A Heuristic Managing Errors for DNA Sequencing*

Jacek Błażewicz[†‡] Piotr Formanowicz[†] Frederic Guinand[§]
Marta Kasprzak[†¶]

Motivation: A new heuristic algorithm for solving DNA sequencing by hybridization problem with positive and negative errors.
Results: A heuristic algorithm providing better solutions than algorithms known from the literature based on tabu search method.
Contact: blazewic@sol.put.poznan.pl.

1 Introduction

One of the primary problems arising in molecular biology consists in determining the bases of a DNA sequence. The DNA sequencing consists in determining a sequence of nucleotides of an examined DNA fragment, cut out from a genome by restriction enzymes or by the shotgun approach. There exist two approaches to sequencing: the chemical one proposed by Maxam and Gilbert which did not stand the test of time [MG77] and the one involving gel electrophoresis by Sanger [SC78] used in bio-labs. The new approach *sequencing by hybridization* (SBH) offers an interesting alternative [Wat95, AG97, SM97, VLM97]. The method is already widely used for SNP analysis and its general usage for sequencing purpose depends mainly on the development of good algorithmic procedures solving the computational phase of this approach. The current paper is devoted to the presentation of the new algorithm which compares favorably with the other algorithms available in the literature.

A DNA fragment is usually written as a sequence of letters A, C, G, and T, representing four nucleotides composing the fragment. i.e. adenine, cytosine, guanine, and thymine, respectively. A short sequence of nucleotides is named an *oligonucleotide*. The aim of the *hybridization experiment* (being the first stage of the SBH approach) [BS88, LYF+88, Sou88, DLB+89, MAR+94] is to detect all oligonucleotides of a given length l (usually 8-12 bases) composing an examined DNA fragment of a known length n (a few hundreds of bases). For this purpose the *oligonucleotide library* is generated, which consists of all

*Partially supported by KBN and Egide grant #3274RH.

†Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland and Institute of Bioorganic Chemistry, Polish Academy of Sciences, 61-704 Poznań, Poland *blazewic@put.poznan.pl {marta,piotr}@cs.put.poznan.pl.*

‡To whom correspondence should be addressed.

§LIH - Le Havre University, BP 540 76058 Le Havre cedex, France *Frederic.Guinand@univ-lehavre.fr.*

¶Fellowship holder of the Foundation for Polish Science.

possible (i.e. 4^l) single-stranded DNA fragments of length l. Next, the library is compared (in the sense of hybridization) with the examined DNA fragment. In order to operate on that great number of molecules, the advanced technology of *microarray chips* [Sou88, FRP$^+$91, PSS$^+$94] has been developed.

After the chip has been generated, it is introduced into an environment with precisely defined physical parameters (e.g. temperature), together with many copies of the examined DNA fragment, labeled with a fluorescent marker. During the hybridization, complementary subfragments of an oligonucleotide from the library and the longer DNA fragment join each other. The most intensive points from the fluorescent image of the chip correspond to oligonucleotides complementary to the ones joined entirely to the DNA fragment. Knowing coordinates of these points one can determine oligonucleotides composing the DNA fragment. These oligonucleotides, written as words of equal length over the alphabet {A, C, G, T}, make a set called a *spectrum*. The computational phase of the sequencing process consists in a reconstruction of an original sequence on the basis of the spectrum.

If the hybridization experiment was carried on without errors, then the spectrum would be *ideal*, i.e. it would contain only all subsequences of length l of the original sequence of the known length n. In this case the spectrum would consist of $n-l+1$ elements and to reconstruct the original sequence one should find an order of spectrum elements such that neighboring elements always overlap on $l-1$ nucleotides. This can be done in polynomial time, e.g. by the reduction to the Eulerian path problem [Pev89]. (Of course, in a graph several Eulerian paths may exist and the problem is to choose the right one.)

However, the hybridization experiment usually produces errors in the spectrum. There are two types of *errors*: *negative* ones, i.e. missing oligonucleotides in the spectrum (e.g. because of subfragment repetitions), and *positive* ones, being erroneous oligonucleotides. If the coordinates of a point on the chip are not correctly read, two errors appear simultaneously: a negative one and a positive one. Therefore, as a result of the hybridization experiment, one obtains a spectrum in which not all words contained in the original sequence appear and words not contained in the original sequence appear. We assume no additional information to be known (e.g. about the probability of existence of an oligonucleotide in the sequence, or about a partial order of oligonucleotides).

The presence of negative errors in a spectrum forces the overlapping between some neighboring oligonucleotides in a sequence on less than $l-1$ letters. The presence of positive errors in a spectrum forces the rejection of some oligonucleotides during the reconstruction process. The existence of errors in the DNA sequencing results in a strongly NP-hard combinatorial problem [BK99]. There exist methods assuming errors in the spectrum, exact and heuristic ones, but almost all of them consider a reduced model of errors [Pev89, DLC91, Lip93, HHO$^+$94, BKK$^+$97]. The only exact method for the DNA sequencing problem allowing for any type of errors and no additional information about the spectrum, has been presented in [BFK$^+$99]. It generates solutions composed of a maximal number of spectrum elements (a version of the Selective Traveling Salesman Problem, i.e. the one where besides usual arc cost also a node profit is assumed and the goal is to find a route with the maximal total node profit without exceeding the total cost of the route), which leads to the reconstruction of original sequences, provided that the majority of spectrum elements are correct. The same criterion function has been used in

the tabu search methods for the problem with the most general model of errors [BFG+99, BFK+00].

The current work presents a new approximation algorithm for this problem. The advantage of this method is that it permits discrimination of the errors, negative ones as well as positive ones, and brings a new and original solution strategy for managing repetitions.

The proposed method builds a *superstring* with the aim of optimizing a cost function based on its thickness. The thickness of the *superstring* at a given position (p) is defined as the number of oligonucleotides that contain the base present at p. This method allows to provide a *reliability factor* for each produced solution. The potential erroneous oligonucleotides (entailing positive as well as negative errors) can also be pointed out.

The paper is organized as follows. Section 2 contains basic definitions and a presentation of the new algorithm. In Section 3 the algorithm's behavior is analyzed, while in Section 4 its computational performance is compared with the previous heuristic method. Section 5 concludes the work.

2 Approximation algorithm

2.1 Definitions

Firstly, some basic definitions useful for the presentation of the new algorithm, are given.

1. *Spectrum*
   We call a *spectrum* the set of oligonucleotides obtained as a result of the biochemical phase of the SBH method. Within a given spectrum all oligonucleotides have the same length l. They are also called l-mers. Due to the DNA-chip technique, no oligonucleotide can appear twice or more in the spectrum, even if it actually appears more than once in the original DNA-segment. We see that two kinds of errors can appear in the spectrum: *positive* and *negative* errors. The former ones correspond to oligonucleotides which are not present in the original segment but that were detected by the sbh method. The latter ones correspond to oligonucleotides which are present within the original sequence but which were not detected either because of an experimental error or because of their repetitive appearance (in this case only one copy of such an oligonucleotide appears in the spetrum).

2. *Sub-segment*
   We call *sub-segment* a DNA-chain built by at least 2 oligonucleotides.

3. *Predecessors and successors of an oligonucleotide*
   Let us consider two different oligonucleotides M and M' of length l. We say that M is a *predecessor* (resp. *successor*) of M' if the k ($1 \leq k \leq l-1$) last (resp. first) nucleotides of M are the same as the first (resp. last) k nucleotides of M'. A predecessor (resp. successor) M of M' is called *immediate* if $k = l - 1$. We will say that M is a *predecessor* of M' (resp. *successor*) *of order* k.

4. *Actual predecessors and successors of order k*
   We call *actual predecessor* (resp. successor) of M' of order k in S (a sub-segment), the oligonucleotide M that precedes (resp. follows) M' in S, and has k consecutive nucleotides in common with M.

5. *Overlapping window*
   Let M' be either a predecessor of M or its successor. Using the previous definitions, the overlapping window of M is the set of predecessors and successors of M of order between C_{min} and $l-1$, where C_{min} is a parameter of the algorithm.

6. *Thickness*
   We start with a definition of the thickness of a nucleotide within an oligonucleotide to be a number of oligonucleotides that have this nucleotide in common (on a given position) within a segment for a given order of oligonucleotide s. We also define the thickness of an oligonucleotide as the mean value of the thicknesses of all its nucleotides, provided the above order is given. This is illustrated in Figure 1.

   Let us note that the notion of oligonucleotide thickness is crucial for the proper reconstruction of the relative order of oligonucleotides composing a sequence. The bigger the oligonucleotide thickness is, the better its current position fits into the reconstructed pattern. Thus, its maximum value for the oligonucleotide placed in a position where no missing neighbors appear is equal to l (its length).

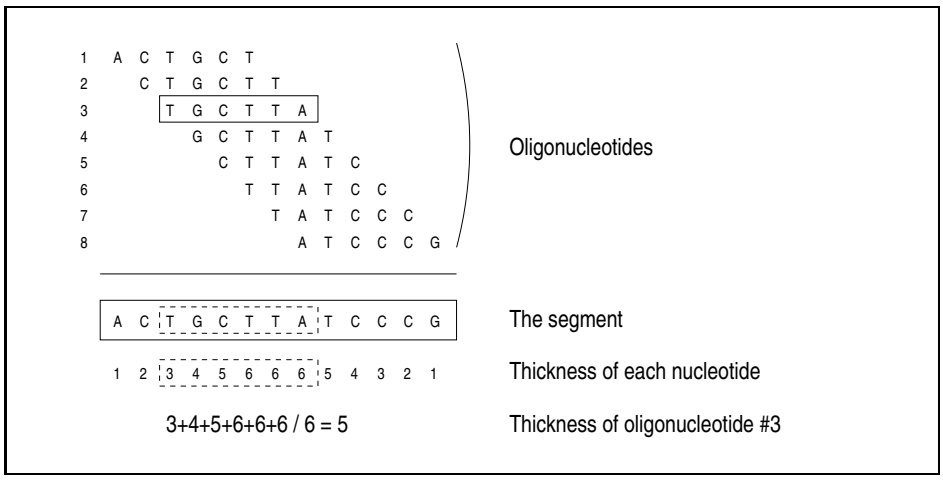


Figure 1: Thickness of an oligonucleotide. An example oligonucleotide #3 has its thickness equal to 5

7. *Free predecessor or successor*
   A successor or a predecessor is said to be free if it is not already assigned.

2.2 General principle of the algorithm

The algorithm is based on both overlapping windows of oligonucleotides, and on fusions of sub-segments. Overlapping windows give a local view of the over-

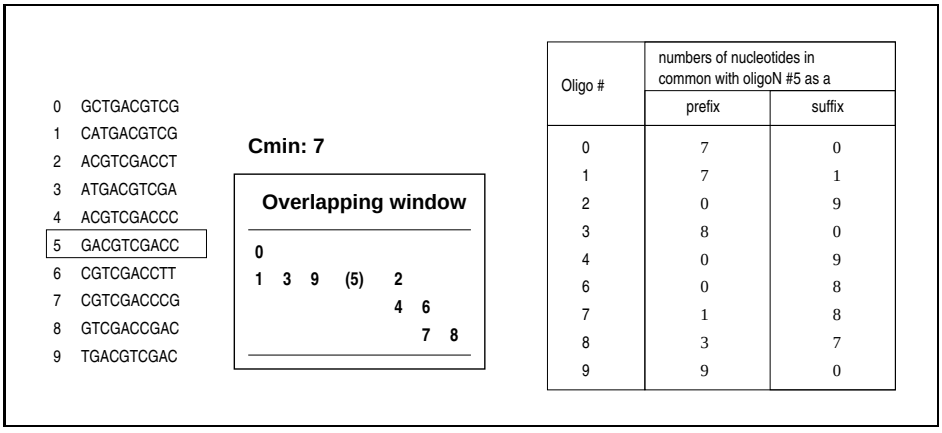
lapping environment of each oligonucleotide, while the fusion corresponds to a global reduction. The analysis of the overlapping windows of oligonucleotides leads to the creation of a set of sub-segments. From this set, the algorithm produces one sequence or several fragments included in the original sequence. The thickness is used as a cost function for the choice of the oligonucleotides involved in the fusion (for resolving the conflicts between oligonucleotides involved). The bigger the oligonucleotide thickness is, the better its current position fits into the reconstructed sequence. Obtaining a unique DNA-segment from the set of sub-segments is not always possible, so, the process can lead to the production of several fragments. To sum up the above considerations, we may say that the algorithm is composed of the following steps.

1. Overlapping windows computation
2. Sub-segment building
3. Reduction (fusion of sub-segments)

Steps #1 and #2 are performed only once, while step #3 is an iterative process leading to a construction of the final sequence. They are described below.

2.3 Overlapping windows

An overlapping window is built for each oligonucleotide. Given an oligonucleotide M, the construction of its overlapping window is straightforward. It could be constructed in linear time by using a prefix tree. An example of such a window is illustrated in Figure 2, and a set of overlapping windows for all the oligonucleotides belonging to a spectrum is given in Figure 3.



Oligo #	numbers of nucleotides in common with oligoN #5 as a	
	prefix	suffix
0	7	0
1	7	1
2	0	9
3	8	0
4	0	9
6	0	8
7	1	8
8	3	7
9	9	0

Figure 2: An example of an overlapping window for oligonucleotide #5. Its closest successors and predecessors are calculated on the basis of suffices and prefices, respectively, presented in the neighboring table

Let us note that in Figure 3 oligonucleotides #16 and #17 are repeated in the sequence and together with a missing oligonucleotide (crossed in Figure 3) constitute negative errors. On the other hand, oligonucleotides #21 and #22 are positive errors in the considered spectrum.

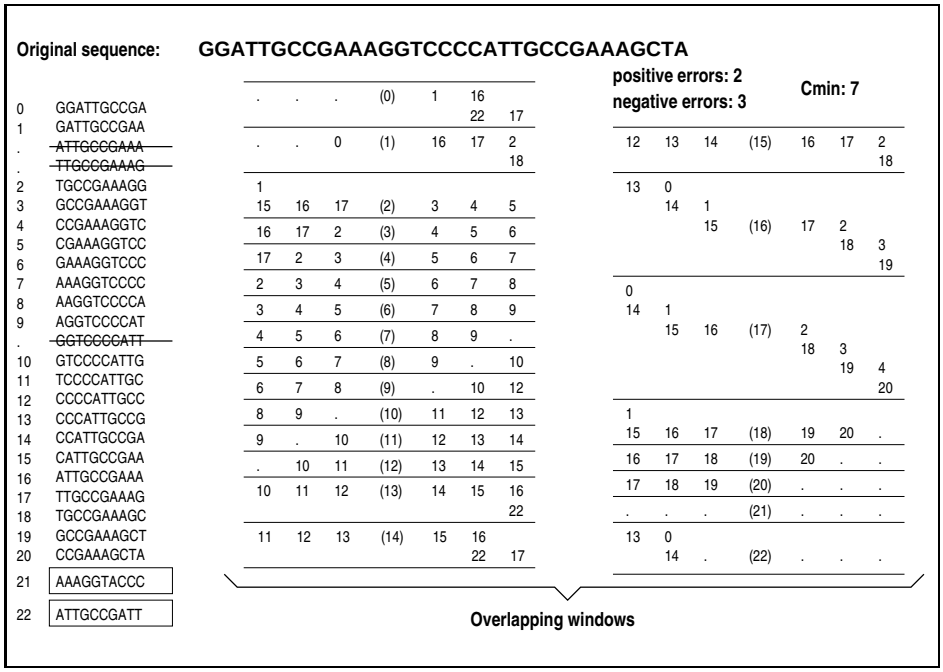


Figure 3: A set of overlapping windows for a given spectrum. Positive errors are highlighted by boxes while negative errors are crossed. Let us note that two of them are the result of repeated fragments while the last one is due to an experimental error.

2.4 Sub-segment building

2.4.1 Notation

- M - an oligonucleotide under current investigation,
- M_s - an immediate successor of M,
- M_p - an immediate predecessor of M,
- M_s^k (resp. M_p^k) - a successor (resp. a predecessor) of M of order k.

2.4.2 Principle

From the set of overlapping windows sub-segments are built. Among the set of oligonucleotides with no predecessor, if any, one is randomly chosen to be the starting oligonucleotide of the sub-segment. If no such oligonucleotide exists, any oligonucleotide can be chosen with equal probability. The algorithm begins sub-segment building by adding a successor to the current oligonucleotide.

The choice of the successor depends on the availability of free successors. Namely, the following situations can occur:

1. M has only one free immediate successor M_s.
   $\rightarrow$ *algorithm behavior:* M_s is added to the sub-segment and becomes the next current oligonucleotide.

2. M is followed only by oligonucleotides which are not free.
   $\rightarrow$ *algorithm behavior:* the current sub-segment building stops. Another sub-segment is begun if there remain some oligonucleotides not yet examined.

3. M is followed by at least two free immediate successors.
   $\rightarrow$ *algorithm behavior:* one of them is randomly chosen.

4. The closest free successor of M with the highest order (called the first free successor) is of order k (M_s^k), $k < l - 1$. (In case of more than one such successor, one of them is randomly chosen.)
   $\rightarrow$ *algorithm behavior:* the algorithm adds $l - 1 - k$ holes (represented by dots in the sub-segments of Figure 4) between M and M_s^k.

Figure 4 illustrates the process based on the set of overlapping windows of Figure 3. For this example, the algorithm starts with oligonucleotide #0. Oligonucleotide #1 is the only free immediate successor, so it is added to the sub-segment. The algorithm proceeds until oligonucleotide #17 which has two immediate successors: #2 and #18. #2 is chosen randomly (below the alternative choice will be described). The process can then continue until oligonucleotide #9, which has a hole as an immediate successor. There is only one free successor of order 8: oligonucleotide #10. So the algorithm adds one hole before oligonucleotide #10. #10 is followed by #11... Until oligonucleotide #15. The only immediate successor of #15 (#16) is not free, so the algorithm tries to find another successor of lower order. The next one (#17 of order 8) is also used. Finally, oligonucleotide #18 of order 7 is free. Then, two holes are added to the sub-segment followed by oligonucleotide #18. Next, #19 and #20 are both added and the sub-segment building stops, because #20 admits no successor. Two other sub-segments can be built being composed by a single oligonucleotide each (oligonucleotides #21 and #22).

2.5 Reduction phase

The previous phase led to the construction of a set of sub-segments. In these sub-segments, each oligonucleotide appears only once.

The aim of this phase is to reduce the number of sub-segments in order to obtain a single segment at the end of the process. Given a sub-segment, its possible extension is studied from $5'$ end and $3'$ end, called later the head and tail respectively. Let M_f be the first oligonucleotide of the first built sub-segment. At first, we try to determine a predecessor M_{fp} of M_f to extend the sub-segment from the head. We do this by searching the set of predecessors of M_f present in its overlapping window, and by choosing the closest predecessor (in case of several possibilities we randomly choose one predecessor). In the most general case, a predecessor appears in another sub-segment. The algorithm simulates the extension of the current sub-segment and validates this coupling if the thickness of the predecessor is at least as big as it was in the previous sub-segment, and if the thickness of the current studied oligonucleotide is not decreased. This operation leads to the creation of two new sub-segments from the previous ones which don't exist any more. Such a situation is illustrated in Figure 5.

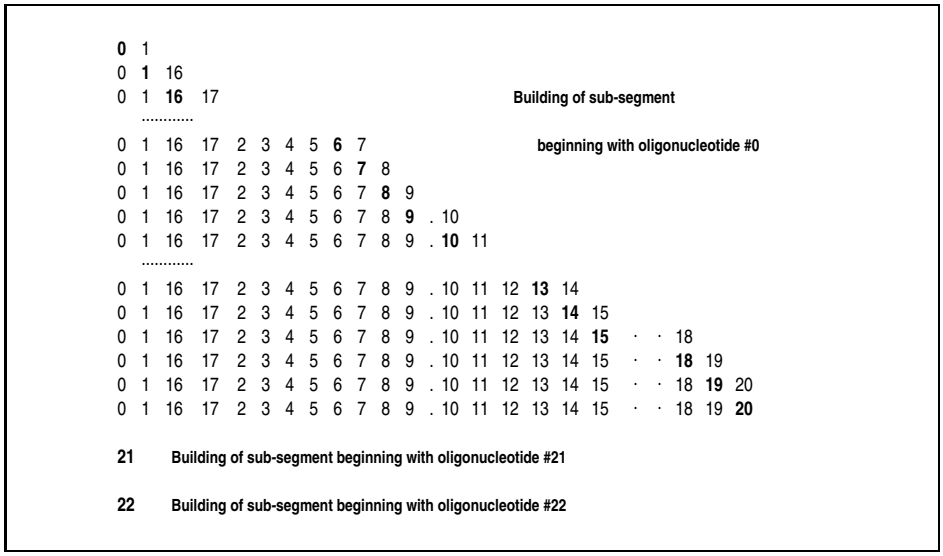


Figure 4: Sub-segment building. The process starts with oligonucleotide #0 and then continues by adding consecutive successors until no free successor is available. Oligonucleotides #21 and #22 have no successors.

If an extension starting from M_f was not possible, the algorithm tests the next oligonucleotide. This process goes on with the same sub-segment until an extension is accepted or until the algorithm reaches the oligonucleotide in position C_{min}. When the extension process from the head is completed, the same process is applied to the tail of the sub-segment.

2.6 General framework

General framework of the algorithm linking the different phases together is presented below.

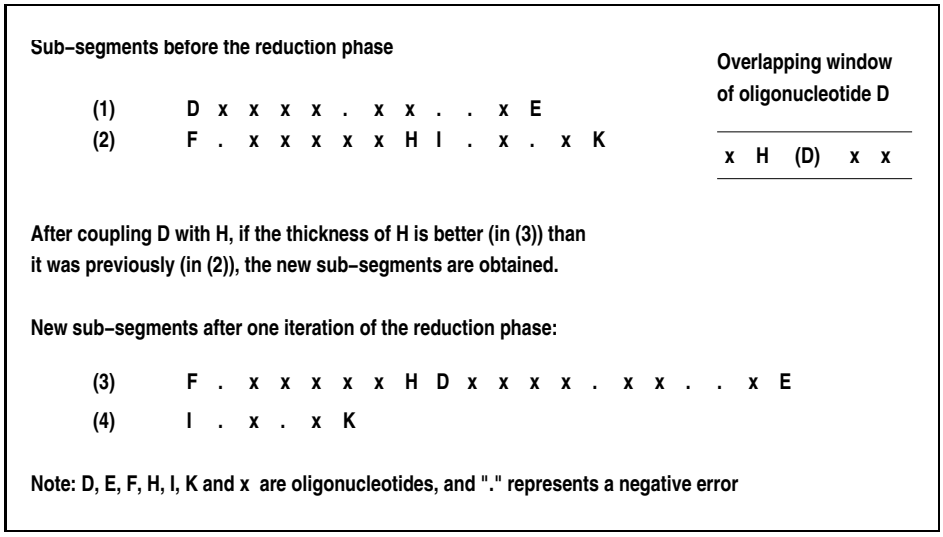


Figure 5: Reduction example. As a result of sub-segment building one gets sub-segments #1 and #2. Then, in the reduction phase the head of the first sub-segment (D) is linked to its predecessor H. If the new thickness of H is bigger, one constructs new sub-segments #3 and #4.

```
overlapping windows computation
for every oligonucleotide do
    L_subSeg ← segments coming from sub-segments building
    current sub-segment ← first element of L_subSeg
    while (|L_subSeg| ≠ 1) and (some sub-segments have not been studied yet) do
            extension of the current sub-segment from the head
            if extension accepted then
              remove the old sub-segments
              add the new sub-segments to the list
            else
              extension of the current sub-segment from the tail
              if extension accepted then
                remove the old sub-segments
                add the new sub-segments to the list
              endif
            endif
            current sub-segment ← next sub-segment in the list
    endwhile
    sub-segments with a thickness less than a given threshold are removed
    print all the elements of L_subSeg (solutions)
endfor
```

3 Algorithm's analysis

The algorithm presented takes into account the specificity of SBH, and returns one solution (either one sequence or a set of fragments). Moreover, most of the positive and negative errors can be detected as well as oligonucleotides which are involved in a repetition scheme.

3.1 Error detection

From the structure of the overlapping windows, some information about the set of oligonucleotides can be deduced.

For example, oligonucleotide #5 (Figure 6) can be considered as a *potential* positive error. Indeed, there exist no oligonucleotide, in the spectrum, close enough (with respect to a value of $C_{min} = 5$) to be considered.

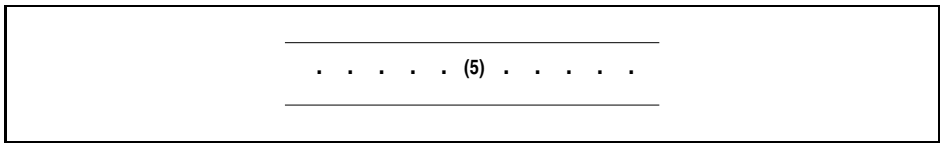


Figure 6: Potential positive error detection. Oligonucleotide #5 and its overlapping window which indicates empty sets of predecessors and successors.

Potential negative errors can also be detected when studying overlapping windows. They appear as holes, as illustrated in Figure 7 by a hole between oligonucleotides #13 and #6. The hole indicates that no oligonucleotide belonging to the spectrum can be placed between oligonucleotide #13 and oligonucleotide #6. Thus, the hole corresponds to a missed oligonucleotide, if in the original sequence oligonucleotide #13 is followed by oligonucleotide #6. Let us note, that the method can also find any number m of repeats of the type (AT)×m provided that the length of the oligonucleotide l is big enough, i.e. $l > 2m$.

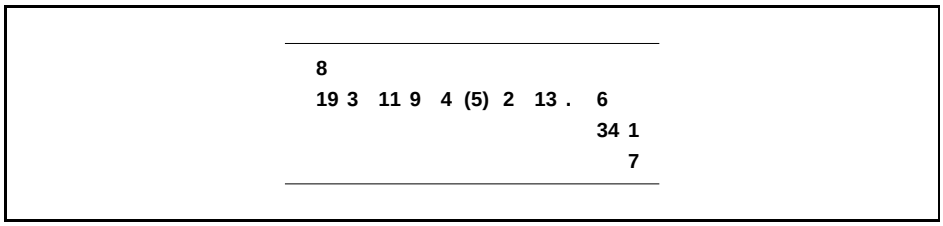


Figure 7: Potential negative error detection. It appears as a hole in the overlapping window

3.2 Successor choice

The two key points of the present algorithm are the successor choice while building the sub-segments, and the cost function used for the reduction phase. For the successor choice, the algorithm's behavior is the following (cf. Section 2.4.2. for the algorithm's description):

1. M is followed by only one immediate successor M_s.
   $\rightarrow$ *method analysis:* if M_s is actually a positive error, one can expect that the remaining part of the sub-segment will have a small thickness such that the end of it will be replaced by a better one during the reduction phase.

2. M is only followed by holes or by oligonucleotides which are not free.
   $\rightarrow$ *method analysis:* there is no way to proceed differently for the current value of C_{min}. But another run of the algorithm with a smaller C_{min} could lead to a better segment construction if M is actually followed by another successor of order $k > C_{min}$.

3. M is followed by at least two immediate free successors M_s and M'_s.
   $\rightarrow$ *method analysis:* the choice can be erroneous.

4. The first free successor of M is of order k, $k < l$.
   $\rightarrow$ *method analysis:* if several such successors exist, one of them is randomly chosen, which may be a wrong choice.

4 Experimental results

In the computational experiment, the results obtained using the tabu search method described in [BFG+99] and [BFK+00] (being somehow a certain standard in the literature), have been compared with those obtained using the current strategy. The tabu algorithm used a greedy procedure, described in [BFK+99], for generating initial solution. The parameters of tabu algorithm have been described in detail in [BFG+99], the most important being the length of the tabu list (i.e. the minimum number of different consecutive algorithm's moves), assumed to be equal to 5. The sequences produced by both methods have been compared with original sequences using a classical pairwise alignment algorithm [Wat95].

The experiment has been performed on CRAY T3E-900 in Poznań Supercomputing and Networking Center. All spectra used in the experiment are derived from real DNA sequences coding human proteins (taken from GenBank, National Institute of Health, USA). Their accession numbers are given in the Appendices 1 and 2. In the first set of experiments the sequences contained 20% of random positive errors and 20% of random negative errors (here sequences had no repetitions longer than 20 nucleotides). Because cardinalities of the spectra varied from 100 to 500 oligonucleotides, they contained from 40 to 200 errors. The following idea has been used to introduce errors into the spectrum (cf. [BFK+99]). For a given cardinality of the spectrum (obtained from the original sequence) 20% of randomly selected (according to a uniform distribution) l-mers (*negative errors*) have been deleted, and next the same number of randomly generated *positive errors* have been included. Moreover, l-mers added to the spectrum had to be different from those already existing in it. The spectra have been sorted alphabetically and the first oligonucleotide of each original sequence has been known. The latter assumption is justified by information coming from biochemical experiments. For the first set of experiments, the size of oligonucleotides has been equal to 10, while it has been equal to 7 for the second set of experiments, in order to introduce errors coming from repetitions

to the spectrum. Indeed, the smaller the size of oligonucleotides is, the larger the number of repeated oligonucleotides is.

An obtained sequence has been called an *optimum* if it matched exactly the corresponding original sequence. If the produced sequence has been shorter than the original one, but this fact has been caused only by negative errors at the end of the sequence (in this case there is no information in spectrum about the last nucleotides), it has been called a *partial optimum* (see Figure 8). In the data used in the experiment there has been no instance with more than 4 nucleotides missing at the end.

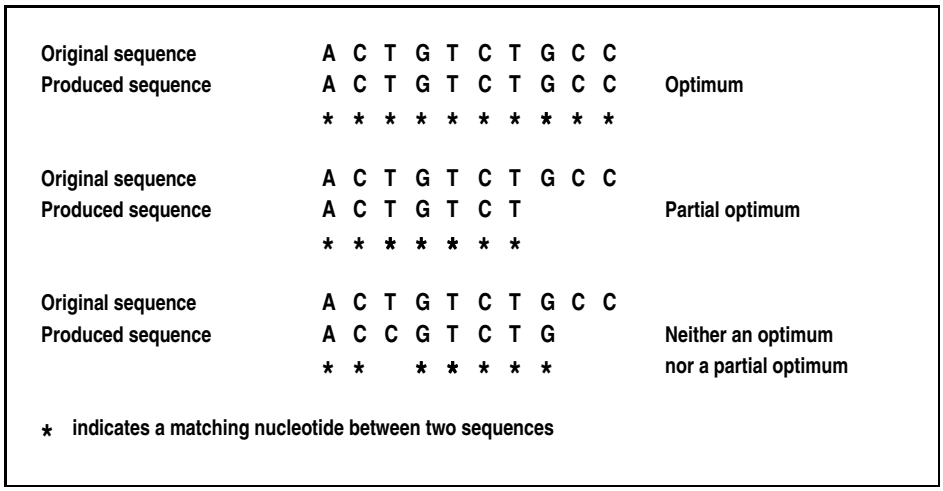


Figure 8: An example of a partial optimum. Some nucleotides are missing in the obtained sequences.

The alignment algorithm, comparing obtained sequences with original ones, has been called with the following parameters: a match (the same nucleotides at a given position in strings) brings a profit of 1 point, a mismatch (different nucleotides) brings a penalty of 1 point (i.e. -1) and a gap (an insertion, a nucleotide against a space) also brings a penalty -1. Therefore, the highest score would be equal to a number of nucleotides in the sequence (in case the two sequences are the same) and the lowest score would be equal to the number of nucleotides in the longer sequence times -1 (in case the two sequences are totally different).

The results of the first set of experiments with random errors are given in Table 1, where every column represents mean values for 40 instances (accession numbers given in Appendix 1) generated in the way described above. Parameters of tabu algorithm have been described above, while the only parameter of the analyzed algorithm C_{min} (being the minimum accepted size of the overlap between the two neighborhing oligonucleotides) was assumed to be 5. The scores are shown as numbers of points (with maximal values from 109 to 509, respectively) and in percents (with the maximum 100% in case two sequences are equal). Next, numbers of optimal and partially optimal results (among 40), and computation times (in seconds), are presented.

The proposed algorithm as compared with the tabu method has some advantages. Firstly, it generates solutions much faster. Moreover, it returns more

Results obtained using the tabu search method					
Spectrum size	100	200	300	400	500
Average similarity score [pt]	105.1	184.5	244.6	315.1	312.3
Average similarity score [%]	98.6	94.1	89.6	88.5	80.7
Optimum #	28	23	17	10	10
Partial optimum #	10	8	9	7	1
Average computation time [sec]	<1	5	14	28	51

Results obtained using the current heuristic					
Spectrum size	100	200	300	400	500
Average similarity score [pt]	107.8	188.8	282.3	350	408
Average similarity score [%]	99.4	95.2	95.7	92.1	90.1
Optimum #	28	20	21	13	14
Partial optimum #	10	9	9	11	3
Average computation time [sec]	<1	<1	<1	<1	<1

Table 1: A comparison of the results obtained by the two heuristic methods.

often optimal (or partially optimal) solutions than the other method.

In addition, a relationship between results obtained for instances with random positive errors and with a more realistic model of this type of errors has been investigated. (The more realistic means that the false oligonucleotides has been generated by a duplication of a correct oligonucleotide from spectrum and changing its first or last nucleotide or both of them. This approximates errors which may appear during hybridization experiments where almost complementary subsequences can create duplexes.) This analysis has not shown any significant differences between a quality and computation time for both types of instances.

Next, another set of experiments has been conducted taking into account errors resulting from repetitions, being the main and realistic source of negative errors occuring while sequencing by hybridization. For these experiments a new set of 78 sequences has been chosen in GenBank, their accession numbers being described in Appendix 2. In the spectra resulting from these sequences, only negative errors caused by repetitions has been assumed. The cardinalities of all these spectra have been equal to 500, while the numbers of negative errors varied between 10 and 20, and $l = 7$. The overlap value was taken equal to 3. The results of these experiments are gathered in Table 2 where each entry is a mean value obtained for 78 sequences. Again studying these results we see the advantage of the method proposed.

The results from Table 1 have been obtained by enforcing on the proposed algorithm the return of a solution exactly in one part. However, in case of many errors in spectra there is often not enough information about some places in sequences. Then, a request for a continuous solution would result with a high probability in a sequence differing a lot from an original one, because not well connected parts would be ordered randomly. It could be more useful for biochemists to get two or more subsequences unordered but being near-exact parts of an original sequence. This information could help them to compose a correct one-part result, e.g. by doing an additional experiment. The proposed algorithm is suited to provide that information. Table 3 shows results of tests

Spectrum size: 500	Tabu search method	Current heuristic
Average similarity score [pt]	113.49	258.05
Average similarity score [%]	61.21	75.49
Optimum #	0	0
Partial optimum #	0	8%
Average computation time [sec]	27	0.5

Table 2: A comparison of the results obtained by the two heuristic methods for sequences with errors resulting from repetitions. The results were obtained for an overlap value of 3 et for sequences containing at least 70% of the spectrum.

with the overlap value C_{min} changing from 3 to 7 (for $l = 10$). In general, the larger the overlap value is, the larger the similarity of subsequences to parts of an original sequence is and the greater the number of disconnected parts is in a solution. However, in the worst case (with overlap = 7 and |spectrum| = 500) we have got only about 5 subsequences on the average. Most of the solutions have consisted of no more than two parts.

Overlap	Spectrum size				
	100	200	300	400	500
3	1.00	1.00	1.00	1.00	1.00
4	1.00	1.13	1.10	1.28	1.35
5	1.03	1.18	1.28	1.58	1.65
6	1.30	1.65	1.68	2.30	2.63
7	1.78	2.73	4.20	4.75	5.68

Table 3: Average number of disconnected subsequences in a solution

5 Conclusions

In this paper the new method for rebuilding sequences from a set of oligonucleotides with the aim of managing both positive and negative errors has been proposed. This method is simple and fast, and behaves surprisingly well when the length of the oligonucleotides is large enough to ensure that only a few of them accept more than one immediate successor. Indeed, the main drawback of the current method remains the choice of the successor. Nevertheless, the method seems to be particularly well suited for detecting both kinds of errors and its improvement by incorporating a tabu search procedure for the choice of the successor when several "good candidates" are available, is planned.

References

[AG97] A. Apostolico and R. Giancarlo. Sequence alignment in molecular biology. In M. Farach, F. Roberts, and M. Waterman, editors, *Mathematical support for molecular biology*, American Mathematical Society, DIMACS, 1997.

[BFG+99] J. Błażewicz, P. Formanowicz, F. Glover, M. Kasprzak, and J. Węglarz. An improved tabu search algorithm for DNA sequencing with errors. In *Proceedings of the III Metaheuristics International Conference,* Angra dos Reis, 69–75, 1999.

[BFK+99] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, and J. Węglarz. DNA sequencing with positive and negative errors. *Journal of Computational Biology*, 6:113–123, 1999.

[BFK+00] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, and J. Węglarz. Tabu search for DNA sequencing with false negatives and false positives. *European Journal of Operational Research*, 125:257–265, 2000.

[BK99] J. Błażewicz and M. Kasprzak. Complexity of DNA-sequencing with errors. In *RECOMB'99 (Lyon)*, 1999.

[BKK+97] J. Błażewicz, J. Kaczmarek, M. Kasprzak, W. T. Markiewicz, and J. Weglarz. Sequential and parallel algorithms for DNA sequencing. *Computer Applications in the Biosciences (CABIOS)*, 13:151–158, 1997.

[BS88] W. Bains and G. C. Smith. A novel method for nucleic acid sequence determination. *Journal of Theoretical Biology*, 135:303–307, 1988.

[DLB+89] R. Drmanac, I. Labat, I. Brukner, and R. Crkvenjakov. Sequencing of megabase plus DNA by hybridization: theory and the method. *Genomics*, 4:114–128, 1989.

[DLC91] R. Drmanac, I. Labat, and R. Crkvenjakov. An algorithm for the DNA sequence generation from k-tuple word contents of the minimal number of random fragments. *Journal of Biomolecular Structure and Dynamics*, 8:1085–1102, 1991.

[FRP+91] S. P. A. Fodor, J. L. Read, M. C. Pirrung, L. Stryer, A. T. Lu, and D. Solas. Light-directed, spacially addressable parallel chemical synthesis. *Science*, 251:767–773, 1991.

[HHO+94] J.N. Hagstrom, R. Hagstrom, R. Overbeek, M. Price, and L. Schrage. Maximum likelihood genetic sequence reconstruction from oligo content. *Networks*, 24:297–302, 1994.

[Lip93] R.J. Lipshutz. Likelihood DNA sequencing by hybridization. *Journal of Biomolecular Structure and Dynamics*, 11:637–653, 1993.

[LYF+88] Lysov, P. Yu, V. L. Florentiev, A. A. Khorlin, K. R. Khrapko, V. V. Shik, and A. D. Mirzabekov. Determination of the nucleotide sequence of DNA using hybridization with oligonucleotides. *Doklady Akademii Nauk SSSR*, 303:1508–1511, 1988.

[MAR+94] W. T. Markiewicz, K. Andrych-Rożek, M. Markiewicz, A. Żebrewska, and A. Astriab. Synthesis of oligonucleotides permanently linked with solid supports for use as synthetic oligonucleotide combinatorial libraries. innovations in solid phase synthesis. In R. Epton, editor, *Biological and Biomedical Applications*, pages 339–346. Mayflower Worldwide, Birmingham, 1994.

[MG77] A. M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences USA*, 74:560–564, 1977.

[Pev89] P.A. Pevzner. l-tuple DNA sequencing: computer analysis. *Journal of Biomolecular Structure and Dynamics*, 7:63–73, 1989.

[PSS+94] A. Caviani Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, and S. P. A. Fodor. Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proceedings of the National Academy of Sciences of USA*, 91:5022–5026, 1994.

[SC78] F. Sanger and A. R. Coulson. The use of thin acrylamide gels for DNA sequencing. *FEBS Letters*, 87:107–110, 1978.

[SM97] J. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, Boston, 1997.

[Sou88] E. M. Southern. United Kingdom Patent Application gb8810400, 1988.

[VLM97] M. Vingron, H. P. Lenhof, and P. Mutzel. Computational molecular biology. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*. J. Wiley and Sons, Chichester, 1997.

[Wat95] M.S. Waterman. *Introduction to Computational Biology: Maps, Sequences and Genomes*. Chapman & Hall, London, 1995.

Appendix 1

Accession numbers for the sequences used in the experiment #1 (random errors).

D00723	D11428	D13510	X13440	X51535	X00351
X02994	X04350	Y00264	X58794	Y00649	X05299
X51841	X02160	X04772	X13561	X14758	X15005
X06537	Y00711	X05908	X07994	X13452	Y00651
X07982	X05875	X53799	X05451	X14322	X14618
X55762	X14894	X57548	X51408	X54867	X02874
X06985	Y00093	X15610	X52104		

Appendix 2

Accession numbers for the sequences used in the experiment #2 (errors resulting from repetitions).

D00723	D11428	D13510	X51535	X56088	X00351
X02994	X03350	X00318	Y00264	X58794	Y00649
X07577	X05299	X02160	X04772	X14758	X15005
Y00062	X06537	X05908	X17206	X13452	Y00651
X07982	X53331	X07362	X12510	X53799	X05451
X14322	Y00695	X14618	X54867	X02874	X15610
X52104	X04217	X04808	X04741	X14034	X05199
X57748	X53605	Y00971	X17610	X56976	X03484
X13973	X12654	X12453	X54534	X52967	X06617
X06614	X04608	X00457	X13697	X52973	Y00064
X02317	X07820	X05232	X52520	X03124	X16064
X16316	D13866	D14705	D10570	D90373	D13892
D90402	D00726	D90224	D16105	D13720	X01060