Analysis of Mutual's output and some contributions

Israel O. Dilán Pantojas
israelodilan@gmail.com
Computer Science Department
University of Puerto Rico - Río Piedras

Advisors:
Dr. Humberto Ortiz-Zuazaga
humberto.ortiz@upr.edu
Computer Science Department
University of Puerto Rico - Río Piedras

Abstract

As new methods of analysing RNA-Seq data keep arising it is important to corroborate and try to improve them. During this first semester of research we attempted to test and review the algorithm implemented by the Mutual software. This algorithm is aimed at recuperating more information of shared transcripts across species, compared to more traditional methods such as direct comparisons of the organisms assembled transcripts, by implementing a heuristic analysis over the organisms transcriptome represented as a de Bruijn graph. In our analysis we corroborated that the algorithm correctly identifies many sequences and shared transcripts utilizing the de Bruijn graph structure.

1 Introduction

In bio-informatics it is common typically to recover information about an organism utilizing RNA-Seq data recovered by utilizing an RNA Sequencer. This information is represented as reads, which are then assembled into transcripts and eventually a genome. A tool utilized by some of the algorithms that assemble these transcripts is a *de Bruijn graph.* In this process, de Bruijn graphs are used as a convenient way of storing information of the relationships between different reads and eventually transcripts, by studying these graphs we can recover information about how an organism's transcriptome was assembled.

Gathering information about the relationships amongst different organisms usually requires the comparison of the organisms' transcriptomes. Traditionally we approach this task by directly comparing the reported transcripts from organism A to organism B's. The algorithm presented by Fu *et al.* [1] found in BMC Genomics, contains a different approach to performing this comparison. By iteratively comparing each node in an organisms transcripts represented as a de Bruijn graph, it is possible to recover more information about the transcripts shared between both organisms.

By utilizing data of a referenced organism, who's genome is know and has been mapped, we studied the correctness of the output reported by Mutual, which is software that implement's the algorithm described in the paper[1] and it is available at: `http://faculty.cse.tamu.edu/shsze/mutual/`. By comparing shared transcripts between two samples of different organisms from the same species., we expected to recover very similar information about shared transcripts in both organisms.

Nemastotella Venectis Embryos RNA-Seq data was utilized during the comparisons. The Nemastotella data utilized was obtained from:

- Nemastotella Embryonic Transcriptome
    - `https://darchive.mblwhoilibrary.org/handle/1912/5613`

2 Methodology

First after the initial setup of the necessary programs to analyze the data, which are: Velvet, Oases and Mutual, the raw data was ran on them. After observing the preliminary results, it was evident that the data had not been quality controlled. Therefore the data had to be quality controlled by utilizing FastQC to analyze the quality of the RNA-Seq Data & also both Scythe and Sickle to remove adapters and trim edges, thus performing the necessary clean up and control of the data. Afterwards the Quality Controlled data was ran with Velvet and Oases, while utilizing the sample arguments given in Mutual's source code home page.

The various methods of comparison that were implemented in the analysis are listed below:

- Comparison between two different organisms of the same species utilizing Mutual, then validating the output with blast using NCBI's database of referenced organisms.

- Comparison between a sample of the same organism against itself, then validating the output while testing for extreme similarity amongst both of the outputs.

- Comparison of transcripts pertaining to the same organisms recovered by Velvet/Oases against Mutual's output.

3 Results

After several tests, it seems that Mutual does in fact work as it claims to do. When we analysed the results of comparison, we encountered a high number of similar transcripts as expected, when an amount of arbitrarily selected transcripts was blasted against NCBI's database, they were correctly identified as genomic data highly similar to Nemastotella Venectis referenced genome data. When utilizing the same organism for control, most of the results were shared amongst the outputs.

The we created a LastGraph file parser, written in Python and utilizing NetworkX, to aid in the examination of connected components within the de Bruijn Graph. It parses the file into several GFA files, containing the connected components found in the de Bruijn graph, there is one file for each connected component in the graph. Thus breaking down the volume of data into some more small workable bits. Files for this parser can be found in the following Megaprobe-Lab repository: `https://github.com/humberto-ortiz/megaprobe-lab/tree/master/content/Mutual_Files/Works`

4 Future Work

- Mutual:
  - It's necessary to create better documentation for the Mutual program.
  - It's also necessary to provide information about it's licensing.
- Parser:
  - The LastGraph parser still needs a way to efficiently manage recovering the full sequence of a node from the forward sequence and it's reverse complement.
- Research:
  - Utilize more quantitative methodology to further test and validate results.
  - Continue improving files contributed to Megaprobe's repository.
  - Compare two organisms from different species.

5 Discussion

While working on this research it was apparent that there is a need to improve on the methods and formats of representing RNA-Seq data. For example, the way the LastGraph file format handles a sequence's forward and reverse complement shifting each by a length K, might in some cases duplicate data unnecessarily, therefore increasing the file's size without need. As it is done with some other formats like the GFA format, it might be more efficient to store only the full forward sequence and then simply calculate the reverse complement when necessary.

References

[1] Fu S, Tarone AM, Sze SH. (2015). Heuristic pairwise alignment of de Bruijn graphs to facilitate simultaneous transcript discovery in related organisms from RNA-Seq data. BMC Genomics 16:S5. http://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-16-S11-S5

[2] Ortiz-Zuazaga H. G.(Mon 01 December 2014). Sequence assembly with read errors. http://ccom.uprrp.edu/~humberto/kmer-graph-with-errors.html

[3] Bolger, A. M., Lohse, M., & Usadel, B (2014). Trimmomatic: A flexible read trimming tool for Illumina NGS data. Bioinformatics, btu170. http://www.usadellab.org/cms/?page=trimmomatic

[4] Melsted P. (2015). The GFA Format Specification. https://github.com/pmelsted/GFA-spec/blob/master/GFA-spec.md