ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-BSSE
Department of Biosystems
Science and Engineering

Markov chains and hidden Markov models

Niko Beerenwinkel

Outline

- Markov chains
- HMM for a single sequence
- CpG islands and genome annotations
- Viterbi decoding
- Forward algorithm
- Backward algorithm
- Baum-Welch algorithm

Markov chain

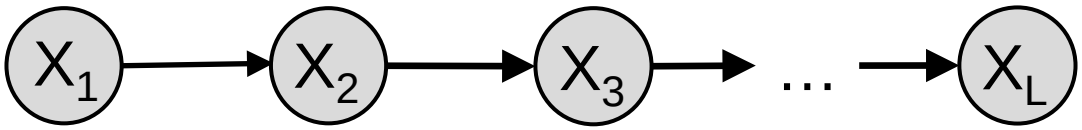
- Let $\{X_1, ..., X_L\}$ be discrete r. v. with common state space [K] = {1, …, K}.
- We always have the factorization

$$\begin{aligned} P(x_1, \ldots, x_L) &= P(x_1, \ldots, x_{L-1}) P(x_L \mid x_{L-1}, \ldots, x_1) \\ &= P(x_1, \ldots, x_{L-2}) P(x_{L-1} \mid x_{L-2}, \ldots, x_1) P(x_L \mid x_{L-1}, \ldots, x_1) \\ &\ldots \\ &= P(x_1) P(x_2 \mid x_1) P(x_3 \mid x_2, x_1) \ldots P(x_L \mid x_{L-1}, \ldots, x_1) \end{aligned}$$

- $\{X_n\}$ is a Markov chain if the **Markov property** holds, i.e., if

$$P(X_n \mid X_{n-1}, \ldots, X_1) = P(X_n \mid X_{n-1})$$

for all $n = 2, ..., L$.



$$X_{n+1} \perp X_{n-1} \mid X_n$$

Transition matrix

- A Markov chain $\{X_n\}$ is homogeneous, if

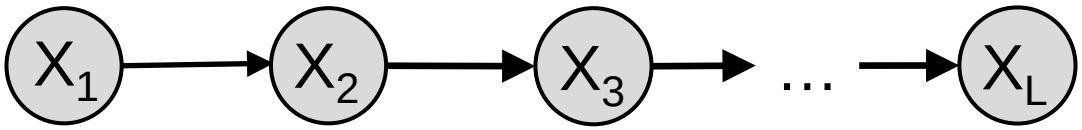
$$P(X_n \mid X_{n-1}) = P(X_2 \mid X_1) \quad \text{for all } n \geq 2$$

- A homogeneous Markov chain is determined by
  - the initial state distribution $I \in \Delta_{K-1}$ defined by

$$I_k = P(X_1 = k)$$

  - and the K $\times$ K transition matrix $T = (T_{kl})$ given by

$$T_{kl} = P(X_{n+1} = l \mid X_n = k)$$

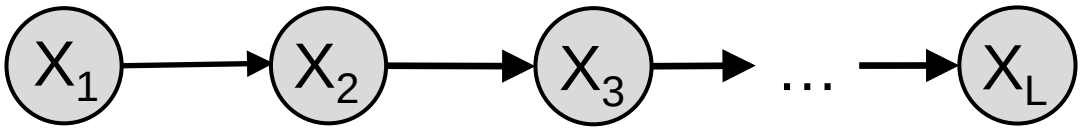


Markov chain model

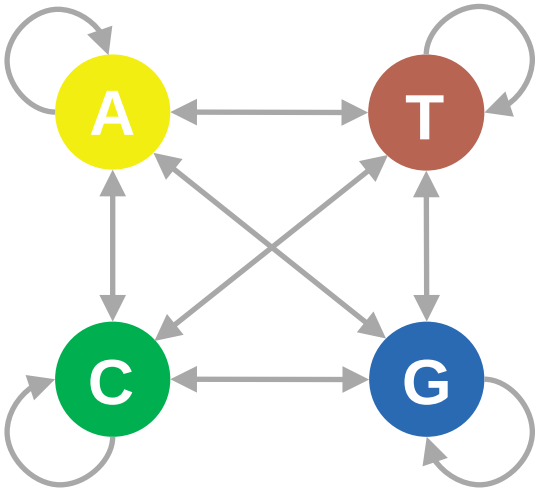
- The probability of an observation x = (x_1, …, x_L) in the Markov chain model MC(I, T) is

$$P(X = x) = P(X_1 = x_1) \prod_{n=1}^{L-1} P(X_{n+1} = x_{n+1} \mid X_n = x_n)$$

$$= I_{x_1} \prod_{n=1}^{L-1} T_{x_n, x_{n+1}}$$

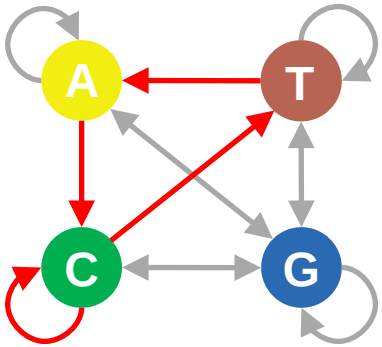


DNA example



$$I = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{pmatrix} .3 \\ .4 \\ .2 \\ .1 \end{pmatrix} \qquad T = \begin{array}{c} \end{array} \begin{array}{cccc} A & C & G & T \\ \end{array} \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{pmatrix} .3 & .1 & .3 & .3 \\ .4 & .1 & .1 & .4 \\ .3 & .2 & .2 & .3 \\ .3 & .2 & .1 & .4 \end{pmatrix}$$

- We consider DNA sequences $x \in \{A,C,G,T\}^*$ as observations of a homogeneous Markov chain $\{X_i\}$.
- For example, P(ACCTA) = 0.3 · 0.1 · 0.1 · 0.4 · 0.3



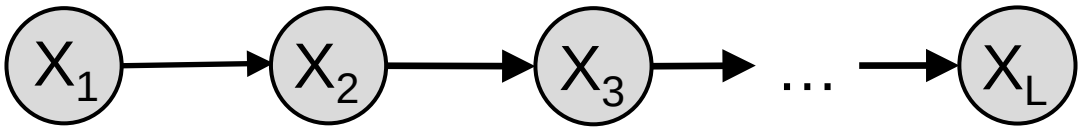
Likelihood

- The likelihood of observed data X = ($X^{(1)}$, …, $X^{(N)}$) is

$$L(I,T) = \prod_{i=1}^{N} \left[I_{X_1^{(i)}} \prod_{n=1}^{L-1} T_{X_n^{(i)},X_{n+1}^{(i)}} \right]$$

$$= \prod_{k\in[K]} I_k^{N_k} \prod_{k,l\in[K]} T_{kl}^{N_{kl}}$$

where N_k is the number of times a chain started in state k, and N_{kl} the total number of k-to-l transitions in the data.



Chapman-Kolmogorov equations

- Denote the probability to jump from state *k* to state *l* in *n* steps by

$$T_{kl}^{(n)} = P(X_{n+j} = l \mid X_j = k)$$

- The Chapman-Kolmogorov equations are

$$T_{kl}^{(n+m)} = \sum_{j=1}^{K} T_{kj}^{(n)} T_{jl}^{(m)}, \qquad n, m \geq 1$$

  or $T^{(n+m)} = T^{(n)} T^{(m)}$ in matrix notation.

- It follows that $T^{(n)} = T^n$.

Ergodicity

- A discrete Markov chain is *ergodic* if it is
  1) aperiodic (return to any state is *always* possible, without a period),
  2) irreducible (any state is accessible from any other, in some #steps),
  3) positive recurrent (any state will eventually be reached with probability 1 and the mean recurrence time is finite).
- **Theorem**: An ergodic Markov chain has a unique stationary distribution $\pi = (\pi_l)_{l \in [K]}$ such that

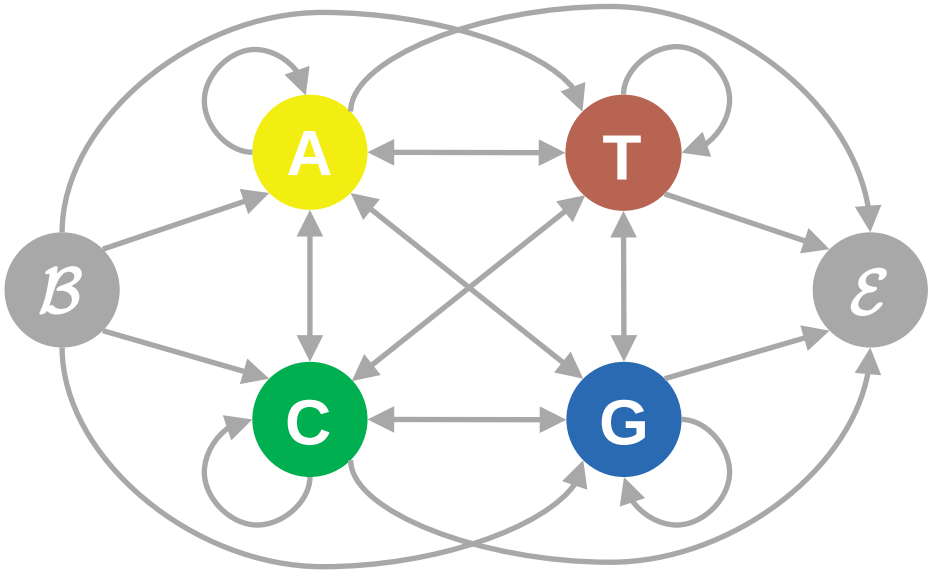
$$\lim_{n \to \infty} T_{kl}^n = \pi_l = \sum_{k \in [K]} \pi_k T_{kl}, \quad l \in [K], \sum_{l \in [K]} \pi_l = 1$$

  independent of the initial distribution I.
- In matrix notation, π is the solution of $\pi^t = \pi^t T$.

https://www.probabilitycourse.com/chapter11/11_2_1_introduction.php

Markov chain for DNA sequences

- Add begin state $\mathcal{B}$ with $x_0 = \mathcal{B}$ and end state $\mathcal{E}$ with $x_L = \mathcal{E}$.



$$P(x) = \prod_{n=1}^{L} T_{x_{n-1},x_n}$$

CpG islands

- CpG islands are stretches of mammalian genomes enriched for the dinucleotide CG, typically 300 to 3,000 bases long.
- Methylated CpG sites tend to mutate as CG > TG, which results in their under-representation: P(CG) < P(C)P(G).
- But in promoter regions, this effect is suppressed and hence CpG islands are more common.

https://en.wikipedia.org/wiki/CpG_site#Under-representation

How can we find CpG islands in a genome?

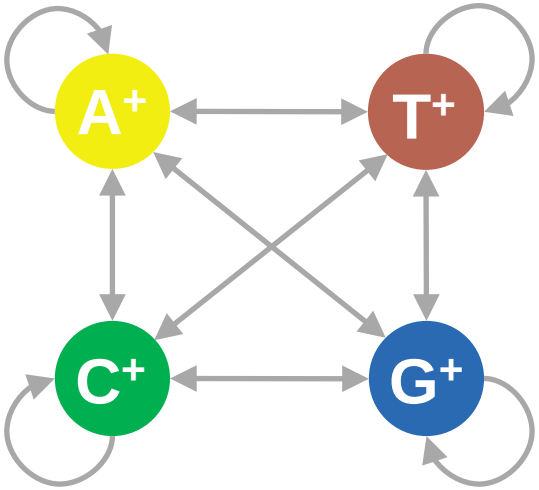
…ACTTCGCGCGCCGATGCCACTGCACATGCATGCATCGCGCGCCGCGCGACAGACTTACG…

Annotating genomic sequences

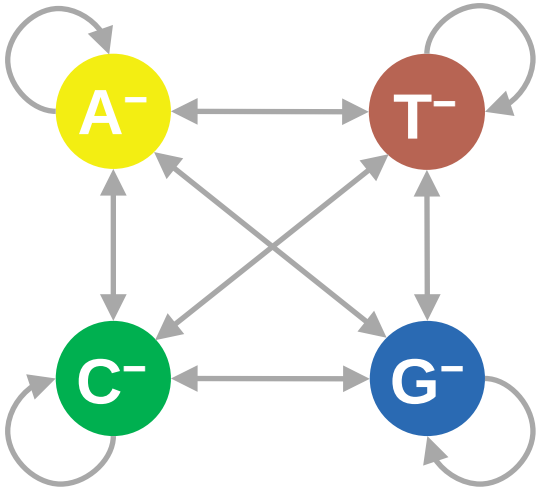
…----++++++++++----------------------------++++++++++++----------…
…ACTTCGCGCGCCGATGCCACTGCACATGCATGCATCGCGCGCCGCGCGACAGACTTACG…

Two Markov chain models

…----++++++++++----------------------++++++++++++----------…
…ACTTCGCGCGCCGATGCCACTGCACATGCATGCATCGCGCGCCGCGCGACAGACTTACG…



CpG island



Non-CpG island

Markov chains for discrimination

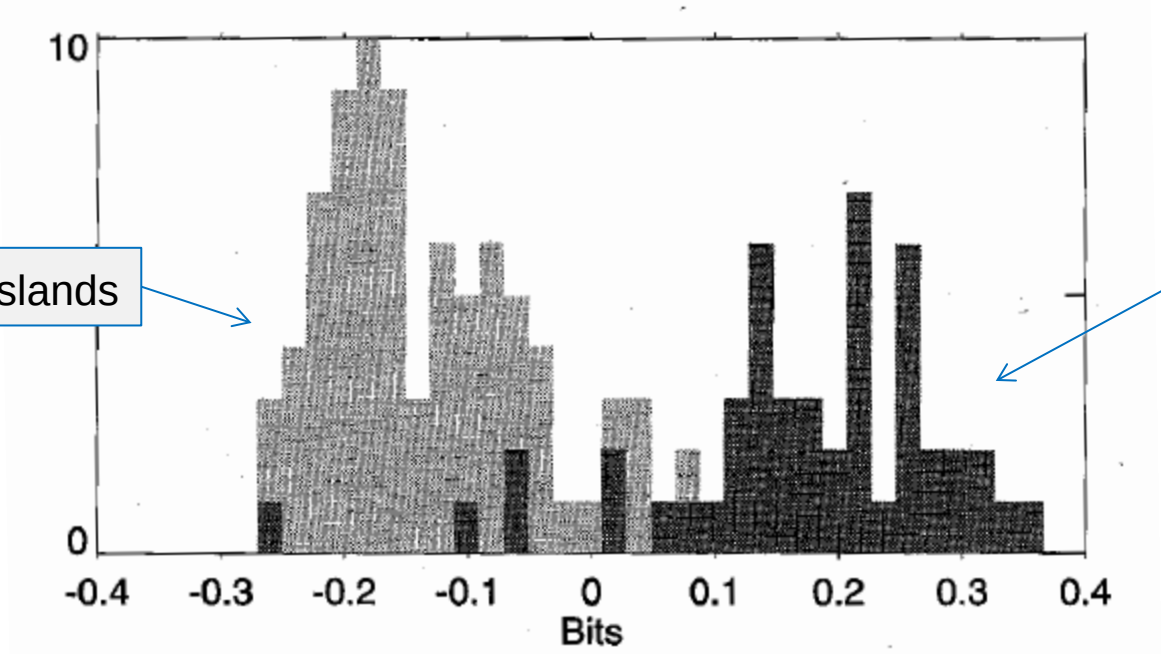
- In a *supervised learning* setting, we are given two sets of DNA sequences labeled as either
  - CpG islands (+), or
  - non-CpG islands (−)
- From each set separately, we estimate the Markov models

$$T_{st}^{+} \quad \text{and} \quad T_{st}^{-}$$

and consider the log-odds score for discrimination:

$$S(x) = \log \frac{P(x \mid T^{+})}{P(x \mid T^{-})} = \log \prod_{n=1}^{L} \frac{T^{+}_{x_{n-1},x_n}}{T^{-}_{x_{n-1},x_n}}$$

Recognition of CpG islands



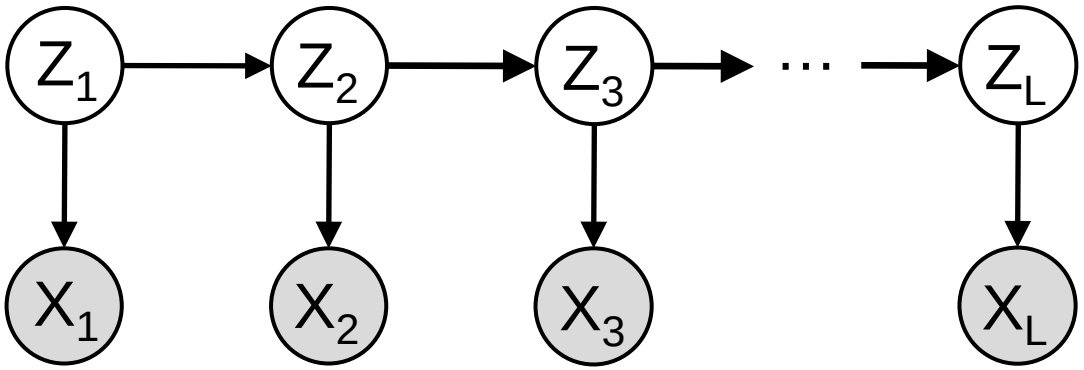
Non-CpG islands

CpG islands

$$S(x) = \sum_{n=1}^{L} \left(\log_2 T^{+}_{x_{n-1},x_n} - \log_2 T^{-}_{x_{n-1},x_n} \right)$$

Hidden Markov model (HMM)

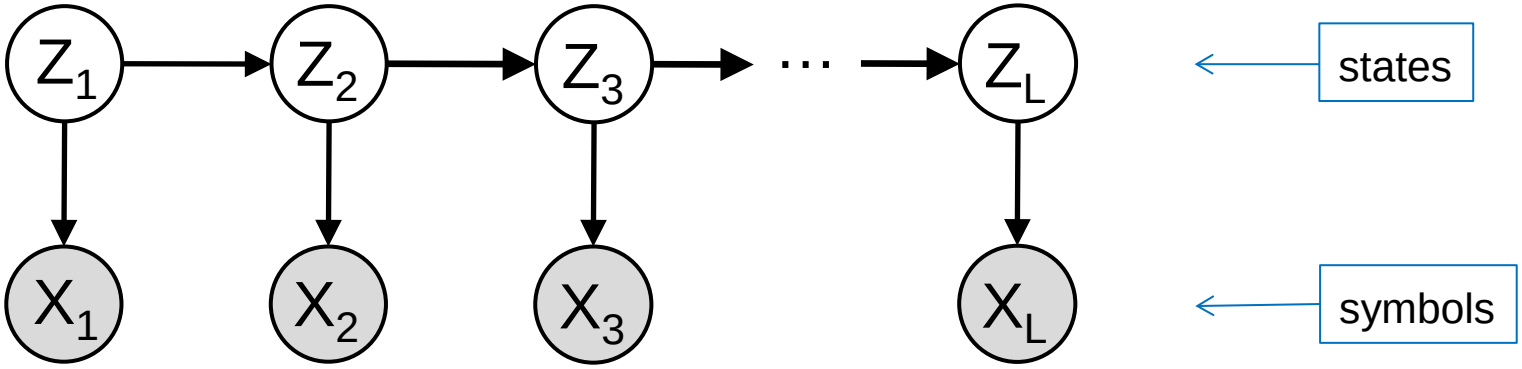
- Hidden (non-observable) random variables $\{Z_n\}$ form a homogeneous Markov chain (the annotation).
  - For example, Z_n indicates whether sequence position n belongs to a CpG island or not, $Z_n \in \{+, -\}$.
- Observed random variables $X_n \in \{A,C,G,T\}$ result from hidden states emitting symbols.



$$Z_{n+1} \perp Z_{n-1} \mid Z_n$$

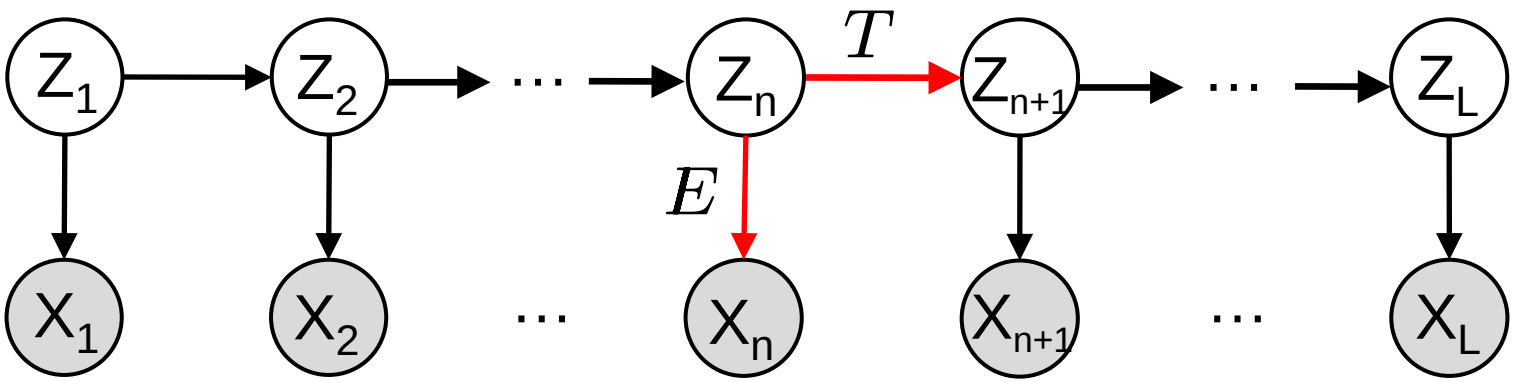
$$X_n \perp X_m \mid Z_n, \; m \neq n$$

Definitions



- Initial state probabilities: $I_k = P(Z_1 = k)$
- Transition probabilities: $T_{kl} = P(Z_n = l \mid Z_{n-1} = k)$
- Emission probabilities: $E_{kx} = P(X_n = x \mid Z_n = k)$

Joint probability



$$P(X,Z) = P(Z_1) \prod_{n=1}^{L} P(X_n \mid Z_n) P(Z_{n+1} \mid Z_n)$$

$$= I_{Z_1} \prod_{n=1}^{L} E_{Z_n, X_n} T_{Z_n, Z_{n+1}}$$

where $P(Z_{L+1} \mid Z_L) = T_{Z_L, Z_{L+1}} \equiv 1$

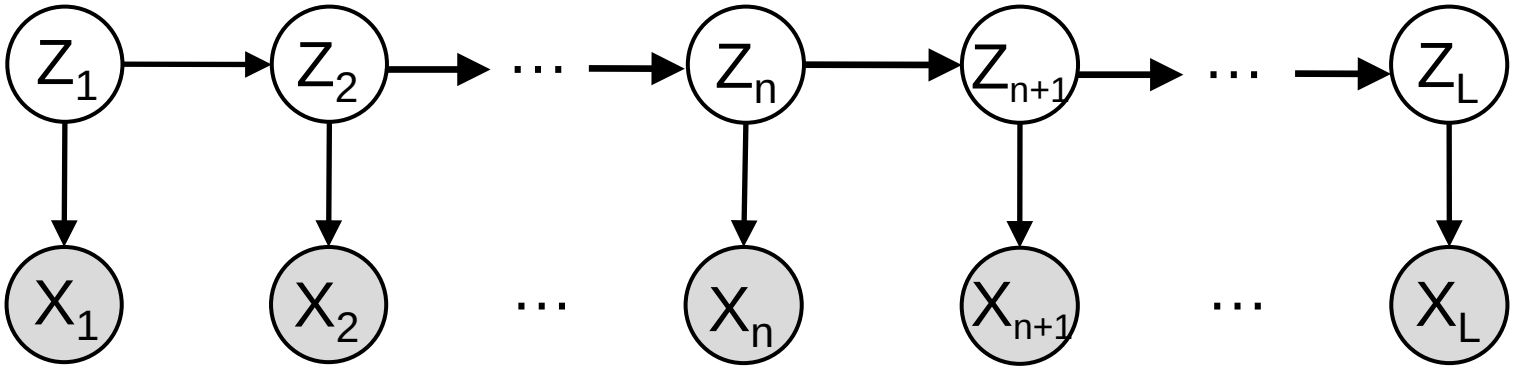
For convenience, use begin and end states:

- Then, with $Z_0 = Z_{L+1} = 0$,

$$P(X,Z) = T_{0,Z_1} \prod_{n=1}^{L} E_{Z_n,X_n} T_{Z_n,Z_{n+1}}$$

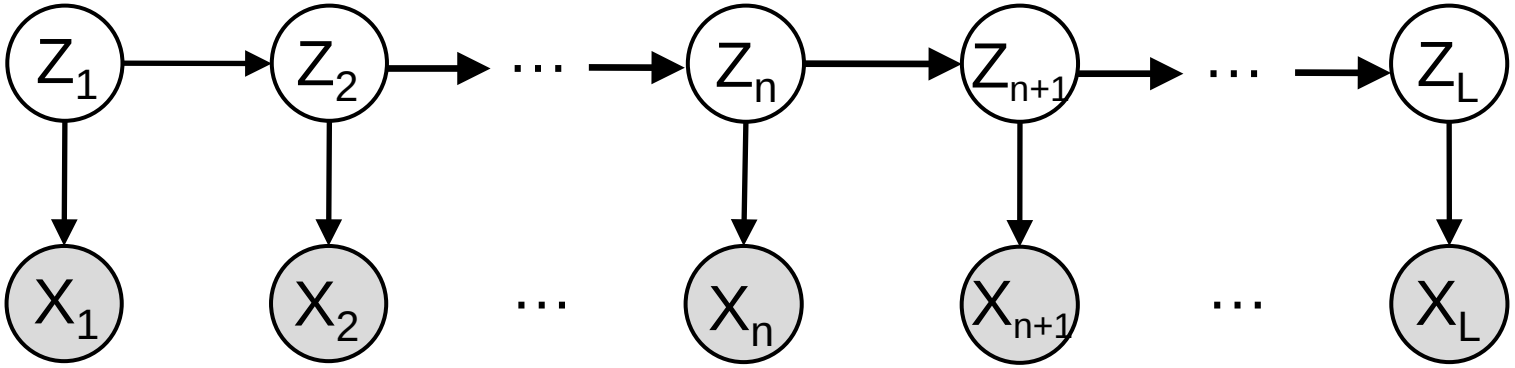
omitting I.

State path



- We observe the DNA sequence X, but we are interested in the hidden states Z of the Markov chain (the *annotation*).
- Each $z = (z_1, \ldots, z_L)$ is called a state path. There are K^L possible paths, where K is the number of (hidden) states.
- Different state path can give rise to the same sequence of observed symbols, but with different probabilities.

Decoding

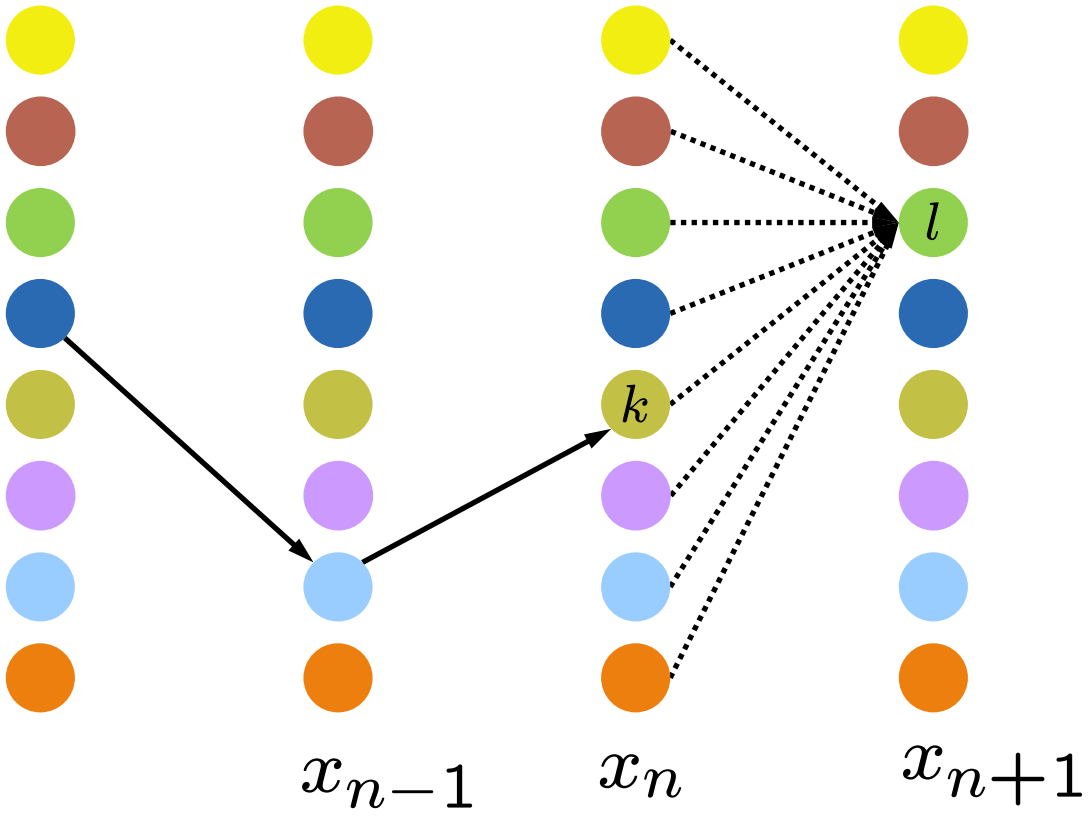


- For given parameters, the decoding problem is to find the most probable state path z* for a given observation x:

$$z^* = \underset{z}{\operatorname{argmax}}\, P(X = x,\, Z = z)$$

Viterbi algorithm: basic idea

- Define $v_k(n)$ as the probability of z* ending in state k with observation x_n
- If $v_k(n)$ is known for all states k, then $v_l(n+1)$ is obtained by maximizing over all states:



$$v_l(n+1) = E_{l,x_{n+1}} \max_k v_k(n) T_{kl}$$

Viterbi algorithm

- Initialization:
  - $v_0(0) = 1$
  - $v_k(0) = 0$ for all $k > 1$
- Recursion: for $n = 1, \ldots, L$,
  - $v_l(n) = E_{l,x_n} \max_k v_k(n-1)T_{kl}$ for all $l = 1, \ldots, K$
  - $ptr_n(l) = \text{argmax}_k v_k(n-1)T_{kl}$ for all $l = 1, \ldots, K$
- Termination (assuming an end state):
  - $P(x, z^*) = \max_k v_k(L)T_{k0}$
  - $z^*_L = \text{argmax}_k v_k(L)T_{k0}$
- Traceback: for $n = L, \ldots, 1$,
  - $z^*_{n-1} = ptr_n(z^*_n)$

- Dynamic programming, $O(LK^2)$ despite K^L paths!

Probability of an observed sequence

- Same trick works for computing

$$P(X) = \sum_Z P(X, Z)$$

- Let

$$f_k(n) := P(X_1 = x_1, \ldots, X_n = x_n, Z_n = k)$$

be the joint probability of the subsequence $x_1, \ldots, x_n$, and the Markov chain ending in state k. Then

$$f_l(n+1) = E_{l,x_{n+1}} \sum_k f_k(n) T_{kl}$$

Forward algorithm

- Initialization:
  - $f_0(0) = 1$
  - $f_k(0) = 0$ for all $k > 1$
- Recursion: for $n = 1, \ldots, L$,
  - $f_l(n) = E_{lx_n} \Sigma_k f_k(n-1)T_{kl}$, for all $l = 1, \ldots, K$
- Termination (assuming an end state):
  - $P(x) = \Sigma_k f_k(L)T_{k0}$

- $O(LK^2)$ despite computing a sum over K^L paths!

Posterior state probabilities

- We want to compute the posterior of each single state,

$$P(Z_n = k \mid x) = \frac{P(x, Z_n = k)}{P(x)}$$

where P(x) is shorthand for P(X = x), etc.

- For the joint probability in the numerator, we find

$$P(x, Z_n = k) =$$

$$\underbrace{P(x_1, \ldots, x_n, Z_n = k)}_{= f_k(n)} \underbrace{P(x_{n+1}, \ldots, x_L \mid Z_n = k)}_{=: b_k(n)}$$

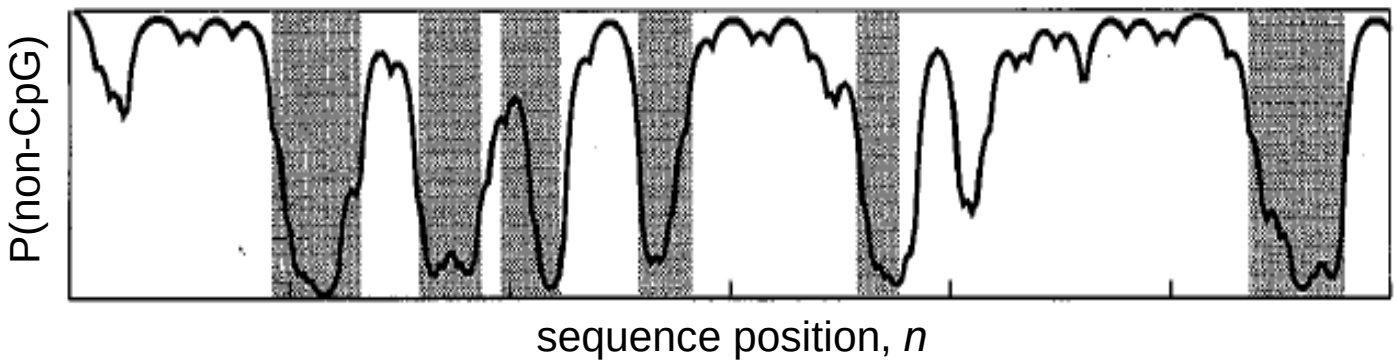
Backward algorithm

- Initialization (assuming an end state):
  - $b_k(L) = T_{k0}$ for all k
- Recursion: for n = L − 1, …, 1,
  - $b_k(n) = \Sigma_l T_{kl} E_{lx_{n+1}} b_l(n+1)$, for all k = 1, …, K
- Termination:
  - $P(x) = \Sigma_l T_{0l} E_{lx_1} b_l(1)$
- $O(LK^2)$

Posterior decoding of CpG islands

- $$P(x_n \text{ is CpG}) = P(Z_n = + \mid x)$$

  is the posterior probability that the base at position *n* lies in a C/G-rich genomic region of *x* and hence may belong to a CpG island.



Predicted CpG islands

Parameter estimation for HMMs

- Suppose we observe sequences $X = \{X^{(1)}, \ldots, X^{(N)}\}$.
- Let us summarize the model parameters T_{kl} and E_{kx} by θ.
- For ML estimation, we have to solve

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \log \sum_{Z} P(X, Z \mid \theta)$$

$$= \operatorname*{argmax}_{\theta} \log \sum_{Z^{(1)}} \cdots \sum_{Z^{(N)}} \prod_{i=1}^{N} P(X^{(i)}, Z^{(i)} \mid \theta)$$

- We can use the EM algorithm!

Joint probability of observation and state path

- For a given path *z*,
  - let $N_{kl}(z)$ be the number of $k \to l$ transitions in *z*, and
  - let $N_{kx}(z)$ be the number of *x* emissions when *z* is in state *k*.

- Then the joint probability of X and Z is

$$P(X,\, Z = z \mid \theta) = \prod_{k,x} E_{kx}^{N_{kx}(z)} \prod_{k,l} T_{kl}^{N_{kl}(z)}$$

E step

- With θ' = θ^{old}, the expected hidden log-likelihood is

$$\begin{aligned}
\mathsf{E}[\ell_{\text{hid}}(\theta)] &= \sum_Z P(Z \mid X, \theta') \log P(X, Z \mid \theta) \\
&= \sum_{i=1}^{N} \sum_{Z^{(i)}} P(Z \mid X, \theta') \left[\sum_{k,x} N_{kx}(Z^{(i)}) \log E_{kx} + \sum_{k,l} N_{kl}(Z^{(i)}) \log T_{kl} \right] \\
&= \sum_{k,x} N_{kx} \log E_{kx} + \sum_{k,l} N_{kl} \log T_{kl}
\end{aligned}$$

where the expected counts are

$$N_{kl} = \mathsf{E}_{Z|X,\theta'} \left[\sum_i N_{kl}(Z^{(i)}) \right], \quad N_{kx} = \mathsf{E}_{Z|X,\theta'} \left[\sum_i N_{kx}(Z^{(i)}) \right]$$

M step

- Maximization w.r.t. θ yields

$$\hat{T}_{kl} = \frac{N_{kl}}{\sum_{l'} N_{kl'}}$$

$$\hat{E}_{kx} = \frac{N_{kx}}{\sum_{x'} N_{kx'}}$$

- The counts N_{kl} and N_{kx} are the *sufficient statistics* of the model.

Computing the sufficient statistics N_{kl}

$$
\begin{aligned}
f_k(n) &= P(x_1, \ldots, x_n, Z_n = k) \\
b_l(n+1) &= P(x_{n+2}, \ldots, x_L \mid Z_{n+1} = l)
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow\ P(Z_n = k, Z_{n+1} = l \mid x) P(x) &= \\
&= P(Z_n = k, Z_{n+1} = l, x) \\
&= f_k(n) T_{kl} E_{l x_{n+1}} b_l(n+1)
\end{aligned}
$$

$$
\Rightarrow\ N_{kl} = \sum_i \frac{1}{P(x^{(i)})} \sum_n f_k^{(i)}(n) T_{kl} E_{l x_{n+1}^{(i)}} b_l^{(i)}(n+1)
$$

Computing the sufficient statistics N_{kx}

- Similarly, one finds

$$N_{ky} = \sum_i \frac{1}{P(x^{(i)})} \sum_{\{n|x_n^{(i)}=y\}} f_k^{(i)}(n) b_k^{(i)}(n)$$

Baum-Welch algorithm (EM for HMMs)

- Initialization:
  - Pick any model parameters
- Recurrence:
  - Set all T and E variables to zero (or add a pseudocount)
  - For each observation i = 1, …, N,
    - Compute $f_k^{(i)}(n)$, for all k, using the forward algorithm
    - Compute $b_k^{(i)}(n)$, for all k, using the backward algorithm
    - Add contribution to T and E.
  - Compute new model parameters
  - Compute new log-likelihood
- Termination:
  - Stop if change in log-likelihood is small

Summary

- Markov chains can model temporal or spatial (linear) dependencies.
- HMMs consist of a hidden state space with a Markov chain structure emitting observable symbols.
- HMMs are frequently used for genome annotation, for example, CpG islands, gene finding, etc.
- The Viterbi algorithm computes the most probable state path and the forward and backward algorithms the likelihood in an efficient way.
- Parameter estimation can be performed using the EM algorithm (Baum-Welch algorithm).

References

- Durbin R, Eddy S, Krogh A, Mitchinson G. Biological Sequence Analysis. Cambridge University Press, 2004. Chapter 3.
- Beerenwinkel N and Siebourg J. Statistics, probability, and computational science. In Maria Anisimova, editor, *Evolutionary Genomics: Statistical and Computational Methods, Volume 1*, chapter 3, pages 77–110. Springer, New York, 2012. DOI: 10.1007/978-1-61779-582-4_3. Section 6.