Walk through

Step 1	Alignment – Map to Reference
Tool	BWA MEM
Input	.fastq files, reference genome
Output	aligned_reads.sam* *Intermediary file, removed from final output
Notes	Need to provide the -M flag to BWA, this tells it to consider split reads as secondary, need this for GATK variant calling/Picard support. Alternate alignment tools: Bowtie2, Novoalign Readgroup info is provided with the -R flag. This information is key for downstream GATK functionality. GATK will not work without a read group tag.
Command	`bwa mem -M -R '@RG\tID:sample_1\tLB:sample_1\tPL:ILLUMINA\tPM:HISEQ\tSM:sample_1' ref input_1 input_2 > aligned_reads.sam`
Step 2	Sort SAM file by coordinate, convert to BAM
Tool	Picard Tools
Input	aligned_reads.sam
Output	sorted_reads.bam* *Intermediary file, removed from final output
Command	`java -jar picard.jar SortSam INPUT=aligned_reads.sam OUTPUT=sorted_reads.bam SORT_ORDER=coordinate`
Step 3	Collect Alignment & Insert Size Metrics
Tool	Picard Tools, R, Samtools
Input	sorted_reads.bam, reference genome
Output	alignment_metrics.txt, insert_metrics.txt, insert_size_histogram.pdf, depth_out.txt

Command	`java -jar picard.jar CollectAlignmentSummaryMetrics R=ref I=sorted_reads.bam O=alignment_metrics.txt`
	`java -jar picard.jar CollectInsertSizeMetrics INPUT=sorted_reads.bam OUTPUT=insert_metrics.txt HISTOGRAM_FILE=insert_size_histogram.pdf`
	`samtools depth -a sorted_reads.bam > depth_out.txt`

Step 4	Mark Duplicates
Tool	Picard Tools
Input	sorted_reads.bam

Output	dedup_reads.bam* metrics.txt *Intermediary file, removed from final output
Command	`java -jar picard.jar MarkDuplicates INPUT=sorted_reads.bam OUTPUT=dedup_reads.bam METRICS_FILE=metrics.txt`

Step 5	Build BAM Index
Tool	Picard Tools
Input	dedup_reads.bam
Output	dedup_reads.bai* *Intermediary file, removed from final output
Command	`java -jar picard.jar BuildBamIndex INPUT=dedup_reads.bam`

Step 6	Create Realignment Targets
Tool	GATK
Input	dedup_reads.bam, reference genome
Output	realignment_targets.list
Notes	This is the first step in a two-step process of realigning around indels
Commad	`java -jar GenomeAnalysisTK.jar -T RealignerTargetCreator -R ref -I dedup_reads.bam -o realignment_targets.list`

Step 7	Realign Indels
Tool	GATK
Input	dedup_reads.bam, realignment_targets.list, reference genome
Output	realigned_reads.bam
Notes	This step performs the realignment around the indels which were identified in the previous step (the ‘realignment targets’)
Command	`java -jar GenomeAnalysisTK.jar -T IndelRealigner -R ref -I dedup_reads.bam -targetIntervals realignment_targets.list -o realigned_reads.bam`

Step 8	Call Variants
Tool	GATK
Input	realigned_reads.bam, reference genome
Output	raw_variants.vcf
Notes	First round of variant calling. The variants identified in this step will be filtered and provided as input for Base Quality Score Recalibration
Command	`java -jar GenomeAnalysisTK.jar -T HaplotypeCaller -R ref -I realigned_reads.bam -o raw_variants.vcf`

Step 9	Extract SNPs & Indels
Tool	GATK

Input	raw_variants.vcf, reference genome
Output	raw_indels.vcf, raw_snps.vcf
Notes	This step separates SNPs and Indels so they can be processed and used independently

Command	`java -jar GenomeAnalysisTK.jar -T SelectVariants -R ref -V raw_variants.vcf -selectType SNP -o raw_snps.vcf`
	`java -jar GenomeAnalysisTK.jar -T SelectVariants -R ref -V raw_variants.vcf -selectType INDEL -o raw_indels.vcf`

Step 10	Filter SNPs
Tool	GATK
Input	raw_snps.vcf, reference genome
Output	filtered_snps.vcf
Notes	The filtering criteria for SNPs are as follows: QD < 2.0 FS > 60.0 MQ < 40.0 MQRankSum < -12.5 ReadPosRankSum < -8.0 SOR > 4.0 Note: SNPs which are ‘filtered out’ at this step will remain in the filtered_snps.vcf file, however they will be marked as ‘basic_snp_filter’, while SNPs which passed the filter will be marked as ‘PASS’
Command	`java -jar GenomeAnalysisTK.jar -T VariantFiltration -R ref -V raw_snps.vcf --filterExpression 'QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0 || SOR > 4.0' --filterName "basic_snp_filter" -o filtered_snps.vcf`
Step 11	Filter Indels
Tool	GATK
Input	raw_indels.vcf, reference genome
Output	filtered_indels.vcf
Notes	The filtering criteria for SNPs are as follows: QD < 2.0 FS > 200.0 ReadPosRankSum < -20.0 SOR > 10.0 Note: Indelss which are ‘filtered out’ at this step will remain in the filtered_indels.vcf file, however they will be marked as ‘basic_indel_filter’, while Indels which passed the filter will be marked as ‘PASS’
Command	`java -jar GenomeAnalysisTK.jar -T VariantFiltration -R ref -V raw_indels.vcf --filterExpression 'QD < 2.0 || FS >`

	`200.0 || ReadPosRankSum < -20.0 || SOR > 10.0' --filterName "basic_indel_filter" -o filtered_indels.vcf`
Step 12	Base Quality Score Recalibration (BQSR) #1
Tool	GATK
Input	realigned_reads.bam, filtered_snps.vcf, filtered_indels.vcf, reference genome
Output	recal_data.table* *Intermediary file, removed from final output
Notes	BQSR is performed twice. The second pass is optional, but is required to produce a recalibration report.
Command	`java -jar GenomeAnalysisTK.jar -T BaseRecalibrator -R ref -I realigned_reads.bam -knownSites filtered_snps.vcf -knownSites filtered_indels.vcf -o recal_data.table`
Step 13	Base Quality Score Recalibration (BQSR) #2
Tool	GATK
Input	recal_data.table, realigned_reads.bam, filtered_snps.vcf, filtered_indels.vcf, reference genome
Output	post_recal_data.table *Intermediary file, removed from final output
Notes	The second time BQSR is run, it takes the output from the first run (recal_data.table) as input
Command	`java -jar GenomeAnalysisTK.jar -T BaseRecalibrator -R ref -I realigned_reads.bam -knownSites filtered_snps.vcf -knownSites filtered_indels.vcf -BQSR recal_data.table -o post_recal_data.table`
Step 14	Analyze Covariates
Tool	GATK
Input	recal_data.table, post_recal_data.table, reference genome
Output	recalibration_plots.pdf
Notes	This step produces a recalibration report based on the output from the two BQSR runs
Command	`java -jar GenomeAnalysisTK.jar -T AnalyzeCovariates -R ref -before recal_data.table -after post_recal_data.table -plots recalibration_plots.pdf`
Step 15	Apply BQSR
Tool	GATK
Input	recal_data.table,

	realigned_reads.bam, reference genome
Output	recal_reads.bam
Notes	This step applies the recalibration computed in the first BQSR step to the bam file.
Command	`java -jar GenomeAnalysisTK.jar -T PrintReads -R ref -I realigned_reads.bam -BQSR recal_data.table -o recal_reads.bam`
Step 16	Call Variants
Tool	GATK
Input	recal_reads.bam, reference genome
Output	raw_variants_recal.vcf
Notes	Second round of variant calling performed on recalibrated bam
Command	`java -jar GenomeAnalysisTK.jar -T HaplotypeCaller -R ref -I recal_reads.bam -o raw_variants_recal.vcf`
Step 17	Extract SNPs & Indels
Tool	GATK
Input	raw_variants_recal.vcf, reference genome
Output	raw_indels_recal.vcf, raw_snps_recal.vcf
Notes	This step separates SNPs and Indels so they can be processed and analyzed independently

Command	`java -jar GenomeAnalysisTK.jar -T SelectVariants -R ref -V raw_variants_recal.vcf -selectType SNP -o raw_snps_recal.vcf`
	`java -jar GenomeAnalysisTK.jar -T SelectVariants -R ref -V raw_variants_recal.vcf -selectType INDEL -o raw_indels_recal.vcf`

Step 18	Filter SNPs
Tool	GATK
Input	raw_snps_recal.vcf, reference genome
Output	filtered_snps_final.vcf
Notes	The filtering criteria for SNPs are as follows: QD < 2.0 FS > 60.0 MQ < 40.0 MQRankSum < -12.5 ReadPosRankSum < -8.0 SOR > 4.0 Note: SNPs which are ‘filtered out’ at this step will remain in the filtered_snps_final.vcf file, however they will be marked as ‘basic_snp_filter’, while SNPs which passed the filter will be marked as ‘PASS’
Command	`java -jar GenomeAnalysisTK.jar -T VariantFiltration -R`

<table>
<tr><td></td><td>ref -V raw_snps_recal.vcf --filterExpression 'QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0 || SOR > 4.0' --filterName "basic_snp_filter" -o filtered_snps_final.vcf</td></tr>
<tr><td>Step 19</td><td>Filter Indels</td></tr>
<tr><td>Tool</td><td>GATK</td></tr>
<tr><td>Input</td><td>raw_indels_recal.vcf,
reference genome</td></tr>
<tr><td>Output</td><td>filtered_indels_final.vcf</td></tr>
<tr><td>Notes</td><td>The filtering criteria for SNPs are as follows:

QD < 2.0
FS > 200.0
ReadPosRankSum < -20.0
SOR > 10.0

Note: Indelss which are ‘filtered out’ at this step will remain in the filtered_indels_recal.vcf file, however they will be marked as ‘basic_indel_filter’, while Indels which passed the filter will be marked as ‘PASS’</td></tr>
<tr><td>Command</td><td>java -jar GenomeAnalysisTK.jar -T VariantFiltration -R ref -V raw_indels_recal.vcf --filterExpression 'QD < 2.0 || FS > 200.0 || ReadPosRankSum < -20.0 || SOR > 10.0' --filterName "basic_indel_filter" -o filtered_indels_recal.vcf</td></tr>
<tr><td>Step 20</td><td>Annotate SNPs and Predict Effects</td></tr>
<tr><td>Tool</td><td>SnpEff</td></tr>
<tr><td>Input</td><td>filtered_snps_final.vcf</td></tr>
<tr><td>Output</td><td>filtered_snps_final.ann.vcf,
snpeff_summary.html,
snpEff_genes.txt</td></tr>
<tr><td>Command</td><td>java -jar snpEff.jar -v snpeff_db filtered_snps_final.vcf > filtered_snps_final.ann.vcf</td></tr>
<tr><td>Step 21</td><td>Compute Coverage Statistics</td></tr>
<tr><td>Tool</td><td>Bedtools</td></tr>
<tr><td>Input</td><td>recal_reads.bam</td></tr>
<tr><td>Output</td><td>genomecov.bedgraph</td></tr>
<tr><td>Notes</td><td>Load the genomecov.bedgraph file into IGV to view a coverage map at the entire genome or chromosome level</td></tr>
<tr><td>Command</td><td>bedtools genomecov -bga -ibam recal_reads.bam > genomecov.bedgraph</td></tr>
</table>

<table>
<tr><td>Step 22</td><td>Compile Statistics</td></tr>
<tr><td>Tool</td><td>parse_metrics.sh (in house)</td></tr>
<tr><td>Input</td><td>alignment_metrics.txt,
insert_metrics.txt,
raw_snps.vcf,
filtered_snps.vcf,
raw_snps_recal.vcf,</td></tr>
</table>

	filtered_snps_final.vcf, depth_out.txt
Output	report.csv
Notes	A single report file is generated with summary statistics for all libraries processed containing the following pieces of information: ● # of Reads ● # of Aligned Reads ● % Aligned ● # Aligned Bases ● Read Length ● % Paired ● Mean Insert Size ● # SNPs, # Filtered SNPs ● # SNPs after BQSR, # Filtered SNPs after BQSR ● Average Coverage