A 3-Approximation Algorithm for the Multiple Spliced Alignment Problem and Its Application to the Gene Prediction Task

Regina Beretta Mazaro, Leandro Ishi Soares de Lima, and Said Sadique Adi

Universidade Federal de Mato Grosso do Sul, Faculdade de Computação
Av. Costa e Silva, s/n, CP 549,
79070-900, Campo Grande, MS, Brazil
regina.beretta@ufms.br, leandro.ishi.lima@gmail.com, said@facom.ufms.br

Abstract. The *Spliced Alignment Problem* is a well-known problem in Bioinformatics with application to the gene prediction task. This problem consists in finding an ordered subset of non-overlapping substrings of a subject sequence g that best fits a target sequence t. In this work we present an approximation algorithm for a variant of the Spliced Alignment Problem, called *Multiple Spliced Alignment Problem*, that involves more than one target sequence. Under a metric, this algorithm is proved to be a 3-approximation for the problem and its good practical results compare to those obtained by four heuristics already developed for the Multiple Spliced Alignment Problem.

Keywords: Approximation algorithm, gene prediction, multiple spliced alignment problem.

1 Introduction

The term *Bioinformatics* has been used since 1970, when Hogeweg and Hesper defined it as "the study of informatic process in biotic systems" [4]. Since then, Biology and its branches have been a valuable source of new and interesting computational tasks involving long strings (genomic DNAs, cDNAs, RNAs, proteins, etc). As such, they require robust and efficient algorithms that work well in both theory and practice. A well-known task in this scenario is that of identifying the genes encoded in a genomic DNA of interest.

Given the practical importance and the difficulties associated with the gene prediction task, a number of computational methods has been developed to deal with it. By considering sequence conservation and the large quantity of entire genomes from many species already annotated, *similarity based* approaches are promising techniques that allow the identification of genes by comparing genomic sequences with related transcript sequences. In this context, Gelfand *et al.* proposed in [3] a theoretical/computational problem, called *Spliced Alignment Problem*, that models the gene prediction task as a combinatorial optimization problem involving (substrings of) a subject sequence (genomic DNA) and a target sequence (cDNA).

E. Moura and M. Crochemore (Eds.): SPIRE 2014, LNCS 8799, pp. 129–138, 2014.

In this work we propose an approximation algorithm for a variant of the Spliced Alignment Problem, called *Multiple Spliced Alignment Problem*, where more than one target sequence is involved. This problem was proved to be NP-complete by Kishi and Adi in [5], where they also proposed some heuristics for it. To the best of our knowledge, there are no approximation algorithms for the Multiple Spliced Alignment Problem in the literature, and it is exactly this gap that the present work wants to narrow.

This paper is organized as follows. In the next section we introduce the Spliced and Multiple Spliced Alignment Problem, and relate both with the gene prediction task. A 3−approximation algorithm for the Multiple Spliced Alignment Problem, that constitutes the main result of this work, is shown in Section 3. In Section 4 we give the details about the experimental results obtained by our approach over real-world instances of the gene prediction task. Finally, in the last section we summarize this work and consider future research directions.

2 The Multiple Spliced Alignment Problem

Among the several regions that comprise a genomic DNA, the protein coding regions, or *genes*, are of main interest for biologists. In eukaryotes, these regions are separated by long stretches of intergenic DNA and their coding fragments, called *exons*, are interrupted by non-coding ones, called *introns*. Given a genomic DNA, the gene prediction task consists in finding the correct exon-intron structure of its genes. In computational terms, this task has as input a genomic DNA sequence g and as output the start and end positions of each exon that constitutes the genes of g.

Given the undeniable practical importance of the gene prediction task, since 1980 many different methods have been proposed to address it. These methods can be roughly classified into *extrinsic* methods, that make use of information concerning fully annotated transcript sequences related to the target gene, and *intrinsic* methods, that rely basically on statistical information about the gene being searched for (see [7, 8] for surveys on this topic).

Among the different extrinsic approaches suggested for the gene prediction task, the one proposed by Gelfand *et al.* in [3] is of particular interest to the string processing field since it lies on a combinatorial optimization problem involving sequences, namely the *Spliced Alignment Problem*. To a better understanding of this problem consider the following definitions.

Let $s = s_1 s_2 \dots s_n$ be a finite string over an alphabet Σ. We denote the *length* of s by $|s|$. A *substring* $b = s_i \dots s_j$ of s is an ordered sequence of consecutive symbols of s. We denote by $first(b) = i$ the position of the first symbol of b in s and by $last(b) = j$ the position of the last symbol of b in s. Let $\mathcal{B} = \{b_1, b_2, \dots, b_k\}$ be a set of k substrings of s. We say that $\mathcal{B}$ is an *ordered set of substrings* if: 1) $first(b_i) < first(b_{i+1})$ or 2) $first(b_i) = first(b_{i+1})$ and $last(b_i) < last(b_{i+1})$, for $1 \leq i \leq k-1$. We also say that a substring $b' = s_o \dots s_p$ of s *overlaps* another substring $b'' = s_q \dots s_r$ of s if $q \leq o \leq r$, or $q \leq p \leq r$, or $o \leq q \leq p$, or $o \leq r \leq p$. Moreover, we say that a substring $b' = s_o \dots s_p$ of s

precedes another substring $b'' = s_q \dots s_r$ of s if $p < q$, and we denote this relation by $b' \prec b''$. A subset $\Gamma = \{b_i, b_j, \dots, b_p\}$ of $\mathcal{B}$ is a *chain* if $b_i \prec b_j \prec ... \prec b_p$ and we denote the string resulting of the concatenation of the elements of a chain Γ by $\Gamma^\bullet$. That is, $\Gamma^\bullet = b_i \bullet b_j \bullet \dots \bullet b_p$, where $\bullet$ is the string concatenation operator. Finally, given two strings s and t, we denote by $\mathtt{sim}_\omega(s,t)$ the *similarity* (or the score of an *optimal alignment*) between s and t under a scoring function $\omega : \Sigma \times \Sigma \to \mathcal{R}$ [9].

With the previous definitions in mind, the SAP is defined as follows [3]:

Spliced Alignment Problem (SAP): Given a subject sequence g, a target sequence t and an ordered set of substrings $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ of g, find a chain Γ of $\mathcal{B}$ such that $\mathtt{sim}_\omega(\Gamma^\bullet, t)$ is maximum among all chains of $\mathcal{B}$.

An instance of the SAP and its solution can be seen in Figure 1.

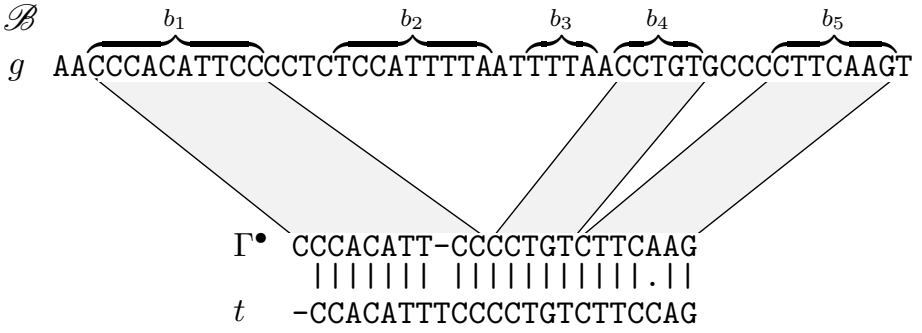

Fig. 1. An instance of the SAP and its solution. The symbols of g that compose each substring $b \in \mathcal{B}$ are disposed below its corresponding horizontal brace. The scoring function used in this instance is $\omega(a,b) = \{1, \text{ if } a = b; -1, \text{ if } a \neq b; -2 \text{ if } a = - \text{ or } b = -\}$ and its solution is $\Gamma = \{b_1, b_4, b_5\}$. Figure adapted from [5].

Looking at the SAP in the context of the gene prediction task, g could be interpreted as a (fragment of a) genomic sequence encoding a gene of interest, t as a transcript sequence related to this gene and $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ as a set of potential exons of g. With these relations in mind, and given the fact that the coding regions of a gene are less susceptible to mutations than the non-coding ones, it is very likely that a solution for the SAP will include the exons of the gene being searched for.

In [3], Gelfand *et al.* propose a polynomial time dynamic programming algorithm for the SAP. To understand the main recurrence of this algorithm, consider the following definitions taken from [3]. Let $b_k = g_l \dots g_i \dots g_m$ be a substring of g containing a position i. The i-prefix of b_k is defined as $b_k(i) = g_l \dots g_i$. Let $\Gamma = \{b_1, b_2, \dots, b_k, \dots, b_t\}$ be a chain such that some substring b_k contains position i and let $\Gamma^\bullet(i) = b_1 \bullet b_2 \bullet \dots \bullet b_k(i)$. The algorithm presented in [3] efficiently calculates a three-dimensional matrix S such that

$$S[i][j][k] = \max_{\text{all chains } \Gamma \text{ containing substring } b_k} \texttt{sim}_\omega(\Gamma^\bullet(i), t[1..j]).$$

After computing S, the value of the optimal solution can be found as

$$\max_{1 \leq k' \leq k} S[last(b_{k'})][|t|][k'].$$

Finally, it is possible to build the optimal solution itself considering the choices the algorithm made to compute its value. Using the dynamic programming technique, this algorithm for the SAP runs in time $\mathcal{O}(mnc+mk^2)$ and space $\mathcal{O}(mnc)$, with $m = |t|$, $n = |g|$, $k = |\mathcal{B}|$ and $c = \frac{1}{n}\sum_{b_i \in \mathcal{B}} |b_i|$.

As we can see, the SAP was originally proposed as a maximization problem. However, we can address it as a minimization problem as well. For this matter, we need to make use of the concept of distance between two strings, instead of similarity. To calculate the distance between two strings, we assign costs to the basic edit operations (insertion, deletion and substitution) and find the least costly series of such operations that transforms one string into the other.

The similarity, as being by definition the score of an optimal alignment, usually assumes a scoring function that rewards matches and penalizes mismatches and spaces in an alignment. The distance measure, on the other hand, requires a specific class of scoring functions, namely *metrics*. If $\omega : \Sigma \times \Sigma \rightarrow \mathcal{R}$ is a metric, then the following three properties hold:

1. $\omega(x, x) = 0$ for all $x \in \Sigma$ and $\omega(x, y) > 0$ for $x \neq y$;
2. $\omega(x, y) = \omega(y, x)$ for all $x, y \in \Sigma$;
3. $\omega(x, y) \leq \omega(x, z) + \omega(z, y)$ for all $x, y, z \in \Sigma$.

In summary, the first property assures that the costs of the basic edit operations are positive. The second property establishes that ω is symmetric. The last and most important property is called *triangle inequality*. It assures that the cost of transforming a symbol x into another symbol y is not greater than the cost of transforming x into z and then z into y. This property can be extended to sequences as a whole.

Given a metric ω and two sequences s and t, we denote by $\texttt{dist}_\omega(s, t)$ the cost, regarding ω, of the least expensive series of edit operations that transforms s into t. We can now reformulate the SAP as follows, noticing that we will refer to this version from now on:

Spliced Alignment Problem (SAP): Given a subject sequence g, a target sequence t, an ordered set of substrings $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ of g and a metric ω, find a chain Γ of $\mathcal{B}$ such that $\texttt{dist}_\omega(\Gamma^\bullet, t)$ is minimum among all chains of $\mathcal{B}$.

Kishi and Adi started exploring in [5] a variant of the SAP called *Multiple Spliced Alignment Problem*. In this variant, instead of only one target sequence t, we have a set of target sequences $\mathcal{T} = \{t_1, t_2, ..., t_u\}$ and the objective is to find a chain Γ of B such that $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^\bullet, t_i)$ is minimum among all

chains of $\mathcal{B}$. Back to the gene prediction task, the Multiple Spliced Alignment Problem is also of practical interest since now the prediction is obtained by taking more evidences into consideration, which tends to give better practical results. A formal definition of the Multiple Spliced Alignment Problem can be found below:

Multiple Spliced Alignment Problem (MSAP): Given a subject sequence g, a set of target sequences $\mathcal{T} = \{t_1, t_2, ..., t_u\}$, an ordered set of substrings $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ of g and a metric ω, find a chain Γ of $\mathcal{B}$ such that $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^\bullet, t_i)$ is minimum among all chains of $\mathcal{B}$.

An instance of the MSAP and its solution can be seen in Figure 2.

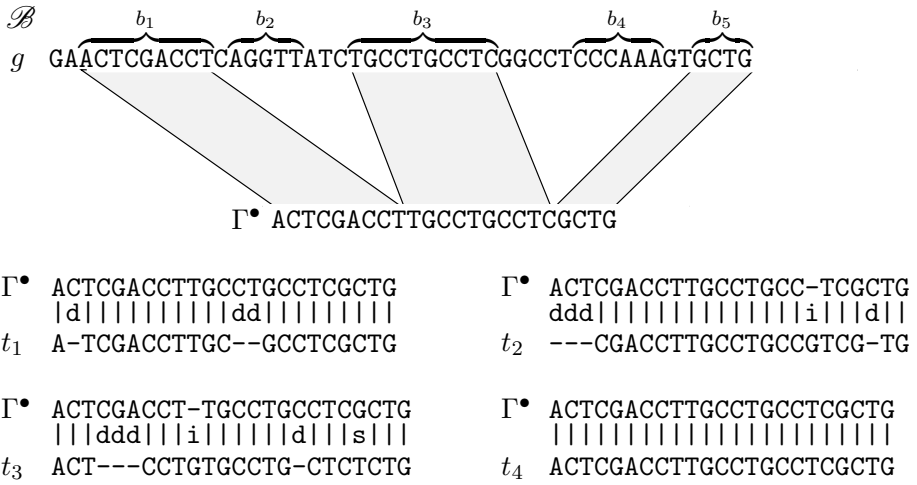

Fig. 2. An instance of the MSAP and its solution. The symbols of g that compose each substring $b \in \mathcal{B}$ are disposed below its corresponding horizontal brace and the set $\mathcal{T}$ is composed by the sequences t_1, t_2, t_3 and t_4. The metric used in this instance is the Levenshtein distance, where `d` indicates a delete operation, `i` indicates an insert operation and `s` indicates a substitution operation. The solution of this instance is $\Gamma = \{b_1, b_3, b_5\}$. Figure adapted from [5].

The MSAP was proved to be NP-complete even for binary sequences by Kishi and Adi in [5]. As a direct result of this fact, two approaches come to mind to deal with such hard problem: heuristics and approximation algorithms. As some heuristics for the MSAP were already developed in [5,6], we present in this work an approximation algorithm for it that deals in a satisfactory way with both theoretical and practical aspects of the problem.

3 A 3-Approximation Algorithm for the MSAP

The approximation algorithm developed in this work is a natural extension of the solution proposed by Gelfand *et al.* in [3] for the Spliced Alignment Problem.

It consists in finding u solutions for the SAP, one for each target sequence t_i, for $1 \leq i \leq u$, and choosing as final solution for the MSAP that chain less distant to all sequences in $\mathcal{T}$.

Algorithm 1, called MSAP-3-APP, details the idea of our approximation. In this algorithm, Γ_i is a chain of $\mathcal{B}$ returned by Gelfand's algorithm taking t_i as target sequence, and Γ corresponds to a Γ_i such that $\sum_{j=1}^{u} \texttt{dist}_\omega(\Gamma_i^\bullet, t_j)$ is minimum among all Γ_i, for $1 \leq i \leq u$.

Algorithm 1. MSAP-3-APP(g, $\mathcal{T}$, $\mathcal{B}$, ω)

Require: Subject sequence g, a set of target sequences $\mathcal{T} = \{t_1, t_2, ..., t_u\}$, a set $\mathcal{B} = \{b_1, b_2, ..., b_k\}$ of ordered substrings of g and a metric ω.
Ensure: A chain Γ of $\mathcal{B}$.

```
1.  Γ ← ∅;
2.  lower ← +∞;
3.  for i ← 1 until u do
4.     Γ_i ← Gelfand(g, t_i, B, ω); //a call to Gelfand's algorithm
5.     sum ← 0;
6.     for j ← 1 until u do
7.        sum ← sum + dist_ω(Γ_i•, t_j);
8.     end for
9.     if sum < lower then
10.       lower ← sum;
11.       Γ ← Γ_i;
12.    end if
13. end for
14. return Γ;
```

As obtaining a solution for SAP by Gelfand's algorithm (line 4 of Algorithm 1) and calculating the distance between two sequences under some metric ω (line 7 of Algorithm 1) are known tasks that can be done in polynomial time, it is easy to see that Algorithm 1 also has polynomial time complexity. More specifically, algorithm MSAP-3-APP runs in time $\mathcal{O}(um(nc + k^2 + un))$, with u, n, c and k as previously defined, and $m = \max_{1 \leq j \leq u}\{|t_j|\}$.

Now, we will show that Algorithm 1 is a 3−approximation for the MSAP. To this end, let Γ^* be an optimal solution for an instance $I = (g, \mathcal{T}, \mathcal{B}, \omega)$ of the problem, i.e. $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^{*\bullet}, t_i) = \texttt{opt}$ is minimum, and consider the following lemma:

Lemma 1. $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma_i^\bullet, t_i) \leq \sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^{*\bullet}, t_i)$

Proof. Suppose, by contradiction, that $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma_i^\bullet, t_i) > \sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^{*\bullet}, t_i)$. Then, there is some i such that $\texttt{dist}_\omega(\Gamma_i^\bullet, t_i) > \texttt{dist}_\omega(\Gamma^{*\bullet}, t_i)$. But this fact contradicts our hypothesis that $\texttt{dist}_\omega(\Gamma_i^\bullet, t_i)$ is minimum as assured by Gelfand's algorithm. ■

The relation between the value of a solution Γ computed by our algorithm, equals to $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^\bullet, t_i)$, and the value of an optimal solution Γ^* for MSAP, equals to $\sum_{i=1}^{u} \texttt{dist}_\omega(\Gamma^{*\bullet}, t_i)$, is given by Theorem 1.

Theorem 1. MSAP-3-APP *is a 3-approximation for MSAP.*

Proof. Firstly, consider the following inequality, that can be verified by the definitions of Γ and Γ_i:

$$\sum_{j=1}^{u}\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq \sum_{j=1}^{u}\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma_j^\bullet,t_i) \tag{1}$$

Given the triangular inequality property of ω, we have that $\mathtt{dist}_\omega(\Gamma_j^\bullet,t_i) \leq \mathtt{dist}_\omega(\Gamma_j^\bullet,\Gamma^{*\bullet}) + \mathtt{dist}_\omega(\Gamma^{*\bullet},t_i)$. Replacing the right side of Inequation 1 with this inequality, we get:

$$\sum_{j=1}^{u}\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq \sum_{j=1}^{u}\sum_{i=1}^{u}(\mathtt{dist}_\omega(\Gamma_j^\bullet,\Gamma^{*\bullet}) + \mathtt{dist}_\omega(\Gamma^{*\bullet},t_i))$$

$$u\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq u\sum_{j=1}^{u}\mathtt{dist}_\omega(\Gamma_j^\bullet,\Gamma^{*\bullet}) + u\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^{*\bullet},t_i) \tag{2}$$

Replacing j by i in Inequation 2, dividing its both sides by u, and making use of the equality $\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^{*\bullet},t_i) = \mathtt{opt}$, we get:

$$\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq \sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma_i^\bullet,\Gamma^{*\bullet}) + \mathtt{opt} \tag{3}$$

Using again the triangular inequality property of ω, we have that $\mathtt{dist}_\omega(\Gamma_i^\bullet,\Gamma^{*\bullet}) \leq \mathtt{dist}_\omega(\Gamma_i^\bullet,t_i) + \mathtt{dist}_\omega(t_i,\Gamma^{*\bullet})$. So, we can expand the term $\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma_i^\bullet,\Gamma^{*\bullet})$ in Inequation 3 as shown below:

$$\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq \sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma_i^\bullet,t_i) + \sum_{i=1}^{u}\mathtt{dist}_\omega(t_i,\Gamma^{*\bullet}) + \mathtt{opt}$$

Now the equality $\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^{*\bullet},t_i) = \mathtt{opt}$ can be applied again:

$$\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq \sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma_i^\bullet,t_i) + \mathtt{opt} + \mathtt{opt} \tag{4}$$

By Lemma 1, we can replace $\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma_i^\bullet,t_i)$ by $\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^{*\bullet},t_i)$ in Inequation 4:

$$\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq \sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^{*\bullet},t_i) + \mathtt{opt} + \mathtt{opt}$$

Finally, applying again the equality $\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^{*\bullet},t_i) = \mathtt{opt}$, we get:

$$\sum_{i=1}^{u}\mathtt{dist}_\omega(\Gamma^\bullet,t_i) \leq 3 * \mathtt{opt} \tag{5}$$

Therefore, the value of the solution computed by algorithm MSAP-3-APP is no worse than 3 times the value of an optimal solution for the MSAP.

■

4 Experimental Results

In order to assess the practical accuracy of our approximation, algorithm MSAP-3-APP was implemented in ANSI C++ and tested on real-world instances of the gene prediction task.

The benchmark taken to evaluate our program was the same one used by Kishi and Adi in [5], so we could compare our approach with the heuristics proposed by them. This benchmark consists of 240 fragment sequences of human DNA, obtained from the chromosomes analyzed by the ENCODE project [10]. All these fragments include only one gene and the corresponding targets were obtained by a search in the HOMOLOGENE [11] database for cDNAs sequences evolutionarily related to the genes being searched for. Finally, the ordered set of substrings for each instance was obtained by means of a HMM-based algorithm implemented by a gene prediction tool called GENSCAN [1].

To assess the accuracy of the programs, we made use of the following measures, introduced by Burset and Guigó in [2] and commonly used in the evaluation of gene prediction tools:

(1) Specificity at the nucleotide level ($Sp_n = \frac{TP}{TP+FP}$): proportion of nucleotides predicted as coding that are really coding;
(2) Sensitivity at the nucleotide level ($Sn_n = \frac{TP}{TP+FN}$): proportion of really coding nucleotides correctly predicted as coding;
(3) Specificity at the exon level ($Sp_e = \frac{NCE}{NPE}$): proportion of predicted exons that match an annotated exon;
(4) Sensitivity at the exon level ($Sn_e = \frac{NCE}{NAE}$): proportion of annotated exons that were correctly predicted.

The approximate correlation, AC, defined as

$$AC = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TP}{TP+FP} + \frac{TN}{TN+FP} + \frac{TN}{TN+FN}\right) - 1$$

has been introduced to summarize sensitivity and specificity in a single measure. At the exon level, the average $Av_e = (Sp_e + Sn_e)/2$ is used instead.

In the previous definitions, TP (true positives) is the number of really coding nucleotides correctly predicted as coding, TN (true negatives) represents the number of really non-coding nucleotides correctly predicted as non-coding, FP (false positives) is the number of really non-coding nucleotides incorrectly predicted as coding and FN (false negatives) is the number of really coding nucleotides incorrectly predicted as non-coding. On the level of complete exons, NCE is defined as the number of correctly predicted exons, NPE as the number of predicted exons and NAE as the number of annotated exons. Here, a predict exon is considered as correctly predicted when its start and end positions match the start and end positions of an annotated exon of the input sequence.

Table 1 summarizes the results obtained by our approach and by the heuristics proposed in [5, 6] on the detailed benchmark. In this table, each column stores the average values of Sn, Sp, AC and Av_e.

Table 1. Results obtained on 240 real-world instances of the gene prediction task

Approach	Nucleotide			Exon		
	Sn_n	Sp_n	AC	Sn_e	Sp_e	Av_e
MSAP-3-APP	0.95	0.96	0.95	0.85	0.81	0.83
Heuristic H1	0.96	0.96	0.95	0.86	0.81	0.83
Heuristic H2	0.96	0.91	0.93	0.83	0.73	0.78
Heuristic H3	0.93	0.96	0.94	0.86	0.84	0.85
Heuristic H4	0.77	0.80	0.77	0.54	0.51	0.53

The values in Table 1 show that our approach presented a good level of sensitivity and specificity on both nucleotide and exon levels. From all the nucleotides predicted as coding by our approximation, 96% are in fact coding. Furthermore, our approach correctly identified 95% of the coding nucleotides. At the exon level, 81% of the predicted exons match an annotated exon, and 85% of the annotated exons were correctly identified by our program.

Obviously, the accuracy of our approach in identifying the correct exon-intron structure of a gene is strongly dependent on the input set $\mathcal{B}$. If this set includes all the annotated exons of the target gene, it is very likely that all of them will be included in the chain returned by our approximation. From a total of 1677 annotated exons, 1550 were included in the sets of candidate exons and only 67 of them were missed by our approach. On the other hand, if an annotated exon is not included in the input set $\mathcal{B}$, it will be missed by our approach. From a total of 1677 annotated exons, 127 were missed by our approach since they could not be found in $\mathcal{B}$.

In comparison with the four heuristics developed so far for the MSAP, our 3-approximation algorithm achieved results comparable to all of them. It outperformed heuristic H4 in all measures, and performed very close to the other three heuristics. At the nucleotide level, for example, our approximation was slightly less sensitive than heuristics H1 and H2, but its value of specificity compares with that obtained by H1 and H3. In summary, looking at the AC column, our algorithm and Heuristic H1 were the approaches with the best values. At the exon level, our approximation outperformed H2, achieved results comparable to H1 and was overwhelmed only by H3. Anyway, in this last case, H3 outperformed our approach with only 1% and 3% of improvement in sensitivity and specificity, respectively.

5 Discussion

In this work we presented a 3-approximation algorithm for the Multiple Spliced Alignment Problem, a combinatorial optimization problem directly related with to gene prediction task. We also compared our approach with 4 previously proposed heuristics for the MSAP, achieving results comparable to the best one. This fact is very encouraging since it shows that our approach can perform as good as previously proposed heuristics for the MSAP when applied to the gene prediction task, beside ensuring its results are not worse than 3 times the optimal solution, no matter which instance is considered.

In a more detailed observation, and taking into account the measures AC and Av_e that summarize the experimental results at the nucleotide and exon levels respectively, our algorithm showed the same accuracy of Heuristic H1, being the best on nucleotide level and the second best in exon level. As Heuristic H1 is based on the idea of choosing a central sequence of $\mathcal{T}$, applying it to obtain a SAP solution with Gelfand's algorithm and extending it to the MSAP in question, it becomes clear that both approaches share similar aspects and therefore such close results are expected.

In further studies, we intend to handle the MSAP by proposing a linear programming model in order to attack it from a third perspective. We already have a preliminary integer linear programming formulation, and experimental tests with it are in course.

References

1. Burge, C., Karlin, S.: Prediction of Complete Gene Structures in Human Genomic DNA. Journal of Molecular Biology 268(1), 78–94 (1997)
2. Burset, M., Guigo, R.: Evaluation of Gene Structure Prediction Programs. Genomics 34(298), 353–367 (1996)
3. Gelfand, M.S., Mironov, A.A., Pevzner, P.A.: Gene Recognition Via Spliced Sequence Alignment. Proceedings of the National Academy of Sciences of the United States of America 93, 9061–9066 (1996)
4. Hogeweg, P.: The Roots of Bioinformatics in Theoretical Biology. PLoS Computational Biology 7(3), 1–5 (2011)
5. Kishi, R.M., dos Santos, R.F., Adi, S.S.: Gene Prediction by Multiple Spliced Alignment. In: Norberto de Souza, O., Telles, G.P., Palakal, M. (eds.) BSB 2011. LNCS, vol. 6832, pp. 26–33. Springer, Heidelberg (2011)
6. Kishi, R.M., dos Santos, R.F., Montera, L., Adi, S.S.: A Similarity-based Genetic Algorithm for the Gene Prediction Problem. In: BSB & EBB Digital Proceedings, Campo Grande, pp. 84–89 (2012)
7. Majoros, W.H.: Methods for Computational Gene Prediction, 1st edn. Cambridge University Press (2007)
8. Mathé, C., Sagot, M.-F., Schiex, T., Rouzé, P.: Current Methods of Gene Prediction, Their Strengths and Weaknesses. Nucleic Acids Research 30(19), 4103–4117 (2002)
9. Needleman, S.B., Wunsch, C.D.: A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. Journal of Molecular Biology 48, 443–453 (1970)
10. The ENCODE Project Consortium: The ENCODE (Encyclopedia of DNA Elements) Project. Science 306(5696), 636–640 (2004)
11. Sayers, E.W., Barrett, T., Benson, D.A., Bolton, E., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., DiCuccio, M., Federhen, S., Feolo, M., Fingerman, I.M., Geer, L.Y., Helmberg, W., Kapustin, Y., Krasnov, S., Landsman, D., Lipman, D.J., Lu, Z., Madden, T.L., Madej, T., Maglott, D.R., Marchler-Bauer, A., Miller, V., Karsch-Mizrachi, I., Ostell, J., Panchenko, A., Phan, L., Pruitt, K.D., Schuler, G.D., Sequeira, E., Sherry, S.T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T.A., Wagner, L., Wang, Y., Wilbur, W.J., Yaschenko, E., Ye, J.: Database resources of the National Center for Biotechnology Information. Nucleic Acids Research 40 (D1), D13–D25 (2012)