

## Computational gene finding

- Gene finding in prokaryotes
- Gene finding in eukaryotes
  - Ab initio
  - Comparative

(c) Devika Subramanian, 2007

17

## Finding genes in prokaryotes

- Prokaryotes are single-celled organisms without a nucleus (e.g., bacteria).
- Few introns in prokaryotic cells. Over 70% of *H. influenzae* genome codes for proteins.
- No introns in coding region.

gene1                      gene2                      gene3

(c) Devika Subramanian, 2007

18

## Finding genes in prokaryotes

- Main idea: if bases were drawn uniformly at random, then a stop codon is expected once every 64/3 (about 21) bases. Since coding regions are terminated by stop codons, a simple technique to find genes is to look for long stretches of bases without a stop codon. Once a stop codon is found, we work backward to find the start codon corresponding to the gene.
- Main problems: misses short genes, overlapping ORFs.

(c) Devika Subramanian, 2007

19

## Segment of Influenza Virus

- <http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?db=nucleotide&val=CY018024>
- 1151 bp segment of Influenza B Virus.
- Has two genes: 4 to 750 and 750 to 1079

(c) Devika Subramanian, 2007

20

## The sequence

```
1 aaatg-cgc tgtttggaga caaatggc tactctgttt cattgcaga agatgggaga
61 gacaaagcag aactagcaga aaaaataaac tgttggttcg gtaggaadga atttgacctt
121 gactctgect tgaatggat aaaaacaaa agatgcttta ctgatataca aaagcaata
181 attggtgect ctatctgctt tttaaaccc aaagaccag aaggaagaag aagattcctc
241 acagagcctc tatcaggaat ggaacaaca gaaacaaa aagaagcctt gatttagct
301 gagagaaaa tgaagagatg tgtgagcttt catgaagcat tgaatagc agaagccat
361 gaaagctcag cgtactata ttgtctcatg gtcattgacc tgaatcctgg aatatttca
421 atgcaagtaa aactaggaac gctctgtgct ttgtgcgaga aacaagcctc acattcaaac
481 agggctata gcaagcagc gaaatttcaa gttcccgag tgaagcagga aatgcagatg
541 gtctcagcta tgaaccagc aaaaacaatg aatgnaatg gaaagagga agacttcaa
601 aagctgacg aagagctca aagaacattt ggaatttga gatctctgg acaagtcaa
661 aagaatggg aaggaattgc aaagatgta atggaatgc taaagcagag ctctatggga
721 aattcagctc ttgtgaaga atatactaa taa tg-tgaacc atttcagatt ctttcaattt
781 gttcttttat ctatcagct ctcaatttca tggcttggac aataggcat ttgaatcaa
841 taaaagag agtaaacatg aaaaacgaa taaaagctc aaacaagag acatanaaa
901 gagagatc atttttgaga cacagtacc aaaaagaat ccagccaaa gaacaatga
961 aggaatcct ctctgcaaac atggagatg tgaatgacca catagtatt gagggcctt
1021 ctgcccagga gataataaaa atggatgaaa cagttttgga gatagaaga ttgat-taa
1081 ttaacttttt tactgtattt ctattatgac atttaagcaa attgtaata atgtrcagaa
1141 ataaactgaa a
```

(c) Devika Subramanian, 2007

21

## GLIMMER

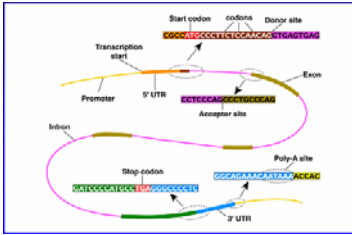
- State of the art prokaryotic gene finder. Based on interpolated Markov models.
- Available at <http://cbcb.umd.edu/software/glimmer>
- 98% accuracy in identifying viral and microbial genes. 2007 paper in Bioinformatics that shows latest version of tool.

(c) Devika Subramanian, 2007

22

# Computational gene finding

## Gene finding in eukaryotic DNA



(c) Devika Subramanian, 2007

# Structure of a human gene

Structure of a Human Gene (PSA)



Exon-intron structure

(c) Devika Subramanian, 2007

```
> HSKIIIE. Human gene for casein kinase II subunit beta (EC 2.7.1.37).
ggggctggaggtgtaaatlagaggagctggagagagcttccaggtttgggttgccttaagaaagggt
ggttccgaatttcocogtgggtggggcccaaatggggggagccacagaggcgggaaggga
gaaactggagcttactgaacagtctcttttcttagctgagctgctatgctcagagggaggttgc
ctggatttctgggttctgtgggctcoggtggcaatgaaftctttctcacttggccaggttccacatct
ctacttggccaggttccacatcttcccacacagcgttcttccacatctccacttctcactcgttctct
aaagcttctatggagagaggtgaggtagctggagagagaccacagctacttctcgtcagttggaggtg
agaaacaaggccacaacagatgcaagttgtttgatgaagaagglctggcccaaggttcaaa
agatgagggatattgatattgggttccctcttggcttccatgtctccacaggtttgcacactccca
ggacaacttgggtgagctcctaatgagcgggtcctctccactcctccatgagcttggctgtctctct
ggaccctgggtccctcagggttcttctgtgtgtgtcgtgcagacttcttctctctcaca
atctctattgggtgagctcctcctcctctccttctcctctcctctcctctctcctcctcctcctc
tcttcttcttggatccctcactccttctctctctccttcttctgacgtatctcacttctctcacttctt
tgcgaaggttccagaacccagtttctgctcctccatggccagagtttgaaagggttaaggccccaaatga
ggttggagcacaactgaaatagcccgctcagatggtggccagggcaactaggaaagctgaaaaaca
agtgttgcatttggccggcgtgtggagagagacaccccccaacagagtgagctg
atttgatgctttaa
ggagacttga
gggtgagctcctcactcctcctctctggcagtaagagacacaggttctctcga
gcacaccctggtttagagaagctagctgagaaaggagggaacccccagaacttgg
cccggccaaatttggagaagggcaacagagatttgaaccaccttagtccagagaaggggacct
ctggacagagttggaagaggtgccgacagagattcacaagaaggggtgctctctct
ttactctacttgcacccccctccaatgtatctcaggttgggtgaggaacttgg
ggttactgtctogtctgtagtctgtagtggaaacagagcctcctccctctctcctcctcctcctcctc
ggaaaagcccggttggacgtctatggaaagagttggggctcaactcttgaggg
ggaggttaggtaggaataggaggtaccctgacctgtgagctcctggctg
cccccgggtgaaagcctggttagagctcacttgcacaaactgaaatagatct
acacctcacacaggaaggcgctacttggcactgtctctctcctcctcctcctcctcctcctcctcctc
gtagccgcccagaagacctggcaaccagtttggctgggtgagggagagctcattgaaggttca
aaagaagggcccaagctccccagagagggggaggcctctcctggaggtctcctcctcctcctcctcctc
cagaataggggatctccctgtagtgaagtagtgggaaagtatttgattatctgtgctttagtctacct
tatgtagaattgttctgagctgagaagttggaaacccagagctttgacctgagcagctcctcattagag
gagctcaggttgggagaggtgggaatgcaaggttgaagctttagtggaggtcactgagctgctgctgctcct
ctgaccttgcctggcctaggctcaaggtttcaagaatggctctaccagctgcaagctcca
agcccgagcaactcagaagcccagctcaagagatcttgaatggctcctcctcctcctcctcctcctcctcct
tttggcttctccttcttcttggccacctttcaaggagaggtttttaggtttaaattaaagga
gtcgttatogtggtaggaaatgaaataagtagaagaaagggcattgagtagtctgctgggtctg
tgaaggggttgaagctggccatggaactcgggctccacggccccagggatgg
```

# Ab initio methods

- Use information embedded in the genomic sequence *exclusively* to predict the gene structure.
- Find structure *G* representing gene boundaries + internal gene structure which maximizes the probability  $P(G|genomic\ sequence)$ .
- Hidden Markov models are the predominant generative method for modeling the problem.

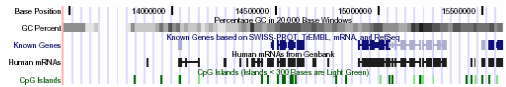
(c) Devika Subramanian, 2007

# Ab-initio methods

- Advantages
  - Intuitive, natural modeling
  - Prediction of 'novel' genes, *i.e.*, with no a priori known cDNA or protein evidence
- Caveats
  - Not effective in detecting alternatively spliced forms, interleaved or overlapping genes
  - Difficulties with gene boundary identification
  - Potentially large number of false positives with over-fitting

(c) Devika Subramanian, 2007

# A simple example: CpG Islands



CpG nucleotides in the genome are frequently methylated. (Write CpG not to confuse with CG base pair)



Methylation often suppressed around genes, promoters  $\rightarrow$  CpG islands

(c) Devika Subramanian, 2007

## Example: CpG Islands

In CpG islands,  
CG is more frequent than in the rest of the genome

(c) Devika Subramanian, 2007 29

## Two problems

- Given a short DNA sequence, does it come from a CpG island or not?
  - Is this part of a CpG island or not?
- How to find the CpG islands in a long sequence?

(c) Devika Subramanian, 2007 30

## Generative models

Models generate sequences of strings in the A,T,C,G alphabet. Model parameters are tuned to reflect characteristics of CpG and non CpG islands.

(c) Devika Subramanian, 2007 31

## Markov processes: a quick intro

- We are interested in predicting weather, which can be either be sunny (s) or rainy (r).
- The weather on a given day depends only on the weather on the previous day.

$$P(w_t | w_{t-1}, \dots, w_1) = P(w_t | w_{t-1})$$

This is the Markov property.

(c) Devika Subramanian, 2007 32

## Markov process example

- We have knowledge of the transition probabilities between sunny and rainy days.

	s	r
s	0.9	0.1
r	0.5	0.5

Rows of the transition matrix sum to 1.

- We know the initial probabilities of s and r.

(c) Devika Subramanian, 2007 33

## Generating weather sequences

- Let the probabilities of weather on the first day be [0.5 0.5]. Lets say we start with a sunny day.
- Now we consult our transition matrix and find that  $P(w_t|s) = [0.9 0.1]$ . It is more likely that the next day will be sunny too.
- We repeat this process, flipping coins biased by the probability  $P(w_t|w_{t-1})$  to get a sequence representing weather for a consecutive set of days.

(c) Devika Subramanian, 2007 34

## Generating sequences (Take 2)

sequence

s

s

r

$$\begin{matrix}
 & \begin{matrix} s & r \end{matrix} \\
 \begin{matrix} s \\ r \end{matrix} & \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}
 \end{matrix}$$

(c) Devika Subramanian, 2007 35

## Prediction

- Suppose day is rainy. We will represent this as a vector of probabilities over the two values.
 
$$\pi(1) = [0 \ 1];$$
- How do we predict weather on day 2 given  $\pi(1)$  and the transition probabilities  $P$ ?
- From  $P$ , we can see that the probability of day 2 being sunny is .5, and for being rainy is 0.5
 
$$\pi(1) * P = [0.5 \ 0.5];$$

(c) Devika Subramanian, 2007 36

## Probability of a sequence

- What is the probability of observing the sequence "rrrrrrs"?

$$\begin{aligned}
 P(X = rrrrrrs) &= \pi(r)P(r|r)P(r|r)P(r|r)P(r|r)P(r|r)P(s|r) \\
 &= \pi(r) \prod_{t=2,7} P(x_t | x_{t-1}) = (0.5)^7
 \end{aligned}$$

(c) Devika Subramanian, 2007 37

## Which weather pattern is more likely?

- Given a transition model
 
$$\begin{matrix}
 & \begin{matrix} s & r \end{matrix} \\
 \begin{matrix} s \\ r \end{matrix} & \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}
 \end{matrix}$$
- And an initial state distribution: [0.5 0.5]
- And two sequences: rrrrrrs and sssssr  
Which is more likely, given the model?

(c) Devika Subramanian, 2007 38

## Comparing likelihoods

$$\begin{aligned}
 P(X = rrrrrrs | Model) &= \pi(r)[P(r|r)]^5 P(s|r) = (0.5)^7 \\
 P(X = sssssr | Model) &= \pi(s)[P(s|s)]^5 P(r|s) = 0.5 * (0.9)^5 * 0.1
 \end{aligned}$$

(c) Devika Subramanian, 2007 39

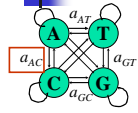
## Markov models (summary)

- States:  $S = \{s_1, \dots, s_N\}$ ,  $N$  states
- Transition probability:
  - $a_{ij} = P(X_{t+1}=s_j | X_t=s_i)$ ,  $i, j$  in  $[1..N]$
- Initial state probability
  - $\pi_i = P(X_1=s_i)$ ,  $i$  in  $[1..N]$

Model generates sequences of states from  $S$ , and we can compute how likely a sequence is given the model.

(c) Devika Subramanian, 2007 40

## Markov models for CpG islands



A state for each of the four letters A, C, G, and T in the DNA alphabet

$$a_{st}^+ = \frac{c_{st}^+}{\sum_{t'} c_{st'}^+}$$

From a set of known CpG islands, and non CpG islands, estimate the transition probabilities

+	A	C	G	T
A	.180	.274	.426	.120
C	.171	.368	.274	.188
G	.161	.339	.375	.125
T	.079	.355	.384	.182

-	A	C	G	T
A	.300	.205	.285	.210
C	.322	.298	.078	.302
G	.248	.246	.298	.208
T	.177	.239	.292	.292

(c) Devika Subramanian, 2007

41

## Using the model

- To use the model for classification of a given sequence, calculate the log-odds ratio.
- Is the sequence more likely to come from a CpG island or a non-CpG region?

$$P(x | CpG) > P(x | nonCpG)$$

$$\frac{P(x | CpG)}{P(x | nonCpG)} > 1$$

$$\log \frac{P(x | CpG)}{P(x | nonCpG)} > 0$$

Log-odds ratio

(c) Devika Subramanian, 2007

42

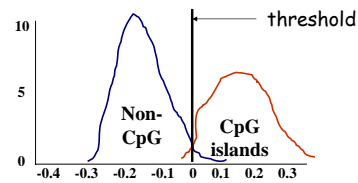
## The log-odds ratio

$$S(x) = \log \frac{P(x | CpG)}{P(x | nonCpG)} = \sum_{i=1}^L \log \frac{a_{x_{i-1}x_i}^+}{a_{x_{i-1}x_i}^-}$$

(c) Devika Subramanian, 2007

43

## Histogram of log-odds scores



Given a short sequence  $x$ , does it come from CpG island (Yes-No question)?

Decision rule: if  $S(x) > 0$  then CpG else non-CpG

(c) Devika Subramanian, 2007

44

## How to locate CpG islands?

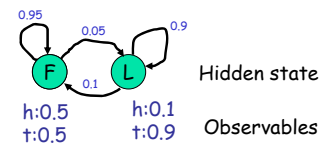
- Given a DNA sequence, find the CpG islands in it, if any.
- Approach: Calculate the log-odds score for a window of  $w$  nucleotides around every base in the sequence. Predict as CpG islands, those with a positive log-odds score.
- Problem: What should the size of the window  $w$  be? Predictions are sensitive to choice of  $w$ .

(c) Devika Subramanian, 2007

45

## The occasionally dishonest casino

- A casino uses a fair coin most of the time, but occasionally they switch to a loaded coin. You can't see which coin they are using, just the results of the flips (heads and tails) are visible.



(c) Devika Subramanian, 2007

46

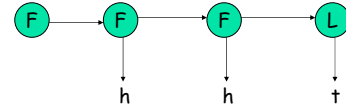
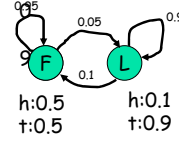
## Generating coin flips

- Start in one of the states, F or L (i.e., pick a fair or loaded coin to start with) (initial probabilities).
- Move to the next state (F or L), based on the transition probabilities. Generate an h or t based on the emission probabilities of that state.
- Repeat above step.

(c) Devika Subramanian, 2007

47

## Generating flips (take 2)



State sequence: FFFL (unobserved)  
Obs sequence : htt (observed)

(c) Devika Subramanian, 2007

48

## Hidden Markov Models

- $S = \{s_1, \dots, s_N\}$ ,  $N$  states
  - $O = \{o_1, \dots, o_M\}$ ,  $M$  observation symbols
  - $a_{ij} = P(S_{t+1}=s_j | S_t=s_i)$ ,  $i, j$  in  $[1..N]$ ; **transition probabilities**
  - $b_i(k) = P(E_t=o_k | S_t=s_i)$ ,  $k$  in  $[1..M]$ ,  $i$  in  $[1..N]$ ; **emission probabilities**
  - $\pi_i = P(S_1=s_i)$ ,  $i$  in  $[1..N]$ ; **initial state probabilities**
- $\lambda = (A, B, \pi)$  specifies the HMM model

(c) Devika Subramanian, 2007

49

## Dishonest casino as an HMM

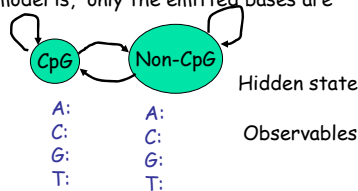
- $N = 2$ ,  $S = \{F, L\}$
- $M = 2$ ,  $O = \{h, t\}$
- $A = \begin{matrix} & F & L \\ F & 0.95 & 0.05 \\ L & 0.10 & 0.90 \end{matrix}$
- $B = \begin{matrix} & h & t \\ F & 0.5 & 0.5 \\ L & 0.1 & 0.9 \end{matrix}$
- $\pi = [1 \ 0]$

(c) Devika Subramanian, 2007

50

## A generative model for CpG islands

- There are two hidden states: CpG and non-CpG. Each state is characterized by emission probabilities of the 4 bases. You can't see which state the model is, only the emitted bases are visible.



(c) Devika Subramanian, 2007

51

## Filtering or the forward computation

- Given an HMM model  $(A, B, \pi)$ , and an observation sequence  $o_1 \dots o_t$ , can we find the most likely hidden state at time  $t$ ,  $S_t$ ?
  - $P(S_t | o_1 \dots o_t)$ : filtering

Observation sequence: h h t t t t

What is the hidden state here (F or L)?

(c) Devika Subramanian, 2007

52

### Filtering (contd.)

What is the distribution of  $S_1$ ?  
 Since,  $s_0=F$ , we can say that  $P(S_1|S_0)=[0.95 \ 0.05]$ , based on the transition probabilities alone.  
 But is that all we know?

$h:0.5$        $h:0.1$   
 $t:0.5$        $t:0.9$

(c) Devika Subramanian, 2007 53

### More filtering

We have also observed **h** at time 1.  
 How can we fold it into the assessment of the distribution of  $S_1$ ?

$h:0.5$        $h:0.1$   
 $t:0.5$        $t:0.9$

(c) Devika Subramanian, 2007 54

### Filtering (contd.)

$$P(S_1 | o_1) = \frac{P(o_1 | S_1)P(S_1)}{P(o_1)}$$

$$P(S_1 = F | o_1 = h) = \alpha P(h | F)0.95 = \alpha(0.5)(0.95)$$

$$P(S_1 = L | o_1 = h) = \alpha P(h | L)0.05 = \alpha(0.1)(0.05)$$

$$\alpha(0.5)(0.95) + \alpha(0.1)(0.05) = 1$$

Therefore,  $P(S_1)=[0.99 \ 0.01]$

(c) Devika Subramanian, 2007 55

### Filtering computation

$P(S_t | o_t, o_1 \dots o_{t-1}) = P(o_t | S_t) \sum_{s_{t-1}} P(S_t | s_{t-1}) P(s_{t-1} | o_1 \dots o_{t-1})$

Recursively computed

(c) Devika Subramanian, 2007 56

### Summary: filtering

Find  $P(S_t | o_1, \dots, o_t) = cP(S_t, o_1, \dots, o_t)$ .

Define  $\alpha_t(i) = P(o_1, \dots, o_t, S_t = s_i)$ .

Initialize:  $\alpha_0(i) = \pi_i, 1 \leq i \leq n$

Recursion:  $\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^n \alpha_t(i) a_{ij}, 0 \leq j \leq n, 1 \leq t \leq T-1$

Termination:  $\alpha_T(i), 1 \leq i \leq n$

Time complexity  $O(n^2T)$

(c) Devika Subramanian, 2007 57

### Smoothing/posterior decoding

Question: can we re-estimate the distribution at  $S_k$  where  $k < t$ , using information about the observed sequence upto time  $t$ ?

That is, what is  $P(S_k | o_1 \dots o_t)$ ?

(c) Devika Subramanian, 2007 58

## Backward computation

$$P(S_k | o_1, \dots, o_t) = \overbrace{cP(o_{k+1}, \dots, o_t | S_k)}^{\text{Backward computation}} \underbrace{P(S_k | o_1, \dots, o_k)}_{\text{Forward computation}}$$

Define  $\beta_k(i) = P(o_{k+1}, \dots, o_t | S_k = s_i)$ .

Initialize:  $\beta_T(i) = 1, 1 \leq i \leq N$ .

Recursion:  $\beta_k(i) = c \sum_{j=1}^N a_{ij} b_j(o_{k+1}) \beta_{k+1}(j), 1 \leq i \leq N, T-1 \leq k \leq 1$

Time complexity:  $O(n^2T)$

(c) Devika Subramanian, 2007 59

## Posterior decoding

$$P(S_k = i | o_1, \dots, o_t) = c\beta_k(i)\alpha_k(i)$$

(c) Devika Subramanian, 2007 60

## Full Decoding

- Given HMM model  $(A, B, \pi)$ , and an observation sequence  $o_1 \dots o_t$ , can we find the most likely hidden state sequence  $s_1 \dots s_t$ ?
  - $\text{argmax}_{\{s_1 \dots s_t\}} P(s_1 \dots s_t | o_1 \dots o_t)$

(c) Devika Subramanian, 2007 61

## The Viterbi algorithm

$$\delta_t(i) = \max_{s_1, \dots, s_{t-1}} P(s_1, \dots, s_{t-1}, S_t = i, o_1, \dots, o_t)$$

Initialize:  $\delta_0(i) = \pi_i, 1 \leq i \leq n$

Recursion:  $\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(o_{t+1}),$   
 $1 \leq t \leq T-1, 1 \leq j \leq n$

Computational complexity =  $O(Tn^2)$

(c) Devika Subramanian, 2007 62

## Learning an HMM: case 1

- Given observation sequences, and the corresponding hidden state sequences, can we find the most likely model  $(A, B, \pi)$  which generated it?

Training data

(c) Devika Subramanian, 2007 63

## Parameter estimation

- Initial state distribution
  - Fraction of times state  $i$  is state 1 in training data
- Transition probabilities
  - $a_{ij}$  = (number of transitions from  $i$  to  $j$ ) / (number of transitions from  $i$ )
- Emission probabilities
  - $b_k(i)$  = (number of times  $k$  is emitted in state  $i$ ) / (number of times state  $i$  occurs)

(c) Devika Subramanian, 2007 64



## Learning an HMM: case 2

- Given just the observation sequences, can we find the most likely model  $\lambda = (A, B, \pi)$  which generated it?

$$\operatorname{argmax}_{\lambda} P(o_1 \dots o_T | \lambda)$$

Annotated training data is difficult to get; so we would like to derive model parameters from observable sequences.

(c) Devika Subramanian, 2007

65

## The EM algorithm

1. Guess a model  $\lambda$
2. Use observation sequence to estimate transition probabilities, emission probabilities, and initial state probabilities.
3. Update model
4. Repeat 2 and 3 till no change in model

(c) Devika Subramanian, 2007

66

## Re-estimating parameters

- What is the probability of being in state  $i$  at time  $t$  and moving to state  $j$ , given the current model and the observation sequence  $O$ ?

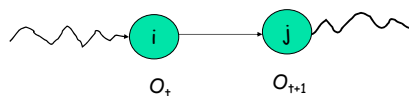
$$\xi_t(i, j) = P(S_t = i, S_{t+1} = j | O, \lambda)$$

(c) Devika Subramanian, 2007

67

## Using forward and backward computation

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^n \sum_{j=1}^n \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}$$



(c) Devika Subramanian, 2007

68

## Re-estimating $a_{ij}$

- The transition probabilities  $a_{ij}$  can be re-estimated as follows

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{j'=1}^n \xi_t(i, j')}$$

(c) Devika Subramanian, 2007

69

## Initial state probabilities

$$\gamma_1(i) = \sum_{j=1}^N \xi_1(i, j) \quad \text{Expected number of times in state } i$$

Initial state probabilities are simply  $\gamma_1(i)$

(c) Devika Subramanian, 2007

70

## Emission probabilities

$\hat{b}_i(k) = \frac{\text{expected number of times in state } i \text{ and observe symbol } k}{\text{expected number of times in state } i}$

$$\hat{b}_i(k) = \frac{\sum_{t=1}^T \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)}$$

(c) Devika Subramanian, 2007

71

## The EM algorithm

1. Guess a model  $\lambda = (a, b, \pi)$
2. Use observation sequence to estimate

$$\xi_t(i, j) \text{ and } \gamma_t(i)$$

3. Use these estimates to recalculate

$$\lambda' = (a', b', \pi')$$

4. Repeat 2 and 3 till no change in model

(c) Devika Subramanian, 2007

72

## Summary of CpG island HMM

- Given a DNA region  $x$ , **Viterbi decoding** predicts locations of CpG islands on it.
- Given a nucleotide  $x_i$ , **Viterbi decoding** tells whether  $x_i$  is in a CpG island in the most likely sequence.
- **Posterior decoding** can assign locally optimal predictions of CpG islands.
- A fully annotated training data set can be used to estimate the generating HMM.
- Even without annotations, we can use the EM procedure to derive model parameters.

(c) Devika Subramanian, 2007

73