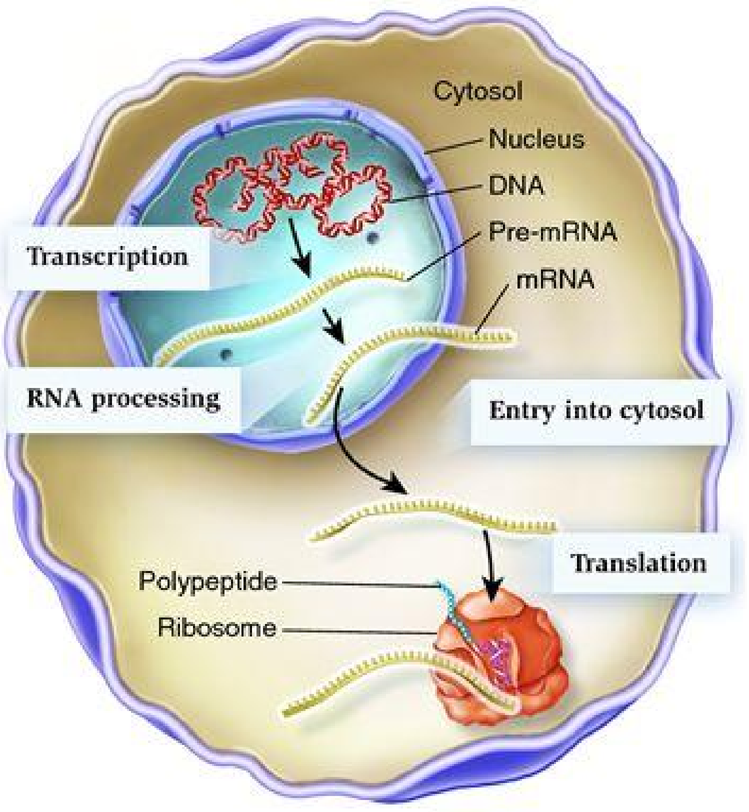
Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields

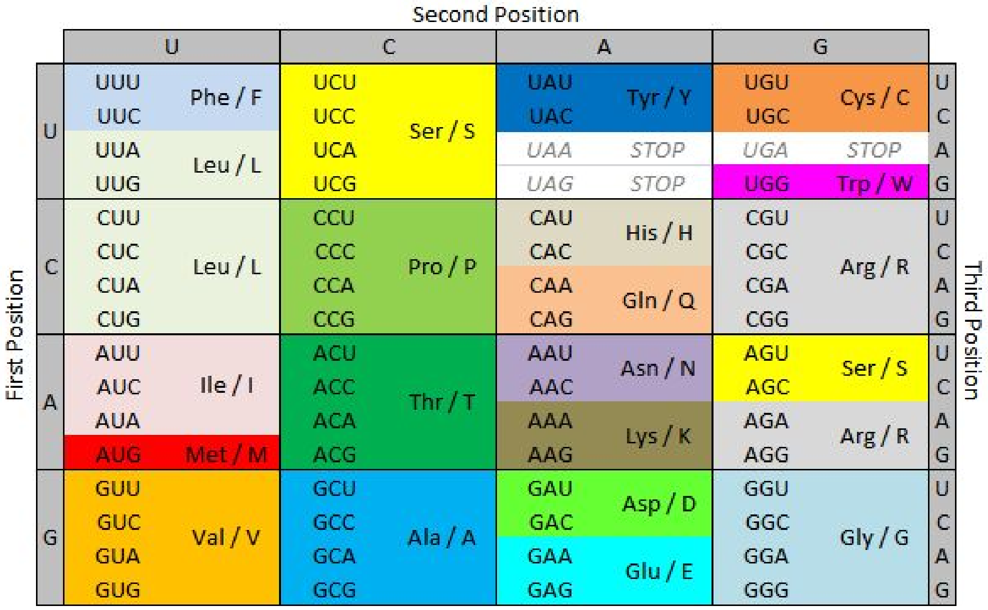
Wang, S., Peng, J., Ma, J., & Xu, J. (2016). Scientific Reports, 6, 18962.
https://doi.org/10.1038/srep18962

Presented by: Shaimaa Bakr, Cici Chen, Daniel Fernandes & Rahul Palamuttam

Central Dogma of Biology



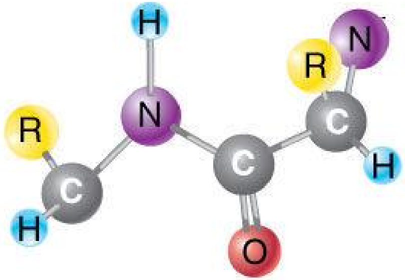
eukaryotes



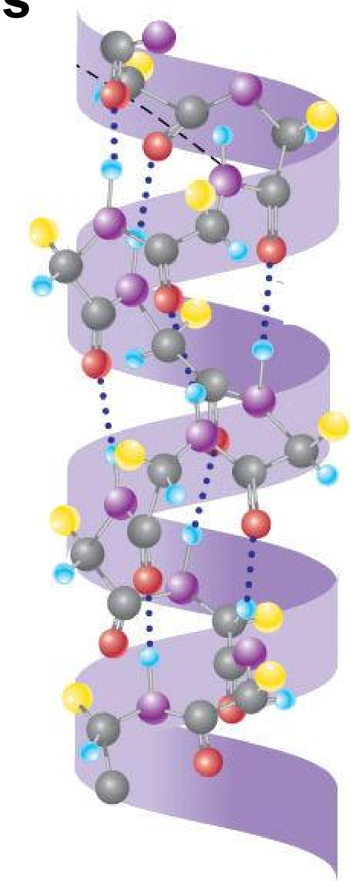
	Second Position								
First Position	U		C		A		G		Third Position
U	UUU	Phe / F	UCU	Ser / S	UAU	Tyr / Y	UGU	Cys / C	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu / L	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG	STOP	UGG	Trp / W	G
C	CUU	Leu / L	CCU	Pro / P	CAU	His / H	CGU	Arg / R	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln / Q	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile / I	ACU	Thr / T	AAU	Asn / N	AGU	Ser / S	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys / K	AGA	Arg / R	A
	AUG	Met / M	ACG		AAG		AGG		G
G	GUU	Val / V	GCU	Ala / A	GAU	Asp / D	GGU	Gly / G	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu / E	GGA		A
	GUG		GCG		GAG		GGG		G

Anticodon Amino Acid Table

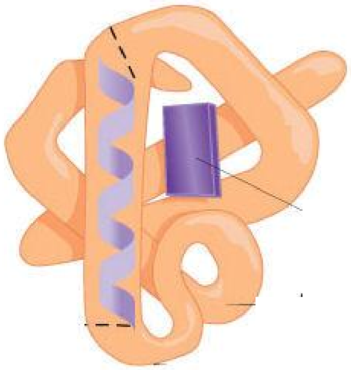
Levels of Protein Structures



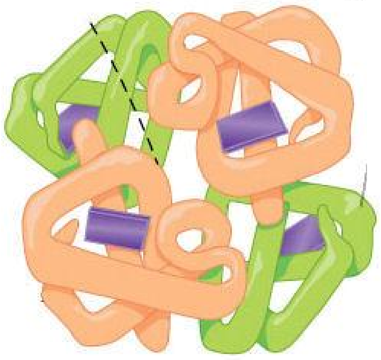
Primary structure



Secondary structure

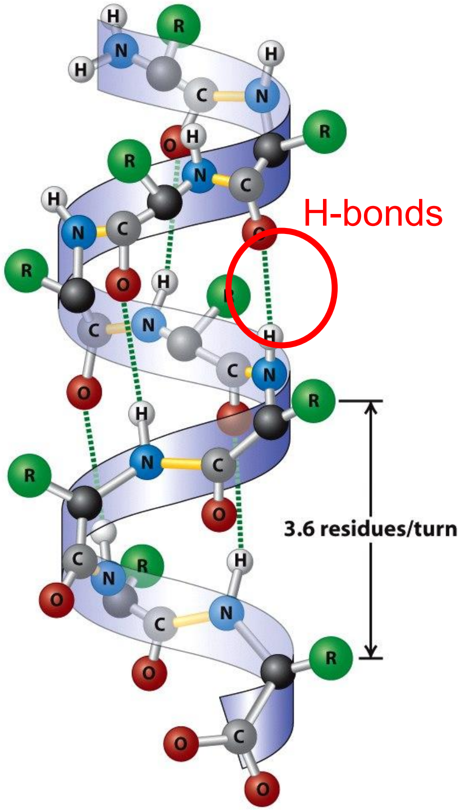


Tertiary structure



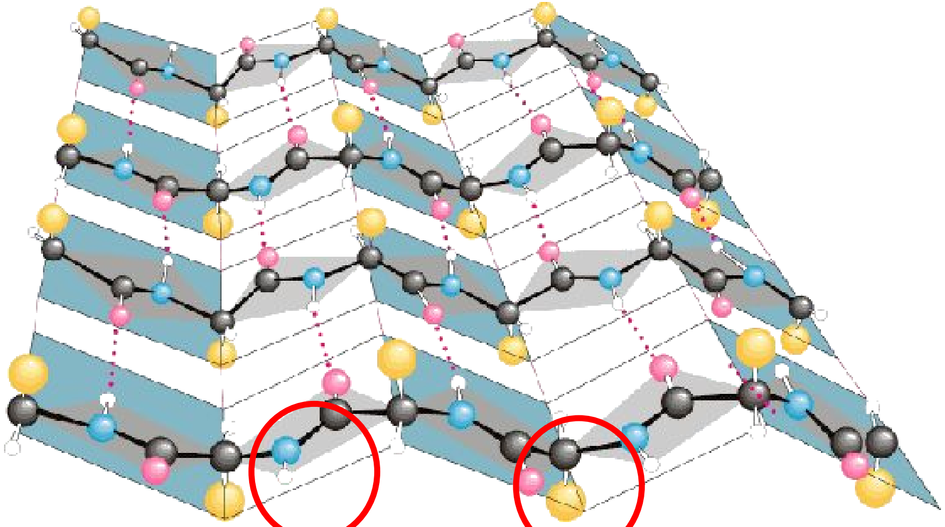
Quaternary structure

Secondary Structure Types



H-bonds

3.6 residues/turn



Straight β-strand

hairpin motif

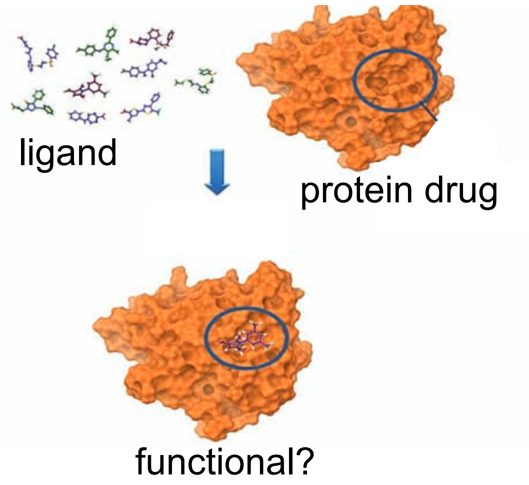
α-helix

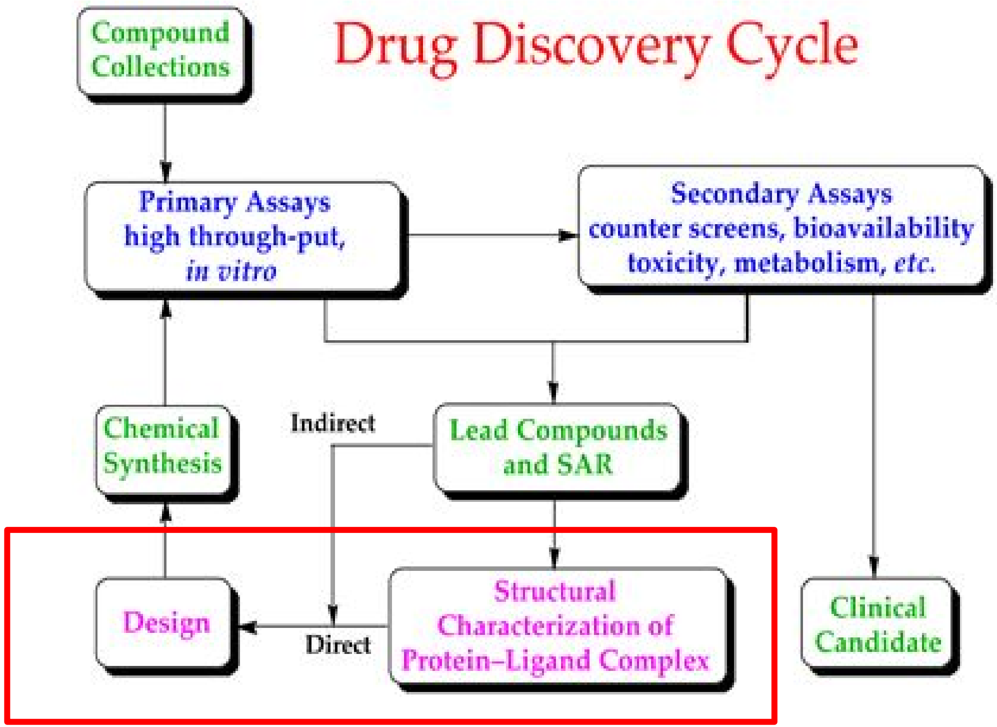
β-sheet /strand

loop /turn

Why predict secondary structure?

- "Bottom up" approach to predict tertiary structure
- Structure informs function

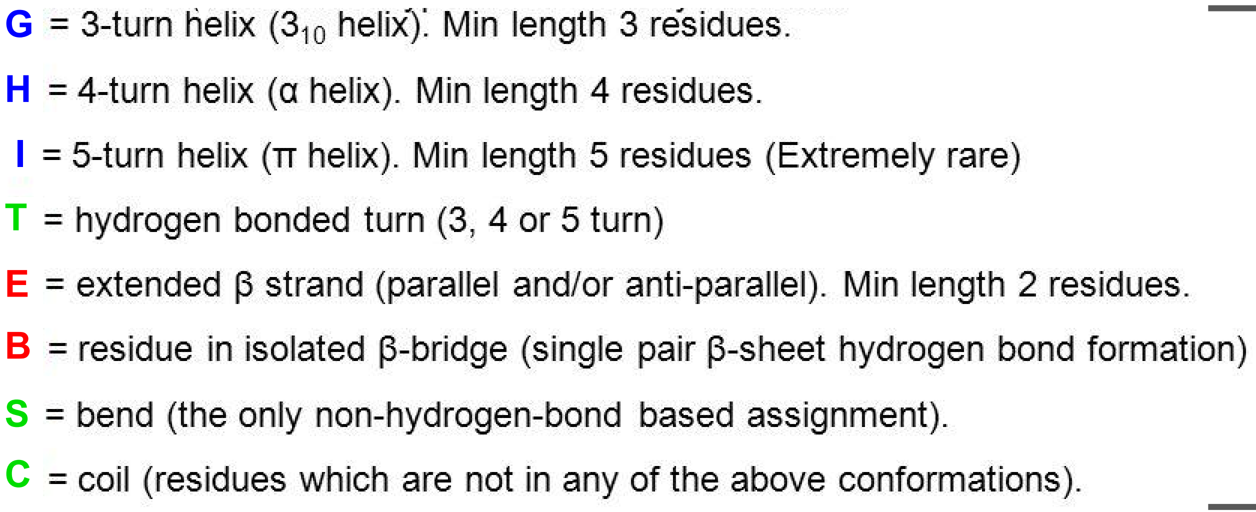



Drug Discovery Cycle
Compound Collections
Primary Assays high through-put, *in vitro*
Secondary Assays counter screens, bioavailability toxicity, metabolism, *etc.*
Chemical Synthesis
Indirect
Lead Compounds and SAR
Design
Direct
Structural Characterization of Protein–Ligand Complex
Clinical Candidate

Labeling Secondary Structures

Alpha helix	Beta sheet	Anything else (loop/coil)
~30%	~20%	~50%

3-state model



G = 3-turn helix (3_{10} helix). Min length 3 residues.

H = 4-turn helix (α helix). Min length 4 residues.

I = 5-turn helix (π helix). Min length 5 residues (Extremely rare)

T = hydrogen bonded turn (3, 4 or 5 turn)

E = extended β strand (parallel and/or anti-parallel). Min length 2 residues.

B = residue in isolated β-bridge (single pair β-sheet hydrogen bond formation)

S = bend (the only non-hydrogen-bond based assignment).

C = coil (residues which are not in any of the above conformations).

8-state model

Problem Statement

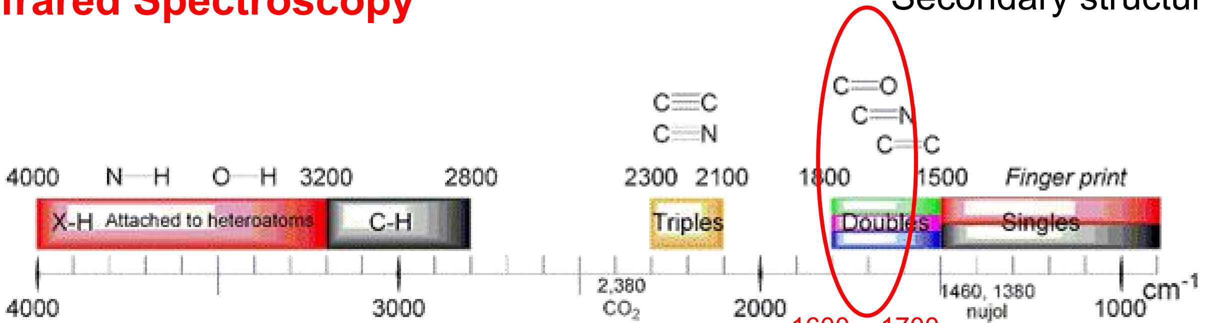
Given a protein with amino acid sequence $r_1\ r_2\ r_3 \ldots r_n$, predict whether each amino acid r_i is in:

(1) an α-helix, a β-strand, or neither. (3-state model)

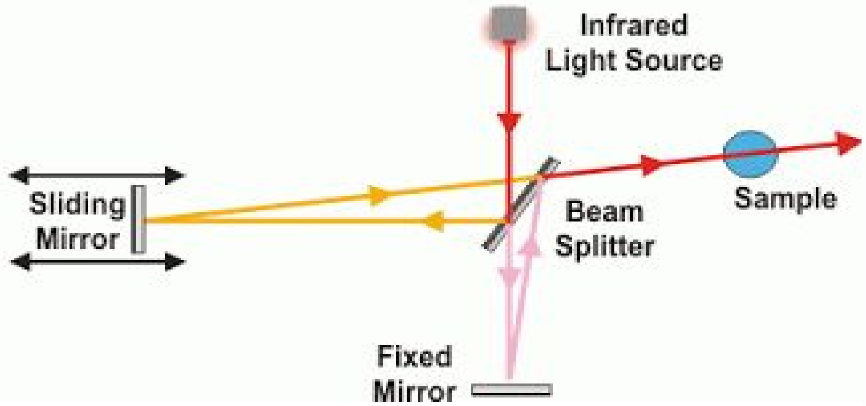
(2) the G, H, I, T, E, B, S, or C state. (8-state model)

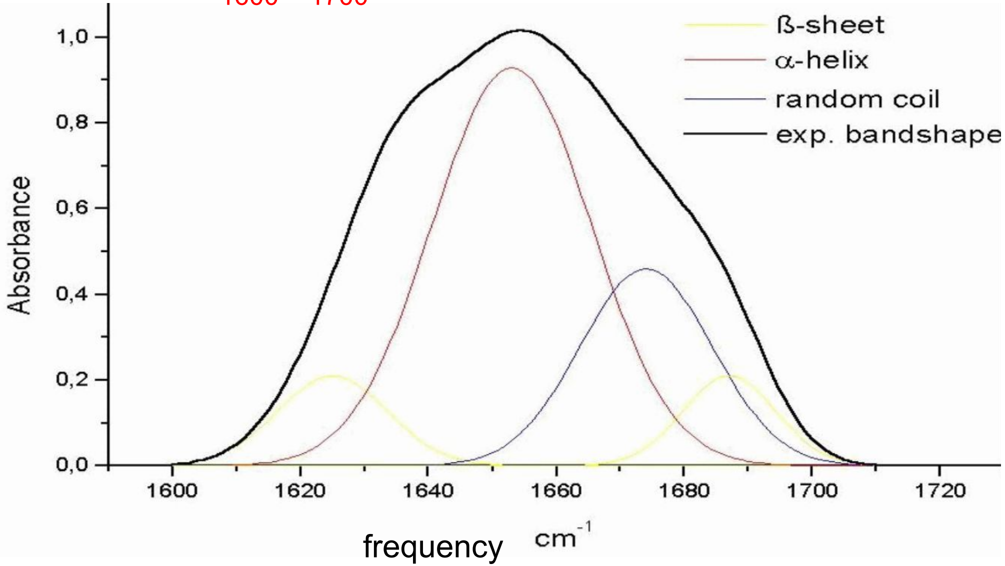
Data from Infrared Spectroscopy

Secondary structures



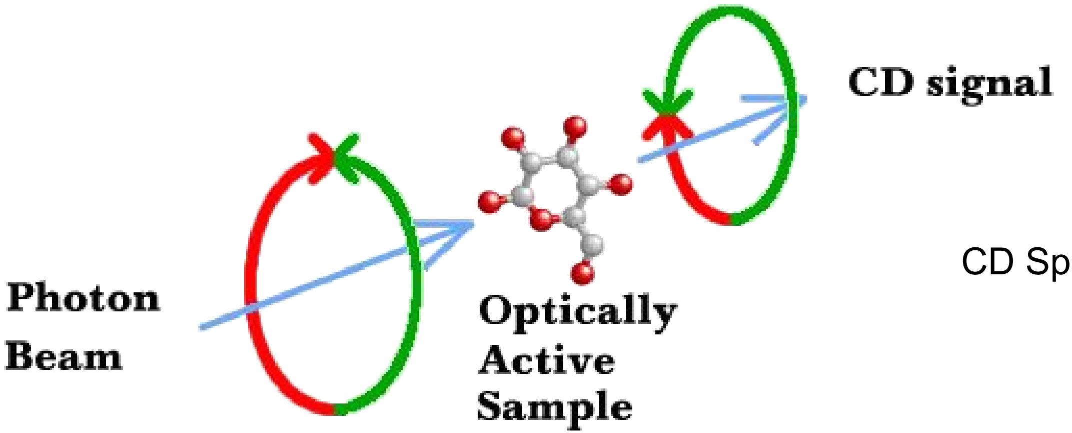
1600 ~ 1700

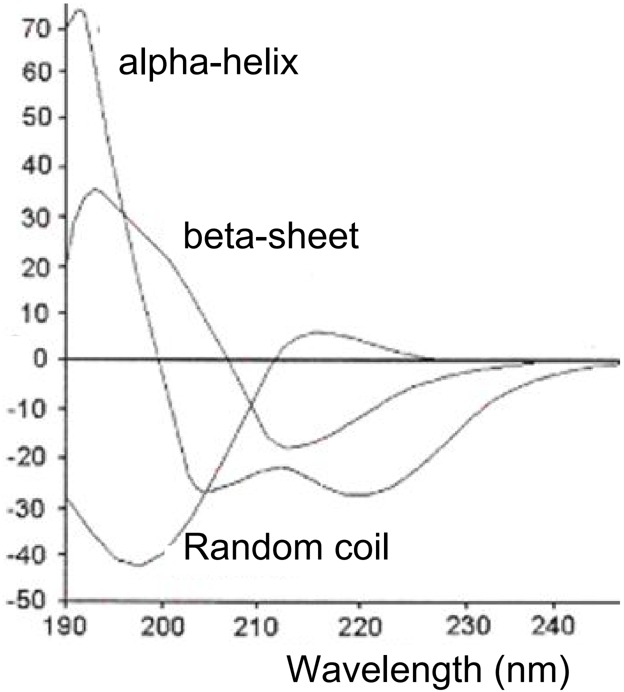




- Obtain light intensity as function of frequency

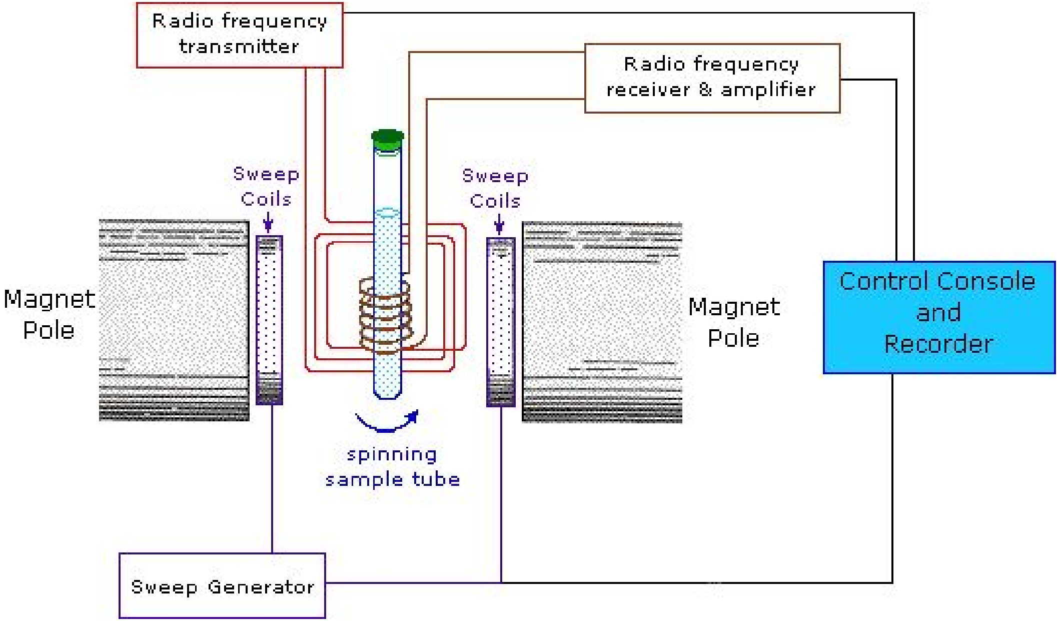
Data from Far-UV Circular Dichroism

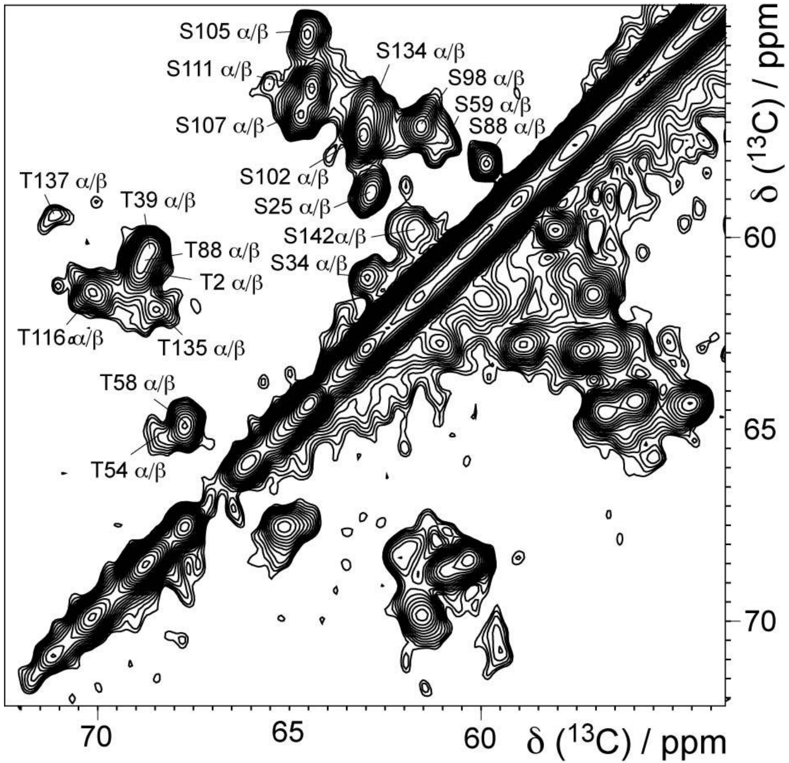




- CD wavelength between 180 and 260 nm for SS
- Obtain discrete voltage values as function of wavelength

Data from Nuclear Magnetic Resonance Spectroscopy (for smaller proteins)

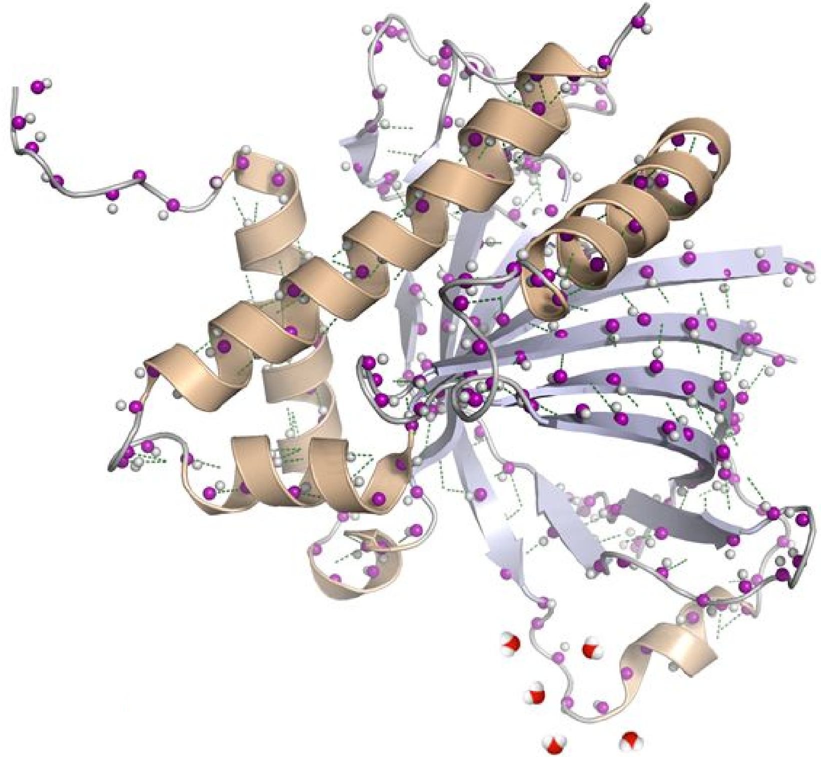

Radio frequency transmitter
Radio frequency receiver & amplifier
Sweep Coils
Sweep Coils
Magnet Pole
Magnet Pole
Control Console and Recorder
spinning sample tube
Sweep Generator



- Free induction decay of specific nuclei as function of radiofrequency excitation pulses

Other considerations about data

- pH & temperature of solution can influence SS
- different organisms can make different forms of the same protein



e.g. H_2O surrounds & interacts with secondary structure

Important Definitions

Residue: a monomer within a polymeric chain (e.g. 1 amino acid in a protein)

Protein superfamily: group of proteins classified according to specific classification schemes

Sequence identity: amount of amino acids that match exactly between 2 sequences

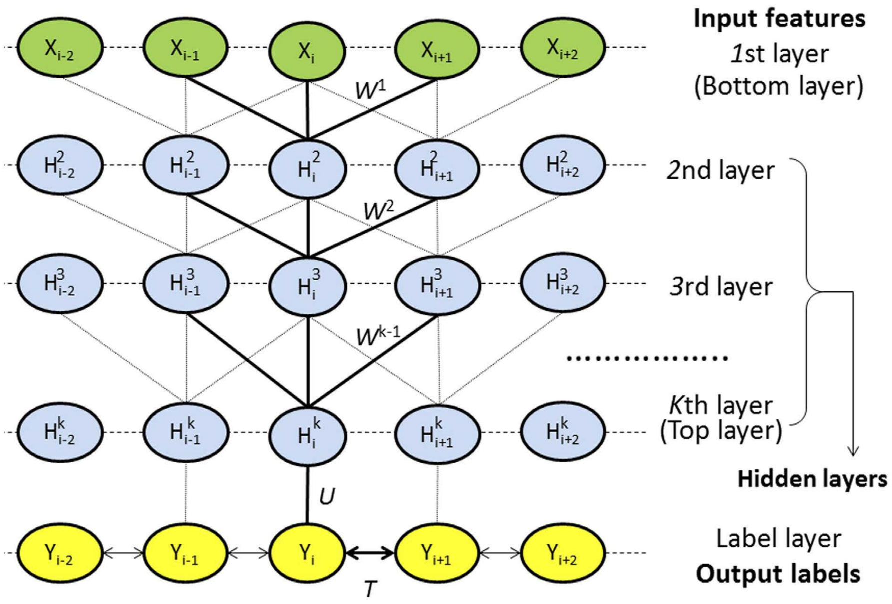
Neff score: measures the average number of effective amino acids across all the residues, ranging from 1 to 20

Similarity: extent to which amino acids are conserved between proteins

Protein homology: inferred from sequence identity, characterizes extent of shared ancestry between proteins

Architecture

- A hybrid of Deep Convolutional Neural Nets and Conditional Neural Fields
- Conditional Neural Fields (CNF) are an extension of Conditional Random Fields (CRFs)



A brief history of...

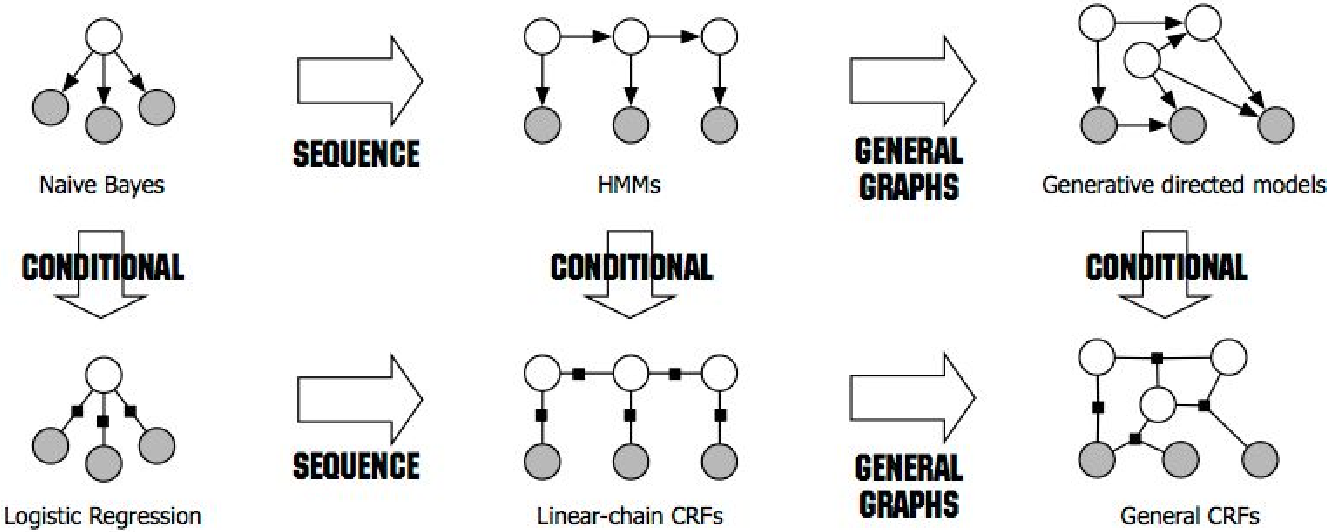


Figure 1.2 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

- Top level: Generative Models
- Bottom level: Discriminative Models

Generative vs Discriminative Models

- Hidden Markov Models are **Generative**
  - Using hidden states, they *generate* a likely observed output
  - Model the joint distribution: P(x,y) = P(y)*P(x|y)
  - Use maximum a posteriori (MAP) classifier to determine which hidden state was most probable
- Conditional Random Fields are **Discriminative**
  - CRFs *describe* a sequence by coloring the sequence with a fixed set of labels
    - (For this paper, our labels will be protein SS)
  - CRFs directly model the data using conditional probabilities
    - P(y|x)
- Use Bayes Rule to convert from one to the other!
  - P(y|x) = P(x|y)*P(y)/P(x)
- In practice, non-naive generative models/statespaces are hard to create

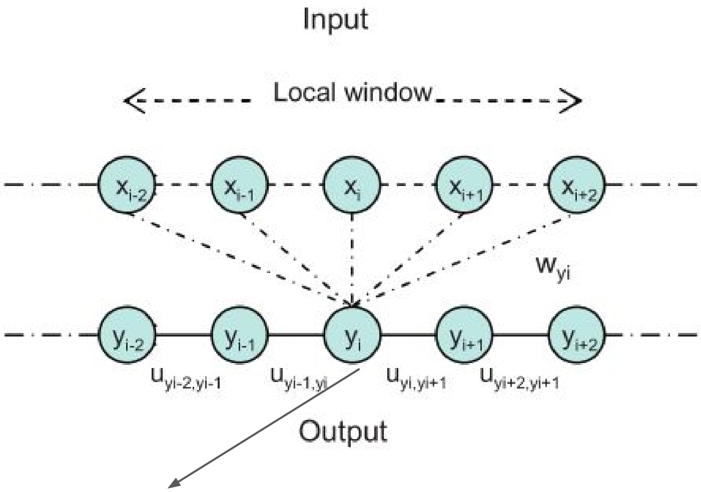
Takeaways

Generative Models	**Discriminative Models**
● Can be applied in unsupervised learning	● Unsupervised learning still an “active area of research”
● Forced to model the input distribution and the conditional probabilities at the same time	● Can model the input distribution and the conditional probabilities separately
● Less prone to overfitting data	● More prone to overfitting data

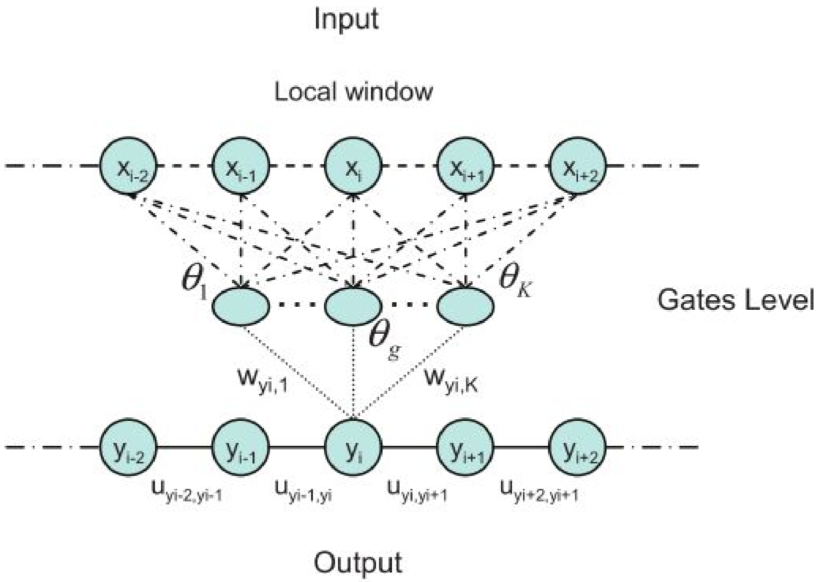
Read for more details:

Sutton, C., & Mccallum, A. (2011). An Introduction to Conditional Random Fields. Machine Learning, 4(4), 267–373. https://doi.org/10.1561/2200000013

(Linear Chain)Conditional Random Fields & Neural Fields



P(output at y | x sequence) =
normed exponential(linear weighted sum of local window + weight of previous output)



P(output at y | x sequence) =
normed exponential(nonlinear weighted sum of local window + weight of previous output)

Conditional Random Fields

Inputs:

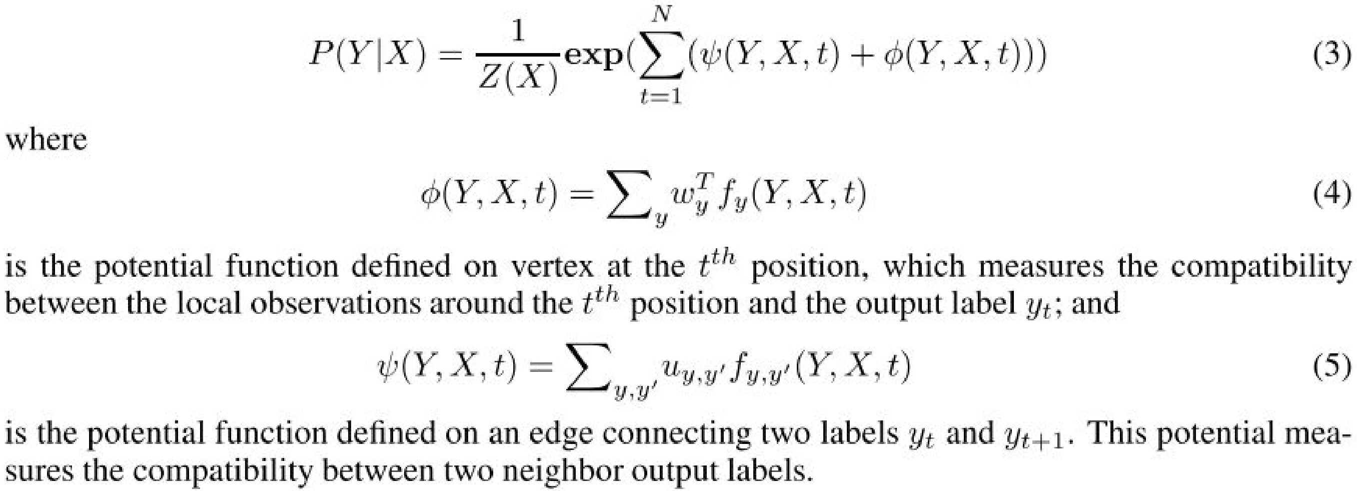
1. The dependency between the neighboring output labels. Essentially a list of transitions.

   Formally: $f_{y,y'}(Y,X,t) = \delta[y_t = y]\delta[y_{t-1} = y']$

   Where δ is an indicator function (only 1 when state at position t is y)
2. The dependency between the label at one position and the observations around this position. Essentially a window on our X inputs.

   Formally: $f_y(Y,X,t) = \mathbf{f}(X,t)\delta[y_t = y]$

Conditional Random Fields



$$P(Y|X) = \frac{1}{Z(X)}\mathbf{exp}(\sum_{t=1}^{N}(\psi(Y,X,t) + \phi(Y,X,t))) \tag{3}$$

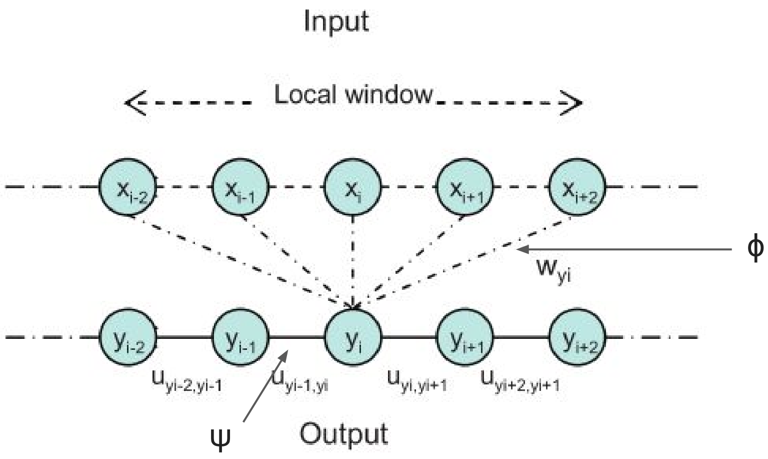
where

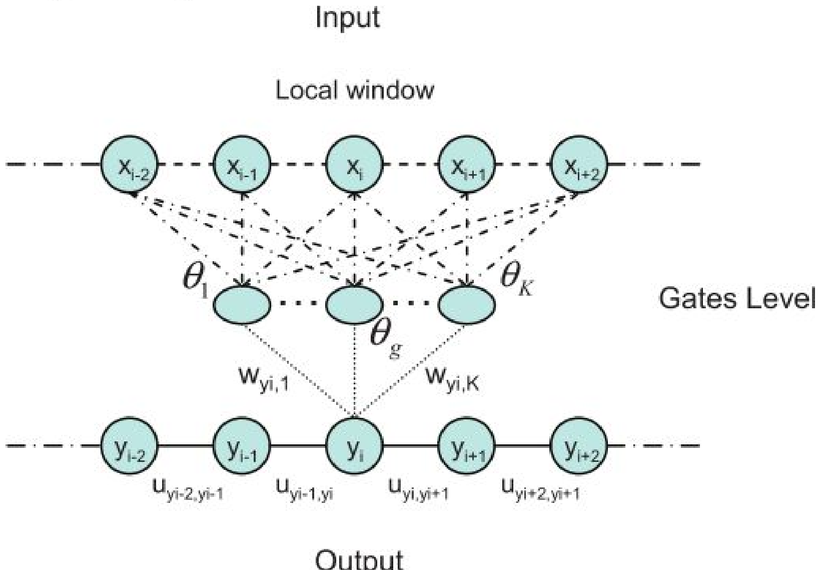
$$\phi(Y,X,t) = \sum\nolimits_{y} w_y^T f_y(Y,X,t) \tag{4}$$

is the potential function defined on vertex at the t^{th} position, which measures the compatibility between the local observations around the t^{th} position and the output label y_t; and

$$\psi(Y,X,t) = \sum\nolimits_{y,y'} u_{y,y'} f_{y,y'}(Y,X,t) \tag{5}$$

is the potential function defined on an edge connecting two labels y_t and y_{t+1}. This potential measures the compatibility between two neighbor output labels.



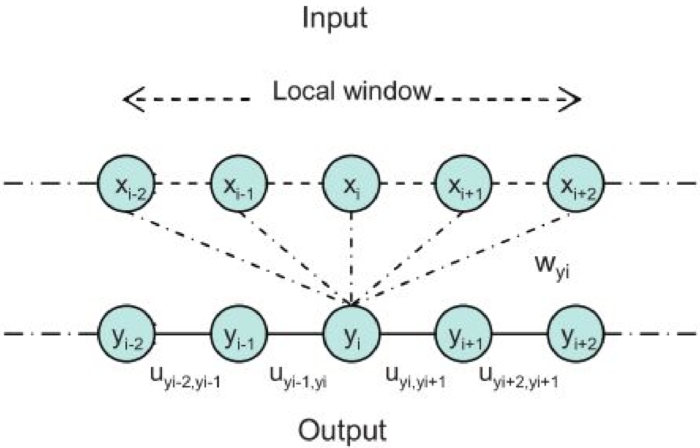


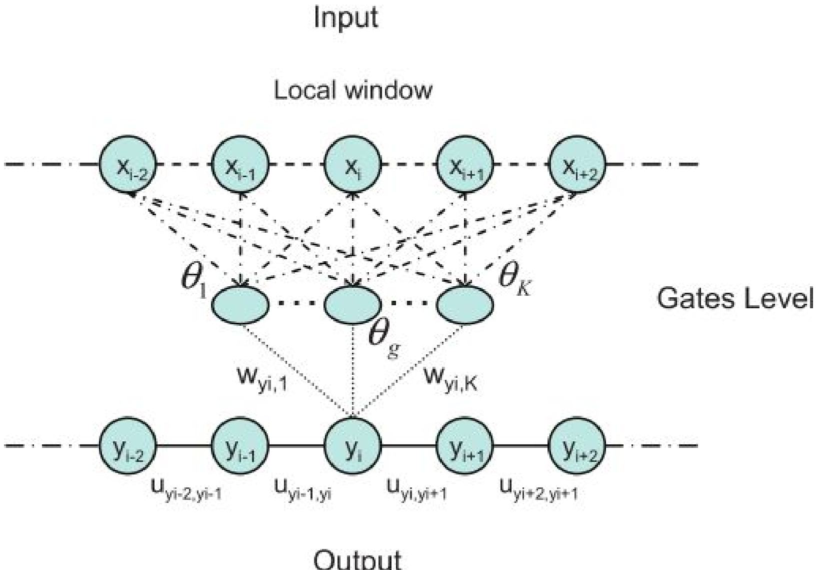
Conditional Neural Fields

$$\phi(Y, X, t) = \sum_y \sum_{g=1}^{K} w_{y,g} h(\theta_g^T \mathbf{f}(X, t)) \delta[y_t = y]$$

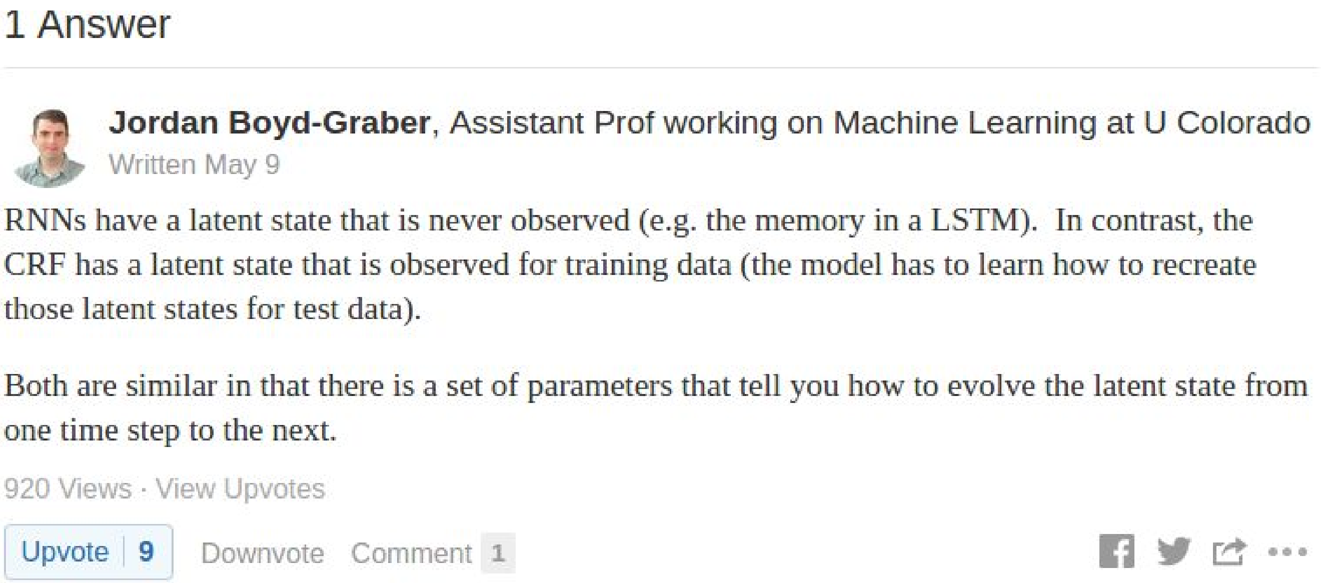
Where h is a nonlinear activation function, like tanh or sigmoid

Lots of work on this, see:
Chen, L.-C., Schwing, A. G., Yuille, A. L., & Urtasun, R. (n.d.). Learning Deep Structured Models.





Difference Between CRF and RNN?



1 Answer

Jordan Boyd-Graber, Assistant Prof working on Machine Learning at U Colorado
Written May 9

RNNs have a latent state that is never observed (e.g. the memory in a LSTM). In contrast, the CRF has a latent state that is observed for training data (the model has to learn how to recreate those latent states for test data).

Both are similar in that there is a set of parameters that tell you how to evolve the latent state from one time step to the next.

920 Views · View Upvotes

1. Regularization
2. Select non-redundant protein sequences?
    a. Protein Sequence Identity
    b. Protein Superfamilies

GOAL : Avoid overfitting

Training Method

$$logP(Y|X) = \sum[\Psi(Y, X, i) + \Phi(Y, X, i)] - logZ(x)))$$

- This is obtained by taking the log of both sides of the CRF equation for conditional probability
- Train model parameters by maximum-likelihood
- Y = Secondary Structure type at residue i
- X = input feature where X_i is a column vector representing the input feature
- Z = partition function

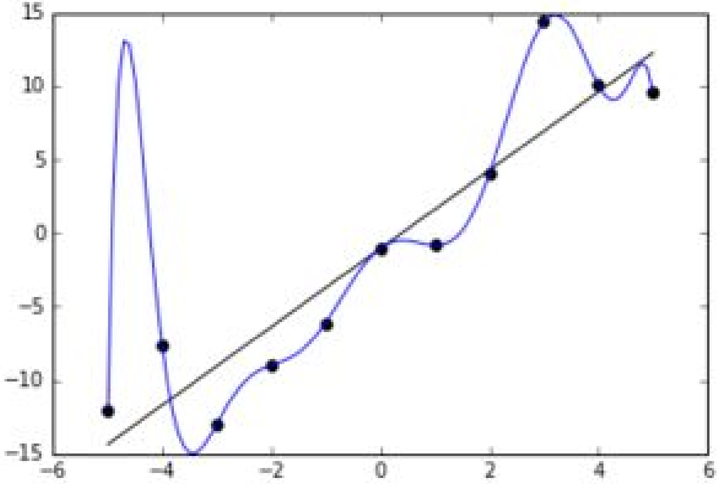
Regularization: The problem of overfitting

- Overfit

$$min \sum V(f(x), y)))$$

- Regularization

$$min \sum V(f(x), y)) + \lambda R(f))$$



L2 Regularization

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

- L2 norm least squares error
- Objective Function :

$$\text{max}\theta \ \log P\theta \ (\boldsymbol{Y}|\boldsymbol{X}) - \lambda\|\theta\| \ 2$$

- To reduce over-fitting, the log-likelihood objective function is penalized with the L2-norm of the model parameters.

Large Regularization Factor

- DeepCNF has many model parameters
- Small L2-norm will restrict the search space of the model parameter
- To prevent overfitting the regularization factor must also be sufficiently large
- Too large of a Regularization factor => underfitting

L-BFGS

- Once we introduce non-linearities
  - CRF’s were convex, had guaranteed global maximas
  - Traditional gradient descent will not work
  - So we need stochastic gradient descent
  - L-BFGS similar to stochastic gradient descent
- Limited BFGS
  - Optimization algorithm that approximates Browden-Fletcher-Goldfarb-Shannon algorithm with limited memory
- Use L-BFGS to search for optimal model parameters
  - Parameter estimation
- Has been successfully used to train CRF and CNF

Training and Test : 25% Sequence Identity

Training Set

- ~5600 CullPDB Proteins
- JPRED 1338 training
  - Use non-redundant proteins
  - Use proteins in different superfamilies
  - reduce bias incurred by sequence profile similarity between training and test proteins

Test Set

- ~ 500 CullPDB Proteins
- 513 CB513 Proteins
- 123 CASP10 Proteins
- 105 CASP11 Proteins
- 179/403 CAMEO test targets

What is Protein Sequence Identity

What is sequence identity?

- Sentences,
  - Similar sequences have common phonemes, letters, and capitalization

- Protein Sequences
  - Similar chemical properties i.e. acidic vs basic, hydrophobic vs hydrophilic

Training and Test : Protein Sequence Identity

PDB - Protein Data Bank

PISCES - Protein Culling Server

- Creates PDB sequence identities via Hidden Markov Models

CullPDB - 25 % Sequence identity to remove redundancies between training and test set

-

RCSB PDB
PROTEIN DATA BANK
PISCES

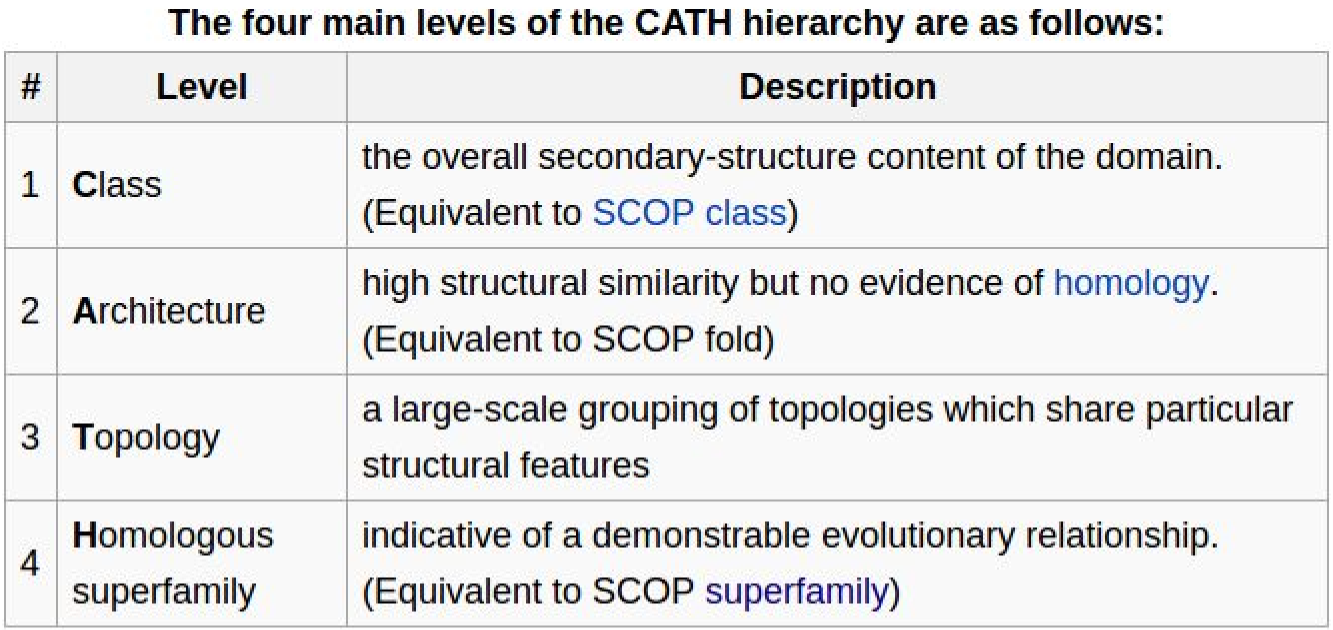
Training and Test : Protein superfamilies

Protein Superfamilies

CATH - Class, Architecture, Topology, Homology

- A hierarchical protein domain classification
- Homologous superfamilies in CATH predict protein function by recognizing sequence patterns associated with a particular function
- Insight : Proteins from different superfamilies have different sequence identities

CATH



The four main levels of the CATH hierarchy are as follows:

#	Level	Description
1	**C**lass	the overall secondary-structure content of the domain. (Equivalent to SCOP class)
2	**A**rchitecture	high structural similarity but no evidence of homology. (Equivalent to SCOP fold)
3	**T**opology	a large-scale grouping of topologies which share particular structural features
4	**H**omologous superfamily	indicative of a demonstrable evolutionary relationship. (Equivalent to SCOP superfamily)

Overfitting Conclusion

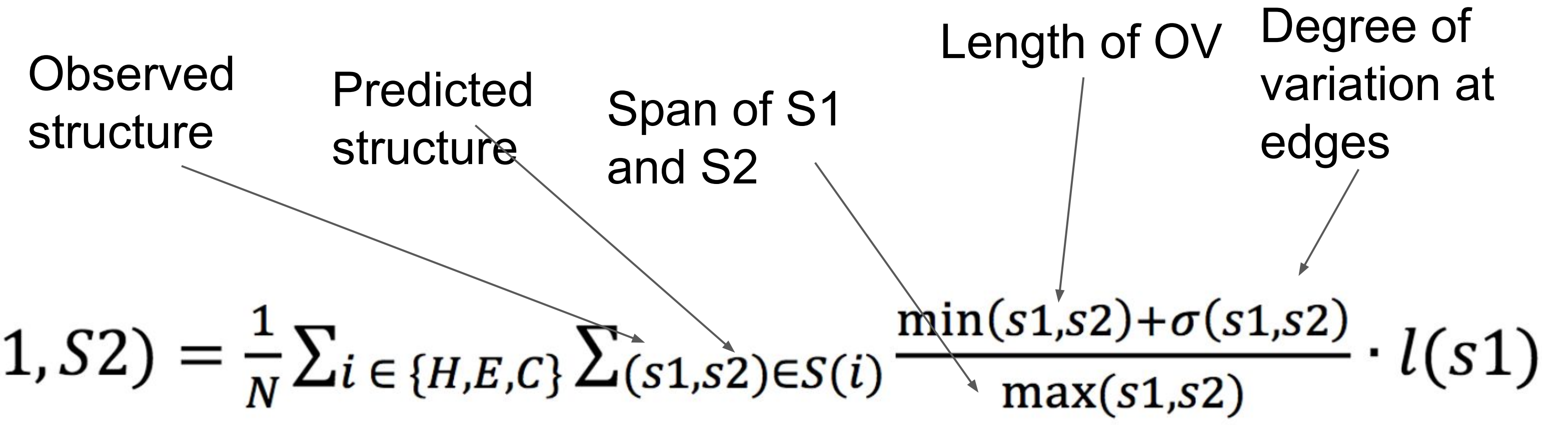
1. They calibrated a regularization factor
2. They artificially introduced more diverse protein sequences
    a. Protein sequence similarity threshold
    b. Unique Protein Families

Results

- **8-class SS prediction**
  - SSpro34, without template and with template, RaptorX-SS833, , ICML201436
- **3-class SS prediction**
  - SSpro, RaptorX-SS8, PSIPRED24, SPINE-X12, JPRED
- **Performance Metrics**
  - Q3, Q8, Precision and Recall, Segment of OVerlap (SOV)

Segment of OVerlap (SOV)

- Is more suitable for segmented nature of SS
- We care more about the *type* and *general location* of SS
- Less with weight on edge errors



Observed structure

Predicted structure

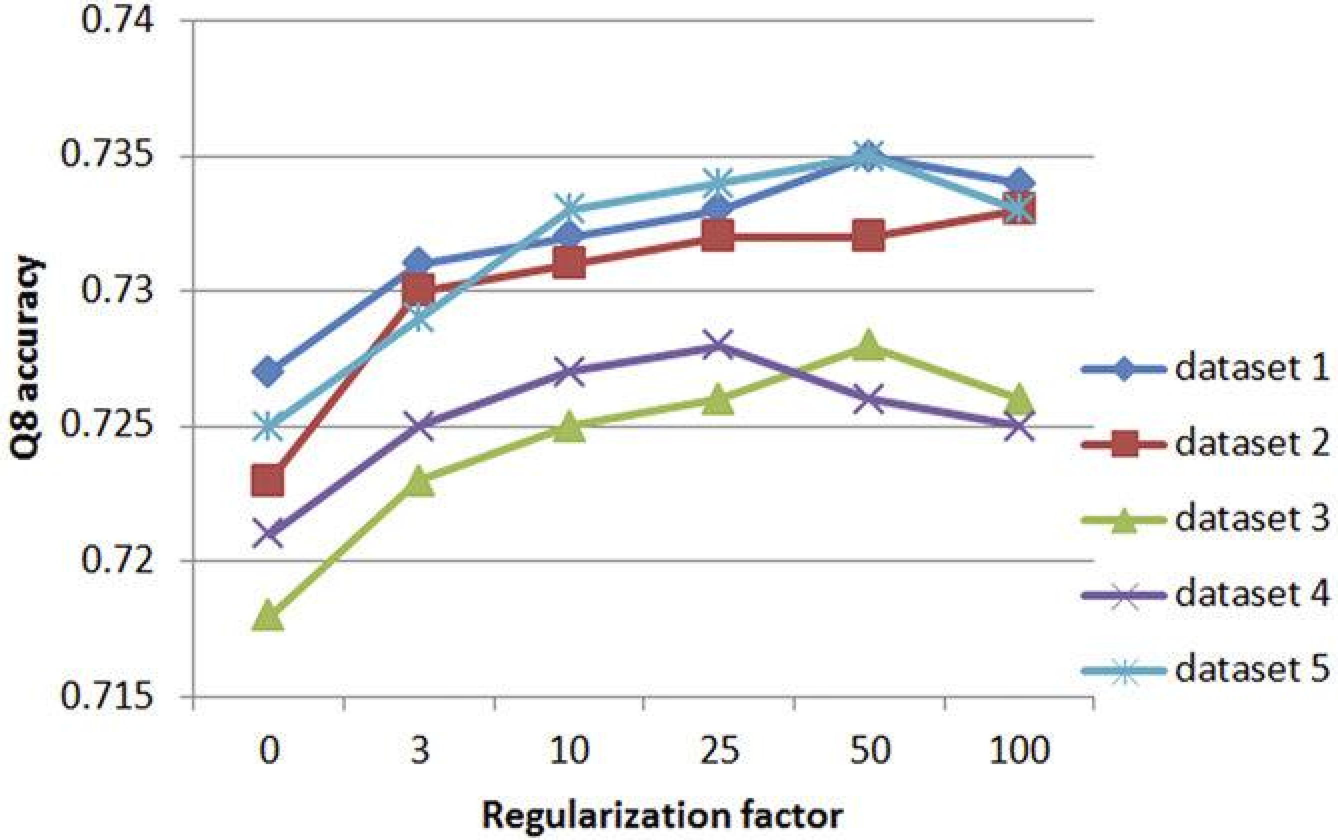
Span of S1 and S2

Length of OV

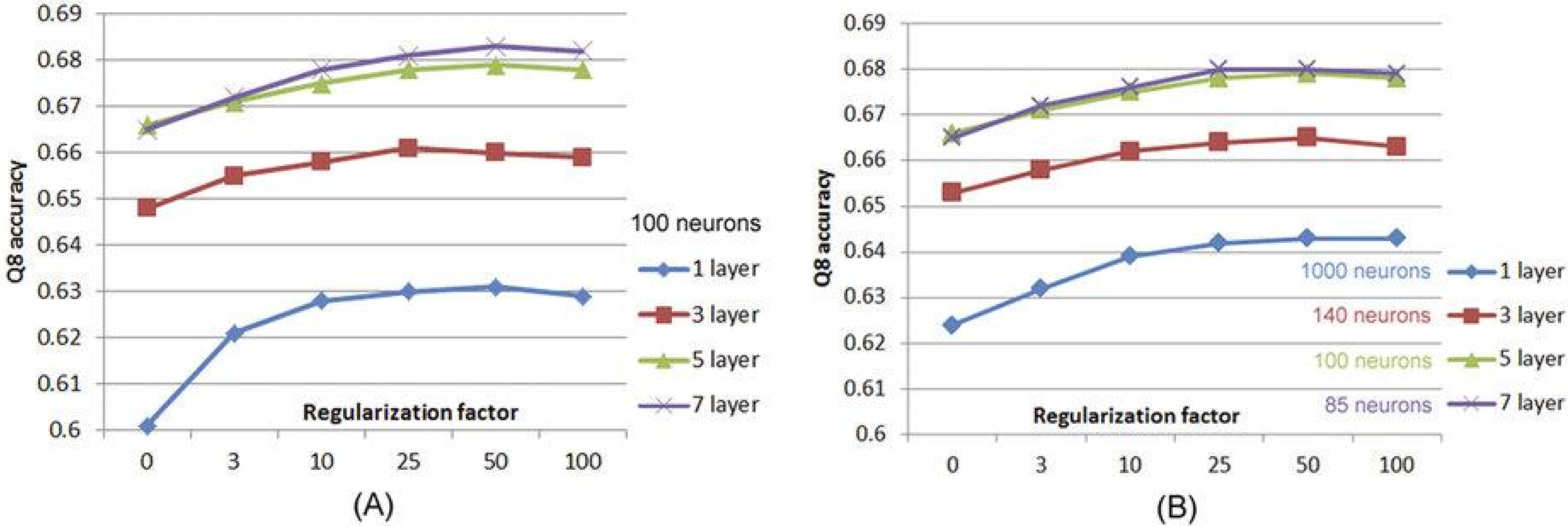
Degree of variation at edges

$$SOV(S1,S2)=\frac{1}{N}\sum_{i\in\{H,E,C\}}\sum_{(s1,s2)\in S(i)}\frac{\min(s1,s2)+\sigma(s1,s2)}{\max(s1,s2)}\cdot l(s1)$$

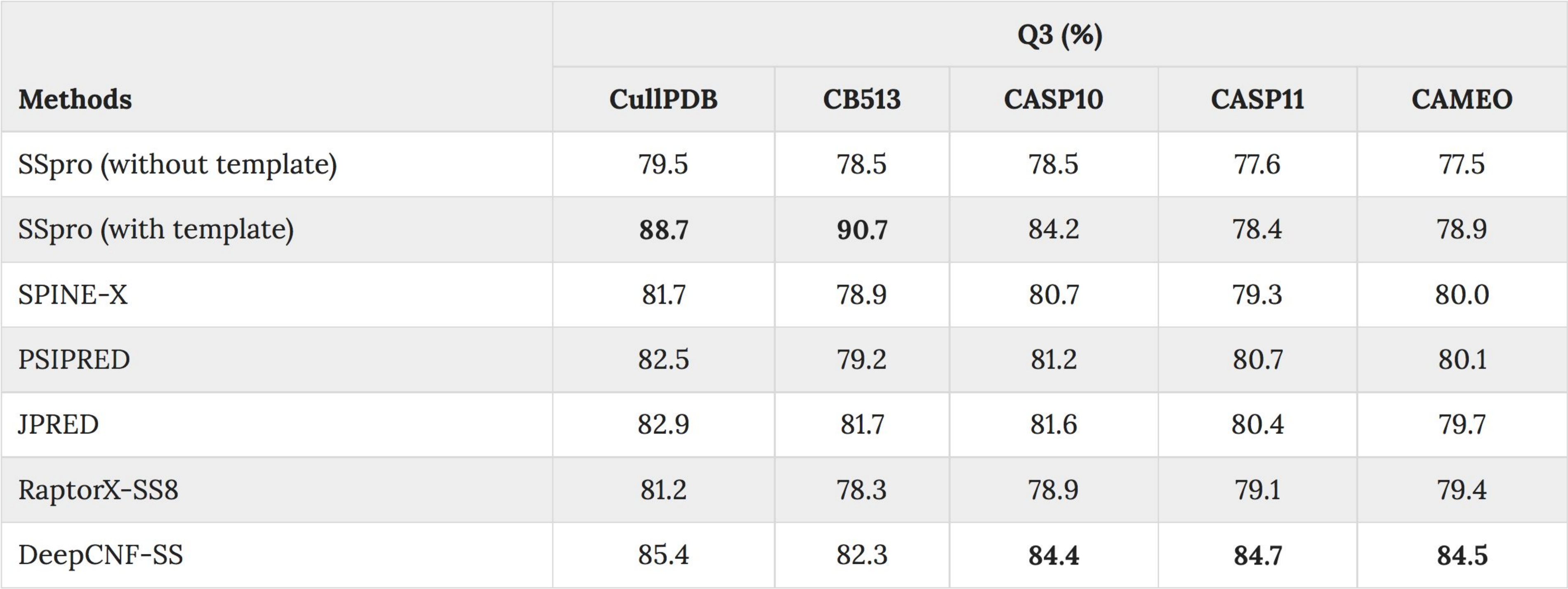
Regularization Factor



CNF Architecture

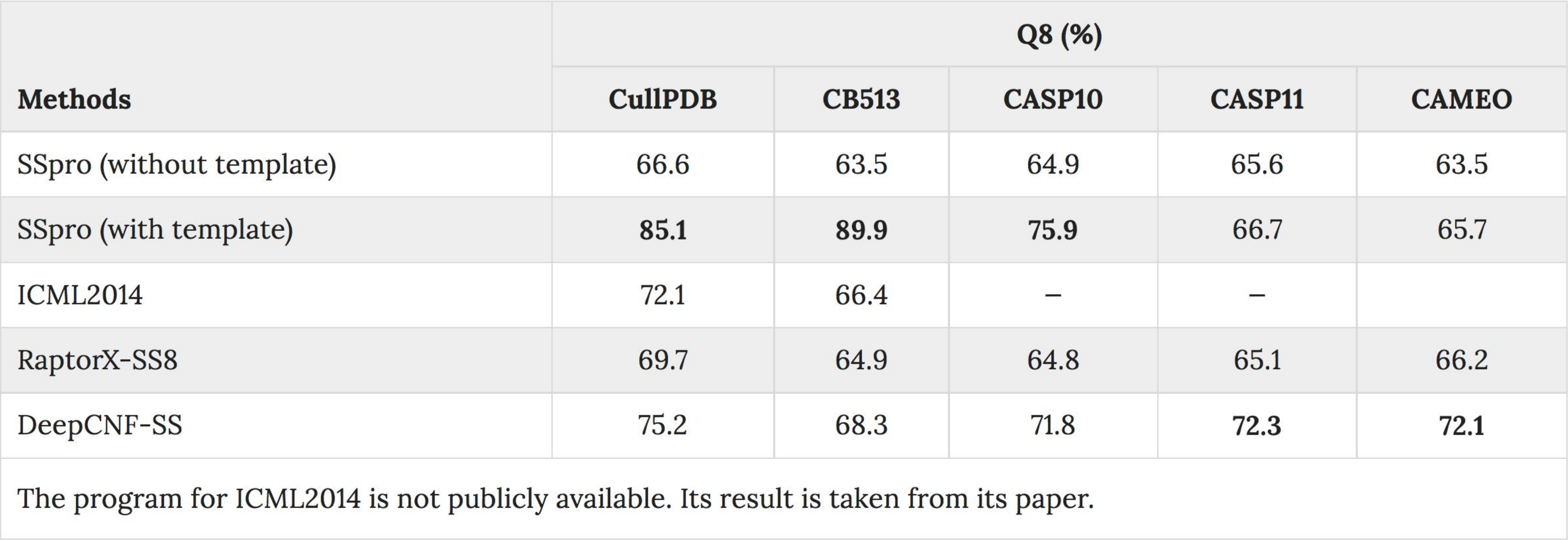


Results



Methods	Q3 (%)				
	CullPDB	**CB513**	**CASP10**	**CASP11**	**CAMEO**
SSpro (without template)	79.5	78.5	78.5	77.6	77.5
SSpro (with template)	**88.7**	**90.7**	84.2	78.4	78.9
SPINE-X	81.7	78.9	80.7	79.3	80.0
PSIPRED	82.5	79.2	81.2	80.7	80.1
JPRED	82.9	81.7	81.6	80.4	79.7
RaptorX-SS8	81.2	78.3	78.9	79.1	79.4
DeepCNF-SS	85.4	82.3	**84.4**	**84.7**	**84.5**

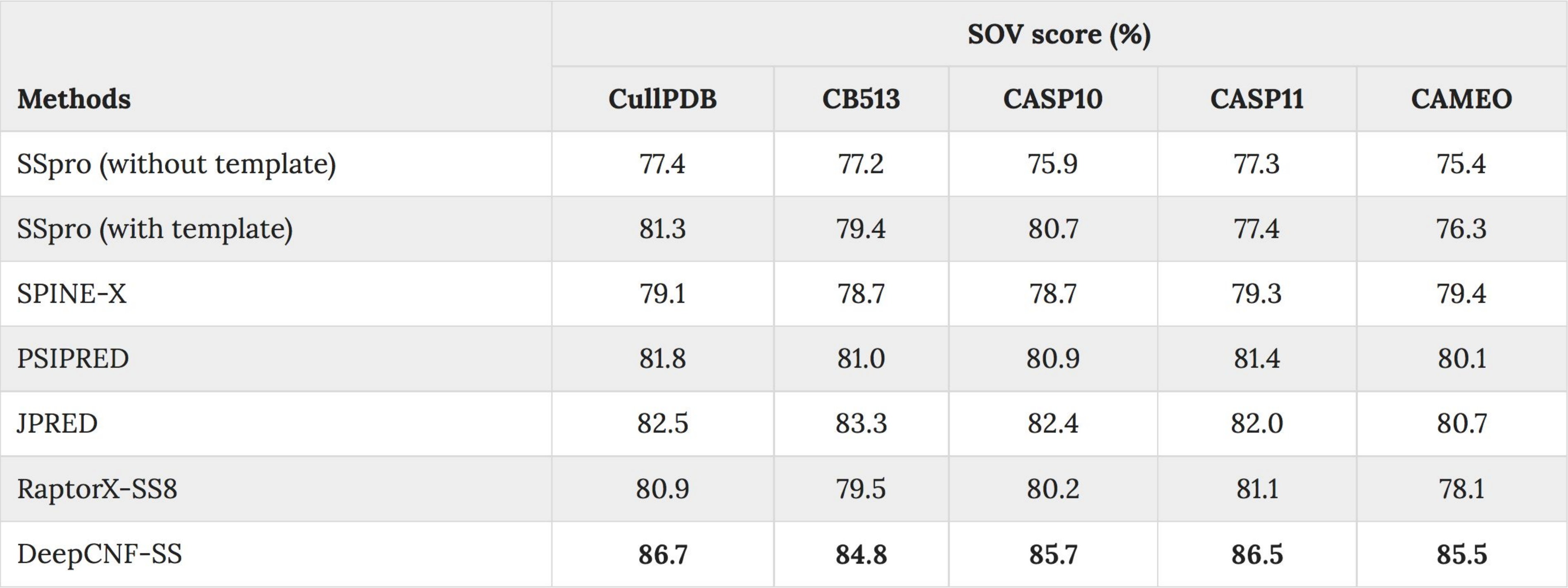
Results



Methods	Q8 (%)				
	CullPDB	CB513	CASP10	CASP11	CAMEO
SSpro (without template)	66.6	63.5	64.9	65.6	63.5
SSpro (with template)	**85.1**	**89.9**	**75.9**	66.7	65.7
ICML2014	72.1	66.4	–	–	
RaptorX-SS8	69.7	64.9	64.8	65.1	66.2
DeepCNF-SS	75.2	68.3	71.8	**72.3**	**72.1**

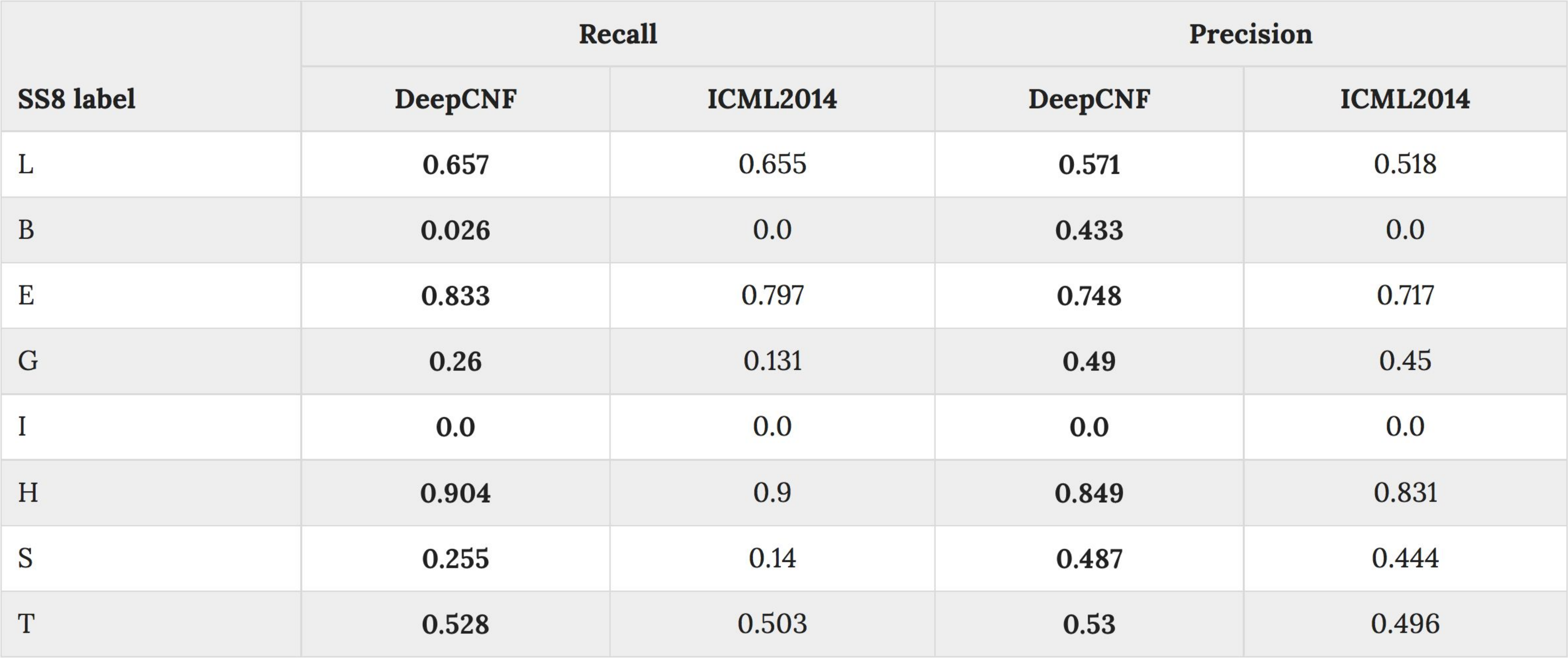
The program for ICML2014 is not publicly available. Its result is taken from its paper.

Results



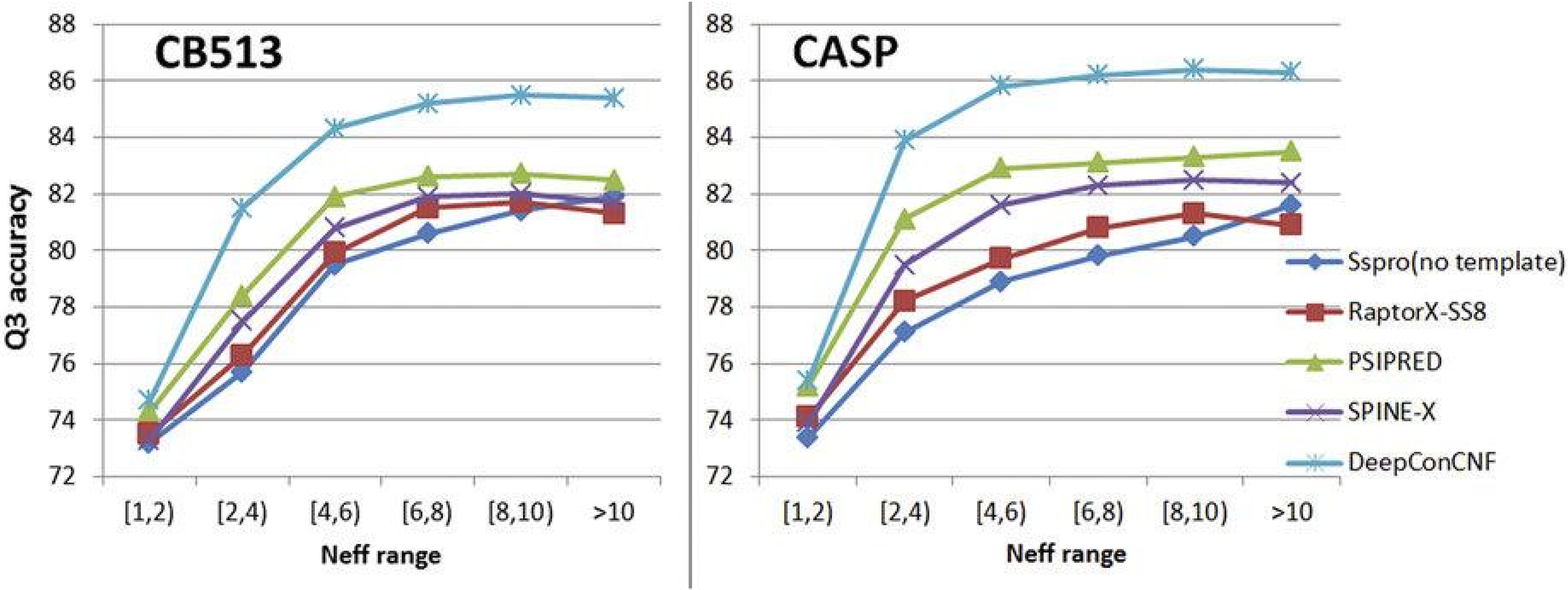
Methods	SOV score (%)				
	CullPDB	**CB513**	**CASP10**	**CASP11**	**CAMEO**
SSpro (without template)	77.4	77.2	75.9	77.3	75.4
SSpro (with template)	81.3	79.4	80.7	77.4	76.3
SPINE-X	79.1	78.7	78.7	79.3	79.4
PSIPRED	81.8	81.0	80.9	81.4	80.1
JPRED	82.5	83.3	82.4	82.0	80.7
RaptorX-SS8	80.9	79.5	80.2	81.1	78.1
DeepCNF-SS	**86.7**	**84.8**	**85.7**	**86.5**	**85.5**

Results - CB513



SS8 label	Recall		Precision	
	DeepCNF	ICML2014	DeepCNF	ICML2014
L	**0.657**	0.655	**0.571**	0.518
B	**0.026**	0.0	**0.433**	0.0
E	**0.833**	0.797	**0.748**	0.717
G	**0.26**	0.131	**0.49**	0.45
I	**0.0**	0.0	**0.0**	0.0
H	**0.904**	0.9	**0.849**	0.831
S	**0.255**	0.14	**0.487**	0.444
T	**0.528**	0.503	**0.53**	0.496

Homologous Information



Where is the improvement from?

Q3 accuracy - 84.9 %

- Stricter experiment with 1338 JPRED proteins for training and 149 for test
  - All proteins belong to different superfamilies
  - Divided training set into 7 and trained 7 DeepCNF models separately
  - Unlikely that test proteins and training proteins share similar sequence profiles

Conclusion : Results are from DeepCNF and not sequence profile similarity!

Thank You

Paper Criticism/Evaluation

- Lack of Methodology

- ICML2014 program not publicly available
  - Only evaluted performance on CASP10, CASP11, and CAMEO test sets -http://jmlr.org/proceedings/papers/v32/zhou14.pdf
- Couldn’t test Cheng’s deep learning method - method not made publicly available
- They do not report SOV for the final experiment for protein superfamilies to filter data sets
- Low precision and recall

Experiment Setup: Comparisons

Q3/Q8 - percent of residues for which predicted secondary structures are correct

8-state SS prediction

- SSPro, RaptorX-SS8, ICML2014

3-state SS prediction also calculate Segment of Overlap score

- SSpro, RaptorX-SS8, PSIPRED, SPINE-X, JPRED
-

$$\mathrm{max}\theta \ \log\ P\theta\ (\boldsymbol{Y}|\boldsymbol{X}) - \lambda\|\theta\|\ 2$$

θ - set of model parameters

λ - regularization factor used to avoid overfitting

- Large regularization factor => L2-norm of model parameters small
- Restrict search space of model parameters and avoid overfitting
- Too large of a regularization factor may restrict model parameter into too small of a search space -> Underfit
- Log-liklihood not convex - only solve for local optimum
-

$$\Psi(Y, X, i) = \sum_{a,b} T_{a,b}\delta(Y_i = a)\delta(Y_{i+1} = b)$$

- Potential Function for correlation among adjacent SS types around position i
- i indicates position
- a and b represent secondary structure states
- δ() is an indicator function

$$\Phi(Y, X, i) = \sum_{a} \sum_{m} U_{a.m} H_{m}(X, i, W) \delta(Y_i = a)$$

- Model’s relationship between Y_i and input features for i
- H(X, i, W) is a neural network function for the m-th neuron at position i of the top layer
- W, U, T are model parameters to be trained
- W - weighting for convolutional neural net
- U - connection of output of neural net to conditional neural field
- T - connection among nodes in neural field