On possible methods for solving the problem of reconstructing the matrix of distances between DNA strings

B Melnikov[1] **and M 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. One of the tasks of biocybernetics is the problem of reconstructing the distance matrix between DNA sequences, in which not all the elements of the matrix under consideration are known at the input of the algorithm. We propose one of the methods for solving this problem, which is based on the method of comparative evaluation of the algorithms for calculating the distances between DNA strings, developed and investigated by us earlier. In this analysis, the badness of each of the triangles of the matrix is applied. 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 the so-called risk function, and the result of this averaging is considered as the received value of the unknown element. In order to optimize this method of solution, we consider the use of the branch and bound method in it. In our interpretation of this method, all possible sequences of unknown elements of the upper triangular part of the matrix are taken as the set of admissible solutions. In each current subtask, any of the blank elements of the matrix is taken as the separating element, and the sum of the badness values for all triangles that have already been formed by the time this subtask is considered is taken as the boundary. Thus, the definition of the elements of a partially filled matrix occurs in such a sequence that the final badness index for all triangles is selected using greedy heuristics that fits perfectly into the classical options for describing the branch and bound method.

1. Introduction

In practice, quite often there is a need to calculate a specially defined distance between sequences of different nature. Such algorithms are often used in bioinformatics, they are a separate, very important kind of problem of finding the distance between the given genetic sequences [1, 2]. The main difficulty that arises when calculating the distance between genetic sequences is a very long length of such a sequence.

Because of this, algorithms that calculate the exact value of the distance between two sequences are inapplicable, and to estimate the distance between such chains we have to use heuristic algorithms that give approximate results [3].

There are various similar algorithms, but their obvious disadvantage is to obtain several different results when using different heuristic algorithms applied to the calculation of the distance between the same pair of DNA strings. Therefore, there is a problem of assessing the quality of the used metrics (distances) and the results obtained in solving this problem; we can some conclusions about the applicability of a specific algorithm for calculating distances to various applied studies.

In addition, one of the problems considered in biocybernetics is the problem of restoring the distance matrix between DNA sequences (below, simply DNA matrix), in which not all the elements of the considered matrix are known at the input of the algorithm [4], [5]. In this regard, there exists another problem: to use the developed method of comparative evaluation of algorithms for calculating distances between sequences for a completely different purpose, namely, i.e., for the problem of restoring the distance matrix between DNA sequences described below.

We have considered a method of comparative analysis for the resulting work of any algorithm for calculating the distances between the genomes of the matrix is based on the consideration of all possible triangles. In this case, we assume that ideally they should be acute isosceles [6]. To answer the question, how "correct" is the matrix obtained as a result of some heuristic algorithm is, we propose to use the "departure characteristic" of the obtained triangles from the "elongated isosceles" the above value of badness.

In this approach (i.e., when the application of the method for comparative evaluation of algorithms for calculating the distance to the recovering of matrices), the recovering is the implementation of multiple computational passes. On each of the passes, for some yet empty (unknown) elements of the matrix, the different estimates are obtained; these estimates are averaged in a special way and the averaging result 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, i.e. dimensional system of bodies whose mass is given by a special risk function [7].

In addition, we consider the possibility of optimizing this algorithm, since the calculation of the unknown elements of an incomplete matrix based on the use of the value of the badness index is carried out on the basis of the fact that it should ideally be equal to zero, and the definition of the elements "left-to-right and top-down", i.e. in a strictly defined sequence, can lead to the fact that for previously defined triangles the value of the badness index will increase significantly. One form of smoothing out this situation is the use of the risk function.

Continuing to improve the algorithms for solving this problem, we consider the application of the branch and bound method in it. To do this, for some known sequence of empty elements, we apply the algorithms described above; however, the sequence we select using a specially developed variant of the branch and bound method.

In our interpretation of this method, we perform the following actions.

- ✓ All possible sequences of unknown elements of the upper triangular part of the matrix are taken as the set of feasible solutions.
- ✓ In each current subproblem, the matrix elements that have not yet been filled are taken as possible separating elements.
- ✓ The boundary is the sum of the badness values for all triangles that have already formed by the time this subproblem is considered.
- ✓ An auxiliary algorithm for selecting the separating element is that among the possible separating elements we choose one for which the boundary between the formed right and left subproblems is obtained in the most possible way.

Thus, the determination of the elements of an incomplete matrix occurs in such a sequence, in which the final value of the badness for all triangles is selected using a greedy heuristics, which fully fits into the framework of the classical variants of the description of the branch and bound method.

2. Preliminaries

This section presents one of the methods of comparative analysis of different algorithms for calculating the distances between DNA sequences, and on its basis, a method for restoring an incomplete matrix is developed.

In order to carry out this comparative analysis, we propose to consider all possible triangles for the resulting algorithm for calculating the distances between genomes, because ideally they should be acute 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 the "elongated isosceles" triangles, i.e., the index of badness.

The formula can be used to calculate the badness index

$$\sigma = \frac{\alpha-\beta}{\alpha}, \quad (1)$$

here, α, β and γ are the angles of the triangle, and we assume that α ≥ β ≥ γ [4].

The sides can also be used in the formula to calculate the value of badness instead of the angles of the triangle:

$$\sigma = \frac{a-b}{a}, \quad (2)$$

where a, b, c side of triangle, and a≥b≥c. [6].

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

- either summing the corresponding badness over all possible triangles of the considered matrices;
- or take the maximum badness for these triangles.

However, when calculating this indicator (the badness of the whole matrix), there may be some (in practice, very small) number of triangles for which the value of badness may differ significantly from others. In particular, you can get triangles where the value of badness is 1.

Based on all this, we use a special averaging to calculate the value of the badness of the whole matrix. 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 given by a special function of the so-called risk function [7]. Badness for all triangles determines the coordinates of bodies, and the risk function of their mass, while the larger the coordinate, the less its mass (i.e., the greater the value of badness, the less its contribution).

So, as already noted above, we believe that in a properly filled matrix of distances between DNA sequences, all possible constructed triangles should be as close to the isosceles sharp-angled, and then on the basis of this conclusion it is possible to restore the matrix of distances between the rows of DNA, which first has a number of unknown elements. We will continue to fill in such matrices.

3. Strictly description of the recovering algorithm

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, given the condition that it is an isosceles acute-angled triangle, we obtain one of the possible values of this unknown side.

Next, we calculate the final value of this side (unknown element) in a special way.

Namely, to calculate it on the basis of all the estimates obtained, the element is assumed to be equal to the arithmetic mean of all the values obtained; or, as an alternative, we can exclude the largest and the smallest of the values obtained.

With a large number of missing elements of the triangle matrix 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 passes, you can either use only the elements of the matrix of the last pass, or use all the matrices obtained on the previous passes.

In the second case, with each subsequent passage in the matrix, there are more elements calculated approximately. Therefore, when evaluating an unknown element, it is possible to use an analogue of the risk function, which will adjust the weight of the elements depending on the number of the passage.

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

$$E = \frac{c_0E_0+c_1E_1+\cdots+c_kE_k}{c_0+c_1+\cdots+c_k}, \quad (3)$$

where: E_i are the values of elements of the matrix, received on i-th pass; c_0, …, c_k are some specially selected coefficients. In practice [8], the good results are achieved when the next formula for coefficients is used: $c_0 = 1$, $c_i = p \cdot c_{i-1}$.

According to [7], the risk function can be *dynamic*: using the last one, we take averaging, depending on the “rough estimate” of the final value: whether it is “good”, “average” or “bad’. Also, we can consider a *sequence* of dynamic risk functions, where at each stage, we rely on the value obtained in the previous step. In our case, to evaluate the unknown element of the matrix of distances between DNA strings, we use the formula

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

where f(x) is a special manner selected decreasing function.

Algorithm 1 *(Restoring the DNA matrix using a static risk function)*

Input: Incomplete definite matrix $= a_{ij}$.

The following *auxiliary variables* are used: b_i is the array of unknown item ratings.

Description of the algorithm.

```
Step 1: Set s : = 1; it is the number of the pass.
Step 2: Calculate h, i.e. the number of elements of the top triangle, equal to 0.
Step 3:
if $a_{ij}$ = 0 and i≠j then begin
  count := 0; {we consider the number of triangles built on an unknown element}
  for k := 0 to n do begin
    if k ≠ I and k ≠ j and $a_{ki}$ ≠ 0 and $a_{kj}$ ≠ 0 then begin
    count := count + 1; $c_0$ := 1; $c_s$:= $c_{s-1}$·p;
    $E_{ki} = \frac{c_0E_{ki}^0+c_1E_{ki}^1+\dots+c_kE_{ki}^s}{c_0+c_1+\dots+c_s}$; $E_{kj} = \frac{c_0E_{kj}^0+c_1E_{kj}^1+\dots+c_kE_{kj}^s}{c_0+c_1+\dots+c_s}$;
    if $E_{ki}>E_{kj}$ then $b_{kol}:=E_{ki}$ else $b_{kol}:=E_{kj}$.
   end;
  end;
end;
$a_{ij} = \frac{b_0+\dots b_{kol}}{kol}$.
Step 4: Calculate $h_1$, i.e. the number of elements of the top triangle, equal to 0 after the next pass.
Step 5:
if $h_1$ = 0 then output 1;
if $h_1$ = h then output 2;
s : = s + 1;
go to step 2.
Output 1: The filled (restored) matrix A.
Output 2: The matrix A cannot be restored. □
```

For performing a comparative analysis of the results of the reconstruction of the matrix (after the execution of the algorithm), we use such an indicator as the discrepancy, it characterizes the deviation of the resulting matrix from the original matrix. Namely, we calculate the discrepancy on the basis of the natural metric.

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

where: $\tilde{\alpha}_{ij}$ 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, for Needleman-Wunsch algorithm); a_{ij} are the elements of the matrix, restored at the result of the above algorithm.

4. The branch and bound method in the problem of recovering DNA matrix

The branch and bound method is designed to develop algorithms for solving optimization problems, and we consider its interpretation close to [9]. It refers to the method of developing algorithms, known as *backtracking programming* and explores a tree-like model of the space of feasible solutions. The purpose of applying the branch and bound method in our case, i.e. to restore the matrix, is finding the optimal *sequence* of calculation of unknown elements; in the end of such calculation, the discrepancy would be minimal (or close to the minimum, i.e., pseudo-minimal). Minimizing the residual is directly related to minimizing the badness value of the entire matrix, so the problem of applying the branch and bound method for matrix recovering is to minimize the badness value of the entire matrix.

Always when using the branch and bound method, the following auxiliary algorithms should be described:

- the branching procedure for the set of feasible solutions;
- the procedure for finding the lower and upper bounds of the goal function.

In the problem we are solving, the space of admissible solutions is all possible sequences for determining the unknown elements of the upper half of the matrix. Branching occurs according to the separating elements known from the very beginning of the algorithm operation, which in our case coincide with the unfilled elements of the matrix. Branching on the set of admissible solutions will be performed as follows: for the selected element, all the sequences are divided into two sequences:

- starting with this element;
- and not starting with it.

One of the important auxiliary heuristics is the heuristic for the selection of the separating element a_{ij} $(i < j)$. It consists in:

- to find all possible k, for which a_{ki} and a_{kj} are known, $k \neq i$ and $k \neq j$;
- to calculate the difference

$$|a_{ki} - a_{kj}|. \tag{6}$$

We first select the elements for which k takes the largest value, and then among them, we choose the one for which the difference (6) is greatest; we consider this element as the separating one.

The boundaries, as noted above, are formed as a sum of the badness values for all triangles that are already formed in the matrix. It follows that the filling of the matrices with new elements leads to an increase (or rather, not a decrease) of this value.

Given the large number of triangles formed and for clarity of calculations, we move from the index of badness for the triangle to the same indicator, but for the elements of the matrix. For each element above the main diagonal, we consider all the triangles built on it and calculate the value of the exponent σ of each triangle. And the elementa_{ij} $(i < j)$ put in the corresponding value

$$\sigma_{ij} = \max\nolimits_{1 \le k \le n} \sigma_k, \tag{7}$$

where σ_k is the badness of a triangle built on elements a_{ij}, a_{ki}, a_{kj}.

For the entire DNA matrix, we determine the sum of the values of badness for all elements, i.e.

$$\sigma = \sum_{i=1}^{n-1} \sum_{j=j+1}^{n} \sigma_{ij}. \tag{8}$$

The goal of matrix reconstruction is to minimize the index (8).

The developed algorithm 1 in a slightly modified form will be used as an auxiliary for the algorithm for restoring the incomplete matrix of distances between DNA chains by the branch and bound method.

Algorithm 2 (Auxiliary algorithm for assigning a value to the selected element)

Input: 1. Incomplete definite matrix $= a_{ij}$.

2. Selected unknown item a_{ij}.

Description of the algorithm.

```
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
        if $a_{ki}$> $a_{kj}$ then $a_{ij}$:= $a_{ij}$+ $a_{ki}$
        else $a_{ij}$:= $a_{ij}$+ $a_{kj}$
end;
$a_{ij} \coloneqq a_{ij}/kol_{ij}$.
```

Output: Value of the element a_{ij}. □

Algorithm 3. (Recovering matrix by the branch and bound method)

Input: 1. Incomplete definite matrix $= a_{ij}$.

Description of the algorithm.

Step 1: Initialization: "zeroing" of the list of subproblems and the current pseudo-optimal solutions.

Step 2: Adding to the list of subproblems of the problem corresponding to the source matrix.

Step 3: If the list of subproblems is empty, output 1.

Step 4: If the algorithm time is over, then output 1.

Step 5: Select a new subproblem from the list of subproblems and delete it.

Step 6: The ``best`` of the unknown element.

```
1) if $a_{ij} = 0$ then
    begin $kol_{ij} \coloneqq 0$ { consider the number of triangles, built on an unknown element }
    $mas\_m_{ij} \coloneqq 0$ { calculate the maximum difference }
    for $k{:}= 0$ to $n-1$ do begin
        if $k \neq i$ and $k \neq j$ and $a_{ki} \neq 0$ and $a_{kj} \neq 0$ then
          begin
            $kol{:}= kol + 1$; m:=| $a_{ki}$- $a_{kj}$|;
            if $m > mas\_m_{ij}$ then $mas\_m_{ij} \coloneqq m$;
          end;
    end;
```

2) Determine the largest value of the array kol_{ij}: $maxk = \max_{\substack{1 \le i \le n \\ 1 \le j \le n}} kol_{ij}$.

3) If $maxk$=0, then output 2.

4) Among the elements for which $kol_{ij} = maxk$, we find the one with the largest mas_m_{ij} value.

Step 7: Call algorithm 2 to calculate the selected item.

Step 8: Calculate the value of the badness metric for it:

```
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
```

```
            sort in descending order$a_{ki}$, $a_{kj}$ и $a_{ij}$
            and denote them accordingly $a, b, c$;
        $\sigma_k = \frac{a-b}{c}$;
    end;
    $\sigma_{ij} := \max_{1 \le k \le n} \sigma_k$
```

Step 9: The formation of the left and the right subproblems based on the comparison of the values of the indicator of badness for the selected item. The left one is the one for which this value is less.

Step 10: Check if the left problem is valid, otherwise go back to step 6.

Step 11: If the value of the badness index of the obtained matrix for the left subproblem is less than the corresponding value of the current pseudo-optimal solution, this subproblem becomes the current one and go to step 2. Otherwise go to step 5.

Output: *1:* Current pseudo-optimal solution (if any).

Output 2: The matrix A cannot be restored.□

5. Some results of computational experiment

This section presents the results of a computational experiment. To carry out the computational experiment, we took, obtained as a result of the application of the Needleman-Wunsch algorithm [10]. This algorithm was applied to the mdNA chains of various animals taken from the NCBI databank [11]; while sequencing of mRNA strands was made for one representative of each of the 28 mammalian orders, since MDCs in the different of these 28 species change only due to a mutation, because they are not recombinant and inherited only on the maternal line (we select the classification of mammals according to [12], we do not consider other variants of classification). The results of applying the algorithms we are considering (in particular, the Needleman-Wunsch algorithm) are usually expressed in "percentages of proximity". According to our previous works [6], [13], etc., we need a similar characteristic ("relative distance")‒obtained by subtracting the obtained "percentage of proximity" from 100 and dividing by 100, the data for convenience in the tables will be denoted by integers formed by the first three significant digits of these values.

So, in the original distance matrix (Table 1 of the application) we removed about 63% of the element pairs (left about 37% of the pairs; we remove some element of the upper triangle together with the corresponding element of the lower one). The resulting matrix is shown in Table 2 of the Appendix. The Appendix is available here [14].

For the comparative analysis, we first made the restoration of the matrix using only the elements of the matrix that was formed on the last pass. As noted in the previous section, two approaches are possible. The results of the restoration using the first approach, where the arithmetic mean was calculated by all estimates obtained from all the regular triangles constructed on this element with two other known sides, are presented in Table 3 of the Appendix. At Table 4 of the Appendix, the matrix obtained using the second approach is given: from all the plurality of the estimates, if some are allowed, we excluded the largest and the smallest elements, and the remaining was calculated arithmetic mean.

On the basis of the analysis of Table 4, obtained by applying the second approach, it can be concluded that this approach is insufficient for the restoration of the DNA matrix. Moreover, this inadequate efficiency is manifested, despite the fact that this approach should give a relatively better option: in fact, we exclude the “extreme situations”. However, in matrices of large dimensions, a large number of identical elements are obtained; this fact explained, that an unknown element is formed small number of triangles with two other known sides, so the exception highest and lowest estimates leads to the fact, that the calculation of the average made for a very small number of estimates.

The number of passes necessary to restore the entire matrix depends on the percentage of missing elements. As the results of computational experiments showed, if the percentage of missing elements is less than about 55%, then the entire matrix is restored in 1 pass. If this number exceeds about 64%, it usually required more than 2 passes are. In addition, in this case the number of passes will depend on the location of the missing elements. When absence of all the elements of some row (column), the restoration of the matrix is completely impossible, and therefore, with increasing the percentage of zeroing the matrix, the probability of its restoration certainly decreases.

To perform a comparative analysis of the various methods of matrix reconstruction, we calculated the discrepancy, and also identified the greatest deviation. The results of the calculations are presented in table 1.

Table 1. Comparison of discrepancies
in the different approaches to matrix reconstruction.

	1st approach		2nd approach	
	max d_{ij}	*d*	*max* d_{ij}	*d*
1st pass	0.214	0.00194	0.214	0,00332
2nd pass	0.236	0,00279	0.350	0,00394

The use of the second approach on the first pass gives a smaller value of the maximum deviation, but, in general, the residual is greater, and the greatest deviation of this residual from the residual obtained for the first approach occurs at the second iteration, when the number of triangles with two other known vertices becomes larger. Thus, in calculating the arithmetic mean of all the “preliminary values” of the element, the value of the residual is much smaller.

The results are obtained by reconstructing the matrix using only the elements of the matrix of the last pass. However, in this case, the larger the pass number, the less accurate the matrix elements.

Next, we present the results obtained by applying both static and dynamic risk functions. For the static risk function, the best result was obtained for the coefficient p = 0,9. And for the dynamic risk function we chose a decreasing function

$$f(x) = 1 - \sqrt{0{,}1x} \quad (9)$$

The use of risk functions has made it possible to reduce the significance of the discrepancy, especially with the second and further passes.

Table 2. Comparison of the discrepancy recovering matrix
with the use of static and dynamic risk functions.

	Restoration of the matrix using a static risk function, $p = 0{,}95$		Recovering the matrix using a dynamic risk function $f(x) = 1 - \sqrt{0{,}1x}$.	
	max d_{ij}	*d*	*max* d_{ij}	*d*
1st pass	0, 1852	0.001715	0.1414	0.001689
2nd pass	0.1852	0.001851	0.1414	0.001801

Thus, the use of the risk function in restoring the matrices makes it possible to reduce the value of the discrepancy, especially with a large number of passes. Tables 5 and 6 of the Appendix show the recovered matrices using the static and dynamic risk functions of the matrix.

The summary table 3 shows the number of passes that were required in the computational experiment for a different percentage of the excluded matrix elements, as well as the residual values after the matrix reconstruction, using the risk function and without it.

Table 3. The summary table of calculations.

%	Number of passes	Matrix recovery based only on the matrix of the last pass		Restoration of the matrix using a static risk function, p = 0,95		Recovering the matrix using a dynamic risk function, $f(x) = 1 - \sqrt{0,1x}$.	
		max d_{ij}	*d*	*max* d_{ij}	*d*	*max* d_{ij}	*d*
50	1	0.193	0.00198	0,186	0.00173	0,158	0,00168
62	2	0.214	0,00279	0,185	0.00185	0,141	0,00180
65	3	0.203	0.00293	0,175	0,00203	0.156	0,00200

Further, we recovered the same matrix by the branch and bound method. The resulting matrix is presented in Table 7 of the Appendix.

When restoring matrix algorithm 1 the value of the indicator of badness of the matrix equal to σ=57,7, and the value of the discrepancy amounted to d=0,00185, and when the restoration branch and bound method σ=45,3 and d=0,00145.

Thus, as a result of applying the branch and bound method, the value of the indicator of badness of the matrix has decreased by 21%, the residual error decreased by 20%. A comparison of the results of different methods for the reconstruction of the matrix of distances between DNA chains is given in table 4.

It is important to note that a significant improvement in the main characteristics of the matrix recovering practically did not affect the increase in the time of the algorithm: it increased only by about 10%.

Table 4. Comparison of the discrepancy of reconstruction of one matrix using the branch and bound method and without it.

	σ	*max* d_{ij}	*d*
Without the branch and bound method	57,7	0,141	0,00185
branch and bound method	45,3	0,106	0,00145
Improvement, %	21,4	24,8	21,6

6. Conclusions

So, in the basis of the proposed method of reconstruction of the matrix of distances between DNA sequences, we propose to use an approach that was previously developed and applied in practice for the comparative evaluation of other algorithms of algorithms for calculating distances between such sequences; simplifying, we can say that we are trying to achieve the property of acute isosceles for all the resulting triangles.

Application of the described method to fill the matrix of distances between DNA sequences will significantly reduce the time of its filling: for example, to build a matrix of the order of 50×50, which recorded the distance calculated by the algorithm Needlman-Wunsch, it takes about 28 hours, and when using the proposed method about 1 hour.

At the same time, the application of the branch and bound method made it possible to determine the unknown elements of the matrix in the sequence that minimizes the value of the matrix badness index and, as a consequence, the discrepancy.

As one of the directions of the future improvement of the heuristic algorithm described here (first of all, reducing the time of its operation), we assume the use of clustering of situations, the application of which is described in [15]. In applying this approach, we try to select the same separating element in different subproblems (situations) obtained on different branches of the tree of the branch and bound method, which (subproblems) correspond to some close ones by some "natural" matrix metric. Such use (if possible) requires virtually no time despite the fact that the time spent on the choice of element is most of the time work branch and bound method.

7. References

[1] Toppi J, De VicoFallani F, Petti M, Vecchiato G, Maglione A G, Cincotti F, Salinari S, Mattia D, Babiloni F and Astolfi L 2013 A new statistical approach for the extraction of adjacency matrix from effective connectivity networks *IEEE Engineering in Medicine and Biology Society (EMBC)* No **3-7** pp 2932-2935

[2] Van der Loo M P J 2014 The Stringdist Package for Approximate String Matching *The R Journal* vol. **6** pp 111-122

[3] Melnikov B, Radionov A and Gumayunov V 2006 Some special heuristics for discrete optimization problems *In: Proceedings of 8th International Conference on Enterprise Information Systems, ICEIS-2006. Paphos* pp 360-364

[4] Eckes B, Nischt R and Krieg T 2010 Cell-matrix interactions in dermal repair and scarring *Fibrogenesis Tissue Repair.* No.**3:4**. doi:10.1186/1755-1536-3-4

[5] Midwood K S, Williams L V and Schwarzbauer J E 2004 Tissue repair and the dynamics of the extracellular matrix *The International Journal of Biochemistry & Cell Biology* Vol. **36** Issue 6 pp 1031-1037

[6] Melnikov B, Pivneva S and Trifonov M 2017 Comparative analysis of the algorithms of calculating distances of DNA sequences and some related problems *Proceedings of the III International Conference "Information technology and nanotechnology (ITNT-2017)"* pp 1640–1645

[7] Melnikov B 2001 Heuristics in programming of nondeterministic games. *Programming and Computer Software* Vol **27** No. **5** pp 277-288

[8] Melnikov B and Melnikova E 2013 Approach to programming nondeterministic GAMES (Part I: Description of general heuristics) *Proceedings of higher educational institutions. Volga Region. Physics and mathematics.* No **4(28)** pp 29-38 (In Russian)

[9] Hromkovic J 2003 *Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics* Springer p 538

[10] Melnikov B, Trenina M and Kochergin A 2018 An approach to improving algorithms for calculating distances between DNA strings (using the example of the Needlman-Wunsch algorithm *Proceedings of higher educational institutions. Volga Region. Physics and mathematics* No **1(45**) pp 46–59 doi 10.21685/2072-3040-2018-1-4 (In Russian)

[11] (2014) NCBI:nucleotidedatabase,availableat:http://www.ncbi.nlm.nih.gov/nuccore.

[12] Ayala F and Kayger J 1980 *Modern genetics V. 1* Menlo Park Calif.(USA) Benjamin/Cummings Pub p 295

[13] 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* Vol. **1902** 4347. (doi: 10.18287/1613-0073-2017-1902-43-50)

[14] Ya.Drive – [Electron. resource]. – Access mode: https://yadi.sk/i/JxAfi8jm3aEtEY, free

[15] Melnikov B, Radionov A, Moseev A and Melnikova E 2006 Some specific heuristics for situation clustering problems *ICSOFT, Technologies, Proceedings 1st International Conference on Software and Data Technologies* pp 272-279