Enumeration problems in RNA-seq data

blerina sinaimeri
The central dogma of molecular biology

The Central Dogma

**DNA**

**Transcription:** the synthesis of an RNA copy of a segment of DNA

**RNA**

**Translation**

**Protein**
From DNA to RNA to proteins in eucaryotes

RNA-splicing in eucaryotes
Given a set of reads $R$ and an integer $k$ we define the de Bruijn graph $B(R,k)$

- Vertices are substrings of length $k$ (k-mers)
- Arcs are $k-1$ suffix-prefix overlaps that appear as a substring in $R$.

Example:

$R = \{\text{ACTGAT}, \text{TCTGAG}\}$, $k=3$

De Bruijn graph:

- Vertices: ACT, GAT
- Edges: ACT $\rightarrow$ GAT, TCT $\rightarrow$ CTG, TGA $\rightarrow$ GAG
Given a set of reads $R$ and an integer $k$ we define the de Bruijn graph $B(R,k)$

- Vertices are substrings of length $k$ (k-mers)
- Arcs are $k-1$ suffix-prefix overlaps that appear as a substring in $R$.

Example:

$R=$\{ACTGAT, TCTGAG\}, $k=3$
Local assembly
Alternative splicing (AS) in RNA
A gene with 2 alternative transcripts

The corresponding de Bruijn graph

A global assembler will search for maximal walks in the graph.
Assembly: Global vs Local

The corresponding De Bruijn graph

4 possible walks corresponding to:

- ABCDE
- ACCE
- ABCCE
- ACDE
Assembly: Global vs Local

4 possible walks corresponding to:

But only 2 alternative transcripts

Every transcript corresponds to a walk but not every walk to a transcript. Global assemblers have to choose the “right” walk.
Local assembly

Main idea: To find an AS event consider only the region of the graph “near” the skipped part (cycle-like pattern)
The problem

Our goal

Identify in RNA-seq data alternative splicing events, without a reference genome. We will only locally assemble them.

Input: A set of reads $R$

Output: The set of AS events
AS Events in de Bruijn graph

AS events will correspond to sequences, $awb$, $ab$. What will these correspond in the de Bruijn graph?

**Example**

$ab=CTGCTT$  $awb=CTGATCTT$

The strings $awb$ and $ab$ will correspond to a bubble, i.e. a pair of internally vertex-disjoint paths, in the de Bruijn graph.
AS Events in de Bruijn graph

Example

\[ ab = \text{CTGCTT} \quad \text{awb} = \text{CTGATCTT} \]

\[ |a| \geq k, \quad |b| \geq k. \]

What characteristics has a bubble generated by an AS event?
AS events in de Bruijn graph

Example

\[ ab = \text{CATCTGCGCA} \quad awb = \text{CATCTGCTCGGCGCA} \]
AS events in de Bruijn graph

**Example**

\[ ab = CATCTGCGCA \quad awb = CATCTGCTCGGCGCA \]

The shortest path has length \(<k-1\) (vertices) as \(w\) and \(b\) share a prefix.
AS Events in de Bruijn graph

Question:

• What is the length of the shorter path for a bubble generated by the pattern \texttt{awb} and \texttt{ab}?
SNPs events in de Bruijn graph

Example

$x=$CATCTACGCAG  $y=$CATCTCCGCAG

Two paths of the same length $k$ (vertices).
Approximate repeats in de Bruijn graph

Example

Inexact repeats may generate bubbles with a similar path length as bubbles generated by AS events. $x=$CATCTTAGGA $y=$CATCTCATCATAGGA CATCTCATCA is an inexact repeat.

This can be easily identified: the longer path contains an inexact repeat. It is sufficient to compare the shorter path with one of the ends of the longer path.
AS Events in de Bruijn graph

Example

• Every AS event generates a bubble.

• Not every bubble with a shorter path with at most k-1 vertices correspond to an AS event.

  • Repeat-associated bubbles: “similar” paths (small edit distance)
Listing all the bubbles

The problem

Given $R$, $k$ list all the bubbles in the de Bruijn graph $B(R,k)$

- The number of bubbles can be exponential in the size of the graph.

- A good algorithm: polynomial delay (polynomial time between two outputs).
Listing \((s,t)\)-paths\)
Listing all (s,t)-paths

The problem

Given a directed graph $G$ list all the (s,t)-paths in $G$.

Idea: Partition the set of solutions

The set of paths $s \rightsquigarrow t$ in $G$ can be partitioned in:

- paths that use $(s, a)$;
- paths that use $(s, b)$;
- paths that use $(s, c)$. 
Listing all \((s, t)\)-paths

**The problem**

Given a directed graph \(G\) list all the \((s, t)\)-paths in \(G\).

**Idea: Recursively partition the set of solutions**

The set of paths \(s \rightsquigarrow t\) in \(G\) can be partitioned in:

- \((s, a)\) plus \(a \rightsquigarrow t\) in \(G - s\);
- \((s, b)\) plus \(b \rightsquigarrow t\) in \(G - s\);
- \((s, c)\) plus \(c \rightsquigarrow t\) in \(G - s\).
Listing all \((s,t)\)-paths

The problem

Given a directed graph \(G\) list all the \((s,t)\)-paths in \(G\).

Idea: **Recursively** partition the set of solutions

The set of paths \(s \leadsto t\) in \(G\) can be partitioned in:

- \((s, a)\) plus \(a \leadsto t\) in \(G - s\);
- \((s, b)\) plus \(b \leadsto t\) in \(G - s\);
- \((s, c)\) plus \(c \leadsto t\) in \(G - s\).
Listing all \((s,t)\)-paths

The problem

Given a directed graph \(G\) list all the \((s,t)\)-paths in \(G\).

Idea: Explore only non-empty partitions

- There is no \(s \leadsto t\) path using \((s, a)\).
- Before exploring a partition, test if it contains at least one solution.
Listing all \((s,t)\)-paths

The algorithm

**Algorithm 1.2:** \texttt{stPaths}(G, s, t, \pi)

- **Input:** An undirected graph \(G\), vertices \(s\) and \(t\), and a path \(\pi\) (initially empty).
- **Output:** The paths from \(s\) to \(t\) in \(G\).

1. \textbf{if} \(s = t\) \textbf{then}
2. \hspace{1em} output \(S\)
3. \hspace{1em} return
4. choose an edge \(e = (s, v)\)
5. \textbf{if} there is a \(vt\)-path in \(G - s\) \textbf{then}
6. \hspace{1em} \texttt{stPaths}(G - s, v, t, \pi(s, v))
7. \textbf{if} there is a \(st\)-path in \(G - e\) \textbf{then}
8. \hspace{1em} \texttt{stPaths}(G - e, s, t, \pi)
Listing all \((s,t)\)-paths

**The algorithm**

**Algorithm 1.2:** \texttt{stPaths}(\(G, s, t, \pi\))

\begin{enumerate}
\item If \(s = t\) then
\begin{enumerate}
\item Output \(S\)
\item Return
\end{enumerate}
\item Choose an edge \(e = (s, v)\)
\item If there is a \(vt\)-path in \(G - s\) then
\begin{enumerate}
\item \texttt{stPaths}(\(G - s, v, t, \pi(s, v)\))
\end{enumerate}
\item If there is a \(st\)-path in \(G - e\) then
\begin{enumerate}
\item \texttt{stPaths}(\(G - e, s, t, \pi\))
\end{enumerate}
\end{enumerate}

\(O(|V| + |E|)\) using DFS
Listing all \((s,t)\)-paths

The algorithm

**Algorithm 1.2: \texttt{stPaths}(G, s, t, \pi)**

- **Input:** An undirected graph \(G\), vertices \(s\) and \(t\), and a path \(\pi\) (initially empty).
- **Output:** The paths from \(s\) to \(t\) in \(G\).

1. if \(s = t\) then
2.     output \(S\)
3.     return
4. choose an edge \(e = (s, v)\)
5. if there is a \(vt\)-path in \(G - s\) then
6.     \texttt{stPaths}(\(G - s, v, t, \pi(s, v)\)) \(\stackrel{O(|V| + |E|)}{\longrightarrow}\) using DFS
7. if there is a \(st\)-path in \(G - e\) then
8.     \texttt{stPaths}(\(G - e, s, t, \pi\))

**Delay:** \(O((|V| + |E|)^2)\).
Listing bubbles
Listing bubbles

**Definition**

\((s,t,a_1,a_2)\)-bubble is a pair of vertex disjoint st-paths with lengths bounded by \(a_1\), \(a_2\).

What if we require a lower bound on the length of the paths?
Listing bubbles

- Two paths \( p_1 = s_1 \leadsto t_1 \) and \( p_2 = s_2 \leadsto t_2 \) are called compatible if \( t_1 = t_2 \) and they respect the upper bounds on the lengths.

- Let \( \mathcal{P}_{\alpha_1, \alpha_2}(s_1, s_2, G) \) be the set of all pairs of compatible paths for \( s_1 \) and \( s_2 \).

\[
\mathcal{P}_{\alpha_1, \alpha_2}(s_1, s_2, G) = \mathcal{P}_{\alpha_1, \alpha_2}(s_1, s_2, G') \cup \bigcup_{v \in \delta^+(s_2)} (s_2, v) \mathcal{P}_{\alpha_1, \alpha'_2}(s_1, v, G - s_2) \\
\]

\( \alpha'_2 = \alpha_2 - w(s_2, v) \)

\( G' = G - \{ (s_2, v) | v \in \delta^+(s_2) \} \)
Listing all \((s,*)\)-bubbles

The algorithm

Algorithm 1: enumerate_bubbles\((s_1, \alpha_1, s_2, \alpha_2, B, G)\)

1. if \(s_1 = s_2\) then
   2. if \(B \neq \emptyset\) then
      3. output(B)
      4. return
   5. else if there is no \((s,t,\alpha_1,\alpha_2)\)-bubble, where \(s = s_1 = s_2\) then
      6. return
   7. end
8. end
9. choose \(u \in \{s_1, s_2\}\), such that \(\delta^+(u) \neq \emptyset\)
10. for \(v \in \delta^+(u)\) do
    11. if there is a pair of compatible paths using \((u,v)\) in \(G\) then
    12. if \(u = s_1\) then
        13. enumerate_bubbles\((v, \alpha_1 - \omega(s_1, v), s_2, \alpha_2, B \cup (s_1, v), G - s_1)\)
    14. else
        15. enumerate_bubbles\((s_1, \alpha_1, v, \alpha_2 - \omega(s_2, v), B \cup (s_2, v), G - s_2)\)
    16. end
    17. end
18. end
19. if there is a pair of compatible paths in \(G - \{(u,v) | v \in \delta^+(u)\}\) then
20. enumerate_bubbles\((v, \alpha_1, s_2, \alpha_2, B, G - \{(u,v) | v \in \delta^+(u)\})\)
21. end
Listing all \((s, *)\)-bubbles

**Lemma 1.** There exists a pair of compatible paths for \(s_1 \neq s_2\) in \(G\) if and only if there exists \(t\) such that \(d(s_1, t) \leq \alpha_1\) and \(d(s_2, t) \leq \alpha_2\).

**Lemma 2.** The test of line 5 can be performed in \(O(n(m + n \log n))\).

**Lemma 3.** The test of line 11, for all \(v \in \delta^+(u)\), can be performed in \(O(m + n \log n)\) total time.

**Theorem 1.** Algorithm 1 has \(O(n(m + n \log n))\) delay.
everything solved?
Listing all the bubbles: Problems

De Bruijn graph: snapshot
Listing all the bubbles: Problems

An alternative splicing event in the SCN5A gene (human) trapped inside a complex region.
Listing all the bubbles: KisSplice

De Bruijn graph: snapshot

An alternative splicing event in the SCN5A gene (human) trapped inside a complex region.

- The complexity comes from highly repeated sequences e.g. TEs in introns of pre-mRNA not yet spliced in RNA-seq data.
Repeat identification

**The problem**

- Can we identify in a de Bruijn graph a subgraph corresponding to repeats?
- What characteristics has the subgraph induced by the repeats?

**Our case**

- no reference genome or repeat database
- no information on the coverage (on RNA-seq this depends also on the expression level of a gene, thus it is not informative)
- high-copy number approximate repeats
Repeats in the de Bruijn graph

The arc \((CTG,TGA)\) can be compressed.

An arc \((u, v)\) is compressible if \(d^+(u) = d^-(v) = 1\).
Repeats in the de Bruijn graph

Idea: Repeats must induce a subgraph of “few” compressible arcs

- The arc \((\text{CTG}, \text{TGA})\) can be compressed.
Is it a good characteristics?

Random sequences

• Choose a set of $m$ sequences of length $n$ randomly from $\{A, C, T, G\}^n$

Repeats

• Let $\alpha$ be the mutation factor, $s_0 \in \{A, C, T, G\}^n$

$$S(m, n, \alpha) =$$

\[
\begin{array}{cccccccc}
A & A & C & T & G & T & A & C & C \\
A & C & C & T & G & T & A & G & C & C \\
A & A & C & T & C & T & A & T & C & C \\
A & A & A & T & G & T & A & T & C & T \\
\end{array}
\]

$s_0$ $s_1$ $s_2$ $s_3$ $s_m$
Is it a good characteristics?

**Random sequences**

- Choose a set of $m$ sequences of length $n$ randomly from $\{A,C,T,G\}^n$

The expected number of compressible edges is $\Theta(mn)$.

**Repeats**

- Let $\alpha$ be the mutation factor, $s_0 \in \{A,C,T,G\}^n$

The expected number of compressible edges is $o(mn)$. 
Identifying the repeat associated subgraph

**Problem (Repeat Subgraph)**

**Instance:** A directed graph $G$ and two positive integers $n$, $t$

** Decide:** If there exists a connected subgraph $G'=(V', E')$ with $|V'| \geq n$ and having at most $t$ compressible edges.

**Theorem**

The Repeat Subgraph Problem is NP-complete even for subgraphs of de Bruijn graphs on an alphabet on 4 symbols.
Identifying the repeat associated subgraph

Problem (STEINER(1,2))

**Instance:** A complete undirected graph $G$, with edge weights in $\{1,2\}$, a set of terminal vertices $N$ and an integer $B$

**Decide:** If there exists a connected subgraph $G'=(V', E')$ with weight at most $B$ containing all terminal vertices in $N$. 

Sketch of the proof
Identifying the repeat associated subgraph

Sketch of the proof

$C_{2|V|}$

no compressible edge
Identifying the repeat associated subgraph

Sketch of the proof

$C_{2|V|}$

$C_{2|V|^2}$

$W_{ab}$

$W_e$
Identifying the repeat associated subgraph

Sketch of the proof

\[ \text{N vertices in } G \rightarrow \text{N} \times 2|V|^2 \text{ vertices in } H \]

subgraph \( G' \) of weight at most \( B \) \( \rightarrow \) subgraph \( H' \) with at most \( B \) compressible edges.
Enumerating bubbles avoiding repeats

For local assembly of AS events we can implicitly avoid repeat-associated subgraphs.

**Main idea**

Avoid paths with “many” branching vertices.

(s,t,a_1,a_2,b)-bubble is a pair of vertex disjoint st-paths with lengths bounded by a_1, a_2 and each one of them containing at most b branching vertices.

- a_1 = 5, a_2 = 6
- b = 3
Enumerating bubbles avoiding repeats

**Algorithm (Main idea)**

(s,t,a₁,a₂,b)-bubble is a pair of vertex disjoint st-paths with lengths bounded by a₁, a₂ and each one of them containing at most b branching vertices.

For every vertex s do
// Generate Bₛ(s, *, a₁, a₂, b)

For every edge e outgoing s do
// bubbles from Bₛ that contain edge e.
\[ B(p₁ e, p₂, G' - u₁) \]
// bubbles from Bₛ that do not contain edge e.
\[ B(p₁, p₂, G' - u₁) \]

Initially
- \( u₁ = u₂ = s \)
- \( p₁ = s \rightarrow u₁ \)
- \( p₂ = s \rightarrow u₂ \)
- \( G' = G \)
Enumerating bubbles avoiding repeats

Algorithm (Main idea)

(s,t,a_1,a_2,b)-bubble is a pair of vertex disjoint st-paths with lengths bounded by a_1, a_2 and each one of them containing at most b branching vertices.

For every vertex s do
// Generate B_s(s, *, a_1, a_2, b)

For every edge e outgoing s do
// bubbles from B_s that contain edge e.
B(p_1 e, p_2, G' - u_1)
// bubbles from B_s that do not contain edge e.
B(p_1, p_2, G' - u_1)

Initially
- u_1 = u_2 = s
- p_1 = s -> u_1
- p_2 = s -> u_2
- G' = G

decide whether these calls are not empty

p'_1 = u_1 -> t and p'_2 = u_2 -> t with | p'_1 | <= a_1; | p'_2 | <= a_2 and at most b branching vertices.
Enumerating bubbles avoiding repeats

Algorithm (Main idea)

(s,t,a₁,a₂,b)-bubble is a pair of vertex disjoint st-paths with lengths bounded by a₁, a₂ and each one of them containing at most b branching vertices.

For every vertex s do
   // Generate Bₛ(s, *, a₁, a₂, b)
   
   For every edge e outgoing s do
      // bubbles from Bₛ that contain edge e.
      B(p₁ e, p₂, G' - u₁)
      // bubbles from Bₛ that do not contain edge e.
      B(p₁, p₂, G' - u₁)

Enumerate bubbles with at most b branching vertices with polynomial delay $O(b |V|^3|E|)$. 
What’s next?
Third Generation Sequencing