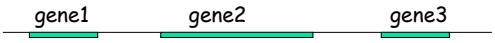
Computational gene finding

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

Finding genes in prokaryotes

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



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.

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

The sequence

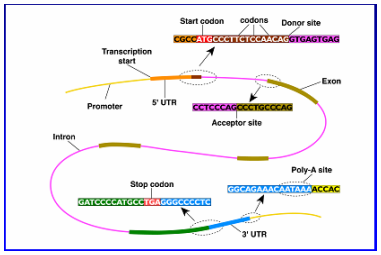
1 aaa**atg**tcgc tgtttggaga cacaattgcc tacctgcttt cattgacaga agatggagaa
61 ggcaaagcag aactagcaga aaaattacac tgttggttcg gtgggaaaga atttgaccta
121 gactctgcct tggaatggat aaaaaacaaa agatgcttaa ctgatataca aaaagcacta
181 attggtgcct ctatctgctt tttaaaaccc aaagaccagg aaaggaaaag aagattcatc
241 acagagcctc tatcaggaat gggaacaaca gcaacaaaaa agaaaggcct gattctagct
301 gagagaaaaa tgagaagatg tgtgagcttt catgaagcat ttgaaatagc agaaggccat
361 gaaagctcag cgctactata ttgtctcatg gtcatgtacc tgaatcctgg aaattattca
421 atgcaagtaa aactaggaac gctctgtgct ttgtgcgaga aacaagcatc acattcacac
481 agggctcata gcagagcagc gagatcttca gtgcccggag tgagacgaga aatgcagatg
541 gtctcagcta tgaacacagc aaaaacaatg aatggaatgg gaaaaggaga agacgtccaa
601 aagctggcag aagagctgca aagcaacatt ggagtattga gatctcttgg agcaagtcaa
661 aagaatgggg aaggaattgc aaaggatgta atggaagtgc taaagcagag ctctatggga
721 aattcagctc ttgtgaagaa atatcta**taa tg**ctcgaacc atttcagatt ctttcaattt
781 gttcttttat cttatcagct ctccatttca tggcttggac aatagggcat ttgaatcaaa
841 taaaaagagg agtaaacatg aaaatacgaa taaaaggtcc aaacaaagag acaataaaca
901 gagaggtatc aattttgaga cacagttacc aaaaagaaat ccaggccaaa gaaacaatga
961 aggaagtact ctctgacaac atggaggtat tgagtgacca catagtgatt gaggggcttt
1021 ctgccgaaga gataataaaa atggggtgaaa cagttttgga gatagaagaa ttgcat**taa**a
1081 ttcaattttt tactgtattt cttattatgc atttaagcaa attgtaatca atgtcagcaa
1141 ataaactgga a

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.

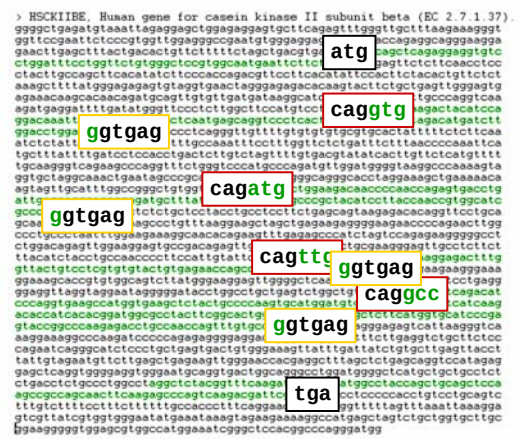
Computational gene finding

- Gene finding in eukaryotic DNA



Structure of a human gene

Exon-intron structure



25

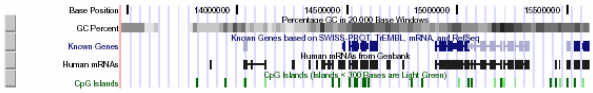
Ab initio methods

- 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.

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

A simple example: CpG Islands

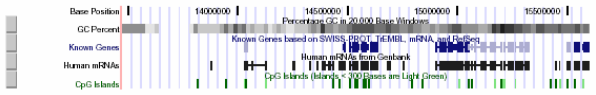


CpG nucleotides in the genome are frequently methylated. (Write CpG not to confuse with CG base pair)

$C \rightarrow$ methyl-$C \rightarrow T$

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

Example: CpG Islands



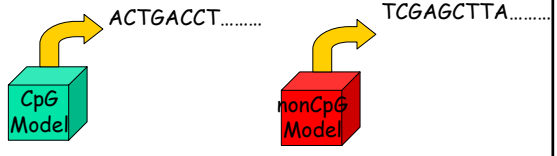
In CpG islands,

CG is more frequent than in the rest of the genome

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?

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.

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.

$$P(w_t \mid w_{t-1},\ldots,w_1) = P(w_t \mid w_{t-1})$$

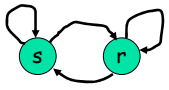
This is the Markov property.

Markov process example

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

Rows of the transition matrix sum to 1.

$$\begin{array}{c} \\ s \\ r \end{array}\begin{array}{c} \begin{array}{cc} s & r \end{array} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \end{array}$$



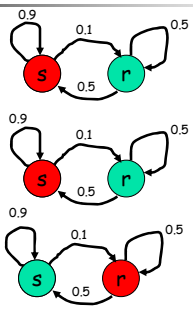
- We know the initial probabilities of s and r.

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|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.

Generating sequences (Take 2)

$$\begin{array}{c} \\ s \\ r \end{array}\begin{array}{c} \begin{array}{cc} s & r \end{array} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \end{array}$$



sequence

s

s

r

Prediction

- Suppose day is rainy . We will represent this as a vector of probabilities over the two values.

$$\pi(1) = [0\ 1];$$

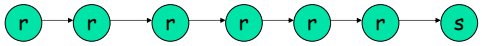
- How do we predict weather on day 2 given pi(1) and the transition probabilities P?
- From P, we can see that the probability of day 2 being sunny is .5, and for being rainy is 0.5

$$\pi(1) * P = [0.5\ 0.5];$$

Probability of a sequence

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

$$P(X = rrrrrrs) = \pi(r)P(r|r)P(r|r)P(r|r)P(r|r)P(r|r)P(s|r)$$
$$= \pi(r)\prod_{t=2..7} P(x_t \mid x_{t-1}) = (0.5)^7$$



Which weather pattern is more likely?

- Given a transition model

$$\begin{array}{c} \\ s \\ r \end{array}\begin{array}{c} \begin{array}{cc} s & r \end{array} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \end{array}$$

- And an initial state distribution: [0.5 0.5]
- And two sequences: rrrrrrs and ssssssr
  Which is more likely, given the model?

Comparing likelihoods

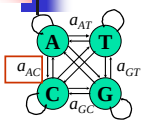
$$P(X = rrrrrrs \mid Model) = \pi(r)[P(r|r)]^5 P(s|r) = (0.5)^7$$
$$P(X = ssssssr \mid Model) = \pi(s)[P(s|s)]^5 P(r|s) = 0.5*(0.9)^5*0.1$$

Markov models (summary)

- States: S = $\{s_1,\ldots,s_N\}$, 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.

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

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?

$$P(x \mid CpG) > P(x \mid nonCpG)$$

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

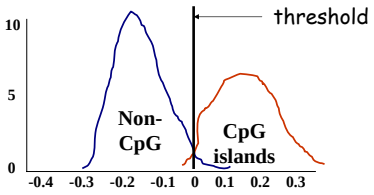
$$\log \frac{P(x \mid CpG)}{P(x \mid nonCpG)} > 0$$

Log-odds ratio

The log-odds ratio

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

Histogram of log-odds scores



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

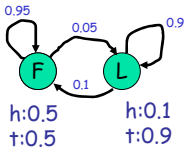
Decision rule: if S(x) > 0 then CpG else non-Cpg

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.

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.



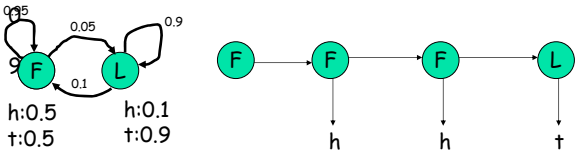
Hidden state

Observables

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.

Generating flips (take 2)



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

Hidden Markov Models

- S = $\{s_1,\ldots,s_N\}$, N states
- O = $\{o_1,\ldots,o_M\}$, 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

Dishonest casino as an HMM

- N = 2, S={F,L}
- M=2, O = {h,t}
- A =

	F	L
F	0.95	0.05
L	0.10	0.90

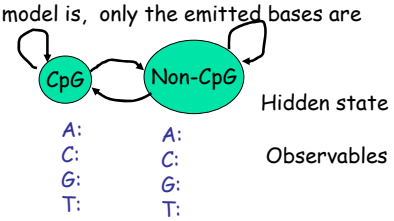
- B=

	h	t
F	0.5	0.5
L	0.1	0.9

- $\pi = [1\ 0]$

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.



Hidden state

Observables

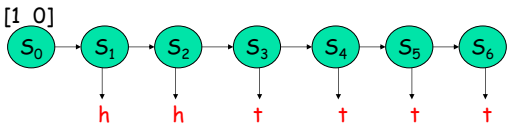
Filtering or the forward computation

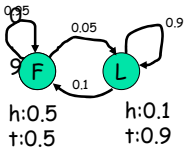
- Given an HMM model (A,B,pi), and an observation sequence $o_1 \ldots o_t$, can we find the most likely hidden state at time t, S_t?
  - $P(S_t | o_1 \ldots o_t)$: filtering

Observation sequence: h h t t t t

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

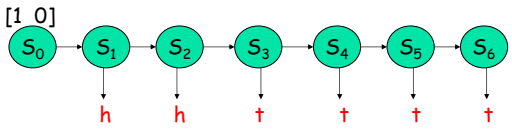
Filtering (contd.)

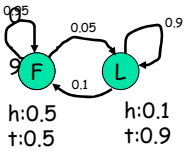




What is the distribution of S_1? Since, s_0=F, we can say that $P(S_1|S_0)$=[0.95 0.05], based on the transition probabilities alone. But is that all we know?

More filtering





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

Filtering (contd.)

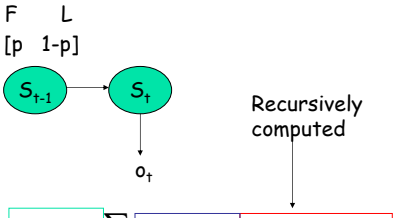
$$P(S_1 \mid o_1) = \frac{P(o_1 \mid S_1)P(S_1)}{P(o_1)}$$

$$P(S_1 = F \mid o_1 = h) = \alpha P(h \mid F)0.95 = \alpha(0.5)(0.95)$$
$$P(S_1 = L \mid o_1 = h) = \alpha P(h \mid L)0.05 = \alpha(0.1)(0.05)$$

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

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

Filtering computation



$$P(S_t \mid o_t, o_1 \ldots o_{t-1}) = P(o_t \mid S_t) \sum_{S_{t-1}} P(S_t \mid s_{t-1}) P(s_{t-1} \mid o_1 \ldots o_{t-1})$$

Summary: filtering

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

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

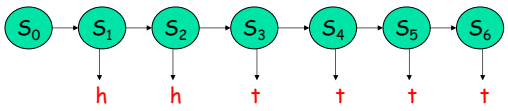
Initialize: $\alpha_0(i) = \pi_i, \quad 1 \le i \le n$

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

Termination: $\alpha_T(i), \quad 1 \le i \le n$

Time complexity $O(n^2T)$

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 $P(S_k \mid o_1 \ldots o_t)$?

Backward computation

$$P(S_k \mid o_1,...,o_t) = cP(o_{k+1},...,o_t \mid S_k) P(S_k \mid o_1,...,o_k)$$

Backward computation

Forward computation

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

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

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

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

Posterior decoding

$$P(S_k = i \mid o_1,...,o_t) = c\beta_k(i)\alpha_k(i)$$

Full Decoding

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

The Viterbi algorithm

$$\delta_t(i) = \max_{x_1,\ldots,x_{t-1}} P(s_1,\ldots,s_{t-1}, S_t = i, o_1,\ldots,o_t)$$

$$\text{Initialize: } \delta_0(i) = \pi_i, 1 \le i \le n$$

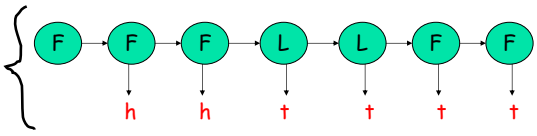
$$\text{Recursion: } \delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(o_{t+1}),$$

$$1 \le t \le T-1, 1 \le j \le n$$

Computational complexity = $O(Tn^2)$

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

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)

Learning an HMM: case 2

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

$$\underset{\lambda}{\operatorname{argmax}} P(o_1 \ldots o_t \mid \lambda)$$

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

The EM algorithm

1. Guess a model λ
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

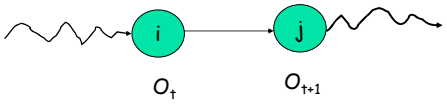
Re-estimating parameters

- 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?

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

Using forward and backward computation

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



Re-estimating a_{ij}

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

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

Initial state probabilities

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j)$$

Expected number of times in state i

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

Emission probabilities

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

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

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

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.