Machine Learning, 21, 35–50 (1995)

Discovering Dependencies via Algorithmic Mutual Information: A Case Study in DNA Sequence Comparisons

ALEKSANDAR MILOSAVLJEVIĆ[*] milosav@anl.gov
Genome Structure Group
Center for Mechanistic Biology and Biotechnology
Argonne National Laboratory
Argonne, Illinois 60439-4833

Editors: David Searls, Jude Shavlik, and Lawrence Hunter

Abstract. Algorithmic mutual information is a central concept in algorithmic information theory and may be measured as the difference between independent and joint minimal encoding lengths of objects; it is also a central concept in Chaitin's fascinating mathematical definition of life. We explore applicability of algorithmic mutual information as a tool for discovering dependencies in biology. In order to determine significance of discovered dependencies, we extend the newly proposed algorithmic significance method. The main theorem of the extended method states that d bits of algorithmic mutual information imply dependency at the significance level $2^{-d+O(1)}$. We apply a heuristic version of the method to one of the main problems in DNA and protein sequence comparisons: the problem of deciding whether observed similarity between sequences should be explained by their relatedness or by the mere presence of some shared internal structure, e.g., shared internal repetitive patterns. We take advantage of the fact that mutual information factors out sequence similarity that is due to shared internal structure and thus enables discovery of truly related sequences. In addition to providing a general framework for sequence comparisons, we also propose an efficient way to compare sequences based on their subword composition that does not require any a priori assumptions about k-tuple length.

Keywords: Minimal length encoding, DNA sequence analysis, Machine discovery, Algorithmic mutual information, Algorithmic significance

1. Introduction

Algorithmic mutual information is a central concept in Chaitin's fascinating mathematical definition of life (Chaitin, 1979). The definition indicates that this abstract concept from algorithmic information theory (Chaitin, 1987b; Li & Vitányi, 1993) captures the deepest properties of the structure of biological knowledge. In this paper we explore applicability of algorithmic mutual information as a tool for discovery in biology. In order to determine significance of algorithmic mutual information, we extend the newly proposed algorithmic significance method. The main theorem of the extended method states that d bits of algorithmic mutual information imply dependence at the significance level $2^{-d+O(1)}$. We apply the method to one of the main problems in DNA and protein sequence comparisons: the problem of deciding whether observed similarity between sequences should be explained by their relatedness or by the mere presence of some shared internal structure, e.g., shared internal repetitive patterns. In addition, we employ

[*] Current address: CuraGen Corporation, 322 East Main Street, Branford, CT 06405

Chaitin's mathematical definition of life to demonstrate that the same method can be applied to discover large-scale dependencies that are characteristic for living systems.

It is becoming increasingly apparent that many DNA sequences exhibit internal structure: either a simple bias in sequence composition, or repetitions of certain words within the sequence. (While in this paper we focus on DNA sequences, this statement holds for amino acid sequences of proteins as well.) A common internal structure may cause two sequences to appear similar even though they are not related: for example, two DNA sequences that contain many repetitions of a tetramer $TCAG$ may appear similar, even though independent multiplication of the tetramer may be a preferred explanation for their similarity.

To avoid the spurious similarities, "masking" procedures are proposed (Claverie & States, 1993; Wootton & Federhen, 1993; Altschul, Boguski, Gish, & Wootton, 1994). These procedures simply eliminate sequences of lower complexity from comparisons. The main drawback of these methods is that they cannot discover related sequences of lower complexity, even though the sequences themselves frequently carry enough information about their relatedness.

The problem can clearly be phrased in terms of the parsimony (Occam's razor) principle: is it more parsimonious to explain similarity of two sequences by postulating relatedness or independence? To formulate the question more precisely, we measure parsimony of the two competing hypotheses in terms of encoding lengths. We define two sequences, one *target* t and the other *source* s. Let $I(t)$ denote the number of bits needed to encode t by taking advantage of its internal structure (as in Milosavljević & Jurka, 1993a) and let $I(t|s)$ be the number of bits needed to encode t relative to s by taking advantage of their mutual similarity (as in Milosavljević, 1993). The difference $I(t;s) = I(t) - I(t|s)$ between the first and the second encoding length is an approximation of the algorithmic mutual information between the two sequences. Any internal structure would lead to the decrease of $I(t)$, and, if the structure is also present in s, to the decrease of $I(t|s)$ as well. Thus, any shared internal structure would not affect $I(t;s)$ and its contribution to the similarity between the sequences would be factored out. In addition to this desirable property, we will show that d bits of mutual information imply dependency between individual objects at the significance level of $2^{-d+O(1)}$. This general method of proving dependencies represents an extension of the recently proposed algorithmic significance method (Milosavljević & Jurka, 1993a).

In the following we present in parallel both the general method and its practical application to DNA sequence comparisons. The practical application will be used both as a motivation and as a test-case for the general method. We start with an encoding method for DNA sequences.

2. Encoding Length and Similarity

In this section we review methods for concisely encoding sequences by taking advantage of repeated subwords. A target sequence t can be encoded in $I(t|s)$ bits by replacing some words in it by pointers to the occurrences of the same words in the source sequence s. This is a standard technique in data compression (Storer, 1988). 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 example,

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

To calculate the exact number of bits needed to encode letters and pointers, we 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 (throughout the paper, logarithms are base 2). 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 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.

Pointers can be encoded more concisely under two plausible assumptions. The first assumption is that the common words occur in similar order in both the target and in the source, in which case the position of a 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. The pointer could specify only the set of occurrences and not any particular occurrence, and thus the pointer itself would require fewer bits.

So far we have discussed only encoding of t relative to s and the number of bits $I(t|s)$. A sequence t can be similarly self-encoded in $I(t)$ bits by replacing a repeated occurrence of a word by a pointer to its previous occurrence within the same sequence. If there is enough repetition within a sequence, $I(t)$ will be small. We omit the details here because they have been discussed elsewhere (Milosavljević & Jurka, 1993a).

3. Extended Algorithmic Significance Method

In the following we extend the recently proposed algorithmic significance method (Milosavljević & Jurka, 1993a) by showing a high level of mutual information is unlikely to occur by chance. The method itself is very general and is also applicable to a wide variety of problems that may not be related to sequence analysis, as discussed in Section 6. The theorems presented below are applicable not only to sequences of finite length, but also to objects from any other countable domain. The particular problem of sequence comparison is here used as a motivation for the development of theorems that form the basis of the general method. The derivations below require some background in information theory (e.g., Cover & Thomas, 1991; Li & Vitányi, 1993); if you wish to avoid technical details, you may skip to Theorem 2.

In the following derivations we start from the fact that high likelihood ratios are unlikely to occur by chance and then we switch from probabilities to encoding lengths to show that high algorithmic mutual information is unlikely to occur by chance as well.

Let P_0 and P_A be probability distributions over sequences (or any other kinds of objects from a countable domain) that correspond to the null and alternative hypotheses respectively; by $p_0(t)$ and $p_A(t)$ we denote the probabilities assigned to a sequence t by the respective distributions. The likelihood ratio for sequence t is $\frac{p_A(t)}{p_0(t)}$. The following elementary inequality states that high likelihood ratios are unlikely to occur by chance

LEMMA 1 *For any null hypothesis P_0 such that $p_0(t) > 0$ for every t and for every alternative hypothesis P_A,*

$$P_0\{\log \frac{p_A(t)}{p_0(t)} \geq d\} \leq 2^{-d}$$

Proof: Lemma 1 is a direct consequence of the Markov inequality applied to the likelihood ratio $\frac{p_A(t)}{p_0(t)}$. Since the expected value $E_0[\frac{p_A(t)}{p_0(t)}]$ by the null hypothesis equals 1, by Markov inequality,

$$P_0\{\frac{p_A(t)}{p_0(t)} \geq c\} \leq \frac{1}{c}.$$

After taking logarithms,

$$P_0\{\log \frac{p_A(t)}{p_0(t)} \geq d\} \leq 2^{-d},$$

where $d = \log c$. ■

In our specific application, the null hypothesis P_0 will be the distribution of probabilities under the assumption that the target sequence is independent from the source. 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. The alternative hypothesis P_A will be the distribution under assumption that the sequences are related.

We now define the alternative hypothesis P_A in terms of encoding length. Let A denote a decoding algorithm that can reconstruct the target t based on its encoding relative to the source s. By $I_A(t|s)$ we denote the length of the encoding. We make the standard assumption that encodings are prefix-free, i.e., that none of the encodings represented in binary is a prefix of another (for a detailed discussion of the prefix-free property, see Cover & Thomas, 1991; Li & Vitányi, 1993). We expect that the targets that are similar to the source will have short encodings. The following theorem states that a target t is unlikely to have an encoding much shorter than $-\log p_0(t)$.

THEOREM 1 *For any distribution of probabilities P_0, decoding algorithm A, and source s,*

$$P_0\{-\log p_0(t) - I_A(t|s) \geq d\} \leq 2^{-d}$$

Proof: Since algorithm A specifies a uniquely decodable code, by Kraft-McMillan inequality, $\sum_t 2^{-I_A(t|s)} \leq 1$. Thus, there is a normalizing constant $b \geq 1$ such that $\sum_t b\,2^{-I_A(t|s)} = 1$, and we can now define the alternative hypothesis as the distribution P_A that assigns probability $p_A(t|s) = b\,2^{-I_A(t|s)}$ to target t. By substituting $b\,2^{-I_A(t|s)}$ for $p_A(t|s)$ in Lemma 1 we obtain

$$P_0\{-\log p_0(t) - I_A(t|s) + \log b \geq d\} \leq 2^{-d}.$$

Finally, since $\log b \geq 0$, we obtain

$$P_0\{-\log p_0(t) - I_A(t|s) \geq d\} \leq 2^{-d}.$$ ■

Similar theorems have been proven in the context of competitive encoding (Cover & Thomas, 1991) and testing theory (for a review of testing theory, see Li & Vitányi, 1993). The theorem is the basis for the algorithmic significance method where·presence of patterns is proven by exhibiting significantly shorter encodings of the observed data than expected by the null hypothesis; the method has been applied to discover simple DNA sequences (Milosavljević & Jurka, 1993a).

Invariance theorem (for a review see Li & Vitányi, 1993) states that there exists a universal encoding method that gives encodings that are as short as the encodings produced by any other method, up to an additive constant. The decoder for the universal method is a universal prefix-free Turing machine: the shortest encoding is the shortest program for the machine that outputs target t; in case of relative encoding, the machine may also have access to a source s. We now assume that A is one such universal machine and that $I_A(t|s)$ is the length of the shortest program.

A universal encoding method can also be used to define a null hypothesis. Let A_0 denote a universal machine and let $|p|$ denote the length of a program p. Halting probability Ω (for a detailed study of Ω, see Chaitin, 1987a) is the probability that A_0 halts when p is constructed bit by bit by random flips of a coin. That is,

$$\Omega = \sum_{p:\ A_0\ halts\ on\ p} 2^{-|p|}$$

The probability $p_{A_0}(t)$ that a halting program outputs t is computed as follows:

$$p_{A_0}(t) = \frac{1}{\Omega} \sum_{p:\ A_0(p)=t} 2^{-|p|}$$

Probability distribution P_{A_0}, discovered by Solomonoff (for a review of history, see (Li & Vitányi, 1993)), has a remarkable property: it cannot be refuted at an arbitrary significance level by any other computable distribution. However, since our alternative hypothesis is conditional on s, we still have a chance. We now assume that $P_0 = P_{A_0}$. The universal coding theorem (Li & Vitányi, 1993) tells us that $-\log p_{A_0}(t) = I_{A_0}(t) + O(1)$, where $I_{A_0}(t)$ denotes the length of the shortest program for A_0 that outputs t. By substituting $I_{A_0}(t)$ for $-\log p_0(t)$ in Theorem 1, and by moving the additive constant $O(1)$ into the exponent on the right-hand side, we obtain the following:

$$P_0\{I_{A_0}(t) - I_A(t|s) \geq d\} \leq 2^{-d+O(1)}.$$

Algorithmic mutual information $I(s;t)$ is defined as the difference $I_{A_0}(t) - I_A(t|s)$, so that the inequality above can now be rewritten in the following compact form:

THEOREM 2

$$P_0\{I(s;t) \geq d\} \leq 2^{-d+O(1)}.$$

This theorem is the basis for the *extended* algorithmic significance method, which enables discovery of significant dependencies in observed data via algorithmic mutual information. This is an ultimate, albeit impractical, method for deciding relatedness of two sequences: algorithmic mutual information $I(s;t) = I_{A_0}(t) - I_A(t|s)$ takes into account both sequence complexity, measured by $I_{A_0}(t)$, and similarity, measured by $I_A(t|s)$.

In order to make this method practical, we need to apply encoding schemes for which encoding lengths are easy to compute. Thus, we approximately estimate the universal encoding lengths $I_{A_0}(t)$ and $I_A(t|s)$ by $I(t)$ and $I(t|s)$, which are the encoding lengths obtained by applying the self-encoding and relative-encoding schemes from the previous section.

The introduction of specific encoding schemes introduces certain bias in the process of inference (which is absent in case of the universal, albeit non-computable, encoding scheme). This bias may in principle be also expressed in terms of a probability distribution. Since the encoding schemes presented in the previous section capture the kind of patterns that indeed occur in the data, they may be thought of as adequate approximations of the encoding by universal machines. In the next section we present an efficient encoding algorithm for our specific encoding scheme.

4. Minimal Length Encoding Algorithms

Not surprisingly, the algorithms for computing minimal encoding lengths $I(t)$ and $I(t|s)$ are very similar. A standard algorithm for computing $I(t)$ has already been presented elsewhere (Milosavljević & Jurka, 1993a). In this section we present a slight variant of the same algorithm that can be used for computing $I(t|s)$. The only difference between the two algorithms is that in the former pointers point to the occurrences of words within the same sequence while in the latter they point to the occurrences of words in the source sequence.

The algorithm for computing $I(t|s)$ takes as an input a target sequence t, a sоurce sequence s, and the encoding length $p \geq 1$ of a pointer. Since it is only the ratio between the pointer length and the encoding length of a letter that matters, we linearly scale the two values so that the encoding length of a letter becomes 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 $I(t_k|s)$ 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:

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

Proof of this recurrence can be found in (Storer, 1988).

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 (Blumer, Blumer, Haussler, Ehrenfeucht, Chen, & Seiferas, 1985). In the second step, the minimal encoding length $I(t|s) = I(t_1|s)$ is computed in linear time in a right-to left pass using the recurrence above.

5. Experiments

The algorithm for pairwise comparisons using mutual information $I(s;t) = I(s) - I(s|t)$ was implemented in C++ on a Sun Sparcstation under UNIX as part of a larger suite of programs for analysis of repetitive DNA sequences (Milosavljević, to appear) (to obtain information about the availability of the programs, send "software" in Subject-line to pythia@anl.gov). We present two experiments that illustrate the potential of the method.

5.1. Experiment 1

In the first experiment, the program was applied to identify occurrences of repetitive patterns in a 4-kbp segment of the human tissue plasminogen activator (TPA) gene (Friezner-Degen, Rajput, & Reich, 1986). The segment $22,001$-$26,000$ that was extracted from the GenBank (Bilofsky & Burks, 1988) entry under accession number K03021, is illustrated in Figure 1. The segment was split into consecutive windows of length 200 with an overlap of 100 basepairs. Every pair of nonoverlapping windows was compared using mutual information $I(s;t)$ in order to identify pairs of windows that contain related sequences.

An encoding length threshold of $31 \geq 7 + 2 * \log 4000$ bits was chosen so that the probability of *any* pair of windows having mutual information beyond the threshold would be guaranteed not to exceed the value of 0.01; one should note that the additive constant from Theorem 2 has been ignored in this calculation, so that the significance value has mostly a heuristic value. A pointer length of 6 bits was chosen for self-encoding and of 12 bits for encoding one sequence relative to the other (that is, it was assumed that that the distance between consecutive common words and common word length can each be encoded in 3 bits on average in case of self-encoding and in 6 bits in case of relative encoding).

As indicated in Figure 1, the segment was known to contain occurrences of two Alu sequences, one between positions 253 and 545 and the other between positions 3620 and 3911, as well as an imperfect $(TGATAGA) * N$ run between positions 1888 and 2458. The idea was to show that the windows containing the two occurrences of Alus would be identified while the windows containing different parts of the long $(TGATAGA) * N$ segment would not be considered similar because of their internal structure and despite their mutual similarity in terms of subword composition.

The following three pairs of windows exhibited mutual information above the threshold:

pair	I(s;t)	window 1	window 2
1	51	201-400	3601-3800
2	37	1901-2100	2201-2400
3	32	301-500	3601-3800

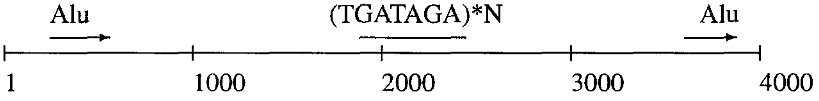

Figure 1. Segment $22,001 - 26,000$ from the TPA gene.

```
Self-parsed window 201-400 from the 4-kbp fragment of the TPA gene:

G-T-G-C-A-A-C-A-G-T-G-G-C-A-G-G-G-C-ACAGTG-C-C-A-C-T-CAGTGCC-T-G-T-C-A
-A-A-A-G-T-A-T-GTGC-T-G-A-G-GCTG-G-A-A-G-G-T-G-GTGCA-TGCCTGT-G-A-T-C-C
-C-A-GCAC-T-T-T-A-G-GAGGC-C-AAGGTGG-GAGG-G-T-C-GCTGGA-G-C-C-C-GGGAG-T-
TCAA-G-A-CCAA-T-C-T-GGGCA-AACA-T-AGCA-A-G-T-C-C-CCTGTC-T-C-T-A-CAAAA-A
-A-T-A-AAAAAAT-TAGC-C-AGACC-T

Local alignment of windows 201-400 (top) and 3601-3800 (bottom):

    @60        @70      @80      @90      @100     @110     @120
GGCTGG--AAGGTGGTGCATGCCTGTGATCCCAGCACTTTAGGAGGCCAAGGTGGGAGGGTCGCTGGAGC
******  :.*****:.**:**   **:*************:***:*******::**.**:**:*:.***.
GGCTGGGCGTGGTGGCTCACGC--GTAATCCCAGCACTTTGGGAAGCCAAGGCAGGTGGATCACCTGAGG
        @30      @40        @50       @60      @70      @80

      @130     @140     @150     @160     @170     @180     @190
CCGGGAGTTCAAGACCAATCTGGGCAAACATAGCAAGTCCCCTGTCTCTACAAAAAATAAAAAAATTAGC
:*:**************::** ***.*****:*::*: .************.*******.**********
TCAGGAGTTCAAGACCAGCCT-GGCCAACATGGTGAA-ACCCTGTCTCTACTAAAAATACAAAAATTAGC
 @90      @100      @110     @120      @130     @140      @150

      @200
CAGACCT
***:*.*
CAGGCAT
   @160
```

Figure 2. Parsing of window 201-400 and its local alignment with window 3601-3800.

Pairs 1 and 3 correspond to the two occurrences of Alu sequences. Figure 2 contains the self-parsing of the window 201-400, exhibiting the internal structure of the Alu sequence, as well as a local alignment with the window 3601-3800.

Pair 2 consists of two windows within the $(TGATAGA) * N$ region. Figure 3 contains the self-parsing of the window 1901-2100, exhibiting the internal structure of the $(TGATAGA) * N$ sequence, as well as a local alignment with the window 2201-2400.

The local alignment indicates that the two windows indeed share more structure than merely due to the presence of the $(TGATAGA) * N$ internal repeat: note that if we denote TGA by x and $TAGA$ by y, then the segment between positions 80 and 115 in window 1901-2100 and the segment between positions 70 and 104 in window 2201-2400 can both be approximately represented as $AAAyxyyxyyxTAAA$. This indicates that, in addition to the simple multiplication of the $TGATAGA$ repeat, larger units of DNA have multiplied as well, increasing the mutual information beyond the threshold. Indeed, a closer inspection of other segments of DNA within the $(TGATAGA) * N$ region indicates that the same large unit occurs in a slightly more decayed form in several copies, which were not detected.

5.2. *Experiment 2*

In the second experiment, the program was applied to identify related sequences between the TPA segment from the previous subsection and the segment 11,001-15,000 of the human C-FMS proto-oncogene for CSF-1 receptor (PO) gene, GenBank (Bilofsky & Burks, 1988) entry under accession number X14720. The PO segment, illustrated in Figure 4, was split into consecutive windows of length 200 with an overlap of 100 basepairs. Every window was compared using mutual information $I(s;t)$ to all the windows from the TPA segment (Figure 1) in order to identify related sequences between the PO and TPA segments. The thresholds and pointer lengths were set the same way as in the previous experiment.

As indicated in Figure 4, the segment was known to contain an Alu fragment between positions 2861 and 3016 as well as a $(TAGA) * N$ run between positions 938 and 1012. It was expected that the Alu fragment would be identified as similar with the Alu sequences from the TPA segment, while the $(TAGA) * N$ would not be identified as similar to the $(TGATAGA) * N$ region, because their overall similarity is exclusively due to the similarity in their internal structure.

The output of the program fully met the expectations. The only pairs of windows with similarity scores above the significance threshold were the ones corresponding to the Alu regions. Because of their internal structure, and despite their similar subword composition, the $(TGATAGA) * N$ and $(TAGA) * N$ regions were not considered similar. Figure 5 contains the self-parsing of the window 901-1100 from the PO segment

```
Self-parsed window 1901-2100 from the 4-kbp fragment of the TPA gene:

A-T-A-G-A-T-G-ATAGA-C-AGAT-A-ATAGATGATAG-G-T-G-ATAGATGATAGA-T-TGATAGAT
GATAGAT-GATAGGTGATAGAT-TAGAT-A-AATAGATGATA-C-ATACAT-GATAGAT-AGATGATA-A
ATAGA-C-G-G-TAGATG-GATGA-C-AGATAGA-C-AGATGATAGGTGATAGAT-AGATGATAGATTGA
TAGATGAT-T-G-A-TAGATAAATAGATGA1

Local alignment of windows 1901-2100 (top) and 2201-2400 (bottom):

         @80       @90       @100      @110      @120      @130      @140
AGATTAGATAAATAGATGATACATACATGATAGATAGATGATAAATAGACGGTAGATGGATGACAGATAG
****.* .*****************.********** ************:*:*****            *
AGATGA-TTAAATAGATGATACATAGATGATAGATA-ATGATAAATAGATGATAGAT------------G
         @70       @80       @90        @100      @110
```

Figure 3. Parsing of window 1901-2100 and its local alignment with window 2201-2400.

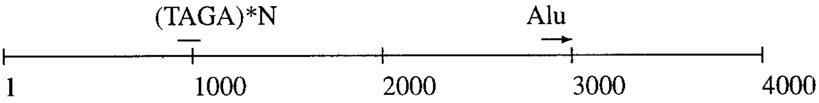

Figure 4. Segment 11, 001 – 15, 000 from the PO gene.

```
Self-parsed window 901-1100 from the 4-kbp fragment of the PO gene:

T-C-C-T-A-C-C-T-G-T-A-A-A-A-T-G-A-A-G-A-T-A-T-T-A-A-C-A-GTAA-C-T-G-C-C
-T-T-C-A-T-AGATA-GAAGATA-GATAGAT-TAGATAGAT-AGATAGATAGATAGATAGATAGATAGA
TAGATAGATAGATAG-GAAG-T-A-C-TTAGA-ACAG-G-G-T-C-T-G-A-C-ACAGG-AAATG-CTGT
-C-C-AAGT-G-T-G-C-A-C-CAGGA-GATAG-T-A-TCTGA-GAAG-G-C-T-C-A-GTCTG-GCACC
A-T

Local alignment of windows 901-1100 PO (top) and 2001-2200 TPA (bottom):

@10       @20         @30       @40        @50       @60       @70
TAAAATGAAGAT--ATTAACAGTAACTGCCTTCATAGATAGA-AGATAGATAGATTAGATAGATAGATAG
**:*..**.***  **.:**:***:.** ..*.*:******* **** *****:*  *********** *
TAGATAGATGATAAATAGACGGTAGATG-GATGACAGATAGACAGAT-GATAGGT--GATAGATAGAT-G
         @10       @20        @30       @40        @50         @60

   @80       @90       @100      @110      @120
ATAGATAGATAGATAGATAGATAGATAGATAGATAGGAAGTACTTAGA
******.******* ***.********:******::.*::***.****
ATAGATTGATAGAT-GATTGATAGATAAATAGATGATAGATACATAGA
    @70        @80       @90       @100      @110
```

Figure 5. Parsing of PO window 901-1100 and its local alignment with TPA window 2001-2200.

including the $(TAGA){*}N$ structure, as well as an alignment of the same window with the window 2001-2201 from the TPA segment including the $(TGATAGA) * N$ structure. The alignment clearly indicates high similarity that is exclusively due to the shared internal structure.

6. Discovering life

The extended algorithmic significance method can be used for discovery of a wide variety of dependencies. The domain of applications includes, but is not restricted to DNA sequence analysis. As we will demonstrate shortly, the method is particularly well suited for applications in biology.

Before we proceed further, we should mention that the definition of mutual information $I(s;t)$ can be slightly modified so that it becomes symmetrical in s and t; the technical details involved in are reviewed elsewhere (e.g., Li & Vitányi, 1993), so we omit them here. We will use the fact that symmetrical version of mutual information can alternatively be defined as the difference between the sum of individual encoding lengths and their joint encoding legth. More precisely,

$$I(s;t) = I(t) + I(s) - I(s,t),$$

where $I(s,t)$ denotes the joint encoding length. By substituting $I(s;t)$ into Theorem 2, we obtain

$$P_0\{I(t) + I(s) - I(s,t) \geq d\} \leq 2^{-d+O(1)}.$$

In other words, d bits of difference between the sum of individual encoding lengths and the joint encoding length implies dependence at the significance level $2^{-d+O(1)}$.

Chaitin (1979) considers the case where a domain of observations t is split into subsegments $t_1, \ldots, t_k$ and then considers algorithmic mutual information, which is computed as the difference between the sum of individual encoding lengths of $t_1, \ldots, t_k$, plus some overhead, and the joint encoding length. As an example, Chaitin considers multidimensional patterns of squares, as illustrated in Figure 6. If the sum of the encoding lengths of individual "windows" significantly exceeds the encoding length of the whole then mutual information is significantly high. This occurs precisely in case when the individual windows of observation are too small to capture a pattern, as illustrated in Figure 6: the whole pattern can be encoded more concisely by taking advantage of its regularity, which is invisible when only small pieces of the pattern are observed.

Chaitin then goes on to consider mutual information as a function of diameter D of windows: if the patterns are small, mutual information becomes negligible even for small D, while high mutual information for large D implies presence of even larger patterns that cannot be observed through windows of diameter D. Chaitin convincingly argues that the living world can be distinguished from the non-living by the following abstract property: algorithmic mutual information in the non-living world becomes negligible even for small diameters D while in case of the living world it remains high even for large D.

It is interesting to point out that the extended algorithmic significance method can be used to discover life, as defined by Chaitin. For a prespecified significance level $2^{-d+O(1)}$ and for a sufficiently large diameter D (to remove "interference" from the patterns in the non-living world, e.g., physical laws) one simply has to show that mutual information exceeds d bits.

Most interestingly, when constructing artificial examples of life, in his Theorem 5 Chaitin (1979) constructs hierarchical structures that resemble repetitive DNA sequences studied in this paper. Chaitin argues that if replication occurs at different hierarchical levels (e.g., tandem repeats of small segments vs. repetitions of larger segments that include many small segments), then the resulting pattern cannot be fully observed unless repetitions on the largest scale fit within a window. That is precisely the choice that we are implicitly making in sequence comparisons: a smaller window accommodates a single sequence while a larger one accommodates both sequences, and the problem

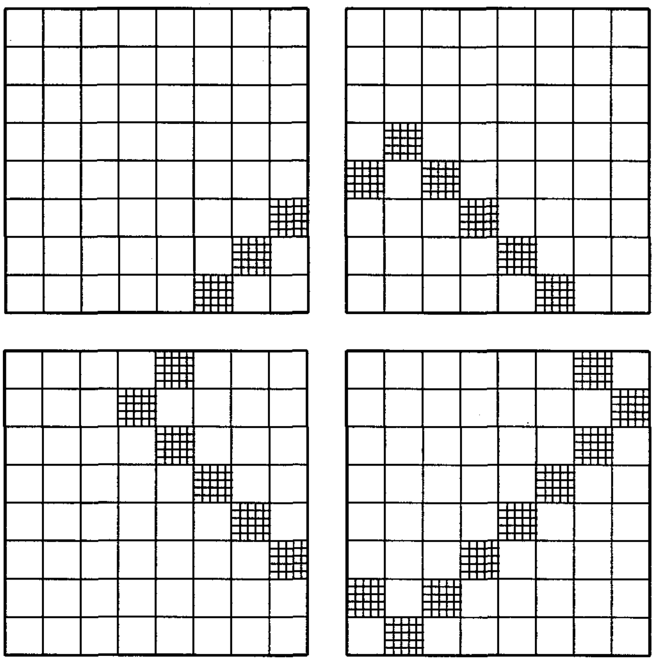

Figure 6. A pattern that does not fit within any of the four small windows. The presence of the global pattern implies high algorithmic mutual information between individual windows. This example is patterned after the one that appears on the cover of Chaitin's book (1987b).

is to decide on the size of the window. If small windows suffice for most concise encoding then sequences are unrelated, i.e., patterns are local; if larger windows give shorter encodings then sequences are related, i.e., the pattern is global.

An interesting recent example illustrating the need for large diameters of observation is the reconstruction of the evolution of Alu sequences (Jurka & Milosavljević, 1991; Milosavljević & Jurka, 1993b). The standard "bottom-up" methods for evolutionary reconstruction that are based on pairwise sequence comparisons have failed in this case: the global evolutionary pattern of Alu sequences was invisible when only two Alu sequences were considered at a time, as in Bains (1986). The evolutionary pattern becomes visible only through a "top-down" approach where a large number of sequences are considered simultaneously, as in Milosavljević and Jurka (1993b).

7. Conclusion

Repetitive patterns in DNA sequences may be complex and hard to discover. When comparisons are made by subword similarity alone, the DNA segments that share common internal repetitive patterns consisting of repetitions of very short words turn out to be most similar, even though they are not related, because different occurrences of short words tend to multiply independently. We have shown how the concept of algorithmic mutual information can be used to discover similarities that are due to relatedness and not due to shared internal structure.

Mutual information is in effect the difference between the complexities of two alternative hypotheses about the observed similarity between two sequences: one hypothesis is based on internal structure (minimal length encoding based on internal structure) while the other is based on pairwise similarity (minimal length encoding based on pairwise similarity). In that sense, we resolve the two competing hypotheses by applying the parsimony principle.

Perhaps the most important contribution of this paper is the extension of the algorithmic significance method. The extended method is based on Theorem 2, which states that d bits of algorithmic mutual information imply dependence between s and t at the significance level $2^{-d+O(1)}$. The method is general in the sense that by applying specific encoding schemes we may discover dependencies of different kinds, while still relying on the same method for establishing significance. DNA sequence comparison is only one of many possible applications of this method; new applications in the context of massive hybridization experiments and alternative sequence representations are currently under development.

An additional contribution of this paper is a new approach to sequence comparison based on subword composition. Current methods (reviewed in Pevzner, 1992) 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. 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 a bound on significance, thus removing need for any arbitrary thresholds. Experiments indicate that this systematic approach can eliminate false positive matches.

Mutual information can also be applied to discover similarity based on sequence alignment. In case of alignments, the target sequence would have to be encoded using a set of edit operations. A minimal length encoding approach to sequence alignment has been discussed in Allison and Yee (1990).

Acknowledgments

Thanks to David Nadziejka for editorial assistance. 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

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

Altschul, S.F., Boguski, M.S., Gish, W. & Wootton, J.C. (1994). Issues in searching molecular sequence databases. *Nature Genetics*, 6:119–129.

Bains, W. The multiple origins of human Alu sequences. (1986). *Journal of Molecular Evolution*, 23:189–199.

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

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

Chaitin, G.J. (1979). *The Maximum Entropy Formalism*, chapter Toward a Mathematical Definition of Life, pages 477–498. MIT Press. Levine, R.D. and Tribus, M. (eds).

Chaitin, G.J. (1987). *Algorithmic Information Theory*. Cambridge University Press.

Chaitin, G.J., (1987) *Information, Randomness and Incompleteness: Papers on Algorithmic Information Theory*. World Scientific.

Claverie, J.-M. & States, D.J. (1993). Information enhancement methods for large scale sequence analysis. *Computers in Chemistry*, 17:191–201.

Cover, T. & Thomas, J. (1991). *Elements of Information Theory*. Wiley.

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

Jurka, J. & Milosavljević, A. (1991). Reconstruction and analysis of human Alu genes. *Journal of Molecular Evolution*, 32:105–121.

Li, M. & Vitányi, P.M.B. (1993). *An Introduction to Kolmogorov Complexity and its Applications*. Springer Verlag.

Milosavljević, A. (1993). Discovering sequence similarity by the algorithmic significance method. *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*. Bethesda, MD: AAAI Press.

Milosavljević, A. (to appear). Repeat analysis. In *Imperial Cancer Research Fund Handbook of Genome Analysis*. Blackwell Scientific Publications.

Milosavljević, A. & Jurka, J. (1993a) Discovering simple DNA sequences by the algorithmic significance method. *Computer Applications in Biosciences*, 9:407–411.

Milosavljević, A. & Jurka, J. (1993b). Discovery by minimal length encoding: A case study in molecular evolution. *Machine Learning*, 12:69–87.

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

Storer, J.A. (1988). *Data Compression: Methods and Theory*. Computer Science Press.

Wootton, J.C. & Federhen, S. (1993). Statistics of local complexity in amino acid sequences and sequence databases. *Computers in Chemistry*, 17:149–163.

Received October 7, 1993
Accepted July 5, 1994
Final Manuscript August 25, 1994