Evolutionary approaches to DNA sequencing with errors*

Jacek Błażewicz[‡†], Fred Glover[§], Marta Kasprzak[‡]

Abstract

In the paper, two evolutionary approaches to the general DNA sequencing problem, assuming both negative and positive errors in the spectrum, are compared. The older of them is based on the idea of genetic approach and is enhanced by a greedy algorithm. The newly proposed algorithm combines the tabu search and the scatter search methods. After conducting experiments with random and coding DNA sequences, our results suggest that the tabu and scatter search algorithm finds solutions of higher quality and more reliably than the genetic algorithm.

1 Formulation of the DNA sequencing problem by hybridization

The goal of the *DNA sequencing problem* is to determine a sequence of nucleotides of a DNA fragment. The data for the problem come from a biochemical experiment, called *hybridization* [BS88, LFK+88, DLB+89]. During the experiment, short subsequences (usually of equal length) of an unknown sequence are recognized. These subsequences, called *oligonucleotides*, can be written as a set of words of length l, over the alphabet {A, C, G, T} representing four nucleotides composing the subsequences. In order to reconstruct the unknown original DNA sequence of a known length n on the basis of this set (*spectrum*), the oligonucleotides should be ordered in such a way that the neighbors overlap each other (see Example 1).

In the ideal case, where there are no errors in the spectrum, all oligonucleotides must be used and the neighboring ones must overlap on $l-1$ letters. Obviously, in that case the spectrum has to contain $n-l+1$ elements. There exists a polynomial time algorithm solving the DNA sequencing problem without errors [Pev89]. However, if the spectrum does not contain some

*The research has been partially supported by KBN grant.

†To whom the correspondence should be addressed: blazewic@put.poznan.pl.

‡Institute of Computing Science, Poznań University of Technology, Piotrowo 3A, 60-965 Poznań, Poland, and Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12, 61-704 Poznań, Poland.

§University of Colorado, Boulder, CO 80309-0419, USA.

subsequences of the original sequence (i.e. *negative errors* appear), or if it contains some oligonucleotides not appearing in the original sequence (i.e. *positive errors*), the DNA sequencing problem becomes strongly NP-hard [BK03]. The hybridization experiment usually produces errors in the spectrum. During the reconstruction of the original sequence on the basis of the spectrum with errors, overlaps on less than $l-1$ letters must be allowed and some oligonucleotides from the spectrum must be rejected (see Example 1).

Example 1 *Suppose the original sequence to be found is CCGACGT, $n = 7$. As a result of the hybridization experiment performed without errors and with $l = 3$, one obtains the ideal spectrum for this sequence, containing all 3-letter substrings of the original sequence: {ACG, CCG, CGA, CGT, GAC}. The reconstruction of the sequence in that case consists in finding such an order of the spectrum elements, where each pair of neighboring elements overlaps on $l - 1 = 2$ letters. The only possible solution is (CCG, CGA, GAC, ACG, CGT). To simulate the experiment in a more realistic way, we introduce some errors into the spectrum. Let the negative error be CGA, and the positive errors be AAT and TTG. Then, the spectrum would have the following components: {AAT, ACG, CCG, CGT, GAC, TTG}. Now it is not possible to use all words to build a sequence of length n; not all neighbors will overlap on $l - 1$ letters as well.* □

The DNA sequencing problem with negative and positive errors in the spectrum, and with no additional information about spectrum elements assumed, for the first time has been formulated in [BFK+99] as a version of the Selective Traveling Salesman Problem. There, a directed graph was constructed in such the way that its vertices corresponded to spectrum elements, and its arcs reflected the costs of connecting oligonucleotides (the greater possible overlap of two oligonucleotides, the smaller cost of the arc connecting corresponding vertices). To each vertex a profit equal to 1 was assigned. Searching for the original sequence was equivalent to searching for a simple path of maximum total profit and of total cost not greater than a given bound equal to $n - l$. Experimental results confirmed, that this model leads to a construction of original sequences. The following integer programming problem is an alternative formulation of the DNA sequencing with errors.

maximize

$$\sum_{i=1}^{z}\sum_{j=1}^{z} b_{ij} + 1 \tag{1}$$

subject to

$$\sum_{i=1}^{z} b_{ik} \leq 1, \quad k = 1, \ldots, z \tag{2}$$

$$\sum_{i=1}^{z} b_{ki} \leq 1, \quad k = 1, \ldots, z \tag{3}$$

$$\sum_{k=1}^{z} \left(\left| \sum_{i=1}^{z} b_{ki} - \sum_{j=1}^{z} b_{jk} \right| \right) = 2 \tag{4}$$

$$\sum_{s_k \in S'} \left(\sum_{s_i \in S'} b_{ik} \cdot \sum_{s_j \in S'} b_{kj} \right) < \left| S' \right|, \quad \forall S' \subset S, \; S' \neq \emptyset \tag{5}$$

$$\sum_{i=1}^{z} \sum_{j=1}^{z} c_{ij} b_{ij} \leq n - l \tag{6}$$

where:

S – the spectrum,
s_i – an element of the spectrum,
z – the cardinality of the spectrum,
n – the length of an original sequence,
l – the length of a spectrum element,
b_{ij} – a boolean variable; it is equal to 1 if element s_i is the immediate predecessor of element s_j in a solution; otherwise it is equal to 0,
c_{ij} – a cost of connection of element s_i with element s_j; it is equal to the difference between l and a number of letters of the common part of s_i and s_j coming from their maximal overlapping.

The *criterion function* (1) to be maximized is equivalent to the number of spectrum elements composing the solution. Inequalities (2) and (3) guarantee that every element of the spectrum will be joined (in the solution) with, respectively, at most one element from the left side and at most one element from the right side. The addition of equation (4) ensures that exactly two elements, connected with other elements from one side only, will appear in the solution. These elements will constitute the beginning and the end of the reconstructed sequence. Supplying the above formulation with inequalities (5) allows to eliminate the solutions including subcycles of elements (when an element in the solution is simultaneously a successor and

the immediate predecessor of another element of the solution). According to inequality (6) the length of the reconstructed sequence cannot exceed its known length. (The length can be shorter, for example in case of negative errors appearing at the end of the sequence.)

Both evolutionary approaches presented in Section 2 and Section 3, respectively, solve the above DNA sequencing problem with negative and positive errors. Their computational outcomes are compared in Section 4 both for random and for DNA sequences coding for proteins. The special attention has been paid to the tests on coding sequences with repetitions, being especially hard for sequencing algorithms. All the tests showed an advantage of the newly proposed tabu and scatter search algorithm over the genetic one.

2 Hybrid genetic algorithm

In this section, the hybrid genetic algorithm proposed in [BKK02] will be shortly described. In this algorithm, a standard genetic approach [Gol89] is supplemented by a heuristic greedy improvement. The genetic representation of an individual (i.e. a *chromosome*) is a permutation of indices of oligonucleotides from the spectrum. An adjacency-based coding has been used: value i at position j in the chromosome means that the oligonucleotide i follows the oligonucleotide j. The function evaluating a fitness of an individual (the *fitness function*) takes the best substring of oligonucleotides in the chromosome, i.e. the one composed of the largest number of elements, provided it produces a sequence of length not greater than n letters. The neighboring oligonucleotides are assumed to be maximally overlapped, what gives the guarantee of including as many elements as possible in the evaluated substring. The normalized fitness value, used in the algorithm, equals the number of oligonucleotides in this substring divided by $n - l + 1$ (being the maximum number of spectrum elements in any valid sequence).

The *initial population* is randomly generated according to a uniform distribution, and its cardinality is a parameter of the method. Each of the individuals has to be a permutation of indices (as mentioned above) and it has to exclude subcycles containing fewer indices than the spectrum cardinality. Next, to each individual the normalized fitness value is assigned. The individual of the greatest value of the criterion function is stored. Then, the fitness values of all individuals in the population are linearly scaled, and the best ones are selected according to the stochastic remainder method without replacement [Gol89]. The *next population* is constructed from the best indi-

viduals, randomly paired, using the greedy crossover, an approach similar to the one from [GGR+85] (see also [Glo77] in the context of a scatter search approach). The greedy crossover is defined as follows. The first oligonucleotide in a chromosome is set randomly. Next, with the probability 20% we choose for a given oligonucleotide in the chromosome the best successor among the remaining oligonucleotides (the ones not yet used to build the chromosome). The best successor is defined to be the oligonucleotide which overlaps the previous one on the highest number of nucleotides. With probability 80% the following move is chosen: we take as the successor of a given oligonucleotide the one with a better overlap in the parents of the chromosome, provided it does not produce a subcycle in the chromosome; otherwise we take a random oligonucleotide among the remaining ones. In all cases, if there is more than one best choice, the first found is chosen. The procedure is iterated until all chromosomes of the population are constructed.

Every new population is submitted to the above series of operations, and each time the best individual found so far is recorded. The steps are repeated until a given number of iterations without improvement of the criterion function value is reached. The solution returned by the algorithm is a part of the best individual found during the computations.

3 Tabu and scatter search algorithm

The main scheme of the algorithm is based on tabu search [GL97], utilizing scatter search [Glo77, Glo99] as a part of the diversification strategy. In our approach the spectrum is represented by two data structures: an ordered list of oligonucleotides composing a *current solution*, and an unordered set of remaining oligonucleotides, called a *trash set*. At each stage of the computation, the number of elements from the list cannot be greater than the one that would produce a sequence of at most n nucleotides (with a maximum possible overlapping of the neighbors on the list). To satisfy this constraint, only the moves that do not lead to sequences of length greater than n are considered. Such moves are called *feasible*. At the beginning, the *initial solution* is created by the greedy heuristic from [BFK+99].

Three basic types of *moves* are used: an *insertion* (a move transferring an oligonucleotide from the trash set to the solution), a *deletion* (a move transferring an oligonucleotide from the solution to the trash set), and a *shift* (a move within the solution). Actions on single oligonucleotides are often not sufficient, so we have added moves using clusters. A *cluster* is a group of neighboring elements from the solution, linked together with

overlaps on $l-1$ letters in each case. The list of clusters is updated after every move. Inserted or shifted oligonucleotides are remembered by storing them on the *tabu list* for a given number of iterations. The list is checked if an attempt to shift or to delete an oligonucleotide is made, and these moves are prevented if the oligonucleotide is on the list. An element found on the tabu list may be deleted or shifted together with the cluster containing it. The element also may be deleted if there is no other feasible move. In such a case, an element that has been on the tabu list for the greatest number of iterations is chosen.

The *global criterion function* to be maximized is the number of spectrum elements composing the solution. On the other hand, a function that is able to compare all kinds of moves is a *condensation*, defined for each solution to be the ratio of the number of oligonucleotides from the spectrum in the solution to the number of nucleotides in the solution. If the moves were compared by the global criterion function only, deletions or shifts would be used very rarely. Maximizing the condensation causes the initial solution to be transformed into a series of collections of well-matched oligonucleotides. (If a maximal value of the condensation is achieved by more than one move, the method selects the move resulting in the greatest number of elements in the solution. Consequently, insertion is the most preferred move with shifts, deletion of an oligonucleotide and deletion of a cluster being next.) Obviously, using the condensation as the only criterion for choosing a move would lead after a number of iterations to the creation of a single cluster of length much less than n letters. Consequently, we use both criterion functions (the global one and the condensation) during the search for a solution: the first one lenghtens the current solution, the second one condenses it. The above process of improving the current solution is the *intensification* part of the algorithm. The elements of the *diversification* strategy consist of extending moves and restarts based on the scatter search.

Extending moves are feasible moves selected by the use of frequency-based memory instead of the condensation function. They are executed after a given number of condensing moves without an improvement of the global criterion function value. The *frequency-based memory* is a tabu search structure that remembers the number of times each element from the spectrum appears in solutions. There are two types of extending moves: the insertion of an oligonucleotide and the deletion of an oligonucleotide. The more highly preferred move is the insertion, and the oligonucleotide with the lowest frequency value is chosen. If no insertion is possible, the oligonucleotide of the highest frequency value is deleted from the solution. After the execution of extending moves, the algorithm returns to the normal scheme with the

condensation as the criterion function. Such a combination of condensing and extending the solution guarantees that the number of oligonucleotides will increase from some value in the initial solution to a near-optimal or even optimal value in the final one.

Diversification is also present in the procedure of *restarting* the algorithm, based on the scatter search approach. During a given number of the cycles of condensing and extending moves, our scatter search approach constructs a *reference set* by remembering a selected number of the best generated solutions. The reference set is used in our present method as a source to generate a new initial solution within the restart procedure. The above use of the scatter search to guide the restarting process is different than its customary role, which operates within the main body of the algorithm.

A solution is a candidate to enter the reference set if it is better than one of the solutions in the set, i.e. it has a greater value of the global criterion function. The worst solution from the set is then deleted. To avoid the situation where a number of highly similar good solutions (e.g. differing only by one move) fill the set, the set can be updated only if at least 10 moves have been executed after the last update. The greater the difference between solutions in the set is, the greater the possibility of a good restart for the next intensification cycle is. This restriction is not used when considering a solution better than all the solutions present in the set. After that, we generate a solution using the greedy heuristic, in the same way as at the beginning of the algorithm. This solution replaces the worst one from the set. Then, we generate a new solution on the basis of the reference set, again using the greedy heuristic. However, this time the heuristic does not operate on all possible connections between oligonucleotides from the spectrum, but takes into account only those connections that are present within the solutions from the reference set. (An exception occurs when the current element has no successors. Then the method chooses the first not yet used oligonucleotide as its successor.) Hence, the graph representing the connections becomes rather sparse, as opposed to the complete graph used in the previous application of the heuristic. Now, as the first oligonucleotide in the solution we take in turn all spectrum elements, and the solution having the greatest value of the global criterion function is chosen as the new initial solution for the next cycle of condensing and extending moves. At the end, all algorithm variables are set to initial values (except for the variables remembering the best solution found so far), and the next search process can start, independently of the previous ones. The number of the restarts is a parameter of the algorithm. Once this number is reached, the solution containing the greatest number of oligonucleotides from the spectrum, found

so far, is returned by the algorithm.

4 Computational experiment

In the computational experiment, the two described algorithms have been compared. The experiment was performed on a PC station with a Pentium II 300 MHz processor, 256 MB RAM and the Linux operating system. We used in the tests random and coding DNA sequences. The coding sequences were taken from GenBank (National Institute of Health, USA). They are fragments of several genes coding human proteins (see Appendix). The random sequences have been generated according to the uniform distribution.

In the first part of the experiment (Tables 1–4), the lengths of sequences varied between 109 and 509 nucleotides (with step 100), and the length of oligonucleotides was always set to 10. First, we generated spectra without errors from these sequences, and their cardinalities were between 100 and 500 oligonucleotides. Next, we introduced randomly generated errors into these spectra: 20% of negative errors and 20% of positive errors, what resulted in spectra of the same cardinality. In the following notation "spectrum size = 500" means, that in the instance 100 randomly chosen oligonucleotides are missing and in addition 100 oligonucleotides are erroneous, i.e. the instance contains 400 oligonucleotides being parts of the original sequence. Finally, the spectra were sorted alphabetically in order to lose the information about the original order of oligonucleotides within sequences.

Parameters of the algorithms were set to values resulting (approximately) in similar computation times. The hybrid genetic algorithm was called with the number of iterations without improvement of the criterion function value set to 20, and with the population size set to 50. The tabu-scatter algorithm was called with: the number of condensing moves performed without improvement of the global criterion function value equal to 2, the number of extending moves equal to 4, the number of the cycles of condensing and extending moves equal to 300, the number of intensification stages (i.e. the number of restarts + 1) equal to 15, the length of the tabu list equal to 10, and the cardinality of the reference set used in restarts equal to 8. Parameter values for both methods were based on initial experimentation.

In Tables 1–4, all entries with average values have been calculated for 40 different instances. The quality is the number of spectrum elements composing a solution. The average quality is the mean value of the maximized criterion function in the algorithms. The optimal quality is the difference between the spectrum size and the number of negative errors. Below the

qualities, the numbers of instances (out of 40) for which the algorithms returned optimal solutions, are shown. Similarity score shows how much the original and generated sequences differ (with the maximum 100% in case the two sequences are equal). The sequences were compared by a classical pairwise alignment algorithm [Wat95], called with the following parameters: match=1, mismatch=-1, and gap=-1.

Spectrum size	100	200	300	400	500
Average quality	80.0	159.5	238.0	316.3	391.8
Optimal quality	80	160	240	320	400
Optimally solved instances	40/40	33/40	21/40	15/40	5/40
Average similarity score [%]	99.8	98.1	91.5	88.4	77.1
Average computation time [sec]	12.8	60.0	144.9	263.6	446.9

Table 1: Results of the hybrid genetic algorithm for random sequences.

Spectrum size	100	200	300	400	500
Average quality	80.0	160.0	239.5	319.1	397.3
Optimal quality	80	160	240	320	400
Optimally solved instances	40/40	40/40	33/40	29/40	18/40
Average similarity score [%]	99.8	99.9	93.1	95.2	85.6
Average computation time [sec]	9.5	42.5	134.4	321.9	567.7

Table 2: Results of the tabu and scatter search algorithm for random sequences.

Tables 1 and 2 present results of both algorithms tested on randomly generated sequences. As we see, the tabu and scatter search algorithm produces better solutions, concerning qualities as well as similarities to original sequences. However, the hybrid genetic algorithm also gives very good results, for example, all smallest instances were solved optimally. (The similarity less than 100% in that case is caused by few missing nucleotides at the ends of the generated sequences, what follows from negative errors placed at the ends.) The average qualities from Table 2 have near optimal values, what shows the high quality of the proposed strategy. Also the percentage of optimally solved instances is very high, especially for the great number of errors introduced to the spectra.

In general, DNA sequences coding human proteins are more difficult to reconstruct for sequencing algorithms than the random ones for they contain a greater number of repetitive subsequences, being a natural cause of ambiguous reconstruction. Tests on these sequences make it possible to check

Spectrum size	100	200	300	400	500
Average quality	80.0	159.2	237.5	316.2	392.8
Optimal quality	80	160	240	320	400
Optimally solved instances	40/40	29/40	22/40	15/40	6/40
Average similarity score [%]	99.7	98.0	91.8	90.9	80.9
Average computation time [sec]	13.1	61.3	145.9	273.9	432.2

Table 3: Results of the hybrid genetic algorithm for DNA coding sequences.

Spectrum size	100	200	300	400	500
Average quality	80.0	159.8	238.9	318.6	397.0
Optimal quality	80	160	240	320	400
Optimally solved instances	40/40	38/40	31/40	23/40	15/40
Average similarity score [%]	99.7	98.9	93.5	89.7	83.5
Average computation time [sec]	9.6	44.2	127.7	321.7	570.9

Table 4: Results of the tabu and scatter search algorithm for DNA coding sequences.

how the algorithms would work in practice. Tables 3 and 4 contain results of tests with spectra generated from DNA coding sequences. The average qualities are slightly lower than for the random sequences (however, not in all entries). It can be justified by the greater number of good connections in the set of oligonucleotides than in the case of random sequences. Once again the results produced by the tabu and scatter search algorithm are a little better than the ones of the genetic algorithm.

In order to make a deeper comparison of the two approaches, we performed the second part of the experiment on sequences with natural repetitions of oligonucleotides. Here, we used other 59 DNA coding sequences of length 509 nucleotides. We cut out spectra from these sequences, with oligonucleotide length set to 10, and we got spectra with natural repetitions of oligonucleotides. The repetitions are treated as special negative errors, and our instances contained from 1 to 32 such errors. Parameters of the algorithms had the same values as in the previous tests. The spectra also were sorted alphabetically. The results of tests performed on these instances are shown in Table 5.

This time the difference between the two algorithms is a bit easier to observe. The qualities obtained by the algorithms are similar, however, the tabu and scatter search algorithm returned the number of optimal solutions two times greater than the other, and it did it in much shorter computational

Algorithm	hybrid GA	tabu+scatter
Average obtained quality	493.6	495.4
Average optimal quality	496.2	496.2
Optimally solved instances	26/59	52/59
Average computation time [sec]	394.6	285.3

Table 5: Results of both algorithms for DNA coding sequences containing repetitions.

time. It should be noticed that the repetitions of oligonucleotides within an original sequence are much harder to solve than random negative errors. Every next repetition can increase the number of potential optimal solutions for the instance. Thus, it is impossible to evaluate the quality of an algorithm by comparing its results with original sequences. For this reason, we did not used the algorithm for pairwise alignment here.

5 Conclusions

In the paper, the DNA sequencing problem has been considered. Two algorithms: the hybrid genetic one and tabu and scatter search have been presented, and tested on instances containing negative and positive errors. Both algorithms returned results of a very high quality. However, the tabu and scatter search approach proves a little better for the easier instances and notably better for the harder instances.

A question arises, what the limit on length of sequences to be solvable is. Probably results of the algorithms for much longer sequences, especially the qualities and computation times, would be also satisfying. However, with the growing length of sequences, the probability that they contain repetitions of oligonucleotides increases. Then, we cannot guarantee that a solution of a very high quality covers the original sequence, since there are many possible optimal solutions (from the combinatorial point of view) for the problem. We suppose, that the applicability of the algorithms is limited to the sequences of lengths between 600 and 800 nucleotides.

The research plans for the future include both developing the existing algorithms and improving the idea of the hybridization experiment. The genetic algorithm may behave better after changing some of its steps towards more deterministic ones, especially the generation of the initial population. On the other side, some modification of the hybridization phase could reduce the number of errors in the spectrum (see the idea of isothermic oligonu-

cleotide libraries in [BFK+04]).

Acknowledgement

The authors acknowledge helpful remarks made by the referees.

References

[BS88] Bains, W. and G.C. Smith. (1988). "A novel method for nucleic acid sequence determination",*J. Theor. Biol.* 135, 303-307.

[BFK+04] Błażewicz, J., P. Formanowicz, M. Kasprzak, and W.T. Markiewicz. (2004). "Sequencing by hybridization with isothermic oligonucleotide libraries", *Disc. Applied Math.*, to appear.

[BFK+99] Błażewicz, J., P. Formanowicz, M. Kasprzak, W.T. Markiewicz, and J. Weglarz. (1999). "DNA sequencing with positive and negative errors", *J. Comp. Biol.* 6, 113-123.

[BK03] Błażewicz, J. and M. Kasprzak. (2003). "Complexity of DNA sequencing by hybridization", *Theor. Comp. Sci.* 290, 1459-1473.

[BKK02] Błażewicz, J., M. Kasprzak, and W. Kuroczycki. (2002). "Hybrid genetic algorithm for DNA sequencing with errors", *J. Heuristics* 8, 495-502.

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

[Glo77] Glover, F. (1977). "Heuristics for integer programming using surrogate constraints", *Decision Sci.* 8, 156-166.

[Glo99] Glover, F. (1999). "Scatter search and path relinking", In D. Corne, M. Dorigo, and F. Glover (eds.), *New Methods in Optimization.* McGraw Hill.

[GL97] Glover, F. and M. Laguna. (1997). *Tabu Search.* Norwell: Kluwer Acad. Publishers.

[Gol89] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading: Addison-Wesley.

[GGR+85] Grefenstette, J.J., R. Gopal, B.J. Rosmaita, and D. Van Gucht. (1985). "Genetic algorithms for the traveling salesman problem", *Proc. International Conference on Genetic Algorithms and Their Applications*, 160-168.

[LFK+88] Lysov, Yu.P., V.L. Florentiev, A.A. Khorlin, K.R. Khrapko, V.V. Shik, and A.D. Mirzabekov. (1988). "Determination of the nucleotide sequence of DNA using hybridization with oligonucleotides. A new method", *Dokl. Akad. Nauk SSSR* 303, 1508-1511.

[Pev89] Pevzner, P.A. (1989). "l-tuple DNA sequencing: computer analysis", *J. Biomol. Struct. Dyn.* 7, 63-73.

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

Appendix

The list of accession numbers from GenBank database corresponding to 40 DNA coding sequences used in the computational experiment (see Tables 3 and 4) is the following: 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.

The accession numbers of 59 DNA coding sequences with natural repetitions of 10-mers (see Table 5) are the following: X58377, X56088, X03350, X01098, X00318, X53279, X07577, X03663, X07173, Y00503, X07696, X03444, X03445, Y00815, Y00062, X13967, X17206, X01393, Y00809, X53331, X07362, X12510, X05450, Y00695, X54304, X13403, X13097, X04217, X04808, X03795, X04741, X52997, X04412, X07767, Y00345, X12385, X13405, X53605, Y00971, X13973, X00129, X54534, X04654, X06617, X13697, X12496, X02317, X07898, X02812, X05615, X01394, X16316, D10570, D28468, D12686, D90224, D14012, D11327, D16105.

The instances used in the computational experiment are available on request to the authors.