2023 3rd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT) | 979-8-3503-0235-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICFEICT59519.2023.00079

Sequence Matching and Genealogy Tree Drawing Based on Edit Distance Algorithm

1st Yuxuan Guo*
School of Aeronautics and Astronautics
Xi'an Jiaotong University
Xian, China
m15202413618@163.com

2nd Wenjian Zhang
Qian Xuesen Honors College
Xi'an Jiaotong University
Xian, China
zwj15809183657@163.com

3rd Haochen Cui
School of Aeronautics and Astronautics
Xi'an Jiaotong University
Xian, China
cuihaochen0531@163.com

4th Ning Li
School of Automation and Information Engineering,
Xi'an University of Technology
Xian, China
lining83@xaut.edu.cn

***Abstract*—Sequence homology is widespread in the genetics and evolution of organisms, and is especially evident in families. Sequence homology is often a biological homology between DNA, RNA or protein sequences, and through homology studies, the relationship between two sequences and the evolution and variation of a common ancestral sequence can be demonstrated by their similarity. The edit distance algorithm has a simple structure, high accuracy, and low time complexity. In this paper, the distance between two sequences is determined by a sequence comparison method based on the edit distance algorithm. Subsequently, the complexity and accuracy of the sequence distance algorithm are analyzed, and based on the sequence distance model, a sequence lineage model is designed to determine the ancestral sequences and draw the lineage tree. The algorithm structure of this method is simple, with high accuracy and low time complexity, which has a positive effect on solving the sequence comparison problem.**

Keywords-sequence; matching; edit distance; genealogy chart; data mining

I. Introduction

Biological sequence alignment is the core task of bioinformatics and is the main way to understand the structure and function of biological macromolecules and the connections and differences in the evolutionary history of organisms [1]. The basic problem of sequence alignment is to compare the similarity or dissimilarity of two or more symbolic sequences, and the alignment of DNA, RNA and protein sequences is an important research element in bioinformatics [2] [3].

RNA is an important carrier of genetic information present in biological cells as well as in some viruses and virus-like organisms. It plays an important role in the evolution of organisms, and the existing studies have shown that the sequence alignment of RNA is a hot research topic in biological sequence alignment. For example, new long-stranded non-coding RNAs were cloned and identified by sequence matching, Sanger sequencing, protoplast transient expression and genetic transformation of poplar [4]; RNA sequence matching analysis was used to show that the three types of viruses with apple discoloration symptoms have high homology in different regions, respectively [5]; sequence matching and other methods were used to understand the DNA sequences of paired pol regions of HIV-infected individuals and RNA Sequence alignment and other methods to understand the differences in the construction of molecular transmission networks of HIV-infected pairs of pol region DNA sequences and RNA sequences [6].

At present, sequence alignment of RNA can be divided into two categories according to the number of selected sequences, and these two types of sequence alignment are pair-wise sequence alignment (PSA) and multiple sequence alignment (MSA) [7]. The dual class sequence alignment is mostly done mathematically, while the multiple sequence alignment mostly uses heuristic algorithms [8], mainly progressive alignment and iterative alignment [9].

The main PSA algorithms are FASTA and BALST algorithms. the FASTA algorithm (using candidate regions) is used to store polymer sequence data in the FASTA file format, which has been the standard for bioinformatics files used for decades [10]. The BLAST algorithm is used by biomedical researchers to classify their query sequences by using the BLAST generated pairs [11]. Find matching fragments between query and target sequences and filter the extensions that are far apart. The high scoring regions with scores above a certain threshold are selected and output in descending order. These two algorithms trade the decrease in the sensitivity of prediction for the increase in speed [8], and the accuracy is not high enough.

The main MSA algorithms are ClustalW, YAMA and MUSCLE algorithms [12].

The ClustalW adopts the progressive matching algorithm, using ClustalW for multiple sequence matching [13], which constructs a distance matrix based on the two-pairs, and then constructs a guide tree, adding each sequence in turn until all sequences are added, and outputs the final matching results. YAMA also adopts a progressive matching method to maximize the summation value of two pairs at a time, pre-set a set of matching patterns, and then select the results that match the previously set patterns from the final high-scoring pairs. The disadvantage of the asymptotic matching method represented by the ClustalW algorithm and the YAMA algorithm is that the algorithm tends to fall into local optimal solutions [9].

The MUSCLE algorithm fine-tunes the sequence comparison results by means of iteration [14], based on an algorithm that produces the comparison, and refines the

979-8-3503-0235-6/23/$31.00 ©2023 IEEE
DOI 10.1109/ICFEICT59519.2023.00079

multiple sequence comparison results by iteration until the comparison results are no longer improved. The iterative matching algorithm does not guarantee optimal matching results, but it is robust and insensitive to the number of sequences. The algorithm has low time complexity but high space complexity.

None of the currently available multiple sequence matching tools is perfect; they all use an approximate algorithm. Most of the existing conventional methods have high spatio-temporal complexity, and the rest of them improve the speed by sacrificing the sensitivity of the comparison. To address the shortcomings of conventional methods, we propose a sequence matching method based on the edit distance algorithm, which has a simple algorithmic structure, high accuracy, and low time complexity. Firstly we quickly measure the distance between two sufficiently long base sequences. Secondly, the complexity and accuracy of the algorithm are analyzed and examples are given to explain and illustrate. Finally we determine the ancestral sequences and draw the genealogical tree.

II. Basic Theory

A. RNA Base Change Mode

RNA base sequence changes usually exist only in the three cases of insert, modify, delete. In view of the small probability of occurrence of the influence of other factors on the way of base sequence change, it is not considered in the process of the study. Figures 1-3 depict the schematics of three types of gene transformations, respectively.

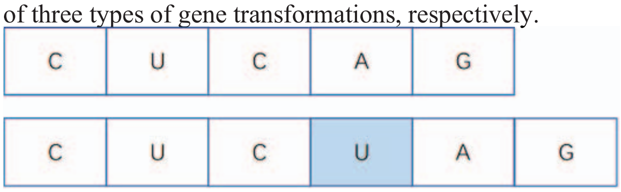

Figure 1. Insert：insert a character

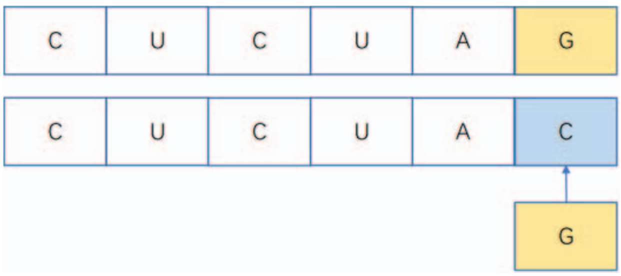

Figure 2. Modify: replace one of the characters with another character

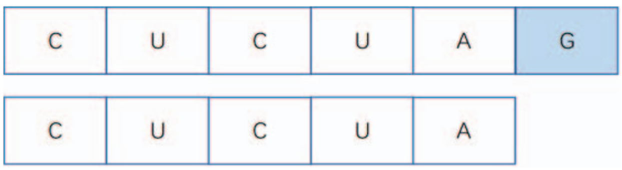

Figure 3. Delete: delete a character

Mutations in base sequences are related to many factors, and it can be assumed that no new changes in their sequences occur during the short period of analysis of base sequence studies. When genealogical analysis is performed for some sequences, the ancestors can be assumed to be in those sequences analyzed and not others, due to the high number of ancestral possibilities, in order to simplify the model.

B. Editing Distance Algorithm Principle

Assume that $d[i, j]$ steps can be used (a two-dimensional array can be used to store this value) to represent the minimum number of steps needed to convert the string $s[1...i]$ to the string $t[1...j]$. In the most basic case, i.e., if the string s is empty when i equals 0, the corresponding $d[0, j]$ is to increase j characters to make s convert to t. If the string t is empty when j equals 0, the corresponding $d[i, 0]$ is to decrease i characters to make s convert to t.

Then consider the general case of dynamic programming, in order to get $s[1...i]$ transformed into $t[1...j]$ after a minimum number of add, delete, or replace operations, it must be possible to add, delete, or replace operations with a minimum number of times before, so that the string s and string t now only need to do one more operation or not to complete the transformation of $s[1...i]$ to $t[1...j]$. The so-called "before" is divided into the following three cases:

1) can convert $s[1...i]$ to $t[1...j-1]$ in k operations

2) can convert $s[1...i-1]$ to $t[1...j]$ in k operations

3) can convert $s[1...i-1]$ to $t[1...j-1]$ inside k steps

For the first case, just add $t[j]$ to $s[1...i]$ at the end to finish the matching, so a total of $k+1$ operations are needed. For the 2nd case, just remove $s[i]$ at the end and then do the k operations, so a total of $k+1$ operations are needed. For case 3, just replace $s[i]$ with $t[j]$ at the end so that $s[1...i] == t[1...j]$ is satisfied, which also requires a total of $k+1$ operations. And if in the 3rd case $s[i]$ is exactly equal to $t[j]$, it can be done using only k operations. Finally, to ensure that the number of operations obtained is always the minimum, the minimum number of operations required to convert $s[1...i]$ to $t[1...j]$ can be selected from the three cases above as the least consuming one.

III. Solution

The steps of the proposed method are illustrated in Figure 4.

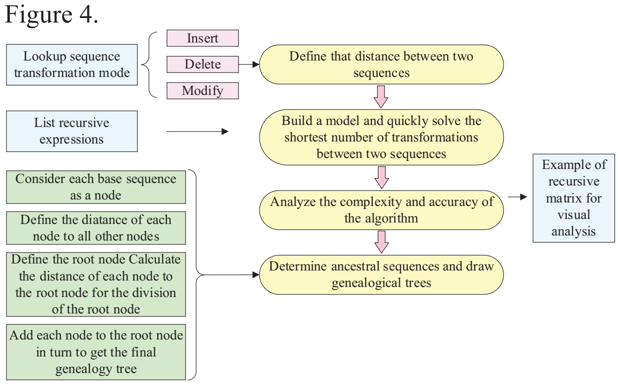

Figure 4. Main steps of the propose and method

The proposed method is divided into three main parts, defining the distance between two sequences and solving it quickly, analyzing the complexity and stability of the algorithm and drawing the family tree.

The distance between two sequences is defined based on the way of base sequence change above, i.e., the minimum number of operations to transform sequence i into sequence j after these three operations. Since the problem can be decomposed into a series of subproblems, it can be solved by a dynamic programming algorithm. Finally, experiments are conducted to illustrate the sequence distance model and visualize it using graphs.

For the sequential distance model, the complexity and accuracy are analyzed. Since the dynamic programming algorithm is used in the solution process, based on the above assumption that the sequences do not change abruptly due to some factors in the study process, and each step in the solution process can be considered accurate at each step. For solving two sequences of length n and m, the time complexity of the sequence distance model can be obtained as $O[nm]$.

Based on the distance sequence model, the sequence genealogy model is designed to determine the ancestral sequence and draw the genealogy tree. The pseudo-code of this algorithm is given in the construction of the genealogy model. In the process of class merging, we use the merge set for optimization to reduce the complexity of merging to $O[n\log n]$. In the solution of the genealogy model, experimental descriptions are given and the genealogy is constructed for visualization.

A. Sequence Distance Definition and Solution

Based on the edit distance algorithm to quickly measure the distance between two sufficiently long base sequences, the number of these bases is often greater than 1000. first we need to define the distance between two bases, we consider that for two bases A, B of length i, j, the distance between the two is the operation method to change A into B. Where operate is the operation method to change one base to another base by increase, modify, delete, and find the minimum number of operations to make A change to B.

In order to transform A to B, we can transform by insert, modify, delete. insert: insert a character Modify: replace one of the characters with another Delete: delete a character Since the subsequent operations depend on t。he previous operations, the problem is a dynamic planning problem.

$dist_{edit}(i,j)$ can be derived recursively from $dist_{edit}(i-1,j)$, $dist_{edit}(i,j-1)$ and $dist_{edit}(i-1,j-1)$.

$$\mathrm{dist}_{\mathrm{edit}}(i,j)=\begin{cases}\max(i,j) & \text{if } \min(i,j)=0,\\ \min\begin{cases}\mathrm{dist}_{\mathrm{edit}}(i-1,j)+1\\ \mathrm{dist}_{\mathrm{edit}}(i,j-1)+1\\ \mathrm{dist}_{\mathrm{edit}}(i-1,j-1)+1_{(a_i\neq b_j)}\end{cases} & \text{otherwise.}\end{cases} \tag{1}$$

$1_{(a_i\neq b_j)}$ is an indicator function, when $a_i=b_j$, its value is 0, other times it equals 1 [15].

B. Algorithm Complexity and Accuracy Analysis

The flowchart in Figure 5 illustrates the algorithm for calculating sequence distance.

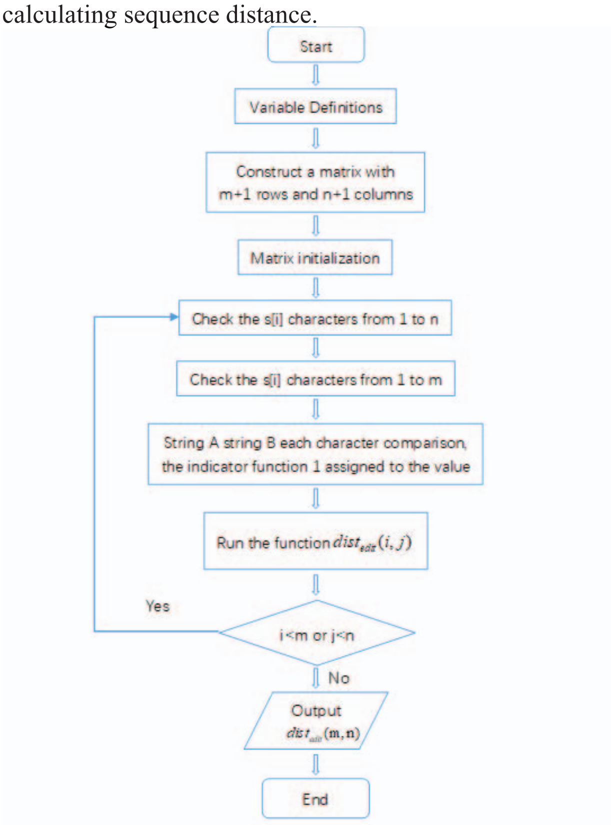

Figure 5. Sequence distance solving algorithm flow chart

For the distance calculation of two sequences of length n,m, the time complexity of the algorithm is $O[nm]$, the algorithm is error-free, no other anomalies occur in the process, and the accuracy is excellent.

C. Mapping Genealogical Trees

After defining the distance between two sequences as $dist_{edit}$, it is assumed that the ancestor nodes are in these selected sequences. For each sequence, we define its distance in the sequence as all_{dist}.For the ith sequence with the total number of sequences m, all_{dist} is defined as follows:

$$all_{dist}(i)=\sum_{j=1\&j!=i}^{j=m} dist_{edit}(i,j) \tag{2}$$

After calculating all_{dist} for each sequence, we select the smallest sequence as the ancestor node and then perform a lineage analysis [15].

We calculate the all_{dist} of each node except the ancestor node, sort them from smallest to largest, find the first node closest to the ancestor node, and put it after the ancestor node. After that, we consider this node and the ancestor node as a class, and select the node closest to this class, and choose whether to pick it up after this node or the ancestor node according to the distance. The above operation is iterated until all nodes are in the same class. The flowchart in Figure 6 illustrates the algorithm for constructing a pedigree tree.

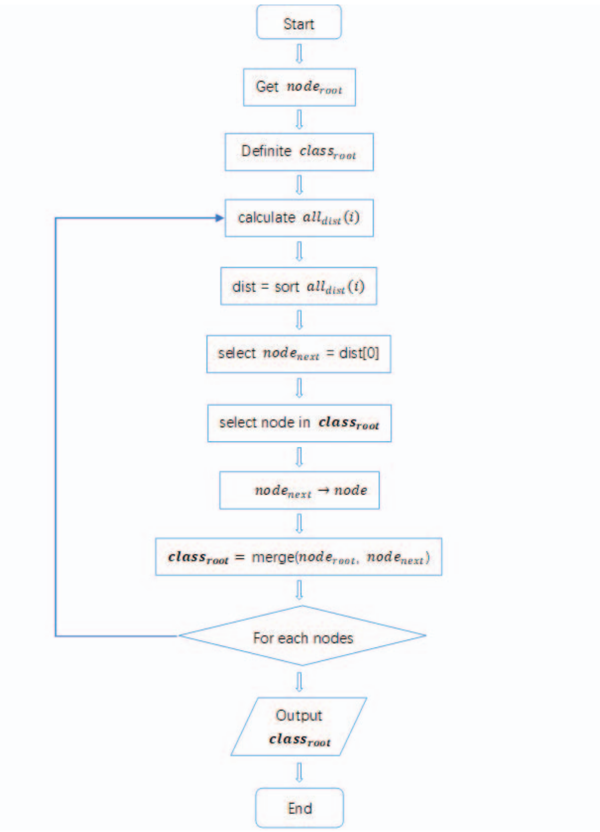

Figure 6. Flow chart of the genealogical tree algorithm

For the merge operation, we can use the merge set for optimization:

1) Initialize class

For each node, we initialize its class to itself.

2) Find root

For each newly selected $node_{next}$, find the root node of the class it wants to merge, i.e. $class_{root}$.

3) Merging these two categories

Merge the class of new nodes into the class of $class_{root}$.

Example:

Step1: Initialize the 8 nodes into 8 classes. As shown in Figure 7.

Figure 7. First step of operation

Step2: Find the 1st, 0th and 7th nodes and merge them to the root node of this class to reduce the complexity of merging two classes subsequently. As shown in Figure 8.

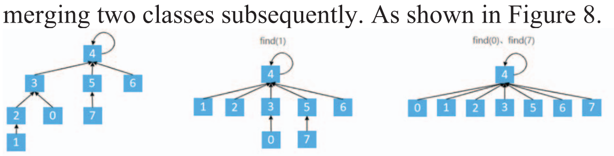

Figure 8. Second step of operation

Step3: Merge the two classes with root node 3 and 4. As shown in Figure 9 and Figure 10.

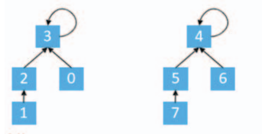

Figure 9. Before merger

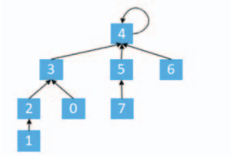

Figure 10. After merger

After the optimal merge operation using the merge set, the time complexity can be effectively reduced to $O[n\log n]$ for the set of sequences with n number of merged nodes.

IV. Experimental Verification

A. Sequence Distance Solving

In the short term process of base sequence research and analysis, no new changes occur in its sequence, only three changes occur between two sequences insert, modify, delete, and no mutation occurs in the sequence during the analysis process, two existing RNA sequences, CUCUAG and CUCGG, use the sequence distance solving algorithm to experimentally find out the distance between the two sequences. Figures 11 through 14 depict the specific solving process.

	C	U	C	U	A	G
C						
U						
C						
G						
G						

Figure 11. Create a matrix to store the distance calculated in the previous step

	C	U	C	U	A	G
C	0	1	2	3	4	5
U	1					
C	2					
G	3					
G	4					

Figure 12. Initialize all distances in the first row and column

	C	U	C	U	A	G
C	0+t=0	1+1=2	2	3	4	5
U	1+1=2	minimum value=0				
C	2	And so on=1				
G	3	2				
G	4	3				

Figure 13. Start a loop and calculate all distances until the last character

	C	U	C	U	A	G
C	0	1	2	3	4	5
U	1	0	1	2	3	4
C	2	1	0	1	2	3
G	3	2	1	1	2	3
G	4	3	2	2	2	2

Figure 14. Final results

The final distance between the two RNA sequences can be obtained as 2.

B. Genealogy tree generation

In the case that the ancestral sequences are among these following these sequences and not others, The 8 RNA sequences in Table 1 can be utilized to conduct lineage analysis and construct a lineage tree with the aforementioned sequences.

TABLE I. BASE SEQUENCE LIST

Number	Base sequence
1	AUCGCU
2	UCGAAU
3	AUCCGU
4	AUCUG
5	AGGUA
6	AGCUA
7	CGGCU
8	UAAGCA

TABLE II. SEQUENCE all_{dist} VALUE TABLE

i	1	2	3	4	5	6	7	8
all_{dist}	22	28	25	24	22	21	24	28

From the results in Table 2, we can see that $all_{dist}(6)$ is the smallest, so the ancestor node is node 6 and node 6 is added to $class_{root}$. Since $dist_{edit}(6,5)$ is the smallest, node 5 is added to $class_{root}$, under node 6. At this point there are two nodes 5 and 6 in $class_{root}$. Then iterate, at this time, treat 5 and 6 as 1 node, calculate $all_{dist}(i)$ (again, at this time $i=1$, $all_{dist}(i)$ obtains the minimum value 21, and $dist_{edit}(6,1)$ is smaller than $dist_{edit}(5,1)$, so connect node 1 to the back of node 6, and similarly connect node 3 to the back of node 1, connect node 4 to the back of node 3, and connect node 2 to the back of node 1. The node 2 is followed by the node 1, the node 7 is followed by the node 1, and the node 8 is followed by the node 5.

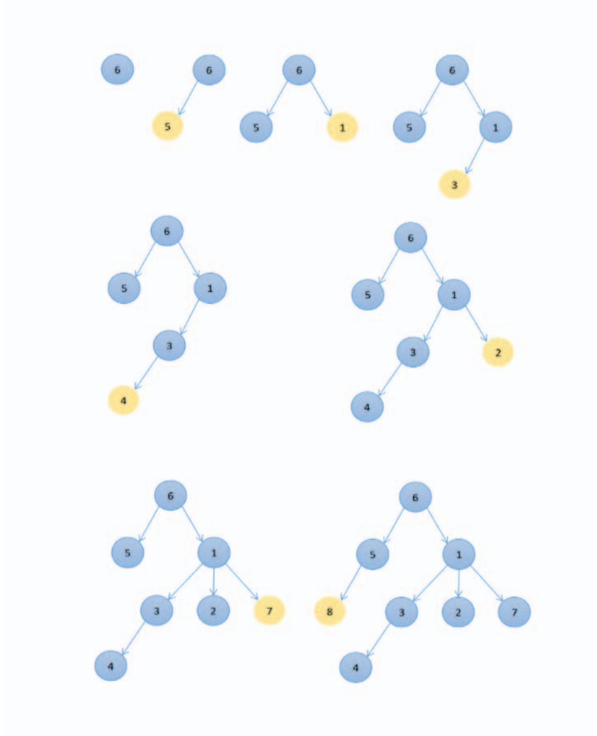

Figure 15. Genealogy tree generation

The final lineage tree obtained according to the sequence branch model is shown in Figure 15.

As the above experiments can be concluded that the algorithm structure of the sequence matching method based on edit distance algorithm is simple, with high accuracy and low time complexity.

V. Conclusion

We define the concept of distance between two sequences, which effectively describes the process of change between sequences. The sequence distance model can solve the minimum operations for the transformation between two sequences in a short complexity. The sequence lineage model designs an efficient, reasonable and intuitive algorithm for lineage establishment and optimizes it using concurrent sets, which effectively reduces the complexity.

Sequence distance models can be used not only for evolutionary studies between base sequences, but also for solving inter-sequence transformations. Sequence genealogy models can be used not only for the establishment of biological base sequence genealogy, but also for background problems with correlation and homology.

The downside is that the sequential distance model can be optimized using new data structures or algorithms.

References

[1] L. S. Ali and K. Mehrdad. (2022) Developing new genetic algorithm based on integer programming for multiple sequence alignment. Soft Computing, PP 1-8.

[2] J. Song. (2013) Application of machine learning methods in biological sequence analysis. Zhejiang Province: Zhejiang University.

[3] H. Liu. (2022) Research on biological multiple sequence matching method and its key technology. Anhui Province: University of Science and Technology of China.

[4] N. Ran, S. Liu, H. Qi, J. Wang, T. Shen, W. Xu, M. Xu. (2023) Long Non-Coding RNA lncWOX11a Suppresses Adventitious Root Formation of Poplar by Regulating the Expression of PeWOX11a.International Journal of Molecular Sciences, Volume 24, Issue 6. PP 5766-5766.

[5] F. Zha et al. (2023) Analysis of the species, molecular characteristics and correlation with symptoms of apple discoloration symptoms accompanied by related virus classes. Journal of Phytopathology.

[6] J. Chen et al. (2023) Comparative pairwise study of molecular transmission networks based on DNA or RNA sequences of HIV-1 pol gene. China Tropical Medicine.

[7] Y. Wang, H. Wu and Y. Cai. (2018) A benchmark study of sequence alignment methods for protein clustering. BMC bioinformatics, Volume 19 , Issue Suppl 19. PP 529.

[8] Y. Zhu and Z. Xiong. (2007) DNA sequence data mining technology. Journal of Software.2007(11).

[9] H. Yang. (2012) Research on biological sequence data mining technology. Journal of Hefei University of Technology (Natural Science Edition)， 2012，35（09）.

[10] H. Youri, J. Guido W. and W. Harmen J. G. van de. (2021) FASTAFS: file system virtualisation of random access compressed FASTA files. BMC Bioinformatics, Volume 22, Issue 1. PP 535-535

[11] C. Christiam, B. Grzegorz M, J. Victor, A. R. Vera and M. Thomas L. (2023) ElasticBLAST: Accelerating Sequence Search via Cloud Computing. bioRxiv : the preprint server for biology.

[12] Edgar, R.C. (2022) Muscle5: High-accuracy alignment ensembles enable unbiased assessments of sequence homology and phylogeny. Nat Commun 13, 6968

[13] H. Jui, W. Zhiping. (2016) Sequence Alignment and Homology Search with BLAST and ClustalW. Cold Spring Harbor Protocols, Volume 2016, Issue 11. PP pdb.prot093088.

[14] Q. Zhao and W. Bin. (2016) Some research discussions on biological sequence data mining techniques. Shandong Industrial Technology.

[15] Z. Feng, J. Zhou and Y. Yu. (2022) Minimum edit distance in terminology research. Chinese Science and Technology Terms. 2022, 24 (03).