Structure Prediction in 1D

[Based on Structural Bioinformatics, chapter 28]

Protein Structure
- Amino-acid chains fold to form 3d structures
- Proteins are sequences that have (more or less) stable 3-dimensional configuration
- Structure is crucial for function:
  - Area with a specific property
  - Enzymatic pockets
  - Firm structures

Levels of structure:
primary structure

Levels of structure:
secondary structure

Levels of structure:
tertiary and quaternary structure

Ramachandran Plot
Determining Structure

- X-Ray and NMR methods allow to determine the structure of proteins and protein complexes
- These methods are expensive and difficult [several months to process one protein]
- A centralized database (PDB) contains all solved protein structures (www.rcsb.org/pdb/)
  - XYZ coordinate of atoms within specified precision
  - ~31,000 solved structures

Sequence from structure

All information about the native structure of a protein is coded in the amino acid sequence + its native solution environment.

Can we decipher the code?

No general prediction of 3d from sequence yet.

Anfinsen, 1973

One dimensional prediction

Project 3d structure onto strings of structural assignments
- A simplification of the prediction problem

Examples:
- Secondary structure state for each residue [α, β, L]
- Accessibility of each residue [buried, exposed]
- Transmembrane helix

Define secondary structure

3D protein coordinates may be converted into a 1D secondary structure representation using DSSP or STRIDE

DSSP = Database of Secondary Structure in Proteins
STRIDE = Secondary Structure IDentification method
Labeling Secondary Structure

Use both hydrogen bond patterns and backbone dihedral angles to label secondary structure tags from XYZ coordinate of amino-acids
- Do not lead to absolute definition of secondary structure

Prediction of Secondary Structure

Input: Amino-acid sequence

Output: Annotation sequence of three classes [alpha, beta, other (sometimes called coil/turn)]

Measure of success: Percentage of residues that were correctly labeled

Accuracy of 3-state predictions

Q3-score = % of 3-state symbols that are correctly measured on a "test set"

Test set = An independent set of cases (proteins) that were not used to train, or in any way derive, the method being tested.

Best methods
PHD (Burkhard Rost): 72-74% Q3
Psi-pred (David T. Jones): 76-78% Q3

What can you do with a secondary structure prediction?

1. Find out if a homolog of unknown structure is missing any of the SS (secondary structure) units, i.e. a helix or a strand.
2. Find out whether a helix or strand is extended or shortened in the homolog.
3. Model a large insertion or terminal domain
4. Aid tertiary structure prediction

Statistical Methods

From PDB database, calculate the propensity for a given amino acid to adopt a certain ss-type

\[ p(a,aa) = \frac{P(a|aa)}{p(a)} = \frac{p(aa)p(a)}{p(a)} \]

Example:

#Ala=2,000, #residues=20,000, #helix=4,000, #Ala in helix=500

\[ p(aa) = 500/20,000, p(a) = 4,000/20,000, p(aa) = 2,000/20,000 \]

\[ P = \frac{500}{(4,000/10)} = 1.25 \]

Used in Chou-Fasman algorithm (1974)

Chou-Fasman Parameters

<table>
<thead>
<tr>
<th>Residue</th>
<th>Ala</th>
<th>Val</th>
<th>Glu</th>
<th>Met</th>
<th>Ile</th>
<th>Gly</th>
<th>Arg</th>
<th>Asp</th>
<th>His</th>
<th>Thr</th>
<th>Ser</th>
<th>Cys</th>
<th>Tyr</th>
<th>Phe</th>
<th>Asn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glu</td>
<td>1.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Met</td>
<td>1.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ile</td>
<td>1.40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gly</td>
<td>1.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg</td>
<td>1.21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asp</td>
<td>1.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>His</td>
<td>1.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thr</td>
<td>1.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ser</td>
<td>1.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cys</td>
<td>1.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tyr</td>
<td>1.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phe</td>
<td>1.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asn</td>
<td>1.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pro</td>
<td>0.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Val</td>
<td>0.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ile</td>
<td>0.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Chou-Fasman: Initiation**

- Identify regions where 4/6 have propensity \( P(H) > 1.00 \)
- This forms a "alpha-helix nucleus"

<table>
<thead>
<tr>
<th>T</th>
<th>S</th>
<th>P</th>
<th>T</th>
<th>A</th>
<th>E</th>
<th>L</th>
<th>M</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>77</td>
<td>57</td>
<td>69</td>
<td>142</td>
<td>151</td>
<td>121</td>
<td>145</td>
<td>98</td>
<td>77</td>
<td>69</td>
<td>57</td>
</tr>
</tbody>
</table>

**Chou-Fasman: Propagation**

- Extend helix in both directions until a set of four residues have an average \( P(H) < 1.00 \).

<table>
<thead>
<tr>
<th>T</th>
<th>S</th>
<th>P</th>
<th>T</th>
<th>A</th>
<th>E</th>
<th>L</th>
<th>M</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>77</td>
<td>57</td>
<td>69</td>
<td>142</td>
<td>151</td>
<td>121</td>
<td>145</td>
<td>98</td>
<td>77</td>
<td>69</td>
<td>57</td>
</tr>
</tbody>
</table>

**Chou-Fasman Prediction**

- Predict as \( \alpha \)-helix segment with
  - \( E[P_{\alpha}] > 1.03 \)
  - \( E[P_{\alpha}] > E[P_{\beta}] \)
  - Not including Proline
- Predict as \( \beta \)-strand segment with
  - \( E[P_{\beta}] > 1.05 \)
  - \( E[P_{\beta}] > E[P_{\alpha}] \)
- Others are labeled as turns/loops.

(Various extensions appear in the literature)

http://fasta.bioch.virginia.edu/o_fasta/chofas.htm

- Achieved accuracy: around 50%
- Shortcoming of this method: ignoring the context of the sequence when predicting using amino-acids
- We would like to use the sequence context as an input to a classifier
- There are many ways to address this.
- The most successful to date are based on neural networks

**A Neuron**

![Image of a neuron](image)

**Artificial Neuron**

Input: \( a_1, a_2, \ldots, a_n \)

Output: \( f(b + \sum W_i a_i) \)

- \( W_i \) = weights assigned to inputs; \( b \) = internal "bias"
- \( f \) = output function (linear, sigmoid)
Artificial Neural Network

Input | Hidden | Output
--- | --- | ---
\(a_1\) | \(\cdots\) | \(a_k\) | \(\cdots\) | \(a_m\)

Neurons in hidden layers compute "features" from outputs of previous layers
Output neurons can be interpreted as a classifier

Example: Fruit Classifier

<table>
<thead>
<tr>
<th>Shape</th>
<th>Texture</th>
<th>Weight</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>Ellipse</td>
<td>Hard</td>
<td>Red</td>
</tr>
<tr>
<td>Orange</td>
<td>Round</td>
<td>Soft</td>
<td>Light</td>
</tr>
</tbody>
</table>

Qian-Sejnowski Architecture

\[ i_{ja} \leftarrow 1(s_{ja} = a) \]
\[ h_i \leftarrow (a_i + \sum_{j} i_{ja} w_{ji,j}) \]
\[ o_i \leftarrow h_i + \sum_j h_j a_{i,j} \]
\[ s \leftarrow \arg\max_j o_j \]

Neural Network Prediction

- A neural network defines a function from inputs to outputs
- Inputs can be discrete or continuous valued
- In this case, the network defines a function from a window of size \(2w+1\) around a residue to a secondary structure label for it
- Structure element determined by \(\max(o_\alpha, o_\beta, o_o)\)

Training Neural Networks

- By modifying the network weights, we change the function
- Training is performed by
  - Defining an error score for training pairs \(\langle\text{input}, \text{output}\rangle\)
  - Performing gradient-descent minimization of the error score
  - Back-propagation algorithm allows to compute the gradient efficiently
- We have to be careful not to overfit training data

Smoothing Outputs

- Some sequences of secondary structure are impossible: \(\alpha\alpha\beta\alpha\alpha\beta\alpha\alpha\)
- To smooth the output of the network, another layer is applied on top of the three output units for each residue

Success rate: about 65% on unseen proteins
Breaking the 70% Threshold

- An innovation that made a crucial difference uses evolutionary information to improve prediction

Key idea:
- Structure is preserved more than sequence
- Surviving mutations are not random
- Exploit evolutionary information, based on conservation analysis of multiple sequence alignments.

Nearest Neighbor Approach

- Predict the secondary structure state, based on the secondary structure of homologous segments from proteins with known 3d structure.
- A key element: the choice of scoring table for evaluation of segment similarity.
- Use \( \max(n_a, n_b, n_c) \)

PHD Approach

- Perform BLAST search to find local alignments
- Remove alignments that are "too close"
- Perform multiple alignments of sequences
- Construct a profile (PSSM) of amino-acid frequencies at each residue
- Use this profile as input to the neural network
- A second network performs "smoothing"
- The third level computes jury decision of several different instantiations of the first two levels.

Psi-pred : same idea

1. Run PSI-Blast \( \rightarrow \) output sequence profile
2. 15-residue sliding window = 315 values, multiplied by hidden weights in 1st neural net.
   Output is 3 values (a weight for each state H, E or L) per position.
3. 60 input values, multiplied by weights in 2nd neural network, summed. Output is final 3-state prediction.

Other Classification Methods

- Neural Networks were used as a classifier in the described methods.
- We can apply the same idea, with other classifiers, e.g.: SVM
  - Advantages: Effectively avoid over-fitting
  - Supplies prediction confidence

Secondary Structure Prediction - Summary

1st Generation - 1970s
- Chou & Fasman, \( Q_3 = 50-55\% \)

2nd Generation - 1980s
- Qian & Sejnowski, \( Q_3 = 60-65\% \)

3rd Generation - 1990s
- PHD, PSI-PRED, \( Q_3 = 70-80\% \)

Failures:
- Long term effects: S-S bonds, parallel strands
- Chemical patterns
- Wrong prediction at the ends of H/E

[The PredictProtein server]

[S. Hua and Z. Sun, (2001)]