# ALGORITHMIC ASPECTS OF GENETIC SEQUENCES AND RELATIVE KOLMOGOROV COMPLEXITY

Alla Grigorieva[1], Dima Grigoriev[2]

[1] St.Petersbourg University, Universitetskaya nab., 7/9,
St.Petersbourg, 199164, RUSSIA

[2]IRMAR, Université de Rennes Beaulieu, 35042, Rennes, FRANCE
e-mail: dima@math.univ-rennes1.fr
URL: http://name.math.univ-rennes1.fr/dimitri.grigoriev

**Abstract:** In this paper we use the concept of the relative Kolmogorov complexity for searching fast computable short representations of finite sequences. We investigate different types of regularities of such sequences in order to obtain polynomial-time algorithms for certain versions of the relative Kolmogorov complexity or for their majorants, which can be applied to genetic sequences.

**AMS Subject Classification:** 68Q30

**Key words:** relative Kolmogorov complexity, computational complexity, information retrieval.

# 1. Introduction

In this paper two versions of the relative Kolmogorov complexity of finite sequences are introduced. We use structural information, different types of repetitions of the motifs, edit operations (like insertion, deletion), complementary words (which arise while studying DNA sequences) in order to obtain their fast computable descriptions.

The fundamental definition of the Kolmogorov complexity $K(x)$ of a finite object $x$ is a minimum of lengths of all the programs (descriptions of the object) that output this object. It is well known that the Kolmogorov complexity is uncomputable Li et al. [7], Zvonkin et al. [12]. In Rivals et al. [9] an *average* optimal representation based on the Kolmogorov complexity was explored.

We use the idea which lies behind the Kolmogorov complexity, i.e. to find optimal (with respect to certain conditions) and necessary computable (and moreover, *efficiently computable*) representations of an object by specifying different sets of admissible operations in the underlying programs Jost [6], Li et al. [7], Zvonkin et al. [12]. These programs use the information about different types of the object's regularities which are applied for compressing the description of genetic sequences Sagot et al. [10]. This leads to the concept of the *relative Kolmogorov complexity.*

*Programs of the type* $F_1$ for a finite word $x$ of a length $n$ contain the operations of the concatenation and four types of repetitions on the words (more precisely, repetitions of $r$-regular subwords). The length of the program is usually the sum of the weights of its operations, we assume all the weights to be equal to 1.

*Programs of the type* $F_2$ use all the operations from the type $F_1$ and also insertions and deletions of the letters. We can assume without loss of generality that $x$ is a word over the alphabet $\{0,1\}$. Two variants $K_{F_1}(x)$ and $K_{F_2}(x)$ of the relative Kolmogorov complexity of the word $x$ are introduced. The relative Kolmogorov complexity was studied not only for (linear) strings, but also for some 2-dimensional objects Grigorieva et al. [3, 4]. An *algorithm for computing* $K_{F_1}(x)$ with the quadratic time complexity $O(n^2)$ is designed, where $n = |x|$ denotes the length of the word $x$. For $K_{F_2}(x)$ we produce a majorant which is computable within time complexity $O(n^6)$.

# 2. Basic definitions

Let $X^* = \{0,1\}^*$ be the set of all the finite strings $x$ over the alphabet $\{0,1\}$. Let $x$ be a word of a length $n$, $x = x_1 \dots x_n, n \geq 1$, when $n = 0$ the word $x$ is empty. Denote by $x_{i,j}$ the segment

$x_{i,j} = x_i x_{i+1} \dots x_j$ if $i \leq j$ or

$x_{i,j} = x_j x_{j-1} \dots x_i$ if $i \geq j$.

When $x = x_1 \dots x_n, y = y_1 \dots y_k$ are two words then $x = y$ if $n = k$ and $x_i = y_i, 1 \leq i \leq n$. As usual, we call the word $y$ the inverse of $x$ when $y = x_n \dots x_1$ and denote $y = x^T$. The word $y$ is a subword of $x$ (we denote this by $y \subset x$) if there exist $1 \leq i \leq j \leq n$ such that $y = x_{i,j}$. The word $y$ is an inverse subword of $x$ (in this case $y^T \subset x$) if $y = x_{i,j}$ for suitable $1 \leq j \leq i \leq n$.

We also study the finite words in the alphabet of *nucleotides* $\Sigma = \{A, C, G, T\}$ where $(A, T)$ and $(C, G)$ are *Watson-Crick base pairs* Carbone et al. [1], Sagot et al. [10]. The letters of the pair $(A, T)$ are called complementary (as well as the letters $(C, G)$). These letters name the four nucleotides from which the DNA is composed.

Let us introduce the notion of a word $p$ being a *$r$-regular subword* of a word $v$ (generalising the relation $p \subset v$) and denote this relation by $p \prec v$ taking

into account 4 following cases. Thus, $p \prec v$ holds if either $p \subset v$ (i.e. either $p$ is a subword of $v$), either $p^T \subset v$ (i.e. either $p$ is an inverse subword of $v$), either $\overline{p} \subset v$ where $\overline{p}$ is obtained from $p$ by means of transposing the complementary letters in the base pairs $(A, T)$ and $(C, G)$, or $\overline{p^T} \subset v$ (i.e. either $p$ or $\overline{p}$ is either a subword or an inverse subword of $v$, respectively).

# 3. Regularities in genetic sequences and relative Kolmogorov complexity

Let $x = yz$, i.e. the word $x = x_1 \dots x_n$ is the concatenation of the words $y = y_1 \dots y_{i-1}$ and $z = z_1 \dots z_m$, $n = i - 1 + m$, in a particular case when $z$ consists of a single letter $a$ the concatenation becomes $x = ya$. The subword $y$ of $x$ is called its prefix, and $z$ is called the suffix of $x$. For integers $k, s$ we denote a subword $x_{k,k+s} = x_k x_{k+1} \cdots x_{k+s}$ of $x$ and by $x_{k+s,k} = x_{k+s} x_{k+s-1} \cdots x_{k+1} x_k$ an inverse subword of $x$.

Introduce the following operations on words:

**Insertion** of a word $v$ into the word $x$ starting with the position $i$, i.e.

$I_i[v](x) = yvz$

where $x = yz$, $y = x_1 \dots x_{i-1}$. Its particular case is the insertion $I_i[a](x)$ of a letter $a$ into the word $x$ at the position $i$ (i.e. $v = a$ in the previous notations).

**Deletion** of a subword $x_{i,i+k}$ from the word $x$ we define as follows:

$D_{i,i+k}(x) = x_{1,i-1} x_{i+k+1,n}$, $0 \le k \le i + k \le n$.

In case when $i + k = n$ we have in the above notations that $D_{i,n}(x) = y$, thus, a subword $z$ is deleted. A particular case of the deletion is the deletion $D_j(x)$ of a letter from the position $j$ (i.e. $k = 0$ in the previous notations).

**Repetition** of a subword $x_{i,i+k}$ of the word $x$ is the concatenation of $x$ and of $x_{i,i+k}$:

$R_{i,i+k}(x) = x x_{i,i+k}$.

**Repetition of the inverse** of a subword $x_{i,i+k}$ of the word $x$ is the concatenation of $x$ and of the inverse $x_{i+k,i}$:

$R_{i+k,i}(x) = x x_{i+k,i}$.

All the listed types of regularities allows one to use important structural information for the purpose of an algorithmic approach to genetic sequences Crochemore et al. [2], Pevzner [8], Sagot et al. [10].

In case of the alphabet $\Sigma = \{A, C, G, T\}$ for the nucleic acid sequences we regard the following analogs of the defined above operations. In addition, for a word $x$ in the alphabet $\{A, C, G, T\}$ denote here by $\overline{x}$ the result of transpositions of the letters in the base pairs $A, T$ and $C, G$, respectively.

Let us introduce some versions of the relative Kolmogorov complexity and estimate the time complexity of their calculation.

Denote by $F_1$ the set of the following three operations on words $x = x_1 \dots x_n, |x| = n$:

1) $I_a(x) = xa$ is the concatenation of the word $x$ and a letter $a$;

2) $R_{l,m}(x) = xx_{l,m}$ is the concatenation of the word $x$ and the segment $x_{l,m}$ such that $x_{l,m} \prec x$;

3) $\overline{R}_{l,m}(x) = x\overline{x}_{l,m}$, in this case we also introduce two types of repetitions of the direct and inverse subwords with the difference that transpositions are made in all the base pairs $A, T$ and $C, G$.

Now for a word $x$ we can use a program (or a description of $x$) over the set $F_1$ which one can treat as a word $f \in F_1^*$ in the alphabet $F_1$. We introduce a *metaprogram* $P_1 : F_1^* \to X^*$ which for a program $f$ outputs its result $x \in X^*$. The weights of these operations $I, R, \overline{R} \in F_1$ in the definition of the relative Kolmogorov complexity $K_{P_1}$ (with respect to $P_1$) equal to 1. More precisely, Grigorieva [5]

$$K_{P_1}(x) = \min\{|f| : P_1(f) = x, f \in F_1^*\}$$

**Definition 3.1** *Define a function $C_1(x)$ for the words $x$ by recursion on their lengths as follows relying on two auxiliary functions:*

$$C_1^{(1)}(x) = \min\{C_1(u) + |s| + 1\}$$

*where* min *ranges over all the representations of the word $x$ in the form $x = uvs$ such that $v \prec u$, provided that $|v| \geq 1$. In order to take into account the case $|v| = 0$ we put*

$$C_1^{(0)}(x) = \min\{C_1(u) + |s|\}$$

*where* min *ranges over all the representations in the form $x = us$.*

*Finally, $C_1(x) = \min\{C_1^{(1)}(x), C_1^{(0)}(x)\}$.*

**Lemma 3.2** *$C_1(x) = K_{P_1}(x)$ for any word $x$.*

*Proof* goes by induction on $n = |x|$. First we show the inequality $C_1(x) \geq K_{P_1}(x)$. Consider one of the minimal representations of the word $x = vps$, i.e. $C_1(x) = C_1(v) + |s| + 1$ where $p \prec v$. Then by the inductive hypothesis we get

$$C_1(x) \geq K_{P_1}(v) + |S| + 1 \geq K_{P_1}(x)$$

We can describe the word $x$ by the program that outputs first the prefix $v$, after that repeats an $r$-regular subword $p$ of $v$ and finally outputs all the letters of the suffix $s$ using $|s|$ operations.

Now we have to verify the inverse inequality $C_1(x) \leq K_{P_1}(x)$. Among the minimal programs for the word $x$ we look for the presentation $x = v_0 p_0 s_0$ such that $p_0 \prec v_0, |p_0| \geq 1$ and

$$K_{P_1}(x) = K_{P_1}(v_0) + |s_0| + 1$$

For this purpose we choose the last segment of the word $x$ being a repetition of a certain $r$-regular subword of the prefix $v_0$.

By the definition, we get

$$C_1(x) \leq C_1(v_0) + |s_0| + 1 \leq K_{P_1}(v_0) + |s_0| + 1 = K_{P_1}(x),$$

provided that $C_1(x) = C_1^{(1)}(x)$ (see definition of $C_1$ above).

In case of $C_1(x) = C_1^{(0)}(x)$, i.e. $|p_0| = 0$ the proof goes in a similar way. □

The (evident) upper bound on the time complexity for calculating $K_{P_1}(x)$ based on the definition of $C_1$ is exponential.

Now we propose a polynomial time algorithm to calculate the relative Kolmogorov complexity $K_{P_1}(x)$ of a genetic sequence $x$.

**Definition 3.3** *For a word $x$ we consider a representation $x = vp$ with the longest possible suffix $p$ such that $p \prec v$. Then we define by recursion on the length of a word $x$ a function*

$C_2(x) = C_2(v) + 1$

*if $|p| \geq 1$, and if $|p| = 0$ we denote $x = x'a$ where $a$ is the last letter of $x$, then*

$C_2(x) = C_2(x') + 1$.

**Lemma 3.4** $C_2(x) = C_1(x)$

*Proof.* Obviously, we have $C_2(x) \geq C_1(x)$.

We establish the inverse inequality $C_2(x) \leq C_1(x)$ by induction on $n = |x|$. Let $x = v_0 p_0 s_0, p_0 \prec v_0$, see the definition of the function $C_1$ (without loss of generality one can assume that $|s_0| \geq 1$), such that for this representation of $x$ holds the equality

$C_1(x) = C_1(v_0) + |s_0| + 1$.

We also take a representation $x = vp, p \prec v$ for which $C_2(x) = C_2(v) + 1$, see the definition of the function $C_2$ (one can assume that $|p| \geq 1$). Three following cases can occur:

1) $|p| \leq |s_0|$. In this case $x = v_0 p_0 s_0 = v_0 p_0 s_0' p$ and therefore, one can diminish the length of the program $f \in F_1^*$ such that $P_1(f) = x$ by means of repeating one of the $r$-regular subwords of $v_0$;

2) $|s_0| < |p| < |s_0| + |p_0|$. Then
$C_2(x) = C_2(v) + 1 \leq C_1(v) + 1 \leq C_1(v_0) + 2 \leq C_1(v_0) + 1 + |s_0| = C_1(x)$

3) $|p| \geq |p_0| + |s_0|$. Then
$C_2(x) = C_2(v) + 1 \leq C_1(v) + 1 \leq C_1(v_0) + 1 + |s_0| = C_1(x)$. $\square$.

One can bound the time complexity $T_2(n)$ of an algorithm which computes $C_2(x)$ (where $|x| = n$) according to its recurisive definition, by $T_2(n) \leq O(n^2)$ because of the recursive formula

$$T_2(n) \leq T_2(n-1) + cn$$

for a suitable constant $c > 0$ due to the linear-time algorithm for pattern matching (see e.g. Slissenko [11]).

Thus, we have designed a polynomial-time algorithm for a version of the relative Kolmogorov complexity based on the operations of repetitions and its appropriate modifications.

# 4. Affinity of genetic sequences and relative Kolmogorov complexity

We introduce a set of the following operations on the words:

$$F_2 = F_1 \cup I_i[a](x) \cup D_j(x), |x| = n, 1 \leq i, j \leq n$$

i.e. we add to $F_1$ two new operations: insertion and deletion of a letter with the weights of both operations (for the purpose of definition of a version of the relative Kolmogorov complexity) equals to 1. If necessary, one could use their composition, namely, the operation of a substitution $S_j[a](x)$ whose result is replacing a letter of the word $x$ at the position $j$ by the letter $a$ imposing the weight of the substitution equal to 2.

In this section we study the relative Kolmogorov complexity involving not only regularities of genetic sequences, but also affinity of pairs of sequences.

Us usual, the *edit distance* $d(y, z)$ between the words $y, z$ (let us denote $|y| = k, |z| = l$) equals to a minimal number of insertions and deletions needed to transform $y$ into $z$. The problem of computing the edit distance is equivalent to the problem of finding the *longest common subsequence* $s(y, z)$ of the words $y, z$, one has $d(y, z) = k + l - 2|s(y, z)|$ (cf. e.g. Pevzner [8]).

Denote by $P_2$ a metaprogram (or in other words, a decoding algorithm) $P_2 : F_2^* \to X^*$ which transforms a program over the operations from $F_2$ into a word $x$ being a result of this program. Then denote by $K_{P_2}(x)$ a version of the relative Kolmogorov complexity with respect to $P_2$. In order to compute $K_{P_2}(x)$ we introduce the following auxiliary function.

**Definition 4.5**

$$C_3(x) = \min\{C_3(v) + d(v, v') + d(p, p') + 1\}$$

*where* min *is taken over all the representations of the form* $x = v'p'$ *and the words* $p, v$ *such that* $p \prec v$

**Lemma 4.6** *1)* $C_3(x) = K_{P_2}(x)$ *for any word* $x$*;*
*2) in the definition of* $C_3$ *it suffices to take* min *just over the words* $v$ *of lengths at most* $n$*, i.e.* $|v| \leq |x|$*.*

*Proof.* The first item can be proved similar to the proof of Lemma 3.2.

To prove the item 2) let us fix some representation of the word $x = v'p'$. Take words $v, p$ such that $v \prec p$ for which min is attained for the expression $C_3(v) + d(v, v') + d(p, p') + 1$ in definition 4.5.

Let $v = v_2 p_1 v_3$ where the subword $p_1$ is taken from the definition of $r$-regular subword of $v$, i.e. $p_1$ equals to $p$ up to (possible) inversions or taking complementary letters in the base pairs. One can transform $v$ into $v'$ and $p$ into $p'$, respectively, using $d(v, v')$ and $d(p, p')$ operations of insertions and deletions. One can assume without loss of generality (cf. Pevzner [8]) that in both transformations first the operations of deletions are accomplished followed by the insertions. Denote by $l_1$ the number of operations of deletions in the sequence of transformations from $v$ to $v'$ which are situated at the position corresponding to the letters of $p_1$, i.e. in the range $\{|v_2| + 1, \ldots, |v_2| + |p|\}$. Denote by $l_2$ the number of operations of deletions in the sequence of transformations from $p$ to $p'$.

One can see that in these transformations no letter which corresponds to the *same position in the word* $p$ is not deleted, because otherwise one could delete this letter from the word $v$, increasing thereby the item $C_3(v)$ by at most by 1, while decreasing each of $d(v, v')$ and $d(p, p')$ by 1, thus decreasing the expression

$$C_3(v) + d(v, v') + d(p, p') + 1$$

which would contradict to its minimality by means of the made choice of $p, v$. Hence the number of all deletions $l_1 + l_2$ which correspond to the positions of the word $p$ does not exceed the length of $p$, i.e. $l_1 + l_2 \leq |p|$. Evidently, we have $|v'| \geq |v| - l_1$, $|p'| \geq |p| - l_2$. Therefore,

$$|v| \leq |v'| + l_1 \leq |v'| + |p| - l_2 \leq |v'| + |p'| \leq |x|$$

**Remark 4.7** *1)* $\log_2 |x| + 1 \leq C_i(x) \leq |x|$, $i = 1, 2, 3$*;*
*2) A time upper bound of an obvious algorithm based on the described recursive formula for computing* $C_3(x) = K_{P_2}(x)$, $|x| = n$ *is* $O(2^{n^2})$*.*

Since due to the latter remark the time bound for computing the relative Kolmogorov complexity $K_{P_2}(x)$ is exponential it is reasonable to give a certain *majorant* for $K_{P_2}(x)$ computable within polynomial time. Namely, we define by recusion the function

$$C_4(x) = \min\{C_4(v) + d_1(v,p) + 1\}$$

where min is taken over all the prefixes $v$ of the word $x$ such that $x = vp$, $|v| < |x|$ and $d_1(v,p) = \min\{d(u,p)\}$ where min ranges over all the words $u$ satisfying $u \prec v$. One can easily prove by recursion on $x$ that $C_3(x) \leq C_4(x)$, taking into account that the definition of $C_4$ is based in fact, on a particular form of sequences (programs) of operations from $F_2$. On the other hand, the direct use of the recursion for defining $C_4$ allows one to compute $C_4(x)$ for $|x| \leq n$ relying on the linear time algorithm for string matching (see e.g. Slissenko [11]). Let us summarize the established above on $C_4$ in the following lemma.

**Lemma 4.8** *1)* $K_{P_2}(x) \leq C_4(x)$ *for any word* $x$*;*
*2) One can compute* $C_4(x)$ *within time* $O(|x|^6)$*.*

Thus, we have considered few algorithmic approaches to the relatively optimal computable descriptions of genetic sequences and their transformations.

# References


[1] A. Carbone and M. Gromov, *Mathematical slices of molecular biology*, Preprint IHES, (2001), 1–85.

[2] M. Crochemore, C. Hancart and T. Lecroq, *Algorithmique du texte*, Vuibert (2001).

[3] R. Granvoskaya, I. Bereznaya and A. Grigorieva, *Perception of form and forms of perception*, Lawrence Erlbaum Ass. Publishers, Hillside, N.J. (1987).

[4] R. Granvoskaya, I. Bereznaya and A. Grigorieva, *Perceptual complexity of form*, Cognition and brain theory, **4** (1981).

[5] A. Grigorieva, *Complexity mesures of the words based on string-matching and edit distance*, J. of Soviet Mathematics, **22** (1983), 1289–1292.

[6] J. Jost, *On the notion of complexity*, Theory Bioscience, **117** (1998), 161–171.

[7] M. Li and P. Vitanyi, *Introduction to Kolmogorov complexity and its applications*, Springer (1997).

[8] P. Pevzner, *Computational molecular biology*, MIT Press (2000).

[9] E. Rivals and J.-P. Delahaye, *Optimal representation in average using Kolmogorov complexity*, Theoret. Comput. Sci., **200** (1998), 261–287.

[10] M.-F. Sagot and A. Viari, *Flexible identification of structural objects in nucleic acid sequences: palindromes, mirror repeats, pseudoknots and triple helices*, Lect. Notes in Comput. Sci., **1264** (1997), 224–246.

[11] A. Slissenko, *Linguistic considerations of devising effective algorithms*, in Proc. International Congress of Mathematicians (1984), 347–357.

[12] A. Zvonkin and L. Levin, *The complexity of finite objects and the algorithmic concepts of information and randomness*, Russian Mathematical Surveys, **25** (1970), 83–124.