EFFICIENT PREDICTION OF ALTERNATIVE SPLICE FORMS USING PROTEIN DOMAIN HOMOLOGY

Michael Hiller[a⋆] Rolf Backofen[a⋆] Stephan Heymann[b⋆] Anke Busch[a]
Timo Mika Gläßer[b] Johann-Christoph Freytag[b]

[a] *Friedrich-Schiller-Universität Jena, Institute of Computer Science*
Chair for Bioinformatics, Ernst-Abbe-Platz 1-4, D-07743 Jena, Germany,
***Email**: {hiller,backofen,busch}@inf.uni-jena.de*

[b] *Humbold-Universität zu Berlin, Institute of Computer Science*
Unter den Linden 6, D-10099 Berlin, Germany,
***Email**: {heymann,glaesser,freytag}@dbis.informatik.hu-berlin.de*

[⋆] *These authors contribute equally to this work.*

Abstract

Alternative splicing can yield manifold different mature mRNAs from one precursor. New findings indicate that alternative splicing occurs much more often than previously assumed. A major goal of functional genomics lies in elucidating and characterizing the entire spectrum of alternative splice forms.

Existing approaches such as EST-alignments focus only on the mRNA sequence to detect alternative splice forms. They do not consider function and characteristics of the resulting proteins. One important example of such functional characterization is homology to a known protein domain family. A powerful description of protein domains are profile Hidden Markov models (HMM) as stored in the Pfam database.

In this paper we address the problem of identifying the splice form with the highest similarity to a protein domain family. Therefore, we take into consideration all possible splice forms. As demonstrated here for a number of genes, this homology based approach can be used successfully for predicting partial gene structures. Furthermore, we present some novel splice form predictions with high-scoring protein domain homology and point out that the detection of splice form specific protein domains helps to answer questions concerning hereditary diseases.

Simple approaches based on a BLASTP search cannot be applied here, since the number of possible splice forms increases exponentially with the number of exons. To this end, we have developed an efficient polynomial-time algorithm, called ASFPred (Alternative Splice Form Prediction). This algorithm needs only a set of exons as input.

Keywords: alternative splicing, novel splice forms, Pfam, protein domain, Viterbi algorithm, profile HMM, gene prediction

1 Introduction

The majority of eukaryotic pre-mRNAs requires the splicing process as an important step in maturation. Splicing removes introns, yielding a mature mRNA whose coding sequence can be translated into a protein sequence. Recent studies estimate that a big portion (up to 60 %) of human genes produces alternative splice forms [1]. Despite this high estimate, alternative splice variants have been detected only for a minor part of human genes to date.

This is because we do not yet fully understand the regulatory processes behind alternative splicing (for review see [2,3]). Additionally, splice form spectra vary considerably depending on cell type, tissue and developmental stage. That is why it is difficult to elucidate the complete set of splice forms of a gene. This difficulty became apparent again recently in a paper by Xu and Lee [4]. They showed that for a number of cancer-associated genes only the cancer specific splice form is known, although another predominant splice variant exist in normal tissue.

In the widespread field of biomedical research numerous aberrant splice events have been reported to be responsible for pathological phenotypes[5,6,7,8,9]. Here, it is crucial to find the possible splice form inherent functions of a gene in order to guide the search for the disease causing splice variant that could be a therapeutic target. Hence, it remains a challenging task to identify alternative splice variants and their coded proteins, both from the view of functional genomics and biomedical research perspective.

Prediction of Alternative Splice Forms Concerning in silico prediction of alternative splice forms, the majority of methods is based on EST-clustering [10,4,11]. This is a very successful approach if there is sufficient EST-coverage, which is not the case for all genes. Moreover, it is hard to detect highly specific as well as low copy number splice variants by EST-based approaches (see [1] for review) or microarray technologies [12,13] since a huge amount of different tissues at different states have to be analysed. Most of the other approaches are working on the genomic level by predicting splice sites and introns/exons using sequence signals and composition differences [14,15]. These methods are more devoted to gene prediction and usually yield only one optimal gene structure. This limits their application for alternative splice form prediction, since arbitrary assembly of suboptimal, inframe exons results in a large number of false positives.

Hence, our aim is to include functional information provided by protein domain descriptions. It is assumed that 70-88 % of alternative splice forms alter the protein product [1,16]. Moreover, splice forms found in EST-based approaches can arise from rare splice errors. These alternative splice variants are more likely to be detected with increasing EST coverage, but they are not considered to be functional[17]. Thus, an additional investigation of splice forms on the protein level is extremely valuable.

Since protein domain coding exons are often disrupted by introns, one

must consider concatenations of exons in order to be able to fully use protein domain information. A statistical analysis in a recent paper [18] showed that a considerable fraction of alternative splicing (28%) concerns domain coding exons, thus changing the domain structure of the protein.

The basic idea of our approach is to take a set of possible exons and splice sites and to search among all exon concatenations for the one whose coded protein demonstrates the highest homology to a protein domain. The exon set can be generated by a gene prediction program or taken from a database. In doing so, we consider **all** possible concatenations and **all** possible protein domains contained in the Pfam database [19]. As the Pfam database covers the majority of Swissprot proteins [19], we expect the majority of genes to have Pfam homologues.

The scope of this approach is twofold. Firstly, it can be used to improve predicted gene structures. Secondly, we use this method to predict alternative splice forms coding novel functional domains currently not annotated for this gene. Since our approach is independent from ESTs, there is no restriction to genes with sufficient EST coverage. Furthermore, prediction of splice variants coding functional domains often indicate in which tissues it might be expressed, which facilitates verification.

Although there are programs that use protein information for gene prediction [15,20,21], these either expect one known homologous protein or they use BLAST in order to find homologous regions. In our approach fast heuristic search methods like BLASTP cannot be applied, since an n-exonic gene can produce 2^n possible splice forms by systematic exon skipping. A simple generate and test approach would have to check 4,294,967,296 different sequences for the 32-exonic human DSCAM gene. Thus, the computational problem is to cope with an exponential number of sequences. Furthermore, our approach aims at concatenating several exons to yield a homology hit and does not rely on independent local hits from a BLAST search.

Description of the Approach Given a set of exons we analyse the complete set of possible splice forms to detect those having high similarity to protein domain families. Protein domains stored in the Pfam database are described by profile HMMs [22,23]. We present here an approach, called ASFPred, that computes in polynomial runtime the splice form that has the highest similarity to an HMM. This algorithm is an extension of the classical Viterbi algorithm [24] which is a variant of dynamic programming (DP). The crucial extension is to allow for exon skipping when calculating the dynamic programming matrix. This means, on the left boundary of exon i we include all skipping events that skip previous exons $j, \ldots, i-1$ for all $j \leq i-1$. The sophisticated part of the algorithm was the necessity to handle frameshifts occurring during the skip process, thus ensuring the existence of an open reading frame (ORF).

Current knowledge declares one or more annotated exons per gene as constitutive ones [3], i.e. they are part of all mature transcript variants. This

a priori classification can be questioned because any exon can be considered as constitutive only as long as no alternative splice form lacking this particular exon is found. Although it would be easy to handle certain exons as constitutive ones in our algorithm, we do not allow for constitutive exons here, since this would mean a restriction to known splice forms.

Plan of the Paper. In the next section we present our algorithm. For an easier understanding of our algorithm we first consider a nucleotide-based HMM. Eventually we will use an HMM on protein level. In the last two sections, we present some encouraging results of our approach and conclude with a discussion.

2 Algorithm

2.1 Nucleotide-level HMM target

For aligning a sequence to an HMM the preferred method (e.g. in the HMMER package [25]) is to compute the path through the HMM having the highest score. This is usually done by means of the Viterbi algorithm [24]. In our case there is an exponential number of splice form sequences that have to be compared with the HMM, so that pair comparison is not feasible. By turning the problem into an optimization problem, we were able to develop an extended Viterbi algorithm to solve this problem in polynomial runtime. Instead of considering all possible splice forms explicitly we search only for the one showing the highest similarity to a given HMM.

Let us formalize the problem. Given a gene G with n exons $e_1, \ldots, e_n$ and an HMM H, we then denote the binary vector $s = (s_1, \ldots, s_n)$, where s_i is 1 if exon i is expressed and 0 if exon i is skipped, as a *splice form*. Furthermore, $splice = \{s \mid s = (s_1, \ldots, s_n) \quad and \quad s_i \in \{0, 1\}\}$ is the set of all possible splice forms. Let $DNA(s)$ be the concatenated DNA sequence of all expressed exon sequences in s and $Sc(H, DNA(s))$ the Viterbi log-odds score of sequence $DNA(s)$ and HMM H. Our algorithm computes the splice form that maximises the Viterbi score $Sc(H, DNA(s))$, that is $s_{max} = argmax_{s \in splice}\{Sc(H, DNA(s))\}$.

The basic idea is to include exon skipping during the calculation of the dynamic programming matrix. Since an HMM can be divided into emitting and silent states, we have to determine which states allow for exon skipping. Clearly, exon skipping has to be handled for all emitting states. But what about silent states? A silent state always has an emitting state as (indirect) predecessor, where the current character is emitted. Hence, we can use standard recursions for silent states and only extend the equations for emitting states so that exon skipping is included.

Let $V_j(i)$ be the log-odds score of the best path through the HMM ending at state j with sequence position i. We denote the log-odds score that state x emits character y by $E_x(y)$, and the log-odds score for the transition from state x to state y by $A_{x,y}$. We write $P(x)$ for the set of predecessors of x, i.e.

all states that have a direct transition to state x. Let $\mathcal{S}$ be the concatenated DNA sequence of all exons of G.

Let $\mathcal{B}_l = \{b_2^l, ..., b_n^l\}$ be the set of left boundaries for exon 2 to n, where b_i^l is the position of the first base of exon i in $\mathcal{S}$. Let $\mathcal{B}_r = \{b_1^r, ..., b_{n-1}^r\}$ be the set of right boundaries for exon 1 to $n-1$, where b_i^r is the position of the last base of exon i in $\mathcal{S}$. These two sets correspond exactly to the set of splice signals. The algorithm requires $\mathcal{S}$, $\mathcal{B}_l$ and $\mathcal{B}_r$ as input.

The recursion equation for the extended Viterbi algorithm is:

$$V_j(i) = \begin{cases} E_j(\mathcal{S}[i]) + back(j, i-1) & \text{if } i \notin \mathcal{B}_l \text{ and } j \text{ emitting} \\ E_j(\mathcal{S}[i]) + \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i}} back(j, l) & \text{if } i \in \mathcal{B}_l \text{ and } j \text{ emitting} \\ back(j, i) & j \text{ silent} \end{cases}$$

where $back(j, i) = \max\limits_{p \in P(j)} \{V_p(i) + A_{p,j}\}$.

Note that the definition of $V_j(i)$ implies only that sequence position i is reached, not that the complete subsequence $\mathcal{S}[1 \dots i]$ is emitted. So $V_j(i)$ gives the score for the best subalignment ending with state j of the best concatenation of upstream exons up to position i.

Since this algorithm guarantees that only disjoint sequence parts (bounded by elements from $\mathcal{B}_l$ and $\mathcal{B}_r$) are concatenated, it finds the best non-overlapping concatenation regardless whether the given exon set contains overlapping exons or not. This means that $\mathcal{B}_l$ and $\mathcal{B}_r$ can comprise alternative 5' and 3' ends of exons, respectively, to allow for alternative 5' and 3' splicing. Moreover, $\mathcal{B}_l$ and $\mathcal{B}_r$ can also contain splice sites predicted by a computational splice site finder. A graphical illustration is given in fig. 1. Thus, our algorithm does not rely on one fixed exon-intron structure, which is often incomplete or erroneous. Furthermore, we have the possibility to find undetected exonic sequences and to expose novel splice sites.

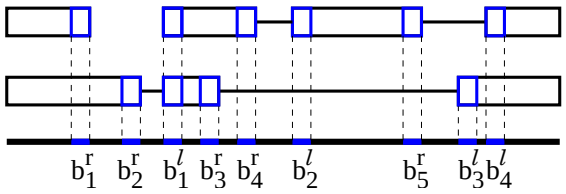

Figure 1: Illustration of the left and right boundaries: Shown are two exon structures that may be generated by two different gene predictors or may be transcripts taken form a database. Since the left boundary of the first exon and the right boundary of the last exon does not correspond to a splice site, they are not contained in $\mathcal{B}_l$ and $\mathcal{B}_r$. The boundaries in $\mathcal{B}_l$ and $\mathcal{B}_r$ are shown in blue.

2.2 *Protein-level HMM target*

The last section showed how to compute efficiently the most similar exon concatenate to an HMM on nucleotide level. We now describe how the algorithm can be modified for an HMM on amino acid level so that frameshifts as well as proper ORFs can be handled simultaneously. Since not all exon lengths are multiples of three nucleotides, frameshifts occur during exon skipping.

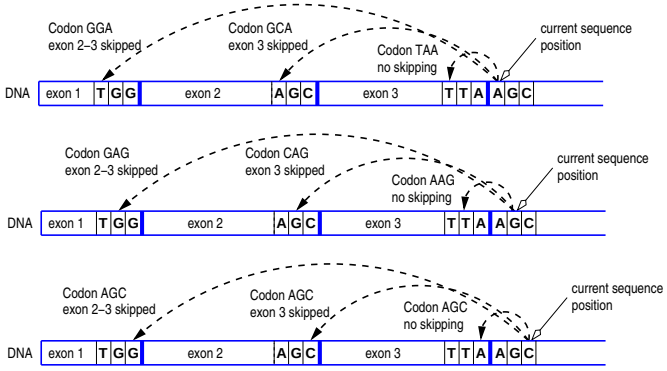

Figure 2: Scheme of the basic idea of the algorithm shown for positions with maximal distance of 3 to the left exon boundary. The current position determines which nucleotides the splice junction codon comprises. Note that different codons arise from different exon skipping events. While computing the DP matrix from left to right, already precomputed entries (to the left) are accessed for the computation of each entry. The arrows indicate the sequence position where the DP matrix is accessed during recursion.

According to section 2.1, now the problem is the computation of $s_{max} = argmax_{s \in splice}\{Sc(H, AA(s))\}$, where $AA(s)$ is the translated DNA sequence $DNA(s)$. To switch to protein level we consider the current sequence position in $\mathcal{S}$ the third codon position and translate the codon consisting of the current and the 2 previous DNA bases. Then the step length is set to 3, i.e. we access $V_j(i-3)$ when computing $V_j(i)$. Since frameshifts can occur during exon skipping, we have to extend the Viterbi algorithm in order to include all 3 reading frames. Hence, exon skipping is allowed if the current sequence position is not more than 3 DNA bases away from a left exon boundary. Figure 2 illustrates the idea of the algorithm. It shows that each skipping variant can lead to a different codon and, thus, to a different amino acid.

The protein domain information is taken from the Pfam database [19]. Thus, we will specify the recursion equations of the extended Viterbi algorithm for the plan7 architecture of a profile HMM [23] because this is the architecture of all Pfam-HMMs. Of course, the algorithm is not restricted to plan7. Plan7 means that direct transitions between insert and delete states

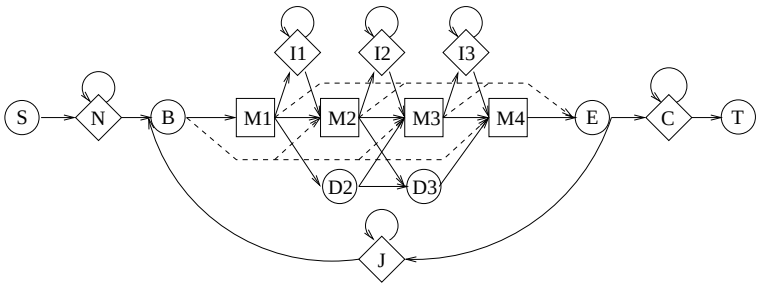

Figure 3: Architecture of a four match state plan7 profile HMM. Match states are shown as squares, insert states as diamonds and silent states as circles. The dashed lines indicate direct entry and exit transitions. Note that the architecture demands that at least one character be emitted in each iteration $B \to M_i \to \ldots \to E \to J \to B$.

and vice versa are not allowed (see fig. 3). The main model is separated by two silent states (B- and E-state). Furthermore, there are three special insert states: one before the main model (N-state), one after the main model (C-state) and one allowing multiple iterations through the main model (J-state). Note that the special insert states only emit characters on the loop transition. Moreover, there are direct entry transitions from B-state to any match state as well as direct exit transitions from any match state to E-state (dashed lines in fig. 3).

Let H be a plan7 profile HMM with m match states, $m-1$ insert and $m-2$ delete states. According to the notation in [26], $V_j^M(i)$ is the log-odds score of the best path through the HMM ending with match state j with sequence position i. Similarly, $V_j^I(i)$, $V_j^D(i)$ are defined for insert and delete states and $V^X(i)$ for the special states, where $X \in \{S, N, B, J, E, C, T\}$.

With $\mathcal{B}_l \oplus 1$ (resp. $\mathcal{B}_l \oplus 2$) we denote the set $\{b_2^l + 1, ..., b_n^l + 1\}$ (resp. $\{b_2^l + 2, ..., b_n^l + 2\}$). Furthermore, we write $\text{codon}_{i,j,k}^{\mathcal{S}}$ for the amino acid that corresponds to the translated codon $\mathcal{S}[i]\mathcal{S}[j]\mathcal{S}[k]$. Hence, we get the following equation for the M-states.

$$V_j^M(i) = \max \begin{cases} \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i}} \left\{ E_{M_j}(\text{codon}_{\ell-1,\ell,i}^{\mathcal{S}}) + \text{back}^M(j, \ell-2) \right\} & \text{if } i \in \mathcal{B}_l \\ \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i-1}} \left\{ E_{M_j}(\text{codon}_{\ell,i-1,i}^{\mathcal{S}}) + \text{back}^M(j, \ell-1) \right\} & \text{if } i \in \mathcal{B}_l \oplus 1 \\ \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i-2}} \left\{ E_{M_j}(\text{codon}_{i-2,i-1,i}^{\mathcal{S}}) + \text{back}^M(j, \ell) \right\} & \text{if } i \in \mathcal{B}_l \oplus 2 \\ E_{M_j}(\text{codon}_{i-2,i-1,i}^{\mathcal{S}}) + \text{back}^M(j, i-3) & \text{otherwise} \end{cases}$$

$$\text{where back}^M(j,\ell) = \max \begin{cases} V^M_{j-1}(\ell) + A_{M_{j-1},M_j} \\ V^I_{j-1}(\ell) + A_{I_{j-1},M_j} \\ V^D_{j-1}(\ell) + A_{D_{j-1},M_j} \\ V^B(\ell) \ + A_{B,M_j} \end{cases}$$

The recursion equation for the I-states is

$$V^I_j(i) = \max \begin{cases} \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i}} \left\{ E_{I_j}(\text{codon}^{\mathcal{S}}_{\ell-1,\ell,i}) + \text{back}^I(j, \ell-2) \right\} & \text{if } i \in \mathcal{B}_l \\ \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i-1}} \left\{ E_{I_j}(\text{codon}^{\mathcal{S}}_{\ell,i-1,i}) + \text{back}^I(j, \ell-1) \right\} & \text{if } i \in \mathcal{B}_l \oplus 1 \\ \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i-2}} \left\{ E_{I_j}(\text{codon}^{\mathcal{S}}_{i-2,i-1,i}) + \text{back}^I(j, \ell) \right\} & \text{if } i \in \mathcal{B}_l \oplus 2 \\ E_{I_j}(\text{codon}^{\mathcal{S}}_{i-2,i-1,i}) + \text{back}^I(j, i-3) & \text{otherwise} \end{cases}$$

$$\text{where back}^I(j,\ell) = \max \begin{cases} V^M_j(\ell) + A_{M_j,I_j} \\ V^I_j(\ell) \ + A_{I_j,I_j} \end{cases}$$

The recursion equation for the special insert state C is a little tricky, since C acts as both a silent *and* a non-silent state. Characters are only emitted via the loop transition, so exon skipping will only be handled for the $C \to C$ transition. This yields the following equation:

$$V^C(i) = \max \begin{cases} \max \begin{cases} \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i}} \left\{ E_C(\text{codon}^{\mathcal{S}}_{\ell-1,\ell,i}) + \text{back}^C(\ell-2) \right\} & \text{if } i \in \mathcal{B}_l \\ \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i-1}} \left\{ E_C(\text{codon}^{\mathcal{S}}_{\ell,i-1,i}) + \text{back}^C(\ell-1) \right\} & \text{if } i \in \mathcal{B}_l \oplus 1 \\ \max\limits_{\substack{\ell \in \mathcal{B}_r, \\ \ell < i-2}} \left\{ E_C(\text{codon}^{\mathcal{S}}_{i-2,i-1,i}) + \text{back}^C(\ell) \right\} & \text{if } i \in \mathcal{B}_l \oplus 2 \\ E_C(\text{codon}^{\mathcal{S}}_{i-2,i-1,i}) + \text{back}^C(i-3) & \text{otherwise} \end{cases} \\ V^E(i) + A_{E,C} \end{cases}$$

$$\text{where back}^C(\ell) = V^C(\ell) + A_{C,C}$$

The equations for N-state and J-state are similar to the C-state equation and are not shown. Just for completeness, we show the recursions for the silent states:

$$
\begin{aligned}
V_j^D(i) &= \max \begin{cases} V_{j-1}^M(i) + A_{M_{j-1},D_j} \\ V_{j-1}^D(i) + A_{D_{j-1},D_j} \end{cases} \\
V^B(i) &= \max \begin{cases} V^N(i) + A_{N,B} \\ V^J(i) + A_{J,B} \end{cases} \\
V^E(i) &= \max_{j=1,\ldots,m} \{V_j^M(i) + A_{M_j,E}\} \\
V^T(i) &= V^C(i) + A_{C,T}
\end{aligned}
$$

Of course, not all possible splice forms will form an ORF, since some of them might lack a start and/or stop codon. The start and stop codon condition can easily be included in the algorithm. Since matrix entries for the start-state are not computed but initialised, we set all of them to $-\infty$ except for the positions where a start codon begins.

$$
V^S(i) = \begin{cases} 0 & \text{if} \quad \mathcal{S}[i+1]\mathcal{S}[i+2]\mathcal{S}[i+3] = \text{ATG} \\ -\infty & \text{otherwise} \end{cases}
$$

To allow for stop codons, we define

$$
E_x(\text{codon}_{i,j,k}^{\mathcal{S}}) = \begin{cases} -\infty & \text{if} \quad \mathcal{S}[i]\mathcal{S}[j]\mathcal{S}[k] \in \{\text{TGA,TAA,TAG}\} \\ E_x(\text{codon}_{i,j,k}^{\mathcal{S}}) & \text{otherwise} \end{cases}
$$

for all non-silent states x. Note that, although alternative starts of the first exon and ends of the last exon are not contained in $\mathcal{B}_l$ and $\mathcal{B}_r$, the consideration of all start and stop codons addresses this implicitly. To compute the highest Viterbi score $Sc_{max}(H, AA(s))$ we have to heed the stop codon condition and that a stop codon can be assembled on exon boundaries. Therefore, during the dynamic programming procedure we compute all positions after which a stop codon occurs and denote this set as $EndPos$

$$
EndPos = \left\{ \begin{array}{lcl}
\ell - 2 & | & \mathcal{S}[\ell-1]\mathcal{S}[\ell]\mathcal{S}[i] \in \{\text{TGA,TAA,TAG}\} : \\
 & & \quad \ell \in \mathcal{B}_r,\, \ell < i,\, i \in \mathcal{B}_l \\
\ell - 1 & | & \mathcal{S}[\ell]\mathcal{S}[i-1]\mathcal{S}[i] \in \{\text{TGA,TAA,TAG}\} : \\
 & & \quad \ell \in \mathcal{B}_r,\, \ell < i-1,\, i \in \mathcal{B}_l \oplus 1 \\
\ell & | & \mathcal{S}[i-2]\mathcal{S}[i-1]\mathcal{S}[i] \in \{\text{TGA,TAA,TAG}\} : \\
 & & \quad \ell \in \mathcal{B}_r,\, \ell < i-2,\, i \in \mathcal{B}_l \oplus 2 \\
i - 3 & | & \mathcal{S}[i-2]\mathcal{S}[i-1]\mathcal{S}[i] \in \{\text{TGA,TAA,TAG}\} : \\
 & & \quad i \notin \mathcal{B}_l \cup \mathcal{B}_l \oplus 1 \cup \mathcal{B}_l \oplus 2
\end{array} \right\}
$$

Then $Sc_{max}(H, AA(s))$ is given by

$$
Sc_{max}(H, AA(s)) = \max\{V^T(i) \,|\, i \in EndPos\}.
$$

A normal traceback determines s_{max}. Backtracking from other positions in $EndPos$ and choosing suboptimal paths on a traceback can be used to find suboptimal splice forms.

Runtime The runtime of the algorithm is as follows. Let M be the number of states in the HMM and k be the length of $\mathcal{S}$. For plan7 profile HMMs $M = m + (m-1) + (m-2) + 7$. The number of direct predecessors $(max_i\{|P(i)|\})$ is bounded for profile HMMs. There are $k-3(n-1)$ sequence positions that do not allow for skipping, resulting in $O(M \cdot k)$ runtime for those columns of the matrix. One of the remaining columns, e.g. column b_i^l, needs $O(M \cdot (i-1))$ runtime because of the $i-1$ possible skipping events. This results in $O(\sum_{i=2}^{n} M \cdot (i-1)) = O(M \sum_{i=1}^{n-1} i) = O(\frac{M \cdot (n-1) \cdot n}{2}) = O(M \cdot n^2)$ and, therefore, $O(M \cdot k + M \cdot n^2)$ for the complete algorithm. Since the matrix $V_j(i)$ is two dimensional, $O(M \cdot k)$ space is needed.

3 Results

In this section we first show that our algorithm can be successfully applied to the prediction of partial gene structures. After that we present interesting examples of novel splice form specific domains found in our approach.

Prediction of Partial Gene Structures We downloaded genomic sequences (+ 5000 nt flanking both sides) of several genes with annotated exon structure and annotated Pfam domains from Ensembl[27] version 17.33. Genscan[14] was used to predict gene structures in those sequences. Although Genscan is considered to be one of the most accurate gene predictors, the program sometimes predicts wrong exons or exon boundaries and misses some exons. We observed that for incorrect predictions the annotated exons are often contained in the suboptimal predictions. Therefore, we let Genscan output all suboptimal exons.

The boundaries of all optimal and suboptimal exons were put into ASFPred and HMMs from Pfam database release 9.0 were used to predict the exon concatenation with the best match to the Pfam domain. For a correct testing procedure, we have to assure that the Ensembl protein is not contained in the seed alignment of a Pfam-HMM. Otherwise we would expect to find the annotated exon structure. Therefore, we rebuilt the HMM without this protein in such cases. Then we compared the prediction of our program with the annotated exon structure and the Genscan prediction. In the majority of cases where Genscan has missed exons or predicted wrong exons we found the correct annotated exon structure. The results are summerized in table 1. Figures 4 and 5 discusses two more complex examples in detail. In all cases where the optimal gene structure of Genscan contains all annotated exons belonging to one domain, ASFPred confirmed the Genscan prediction. These cases are not shown in table 1.

In few cases (RBM5, WRN-PF00271 in table 1 and fig. 5) we found an exon concatenation that yields a higher score than the Ensembl protein and is different from the annotated one. In all these cases, the suboptimal output of ASFPred contained the annotated exon concatenation. Of course, it is possible that these predictions with a higher score, as well as Genscan exons not being annotated, belong to alternative splice forms that are not yet discovered. Note, that our algorithm addresses only exons coding the Pfam domain. Since the input only consists of the Genscan prediction we are not able to predict an exon that is not among the suboptimal exons. Nevertheless, the results presented below show that our approach is successful in improving a predicted gene structure.

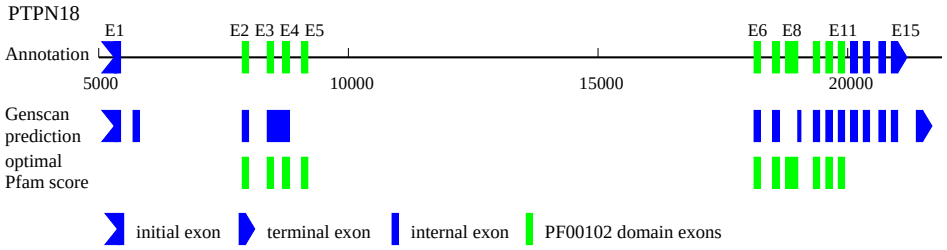

Figure 4: The annotated PF00102 domain of PTPN18 covers exon 2–11 (colored green) as shown on the top line. Genscan predicts one big exon instead of the 2 smaller exons 3 and 4, but both are contained in the suboptimals. Furthermore, it missed exon 8, but predicts a smaller downstream exon. Again annotated exon 8 is among the suboptimals. As shown on the last line, ASFPred outputs an exon structure that corresponds exactly to the annotated one. This improves the gene structure for the part that is covered by the Pfam domain.

Prediction of Alternative Splice Forms In the second part we apply this approach to the prediction of alternative splice forms. Here, we aim at detecting splice form specific protein domains that are currently not associated with the corresponding gene. To this end, we exclude all annotated Pfam domains from the search. Our aims are, firstly, to investigate the spectrum of protein domains (and thus protein functions) that can be coded by one gene and, secondly, to guide continuative experimental efforts, both in silico and in the wet lab.

We have scanned 125 arbitrary selected Ensembl-genes (version 17.33) with ASFPred. For the results discussed below Pfam database release 9.0 containing 2846 human profile HMMs was used and cut-off parameters were set to standard values as recommended in Eddy [25]. Complete mRNA – exonic and intronic – sequences were downloaded from Ensembl and known splice sites derived from known exons were added to the set of boundaries. Additionally predicted splice sites from Genesplicer [28] and Genscan [14] were put into the algorithm.

HUGO	Exons	Pfam	Genscan prediction	in suboptimals	prediction of ASFPred
FGR	11	PF00069	exon 11 beginning 15 nt upstream of correct site	exon 11 $\surd$	wrong beginning corrected, perfect prediction
LASP1	8	PF00412	wrong exon predicted (1.05)		wrong exon skipped, perfect prediction
		PF00880	2 wrong exons predicted (1.07, 1.09)		both wrong exons skipped, perfect prediction
APBA3	11	PF00640	wrong exon predicted (1.06)		wrong exon skipped, perfect prediction
ABCB1	29	PF00664	exon 18 and 24 missed	exon 18 $\surd$ exon 24 $\emptyset$	exon 18 predicted, suboptimal S.239 included that is not identical with exon 24
ATRX	35	PF00176	exon 18 missed	exon 18 $\surd$	exon 18 predicted, perfect prediction
		PF00271	wrong exon predicted (9.14)		wrong exon skipped, perfect prediction
WRN	35	PF01612	exon 3 and 4 missed, exon 5 as shortened initial exon predicted (exon end correct)	exon 3 $\surd$ exon 4 $\surd$ exon 5 $\emptyset$	exon 3 and 4 predicted, a suboptimal exon 5 with beginning 42 nt downstream of correct site but correct end and same reading frame predicted
		PF00270	exon 13 and 16 missed, exon 14 beginning 22 nt upstream of correct site, exon 19 beginning 15 nt upstream	exon 13 $\surd$ exon 14 $\surd$ exon 16 $\surd$ exon 19 $\surd$	exon 16 and 19 predicted, wrong exon 14 beginning not corrected, exon 13 not found instead upstream exon (2.03) taken (2.03 is an annotated exon but not considered to code a part of PF00270 domain)
		PF00271	exon 29 missed	exon 29 $\surd$	predict exon 20 instead of 29 since this increases Pfam-score in comparison to score from the Ensembl-protein (here only the last 4 amino acids of exon 29 contribute to the domain and the last 4 amino acids of exon 20 give a higher score)
RBM5	25	PF00076	exon 5 and 6 missed	exon 5 $\surd$ exon 6 $\surd$	exon 5 and 6 predicted, instead of annotated exon 7 an upstream suboptimal Genscan-exon taken since this results in a higher score than the Ensembl-protein

Table 1: The columns are: gene name, number of exons, Pfam accession number for annotated domains, errors in Genscan prediction, the correct exons that are among the suboptimals and prediction of ASFPred. If annotated exons are among suboptimals a $\surd$ is shown otherwise a $\emptyset$. The numbers in brackets correspond to the Genscan output (1.05 for exon in optimal gene structure, S.239 for a suboptimal exon).

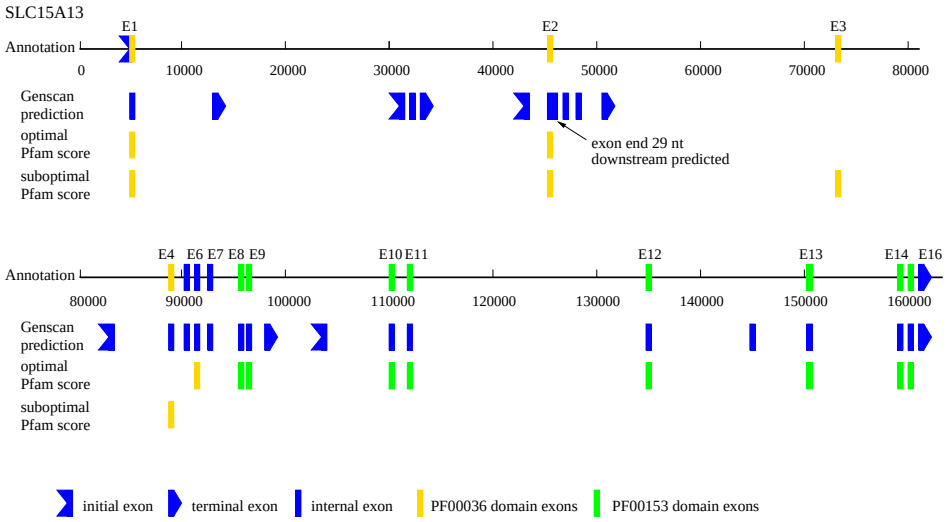

Figure 5: SLC25A13 has two annotated domains. PF00036 (colored yellow) is coded by exon 1–4, PF00153 (green) consists of exon 8–15. Genscan predicts 4 complete genes and one partial gene at the beginning. The exon concatenation yielding the best score is shown on the third line. For PF00036 we are able to correct the wrong 3' end of exon 2. But we predict exon 6 instead of exon 3 and 4 since this increases the score from 27.8 for the Ensembl protein to 31.8. A possible alternative splice variant that skips exon 3–5 and still codes this Pfam domain cannot be excluded. The second best exon concatenation (shown on the last line) gives the correct annotated exon structure. For PF00153 the optimal exon concatenation equals the annotated structure. Here, ASFPred is not only able to skip wrongly predicted exons or correct wrong boundaries but gives a clear hint that this genomic region codes only one connected gene and not several as predicted by Genscan.

For most genes additional Pfam hits were found. Here, we present only high-scoring hits with much higher scores than the trusted-cutoff values given by the Pfam database. Those hits are shown in table 2. We are able to predict a function for some genes that have currently no Pfam annotation. For example, ENSG00000006634 yields a BRCA1 C-Terminus domain by skipping exon 4. Moreover, ENSG00000073578 is predicted to produce a splice form that replaces the C-terminal domain (PF02910) with a Integrase-domain (PF00665) by inclusion of two Genscan exons and usage of annotated exon 14. Interesting is that exon 14 (114 nt long) is used in reading frame 0 to code for PF02910 in the Ensembl transcript, while our prediction uses reading frame 1 to code a part of PF00665.

For a molecular etiology study, the 10-exonic marenostrin gene (MEFV, ENSG00000103313), about 20 mutations of which are known to be causal for Familial Mediterranean Fever (FMF, see OMIM 249100) was subjected to ASFPred. Among the new Pfam signals recorded, PF02177 and PF00050

EnsemblID	ENSG00000005059
Pfam	PF04588 (Hypoxia induced protein conserved region)
score	131
E-value	1.45e-36
splice form	13659 - 13880 (Genscan exon)
	25057 - 25173 (Ensembl exon 5)
EnsemblID	ENSG00000006634
Pfam	PF00533 (BRCA1 C Terminus (BRCT) domain)
score	29
E-value	6.73e-06
splice form	1833 - 2005 (Ensembl Exon 2)
	8759 - 8938 (Ensembl Exon 3)
	11109 - 11178 (Ensembl Exon 5)
	11763 - 16839 (Ensembl Exon 6)
EnsemblID	ENSG00000009780
Pfam	PF00834 (Ribulose-phosphate 3 epimerase family)
score	136
E-value	3.92e-38
splice form	4151 - 4252 (Ensembl exon 3)
	13914 - 14387 (Genscan exon)
	22412 - 22462 (Genesplicer acceptor)
EnsemblID	ENSG00000076258
Pfam	PF01028 (Eukaryotic DNA topoisomerase I, catalytic core)
score	126
E-value	2.99e-35
splice form	429 - 578 (Genscan exon)
	3178 - 3366 (Ensembl exon 4)
	5875 - 5885 (Genscan exon)
	19539 - 19858 (Genscan acceptor)
EnsemblID	ENSG00000073578
Pfam	PF00665 (Integrase core domain)
score	104
E-value	1.93e-28
splice form	29143 - 29583 (Genscan exon)
	34701 - 34724 (Genscan exon)
	36037 - 36150 (Ensembl exon 14)
EnsemblID	ENSG00000079785
Pfam	PF05330 (Protein of unknown function (DUF741))
score	336
E-value	1.79e-98
splice form	14391 - 16312 (Genesplicer acceptor and donor)
	18086 - 18146 (Ensembl exon 14)

Table 2: Partial splice form predictions with hits that are much higher than Pfam trusted-cutoff scores. The splice form positions refer to the absolute position downstream from the transcription start. Here, only the part of the spliceform that codes the domain is shown.

attracted our co-operators' attention (T. Ghewondian, Yerevan, personal communication). PF02177 "Amyloid A4 extracellular domain" is a functional domain in brain proteins related to amyloid plaque formation during neurodegeneration. The occurrence of a sequence fragment in alternatively spliced marenostrin resembling amyloidosis pathomechanisms is worth further investigation, because FMF patients suffer from protein deposit formation in kidneys (secondary amyloidosis). The other signal, the Kazal-type serine protease inhibitor domain may have direct influence on protein deposit formation and/or disturbed removal of such deposits in patient kidney as well. The possible direct influence of the according splice variants in normal and FMF-tissue is now under investigation.

4 Discussion

Our approach allows protein domain analyses of a large spectrum of splice forms a gene can produce. We presented an algorithm that copes with the exponential number of theoretically possible splice forms in polynomial runtime. Moreover, we demonstrated how to assess the splice form spectra in terms of protein domain homology. Furthermore, we showed how to include protein domain homology into gene prediction.

Evidence for the existence of alternative splice forms are expected from EST-based alignment methods. Since the ESTs from publicly available databases do not cover a number of genes sufficiently, a big problem arises in case of rarely expressed mRNA species and minor splice form components. The protein domain homology of a predicted alternative splice form sometimes even narrows the range of tissues where it might be expressed. This is extremely valuable for wet lab techniques like PCR methods since it facilitates and speeds up verification. Furthermore, we will use mass spectrometry data from proteome research for verification purposes. A web server for testing ASFPred is in preparation.

In summary, the approach outlined in this paper complements existing bioinformatic techniques for predicting gene structures and alternative splice forms.

1. Barmak Modrek and Christopher Lee. A genomic view of alternative splicing. *Nat Genet*, 30(1):13–9, 2002.
2. A. J. Lopez. Alternative splicing of pre-mRNA: developmental consequences and mechanisms of regulation. *Annual review of genetics*, 32:279–305, 1998.
3. D. L. Black. Mechanisms of alternative pre-messenger RNA splicing. *Annual review of biochemistry*, 2003.
4. Qiang Xu and Christopher Lee. Discovery of novel splice forms and functional analysis of cancer-specific alternative splicing in human expressed sequences. *Nucleic Acids Research*, 31(19):5635–43, 2003.
5. G. Loudianos, M. Lovicu, V. Dessi, M. Tzetis, E. Kanavakis, L. Zancan, L. Zelante, C. Galvez-Galvez, and A. Cao. Abnormal mRNA splicing

resulting from consensus sequence splicing mutations of ATP7B. *Hum Mutat*, 20(4):260–6, 2002.

6. Mary Anna Carbone, Derek A. Applegarth, and Brian H. Robinson. Intron retention and frameshift mutations result in severe pyruvate carboxylase deficiency in two male siblings. *Hum Mutat*, 20(1):48–56, 2002.
7. Wei-Qun Ding, Susan M. Kuntz, and Laurence J. Miller. A miss-pliced form of the cholecystokinin-B/gastrin receptor in pancreatic carcinoma: role of reduced sellular U2AF35 and a suboptimal 3'-splicing site leading to retention of the fourth intron. *Cancer Res*, 62(3):947–52, 2002.
8. Jun Hyeog Jang and Chong Pyoung Chung. Loss of ligand-binding specificity of fibroblast growth factor receptor 2 by RNA splicing in human chondrosarcoma cells. *Cancer Lett*, 191(2):215–22, 2003.
9. Mariko Yagi, Yasuhiro Takeshima, Hiroko Wada, Hajime Nakamura, and Masafumi Matsuo. Two alternative exons can result from activation of the cryptic splice acceptor site deep within intron 2 of the dystrophin gene in a patient with as yet asymptomatic dystrophinopathy. *Hum Genet*, 112(2):164–70, 2003.
10. B. Modrek, A. Resch, C. Grasso, and C. Lee. Genome-wide detection of alternative splicing in expressed sequences of human genes. *Nucleic Acids Research*, 29(13):2850–9, 2001.
11. Zhining Wang, H. Shuen Lo, Howard Yang, Sheryl Gere, Ying Hu, Kenneth H. Buetow, and Maxwell P. Lee. Computational analysis and experimental validation of tumor-associated alternative RNA splicing in human cancer. *Cancer Res*, 63(3):655–7, 2003.
12. Joanne M. Yeakley, Jian-Bing Fan, Dennis Doucet, Lin Luo, Eliza Wickham, Zhen Ye, Mark S. Chee, and Xiang-Dong Fu. Profiling alternative splicing on fiber-optic arrays. *Nat Biotechnol*, 20(4):353–8, 2002.
13. G. K. Hu, S. J. Madore, B. Moldover, T. Jatkoe, D. Balaban, J. Thomas, and Y. Wang. Predicting splice variant from DNA chip expression data. *Genome Res*, 11(7):1237–45, 2001.
14. C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94, 1997.
15. M. G. Reese, D. Kulp, H. Tammana, and D. Haussler. Genie–gene finding in Drosophila melanogaster. *Genome Res*, 10(4):529–38, 2000.
16. International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.
17. Zhengyan Kan, David States, and Warren Gish. Selecting for functional alternative splices in ESTs. *Genome Res*, 12(12):1837–45, 2002.
18. Evgenia V. Kriventseva, Ina Koch, Rolf Apweiler, Martin Vingron, Peer Bork, Mikhail S. Gelfand, and Shamil Sunyaev. Increase of functional diversity by alternative splicing. *Trends in Genetics*, 19(3):124–

8, 2003.
19. A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. Sonnhammer. The Pfam protein families database. *Nucleic Acids Res*, 30(1):276–80, 2002.
20. M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. USA*, 93(17):9061–6, 1996.
21. R. F. Yeh, L. P. Lim, and C. B. Burge. Computational inference of homologous gene structures in the human genome. *Genome Res*, 11(5):803–16, 2001.
22. A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology. Applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501–31, 1994.
23. S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–63, 1998.
24. A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.
25. Sean Eddy. Hmmer user's guide. Version 2.1.1, see http://hmmer.wustl.edu, December 1998.
26. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK, 1998.
27. M. Clamp, D. Andrews, D. Barker, P. Bevan, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, T. Hubbard, A. Kasprzyk, D. Keefe, H. Lehvaslaiho, V. Iyer, C. Melsopp, E. Mongin, R. Pettett, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and E. Birney. Ensembl 2002: accommodating comparative genomics. *Nucleic Acids Research*, 31(1):38–42, 2003.
28. M. Pertea, X. Lin, and S. L. Salzberg. Genesplicer: a new computational method for splice site prediction. *Nucleic Acids Research*, 29(5):1185–90, 2001.