HBS-tools user guide (v1.1)

HBS-tools user guide (v1.1) *1*

1. General information 2

1.1 Download 2

1.2 Software requirements 2

2. Introduction 2

3. Installation 3

4. Algorithm Overview 3

4.3 hbs_methylation_extrator 4

4.4 hbs_cg_mlmf and hbs_ch_ml 4

5. Usage 5

5.1 hbs_process 5

5.2 hbs_mapper 6

5.3 hbs_methylation_extractor 8

5.4 hbs_cg_mlmf and hbs_ch_ml 8

6. Output explanation 9

6.1 hbs_process output 9

6.2 Mapper output output 9

6.3 hbs_methylation_extractor output 10

6.4 hbs_cg_mlmf and hbs_ch_ml output 11

Reference 12

1. General information

1.1 Download

HBS-tools can be downloaded from: http://sourceforge.net/projects/hbs-tools

1.2 Software requirements

HBS-tools are a set of command line based programs implemented in Perl and C. Hence a working version of Perl is required. To speed up the global alignment between read 1 and read 2, the Needleman-Wunsch alignment is implemented in ANSI C as 'nw_align.c'. It has been compiled under Ubuntu Linux. If 'nw_align' is not executable, please go to the software folder and compile it by running the following command: 'gcc nw_align.c –o nw_align'. After running the command, the 'nw_align' file will be created in the software folder.

Several programs may be invoked during the processing and mapping steps: cross_match (Gordon, et al., 1998) will be invoked by hbs_process for sequence matching during pre-processing step, and Bowtie1 or Bowtie2 (Langmead and Salzberg, 2012; Langmead, et al., 2009) will be invoked by hbs_mapper for mapping reads to the reference genome. Thus, please make sure these programs have already been installed properly.

2. Introduction

HBS-tools are a set of tools for the processing and analysis of hairpin-bisulfite sequencing (hairpin-BS-Seq) data (Zhao, et al., 2014). It can accept raw hairpin-BS-Seq data in FASTQ format, perform pre-processing, recover original sequences from read 1 and read 2, align the original sequences to the reference genome, and get methylation calling. The functions of core modules are summarized in **Table 1**.

Table 1 Summary of the programs included in HBS-tools

Module name	Function
hbs_process	Pre-processing of raw reads, including bad quality bases trimming, sequencing adaptor and hairpin adaptor removal
hbs_mapper	Original sequence recovery, mapping, and SAM file output
hbs_methylation_extractor	Extract and output methylation pattern from the SAM file
hbs_cg_ml_mf	Summarize the methylation level and fidelity for covered CpG sites
hbs_ch_ml	Summarize the methylation level for covered non_CpG sites

3. Installation

The software can be downloaded here: http://sourceforge.net/projects/hbs-tools. Use the following command to decompress the hbs_tools_v1.1.tar.gz file:

tar xzf hbs_tools_v1.1.tar.gz

The core scripts are written in PERL, and can be used directly. However, the Needleman-Wunsch alignment, which will be used by hbs_mapper, is implemented in ANSI C as 'nw_align.c'. It has been compiled under Ubuntu Linux. If 'nw_align' is not executable, please go to the software folder and compile it by running the following command: 'gcc nw_align.c –o nw_align'. Accordingly, the 'nw_align' file will be created in the software folder. Now the HBS-tool is ready to be used.

Finally, the users should make sure cross_match (Gordon, et al., 1998) and Bowtie1 or Bowtie2 (Langmead and Salzberg, 2012; Langmead, et al., 2009) has already been installed properly.

4. Algorithm Overview

4.1 hbs_process

The hbs_process is designed for the processing of hairpin-BS-Seq data. It takes raw fastQ file from hairpin-BS-Seq as input, and integrates functions including: 1) trimming bad quality residues from the input sequence; 2) filtering hairpin adaptors; 3) filtering sequencing adaptors; 4) discards read pairs with any read shorter than the given threshold after the 1-3 steps.

4.2 hbs_mapper

The hbs_mapper is the program for mapping hairpin-BS-Seq reads to the reference genome, and obtaining methylation calls subsequently. Unlike previously published tools which usually map bisulfite-converted reads to reference genomes after C-to-T and G-to-A conversion of both reads and reference genomes, hbs_mapper fully takes advantage of the merits of hairpin-BS-Seq reads, and utilized a special recover-then-mapping strategy for read mapping. In brief, it first recovers the original sequences after globally alignment of read1 and read2 with the Needleman-Wunsch algorithm using a modified scoring matrix which tolerates inconsistence due to bisulfite conversion (C-to-T in read1 and G-to-A in read2), and then maps the recovered sequences to the reference using Bowtie1 or Bowtie2 (Langmead and Salzberg, 2012; Langmead, et al., 2009). By mapping it this way, it overcomes the reduced sequence complexity which is evident for traditional BS-Seq, and thus improves mapping efficiency.

After global alignment of read1 and read2, the original sequence is recovered by following four simple rules: (a) a T in read1 and a C in read2 represents a C-to-T conversion during bisulfite treatment and hence the original sequence must have had a C. (b) a G in read1 and an A in read2 represents a G-to-A conversion and hence

the original sequence must have had a G. (c) when read1 and read2 have the same nucleotide it represents no modification and hence stays the same in the recovered original sequence. (d) when read1 and read2 have different nucleotide that are not due to C-to-T or G-to-A conversion, the one with the better quality score will be kept. The recovered original sequence is then mapped to the reference genome using Bowtie1 or Bowtie2 (Langmead and Salzberg, 2012; Langmead, et al., 2009). Having tracked the reference genome fragment that corresponds to the original sequence, the raw read1 and read2 are compared to the reference genome fragment to call the methylation statuses for covered cytosines.

The methylation calls and the alignment information are generated in standard SAM format. The output contains information such as read ID, chromosome, genomic position and methylation calls. The methylation call string is designed in a fashion so as to represent the methylation statuses of cytosines in three possible contexts. The small and capital letters of 'z', 'x' and 'h' are used to represent the un-methylated and methylated events at CpG, CHG and CHH sites, respectively. The mapping output can be used for post-processing to extract methylation call information of individual cytosine. Other nucleotide bases other than cytosine in read 1 and guanine in read2 are represented with a dot '.'.

The alignment output writing follows the methylation calling. The output is written in SAM format (Li, et al., 2009) along with the methylation call strings. Hence each read pair will have two lines in the output file.

4.3 hbs_methylation_extrator

The hbs_methylation_extractor takes the SAM file generated by hbs_mapper as input, parses the methylation call strings, and extracts the methylation statuses for the cytosines covered by hairpin-BS-Seq reads. It provides the options to either output methylation information for CpG and non-CpG contexts separately or together.

In the hbs_mapper output, each line represents the mapping and methylation call for a sequence read. In the extractor output, each line contains the information for the methylation status of one cytosine covered by a sequence read. Apart from the read ID and methylation status of a cytosine, the extractor output also contains chromosome, genomic coordinate and strand information.

4.4 hbs_cg_mlmf and hbs_ch_ml

Instead of the methylation pattern obtained from each read, what we are most interested in are usually the methylation patterns for each CpG dyads along the genome. Thus two simple yet useful scripts, hbs_cg_ml_mf and hbs_ch_ml, were designed to summarized the methylation pattern for CpG and non-CpG sites, respectively. hbs_cg_ml_mf takes the CpG methylation callings generated from the hbs_methylation_extractor as input, and calculates the methylation level, methylation fidelity and other related information for each covered CpG site along

the genome. Similarly, hbs_ch_ml takes the non_CpG methylation calling result as input, and calculates the methylation level for each covered non_CpG site. The output can be used for visualization, or further comparison between different samples.

Methylation level shows the number of methylated C's out of the total number of C's. We can define this as:

$$ML(C) = \frac{reads(mC)}{reads(mC) + reads(C)}$$

However since CpG dyads have multiple classifications such as unmethylated, asymmetrically methylated, or symmetrically methylated, we can define the methylation fidelity as:

$$MF(CpG) = \frac{reads(mCG/mCG) + reads(CG/CG)}{reads(mCG/mCG) + reads(mCG/CG) + reads(CG/mCG) + reads(CG/CG)}$$

where reads(mCG/mCG) is the number of fully methylated CpG dyad detected, reads(CG/CG) is the number of fully unmethylated CpG dyad detected, and reads(mCG/CG) and reads(CG/mCG) is the number of hemi-methylated CpG dyad detected for a given CpG site.

5. Usage

5.1 hbs_process

USAGE:

hbs_process [options] <read file 1> <read file 2> <sequencing adaptor file> <hairpin adaptor file>

INPUT FILES:

<read file 1>	**FastQ file for read 1**
<read file 2>	**FastQ file for read 2**
<sequencing adaptor file>	**FastA file for sequencing adaptor. If not sure, you can use the last 24 bp of Illumina adaptor "ATCTCGTATGCCGTCTTCTGCTTG" to make a fasta file.**
<hairpin adaptor file>	**FastA file for hairpin sequence. The script will automatically make the C>T and G>A converted hairpins for read1 and read2, respectively.**

OPTIONS:

--phred33-quals	**FASTQ qualities are ASCII chars equal to the Phred quality plus 33. Default: on.**

--phred64-quals	**FASTQ qualities are ASCII chars equal to the Phred quality plus 64. Default: off.**
--length	**Length cut-off, default = 10. Read pairs with any read shorter than the threshold will be discarded.**
--help	**Show help information**

EXAMPLE:

hbs_process --phred33-quals --length 10 read1.fq read2.fq adaptor.fa hairpin.fa

5.2 hbs_mapper

USAGE: hbs_mapper [options] <bowtie index> <read1 file> <read2 file>

ARGUMENTS:

<bowtie index>	**Prefix for index generated by bowtie-build or bowtie2-build**
<read1 file>	**Read file1, can be fasta or fastq format**
<read2 file>	**Read file2, can be fasta or fastq format. Read1 and read2 files should be of the same format, and reads are of identical order.**

OPTIONS:

-f/--fasta	**Input files are of fastA format**
-q/--fastq	**Input files are of fastQ format**
-i/--identity	**Minimum identity required between read pairs (between 0 to 1, default: 0.9)**
-s/--skip <int>	**Skip the first <int> reads/pairs from the input (default: no skip)**
-u/--upto <int>	**Only aligns the first <int> reads/pairs from the input (default: no limit)**
--phred33-quals	**FASTQ qualities are ASCII chars equal to the Phred quality plus 33. (default: on)**
--phred64-quals	**FASTQ qualities are ASCII chars equal to the Phred quality plus 64. (default: off**

--path_to_bowtie	**The full path to the Bowtie (1 or 2). If not specified, it is assumed that Bowtie (1 or 2) is in the PATH.**
--keep-temp	**Keep the temporary files generated during the running of hbs_mapper. 1 for keep and 0 for delete. (default: 0)**

Bowtie1 OPTIONS:

-l/--seedlen <int>	**Seed length; i.e., the number of bases of the high quality end of the read to which the -n ceiling applies. (default: 28)**
-n/--seedmms <int>	**Maximum number of mismatches permitted in the "seed", i.e. the first L base pairs of the read (where L is set with -l/--seedlen). (0|1|2|3, default: 2)**
-e/--maqerr <int>	**Maximum permitted total of quality values at all mismatched read positions throughout the entire alignment, not just in the "seed". (default: 70)**
-m <int>	**Suppress all alignments if > <int> exist (default: 1)**
--best	**Hits guaranteed best stratum; ties broken by quality**
--no_best	**Disables the --best option which is on by default**
--chunkmbs <int>	**Megabytes of memory a given thread is given to store path descriptors (default 512)**

Bowtie 2 OPTIONS

--bowtie2	**Use bowtie2 instead of bowtie1**
-N <int>	**Mismatches allowed (0|1, default: 0)**
-L <int>	**Seed length to align during multispeed alignment(default: 22)**
--ignore-quals	**When calculating a mismatch penalty, always consider the quality value at the mismatched position to be the highest possible, regardless of the actual value.**
-D <int>	**Up to <int> consecutive seed extension attempts can "fail" before Bowtie 2 moves on, using the alignments found so far (default: 15)**

-R <int>	**The maximum number of times Bowtie 2 will "re-seed" reads with repetitive seeds (default: 2)**
--score_min <func>	**Sets a function governing the minimum alignment score needed for an alignment to be considered "valid" (default: L,0,-0.2)**

Output options:

-o/--output_dir <dir>	**Write all output files into this directory (default: ./)**
--sam-no-hd	**Suppress SAM header lines (starting with @) (default: output SAM header lines)**

Other options:

-h/--help	**Display help information**
-v/--version	**Display version information**

5.3 hbs_methylation_extractor

USAGE: hbs_methylation_extractor [options] <filenames>

ARGUMENTS:

<filenames>	**A space-separated list of HBS_mapper result files in SAM format**

OPTIONS:

--merge_non_CpG	**This will produce two output files for Cs in CpG context and non-CpG context, respectively**
--report	**Prints out a short methylation summary as well as the paramaters used to run this script.**
--version	**Displays version information.**
-h/--help	**Displays this help file and exits.**

5.4 hbs_cg_mlmf and hbs_ch_ml

The hbs_cg_mlmf is used to summarize the methylation level and fidelity for each covered CpG dyad.

Usage:

hbs_cg_mlmf <hbs_methylation_extractor results for CpG context>

The hbs_ch_ml is used to summarize the methylation level for each covered non_CpG site.

Usage:

hbs_ch_ml <hbs_methylation_extractor results for Non_CpG context>

NOTE: Since the methylation callings are outputted without sorting by genomic coordinates, this script needs to read into all methylation calling into memory, thus may need huge memory. To reduce memory usage, it is suggested to cut data to each chromosome, and then calculate separately.

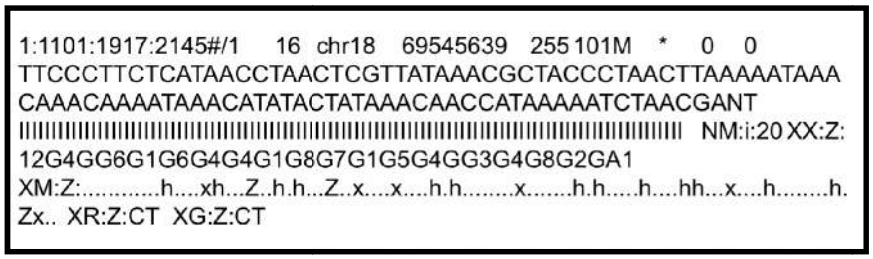
6. Output explanation

6.1 hbs_process output

The hbs_process script takes raw paired-end hairpin-BS-Seq data (FastQ format) as input, and output two files with suffix ".processed" for read 1 and read 2, respectively.

6.2 Mapper output output

As mentioned before, the hbs_mapper output is in SAM format but with the added methylation call information. A typical output line will look like the following:



The above figure contains sixteen tab separated columns and their explanation are as listed below:

SEQ_ID: Read ID. '/1' and '/2' are used at the end of the SEQ_ID to denote the read 1 and 2, respectively

FLAG: Bitwise FLAG [0, $2^{16} - 1$]

CH_ID: Chromosome number

POS: Genomic start position

MAPQ: Mapping quality is invariably 255 indicating that the mapping quality is not available

CIGAR: CIGAR string

RNEXT: Reference name of the mate/next read

PNEXT: Position of the mate/next read

TLEN: Observed Template LENgth

SEQ: Genomic fragment mapped to the original sequence

QUAL: Quality score in Phred33 format. 'I' in the place of the quality scores denote that quality scores are not stored

NM-tag: Edit distance to the reference

XX-tag: Mismatches between the read and reference

XM-tag: Methylation call string

XR-tag: Read conversion state for the alignment

XG-tag: Genome conversion state for the alignment

6.3 hbs_methylation_extractor output

The hbs_methylation_extractor, as mentioned in the algorithm section, prints the methylation status of the read1 cytosines and read 2 guanines. Read 2 guanines represent cytosines in the pre-bisulfite treated DNA. The following is a typical methylation extractor output:

```
hbs_methylation_extractor version 1.1
1:1101:12253:1835#/1    chr14   +       101102590       Z
1:1101:12253:1835#/1    chr14   +       101102612       Z
1:1101:12253:1835#/2    chr14   -       101102591       Z
1:1101:12253:1835#/2    chr14   -       101102613       Z
1:1101:18833:1928#/1    chr4    -       77993707        Z
1:1101:18833:1928#/2    chr4    +       77993706        Z
1:1101:19604:1962#/1    chr3    +       14362390        Z
1:1101:19604:1962#/1    chr3    +       14362437        Z
1:1101:19604:1962#/2    chr3    -       14362391        Z
1:1101:19604:1962#/2    chr3    -       14362438        Z
```

The hbs_methylation_extractor output in a tab separated text file. The columns in the output are as follows:

SEQ_ID: Read ID. '/1' and '/2' are used at the end of the SEQ_ID to denote the read 1 and 2, respectively

CHR_ID: Chromosome name

STRAND: '+' and '-' signs represent the cystosine occurs in the Watson (+) or Crick (-) strand, respectively. For a CpG dyad, the C is with "+", and the following G with "-".

POS: Genomic position

M_Call: Methylation calling as explained in table 1. Figure 2 shows a series of x and X as the output refers to the CHG context.

The hbs_methylation_extractor has one parameter '--merge_non_CpG'. Typically the methylation extractor program creates six files and they are as follows:

Methylation call for read 1 CpG

Methylation call for read 2 CpG

Methylation call for read 1 CHG

Methylation call for read 2 CHG

Methylation call for read 1 CHH
Methylation call for read 2 CHH

The '--merge_non_CpG' parameter combines all non-CpG methylation calls i.e. methylation calls from CHG and CHH sites.

6.4 hbs_cg_mlmf and hbs_ch_ml output

The hbs_cg_mlmf output prints out the methylation level and fidelity (as well as other statistics) for each covered CpG dyad. See below for a typical methylation fidelity output:

chr19	12003045	0.31250	1.00000	10	0	0	22	0	0	0	0
chr19	12003072	0.75926	0.96296	20	1	0	6	0	0	0	0
chr19	12003112	0.69767	1.00000	30	0	0	13	0	0	0	0
chr19	12003120	0.51163	0.30233	7	4	26	6	0	0	0	0
chr19	12008806	0.86885	0.72414	21	8	0	0	3	0	0	0
chr19	12041321	0.86420	0.72500	29	11	0	0	1	0	0	0

The output is displayed in a tab separated text file. The columns in the output file are as follows:

1. chromosome name
2. CpG position
3. methylation level
4. methylation fidelity
5. count of mC/mC
6. count of mC/C
7. count of C/mC
8. count of C/C
9. count of mC/?
10. count of C/?
11. count of ?/mC
12. count of ?/C

NOTE: ? for missed methylation pattern. Due to reasons such as sequencing errors, for certain CpGs, occasionally only one read from a read pair could report a methylation calling.

The hbs_ch_ml script outputs the methylation pattern for each covered non_CpG sites. The output is displayed in a tab separated text file. The columns in the output file are as follows:

1. chromosome name
2. non_CpG position
3. methylation level
4. count of mC and C
5. count of mC

Reference

Gordon, D., Abajian, C. and Green, P. Consed: a graphical tool for sequence finishing. *Genome research* 1998;8(3):195-202.

Langmead, B. and Salzberg, S.L. Fast gapped-read alignment with Bowtie 2. *Nature methods* 2012;9(4):357-359.

Langmead, B.*, et al.* Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology* 2009;10(3):R25.

Li, H.*, et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 2009;25(16):2078-2079.

Zhao, L.*, et al.* The dynamics of DNA methylation fidelity during mouse embryonic stem cell self-renewal and differentiation. *Genome research* 2014;24(8):1296-1307.