48. Carrillo, H. and Lipman, D., The multiple sequence alignment problem in biology, *SIAM J. Appl. Math.*, in press.
49. Altschul, S. F. and Lipman, D., Trees, stars, and multiple biological sequence alignment, *SIAM J. Appl. Math.*, in press.
50. Pearson, W. R. and Lipman, D. J., Improved tools for biological sequence comparisons, *Proc. Natl. Acad. Sci. U.S.A.*, 85, 2444, 1988.

Chapter 4

CONSENSUS PATTERNS IN SEQUENCES

Michael S. Waterman

TABLE OF CONTENTS

I. INTRODUCTION

One of the most difficult pattern recognition problems in sequence analysis is that of locating consensus patterns; its solution is of much importance to molecular biologists. These consensus patterns can occur among a set of sequences or within a single sequence. Sometimes the patterns all have exactly the same letters which occur in identical locations in all sequences of interest; there is little need for sophisticated programs designed to find such obvious features. For example, ACC occurs at the 3′ end of all tRNA molecules, in the same position relative to the acceptor stem. Features such as this one have been conserved over vast amounts of evolutionary time and are, without doubt, essential to the functioning of the organisms. It is more frequently the case, however, that a feature is conserved, but not conserved precisely in location or in pattern. There are different reasons for these various degrees of conservation; one classic example is discussed next.

The famous TATAAT box in bacterial promoters is located approximately 10 bases upstream (5′ or left) from the transcription start site. This 6-letter pattern occurs, however imperfectly, in the −10 region of all bacterial promoters. The variations in location and pattern might lead to skepticism regarding its existence or relevance to function. However, a number of experiments have essentially settled those issues. In Hawley and McClure,[1] the known −10 and −35 patterns were searched for in sequenced bacterial promoters, aligned on, and the consensus patterns refined. These two consensus sequences are known from experimental evidence to contain functional information that affects promoter activity.[1,2] Even after these detailed studies, it is not evident whether other features of DNA promoter sequences affect promoter activity. Other effects might not be as large as those attributed to the −10 and −35 consensus patterns, but still might be very important.

In Section II, we give a method for locating unknown patterns occurring imperfectly in both composition and location in a set of many sequences. The techniques work well on sequence sets such as bacterial promoters, and a subset of the known bacterial promoters is analyzed for illustration. Required for this analysis is a precise definition of consensus sequence specifying the amount of mismatch and/or gap allowed, as well as the amount of shifting permitted. These parameters and the quantity to be optimized are used to define consensus. The number of possible alignments in these sequence sets is enormous. Nonetheless, the techniques work rapidly on most problems of interest.

Next, in Section III, a related problem is considered. While homology in sequence pattern is an important biological feature, the sequence patterns can have other additional properties. Many (but not all) known protein binding sites have an approximate palindromic symmetry and are composed of inverted, complementary repeats. (A palindrome in nucleic acids is a sequence such as ACTGCAGT or TTAGCGGCTAA.) The knowledge that the pattern sought is a palindrome allows us to look more deeply into the sequences and to detect even weaker consensus patterns. A well-known pattern of this type is that found by Pelham[3] in promoters of heat-shock genes in *Drosophila*. The modification of the above method to the search for palindromes is necessary to detect signals or patterns of the strength found in the heat-shock sequences, and these sequences are used to illustrate the analysis.

Repeats in a single sequence have also been of much interest. Exact repeats are the basis of the algorithms of Karlin et al.[4] and Martinez and Sobel,[5,6] and the computer methods to rapidly find exact repeats are usually based on hashing. Hashing techniques do not apply to inexact repeats. Our interest here is in inexact repeats, and a modification of Section II will let us study inexact repeats in Section IV. Of necessity, the search is restricted to smaller patterns, up to 9 to 12 bases in length. While hashing methods are routinely used in data base searches,[7] our methods do not apply to those problems.

Just as palindromes are of interest in the study of regulatory patterns common to a set of sequences, consensus palindromes along a single sequence reveal possible binding sites for

a regulatory protein. The patterns are usually weak and quite hard to find by inspection. An algorithm adapting those of Sections II and III to a single sequence is given in Section IV along with application to the GAL1 promoter sequence.

Many weak consensus patterns in molecular biology involve long patterns. For example, the Alu family of repeats involves 300 base patterns, which occur thousands of times in the human genome. The methods used above involve storage of all possible patterns of interest. This approach is impossible for longer patterns. For example, there are $4^{300} \approx 4.1 \times 10^{180}$ patterns of length 300. Other methods, perhaps less optimal, must be developed. A recent approach to these problems is described in Section V.

Whenever a consensus pattern is located, whether by inspection or by computer, the question of statistical significance often arises. Biological significance cannot be equated with statistical significance. However, patterns that are extremely likely to occur in random sequences of similar composition seem unlikely to be biologically significant. We prefer simple models of randomness for doing these calculations. These simple models do not model the real sequences themselves particularly well,[8] but the distribution of matching between unrelated real sequences is modeled very well by that between independent sequences of the same composition.[9] For our purposes here, studying weak matchings between many sequences, the theory of large deviations is very useful. In Section VI, this theory, as well as the log(n) distribution and simulation, is discussed.

II. CONSENSUS WORDS IN MULTIPLE SEQUENCES

As described in the introduction, in this section we study the problem of determining consensus words that occur in a set of sequences, where the occurrences of the words are inexact and differ in location from sequence to sequence. We first give a combinatorial treatment of the number of alignments possible with some constraints on amount of shifting. Then we describe the algorithm we employ to identify consensus patterns. Lastly, to illustrate the analysis, we study a set of bacterial promoters with these methods.

A. Combinatorics

In a direct approach to the problem, the R sequences under consideration can be analyzed for consensus words by placing them into various alignments. For each alignment, the various columns are examined for "consensus" letters. Groups of consensus letters can then be identified as consensus patterns. The goal here is to decide the difficulty in such a straightforward approach to identifying consensus patterns.

Let the initial alignment of R sequences of the same length N be given as:

$$\begin{array}{llll} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \\ a_{R1} & a_{R2} & \cdots & a_{RN} \end{array}$$

The simplest scheme for alternate alignment is to allow shifts of the sequences relative to one another, without any gaps inserted into the sequences. Usually the amount of shifting is limited to a fixed number of bases:

$$\begin{array}{lccccccc} \text{original position:} & & a_{11} & \cdots & a_{1,k-1} & \cdots & a_{1,N} \\ \text{shifted by k bases:} & a_{1,1} & \cdots & a_{1,k+1} & \cdots & a_{1,N} & & \end{array}$$

If each sequence can be shifted up to k bases, then there are $k + 1$ choices for the positioning of each sequence. Consequently, there are $(k + 1)^R$ of these alignments, since each sequence

can be put into k new positions. In our example problem below, there are R = 59 sequences. With only k = 1, there are $(k + 1)^{59} = 2^{59} \approx 5.77 \times 10^{17}$, and no computer could possibly analyze all these alignments directly. The first analysis we perform in subsection C will have k = 3 so that $(3 + 1)^{59} \approx 3.32 \times 10^{35}$.

Additionally, we might want to insert gaps into the sequences. The number of alignments when gaps are allowed greatly increases the numbers obtained above (see Chapter 3[19]). These simple observations about the number of alignments show that a direct approach to consensus pattern identification is impossible. The ideas here apply equally well to the search for consensus palindromes (Section III) or to any situation where the correct sequence alignment is not precisely known.

B. Algorithm

The brute force approach of the last section has another problem in addition to that of combinatorics. There, an analysis could be based on percentages of bases in columns of an alignment. Here, we give an analysis based on the occurrence of k-letter words, and our consensus pattern is some k-letter word. In this way, it is possible to directly study the objects of interest. The algorithm was first presented by Waterman et al.[11]

Fundamental to our analysis is the concept of neighborhood of a word. Suppose k = 6 and w = TATAAT. In the set of 4^k k-letter words, there is one word equal to w, $\binom{6}{1}3 = 18$ words within 1 mismatch of w, and $\binom{6}{2}3^2 = 15 \cdot 9 = 135$ words within 2 mismatches of w.

No. of length 6 words	Mismatches from w = TATAAT
TATAAT	0
AATAAT	1
CATAAT	1
GATAAT	1
TCTAAT	1
TGTAAT	1
TTTAAT	1
TAAAAT	1
TACAAT	1
TAGAAT	1
TATCAT	1
TATGAT	1
TATTAT	1
TATACT	1
TATAGT	1
TATATT	1
TATAAA	1
TATAAC	1
TATAAT	1

The neighborhood of a word is limited by the algorithm to words within a certain number of mismatches, deletions, and insertions of the consensus word. For example, the first analysis of subsection C allows a consensus word to be "found" in a neighborhood of up to 2 mismatches away from the consensus and does not allow any insertions or deletions.

Another parameter that must be specified is W, the window width. W is the number of

sequence letters that can be searched for a consensus word of k-letters. This allows shifts of up to k-letters. Below, for example, we search the promoter sequences for words of six letters in a window of 9 bases. If the window is set too wide, statistically insignificant patterns will be found. On the other hand, if a window is too narrow, the true consensus pattern can be missed. There is a balance of pattern length and neighborhood size with window width. As the likelihood of finding an acceptable pattern in random data increases, the window width should be decreased.

We now begin an explicit definition of the algorithm. The window specifies a search of the sequences $a_1, a_2, \ldots a_R$ from column j + 1 to column j + W. It is assumed that the sequences have been placed in an initial alignment, simply aligning on right (3') ends, on left (5') ends, or on some known biological feature in the data. The window reveals the sequences

Column	j + 1	j + 2	...	j + W
Sequence 1	$a_{1,j+1}$	$a_{1,j+2}$	...	$a_{1,j+w}$
Sequence 2	$a_{2,j+1}$	$a_{2,j+2}$	...	$a_{2,j+w}$
	.....		...	.....
Sequence R	$a_{R,j+1}$	$a_{R,j+2}$	...	$a_{R,j+w}$

We index the neighborhood of a word w by d = 0,1,2, . . . , where d = 0 indicates the word w, and d = 1 might, for example, indicate the 1 mismatch neighborhood of w. For each sequence $a_{i,j+1} \ldots a_{i,j+w}$, let q(i,w,d) = 1 if the best occurrence of w in the ith sequence is as a d^{th} neighbor, and q(i,w,d) = 0, otherwise. We order the neighbors by the penalties given below; let is suffice here that exact (d = 0) is best, d = 1 is next best, etc.

There are several approaches to computing

$$Q(i) = (q(i,w,0), q(i,w,1), \ldots)$$

In this representation of Q(i), there are 4^k lines, each corresponding to some w. In our program, each k-letter word in the sequence itself is used to produce all neighbors, and these neighbors are used to construct q(i,w,·) for all 4^k words w. This involves storing all 4^k words and finding their best occurrence in $a_{i,j+1} \ldots a_{i,j+w}$. We do not directly search the sequence for all the 4^k words but instead use the k-letter sequences in $a_{i,j+1} \ldots a_{i,j+w}$ to find the neighbors.

Next set

$$V = \sum_{i=1}^{R} Q(i)$$

$V = (v_{w,d})$ is useful since $v_{w,d}$ is an integer equal to the number of times or lines that w has its best occurrence as a d^{th} neighbor. We are now ready to define the score associated with word w:

$$s_w = \sum_{d \geq 0} \lambda_d v_{w,d}$$

where λ_d is the weight given to having a best occurrence of w as a d^{th} neighbor. The weight λ_d gives the preferences among the neighbors of w. In our program, we use the ratio of matching letters to w and the length of word k. That is

$$\lambda_{\text{exact}} = k/k = 1$$

and

$$\lambda_{1\ \text{mismatch}} = (k - 1)/k = 1 - 1/k$$

For words of length 6, six occurrences of 1 mismatch neighbors of w is equal in weight to five exact occurrences of w. There is no virtue in this scheme except that of simplicity, and λ_d can be easily changed.

Finally, we define a winning word w to satisfy

$$S(w) = \max_w\{s_w\}$$

Generally, with fixed W and n(d) = number of words in the neighborhood of w, the computation time (for R sequences of length N) is proportional to

$$R(N - W + 1)(W - k + 1)n(d)$$

It is pleasant, for W and n(d) fixed and N much larger than W, that the running time is approximately proportional to RN, so that twice as many sequences take twice the running time, and the same holds true for sequences of twice the original length. The maximum of n(d) is 4^k, but usually we have n(d) much less. For k = 6, $4^6 = 4096$ while allowing up to d = 2 mismatches gives n(2) = 1 + 6 + 135 = 154 and d = 3 mismatches gives n(3) = 1 + 6 + 135 + 540 = 682.

Many useful extensions can be made to these ideas. Certainly it is easy to relax the requirement that the consensus letters be contiguous. The positions of the consensus letters relative to one another must be fixed before the search is made. Otherwise, we would be faced with the task of searching over all $\binom{W}{k}$ ways of taking k consensus letters within the window. An algorithm could be devised, but we are not confident that it would be very useful.

When a signal or consensus word is located, the sequences could be aligned on the consensus word and then the sequences studied with this new alignment. We have implemented this feature and have found it quite useful. If two distinct patterns are located (approximately) a fixed distance apart, then aligning on the stronger pattern could allow the weaker pattern to become evident.

Another natural modification that we allow is search in all alphabets. The alternate alphabet most often invoked is the purine/pyrimidine alphabet, {R,Y}, where R = A or G, and Y = C or T. There is some hope of detecting DNA structural patterns by these sub-alphabets.[12] In addition, various k-letter structural motifs might be identified with their k-letter patterns and, with a proper concept of neighborhood, consensus structural patterns could be studied.

Finally, each sequence occurrence could be weighted by another constant, K_i. The definition of V would change to

$$V = \sum_{i=1}^{n} K_i Q(i)$$

Our motivation here is that a measurement of promoter strength, for example, might give a weighting of how much importance we should associate with patterns from each sequence. Unfortunately, we have not yet found an appropriate example to which we can apply this algorithm.

It has occurred to analysts that the signal might really be missing patterns rather than abundant patterns.[12] For these searches, the word of interest has score $\min_w\{s_w\}$, and there is no difficulty in including these searches.

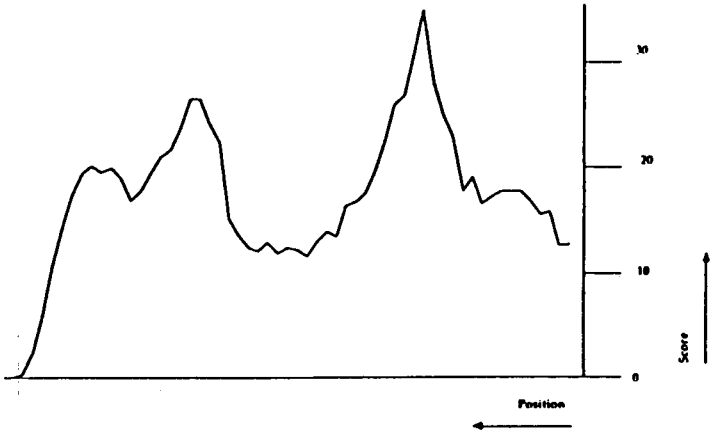


FIGURE 1. Graph of scores of the 59 bacterial promoter sequences that appear, along with their names, in Figure 2. The window width W = 9, the word size k = 6, and the neighborhood is up to mm = 2 mismatches.

C. Bacterial Promoters

The example analyzed in this section consists of 59 *Escherichia coli* promoter sequences originally analyzed by Hawley and McClure.[1] One or two bases were added to the sequences where the sequences are known, with the data taken from the references given by Hawley and McClure.[1] The analysis of this section follows Galas et al.[13] The graph of scores appears in Figure 1, and sequences are presented with their names and the three major consensus patterns in Figure 2; this is discussed in detail later. The sequences are aligned on the trascription start site, which is position 10 in Figure 2. The usual biological indexing would have this position indexed + 1, the numbering increasing to the right; any position to the left of the start of transcription is negative. There is no 0 in the biological scheme of indexing positions. For ease of sequence handling, we number the sequences from + 1, right to left, as indicated in Figure 2.

The first analysis has the window width set at W = 9, with the word size k = 6, and allows a neighborhood of up to 2 mismatches. To index all possible window positions, we take the right-hand edge of the window. The resulting graph appears in Figure 1. The horizontal axis represents window position in the sequences. When a feature of interest is located on a graph, the program allows us to move the window to the position pointed to on the graph. For example, the sharp peak at the right-hand edge of the graph corresponds to the sequence pattern indicated in the right-hand column of Figure 2. The black region indicates the window; notice that it is of width 9. The lower case letters show patterns found to produce the score at the peak. In the right-hand column, the consensus pattern is, approximately (within the neighborhood), the well-known −10 pattern TATAAT. It occurs exactly 9 times; in 17 sequences, its best occurrence is with 1 mismatch (mm), while in 18 sequences, its best occurrence is with 2 mm. The resulting score is s = 35.17.

The middle column represents the −35 consensus pattern. In this analysis, the consensus word is w = TTGACA with 4 exact occurrences, 15 with 1 mm and 13 with 2 mm. The $s_{TTGACA} = 26.83$. Aligning on the −10 pattern does not enhance the −35 pattern, nor does aligning on the −35 pattern enhance the −10 pattern.[13] We conclude that, while the sequence patterns are about 17 bases apart, the pattern spacing is not too closely linked.

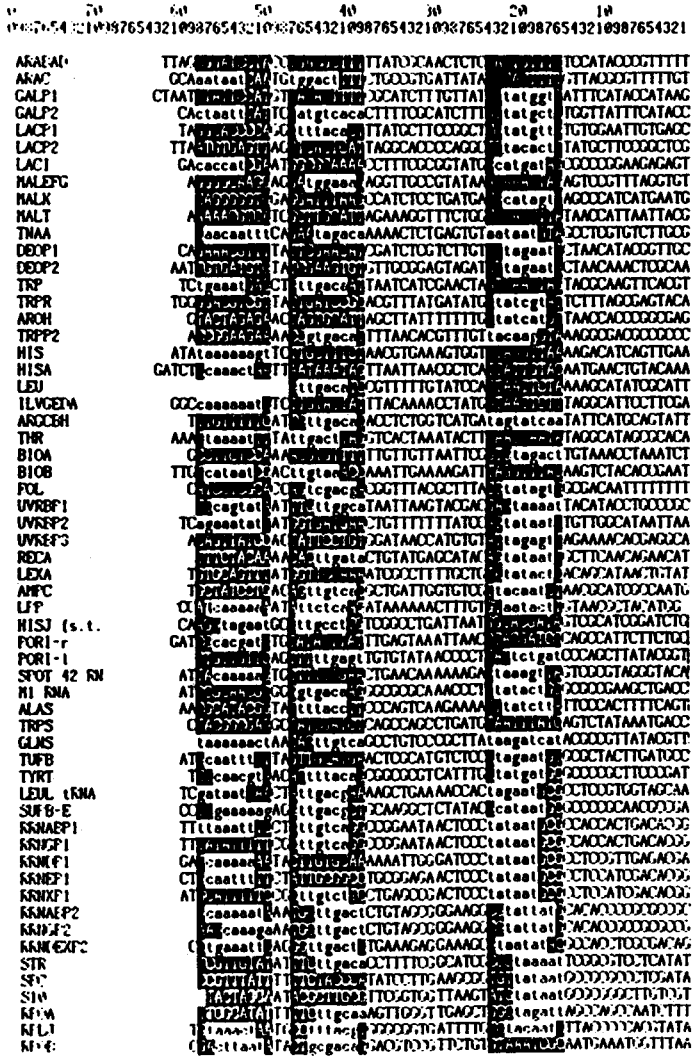


FIGURE 2. The 59 bacterial promoter sequences analyzed. The three black columns indicate the window locations of the −44, −35, and −10 patterns, with the patterns shown in lower case.

Table 1
C-POLYA-T PATTERNS IN −44 REGION

A

lpp	CAAAAAAAT
malT	TAAAAAAAC
his	TAAAAAAG
glnS	TAAAAAAC
spoT 42 RNA	CAAAAG
rrnAB P2	CAAAAT
supB − E	GAAAAAG
thr	TAAAT
recA	CAAAC
bioA	CAAAAC
trp P2	GAAAC
hisA	CAAAC
araC	CAAAT
rplJ	TAAAC
tnaA	TAAAC
rrnAB P1	TAAAT
rrnD P1	GAAAT
deo P1	GAAAC
trp	GAAAT
uvrB P2	GAAAT
rrnG P2	GAAAT
rrnDEX P2	GAAAT
hisJ	TAGAAT
uvrB P1	TATAAT
bio B	CATAAT

B

spc	GAAAAAAT
tufB	TAAAAAAT
rpoA	GAAAAAT
rrbG P1	GAAAAAT
rrnE P1	GAAAAAT
rrnX P1	GAAAAAT
thr	TAAAAT
deo P1*	TAAAAC
rrnAB P1*	TAAAAT
lexA	TAAAC
gal 8P2	TAAAT

Note: In column A the sequence patterns are shown, while those sequence patterns with pattern in reverse orientation are shown in column B.

Next, we examine the new pattern found by Galas et al.,[11] approximately at position −44: CAAAAT. In Hawley and McClure, an "A" was noted in this region. The pattern C-polyA-T appears, in forward or reverse orientation, in approximately half this data set (see Table 1 for these explicit sequences). Wu and Crothers[14] and Crothers et al.[15] have presented evidence of an unusual conformation associated with these sequences, a bending of the DNA. The sequence may have functional importance in these promoters. The CRP protein (CAP) alters the conformation in the *lac* control region when it binds to a site just 5′ of the −35 region.[16]

This suggests that the role of CAP could be played by sequence conformation in some bacterial promoters.

To see how to set the window and mismatch parameters, we present some graphs in Figure 3. With fixed W and k, let mm increase from 0 to 3. At mm = 0, the signal is small; perhaps the −10 signal might be suspected, but even that is doubtful. As mm increases to 1 and then to 2, the sequence features become more clear. When mm = 3 the noise of the sequences is beginning to affect the signal. It is even more interesting to vary the window width. At W = k = 6, no shifting is allowed. Then the signals become a little more evident at W = 7 and even clearer at W = 9. Finally, with W = 15 the signals have again become swamped by sequence noise. Much can be learned from sitting at a graphics terminal and varying these parameters.

To illustrate sequence patterns quite invisible to the eye, we take the three alphabet C,T,R = {A,G}. Figure 4 shows the graphs for two runs. Some new patterns appear besides those at −10, −35, and −44. Figure 4A finds at approximately −23, a pattern TRR which occurs 23 times with W = 6, k = 3, and mm = 0. This is above that expected from random sequences of these compositions. The literature assigns a CAT pattern at +1, the transcription start site. Here we find CRT occurring 24 times at the transcription start site. In Figure 4B, we have W = 5, k = 4, and mm = 1. The pattern TRRR at −23 occurs 5 times exactly and 27 times with mm = 1. This is less statistically significant than the k = 3 version of the pattern.

When the program is run on the data set searching for absent patterns, $s^* = \min_w\{s_w\}$, the only features of interest correspond exactly to −10 and −35. We interpret this as being due to the occurrence of the −10 and −35 consensus words. Nothing new is learned from this analysis.

III. CONSENSUS PALINDROMES IN MULTIPLE SEQUENCES

It is feasible that additional information is available regarding the possible consensus patterns. This knowledge can be used to restrict the set of possible consensus patterns from the 4^k possibilities for k-letter words. Having a smaller set is certainly useful in terms of storage and can help reduce running time if the neighborhood structure is convenient. In addition, detection of a pattern in this reduced set of patterns can be more sensitive.

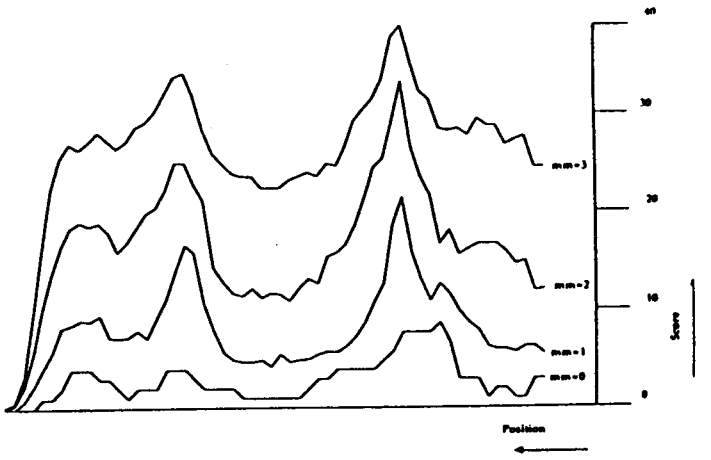
This section treats one such example, that of palindromes or patterns with reverse complement. Protein binding sites are frequently approximate palindromes, the motivation for these considerations. The example studied in Section III.B is the *Drosophila* heat-shock promoters. In these sequences, the determinant of heat-shock response does not seem to be detectable with the consensus word method of Section II, but some signal does appear when the consensus palindrome method is used.

A. Algorithm

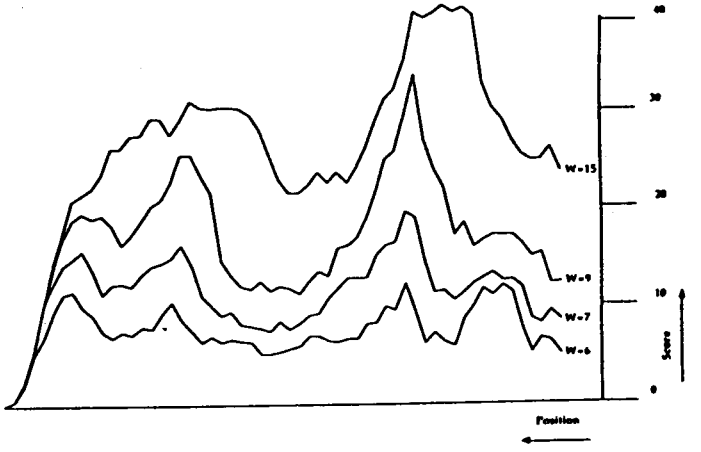
As discussed above, we restrict the set of consensus patterns to palindromes. Specifically, palindromes of length 2k + 1 have the form

$$A_1A_2 \ldots A_kNB_k \ldots B_2B_1$$

where $\bar{B}_i = A_i$ [$\bar{A} = T, \bar{G} = C, \bar{T} = A$, and $\bar{C} = G$] and N, of course, denotes an arbitrary base. Palindromes of length 2k do not have the N in the middle. Thus, there are 4^k palindromes of length 2k or 2k + 1. The idea here is similar to that for consensus words. Each sequence or word can contribute to the score of a palindrome if it is in the palindrome's neighborhood. Therefore, it is of interest to look at an example, w = TAAGGCTA. Notice that w is not a palindrome and, in fact, no palindrome is 1 mismatch (mm) from w. If the neighborhood allows up to 3 mismatches, then the following neighbors result.



A



B

FIGURE 3. A study of the effects of varying algorithm parameters. (A) The mismatch parameter is set at mm = 0,1,2,3 with W = 9 and k = 6. (B) The window width is set at W = 6,7,9,15 with k = 6 and mm = 2.

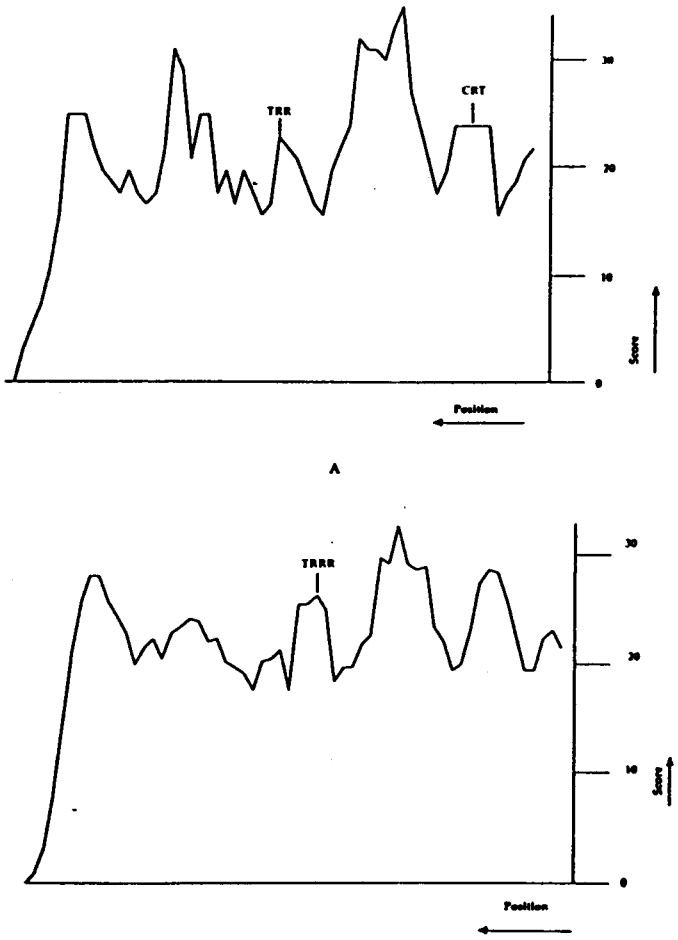


FIGURE 4. Analysis with the alaphabet C,T,R = {A,G}. (A) W = 6, k = 3, and mm = 0. (B) W = 5, k = 4, and mm = 1. A new pattern TRR or TRRR appears at approximately −23.

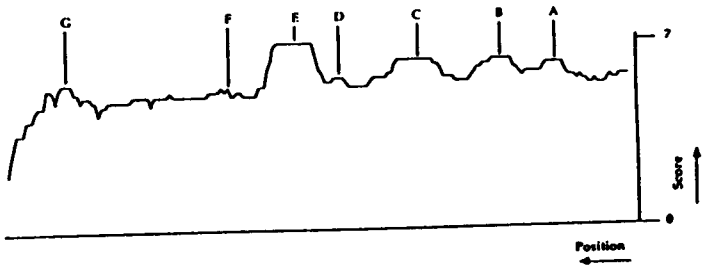


FIGURE 5. Graph for consensus word scores in seven *Drosophila* heat-shock gene promoters. Here w = 20, k = 6, and mm = 2. Scores of some patterns of interest are labeled.

Length 6 palindromes	mm from w = TAAGGCTA
TAAGCTTA	2
TAACGTTA	2
TAGGCCTA	2
TAGCGCTA	2
TAATATTA	3
TAAATTTA	3
TAGTACTA	3
TAGATCTA	3
TATCGATA	3
TACCGGTA	3
TATGCATA	3
TACGCGTA	3
TATGCATA	3

The remainder of setting up the algorithm goes as with the previous algorithm in Section II.B. Let the window and neighborhood be defined. Then let $V = \{v_{p,d}\}$ be defined by setting $v_{p,d}$ equal to the number of lines a palindrome p has its best occurrence as a d^{th} neighbor. Then the score for p is

$$s_p = \sum_{d \geq 0} \lambda_d v_{p,d}$$

where as before λ_d is the weight given to a d^{th} neighbor occurrence of the palindrome p. The winning palindrome is that with $\max_p\{s_p\}$. The algorithms to accomplish these tasks are similar to those of Section II.B.

B. Heat-Shock Promoters

In higher organisms, there is a collection of genes that are principally expressed at higher temperatures. These genes seem to be stimulated by a number of physiological stresses, one of which is heat. It is natural to study these genes to determine the features that differentiate their control from that of other genes.

In *Drosophila*, these genes are known as heat-shock genes. A number of *Drosophila* heat-shock genes and their 5' flanking DNA have been sequenced. While there are no patterns which obviously differentiate these sequences, several studies have addressed this issue.[3,11] The consensus pattern suggested by those authors is a weak palindrome. In this section, we study these sequences using the tools described above.

The first approach is by the analysis for consensus word. The sequences are aligned on the start of transcription, approximately position 90 of Figure 6. In Figure 5, we give the

graph of scores vs. positions for window width W = 20, word size k = 6, and a neighborhood of up to 2 mm. Clearly, there are some patterns of interest and those appear, along with the sequences, in Figure 6. Patterns (A) and (B) have not been noticed before, while (C), the pattern TTCAAA, 3' of the transcription start site has been described earlier. Pattern (E) is the famous TATAAA box and is almost perfectly aligned with the start of transcription. Pattern (F) would not be noticed, but it appears approximately where the consensus palindrome was noted by Pelham. Finally, we show pattern (G) which is some distance 5' from signals previously noticed.

The Pelham pattern is not located by the consensus word search. For reasons of computer storage, we are unable to use longer words, such as length 14, in that program. The consensus palindrome algorithm allows longer words since there are, for example, 4' palindromes of length 14. The window is set at W = 25 and up to 6 mismatches are allowed. Some patterns found by this search for consensus palindromes are shown in Figure 8, with the graph appearing in Figure 7.

The strongest palindrome (C) involves the region near the transcription start site. It is possible to include the TATAAA box in a consensus palindrome (D).

Two other interesting patterns are noticeable, 5' from the TATAAA box. The first of these shown in Figure 9A has not been noticed before. This new palindrome is disjoint from and just 3' of the pattern in Figure 9B, which is approximately the Pelham pattern. In Figure 9C, we show the pattern of Pelham.[3]

IV. CONSENSUS IN ONE SEQUENCE

The algorithms in Sections II and III are designed to find consensus words and palindromes between several sequences where the sequences are in some initial alignment and a fixed amount of shifting is allowed. The motivation for those problems was the determination of consensus protein binding sites. Related motivation exists for studying consensus binding sites in a single sequence. Below, we give algorithms for consensus words and consensus palindromes in a single sequence. The algorithms are applied to study possible binding sites in a yeast regulatory sequence.

A. Algorithms

The sequence being studied is $a = a_1 a_2 \ldots a_N$. Initially, we discuss determining k-letter consensus words in the sequence, where the repeating words are not allowed to overlap. Our approach is related to that of the earlier sections.

Set $\lambda_w(v)$ = weight of the word v in the neighborhood of the k-letter word w. For a fixed w, set

$$S_i(w) = \max \left\{ \sum_v \lambda_w(v) \right\}$$

where the maximum is over nonoverlapping words v in $a_1 a_2 \ldots a_i$. The idea is to find the largest sum of weights, not allowing the words v to overlap. The winning word w* maximizes $S_N(w)$,

$$S(w^*) = \max_w S_N(w)$$

over the entire sequence $a_1 a_2 \ldots a_N$. The approach here is to find $S_N(w)$ for each w. This is easily done by recursion:

$$S_i(w) = \max \{S_{i-1}(w);\ S_{i-k} + \lambda_w(a_{i-k+1} \ldots a_i)\}$$

FIGURE 6. The patterns and their scores that correspond to A through G in Figure 5.

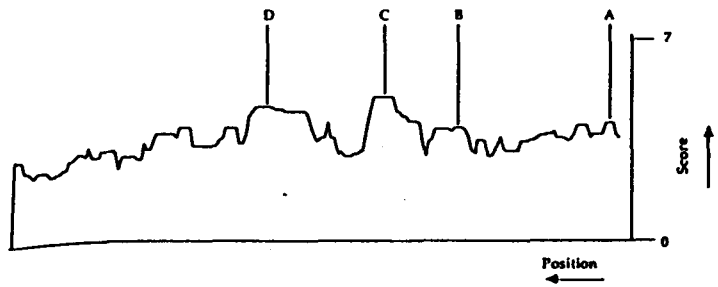


FIGURE 7. Graph for consensus palindrome scores in seven *Drosophila* heat-shock gene promoters. Here W = 25, palindrome size is 14, and up to mm = 6 are allowed.

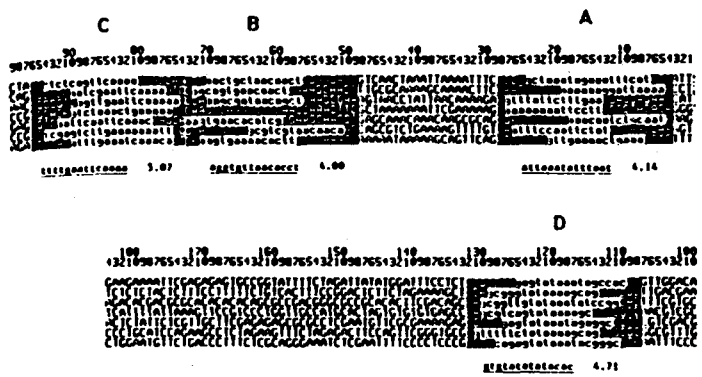


FIGURE 8. The patterns and scores that correspond to the labels A through D in Figure 7.

If $a_{i-k+1} \ldots a_i$ is in the neighborhood of w, and this is optimal, then $S_i(w)$ equals the second expression within the brackets above. If $a_{i-k+1} \ldots a_i$ is not in the neighborhood of w or if such a pattern is not optimal, then $S_{i-1}(w) = S_i(w)$. This establishes the recursion.

It would appear that $\max_w\{S_n(w)\}$ would take time $4^k \cdot N$ to compute. This number can be reduced, since the $S_i(w)$ need only be updated if $a_{i-k+1} \ldots a_i$ is in the neighborhood of w. Therefore, for every sequence word $a_{i-k+1} \ldots a_i$, we need only update $S_i(w)$ with w in its neighborhood. Therefore, the time complexity of this method is n(d)N, where n(d) is neighborhood size.

It is only the complication of computing neighborhoods that makes the palindrome algorithm distinct from the algorithm just discussed.

B. A Sequence from Yeast

Giniger et al[18] studied the yeast regulatory protein GAL4 which binds to four sites in the sequence UAS_G, to activate transcription of GAL1 and GAL10. The sequence from position 361 to position 486 is presented in Figure 10, along with our analysis. In Figure 10A, we

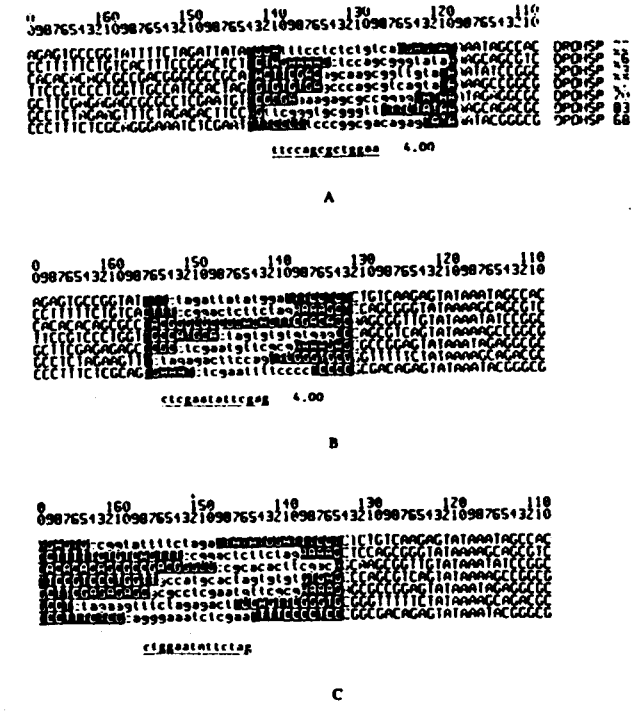


FIGURE 9. Additional consensus palindromes of interest, just upstream of the TATAAA box. A is a new pattern just 3' of the pattern in B, which shows the computer's version of Pelham's pattern, itself shown in C.

find the best consensus 17-letter palindrome, allowing up to 6 mismatches (out of the 16 letters, excluding the middle letter). The pattern is CGGATGAGNCTCATCCG and achieves score 2.75 with four occurrences. From 5' to 3', the first three patterns coincide with those of Giniger et al., while the fourth (rightmost) pattern differs in location.

For completeness, we also give a repeats analysis for consensus words in Figure 10B, with k = 8, allowing 2 mismatches. The consensus pattern CGCCGTCC occurs five times with score 3.75.

V. LONG CONSENSUS PATTERNS

The consensus algorithms of Sections II, III, and IV depend on storing scores (and other information) for all patterns of interest. With k-letter words of DNA, this implies that storage is proportional to 4^k. For k = 10, 4^k = 1,048,576. In the introduction we mention the Alu family of repeats where k = 300. Obviously, the techniques above, practical for k = 9 to 12 larger, will not be of any use for these problems.

While we do not know, for larger k, of any algorithms guaranteed to be optimal for computing

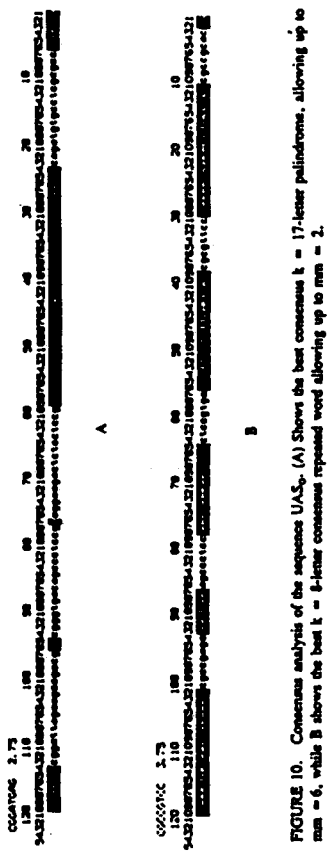


FIGURE 10. Consensus analysis of the sequence UAS_G. (A) Shows the best consensus k = 17-letter palindrome, allowing up to mm = 6, while B shows the best k = 8-letter consensus repeated word allowing up to mm = 2.

$$\max_{-}\{s_{-}\}$$

we do have some useful and practical algorithms to suggest. The following method was developed for a study in progress with Wool, McNalley, and Jones. Our problem is finding the long consensus words among a set of sequences.

Recall that above, in the passage from the set of all k-letter words to the set of all k-letter palindromes, we effectively decreased the size of the set of possible consensus patterns from 4^k to $4^{k/2}$. The simple idea here is to decrease the set of possible consensus words to include only those k-letter words actually occurring in the DNA sequences themselves. That is, our consensus word w must appear exactly somewhere in

$$\begin{array}{lll} a_{1,1}a_{1,2} & \cdots & a_{1,N} \\ a_{2,1}a_{2,2} & \cdots & a_{2,N} \\ \cdots\cdots & \cdots & \cdots \\ a_{n,1}a_{n,2} & \cdots & a_{n,N} \end{array}$$

We then look for the best occurrence of w in each sequence. It is guaranteed to occur exactly in at least one sequence. If no insertions or deletions are allowed, searching one sequence for w can be done in time and storage $k(N - k + 1)$. (Actually, using techniques of computer science, this number can be reduced to approximately $N + k$.) However, it is unlikely that in long DNA patterns, insertions and deletions can be ignored. The dynamic programming methods of Chapter 3[10] allow us to find the best fit of w into $a_{i,1} \ldots a_{i,N}$ in time and storage proportional to k.N, allowing insertions and deletions. Thus, the score s_{-} can be found in R·k·N steps. Still, this must be done for each sequence word, so that the time is proportional to $R^2(N - k + 1)(k)(N)$.

This number is large for many problems, but is not much larger than the time of making all pairwise comparisons, $\binom{n}{2}N^2$. The difficulty of the pairwise comparisons approach is that best matching segments between two sequences might not have any relationship with the consensus word between (say) 60 sequences. This is certainly the case with the bacterial promoters discussed in Section II.

The method can be made adaptive by taking the words matching the consensus w and further refining it, modifying the pattern to one not actually in the data. The procedure is iterated until the pattern does not change.

There is a way to use all pairwise comparisons which does not require us to set word size before the search. The maximum segments algorithm from Chapter 3[10] computes a matrix $H = (H_{ij})$ where H_{ij} is interpreted as the maximum of all subsequences ending in i (from one sequence) matched with a subsequence ending in j from the second sequence. If the first sequence is $a_i = a_{i,1} \ldots a_{i,j} \ldots a_{i,N}$, then $\max_j H_{ij}$ gives the best match of $a_{i,\ell} \ldots a_{i,j}$ matching the second sequence for all $\ell \leq i$. We use

$$\sum_{\text{sequences}} \max_j H_{ij}$$

to score subsequences from a that end at $a_{i,j}$.

Clearly, we have suggested algorithms that are not too efficient. Still, they will often run on available computers in reasonable time and are much more informative than trying to look at the sequences two at a time.

VI. ESTIMATES OF STATISTICAL SIGNIFICANCE

Statistics is useful in sequence analysis to provide clues about what patterns or alignments should be taken seriously. It is by no means our intention to equate biological significance

with statistical significance, however, either of these terms might be defined. To use a statistical approach, the sequences must be viewed as following some model of randomness. Then the analyst estimates the likelihood, under the model, of observing some pattern at least as extreme as that actually observed in the sequences. The logic employed is that patterns that are extremely likely to occur in random sequences of similar composition should not be examined as carefully as patterns which are very unlikely to occur. Biology must provide the resolution of these issues; statistics is only a tool to distinguish patterns of possible interest.

We now discuss the issue of what model of randomness to use. Sometimes there is confusion of what actually needs to be modeled. To model the sequences themselves is not the object. If it were, higher order Markov chains, with memory of at least 2, would be required.[8] However, what is really needed is a model of randomness that produces, for example, the same distribution of maximum scores as that obtained from unrelated biological sequences. The distribution of scores from best matching segments between two unrelated biological sequences has been shown to be very well modeled by that from comparing sequences of identical composition with independent and identically distributed bases. See Smith et al.[9] or Chapter 3[10] for discussions of this work. This is the basis for our assumptions of very simple statistical models in this section.

When long sequences are being compared, the recently developed extreme value theory, referred to here as the log(n) distribution, is very useful in assessing the best matching segments between the sequences. Karlin and collaborators[4] have developed a theory which gives the distribution of the longest exact match, common between L or more sequences of length N, with a total of R sequences ($2 \leq L \leq R$). A theory allowing increasing amounts of mismatch (including insertions and deletions) has been developed elsewhere and is discussed in Section VII of Chapter 3.[10] Since the log(n) theory is treated elsewhere in this book, we discuss other approaches here.

A. The Binomial Distribution and Large Deviations

When the sequences are not long enough for the log(n) theory to apply, there is another useful theory to consider. Let α be the probability of a word occurring within the neighborhood of w in a sequence, within the window. We refer to this event as a "success". Here, we give an estimate of the probablity of $n \geq \beta R$ successes where $\beta > \alpha$. The probability of n successes is, by the binomial distribution,

$$\binom{R}{n} \alpha^n (1 - \alpha)^{R-n}$$

and we wish to evaluate

$$\sum_{n \geq \beta R} \binom{R}{n} \alpha_n (1 - \alpha)^{R-n}$$

If R is small, this sum can be directly calculated.

When R is not small, it is necessary that there be enough sequences for the law of large numbers to be valid. The case of the *Escherichia coli* promoters with R = 59 sequences is a good one. The approach we now discuss is from the theory of large deviations and is found in Bahadur[19] and Ellis.[20] Large deviation theory deserves to be more widely applied.

To present the most basic feature of the theory, assume that we have R trials with the probability α of "success" at each trial. Since R is large, the law of large numbers assures us of about αR successes. The central limit theorem gives additional information about the number of successes. The large deviation estimate of this probability is

$$P(\text{at least } \beta R \text{ successes}) \sim e^{-R\mathrm{H}(\beta, \alpha)}$$

where

$$\mathrm{H}(\beta, \alpha) = \beta \log(\beta/\alpha) + (1 - \beta)\log(1 - \beta)/(1 - \alpha)$$

When $\beta = 1$, $\mathrm{H}(\beta, \alpha) = \log(1/\alpha)$ and $e^{-R\mathrm{H}(\beta,\alpha)} = \alpha^R$.

How is the large deviation estimate to be used in the case of consensus patterns in *E. coli* promoters? Let us take a window size of W = 9, a word size of k = 6, and allow up to mm = 2 mismatches. The sequences are approximately N = 60 bases long. It is assumed that all bases are equally likely, P(A) = P(T) = P(G) = P(C) = 1/4, and independent.

The probability that a given pattern appears on a given line with fixed window position

$$\alpha = (W - k + 1) \cdot F \cdot 4^{-k}$$

where F is the neighborhood size. The factor W − k + 1 is to allow for the different positions of the word in the window. The probability p that at least βR of the lines have the consensus word w for a *fixed* window position is

$$p = e^{-R\mathrm{H}(\beta,\alpha)}, \text{ fixed window position and fixed word}$$

while

$$p = (N - W + 1)\, e^{-R\mathrm{H}(\beta,\alpha)}, \text{ any window position and fixed word}$$

is the probability for some window position. Since the analysis has been done for a fixed word w, we can relax that to any word, any window position by multiplying by 4^k, the number of k-letter words:

$$p = (N - w + 1)\, 4^k\, e^{-R\mathrm{H}(\beta,\alpha)}, \text{ any window position and any word}$$

For our specific numbers, k = 6, N = 60, R = 59, and W = 9, we still need to determine α and β. To calculate α, recall that

$$\alpha = (W - k + 1)F\, 4^{-k}$$

where F is neighborhood size. Since F = 154 for up to 2 mm,

$$\alpha = (9 - 6 + 1)(154)4^{-6}$$
$$= 0.150\ldots$$

In Section II, we found 9 exact occurrences of TATAAT, 17 with 1 mm, and 18 with 2 mm. This makes $\beta = 44/59 \sim 0.746$. Then $\mathrm{H}(0.746, 0.150) = 0.890$. For any window position and any word,

$$p = (60 - 6 + 1)\, 4^6\, e^{-59(0.890)} \sim 4.2 \times 10^{-18}$$

Therefore, the TATAAT pattern is extremely unlikely in random sequences.

B. Simulations

The promoter sequences unfortunately are not even close to all being composed of 25% A, etc. The problem is that base composition differs from sequence to sequence. The large deviation theory could be applied for sequences of differing compositions, but that would

be difficult. In these cases, we turn to simulation for additional insight, although simulation will never be able to estimate significance levels such as 4.2×10^{-16}.

Not only is the theory of large deviations hard to apply when the sequences have different statistical characteristics, but it only predicts the number of sequences in a neighborhood. We are actually interested in the maximum score of consensus words. For these reasons we employ simulation.

The approach is quite simple; we generate sequences with the same number of A, T, G, and C as the promoter sequences, but put those letters in a random order. Then we run the program to get a sample of the maximum score of consensus words. The entire data set (N = 60 letters) is scanned, with the window in all positions. The value of the maximum in these sequences, with N = 60, w = 9, k = 6, and mm = 2, is approximately 17. The score of TATAAT is, from Section II.C, 35.17.

ACKNOWLEDGMENT

This work was supported by grants from the National Institutes of Health and the System Development Foundation.

REFERENCES

1. Hawley, D. K. and McClure, W. R., Compilation and analysis of *Escherichia coli* promoter DNA sequences, *Nucl. Acids Res.*, 11, 2237, 1983.
2. Mulligan, M. E., Hawley, D. K., Entriken, R., and McClure, W. R., *Escherichia coli* promoter sequences predict *in vitro* RNA polymerase selectivity, *Nucl. Acids Res.*, 12, 789, 1984.
3. Pelham, H. R. B., A regulatory upstream promoter element in the drosophila hsp 70 heat-shock gene, *Cell*, 30, 517, 1982.
4. Karlin, S., Ghandour, G., Ost, F., Tavare, S., and Korn, L. J., New approaches for computer analysis of nucleic acid sequences, *Proc. Natl. Acad. Sci. U.S.A.*, 80, 5660, 1983.
5. Martinez, H. M., An efficient method for finding repeats in molecular sequences, *Nucl. Acids Res.*, 11, 4629, 1983.
6. Sobel, E. and Martinez, H. M. A multiple sequence alignment program, *Nucl. Acids Res.*, 14, 363, 1986.
7. Wilbur, W. J. and Lipman, D. J., Rapid similarity searches of nucleic acid and protein data banks, *Proc. Natl. Acad. Sci. U.S.A.*, 80, 726, 1983.
8. Smith, T. F., Waterman, M. S., and Sadler, J. R., Statistical characterization of nucleic acid sequences functional domains, *Nucl. Acids Res.*, 11, 2205, 1983.
9. Smith, T. F., Waterman, M. S., and Burks, C., The statistical distribution of nucleic acid similarities, *Nucl. Acids Res.*, 13, 645, 1985.
10. Waterman, M. S., Sequence alignments, in *Mathematical Methods of DNA Sequences*, Waterman, M. S., Ed., CRC Press, Boca Raton, Fla., 1988.
11. Waterman, M. S., Galas, D., and Arratia, R., Pattern recognition in several sequences: consensus and alignment, *Bull. Math. Biol.*, 46, 515, 1984.
12. Mengeritsky, G. and Smith, T. F., Recognition of characteristic patterns in sets of functionally equivalent DNA sequences, *CABIOS*, 3, 223,1987.
13. Galas, D. J., Eggert, M., and Waterman, M. S., Rigorous pattern-recognition methods for DNA sequences. Analysis of promoter sequences from *Escherichia coli*, *J. Mol. Biol.*, 186, 117, 1985.
14. Wu, H. M. and Crothers, D., The locus of sequence-directed and protein-induced DNA bending, *Nature (London)*, 308, 509, 1984.
15. Koo, H.-S., Wu, H.-M., and Crothers, D. M., DNA bending at adenine-thymine tracts, *Nature, (London)*, 320, 501, 1986.
16. Kolb, A., Spassky, A., Chapon, C., Blazy, B., and Buc, H., On the different binding affinities of CRP at the *lac*, *gal*, *malT* promoter regions, *Nucl. Acids Res.*, 11, 7833, 1983.
17. Pelham, H. R. B. and Bienz, M., A synthetic heat-shock promoter element confers heat-inducibility on the herpes simplex virus thymidine kinase gene, *EMBO J.* 1(1), 1473, 1982.

18. Giniger, E., Varnum, S. M., and Ptashne, M., Specific DNA binding of GAL4, a positive regulatory protein of yeast, *Cell*, 40, 767, 1985.
19. Bahadur, R. R., Some limit theorems in statistics, *SIAM*, Philadelphia, 1971.
20. Ellis, R. S., *Entropy, Large Deviations, and Statistical Mechanics*, Springer-Verlag, New York, 1985, 71.