An approach to analysis of the similarity of DNA-sequences

B F Melnikov[1], E A Melnikova[1], S V Pivneva[1] and M A Trenina[2]

[1]Russian State Social University, Wilhelm Pieck str. 4, Moscow, Russia, 129226
[2]Togliatti State University, Belorusskaya str. 14, Togliatti, Russia, 445020

Abstract. This paper describes algorithms, corresponding computer programs and the results of computations, supplementing results published earlier. We consider the multiple sequence alignment problem, which can be nominated by a central problem in computational biology. For it, we continue to consider some different versions of so-called "triangular norm" defined on the set of triangles formed by the different distance between genomes computed by different algorithms.
Besides, one of the problems considered in biocybernetics is the problem of reconstructing the distance matrix between DNA sequences, when not all the elements of the matrix under consideration are known at the input of the algorithm. In this connection, the problem arises that the developed method of comparative evaluation of algorithms for calculating the distances between sequences should be used for another problem, i.e., for reconstructing the matrix of distances between DNA sequences.
In this paper, we consider the possibility of applying the method of comparative evaluation of the algorithms for calculating the distances between a pair of DNA strings that we developed and studied earlier for the reconstruction of a partially filled distance matrix. The restoration of the matrix occurs as a result of several computational passes. Estimates of unknown matrix elements are averaged in a special way using so-called risk functions, and the result of this averaging is considered as the received value of the unknown element.

1. Introduction and motivation

In this paper, we describe the continuation of research on the issues that we started in [1, 2, 3, 4, 5]. We describe algorithms, corresponding computer programs and the results of computations, supplementing results published earlier. We consider the multiple sequence alignment problem, which can be nominated by a central problem in computational biology. For it, we continue to consider some different versions of so-called "triangular norm". (The name "triangular metric", also sometimes encountered in our previous papers, is also quite possible and does not contain errors, we will not give detailed comments on this thing. However, the "norm" in our case is some more correct, both when we speak about the whole matrix, and when we speak only about the only triangle.) Such norm defined on the set of triangles formed by the different distance between genomes computed by different algorithms. Thus, in this paper the word "metric" will be used only as the distance between the genomes, and the word "norm" as an indicator of the badness of a certain set of such distances.

In previous cited papers, we described Panin's metric and some algorithms for its improvement. In this paper, we consider some variants of investigation of various variants of the triangular norm. Let us remark, that both these directions are interrelated. Namely,

we improve the interpretation of the Panin's metric in the same way as it is done in some interpretations of genetic algorithms: we are trying to achieve a combination of parameters, for which the triangular norm reaches a minimum value (or is close to it).

Considering this second problem, i.e., the study of variants of the triangular norm, we proceed as follows. We consider incorrect variants of obtaining the triangle inequality, which for matrices of the order of about 50×50 is violated in the two most successful metrics (including the Panin's metric earlier developed by us) in less than 1% of cases. It is important to note that in order to improve the decrease in the quantitative index of the badness of the entire matrix of considerations, we also, first of all, consider triangles in which the triangle inequality is not violated, but in which the badness value is relatively large. Possible improvements are related to the use of neural networks that were not used by us in previous calculations. In this case, neural networks solve the inverse problem: we improve (reduce) the overall badness of the matrix of distances between genomes, forcibly changing the previously obtained distances; further, we try to reflect these forced changes in the original algorithms for calculating distances.

Thus, to determine the distance between genomes, we need *heuristic* algorithms; and, if possible, they should not require too much time. There are various such algorithms, but their obvious disadvantage is getting a few *differing results* when using different heuristic algorithms applied to the distance calculation between *the same pair of DNA strings*. Therefore, the problem of *quality evaluation* of the metrics used (distances) arises; and based on the results obtained in solving this problem, one can draw conclusions about the applicability of a particular algorithm for calculating distances to various applied studies. A possible approach to determining the quality of metrics was given in [4]; also partially we consider these approaches below in this paper.

However, we used the same approach for a completely different problem; it is as follows. Even heuristic algorithms require often large time-consuming costs: for example, to construct a matrix of the order of 50×50 into which the distances computed by the Needleman – Wunsch algorithm ([6] etc.) are recorded, it takes about 28 hours (at a processor clock frequency of about 2 GHz, see [4]). Therefore, one of the problems considered in the biocybernetics is the problem of *restoration* of the distance matrix between DNA sequences (below we simply shall write "DNA matrix"), in which *not all* elements of the matrix under consideration are known at the input of the algorithm, see [7, 8], compare also [9, 10]. In connection with all this, another problem arises: to use the developed method of comparative evaluation of algorithms for calculating distances between sequences for a completely different purpose, namely, for the briefly described *problems of reconstructing the distance matrix* between DNA sequences. For this problem, in this article we consider the application of the previously developed *method of comparative evaluation of distance calculation algorithms* between a pair of DNA strings.

Using this approach (i.e., using the method of comparative evaluation of algorithms for calculating distances to matrix reconstruction), the reconstruction itself occurs as a result of several computational passes. On each of the passes, for some of the as-yet-unfilled (unknown) elements of the matrix, different estimates are obtained; these estimates are averaged in a special way-and the result of averaging is taken as the value of the unknown element. From the physical point of view, the applied averaging gives the position of the center of gravity of a one-dimensional system of bodies whose mass is specified by a special function, i.e., the risk function, see [11, 12]. We note that earlier we used risk functions in completely different subject areas; these areas were always connected with auxiliary algorithms related to multicriteria optimization.

Below, the matrices to be reconstructed are referred to as *incompletely filled with distance matrices*. We *enter this term for the matrix*, from which a number of elements are "crossed out".

2. Preliminaries

Thus, like our previous papers, we consider the square symmetric matrix of distances between genomes. There it is necessary to note the following. First, the genomes we choose random enough and took them off the site [13]. Second, like previous works, we consider in fact *three* variants for each of the considered problems:

- for very distant species, including, for example, a mammal "Bison bison" and a reptilia "Apalone spinifera" (we use the official scientific Latin names), see detailed the species' list by the link for direct download [14];
- for a sufficiently close species (human and apes);
- and also for human races (in fact, they can be considered as subspecies of a biological species).

Let us note, looking ahead, we believe that our a theoretical construction is best applicable for more distant species, however, acceptable results are obtained also in two other cases.

Thus, we look at algorithms of comparing the quality of different algorithms that calculate the distance between two genomes. Apparently, all these algorithms are based on the use of various versions of the Levenshtein distance (or Levenshtein metric), see [15] and very many other following papers. It is very important to note, that for the simplest formulation of the problem (to make the strict calculating the value of Levenshtein metric for two given genomes), we unfortunately obtain a very long-running algorithm (program): it has to do with the actual length of the strings of genomes. Therefore, in each of the actual algorithms (see [16, 17, 18, 19] etc.), computation of distances between genomes in reality is a heuristic extension of the exact algorithm for calculating the Levenshtein distance. And apparently, the approach closest to our one is given by [20]; let us note, a little running forward, that it also uses a version of the branch-and-bound method.

Thus, the considered in our previous papers Panin's metric is no exclusion, It also is a heuristic algorithm for calculating the close version of such metric; it is an optimization problems, see [21] etc. However, we used in it a special approach (so called *multi-heuristic approach for discrete optimization problems*), and for it, we use the same heuristic as in very different problems. From many such problems, let us mention two ones only:

- the classical traveling salesman problem (however, we consider our own approach to this problem, and, most importantly, our original way of specifying the input data, different from the traditional geometric placement, see, e.g., [22]);
- and the problem of state-minimization for nondeterministic finite automata, see, e.g.,[23, 24].

3. The triangular norms: their study and possible attempts of improvement

It was justified in our previous works cited above, there is desirable that in the matrix of distances between genomes, any of the resulting triangles be close to an acute angled isosceles one with two angles exceeding 60 degrees. Several various empirically selected numerical characteristics describing such differences are also given in our works, see [4] etc. However, in the previous articles we did not consider detailed examples, let us consider them in this section.

In [14], the results of calculations for several metrics and several norms are presented. In this section, we shall consider the Panin's metric only, and 3 norms ("badnesses" for the triangle under consideration). Thus, for each triangle with the sides $a \geq b \geq c$ and the angles $\alpha \geq \beta \geq \gamma$ we considered the following norms:

$$\mathrm{bad}_1 = (\alpha - \beta)/\pi\,, \quad \mathrm{bad}_2 = (\alpha - \beta)/\alpha\,, \quad \mathrm{bad}_3 = (a - b)/a\,.$$

In case if $\alpha \geq 90^o$, we have considered each norm by the maximum possible (1.0) or even usually exceeding this value. We assigned an even greater value to the value of badness in the case when the three considered sides do not form a triangle at all (that is, they do not satisfy the triangle inequality); let us note, running ahead, that similar situations is happened for any of metrics considered by us.

The resulting value of the norm *of the whole matrix* was considered as the arithmetic mean of the norms of all triangles. We note that for the matrices of distances between genomes (usually from 30×30 to 50×50), the number of triangles is

$$\frac{30 \cdot 29 \cdot 28}{2 \cdot 3} = 4060$$

for dimensions 30, and 19600 for dimension 50; from these values, it is clear that the calculations we need are quite difficult.

Thus, let us consider the part of the table given on the page titled "Panin's metrics" of [14] (they are designed as an `xlsx`-file and are available for the direct download), see the table on Fig. 1. (The names of the considered species can be found there on the page titled "Types of animals".)

No. of genomes	1	2	3	4	5	6	7	8	9	10	...	15	...	25	...
1	0	2200	2904	5996	2580	4149	4206	6336	3057	3222	...	3182	...	2300	...
2	2200	0	2638	5998	2860	3922	4416	6000	3373	2982	...	2962	...	2150	...
3	2904	2638	0	6068	2890	4037	4639	6414	3647	3202	...	3201	...	2703	...
4	5996	5998	6068	0	5647	5849	5918	6066	5858	5508	...	5596	...	5618	...
5	2580	2860	2890	5647	0	4426	4145	6445	3274	3589	...	3533	...	3142	...
6	4149	3922	4037	5849	4426	0	4682	6492	4397	3996	...	4006	...	3919	...
7	4206	4416	4639	5918	4145	4682	0	6581	4230	4586	...	4577	...	4571	...
8	6336	6000	6414	6066	6445	6492	6581	0	5893	5731	...	5776	...	5950	...
9	3057	3373	3647	5858	3274	4397	4230	5893	0	3651	...	3579	...	3447	...
10	3222	2982	3202	5508	3589	3996	4586	5731	3651	0	...	1985	...	2953	...
...	...	...	...	...	...	...	...	...	...	...	0	...	...	...	...
15	3182	2962	3201	5596	3533	4006	4577	5776	3579	1985	...	0	...	2940	...
...	...	...	...	...	...	...	...	...	...	...	...	...	0	...	...
25	2300	2150	2703	5618	3142	3919	4571	5850	3447	2953	...	2940	...	0	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	0

Figure 1. The part of the table of Panin's metric.

Let us choose species with the numbers 5, 15, 25; while doing so, we specifically chose exactly the three of those considered, where the metric gives rather poor results (that means the following: the badnesses of Panin's metric give relatively worse results than other metrics comparing most other triangles of the matrix under consideration). For all 5 considered metrics, we obtain the following 5 triangles corresponding to the species with the numbers 5, 15, 25:

1) sides 3533, 3142, and 2940 (Panin's metric);
2) sides 1215, 1179, and 704 (van der Loo's 1st metric);
3) sides 4379, 4036, and 4029 (van der Loo's 2nd metric);
4) sides 2943, 2674, and 2492 (Pages' 1st metric);
5) sides 3444, 3046, and 2838 (Pages' 2nd metric)

(here, the numbers correspond to numbers of metrics of [4, 14]). Let us consider these 5 triangles and also 3 other ones, see Fig. 2:

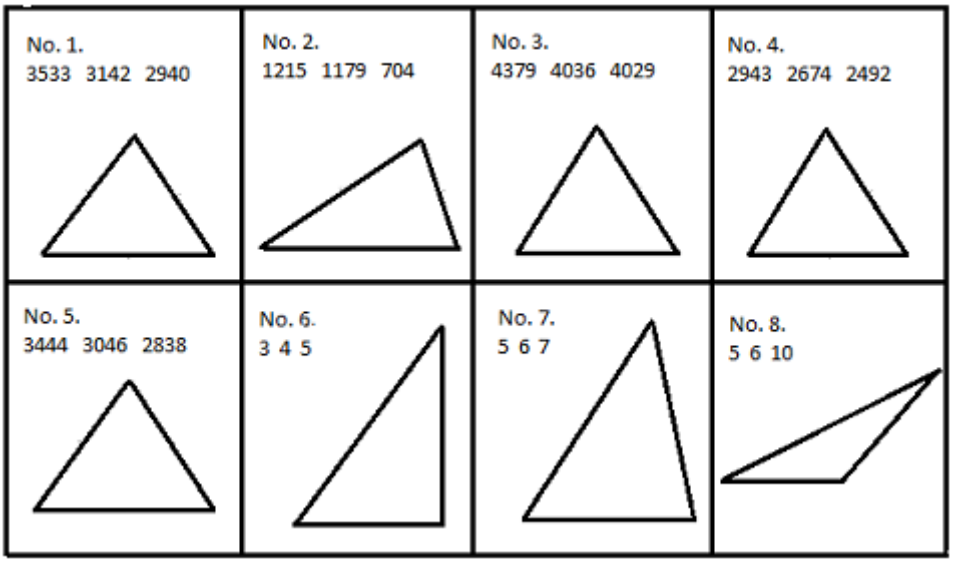

Figure 2. Examples of triangles of metrics (No.No. 1–5) and 3 other ones.

Let us note once again, that we only need the relative lengths of the sides of the triangle.

The further calculations yield the following auxiliary values and values of badnesses, see Table 1.

Table 1. The values of badnesses and the auxiliary calculations.

No.	a	b	c	$\cos(\alpha)$	$\cos(\beta)$	$\cos(\gamma)$	$\alpha, ^o$	$\beta, ^o$	$\gamma, ^o$	bad_1	bad_2	bad_3
1	3533	3142	2940	0.33	0.54	0.62	70.1	57.2	51.8	0.076	0.194	0.111
2	1215	1179	704	0.25	0.34	0.83	75.4	70.2	34.2	0.029	0.068	0.027
3	4379	4036	4029	0.41	0.541	0.544	65.8	57.19	57.05	0.048	0.130	0.078
4	2943	2674	2492	0.35	0.53	0.61	69.4	58.1	52.5	0.062	0.160	0.091
5	3444	3046	2838	0.32	0.54	0.62	71.6	57.0	51.4	0.081	0.203	0.116
6	5	4	3	0	0.6	0.8	90	53.1	36.9	–	–	–
7	7	6	5	0.2	0.54	0.71	78.4	57.1	44.4	0.119	0.272	0.143
8	10	6	5	−0.65	0.89	0.93	130.5	27.1	22.3	–	–	–

We can see, that three bad orderings for all five triangles give the same sequence of metrics. Once again, we mention (above, this thing has already been said, but in connection with other facts) that this ordering differs significantly from the ordering of the badness considered for complete matrices, see the results of the calculations below and, in more detail, in [4]. However, in all our calculations (in any case, with the exception of less than 1% of all triangles, i.e., including ordering for complete matrices), such ordering turns out to be the same for all three norms (badnesses).

Another option for investigating the comparative characteristics of norms is the following one (we will continue to consider 30 species, and the matrix of distance between genomes, given in [14]). We choose two metrics and for any one fixed norm we arrange 4060 triangles in order of increasing values of this norm. At the same time, when reading that both these norms are good (that is, they give acceptable results), we should ideally obtain *identical sequences* of triangles. Actually, one of these two sequences of triangles is obtained from the other by some sequence transpositions of neighboring elements. Since, as we have noted, the number of triangles in our case is 4060, then the maximum possible number of transpositions of neighboring elements is

equal to

$$\frac{4060 \cdot 4059}{2} = 8\,239\,770.$$

Let us give concrete results of work (Table 2) for the first norm (value "bad_1") only.

Table 2. The joint study of pairs of metrics for the selected first norm (bad1).

Pairs	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{1,5\}$	$\{2,3\}$	$\{2,4\}$	$\{2,5\}$	$\{3,4\}$	$\{3,5\}$	$\{4,5\}$
Transpositions	175199	163214	169720	175267	154301	159737	160614	180413	179384	181700
Percentages	2.13	1.98	2.06	2.13	1.87	1.94	1.95	2.19	2.18	2.21
Correlation	0.957	0.960	0.959	0.957	0.963	0.961	0.961	0.956	0.956	0.956

Let us give some comments to this table. The pair of metrics is selected in the first line. The number of transpositions of neighboring elements (for obtaining the monotone sequence) is given in the second line. The percent of the maximum possible number of transpositions of neighboring elements (8 239 770) is given in the third line. We do *not* use Spearman's rank correlation coefficient (and some others correlation coefficients used in similar problems); instead of them, we use the linear function having value 1 for 0 transpositions and value -1 for 8 239 770 transpositions.

Still, we note that the two other norms give somewhat worse results; but for them, the number of transpositions does not exceed 600 000 (i.e., less than 7.5%), and, therefore, calculated by our method rank correlation coefficient is more than 0.85.

Thus, all three norms almost always give similar results. Therefore we often use the singular for the word "norm": "some different versions of so-called triangular norm" etc.

4. About one method of DNA matrix reconstruction

In this section, one of the methods of comparative analysis of various algorithms for calculating distances between DNA sequences is presented, and on the basis of this, a *method for reconstructing an incompletely filled matrix is developed.* In order to carry out this comparative analysis, we propose for the resulting algorithm to calculate the distances between genomes, consider all possible triangles, because ideally they should be acute-angled isosceles.

To answer the question of how "correct" is the matrix obtained as a result of some heuristic algorithm, we propose to use the "characteristic of the departure" of the obtained triangles from "elongated isosceles" triangles; i.e., the "badness", below we shall write this term without quotes. In this case, as one of the variants of the badness, the following formula can be used:

$$\sigma = \frac{\alpha - \beta}{\gamma} \tag{1}$$

where α, β and γ are the angles, and we admit, that $\alpha \geq \beta \geq \gamma$ [4]. In the opinion of the authors of this paper, this formula best characterizes the requirements described by us. Here is the *informal* explanation for this: The closer the triangle to the isosceles triangle, the less the difference between α and β, and in the ideal case, the numerator is 0; in accordance with our assumptions, an obtuse (or rectangular) isosceles triangle cannot be obtained. The performance of the properties of acute angles increases the denominator. Consequently, the approximation of a triangle to an isosceles triangle reduces the numerator and increases the denominator in the formula, i.e., σ tends to 0.

When calculating the badness of the entire matrix for each recovery option, we can:

- either summing the corresponding badness over all possible triangles of the matrices in question;

- or take the maximum badness for these triangles.

In the future, we propose to consider other approaches to calculating the badness of the entire matrix.

To determine the unknown element, we consider all the possible triangles formed from elements of this matrix for which one of the sides is unknown. For each such triangle, from the condition that it is isosceles acute-angled, we get *one of the possible values* of this unknown side. Next, we calculate the final value of this side (an unknown element) in a special way. Namely, for its calculation, on the basis of all the estimates obtained, the element is assumed to be equal to the arithmetic mean of all the values obtained. As an alternative, we can exclude the largest and smallest of the values obtained.

With a large number of missing elements, the matrix of the triangles with two known sides will be small, so the restoration of the matrix in one pass is usually impossible. When restoring the matrix on the second and subsequent passages, we can either use only the elements of the matrix of the last pass, or use all the matrices obtained in the previous passes. In the second case, with each successive passage in the matrix, there are more and more elements calculated approximately. Therefore, when evaluating an unknown element, it is possible to use the analog of the risk function, which will adjust the weight of the elements *depending on the pass number*.

When using the so-called *static* risk function, the weight of the elements with each pass decreases with the same coefficient, and to estimate the unknown element of the matrix, formula

$$E = \frac{c_0 E_0 + c_1 E_1 + \ldots + c_k E_k}{c_0 + c_1 + \ldots + c_k}, \tag{2}$$

where:

- E_i is the value of the matrix element, calculated on i-th pass;
- $c_0, \ldots, c_k$ are some specially chosen coefficients.

In practice (see [5]), good results are achieved when the following formulas are used for the coefficients:

$$c_0 = 1, \quad c_i = p\, c_{i-1}. \tag{3}$$

By [11, 12], the risk function can be also *dynamic*: when using it, we take averaging, depending on the "rough estimate" of the final value: whether it is "good", "middle" or "bad". Besides, we can also consider the *sequence* of such dynamic risk functions, where at each stage we rely on the value obtained in the previous step as such a "rough estimate". In our case, to evaluate the unknown element of the matrix of distances between DNA strings, the formula

$$\frac{\sum_{i=1}^{k} a_i f(a_i)}{\sum_{i=1}^{k} f(a_i)}, \tag{4}$$

is used; $f(x)$ is some specially chosen decreasing function.

5. The formal description of the algorithm

Let us consider the detailed *formal description* of the algorithm briefly considered before.

Algorithm 1 (Restoring a matrix using a static risk function)

Input: Incompletely defined matrix $A = a_{ij}$ (all elements equal to zero outside the main diagonal are assumed to be unknown).

Used auxiliary variables: b_i is the array of unknown element estimates.

Description of the algorithm.

Step 1: We set $s := 1$ (the number of the pass).

Step 2: We count h, i.e. the number of elements of the upper triangle, which are equal to 0.

Step 3:
if $a_{ij} = 0$ and $i \neq j$ then
  begin
    $kol := 0$ {We count the number of triangles,
      built on the unknown element under consideration}
    for $k := 0$ to n do begin
      if $k \neq i$ and $k \neq j$ and $a_{ki} \neq 0$ and $a_{kj} \neq 0$ then
        begin
          $kol := kol + 1$; $c_0 := 1$; $c_s := c_{s-1} \cdot p$;

$$E_{ki} := \frac{c_0 E^0_{ki} + \ldots + c_s E^s_{ki}}{c_0 + \ldots + c_s};$$

$$E_{kj} := \frac{c_0 E^0_{kj} + \ldots + c_s E^s_{kj}}{c_0 + \ldots + c_s};$$

          if $E_{ki} > E_{kj}$ then $b_{kol} := E_{ki}$ else $b_{kol} := E_{kj}$
        end;
      end;
  end;

$$a_{ij} := \frac{b_1 + \ldots + b_{kol}}{kol}.$$

Step 4: We count h_1, i.e., the number of elements of the upper triangle, which are equal to 0 after the next pass.

Step 5:
if $h_1 = 0$ then goto Output 1;
if $h_1 = h$ then goto Output 2;
$s := s + 1$;
goto Step 2.

Output 1: Filled matrix A.

Output 2: Matrix A cannot be filled.

End of description of the algorithm.

After execution of the algorithm for performing a comparative analysis of the results of the reconstruction of the matrix, we use such an indicator as the residual; it characterizes the deviation of the resulting matrix from the original one. We calculate the residual on the basis of the natural metric

$$d = \frac{\sqrt{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (a_{ij} - \widetilde{a_{ij}})^2}}{n(n-1)/2}, \tag{5}$$

where:

- $\widetilde{a_{ij}}$ are elements of the matrix obtained as a result of applying some algorithm for calculating the distances between a pair of genomes (in our case, the Needleman – Wunsch algorithm);
- a_{ij} are elements of the matrix restored as a result of the above algorithm.

Some examples of operation of the algorithm and corresponding values of residual are given in [5].

6. Conclusion

Thus, a very small change of the given matrix greatly reduces its badness. The values marked in red are chosen by us manually. However, at the present time we have a neural network that implements such an algorithm; we are going to describe this neural network in the following publication. But the following is much more important: these "red" changes give *input* information to another neural network, the one that computes the constants for the metric. Thus, along with one "loop" of algorithms already described in previous section, we get one more. Table 3 above already takes into account similar improvements in the metric.

As we said before, the received results of our computer programs are designed as an `xlsx`-file and are available for the direct download by [14]. The whole article i s actually a comment to this file.

We also note that different "places" of different metrics for different cases (i.e., the case of distant animal species, the case of close animal species and the case of subspecies) talk about the need to continue research in this direction.

Also in the near future we expect to develop other approaches for comparative analysis of various algorithms for calculating distances between sequences, as well as describe the algorithms for reconstructing matrices based on these approaches. At present, we are working on comparing two of such algorithms, both for application in the "normal" problems of DNA analysis, and in the problems of DNA matrix reconstruction close to those considered in the present paper.

7. References

[1] Melnikov B and Panin A 2012 On a parallel implementation of the multi-heuristic approach in the problem of comparison of genetic sequences *Vektor Nauki of Togliatti State University* **4(22)** 83-86 (in Russian)
[2] Makarkin S, Melnikov B and Panin A 2013 A parallel implementation of the multi-heuristic approach in the task of comparing genetic sequences *Applied Mathematics* **4(10)** 35-39 DOI: 10.4236/am.2013.410A1006
[3] Melnikov B, Pivneva S and Trifonov M 2017 Multiheuristic approach to compare the quality of defined metrics on the set of DNA sequences *Modern Information Technologies and IT Education* **13(2)** 89-96 (in Russian) DOI: 10.25559/SITITO.2017.2.235
[4] Melnikov B, Pivneva S and Trifonov M 2017 Various algorithms, calculating distances of DNA sequences, and some computational recommendations for use such algorithms *CEUR Workshop Proceedings* **1902** 43-50
[5] Melnikov B and Trenina M 2018 On a problem of the reconstruction of distance matrices between DNA sequences *International Journal of Open Information Technologies* **6** 1-13 (in Russian)
[6] Needleman S and Wunsch Ch 1970 A general method applicable to the search for similarities in the amino acid sequence of two proteins *Journal of Molecular Biology* **48(3)** 443-453
[7] Eckes B, Nischt R and Krieg T 2010 Cell-matrix interactions in dermal repair and scarring *Fibrogenesis Tissue Repair* **3(4)** 1-13 DOI: 10.1186/1755-1536-3-4
[8] Midwood K S, Williams L V and Schwarzbauer J E 2004 Tissue repair and the dynamics of the extracellular matrix *International Journal of Biochemistry & Cell Biology* **36(6)** 1031-1037
[9] Evdokimova N I and Kuznetsov A V 2017 Local patterns in the copy-move detection problem solution *Computer Optics* **41(1)** 79-87 DOI: 10.18287/2412-6179-2017-41-1-79-87
[10] Evsutin O O, Shelupanov A A, Meshcheryakov R V and Bondarenko D O 2017 An algorithm for information embedding into compressed digital images based on replacement procedures with use of optimization *Computer Optics* **41(3)** 412-421 DOI: 10.18287/2412-6179-2017-41-3-412-421
[11] Melnikov B and Radionov A 1998 On the choice of strategy in nondeterministic antagonistic games *Programming and Computer Software* **5** 55-62 (in Russian)
[12] Melnikov B 2001 Heuristics in programming of nondeterministic games *Programming and Computer Software* **5** 277-288

[13] *Nucleotide (The Nucleotide database)* (Access mode: http://www.ncbi.nlm.nih.gov/nuccore)
[14] Melnikov B *The processed results of the computer calculations* (Access mode: http://bormel.ru/BorMel-DNA.xlsx)
[15] Levenshtein V 1966 Binary codes capable of correcting deletions, insertions, and reversals *Soviet Physics Doklady* **10(8)** 707-710
[16] Winkler W 1990 String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage *Proceedings of the survey research methods sections, American Statistical Association* **4(22)** 354-359
[17] Pages H, Aboyoun P, Gentleman R and DebRaoy S *Biostrings: String objects representing biological sequences, and matching algorithms* (Access mode: https://rdrr.io/bioc/Biostrings/)
[18] Morgan M, Anders S and Lawrence M *ShortRead: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data* (Access mode: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2752612/)
[19] Van der Loo M 2014 The Stringdist Package for Approximate String Matching *R Journal* **6** 111-122
[20] Althaus E, Caprara A, Lenhof H-P and Reinert K 2006 A branch-and-cut algorithm for multiple sequence alignment *Mathematical Programming* **105** 387-425
[21] Melnikov B 2006 Multiheuristic approach to discrete optimization problems *Cybernetics and Systems Analysis* **42(3)** 335-41
[22] Makarkin S and Melnikov B 2013 Geometrical methods for solving the pseudo-geometric version of the traveling salesman problem *Stochastic optimization in informatics* **9(2)** 54-72 (in Russian)
[23] Melnikov B 2000 Once more about the state-minimization of the nondeterministic finite automata *Journal of Applied Mathematics and Computing* **7(3)** 655-662
[24] Melnikov B and Tsyganov A 2012 The state minimization problem for nondeterministic finite automata: the parallel implementation of the truncated branch and bound method *Proceedings 5th International Symposium on Parallel Architectures, Algorithms and Programming (Taipei)* 194-201

Acknowledgements
The authors of the article express their gratitude to Vladislav Dudnikov (Togliatti State University, Russia) for his help in preparing this paper. The reported study was partially supported by RFBR according to the research project No. 16-47-630829.