Discovering Related DNA Sequences via Mutual Information

Aleksandar Milosavljević

Genome Structure Group

Center for Mechanistic Biology and Biotechnology

Argonne National Laboratory

Argonne, Illinois 60439-4833

e-mail: milosav@anl.gov

RECEIVED
OCT 14 1994
OSTI

Abstract

One of the problems in DNA and protein sequence comparisons is to decide whether the observed similarity should be explained by their relatedness or by the mere presence of some shared internal structure, e.g., shared internal repetitive patterns. Machine discovery of related DNA sequences critically depends on a solution to this problem. In this paper we propose a general solution that is based on minimal length encoding: we measure mutual information between two sequences as the difference between the encoding length of one of the sequences sequence and its encoding length relative to the other sequence; two sequences are considered similar if the mutual information exceeds a threshold of significance; the significance is determined using an extension of the newly proposed algorithmic significance method. We show 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 this general method, 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.

The submitted manuscript has been authored by a contractor of the U. S. Government under contract No. W-31-109-ENG-38. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.

MASTER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

1 Introduction

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 the similarity.

Ideally, a measure of sequence similarity should reflect the likelihood that two sequences are related. Only the simplest kind of internal structure, the bias in sequence composition, is typically taken into account: in order to establish the significance of a particular similarity score, the nucleotides in one of the sequences are randomly permuted many times and the distribution of the similarity score is established. This approach not only involves much computation, but also the incorrect assumption that the similarity score exhibits a Gaussian distribution. Repetitions of subwords are not considered, even though they abound in the genome.

The problem can clearly be phrased in terms of the parsimony principle (Occam's Razor): is it more parsimonious to explain a similarity of two sequences by postulating an ancestral or functional relationship between them or by postulating that their common internal structures have emerged independently? We apply the most general formulation of the parsimony principle in terms of minimal length encoding in order to answer this question. The main strategy will be to compute the encoding lengths that result from adopting each hypothesis, and to choose one over the other if it leads to a significantly shorter encoding length. (For a broader introduction to minimal length encoding and its relationship to inductive inference, see [4, 6].)

Let us 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 [8] and let $I(t|s)$ be the number of bits needed to encode t relative to s by taking advantage of the similarity in their structure [7]. The difference $I(t; s) = I(t) - I(t|s)$ between the first and the second encoding length is the 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 also employ the newly proposed algorithmic significance method [8] to show that the sequences related at the significance level $2^{-I(t; s)}$.

While the general approach using minimal length encoding and mutual information can be applied to comparisons of any two objects, we here restrict discussion to comparison of sequences using their subword composition. The search for sequence similarity based on subword composition is a common concept now. Pevzner [9] 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.

The main improvement over the current methods for sequence comparison is the removal of unnecessary assumptions about word length. Current methods are 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.) 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.

2 Minimal Encoding Length and Similarity

In this section we review methods for concisely encoding sequences by taking advantage of repeated subwords. We first assume that there are two sequences; one of them, called target, will be encoded by using references to the other, which we call source. A target sequence can be encoded concisely by replacing some words in it by pointers to the occurrences of the same words in the source sequence. This is a standard technique in data compression [10]. 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

For 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 than 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.

Not only can a sequence be encoded relative to another sequence, but it can also be encoded relative to itself. The main idea here is that a repeated occurrence of a word can be replaced by a pointer to its previous occurrence within the same sequence. If there is enough repetition within a sequence, bits can be saved. We omit the details here because they have been discussed elsewhere [8].

3 Algorithmic Significance and Mutual Information

In the following we show that a high level of mutual information between two sequences is unlikely to occur by chance; this is based on an extension of the algorithmic significance method [8]. 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 mutual information is unlikely to occur by chance as well.

Let P_0 and P_A be the probability distributions over sequences 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 $d = \log c$. $\square$

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, and will be soon defined in terms of encoding length.

Let A denote a decoding algorithm that can reconstruct the target t based on its encoding and based on access to the source s . By $I_A(t|s)$ we denote the length of the encoding. 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 over target sequences, decoding algorithm A , and source sequence 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 a sequence t . Now we apply Lemma 1 to obtain

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

After substituting $p_A(t|s)$, 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

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

□

Theorem 1 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 [8, 7].

Note that so far the null hypothesis has been specified by a probability distribution. We now redefine it in terms of minimal length encoding. For that purpose we introduce algorithm A_0 for decoding the target without reference to the source. We make two assumptions about A_0 :

- A_0 halts and outputs a sequence for all inputs.
- Every sequence has a unique encoding.

Since these assumptions are typically not true in practice, the value of $p_0(t)$ may be underestimated, which may lead an overestimate of the significance of similarity and, consequently, to false positive matches. This may be compensated by choosing a slightly higher significance threshold. The experiments presented in the following sections indicate that the problem of false positive matches does not arise even for reasonably low significance thresholds.

Under the assumptions above, the null hypothesis P_0 can be defined as $p_0(t) = 2^{-I_{A_0}(t)}$ and the mutual information between t and s can be defined as $I(t; s) = I_{A_0}(t) - I_A(t|s)$. Note that any shared internal structure in t and s would decrease both $I_{A_0}(t)$ and $I_A(t|s)$. Thus, mutual information enables

us to detect similarity that is due only to the relatedness of sequences, and not due to shared internal structure.

Because we do not use powerful enough algorithms A_0 and A , our definition of mutual information is only an approximation of the universal definition in the sense of Kolmogorov (e.g., [6]), and it essentially depends on our choice of algorithms A and A_0 . The approximation can be justified by the fact that A and A_0 indeed capture the kinds of structure that are expected to occur in the sequences.

Based on the definitions of P_0 and $I(t; s)$, and based on Theorem 1, we conclude that significance of sequence similarity is exponentially related to the mutual information of two sequences. In other words,

Lemma 2

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

The exponential relationship 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 a mutual information of $I(s; t) = 7 + \log L$ bits would suffice for a significance level of 0.01.

4 Minimal Length Encoding Algorithm

To compute mutual information between two sequences, algorithms are needed that would compute the minimal lengths of the encodings of sequences for the decoding algorithms A_0 and A . We are interested in the decoding algorithms that would compress sequences based on repeated occurrences of words: in the encodings accepted by A_0 , the repeated occurrences of words are replaced by pointers to the previous occurrences within the same sequence, while in the encodings accepted by A , the occurrences of common words in one sequence are replaced by the pointers to the occurrences of the same word in the other sequence.

Not surprisingly, the algorithms for computing minimal encoding lengths for A_0 and A are very similar. A standard algorithm for self-encoding has already been presented elsewhere [8]. In this section we present a slight

variant of the same algorithm that can be used for encoding one sequence relative to the other. The only difference between the two is that in the former case pointers point to the occurrences of words within the same sequence while in the other case they point to the occurrences of the words in the other sequence.

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 . 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 [10].

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 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. We present two experiments that illustrate the potential of the method.

Figure 1: Segment 22,001 – 26,000 from the TPA gene.

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 [5]. The segment 22,001-26,000 that was extracted from the GenBank [2] 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. A pointer length of 6 bits was chosen for self-encoding and 12 bits for encoding of one sequence relative to the other (that is, it was assumed 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

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 $AAAyxyxyxTAAA$. 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 [2] 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

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

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

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

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.

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 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 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 Discussion

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

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-AGATAGATAGATAGATAGATAGATA
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
TAAATGAAGAT--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.

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. It is pleasing to see that the difference, in bits, between the two competing hypotheses is in fact the exponent of algorithmic significance.

An additional contribution of this paper is a new approach to sequence comparison based on subword composition. 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. 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. For this case, 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 [1].

7 Acknowledgements

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] S.J. Friezner-Degen, B. Rajput, and E. Reich. The human tissue plasminogen activator gene. *Journal of Biological Chemistry*, 261:6972–6985, 1986.
- [6] M. Li and P.M.B. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer Verlag, 1993.
- [7] A. Milosavljevic. Discovering sequence similarity by the algorithmic significance method. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1993.
- [8] A. Milosavljević and J. Jurka. Discovering simple DNA sequences by the algorithmic significance method. *Computer Applications in Biosciences*, 9(4), 1993.
- [9] P.A. Pevzner. Statistical distance between texts and filtration methods in sequence comparison. *Computer Applications in Biosciences*, 8(2):121–127, 1992.
- [10] J.A. Storer. *Data Compression: Methods and Theory*. Computer Science Press, 1988.