

# High-Order and PWM Based Hidden Markov Model (HOP-HMM)

August 15, 2017

## Abstract

## Introduction

[enhancer background]  
[PWMs and motif to classify tissue specific enhancers]  
[k-mer to classify tissue specific enhancers]  
[HMM to classify tissue specific enhancers]  
[other machine learning work to classify tissue specific enhancers]  
[Why the HOP-HMM approach to the problem differently]

## Background

[generative models]  
[sequences that could have been generated by HOP-HMM]

## Methods

Generalized Hidden Markov Model (GHMM)

HOP-HMM is a type of Generalized Hidden Markov Model (GHMM), which is a generative model, meaning it relies on the assumption that the observed DNA sequence was generated by a parameterized model  $\theta$ . Our goal is to find a model that fits the set of training sequences or in other words a  $\theta$  that maximizes the log likelihood of the observed sequence.

$$l(x_1, \dots, x_L | \theta) = \log P(x_1, \dots, x_L | \theta)$$

As in a standard HMM, we assume an underlying hidden variables are present, and these variables indicate the state of the model while creating the sequence. As seen in figure 2 and figure 3,

TOMMY:

Generative model. A set of sequences could have been generated by this model. Our goal is to optimize the model parameters such that the likelihood is optimized. While so, we would "annotate" the sequences (explain how and why). Begin by describing the model automaton (state machine) with a figure. Only then talk about the emissions. Wrap it up by describing how data is generated (according to the model) and how this would help you understand enhancer sequences.

### Setup

Let us consider a high-order emission base-states and PWM emission sub-states HMM from a dataset of N observations sequences  $X = (X_1, \dots, X_N)$  where each observation sequence is L nucleotides long  $X_i = (x_1^i, \dots, x_L^i)$ . We assume an underlying hidden variable sequences  $Y = (Y_1, \dots, Y_N)$  where each underlying sequence is also L variables long  $Y_i = y_1^i, \dots, y_L^i$ . Let the space of underlying states be  $\Upsilon = \{1, 2, \dots, m\} \times \{0, 1, \dots, k\}$

TOMMY:

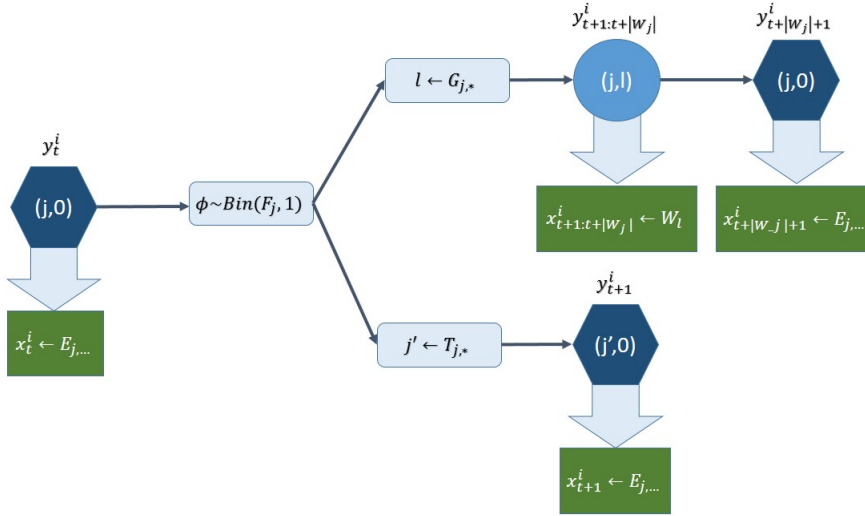
quite abstract and hard to follow. try to be more concrete. See above. Add Figure!

not clear enough. I think your best option is to separate the figure into two parts, like I said before. Begin by a single "layer" and first describe the automaton and the transition probabilities (maybe with a matrix). Then show the generated sequence with the (hidden) states above. You can also plot the dependencies among them (Markovian model). Then make the model more complex, and keeping explaining with automaton figures.

### Emission and Transition

Underlying states emit the observed sequence are of two types: base-states and their sub-states. We mark the  $j$ 'th base-state as  $(j, 0)$  for  $j \in \{1, \dots, m\}$  and its  $l$ 'th sub-state as  $(j, l)$  for  $l \in \{1, \dots, k\}$ . Denote the base-state emission order by  $o$ , meaning a base-state emits a letter sampled from an emission matrix  $E$  that depends on previous  $o - 1$  letters.

Sub-state emits multiple letters sampled from a PWM that is fixed and isn't learned in the training. Denote  $W_l$  the PWM of the  $l$ 'th sub-states, which is shared between the  $l$ 'th sub-states of all base-states.



**Figure 4: emission and transition process between base-state. The upper flow represent transition from base-state to the same base-state, through a sub-state that emits a motif. The lower flow represent transition between two base-state using the  $T$  transition matrix, similar to the conventional HMM.**

After emitting a single letter, the  $j$ 'th base-state has a probability  $F_j$  to make a transition into one of its sub-state and emit a motif and probability  $1 - F_j$  to make a transition into one of the base-states and emit a single letter. The distribution of transitions between base-states is set by  $T$  matrix, and between base-state to its sub-states by  $G$  matrix. After emitting a motif in a sub-state, the next state will be the sub-state's base-state where it will emit a single letter.

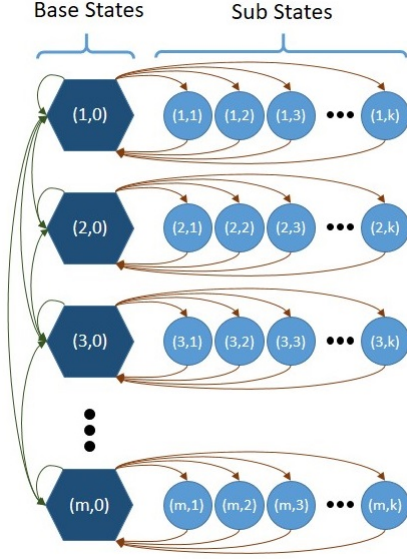


Figure 1: The hidden variable states graph of the HOP-HMM. The left hexagons represent base-states, and the circles in the right part of each row's represent its sub-states.

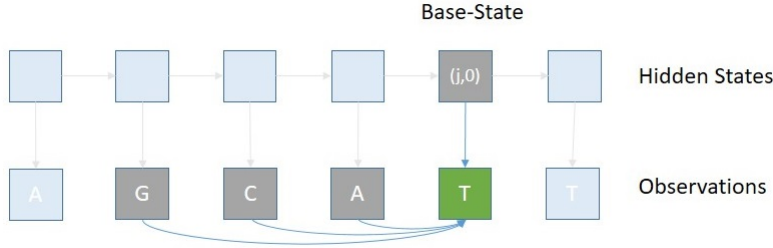


Figure 2: high-order emission of base-states. Each emission is dependent on the hidden base-state and o-1 previous observations.

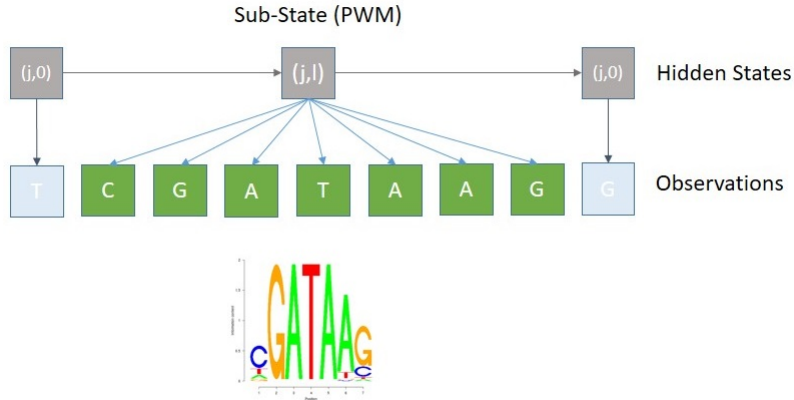


Figure 3: PWM emission of sub-states.

### Parameters

An HOP-HMM  $\theta = \{\pi, E, T, G, F\}$  is parameterized by:

- $\pi$  :  $m \times 1$  initial base-state distribution vector

$$\pi_j = P(y_1^i = j)$$

- $E$  :  $m \times \underbrace{4 \times 4 \times \dots \times 4}_{o \text{ times}}$  the base-state high-order emission probability matrix

$$E_{j,b_1,b_2,\dots,b_o} = P(x_t^i = b_o | y_t^i = (j, 0), x_{t-o+1}^i = b_1, \dots, x_{t-1}^i = b_{o-1})$$

- $T$  :  $m \times m$  the transition probability matrix

$$T_{j_1,j_2} = P(y_t^i = (j_2, 0) | y_{t-1}^i = (j_1, 0))$$

- $G$  :  $m \times k$  the sub-state entry probability matrix

$$G_{j,l} = P(y_t^i = (j, l) | y_{t-1}^i = (j, 0), l > 0)$$

- $F$  :  $m \times 1$  the probability to enter one of the sub-states from a base state

$$F_j = P(l > 0 | y_{t-1}^i = (j, 0), y_t^i = (j, l))$$

note: since these describe distributions:  $\sum_{b \in \{1, \dots, n\}} E_{i,b_1,b_2,\dots,b_{o-1},b} = 1$ ,  $\sum_{l \in \{1, \dots, k\}} G_{j,l} = 1$  and  $\sum_{j_2 \in \{1, \dots, m\}} T_{j_1,j_2} = 1$

note 2: we marked here the index of the sequence by  $i \in \{1, \dots, N\}$

## EM Algorithm

Our task

### E-Step

#### Forward Algorithm

denote  $L_M(\bar{x})$  as the likelihood of motif  $\bar{x}$ , i.e. the probability that  $\bar{x}$  was generated by PWM M

$$L_M(\bar{x}) = P(\bar{x} | M) = \prod_{i \in \{1, \dots, |\bar{x}|\}} M_{\bar{x}_i, i}$$

$\alpha - N \times m \times L$

we generate  $\alpha$  by iterating over  $t = 1, 2, \dots, L$  for  $t = 0$ :

$$\alpha_{i,j,0} = P(y_0^i = j) = \pi_j$$

For  $t \in \{1 \dots L\}$  the table is filled dynamically:

$$\begin{aligned} \alpha_{i,j,t} &= P(y_t^i = j, x_{1:t}^i) = \\ &= \underbrace{\sum_{j' \in \{1, \dots, m\}} \alpha_{i,j',t-1} \cdot (1 - F_{j'}) \cdot T_{j',j} \cdot E_{j,x_{t-o+1}^i, \dots, x_t^i}}_{\text{base-state step}} \\ &+ \underbrace{\sum_{l \in \{1, \dots, k\}} \alpha_{i,j,t-|W_l|-1} \cdot F_j \cdot G_{j,l} \cdot L_{W_l} \left( x_{t-|W_l|}^i, \dots, x_{t-1}^i \right) \cdot E_{j,x_{t-o+1}^i, \dots, x_t^i}}_{\text{sub-state step}} \end{aligned}$$

TODO: how out of range is handled for the PWMs and high-order emission.

#### Backward Algorithm

$\beta - N \times m \times L$

$$\begin{aligned} \beta_{i,j,t} &= P(y_t^i = j, x_{t+1:L}^i) = \\ &= \underbrace{\sum_{u \in \{1, \dots, m\}} (1 - F_j) \cdot T_{j,u} \cdot E_{u,x_{t-o+2}^i, \dots, x_{t+1}^i} \cdot \beta_{i,u,t+1}}_{\text{base-state step}} \\ &+ \underbrace{\sum_{l \in \{1, \dots, k\}} F_j \cdot G_{j,l} \cdot L_{W_l} \left( x_{t+1}^i, \dots, x_{t+|W_l|}^i \right) \cdot E_{j,x_{t-o+|W_l|+2}^i, \dots, x_{t+|W_l|+1}^i} \cdot \beta_{i,j,t+|W_l|+1}}_{\text{sub-state step}} \end{aligned}$$

TODO: how out of range is handled for the PWMs. The problem of peaking before the t when doing a high-order emission of the base-states

### M-Step

First we calculate auxiliary variables:

$$\begin{aligned}\psi_{i,j,l,t} &= P\left(y_t^i = (j, 0), y_{t+1:t+|W_l|}^i = (j, l), y_{t+|W_l|+1}^i = (j, 0), X_i\right) \\ &= \alpha_{i,j,t} \cdot F_j \cdot G_{j,l} \cdot L_{W_l} \left(x_{t+1}^i, \dots, x_{t+|W_l|}^i\right) \cdot E_{j, x_{t+|W_l|-o+2}^i, \dots, x_{t+|W_l|+1}^i} \cdot \beta_{i,j,t+|W_l|+1}\end{aligned}$$

$$\begin{aligned}\gamma_{i,j,t} &= P\left(y_t^i = (j, 0) | X_i\right) = \frac{P\left(y_t^i = (j, 0), X_i\right)}{P(X_i)} \\ &= \frac{\alpha_{i,j,t} \cdot \beta_{i,j,t}}{\sum_{j' \in \{1, \dots, m\}} \left( \alpha_{i,j',t} \cdot \beta_{i,j',t} + \sum_{l \in \{1, \dots, k\}} \sum_{s \in \{1, \dots, |W_l|\}} \psi_{i,j',l,t-s} \right)}\end{aligned}$$

TODO: does different t gives different  $P(X_i) = \sum_{j' \in \{1, \dots, m\}} \alpha_{i,j',t} \cdot \beta_{i,j',t}$ ? Should it?

$$\begin{aligned}\xi_{i,j_1,j_2,t} &= P\left(y_t^i = (j_1, 0), y_{t+1}^i = (j_2, 0) | X_i\right) = \frac{P\left(y_t^i = (j_1, 0), y_{t+1}^i = (j_2, 0), X\right)}{P(X_i)} \\ &= \frac{\alpha_{i,j_1,t} \cdot (1 - F_{j_1}) \cdot T_{j_1,j_2} \cdot E_{j_2, x_{t-o+2}^i, \dots, x_{t+1}^i} \cdot \beta_{i,j_2,t+1}}{\sum_{j'_1, j'_2 \in \{1, \dots, N\}} \alpha_{i,j'_1,t} \cdot (1 - F_{j'_1}) \cdot T_{j'_1,j'_2} \cdot E_{j'_2, x_{t-o+2}^i, \dots, x_{t+1}^i} \cdot \beta_{i,j'_2,t+1} + \sum_{j' \in \{1, \dots, N\}} \sum_{l \in \{1, \dots, k\}} \left( \sum_{s \in \{0, \dots, |W_l|\}} \psi_{i,j',l,t-s} \right)}\end{aligned}$$

TODO: maybe denote a new variable to make above formula nicer?

$$\begin{aligned}\eta_{i,j,l,t} &= P\left(y_t^i = (j, 0), y_{t+1:t+|W_l|}^i = (j, l) | X_i\right) \\ &= \frac{P\left(y_t^i = (j, 0), y_{t+1:t+|W_l|}^i = (j, l), X_i\right)}{P(X_i)} \\ &= \frac{\psi_{i,j,l,t}}{\sum_{j' \in \{1, \dots, m\}} \left( \alpha_{i,j',t} \cdot \beta_{i,j',t} + \sum_{l \in \{1, \dots, k\}} \sum_{s \in \{1, \dots, |W_l|\}} \psi_{i,j',l,t-s} \right)}\end{aligned}$$

We use the temporary auxiliary variables to calculate the  $\theta_{max}$  that maximizes likelihood of the observations.

TODO: add mid steps to the calculation to make more readable

$$\begin{aligned}E_{j,b_1,b_2,\dots,b_o} &= \frac{\sum_{i \in [N]} \sum_{t \in [L]} \gamma_{i,j,t} \cdot \mathbf{1}_{b_1, \dots, b_o}(x_{t-o+1}^i, \dots, x_t^i)}{\sum_{i \in [N]} \sum_{t \in [L]} \gamma_{i,j,t}} \\ T_{j_1,j_2} &= \frac{\sum_{i \in [N]} \sum_{t \in [L]} \xi_{i,j_1,j_2,t}}{\sum_{i \in [N]} \sum_{t \in [L]} \gamma_{i,j_1,t}} \\ F_j &= \frac{\sum_{i \in [N]} \sum_{t \in [L]} \sum_{l \in [k]} \eta_{i,j,l,t}}{\sum_{i \in [N]} \sum_{t \in [L]} \gamma_{i,j,t}}\end{aligned}$$

$$G_{j,l} = \frac{\sum_{i \in [N]} \sum_{t \in [L]} \eta_{i,j,l,t}}{\sum_{i \in [N]} \sum_{t \in [L], l' \in [k]} \eta_{i,j,l',t}}$$

$$\pi_j = \frac{\sum_{i \in [N]} \gamma_{i,j,1}}{\sum_{i \in [N]} \sum_{j' \in [m]} \gamma_{i,j',1}}$$

[Roadmap enhancers preprocessing]  
[training on roadmap data]  
[classification of regulation modules]

## Results

[test accuracy on roadmap enhancers]  
[prediction on roadmap regulation modules]  
[Whole genome classification?]  
[Was HOP-HMM better?]

### Possible Applications

labeled enhancer seqs from multiple motifs-> EM to learn E M F per floor + setting  $T = \mathbb{I}_{m \times m}$  -> posterior of whole genome with sliding window -> classify whole genome  
learn E M F -> check correlation with TF expression  
run EM on whole genome -> posterior of whole genome -> check correlation of posterior to ChIP-Seq of histone modifications  
E M F T-> posterior of whole genome -> see if known critical SNPs are critical in classification

## Discussion