Volume 14 Number 1 1986 Nucleic Acids Research

Sequence landscapes

B.Clift, D.Haussler, R.McConnell, T.D.Schneider[1] and G.D.Stormo[1]

Department of Mathematics and Computer Science, University of Denver, Denver, CO 80208, and [1]Department of Molecular, Cellular and Developmental Biology, University of Colorado, Boulder, CO 80309, USA

Received 16 July 1985

ABSTRACT

We describe a method for representing the structure of repeating sequences in nucleic-acids, proteins and other texts. A portion of the sequence is presented at the bottom of a CRT screen. Above the sequence is its landscape, which looks like a mountain range. Each mountain corresponds to a subsequence of the sequence. At the peak of every mountain is written the number of times that the subsequence appears. A data structure called a DAWG, which can be built in time proportional to the length of the sequence, is used to construct the landscape. For the 40 thousand bases of bacteriophage T7, the DAWG can be built in 30 seconds. The time to display any portion of the landscape is less than a second. Using sequence landscapes, one can quickly locate significant repeats.

INTRODUCTION

It is often useful to examine nucleic-acid sequences for subsequences that are either unusually common or rare. This may be done by determining the composition up to some fixed length of subsequence, an operation requiring $O(n)$ steps[1]. One can also create an index of all minimum unique subsequences in $O(n\log(n))$ steps for an average sequence (1,2). Once unusual subsequences are found, it is useful to locate them on the entire sequence to determine their positions relative to one another and to other important features of the sequence.

We report here a method for displaying the number of occurrences of every subsequence. We call this a "landscape"[2]. The landscape is derived from a directed acyclic word graph (DAWG) of the sequence, which is created in $O(n)$ steps (3,4). Unusually long repeated sequences show up as peaks on the landscape, and unusually rare ones are seen as valleys. The positions of subsequence features are maintained because the landscape is displayed above the linear sequence. In addition, short sequences that predict (are always part of) longer sequences are obvious in the landscape.

Another common analysis is to compare a sequence with other sequences to find significant similarities. This operation can be done with a landscape. The DAWG is created from one or more

[1]The notation $O(n)$, pronounced "*order of n*," means that for a sequence n bases long the calculation time will be An+B, where A and B are constants.
[2] The landscape program is written in C and runs under the UNIX (TM) operating system with most CRT terminals. It is available for cost of distribution over CSNET or by ASCII tape written at 1600 bpi. Inquiries should be made to Schneider and Stormo (toms@boulder or stormo@boulder).

© IRL Press Limited, Oxford, England.

sequences, called the source sequences. The landscape is displayed over a target sequence so that the numbers in the landscape refer to how many times each subsequence of the target sequence occurs in the set of source sequences. Peaks are unusually long subsequences in the target sequence that also occur in the source set of sequences. Regions of the target sequence containing exact matches to sequences in the source set will be identified, even if their orders are scrambled. Also, long inexact matches can often be identified because they tend to contain many shorter exact matches concentrated in one region. An inexact match may be significant because it is similar to a *collection* of sequences but not necessarily to any one of those sequences.

METHODS

Definitions

The methods for displaying landscapes of sequences are completely general, and apply to sequences composed of letters from any finite alphabet. We use the generic term *string* to refer to a sequence of characters chosen from some arbitrary (but fixed) finite alphabet. A *substring* of a string s is a string y of letters that occur contiguously in s, i.e. such that $s = xyz$ for some (possibly empty) strings x and z. The *frequency* of y in s is the number of distinct (possibly overlapping) occurrences of y in s.

The Landscape and What It Means

A landscape consists of a portion of a *target* string displayed on the bottom line of a CRT screen and the two-dimensional array in the plane above the string, depicting the number of times the substrings of the target string occur in a *source* string (multiple source strings are also allowed).

To understand the meaning of the landscape, it is useful to think of the plane above the target string as consisting of *cells*. Each cell refers to a unique substring of the target string. A cell's position is specified by its row, counting from the bottom of the display with row 1 just above the string, and its column, counting from the beginning of the string with column 1. A cell in row i and column j refers to the substring of length i ending at position j of the string. In a *primitive* landscape every cell contains a number giving the number of occurrences of the substring in the source string.

The substring a cell refers to may be visualized by projecting straight down to the target string at the bottom of the display for the right end of the substring and diagonally down to the left for the beginning of the substring. For instance, a *primitive* landscape with target string GTAGTAAAC and source string GTAGTAAAC is shown in Figure 1a. Since target and source strings are identical, the numbers in the landscape are frequencies in the target string itself. The landscape software is often used in this mode. The cell containing the 1 at the top of the triangle refers to the entire string (positions 1-9) and indicates that that string occurs only once. The cell directly below it refers to the substring in positions 2-9 (i.e. TAGTAAAC), while the cell below and to the left refers to the substring in positions 1-8. The cells in row 1 refer to single characters in the string.

Because these numbers are dense, the raw data are hard to interpret on large strings with complex internal structures. The landscape program presents a more abstract representation of the

a)
```
        1
       11
      111
     1111
    11111
   111111
  2112111
 22122221
224224441
GTAGTAAAC
```

b)
```
        1
       /|
      / |
     /  |
    /   |
   /    |
  2  2  |
 /| /|22|
/ 4/ 444|
GTAGTAAAC
```

c)
```
        1
       /|
      / |
     /  |
    2  2|1
   /| /|2|    2
11/ 4/ 44|114/42222
CCGTAGTAACCCATATTGG
```

Figure 1. Example landscapes of DNA sequences.
a. The primitive (complete) landscape of the sequence GTAGTAAAC (see text for the meaning of each number).
b. The processed landscape of the same sequence as in part a.
c. The landscape of the sequence CCGTAGTAACCCATATTGG using GTAGTAAAC as the source string.

data. Figure 1b demonstrates how the *primitive* landscape in figure 1a is processed to be more legible. The primitive landscape is partitioned into regions of cells whose corresponding substrings all occur with the same frequency. The number of occurrences is shown only in the highest cell of the region. Dash marks show the boundary, and the interior is filled with spaces.

These regions are defined using the concepts of implication and equivalence for strings (4). In the string GTAGTAAAC, every time T occurs it is preceded by G and followed by A, thus T always occurs in the larger context of the string GTA. There is no string longer than G that always precedes T, and no string longer than A that always follows T. Thus GTA is the maximal context "implied" by T.

In general, whenever a substring y of a target string s is always preceded by a string x and followed by a string z, we say that *y implies xyz (with respect to s)*. Either x or z may be empty. The longest such string xyz is called the *implication of y (in s)*. Every substring of a target string s will have a unique implication. In our example string, the implication of each of the strings T, G, GT, TA, and GTA is GTA. GTA is its own implication since there is no string that always follows GTA or always precedes GTA. Likewise, the strings A and AA are their own implications. This may seem surprising at first, since it appears that AA always occurs in the context of AAA. However, according to our definition AA does not imply AAA, since it is not the case that both occurrences of AA are preceded by A (or followed by A). Finally, because they occur only once, the remaining substrings all have the same implication, namely, the entire string GTAGTAAAC.

Two substrings of *s* are called *equivalent (with respect to s)* if they have the same implication. This obviously means that they also have the same frequency of occurrence. Sets of substrings with the same implication are called *two-way equivalence classes* (two-way because implication can add context both to the left and to the right of a substring). In our example, the set of substrings of GTAGTAAAC is partitioned into four two-way equivalence classes, {T,G,GT,TA,GTA} (freq. 2), {A} (freq. 4), {AA} (freq. 2) and a class for the remaining substrings (freq. 1). It is these equivalence classes that form the regions of the processed landscape (henceforth called simply "the landscape"). The frequency is displayed in the cell that denotes the common implication of the substrings in the class (called the *representative* of the class). The representative will always be the longest substring in the class and other members of the class will be substrings of this string. Taken together, the cells denoting substrings in a two-way equivalence class form a region that is triangular on top, with frequency label at the apex and dashed lines forming the sides, but in general ragged on the bottom. This is because there are often equivalence classes inside a mountain which are composed of smaller substrings. The class with representative GTAGTAAAC, with apex labeled 1 in figure 1b has this typical shape. The entire landscape has the appearance of a mountain range, with larger triangular peaks overlayed by smaller peaks.

Even a glance at a landscape display gives a wealth of statistical and structural information about a string. For example, the left most peak with '2' at the top in figure 1b indicates that GTA occurs twice in the string. The region below each '2' indicates that T, G, GT, and TA imply GTA and thus every occurrence of G is followed by TA, every occurrence of T is preceded by G and followed by A, etc. The overlayed "peak" for the substring A, labeled '4', indicates that A does not imply GTA, but in fact also occurs in two other contexts (i.e. not preceded by GT). Finally, GTA has the highest peak with a frequency greater than 1, so GTA is the longest repeating substring in the string.

When the source string is different from the target string the frequency number given at the apex of a peak refers to the number of times the substring below it occurs in the source string. Figure 1c is the landscape obtained by using GTAGTAAAC as the source string and CCGTAGTAACCCATATTGG as the target string. The tallest peak shows that the longest substring of the target string that also occurs in the source string is GTAGTAA. Inside this is the peak denoting GTA, which occurs twice in the source string. AAC also occurs once in the source string, and is implied by AC and C. Note that although TA occurs 3 times in the target, it only occurs 2 times in the source, and each TA in the target is therefore labeled with a 2.

The landscape can also be given for a target string using frequencies taken from a set of source strings. Here the numbers displayed indicate the frequencies of occurrence in the combined set of source strings. More examples of these various types of landscapes appear in the RESULTS section of this paper. In examples where source and target strings are identical, only peaks with frequency greater than 1 are displayed since the single peak with frequency 1 always sits above the entire string, and thus gives no additional information. Another convention is to use an asterisk to represent frequency numbers that are 10 or larger, so that the proper alignment of the cells is preserved (see also the discussion of the "zoom-in" command below).

Functions Provided By the Landscape Program

The landscape program is cursor-oriented. The user may move the cursor to any cell on the screen, or may scroll horizontally to see other sections of the string and vertically to see the tops of tall mountains. Upon placing the cursor on a cell, one may quickly discover what substring the cell refers to by using the *show* command which puts this substring into reverse video and capitalizes it.

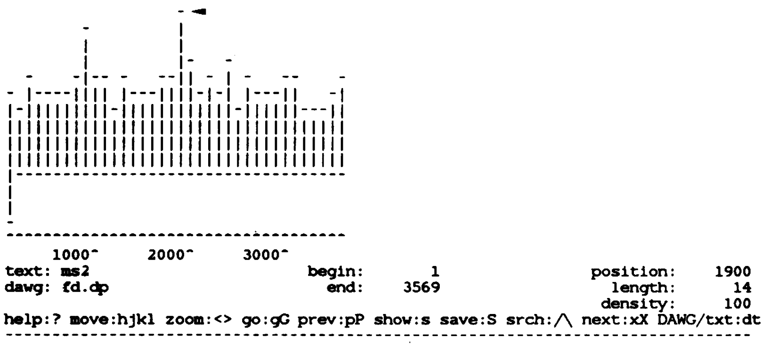
The *search* function provides a quick and easy method for locating substrings in the target string. The string used as the search key may be obtained from the screen by placing the cursor on its corresponding cell, from the keyboard, or from an external file. The search function moves the cursor to the cell corresponding to the next occurrence of the search key in the target string and the landscape surrounding that substring is displayed.

The *zoom-in* command is provided for expanding the asterisks, displaying larger frequencies in each cell. Although this reduces the amount of landscape that fits on a screen, it allows the user to examine small sections of a string more closely.

The *zoom-out* command provides a more global view of a string using a *condensed landscape*. This allows quick identification of substrings with unusually large or small frequencies. Each column in a condensed landscape summarizes a block of columns from a normal landscape, where the size of the block is some factor of ten. Vertical lines, bounded by dashes at the top and bottom, are displayed in these columns showing the highest mountain and lowest valley in the section of the normal landscape that they represent.

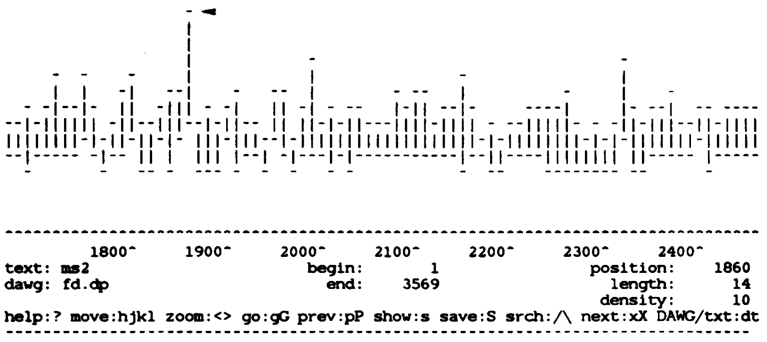
Figure 2a shows a condensed landscape using all of bacteriophage ms2 as the target string (5) and bacteriophage fd as the source (6). The user may now place the cursor on the highest mountain, at 1900. This position summarizes the landscape from 1801 to 1900. By typing '>', for *zoom-in*, the user increases the magnification and begins to zero in on this tallest mountain. Figure 2b shows the new display. The cursor appears at 1850, since that is the midpoint of the previous range (1801 - 1900). The mountain is now at position 1860, so the user may move the cursor one cell to the right, and *zoom-in* once again (figure 2c). The cursor appeared at 1855 and has been moved to 1857, where the user has used the show command to highlight the substring that corresponds to this mountain, here shown in capitals. This simple method has easily located the longest substring found in fd that also occurs in ms2. The 1 at the top of the peak indicates that this string occurs only once in fd. The search function described above may be used to locate other places in ms2 that this substring occurs, if any exist. On a Pyramid 90x computer, each of these screens is computed as fast as a 9600 baud terminal can display it.

At any time while using the landscape program, the user is allowed to change either the source or the target string. This may be used to view a target with landscapes derived from different sources. The user's position in the current target string remains, but the frequencies in the landscape change as the target is compared to different source strings. On the other hand, by let-a)



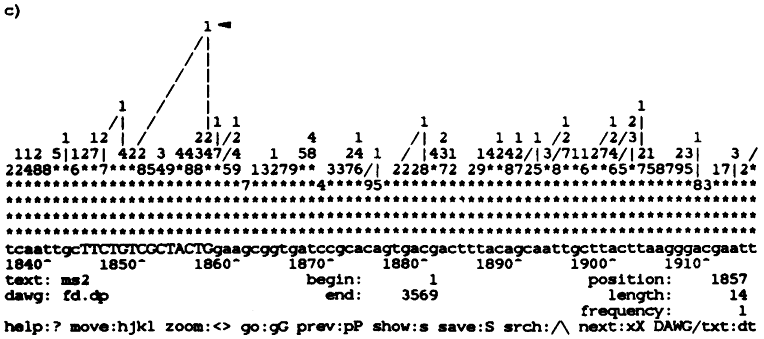
```
     1000^       2000^       3000^
text: ms2                          begin:       1               position:    1900
dawg: fd.dp                          end:    3569                 length:      14
                                                                 density:     100
help:? move:hjkl zoom:<> go:gG prev:pP show:s save:S srch:/\ next:xX DAWG/txt:dt
```

b)



```
          1800^       1900^       2000^       2100^       2200^       2300^      2400^
text: ms2                          begin:       1               position:    1860
dawg: fd.dp                          end:    3569                 length:      14
                                                                 density:      10
help:? move:hjkl zoom:<> go:gG prev:pP show:s save:S srch:/\ next:xX DAWG/txt:dt
```

c)



```
                         1 ◄
                        /|
                       / |
                      /  |
                     /   |
            1       /    |                                                 1
           /|      /     |1 1                  1              1     1 2|
  1   12  |  /       22|/2        4    1       /| 2       1 1 1 /2    /2/3|      1
 112 5|127| 422 3 44347/4    1   58    24 1   / |431   14242/|3/711274/|21   23|    3 /
22488**6**7***8549*88**59 13279** 3376/| 2228*72 29**8725*8**6**65*758795| 17|2*
***************************7*******4****95*************************************83*****
****************************************************************************************
****************************************************************************************
****************************************************************************************
tcaattgcTTCTGTCGCTACTGgaagcggtgatccgcacagtgacgactttacagcaattgcttacttaagggacgaatt
1840^     1850^      1860^      1870^      1880^      1890^      1900^      1910^
text: ms2                          begin:       1               position:    1857
dawg: fd.dp                          end:    3569                 length:      14
                                                               frequency:       1
help:? move:hjkl zoom:<> go:gG prev:pP show:s save:S srch:/\ next:xX DAWG/txt:dt
```

Figure 2. Some example landscapes as seen on a CRT terminal.
a. The condensed landscape of ms2, using fd as the source string. Each vertical line shows the maximum peak and minimum valley for 100 bases of sequence. The cursor (arrow) is pointed at the highest peak, at position 1801 to 1900 and level 14.
b. Each vertical line now shows the maximum peak and minimum valley for 10 bases. The cursor is now at position 1851 to 1860.
c. The landscape showing the sequence on the bottom line. The cursor is at the peak at position 1857, where a 14-long sequence, TTCTGTCGCTACTG, occurs that is also found in fd. The show feature, s, has been used to capitalize the string associated with the cursor.

ting the source be a fixed set of related strings and varying the target among the individual strings in this set, patterns common in the entire set can be recognized in each of the individual strings.

Other Program Features

Strings are stored in external UNIX files according to a simple format that allows multiple strings in one file. Texts from a single file are treated as a group when used as the source string and individually when used as the target. An interface to the Delila system programs (7,8) has been written that allows one to extract sets of sequences from the GenBank (TM) or EMBL databases (9).

Many features are provided for recording sessions. The first of these is the tracing feature. At the start of a session, the user may specify that the session is to be recorded. This causes the user's keyboard input to be saved, character by character. When the program is terminated, the user is told the name of the file in which the session has been saved, which is built directly from the date and time of the session. Running the program later and specifying the trace option causes a "movie" to be run of the recorded session. The user sees the session fly by on the screen, just as if it was being controlled from the keyboard. This movie may be stopped at any time, allowing new branches off the previous session. These tracing sessions may also be recorded. This versatile capability may be seen as the beginnings of an automatic laboratory book facility, with sessions viewed as experiments.

Also desirable is the capability of printing landscapes on hardcopy devices. This may be done in two ways. At any time the user may save the screen in a UNIX file. Later these screens can be printed out on a variety of hardcopy devices. An alternative method is to run the program from a hardcopy terminal, in which case the landscape is printed to the right of the sequence that runs down the page. This has the advantage of producing long, uninterrupted landscapes, but all the screen-oriented capabilities are lost.

How the Landscape is Computed

Description of the DAWG. Every number in a landscape indicates how many times a substring in the target occurs in the source. To determine each number one could simply search the source for each target substring. However, for a target of size n, there are $n(n+1)/2$ substrings. Even with fast string search algorithms (10) this procedure would be slow. Nevertheless, the landscape program rapidly performs the analysis on large strings.

Before building a landscape, the program first builds a representation of the source string which is called the DAWG (Directed Acyclic Word Graph) (3). The DAWG speeds the computation of the landscape in two ways. First, given a substring of the target, the DAWG quickly reveals the frequency of the substring in the source, eliminating the need to repeatedly traverse the source string. Second, substrings of the source are arranged in the DAWG so that their implications are easily calculated, facilitating the construction of the landscape mountains.

Using the DAWG, the limiting factor on the size of strings that can be analyzed is not the time to compute the landscape, but the size of the random access memory (RAM) required to store the DAWG efficiently. For instance, constructing the DAWG using the DNA of bacteriophage T7 as the source string (40000 base pairs) takes 30 seconds on a Pyramid 90X containing 8 megabytes of RAM memory. With only 4 megabytes available, disk accesses increase the time to 10 minutes.

A DAWG is a directed graph, i.e. a set of nodes connected by directed edges (see e.g. 11). Each edge is labeled with a character from the source string, while each node is labeled with a frequency. There is also a special node called the *start node.* The set of paths from the start node to other nodes in the DAWG corresponds to the set of substrings of the source string. The path for a given substring is found by starting at the start node and traversing the set of edges that spells the substring. The frequency label in the node at the end of the path gives the number of occurrences of the substring in the source string.

Figure 3 shows the DAWG for the string GTAGTAAAC. Suppose you want to find the number of times AAC occurs in GTAGTAAAC. If you begin at the start node and move along the edge labeled A, you get to a node labeled with frequency 4, indicating that A occurs four times in GTAGTAAAC. When you then follow the edge labeled A leading out of this node, you get to a node labeled with frequency 2, indicating that AA occurs twice. If you then follow the edge labeled C, you get to an edge labeled with frequency 1, indicating that AAC occurs only once in GTAGTAAAC. As another example, if you trace the path for the string TAT in the DAWG you will find that there is no edge leading out of the node for TA that is labeled T. This indicates that TAT does not occur at all in GTAGTAAAC.

There are often multiple paths from the start node to a node. These paths spell out the set of substrings the node stands for. All substrings in the set have the same frequency in the source string, making it possible for the node to have a single frequency label yielding the correct frequency for any of these substrings. This is because the nodes of the DAWG are related to the two-way equivalence classes of substrings of the source string, introduced above.

As described above, the landscape for a string with respect to itself is partitioned into regions of cells denoting two-way equivalence classes of substrings. Each of these regions can be further partitioned into columns of cells. Each of these columns defines a *right equivalence class.* All substrings in a right equivalence class lead to the same node in the DAWG, and there is one node in the DAWG for each right equivalence class in the string. For example, the two-way equivalence class represented by GTA in GTAGTAAAC contains three columns and thus is partitioned into three right equivalence classes, {G}, {T, GT} and {TA, GTA}. The two-way equivalence class represented by GTAGTAAAC itself is partitioned into six right equivalence classes. The other two two-way equivalence classes, for A and AA, cannot be partitioned further, and thus are already right equivalence classes. This gives a total of 11 right equivalence classes for GTAGTAAAC. These, combined with the start node, give the 12 nodes of the DAWG for this string. To make the correspondence between the DAWG and the landscape complete, we define a special right equivalence class containing only the empty string which corresponds to the start node.

Because they come from a column in the landscape, the substrings in a right equivalence class are all suffixes of the longest string in the class, corresponding to the cell at the top of the

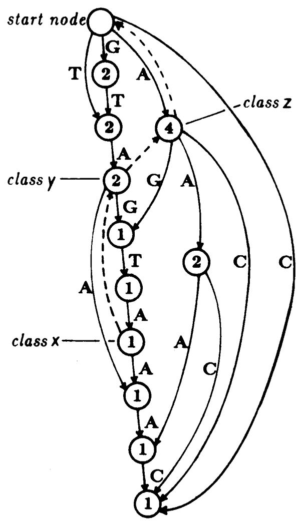


Figure 3. The DAWG for the sequence GTAGTAAAC. See text for details.

column. As with two-way equivalence classes, this longest string will be called the *representative* of the right equivalence class. Not all suffixes of the representative will be in the right equivalence class, since any particular column of the landscape may cut through several two-way equivalence classes, thus generating several right equivalence classes. The *suffix class* of a right equivalence class is the right equivalence class that lies immediately below it in the column of a landscape. For instance, in the 6th column of Figure 1b, the suffix class of {GTAGTA, TAGTA, AGTA} is {GTA, TA} and the suffix class of {GTA, TA} is {A}. The suffix class of {A} is empty. Nodes for these suffix classes in the DAWG for GTAGTAAAC are marked as classes x, y and z, respectively, in Figure 3.

The DAWG gives a complete record of the frequencies of all the substrings that occur in the source string it is built for, and even shows which strings are not substrings of the source string, as described above. Moreover, this information can be retrieved in time proportional to the length of substring, not the length of the source string. This amounts to a nearly instantaneous retrieval of the frequency of any short substring of a large string. Even though there can be up to $n(n+1)/2$ substrings in a source string that is n characters long, this string will have at most $2n-1$ right equivalence classes, so the DAWG will contain no more than $2n-1$ nodes (3). In addition, the DAWG contains no more than $3n - 3$ edges, and can be built in $O(n)$ time (3).

Because of fast construction and retrieval times, the only limiting factor in using the DAWG is the space it occupies (about 50 times more space than the raw source string in our implementation). A substantial part of this space should be RAM memory, for reasons discussed in (4). More compact versions of the DAWG are also discussed in that paper, where nodes correspond to two-way equivalence classes of the source string. Average case size analysis for the DAWG built from random strings is given in (12). Our experience with DNA indicates that theoretical size predictions for random strings on four letters (each equally likely) are good estimates for most DNA sequences.

How the Landscape is Drawn Using the DAWG. The landscape is constructed one column at a time from left to right. The algorithm begins with the highest right equivalence class in a column, and puts a frequency number or slash in the cell denoting the upper boundary of this class. This number or slash is called a *boundary mark*. If the boundary mark is a slash, the algorithm fills in the rest of the cells down to the next boundary mark with blanks; if it is a number, it fills them in with vertical bars. Then it inserts a number or a slash in the cell denoting the top boundary of the next right equivalence class down, and repeats the process for successive equivalence classes until the bottom of the column is reached.

Before the landscape is drawn, each node of the DAWG (except the start node) must be augmented with a label indicating the length of the representative of the class it denotes (a *length label*) and a pointer to the node denoting its suffix class (a *suffix pointer*). Pointers to suffix classes are created during construction of the DAWG, and there are fast methods of labeling nodes with the length of their longest member (4). The suffix pointers of the nodes for the three right equivalence classes discussed above are illustrated in Figure 3 with dashed lines.

Let *topnode* be the node denoting the right equivalence class appearing at the top of the landscape in the column currently being drawn and *topheight* be the height of the boundary mark for this class. The nodes denoting successively lower right equivalence classes in the column can be found by starting at topnode and traversing suffix pointers until the start node is encountered. The length label of each of these nodes denotes the length of the representative of each of these classes, and hence, the height in the column of the boundary marks for each of the classes.

A right equivalence class is drawn in the current column as follows. Let *currentnode* be the node that denotes this class. Let *nextletter* denote the character in the next column to the right. If currentnode is topnode, then currentheight is topheight, otherwise it is the value in the length label of the node. The boundary mark, which must be drawn at currentheight, is the mountain top if the strings formed by appending nextletter to the strings in currentnode's class have a different frequency in the source; otherwise it is the left side of a mountain. Thus, if the frequency of the node reached by nextletter is different from the frequency of currentnode, then insert the frequency of currentnode into the cell at currentheight, otherwise insert a slash.

The only remaining problem is how to find topnode and topheight for the current column. Let *currentletter* be the letter at the bottom of the current column. Let *oldtopnode* and *oldtopheight* be the values of topnode and topheight in the preceding column (or the start node and zero if the current column is the first column of the landscape). Determine if there is an edge leading out from oldtopnode that is labeled with currentletter. If this edge exists and the node it leads to has a frequency at or above the minimum frequency to be printed in the landscape (2 for a landscape of a string with respect to itself), then topnode becomes this node. Otherwise repeat the process for successive suffix nodes, starting at oldtopnode, until a node with the minimum frequency is found. This will be the new topnode.

In the process, oldtopnode, or a node in its chain of suffix nodes, is found to be the *parent* of topnode. To compute topheight, we observe that the boundary mark for newtopnode's class is the diagonal upward extension of the boundary mark for its parent's class in the previous column. Thus, topheight is given by one plus oldtopheight, if its parent is oldtopnode, or one plus its parent's length label if it is not.

RESULTS

T7 examples

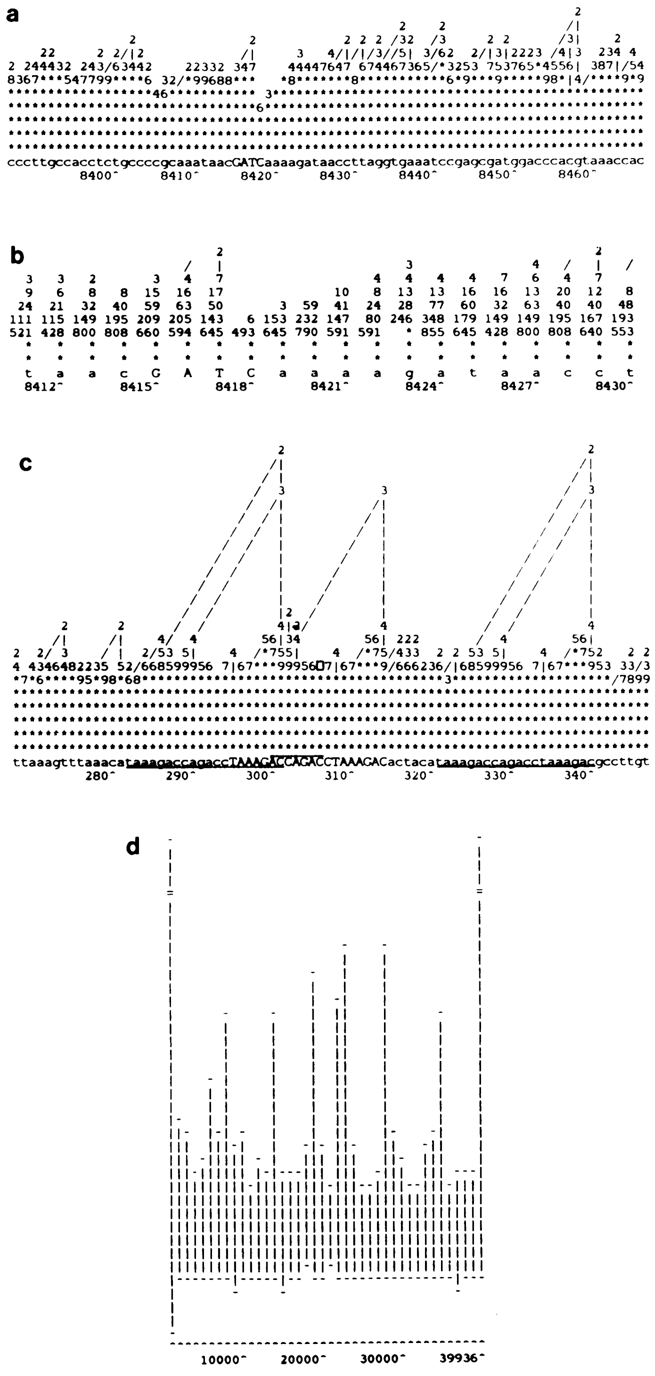
Figure 4 shows several landscapes from bacteriophage T7 (13). Figure 4a displays a distinctive valley directly above nucleotide 8418. The sequence GATC occurs only six times in T7, and the longer cGATC is unique, as indicated by the blank space for that sequence. In a random sequence the length of T7, a tetramer is expected to occur 156 times. The avoidance of the GATC sequence in T7 is particularly evident when the lower level numbers are revealed by a *zoom-in*, as in Figure 4b. From the frequencies of GAT (645), ATC (493) and AT (2260, under an asterisk in figure 4b), one would expect 141 GATC's. T7 has evolved to avoid the sequence GATC, which is methylated by the *E. coli* dam methylase (14). The reason for this avoidance is unknown.

Figure 4c shows some significant peaks from T7. Twenty levels above nucleotide 314 is a peak labeled with a 3 that signifies the twenty-long sequence beneath it, in capitals, occurs three times in T7. The other two occurrences are nearby, one on each side (underlined). Both of these are preceded by ACA, and so are part of a 23-long sequence that occurs twice. The blanks, as well as the vertical and diagonal lines, beneath the 3 of the peaks, also represent sequences that occur three times. The lowest level of those, at level seven and occurring between the 6 and 7 (the blank is boxed) is associated with the sequence ACCAGAC (overlined). This means that in T7 every time the sequence ACCAGAC occurs it is preceded by TAAAG and followed by CTAAAGAC. The twenty-long sequence *is the implication of* the seven-long sequence. These sequences include some of the SRL repeats seen by Dunn and Studier (13); their function is unknown.

Figure 4d shows the global view of the T7 landscape, in which the entire sequence is condensed into a single view. Each vertical line is a condensation of the landscape from 1000 bases. The top of the line indicates the maximum peak in the region, and the bottom of the line is the minimum valley in the same region. The peaks in this view of T7 are primarily late promoter sites, which are very similar to one another (13, 15). The three lowest valleys are GATC's (the other 3 GATC's are not visible because they are part of longer repeats, and so are "filled in"). The enormous peaks on the ends of the genome, which extend higher than the figure, represent the 160 base pair direct repeats (13).

fd Repeat Sequences

Figure 5a shows the global view of the bacteriophage fd landscape (6). There are two peaks of height 32, one at about 1900 and the other at about 2400 in the fd sequence. Figures 5b and 5c show the two regions in detail. There are actually two 32-long sequences that each occur twice; the


a
ccccttgccacctctgccccgcaaataacGATCaaaagataaccttaggtgaaatccgagcgatggacccacgtaaaccac
8400ˆ 8410ˆ 8420ˆ 8430ˆ 8440ˆ 8450ˆ 8460ˆ
b
t a a c G A T C a a a a g a t a a c c t
8412ˆ 8415ˆ 8418ˆ 8421ˆ 8424ˆ 8427ˆ 8430ˆ
c
ttaaagtttaaacataaagaccagaccTAAAGACCAGACCTAAAGACactacataaagaccagacctaaagacgccttgt
280ˆ 290ˆ 300ˆ 310ˆ 320ˆ 330ˆ 340ˆ
d
10000ˆ 20000ˆ 30000ˆ 39936ˆ

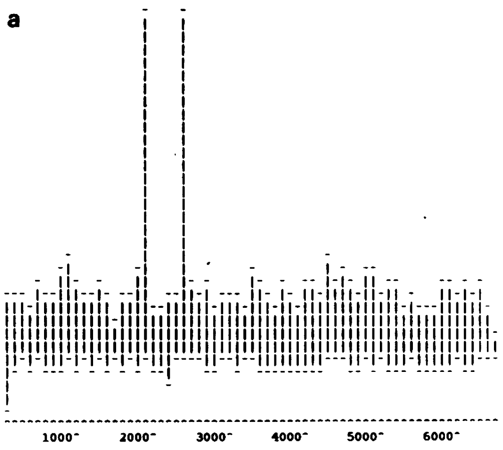
bases from 1843 to 1874 are repeated at 2329 to 2360 and the bases 2317 to 2348 are repeated at 2347 to 2378. The detail of the repeats can be seen easily in the landscape. The common core is a fourteen-long sequence that occurs 7 times (in capitals). In five of those seven cases it is followed by a T, and the other two times by a C. Three of the repeats are in the 1900 region and the other four are in the 2400 region. All seven of the fifteen-long sequences, ending in either the T or the C, are in the same reading frame of gene III, which codes for the phage capsid protein responsible for adsorption to the host (16, 17, 18). The amino-acid repeat is *ser-glu-gly-gly-gly*, and it occurs seven times in the protein (16). Figure 5d shows the landscape of the protein sequence. The detail of the amino-acid repeats is easily seen, including several partial repeats on the ends. The sequence EGGG occurs eight times, and the sequence GGGS occurs 11 times (the "*" on the same level as the 8's). These DNA repeats lead to amino-acid repeats suggesting that their function is at the protein level. Indeed, the closely related phage f1 and M13 differ from fd at 5 nucleotides in the repeat regions, but the proteins remain identical (16). However, there are 768 (6x2x4x4x4) different nucleotide sequences that would generate the same amino-acid repeat sequence, and only a few of them are used. For example, of the three glycine codons in each repeat, the first one almost always has a T in the third position, the second one usually has C and the third has 50% T and C. Thus it seems likely that either the DNA repeats were formed by recent duplication events or that they play an important role for the phage at the nucleotide sequence level.

G4 to ϕX174 Comparisons

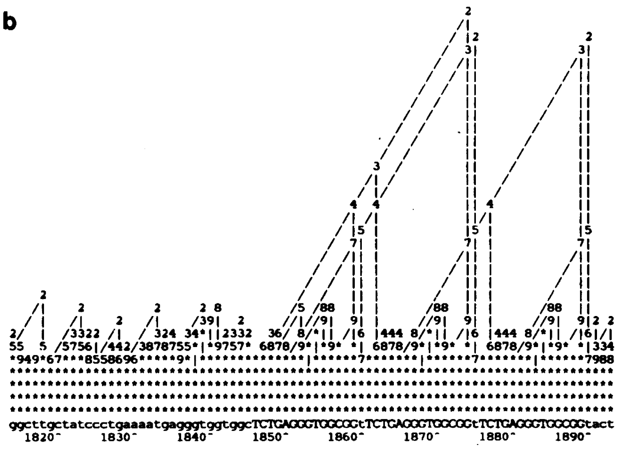
Figure 6a is the landscape of G4 (19), using the DAWG from the closely related phage ϕX174 (20). Peaks are now sequences from G4 that occur at least once in ϕX, and valleys are sequences from G4 that do not occur in ϕX. There are two 30-long, one 29-long and one 28-long sequences that occur in both G4 and ϕX. The global view shows that different regions of the two phages are conserved to different degrees. Figure 6b is the inverse of 6a: ϕX is the target sequence and the DAWG is from G4. The same peaks are observed because they are sequences that occur in each phage. The arrangement of the peaks looks, at first glance, to be different. However, these genomes are each circular and the sequences have been arbitrarily linearized at different places. Alignment of the two compressed landscapes gives the proper alignment of the two genomes. The first 30-level peak of G4 (from nucleotides 500 to 529) is the same sequence as the second 30-level peak of ϕX (from nucleotides 4299 to 4328). This is within the coding region for the gene A protein of each phage, and it also overlaps the beginning point of viral strand synthesis. It is perhaps the dual function of this sequence that accounts for its high conservation.

Figure 4. Some landscapes from bacteriophage T7.
a. Landscape around the GATC at position 8418. This string only occurs six times in T7. The "*" represent strings that occur more than 9 times.
b. The same region as in 4a expanded (by a "zoom-in") so that larger numbers are visible.
c. The landscape around position 300. See text for details.
d. The condensed landscape for all of T7.

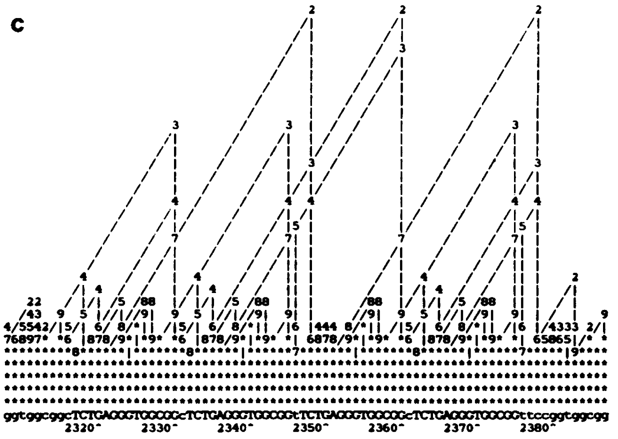
a


1000
2000
3000
4000
5000
6000

b


ggcttgctatccctgaaaatgagggtggtggcTCTGAGGGTGGCGGtTCTGAGGGTGGCGGtTCTGAGGGTGGCGGtact
1820
1830
1840
1850
1860
1870
1880
1890

c


ggtggcggcTCTGAGGGTGGCGGcTCTGAGGGTGGCGGtTCTGAGGGTGGCGGcTCTGAGGGTGGCGGttccggtggcgg
2320
2330
2340
2350
2360
2370
2380

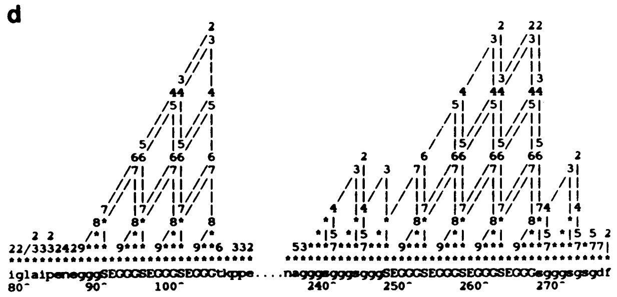


Figure 5. Repeat sequences in fd.
a. The condensed landscape for all of fd.
b and c. The detail from the regions of the two peaks in part a. See text for details.
d. The landscape for the gene III protein in the region of the repeating sequences. The DNA repeats are in the same reading frame, leading to repeats in the protein as well.

The other 30-level peak occurs at nucleotides 594 to 623 of ϕX and nucleotides 2180 to 2209 of G4. This region of thirty conserved bases is within the coding regions of both genes E and D, which are overlapping but in different reading frames. Both protein sequences are conserved in this region, gene E for nine amino acids and gene D for 15, leading to the high conservation at the DNA sequence level. The 29- and 28-long repeats are in the coding regions of genes A and A*; the 29-long one is also in the reading frame of the overlapping out-of-frame gene K.

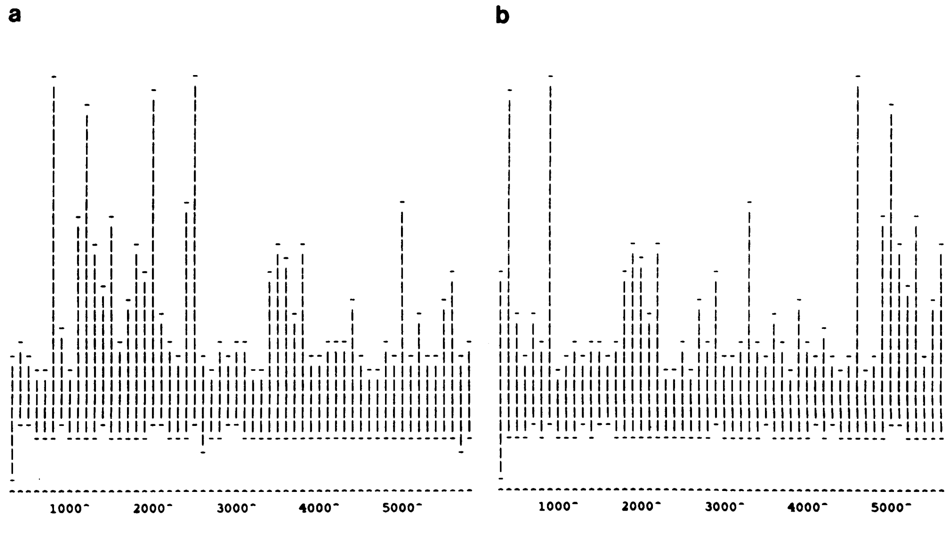


Figure 6. Comparisons of G4 to ϕX174.
a. Landscape of G4, using ϕX174 as the source string.
b. Landscape of ϕX174, using G4 as the source string.

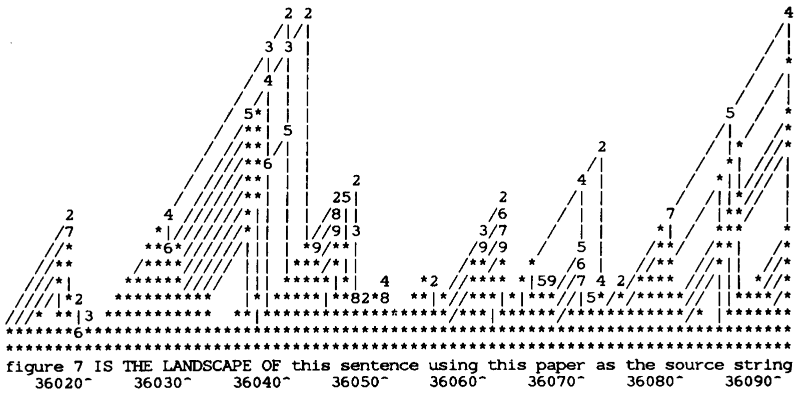


figure 7 IS THE LANDSCAPE OF this sentence using this paper as the source string
36020^ 36030^ 36040^ 36050^ 36060^ 36070^ 36080^ 36090^

Figure 7. Landscape of a sentence from this paper. All letters are considered lower case; otherwise the text is unmodified. The landscape is nearly the same if blanks and punctuation are removed. The capitalized text corresponds to the peak at 36040.

DISCUSSION

The directed acyclic word graph (DAWG) is a fast ($O(n)$) algorithm for determining the number of occurrences of every substring in a string. With the landscape display described in this paper, it provides a fast and powerful way to examine the patterns of nucleic-acid sequences, amino-acid sequences or other texts. Unusually common or rare sequences appear as peaks or valleys, respectively, in the landscape. In addition, subsequences that always occur in a particular context are easily seen.

Rules of the form "every time y occurs it is preceded by x and followed by z", which form the basis of the notion of implication, constitute a primitive "grammar" for a set of strings to the extent that they place restrictions on the allowable substrings of these strings. The notion of two-way equivalence classes, derived from these rules, leads to a unique decomposition of a string into a loose hierarchy of primary subunits. These subunits and their relationships are graphically displayed by the mountains of the landscape. Further, each subunit can be uniquely associated with a substring of the string, namely the representative of the class. These representatives are the largest substrings that occur in two or more distinct contexts in the string. We have observed that in DNA sequences the larger representatives often correspond to substrings with known biological functions, such as the T7 late promoters, and in human languages they often correspond to words and short phrases that identify key elements of the passage. Figure 7 is the landscape of this sentence using this paper as the source string. While we do not expect this notion of grammar to ever explain more than a small part of the structure of DNA or natural language strings, it could prove an interesting starting point. It has the advantages of being entirely universal and computationally tractable.

To be of further use in biology, this model should probably first be extended to allow a probabilistic notion of implication, which would systematically include the "near misses" as well. Even though the DAWG structure is highly sensitive to small changes in the source string, it is still useful for approximate string matching. Examples of near misses that still show up in landscapes because they contain large exact matches are plentiful. This idea can be formalized in a metric between strings that takes a prevalence of smaller exact matches between two strings as an indication of relatedness (regardless of the order in which the matches occur). Details are given in (21). This new metric may make fast phylogeny measurements possible.

We have shown in this paper a few of the uses of a sequence landscape. Others come to mind that we have not yet fully explored. A landscape of a sequence, using its complement for the DAWG, would be a fast way of finding significant palindromes and potential RNA structures. This may prove a useful first step in structure prediction of long sequences. A DAWG from a set of functionally related sequences could be used to show features that are most conserved even when the spacing between, or order of, important parts are variable. The DAWG created from a collection of functionally related sequences, such as promoters or ribosome binding sites, might do better than consensus searching approaches at identifying new sites. A significant density of short exact matches to a large collection of related sequences should be seen by appropriate analyses of the lower landscape levels. Any analysis of sequences that relies on patterns of characters can probably be augmented by use of a landscape, because of its speed of calculation and the wealth of information displayed.

ACKNOWLEDGEMENTS

B. Clift, D. Haussler and R. McConnell were supported by NSF grant IST-8317918. T. Schneider and G. Stormo were supported by NIH grant GM28755. The Pyramid 90x computer was purchased for the Department of MCD Biology by NIH grant RR01538. We would like to thank A. Ehrenfeucht, A. Blumer and J. Blumer for their help and criticism during the gradual evolution of this software over the past year and a half.

REFERENCES

1. Korn, L. J., Queen, C. L. and Wegman, M. N. (1977) Proc. Natl. Acad. Sci. USA 74, 4401-4405.
2. Wirth, K. (1976) Algorithms + Data Structures = Programs, Prentice-Hall, Englewood Cliffs, N. J.
3. Blumer, A., Blumer, J., Ehrenfeucht, A., Haussler, D., Chen, M. T., and Seiferas, J., "The smallest automaton recognizing the subwords of a text." Theoretical Computer Science, in press.
4. Blumer, A., Blumer, J., Ehrenfeucht, A., Haussler, D., McConnell, R. (1984) Proc. 16th ACM Symp. on Theoretical Comp. Sci., May, 349-358.
5. Fiers, W., Contreras, R., Duerinck, F., Haegeman, G., Iserentant, D., Merregaert, J., Min Jou, W., Molemans, F., Raeymaekers, A., Van Den Berghe, A., Volckaert, G. and Ysebaert, M. (1976) Nature 260, 500-507.
6. Beck, E., Sommer, R., Auerswald, E. A., Kurz, C., Zink, B., Osterburg, G., Schaller, H., Sugimoto, K., Sugisaki, H., Okamoto, T. and Takanami, M. (1978) Nucl. Acid. Res. 5, 4495-4503.
7. Schneider, T. D., Stormo, G. D., Haemer, J. S., and Gold, L. (1982) Nucl. Acid. Res. 10, 3013-3024.
8. Schneider, T. D., Stormo, G. D., Yarus, M. A., and Gold, L. (1984) Nucl. Acid. Res. 12, 129-140.
9. Armstrong, J. *et. al.*, Nucleotide Sequences (1985) IRL Press, Oxford.
10. Boyer, R. S. and Moore J.S. (1977) Comm. ACM 10, 762-772.
11. Bogart, K.P. Introductory Combinatorics, Pitman, Boston Mass., 1983.
12. Blumer, A., Ehrenfeucht, A. and Haussler, D. "Average sizes of suffix trees and DAWGs", in preparation.

13. Dunn, J. J. and Studier, F. W. (1983) J. Mol. Biol. 166, 477-535.
14. Geier, G. E. and Modrich, P. (1979) J. Biol. Chem. 254, 1408-1413.
15. Rosa, M. D. (1979) Cell 16, 815-825.
16. Beck, E. and Zink, B. (1981) Gene 16. 35-58.
17. Gray, C. W., Brown, R. S. and Marvin, D. A. (1981) J. Mol. Biol. 146, 621-627.
18. Smith, G. P. (1985) Science 228, 1315-1317.
19. Godson, G. N., Barrell, B. G., Staden, R. and Fiddes, J. C. (1978) Nature 276, 236-247.
20. Sanger, F., Coulson, A. R., Friedmann, T., Air, G. M., Barrell, B. G., Brown, N. L., Fiddes, J. C., Hutchison, C. A. III., Slocombe, P. M. and Smith, M. (1978) J. Mol. Biol. 125, 225-246.
21. Ehrenfeucht, A. and Haussler, D., "A new distance metric for strings," in preparation.