GradingPoolSeq——pipeline

Prerequisites

•Install the following programs:

1. Perl (v5.22.1)
2. Perl module (Statistics::Distributions)
3. R(v3.2.5)
4. R package (ggplot2)
5. BWA(v0.71)
6. GATK(v4.1.0.0)
7. Picard(v1.119)

Based on linux system.
Installation of perl: https://www.perl.org/get.html
Installation of R: https://www.r-project.org/
Installation of BWA: https://github.com/lh3/bwa
Installation of GATK: https://software.broadinstitute.org/gatk/
Installation of Picard: https://broadinstitute.github.io/picard/
Expected time of installation would be within two hours.
None required non-standard hardware.

Demo :

Demo is in demo file, including instructions and expected outputs. If you already have vcf file, then expected running time is within 20 minutes.

Requirement:

1. Reference genome.
2. Fastq files of each Bulk.

OR

VCF file obtained by SNP calling(GATK “best practices”).

Pipeline:

Step 1: Install this pipeline framework to your system.

```
ex) $ cd /GradingPool                    #Working place
    $ cp GradingPool_framework.tar.gz
    $ tar zxvf GradingPool_framework.tar.gz
```

Step 2: Prepare each bulks and reference genome sequencing data and perform alignment as GATK “best practice” to obtain the results of variant calling. If you already have vcf file, you can skip this step.

Suppose we have six bulks here, four for F_2 generations (bulk1-bulk4) and two for parental lines (bulk5 and bulk6).

1) Build index of reference genome.

```
bwa index ref.fa
```

2) Align samples' fastq files to reference genome.

```
bwa mem -M -R "@RG\tID:Pool1\tSM:Pool1" ref.fa Pool1.fastq.gz > Pool1.sam
bwa mem -M -R "@RG\tID:Pool2\tSM:Pool2" ref.fa Pool2.fastq.gz > Pool2.sam
bwa mem -M -R "@RG\tID:Pool3\tSM:Pool3" ref.fa Pool2.fastq.gz > Pool3.sam
bwa mem -M -R "@RG\tID:Pool4\tSM:Pool4" ref.fa Pool3.fastq.gz > Pool4.sam

bwa mem -M -R "@RG\tID:Pa1\tSM:Pa1" ref.fa Parent1.fastq.gz > Parent 1.sam
bwa mem -M -R "@RG\tID:Pa2\tSM:Pa2" ref.fa Parent2.fastq.gz > Parent 2.sam
```

F_2 generations' files

Parental files

3) Covert sam file to bam file

```
samtools sort -o Pool1.bam Pool1.sam
samtools sort -o Pool2.bam Pool2.sam
samtools sort -o Pool3.bam Pool3.sam
samtools sort -o Pool4.bam Pool4.sam

samtools sort -o Parent1.bam Parent1.sam
samtools sort -o Parent2.bam Parent2.sam
```

F_2 generations' files

Parental files

4) Duplicates Marking

```
java -jar picard.jar MarkDuplicates INPUT=Pool1.bam OUTPUT= Pool1.dedup.bam
METRICS_FILE= Pool1.metrics &
java -jar picard.jar MarkDuplicates INPUT=Pool2.bam OUTPUT= Pool2.dedup.bam
METRICS_FILE= Pool2.metrics &
java -jar picard.jar MarkDuplicates INPUT=Pool3.bam OUTPUT= Pool3.dedup.bam
METRICS_FILE= Pool3.metrics &
java -jar picard.jar MarkDuplicates INPUT=Pool4.bam OUTPUT= Pool4.dedup.bam
METRICS_FILE= Pool4.metrics &

java -jar picard.jar MarkDuplicates INPUT=Parent1.bam OUTPUT=
Parent1.dedup.bam METRICS_FILE= Parent1.metrics &
java -jar picard.jar MarkDuplicates INPUT=Parent2.bam OUTPUT=
Parent2.dedup.bam METRICS_FILE= Parent2.metrics &
```

F_2 generations' files

Parental files

5) Index bam files

```
samtools index Pool1.dedup.bam
samtools index Pool2.dedup.bam
samtools index Pool3.dedup.bam
samtools index Pool4.dedup.bam

samtools index Parent1.dedup.bam
samtools index Parent2.dedup.bam
```

F_2 generations' files

Parental files

6) SNP calling

```
java -jar gatk HaplotypeCaller -R ref.fa -I Pool1.dedup.bam -I Pool2.dedup.bam -I
Pool3.dedup.bam -I Pool4.dedup.bam -I Parent1.dedup.bam -I Parent2.dedup.bam
-O Results.vcf
```

Step 3: Undertake filtering process

Suppose we have six bulks here, four of F_2 generations (pool1-pool4) and two of parental lines (pool5 and pool6). If you want to investigate the results of the combination of two or three pools, you can change the parameter which I point out specifically.

1) filter out low-quality variants

```
perl Filter_Quality.pl Results.vcf > filter1.txt 100
#input file: Headingdate .vcf
#output file: filter1.txt
#100: quality threshold we set here as an example.
```

2) select variants with appropriate depth(5%~95%)

```
perl Filter_Depth.pl filter1.txt > filter2.txt 6 0.05 0.95
#input file: filter1.txt
#output file: filter2.txt
#6: total pools (including parents' pools)
#0.05&0.95: depth range from 5% to 95%.
```

3) screen out variants that both parental lines present homogeneous and different genotypes(optional)

```
perl Filter_Parent.pl filter2.txt > filter3.txt 5 6
#input file: filter2.txt
#output file: filter3.txt
#5&6: numerical order of two parents' pool
```

4) filter out of the SNPs for which sequence reads from all pools only showed non-reference bases.

```
perl Filter_SNP.pl filter3.txt > filter4.txt 6 1 2 3 4
#input file: filter3.txt
#output file: filter4.txt
#6: total pools (including parents' pools)
#1,2,3,4: numerical order of second generations ' pool. If your want to find out the result of other combinations, for example, the first pool and the fourth pool, the parameter will be 1 and 4, and so on.(perl Filter_SNP.pl filter3.txt > filter4.txt 6 1 4)
```

Step 4: calculate p-value

```
perl Ridit.pl filter4.txt > pvalue.txt 1 2 3 4
#input file: filter4.txt
#output file: pvalue.txt
#1,2,3,4: numerical order of second generations ' pool. If your want to find out the result of other combinations, for example, the first pool and the fourth pool, the parameter will be 1 and 4, and so on.(perl Ridit.pl filter4.txt > pvalue.txt 1 4)
```

Step 5: denoise strategy

```
perl denoise.pl pvalue.txt > ratio.txt 4 10 12
#input file: pvalue.txt
#output file: ratio.txt
#2: set 4(X100kb) as a defined genomic interval (400kb in each chromosome window)
#10: threshold for high significant variants, generally we set 10.
#12: the number of chromosomes
```

Step 6: graph analysis

First install packages in R:

1) generate P-value plot

```
Rscript pos-pvalue.R
#input file: pvalue.txt
#output file: pvalue.png
```

2) generate ratio plot

```
Rscript pos-ratio.R
#input file: ratio.txt
#output file: ratio.png
```

Step 7: identify peak interval

```
perl p-max.pl ratio.txt 1
#input file: ratio.txt
#1: chromosome
#output: peak point and peak interval
```