Representation of a DNA Sequence by a Subchain of its Genetic Information

Bacem Saada, Jing Zhang

***Abstract*—The technological developments of recent years has helped biologists to extract, examine and store the genetic information of living beings. Thus, the databases become very large and contain a large amount of redundant or poorly analyzed information. This increase in size becomes a great challenge for the data storage. There will be, in this case, a difficulty to properly analyse, rank well, and save all the data. As a solution to this problem, we propose, through this article, an algorithm for determining the optimal local alignment and in which we offer the possibility of representing a DNA sequence by a substring of its genetic information and therefore reduce the amount of data banks information.**

***Index Terms*—bioinformatics, DNA sequence alignment, database, High Throughput Sequencing;**

I. INTRODUCTION

To determine the membership 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, then we conclude that this sequence belongs to a well-defined specie. The comparison between sequences can also allow comparing different species between them. 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 Needleman and Wunsch [2] which determines a global alignment between DNA sequences.

II. STATE OF THE ART

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

On first view, today, there are several databases open access to all publicly available DNA sequences and their protein translations. These banks continue to grow at a positive 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 its more than 154.2 billion base. Nowadays, the quantity of information can reach petabytes in size [4]. In this case, make a treatment on a large number of sequences to infer a sequence belonging to a given species, is very expensive in terms of execution time and resources to allocate. 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.

From another point of view, storage of such alignment is also a problem. Thereafter, any analysis, interpretation or operation of this alignment would be impossible. And if the researcher decides to use a portion of the sequence, no current algorithm allows him to choose, optimally, the length desired to extract from the original chain.

To solve the problems described above and to optimize the use and performance of alignment algorithms and comparison of DNA sequences, a set of research was conducted.

Some researchers have tried to reduce the complexity of dynamic programming algorithms. For example, Yongchao Liu, Douglas L. Maskell, Bertil Schmidt [5] and Yongchao Liu, Bertil Schmidt, Douglas L. Maskell [6] have tried to reduce the total running time of the algorithm of Smith and Waterman exploiting multicore processors architecture Nvidia and their Cuda technology which optimizes the use of these GPUs.

Granger G. Sutton, Owen White, Mark D. Adams and Anthony R. Kerlavage [7] tried also to propose an algorithm that divides the genome into regions while detecting similar regions in order to reduce alignment operations between nucleotides.

Furthermore, recently, the growth of the new DNA sequences alignment technologies has enabled the study of human genome [8, 9]. The size of those genomes reach 3 billion bases. Other species can even reach more than 100 billion bases such us some amphibian species [10]. The use of conventional algorithms for alignment and comparison of DNA sequences is not possible. Indeed the result of an alignment between entire genomes would be an alignment of millions of base pairs, including the time of execution

Manuscript received July 01, 2015; revised July 19, 2015.

Bacem Saada, Ph.D. Student with Harbin Engineering University, College of Computer Science and Technology, Harbin, China, (email:basssoum@gmail.com).

Jing Zhang, Ph.D. Professor with Harbin Engineering University, College of Computer Science and Technology, Harbin, China, (email: zhangjing@hrbeu.edu.cn).

that makes the application of such an operation impossible for usual microcomputers.

The collection, analyzing and understanding this huge quantity of information became a challenge for taxonomic researches. It is used, in some studies, to determinate the organism evolution [8]. In agriculture studies, it is used to study pathogen interactions [11, 12].

This has led to the development of algorithms for DNA compression. Based on the English text compression of the four bases {A, C, G, T}, those algorithms try to reduce the ratio “bits per base” [13, 14, 15].

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

III. NEW APPROACH FOR ALIGNING DNA SEQUENCES

In this section, we propose an approach alignment and comparison of DNA sequences can represent DNA sequences in a chain of their genetic information. To do this, we list in the first instance, the motivations of our approaches. We present approaches while offering a study of complexity theory.

A. *Motivations of the Approach*

To overcome the problems described above, we tried to propose a new approach that attempts to combine the performance of algorithms for alignment and comparison of DNA sequences and to reduce significantly the size of the storage of 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 the algorithm 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 subchain: 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 subchain.
- 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.Abbreviations and Acronyms

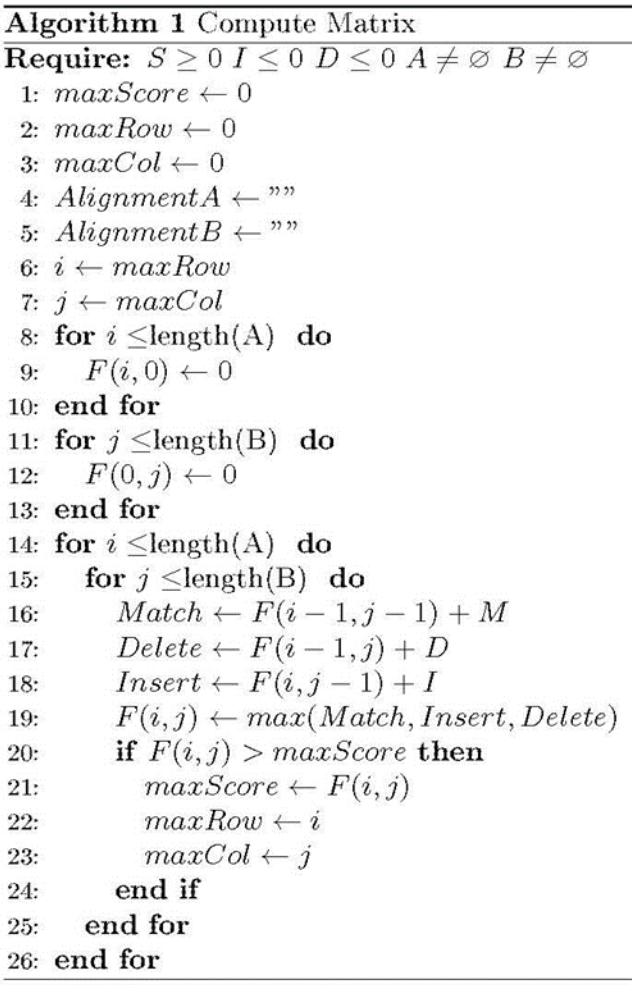
B. *Algorithms*

In this part, we will present the algorithms created for us to solve the problematic of this paper.

a. *Algortihm1: construction of the Matrix*

1. *Presentation of the Algorithm*

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



```
Algorithm 1 Compute Matrix
Require: $S \geq 0$ $I \leq 0$ $D \leq 0$ $A \neq \varnothing$ $B \neq \varnothing$
 1: $maxScore \leftarrow 0$
 2: $maxRow \leftarrow 0$
 3: $maxCol \leftarrow 0$
 4: $AlignmentA \leftarrow$ ""
 5: $AlignmentB \leftarrow$ ""
 6: $i \leftarrow maxRow$
 7: $j \leftarrow maxCol$
 8: for $i \leq$length(A) do
 9:     $F(i,0) \leftarrow 0$
10: end for
11: for $j \leq$length(B) do
12:     $F(0,j) \leftarrow 0$
13: end for
14: for $i \leq$length(A) do
15:     for $j \leq$length(B) do
16:         $Match \leftarrow F(i-1,j-1) + M$
17:         $Delete \leftarrow F(i-1,j) + D$
18:         $Insert \leftarrow F(i,j-1) + I$
19:         $F(i,j) \leftarrow max(Match, Insert, Delete)$
20:         if $F(i,j) > maxScore$ then
21:             $maxScore \leftarrow F(i,j)$
22:             $maxRow \leftarrow i$
23:             $maxCol \leftarrow j$
24:         end if
25:     end for
26: end for
```

The algorithm begins by initializing the variables that will be used in the treatment (Line2-8). Subsequently, the algorithm fills the matrix score while the tests necessary to keep track of the box score with the greatest similarity and its coordinates (Line 15-27).

2. *Complexity of the Algorithm*

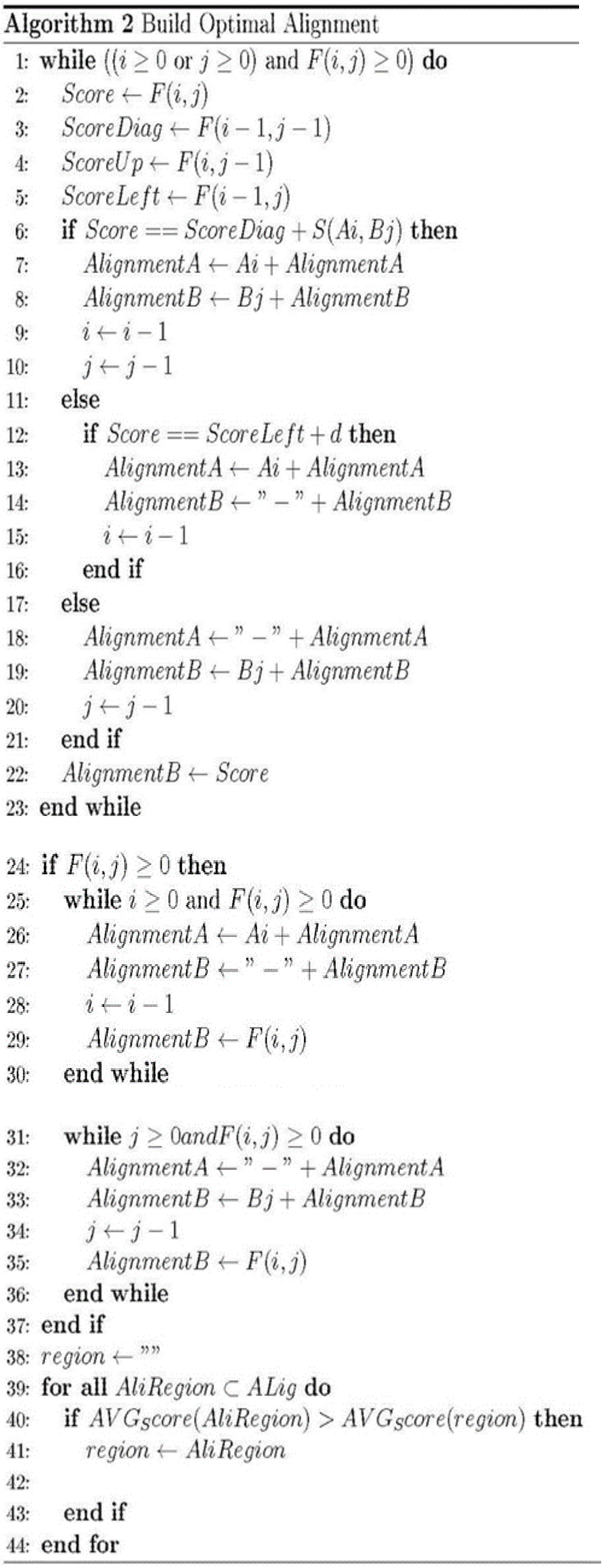
Let two sequences seq1 and seq2. Let l1, l2, l3 the respective lengths of the first sequence, the second sequence and the size of the built alignement and l the length requested by the researcher.

The complexity of the filling phase of the matrix is O(3l1l2). Indeed to fill each cell of the matrix, we perform three arithmetic operations.

b. *Algortihm2: Build Optimal Alignment*

1. *Presentation of the Algorithm*

The algorithm constructs the optimal alignment, then it tries to determine the region with the greater similarity score.



Algorithm 2 Build Optimal Alignment

```
1: while (($i \geq 0$ or $j \geq 0$) and $F(i,j) \geq 0$) do
2:   $Score \leftarrow F(i,j)$
3:   $ScoreDiag \leftarrow F(i-1,j-1)$
4:   $ScoreUp \leftarrow F(i,j-1)$
5:   $ScoreLeft \leftarrow F(i-1,j)$
6:   if $Score == ScoreDiag + S(Ai,Bj)$ then
7:     $AlignmentA \leftarrow Ai + AlignmentA$
8:     $AlignmentB \leftarrow Bj + AlignmentB$
9:     $i \leftarrow i-1$
10:    $j \leftarrow j-1$
11:  else
12:    if $Score == ScoreLeft + d$ then
13:      $AlignmentA \leftarrow Ai + AlignmentA$
14:      $AlignmentB \leftarrow "-" + AlignmentB$
15:      $i \leftarrow i-1$
16:    end if
17:  else
18:    $AlignmentA \leftarrow "-" + AlignmentA$
19:    $AlignmentB \leftarrow Bj + AlignmentB$
20:    $j \leftarrow j-1$
21:  end if
22:  $AlignmentB \leftarrow Score$
23: end while

24: if $F(i,j) \geq 0$ then
25:   while $i \geq 0$ and $F(i,j) \geq 0$ do
26:     $AlignmentA \leftarrow Ai + AlignmentA$
27:     $AlignmentB \leftarrow "-" + AlignmentB$
28:     $i \leftarrow i-1$
29:     $AlignmentB \leftarrow F(i,j)$
30:   end while

31:   while $j \geq 0andF(i,j) \geq 0$ do
32:     $AlignmentA \leftarrow "-" + AlignmentA$
33:     $AlignmentB \leftarrow Bj + AlignmentB$
34:     $j \leftarrow j-1$
35:     $AlignmentB \leftarrow F(i,j)$
36:   end while
37: end if
38: $region \leftarrow ""$
39: for all $AliRegion \subset ALig$ do
40:   if $AVG_Score(AliRegion) > AVG_Score(region)$ then
41:     $region \leftarrow AliRegion$
42:
43:   end if
44: end for
```

The algorithm keeps track of the cell of the matrix that contains the highest similarity score. The construction of the alignment starts from the box (Line1-37). 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 (Line 38-44).

2. *Complexity of the Algorithm*

The complexity of the construction phase of the optimal alignment for the required length is O (3l), this complexity is due to determine the region with the highest similarity score, our algorithm will perform the arithmetic operations, for the calculation of this region's similarity score in (l3-l) regions in total.

The total complexity of this approach is O(3l1l2+3l+[l3-l]l). The complexity of this approach is polynomial of order 2.

IV. Experimental results

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.

A. *Species of the experiments*

a. *Experimental results for species of the same genus*

We analyze 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 will analyze the percentages of similarity of alignment operations between these DNA sequences and infer relations of similarities between species of the same genus.

b. *Experimental results for a set of species of the phylum Firmicutes*

We analyze 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 will determine similarity relationships between the genera in terms of execution time, similarity percentage, and if our approaches can build a hierarchical classification according to the taxonomic classification of species.

c. *Experimental results for random species*

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

B. *Results in term of Percentages of Similarity*

a. *Sequences from same Gender*

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. 1).

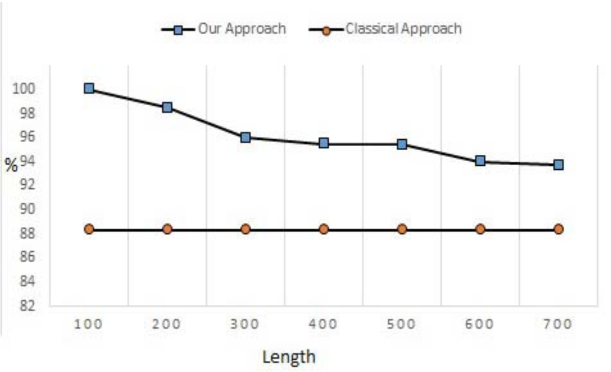


Fig. 1. Experiments on Sequences from same Gender

b. 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. 2).

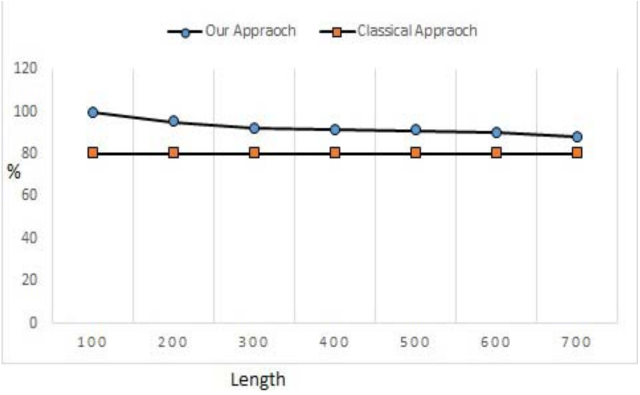


Fig. 2. Experiments on Sequences from same Phylum

c. 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. 3).

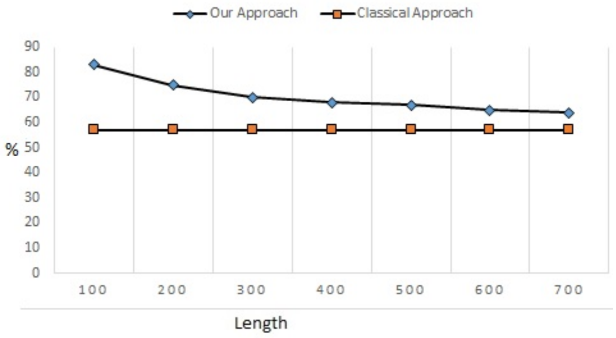


Fig. 3. Experiments on random sequences

C. Experiments in Time execution

Regarding the execution time, we note that the execution time is inversely proportional to the size of the sub chain requested by the researcher. Indeed, when the size of the sub string to extract is small, the number of times of execution of the method for determining the sub chain increases. We also note that for large enough sizes, the execution time does not increase. Execution time remains stable because the phase which consumes too much execution time is the construction phase of the matrix score.

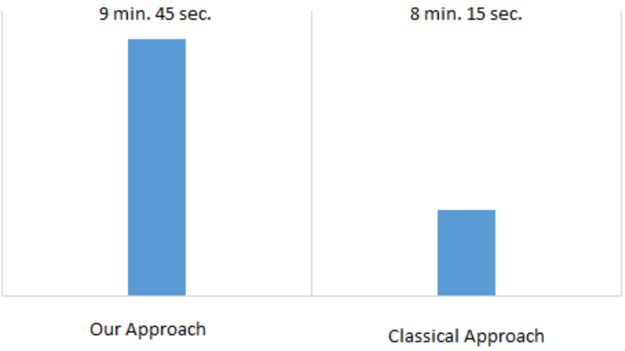


Fig. 4. Execution time of our approach for 4000 constructed alignments

The difference in execution time between the two algorithms is not very important and does not exceed, in the worst case, one minute and 30 seconds for Tests with 4000 constructed alignments. This similarity in execution time favors, at most, the use of our optimal approach (fig. 4).

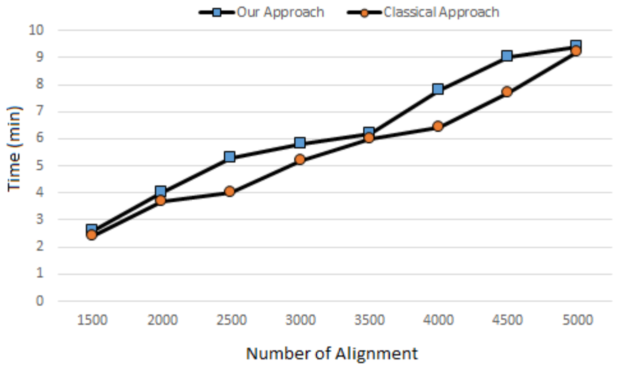


Fig. 5. Execution time of our approach and The Smith & Waterman Algorithm

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. 5).

V. Conclusion and Future Work

Our optimal approach allows for researchers to find a sub string called representative of a given DNA sequence. The percentages of similarity are better than the algorithm of Smith and Waterman, and reach 100% for small lengths. Our approach, then, allow 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 the DNA sequences information and reduce its representation, which will reduce the size of the data in databases.
- **Represent a set on 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 presents a big problem to biologists. We may have a new representation of DNA sequences.

REFERENCES

[1] Needleman, S. B. &Wunsch, C. D. A. general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48, 443–453 (1970).

[2] Smith, T. F. & Waterman, M. S. Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981).

[3] Genbank size, (2013). Available: http://ftp.ncbi.nih.gov/genbank/gbrel.txt

[4] R. Grossi et al., Eds., Reference Sequence Construction for Relative Compression of Genomes, Lecture Notes in Computer Science, Pisa, Italy: Springer, 2011, vol. 7024, 420-425.

[5] Yongchao Liu, Douglas L. Maskell, Bertil Schmidt: CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units". BMC Research Notes, 2009, 2:73

[6] Yongchao Liu, Bertil Schmidt, Douglas L. Maskell: CUDASW++2.0: enhanced Smith-Waterman protein database search on CUDA-enabled GPUs based on SIMT and virtualized SIMD abstractions". BMC Research Notes, 2010, 3:93

[7] GRANGER G. SUTTON, OWEN WHITE, MARK D. ADAMS, and ANTHONY R. KERLAVAGE. Genome Science and Technology. 1995, 1(1): 9-19. doi:10.1089/gst.1995.1.9

[8] Horner, D. S., Pavesi, G., Castrignanò, T., et al. , 2010, Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing, Briefings in Bioinformatics, 11(2), 181–197.

[9] Pushkarev, D., Neff, N. F., and Quake, S. R., 2009, Single-molecule sequencing of an individual human genome, Nature Biotechnology, vol. 27, 847–852.

[10] Korodi, G., Tabus, I., Rissanen, J., et al., 2007, DNA Sequence Compression Based on the normalized maximum likelihood model, Signal Processing Magazine, IEEE, 24(1), 47-53.

[11] Joosen, R. V., Ligterink, W., Hilhorst, H. W., et al., 2009, Advances in genetical genomics of plants, Current Genomics, 10(8), 540–549.

[12] Womack, J. E., 2005, Advances in livestock genomics: opening the barn door, Genome Research, 15(12), 1699–1705.

[13] Matsumoto, T., Sadakane, K., Imai, H., et al., 2000, Can General-Purpose Compression Schemes Really Compress DNA Sequences?, Computational Molecular Biology, Universal Academy Press, 76–77.

[14] Matsumoto, T., Sadakane, K., and Imai, H., 2000, Biological Sequence Compression Algorithms, Genome Informatics, vol. 11, 43–52.

[15] Sato, H., Yoshioka, T., Konagaya, A., et al., 2001, DNA Data Compression in the Post Genome Era, Genome Informatics, vol. 12, 512–514.