SNP calling pipeline for Pool-seq

James Reeve (Yeaman Lab)
james.reeve1@ucalgary.ca
16th August 2018
Pipeline Steps

Use GATK’s best practices for “Germline SNPs + Indels”

My pipeline is modified for pool-seq

GATK’s Best Practices - 26th July 2018
https://software.broadinstitute.org/gatk/best-practices/workflow?id=11145
Pipeline Steps

Use GATK’s best practices for “Germline SNPs + Indels”

My pipeline is modified for pool-seq
Pipeline Steps - Major steps

1. Read Trimming
2. Read Mapping
3. Duplicate Removal
4. SNP Calling
5. SNP Filtering
6. Recalibration
7. SNP Calling
8. SNP Filtering
**Read Trimming**

**Objective:**
Remove adapters and low quality read ends

**Software:**
Trimmomatic (v0.36)

```
$ java -jar trimmomatic-0.36.jar PE
    -threads 12 -phred33 \
    [Filename forward read].fastq  \
    [Filename reverse read].fastq  \
    -baseout [Filename output] \ 
    ILLUMINACLIP:[adapter name] \ 
    MAXINFO:150:0.5 \ 
    TRAILING:20 \ 
    LEADING:20 \ 
    MINLEN:120
```
Objective:
Align reads to reference genome

Software:
BWA (v0.7.12)

# Create Reference Genome Index
$ bwa index -a bwtsw [Reference genome].fasta > [Filename output]

# Running BWA (MEM algorithm)
$ bwa mem -P -M \n    [Reference Genome].fasta \n    [Filename forward read] \n    [Filename reverse read] > \n    [Filename output]

- **a** = indexing algorithm (bwtsw = Smith-Waterman)
- **P** = paired-end mode (requires 2 input files)
- **M** = mark shorter splits (needed for Picard)

Green = input
Red = output
Blue = argument
Objectives:
● Compress file
● Sort reads by position
● Add read group info

Software:
Samtools (v1.3.1)
Picard (v2.18.11?)

# Compress SAM to BAM
$ samtools view -b [filename].sam > [filename].bam

# Sorting and adding read group information
$ java -jar picard.jar AddOrReplaceReadGroups \
   I=[filename input] \ 
   O=[filename output] \ 
   SORT_ORDER=coordinate \ 
   RGID=E00434:123 \ 
   RGLB=MPS12343188-F07 \ 
   RGPL=illumina \ 
   RGPU=HI.4698.008 \ 
   RGSM=TuAK \ 
   RGCN=Genome_Quebec \ 
   RGDS=Pooled_sample_44_fish \ 
   RGDT=2018-06-04 \ 
   RGPI=150

Green = input
Red = output
Blue = argument
Objectives:
- Compress file
- Sort reads by position
- Add read group info

Software:
- Samtools (v1.3.1)
- Picard (v2.18.11?)

# Compress SAM to BAM
$ samtools view -b [filename].sam > [filename].bam

# Sorting and adding read group information
$ java -jar picard.jar AddOrReplaceReadGroups \\
  I=[filename input] \\
  O=[filename output] \\
  SORT_ORDER=coordinate \\
  RGLB=MPS12343188-F07 \\
  RGPL=illumina \\
  RGPU=HI.4698.008 \\
  RGSM=TuAK \\
  RGID=E00434:123 \\
  RGCN=Genome_Quebec \\
  RGDS=Pooled_sample_44_fish \\
  RGDT=2018-06-04 \\
  RGPI=150

Green = input
Red = output
Blue = argument
Objectives:
- Compress file
- Sort reads by position
- Add read group info

Software:
- Samtools (v1.3.1)
- Picard (v2.18.1?)

# Compress SAM to BAM
$ samtools view -b [filename].sam > [filename].bam

# Sorting and adding read group information
$ java -jar picard.jar AddOrReplaceReadGroups \
  I=[filename input] \
  O=[filename output] \
  SORT_ORDER=coordinate \
  RGID=E00434:123 \
  RGLB=MPS12343188-F07 \
  RGPL=illumina \
  RGPU=HI.4698.008 \
  RGSM=TuAK \
  RGCN=Genome_Quebec \
  RGDS=Pooled_sample_44_fish \
  RGDT=2018-06-04 \
  RGPI=150
Objective:
Remove duplicated reads

Software:
Picard (v2.18.11?)

Duplicate Removal

$ java -Xmx32G -Djava.io.tmpdir=[temp directory] \
-jar picard.jar MarkDuplicates \
I=[Filename input] \
O=[Filename output] \
M=[Duplicate metrics file] \
OPTICAL_DUPLICATE_PIXEL_DISTANCE=2500 \
CREATE_INDEX=true

Green = input
Red = output
Blue = argument
Indel realignment is NOT needed in GATK4
Objective:
Detect variant sites in the dataset

Software:
Samtools (v1.3.1)
Picard (v2.18.11?)
GATK (v4.0.4.0)

# Create a dictionary of the reference genome
$ java -Xmx32G -Djava.io.tmpdir=\[temp directory\] \
   -jar picard.jar CreateSequenceDictionary \
   R= [Reference genome] \
   O= [output].dict

# Index reference genome
$ samtools faidx [Reference genome].fasta

# Index input file
$ samtools index [Filename input].bam

Green = input
Red = output
Blue = argument
Objective: Detect variant sites in the dataset

Software: Samtools (v1.3.1) Picard (v2.18.11?) GATK (v4.0.4.0)

$ java -Xmx3g -jar gatk.jar \
   HaplotypeCaller \
   -R [Reference genome].fasta \
   -I [Filename input].bam \
   --sample-ploidy [2 x pool size] \
   --max-genotype-count \[F(P,A)\] \
   -O [Filename output].vcf
Objective:
Detect variant sites in the dataset

Software:
Samtools (v1.3.1)
Picard (v2.18.11?)
GATK (v4.0.4.0)

Max Genotype Count:
- This parameter must be set to allow multi-allelic SNP calls
- Value is maximum number of unique genotypes \( F(P,A) \)

\[
F(P, A) = \binom{P + A - 1}{A - 1} = \frac{(P + A - 1)!}{P! (A - 1)!}
\]

Where:
\( P = \text{Ploidy} \)
\( A = \text{Allele count} \)

- For triallelic sites \( (A = 3) \) it simplifies to;

\[
F(P, 3) = \frac{(P + 2)(P + 1)}{2}
\]
Objective:
Create a subset of only SNPs

Software:
  - Samtools (v1.3.1)
  - Picard (v2.18.11?)
  - GATK (v4.0.4.0)

```bash
$ java -Xmx3g -jar gatk.jar
   SelectVariants
   -R [Reference genome].fasta
   -V [Filename Input].vcf
   -O [Filename output].vcf
   --select-type-to-include SNP
```
Objective:
Remove low quality SNPs*.

Software:
GATK (v4.0.4.0)

*There is no rule of thumb. The data has to be viewed to determine the quality thresholds.

```sh
# Create a table of quality scores
$ java -Xmx3g -jar gatk.jar \
    VariantsToTable \
    -R [Reference genome].fasta \
    -V [Filename input].vcf \
    -O [Filename output] \
    -F CHROM \
    -F POS \
    -F QUAL \
    -F DP \
    -F QD \
    -F MQ \
    -F FS \
    -F SOR \
    -F MQRankSum
```

Green = input
Red = output
Blue = argument
These plots show the density of each quality score across the full VCF file. *Plots generated by the R package ggplot2*
Objective:
Remove low quality SNPs*.

Software:
GATK (v4.0.4.0)

*There is no rule of thumb. The data has to be viewed to determine the quality thresholds.

```
# Apply filters
$ java -Xmx3g -jar gatk.jar \\
   VariantFiltration \\
   -R [Reference genome].fasta \\
   -V [Filename input].vcf \\
   -O [Filename output].vcf \\
   -filter "QUAL < 20" --filter-name Low_Qual \\
   -filter "DP < 88" --filter-name Low_Cov \\
   -filter "QD < 20.0 || MQ < 50.0 || FS > 1.0 || \ 
    SOR > 1.5 || MQRendSum < -1.0" \\
   --filter-name Secondary_filter
```

Green = input
Red = output
Blue = argument
Objective:
Remove low quality SNPs*.

Software:
GATK (v4.0.4.0)

*There is no rule of thumb. The data has to be viewed to determine the quality thresholds.

# Removing reads that fail the filter
$ java -Xmx3g -jar gatk.jar \
  SelectVariants \
  -R [Reference genome].fasta \
  -V [Filename input].vcf \
  -O [Filename output].vcf \
  --exclude-filtered
Wait... Aren’t we going backwards?

Yes. GATK’s guidelines recalibrate the base quality scores after removing duplicates. However, you need a known set of ‘true SNPs’ to do this.

For non-model organisms you will need to create this ‘true SNPs’ dataset from your own data (bootstrapping).
## Recalibration

### Objective:
Recalibrate the quality scores of each base in read files.

### Software:
GATK (v4.0.4.0)

```
# Create recalibration table
$ java -jar gatk.jar \
  BaseRecalibrator \
  -R [Reference genome].fasta \
  -I [Filename input].bam \
  -O [Filename output] \
  --known-sites [true SNPs].vcf
```
Objective:
Recalibrate the quality scores of each base in read files.

Software:
GATK (v4.0.4.0)

Note:
Some older best practices use PrintReads. This program is gone from GATK4

```
# Base quality score recalibration
$ java -jar gatk.jar \
   ApplyBQSR \
   -R [Reference genome].fasta \
   -I [Filename input].bam \
   --bqsr-recal-file [Recalibration table] \
   -O [Filename output]
```
Finding SNPs in the genome.

For real this time.
SNP calling is repeated using the same code.

SNP filtering is done with less strict filters, as wider genome representation is more important than SNP quality.
Pipeline Overview

Reads

- Read Trimming
  - Trimmmomatic

Read Mapping
- BWA-MEM

Sort Reads
- Picard-AddOrReplaceReadGroups

Duplicate Removal
- Picard-MarkDuplicates

SNP Calling
- GATK-HaplotpyeCaller

BQSR
1. GATK-BaseRecalibrator
2. GATK-ApplyBQSR

SNP Filtering
- GATK-VariantFiltration

SNPs