HIDDEN MARKOV MODELS FOR DNA SEQUENCING

Petros Boufounos[*], *Sameh El-Difrawy, Dan Ehrlich* [†]

petrosb@mit.edu
Massachussetts Institute of Technology
Research Lab For Electronics
77 Massachussetts Ave. Rm 36-643
Cambridge, MA 02139

{sameh,ehrlich}@wi.mit.edu
Whitehead Institute for Biomedical Research
9 Cambridge Center
Cambridge, MA 02142

ABSTRACT

In this paper we propose Hidden Markov Models as an approach to the DNA basecalling problem. We model the state emission densities using Artificial Neural Networks, and provide a modified Baum-Welch re-estimation procedure to perform training. Moreover, we develop a method that exploits consensus sequences to label training data, thus minimizing the need for hand-labeling. Our results demonstrate the potential of these models and suggest further research. We also perform a careful study of the basecalling errors and propose alternative HMM topologies that might further improve performance. We conclude by suggesting further research directions.

1. INTRODUCTION AND BACKGROUND

In recent years DNA sequencing has become a popular tool in Biology, significantly affecting the practice in the field. The impact of this method has created a need to automate the translation of sequencing signals (*electropherograms*) to the corresponding sequence of bases, a process known as *basecalling*. The most successful basecaller is PHRED [1, 2], currently used by the Human Genome project. More recently researchers have attempted to provide some statistical foundations to the problem and use statistical models to solve it (see e.g. [3, 4, 5]). In fact, in [4] the process is modeled as a Markov Chain and analyzed using Markov Chain Monte Carlo methods.

In this paper we use Hidden Markov Models (HMMs) to provide an alternative statistical foundation to the problem. Our approach has the advantage that it does not assume a particular peak shape, but with the cost of requiring some initial training. An excellent summary of HMMs, is contained in [6].

[*]This author performed the work while at Whitehead Institute

[†]The authors would like to acknowledge the support of the following grants: National Institutes of Health grant HG01389, Air Force Office of Scientific Research grant F49620-98-1-0235, and the National Institute of Justice award 98-LB-VX-K022.

2. MODEL TOPOLOGIES FOR DNA SEQUENCING

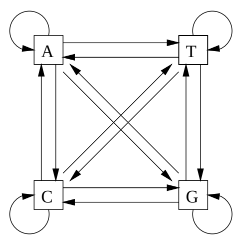
A key observation is that the DNA basecalling problem is similar to the speech recognition problem: Specifically, a time signal is to be translated to a sequence of symbols under a particular set of rules. In the case of speech recognition the grammar of the language forms these rules. On the other hand, in the case of DNA sequencing the “grammar” is very simple: the sequence is an i.i.d. process drawn from $\{A, T, C, G\}$. This similarity makes HMMs potentially applicable to the basecalling problem.

In order to use HMMs we need to formulate two models: the Markov Chain that models the underlying Markov process and the probability density model of the state emissions. Early experimentation with the data showed that Artificial Neural Networks (ANNs) can capture the state emission statistics more accurately than Gaussian Mixture Models. Thus we use a 3-layer ANN with a *softmax* output function. The input feature vector will be a 33-sample window of the four-channel electropherogram, centered at the current time point. We will normalize the 132 points feature vector to have a maximum of 1. Using ANNs for the state emission models requires some modifications to the Baum-Welch re-estimation algorithm, which we will describe in Appendix A.

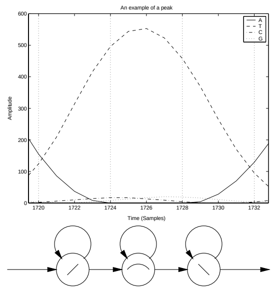
2.1. The basic models for recognition and training

We model the sequence using the Markov Chain of Fig. 1(a), which generates an i.i.d. sequence, as desired. We will represent each base in the sequence by the three-state model of Fig. 1(b), corresponding to the rise, the apex, and the fall of the corresponding peak in the electropherogram. The resulting model is shown in Fig. 1(c).

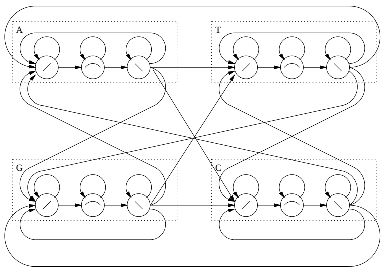
Similarly, we need a model to train the system. This should be generated by the sequences corresponding to the training data. For example the sequence AATCA should produce the model in Fig. 2. The same type of model will also be used in Section 3 to generate training data.



(a) Model of an i.i.d base sequence (all transitions have probability $\frac{1}{4}$)



(b) The model for each base and the part of the peak corresponding to each state.



(c) The result of combining (a) and (b)

Fig. 1. The model used for sequencing.

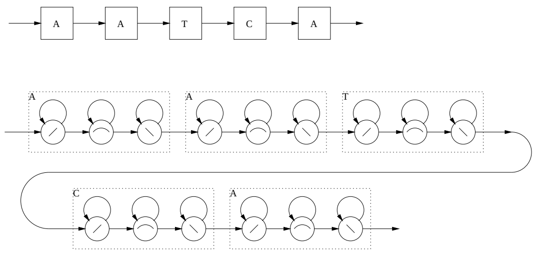
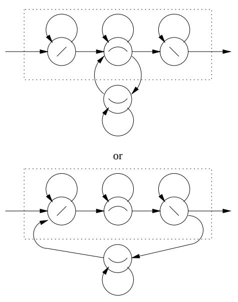


Fig. 2. The left-to-right model used to train the basecaller, generated from the sequence of the training samples.

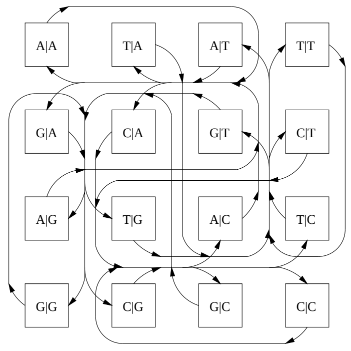
2.2. Alternative model configurations

The topologies we propose are not the only possible ones, and possibly not the best ones. Figure 3 illustrates some possible modifications—motivated by the analysis in Section 4—intended to accommodate effects that our design does not. For example, Fig 3(a) shows how an additional state captures the merging of the peaks when two bases of the same type (e.g. AA) appear next to each other.

Another alternative is to assume a second order underlying Markov chain and replace Fig. 1(a) with Fig. 3(b). This generates a 16-state Markov chain, where each state corresponds to the last two symbols of the sequence at any given time. This model captures any effects that require a second order Markov chain, such as the merging of the peaks described above and the compression effects described in [7]. Each of the boxes in the figure corresponds to a three state



(a) A replacement for 1(b) intended to reduce undercalls.



(b) A model that incorporates second order effects.

Fig. 3. Alternative models that can be used for basecalling.

sequence, as in Fig. 1(b), and the result is a 48-state Hidden Markov Model. Further discussion on alternative models can be found in [8].

3. GENERATING TRAINING DATA AND TRAINING THE MODEL

One issue in DNA sequencing problems is the lack of labeled training data, especially for new equipment. Hand labeling of electropherograms is very expensive and labor intensive. Basecaller-labeled data, on the other hand, are inaccurate for training purposes and create a chicken-and-egg problem.

Instead of labeled data, however, we can use electropherograms of fragments of published consensus sequences. Yet, these electropherograms do not all start from the same location in the sequence and do not have the same length. In order to generate labeled data, we will need to find which portion of the sequence corresponds to each electropherogram.

To locate electropherograms in the consensus sequences we use a variation of the Viterbi algorithm. We apply it on a left-to-right model like the one in Fig 2, generated by the consensus sequence. This model is large and requires significant computation, but we only need to label the data once. In any case, this method is cheaper and faster than hand-labeling.

The data generation involves partially training a HMM, with a set of parameters denoted by λ_p, using very few labeled electropherograms. To obtain these, some minimal human labor is necessary, either to locate them in the consensus sequence, or to hand-label them. Based on λ_p we will generate the left-to-right model with parameters λ_c which corresponds to the consensus sequence. The difference is that we will assume that the initial probabilities are the same for all the states: $\pi_i = \frac{1}{3N}$, where N is the number of bases in the consensus sequence. Then we will execute the Viterbi algorithm on the electropherograms that we need to locate in the consensus sequence. Indeed, this will result to the most likely path in the sequence, which we will use to label the data. The method we propose only requires a poorly trained model because it exploits the significant side information embedded in the consensus sequence. Details of the process are described in [8].

4. RESULTS

We implemented the proposed model in MATLAB and evaluated its performance on 10 electropherograms of PBluescript sequences (different than the training sequences), preprocessed with the standard software of the ABI 377 sequencing system with primer-dye chemistry. We compared our calls to the consensus sequence [9] using CROSS_MATCH [1] to determine the number of insertion, deletion and substitution errors. Figure 4 shows the comparison of our basecaller with PHRED version 0.990722g. We are plotting the average number of errors of each type versus the read length, as well as the average of the total number of errors. We should note here that PHRED is heavily optimized, and its preprocessor is tuned to work with its basecaller.

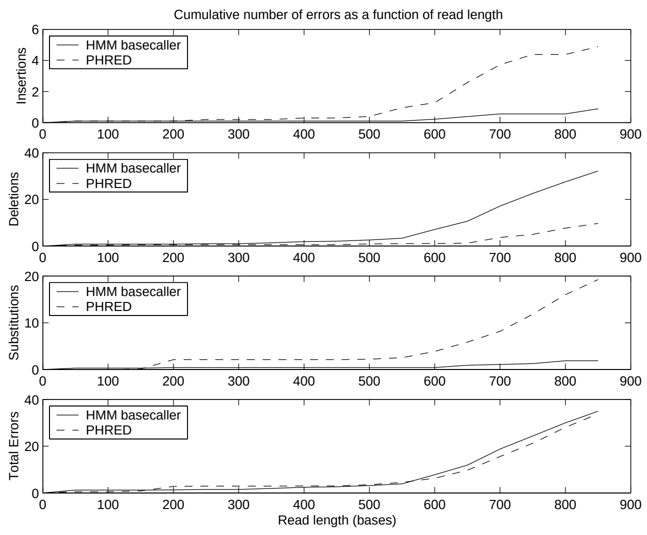


Fig. 4. Comparison of PHRED to the HMM basecaller. The total error performance is comparable.

From the results we see that although the HMM basecaller performs better than PHRED in terms of insertions and substitutions, it generates a significant number of deletion errors. However, the overall performance is comparable, encouraging further research and fine tuning.

Close inspection of the deletion errors showed that there is room for improvement. Specifically, the basecaller often merged bases of the same type and recognized them as only one peak. For example it would call a GAATC as GATC. This is due to the differences in the feature statistics of the AA transition compared to, for example, the GA transition. It is reasonable to believe that the models in Figure 3 will significantly improve performance.

5. CONCLUSIONS AND FUTURE WORK

We believe we have demonstrated the suitability and the potential of Hidden Markov Models as a basecalling tool. In fact, the results encourage further research in several areas:

5.1. Confidence Measures

Although HMMs provide a direct probabilistic interpretation of the results in terms of likelihoods, this interpretation is different than the PHRED confidence scores described in [2]. Further work is needed to provide intuition on comparing these two measures.

5.2. Model Topology

Our system produced a significant number of deletion errors using the simple model to perform recognition. Although we believe that the alternative models we propose will be a significant improvement in that area, we still need to test them. Furthermore, other models might exhibit even better performance and research is needed to determine them.

5.3. Features Selection

In this work we did not try to optimize our feature selection. With appropriate features selection the data might be better represented, improving performance or reducing complexity. For example we might be able to use Gaussian Models for emission densities and thus achieve faster convergence in training.

5.4. Preprocessing

Another step of the process we have not examined is the preprocessing of the electropherogram before it is fed to the basecaller. Especially the low-pass filtering used to remove the noise has the potential of destroying subtle but important features. Our basecaller does not depend on a noise-free signal to operate since a noise model can be incorporated in the state emission statistics. Thus it might be desirable to modify the preprocessing steps to improve performance.

5.5. Extensions

The potential applications of HMMs in other areas of biology, such as SNP detection and DNA fingerprinting are very promising. Furthermore, similar models can be used for protein sequencing, aiding the field of proteomics. We believe that the potential applications in other fields are numerous.

6. REFERENCES

[1] Ewing B. et al., “Base-calling of automated sequencer traces using *Phred*. I. Accuracy assessment.,” *Genome Res*, vol. 8, no. 3, pp. 175–85, Mar 1998.

[2] Ewing B., Green P., “Base-calling of automated sequencer traces using *Phred*. II. Error probabilities.,” *Genome Res*, vol. 8, no. 3, pp. 186–94, Mar 1998.

[3] Nelson D., “Improving DNA Sequencing Accuracy And Throughput.,” in *Genetic mapping and DNA sequencing.*, pp. 183–206. Springer, New York, 1996.

[4] Haan N.M. and Godsill S.J., “Modelling electropherogram data for DNA sequencing using variable dimension MCMC,” in *Acoustics, Speech, and Signal Processing, 2000. ICASSP ’00. Proceedings. 2000 IEEE International Conference on*, Instanbul, Turkey, June 2000, IEEE, vol. 6, pp. 3542 – 3545.

[5] Pereira M. et al., “Statistical Learning Formulation of the DNA Base-calling Problem and its Solution Using a Bayesian EM framework,” *Discrete Applied Mathematics*, vol. 104, no. 1-3, pp. 229–258, 2000.

[6] Rabiner L.R., “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[7] Bowling J.M. et al., “Neighboring nucleotide interactions during DNA sequencing gel electrophoresis.,” *Nucleic Acids Res*, vol. 19, no. 11, pp. 3089–97, Jun 1991.

[8] Boufounos P., “Signal Processing for DNA Sequencing,” M.Eng. Thesis, MIT, June 2002.

[9] Short J.M. et al., “Lambda ZAP: a bacteriophage lambda expression vector with in vivo excision properties.,” *Nucleic Acids Res.*, vol. 16, no. 15, pp. 7583–600, Aug 1998.

[10] Hennebert J. et al., “Estimation of Global Posteriors and Forward-Backward Training of Hybrid HMM/ANN Systems,” in *Eurospeech’97*, Rhodes, Greece, 1997, pp. 1951–1954.

A. COMBINING HMMS AND ANNS

In this appendix we present a modification of the Baum-Welch algorithm to train HMMs with emission probabilities modeled by ANNs, similar to [10]. In doing so we use the ANN to estimate the N-dimensional emission probability density vector $b_i[t] = P(q[t] = i|\mathbf{O}[t])$, where $\mathbf{O}[t]$ is the observation vector and $q[t]$ the state at time t.

Using b_i we compute $\alpha_i[t] = \frac{P(\mathbf{O}[1\ldots t], q[t]=i|\lambda)}{P(\mathbf{O}[1\ldots t])}$ and $\beta_i[t] = \frac{P(\mathbf{O}[(t+1)\ldots T]|q[t]=i,\lambda)}{P(\mathbf{O}[(t+1)\ldots T])}$ (here λ is the model) using a slightly modified forward-backward algorithm:

$$\alpha_j[t] = \begin{cases} \pi_j \frac{b_j[1]}{P_i} & t = 1 \\ \left[\sum_{i=1}^{N} \alpha_j[t-1]a_{ij}\right] \frac{b_j[t]}{P_j} & t > 1 \end{cases}$$

$$\beta_i[t] = \begin{cases} 1 & t = T \\ \sum_{j=1}^{N} a_{ij} \frac{b_j[t+1]}{P_j} \beta_j[t+1] & t < 1 \end{cases}$$

Thus, we can estimate $\gamma_i[t] = P(q[t] = i|\mathbf{O}[1\ldots T], \lambda)$ and $\xi_{ij}[t] = P(q[t] = i, q[t+1] = j|\mathbf{O}[1\ldots T], \lambda)$ using:

$$\xi_{ij}[t] = \frac{\alpha_i[t]a_{ij}\frac{b_j[t+t]}{P_j}\beta_j[t+1]}{\sum_{k=1}^{N}\sum_{l=1}^{N}\alpha_k[t]a_{kl}\frac{b_l[t+t]}{P_l}\beta_l[t+1]}$$

$$\gamma_i[t] = \frac{\alpha_i[t]\beta_i[t]}{\sum_{j=1}^{N}\alpha_j[t]\beta_j[t]} = \sum_{j=1}^{N}\xi_{ij}[t]$$

Given γ and ξ, we use the usual re-estimation formulas for all the model parameters except the emission densities:

$$\bar{\pi}_i = \gamma_i[1]$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_{ij}[t]}{\sum_{t=1}^{T-1}\gamma_j[t]}$$

$$P_i = \frac{\sum_{t=1}^{T}\gamma_i[t]}{T}$$

Finally, we use $(\mathbf{O}[t], \gamma_i[t])$ as input-output vector pairs to update the parameters of the ANN. This training schedule is summarized in Figure 5. [8] provides an extended discussion of the method.

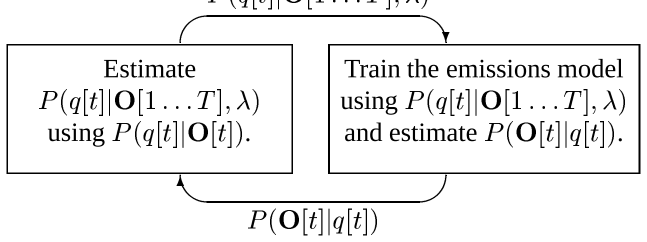

$P(q[t]|\mathbf{O}[1\ldots T], \lambda)$
Estimate $P(q[t]|\mathbf{O}[1\ldots T], \lambda)$ using $P(q[t]|\mathbf{O}[t])$.
Train the emissions model using $P(q[t]|\mathbf{O}[1\ldots T], \lambda)$ and estimate $P(\mathbf{O}[t]|q[t])$.
$P(\mathbf{O}[t]|q[t])$

Fig. 5. The modified Baum-Welch procedure.