

Various algorithms, calculating distances of DNA sequences, and some computational recommendations for use such algorithms

B.F. Melnikov¹, S.V. Pivneva², M.A. Trifonov²

¹Center of Information Technologies and Systems for Executive Power Authorities, 19, str. 1, Presnenskiy Val, 123557, Moscow, Russia
²Togliatti State University, 14, Belorusskia st., 445020, Togliatti, Russia

Abstract

In recent years, various approaches have been described to determine the similarity of the DNA sequences; each of which defines a metric for the set of DNA sequences. In this paper, we propose a new approach to solving this problem; moreover, algorithms for its implementation are based on multiheuristic approach to the discrete optimization problems previously developed by us. However, the main focus of this article is to describe our original approach to compare the quality of defined metrics on the set of DNA sequences. The last approach is based on the fact, that the triples of distances between genomes should ideally form isosceles acute triangles. On the basis of this assumption, we proposed value of the norm, gives in practice acceptable results; the validity of this approach is also discussed in the article. In the course of work on the implementation of algorithms have been carried out computational experiments with 100 DNA of “distant” species, as well as with representatives of several genomes of great apes and humans. Several possible standards defined comparative quality algorithms describing metric distances on DNA sequences.

Thus, the main focus of this article is to describe our original approach to compare the quality of defined metrics on the set of DNA sequences. The approach is based on the fact, that the triples of distances between genomes should ideally form isosceles acute triangles. On the basis of this assumption, we proposed value of the norm, gives in practice acceptable results. In the course of work on the implementation of algorithms have been carried out computational experiments with 100 DNA of “distant” species, as well as with representatives of several genomes of great apes and humans.

Keywords: metric evaluation; algorithms; multiheuristic approach; original approach to compare the quality of defined metrics on the set of DNA

1. Introduction

The problem of determining the similarity of DNA is a special case of non-exact matching sequences[1]. The “non-exacting” (“mistaking”) is that when comparing lines it is possible to identify similar sequences, despite the errors and distortions in them, for example, changing, deleting, or inserting some characters. The amount of such distortion sets metric on the set of rows, which is determined by the minimum number of edit operations that provide a single line of another. This problem occurs in many areas. For example, a comparison of the genes and chromosomes of proteins is a major challenge and one of the main tools of molecular biology and bioinformatics [1,2,3,4,5,6,7]. The exact comparisons of nucleotide chains (and also computing distances using such comparisons) are unacceptable because of errors in the data, and due to possible mutations. Inaccurate mapping is carried out like the text processing. One of the metrics obtained by comparing the words (Levenshtein distance) is used to correct errors, to enhance recognition of scanned documents, to search in the information systems and databases [1]. To find an approximate solution, there are different algorithms in different subject areas, for example, to search a database of genetic information is widely used algorithm BLAST ([2]etc.), approximating Needleman-Wunsch algorithm.

Thus, in Section 2 of this paper we describe the application to the defining similarity of DNA sequences so called multiheuristic approach [8,9], which is in fact the big extension of the branch and bound method. Note that previously, before our works, branch and bound method, apparently, was not used to solve this problem.

Thus, the calculation of the distance (metric) between the rows of DNA of different species of organisms is one of the most important tasks of modern bioinformatics. As already noted, today there are many algorithms allowing to make an approximate calculation of the polynomial time ([4,5,6,7,10] and many others). The obvious disadvantage when calculating the distance between the one and the few lines of DNA is to provide different results when using different algorithms to calculate metrics. However, the authors do not know the work, which compared to a variety of algorithms for solving this problem. In this regard, one of the tasks that are discussed in this article was to develop a method for the comparative evaluation of such algorithms; moreover, this problem seems to be the most important our consideration. As a result, we have proposed a method of evaluation using the properties of an isosceles triangle in a metric space (Section 3, so called “triangular norm”, we calculate using it so called badness, related to some metric for several species).

We are also looking at options to improve some existing metrics. In this case, none of the methods we are considering the construction of the distance between the strands of DNA is not a disadvantage to use it to evaluate the distances between the neighbors as species (pairs “human – chimpanzee” and “human – bonobo” etc.), and between more distant species (pairs “human – crocodile” and “chimpanzee – crocodile” etc.). This is because we consider first of all corners of the triangles in the Euclidean space. However, we have made some computational experiments connected with the application for conversion metrics of continuous monotone functions (Section 4).

Brief results of computational experiments over 100 genomes are considered in Section 5. Among these results, it is worth noting the following. Firstly, for the “distant” species badness is very small, which indicates that the right choice of our approaches and relevant specific algorithms; in this case the fact is true for a number of different metrics. Secondly, as for the “distant” species we proposed approach to the definition of the metric gives the best results (all considered “triangular”

standards), among 5 considered metrics [4,5,6,7,10]. For the “near” species (human and apes), the results are somewhat worse (value of badness is increased, and, besides, our version of the metric gives 2nd for the quality of the result). Third, it is unlikely any of these metrics are appropriate for determining the distance between the subspecies: so that the application of these algorithms to the human race sometimes arises violation of the triangle inequality. The accurate explanations of recent facts, apparently, should lead biologists, but we also try to explain them, from our point of view.

Some possible areas for further work already ongoing by our group at the moment are summarized in Conclusion (Section 6).

2. Algorithm for determining the distance between nucleotide sequences based on the multiheuristic approach

As we said before, the multiheuristic approach to the problems of discrete optimization was considered [8,9] and in many other following papers. In this section, we describe its version for determining the distance between nucleotide sequences. For this problem, it was used as follows¹. Let x, y be corresponding strings, i, j be indexes of considered symbols of strings x and y correspondently, r be the value of metric to be found. By shifting the line, we mean increasing by 1 of the corresponding index. The general scheme of the algorithm can be described as follows.

```

Input:          strings x and y.
Step 1:         i := 0, j := 0, r := 0;
Step 2:         if x[i] = y[j] then begin
                  shift both lines;
                  r := the cost of matching of symbols x[i] and y[j];
                end
                else begin
                  apply heuristics for generating
                  possible "trajectories" of the shift
                  in the position of i' and j',
                  such that x[i'] = y[j'];
                  evaluate them with other heuristics;
                  average these estimates using risk functions;
                  make shifting
                  (value can be changed);
                end;
Step 3:         repeating the second step until
                it reaches the end of one of the lines.

```

The cost of matching two symbols in a simplest case equals to 1; for DNA, it can be defined using some table of amino acid replacement costs, e.g. BLOSUM, see [1, 2, 11].

For this algorithms, the following heuristics were used.

1. We select such trajectories that the value $(i' - i) + (j' - j)$ is minimum, or close to minimum. E.g. we first lookup all the trajectories with one string shifted by one symbol; next with one string shifted by two symbols or both strings shifted by one symbol, etc.
2. We shift a string, which current symbol found less frequent in the other string. For this heuristics it's preferable to know probabilities of appearance of a given symbol in each of the strings. If those probabilities are not known a priori, we consider them being equal. While following the algorithm we can adjust those probabilities or use aging algorithm [12], such that probability of a given symbol will be defined by some fragment of a string instead of a whole string. If probabilities for both strings are equal, we shift a string in which more symbols are left.
3. Combination of previous heuristics (1 and 2); to calculate the position using second heuristics we sum probabilities of finding other string for all symbols that will be passed by a shift.
4. Use of an algorithm of a longest common subsequence search for $x[i..i+k]$ and $y[j..j+k]$, where $k \sim 15$. For shift we use i', j' , at which the longest common subsequence ends. If no common subsequence found, the search range is increased. When using this heuristics the result is close to the longest common subsequence value.
5. Combination of 3 and 4; the position (i', j') given by forth heuristics is a ratio of length of the longest common subsequence of strings $x[i..i']$ and $y[j..j']$ to an average shift length from (i, j) to (i', j') .
6. We use algorithm [13, 14] for strings $x[i..i+k]$ and $y[j..j+k]$, where $k \sim 15$, then shift to (i', j') , having the greatest value in Needleman-Wunsch table.

Combination of 3 and 6; the position (i', j') given by sixth heuristics is a ratio of a value in Needleman-Wunsch table, corresponded to that position, to average shift length from (i, j) to (i', j') .

3. The decision-making using some different greedy heuristics simultaneously

Thus, we shall to use not only the heuristics, which forms branch-and-bounds method (BBM) for the considered discrete optimization problem (DOP), but also the other one. Namely, it is heuristics for selecting element, which separate the considered problem into right and left sub-problems. However, we can often replace a heuristics separating algorithm for some other. Moreover, solving a DOP using BBM, it is often desirable to choose one of some separating algorithms, depend upon the solved sub-problem. The various separating algorithms can be selected depend upon dimension of the solved problem, its bound, and also many other descriptions, which are based on the solved DOP.

¹ We have changed here the description of the algorithm given in [10]. The authors are willing to send me the source code when prompted by email.

In classical examples of using BBM for TSP, some good separating algorithms were used (it means, that they give the reasonable results comparing other ones). However, long before, for the BBM-branching various other heuristics were used. Let us mention, for example, the following heuristics for the reduced TSP-matrix: total number of zeroes, sum of minimums for all the rows and columns, sum of some minimum values of considered row and columns multiplied by special “damnation constants”; all these values are computed by the TSP-matrix after reducing and selecting separating element (i.e., separating edge for branching). Probably, the author mentioned here less than 10% of the heuristics used before.

Thus, how can we use the fact, that in various situations (i.e., in various sub-problems of the same considered DOP) different heuristics relatively better are used? (This question can be putted for both exact and unfinished algorithms.) We need to make a decision for selecting the separating element for branching. We have information of various experts, i.e., of various special heuristics, so called predictors (or estimators). The predictors often give discrepant information, and we have to average it in a special way. Unlike all the algorithms published before, the authors, like programming nondeterministic games, use dynamic risk functions for this thing.

Since various heuristics give values of various units, we have to normalize them for computing the final result. Using the special set of normalizing coefficients (it is adjusted, for example, for genetic algorithms, we shall consider them below) is, probably, a possible method, but it is not the main one for this paper. The authors used only one algorithm in various DOP; that was a special modification of “voting method”, where each of heuristics gives the considered variants of selecting (e.g., for traveling salesman problem, those are zeroes of the reduced matrix, corresponding some edges, which are the candidates for branching). After that, we use special dynamic risk functions for the results of voting.

Thus, selecting edge for branching is constructed for BBM by using dynamic risk functions; see previous sections for details. It is important to note, that the dynamic selecting of the particular risk function is similar to selecting it in programming nondeterministic games, considered in. Since we consider here DOP (not programming nondeterministic games), we have to add here a new heuristics, i.e., one for selecting “current position estimation”, i.e., evaluation of the situation, which is obtained by the solving some DOP using BBM.

Thus, let us have some various heuristics for selecting the element for the next step of BBM (or, generally speaking, for selecting the strategy of solution some DOP). Let each of possible strategies have some various expert evaluations of availability (i.e., let us have some independent expert sub-algorithms, i.e., predictors). Then the concluding strategy could be chosen by maximum of average values. However, let us consider the following example [24]; this example is connected with backgammon programming, because it uses 36 predictors).

Let expert evaluations of availability have values in the segment [0,1]. Let for the 1st strategy, the 1st expert has the evaluations of availability being equal to 1, and other 35 experts have the evaluation 0.055. And for the 2nd strategy, 2 experts have the evaluation 0.95, and other 34 experts have the evaluation 0. Very likely, each user (expert-human) on basis of these values chooses the 2nd strategy.

However, averaging-out by the simplest algorithm (i.e., simply the simple average of expert evaluations) gives 0.081 for the 1st case and 0.053 for the 2nd one; i.e., do we have to choose the 1st strategy?

Let us make computations like to our previous papers, i.e., having the same algorithms of dynamical risk functions (DRF) constructing. For the 1st strategy, we obtain the following risk function:

$$-0.685 \times x_2 + 1.300 \times x + 0.386,$$

and for the 2nd strategy:

$$-0.694 \times x_2 + 1.374 \times x + 0.321.$$

The final values of averaging-out of expert evaluations using these risk functions are 0.111 for the 1st strategy and 0.147 for the 2nd strategy. Therefore, using such algorithms of DRF for special averaging-out of expert evaluations gives “natural” answers.

Remark that twice repeated using of averaging-out (i.e., averaging-out using preliminary values of the first step of DRF-using) chooses the 1st strategy again. However, in the limit we have “natural” answers again. Let us describe these results by the following table; the names of columns are equal to the number of step of averaging-out using DRF (i.e., the number of using algorithm constructing DRF). Then the column 0 is the simple average of expert evaluations), and the column ¥ is the limit value.

	0	1	2	3	4	5	...	¥
1 st strategy	0.081	0.111	0.104	0.106	0.105	0.105	...	0.105
2 nd strategy	0.053	0.147	0.094	0.118	0.106	0.112	...	0.110

Let us remark, that this example is really possible in solving real problems: in the real computations for the mentioned DOP, the situations, when the difference between values of maximum and minimum expert values is more than 0.5 (i.e., more than 50% of the segment of expert values) are very often; for example, for accidental traveling salesman problem having dimension 75 and some of predictors mentioned before, they contain, by statistics of the author, about 10%.

4. Some versions of “triangular” norm of quality definitions for distance metric

So, there are various algorithms to determine the distances between genomes; they can be called algorithms definition of the metric on the set of genomes². However, this raises not only the usual questions about the adequacy of the corresponding mathematical models (from the point of view of the authors, they are usually solved in this domain by experts in biology, [15] etc.), but also on the comparative evaluation of these models. The most important matter in this case appears the following one: can we talk about the effectiveness of such algorithms and the adequacy of these models based on only one analysis matrices

² Mathematical aspects of the correctness of using the concept “metric” in this situation, we are expecting to discuss in a future article.

proximity (distance) between the genomes, without the involvement of biologists? The authors of this paper believe that this question should be answered in the affirmative.

For several different algorithms [4,5,6,7,10], we consider the matrices of distances between the genomes; in our computational experiments (see. below), we used five different algorithms³ and made corresponding distance matrices, in which the number of genomes reached 100.

In this case, we used the following natural philosophy (we have not found analogues in the literature); we give it for the example of human (H), chimpanzee (C) and bonobo (B). According to biologists, Sand B dispersed (had a common ancestor), according to various estimates, about 2–2.5 million years ago (no wonder; the alternative name of B is “pygmy C”, [16]), and H dispersed with both other species 5.5–7 million years ago⁴. In this connection, the following question arises: why H should be closer to B comparing S? or vice versa: why it should be closer to C comparing B? Obviously, the answer to both these questions is negative, i.e., by other words, the explanation of the greater intimacy cannot exist. Therefore, in the matrix of distances between the genomes of all received triangles *should ideally be acute isosceles* ones.

To compare the quality of algorithms for constructing the distance we have offered several versions of “waste” (so called badness) of such “longisosceles” triangles. Apparently, when calculating the badness of the whole matrix for each option, we should always appropriate to summarize all the badness of all possible triangles of the matrices; we make this thing in our work.

So, *in simple cases*⁵, we will assume badness (norm) of the entire sum of the distance matrix, and for the badness of each triangle will apply one of the following 4 options. (We assume everywhere, that the considered triangle has sides a , b and c , moreover $a \geq b \geq c$; the angles are α , β and γ , moreover $\alpha \geq \beta \geq \gamma$.)

1. $(\alpha - \beta) / \pi$.
2. $(\alpha - \beta) / \alpha$.
3. $(a - b) / a$.
4. For the final norm, we consider *separately* “violation of an isosceles” and “violation of an acute-angled”:
 - (A) $1 - \min(b/a, c/b)$;
 - (B) $\max(a - \pi/3, 0) / (2\pi/3)$;
 and the general answer is $(A+B) / 2$.

The maximum value of badness (in each of these four cases) to a triangle may be equal to 1. At the same bad case of algorithms for constructing metrics (i.e., when violation of the triangle inequality occurs) we believe this value from 1 to 2 (also depending on the quantitative characteristics of the violation).

As we noted above, some results of calculations are given below.

5. The preprocessor computing as a special normalization of data

Results and discussion may be presented in separate sections or combined into a single section, whichever format conveys the results in the most lucid fashion. The authors should discuss the significance of observations, measurements, or computations and should also point out how these contribute to the aims indicated in the Introduction. Tables, Figures, and Figure Captions should be embedded within the Section.

In this section we consider another heuristic, which can be considered optional for all heuristics of “violation of an isosceles and of an acute-angled” considered before. For this thing, we consider a function of the type $f(x) = x^\alpha$, where the value α (usually, $0 < \alpha < 1$) is chosen for each matrix of distances (see below some more about selecting α). Where each of the x of distance matrix is replaced by $f(x)$.

To select specific values of α , *improving, from our point of view, the quality of the choice of metrics*, we applied the following considerations. Below, considering the description of the results of computational experiments, we will show, that various heuristics select metrics are relatively different priorities for genomes for “distant” and “close” species; and it is worth noting that such a priority varies little with his study at different rates described above. Attempts to improve the value of these norms (badness) applying some functions of the type $f(x) = x^\alpha$ are unsuccessful: solutions of the corresponding minimization problems given either the maximum or minimum value α (among all the possible ones); it is easy to understand, that in this case, we obtain the matrix of distances between genomes triangles “are closest to the acute-angled isosceles”. There fore, if we really try to improve quality metrics, it is necessary to use a fundamentally different heuristics. For this thing, we were trying to find a function of the above type in which the set of values of the distance matrix, viewed as a distribution of a random variable, is obtained as close as possible to a uniform distribution⁶; in advance, we note *that for different tasks* (i.e., for different concrete matrices of distances), the values of α , obtained by pseudo-optimal real-time algorithms (which are realized by algorithms of [8,9,14] etc.) *are different*.

In this case, we have chosen the goal function on the basis of the entropy maximization ([17] and many others). Specific outcomes associated with the use of this heuristic are given below.

³ Especially note again, that among these algorithms is one of our, the original one.

⁴ It is important to note that the exact values of time such models are not important!

⁵ We note in advance that we will consider a somewhat more complex option.

⁶ Informally that can explain, for example, as follows. We already know, that in our model, genomes of human (H), bonobo (B) and crocodile (C) form a “stretched” acute-angled triangle, which is close to an isosceles one. In this case, the exact values of the lengths H–C and B–C unlikely to be of interest; it is only important, that they are approximately equal. Also unlikely, that the value ratio of the length of H–B to the length of H–C is to be of interest.

6. Some local results of calculations

Table 1. Panin's algorithm.

	Bison bison	Bos taurus	Canis lupus	Drosophila simulans	Felis catus	Gadus morhua	Gallus gallus	His1 virus	Homo sapiens
Bison bison	100	89,32	70,96	40,04	74,2	58,51	57,94	36,64	69,43
Bos taurus	89,32	100	73,62	40,02	71,4	60,78	55,84	36,41	66,27
Canis lupus	70,96	73,62	100	39,32	71,1	59,63	53,61	35,86	63,53
Drosophila simulans	40,04	40,02	39,32	100	43,53	41,51	40,82	39,34	41,42
Felis catus	74,2	71,4	71,1	43,53	100	55,74	58,55	35,55	67,26
Gadus morhua	58,51	60,78	59,63	41,51	55,74	100	53,18	35,08	56,03
Gallus gallus	57,94	55,84	53,61	40,82	58,55	53,18	100	34,19	57,7
His1 virus	36,64	36,41	35,86	39,34	35,55	35,08	34,19	100	41,07
Homo sapiens	69,43	66,27	63,53	41,42	67,26	56,03	57,7	41,07	100

Table 2. Winkler's algorithm.

	Bison bison	Bos taurus	Canis lupus	Drosophila simulans	Felis catus	Gadus morhua	Gallus gallus	His1 virus	Homo sapiens
Bison bison	1	0,8763	0,8614	0,8076	0,8652	0,8376	0,85	0,8032	0,8432
Bos taurus	0,8763	1	0,905	0,7834	0,8791	0,8556	0,8334	0,8442	0,8587
Canis lupus	0,8614	0,905	1	0,7851	0,8765	0,8687	0,8304	0,8438	0,853
Drosophila simulans	0,8076	0,7834	0,7851	1	0,7835	0,7647	0,8176	0,7683	0,7493
Felis catus	0,8652	0,8791	0,8765	0,7835	1	0,857	0,8317	0,8207	0,8555
Gadus morhua	0,8376	0,8556	0,8687	0,7647	0,857	1	0,8151	0,8409	0,8385
Gallus gallus	0,85	0,8334	0,8304	0,8176	0,8317	0,8151	1	0,7838	0,8682
His1 virus	0,8032	0,8442	0,8438	0,7683	0,8207	0,8409	0,7838	1	0,8092
Homo sapiens	0,8432	0,8587	0,853	0,7493	0,8555	0,8385	0,8682	0,8092	1

Table 3. Third algorithm.

	Bison bison	Bos taurus	Canis lupus	Drosophila simulans	Felis catus	Gadus morhua	Gallus gallus	His1 virus	Homo sapiens
Bison bison	1	0,846	0,604220	0,3181985	0,625668764	0,469214183	0,468422307	0,322671569	0,57058362
Bos taurus	0,846	1	0,61892522	0,317847962	0,602974896	0,485864878	0,450071497	0,322071245	0,54511437
Canis lupus	0,604220216	0,61892522	1	0,329786598	0,606326063	0,475700879	0,42999285	0,331161456	0,520353877
Drosophila simulans	0,318198529	0,317847962	0,329786598	1	0,324592863	0,308996167	0,280326501	0,292271663	0,28758525
Felis catus	0,625668764	0,602974896	0,606326063	0,324592863	1	0,451408078	0,475042624	0,335116703	0,563642777
Gadus morhua	0,469214183	0,485864878	0,475700879	0,308996167	0,451408078	1	0,409675882	0,322712027	0,432498802
Gallus gallus	0,468422307	0,450071497	0,42999285	0,280326501	0,475042624	0,409675882	1	0,306720686	0,475869876
His1 virus	0,322671569	0,322071245	0,331161456	0,292271663	0,335116703	0,322712027	0,306720686	1	0,307683023
Homo sapiens	0,57058362	0,54511437	0,520353877	0,28758525	0,563642777	0,432498802	0,475869876	0,307683023	1

Table 4. Winkler's algorithm.

	Bison bison	Bos taurus	Canis lupus	Drosophila simulans	Felis catus	Gadus morhua	Gallus gallus	His1 virus	
Bison bison	1	0,9041	0,7588	0,5315	0,7771	0,6719	0,6723	0,5195	0,735
Bos taurus	0,9041	1	0,7679	0,5307	0,762	0,6824	0,6603	0,5186	0,7186
Canis lupus	0,758	0,7679	1	0,5275	0,758	0,6711	0,644	0,5139	0,6952
Drosophila simulans	0,5185	0,5183	0,5275	1	0,5883	0,5758	0,556	0,5412	0,5604
Felis catus	0,8486	0,8331	0,8486	0,5883	1	0,6477	0,6651	0,5093	0,7142
Gadus morhua	0,6447	0,6554	0,6601	0,5059	0,6477	1	0,6315	0,5095	0,6413
Gallus gallus	0,6572	0,6462	0,6453	0,4977	0,6775	0,6315	1	0,4972	0,6658
His1 virus	0,5051	0,5048	0,5122	0,4819	0,5161	0,5068	0,4972	1	0,5768
Homo sapiens	0,8294	0,8118	0,8043	0,5791	0,84	0,7404	0,7727	0,5768	1

7. Some final results of calculations

Below, we shall call:

- our original algorithm for constructing a metric between genomes by the *first* one (below No. 1, see [10]);
- one of algorithms of M. van der Loo etc. (below No. 2, see [5], used function is `jarowinkler()`) by the *second* one;
- another algorithm of M. van der Loo etc. (below No. 3, see again [5], used function is `stringdist()`) by the *third* one;
- one of algorithms of H. Pages etc. (below No. 4, see [6], used function is `stringDist()`) by the *fourth* one;
- another algorithm of H. Pages etc. (below No. 5, see [6], used function is `pairwiseAlignment()`) by the *fifth* one.

Let us denote, that algorithms No. 4 and No. 5 are “non-symmetrical” ones, and, when filling in the distance matrix, we used half-sums of the two obtained values. Also let us note that the violations of the triangle inequality were recorded only as a result of the algorithms No. 4 and No. 5; however, in the case of “distant” species, we had a few such results: approximately, 1 case per 2000 examined potential triangles.

For further counting, we firstly have randomly chosen genomes of 100 representatives of the species, given in [18] (the case of considering “distant species”⁷). Some results of computations (the table 100x100, i.e., $100 \cdot 99 / 2 = 4950$ values, making $(100 \cdot 99 \cdot 98) / (2 \cdot 3) = 161700$ triangles) are given below in Table 1, where:

- the rows are number of the algorithms (as we write before);
- the columns: approximate time of the creation of the distance matrix (for making all the 4950 values, CPU clock speed is approximately 2 GHz); number of violations of the triangle inequality (in average 1000 launches for triangles); middle badness, counted for all the algorithms 1–4 counting badness for each triangle, see Section 3.

All values badness we give up to 3 decimal places (the time of algorithms for constructing matrices was recorded some less accurately). In all tables, we celebrated the best metric for the considered norm (it is singled twice) and also the 2nd place (it is singled once).

Table 5. “Distant” species.

No.	time (h)	violations	badness-1, $(\alpha - \beta) / \pi$	badness-2, $(\alpha - \beta) / \alpha$	badness-3, $(a - b) / a$	badness-4, $(A + B) / 2$
1	27	0	0,0372	0,0822	0,0416	0,196
2	2.1	0	0,0954	0,197	0,0926	0,252
3	2.3	0	0,345	0,476	0,163	0,468
4	28	0.37	0,0416	0,0907	0,0469	0,176
5	28	0.38	0,0549	0,116	0,0556	0,214

As we can see, the algorithm implemented by our group, the majority of rules is optimal. And there is very important to remark (it follows from the above), that *heuristics that were used to create this algorithm had absolutely no connection with the heuristics used to describe the “triangle rules” defined below.*

Secondly (the case of considering “near” species), we also randomly have chosen genomes of human and 5 apes (bonobo, chimpanzee, gorilla, orangutan, gibbon), which are also given in [18]. In this case, each species taking 4-5 representatives (of 28 genomes); for the human, we took the genomes of different races. Some results of computational experiments are given in Table 2, where, unlike Table 1, we failed build time. Furthermore, due to the small total number of triangles (less than 5000), we have brought the number of violations of the triangle inequality (rather than relative values of this quantity).

⁷ All lists of specific species corresponding genomes taken primarily from the site [18]. The authors are ready to send the obtained values of distance matrices, as well as the source code, by e-mail (at your request). We are also ready to send the detailed results of calculations of the badness, including not only the averaged, but all produced in the process value.

Table 6. “Near” species.

No.	violations	badness-1, $(\alpha-\beta) / \pi$	badness-2, $(\alpha-\beta) / \alpha$	badness-3, $(a-b) / a$	badness-4, $(A+B) / 2$
1	0	0,0757	0,152	0,0645	0,364
2	1	0,0333	0,0687	0,0302	0,215
3	1	0,514	0,622	0,170	0,582
4	32	0,0595	0,122	0,0496	0,341
5	39	0,0741	0,151	0,0615	0,350

As we can see, the relative number of violations of the triangle inequality significantly increases. In addition, our original distance metric between genomes is now not optimal.

Thirdly, we used “preprocessor” algorithms as previously described. It should be noted that the application of these auxiliary algorithms decreased value of badness almost all cells; however, it was *not the goal* of this algorithm. Besides, “leaders little changed”, i.e., our algorithm for constructing the metric (string 1) shows better results (comparing the absence of these auxiliary algorithms); however, the latter fact is just and can be explained by “tuning” the algorithm 1 for its use for a greater range of values. The results of computational experiments are given below in Table 3.

Table 7. “Near” species. (after pre-application of “preprocessor” algorithm).

No.	violations	badness-1, $(\alpha-\beta) / \pi$	badness-2, $(\alpha-\beta) / \alpha$	badness-3, $(a-b) / a$	badness-4, $(A+B) / 2$
1	0	0,0522	0,121	0,0527	0,351
2	0	0,0314	0,0692	0,0290	0,205
3	0	0,501	0,600	0,154	0,580
4	12	0,0527	0,122	0,0482	0,323
5	14	0,0732	0,150	0,0608	0,320

And fourthly, we applied the same algorithm to the genomes of human races (white man, yellow man, black man, bushman, Australian man). In this case, each race took 3–4 representatives (total 18 genomes). Some results of calculations are given in Table 4, where the columns mean the same as in Table 2.

Table 8. Races of human.

No.	violations	badness-1, $(\alpha-\beta) / \pi$	badness-2, $(\alpha-\beta) / \alpha$	badness-3, $(a-b) / a$	badness-4, $(A+B) / 2$
1	17	0,140	0,243	0,0924	0,325
2	29	0,119	0,173	0,0359	0,342
3	30	0,420	0,527	0,187	0,493
4	30	0,119	0,218	0,0880	0,313
5	26	0,129	0,229	0,0881	0,306

We could make a lot of comments to the values listed in this table; let us consider only the main one. A relatively large number of violations of the triangle inequality (and consequently, significantly larger values of badness, at its counting on any of the rules), is apparently due to the large number of people crossing concrete already after the separation of races. I.e., apparently, these algorithms should not be used to the genomes of subspecies (without their further modifications).

However, despite this fact, we are going to publish some further improvement of our original algorithm for constructing metrics, as well as our approach to the description of the badness. Besides, different algorithms for constructing metrics may be more appropriate with respect to different situations.

8. Conclusion. Some future ways for research

In this section, we look briefly at some algorithms that are going to be published in subsequent papers.

As a possible connection between the two approaches to solving problems biocybernetic and the traveling salesman problem (at first, so called its pseudo-geometrical version, see [8,19] etc.) we may call not only the above-mentioned multi-heuristic approach to the problems of discrete optimization, but also so called algorithms for pseudo-placing dots in k -dimensional Euclidean space [19]. These auxiliary algorithms improve the performance of other our algorithms. Moreover, algorithms similar to ones used by us in [13,14] could also be considered as such auxiliary algorithms; they some improve algorithms described in this article. Described in these articles use the risk functions of this direction is also, and we apply the special applications of the well-known “3 sigma rule”.

Besides, one of the most frequently discussed in biocybernetics problems is that the recovery of the distance matrix, when we know a part of the completed element only [11, 20]. Using the same “triangular norm”, we propose an original algorithm for such recovery; we are going to write about it in the near future.

Acknowledgements

The reported study was funded by RFBR according to the research project № 16-47-630829.

References

- [1] Gusfield D. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge, 1997; 631 p.
- [2] Toppi J, De Vico Fallani F, Petti M, Vecchiato G, Maglione AG, Cincotti F, Salinari S, Mattia D, Babiloni F, Astolfi L. A new statistical approach for the extraction of adjacency matrix from effective connectivity networks. IEEE Engineering in Medicine and Biology Society (EMBC) 2013; 3-7: 2932–2935.
- [3] Torshin IYu. Bioinformatics in the Post-Genomic Era: The Role of Biophysics. Nova Biomedical Books, NY, 2006.
- [4] Winkler WE. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Survey

- [5] Van der Loo MPJ. The Stringdist Package for Approximate String Matching. *The R Journal* 2014; 6: 111–122.
- [6] Pages H, Aboyou P, Gentleman R, DebRaoy S. Biostrings: String Objects Representing Biological Sequences and Matching Algorithms. R package version 2.10.1, 2009.
- [7] Morgan M, Lawrence M. ShortRead: Base classes and methods for high-throughput short-read sequencing data. R package version 1.0.6, 2009.
- [8] Melnikov BF. Multiheuristic approach to discrete optimization problems. *Cybernetics and Systems Analysis* 2006; 3: 335–341.
- [9] Melnikov BF. Discrete optimization problems some new heuristic approaches. *Proceedings – Eighth International Conference on High-Performance Computing in Asia-Pacific Region, HPC Asia 2005 8th International Conference on High-Performance Computing in Asia-Pacific Region, China Computer Federation, Beijing, 2005: 73–80.*
- [10] Makarkin S, Melnikov B, Panin A. On the metaheuristics approach to the problem of genetic sequence comparison and its parallel implementation. *Applied Mathematics (Scientific Research Publishing)* 2013; 4(10): 35–39.
- [11] Eckes B, Nischt R, Krieg T. Cell-matrix interactions in dermal repair and scarring. *Fibrogenesis Tissue Repair* 2010; 3-4. DOI: 10.1186/1755-1536-3-4.
- [12] Carr RW, Hennessy J L. WSCLOCK – a simple and effective algorithm for virtual memory management. *SOSP '81 Proceedings of the eighth ACM symposium on Operating systems principles, 1981: 87–95.*
- [13] Melnikov BF. Heuristics in programming of nondeterministic games. *Programming and Computer Software* 2001; 5: 277–288.
- [14] Melnikov B, Radionov A, Moseev A, Melnikova E. Some specific heuristics for situation clustering problems. *ICSOFIT, Technologies, Proceedings 1st International Conference on Software and Data Technologies 2006: 272–279.*
- [15] Foley J. Fossil Hominids: mitochondrial DNA 2011; URL: <http://www.talkorigins.org/faqs/homs/mtDNA.html>
- [16] Frans BM. *Bonobo: The Forgotten Ape.* University of California Press, 1998, 224 p.
- [17] Popkov YS. Substantiation of the entropy maximization method for problems of image restoration from projections. *Automation and Remote Control, 1995; 56(1): 77–82.*
- [18] NCBI: nucleotide database, 2014, URL: <http://www.ncbi.nlm.nih.gov/nucleotide>.
- [19] Makarkin SB, Melnikov BF. Stochastic Optimization in Informatics. Geometric methods for solving the traveling salesman problem pseudo-version 2013: 54–72. (in Russian)
- [20] Midwood KS, Williams LV, Schwarzbauer JE. Tissue repair and the dynamics of the extracellular matrix. *The International Journal of Biochemistry & Cell Biology* 2004; 36(6): 1031–1037.
- [21] Shao M, Lin Y, Moret B. An Exact Algorithm to Compute the DCJ Distance for Genomes with Duplicate Genes. *Research in Computational Molecular Biology, Lecture Notes in Computer Science Volume 2014; 8394: 280–292.*
- [22] Melnikov B, Radionov A. On the decision of strategy in non-deterministic antagonistic games. *Programming and Computer Software* 1998; 24(5): 247–252.