Navigating in a Sea of Repeats in RNA-seq without Drowning

Gustavo Sacomoto[1,2], Blerina Sinaimeri[1,2], Camille Marchet[1,2],
Vincent Miele[2], Marie-France Sagot[1,2], and Vincent Lacroix[1,2]

[1] INRIA Grenoble Rhône-Alpes, France
[2] UMR CNRS 5558 - LBBE, Université Lyon 1, France

Abstract. The main challenge in *de novo* assembly of NGS data is certainly to deal with repeats that are longer than the reads. This is particularly true for RNA-seq data, since coverage information cannot be used to flag repeated sequences, of which transposable elements are one of the main examples. Most transcriptome assemblers are based on de Bruijn graphs and have no clear and explicit model for repeats in RNA-seq data, relying instead on heuristics to deal with them. The results of this work are twofold. First, we introduce a formal model for representing high copy-number repeats in RNA-seq data and exploit its properties to infer a combinatorial characteristic of repeat-associated subgraphs. We show that the problem of identifying in a de Bruijn graph a subgraph with this characteristic is NP-complete. In a second step, we show that in the specific case of a local assembly of alternative splicing (AS) events, using our combinatorial characterization we can *implicitly* avoid such subgraphs. In particular, we designed and implemented an algorithm to efficiently identify AS events that are not included in repeated regions. Finally, we validate our results using synthetic data. We also give an indication of the usefulness of our method on real data.

1 Introduction

Transcriptomes can now be studied through sequencing. However, in the absence of a reference genome, de novo assembly remains a challenging task. The main difficulty certainly comes from the fact that sequencing reads are short, and repeated sequences within transcriptomes could be longer than the reads. This short read / long repeat issue is of course not specific to transcriptome sequencing. It is an old problem that has been around since the first algorithms for genome assembly. In this latter case, the problem is somehow easier because coverage can be used to discriminate contigs that correspond to repeats, *e.g.* using Myer's A-statistics [8] or [9]. In transcriptome assembly, this idea does not apply, since the coverage of a gene does not only reflect its copy-number in the genome, but also and mostly its expression level. Some genes are highly expressed and therefore highly covered, while most genes are poorly expressed and therefore poorly covered.

Initially, it was thought that repeats would not be a major issue in RNA-seq, since they are mostly in introns and intergenic regions. However, the truth

D. Brown and B. Morgenstern (Eds.): WABI 2014, LNBI 8701, pp. 82–96, 2014.

is that many regions which are thought to be intergenic are transcribed [3] and introns are not always already spliced out when mRNA is collected to be sequenced. Repeats, especially transposable elements, are therefore very present in real samples and cause major problems in transcriptome assembly.

Most, if not all current short-read transcriptome assemblers are based on de Bruijn graphs. Among the best known are Oases [14], Trinity [4], and to a lesser degree Trans-Abyss [11] and IDBA-tran [10]. Common to all of them is the lack of a clear and explicit model for repeats in RNA-seq data. Heuristics are thus used to try and cope efficiently with repeats. For instance, in Oases short nodes are thought to correspond to repeats and are therefore not used for assembling genes. They are added in a second step, which hopefully causes genes sharing repeats not to be assembled together. In Trinity, there is no attempt to deal with repeats explicitly. The first module of Trinity, Inchworm, will try and assemble the most covered contig which hopefully corresponds to the most abundant alternative transcript. Then alternative exons are glued to this major transcript to form a splicing graph. The last step is to enumerate all alternative transcripts. If repeats are present, their high coverage may be interpreted as a highly expressed link between two unrelated transcripts. Overall, assembled transcripts may be chimeric or spliced into many sub-transcripts.

In the method we developed, KisSplice, which is a local transcriptome assembler [12], repeats may be less problematic, since the goal is not to assemble full-length transcripts. KisSplice instead aims at finding variations expressed at the transcriptome level (SNPs, indels and alternative splicings). However, as we previously reported in [12], KisSplice is not able to deal with large portions of a de Bruijn graph containing subgraphs associated to highly repeated sequences, *e.g.* transposable elements, the so-called complex BCCs.

Here, we try and achieve two goals: (i) give a clear formalization of the notion of repeats with high copy-number in RNA-seq data, and (ii) based on it, give a practical way to enumerate bubbles that are lost because of such repeats. Recall that we are in a *de novo* context, so we assume that neither a reference genome/transcriptome nor a database of known repeats, *e.g.* RepeatMasker [15], are available.

First, we formally introduce a model for representing high copy-number repeats and exploit its properties to infer a parameter characterizing repeat-associated subgraphs in a de Bruijn graph. We prove its relevance but we also show that the problem of identifying, in a de Bruijn graph, a subgraph corresponding to repeats according to such characterization is NP-complete. Hence, a polynomial time algorithm is unlikely. We then show that in the specific case of a local assembly of alternative splicing (AS) events, by using a strategy based on that parameter, we can *implicitly* avoid such subgraphs. More precisely, it is possible to find the structures (*i.e.* bubbles) corresponding to AS events in a de Bruijn graph that are not contained in a repeat-associated subgraph. Finally, using simulated RNA-seq data, we show that the new algorithm improves by a factor of up to 2 the sensitivity of KisSplice, while also *improving* its precision. For the specific tasks of calling AS events, we further show that our algorithm more sensitive, by a factor of 2, than Trinity, while also being slightly more precise. Finally, we give an indication of the usefulness of our method on real data.

2 Preliminaries

Let Σ be an alphabet of fixed size σ. Here we always assume $\Sigma = \{A, C, T, G\}$. Given a sequence (string) $s \in \Sigma^*$, let $|s|$ denote its length, $s[i]$ the ith element of s, and $s[i, j]$ the substring $s[i]s[i+1] \ldots s[j]$ for any $1 \leq i < j \leq |s|$.

A *k-mer* is a sequence $s \in \Sigma^k$. Given an integer k and a set S of sequences each of length $n \geq k$, we define $span(S, k)$ as the set of all distinct k-mers that appear as a substring in S.

Definition 1. *Given a set of sequences (reads) $R \subseteq \Sigma^*$ and an integer k, we define the directed de Bruijn graph $G_k(R) = (V, A)$ where $V = span(R, k)$ and $A = span(R, k+1)$.*

Given a directed graph $G = (V, A)$ and a vertex $v \in V$, we denote its *out-neighborhood* (resp. *in-neighborhood*) by $N^+(v) = \{u \in V \mid (v, u) \in A\}$ (resp. $N^-(v) = \{u \in V \mid (u, v) \in A\}$), and its out-degree (resp. in-degree) by $d^+(v) = |N^+(v)|$ ($d^-(v) = |N^-(v)|$). A (simple) *path* $\pi = s \rightsquigarrow t$ in G is a sequence of distinct vertices $s = v_0, \ldots, v_l = t$ such that, for each $0 \leq i < l$, (v_i, v_{i+1}) is an arc of G. If the graph is weighted, *i.e.* there is a function $w : A \to Q_{\geq 0}$ associating a weight to every arc in the graph, then the *length* of a path π is the sum of the weights of the traversed arcs, and is denoted by $|\pi|$.

An arc $(u, v) \in A$ is called *compressible* if $d^+(u) = 1$ and $d^-(v) = 1$. The intuition behind this definition comes from the fact that every path passing through u should also pass through v. It should therefore be possible to "compress" or contract this arc without losing any information. Note that the compressed de Bruijn graph [4,14] commonly used by transcriptomic assemblers is obtained from a de Bruijn graph by replacing, for each compressible arc (u, v), the vertices u, v by a new vertex x, where $N^-(x) = N^-(u)$, $N^+(x) = N^+(v)$ and the label is the concatenation of the k-mer of u and the k-mer of v without the overlapping part (see Fig. 1).

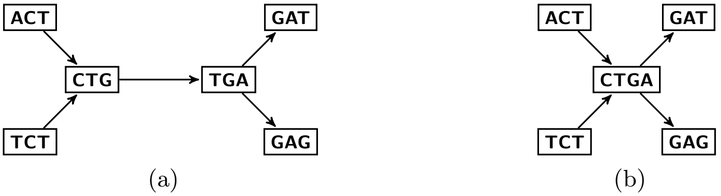

Fig. 1. (a) The arc (CTG, TGA) is the only compressible arc in the given de Bruijn graph ($k = 3$). (b) The corresponding compressed de Bruijn graph.

3 Repeats in de Bruijn Graphs

Given a de Bruijn graph $G_k(R)$ generated by a set of reads R for which we do not have any prior information, our goal is to identify whether there are subgraphs of $G_k(R)$ that correspond each to a set of high copy-number repeats in R. To this end, we identify and then exploit some of the topological properties of the subgraphs that are induced by repeats. Starting with a formal model for representing repeats with high-copy number, we show that the number of compressible arcs, which we denote by γ, is a relevant parameter for such a characterization. This parameter will play an important role in the algorithm of Section 4. However, we also prove that, for an arbitrary de Bruijn graph, identifying a subgraph G' with bounded $\gamma(G')$ is NP-complete.

3.1 Simple Uniform Model for Repeats

We now present the model we adopted for representing high copy-number repeats, *e.g.* transposable elements, in a genome or transcriptome. Basically, our model consists of several "similar" sequences, each generated by uniformly mutating a fixed initial sequence. This model is a simple one and as such should be seen as only a first approximation of what may happen in reality. It is important to point out however that such model is realistic enough in some real cases. In particular, it enables to model well recent invasions of transposable elements which often involve high copy-number and low divergence rate (*i.e.* divergence from their consensus sequence). Consider indeed as an example the recent subfamilies AluYa5 and AluYb8 with 2640 and 1852 copies respectively, which both present a divergence rate below 1% [2] (see [6] for other subfamilies with high copy-number and low divergence).

The model is as follows. First, due to mutations, the sequences $s_1, \ldots, s_m$ that represent the repeats are not identical. However, provided that the number of such mutations is not high (otherwise the concept of repeats would not apply), the repeats are considered "similar" in the sense of having a small pairwise Hamming distance between them. We recall that, given two equal length sequences s and s' in Σ^n, their *Hamming distance*, denoted by $d_H(s, s')$, is the number of positions i for which $s[i] \neq s'[i]$. Indels are thus not consider in this model. Mathematically, it is more convenient to consider substitutions only, but this is not a crucial part of the model.

The model has then the following parameters: Σ, the length n of the repeat, the number m of copies of the repeat, an integer k (for the length of the k-mers considered), and the mutation rate, α, *i.e.* the probability that a mutation happens in a particular position. The sequences $s_1, \ldots, s_m$ are then generated by the following process. We first choose uniformly at random a sequence $s_0 \in \Sigma^n$. At step $i \leq m$, we create a sequence s_i as follows: for each position j, $s_i[j] = s_0[j]$ with probability $1 - \alpha$, whereas with probability α a value different from $s[j]$ is chosen uniformly at random for $s_i[j]$. We repeat the whole process m times and thus create a set $S(m, n, \alpha)$ of m such sequences from s_0 (see Fig. 2 for a small example). The generated sequences thus have an expected Hamming distance of αn from s_0.

$$\begin{array}{cc}
\begin{array}{cccccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} \end{array} & \\
\left(\begin{array}{cccccccccc}
A & A & C & T & G & T & A & T & C & C \\
A & C & C & T & G & T & A & G & C & C \\
G & A & C & T & C & A & A & T & C & C \\
A & A & C & T & C & T & A & T & C & C \\
A & A & C & A & G & T & A & T & C & A \\
A & A & T & T & G & T & A & G & C & C \\
A & G & C & T & G & T & A & T & C & A \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
A & A & G & T & G & A & A & T & C & C
\end{array}\right) &
\begin{array}{c} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ \\ s_{20} \end{array}
\end{array}$$

Fig. 2. An example of a set of repeats $S(20, 10, 0.1)$

3.2 Topological Characterization of the Subgraphs Generated by Repeats

Given a de Bruijn graph $G_k(R)$, if a is a compressible arc labeled by the sequence $s = s_1 \dots s_{k+1}$, then by definition, a is the only outgoing arc of the vertex labeled by the sequence $s[1, k]$ and the only incoming arc of the vertex labeled by the sequence $s[2, k+1]$. Hence the $(k-1)$-mer $s[2, k]$ appears as a substring in R, always preceded by the symbol $s[1]$ and followed by the symbol $s[k+1]$. We refer to such $(k-1)$-mers as being *boundary rigid*. It is not difficult to see that the set of compressible arcs in a de Bruijn graph $G_k(R)$ stands in a one-to-one correspondence with the set of boundary rigid $(k-1)$-mers in R.

We now calculate and compare among them the expected number of compressible arcs in $G = G_k(R)$ when R corresponds to a set of sequences that are generated: (i) uniformly at random, and (ii) according to our model. We show that γ is "small" in the cases where the induced graph corresponds to similar sequences, which provides evidence for the relevance of this parameter.

Claim. Let R be a set of m sequences randomly chosen from Σ^n. Then the expected number of compressible arcs in $G_k(R)$ is $\Theta(mn)$.

Proof. The probability that a sequence of length $k-1$ occurs in a fixed position in a randomly chosen sequence of length n is $(1/4)^{k-1}$. Thus the expected number of appearances of a sequence of length $k-1$ in a set of m randomly chosen sequences of length n is given by $m(n-k+2)(1/4)^{k-1}$. If $m(n-k+2) \leq 4^k$, then this value is upper bounded by 1, and all the sequences of length $k-1$ are boundary rigid (as a sequence appears once). The claim follows by observing that there are $m(n-k+1)$ different k-mers. □

We consider now $\gamma(G_k(R))$ for $R = S(m, n, \alpha)$. We upper bound the expected number of compressible arcs by upper bounding the number of boundary rigid $(k-1)$-mers.

Theorem 1. *Given integers k, n, m with $k < n$ and a real number $0 \leq \alpha \leq 3/4$, the de Bruijn graph $G_k(S(m,n,\alpha))$ has $o(nm)$ expected compressible arcs.*

Proof. Let s_0 be a sequence chosen randomly from Σ^n. Let $S(m,n,\alpha)$ be the set $\{s_1,\ldots,s_m\}$ of m repeats generated according to our model starting from s_0. Consider now the de Bruijn graph $G = G_k(S(m,n,\alpha))$. Recall that the number of compressible arcs in this graph is equal to the number of boundary rigid $(k-1)$-mers in $S(m,n,\alpha)$. Let X be a random variable representing the number of boundary rigid $(k-1)$-mers in G. Consider the repeats in $S(m,n,\alpha)$ in a matrix-like ordering as in Fig.2 and observe that the mutations from one column to another are independent. Due to the symmetry and the linearity of expectation, $E[X]$ is given by $m(n-k-1)$ (the total number of $(k-1)$-mers) multiplied by the probability that a given $(k-1)$-mer is boundary rigid.

The probability that the $(k-1)$-mer $\hat{s} = s[i, i+k-2]$ is boundary rigid clearly depends on the distance from the starting sequence $\hat{s}_0 = s_0[i, i+k-2]$. Let d be the distance $d_H(\hat{s}, \hat{s}_0)$.

Observe that if the $(k-1)$-mer $s[i]\ldots s[k-1]$ is not boundary rigid then there exists a sequence y in $S(m,n,\alpha)$ such that $y[j] = s[j]$ for all $i \leq j \leq i+k-2$ and either $y[i+k-1] \neq s[i+k-1]$ or $y[i-1] \neq s[i-1]$. It is not difficult to see that the probability that this happens is lower bounded by $(2\alpha - 4/3\alpha^2)(1-\alpha)^{k-1-d}(\alpha/3)^d$. Hence we have:

$$Pr[\hat{s} \text{ is boundary rigid} | d_H(\hat{s}, \hat{s}_0) = d] \leq \Big(1-(2\alpha-4/3\alpha^2)(1-\alpha)^{k-1-d}(\alpha/3)^d\Big)^{m-1}$$

By approximating the above expression we therefore have that,

$$\begin{aligned} E[X] &\leq (n-k-1)m \sum_{d=0}^{k-1} Pr[\hat{s} \text{ is boundary rigid} | d_H(\hat{s}, \hat{s}_0) = d] \qquad (1) \\ &\leq (n-k-1)m e^{-(m-1)(2\alpha-4/3\alpha^2)/(\frac{\alpha}{3})^{k-1}} \end{aligned}$$

For a sufficiently large number of copies (*e.g.* $m = \binom{k}{\alpha k}$) and using the fact that $\binom{k}{\alpha k} \geq (1/\alpha)^{\alpha k}$, we have that $E[X]$ is $o(mn)$. This concludes the proof. □

The previous result shows that the number of compressible arcs is a good parameter for characterizing a repeat-associated subgraph.

3.3 Identifying a Repeat-Associated Subgraph

As we showed, a subgraph due to repeated elements has a distinctive feature: it contains few compressible arcs. Based on this, a natural formulation to the repeat identification problem in RNA-seq data is to search for large enough subgraphs that do not contain many compressible arcs. This is formally stated in Problem 1. In order to disregard trivial solutions, it is necessary to require a large enough *connected* subgraph, otherwise any set of disconnected vertices or any small subgraph would be a solution. Unfortunately, we show that this problem is NP-complete, so an efficient algorithm for the repeat identification problem based on this formulation is unlikely.

Problem 1 (Repeat Subgraph).

INSTANCE: A directed graph G and two positive integers m, t.

DECIDE: If there exists a connected subgraph $G' = (V', E')$, with $|V'| \geq m$ and having at most t compressible arcs.

In Theorem 2, we prove that this problem is NP-complete for all directed graphs with (total) degree, *i.e.* sum of in and out-degree, bounded by 3. The reduction is from the Steiner tree problem which requires finding a minimum weight subgraph spanning a given subset of vertices. It remains NP-hard even when all arc weights are 1 or 2 (see [1]). This version of the problem is denoted by STEINER$(1, 2)$. More formally, given a complete undirected graph $G = (V, E)$ with arc weights in $\{1, 2\}$, a set of *terminal* vertices $N \subseteq V$ and an integer B, it is NP-complete to decide if there exists a subgraph of G spanning N with weight at most B, *i.e.* a connected subgraph of G containing all vertices of N.

We specify next a family of directed graphs that we use in the reduction. Given an integer x we define the directed graph $R(x)$ as a cycle on $2x$ vertices numbered in a clockwise order and where the arcs have alternating directions, *i.e.* for any $i \leq x$, (v_{2i}, v_{2i+1}) is an arc. Note that in $R(x)$ all vertices in even positions, *i.e.* all vertices v_{2i}, have out-degree 2 and in-degree 0, while all vertices v_{2i+1}, have out-degree 0 and in-degree 2. Clearly, none of the arcs of $R(x)$ is compressible.

Theorem 2. *The* Repeat Subgraph Problem *is NP-complete even for directed graphs with degree bounded by d, for any $d \geq 3$.*

Proof. Given a complete graph $G = (V, E)$, a set of terminal vertices N and an upper bound B, *i.e.* an instance of STEINER$(1, 2)$, we transform it into an instance of *Repeat Subgraph Problem* for a graph G' with degree bounded by 3. Let us first build the graph $G' = (V', E')$. For each vertex v in $V \setminus N$, add a corresponding subgraph $r(v) = R(|V|)$ in G' and for each vertex v in N, add a corresponding subgraph $r(v) = R(|E| + |V|^2 + 1)$ in G'. For each arc (u, v) in E with weight $w \in \{1, 2\}$, add a simple directed path composed by w compressible arcs connecting $r(u)$ to $r(v)$ in G'; these are the subgraphs corresponding to u and v. The first vertex of the path should be in a sink of $r(u)$ and the last vertex in a source of $r(v)$. By construction, there are at least $|V|$ vertices with in-degree 2 and out-degree 0 (sink) and $|V|$ vertices with out-degree 2 and in-degree 0 (source) in both $r(v)$ and $r(u)$. It is clear that G' has degree bounded by 3. Moreover, the size of G' is polynomial in the size of G and it can be constructed in polynomial time.

In this way, the graph G' has one subgraph for each vertex of G and a path with one or two (depending on the weight of the corresponding arc) compressible arcs for each arc of G. Thus, there exists a subgraph spanning N in G with weight at most B if and only if there exists a subgraph in G' with at least $m = 2|N| + 2|E||N| + 2|V|^2|N|$ vertices and at most $t = |B|$ compressible arcs.

This follows from the fact that any subgraph of G' with at least m vertices necessarily contains all the subgraphs $r(v)$, where $v \in N$, since the number of vertices in all $r(v)$, with $v \in V \setminus N$, is at most $|E| + 2|V|^2$ and the only compressible arcs of G' are in the paths corresponding to the arcs of G. □

We can obtain the same result for the specific case of de Bruijn graphs. The reduction is very similar but uses a different graph family.

Theorem 3. *The* Repeat Subgraph Problem *is NP-complete even for subgraphs of de Bruijn graphs on $|\Sigma| = 4$ symbols.*

4 Bubbles "Drowned" in Repeats

In the previous section, we showed that an efficient algorithm to *directly* identify the subgraphs of a de Bruijn graph corresponding to repeated elements, according to our model (*i.e.* containing few compressible arcs), is unlikely to exist since the problem is NP-complete. However, in this section we show that in the specific case of a local assembly of alternative splicing (AS) events, based on the compressible-arc characterization of Section 3.2, we can *implicitly* avoid such subgraphs. More precisely, it is possible to find the structures (*i.e.* bubbles) corresponding to AS events in a de Bruijn graph that are not contained in a repeat-associated subgraph, thus answering to the main open question of [12].

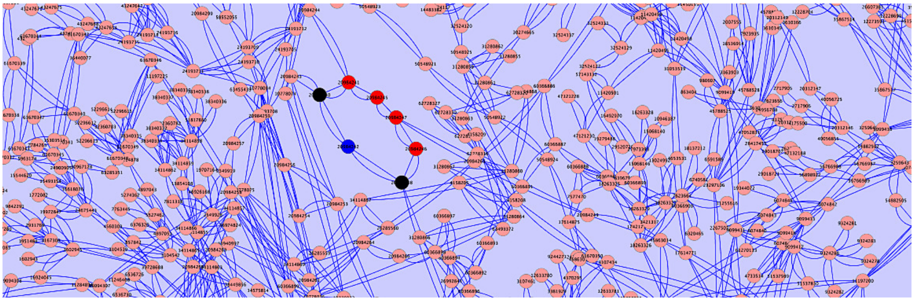

Fig. 3. An alternative splicing event in the SCN5A gene (human) trapped inside a complex region, likely containing repeat-associated subgraphs, in a de Bruijn graph. The alternative isoforms correspond to a pair of paths shown in red and blue.

KisSplice [12] is a method for *de novo* calling of AS events through the enumeration of so-called *bubbles*, that correspond to pairs of vertex-disjoint paths in a de Bruijn graph. The bubble enumeration algorithm proposed in [12] was later improved in [13]. However, even the improved algorithm is not able to enumerate all bubbles corresponding to AS events in a de Bruijn graph. There are certain complex regions in the graph, likely containing repeat-associated subgraphs but also real AS events [12], where both algorithms take a huge amount of time. See

Fig. 3 for an example of a complex region with a bubble corresponding to an AS event. The enumeration is therefore halted after a given timeout. The bubbles *drowned* (or trapped) inside these regions are thus missed by KisSplice.

In Section 3, the repeat-associated subgraphs are characterized by the presence of few compressible arcs. This suggests that in order to avoid repeat-associated subgraphs, we should restrict the search to bubbles containing many compressible arcs. Equivalently, in a compressed de Bruijn graph (see Section 2), we should restrict the search to bubbles with few branching vertices. Indeed, in a compressed de Bruijn graph, given a fixed sequence length, the number of branching vertices in a path is inversely proportional to the number of compressible arcs of the corresponding path in the non-compressed de Bruijn graph. We thus modify the definition of $(s, t, \alpha_1, \alpha_2)$-bubbles in compressed de Bruijn graphs (Def. 1 in [13]) by adding the extra constraint that each path should have at most b branching vertices.

Definition 2 ($(s, t, \alpha_1, \alpha_2, b)$-bubbles). *Given a weighted directed graph $G = (V, E)$ and two vertices $s, t \in V$, an $(s, t, \alpha_1, \alpha_2, b)$-bubble is a pair of vertex-disjoint st-paths π_1, π_2 with lengths bounded by α_1, α_2, each containing at most b branching vertices.*

By restricting the search to bubbles with few branching vertices, we are able to enumerate them in complex regions implicitly avoiding repeat-associated subgraphs. Indeed, in Section 5 we show that by considering bubbles with at most b branching vertices in KisSplice, we increase both its sensitivity and precision. This supports our claim that by focusing on $(s, t, \alpha_1, \alpha_2, b)$-bubbles, we avoid repeat-associated subgraphs and recover at least part of the bubbles trapped in complex regions.

4.1 Enumerating Bubbles Avoiding Repeats

In this section, we modify the algorithm of [13] to enumerate all bubbles with at most b branching vertices in each path. Given a weighted directed graph $G = (V, E)$ and a vertex $s \in V$, let $\mathcal{B}_s(G)$ denote the set of $(s, *, \alpha_1, \alpha_2, b)$-bubbles of G. The algorithm recursively partitions the solution space $\mathcal{B}_s(G)$ at every call until the considered subspace is a singleton (contains only one solution), and in that case it outputs the corresponding solution. In order to avoid unnecessary recursive calls, it maintains the invariant that the current partition contains at least one solution. The algorithm proceeds as follows.

Invariant: At a generic recursive step on vertices u_1, u_2 (initially, $u_1 = u_2 = s$), let $\pi_1 = s \rightsquigarrow u_1, \pi_2 = s \rightsquigarrow u_2$ be the paths discovered so far (initially, π_1, π_2 are empty). Let G' be the current graph (initially, $G' := G$). More precisely, G' is defined as follows: remove from G all the vertices in π_1 and π_2 but u_1 and u_2. Moreover, we also maintain the following invariant $(*)$: there exists at least one pair of paths $\bar{\pi}_1$ and $\bar{\pi}_2$ in G' that extends π_1 and π_2 so that $\pi_1 \cdot \bar{\pi}_1$ and $\pi_2 \cdot \bar{\pi}_2$ belong to $\mathcal{B}_s(G)$.

Base case: When $u_1 = u_2 = u$, output the $(s, u, \alpha_1, \alpha_2, b)$-bubble given by π_1 and π_2.

Recursive rule: Let $\mathcal{B}_s(\pi_1, \pi_2, G')$ denote the set of $(s, *, \alpha_1, \alpha_2, b)$-bubbles to be listed by the current recursive call, *i.e.* the subset of $\mathcal{B}_s(G)$ with prefixes π_1, π_2. It is the union of the following disjoint sets[1].

- The bubbles of $\mathcal{B}_s(\pi_1, \pi_2, G')$ that use e, for each arc $e = (u_1, v)$ outgoing from u_1, that is $\mathcal{B}_s(\pi_1 \cdot e, \pi_2, G' - u_1)$, where $G' - u_1$ is the subgraph of G' after the removal of u_1 and all its incident arcs.
- The bubbles that do not use any arc from u_1, that is $\mathcal{B}_s(\pi_1, \pi_2, G'')$, where G'' is the subgraph of G' after the removal of all arcs outgoing from u_1.

In order to maintain the invariant $(*)$, we only perform the recursive calls when $\mathcal{B}_s(\pi_1 \cdot e, \pi_2, G' - u)$ or $\mathcal{B}_s(\pi_1, \pi_2, G'')$ are non-empty. In both cases, we have to decide if there exist a pair of (internally) vertex-disjoint paths $\bar{\pi}_1 = u_1 \rightsquigarrow t_1$ and $\bar{\pi}_2 = u_2 \rightsquigarrow t_2$, such that $|\bar{\pi}_1| \leq \alpha'_1$, $|\bar{\pi}_2| \leq \alpha'_2$, and $\bar{\pi}_1, \bar{\pi}_2$ have at most b_1, b_2 branching vertices, respectively. Since both the length and the number of branching vertices are monotonic properties, *i.e.* the length and the number of branching vertices of a path prefix is smaller than this number for the full path, we can drop the vertex-disjoint condition. Indeed, let $\bar{\pi}_1$ and $\bar{\pi}_2$ be a pair of paths satisfying all conditions but the vertex-disjointness one. The prefixes $\bar{\pi}_1^* = u_1 \rightsquigarrow t^*$ and $\bar{\pi}_2^* = u_2 \rightsquigarrow t^*$, where t^* is the first intersection of the paths, satisfy all conditions and are internally vertex-disjoint. Moreover, using a dynamic programming algorithm, we can obtain the following result.

Lemma 1. *Given a non-negatively weighted directed graph $G = (V, E)$ and a source $s \in V$, we can compute the shortest paths from s using at most b branching vertices in $O(b|V||E|)$ time.*

As a corollary, we can decide if $\mathcal{B}_s(\pi_1, \pi_2, G)$ is non-empty in $O(b|V||E|)$ time. Now, using an argument similar to [13], *i.e.* leaves of the recursion tree and solutions are in one-to-one correspondence and the height of the recursion tree is bounded by $2n$, we obtain the following theorem.

Theorem 4. *The $(s, *, \alpha_1, \alpha_2, b)$-bubbles can be enumerated in $O(b|V|^3|E||\mathcal{B}_s(G)|)$ time. Moreover, the time elapsed between the output of any two consecutive solutions (i.e. the delay) is $O(b|V|^3|E|)$.*

5 Experimental Results

5.1 Experimental Setup

To evaluate the performance of our method, we simulated RNA-seq data using the FluxSimulator version 1.2.1 [5]. We generated 100 million reads of 75 bp using its the default error model. We used the RefSeq annotated Human transcriptome (hg19 coordinates) as a reference and we performed a two-step pipeline to obtain a mixture of mRNA and pre-mRNA (*i.e.* with introns not

[1] The same holds for u_2 instead of u_1.

yet spliced). To achieve this, we first ran the FluxSimulator with the Refseq annotations. We then modified the annotations to include the introns and re-ran it on this modified version. In this second run, we additionally constrained the expression values of the pre-mRNAs to be correlated to the expression values of their corresponding mRNAs, as simulated in the first run. Finally, we mixed the two sets of reads to obtain a total of 100M reads. We tested two values: 5% and 15% for the proportion of reads from pre-mRNAs. Those values were chosen so as to correspond to realistic ones as observed in a cytoplasmic mRNA extraction (5%) and a total (cytoplasmic + nuclear) mRNA extraction (15%) [16].

On these simulated datasets, we ran KisSplice [12] versions 2.1.0 (KsOld) and 2.2.0 (KsNew, with a maximum number of branching vertices set to 5) and obtained lists of detected bubbles that are putative alternative splicing (AS) events. We also ran the full-length transcriptome assembler Trinity version r2013_08_14 on both datasets, obtaining a list of predicted transcripts, from which we then extracted a list of putative AS events.

In order to assess the precision and the sensitivity of our method, we compared our set of *found* AS events to the set of *true* AS events. Following the definition of Astalavista, an AS event is composed of two sets of transcripts, the inclusion/exclusion isoforms respectively. An AS event is said to be *true* if at least one transcript among the inclusion isoforms and one among the exclusion isoforms is present in the simulated dataset with at least one read. We stress that this definition is very permissive and includes AS events with very low coverage. This means that our ground truth, *i.e.* the set of *true* AS events, contains some events that are very hard, or even impossible, to detect. We chose to proceed in this way as it reflects what happens in real data.

To compare the results of KisSplice with the *true* AS events, we propose that a true AS event is a *true positive* (TP) if there is a bubble such that one path matches the inclusion isoform and the other the exclusion isoform. If there is no such bubble among the results of KisSplice, the event is counted as a *false negative* (FN). If a bubble does not correspond to any *true* AS event, it is counted as a *false positive* (FP). To align the paths of the bubbles to transcript sequences, we used the Blat aligner [7] with 95% identity and a constraint of 95% of each bubble path length to be aligned (to account for the sequencing errors simulated by FluxSimulator). We computed the sensitivity TP/(TP+FN) and precision TP/(TP+FP) for each simulation case and we report their values for various classes of expression of the minor isoform. Expression values are measured in reads per kilobase (RPK).

5.2 KsNew vs KsOld

The plots for the sensitivity of each version on the two simulated datasets are shown in Fig. 4. On the one hand, both versions of KisSplice have similar sensitivity in the 5% pre-mRNA dataset, with KsNew performing slightly better, especially for highly expressed variants. The overall sensitivity in this dataset is 32% and 37% for KsOld and KsNew, respectively. On the other hand, the sensitivity of the new version is considerably better over all expression levels in

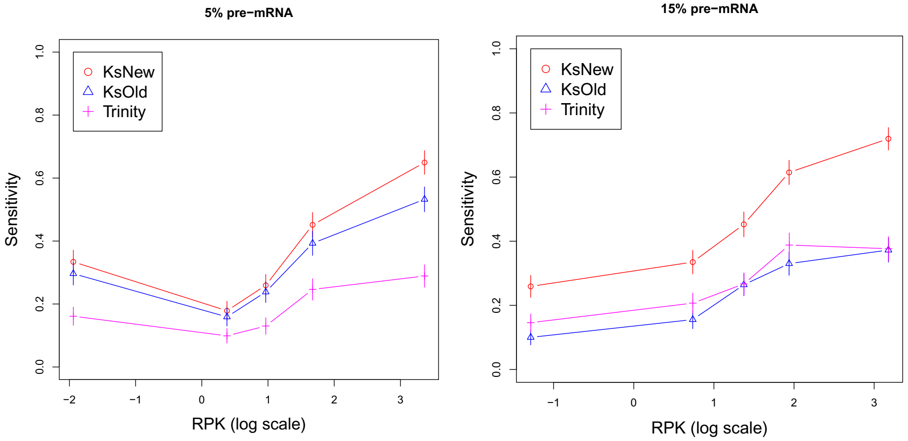

Fig. 4. Sensitivity of KsNew, KsOld and Trinity for several classes of expression of the minor isoform. Each class (*i.e.* point in the graph) contains the same number of AS events (250). It is therefore an average sensitivity on a potentially broad class of expression.

the 15% pre-mRNA dataset. In this case, the sensitivity for KsNew and KsOld are 24% and 48%, respectively. This represents an improvement of 100% over the old version. The results reflect the fact that the most problematic repeats are in intronic regions. A small unspliced mRNA rate leads to few repeat-associated subgraphs, so there are not many AS events drowned in them (which are then missed by KsOld). In this case, the advantage of using KsNew is less obvious, whereas a large proportion of pre-mRNA leads to more AS events drowned in repeat-associated subgraphs which are identified by KsNew and missed by KsOld.

Clearly, any improvement in the sensitivity is meaningless if there is also a significant decrease in precision. This is not the case here. In both datasets, KsNew *improves* the precision of KsOld. It increases from 95% to 98% and from 90% to 99%, in the 5% and 15% datasets, respectively. The high precision we obtain indicates that very few FP bubbles, including the ones generated by repeats, are mistakenly identified as AS events. Moreover, both running times and memory consumption are very similar for the two versions.

5.3 KsNew vs Trinity

The plots for the sensitivity of Trinity on the two simulated datasets are also shown in Fig. 4. In both cases, KsNew performs considerably better than Trinity over all expression levels, with a larger gap for highly expressed variants. The overall sensitivity of Trinity for the 5% and 15% pre-mRNA datasets is 18% and 28%, whereas for KsNew we have 37% and 48%, respectively. Similarly to both KsNew and KsOld, the specificity of Trinity improved from the

5% pre-mRNA to the 15% pre-mRNA dataset. However, this improvement was coupled with a *decrease* of precision from 94% to 75%. This drop in precision is actually mostly due to the prediction of a large number of intron retention, since Trinity assembles both the mRNA and pre-mRNA. KisSplice does not have this problem because most of these apparent intron retentions are bubbles with more than 5 branches (KsNew) or drowned in complex regions of the graph (KsOld). To summarize, KsNew is almost a factor of 2 more sensitive than Trinity, while also being slightly more precise.

As it was already reported in [12], KisSplice (*i.e.* both KsNew and KsOld) is faster and uses considerably less memory than Trinity. For instance, on these datasets, KisSplice uses around 5GB of RAM, while Trinity uses more than 20GB. However, it should be noted that Trinity tries to solve a more general problem than KisSplice, that is reconstructing the full-length transcripts.

5.4 On the Usefulness of KsNew

In order to give an indication of the usefulness of our repeat-avoiding bubble enumeration algorithm with real data, we also ran KsNew and KsOld on the SK-N-SH Human neuroblastoma cell line RNA-seq dataset (wgEncodeEH000169, total RNA). In Fig. 5, we have an example of a *non-annotated* exon skipping event not found by KsOld. Observe that the intronic region contains several transposable elements (many of which are Alu sequences), while the exons contain none. This is a good example of a bubble (exon skipping event) drowned in a complex region of the de Bruijn graph. The bubble (composed by the two alternative paths) itself contains no repeated elements, but it is surrounded by them. In other words, this is a bubble with few branching vertices that is surrounded by repeat-associated subgraphs. Since KsOld is unable to differentiate between repeat-associated subgraphs and the bubble, it spends a prohibitive amount of time in the repeat-associated subgraph and fails to find the bubble.

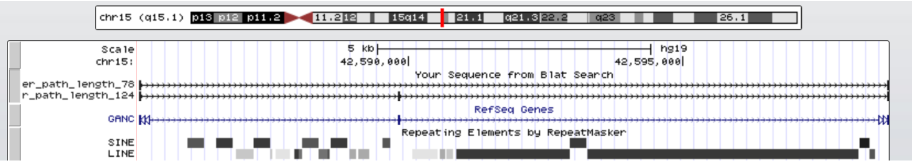

Fig. 5. One of the bubbles found only by KsNew with the corresponding sequences mapped to the reference human genome and visualized using the UCSC Genome Browser. The first two lines correspond to the sequences of, respectively, the shortest (exon exclusion variant) and longest paths (exon inclusion variant) of the bubble mapped to the genome. The blue line is the Refseq annotation. The last line shows the annotated SINE and LINE sequences (transposable elements).

6 Conclusion

Although transcriptome assemblers are now commonly used, their way to handle repeats is not satisfactory, arguably because the presence of repeats in transcriptomes has been underestimated so far. Given that most RNA-seq datasets correspond to total mRNA extractions, many introns are still present in the data and their repeat content cannot be simply ignored. In this paper, we first proposed a simple formal model for representing high copy-number repeats in RNA-seq data. Exploiting the properties of this model we established that the number of compressible arcs is a relevant quantitative characteristic of repeat-associated subgraphs. We proved that the problem of identifying in a de Bruijn graph a subgraph with this characteristic is NP-complete. However, this characteristic drove the design of an algorithm for efficiently identifying AS events that are not included in repeated regions. The new algorithm was implemented in KisSplice (KsNew), and by using simulated RNA-seq data, we showed that it improves by a factor of up to 2 the sensitivity of the previous version of KisSplice, while also improving its precision. In addition, we compared our algorithm with Trinity and showed that for the specific tasks of calling AS events, our algorithm is more sensitive, by a factor of 2, while also being slightly more precise. Finally, we gave an indication of the usefulness of our method on real data.

Clearly our model could be improved, for instance by using a tree-like structure to take into account the evolutionary nature of repeat (sub)families. Indeed, many TE families are composed by different subfamilies that can be divergent from each other. Consider for instance the human ALU family of TEs that contains at least 7 high copy-number subfamilies with intra-family divergence less than 1% and substantially higher inter-family divergence [6]. In this model, the repeats are generated through a branching process on binary trees. Starting from the root to which we associate a sequence s_0, the tree generation process follows recursively the following rule: each node has probability γ to give birth to two children and $1 - \gamma$ to give birth to a single child. In each case the node is associated to a sequence obtained by independently mutating each symbol of the parent sequence with probability α. In this way, the height of the tree reflects the passing of the time. Hence, the maximum height of the tree would correspond to the time passed since the appearance of the first element of this repeat family. The leaves will be associated to the set of repetitions of s_0 in a genome. Beside representing in a more realistic way the generation of copies of transposable elements, this would also allow to model subfamilies of repeats. Indeed, sequences corresponding to leaves of the same subtree are more similar between them then to sequences belonging to leaves outside the subtree.

However, a formal mathematical analysis on this model seems more difficult to obtain. Observe that in the case α is sufficiently small, such model would converge to the one presented in this paper.

Finally, an interesting open problem remains on how to efficiently enumerate AS events for which their variable region (*i.e.* the skipped exon) is itself a high copy number and low divergence repeat.

Acknowledgments. We thank Alexandre Trindade for implementing an early prototype of the model. This work was supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. [247073]10, and the French project ANR-12-BS02-0008 (Colib'read). Part of this work was supported by the ABS4NGS ANR project (ANR-11-BINF-0001-06) and Action n3.6 Plan Cancer 2009-2013.

References

1. Bern, M., Plassmann, P.: The steiner problem with edge lengths 1 and 2. Information Processing Letters (1989)
2. Carroll, M.L., Roy-Engel, A.M., Nguyen, S.V., Salem, A.-H., et al.: Large-scale analysis of the alu ya5 and yb8 subfamilies and their contribution to human genomic diversity. Journal of Molecular Biology 311(1), 17–40 (2001)
3. Djebali, S., Davis, C., Merkel, A., Dobin, A., et al.: Landscape of transcription in human cells. Nature (2012)
4. Grabherr, M., Haas, B., Yassour, M., Levin, J., et al.: Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat. Biot. (2011)
5. Griebel, T., Zacher, B., Ribeca, P., Raineri, E., et al.: Modelling and simulating generic RNA-Seq experiments with the flux simulator. Nucleic Acids Res. (2012)
6. Jurka, J., Bao, W., Kojima, K.: Families of transposable elements, population structure and the origin of species. Biology Direct 6(1), 44 (2011)
7. Kent, W.J.: BLAT–the BLAST-like alignment tool. Genome Res. 12 (2002)
8. Myers, E., Sutton, G., Delcher, A., Dew, I., et al.: A whole-genome assembly of drosophila. Science 287(5461), 2196–2204 (2000)
9. Novák, P., Neumann, P., Macas, J.: Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. BMC Bioinf. (2010)
10. Peng, Y., Leung, H., Yiu, S.-M., Lv, M.-J., et al.: IDBA-tran: a more robust de novo de bruijn graph assembler for transcriptomes with uneven expression levels. Bioinf. 29(13) (2013)
11. Robertson, G., Schein, J., Chiu, R., Corbett, R., et al.: De novo assembly and analysis of RNA-seq data. Nat. Met. 7(11), 909–912 (2010)
12. Sacomoto, G., Kielbassa, J., Chikhi, R., Uricaru, R., et al.: KISSPLICE: denovo calling alternative splicing events from RNA-seq data. BMC Bioinformatics 13(Suppl 6), S5 (2012)
13. Sacomoto, G., Lacroix, V., Sagot, M.-F.: A polynomial delay algorithm for the enumeration of bubbles with length constraints in directed graphs and its application to the detection of alternative splicing in RNA-seq data. In: Darling, A., Stoye, J. (eds.) WABI 2013. LNCS, vol. 8126, pp. 99–111. Springer, Heidelberg (2013)
14. Schulz, M., Zerbino, D., Vingron, M., Birney, E.: Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. Bioinf. (2012)
15. Smit, A.F.A., Hubley, R., Green, P.: RepeatMasker Open-3.0, 1996-2004
16. Tilgner, H., Knowles, D., Johnson, R., Davis, C., et al.: Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs. Genome Res. (2012)