Learning Sequence Motif Models Using Gibbs Sampling

BMI/CS 776

www.biostat.wisc.edu/bmi776/

Spring 2019

Colin Dewey

colin.dewey@wisc.edu

Goals for Lecture

Key concepts:

- Markov Chain Monte Carlo (MCMC) and Gibbs sampling
  - CS 760 slides for background
- Gibbs sampling applied to the motif-finding task
- parameter tying
- incorporating prior knowledge using Dirichlets and Dirichlet mixtures

Gibbs Sampling: An Alternative to EM

- EM can get trapped in local maxima
- One approach to alleviate this limitation: try different (perhaps random) initial parameters
- Gibbs sampling exploits randomized search to a much greater degree
- Can view it as a stochastic analog of EM for this task
- In theory, Gibbs sampling is less susceptible to local maxima than EM
- [Lawrence et al., *Science* 1993]

Gibbs Sampling Approach

- In the EM approach we maintained a distribution $Z^{(t)}{}_i$ over the possible motif starting points for each sequence at iteration t
- In the Gibbs sampling approach, we'll maintain a **specific** starting point for each sequence a_i but we'll keep randomly resampling these

Gibbs Sampling Algorithm for Motif Finding

given: length parameter W, training set of sequences

choose random positions for a

do

pick a sequence X_i

estimate p given current motif positions a

(using all sequences but X_i) (predictive update step)

sample a new motif position a_i for X_i (sampling step)

until convergence

return: p, a

Markov Chain Monte Carlo (MCMC)

- Consider a Markov chain in which, on each time step, a grasshopper randomly chooses to stay in its current state, jump one state left or jump one state right.

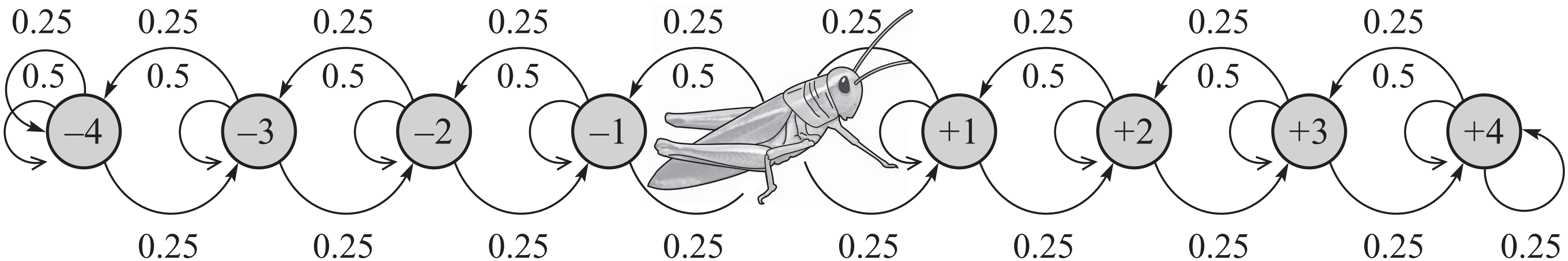


Figure from Koller & Friedman, Probabilistic Graphical Models, MIT Press

- Let $P^{(t)}(u)$ represent the probability of being in state u at time t in the random walk

$$P^{(0)}(0) = 1 \qquad P^{(0)}(+1) = 0 \qquad P^{(0)}(+2) = 0$$

$$P^{(1)}(0) = 0.5 \qquad P^{(1)}(+1) = 0.25 \qquad P^{(1)}(+2) = 0$$

$$P^{(2)}(0) = 0.375 \qquad P^{(2)}(+1) = 0.25 \qquad P^{(2)}(+2) = 0.0625$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$P^{(100)}(0) \approx 0.11 \qquad P^{(100)}(+1) \approx 0.11 \qquad P^{(100)}(+2) \approx 0.11$$

The Stationary Distribution

- Let $P(u)$ represent the probability of being in state u at any given time in a random walk on the chain

$$P^{(t)}(u) \approx P^{(t+1)}(u)$$ (for some sufficiently large *t*)

$$P^{(t+1)}(u) = \sum_v P^{(t)}(v)\tau(u \mid v)$$

probability of state v

probability of transition $v \rightarrow u$

- The stationary distribution is the set of such probabilities for all states

Markov Chain Monte Carlo (MCMC)

- We can view the motif finding approach in terms of a Markov chain
- Each state represents a configuration of the starting positions (a_i values for a set of random variables $A_1 \ldots A_n$)

- Transitions correspond to changing selected starting positions (and hence moving to a new state)

$A_1=5$	ACATCCG		ACATCCG	$A_1=3$
	CGACTAC		CGACTAC	
	ATTGAGC		ATTGAGC	
	CGTTGAC		CGTTGAC	
	GAGTGAT	$\xrightarrow{\quad\quad}$	GAGTGAT	
	TCGTTGG	$\tau(v \mid u)$	TCGTTGG	
	ACAGGAT		ACAGGAT	
	TAGCTAT		TAGCTAT	
	GCTACCG		GCTACCG	
	GGCCTCA		GGCCTCA	
	state u		state v	

Sampling with MCMC

- Suppose we have a probability distribution $P(\boldsymbol{X})$ for which we would like to
  - find the mode: $\mathrm{argmax}_{\boldsymbol{x}}\, P(\boldsymbol{x})$
  - sample from
- But it may be intractable to do either directly
- **Key idea**: construct a Markov chain with
  - states corresponding to configurations of $\boldsymbol{X}$
  - stationary distribution equal to $P(\boldsymbol{X})$
- Running MCMC with such a Markov chain allows us to address both tasks
  - even when the number of configurations is generally quite large!

Markov Chain Monte Carlo

- How do we construct a Markov chain with a stationary distribution equal to our probability distribution, *P,* of interest?
- Set the transition probabilities such that the condition of **detailed balance** holds for all pairs of states, *u and v*:

$$P(u)\tau(v \mid u) = P(v)\tau(u \mid v)$$

probability of state u

probability of transition $u \rightarrow v$

- When detailed balance holds, if we perform MCMC with N samples and *count(u)* is the number of times we are in state *u,* then:

$$\frac{1}{N} \lim_{N \rightarrow \infty} count(u) = P(u)$$

MCMC with Gibbs Sampling

Gibbs sampling is a special case of MCMC in which

- Markov chain transitions involve changing one variable at a time

- Transition probability is **conditional probability** of the changed variable given all others

- We sample the joint distribution of a set of random variables $P(A_1 \ldots A_n)$ by iteratively sampling from $P(A_i \mid A_1 \ldots A_{i-1}, A_{i+1} \ldots A_n)$

Gibbs Sampling Approach

- Possible state transitions when first sequence is selected

ACATCCG
CGACTAC
ATTGAGC
CGTTGAC
GAGTGAT
TCGTTGG
ACAGGAT
TAGCTAT
GCTACCG
GGCCTCA

ACATCCG
CGACTAC
ATTGAGC
CGTTGAC
GAGTGAT
TCGTTGG
ACAGGAT
TAGCTAT
GCTACCG
GGCCTCA

ACATCCG
CGACTAC
ATTGAGC
CGTTGAC
GAGTGAT
TCGTTGG
ACAGGAT
TAGCTAT
GCTACCG
GGCCTCA

ACATCCG
CGACTAC
ATTGAGC
CGTTGAC
GAGTGAT
TCGTTGG
ACAGGAT
TAGCTAT
GCTACCG
GGCCTCA

ACATCCG
CGACTAC
ATTGAGC
CGTTGAC
GAGTGAT
TCGTTGG
ACAGGAT
TAGCTAT
GCTACCG
GGCCTCA

Gibbs Sampling Approach

- The probability of a state is given by

$$P(u) \propto \prod_{c} \prod_{j=1}^{W} \left(\frac{p_{c,j}}{p_{c,0}} \right)^{n_{c,j}(u)}$$

count of
c in motif position j

background probability
for character c

probability of
c in motif position j

u

ACATCCG
CGACTAC
ATTGAGC
CGTTGAC
GAGTGAT
TCGTTGG
ACAGGAT
TAGCTAT
GCTACCG
GGCCTCA

$n(u)$

	1	2	3
A	1	3	1
C	5	2	1
G	2	2	6
T	2	3	2

See Liu et al., *JASA*, 1995
for the full derivation

Gibbs Sampling Approach

- How do we get the transition probabilities when we don't know what the motif looks like?

Sampling New Motif Positions

- For sampling a new motif position in sequence i
- Estimate p from all sequences except sequence i
- For each possible starting position, $A_i = j$, compute the likelihood ratio

$$LR(j) = \frac{\prod_{k=j}^{j+W-1} p_{c_k,\, k-j+1}}{\prod_{k=j}^{j+W-1} p_{c_k,\, 0}}$$

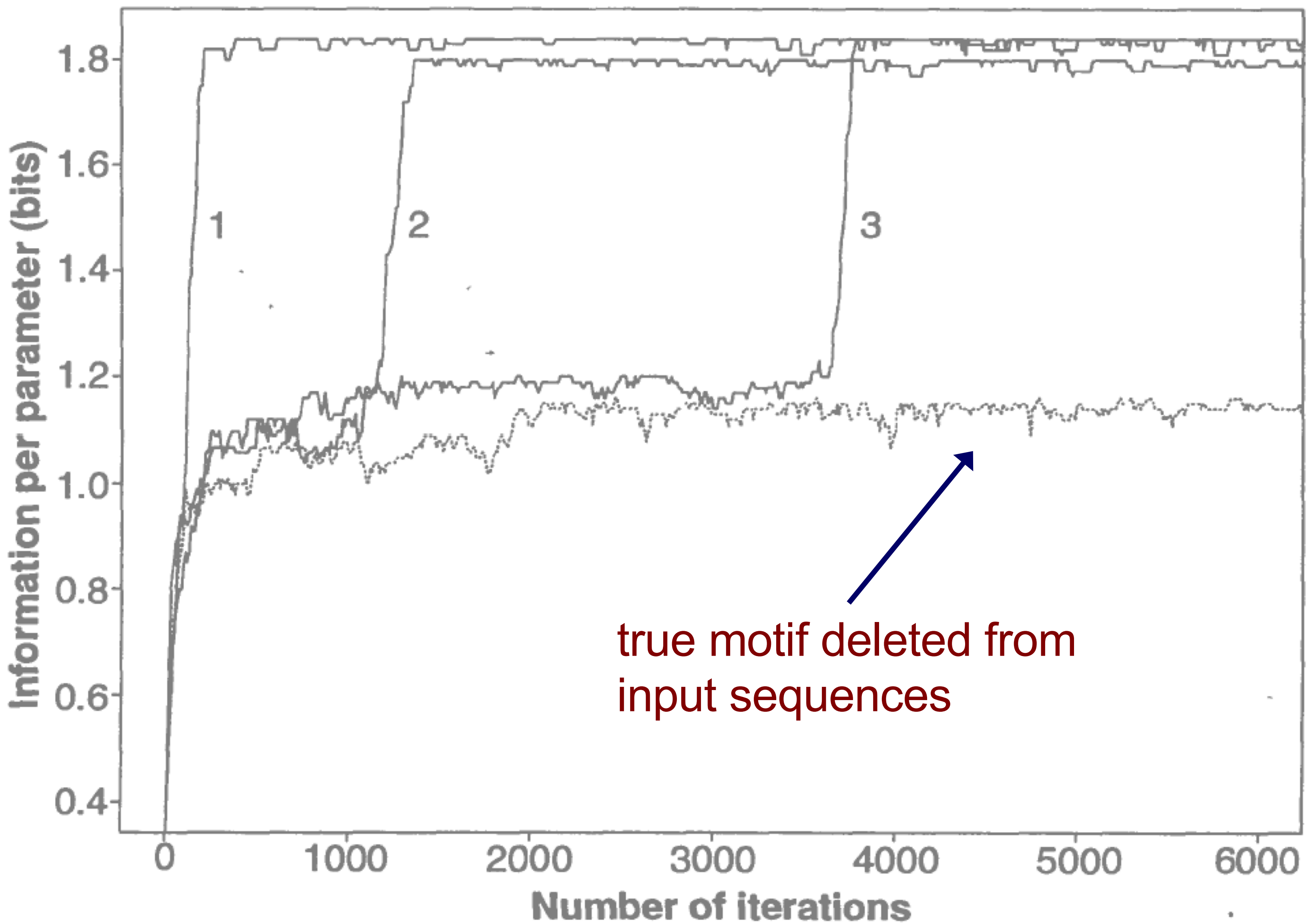
- Randomly select a new starting position $A_i = j$ with probability

$$\frac{LR(j)}{\sum_{k \in \{\text{starting positions}\}} LR(k)}$$

The Phase Shift Problem

- Gibbs sampler can get stuck in a local maximum that corresponds to the correct solution shifted by a few bases
- Solution: add a special step to shift the a values by the same amount for all sequences
- Try different shift amounts and pick one in proportion to its probability score

Convergence of Gibbs



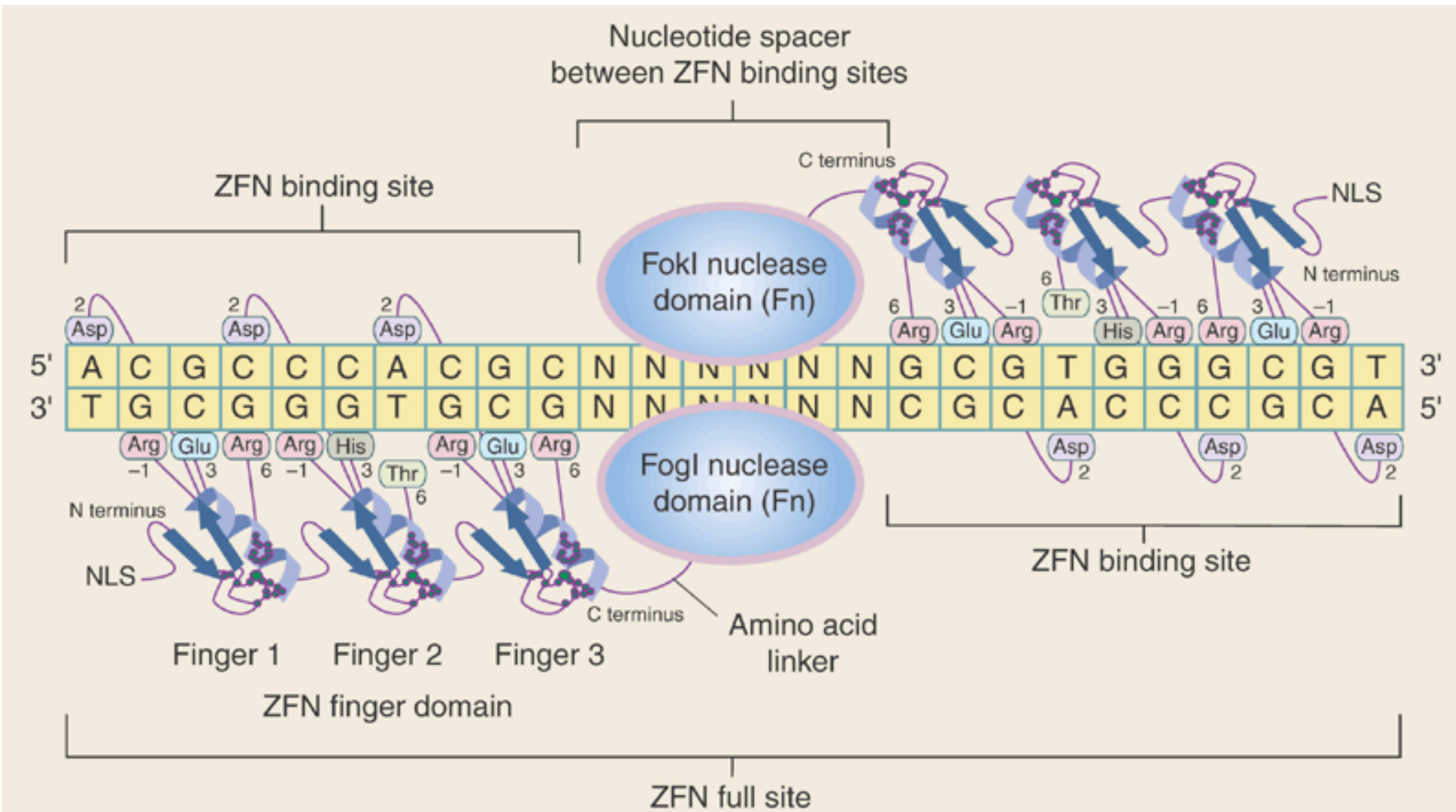
Using Background Knowledge to Bias the Parameters

Let's consider two ways in which background knowledge can be exploited in motif finding

1. Accounting for palindromes that are common in DNA binding sites
2. Using Dirichlet mixture priors to account for biochemical similarity of amino acids

Using Background Knowledge to Bias the Parameters

- Many DNA motifs have a palindromic pattern because they are bound by a protein *homodimer*: a complex consisting of two identical proteins



Porteus & Carroll *Nature Biotechnology* 2005

Representing Palindromes

- Parameters in probabilistic models can be "tied" or "shared"

$$\begin{bmatrix} p_{a,0} & p_{a,1} & \cdots & p_{a,W} \\ p_{c,0} & p_{c,1} & \cdots & p_{c,W} \\ p_{g,0} & p_{g,1} & \cdots & p_{g,W} \\ p_{t,0} & p_{t,1} & \cdots & p_{t,W} \end{bmatrix}$$

- During motif search, try tying parameters according to palindromic constraint; accept if it increases likelihood ratio test (half as many parameters)

Updating Tied Parameters

$$\begin{bmatrix} p_{a,0} & p_{a,1} & \cdots & p_{a,W} \\ p_{c,0} & p_{c,1} & \cdots & p_{c,W} \\ p_{g,0} & p_{g,1} & \cdots & p_{g,W} \\ p_{t,0} & p_{t,1} & \cdots & p_{t,W} \end{bmatrix}$$

$$p_{a,1} \equiv p_{t,W} = \frac{n_{a,1} + n_{t,W} + d_{a,1} + d_{t,W}}{\sum_b (n_{b,1} + d_{b,1}) + \sum_b (n_{b,W} + d_{b,W})}$$

Including Prior Knowledge

- Recall that MEME and Gibbs update parameters by:

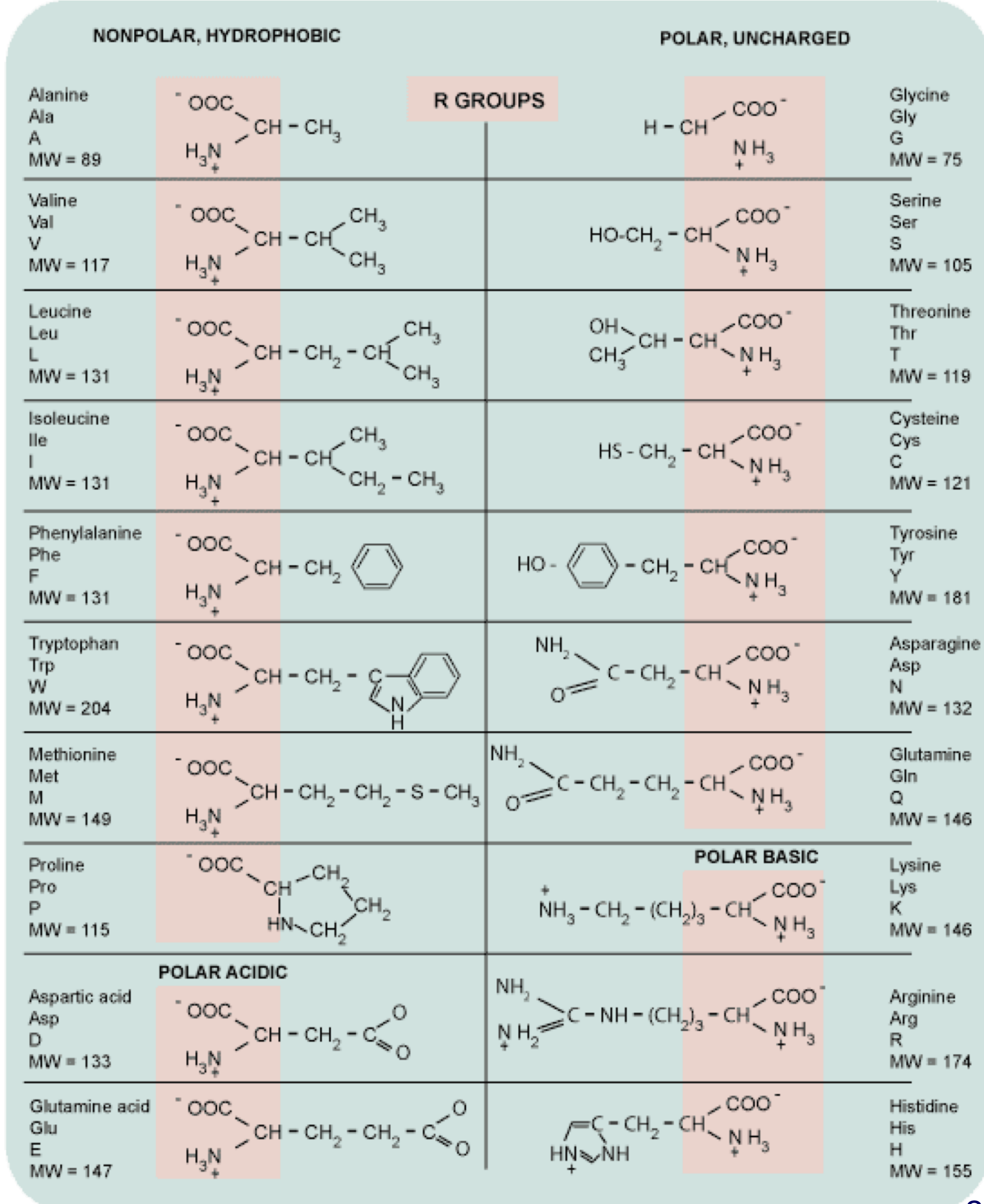
$$p_{c,k} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$

- Can we use background knowledge to guide our choice of pseudocounts ($d_{c,k}$)?

- Suppose we're modeling protein sequences…

Amino Acids

- Can we encode prior knowledge about amino acid properties into the motif finding process?
- There are classes of amino acids that share similar properties



Using Dirichlet Mixture Priors

- Prior for a single PWM column, not the entire motif
- Because we're estimating multinomial distributions (frequencies of amino acids at each motif position), a natural way to encode prior knowledge is using Dirichlet distributions
- Let's consider
  - the Beta distribution
  - the Dirichlet distribution
  - mixtures of Dirichlets

The Beta Distribution

- Suppose we're taking a Bayesian approach to estimating the parameter θ of a weighted coin
- The Beta distribution provides an appropriate prior

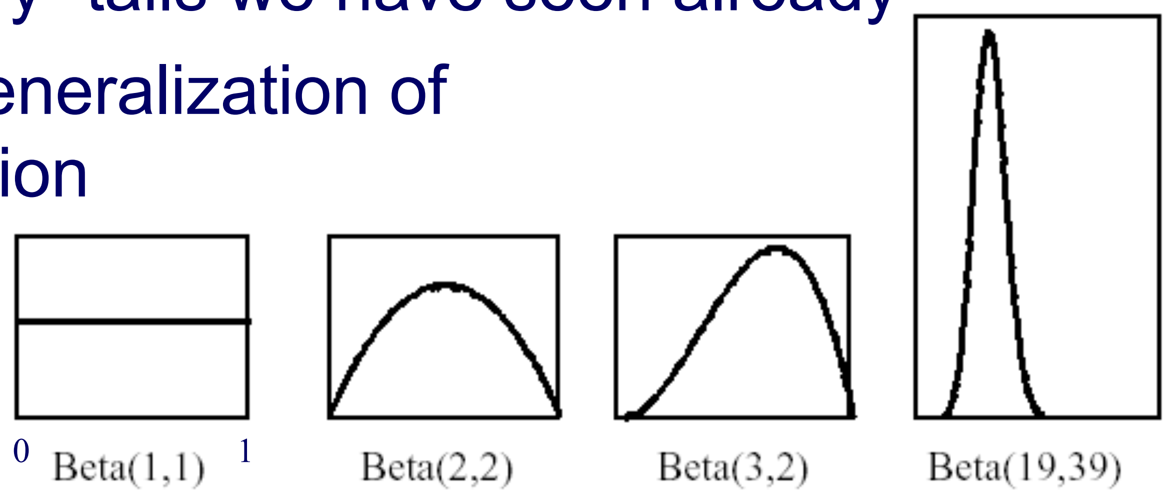
$$P(\theta) = \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1}$$

where

α_h # of "imaginary" heads we have seen already

α_t # of "imaginary" tails we have seen already

Γ continuous generalization of factorial function



The Beta Distribution

- Suppose now we're given a data set D in which we observe D_h heads and D_t tails

$$P(\theta \mid D) = \frac{\Gamma(\alpha + D_h + D_t)}{\Gamma(\alpha_h + D_h)\Gamma(\alpha_t + D_t)} \theta^{\alpha_h + D_h - 1}(1-\theta)^{\alpha_t + D_t - 1}$$

$$= \mathrm{Beta}(\alpha_h + D_h, \alpha_t + D_t)$$

- The posterior distribution is also Beta: we say that the set of Beta distributions is a *conjugate* family for binomial sampling

The Dirichlet Distribution

- For discrete variables with more than two possible values, we can use *Dirichlet* priors

- Dirichlet priors are a *conjugate* family for multinomial data

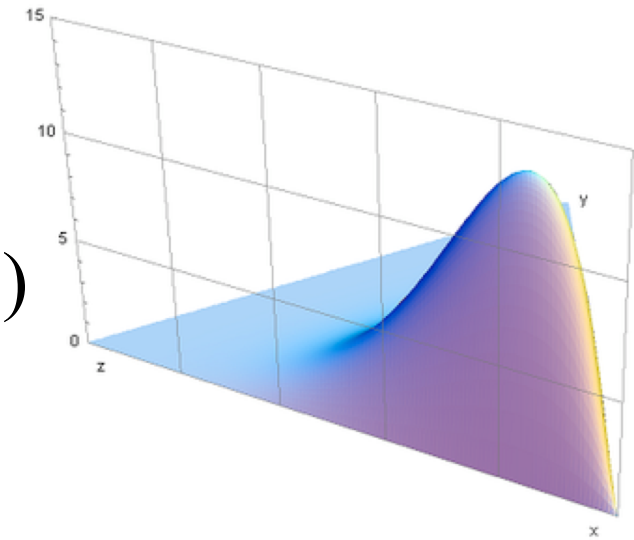
$$P(\theta) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

- If $P(\theta)$ is Dirichlet$(\alpha_1, \ldots, \alpha_K)$, then $P(\theta|D)$ is Dirichlet$(\alpha_1+D_1, \ldots, \alpha_K+D_K)$, where D_i is the # occurrences of the i^{th} value

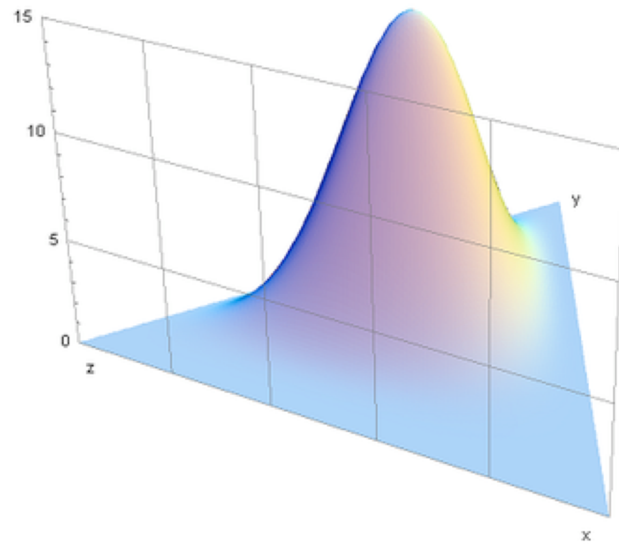
Dirichlet Distributions

Probability density (shown on a simplex) of Dirichlet distributions for K=3 and various parameter vectors α

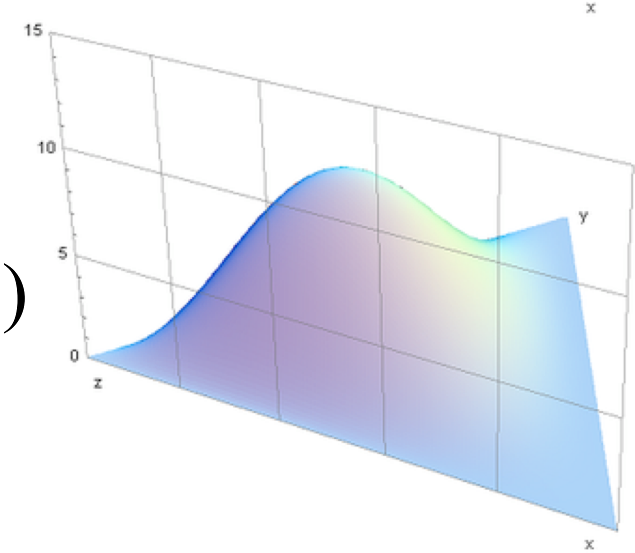
$\alpha = (6, 2, 2)$



$\alpha = (3, 7, 5)$



$\alpha = (2, 3, 4)$



$\alpha = (6, 2, 6)$

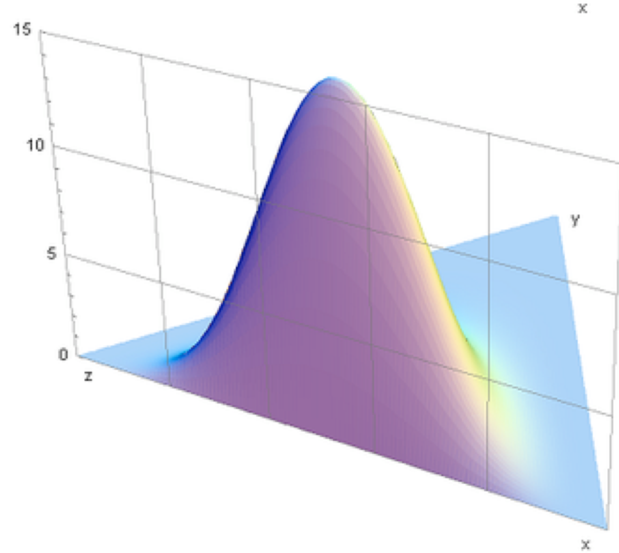


Image from Wikipedia, Python code adapted from Thomas Boggs

Mixture of Dirichlets

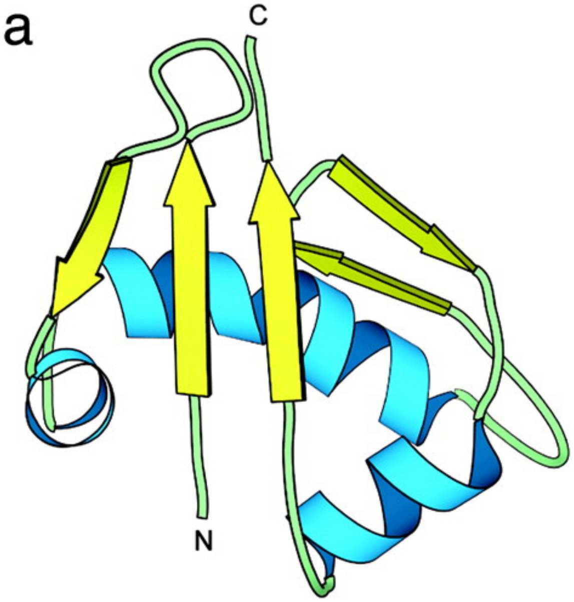
- We'd like to have Dirichlet distributions characterizing amino acids that tend to be used in certain "roles"

- Brown et al. [*ISMB* '93] induced a set of Dirichlets from "trusted" protein alignments
  - "large, charged and polar"
  - "polar and mostly negatively charged"
  - "hydrophobic, uncharged, nonpolar"
  - etc.

Trusted Protein Alignments

- A trusted protein alignment is one in which known protein structures are used to determine which parts of the given set of sequences should be aligned

a



c

```
(a) 2580558  Hs  886 HLSLIVRFPNQGRQVDELDIWSHTNDTIGSVRRCIVNRIKA-N 927
    6678523  Mm  885 HLSFIVRFPNQGRQVDDLEVWSHTNDTIGSVRRCILNRIKA-N 926
    22507351 Mm  885 HLSFTVRFPNQGKEVEDLDILSHTNATIGSVRRCILNRMNV-N 926
    31235452 Ag  835 QVELIVKFQTPGRQLDDIELLSHSNETMHSFKRNLLRRIKVLK 877
    24651755 Dm  979 NTILYIRFQNPGRSIDDMEIVTHSNETMAAFKRNLLKRIKGTS 1021
```

Using Dirichlet *Mixture* Priors

- Recall that the EM/Gibbs update the parameters by:

$$p_{c,k} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$

- We can set the pseudocounts using a *mixture* of Dirichlets:

$$d_{c,k} = \sum_j P(\alpha^{(j)} \mid \mathbf{n}_k)\alpha_c^{(j)}$$

- where $\alpha^{(j)}$ is the j^{th} Dirichlet component

Using Dirichlet Mixture Priors

$$d_{c,k} = \sum_j P(\alpha^{(j)} \mid \mathbf{n}_k)\, \alpha_c^{(j)}$$

probability of j^{th} Dirichlet given observed counts

parameter for character c in j^{th} Dirichlet

- We don't have to know which Dirichlet to pick
- Instead, we'll hedge our bets, using the observed counts to decide how much to weight each Dirichlet

See textbook section 11.5

Motif Finding: EM and Gibbs

- These methods compute *local*, *multiple* alignments
- Optimize the likelihood or likelihood ratio of the sequences
- EM converges to a local maximum
- Gibbs will “converge” to a global maximum, *in the limit;* in a reasonable amount of time, probably not
- Can take advantage of background knowledge by
  - tying parameters
  - Dirichlet priors

- There are *many* other methods for motif finding
- In practice, motif finders often fail
  - motif “signal” may be weak
  - large search space, many local minima
  - do not consider binding context