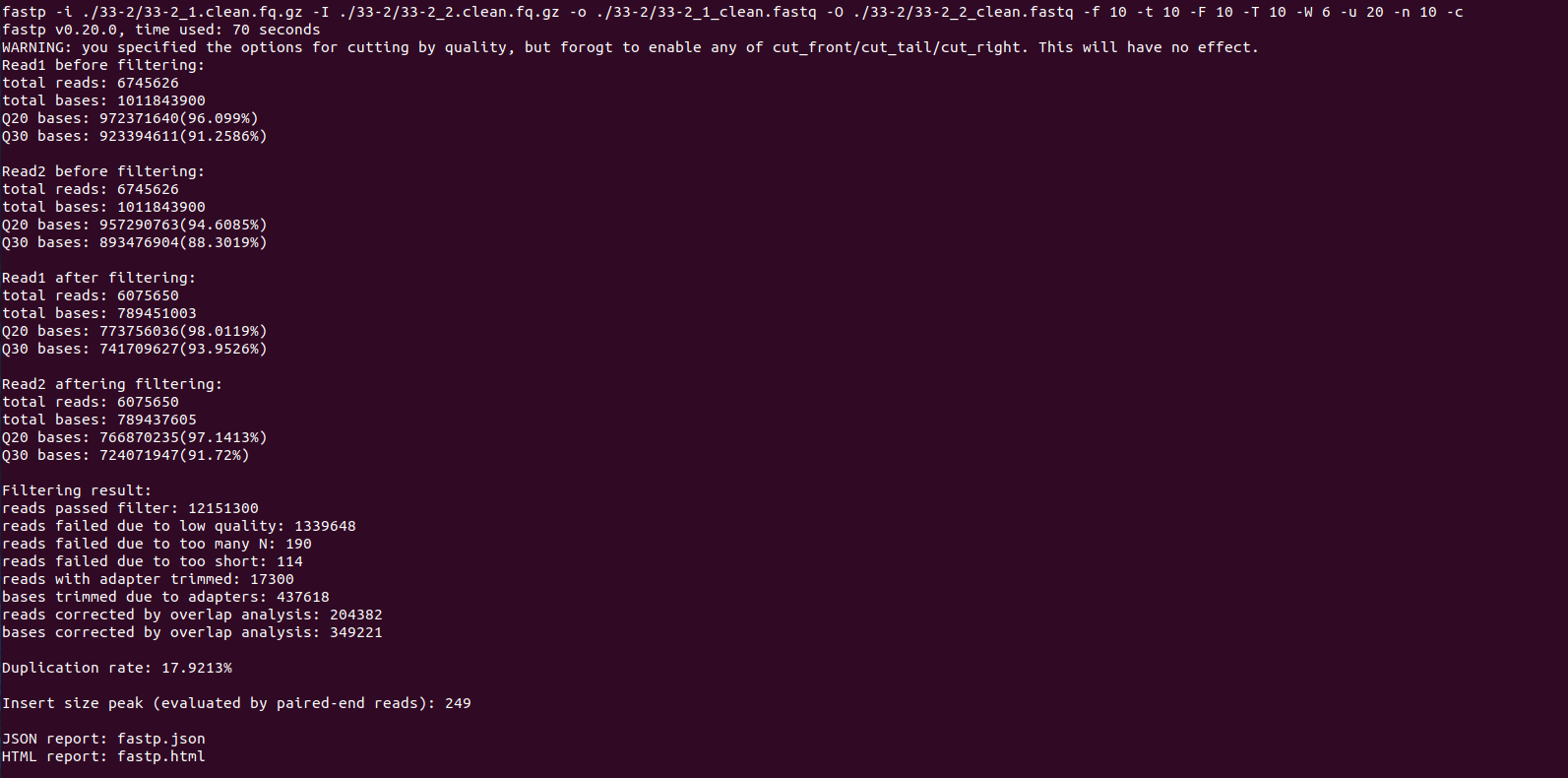
**GWAS analysis for MEM25：**

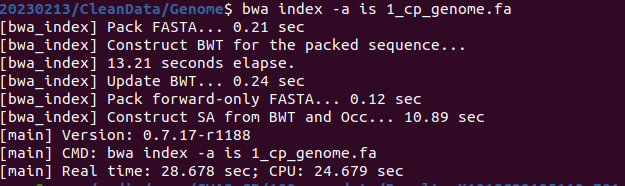
**Attention: It is recommended to use gatk for mutation detection in GWAS**

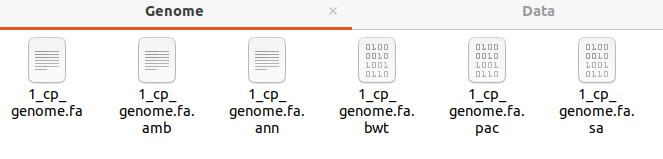
1. After receiving the data, verify the sequencing report; MD5 verifies the integrity and accuracy of file copies to prevent file loss or errors.
2. Re filter the dual ended or single ended data of each sample (different software filtering methods may vary, it is recommended to re filter to understand the data content) (it is recommended to use FastP filtering, fast and integrated)

Use Fastp to perform quality control filtering on the fq.gz files in each folder. Please refer to the Fastp user manual for specific parameters and personal preferences. As shown in the figure below, after strict filtering and cutting, a large proportion of data in cleandata was still cleared, effectively improving the accuracy of sequencing data, as shown in Q30 and Q20.

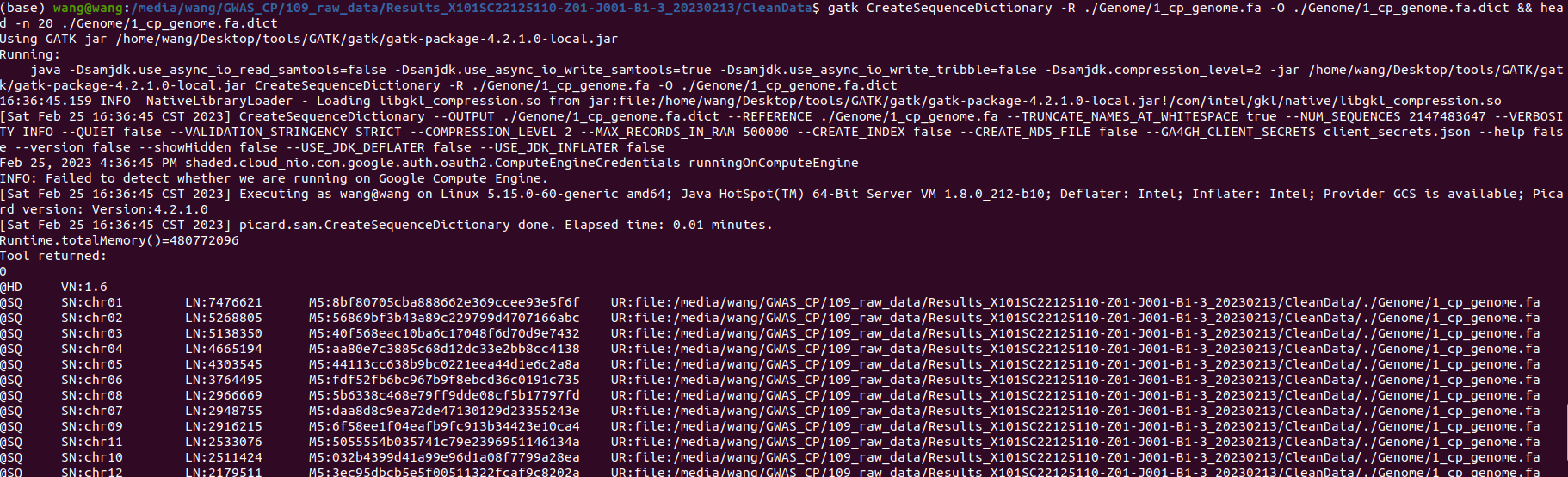


1. Build a genome index using BWA and generate 5 index files outside the genome sequence. It is also necessary to use Samtools Faidx to build indexes.





Additionally, use GATK to construct a genomic dict index file for mutation detection.



1. BWA Mem will align the filtered fastq files from step 2 with the reference genome (note that header information needs to be added for future GATK).

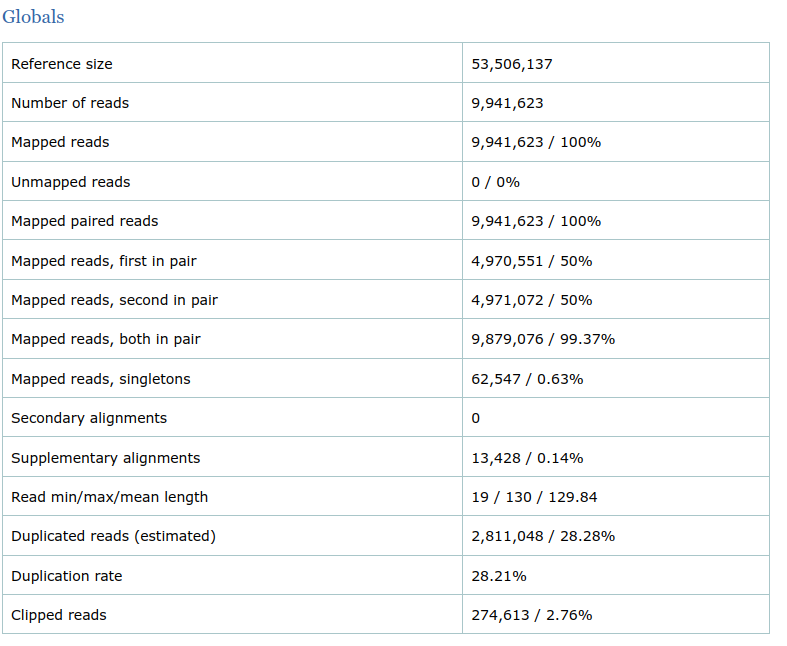
Use bwa mem to align the fastq file to the genome, and add headers and BAM sorting. Sorting can be done using Picard, but it is recommended to use more stable Samtools:

The parameter: -R '@ RG \ ti: {sample} \ tsM: {sample} \ tpL: Illumina' \ # Add header information. -Add a head to R; TID: This is the group ID of the Read Group. If there are parallel samples, they share the same tID; TSM: Sample ID, each sample has its own independent ID for each parallel; tPL: Refers to the sequencing platform used. The purpose is to distinguish data and set grouping information.

Complete code example：bwa mem -t 2 -R '@RG\tID:WT\tSM:WT1\tPL:Illumina' ./Genome/1\_cp\_genome.fa ./WT1/WT1\_1\_clean.fastq ./WT1/WT1\_2\_clean.fastq 2>./WT1/WT1.bwa.log | samtools sort -@ 2 -m 1G -o ./WT1/WT1.sort.bam

After this step is completed, you can add the step, samtools view -h -b -q30 ./WT1/WT1. sort.bam>./WT1/WT1.q30.sort.bam, operation to strictly limit the results, with a base mass of at least 30.

As shown in the following figure, the BAM filtering results are quite good (can be verified using qualimap):

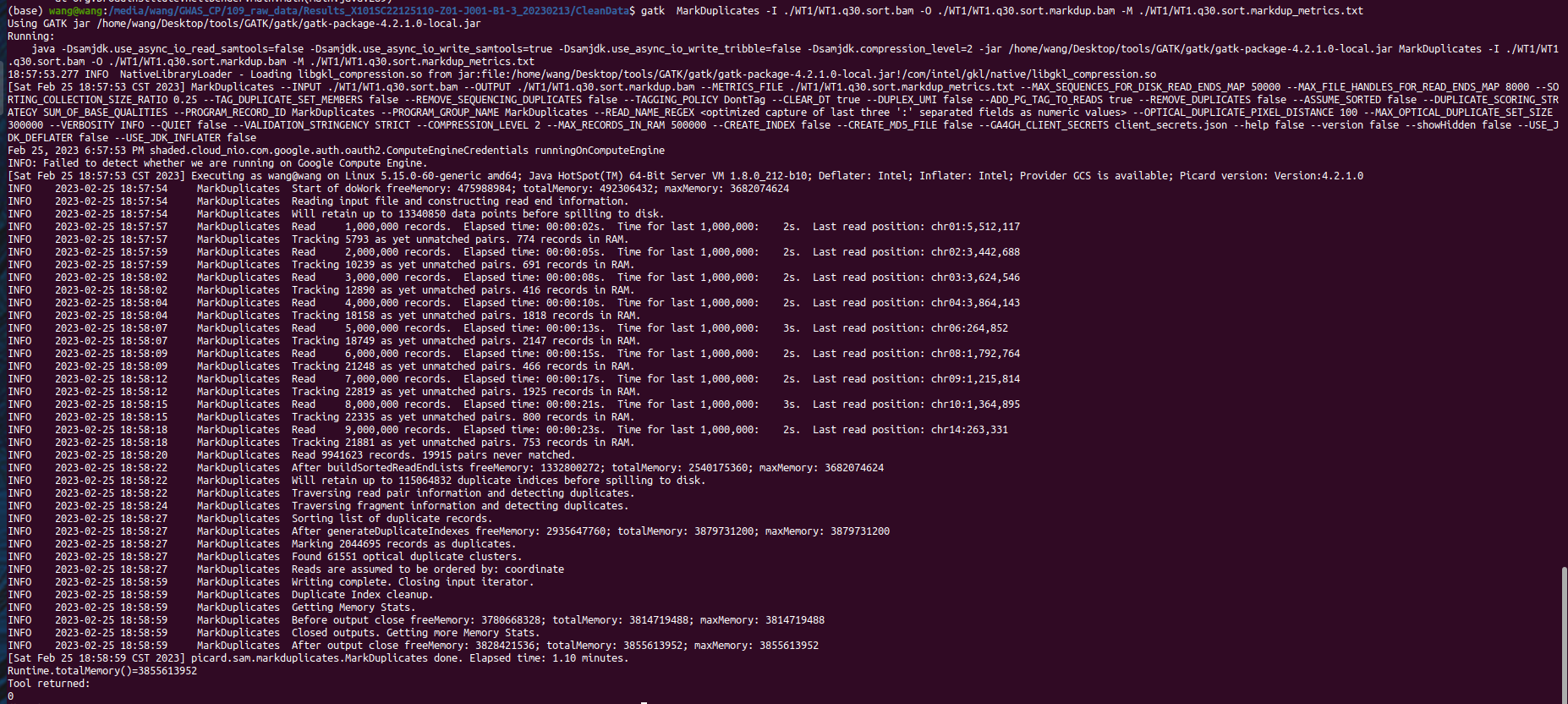


However, the cost is that the sequencing depth has been reduced, from approximately 35X of data to 25X, as shown in the following figure:



1. The process of generating sequencing data involves library amplification and cluster formation, which can easily generate some duplicates (i.e. PCR duplicates and Optical duplicates) that cannot be used as evidence for mutation detection. Using the "MarkDuplicates" feature in the Picard software package to remove duplicates, that is, if multiple Paired Reads have the same chromosome coordinates after alignment, only the Paired Reads with the highest score will be retained. Use code:

gatk MarkDuplicates -I ./WT1/WT1.q30.sort.bam -O ./WT1/WT1.q30.sort.markdup.bam -M ./WT1/WT1.q30.sort.markdup\_metrics.txt



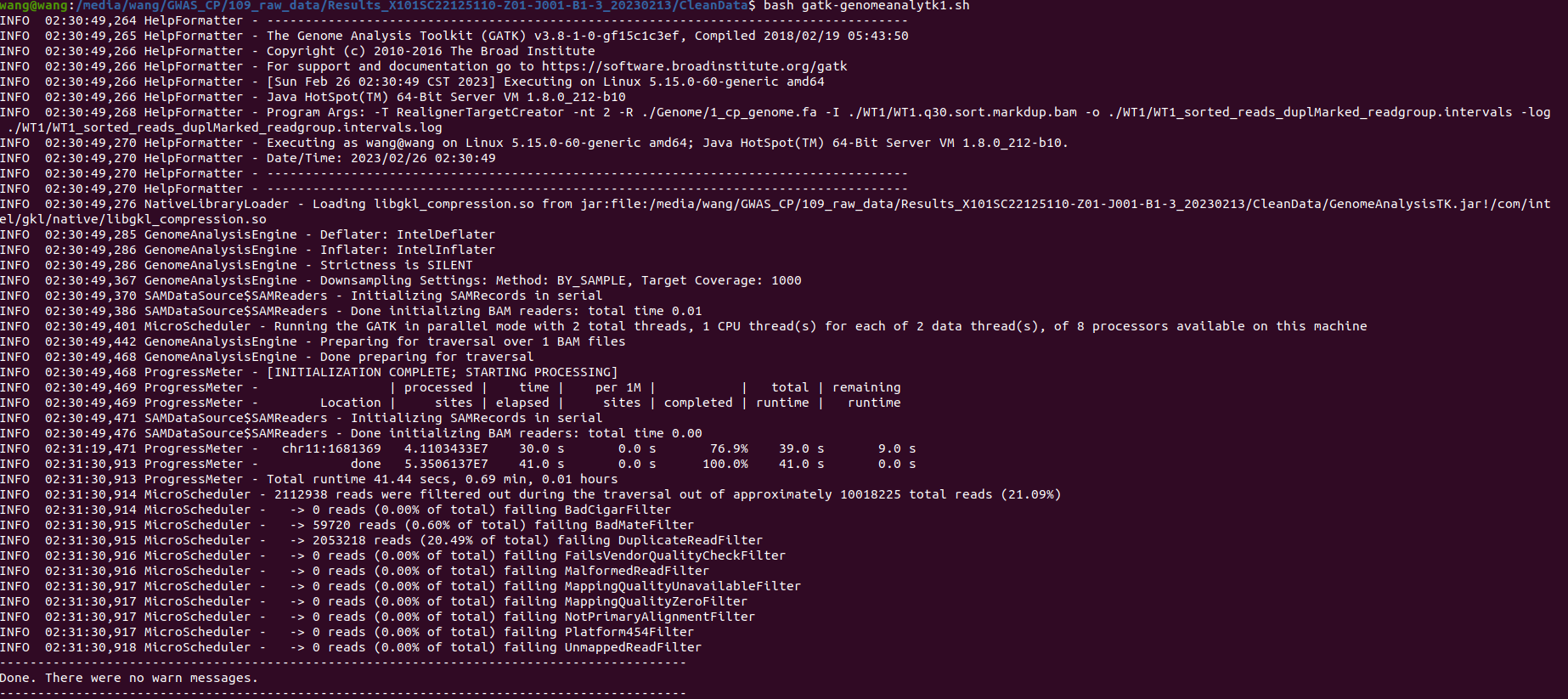
1. Use Samtools to create indexes for all $i\_q30.sortmarkdup.bam files generated in the previous step.

example：samtools index ./WT1/WT1.q30.sort.markdup.bam

1. GATK sometimes has an operation called BQSR (Base Quality Score Recalibration), where BQSR stands for Base Quality Score Recalibration. This is a data preprocessing step for detecting systematic errors, used to estimate the accuracy of each base detection. We mainly consider three factors: the position of bases in reads, the contextual environment, and the original quality value. Firstly, we calculate the distribution model of erroneous bases in the original base quality, and then use this model to correct the base quality and generate new base quality values. As a new non model species, Microcystis aeruginosa MEM25 has no true set of mutations and almost no non model species. And BQSR verifies sequencing quality based on mutation true sets, so this step can be skipped. Refer to steps 2 and 4, as compensation, all reads that may result in inaccurate results have been removed.
2. .Realign Indels， Re compare around INDEL:

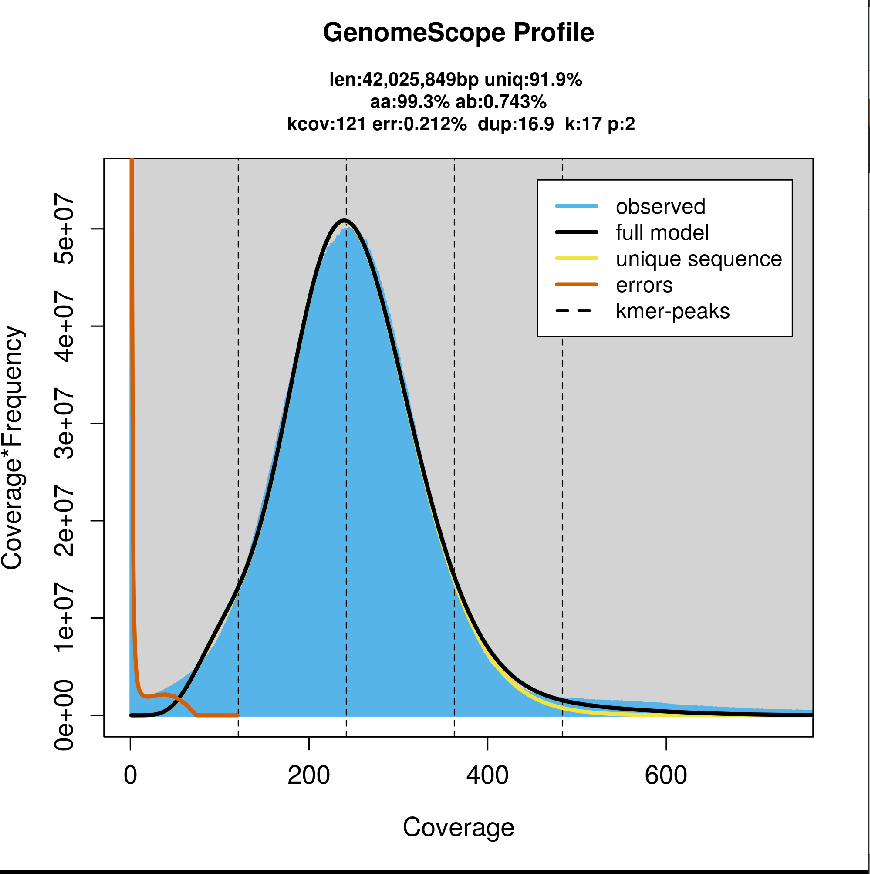
code: java -Xmx2g -jar ./GenomeAnalysisTK.jar -T RealignerTargetCreator -nt 2 -R ./Genome/1\_cp\_genome.fa -I ./WT1/WT1.q30.sort.markdup.bam -o ./WT1/WT1\_sorted\_reads\_duplMarked\_readgroup.intervals -log ./WT1/WT1\_sorted\_reads\_duplMarked\_readgroup.intervals.log

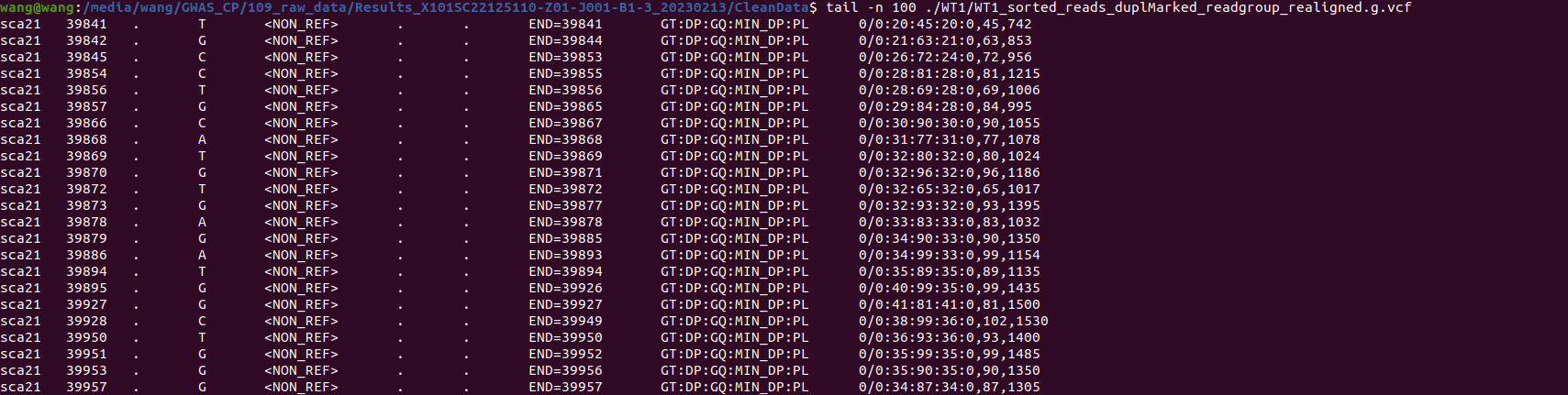
java -Xmx2g -jar ./GenomeAnalysisTK.jar -T IndelRealigner -R ./Genome/1\_cp\_genome.fa -I ./WT1/WT1.q30.sort.markdup.bam -targetIntervals ./WT1/WT1\_sorted\_reads\_duplMarked\_readgroup.intervals -o ./WT1/WT1\_sorted\_reads\_duplMarked\_readgroup\_realigned.bam



1. Calculate GVCF for each sample. （VCF，Variant Call Format）

Based on previous sequencing results, such as genome wild-type or HIC data or RNA seq, it is necessary to first evaluate the heterozygosity of the genome and recommend the Jellyfish+genomescope method. The heterozygosity of MEM25 is approximately 0.00743.



Bbe careful, gvcf。 Be sure to name the output file with the suffix g.vcf, otherwise an error will occur.

note:

-hets,--heterozygosity <heterozygosity> ：Heterozygosity value used to compute prior likelihoods for any locus

-indelHeterozygosity,--indel\_heterozygosity <indel\_heterozygosity> Heterozygosity for indel calling

Generate gvcfs for each sample:

java -Xmx2g -jar ./GenomeAnalysisTK.jar -T HaplotypeCaller -nct 2 -R ./Genome/1\_cp\_genome.fa -I ./WT1/WT1\_sorted\_reads\_duplMarked\_readgroup\_realigned.bam -ERC GVCF -hets 0.001 -o ./WT1/WT1\_sorted\_reads\_duplMarked\_readgroup\_realigned.g.vcf

1. Merge all gvCF files of the samples into one file. Obtain the population variation file vcf. The code is as follows:

java -Xmx2g -jar ./GenomeAnalysisTK.jar \

-T GenotypeGVCFs \

-nt 2 \

-R ./Genome/1\_cp\_genome.fa \

-V sample1

-V sample2

......

-hets 0.001 \

-stand\_call\_conf 30.0 \

-allSites \

-o ./new/GVCFall.vcf

1. Select SNPs and INDELs from the merged gvCF file separately.

java -Xmx4g -jar GenomeAnalysisTK.jar -T SelectVariants -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall.vcf -selectType SNP -o GVCFall\_SNPs.vcf

java -Xmx4g -jar GenomeAnalysisTK.jar -T SelectVariants -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall.vcf -selectType INDEL -o GVCFall\_INDELs.vcf

1. Mutation filtering, SNP and INDEL need to be filtered separately, and the standards for the two are slightly different. It is necessary to first calculate the following indicators, set thresholds on your own, and then filter them.

————————————————

The parameters:

QualByDepth (QD): The reliability of the mutation site divided by the number of unfiltered non reference reads.

Fisher Strand (FS): Fisher's exact test evaluates the likelihood that the current variation is a strand bias, with a value between 0-60.

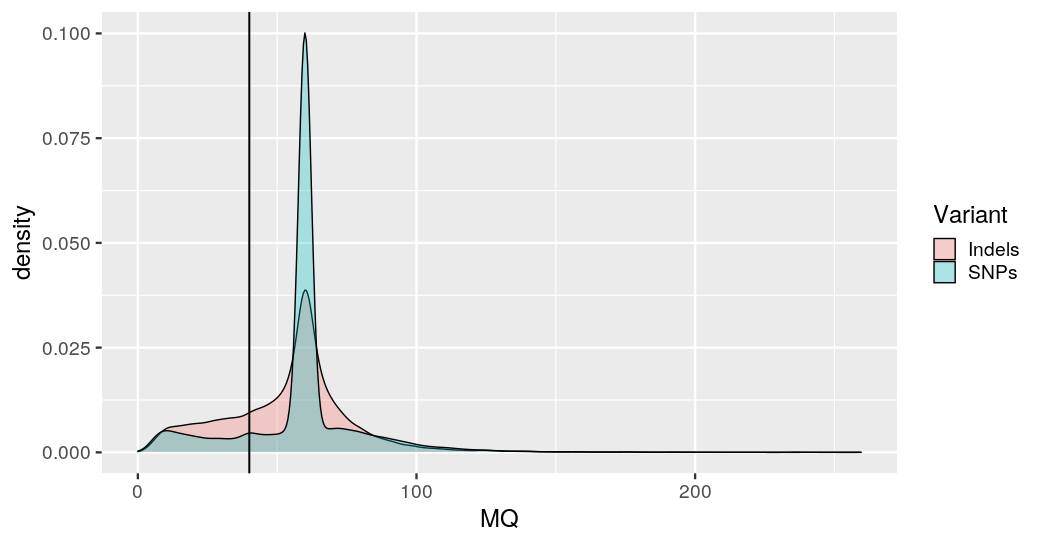
RMSMappingQuality (MQ): The square root of the comparison quality in all samples.

Mapping QualityRankSumTest (MQRankSumTest): Evaluate credibility based on the comparison quality of REF and ALT reads.

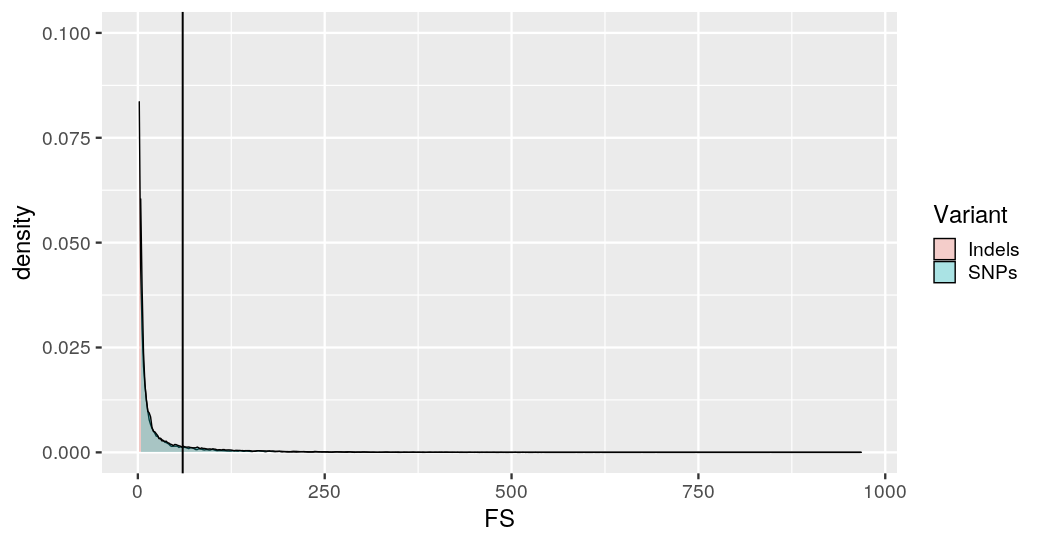
ReadPosRankSumTest (ReadPosRankSumTest): evaluates the reliability of mutations by mutating at the position of the read, usually with high error rates at both ends of the read.

StrandOddsRatio (SOR): Comprehensive assessment of the likelihood of strand bias.

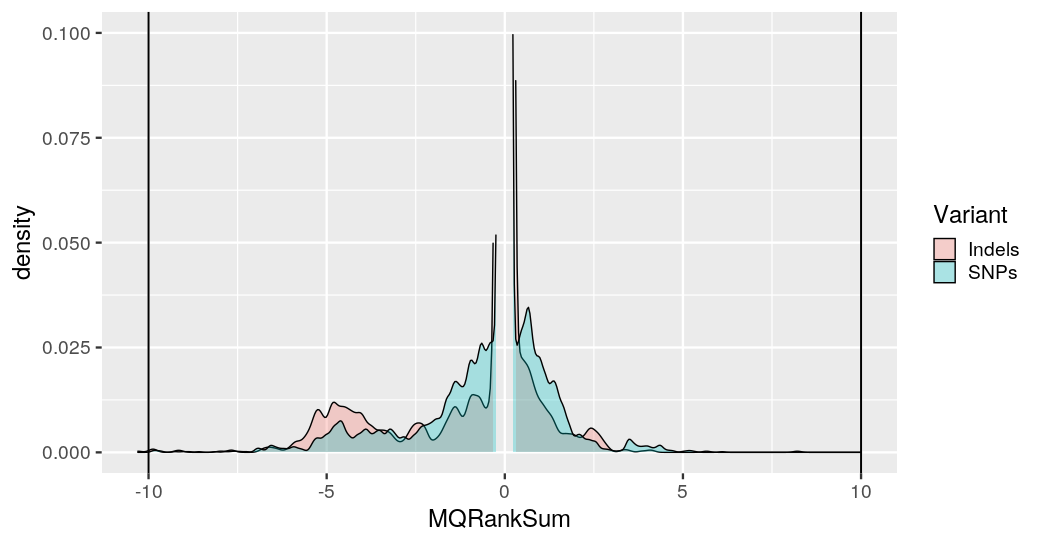
MQ density distribution map, the black vertical line in the figure represents MQ=40.



FS density distribution map, the black vertical line in the figure represents FS=60.



MQRankSum density distribution map, with black vertical lines indicating MQRankSum=10 and MQRankSum=-10.



The SNP filtering code is as follows:

java -jar GenomeAnalysisTK.jar -T VariantFiltration -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_SNPs.vcf --filterExpression "QUAL < 30 || MQ < 40.00 || SOR > 4.000 || QD < 2.00 || FS > 60.000 || MQRankSum < -10.000 || ReadPosRankSum < -10.000 || ReadPosRankSum > 10.000" --filterName "my\_snp\_filter" -o GVCFall\_SNPs\_filter.vcf

grep -E '^#|PASS' GVCFall\_SNPs\_filter.vcf > GVCFall\_SNPs\_filterPASSED.vcf

The INDEL filtering code is as follows:

java -jar GenomeAnalysisTK.jar -T VariantFiltration -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_INDELs.vcf --filterExpression "QUAL < 30 || MQ < 40.00 || SOR > 10.000 || QD < 2.00 || FS > 200.000 || ReadPosRankSum < -20.000 || ReadPosRankSum > 20.000" --filterName "my\_indel\_filter" -o GVCFall\_INDELs\_filter.vcf

grep -E '^#|PASS' GVCFall\_INDELs\_filter.vcf > GVCFall\_INDELs\_filterPASSED.vcf

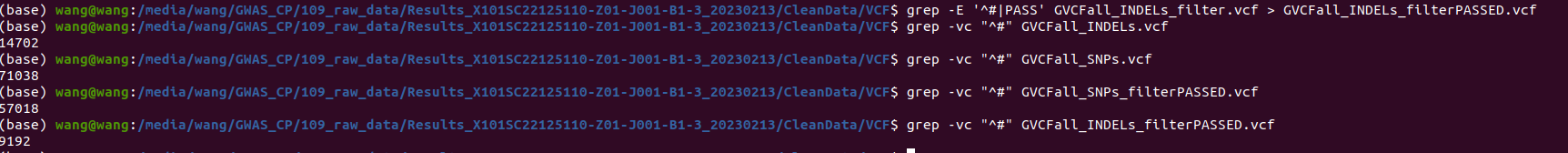
1. Statistics before and after filtering:

grep -vc "^#" GVCFall\_INDELs.vcf

grep -vc "^#" GVCFall\_SNPs.vcf

grep -vc "^#" GVCFall\_SNPs\_filterPASSED.vcf

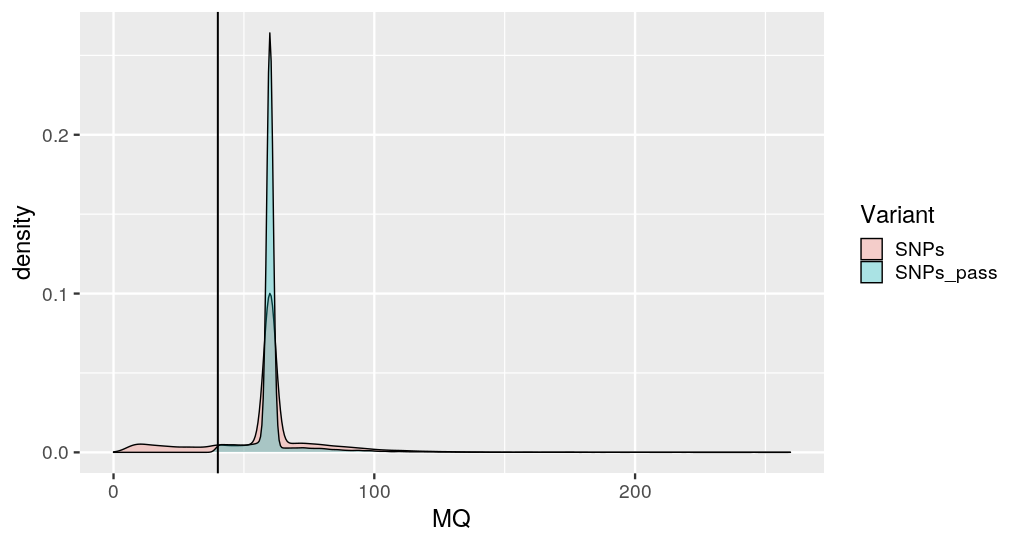
grep -vc "^#" GVCFall\_INDELs\_filterPASSED.vcf



These samples had a total of 18504 INDELs and 87716 SNPs identified, among which there were many low confidence results. After GATK hard filtering, there were 11485 INDELs and 70104 SNPs. Check if the filtering was successful.



It is not difficult to find that after filtering, the MQ quality distribution peak of the SNP set is more convergent and reliable compared to before.

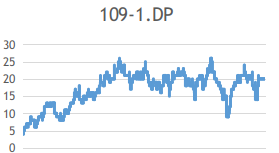


1. Use the VariantsToTable feature in the GATK toolset to convert variant data in VCF format into a table format for subsequent statistical analysis.

java -Xmx2g -jar GenomeAnalysisTK.jar -T VariantsToTable -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall.vcf -F CHROM -F POS -GF GT -GF DP -o GVCFall.DP.table

1. Continue quality control, mark mutations, mark filtered genotypes， mark DP<6 or DP>100.

First, calculate the content of GVCFall.DP.table to understand the average sequencing depth of each variant site in each sample. Select a sample and randomly select 2000 loci, such as 109-1, with an average DP sequencing depth close to 25X. Trim the VCF file based on a threshold of 5X.



java -jar ./GenomeAnalysisTK.jar -T VariantFiltration -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall.vcf -G\_filter "DP < 6 || DP > 100" -G\_filterName "DP\_6-100" -o GVCFall\_DPfilter.vcf

java -jar ./GenomeAnalysisTK.jar -T VariantFiltration -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_SNPs\_filterPASSED.vcf -G\_filter "DP < 6 || DP > 100" -G\_filterName "DP\_6-100" -o GVCFall\_SNPs\_filterPASSED\_DPfilter.vcf

java -jar ./GenomeAnalysisTK.jar -T VariantFiltration -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_INDELs\_filter.vcf -G\_filter "DP < 6 || DP > 100" -G\_filterName "DP\_6-100" -o GVCFall\_INDELs\_filter\_DPfilter.vcf

1. Pick out unmarked mutations, Set filtered sites to no call， That is, the mutation result we ultimately need.

java -jar ./GenomeAnalysisTK.jar -T SelectVariants -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_DPfilter.vcf --setFilteredGtToNocall -o GVCFall\_DPfilterNoCall.vcf

java -jar ./GenomeAnalysisTK.jar -T SelectVariants -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_SNPs\_filterPASSED\_DPfilter.vcf --setFilteredGtToNocall -o GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.vcf

java -jar ./GenomeAnalysisTK.jar -T SelectVariants -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_INDELs\_filterDPfilter.vcf --setFilteredGtToNocall -o GVCFall\_INDELs\_filterPASSED\_DPfilterNoCall.vcf

1. Convert VCF to tab readable format.

java -Xmx2g -jar ./GenomeAnalysisTK.jar -T VariantsToTable -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_DPfilterNoCall.vcf -F CHROM -F POS -GF GT -o whole\_Genome.table

java -Xmx2g -jar ./GenomeAnalysisTK.jar -T VariantsToTable -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.vcf -F CHROM -F POS -GF GT -o GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.table

java -Xmx2g -jar ./GenomeAnalysisTK.jar -T VariantsToTable -R /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/Genome/1\_cp\_genome.fa -V GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.vcf -F CHROM -F POS -GF GT -o GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.table

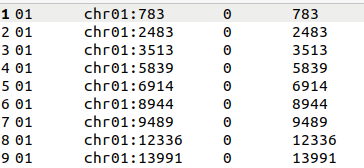
1. Before conducting GWAS analysis using the plink, conduct a population test to see if there are any issues with the sample.

During GWAS research, we often encounter the phenomenon of population stratification, which refers to the diversity of ancestral sources in the population. If the SNP frequencies of different populations are not the same, it may lead to false positive sites (not necessarily significant signal sites related to the phenotype, but may be related to differences in population SNP frequencies) during association analysis. Therefore, we need to perform PCA analysis on the population before association analysis, and then add the PCA results as covariates to the association analysis (covariates are independent variables (explanatory variables) that are not manipulated by the experimenters, but still affect the experimental results, such as random errors outside the main experimental conditions).

First, format conversion is performed to generate ped and map files. The content and format of the ped file are shown in the following figure. The first column is the group ID, the second column is the sample ID, the third column is the parent ID, where 'none' means 0, the fourth column is the parent ID, where 'none' means 0 or -9, the fifth column is gender, which means 0 if not distinguished, and the sixth column shows whether the sample belongs to the treatment group or the control group. The first column is the control group, the second column is the processing group, and 0 is not distinguished. The sequence information on each SNP base is anchored by the map file. In practical operation, the sixth column file of ped can be a continuity property, which means there are other values besides 1, 2, 0, and -9. When there are multiple phenotypes, separate files need to be prepared, while ignoring the phenotypes in the ped

vcftools --vcf GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.vcf --plink --out snp

The map file is as follows, containing information about each SNP locus (chromosome number, SNP ID, position (this defaults to 0), coordinates):



Then convert the ped file to a bed file for PCA analysis:

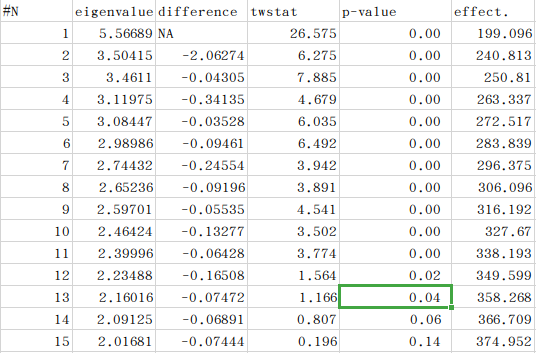
plink --file snp --make-bed --out myfile

Perform PCA calculation using plink again:

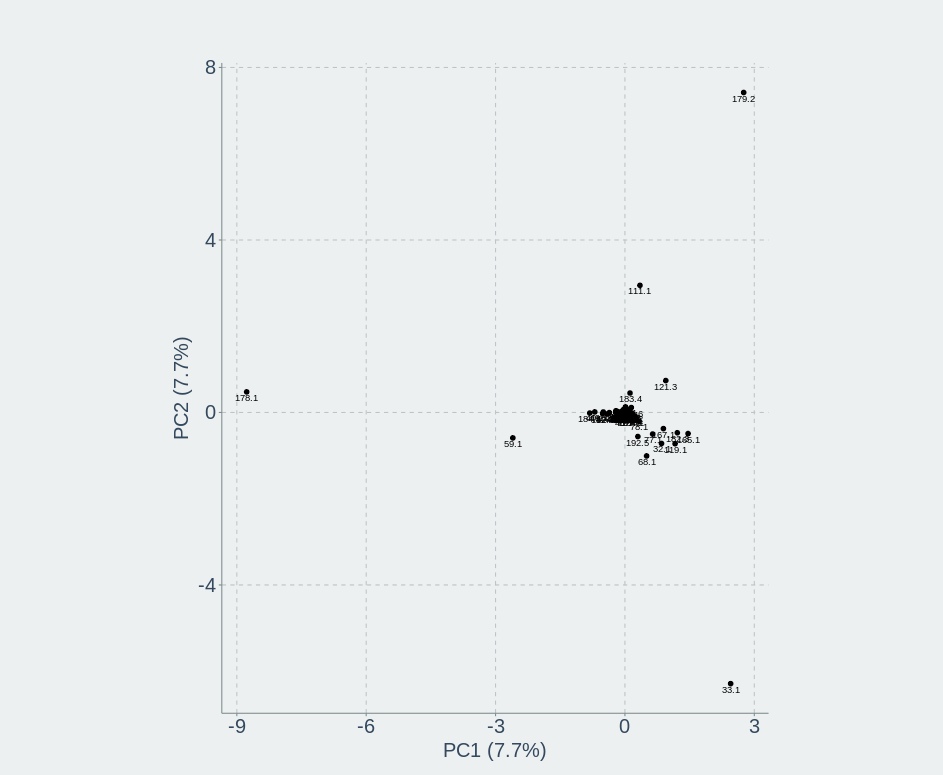
plink --bfile myfile --pca 100 --out myfile\_pca

Then use twstats to calculate how PCA should be valued:

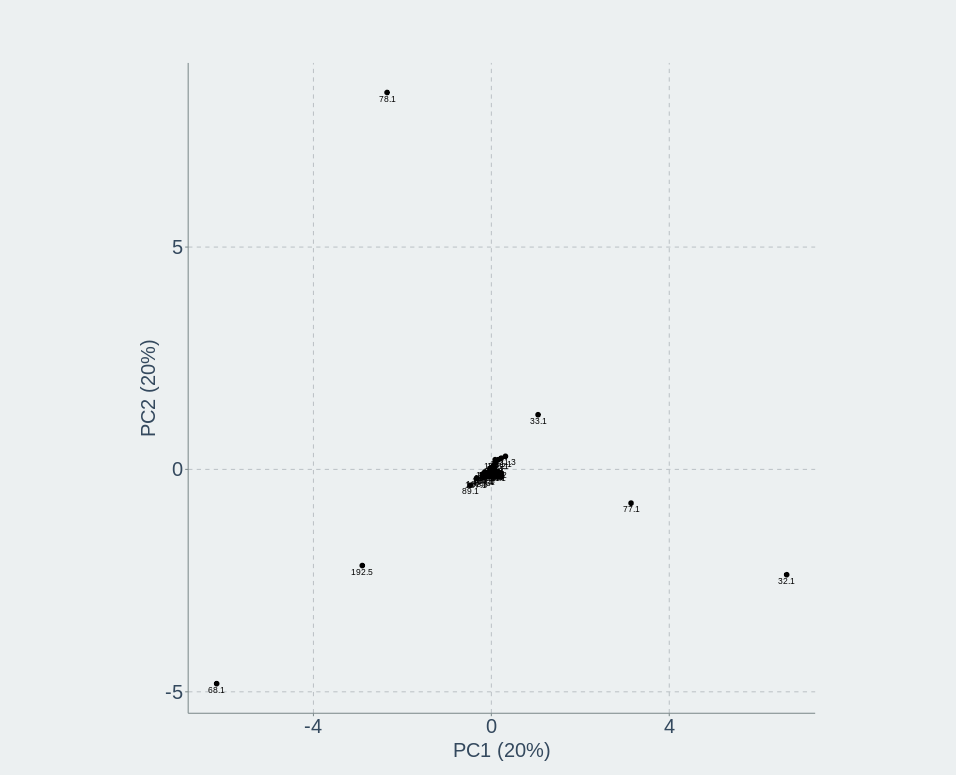
twstats -t twtable -i myfile\_pca.eigenval -o eigenvaltw.out，

The generated eigenvaltw.out is as follows. The P-values of the first 13 PCA are less than 0.05, indicating that principal components 1-13 have a significant impact on population structure (P<0.05). Therefore, when conducting association analysis, the first 13 PCA should be added to the covariates (in fact, the number of PCA selected has little impact, just to make the data more statistically significant after normalization).

Choose PCA 13 and regenerate the target file: plink --bfile myfile --PCA 13 --out myfile\_pca. The PCA result is as follows (matrix transposition is required before using the Eigenvec file to draw the PCA image):



When PCA selects 5, the result is as follows. As can be seen, there are differences, but not significant. The main focus here is to check whether there is population stratification in the MEM25 mutants, because all mutants of MEM25 come from the same ancestor and are induced by the same concentration of EMS. The frequency of gene mutations is roughly the same. Therefore, in the PCA diagram, the samples should be close to each other and there should be no stratification phenomenon. As shown in the PCA diagram, except for a few samples due to special reasons, the rest tend to be consistent without population stratification, indicating that the samples come from the same ancestor, which is in line with expectations.

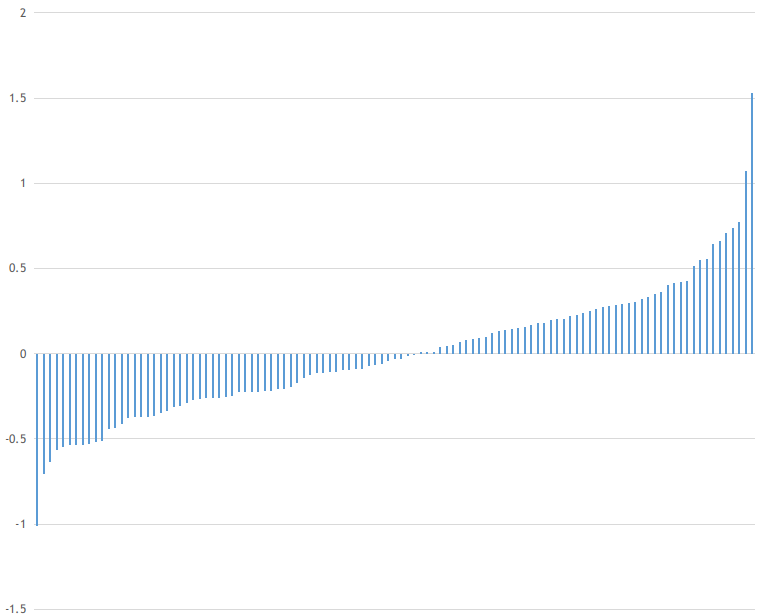


1. Phenotypic data organization:

MEM25, as a single-cell green algae, has excellent industrial potential and a simple and easy to operate genetic transformation system. Here, dry weight, cell size, and cell count are selected as phenotypic indicators to evaluate the growth potential of CP, and GWAS is used to seek methods that can enhance or inhibit CP growth potential, in order to expand the industrial value of CP and optimize the genetic transformation system. It should be noted that phenotype determination is limited by space, equipment, time, and other factors, and all phenotypes of all mutants cannot be obtained at the same time and in the same batch, resulting in batch effects between mutant phenotypes.

What we need to clarify here is that quality traits only involve phenotypes such as life and death, and in this case, only the batch issue of sequencing data needs to be considered; In GWAS, for continuous traits, association analysis generally belongs to linear models, and data needs to follow a normal distribution, mainly based on the trend of data size rather than absolute data size.

To address the batch effect of continuity traits, the dry weight values of each sample in each batch were subtracted from the dry weight values of the WT wild type in each batch, and the difference between them and the wild type was used as the final data. The data is shown in the following figure, which includes experimental data from three batches. The final result conforms to a normal distribution (WT wild type is not selected, and other batches with overlapping samples can also be selected, which requires sample overlap between batches in the experimental process for correction).



1. Phenotypic genotype association analysis:

bcftools view -S ID.txt /media/wang/GWAS\_CP/109\_raw\_data/Results\_X101SC22125110-Z01-J001-B1-3\_20230213/CleanData/new/GVCFall\_SNPs\_filterPASSED\_DPfilterNoCall.vcf -Ov > batch1.vcf

Filter VCF files to further eliminate low confidence genotypes, filter out genotypes with maf less than or equal to 0.05, and remove genotypes with a deletion rate of 0.2 between samples:

vcftools --vcf batch1.vcf --min-alleles 2 --max-alleles 2 --maf 0.05 --max-missing 0.2 --minQ 20 --recode --out test

Sort the test.record.vcf files filtered in the previous step:

run\_pipeline.pl -Xmx2g -Xms1G -SortGenotypeFilePlugin -inputFile ./batch1/test.recode.vcf -outputFile ./batch1/snp.sort.vcf -fileType VCF

Calculate the relationship matrix:

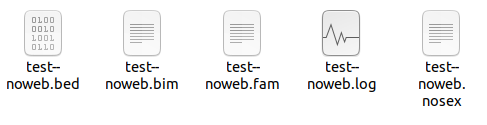
run\_pipeline.pl -fork1 -Xms512m -Xmx2g -vcf ./batch1/snp.sort.vcf -ck -export ./batch1/bnapus.kinship.txt -runfork1

Generate ped format:

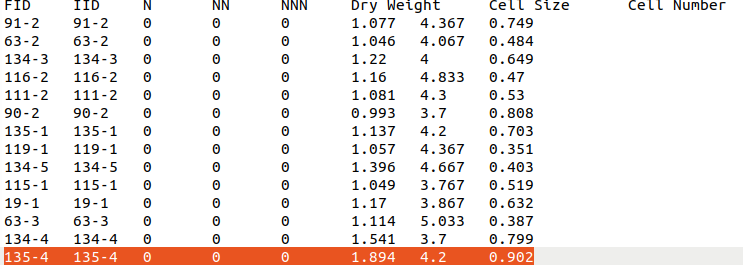
vcftools --vcf batch1/snp.sort.vcf --plink --out ./batch1/out

Convert ped format to bed file (this step sometimes produces strange errors, as if it is a semi call issue):

plink --file out --make-bed --out test--noweb



Prepare phenotype file. The first two columns of Pheno are FID and IID, and the third column is Phenotype.



#Continuous phenotype GWAS analysis

plink --bfile test --pheno Pheno.txt --linear --out gwas\_results