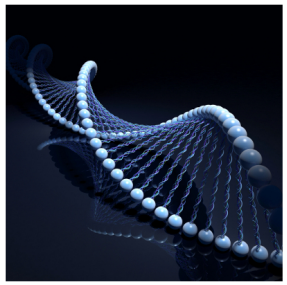
LGBIO2010: Hidden Markov Models

Pierre Dupont



UCL – ICTEAM

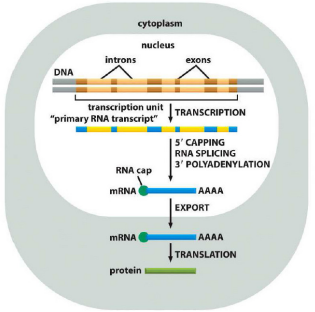
Outline

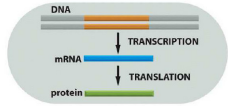
1. Motivating example: gene finding
2. HMM definition
3. The 3 fundamental questions
4. Back to concrete examples
   - Segmentation into CpG islands
   - Profile HMMs

Outline

1. Motivating example: gene finding
2. HMM definition
3. The 3 fundamental questions
4. Back to concrete examples

Ab initio gene finding





One strand of a DNA sequence

. . . ACGCG *TATAAT* GC. . . CAT **ATG** ACGCGT. . . AATCCG **TAA** CGGTCGAAAAA . . .

Gene finding problem

- Find **coding** versus non-coding **fragments**
- Identify splicing between **exons** and **introns** (eukaryotes)

Illustrations from Molecular Biology of the Cell (© Garland Science 2008)

Baseline approach

...ACGCG*TATAAT*GC...CAT**ATG**ACGCGT...AATCCG**TAA**CGGTCGAAAAA...

1. Find all ORFs in the original sequence
2. Find all ORFs in random permutation(s) of the original sequence
3. Accept as significant ORFs, any ORF in the original sequence longer than a prescribed length: *e.g.* the 99% percentile of random ORF lengths ($p-$value = 1%)

Limitations

- Coding vs non-coding fragments are not only characterized by their length
- Intron-exon boundaries also need to be identified for eukaryotes

An alternative approach

Learning

Given some labeled data

- estimate a statistical model M_+ for **coding** fragments (or exons)
- estimate a statistical model M_- for **non-coding** fragments (or introns)

Segmentation

For any new sequence to analyze

- look at a subsequence x defined by a sliding window of size L
- compute log-odds ratio

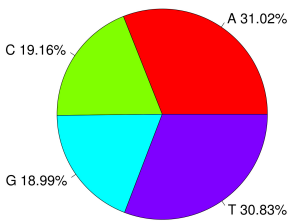
$$\text{If } \log \frac{P(x|M_+)}{P(x|M_-)} = \log \frac{\prod_{i=1}^{L} P(x_i|M_+)}{\prod_{i=1}^{L} P(x_i|M_-)} = \sum_{i=1}^{L} \log \frac{P(x_i|M_+)}{P(x_i|M_-)} > 0$$

then decide x is **coding**
else decide x is **non-coding**

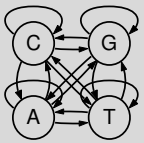
Questions to address

1. Which statistical model to represent specific fragments?
   - Multinomial model
   - Markov chain
   - Hidden Markov model
2. Sliding window size L?
   - No need to use such a fixed window size

Multinomial model



An equivalent representation



- each symbol is generated on a specific state
- all transition probabilities pointing to the same state are equal
- one such model for each segment type (*e.g.* coding vs non-coding)

Markov chain

- Each transition probability $P(x_i|x_{i-1}, M)$ is now specific to a dimer

$$\hat{P}(x_i|x_{i-1}, M) = \frac{f(x_{i-1}x_i)}{f(x_{i-1})} \text{ from segments modeled by } M$$

- Log-odds computation

$$\log \frac{P(x|M_+)}{P(x|M_-)} = \sum_{i=1}^{L} \log \frac{\hat{P}(x_i|x_{i-1}, M_+)}{\hat{P}(x_i|x_{i-1}, M_-)}$$

Second-order Markov chain

- Each transition probability $P(x_i|x_{i-1}, x_{i-2}, M)$ is specific to a 3-mer

$$\hat{P}(x_i|x_{i-1}, x_{i-2}, M) = \frac{f(x_{i-2}x_{i-1}x_i)}{f(x_{i-2}x_{i-1})} \text{ from segments modeled by } M$$

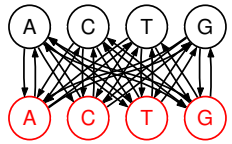
- Log-odds computation

$$\log \frac{P(x|M_+)}{P(x|M_-)} = \sum_{i=1}^{L} \log \frac{\hat{P}(x_i|x_{i-1}, x_{i-2}, M_+)}{\hat{P}(x_i|x_{i-1}, x_{i-2}, M_-)}$$

Limitations of the Markov chain approach

- Estimate each MC from well annotated segments
  - the problem needs to be solved "manually" on a sufficiently large set of segments
- Gene length variability
  - sliding window length is arbitrary
  - computation with many possible window lengths is expensive
  - need to decide which window length is more relevant
- Eukaryotes
  - at least 3 models needed: out-of-genes, introns, exons
  - with even more segment length variability

A combined model



. . . ACGCG *TATAAT* GC. . . CAT **ATG** ACGCGT. . . AATCCG **TAA** CGGTCGAAAAA . . .

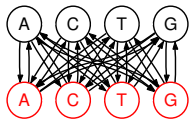
Learning

- Define a single model for all possible segments
  Note: *not all transitions are depicted here*
- Estimate all transition probabilities simultaneously

Segmentation

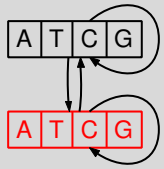
Find the most likely state sequence to generate the whole sequence

Hidden Markov Model



A more compact representation

- transition probabilities between states
- discrete emission probabilities on states
- one state for each segment type:
  (*e.g.* coding vs non-coding)
- states are hidden but their emissions are observed



Outline

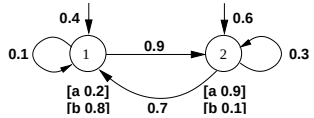
1 Motivating example: gene finding

2 HMM definition

3 The 3 fundamental questions

4 Back to concrete examples

Hidden Markov Model

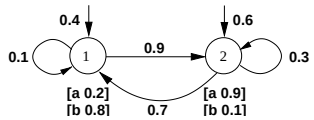


Definition

A discrete HMM (with state emission)

- Σ is a finite alphabet
- Q is a set of states
- $\mathbf{A}$ a $|Q| \times |Q|$ transition probability matrix ($\sum_{q' \in Q} \mathbf{A}_{qq'} = 1$)
- $\mathbf{B}$ a $|Q| \times |\Sigma|$ emission probability matrix ($\sum_{a \in \Sigma} \mathbf{B}_{qa} = 1$)
- π an initial probability distribution ($\sum_{q \in Q} \pi_q = 1$)

An HMM example



$$\Sigma = \{a, b\} \qquad Q = \{1, 2\}$$

$$\mathbf{A} = \begin{bmatrix} 0.1 & 0.9 \\ 0.7 & 0.3 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.2 & 0.8 \\ 0.9 & 0.1 \end{bmatrix} \qquad \pi = \begin{bmatrix} 0.4 & 0.6 \end{bmatrix}$$

Note

Σ and Q need not have the same size

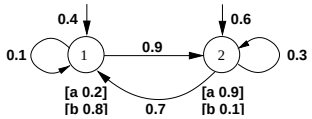
DNA: $\Sigma = \{$A, T, C, G$\}$

Protein: $\Sigma = \{$the 20 amino acids$\}$

Path likelihood

The likelihood $P(s, \nu|M)$ of a sequence $s = s_1 \dots s_{|s|}$ along a path or state sequence $\nu = q_1 \dots q_{|s|}$ in a HMM M

$$P(s, \nu|M) = \prod_{i=1}^{|s|} P(s_i, q_i|M) = \pi_{q_1} \mathbf{B}_{q_1 s_1} \prod_{i=2}^{|s|} \mathbf{A}_{q_{i-1} q_i} \mathbf{B}_{q_i s_i}$$

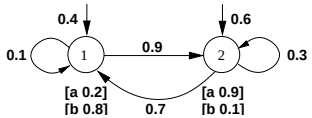


$$P(abb, 122|M) = 0.4 \times \underset{a}{0.2} \times 0.9 \times \underset{b}{0.1} \times 0.3 \times \underset{b}{0.1}$$

Probability of generating a sequence from an HMM

The likelihood $P(s|M)$ of a sequence $s = s_1 \dots s_{|s|}$ in an HMM M

$$P(s|M) = \sum_{\nu \in Q^{|s|}} P(s, \nu|M)$$



$$\begin{aligned} P(abb|M) &= P(abb, 111|M) + P(abb, 112|M) + P(abb, 121|M) + \\ &\quad P(abb, 122|M) + P(abb, 211|M) + \dots \end{aligned}$$

$\mathcal{O}(|Q|^{|s|})$ possible state sequences **!!**

Outline

1 Motivating example: gene finding

2 HMM definition

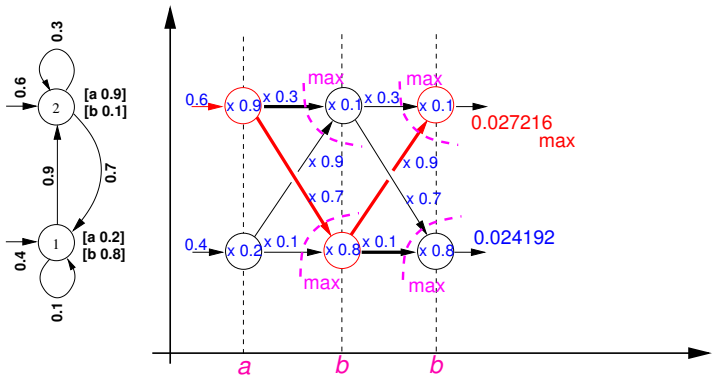
3 The 3 fundamental questions
- How to compute the most likely state sequence?
- How to compute a sequence likelihood?
- How to estimate HMMs parameters?

4 Back to concrete examples

Viterbi algorithm

$$\nu^* = \text{argmax}_\nu \, P(s, \nu|M)$$

Most likely state sequence for $abb = 212$

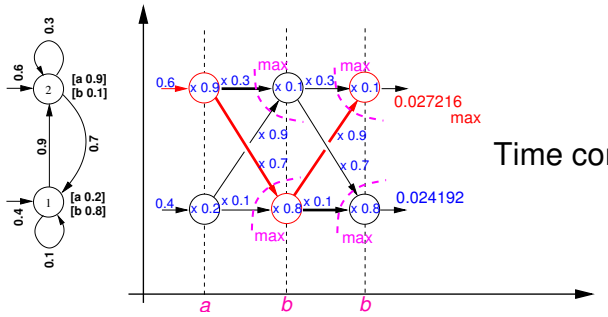


Viterbi recurrence

$$\nu^* = \text{argmax}_\nu \, P(s, \nu | M)$$

Auxiliary quantity: $\gamma(k,t) = P(s_1 \dots s_t, \nu_t^* = k | M)$
The probability of a most likely path ν^* reaching state k at step t

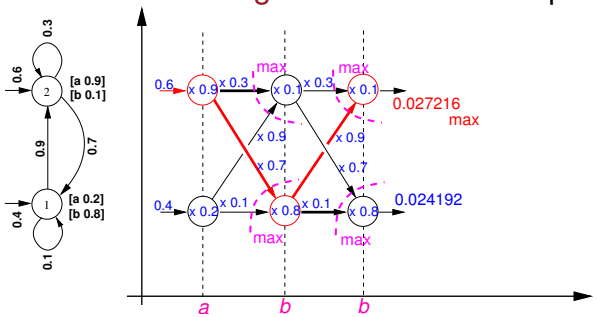
Initialization: $\gamma(k,1) = \pi_k \mathbf{B}_{ks_1}$
Recurrence: $\gamma(k,t) = \max_l [\gamma(l,t-1)\mathbf{A}_{lk}]\mathbf{B}_{ks_t}$
Termination: $P(s, \nu^* | M) = \max_l \gamma(l, |s|)$



Time complexity: $\Theta(m|s|)$

Viterbi alignment

- $P(s, \nu^*)$ gives the probability of an optimal path ν^*
- Computations are done usually with log's:
  $$-\log \gamma(k,t) = \min_l [-\log \gamma(l, t-1) - \log \mathbf{A}_{lk}] - \log \mathbf{B}_{ks_t}$$
- The actual path ν^* can be recovered through the backpointers
- Time complexity is $\Theta(m|s|)$ with m the number of HMM transitions
- The path ν^* defines an alignment between states and symbols
- This alignment defines a segmentation of the sequence : $\underset{2}{a} \mid \underset{1}{b} \mid \underset{2}{b}$

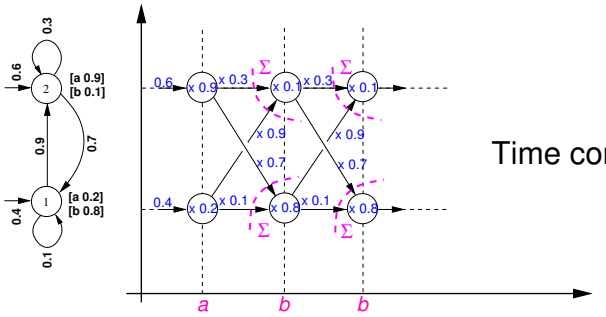


Forward recurrence

$$P(s|M) = \sum_\nu P(s, \nu | M)$$

Auxiliary quantity: $\alpha(k,t) = P(s_1 \dots s_t, \nu_t = k \mid M)$
The likelihood of emitting the first t symbols and reaching state k at time t

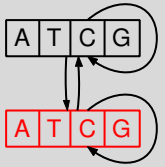
Initialization: $\alpha(k,1) = \pi_k \mathbf{B}_{ks_1}$
Recurrence: $\alpha(k,t) = \sum_l [\alpha(l,t-1)\mathbf{A}_{lk}]\mathbf{B}_{ks_t}$
Termination: $P(s|M) = \sum_l \alpha(l, |s|)$



Time complexity: $\Theta(m|s|)$

The learning problem

Given an HMM structure and one or several sequence(s) to model



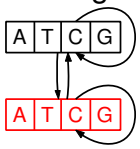
estimate the HMM parameters: $\mathbf{A}, \mathbf{B}, \pi$

Supervised learning

Assumption

The learning sequences are annotated with their respective segments

...GCCAT**ATG**ACGCGT...AATCCG**TAA**CGGT...



- $\mathbf{B}_{ki} = \frac{f(k,i)}{f(k)}$
  - $f(k,i)$ = number of times symbol i is observed on state k
  - $f(k)$ = number of times state k is used
- $\mathbf{A}_{kl} = \frac{f(k,l)}{f(k)}$
  - $f(k,l)$ = number of times a transition from state k to state l is used
- $\pi_k = \frac{f_1(k)}{\sum_{j=1}^{|Q|} f_1(j)}$
  - $f_1(k)$ = number of times state k is used as first state (or $f_1(k) \stackrel{\triangle}{=} 1 \Rightarrow \pi_k = \frac{1}{|Q|}$)

Viterbi training

Unsupervised learning

1. Fix initial parameter values $\mathbf{A}^0, \mathbf{B}^0, \pi^0$
2. **repeat**
   1. compute a most likely path through a Viterbi alignment for each learning sequence given the parameters $\mathbf{A}^i, \mathbf{B}^i, \pi^i$
   2. estimate the emission and transition frequencies from such path(s)
   3. recompute the parameter values $\mathbf{A}^{i+1}, \mathbf{B}^{i+1}, \pi^{i+1}$ from those frequencies

   till some stopping criterion is met (*e.g.* max number of iterations)

Forward-Backward or Baum-Welch Algorithm

Unsupervised learning

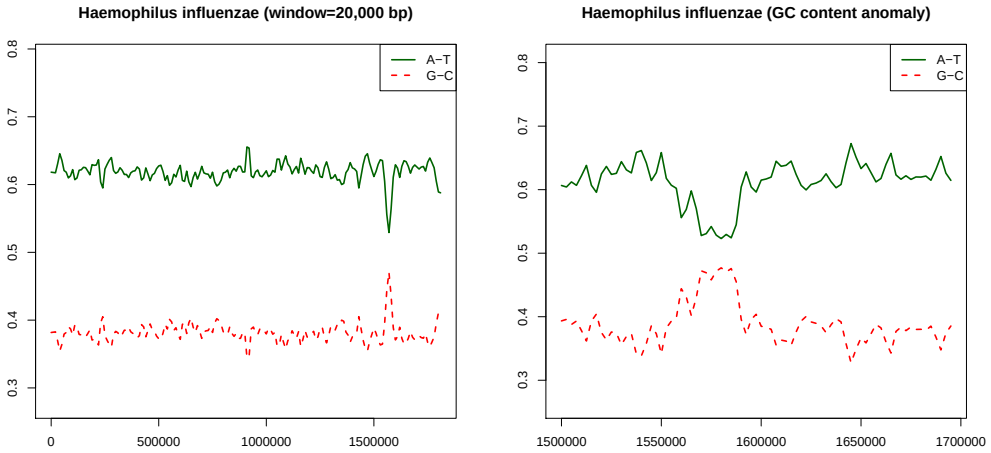
- Viterbi training is an approximation as it considers that each learning sequence is generated along a single path (a most likely one)
- A more accurate estimation is obtained if one considers all possible paths to generate each sequence
  - actual frequencies are replaced by expected frequencies
  - special case of expectation-maximization (EM) procedure

Outline

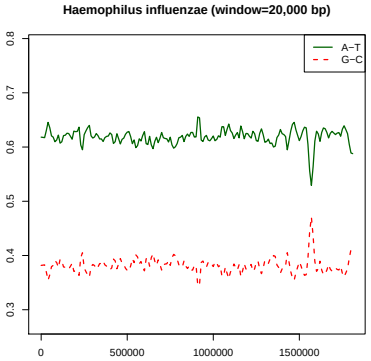
1. Motivating example: gene finding
2. HMM definition
3. The 3 fundamental questions
4. Back to concrete examples
   - Segmentation into CpG islands
   - Profile HMMs

GC content

Haemophilus influenzae (NC_000907)

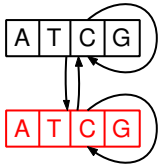


CpG islands



- GC-rich regions are also called CpG islands ($\neq$ GC or CG dimers)
- Baseline analysis use a sliding window of a prescribed length
  - short length: tends to be noisy
  - large length: may miss some short islands
  - actual length varies

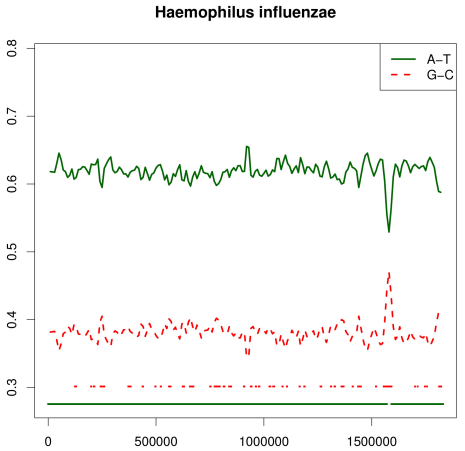
2-state HMM to model CpG islands

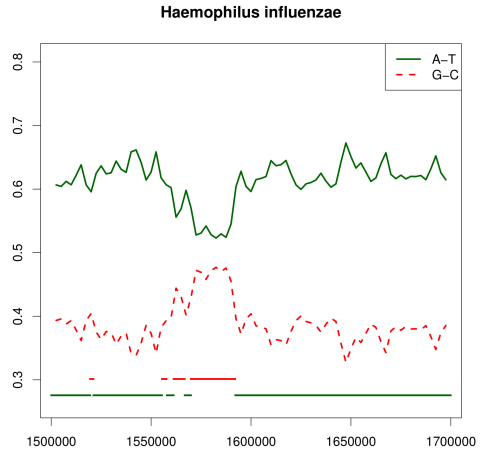


- GC-rich state: higher probability to emit G or C
- alternative state: lower probability to emit G or C
- transition probabilities reflect that is much more likely (*e.g.* 0.998) to stay in a given state than switching regime (*e.g.* 0.002)
- initial probabilities reflect the chance to start or not with a CpG island (by default, $\pi_1 = \pi_2 = 0.5$)

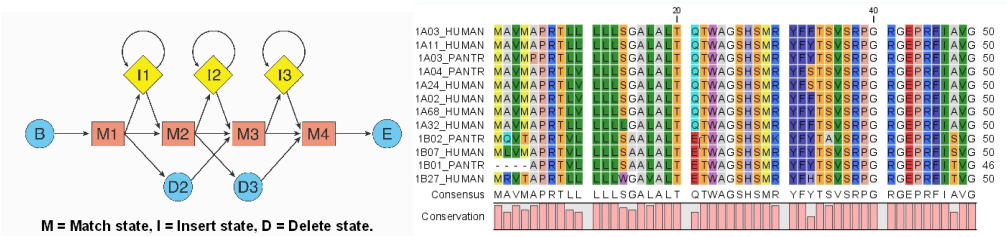
Those probabilities can either be *a priori* fixed or estimated through supervised learning, Viterbi or Baum-Welch

Segmentation into CpG islands





Profile HMMs

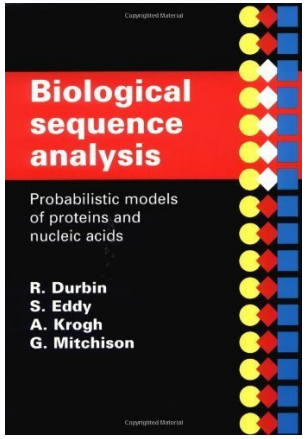


- The topology of a **profile HMM** reflects the structure of a multiple alignment between proteins belonging to the same family
  - match states correspond to conserved parts
  - insert states define possible insertions in less conserved parts
  - delete states represent gaps
- A new sequence can be **aligned** to the model to segment it in conserved *versus* less conserved parts

Software tools

- R packages
  http://cran.r-project.org/web/packages/HMM/
  http://cran.r-project.org/web/packages/hmm.discnp/
- HMMER: biosequence analysis with profile HMMs
  http://hmmer.org/

Further reading



Chapter 3: Markov chains and hidden Markov models