Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 12
December 2023

Algorithms for Calculating DNA Distances Based on the Pair Correlation

Boris Melnikov
Shenzhen MSU – BIT University, Shenzhen, China

Abstract

We consider the generalization of the results of comparing algorithms for calculating DNA distances based on pair correlation. We believe that the “good” algorithms for counting the distances between a pair of given DNA sequences will give a value close to some optimal number; of course, we do not know this number in advance, but it is not necessary for this approach. Therefore, the “best” algorithms give badness values close to 0 for different triangles, and the value of the pair correlation of all triangles of the matrix (their order is N3) is close to 1. Possible deviations from this value for some pair of genomes (associated, for example, with a larger than usual number of mutations in one of these two species) should lead to approximately the same change in the value of badness for any of the triangles in which this pair forms a side.

Therefore, approximately the same increase in the value of badness is formed. Then we believe that considering all the resulting triangles, we have them in ascending order of the value of badness; and we believe that for two “good” distance calculation algorithms should get a relatively large value of the pair correlation. Conversely, the “bad” algorithms that do not correctly estimate genome proximity values are very unlikely to give 0 badness value for the triangle, as well as close badness values if the latter is relatively far from

Therefore, the “good” algorithms in the transition to a very large number of triangles should give a small value of the pair correlation coefficient.

Keywords: DNA distance, distance matrix, triangular metrics, pair correlation.

1. Introduction

In practice, it is often necessary to calculate in a special way certain distances between sequences of symbols of different nature. Here is some reasons and corollaries from the author's paper [1] on this occasion.

Similar algorithms are often used in bioinformatics; they represent a separate, very important type of task for calculating the distance between the sequences, exactly, the problem of determining the distance between genetic sequences. The main difficulty that arises when calculating the distance between such sequences is their very long length. For example, even for a very short human’s mitochondrial DNA (mtDNA) sequence length exceeds 16 000 characters, and the total length of human DNA exceeds 3 000 000 000 characters, see [2] and many others books and papers.

Because of this, algorithms calculating the exact value of the distance between two sequences are not applicable, and for assessments distances between such chains, it is necessary to use heuristic algorithms. Among a very large number of papers on this topic, we mention only two our papers: [3, 4]; we can say that these papers consider the areas of work, which will be discussed later in the current paper.

Thus, to determine the distance between genomes, we need heuristic algorithms. There are various similar algorithms, but they give different results with the same input data. For ex- ample, not only in popular science, but also in the scientific literature, the distance between the genomes of humans and chimpanzees ranges from 98% to 99.5%, distance between humans and mice is in an even larger range... In other words, the question is which estimate is more correct? All these values can be easily verified using the algorithms that we always cite in our papers. Therefore, there are various similar algorithms, but their obvious disadvantage is to obtain slightly different results when using different heuristic algorithms applied to calculating the distance between the same pair of DNA strings. Then the task arises of assessing the quality of the metrics (distances) used, and based on the results obtained in solving this problem, it is possible to draw conclusions about the applicability of a specific algorithm for calculating

distances to various applied research. A possible approach to determining the quality assessment of metrics was given in [3, 4], we repeat its description below in this paper.

Therefore, the important problem arises of estimating the quality of the metrics used; and based on the results obtained in solving this problem, one can draw conclusions about their applicability. As a variant of "evaluating multiple estimates" (in other words, "heuristics for comparing heuristics"), author started working on the following method a long time ago. As an illustration, we consider human, chimpanzee, and bonobo; according to biologists, the ancestors of chimpanzees and bonobos diverged about 2 500 000 years ago, and the ancestors of humans with both of them diverged about 7 000 000 years ago (while the exact values are not particularly important). The next question arises why a man (his genome) should be closer to a chimpanzee than to a bonobo (or, conversely, to a bonobo); obviously, there can be no answer to this question. The same can be considered true for any three species: after all, always first one of them diverges from the other two, and then these two diverge from each other; at the same time, almost nothing will change if all three species diverged at the same time. Therefore, if we consider triangles with sides displaying distances between views (this is the same "proximity", only "vice versa", subtracted from 100%), then these triangles should turn out to be acute-angled isosceles.

2. Some Results of Comparing Algorithms for Calculating DNA Distances Based on a Triangular Norm

In this section, we use only one of the ones discussed above triangular norms, exactly the norm number 0; it seems to be somewhat more adequate than the others. For it, we present some results of comparing algorithms for calculating DNA distances. However, it would be more accurate to say that we present an approach to obtaining such results; this approach is based on the application of the relevant metrics described in the section.

Once again, the example given in the introduction about humans, chimpanzees and bonobos can be generalized *to any three species*; that is why the triangular norm defined in [3, 4] can be considered. At the same time, we add that, of course, mutations in genomes accumulate more or less proportionally to the number of past generations, it is clear that humans have fewer of them than in the same period in chimpanzees; but conditionally, since the genomes of chimpanzees and bonobos separated from the human genome at the same time, then the number of mutations they should be closer to each other than from both of them to the person. This additive also justifies the possibility of using a triangular norm.

Thus, let us first give a table of the results of previously performed calculations, after which we shall give detailed comments on it.

"Near" species, after preprocessing							
No	Time (h)	Vio	(0) (α–β) / γ	(1) (α–β) / π	(2) (α–β) / α	(3) (a–b) / a	(4) (A+B) / 2
1	27	0	**0.155**	**0.0522**	**0.121**	0.0527	0.351
2	2.1	0	**0.101**	**0.0314**	**0.0692**	**0.0290**	**0.205**
3	2.3	0	1.331	0.501	0.600	0.154	0.580
4	28	12	0.155	0.0527	0.122	**0.0482**	**0.323**
5	28	14	0.200	0.0732	0.150	0.0608	0.320

- The auxiliary algorithm "preprocessing" is explained in [4].
- We have provided one of the four tables given also in [4], but we added a column corresponding to the first table, as well as a column with a new variant of bad- ness; those are columns (0) and (5).
- We considered "near" animal species and used prepro- cessing (see [4] for some details); this is noted in the subtitle of the table.

- Values A and B are explained also in [4]. Now let us describe the other organization of the given table.
- The numbers of heuristic algorithms for calculating distances between chains are arranged along the lines; full information about the algorithms can be found also in [4], here we only note that line 1 is our own version of the Levenshtein distance, and line 2 is the Needleman – Wunsch algorithm.
- Column "Time" includes the time for filling in the table of dimension 30×30 with the algorithm in question (to get all the values of distance matrix, the processor clock speed is ~2 GHz). It is important to note that even for such a dimension, the time is quite long.
- Column "Vio" includes the average number of violations of the triangle inequality for all generation problem instances. We recalculated this number "per 1 000 elements" and rounded the result to integers; therefore, the value 12 corresponds approximately to 1.2%, and higher accuracy, apparently, is not interesting here. Note that these violations of the triangle inequality exist for standard algorithms for calculating distances between genomes.
- The remaining columns have the same meaning as in [4]. At the same time, the set of badness options is slightly different; but this is not important, the options most interesting for comparing are available in both tables.

3. The Generalization of the Results of Comparing Algorithms for Calculating DNA Distances Based on the Pair Correlation

In our "pair correlation approach", we assume the following thing. The "good" algorithms for counting the distances between a pair of given DNA sequences will give a value close to some optimal number. Therefore, the "best" algorithms give badness values close to 0 for different triangles; of course, we do not know this number in advance, but it is not necessary for this approach.

Besides, the value of the pair correlation of all triangles of the matrix (their order is $\sim N^3$) is close to 1, see [5] etc. Moreover, possible deviations from this value for some pair of genomes (associated, for example, with a larger than usual number of mutations in one of these two species) should lead to approximately the same change (the same increase) in the value of badness for any of the triangles in which this pair forms a side.

Therefore, approximately the same increase in the value of badness is formed. Then we believe that considering all the resulting triangles (recall that they are formed quite a lot, a few thousand for the original matrices of dimension about 30), we have them in ascending order of the value of badness; and we believe that for two "good" distance calculation algorithms should get a relatively large value of the pair correlation. Conversely, the "bad" algorithms that do not correctly estimate genome proximity values are very unlikely to give 0 badness value for the triangle, as well as close badness values if the latter is relatively far from 0. Therefore, the "good" algorithms in the transition to a very large number of triangles should give a small value of the pair correlation coefficient.

The concrete variants of the coefficients of pair correlation is below. We used Spearman's and Kendall's correlation coefficients [5], and also a coefficient developed by us.

The last one can be called the *swapping rank correlation coefficient*. For it, we propose to use the correlation coefficient calculated by the swapping method:

$r = 1 – (2 \cdot Q) / (n \cdot (n–1))$,

where:

- Q is the number of permutations required to convert the second row to the first, which is presorted in ascending order of its values,
- $(n \cdot (n–1)) / 2$ is the number of exchanges of the worst-case sorting algorithm.

For this definition of coefficient, the bubble sorting algorithm was applied, see [6] etc.

For the computational experiment, we considered the sets of species of 30 animals.

"Far species" of animals	
Algorithm	Badness
Jaro – Winkler (JW)	70.8
Damerau – Levenstein (DL)	57.7
Needleman – Wunsch (NW)	47.0
Smith – Waterman (SW)	46.0

Let us only explain that the total value of badness 70.8 gives the mean value of the average badness of triangles 70.8 / 3276 ≈ 0.0216; this value corresponds to “a good enough” triangle (from the point of view of this paper). The examples of some “good” and “bad” triangles and corresponding values of badness were considered in [4]. Note that according to the metric (0) we are formulating before, this value practically coincides with the badness of the triangle with the sides 19, 18 and 17; of course, the last triangle is visually indistinguishable from an equilateral one.

Next, let us calculate the pair correlation for the pairs of metrics; as the source data for all the pairs, we chose 34 species of monkeys, therefore, we considered 34 · 33 · 32 / 6 = 5984 triangles. For three variants of algorithms (two of [5], as well as so called “swapping correlation”), we obtained the following results.

Pairs of algorithms for metrics and algorithms of pair correlation (34 species, 5984 triangles)			
Pairs of algorithms	Spearman correlation	Kendall correlation	Swapping correlation
JW, NW	0.120	0.065	0.533
JW, SW	0.121	0.069	0.535
JW, DL	0.185	0.110	0.556
NW, SW	0.622	0.483	0.742
NW, DL	0.908	0.753	0.877
SW, DL	0.770	0.622	0.812

As we can see, the JW algorithm by badness is the worst, and in all 3 cases of pair correlation, those variants of pairs where JW is present give the minimum (bad) correlation values. That is, our assumption is fully confirmed.

For the remaining 3 pairs of metrics, apparently, no conclusions can be made yet (the amount of data from computational experiments is insufficient). However, the very first assumption is that SW is the good algorithm, and the other two (i.e. NW and DL) are some more better, and, besides, “are ideologically close”.

A third alternative method for evaluating the quality of various algorithms can also be given; this method is per- formed in [7] based on the so-called ultrametric space, which has the so-called enhanced triangle inequality; the results of the estimates of this third method based on the analysis of data from the mitochondrial DNA of monkeys also give precedence to the Needleman – Wunsch algorithm. Thus, we see that the *three completely different* “heuristic algorithms for comparing heuristic algorithms” (based on completely different concepts) *give approximately the same results*.

At the same time, similarly to “algorithms for comparing algorithms”, we can also calculate “pair correlation for comparing different variants of pair correlation”; of course, this it is still a very vague topic (after all, it is ideally necessary to consider different tables, types near and far, etc.), but, as can be seen from the last table, in the options we are considering, we always get the maximum possible value 1. It is worth noting that this possible direction of future work is related to the remark made above in this section that, despite not the best values of pair correlation for a pair of Needleman – Wunsch and Smith – Waterman algorithms, the absolute value of this pair for the swapping correlation variant in practice is almost indistinguishable from two “better” pairs of algorithms.

4. Conclusion

The “usual pair correlation” of the algorithms of the last table, in the future we can also consider the “pair correlation with penalties”. For it, there is a “penalty for exchanging elements” equal to 0 if no exchange is made and equal to 1 for a predetermined maximum value of the difference of ex- changed elements arranged in the wrong order; if this difference is exceeded, it is also convenient to consider the penalty equal to 1. However, this topic is beyond the scope of this paper, it may be the subject of further re- search.

This work is supported by a grant from the scientific program of Chinese universities “Higher Education Stability Support Program” (section “Shenzhen 2022 – Science, Technology and Innovation Commission of Shenzhen Municipality”) – 深圳 市2022年高等院校定支持划助目.

References

[1] B. Melnikov, Y. Zhang, and D. Chaikovskii, “An inverse problem for matrix processing: an improved algorithm for re- storing the distance matrix for DNA chains,” Cybernetics and Physics, vol. 11, no. 4, pp. 217–226, 2022.

[2] C. Calladine, H. Drew, B. Luisi, and A. Travers, Understanding DNA: The Molecule and How it Works. NY: Academic Press, 2004.

[3] B. Melnikov, E. Melnikova, S. Pivneva, and M. Trenina, “An approach to analysis of the similarity of DNA-sequences,” CEUR Workshop Proceedings, vol. 2212, pp. 63–72, 2018.

[4] B. Melnikov, S. Pivneva, and M. Trifonov, “Various algo¬rithms, calculating distances of DNA sequences, and some computational recommendations for use such algorithms,” CEUR Workshop Proceedings, vol. 1902, pp. 43–50, 2017.

[5] M. Lagutin, Visual mathematical statistics. Moscow: Binom. Laboratory of Knowledge, 2012. (In Russian.)

[6] N. Wirth, Algorithms + Data Structures = Programs. NY: Prentice Hall, 1976.

[7] B. Melnikov, S. Pivneva, and M. Trifonov, “The evaluation of algo¬rithms for calculating the distance of DNA strings,” Uni¬ver¬sity proceedings. Volga region. Physical and math. sci. Mathematics, vol. 34, no. 2, pp. 57–67, 2015. (In Russian.)