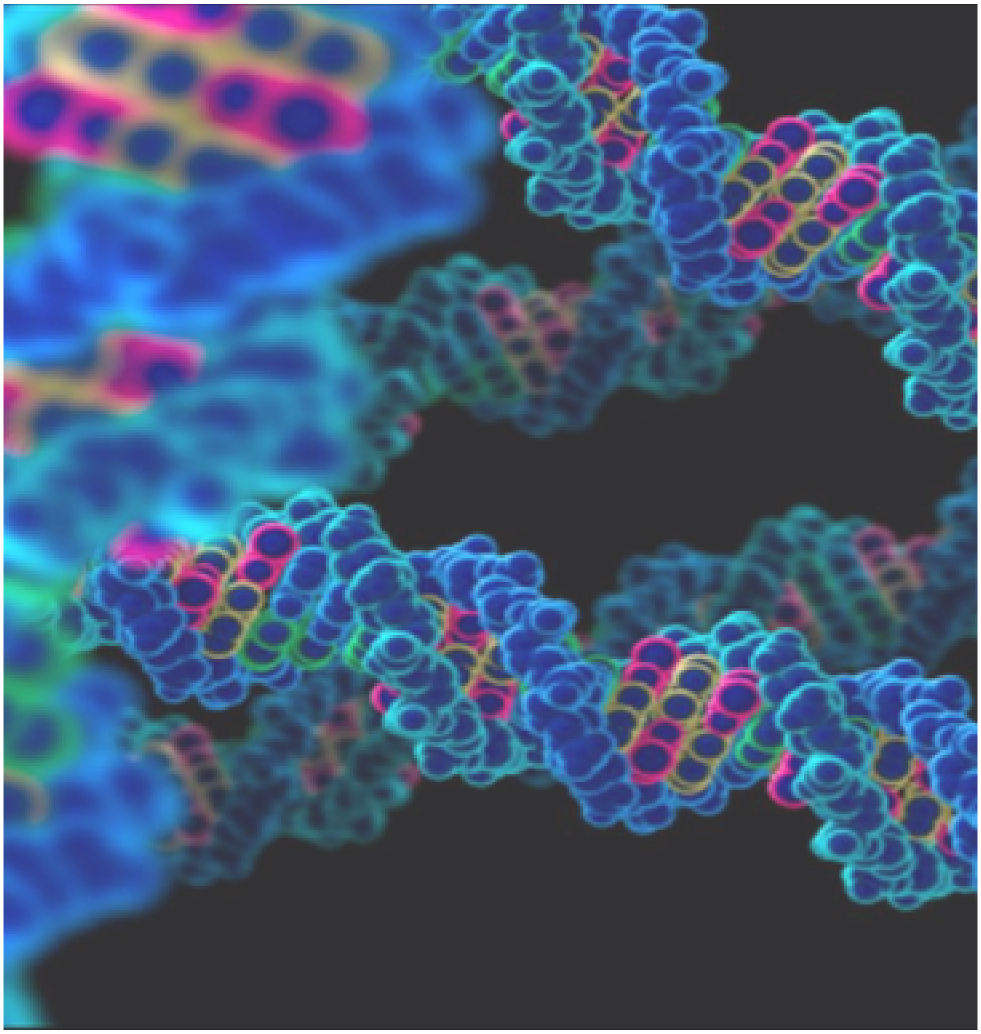
Find CpG Islands by HMM

UIUC CS 466 Project Report



Tanqiu Liu, Wenjie Zhu

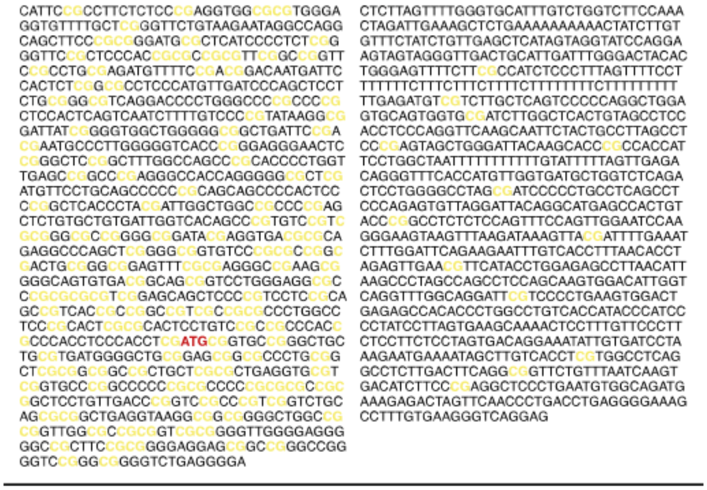
SPRING 2018

INTRODUCTION

1. CpG Island

   CpG sites are sites in DNA sequence where a cytosine (“C”) nucleotide is followed by a guanine (“G”) nucleotide[1]. (See Figure 2). Cytosine in CpG sites are usually methylated by DNA-methyltransferases. Methylated CpG sites affect the expression level of genes they related to and play an important role in gene regulation network in mammalian cells. CpG islands are regions (usually > 200 bp in length) of DNA sequence with high frequency of CpG sites.

   CpG island is usually associated with gene promoters and therefore becomes an important feature in gene/promoter prediction and epigenetic analysis.



Left: CpG sites at 1/10 nucleotides, constituting a CpG island. The sample is of a gene-promoter, the highlighted ATG consitutes the start codon.
Right: CpG sites present at every 1/100 nucleotides, consituting a more normal example of the genome, or a region of the genome that is commonly methylated.

Fig 1. CpG sites

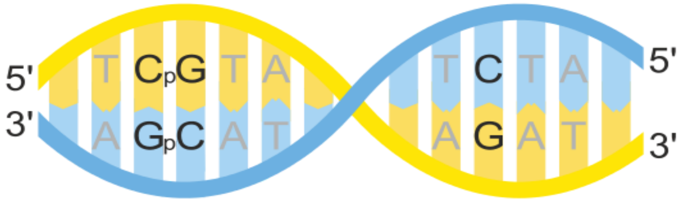


Fig 2: CpG, "—C—phosphate—G—" nucleotides on one DNA strand (left), and complementary C-G base pairing on two DNA strands (right)

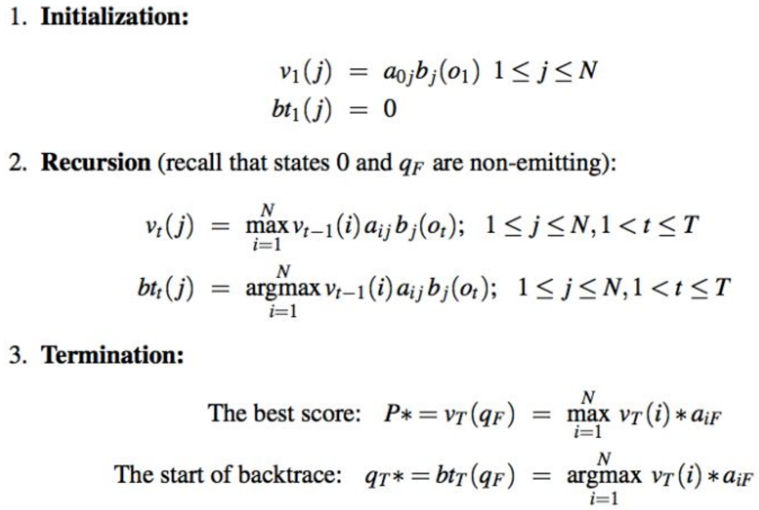
[1] https://en.wikipedia.org/wiki/CpG_site

2. HMM

The Hidden Markov Model (HMM) is a statistical model for sequences of discrete symbols. HMM is defined by Alphabets, Sequence, i-th letter in sequence, and a set of sequence. In HMM the states in the machine are not directly visible but some output at certain states are observable. Each state has a probability distribution over possible output states[2].

3. Viterbi Algorithm

Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states, for each intermediate state, until it reaches the end state. At each time only the most likely path leading to each state survives. From the lecture we learnt detailed viterbi algorithm defined in following:



1. **Initialization:**

$$\begin{aligned} v_1(j) &= a_{0j} b_j(o_1) \;\; 1 \le j \le N \\ bt_1(j) &= 0 \end{aligned}$$

2. **Recursion** (recall that states 0 and q_F are non-emitting):

$$\begin{aligned} v_t(j) &= \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T \\ bt_t(j) &= \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T \end{aligned}$$

3. **Termination:**

$$\text{The best score:} \quad P* = v_T(q_F) = \max_{i=1}^{N} v_T(i) * a_{iF}$$

$$\text{The start of backtrace:} \quad q_T* = bt_T(q_F) = \operatorname*{argmax}_{i=1}^{N} v_T(i) * a_{iF}$$

Fig. Viterbi Algorithm

4. Project Goal

Our goal of this project is to find the existences of CpG island structures in our sample data sequence, by applying Hidden Markov Model and Viterbi algorithm. Finally we will compare our detected results with ground truth and then find out ways of further improvements.

DATA

We used human genome assembly hg38 with annotation to build the training sequence data and CpG island label for our project which is shown as following:

[2] http://www.cs.ubbcluj.ro/~csatol/mach_learn/bemutato/Mate_Korosi_HMMpres.pdf

Sample CpG annotation:

bin	chrom	chromStart	chromEnd	name	length	cpgNum	gcNum	perCpg	perGc	obsExp
585	chr1	28735	29810	CpG: 116	1075	116	787	21.6	73.2	0.83
586	chr1	135124	135563	CpG: 30	439	30	295	13.7	67.2	0.64
587	chr1	327790	328229	CpG: 29	439	29	295	13.2	67.2	0.62
588	chr1	437151	438164	CpG: 84	1013	84	734	16.6	72.5	0.64
588	chr1	449273	450544	CpG: 99	1271	99	777	15.6	61.1	0.84
589	chr1	533219	534114	CpG: 94	895	94	570	21	63.7	1.04
589	chr1	544738	546649	CpG: 171	1911	171	1405	17.9	73.5	0.67
590	chr1	713984	714547	CpG: 60	563	60	385	21.3	68.4	0.92
590	chr1	762416	763445	CpG: 115	1029	115	673	22.4	65.4	1.07
591	chr1	788863	789211	CpG: 28	348	28	192	16.1	55.2	1.06

This dataset can be downloaded at: *http://hgdownload.soe.ucsc.edu/downloads.html*.

METHOD & IMPLEMENTATION

1. Data Preprocessing

   We used Python to implement this project. The predictCPG.py is the main function to run this project.

   Usage: python predictCPG.py <train_start> <train_end> <test_start> <test_end>

   e.g. python predictCPG.py 2500000 3500000 1500000 1600000

   Typically, test sequence should be long enough to make sure there is CpG islands in it.

   .fasta sequence file is loaded with package biopython. CpG annotation file is parsed into a pandas DataFrame and a new CpG annotation data frame is created according to the training and testing sequences specified in the parameters.

We used ten possible states in total: {A+, A-, T+, T-, G+, G-, C+, C-, N+, N-}, where +/- is used to indicate if it is in CpG island or not. N+ and N- normally appear only in start and ending region of the chromosome and we should avoid incorporating N’s in either training sequence and testing sequence.

2. Transition Probability & Prior Probability

   We go through the training sequence to count the transition frequencies and nucleotides frequencies (see function getFreq). The probabilities are convert into log base to avoid numerical underflow. (See function getLogTransitionProb and getLogPriorProb)

3. Emission Transition Probability

   By observation, the emission transition probability from A+/A- is:

   $$b_{A^+}(A) = 1,\ b_{A^+}(T) = 0,\ b_{A^+}(G) = 0,\ b_{A^+}(C) = 0,\ b_{A^+}(N) = 0$$

   $$b_{A^-}(A) = 1,\ b_{A^-}(T) = 0,\ b_{A^-}(G) = 0,\ b_{A^-}(C) = 0,\ b_{A^-}(N) = 0$$

   Where the probability from A+/A- to A is 1 and for 0 otherwise. Similar for other states.

4. Viterbi Algorithm

   Since the emission transition matrix only contains a few non zero entries regularly, to reduce unnecessary computations in Viterbi, we did not store the emission transition as actual matrix form, but a simple state dictionary. Specifically, when we do computation steps in Viterbi, we assign rest of states to be negative infinity and only access to the two nonzero states through the state dictionary.

   The viterbi takes three inputs: sequence, log transition probability, and log prior probability, and returns an optimal path and the correspond best score. We used one matrix to compute the probability for each reachable state, and one matrix to store previous best path for each state. After we get the resulted path, then we trace back determine the optimal path and convert the path into the form of CpG annotation. (See function path2cpg, getCpgInfo). The result annotation is stored in /output folder.

5. IOU & Scoring Matrix

Our scoring method is based on IOU (Intersection Over Union) which compares the predicted results with the ground truth. IOU is defined as following:

$$IOU = \frac{DetectionResult \bigcap GroundTruth}{DetectionResult \bigcup GroundTruth}$$

When the IOU is higher, it implies result is more accurate. In best case, when detected result is exactly same as grounded truth, IOU = 1.

Based on that, we set a threshold = 0.5 as definition of “hits”, i.e. the True Positive (TP) predictions; For regions in ground truth that do not have a corresponding predicted region with IOU > threshold is counted as a False Negative (FN); For regions in predicted regions that do not have a corresponding ground truth region with IOU > threshold is counted as a False Positive (FP).

It follows that our score of the detection is defined as following:

$$Detection\ Score\ = \frac{TP}{TP + FP + FN}$$

, which will be used as evaluate metric for overall accuracy of our results.

RESULTS & CONCLUSION

We run a few data set to see the performance of our implementation:

1) Human genome assembly hg38:

   train sequence: length 200000 to 300000;

   testing sequence: length 100000 to 110000.

   Our result of CpG Islands displays below:

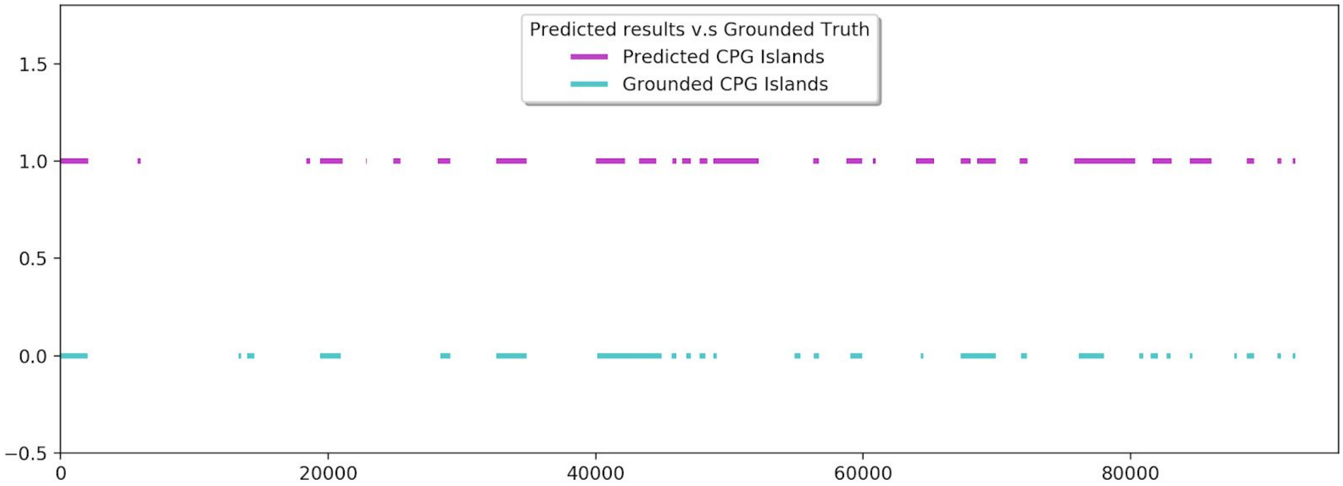


Fig. Predicted CpG v.s Grounded Truth

In this dataset, we achieved evaluation score about 0.709 with our evaluation metric defined above.

2) Human genome assembly hg38:

train sequence: length 250000 to 350000;

testing sequence: length 160000 to 170000

Our result of CpG Islands displays below:

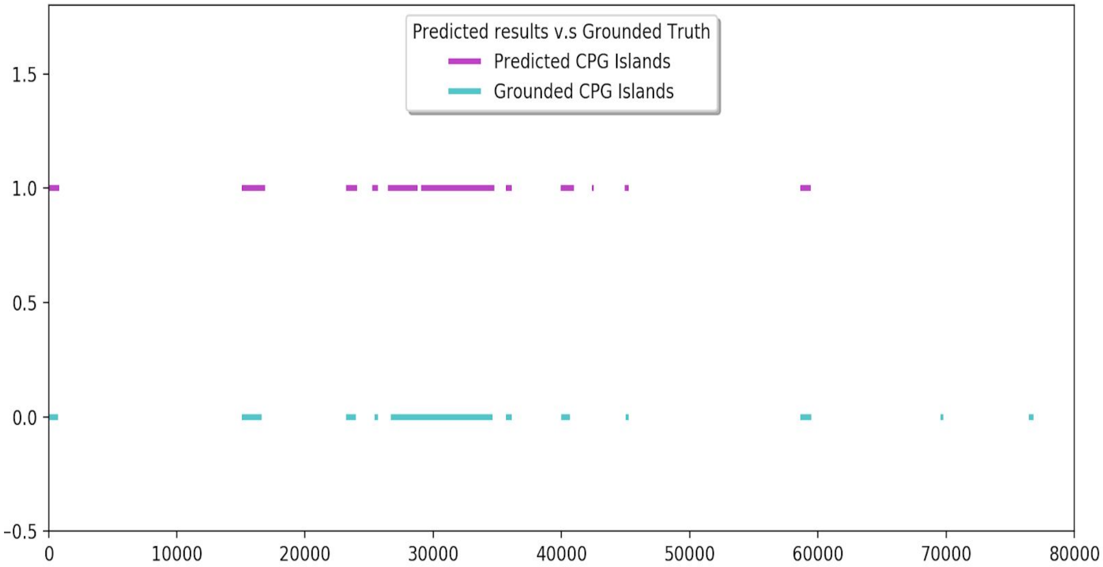


Fig. Predicted CpG v.s Grounded Truth

In this dataset, we achieved evaluation score about 0.733.

3) Human genome assembly hg38:

   train sequence: length 250000 to 350000

   testing sequence: length 150000 to 160000

   Our result of CpG Islands displays below:

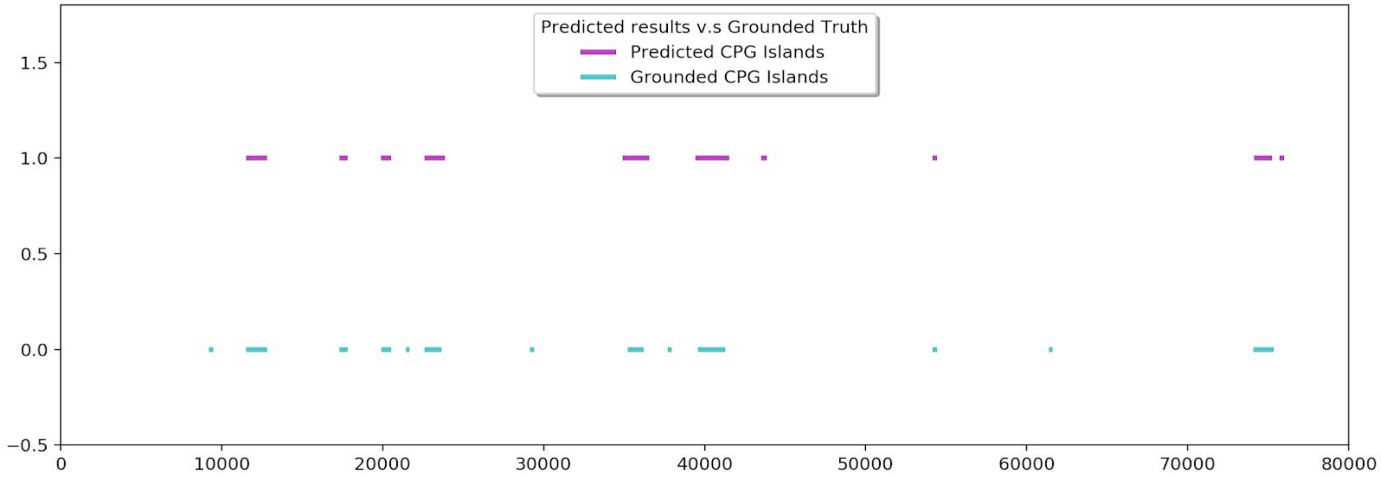


Fig. Predicted CpG v.s Grounded Truth

In this dataset, we achieved evaluation score about 0.529.

REFERENCES

1. Lecture notes, Hidden Markov Model, by Jian Peng, http://courses.engr.illinois.edu/cs466/sp2018/slides/Lecture28_HMM4.pdf, 2018
2. CpG_Site, *Wikipedia*, https://en.wikipedia.org/wiki/CpG_site
3. Lecture notes, HMM, by Máthé Zoltán K őrösi Zoltán, http://www.cs.ubbcluj.ro/~csatol/mach_learn/bemutato/Mate_Korosi_HMMpres.pdf, 2006
4. Intersection over Union (IoU) for object detection, by Adrian Rosebrock, https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-objec*t-detection/, 2016*
5. Schema for CpG Islands - CpG Islands (Islands < 300 Bases are Light Green), *h*ttp://genome.ucsc.edu/cgi-bin/hgTables?db=hg38&hgta_group=regulation&hgta_track=cpgIslandExt&hgta_table=cpgIslandExt&hgta_doSchema=describe%20table%20schema 1987