Scientific Research Publishing

Journal of Biosciences and Medicines, 2023, 11, 310-320
https://www.scirp.org/journal/jbm
ISSN Online: 2327-509X
ISSN Print: 2327-5081

An Algorithm for the Inverse Problem of Matrix Processing: DNA Chains, Their Distance Matrices and Reconstructing

Boris F. Melnikov[1], Ye Zhang[2], Dmitrii Chaikovskii[1]*

[1]Faculty of Computational Mathematics and Cybernetics, Shenzhen MSU-BIT University, Shenzhen, China
[2]School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, China
Email: bormel@mail.ru, ye.zhang@bit.edu.cn, *mitichya@yandex.ru

How to cite this paper: Melnikov, B.F., Zhang, Y. and Chaikovskii, D. (2023) An Algorithm for the Inverse Problem of Matrix Processing: DNA Chains, Their Distance Matrices and Reconstructing. *Journal of Biosciences and Medicines*, **11**, 310-320.
https://doi.org/10.4236/jbm.2023.115023

Received: March 6, 2023
Accepted: May 28, 2023
Published: May 31, 2023

Abstract

We continue to consider one of the cybernetic methods in biology related to the study of DNA chains. Exactly, we are considering the problem of reconstructing the distance matrix for DNA chains. Such a matrix is formed on the basis of any of the possible algorithms for determining the distances between DNA chains, as well as any specific object of study. At the same time, for example, the practical programming results show that on an average modern computer, it takes about a day to build such a 30 × 30 matrix for mnDNAs using the Needleman-Wunsch algorithm; therefore, for such a 300 × 300 matrix, about 3 months of continuous computer operation is expected. Thus, even for a relatively small number of species, calculating the distance matrix on conventional computers is hardly feasible and the supercomputers are usually not available. Therefore, we started publishing our variants of the algorithms for calculating the distance between two DNA chains, then we publish algorithms for restoring partially filled matrices, *i.e.*, the inverse problem of matrix processing. Previously, we used the method of branches and boundaries, but in this paper we propose to use another new algorithm for restoring the distance matrix for DNA chains. Our recent work has shown that even greater improvement in the quality of the algorithm can often be achieved without improving the auxiliary heuristics of the branches and boundaries method. Thus, we are improving the algorithms that formulate the greedy function of this method only.

Keywords

DNA Chains, Distance Matrix, Optimization Problem, Restoring Algorithm, Greedy Algorithm, Heuristics

1. Introduction

In this paper, we continue to consider one of the cybernetic methods in biology related to the study of DNA chains. Namely, we are considering one of the important tasks of this topic, *i.e.*, the problem of reconstructing the distance matrix for DNA chains. In this case, the distance matrix is formed on the basis of any of the possible algorithms for determining the distances between DNA chains, as well as any specific object of study.

Let us firstly remark, that the total length of the human genome exceeds 3 × 10^9 characters, which is about 200,000 times longer than mtDNA (see below). This fact indirectly confirms the need to apply heuristics when considering DNA algorithms.

It is important to note that currently it is easy to find only a few similar algorithms on the Internet, [1] [2] [3] etc.; see also the description of our algorithm in [4] and some of our other papers cited there. The objects of research of these algorithms (for mammals) are, as a rule, one of the following 3 variants: the mitochondrial DNA (mtDNA); "the tail" of Y chromosome; the main histocompatibility complex (MHC) [5] [6] [7] [8].

However, such a small number of variants (less than 10 algorithms and 3 objects of research) does not negate the need to create effective algorithms for processing DNA chains, in particular, constructing (for one of these variants) a matrix of distances between such chains. At the same time, for example, the practical programming results show that on an average modern computer, it takes about a day to build such a 30 × 30 matrix for mnDNAs using the Needleman-Wunsch algorithm [1]; therefore, for such a 300 × 300 matrix, about 3 months of continuous computer operation is expected. Such dimensions come from real problems: for example, in the class of mammals there are about 30 orders, in the order of primates there are about 20 families, more than 80 genera and more than 500 species. At the same time, the exact values differ in different classification options, but they are not interesting to us: we are interested in approximate values only.

Thus, even for a relatively small number of species (smaller than the total number of primate species), calculating the distance matrix on conventional computers is hardly feasible; the use of supercomputers, firstly, is not always possible, and, secondly, it often requires significant revision of existing software. In this regard, the task arises of restoring such a partially filled matrix. We started publishing our variants of similar algorithms for restoring partially filled matrices in [4] (the simplest algorithm was described very briefly there), after which we returned to this problem in [9] [10] [11], where a variant of the algorithm using the method of branches and boundaries was described in detail.

In this paper, we revisit algorithm variants that do not rely on the branches and boundaries method, despite prior successful results. Our recent research, particularly in graph theory and ultra-large communication networks, has shown that improving auxiliary heuristics of this method may not always lead to significant algorithm quality improvements (see [12]). Instead, we focus solely on enhancing the greedy function formulation of the method's algorithms, which we consider auxiliary.

Specifically, we describe an improvement to the greedy algorithm for reconstructing the distance matrix between DNA chains. Our computational experiments (see Section 5) demonstrate that this approach yields better results than simpler variants of the branches and boundaries method. It is important to note that although we previously used the branches and boundaries method for the inverse problem of matrix processing, *i.e.*, restoring partially filled matrices, we do not utilize it in this paper.

In connection with the above, the question arises about the concept of "partially filled matrix": how to determine this partial filling. It is clear that the greater the percentage of values will not be calculated, the less time will be spent on these calculations: after all, as follows from the above estimates, the calculation of one value for two considered mtDNAs requires about 3 minutes of computer operation, which is approximately equal to the total time required for matrix recovery even using the long-running method of branches and boundaries. On the other hand, a too small percentage of the values left in the matrix (*i.e.*, calculated by the special previous algorithms), of course, cannot give adequate results; in this regard, we have been using in this work the percentage of values calculated by the algorithm of about 10% - 12%.

The second important question that cannot but arise on the basis of the above text is how exactly we can analyze the quality of the solution obtained using the recovery algorithm(s). For more information, see Section 5 below.

2. A Brief Description of the Greedy Algorithms of Restoring the Distance Matrix

It is clear that with smaller dimensions of the matrix, a larger percentage of non-deleted elements is required. Thus, for small dimensions (of the order of 10), algorithms often do not work with a small number of nonremovable elements. Of course, in principle, there are options when, under the conditions we have given (*i.e.*, about 10% of non-removable elements with matrix dimensions of about 30), it is impossible to restore the matrix: for example, when an empty line is obtained.

We considered the simplest heuristic for filling in the distance matrix without using the method of branches and boundaries in [9]. (We note in advance that the results of the computational experiments given in that paper will be compared below with newer results using other heuristic algorithms.) Further, as we have already noted, our publications were devoted to the application of the branches and boundaries method; but in this paper, we again abandon it. At the same time, we complicate the greedy heuristics of [9].

Let us briefly describe the greedy matrix filling algorithm used in this paper. First of all, we choose an element that, if filled in, forms the largest number of newly formed triangles; *i.e.*, for $n \times n$-dimensional distance matrix $\left(m_{i,j}\right)$, we choose the pair (i, j) such that: $m_{i,j} < 0$ and

$$\max_{1 \le i,j \le n, i \ne j} \sum_{1 \le k \le n, k \ne l, k \ne j} \left(\operatorname{sgn}\left(m_{k,j}\right) + \operatorname{sgn}\left(m_{k,j}\right)\right) \qquad (1)$$

is achieved.

If there are several such elements, we choose any of them. Next, we consider all the resulting triangles, and minimize the total value of the badness.

One of the variants of such an assessment of badness for one triangle is the formula

$$\frac{\alpha - \beta}{\gamma} \qquad (2)$$

where α, β and γ are the angles of the derived triangle, and $\alpha \ge \beta \ge \gamma$. If the three sides do not satisfy the triangle inequality, we assumed a large value as the value of badness, usually from 1.0 for the case $a = b + c$ to 2.0 for "absolutely impossible" triangles.

The total value of badness is always (*i.e.*, both for choosing a value in the described algorithm and for a posteriori evaluation of the quality of the algorithm) considered simply as the sum of the values of badness of all triangles. In the described algorithm, we are trying to minimize this badness value for all newly formed triangles.

The minimization method is given in the next section, where the justification for the possibility of piecemeal filling of the matrix is given, to obtain a value of badness close to optimal (*i.e.*, in terminology of [13], "to obtain a pseudo-optimal solution"). Simplifying it, we can say that that, taking the average values of the maximum sides of the formed triangles (*i.e.*, a in previous formulas; note that in [9], this value was counted final) as the beginning of the iterative process, we get a pseudo-optimal value in a few iterations.

3. The Theoretical Substantiation of the Possibility of Improvement of the Greedy Algorithm

Thus, in this paper we reconstruct the matrix of distances between DNA sequences of different species of organisms. We shall restore the distance function of the matrix and find its derivative. The main problem is that it is an ill-posed problem, which means that a small error in the source data can lead to a large error in the calculated derivatives [14] [15] [16] [17], etc.

If we introduce the coordinate axes x and y, located in the horizontal and vertical axes, respectively, and consider the matrix as a two-dimensional array with noisy data defined on the domain $\Omega \le R^2$, then the matrix elements will represent the noisy values of the function of two variables $u_{i,j}^{\delta}$. It is natural to assume that the domain Ω is divided into $N \le n^2$ parts $\left\{\Omega_i\right\}_{i=1}^{N}$, and there is the only one value $u_{i,j}^{\delta}$ in each part. Denote d_i as the diameter of Ω_i and let $d = \max\left\{d_i\right\}$.

In this case, we obtain the deterministic model

$$\max\left|u\left(x_i,y_i\right)-u_{i,j}^{\delta}\right|\le\sigma, \tag{4}$$

between the noisy data $u_{i,j}^{\delta}$ and the corresponding exact values $\left\{u\left(x_i,y_i\right)\right\}$ at grid points $X_n:=\left\{x_1<x_2<\ldots<x_n\right\}$ and $Y_n:=\left\{y_1<y_2<\ldots<y_n\right\}$.

Let us suppose that the reconstructed function $u^{\epsilon}\in(x,y)$ is formed according to the following optimization problem:

$$\begin{aligned}u^{\epsilon}=\arg\min_{v\in C^1(\Omega)}&\left(\frac{1}{n^2}\cdot\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n}\left(v\left(x_i,y_j\right)-u_{i,j}^{\delta}\right)^2\right.\\&\left.+\epsilon\left(\left\|\frac{\partial^2 v(x,y)}{\partial x^2}\right\|_{L^2(\Omega)}^2+\left\|\frac{\partial^2 v(x,y)}{\partial y^2}\right\|_{L^2(\Omega)}^2\right)\right),\end{aligned} \tag{5}$$

where $v(x,y)$ is a cubic spline, and regularization parameter ϵ satisfies the expression:

$$\frac{1}{n^2}\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}\left(v\left(x_i,y_j\right)-u_{i,j}^{\delta}\right)^2=\delta^4. \tag{6}$$

Then by ([18], Theorem 3.3), the following proposition holds.

Proposition 1. Let $u(\cdot,\cdot)\in H^2(\Omega)$. Let $u^{\epsilon}(x,y)$ be the minimizer of the problem (1). Then for $\epsilon=\sigma^2$, we have

$$\left\|u^{\epsilon}(\cdot,\cdot)-u(\cdot,\cdot)\right\|_{H^1(\Omega)}\le C_1 d^{1/4}+C^2\sqrt{\delta}, \tag{7}$$

where C^1 and C^2 are some constants depending on the area Ω and on

$$\left\|\Delta u(x,y)\right\|_{L^2(\Omega)}.$$

Based on Proposition 1, we obtain the following fact. For sufficiently small values d and δ, after solving the optimization problem (1) values

$$\left\{\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right\}$$

can also be found with sufficiently high accuracy. We can do it taking derivatives

$$\left\{\frac{\partial u^{\epsilon}}{\partial x},\frac{\partial u^{\epsilon}}{\partial y}\right\}.$$

Due to the fact that with an increase in the value of the norm $\left\|\Delta u(x,y)\right\|_{L^2(\Omega)}$ the value of constants C_1 and C_2 will increase, we conclude that the smoother the function $u(x,y)$ is, the smaller this norm will be, and the more accurate the regularization result will be. If the function is not smooth enough, we shall need noisier data to obtain the necessary accuracy.

It also follows from the proposition 1 that if we set $\delta\to 0$, the error of restoring the function will depend mainly on the diameter d, which is the larger, the more missing data $\left\{u_{i,j}^{\delta}\right\}$ at the grid points. And due to the fact, that 65% of data is missing in the task we have set, we need to introduce additional conditions to restore the function. One of such conditions is the regularity found in the paper [4] for distance matrices, which consists in the fact that the three elements of the matrix

$$\left(m_{i,j}, m_{k,i}, m_{k,j}\right)$$

form the sides of an isosceles triangle. Thus, the formulas below reduce such a metric to a function of several variables, and a "triangular" norm for determining the quality of the distance metric can be introduced, which can be represented in the following way.

For the matrix $M = \left(m_{i,j}\right)$ and its elements $m_{i,j}$, we always suppose that $i, j, k \in \{1, 2, \ldots, n\}$ and do not consider diagonal elements (*i.e.*, elements $m_{i,i}$ and the arithmetical expressions with these elements are ignored in formulas). The total error σ is defined as follows:

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

where one of the calculation variants is the sequential usage of the following formulas:

$$\begin{aligned}
&-r_{i,j,k}^{(1)} = \max\left(m_{i,j}, m_{k,i}, m_{k,j}\right),\\
&-r_{i,j,k}^{(2)} = \min\left(m_{i,j}, m_{k,i}, m_{k,j}\right),\\
&-\text{ and } \sigma_{i,j} \text{ is as follows:}\\
&\max_{1\le k\le n k\ne i, k\ne j} \frac{2r_{i,j,k}^{(1)} + r_{i,j,k}^{(2)} - m_{i,j} - m_{k,i} - m_{k,j}}{r_{i,j,k}^{(2)}}.
\end{aligned} \tag{9}$$

Then the original problem can be reformulated into the problem of minimizing the error value (as we already said, it often was called "badness" in our previous papers) by piecemeal filling in the missing elements.

It is very important that we fill missing elements in the table sequentially, piecemeal, "step by step"; thereby we greatly simplify the implementation of the corresponding algorithm.

Filling the table in this way, we obtain a matrix with noisy data $\left\{u_{i,j}^{\delta}\right\}$ and then we restore the u^{ϵ} function by solving (1). The level of noise δ generated by this algorithm for restoring missing values can be estimated by analyzing the results of violations of the "isosceles triangle" regularity in [4].

Thus, by sequentially filling in the missing elements of the matrix, we can guarantee a consistent improvement of the resulting solution, which theoretically justifies the possibility of abandoning the branch and boundary method, which works much longer than the greedy algorithm for obtaining the value of one element considered here.

4. Quality Criteria for the Numerical Solution of the Problem

As we said before, an important question arises, is how exactly can we analyze the quality of the solution obtained using the recovery algorithm(s).

However, the above model of calculations does not give a complete answer to the question of the quality of matrix restoring.

Therefore, the simplest quality criterion would be a comparison (by some natural metric) of the reconstructed matrix and the actually obtained distance matrix, which we can obtain for some examples of a small dimension; however, it is obvious that such a comparison can be made only a limited number of times, probably during the initial debugging of the algorithms.

Therefore, in this section we formulate two possible quality criteria for the numerical solution of such restoring problems:

- the first criterion compares the matrix reconstructed by the simplified algorithm under consideration with the matrix obtained as a result of applying a general algorithm for the formation of each of its elements, like [9] [10]; we shall denote by σ the value of this criterion;
- and the second criterion considers the discrepancy in a special way, it applies the same algorithms that are used as auxiliary ones of the general recovery algorithm considered in this paper; we shall denote by δ the value of this criterion (or by d in some previous papers).

In both cases, the goal is to reduce the values obtained by the applied criterion. The exact formulas are as follows.

1) For σ, we usually set

$$\sigma = \sqrt{\frac{2}{n\cdot(n-1)}\cdot\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(m_{i,j}-\tilde{m}_{i,j}\right)}, \tag{10}$$

where all the elements $\tilde{m}_{i,j}$ are obtained by applying the original algorithm (for instance, already cited Needleman-Wunsch algorithm), *i.e.*, without restoring any elements. Note that for obvious reasons, we cannot often use this method, and also, we cannot apply it for large matrices obtained by some distance determination algorithms; therefore, the following criterion δ can be called more universal.

2) For δ, we usually set

$$\delta = \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{k=j+1}^{n}\delta_{i,j,k}, \tag{11}$$

(we specifically note once again, that the values $\tilde{m}_{i,j}$ are not used here). Each value $\delta_{i,j,k}$ (where $1\le i,j,k\le n, i\ne j, i\ne k, j\ne k$) is the "badness" of corresponding triangle; it is usually counted in the following way. 2a) Firstly, we rename $m_{i,j}, m_{i,k}$ and $m_{j,k}$ into a, b and c, where $a\ge b\ge c$.

2b) If $a\ge b+c$ (*i.e.*, the triangle inequality is violated), we choose in advance a constant ω (usually, =2) and set

$$\delta_{i,j,k} = \max\left(\frac{a}{b+c},\omega\right). \tag{12}$$

2c) Otherwise, for usual triangle, we count its angles; let they be α,β and γ, where $\alpha\ge\beta\ge\gamma$.

2d) Then we set

$$\delta_{i,j,k} = \frac{\alpha-\beta}{\gamma}. \tag{13}$$

Let us especially note that δ, unlike σ, is calculated quickly, despite we need to consider $\sim n^3$ triangles.

Let us also note the relationship of both of these criteria with the task we are considering: for example, for a "random" matrix, we obtained significantly worse results of calculation by the criterion δ, even for small dimensions; to say, for such a matrix of dimension 13 × 13 we obtained δ in limits about 0.4 - 0.5, this is several times higher than the corresponding values for the "correct" matrices of dimension 28 × 28 and significantly lower percentage of initial fullness, see the next section.

5. Some Results of Computational Experiments

As previously mentioned, we place great emphasis on evaluating the results of our computational experiments. We have achieved improved algorithm performance compared to the simplest variant of the branch and boundary method (as described in [9] [10]). By "simplest variant," we refer to the basic greedy heuristic used to select the next separating element. In this paper, we employ a more complex greedy heuristic while abandoning the branch and boundary method. Although the combination of both methods and a more complex greedy heuristic would likely yield even better results according to our quantitative criteria, the time constraints would be unacceptable. We have not conducted detailed computational experiments for this case.

We will briefly describe the computational experiments that we have conducted. In this paper, we have used only one variant of input data (mtDNA from 28 monkeys, which we briefly discussed earlier). However, similar results have been obtained for all input data variants used. The initial matrix, filled in as a result of the Needleman-Wunsch algorithm, is presented in ([9], Tab. 8). The initial matrix with approximately 10% of remaining elements is presented in ([10], Tab. 9). It should be noted that while the calculations in that paper were accurate, the calculations of the values σ and δ were erroneous. However, these mistakes did not affect the relative quality indicators of the algorithms. In this paper, we present completely accurate results, which can be easily verified.

The column designations in **Table 1** are clear, and the row designations have the following meaning:

- (A) corresponds to the matrix, obtained by the best algorithm of ([9], Tab. 13); algorithm does not use branches and boundaries method;
- (B) corresponds to the matrix, obtained by the best algorithm of ([10], Tab.17); the algorithm uses branches and boundaries method;
- (C) corresponds to the matrix, obtained by the simplest greedy algorithm of ([9], Tab. 10); the algorithm does not use branches and boundaries method;;
- (D) corresponds to the matrix, obtained by the complicated greedy algorithm of the current paper; the algorithm does not use branches and boundaries method.

Certainly, all the calculation results shown in this table can be quickly checked using a simple supportive computer program.

Some unusual feature of the results obtained is that the reduction of the values of σ and δ by different algorithms occurs independently of each other, *i.e.*, the binary relationship between the algorithms, which consist in the fact that the first of them gives the best results for both criteria, forms a partial order only. Similar relative results are obtained for other objects of study (not for monkeys etc.). Of course, the best option would be the simultaneous minimization of σ and δ, which we shall achieve in the subsequent work. However, the results presented in this paper are of big interest.

6. Conclusion

Let us formulate some problems for the future solution, *i.e.*, consider a brief description of the nearest directions for further work related to the modification and improvement of the described algorithms.

We introduce a special "reliability coefficient" (let it be $R < 1$, to say, $R = 0.9$ in the following description), which we use as follows. We consider that the initial values of the matrix (in the example considered in the paper, the remaining 10% of the elements after the removal) have a weight of 1.0. The elements derived from only the initial ones (*i.e.*, in the beginning of filling) have a weight of R.

And in the general case (*i.e.*, after filling in some elements) we proceed as follows. As in the greedy algorithm already discussed in this paper, we form all possible triangles obtained together with the element selected for filling, *i.e.*, if the considered unfilled element of the matrix is $m_{i,j}$, then, as before, we consider all such k that $m_{i,k}$ and $m_{j,k}$ are already filled. However, we calculate the obtained values with the reliability coefficients already assigned to these values, *i.e.*, we minimize the general function, which includes values with these coefficients; for the reliability coefficients $R_{i,k}$ and $R_{j,k}$, we assume that the reliability coefficient of the considered triangle is

$$R_{\Delta} = \frac{R_{i,k} + R_{j,k}}{2}. \tag{14}$$

The resulting value obtained as a result of minimization is placed in a matrix with its new reliability coefficient equal to the a priori value of R multiplied by the average reliability coefficient of all considered triangles that form the element $m_{i,j}$: using (3) and assuming we are considering m triangles, this new coefficient can be written as follows:

Table 1. General results of some computational experiments.

T	σ	δ
(A)	0.091	0.110
(B)	**0.029**	0.133
(C)	0.079	0.103
(D)	0.038	**0.044**

The most successful values of σ and δ are highlighted in bold.

$$\frac{\left(R_{\Delta}^{(1)}+R_{\Delta}^{(2)}+\ldots+R_{\Delta}^{(m)}\right)\cdot R}{m}. \tag{15}$$

And, of course, the best value of the reliability coefficient R should be obtained as a result of some self-learning process.

Founding

This work was supported in part by the Higher Education Stability Support Program of Chinese Universities (section "Shenzhen 2022—Science, Technology and Innovation Commission of Shenzhen Municipality").

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Needleman, S. and Wunsch, C.D. (1970) A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, **48**, 443-453. https://doi.org/10.1016/0022-2836(70)90057-4

[2] Winkler, W. (1990) String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Survey Research Methods Sections*, *American Statistical Association*, 354-359.

[3] Van der Loo, M.P.J. (2014) The Stringdist Package for Approximate String Matching. *The R Journal*, **6**, 111-122. https://doi.org/10.32614/RJ-2014-011

[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**, 4347. https://doi.org/10.18287/1613-0073-2017-1902-43-50

[5] Maloy, S. and Hughes, K. (2013) Brenner's Encyclopedia of Genetics. Elsevier, Amsterdam.

[6] Cibelli, J., Wilmut, I., Jaenisch, R., *et al.* (2014) Principles of Cloning. Academic Press, New York.

[7] Sykes, B. (2003) Adam's Curse: The Science That Reveals Our Genetic Destiny. W.W. Norton and Company, New York.

[8] Lennarz, J. and Lane, M. (2013) Encyclopedia of Biological Chemistry. Elsevier, Amsterdam.

[9] 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)

[10] Melnikov, B. and Trenina, M. (2018) Application of the Branches and Boundaries Method in a Problem of the Reconstruction of Distance Matrices between DNA Sequences. *International Journal of Open Information Technologies*, **6**, 1-13. (In Russian)

[11] Melnikov, B., Trenina, M. and Melnikova, E. (2020) Different Approaches to Solving the Problem of Reconstructing the Distance Matrix Between DNA Chains. In: Sukhomlin, V., Zubareva, E., Eds., *Modern Information Technology and IT Educa-*

tion. SITITO 2018. *Communications in Computer and Information Science*, Springer, Cham, 1201. https://doi.org/10.1007/978-3-030-46895-8_17

[12] Melnikov, B. and Terentyeva, Y. (2021) Building Communication Networks: On the Application of the Kruskal's Algorithm in the Problems of Large Dimensions. In *IOP Conference Series*: *Materials Science and Engineering*, **1047**, 012089. https://doi.org/10.1088/1757-899X/1047/1/012089

[13] Melnikov, B., Melnikova, E., Pivneva, S., Churikova, N., Dudnikov, V. and Prus, M. (2018) Multi-Heuristic and Game Approaches in Search Problems of the Graph Theory. In *Information Technologies and Nanotechnologies. Collection of Works of ITNT*-2018. Samara National Research University Named after Academician S. P.Korolev., 2884-2882. (in Russian)

[14] Tikhonov, A.N. (1963) On the Solution of Ill-Posed Problems and the Method of Regularization. *Proceedings of the USSR Academy of Sciences*, **151**, 501-504.

[15] Groetsch, C. (1984) The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind. Pitman Advanced Publishing Program, Boston.

[16] Hanke, M. and Scherzer, O. (2001) Inverse Problems Light: Numerical Differentiation. *The American Mathematical Monthly*, **108**, 512-521. https://doi.org/10.1080/00029890.2001.11919778

[17] Chaikovskii, D. and Zhang, Y. (2022) Convergence Analysis for Forward and Inverse Problems in Singularly Perturbed Time-Dependent Reaction-Advection-Diffusion Equations. *Journal of Computational Physics*, **470**, 111609. https://doi.org/10.1016/j.jcp.2022.111609

[18] Wang, Y.B. and Wei, T. (2005) Numerical Differentiation for Two-Dimensional Scattered Data. *Journal of Mathematical Analysis and Applications*, **312**, 121-137. https://doi.org/10.1016/j.jmaa.2005.03.025