Variant Calling Pipeline for SRA Accession SRR33784444

This document provides a step-by-step guide to the viral variant calling pipeline for SRA accession **SRR33784444**. Each section lists the tools used, the purpose of the step, and the exact commands needed to ensure reproducibility.

---

1. Initialization and Logging

Tools:

- **bash**

Purpose: Set strict error handling and initialize a log file with timestamped entries for monitoring.

```
#!/bin/bash
set -e

LOGFILE="variant_calling.log"
exec > >(tee -a "$LOGFILE") 2>&1

# Define logging function
gen_log() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $1"
}

gen_log "Starting viral variant calling pipeline with BWA-MEM..."
```

---

2. Reference Genome Download

Tools:

- **wget**

Purpose: Fetch the reference genome FASTA from NCBI.

```
gen_log "Downloading reference genome..."
wget --quiet --show-progress \
     -O reference.fa \
     "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nuccore&id=NC_001802.1&re
gen_log "Reference genome downloaded."
```

---

3. Data Acquisition and FASTQ Conversion

Tools:

- **SRA Toolkit** (`prefetch`, `fastq-dump`)

Purpose: Download SRA data and convert to paired-end FASTQ.

```
gen_log "Prefetching SRR33784444..."
prefetch SRR33784444 --progress
gen_log "Prefetch complete."

gen_log "Running fastq-dump to split FASTQ files..."
fastq-dump --split-files SRR33784444/
gen_log "fastq-dump complete."
```

---

4. Quality Control of Raw Reads

Tools:

- **FastQC**

Purpose: Generate quality reports for raw FASTQ files.

```
gen_log "Running FastQC on raw FASTQ files..."
fastqc SRR33784444_1.fastq SRR33784444_2.fastq
gen_log "FastQC on raw FASTQ files complete."
```

---

5. Read Trimming

Tools:

- **Trimmomatic**

Purpose: Remove adapters and low-quality regions to improve alignment.

```
gen_log "Creating directory for trimmed reads and running Trimmomatic..."
mkdir -p Trimmed && cd Trimmed
cp ../reference.fa .

trimmomatic PE ../SRR33784444_1.fastq ../SRR33784444_2.fastq \
    SRR33784444_1_paired.fastq SRR33784444_1_unpaired.fastq \
    SRR33784444_2_paired.fastq SRR33784444_2_unpaired.fastq \
    ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 \
    SLIDINGWINDOW:4:20 MINLEN:50
gen_log "Trimmomatic processing complete."

# Back to root
```

```
gen_log "Running FastQC on trimmed FASTQ files..."
fastqc SRR33784444_1_paired.fastq SRR33784444_2_paired.fastq SRR33784444_1_unpaired.fastq SR
gen_log "FastQC on trimmed FASTQ files complete."
cd ..
mv Trimmed/alignment_bwa.sam . || true
```

---

6. Reference Indexing

Tools:

- **BWA**

Purpose: Create index files for the reference FASTA to enable alignment.

```
gen_log "Building BWA index for reference genome..."
bwa index reference.fa
gen_log "BWA index building complete."
```

---

7. Alignment with BWA-MEM

Tools:

- **BWA-MEM**

Purpose: Align trimmed paired-end reads to the reference, embedding read-group metadata.

```
gen_log "Running BWA-MEM alignment..."
bwa mem -t 4 -M \
    -R "@RG\tID:Sample\tSM:Sample\tPL:ILLUMINA\tLB:lib1\tPU:unit1" \
    reference.fa \
    SRR33784444_1_paired.fastq \
    SRR33784444_2_paired.fastq \
    > alignment_bwa.sam
gen_log "BWA-MEM alignment complete."
```

---

8. Alignment Statistics

Tools:

- **SAMtools**

Purpose: Calculate and save basic alignment metrics.

```
gen_log "Checking alignment statistics..."
samtools flagstat alignment_bwa.sam | tee alignment_stats.txt
gen_log "Alignment statistics saved to alignment_stats.txt"
```

---

9. SAM→BAM Conversion, Sorting, and Indexing

Tools:

- **SAMtools**

Purpose: Convert aligned SAM to sorted BAM and create an index for efficient access.

```
gen_log "Converting SAM to BAM..."
samtools view -bS alignment_bwa.sam > temp.bam

gen_log "Sorting BAM file..."
samtools sort temp.bam -o sorted.bam
rm temp.bam

gen_log "Indexing sorted BAM file..."
samtools index sorted.bam
```

---

10. Read-Group Verification

Tools:

- **SAMtools**

Purpose: Check that read-group headers are present in the BAM for GATK compatibility.

```
gen_log "Verifying read groups in sorted BAM file..."
samtools view -H sorted.bam | grep "@RG" || gen_log "WARNING: No read groups found"
```

---

11. Coverage Calculation

Tools:

- **SAMtools**, **awk**

Purpose: Compute per-base coverage and summarize the average.

```
gen_log "Calculating coverage statistics..."
samtools depth sorted.bam > coverage.txt
```

```
awk '{sum+=$3; count++} END {print "Average coverage:", sum/count}' coverage.txt | tee cover
gen_log "Coverage statistics saved."
```

---

12. Duplicate Marking

Tools:

- **GATK MarkDuplicates**

Purpose: Mark PCR duplicates to avoid bias in downstream variant calls.

```
gen_log "Marking duplicates with GATK..."
gatk MarkDuplicates \
    -I sorted.bam \
    -O temp_unique_reads.bam \
    -M metrics.txt \
    --CREATE_INDEX true
gen_log "Duplicates marked."
```

---

13. Read-Group Correction

Tools:

- **GATK AddOrReplaceReadGroups**

Purpose: Explicitly assign or correct read-group tags for GATK tools.

```
gen_log "Adding/fixing read groups for GATK compatibility..."
gatk AddOrReplaceReadGroups \
    -I temp_unique_reads.bam \
    -O unique_reads.bam \
    -RGID Sample -RGSM Sample \
    -RGPL ILLUMINA -RGLB lib1 -RGPU unit1 \
    --CREATE_INDEX true
gen_log "Read groups added/fixed."
rm temp_unique_reads.bam temp_unique_reads.bai
```

---

14. Reference Prep for GATK

Tools:

- **GATK CreateSequenceDictionary**, **SAMtools faidx**

Purpose: Generate a sequence dictionary and FASTA index required by GATK.

```
gen_log "Creating sequence dictionary for reference..."
gatk CreateSequenceDictionary -R reference.fa -O reference.dict
gen_log "Indexing reference with samtools..."
samtools faidx reference.fa
```

---

15. Variant Calling

Tools:

- **GATK HaplotypeCaller**

Purpose: Identify SNPs and INDELs using viral-optimized parameters.

```
gen_log "Calling variants with GATK HaplotypeCaller..."
gatk HaplotypeCaller \
    -R reference.fa \
    -I unique_reads.bam \
    -O output.vcf \
    --native-pair-hmm-threads 4 \
    --standard-min-confidence-threshold-for-calling 10 \
    --minimum-mapping-quality 20 \
    --max-reads-per-alignment-start 0
gen_log "Variant calling complete."
```

---

16. Variant Filtration

Tools:

- **GATK VariantFiltration**

Purpose: Filter low-confidence variants based on quality metrics.

```
gen_log "Filtering variants with GATK VariantFiltration..."
gatk VariantFiltration \
    -R reference.fa \
    -V output.vcf \
    -O filtered_output.vcf \
    --filter-expression "QD < 2.0" --filter-name "LowQD" \
    --filter-expression "FS > 60.0" --filter-name "HighFS" \
    --filter-expression "MQ < 30.0" --filter-name "LowMQ" \
    --filter-expression "DP < 10" --filter-name "LowDepth" \
    --filter-expression "QUAL < 30.0" --filter-name "LowQual"
gen_log "Variant filtration complete."
```

---

17. High-Confidence Variant Selection

Tools:

- **GATK SelectVariants**

Purpose: Isolate variants that pass all filters for downstream analysis.

```
gen_log "Creating high-confidence variant set..."
gatk SelectVariants \
    -R reference.fa \
    -V filtered_output.vcf \
    -O high_confidence_variants.vcf \
    --exclude-filtered true
gen_log "High-confidence variants selected."
```

---

18. Statistics and Consensus Generation

Tools:

- **bcftools**

Purpose: Produce variant summary stats and generate a consensus sequence.

```
gen_log "Generating variant summary statistics..."
bcftools stats filtered_output.vcf > variant_stats.txt
gen_log "Variant statistics saved to variant_stats.txt"

# Consensus sequence
bcftools consensus -f reference.fa filtered_output.vcf > consensus_sequence.fa
sed -i 's/>.*/>Patient_Sample_Consensus/' consensus_sequence.fa
gen_log "Consensus sequence created as consensus_sequence.fa"
```

---

19. Coverage Plot Data

Tools:

- **SAMtools**, **awk**

Purpose: Extract coverage profile for plotting.

```
gen_log "Generating coverage data for visualization..."
samtools depth sorted.bam | awk '{print $2"\t"$3}' > coverage_plot.txt
gen_log "Coverage plot data saved to coverage_plot.txt"
```

---

20. Final Summary Report

Tools:

- **bash**, **samtools**, **grep**

Purpose: Aggregate key metrics and file outputs into a summary text file.

```
gen_log "Generating final summary..."
echo "=== Viral Variant Calling Pipeline Summary ===" > pipeline_summary.txt
```