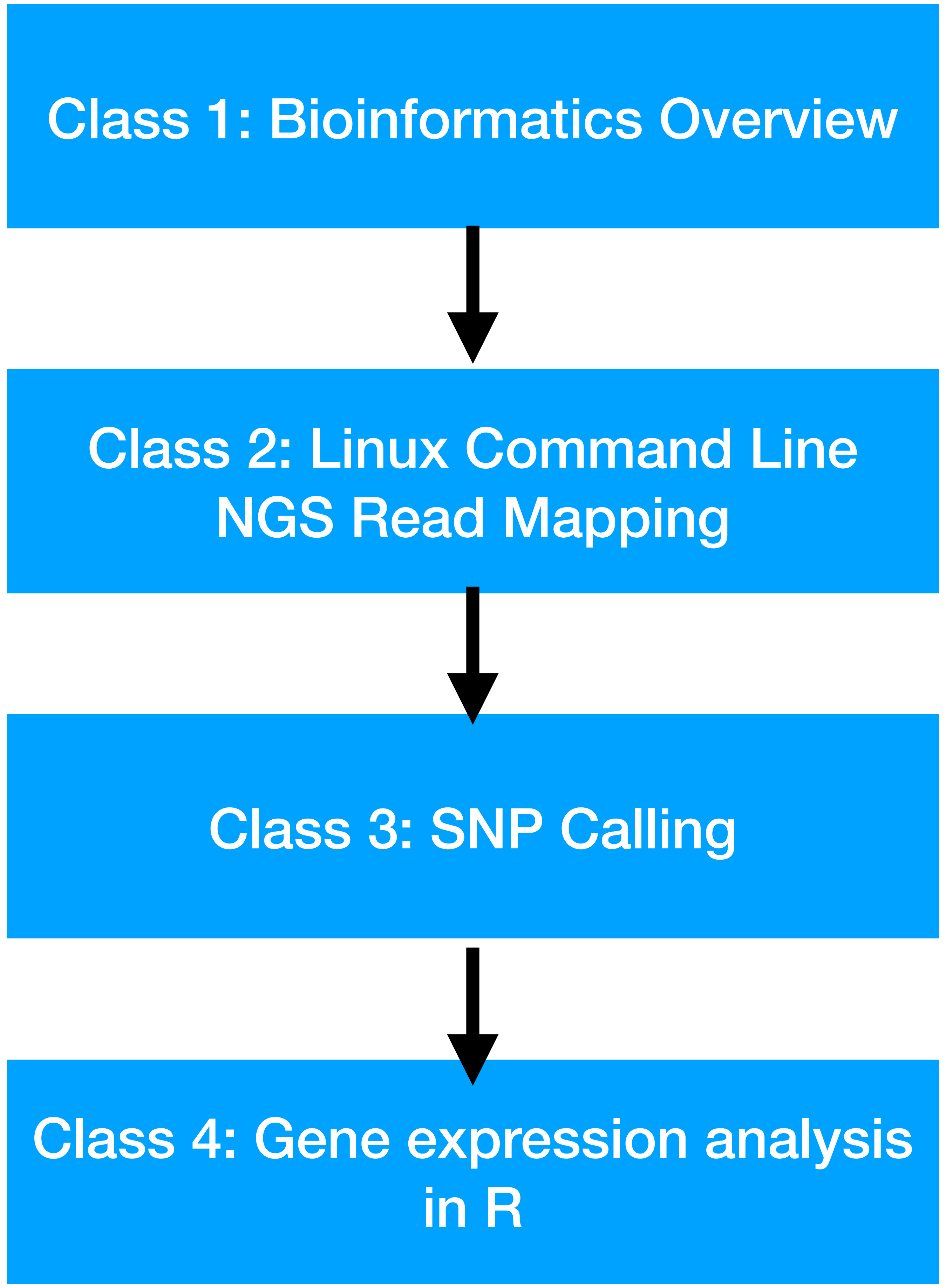
Bioinfo Class 3:
SNP Calling
7/12/23

Course Overview



Class 1: Bioinformatics Overview

Class 2: Linux Command Line
NGS Read Mapping

Class 3: SNP Calling

Class 4: Gene expression analysis in R

What did we do last time?

- Mapped NGS data from *S. pimpinellifolium* to *S. lycopersicum* with hisat2
- Ran stringtie to get a counts table for a future class
- Used a shell script, learned about loops and variables

Connect to remote server ‘thompson’

- Open terminal or Putty and type your username and password. Example (this is NOT your username):

```
ssh bioinfo0@thompson.sgn.cornell.edu
```

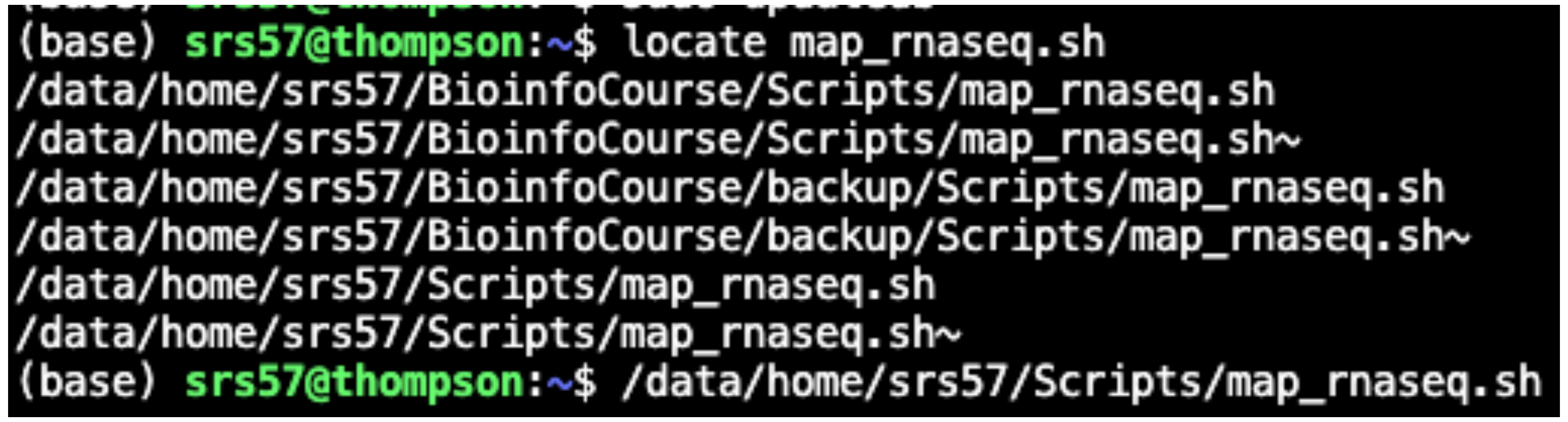
- #The password is bioinfo00

So we mapped one file, how do we map all the files?

- Make the computer work for you - Shell scripts and loops!
- **Exercise 6:** run the shell script called:
  - map_rnaseq.sh.

So we mapped one file, how do we map all the files?

- Make the computer work for you - Shell scripts and loops!
- **Exercise 6:** run the shell script called:
    - map_rnaseq.sh.



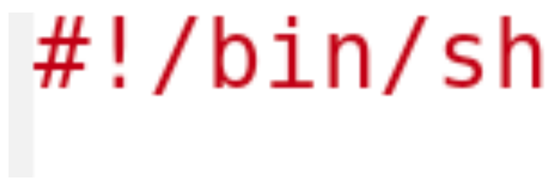
```
(base) srs57@thompson:~$ sudo updatedb
(base) srs57@thompson:~$ locate map_rnaseq.sh
/data/home/srs57/BioinfoCourse/Scripts/map_rnaseq.sh
/data/home/srs57/BioinfoCourse/Scripts/map_rnaseq.sh~
/data/home/srs57/BioinfoCourse/backup/Scripts/map_rnaseq.sh
/data/home/srs57/BioinfoCourse/backup/Scripts/map_rnaseq.sh~
/data/home/srs57/Scripts/map_rnaseq.sh
/data/home/srs57/Scripts/map_rnaseq.sh~
(base) srs57@thompson:~$ /data/home/srs57/Scripts/map_rnaseq.sh
```

Shell scripting

- map_rnaseq.sh
- What is this file?
- Bash shell script - Linux commands in a text file, runs line by line.
- Let’s check it out with less!

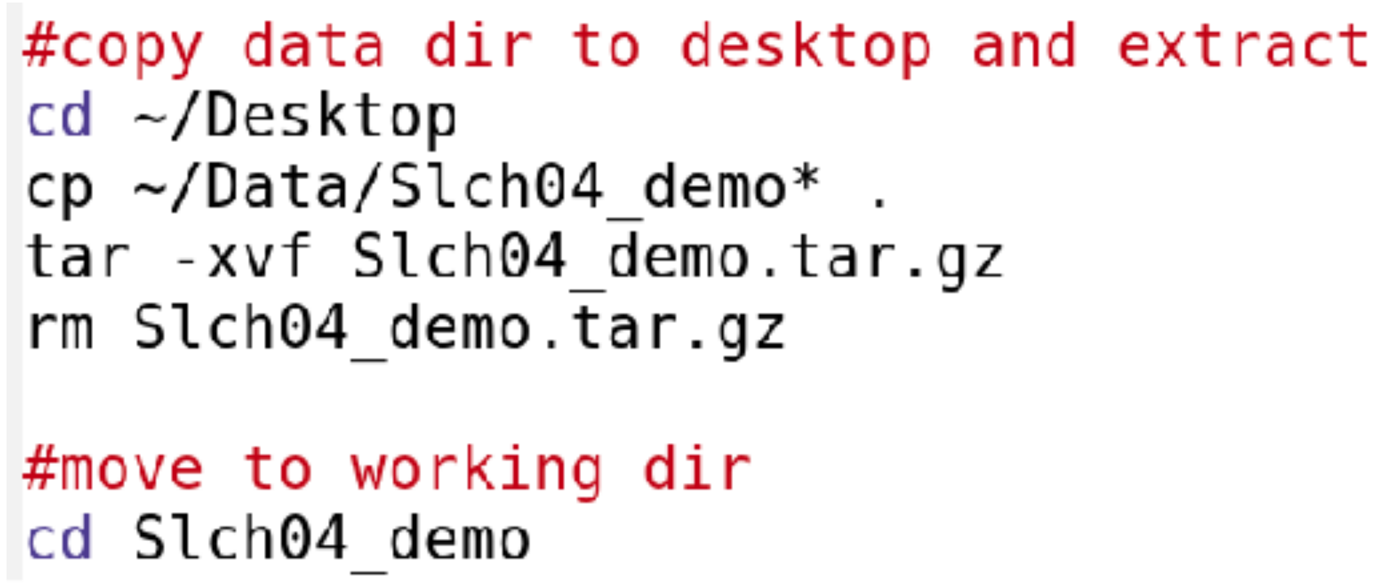
Shell scripting

- Shebang



```
#!/bin/sh
```

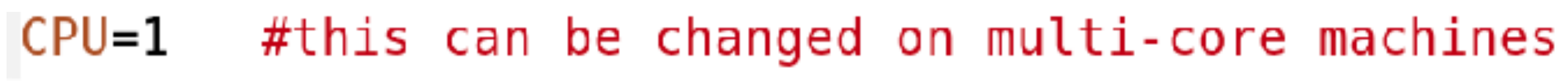
- Move files where they need to be



```
#copy data dir to desktop and extract
cd ~/Desktop
cp ~/Data/Slch04_demo* .
tar -xvf Slch04_demo.tar.gz
rm Slch04_demo.tar.gz

#move to working dir
cd Slch04_demo
```

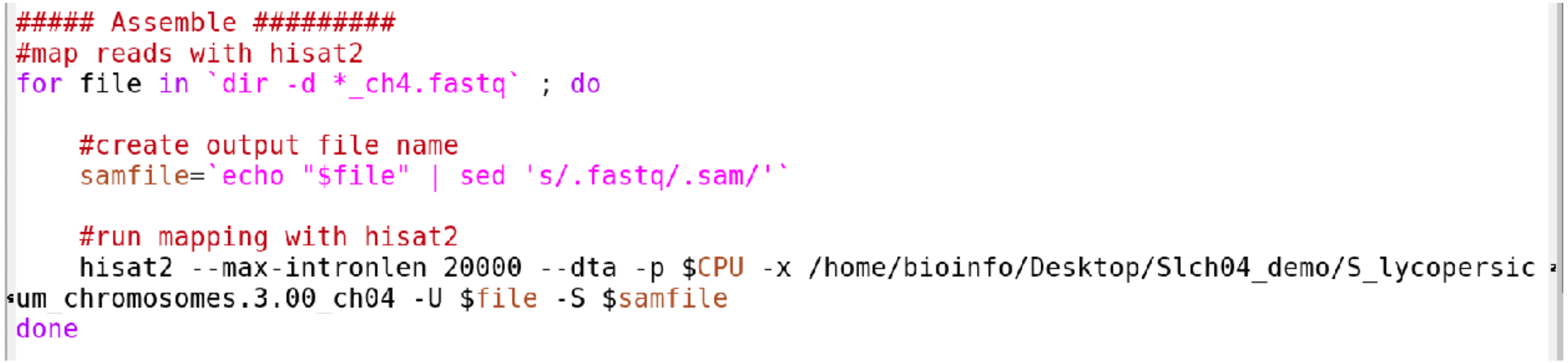
Shell scripting

- Increase cores option



```
CPU=1    #this can be changed on multi-core machines
```

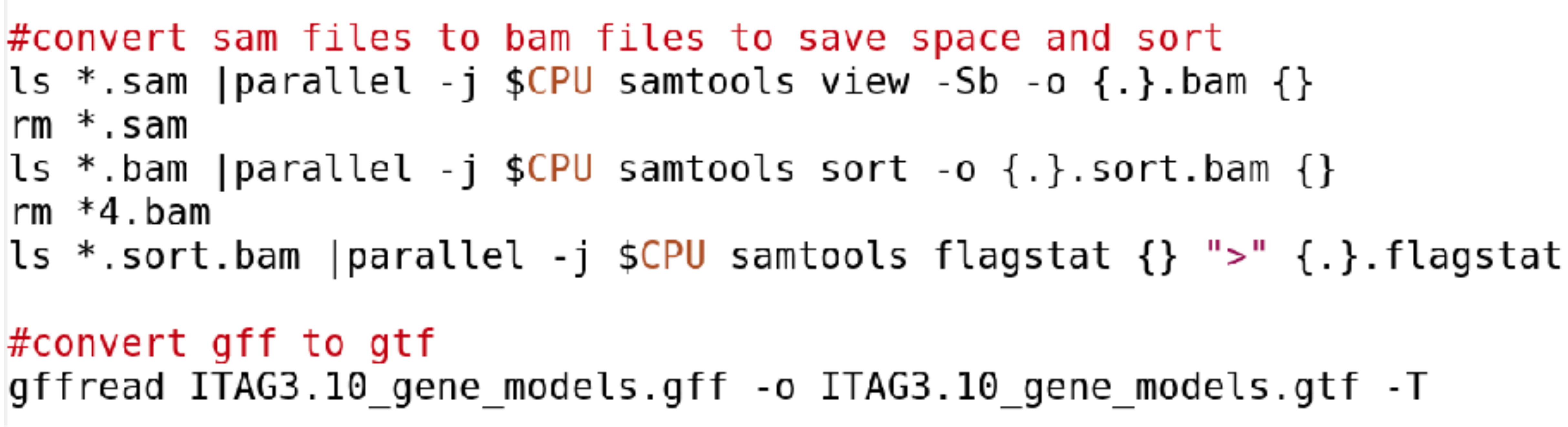
- Assemble with a loop



```
##### Assemble #########
#map reads with hisat2
for file in `dir -d *_ch4.fastq` ; do

    #create output file name
    samfile=`echo "$file" | sed 's/.fastq/.sam/'`

    #run mapping with hisat2
    hisat2 --max-intronlen 20000 --dta -p $CPU -x /home/bioinfo/Desktop/Slch04_demo/S_lycopersic
um_chromosomes.3.00_ch04 -U $file -S $samfile
done
```

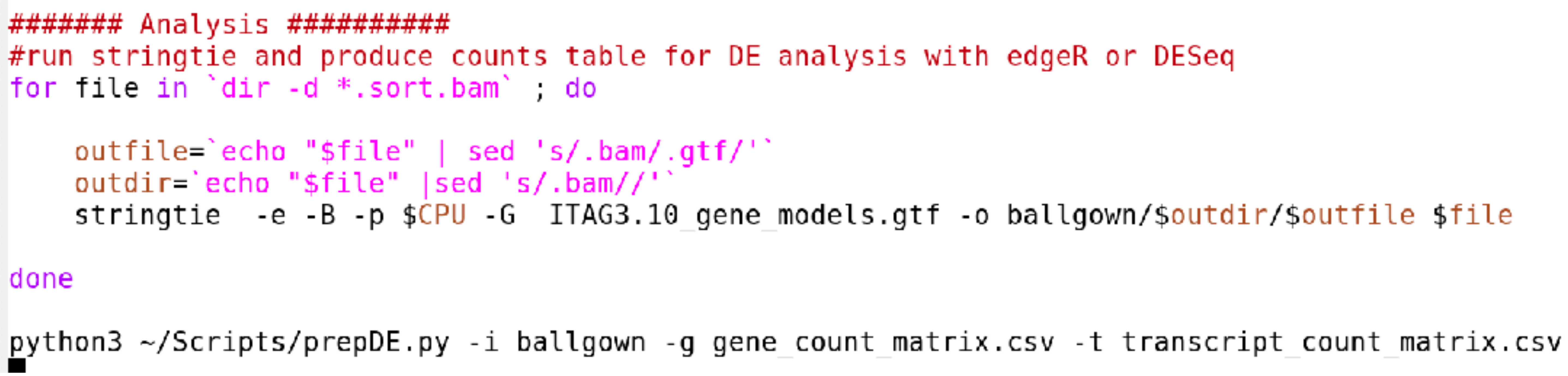
Shell scripting

- File format conversions



```
#convert sam files to bam files to save space and sort
ls *.sam |parallel -j $CPU samtools view -Sb -o {.}.bam {}
rm *.sam
ls *.bam |parallel -j $CPU samtools sort -o {.}.sort.bam {}
rm *4.bam
ls *.sort.bam |parallel -j $CPU samtools flagstat {} ">" {.}.flagstat

#convert gff to gtf
gffread ITAG3.10_gene_models.gff -o ITAG3.10_gene_models.gtf -T
```

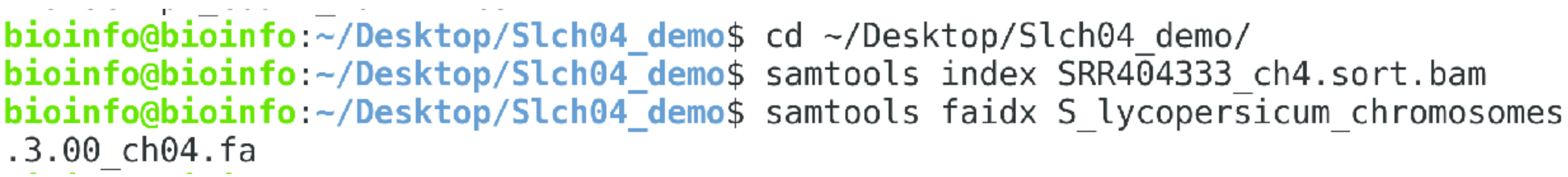
- Make a counts file for DE



```
####### Analysis ##########
#run stringtie and produce counts table for DE analysis with edgeR or DESeq
for file in `dir -d *.sort.bam` ; do

    outfile=`echo "$file" | sed 's/.bam/.gtf/'`
    outdir=`echo "$file" |sed 's/.bam//'`
    stringtie  -e -B -p $CPU -G  ITAG3.10_gene_models.gtf -o ballgown/$outdir/$outfile $file

done

python3 ~/Scripts/prepDE.py -i ballgown -g gene_count_matrix.csv -t transcript_count_matrix.csv
```

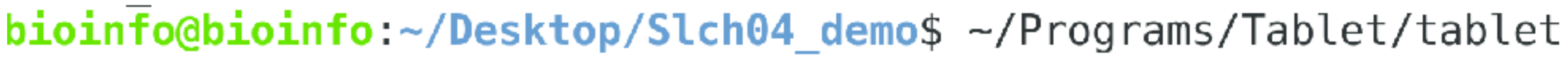
Exercise 7: Viewing bam files with Tablet

- Index the files



```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ cd ~/Desktop/Slch04_demo/
bioinfo@bioinfo:~/Desktop/Slch04_demo$ samtools index SRR404333_ch4.sort.bam
bioinfo@bioinfo:~/Desktop/Slch04_demo$ samtools faidx S_lycopersicum_chromosomes
.3.00_ch04.fa
```

- Load in tablet



```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ ~/Programs/Tablet/tablet
```

SNP Calling

Today's Objectives:

- Learn how to use bam files (read mapping files) from the last class to identify SNPs and indels between a cultivated and a wild tomato species
- Identify the effect of the SNPs on coding regions, UTRs, etc

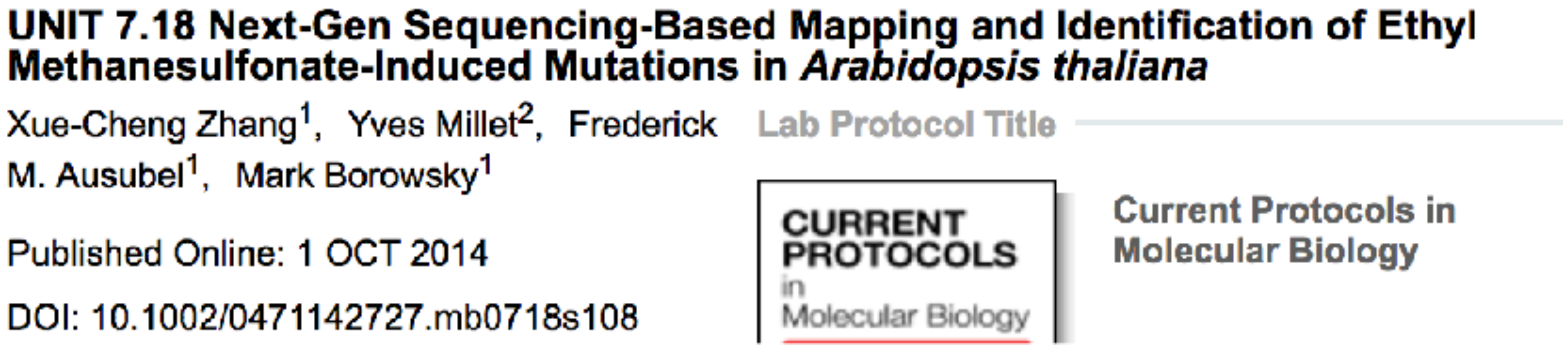
Why Call SNPs?

> Current Issue > vol. 108 no. 17 > Jesse A. Poland, 6893–6898, doi: 10.1073/pnas.1010894108

Check for updates

Genome-wide nested association mapping of quantitative resistance to northern leaf blight in maize

Jesse A. Poland[a,1], Peter J. Bradbury[a,b], Edward S. Buckler[a,b], and Rebecca J. Nelson[a,c,2]



UNIT 7.18 Next-Gen Sequencing-Based Mapping and Identification of Ethyl Methanesulfonate-Induced Mutations in *Arabidopsis thaliana*

Xue-Cheng Zhang[1], Yves Millet[2], Frederick M. Ausubel[1], Mark Borowsky[1]

Published Online: 1 OCT 2014

DOI: 10.1002/0471142727.mb0718s108

Lab Protocol Title

CURRENT PROTOCOLS in Molecular Biology

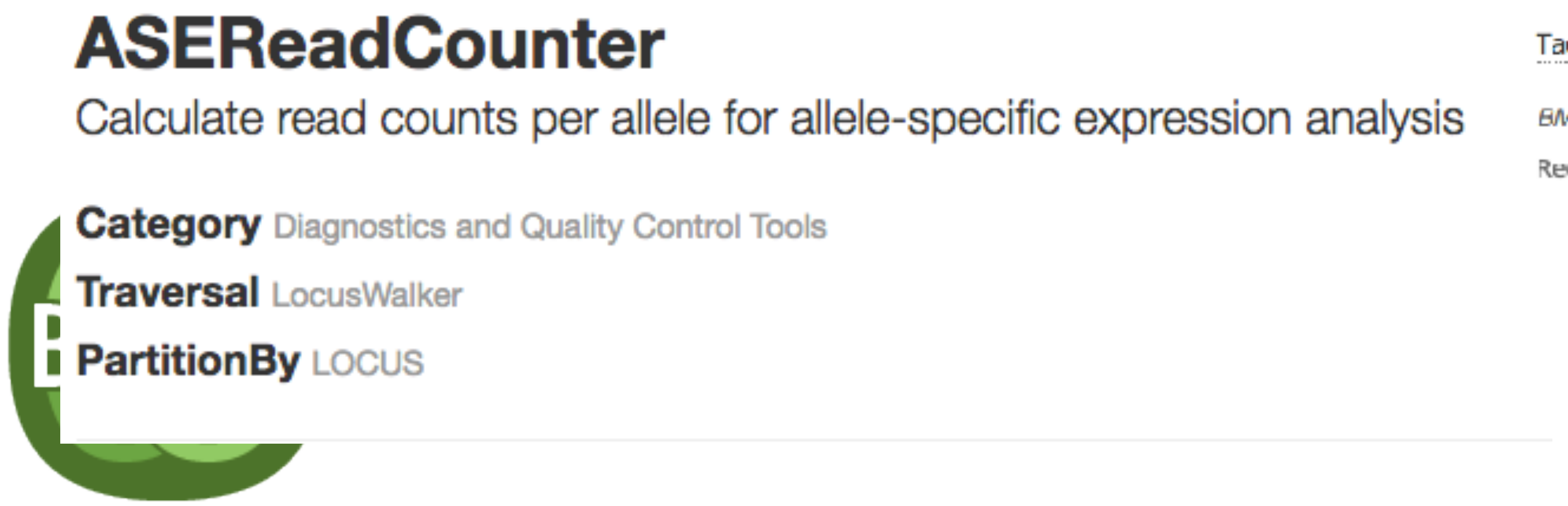
Current Protocols in Molecular Biology

SNPhylo: a pipeline to construct a phylogenetic tree from huge SNP data

Tae-Ho Lee, Hui Guo, Xiyin Wang, Changsoo Kim and Andrew H Paterson

BMC Genomics 2014 15:162 | DOI: 10.1186/1471-2164-15-162 | © Lee et al.; licensee BioMed Central Ltd. 2014

Received: 25 September 2013 | Accepted: 18 February 2014 | Published: 26 February 2014



ASEReadCounter

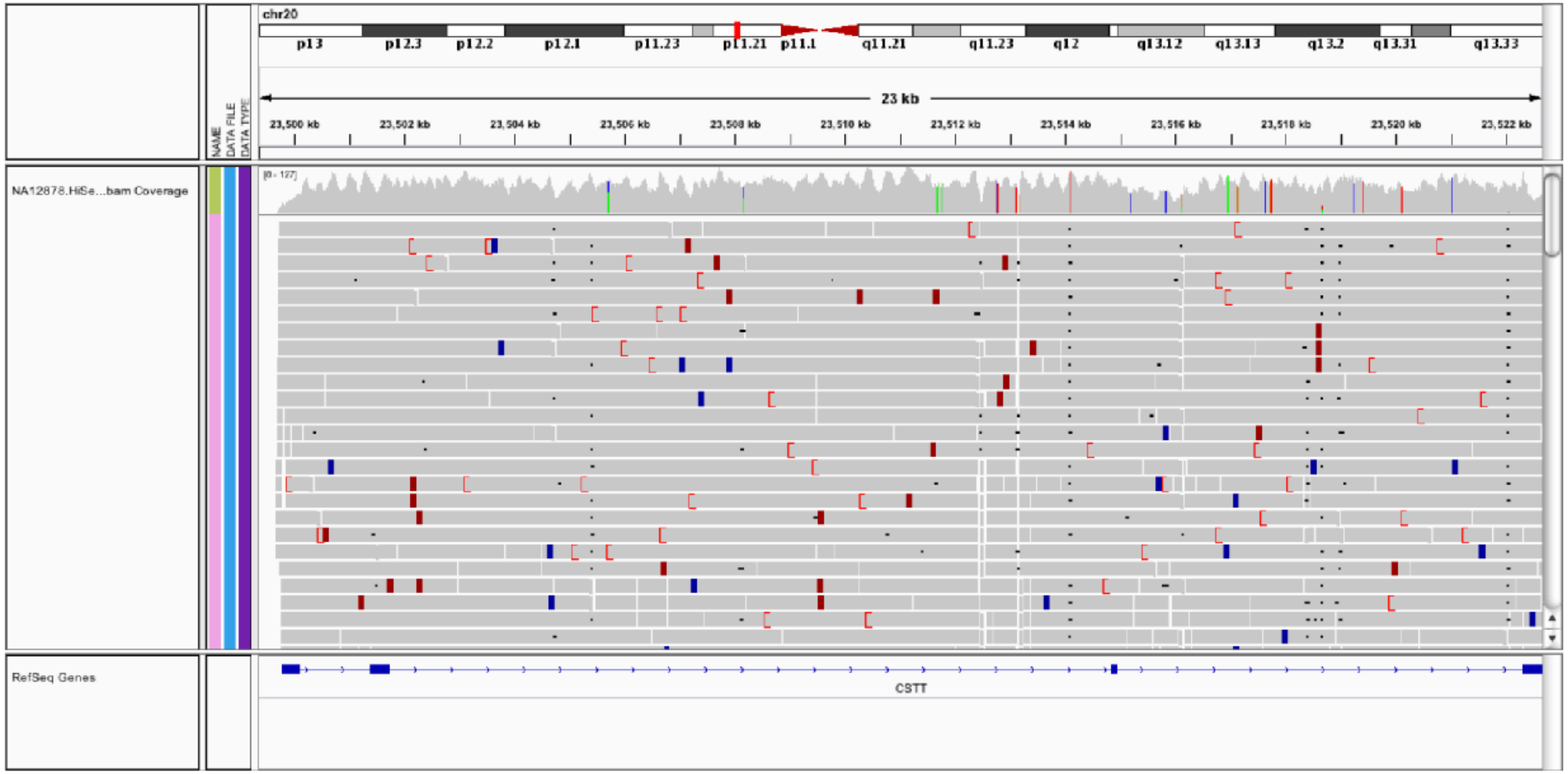
Calculate read counts per allele for allele-specific expression analysis

Category Diagnostics and Quality Control Tools

Traversal LocusWalker

PartitionBy LOCUS

Which mismatches are real mutations and which are noise/error?



https://www.broadinstitute.org/gatk/events/3391/GATKw1310-BP-0A-Intro_to_NGS.pdf

Which SNP caller to use?

Several possible considerations:

1. Input/Output Formats
2. Run Time
3. Quality Awareness
4. Sensitivity and Artifacts

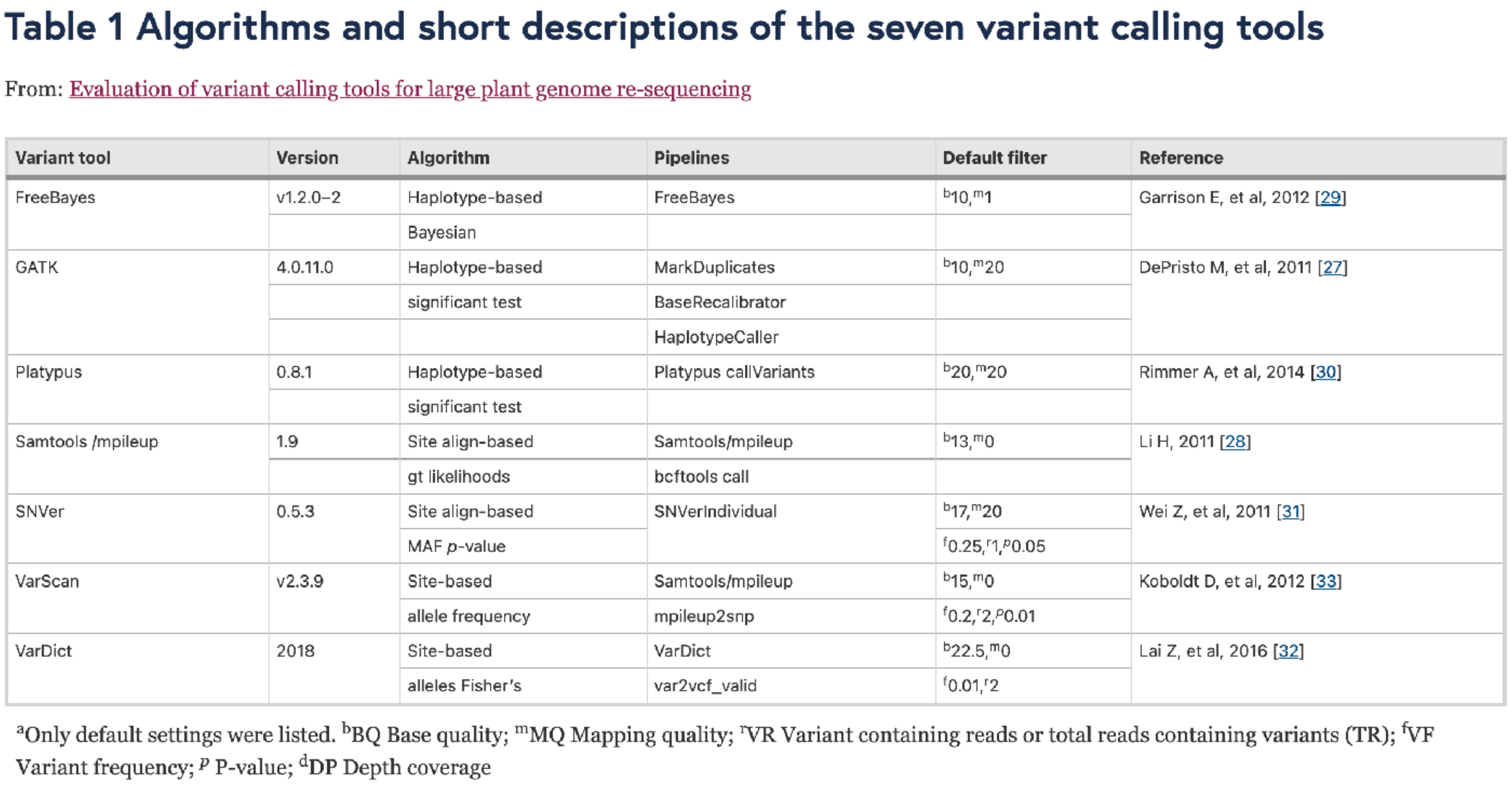


Table 1 Algorithms and short descriptions of the seven variant calling tools

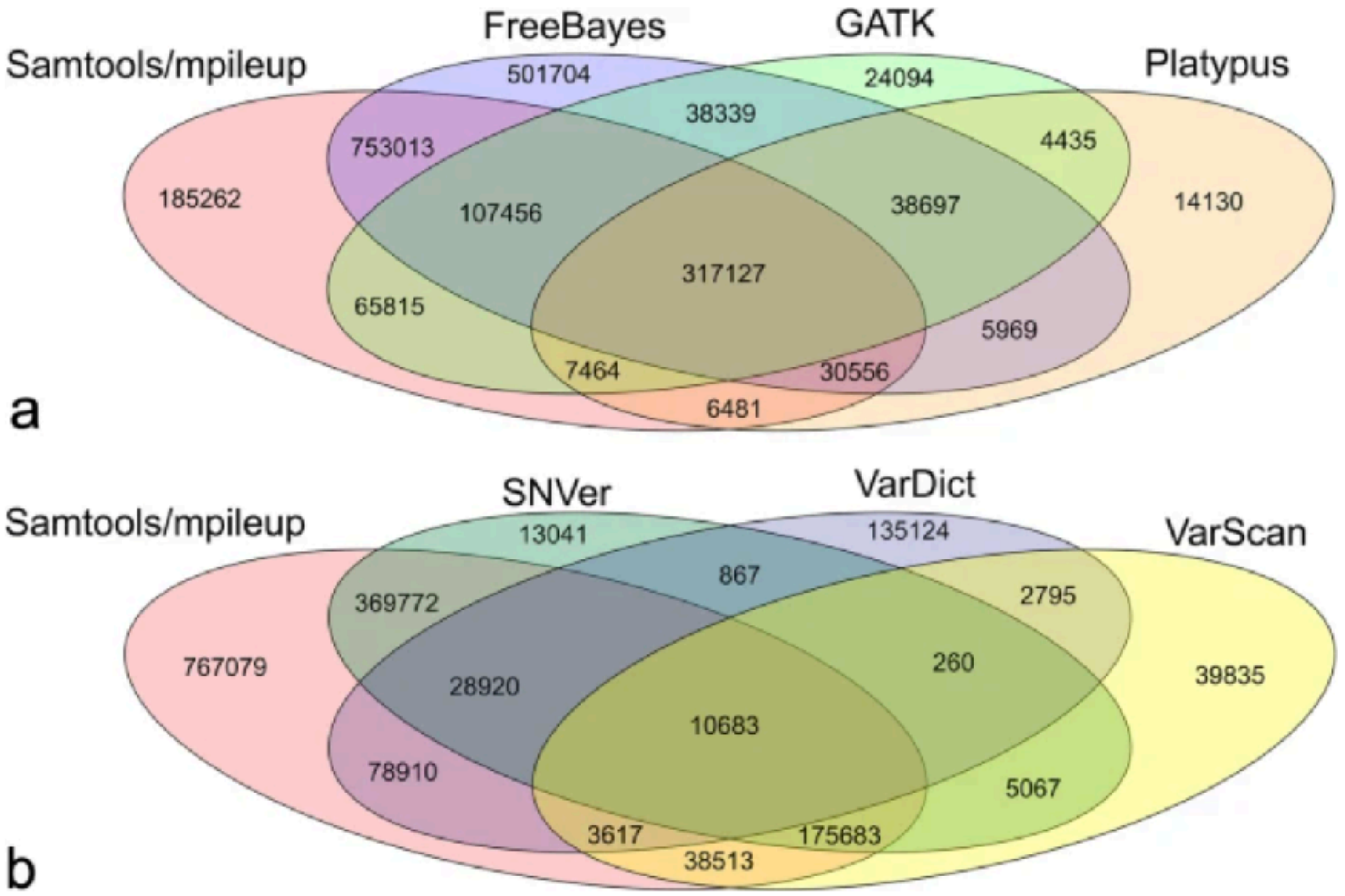
From: Evaluation of variant calling tools for large plant genome re-sequencing

Variant tool	Version	Algorithm	Pipelines	Default filter	Reference
FreeBayes	v1.2.0–2	Haplotype-based	FreeBayes	^{b}10,^{m}1	Garrison E, et al, 2012 [29]
		Bayesian			
GATK	4.0.11.0	Haplotype-based	MarkDuplicates	^{b}10,^{m}20	DePristo M, et al, 2011 [27]
		significant test	BaseRecalibrator		
			HaplotypeCaller		
Platypus	0.8.1	Haplotype-based	Platypus callVariants	^{b}20,^{m}20	Rimmer A, et al, 2014 [30]
		significant test			
Samtools /mpileup	1.9	Site align-based	Samtools/mpileup	^{b}13,^{m}0	Li H, 2011 [28]
		gt likelihoods	bcftools call		
SNVer	0.5.3	Site align-based	SNVerIndividual	^{b}17,^{m}20	Wei Z, et al, 2011 [31]
		MAF *p*-value		^{f}0.25,^{r}1,^{p}0.05	
VarScan	v2.3.9	Site-based	Samtools/mpileup	^{b}15,^{m}0	Koboldt D, et al, 2012 [33]
		allele frequency	mpileup2snp	^{f}0.2,^{r}2,^{p}0.01	
VarDict	2018	Site-based	VarDict	^{b}22.5,^{m}0	Lai Z, et al, 2016 [32]
		alleles Fisher's	var2vcf_valid	^{f}0.01,^{r}2	

^{a}Only default settings were listed. ^{b}BQ Base quality; ^{m}MQ Mapping quality; ^{r}VR Variant containing reads or total reads containing variants (TR); ^{f}VF Variant frequency; ^{p} P-value; ^{d}DP Depth coverage

Yao, Z., You, F.M., N'Diaye, A. *et al.* Evaluation of variant calling tools for large plant genome re-sequencing. *BMC Bioinformatics* **21,** 360 (202
10.1186/s12859-020-03704-1

Fig. 3



Venn diagrams for variant calling tool comparison. SNP variants were called using different variant calling tools and filtered through the same stringent filtering criteria. The numbers of overlap and unique SNP loci were displayed. **a**. Samtools/mpileup compared with FreeBayes, GATK, and Platypus. **b** Samtools/mpileup compared with SNVer, VarDict, and VarScan

Yao, Z., You, F.M., N'Diaye, A. *et al.* Evaluation of variant calling tools for large plant genome re-sequencing. *BMC Bioinformatics* **21,** 360 (202
10.1186/s12859-020-03704-1

SNP Calling Using GATK (HaplotypeCaller)

Today’s Exercises!

Exercise 1

Run GATK and samtools on your *S. pimpinellifolium* to *S. lycopersicum* mapping files from last week. Then compare the results.

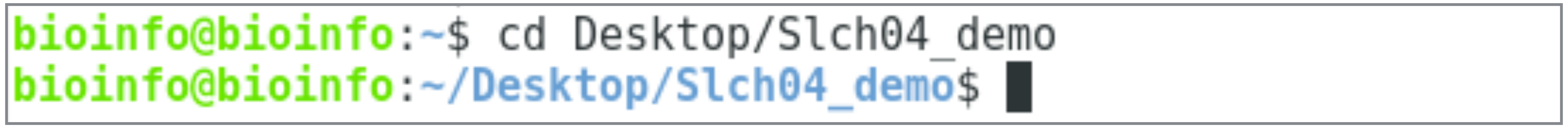
You can run_snpcalling.sh instead of executing individual commands by hand.

Make a directory in ~/Desktop/Slch04_demo called variants to keep the results in

- Merge all bam files into one file and sort (samtools merge and samtools sort)
- Mark duplicate reads from the sorted bam file (Picard MarkDuplicates)
- Add read groups (Picard AddOrReplaceReadGroups)
- Create a sequence dictionary (Picard CreateSequenceDictionary) and Index the bam file output (samtools index)
- Find regions for local realignment around indels
- Call raw variants GATK (HaplotypeCaller)
- Call variants with samtools

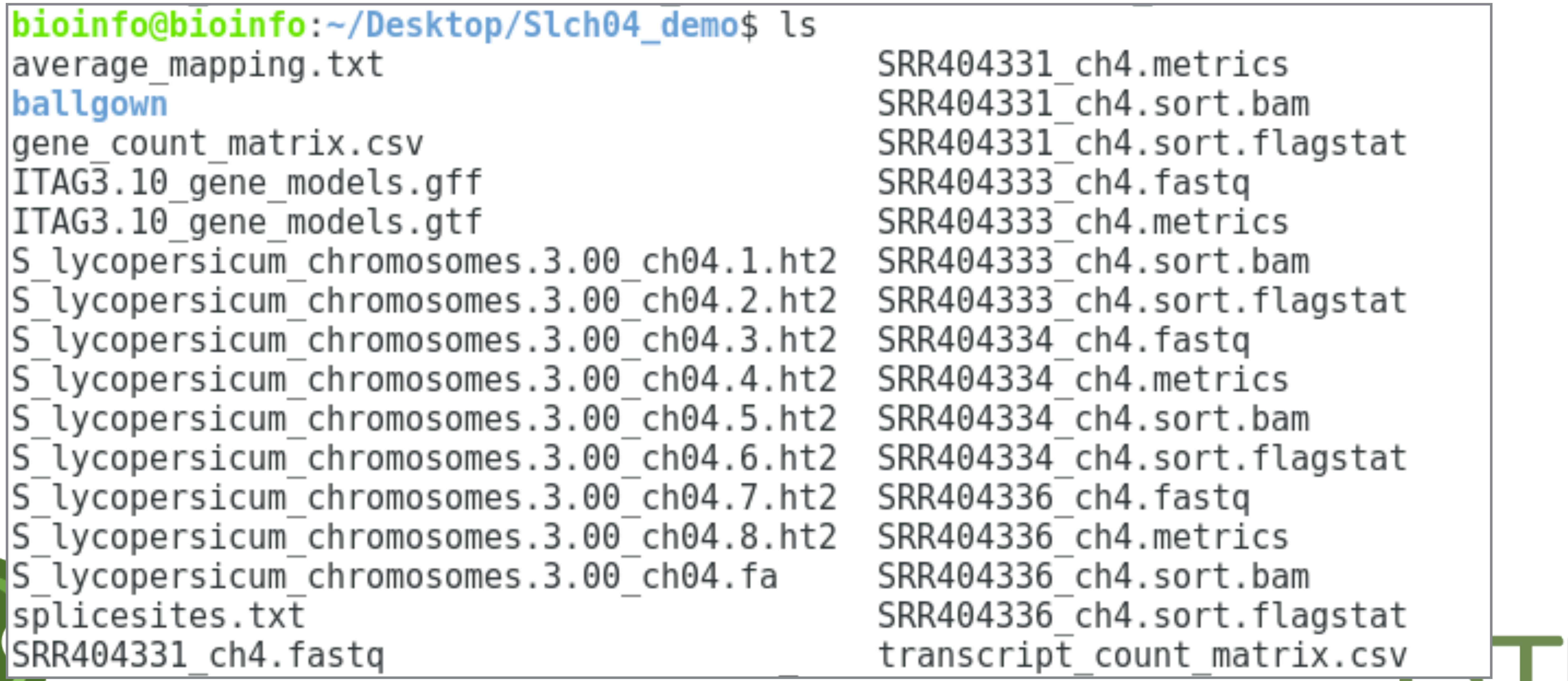
Solution: You can 'cheat' by looking at the contents of run_snpcalling.sh (or by running it).

1. Use **cd** to change directory to the folder we were using last week:



```
bioinfo@bioinfo:~$ cd Desktop/Slch04_demo
bioinfo@bioinfo:~/Desktop/Slch04_demo$ 
```

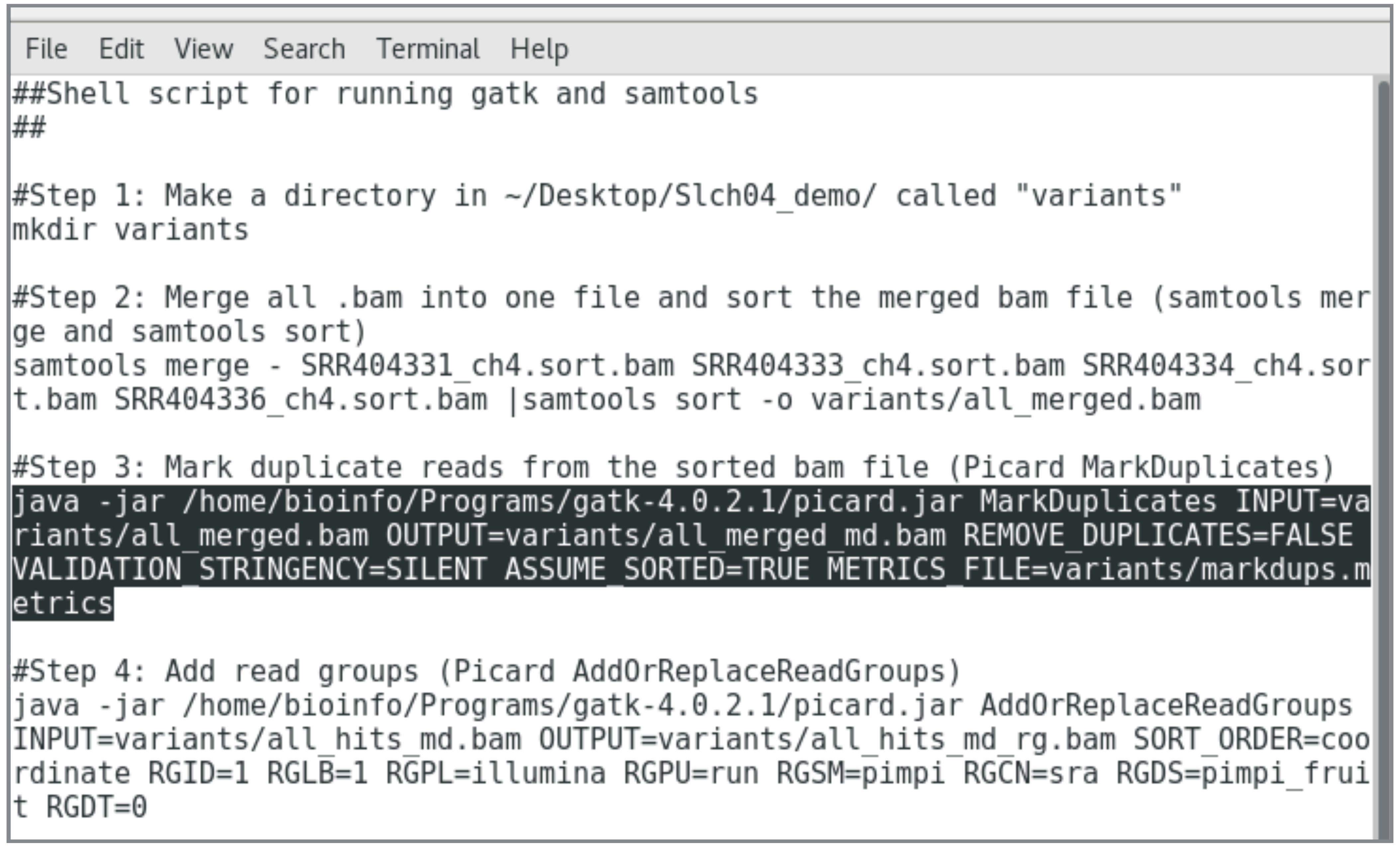
2. Use **ls** to check that you have the necessary files:



```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ ls
average_mapping.txt                        SRR404331_ch4.metrics
ballgown                                   SRR404331_ch4.sort.bam
gene_count_matrix.csv                      SRR404331_ch4.sort.flagstat
ITAG3.10_gene_models.gff                   SRR404333_ch4.fastq
ITAG3.10_gene_models.gtf                   SRR404333_ch4.metrics
S_lycopersicum_chromosomes.3.00_ch04.1.ht2  SRR404333_ch4.sort.bam
S_lycopersicum_chromosomes.3.00_ch04.2.ht2  SRR404333_ch4.sort.flagstat
S_lycopersicum_chromosomes.3.00_ch04.3.ht2  SRR404334_ch4.fastq
S_lycopersicum_chromosomes.3.00_ch04.4.ht2  SRR404334_ch4.metrics
S_lycopersicum_chromosomes.3.00_ch04.5.ht2  SRR404334_ch4.sort.bam
S_lycopersicum_chromosomes.3.00_ch04.6.ht2  SRR404334_ch4.sort.flagstat
S_lycopersicum_chromosomes.3.00_ch04.7.ht2  SRR404336_ch4.fastq
S_lycopersicum_chromosomes.3.00_ch04.8.ht2  SRR404336_ch4.metrics
S_lycopersicum_chromosomes.3.00_ch04.fa     SRR404336_ch4.sort.bam
splicesites.txt                            SRR404336_ch4.sort.flagstat
SRR404331_ch4.fastq                        transcript_count_matrix.csv
```

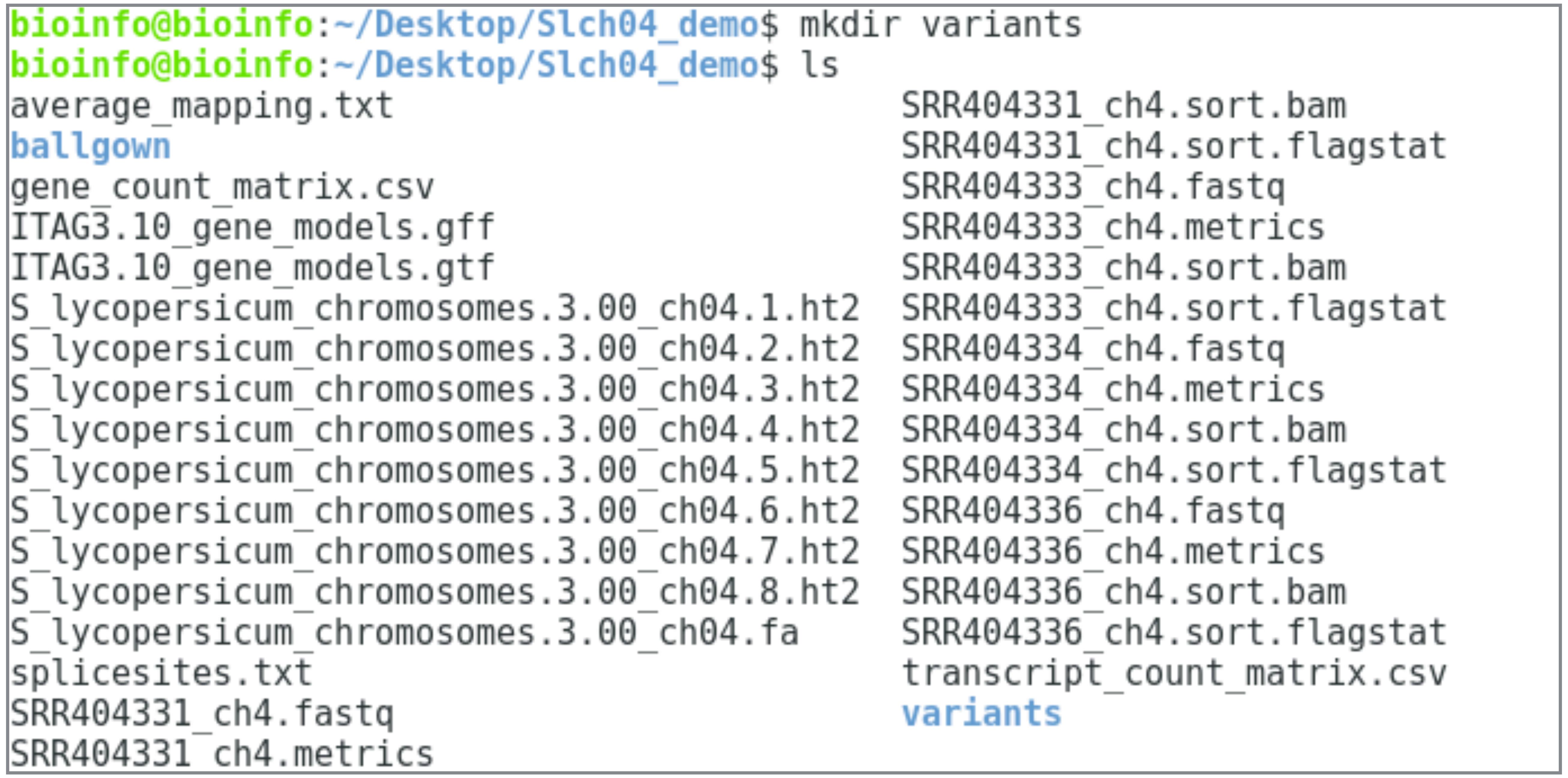
3. **Locate** the SNP calling script (run_snpcalling.sh):

4. Try **less** to look inside the script. Note that the script is just a file containing a collection of commands.



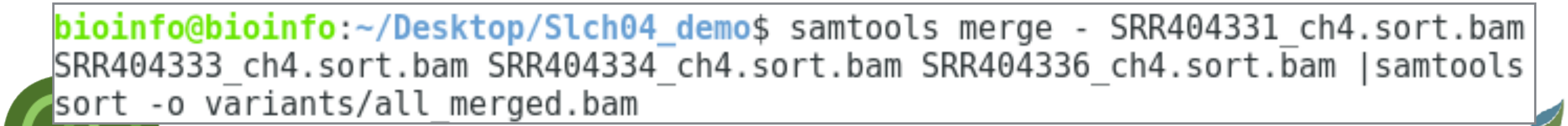
```
##Shell script for running gatk and samtools
##

#Step 1: Make a directory in ~/Desktop/Slch04_demo/ called "variants"
mkdir variants

#Step 2: Merge all .bam into one file and sort the merged bam file (samtools mer
ge and samtools sort)
samtools merge - SRR404331_ch4.sort.bam SRR404333_ch4.sort.bam SRR404334_ch4.sor
t.bam SRR404336_ch4.sort.bam |samtools sort -o variants/all_merged.bam

#Step 3: Mark duplicate reads from the sorted bam file (Picard MarkDuplicates)
java -jar /home/bioinfo/Programs/gatk-4.0.2.1/picard.jar MarkDuplicates INPUT=va
riants/all_merged.bam OUTPUT=variants/all_merged_md.bam REMOVE_DUPLICATES=FALSE
VALIDATION_STRINGENCY=SILENT ASSUME_SORTED=TRUE METRICS_FILE=variants/markdups.m
etrics

#Step 4: Add read groups (Picard AddOrReplaceReadGroups)
java -jar /home/bioinfo/Programs/gatk-4.0.2.1/picard.jar AddOrReplaceReadGroups
INPUT=variants/all_hits_md.bam OUTPUT=variants/all_hits_md_rg.bam SORT_ORDER=coo
rdinate RGID=1 RGLB=1 RGPL=illumina RGPU=run RGSM=pimpi RGCN=sra RGDS=pimpi_frui
t RGDT=0
```

5. Use **mkdir** to make a directory called ‘variants’:



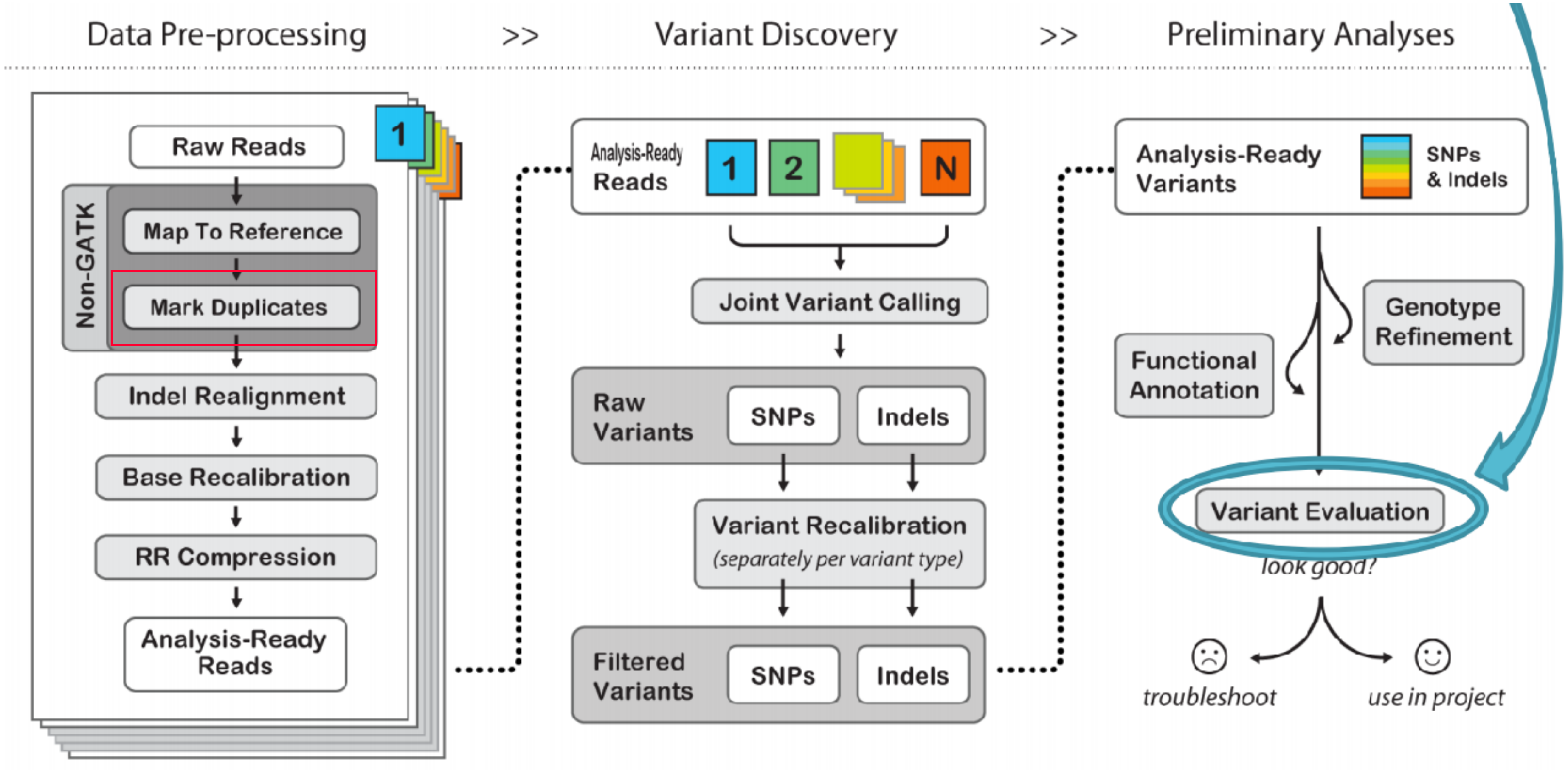
```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ mkdir variants
bioinfo@bioinfo:~/Desktop/Slch04_demo$ ls
average_mapping.txt                         SRR404331_ch4.sort.bam
ballgown                                    SRR404331_ch4.sort.flagstat
gene_count_matrix.csv                       SRR404333_ch4.fastq
ITAG3.10_gene_models.gff                    SRR404333_ch4.metrics
ITAG3.10_gene_models.gtf                    SRR404333_ch4.sort.bam
S_lycopersicum_chromosomes.3.00_ch04.1.ht2  SRR404333_ch4.sort.flagstat
S_lycopersicum_chromosomes.3.00_ch04.2.ht2  SRR404334_ch4.fastq
S_lycopersicum_chromosomes.3.00_ch04.3.ht2  SRR404334_ch4.metrics
S_lycopersicum_chromosomes.3.00_ch04.4.ht2  SRR404334_ch4.sort.bam
S_lycopersicum_chromosomes.3.00_ch04.5.ht2  SRR404334_ch4.sort.flagstat
S_lycopersicum_chromosomes.3.00_ch04.6.ht2  SRR404336_ch4.fastq
S_lycopersicum_chromosomes.3.00_ch04.7.ht2  SRR404336_ch4.metrics
S_lycopersicum_chromosomes.3.00_ch04.8.ht2  SRR404336_ch4.sort.bam
S_lycopersicum_chromosomes.3.00_ch04.fa     SRR404336_ch4.sort.flagstat
splicesites.txt                             transcript_count_matrix.csv
SRR404331_ch4.fastq                         variants
SRR404331_ch4.metrics
```

6. Run **samtools merge** to merge together the .bam files from last week, then perform **samtools sort** to sort the new file.



```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ samtools merge - SRR404331_ch4.sort.bam
SRR404333_ch4.sort.bam SRR404334_ch4.sort.bam SRR404336_ch4.sort.bam |samtools
sort -o variants/all_merged.bam
```

GATK Pipeline

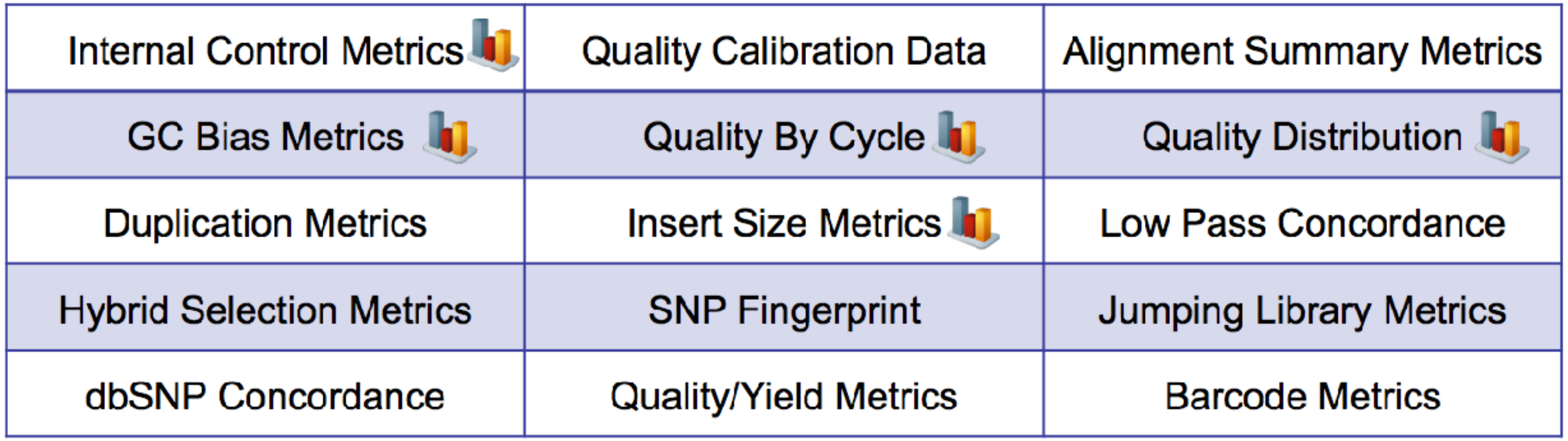


gatk

Picard

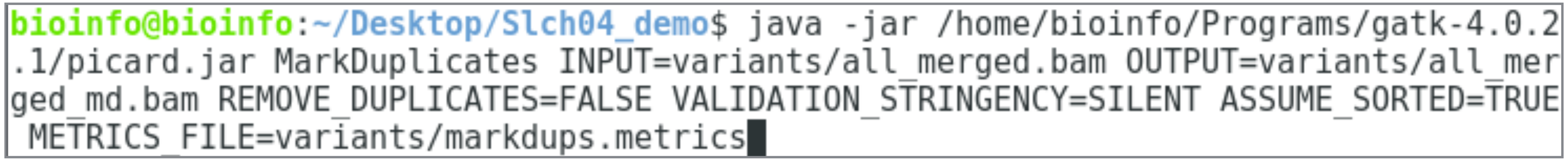
https://github.com/utgenome/picard

- Many tools including:
  - Duplicate read tagging/removal
  - Adding read group info



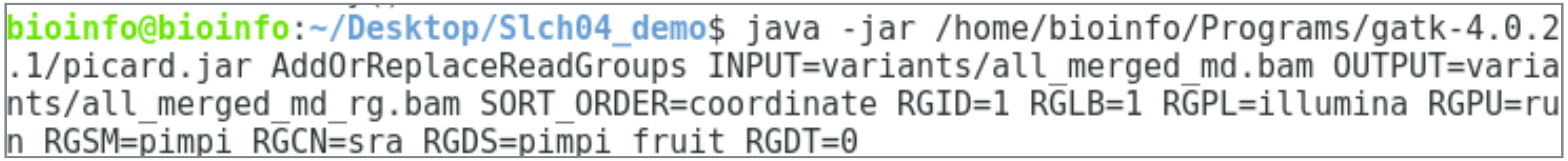
Internal Control Metrics 📊	Quality Calibration Data	Alignment Summary Metrics
GC Bias Metrics 📊	Quality By Cycle 📊	Quality Distribution 📊
Duplication Metrics	Insert Size Metrics 📊	Low Pass Concordance
Hybrid Selection Metrics	SNP Fingerprint	Jumping Library Metrics
dbSNP Concordance	Quality/Yield Metrics	Barcode Metrics

7. Run **MarkDuplicates**



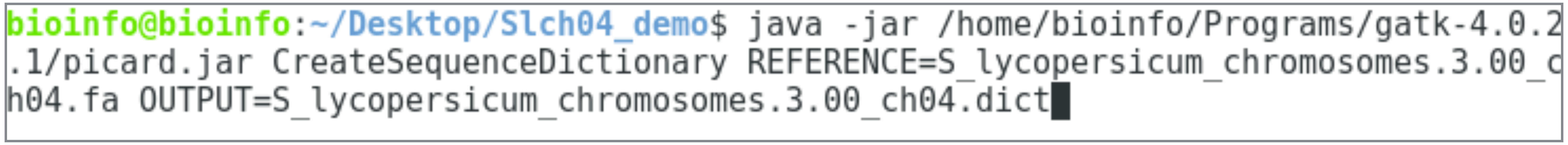
```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ java -jar /home/bioinfo/Programs/gatk-4.0.2
.1/picard.jar MarkDuplicates INPUT=variants/all_merged.bam OUTPUT=variants/all_mer
ged_md.bam REMOVE_DUPLICATES=FALSE VALIDATION_STRINGENCY=SILENT ASSUME_SORTED=TRUE
 METRICS_FILE=variants/markdups.metrics
```

8. Run **AddOrReplaceReadGroups**



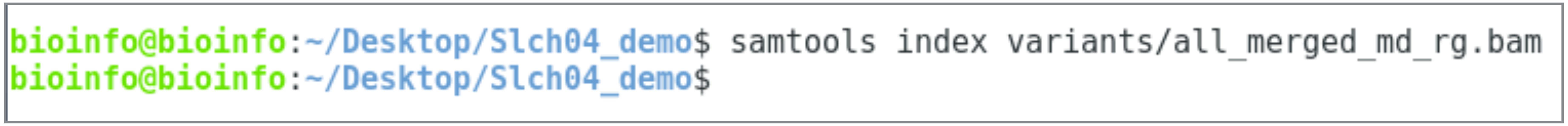
```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ java -jar /home/bioinfo/Programs/gatk-4.0.2
.1/picard.jar AddOrReplaceReadGroups INPUT=variants/all_merged_md.bam OUTPUT=varia
nts/all_merged_md_rg.bam SORT_ORDER=coordinate RGID=1 RGLB=1 RGPL=illumina RGPU=ru
n RGSM=pimpi RGCN=sra RGDS=pimpi_fruit RGDT=0
```

9. Run **CreateSequenceDictionary**



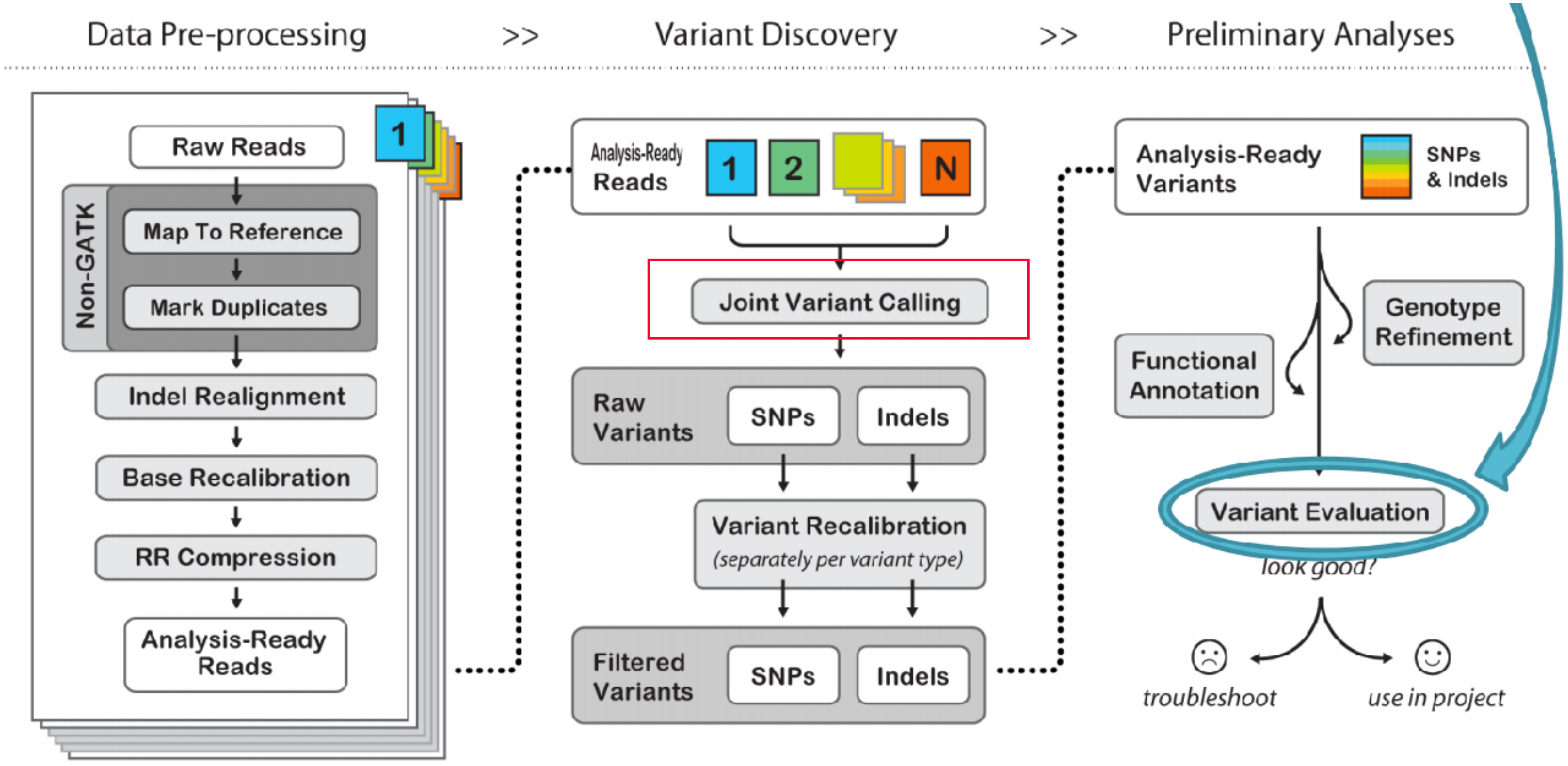
```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ java -jar /home/bioinfo/Programs/gatk-4.0.2
.1/picard.jar CreateSequenceDictionary REFERENCE=S_lycopersicum_chromosomes.3.00_c
h04.fa OUTPUT=S_lycopersicum_chromosomes.3.00_ch04.dict
```

10. Index the new merged .bam file with **samtools index**



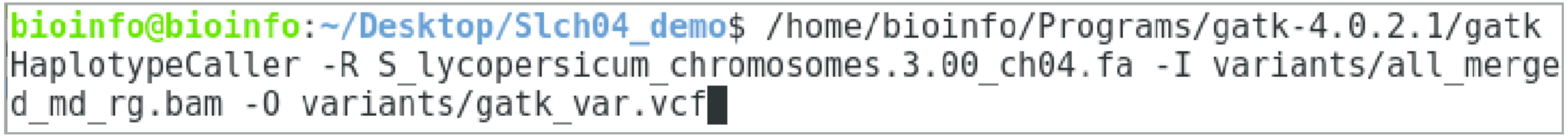
```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ samtools index variants/all_merged_md_rg.bam
bioinfo@bioinfo:~/Desktop/Slch04_demo$
```

GATK Pipeline



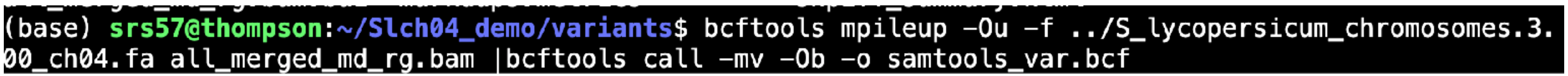
gatk

11. Run **HaplotypeCaller**.



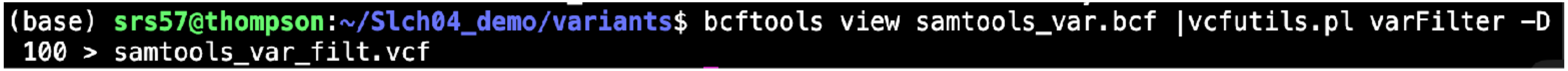
```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ /home/bioinfo/Programs/gatk-4.0.2.1/gatk
HaplotypeCaller -R S_lycopersicum_chromosomes.3.00_ch04.fa -I variants/all_merge
d_md_rg.bam -O variants/gatk_var.vcf
```

12. Now, for comparison, we will also call variants using **samtools/bcftools**:



```
(base) srs57@thompson:~/Slch04_demo/variants$ bcftools mpileup -Ou -f ../S_lycopersicum_chromosomes.3.
00_ch04.fa all_merged_md_rg.bam |bcftools call -mv -Ob -o samtools_var.bcf
```

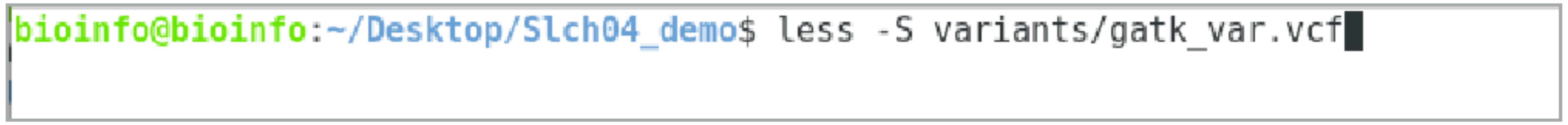
13. Convert samtools_var.bcf to .vcf and filter:



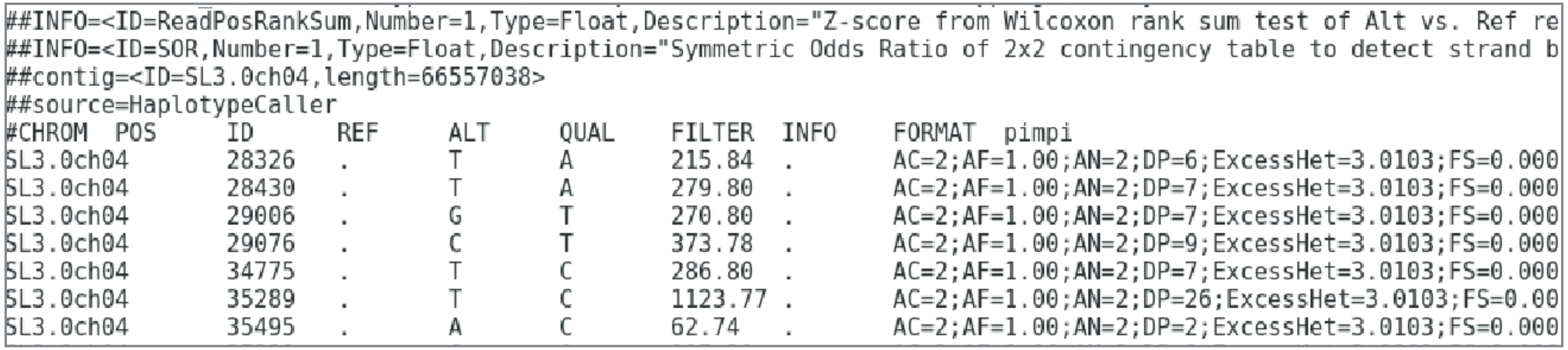
```
(base) srs57@thompson:~/Slch04_demo/variants$ bcftools view samtools_var.bcf |vcfutils.pl varFilter -D
 100 > samtools_var_filt.vcf
```

VCF Format

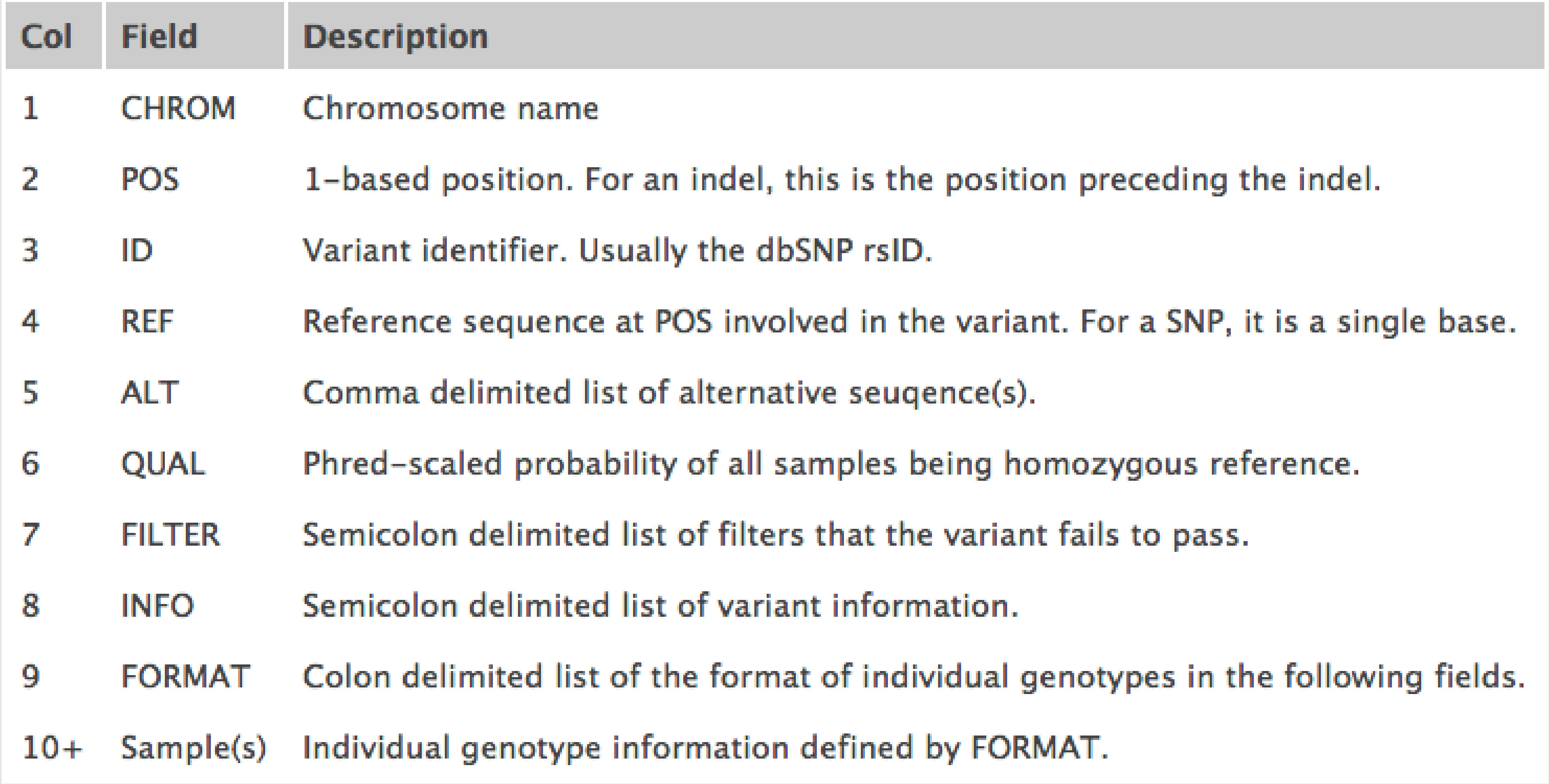
Let’s take a moment to look at a .vcf file we have produced:



```
bioinfo@bioinfo:~/Desktop/Slch04_demo$ less -S variants/gatk_var.vcf
```



```
##INFO=<ID=ReadPosRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt vs. Ref re
##INFO=<ID=SOR,Number=1,Type=Float,Description="Symmetric Odds Ratio of 2x2 contingency table to detect strand b
##contig=<ID=SL3.0ch04,length=66557038>
##source=HaplotypeCaller
#CHROM  POS       ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  pimpi
SL3.0ch04       28326   .       T       A       215.84  .       AC=2;AF=1.00;AN=2;DP=6;ExcessHet=3.0103;FS=0.000
SL3.0ch04       28430   .       T       A       279.80  .       AC=2;AF=1.00;AN=2;DP=7;ExcessHet=3.0103;FS=0.000
SL3.0ch04       29006   .       G       T       270.80  .       AC=2;AF=1.00;AN=2;DP=7;ExcessHet=3.0103;FS=0.000
SL3.0ch04       29076   .       C       T       373.78  .       AC=2;AF=1.00;AN=2;DP=9;ExcessHet=3.0103;FS=0.000
SL3.0ch04       34775   .       T       C       286.80  .       AC=2;AF=1.00;AN=2;DP=7;ExcessHet=3.0103;FS=0.000
SL3.0ch04       35289   .       T       C       1123.77 .       AC=2;AF=1.00;AN=2;DP=26;ExcessHet=3.0103;FS=0.00
SL3.0ch04       35495   .       A       C       62.74   .       AC=2;AF=1.00;AN=2;DP=2;ExcessHet=3.0103;FS=0.000
```



Col	Field	Description
1	CHROM	Chromosome name
2	POS	1-based position. For an indel, this is the position preceding the indel.
3	ID	Variant identifier. Usually the dbSNP rsID.
4	REF	Reference sequence at POS involved in the variant. For a SNP, it is a single base.
5	ALT	Comma delimited list of alternative seuqence(s).
6	QUAL	Phred-scaled probability of all samples being homozygous reference.
7	FILTER	Semicolon delimited list of filters that the variant fails to pass.
8	INFO	Semicolon delimited list of variant information.
9	FORMAT	Colon delimited list of the format of individual genotypes in the following fields.
10+	Sample(s)	Individual genotype information defined by FORMAT.

Exercise 2

- Let’s use **snpEff** to learn a bit more about the SNPs we’ve got…. Do they occur in genes? Are they likely to affect function?

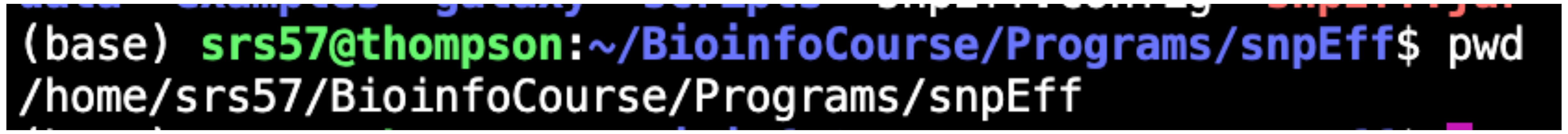
SNP calling: effect prediction

SnpEff http://snpeff.sourceforge.net/

Read the manual!
http://snpeff.sourceforge.net/SnpEff_manual.html

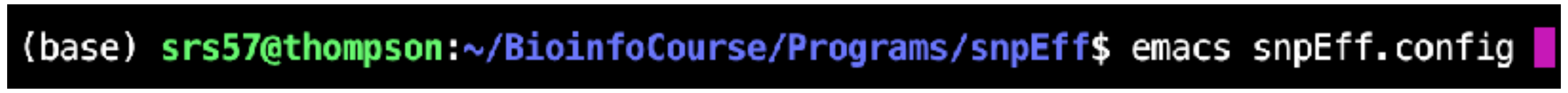
- SnpEff is a variant annotation and effect prediction tool. It annotates and predicts the effects of genetic variants (such as amino acid changes).

1. **cd** to the snpEff directory under programs:

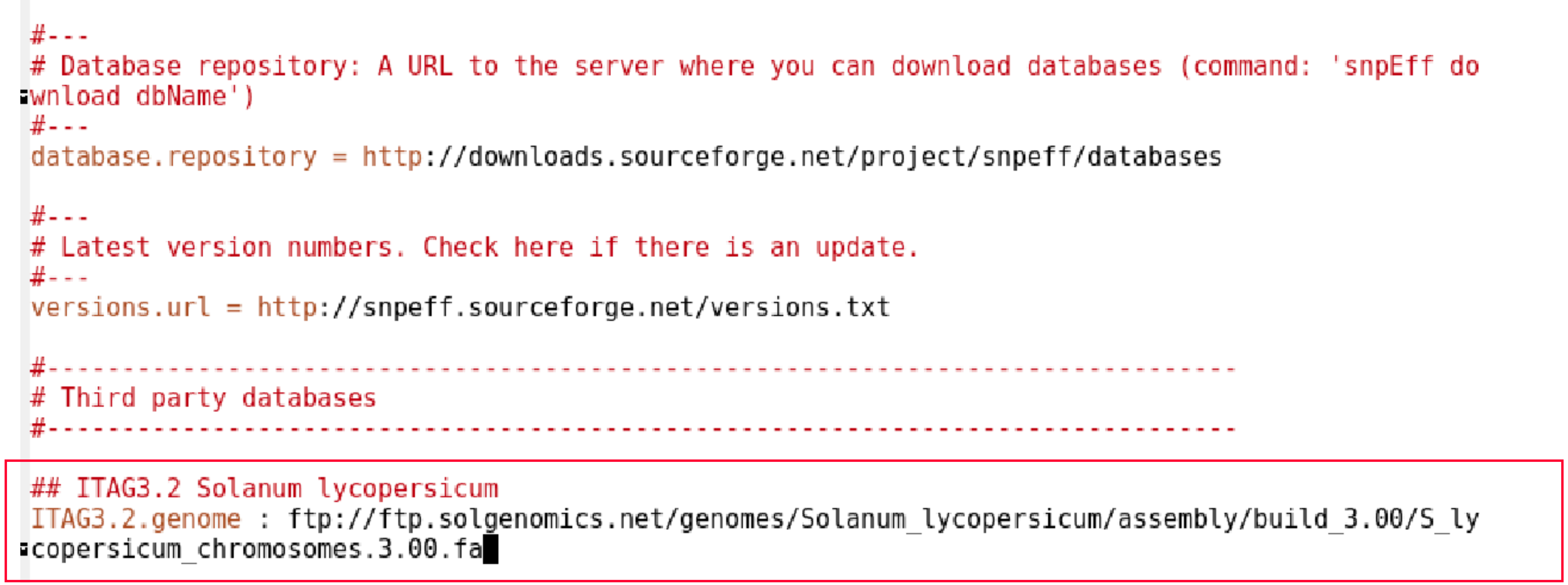


```
(base) srs57@thompson:~/BioinfoCourse/Programs/snpEff$ pwd
/home/srs57/BioinfoCourse/Programs/snpEff
```

2. Open snpEff.config using **emacs**; include the additional lines:

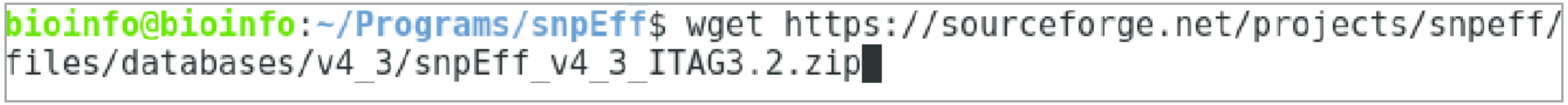


```
(base) srs57@thompson:~/BioinfoCourse/Programs/snpEff$ emacs snpEff.config
```



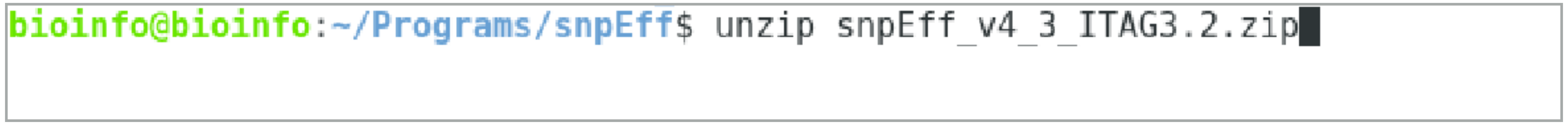
```
#---
# Database repository: A URL to the server where you can download databases (command: 'snpEff do
wnload dbName')
#---
database.repository = http://downloads.sourceforge.net/project/snpeff/databases

#---
# Latest version numbers. Check here if there is an update.
#---
versions.url = http://snpeff.sourceforge.net/versions.txt

#-------------------------------------------------------------------------------
# Third party databases
#-------------------------------------------------------------------------------

## ITAG3.2 Solanum lycopersicum
ITAG3.2.genome : ftp://ftp.solgenomics.net/genomes/Solanum_lycopersicum/assembly/build_3.00/S_ly
copersicum_chromosomes.3.00.fa
```

3. Use **wget** to obtain the snpEff db:



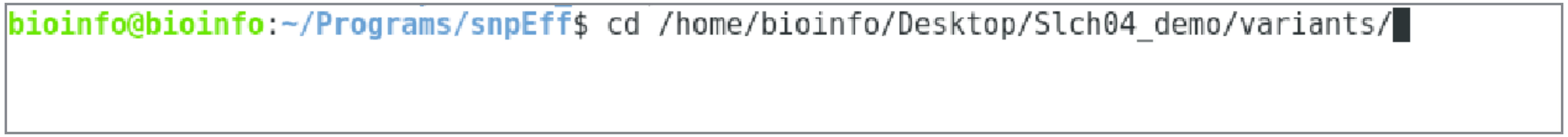
```
bioinfo@bioinfo:~/Programs/snpEff$ wget https://sourceforge.net/projects/snpeff/
files/databases/v4_3/snpEff_v4_3_ITAG3.2.zip
```

4. Unzip the file you just obtained:



```
bioinfo@bioinfo:~/Programs/snpEff$ unzip snpEff_v4_3_ITAG3.2.zip
```

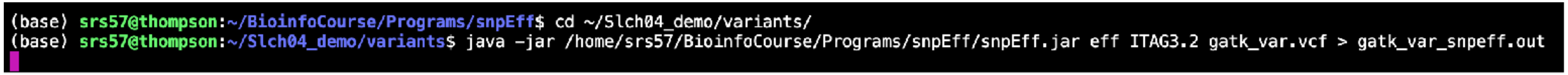
5. **cd** back to the variants directory:



```
bioinfo@bioinfo:~/Programs/snpEff$ cd /home/bioinfo/Desktop/Slch04_demo/variants/
```

SNP calling: effect prediction

6. Run **snpEff**:



```
(base) srs57@thompson:~/BioinfoCourse/Programs/snpEff$ cd ~/Slch04_demo/variants/
(base) srs57@thompson:~/Slch04_demo/variants$ java -jar /home/srs57/BioinfoCourse/Programs/snpEff/snpEff.jar eff ITAG3.2 gatk_var.vcf > gatk_var_snpeff.out
```

- ***.out*** file has the snpEff stats
- ***snpEff_genes.txt*** : SNPs in genes
- ***snpEff_summary.html***

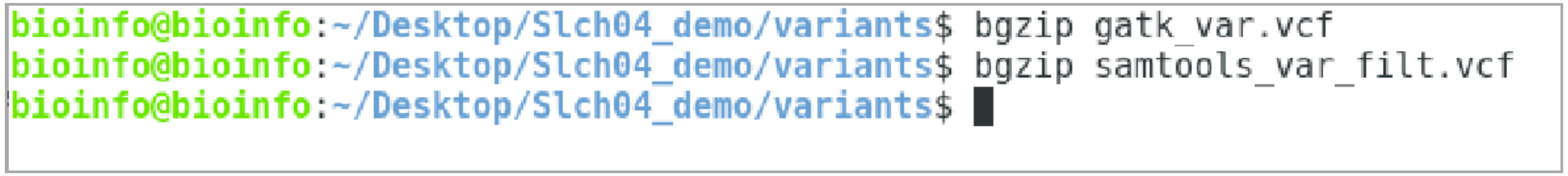
Look at the output and

- Count the number of genes with SNPs
- How many synonymous SNPs?
- How many are non-synonymous?

Exercise 3

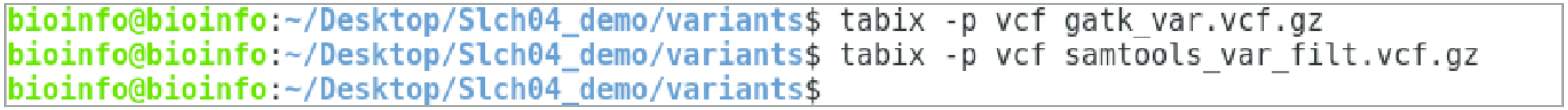
How many SNPs are the same (Intersect) between GATK and samtools output?

1. First, **gzip** the .vcf files:



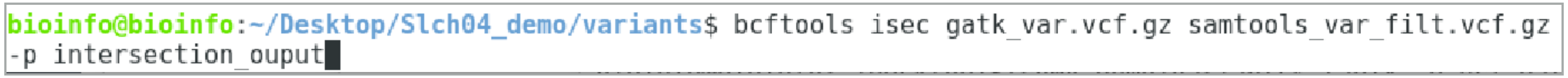
```
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$ bgzip gatk_var.vcf
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$ bgzip samtools_var_filt.vcf
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$ 
```

2. Use **tabix** to index the bgzipped .vcf files:



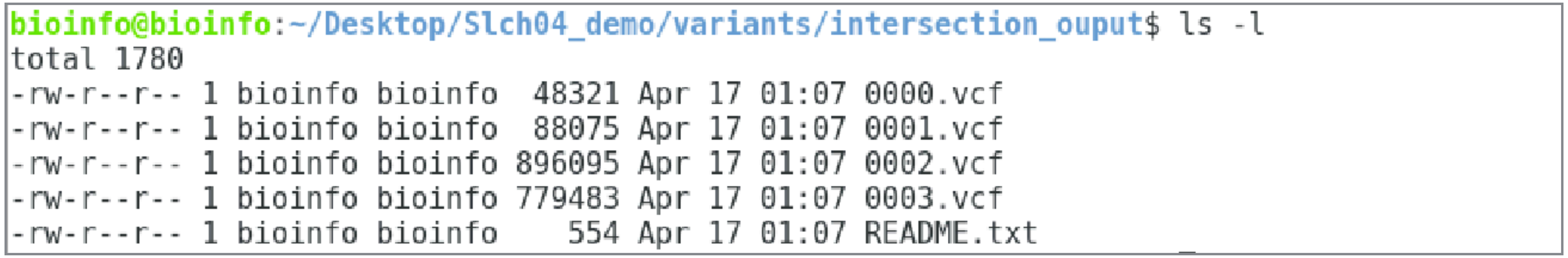
```
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$ tabix -p vcf gatk_var.vcf.gz
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$ tabix -p vcf samtools_var_filt.vcf.gz
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$
```

3. Run **bcftools isec**:



```
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants$ bcftools isec gatk_var.vcf.gz samtools_var_filt.vcf.gz
-p intersection_ouput
```

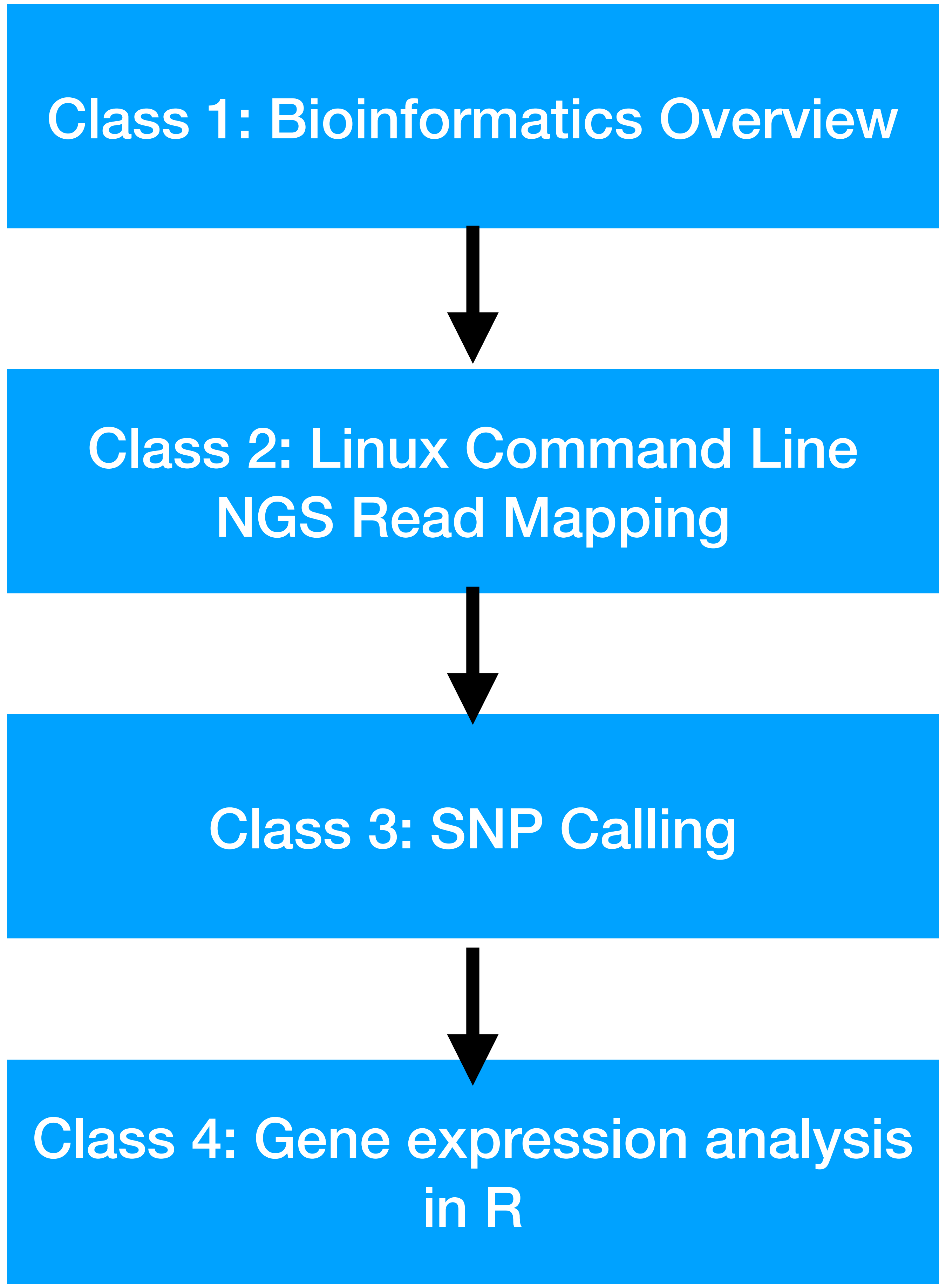
4. Explore the output of bcftools isec (located in the new directory called intersection_output)….



```
bioinfo@bioinfo:~/Desktop/Slch04_demo/variants/intersection_ouput$ ls -l
total 1780
-rw-r--r-- 1 bioinfo bioinfo  48321 Apr 17 01:07 0000.vcf
-rw-r--r-- 1 bioinfo bioinfo  88075 Apr 17 01:07 0001.vcf
-rw-r--r-- 1 bioinfo bioinfo 896095 Apr 17 01:07 0002.vcf
-rw-r--r-- 1 bioinfo bioinfo 779483 Apr 17 01:07 0003.vcf
-rw-r--r-- 1 bioinfo bioinfo    554 Apr 17 01:07 README.txt
```

Course Overview

Make sure you have gene_counts_matrix.csv for next time!!! :)



Class 1: Bioinformatics Overview

Class 2: Linux Command Line
NGS Read Mapping

Class 3: SNP Calling

Class 4: Gene expression analysis in R