

Predicting Gene Structures from Multiple RT-PCR Tests (Extended Abstract)

Jakub Kováč¹, Tomáš Vinar², and Broňa Brejová¹

¹ Department of Computer Science, Comenius University, Mlynská Dolina,
842 48 Bratislava, Slovakia, e-mail: kuko@ksp.sk, brejova@dcs.fmph.uniba.sk

² Department of Applied Informatics, Comenius University, Mlynská Dolina,
842 48 Bratislava, Slovakia, e-mail: vinar@ii.fmph.uniba.sk

Abstract. It has been demonstrated that the use of additional information such as ESTs and protein homology can significantly improve accuracy of gene prediction. However, many sources of external information are still being omitted from consideration. Here, we investigate the use of product lengths from RT-PCR experiments in gene finding. We present hardness results and practical algorithms for several variants of the problem. We also apply our methods to a real RT-PCR data set in the *Drosophila* genome. We conclude that the use of RT-PCR data sets can improve the sensitivity of gene prediction and locate novel splicing variants.

Keywords: gene finding, RT-PCR, NP-completeness, dynamic programming, splicing graph

1 Introduction

In spite of recent progress, gene finding remains a difficult problem, particularly in the presence of ubiquitous alternative splicing (Guigo et al., 2006). Nonetheless, prediction accuracy of gene finders can be significantly improved by incorporating various sources of experimental evidence. Reverse transcription-polymerase chain reaction (RT-PCR) is an experimental method often used to confirm or reject predicted gene structures (Siepel et al., 2007). However, the results of RT-PCR could also be used as additional evidence in gene finding to propose new transcripts. In this paper, we study the problem of using the estimated lengths of RT-PCR products for this purpose. We also present a proof-of-concept study on a recently acquired data set of RT-PCR products in the *Drosophila* genome (Brent et al., 2007) designed to verify previously unknown transcripts predicted by CONTRAST (Gross et al., 2007) and NSCAN-EST (Wei and Brent, 2006).

In an RT-PCR experiment, we select two short sequences, called primers, from two predicted exons of the same gene. If both primers are indeed present in the same transcript of the gene, RT-PCR will amplify the region of the transcript

between the primers, and we can estimate its length (with the introns spliced out) on an electrophoresis gel. In case of alternative splicing, we will observe multiple lengths. RT-PCR products can be further cloned and sequenced; however, this process incurs further costs.

We propose a new computational problem in which we want to select a set of transcripts that best agrees with the observed results of several RT-PCR tests. In particular, we represent an RT-PCR test by the positions p_1, p_2 of the two primers in the sequence, and a (potentially empty) list of product lengths. Each length is an interval $[m, M]$, since it is impossible to estimate the length of the product exactly.

As an input to our algorithm, we use a set of potential transcripts of one gene represented as a *splicing graph* (Heber et al., 2002). Vertices of this graph correspond to non-overlapping segments of the DNA sequence. An edge (u, v) in the splicing graph indicates that segment v immediately follows segment u in some transcript (examples of splicing graphs can be seen in Figures 3 and 4). Moreover, two special vertices s and t mark the beginning and the end of a transcript. Thus, every transcript is represented by an (s, t) path in the splicing graph. Note that the vertices do not necessarily correspond to whole exons: a single exon can be split into several vertices, for example due to alternative splice sites. Vertices and edges of the splicing graph are assigned scores, which correspond to the confidence we have that a particular predicted intron or exon segment is correct.

We say that a path π through the splicing graph *explains* test $T = (p_1, p_2, [m_1, M_1], [m_2, M_2], \dots, [m_k, M_k])$ if the path contains both primers p_1 and p_2 , and the distance between the two primers in the transcript defined by π belongs to one of the intervals $[m_1, M_1], \dots, [m_k, M_k]$. If the path contains both primers, but the distance is not covered by any of the intervals associated with T , we say that the path π is *inconsistent with* test T . We can now define a *score of a path π with respect to a set of tests S* as a sum of the scores of all of its vertices and edges, plus a bonus B for each explained test from S , and minus a penalty P for each inconsistent test.

Definition 1 (Gene finding with RT-PCR tests.). *For a given splicing graph G and a set of RT-PCR tests S , find the path π with the highest score.*

There are several practical ways to obtain a splicing graph at a given locus in the genome using *ab initio* gene finders based on hidden Markov models (HMMs). Besides the actual highest probability gene structure in the HMM, Genscan (Burge and Karlin, 1997) can output additional exons that have reasonably high posterior probability. These exons can be joined to potential transcript based on compatibility of their reading frames. Gene finder Augustus (Stanke et al., 2006) uses an HMM path sampling algorithm to output multiple predictions that may correspond to alternative transcripts of the same gene. These methods can also be generalized to take into account additional information, such as ESTs or protein similarity (Stanke et al., 2008). Regardless of the method of generating candidate exons or transcripts, our goal is to set the parameters of these algorithms to achieve high sensitivity even at a cost of low specificity.

A similar problem has been previously investigated by Agrawal and Stormo (2006) who designed a heuristic to find a transcript that explains a single RT-PCR test. Our approach improves on their work by using an exact algorithm rather than a heuristic and by integrating information from multiple overlapping tests.

We have also investigated a simpler problem, where we do not consider path scores and product lengths. Instead, the tests are simply pairs of vertices in the splicing graph, and we either seek a path that contains as many pairs as possible (paths passing useful pairs, PPUP), or try to avoid these pairs altogether (paths avoiding forbidden pairs, PAFP). While even these simpler versions of the problem are strongly NP-hard in general, we show polynomial-time algorithms for several special cases which also extend to the original scenario with lengths and scores and result in pseudo-polynomial algorithms. The PAFP problem has also been studied in connection with automated software testing (Krause et al., 1973; Srimani and Sinha, 1982; Gabow et al., 1976) and tandem mass spectrometry (Chen et al., 2001). We explore several new variants and improve the recent results of Kolman and Pankrác (2009) in this area.

2 General algorithm

In general, the problem of finding a path through the splicing graph consistent with even a single RT-PCR test is NP-hard. This is easy to see, since one can create an instance with a set of disjoint exons of various lengths, and an edge between any two exons. The (s, t) path in such a graph will correspond to a solution of the NP-hard subset-sum problem (Garey and Johnson, 1979), where the single RT-PCR test specifies the target sum.

Fortunately, we can easily design a simple practical dynamic programming algorithm. First, assume a single RT-PCR test $(s, t, [m, M])$. Our task is to find the highest scoring (s, t) path of length between m and M . Let $H[i, \ell]$ be the highest score we can achieve by an (s, v_i) path of length ℓ , or $-\infty$ if there is no such path. Let $\ell(i)$ be the length of vertex v_i and $S(i, j)$ be the score of edge (v_i, v_j) plus the score of vertex v_i . The values of $H[i, \ell]$ can be computed by a simple recurrence in $O(ME)$ time: $H[i, \ell] = \max_{j:(v_j, v_i) \in E} H[j, \ell - \ell(i)] + S(j, i)$, with base cases $H[0, 0] = 0$ and $H[0, \ell] = -\infty$ for $\ell > 0$. The highest value of $H[n + 1, \ell]$ for $m \leq \ell \leq M$ represents the desired path. We can further improve this algorithm by considering only achievable lengths ℓ for each vertex i . We can also eliminate lengths that cannot achieve the target length. In particular, we calculate for each vertex length of the minimum and maximum (v, t) path $\text{mind}(v)$ and $\text{maxd}(v)$, and we can ignore all lengths ℓ for which $\ell + \text{mind}(v) > M$ or $\ell + \text{maxd}(v) < m$. Since we can expect that in practical cases H will be sparse, and values of M are small due to limitations of RT-PCR, this algorithm is a practical solution to the problem.

The algorithm is easily extended to multiple tests and to the general problem with bonuses and penalties as defined above. Even though the running time $O(M^k E)$ grows exponentially with the maximum number of overlapping RT-

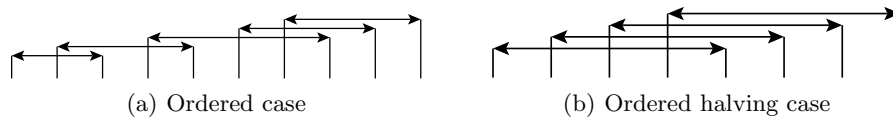


Fig. 1. Two special cases of primer pair positions.

PCR tests k , this number is typically small in real data sets (such as the data set that we consider below, where $k = 4$).

3 Primer positioning and hardness

In the previous section, we have demonstrated that the general problem is NP-hard. We have shown an algorithm that is pseudo-polynomial for a single test, and exponential in the number of overlapping tests in general case. Here, we demonstrate several complexity results for various special cases of interest characterized by a particular placement of primer pairs.

First, we examine two cases illustrated in Fig. 1. We say that two tests *halve* each other if the corresponding intervals of the DNA sequence overlap each other, but neither is completely included in the other. In the *ordered case*, all the left primers are arranged in the same order as the corresponding right primers (i.e., every two tests are either disjoint, or they halve each other). In the *ordered halving case*, the tests are ordered and moreover every two tests halve each other.

Even though the distinction between these two cases seems rather subtle, we will show that in the ordered case, the problem is strongly NP-hard, while in the ordered halving case there exist a pseudo-polynomial solution. Moreover, this is true even if we do not consider lengths associated with the primers and scores associated with the exons.

In this simplified scenario, every pair of primers is either *useful* or *forbidden*. We consider two versions of the problem. First, the *PAFP problem* seeks paths avoiding all forbidden pairs, i.e. from each pair we are allowed to include at most one end vertex. Second, the *PPUP problem* seeks paths passing as many useful pairs as possible. The PAFP problem corresponds to RT-PCR tests without any products, while the PPUP corresponds to successful RT-PCR tests, but without considering product lengths.

Theorem 1. *The PAFP and PPUP problems with ordered set of pairs are strongly NP-complete.*

Proof. We will prove the claim by reduction from 3-SAT. Let φ be a conjunction of n clauses $\varphi_1 \wedge \dots \wedge \varphi_n$ over m variables x_1, \dots, x_m , where $\varphi_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$ and each literal $\ell_{i,j}$ is either x_k or \bar{x}_k . We will construct graph G and a set of pairs S such that the solution of the corresponding PPUP problem gives the satisfying assignment of φ .

Graph G consists of several copies of a block B of $2m$ vertices as shown in Fig. 2(a). Any left-to-right path through the block B naturally corresponds to an

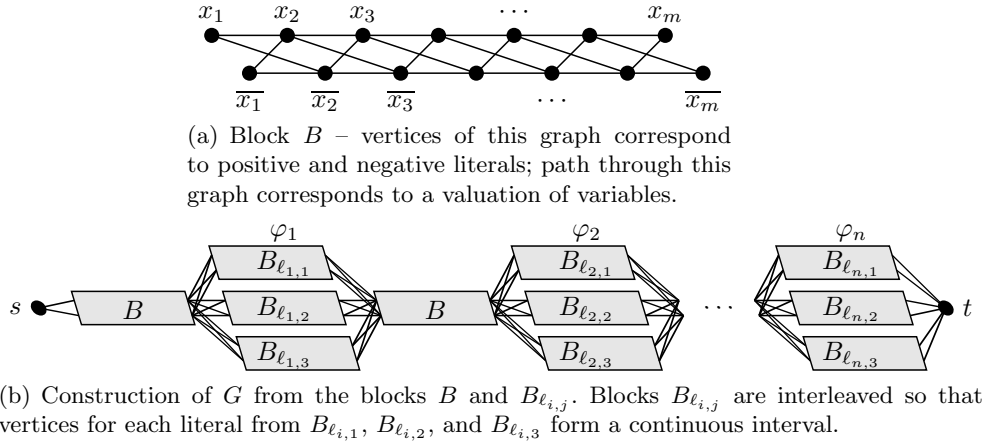


Fig. 2. Construction of the graph G for a 3-SAT formula φ .

assignment of the variables. For each literal $\ell_{i,j}$, we also construct a block $B_{\ell_{i,j}}$ which is identical to B , except that the vertex corresponding to $\overline{\ell_{i,j}}$ is missing. The blocks are joined together as outlined in Fig. 2(b). The path passing through a construct corresponding to a clause φ_i must pass through one of the three blocks, and thus choose an assignment that satisfies the clause.

The set of pairs S will enforce that the assignment of the variables is the same in all blocks. This is done by adding a useful pair between corresponding literals in block $B_{\ell_{i,j}}$ and the preceding B -block and the following B -block. A path corresponding to the solution of PPUP in (G, S) will thus give a unique satisfying assignment of the variables x_1, \dots, x_m if the path contains at least $(2n - 1)m$ pairs (otherwise there is no satisfying assignment).

The resulting set S is not ordered, since three nested intervals start in each node of B . The issue can be easily fixed by splitting each vertex of B into a path of length three and using a different vertex of the path for each of the three intervals.

The reduction is analogous for PAFP. In this case, we reverse the order of vertices x_i and $\overline{x_i}$ in B -blocks (but we keep the order in $B_{\ell_{i,j}}$ blocks the same). The set of ordered pairs S will now be composed from forbidden pairs between atoms x_i in $B_{\ell_{i,j}}$ blocks with their counterparts $\overline{x_i}$ in the previous and the following B -block. \square

We have demonstrated that PAFP and PPUP are strongly NP-hard on ordered pairs. Consequently, the general problem explored in the previous section is also strongly NP-hard, since PPUP is a special case of that problem. However, both PAFP and PPUP can be solved in polynomial time for the ordered halving pairs. Before we demonstrate the algorithm, we note a simplifying transformation on graph G .

Lemma 1 (Single pair per vertex). *Every graph G and a set S of either forbidden or useful pairs can be transformed to a graph G' and a set S' such that*

in each vertex starts or ends exactly one pair from S' , and there is a one-to-one correspondence between the solutions of PAFP and PPUP problems for G, S and for G', S' . Moreover, the transformation can be done efficiently in $O(PE)$ time, where P is the number of pairs in S , and the halving and ordering structure of the graph is preserved.

The above lemma allows us to simplify the algorithms below, since we do not need to consider multiple pairs starting or ending at a particular node, and at the same time we reduce the size of the graph in most cases. The PAFP problem has recently been solved for ordered halving case by Kolman and Pankrác (2009) in $O(PE + P^5)$ time. Here, we show an algorithm for the PPUP problem with ordered halving pairs.

Theorem 2. *The PPUP problem with ordered halving useful pairs can be solved in $O(PE + P^3)$ time.*

Proof. First, we can remove all useful pairs (v_i, v_j) for which there is no (s, v_i) path or no (v_j, t) path because these pairs will never be used on any (s, t) path. Furthermore, due to Lemma 1, we can assume without loss of generality that the graph contains exactly one pair starting or ending at each vertex of the graph G . Such a graph G will have $2P$ vertices v_1, \dots, v_{2P} , and useful pairs $(v_1, v_{P+1}), (v_2, v_{P+2}), \dots, (v_P, v_{2P})$.

To search for the (s, t) path containing the largest number of useful pairs, we construct a new graph H with vertices w_1, \dots, w_P , each vertex corresponding to a single useful pair. We will say that w_i and w_j are connected by a *blue edge* if there are left-to-right (v_i, v_j) and (v_{P+i}, v_{P+j}) paths in graph G . Moreover, vertices w_i and w_j are connected by a *red edge* if there is a left-to-right (v_j, v_{P+i}) path in G . Graph H can be constructed in $O(PE + P^2)$ time.

Searching for the PPUP in G now translates into searching for the longest (w_i, w_j) left-to-right blue path in H , where w_i and w_j are also connected by a red edge. The longest such paths for all pairs of w_i and w_j can be easily found by dynamic programming in $O(P^3)$ time, and thus the total running time of the algorithm is $O(PE + P^3)$ (including the preprocessing time required by the transformation in Lemma 1). \square

We have also investigated the complexity of PAFP and PPUP problem for other special conformations of pairs: *disjoint pairs* (no two intervals defined by the pairs overlap), *well-parenthesized pairs* (no two intervals halve each other), *halving structure* (every two intervals halve each other or they are nested), and *nested pairs* (for any two intervals, the smaller interval is nested in the larger interval). The results are summarized in Table 1 and the proofs are omitted due to the space restrictions. Note that the NP-hardness result on general PAFP problem is due to Gabow et al. (1976), and several of the other forms of the PAFP problem have been recently investigated by Kolman and Pankrác (2009); all the other results are new.

The hardness proofs obviously carry over to the more general problem of finding the best (highest score) (s, t) path in a splicing graph, where each vertex

Table 1. Complexity of the PAFP and PPUP problem, where E is the number of edges in the input graph and P is the number of forbidden/useful pairs. NP-hardness of the general problem (*) was proved by Gabow et al. (1976); results marked by † were first proved by Kolman and Pankrác (2009).

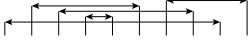

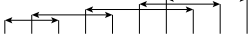
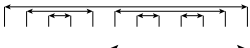
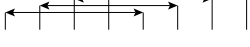
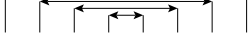
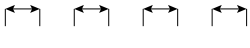

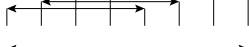
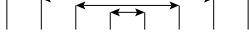
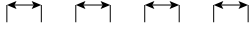
PROBLEM	FORBIDDEN (PAFP)	USEFUL (PPUP)	EXAMPLE
<i>general problem</i>	NP-hard *	NP-hard	
<i>halving structure</i>	NP-hard †	NP-hard	
<i>ordered</i>	NP-hard	NP-hard	
<i>well-parenthesized</i>	$O(PE + P^3)$ †	$O(PE + P^3)$	
<i>ordered halving</i>	$O(PE + P^5)$ †	$O(PE + P^3)$	
<i>nested</i>	$O(PE + P^3)$	$O(PE)$	
<i>disjoint</i>	$O(E)$	$O(E)$	

Table 2. Complexity of the general problem of finding an (s, t) path in a splicing graph with zero penalty, where E is the number of edges, P is the number RT-PCR tests, M is the maximum measured length and Δ is the maximum number of lengths in an RT-PCR test that get any bonus (e.g. $\Delta = M - m + 1$ for one measured length $[m, M]$).

PROBLEM	COMPLEXITY	EXAMPLE
<i>well-parenthesized</i>	$O(PME + P^3M^2)$	
<i>ordered halving</i>	$O(PME + P^3\Delta^3M^2)$	
<i>nested</i>	$O(PME + P^2M^2)$	
<i>disjoint</i>	$O(ME)$	

has a length, each edge has a score and we get bonus B for explaining a length of an RT-PCR test. Thus we cannot hope for even a pseudopolynomial algorithm for the halving or ordered case (unless $P=NP$). On the other hand, positive results for the PPUP problem can be extended to pseudopolynomial algorithms for the more general setting. Our results are summarized in Table 2; we omit the proofs due to the space restrictions. In the well-parenthesized case we were able to further generalize the algorithm by considering penalties for inconsistent RT-PCR tests, achieving the same running time.

4 Finding genes in *D. melanogaster*

To test the improvement in gene finding accuracy achieved by incorporating RT-PCR product lengths, we have used a set of 2784 actual RT-PCRs from the

Brent laboratory on the *Drosophila melanogaster* genome (Brent et al., 2007). These experiments were designed to test novel transcripts predicted by CONTRAST (Gross et al., 2007) and NSCAN-EST (Wei and Brent, 2006). Each RT-PCR product was sequenced and aligned to the genome. We have estimated the length of each product by locating likely positions of the two primers in the sequenced product. We have added $\pm 15\%$ margin or at least ± 50 nucleotides to simulate the uncertainty in length estimates from electrophoresis gels. After discarding primers spanning exon boundaries or located in the predicted untranslated regions, and merging primer pairs with identical primer coordinates, we were left with 1955 RT-PCR tests in 1159 loci, each locus corresponding to a set of overlapping predicted or known genes. Overall, 942 tests have produced at least one product and fewer than 10 tests have produced two products with significantly different lengths. Note that approximately 2% of the products mapped to other than the intended genomic locus; thus the estimated lengths in our test do not always correspond to real transcripts at the locus of interest. We expect that this type of error would also occur in practice, since without sequencing we cannot determine whether the product indeed maps to the expected location.

To obtain the splicing graph, we have used Augustus gene finder (Stanke et al., 2006) capable of sampling random gene structures from the posterior probability distribution defined by the underlying generalized HMM. For each locus, we have created the splicing graph based on 1000 samples. For each vertex of the graph, we have estimated the posterior probability p as a fraction of the samples that contain this vertex. Score of the vertex was then set to $p - C(1 - p)$ for some constant p (we have used $C = 0.5$). Edge scores were computed analogously. A similar scoring scheme was used as an optimization criterion in the recent discriminative gene finder CONTRAST (Gross et al., 2007).

We have implemented the general $O(M^k E)$ algorithm, which finds the (s, t) path with the maximum score that aside from scores of vertices and edges on the path also includes bonus $B = 5$ for each explained test and penalty $P = 1$ for each inconsistent test. This path may explain several of the observed RT-PCR product lengths. For each length not explained by the path, we run the algorithm again, this time finding the highest scoring path that explains this length, if there is any. In this way, we may obtain multiple alternative transcripts.

Table 3 shows the results of our algorithm on 1022 loci for which Augustus sampling produced at least two different transcripts. We compare our algorithm with the most probable transcript produced by Augustus run in the Viterbi mode and also with the highest scoring path in our splicing graph (without considering any RT-PCR bonuses or penalties). All three versions have almost identical accuracy compared to the known RefSeq genes. In particular, although our version is capable of producing multiple transcripts per gene, this does not lead to significant decrease in specificity.

The RT-PCR tests were designed to discover new transcripts, and so their results may not help to obtain predictions closer to the RefSeq annotations. Thus we also compare the predictions to the exons defined by aligning the sequenced RT-PCR products to the *Drosophila* genome. Since the products often do not

Table 3. Gene prediction accuracy on 1022 loci. Sensitivity is the fraction of annotated coding nucleotides, exons, or splice sites that were correctly predicted and specificity is the fraction of predictions that are correct.

Compared to RefSeq:	with PCR	w/o PCR	Augustus	
Exon Sensitivity	65%	64%	63%	
Exon Specificity	58%	58%	59%	
Nucleotide Sensitivity	84%	84%	83%	
Nucleotide Specificity	75%	75%	76%	
Compared to RT-PCR products:	with PCR	w/o PCR	Augustus	RefSeq
Acceptor Sensitivity	75%	73%	72%	73%
Donor Sensitivity	76%	74%	73%	74%

span whole exons, we instead compare individual splice sites (donors and acceptors). In this test, we see some improvement of sensitivity: for example Augustus predicts 72% of such donor sites while our program 75%. This is even more than RefSeq, which includes only 73% of these sites (Table 3).

The accuracy of our program is limited by two factors. First, we can only predict transcripts that have an (s, t) -path in the splicing graph. In this test, Augustus splicing graphs contained 85% of all donors and acceptors supported by aligned RT-PCR products, so even under the ideal conditions our approach could not exceed this level of sensitivity. Moreover, we rely on the Augustus scores together with bonuses and penalties to choose among possible transcripts, and therefore improved quality of these scores would also improve our prediction. Second limitation stems out of the density and quality of the RT-PCR tests. In our data set, 62% of the loci have only one RT-PCR test, and only 16% have three or more tests. Also, we add minimum of 100 bp tolerance to the observed lengths which means that we are unable to detect presence or absence of smaller exons unless they contain a primer. Perhaps this problem can be alleviated by a careful study of observation errors of length estimates obtained from electrophoresis gels.

Figures 3 and 4 illustrate both advantages and problems of our approach. In Fig.3, our algorithm successfully predicts RefSeq exons omitted by Augustus. However, one of the downstream predicted exons is shorter on its 3' end because the splicing graph does not contain the correct form of the exon. Moreover, even the mispredicted shorter form satisfies the length tolerance of the test and gets a bonus. In Fig.4, the lengths from two tests allowed inference of a gene structure quite different from both RefSeq and Augustus transcripts. The prediction agrees well with the sequenced test products that were not part of the input.

5 Discussion and future work

In this paper, we introduced a new computational problem inspired by integrating RT-PCR information into gene finding. We have shown a practical algorithm and explored the boundary between NP-hard and polynomially solvable special

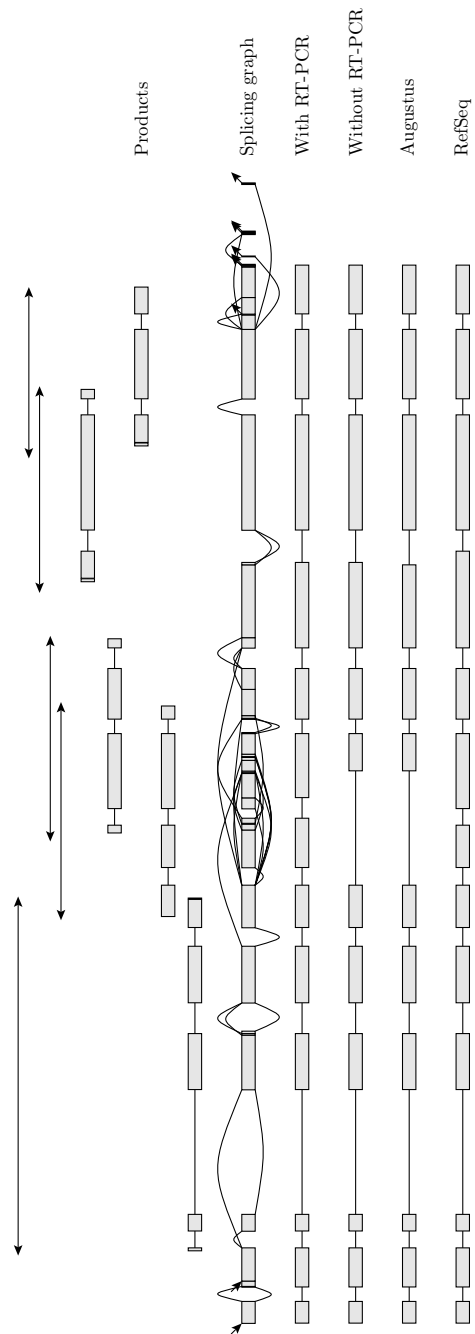


Fig. 3. Locus of the GluRIIB gene (glutamate receptor). The locus has five RT-PCR tests, each with a sequenced product mapping to this region.

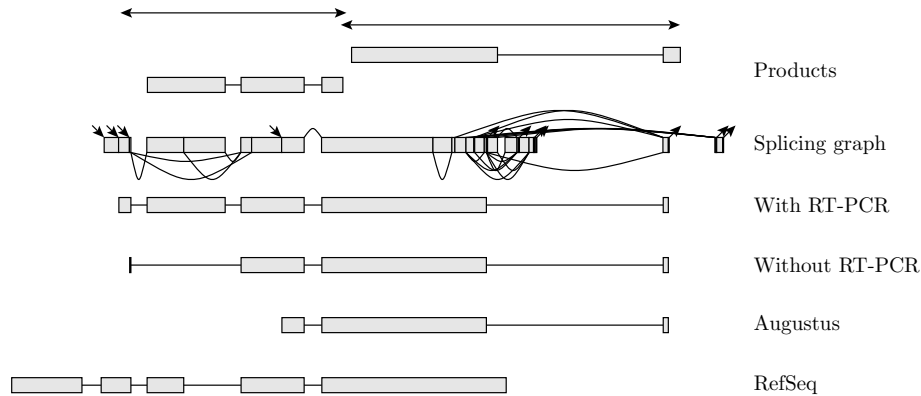


Fig. 4. Locus of the Q7KTW2 gene belonging to the amiloride-sensitive sodium channel family. Our algorithm predicts a transcript different from RefSeq structure yet agreeing well with one of the sequenced RT-PCR products.

cases of the problem. Finally, we have demonstrated that this method is indeed able to locate new splicing variants.

One problem we have not explored in this paper is the design of RT-PCR experiments. The current state of the art first uses a gene finder to predict individual transcripts, and then concentrate on predictions that are novel or different from established annotations (Siepel et al., 2007). In contrast, our approach suggests the possibility of designing primers based on a splicing graph representing exponential number of potential transcripts. In fact, we have previously investigated a theoretical problem related to this question (Biedl et al., 2004). The results presented here suggest that while the general problem of gene finding with RT-PCR product lengths is NP-hard, it is possible to position the queries in such a way that they can be analyzed efficiently.

It has been suggested (Agrawal and Stormo, 2006) that RT-PCR experiments followed by estimation of the product lengths on an electrophoresis gel can be considered a high-throughput method, especially if the gels could be analyzed computationally. In principle, the products of RT-PCR can be sequenced, and the cost of this is no longer prohibitive. In addition, new sequencing technologies suggest the possibility to exhaustively sequence large cDNA libraries in the near future. Nonetheless, we believe that the approach described in this paper will remain relevant for some time for smaller laboratories wishing to concentrate on non-model organisms or particular genomic loci. In the RT-PCR dataset explored in this paper, 84% of the loci have fewer than three RT-PCR tests, but instead one could cover a single locus in a more exhaustive way, possibly reusing the same primers in different combinations. Such an approach can lead to a very detailed characterization of transcripts at a particular locus of interest.

Acknowledgements. We would like to thank Michael Brent and Charles Comstock for providing RT-PCR experimental data. This research was supported by European Community FP7 grants IRG-224885 and IRG-231025.

Bibliography

- Agrawal, R. and Stormo, G. D. (2006). Using mRNAs lengths to accurately predict the alternatively spliced gene products in *caenorhabditis elegans*. *Bioinformatics*, 22(10):1239–1244.
- Biedl, T., Brejova, B., Demaine, E., Hamel, A., Lopez-Ortiz, A., and Vinar, T. (2004). Finding hidden independent sets in interval graphs. *Theoretical Computer Science*, 310(1-3):287–307.
- Brent, M., Langton, L., Comstock, C. L., and van Baren, J. (2007). Exhaustive RT-PCR and sequencing of all novel NSCAN predictions in *Drosophila melanogaster*. Personal communication.
- Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94.
- Chen, T., Kao, M. Y., Tepel, M., Rush, J., and Church, G. M. (2001). A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *J Comput Biol*, 8(3):325–327.
- Gabow, H. N., Maheswari, S. N., and Osterweil, L. J. (1976). On two problems in the generation of program test paths. *IEEE Trans. Soft. Eng.*, 2(3):227–231.
- Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Gross, S. S., Do, C. B., Sirota, M., and Batzoglou, S. (2007). CONTRAST: a discriminative, phylogeny-free approach to multiple informant de novo gene prediction. *Genome Biol*, 8(12):R269.
- Guigo, R. et al. (2006). EGASP: the human ENCODE genome annotation assessment project. *Genome Biol*, 7 Suppl 1:S2.
- Heber, S., Alekseyev, M., Sze, S. H., Tang, H., and Pevzner, P. A. (2002). Splicing graphs and EST assembly problem. *Bioinformatics*, 18 Suppl 1:S181–188.
- Kolman, P. and Pankrác, O. (2009). On the complexity of paths avoiding forbidden pairs. *Discrete Applied Mathematics*. To appear.
- Krause, K. W., Smith, R. W., and Goodwin, M. A. (1973). Optional software test planning through automated network analysis. *Proceedings 1973 IEEE Symposium on Computer Software Reliability*, pages 18–22.
- Siepel, A., Diekhans, M., Brejova, B., Langton, L., Stevens, M., Comstock, C. L., Davis, C., Ewing, B., Oommen, S., Lau, C., Yu, H. C., Li, J., Roe, B. A., Green, P., Gerhard, D. S., Temple, G., Haussler, D., and Brent, M. R. (2007). Targeted discovery of novel human exons by comparative genomics. *Genome Res*, 17(12):1763–1763.
- Srimani, P. K. and Sinha, B. P. (1982). Impossible pair constrained test path generation in a program. *Inf. Sci.*, 28(2):87–103.
- Stanke, M., Diekhans, M., Baertsch, R., and Haussler, D. (2008). Using native and syntenically mapped cDNA alignments to improve de novo gene finding. *Bioinformatics*, 24(5):637–644.
- Stanke, M., Keller, O., Gunduz, I., Hayes, A., Waack, S., and Morgenstern, B. (2006). AUGUSTUS: ab initio prediction of alternative transcripts. *Nucleic Acids Res*, 34(Web Server issue):W435–439.
- Wei, C. and Brent, M. R. (2006). Using ESTs to improve the accuracy of de novo gene prediction. *BMC Bioinformatics*, 7:327.