CYBERNETICS AND PHYSICS, VOL. 11, NO. 4, 2022, 217–226

AN INVERSE PROBLEM FOR MATRIX PROCESSING: AN IMPROVED ALGORITHM FOR RESTORING THE DISTANCE MATRIX FOR DNA CHAINS

Boris Melnikov
Department of Computational Mathematics and Cybernetics, Shenzhen MSU – BIT University, China
bormel@smbu.edu.cn, bormel@mail.ru

Ye Zhang
School of Mathematics and Statistics, Beijing Institute of Technology, China
ye.zhang@bit.edu.cn

Dmitrii Chaikovskii*
Department of Computational Mathematics and Cybernetics, Shenzhen MSU – BIT University, China
dmitriich@smbu.edu.cn, mitichya@yandex.ru

Article history:
Received 27.09.2022, Accepted 05.12.2022

Abstract

We consider one of the cybernetic methods in biology related to the study of DNA chains. Namely, 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. The objects of research of these algorithms (for mammals), as a rule, are one of the following 3 variants: the main histocompatibility complex, the mitochondrial DNA, and "the tail" of the Y chromosome.

In the paper we give an improved algorithm for restoring the distance matrix for DNA chains. Compared to our previous publications, the following changes have been made to the algorithm. We abandoned the use of the branches and bounds method, but at the same time significantly improved the greedy auxiliary algorithm used in it. In this paper, we apply only this greedy algorithm to the general solution of the distance matrix reconstruction problem.

As a result of the conducted computational experiments carried out on one of the two considered criteria for the quality of the algorithms, significant improvements were obtained compared to the results given in our previous publications. At the same time, the total running time of the algorithm remained almost the same as in the previous version.

Key words

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 of monkeys, as well as any specific object of study.

"Plants, animals and bacteria all contain the essential biological molecule known as DNA or deoxyribonucleic acid. DNA contains all the information required to build and maintain living organisms. You can think of it as nature's very own top-secret instruction manual . . . "

"This manual is written in multiple combinations, but limited to just 4 letters: A, T, G and C. Each letter denotes a nitrogenous base: A for adenine, T for thymine, G for guanine and C for cytosine. Every living being has a huge supply of these 4 bases, each of which is attached to a pentose sugar and a phosphate molecule. Together, they are known as a nucleotide. These nucleotides are arranged in two long coiled strands like a hair braid."

"Every single cell which builds up a living organism carries information for various functions necessary for the survival of the cell. This genetic information in each cell is stored in molecules called nucleic acids. The most stable form of nucleic acids is called deoxyribonucleic acid (DNA). Each of the DNA strands forms helical structures that are long polymers of millions of linked nucleotides. These nucleotides consist of one of four nitrogen bases, a five-carbon sugar, and a phosphate group. The nitrogen bases - A (Adenine), T (Thymine), G (Guanine), C (Cytosine) encodes the genetic information . . . "
(`https://www.scienceabc.com/pure-sciences` and `https://whatisdna.net/`)

However let us remark, that the total length of the human genome exceeds $3 \cdot 10^9$ characters, which is about

*Corresponding author.

200 000 times longer than mt DNA (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, [Needleman and Wunsch, 1970; Winkler, 1990; van der Loo, 2014] etc. (the authors' usage of Internet searches give about 10 similar algorithms only); see also the description of our algorithm in [Melnikov, Pivneva, and Trifonov, 2017] 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:

- mt DNA, the mitochondrial DNA, inheritance in the "direct female line", see [Maloy and Kelly (Eds), 2013; Cibelli et al. (Eds), 2014] etc.; for human, the the length of its sequence exceeds 16 000 characters;
- "the tail" of the Y chromosome, inheritance in the "direct male line", see [Sykes, 2003, p. 290] etc.; for human, the the length of its sequence exceeds 50 000 characters;
- MHC, the main (major) histocompatibility complex, see [Lennarz and Lane (Eds), 2013] etc.; usually, we cannot say about its length.

"The structure of MHC allows to bind peptides of varying lengths because both ends of the peptide are free . . . ", see ibid.

"The MHC complex encodes the α-chains of the MHC class I molecules human leukocyte antigen (HLA)-A, HLA-B, and HLA-C and the α- and β-chains of the MHC class II molecules HLA-DR, HLA-DP, and HLA-DQ, all of which are expressed in a co-dominant fashion."

"MHC class I is expressed by all nucleated cells and platelets in jawed vertebrates, although the amount on the cell surface varies among cell types and under different inflammatory conditions."

"The folded MHC class II molecule consists of two transmembrane proteins, an α-chain and a β-chain, which together form a protein . . . Peptides bound by MHC class II molecules typically are longer than 10 amino acids and occasionally more than 20 amino acids."

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, the distances between DNA sequences are often used in scientific and popular science literature. However, as we already said, there are several different algorithms for calculating them, and for each pair of sequences, the operation of any of these algorithms takes a lot of 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 mtDNAs using the Needleman – Wunsch algorithm [Needleman and Wunsch, 1970]; 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 of restoring such a partially filled matrix arises. We started publishing our variants of similar algorithms for restoring partially filled matrices in [Melnikov, Pivneva, and Trifonov, 2017] (the simplest algorithm was described very briefly there), after which we returned to this problem in [Melnikov and Trenina, 2018a; Melnikov and Trenina, 2018b], where a variant of the algorithm using the method of branches and boundaries was described in detail.

Remark. At the same time, it is worth noting that in the last two papers we made computational mistakes, which, however, did not affect the overall assessment of the results of calculations given in these papers at all. For example, it was said that we leave about 30—35 % of the elements in the matrices of dimension about 30×30, obtained for processing, while we left a significantly smaller number of elements, i.e., about 10 %; thus, we solved a more difficult task. Certainly, the latter fact indicates much greater possibilities in the application of the algorithms we are considering. In addition, some computational errors were made when obtaining obtaining the values of σ and δ, which, again, did not affect the evaluation of the calculation results.

Despite the previously successful results of calculations, we return in this paper to the algorithm variants that *do not use* the branches and boundaries method: as our recent work has shown (primarily in the subject areas related to graph theory and the development of ultra-large communication networks, [Melnikov and Terentyeva, 2021] etc.), even greater improvement in the quality of the algorithm can often be achieved without improving the auxiliary heuristics of the branches and boundaries method. We are improving the algorithms that formulate the greedy function of this method only; however, these algorithms can also be called auxiliary to the method of branches and boundaries.

In this paper, we describe a similar improvement of the greedy algorithm, now for the task of reconstructing the matrix of distances between DNA chains. As the obtained results of computational experiments show (see Section 5), they are better than ones obtained using simple variants of the branches and boundaries method. Let us repeat that for restoring partially filled matrices, i.e., for the inverse problem of matrix processing, we used the method of branches and boundaries before, but in this paper, we do not use it.

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 mt DNA s 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.

The paper has the following structure. In Section 2, we consider a brief description of the greedy algorithms of restoring the distance matrix. In Section 3, we give the theoretical substantiation of the possibility of improvement of the greedy algorithm (without the variants of the method of branches and boundaries).

In Section 4, 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 formation algorithm for each of its elements; 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. In both cases, the goal is to reduce the values obtained by the applied criterion.

In Section 5, we give some results of computational experiments; we evaluate these results well. In Section 6, we formulate some problems for the future solution. And in Conclusion (Section 7), we briefly repeat the content of this paper.

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 non-removable 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. When calculating using the elementary formulas of probability theory, we find that the probability of this event is about 25 %. However, we simply do not consider such examples as source data.

We considered the simplest heuristic for filling in the distance matrix without using the method of branches and boundaries in [Melnikov and Trenina, 2018a]. (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 [Melnikov and Trenina, 2018a].

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 $(m_{i,j})$, we choose the pair (i, j) such that:

- $m_{i,j} < 0$ (in our natural notation, this means that there is no corresponding matrix element in the input data);
- and the following formula is achieved:

$$\max_{\substack{1 \leqslant i,j \leqslant n \\ i \neq j}} \sum_{\substack{1 \leqslant k \leqslant n \\ k \neq i,\, k \neq j}} \big(\operatorname{sgn}(m_{k,i}) + \operatorname{sgn}(m_{k,j})\big).$$

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

$$\frac{\alpha - \beta}{\gamma},$$

where α, β and γ are the angles of the derived triangle, and $\alpha \geqslant \beta \geqslant \gamma$.

(Note that earlier we sometimes used another formula, $\frac{a-b}{c}$, where a, b and c are the sides of the derived triangle, and $a \geqslant b \geqslant c$. At the same time, in both cases, 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 [Melnikov et al., 2018c], "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. α in previous formulas; note that in [Melnikov and Trenina, 2018a], this value was counted final) as the beginning of the iterative process, we get a pseudo-optimal value in a few iterations. (We usually limited the number of iterations to 10, this gives an acceptable value of the calculation time.)

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 problems, which means that a small error in the source data can lead to a large error in the calculated derivatives [Tikhonov, 1963; Groetsch, 1984; Hanke and Scherzer, 2001; Chaikovskii and Zhang, 2022], 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 $\Omega \in \mathbb{R}^2$, then the matrix elements will represent the noisy values of the function of two variables $u^{\delta}_{i,j}$. It is natural to assume that the domain Ω is divided into $N \leqslant n^2$ parts $\{\Omega_i\}_{i=1}^{N}$, and there is the only one value $u^{\delta}_{i,j}$ in each parts. Denote d_i as the diameter of Ω_i and let $d = \max\{d_i\}$.

In this case, we obtain the deterministic model

$$\max|u(x_i, y_j) - u^{\delta}_{i,j}| \leqslant \delta$$

between the noisy data $\{u^{\delta}_{i,j}\}$ and the corresponding exact values $\{u(x_i, y_i)\}$ at grid points

$$X_n := \{x_1 < x_2 < \cdots < x_n\}$$
$$\text{and } Y_n := \{y_1 < y_2 < \cdots < y_n\}.$$

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

$$u^{\varepsilon} = \arg \min_{v \in C^1(\Omega)} \left(\frac{1}{n^2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \left(v(x_i, y_j) - u^{\delta}_{i,j}\right)^2 + \right.$$

$$\left. \varepsilon \left(\left\| \frac{\partial^2 v(x,y)}{\partial x^2} \right\|^2_{L^2(\Omega)} + \left\| \frac{\partial^2 v(x,y)}{\partial y^2} \right\|^2_{L^2(\Omega)} \right) \right), \quad (1)$$

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

$$\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(u^{\varepsilon}(x_i, y_j) - u^{\delta}_{i,j}\right)^2 = \delta^4.$$

Then by [Wang and Wei, 2005, Theorem 3.3], the following proposition holds.

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

$$\|u^{\varepsilon}(\cdot, \cdot) - u(\cdot, \cdot)\|_{H^1(\Omega)} \leqslant C_1 d^{1/4} + C_2 \sqrt{\delta},$$

where C_1 *and* C_2 *are some constants depending on the area* Ω *and on* $\|\Delta u(x, y)\|_{L^2(\Omega)}$. $\square$

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

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

Due to the fact that with an increase in the value of the norm $\|\Delta u(x, y)\|_{L^2(\Omega)}$ 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 more noisy data to obtain the necessary accuracy. Another smoothing technique can be used is the convolution, see [Gulliksson et al., 2016, Theorem 3] and [Lin, Cheng, and Zhang, 2018, Section 4] for details.

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 $\{u^{\delta}_{i,j}\}$ 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 [Melnikov, Pivneva, and Trifonov, 2017] for distance matrices, which consists in the fact that the three elements of the matrix

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

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 $M = (m_{i,j})$ and its elements $m_{i,j}$, we always [1] 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:

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

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

[1] I.e., when considering summation, as well as when taking minima and maxima.

- $r^{(1)}_{i,j,k} = \max\left(m_{i,j}, m_{k,j}, m_{k,i}\right)$,
- $r^{(2)}_{i,j,k} = \min\left(m_{i,j}, m_{k,j}, m_{k,i}\right)$,
- and $\sigma_{i,j}$ is as follows:

$$\max_{\substack{1\leqslant k\leqslant n\\ k\neq i, k\neq j}} \frac{2r^{(1)}_{i,j,k} + r^{(2)}_{i,j,k} - m_{i,j} - m_{k,j} - m_{k,i}}{r^{(2)}_{i,j,k}}.$$

Then the original problem can be reformulated into the problem of minimizing the error value σ (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 $\{u^{\delta}_{i,j}\}$ 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 [Melnikov, Pivneva, and Trifonov, 2017].

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 more 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:

(1) 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 [Melnikov and Trenina, 2018a; Melnikov and Trenina, 2018b]; we shall denote by σ the value of this criterion;

(2) 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

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

where all the elements $\widetilde{m}_{i,j}$ 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

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

(we specifically note once again, that the values $\widetilde{m}_{i,j}$ are not used here). Each value $\delta_{i,j,k}$ (where $1 \leqslant i, j, k \leqslant n$, $i \neq j$, $i \neq k$, $j \neq k$) 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 $a \geqslant b \geqslant c$.

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

$$\delta_{i,j,k} = \min\left(\frac{a}{b+c}, \omega\right). \tag{2}$$

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

(2d) Then we set

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

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

[2] In some our previous papers, we wrote that the number of such violations ranged from 2 (the minimum value in previous calculations) to several dozen (for matrices about 30×30); it depended on the subject area, as well as on the specific algorithm. In the case under consideration here (28 species of monkeys of different genera, mt DNA, Needleman – Wunsch algorithm), we obtained no such violations at all, then the item (2b) was not used.

Table 1. The result of the recovery program using a greedy algorithm, which is some more complicated than the algorithm of [Melnikov and Trenina, 2018a]. In both cases, the branches and boundaries method is not used. Like [Melnikov and Trenina, 2018a], the value of the table 315 corresponds to the obtained value 0.315, etc.

0	313	258	334	328	341	334	326	505	324	344	421	315	334	335	335	262	334	332	337	334	997	506	329	338	325	325	344
313	0	330	313	303	342	323	222	505	320	340	420	312	313	334	334	297	328	331	332	292	997	504	269	336	316	320	342
258	330	0	335	343	339	333	320	505	327	347	422	319	335	334	334	296	332	331	343	341	997	505	335	337	273	309	344
334	313	335	0	329	340	292	330	505	327	344	420	328	311	334	334	332	322	333	333	257	997	503	333	335	333	330	344
328	303	343	329	0	343	332	236	505	330	341	417	328	329	336	336	325	333	334	304	327	997	505	329	338	320	328	343
341	342	339	340	343	0	341	341	505	341	351	421	341	340	341	341	341	341	341	344	337	997	506	343	341	341	341	345
334	323	333	292	332	341	0	332	505	318	344	421	326	329	335	335	334	279	334	337	331	997	499	334	331	334	332	344
326	222	320	330	236	341	332	0	505	329	344	420	326	330	336	336	322	333	333	333	330	996	504	325	337	298	324	344
505	505	505	505	505	505	505	505	0	505	505	505	506	505	505	505	505	505	505	505	505	996	523	505	505	505	505	505
324	320	327	327	330	341	318	329	505	0	344	421	302	348	354	354	328	350	351	337	333	996	508	333	354	332	347	360
344	340	347	344	341	351	344	344	505	344	0	419	344	359	361	361	344	361	360	326	341	996	509	332	360	344	359	268
421	420	422	420	417	421	421	420	505	421	419	0	420	427	428	428	420	428	427	315	419	996	514	420	427	420	427	428
315	312	319	328	328	341	326	326	506	302	344	420	0	355	361	361	297	358	357	337	332	996	509	332	360	330	353	366
334	313	335	311	329	340	329	330	505	348	359	427	355	0	361	361	355	358	359	356	264	999	678	355	361	356	357	367
335	334	334	334	336	341	335	336	505	354	361	428	361	361	0	301	358	357	264	362	361	997	534	401	300	401	364	367
335	334	334	334	336	341	335	336	505	354	361	428	361	361	301	0	358	355	321	362	361	997	534	401	256	401	364	367
262	297	296	332	325	341	334	322	505	328	344	420	297	355	358	358	0	359	326	337	331	996	509	330	342	327	354	366
334	328	332	322	333	341	279	333	505	350	361	428	358	358	357	355	359	0	356	361	359	996	287	360	265	359	359	367
332	331	331	333	334	341	334	333	505	351	360	427	357	359	264	321	326	356	0	360	358	997	534	399	310	398	361	367
337	332	343	333	304	344	337	333	505	337	326	315	337	356	362	362	337	361	360	0	303	996	509	336	361	336	359	365
334	292	341	257	327	337	331	330	505	333	341	419	332	264	361	361	331	359	358	303	0	997	509	330	360	333	357	366
997	997	997	997	997	997	997	996	996	996	996	996	996	999	997	997	996	996	997	996	997	0	995	997	997	997	997	997
506	504	505	503	505	506	499	504	523	508	509	514	509	678	534	534	509	287	534	509	509	995	0	509	531	509	533	535
329	269	335	333	329	343	334	325	505	333	332	420	332	355	401	401	330	360	399	336	330	997	509	0	401	332	395	404
338	336	337	335	338	341	331	337	505	354	360	427	360	361	300	256	342	265	310	361	360	997	531	401	0	401	397	401
325	316	273	333	320	341	334	298	505	332	344	420	330	356	401	401	327	359	398	336	333	997	509	332	401	0	244	405
325	320	309	330	328	341	332	324	505	347	359	427	353	357	364	364	354	359	361	359	357	997	533	395	397	244	0	402
344	342	344	344	343	345	344	344	505	360	268	428	366	367	367	367	366	367	367	365	366	997	535	404	401	405	402	0

$$\sigma = 0.079, \quad \delta = 0.103$$

Table 2. The result of the recovery program using a complicated greedy algorithm. The branches and boundaries method is not used.

0	313	258	332	328	341	332	326	505	317	344	317	286	332	330	330	262	332	329	333	332	997	332	329	332	270	270	344
313	0	330	284	245	342	284	222	505	315	340	316	311	284	317	285	297	280	321	302	292	997	273	269	286	316	315	342
258	330	0	334	343	339	332	320	505	320	347	340	283	334	330	330	269	332	329	343	341	997	332	335	331	273	264	344
332	284	334	0	285	340	292	283	505	325	344	329	325	253	322	285	328	282	325	293	257	997	276	283	285	327	327	344
328	245	343	285	0	343	284	236	505	326	341	323	325	280	323	282	322	277	326	304	295	997	273	267	281	320	326	343
341	342	339	340	343	0	341	341	505	341	351	332	341	340	341	341	341	341	341	344	337	997	341	343	341	341	341	345
332	284	332	292	284	341	0	282	505	318	344	318	322	280	324	283	327	279	327	292	287	997	276	281	281	327	329	344
326	222	320	283	236	341	282	0	505	324	344	314	324	276	324	278	321	276	326	297	290	996	271	262	277	298	323	344
505	505	505	505	505	505	505	505	0	505	505	505	506	505	505	505	505	505	505	505	505	996	505	505	505	505	505	505
317	315	320	325	326	341	318	324	505	0	344	313	302	314	309	307	324	315	311	330	329	996	310	329	311	325	312	327
344	340	347	344	341	351	344	344	505	344	0	325	344	326	360	360	344	328	359	326	341	996	355	332	356	344	359	268
317	316	340	329	323	332	318	314	505	313	325	0	311	308	310	307	310	311	311	315	309	996	308	306	310	306	311	323
286	311	283	325	325	341	322	324	506	302	344	311	0	309	306	305	297	310	308	330	330	996	306	329	308	291	276	354
332	284	334	253	280	340	280	276	505	314	326	308	309	0	304	266	337	268	308	300	264	999	282	292	268	336	309	353
330	317	330	322	323	341	324	324	505	309	360	310	306	304	0	301	308	303	264	304	301	997	300	299	300	331	307	355
330	285	330	285	282	341	283	278	505	307	360	307	305	266	301	0	335	264	321	297	291	997	289	288	256	330	335	355
262	297	269	328	322	341	327	321	505	324	344	310	297	337	308	335	0	338	326	330	329	996	338	328	332	264	272	354
332	280	332	282	277	341	279	276	505	315	328	311	310	268	303	264	338	0	306	300	294	996	287	291	265	336	336	353
329	321	329	325	326	341	327	326	505	311	359	311	308	308	264	321	326	306	0	307	305	997	304	303	310	334	305	355
333	302	343	293	304	344	292	297	505	330	326	315	330	300	304	297	330	300	307	0	303	996	300	293	300	327	336	353
332	292	341	257	295	337	287	290	505	329	341	309	330	264	301	291	329	294	305	303	0	997	294	288	293	327	336	354
997	997	997	997	997	997	997	996	996	996	996	996	996	999	997	997	996	996	997	996	997	0	995	997	997	997	997	997
332	273	332	276	273	341	276	271	505	310	355	308	306	282	300	289	338	287	304	300	294	995	0	265	294	336	330	349
329	269	335	283	267	343	281	262	505	329	332	306	329	292	299	288	328	291	303	293	288	997	265	0	294	327	330	347
332	286	331	285	281	341	281	277	505	311	356	310	308	268	300	256	332	265	310	300	293	997	294	294	0	334	328	346
270	316	273	327	320	341	327	298	505	325	344	306	291	336	331	330	264	336	334	327	327	997	336	327	334	0	244	344
270	315	264	327	326	341	329	323	505	312	359	311	276	309	307	335	272	336	305	336	336	997	330	330	328	244	0	344
344	342	344	344	343	345	344	344	505	327	268	323	354	353	355	355	354	353	355	353	354	997	349	347	346	344	344	0

$$\sigma = 0.038, \quad \delta = 0.044$$

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

First of all, let us give a few comments on the given large tables of results. Both are given for the reader's possible verification of these results; at the same time, anyone can either simply recognize the table as a picture, or request from the authors, after which we shall send the same tables in the form of text. Having these tables, anyone can simply check their characteristics (badness, etc., according to the formulas given in the paper). It is possible to say, simplifying a little, that *the topic of the article is how to obtain the missing values in these tables with minimum badness*.

As we have already noted, we evaluate the results of computational experiments well. Namely, an improvement in the performance of the algorithm was obtained (according to both criteria given in the previous section), compared with the simplest variant of the branch and boundary method, see [Melnikov and Trenina, 2018a; Melnikov and Trenina, 2018b] etc. More precisely, by the words "simplest variant", we mean that the simplest greedy heuristic used to select the next separating element. Here we apply a more complex greedy heuristic, while abandoning the method of branches and boundaries. It is clear that even more successful results (from the point of view of the quantitative criteria formulated above) we would have obtained by using both the branch and boundary method and a more complex greedy heuristic at the same time; however, it seems that we shall not satisfy acceptable time constraints. Though, we did not conduct detailed computational experiments for this case.

To perform all the computational experiments described in the article, we used a computer with the following characteristics:

```
Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
```

In all our computational experiments, the total time of the computer was extremely short, and we did not record it, since it is significantly less than the time required to output the results of the algorithm (especially in comparison with the time necessary for the initial filling of only one cell of the matrix). For comparison, we repeat once again:

- such calculations by the method of branches and boundaries for dimensions of the order of 30×30 take about 1 second;
- the calculation of the distance by the Needleman – Wunsch algorithm between two sequences describing mtDNA takes about 3 minutes;
- and the filling of the entire matrix by the Needleman – Wunsch algorithm of the order of 30 takes about 1 day.

Thus, let us briefly describe the computational experiments already carried out. In this paper, only one variant of the input data is used (28 monkeys, mt DNA, we briefly talked about this variant above); but we note in advance that *similar results were obtained on all variants* of the input data used. The initial matrix filled in as a result of the Needleman – Wunsch algorithm is given in [Melnikov and Trenina, 2018a, Tab. 8]. The initial matrix with about 10 % of remaining elements is given in [Melnikov and Trenina, 2018a, Tab. 9]. Let us remind once again that the calculations of the matrices in that paper were correct, but the calculations of the values σ and δ were erroneous. However, the mistakes did not affect the relative quality indicators of the algorithms. In the current paper, we present completely correct results, they are easy to check.

The column designations in Table 3 are clear, and the row designations have the following meaning:

(A) corresponds to the matrix, obtained by the best algorithm of [Melnikov and Trenina, 2018a] (that matrix was given on [Melnikov and Trenina, 2018a, Tab. 13]); let us remind once again that the algorithm does not use branches and boundaries method;
(B) corresponds to the matrix, obtained by the best algorithm of [Melnikov and Trenina, 2018b] (that matrix was given on [Melnikov and Trenina, 2018b, Tab. 17]); the algorithm uses branches and boundaries method;
(C) corresponds to the matrix, obtained by the simplest greedy algorithm of the current paper; the algorithm does not use branches and boundaries method; see its results (i.e., the obtained matrix and its characteristics) in Table 1 of this paper;
(D) corresponds to the matrix, obtained by the complicated greedy algorithm of the current paper; the algorithm does not use branches and boundaries method; see its results in Table 2 of this paper;

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

Table 3. 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.

A brief discussion of the obtained results is given in Conclusion, Section 7.

6 Some possible directions for further work on this subject

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.

For the *first possible direction*, we shall temporarily assume that the original processed algorithm under consideration (the Needleman – Wunsch algorithm is one of the possible examples only) is obviously not optimal, and *requires improvement*. Note that this fact, of course, is always true, regardless of everything else: for example, almost all the original algorithms (i.e., determining the distance between two given DNA sequences) are based on the long-known algorithm for constructing the Levenshtein metric (or Levenshtein distance), [Levenshtein, 1966], to which the "penalties" are additionally added. At the same time, the numerical values of such penalties are always selected based on preliminary expert assessments (or even simply assumed to be equal to 1), [Christen, 2012a; Christen, 2012b; Yu et al., 2016; Suganthan et al., 2018]. However, it is clear that any self-study procedure should give an improvement of such values (penalties, etc.). Therefore, another inverse problem arises: to achieve such an improvement, where the minimization of the value δ is used as the criterion.

In contrast to the first possible direction briefly described before, in other variants for further work we consider the given algorithm (the Needleman – Wunsch algorithm etc.) as something "God-given", i.e., not as the subject to change; we try, as above in this paper, to tune in to it.

The second direction. 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

$$R_\Delta = \frac{R_{i,k} + R_{j,k}}{2}. \tag{3}$$

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:

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

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

The third direction. Here we propose to continue the simultaneous application of increasingly complex greedy heuristics and the method of branches and boundaries. We hope that the results will surpass those obtained in this paper and in [Melnikov and Trenina, 2018a; Melnikov and Trenina, 2018b]. At the same time, we note that almost no time is spent on the method of branches and boundaries, at least in comparison with the time that needs to be spent on filling in only one initial element of the matrix.

The fourth direction. This is a detailed study of the quality of algorithms depending on the dimension of the matrix and the percentage of its initial fullness. (Note that we have not actually started solving this problem yet.)

The fifth direction. We believe that it is possible to choose which elements of the matrix should be initially filled, of course, within a predetermined total number of them. This possibility sometimes reflects the subject area under consideration: after all, the total time of such filling will practically not depend on specific elements, but will depend on their number only. Is it true that in this case, the elements for the initial filling should be selected so that there would be approximately the same number of them in all rows of the matrix? (Answering this question is the fifth possible direction of work.)

The sixth direction represents a new approach to comparing different heuristics for distances between DNA chains, i.e., an alternative approach to the one discussed in [Melnikov, Pivneva, and Trifonov, 2017] and some of our other papers cited there. Namely, after filling in several distance matrices with various algorithms (i.e., algorithms for obtaining distances between pairs of DNA chains), we consider all possible triangles in these matrices; note again that their number is quite large, of the order $\sim n^3$. Next, for each initial filling algorithm, we consider a list of these triangles, ordered, for example, by non-increasing values of the badness (considered, for example, as $\delta_{i,j,k}$, see (2)).

The main idea of this heuristics is that we assume that all the algorithms described in the literature and on the Internet for obtaining distances between pairs of DNA chains are logically correct. Therefore, considering some "natural" metric on such ordered sequences of triangles, for the "best" algorithm for the initial filling of matrices (the best algorithm for calculating the distance between pairs of DNA chains), we obtain the minimum value of the sum of the distances to other ordered sequences of triangles.

As such a natural metric on ordered sequences of triangles, we can choose some natural function from the pairwise correlation between the sequences. In our preliminary calculations, we choose a linear function as such one:

- having a value 0 in the case of matching sequences;
- having a value 1 (the maximum possible value) in the case of the maximum possible number of the minimum number of exchanges required to convert from the first sequence to the second one; note that for a matrix of the order $n \times n$, the number of its triangles is $\sim n^3$, therefore the number of possible exchanges is $\sim n^6$;
- and intermediate values otherwise; these values are calculated, as we already noted, using the simplest linear function.

Note that our version of the pairwise correlation is obtained here with another version of the linear function, i.e. when simultaneously replacing 0 with 1 and 1 with -1 in the items above.

After that, we propose to consider "pairwise correlation between pairwise correlations": for this, we need to arrange the heuristic algorithms for the initial placement of DNA chains in two ways (i.e. according to [Melnikov, Pivneva, and Trifonov, 2017] and according to the above).

The seventh direction. And of course, as a possible direction for further work, it is necessary to consider new *objects of application* of the described algorithms:

- other species (besides monkeys),
- Y-chromosomes instead of mt DNA s,
- other initial filling algorithms (instead of Needleman – Wunsch) . . .

Besides, for monkeys, we propose to consider a very strong increase in dimension (it is optimal to consider all types, $500+$) with a simultaneous decrease in the percentage of initially filled matrix cells (to say, to $5\,\%$ instead of $10\,\%$).

Of course, these seven directions do not limit further possible work on the topics described here . . .

7 Conclusion

In this article, we propose an improved algorithm for recovering missing data in distance matrices. *An unusual feature* of the results is that the reduction of the quality criteria σ and δ used in several of our articles by different algorithms occurs independently of each other, i.e., there is a 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.

8 Acknowledgement

This work was supported by the Natural Science Foundation of Guangdong Province (No. 2019A1515110971) and Shenzhen National Science Foundation (No. 20200827173701001).

References

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.

Christen, P. (2012a). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, Berlin.

Christen, P. (2012b). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE TKDE*, **24** (9), pp. 1537–1555.

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

Groetsch, C. (1984). *The theory of Tikhonov regularization for Fredholm equations of the first kind. Notes Math., vol. 105*. Pitman Advanced Publishing Program, Boston.

Gulliksson, M., Holmbom, A., Persson, J., Zhang, Y. (2016). A separating oscillation method of recovering the G-limit in standard and non-standard homogenization problems. *Inverse Problems*, **32**(2), 025005.

Hanke, M. and Scherzer, O. (2001). Inverse problems light: numerical differentiation. *The American Mathematical Monthly*, **108**, pp. 512–521.

Lennarz, J. and Lane, M. (Eds) (2013). *Encyclopedia of Biological Chemistry*. Elsevier Publ., Amsterdam.

Levenshtein, V. (1966). Binary codes capable of correcting. Deletions, insertions, and reversals. *Soviet Physics. Doklady*, **10**, pp. 707.

Lin, G., Cheng, X., Zhang, Y. (2018). A parametric level set based collage method for an inverse problem in elliptic partial differential equations. *Journal of Computational and Applied Mathematics*, **340**, 101-121.

Maloy, S. and Hughes, K. (Eds) (2013). *Brenner's Encyclopedia of Genetics*. Elsevier Publ., Amsterdam.

Melnikov, B., Pivneva, S., and Trifonov, M. (2017). Various algorithms, calculating distances of DNA sequences, and some computational recommendations

for use such algorithms. *In: CEUR Workshop Proceedings, 1902*, pp. 43–50.

Melnikov, B. and Trenina, M. (2018a). On a problem of the reconstruction of distance matrices between DNA sequences. *International Journal of Open Information Technologies*, **6** (6), pp. 1–13 (in Russian).

Melnikov, B. and Trenina, M. (2018b). 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** (8), pp. 1–13 (in Russian).

Melnikov, B., Melnikova, E., Pivneva, S., Churikova, N., Dudnikov, V., and Prus, M. (2018c). 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.*, pp. 2884–2882 (in Russian).

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(1), 012089*.

Needleman, S. and Wunsch, Ch. (1970). A general method is applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48** (3), pp. 443–453.

Suganthan, P. G. C., Ardalan, A., Doan, A., and Akella, A. (2018). Smurf: Self-Service String Matching Using Random Forests. *Proceedings of the VLDB*, **12** (3), pp. 278–291.

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

Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. *The Dokl. Akad. Nauk SSSR*, **151** (3), pp. 501–504.

van der Loo, M. (2014). The stringdist package for approximate string matching. *The R Journal*, **6**, pp. 111–122.

Wang, Y. B. and Wei, T. (2005). Numerical differentiation for two-dimensional scattered data. *Journal of Mathematical Analysis and Applications*, **312** (1), pp. 121–137.

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*, pp. 354–359.

Yu, M., Li, G., Deng, D., and Feng, J. (2016). String similarity search and join: A survey. *Frontiers of Computer Science*, **10** (3), pp. 399–417.