

Alignment-Free Sequence Comparison with Vector Quantization and Hidden Markov Models

Tuan Pham

School of Computing and Information Technology

Griffith University

Nathan Campus, QLD 4111, Australia

t.pham@griffith.edu.au

Abstract

We introduce the concept of multiresolutions using vector quantization and hidden Markov models as a basis for alignment-free comparison of sequences. Different similarity measures can be discovered at different resolutions of the two sequences. The proposed approach provides a new aspect for studying the complexity of biological data and is effective for real-time processing.

1. Introduction

Alignment-free sequence domain for the comparison of biological data is still a very recent area of research in regard to alignment-based sequence methods [5]. This paper presents a new approach for alignment-free biological sequence comparisons, which at least overcomes some problems encountered by the frequency-based method in both computational speed and numerical analysis. It has been reported [6] that for long sequences, the frequency-based approach encounters a memory problem; the use of Mahalanobis distance causes singularity in the conversion of covariance matrices; and the computational process will take a considerably long time. In this study, a given long sequence of alphabets is converted into a sequence of numbers and modeled in the context of multi-resolutions by the use of vector quantization (VQ). Based on the VQ codebook, a hidden Markov model (HMM) is then built for each sequence, then the comparisons of similarity/dissimilarity between sequences can be made between the two HMMs.

2. Vector Quantization

Given a training set $\mathcal{T} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$, where each source vector \mathbf{x}_m is of k dimensions. Let N be a given number of codewords or codevectors and $\mathcal{B} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N\}$

represents the codebook of size N , where each codeword $\mathbf{c}_y = (c_{y1}, c_{y2}, \dots, c_{yk})$, $y = 1, 2, \dots, N$. Each codeword \mathbf{c}_y is assigned to an encoding region R_y in the partition $\Omega = \{R_1, R_2, \dots, R_N\}$. Then the source vector \mathbf{x}_m can be represented by the encoding region R_y and expressed by

$$V(\mathbf{x}_m) = \mathbf{c}_y, \text{ if } \mathbf{x}_m \in R_y$$

In general, the VQ design can be stated as follows. Given a training set \mathcal{T} , the size N of the codebook, we seek to find the codebook \mathcal{B} , and the partition Ω such that the average distortion D is minimized. One of the most well-known technique for VQ design is the LBG algorithm [3], which requires an initial codebook, and then iteratively bi-partitions the codevectors based on the optimality criteria of nearest-neighbor and centroid conditions until the number of codevectors is reached.

3. Hidden Markov Models

Let N be the number of hidden states, M the number of observation symbols, and $A = \{a_{ij}\}$ the state-transition probability distribution. The elements of a hidden Markov model $\lambda = (A, B, \pi)$ are defined as [4]

$$a_{ij} = P(q_{t+1} = j | q_t = i), 1 \leq i, j, \leq N$$

where q_t denotes the state at time t . $B = \{b_j(k)\}$: the observation symbol distribution, in which $b_j(k)$ is the symbol distribution in state j , $j = 1, \dots, N$. This can be expressed as

$$b_j(k) = P(o_t = v_k | q_t = j), 1 \leq k \leq M$$

where v_k denotes an individual symbol.

$\pi = \{\pi_i\}$: the initial state distribution, which is expressed as

$$\pi_i = P(q_1 = i), 1 \leq i \leq N$$

We can measure the similarity between two HMM models $\lambda_1 = (A_1, B_1, \pi_1)$, and $\lambda_2 = (A_2, B_2, \pi_2)$, using the concept of a distance measure [4] as follows:

$$S(\lambda_1, \lambda_2) = \frac{D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)}{2} \quad (1)$$

where

$$D(\lambda_1, \lambda_2) = \frac{1}{T} [\log P(O_{\lambda_2} | \lambda_1) - \log P(O_{\lambda_2} | \lambda_2)] \quad (2)$$

where $O_{\lambda_2} = (o_1 o_2 \dots o_T)$ is a sequence of observations generated by model λ_2 .

$$D(\lambda_2, \lambda_1) = \frac{1}{T} [\log P(O_{\lambda_1} | \lambda_1) - \log P(O_{\lambda_1} | \lambda_2)] \quad (3)$$

where $O_{\lambda_1} = (o_1 o_2 \dots o_T)$ is a sequence of observations generated by model λ_1 .

The forward or backward algorithm is used to compute $P(O|\lambda)$ [4]. The forward variable, denoted by $\alpha_t(i)$, is the probability of partial observation sequence $\langle o_1, o_2, \dots, o_t \rangle$ and state i at time t :

$$\alpha_t(i) = P(o_1 o_2 \dots, q_t = 1 | \lambda)$$

The term $\alpha_t(i)$ can be solved by induction as

$$\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}),$$

where $1 \leq t \leq T - 1$; $1 \leq j \leq N$

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

The backward procedure can also be constructed for calculating $P(O|\lambda)$. Given the model λ and state i at time t , the backward variable, denoted by $\beta_t(i)$, is defined as the probability of partial observation sequence from time $t + 1$ to the end of the sequence.

Reestimation of (A, B, π) is carried out to satisfy an optimization criteria using the Baum-Welch method which is also known as the expectation-maximization (EM) method [2].

4. Algorithm for Alignment-Free Comparison

1. Convert sequences of alphabets into numbers.
2. For each vector of size k :

3. Segment the sequences
4. Do VQ to obtain a codebook for each sequence
5. Compute and reestimate λ for each codebook
6. Compute $P(O|\lambda)$.
7. Compute $D(\lambda_1, \lambda_2)$.
8. Repeat steps 3-7 until the last size k .

5. Experimental Results

The algorithm is tested with the genome of *E.coli* K-12 MG1655 originally obtained from the University of Wisconsin [1]. The sequences are the threonine genes *thrA* (337-2799), *thrB* (2801-3733), and *thrC* (3734-5020), which are the second, third, and fourth open reading frames respectively. To build an HMM, the alphabets a, c, g, and t are the symbols, and a codebook size = 4, which correspond to 4 hidden states. Table 1 shows the absolute log-likelihood distance measures between (*thrA*, *thrB*), (*thrA*, *thrC*), and (*thrB*, *thrC*) at different resolutions with $k = 1, 2, 3$, and 4.

Table 1. Distance measures of *thrA*, *thrB*, and *thrC*

Vector size	1	2	3	4
(thrA,thrB)	8.75e-4	2.73e-5	0.0013	0.0011
(thrA,thrC)	5.75e-4	7.64e-6	0.0038	1.17e-4
(thrB,thrC)	0.0018	6.6374e-6	3.1180e-5	3.3036e-4

6. Conclusion

We have presented a multiresolution approach for computing stochastic distance measures between genomic sequences. The results obtained at different resolutions reveal detailed information for gaining insight into the complexity of biological data.

References

- [1] J.S. Almeida, J.A. Carrico, A. Maretzek, P. A. Noble, and M. Fletcher, Analysis of genomic sequences by chaos game representation, *Bioinformatics*, 17:5 (2001) 429-437.
- [2] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc.*, 39:1 (1977) 1-38.
- [3] Y. Linde, A. Buzo, and R.M. Gray, An algorithm for vector quantizer design, *IEEE Trans. Communications* 28:1 (1980), 84-95.
- [4] L. Rabina, and B.H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey, 1993.
- [5] S. Vinga, and J. Almeida, Alignment-free sequence comparison A review, *Bioinformatics*, 19:4 (2003) 513-523.
- [6] <http://www.bioinformatics.musc.edu/resources.html>