User guide

User Guide – version 2.4.0

GPU-BSM

GPU-BSM (standing for GPU-BiSulfite reads Mapping) is a GPU-based tool devised to map bisulfite-treated reads. It has been designed to support directional and non-directional libraries generated from whole-genome bisulfite sequencing (WGBS) and RRBS sequencing.
Basically, GPU-BSM adopts an unbiased strategy that reduces the complexity of involved sequences converting cytosines to thymines. Then, sequences represented with a simplified 3-letter nucleotide alphabet are aligned using the SOAP3-dp short-read mapping tool.

Mapping strategy

GPU-BSM creates two sequences from the original forward genomic strand. The first sequence is obtained by converting cytosines to thymines, whereas the second sequence is obtained by converting guanines to adenines.
As for RRBS libraries, these sequences are generated analyzing a modified reference genome that take into account only those genomic fragments compatible with the sequencing experiment.
Directional and non-directional libraries are treated differently.

To map directional reads, GPU-BSM performs two different mappings using SOAP3-dp. The first mapping is obtained by converting cytosines to thymines in the reads and then aligning them to the first sequence; the second is obtained by converting guanines to adenines in the reverse complement of the reads and then aligning them to the second sequence.

To map non-directional reads, GPU-BSM performs four different mappings. In addition to the mappings performed for a directional library, GPU-BSM uses SOAP3-dp to map the reverse complement of the reads with cytosines converted to thymines to the first sequence, and the reverse complement of the reads with guanines converted to adenines to the second sequence.

Then, GPU-BSM analyzes the mapped reads, detecting and removing ambiguous reads and those that are in fact false positives. We consider ambiguous those reads for which *i)* exist a best match for at least two of two/four alignments performed according to the exploited library or *ii)* exist at least two best hits for a single alignment.
GPU-BSM calculates the number of mismatches of the mapped reads using the 4-letter nucleotide alphabet. Due to the bisulfite treatment, a thymine in a read can be aligned to a cytosine in the reference sequence. Similarly, a guanine in the reverse complement of a read can be aligned to an adenine in the reference sequence.

Supported GPUs

GPU-BSM works on CUDA enabled GPU-cards. It has been tested on two families of NVIDIA GPU cards: the NVIDIA FERMI architecture based GTX 480 card, and the NVIDIA Kepler architecture based k10 and k20c cards.

Multiple GPUs

GPU-BSM automatically detects the number of GPU installed in your computer and it runs in parallel the two (four) different alignments for directional (non-directional) libraries. For machine equipped with a single GPU card, GPU-BSM sequentially performs the different alignments.

Dependencies

GPU-BSM works on linux based systems with a custom installation of Python (release>=2.7.3) and equipped with a CUDA enabled GPU-card with cc 2.0. Moreover, GPU-BSM requires the installation of SOAP3-dp.
Currently, SOAP3-dp is also available for the latest release CUDA 5.0.

Install

Install SOAP3-dp

SOAP3-dp can be downloaded at the following addresses http://www.cs.hku.hk/2bwt-tools/soap3-dp/.

Run the following commands to extract the different programs:

```
% gunzip soap3-dp-<<release>>.tar.gz

% tar -xvf soap3-dp-<<release>>.tar.gz
```

GPU-BSM has been tested with the SOAP3-dp rel. 2.3.172

Install GPU-BSM

To install GPU-BSM run the following command

```
% pip install GPU-BSM
```

Usage

Indexing

As previously highlighted, GPU-BSM creates two sequences from the original forward genomic strand and a FM-index must be created for each of them.
To create the sequences and to build the indexes use the *"GPUBSM-builder"* command with the following options:

-h, --help	*show the help message and exit*
-r FILE, --reference=FILE	*The reference fasta file*
-i PATH, --index_path=PATH	*The directory of the indexes [indexes/]*
-S PATH, --soap3=PATH	*The path of SOAP3-dp [~/soap3-dp/]*
-R, --rrbs	*Use this option for RRBS experiments.*
-d STR, --red_site=STR	*Use this option to set the restriction enzime site: e.g., C-CGG for MspI digestion, T-CGA for TaqI digestion (only for RRBS data). [C-CGG]*
-m INT	*Minimum DNA fragments length compatible with the RRBS protocol [default: 40]*
-M INT	*Maximum DNA fragments length compatible with the RRBS protocol [defailt: 220]*

Syntax (*for WGBS experiments*):

% GPUBSM-builder -f <ref sequence file> -S <SOAP3-dp path>

For example:

% GPUBSM-builder -f reference.fa -S ~/soap3-dp/ # for WGBS experiments

Syntax (for RRBS experiments):

% GPUBSM-builder -f <ref sequence file> -R -d <r. enz.> -m <min DNA fragments length> -M <max DNA fragments length> -S <SOAP3-dp path>

For example:

% GPUBSM-builder -f reference.fa -R -d T-CGA -m 50 -M 200 -S ~/soap3-dp/

<ref sequence file> can be a file with a single reference sequence or a multi-fasta file.

Mapping

GPU-BSM uses SOAP3-dp to look for gapped or ungapped alignments. Mapping is performed in two steps. In the first step GPU-BSM looks for ungapped alignments that meet a given constraint on the allowed number of mismatches. Up to 4 mismatches are allowed for this step and none heuristic is used. In the second step, dynamic programming is exploited to look for gapped alignments. It is also possible to skip the first step with the aim to align all reads exploiting the dynamic programming and reducing the computing time.
By default, GPU-BSM has been designed to look for ungapped alignments in the first step with up to four mismatches. Users can change this value in GPU-BSM as well as disable ungapped alignments. We deem that this is a good constraint and did not modified it in GPU-BSM. Users can easily modify this value to decrease, increase, or avoid the upper limit to the alignments that may be found for each sequence. By default, GPU-BSM analyzes all 3-letter valid alignments found by SOAP3-dp. However, GPU-BSM also permits to analyze only the unique best alignment or only all best alignments obtained by SOAP3-dp.

To map the bisulfite-treated reads user can run the *"GPUBSM-aligner"* with the following options:

-h, --help	*show the help message and exit*
-s FILE, --reads=FILE	*The query file (FASTA or FASTQ format) [For alignments of single-end reads]*
-1 FILE, --reads1=FILE	*A query file (FASTA or FASTQ format) [For alignments of pair-end reads]*
-2 FILE, --reads2=FILE	*A query file (FASTA or FASTQ format)[For alignments of pair-end reads]*
-S SOAP3PATH, --soap3=SOAP3PATH	*The path of SOAP3-dp [~/soap3-dp/]*
-g GPU, --gpu=GPU	*Use this option to specify the a gpu identifier. If not specified GPU-BSM uses up to four GPU cards (up to two cards for methyl-seq libraries and up to four cards or BS-seq libraries).*

-u MAXINSERTSIZE, --max_insert_size= MAXINSERTSIZE	*Maximum value of insert size*
-v MININSERTSIZE, *--min_insert_size=MININSERTSIZE*	*Minimum value of insert size*
-i INDEX_PATH, --index_path=INDEX_PATH	*The directory of the indexes (generated in preprocessing genome) [indexes/]*
-m MISMATCHES, --mismatches=MISMATCHES	*Use this option to set the maximum number of mismatches allowed in the first step of SOAP3-dp [default 4]*
-l	*GPU-BSM discards those alignments obtained with more than "l" differences [default 10].*
-H TYPE_OF_HITS, --hits=TYPE_OF_HITS	*All valid alignments: 1 (DEAULT) - All best alignments: 2 - Unique best alignments: 3*
-l LIBRARY, --library=LIBRARY	*1 for directional and 2 for non-directional*
-L LENGTH, --length=LENGTH	*Length of the longest read in the query file [120]*
-R, --rrbs	*Use this option for RRBS experiments*
-d STR, --red_site=STR	*[Only for RRBS]. Use this option to set the restriction enzime: e.g.,C-CGG for MspI digestion, T-CGA for TaqI digestion (only for RRBS data). [C-CGG]*
-A ADAPTER	*Adapter sequence(s) to be removed form 3' of the reads. Adapters are trimmed using cutadapt. By default adapter trimming is not performed.*
-e ADP_ERROR_RATE	*Maximum allowed error rate for adapters (default: 0.1)*
-O ADP_OVERLAP, --overlap=ADP_OVERLAP	*Minimum overlap length. If the overlap between the read and the adapter is shorter than LENGTH, the read is not modified. This reduces the no. of bases trimmed purely due to short random adapter matches (default: 3).*
-a, --ambiguous	*Use this option to not remove ambiguous mapped reads*
--dp	*Use only dynamic programming to look for both gapped and ungapped alignments.*
--ungapped	*Use this option to look only for ungapped alignments*
-M, --methylation	*Use this option to calculate methylation levels*

Trimming of adapter sequences is performed with cutadapt[1].

*1. MARTIN, M.. Cutadapt removes adapter sequences from high-throughput sequencing reads. **EMBnet.journal**, North America, 17, may. 2011. Available at:*

<*http://journal.embnet.org/index.php/embnetjournal/article/view/200/479*>. *Date accessed: 04 Oct. 2013.*

Mapping single-end reads

Syntax *(for WGBS experiments)*:

% GPUBSM-aligner -s <reads file> -i <indexes path> -m <number of mismatches> -S <SOAP3-dp path> -l <bs protocol> -L <max length of the reads> -g <gpu-id>

For example to map a library with reads of length 76nt with up to 4 mismatches on a single GPU

% GPUBSM-aligner -s library.fastq -i indexes/ -m 4 -S ~/soap3-dp/ -l 1 -L 76 -g 0

Syntax *(for RRBS experiments)*:

% GPUBSM-aligner -s <reads file> -i <indexes path> -m <number of mismatches> -S <SOAP3-dp path> -l <bs protocol> -L <max length of the reads> -R -d <r. enz.> -g <gpu-id>

For example:

% GPUBSM-aligner -s library.fastq -i indexes/ -m 4 -S ~/soap3-dp/ -l 1 -L 76 -R -d T-CGA -g 0

To use alla available GPU do not use *-g* option.

Mapping pair-end reads

Syntax:

% GPUBSM-aligner -1 <reads file> -2 <reads file> -i <indexes path> -m <number of mismatches> -u <max insert size> -v <min insert size> -S <SOAP3 path> -l <bs protocol> -L <max length of the reads> g <gpu-id>

For example to map a library with reads of length 76nt with up to 4 mismatches

% GPUBSM-aligner -1 query1.fastq -2 query2.fastq -i indexes/ -m 4 -u 500 -v 200 -S ~/soap3/ -l 1 -L 76 -g 0

Output

Results are stored in the output directory. In particular, for each job a new directory named with a job identifier is automatically generated. Results are stored using the SAM format. For each mapped read the following field are reported:

QNAME	Read identifier

FLAG	Only bit 0x10
RNAME	Chromosome name
POS	Mapping position
MAPQ	Mapping quality
CIGAR	CIGAR string
RNEXT	Reference name of the mate/next segment
PNEXT	Position name of the mate/next segment
TLEN	Observer template length
SEQ	Segment sequence
QUAL	Phred-scaled base quality 33
XM	Methylation call string (if -M option is used)
XR	Read conversion for the alignment
XG	Genome conversion for the alignment

License

Contact

Please contact Andrea Manconi (andrea.manconi@itb.cnr.it) for problems and suggestions.