

CHAPTER IV

IMPLEMENTATION OF DIALIGN ALGORITHM

In this chapter we present the program implementation and simulation of DIALIGN algorithm in producing optimal alignments from a pair of DNA sequences. Program implementation of DIALIGN algorithm to find optimal sequence alignment will be discussed in section 4.1 and the program simulation will be discussed in section 4.2.

4.1 IMPLEMENTATION

The program is implemented in MATLAB 7.0.1. Some functions used in this program are dialign, backtrack, and dispalign (complete listing of the program see Appendix 1). Dialign is the main function of DIALIGN algorithm while backtrack and dispalign functions will be called in this dialign function. Backtrack function is utilized to trace all alignment paths which compose optimal alignments. Dispalign function is useful to find which residues that are to be aligned to gaps and to display the optimal alignments.

As an input, DIALIGN requires two sequences. Since we discuss about DNA sequence alignment, the input must be a string of DNA alphabet (C, G, T, A) and contains no spaces. The sequence input is saved in a text file

whose names could be entered by user. This file saves two DNA sequences. The program will produce optimal alignments as an output.

The program is called by typing “dialign” in MATLAB’s command window. When it is run, user will be asked to type the path of the input file that already saved the DNA sequences. The program will stop if it obtains all possible optimal alignments and display these alignments in MATLAB’s command window.

Having the DNA strings, the program will construct possible diagonals from those strings and calculate weights of these diagonals based on match probabilities. After that, the program computes a matrice of alignment scores. Every number of matches, $\sigma(D_{i,j})$ and scores is saved in a cell which later will be utilized in backtracking process. The process of backtracking is called recursively in a backtrack function and saves the result in a cell. Next, the program calls dispalign function to display the alignments.

This program traces all alignment paths and resulting all optimal alignments. Here is an example of using the program to find alignment from example 3.1.

Example 4.1

We use sequences in example 3.1, it is a sequence with length of three which is going to be compared to a sequence with length of two. We type dialign in MATLAB’s command window and press enter, then it will appear a window-like as in Figure 4.1. In this example we use sequences

from example 3.1 which is saved in C:\Users\User\Documents\skripsi quw\dialign\sequen.txt. So, when the program asks user to type the path of input file we paste C:\Users\User\Documents\skripsi quw\dialign\sequen.txt in MATLAB's command window and press enter, then it will process the string input. Sequences that already saved in sequen.txt is recognized as two strings. The program is provided in appendix one.

In Figure 4.1, it can be seen that sequences which are to be compared are CTG and CG. Figure 4.2 shows us the input display. In Figure 4.3, we see that the program is able to align a pair of sequence from example 3.1 in 3.1×10^{-2} seconds and the only one optimal sequence alignment is exactly identical to the result in example 3.4.

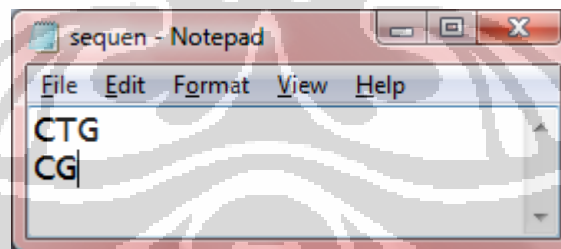


Figure 4.1 Content of sequen.txt

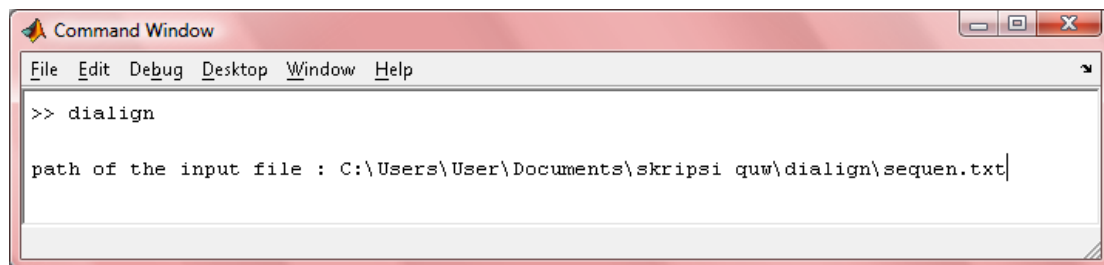
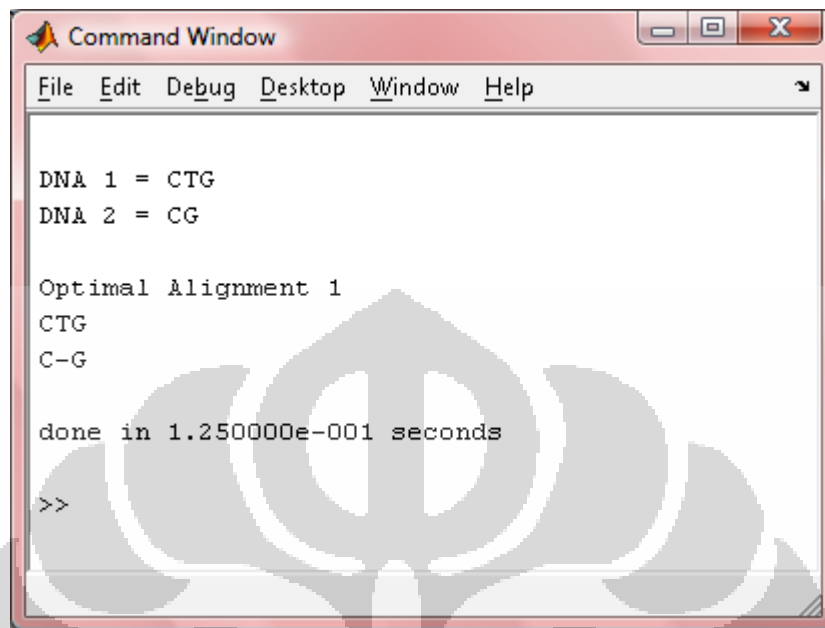


Figure 4.2 Input display for example 3.1



```
Command Window
File Edit Debug Desktop Window Help
DNA 1 = CTG
DNA 2 = CG
Optimal Alignment 1
CTG
C-G
done in 1.250000e-001 seconds
>>
```

Figure 4.3 Output display for example 3.1

The program is succeeded in finding optimal alignment from a pair of sequence in example 3.1. Next section will discuss about simulation of the program of another sequences.

4.2 SIMULATION

As experiments, we try seven samples taken at random with length of a sequence not more than 100. The resulted optimal alignment of sample 1 up to sample 3 will be discussed, the remaining samples can be seen in Appendix 2. The number of optimal alignments and running time of each samples is given in Table 4.1.



```

Command Window
File Edit Debug Desktop Window Help
DNA 1 = AGGTCGTGTGCGATATGAACTTATGAGCAAAAATATCGTAAACACATTAGACAACAATCGAAAT
DNA 2 = TATGTCCACATAACGTGACAACGATT

Optimal Alignment 1
AGGTCGTGTGCGATATGAACTTATGAGCAAAAATATCGTAAACACATTAGACAACAATCGAAAT
---TA-TGTCCC-----AC--AT-A--A-----CGT-----GACAACGAT-----T

Optimal Alignment 2
AGGTCGTGTGCGATATGAACTTATGAGCAAAAATATCGTAAACACATTAGACAACAATCGAAAT
---TA-TGTCCCA-----C--AT-A--A-----CGT-----GACAACGAT-----T

done in 2.844000e+000 seconds
>>

```

Figure 4.4 Output display for sequen1

The first experiment (sequen1) is sequences with length of 62 and 27. From these sequences, we get two optimal alignments in 2.884 seconds. Figure 4.4 shows us the resulted optimal alignments for sequences in sequen1. We've already discussed about the sequence alignment properties in chapter two and alignments in figure 4.4 satisfy the two properties stated by [ACL⁺ 02].

Sequen2 consists of sequences with length of 45 and 63. In this second experiment, we obtain six optimal alignments in 155.8 seconds. The program takes much more time to solve alignment problem in sequen2 than in sequen1 because of the number of constructed diagonals which need much time in calculating weights and scores. Figure 4.5 shows us the output for sequences in sequen2.

```

Command Window
File Edit Debug Desktop Window Help

DNA 1 = AGACCTCGTCGATCTCTAAGATCACAAATGGCCTTCTAGGCCGTA
DNA 2 = CACTGTACCCTACTACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGGCTTGACGAGGA

Optimal Alignment 1
-A--G-AC-CT-CGT-CGA---TCTC----TAA-GATCA--C--A--AA-TGGCCTT--CTAGGCCGTA
CACTGTACCCTAC-TACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGGC-TTGACGAGG----A

Optimal Alignment 2
-A--G-A-CCT-CGT-CGA---TCTC----TAA-GATCA--C--A--AA-TGGCCTT--CTAGGCCGTA
CACTGTACCCTAC-TACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGGC-TTGACGAGG----A

Optimal Alignment 3
-A--G-ACC-T-CGT-CGA---TCTC----TAA-GATCA--C--A--AA-TGGCCTT--CTAGGCCGTA
CACTGTACCCTAC-TACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGGC-TTGACGAGG----A

Optimal Alignment 4
-A--G-AC-CT-CGT-CGA---TCTC----TAA-GATCA--C--A--AA-TGGCCTT--CTAGGCCGTA
CACTGTACCCTAC-TACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGG-CTTGACGAGG----A

Optimal Alignment 5
-A--G-A-CCT-CGT-CGA---TCTC----TAA-GATCA--C--A--AA-TGGCCTT--CTAGGCCGTA
CACTGTACCCTAC-TACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGG-CTTGACGAGG----A

Optimal Alignment 6
-A--G-ACC-T-CGT-CGA---TCTC----TAA-GATCA--C--A--AA-TGGCCTT--CTAGGCCGTA
CACTGTACCCTAC-TACAAAAGTCTTAGAATAATGATCAGTCGGATTAACCTGG-CTTGACGAGG----A

done in 1.557960e+002 seconds

>> |

```

Figure 4.5 Output display for sequen2

In the third experiment, we compare a sequence with length of 88 to a sequence with length of 34. It produces twelve optimal alignments in 17.953 seconds. In Figure 4.6, we see the resulted optimal alignments for sequences in sequen3. The output display for sequen4 up to sequen7 is provided in Appendix 2.

```

Command Window
File Edit Debug Desktop Window Help

DNA 1 = GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGTGCCCCGGCTGCGGTTGTATCCTGAATACGCC
DNA 2 = ATGCGCCAGTGGACTGCGTAGACCGCTTCATCAT

Optimal Alignment 1
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A-----CT-GC--GTAGACCG-CTTC-----ATC---AT-----

Optimal Alignment 2
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACCG-CTTC-----ATC---AT-----

Optimal Alignment 3
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACCG-CTTC-----ATC---AT-----

Optimal Alignment 4
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A-----CT-GC--GTAGACC-GCTTC-----ATC---AT-----

Optimal Alignment 5
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACC-GCTTC-----ATC---AT-----

Optimal Alignment 6
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACC-GCTTC-----ATC---AT-----

Optimal Alignment 7
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A-----CT-GC--GTAGACCG-CTTC-----ATC---A-T-----

Optimal Alignment 8
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACCG-CTTC-----ATC---A-T-----

Optimal Alignment 9
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACCG-CTTC-----ATC---A-T-----

Optimal Alignment 10
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A-----CT-GC--GTAGACC-GCTTC-----ATC---A-T-----

Optimal Alignment 11
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACC-GCTTC-----ATC---A-T-----

Optimal Alignment 12
GTTGACTTCTACTAAAAGCAAGCTCCTGAGTAGCTGGCCAAGCGAGCTTGCTTGT-GCCCCGGCTGCGGTTGTATCCTGAATACGCC
-----A-T-----GC--GC-C---AGT-G--G---A--C---T-GC--GTAGACC-GCTTC-----ATC---A-T-----

done in 1.795300e+001 seconds

>>

```

Figure 4.6 Output display for sequen3

Table 4.1 stores seven samples of sequences taken at random. In Table 4.1, first column corresponds to the input file names which consist of two sequences at each, second column represents each length of the sequences, next column is a column of the number of optimal alignments obtained, and the last column exhibits running time of the program's performance.

Table 4.1 The number of optimal alignment and running time of the seven samples taken at random

Input File	Length		Number of Alignment	Running Time
sequen1	1	62	2	2.884
	2	27		
sequen2	1	45	6	1.558×10^2
	2	63		
sequen3	1	88	12	17.953
	2	34		
sequen4	1	20	24	7.5×10^{-1}
	2	50		
sequen5	1	24	24	1.094
	2	65		
sequen6	1	92	24	2.672
	2	31		
sequen7	1	62	64	5.676×10^2
	2	79		

From Table 4.1, it can be seen that the program performs excellent on short sequences. From those seven samples, the optimal alignments obtained are reasonable and it can be done in a short time. There is no pattern formed from the number of optimal alignment and running time of the

program. According to Table 4.1, we see that the number of alignment does not depend on the length of sequences but depends on the content of sequences whereas running time of the program is effected by the length of sequences and the number of optimal alignment.

