

Genomes Comparision via de Bruijn graphs

Student: Ilya Minkin
Advisor: Son Pham

St. Petersburg Academic University

April 28, 2012

Synteny Blocks: Algorithmic challenge

- ▶ Suppose that we are given two genomes
- ▶ The question is: how are they evolutionary related to each other?
- ▶ In order to do rearrangements analysis we must decompose genomes into synteny blocks
- ▶ Synteny blocks are evolutionary conserved segments of the genome
- ▶ These blocks cover most of the genome
- ▶ Occur in both genomes with possible variations

Academic Project

Project: Identify synteny blocks for duplicated genomes represented as sequences of **nucleotides**.

- ▶ **None** of the previous synteny blocks reconstruction software (DRIMM-Synteny (Pham And Pevzner 2010) included) can efficiently solve this problem.
- ▶ DRIMM-Synteny can find the synteny blocks for complicated genomes. But:

Academic Project

Project: Identify synteny blocks for duplicated genomes represented as sequences of **nucleotides**.

- ▶ **None** of the previous synteny blocks reconstruction software (DRIMM-Synteny (Pham And Pevzner 2010) included) can efficiently solve this problem.
- ▶ DRIMM-Synteny can find the synteny blocks for complicated genomes. But:
- ▶ It requires the genome to be represented as sequence of genes.

General Idea: de Bruijn Graph

- ▶ We are given an alphabet Σ and a string S over it, $|\Sigma| = m$
- ▶ A substring T , $|T| = k$ is called *k-mer*
- ▶ De Bruijn graph is a multigraph $G_k = (V, E)$, where
$$V = \Sigma^{k-1} = \{\text{all possible } (k-1)\text{-mers}\}$$
- ▶ If *k-mer* T is presented in S , then we add an oriented edge $(T[1, k-1], T[2, k])$ to the graph
- ▶ Create de Bruijn graph from the nucleotide sequence
- ▶ Conserved regions will yield non-branching paths

Challenges

- ▶ Variations in synteny blocks generate cycles, so we need to simplify the graph
- ▶ Double strandness: conserved regions may occur on both strands. Example:
5' AACCGGTT 3'
3' TTGGCAA 5'
Such blocks are reverse complementary to each other \Rightarrow no non-branching paths
- ▶ Spurious similarity
- ▶ Memory efficiency

Colored graph

- ▶ We use colored de Bruijn graphs [Iqbal et al., 2012] to handle double-strandness
- ▶ Suppose that S^+ and S^- are positive and negative strands of the chromosome
- ▶ Colored de Bruijn graph is a multigraph $G_k = (V, E)$ where $V = \Sigma^{k-1}$
- ▶ For each k -mer T^+ in S^+ add edge $(T^+[1, k-1], T^+[2, k])$ to G_k and mark it *blue*
- ▶ For each k -mer T^- in S^- add edge $(T^-[1, k-1], T^-[2, k])$ to G_k and mark it *red*

Edge labeling

- ▶ Note that our graph is built from a string, not set of reads
- ▶ Each walk in the graph represents a string
- ▶ We are interested only in walks that represent substrings of the source string
- ▶ Assign to each edge e label $L(e) =$ position of the corresponding k -mer on the positive strand
- ▶ Walk $W = (v_1 e_1 v_2 e_2 \dots)$ is considered valid iff:
 1. e_i and e_{i+1} are of the same color
 2. $|L(e_i) - L(e_{i+1})| = 1$

Example

5' ACCTGTCAGT 3'
3' TGGACA GTCA 5'

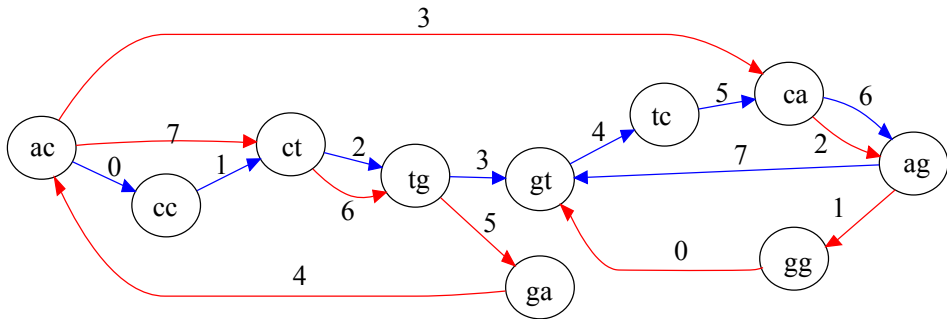


Figure 1: Colored de Bruijn graph built from two strands

Graph simplification

- ▶ Bulges spoil non-branching path
- ▶ A pair of walks W_1, W_2 is a bulge iff:
 - 1) Start vertices of W_1 and W_2 are the same
 - 2) End vertices of W_1 and W_2 are the same
 - 3) W_1 and W_2 have no common edges
 - 4) $|W_1| + |W_2| \leq \delta$
- ▶ Bulges indicate indels and mismatches

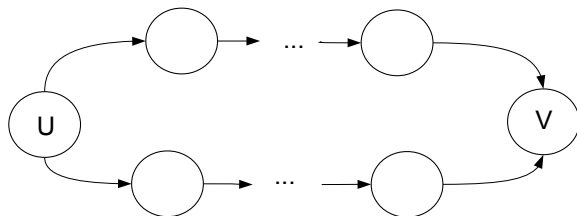


Figure 2: A bulge

Bulge removal algorithm

Remove bulges($G = (V, E)$, $minCycleSize$)

```
1: for all  $v \in V$  do
2:   while  $RemainBulges(v, minCycleSize)$  do
3:      $(W_1, W_2) = FindBulge(v, minCycleSize)$ 
4:     if  $|W_1| < |W_2|$  then
5:        $Collapse(W_1, W_2)$ 
6:     else
7:        $Collapse(W_2, W_1)$ 
```

$Collapse(smallerWalk, biggerWalk)$

```
1:  $W = FindParallelWalks(biggerWalk)$ 
2: for all  $w \in W$  do
3:    $ReplaceSubstring(w, smallerWalk)$ 
```

Bulge removal illustration

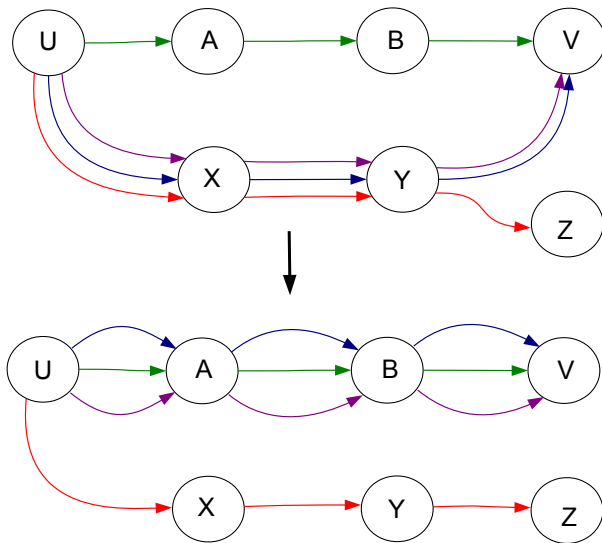


Figure 3: An example of removing bulges from the graph

Non-branching paths

- ▶ How to turn paths into syntenic blocks?
- ▶ Case with *multiplicity* = 2 looks simple, but what about more occurrences?
- ▶ Spurious similarity spoils "good" paths

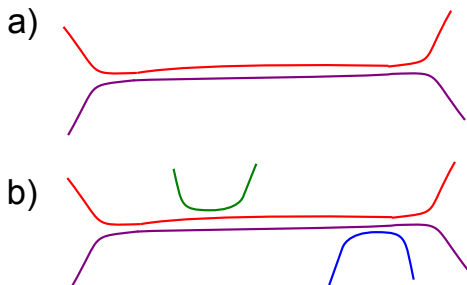


Figure 4:

- a) Syntenic block with multiplicity of 2 is easy to find
- b) Spurious similarity destroys long paths

Current Progress

Now:

- ▶ Framework for working with colored de Bruijn graphs
- ▶ Graph simplification procedure
- ▶ Very draft version of program for finding syntenic blocks
(covers $\leq 10\%$ Arabidopsis genome)

Near future:

- ▶ Develop an efficient approach to cope with spurious similarity
- ▶ Improve performance

References

- ▶ 1. Pevzner P and Tesler G, (2003) Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution.
- ▶ 2. Pham S and Pevzner P, (2010) DRIMM-Synteny: Decomposing Genomes into Evolutionary Conserved Segments
- ▶ 3. Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G, (2012) De novo assembly and genotyping of variants using colored de Bruijn graphs

Thank you!