View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by HAL-Rennes 1

kisSplice, détection d'évènements d'épissages alternatifs dans les données RNA-seq

Gustavo Akio Tominaga Sacomoto, J. Kielbassa, Pavlos Antoniou, Rayan Chikhi, Raluca Uricaru, Marie-France Sagot, Pierre Peterlongo, Vincent Lacroix

To cite this version:

Gustavo Akio Tominaga Sacomoto, J. Kielbassa, Pavlos Antoniou, Rayan Chikhi, Raluca Uricaru, et al.. kisSplice, détection d'évènements d'épissages alternatifs dans les données RNA-seq. [Research Report] RR-7852, INRIA. 2011, pp.12. <hal-00654249v2>

HAL Id: hal-00654249
https://hal.inria.fr/hal-00654249v2

Submitted on 20 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

KISSPLICE: *de-novo calling alternative splicing events from RNA-seq data*

Gustavo A.T. Sacomoto — Janice Kielbassa — Pavlos Antoniou — Rayan Chikhi — Raluca Uricaru — Marie-France Sagot — Pierre Peterlongo* — Vincent Lacroix

N° 7852

Decembre 2011

Computational Biology and Bioinformatics

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7852--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

kisSplice: de-novo calling alternative splicing events from RNA-seq data

Gustavo A.T. Sacomoto , Janice Kielbassa , Pavlos Antoniou ,
Rayan Chikhi , Raluca Uricaru , Marie-France Sagot , Pierre
Peterlongo[⋆] , Vincent Lacroix[*]

Theme : Computational Biology and Bioinformatics
Computational Sciences for Biology, Medicine and the Environment
Équipes-Projets Bamboo & Symbiose

Rapport de recherche n° 7852 — Decembre 2011 — 13 pages

Abstract: In this paper, we address the problem of identifying polymorphisms in RNA-seq data when no reference genome is available, without performing an assembly of the transcripts. Based on the fundamental idea that each polymorphism will correspond to a recognisable pattern in a De Bruijn graph constructed from the RNA-seq reads, we propose a general model for all polymorphisms in such graphs. We then introduce an exact algorithm to extract alternative splicing events and show that it enables to identify more correct events than current transcriptome assemblers. Additionally, when we applied our method on a 71M reads dataset from human, we were able to identify 3884 events, out of which 57% are not present in the annotations, which confirms recent estimates showing that the complexity of alternative splicing has been largely underestimated so far.

Key-words: algorithms, genome, bioinformatics, NGS, transcripts splicing, RNA-seq

---

[*] Corresponding authors

kisSplice, détection d'évènements d'épissages alternatifs dans les données RNA-seq

Résumé : Nous proposons une approche permettant l'identification de polymorphismes dans les données RNA-seq. Nous nous plaçons dans le cadre où l'on ne dispose pas de génome de référence et où l'on ne construit pas par assemblage un tel génome. Notre approche s'appuie sur l'idée fondamentale que chaque polymorphisme génère un motif reconnaissable dans le graphe de De-Bruijn construit à partir des données RNA-seq. Nous proposons un modèle général pour tout type de polymorphisme avant de proposer un algorithme exact pour la détection d'évènements d'épissages alternatifs. Nous montrons que cette approche détecte plus finement ces évènements que les approches basées sur l'assemblage des données. Nous avons appliqué notre approche sur un jeux de données de 71 millions de lectures provenant de cellules humaines. Ceci nous a permis d'identifier 3884 évènements, dont 57% n'étaient pas référencés dans les annotations. Ceci confirme les estimations récentes qui démontrent que la complexité de l'épissage alternative à largement été sous estimé jusqu'à présent.

Mots-clés : algorithms, génome, bio-informatique, NGS, épissage alternatifs, RNA-seq

1 Introduction

Thanks to recent technological advances, sequencing is no longer restricted to genomes and can now be applied to many new areas, including the study of gene expression and splicing. The so-called RNA-seq protocol consists in applying fragmentation and reverse transcription to an RNA sample followed by sequencing the ends of the resulting cDNA fragments. The short sequencing reads then need to be reassembled in order to get back to the initial RNA molecules. A lot of effort has been put on this assembly task [4], whether in the presence or in the absence of a reference genome but the general goal of identifying and quantifying all RNA molecules initially present in the sample remains hard to reach. The main challenge is certainly that reads are short, and can therefore be ambiguously assigned to multiple transcripts. In particular, in the case of alternative splicing (AS for short), reads stemming from constitutive exons can be assigned to any alternative transcript containing this exon. Finding the correct transcript is often not possible given the data we have, and any choice will be arguable. As pointed out in Martin and Wang's review [4], reference-based and de novo assemblers each have their own limitations. Reference-based assemblers depend on the quality of the reference while only a small number of species currently have a high-quality reference genome available. De novo assemblers implement reconstruction heuristics which may lead them to miss unfrequent alternative transcripts while highly similar transcripts are likely to be assembled into a single transcript. We argue here that it is not always necessary to aim at the difficult goal of assembling full-length molecules. Instead, identifying the variable parts between molecules (polymorphic regions) is already very valuable and does not require to solve the problem of assigning a constitutive read to the correct transcript.

We therefore focus in this paper on the simpler task of identifying polymorphisms in RNA-seq data. Three kinds of polymorphisms have to be considered: i) alternative splicing that produces several alternative transcripts for a same gene, ii) SNPs (single nucleotide polymorphism) that may also produce several transcripts for a same gene whenever they affect transcribed regions, and iii) CNVs (copy number variation) which affect the number of copies of tandem repeats. Our contribution in this paper is double: we first give a general model which captures these three types of polymorphism by linking them to characteristic structural patterns of a De Bruijn graph (DBG for short) built from a set of RNA-seq reads, and second, we propose a method dedicated to the problem of identifying alternative splicing events in a DBG.

Extracting AS events from a splicing graph has been studied before [13] but a significant difference between splicing graphs and De Bruijn graphs is that in the former, nodes are genomically ordered (through the use of a reference annotated genome) therefore leading to a DAG, whereas DBGs are general graphs, that furthermore do not require any additional information to be built.

When no reference genome is available, efforts have focused on assembling the full-length RNA molecules, not the variable parts which are our interest here. Most RNA-seq assemblers [1, 12, 15] do rely on the use of a DBG, but, since the primary goal of an assembler is to produce the longest contigs, heuristics are applied, such as tip or bubble removal, in order to linearise the graph. The application of such heuristics results in a loss of information which may in fact be crucial if the goal is to study polymorphism.

The identification of SNPs without a reference genome has been studied in the context of genome sequencing [8, 10], not transcriptome sequencing. To our knowledge, the identification of alternative splicing events without a reference genome has not been studied before.

The paper is organised as follows. We first present the model (Section 2) linking structures of the DBG for a set of RNA-seq reads to polymorphism, and then introduce a method, that we call KISSPLICE, for identifying DBG structures associated with AS events (Section 3). We show in Section 4 the results of using KISSPLICE on simulated and real data, and compare them to the results obtained with Trinity [1], the currently most performing de novo transcriptome assembler.

2 De-Bruijn graph models

2.1 De-Bruijn graph

DBGs were first used in the context of genome assembly in 2001 by Pevzner *et al.* [9]. In 2007, Medvedev *et al.* [5] modified the definition to better model DNA as a double stranded molecule. In such a context, a DBG is a bidirected multigraph, each node N storing a sequence w and its reverse complement $\overline{w}$. The sequence w, denoted by $F(N)$, is the forward sequence of N, while $\overline{w}$, denoted by $R(N)$, is the reverse complement sequence of N. An arc exists from node N_1 to node N_2 if the suffix of length $k-1$ of $F(N_1)$ or $R(N_1)$ overlaps perfectly with the prefix of $F(N_2)$ or $R(N_2)$. Each arc is labelled with a string in $\{F,R\}^2$.

The first letter of the arc label indicates which of $F(N_1)$ or $R(N_1)$ overlaps $F(N_2)$ or $R(N_2)$, this latter choice being indicated by the second letter. Because of reverse complements, there is an even number of arcs in the DBG: if there is an arc from N_1 to N_2 then, necessarily, there is an arc from N_2 to N_1 (*e.g.* if the first arc has label FF then the second has label RR). Examples of DBGs are presented in Figure 1.

Definition 1 (Valid path)**.** *The traversal of a node is said to be valid if the rightmost label (F or R) of the arc entering the node is equal to the leftmost label of the arc leaving the node.*

A path in the graph is valid if for each node involved in the path, its traversal is valid, that is, each pair of adjacent arcs in the path are labelled, respectively, XY *and* YZ *with* $X,Y,Z \in \{R,F\}$.

Examples of valid paths are shown in Figure 1.a$_0$. For instance, the path starting from the leftmost encircled node, going by the upper path to the rightmost encircled node is valid. More generally, in Figure 1, all paths except those represented by a couple of red arrows are valid.

A DBG can be compressed without loss of information by merging *simple* nodes. A simple node denotes a node linked to at most two other nodes. Two simple nodes are merged into one by removing the redundant information (the overlapping part). A valid path composed by $i > 1$ uncompressed nodes is compressed into one node storing a sequence of length $k+(i-1)$ as each node adds one new character to the first node. Figure 1.a represents the compressed DBG shown in Figure 1.a$_0$. In the remaining of the paper, we denote by cDBG a compressed DBG.

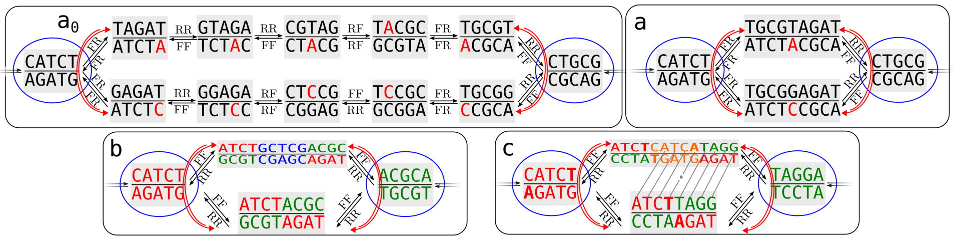
a0
CATCT
AGATG
TAGAT
ATCTA
GTAGA
TCTAC
CGTAG
CTACG
TACGC
GCGTA
TGCGT
ACGCA
GAGAT
ATCTC
GGAGA
TCTCC
CTCCG
CGGAG
TCCGC
GCGGA
TGCGG
CCGCA
CTGCG
CGCAG
RR
FF
RF
FR
a
TGCGTAGAT
ATCTACGCA
TGCGGAGAT
ATCTCCGCA
b
ATCTGCTCGACGC
GCGTCGAGCAGAT
ATCTACGC
GCGTAGAT
ACGCA
TGCGT
c
ATCTCATCATAGG
CCTATGATGAGAT
ATCTTAGG
CCTAAGAT
TAGGA
TCCTA

Figure 1: Part of non-compressed **(a_0)** and compressed **(a,b,c)** de Bruijn graphs ($k = 5$). Each node contains a word (upper text of each node) and its reverse complement (lower text of each node). In the uncompressed graph, the word is a k-mer. Encircled nodes are switching with respect to red paths (pointed out by red arrows). **(a_0,a)** Mouth due to a substitution (red letter). Starting from the forward strand in the leftmost (switching) node would generate the sequences $CATCTACGCAG$ (upper path) and $CATCTCCGCAG$ (lower path).**(b)** Mouth due to the skipped exon $GCTCG$. This mouth is generated by the sequences $CATCTACGCA$ and $CATCTGCTCGACGCA$. **(c)** Mouth due to a variation in the number of copies (CNV) of an inexact repeat. This mouth is generated by the sequences $CATCTTAGGA$ and $CATCTCATCATAGGA$, where $CATC\mathbf{T}CATC\mathbf{A}$ is an inexact tandem repeat.

2.2 Mouth patterns in the cDBG

Polymorphisms (*i.e.* variable parts) in a transcriptome or a genome, will correspond to recognisable patterns in the cDBG, which we call a **mouth**. Intuitively, the variable parts will correspond to alternative paths and the common parts will correspond to the beginning and end points of these paths. We now formally define the notion of mouth.

Definition 2 (Node switching with respect to a path)**.** *A node is switching with respect to a path if this path is invalid during the traversal of this node.*

Definition 3 (Mouth)**.** *In the cDBG, a mouth is a simple cycle involving at least three distinct nodes such that exactly two nodes SN_{left} and SN_{right} are switching w.r.t. the path of the cycle. By definition, two valid paths exist between these two switching nodes. In the remaining of the paper, we refer to these two paths as the paths of the mouth. If they differ in length, we refer to, respectively, the longer and the shorter path of the mouth.*

Figure 1 presents four mouths. For each one, their switching nodes are encircled in blue.

In general, any process generating patterns asb and $as'b$ in the sequences, with $a, b, s, s' \in \Sigma^*$, $|a| \geq k, |b| \geq k$ and $s \neq s'$, creates a mouth in the cDBG. Indeed, all k-mers entirely contained in a (resp. b) compose the node SN_{left} (resp. SN_{right}). Since $|a| \geq k$ and $s \neq s'$, there is at least one pair of k-mers, one in as and the other in as', sharing the $k-1$ prefix and differing by the last letter, thus creating a branch in SN_{left} from which the two paths in the mouth diverge. The same applies for sb, $s'b$ and SN_{right}, where the paths merge again. All k-mers contained in s (resp. s') and in the junctions as and sb (resp. as'

and $s'b$) compose the paths of the mouth. In the case of AS events and CNVs, s is empty and the shorter path is composed of k-mers covering the junction ab.

This model is general as it captures SNPs, CNVs and AS events, as shown in Figure 1. The main focus of the algorithm we present in this paper is the detection of mouths generated by AS events.

2.3 Mouths generated by alternative splicing events

A single gene may give rise to multiple alternative spliceforms through the process of alternative splicing. Alternative spliceforms differ locally from each other by the inclusion or exclusion of subsequences. These subsequences may correspond to exons (exon skipping), exon fragments (alternative donor or acceptor sites) or introns (intron retention) as shown in Figure 2. A splicing event corresponds to a local variation between two alternative transcripts. It is characterised by two common sites and a variable part. In the cDBG, the common sites correspond to the switching nodes and the variable part to the longer path. The shorter path is composed of at most $k-1$ k-mers, *i.e.* a path of length at most $2k-2$ in the cDBG, corresponding to the junction between the two common sites. An example is given in Figure 1.b.

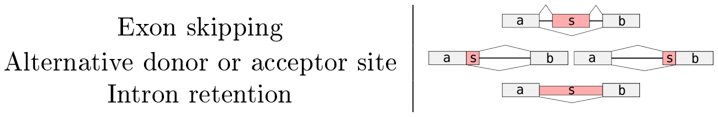

Figure 2: Alternative splicing events generating a mouth in the DBG. All these events create a mouth in the DBG or cDBG, in which the shorter path is composed by k-mers covering the ab junction. This path, composed by $k-1$ nodes in the DBG, is compressed into a sequence of length $2k-2$ in the cDBG (Fig 1.b)

The shorter path of a mouth generated by an AS event has length exactly $2k-2$ iff the last nucleotide (nt for short) of the variable part is distinct from the last nt of the left switching node SN_{left}, and the first nt of the variable part is distinct from the first nt of the right switching node SN_{right}. Otherwise, the two alternative paths join earlier and the shorter path may be smaller. In human, 99% of the annotated exon skipping events yield a mouth with a shorter path length between $2k-8$ and $2k-2$.

2.4 Mouths generated by SNPs and CNVs

Polymorphism at the genomic level will necessarily also be present at the transcriptomic level whenever it affects transcribed regions. Two major kinds of polymorphism can be observed at the genomic level: SNPs and CNVs. As shown in Fig. 1, these two types of polymorphism generate mouths in the cDBG.

However, these mouths have characteristics which enable to differentiate them from mouths generated by AS events. Indeed, mouths generated by SNPs exhibit paths of length $2k-1$, which is larger than $2k-2$, the maximum size of the shorter path in a mouth generated by an AS event.

CNVs may generate mouths with a similar path length as mouths generated by splicing events, but the sequences of the paths exhibit a clear pattern which can be easily identified: the longer path contains an inexact repeat. More precisely, as outlined in Fig 1.c, it is sufficient to compare the shorter path with one of the ends of the longer path.

Finally, genomic insertions or deletions (indels for short) may also generate mouths with similar path lengths as mouths generated by splicing events. In this case, the difference of length between the two paths is usually smaller (less than 3 nt for 85% of indels in human transcribed regions [16] whereas it is more than 3 nt for 99% of AS events). In our method, when the difference of path lengths is below 3, we classify the mouth as an indel. Otherwise, we do not decide, which means that a fraction of the mouths we report as AS events will correspond to indels.

In the following, we focus on mouths generated by AS events. We do provide as a collateral result two additional collections of mouths: one corresponding to putative SNPs and one corresponding to putative CNVs. The post-treatment of these collections to discard false positives caused by sequencing errors is beyond the scope of this paper.

3 The kisSplice algorithm

The kisSplice algorithm detects in the cDBG all the mouth patterns generated by AS events, *i.e.* the mouths having a shorter path of length at most $2k-2$. Essentially, the algorithm enumerates all the cycles verifying the two following criteria: i) the path obtained by following all the nodes of the cycle generates exactly two nodes that are switching for this path, and ii) the length of the shorter path linking the two switching nodes must be no longer than $2k-2$. Further criteria are applied to make the algorithm more efficient without loss of information, and to eliminate polymorphism events that do not correspond to alternative splicing.

Since the number of cycles in a graph may be exponential with the size of the graph, the naive approach of enumerating all cycles of the cDBG and verifying which of them satisfy our conditions is only viable for very small cases. Nonetheless, kisSplice is able to enumerate a potentially exponential number of mouths for real-sized data in very reasonable time and memory. This is in part due to the fact that, previous to cycle enumeration, the graph is pre-processed in a way that, along with the pruning criteria of Step 4, is responsible for a good performance in practice.

kisSplice is indeed composed of six main steps which are described next. The pre-processing just mentioned corresponds to Step 2. As far as we know, it is the first time it is used in conjunction with cycle enumeration.

Step 1. Construction of the cDBG of the reads of one or two RNA-seq experiments. Each node contains the coverage of the corresponding k-mer in each experiment. In order to get rid of most of the sequencing errors, nodes with a minimal coverage of 1 may be removed.

Step 2. Biconnected component (BCC for short) decomposition. A connected undirected graph is *biconnected* if it remains connected after the removal of any vertex. A BCC of an undirected graph is a maximal biconnected subgraph. Moreover, it is possible to show that the BCCs of an undirected graph form a partition of the edges with two important properties: every cycle is contained in exactly one BCC, and every edge not contained in a cycle forms a singleton BCC.

Applying on the underlying undirected graph of the cDBG Tarjan's lowpoint method [17] which performs a modified depth-first search traversal of the graph, Step 2 detects all BCCs, and discards all singleton ones that could not contain any mouth. Without modifying the results, this considerably reduces the memory footprint and the computation time of the whole process. To give an idea of the effectiveness of this step, the cDBG of a 5M dataset had 1.7M nodes, but the largest BCC only 2961 nodes.

Step 3. Four-nodes compression. Single substitution events (SNPs, sequencing errors) generate a large number of cycles themselves included into bigger ones, creating a combinatorial explosion of the number of possible mouths. This step of KISSPLICE detects and compresses all mouths composed by just four nodes: two switching nodes and two *non-branching internal nodes* each storing equal length sequences differing by just one position. Figure 1.a shows an example of a four-nodes mouth. Four-nodes mouths are output as potential SNPs and then reduced to a three-nodes path. The two non-branching internal nodes are merged into one, storing a consensus sequence where the unique substitution is replaced by a wildcard character.

Step 4. Mouths enumeration. The cycles are detected in the cDBG using a backtracking procedure [19] augmented with two pruning criteria. The exploration of one cycle is stopped if the path contains more than two nodes that are switching relative to the path that is being followed, or the length of the shorter path is bigger than $2k - 2$. This approach has the same theoretical time complexity of Tiernan's algorithm for cycle enumeration [19], which is worse than Tarjan's [18] polynomial delay algorithm but it appears to be not immediate how to use the pruning criteria with the latter while preserving its theoretical complexity. We however were able to show that in practice, the pruning criteria are very effective for the type of instances we are dealing with. Indeed, we compared the three following implementations on a 1M reads dataset: i) Tiernan ii) Tarjan iii) Tiernan with prunings (our method). The results clearly showed that, while Tarjan (22 min) outperforms Tiernan (32 min), both are clearly outperformed when the prunings are used (4 s).

Step 5. Results filtration and classification. The two paths of each mouth are aligned. If the whole of the shorter path aligns with high similarity to the longer path, we decide that the mouth is due to a CNV (see Section 2.4). After this alignment, a mouth is classified either as an AS event, a CNV, or a small indel (less than 3 nt).

Step 6. Events sorting. The mouths obtained in each category are ranked and sorted w.r.t. this rank. The higher the difference of expression between the two paths, the higher the rank. Formally the rank is equal to $\sum_{i=1,2} \frac{\max(cov_i(longer),cov_i(shorter))}{\min(cov_i(longer),cov_i(shorter))}$, where $cov_i(longer)$ (resp. $cov_i(shorter)$) is the coverage in experiment i of the longer (resp. shorter) path.

4 Results

4.1 Simulated data

In order to assess the sensitivity and specificity of our approach, we simulated the sequencing of genes for which we are able to control the number of alternative transcripts. We show that the method is indeed able to recover AS events whenever the alternative transcripts are sufficiently expressed. For our sensitivity tests, we used simulated RNA-seq single end reads (75bp) with sequencing errors. We first tested a pair of transcripts with a 200 nt skipped exon. Simulated reads were obtained with MetaSim [11] which is a reference software for simulating sequencing experiments. As in real experiments, it produces heterogeneous coverage and authorises to use realistic error models.

By definition, our method is able to identify an AS event if the two alternative transcripts are locally covered by reads which overlap with at least k-1 nt. Otherwise, the mouth is only partially present in the DBG and we cannot detect it. In principle, if the coverage was constant, it would be sufficient that the transcripts have a 2X coverage. In practice, the coverage is very heterogeneous and the average coverage for each transcript has to be higher to ensure that the local coverage of the exon is sufficient for KISSPLICE to detect the AS event.

In order to find the minimum coverage for which we are able to work, we created datasets for several coverages (from 4X to 20X, which corresponds to 60 to 300 Reads Per Kilobase or RPK for short), with 3 samples for each coverage, and tested them with different values of k ($k = 13, \ldots 41$). For coverages below 8X (120 RPK), KISSPLICE found the correct event in some but not all of the 3 tested samples. The failure to detect the event was due to the heterogenous and thus locally very low coverage around the skipped exon, *e.g.* some nt were not covered by any read or the overlap between the reads was smaller than k-1. Above 8X (120 RPK), KISSPLICE detected the correct exon skipping event in all samples.

For each successful test, there was a maximal value k_{max} for k above which the event was not found, and a minimal value k_{min} below which KISSPLICE also reported false positive events. Indeed, if k is too small, then the pattern ab, $as'b$, with $|a| \geq k$, $|b| \geq k$ is more likely to occur by chance in the transcripts, therefore generating a mouth in the DBG. Between these two thresholds, KISSPLICE found only one event: the correct one. The values of k_{min} and k_{max} are clearly dependent on the coverage of the gene. At 8X (120 RPK), the 200 nucleotides exon was found between $k_{min} = 17$ and $k_{max} = 29$. At 20X (300 RPK), it was found for $k_{min} = 17$ and $k_{max} = 39$. We performed similar tests on other datasets, varying the length of the skipped exon. As expected, if the skipped exon is shorter (longer), KISSPLICE needed a lower (higher) coverage to recover it.

To compare our results to existing assemblers, we ran Trinity [1] on the same datasets. We found that Trinity was able to recover the AS event in all 3 samples only when the coverage was above 18X (270 RPK), which clearly shows that KISSPLICE is more sensitive. This can be explained by the fact that Trinity uses heuristics which consist in discarding a k-mer in the DBG whenever it is 20 times less frequent than an alternative k-mer branching at the same location in the DBG. Another reason is that Trinity works with a fixed value of k ($k = 25$), while KISSPLICE is flexible in the choice of k. Thus, Trinity missed some events, where KISSPLICE was able to detect them with $k < 25$. Finally, another advantage of KISSPLICE is its speed; for all our toy data sets KISSPLICE ran much faster than Trinity (*e.g.* 0.6 s vs. 10.5 s for one of the samples).

In order to have an idea of the proportion of alternative transcripts which are above the resolution level of KISSPLICE, we used Cufflinks [20] to compute the expression levels of transcripts in a real dataset, composed of 6M RNA-seq reads from *Drosophila*. The expression level of a transcript is generally given in Reads per Kilobase per Million mapped reads (RPKM) [6]. If the sequencing effort is of 1M reads, then the RPK and RPKM measures coincide. If the sequencing effort is 6M reads, then it means we can deal with transcripts with an expression level of 120/6=20 RPKM. We found that 87% of the transcripts detected by Cufflinks are above this threshold.

All these results were obtained using a minimal k-mer coverage (mkC for short) of 1. We also tested with mkC=2 (*i.e.* k-mers present only once in the dataset are discarded), leading to the same main behaviour. We noticed however a loss in sensitivity for both methods, but a significant gain in the running time. KISSPLICE found the event in all 3 samples for a coverage of 12X (180 RPK) which is still better than the sensitivity of Trinity for mkC=1, but much faster.

We also tested simulated RNA-seq data without sequencing errors and with a uniform coverage. For these idealised conditions also, KISSPLICE performed always better than Trinity. Interestingly, our method performed better when the sequencing coverage was not uniform, as in real sequencing experiments. Even though this result seems counter-intuitive, this can be explained by the fact that KISSPLICE is only affected by a low coverage of the skipped exon, whereas whole-transcript assemblers like Trinity are affected by low coverage of both the skipped and the constitutive exons.

4.2 Real data

We further tested our method on RNA-seq data from human. Even though we do not use any reference genome in our method, we applied it to cases where an annotated reference genome is indeed available in order to be able to assess if our predictions are correct.

We ran KISSPLICE with $k = 25$ and mkC=2 on the human dataset, which consists of 32M reads from brain and 39M reads from liver from the Illumina Body Map 2.0 Project (downloaded from the Sequence Read Archive, accession number ERP000546). The most time and memory consuming step was the DBG construction which we performed on a cluster. KISSPLICE identified 5923 biconnected components which contained at least one mouth, 664 of which consisted of mouths generated by CNVs and 1160 consisted of mouths generated by short indels (less than 3 nt). Noticeably, the BCCs which generated most cycles and were most time consuming were associated to CNVs. As these mouths are not of interest for KISSPLICE, this observation prompted us to introduce an additional parameter in KISSPLICE to stop the computation in a BCC if the number of cycles being enumerated reaches a threshold. This enabled us to have a significant gain of time. We however advise not to use this threshold if the purpose is to identify AS events associated to CNVs, which we did not address here.

For each of the 4099 remaining BCCs, we tried to align the two paths of each mouth to the reference genome using Blat [2]. If the two paths align with the same initial and final coordinates, then we consider that the mouth is a real AS event. If they align with different initial and final coordinates, then we consider that it is a false positive. Out of the 4099 BCCs, 3884 (95%) corresponded to real AS events, while the remaining corresponded to false positives. A first inspection of these false positives lead to the conclusion that the majority of them correspond to chimeric transcripts. Indeed, the shorter path and the longer path both map in two blocks within the same gene, but the second block is either upstream of the first block, or on the reverse strand, in both cases contradicting the annotations and therefore suggesting that the transcripts are chimeric and could have been generated by a genomic rearrangement or a trans-splicing mechanism.

For each of the 3884 real cases, we further tried to establish if they corresponded to annotated splicing events. We therefore first computed all annotated AS events using AStalavista [14] and the UCSC Known Genes annotation [3]. Then, for each aligned mouth, we checked if the coordinates of the aligned blocks matched the splice sites of the annotated AS events. If the answer was positive, then we considered that the AS event we found was known, otherwise we considered it was novel. Out of a total of 3884 cases, we find that only 1658 are known while 2226 are novel. This clearly shows that current annotations largely underestimate the number of alternative transcripts per multi-exon genes as was also reported recently [21].

Additionally, we noticed that 890 BCCs contained more than one AS event, which all mapped to the same gene. This corresponds to complex splicing events which involve more than 2 transcripts. Such events have been described in Sammeth et al.[13]. Their existence suggests that more complex models could be established to characterise them as one single event, and not as a collection of simple pairwise events. An example of novel complex AS event is given in figure 3.

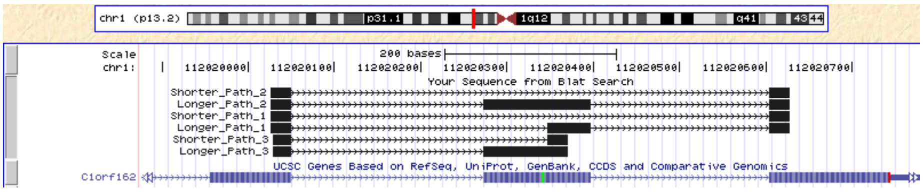
chr1 (p13.2)
p31.1
1q12
q41
43
44
Scale
chr1:
200 bases
112020000
112020100
112020200
112020300
112020400
112020500
112020600
112020700
Your Sequence from Blat Search
Shorter_Path_2
Longer_Path_2
Shorter_Path_1
Longer_Path_1
Shorter_Path_3
Longer_Path_3
UCSC Genes Based on RefSeq, UniProt, GenBank, CCDS and Comparative Genomics
C1orf162

Figure 3: BCC corresponding to a novel complex AS event. The intermediate annotated exon is either present, partially present, or skipped. The annotations (blue track) report only the version where it is present.

We also found the case where the same AS event maps to multiple locations on the reference genome (489 cases). This corresponds to a family of paralogous genes, which are "collectively" alternatively spliced. In this case, we are unable to decide which of the genes of the family are producing the alternative transcripts, but we do detect an AS event.

5 Conclusions and future work

This paper presents two main contributions. First, we introduced a general model for detecting polymorphisms in De Bruijn graphs, and second, we developed an algorithm, KISSPLICE, to detect AS events in such graphs. This approach avoids the assembly phase, which may be costly and uses heuristics that may lead to a loss of information. To our knowledge, this approach is new.

Results on human data show that this approach enables de-novo calling of AS events with a higher sensitivity than obtained by the approaches based on a full assembly of the reads. Results on real data show that 95% of the AS events we identify are real and that 43% of the real AS events correspond to annotated events, while the 57% remaining are novel, which confirms that the number of multi-exon genes which undergo alternative splicing is currently largely underestimated.

KISSPLICE is now available for download at http://alcovna.genouest.org/kissplice/ and can already be used to establish a more complete catalog of AS events in many species, whether they have a reference genome or not. There is of course room for future work. The KISSPLICE algorithm could be improved in several ways. The coverage could be used both for distinguishing SNPs from sequencing errors, and for quantifying the expression of AS events. Moreover, the sequences surrounding mouths could be locally assembled using a third party tool [7]. This would allow to output their context or the full contig they belong to.

Last, the complex structure of BCCs associated to CNVs seems to indicate that more work on the model and on the algorithms is required to efficiently deal with the identification of real CNV events, which may be highly intertwined.

References

[1] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, and D. A. T. *et. al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat Biotechnol*, 29(7):644–652, Jul 2011.

[2] W. J. Kent. BLAT–the BLAST-like alignment tool. *Genome Research*, 12(4):656–664, 2002.

[3] R. M. Kuhn, D. Karolchik, A. S. Zweig, T. Wang, and K. E. S. *et. al.* The UCSC Genome Browser Database: update 2009. *Nucleic Acids Research*, 37(Database issue):D755–D761, 2009.

[4] J. A. Martin and Z. Wang. Next-generation transcriptome assembly. *Nature Reviews Genetics*, pages 1–12, 2011.

[5] P. Medvedev, K. Georgiou, G. Myers, and M. Brudno. Computability of models for sequence assembly. In *WABI*, pages 289–301, 2007.

[6] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628, 2008.

[7] P. Peterlongo and R. Chikhi. Mapsembler, targeted assembly of larges genomes on a desktop computer. Research report RR-7565, INRIA, 2011.

[8] P. Peterlongo, N. Schnel, N. Pisanti, M.-F. Sagot, and V. Lacroix. Identifying SNPs without a reference genome by comparing raw reads. *LNCS, SPIRE 2010*, 6393/2010:147–158, 2010.

[9] P. A. Pevzner, H. Tang, and M. S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 98(17):9748–53, 2001.

[10] A. Ratan, Y. Zhang, V. M. Hayes, S. C. Schuster, and W. Miller. Calling SNPs without a reference sequence. *BMC Bioinformatics*, 11:130, 2010.

[11] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson. MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS One*, 3(10):e3373, 2008.

[12] G. Robertson, J. Schein, R. Chiu, R. Corbett, M. Field, S. D. Jackman, K. Mungall, S. Lee, H. M. Okada, J. Q. Qian, M. Griffith, A. Raymond, N. Thiessen, T. Cezard, Y. S. Butterfield, R. Newsome, S. K. Chan, R. She, R. Varhol, B. Kamoh, A.-L. Prabhu, A. Tam, Y. Zhao, R. A. Moore, M. Hirst, M. A. Marra, S. J. M. Jones, P. A. Hoodless, and I. Birol. De novo assembly and analysis of RNA-seq data. *Nature Methods*, 7(11):909–912, 2010.

[13] M. Sammeth. Complete alternative splicing events are bubbles in splicing graphs. *Journal of Computational Biology*, 16(8):1117–1140, 2009.

[14] M. Sammeth, S. Foissac, and R. Guigó. A general definition and nomenclature for alternative splicing events. *PLoS Computational Biology*, 4(8):e1000147, 2008.

[15] M. H. Schulz. *Data Structures and Algorithms for Analysis of Alternative Splicing with RNA-Seq Data.* PhD thesis, Free University of Berlin, 2010.

[16] S. T. Sherry, M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, Jan 2001.

[17] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[18] R. E. Tarjan. Enumeration of the elementary circuits of a directed graph. *SIAM Journal on Computing*, 2(3):211–216, 1973.

[19] J. C. Tiernan. An efficient search algorithm to find the elementary circuits of a graph. *Communonications ACM*, 13:722–726, 1970.

[20] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010.

[21] E. T. Wang, R. Sandberg, S. Luo, I. Khrebtukova, L. Zhang, C. Mayr, S. F. Kingsmore, G. P. Schroth, and C. B. Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221):470–476, Nov 2008.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399