Gerstein Lab, Yale - PI: mark@gersteinlab.org, dev: donghoon.lee@yale.edu

Supplementary Information for Gerstein Lab STARR-seq Pipeline

0. Preamble

This document describes an analysis pipeline for the ORI-STARR-seq library consists of 160 replicate sub-reactions. We assume all sequencings were performed at a minimum of 100bp paired-end for a full lane depth.

1. Mapping and preprocessing

Prerequisites

- BWA v0.7.17
- SAMtools v1.5
- Picard v2.9.0
- GRCh38 genome (https://www.encodeproject.org/files/GRCh38_no_alt_analysis_set_GCA_0000001405.15/)

Processing Summary

For each of 160 replicates, paired-end sequencing reads were aligned to the human reference genome GRCh38 using BWA-mem. First, alignments were passed to Picard and optical duplicates were removed. Second, deduplicated alignments were filtered using SAMtools for unmapped, secondary (supplementary) alignments, mapping quality score less than 30, and PCR duplicates. All of the replicates were pooled and sorted for downstream analysis.

Code Example

```
PREFIX={specify prefix}
READ1={specifiy path to read 1}
READ2={specifiy path to read 2}
```

```
GENOME=/path/to/GRCh38_no_alt_analysis_set_GCA_000001405.15.fasta

CPU_COUNT=2
MAX_MEM=16g
OUTDIR=align

mkdir -p $OUTDIR

### alignment
bwa mem -t 4 "$GENOME" "$READ1" "$READ2" > "$OUTDIR"/"$PREFIX".sam
samtools sort -@ "$CPU_COUNT" -o "$OUTDIR"/"$PREFIX"_sorted.bam "$OUTDIR"/"$PREFIX".sam
rm "$OUTDIR"/"$PREFIX".sam

### remove duplicate
java -Xmx$MAX_MEM -jar $EBROOTPICARD/picard.jar MarkDuplicates
INPUT="$OUTDIR"/"$PREFIX"_sorted.bam OUTPUT="$OUTDIR"/"$PREFIX"_sorted_dedup.bam
CREATE_INDEX=true ASSUME_SORTED=true REMOVE_DUPLICATES=true VALIDATION_STRINGENCY=LENIENT
METRICS_FILE="$OUTDIR"/"$PREFIX"_sorted_dedup.picard_metrics.txt

### filter: -F excludes reads with any bits set in FLAG; -f requires FLAG 2: properly
aligned; -q 40 excludes MAPQ < 40;

# 3852
#    4 read unmapped (0x4)
#    8 mate unmapped (0x8)
#  256 not primary alignment (0x100)
#  512 read fails platform/vendor quality checks (0x200)
# 1024 read is PCR or optical duplicate (0x400)
# 2048 supplementary alignment (0x800)

samtools view -F 3852 -f 2 -q 40 -b "$OUTDIR"/"$PREFIX"_sorted_dedup.bam -o
"$OUTDIR"/"$PREFIX"_sorted_dedup_filtered.bam
samtools index "$OUTDIR"/"$PREFIX"_sorted_dedup_filtered.bam
```

2. Peak calling

Prerequisites

- STARRPeaker v1.0 - https://github.com/gersteinlab/starrpeaker
- Covariates bigWig(s) - http://gofile.me/4kBY9/EKNVQrVHM
- GRCh38 genome (https://www.encodeproject.org/files/GRCh38_no_alt_analysis_set_GCA_000001405.15/)
- GRCh38 blacklist - https://www.encodeproject.org/files/ENCFF419RSJ/
- (optional) miniconda

Dependencies for STARRPeaker (version tested)

- Python 2.7 (v2.7.15)
- pysam (v0.15.3)
- pybedtools (v0.8.1)
- pyBigWig (v0.3.13)
- numpy (v1.15.4)
- scipy (v1.2.0)
- pandas (v0.24.1)
- statsmodels (v0.10.1, use v0.10.2 or earlier, new function statsmodels/tools/validation/validation.py introduced in v0.11.0 may introduce error in Python 2)
- scikit-learn (v0.20.3)

Installation Example

```
### Preferably, create a conda environment with Python 2.7
conda create -n starrpeaker python=2.7 pybedtools
conda activate starrpeaker
pip install git+https://github.com/gersteinlab/starrpeaker
starrpeaker -h
```

Processing Summary

STARR-seq peaks were called using STARRPeaker v1.0 package with default parameters. In addition to genomic input, three extra covariate tracks were utilized. They are namely GC-content, mappability, and folding energy prediction.

Code Example

```
PREFIX={prefix}
CHROMSIZE={path to chrom.size}
BLACKLIST={path to blacklist bed file i.e., refer ENCFF419RSJ}
INPUT={path to (merged) input bam}
OUTPUT={path to (merged) output bam}

python starrpeaker.py --prefix $PREFIX --chromsize $CHROMSIZE --length 500 --step 100
--blacklist $BLACKLIST --cov cov1.bw cov2.bw cov3.bw --input $INPUT --output $OUTPUT
--threshold 0.05
```

Outputs File(s)

```
{prefix}.bin.bed: Genomic bin BED file
{prefix}.bam.bct: Alignment counts in BST format (1st col: input, 2nd col:
output, 3rd col: normalized input)
{prefix}.cov.tsv: Covariate matrix in TSV format
{prefix}.input.bw: Input fragment coverage in bigWig format
{prefix}.output.bw: Output fragment coverage in bigWig format
{prefix}.peak.bed: Initial peak calls (before centering and merging)
{prefix}.peak.final.bed: Final peak calls
{prefix}.peak.pval.bw: P-value track in bigWig format (-log10)
{prefix}.peak.qval.bw: Q-value track in bigWig format (-log10)
```

Final Peak Call Format (BED6+4)

Column 1: Chromosome
Column 2: Start position
Column 3: End position
Column 4: Name (peak rank based on score, 1 being the highest rank)
Column 5: Score (integer value of "100 * fold change", maxed at 1000 per BED format specification)
Column 6: Strand
Column 7: Fold change (output/normalized-input)
Column 8: Output fragment coverage
Column 9: -log10 of P-value
Column 10: -log10 of Q-value (Benjamini-Hochberg False Discovery Rate, FDR)

BED format specification: https://genome.ucsc.edu/FAQ/FAQformat.html#format1

3. QC: Aggregate histone enrichments around peaks

Prerequisites

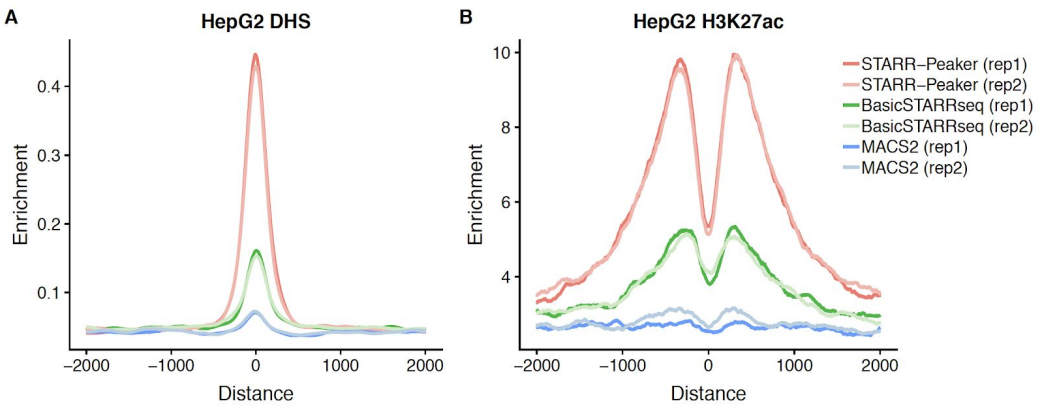
- `bwtool v1.0`
- `R: ggplot2`

- `R: cowplot`

Code Example

```
PREFIX={prefix}
PEAK={path to peak file}

DHS={path to DNase-seq bigwig file}
H3K27ac={path to H3K27ac bigwig file}

awk -F'\t' 'BEGIN {OFS="\t"}{print $1,$2,$3,$1":"$2+1"-"$3}' $PEAK > "$PREFIX".tmp.bed

bwtool agg 2000:2000 "$PREFIX".tmp.bed $DHS "$PREFIX"_agg_DHS.tsv
bwtool agg 2000:2000 "$PREFIX".tmp.bed $H3K27ac "$PREFIX"_agg_H3K27ac.tsv

rm "$PREFIX".tmp.bed
```

Plot example (plotted using R ggplot2 and cowplot)



4. QC: Concordance between replicates

Processing Summary

Based on STARR-seq alignments, we can use deepTools to quickly calculate concordance between replicates

Prerequisites

- `deepTools v3.1.0`

Code Example

```
PREFIX={prefix}

multiBamSummary bins --bamfiles {path to replicate 1 bam file} {path to replicate 1 bam
file} -o "$PREFIX"_multiBamSummary.npz

plotCorrelation --corData "$PREFIX"_multiBamSummary.npz --corMethod pearson --whatToPlot
scatterplot --plotFile "$PREFIX"_scatterplot.pdf --skipZeros --labels rep1 rep2
--plotTitle "$PREFIX" --removeOutliers --log1p --plotHeight 30 --plotWidth 30
```

Plot Example