RRBS SCP

https://github.com/ltlam/RRBS_SCP

Introduction

This guide provides steps for performing alignment and sample visualization of MspI digested RRBS reads.

- Convert qseq to fastq
- BSseeker2 alignment & methylation calling
- RRBS coverage calculation
- Sample methylation distribution & dendrogram using [R]
- Fragment methylation calculation

Requirements

- Unix environment
- Python 2.6+
- R

Software/Packages

- Bowtie2 (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml)
- Pysam (https://code.google.com/archive/p/pysam/)
- BSseeker2 (https://github.com/BSSeeker/BSseeker2)
- ggplot2 (http://ggplot2.org/)
- reshape2 (https://github.com/hadley/reshape)

Data

The sample files were extracted from MspI digested human RRBS reads that have been demultiplexed. Each file contains reads from chr19.

- /data/TBS_14A_chr19.qseq.gz
- /data/TBS_14B_chr19.qseq.gz
- /data/TBS_18C_chr19.qseq.gz
- /data/TBS_26A_chr19.qseq.gz

- /data/TBS_50A_chr19.qseq.gz

Convert qseq to fastq

This step is not required for alignment but may be useful if other tools require fastq files.

1. unzip test files

```
cd <path to gzipped qseq files>
gunzip *.gz
```

2. Convert qseq files to fastq

```
cat TBS_14A_chr19.qseq | java -jar qseq2fastq.jar -phred64 > TBS_14A_chr19.fastq
cat TBS_14B_chr19.qseq | java -jar qseq2fastq.jar -phred64 > TBS_14B_chr19.fastq
cat TBS_18C_chr19.qseq | java -jar qseq2fastq.jar -phred64 > TBS_18C_chr19.fastq
cat TBS_26A_chr19.qseq | java -jar qseq2fastq.jar -phred64 > TBS_26A_chr19.fastq
cat TBS_50A_chr19.qseq | java -jar qseq2fastq.jar -phred64 > TBS_50A_chr19.fastq
```

BSseeker2 Alignment & Methylation Calling

This step assumes you have setup both BSseeker2 and bowtie2. This step also assumes an RRBS index file was built with fragment range of 40-500bp.

1. Alignment using BSseeker2/bs_seeker2-align.py

- -i = input file of reads
- -m = number of mismatches allowed
- --aligner = bowtie2, installed aligner
- --bt2 = bowtie2 alignment method, --end-to-end recommended
- -o = output file
- -f = output file format, bam
- -g = RRBS indexed reference genome
- -r = RRBS alignment flag
- --low = lower bound of RRBS fragment
- --up = upper bound of RRBS fragment
- -a = path to adapter sequence file adapter.txt, CGAGATCGGAAGAGCACACGTC recommended for MspI

```
python bs_seeker2-align.py -i TBS_14A_chr19.fastq -m 5 --aligner=bowtie2 --bt2--end-t
o-end -o TBS_14A_chr19.bam -f bam -g hg19_tb.fa -r --low=40 --up=500 -a adapter.txt
python bs_seeker2-align.py -i TBS_14B_chr19.fastq -m 5 --aligner=bowtie2 --bt2--end-t
o-end -o TBS_14B_chr19.bam -f bam -g hg19_tb.fa -r --low=40 --up=500 -a adapter.txt
python bs_seeker2-align.py -i TBS_18C_chr19.fastq -m 5 --aligner=bowtie2 --bt2--end-t
o-end -o TBS_18C_chr19.bam -f bam -g hg19_tb.fa -r --low=40 --up=500 -a adapter.txt
python bs_seeker2-align.py -i TBS_26A_chr19.fastq -m 5 --aligner=bowtie2 --bt2--end-t
o-end -o TBS_26A_chr19.bam -f bam -g hg19_tb.fa -r --low=40 --up=500 -a adapter.txt
python bs_seeker2-align.py -i TBS_50A_chr19.fastq -m 5 --aligner=bowtie2 --bt2--end-t
o-end -o TBS_50A_chr19.bam -f bam -g hg19_tb.fa -r --low=40 --up=500 -a adapter.txt
```

2. Methylation calling with BSseeker2/bs_seeker2-call_methylation.py

- -x = enable filtering of reads not fully converted by bisulfite treatment
- -i = input bam file
- -o = output prefix
- --db = RRBS reference index folder path

```
python bs_seeker2-call_methylation.py -x -i TBS_14A_chr19.bam -o TBS_14A_chr19 --db <
BSSeeker2 Path>/bs_utils/reference_genomes/hg19_tb.fa_rrbs_40_500_bowtie2/
python bs_seeker2-call_methylation.py -x -i TBS_14B_chr19.bam -o TBS_14B_chr19 --db <
BSSeeker2 Path>/bs_utils/reference_genomes/hg19_tb.fa_rrbs_40_500_bowtie2/
python bs_seeker2-call_methylation.py -x -i TBS_18C_chr19.bam -o TBS_18C_chr19 --db <
BSSeeker2 Path>/bs_utils/reference_genomes/hg19_tb.fa_rrbs_40_500_bowtie2/
python bs_seeker2-call_methylation.py -x -i TBS_26A_chr19.bam -o TBS_26A_chr19 --db <
BSSeeker2 Path>/bs_utils/reference_genomes/hg19_tb.fa_rrbs_40_500_bowtie2/
python bs_seeker2-call_methylation.py -x -i TBS_50A_chr19.bam -o TBS_50A_chr19 --db <
BSSeeker2 Path>/bs_utils/reference_genomes/hg19_tb.fa_rrbs_40_500_bowtie2/
```

Calculate Coverage Across RRBS Fragments

This step reports the mean number of reads aligned across the RRBS fragments. The mappable region file is located in the BSseeker2 subdirectory.

example:
BSseeker2/bs_utils/reference_genomes/hg19_tb.fa_rrbs_40_500_bowtie2/RRBS_mappable_regions.txt

1. use /scripts/bam_to_cov.py

The script outputs to a tab delimeted .cov file. The first value in the .cov files is the sum of the coverage level at each bp. The second value is the total number of mappable bp. The third value is the sum of the coverage

divided by number of mappable bases.

- -r = RRBS mappable region file
- -b = input bam file

```
python bam_to_cov.py -r RRBS_mappable_regions_chr19.txt -b TBS_14A_chr19.bam_sorted.b
am
python bam_to_cov.py -r RRBS_mappable_regions_chr19.txt -b TBS_14B_chr19.bam_sorted.b
am
python bam_to_cov.py -r RRBS_mappable_regions_chr19.txt -b TBS_18C_chr19.bam_sorted.b
am
python bam_to_cov.py -r RRBS_mappable_regions_chr19.txt -b TBS_26A_chr19.bam_sorted.b
am
python bam_to_cov.py -r RRBS_mappable_regions_chr19.txt -b TBS_50A_chr19.bam_sorted.b
am
```

File	Total Coverage	Total Mappable Positions (bp)	Coverage
TBS_14A_chr19.bam_sorted.bam.cov	76001603	12251172	6.203619
TBS_14B_chr19.bam_sorted.bam.cov	74028635	12251172	6.042576
TBS_18C_chr19.bam_sorted.bam.cov	94881136	12251172	7.744658
TBS_26A_chr19.bam_sorted.bam.cov	114096854	12251172	9.313138
TBS_50A_chr19.bam_sorted.bam.cov	26082039	12251172	2.128942

Sample Methylation Distribution & Dendrogram Using [R]

This step visualizes the methylation distribution across samples at common CpG sites with a minimum of 4x coverage.

0. gunzip tab delimeted plain text CGmap files (BSSeeker2 methylation level file https://github.com/BSSeeker/BSseeker2)

```
CGmap Column description:

(1) chromosome
(2) nucleotide on Watson (+) strand
(3) position
(4) context (CG/CHG/CHH)
(5) dinucleotide-context (CA/CC/CG/CT)
(6) methylation-level = #_of_C / (#_of_C + #_of_T).
(7) #_of_C (methylated C, the count of reads showing C here)
(8) = #_of_C + #_of_T (all Cytosines, the count of reads showing C or T here)
```

```
gunzip *.CGmap.gz
```

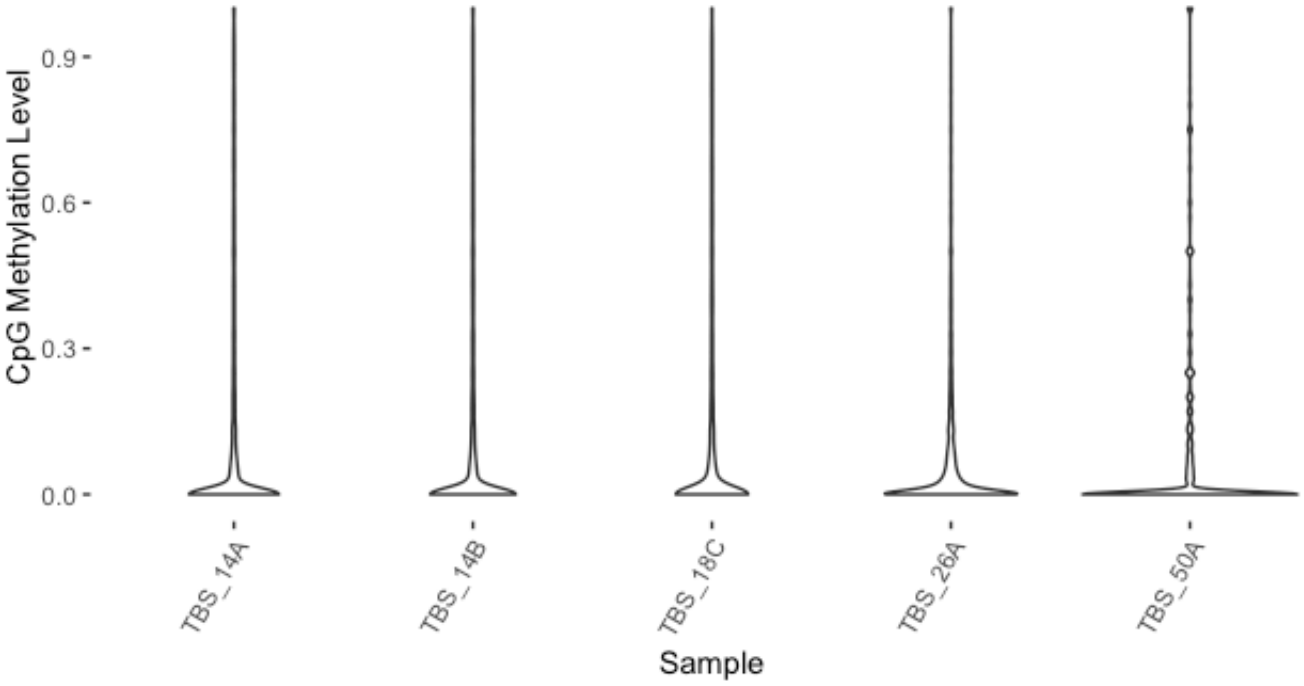
1. Generate table of common CpGs with minimum 4x coverage using /scripts/cgmap_data_frame.py

- -in_sam = comma separated list of cgmap files converted by bisulfite treatment
- -out_sam = output tab delimeted file
- -cov_sam = minimum coverage filter, default 10
- -f_sam = context filter CG/CHH/CHG, default CG

```
python cgmap_data_frame.py -in_sam TBS_14A_chr19.CGmap,TBS_14B_chr19.CGmap,TBS_18C_ch
r19.CGmap,TBS_26A_chr19.CGmap,TBS_50A_chr19.CGmap -out_sam TBS_Common_Meth.txt -cov_s
am 4
```

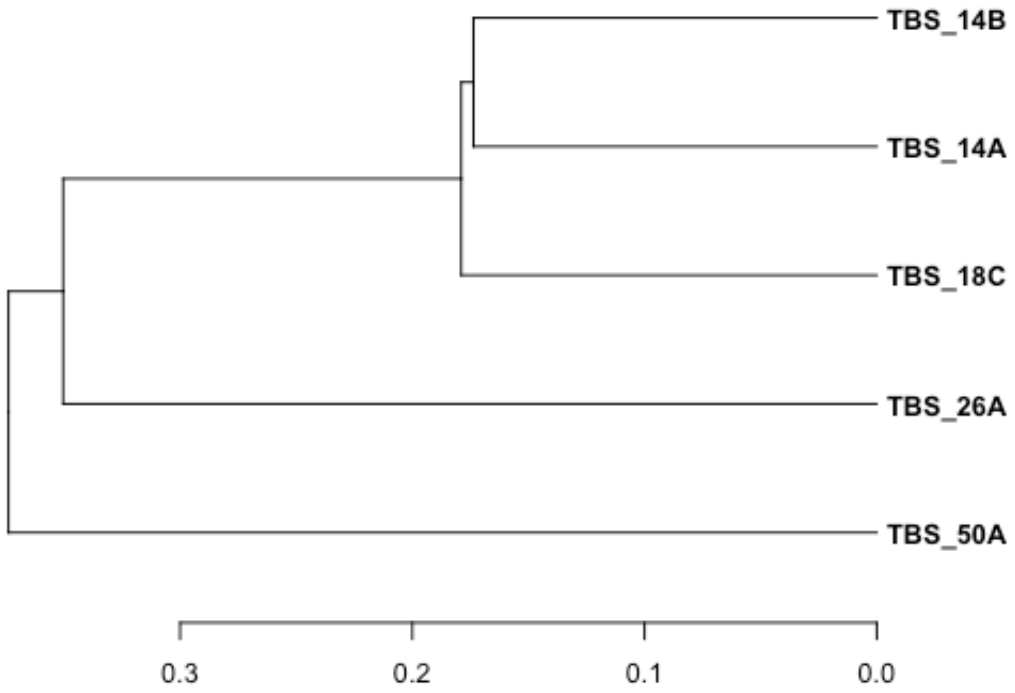
2. Generate violin plot in [R]

```
library(ggplot2)
library(reshape2)
meth_df = read.table("TBS_Common_Meth.txt",header=TRUE,sep='\t',row.names=1)
labels = gsub("_chr19$","",colnames(meth_df))
colnames(meth_df) = labels
mdata <- reshape2::melt(meth_df, id.vars = NULL)
colnames(mdata) = c("Sample","Methylation")
ggplot(mdata,aes(x = Sample, y = Methylation))+ geom_violin() +xlab("Sample")+ylab("C
pG Methylation Level")+ggtitle("") + theme(panel.grid.major = element_blank(), panel.
grid.minor = element_blank(), panel.background = element_blank(), axis.line = element
_line(colour = "black"))+ylim(0, 1.105)+theme(axis.text.x = element_text(angle = 60,
hjust = 1))
```



3. Generate Sample Dendrogram in [R]

```
dSpear = cor(meth_df, use="pairwise.complete.obs", method="spearman")
dSpear.dist = as.dist(1-dSpear)
dSpear.tree = hclust(dSpear.dist, method="average")
par(cex=0.8,font=2,mar=c(3,1,1,12))
plot(as.dendrogram(dSpear.tree),horiz=T,cex.lab=1, cex.axis=1, cex.main=0.5, cex.sub=
0.5)
```



RRBS Fragment Methylation

1. Bind the RRBS fragment to the common CpG methylation table using /scripts/pos_to_frag.py

The script will create a new methylation table (_frag.txt) that will include a fragment column (FRAG) for each CpG row from the sample methylation table.

- -m = mappable region file
- -f = input sample methylation table

```
python pos_to_frag.py -m RRBS_mappable_regions_chr19.txt -f TBS_Common_Meth.txt
```

2. Collapse the methylation values to the fragment level /scripts/frag_avg.py

The script will merge the CpGs/rows to the fragments into a new methylation table (_frag_avg.txt). The script will also add a new column (NUM_CG) to indicate the number of CpGs within the fragment.

```
python frag_avg.py -f TBS_Common_Meth_frag.txt
```