

# Hidden Markov Models and their Applications in Biological Sequence Analysis

Heitor Baldo  
hbaldo@usp.br  
Algoritmos em Bioinformática (IBI5037)  
Universidade de São Paulo

Dezembro, 2019

# Introdução

Esta apresentação baseia-se no artigo:

Byung-Jun Yoon, *Hidden Markov Models and their Applications in Biological Sequence Analysis*, Current Genomics, 2009, 10, 402-415.

## Resumo

Os Modelos Ocultos de Markov (HMMs) têm sido amplamente utilizados na análise de sequências biológicas. Nesta apresentação, faremos uma revisão dos HMMs e de suas aplicações em problemas na área da biologia molecular.

Embora o artigo trate de três tipos de HMMs, a saber, profile-HMMs, pair-HMMs e HMMs sensíveis ao contexto (csHMMs), discorreremos apenas sobre os dois primeiros tipos.

# Sumário da Apresentação

## 1. Modelos Ocultos de Markov (HMMs)

- ▶ Definição formal de HMM
- ▶ Um HMM simples para modelar genes eucarióticos
- ▶ Problemas básicos e algoritmos para HMMs
- ▶ Os algoritmos Forward, de Viterbi, e de Baum-Welch

## 2. Variantes de HMMs

- ▶ Profile-HMMs
- ▶ Aplicações de Profile-HMMs
- ▶ Pair-HMMs
- ▶ Aplicações de Pair-HMMs

# Modelos Ocultos de Markov (HMMs)

Um **Modelo Oculto de Markov** (Hidden Markov Model) é um modelo estatístico (um processo duplamente estocástico) que pode ser usado para descrever a evolução de eventos observáveis que dependem de fatores internos, os quais não são diretamente observáveis. Chamamos o evento observado de “símbolo” e o fator invisível subjacente à observação de “estado”.

Os estados ocultos formam uma *cadeia de Markov*.

## Definição Formal de HMM

Seja  $x = x_1 \dots x_L$  a sequência de símbolos observados e  $y = y_1 \dots y_L$  a sequência de estados subjacentes,  $x \in O = \{O_1, \dots, O_N\}$ ,  $y_k \in S = \{1, 2, \dots, M\}$ . A probabilidade de transição do estado  $i$  para o  $j$ ,  $i, j \in S$ , é dada por:

$$P(y_{n+1} = j | y_n = i, y_{n-1} = i_{n-1}, \dots, y_1 = i_1) = P(y_{n+1} = j | y_n = i) = t(i, j). \quad (1)$$

Para o estado inicial  $y_1$ , a probabilidade inicial é  $\pi(i) = P(y_1 = i)$ . A probabilidade de emissão de  $x$  no estado  $i$  é:

$$P(x_n = x | y_n = i, y_{n-1}, x_{n-1}, \dots) = P(x_n = x | y_n = i) = e(x|i). \quad (2)$$

As três medidas de probabilidade  $t(i, j)$ ,  $\pi(i)$ , e  $e(x|i)$  especificam completamente um HMM, e denotamos o conjunto desses parâmetros por  $\Theta$ .

## Um HMM simples para modelar genes eucarióticos

O HMM tem quatro estados ocultos,  $E_1, E_2, E_3, I$ , em que  $E_k$  são usados para modelar exons, e o estado  $I$  é usado para modelar introns.

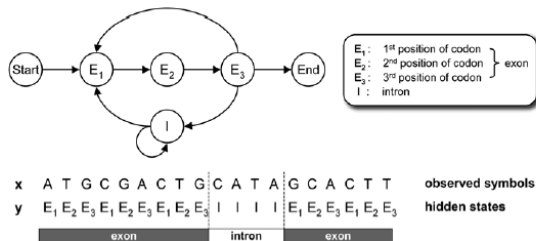


Fig. (1). A simple HMM for modeling eukaryotic genes.

Figure: Figura retirada de (Yoon, B.-J., 2009).

## Um HMM simples para modelar genes eucarióticos

A HMM construída anteriormente pode ser usada para analisar novas sequências. Por exemplo, considera a sequência de DNA  $x = x_1 \dots x_{19} = \text{ATGCGACTGCATAGCACTT}$ . Como podemos saber se essa sequência é ou não um gene codificante?

Podemos responder à essa questão pelo cálculo da probabilidade de observação de  $x$  baseado no HMM que modela genes codificantes. Se a probabilidade for alta, implica que provavelmente essa sequência de DNA seja um gene codificante.



# Os três problemas básicos de HMMs

A maioria das aplicações de HMMs envolve a solução de três problemas básicos:

1. Suponha que tenhamos uma sequência de símbolos  $x = x_1 \dots x_L$ . Como podemos calcular a probabilidade de observação  $P(x|\Theta)$  com base em um dado HMM  $\Theta$ ? (**Scoring Problem**)
2. Para uma sequência de símbolos  $x = x_1 \dots x_L$ , dado o HMM  $\Theta$ , encontrar a sequência ótima de estados (ou o caminho ótimo), tal que maximize a probabilidade de observação da sequência  $x$ . Ou seja, queremos a sequência de estados  $y$  tal que  $P(y, x|\Theta)$  seja máxima. (**Optimal Alignment Problem**)

## Os três problemas básicos de HMMs

3. Estimar os parâmetros do modelo  $\Theta$  de modo que a probabilidade  $P(x|\Theta)$  ou  $P(y, x|\Theta)$  seja maximizada. (**Training Problem**).

Os algoritmos comumente utilizados para solução desses problemas são o **algoritmo Forward**, o **algoritmo de Viterbi** e o **algoritmo de Baum-Welch**, respectivamente.

## Algoritmo Forward

O algoritmo Forward é um algoritmo de programação dinâmica utilizado para calcular  $P(x|\Theta)$  de uma maneira eficiente.

Ao invés de enumerar todas as sequências de estados possíveis, definimos a *variável forward*:

$$\alpha(n, i) = P(x_1 \dots x_n, y_n = i | \Theta). \quad (3)$$

Esta variável pode ser recursivamente computada usando a fórmula:

$$\alpha(n, i) = \sum_k [\alpha(n-1, k) t(k, i) e(x_n | i)], \quad (4)$$

para  $n = 2, \dots, L$ . Ao final das recursões, podemos calcular  $P(x|\Theta) = \sum_k \alpha(L, k)$ . Este algoritmo calcula  $P(x|\Theta)$  em tempo  $O(ML^2)$ .

## Algoritmo de Viterbi

O algoritmo de Viterbi é um algoritmo de programação dinâmica que é usado para calcular o caminho ótimo  $y^* = \arg \max_y P(y|x, \Theta)$  de forma eficiente. Este algoritmo define a variável:

$$\gamma(n, i) = \max_{y_1 \dots y_{n-1}} P(x_1 \dots x_n, y_1 \dots y_{n-1} y_n = i | \Theta), \quad (5)$$

e a calcula de forma recursiva usando a seguinte fórmula:

$$\gamma(n, i) = \max_k [\gamma(n-1, k) t(k, i) e(x_n | i)]. \quad (6)$$

Ao final das recursões, podemos obter a *máxima probabilidade*:

$$P^* = \max_y P(x, y | \Theta) = \max_k \gamma(L, k). \quad (7)$$

## Algoritmo de Viterbi

O caminho ótimo  $y^*$  pode ser facilmente encontrado fazendo o *traceback* das recursões que conduzem à máxima probabilidade  $P^* = P(x, y^* | \Theta)$ . Este algoritmo encontra a sequência ótima de estados em tempo  $O(ML^2)$ .

O algoritmo de Viterbi encontra o caminho ótimo que maximiza a probabilidade de observação da sequência de símbolos inteira. Para predizer o estado ótimo em uma posição específica usamos a abordagem de **posterior-decoding**

$$\hat{y}_n = \arg \max_i P(y_n = i | x, \Theta), \quad (8)$$

em que a *probabilidade a posteriori*  $P(y_n = i | x, \Theta)$  pode ser calculada usando o **algoritmo backward**.

## Algoritmo de Baum-Welch

Embora não exista uma maneira ótima para estimar os parâmetros a partir de um número limitado de sequências finitas, existem maneiras de encontrar os parâmetros do HMM que maximizam localmente a probabilidade de observação.

Por exemplo, podemos usar o algoritmo de Baum-Welch para treinar o HMM. Este algoritmo é um algoritmo de *Maximização de Expectativa* (EM) que, iterativamente, calcula e atualiza os parâmetros  $\Theta$  pelo método *forward-backward*.

## Profile-HMMs

Profile-HMMs são especialmente úteis para representação do perfil de um alinhamento múltiplo de sequências. Eles são HMMs com uma arquitetura específica que é adequada para modelar perfis de sequências.

Um profile-HMM usa repetidamente três tipos de estados ocultos, a saber, estados de **match**  $M_k$ , estados de **inserção**  $I_k$  e estados de **deleção**  $D_k$ .

A seguir, construímos um profile-HMM baseado em um alinhamento múltiplo.

# Construindo uma Profile-HMM

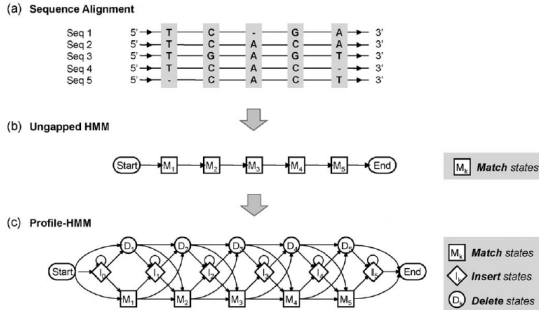


Fig. (2). Profile hidden Markov model. (a) Multiple sequence alignment for constructing the profile-HMM. (b) The ungapped HMM that represents the consensus sequence of the alignment. (c) The final profile-HMM that allows insertions and deletions.

Figure: Figura retirada de (Yoon, B.-J., 2009).

O número de estados de match no profile-HMM resultante é idêntico ao comprimento da sequência consenso.



## Construindo uma Profile-HMM

A probabilidade de emissão  $e(x|M_k)$  no  $k$ -ésimo estado de match  $M_k$  reflete a frequência do símbolo observado na  $k$ -ésima coluna consenso.

Construímos o HMM *sem gaps* pela interconexão dos estados de match  $M_1, \dots, M_5$ . Uma vez que tenhamos construído esse HMM sem gaps, adicionamos os estados de inserção  $I_k$  e de deleção  $D_k$  ao modelo, assim podemos modelar inserções e deleções. Subsequentemente, obtemos o profile-HMM final.

Para permitir pequenas probabilidades nas transições de estados ou nas emissões de símbolos que não são observados no alinhamento original, podemos adicionar **pseudocontadores**.

## Aplicações de Profile-HMMs

1) Profile-HMMs podem ser usados para encontrar *alinhamentos múltiplos de sequências*, para *classificação de proteínas* e na *detecção de motivos*. O software **HMMER** pode ser utilizado para construção e treino de profile-HMMs.

Durbin, R., et al. *Biological Sequence Analysis*, Camb. Uni. Press, 1998.

2) Profile-HMMs também podem ser usados para comparação de perfis de sequência, que pode ser útil para detecção de homólogos remotos. Por exemplo, o **COACH** permite comparar alinhamentos de sequências, pela construção de um profile-HMM a partir de um alinhamento e alinhando o outro alinhamento ao profile-HMM construído.

Edgar, R. C., et al. *Bioinformatics*, 2004, 20, 1309-1318.

## Aplicações de Profile-HMMs

3) Ainda, profile-HMMs podem ser usados para modelar símbolos de sequências secundárias de proteínas: helix (H), strand (E), e coil (C).

Di Francesco et al., Bioinformatics, 1999, 15, 131-140.

## Pair-HMMs

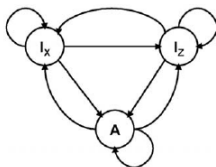
Pair-HMM é uma variante do HMM básico que é especialmente útil para encontrar alinhamentos de sequências e avaliar a significância dos símbolos alinhados.

Ao contrário do HMM original, o qual produz uma única sequência, um pair-HMM produz um par alinhado de sequências.

Um exemplo simples de pair-HMM é considerado a seguir.

## Pair-HMMs

Pair HMM



$I_x$ : insertion in  $x$  (seq 1)

$I_z$ : insertion in  $z$  (seq 2)

$A$ : aligned symbols in  $x$  and  $z$

$x$  (seq 1) : T T C C G - -

$z$  (seq 2) : - - C C G T T

$y$  (states) :  $I_x I_x A A A I_z I_z$

Fig. (3). Example of a pair hidden Markov model. A pair-HMM generates an aligned pair of sequences. In this example, two DNA sequences  $x$  and  $z$  are simultaneously generated by the pair-HMM, where the underlying state sequence is  $y$ . Note that the state sequence  $y$  uniquely determines the pairwise alignment between  $x$  and  $z$ .

Figure: Figura retirada de (Yoon, B.-J., 2009).

Este pair-HMM transita entre os estados  $I_x$ ,  $I_z$  e  $A$  para gerar simultaneamente duas sequências alinhadas de DNA,  $x$  e  $z$ .

## Pair-HMMs

Com base na estrutura do pair-HMM, o problema de encontrar o melhor alinhamento entre  $x$  e  $z$  se reduz ao problema de encontrar a seguinte sequência ótima de estados:

$$y^* = \arg \max_y P(y|x, z, \Theta).$$

Uma importante vantagem do pair-HMM sobre os algoritmos tradicionais de alinhamento é que podemos utilizá-lo para calcular a probabilidade do alinhamento de um par de sequências.

## Aplicações de Pair-HMMs

1) O método **MCALIGN2** adota pair-HMMs, com uma estrutura ligeiramente diferente, para o alinhamento global pareado de segmentos de DNA não-codificantes.

Wang, J., et al. *BMC Bioinform.*, 2006, 7, 292.

2) O algoritmo **ProbCons**, utilizado para alinhamento múltiplo de seqüências (MSA), utiliza pair-HMMs para calcular as probabilidades *a posteriori* de alinhamentos.

Do, C. B., et al. *Genome Res.*, 2005, 15, 330-340.

## Aplicações de Pair-HMMs

3) Pair-HMMs são também utilizados para a predição de genes:

Pachter, L., et al. J. Comput. Biol., 2002, 9, 389-399.

Alexandersson, M., et al. Genome Res., 2003, 13, 496-502.

Arumugam, M., et al. Genome Biol., 2006, 7(Suppl 1), S5.1-10.

4) Pair-HMMs podem ser estendidos para o alinhamento de estruturas mais complexas, como, por exemplo, *árvores*. Os modelos PHMMTs estendem os pair-HMMs para esse tipo de alinhamento.

Sakakibara, Y. Bioinformatics, 2003, 19, i232-i240.



Obrigado!