Protein structure prediction

CS/CME/BioE/Biophys/BMI 279
Oct. 8 and 10, 2019
Possu Huang
Ron Dror
Announcements

Meera will be holding a PyMOL tutorial: this Wednesday Oct 9, 6-7pm in 420-041

Due to BioE departmental Retreat
Possu’s office hour this week will be Thursday after lecture
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)
• What’s the best structure prediction method?
• Structure prediction games
• Comparing protein structures
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.

SVYDAAAQLTADVKKDLRDSW
KVIGSDKKGNGVALMTTLFAD
NQETIGYFKRLGNVQSQGMAND
KLRGHSITLMYALQNFIDQLD
NPDSLDLVCS.......
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
Going back to our original problem definition:

*Given an amino acid sequence, predict its 3D structure*

Why? So we can try to figure out its function!

Ironically, you get more of that from just the sequences alone (BIOINFORMATICS!)

How does a protein interact with something?
  Where does a drug bind?
  What drug may bind?
Where to make a mutation? what mutation?
  Interpret experimental data!
  Or use it to obtain experimental data!
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
• Predicting the function of a protein
• Identifying the mechanism by which a protein functions, and 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
Maybe we’ve exhausted the folds

maybe at some point new structures are more or less the same thing...

just maybe, we don’t need to solve another structure...
Can we use (pure) physics-based methods?
Why not just simulate the folding process by molecular dynamics?

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

Chignolin  Trp-cage  BBA  Villin
WW domain  NTL9   BBL  Protein B
Homeodomain  Protein G  α3D  λ-repressor

Lindorff-Larsen et al., Science, 2011
For most proteins, this doesn’t work

1. Folding timescales are usually much longer than simulation timescales.
2. Current molecular mechanics force fields aren’t sufficiently accurate.
3. Disulfide bonds form during the real folding process, but this is hard to mimic in simulation.
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
Basic idea behind knowledge-based (data-driven) methods

- We have experimental structures for over 300,000 proteins.
- 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.
  - 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.

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.

Unstructured (disordered) amino acids
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 main 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
  – Used when one cannot identify any likely homologs of known structure
  – 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

• 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?

• 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) (but 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/](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
Phyre2 algorithmic pipeline

LA Kelley et al.,
*Nature Protocols*
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
Template based modeling

So, how do we find homologs?

Query sequence -> Massive amount of sequences

Protein sequence profile: PSSM (position specific scoring matrix)
So, how do we find homologs?

**Query sequence** → **Massive amount of sequences**

**profile hidden Markov Model**

---

you are not responsible for knowing about HMMs
Hidden Markov Model example (from Wikipedia)

you are not responsible for knowing about HMMs
HMM encodes alignment information (say a protein family), and a new query sequence can be compared to the model for alignments (locally or globally). A giant database of profile HMMs is pFAM. This information also often implies function!

Multiple sequence alignment

```
Sequence 1: FLKLGLSHCILLYF
Sequence 2: FKAPGQTMMFQ
Sequence 3: YPIVGQEBLLG
Sequence 4: FPVVKELAIKL
Sequence 5: FKVLAADVIALD
Sequence 6: LEFISECIIQ
Sequence 7: FKLGLGNVLC
```

**Multiple sequence alignment**

You are not responsible for knowing about HMMs.
A commonly used tool for finding protein homologs

HHBlits (kinda like PSI-BLAST, but faster and more sensitive)

A “template” is a structural hit from this kind of search:
It can be a whole protein or just a segment
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).
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


You’re not responsible for this
Template based modeling methods usually produce superior quality models

Different software packages differ in scoring/schedule/information used, but the ideas are the same.
Q&A

What if there are regions with no coverage? Alignment error?

What if the homologs are all very distant? How do you pick which pieces to keep?

What if the templates all come out different? Which one is correct?
Comparing protein structures
Comparing structures of a protein

• The most common measure of similarity between two structures for a given protein is *root mean squared distance/deviation (RMSD)*, defined as

\[
\sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - w_i)^2}
\]

where \( x \) gives the coordinates for one structure and \( w \) the coordinates for the other

• We generally want to align the structures, which can be done by finding the rigid-body rotation and translation of one structure that will minimize its RMSD from the other
  – The relevant measure of similarity is RMSD after alignment.
An energy function with discriminative power can guide/score a simulation trajectory.
Two major approaches to protein structure prediction

Ab initio modeling (e.g., Rosetta)
Two main 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
  – Used when one cannot identify any likely homologs of known structure
  – 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 15–20 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: $\Phi$, $\Psi$, $\omega$)
  – 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>
<thead>
<tr>
<th>Name</th>
<th>Description (putative physical origin)</th>
<th>Functional form</th>
<th>Parameters (values)</th>
</tr>
</thead>
<tbody>
<tr>
<td>env&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Residue environment (solvation)</td>
<td>$\sum_{i} - \ln [P(aa_i</td>
<td>nb_i)]$</td>
</tr>
<tr>
<td>pair&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Residue pair interactions (electrostatics, disulfides)</td>
<td>$\sum_{i} \sum_{j&gt;i} - \ln \left[ \frac{P(aa_i, aa_j</td>
<td>\theta_{ij}d_{ij})}{P(aa_i</td>
</tr>
</tbody>
</table>
| SS<sup>a</sup> | Strand pairing (hydrogen bonding) | SchemeA: $SS_{\Delta_\phi} + SS_{bb} + SS_{d}$  

SchemeB: $SS_{\Delta_\phi} + SS_{bb} + SS_{sp}$  

where $SS_{\Delta_\phi} = \sum_{m} \sum_{n > m} - \ln \left[ P(\psi_{mn}, \theta_{mn}|d_{mn}, s_{mn}) \right]$  

$SS_{bb} = \sum_{m} \sum_{n > m} - \ln \left[ P(\theta_{mn}|d_{mn}, s_{mn}) \right]$  

$SS_{d} = \sum_{m} \sum_{n > m} - \ln \left[ P(\Delta_\phi|d_{mn}, s_{mn}) \right]$  

$SS_{sp} = \sum_{m} \sum_{n > m} - \ln \left[ P(\sigma_{mn}|\rho_{mn}) \right]$ | $m, n =$ 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 $\vec{V}_{m}$ and $\vec{V}_{m}$ midpoints, $x =$ unit vector along carbon-oxygen bond of first dimer residue, $y =$ unit vector along oxygen-carbon bond of second dimer residue, $\psi, \theta =$ polar angles between $\vec{V}_{m}$ and $\vec{V}_{n}$ (36° bins), $hh =$ dimer twist, $\sum_{i} 0.5|\hat{n}_i \cdot \hat{x}_i| + |\hat{n}_i \cdot \hat{y}_i| (< 0.33, 0.33–0.66, 0.66–1.0, 1.0–1.33, 1.33–1.6, 1.6–1.8, 1.8–2.0$), $d =$ distance between $\vec{V}_{m}$ and $\vec{V}_{n}$ midpoints ($< 6.5$ Å), $\sigma =$ angle between $\vec{V}_{m}$ and $\vec{V}_{n}$ midpoints ($18^\circ$ bins), $sp =$ sequence separation between dimer-containing strands ($< 2, 2–10, > 10$ residues), $s =$ sequence separation between dimers ($>5$ or $>10$), $\rho =$ mean angle between vectors $\vec{n}, \vec{x}$ and $\vec{n}, \vec{y}$ ($180^\circ$ bins) |

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

You’re not responsible for the details!
Rosetta energy function

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

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 event \(A\)
  – This is essentially the free energy of event \(A\). Recall definition of free energy:

\[
G_A = -k_B T \log_e(P(A)) \quad P(A) = \exp\left(\frac{-G_A}{k_B T}\right)
\]
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 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 sometimes described as potential energy functions, but they are closer to approximate free energy functions
  – This means that searching for the “minimum” energy is more valid
  – 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
De novo Modeling with Rosetta

Stage I. Fragment Assembly
*De novo* Modeling with Rosetta

Stage II. All-atom refinement
Rosetta in action
A ~1000-fold increase in computational power

Cα RMSD to native structure

Native (CheY)
A ~1000-fold increase in computational power

Native (CheY)

Cα RMSD to native structure

Rosetta@home

Lowest energy Rosetta structure
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
Driving innovation in protein structure prediction: “CASP”

Critical Assessment of Structure Prediction

Five blind predictions per target

CASP1 (1994)

RMSD: 16.0 Å

From Neil Clarke, CASP7 assessor’s talk

Prediction slides are (mostly) from Rhiju Das
Driving innovation in protein structure prediction: “CASP”

Critical Assessment of Structure Prediction

Five blind predictions per target

DAVID BAKER & colleagues

CASP3 (1998)

RMSD: 4 to 6 Å
CASP6 (2004)

T0281 (1.6 Å over 70 residues)

DAVID BAKER & colleagues
What’s the best structure prediction method?
What’s the best protein structure prediction method?

• Currently, it’s probably I-TASSER
  – http://zhanglab.ccmb.med.umich.edu/I-TASSER/
• I-TASSER is template-based, but it uses *threading*, meaning that when selecting a template it maps the query sequence onto the template structure and evaluates the quality of the fit
  – This allows detection of very remote homology
• I-TASSER combines *many* algorithms
  – It incorporates a surprisingly large number of different components and strategies, including an ab initio prediction module
  – It runs many algorithms in parallel and then looks for a consensus between the results
    • Example: at least seven different threading algorithms
  – Inelegant but effective
Structure prediction games
FoldIt: Protein-folding game

- [https://fold.it/](https://fold.it/)
- Basic idea: allow players to optimize the Rosetta all-atom energy function
  - Game score is negative of the energy
Example of human players outperform automated methods

Each dot is a structure, structural comparison is plotted as Rosetta energy vs. RMSD to native structure. The green cloud is produced by human players, and they achieve lower energy and better RMSD.

Red: Computer model  
Blue: experimental NMR structure  
green: best FoldIt solution
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)