Hadoop Optimized De Novo Assembler

Student: Aksenov V.
Advisor: Maxim Mikheev / BIODATOMICS

Problem

- Most of de novo assemblers require huge amount of RAM;
- There are several MPI based implementations;
- There is only one implementation of de novo assembly on Hadoop. This implementation analyses reads in groups independently. It provides lower quality than assemblers running on full dataset.

de novo Assemblers

- Steps in assembly:
  - Building de Bruijn graph;
  - Compressing;
  - Error correction;
  - Scaffolding.
- All implementation have their own implementation of Graph database.

Goal

- We need to make de novo assembler which will be natively run on the Hadoop;
- Use existing Graph Database which is working on top of Hadoop.

What is done?

- Reading FastQ Formatted file;
- Building de Brujin graph:
  - Standard graph;
  - List of edges with ids for GraphX DB;
- Compressing.

Reading FastQ Formatted file

- Download biodt-fs repository;
- Building with maven;
- Reading fastq file using Hadoop.

Building standard de Brujin graph

- Read file map read to one edge with mark;
- Vertices marked by sequence.

Building list of edges

- List and enumerate all the vertices;
- Run standard graph builder;
- Twice make join of 2 tables - enumerated vertices and graph.

Compressing

- Read edges from standard graphs;
- Map each edge to in and out edge;
- If vertex contains one in and one out edge, then we could add new edge to output, else add all out edges from that vertex;
- Repeat this log(max path length).

And it works

- The program was runned on some small synthetic tests;
- Only check for correctness of the algorithm;
- Repository is private :-(.

Questions?

Thanks for your attention