High-Order Generalized Hidden Markov Model for DNA Regulatory Sequences Classification

September 13, 2019

Background

The Genome

The genome of every organism contains the inherited information that defines its complex structure and function. The genome is built out of Deoxyribonucleic acid (DNA) molecule, that is a built out of two chains of nucleotides units that form a double helix shape. Each nucleotide is built our of 4 different types bases: cytosine, guanine, adenine or thymine or in short A,C,G and T. The nucleotides are organized in pairs called base pairs where each of the paired nucleotides are complimentary to each other and provide redundancy.

Proteins are macromolecoles, which carry various roles and functions within organisms. They are built out of 20 different amino acids, which order and structure is encoded inside genetic segments in the genome called genes. Through the transcription and translation processes, the genes are expressed and result in the formation of proteins. In the transcription process the gene is read and transcribed into a single strand sequence of RNA. Later, the RNA molecules are translated into a sequence of amino acids that constitute a protein.

Genes

Gene sequences are built out fragmented introns and exons, where only the exons becomes the RNA molecules that translates into proteins while the introns are spliced away beforehand. Although the exons alone hold the recipe for the construction of the organism's proteins, the complexity of the organism is not a product of their number or their length. For example, the humans and Caenorhabditis elegans roundworms both have about 19,000 genes (Ezkurdia et al., 2014; Ainscough et al., 1998), with roughly the same total exon length and number, although the human body is vastly more diverse and complex. The source for the organisms complexity differences is attributed to the gene regulation mechanism. The human genome is 3.23 Gb long, and it is estimated that gene regulation regions involve 10-20% of it (Pennacchio et al., 2015), compared to exon regions that involve only 1% (Ng et al., 2009).

Enhancers

Enhancers are are non-coding regulatory DNA sequences that play a key role in the regulation transcription of genes. In humans there are hundreds of thousands of enhancers, scattered over the non-coding regions of the genome, and their length are usually between 100-1000 bp. When activated, the DNA folding draws the enhancer spatially closer to another type of regulatory element called promoter, resulting in the translation of a gene adjacent to the promoter (see figure 1). The enhancer's target gene is the expressed gene from this activation process. It can be located up to a megabase upstream or downstream from their activating enhancer (Williamson et al., 2011), and are orientation independent to it. Moreover, the gene-enhancer connection is not exclusive, and the common case is that each enhancer has several target genes and vice versa (Fishilevich et al., 2017).

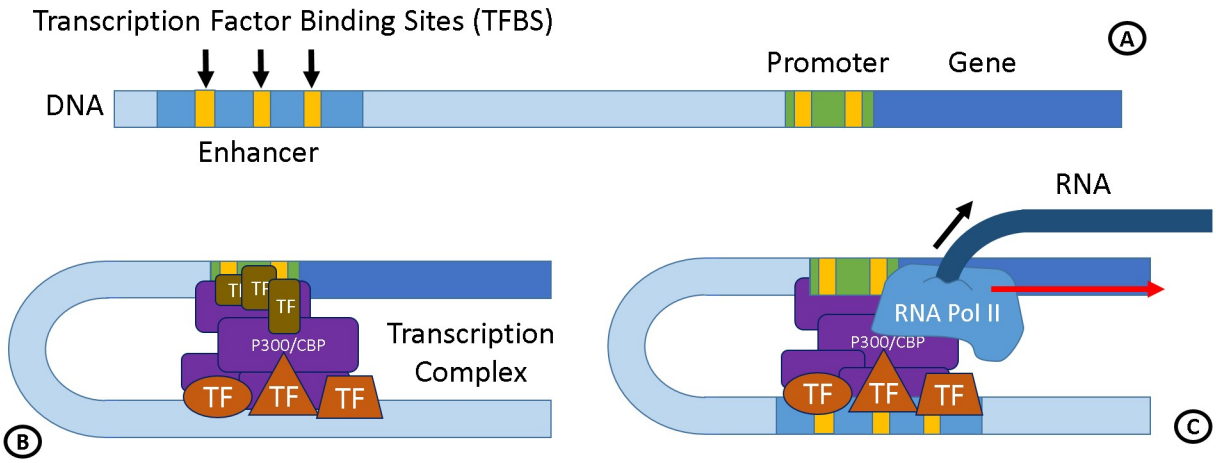


Figure: A) An enhancer and its distal target gene. B) The DNA folds and the attached with transcription factors draw other co-factor proteins that together form the transcription complex. C) The RNA Polymerase II is recruited and while moving along the gene it generates a new RNA molecule that is transcribed off the gene sequence.

In VISTA Project (Visel et al., 2007), mouse fertilized eggs where injected enhancers sequences, adjacent to LacZ reporter gene, encoding enzyme with blue color. The injected DNA sequences bared no epigenetic information and integrated in an arbitrary position in the mouse genome. The transgenic embryos where photographed after 11.5 days and, for some of the DNA sequences, a similar pattern was present over several instances. These results imply that for many DNA sequences, the DNA code alone possess the potential to become a tissue specific enhancer, even without epigenetic information.

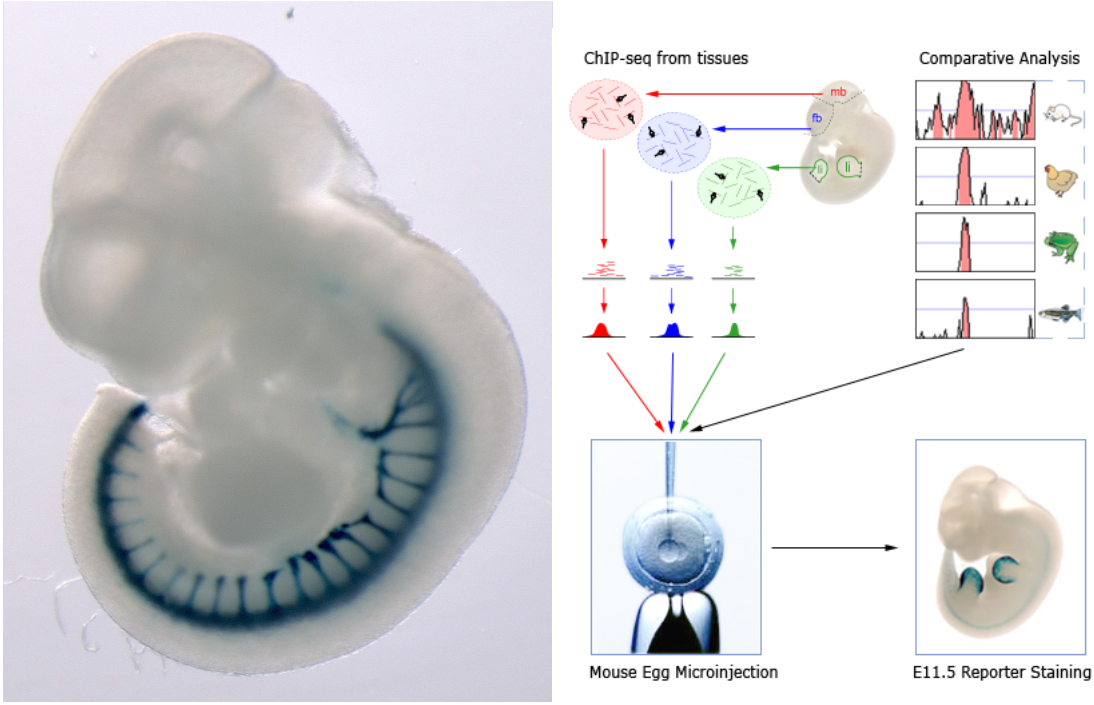


Figure: Transgenic mouse embryo in the 11.5 day. As an fertilized egg a synthetic enhancer sequence was injected, which is related to the dorsal root ganglia spinal neurons. The enhancer became activated and caused the expression of the blue color marker gene that was coupled to it. Taken from Vista Enhancer Browser, experiment hs-51 embryo 2.

Transcription Factor Binding

Transcription factors (TF) are proteins that bind to the DNA, and together with other co-factor proteins initiate the gene transcription process. TFs tend to bind to certain transcription factor binding sites (TFBS), which are motifs of nucleotides on

the DNA with average length of 12 bp in humans (Kulakovskiy et al., 2011) that are conserved between species (Doniger et al., 2005). On genome-wide association studies (GWAS) done with ChIP-seq method, different TFs have different distributions of TFBS they are observed attached to (Khan et al., JASPA, 2018).

Both enhancers and promoters contain TFBSs that are critical for the their correct regulatory operation. Multiple studies have shown that genetic alternations in TFBS can affect the expression of the regulated gene and are a major cause of different human diseases (Kreimer et al., 2017; Miguel-Escalada et al., 2015; Soldner et al., 2016; Smemo S., 2012; Benko et al., 2009; Emison et al., 2005; Lettice et al., 2003). From the sequence aspect, enhancers and promoters have a similar structure of a background nucleotide sequence with distribution different from other part of the genome, with TFBS motifs tiled inside this background sequence.

The enrichment of TFBS is a good predictor for the location of promoter and enhancer regulatory regions and the type of cells they will be active in. Folding of DNA allows the enhancer-promoter interactions, in which the TFs take major part. Once bounded to the DNA, the TFs recruit other cofactor proteins to them, and together they form a transcription preinitiation complex (PIC), a very large assembly of proteins. Out of the tens of proteins constructing the PIC, the sub-unit RNA Polymerase (RNA pol II) has the role of transcribing the adjacent gene. it opens the double stranded DNA, so that one strand of nucleotides is exposed and becomes a template for RNA synthesis.

PWMs

Generating a compact model for estimating the binding potential of a DNA sequence to a TF, i.e.$P(x_{1:n}|binding)$, is not trivial as might seem on first look. The peaks of the ChIP-seq data are used as the ground truth of TF binding locations, from which the model is built. Position weight matrix (PWM) is a commonly used simplistic method to address this task. The underlying assumption of the PWM model is that every position in the DNA sequence has an independent probability to attach to the TF, and therefore the total binding probability is a multiplication of all the per-position probabilities in the motif:

$$P(x_{1:n}|binding) = \prod_{i\in[n]} P(x_i|binding)$$

Where n is the size of relevant sequence. The size of the sequence that is affected by the binding event is derived from the physical characteristics of the TF.

$P(x_i|binding)$ is estimated by counting the nucleotides frequency in every position of the observed binding sites, which are the ChIP-seq peaks. For a motif of length J, this probability estimation is stored in a PWM matrix W as followed:$W_{i,j} = \frac{1}{N}\sum_{k\in N} 1(X_{i,k} = j)$ where$i \in [J]$ the position in the motif and$j \in [4]$ the nucleotide index of A,C,G and T.

From a generative model point of view, the sequence is generated by a TFBS motifs emission system. For this needs, the log of the matrix often comes handy for calculation of$log\,(L(W;x_{1:n}))$, the log of the probability that a motif was generated by a PWM W. This calculation is done be a convolution of$log(W)$ on a one-hot encoding of the sequence.

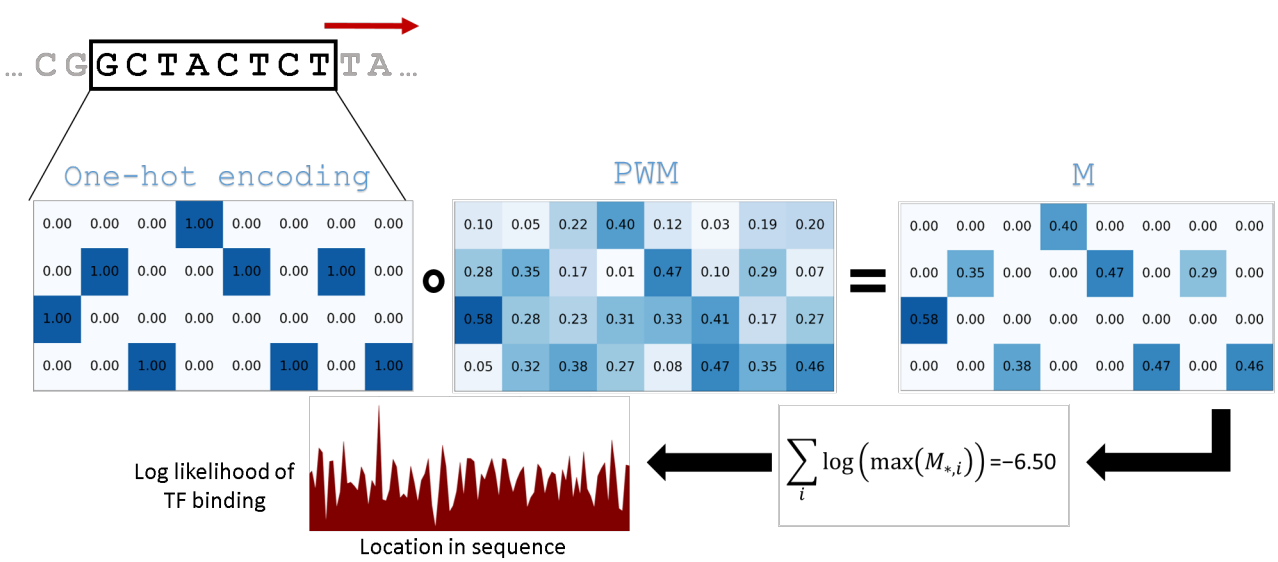


Figure 2: a sub-sequence out of the DNA is represented in a one-hot encoding, then entry-wise multiplied with the PWM. Then, the sum of the logs of the maximal values of each column in the result matrix is calculated, which is the log likelihood of the TF binding to the sub-sequence. This log likelihood is calculated for each location in the sequence, where location with high values indicate high likelihood of TF binding.

Inter TFBS Sequences and 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 known as genomic regulatory block (GRB), with a mean length of 1.4 Mb (Akalin 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 species (Friedli et al., 2010; Rosin et al., 2013; Taher et al., 2011; Lindblad-Toh et al., 2011) and some highly conserved areas in the mouse genome are not associated to regulatory activity and their deletion and yielded viable mice (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 (Kim et al., 2010; Andersson et al., 2014).

Epigenetics

Almost all cells in every organism contain its genome, but only part of genome is active in any specific cell. Cells of different types and in different operation modes differ by gene expression patterns. The reason for that lies in regulation components that are outside of the genomic sequence. The location and presence of TFBS, background nucleotides distribution and other sequence related properties are not enough to explain regulatory role of regions in the genome.

Several epigenetic features (which do not involve the nucleotides sequence directly) correlate with enhancer regions in the genome:

- Accessibility
- TF & cofactors binding
- Histone modifications
- DNA methylation

These properties and mechanisms have measurable features that lie on top of the genome. Their combination is the main source of identification for enhancer regions in the genome. Each cell has its own epigenetic features, in a binaric form, e.g. a specific part of the genome can be either accessible, or not. When several similar cells from the same tissue sample are measured, a frequency or count of the feature is measured per DNA loci, and generates epigenetic data. The epigenetic data is commonly used as the ground truth indication for enhancer sequences, as done for the human genome in the ENCODE project.

Accessibility

In eukaryotes, the DNA is packed around a structure of 8 histone proteins, together forming a nucleosome core. The location of the nucleosome binding is not random over the DNA sequence, but has a tendency for specific DNA binding sites (Cutter et al., 2015). DNA that is wrapped around a nucleosome has a lesser probability to interact with proteins, as it is physically inaccessible. Both the enhancer, the promoter and the gene need to be accessible for a successful transcription.

Since the scenario of TF binding on an enhancer requires an accessible DNA region, 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.

Histone Marks

Chromatin modifications signatures, also called histone marks, are predictive of enhancer position and activity status (Visel et al., 2007; Heintzman et al., 2009; Fernandez et al., 2012). The histone marks are considered to contain a certain"histone code" which encode complex information, additionally to the DNA, regarding the transcription regulation and other aspects. Comparing to other epigenetic information, and especially DNA methylation (Przybilla et al., 2012), chromatin modifications have a short time-scale of seconds or hours (Hayashi-Takanaka et al., 2011), hence they are considered part of the dynamic changes of the cell's modes.

H3K4me1 and H3K27ac are among the predominant histone marks of active enhancers, where H3K4me1 are enriched on transcribed genes and enhancers prior to 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; Karmodiya et al., 2012; 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).

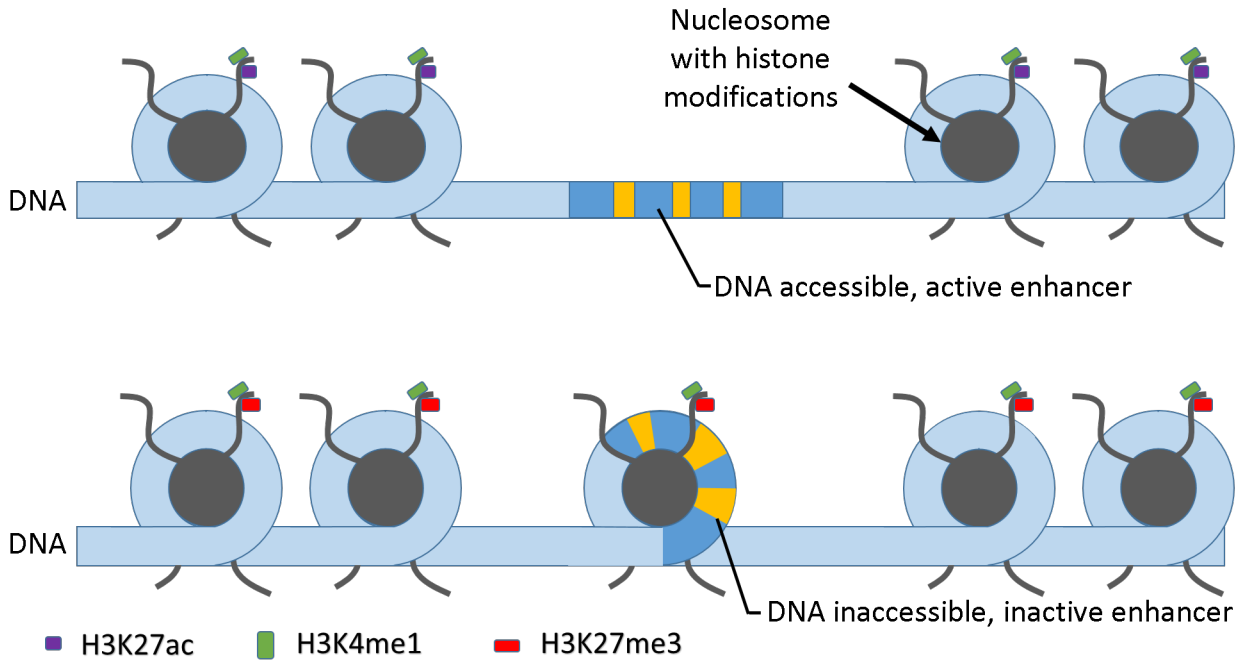


Figure 3: The accessibility of the enhancer's sequence and its surrounding histone marks are connected to its regulatory activity state. On the upper part an active enhancer sequence that is accessible for protein interaction needed for transcription, where as on the lower part an inactive enhancer is inaccessible since it is wrapped around a nucleosome.

DNA Methylation

DNA methylation at cytosine and CpG sites has been involved in genome silencing in multiple processes (Jones et al., 2012), and has been documented as largely correlated with gene expression inhibition when present in promoters. In enhancer elements, anti-correlation was found between DNA methylation density and enrichment of active enhancer histone marks and TF binding (Stadler et al., 2011; Thurman et al., 2012), although the cause and consequence relationship underlying these correlations is not yet clear.

Epigenetics Limitation

The currently most accurate method for predicting the location of tissue specific enhancers in a genome wide scale, 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).

Several approaches have faced the problem of locating enhancers by modeling gene expression based on epigenetic marks. However, these models rely on experimental data, and are inherently limited to the specific tissues we can extract and isolate for epigenetic examination. Furthermore, such models do no supply a classification of enhancers for new variation found in the population. Another disadvantage is the need for live cells for the verification of the regulatory activity of a sequence. The ultimate goal of an efficient computational method for predicting and explaining the reason for the functional nature of sequences"in-silico" has produced positive, yet far from sufficient results in the last years, as reviewed in (Kleftogiannis et al., 2016).

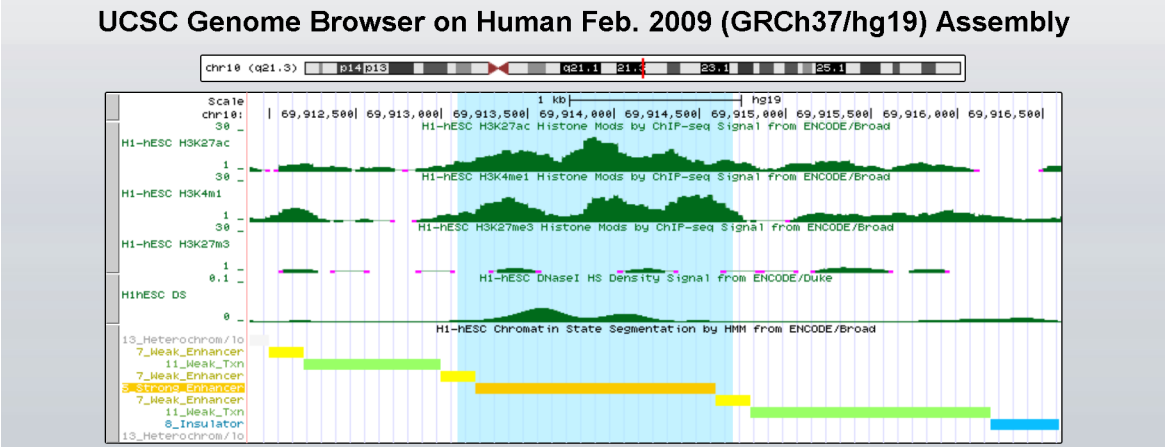


Figure 4: Epigenetic feature tracks measured by ENCODE, taken from the tenth chromosome of a H1-hESC cell line. Highlighted in light blue, the peaks of the H3K27ac (1st green plot) and H3K4me1 (2nd green plot) histone marks and the DNaseI hyper sensitivity feature (4th green plot) together with the lack of H3K27me3 (3rd green plot) signal are indication of an active enhancer, as indicated by the ChromHMM classification (bottom). Note the decrease between the two peaks of H3K27ac and H3K4me1 is located on top of the increase of the DNaseI hyper sensitivity, which implies a cleavage in between two nucleosomes with modifications.

Previous Work

There are several achievements in the task of predicting epigenetic and regulatory properties of DNA elements given only their sequence using machine learning algorithms. DeepSEA (Zhou and Troyanskaya, 2015) deep convolutional neural network (DCNN) is fed with 1000 bp DNA sequence and predicts an output vector of 919 binary features which represents the chromatin modifications of 200 bp bin in the center of the input sequence. The training labels used are the chromatin modification are extracted from ENCODE and Roadmap Epigenomics data releases.

Basset (Khan et al., JASPA, 2018) also used DCNN with known PWM as weights initialization on ENCODE and Roadmap Epigenomics data to predict a binary vector that represents accessibility in 164 cell types based on 600 bp DNA sequence. In DeepBind (Alipanahi et al., 2015) a DCNN was used to predict binding of 538 TFs and 194 RNA binding proteins from DNA sequences of varying lengths. In gkm-SVM (Beer et al., 2014), gapped kmers presence indicator vector were used as features for an SVM classifier to predict the role of DNA sequences with varying lengths.

ChromHMM (Ernst and Kellis, 2012) is a widely used software that tackles the problem of analyzing the epigenetic data for concluding roles in the genomic sequence. The algorithm uses chromatin mark reads, threshold to binary values, as input to HMM which then allows classifying the genome state in each position in the genome.

A disadvantage of these method is their need for a training data of known regulatory elements or with epigenetic data, which is commonly obtained 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 (Diehl et al., 2016), and so we cannot know in certainty the number and location of tissue specific enhancers active in most of these cell types.

HOP-HMM extends the algorithm of Kaplan et al., 2011 (Kaplan et al., 2012), which also contains hidden states that emit TFBS sampled from PWMs to predict enhancers location in the genome. Both algorithms are part of the generalized hidden Markov model (gHMM) family, which are HMM variants that contain hidden states that may emit multiple observed variables. Although HMM variants using higher-order HMM, in which the transition and emission are dependent on previous hidden states where used previously (Ferguson, 1980; Preez, 1998), an HMM variant in which the emission is dependent on previous emissions is a less researched field.

Data Representation

The DNA sequence, when read from cells, is usually stored in files, such as .fa, as a sequence of letters A,C,G and T. For an algorithm to process it, the characters are mapped into integers 1,2,3 and 4 respectively. For many algorithms, such as in DeepSEA, Basset, and our HOP-Baum-Welch, it is more suitable to represent the DNA sequence in a one-hot encoding as described in figure 3.

A common feature extraction technique is representing a DNA sequence as a vector of the in-sequence frequencies of all the possible kmer as used in gkm-SVM. In this technique, similarly to the bag of words technique in text analysis and natural language processing, the order of the kmer locations is sacrificed for a more meaning-oriented, structured and fixed-length data encoding.

Research Question

The main mission of this research is to locate genomic regions with potential enhancer activity, both in cell types from which we have epigenetic data and from cell types we don't.

It has been shown in vivo (Visel et al., 2007) that the insertion of an enhancer sequence and an adjacent target gene will often cause an activation of the target gene in mice. The transgenic mice shows the activation of the inserted enhancer although it was inserted to an arbitrary location in the genome and without epigenetic information. This strengthen the possibility that classifying sequences as enhancers could be achieved even without the epigenetic information, which is in missing for many types of cells.

TODO: remove duplication from background. add: the conclusion\ hypothesis raised by the mouse experiment is that the info is in the DNA, not the epigenetics. if the hypothesis is true, it raises the research question, can we recognize enhancer from the sequence? explain why HMM - the structure\ composition of the enhancer.

Markov Models

Markov model (Markov, 1906) is a stochastic model named after Andrey Markov, a Russian mathematician. In a Markov model, at any time the model 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$T_{i,j} = P(y_t = S_i | y_{t-1} = S_j)$. The model's travel over the states is called a Markov process, and the sequence of states visited in the process is called a Markov chain.

The likelihood of a Markov chain X generated by a Markov Model$\theta = \{\pi, T\}$ is a joint probability of the first state and all following transition, which due to the independence between transition events can be written as :

$$L(\theta;X) = P_\theta(x_0,x_1,...,x_L) = \pi_{x_0} \cdot T_{x_0,x_1} \cdot T_{x_1,x_2} \cdot ... \cdot T_{x_{L-1},x_L}$$

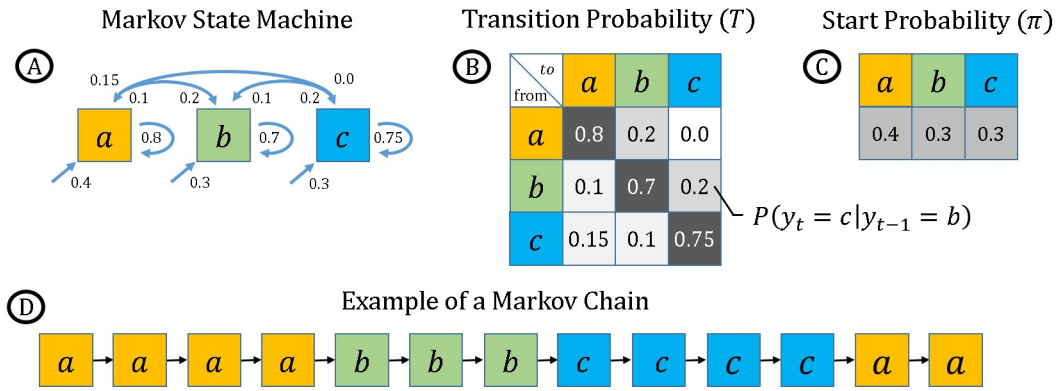


Figure 6: A) Markov model with 3 states (yellow green and blue). B,C) The model starts with a state sampled from π, and travels between the states with a transition distribution T. D) The model can generate Markov chains of states, where the transition between the states is conditioned on the previous state alone, causing the Markov process to be memoryless.

Generative & Discriminative Models

There are a two main distinguished approaches in the machine learning models, the generative models and discriminative models. Both assumes an observed variables X and target variables Y, also commonly referred to as labels.

- The generative models assume a joint probability $P(X,Y)$. Using the data one can estimate the distribution $P(X,Y)$, then from it estimate $P(Y|X)$. It is assumed that such a model can generate the random instances of the data either as pairs of (x,y) or generate instances of x given y.
- Discriminative models assume conditional probability $P(Y|X)$, which is estimated directly from the data.

In classification problems, the task at hand is to arrive from the observed X to its label Y, e.g. given a DNA sequence X, deciding its role label Y. Both models eventually use the $P(Y|X)$ estimation to base their classification. Namely, classifying a data sample x by $y_{est} = argmax_y P(Y = y|X = x)$.

Discriminative models are more widely used than generative models. They are often easier to use and build since they require less assumptions on the origin or generation of the data. For example, a discriminative model such as a DNN classifying the role of DNA sequence assumes very little on the way the DNA sequence is related to it's role and generated based on it, but instead it finds features in the sequence that indicate its role. Such a model often gives very little for later understanding of the nature of the data generation process, and can generate no new data later for other uses.

Hidden Markov model

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 extensively in various engineering fields since the 1980s, especially in speech recognition (Rabiner & Juang, 1993), handwriting recognition (Hu et al., 1996) and digital communication (Turin and Sondhi, 1993) and was adopted in the computational biology field.

Hidden Markov model (HMM) is a model that travels over hidden states in a Markov process, and while doing so it emits variables called observed variables. As the Markov model, HMM is an generative model and it assumes the existence of a joint probability $P(x_{1:L}, y_{1:L})$ that is derived from the compact parameters θ. As a generative model, HMM relies on the assumption that the observed DNA sequence $X = x_1, ..., x_L$ can be generated by a parameterized model θ, and has an hidden state sequence $Y = y_1, ..., y_L$ that are generated alongside it. In this generation process, a single observed variable is emitted per step of the model, 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 $E_{i,j} = P(x_t = V_j|y_t = S_i)$, that is conditioned on the hidden state of the model. Similarly to the Markov model, the distribution to the first hidden state is marked as π and the transition distribution is marked as T.

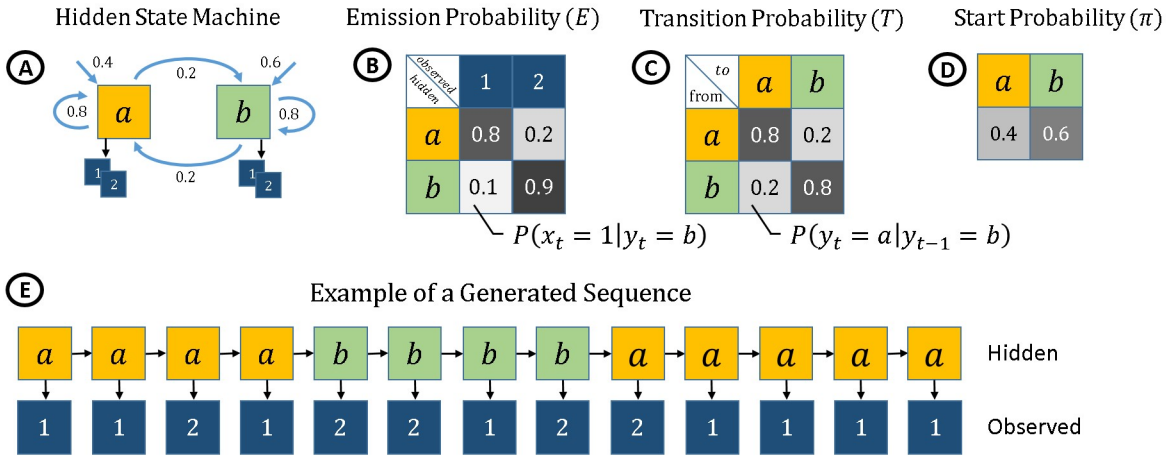


Figure 7: A) a HMM with 2 hidden states. B) The observed variables (dark blue) are emitted by the hidden state at their location, sampled from the discrete conditional distribution E. C,D) The hidden states (yellow and green) behave as Markov model states with starting and transition probabilities,π and$T$. E) The output of the model is a observable sequence with an underlying hidden sequence. The hidden sequence is a Markov chain, where on each step the hidden state emits a single observed variable.

In the case where the observable sequence is made out of DNA, we can assume the DNA sequence is composed out of 4 types of regions: genes, promoter enhancers and background regions. Each of these types will have different nucleotide frequency, and we assume the DNA sequence was generated by a HMM with underlying sequence of 4 hidden states, one for each region type. The emitted observed DNA sequence x is determined by the underlying hidden sequence y that describes the"mode" of the sequence in each position.

HMM Likelihood and Posterior Probability

Having a HMM withθ on hand, two questions arise given an observed sequence x:

- What is the likelihood that x was generated by the HMM?$P_\theta(x_{1:L})$
- What is the probability of a hidden state at every location?$P_\theta(y_t = j|x_{1:L})$

The two probabilities in the questions above are the likelihood and the posterior probabilities of HMM.

As in many generative models, HMM's likelihood function$\mathscr{L}(\theta|x_{1:L})$ which we want to compute can be split by the total probability law to the sum of all possible hidden sequences:

$$\mathscr{L}(\theta|x_{1:L}) = P_\theta(x_{1:L}) = \sum_{y_{1:L}\in[m]^L} P_\theta(x_{1:L}, y_{1:L}) \tag{1}$$

The probability$P_\theta(x_{1:L})$ is called the incomplete-data likelihood function and the probability$P_\theta(x_{1:L}, y_{1:L})$ is called the complete-data likelihood function. In the case of HMM with parametersθ, the complete-data can be calculated by:

$$P_\theta(x_{1:L}, y_{1:L}) = P_\theta(y_1)\cdot P_\theta(x_1|y_1)\cdot\prod_{i=2}^{N} P_\theta(y_i|y_{i-1})\cdot P_\theta(x_i|y_i) = \pi_{y_1}E_{y_1,x_1}\prod_{i=2}^{L} T_{y_{i-1},y_i}E_{y_i,x_i} \tag{2}$$

Although the complete-data likelihood computation is linear-by-L, computing the incomplete-data with all possible hidden sequences would be exponential-by-L and is not efficient.

An alternative method, which answers both the likelihood and the posterior questions above, is a method that utilizes the Markovian memoryless feature of HMM. It is the Forward-Backward algorithm (Rabiner, 1989), a method that dynamically calculates two matrices,α andβ, both of size$m \times L$. The values ofα andβ hold:

Algorithm 1 Forward Algorithm

Input:

$x_{1:L}$ - Observed DNA sequence

Algorithm:

$for\, j = [1, ..., m]:$

$\alpha_{j,1} = \pi_j \cdot E_{j,x_1}$

$fort = [2, ..., L]:$

$for\, j = [1, ..., m]:$

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

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

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

Forward Algorithm

The forward probabilities matrixα holds the probability that a sequence$x_{1:t}$ was emitted and that the hidden sequence ended with j:

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

It is calculated the dynamic algorithm:

The building of the table is based on the HMM basic assumptions that each hidden statey_t is dependent only on the previous oney_{t-1} and that each observed variablex_t is dependent only on its hidden state that emitted ity_t.

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

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

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

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

Backwards Algorithm

The backwards probabilities matrixβ holds the probability that a sequence$x_{t+1:L}$ was emitted given the hidden state at position t had value j:

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

It is calculated by the dynamic algorithm:

This matrix building process is similarly explained by:

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

Algorithm 2 Backward Algorithm

Input:
X - Observed DNA sequence
Algorithm:
$\beta_{1:m,L} = 1$
$for\, t = [L-1, ..., 1]:$
$for\, j = [1, ..., m]:$
$\beta_{j,t} = \sum_{j' \in [m]} \beta_{j',t+1} \cdot T_{j,j'} \cdot E_{j',x_t}$

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

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

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

Once we obtainα andβ probabilities, the incomplete-data likelihood of HMM can be linearly calculated:

$$P_\theta\left(x_{1:L}\right) = \sum_{j \in [m]} P_\theta\left(y_t = j, x_{1:t}\right) = \sum_{j \in [m]} \alpha_{j,L}$$

And now the posterior probability can be computed:

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

HMM Limitations Although HMM is simple and efficient, applying it on DNA sequences has a major disadvantage which is the inherit Markovian lack-of-memory property. This property means that on every step of the model, the next state is dependent only on the previous state, without further history consideration. For the task of emitting a motif, where each position has a different emission distribution depending on the location in the motif, a HMM model would need to contain different hidden states per position in the motif. This means that for an HMM to be able to emit even a small number of short motifs, it needs to hold a large number of states that require learning a large number of parameters, e.g. for the ability to emit 50 motifs of length 5, an HMM needs to have over 60,000 parameters. Furthermore, the enhancer modeling task at hand is even more complex, since we would like to model multiple enhancers and backgrounds states, each having different probability of emitting motifs and unique k-order emission distribution when not in those motifs. For our data structure prior assumption the required number of model's parameters would have been about10^7, large enough to introduce problems such as unfeasible memory complexity and overfitting.

A common way to avoid overfitting the data when training machine learning models is reducing the model's complexity by fixing some of its parameters. Our proposed HOP-HMM addresses both the memory issue and the overfitting issue while remaining equivalent to a regularized HMM with a large number of states with fixed parameters. Namely, most of the transition probabilities are fixed to zero and therefore never stored in memory, and some of the emission probabilities are predetermined and are fixed during the training. This allows us to learn a model with the enhancer prior assumptions of motifs and high-order emission without overfitting, and with reasonable memory complexity.

Higher Order HMM

A way that was previously used to increase the complexity of HMM in the aspect of higher conditioning order is by making the transition and emission dependent on previous hidden states (Mari et al., 1997; Preez, 1998; Lee and Lee, 2006). Although these HMM variants are capable of expressing the complex structure of a DNA sequence (different k-mers frequencies in the genomic regions), the number of parameters required for this task tends to be high. With the increase of the assumed

complexity of the structure of the DNA sequence, the number of hidden states needed raises. This increase of parameters therefore calls for parameters fixing for avoiding overfitting in the learning process.

Instead depending on the previous hidden states, we suggest here the use of high-order emission which depends on previous emitted observable variables. This HMM variant fits better to the locality nature for the task of emission k-mer structures, and only requires$O\left(m^2+4^K\right)$ compared to$O\left(m^K\right)$ parameters, where m is the number of hidden states, and K is the number of previous states in the dependency.

Generalized HMM

TODO: ask Tommy for a gHMM that states that emit multiple variables to cite

Generalized hidden Markov models (gHMM) are a family of variants of HMM, some of which contain hidden states that emit zero or multiple observable variables. Applied on DNA, these models can assume and learn a TFBS frequency across DNA sequences. It is a way to generalize emission of the HMM, and the EM algorithm can be adjusted to accommodate such generalization, as we will demonstrate in our EM adaptation.

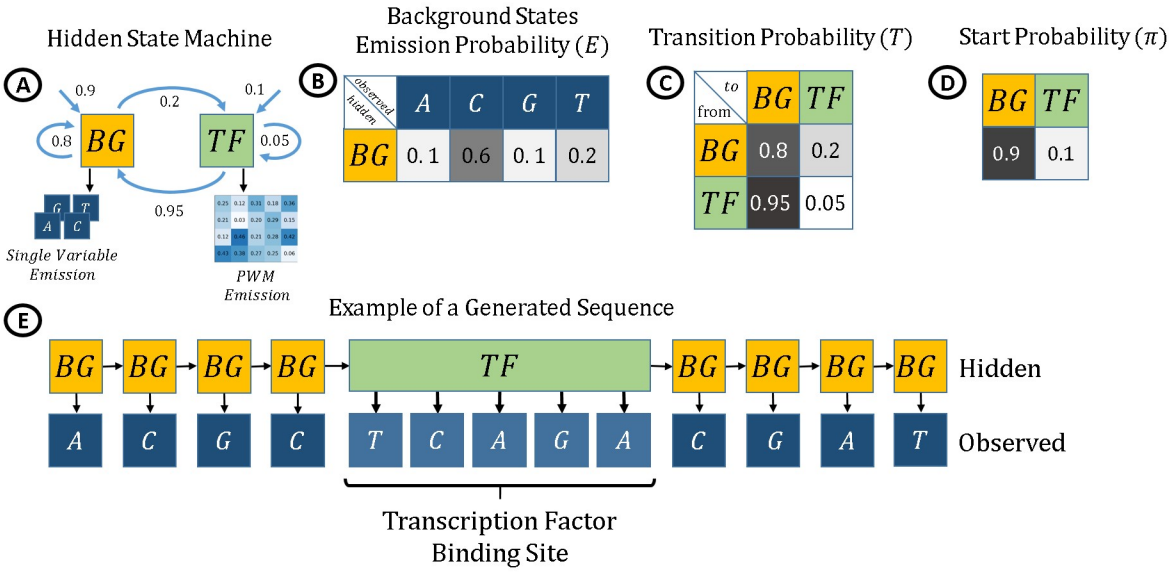


Figure: A) A gHMM with background hidden state (yellow) and TF hidden state. As in regular HMM, the emission of the background hidden state is sampled from E (B), the hidden states transitions and are sampled from T (C) and start hidden state is sampled fromπ (D). E) Unlike regular HMM, the emission of the TF hidden state is sampled from a PWM, which generates motif of 6 observable variables.

HOP-HMM

Here we present HOP-HMM, a variant model of HMM, that is well fitted to utilize the structure of enhancers containing TFBSs inside them, due to the TFBS emitting TF-states that take part in the generation process of the sequence.

HOP-HMM extends the algorithm of Kaplan et al., 2011 (Kaplan et al., 2012), which also contains hidden states emitting TFBS sampled from PWMs to predict enhancers location in the genome. Both algorithms are part of the generalized hidden Markov model (gHMM) family, which are HMM variants that contain hidden states that may emit multiple observed variables. Although HMM variants using higher-order HMM, in which the transition and emission are dependent on previous hidden states where used previously (Ferguson, 1980; Preez, 1998), an HMM variant in which the emission is dependent on previous emissions is a less researched field.

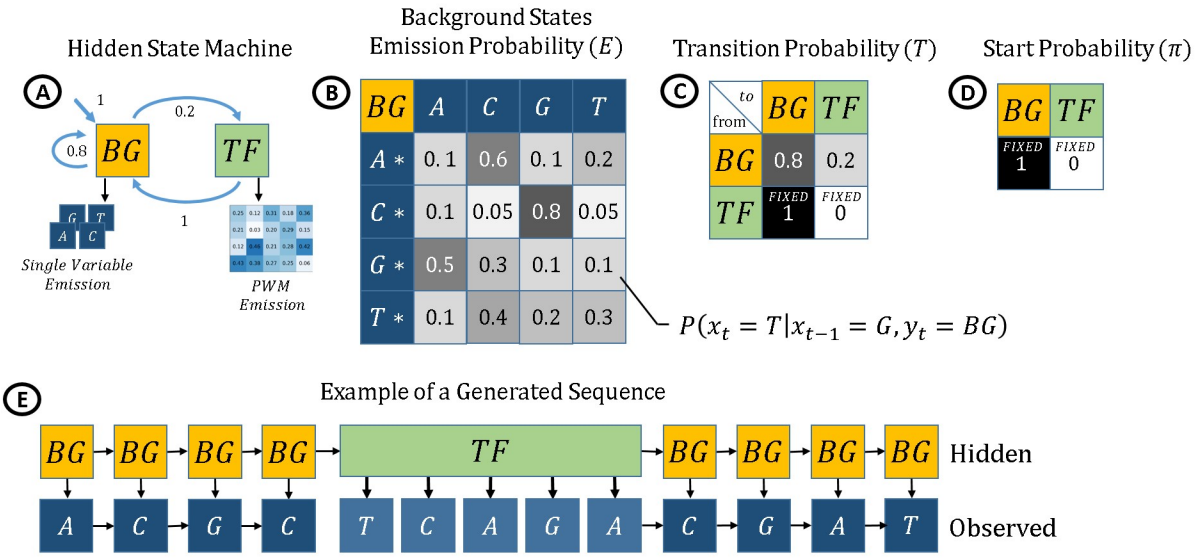


Figure 9: A small HOP-HMM that similarly to gHMM in figure 8, has one background-state that emits a single observable variable, and one TF-state which emits a TFBS sampled from its PWM. Unlike gHMM, in HOP-HMM TF-state can not be the start hidden state and the emission of the background-state is of order two (o=2), meaning it is conditioned on the previous observable variable.

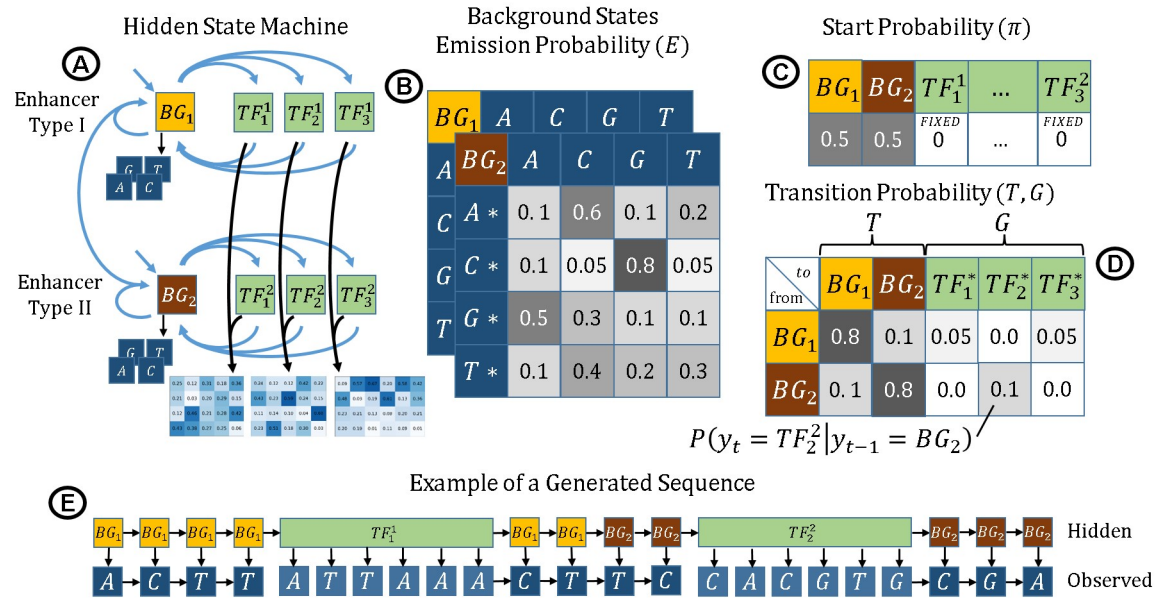


Figure 10: A) A more complex HOP-HMM with two background-stateBG_1 andBG_2, where each has 3 TF-states. B) each of the background-states has its own 2-order emission distribution in a 4x4 matrix. C) The start hidden state distributionπ allows only background-states to start the hidden sequence. D) The transition probability is held by matrices T and G. E) The example generated sequence is built out of two types of sequences, each has its own TFBS frequency and background nucleotide bigram frequency, representing two alternating types of enhancers.

Hidden States Indexing

We use two indices to describe a hidden-state:

- Background-states are indexed as$(j,0)$ where$j \in [m]$ and m is the number background-states.
- TF-states are indexed as(j,l) where$j \in [m]$,$l \in [k]$. and k is the number of TF-states each of the background-states has.

For example, in figure 10 we see a HOP-HMM with $m = 2$ and $k = 3$ and a total of 8 hidden-states. The TF-state indexed (j,l) belongs to the $(j,0)$ background-state (see figure 11), and the model can transfer into (j,l) only from $(j,0)$. Note that we used simpler notation in figures 9 and 10 for readability.

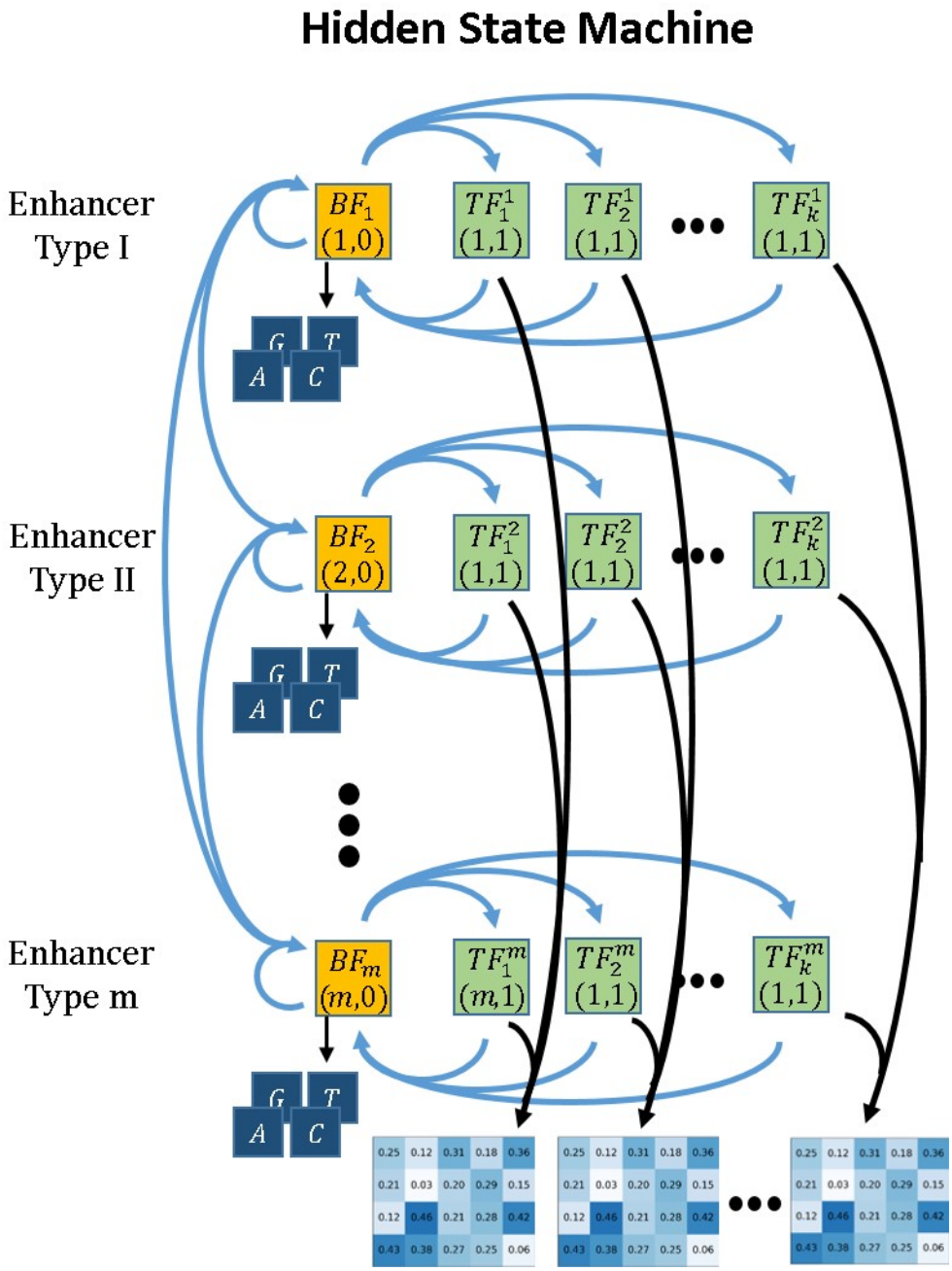


Figure 11: general hidden states graph of HOP-HMM. Each row represents a sequence type, where each of the m background-states (yellow) has k TF-states (green). Not all transitions are possible, moving between the rows is possible only by a background-state to background-state transition.

While most background-states represent an enhancer type, we also would like to have background regions in between the enhancer. For that end we predefine one or more background-states as non-enhancers, by restricting the probability of transferring from these background-states into their TF-states.

Emission

HOP-HMM is defined with k PWMs$W_1, W_2, ..., W_k$ that remain fixed during training. Each of the k PWMs is shared with m TF-states, e.g. the PWMW_l, where$l \in [k]$, is shared between subs-states$(1,l),(2,l),...,(m,l)$ and is used for the TF-state emission sampling. The PWMs vary in their column amounts (as the different TFBSs vary in length), where each column represents a nucleotide distribution at that position. When the model enters a TF-state, it emits a motif by sampling from a PWM column by column independently, as described in Figure 3.

The background-states, denoted as$(1,0),(2,0),...,(m,0)$, are responsible for the emission of inter-TFBS parts of the enhancers lacking long motifs. Similarly to regular states in HMM, background-states emit single nucleotides, where their emission is conditional on the previous of letters emitted in the DNA sequence. The emission from background-states is done by sampling a nucleotide from the distributions stored in E tensor. E dimension is o+1, and its size is $m \times 4 \times 4 \times ... \times 4$ (with o fours) and its values are describe the emission probability$E_{j,x_{t-o+1},x_{t-o+2},...,x_t} = P(x_t|y_t = (j,0), x_{t-o+1},...,x_{t-1})$, meaning that when$x_t$ is sampled by the model, the preceding$o-1$ observed variables are used as indices of the tensor for getting emission probability vector$E_{j,x_{t-o+1},x_{t-o+2},...,x_{t-1},*}$.

For the first variables emitted in the sequence, the missing dimensions of the preceding variables are summed to form the probability vector, e.g. if$t = o-1$, a single variable is missing for emittingx_t and the distribution used for emission sampling is$\sum_{i\in[4]} \frac{E_{j,i,x_1,...,x_{t-1}}}{4}$.

Transition

In HOP-HMM, the first hidden state in a sequence can only be a background-state. The first background-state, as in HMM, is chosen by sampling fromπ, a probability vector$\pi_j = P(y_1 = (j,0))$. Once in a background-state, the model can only transit into a small subset of states, and since most of the possible transition are not allowed, a single transition matrix from all states to all states would be sparse. Instead, as described in figure 4, we hold only the possible transition probabilities in two matrices, representing the two types of allowed transitions:

- T for background-state to background-state transitions, a$m \times m$ matrix where$T_{j_1,j_2} = P(y_{t+1} = (j_2,0)|y_t = (j_1,0))$.
- G for background-state to TF-state transitions a$m \times k$ matrix where$G_{j,l} = P\left(y_{t+1:t+|W_l|} = (j,l)|y_t = (j,0)\right)$.

When in a background-state, after the observable variable emission, the model samples its next hidden state from a probability vector that is a concatenation of a row in T and a row in G. If a TF-state is chosen, after the TF-state's motif emission, the model returns back to the background-state to emit another single observable variable and so on.

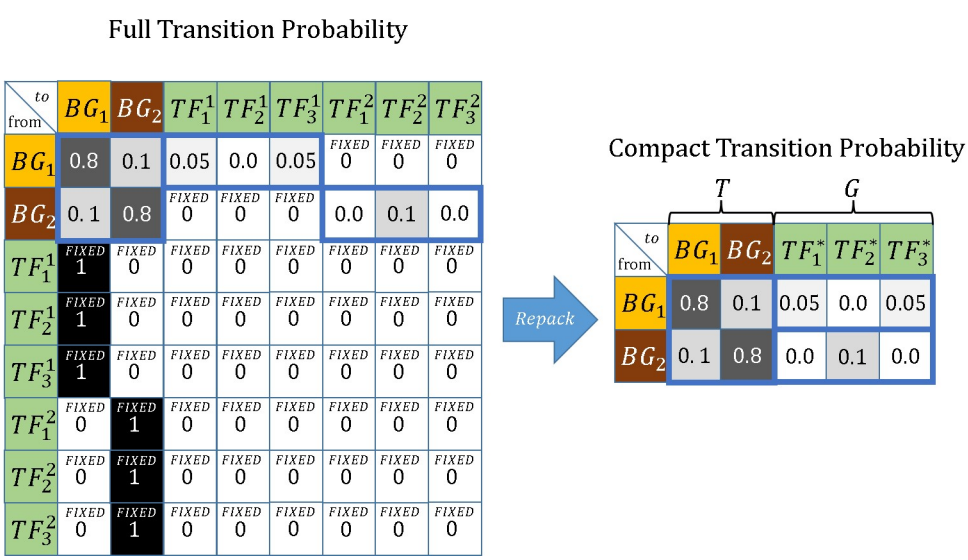


Figure 12: Instead of holding a single sparse 8x8 transition matrix, an alternative compact form holds only the non-fixed transition probabilities, split into T and G matrices. The transitions non-fixed transitions probabilities are in between background-states, and between background-states to their TF-states (marked in blue). A concatenation of a row in T and G holds the probability of the next hidden-state given the current background-state.

Baum-Welch Algorithm

To be able to learn a DNA sequence with a HMM, we need to find parameters of the model that best explain the input sequence x. Formaly, given the observed DNA sequence$x_{1:L}$ where L is the length of the sequence, we would like find the parameters that maximize the incomplete-likelihood:

$$\theta^* = argmax_\theta \mathscr{L}\left(\theta | x_{1:L}\right)$$

Even though the incomplete-data likelihood of HMM in (1) is derivable byθ, optimizing it involves a summation of exponential-by-L complete-data elements, which is infeasible. Instead, the strategy of the EM algorithm is to optimize the expected value of the complete-data log-likelihood$log\left(P\left(x_{1:L}, y_{1:L} | \theta'\right)\right)$ whereθ' 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:

$$Q\left(\theta, \theta'\right) = E_Y\left[log\left(P_\theta\left(x_{1:L}, y_{1:L}\right)\right) | x_{1:L}, \theta'\right] = \sum_{y \in [m]^N} log\left(P_\theta\left(x_{1:L}, y_{1:L}\right)\right) P_{\theta'}\left(x_{1:L}, y_{1:L}\right) \tag{3}$$

Note that here, E is expressing an expected value and not the HMM emission probability. Every EM iteration is built of two parts: the E-step and the M-step. In the E-step we calculate the probabilities needed for the maximization of Q 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 usage.

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

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

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

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

Each of the three parts could be derived and 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 multiplier λ:

$$\frac{\partial}{\partial \pi_j}\left(\sum_{j'\in[m]} log\pi_{j'} P_{\theta'}\left(x_{1:L}, y_1 = j'\right) + \lambda\left(\sum_{j'\in[m]} 1 - \pi_{j'}\right)\right) = 0$$

we derive the equations and get $\frac{P_{\theta'}(x_{1:L}, y_1 = j)}{\pi_j} = \lambda$ for $j \in [m]$. Then we multiply each equation by the denominator and sum these m equations to receive $\lambda = P_{\theta'}(x_{1:L})$, which yields:

$$\pi_j = \frac{P_{\theta'}(x_{1:L}, y_1 = j)}{P_{\theta'}(x_{1:L})} = P_{\theta'}(y_1 = j | x_{1:L}) \tag{4}$$

We may follow the Lagrange multipliers method in the second and third parts, from which we'll receive:

$$T_{j_1,j_2} = \frac{\sum_{t\in 2\ldots L} P_{\theta'}(x_{1:L}, y_{t-1} = j_1, y_t = j_2)}{\sum_{t\in 2\ldots L} P_{\theta'}(x_{1:L}, y_{t-1} = j_1)} = \frac{\sum_{t\in 2\ldots L} P_{\theta'}(y_{t-1} = j_1, y_t = j_2 | x_{1:L})}{\sum_{t\in 2\ldots L} P_{\theta'}(y_{t-1} = j_1 | x_{1:L})} \tag{5}$$

and

$$E_{j,b} = \frac{\sum_{t\in[L]} P_{\theta'}(x_{1:L}, y_t = j)\, 1_b(x_t)}{\sum_{t\in[L]} P_{\theta'}(x_{1:L}, y_t = j)} = \frac{\sum_{t\in[L]} P_{\theta'}(y_t = j | x_{1:L})\, 1_b(x_t)}{\sum_{t\in[L]} P_{\theta'}(y_t = j | x_{1:L})} \tag{6}$$

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

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 to resolve all the parameters update states in (4), (5) and (6), it is enough to calculate the two terms during the E-step:

$$P_{\theta'}(y_t = j | x_{1:L}) \tag{7}$$

$$P_{\theta'}(y_{t-1} = j_1, y_t = j_2 | x_{1:L}) \tag{8}$$

We will calculate these terms by using the Forward-Backward algorithm.

Auxiliary Probabilities

Having the forward and backward probability matrices α and β, we now have all that is needed to calculate(8) and (9), using once more the HMM conditional independence in our calculations.Here we used the fact that

We denote the terms values of (7) as γ, a matrix of size $L \times m$:

$$\gamma_{t,j} = P\left(y_t = j | X, \theta'\right) = \frac{P\left(y_t = j, x_{1:L} | \theta'\right)}{P\left(x_{1:L} | \theta'\right)} = \frac{P\left(x_{1:L} | y_t = j, \theta'\right) \cdot P\left(y_t = j | \theta'\right)}{P\left(x_{1:L} | \theta'\right)} =$$

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

And we denote the values of (8) as ξ, a matrix of size $L - 1 \times m \times m$:

$$\xi_{t,j_1,j_2} = P\left(y_{t-1} = j_1, y_t = j_2 | x_{1:L}, \theta'\right) = \frac{P\left(y_{t-1} = j_1, y_{t]} = j_2, x_{1:L} | \theta'\right)}{P\left(x_{1:L} | \theta'\right)} =$$

$$= \frac{P\left(y_{t-1} = j_1, x_{1:t-1}\right) \cdot P\left(y_t = j_2 | y_{t-1} = j_1\right) \cdot P\left(x_t | y_t = j_2\right) \cdot P\left(x_{t+1:L} | y_t = j_2\right)}{\sum_{j' \in [m]} P\left(y_t = j, x_{1:t}\right) \cdot P\left(x_{t+1:L} | y_t = j\right)} =$$

$$= \frac{\alpha_{j_1,t-1} \cdot T_{j_1,j_2,} \cdot E_{j_2,x_t} \cdot \beta_{j',t}}{\sum_{j' \in [m]} \alpha_{j',t} \cdot \beta_{j',t}}$$

Baum-Welch Algorithm for HOP-HMM

Since the transitions and emissions assumptions of HOP-HMM are different, the complete-data likelihood requires different calculation for the EM algorithm to hold. The Baum-Welch algorithm can be adjusted to infer the parameters of the HOP-HMM variant$\theta = \{\pi, E, G, T\}$ from a DNA sequence X. As in the regular Baum-Welch algorithm covered in the previous section, given a sequence X at each EM iteration we calculate and optimize a Q function:

$$\begin{aligned}
Q\left(\theta, \theta'\right) = & \sum_{j \in [m]} log\pi_j \cdot P_{\theta'}\left(x_{1:L}, y_1 = (j,0)\right) \\
& + \sum_{t \in 2 \ldots L} \sum_{j_1, j_2 \in [m]} logT_{j_1,j_2} \cdot P_{\theta'}\left(x_{1:L}, y_{t-1} = (j_1,0), y_t = (j_2,0)\right) \\
& + \sum_{t \in 2 \ldots L} \sum_{j \in [m], l \in [k]} logG_{j,l} \cdot P_{\theta'}\left(x_{1:L}, y_{t-1} = (j,0), y_t = (j,l)\right) \\
& + \sum_{t \in [L]} \sum_{j \in [m]} logE_{j,x_t} \cdot P_{\theta'}\left(x_{1:L}, y_t = (j,0)\right) \\
& + \sum_{t \in [L]} \sum_{l \in [k]} logL_W\left(x_{t:t+|W_l|}\right) \cdot P_{\theta'}\left(x_{1:L}, y_{t:t+|W_l|} = (j,l)\right)
\end{aligned}$$

where$L_W(\bar{x})$ is the likelihood of the TFBS$\bar{x}$ to be emitted by PWMW:$L_M(\bar{x}) = P(\bar{x}|W) = \prod_{i \in \{1,\ldots,|\bar{x}|\}} W_{\bar{x}_i,i}$. The last component, with the TFBS likelihood does not contain elements fromθ, therefore it is not optimized in the m-steps.

Theθ which optimizes Q, i.e.$\theta_{max} = argmax_\theta Q(\theta, \theta')$, is built similarly as in the regular Baum-Welch algorithm, as following:

$$\pi_j = \frac{P_{\theta'}\left(x_{1:L}, y_1 = (j,0) | \theta'\right)}{P_{\theta'}\left(x_{1:L} | \theta'\right)} = P_{\theta'}\left(y_1 = (j,0) | x_{1:L}\right) \tag{9}$$

$$T_{j_1,j_2} = \frac{\sum_{t \in 2 \ldots L} P_{\theta'}\left(x_{1:L}, y_{t-1} = (j_1,0), y_t = (j_2,0)\right)}{\sum_{t \in 2 \ldots L} P_{\theta'}\left(x_{1:L}, y_{t-1} = (j_1,0)\right)} = \frac{\sum_{t \in 2 \ldots L} P_{\theta'}\left(y_{t-1} = (j_1,0), y_t = (j_2,0) | x_{1:L}\right)}{\sum_{t \in 2 \ldots L} P_{\theta'}\left(y_{t-1} = (j_1,0) | x_{1:L}\right)} \tag{10}$$

$$G_{j,l} = \frac{\sum_{t \in 2 \ldots L} P_{\theta'}\left(x_{1:L}, y_{t-1} = (j,0), y_t = (j,l)\right)}{\sum_{t \in 2 \ldots L} P_{\theta'}\left(x_{1:L}, y_{t-1} = (j,0)\right)} = \frac{\sum_{t \in 2 \ldots L} P_{\theta'}\left(y_{t-1} = (j,0), y_t = (j,l) | x_{1:L}\right)}{\sum_{t \in 2 \ldots L} P_{\theta'}\left(y_{t-1} = (j,0) | x_{1:L}\right)} \tag{11}$$

and

$$E_{j,b_1,\ldots,b_o} = \frac{\sum_{t \in o,\ldots,L} P_{\theta'}\left(x_{1:L}, y_t = (j,0)\right) 1_{b_1,\ldots,b_o}\left(x_{t-o+1,\ldots,t}\right)}{\sum_{t \in o,\ldots,L} P_{\theta'}\left(x_{1:L}, y_t = (j,0)\right)} = \tag{12}$$

$$= \frac{\sum_{t \in o,\ldots,L} P_{\theta'}\left(y_t = (j,0) | X\right) 1_{b_1,\ldots,b_o}\left(x_{t-o+1,\ldots,t}\right)}{\sum_{t \in o,\ldots,L} P_{\theta'}\left(y_t = (j,0) | x_{1:L}\right)}$$

Algorithm 3 HOP Forward Algorithm

Input:

X - Observed DNA sequence

Algorithm:

$for\, j = [1, ..., m]:$

$\alpha_{j,1} = \pi_j \cdot E_{j,x_1}$

$for\, t = [2, ..., L]:$

$for\, j = [1, ..., m]:$

$$\alpha_{j,t} = \underbrace{\sum_{j' \in [m]} \alpha_{j',t-1} \cdot T_{j',j} \cdot E_{j,x_{t-o+1},...,x_t}}_{\text{base-state transitions}} + \underbrace{\sum_{l \in [k]} \alpha_{j,t-|W_l|-1} \cdot G_{j,l} \cdot L_{W_l}\left(x_{t-|W_l|}, ..., x_{t-1}\right) \cdot E_{j,x_{t-o+1},...,x_t}}_{\text{sub-state transitions}}$$

Hence to complete the EM iteration, the three missing components in (9), (10), (11), (12) are:

$$P_{\theta'}\left(y_t = (j,0) | x_{1:L}\right) \tag{13}$$

$$P_{\theta'}\left(y_{t-1} = (j_1,0), y_t = (j_2,0) | x_{1:L}\right) \tag{14}$$

$$P_{\theta'}\left(y_{t-1} = (j,0), y_t = (j,l) | x_{1:L}\right) \tag{15}$$

For the calculation of these probabilities, we will need to calculate the forward and backward probabilities, and then few other auxiliary terms. The forward and backward probabilities are only for indicating the sequence entering-to and exiting-from background-states:

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

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

HOP-Forward Algorithm

We calculate α of size $m \times L$, iterating over $t = 1, 2, ..., L$ as following:

Calculation notes: In the beginning of the sequence, when $1 \leq t < o$, part of the preceding observable variables are missing. Since E has $o+1$ dimensions, $E_{j,x_1,...,x_t}$ is not defined, so we define it here as $E_{j,x_1,...,x_t} = \sum_{b_1,...,b_{o-t} \in \{A,C,G,T\}} \frac{1}{4^{o-t}} \cdot E_{j,b_1,...,b_{o-t},x_1,...,x_t}$ that is the expected probability upon possible preceding variables. We used the fact that $P(A) = \sum_{b \in B} P(b) \cdot P(A|b)$ and the assumption that the observable variables preceding the sequence came from a uniform distribution, $P(x_i) = \frac{1}{4}$ where $i < 1$. Also, when summing the TF-state transition part $l \in [k]$, the PWMs W_l with length equal or bigger than $t+1$ are not part of the summation.

HOP-Backward Algorithm

For β of size $m \times L$, we iterating over $t = L, L-1, ..., 1$ as following:

Note that when $t > L - |W_l|$, there are missing observable variables to fully calculate the TF-state transition. In these situations this contribution of these component to the summation is zero, meaning our HOP-HMM as the behavior of avoiding a transition into a TF-state when the PWM is too long to fit into the sequence X length.

Algorithm 4 HOP Backward Algorithm

Input:

X - Observed DNA sequence

Algorithm:

$\beta_{1:m,L} = 1$

$for\, t = [L-1, ..., 1]:$

$for\, j = [1, ..., m]:$

$$\beta_{j,t} = \underbrace{\sum_{j' \in [m]} \beta_{j',t+1} \cdot E_{j',x_{t-o+2},...,x_{t+1}} \cdot T_{j,j'}}_{\text{base-state transitions}} + \underbrace{\sum_{l \in [k]} \beta_{j,t+|W_l|+1} \cdot L_{W_l}\left(x_{t+1}, ..., x_{t+|W_l|}\right) \cdot E_{j,x_{t-o+|W_l|+2},...,x_{t+|W_l|+1}} \cdot G_{j,l}}_{\text{sub-state transitions}}$$

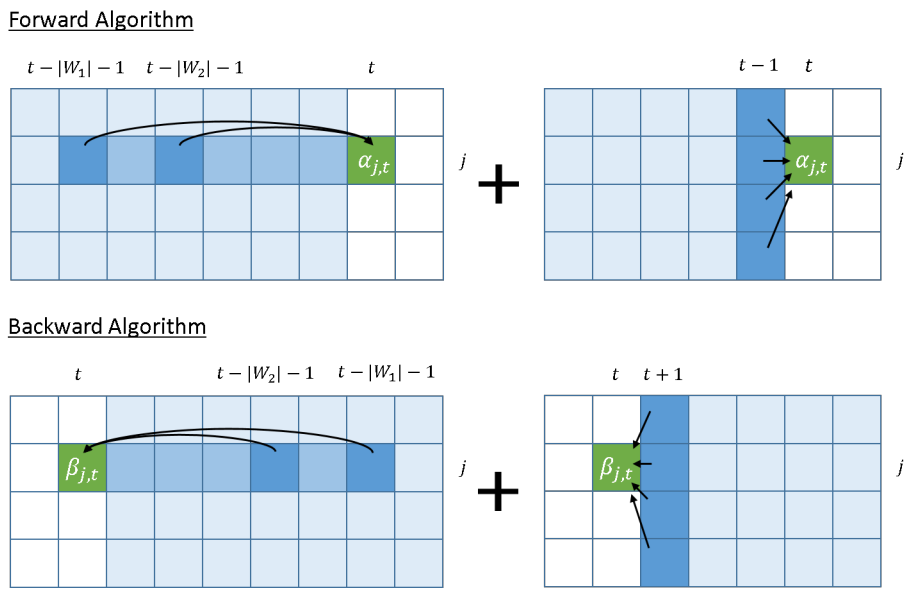


Figure 5: In our version of the forward-backward algorithm, each of the α and β cells are filled from both the adjacent background-states transitions and the background-states preceding or proceeding the motifs emitted by the TF-states.

Forward Algorithm Explanation

We will now explain why the described dynamic calculation result with $\alpha_{j,t} = P(y_t = (j,0), x_{1:t})$ and $\beta_{j,t} = P(x_{t+1:L} | y_t = (j,0))$, starting with the forward probabilities α. 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 background-state (j,0):

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

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

We could develop the right term of a TF-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 of x_t is dependent on only y_t (and also $x_{t-o:t-1}$ if y_t is a background-state) and y_t is dependent on only y_{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 of θ and with already filled α cells:

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

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

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

Using the conditional independencies to simplify the probabilities:

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

$$
=\alpha_{j',t-1}\cdot T_{j',j}\cdot E_{j,x_{t-o+1},\ldots,x_t}
$$

Backward Algorithm Explanation

For the backward probabilities β, the explanation is similar. The main difference between the regular HMM backward probability is the condition on the o-1 preceding observable variables $x_{t-o+2:t}$, which are necessary for the background-state emission is conditional on them.

Using the law of total probability:

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

$$
=\underbrace{\sum_{j'}P\left(y_{t+1}=(j',0),x_{t+1:L}|y_t=(j,0),x_{t-o+2:t}\right)}_{\text{base-state transition}}+\underbrace{\sum_{l}P\left(y_{t+1:t+|W_l|}=(j,l),y_{t+|W_l|+1}=(j,0),x_{t+1:L}|y_t=(j,0)\right)}_{\text{sub-state transition}}
$$

For the background-state transition term, we can use the chain rule and the Markovian independence of the transitions and emissions:

$$P\left(y_{t+1}=(j',0),x_{t+1:L}|y_t=(j,0),x_{t-o+2:t}\right)=$$

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

$$\cdot P\left(y_{t+1}=(j',0)|y_t=(j,0),x_{t-o+2:t}\right)=$$

$$=P\left(x_{t+2:L}|y_{t+1}=(j',0),x_{t-o+3:t+1}\right)\cdot P\left(x_{t+1}|y_{t+1}=(j',0),x_{t-o+2:t}\right)\cdot P\left(y_{t+1}=(j',0)|y_t=(j,0)\right)=$$

$$=\beta_{j',t+1}\cdot E_{j',x_{t-o+2},\ldots,x_{t+1}}\cdot T_{j,j'}$$

For the TF-state transition term, we use once more the chain rule, followed the simplification using the conditional independencies:

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

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

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

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

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

$$=P\left(x_{t+|W_l|+2:L}|y_{t+|W_l|+1}=(j,0)\right)\cdot P\left(x_{t+|W_l|+1}|y_{t+|W_l|+1}=(j,0)\right)\cdot$$

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

$$=\beta_{j,t+|W_l|+1}\cdot E_{j,x_{t-o+|W_l|+1},\ldots,x_{t+|W_l|+1}}\cdot L_{W_l}\left(x_{t+1},\ldots,x_{t+|W_l|}\right)\cdot G_{j,l}$$

Auxiliary Probabilities

Using the forward and the backward probability matrices, we are ready to calculate the auxiliary probabilities (13) (14) (15). The first probability that will help us for that task is ψ, a matrix of size $m\times k\times L$:

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

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

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

$$\cdot P\left(x_{t+|W_l|+1}|y_{t+|W_l|+1}=(j,0)\right)\cdot P\left(x_{t+|W_l|+2:L}|y_{t+|W_l|+1}=(j,0),x_{t+|W_l|-o+3:t+|W_l|+1}\right)=$$

$$=\alpha_{j,t}\cdot G_{j,l}\cdot L_{W_l,}\left(x_{t+1},...,x_{t+|W_l|}\right)\cdot E_{j,x_{t+|W_l|-o+2},...,x_{t+|W_l|+1}}\cdot\beta_{j,t+|W_l|+1}$$

The second probability, is of the observed sequence X:

$$P(X)=\sum_{j\in[m]}\left(\alpha_{j,t}\cdot\beta_{j,t}+\sum_{l\in[k],\ t'\in[|W_l|]}\psi_{j,l,t-s}\right)$$

Now we can calculate probability (13) of the background-state at a given position given the sequence X, denoted asγ of size$m\times L$:

$$\gamma_{j,t}=P(y_t=(j,0)|X)=\frac{P(y_t=(j,0),x_{1:t})\cdot P(x_{t+1:L}|x_{1:t},y_t=(j,0))}{P(x_{1:L})}=$$

$$=\frac{P(y_t=(j,0),x_{1:t})\cdot P(x_{t+1:L}|x_{t-o+1:t},y_t=(j,0))}{P(x_{1:L})}=\frac{\alpha_{j,t}\cdot\beta_{j,t}}{P(x_{1:L})}$$

The probability (14) is the background-state to background-state transition given the sequence X, denoted asξ of size$m\times m\times L$:

$$\xi_{j_1,j_2,t}=P(y_{t-1}=(j_1,0),y_t=(j_2,0)|x_{1:L})=\frac{P(y_{t-1}=(j_1,0),y_t=(j_2,0),x_{1:L})}{P(x_{1:L})}=$$

$$=\frac{P(x_{1:t-1},y_{t-1}=(j_1,0),y_t=(j_2,0))\cdot P(x_{t:L}|y_t=(j_2,0),x_{1:t-1})}{P(x_{1:L})}=$$

$$=\frac{P(x_{1:t-1},y_{t-1}=(j_1,0))\cdot P(y_t=(j_2,0)|y_{t-1}=(j_1,0))\cdot P(x_t|y_t=(j_2,0),x_{1:t-1})\cdot P(x_{t+1:L}|y_t=(j_2,0),x_{1:t-1})}{P(x_{1:L})}=$$

$$=\frac{\alpha_{j_1,t-1}\cdot T_{j_1,j_2}\cdot E_{j_2,x_{t-o+1},...,x_t}\cdot\beta_{j_2,t}}{P(x_{1:L})}$$

Finally, the probability (15) is the background-state to background-state transition given the sequence X, denoted asξ of size$m\times k\times L$:

$$\eta_{j,l,t}=P(y_{t-1}=(j,0),y_t=(j,l)|x_{1:L})=\frac{\psi_{j,l,t}}{P(x_{1:L})}$$

Now with (13), (14), (15) at hand, we can calculateθ_{max} by assigning them at (9), (10), (11), (12).

HOP Baum-Welch

To conclude, the total EM algorithm for HOP-HMM:

Algorithm 5 HOP Baum-Welch

Input:
X - Observed DNA sequence
Algorithm:
for s=[1...MAX_EM_ITERATIONS]:
//e-step
$\alpha = \text{hop\_forward\_alg}(x_{1:L})$
$\beta = \text{hop\_backward\_alg}(x_{1:L})$
for $j = [1,...,m], l = [1,...,k], t = [1,...,L]$:
$$\psi_{j,l,t} = \begin{cases} \alpha_{j,t} \cdot G_{j,l} \cdot L_{W_{l,}}\left(x_{t+1:t+|W_l|}\right) \cdot E_{j,x_{t+|W_l|-o+2},...,x_{t+|W_l|+1}} \cdot \beta_{j,t+|W_l|+1} & |t+|W_l|+1 \leq L \\ 0 & |otherwise \end{cases}$$
$Px = \sum_{j \in [m]} \alpha_{j,L}$
for $j = [1,...,m], t = [1,...,L]$:
$\gamma_{j,t} = \frac{\alpha_{j,t} \cdot \beta_{j,t}}{Px}$
for $j = [1,...,m], l = [1,...,k], t = [1,...,L]$:
$\eta_{j,l,t} = \frac{\psi_{j,l,t}}{Px}$
for $j_1 = [1,...,m], j_2 = [1,...,m], t = [1,...,L]$:
$\xi_{j_1,j_2,t} = \frac{\alpha_{j_1,t-1} \cdot T_{j_1,j_2} \cdot E_{j_2,x_{t-o+1},...,x_t} \cdot \beta_{j_2,t}}{Px}$
//m-step
for $j = [1,...,m]$:
$\pi_j = \gamma_{j,1}$
for $b_1,...,b_o = [1,...,1],...,[4,...,4]$:
$E_{j,b_1,b_2,...,b_o} = \frac{\sum_{t \in o,...,L} \gamma_{j,t} \cdot 1_{b_1,...,b_o}(x_{t-o+1},...,x_t)}{\sum_{t \in o,...,L} \gamma_{j,t}}$
for $l = [1,...,k]$:
$G_{j,l} = \frac{\sum_{t \in 2,...,L} \eta_{j,l,t}}{\sum_{t \in 1,...,L-1} \gamma_{j_1,t}}$
for $j_2 = [1,...,m]$:
$T_{j,j_2} = \frac{\sum_{t \in 2,...,L} \xi_{j,j_2,t}}{\sum_{t \in 1,...,L-1} \gamma_{j,t}}$
If θ converged, break EM for loop

Learning Multiple Sequences at Once

The algorithm here is described for learning a single sequence of observable variables X. For learning the parameters θ from multiple sequences at once, we can use the same method as introduced in the original paper of Baum-Welch algorithm (Rabiner, 1989), which calculates the E step probabilities for every sequence, and in the m-step sums all positions from all sequence for the parameters update.

Classifying DNA a Sequence

One of the goals of the HMM and HOP-HMM learning process described above is to be able to classify the DNA sequences by deciding their hidden state per location. After the Baum-Welch algorithm procedure, we might want to reach that goal by choosing per position t the max-likelihood hidden state. In the HOP-HMM case, this means getting the hidden sequence:

$$y_t^* = argmax_{(j,l)} P(y_t = (j,l) | x_{1:L})$$

Algorithm 6 HOP-Viterbi Algorithm

Input:

θ- HOP-HMM parameters$\{E,T,G,\pi\}$

X - Observed DNA sequence

Algorithm:

$V^1_{j,1} = \pi_j \cdot E_{j,x_1}$

$V^2_{j,1} = 0$

$for\, t = [2,...,L]$:

$for\, j = [1,...,m]$:

create two vectors:

$A = \left\{V_{j',t-1} \cdot T_{j',j} \cdot E_{j,x_{t-o+1},...,x_t} | j' \in [m]\right\}$

$B = \left\{V_{j,t-|W_l|-1} \cdot G_{j,l} \cdot L_{W_l}\left(x_{t-|W_l|},...,x_{t-1}\right) \cdot E_{j,x_{t-o+1},...,x_t} | l \in [k]\right\}$

$V^1_{j,t} = max\,(A \cup B)$

$V^2_{j,t} = \begin{cases} (argmax(A),0) & max(A) > max(B) \\ (j, argmax(B)) & otherwise \end{cases}$

$y_L = \left(argmax_j V^1_{j,L}, 0\right)$

t=L

while t > 1:

$(j,l) = V^2_{y_t[0],t}$

if$l = 0$:*# if the hidden state at t-1 is a background-state*

$y_{t-1} = (j,0)$

$t = t-1$

else:

$y_{t-|W_l|:t-1} = (j,l)$

$y_{t-|W_l|-1} = y_t$

$t = t-|W_l|-1$

Although simple to calculate with theγ auxiliary probability, such a hidden sequence could be problematic. Each hidden state maximizes the likelihood at its location, but the hidden sequence together with its transitions might not be the maximal likelihood one, and in fact might even include illegal transition (with probability 0).

Our task here is reaching the hidden sequence that maximizes the likelihood of the observed sequence, where all the sequence is considered:

$$y^*_{1:L} = argmax_{y_{1:L}} P\left(y_{1:L} | x_{1:L}\right) \tag{16}$$

HOP-Viterbi Algorithm

A common way to derive the hidden sequence that maximizes (16) is the Viterbi algorithm, which is similar to the Forward algorithm. The main differences between the two algorithms are the usage of*max* instead of summing over the possible transitions on each step of the dynamic algorithm, and keeping information of the chosen maximal value used via the*argmax* function. The HOP-HMM adaptation to this algorithm includes supporting the TF-states and the high order emission of the background-states.

Results

Synthetic Sequences

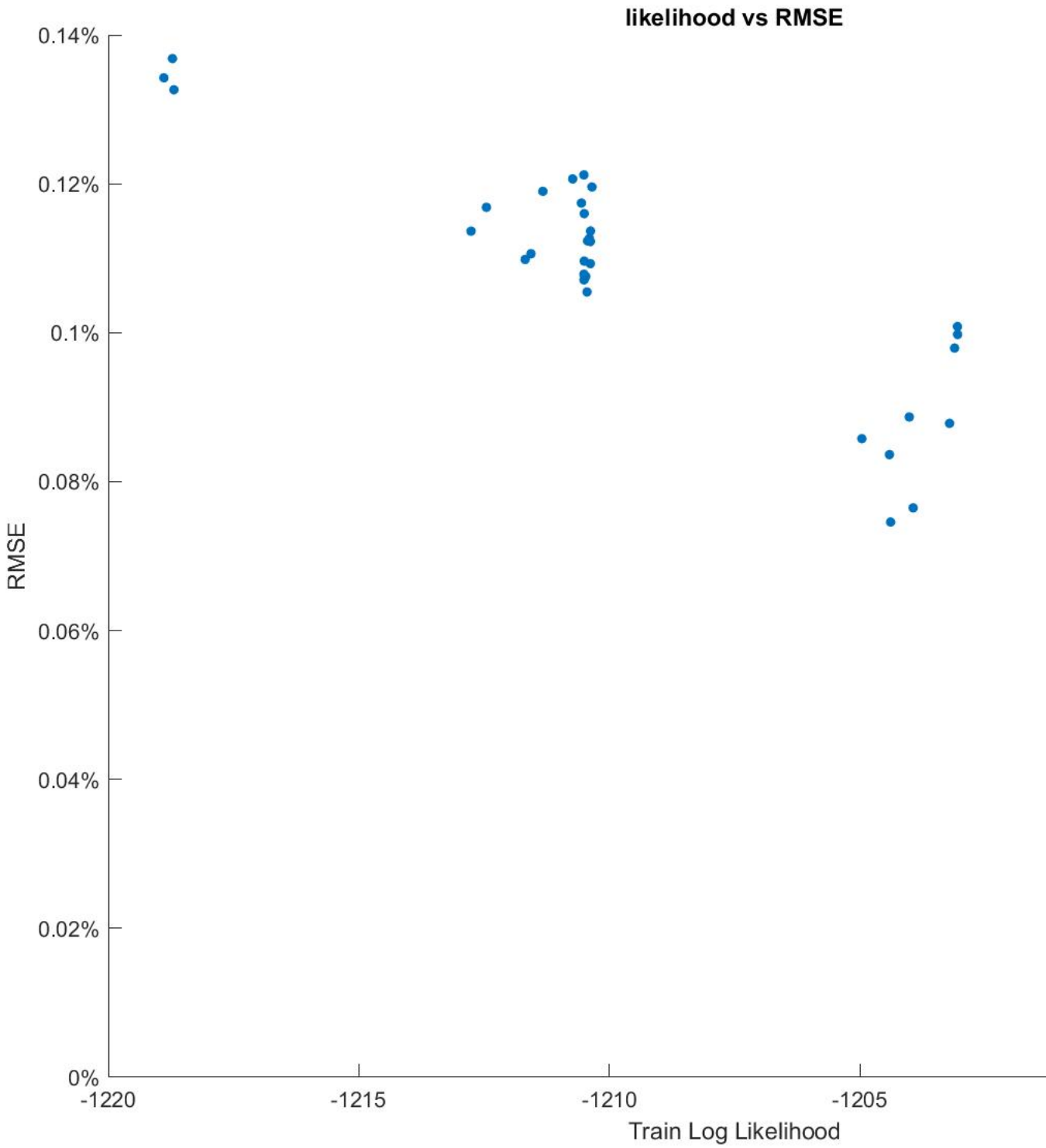
Before applying the algorithm to a real DNA data, we checked its capabilities on synthetic data.

We sampledθ parameters, while making sure some background-states had chance to transition into TF-states to mimic enhancers, and some background-states had almost no chance of such transitions, to mimic the PWM lacking background around

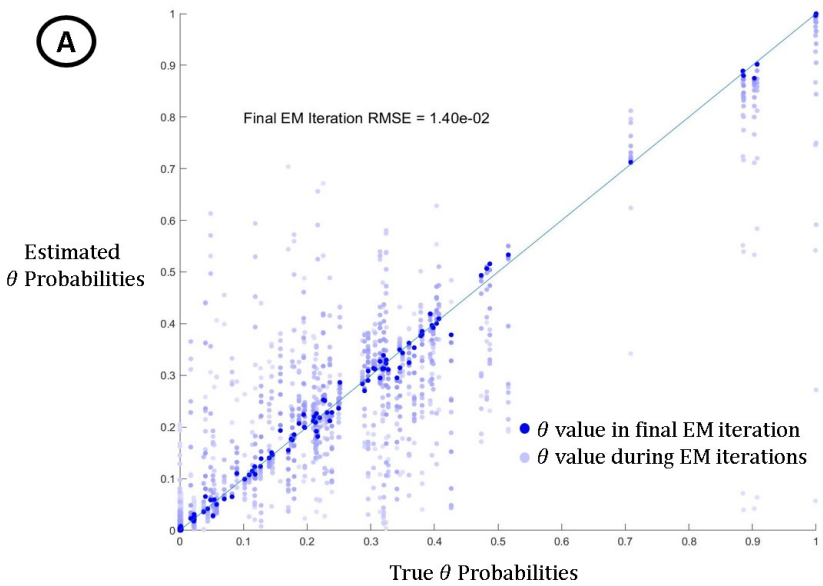
the shoulders of enhancers in the DNA. We marked the sampledθ as the trueθ which we would like to estimate in the HOP Baum-Welch algorithm. We then generated several sequences using a HOP-HMMs with the trueθ, keeping both the observed and the hidden sequences. Finally, we split the sequences into train and test and estimated aθ using the train sequences, and measuring its accuracy over both the train and the test sequences.

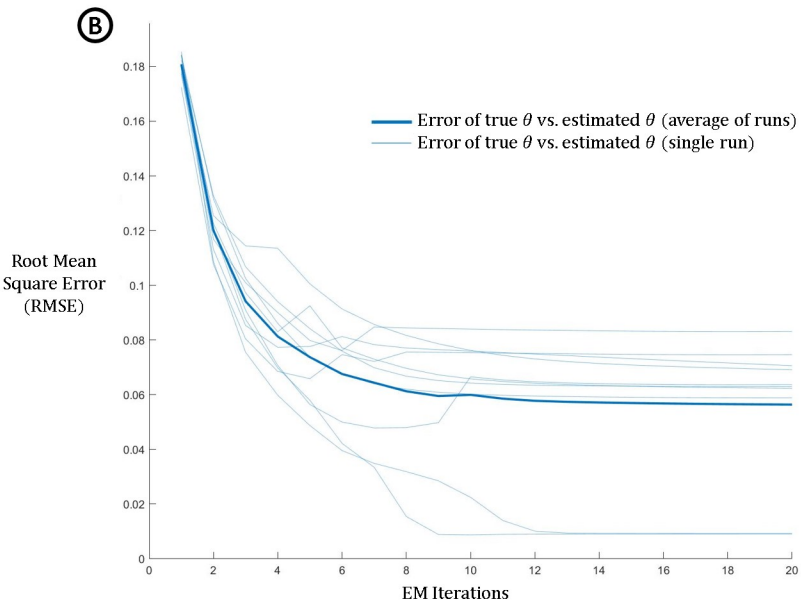
During our experiments, we noticed 2 interesting points:

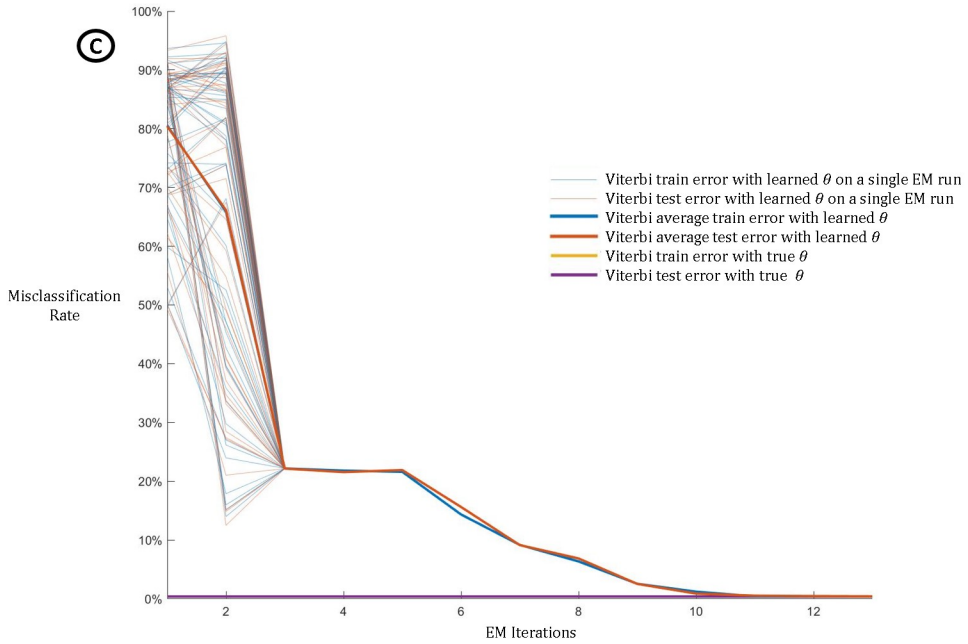
- Running the HOP Baum-Welch algorithm multiple times with different initializationθ converged into different result parameters. There is a correlation between the log likelihood yielded by theθ estimation and the RMSE between the estimated and trueθ (see figure 12). This is important since on observed sequences in real data, only theθ likelihood is known while the trueθ is unknown and so it the error. This correlation suggests a strategy of redoing several EM runs, and choosing the one with the highest likelihood will result in an estimation closer to the trueθ.
- Adding regularization on T inθ by limiting it so the major diagonal of T (probability of staying in the same hidden state) is larger than the sum of the rest of its values by a factor caused to convergence to betterθ estimations, which are closer to the trueθ.



12: Over multiple runs of HOP-Baum-Welch, a correlation exists between the estimatedθ likelihood over the learned sequences and the error comparing the trueθ.


A
Final EM Iteration RMSE = 1.40e-02
Estimated
θ Probabilities
θ value in final EM iteration
θ value during EM iterations
True θ Probabilities


B
Error of true θ vs. estimated θ (average of runs)
Error of true θ vs. estimated θ (single run)
Root Mean
Square Error
(RMSE)
EM Iterations



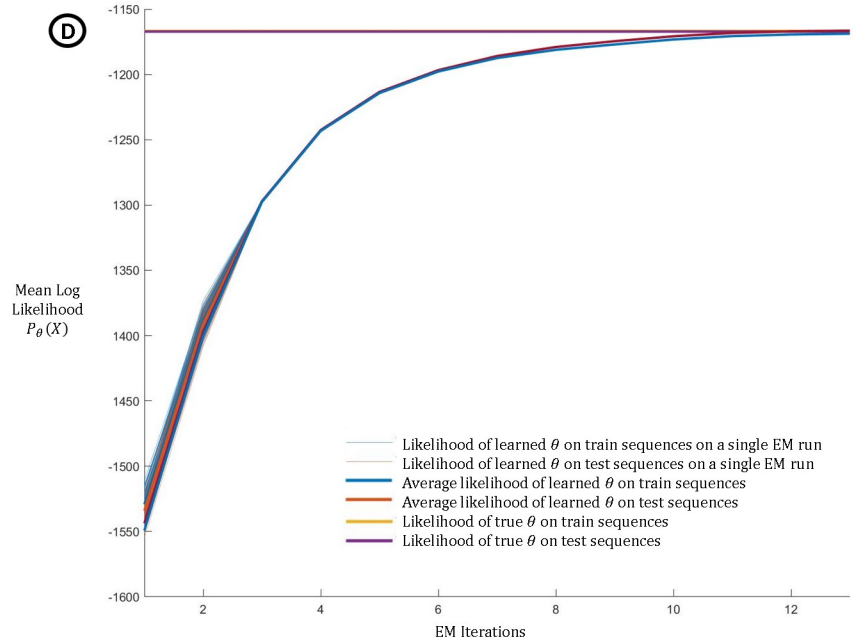


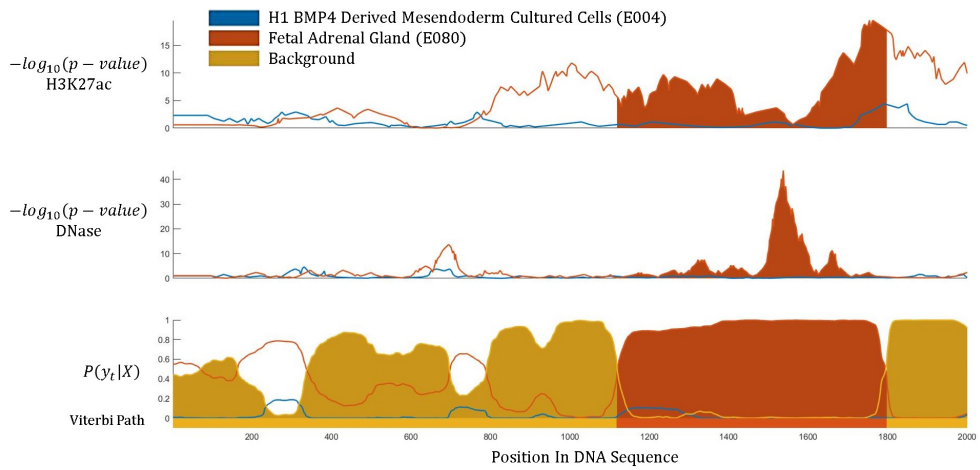
Figure: (A) the EM iterations draws the estimatedθ values mostly closer to the values of the trueθ. (B) The error between the true and estimatedθ decrease, and after a few iterations converge to the same path regardless of the initialization. (C) During the EM iterations, the learnedθ yields a more accurate Viterbi estimation of the hidden states. Note that not even the trueθ could produce Viterbi paths that is a perfect match to the true hidden sequences. (D) The mean log likelihood of the sequences increases during the EM iterations. The experiment was done on 500 synthetic sequences (85% train, 15% test), 1000 long. The trained model had 6 hidden background-states with emission order of 2, each background-state had 25 PWMs-states.

TODO: confusion graph of the states classification of the Viterbi algo.

Roadmap Dataset

Perprocessing downloaded from Roadmap (links) tissues with all the histone mark data (number of tissues). Manipulated bed files with Bedtools, center of peaks +- 500. taking parts with H3K27ac and H3K4me3 and without ...

Choosing the PWMs Part of the definition of a HOP-HMM is a set of PWMs emitted by the TF-states. JASPAR has 719 vertebrates PWMs, and because of limitations related to the implementation of algorithm, we needed to select around 30 out of the possible 719 PWMs. The selected PWMs should be the most indicative for the tissue specific enhancers. In the process of choosing the PWMs, we aim to choose the PWMs that appear in one of the tested enhancers and not the others. we used the correlation (marked as$L_W\left(x_{t-|W|},\ldots,x_{t-1}\right)$ in above sections) to measure the appearance of a PWM in a sequence in a position in a sequence. We calculated the 3 best matching positions of every PWM in every sequence, and calculated how successful, in terms of AUC-ROC, this sum can separate one sequence type from the other using a threshold.



TODO: Violine graph of the posterior of places Viterbi marked as PWMs and didnt mark as PWMs. violine of the acc. of the places marked by the Viterbi on real DNA. also, deep tool of the acc. and ME3 of these sequences.

TODO: confusion graph of the states classification of the Viterbi algo.

TODO: deeptools: aligned heatmap sequences with the peak of the H3K27ac and post. prob., are they similar?]

Figure: HOP-Viterbi classification of a region classified as an enhancer, ChromHMM classification and the H3K27ac features

TODO: Use HUJI theses format, 12 point font size, with 1.5 line spacing

1. Opening page including:

a. Heading – "The Hebrew University of Jerusalem – the Faculty of Mathematics and Natural Science – the Institute of (Chemistry, Life Sciences, Physics, Mathematics, Earth Sciences, Computer Science)

b. Title of the thesis in Hebrew and English

c. Author's name (+ student number)

d. Supervisor's name

e. "Thesis for Master's degree in Natural Science"

f. Date of submission in English and Hebrew

2. Personal page (dedication etc.)

3. Abstract

4. Table of contents

5. Body of the work

6. Bibliography

7. Appendices

The body of the work (section 5 above) will include:

1. Scientific introduction

2. Thesis aims

3. Methods (theoretical or experimental part)

4. Results

5. Discussion and summary.

References

Ahituv et al., 2007

Misc

Pretraining When sequences are labeled as tissue-specific enhancers are available, it is possible to pretrain a multi-background-state HOP-HMM with them. The E and G parts of such a model could be initialized with parameters taken from a smaller HOP-HMM, trained on 2 class datasets built out of tissue-specific enhancers and background DNA sequences. For each tissue, we build a dataset and trained a 2 background-state HOP-HMM model (one background-state for the enhancer and one for the background) and used the learned parameters of the enhancer background-state to initialize the bigger multi-background-state HOP-HMM.

TODO: number the chapters and figures

TODO: abstract

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"background" DNA, and was used to classify enhancers from background using only the k-mer distribution (Inbar and tommy, 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.

<u>TODO: Semi-Supervised Learning Scheme:</u>

1. **Pre-training:** Calculate maximal likelihood initialization parametersθ_0 with from observed labeled dataset of sequencesY_0 andX_0

2. **Unsupervised learning:** Learn theθ given X unlabeled sequences by approximating$\theta_{best} = argmax_\theta \mathscr{L}(\theta|X)$

3. **Predicting labels:**Given learnedθ, infer hidden states Y for unlabeled X
4. **Tuning:** Preform hyperparameters optimization

experiment graphs:

per nucleotide binary classification (background vs enhancer) - heatmap (x-location relative to peak, y-sequence index, color-post. probability) use the$\gamma_{i,*}$ of enhancers where i is the state of the enhancer, pick only the 2000+- around the enhancer's peak, where the center is the max of the H3K27ac signal, showing the sequences are most likely enhancers surrounded by non enhancers .

per sequence mutli-class classification (background vs enhancer) - heatmap (state number, y-sequence index, color-maximal post. probability for the sequence of that state) ofγ of enhancers, where the center is the max of the H3K27ac, showing the sequences are most likely enhancers surrounded by non enhancers

per sequence classification

[per sequence binary classification - background vs enhancer]

[per nucleotide binary classification - background vs enhancer]

[per sequence binary classification - background vs enhancer]

TODO: 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

References

Ahituv, N., Zhu, Y., Visel, A., Holt, A., Afzal, V., Pennacchio, L. A., & Rubin, E. M. (2007). Deletion of ultraconserved elements yields viable mice. PLoS biology, 5(9), e234.

Ainscough, R., Bardill, S., Barlow, K., Basham, V., Baynes, C., Beard, L., ... & Burrows, C. (1998). Genome sequence of the nematode C. elegans: a platform for investigating biology. Science, 282(5396), 2012-2018.

Akalin, A., Fredman, D., Arner, E., Dong, X., Bryne, J. C., Suzuki, H., ... & Lenhard, B. (2009). Transcriptional features of genomic regulatory blocks. Genome biology, 10(4), R38.

Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature biotechnology, 33(8), 831.

Andersson, R., Gebhard, C., Miguel-Escalada, I., Hoof, I., Bornholdt, J., Boyd, M., ... & Ntini, E. (2014). An atlas of active enhancers across human cell types and tissues. Nature, 507(7493), 455.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. The annals of mathematical statistics, 37(6), 1554-1563.

De Beer, Z. W., Duong, T. A., Barnes, I., Wingfield, B. D., & Wingfield, M. J. (2014). Redefining Ceratocystis and allied genera. Studies in Mycology, 79, 187-219.

Benko, S., Fantes, J. A., Amiel, J., Kleinjan, D., Thomas, S., Ramsay, J., et al. (2009). Highly conserved non. Nature Genetics 64(2), p. 10-12.

Calo, E., & Wysocka, J. (2013). Modification of enhancer chromatin: what, how, and why?. Molecular cell, 49(5), 825-837.

Creyghton, M. P., Cheng, A. W., Welstead, G. G., Kooistra, T., Carey, B. W., Steine, E. J., ... & Boyer, L. A. (2010). Histone H3K27ac separates active from poised enhancers and predicts developmental state. Proceedings of the National Academy of Sciences, 107(50), 21931-21936.

Cutter, A. R., & Hayes, J. J. (2015). A brief review of nucleosome structure. FEBS letters, 589(20), 2914-2922.

Diehl, A. D., Meehan, T. F., Bradford, Y. M., Brush, M. H., Dahdul, W. M., Dougall, D. S., ... & Van Slyke, C. E. (2016). The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. Journal of biomedical semantics, 7(1), 44.

Doniger, S. W., Huh, J., & Fay, J. C. (2005). Identification of functional transcription factor binding sites using closely related Saccharomyces species. Genome research, 15(5), 701-709.

Emison, E. S., McCallion, A. S., Kashuk, C. S., Bush, R. T., Grice, E., Lin, S., ... & Chakravarti, A. (2005). A common sex-dependent mutation in a RET enhancer underlies Hirschsprung disease risk. Nature, 434(7035), 857.

Ernst, J., & Kellis, M. (2012). ChromHMM: automating chromatin-state discovery and characterization. Nature methods, 9(3), 215.

Ernst, J., Kheradpour, P., Mikkelsen, T. S., Shoresh, N., Ward, L. D., Epstein, C. B., ... & Ku, M. (2011). Mapping and analysis of chromatin state dynamics in nine human cell types. Nature, 473(7345), 43.

Ezkurdia, I., Juan, D., Rodriguez, J. M., Frankish, A., Diekhans, M., Harrow, J., ... & Tress, M. L. (2014). Multiple evidence strands suggest that there may be as few as 19 000 human protein-coding genes. Human molecular genetics, 23(22), 5866-5878.

Ferguson, J. D. (1980). pp. 143–179, Variable duration models for speech. In Proc. of the Symposium on the applications of hidden Markov models to text and speech, JD Ferguson, Ed. Princeton: IDA-CRD.

Fishilevich, S., Nudel, R., Rappaport, N., Hadar, R., Plaschkes, I., Iny Stein, T., ... & Lancet, D. (2017). GeneHancer: genome-wide integration of enhancers and target genes in GeneCards. Database, 2017.

Friedli, M., Barde, I., Arcangeli, M., Verp, S., Quazzola, A., Zakany, J., ... & Duboule, D. (2010). A systematic enhancer screen using lentivector transgenesis identifies conserved and non-conserved functional elements at the Olig1 and Olig2 locus. PLoS One, 5(12), e15741.

Galperin, M. Y., & Fernández-Suarez, X. M. (2011). The 2012 nucleic acids research database issue and the online molecular biology database collection. Nucleic acids research, 40(D1), D1-D8.

Hayashi-Takanaka, Y., Yamagata, K., Wakayama, T., Stasevich, T. J., Kainuma, T., Tsurimoto, T., ... & Kimura, H. (2011). Tracking epigenetic histone modifications in single cells using Fab-based live endogenous modification labeling. Nucleic acids research, 39(15), 6475-6488.

Heintzman, N. D., Stuart, R. K., Hon, G., Fu, Y., Ching, C. W., Hawkins, R. D., ... & Wang, W. (2007). Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nature genetics, 39(3), 311.

Heintzman, N. D., Hon, G. C., Hawkins, R. D., Kheradpour, P., Stark, A., Harp, L. F., ... & Ching, K. A. (2009). Histone modifications at human enhancers reflect global cell-type-specific gene expression. Nature, 459(7243), 108.

Hu, J., Brown, M. K., & Turin, W. (1996). HMM based online handwriting recognition. IEEE Transactions on pattern analysis and machine intelligence, 18(10), 1039-1045.

Jin Q, Yu L-R, Wang L, Zhang Z, Kasper LH, Lee J-E, Wang C, Brindle PK, Dent SYR, Ge K. 2011. Distinct roles of GCN5/PCAF-mediated H3K9ac and CBP/p300-mediated H3K18/27ac in nuclear receptor transactivation. The EMBO Journal 30:249–262.

Jones, P. A. (2012). Functions of DNA methylation: islands, start sites, gene bodies and beyond. Nature Reviews Genetics, 13(7), 484.

Kaplan, T., & Biggin, M. D. (2012). Quantitative models of the mechanisms that control genome-wide patterns of animal transcription factor binding. In Methods in cell biology (Vol. 110, pp. 263-283). Academic Press.

Karmodiya, K., Krebs, A. R., Oulad-Abdelghani, M., Kimura, H., & Tora, L. (2012). H3K9 and H3K14 acetylation co-occur at many gene regulatory elements, while H3K14ac marks a subset of inactive inducible promoters in mouse embryonic stem cells. BMC genomics, 13(1), 424.

Kelley, D. R., Snoek, J., & Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome research, 26(7), 990-999.

Khan, A., Fornes, O., Stigliani, A., Gheorghe, M., Castro-Mondragon, J. A., van der Lee, R., ... & Baranasic, D. (2017). JASPAR 2018: update of the open-access database of transcription factor binding profiles and its web framework. Nucleic acids research, 46(D1), D260-D266.

Kim, T. K., Hemberg, M., Gray, J. M., Costa, A. M., Bear, D. M., Wu, J., ... & Markenscoff-Papadimitriou, E. (2010). Widespread transcription at neuronal activity-regulated enhancers. Nature, 465(7295), 182.

Kleftogiannis, D., Kalnis, P., Arner, E., & Bajic, V. B. (2016). Discriminative identification of transcriptional responses of promoters and enhancers after stimulus. Nucleic acids research, 45(4), e25-e25.

Karmodiya, K., Krebs, A. R., Oulad-Abdelghani, M., Kimura, H., & Tora, L. (2012). H3K9 and H3K14 acetylation co-occur at many gene regulatory elements, while H3K14ac marks a subset of inactive inducible promoters in mouse embryonic stem cells. BMC genomics, 13(1), 424.

Kreimer, A., Zeng, H., Edwards, M. D., Guo, Y., Tian, K., Shin, S., ... & Li, Y. (2017). Predicting gene expression in massively parallel reporter assays: a comparative study. Human mutation, 38(9), 1240-1250.

Kulakovskiy, I. V., Belostotsky, A. A., Kasianov, A. S., Esipova, N. G., Medvedeva, Y. A., Eliseeva, I. A., & Makeev, V. J. (2011). A deeper look into transcription regulatory code by preferred pair distance templates for transcription factor binding sites. Bioinformatics, 27(19), 2621-2624.

Kundaje, A., Meuleman, W., Ernst, J., Bilenky, M., Yen, A., Heravi-Moussavi, A., ... & Amin, V. (2015). Integrative analysis of 111 reference human epigenomes. Nature, 518(7539), 317.

Lee, L. M., & Lee, J. C. (2006, June). A study on high-order hidden Markov models and applications to speech recognition. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 682-690). Springer, Berlin, Heidelberg.

Lettice, L. A., Heaney, S. J., Purdie, L. A., Li, L., de Beer, P., Oostra, B. A., ... & de Graaff, E. (2003). A long-range Shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. Human molecular genetics, 12(14), 1725-1735.

Lindblad-Toh, K., Garber, M., Zuk, O., Lin, M. F., Parker, B. J., Washietl, S., ... & Ward, L. D. (2011). A high-resolution map of human evolutionary constraint using 29 mammals. Nature, 478(7370), 476.

Mari, J. F., Haton, J. P., & Kriouile, A. (1997). Automatic word recognition based on second-order hidden Markov models. IEEE Transactions on speech and Audio Processing, 5(1), 22-25.

Markov, A. A. (1906). Extension of the law of large numbers to dependent quantities. Izv. Fiz.-Matem. Obsch. Kazan Univ.(2nd Ser), 15, 135-156.

Miguel-Escalada, I., Pasquali, L., & Ferrer, J. (2015). Transcriptional enhancers: functional insights and role in human disease. Current opinion in genetics & development, 33, 71-76.

Ng, S. B., Turner, E. H., Robertson, P. D., Flygare, S. D., Bigham, A. W., Lee, C., ... & Bamshad, M. (2009). Targeted capture and massively parallel sequencing of 12 human exomes. Nature, 461(7261), 272.

Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., & Bejerano, G. (2013). Enhancers: five essential questions. Nature Reviews Genetics, 14(4), 288.

du Preez, J. A. (1998). Efficient training of high-order hidden Markov models using first-order representations. Computer speech & language, 12(1), 23-39.

Przybilla, J., Galle, J., & Rohlf, T. (2012). Is adult stem cell aging driven by conflicting modes of chromatin remodeling?. Bioessays, 34(10), 841-848.

Rabiner, L., & Juang, B. H. (1993). Fundamentals of speech processing. Prantice Hall.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.

Rada-Iglesias, A., Bajpai, R., Swigut, T., Brugmann, S. A., Flynn, R. A., & Wysocka, J. (2011). A unique chromatin signature uncovers early developmental enhancers in humans. Nature, 470(7333), 279.

Rosin, J. M., Abassah-Oppong, S., & Cobb, J. (2013). Comparative transgenic analysis of enhancers from the human SHOX and mouse Shox2 genomic regions. Human molecular genetics, 22(15), 3063-3076.

Smemo, S., Campos, L. C., Moskowitz, I. P., Krieger, J. E., Pereira, A. C., & Nobrega, M. A. (2012). Regulatory variation in a TBX5 enhancer leads to isolated congenital heart disease. Human molecular genetics, 21(14), 3255-3263.

Soldner, F., Stelzer, Y., Shivalila, C. S., Abraham, B. J., Latourelle, J. C., Barrasa, M. I., ... & Jaenisch, R. (2016). Parkinson-associated risk variant in distal enhancer of α-synuclein modulates target gene expression. Nature, 533(7601), 95.

Stadler, M. B., Murr, R., Burger, L., Ivanek, R., Lienert, F., Schöler, A., ... & Tiwari, V. K. (2011). DNA-binding factors shape the mouse methylome at distal regulatory regions. Nature, 480(7378), 490.

Taher, L., McGaughey, D. M., Maragh, S., Aneas, I., Bessling, S. L., Miller, W., ... & Ovcharenko, I. (2011). Genome-wide identification of conserved regulatory function in diverged sequences. Genome research, 21(7), 1139-1149.

Thurman, R. E., Rynes, E., Humbert, R., Vierstra, J., Maurano, M. T., Haugen, E., ... & Garg, K. (2012). The accessible chromatin landscape of the human genome. Nature, 489(7414), 75.

Turin, W., & Sondhi, M. M. (1993). Modeling error sources in digital channels. IEEE Journal on Selected Areas in Communications, 11(3), 340-347.

Visel, A., Minovitsky, S., Dubchak, I., & Pennacchio, L. A. (2007). VISTA Enhancer Browser—a database of tissue-specific human enhancers. Nucleic Acids Research, 35(Database issue), D88.

Visel, A., Blow, M. J., Li, Z., Zhang, T., Akiyama, J. A., Holt, A., ... & Afzal, V. (2009). ChIP-seq accurately predicts tissue-specific activity of enhancers. Nature, 457(7231), 854.

Williamson, I., Hill, R. E., & Bickmore, W. A. (2011). Enhancers: from developmental genetics to the genetics of common human disease. Developmental cell, 21(1), 17-19.

Zentner, G. E., Tesar, P. J., & Scacheri, P. C. (2011). Epigenetic signatures distinguish multiple classes of enhancers with distinct cellular functions. Genome research, 21(8), 1273-1283.

Zhou, J., & Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. Nature methods, 12(10), 931.