DNA Sequence Alignment Method Based on Trilateration

Veska Gancheva(✉) and Hristo Stoev

Faculty of Computer Systems and Control, Technical University of Sofia,
8 Kliment Ohridski, 1000 Sofia, Bulgaria
vgan@tu-sofia.bg, hristomihaylovstoev@gmail.com

Abstract. The effective comparison of biological data sequences is an important and a challenging task in bioinformatics. The sequence alignment process itself is a way of arranging DNA sequences in order to identify similar areas that may have a consequence of functional, structural or evolutionary relations between them. A new effective and unified method for sequence alignment on the basic of trilateration, called CAT method, and using C (cytosine), A (adenine) and T (thymine) benchmarks is presented in this paper. This method suggests solutions to three major problems in sequence alignment: creating a constant favorite sequence, reducing the number of comparisons with the favorite sequence, and unifying/standardizing the favorite sequence by defining benchmark sequences.

Keywords: Bioinformatics · DNA · Sequence alignment · Trilateration

1 Introduction

Main task in biological data processing is searching of a similar sequence in database [1]. Algorithms such Needleman-Wunsch [2] and Smith-Waterman [3], which accurately determine the level of similarity of two sequences, are very time consuming applying them on large dataset. In order to increase the searching in large database, scientists apply heuristic methods, which significantly accelerate the searching time, but quality of the results decreases. FASTA is a DNA and protein sequence alignment software package introducing heuristic methods for aligning a query sequence to entire database [4]. BLAST is one of the most widely used sequence searching tool [5]. The heuristic algorithm it uses is much faster than other approaches, such as calculating an optimal alignment. This emphasis on speed is vital to turning the algorithm into the practice of the vast genomic databases currently available, although the following algorithms may be even faster. BLAST is more time efficient comparing with FASTA, searching only for the more significant sequences but with comparative sensitivity. Even parallel implementation of above algorithms is limited by the hardware systems [6–9]. A metaheuristics method for multiple sequence alignment, currently used for increasing the performance, have adopted the idea of generating of sequence favorite, after that all other sequences from the database are comparing to the favorite sequence [10]. On this way the sequence favorite turns into a benchmark for the other sequences in the database. Using this approach some problems occurs like the case of entering

I. Rojas et al. (Eds.): IWBBIO 2019, LNBI 11466, pp. 271–283, 2019.
https://doi.org/10.1007/978-3-030-17935-9_25

new data into the database or deleting some of the existing records. Since the sequence favorite is generated based on the records: (1) Changing the data leads to recalculation the sequence favorite. (2) Each of sequences at the database have to be compared again with the new sequence favorite in order to get new result, and this takes time and resources for the calculation. (3) There is a different favorite sequence for each database, and it can lead problems in merging different databases, especially in big data – collection of multiple different database structures and access.

In order to enhance existing heuristics algorithms' idea, improvements are proposed in each of the following fields:

1. Constant sequence favorite – i.e. it should not depend on the data set and should remain unchanged in case data set is modified.
2. Avoid comparisons or reduce the number of comparisons with the favorite sequence in the database searching (for each sequence we apply complicated algorithm for comparison with sequence favorite).
3. Unify/standardize of sequence favorite for all databases.

The goal of this paper is to present a new effective and unified method for sequence alignment on the basic of trilateration method. This method suggests solutions to three major problems in sequence alignment: (1) creating a constant favorite sequence, (2) reducing the number of comparisons with the favorite sequence, and (3) unifying/standardizing the favorite sequence by defining benchmark sequences.

2 An Effective and Unified Method for DNA Sequence Alignment Based on Trilateration

If we introduce some kind of coordinate system or find 3 or more benchmarks, then with the help of analytic geometry or in particular trilateration, we could fix the position of the points one against another, which actually represents the similarity between the database records. Also, the necessity of calculation of the sequence favorite will disappear. The idea of a sequence favorite is to find a starting point – benchmark, based on which to compare and analyze the rest of the records in the database. From a mathematical point of view, the favorite sequence can be represented as a function of N variables (in the case of DNA, the variables are the 4 bases: adenine, thymine, guanine and cytosine). Then we can represent the rest of the records in the database again as a function of the same variables. In that case, similarity comparison would be represented by the distance between each of the sequence to the sequence favorite. In other words, the location of the point representation, described by the function of the sequence, is localized against another point defined by the function of the favorite sequence. Representing each record from the data base with a point, will form a cloud of points, and point representation of sequence favorite, must be somewhere in the middle of this cloud. Since we do not have a specified coordinate system, each data base would form its own could of points with own center and merging of those data base will be very hard. If we introduce standardized coordinate system, applicable for all data basses, it would be possible, with the aid of elementary analytical geometry or trilateration, to determine the positions of the points relative to each other,

which will reflect the degree of similarity between the records in the database and will be the same across database. Also, the need of calculate the favorite sequence will be dropped. Since DNA is built of only 4 bases, we can compose coordinate system with benchmark of endless sequence generated from each of the base. This will allow us to apply principals of trilateration, in order to fix the position of each sequence of the database against the coordinate system. Assess faster similarity of two or more sequence, and if we should process them further with more accurate algorithm (Similar method is used in Global Positioning System GPS [11]).

Trilateration is a method for positioning of objects using circle geometry. This method uses the known position of two or more reference points and measure the distance between the object and each of the reference points [12, 13]. In order to use trilateration, at least three reference points are required to determine accurately the point position in the plane. Trilateration is a method similar to triangulation, using angular measurements and a certain distance for position determining (Fig. 1).

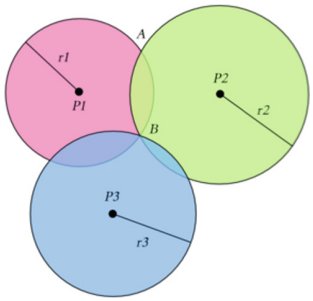

Fig. 1. Schematic overview of trilateration method

The relative position of point B relative to the points P1, P2 and P3 is determined by measuring the distance r1, r2 and r3. The distance r1 determines the position as a point of a circle with a radius r1 and center P1. r2 reduces opportunities position to be one of the crossed letters A or B. The third measure r3 determine the exact coordinates of the point. More than three measurements can be made in order to reduce the error. If applied similar reasoning to compare DNA data, points P1, P2 and P3 will be the benchmark sequences on each of the bases. Here a question arises "Once the bases are 4 why do we have 3 benchmark sequences?". In the structure of DNA there are certain rules that bases can connect: A + T; T + A; G + C; C + G, so that an "A" on one string of DNA will "connect" successfully only with "T" on the other string. It should be noted that the order is important: A + T is not equivalent to T + A and C + G is not the same as G + C. As a consequence of this rule, it can be stated that knowing one half of the helix of DNA, automatically can be reproduced the other. This means that from the mathematics viewpoint is not necessary to examine the entire helix. Or only benchmark of two bases would be enough for the accuracy of the results from a biological viewpoint, but we also introducing the third benchmark for greater precision.

Paying attention to the notes exposed a few lines above ("order is important: A + T is not equivalent to T + A") means that the generating of basic benchmarks as an infinite number of single basis is not good idea, because in fact position of the basis is matter and if the benchmark is formed form single basis we will result of simple counting of that basis of the processed sequence. If we write every base with the letter of the Latin alphabet – ACGT and repeat this string endlessly, we will get first proposed a benchmark sequence. It will look as following:

ACGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT……

It can be called A-benchmark. In order to obtain good results by trilateration, it is advisable to have a uniform distribution of the benchmarks. Considering this recommendation and the way bases bound, the next benchmark sequence can be derived. It will be the second part of the hypothetical DNA created from A – A-benchmark or the benchmark should be supplemented to create a full DNA helix. Thus it is introduced exactly mirror allocation of A-benchmark, which provide more accurate results for the implementation of trilateration. Obtained following a benchmark sequence:

ACGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT**A**CGT… – A-Benchmark

TGCA**T**GCA**T**GCA**T**GCA**T**GCA**T**GCA**T**GCA**T**GCA**T**GCA… – T-Benchmark

Thus, the second benchmark sequence could be named T-Benchmark. For a more precise application of the method of trilateration it is necessary to introduce a third benchmark, preferably equidistant from the previous two. Since two of the bases are already in use, to generate the next benchmark should be chosen between the other two, but at the same time the third benchmark have to be equal distant from both A-benchmark and T- benchmark.

Ideally, it should be 50% identical to the A-benchmark and 50% identical to the T-benchmark. To achieve this requirement, there is provided a sequence of bases where the two coincide in position with the respective bases of the A-benchmark and two bases coincide in position with T-benchmark. Or third benchmark would type:

ACGTACGTACGTACGTACGTACGTACGTACGTACGT...– A-Benchmark

TGCATGCATGCATGCATGCATGCATGCATGCATGCA… – T-Benchmark

TGCATGCATGCATGCATGCATGCATGCATGCATGCA…– medial Benchmark 1

But if applied similar reasoning T-Benchmark against A-Benchmark is obtained:

ACGTACGTACGTACGTACGTACGTACGTACGTACGT...– A-Benchmark

TGCATGCATGCATGCATGCATGCATGCATGCATGCA… – T-Benchmark

TCGATCGATCGATCGATCGATCGATCGATCGATCGA… – medial Benchmark 2

Where and how to fit median benchmarks in the method of trilateration?

In the process of trilateration than centers of circles, which we will attempt to represent with reference sequences, we should introduce equivalent radiuses. Since our task is to identify similarity between sequences, it is reasonable, radius to be equated with the grade of similarity. For the degree of similarity, a percentage of complete alignment of the bases

from the benchmark sequence to the test sequence is selected. In order to calculate this rate, the benchmark sequence is limited to the size of the test sequence and thus calculates the percentage of matches. So, the number of matches to the full length of the test sequence will give wanted percentage of matches. The algorithm is with linear complexity and will always give the same result for the same sequence. This means that these calculations can be done once for all sequences in the database, then can be stored as sequences meta data and to be used in the study, which is a good solution for the problem (2).

Let's go back to the radius and how the degree of similarity must be applied in this case. The closer to the center of the circle is the benchmark sequence, the degree of matching is greater. When the match is 100%, the examined sequence is in the center of the circle. At this way the radius in absolute value is 1, as in the center of the circle, the distance is 1 and in the periphery is 0. The center of the circle of A-benchmark should lie on the periphery of the circle of T-benchmark and the center of the circle of T-benchmark should lie on the periphery of the circle of A-benchmark (Fig. 2).

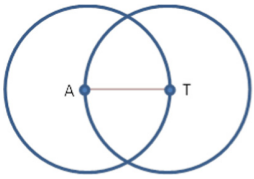

Fig. 2. Intersection between A-benchmark and T-benchmark area of sameness

This definition is used to determine the location of the medial benchmarks to A-benchmark and T-benchmark. In essence the medial benchmarks are 50% identical with A-benchmark and 50% identical with T-benchmark, which means that the benchmark should lie in the middle of the radius of the circle describing the similarity with A-benchmark and T-benchmark. That is, the middle benchmark is lying on the line between the centers of the two circles. This fact actually does not help us locating by the trilateration method, because the angle between the intersection point and the center of any of the circles is 0 degrees, but it raises an interesting question.

There are two middle benchmarks, each of which has an intersection point with the line created by the centers. These points of intersection are the middle of the intercept between the two centers, and they are perpendicular to the line passing through the intercept. Mathematically, this should mean that both middle benchmarks coincide, but this is not true. Then what creates this paradox in mathematics? The answer lies in biology. At the beginning of this article, we noted that the order of base pairing is important. This characteristic is expressed in the mathematical models applied to the biological data. In the first case there is alignment - A-benchmark against T-benchmark (left to right) with middle benchmark 1, in the other case the order is T-benchmark against A-benchmark (left to right) with middle benchmark 2.

Regarding trilateration it is not good to select three points lying on the same line. Therefore we continue to seek a third point, which will help to implement trilateration with certain accuracy. In this case, would it be possible to seek "equation" of the intersections of the circles described by A-benchmark and T-benchmark?

The radius has been defined such that the periphery of the circle is equal to 0% match. This means that the intersections in the circles should be 0% similar to the A- and T-benchmarks. Since the bases are only 4 and already two specific layouts were used and in the same time we seek for arrangements that do not match with the positions of the existing ones, we can easily write the following arrangements:

CTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAG

GATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATC

CATGCATGCATGCATGCATGCATGCATGCATGCATGCATGCATG

GTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTAC

……

Which of the above sequences are the targeted ones? The targeted sequences should meet another one condition. Let's try to find conditions that point M from Fig. 3 must satisfy. Pint M point is intersection of A-benchmark and T-benchmark circles, which meant that it should not match at all any of A and T benchmark. It is a peak of the ATM triangle, and this triangle is equilateral triangle, all sides are equal to the radius. The height from peak M to line AT, is a median of that triangle. And this is a point, which sequence we already know, or we can easily form. So, if we find the value of h, we will know the percentage of similarity of M to the medial sequence and we can find suitable sequence for point M.

At Fig. 3 we made de calculations (1), (2) and (3) and the similarity between medial sequence and sequence represented by point M should be around $\sim$13% (Fig. 3).

$$h^2 = \mathrm{r}^2 - \left(\frac{r}{2}\right)^2 \tag{1}$$

$$h^2 = \frac{3r^2}{4} \tag{2}$$

$$h = \frac{\sqrt{3}}{2}r \tag{3}$$

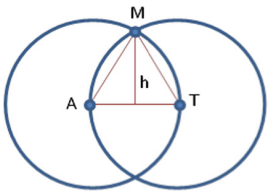

Fig. 3. Calculation of h – distance with Pythagoras theorem

The formed triangle by the centers of the circles and the intersection of triangle is equilateral. The distance from the edge of the triangle to the angle is 0.866025403 78443864676372317075294. But the further away from the center, so the coincidence decreases, which means that the middle point is $\sim$13% match.

It would be quite difficult to find the right "ingredients" of this sequence to satisfy exact match of the middle point of 13% and it would be fairly long sequence. Each match from M to the middle point automatically means matching A-benchmark or T-benchmark. Forming of sequence satisfying exact match of 13%, will require diving deep in the deeps of mathematics and will complicate further processing of the algorithm. Therefore, it's better to seek for a simpler solution of the problem and if possible to limit the sequence with a string of 4 bases which would be repeated endlessly.

Equilateral triangle formed by the centers A, T and the intersection of the circles will be used again. But this time an intercept with the length of 0.5 will be taken from the medial point. That's where would be point C and the Pythagorean Theorem will be applied in order to find the length of the hypotenuse, which should give the percentage of match with A-benchmark. Or (Fig. 4):

$$x^2 = \left(\frac{r}{2}\right)^2 + \left(\frac{r}{2}\right)^2 \tag{4}$$

$$x^2 = \frac{2r^2}{4} \tag{5}$$

$$x = \frac{1}{\sqrt{2}} r \tag{6}$$

$$x \approx 0.71 \tag{7}$$

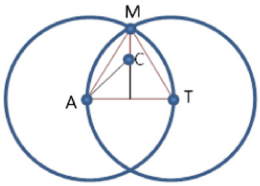

Fig. 4. Calculation of C-Distance with Pythagoras theorem

The further away from the center, the more similarity is reduced. This means that the point C is below 30% match similar. At 4 numbers of bases 25% match means that a base from the ideal A-benchmark must match a base of C-benchmark, and two bases of C-benchmark should match with two bases of medial benchmark (the distance from the tip to C is 0.5 radiuses, a 50% match). If the same reasoning is applied to T-benchmark, this means that one base of T-benchmark also must match with one of C-benchmark. After the present, it is much easier to find the components of the C-benchmark.

ACGTACGTACGTACGTACGTACGTACGTACGTACGTACGT... – A-Benchmark

TGCATGCATGCATGCATGCATGCATGCATGCATGCATGCA… – T-Benchmark

CGATCGATCGATCGATCGATCGATCGATCGATCGAT… – C-Benchmark

GCTAGCTAGCTAGCTAGCTAGCTA…– G-Benchmark fullfit Benchmark C

By drawing all the circles for each of the found benchmarks (A, C, G and T), the area in which the four circles intersect contains all the possible profiles comprised of the four bases (Fig. 5). For the purpose of trilateration are required only 3 circles. Therefore, only benchmarks C, A and T (CAT) will be considered.

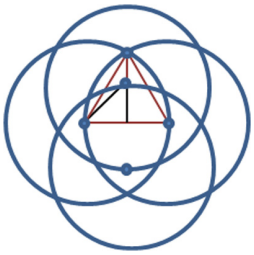

Fig. 5. Intersection between A, T, C and G benchmarks area of sameness. All possible sequences are locked this area

Once the three constant benchmarks have been established to implement trilateration, a problem (1) is solved. Constant sequence favorite does not depend on the data in the database and remain the same when data set changed. In fact, there are three constant sequences as a favorite sequence, which, if followed by standard multiple sequence alignment algorithms with sequence favorite, means there are three times more comparisons. Triples problem (2) - avoid comparisons or reduce the number of comparisons with sequences favorite during search in the database (for each sequence applies a complex algorithm to compare against the sequence favorite).

As founded benchmarks sequences are constant (i.e. not depend on either the data or the number), this allows to make comparisons at the outset - the introduction of the sequences in the database and this is description information (meta data) accompanying each sequence. In this way it is not needed comparison of the sequences during the searching process (which is the slowest operation), but instead will be compared only description information generated at the data input process.

By establishing benchmark sequences is solved problem (3) - Unification/ standardization of sequences favorites for all database. There is now a uniform sequences that are standardized for all the bases, using the described algorithm for comparison.

When two sequences have same profiles, this means that they have sections with same alignment and can be expected complete coincidence of one sequence to the other. But how to evaluate sequences that does not have identical profiles?

For the evaluation of random sequences, it is necessary to calculate the distance of the S1S2 segment in Fig. 6.

$$\sqrt{|AD_1 - AD_2|^2 + |h_1 - h_2|^2} \tag{8}$$

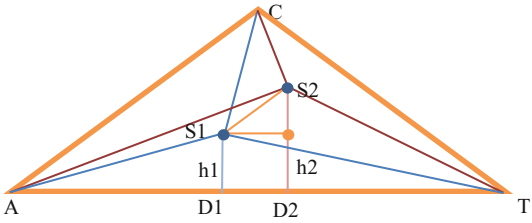

Fig. 6. Calculation of the distance between two profiles with cosine theorem

Currently regarded triangle AS1T, then carry out similar calculations and reasoning for AS2T. What is known about the upper triangle are the sides AT = |1|, AS1 = distance from S1 to A-benchmark (it is known), S1T = distance from S1 to T-benchmark. Used cosine theorem to find angle TAC, after which the side AD1:

$$S_1T^2 = AS_1^2 + AT^2 - 2.AS_1.AT.\cos(\alpha_1) \tag{9}$$

$$\cos(\alpha_1) = \frac{AS_1^2 + AT^2 - S_1T^2}{2.AS_1.AT} \tag{10}$$

$$AD_1 = AS_1.\cos(\alpha_1) \tag{11}$$

$$h_1 = \sqrt{AS_1^2 - (AS_1.\cos(\alpha_1))^2} \tag{12}$$

Similarly for triangle AS1T:

$$S_2T^2 = AS_2^2 + AT^2 - 2.AS_2.AT.\cos(\alpha_2) \tag{13}$$

$$\cos(\alpha_2) = \frac{AS_2^2 + AT^2 - S_2T^2}{2.AS_2.AT} \tag{14}$$

$$AD_2 = AS_2.\cos(\alpha_2) \tag{15}$$

$$h_2 = \sqrt{AS_2^2 - (AS_2.\cos(\alpha_2))^2} \tag{16}$$

After replacing the values found, a value for S1S2 is obtained as distance between two points on the formula:

$$S_1S_2 = \sqrt{(S_1D_1 - S_1D_2)^2 + (AD_1 - AD_2)^2} \tag{17}$$

The lower value for the segment means the greater probability of complete match, expressed in percentages. This allows quicker and accurate enough sequence database search, with a certain percentage of similarity, that later will be aligned and compared with more accurate algorithms such as Needleman-Wunsch or Smith-Waterman.

3 Experimental Results and Analysis

The objective of the experiments is to estimate experimentally efficiency of the designed CAT method for DNA sequence alignment. For this purpose a program implementation is developed. Let us examine these two sequences and their optimal alignment using Needleman-Wunsch algorithm:

```
G A A T T C A G T T A
|   | |   |   |     |
G G A T - C - G - - A
```

After the alignment there are six full matches out of a possible seven or 85.71% matched according Needleman-Wunsch algorithm. The same data have been used for experiments with both methods. Experimental results in the case of proposed CAT method are shown in Fig. 7.

```
GAATTCAGTTA
C-benchmark - 0% off 1
A-benchmark - 18.18% off 0.8181
T-benchmark - 18.18% off 0.8181
cosA = 0.6111111;
AD = 0.5;
h = 0.64762758403522769;
GGATCGA
C-benchmark - 42.85% off .5714285714285714
A-benchmark - 14.28% off .8571428571428571
T-benchmark - 28.57% off .7142857142857143
cosA = 0.71428571428571;
AD = 0.612244897959183;
h = 0.599875039048941;
S1S2 = 0 = 12198041920953887
=> sameness % 87.80%
```

Fig. 7. Experimental results for DNA sequence alignment method based on the trilateration

After finding suitable sequences in the database, such as the minimum value for S1S2, a more accurate alignment algorithm could be applied. Calculations obtained through the proposed CAT method are relatively simple and quick for implement; make it suitable to be applied as a first step in more accurate algorithm such as FASTA and to perform experimental studies utilizing big data sets.

A series of sequence alignment experiments have been carried out through different combinations of DNA sequences comparing both methods. The results for the calculated similarity of the fixed-length sequence alignment are shown in Table 1. DNA1

consists of 100 nucleotides and DNA2 consists of 30 nucleotides respectively. Results of the calculated similarity after alignment of random generated DNA sequences with random length are presented in Table 2. Table 3 presents the results of a benchmark sequence alignment, wherein DNA2 sequence is a subsequence of DNA1 sequence, i.e. there is a 100% match of the second over the first, which is obvious from the column with results of Needleman-Wunsch algorithm. Column Delta in all tables represents deviation between calculations based on CAT algorithm with respect to the Needleman-Wunsch algorithm.

The analysis of experimental results obtained by the sequence alignment shows little deviation of CAT method that could be ignored if this deviation is permissible at the expense of performance. The execution time of Needleman-Wunsch algorithm increases with increasing the sequences length. Time performance of the CAT method remains constant regardless of the sequences length. Therefore, the advantage of the proposed method is the rapid operation of large sequence alignment for which the accurate algorithms execution takes a long time.

Table 1. Experimental results of sequence alignment in case of fixed length sequences

Experiment no	DNA1 length	DNA2 length	CAT similarity	Exact match	Needleman Wunsch	Delta
1	110	30	0.668481925	24	0.8	0.131518075
2	110	30	0.901684169	23	0.766666667	0.135017502
3	110	30	0.709322998	26	0.866666667	0.157343669
4	110	30	0.949633278	24	0.8	0.149633278
5	110	30	0.844141279	25	0.833333333	0.010807946
6	110	30	0.845657014	24	0.8	0.045657014
7	110	30	0.868962098	26	0.866666667	0.002295431
8	110	30	0.861974197	22	0.733333333	0.128640864
9	110	30	0.825153668	26	0.866666667	0.041512999
10	110	30	0.794094919	21	0.7	0.094094919

Table 2. Experimental results of sequence alignment in case of various length sequences

Experiment no	DNA1 length	DNA2 length	CAT similarity	Exact match	Needleman Wunsch	Delta
1	978	149	0.92892273	133	0.89261745	0.03630528
2	945	194	0.958308962	171	0.881443299	0.076865663
3	784	257	0.956673586	210	0.817120623	0.139552964
4	877	280	0.982422073	226	0.807142857	0.175279216
5	907	182	0.886614319	154	0.846153846	0.040460473
6	503	283	0.926791731	197	0.696113074	0.230678657
7	542	136	0.947658604	117	0.860294118	0.087364487
8	723	203	0.951157669	170	0.837438424	0.113719245
9	742	346	0.98153563	263	0.760115607	0.221420023
10	667	379	0.945220218	265	0.701058201	0.244162017

Table 3. Experimental results of sequence alignment in case of exact matching

Experiment no	DNA1 length	DNA2 length	CAT similarity	Exact match	Needleman Wunsch	Delta
1	110	30	0.797937803	30	1	0.202062197
2	110	30	0.878802211	30	1	0.121197789
3	110	30	0.780298459	30	1	0.219701541
4	110	30	0.83460827	30	1	0.16539173
5	110	30	0.790112287	30	1	0.209887713
6	110	30	0.805192591	30	1	0.194807409
7	110	30	0.791356336	30	1	0.208643664
8	110	30	0.902574547	30	1	0.097425453
9	110	30	0.893694124	30	1	0.106305876
10	110	30	0.779823187	30	1	0.220176813

4 Conclusion

An innovative method for DNA sequences alignment based on the trilateration method has been proposed in this paper. Three constant benchmarks for the trilateration implementation have been defined, which create a constant favorite sequence, i.e. it does not depend on the data in the database. This allows making comparisons at the outset – during input of the sequences in the database and it can be stored as meta data to each sequence. Thus, there is no need to make a comparison of the sequences during the search, but instead will only compare the meta data. By establishing benchmark sequences have been solved the problem of unification/standardization of favorite sequence for all facilities using the described algorithm to compare. Calculations obtained in the proposed CAT method are relatively simple and quick to implement, making it suitable for application as a first step in more accurate algorithm. Future work is to apply the proposed algorithm and to perform experimental studies with big data sets in order to investigate the accuracy.

Acknowledgment. This work is supported by Grant DN07/24.

References

1. Mount, D.: Bioinformatics: Sequence and Genome Analysis, 2nd edn. Cold Spring Harbor Laboratory Press, New York (2009)
2. Needleman, S., Wunsch, C.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**, 443–453 (1970)
3. Smith, T., Waterman, M.: Identification of common molecular subsequences. J. Mol. Biol. **147**, 195–197 (1981)
4. Lipman, D.J., Pearson, W.R.: Rapid and sensitive protein similarity searches. Science **227** (4693), 1435–1441 (1985). https://doi.org/10.1126/science.2983426. PMID 2983426
5. Altschul, S., et al.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)

6. Altschul, S., et al.: Gapped BLAST and PSIBLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**, 3389–3402 (1997)
7. Lin, H., et al.: Efficient data access for parallel BLAST. In: 19th International Parallel & Distributed Processing Symposium (2005)
8. Zhang, F., Qiao, X.Z., Liu, Z.Y.: A parallel Smith-Waterman algorithm based on divide and conquer. In: Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing ICA3PP 2002 (2002)
9. Farrar, M.: Striped Smith-Waterman speeds database searches six times over other SIMD implementations. Bioinformatics **23**(2), 156–161 (2007)
10. Borovska, P., Gancheva, V., Landzhev, N.: Massively parallel algorithm for multiple biological sequences alignment. In: Proceeding of 36th IEEE International Conference on Telecommunications and Signal Processing (2013). https://doi.org/10.1109/TSP.2013.6614014
11. Motlagh, O., Tang, S.H., Ismail, H., Ramli, A.R.: A review on positioning techniques and technologies: a novel AI approach. J. Appl. Sci. **9**, 1601–1614 (2019). https://doi.org/10.3923/jas.2009.1601.1614
12. Thapa, K., Case, S.: An indoor positioning service for bluetooth ad hoc networks. In: Proceedings of the Midwest Instruction and Computing Symposium, 11–12 April 2003, Duluth, MN, USA, pp. 1–11 (2003)
13. Ciurana, M., Barcelo-Arroyo, F., Izquierdo, F.: A ranging system with IEEE 802.11 data frames. In: 2007 IEEE Radio and Wireless Symposium (2007). https://doi.org/10.1109/rws.2007.351785