SyntenyFinder: A Synteny Blocks Generation and Genome Comparison Tool

Ilya Minkin[1], Nikolay Vyahhi[1], and Son Pham[2]

[1] St. Petersburg Academic University, St. Petersburg, Russia
[2] University of California, San Diego, USA

Introduction

Recent advances in sequencing and genome assembling technologies are resulting in many finished genomes. The comparison of these genomes has been emerging as a powerful tool for genome interpretations and has led to many important scientific discoveries. The genome comparison tasks often require genomes to be decomposed to a collection of synteny blocks – long regions of conserved DNA.

[Here we insert a colorful picture about synteny blocks]

Currently existing tools can reconstruct synteny blocks from genomes represented as sequences of enumerated local alignments, or *anchors*. We propose *SyntenyFinder* – a tool for finding synteny blocks in genomes represented as nucleotide sequences. Our approach is based on de Bruijn graph and can be applied to closely related genomes.

De Bruijn graphs

K-dimensional de Bruijn graph $G(k)$ is a graph which vertex set consists of all possible k-length strings over the alphabet $\{A, C, G, T\}$. For each $(k+1)$ - substring w in the given string S we add to $G(k)$ an edge that connects k-prefix of w with k-suffix of w and label the edge with index of the first character of w.

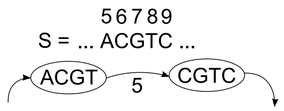


Figure 1: Forming edge set of a **4**-dimensional de Bruijn graph

In this graph we allow only paths that have consecutive labels on edges, so each valid path spells a substring from S. Non-branching paths in the graph $G(k)$ indicate repeats in the string S, i.e. synteny blocks. But in real genomes synteny blocks contain variations that disrupt non-branching paths. These variations create so called *bulges* in the graph. In order to reconstruct synteny blocks we must *simplify* the graph.

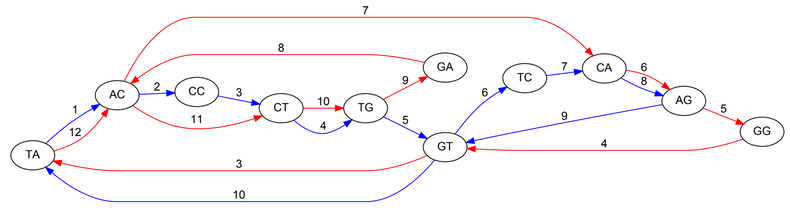
SyntenyFinder algorithm

Given two numbers k and δ and a set $S = \{S_1, S_2, \ldots, S_n\}$ of chromosomes represented as nucleotide strings, our algorithm works as follows:

- Concatenate all string in S into the supergenome $\hat{S}$
- Construct graph $G_+(k)$ from $\hat{S}$ and color all it's edges *blue*
- Construct graph $G_-(k)$ from reverse-complementary of $\hat{S}$ and color all it's edges *red*
- Obtain $G(k) = G_+(k) \cup G_-(k)$ and change $\hat{S}$ so that $G(k)$ doesn't contain bulges with both branches having size $< \delta$
- Output non-branching paths

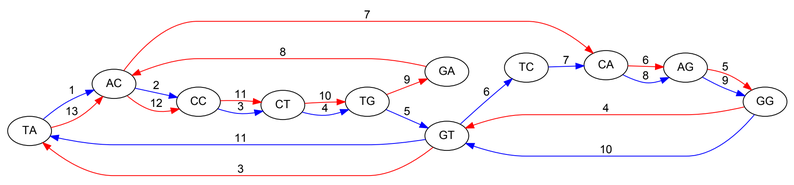
At any moment during simplification there is one-to-one correspondence between the string $\hat{S}$ and the graph – any changes in $\hat{S}$ are immediately reflected in $G(k)$.

5' TACCTGTCAGTA 3'
3' ATGGACAGTCAT 5'



(a) De Bruijn graph $G(k)$ built from $\hat{S} =$ "*TACCTGTCAGTA*". Two paths ("*AC*", "*CC*", "*CT*") and ("*AC*", "*CT*") form a *bulge* that indicate an indel in the substrings "*ACCT*" and "*ACT*".

5' TACCTGTCAGGTA 3'
3' ATGGACAGTCCAT 5'



(b) The same graph after collapsing the bulge. Now long non-branching path spells synteny block "*TACCTG*".

Figure 2: Illustration of *de Bruijn* graphs and simplification process

Results

You can make a poster very quickly and easily by cutting and pasting the LaTeX codes from the paper!

Discussion

You can make a poster very quickly and easily by cutting and pasting the LaTeX codes from the paper!

References

You can make a poster very quickly and easily by cutting and pasting the LaTeX codes from the paper!

Acknowledgments

This work was supported by the Government of the Russian Federation (grant 11.G34.31.0018) and the National Institutes of Health (NIH grant 3P41RR024851-02S1).