

Optimal Path Selection for Sequence Alignment using Hirschberg's based Dynamic Programming Model

N. Senthilkumaran

Department of Computer Science and Applications, The Gandhigram Rural Institute (Deemed to be University), Dindigul, India.

Email: nasenthilkumaran@gmail.com

Abstract—Sequence alignment is the technique of comparing and detecting similarities between biological molecules sequences. Sequence alignment is most beneficial in many ways in bioinformatics programs. Alignments are one of the maximum basic and vital approaches to analyze the similarity among two or more sequences. Hirschberg's algorithm is one of the methods used to align the biological molecules sequences based on dynamic programming approach. The essential application of this algorithm is to go looking most beneficial sequence alignment among two nucleotide or amino acid sequences. The primary intention of this research paper is to observe and evaluate Hirschberg's algorithm in phrases of seek area and time complexity to obtain optimal path selection for sequence alignment with dynamic programming approach in Bioinformatics.

Index Terms— Sequence Alignment, Dynamic Programming, Hirschberg's Algorithm, Optimal Path Selection, Bioinformatics.

I. INTRODUCTION

Sequencing is the method of detecting the precise order of nucleotides of a given DNA sequence. It is used to determine the order of the four bases *adenine (A)*, *guanine (G)*, *cytosine (C)* and *thymine (T)*, in a strand of DNA. DNA sequencing is used to determine the sequence of individual genes, full chromosomes or entire genomes of an organism[1]. Sequence alignment is a strategy of orchestrating sequences of DNA, RNA or protein to find zones of similarities. The closeness in the similarities are being recognized as a viable, auxiliary (or) development connections among the sequences. Which looks for structural and functional relationship between the pair of two biological sequences. Due to the more prominent time it has outperformed and also mutation occurred in the sequence. These changes can make substitutions, addition or cancellations of mutations[2]. Dynamic programming is an algorithmic method commonly used in series evaluation. It is used in sequence comparison, sequence structure prediction etc., It finds possible matches and gaps at all positions in the biological sequences. This dynamic programming approach locates the ideal alignments between the two sequences based on their similarity or distance after assessing the significance of this score[3]. Sequence analysis in bioinformatics refers to the process of subjecting a DNA, RNA or peptide sequence to any of a wide range of analytical methods to analyze its features, function, structure, or evolution[4]. An effective methodologies are required for sequence alignment using the existing databases as well as adding the new sequences to its respective databases, because the collection of sequences in such a database cannot be easily interpreted by the scientists. But, comparing these new sequences to those with known functions is one of the

ways of understanding the biology of an organism from which the new sequence arises. Pairwise sequence alignment in one such methods which compares only the two sequences[5]. Mutations on a sequence overtime may spilt the sequence into some substitutions[6]. Thus, sequence analysis can be used to assign function to genes and proteins by the study of the similarities between the compared sequences. Nowadays, there are many tools and techniques that provide the sequence comparisons and analyze the alignment to understand its biology[7]. This proposed work is to find optimal alignment between the given two sequences based on scoring and similarity and on distance measures.

In this work, the Hirschberg's algorithm based on dynamic programming approach is used for optimal path selection for sequence alignment. The remaining part of the paper is organized as follows, Introduction is given in Section I. Related works are in Section II and Section III describe the methodology, the results and discussion are in section IV. The conclusions are given in section V.

II. RELATED WORKS

The biological sequence database composed of DNA sequences, the existing search algorithm is time-consuming and requires multiple scans of the database. To overcome these disadvantages, in the method[8], a threshold based conditions is used for mining the sequence patterns. The computed degree value for the subsequences are compared with the threshold value to find the sequence patterns. This algorithm found to have minimum computation time and improved sequence patterns. In[9], the author have introduced an effective data structure for storing the correlation matrix for mining the key fragments in DNA sequences. This new approach could handle ultra-long DNA sequences even with the limited storage space. This method is considered as one of the novel approaches for sequence mining. Zhou et al.[10] introduced an algorithm named as mMBioPM for sequence pattern mining by optimizing the hash table structure with pattern division features. This method overcome the problem of data complexity and computation time. A novel implementation based on PWSA algorithm was introduced by Khaled *et al.* [11], they used GPU (General-purpose Graphics Processing Unit) architecture in addition to dynamic programming . They used the above proposed algorithm for sequencing DNA structure. An efficient method to calculated the alignment matrix by using the divide-and-conquer strategy was implemented in [12]. It uses a CUDA-compatible GPUs and divided the entire matrix into several sub matrix and allocated the available threads and memories to each of the sub matrix.

An another parallel approach based on NW algorithm was developed by Chaudhary [13]. They used skewing transformation for traversal and calculation of DP matrix. The comparison between CPU-based and parallel GPU-based implementation for calculating DP matrix was performed in their proposed method.

To achieve an optimal local alignment between the pairwise SW sequence, a novel framework based on CUDA parallel paradigm was introduced in [14]. As the researchers are focusing to use parallel architecture for the computation involved in SW Sequence alignment algorithm, heterogeneous anti-diagonal approach was introduced in [15] to have interaction between the CPU based serial design and GPU based parallel design. Parallelism of SW algorithm based on columns of the similarity matrix is introduced in [16]. When compared to OSEARCH, this proposed method achieves 37 times higher speed, and exhibits the advantages of CPU and GPU architecture. Performance of NW algorithm [17] were increased by using these implementation approach based on the mixture of specialized software and hardware solution. This method produced better outcome on large scale input when compared with the existing algorithms.

Recently, concept of progressive method have been used in many of the new multiple sequence alignment method [18]. In general, the conventional progressive method start by aligning the most closely similar sequences, then it starts aligning the new sequences by increasing the distance measure for similarity checking. A guide tree is used for choosing the new sequence for alignment to build MSA, and a common progressive method was introduced in [19]. GLProbs- Random and GLProbs-Reference [20] are the improved guide tree based progressive alignment method, and the guide tree is randomly generated.

A hidden Markov model for enhancing MSA was implemented in ProbCons [21]. This ProbCons [21] is a MSA tool, which uses Probabilistic and Consistency in obtaining the sequence. Probalign [22] developed a similar approach named as ProbCons as that of Hidden Markov Model (HMM). Another method called ProbPFP, combines PSO along with partition function for progressive alignment [23]. In [24], they introduced, T-Coffee, which is a consistency-based approach for progressive alignment. In T-Coffee, instead of substitution matrix, position-specific scoring is used for progressive alignment. A hybrid approach based on pair hidden Markov model along with partition function is used to find the posterior probabilities for aligning the sequences[25].

III. METHODOLOGY

A. Measurements of Scores

In general, sequence alignment problem uses the scoring function to measures the scores of aligning characters in the sequence at their position and also calculate the total scores on the entire sequence by adding the scores in all positions[26]. Once all possible sequence and its scores have been calculated, it finds the maximum total score for every sequence. After measuring the score, it needs to find the similarity between the given sequences[27]. Most of the existing sequence alignment algorithms mainly focus to get the output alignment by considering the speed of the computation not on the accuracy of the alignment sequence. Scoring process in the prerequisite for the sequence alignment process. It first construct scoring matrix to keep track of the assigned scores for the given pair of two sequences[28]. The matrix is filled as per the match, mismatch and gap value assigned in the sequence. The Figure 1 illustrates the match, mismatch and gap between the pair of sequences. Figure 2, demonstrates the alignment of the given two input sequence after finding match, mismatch and gap.

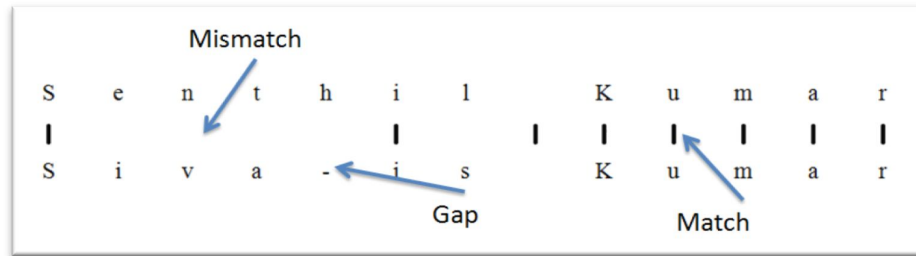


Figure 1. Match, Mismatch and Gap in input sequence

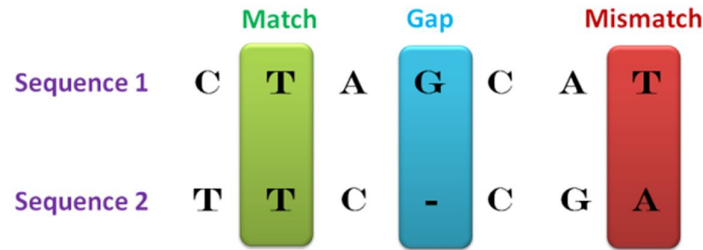


Figure 2. Alignment of Match, Mismatch and Gap

B. Dynamic programming approach for sequence alignment

Dynamic programming approaches are most widely used and acceptable approach for sequence alignment to find all possible alignments between the given two input sequences[29]. In this approach, it considers the entire length of the two sequences and finds the matches and mismatches of the sequences. Let us consider the following two sequence X and Y.

$$X = x_1, x_2, x_3, \dots, x_n \quad (1)$$

$$Y = y_1, y_2, y_3, \dots, y_m \quad (2)$$

The dynamic programming approach finds the length of the sequences X and Y, and insert gaps (or) spaces to X and Y until the length of the both sequence becomes same[30]. The modified sequences are fed into a sequence alignment algorithm to find all possible alignments between the given two input sequences. In general, the sequence alignment may produce an overlap alignment (or) banded alignments[31]. The overlapping alignment do not leave gaps at the beginning and end of the sequence, in such a case, the end of the one sequence will form the beginning for the next sequence. This kind of alignment are referred as free end-gaps alignment, as shown in Figure 3.

On the other hand, banded alignment determines the position where the lower diagonal of the band crosses the vertical axis, and also it specifies the position where the upper diagonal of the band crosses the horizontal axis as illustrated in Figure 4.

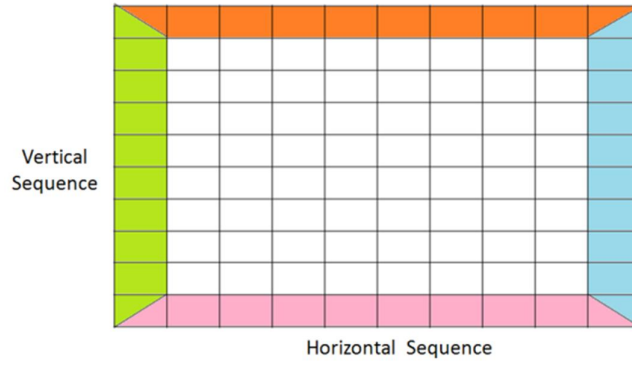


Figure 3. Free end-gaps alignment

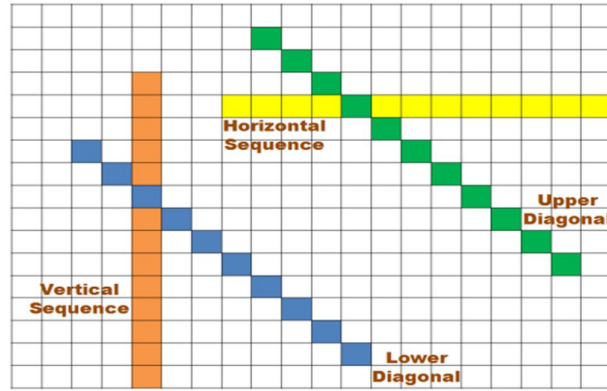


Figure 4. Banded Alignment

In order to find the matrix value for the banded alignment, we have to consider the point where the diagonals crosses the horizontal and vertical axis. Then the value is computed using the following equation[1].

$$f(x) = -x \quad (3)$$

It assigns with positive/negative score based on crossing of diagonal or horizontal and vertical axis. If the lower and upper diagonals are equal, then it uses the hamming distance to find the score.

C. Optimal Path Selection usign Hirschberg's Algorithm

Hirschberg's algorithm is one such algorithms used for sequence alignment. It produces optimal alignment by considering the optimal midpoint using the computed score matrix in the two sub problems[1][32]. Figure 5 Illustrates optimal midpoint selections between the two sub problems.

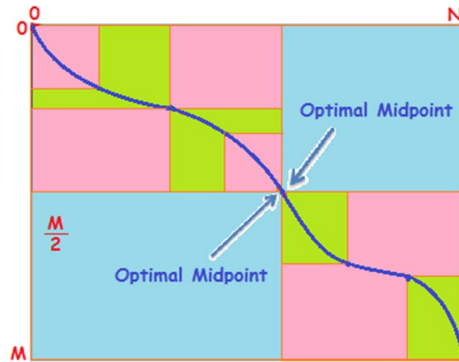


Figure 5. Selection of optimal midpoint by Hirschberg's algorithm

If need arises, the subproblem, may further sub divided into subsubproblem and so on, it can be extended until an optimal path is determined based on the score matrix[9][33]. Often the computation time for this approach is directly proportional to the problem size. After determine all the mid points, it is necessary to find the optimal path sequence for the given input sequence. The following are the summary of the steps involved in Hirschberg's algorithm[1][10] used for finding the optimal path. Here (i,j) and (i',j') denotes the beginning and end of the path respectively .

The computations used to find optimal path in Hirschberg's algorithm:

- $$\begin{aligned}
 & \text{Hirschberg}(i, j, i', j') \\
 & 1. \text{ if } j' - j > 1 \\
 & 2. \quad i^* \leftarrow \arg \max_{0 \leq i \leq m} wt(i) \\
 & 3. \text{ Report } \left(i^*, j + \frac{j' - j}{2} \right) \\
 & 4. \text{ Hirschberg} \left(i, j, i^*, j + \frac{j' - j}{2} \right) \\
 & 5. \text{ Hirschberg} \left(i^*, j + \frac{j' - j}{2}, i', j' \right)
 \end{aligned}$$

In Figure 6, the uncoloured area denotes the aligned pairs on the optimal path from (i₁,j₁) to (i₂,j₂), the colored area indicate the areas where the optimal path is yet to be computed.

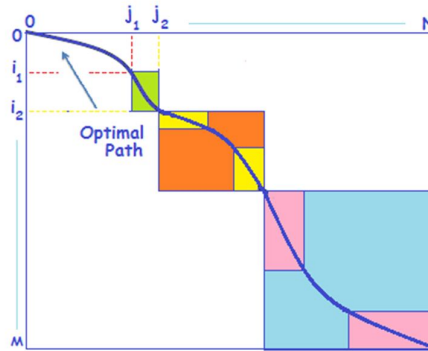


Figure 6. Optimal path selection using Hirschberg's algorithm

IV. RESULTS AND DISCUSSION

This proposed work is experimented with randomly selected input sequences from publicly available GenBank to determine the optimal sequence for the given pair of input sequences. Consider the following input sequences.

Sequence a: AATCG

Sequence b: AACG

Scoring in s: Match -1 Mismatch 1 Gap γ 2

For distance minimization, match scores should be smaller than all other scores.

Recursions:

The computation of recursion for the input sequence is given below.

$$\begin{aligned}
(4) \quad D_{i,j} &= \max \begin{cases} D_{i-1,j-1} + s(a_i, b_j) \\ D_{i-1,j} + \gamma \\ D_{i,j-1} + \gamma \end{cases} = \max \begin{cases} D_{i-1,j-1} + -1 & a_i = b_j \\ D_{i-1,j-1} + 1 & a_i \neq b_j \\ D_{i-1,j} + 2 & b_j = - \\ D_{i,j-1} + 2 & a_i = - \end{cases} \\
(5) \quad D'_{i,j} &= \max \begin{cases} D'_{i+1,j+1} + s(a_{i+1}, b_{j+1}) \\ D'_{i+1,j} + \gamma \\ D'_{i,j+1} + \gamma \end{cases} = \max \begin{cases} D'_{i+1,j+1} + -1 & a_{i+1} = b_{j+1} \\ D'_{i+1,j+1} + 1 & a_{i+1} \neq b_{j+1} \\ D'_{i+1,j} + 2 & b_{j+1} = - \\ D'_{i,j+1} + 2 & a_{i+1} = - \end{cases}
\end{aligned}$$

Output:

The optimal path of the given input sequence is given below,

$$D'n, m = 0, \quad D'n, j = (m - j)\gamma, \quad D'i, m = (n - i)\gamma$$

Visualization of the final traceback path and its optimal alignment are shown in Figure 7 and Figure 8 respectively.

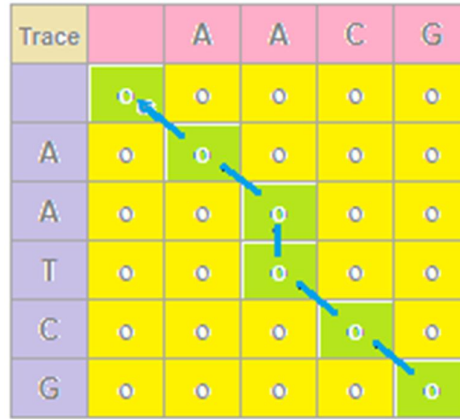


Figure 7. Visualization of the final traceback path



Figure 8. Optimal alignment for traceback

The final alignment for which the traceback was compared is marked as green color in Figure 8. For this given input, the Hirschberg's algorithm computed only one traceback path. The computed minima is marked in Red color. The color Blue and Green are used to mark the start and end of the traceback path. In this way, the Hirschberg's algorithm is effectively used to find optimal path for the given pair of sequences.

V. CONCLUSIONS

Sequence alignment of biological data is extremely useful in the field of computational biology. Alignment of multiple sequence are necessary for a variety of evolution studies. This research work uses Hirschberg's based dynamic programming approach for sequence alignment by computing the score matrix and an optimal sequence path pattern between the given pair of sequence. This method finds, match and mismatch score by minimizing the distance between the given input sequence. This method minimizes the computation time required for the computation of recursion for the randomly selected sequences. The output produced by this research work is accurate and require less computational cost. This research work could be modified with different approach for optimal path selection in future.

REFERENCES

- [1] Hirschberg, D. S., "A linear space algorithm for computing maximal common subsequences", *Comm. ACM*, 18, 341-343, 1975.
- [2] T. Akutsu, "Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots", *Discrete Applied Mathematics*, 104:45-62, 2000.
- [3] Gotoh, O, "Alignment of three biological sequences with an efficient traceback procedure", *J. Theor. Biol.*, 121, 327-337, 1986.
- [4] Talal Bonny, K.N. Salama, "Fast Global Sequence Alignment Technique", *Conference Record of the Forty Fifth Asilomar Conference on Signal, Systems and Computers (ASILOMAR)*, Nov 2011.
- [5] P.V.S.L. Jagadamba, M.S.P. Babu, A.A. Rao, P.K.S. Rao, "An Improved Algorithm for Multiple Sequence Alignment using Particle Swarm Optimization", *IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*, July 2011.
- [6] E.W. Myers and W. Miller, "Optimal Alignments in Linear Space," *Computer Applications in the Biosciences*, vol. 4, no. 1, pp. 11-17, 1988.
- [7] H. Yousefi, M. Ahmadi, H.R. Roshani, "Fast Sequence Alignment Algorithm Using BloomFilters", *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, May 2012.
- [8] Junyan, Z., and Chenhui, Y. (2015). "Sequence pattern mining based on markov chain," in *Proceedings of the 2015 7th International Conference on Information Technology in Medicine and Education (ITME)* (Piscataway, NJ: IEEE), 234–238. doi: 10.1109/ITME.2015.49
- [9] Mao, G. (2019). "Association matrix method and its applications in mining DNA sequences," in *Proceedings of the International Conference on Applied Human Factors and Ergonomics* (Piscataway, NJ: IEEE), 154–159. doi: 10.1007/978-3-030-20454-9_15
- [10] Zhou, Q., Jiang, Q., and Li, S. (2010). "An efficient algorithm for protein sequence pattern mining," in *Proceedings of the 2010 5th International Conference on Computer Science & Education* (Piscataway, NJ: IEEE), 1876–1881. doi: 10.1109/ICCSE.2010.5593815
- [11] H. Khaled, R. El Gohary, N. L. Badr, and H. M. Faheem, "Accelerating pairwise DNA sequence alignment using the CUDA compatible GPU," *Int. J. Comput. Appl.*, vol. 84, no. 1, pp. 25_31, Dec. 2013.
- [12] C. Ling, K. Benkrid, and T. Hamada, "A parameterisable and scalable Smith_Waterman algorithm implementation on CUDA-compatible GPUs," in *Proc. IEEE 7th Symp. Appl. Speci_c Processors*, Jul. 2009, pp. 94_100.
- [13] A. Chaudhary, D. Kagathara, and V. Patel, "A GPU based implementation of Needleman_Wunsch algorithm using skewing transformation," in *Proc.8th Int. Conf. Contemp. Comput. (IC)*, Aug. 2015, pp. 498_502.
- [14] H. Khaled, G. R. El, N. L. Badr, and H. M. Faheem, "Hybrid framework for pairwise DNA sequence alignment using the CUDA compatible GPU," in *Proc. Int. Conf. Bioinf. Comput. Biol. (BIOCOMP)*, 2013, p. 1.
- [15] Y. S. Lee, Y. S. Kim, and R. L. Uy, "Serial and parallel implementation of Needleman_Wunsch algorithm," *Int. J. Adv. Intell. Inform.*, vol. 6, no. 1, pp. 97_108, 2020.
- [16] B. Chen, Y. Xu, J. Yang, and H. Jiang, "A new parallel method of Smith_Waterman algorithm on a heterogeneous platform," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, in *Lecture Notes in Computer Science*, vol. 6081. Berlin, Germany: Springer-Verlag, 2010, pp. 79_90.
- [17] Y. Jararweh, M. Al-Ayyoub, M. Fakirah, L. Alawneh, and B. B. Gupta, "Improving the performance of the Needleman_Wunsch algorithm using parallelization and vectorization techniques," *Multimedia Tools Appl.*, vol. 78, no. 4, pp. 3961_3977, Feb. 2019.
- [18] Chatzou M, Magis C, Chang JM, Kemena C, Bussotti G, Erb I, Notredame C. 2016 *Multiple sequence alignment modeling: methods and applications*. Briefings in bioinformatics. 17(6):1009-23.
- [19] Thompson JD, Higgins DG, Gibson TJ. 1994 *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*. Nucleic acids research. 22(22):4673-80.
- [20] Zhan Q, Ye Y, Lam TW, Yiu SM, Wang Y, Ting HF. 2015 *Improving multiple sequence alignment by using better guide trees*. In *BMC bioinformatics* (Vol. 16, No. 5, pp. 1-6). BioMed Central.

- [21] Do CB, Mahabhashyam MS, Brudno M, Batzoglou S. 2005 *ProbCons: Probabilistic consistency-based multiple sequence alignment*. *Genome research*. 15(2):330-40.
- [22] Roshan U, Livesay DR. 2006 *Proalign: multiple sequence alignment using partition function posterior probabilities*. *Bioinformatics*. 22(22):2715-21.
- [23] Zhan Q, Wang N, Jin S, Tan R, Jiang Q, Wang Y. 2019 *ProbPFP: a multiple sequence alignment algorithm combining hidden Markov model optimized by particle swarm optimization with partition function*. *BMC bioinformatics*. 20(18):1-0.
- [24] Notredame C, Higgins DG, Heringa J. 2000 *T-Coffee: A novel method for fast and accurate multiple sequence alignment*. *Journal of molecular biology*. 302(1):205-17.
- [25] Liu Y, Schmidt B, Maskell DL. 2010 *MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities*. *Bioinformatics*. 26(16):1958-64.
- [26] Da Li, M. Becchi, "Multiple Pairwise Sequences Alignments with NeedlemanWunsch Algorithm on GPU", SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC), Nov 2012.
- [27] Li H, Homer N, "A Survey of Sequence Alignment Algorithms for Next-generation Sequencing", *Briefings in Bioinformatics*. 2010;11(5): 473–83.
- [28] Hasan L, Zafar H, "Performance Versus Power Analysis for Bioinformatics Sequence Alignment", *J Appl Res Technol*. 10(6):920–8, 2012.
- [29] O. Gotoh, "An Improved Algorithm for Matching Biological Sequences," *J. Molecular Biology*, vol. 162, no. 3, pp. 705-708, Dec. 1982.
- [30] Needleman, S. B. and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequences of two proteins", *J. Mol. Biol.*, 48, 443-453, 1970.
- [31] Waterman, M. S., T. F. Smith and W. A. Beyer, "Some biological sequence metrics", *Adv. Math.* 20, 367-387, 1976.
- [32] Waterman, M. S. and M. Eggert, "A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons", *J. Mol. Biol.* 197, 723-728, 1987.