MSPGraphBuilder: An efficient de Bruijn graph constructor

January 17, 2012

Version 0.1

Abstract

MSPGraphBuilder is a disk-based software to build de Bruijn graph from DNA sequences.

1. Synopsis

```
java -jar PartitionExtension.jar -in InputPath -k kmerLength -L readLength [-NB NumberOfBlocks] [-p MinimumSubstringLength] [-t threads] [-b bufferSize]

java -jar HashExtension.jar -k kmerLength -NB NumberOfBlocks [-t threads] [-b bufferSize] [-c capacity] [-cut coverageCut]

java -jar Link64.jar -edgeDir EdgePath -k kmerLength -NB NumberOfBlocks [-b bufferSize] [-c capacity]
```

2. Description

MSPGraphBuilder is a de Bruijn graph constructor based on the minimum substring partitioning technique. It will first partition the k-mers in DNA sequences into several disjoint partitions and compress consecutive k-mers to reduce I/O cost. Then it will build unipaths in each partition individually. Finally it will link these unipaths in all partitions to generate the global de Bruijn graph. MSPGraphBuilder can significantly reduce the memory consumption since it compresses k-mers on the unipaths in each partition first, before loading all the partitions into memory. In contrast, the direct approach cannot do this compression on the fly before loading all the k-mers into the memory.

To build de Bruijn graph with MSPGraphBuilder, use the commands like:

```
java -jar PartitionExtension.jar -in input.fasta -k 55 -L 101 -NB 256 -p 6 -t 8
java -jar HashExtension.jar -k 55 -NB 256 -t 8 -cut 1
java -jar Link64.jar -edgeDir /home/Edge -k 55 -NB 256
```

These three commands will build the de Bruijn graph with the 55-mes in input.fasta using 8 threads. Specifically speaking, the first command will partition the short reads data input.fasta (whose average read length is 101) into 256 partitions using minimum substring partitioning, with the minimum substring length being 6; and the second command will hash the 55-mers and build unipaths (after removing all single coverage 55-mers) in these 256 partitions with 8 threads; and the last command will link the unipaths in all 256 partitions to generate the desired global de Bruijn graph.

3. Options

3.1 PartitionExtension

Function: Partition short reads data (in fasta format) using minimum substring partitioning

Usage: `java -jar PartitionExtension.jar [options]`

Options Available:

```
[-help]: Print Help Information and Exit
[-in InputPath]: (String) Input Short Reads Data Path (Mandatory)
[-k k]: (Integer) K-mer Length, should be odd number smaller than 64 (Mandatory)
[-L readLength]: (Integer) Average Short Read Length (Mandatory)
[-NB numOfBlocks] :  (Integer) Number Of K-mer Blocks/Partitions. Default: 256
[-p pivotLength] :  (Integer) Minimum Substring Length. Default: 6
[-t numOfThreads] :  (Integer) Number Of Threads. Default: 8
[-b bufferSize] :  (Integer) Read/Writer Buffer Size. Default: 8192
```

3.2 HashExtension

Function: Hash the k-mers and build unipaths in each minimum substring partitions

Usage: `java -jar HashExtension.jar [options]`

Options Available:

```
[-help]: Print Help Information and Exit
[-k k]: (Integer) K-mer Length, should be odd number smaller than 64 (Mandatory)
[-NB numOfBlocks] :  (Integer) Number Of K-mer Blocks/Partitions. (Mandatory)
[-t numOfThreads] :  (Integer) Number Of Threads. Default: 8
[-b bufferSize] :  (Integer) Read/Writer Buffer Size. Default: 8192
[-c capacity] : (Integer) Hash Table Initial Capacity. Default: 1000000
[-cut covCut] : (Integer) Coverage Cut Threshold. Default: 5
```

Note: The settings of ***k*** and ***NB*** should be in consistent with those in PartitionExtension.

3.3 Link

Function: Link the unipaths in all minimum substring partitions to generate the global de Bruijn graph.

Usage: `java -jar Link64.jar [options]`

Options Available:

```
[-help]: Print Help Information and Exit
[-edgeDir edgePath] : (String) Path of Unipaths from HashExtension (Mandatory)
[-k k]: (Integer) K-mer Length, should be odd number smaller than 64 (Mandatory)
[-NB numOfBlocks] :  (Integer) Number Of K-mer Blocks/Partitions. (Mandatory)
[-b bufferSize] :  (Integer) Read/Writer Buffer Size. Default: 8192
[-c capacity] : (Integer) Hash Table Initial Capacity. Default: 1000000
```

Note: The settings of ***k*** and ***NB*** should be in consistent with those in PartitionExtension.

4. Version

Version: 0.1 of January 17, 2012

5. Bug Reports

For bugs or questions or comments, please write to *yangli* at *cs* dot *ucsb* dot *edu*

6. Copyright