Abstract -  is a short description of the field of investigation, methods used, and its conclusions. ///// what you did and what you found out.

The advent of next generation sequencing (NGS) has allowed many researchers a flux of …

In // section

By using a combination of overlap-based construction and suffix/tree using the TRIE data structure.

Decoding DNA symbols using next-generation sequencers was a major breakthrough in genomic research. Despite the many advantages of next-generation sequencers, e.g., the high-throughput sequencing rate and relatively low cost of sequencing, the assembly of the reads produced by these sequencers still remains a major challenge

Background - would say why this field was chosen, what is already known about it, and why further investigation would be useful. // why you did it. Motivation:: // Sequence Assembly // Reference Geneome is unavailable // Importance

Sequence assembly using short reads is important because current sequence technologies are unable to sequences a whole genome [3].

Sequence assembly is important because

1.

2.

Several approaches to sequence assembly shotgun approach.

1. K-Spectrum-Based Construction

2. Overlap-Based Construction

3. Greedy-Based Construction

The Greedy-Based algorithm seeks after the greatest immediate sequences to the construction of the

There’s no reference genome.

The *De novo* sequence assembly is the process where individual sequences from the original DNA sequence are combined to form a larger sequence based on the information available of similar nucleotides to reconstruct the original sequence each individual sequences were derived from [1].

There are many challenges for short sequence assembly applications.

Short reads.

Errors due to less unique end-pair reads

Repeated sequences larger than the read lengths

Problem

The assembler accepts a data file that houses DNA sequences that have been split into multiple pieces of smaller sequences. Once the reads have been read, it will then attempt to reconstruct the smaller sequences and assemble the genome it once represented.

Development Environment.

For the test, an Intel Core i7-3537U CPU @ 2.0 GHZ with 4GB RAM laptop computer was used.

Application Parameters

SALSA operates with three specific parameters that determine its results: A sample data file, the read length, and number of reads generated. The sample data file can be generated using an optional parameter or be set by the user manually in the resource file. The read length is a pivotal value that determines the length of each read sequence split from the aforementioned sample data file. Finally, the number of reads generated when increased will increase the chance of an overlap (coverage).

Input:

Reads taken from genome

Output:

Assemble Genome

Baseline Method:

In order to compare the performance of SALSA, a baseline algorithm was created

The baseline algorithm begins its preparation stage similar to

Surprisingly the Greedy method

Application Parameters

Sample Data File

Frag Length

Reads

Baseline Method

Preparation

Get sample data

Chop it up

Put it all into a set

Method:

Choose a

My Method

SALSA:

In the Graph-based SALSA approach

TRIE datastructure

Preparation

Read in sample data

Construct Graph | Edge

Evaluate Weight

Reconstruction of sample data

Measure of Performance

How similar

Error in assembled genome

Speed of computation

Memory usage

Compare above vs baseline

Performances

Future works

Statistics for repeats

Conclusion

Real Data construction much more difficult than simulated data? Why? Real Data has a lot of repeated sequences making the construction really difficult. Not a surprise because of Article in[1][2]

References

[1] Paszkiewicz K., Studholme D. *De novo assembly of short sequence reads.* Briefing in Bioinformatics vII;5: 2010

[2] Kingsford C., Schatz MC., Pop M. *Assembly complexity of prokaryotic genomes using short reads.* BMC Bioinformatics 2010; 11:21

[3] El-Metwally S., Hamza T., Zakaria M, Helmy M., *Next-Generation Sequence Assembly: Four Stages of Data Processing and Computational Challenges.* PLOS Computational Biology. Vol 9;12 Dec 2013