Protein Structure Prediction

The Genes/Proteins Gap

- Large-scale DNA-sequencing initiatives have produced impressive information output on gene sequences -> protein 1D structure
- On the other hand, experimental determination of protein 3D structure is far more difficult (limited information output)
- As a consequence, the knowledge gap between genes and 3D structure of the corresponding proteins is widening…

Rationale

- Why protein structure prediction is important?
- In medicine
  - Comprehension of molecular basis of diseases
  - Drug design
- In biotechnology (e.g. for the design of new enzymes)
  - Emerging discipline: Protein Engineering

- Protein structure prediction is considered today the most important and challenging problem in computational biology (and bioinformatics as well)

The Leventhal Paradox

- Let's consider a small protein with 100 residues
- For the sake of simplicity, let's assume that each peptide bond could assume 3 possible positions:
  - $3^{99} \approx 1.7 \times 10^{47}$ conformations
- Fastest motions $\approx 10^{-15}$ s , so:
  sampling all conformations would take 1.7×10^{32} sec
- How much time is it?
  - $60 \times 60 \times 24 \times 365 = 31536000$ seconds in a year
- Sampling all conformations will take 5.5×10^{24} years!!!

- But…
  each protein folds quickly into a single stable native conformation!

Approaches: Classification

- Ab initio → only basic physics/geometry principles are used
- Comparative Methods (aka template-based methods) → exploitation of information on experimentally known 1D/3D structures
  - Homology Modeling
  - Protein Threading

Ab Initio: Limitations

- Some particular proteins can assume different conformations, depending on the environmental conditions
- Some proteins reach their native state after binding other molecular partners (not known a priori)
- Some proteins reach their native state through the operation of external agents (e.g. chaperons)
- Not always the biologically significant conformation corresponds to a global energy minimum

Comparative Methods: Rationale

- The number of unique structural folds is small (currently <2000, possibly a few thousand)
- 90% of new structures submitted to PDB in the last years have similar folds in PDB

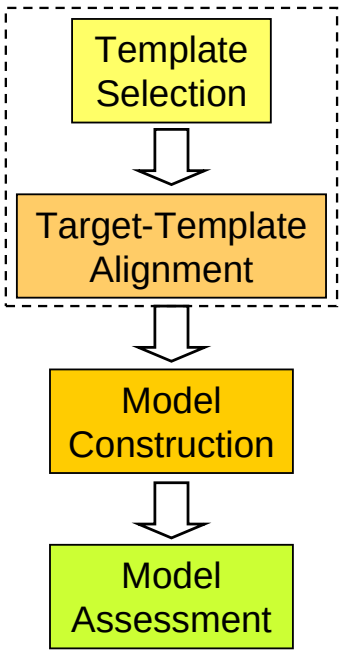
Homology Modeling

- Based on the observation that:
  significant levels of sequence similarity usually imply significant structural similarity.

- It try in the first place to identify one/multiple known protein structures likely to resemble the structure of the target sequence
- Upon the identification of "homologous" proteins, an alignment is obtained that maps the target sequence onto the template one.
- The sequence alignment and template structure are then used to produce a structural model of the target.

- With poor alignment score (<25%), the overall approach fails.

Homology Modeling: Steps



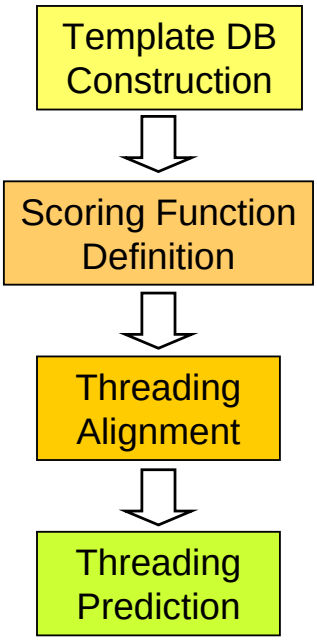
- The first two steps are critical, and usually are based on alignment techniques like FASTA and BLAST, or multiple alignment
- The model construction starts with dealing with "conserved regions", and then performing *loop modeling*.
- Model assessment can be done in different ways, e.g. by exploiting physical potentials or statistical potentials (e.g. based on observed residue-residue contact frequencies)

What's "Threading"?

- "Threading" in this context means placing, aligning each aa in the target sequence onto a position in a template structure

- Main difference between homology modeling and protein threading:
- Threading uses the structure to compute energy function during alignment

Protein Threading: Steps



- Construction of a **structural template database**
- Definition of a **scoring function**, e.g. a sequence–structure energy function
- **Threading alignment**, i.e. alignment of the target sequence with each of the structure templates
- **Threading prediction**, i.e. based on the best-fit selected template, perform local refinements on the target (e.g. on secondary structure, loop prediction, side chains, etc.)

- Report best resulting structural model

PT: Template DB

- How to build up a structural template DB?
- By inspecting PDB, FSSP, SCOP, CATH, select protein structures from the protein structure databases as structural templates.
- Remove pairs of proteins with highly similar structures.
- In some approaches, a template is split into **cores**, i.e. structurally conserved regions, to be used in the alignment algorithms.

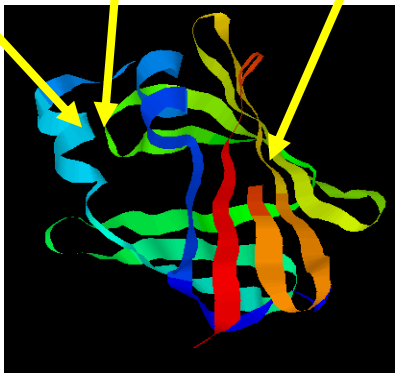
PT: Energy Function

The scoring function has to take into account:

- mutation potential
- environment fitness potential
- pairwise potential
- secondary structure compatibilities
- gap penalties

PT: Energy Function

MTYKLILNGKTKGETTTEAVDAATAEKVFQYANDNGVDGEWTYTE



how preferable to put two particular residues nearby: E_p

alignment gap penalty: E_g

compatibility with local secondary structure prediction: E_{ss}

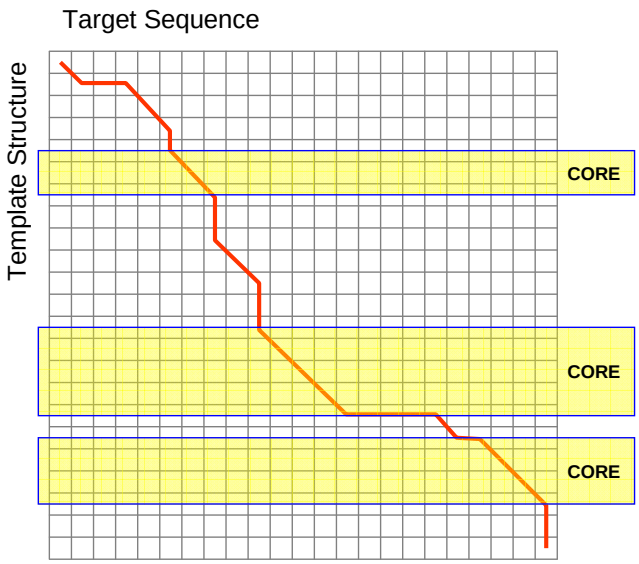
how well a residue fits a structural environment: E_s

how often a residue mutates to the template residue: E_m

total energy: $w_m E_m + w_s E_s + w_p E_p + w_g E_g + w_{ss} E_{ss}$

Exploration of the Threading Search Space

- The alignment is performed for each template in the DB, *optimizing the chosen scoring function*.
- This is the most tough task in the approach, and it has been implemented via dynamic programming and/or integer programming.
- Identification of cores may play an important role.

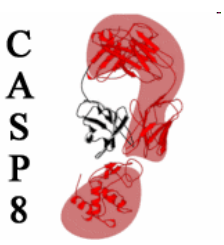


Protein Treading Tools

- One of the most sophisticated tools (RAPTOR) exploit a threading module based on integer programming for best performance

Benchmarking: CASP Contest

- CASP: Critical Assessment of Techniques for Protein Structure Prediction
- It is a community-wide experiment for protein structure prediction taking place every two years since 1994
- Several prediction categories are included:
  - tertiary structure prediction,
  - secondary structure prediction,
  - prediction of structure complexes (CAPRI),
  - residue-residue contact prediction,
  - disordered regions prediction,
  - domain boundary prediction,
  - function prediction,
  - model quality assessment,
  - model refinement



Ab Initio Methods

- Ab initio methods deal with prediction by leveraging physics/geometry principles
- Regular MD is computationally unfeasible in this context
- The problem solution relies on some kind of global optimization procedure, to be used for conformational search
- The search must start from one or more feasible conformations, obtained by a **build-up method**

  Possible forms:
  - Attach one residue after the other, minimize at each step
  - Minimize after the attachment of all residues

Buildup Method (simple backtracking)

```
GenerateStructure(n, L)
    if (n==N) return
    else if (n==0)
            L=AppendVectorToList(L, 0)
            GenerateStructure(n+1, L)
    else
            w=LastElementOfList(L)
            v=GenerateAdjacentVectorOf(w)
            if IsStructureSterицallyFeasible (v, L) then
                    GenerateStructure(n-1, L)
            else
                    L=AppendVectorToList(v, L)
                    GenerateStructure(n+1, L)
```

Probabilistic Choice

BACKTRACKING

ADVANCE

Heuristic Methods: Simulated Annealing

```
SAConformationalSearch(V, T0, Tf, DT)
    T=T0
    Phi= GenerateStructure(N, 0)
    while T>Tf
        Psi=GenerateMutant(Phi)
        if V(Psi) < V(Phi) then
            Phi=Psi
        else
            r=randomZeroOne()
            b=exp( -(V(Psi)-V(Phi) )/T)
            if r < b then
                Phi=Psi
        T=T*DT
```

Boltzmann factor for acceptance of higher-energy conformation

METROPOLIS STEP

TD is a % decrease, e.g. 0.99

Heuristic Methods: Genetic Algorithm

```
GenConformationalSearch(V, beta, Pop, M, pm, pc)
    P = GeneratePopulation (Pop)
    for i in range(M)
        P=SelectionOfConformations(P, Pop, V, beta)
        foreach Phi in P
            Phi=GenerateMutant(Phi, pm)
        P2=GeneratePairsOfConformations(P)
        foreach (Phi1, Phi2) in P2
            (Phi1,Phi2)=GenerateDescendsWithProb((Phi1,Phi2), pc)
            P.add( (Phi1, Phi2) )
```

#Pop members (p) are selected with probability $\exp(-\beta V(p_0)) / \Sigma_{Pop} \exp(-\beta V(p))$

- the lower V, the more likely -

Mutants are “neighbors”, obtained applying changes with probability pm

CROSSOVER