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

December 17, 2018

Abstract

Introduction

[TODO: rewrite this]

The TF inside the nucleus of specific tissues are thought to be a key factor in the activation of specific enhancer. The TFs form a transcription complex and are connected to the enhancer and promoter sequences on top of the TF binding sites (TFBS). Studies using TFBS of TF present in specific cell types are used to classify cell specific enhancer sequences. show heat map of AUC-ROC results. Between these TFBS, k-mer frequencies varies between enhancers and non regulatory“backgound” DNA, and was used to classify enhancers from backgound using only the k-mer distribution (Inbar and tomy, gkm-SVM), and is thought to play a role in spacial properties, nucleosome location and cleavage that cause accessibility of near-by TFBS. Using 44 out of 127 epigenetic data of Roadmap Project to select tissue specific enhancer sequences dataset. In our method, we look for different TFBS and k-mer presence in sequences to classify cell-specific sequences, inside regulatory modules.

Background

[description of enhancers, target genes and DNA folding]

Enhancers are non-coding regulatory DNA sequences that play a key role in the regulation of gene expression. In humans there are hundreds of thousands of enhancers, scattered in 98% of the non-coding regions of the genome, and they are usually between 100-1000 bp long. When activated, the DNA folding draws the enhancer spatially closer to a regulatory element called promoter, resulting in the translation of a gene adjacent to the promoter. The enhancer’s“target gene” (or target genes) is the expressed gene from this activation process. It can be located up to a megabase upstream or downstream from their activating enhancer (or enhancers), and are orientation independent to it.

[TADs]

Hi-C are techniques for measuring the spatial organization of chromatin in the nucleus. These procedures measure the number of occurrences were genome loci are in proximity. In recent years, Hi-C data was generated from multiple organisms (Dixon, J. R. et al. 2012; Rao, S. S. P. et al. 2014; Dixon, J. R. et al. 2015; Sexton, T. et al. 2012), and shown that the genome is partitioned into topologically associating domains (TADs), mostly under 10 mb long (F. Within TADs sequence to sequence interaction are more frequent. TADs boundaries are enriched with DNA-biding protein motifs, and their location play a significant role in gene regulation. Gene-enhancer interactions occur when both are in the same TAD, and deletion of the motifs from TAD’s boundaries may cause changes in the gene activation al. 2015). It has been argued (S. Kadauke 2018) that the TADs are dynamic throughout the cell’s life cycle, and with the change of the TADs structure the enhancer-gene activation couples may vary as well, suggesting the target gene is not predetermined by the the enhancer’s sequence.

[TFs, cofactors and PIC]

TF usually bind to motifs on the DNA, which are short series of nucleotides, sometimes with gaps called TFBS. TFBSs are common in enhancer and promoter elements, and they are good predictors for enhancers location and the type of cell it will be active in.

Multiple studies have shown that genetic alternations in TFBS can affect the expression of its target gene and are a major cause of human diseases (reviewed in I. Miguel-Escalada 2015) [examples: (Soldner F. 2016, Smemo S. 2012, Benko S. 2009, E. S. Emison, 2005, Lettice L.A 2003]. Folding of DNA allows the enhancer-promoter interactions, and recruiting of other cofactor proteins to the already bounded TFs. Some of the recruited cofactors cause nearby chromatin modification and transcription activation. The TFs and cofactors form a transcription preinitiation complex (PIC) which is a very large assembly of proteins (more than one hundred in humans) that recruit the RNA Polymerase (RNA pol II) to invokes the transcription process of the adjacent gene: it opens the double stranded DNA, so that one strand of nucleotides is exposed and becomes a template for RNA synthesis.

[enhancer status, chromatine methylation and accessibility]

Cells of different types and in different operation modes differ by gene expression patterns that are regulated by enhancers and promoters activity patterns. The activity status of enhancers is connected to multiple factors that are detectable on a genome scale.

Chromatin modifications signatures, or“histone marks” are predictive of enhancer position and activity status and can be assessed (Visel et al. 2009; Firbi et al. 2010; Fernandez et al. 2012) . H3k4me1 and H3k27ac are among the predominant histone marks of active enhancers, where H3k4me1 are enriched on transcribed genes and“primed” enhancers prior activation (calo et al 2013), and is thought to precede the H3k27ac modification (Creyghton et al., 2010; Rada-Iglesias et al., 2011; Zentner et al., 2011) which is known to occur during the activation. Other histone marks that are present on active enhancers and are used for their detection are H3k9ac (Ernst et al., 2011; Karmodiya et al., 2012; Krebs et al., 2011; Zentner et al., 2011) and H3K18ac (Jin et al., 2011).
Even though H3k27ac have been identified as an important mark for distinguishing active enhancers from poised enhancers (Creyghton et al. 2010), it is not enough as its own since when present alongside H3k4me3 it is an indication for active promoters [Heintzman et al., 2007]. In contrast, H3k27ac absence and H3k4me1and H3k27me3 enrichment are typical for poised enhancers (Creyghton et al, 2010).

DNA methylation at 5-methycytosine has been involved in genome silencing in multiple processes (Jones, 2012), and has been documented as largely correlated with gene expression inhibition when present in promoters. In enhancer elements, anticorrelation was found between DNA methylation density and enrichment of active enhancer histone marks (Koch and Andrau, 2011; Schmidl et al., 2009; Stadler et al., 2011; Thurman et al., 2012) and TF binding (Neph et al., 2012; Stadler et al., 2011; Thurman et al., 2012), although the cause and consequence relationship underlying these correlations is not yet clear. Since the scenario of TF binding on an enhancer requires an accessible DNA region, DNase I hypersensative sites are used for detecting a potential DNA cleavages that have the potential of being regulatory elements, in usually a better resolution than histone marks.

[inter TFBS sequence, Conservation]

Conserved non-coding elements (CNE) reside in clusters, usually with low gene density but with vicinity to genes. Typically, CNE are structured in arrays called GRB, with a mean length of 1.4 Mb (X. Dong et al. 2009). The correlation between conservation of non-coding region and enhancer functionality is not strong. Some verified enhancers are weakly or not conserved between distant species (M. Friedli et al.2010; J.M. Rosin et al. 2013; L. Taher 2011; D Boffelli, 2004, K. Lindblad-Toh 2011) and some highly conserved areas in the mouse genome are not associated to regulatory activity and their deletion and yielded viable mice (N. Ahituv et al. 2007). Nevertheless, an assay of elements with 100% sequence identity of over 200 bp between human and mouse found that 50% showed enhancers activity in mice (Visel et al., 2007). The reason for such ultra-conservation of 200 bp sequences when the TFBS is only 4-8 bp long is unclear. It is possible that these conserved sequences are actually long assembly of overlapping TFBS or that the enhancer has another function as a eRNA, that the exact nature of its mechanisms is no understood (M. Haeussler 2011, Andersson et al. 2014).

TODO: add a word on conservation of TFBS

[problems with epigenetic data to identify enhancers]

The currently most accurate method for predicting the location of tissue specific enhancers, is analyzing the histone marks and TF and cofactors presence using ChIP-seq from a cell line or from a tissue, combined with DNase I hypersensative (DHS). The main disadvantage of this method is this process is inherently limited to the tissues we can extract and isolate for the epi-genetic examination. Another disadvantage of this method is the need for live cells for the verification of the regulatory activity of a sequence. The persecute for an efficient computational method for predicting the functional nature of sequences“in-silico” has produced positive, yet far from sufficient results in the last years, as reviewed in (D. Kleftogiannis et al. 2016).

[machine learning]

There are several achievements in the task of predicting epi-genetic properties of DNA elements given only their sequence using machine learning algorithms. DeepSEA predicts chromatin modifications given a sequence, from which a regulatory activity could be deduced, and Basset and deltaSVM predicts accessibility. gkm-SVM (Beer et al. 2014) uses gapped kmers as features for an SVM classifier to predict the role of DNA sequences. The disadvantage of these method is their need for a training data of known regulatory elements, which are known mainly from GWAS surveys done on 127 obtained human cell types in the Roadmap and ENCODE projects (Kundaje et al. 2015; Ernst et al. 2011). The number of different cell types in the human body is estimated to be higher than 2200 (Hatano et al. 2011, Diehl et al. 2016), where then number and location of tissue specific enhancers of the rest of the cell types is a mystery.

[HMM background]

Multiple signal processing algorithms have been used in computational biology, and HMM is especially popular among them. Hidden Markov model (HMM) is a statistical model proposed by Leonard Baum (Baum et al. 1966) and is based on the Markov model for modeling regions with alternating frequencies of patterns and symbols.It was used in various engineering fields since the 1980s especially in speech recognition, character recognition and digital communication and was adopted in the computational biology field. The reason HMM is effective in the task of DNA classification is the nucleotide frequencies differences between different types of DNA elements. The Baum-Welch and Viterbi algorithms can detect elements with distinguishable letters distributions without having to train on a labeled dataset. The disadvantage of the model is the lack of consideration for TFBSs or multi-necleotides motifs frequency differences between DNA segments.

[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]

[other stories]

There are other mechanisms and non classic scenarios that involve enhacners: are intergenic enhancers, very long enhancers

TODO: k-mer frequencies varies between enhancers and non regulatory“background” DNA, and was used to classify enhancers from backgound using only the k-mer distribution (Inbar and Tommy),

TODO: [Levin, VISTA] experiments shows that insertion of a sequence without any epigenetic information will activate an enhancer with a near by blue coloring gene. This implies that for the newly introduced sequences to operate as an enhancer and target gene, all that is needed is its sequence in arbitrary location without additional epigenetic information.

From (C.“Probing instructions for expression regulation in gene nucleotide compositions”):

“Several approaches have tackled this problem by modeling gene expression based on epigenetic marks, with the ultimate goal of identifying driving regions and associated genomic variations that are clinically relevant in particular in precision medicine. However, these models rely on experimental data, which are limited to specific samples (even often to cell lines) and cannot be generated for all regulators and all patients. In addition, we show here that, although these approaches are accurate in predicting gene expression, inference of TF combinations from this type of models is not straightforward. Furthermore these methods are not designed to capture regulation instructions present at the sequence level, before the binding of regulators or the opening of the chromatin.”

Methods

[sequence classification, generative models, HMM]

Markov model (A.A. Markov, 1906) is a stochastic model named after Andrey Markov a Russian mathematician. In a Markov model, at any time the system is at one of m states $\{S_1, ..., S_m\}$, where the first state is sampled from a distribution $\pi_i = P(y_1 = S_i)$ and the probability of transitions between the states is denoted by T $T_{i,j} = P(y_t = S_i | y_{t-1} = S_j)$. The system's travel over the states is called a Markov process, and the sequence of states visited in the process is called a Makov chain.

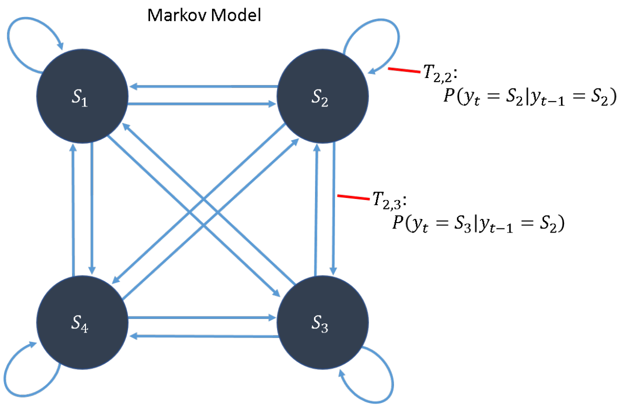
Markov Model
S_1
S_2
S_3
S_4
$T_{2,2}$:
$P(y_t = S_2 | y_{t-1} = S_2)$
$T_{2,3}$:
$P(y_t = S_3 | y_{t-1} = S_2)$

Figure: Markov model with 4 states, marked asS_1, S_2, S_3, S_4 with two of all possible transition probabilities written. T matrix describes the transition distributions, where$T_{i,*}$ is the distribution of the system's next step when inS_i.

Hidden Markov model (HMM) is a system that travels over hidden states in a Markov process, and while doing so it emits variables called observed variables. In this generation process, a single observed variable is emitted per system's step, and so the observed sequence is generated with the same length as the hidden Markov chain. The observed variables$V_1, ..., V_n$ are sampled from an emission distribution E$E_{i,j} = P(x_t = V_j | y_t = S_i)$, that is conditioned on the system's hidden state. Similarly to the Markov model, the distribution to the first hidden state is marked asπ and the transition distribution is marked as$T$.

For example, assuming the DNA are composed of genes enhancers and background regions, with each having different nucleotide frequency, then we can say that the DNA sequence was generated by a HMM with underlying sequence of 4 hidden states: gene, promoter, enhancer and background where each has its own nucleotide frequency. The emitted observed DNA sequence X is determined by the underlying hidden sequence Y that describes the"mode" of the sequence in each position.

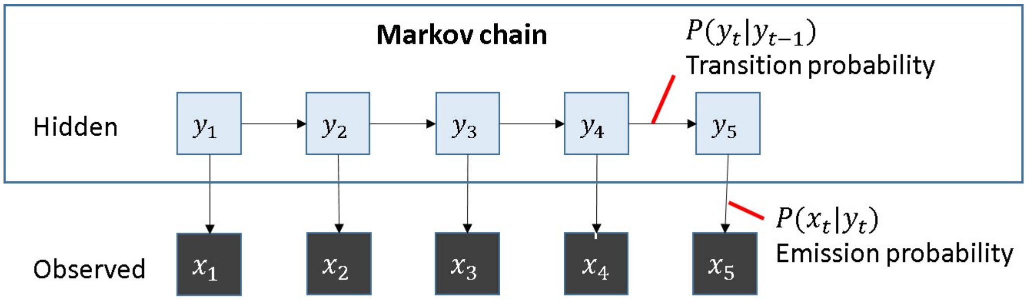

Fig: a HMM chain with length of 5. The underlying hidden states (up) are generated in a Mrokov process and each state emits an observed variable, sampled from the state's emission probability.

Goals

As a generative model, HMM relies on the assumption that the observed DNA sequence$\overrightarrow{X}^L = x_1, ..., x_L$ was generated by a parameterized modelθ, and has an hidden state sequence$\overrightarrow{Y}^L = y_1, ..., y_L$ that was generated alongside it. Given an observed sequence$\overrightarrow{X}^L$, our three goals here are:

1. Finding model parametersθ that fits the most to the observed sequence. Formally, finding theθ that maximizes the likelihood of the observed sequences

$$argmax_{\theta}\mathcal{L}\left(\theta|\overrightarrow{X}^{L}\right) = argmax_{\theta}\sum_{i\in[N]} log\left(P_{\theta}(x_i)\right)$$

2. Using the maximum likelihoodθ, we would like to find the posterior probability of each individual hidden state,$P_{\theta}\left(y_t|X\right)$

3. Using the maximum likelihoodθ, we would like to find the most probable hidden sequence Y,$argmax_{\overrightarrow{Y}^{L}}\mathcal{L}\left(\overrightarrow{Y}^{L}|\overrightarrow{X}^{L};\theta\right)$

We will see that goal 1 and 2 can be achieved by using the Baum-Welch algorithm, named after Leonard E. Baum and Lloyd R. Welch who developed it and the HMM it solves during the 1960s and 1970s. It applies the well known EM algorithm to that infers the maximize the likelihood HMM parameters and then the posterior probability, iteratively. Goal 3 could then be achieved by using the Viterbi algorithm on the posterior probability, named after Andrew Viterbi who proposed it in 1967.

Baum-Welch Algorithm, EM for HMM

In the maximal-likelihood estimation problem before us, we have the observed$X$ and we would ultimately like find the parameters that maximize the likelihood of it,

$$\theta^{*} = argmax_{\theta}L\left(\theta|X\right)$$

This likelihood function$L\left(\theta|X\right)$, also called the incomplete-data likelihood function, could be written, using the total probability law, as :

$$L\left(\theta|X\right) = P(X|\theta) = \sum_{y\in[m]^{N}} P\left(X,y|\theta\right) \tag{1}$$

where the probability on the right side$P(X,Y|\theta) = L(\theta|X,Y)$ is called the complete-data likelihood function. In the case of the basic HMM the complete-data likelihood function is calculated:

$$P(X,y|\theta) = \pi_{y_1}\prod_{i=1}^{N} T_{y_{i-1},y_i}E_{y_i,x_i} \tag{2}$$

Note: while in our variation HOP-HMM (describe in later chapters [TODO: write specific chapter numbers]), we assume different transitions and emissions hence requires different calculation, but otherwise the rest of the EM algorithm holds.

Unfortunately, optimizing or calculating the incomplete-data likelihood in (1) involves a summation of exponential-by-N elements (N is the length of the DNA sequence), which is infeasible. Instead, the strategy of the EM algorithm is to optimize the expected value of the complete-data log-likelihood$log\left(P\left(X,Y|\theta^{'}\right)\right)$ where $\theta^{'}$ is the model's parameters from previous EM iteration (or guessed parameters in the first iteration) and while assuming a fixed observed X, as it is our DNA sequence. For this task we can formally define our target function Q:

$$Q\left(\theta,\theta^{'}\right) = E_{Y}\left[log\left(P\left(X,Y|\theta\right)\right)|X,\theta^{'}\right] = \sum_{y\in[m]^{N}} log\left(P\left(X,y|\theta\right)\right)P\left(X,y|\theta^{'}\right) \tag{3}$$

Every EM iteration is built of two parts: the E-step and the M-step. In the E-step we calculate the Q function and in the M-step we infer theθ that maximizes it. Although this seemingly still requires an exponential summation, we can use a dynamic programming approach to overpass it, with the cost of$O(N\cdot m)$ memory uasge.

Using equations (2) and (3) allows us to split the Q function to three independent parts.

$$
\begin{aligned}
Q\left(\theta,\theta^{'}\right) &= \sum_{y\in[m]^N} log\pi_{y_1}P\left(X,y|\theta^{'}\right) \\
&+ \sum_{y\in[m]^N}\left(\sum_{t\in 2\ldots L} logT_{y_{t-1},y_t}\right)P\left(X,y|\theta^{'}\right) \\
&+ \sum_{y\in[m]^N}\left(\sum_{t\in[L]} logE_{y_t,x_t}\right)P\left(X,y|\theta^{'}\right)
\end{aligned}
$$

then by manipulating the summation and the state sequence cases could be simplified to

$$
\begin{aligned}
Q\left(\theta,\theta^{'}\right) &= \sum_{j\in[m]} log\pi_j P\left(X,y_1=j|\theta^{'}\right) \\
&+ \sum_{t\in 2\ldots L}\sum_{j_1,j_2\in[m]} logT_{j_1,j_2}P\left(X,y_{t-1}=j_1,y_t=j_2|\theta^{'}\right) \\
&+ \sum_{t\in[L]}\sum_{j\in[m]} logE_{j,x_t}P\left(X,y_t=j|\theta^{'}\right)
\end{aligned}
$$

the probabilities occurring in the parts are calculated in the E-step using the forward and backward algorithms. In the M-step each of the three parts is maximized independently using a Lagrange multipliers, under the following probability constrains:

$\sum_{j\in[m]}\pi_j=1$

$\sum_{j_2\in[m]}T_{j_1,j_2}=1$ for all$j_1\in[m]$

$\sum_{b\in[n]}E_{j,b}=1$ for all$j\in[n]$

where m is the number of different hidden states and n is the number of different observed variables (4 in our case of DNA)

The first part is maximized using Lagrange multiplierλ:

$$
\frac{\partial}{\partial\pi_j}\left(\sum_{j\in[m]} log\pi_j P\left(X,y_1=j|\theta^{'}\right)+\lambda\left(\sum_{j\in[m]}\pi_j-1\right)\right)=0
$$

we derive the equations and get $\frac{P\left(X,y_1=j|\theta^{'}\right)}{\pi_j}=-\lambda$ then sum them to receive $\lambda=P\left(X|\theta^{'}\right)$ and assign it and deduce:

$$
\pi_j=\frac{P\left(X,y_1=j|\theta^{'}\right)}{P\left(X|\theta'\right)}=P\left(y_1=j|X,\theta^{'}\right) \tag{4}
$$

we follow this method similarly in the second and third parts, from which we receive:

$$
T_{j_1,j_2}=\frac{\sum_{t\in 2\ldots L}P\left(X,y_{t-1}=j_1,y_t=j_2|\theta^{'}\right)}{\sum_{t\in 2\ldots L}P\left(X,y_{t-1}=j_1|\theta'\right)}=\frac{\sum_{t\in 2\ldots L}P\left(y_{t-1}=j_1,y_t=j_2|X,\theta^{'}\right)}{\sum_{t\in 2\ldots L}P\left(y_{t-1}=j_1|X,\theta'\right)} \tag{5}
$$

and

$$E_{j,b} = \frac{\sum_{t\in[L]} P\left(X, y_t = j|\theta^{'}\right)\mathbf{1}_b(x_t)}{\sum_{t\in[L]} P\left(X, y_t = j|\theta'\right)} = \frac{\sum_{t\in[L]} P\left(y_t = j|X, \theta^{'}\right)\mathbf{1}_b(x_t)}{\sum_{t\in[L]} P\left(y_t = j|X, \theta'\right)} = \tag{6}$$

where$\mathbf{1}_b(x_t) = \begin{cases} 1 & b = x_t \\ 0 & otherwise \end{cases}$

After stating the intentions of each EM iteration in (4), (5) and (6), we now need to calculated them in order to successfully learn the HMM parameters. Specifically, notice that it is enough to calculate the terms $P\left(y_t = j|X, \theta^{'}\right)$ and $P\left(y_{t-1} = j_1, y_t = j_2|X, \theta^{'}\right)$ to resolve all the parameters update states in (4), (5) and (6).

The Forward Backward Algorithm (Rabiner, 1989) is a method to dynamically calculate two matrices, α and β, both of size $m \times L$. The forward probabilities matrix α holds:

$$\alpha_{j,t} = P(y_t = j, x_{1:t})$$

which is, in other words, that $\alpha_{j,t}$ is the probability that a sequence $x_{1:t}$ was emitted and the hidden states series ended with the hidden state j.

The calculation is done by iterating over $t = 1, 2, ..., L$, in each iteration filling all $j \in [m]$ matrix cells as following:

$$\begin{aligned} for\, t = 1: & \\ & \alpha_{j,t} = \pi_j \cdot E_{j,x_1} \\ for\, t = 2, ..., L: & \\ & \alpha_{j,t} = \sum_{j'\in[m]} \alpha_{j',t-1} \cdot T_{j',j} \cdot E_{j,x_t} \end{aligned}$$

The building of the table is based on the HMM basic assumptions that each hidden state y_t is dependent only on the previous one y_{t-1} and that each observed variable x_t is dependent only on its hidden state that emitted it y_t.

$$\alpha_{j,t} = P\left(y_t = j, x_{1:t}\right) = P\left(x_t|y_t = j, x_{1:t-1}\right) \cdot P\left(y_t = j, x_{1:t-1}\right) =$$

$$= P\left(x_t|y_t = j\right) \cdot \sum_{j'\in[m]} P\left(y_t = j, y_{t-1} = j', x_{1:t-1}\right) =$$

$$= P\left(x_t|y_t = j\right) \cdot \sum_{j'\in[m]} P\left(y_t = j|y_{t-1} = j'\right) \cdot P\left(y_{t-1} = j', x_{1:t-1}\right) =$$

$$= E_{j,x_t} \cdot \sum_{j'\in[m]} T_{j',j} \cdot \alpha_{j',t-1}$$

The backwards probabilities matrix β holds:

$$\beta_{j,t} = P\left(x_{t+1:L}|y_t = j\right)$$

which is the probability that a sequence $x_{t+1:L}$ was emitted given the hidden state at position t had value j.

Filling the β is done in the opposite direction $t = L, L-1, ..., 1$, and for all $j \in [m]$ as following:

$$
\begin{aligned}
&for\, t = L :\\
&\qquad \beta_{j,t} = \frac{1}{m}\\
&for\, t = L-1, ..., 1 :\\
&\qquad \beta_{j,t} = \sum_{j' \in [m]} \beta_{j',t+1} \cdot T_{j,j'} \cdot E_{j',x_t}
\end{aligned}
$$

This matrix building process is similarly explained by:

$$\beta_{j,t} = P\left(x_{t+1:L} | y_t = j\right) = \sum_{j' \in [m]} P\left(y_{t+1} = j', x_{t+1:L} | y_t = j\right) =$$

$$= \sum_{j' \in [m]} P\left(x_{t+2:L} | y_t = j\right) \cdot P\left(x_{t+1} | y_t = j, y_{t+1} = j'\right) \cdot P\left(y_{t+1} = j' | y_t = j\right) =$$

$$= \sum_{j' \in [m]} P\left(x_{t+2:L} | y_{t+1} = j'\right) \cdot P\left(x_{t+1} | y_{t+1} = j'\right) \cdot P\left(y_{t+1} = j' | y_t = j\right) =$$

$$= \sum_{j' \in [m]} \beta_{j',t+1} \cdot E_{j',x_{t+1}} \cdot T_{j,j'}$$

[TODO: copy from later parts only the hmm relevant, cover the basic practical parts and then leave only the variation HOPHMM part for the next parts]

###

http://imaging.mrc-cbu.cam.ac.uk/methods/BayesianStuff?action=AttachFile&do=get&target=bilmes-em-algorithm.pdf

Page 11 and page 2

##

We now have all that is needed to calculate the terms $P\left(y_t = j | X, \theta^{'}\right)$, denoted as γ a matrix of size $L \times m$:

$$\gamma_{t,j} = P\left(y_t = j | X, \theta^{'}\right) = \frac{P\left(y_t = j, X | \theta^{'}\right)}{P\left(X | \theta'\right)} =$$

$$= \frac{P\left(y_t = j, x_{1:t}\right) P\left(x_{t+1:L} | y_t = j\right)}{\sum P\left(y_t = j, x_{1:t}\right) P\left(x_{t+1:L} | y_t = j\right)} = \frac{\alpha_{j,t} \cdot \beta_{j,t}}{\sum_{j' \in [m]} \alpha_{j',t} \cdot \beta_{j',t}}$$

HOP-HMM

HOP-HMM models the structure of enhancers containing TFBSs inside them. In this model, there are two types of states: PWM sub state, and base states, where each type of enhancer has its own base state, and the base can transfer into one of it's PWM sub states. Each PWM sub state emits a single motif sampled from its PWM, which is fixed and is given to the model. The base states emit a single letter each time, where the emission is conditional to the previous k letters emitted. This method is useful to express both the lack of frequent long motifs in the inter-TFBS parts of the enhancer, and the motifs that are not in the fixed given set of PWMs of the HOP-HMM model.

A model that contains states that emit motifs sampled from PWMs was previously demonstrated (as described in [T. Kaplan et al 2011]) and as such it is considered a generalized hidden Markov model (gHMM) is a variant of HMM in which states may emit multiple letters.other states that emit letters depending on previous letters in the observed sequence. Similarly to HMM, it assumes an underlying hidden sequence is present that are sampled from a Markov chain.

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

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

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$j \in [m]$ and its l'th sub-state as(j, l) for$l \in \{1, ..., k\}$. 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. DenoteW_l the PWM of the l'th sub-states, which is shared between the l'th sub-states of all base-states.

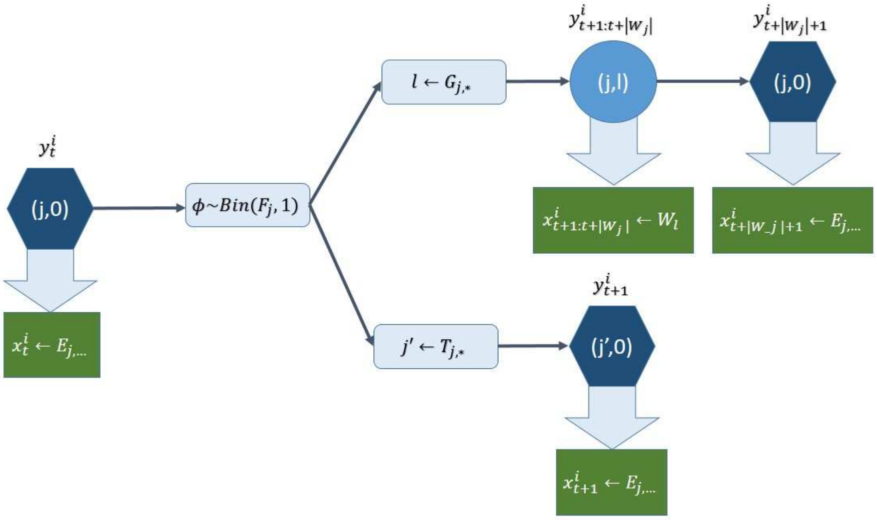

Figure 4: emission and transition process between base-state. The upper flow represent transition from base-state to the same base-state, through a sub-state that emits a motif. The lower flow represent transition between two base-state using the T transition matrix, similar to the conventional HMM.

After emitting a single letter, the j'th base-state has a probabilityF_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.

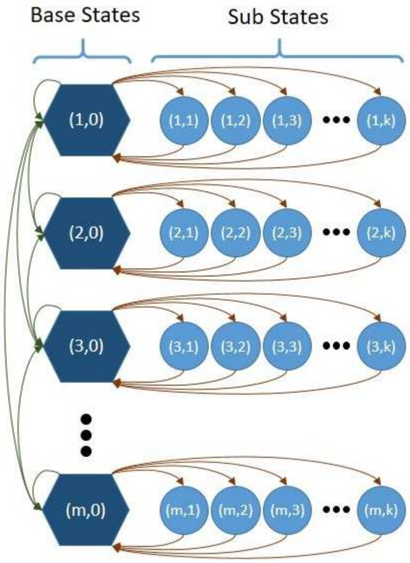

Figure 1: The hidden variable states graph of the HOP-HMM. The left hexagons represent base-states, and the circles in the right part of each row's represent its sub-states.

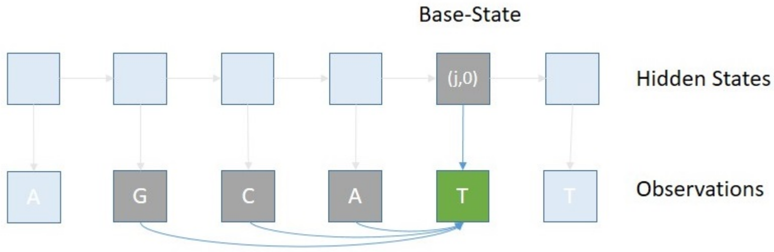

Figure 2: high-order emission of base-states. Each emission is dependent on the hidden base-state and o-1 previous observations.

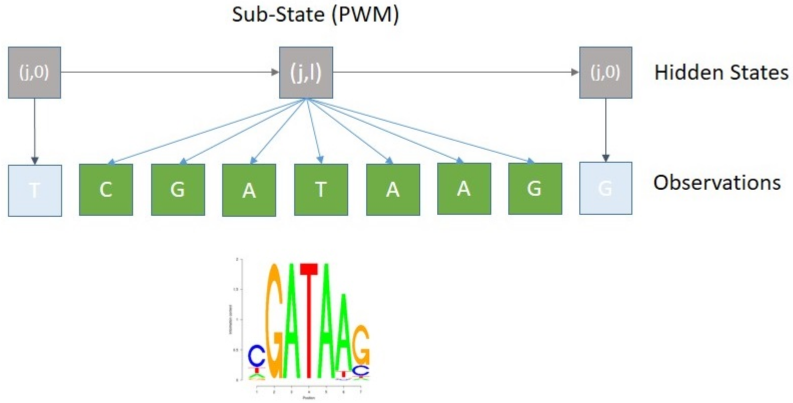

Figure 3: PWM emission of sub-states.

Parameters

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

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

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

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

$$E_{j,x_{t-o+1},x_{t-o+2},...,x_t} = P\left(x_t | y_t = (j,0), x_{t-o+1}, ..., x_{t-1}\right)$$

$$T_{j_1,j_2} = P\left(y_{t+1} = (j_2,0) | y_t = (j_1,0)\right)$$

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

$$G_{j,l} = P\left(y_{t+1:t+|W_l|} = (j,l) | y_t = (j,0)\right)$$

HOP-EM Algorithm

(TODO: where should I put this?)

Denote$L_M(\overline{x})$ as the likelihood of motif$\overline{x}$, i.e. the probability that$\overline{x}$ was generate by PWM M

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

HMM problem:

Given the observation sequence X, and a model$\theta = (\pi, E, T, G)$

How can we compute the probability of the hidden states at each position$P(y_t|X,\theta)$?

How can we calculate$P(X|\theta)$, and what modelθ' maximizes the$P(X|\theta)$?

E-Step: Forward Algorithm

As in the Forward Algorithm (Rabiner, 1989) we calculate the distribution of the t'th hidden-state, considering the observations from the beginning of the sequence until the t'th letter. In this version, it is enough to calculate only the probability of being in the base-states and not the sub-states, i.e.$\alpha_{j,t} = P\left(y_t = (j,0), x_{1:t}\right)$. We calculate$\alpha_{j,t}$ iterating over$t = 1, 2, ..., L$ where the initial for$t = 1$:$\alpha_{j,1} = P\left(y_1 = (j,0), x_1\right) = \pi_j \cdot \sum_{i_1,...,i_{o-1}} E_{j,i_1,...,i_{o-1},x_1}$

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

$$\begin{aligned} \alpha_{j,t} = & P(y_t = (j,0), x_{1:t}) = \\ = & \underbrace{\sum_{j' \in [m]} \alpha_{j',t-1} \cdot T_{j',j} \cdot E_{j,x^i_{t-o+1},...,x^i_t}}_{\text{base-state transitions}} \\ + & \underbrace{\sum_{l \in \{1,...,k\}} \alpha_{j,t-|W_l|-1} \cdot G_{j,l} \cdot L_{W_l}\left(x_{t-|W_l|}, ..., x_{t-1}\right) \cdot E_{j,x_{t-o+1},...,x_t}}_{\text{sub-state transitions}} \end{aligned}$$

Explanation:

From the law of total probability, the probability$\alpha_{j,t}$ is the sum of probabilities of all the possible transition that ended in the base-state (j,0). The possible transitions are base-state to base-state transitions and sub-state to base-state transitions.

$$\alpha_{j,t} = P\left(y_t = (j,0), x_{1:t}\right) =$$

$$= \underbrace{\sum_{j' \in [m]} P\left(y_{t-1} = (j',0), y_t = (j,0), x_{1:t}\right)}_{\text{base-state transitions}} + \underbrace{\sum_{l \in \{1,\dots,k\}} P\left(y_{t-|W_l|:t-1} = (j,l), y_{t-|W_l|-1} = (j,0), x_{1:t}\right)}_{\text{sub-state transitions}}$$

Develop of a sub-state transition using the chain rule:

$$\begin{aligned} P\left(y_{t-|W_l|:t-1} = (j',l), y_{t-|W_l|-1} = (j,0), x_{1:t}\right) = & P\left(y_{t-|W_l|-1} = (j,0), x_{1:t-|W_l|-1}\right) \cdot \\ & \cdot P\left(y_{t-|W_l|:t-1} = (j,l) | y_{t-|W_l|-1} = (j,0), x_{1:t-|W_l|-1}\right) \\ & \cdot P\left(x_{t-|W_l|:t-1} | y_{t-|W_l|:t-1} = (j,l), y_{t-|W_l|-1} = (j,0), x_{1:t-|W_l|-1}\right) \\ & \cdot P\left(x_t | y_t = (j,0), x_{1:t-1}, y_{t-|W_l|:t-1} = (j,l), y_{t-|W_l|-1} = (j,0)\right) \end{aligned}$$

Because ofx_t is dependent on onlyy_t (and also$x_{t-o:t-1}$ ify_t is a base-state) andy_t is dependent on onlyy_{t-1}, we can simplify the probabilities:

$$\begin{aligned} P\left(y_{t-|W_l|:t-1} = (j',l), y_{t-|W_l|-1} = (j,0), x_{1:t}\right) = & P\left(y_{t-|W_l|-1} = (j,0), x_{1:t-|W_l|-1}\right) \\ & \cdot P\left(y_{t-|W_l|:t-1} = (j,l) | y_{t-|W_l|-1} = (j,0)\right) \\ & \cdot P\left(x_{t-|W_l|:t-1} | y_{t-|W_l|:t-1} = (j,l)\right) \\ & \cdot P\left(x_t | y_t = (j,0), x_{t-o:t-1}\right) \end{aligned}$$

We can now replace the received terms with the components ofθ and with already filledα cells:

$$P\left(y^i_{t-|W_l|:t-1} = (j,l), y^i_{t-|W_l|-1} = (j,0), x^i_{1:t}\right) = \alpha_{j,t-|W_l|-1} \cdot G_{j,l} \cdot L_{W_l}\left(x_{t-|W_l|}, \dots, x_{t-1}\right) \cdot E_{j,x_{t-o+1},\dots,x_t}$$

This process is similar to the base-state transition. Using the chain rule:

$$P\left(y_{t-1} = (j',0), y_t = (j,0), x_{1:t}\right) = P\left(y_{t-1} = (j',0), x_{1:t-1}\right) \cdot P\left(y_t = (j,0) | y_{t-1} = (j',0), x_{1:t-1}\right) \cdot P\left(x_t | y_t = (j,0), y_{t-1} = (j',0), x_1\right.$$

Using the conditional independencies to simplify the probabilities:

$$= P(y_{t-1} = (j',0), x_{1:t-1}) \cdot P\left(y_t = (j,0) | y_{t-1} = (j',0)\right) \cdot P\left(x_t | y_t = (j,0), x_{1:t-1}\right) = \alpha_{j',t-1} \cdot T_{j',j} \cdot E_{j,x_{t-o+1},\dots,x_t}$$

Backward Algorithm

$$
\begin{aligned}
\beta_{j,t} =& P\left(x_{t+1:L} | y_t = (j,0)\right) = \\
=& \underbrace{\sum_{j' \in [m]} \beta_{j',t+1} \cdot E_{u,x_{t-o+2},\ldots,x_{t+1}} \cdot T_{j,j'}}_{\text{base-state transitions}} \\
+& \underbrace{\sum_{l \in \{1,\ldots,k\}} \beta_{j,t+|W_l|+1} \cdot L_{W_l}\left(x_{t+1},\ldots,x_{t+|W_l|}\right) \cdot E_{j,x_{t-o+|W_l|+2},\ldots,x_{t+|W_l|+1}} \cdot G_{j,l}}_{\text{sub-state transitions}}
\end{aligned}
$$

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

Law of total probabilty:

$$P\left(x_{t+1:L} | y_t = (j,0)\right) =$$

$$= \underbrace{\sum_{j'} P\left(y_{t+1} = (j',0), x_{t+1:L} | y_t = (j,0)\right)}_{\text{base-state transition}} + \underbrace{\sum_{l} P\left(y_{t+1:t+|W_l|} = (j,l), y_{t+|W_l|+1} = (j,0), x_{t+1:L} | y_t = (j,0)\right)}_{\text{sub-state transition}}$$

For the base-state transition term,using the chain rule:

$$
\begin{aligned}
P\left(y_{t+1} = (j',0), x_{t+1:L} | y_t = (j,0)\right) =& P\left(x_{t+2:L} | y_{t+1} = (j',0), y_t = (j,0), x_{t+1}\right) \\
& \cdot P\left(x_{t+1} | y_{t+1} = (j',0), y_t = (j,0)\right) \\
& \cdot P\left(y_{t+1} = (j',0) | y_t = (j,0)\right)
\end{aligned}
$$

Using the conditional independencies to simplify the probabilities:

$$
\begin{aligned}
P\left(y_{t+1} = (j',0), x_{t+1:L} | y_t = (j,0)\right) =& P\left(x_{t+2:L} | y_{t+1} = (j',0)\right) \\
& \cdot P\left(x_{t+1} | y_{t+1} = (j',0), y_t = (j,0)\right) \\
& \cdot P\left(y_{t+1} = (j',0) | y_t = (j,0)\right) = \\
=& \beta_{j',t+1} \cdot E_{u,x_{t-o+2},\ldots,x_{t+1}} \cdot T_{j,j'}
\end{aligned}
$$

$P\left(y_{t-|W_l|:t-1} = (j',l), y_{t-|W_l|-1} = (j,0), x_{1:t}\right) = P\left(y_{t-|W_l|-1} = (j,0), x_{1:t-|W_l|-1}\right) \cdot \cdot P\left(y_{t-|W_l|:t-1} = (j,l) | y_{t-|W_l|-1} = (j,0), x_{1:t-}\right.$
$P\left(x_{t-|W_l|:t-1} | y_{t-|W_l|:t-1} = (j,l), y_{t-|W_l|-1} = (j,0), x_{1:t-|W_l|-1}\right) \cdot P\left(x_t | y_t = (j,0), x_{1:t-1}, y_{t-|W_l|:t-1} = (j,l), y_{t-|W_l|-1} = (j,0)\right)$

For the sub-state transition term, using the chain rule:

$$
\begin{aligned}
P\left(y_{t+1:t+|W_l|} = (j,l), y_{t+|W_l|+1} = (j,0), x_{t+1:L} | y_t = (j,0)\right) =& P\left(x_{t+|W_l|+2:L} | x_{t+1:t+|W_l|+1}, y_{t+1:t+|W_l|} = (j,l), y_{t+|W_l|+1} = (j,0), y_t\right. \\
& \cdot P\left(x_{t+|W_l|+1} | x_{t+1:t+|W_l|}, y_{t+1:t+|W_l|} = (j,l), y_{t+|W_l|+1} = (j,0), y_t =\right. \\
& \cdot P\left(x_{t+1:t+|W_l|} | y_{t+1:t+|W_l|} = (j,l), y_{t+|W_l|+1} = (j,0), y_t = (j,0)\right) \\
& \cdot P\left(y_{t+1:t+|W_l|} = (j,l), y_{t+|W_l|+1} = (j,0) | y_t = (j,0)\right)
\end{aligned}
$$

Using the conditional independencies to simplify the probabilities:

$$\begin{aligned}P\left(y_{t+1:t+|W_l|}=(j,l),y_{t+|W_l|+1}=(j,0),x_{t+1:L}|y_t=(j,0)\right)=&P\left(x_{t+|W_l|+2:L}|y_{t+|W_l|+1}=(j,0)\right)\\ &\cdot P\left(x_{t+|W_l|+1}|y_{t+|W_l|+1}=(j,0)\right)\\ &\cdot P\left(x_{t+1:t+|W_l|}|y_{t+1:t+|W_l|}=(j,l)\right)\\ &\cdot P\left(y_{t+1:t+|W_l|}=(j,l),y_{t+|W_l|+1}=(j,0)|y_t=(j,0)\right)=\\ =&\beta_{j,t+|W_l|+1}\cdot E_{j,x_{t-o+|W_l|+2},...,x_{t+|W_l|+1}}\cdot L_{W_l}\left(x_{t+1},...,x_{t+|W_l|}\right)\cdot G_{j,l}\end{aligned}$$

M-Step

First we calculate auxiliary variables:

$$\begin{aligned}\psi_{i,j,l,t}=&P\left(y^i_t=(j,0),y^i_{t+1:t+|W_l|}=(j,l),y^i_{t+|W_l|+1}=(j,0),X_i\right)\\ =&\alpha_{i,j,t}\cdot F_j\cdot G_{j,l}\cdot L_{W_{l,}}\left(x^i_{t+1},...,x^i_{t+|W_l|}\right)\cdot E_{j,x^i_{t+|W_l|-o+2},...,x^i_{t+|W_l|+1}}\cdot\beta_{i,j,t+|W_l|+1}\end{aligned}$$

$$\begin{aligned}\gamma_{i,j,t}=&P\left(y^i_t=(j,0)|X_i\right)=\frac{P\left(y^i_t=(j,0),X_i\right)}{P\left(X_i\right)}\\ =&\frac{\alpha_{i,j,t}\cdot\beta_{i,j,t}}{\sum\limits_{j'\in[m]}\left(\alpha_{i,j',t}\cdot\beta_{i,j',t}+\sum\limits_{l\in\{1,...,k\}}\sum\limits_{s\in\{1,...,|W_l|\}}\psi_{i,j',l,t-s}\right)}\end{aligned}$$

$$P\left(y_t=(j,0),x_{1:L}\right)=P\left(y_t=(j,0),x_{1:t}\right)\cdot P\left(x_{t+1:L}|x_{1:t},y_t=(j,0)\right)\approx P\left(y_t=(j,0),x_{1:t}\right)\cdot P\left(x_{t+1:L}|y_t=(j,0)\right)=\alpha_{j.t}\cdot\beta_{j,t}$$

$$\approx=$$

$$P\left(x_{1:L}\right)=\sum_{j\in[m]}\alpha_{j,1}\cdot\beta_{j,1}=\sum_{j\in[m]}P\left(y_1=(j,0)|x_1\right)\cdot P\left(x_{2:L}|y_1=(j,0)\right)=\sum_{j\in[m]}\tfrac{P(y_1=(j,0),x_1)}{P(x_1)}\cdot\tfrac{P(y_1=(j,0),x_{2:L})}{P(y_1=(j,0))}=$$

TODO: does different t gives different$P\left(x_{1:L}\right)=\sum\limits_{j'\in[m]}\alpha_{i,j',t}\cdot\beta_{i,j',t}$? Should it?

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

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

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

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

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

$$E_{j,b_1,b_2,\ldots,b_o} = \frac{\sum\limits_{i\in[N]\,t\in[L]}\gamma_{i,j,t}\cdot\mathbf{1}_{b_1,\ldots,b_o}(x_{t-o+1}^i,\ldots,x_t^i)}{\sum\limits_{i\in[N]\,t\in[L]}\gamma_{i,j,t}}$$

$$T_{j_1,j_2} = \frac{\sum\limits_{i\in[N]\,t\in[L]}\xi_{i,j_1,j_2,t}}{\sum\limits_{i\in[N]\,t\in[L]}\gamma_{i,j_1,t}}$$

$$F_j = \frac{\sum\limits_{i\in[N]\,t\in[L]\,l\in[k]}\eta_{i,j,l,t}}{\sum\limits_{i\in[N]\,t\in[L]}\gamma_{i,j,t}}$$

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

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

[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

References