MindTheGap — User Manual

G. Rizk
guillaume.rizk@inria.fr

April 4, 2014

Abstract

MindTheGap is a software that performs detection of DNA insertions (aided by a reference sequence) and *de novo* assembly of detected insertions. It takes as input a set of Illumina reads and a reference sequence. It outputs two sets of FASTA sequences: one is the set of breakpoints of detected insertion sites, the other is the set of assembled insertions for each breakpoint. For each breakpoint, MindTheGap either returns a single insertion sequence (when there is no assembly ambiguity), or a set of candidate insertion sequences (due to ambiguities) or nothing at all (when the insertion is too complex to be assembled). MindTheGap performs *de novo* assembly using Minia. Hence, the computational resources required to run MindTheGap are significantly lower than that of other assemblers.

Contents

1 Software manual

1.1 Installation

To install MindTheGap, just type `make` in the MindTheGap folder. MindTheGap has been tested on Linux 32 or 64 bits and Mac OS X 10.8.
By default maximum kmer size is 32. To allow kmer size up to size, recompile with for example `make k=size`
To run MindTheGap, type `./mindthegap`.

1.2 MindTheGap modules

MindTheGap has three distinct operating modules: index, find and fill. These modules correspond to respectively

- indexing the reads,
- finding insertion sites (called *breakpoints*), and
- assembling insertion sequences for each breakpoint.

A technical explanation of each module follows in this manual.

The index module needs to be executed first, as the two other modules rely on the index. Then, unless the user already possess a set of breakpoint location, the find module is executed on a reference sequence: it produces a breakpoint file. Finally, the fill module is executed to perform assembly.

1.3 Parameters

The index module requires several parameters, which will be saved to a project configuration file (`<project_name>.config`) . The find and fill modules reuse parameters from the project configuration file. `[..]` denotes an optional parameter.

The parameters for the index module are:

1. `reads` – the input reads file, can be in FASTA or FASTQ format, gzipped or not. To use a list of read files, pass to this parameter a file containing the list of read files, one per line.
2. `[-k kmer_size]` – k-mer length ($1 \leq k \leq 64$ on a default configuration. It can be changed by e.g. typing `make k=128`)
3. `[-t min_abundance]` – filters out k-mers seen less than the specified number of times
4. `[-d max_disk_space]` – in MB, for index phase, default: min(available disk space / 2, reads file size)
5. `<-g estimated_genome_size>` – estimation of the size of the genome present in the reads, in base pairs. Doesn't need to be accurate
6. `<-p project_name>` – any string naming the current project, will serve as a prefix for all intermediate and results files.

The parameters for the find module are:

1. `reference.fa` – a reference genome in the FASTA format
2. `<-p project_name>` – same project name as in the index module
3. `[-mrep maximal_repetition_size]` – default 5; maximal repetition size for *fuzzy* sites (in nt)
4. `[-hom]` – only search for homozygous insertions.

The parameters for the fill module are:

1. `breakpoints.fa` – a list of breakpoints (e.g. produced by the find module) in the FASTA format
2. `<-p project_name>` – same project name as in the index module
3. `[-i max_insertions_size]` – maximal length of insertions that will be assembled (in nt)
4. `[-n max nodes]` – default 100; maximum number of nodes in contig graph
5. `[-m nb mismatches]` – default 0; maximum number of mismatches in right anchor
6. `[-h nb_repet]` – default 1; maximum number of time flanking kmers of heterozygous insertion site are allowed to be repeated in reference genome.
7. `[-r reverse]` – default 2; Graph traversal search 0 : forward only, 1 : reverse only, 2: both

1.4 Explanation of parameters

`kmer_size` The k-mer length strongly depends on the input dataset. A typical (and relatively arbitrary) value to try for short Illumina reads of read length above 50 is 27. For longer Illumina reads ($\approx$ 100 bp), do not hesitate to try larger values of k, up to $60 - 70$.

`min_abundance` The `min_abundance` parameter also strongly depends on the dataset. It corresponds to the smallest amount of times a correct k-mer is expected to be seen in the reads. A typical value is 3. Setting it to 1 is not recommended, as no erroneous k-mer would be discarded, which would result in a very large memory usage.

`estimated_genome_size` The estimated genome size parameter helps limiting the memory usage during the index phase of MindTheGap. It only needs to be accurate within an order of magnitude.

`project_name` The `project_name` parameter is any arbitrary file name prefix, for example, `test_project`. This prefix can include a path to choose the project location (useful if temp files need to be stored on a specific disk), for example `/data/test_project`

`mrep` The `mrep` parameter is the maximal repetition size allowed at a repetition site. In the MindTheGap paper, such sites are called *fuzzy* sites : the prefix of the insertion is the same as the prefix of the right flanking kmer. Higher value will lead to higher false positive rate.

`h` Only used for heterozygous insertion site detection. Higher valuer will lead to higher false positive rate, and higher recall. -h 1 is the default, and seems to be the best trade-off.

1.5 Output

The output of the index module of MindTheGap is a de Bruijn graph in an ad-hoc format, that will be used in the find and fill modules.

The output of the find module of MindTheGap is a set of breakpoints in the FASTA format, in the file `<project_name>.breakpoints.fa`.

The output of the fill module of MindTheGap is a set of insertions in the FASTA format, in the file `<project_name>.insertions.fa`.

2 Usage example

With the two files reads.fasta and ref.fasta present in the demo folder :

```
./mindthegap index reads.fasta -g 10000 -p project -k 27 -t 4
./mindthegap find ref.fasta -p project
./mindthegap fill project.breakpoints -p project
```