Documentation for BFGraph

A memory efficient De Brujin graph assembler using Bloom Filters

Pall Melsted
Trausti Saemundsson

Questions and bug reports: pmelsted@gmail.com

Contents

Cover photo is from Wikipedia[6]

Chapter 1

Introduction

This is a documentation for the program BFGraph. BFGraph is a De Brujin graph assembler. Currently it makes the pregraph but does not simplify it like SOAPdenovo[4] and Velvet[5]. It makes the pregraph by using Bloom Filters in contrast to most other assemblers, that use hash tables. This saves a lot of memory and the result is independent of the false positive rate of the Bloom Filter.

A Bloom Filter is quite similar to a hash table but. If a Bloom Filter is queried for a specific key, it answer correctly if the key is not stored but answers incorrectly according to the false positive rate if the key is stored within it.
The first phase of BFGraph makes the pregraph according to a Bloom Filter but then fixes the graph afterwards which makes it independent of the probabilistic nature of the Bloom Filter. The memory usage of BFGraph is a lot lower than that of most other assemblers, and it is compared in the paper about this program.

Chapter 2

Definitions

kmer[7]: string of A,C,G,T which has length k (k is often 31)
twin: interchange A <-> T and C <-> G in a string and then reverse it
contig: string of A,C,G,T which has length greater or equal to the kmer size

Chapter 3

Dependencies

Inside the program directory are a few programs made by others:

- **sparse_hash**[2] is inside the directory google, a memory efficient hash table made by Google** used in this program for storing common kmers.
- **libdivide**[3] in the file *libdivide.h* is used for fast integer division.
- **kseq**[1] in the file *kseq.h* is used for fasta/fastq file reading.

Chapter 4

Usage

Chapter 5

Structure of the program

5.1 Unused files

Every file in BFGraph's base directory is used by the program except the following files:

- *CountBF.hpp*
- *CountBF.cpp*
- *DumpBF.hpp*
- *DumpBF.cpp*
- *KmerIntPair.hpp*
- *KmerIntPair.hpp*
- *bloom_filter.hpp*
- *BlockedBloomFilter.hpp*

5.2 Used files

The following classes reside in files by a similar name except ContigRef, it is located in *KmerMapper.hpp* and FastqFile is located in *fastq.hpp*

(a) Classes

- BloomFilter
- CompressedCoverage
- CompressedSequence
- Contig
- ContigRef
- FastqFile
- Kmer

- KmerIterator
- KmerMapper

(b) Structs

- *FilterReads_ProgramOptions* is used as the name indicates to store parameter values from the user for the subcommand **filter**, located in *FilterReads.cpp*
- *BuildContigs_ProgramOptions* is used to store parameter values from the user for the subcommand **contigs**, located in *BuildContigs.cpp*
- *CheckContig*
- *FindContig*
- *MakeContig*
- *NewContig*
- *KmerHash*

Chapter 6

Important methods in the program

Chapter 7

Final words

Thanks.

**Trademark, service mark, or registered trademark of Google Inc.

Bibliography

[1] attractivechaos. *Generic stream buffer plus a FASTA_Q parser*. [Online; accessed 10-August-2012]. 2012. URL: https://github.com/attractivechaos/klib/blob/master/kseq.h.

[2] Google. *sparsehash - An extremely memory-efficient hash_map implementation*. [Online; accessed 10-August-2012]. 2012. URL: http://code.google.com/p/sparsehash/.

[3] ridiculous_fish. *libdivide is an open source library for optimizing integer division*. [Online; accessed 10-August-2012]. 2012. URL: http://libdivide.com/.

[4] SEQanswers - SEQwiki. *SOAPdenovo - SEQwiki*. [Online; accessed 10-August-2012]. 2012. URL: http://seqanswers.com/wiki/SOAPdenovo.

[5] SEQanswers - SEQwiki. *Velvet - SEQwiki*. [Online; accessed 10-August-2012]. 2012. URL: http://seqanswers.com/wiki/Velvet.

[6] Michael Ströck. *An overview of the structure of DNA*. [Online; accessed 9-August-2012]. 2006. URL: http://commons.wikimedia.org/wiki/File:DNA_Overview2.png.

[7] Wikipedia. *k-mer — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-August-2012]. 2012. URL: http://en.wikipedia.org/wiki/K-mer.