Chapter 12
Representation of a DNA Sequence by a Substring of Its Genetic Information

Bacem Saada and Jing Zhang

12.1 Introduction

To determine the affiliation of a strain to a given specie, biologists compare it with a known sequence of reference of the specie to which it is presumed to belong. If the similarity percentage is very large, we conclude that this sequence belongs to a well-defined specie. The comparison between sequences can also allow the comparison between different species. These comparisons lead to the conclusion that two species have a common ancestor or not.

In order to properly analyze the results of alignment methods and comparison of DNA sequences, we assign weights to the various pairs of the sequence to calculate the degree of similarity and the costs of non-similarity between sequences. This operation allows us to infer relationships between the sequences. This relationship is described as the degree of similarity between sequences. This degree of similarity is quantified by a score. The most commonly used alignment algorithms between sequences are the Smith-Waterman algorithm [1] which determines local alignment between DNA sequences and the algorithm of Smith and Waterman [2] which determines a global alignment between DNA sequences.

B. Saada (✉) · J. Zhang
College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China
e-mail: basssoum@gmail.com

J. Zhang
e-mail: zhangjing@hrbeu.edu.cn

S.-I. Ao et al. (eds.), *Transactions on Engineering Technologies*,
DOI 10.1007/978-981-10-2717-8_12

12.2 State of the Art

The process of alignment and comparison of DNA sequences presents several problems:

Today, there are several open access DNA sequences databases. These banks continue to grow at an exponential rate. In 2006, GenBank, for example, created within the framework of international collaboration on nucleotide sequencing, contained over 65 billion nucleotide bases [3]. In 2013 it grew to 154.2 billion bases. Nowadays, the quantity of information can reach petabytes in size. In this case, applying a treatment on a large number of sequences to infer which sequence belongs to a given specie, is very costly in terms of execution time and needed resources. To decrease the amount of stored information, researchers are trying to reduce the number of DNA sequences stored in their databases and keep only the DNA sequences that best characterize each specie.

Storage of such alignment is also a problem. Thereafter, any analysis or interpretation of this alignment would be strenuous. If the researcher decides to use a portion of the sequence, no current algorithm allows him to optimally choose the desired length to extract from the original chain.

To solve the problems described above and to optimize the use and performance of DNA sequences alignment algorithms, several researches were conducted.

Furthermore, the growth of the new DNA sequences alignment technologies has enabled the study of human genome. The size of those genomes reaches 3 billion bases. It can even reach more than 100 billion bases in some amphibian species. It is not possible to use conventional algorithms for the alignment of DNA sequences. Indeed, the result of an alignment between entire genomes would be an alignment of millions of base pairs. Taking into consideration the execution time, the application of such an operation is impossible for usual microcomputers.

The collection, analysis and understanding of this enormous amount of information became a challenge for taxonomic researches. This has led to the development of DNA compression algorithms. Based on the English text compression of the four bases {A, C, G, T}, those algorithms try to reduce the ratio "bits per base" [4, 5].

As a conclusion, research themes were therefore based on the parallelization of classical DNA sequences alignment algorithms to reduce the execution time of these algorithms or to compress DNA information. No research addresses the reduction of the size of the DNA bases to be stored.

12.3 New Approaches to DNA Sequences Alignment

In this section, we propose an approach to DNA sequences alignment that can represent DNA sequences with a sub chain of their genetic information. First, we list in the first instance, the motivations of our approach. After that, we will present our approach while offering a study of its complexity.

12.3.1 Motivations

To overcome the problems described above, we tried to propose a new approach that attempts to combine the performance of algorithms for DNA sequences alignment and significantly reduce the size of stored genetic information.

Our approach will essentially provide an algorithm that is:

- **Able to determine an optimal alignment for a length requested by the researcher**: Usually alignment resulting from the implementation of the Smith-Waterman algorithm has a length of 1500 base pairs. We will try to present our approach through an algorithm able to give an optimal alignment for a length less than half the size of the sequences to be aligned.
- **Able to represent a DNA sequence, not by its full genetic information but by a smaller sub chain**: It is highly desirable that a DNA sequence is represented not by its full genetic information but by some of its DNA only. Our approach will also try to represent a set of DNA sequences by a sub chain.
- **Able to reduce the amount of data stored in databases**: By reducing the size of the genetic information representative of a DNA sequence, the amount of data stored in the database will be reduced. And thus all data can be stored in the same storage media.

12.3.2 Algorithm for Determining a Best Local Optimal Alignment

12.3.2.1 Introducing the Approach

The major reason that led researchers to use heuristics approaches is that with a large number of sequences, the dynamic programming algorithms are quite expensive in terms of execution time. If further treatment is added, the total execution time will grow more. For that an alignment would be well analyzed. It is desirable to scan the entire alignment. Therefore, the approach that we present seeks to analyze the similarity score of the entire alignment to extract the sub chain with the highest percentage of similarity between the two input data sequences. Indeed, this algorithm, after building the entire sequence alignment, saves the values of the matrix score. Thus, it is possible to find the alignment portion with the highest similarity score to better represent the sequences. This part will be determined after calculating the score of similarity between the different regions of the alignment.

12.3.2.2 Algorithm

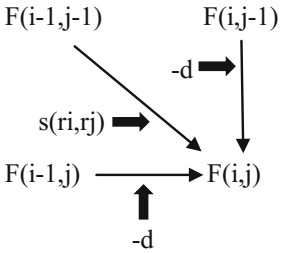
As well as the alignment algorithms between DNA sequences, we perform computing of a score matrix. We will use the following recurrence formula:

$$\mathrm{F(i,\ j)}\ = \max \begin{cases} 0, \\ F(i-1,\ j-1) + s(si,\ yj), \\ F(i-1, j) + d, \\ F(i,\ j-1) + d. \end{cases}$$

For the energy costs of each alignment operation, we will use the following values:

Del = −1; ins = −1; sub = 1; id = 1.

The selection of these values as parameters was made so that energy costs would be uniform and equal in absolute value. This choice will exclude the consideration of these energy costs in any subsequent treatments. We keep track of the highest value in the matrix of the score. The alignment will be built starting with this value following this rule:



For a good alignment analysis, it is desirable to scan the entire alignment built. Therefore, the algorithm starts, like any classical algorithm for DNA sequences alignment, by calculating the score matrix over the entire sequences.

Algorithm 1 Build Local ALignment

Require: $S \geq 0\ I \leq 0\ D \leq 0\ A \neq \varnothing\ B \neq \varnothing$

```
{Compute Matrix}
maxScore ← 0
maxRow ← 0
maxCol ← 0
AlignmentA ← ""
AlignmentB ← ""
i ← maxRow
j ← maxCol
for i ≤length(A) do
  F(i,0) ← 0
end for
for j ≤length(B) do
  F(0,j) ← 0
end for
```

```
for i ≤length(A) do
  F(i,0) ← 0
end for
for j ≤length(B) do
  F(0,j) ← 0
end for
for i ≤length(A) do
  for j ≤length(B) do
    Match ← F(i − 1, j − 1) + M
    Delete ← F(i − 1, j) + D
    Insert ← F(i, j − 1) + I
    F(i,j) ← max(Match, Insert, Delete)
    if F(i,j) > maxScore then
      maxScore ← F(i,j)
      maxRow ← i
      maxCol ← j
    end if
  end for
end for
```

The algorithm keeps track of the matrix cell that contains the highest similarity score. The construction of the alignment starts from the box. After building this alignment, the algorithm seeks to determine the alignment region best suited, in terms of similarity score, to represent the sequences. This part will be determined after calculating the average of the similarity scores of the alignment's different regions.

```
{Function : Build Optimal Alignment}
while ((i ≥ 0 or j ≥ 0) and F(i,j) ≥ 0) do
  Score ← F(i,j)
  ScoreDiag ← F(i − 1, j − 1)
  ScoreUp ← F(i, j − 1)
  ScoreLeft ← F(i − 1, j)
  if Score == ScoreDiag + S(Ai, Bj) then
    AlignmentA ← Ai + AlignmentA
    AlignmentB ← Bj + AlignmentB
    i ← i − 1
    j ← j − 1
  else
    if Score == ScoreLeft + d then
      AlignmentA ← Ai + AlignmentA
      AlignmentB ← " − " + AlignmentB
      i ← i − 1
    end if
```

```
    else
      if Score == ScoreLeft + d then
        AlignmentA ← Ai + AlignmentA
        AlignmentB ← " − " + AlignmentB
        i ← i − 1
      end if
    else
      AlignmentA ← " − " + AlignmentA
      AlignmentB ← Bj + AlignmentB
      j ← j − 1
    end if
    AlignmentB ← Score
  end while
  if F(i, j) ≥ 0 then
    while i ≥ 0 and F(i, j) ≥ 0 do
      AlignmentA ← Ai + AlignmentA
      AlignmentB ← " − " + AlignmentB
      i ← i − 1
      AlignmentB ← F(i, j)
    end while
    while j ≥ 0andF(i, j) ≥ 0 do
      AlignmentA ← " − " + AlignmentA
      AlignmentB ← Bj + AlignmentB
      j ← j − 1
      AlignmentB ← F(i, j)
    end while
  end if
  region ← ""
  for all AliRegion ⊂ ALig do
    if AVGscore(AliRegion) > AVGscore(region) then
      region ← AliRegion

    end if
  end for
```

12.3.2.3 Complexity of the Approach

Consider two sequences seq_1 seq_2. Let $l_1 l_2 l_3$ the respective lengths of the first sequence, the second sequence and of the alignment result; and consider l the length of the chain requested by the researcher.

The complexity of the filling phase is O ($3l_1l_2$). Indeed, to fill each cell of the matrix, we must realize three arithmetic operations. The complexity of the optimal alignment of the construction phase for the required length of O (3l); Indeed, to detect the region with the highest similarity score, our algorithm will perform l arithmetic operations. It would calculate the average of the similarity scores of this region in (l3 − l) total regions.

The total complexity of this approach is O ($3l_1l_2 + 3l + [l_3 - l]\ l$). The complexity of this approach is polynomial of order 2.

12.4 Experimental Results

In this section, we will illustrate the experimental results of our approach over the Smith and Waterman algorithm. In the first part, we will interpret the results of experiments obtained on the similarities percentages. Subsequently, we will present a comparative study based on the execution time of our approach compared to the classical approach.

12.4.1 Species of the Experiments

To measure our approach's performance, we used a set of DNA sequences of different genres. The size of the DNA sequences varies between 1300 and 1550 base pairs.

The diversity of the classification of these sequences allowed us to conduct a comparative study presented in three steps:

- Experimental results for species of a same genus.
- Experimental results for a set of species of the genera of the phylum Firmicutes.
- Experimental results for random species.

12.4.1.1 Experimental Results for Species of the Same Genus

We analyzed the experimental results for 11 species of the genus Bacillus. The species used are amyloliquefaciens Anthracis, Azotoformans, Badius, Cereus, Circulans, coagulans, licheniformis, megaterium mycoides Psychrosaccharolyticus, pumilus. This experiment would analyze the percentages of similarity of alignment operations between these DNA sequences and infer relations of similarities between species of the same genus.

12.4.1.2 Experimental Results for a Set of Species of the Phylum Firmicutes

We analyzed the performance results of our approach on a set made of 33 different species. These species are from different genera but are in the same phylum. All the species used are species of the genera: Alicyclobacillus, Anoxybacillus, Bacillus, Geobacillus, Lactobacillus, Lysinibacillus, Paenibacillus, Sporosarcina. These experiments would determine similarity relationships between the genera in terms of execution time, similarity percentage, and if our approach could build a hierarchical classification according to the taxonomic classification of species.

12.4.1.3 Experimental Results for Random Species

In this part, we analyzed the experiments conducted on any specie regardless of any hierarchical classification. The number of species used to make the experiments was 500 species.

12.4.2 Results in Term of Percentages of Similarity

12.4.2.1 Sequences from the Same Genus

We note that the percentages of similarity of our approach are higher than those of the Smith-Waterman algorithm. For lengths equal to or less than 500 base pairs, the percentage similarities are higher than or equal to 95 %. These similarity percentages describe, at best, in this case, regions with high similarity between species of the same genus (Fig. 12.1).

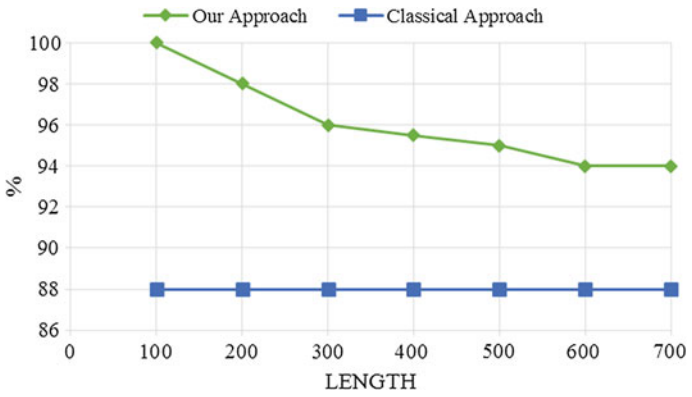


Fig. 12.1 Experiments on sequences from same gender

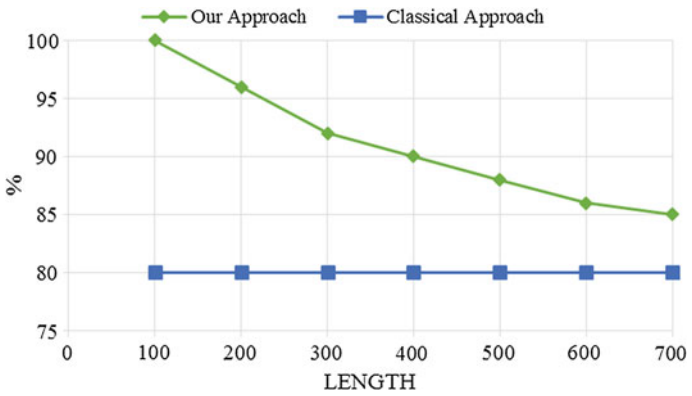


Fig. 12.2 Experiments on sequences from same phylum

12.4.2.2 Sequences from the Same Phylum

The percentages of similarity of our approach reach 92 % for a length of 300 base pairs. Still for shorter lengths, percentages of similarity are around 88 %. These similarity percentages are by 7 % better than the algorithm of Smith and Waterman (Fig. 12.2).

12.4.2.3 Sequences from Random Species

We note that the percentages of similarity have significantly decreased compared to our previous experiments. Indeed, the species used are not similar in taxonomic classification. We also note that for small lengths, lower than 400, the percentages of similarity are higher than 70 %. While for longer lengths, percentages are around 64 % but remain higher than those of the Smith-Waterman algorithm which is less than 57 % (Fig. 12.3).

12.4.3 Experiments in Terms of Execution Time

In this section, we will present a comparative study between our approach and the classical approach of determining an optimal alignment in terms of execution time.

The difference in execution time between the two algorithms is not very important and does not exceed, in the worst case, 1 min and 30 s for Tests with 4000 constructed alignments. This similarity in execution time favors, at most, the use of our optimal approach (Fig. 12.4).

We can therefore conclude that it is desirable to use our approach that determines optimal local alignment not only because it has a higher similarity percentage compared to the other two approaches, but also because in terms of execution time, the difference between the approaches is not quite significant (Fig. 12.5).

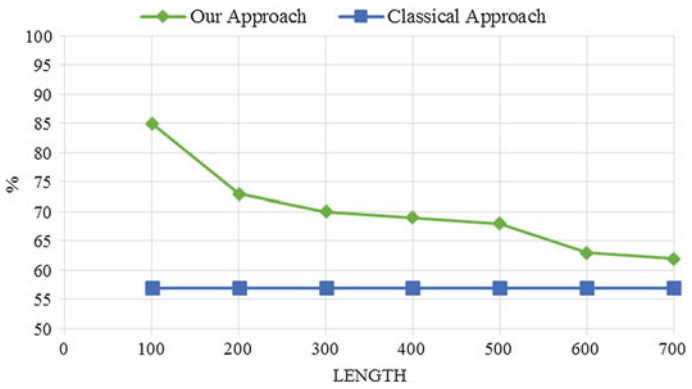


Fig. 12.3 Experiments on random sequences

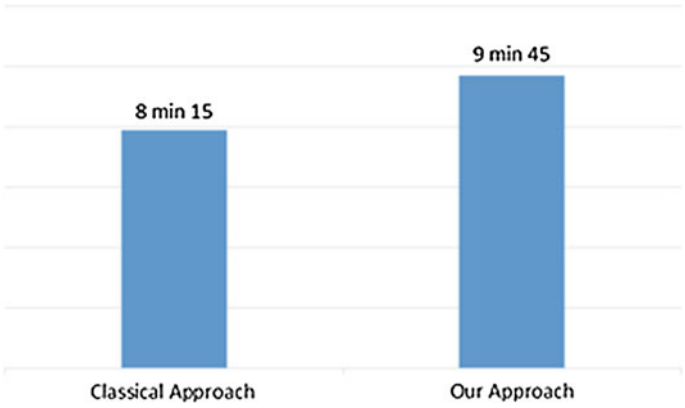


Fig. 12.4 Execution time of our approach for 4000 constructed alignments

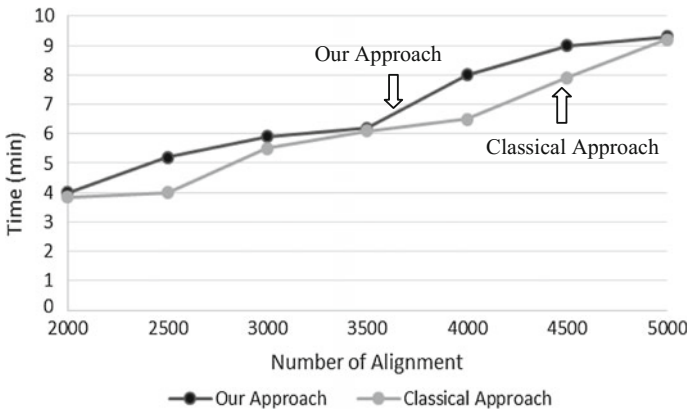


Fig. 12.5 Execution time of our approach and the Smith and Waterman algorithm

12.5 Conclusion and Future Work

Our optimal approach allows researchers to find a sub string called representative of a given DNA sequence [6]. The percentages of similarity are better than the algorithm of Smith and Waterman, and reach 100 % for small lengths. Our approach, then, allows them to reduce the amount of information stored in their databases. This considerable reduction in the size of DNA sequence alignments can reduce the size of databases by a factor of 2.

Nevertheless, we try to do other research in this area to:

- **Propose an algorithm for the compression of DNA sequences representation**: as in the networks, we will try to develop an algorithm for compressing DNA sequences information and reduce its representation, which will reduce the size of the data in databases.
- **Represent a set of DNA sequences by a unique string**: based on our approach, we will try to group multiple species and represent them by a unique string.
- **Find a new representation of DNA information**: it is true that our contribution proposes a decrease in the size of the sequences alignment comparison. Thus, the analysis and the treatment of large number of sequences present big challenges to biologists. We may have a new representation of DNA sequences.

Acknowledgements This work was funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

References

1. Needleman SB, Wunsch CDA (1970) General method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
2. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197
3. Genbank. http://www.ncbi.nlm.nih.gov/genbank/
4. Saada B, Zhang J (2015) Vertical DNA sequences compression algorithm based on hexadecimal representation. Lecture notes in engineering and computer science: proceedings of the world congress on engineering and computer science 2015, WCECS 2015, 21–23 Oct 2015, San Francisco, USA, pp 570–574
5. Saada B, Zhang J (2015) DNA sequences compression algorithm based on extended-ASCII representation. Lecture notes in engineering and computer science: proceedings of the world congress on engineering and computer science 2015, WCECS 2015, 21–23 Oct 2015, San Francisco, USA, pp 556–560
6. Saada B, Zhang J (2015) Representation of a DNA sequence by a subchain of its genetic information. Lecture notes in engineering and computer science: proceedings of the world congress on engineering and computer science 2015, WCECS 2015, 21–23 Oct 2015, San Francisco, USA, pp 536–540