n>m} -\ln[P(d_{mn} \mid s_{mn})]$ $\mathrm{SS}_{d\sigma} = \sum_m \sum_{n>m} -\ln[\mathrm{P}(d_{mn}\sigma_{mn} \mid \rho_m, \rho_n)]$	m, n = strand dimer indices; dimer is two consecutive strand residues V = vector between first N atom and last C atom of dimer 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, $\sum_{k=m,n} 0.5(\lvert\hat{\boldsymbol{m}}\cdot\hat{\boldsymbol{x}}_k\rvert + \lvert\hat{\boldsymbol{m}}\cdot\hat{\boldsymbol{y}}_k\rvert)$ (< 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



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	$\sum_i \sum_{j>i} \frac{(r_{ij}^2 - d_{ij}^2)^2}{r_{ij}};\ d_{ij} < r_{ij}$	i, j = residue (or centroid) indices d = interatomic distance r = summed van der Waals radii[h]

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

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

You're not responsible for the details!

Rosetta energy function: take-aways

- The (coarse-grained) Rosetta energy function is essentially entirely knowledge-based
  - Based on statistics computed from the PDB
- Many of the terms are of the form $-\log_e[P(A)]$, where $P(A)$ is the probability of some set A
  - This is essentially the free energy of set A. Recall definition of free energy:

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

k_b = Boltzmann's constant

T = temperature

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 energy functions of previous lectures were 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?

- 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

An additional approach: analysis of multiple sequences (coevolution)

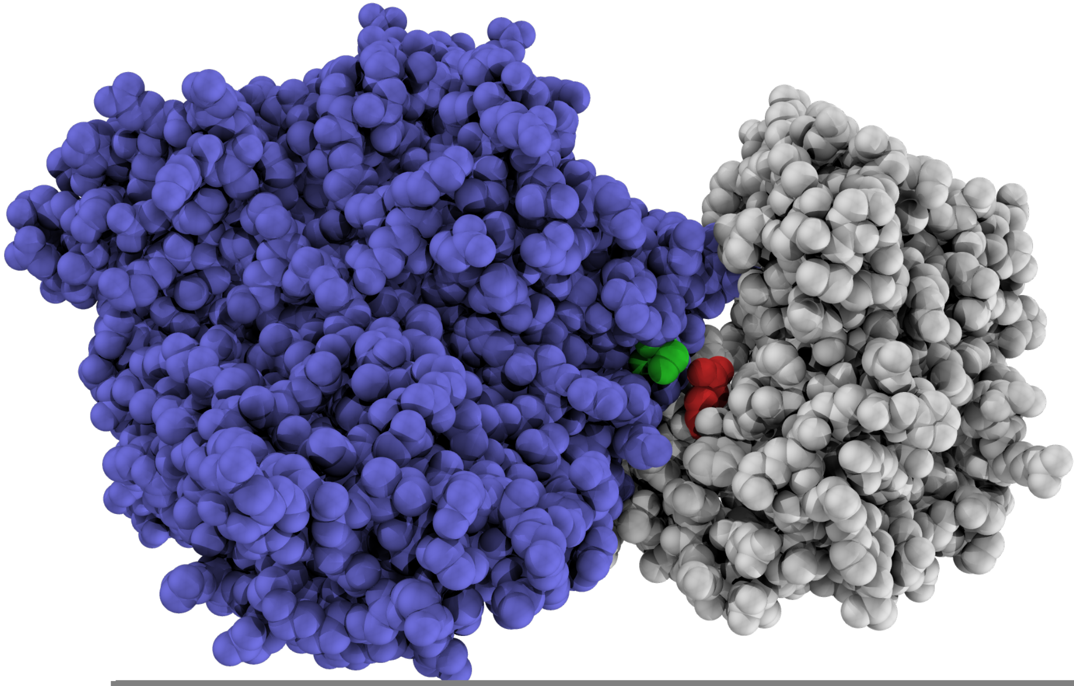
We've discussed two approaches to protein structure prediction

- 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 homologs
  - Even *ab initio* approaches usually take advantage of available structural data, but in more subtle ways

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

Tricky to decide if you want to use template based modeling or ab initio structure prediction…

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

...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**VVTYRHFMNASGG...

...VEALMA**R**VLSYRHFMNASGG...

...VATVMK**Q**VMTYRHYLRATGG...

...VARAMR**E**IGKYAQVLKISRG...

...VPELMQ**D**LTSYRHFMNASGG...

...ADHVLR**R**LSDFVPALLPLGG...

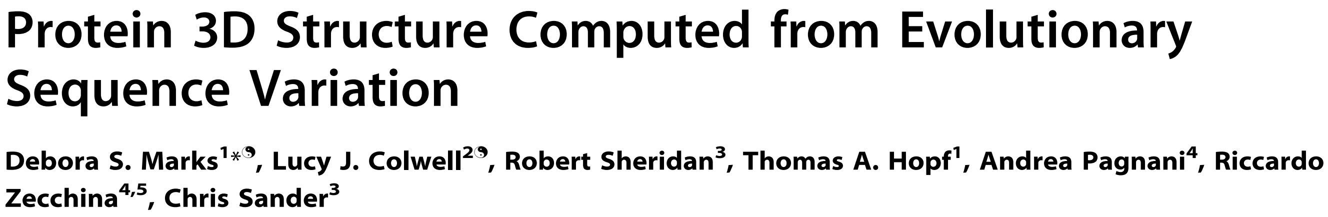
...FERART**A**LEAYAAPLRAMGG...

...VPEVMK**K**VMSYRHYLKATGG...

Can we use this observation to predict structure?

- Given many sequences of related proteins (whose structure is assumed to be similar), look for amino acids that coevolve. They are probably in contact
- This idea has been around for some time, but it has become practically useful recently, thanks to:
  - A dramatic increase in amount of sequence data available Allows you to find more sequences of homologs’
  - Better computational methods

Some key papers on this approach



Protein 3D Structure Computed from Evolutionary Sequence Variation

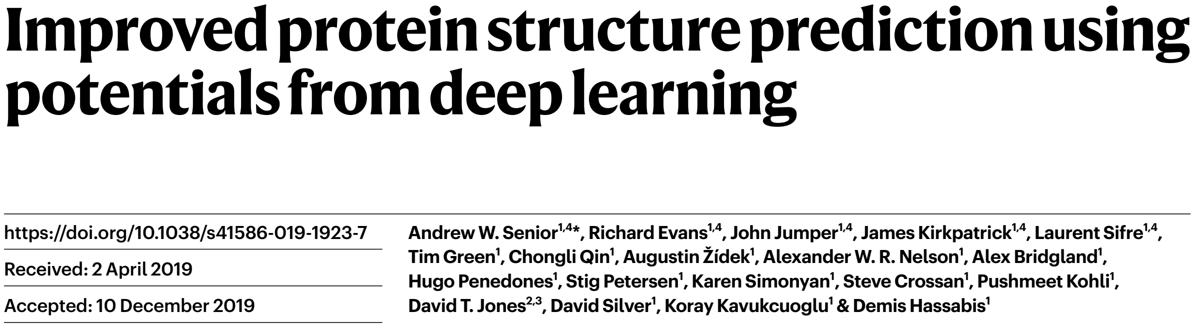
Debora S. Marks[1*], Lucy J. Colwell[2], Robert Sheridan[3], Thomas A. Hopf[1], Andrea Pagnani[4], Riccardo Zecchina[4,5], Chris Sander[3]

PLoS ONE, 2011

Distance-based protein folding powered by deep learning

Jinbo Xu[a,1]

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

This paper from DeepMind describes the original AlphaFold method, but the current AlphaFold method is substantially different, as we'll see later

Structure prediction games

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!

Rank: 119 (soloist)

Score: 9133

123: Grand Challenge 7

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!

EteRNA: RNA design game

- Similar idea, but:
  - For RNA rather than protein.
  - Goal is RNA *design*. Users collective design RNA sequences, which are tested experimentally.
- From Rhiju Das (Stanford) and Adrien Treuille (CMU)

Good. Now boost the rest of the loops with Gs at the lowest numbered nucleotides and solve the rest of the puzzle by turning all of the objective boxes green.