CpG Island Finding Using Graphical Models

Gang Ji and Tim Ng
{gang,tng}@ee.washington.edu
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

Lingyun Huang
huangly@u.washington.edu
Department of Bioengieering
Univesity of Washington
Seattle, WA 98195

UW CSE527 Project Report
Number CSE527-2004-0001
December 2004

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

CpG Island Finding Using Graphical Models

Gang Ji and Tim Ng
{gang,tng}@ee.washington.edu
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

Lingyun Huang
huangly@u.washington.edu
Department of Bioengieering
Univesity of Washington
Seattle, WA 98195

University of Washington, Dept. of EE, CSE527-2004-0001
December 2004

Abstract

CpG islands are short stretches in DNA sequence whose frequency of cytosine(C)and guanine (G) is higher than background of DNA sequence. They are around the promoter of frequently expressed genes. The conventional way to recognize CpG islands is to use the hidden Markov models (HMMs). While HMMs are known to suffer from not being able to capture long dynamic range information, they usually doesn't provide satisfying results.

In this work, we will try to find CpG islands with improved HMM systems (by means of introducing language model weights) as well as other family of graphical models: dynamic Bayesian networks (DBNs) [11] and conditional random fields (CRFs) [9]. By using different weights to different kinds of links in an HMM, we can get some improvements on the recognition. Significant improvements can be achieved by adding dependencies to the observation variables and thus change the structure of graphical models.. The newly developed gene-trigram model can reduce the equal error rate by 42.3% relatively to baseline system. Unfortunately, even though CRFs show big benefit in tasks like text segmentation and part-of-speech tagging, it didn't recognized any CpG island in our preliminary experiments.

1 Introduction

CpG island is a short stretch of DNA where the frequency of the occurrence of cytosine (C) and guanine (G) is higher than the frequency of C and G in other parts of DNA sequence[3]. CpG islands are longer than 200 base pairs (bp) and have over 50% of G+C content and frequency, at least 0.6 of that statistically expected[7]. The "p" between C and G only indicates that C and G are connected by phosphodiester bond.

Hidden Markov Model (HMM) is used to model and recognize the CpG island subsequence[5]. An HMM is shown in Figure 3 where the hidden states form a Markov chain and emit different observations according to some distribution.

An underlined Markov chain is constructed by eight hidden states $A+, A-, C+, C-, G+, G-, T+, T-$. The transition probability between hidden states and the emission probability between states and observed nucleotides can

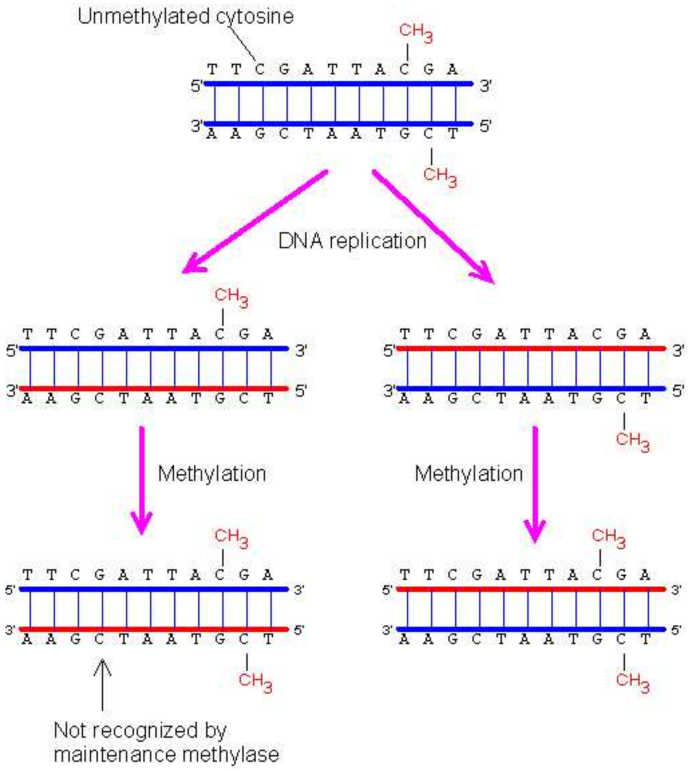


Figure 1: Methylated cytosine

be acquired during training by labeled DNA sequence. The recognition procedure, which is usually called decoding in the application, is implemented by Viterbi algorithm.

The training is usually carried out with standard expectation-maximization (EM) algorithm and decoding is calculating the best state sequence by the so-called Viterbi algorithm[8]. One advantage of such HMMs is it is very efficient to train and inference on them. Despite the fact that only first order Markov chain is used in hidden nodes, they usually provide good results in practice in domains such as part-of-speech tagging.

However, the conditional independence assumptions made from HMMs are sometimes too strong. For an example, it states that the genes at different frame in a sequence are independent of each other if the hidden states are known. Therefore, HMMs cannot capture long dynamic range relations quite well (the so-called duration problem).

One typical solution in speech community is adding language model weights into the system. By doing this, links with different confidence in prediction will have different weights in inference. Our results show that by adding this language model weights, we can get some improvement in the CpG island finding.

A more direct way to resolve the strong conditional independence statements is to add links between desired random variables. This leads us into the dynamic Bayesian networks family[11] where HMM is a special example. Our results show that by simply adding bigram/trigram links in the gene sequence random variables, we can get a huge improvement in the CpG island prediction.

We could take this one step further and look models not in the Bayesian network family but other general graphical models. Conditional random fields (CRFs) [9] have shown great advantages in analyzing sequential data such as name entity recognition or part-of-speech tagging. We can treat CpG island finding as either a segmentation task or a tagging task. In our preliminary work, however, CRF didn't find any CpG island as we expected.

The project report is organized as follows: Section 2 gives the properties of evaluation data corpus with scoring

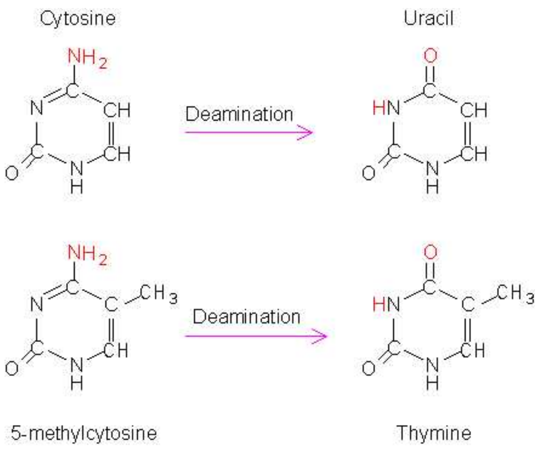


Figure 2: Deamination of cytosine

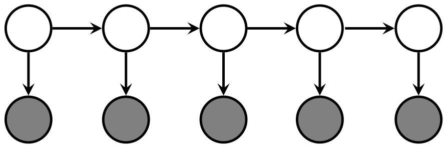


Figure 3: Hidden Markov models

metric. Section 3 provides the details and performance of introducing language model weights into the HMM systems. Section 4 presents dynamic Bayesian networks and their improvements over HMM baseline systems. Finally, Section 5 shows conditional random fields and details of experiments.

2 Data Corpus and Evaluation Metric

2.1 Data Corpus

We retrieved both gene sequences and CpG island specifications from EMBL-EBI (European Biological Institute). The one we utilized contains 1710 gene segmentations. 90% of the data is used as training set and the rest 10% is used as evaluation set. The gene sequences have been segmented by human to indicate some specific functions. Table 1 shows some statistics of our data corpus.

Table 1: CpG island corpus

	CpG length	DNA length
maximum	2240	185775
minimum	181	44
mean	465	3787

2.2 Provision and Recall

So far, there is no good quantitative metric for measuring the performance of CpG island finding systems. Here in this project, we propose to use the precision/recall scheme as illustrated in Figure 4.

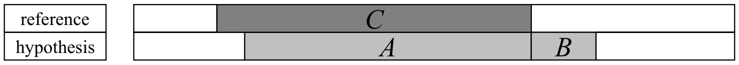


Figure 4: Scoring CpG island finding performance.

In Figure 4, the upper region shows the reference and the lower region shows the testing hypothesis. There is a CpG island in the gray area (C) and the hypothesis thinks that $A+B$ is an island. There are two kind of errors, namely false positive (type II error) and false negatives (type I error). In our experiments, precision and recall are defined as

$$\begin{cases} P & \triangleq & \frac{A}{A+B}, \\ R & \triangleq & \frac{A}{C}. \end{cases} \tag{1}$$

According to this definition, precision gives among all those retrieved island states, how many are really true and recall gives among all real island states, how many are correctly retrieved.

Furthermore, the F-measure of a system is defined as the harmonic mean of precision and recall:

$$F \triangleq \frac{PR}{P+R}. \tag{2}$$

2.3 ROC Curve

Most of the time, a information retrieval system has a trade-off between precisions and recall. For an example, we can classify all states to be an island state so that the recall is 100%, but with a very low precision rate. Receiver Operating Characteristic (ROC) curve gives the overall performance of those systems. Figure 5 gives two toy ROC curves of two different system on a same task.

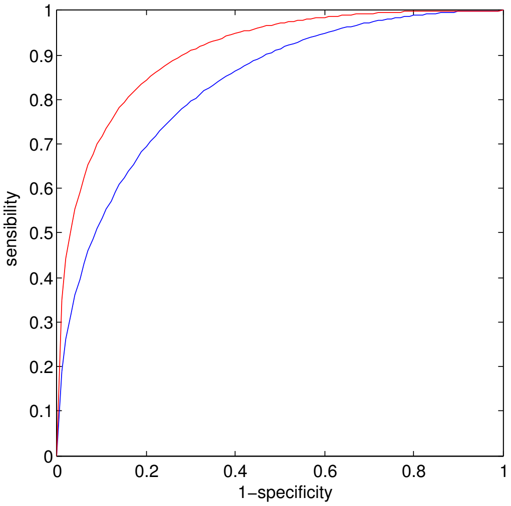


Figure 5: Examples of ROC curve

In Figure 5, horizontal axis gives 1 minus specificity (precision) and the vertical axis gives sensitivity (recall). In the toy example we show, the red curve is better than the blue one because when the two system have the same rate of precision, the red one has higher recall. On the other hand, when the two systems have the same rate of recall, the red one has better precision.

In order to calculated a quantitative comparison of two systems, people have introduced the equal error rate (EER). The EER of ROC is defined when the precision equals recall ($P^* = R^*$):

$$EER \triangleq 1 - P^* \equiv 1 - R^* \tag{3}$$

3 Language Modeling in HMM

3.1 Motivations

Hidden Markov Model (HMM) has been commonly used in many areas of pattern recognition. Especially in speech recognition, it has been proved to be a promising approach. However, there are some drawbacks of using conventional HMM. State occupancy in HMM decreases exponentially with time: $d_i(t) = a_{ii}^t(1 - a_{ii})$. Therefore, HMM favorites more insertions and has poor duration modeling ability. On the other hand, the conventional HMM assumes the observations only depend on the states that generate them. Hence, it is difficult to use the conventional HMM to model the non-stationary with high correlation between the observations.

In speech recognition, language models have been used to relief these constrains of the HMM and it has been proved to be a promising method. In this section, we are going to investigate the effect of language model on the task of CpG island detection.

3.2 Language Models in HMM

We employed the open source software Hidden Markov Model Toolkit (HTK), which is a powerful and famous utility for speech recognition, to training our HMMs and language model. It has been used to do the decoding as well. The procedure for training using HTK is as follows:

1. Map the symbols (a, c, g, t) into indices, that were our discrete features.

2. Define the topology for the HMMs. The topologies of the HMMs are as in Figure 6.

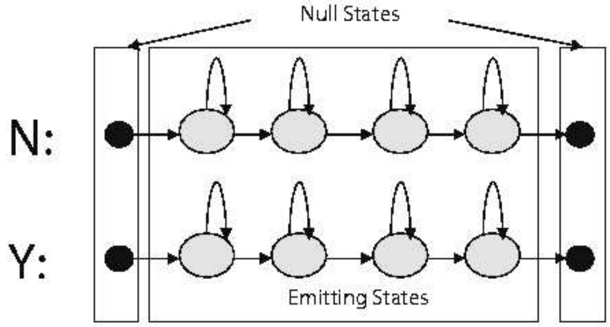


Figure 6: State transition topology for HMM

   where N is the HMM for non-island observation sequences and Y is for island ones. The HMMs here are all strictly left-to-right without jumping transitions. This topology is also commonly used in speech recognition. Since the observations are discrete, the HMMs are discrete HMMs.

3. In our training data, the model alignments are given and only the state sequences are hidden. Thus, Baum-Welch (Forward-Backward) is performed within the models [14].

The cost function for decoding with HMM and language models are as follows:

$$\begin{aligned}\hat{W} &= \underset{W}{\text{argmax}}\ P(W|O) \\ &= \underset{W}{\text{argmax}}\ \frac{P(W)P(O|W)}{P(O)} \\ &= \underset{W}{\text{argmax}}\ P(W)P(O|W) \\ &= \underset{W}{\text{argmax}}\ [\log P(W) + \log P(O|W)],\end{aligned} \tag{4}$$

where W is the sequence of N's and Y's, or island and non-land labels. O is a given observation sequence, the symbols of a, c, g and t, $\log P(W)$ is the probability for a sequence of class labels, and $\log P(O|W)$ is the probability of O given class label sequence W, which is a probability from the HMMs in our case.

However, due to the nature of Markov and the independence assumptions of HMM, $\log P(O|W)$ is usually underestimated. Equivalently speaking, the probability from the language model is over emphasized in the cost function. A parameter, language model weight (LW), was introduced to balance the two probability quantities. Thus the cost function became:

$$\hat{W} = \underset{W}{\text{argmax}}\ [LW * \log P(W) + \log P(O|W)], \tag{5}$$

where LW is tunable and usually greater than 1.0 and it is task dependent.

3.3 Experiments

3.3.1 Baseline System

Our baseline system here is a HMM system without language model. No language model means all sequences are equally likely. The parameter WP is a tunable parameter to trade-off between insertion and deletion.

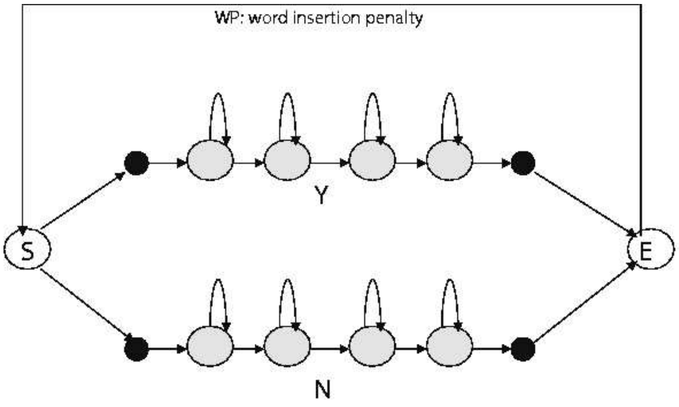


Figure 7: Baseline decoding diagram

In our experiments the word insertion penalty for our baseline system is 1.0.

3.3.2 Bigram Language Model

The bigram assumption says

$$P(W) \approx P(w_1)P(w_2|w_1)\ldots P(w_n|w_{n-1}).$$

As shown in Figure 8, At each frame, a language model weight is added into the log likelihood for language score. By doing this, we apply different weights into transition and emission scores.

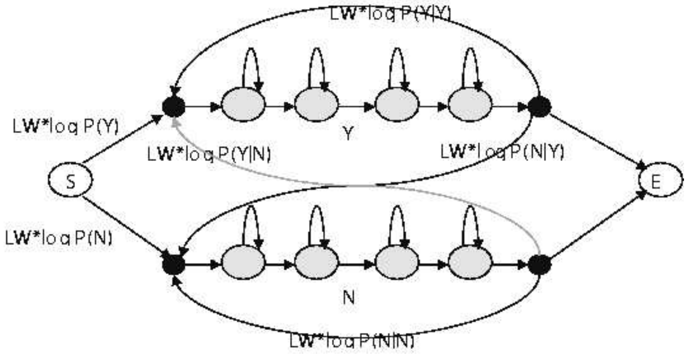


Figure 8: Decoding diagram with language model

3.3.3 Results

In Table 2 we show the results for applying language model weights in to standard HMMs.

Table 2: Language model results

	precision	recall	F-measure
baseline	29.5%	77.7%	0.214
LM bigram	36.3%	75.0%	0.245

From the Table 2 we see that by applying language model weights into standard HMM, one can get a higher precision while maintain recall to be roughly the same. Therefore, the overall F-measure is better in LM bigram case.

4 Dynamic Bayesian Networks

In this part of the project, we will look at the disadvantages of HMMs and find better model in the DBN family.

4.1 Motivation

In Section 3, we have tried introducing language model weights into CpG island finding HMM systems. There are two kinds of links in HMM models, the one for hidden state transitions, and the one for state emissions. These two kinds of links have different effects on the CpG island prediction. Therefore, the scores come from these two kinds of links will have different confidence level. For an example, since HMM doesn't capture good dynamic behavior, we might trust more on the emission probabilities more than the transition probabilities. The idea of the work in Section 3 is therefore applying different weights on the likelihood scores from different types of links. As we can see from the results, we do get some improvements with this technique.

There are several issues with this approach. First of all, by replacing the log likelihood

$$\sum_i [\log P(w_i|w_{i-1}) + \log P(o_i|w_i)]$$

by

$$\sum_i [\lambda_{LM} \log P(w_i|w_{i-1}) + \log P(o_i|w_i)],$$

where w_i is the hidden state (island or not) and o_i is the gene observation, one has implicitly replaced the well defined probability distribution

$$P(W, O) = \prod_i P(w_i|w_{i-1}) P(o_i|w_i)$$

by

$$\prod_i [P(w_i|w_{i-1})]^{\lambda_{LM}} P(o_i|w_i),$$

which is no longer a probability distribution. Therefore, even though the motivation is clear, it doesn't have a consistent mathematical explanation.

Furthermore, the techniques in Section 3 is still in the HMM family, which still suffers from the strong conditional independence assumptions. One example,

$$O_t \perp\!\!\!\perp O_{\hat{t}} | W_t,$$

says that the gene sequence is independent of each other give the hidden island state is known. Therefore, given the hidden states, the dynamic behavior of the gene sequence is entirely ignored. Since we expect the gene sequence usually has very long dynamic range, this assumption is not appropriate. In order to solve this problem, we propose to use more general dynamic Bayesian networks as described next.

4.2 Dynamic Bayesian Networks

A graphical model[10, 6] is a graph with special semantics. The nodes of the graph represent random variables and the edges of the graph encode the factorization of the overall probability.

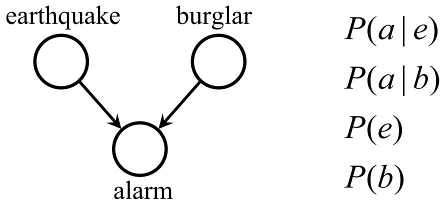


Figure 9: A simple graphical model example

Figure 9 gives a simple example. There are three random variables: “earthquake”, “burglar”, and “alarm”. The graph encodes that both “earthquake” and “burglar” and trigger “alarm”, and “earthquake” is independent of “burglar”. The right of the figure gives all the necessary parameters to describe this mechanism.

There are several kind of graphical models: directed graphical models, or Bayesian networks where the graphs are directed acyclic graphs; undirected graphical models, or Markov random fields, are undirected graphs; and the mixture of these two. In this work, we will focus on a special kind of Bayesian network called dynamic Bayesian networks (DBNs) [11] and a special kind of Markov random fields called conditional random fields.

When dealing with sequential data, such as speech recognition and gene sequence analysis, we need a mechanism to extend the idea of Bayesian networks into the whole sequence. Dynamic Bayesian networks (DBNs) [11] shown in Figure 10 is such an extension.

In our example, each DBN has a prologue, the first frame of the graph, an epilogue, the last frame, and a chunk

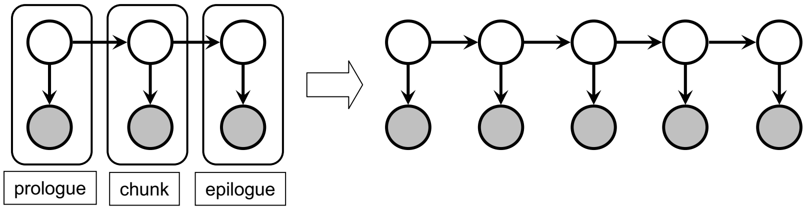
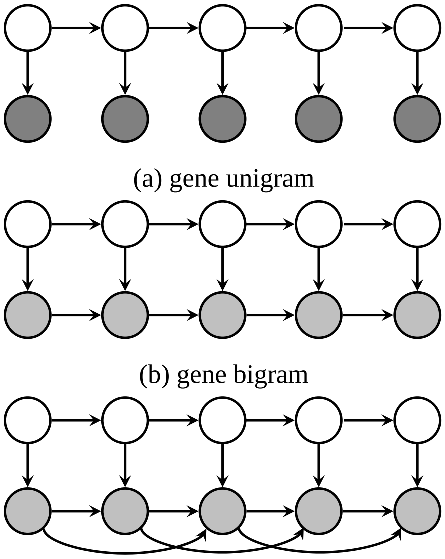


Figure 10: Dynamic Bayesian networks framework

which repeats itself to fit the length of the whole sequence. Recall the properties of an HMM, it is a special example of DBN.

4.3 Different DBNs for CpG Island Finding

As discussed before, the problem of HMM system is the independence assumption of gene sequences when the island state is known. In order to solve this, one solution can simply be adding dependencies in the gene sequences as shown in Figure 11.



(a) gene unigram

(b) gene bigram

(c) gene trigram

Figure 11: Different DBNs for CpG island finding

In Figure 11, we propose several DBNs for finding CpG islands. The first model “gene unigram” is the baseline HMM system. In “gene bigram” and “gene trigram” each gene observation is depended on the one/two genes. By adding these links, the genes are no longer independent of each other even the hidden island state is known and therefore capture more dynamic information from the gene sequence than a simple HMM.

4.4 Training and Decoding by GMTK

The training and inference in general graphs are well developed and can be found in most literatures such as[10]. In this work, we employed the graphical models toolkit (GMTK)[1] to evaluate our models. GMTK was developed in the Signal Speech and Language Interpretation Lab in the University of Washington. It can handle all kinds of complicated graphical models.

When training the graphical models, standard expectation-maximization (EM) algorithm[4] is used. At initial stage, a guess of hidden states is assigned. Then the parameters are trained using this guess. After that, a better hidden state assignment is given using trained parameters. This procedure is repeated until some criteria for convergence.

4.4.1 Junction Tree Algorithm

The inference of the graph is based on junction tree algorithm[2]. The procedure of Viterbi algorithm can be interpreted as a message passing along the HMM. This message passing scheme only works on a tree (HMM is a tree). For more general graphical models, the idea is translate the graphical models into a junction tree where the nodes are a clique of random variables. Message passing along this junction tree can guarantee the correct inference given some evidence.

The steps of a junction tree algorithm for a Bayesian network is

1. Moralisation: For each node, if the two parents are not connected, connect them with an undirected link. After that, drop all the arrows of all the links.

2. Triangulation: Add links to the graph so that all cycles with length greater than 3 have at least an arch.

3. Construct a junction tree from the triangulated graph where the nodes of the tree are the cliques of the graph and separators of the junction tree are the separators of the cliques.

4. Perform message passing: Distribute and collect all evidences on the junction to inference the hidden nodes.

4.5 Evaluations

We applied our models in the same task in Section 3. There are 8 hidden states and 4 observations (A, C, G, T). Unlike in Section 3, where feed-forward HMM is used, here an upper triangle hidden state transition probability matrix is utilized within island or non-island states. Only the last state of island states can transit into the first state of non-island states and only last state of non-island states can transit into the first state of island states. In other words, the hidden state transition matrix for 4 states (1-4) in island and 4 states (5-8) in non-island is something like:

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} & 0 & 0 & 0 & 0 \\ 0 & p_{22} & p_{23} & p_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{33} & p_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{44} & p_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{55} & p_{56} & p_{57} & p_{58} \\ 0 & 0 & 0 & 0 & 0 & p_{66} & p_{67} & p_{68} \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{77} & p_{78} \\ p_{81} & 0 & 0 & 0 & 0 & 0 & 0 & p_{88} \end{pmatrix}$$

This is an extension of the work in Section 3 where it can be shown that the feed-forward HMM is just by setting

$$p_{13} = p_{14} = p_{24} = p_{57} = p_{58} = p_{68} = 0.$$

As we will see later, this extension of transition matrix will give better baseline because there is more flexibility in the model.

The model was trained by standard EM and converged after 4 iterations. After decoding, the results are shown in Figure 12.

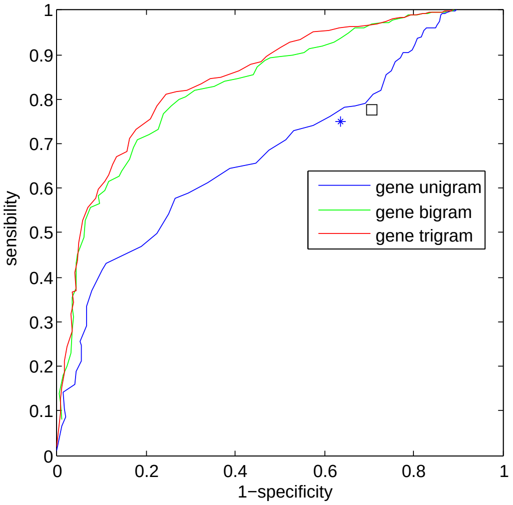


Figure 12: ROC of different DBNs for CpG island finding

In Figure 12, the star point is the HMM baseline created by HTK using feed-forward diagram. The square point is the one with language model weight applied from Section 3. The blue curve is our HMM baseline system. The green line shows the result for gene-bigram model and the red curve gives the result for gene-trigram model.

First of all, our baseline result is better than Section 3 as they are "under the blue curve". The reason is that we used a better hidden state transition diagram. One can clearly see that we get a big improvement by adding just one link in the gene sequence. Further improvement can be reached by adding a trigram link in the gene sequence.

Table 3: Equal error rate for different DBNs

model	EER	rel. imp.
unigram	38.8%	-
bigram	25.5%	34.3%
trigram	22.4%	42.3%

In Table 3, we show the equal error rates of different DBN models in our experiment. The first row gives unigram which is our HMM baseline result. The relative improvement over baseline is provide in the last column of the table. From the table, we can see that by adding one link in the gene observation nodes, the equal error rate can be reduced by 34.3% relatively. The gene-trigram model can reduce the equal error rate as significant as 42.3%.

4.6 Discussions

We have shown big improvement by changing the structure of DBNs. There are other things we can do. When specifying a DBN, there are two things need to be clear: the meaning of the nodes, and the representations of the links. In our work, we have fixed our representation that the observation nodes are just gene sequences, A, C, G, T. Another possible representation is using features with dynamic information, an idea borrowed from speech recognition (delta's and double delta's). In this case, we can use the feature as $_{C-}A_{+T}$ which means the current gene is A, and it is preceded by C and followed by T. With this representation, we captured some dynamic information of the sequence without referring to gene bigram or gene trigram links.

In our frame work, we fixed the number of hidden states to be 8, the value used in most literatures. We can also change the number of hidden states to see the effect of this on the performance. On the other hand, the hidden state transition diagram is also important. We have used upper triangle matrix for transition probability table in our DBN work. We can also try something like in HTK where feed-forward transition table is used.

5 Conditional Random Fields

5.1 Introduction of conditional random field

In the tasks of labeling a set of sequential observation, HMM is widely used and associated with strong conditional independent hypothesis which assumes the observations are totally independent. HMM is a form of generative model by defining a joint probability $p(X, Y)$.It is computationally intractable to enumerate all the possible observation sequences so that the observations are only determined by the hidden states and independent with each other in HMM.

CRF, on the other hand, chooses the conditional probability $p(Y|X)$ instead of joint probability $p(X, Y)$, where a new observation sequence x is labeled by a state sequence y when y maximize the conditional probability $p(y|x)$. This property ensures that any attribute of both observation sequences and state sequences can captured by the model. Furthermore, CRF is an undirected graphic model compared to other directed conditional Markov graphical model, such as Maximum Entropy Markov models (MEMM), which avoids the label bias problem in those models[9, 12]

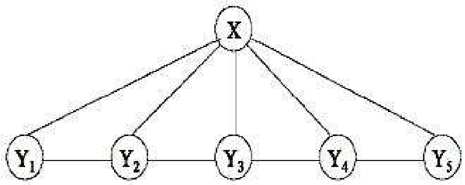


Figure 13: Illustration of CRF

5.2 Probabilistic structure of CRF

For X is a random variable over observation sequence and Y is a random variable over state sequence. $Y = Y_{v,v \in V}$, indexed by the vertexes of $G(V, E)$, if $p(Y_v|X, Y_w, w \neq v = p(Y_v|X, Y_w, w \in ne(v))$, then (X, Y) forms a conditional random field.

The graphical structure of CRF is represented by potential functions. Each potential functions operates on a clique of neighbored vertexes in $G(V, E)$. An isolated potential function itself does not have a direct probabilistic interpretation but represents constraints on the configurations of defined random variables.

In [9], each potential function is defined as the following

$$\exp\left(\sum_{e \in E_j} \lambda_j t_j(e, y|e, x) + \sum_{v \in V, k} \mu_k s_k(v, y|v, x)\right), \tag{6}$$

where $s_k(v, y|v, x)$ is called state feature, which represents the state at k-th vertex and the entire observation sequence $t_j(e, y|e, x)$ is called transition feature between adjacent states at j-th edge of $G(V, E)$ and the entire observation sequence.

$s_k(v, y|v, x)$ and $t_j(e, y|e, x)$ are also called local feature functions. A global feature function is defined by the summation of local feature functions

$$F_j(y, x) = \sum_j t_j(e, y|e, x) + s_j(v, y|v, x). \tag{7}$$

The probability of a state sequence y given an observation sequence x is written as

$$p(y|x, \lambda) = \frac{1}{Z(x)} \exp\left(\sum_j \lambda_j F_j(y, x)\right), \tag{8}$$

where $Z(x)$ is a normalization factor.

5.3 Training and Decoding of CRF

With training labeled data $\{x^{(k)}, y^{(k)}\}$ and the probability function as defined in Equation 9, the parameters is derived by maximum likelihood function.

$$L(\lambda) = \sum_k \left[\log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k),x^{(k)})}\right] \tag{9}$$

To maximize the likelihood function, the likelihood function is differentiated as

$$\frac{\partial L(\lambda)}{\partial \lambda_j} = E_{\tilde{p}(Y,X)}[F_j(Y, X)] - \sum_k E_{p(Y|x^{(k)},\lambda}[F_j(Y, x^{(k)})],$$

where $\tilde{p}(Y, X)$ is the empirical distribution of training data $E_p[\cdot]$ and denotes expectation with respect to distribution p [12]. In practice, the feature in a high dimensional vector space so that it is computationally intensive to tune all the parameters at the same time. Only one parameter is tuned to find the maxima first with all other parameters fixed, then the next parameter is tuned as the first one and so on [9, 13], this strategy is called generalized iterative scaling.

After training, the observation sequence will be input into the CRF with tuned parameters. The state sequence y which can maximize the likelihood function will be assigned. This is shown in the following formula

$$\hat{y} = \underset{y}{\text{argmax}}\, p_\lambda(y|x) = \underset{y}{\text{argmax}}\, \lambda F(y, x).$$

The decoding procedure is explained in [9, 13], by Viterbi algorithm.

5.4 Result of experiment

The program we used is the CRF toolkit on `http://crf.sourceforge.net` by Dr. Sunita Sarawagi in IIT Bombay which is based on [13].

During training in this project, each DNA sequence is truncated every 100 bp or when meeting the nucleotide belonging to different region (non-CpG island or CpG island). For testing, we inputted truncated DNA subsequence or the whole DNA sequence. The result is disappointed for none of the CpG island was picked up by the CRF toolkit. The reason for the result is possibly two fold. One is that we are not quite familiar with the source code of the toolkit for it is a large program in Java, the other is that the truncating strategy might not fit the toolkit.

References

[1] Jeff Bilmes. *The Graphical Models Toolkit*, 2004.

[2] Robert G. Cowell, A. Philip Dawid, and Steffen L. Lauritzen. *Probabilistic Networks and Expert Systems (Statistics for Engineering and Information Science)*. Springer-Verlag, 1999.

[3] DNA methylation and CpG islands, from the link http://www.web-books.com/mobio/free/ch7f2.htm.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B. 39:1–38, 1977.

[5] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[6] D. Edwards. *Introduction to Graphical Modelling*. Springer-Verlag, second edition, 2000.

[7] Ilya P. Ioshikhes and Michael Q. Zhang. Large-scale human promoter mapping using CpG islands. *Nature genetics*, 26(1):61–63, 2000.

[8] Michael I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19:140–155, 2004.

[9] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

[10] S. L. Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.

[11] K. Murphy. *Dynamic Bayesian Networks, Representation, Inference, and Learning*. PhD thesis, MIT, Department of Computer Science, 2002.

[12] Hanna M.Wallach. Conditional random fields: An introduction. Technical Report MS-CIS-04-21, University of Pennsylvania CIS, 2004.

[13] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceeding of HLT-NAACL*, 2003.

[14] Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*, 2002.