Discovering Sequence Similarity by the Algorithmic Significance Method

Aleksandar Milosavljević
Genome Structure Group
Biological and Medical Research Division
Argonne National Laboratory
Argonne, Illinois 60439-4833
e-mail: milosav@anl.gov

Abstract

The minimal-length encoding approach is applied to define concept of sequence similarity. A sequence is defined to be similar to another sequence or to a set of keywords if it can be encoded in a small number of bits by taking advantage of common subwords. Minimal-length encoding of a sequence is computed in linear time, using a data compression algorithm that is based on a dynamic programming strategy and the directed acyclic word graph data structure. No assumptions about common word ("k-tuple") length are made in advance, and common words of any length are considered. The newly proposed algorithmic significance method provides an exact upper bound on the probability that sequence similarity has occurred by chance, thus eliminating the need for any arbitrary choice of similarity thresholds. Preliminary experiments indicate that a small number of keywords can positively identify a DNA sequence, which is extremely relevant in the context of partial sequencing by hybridization.

1 Introduction

The search for sequence similarity based on subword composition is a common concept now. Pevzner [12] recently reviewed many different methods based on this concept. In the following, we present several improvements over current methods that are based on the minimal length encoding approach.

Subword similarity searching is typically based on words ("k-tuples") that do not exceed a specified length. The word length is restricted in order to limit the time needed for straightforward counting of occurrences of words or to limit the size of the Markov model for the sequence. (Both variables grow exponentially with the length of the words considered.) In this paper, we eliminate the assumption about fixed word length by applying the directed acyclic word graph [3] data structure and a linear-time data compression algorithm that employs a dynamic programming strategy.

The significance of sequence similarity is typically determined by more or less ad-hoc methods that are valid only under oversimplified assumptions about the distributional properties of the occurrences of words in sequences [13]. Indeed, the overall subword similarity of two sequences is a very complex statistic having distributional properties that are hard to determine, except in the most restricted cases. In this paper, we show that significance of sequence similarity can be rigorously determined under very few assumptions by applying the recently introduced algorithmic significance method [11].

Massive hybridization experiments that are performed as part of the partial sequencing by hybridization project [5] are producing information about presence of particular words within a huge number of cloned DNA sequences. A *partial* sequence consisting of a set of keywords can be used to recognize the similarity of clones to known DNA sequences without having to sequence the clones completely. In this paper we propose a method to discover sequences that may be similar not only to another complete sequence but also to a set of keywords that may come from a partially sequenced clone.

2 Minimal Encoding Length and Similarity

The main idea behind the approach presented in this paper is that similarity can be defined via minimal encoding length. We ask whether or not a particular sequence which we call the *target* can be concisely encoded using another sequence or a set of keywords, which we call the *source*. If there is no similarity, then the knowledge of the source does not help, but if we succeed in using the source to find an encoding of the target sequence that is much shorter than is likely by chance, then we can prove at a high significance level that the source and the target are related. For a broader introduction to minimal length encoding see [4].

A target sequence is defined to be similar to a source sequence if it can be encoded concisely by replacing some words in it by pointers to the occurrences of the same words in the source. This is a standard technique in data compression [14]. Consider an example where the target sequence is

```
GATTACCGATGAGCTAAT
```

and the source sequence is

```
ATTACATGAGCATAAT
```

The occurrences of some words in the target may be replaced by pointers indicating the beginning and the length of the occurrences of the same words in the source. In the following, a pointer is denoted by a pair of integers in parentheses, the first indicating the position of occurrence in the source and the second the length of the common word. For example,

```
G(1,4)CCG(6,6)(13,4)
```

One can think of the encoded sequence as being parsed into words that are replaced by pointers and into the letters that do not belong to such words. One may then represent the encoding of a sequence by inserting dashes to indicate the parsing. For the encoding above, the parsing is indicated as follows:

```
G-ATTA-C-C-G-ATGAGC-TAAT
```

Let us now count the exact number of bits needed to encode letters and pointers. We may assume that the encoding of a sequence consists of units, each of which corresponds either to a letter or to a pointer. Every unit contains a (log 5)-bit field that either indicates a letter or announces a pointer. A unit representing a pointer contains two additional fields with positive integers indicating the position and length of a word. These two integers do not exceed n, the length of the source sequence. Thus, a unit can be encoded in log 5 bits in case of a letter or in $\log 5 + 2 \log n$ bits in case of a pointer.

If it takes more bits to encode a pointer then to encode the word letter by letter, then it does not pay off to use the pointer. Thus, the encoding length of a pointer determines the minimum length of common words replaced by pointers. In order to take advantage of shorter common words, we must encode the pointers more concisely.

The pointers can be encoded more concisely under two plausible assumptions. The first assumption is that the common words occur in similar order in the target as in the source, in which case the position of the common word in the source can be indicated relative to the previous common word; this relative distance may fall into a smaller range than the absolute position and thus it may be represented in fewer bits. The second assumption is that the lengths of the common words fall into a smaller range. Under these two assumptions, one may encode a pointer in much less than $\log 5 + 2 \log n$ bits.

If a word to be replaced by a pointer occurs more than once in the source, then the information about the particular occurrence contained in the pointer may be more than is necessary. If the pointer could specify only the set of occurrences and not any particular occurrence, then the pointer itself would require fewer bits. We will come back to the problem of pointer size later in the experimental section.

Consider the case of a target sequence that is encoded using a set of keywords as the source. A sequence is in this case encoded using a pointer that consists of three numbers: an index of the keyword and the beginning and the end of the subword within the keyword.

Consider the following target sequence:

```
GATTACCGATGAGCTAAT
```

and the source keywords:

```
1 AAAAGGGGGG

2 ATTACATG

3 AGCATAAT

4 ATGAGCATA
```

The following is an encoding of the target sequence relative to the source keywords:

```
G(2,1,4)CCG(4,1,6)(3,5,4)
```

It is easy to see from these two examples that one can construct a decoding algorithm that within it contains a source sequence or source keywords and that reconstructs the target sequence from its encoding.

3 Algorithmic Significance Method

Let A denote a decoding algorithm that can reconstruct the target sequence. We may assume that the source (sequence or set of keywords) is part of the algorithm A so that only the encoded target needs to be supplied. We expect that the targets that are similar to the source will have short encodings. By $I_A(t)$ we denote the length of the encoded target t.

Let P_0 denote the null hypothesis, i.e., the distribution of probabilities under the assumption that the target isequence is independent from the source. Let $p_0(t)$ be the probability assigned to a target t by the null hypothesis. For example, if we assume that every letter is generated independently with probability p_x, where $x \in \{A, G, C, T\}$ denotes the letter, then the probability of a target sequence t is $p_0(t) = \prod_x p_x^{n_x(t)}$, where $n_x(t)$ is the number of occurrences of letter x in t. If we assume a Markov dependency, then the probability of a sequence can efficiently be computed for the given Markov model.

The following theorem states that a target sequence t is unlikely to have an encoding much shorter than $-\log p_0(t)$.

Theorem 1 *For any distribution of probabilities P_0 over sequences and for any decoding algorithm A,*

$$P\{-\log p_0(t) - I_A(t) \geq d\} \leq 2^{-d}$$

where $p_0(t)$ is the probability assigned to sequence t by distribution P_0 and $I_A(t)$ is the length of an encoding of sequence t using algorithm A.

Proof of Theorem 1 can be found in [11]. (Similar theorems are proven in the context of competitive encoding [4].)

This theorem enables us to use algorithm A as an *alternative hypothesis* to refute the null hypothesis P_0 at the significance level 2^{-d}. Applying the inequality above to our example, the probability that a target sequence t will have an encoding $d = 7$ bits less than $-\log p_0(t) = -\sum_x n_x \log p_x(t)$ is less than 2^{-7}, which is less than the standard significance threshold of 0.01.

The exponential relationship between encoding length and probability allows us to establish significance even when very large sequence libraries are searched. If the sequence library searched contains sequences of total length L, then to refute the null hypothesis at the significance level of 0.01 for *any* sequence in the library, $d = 7 + \log L$ bits would suffice (but may not be necessary).

The algorithmic significance method is conceptually very similar to the concept of statistical significance in the Neyman-Pearson hypothesis testing framework (see, e.g., [9]). The main difference is that the alternative hypothesis is now represented by a decoding algorithm instead by an explicit distribution of probabilities.

In contrast to the concept of statistical significance that is based on the approximate (asymptotic) estimation of tails of distributions ("p-value") of statistics like length of the longest common word or number of common words (see, e.g., [8]), the algorithmic significance method provides an exact significance value. Moreover, algorithmic significance is directly applicable for any null hypothesis for which $p_0(t)$ can be computed, as opposed to the statistical significance value which is applicable only for a specific distribution.

4 Minimal Length Encoding Algorithm

Short encodings are not only significant, but they can also be computed efficiently. In this section we present a minimal length encoding algorithm, which is a slight variant of the algorithm for discovering simple sequences [11] and which is frequently used in data compression [14].

The algorithm takes as an input a target sequence t and the encoding length $p \geq 1$ of a pointer and computes a minimal length encoding of t for a given source s (here s denotes either a source sequence or keywords). Since it is only the ratio between the pointer length and the encoding length of a letter that matters, we assume, without loss of generality, that the encoding length of a letter is 1.

Let n be the length of sequence t and let t_k denote the $(n-k+1)$-letter suffix of t that starts in the k^{th} position. Using a suffix notation, we can write t_1 instead of t. By $I(t_k)$ we denote the minimal encoding length of the suffix t_k. Finally, let $l(i)$, where $1 \leq i \leq n$, denote the length of the longest word that starts at the i^{th} position in target t and that also occurs in the source s. If the letter at position i does not occur in the source, then $l(i) = 0$. Using this notation, we may now state the main recurrence:

$$I(t_i) = min(1 + I(t_{i+1}), p + I(t_{i+l(i)}))$$

Proof of this recurrence can be found in [14].

Based on this recurrence, the minimal encoding length can now be computed in linear time by the following two-step algorithm. In the first step, the values $l(i)$, $1 \leq i \leq n$ are computed in linear time by using a directed acyclic word graph data structure that contains the source s [3]. In the second step, the minimal encoding length $I(t) = I(t_1)$ is computed in linear time in a right-to left pass using the recurrence above.

5 Implementation and Experiments

The algorithm above was implemented in C++ on a Sun Sparcstation under UNIX. The program was applied to identify occurrences of Alu sequences in the Human Tissue Plasminogen Activator (TPA) gene [6]. Three kinds of sources were used: the complete Alu consensus sequence [7] and keyword sets consisting of eight and four 8-mers chosen to be evenly distributed along the consensus.

The complete TPA gene, GenBank [2] accession number K03021, containing 36,594 bases was searched. The sequence was considered one window at a time, with windows of length 350 and an overlap between adjacent windows of 175. The value of $\log p_0(t)$ was computed under the assumption that letters are generated independently by a uniform distribution. An encoding length threshold of $22 \geq 7 + \log 36594$ bits was chosen so that the probability of *any* window having short encoding would be guaranteed not to exceed the value of 0.01.

The only difference in program input between the three experiments (except, of course, the source itself) was the pointer size. When the source was the complete Alu consensus sequence, a pointer length of 8 bits was chosen under the assumption that the distance between consecutive common words and common word length can each be encoded using 4 bits on average. In the case of keyword sets, the pointer size was chosen as the binary logarithm of the number of keywords plus 2 bits to locate a subword within a keyword. One may argue that more bits would be required to locate a subword; our choice of fewer bits can be justified by the fact that only the subwords that exceed a certain length need to be located, and also that short subwords are likely to have multiple occurrences, so that a pointer to any particular occurrence would contain excess information.

The windows that could be encoded in 22 bits less than postulated by the null hypothesis were merged to obtain a set of non-overlapping regions. Table 1 contains a comparison of occurrences of Alu and half-Alu sequences in the TPA gene in direct orientation with the regions identified as Alu-like. There were no false positive matches.

Figure 1 contains an example of a window parsed using a complete Alu consensus sequence as a source, together with an alignment of the Alu consensus with a segment within the window. Figure 2 contains the eight- and four-keyword sets used and a window parsed using the four-keyword set.

```
Parsed window 22226-22575 from the TPA gene:

C-T-CAGTG-C-C-T-G-TCAAAA-G-T-A-T-G-T-GCTGAGGC-T-G-G-A-A-G-GTGGTG-C-A-T
-GCCTGT-G-ATCCCAGCACTTT-A-GGAGGCC-A-A-G-G-TGGGAGG-G-TCGCT-G-G-A-G-CCCG
GGAG-T-T-C-A-AGACCA-A-T-CTGGGC-A-AACAT-A-G-C-A-A-G-T-C-C-C-C-T-GTCTCTA
C-A-AAAAATA-A-AAAAATTAGCC-AGACC-T-G-G-T-A-T-G-T-A-G-TCCCA-A-C-T-A-C-TT
GGGAGG-T-TGAGGCAG-A-A-GGATCAC-T-TGAGCC-CAGGAGTT-GGAGGCTG-C-A-GTAATC-T-
A-C-G-A-T-T-A-T-GCCACTGCA-T-T-T-C-A-A-C-C-T-CAGTGA-C-A-G-G-G-C-A-A-G-C
-C-C-TCACCT-CTAAAA-C-A-A-A-ACAAAA-CAACA-C-A-A-ACAAAAA-CAAAAA-C-ACAGA-A
-A-A-G-C-C-C

Alignment (Alu consensus is on top and a fragment from the parsed window
is on the bottom):

         @10        @20        @30        @40        @50        @60        @70
GGCCGGGCGCGGTGGCTCACGCCTGTAATCCCAGCACTTTGGGAGGCCGAGGCGGGCGGATCACCTGAGG
***:**  :.*****:.**:******:*************:*******:***:***.**:**:*:.***.
GGCTGG--AAGGTGGTGCATGCCTGTGATCCCAGCACTTTAGGAGGCCAAGGTGGGAGGGTCGCTGGAGC
 @30          @40        @50        @60        @70        @80        @90

         @80        @90         @100       @110       @120       @130
TCAGGAGTTCGAGACCAGCCT-GGCCAACATGGTGAAACCCC-GTCTCTACTAAAAATACAAAAATTAGC
:*:*******:******::** ***.*****:*::*:.**** ********.*******.**********
CCGGGAGTTCAAGACCAATCTGGGCAAACATAGCAAGTCCCCTGTCTCTACAAAAAATAAAAAAATTAGC
   @100       @110       @120       @130          @140       @150       @160

 @140       @150       @160       @170       @180       @190       @200
CGGGCGTGGTGGCGCGCGCCTGTAATCCCAGCTACTCGGGAGGCTGAGGCAGGAGAATCGCTTGAACCCG
*:*:*.****:          ****:*****:*****:******:********:**:***:*****:***:
CAGACCTGGTA---------TGTAGTCCCAACTACTTGGGAGGTTGAGGCAGAAGGATCACTTGAGCCCA
      @170                @180       @190           @200       @210

  @210       @220       @230       @240       @250       @260       @270
GGAGGCGGAGGTTGCAGTGAGCCGAGATCGCGCCACTGCACTCCAGCCTGGGCGACAGAGCGAGACTCCG
****.:*****:******:*.*::.***:::*********:*:**:***.:*:*****:**:**.*::*:
GGAGTTGGAGGCTGCAGTAATCTACGATTATGCCACTGCATTTCAACCTCAGTGACAGGGCAAGCCCTCA
@220       @230       @240       @250       @260       @270       @280

   @280
TCTCAAAAAAAA
:***.****.**
CCTCTAAAACAA
  @290
```

Figure 1: **Parsing of window** 22226 – 22575 **from the TPA gene.**

Alu occurrence	identified Alu regions		
	Alu consensus	Eight 8-mers	Four 8-mers
740...1033	526...1225	526...1225	701...1050
8864...9176	8576...9275	8751...9275	
10067...10365	9801...10500	9976...10500	
16794...17125, 17170...17466	16626...17675	16976...17500	16976...17325
18879...19178	18726...19425	18726...19250	18901...19250
20946...21259, 21280...21578	20826...21700	20826...21700	21176...21700
22253...22545	22051...22750	22226...22575	
25620...25911	25376...26075	25551...25900	
26524...26821, 26941...27239	26251...27475	26426...26926, 26927...27475	26426...26775
27880...28145	27651...28350	27826...28350	
28804...29100, 29297...29431	28526...29575	28701...29575	
32921...33220	32726...33425	32726...33425	32901...33250
34234...34525	33951...34650	34126...34650	

Table 1: **Occurrences of Alu sequences in the TPA gene and the identified Alu regions. Some rows correspond to pairs of Alu sequences that occur close together. There were no false positive matches.**

```
Alu keyword sets used in search:

Eight 8-mers: 1 GGCCGGGC  5 AAAAATTA          Four 8-mers: 1 GGCCGGGC
              2 AGCACTTT  6 TAATCCCA                       2 CTGAGGTC
              3 CTGAGGTC  7 GAATCGCT                       3 AAAAATTA
              4 ACATGGTG  8 GTGAGCCG                       4 GAATCGCT

Window 18901-19250 from the TPA gene parsed using the set of four
8-mers as a source:

T-G-T-AATC-T-C-A-G-C-A-C-T-T-T-GGG-A-GGCCG-AGGT-G-G-GAGG-ATCGCT-TGAG-C
-C-C-AGG-A-G-T-T-G-GAG-A-C-C-A-GTC-T-GGGC-AAT-A-T-A-G-TGAG-A-T-GCT-GTC
-T-C-T-A-C-A-AAAAATT-T-AAAAATTA-GCCGGG-T-G-T-A-C-T-A-G-T-A-T-G-C-A-C-C
TG-T-GGTC-C-C-A-GCT-A-C-T-C-A-G-GAGG-CTGAGG-CGGG-AGG-ATCGCT-T-A-A-G-T-
T-C-A-G-GAGGT-T-GGG-A-C-T-T-C-A-G-TGAG-A-T-A-T-G-ATTA-CGC-C-AAT-G-C-A-
C-T-C-C-A-GCC-T-GGG-TGA-C-AAA-CTGAG-ATC-CTG-T-C-T-C-AAAAA-AAAAA-AAAAA-
AAAA-G-AAAAA-GGC-AAT-G-C-A-A-G-T-GCC-A-TGAG-C-C-A-C-A-G-C-A-GAGG-C-AAA
-CTG-C-C-C-T-C-A-T-GGT-GCC-T-A-C-T-C-T-C-T-A-G-CTGA
```

Figure 2: **Alu keyword sets used in the search and an example of a window parsed using the list of four keywords.**

6 Discussion

The experiments indicate that a surprisingly small number of keywords suffice to identify Alu sequences. The exact number of necessary keywords may depend on the degree of similarity between sequences of a particular class, but it seems that 10 or less 8-mers would suffice to identify a similarity of 80% in sequences about 300 long. This means that the sequence similarity of many cloned DNA sequences to known sequences may be determined without the exact knowledge of their complete sequence; a partial sequence consisting of a set of keywords that have been identified in the clone may suffice [10] [5].

The method described in this paper has also been succesfully used for rapid screening for sequence similarity by the Pythia email server for identification of Human repetitive DNA (for more information about Pythia, send "help" in Subject-line to Internet address pythia@anl.gov).

Current methods typically require two arbitrary assumptions to be made for each similarity search: one about the length of the longest common word that is to be considered and the other about the threshold of similarity for significant matches. At the same time, the exact significance of the match is not computed. The method proposed in this paper removes the need for any restrictions on word length while keeping the computation time linear, and it also provides an exact bound on significance, thus removing need for any arbitrary thresholds. Experiments indicate that this systematic approach can eliminate false positive matches while retaining sensitivity.

The algorithmic significance method can also be applied to discover similarity based on sequence alignment. For this case, the target sequence may have to be encoded using a set of edit operations. A minimal length encoding approach to sequence alignment has been discussed in [1]; the coding techniques presented there can be combined in conjunction with the algorithmic significance method to obtain exact bounds on significance.

The applications mentioned in this paper are just a small sample of possible applications of the concept of algorithmic significance. The key feature of the concept is its applicability in combinatorial domains like DNA sequence analysis, in contrast to the more standard statistical approaches which have mostly been motivated by real-valued measurements. Instead of focusing on a particular parameter (e.g., length of the longest repeated word), the algorithmic significance approach enables us to focus on the information content (as measured by the encoding length), which is a much more widely applicable parameter.

7 Acknowledgements

Discussions with Radoje Drmanac, Ivan Labat, Radomir Crkvenjakov, Jerzy Jurka, and Thomas Cover have greatly contributed to this work.

This work was supported in part by U.S. Department of Energy, Office of Health and Environmental Research, under Contract W-31-109-Eng-38 and in part by U.S. Department of Energy grant DE-FG03-91ER61152.

References

[1] L. Allison and C.N. Yee. Minimum message length encoding and the comparison of macromolecules. *Bulletin of Mathematical Biology*, 52:431–453, 1990.

[2] H.S. Bilofsky and C. Burks. The GenBank (R) genetic sequence data bank. *Nucleic Acids Research*, 16:1861–1864, 1988.

[3] A. Blumer, J. Blumer, D. Haussler, A. Ehrenfeucht, M.T. Chen, and J. Seiferas. The smallest automaton recognizing the subwords of a text. *Theoretical Computer Science*, 40:31–55, 1985.

[4] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[5] R. Drmanac, G. Lennon, S. Drmanac, I. Labat, R. Crkvenjakov, and H. Lehrach. Partial sequencing by hybridization: Concept and applications in genome analysis. In *In: The First International Conference on Electrophoresis, Supercomputing and the Human Genome*, pages 60–74. World Scientific, Singapore, 1991.

[6] S.J. Friezner-Degen, B. Rajput, and E. Reich. The human tissue plasminogen activator gene. *Journal of Biological Chemistry*, 261:6972–6985, 1986.

[7] J. Jurka and A. Milosavljević. Reconstruction and analysis of human Alu genes. *Journal of Molecular Evolution*, 32:105–121, 1991.

[8] S. Karlin, F. Ost, and B.E. Blaisdell. *Mathematical Methods for DNA Sequences*, chapter Patterns in DNA and Amino Acid Sequences and their Statistical Significance. CRC Press, Boca Raton, Florida, 1989.

[9] J.C. Kiefer. *Introduction to Statistical Inference.* Springer-Verlag, 1987.

[10] G.G. Lennon and H. Lehrach. Hybridization analyses of arrayed cDNA libraries. *Trends in Genetics*, 7(10):314–317, 1991.

[11] A. Milosavljević and J. Jurka. Discovering simple DNA sequences by the algorithmic significance method. *Computer Applications in Biosciences. in press.*

[12] P.A. Pevzner. Satistical distance between texts and filtration methods in sequence comparison. *Computer Applications in Biosciences*, 8(2):121–127, 1992.

[13] P.A. Pevzner, M.Y. Borodovsky, and A.A. Mironov. Linguistics of nucleotide sequences I: The significance of deviations from mean statistical characteristics and prediction of the frequencies of occurrence of words. *Journal of Biomolecular Structure and Dynamics*, 6:1013–1026, 1989.

[14] J.A. Storer. *Data Compression: Methods and Theory.* Computer Science Press, 1988.