

Practical Guideline for Whole Genome Sequencing



SOLID 5500



Illumina HiSeq2000

Disclosure

Kwangsik Nho

Assistant Professor

Center for Neuroimaging

Department of Radiology and Imaging Sciences

Center for Computational Biology and Bioinformatics

Indiana University School of Medicine

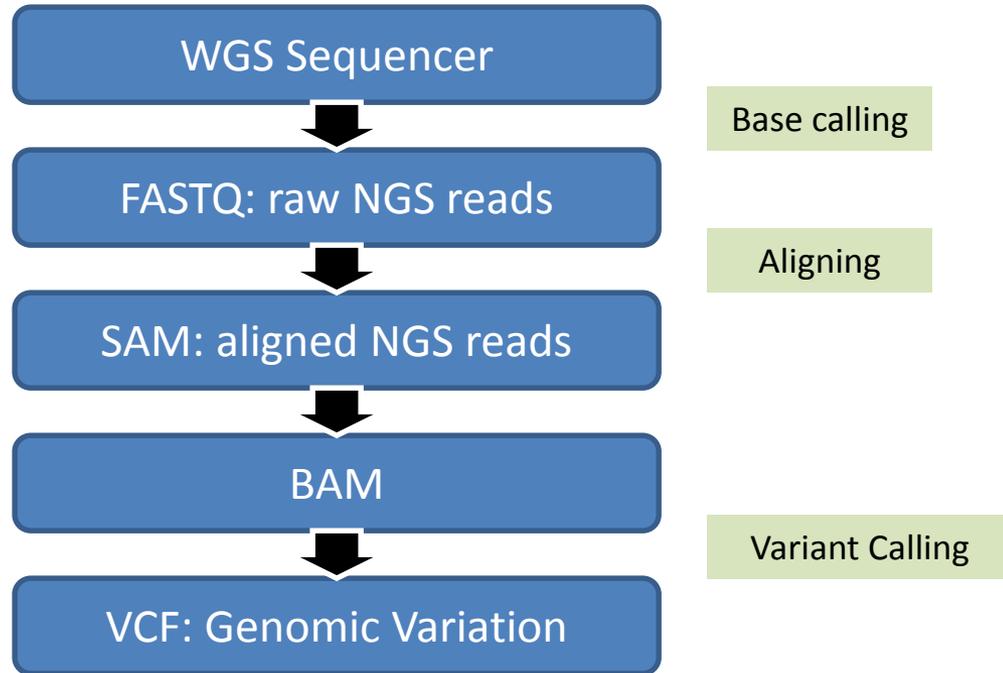
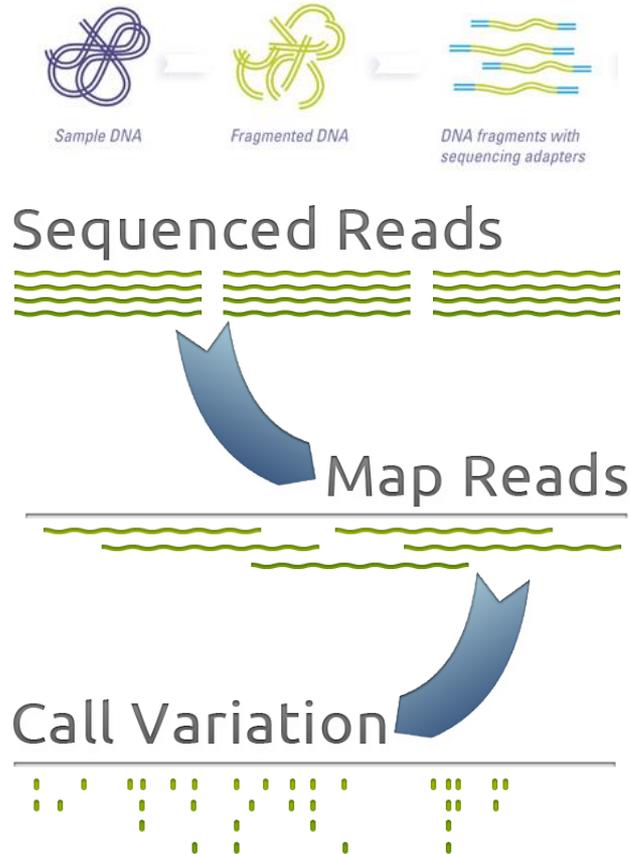


- Kwangsik Nho discloses that he has no relationships with commercial interests.

What You Will Learn Today

- Basic File Formats in WGS
- Practical WGS Analysis Pipeline
- WGS Association Analysis Methods

Whole Genome Sequencing File Formats



How have BIG data problems been solved in next generation sequencing?

Whole Genome Sequencing File Formats

Line 1 begins with a '@' character and is followed by a sequence identifier and an *optional* description

Line 2 is the raw sequence letters

Line 3 begins with a '+' character and is *optionally* followed by the same sequence identifier (and any description) again

Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence

@HS2000-306_201:6:1204:19922:79127/1

Column	Brief Description
HS2000-306_201	the unique instrument name
6	flowcell lane
1204	tile number within the flowcell lane
19922	x-coordinate of the cluster within the tile
79127	y-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 (paired-end)

Whole Genome Sequencing File Formats

Line 1 begins with a '@' character and is followed by a sequence identifier and an *optional* description

Line 2 is the raw sequence letters

Line 3 begins with a '+' character and is *optionally* followed by the same sequence identifier (and any description) again

Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence

```
ACGTCTGGCCTAAAGCACTTTTTCTGAATTC...
```

Sequence

```
+
```

```
BC@DFDFFHHHHJJJJJJJJJJJJJJJJJJJJH...
```

Base Qualities

Base Qualities = ASCII 33 + Phred scaled Q

$$\text{Phred scaled Q} = -10 \cdot \log_{10}(e)$$

e: base-calling error probability

SAM encoding adds 33 to the value because ASCII 33 is the first visible character

Whole Genome Sequencing File Formats

- The Alignment section contains the information for each sequence about where/how it aligns to the reference genome
 - are all fragments properly aligned?
 - is this fragment unmapped?
 - did this read fail quality controls?
 - is this read a PCR or optical duplicate?
 - ...

Whole Genome Sequencing File Formats

- The **SAM/BAM (Sequence Alignment/Map)** file format comes in a plain text format (SAM) and a compressed binary format (BAM)
- The BAM format stores aligned reads and is technology independent

Whole Genome Sequencing File Formats

- **VCF (Variant Call Format):** a text file format containing meta-information lines; a header line, and then data lines (each containing information about a position in the genome)

```
##fileformat=VCFv4.1
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=BaseQRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities">
##reference=file:///N/dc2/projects/adniwgs/Human_Reference/human_g1k_v37.fasta
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT LP6005123-DNA_D06
1 14673 . G C 48.77 . AC=1;AF=0.500;AN=2;DP=12;FS=3.090;MLEAC=1;MLEAF=0.500;MQ=24.16;MQ0=0;QD=6.97 GT:AD:DP:GQ:PL 0/1:8,4:12:77:77,0,150
1 14907 rs79585140 A G 476.77 . AC=1;AF=0.500;AN=2;DB;DP=43;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=30.93;MQ0=0;QD=30.63 GT:AD:DP:GQ:PL 0/1:21,22:43:99:505,0,437
1 14930 rs75454623 A G 589.77 . AC=1;AF=0.500;AN=2;DB;DP=60;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=29.24;MQ0=0;QD=29.09 GT:AD:DP:GQ:PL 0/1:27,33:60:99:618,0,513
1 15211 rs78601809 T G 169.84 . AC=2;AF=1.00;AN=2;DB;DP=6;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=39.00;MQ0=0;QD=34.24 GT:AD:DP:GQ:PL 1/1:0,6:6:18:198,18,0
```

Whole Genome Sequencing File Formats

- **VCF (Variant Call Format):** a text file format containing meta-information lines; a header line, and then data lines (each containing information about a position in the genome)

```
##fileformat=VCFv4.1
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=BaseQRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities">
##reference=file:///N/dc2/projects/adniwgs/Human_Reference/human_g1k_v37.fasta
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT LP6005123-DNA_D06
```

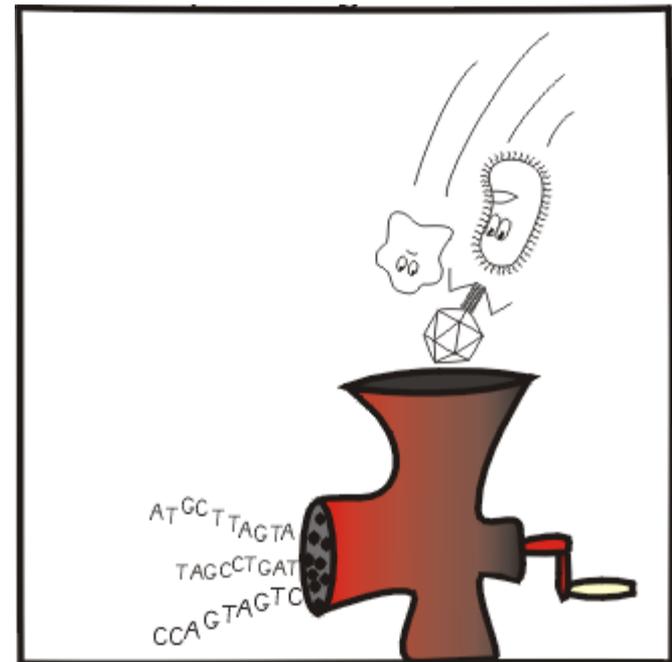
header

```
1 14673 . G C 48.77 . AC=1;AF=0.500;AN=2;DP=12;FS=3.090;MLEAC=1;MLEAF=0.500;MQ=24.16;MQ0=0;QD=6.97 GT:AD:DP:GQ:PL 0/1:8,4:12:77:0,150
1 14907 rs79585140 A G 476.77 . AC=1;AF=0.500;AN=2;DB;DP=43;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=30.93;MQ0=0;QD=30.63 GT:AD:DP:GQ:PL 0/1:21,22:43:99:505,0,437
1 14930 rs75454623 A G 589.77 . AC=1;AF=0.500;AN=2;DB;DP=60;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=29.24;MQ0=0;QD=29.09 GT:AD:DP:GQ:PL 0/1:27,33:60:99:618,0,513
1 15211 rs78601809 T G 169.84 . AC=2;AF=1.00;AN=2;DB;DP=6;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=39.00;MQ0=0;QD=34.24 GT:AD:DP:GQ:PL 1/1:0,6:6:18:198,18,0
```

variant records

Pipeline for Whole Genome Sequencing

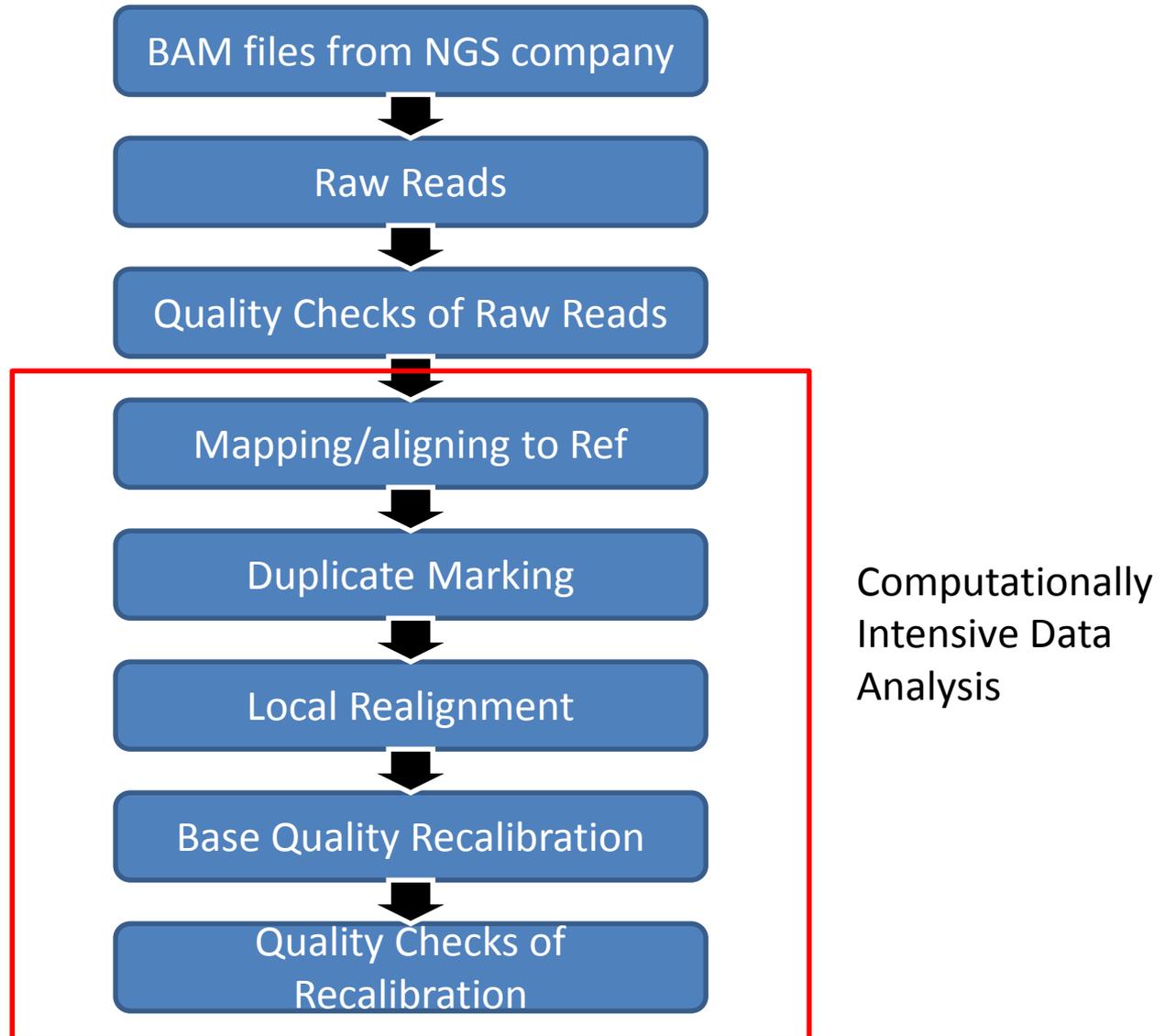
- Data Pre-Processing
- Variant Calling
- Preliminary Analysis



by Viktor S. Poór

Data Pre-Processing

Data Pre-Processing



Data Pre-Processing

Preparing a reference for use with BWA and GATK

❖ Prerequisites: Installed BWA, SAMTOOLS, and PICARD

1. Generate the BWA index

➤ Action:

```
Bwa index -a bwtsv reference.fa
```

2. Generate the fasta file index

➤ Action:

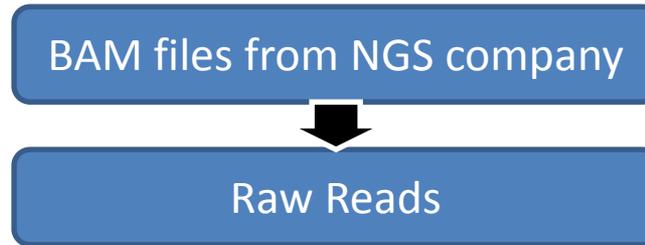
```
Samtools faidx reference.fa
```

3. Generate the sequence dictionary

➤ Action:

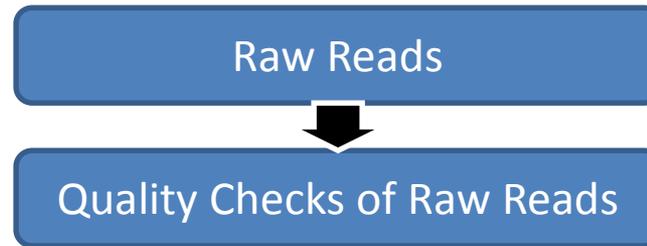
```
java -jar CreateSequenceDictionary.jar REFERENCE=reference.fa  
OUTPUT=reference.dict
```

Data Pre-Processing



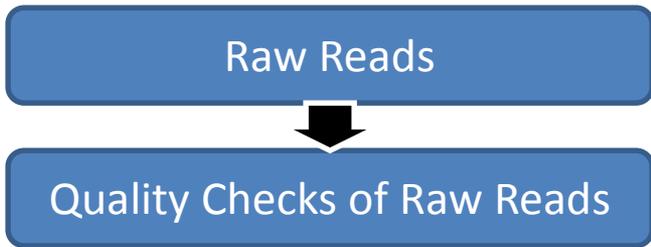
- ❖ Prerequisites: Installed HTSlib (<https://github.com/samtools/htslib>)
- 1. Shuffling the reads in the BAM file
 - Action:
`htscmd bamshuf -uOn 128 in.bam tmp > shuffled_reads.bam`
- 2. Revert the BAM file to FastQ format
 - Action:
`htscmd bam2fq -aOs singletons.fq.gz shuffled_reads.bam > interleaved_reads.fq`

Data Pre-Processing

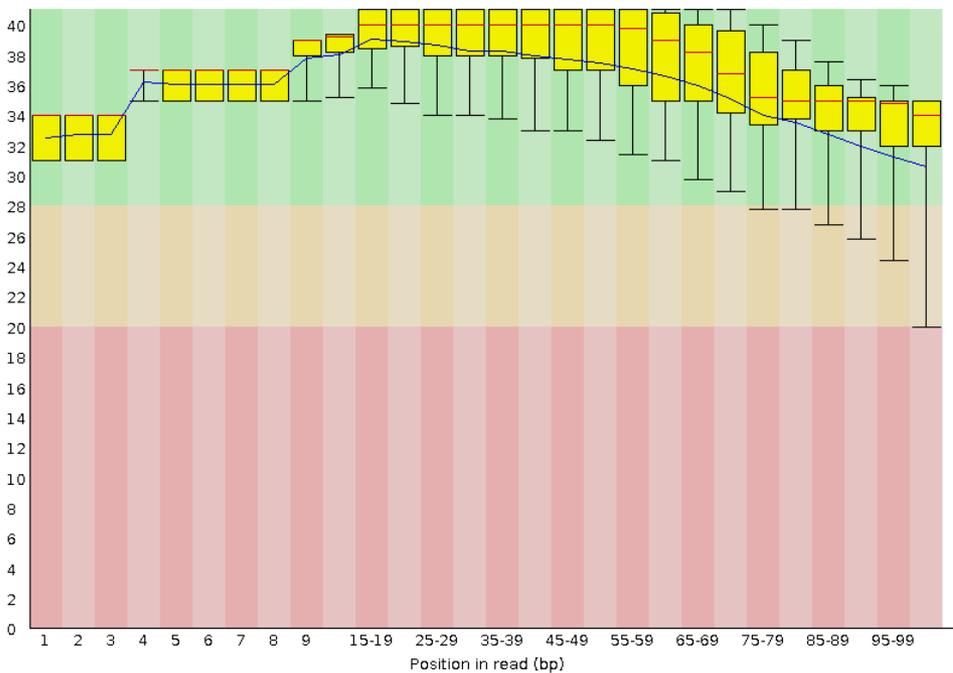


- ❖ Prerequisites: Installed FastQC
(<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>)
- 1. Checks whether a set of sequence reads in a FastQ file exhibit any unusual qualities
 - Action:
`fastqc input.fastq --outdir=/net/scratch2/FastQC`

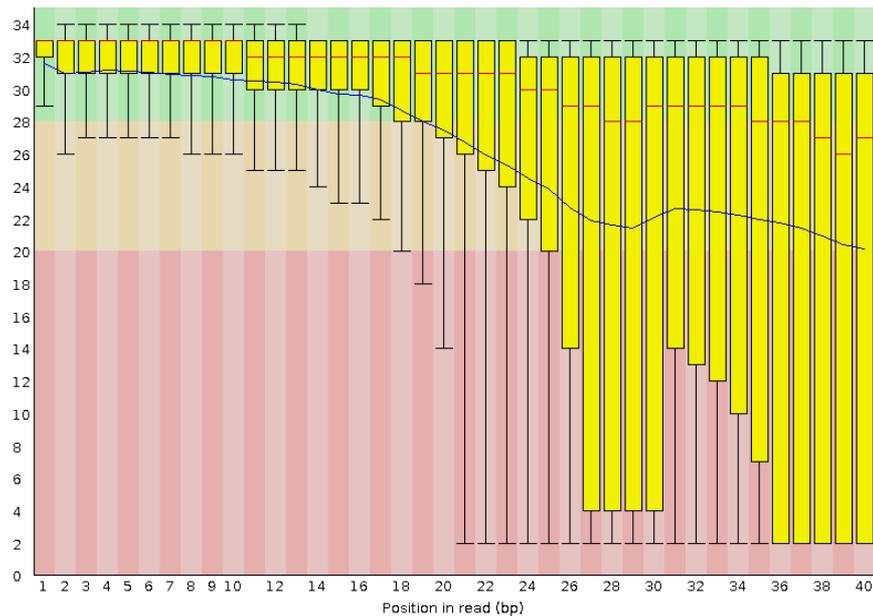
Data Pre-Processing



Quality Scores



Quality Scores



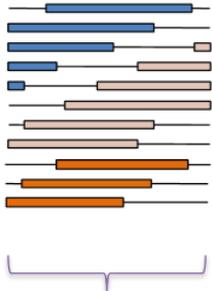
poor dataset

Data Pre-Processing

Quality Checks of Raw Reads

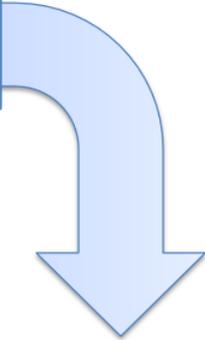


Mapping/aligning to Ref

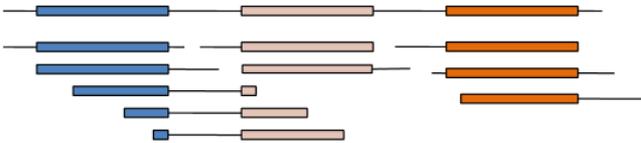


Enormous pile of short reads from NGS

Mapping and alignment algorithms



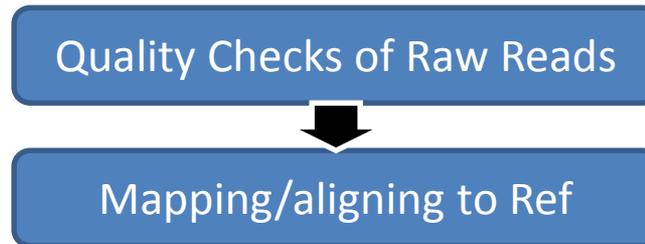
→ Mapping quality (MQ)



Reference genome

Reads mapped to reference

Data Pre-Processing



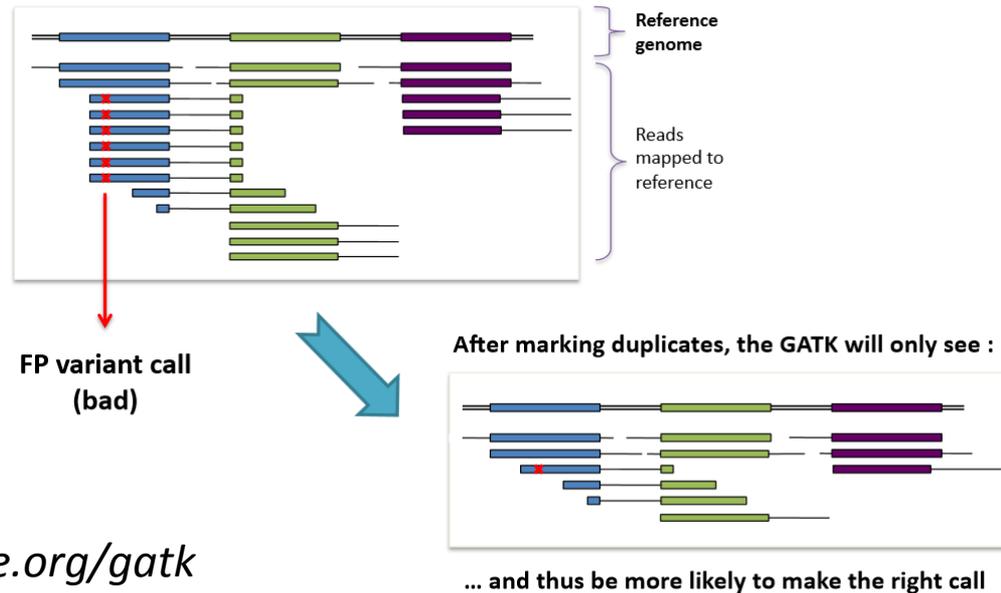
- ❖ Prerequisites: BWA (<http://bio-bwa.sourceforge.net/>), SAMTOOLS (<http://samtools.sourceforge.net/>), Human Reference (hg19)
- 1. Mapping sequencing reads against a reference genome
 - Action:
BWA mem -aMp -t #ofCPUs ref.fa -R
"@RG\tID:**\tLB:**\tPL:ILLUMINA\tSM:**\tPU:BARCODE" >
output.sam
- 2. Converting a SAM file to a BAM file and sorting a BAM file by coordinates
 - Action:
SAMTOOLS view -S -h -b -t ref.fa output.sam -o output.bam
SAMTOOLS sort -m 10000000000 output.bam output.sorted

Data Pre-Processing

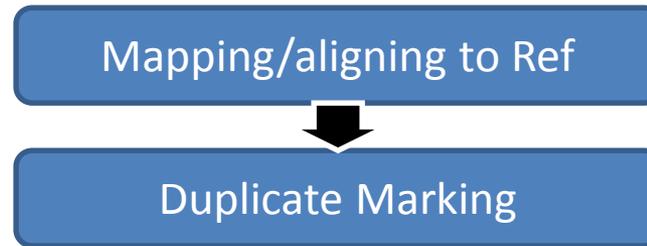


1. Duplicates originate mostly from DNA preparation methods
2. Sequencing error propagates in duplicates

✖ = sequencing error propagated in duplicates



Data Pre-Processing



❖ Prerequisites: JAVA and PICARD (<http://picard.sourceforge.net/>)

1. Examining aligned records in the BAM file to locate duplicate reads

➤ Action:

```
java -Xmx6g -jar PICARD/MarkDuplicates.jar INPUT=output.sorted.bam  
MAX_RECORDS_IN_RAM=2000000 REMOVE_DUPLICATES=false  
VALIDATION_STRINGENCY=SILENT ASSUME_SORTED=true  
METRICS_FILE=output.dups OUTPUT=output.sortedDeDup.bam
```

Data Pre-Processing



❖ Prerequisites: SAMTOOLS, GATK (<https://www.broadinstitute.org/gatk/>)

1. Indexing sorted alignment for fast random access

➤ Action:

SAMTOOLS index output.sortedDeDup.bam

2. Performing local realignment around indels to correct mapping-related artifacts

1) Create a target list of intervals to be realigned

➤ Action:

```
java -Xmx6g -jar GATK -T RealignerTargetCreator -nt #ofCPUs -R Reference  
-I output.sortedDeDup.bam -known INDEL1 -known INDEL2 -log  
output.intervals.log -o output.ForIndelRealigner.intervals
```

Data Pre-Processing



- ❖ Prerequisites: SAMTOOLS, GATK (<https://www.broadinstitute.org/gatk/>)
- 2. Performing local realignment around indels to correct mapping-related artifacts
 - 1) Create a target list of intervals to be realigned
 - 2) Perform realignment of the target intervals
 - Action:

```
java -Xmx6g -jar GATK -T IndelRealigner -R Reference -l  
output.sortedDeDup.bam -targetIntervals  
output.ForIndelRealigner.intervals -known INDEL1 -known INDEL2 -  
model USE_READS -LOD 0.4 --filter_bases_not_stored -log  
output.realigned.log -o output.GATKrealigned.bam
```

Data Pre-Processing



❖ Prerequisites: GATK

1. Recalibrating base quality scores in order to correct sequencing errors and other experimental artifacts

➤ Actions:

```
java -Xmx6g -jar GATK -T BaseRecalibrator -R Reference -I
  output.GATKrealigned.bam -nct #ofCPUS --default_platform
  ILLUMINA --force_platform ILLUMINA -knownSites DBSNP -
  knownSites INDEL1 -knownSites INDEL2 -l INFO -log
  output.BQRecal.log -o output.GATKrealigned.recal_data.table
java -Xmx6g -jar GATK -T PrintReads -R Reference -I
  output.GATKrealigned.bam -nct #ofCPUS -BQSR
  output.GATKrealigned.recal_data.table -l INFO -log
  output.BQnewQual.log -o output.GATKrealigned.Recal.bam
```

Data Pre-Processing



❖ Prerequisites: GATK

1. Generating a plot report to assess the quality of a recalibration

➤ Actions:

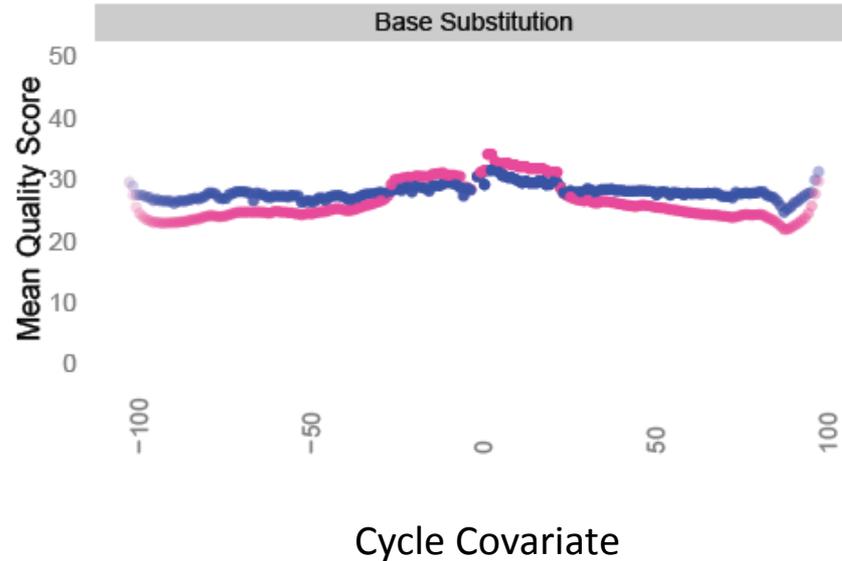
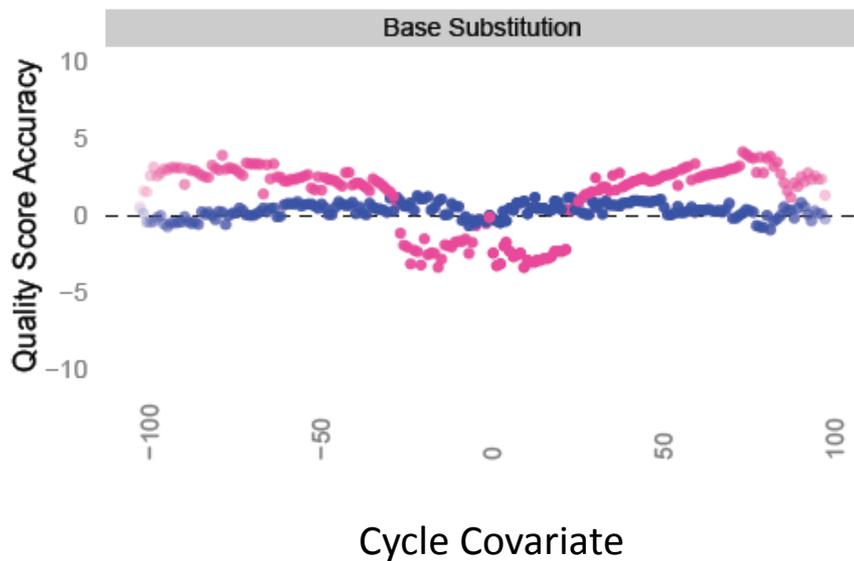
```
java -Xmx6g -jar GATK -T BaseRecalibrator -R Reference -I  
output.GATKrealigned.Recal.bam -nct #ofCPUS --default_platform  
ILLUMINA --force_platform ILLUMINA -knownSites DBSNP -  
knownSites INDEL1 -knownSites INDEL2 -I INFO -BQSR  
output.GATKrealigned.recal_data.table -log output.BQRecal.After.log -  
o output.GATKrealigned.recal_data_after.table  
java -Xmx6g -jar GATK -T AnalyzeCovariates -R Reference -before  
output.GATKrealigned.recal_data.table -after  
output.GATKrealigned.recal_data_after.table -plots output.plots.pdf -  
csv output.plots.csv
```

Data Pre-Processing

Base Quality Recalibration

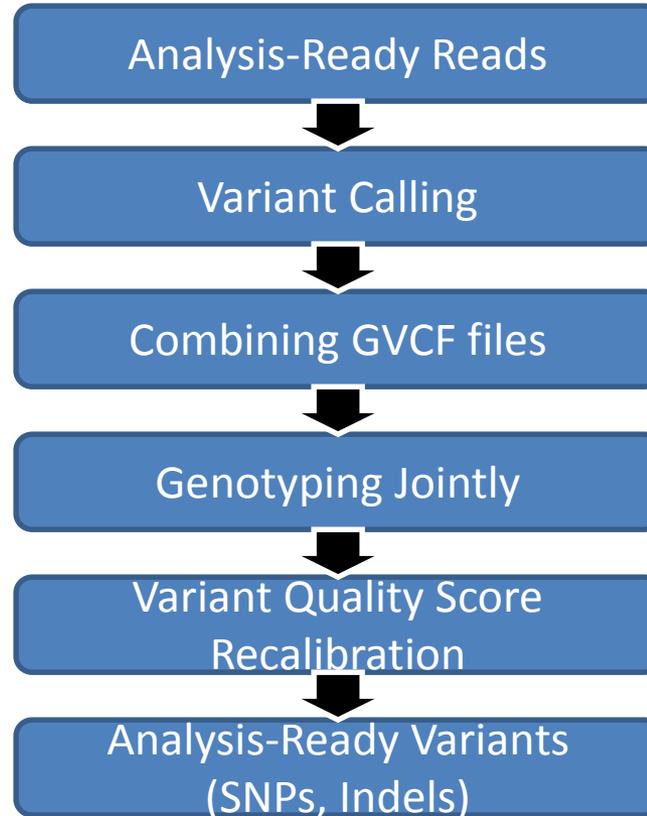


Quality Checks of
Recalibration



Variant Calling

Variant Calling



Variant Calling



Method1: Call SNPs and indels separately by considering each variant locus independently; very fast, independent base assumption

Method2: Call SNPs, indels, and some SVs simultaneously by performing a local de-novo assembly; more computationally intensive but more accurate

Variant Calling



❖ Prerequisites: GATK

1. Calling SNVs and indels simultaneously via local de-novo assembly of haplotypes

➤ Actions:

```
java -Xmx25g -jar GATK -T HaplotypeCaller -nct #ofCPUs -R Reference -l  
output.GATKrealigned.Recal.bam --genotyping_mode DISCOVERY --  
minPruning 3 -ERC GVCF -variant_index_type LINEAR -  
variant_index_parameter 128000 -stand_emit_conf 10 -  
stand_call_conf 30 -o output.raw.vcf
```

Tips:

- stand_call_conf: Qual score at which to call the variant
- stand_emit_conf: Qual score at which to emit the variant as filtered
- minPruning: Amount of pruning to do in the deBruijn graph

Raw variant files are often very large and full of false positive variant calls.

Variant Calling



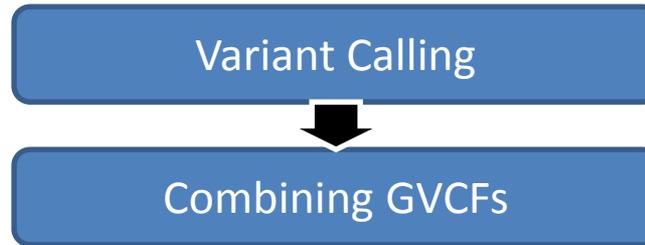
❖ Prerequisites: GATK

1. Calling SNVs and indels simultaneously via a Bayesian genotype likelihood model

➤ Actions:

```
java -Xmx6g -jar GATK -T UnifiedGenotyper -glm BOTH -nt #ofCPUs -R  
Reference -S SILENT -dbsnp DBSNP -l  
output.GATKrealigned.Recal.bam -l INFO -stand_emit_conf 10 -  
stand_call_conf 30 -dcov 200 -metrics output.SNV.1030.raw.metrics -  
log output.SNV.1030.raw.log -o output.raw.vcf
```

Variant Calling



❖ Prerequisites: GATK

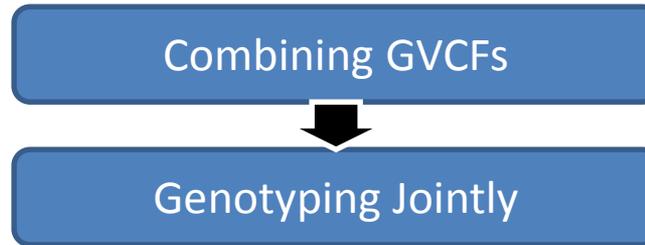
1. Combining any number of gVCF files that were produced by the Haplotype Caller into a single joint gVCF file

➤ Actions:

```
java -Xmx6g -jar GATK -T CombineGVCFs -R Reference --variant  
GVCFList.list -o combined.raw.vcf
```

Tip: if you have more than a few hundred WGS samples, run CombineGVCFs on batches of ~200 gVCFs to hierarchically merge them into a single gVCF.

Variant Calling



❖ Prerequisites: GATK

1. Combining any number of gVCF files that were produced by the Haplotype Caller into a single joint gVCF file

➤ Actions:

```
java -Xmx6g -jar GATK -T GenotypeGVCFs -R Reference -nt #OfCPUs --  
variant CombinedGVCFList.list --dbSNP DBSNP -o  
AllSubject.GenotypeJoint.raw.vcf
```

Variant Calling



Purpose: Assigning a well-calibrated probability to each variant call in a call set

1. **VariantRecalibrator:** Create a Gaussian mixture model by looking at the annotations values over a high quality subset of the input call set and then evaluate all input variants
2. **ApplyRecalibration:** Apply the model parameters to each variant in input VCF files producing a recalibrated VCF file

Tips: Recalibrating first only SNPs and then indels, separately

Variant Calling



❖ Prerequisites: GATK

➤ Actions:

- 1) `java -Xmx6g -jar GATK -T VariantRecalibrator -R Reference -input raw.vcf -nt #OfCPUs -an DP -an QD -an FS {...} -resource RESOURCE -mode SNP -recalFile SNP.recal -tranchesFile SNP.tranches`
- 2) `java -Xmx6g -jar GATK -T ApplyRecalibration -R Reference -input raw.vcf -nt #OfCPUs -mode SNP -recalFile SNP.recal -tranchesFile SNP.tranches -o recal.SNP.vcf -ts_filter_level 99.5`

Variant Calling



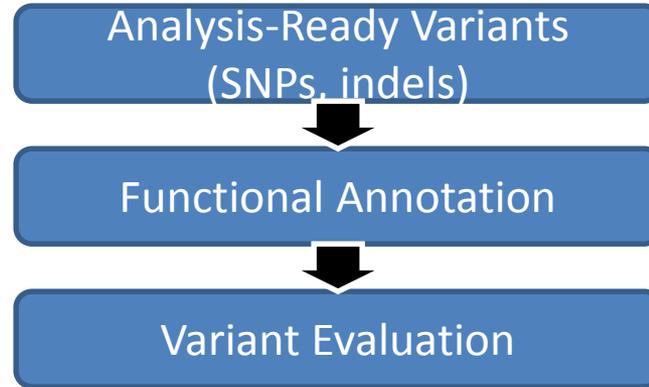
❖ Prerequisites: GATK

➤ RESOURCE

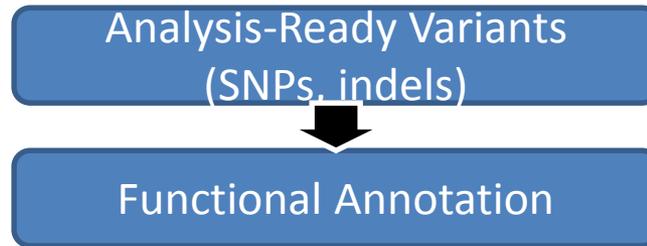
```
-resource:hapmap,known=false,training=true,truth=true,prior=15.0 HAPMAP  
-resource:omni,known=false,training=true,truth=true,prior=12.0 OMNI  
-resource:1000G,known=false,training=true,truth=false,prior=10.0 G1000  
-resource:dbsnp,known=true,training=false,truth=false,prior=2.0 DBSNP
```

Preliminary Analysis

Preliminary Analysis

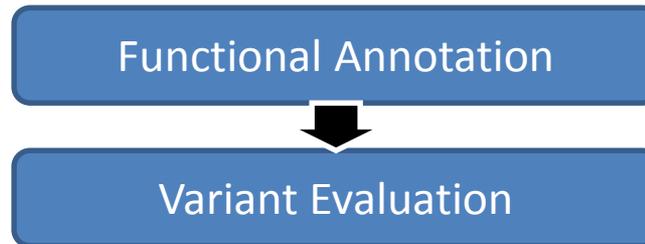


Preliminary Analysis



- ❖ Prerequisites: ANNOVAR (<http://www.openbioinformatics.org/annovar/>)
- 1. Utilizing update-to-date information to functionally annotate genetic variants detected from diverse genomes (including human genome hg18, hg19, as well as mouse, worm, fly, yeast and many others)
 - Actions:
 - 1) `convert2annovar.pl -format vcf4old merged_818subjects.vcf > merged_815subjects.avinput`
 - 2) `table_annovar.pl merged_818subjects.avinput humandb/ -buildver hg19 -out ADNI_WGS_818Subjects -remove -protocol refGene,phastConsElements46way,genomicSuperDups,esp6500si_all,1000g2012apr_all,snp135,ljb2_all -operation g,r,r,f,f,f -nastring NA -csvout`

Preliminary Analysis



❖ Prerequisites: GATK, PLINK

1. General-purpose tool for variant evaluation (% in dnSNP, genotype concordance, Ti/Tv ratios , and a lot more)

➤ Actions:

- 1) `java -Xmx6g -jar GATK -T VariantEval -R Reference -nt #OfCPUs --eval merged_818subjects.vcf --dbsnp DBSNP -o merged_818subjects.gatkreport`
- 2) Comparing SNPs from sequencing and SNPs from genotyping if any

Primary Analysis

- Common Variants ($MAF \geq 0.05$)
 - PLINK (<http://pngu.mgh.harvard.edu/~purcell/plink/>)
- Rare Variants ($MAF < 0.05$): gene-based analysis
 - SKAT-O
(<ftp://cran.r-project.org/pub/R/web/packages/SKAT/>)
 - Variants was assigned to genes based on annotation

SKAT-O

```
>install.packages("SKAT")
>library(SKAT)

>setwd("/net/scratch1/PARSED_CHR19_WGS/Extract_SNVs/RELN")

>Project.BED="merged_RELN_mafLT005_final_Nonsyn.bed"
>Project.BIM="merged_RELN_mafLT005_final_Nonsyn.bim"
>Project.FAM="merged_RELN_mafLT005_final_Nonsyn.fam"
>Project.SetID="merged_RELN_mafLT005_final_Nonsyn.SetID"
>Project.SSD="merged_RELN_mafLT005_final_Nonsyn.SSD"
>Project.Info="merged_RELN_mafLT005_final_Nonsyn.SSD.info"

>Generate_SSD_SetID(Project.BED,Project.BIM,Project.FAM,Project.SetID,Project.SSD,Project.Info)
```

Check duplicated SNPs in each SNP set

No duplicate

757 Samples, 1 Sets, 48 Total SNPs

[1] "SSD and Info files are created!"

>

```
> SSD.INFO=Open_SSD(Project.SSD,Project.Info)
```

757 Samples, 1 Sets, 48 Total SNPs

Open the SSD file

SKAT-O

```
knho@login1: awk '{print "RELN", $2}' merged_RELN_mafLT005_final_Nonsyn.bim >  
merged_RELN_mafLT005_final_Nonsyn.SetID
```

```
knho@login1: awk '{print $2}' merged_RELN_mafLT005_final_Nonsyn.bim >  
merged_RELN_mafLT005_final_Nonsyn.SSD
```

SKAT-O

```
>Project.Cov="ADNI_AV45_pheno_AV45_Global_CBL_final_110613_Knho.txt"
```

```
>Project_Cov=Read_Plink_FAM_Cov(Project.FAM,Project.Cov,ls.binary=TRUE)
```

```
>
```

```
>y=Project_Cov$AV45_Global_CBL
```

```
>x1=Project_Cov$PTGENDER
```

```
>x2=Project_Cov$Age_PET
```

```
>
```

```
> obj=SKAT_Null_Model(y~x1+x2,out_type="C")
```

Warning message:

110 samples have either missing phenotype or missing covariates. They are excluded from the analysis!

SKAT-O

```
>out=SKAT.SSD.All(SSD.INFO,obj,method="optimal.adj")
Warning message:
5 SNPs with either high missing rates or no-variation are excluded!
> out
$results
  SetID  P.value N.Marker.All N.Marker.Test
1 RELN 0.0050259      48      43

$P.value.Resampling
NULL

attr("class")
[1] "SKAT_SSD_ALL"
```