Substitution Matrix Generation and Hidden Markov Model Statistical Parameter Estimation.

Jacob Friedberg
frie2142@vandals.uidaho.edu

Garrett Wells
well1157@vandals.uidaho.edu

April 28, 2023

Abstract

Substitution matrices are important tools in the bio-informatics toolkit. They allow researchers to determine how similar two sequences are to one another. Hidden Markov models also provide a similar usefulness by allowing researchers to determine how much parts of sequences contribute to characteristics of a gene. Using a list of 200 sequences we generate a substitution matrix using log-odds and estimate parameters for a Hidden Markov Model. We provide a substitution matrix, emission probability table, and state transition table from the data. The methods employed here produced good results which we are moderately confident in their accuracy. However, due to the small dateset size we are not confident that the substitution matrix or HMM parameters can be used generally on all types of sequences.

1 Introduction

In bio-informatics one of the crucial tools used to determine how similar two sequences are is a substitution or scoring matrix. These matrices give us a score when aligning sequences that can give us an idea of how related one sequence is to another. Two sequences are aligned with each other and character by character a score is accumulated across the sequence. There are several commonly used substitution matrices including BLOSUM-50, BLOSUM-62, and PAM. To generate these matrices, it is typical calculate the log-odds score for each cell of the matrix. This score is created by analyzing the frequency of at which a pair of characters appears in a large data set of sequences compared to the background frequency of each character and taking the log of that value. A scaling factor is used to scale the values to the range of $[-10, 10]$.

Of similar importance in bio-informatics is modeling the probability of observing gene sequences which contribute to characteristics of interest in a species. Such a model may then be used to analyze new gene samples to predict which characteristics they correspond to without employing the skills of a highly trained biologist. The model generated with this data is called a Hidden Markov Model(HMM) and uses the sequence of amino acids to predict the most probable state(genomic characteristic) that the sequence encodes for. The statistical information HMMs rely on is defined below:

Emission a symbol occurring(emitted) in a genome while in a specific state.

Transition a change from one state to another, representing a change in the function or type of the sequence.

This information is then used to produce a state machine which relies on emissions and transitions to guide the model's state prediction. The algorithms used to calculate these key pieces of data and tables are described below.

In this paper we implement two algorithms: Generating a BLOSUM-like substitution matrix using a provided set of 200 sequences and calculating the emission and transition probabilities given a set of sequences that are annotated to indicate the locations where specific amino acids within the gene contribute to different characteristics of the *Silacus Soulas* insect.

2 Algorithm Descriptions

2.1 Substitution Matrix

To generate a BLOSUM-like substitution matrix a log-odds score is used which is defined equation 1.

$$S_{(i,j)} = \frac{1}{\lambda} \log \frac{P_{ij}}{F_i F_j} \tag{1}$$

$$\frac{1}{\lambda} = \text{Scaling factor} \tag{2}$$

$$P_{ij} = \frac{\text{Number of i,j and j,i character pairs in all sequences}}{\text{Total pairs possible in all sequences}} \tag{3}$$

$$F_i = \frac{\text{Occurrence of character i in all sequences}}{\text{Total number of characters in all sequences}} \tag{4}$$

$$F_j = \frac{\text{Occurrence of character j in all sequences}}{\text{Total number of characters in all sequences}} \tag{5}$$

A score for a pair of amino acid characters i and j is defined as the a scaling factor $\frac{1}{\lambda}$ multiplied by the log of the probability of an i,j character pair over the background frequency of i and j. To calculate P_{ij}, all sequences are aligned with each other and a column by column analysis takes place. For each column the number of i,j and j,i pairs are accumulated and divided by the total number of pairs possible for all sequences. F_j and F_i are calculated by counting the occurrences of the characters divided by the total number of characters amongst all sequences. A scaling factor of $\lambda = 0.16$ or 6.25 times was used for our algorithm.

While devising the algorithm for implementing a log-odds score two key mathematical shortcuts were found. First, if we analyze the pattern of possible pairs created by moving down a 4 sequence list of length 1 as seen in Figure 1, we can see something interesting.

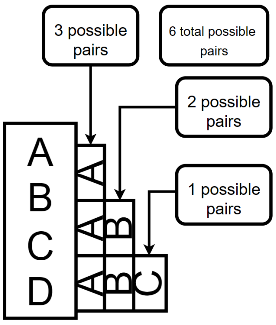

Figure 1

The number of total possible pairs in a given column is characterized by summation $\sum_{k=1}^{n-1} a_k$ where n is the number of sequences in the column. The closed form of this summation is:

$$\text{Total number of pairs in a column} = \sum_{k=1}^{n-1} a_k = \frac{n^2+n}{2} \tag{6}$$

The second shortcut was recognizing that since we are finding both i,j and j,i pairs, we can calculate the number of these pairs based purely on how many i and j characters are in a given column.

$$\text{i,j and j,i pairs in column} = \text{count of j in col} * \text{count of i in col} \tag{7}$$

However, there is a limitation in the use of equation 7 as it can only be used for i,j pairs that are not the same character which would occur along the diagonal of our substitution matrix. Instead, equation 6 can be used where n is the number of a's in a column for an P_{aa} calculation.

Using these two equations we can reduce the time complexity of our program immensely instead of having to go through all possible pairs of a column for the entire length of all the sequences which would incur a $K * \frac{n^2+n}{2}$ complexity where K is the length of our sequences and n is the number of sequences we have. We can reduce that down to just $K * n$ by using equations 6 and 7 to determine how many pairs should exist based on the counts of characters we are interested in rather than manually indexing all sequences in a column and finding the pairs manually through comparison.

2.2 Emission and Transition Probabilities

As introduced in the definition for emission, a symbol refers to one of the 20 amino acids, represented by its single letter substitute. Examples can be seen in emission table 2. In the emission parameters are calculated individually for each symbol based on state. The equation used for these calculations is listed below. The term, $e_{s_x symbol_y}$, represents the number of occurrences of $symbol_y$ in state s_x. This term is divided by all occurrences of any symbol in state s_x to produce a probability of seeing $symbol_y$ in s_x.

$$\frac{e_{s_x symbol_y}}{\sum_{k=1}^{n} e_{s_x symbol_k}} \tag{8}$$

Transition parameters are described as the probability of "transitioning from state X to state Y"(see table 3). They are calculated as the number of observed transitions between two states, s_x and s_y, divided by all observed transitions out of state s_x. Transitions between the symbols in the same state are counted to estimate the probability of remaining in the current state. The equation for this calculation is more formally stated below with t representing observed transitions between two states.

$$\frac{t_{s_x s_y}}{\sum_{k=0}^{n} t_{s_x s_k}} \tag{9}$$

As may be expected, all probabilities sum to 1, though in slightly different ways for each table. For the emission table, all probabilities for symbols seen in state 0(seen as one column of the table), for example will sum to 1. For the transition table, however, one horizontal row should be expected to sum to 1. Also note that though some transition probabilities are 0, this is considered acceptable since it is this represents that some segments of genome do not occur following others.

3 Results

	a	r	n	d	c	q	e	g	h	i	l	k	m	f	p	s	t	w	y	v
a	7	-1	1	3	2	-1	-1	-2	1	-2	-2	-1	-1	-2	-1	-2	3	1	-1	1
r	-1	4	2	1	1	1	2	1	2	1	1	2	3	2	2	2	1	1	1	1
n	1	2	4	1	1	2	2	1	2	2	1	1	2	1	3	1	2	1	1	2
d	3	1	1	3	2	1	2	2	2	2	1	2	1	2	1	2	2	1	1	2
c	2	1	1	2	4	0	0	2	1	1	2	2	2	1	2	2	0	0	3	1
q	-1	1	2	1	0	4	1	0	2	2	3	1	2	2	1	1	2	1	1	0
e	-1	2	2	2	0	1	4	2	3	3	0	1	2	1	0	2	1	2	1	0
g	-2	1	1	2	2	0	2	4	1	2	2	1	1	2	2	3	1	2	0	1
h	1	2	2	2	1	2	3	1	4	2	1	2	1	1	0	2	3	1	2	0
i	-2	1	2	2	1	2	3	2	2	3	1	2	3	3	2	2	1	1	1	0
l	-2	1	1	1	2	3	0	2	1	1	5	1	1	1	2	2	1	1	2	1
k	-1	2	1	2	2	1	1	1	2	2	1	3	1	0	2	1	3	2	2	1
m	-1	3	2	1	2	2	2	1	1	3	1	1	3	2	1	1	1	2	2	1
f	-2	2	1	2	1	2	1	2	1	3	1	0	2	3	2	1	1	1	1	1
p	-1	2	3	1	2	1	0	2	0	2	2	2	1	2	4	2	0	1	1	1
s	-2	2	1	2	2	1	2	3	2	2	2	1	1	1	2	3	2	1	1	2
t	3	1	2	2	0	2	1	1	3	1	1	3	1	1	0	2	3	1	2	1
w	1	1	1	1	0	1	2	2	1	1	1	2	2	1	1	1	1	3	0	2
y	-1	1	1	1	3	1	1	0	2	1	2	2	2	1	1	1	2	0	4	1
v	1	1	2	2	1	0	0	1	0	0	1	1	1	1	1	2	1	2	1	4

Table 1: Generated BLOSUM-like scoring matrix

Table 2: Emission Percentage By State

Amino Acid	**State**		
-	0	1	2
a	3.91	7.94	1.59
c	3.84	2.75	6.86
d	1.87	1.72	4.18
e	3.91	2.62	5.62
f	6.60	4.96	5.55
g	3.73	1.85	4.86
h	3.65	1.77	3.63
i	2.91	2.13	5.01
k	3.29	2.40	4.96
l	3.48	5.26	4.22
m	4.67	4.28	4.79
n	5.58	2.35	5.41
p	5.84	3.71	5.75
q	5.44	2.18	5.38
r	2.82	1.75	5.18
s	5.77	4.17	4.90
t	5.96	11.10	5.75
v	11.35	14.54	5.38
w	7.43	11.86	6.43
y	7.97	10.66	4.56

Table 3: Transitions Percentage Between States

From State	**To State**		
-	0	1	2
0	96.1	3.89	0.00
1	0.00	92.01	7.99
2	1.54	0.00	98.5