Computing Substitution Matrices for Genomic Comparative Analysis

Minh Duc Cao[1], Trevor I. Dix[1,2], and Lloyd Allison[1]

[1] Clayton School of Information Technology,
Monash University, Clayton 3800, Australia.
[2] Faculty of Information & Communication Technologies,
Swinburne University of Technology, Hawthorn 3122, Australia.

Emails: {minhduc,trevor,lloyd}@infotech.monash.edu.au

Abstract. Substitution matrices describe the rates of mutating one character in a biological sequence to another character, and are important for many knowledge discovery tasks such as phylogenetic analysis and sequence alignment. Computing substitution matrices for very long genomic sequences of divergent or even unrelated species requires sensitive algorithms that can take into account differences in composition of the sequences. We present a novel algorithm that addresses this by computing a nucleotide substitution matrix specifically for the two genomes being aligned. The method is founded on information theory and in the expectation maximisation framework. The algorithm iteratively uses compression to align the sequences and estimates the matrix from the alignment, and then applies the matrix to find a better alignment until convergence. Our method reconstructs, with high accuracy, the substitution matrix for synthesised data generated from a known matrix with introduced noise. The model is then successfully applied to real data for various malaria parasite genomes, which have differing phylogenetic distances and composition that lessens the effectiveness of standard statistical analysis techniques.

1 Introduction

Most important tools for mining in biological data such as sequence alignment and phylogenetics generally rely on a substitution matrix which ideally reflects the probability of mutating a character in a sequence to another in other sequences. Most sequence alignment algorithms attempt to find the optimal match of sequences where matching scores are derived from a substitution matrix [1, 2]. It is well known that using a reliable substitution matrix significantly improves the sensitivity of sequence alignment and database search tools [3, 4]. Substitution matrices also provide clues to dating of various evolutionary events and many molecular evolution mechanisms, and thus are often used in phylogenetic analysis [5, 6].

Classically, a substitution matrix is empirically selected based on some assumptions about the sequences being analysed. For protein analysis, the PAM substitution matrices [7] are calculated by observing the differences in related sequences with a certain ratio of substitution residues. The PAM-n matrix estimates what rate of substitution would be expected if $n\%$ of the amino acids had changed. On the other hand, the BLOSUMs [3] are derived from segments in a block with a sequence identity above a certain threshold.

While much research has been done on substitution matrices for protein, little attention has been paid to DNA substitution matrices despite the need of reliable tools for aligning genome size sequences of the next generation sequencing technology. Since not all DNA substitutions change the encoded amino acids, looking at the amino acid level only would lose some information. As more than one codons can code for the same amino acid, and different strains can show different preferences for codons that encode a given amino acid [8], a generic PAM or BLOSUM like substitution matrix for nucleotides such as RIBOSUM [9] can hardly work well on specific DNA sequences.

Work on DNA substitution matrices [10–12] often bases on a substitution model. Such examples of substitution models are the CJ69 model [13] which assumes all changes among four nucleotides occurring with equal probability and the K80 model [14] which allows transitions and transversions to occur with different rates. These models are rarely precise in practice. Traditional substitution matrix derivation methods also depend on sequence alignment, which in turn is plausible only when a reliable substitution matrix is used.

In this paper, we introduce a novel method to generate DNA substitution matrices for genomic comparative study. The method is based on information theory foundation [15] and is in expectation maximisation framework. Our method finds the substitution matrix directly from the data being analysed without having to make any assumptions. It considers the substitution matrix as parameters to align sequences and applies an expectation maximisation approach to estimate the parameters that optimise the alignment score. To the best of our knowledge, this method is the first to be able to compute specific substitution matrices for genome size sequences without any assumptions or a prior alignment of the data. The presented technique could be generalised to other types of data as well.

2 Methods

Information theory directly relates entropy to the transmission of a sequence under a statistical model of compression. Suppose a sequence X is to be efficiently transmitted over a reliable channel. The sender first compresses X using a compression model and transmits the encoded message to the receiver, who decodes the compressed stream using the same model to recover the original message. The *information content* $\mathcal{I}_X$ of X is the amount of information actually transmitted, i.e. the length of the encoded message.

Suppose a sequence Y related to X is available to both parties, the sender needs to transmit only the information of X that is not contained in Y. Since the receiver also knows Y, X can be recovered correctly. The amount of information actually transmitted in this case is called *conditional information content* of X given Y, denoted as $\mathcal{I}_{X|Y}$. The more related the two sequences are, the more information the two sequences share, the shorter message is transmitted. The *mutual information* of X and Y is defined as the difference between the information content and the conditional information content: $\mathcal{I}_{X;Y} = \mathcal{I}_X - \mathcal{I}_{X|Y}$.

Compression of sequences requires a compression model. To measure the conditional information content of one sequence given another, the compression model needs to use a substitution matrix as parameters. In light of the *Minimum Message Length* principle [16, 17], we propose an expectation-maximisation (EM) algorithm to find the substitution matrix that can produce the most compact encoding of the two sequences. In the E-step, one sequence is compressed on the background knowledge of the other to measure the conditional information content. The compression uses a substitution matrix, initialised to some default values, as the parameters to be estimated. We use the *expert model* [18] as the compression model because of its efficient performance and its ability to produce an local alignment of the two sequences [19]. In the M-step, the substitution matrix is then re-estimated based on the mutations observed from the local alignment. The EM process continues until the conditional information content obtained converges to an optimal value.

2.1 The Expert Model

The expert model algorithm [18] compresses a sequence X, symbol by symbol by forming the probability distribution of each symbol and then using a primary compression scheme to encode it. The probability distribution at a position is based on all symbols seen previously. Correspondingly, the decoder, having seen all previous decoded symbols, is able to compute the identical probability distribution and thus can recover the symbol. The information content of symbol x_i is computed as the negative log of the probability of the symbol [15]:

$$\mathcal{I}(i) = -logPr(x_i) \tag{1}$$

The algorithm maintains a set of *experts* to estimate the probability of a symbol. An expert is any entity that can provide a probability distribution of the symbol. An example is the *Markov expert* of order k which uses a Markov model learnt from the statistics of X to give the probability of a symbol given k preceding symbols. If a related sequence Y is available, the expert model employs *align experts* each of which considers the next symbol x_i in X to be part of a homologous region and align with a symbol y_j in Y. The align experts assume a substitution matrix P the entry $P(x, y)$ of which is the probability of substituting symbol y in Y by symbol x in X. The probability of symbol x_i predicted by an align expert is $Pr(x_i|y_j) = P(x_i, y_j)$. The expert model uses a hash table to propose align expert candidates. A hash table of hash size h suggests every matching h-mer as an align expert which is then evaluated and is discarded if it does not perform significantly better than the Markov expert.

The core part of the expert model is the combination of expert predictions. Suppose a panel of experts E is available to the encoder. Expert θ_k gives the prediction $Pr(x_{m+1}|\theta_k, x_{1..m})$ of symbol x_{m+1} based on its observations of preceding

m symbols. Expert predictions are combined based on Bayesian averaging:

$$\begin{aligned} Pr(x_{m+1}|x_{1..m}) &= \sum_{k \in E} Pr(x_{m+1}|\theta_k, x_{1..m}) w_{\theta_k,m} \\ &= \sum_{k \in E} Pr(x_{m+1}|\theta_k, x_{1..m}) Pr(\theta_k|x_{1..m}) \end{aligned} \tag{2}$$

The weight $w_{\theta_k,m}$ of expert θ_k for encoding x_{m+1} is assigned to $Pr(\theta_k|x_{1..m})$ and can be estimated by Bayes's theorem:

$$w_{\theta_k,m} = Pr(\theta_k|x_{1..m}) = \frac{\prod_{i=1}^{m} Pr(x_i|\theta_k, x_{1..i-1}) Pr(\theta_k)}{\prod_{i=1}^{m} Pr(x_i|x_{1..i-1})} \tag{3}$$

where $Pr(\theta_k)$ is the prior probability of expert θ_k before encoding any symbol. As Eq. 2 can be normalised to have $\sum Pr(x_{m+1}|x_{1..m}) = 1$, we can ignore the common denominator in Eq. 3 and take the negative log of the numerators:

$$-log_2(w_{\theta_k,m}) \propto -\sum_{i=1}^{m} log_2 Pr(x_i|\theta_k, x_{1..i-1}) - log_2 Pr(\theta_k) \tag{4}$$

Since $-log_2 Pr(x_i|\theta_k, x_{1..i-1})$ is the cost of encoding symbol x_i by expert θ_k, the right hand side of Eq. 4 represents the length of encoding subsequence $x_{1..m}$ by expert θ_k. As experts are evaluated on a recent history of w symbols, the message length of encoding $x_{m-w+1..m}$ is used to determine the weights of experts. Rewriting Eq. 4 for the weight of expert θ_k at position $m+1$ gives:

$$w_{\theta_k,m} \propto 2^{MsgLen(x_{m-w+1..m}|\theta_k) - log_2 Pr(\theta_k)} \tag{5}$$

Using only the Markov expert can produce the information content of sequence X. The conditional content of X given Y is obtained by combining the Markov expert with align experts. Align experts are first combined according to Eq. 5 to become the *blended align expert*, whose prediction is then combined with the Markov expert's prediction. The experts' weights specified in Eq. 5 involves the prior probability $Pr(\theta_k)$ of each expert. As all align experts are proposed by the same hash table, they have the same prior probability and hence the common factor $2^{-log_2 Pr(\theta_k)}$ can be ignored. However, for combination of the blended align expert and the Markov expert, their prior probabilities have to be specified. The prior probability of the blended align expert can be estimated from previous iterations of the EM process.

An align expert might be proposed simply by chance. The algorithm considers an align expert plausible if it performs significantly better than the Markov expert. It must encode the last w symbols better than the Markov expert by a threshold T bits, which is a parameter of the algorithm. When the align expert predicts beyond its homologous region, its performance worsens and it is discarded subsequently. Each align expert suggests an alignment of the region starting at the position it is proposed and ending at the position it is discarded. This region is called the *maximum-scoring segment pair* (MSP). The set of MSPs forms an local alignment of the two sequences.

2.2 Alignment Score and Mutual Information Content

Consider an align expert that uses a substitution matrix P and aligns x_i in X to y_j in Y. The alignment score is specified by the logarithm of the odds ratio of model H which assumes homology, and model R assuming random[20]:

$$S(x_i, y_j) = log_2 \frac{Pr(x_i, y_j|H)}{Pr(x_i, y_j|R)} = log_2 \frac{Pr(x_i, y_j|H)}{Pr(x_i)Pr(y_j)} \tag{6}$$

By Bayes's theorem, the numerator of the right hand side can be expressed as:

$$Pr(x_i, y_j|H) = Pr(x_i|y_j, H)Pr(y_j) = P(x_i, y_j)Pr(y_j) \tag{7}$$

Therefore,

$$S(x_i, y_j) = log_2 \frac{P(x_i, y_j)Pr(y_j)}{Pr(x_i)Pr(y_j)} = log_2 P(x_i, y_j) - log_2 Pr(x_i) \tag{8}$$

The alignment score of a MSP is the sum of alignment scores of all symbols in the region. If the MSP is from two regions starting at x_m and y_n respectively and is k symbols long, its alignment score is

$$S(x_m, y_n, k) = \sum_{i=0}^{k-1} -log_2 Pr(x_{m+i}) - \sum_{i=0}^{k-1} -log_2 Pr(x_{m+i}, y_{n+i}) \tag{9}$$

The two terms are the lengths of the compressed messages of the region $x_{m,k}$ by the Markov expert and by the align expert, respectively. In other words, the alignment score of a MSP is the mutual information content of the two regions.

2.3 Computing the Substitution Matrix

Once the local alignment of the two sequences is constructed, the substitution matrix is computed from the substitutions observed from the alignment. Entry $P(x, y)$ of the substitution matrix gets the value

$$P(x, y) = \frac{C_{x|y}}{C_y} \tag{10}$$

where $C_{x|y}$ is the number of symbol x in X that are aligned to symbol y in Y, and C_y is the number of symbol y in all MSPs.

A statistical hypothesis testing is performed to select the "good" MSPs to compute the substitution matrix. From Karlin-Altschul statistics [21], the E-value of occurrences of MSPs with a score S or greater is $E = KMN2^{-S}$ where M and N are the lengths of the two sequences and K is the Karlin-Altschul parameter. The occurrences of MSPs can be modelled by a Poisson process with characteristic parameter E. At the significance level $\alpha = 0.05$, the substitution matrix is estimated from mutations in MSPs having E-value $\leqslant \alpha$ or a score:

$$S \geqslant -log_2 \frac{\alpha}{KMN} \tag{11}$$

3 Experiment Results

We implemented the algorithm in Java and ran experiments on a PC with Intel Core 2 Duo 2.33Ghz CPU and 8GB of RAM, using Sun Java runtime environment 1.5. In our experiments, we used a hash table with hash key of 20 to propose align experts. The threshold T was set to 0.5 bits. The initial substitution matrix is set to have entries of 0.7 on the diagonal and 0.1 off the diagonal.

It is hard to verify substitution matrices derived from real data. We therefore performed experiments on a set of synthesised data so that the substitution matrix computed can be compared with the matrix used to generate data. The experiment is described in Subsection 3.1. We then ran experiments on a set of real data, as described in Subsection 3.2.

3.1 Experiment on Synthesised Data

Synthesised data was used to ensure the correct derivation of substitution matrices. The benefit of using artificial data is that the data can be generated with added noise from a known substitution matrix, and hence the computed matrix can be verified. We generated two "model genomes" each of which is one million bases long. About 10% of the first genome is "coding regions" which are copied to the second genome with substitution rates specified by a matrix P_{target}. The "non-coding regions" of the two genomes are independent on each other.

Table 1. The target and computed substitution matrices in the synthesised data experiment

$$P_{target} = \begin{vmatrix} .600 & .050 & .300 & .050 \\ .030 & .650 & .070 & .250 \\ .300 & .040 & .600 & .060 \\ .050 & .300 & .050 & .600 \end{vmatrix} \qquad P_{computed} = \begin{vmatrix} .596 & .051 & .300 & .052 \\ .029 & .652 & .009 & .250 \\ .299 & .041 & .599 & .061 \\ .052 & .299 & .050 & .598 \end{vmatrix}$$

The substitution matrix is reconstructed from the data by aligning the second genome against the first one. After the fifth iteration the changes to the matrix between two consecutive iterations were negligible. In other words, the matrix converges after 5 iterations and in less than 10 minutes. Table 1 presents the target matrix P_{target} and the computed matrix $P_{computed}$ whose rows and columns are in $ACTG$ order. Given the noise introduced during the generation of the two sequences, the similarity of the computed matrix and the target matrix shows the effectiveness of our algorithm.

3.2 Experiment on Plasmodium Genomes

We analysed the genomes of four *Plasmodium* species, namely *P. falciparum, P. knowlesi, P. vivax* and *P. yoelii* which cause malaria in various hosts. The genomes are obtained from PlasmoDB release 5.4 (`http://www.plasmodb.org/common/downloads/release-5.4/`). The nucleotide compositions in these species' genomes are very different. The AT content in the genome of *P. falciparum* is as high as 80% and in coding regions is 76.22% while the AT content

Table 2. Plasmodium genomes characteristics.

Species	Host	Genome Size (Mb)	%(AT) in Genome AT	%(AT) in CDS
P.falciparum	Human	23.2	80.63%	76.22%
P.vivax	Human	26.9	57.71%	53.70%
P.knowlesi	Monkey	23.4	60.79%	69.77%
P.yoelii	Rodent	20.1	77.36%	75.22%

Table 3. The substitution matrices of different malaria genomes

$$P_{Pf-Pk} = \begin{vmatrix} .701 & .074 & .144 & .081 \\ .107 & .707 & .054 & .131 \\ .184 & .066 & .642 & .108 \\ .089 & .156 & .075 & .680 \end{vmatrix} \quad P_{Pk-Pf} = \begin{vmatrix} .779 & .040 & .081 & .100 \\ .137 & .372 & .057 & .436 \\ .419 & .060 & .381 & .140 \\ .103 & .083 & .040 & .774 \end{vmatrix}$$

$$P_{Pf-Pv} = \begin{vmatrix} .613 & .086 & .227 & .074 \\ .085 & .705 & .077 & .133 \\ .146 & .084 & .687 & .083 \\ .073 & .233 & .086 & .608 \end{vmatrix} \quad P_{Pv-Pf} = \begin{vmatrix} .797 & .039 & .069 & .095 \\ .136 & .386 & .049 & .429 \\ .428 & .053 & .378 & .141 \\ .095 & .072 & .037 & .796 \end{vmatrix}$$

$$P_{Pf-Py} = \begin{vmatrix} .762 & .041 & .084 & .113 \\ .112 & .613 & .059 & .216 \\ .226 & .059 & .603 & .112 \\ .115 & .082 & .040 & .763 \end{vmatrix} \quad P_{Py-Pf} = \begin{vmatrix} .765 & .041 & .082 & .112 \\ .114 & .567 & .059 & .260 \\ .236 & .057 & .593 & .113 \\ .112 & .080 & .043 & .765 \end{vmatrix}$$

$$P_{Pk-Pv} = \begin{vmatrix} .741 & .063 & .145 & .051 \\ .060 & .754 & .076 & .110 \\ .101 & .072 & .757 & .060 \\ .052 & .142 & .065 & .741 \end{vmatrix} \quad P_{Pv-Pk} = \begin{vmatrix} .808 & .050 & .083 & .059 \\ .091 & .677 & .061 & .171 \\ .200 & .067 & .641 & .092 \\ .063 & .084 & .050 & .803 \end{vmatrix}$$

$$P_{Pk-Py} = \begin{vmatrix} .796 & .036 & .068 & .100 \\ .140 & .451 & .051 & .358 \\ .357 & .051 & .450 & .142 \\ .101 & .068 & .036 & .795 \end{vmatrix} \quad P_{Py-Pk} = \begin{vmatrix} .687 & .075 & .146 & .092 \\ .107 & .577 & .066 & .250 \\ .121 & .048 & .726 & .105 \\ .073 & .124 & .074 & .729 \end{vmatrix}$$

$$P_{Py-Pv} = \begin{vmatrix} .630 & .086 & .212 & .072 \\ .081 & .696 & .077 & .146 \\ .134 & .069 & .715 & .082 \\ .071 & .208 & .085 & .636 \end{vmatrix} \quad P_{Pv-Py} = \begin{vmatrix} .822 & .034 & .056 & .088 \\ .146 & .444 & .046 & .364 \\ .363 & .047 & .442 & .148 \\ .088 & .057 & .033 & .822 \end{vmatrix}$$

in the *P. vivax* genome and *P. vivax* coding regions is 57.71% and 53.70% respectively. The characteristics of these genomes are presented in table 2.

We applied our method to find the substitution matrix for each pair of these genomes. To compute the substitution matrix P_{Y-X} of genome Y to genome X, we compressed the genome X on the background knowledge of genome Y. Generally, about 4 or 5 iterations were required for convergence. The substitution matrices of these genomes are presented in Table 3.

4 Conclusions

We have presented a method for dynamically deriving optimal substitution matrices for analysis of long DNA sequences. The method is based on the sound theoretical foundation from information theory. We have shown that the method successfully regains the substitution matrix from synthesised data derived from a known matrix with introduced noise. The method has also been applied on real data with differing phylogenetic distances and nucleotide composition which would mislead classical statistical methods. Unlike traditional methods, our algorithm does not rely on the pre-alignment of sequences or on a substitution model. It incorporates the alignment of sequences and the substitution matrix computed in a expectation maximisation process. Furthermore, it can handle very long sequences in practical running time. The method therefore, would facilitate knowledge discovery in large and statistical biased databases.

References

1. Altschul, S.F., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl. Acids Res. **25**(17) (1997) 3389–3402
2. Kurtz, S., Phillippy, A., Delcher, A.L., Smoot, M., Shumway, M., Antonescu, C., Salzberg, S.: Versatile and open software for comparing large genomes. Genome Biol. **5**(2) (2004)
3. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. **89**(22) (1992) 10915–10919
4. Altschul, S.F., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic local alignment search tool. J. Mol. Biol. **215** (1990) 403–410
5. Lio, P., Goldman, N.: Models of Molecular Evolution and Phylogeny. Genome Res. **8**(12) (1998) 1233–1244
6. Felsenstein, J.: Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Biol. **76**(6) (1981) 368–376
7. Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C.: A model for evolutionary change in proteins. National Biochemical Research Foundation, Washington DC (1978).
8. Comeron, J.M., Aguade, M.: An evaluation of measures of synonymous codon usage bias. J. Mol. Biol. **47**(3) (1998) 268–274
9. Klein, R., Eddy, S.: Rsearch: Finding homologs of single structured RNA sequences. BMC Bioinformatics **4**(1) (2003)
10. Goldman, N.: Statistical tests of models of DNA substitution. J. Mol. Evol. **36**(2) (1993) 182–198
11. Yang, Z.: Estimating the pattern of nucleotide substitution. J. Mol. Evol. **39**(1) 1994) 105–111
12. Yap, V.B., Speed, T.P.: Modeling dna base substitution in large genomic regions from two organisms. J. Mol. Evol. **58**(1) (2004) 12–18
13. Jukes, T.H., Cantor, C.: Evolution of protein molecules. Mammalian Protein Metabolism (1969) 21–132
14. Kimura, M.: A simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. **16** (1980) 111–120
15. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27** (1948) 379–423
16. Wallace, C.S., Boulton, D.M.: An information measure for classification. Computer Journal **11**(2) (1968) 185–194
17. Wallace, C.S., Freeman, P.R.: Estimation and inference by compact coding. Journal of the Royal Statistical Society series **49**(3) (1987) 240–265
18. Cao, M.D., Dix, T.I., Allison, L., Mears, C.: A simple statistical algorithm for biological sequence compression. Data Compression Conference (2007) 43–52
19. Cao, M.D., Dix, T.I., Allison, L.: A genome alignment algorithm based on compression. Technical Report 2009/233, FIT, Monash University, (2009).
20. Altschul, S.F.: Amino acid substitution matrices from an information theoretic perspective. J. Mol. Biol. **219**(3) (1991) 555–565
21. Karlin, S., Altschul, S.F.: Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc. Nat. Acad. Sci., **87**(6) (1990) 2264–2268