BIOINFORMATICS

Vol. 18 Suppl. 1 2002
Pages S181–S188

Splicing graphs and EST assembly problem

Steffen Heber, Max Alekseyev, Sing-Hoi Sze, Haixu Tang and Pavel A. Pevzner

Department of Computer Science & Engineering, University of California, San Diego, La Jolla, CA, 92093-0114, USA

Received on January 24, 2002; revised and accepted on April 1, 2002

ABSTRACT

Motivation: The traditional approach to annotate alternative splicing is to investigate every splicing variant of the gene in a case-by-case fashion. This approach, while useful, has some serious shortcomings. Recent studies indicate that alternative splicing is more frequent than previously thought and some genes may produce tens of thousands of different transcripts. A list of alternatively spliced variants for such genes would be difficult to build and hard to analyse. Moreover, such a list does not show the relationships between different transcripts and does not show the overall structure of all transcripts. A better approach would be to represent all splicing variants for a given gene in a way that captures the relationships between different splicing variants.

Results: We introduce the notion of the *splicing graph* that is a natural and convenient representation of all splicing variants. The key difference with the existing approaches is that we abandon the linear (sequence) representation of each transcript and replace it with a graph representation where each transcript corresponds to a path in the graph. We further design an algorithm to assemble EST reads into the splicing graph rather than assembling them into each splicing variant in a case-by-case fashion.

Availability:
http://www-cse.ucsd.edu/groups/bioinformatics/software.html

Contact: sheber@ucsd.edu

Keywords: EST assembly; splicing graph; alternative splicing.

INTRODUCTION

Recent studies provide evidence that oncogenic potential in human cancer may be modulated by alternative splicing. For example, the progression of prostate cancer from an androgen-sensitive to an androgen-insensitive tumor is accompanied by a change in the splicing pattern of fibroblast growth factor receptor 2 (Carstens *et al.*, 1997). Another study (Heuze *et al.*, 1999) characterized a prominent alternatively spliced variant for Prostate-Specific Antigen, which is the most important marker available today for diagnosing and monitoring patients with prostate cancer. In these studies, the found isoforms were discovered by chance in a case-by-case fashion—the question whether other alternatively spliced variants of these genes are implicated in cancer remains open.

The first systematic attempt to elucidate the splicing variants of genes implicated in (ovarian) cancer was undertaken by Hu *et al.* (1998). They found a new splicing variant for the human multidrug resistance gene MDR1 and the major vault protein MVP. However, the question of how to find all alternatively spliced variants of a given gene remained open. We argue that the splicing graph (defined below) built from available EST and cDNA data is a tool to visualize *all* potential splicing variants, to guide further research efforts, and to decide which putative splicing variants are really expressed in certain tissues.

Let $\{s_1, \ldots, s_n\}$ be the set of all RNA transcripts for a given gene of interest. Each transcript s_i corresponds to a set of genomic positions V_i with $V_i \neq V_j$ for $i \neq j$. Define the set of all *transcribed* positions $V = \bigcup_{i=1}^{n} V_i$ as the union of all sets V_i. The splicing graph G is the directed graph on the set of transcribed positions V that contains an edge (v, w) if and only if v and w are consecutive positions in one of the transcripts s_i. Every transcript s_i can be viewed as a path in the splicing graph G and the whole graph G is the union of n such paths. We usually collapse vertices with $indegree = outdegree = 1$ to obtain a more compact representation of the splicing graph. Splicing graphs are similar to *gene models* that represent exons connected by edges if they are consecutive in a transcript (Figure 1a). However, in contrast to from gene models, splicing graphs can be built solely from transcript data without any knowledge of the genomic sequence.

Splicing graphs may be rather complicated. For example, the gene model of the *Drosophila Dscam* gene implies roughly 38 000 potential transcripts (Graveley, 2001). Representation of these transcripts in a case-by-case fashion does not show the relationships between different potential transcripts and makes it difficult to design PCR

primers (or DNA arrays) for further identification of transcripts present in certain tissues.

Information about alternative splicing is often derived from EST assemblies. Most available EST assemblies are built by traditional fragment assemblers that attempt to assemble reads into linear sequences rather than into a graph reflecting information about splicing variants. Since these assemblers were designed for a very different combinatorial problem, we argue that it is an inadequate approach that becomes infeasible when a gene has many alternatively spliced variants. A better approach would be to assemble EST reads into a splicing graph. Below we describe a fragment assembly algorithm that assembles ESTs into graphs that represent *all* potential splicing variants rather than assembling them in a case-by-case fashion.

The EST assembly problem is more difficult than the traditional fragment assembly problem. In the first approximation, EST assembly corresponds to the Graph Reconstruction Problem (GRP) that generalizes the String Reconstruction Problem in the traditional fragment assembly. In GRP, one assumes that there exists an (unknown) directed graph (splicing graph) with each vertex labeled by a letter from the {A,T,G,C} alphabet. A path in this graph can be interpreted as a string by concatenating the labels of the traversed vertices. The input data to the GRP is a collection of strings (ESTs) corresponding to a set of paths. The GRP problem is to reconstruct the graph from the collection of these strings (i.e. to build a graph in which EST reads correspond to paths). Similarly to the traditional fragment assembly, the objective for such reconstruction may vary. For example, one may want to reconstruct the graph with minimal number of edges (compare with the Shortest Superstring Problem).

Our approach to the Graph Reconstruction Problem takes advantage of the Eulerian approach to fragment assembly. It is based on the observation that the splicing graph is the union of paths corresponding to ESTs. One therefore can visualize such a graph as the 'gluing' of all similar segments in ESTs. The mathematical technique for such gluing amounts to *de Bruijn graphs* that were studied in Pevzner *et al.* (2001). The difficulty in applying this approach to EST assemblies is the high error rate in EST data. Every error in ESTs creates a bulge in the constructed graph thus concealing the underlying structure of the splicing graph. Below we describe an approach that resolves this problem and generates the splicing graphs.

EST ASSEMBLIES

Nucleotide sequence databases are growing rapidly and expressed sequence tags (ESTs) are their fastest growing division. ESTs are important tools for gene and exon finding (Burke *et al.*, 1998), gene expression analysis (Schmitt *et al.*, 1999), detection of alternative splicing (Kan *et al.*, 2001) and SNPs (Picoult-Newberg *et al.*, 1999), as well as for investigating the proteome (Lisacek *et al.*, 2001). ESTs are collected in *gene indices* like UniGene, the TIGR Gene Index, GeneNest, and STACK (see Bouck *et al.* (1999) for an overview). They are clustered into sets which are supposed to correspond to single genes.

Due to the fragmentary nature and low quality of EST sequences, biologists assemble them into consensus sequences in order to form EST contigs, to eliminate sequencing errors, and to analyse alternative splicing variants (Burke *et al.*, 1998; Zhuo *et al.*, 2001; Coward *et al.*, 2002). Reconstruction of all putative transcripts from a collection of EST reads is a difficult problem. Conventional EST assembly approaches like Phrap (Green, 1994), CAP3 (Huang and Madan, 1999), TIGR Assembler (Sutton *et al.*, 1995), typically use algorithms borrowed from fragment assembly that were designed for the very different problem of assembling reads into a single linear consensus sequence. Most efforts in these algorithms are invested into coping with repeats, which is not a bottleneck for EST assemblies. As a result, it is not clear how such algorithms perform in the presence of alternative splicing, which is estimated to occur in up to 60% of human genes (Mironov *et al.*, 1999; Modrek and Lee, 2001) and may produce thousands of transcripts (Graveley, 2001). Possible problems in EST assemblies are truncated and misassembled (erroneous) contigs. See Figure 1b and the Results Section for examples. Our new assembly approach is tailored especially for EST assembly in the presence of alternative splicing. It overcomes the above problems by assembling ESTs into a *splicing graph* instead of a linear consensus sequence. This enables us to integrate, in a natural and unambiguous way, transcripts which originate from the same gene, but differ due to alternative splicing or polymorphisms.

The splicing graph combines reoccurring EST parts into single paths and displays sequence variations and alternative splicing as bifurcations in the graph. The result is a compact representation of EST data. In contrast to other bioinformatical approaches (Mironov *et al.*, 1999; Kan *et al.*, 2001; Modrek *et al.*, 2001; Coward *et al.*, 2002), the splicing graph can be constructed without any knowledge of the genomic sequence. We consider this as an advantage, since although genome sequencing advances rapidly, a large amount of accessible EST data still cannot be mapped onto genomic sequences (e.g. because of this, in Modrek *et al.* (2001) about 45% of all UniGene clusters had to be discarded from further analysis). In addition, there are several organisms with EST data available but without their own genome sequencing projects.

ALGORITHMS

Splicing graphs

We first describe the construction of the splicing graph in the error-free case. Our approach follows the ideas of Pevzner (1989) and Idury and Waterman (1995). For a collection $S = \{s_1, \ldots, s_n\}$ of ESTs, we define $Spec_k(S)$ as the set of all $k$mers and their reverse complements contained in S ($k = 20$ by default). Let $V = Spec_{k-1}(S)$ be the set of $(k-1)$mers in S. We construct a graph G with vertex set V, where for each k-mer $x_1 \ldots x_k \in Spec_k(S)$, there exists an edge $e = (x_1 \ldots x_{k-1}, x_2 \ldots x_k)$ between vertices $(x_1 \ldots x_{k-1})$ and $(x_2 \ldots x_k)$. See Pevzner *et al.* (2001) for the analysis of this approach as compared to the traditional *overlap graph* approach to fragment assembly.

Error correction

Sequencing errors are a serious problem for the construction of the splicing graph. They 'blur' the graph by adding tangles of erroneous edges, making it very hard to recognize any structure. The error correction procedure described in Pevzner *et al.* (2001) works for relatively low error rates (like in traditional fragment assembly) and may fail for noisy EST data. We overcome this problem by developing a different error correction approach based on the evaluation of multiple alignments among overlapping reads. An overlap is only accepted if the corresponding alignment satisfies certain constraints (i.e. alignment length larger than 30 bases, identity score higher than 95% in the overlapping segments). We correct only those positions where a base is overwhelmingly out-voted by bases aligned to it. Finally, we trim reads to exclude regions with low quality value.

In simulated EST data we found that our error correction procedure removes about 99% of the sequencing errors. It is rarely possible that it introduces errors, but these cases occur mostly in ambiguous low coverage regions and at read ends. They usually do not affect the topology of the splicing graph and therefore can be easily fixed at the consensus stage (see below).

Although our error correction procedure is very efficient, it leaves few errors uncorrected. These errors lead to spurious bifurcations in the splicing graph and need to be deleted. To detect such spurious bifurcations, we align reads which pass through a bifurcation node but diverge afterwards. If they align well, it is an indication of an uncorrected sequencing error rather than alternative splicing. In this case, we eliminate the bifurcation from the graph. We also mask non-alignable read ends.

Consensus derivation

Once we have the read layout on the splicing graph, we deduce the consensus nucleotide at a certain position (vertex of the graph) by aligning all overlapping reads at this position. A quality-weighted vote determines the consensus base. The benefit of the splicing graph approach at this point is that it takes into account all ESTs derived from different transcripts which cover a given position (vertex) rather than only ESTs derived from a single transcript as in the conventional approach.

Visualization

We display the splicing graph in a user-friendly interactive graphical viewer using the LEDA package (Melhorn and Näher, 1999). To reduce the complexity of the graph, we merge successive nodes with in- and out-degree equal to one to a single *supernode*. We draw supernodes by rectangles, with width corresponding to the length of the represented sequence and indicate the number of supporting reads of a splice site by edge thickness. The corresponding sequences and supporting reads can be accessed by selecting vertices and edges. Additionally, we offer further graph simplifications based on coverage based thresholding, where we remove bifurcations which are supported only by a small number of reads.

Alternative splicing

In the resulting splicing graph (without repeated k-mers), vertices of in- or outdegree larger than one point to alternative splicing events. For each such bifurcation we select reads which witness the sequence variance and optimize quality constraints such as the position of splice site and error rate. They can be used for further validation of the alternative splicing by additional experiments and are very useful to discern bifurcations generated by contaminations, immature mRNA, or data processing artifacts.

RESULTS

Input Data

We applied our approach to UniGene clusters of UniGene Build #141 (ftp://ncbi.nlm.nih.gov/repository/UniGene) and to assemblies of the TIGR Human Gene Index (HGI) Release 7.0 (http://www.tigr.org/tdb/hgi/index.html). To evaluate our results, we used the human genomic assembly sequences in GenBank.

Splicing graphs of UniGene clusters

Our goal is to represent a UniGene cluster by a splicing graph and to determine a corresponding catalog of alternative splicing candidates.

To validate our approach we applied it to the human adenylosuccinate lyase (*ADSL*) gene (Kmoch *et al.*, 2000). ADSL is about 20 kb long, and contains 13 exons of overall length about 2 kb. The ADSL gene model is shown in Figure 1a.

To demonstrate the accuracy of our approach we built the splicing graph of the ADSL gene (UniGene cluster Hs.75527) using only EST sequences in a blind exper-

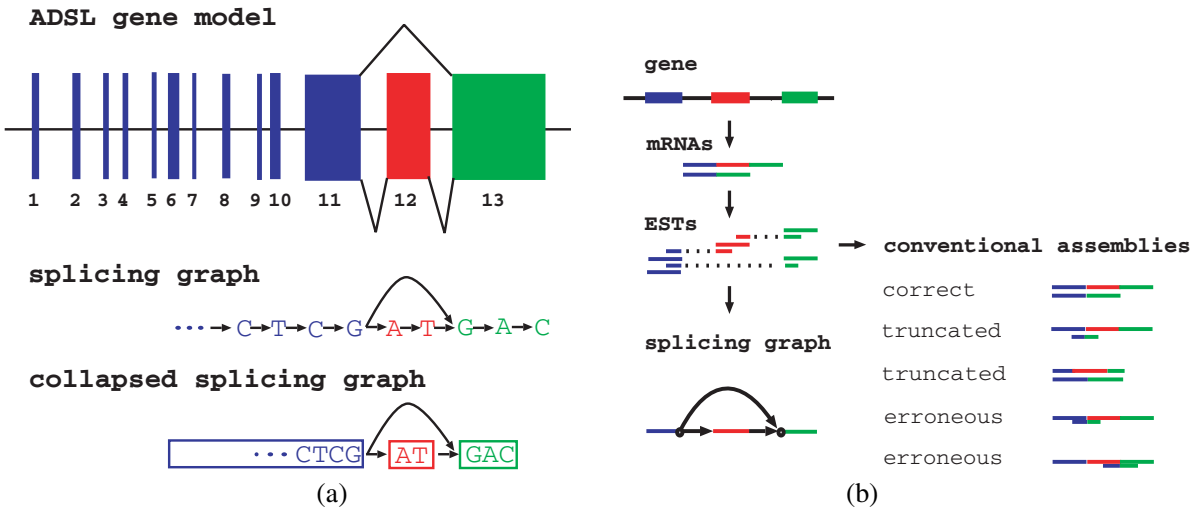


Fig. 1. (a) Gene model and splicing graph of the ADSL gene. Boxes represent individual exons. Bent lines indicate alternative splicing that amounts to skipping exon 12. (Not drawn to scale!) (b) Problems of EST assemblies in the presence of alternative splicing. While the ESTs can have ambiguous or degenerate conventional assemblies (some examples are shown), the splicing graph is uniquely defined.

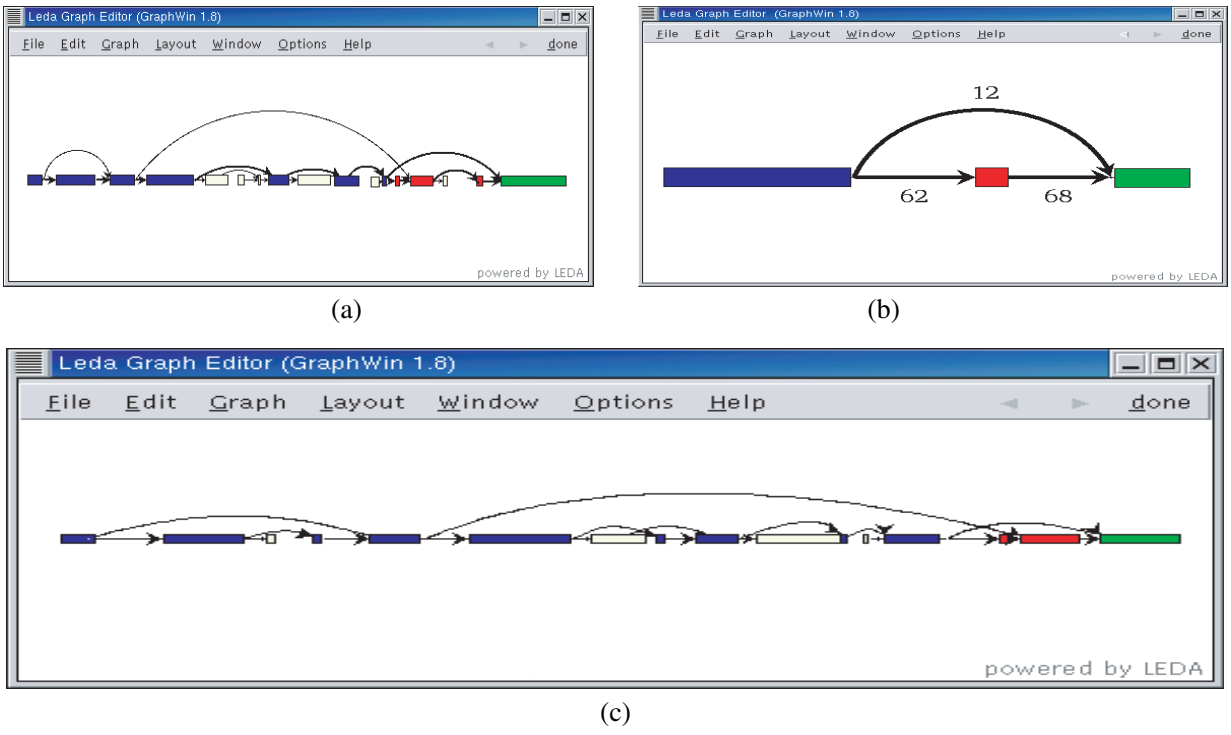


Fig. 2. (a) Visualization of the ADSL splicing graph of UniGene cluster Hs.75527 using word size $k = 20$. (b) The same graph after coverage thresholding ($t = 3$). The numbers on edges show the number of EST reads supporting each alternative splicing junction. (c) Splicing graph of UniGene cluster Hs.75527 using genomic sequence for error correction and word size $k = 20$. (Colors indicate matching parts in the gene model.)

iment (i.e. we ignored all cDNA full-length sequences and the genomic template). After coverage thresholding ($t = 3$) the resulting splicing graph (Figure 2b) perfectly reflects the ADSL gene model (Figure 1a). It shows two alternatively spliced variants: P_1 = exon 1–11 → exon 12 → exon 13 and P_2 = exon 1–11 → exon 13.

We aligned the sequence of P_1 with GenBank entry XM_010008.5 (ADSL mRNA). Except for the first 3 bases, P_1 aligned with only one mismatch to the mRNA (99.9% sequence identity). The mRNA showed an additional region of about 30 bases at the 5′-end and an unmatched region of about 60 bases at the 3′-end. This corresponds nicely to the trimmed sequence in our error correction step. We also compared the sequence of P_2 with GenBank entry AF067854.1 (alternatively spliced ADSL mRNA). Except for the first 3 bases, P_2 aligned perfectly to the mRNA. The mRNA showed an additional unmatched region of about 30 bases at the 5′-end and P_2 showed an unmatched region of about 180 bases at the 3′-end. The missing sequence at the 5′-end of P_2 can be explained by sequence trimming, while the additional sequence at the 3′-end corresponds to a shorter transcript.

We emphasize that Figure 2b (which perfectly fits the ADSL gene model) represents the original splicing graph (Figure 1a) after thresholding. Thresholding removes all 'weak' alternative splicing junctions (i.e. splicing junctions supported by less than 3 ESTs). Some of these removed junctions may be EST artifacts while others may correspond to rarely appearing, biologically important splicing variants. Splicing graphs like the one in Figure 2a may guide further PCR primer design to verify which of the putative transcripts are expressed in certain tissues.

To further evaluate the accuracy of our error correction procedure, we built the ADSL splicing graph with an ideal error correction procedure under the assumption that the genomic sequence is known (for similar approaches using genomic sequence, see Kan *et al.* (2001); Modrek *et al.* (2001)). We used sim4 (Florea *et al.*, 1998) to match the ESTs to their genomic template. Poorly aligned sequences (less than 95% identity, less than 80% of the sequence unmatched, or unmatched interior splice sites) were deleted. In the remaining sequences we corrected matched sequence parts correspondingly to their genomic counterparts and masked unmatched fragment ends. The resulting splicing graph is shown in Figure 2c. One can see that the blind experiment (i.e. assuming that genomic sequence is unavailable) produces roughly the same result as the 'ideal' experiment with genomic sequence available.

Analysis of TIGR Tentative Human Consensus sequences

We analyse Tentative Human Consensus sequences (THCs) and demonstrate that the potential problems of conventional EST assembly approaches can be detected and corrected using splicing graphs.

We applied our splicing graph approach to the human *hippostasin* gene for which alternative splicing has been recently shown to be connected with prostate cancer (Nakamura *et al.*, 2001). The gene is about 9 kb long and contains 6 exons (Figure 3). The two isoforms have an overall length of about 1.3 kb (prostate form) and about 1.1 kb (brain form).

We downloaded the Tentative Human Consensus sequences THC683186 (1337 bp) and THC683187 (861 bp). Figure 4 shows the cartoons of the assemblies (provided by TIGR) showing the location of individual ESTs within the consensus sequence. THC683186 corresponds to the full prostate-type hippostasin while THC683187 corresponds to brain-type hippostasin with an end-truncation of about 300 bases in exon 6. The TIGR assembly contains at this position only the suspicious read zd29b01.r1 with a deletion of 30 bases or an unreported skipped exon. We hypothesize that this truncation is due to the fact that all reads which could possibly extend zd29b01.r1 were assembled into THC683186 yielding the degenerate assembly.

For comparison, we show the corresponding splicing graph generated only from the inputs of the two TIGR assemblies in Figure 5a and a schematic representation indicating the positions of THC683186, THC683187, and zd29b01.r1 in Figure 5b. The splicing graph contains four paths $P_1 = 1 \to 3 \to 4 \to 5$, $P_2 = 1 \to 3 \to 5$, $P_3 = 2 \to 3 \to 4 \to 5$ and $P_4 = 2 \to 3 \to 5$. Here P_1 and P_3 correspond to the (untruncated!) brain- and prostate-forms while P_2 and P_4 correspond to similar transcripts containing the suspicious read zd29b01.r1. We also computed the hippostasin splicing graph visualization (see Figure 5c) based on the corresponding UniGene cluster Hs.57771. This graph indicates further possible alternative transcripts missed in the TIGR assemblies. After thresholding ($t = 3$), we obtain a splicing graph (see Figure 5d) which agrees with the gene model postulated in Nakamura *et al.* (2001).

Tropomyosin 1 (alpha) splicing graph

Among human genes, the tropomyosin 1 alpha gene encodes one of the most extensively alternatively spliced transcripts known (Balvay and Fiszman, 1994). A visualization of the corresponding UniGene cluster Hs.77899 with about 1000 EST reads is shown in Figure 6. Even after thresholding with $t = 7$ one can see that the corresponding splicing graph is very complicated, with thousands of potential transcripts. We emphasize that each of these transcripts deserves further analysis since every splicing junction in these transcripts is supported by at least 7 EST reads. The splicing graph would be an invaluable tool for designing PCR experiments or DNA arrays to study these potential splicing variants.

DISCUSSION

In contrast to other approaches, our algorithm does not assemble ESTs into linear sequences, but integrates the whole data set into one unambiguously defined splicing

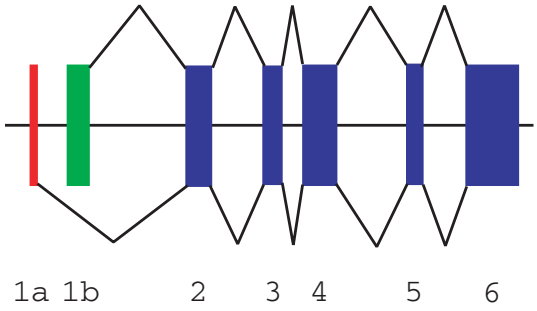


Fig. 3. Gene model of hippostasin gene. Exon 1a is spliced out in prostate-type hippostasin, while exon 1b is spliced out in brain-type hippostasin. (Not drawn to scale!)

Fig. 4. Cartoon of the assemblies THC683186 (left) and THC683187 (right), showing the location of individual ESTs within the consensus sequences. Adapted from the TIGR Human Gene Index.

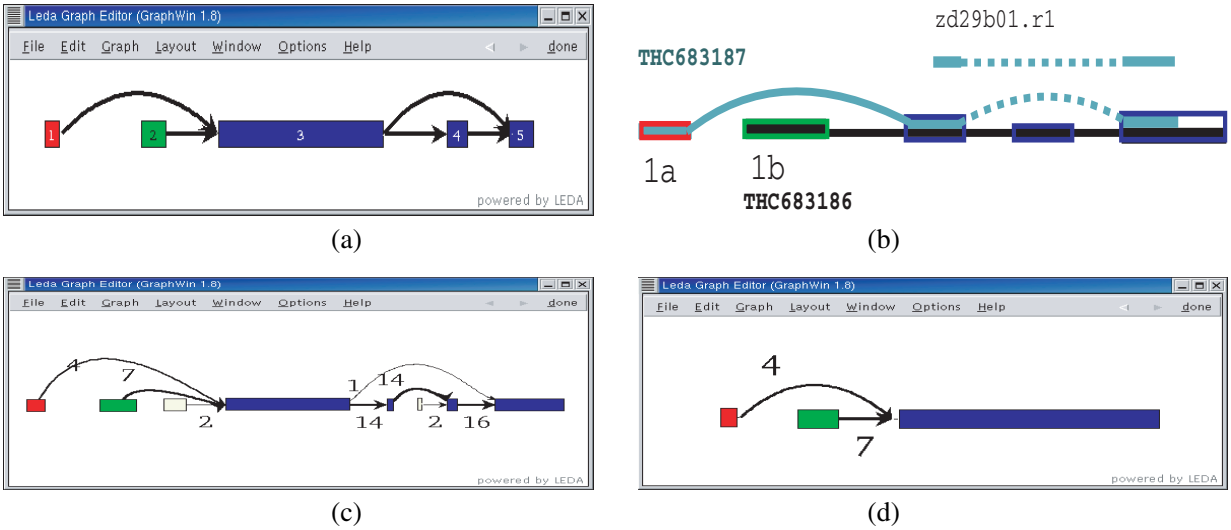
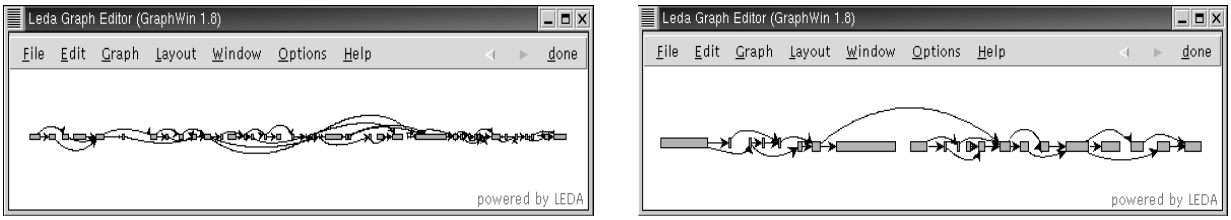


Fig. 5. (a) Splicing graph of THC683186 and THC683187 using word size $k = 20$. (b) Schematic representation (THC683186 marked black, THC683187 marked blue grey) indicating the locations of EST zd29b01.r1. (c) Hippostasin splicing graph of UniGene cluster Hs.57771 using word size $k = 20$ and referring to genomic sequence for error correction. (d) The same graph after thresholding ($t = 3$). (Colors indicate matching parts in the gene model.)



(a) (b)

Fig. 6. (a) Tropomyosin 1 (alpha) splicing graph of UniGene cluster Hs.77899. (b) The same graph after thresholding ($t = 7$).

graph. The splicing graph visualizes the alternative splicing variants. Although sequence variations on a smaller scale (e.g. small exon sliding events or SNPs) may be suppressed by the error correction algorithm, we emphasize that these variations are not lost and are easily restored at the later consensus stage.

The resulting splicing graph is a compact but biologically meaningful representation of the huge EST/cDNA data set, an important requirement for any subsequent exploratory data analysis. In the ideal situation, the transcripts correspond to paths in the splicing graph and splice sites correspond to vertices with in/outdegree larger than one.

Our assembly program has been tested on simulated data, on UniGene EST-clusters, and on THCs of TIGR Human Gene Index. In most cases with sufficient coverage, the quality of the assembled sequence is excellent. While transcript truncations are hard to determine, the splicing graph generates many putative transcripts and accurately depicts alternative splicing by bifurcations, even in the absence of known genomic sequence. For THCs the splicing graph is usually very 'clean', showing mostly biologically meaningful and confirmed splicing events. The splicing graphs built from EST clusters depend on data quality and stipulated coverage. Their complexity usually decreases with increasing quality of EST reads and increasing coverage threshold. Unfortunately, there is an unavoidable trade-off between reducing the representation complexity (by thresholding or error correction) and the danger of eliminating biologically meaningful information (by incorrectly homogenizing differing transcripts). Another problem is caused by overlapping genes and paralogs that may 'glue' different splicing graphs together.

In the future, we plan to build a catalog of splicing graphs and putative splicing variants as well as SNPs for all UniGene clusters. This immediately raises the (still unsolved) question of how to distinguish alternative splices from 'biological noise', and how to assess their biological importance. This will ultimately need additional experiments.

ACKNOWLEDGMENTS

The authors are grateful to Tim Beißbarth, Eivind Coward, Stefan Haas, Antje Krause, Christopher Lee, and Zufar Mulyukov for help and very valuable discussions. This work was supported by National Institute of Health grant 1 R01 HG02366-01.

REFERENCES

Balvay,L. and Fiszman,M. (1994) Analysis of the diversity of tropomyosin isoforms. *C. R. Seances Soc. Biol. Fil.*, **188**, 527–540.

Bouck,J., Yu,W., Gibbs,R. and Worley,K. (1999) Comparison of gene indexing databases. *Trends Genet.*, **15**, 159–162.

Burke,J., Wang,H., Hide,W. and Davison,D. (1998) Alternative gene form discovery and candidate gene selection from gene indexing projects. *Genome Res.*, **8**, 276–290.

Carstens,R., Eaton,J., Krigman,H., Walther,P. and Garcia-Blanco,M. (1997) Alternative splicing of fibroblast growth factor receptor 2 (FGF-R2) in human prostate cancer. *Oncogene*, **15**, 3059–3065.

Coward,E., Haas,S. and Vingron,M. (2002) SpliceNest: visualizing gene structure and alternative splicing based on EST clusters. *Trends Genet.*, **18**, 53–55.

Florea,L., Hartzell,G., Zhang,Z., Rubin,G. and Miller,W. (1998) A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.*, **8**, 967–974.

Graveley,B. (2001) Alternative splicing: increasing diversity in the proteomic world. *Trends Genet.*, **17**, 100–107.

Green,P. (1994) Phrap documentation. http://www.phrap.org.

Heuze,N., Olayat,S., Gutman,N., Zani,M. and Courty,Y. (1999) Molecular cloning and expression of an alternative hKLK3 transcript coding for a variant protein of prostate-specific antigen. *Cancer Res.*, **59**, 2820–2824.

Hu,Y., Tanzer,L., Cao,J., Geringer,C. and Moore,R. (1998) Use of long RT-PCR to characterize splice variant mRNAs. *Biotechniques*, **25**, 224–229.

Huang,X. and Madan,A. (1999) CAP3: A DNA sequence assembly program. *Genome Res.*, **9**, 868–877.

Idury,R. and Waterman,M. (1995) A new algorithm for dna sequence assembly. *J. Comput. Biol.*, **2**, 291–306.

Kan,Z., Rouchka,E., Gish,W. and States,D. (2001) Gene structure prediction and alternative splicing analysis using genomically aligned ESTs. *Genome Res.*, **11**, 889–900.

Kmoch,S., Hartmannová,H., Stibùrková,B., Krijt,J., Zikánova,M. and Sebesta,I. (2000) Human adenylosuccinate lyase (ADSL), cloning and characterization of full-length cDNA and its isoform, gene structure and molecular basis for ADSL deficiency in six patients. *Hum. Mol. Genet.*, **9**, 1501–1513.

Lisacek,F., Traini,M., Sexton,D., Harry,J. and Wilkins,M. (2001) Strategy for protein isoform identification from expressed sequence tags and its application to peptide mass fingerprinting. *Proteomics*, **1**, 186–193.

Melhorn,K. and Näher,S. (1999) *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press.

Mironov,A., Fickett,J. and Gelfand,M. (1999) Frequent alternative splicing of human genes. *Genome Res.*, **9**, 1288–1293.

Modrek,B. and Lee,C. (2001) A genomic view of alternative splicing. *Nature Genet.*, **30**, 13–19.

Modrek,B., Resch,A., Grasso,C. and Lee,C. (2001) Genome-wide detection of alternative splicing in expressed sequences of human genes. *Nucleic Acids Res.*, **29**, 2850–2859.

Nakamura,T., Mitsui,S., Okui,A., Kominami,K., Nomoto,T., Ukimura,O., Kawauchi,A., Miki,T. and Yamaguchi,N. (2001) Alternative splicing isoforms of hippostasin (PRSS20/KLK11) in prostate cancercell lines. *Prostate*, **49**, 72–78.

Pevzner,P. (1989) *l*-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, **7**, 63–73.

Pevzner,P., Tang,H. and Waterman,M. (2001) An Eulerian path approach to DNA fragment assembly. *Proc. Natl Acad. Sci. USA*, **98**, 9748–9753.

Picoult-Newberg,L., Ideker,T., Pohl,M., Taylor,S., Donaldson,M., Nickerson,D. and Boyce-Jacino,M. (1999) Mining SNPs from EST databases. *Genome Res.*, **9**, 167–174.

Schmitt,A., Specht,T., Beckmann,G., Dahl,E., Pilarsky,C., Hinzmann,B. and Rosenthal,A. (1999) Exhaustive mining of EST libraries for genes differentially expressed in normal and tumour tissues. *Nucleic Acids Res.*, **27**, 4251–4260.

Sutton,G., White,O., Adams,M. and Kerlavage,A. (1995) TIGR assembler: a new tool for assembling large shotgun sequencing projects. *Genome Science and Technology*, **1**, 9–19.

Zhuo,D., Zhao,W., Wright,F., Yang,H., Wang,J., Sears,R., Baer,T., Kwon,D., Gordon,D., Gibbs,S., Dai,D., Yang,Q., Spitzner,J., Krahe,R., Stredney,D., Stutz,A. and Yuan,B. (2001) Assembly, annotation, and integration of UNIGENE clusters into the human genome draft. *Genome Res.*, **11**, 904–918.