Performance improving extensions of the Smith-Waterman Algorithm

Carsten Tanke, Christian Rohrandt and Ulrich Jetzek
Kiel University of Applied Sciences
Kiel, Germany

Abstract— A key element in the field of bioinformatics is the alignment of two sequences. Alignment in this case means the comparison of the mentioned two sequences using a certain metric to evaluate the similarity of the sequences. A well-known und still commonly used method is the Smith-Waterman-Algorithm, which is based upon a two-step alignment procedure: Both sequences form the two axes of a n x m-matrix, where n and m are the respective sequence lengths. In the first step the matrix elements are calculated in an iterative fashion. In the second step, the so called back tracking the best alignment path within the matrix is being detected.

If one assumes the two sequences to be of the same length, the computational effort of the Smith-Waterman-Algorithm grows with O (n^2). Hence, if e.g. the sequence length n=1024, more than one million matrix elements need to be calculated.

Therefore it would obviously be advantageous to look for methods, where the computational effort for the alignment can be significantly reduced, while at the same time the alignment quality is kept at the same or almost the same level. Since the optimal alignment path usually orientates rather close to the main diagonal of the matrix, the idea is to work with submatrices of length $k < n$.

A corresponding extension of the Smith-Waterman-Algorithm has been studied intensively. The results are being presented in this paper. Two versions with different arrangements of submatrices were studied. The first version arranges the submatrices along the main diagonal of the matrix while the second version orientates them along the respective maximum of the former submatrix.

To ensure no or only minimal loss in quality of an alignment, the dimensions for the submatrices and the overlap between two submatrices were researched with statistical methods. For all three versions of the algorithm the calculation times as well as the resulting alignments were compared.

For two sequences with the lengths of 1024 the number of calculated cells could be reduced from 1 million in the original algorithm to roughly half a million in the versions with submatrices. This causes also a reduced calculation time of about 50 %. The quality impact is 1.2 % at a length of 1000, but decreases significantly with growing lengths.

***Index Terms*—Sequence Alignment, Smith-Waterman Algorithm, submatrices, calculation costs.**

I. Introduction

Mathematic methods play a significant role in analysis of DNA- and protein sequences. A major topic is to find similarities between two sequences and a tool for that is the sequence alignment.

Nowadays many different algorithms to align two sequences exist. All of these have their advantages as well as disadvantages. The Smith-Waterman algorithm (SW) is widely distributed and finds a local alignment for two sequences. One disadvantage is that the algorithm is very CPU-intensive, because for two sequences with the lengths m and n a matrix of m+1 by n+1 is needed. At least there are m*n cells to calculate and the computation space increase with the order O (n^2).

II. THE SMITH-WATERMAN ALGORITHM

In 1970 Saul B. Needleman and Christian D. Wunsch developed a method to find similarities between two sequences [1]. They calculated a matrix and found an alignment path from the lower right corner back to the upper left corner which results in a global alignment.

In 1981 Temple F. Smith and Michael S. Waterman modified this algorithm in such a way that the result can also be a local alignment [2]. The algorithm compares two DNA – sequences, calculates a similarity score and makes an alignment with the two sequences. The algorithm can be divided in two phases: matrix calculation and backtracking.

The locally optimal alignment path starts at the cell with the maximum value and ends when a cell with the value zero or a predefined threshold is reached.

A. Matrix calculation

For two sequences with the lengths m and n a matrix of m+1 by n+1 has to be created. The first row and the first column are initialized with zeros. All other cells are calculated with equations 1 and 2 where Eq.1 serves as a weight function to separate matches and mismatches in the two sequences with indices a and b. The value for a gap penalty is usually -1.

$$\text{w}=\begin{cases}+2\ if\ Seq1(a)=Seq2(b)\\-1\ if\ Seq1(a)\neq Seq2(b)\end{cases}\qquad(1)$$

$$F(i,j) = max \begin{cases} F(i-1,j-1)+w \\ F(i-1,j)+gap \\ F(i,j-1)+gap \\ 0 \end{cases} \qquad (2)$$

B. Backtracking

The starting point for the backtracking is the cell with the maximum value. From this cell an alignment path (top, top-left or left) through the matrix has to be followed until a cell with the value zero or a predefined threshold is reached. This path results in the alignment with direct assignments, insertions or deletions. More than one maximum in the matrix leads to several alignments that can differ in length and quality.

III. USE OF SUBMATRICES

Since the alignment path in the original Smith-Waterman algorithm usually is orientated rather close to the main diagonal of the matrix, many cells in the upper right and lower left part are calculated but have no effect on the alignment path. Therefore the idea is to use submatrices which are arranged in the original matrix. The amount of calculated cells can be reduced and so the calculation time decreases, too. Two versions with different arrangements of submatrices were tested. The first version is shown in Figure 1 and orientates the submatrices along the main diagonal while the second version orientates them along the respective maximum of the former submatrix (see Figure 2). With MATLAB three functions were programmed to realize the original matrix calculation as well as the two versions with submatrices. For the submatrices two parameters are important for the results of the alignment: the dimensions of the submatrices and the overlap between two consecutive submatrices. To find the best dimensions a statistical study was executed. For random sequences of A, G, C and T (representing the four bases adenine, guanine, cytosine and thymine of a DNA-sequence) the cell indices with the maxima were recorded and counted. Different dimensions of submatrices lead to different coverage of the recorded maxima, which is represented in Table 1. By using submatrices with a dimension greater than one third of the original matrix all maxima from the statistic research were covered. An overlap of one fourth leads to a corridor along the main diagonal that is wide enough for the backtracking path.

Table 1 : Dimensions of submatrices and resulting coverage of maxima

Dimension of submatrices (% of original matrix)	Coverage of maxima
20 %	95.4 %
25 %	98.6 %
30 %	99.7 %
35 %	100 %

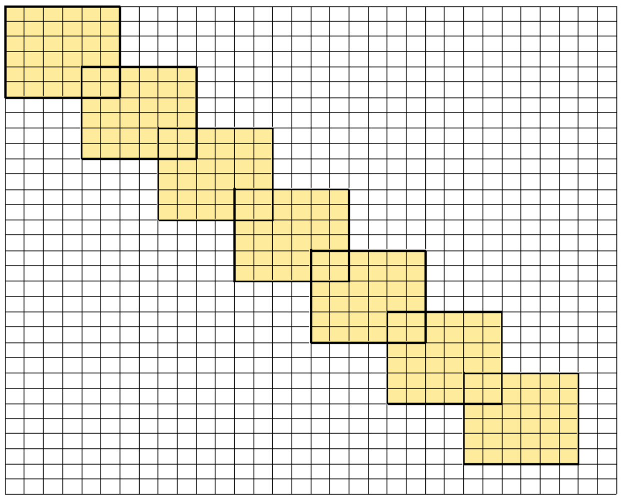

Figure 1 : Use of submatrices along main diagonal

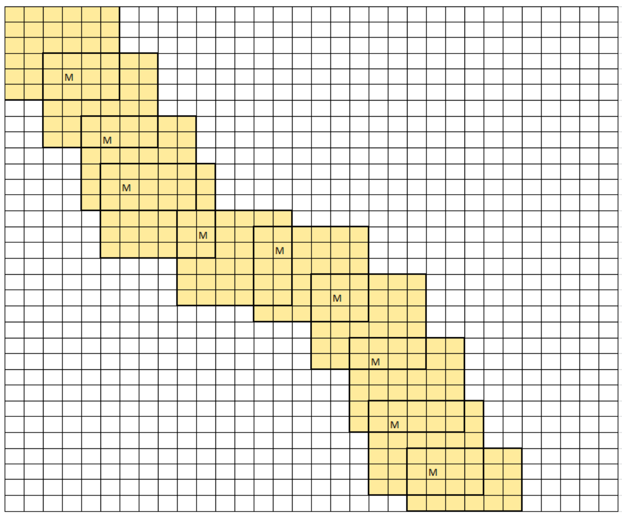

Figure 2 : Use of submatrices along maximum of former submatrix

IV. QUALITY OF ALIGNMENT

For the original Smith-Waterman algorithm and the two versions with submatrices statistical data was produced to compare the quality of resulting alignments. 1000 randomly generated sequence pairs were produced for different sequence lengths to test the algorithms. As a quality index the relation between matches and length of alignment was used. The original Smith-Waterman algorithm produced the highest values for the tested sequences. The algorithm with submatrices along the main diagonal provided nearly identical results for sequences longer than 3000 bases. The last version, with the submatrices along the maxima of the former submatrix, produced worse quality. For sequences of 1000 the quality was about 1.5 % below the original algorithm. With increasing lengths of sequences the quality increases. For a length of 17000 the difference was only 0.33 %. The resulting qualities of alignment are shown in Table 2.

Table 2: Quality index of alignment for the three SW versions

Sequence length	Original SW	SW with submatrices along main diagonal	SW with submatrices along maxima
1000	0.3355	0.3352	0.3314
2000	0.3350	0.3348	0.3319
3000	0.3350	0.3348	0.3325
4000	0.3351	0.3350	0.3331
5000	0.3350	0.3350	0.3331
6000	0.3349	0.3348	0.3331
7000	0.3348	0.3348	0.3334
8000	0.3350	0.3349	0.3336
9000	0.3352	0.3351	0.3337
10000	0.3353	0.3353	0.3340
11000	0.3352	0.3352	0.3341
12000	0.3351	0.3351	0.3340
13000	0.3353	0.3352	0.3342
14000	0.3354	0.3354	0.3343
15000	0.3351	0.3351	0.3341
16000	0.3353	0.3353	0.3344
17000	0.3355	0.3355	0.3345

V. COMPUTATIONAL COSTS

For the three versions of the algorithm statistical data was produced to compare the calculation times for the algorithms. Again 1000 randomly generated sequence pairs were produced to time the algorithms. The original Smith-Waterman algorithm calculates all cells of the matrix and is therefore the slowest one. The two versions with submatrices do not calculate all cells and save computational costs. For different length of sequences the calculation times are listed in Table 3. While the calculation time for the original Smith-Waterman algorithm increases as an exponential function the two versions with submatrices only increase as a power function with the order 2 (see Figure 3).

Table 3: Calculation times (sec) for the three SW versions

Sequence length	Original SW	SW with submatrices along main diagonal	SW with submatrices along maxima
1000	0.0779	0.0393	0.0544
2000	0.3286	0.1457	0.2039
3000	0.7306	0.3215	0.4482
4000	1.2969	0.6132	0.8234
5000	2.0434	0.9478	1.2755
6000	2.9386	1.3681	1.8339
7000	4.0635	1.8643	2.5112
8000	5.3295	2.4253	3.2593
9000	7.3260	3.1127	4.1767
10000	9.3730	3.8827	5.2118
11000	12.1161	4.7336	6.3517
12000	18.3522	5.5615	7.4443
13000	20.4468	6.5436	8.7319
14000	25.9288	7.5970	10.1192
15000	39.3383	8.7240	11.5664
16000	81.4771	10.0552	13.3807
17000	98.7519	11.3490	15.1290

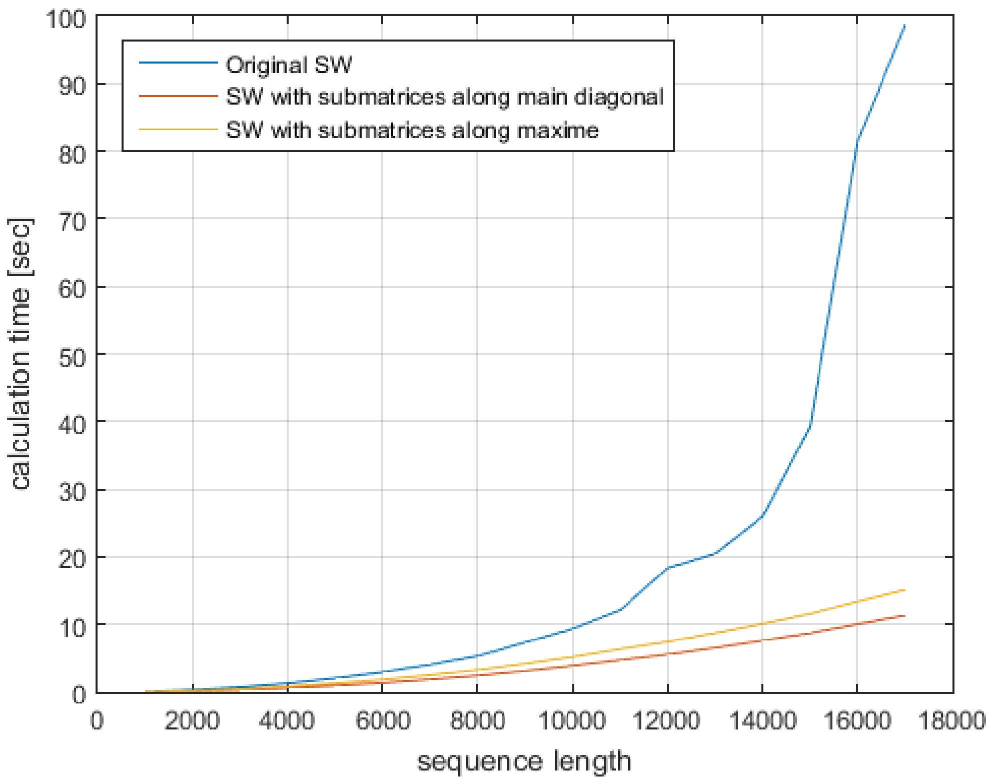

Figure 3: Calculation times for the different SW versions

VI. RESULTS

For sequences with lengths below 3000 the quality of the original Smith-Waterman algorithm is the best. For longer sequences the algorithm with submatrices along the main diagonal has nearly identical quality indices while the version with submatrices along the maximum of the former submatrix has indices slightly below the others.

The computation of submatrices instead of the whole matrix leads to a significant reduction of calculation time. It could be reduced from an exponential function to a power function with the order 2 by use of submatrices. In combination with the results of the quality indices the algorithm with submatrices along the main diagonal is an efficient alternative to the original Smith-Waterman algorithm. It should be mentioned that the sequences for this research were created randomly with a distribution of bases as in the human genome but do not reflect real DNA-sequences from nature.

REFERENCES

[1] S. Needleman and C. Wunsch, “A general method applicable to the search of similarities in the amino acid sequence of two proteins” Journal of Molecular Biology, vol. 48, pp. 443–453, 1970.

[2] T. Smith and M. Waterman, “Identification of common molecular subsequences” Journal of Molecular Biology, vol. 147, pp. 195–197, 1981.

[3] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, “Biological sequence analysis-Probabilistic models of proteins and nucleic acids”, Cambridge: University Press, 1998, pp.17–24.

[4] A. Lesk, “Introduction to bioinformatics”, Oxford: University Press, 2005, pp. 175–182.

[5] M. Hütt, M. Dehnert, “Methoden der Bioinformatik”, Heidelberg: Springer, 2016, pp. 172-181 .

[6] A. Hosny, H. Shedeed, A. Hussein, M. Tolba, “An efficient solution for aligning huge DNA sequences”, International Journal of Computer Applications, vol. 32, 2011.

.