Graphical Models and Bayesian Methods in Bioinformatics

Zoubin Ghahramani*
http://www.gatsby.ucl.ac.uk

October 21, 2003

Collaborators:

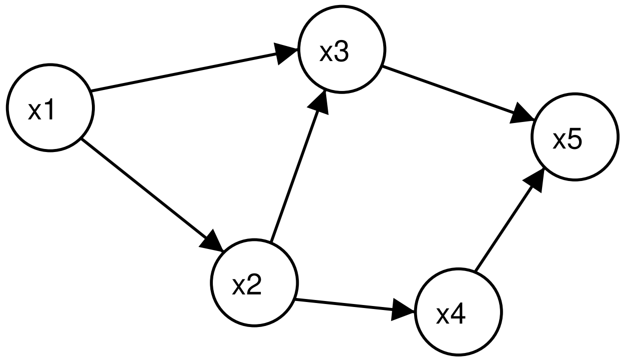
David L. Wild, Keck Graduate Institute of Applied Life Sciences, USA
Wei Chu, *Gatsby Computational Neuroscience Unit, University College London, UK
Carl E. Rasmussen, Max Planck Institute for Biological Cybernetics, Tübingen, Germany
Matthew J. Beal, Dept of Computer Science, University of Toronto, Canada
Francesco Falciani, Biosciences, University of Birmingham, UK

Outline

- Graphical Models
- Bayesian Methods
- Three Applications in Bioinformatics:
  - Protein Structure Prediction *(using Segmental Semi-Markov Models)*
  - Clustering Protein Sequences and Structures *(using Infinite Mixtures)*
  - Inferring Genetic Regulatory Networks *(using State-Space Models)*

Graphical Models

Directed acyclic graph where each node corresponds to a random variable.

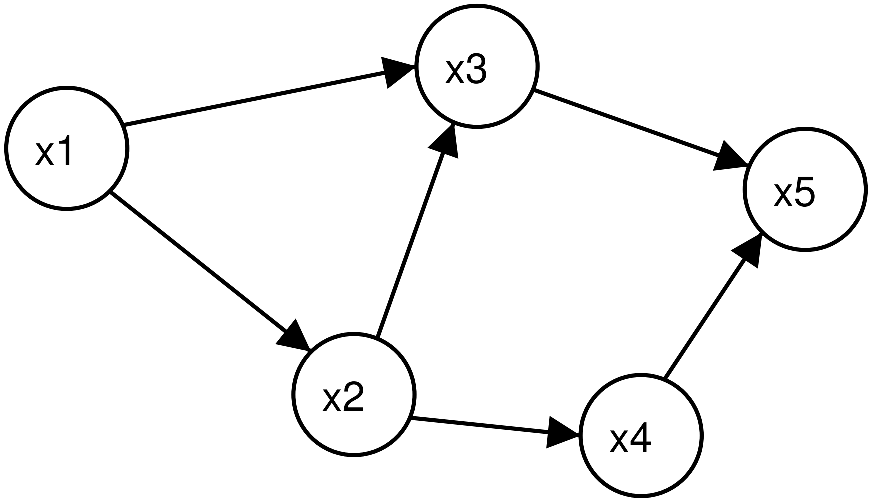


$$P(\mathbf{x}) = P(\mathbf{x}_1)P(\mathbf{x}_2|\mathbf{x}_1)P(\mathbf{x}_3|\mathbf{x}_1,\mathbf{x}_2)$$
$$P(\mathbf{x}_4|\mathbf{x}_2)P(\mathbf{x}_5|\mathbf{x}_3,\mathbf{x}_4)$$

Key quantity: joint probability distribution over nodes: $P(\mathbf{x}) = P(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$
The graph specifies a factorization of this joint pdf: $P(\mathbf{x}) = \prod_i P(\mathbf{x}_i|\text{pa}_i)$
Semantics: Given its parents, each node is *conditionally independent* from its *non-descendents*
Definition: A is *conditionally independent* from B given C if $P(A,B|C) = P(A|C)P(B|C)$ for all A, B, and C s.t. $P(C) \neq 0$.

Also known as Bayesian Networks, Belief Nets and Probabilistic Independence Nets.

Graphical Models

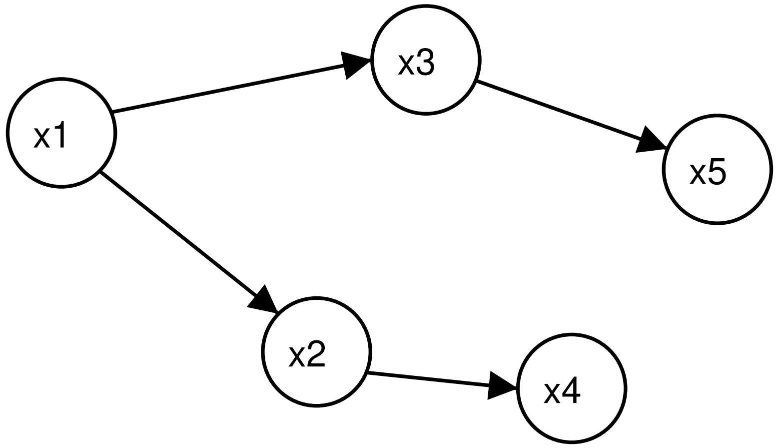


Important advance in statistics and machine learning:

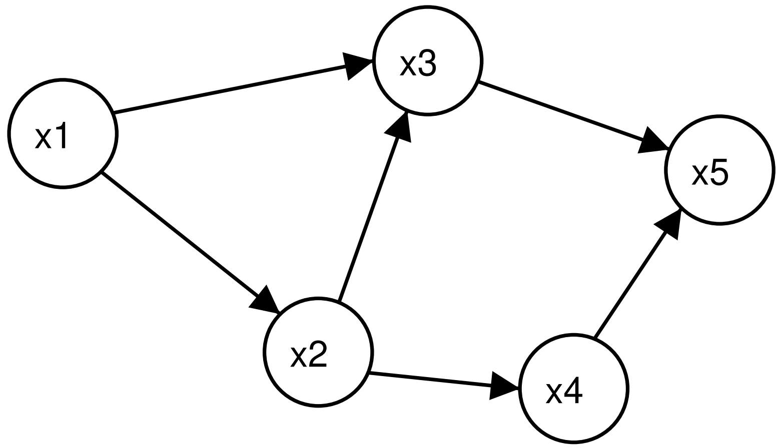
- An intuitive representation of complex multivariate models
- Essential for developing local message passing algorithms for efficient inference.

Inference in Graphical Models

Inference is the problem of computing the probability distribution over some unobserved/hidden/latent variable(s) given some observed variable(s), e.g. $P(\mathbf{x}_i|\mathbf{x}_j)$



Singly connected nets:
The *belief propagation* algorithm.
(e.g. forward–backward for HMMs and Kalman filtering for linear dynamical systems).



Multiply connected nets:
The *junction tree* algorithm.

These are efficient message passing algorithms for applying Bayes rule using the conditional independence relationships implied by the graphical model.

Bayesian Learning

Consider a data set $\mathcal{D}$, and a model m with parameters θ.

Likelihood of model parameters for data set $\mathcal{D}$: $p(\mathcal{D}|\theta, m)$
Prior over model parameters: $p(\theta|m)$
Prior over model class: $p(m)$

The likelihood and parameter priors are combined into the posterior for a particular model using Bayes Rule:

$$p(\theta|\mathcal{D}, m) = \frac{p(\mathcal{D}|\theta, m)p(\theta|m)}{p(\mathcal{D}|m)}$$

Predictions are made by integrating over the posterior:

$$p(x|\mathcal{D}, m) = \int d\theta \; p(x|\theta, \mathcal{D}, m) \; p(\theta|\mathcal{D}, m).$$

Bayesian Model Comparison

A data set $\mathcal{D}$, and a model m with parameters θ.

Likelihood of model parameters for data set $\mathcal{D}$: $p(\mathcal{D}|\theta, m)$
Prior over model parameters: $p(\theta|m)$
Prior over model class: $p(m)$

To compare models m and m', we again use Bayes' rule and the prior on models

$$\frac{p(m|\mathcal{D})}{p(m'|\mathcal{D})} = \frac{p(\mathcal{D}|m)\; p(m)}{p(\mathcal{D}|m')\; p(m')}$$

This also requires an integral over θ:

$$p(\mathcal{D}|m) = \int d\theta\; p(\mathcal{D}|\theta, m)\; p(\theta|m)$$

For interesting models, these integrals may be difficult to compute. *Approximations.*

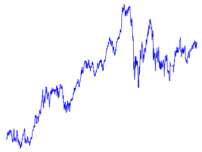
Why be Bayesian?

- Cox Axioms lead to the following:
  If you want to represent uncertain beliefs numerically then, given some basic desiderata, Bayes Rule is the only coherent way to manipulate them.
- The Dutch Book Theorem:
  If you are willing to accept bets with odds proportional to your beliefs, then unless your beliefs satisfy Bayes rule, there exists some set of simultaneous bets which you would accept, but which are guaranteed to lose you money, no matter what the outcome!
- Asymptotic Convergence and Consensus: You will converge to the truth if you assigned it non-zero prior; and different Bayesians will converge to the same posterior as long as they assigned non-zero prior to the same set.
- Automatic: no arbitrary learning rates, no overfitting (no fitting!), honest about ignorance, principled framework for model selection and decision making,...
- Ubiquitous/Fashionable: computer vision, cue integration, concept learning in humans, language learning, robotics, movement control, machine learning, bioinformatics...

Model Comparison and Selection

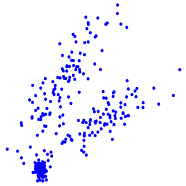
- **Feature Selection**

  Is some input relevant to predicting some output ?



- **Cardinality of Discrete Latent Variables**

  How many clusters in the data?



  How many states in a hidden Markov model?

  SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTY

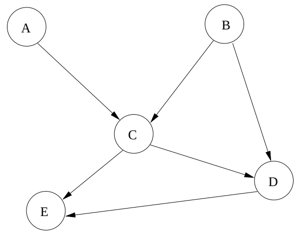
- **Dimensionality of Real Valued Latent Vectors**

  What choice of dimensionality in a PCA/FA model of the data?

  How many state variables in a linear-Gaussian state-space model?

- **Conditional Independence Structure**

  What is the structure of a probabilistic graphical model?



Bayesian Occam's Razor and Model Comparison

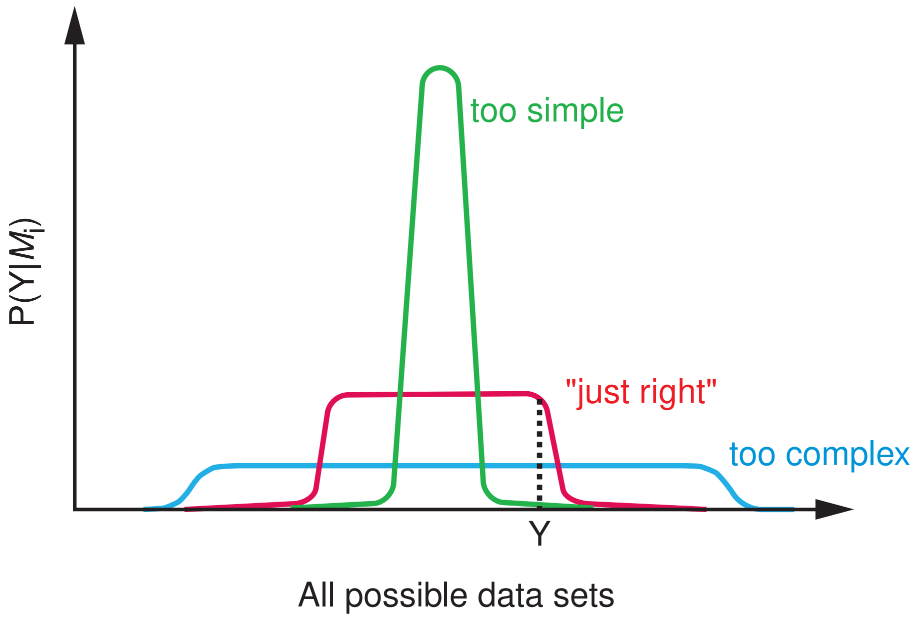
The posterior probability of a model class m_i, given the data, $\mathcal{D}$:

$$p(m_i|\mathcal{D}) = \frac{p(\mathcal{D}|m_i)p(m_i)}{p(\mathcal{D})}, \qquad p(\mathcal{D}|m_i) = \int p(\mathcal{D}|\theta, m_i)\, p(\theta|m_i)\, d\theta$$

Interpretation of the Marginal Likelihood ("evidence"): The probability that *randomly selected* parameters from the prior would generate $\mathcal{D}$.

Model classes that are too simple are unlikely to generate the data set.

Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.

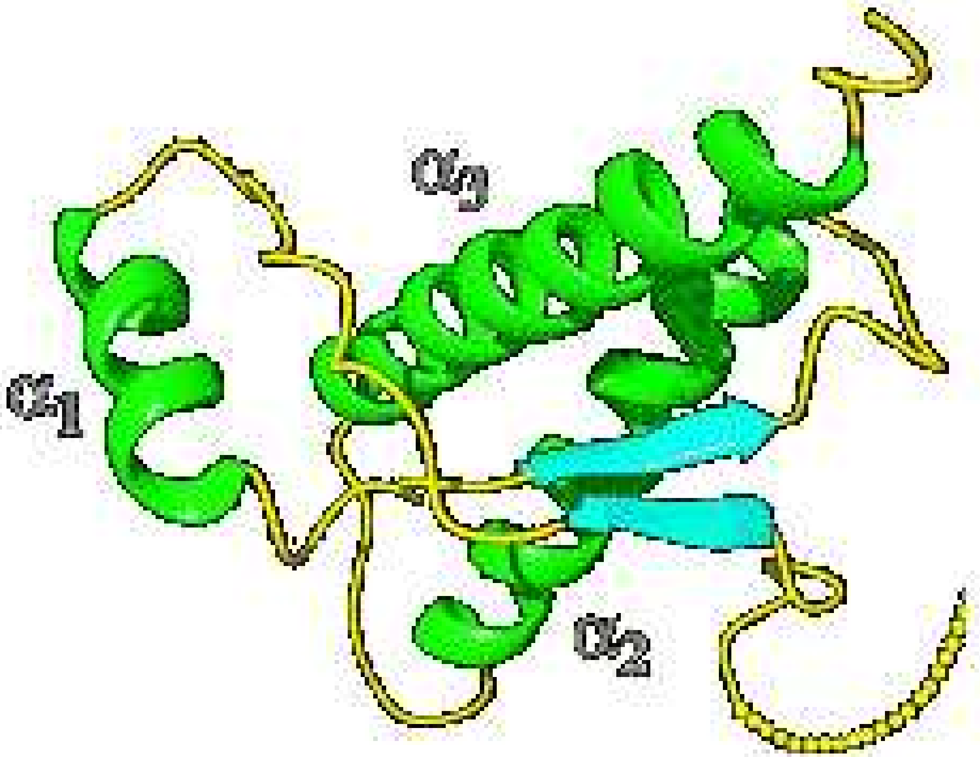


Outline

- Graphical Models
- Bayesian Methods
- Three Applications in Bioinformatics:
  - Protein Structure Prediction *(using Segmental Semi-Markov Models)*
  - Clustering Protein Sequences and Structures *(using Infinite Mixtures)*
  - Inferring Genetic Regulatory Networks *(using State-Space Models)*

Protein Structure Prediction
Using Segmental Semi-Markov Models

Protein Structures



Primary Structure → Secondary Structure → Tertiary Structure

Our Approach

- build a probabilistic generative graphical model that describes the relationship between protein primary structure and its secondary structure;
- incorporate biological constraints and expert knowledge;
- learn model parameters from data sets of proteins with known structure;
- predict structure of novel proteins using Bayesian inference, e.g. belief propagation (forward–backward) and Markov chain Monte Carlo methods.
- extension of Schmidler (2002).

Biological Constraints and Model Properties

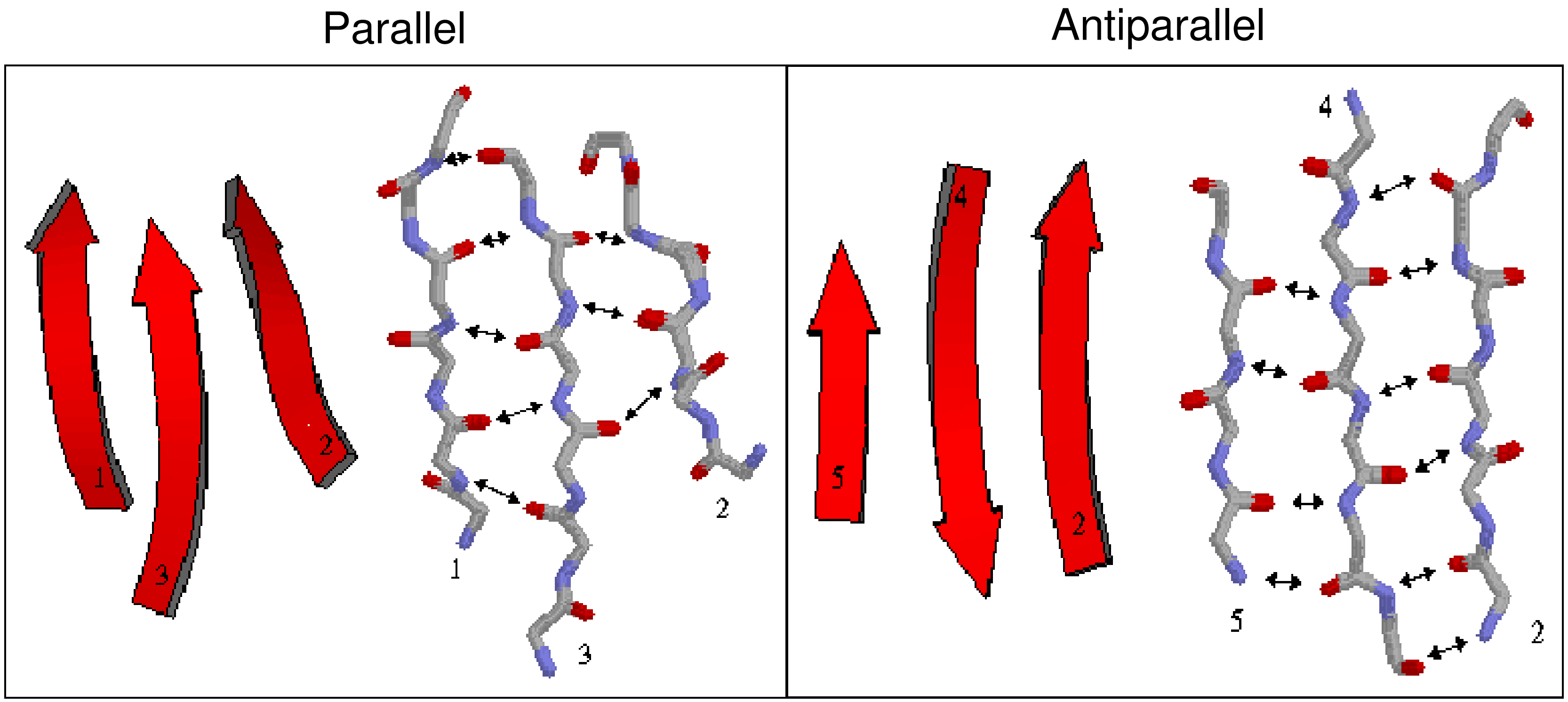
- Basic Protein Properties
  - three segment types (α-Helix, β-Strand, and Coil);
  - different length distributions for different segment types;
  - each segment type has preference on amino acid residues;
  - there are strong local correlations between the amino acid residues;
  - the beginning (N-cap) and end (C-cap) parts of a segment have different distributions of amino acid residues, compared with the internal parts.

Biological Constraints and Model Properties

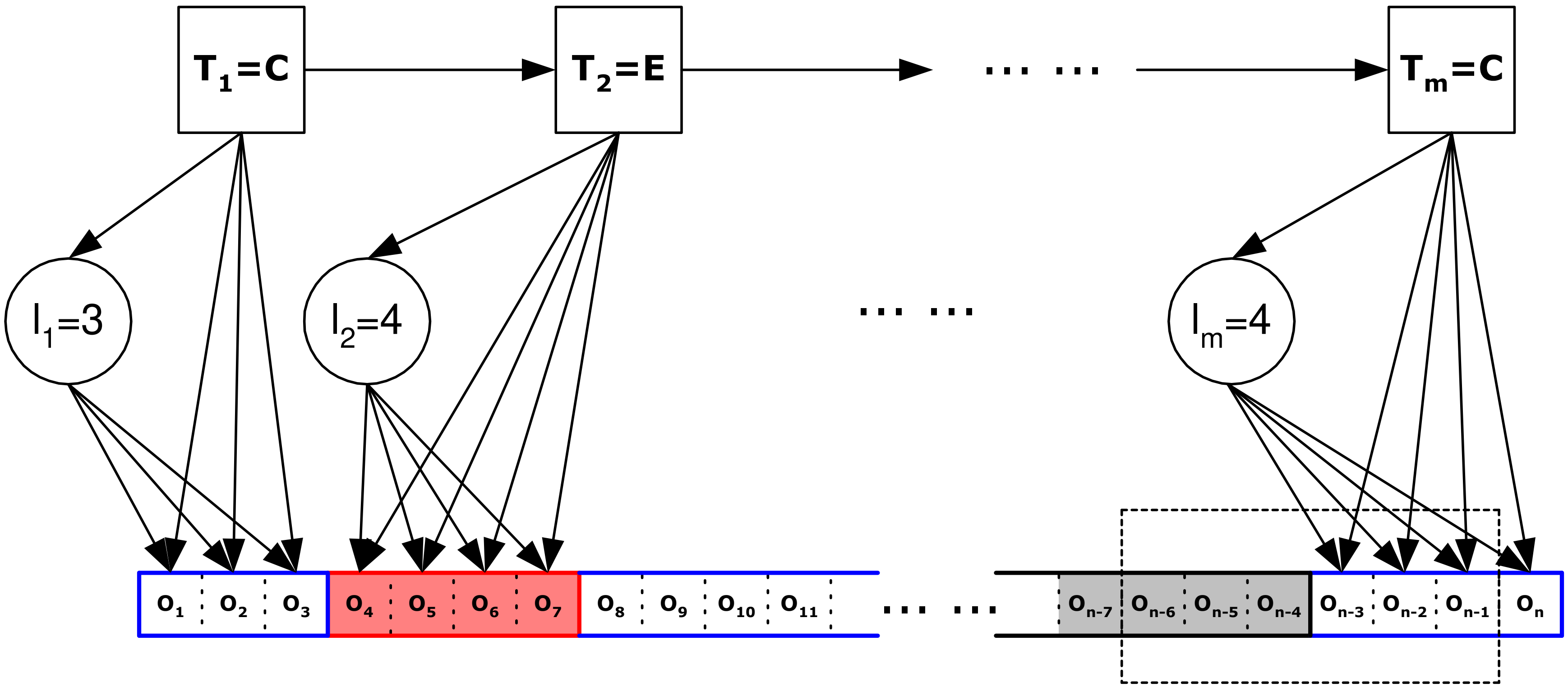
- Evolutionary Information
  - members of a protein family are evolutionarily related;
  - difficult to model evolution explicitly;
  - may use multiple sequence alignment information.
- Multiple Sequence Alignment Profiles
  - use knowledge about closely related sequences to help predict structure;
  - pairwise sequence comparison to search for homologs in a non-redundant sequence database;
  - represent the alignment results as a profile using the technique of multiple sequence alignment.

Biological Constraints and Model Properties

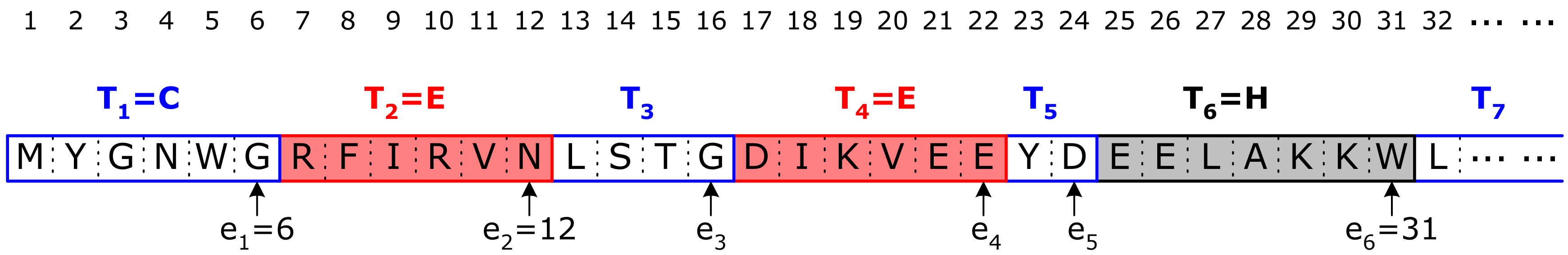
- Long-range Interactions
  - β-strand segments interact through hydrogen bonds to form β-sheets;
  - S-S bonds, etc.



Segmental Semi-Markov Models



Notation for Segmental Models



1. A primary structure, i.e. a sequence of n amino acid residues $R = [R_1, R_2, \ldots, R_n]$;

2. A set of segmental variables, (m, e, T), where m is the number of segments, the segment endpoints $e = [e_1, e_2, \ldots, e_m]$ and the segment types $T = [T_1, T_2, \ldots, T_m]$.

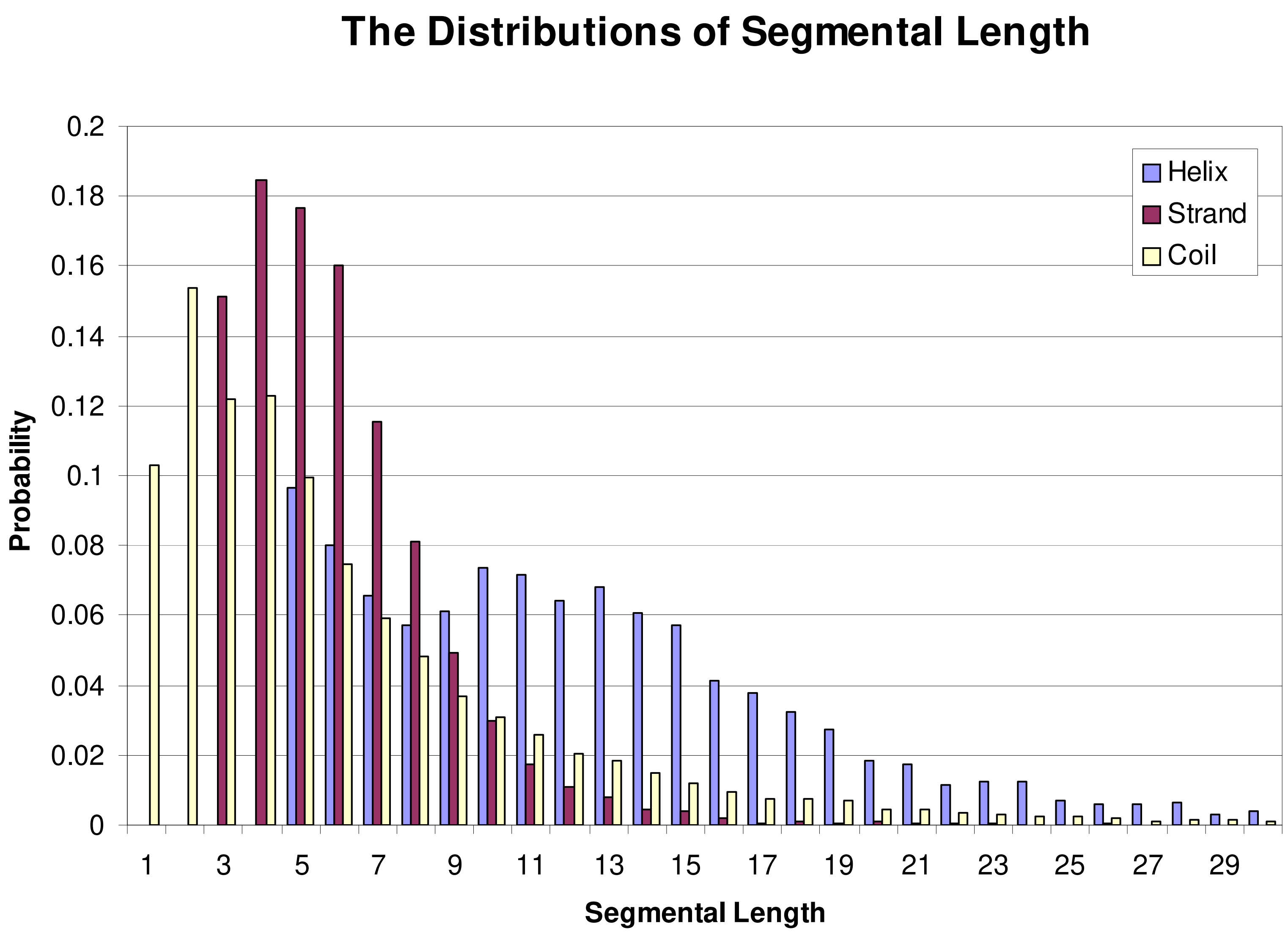
Bayesian Framework

- Prior probability $\mathcal{P}(m, e, T)$:

$$\mathcal{P}(m, e, T) = \mathcal{P}(m)\mathcal{P}(e, T|m) = \mathcal{P}(m)\prod_{i=1}^{m}\mathcal{P}(e_i|e_{i-1}, T_i)\mathcal{P}(T_i|T_{i-1})$$

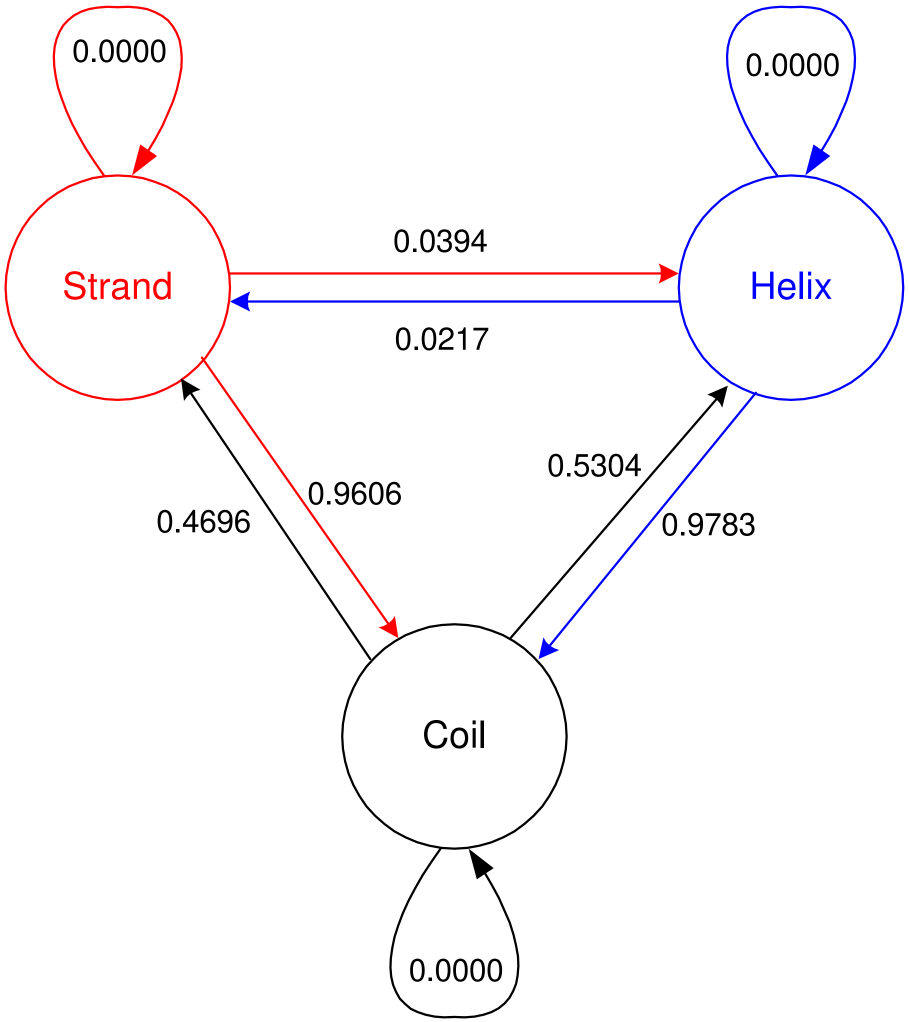
  - segment length distributions $\mathcal{P}(e_i|e_{i-1}, T_i) = \mathcal{P}(l_i = e_i - e_{i-1}|T_i)$;
  - segment type transition probabilities $\mathcal{P}(T_i|T_{i-1})$.
  - a uniform prior for the number of segments m, $\mathcal{P}(m) \propto 1$

A Graph for $\mathcal{P}(l_i = e_i - e_{i-1} | T_i)$



A Graph for $\mathcal{P}(T_i|T_{i-1})$

Segment Type Transition Probabilities



Bayesian Framework

- Likelihood $\mathcal{P}(O|m,e,T)$:

$$\mathcal{P}(O|m,e,T) = \prod_{i=1}^{m} \prod_{k=e_{i-1}+1}^{e_i} \mathcal{P}(O_k|T_i, O_{[1:k-1]})$$

m - the number of segments;
e - the segment endpoints;
T - the segment types;
O - the observations: the sequence of residues and the profile;
O_k - the observation at the k-th residue, i.e. the residue type R_k and the alignment information M_k.

Multiple Sequence Alignment Profiles

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	··· ···
			T_1=C			e_1=6			T_2=E			e_2=12		T_3		e_3				T_4=E		e_4	T_5	e_5			T_6=H				e_6=31		T_7
Residue	M	Y	G	N	W	G	R	F	I	R	V	N	L	S	T	G	D	I	K	V	E	E	Y	D	E	E	L	A	K	K	W	L	··· ···
Alignment 1	M	Y	G	N	W	G	R	F	I	R	V	N	L	S	T	G	D	I	K	V	E	E	Y	D	E	E	L	A	K	K	W	L	··· ···
Alignment 2	A	N	G	W	T	G	N	I	L	R	V	N	L	T	T	G	N	I	T	L	E	D	S	S	K	.	.	F	K	S	F	V	··· ···
Alignment 3	.	.	.	M	R	D	V	V	L	V	I	D	L	N	N	.	Y	T	Y	E	V	E	H	R	P	E	L	F	E	K	W	L	··· ···
Alignment 4	M	Y	G	Y	C	G	K	I	L	R	V	D	L	S	E	K	T	V	S	T	L	I	P	E	E	K	D	L	R	E	F	I	··· ···
Alignment 5	M	K	G	W	W	G	K	I	L	R	V	D	L	T	N	N	K	V	W	V	Q	E	Y	S	P	E	V	A	K	N	F	I	··· ···
Alignment 6	M	M	G	F	H	G	K	I	L	E	V	N	L	S	N	S	T	V	S	E	I	E	V	T	E	E	M	I	K	K	F	V	··· ···
Alignment 7	A	G	G	Y	M	G	K	I	L	K	V	N	L	S	E	G	S	I	K	E	L	P	I	D	E	E	I	A	R	K	Y	L	··· ···

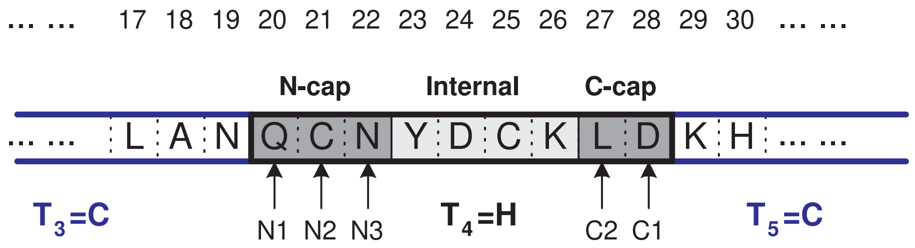
Appearance counts for each amino acid down a column in the profile is denoted as $M = [M_1, M_2, \ldots, M_n]$, where M_i is a 20×1 counting vector.

Observation Model

The likelihood of individual observations

$$\mathcal{P}(O_k|T_i, O_{[1:k-1]}) = \mathcal{P}(M_k|R_k, O_{[1:k-1]}, T_i) \cdot \mathcal{P}(R_k|O_{[1:k-1]}, T_i)$$

- $\mathcal{P}(R_k|O_{[1:k-1]}, T_i) = \mathcal{P}_c(R_k|T_i)$, as position-specific distributions for capping signals, where $c \in \{N1, N2, \ldots, \mathrm{Internal}, \ldots, C2, C1\}$;



- $\mathcal{P}(M_k|R_k, O_{[1:k-1]}, T_i, \boldsymbol{W}_{T_i})$, as Dirichlet-Multinomial distributions with parameterization for segmental dependency.

Posterior Probability

The posterior probability can be written as

$$\mathcal{P}(m, e, T|O) = \frac{\mathcal{P}(O|m, e, T) \cdot \mathcal{P}(m, e, T)}{\mathcal{P}(O)}$$

- **MAP:** the most likely segmentation, $\arg \max_{(m,e,T)} \mathcal{P}(m, e, T|O)$; (Viterbi algorithm)

- **Marginal:** the marginal posterior distribution of the segment type at each residue, $\mathcal{P}(T_{O_i}|O)$, where T_{O_i} denotes the segment type at the i-th residue. (Forward–Backward Algorithm)

Parameter Estimate

- The parameters for discrete distributions, including the state transition probability $\mathcal{P}(T_i|T_{i-1})$, segmental length distribution $\mathcal{P}(e_i|e_{i-1}, T_i)$, and the position-specific distributions of amino acids $P_c(R|T_i)$ etc, can be estimated by their relative frequency of occurrence in the training dataset.

- The weights in segmental likelihood, which consist of three subsets for different segmental types, i.e. $\{\boldsymbol{W}_\tau\}$ with $\tau \in \{H, E, C\}$.

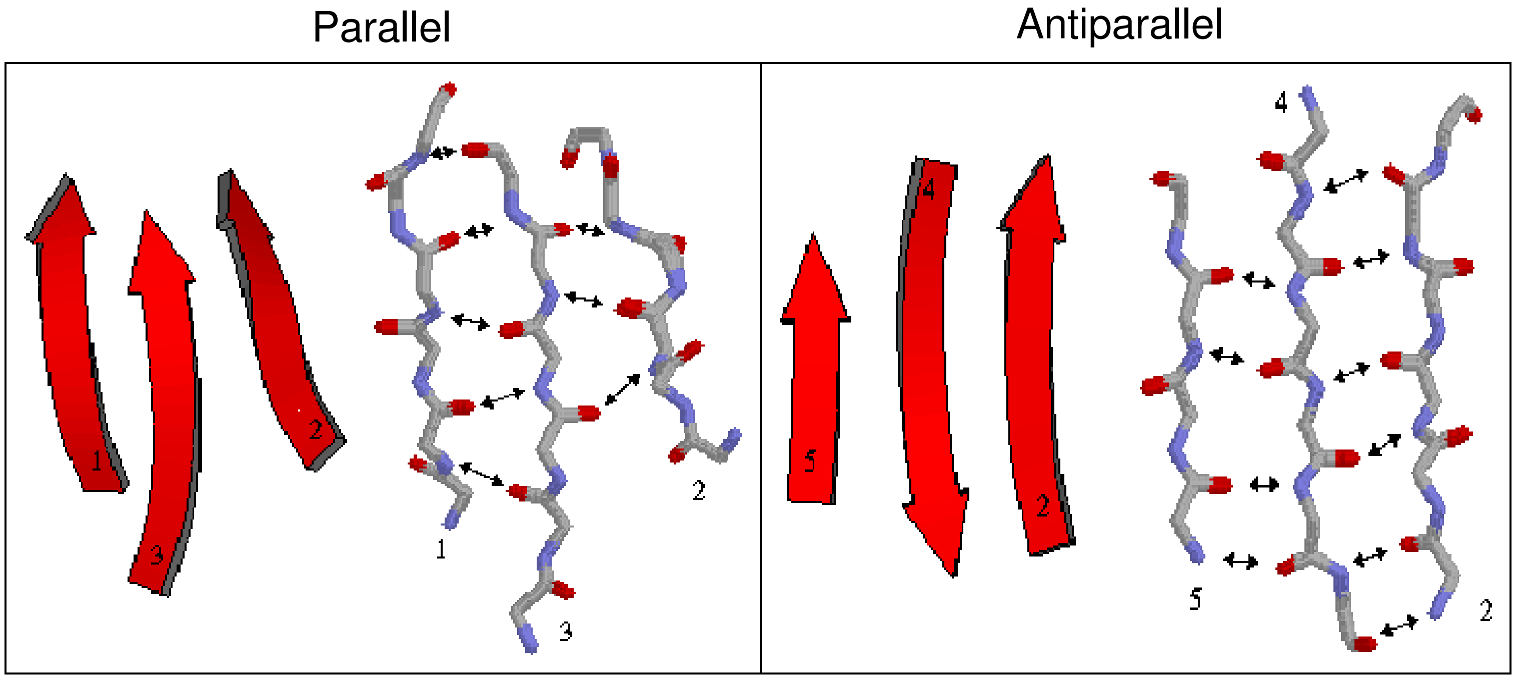
$$\arg\max_{\boldsymbol{W}_\tau} \mathcal{P}(\{O, m, e, T, \mathcal{I}\}|\boldsymbol{W}_\tau)\mathcal{P}(\boldsymbol{W}_\tau)$$

where $\mathcal{P}(\boldsymbol{W}_\tau) \propto \exp(-\frac{C_\tau}{2}\|\boldsymbol{W}_\tau\|_2^2)$ with $C_\tau \geq 0$.

Results of Cross Validation on Benchmark

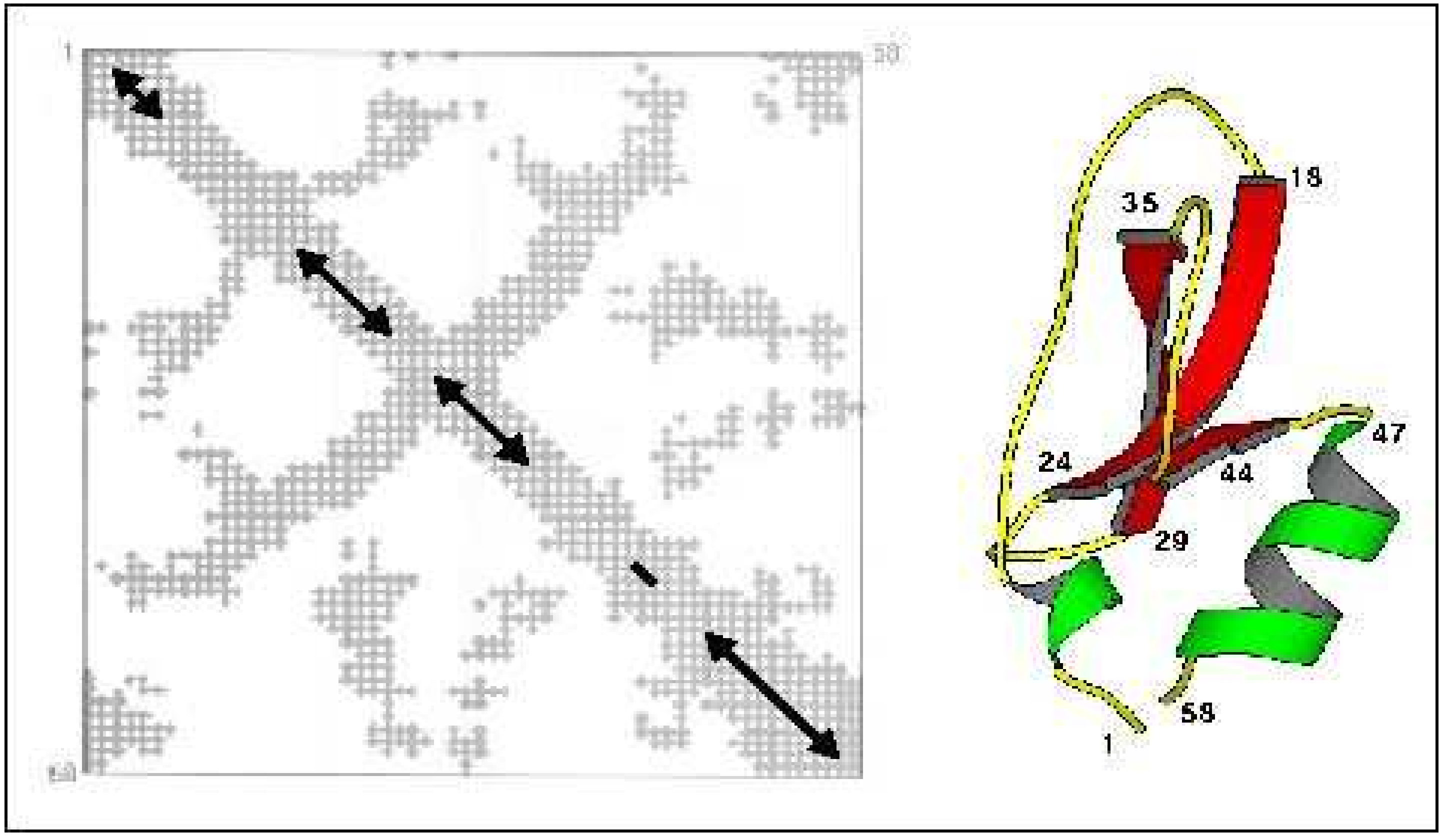
Dataset	480 Protein Sequences from CB513				119 Protein Sequences from CB513			
	Schmidler et al. (2000)		with Profiles		Schmidler et al. (2000)		with Profiles	
	MAP	MARG	MAP	MARG	MAP	MARG	MAP	MARG
Q_3	56.83%	**64.94%**	67.40%	**72.34%**	55.70%	**64.63%**	68.26%	**72.83%**
Q_H^{obs}	60.65%	66.33%	78.07%	78.34%	57.73%	67.05%	77.18%	79.87%
Q_E^{obs}	28.97%	46.74%	42.96%	56.73%	20.46%	43.48%	39.52%	53.78%
Q_C^{obs}	67.62%	72.87%	71.10%	75.35%	69.73%	72.04%	73.53%	75.37%
Q_H^{pred}	60.30%	66.81%	68.69%	74.46%	60.23%	67.09%	71.15%	74.89%
Q_E^{pred}	42.80%	58.17%	66.34%	69.55%	38.33%	56.72%	67.20%	70.35%
Q_C^{pred}	58.51%	66.06%	66.63%	71.74%	56.06%	65.20%	66.10%	71.86%
C_H	0.3289	0.4510	0.5385	0.6083	0.2876	0.4364	0.5409	0.6072
C_E	0.1691	0.3776	0.4157	0.5212	0.1200	0.3657	0.4141	0.5219
C_C	0.2416	0.3966	0.4090	0.5036	0.2160	0.3851	0.4219	0.5091
SOV_H	38.22%	59.96%	54.08%	68.72%	36.84%	59.19%	55.27%	68.45%
SOV_E	45.76%	60.51%	59.87%	71.48%	45.70%	62.25%	59.50%	72.63%
SOV_C	28.97%	58.03%	46.41%	66.70%	20.22%	53.50%	44.44%	63.71%

Long-range Interactions in β-sheets



- The β-sheet space is the set of all the possible combinations of β-sheets;
- A set of interaction variables, $\mathcal{I}$, to describe one possible case.
- Long-range interactions make exact inference intractable, but we can sample from $\mathcal{I}$ using Metropolis MCMC with reversible-jumps (Green, 1995)

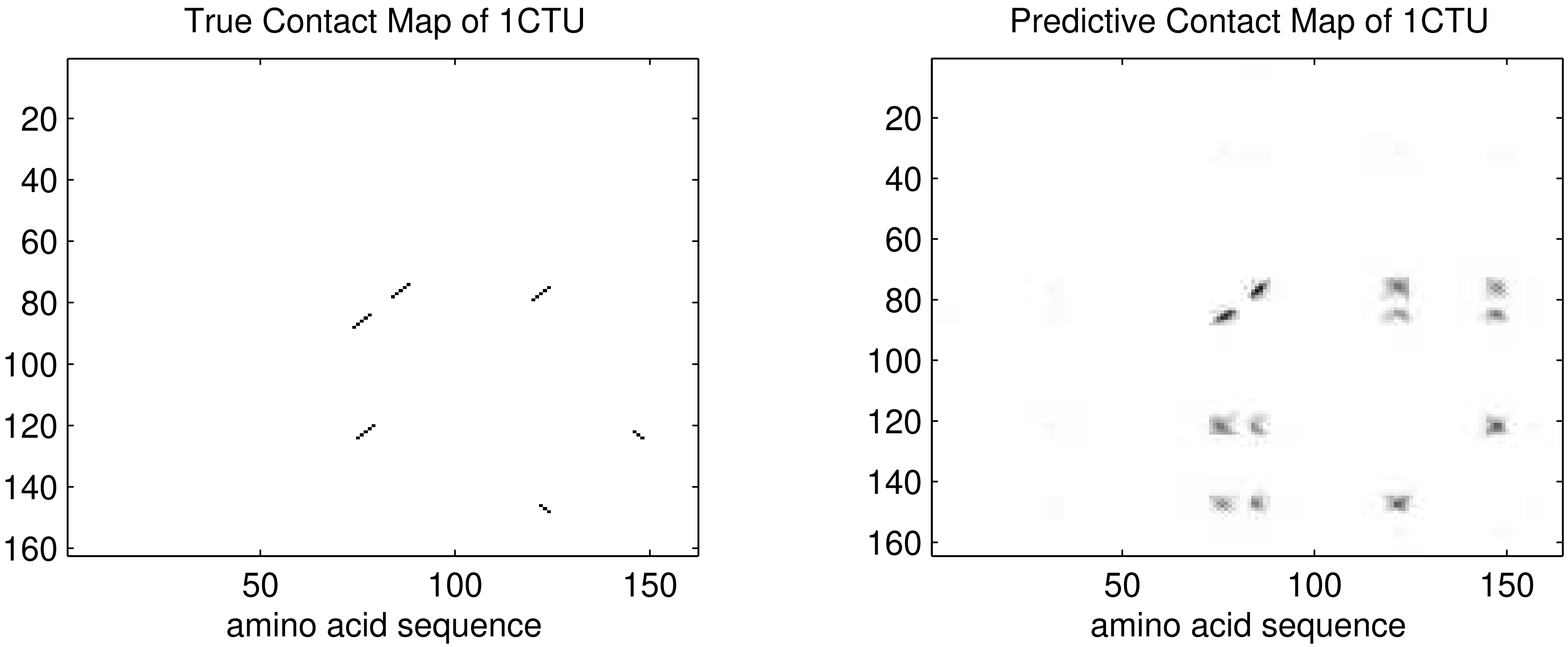
Contact Maps



Inferring Contact Maps

An interaction set $\mathcal{I}$ is associated with a β-sheet contact map defined by a $n \times n$ matrix $\mathcal{C}$ whose ij-th entry $\mathcal{C}^{ij}$ defined as

$$\mathcal{C}^{ij}(\mathcal{I}) = \begin{cases} 1 & \text{if } R_i \text{ and } R_j \text{ are paired (in contact) in the interaction set } \mathcal{I}; \\ 0 & \text{otherwise} \end{cases}$$



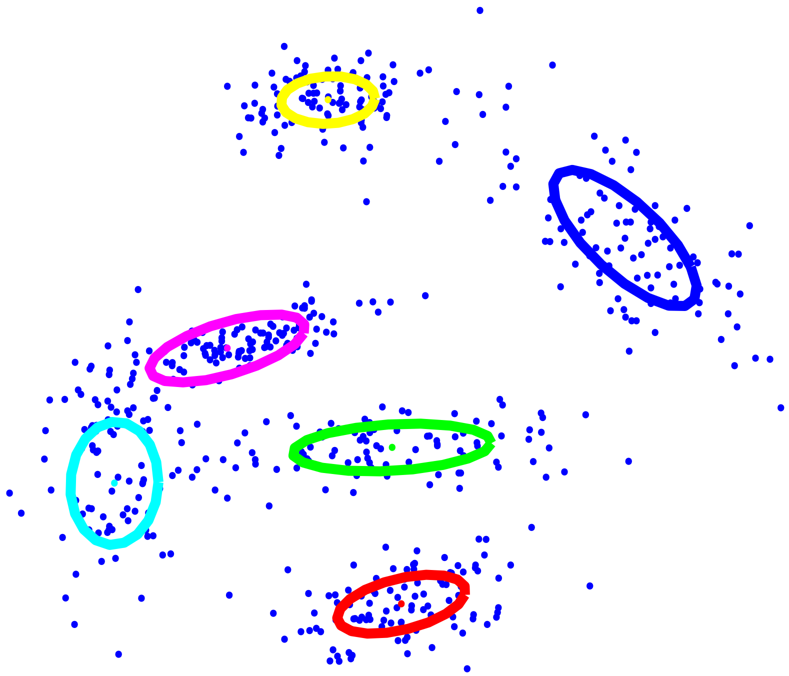
Summary

- build a probabilistic generative model for protein structure prediction;
- employ evolutionary information in the form of multiple sequence alignment profiles;
- the incorporation of the profiles greatly improves performance;
- extend the model to incorporate the long-range interactions in β-sheets;
- carry out inference on contact maps in the Bayesian segmental framework;
- attempt tertiary structure prediction in near future ?

Clustering Protein Sequences and Structures using Infinite Gaussian Mixture Models

Finite Gaussian Mixture Models

- **Data:** features of protein sequences or gene expression profiles which can be arranged into p-dimensional vectors $\boldsymbol{y}$;
- **Model:** a Gaussian mixture model with a finite number (k) of Gaussians, with parameters ϕ_j, $\boldsymbol{\mu}_j$ and Σ_j:



$$
\begin{aligned}
P(\boldsymbol{y}) &= \sum_{j=1}^{k} \phi_j P_j(\boldsymbol{y}) \\
\phi_j &: \text{ the mixing proportion for cluster } j \\
&\quad (\textstyle\sum_j \phi_j = 1;\ \phi_j \geq 0) \\
P_j(\boldsymbol{y}) &: \text{ a Gaussian with mean } \boldsymbol{\mu}_j \text{ and cov matrix } \Sigma_j
\end{aligned}
$$

Infinite Gaussian Mixture Models

- We can choose a flexible model with *infinitely* many components, as long as we do Bayesian average over parameters! The data automatically tells us how many components we need.
- Let $c_i = j$ denote "data point i belongs to cluster j"
- Before observing $\boldsymbol{y}$: $P(c_i = j|\phi) = \phi_j$, proportional to the mixing proportion;
- Choose prior over ϕ_j to be the symmetric Dirichlet distribution, where α controls how evenly mass is spread between clusters: $P(\phi|\alpha) = \dfrac{\Gamma(\alpha)}{\Gamma(\alpha/k)^k} \displaystyle\prod_{j=1}^{k} \phi_j^{\alpha/k-1}$
- *Integrating over* all possible settings of ϕ we get the conditional probability:
$$P(c_i = j|\mathbf{c}_{-i}, \alpha) = \int P(c_i = j|\mathbf{c}_{-i}, \phi) P(\phi|\mathbf{c}_{-i}, \alpha)\, d\phi = \frac{n_{-i,j} + \alpha/k}{n - 1 + \alpha}$$
- In the extreme case $k \to \infty$ we get infinite Gaussian mixtures (a.k.a. Dirichlet process mixtures).
- **Goal:** to infer $P(c_i = c_\ell|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n, \alpha)$: "the probability that proteins i and ℓ belong to the same cluster"

Methodology

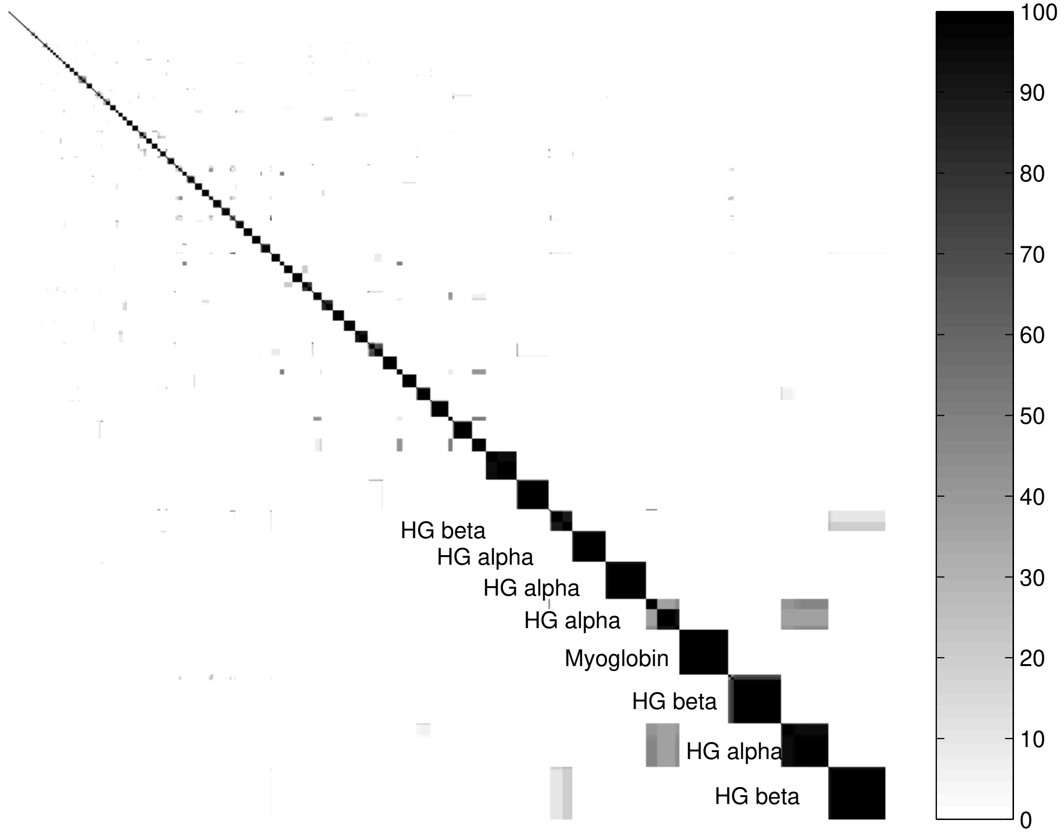
- vector representation: Fisher score vector representation described by Jaakkola et al (2000);

- structural information included using structure-based hidden Markov models (Raval et al. 2000);

- principal components analysis: the dimensionality of the Fisher score vector was then reduced to 10;

- infinite Gaussian mixture models: a large number of Gibbs sampling sweeps are performed.

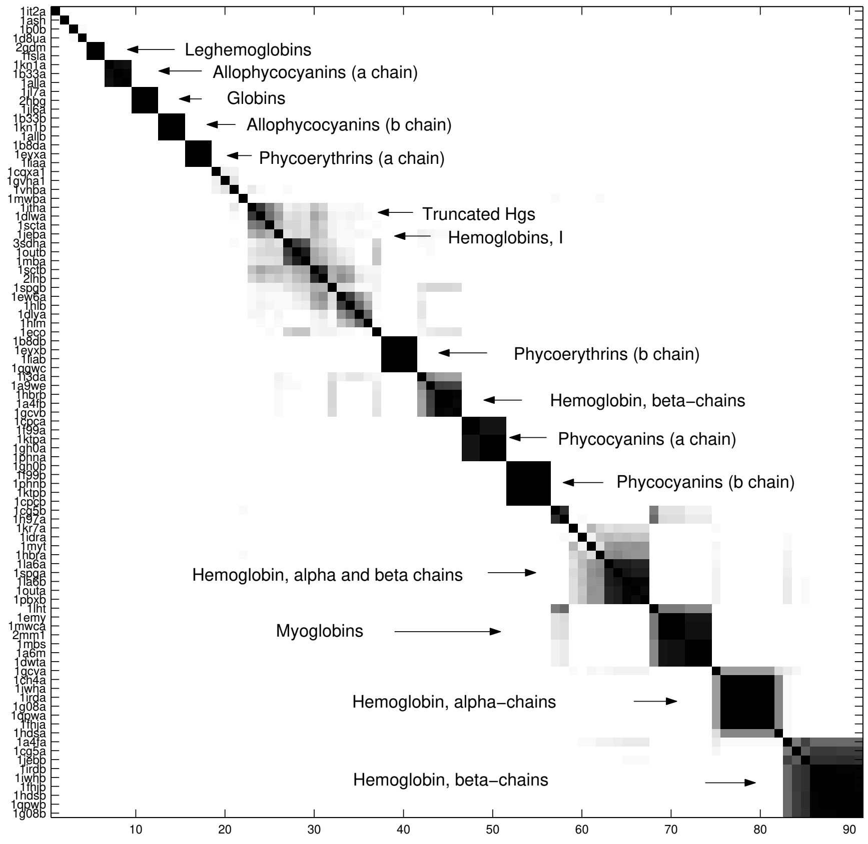
Results - Globin Sequences



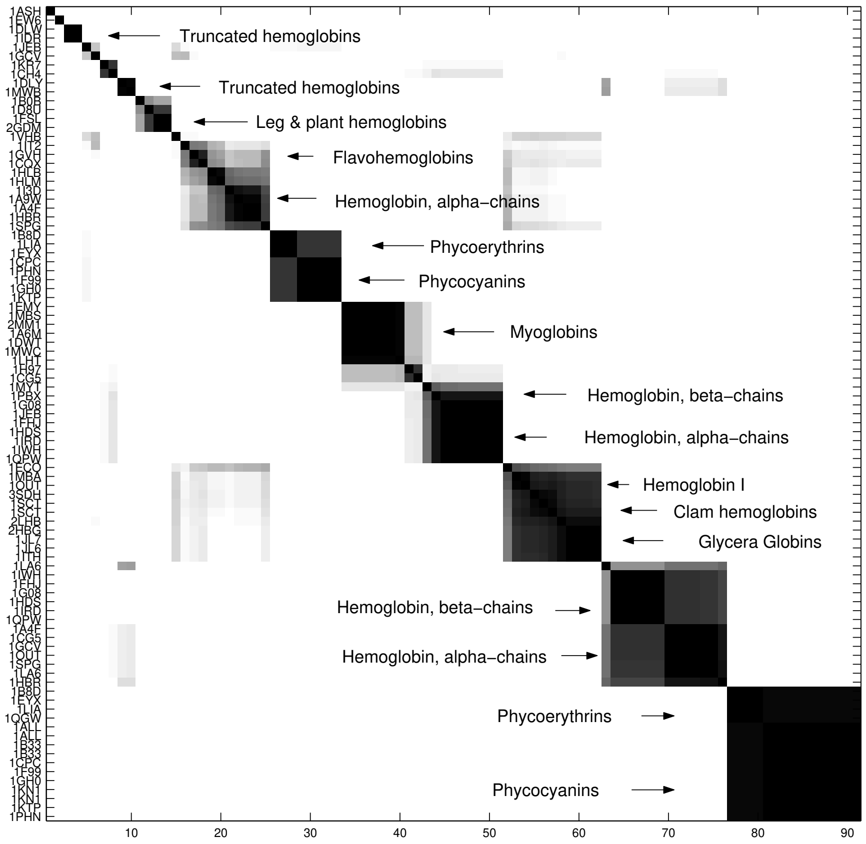
Finds larger number of clusters including biologically meaningful sub-clusters.

Results - SCOP Globin Sequences

sequence only

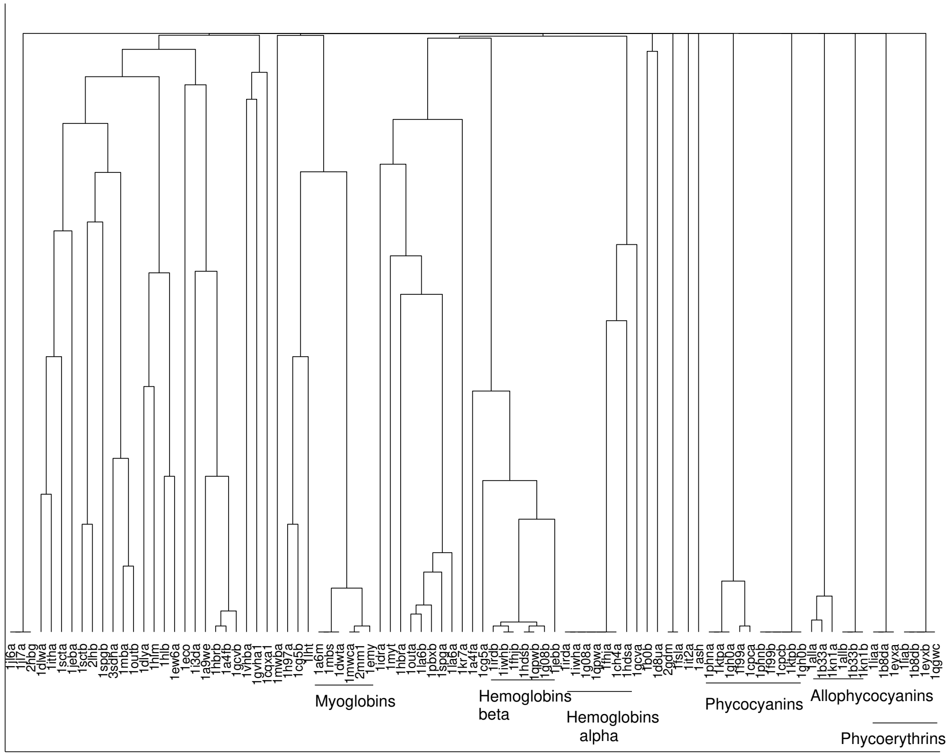


sequence + structural information

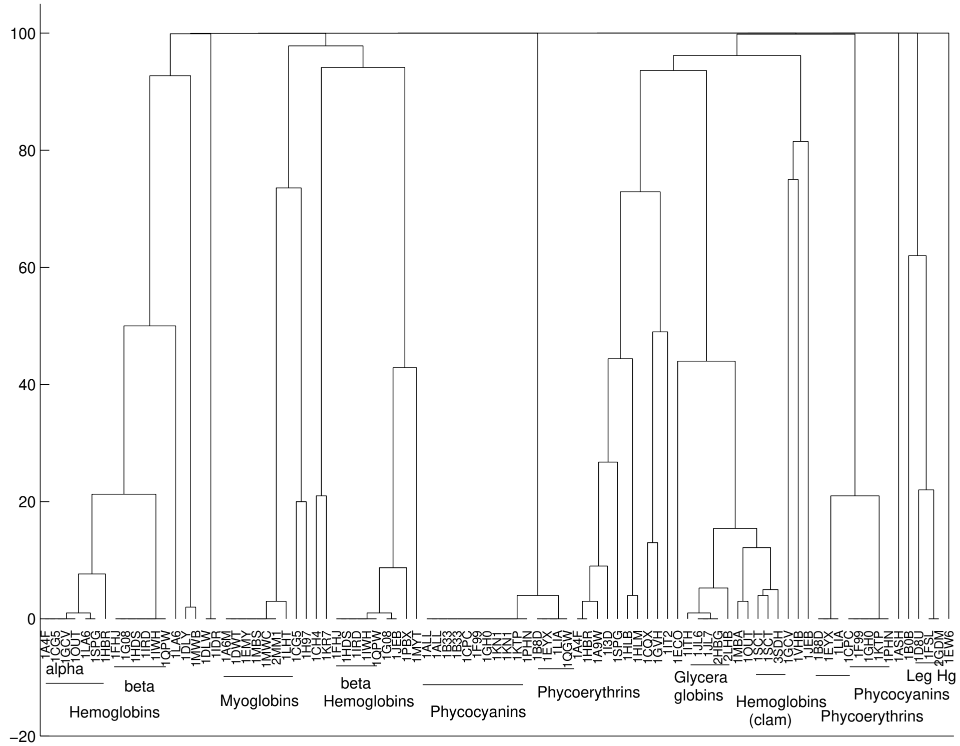


Results - SCOP Globin Sequences

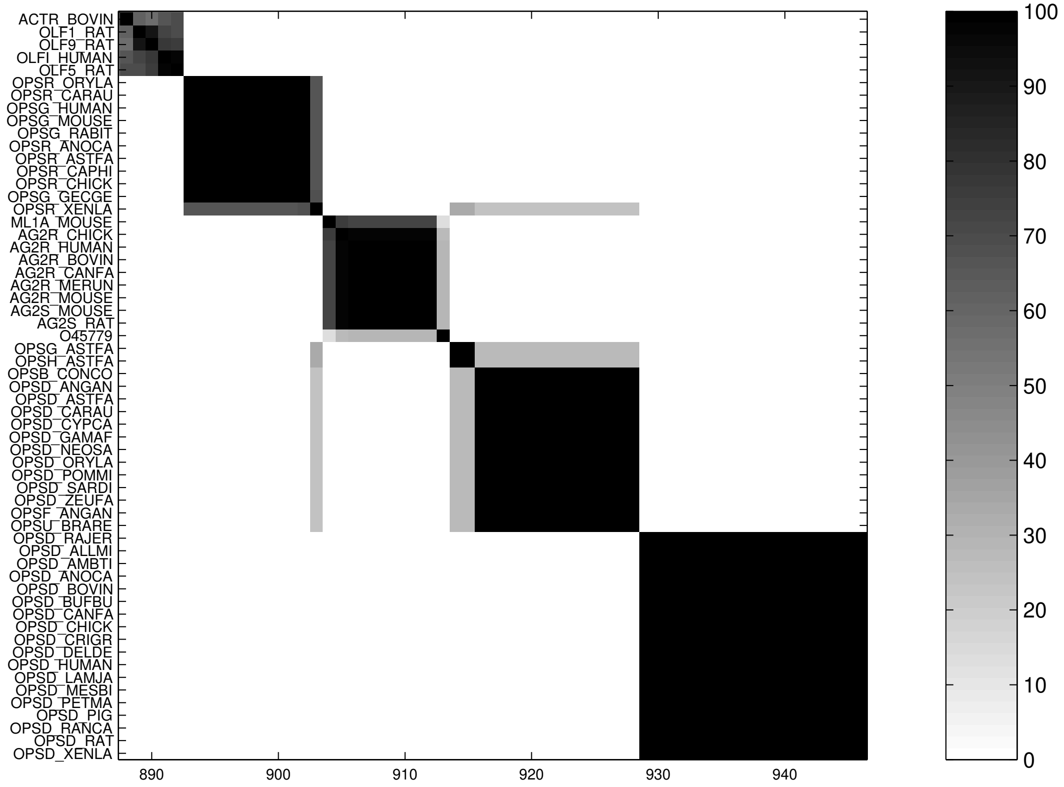
sequence only



sequence + structural information



Results - GPCR Sequences



(sequence only)

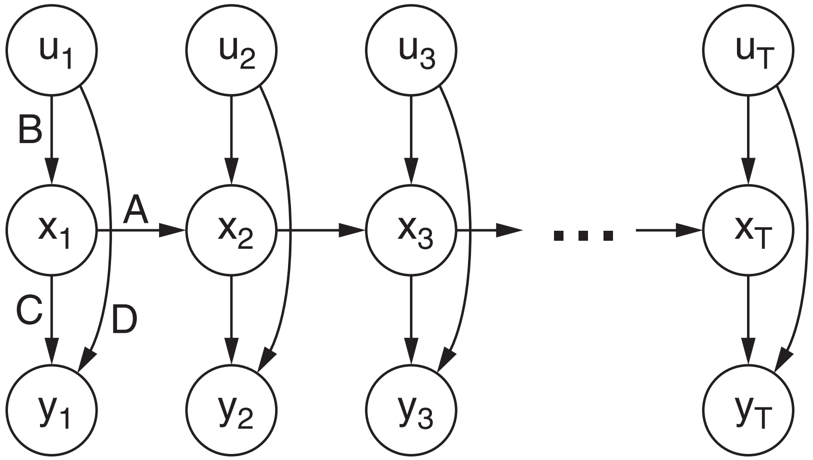
Some are orphan receptors with no known function

We can use the clustering to suggest putative functions.

We've also used infinite mixtures for clustering microarray gene expression data.

Inferring Genetic Regulatory Networks using State-Space Models

Linear-Gaussian State-space models (SSMs)



$$P(\mathbf{x}_{1:T}, \mathbf{y}_{1:T} | \mathbf{u}_{1:T}) = P(\mathbf{x}_1 | \mathbf{u}_1) P(\mathbf{y}_1 | \mathbf{x}_1, \mathbf{u}_1) \prod_{t=2}^{T} P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) P(\mathbf{y}_t | \mathbf{x}_t, \mathbf{u}_t)$$

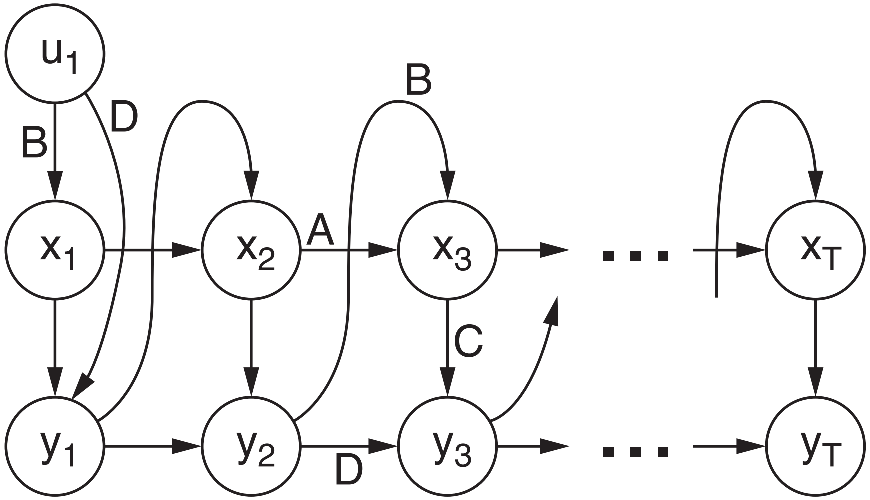
Output equation: $\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t + \mathbf{v}_t$

State dynamics equation: $\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + \mathbf{w}_t$

where $\mathbf{x}_t$, $\mathbf{u}_t$ and $\mathbf{y}_t$ are real-valued vectors and $\mathbf{v}$ and $\mathbf{w}$ are uncorrelated zero-mean Gaussian noise vectors.

These models are a.k.a. stochastic linear dynamical systems, Kalman filter models. Continuous state version of HMMs.

State-Space Models with Feedback



Output equation: $$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{y}_{t-1} + \mathbf{v}_t$$

State dynamics equation: $$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{y}_{t-1} + \mathbf{w}_t$$

Key Concept: $\mathbf{y}_t$ represents the measured gene expression level at time step t and $\mathbf{x}_t$ models the many unmeasured (hidden) factors such as

- genes that have not be included in the microarray,
- levels of regulatory proteins,
- the effects of mRNA and protein degradation, etc.

Our Approach

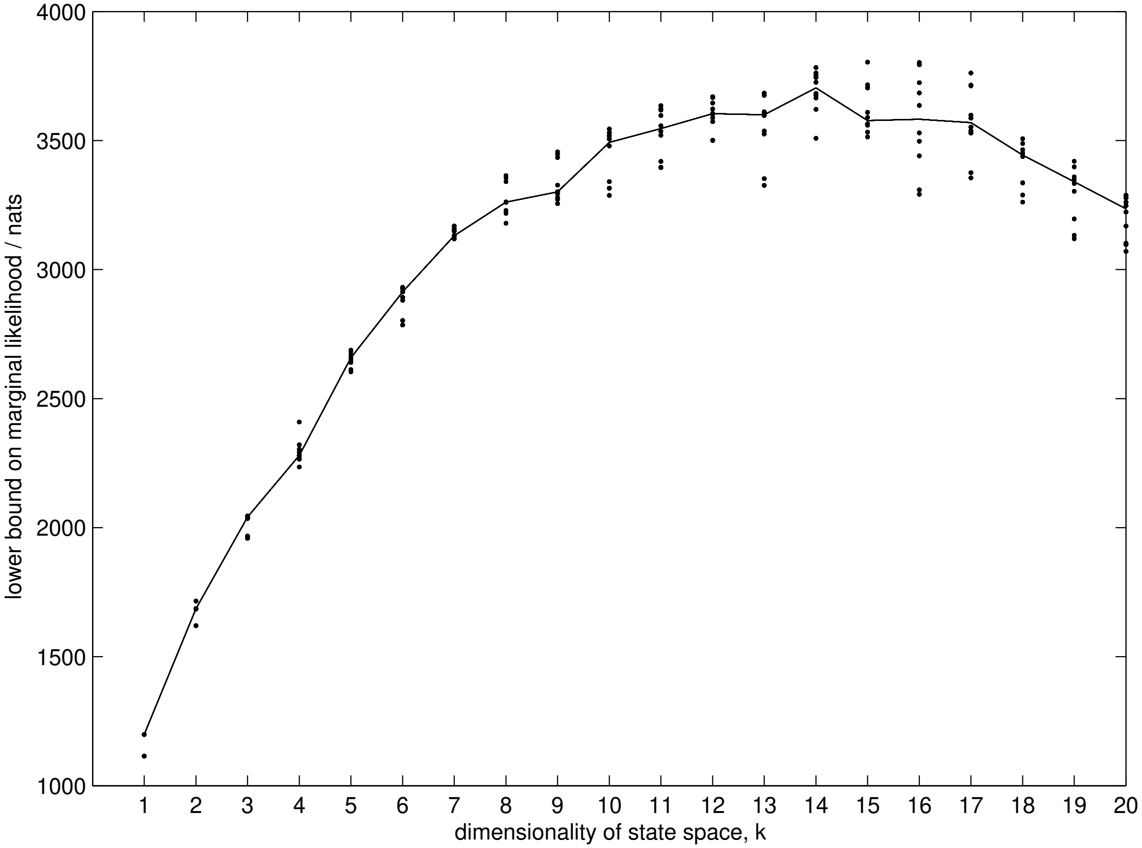
- Let $\theta = \{A, B, C, D, R\}$ be the parameters of the model (R models noise covariance).

- Exact Bayesian inference would give us $p(\theta|\mathcal{D})$, which tells us confidence in each parameter and can be used to infer model structure.

- Unfortunately, exact inference is computationally intractable.

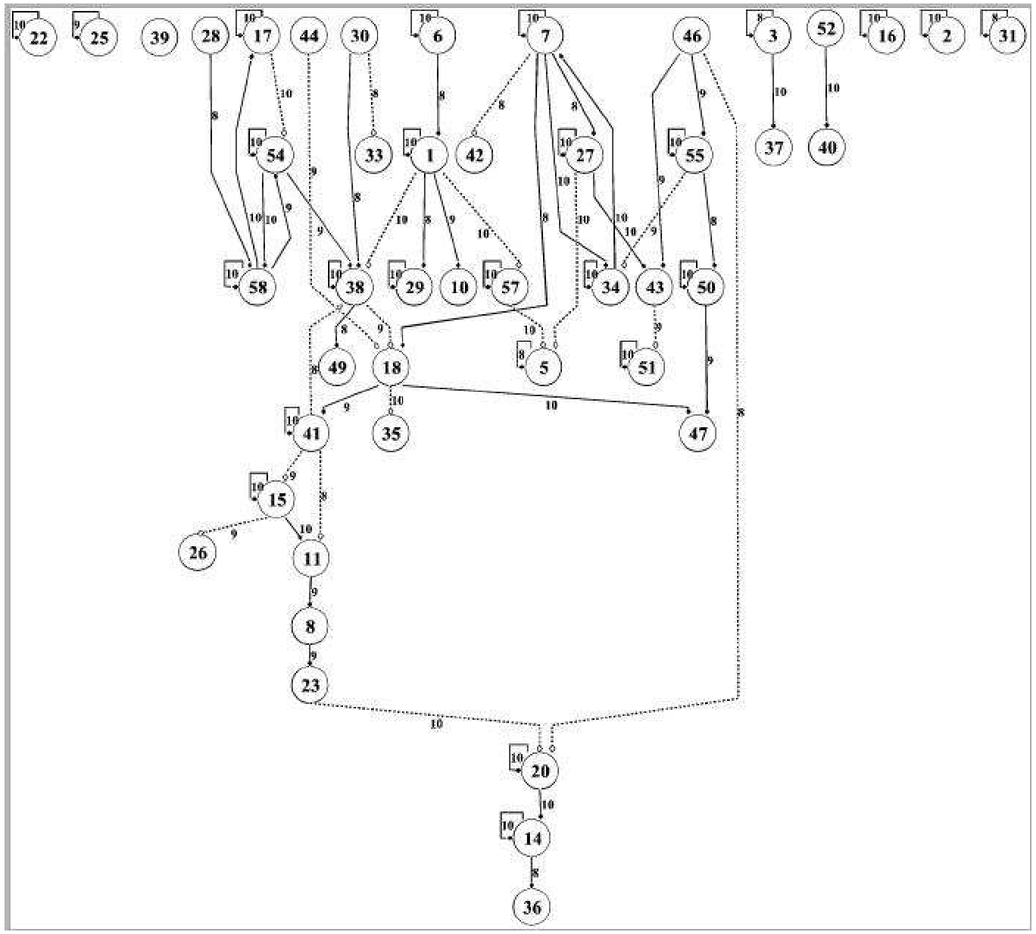
- Use variational approximations to approximate Bayesian inference in state-space models (Beal & Ghahramani, 2003).

Inferring the Number of Hidden States



We can do use a lower bound on the marginal likelihood $P(\mathcal{D}|m)$ to select an appropriate model class m.

Inferring Regulatory Networks (preliminary)



transcriptional networks in T cell activation $\rightarrow$ testable hypotheses

Conclusions

- Graphical models and Bayesian methods can be used for a variety of modelling problems in bioinformatics

- These allow large-scale statistical models to be learned and sources of noise and uncertainty to be included in a principled manner

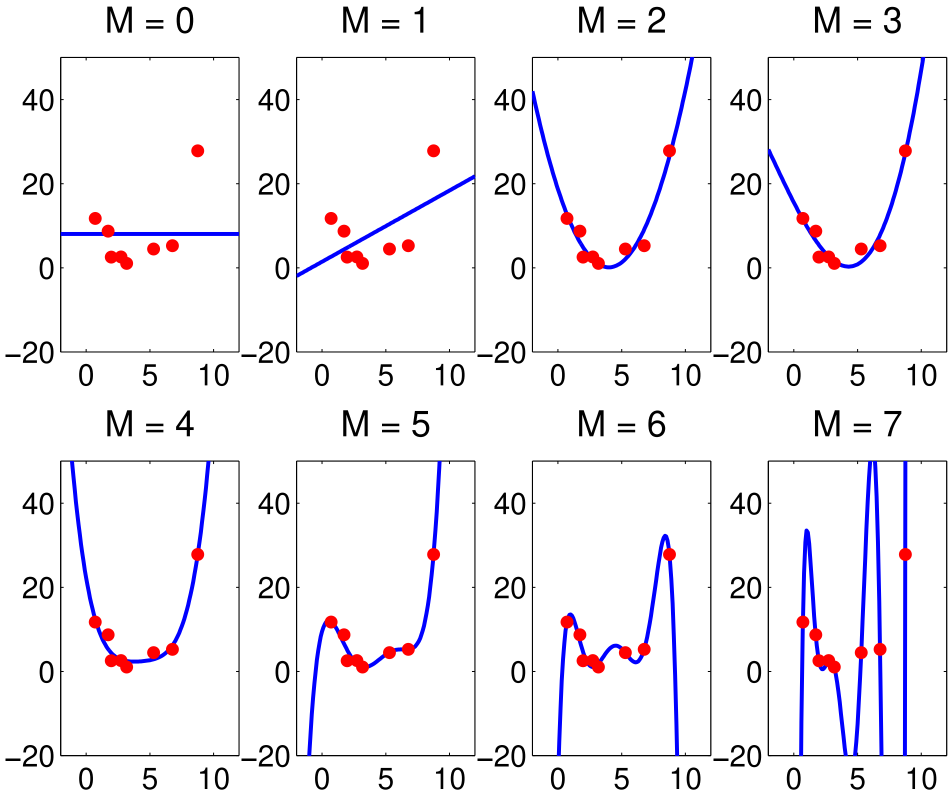
- We have looked at three problem domains: protein structure determination, protein clustering, and inferring genetic regulatory networks.

Acknowledgements

- Anaya Dubey, Seungwoo Hwang, Claudia Rangel, KGI

- This work is supported by the National Institutes of Health (NIH) and its National Institute of General Medical Sciences (NIGMS) division under Grant Number 1 P01 GM63208 (NIH/NIGMS grant title: Tools and Data Resources in Support of Structural Genomics)

Model structure and overfitting: a simple example



Protein Secondary Structure Prediction

- Discriminant approach with neural networks,
  - seminal work by Qian and Sejnowski (1988);
  - PHD (Rost and Sander, 1993) employed evolutionary information in the form of profiles derived from multiple sequence alignment;
  - Jones (1999) proposed an algorithm on the basis of position-specific scoring matrices (PSSM);
  - Cuff and Barton (2000) evaluated different types of multiple sequence alignment profiles.

- Generative modelling, e.g. segmental semi-Markov models (SSMM) (Schmidler, 2002), using primary structure only with lower prediction accuracy.

Protein Tertiary Structure Prediction

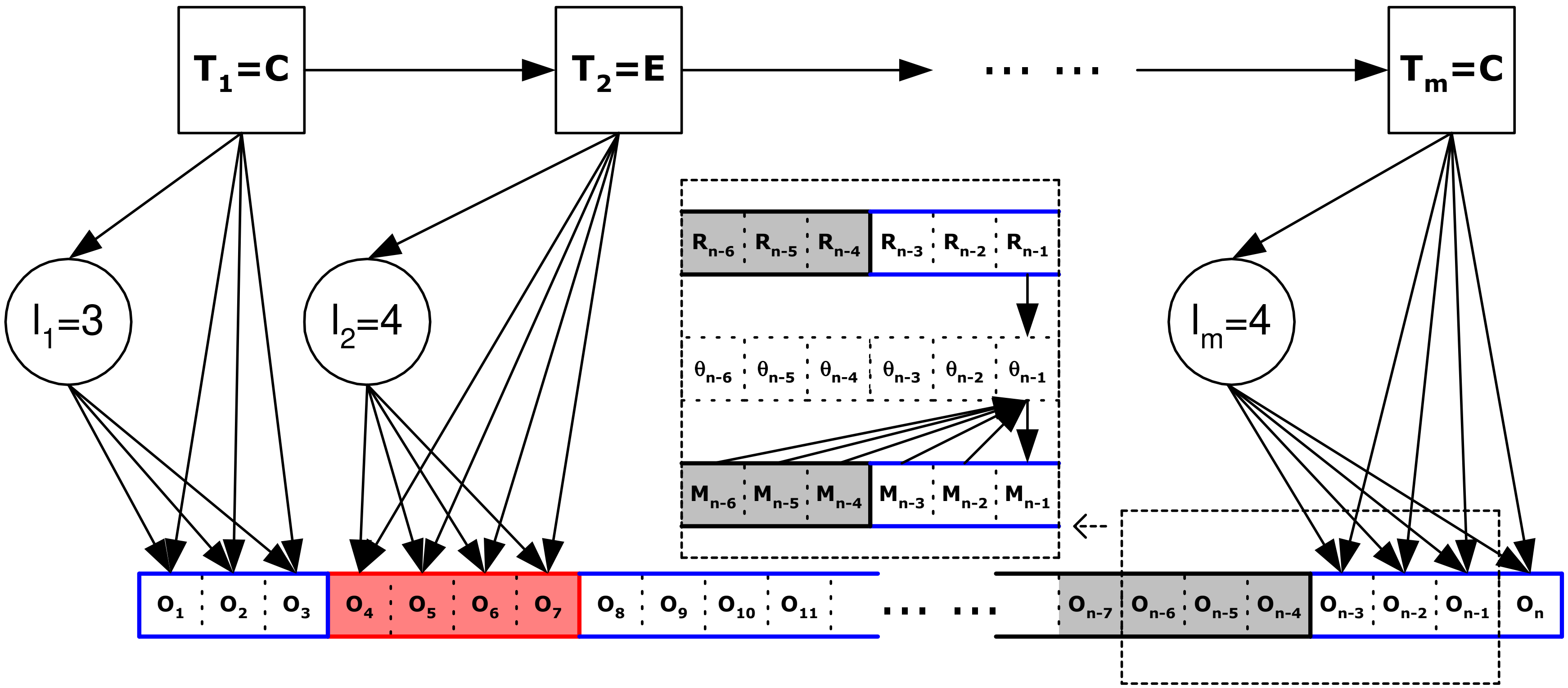
- 3D-profile or threading approach;
- ab-intio approaches;
  - LINUS (1995) Srinivasan and Rose. Local Independently Nucleated Units of Structure;
  - ROSETTA (1997) Baker. Monte Carlo simulated annealing procedure;
  - FRAGFOLD (2001) Jones. Simulated annealing of supersecondary structural fragments;

Multiple Sequence Alignment Profiles

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	⋯	⋯
			T_1=C			e_1=6			T_2=E			e_2=12		T_3		e_3			T_4=E			e_4	T_5	e_5			T_6=H				e_6=31	T_7		
Residue	M	Y	G	N	W	G	R	F	I	R	V	N	L	S	T	G	D	I	K	V	E	E	Y	D	E	E	L	A	K	K	W	L	⋯	⋯
Alignment 1	M	Y	G	N	W	G	R	F	I	R	V	N	L	S	T	G	D	I	K	V	E	E	Y	D	E	E	L	A	K	K	W	L	⋯	⋯
Alignment 2	A	N	G	W	T	G	N	I	L	R	V	N	L	T	T	G	N	I	T	L	E	D	S	S	K	.	.	F	K	S	F	V	⋯	⋯
Alignment 3	.	.	.	M	R	D	V	V	L	V	I	D	L	N	N	.	Y	T	Y	E	V	E	H	R	P	E	L	F	E	K	W	L	⋯	⋯
Alignment 4	M	Y	G	Y	C	G	K	I	L	R	V	D	L	S	E	K	T	V	S	T	L	I	P	E	E	K	D	L	R	E	F	I	⋯	⋯
Alignment 5	M	K	G	W	W	G	K	I	L	R	V	D	L	T	N	N	K	V	W	V	Q	E	Y	S	P	E	V	A	K	N	F	I	⋯	⋯
Alignment 6	M	M	G	F	H	G	K	I	L	E	V	N	L	S	N	S	T	V	S	E	I	E	V	T	E	E	M	I	K	K	F	V	⋯	⋯
Alignment 7	A	G	G	Y	M	G	K	I	L	K	V	N	L	S	E	G	S	I	K	E	L	P	I	D	E	E	I	A	R	K	Y	L	⋯	⋯

Appearance counts for each amino acid down a column in the profile is denoted as $M = [M_1, M_2, \ldots, M_n]$, where M_i is a 20×1 counting vector.

Likelihood Evaluation



Evaluation of $\mathcal{P}(M_k|R_k, O_{[1:k-1]}, T_i)$

In the multinomial distribution defined by θ_k, we observed M_k,

- multinomial probability: $\mathcal{P}(M_k|\theta_k, T_i) = \frac{(\sum_a M_k^a)!}{\prod_a M_k^a!} \prod_a (\theta_k^a)^{M_k^a}$
- a Dirichlet prior on θ_k with positive weight parameters to capture the dependency:

$$\mathcal{P}(\theta_k|R_k, O_{[1:k-1]}, T_i) = \frac{\Gamma(\sum_a \gamma_k^a)}{\prod_a \Gamma(\gamma_k^a)} \prod_a (\theta_k^a)^{\gamma_k^a - 1}$$

with $\gamma_k = R_k + W^0 + \sum_{j=1}^{\ell_k} W^j \cdot M_{k-j} + \sum_{j=\ell_k+1}^{\ell} W_A^j \cdot M_{k-j}$, the Gamma function $\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t)\, dt$, and $\ell_k = \min(k - e_{i-1} - 1, \ell)$.

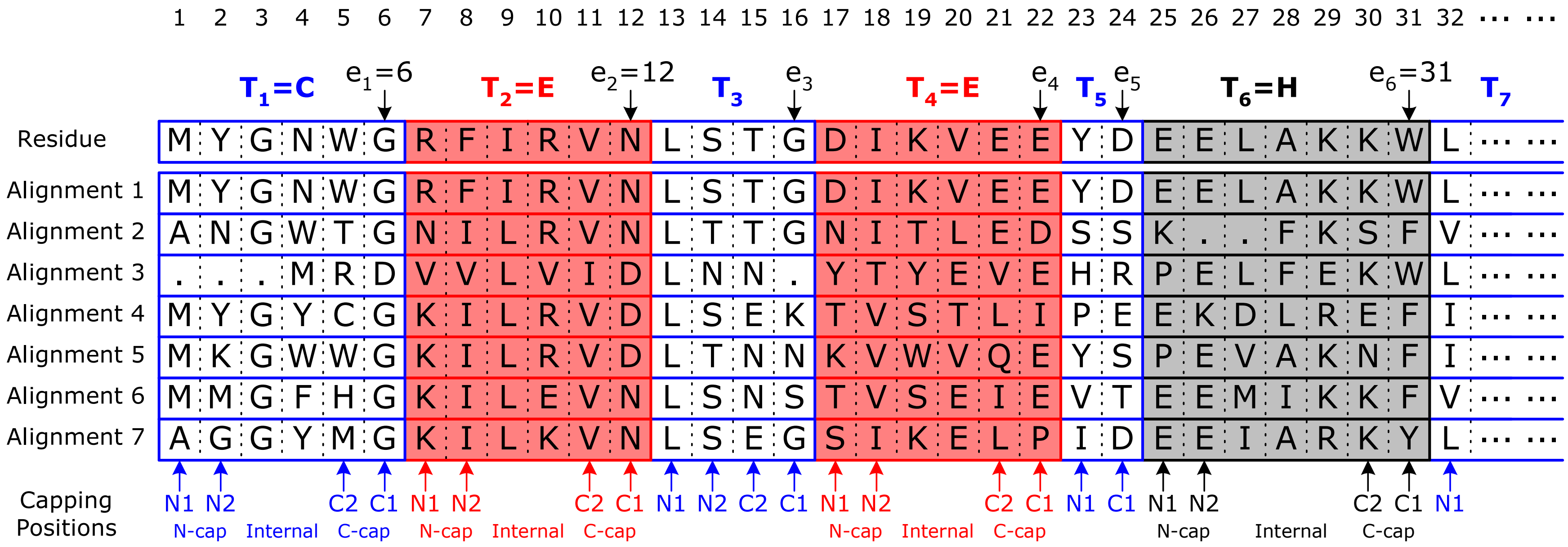
Evaluation of $\mathcal{P}(M_k|R_k, O_{[1:k-1]}, T_i)$

Since θ_k is a vector of latent variables, $\mathcal{P}(M_k|R_k, O_{[1:k-1]}, T_i)$ can be obtained as an integral over the space of θ_k, i.e.

$$
\begin{aligned}
\mathcal{P}(M_k|R_k, O_{[1:k-1]}, T_i) &= \int_{\theta_k} \mathcal{P}(M_k|\theta_k, \tau)\mathcal{P}(\theta_k|R_k, O_{[1:k-1]}, T_i)\, d\theta_k \\
&= \frac{\Gamma(\sum_a \gamma_k^a) \cdot \prod_a \Gamma(\gamma_k^a + M_k^a)}{\Gamma\left(\sum_a(\gamma_k^a + M_k^a)\right) \cdot \prod_a \Gamma(\gamma_k^a)} \cdot \frac{(\sum_a M_k^a)!}{\prod_a M_k^a!}
\end{aligned}
$$

where $\gamma_k = R_k + W^0 + \sum_{j=1}^{\ell_k} W^j \cdot M_{k-j} + \sum_{j=\ell_k+1}^{\ell} W_A^j \cdot M_{k-j}$ with positive weights.

Capping Positions in Segments



The capping position of each residue within a segment can be determined uniquely, and the distributions can be estimated from the training dataset.

Expanded Prior Probability

A set of interaction variables, $\mathcal{I} = \left\{\{S_j, S_{j'}, d_{jj'}, a_{jj'}\}_{j=1}^{i}\right\}$, is introduced to describe one possible case of the β-sheets.

$$\mathcal{P}(m, e, T, \mathcal{I}) = \mathcal{P}(\mathcal{I}|m, e, T) \cdot \mathcal{P}(m, e, T)$$

$\mathcal{P}(\mathcal{I}|m, e, T)$ is the distribution over β-sheet space. The β-sheet space is the set of all the possible β-sheets defined by $\mathcal{I}$.

$$\mathcal{P}(\mathcal{I}|m, e, T) = \mathcal{P}(i|k)\mathcal{P}(\{S_j, S_{j'}\}_{j=1}^{i}) \prod_{j=1}^{i} \mathcal{P}(d_{jj'}|S_j, S_{j'})\mathcal{P}(a_{jj'}|S_j, S_{j'}, d_{jj'})$$

where i is the number of interacting pairs, k is the number of β-strands, and $\{S_j, S_{j'}\}_{j=1}^{i}$ denotes a set of i interacting pairs of the β-strands.

Distribution in β-sheet Space

The probability in the β-sheet space:

$$\mathcal{P}(\mathcal{I}|m,e,T) = \mathcal{P}(i|k)\mathcal{P}(\{S_j, S_{j'}\}_{j=1}^{i}) \prod_{j=1}^{i} \mathcal{P}(d_{jj'}|S_j, S_{j'})\mathcal{P}(a_{jj'}|S_j, S_{j'}, d_{jj'})$$

- the sum over the β-sheet space: $\sum_{\mathcal{I}} \mathcal{P}(\mathcal{I}|m,e,T) = 1$;
- $\mathcal{P}(\{S_j, S_{j'}\}_{j=1}^{i}) \propto 1$ if the set of β-strand pairs is valid, otherwise zero;
- $\mathcal{P}(i|k)$, $\mathcal{P}(d_{jj'}|S_j, S_{j'})$ and $\mathcal{P}(a_{jj'}|S_j, S_{j'}, d_{jj'})$ are discrete distributions.

Joint Segmental Likelihood

A 20×20 weight matrix W_E to capture the correlation between distal interacting pairs.

$$\mathcal{P}(S_i|E, S_{-i}, \mathcal{I}) = \prod_{k=e_{i-1}+1}^{e_i} \frac{\Gamma(\sum_a \tilde{\gamma}_k^a) \cdot \prod_a \Gamma(\tilde{\gamma}_k^a + M_k^a)}{\Gamma\left(\sum_a (\tilde{\gamma}_k^a + M_k^a)\right) \cdot \prod_a \Gamma(\tilde{\gamma}_k^a)} \cdot \frac{(\sum_a M_k^a)!}{\prod_a M_k^a!} \cdot \mathcal{P}_c(R_k|E)$$

$$\tilde{\gamma}_k = R_k + W^0 + \sum_{j=1}^{\ell_k} W^j \cdot M_{k-j} + \sum_{j=\ell_k+1}^{\ell} W_A^j \cdot M_{k-j} + \sum_{\{k^*\}} W_E \cdot M_{k^*}$$

with $\{k^*\}$ denotes the set of interacting residues of R_k that can be determined by $\mathcal{I}$.

Sampling Scheme for Prediction

$$\mathcal{P}(m,e,T|O) \propto \mathcal{P}(m,e,T,O) = \sum_{\mathcal{I}} \mathcal{P}(m,e,T,\mathcal{I},O)$$
$$= \mathcal{P}(m,e,T) \prod_{S_i \notin \mathcal{I}} \mathcal{P}(S_i|T_i,S_{-i}) \cdot \sum_{\mathcal{I}} \mathcal{P}(\mathcal{I}|m,e,T) \prod_{S_i \in \mathcal{I}} \mathcal{P}(S_i|T_i,S_{-i},\mathcal{I})$$

We use the Metropolis-Hasting scheme with reversible-jumps (Green, 1995) in sampling. The acceptance probability for a new proposal (m^*,e^*,T^*) is

$$\rho\Big((m,e,T),(m^*,e^*,T^*)\Big) = \frac{\mathcal{P}(m^*,e^*,T^*,O)}{\mathcal{P}(m,e,T,O)} \times \frac{\mathcal{P}(m,e,T \leftarrow m^*,e^*,T^*)}{\mathcal{P}(m^*,e^*,T^* \leftarrow m,e,T)}$$

Enhanced Framework

- Prior: $\mathcal{P}(m,e,T,\mathcal{I}) = \mathcal{P}(\mathcal{I}|m,e,T)\mathcal{P}(m,e,T)$, where $\mathcal{P}(\mathcal{I}|m,e,T)$ is the distribution over β-sheet space;

- Likelihood $\mathcal{P}(O|m,e,T,\mathcal{I})$:
  - extend the dependency window to include the long-range interacting residues
  - introduce parameters to capture the contributions from hydrogen bonds

- Posterior: $\mathcal{P}(m,e,T,\mathcal{I}|O) \propto \mathcal{P}(O|m,e,T,\mathcal{I})\mathcal{P}(m,e,T,\mathcal{I})$;

Sampling Scheme for Prediction

We use Metropolis-Hasting scheme with reversible-jumps (Green, 1995) to collect samples in $\mathcal{P}(m, e, T|O)$.

- Segment split: $(m, e, T) \to (m + 1, e^*, T^*)$;
- Segment merge: $(m, e, T) \to (m - 1, e^*, T^*)$;
- Type change: $(m, e, T) \to (m, e, T^*)$;
- Endpoint change: $(m, e, T) \to (m, e^*, T)$.

Inference on Contact Maps

A contact map refers to a symmetrical, square, boolean matrix for the representation of pairwise, inter-residue contacts (1 if the distance less than 7Å, otherwise 0). An interaction set $\mathcal{I}$ is associated with a β-sheet contact map defined by a $n \times n$ matrix $\mathcal{C}$ whose ij-th entry $\mathcal{C}^{ij}$ defined as

$$\mathcal{C}^{ij}(\mathcal{I}) = \begin{cases} 1 & \text{if } R_i \text{ and } R_j \text{ are paired in the interaction set } \mathcal{I}; \\ 0 & \text{otherwise} \end{cases}$$

Our interest is the marginal $\mathcal{P}(\mathcal{C}^{ij} = 1) = \sum_{\{m,e,T,\mathcal{I}\}} \mathcal{C}^{ij}(\mathcal{I})\, \mathcal{P}(m, e, T, \mathcal{I}|O)$

Inference on Contact Maps

Using the samples we have collected in the distributions $\mathcal{P}(m,e,T|O)$ and $\mathcal{P}(\mathcal{I}|m,e,T)$, $\mathcal{P}(\mathcal{C}^{ij}=1)$ can be estimated by

$$\mathcal{P}(\mathcal{C}^{ij}=1) = \sum_{\{m,e,T\}} \sum_{\{\mathcal{I}\}} \mathcal{C}^{ij}(\mathcal{I})\, \mathcal{P}(\mathcal{I}|m,e,T,O)\, \mathcal{P}(m,e,T|O)$$

$$= \sum_{\{m,e,T\}} \sum_{\{\mathcal{I}\}} \mathcal{C}^{ij}(\mathcal{I}) \frac{\mathcal{P}(O|m,e,T,\mathcal{I})\mathcal{P}(\mathcal{I}|m,e,T)}{\sum_{\{\mathcal{I}\}} \mathcal{P}(O|m,e,T,\mathcal{I})\mathcal{P}(\mathcal{I}|m,e,T)} \mathcal{P}(m,e,T|O)$$

where the samples $\{\mathcal{I}\}$ are collected from $\mathcal{P}(\mathcal{I}|m,e,T)$, and $\{m,e,T\}$ are from $\mathcal{P}(m,e,T|O)$.

Clustering Protein Sequences and Structures

- pairwise sequence alignment scores and hierarchical clustering (single linkage)
  - BLASTCLUST (ncbi)
  - PROTOMAP (1999) Yona, Linial and Linial
  - SYSTERS (1998) Krause and Vingron
  - GENERAGE (2000) Enright and Ouzounis

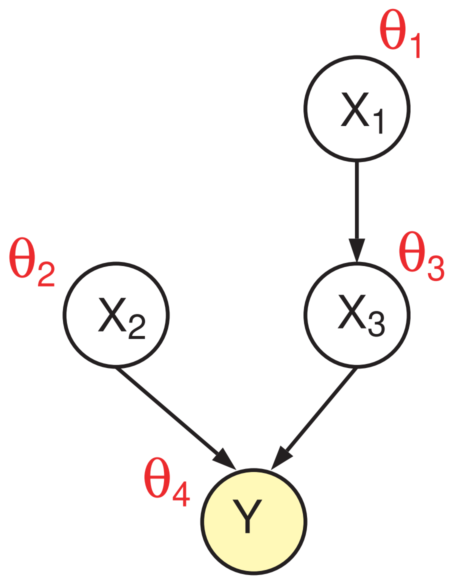
- All require settng threshold to distinguish cluster members from non-members
  - Krogh et al. (1994) Mixture of HMMs - requires number of clusters to be specified

Results - Globin Sequences by Krogh et al

The mixture of HMMs method of Krogh et al:

1. **Class 1** 233 sequences: principally all α, a few ζ (an α-type chain of mammalian embryonic hemoglobin), π/π' (the counterpart of the α chain in major early embryonic hemoglobin P), and $\theta - 1$ chains (early erythrocyte α-like).
2. **Class 2** 232 sequences: almost all β, a few δ (β-like), ϵ (β-type found in early embryos), γ (comprises fetal hemoglobin F in combination with two α chains), ρ (major early embryonic β-type chain) and θ chains (embryonic β-type chain).
3. **Class 3** 71 myoglobins.
4. **Class 4** 58 sequences. The 13 highest scoring in this cluster were leghemoglobins. This class contained a variety of sequences including 3 non-globins in the original data set.
5. **Class 5** 19 sequences. Midge globins.
6. **Class 6** 8 sequences. Globins from agnatha (jawless fish).
7. **Class 7** 7 sequences. varied.

EM in Graphical Models: Learning with Hidden Variables



Assume a model parameterised by θ with observable variables $\mathbf{y}$ and hidden variables $\mathbf{x}$

Goal: maximise log likelihood of observables.

$$\mathcal{L}(\theta) = \ln P(\mathbf{y}|\theta) = \ln \int P(\mathbf{y}, \mathbf{x}|\theta)\, d\mathbf{x}$$

- **E-step**: Infer $P(\mathbf{x}|\mathbf{y}, \theta_{\text{old}})$. This requires solving the *inference* problem: finding "explanations", $\mathbf{x}$, for the data, $\mathbf{y}$ given the current model θ. This step can make use of **belief propagation** or **junction tree** inference algorithms.
  Intuition: The **E-step** "fills in" values for the hidden variables. With no hidden variables, the likelihood is a simpler function of the parameters.
- **M-step**: find θ_{new} using complete data learning.
  Intuition: The **M-step** for the parameters at each node can be computed independently, and depends only on the values of the variables at that node and its parents.