Alignment-free Sequence Searching over Whole Genomes Using 3D Random Plot of Query DNA Sequences

Da-Young Lee, Hae-Sung Tak, Han-Ho Kim and Hwan-Gue Cho
Dept. of Electrical and Computer Engineering, Pusan National University, South Korea
E-mail: {schematique, tok33, quant1216, hgcho}@pusan.ac.kr

Keywords: sequence search, sequence visualization, whole genome, 3D random plot

Received: March 29, 2018

Most genomic data studies are based on sequence comparisons and searches, and comparison models based on alignment algorithms are most commonly used. This method is very accurate, but it is useful when the query is short in kilobytes, because it requires the quadratic time and space complexity, $O(n^2)$ where n is the length of target and query sequences. With the development of Next Generation Sequencing techniques, researches on whole genome sequence data of megabyte size are being actively studied, and new comparison and search methods for large-scale sequence data are needed. We propose a new alignment-free sequence comparison and search method to overcome the limitations of the alignment-based model. In this graphical model, the sequence searching problem in DNA strings can be reduced to find some parts of geometric object within a relatively small-scale geometric space. When comparing similarity by modifying sequences of similar length, we can confirm that the comparison model is appropriate by accurately reflecting the degree of similarity. When searching the query sequence comparison model based on 200MB sized whole genome sequence, using the compressed coordinate information, it was able to search the 10MB sequences in 22s, which is a very reduced time compared to alignment. Although it is not possible to find the exact position of the base pair unit as in the alignment result, it is a model that can be used as a preprocessing process to quickly search a whole genome sequence of several hundred megabytes-size.

Povzetek: Na podlagi 3D vizualizacije celotnega zaporedja genoma so avtorji pokazali, da je na dolžini poizvedbe možno prilagodljivo hitro iskanje.

1 Introduction

Genomic data studies are done through sequence comparisons, mostly using a model based on an alignment algorithm. For example, Basic Local Alignment Search Tool (BLAST)[1] is the most common method to search for sequences in a database. It divides the query sequence into three characters, finds the matching region, and gradually widens the region to select candidates for alignment. Although it is very useful when searching for a short query in the whole database, since it is based on alignment, it is difficult to obtain an immediate processing result in the case of a large sequence such as a megabyte-scale chromosome owing to a large increase in computational cost. When utilizing the actual BLAST service, it is recommended to reduce the database search scope when the query size is of the order of megabytes, and it is often time consuming to search and provide results by mail, rather than providing it immediately.

In addition, since gene recombination is different from sequence alignment based on conservation of contiguity between homologous segments, in order to overcome this problem, alignment-free comparison method such like word-frequency statistics, a method of calculating distance in space defined by frequency vectors, is also actively underway[2]. Such research is also widely used as a pre-filter for processing queries of alignment-based models.

In this paper, we propose a geometric-based heuristic technique that enables the rapid comparison and search of sequences in personal computers. In this regard, AMSS[3] is a model that provides shape-based similarity comparison, assuming that the time series data is a vector sequence. Instead of focusing on individual points of time series data, the model focuses on vectors and compares similarities between data using cosine similarity. This method is advantageous in that it is effective for amplitude and time shifting. In this study, we also aimed to reduce the time and space complexity by converting the genetic sequence into a geometric object such as a random plot and performing comparison and search, taking into account that the genetic sequence data is ordered sequence data. Instead of considering a single separate base, as in the alignment algorithm, the method compares the vector generated based on the sequence of the predetermined unit only once, and it is possible to significantly reduce the time required for comparison operation by visualizing a sequence search result and presenting the information more intuitively. In addition, the high-speed heuristic search technique can be applied to large amounts of data, and it is possible to specify the necessary precise alignment analysis.

Compared to [14], we present an improved similarity computation algorithm that considers input sequences with different lengths. We show the effectiveness of the proposed method with experiments on searching for short query sequences on a long sequence.

2 Related work

2.1 Genome Sequence Visualization

Most genetic data have a huge volume, and it is difficult to find meaningful patterns in such data owing to the irregular configuration of the four bases. The visualization of sequence information and sequence analysis information can help in forming an intuitive understanding of the genomic data and enable the efficient representation of the results. Genome visualization research focuses on two aspects. The first is the visualize of a large amount of genetic information in a short time and a limited space, and the second is the representation of complex information as intuitively as possible.

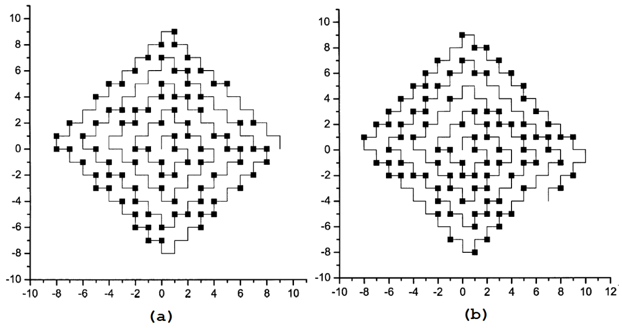

Figure 1: The compact graphical representation [4] of the first exon of human β-globin gene(a) and gorilla β-globin gene. The visualization of search result for query sequence of 10M size in human chromosome 1.

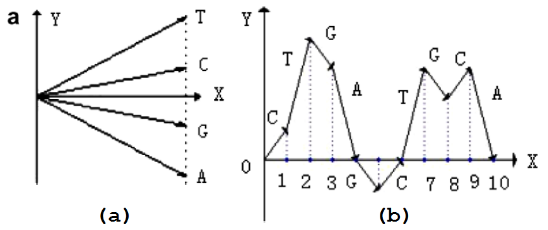

Figure 2: The vector design of 'H-L curve'[5] (a) and graphical representation for the DNA sequence $s =$ 'ATGGCATGCA' (b).

The 'Worm Curve'[4, 6] represents genome information in a limited space, and it assigns a binary code to each base. It is plotted on a Cartesian coordinate system, and its most significant biggest advantage is that the curve can represent all the information in a relatively small space, despite how little the point intersects with each other. Studies have been actively conducted using a variety of curves to intuitively represent complex information. For example, the 'Dual-Base Curve' (DB-Curve)[7] has been designed to visualize the features of a genome sequence at a glance. In this curve, the two different bases are configured as a combination, and a two-dimensional vector is assigned, where the y component is assigned as a constant (+1) and the x components are assigned separately. In this visualized, since the curve is continuous in the positive direction of the y axis, there is no point at which it crosses with itself. Obtaining a ratio of the x-coordinates of the end points can confirm the relative existing ratio of the two bases to obtain the statistical information of the sequence in an intuitive manner.

In contrast, the 'H-L curve'[5] is a method of assigning a two-dimensional vector for the four bases with a constant x component, and this curve avoids intersection with itself because different y-components are assigned. Since the progress of a DNA sequence matches one-to-one with the 'H-L Curve,' it has the advantage that the main difference of each sequence with other sequences can be checked quickly.

In addition to visualizing curves, there is a 'Four-Color Map'[8], which assigns colors to each base and fills areas proportional to the frequency of occurrence with the corresponding color, and 'Circos'[9, 10], which visualizes the whole genome in a circular track form. 'Circos' represents a chromosome as a piece of a circular track, and connects the interactive chromosome tracks with a curve, thereby effectively expressing the internal relation of the whole genome. Although most relational connection visualization methods express only one-to-one associations, 'Circos' can express many-to-many associations as well by using circular tracks.

2.2 Visualization Tool for Genome Sequence

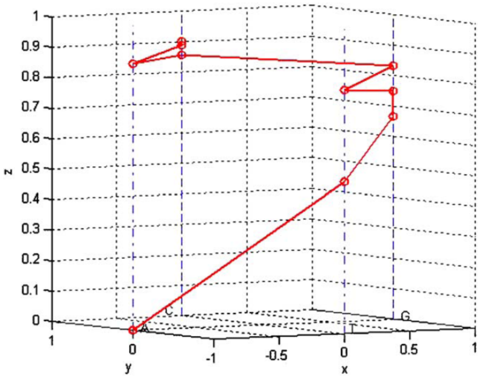

Figure 3: 3D graphical representation of DNA sequence using Z-axis as time axis[11]. The graphical representation for the sequence 'ATGGTGCACC'.

To compensate for the drawbacks of the sequence alignment method in terms of processing speed, a heuristic method based on visualization is utilized. By converting a large amount of text information composed of only four kinds of bases, the meaning of which is difficult to intuitively grasp, to geometry information, heuristic methods are able to identify the type of data through visual examination to easily find patterns that cannot be revealed using computational methods[12]. Furthermore, geometric rules found in the visible results often have a meaningful relationship with genomic analysis in the field. Heuristic methods are especially useful when utilized for quickly calculating similarity or dissimilarity.

For example, large-scale genomic sequence information is converted into information on a polygon domain, and the problem of finding similarity is solved by replacing the comparison of similarity of sequences with the comparison of image similarity[13]. By setting a direction for each base, the sequence is converted to a random plot in which the polygon area is simplified with the k-convex hull, and the homology of two random plots is compared. Studies [14, 15] have considered the extended space up to three dimensions in the vector assignment for each base. Consequently, a random plot can be visualized on three dimensions, and the similarity can be compared by simplifying it to be close to the actual random plot.

Since direct comparison is difficult for a walk-plot object in three dimensions, a random plot is populated in a certain space around the polygon area, and the orthogonal projection of this space on each plane (X-Y, Y-Z, and X-Z) is used to compare the degree of similarity using the overlap area ratio. However, the comparison method based on the overlapping area has a drawback in that it does not take into account the random plot present in the local area. To overcome this drawback without simplifying the random plot, the shape of the line is maintained while the shortest distance between any points of two random plots is calculated for comparing the degree of similarity between two sequences[16].

Previously, an alignment method called ‘Four Line’ involving graphical-domain sequence alignment, rather than string alignment, was proposed[17]. By assigning the four bases to different points on the Y-axis and connecting the matched points in the sequence to be subjected to alignment in the X-axis to make a visualization of the zigzag curve, the visualization result of the two sequences are compared to conduct alignment.

In order to overcome the disadvantages such as loss of information and self-intersection of existing two-dimensional visualization methods, there is a study in which a DNA sequence is three-dimensionally utilized as a time axis[11]. Regardless of the information of the base to the z-axis will always increases, and by assigning vectors x, y axis is increased or decreased for each base. Not only it limited to visualization, to derive the geometrical center of the curve, this time the center of this curve is important information indicating the distribution of each base. In this study, a similarity comparison model was devised by assigning vectors to each other in different ways and using the Euclidean distance and angle correlation of the distance to the start and end points of the vector through eight transform. As a result, they could construct the similarity matrix, it shown that the similar species such as human and gorilla have high similarity.

In this manner, visualization results can be used not only for the intuitive delivery of sequence information but also as an analysis target to improve the processing speed and to obtain meaningful results. In this study, by focusing on this point, we convert a whole genome sequence to a walk-plot object in three-dimensional space, extract a vector, and compare and search for the sequence with improved speed. Furthermore, by visualizing a search query sequence together with the random plot of the whole genome sequence, the position and distribution of the obtained similar sequence can be transferred in an intuitive form.

Table 1: Functional Performance of Previous Research

Research	Plotting space dimension	Supports large-scale sequence	Global similarity compute	Local similarity compute
BLAST [1]	N/A	△	O	O
Compact 2D [4]	2D	O	O	X
H-L Curve [5]	2D	△	X	X
Bo Liao [11]	3D	△	O	X
3D Random [15]	3D	O	O	X
Proposed	3D	O	O	O

3 New method using 3D random plot

3.1 Sequence Searching method with 3D Random Plot Structure

An overview of our algorithm framework is shown in Figure 4. Generally, all types of biological sequence comparison exploit the sequence alignment based on a dynamic programming approach. One popular alignment algorithm is the Needlemann–Wunsch algorithm, which is widely used in molecular biology. There are many variations in sequence alignment, such as global alignment, local alignment, and semi-global alignment. Though the alignment approach has many advantages, it has a critical drawback in that it involves high complexity in terms of execution-time complexity and space complexity. The complexity of the basic alignment algorithm is $O(m \cdot n)$ if the lengths of two input sequences are n and m. If $\Theta(n) = \Theta(m)$, the complexity is quadratic: $O(n^2)$. When the size of the input sequence is greater than 100 megabytes, this alignment is impractical, because it requires a main memory greater than the order of gigabytes. To overcome these problems, researchers developed heuristic alignment techniques such as BLAST-like tools. Another problem in the alignment algorithm is that it is not easy to define the score/penalty matrix to meet the many different constraints in biological sequence comparison.

The basic idea of our approach is that we compute the similarity of two sequences in ‘geometric random plot’

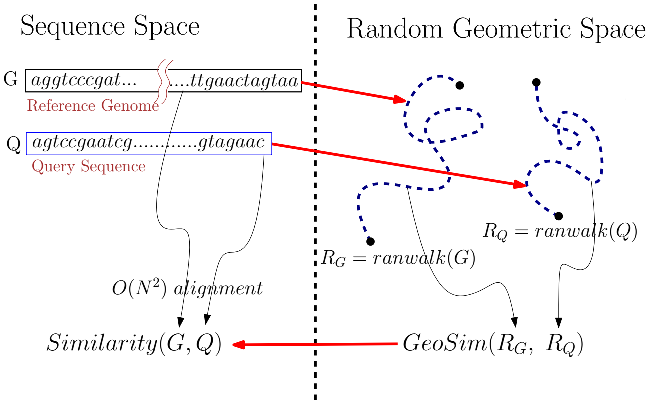

Figure 4: Space transform from sequence to 3D geometric shape.

space, rather than 'string sequence' space. As shown in Figure 4, we first transform the input sequences into random plot in 3D space. Then, we compare or search for a target sequence in 3D geometric object.

This transformed random plot can be visualized on an appropriately sized grid, and a sequence of megabytes in size can be represented by a list of pixels much smaller than the actual number of bp.

Thus, we can say that our geometric transformation is a type of approximation with visualization. The advantage of our transformation is that the global structure can be shown by hiding the biological noise embedded in the sequence. The main merit of our approach is that it is useful and efficient in comparing very long sequences. Assume that we are asked to find the location of a sequence that is a few megabytes in length in a whole genome longer than 100 megabytes.

3.2 Vector Allocation for random Plot

Sequence data are string information composed of {a,g,t,c}; therefore, they must be converted into graphical information for visualization. Previous 2-D visualization methods have visualized genome sequences by assigning a separate base in the positive and negative directions of each axis (x and y). This method has a disadvantage in that a large amount of information is lost when a base having a vector in opposite directions is continuously repeated. Furthermore, if the same pattern is continuously repeated, it is impossible to visualize a large volume of data in a limited space. To overcome this disadvantage, [15] used a 3D vector. A vector is assigned to each base, but a combination of two bases constitutes a random plot. When the two bases are coupled together with the vector in the opposite direction, the representation is made three-dimensional with a z-axis to minimize the lost information. In this study, by using a 3D vector allocation model[15], we calculate the vector character of the sequence data and obtain sequence search positions to visualize the results.

Table 2: Vector allocation method for each 2-mer base in a genome sequence in three-dimensional geometric space

2-mer	Vector	2-mer	Vector
AA	(2, 0, 0)	AG(GA)	(1, 1, 0)
AC(CA)	(1, -1, 0)	AT(TA)	(0, 0, -2)
CC	(0, -2, 0)	CG(GC)	(0, 0, +2)
CT(TC)	(-1, -1, 0)	GG	(0, 2, 0)
GT(TG)	(-1, 1, 0)	TT	(-2, 0, 0)

Table 2 summarizes the vector allocation method for each 2-mer. In Table 2, the base pairs AT and GC are represented on the z axis. The other base pairs are represented as the sum of two unit vectors for each base, as given by the WS-curve method.

After the vector transition for DNA genome data information, those vectors are visualized in three-dimensional space. The method of visualization is the same as that of two-dimensional visualization, where the sum of vector values is computed according to the order of sequences and the results are connected with a line to provide the final visualization result. For the random plot R, the starting point is $R(0) = (X_0, Y_0, Z_0)$ $(X_0 = Y_0 = Z_0 = 0)$. $Unit^{3d}(i)$ is the converted value of the ith 2-mer of the unit vector. The ith point $R(i) = (X_i, Y_i, Z_i)$ of the random plot is computed as follows:

$$R(i) = R(i-1) + Unit^{3d}(i) = \sum_{k=1}^{i} Unit^{3d}(k) \qquad (1)$$

Figure 5 shows the direction of the random plot for each 2-mer read. Since the first 2-mer read 'AA' is on the x-axis (+2), it can be confirmed from figure (a) that the positive x-axis moves from the origin O. Since the next 2-mer read is 'AT', a movement in the z-axis by (-2) can be confirmed.

This vector transformation rule are determined empirically in order to discriminate different sequences effectively. As Figure 6, similar sequences are likely to produce similar walk plots.

In this way, the transformed random plot is visualized in an appropriate sized three-dimensional grid. The default grid size 500 × 500 × 500 is what we empirically figured out at which this trade off between speed and correctness of comparison is well balanced for the sequences used in the experiments.

In case of the short genome sequence, it can be represented in a $500 \times 500 \times 500$ grid easily. But the large size sequence needs space normalization to visualize the random plot in limited space. When the vectors of the random plot are calculated, the points that are farthest from the origin $O(0, 0, 0)$ to the X, Y, and Z axes are max_x, max_y, max_z, and the view size of visualization is V, the normalized ith point $R(i) = (X_i, Y_i, Z_i)$ can be expressed as:

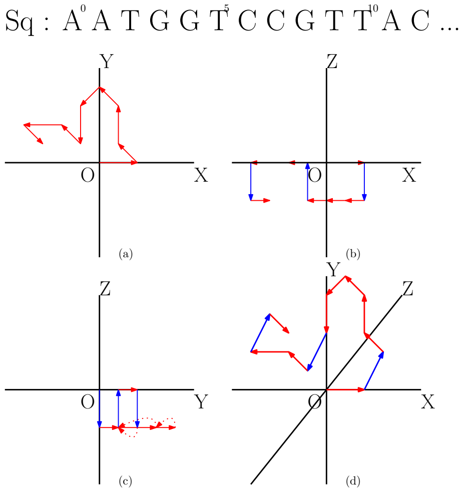

Figure 5: Movement of the random plot for each 2-mer read. (a), (b), (c) and (d) show plots in the form of walks in the X-Y, X-Z, and Y-Z planes in three-dimensional space. From $O(0,0,0)$, the random plot proceeds in accordance with the base assigned to 2-mer. The red random plot represents movement on the X-Y plane, and the blue random plot represents movement on the Z axis.

$$Regular(R(i)) = (X_i \cdot \frac{V}{max_x}, Y_i \cdot \frac{V}{max_y}, Z_i \cdot \frac{V}{max_z}) \qquad (2)$$

This visualization model is so useful to compare the huge whole genome. Figure 6 shows advantage of this works[15]. We have constructed the 3D random plots from two whole genomes such as Human Chromosome 1 and Chimpanzee Chromosome 1. In Figure 6, red random plot represents the Human and green one represents the Chimpanzee. Red random plots are up in the positive direction of the X and Y-axis than the green one. This visualization method directly make us to confirm that two genomes are quite similar and the Human chromosome has more 'G' and 'A' base compared to Chimpanzee.

3.3 Vector Extraction from Random Plot

For G, a genome sequence consisting of 4 DNA bases { a, g, t, c }, $ranwalk(G)$ represents a three-dimensional geometric object constructed by our proposed algorithm. Therefore, $ranwalk(G_i)$ consists of a list of linked pixels as follows:

Definition 1.

$$ranwalk(G) = < P_1, P_2, \ldots, P_l >$$

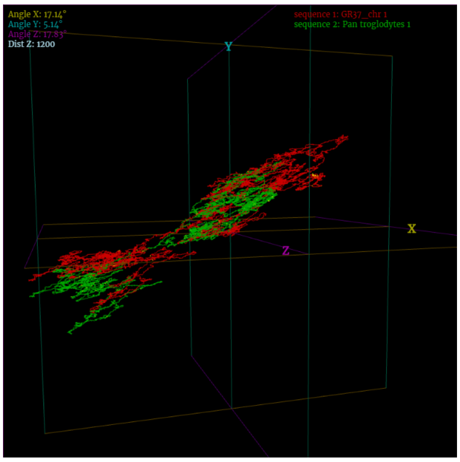

Figure 6: Visualization result of Human and Chimpanzee chromosome 1. Red plot is constructed from Human chromosome 1 and the green random plot is constructed from the whole genome of Chimpanzee (Pan troglodytes) chromosome 1.

The position of a $ranwalk$ pixel is denoted $P_i = (x_i, y_i, z_i)$ satisfying $|x_i - x_{i+1}| \leq 1$, $|y_i - y_{i+1}| \leq 1$ and $|z_i - z_{i+1}| \leq 1$, which means two pixels P_i and P_{i+1} are adjacent to each other, sharing a common face. We say P_i and P_{i+1} are 'adjacent' if they are within a distance of 1.

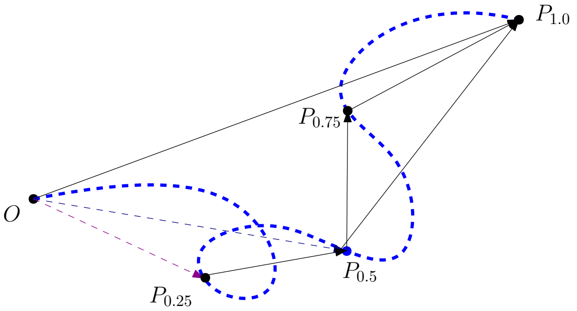

Figure 7: A geometric random plot (blue dotted line) and corresponding vectors.

Now, we explain how to compute the distance between two $ranplot$ pixels obtained from two genomes G_a and G_b to be compared. Assume that we constructed two geometric objects, $R_a = ranplot(G_a)$ and $R_b = ranplot(G_b)$. The proposed distance measure, *random plot distance* ($Rdist$), is a vector with two components $\Delta Span$ and $\Delta Degree$. The proposed $Rdist()$ measure has another parameter, depth k. The distance between two random plot R_a and R_b at depth k is defined recursively as follows. In this definition, R_a1 is the first half of R_a, and R_a2 is the last half of R_a. R_{b1} and R_{b2} are defined in a similar manner. Thus, $R_a = R_{a1} \odot R_{a_2}$, where $\odot$ denotes the geometric concatenation operation.

Definition 2.

$$Rdist(R_a, R_b, k) = Rdist(R_{a1}, R_{b1}, k+1) + Rdist(R_{a2}, R_{b2}, k+1)$$

Now, we explain how to compute $Rdist(R_a, R_b, k = 1)$ at the basic depth = 1 level. In Figure 7, the thick blue

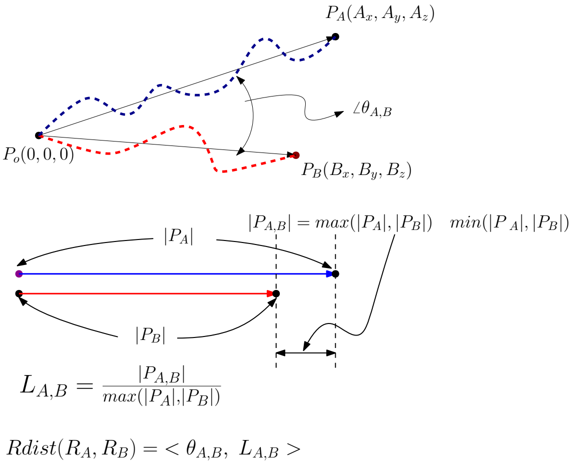
$P_A(A_x, A_y, A_z)$
$\angle\theta_{A,B}$
$P_o(0,0,0)$
$P_B(B_x, B_y, B_z)$
$|P_{A,B}| = max(|P_A|, |P_B|)$
$min(|P\ A|, |P_B|)$
$|P_A|$
$|P_B|$
$L_{A,B} = \frac{|P_{A,B}|}{max(|P_A|,|P_B|)}$
$Rdist(R_A, R_B) = <\theta_{A,B},\ L_{A,B}>$

Figure 8: Two comparison parameters $\{\theta_{AB}, L_{AB}\ \}$.

dotted curve represents the random plot for a genome sequence. Symbols $P_0(O)$ and P_1 denote the first and last pixel of a random plot, respectively. P_t denotes the first t-percentile pixel. Thus, $P_{0.5}$ denotes the exact middle pixel in the list of pixels generated by our transformation algorithm.

For an interval in a random walk, we obtain a parameter, the length of the direction vector (P_0, P_1). If two random walks to be compared start with the origin $(0,0,0)$, then we can obtain the lengths of two direction vectors from R_a and R_b and compute the angle difference between two vectors Pa_1 and Pb_1.

Assume the start and end points of R_a are P_{a0}, P_{a1}, and those of R_b are P_{b0}, P_{b1}. If $k = 1$ is, the comparison target is $\overrightarrow{P_{a0}P_{a1}}$ and $\overrightarrow{P_{b0}P_{b1}}$. If $k = 2$, further down one step,divided into two vectors are compared both front and rear vector. Therefore, the comparison target are $\overrightarrow{P_{a0}P_{a0.5}}$ and $\overrightarrow{P_{b0}P_{b0.5}}$, $\overrightarrow{P_{a0.5}P_{a1.0}}$ and $\overrightarrow{P_{b0.5}P_{b1.0}}$. If $k = 3$, by applying the same method, it performs a comparison of eight times (2^k).

If the length of divided vector drops below the appropriate length D, the recursion is aborted. In this paper, the threshold D value is set to 100 times the unit size, where unit size is the number of bp per pixel when visualized. The D value was determined experimentally because at least the length of the vector was more than 100px, meaningful comparison was possible.

3.4 Computing Similarity and Search on Random Plot

Rdist refers to the similarity distance between the two vectors. Figure 8 shows that two parameters of $\theta_{A,B}, L_{A,B}$ for *Rdist*. $\theta_{A,B}$ refers to the angle between the two vectors, and $L_{A,B}$ refers to the ratio between the length of two vectors differ and from those of the longer vector. If the two vectors have the same orientation, $\theta_{A,B} = 0$, two vectors, if the length is equal to $L_{A,B} =$ is 0$(0 \leq \theta_{A,B} \leq 180, 0 \leq L_{A,B} \leq 1)$.

Algorithm 2 Comparison Algorithm

```
initialize $beg \leftarrow 0$
initialize $end \leftarrow$ len$(R_a)$
initialize $O \leftarrow \{0,0,0\}$
initialize $D \leftarrow threshold\,lenth\,of\,vector$
procedure Sim($beg$, $end$ : index of vector list, $R_a$, $R_b$ : random plot of $G_a, G_b$, threshold $\theta_s$, $L_s$)
    $mid \leftarrow (end - beg)/2 + beg$
    $cnt \leftarrow 0$
    if $end - beg > D$ then
        $cnt+ = Sim(beg, mid, R_a, R_b)$
        $cnt+ = Sim(mid + 1, end, R_a, R_b)$
    else
        $V_a \leftarrow R_a[end] - R_a[beg]$
        $V_b \leftarrow R_b[end] - R_b[beg]$
        $Len_a \leftarrow \mathbf{euclideanDist}(O, V_a)$
        $Len_b \leftarrow \mathbf{euclideanDist}(O, V_b)$
        $\theta_{a,b} \leftarrow \mathbf{acos}(\frac{\mathbf{dotProduct}(V_a,V_b)}{Len_a \times Len_b}) \times 180$
        ${}_{a,b} \leftarrow \frac{\mathbf{abs}(Len_a - Len_b)}{\mathbf{max}(Len_a, Len_b)}$
        if $\theta_{a,b} \leq \theta_s$ and ${}_{a,b} \leq L_s$ then
            return 1
        else
            return 0
        end if
    end if
    return $cnt$
end procedure
```

To compare and visualize the random plot in a limited space, compression is necessary, as described earlier formula 2. However, in the case of the reference sequence, to calculate the overall similarity of the two vectors, it maintains the two normalized values set. One is a normalized value that is used to process the query sequence, and the other is a normalized value of the calculated original reference sequence. When comparing the sequence to search when the use of normalized values of the query, and visualization uses the original normalized value. This is because it can not be an accurate comparison due to the size difference between the reference and the query, the normalized values differ.

After the normalization of the reference sequence and query sequence the normalized according to the normalization value of the query sequence, extend the depth to a predetermined level k to proceed comparison by dividing a random plot as unit size. Compare all the pieces of the vector unit size extracted from the two random plot by $Rdist()$. When processing the results meet the predetermined reference range, the higher the degree of similarity ($\theta_{A,B} \leq \theta_s$ and $L_{A,B} \leq L_s$). The ratio between the number of the unit vectors that meet the conditions and the total number of vector is similarity between two sequences.

3.5 Reference Sequence Slot

If the length of the query is long enough, the sequence information is compressed at an appropriate rate during visualization in a limited space. Therefore, it is possible to perform in the on-memory state by applying the same compression ratio when searching in the reference sequence. However, sequences with short lengths, such as the LTR sequence, are only kilo-bytes in size and remain uncompressed in the visualization process. In this case, vector information becomes large, and query search becomes impossible in on-memory state. In order to compensate for this, when the length of the reference sequence differs by more than 200 times, the reference sequence is divided into an appropriate number of slots to perform the search. A slot is like a window. By reducing the search range by multiple of the query length at a certain point in time, the method described above can be applied even in a case where a search is required at a low compression ratio in a large size sequence.

$$|Slot(Q,R)| = \frac{|ranwalk(R)| - c_0 \cdot |ranwalk(Q)|}{|ranwalk(Q)| \cdot (c_0 - 1)} \quad (3)$$

Equation 3 is the number of slots created when a query and reference sequence are given. Q and R are Query and Reference sequence respectively, and $len(ranwalk(X))$ represents the length of the whole vector information when X sequence is expressed as a random plot. c_0 is a control constant, which is the size of the space in which a vector should be searched when a certain size query vector is given. In this paper, c_0 is set to around 200.0. Since the query may exist at the point where the slot is divided, the boundaries of each slot are overlapped by the length of the query vector. Figure 9 shows that the vector of the reference sequence is divided into slots.

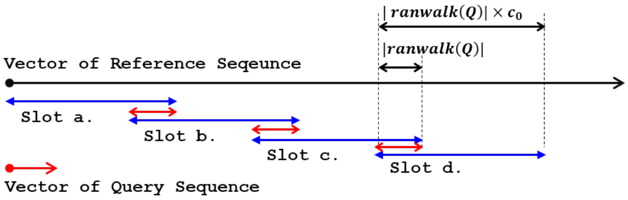

Figure 9: Slot division in reference sequence vector based on the vector length of the query sequence.

4 Experiments

4.1 Dataset Preparation

Actual biological sequence data were used for the searching experiment, and artificial data were used to validate the similarity comparison model. The biological sequences are Human chromosome 1 (246MB size) and the sequence of a 1M-10M size extracted from chromosome 1. Artificial sequence data are obtained by extracting a sequence of 1-10 MB length from the Human chromosome 1 sequence at a random location and inserting noise in a predetermined ratio. A number of bases with different sizes are deleted, inserted, and replaced by a ratio of 1% to 50%. The artificial data information such as ratio and the b.p. size and number of pixels and compression ratio is shown in Tables 3 and 4. 'A1-0' means that the artificial data of 1M size and 0% modified, namely it is just extracted from Human sequence, not modified. But 'A10-25' means that the artificial data of 10M size and 25% modified.

This modification rate is expressed as 'M' (M.Rate) in Table 3 and 4. 'M' (M.Rate) refers to the modified ratio of the number of B.P. on origin sequence. For verification of the similarity comparison model, this rate was set higher gradually as the experiment was repeated.

'Ratio' refers to the compression ratio of the number of B.P. and pixels of the actual sequence to be converted to a random plot. For example, in the Table 3, since A1-1 sequence has 1000.02K bases, and random plot size consists of 36K pixel, the compression ratio is 3.58%. 'Sim' means that the similarity result of origin sequence and modified sequence and 'Comp.t' represents the comparison time.

Table 3: Specification of artificial data of 1M, 2M size extracted from Human chromosome 1 and comparison result

Sq N.	M (%)	Length (K bp)	Plot (K px)	Ratio (%)	Sim. (%)	Cmp.t (s)
A1-0	0	1000.02	36.00	3.58	100.00	0
A1-1	1	999.93	35.79	3.58	99.59	0
A1-2	2	1000.01	36.17	3.62	99.45	0
A1-5	5	999.89	36.67	3.67	98.23	0
A1-8	8	999.97	37.74	3.77	96.06	0
A1-10	10	1000.49	38.05	3.80	91.73	0
A1-15	15	999.78	40.74	4.07	93.58	0.016
A1-20	20	1000.29	42.49	4.25	91.76	0
A1-25	25	999.92	44.2	4.42	86.14	0
A1-30	30	999.79	47.18	4.72	84.23	0.015
A1-40	40	1001.12	50.86	5.08	69.86	0.015
A1-50	50	999.47	58.36	5.84	63.53	0.016
A2-0	0	2000.04	67.09	3.35	100.00	0
A2-1	1	1999.96	66.89	3.34	98.03	0
A2-2	2	2000.15	67.27	3.36	95.85	0
A2-5	5	2000.26	68.99	3.45	94.65	0
A2-8	8	2000.2	70.4	3.52	90.5	0
A2-10	10	2000.14	69.64	3.48	91.2	0.016
A2-15	15	1999.94	70.84	3.54	85.71	0
A2-20	20	2000.18	77.56	3.88	83.62	0
A2-25	25	2000.66	79.97	4.00	72	0
A2-30	30	1999.85	89.15	4.46	73.37	0
A2-40	40	2001.5	88.54	4.42	63.34	0.016
A2-50	50	2000.62	104.11	5.20	54.91	0.016

Tables 5 and 6 are data for searching for LTR sequences that are frequently handled in real bioinformatics analysis. In the table 5, R-F-1 is the reference sequence and means chromosome 1 sequence of the Flatfish. In the corresponding table 6, Q-F-1 is the query sequence of R-F-1 and is the LTR sequence extracted from R-F-1. The biggest difference from the artificially generated data is that the LTR sequence is too short and thus has a low compression rate

Table 4: Specification of artificial data of 4M, 10M size

Sq N.	M (%)	Length (K bp)	Plot (K px)	Ratio (%)	Sim. (%)	Cmp.t (s)
A4-0	0	4000.09	42.62	1.07	100.00	0
A4-1	1	4000.18	42.69	1.07	99.3	0
A4-2	2	3999.71	42.15	1.05	98.93	0
A4-5	5	3999.51	44.13	1.10	98.18	0
A4-8	8	3999.36	44.08	1.10	96.03	0
A4-10	10	4000.1	45.95	1.15	96.27	0
A4-15	15	3999.75	45.69	1.14	94.63	0
A4-20	20	4000.23	49.33	1.23	91.33	0
A4-25	25	3999.7	49.78	1.24	90.93	0
A4-30	30	4001.21	53.79	1.34	84.36	0.016
A4-40	40	3999.59	57.16	1.43	76.82	0.015
A4-50	50	4000.14	64.1	1.60	66.87	0
A10-0	0	10000.05	65.26	0.65	100.00	0
A10-1	1	10000.03	65	0.65	98.08	0
A10-2	2	10000.13	64.81	0.65	97.29	0
A10-5	5	9999.47	66.32	0.66	96.76	0.015
A10-8	8	9999.74	68.75	0.69	95.12	0
A10-10	10	10000.71	67.93	0.68	94.9	0.015
A10-15	15	9999.97	75.13	0.75	91.18	0
A10-20	20	9998.82	74.38	0.74	90.24	0
A10-25	25	9999.4	78.34	0.78	87.68	0.016
A10-30	30	9999.24	82.29	0.82	82.49	0
A10-40	40	9999.82	87.51	0.88	78.48	0
A10-50	50	10001.48	94.45	0.94	66.47	0

in the visualized space. This is because visualization is possible in a limited space without compression. Since the reference sequences are based on the compression ratio of the query sequence, we can see that the random plot size of the reference sequence is very large relatively.

Table 5: Specification of biological data for reference

Sq N.	Chr.	Species	Length (M bp)	Plot (M px)	Ratio (%)
R-F-1	1	Flatfish	19.80	19.02	95.06
R-F-2	2	Flatfish	20.14	19.34	96.02
R-F-3	3	Flatfish	22.24	21.36	96.04
R-F-5	5	Flatfish	23.64	22.69	95.98
R-H-1	1	Human	246.89	236.44	95.77

Table 6: Specification of biological data for query

Sq N.	Chr.	Species	Length (K bp)	Plot (K px)	Ratio (%)
Q-F-1	1	LTR	0.41	0.41	100.00
Q-F-2	2	5’LTR	1.56	1.54	98.72
Q-F-3	3	Gypsy	4.84	4.78	98.76
Q-F-5	5	LTR	8.55	6.44	75.32
Q-H-1	1	HERV-K	9.26	8.06	87.04

4.2 Experiment:Comparison Between Modification ratio and Similarity based proposed Model

Table 3 and Figure 12 show the result of similarity analysis of origin extracted sequence and modified sequences. In Table 3, ‘Sim’ means that the similarity result of origin sequence and modified sequence and ‘Comp.t’ represents the comparison time. As the modification ratio increases, the degree of similarity decreases. Thus, it can be confirmed that the similarity comparison model proposed in this study accurately reflects the similarity of the sequences. In addition, except for sequence generation, the time required for comparison is 0.02 seconds, which means that it can be processed at a very high speed.

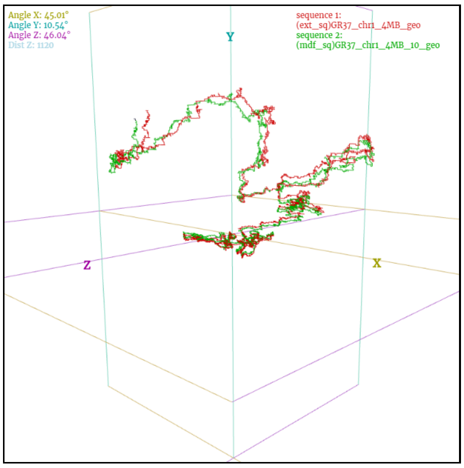

Figure 10: Red random plot represents one part of Human chromosome 1, the length of which is 4 MB, in terms of nucleotide bases. Green random plot represents the 10% distorted sequence of the red one, Human chromosome 1.

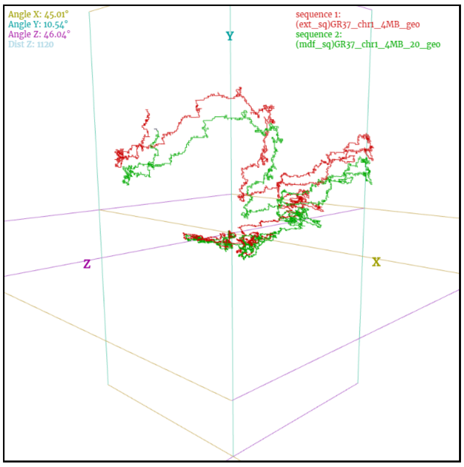

Figure 11: Red random plot represents one part of Human chromosome 1, the length of which is 4 MB, in terms of nucleotide bases. Green random plot represents the 30% distorted sequence of the red one, Human chromosome 1.

4.3 Experiment:Artificial Sequence Search over whole genome sequence

Table 7 is the result of sequence searching process for extracted original sequence from Human chromosome 1 and the modified sequences. ‘Unit B. P. ’ is the size of B.P. as

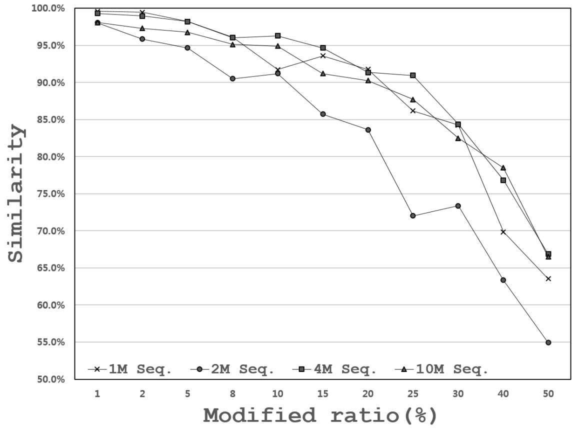

Figure 12: Similarity between origin sequence and modified sequences in each size 1-10MB.

a unit of search,‘ Unit Vector’ refers to the size of the vector to consider when comparing a time. ‘Error Dist.’ is the distance between the actual sequence position and the result of search position. ‘Find.t’ shows the amount of time spent on search. The original sequence (0% modified sequence) search, as well as about the modified sequence of up to 20% are also searched in a short time. The difference between the actual position and the search result is relatively accurate, as the query size is less than 200 B.P. when the query size is 1M, and only about 2000 B.P. when the query is 10M. Figure 13 and 14 are the visualization result of search for the query sequence of 1MB, 10MB in the chromosome 1 of the Human. Red random plot is a visualization of Human chromosome 1, and blue point is the location where the query was searched. Through the visualization results, we can see that a query of 1MB size was found at a relatively early stage of the reference sequence, and a query of 10MB size was at the end of the sequence. This is consistent with the position in the actual sequence, and represents a search result in a more intuitive.

Table 7: The result of sequence search for origin sequence and modified sequence in Human chromosome 1

Q sq.	Unit sz. (bp)	Vec.sz (px)	error Dist.	Sim. (%)	Find.t (s)
A1-0	28	11200	0	99.29	17.269
A1-5	27	10800	150	97.27	21.341
A1-10	26	10400	840	91.34	23.213
A1-20	23	9200	120	88.75	22.514
A4-0	92	36800	1160	92.81	6.537
A4-5	90	36000	160	98.41	6.896
A4-10	88	35200	1040	92.68	7.678
A4-20	80	32000	1040	86.3	9.132
A10-0	154	61600	1120	93.88	13.665
A10-5	150	60000	560	97.21	16.065
A10-10	148	59200	280	95.09	14.245
A10-20	134	53600	2020	81.95	22.241

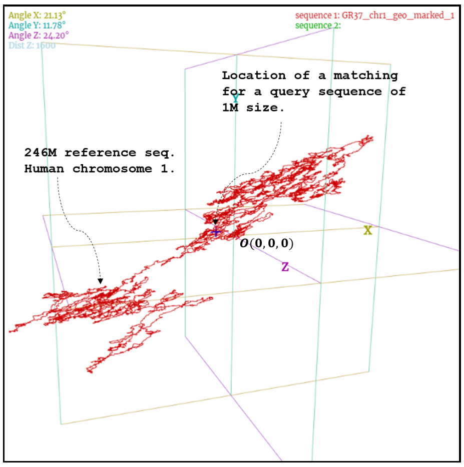

Figure 13: Searching result of query sequence (A1-0) in reference sequence (Human chromosome 1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

4.4 Experiment:Biological Sequence Search over whole genome sequence

Table 8 shows the results of searching a biological query sequence in a whole genome sequence. The search for the LTR sequence (Q-F-1) extracted from the flatfish chromosome 1 resulted in a similarity of 85.7% within 90 B.P. of the actual query position within about 0.4 seconds of search time. On the other hand, the HER-V sequence (Q-H-1) extracted from Human chromosome 1 took relatively longer time, longer than 40 seconds because the length of the query sequence was short and the length of the reference sequence was long. The difference between the actual position and the search result is about 2000 B.P., which is relatively accurate considering that the length of the reference sequence is more than 200M.

Figures 15,16,17 and 18 visualize the flatfish chromosome 1,2,3,5 sequences, respectively. The red one is a visualization of the whole genome of a flatfish, and the area marked in blue is where each query was searched. Figures 17 and 18 show that the marked positions are almost identical to the origin, reflecting that the Q-F-3 and Q-F-5 queries are actually located within 0.5 % of the flatfish whole genome sequence. On the other hand, Figures 15 and 16 reflect that the marked positions are relatively far away from the origin, that the positions of the Q-F-1 and Q-F-2 queries are actually located within 7% and 10% of the flatfish whole genome sequence. Figure 19 visualizes the Human chromosome 1 sequence and marks the result of searching the Q-H-1 query. It is well reflected that the Q-H-1 query is actually located in the early 63 % (about 155 MB.P.) of the Human sequence. Figure 20 is the result of original query sequence (Q-H-1) and enlarged subsequence of the reference sequence (R-H-1) at searched position. The similarity of the searched sequence in the reference (green plot) was 78%, and it can be confirmed that

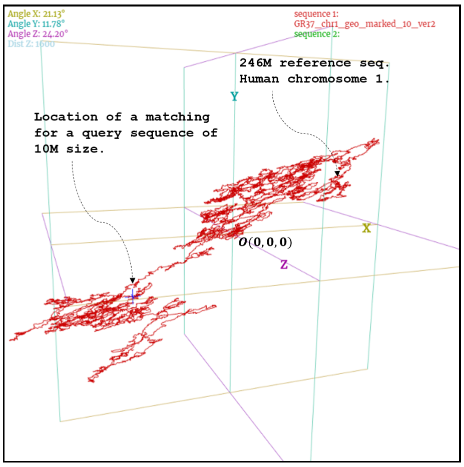

Figure 14: Searching result of query sequence (A10-0) in reference sequence (Human chromosome 1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

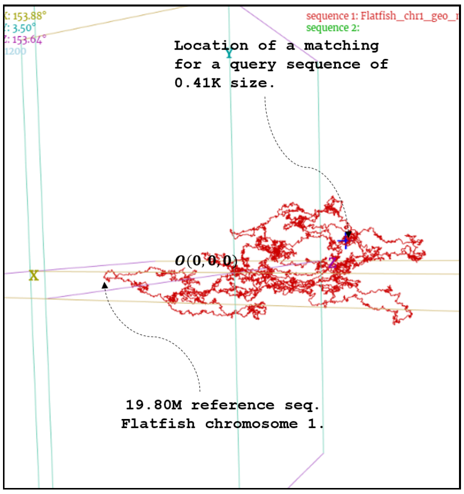

Figure 15: Searching result of query sequence (Q-F-1) in reference sequence (R-F-1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

the query is very similar to the query when matched with the query sequence.

Table 8: The result of sequence search for biological query sequence in flatfish and Human chromosome 1.

Q	Unit sz.	Vec.sz	error	Sim.	Find.t
sq.	(bp)	(px)	Dist.	(%)	(s)
Q-F-1	1	413	90	85.70	0.400
Q-F-2	1	1540	180	72.40	1.030
Q-F-3	1	4780	960	69.10	0.452
Q-F-5	1	6443	1230	75.20	2.038
Q-H-1	1	8063	2130	78.40	41.011

5 Conclusion

Most genome sequence analyses proceed through comparative analysis by finding similar sequence data. Therefore, there is a need for a technique to quickly compare and search for large amounts of sequence data. The alignment technique is a very accurate method to compare sequences, but its high time and space complexity is inadequate to handle large sequences. To overcome these disadvantage, we suggest a new method for comparison and finding for Mega size sequence. Converts the genome sequence as a random plot on the three-dimensional, followed by replacing the sequence comparison problem with geometric object comparison problem. As a result of experiments, similarity precessed by our comparison model accurately reflects the modified ratio between the modified sequence and the original sequence. Most analytical studies based on visualization derive only a single result because they derive a numerical value based on the final result of the visualization. The search and comparison method based on the sequence visualization proposed in this study has high value of utilization of information because all compressed partial visualization information is used for searching sequence. It is useful in that the partial similarity of the sequence can be measured. In addition, a query sequence of size 1-10M was searched in a whole genome sequence of 200M or more, and a relatively precise position was found for the original sequence as well as the modified sequence up to 20%. Also the search time 25 seconds or less, was confirmed handled in a very improved speed compared to the alignment algorithm.

On the other hand, when a sequence with a shorter kilobyte unit length is used as a query, such as an LTR sequence, the compression rate is lowered at the time of visualization, resulting in a lower compression rate of the reference sequence, which leads to a longer search time. However, considering the length of the reference, we can confirm that the position searched is relatively accurate.

The proposed alignment-free searching method is very fast and effective to find a long query sequence over the whole genomes whose size is more than multi-hundreds mega-bytes. It was able to compare and search the sequence at a much improved rate than the alignment-based model by modifying the sequence data into a three-dimensional random plot object and comparing the similarity with the compressed information. Searching algorithm based on alignment method is popular and works good biological sequence comparison but if the size of query and target reference is very large (more than 100 mega bases) the alignment base algorithm requires huge memory space and takes a long computation time. Though our algorithm can't locates the position of query sequence exactly by the DNA base unit, but we can use this procedure as one preprocessing step to find query sequence.

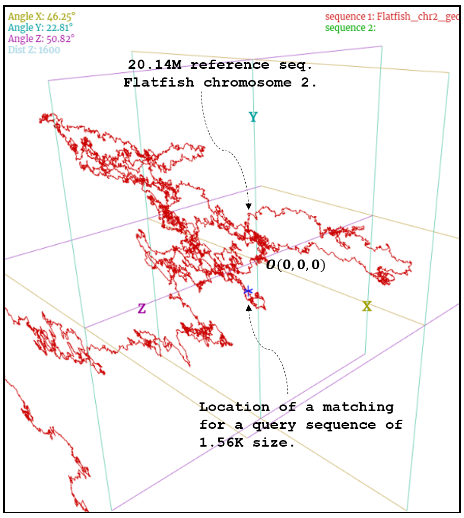

Figure 16: Searching result of query sequence (Q-F-2) in reference sequence (R-F-2). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

Acknowledgement

This research was supported by the Collaborative Genome Program of the Korea Institute of Marine Science and Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries (MOF) (No. 20140428).

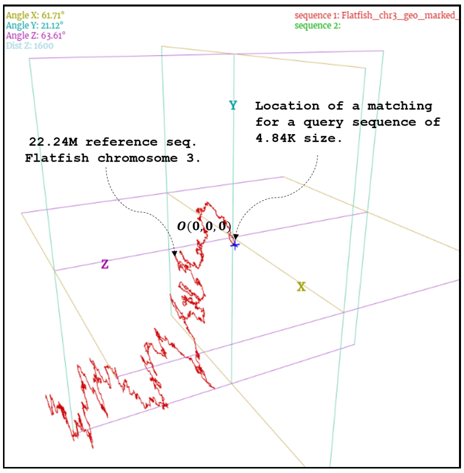

Figure 17: Searching result of query sequence (Q-F-3) in reference sequence (R-F-3). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

References

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990. `https://doi.org/10.1016/S0022-2836(05)80360-2`.

[2] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison—a review. *Bioinformatics*, 19(4):513–523, 2003. `https://doi.org/10.1093/bioinformatics/btg005`.

[3] Tetsuya Nakamura, Keishi Taki, Hiroki Nomiya, Kazuhiro Seki, and Kuniaki Uehara. A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis and Applications*, 16(4):535–548, 2013. `https://doi.org/10.1007/s10044-011-0262-6`.

[4] Milan Randić, Marjan Vračko, Jure Zupan, and Marjana Novič. Compact 2-d graphical representation of dna. *Chemical physics letters*, 373(5):558–562, 2003. `https://doi.org/10.1016/S0009-2614(03)00639-0`.

[5] Yongfan Li, Guohua Huang, Bo Liao, and Zanbo Liu. H-l curve: a novel 2d graphical representation of protein sequences. *MATCH-COMMUNICATIONS IN MATHEMATICAL AND IN COMPUTER CHEMISTRY*, 61(2):519–532, 2009. `https://doi.org/10.1016/j.cplett.2008.07.046`.

[6] Milan Randić. Graphical representations of dna as 2-d map. *Chemical Physics Letters*, 386(4):468–471, 2004. `https://doi.org/10.1016/j.cplett.2004.01.088`.

[7] Yonghui Wu, Alan Wee-Chung Liew, Hong Yan, and Mengsu Yang. Db-curve: a novel 2d method of dna sequence visualization and representation. *Chemical Physics Letters*, 367(1):170–176, 2003. `https://doi.org/10.1016/S0009-2614(02)01684-6`.

[8] Milan Randić, Nella Lerš, Dejan Plavšić, Subhash C Basak, and Alexandru T Balaban. Four-color map representation of dna or rna sequences and their numerical characterization. *Chemical physics letters*, 407(1):205–208, 2005. `https://doi.org/10.1016/j.cplett.2005.03.086`.

[9] Martin Krzywinski, Jacqueline Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645, 2009. `https://doi.org/10.1101/gr.092759.109`.

[10] Jiyuan An, John Lai, Atul Sajjanhar, Jyotsna Batra, Chenwei Wang, and Colleen C Nelson. J-circos: an interactive circos plotter. *Bioinformatics*, 31(9):1463–1465, 2015. `https://doi.org/10.1161/CIRCULATIONAHA.115.015220`.

[11] Bo Liao and Kequan Ding. A 3d graphical representation of dna sequences and its application. *Theoreti-*

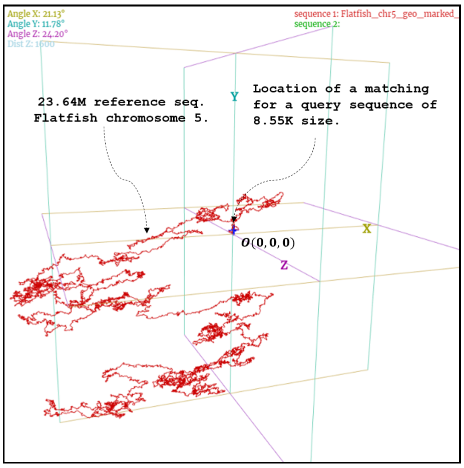

Figure 18: Searching result of query sequence (Q-F-5) in reference sequence (R-F-5). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

cal Computer Science, 358(1):56–64, 2006. https://doi.org/10.1016/j.tcs.2005.12.012.

[12] Alexey Pasechnik, Aleksandr Mylläri, Tapio Salakoski, A Mylläri, T Salakoski, and T Salakoski. Dynamical visualization of the dna sequence and its nucleotide content. *Proceedings of KRBIO*, 5:47–50, 2005.

[13] Min-Ah Kim, Eun-Jeong Lee, Hwan-Gue Cho, and Kie-Jung Park. A visualization technique for dna walk plot using k-convex hull. *Journal of WSCG*, 5(1-3):212–221, 1997.

[14] Daegeon Kwon. Whole genome data visualization and analysis using 3d random walk plot. Master's thesis, Pusan National University, 2015.

[15] Lee Da-Young, Kim Kyung-Rim, Kim Taeyong, and Cho Hwan-Gue. Comparison-specialized visualization model for whole genome sequences. *Journal of WSCG*, 24(2):43–52, 2016.

[16] Hwan-gue Cho Dayoung Lee, Daegeon Kwon. WebGL based Visualization System for Whole Genomes. In *Proceedings of KIISE*, pages 1414–1416. KOREA INFORMATION SCIENCE SOCIETY, 2016.

[17] Milan Randić, Jure Zupan, Dražen Vikić-Topić, and Dejan Plavšić. A novel unexpected use of a graphical representation of dna: Graphical alignment of dna sequences. *Chemical Physics Letters*, 431(4):375–379, 2006. https://doi.org/10.1016/j.cplett.2006.09.044.

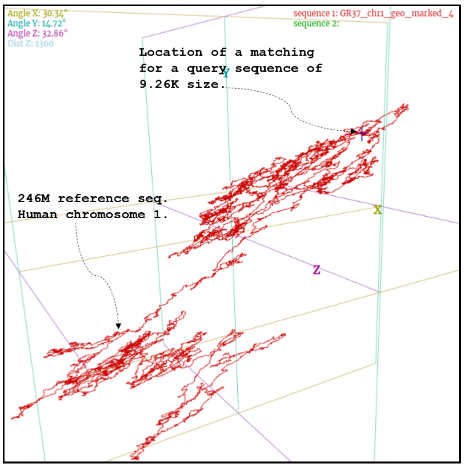

Figure 19: Searching result of query sequence (Q-H-1) in reference sequence (R-H-1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

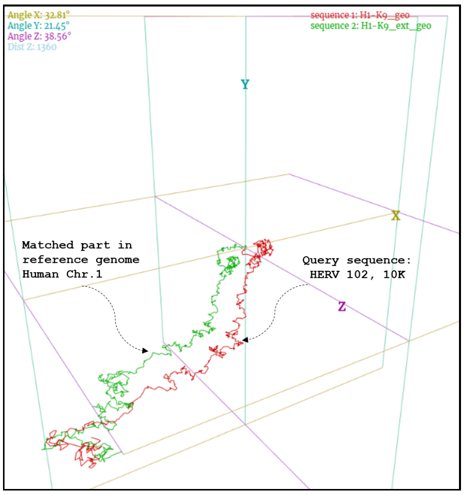

Figure 20: Matching result between the query sequence (Q-H-1) and the extended subsequence of reference sequence (R-H-1), which was depicted as a blue cross in Figure 19.