Methpipe Manual

1 Pre-mapping processing

read-quality-prof.cpp This program take a fastq file as input and output the base composition and quality scores for each column

quality-prof.R This R script define a function that takes the output of read-quality-prof.cpp as input and draw the figure of base composition

trim-adapter.cpp This program expects a fastq file as input and trim the adapter sequence from the 3' end of reads if there is.

visireads.cpp This program takes a fastq file as input and output a BED file displaying Cs in the sequences

2 Mapping

rmapbs.cpp This program takes fastq file as input and output mapped read file

3 Post-mapping processing

merge.cpp merge sorted MappedRead file, with option to control whether remove redundant file

mask-overlap.cpp mask overlapping region of paired-reads, also generate summary of fragment length, number of unpaired reads

unique.cpp filter program to remove duplicate reads

sort.cpp sort MappedRead file: either by genomic location or by name

revcomp.cpp do reverse complement operation on MappedRead

4 Analysis

methcount.cpp This program reads mapped read file and output methylation frequency for each CpG site.

bsrate.cpp This program reads mapped read file and estimate bisulfite conversion rate by checking methylation status of non-CpG C's

5 A sample work flow

This part shows how these tools are connected to get the methylation profile. Suppose we have the following BS-Seq library

```
reads/s_1_1.txt reads/s_1_2.txt reads/s_2_1.txt reads/s_2_2.txt
```

The final result can be obtained as following. For clarity, we show all the intermediate result. In real application, some intermediate files can be avoided by using pipes.

```
# Pre-mapping processing #

## trim adapter ##
$ ./trim-adapter reads/s_1_1.txt preprocessed/s_1_1.txt
$ ./trim-adapter reads/s_1_2.txt preprocessed/s_1_2.txt
$ ./trim-adapter reads/s_2_1.txt preprocessed/s_2_1.txt
$ ./trim-adapter reads/s_2_2.txt preprocessed/s_2_2.txt

# Mapping #
$ ./rmapbs -c genome_seq_dir -o mapped/s_1_1.mr preprocessed/s_1_1.txt
$ ./rmapbs -c genome_seq_dir -o mapped/s_1_2.mr preprocessed/s_1_2.txt
$ ./rmapbs -c genome_seq_dir -o mapped/s_2_1.mr preprocessed/s_2_1.txt
$ ./rmapbs -c genome_seq_dir -o mapped/s_2_2.mr preprocessed/s_2_2.txt

# post-mapping processing

## reverse complement A-rich strand ##
$ ./revcomp mapped/s_1_2.mr > tmpfile && mv tmpfile mapped/s_1_2.mr
$ ./revcomp mapped/s_2_2.mr > tmpfile && mv tmpfile mapped/s_2_2.mr

## mask overlapping ##
#### first sort by name ####
$ ./sort -N mapped/s_1_1.mr -o tmpfile && mv tmpfile  mapped/s_1_1.mr
$ ./sort -N mapped/s_1_2.mr -o tmpfile && mv tmpfile  mapped/s_1_2.mr
$ ./sort -N mapped/s_2_1.mr -o tmpfile && mv tmpfile  mapped/s_2_1.mr
$ ./sort -N mapped/s_2_2.mr -o tmpfile && mv tmpfile  mapped/s_2_2.mr

#### masking ####
$ ./mask-overlap mapped/s_1_1.mr mapped/s_1_2.mr masked/s_1_1.mr masked/s_1_2.mr
$ ./mask-overlap mapped/s_2_1.mr mapped/s_2_2.mr masked/s_2_1.mr masked/s_2_2.mr

#### sort by genomic location ####
$ ./sort masked/s_1_1.mr -o tmpfile && mv tmpfile  masked/s_1_1.mr
$ ./sort masked/s_1_2.mr -o tmpfile && mv tmpfile  masked/s_1_2.mr
$ ./sort masked/s_2_1.mr -o tmpfile && mv tmpfile  masked/s_2_1.mr
$ ./sort masked/s_2_2.mr -o tmpfile && mv tmpfile  masked/s_2_2.mr

## combine all result ##
#### merge ####
$ ./merge -o all.mr masked/s_1_1.mr masked/s_1_2.mr masked/s_2_1.mr masked/s_2_2.mr

#### jackpot removal #####
$ ./unique all.mr -o tmpfile && mv tmpfile all.mr
```

```
# analysis #
## methcounts ##
$ ./methcounts -c genome_sequence_file -o all-methcounts.bed all.mr
```