

## Variant Calling analysis, Part 2: Analyzing results (Group Exercise)

### Learning objectives:

- Share and publish your workflow histories.
- Examine the outputs.
- View VCF files in JBrowse.
- Examine the filtered VCF file, extract Gene IDs, and create a Venny diagram.

#### • Share workflow histories with others.

1. Make sure your history has a useful name (e.g., Group3 SNPs, etc.) and click on the history action menu icon.
2. Select the “Make History Accessible and Publish” option and check to make sure that all objects within History are accessible.

1 History

Mycelium vs Spore

15 shown, 19 deleted, 148 hidden

49.74 GB

History Actions

Copy

Share or Publish

Show Structure

#### Make History Accessible via Link and Publish It

This history is currently restricted so that only you and the users listed below can access it. You can:

Also make all objects within the History accessible.

Generates a web link that you can share with other people so that they can view and import the history.

Also make all objects within the History accessible.

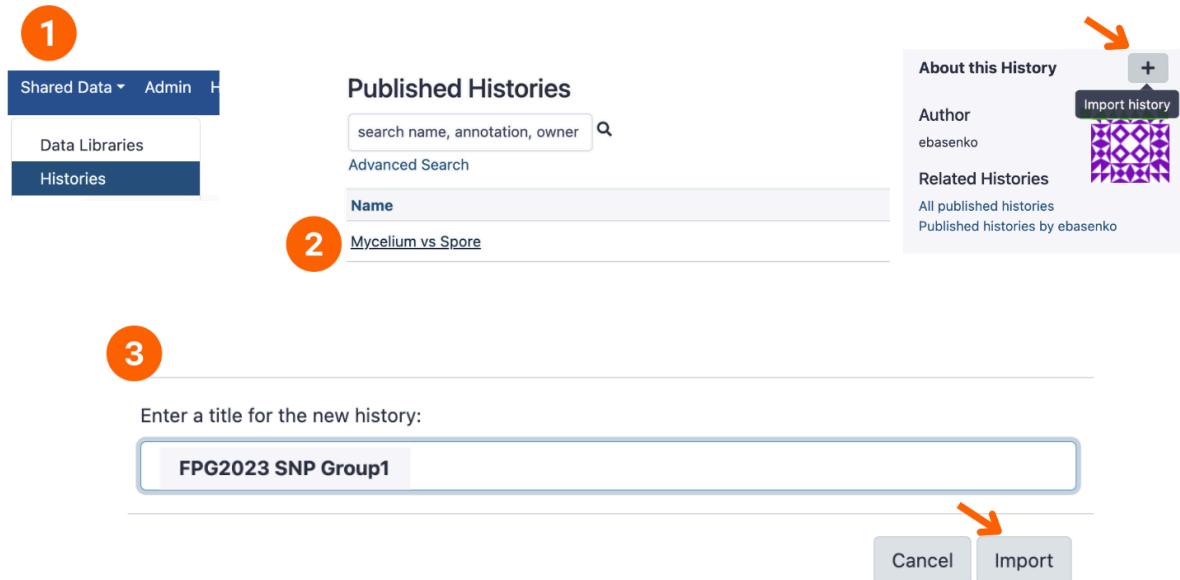
Makes the history accessible via link (see above) and publishes the history to Galaxy's Published Histories section, v

#### Share History with Individual Users

You have not shared this history with any users.

#### • Importing workflow histories and output files into your own Galaxy workspace.

1. Click on “Shared Data” at the top and select “Histories”.
2. Click on the history shared by your colleague, click on the plus icon on the far right and choose to import the history.
3. You can give it a descriptive name if you prefer or leave it as is.



If everything worked out, you should see a list of completed workflow steps highlighted in green. The workflow generates many output files, however not all of the output files are visible. You can explore all the hidden files clicking on the word “hidden” (orange circle) – this will reveal all hidden files.

The Variant calling workflow has three major components: (1) mapping of raw reads to the reference genome, (2) calling variants, and (3) annotating variants. This workflow can be used to call single nucleotide polymorphisms, insertions and deletions (also defined as indels), and multiple nucleotide polymorphisms.

In this workflow, we used Bowtie2 to align and map sequences to a reference genome. Once they are aligned it may be worth checking the quality of this process because misalignments lead to false SNP calls.

SAM or BAM files provide sore this information and you can find these files to export in the hidden workflow steps.

After reads have been aligned, they are sorted based on the chromosomal position. The tool that we are using is called Sort and it belongs to the suite of SAMtools. The sorted file is an input for downstream FreeBayes that calls SNPs and outputs into SnpEff that annotates variants.

FPG2023 SNP GROUP5	
9 shown, 2 deleted, 7 hidden	(orange circle)
11.86 GB	
Many more output files are available to explore →	
filter VCF files using arbitrary expressions →	18: SnpSift Filter on data 16
SnpEff: Analyze and annotate of variants, and calculation of the effects →	17: SnpEff on data 15
Bowtie: Align reads to a reference genome →	16: SnpEff on data 15
	13: BAM to BigWig on data 12
	12: Bowtie2.4.4 on data 8 and data 7: alignments
	10: FastQC on data 4: Webpage
	5: FastQC on data 3: Webpage
	4: SRR10728586_2.fastq.gz
	3: SRR10728586_1.fastq.gz

Analysis and annotation of the genomic variants are carried out by the SnpEff tool. SnpEff is a variant annotation and effect prediction tool. It annotates and predicts the effects of variants on genes (such as amino acid changes). It uses reference genome to annotate genomic variants based on their genomic location and also predicts SNP coding effects. The genomic location features are intronic regions, 5' and 3' UTRs, and upstream, downstream, splice site and intergenic regions. SNP coding effects are categorized based on the effect of the amino

acid change and are classified into synonymous and non-synonymous, gain or loss of start codons, gain of loss of stop codon, and frame shifts.

The SnpSift tool annotates, filters, and manipulates genomic annotated variants. Once you annotated your files using SnpEff, you can use SnpSift to help you filter large genomic datasets (e.g. sort on high or moderate impact SNPs, etc.).

- Examine your results.

1. Click on the *hidden* files link in the history panel to reveal all workflow output files.
2. Examine the output files.
3. What does the tool FASTQC do?
4. What about Sickle?

The output of Sickle is used by a program called Bowtie2.

Bowtie generates a file called a BAM file. Whenever dealing with sequence alignment files

you will likely hear of file formats called SAM or BAM. SAM stands for Sequence Alignment/Map format, and BAM is the binary version of a SAM file.

Many of the downstream analysis programs that use BAM files require a sorted BAM file. This allows for more efficient analysis.

The sorted BAM file is the input for a program called FreeBayes. This program is a Bayesian genetic variant detector designed to find small polymorphisms, specifically SNPs (single-nucleotide polymorphisms), indels (insertions and deletions), MNPs (multi-nucleotide polymorphisms), and complex events (composite insertion and substitution events) smaller than the length of a short-read sequencing alignment. The output for many variant callers is a file called a VCF file. VCF stands for variant interchange format.

5. Examine the VCF file in your results (click on the *eye* icon to view its contents). Detailed information about VCF file content is available here:  
<https://samtools.github.io/hts-specs/VCFv4.2.pdf>

**FPG2023 SNP GROUP5**

9 shown, 2 deleted, 7 hidden  
11.86 GB

**18: SnpSift Filter on data 16**

**17: SnpEff on data 15**

**16: SnpEff on data 15**

**13: BAM to BigWig on data 12**

**12: Bowtie2.4.4 on data 8 and data 7: alignments**

**10: FastQC on data 4: Webpage**

**5: FastQC on data 3: Webpage**

**4: SRR10728586\_2.fastq.gz**

**3: SRR10728586\_1.fastq.gz**

**15: FreeBayes on data 12 (variants) filtered by quality**

This dataset has been hidden  
Unhide it

**14: FreeBayes on data 12 (variants)**

**13: BAM to BigWig on data 12**

**12: Bowtie2.4.4 on data 8 and data 7: alignments**

This dataset has been hidden  
Unhide it

**11: FastQC on data 4: RawData**

**10: FastQC on data 4: Webpage**

This dataset has been hidden  
Unhide it

**9: Singletons from paired-end output of Sickle on data 4 and data 3**

This dataset has been hidden  
Unhide it

**15: FreeBayes on data 12 (variants) filtered by quality**

~300,000 lines

format: vcf, database: FungiDB-34\_ZtriticilPO323\_Genome

Traceback (most recent call last):  
File "metadata/set.py", line 1, in <module>  
from galaxy\_ext.metadata.set\_metadata import set\_metadata;  
set\_metadata()  
File "/opt/galaxy/lib/galaxy\_ext/metadata/set\_metadata.py", line 20, in <module>  
from gal

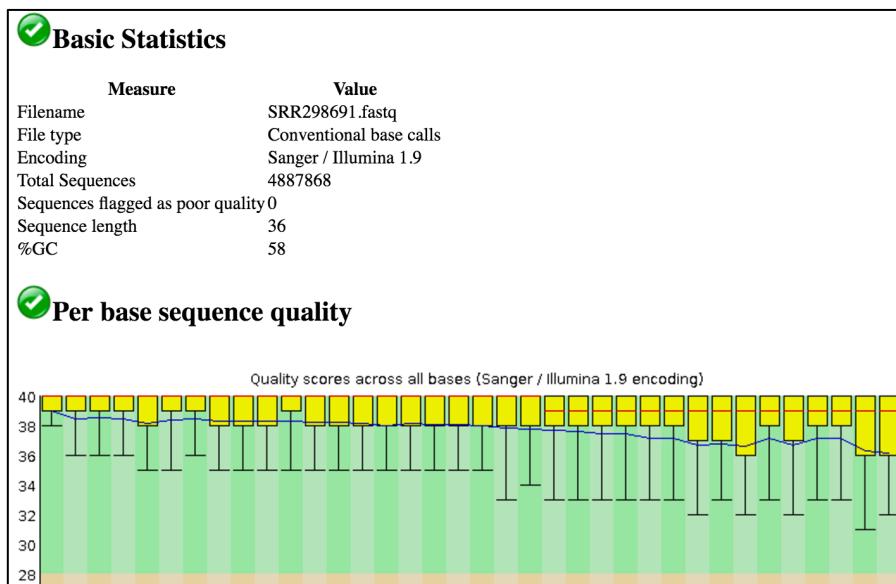
display with IGV local

**1. Chrom**

```
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total read depth a
##INFO=<ID=DPB,Number=1,Type=Float,Description="Total read depth pe
##INFO=<ID=AC,Number=A,Type=Integer,Description="Total number of al
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of al
##INFO=<ID=AF,Number=A,Type=Float,Description="Estimated allele fre
```

- Examine sequence quality based on FastQC quality scores.

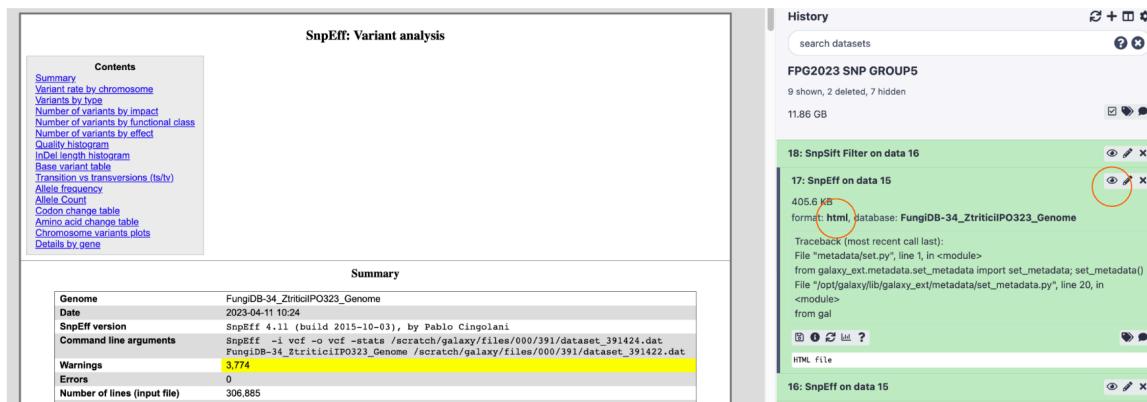
FastQC provides an easy-to-navigate visual representation sequencing data quality and distribution of nucleotides per read position. What does the report tell you about the quality ?



- Examine SnpEff summaries (html)

- Click on the *View data icon* (eye) in the SnpEff output file that has the html format.

This will open the html file in Galaxy for your review.



The header contains a short summary and information about the run and it has several major components:

**The Summary** contains warnings about possible genomic annotation errors or inconsistencies identified in the reference genome. If there are many, use caution interpreting results and examine associated gff files for any issues (*e.g.* missing feature values in gff files, incomplete gene sequences, more than one stop codon per gene, *etc.*). Other components:

- Number of line (input file) - number of lines in vcf file
- Number of not variants: 0 - some packages report non-variant observations for nt positions between reference genome and vcf file generate.
- Number of known variants and multi-allelic VCF entries - if you work with a model organism where some variants were given an accession number (most commonly in mice and human projects) any recognised variants will be listed here

Summary	
Genome	FungiDB-34_ZtriticiciIPO323_Genome
Date	2023-04-11 10:24
SnpEff version	SnpEff 4.11 (build 2015-10-03), by Pablo Cingolani
Command line arguments	SnpEff -i vcf -o vcf -stats /scratch/galaxy/files/000/391/dataset_391424.dat FungiDB-34_ZtriticiciIPO323_Genome /scratch/galaxy/files/000/391/dataset_391422.dat
Warnings	3,774
Errors	0
Number of lines (input file)	306,885
Number of variants (before filter)	307,538
Number of not variants (i.e. reference equals alternative)	0
Number of variants processed (i.e. after filter and non-variants)	307,538
Number of known variants (i.e. non-empty ID)	0 ( 0 % )
Number of multi-allelic VCF entries (i.e. more than two alleles)	653
Number of effects	1,280,819
Genome total length	39,730,198
Genome effective length	39,730,198
Variant rate	1 variant every 129 bases

Variants rate details			
Chromosome	Length	Variants	Variants rate
Ztri_MitoSciffold	43,947	18	2,441
Ztri_chr_1	6,088,797	44,156	137
Ztri_chr_10	1,682,575	15,039	111
Ztri_chr_11	1,624,292	14,012	115
Ztri_chr_12	1,462,624	12,767	114
Ztri_chr_13	1,185,774	10,694	110
Ztri_chr_14	773,098	2,064	374
Ztri_chr_15	639,501	7,821	81
Ztri_chr_16	607,044	5,094	119

- Number of effects - SNP effects summary by type and regions
- Genome total length - number of bp in the reference genome
- Genome effective length - how many nucleotides can be mapped back to the genome
- Variant rate - higher frequency of variants before samples can indicate selective pressure

## Summary statistics for variant types

Here is an example of variant calls and what they mean in terms of nucleotide changes:

Number variantss by type

Type	Total
SNP	114,034
MNP	12,864
INS	6,907
DEL	7,304
MIXED	2,180
INTERVAL	0
<b>Total</b>	<b>143,289</b>

Type	What is means	Example
SNP	Single-Nucleotide Polymorphism	Reference = 'A', Sample = 'C'
Ins	Insertion	Reference = 'A', Sample = 'AGT'
Del	Deletion	Reference = 'AC', Sample = 'C'
MNP	Multiple-nucleotide polymorphism	Reference = 'ATA', Sample = 'GTC'
MIXED	Multiple-nucleotide and an InDel	Reference = 'ATA', Sample = 'GTCAGT'

### Statistics for the variant effects and impacts:

- **High impact** normally refers to frame shift or new stop codon detections as those changes will generate profound effects on gene function.
- **Modifier SNPs** can affect promoter function, while low and moderate SNPs are most commonly identified inside genes and are either non-coding or non-synonymous SNPs.
- Base changes summary. SnpEff html files provide a breakdown of SNPs across gene features:

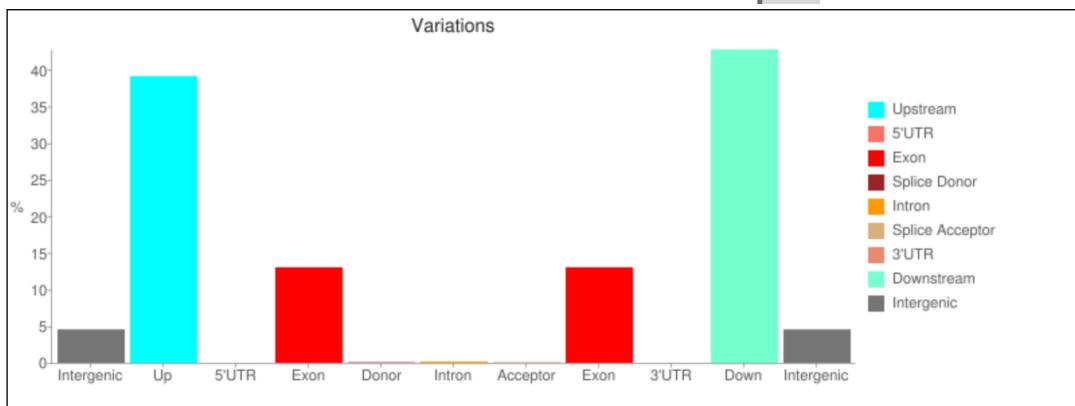
Number of effects by impact

Type (alphabetical order)	Count	Percent
HIGH	1,857	0.145%
LOW	87,874	6.861%
MODERATE	41,970	3.277%
MODIFIER	1,149,118	89.717%

Number of effects by functional class

Type (alphabetical order)	Count	Percent
MISSENSE	29,331	28.472%
NONSENSE	370	0.359%
SILENT	73,317	71.169%

Type (alphabetical order)	Count	Percent
DOWNSTREAM	321,858	40.292%
EXON	67,505	8.451%
INTERGENIC	74,749	9.358%
INTRON	1,064	0.133%
NONE	1	0%
SPlice_SITE_ACCEPTOR	5	0.001%
SPlice_SITE_DONOR	4	0.001%
SPlice_SITE_REGION	176	0.022%
TRANSCRIPT	12	0.002%
UPSTREAM	333,432	41.741%



Additionally, you may see several SNPs being reported in several classes: missense variant + splice region variant. This means that some SNPs that are found within certain splice sites

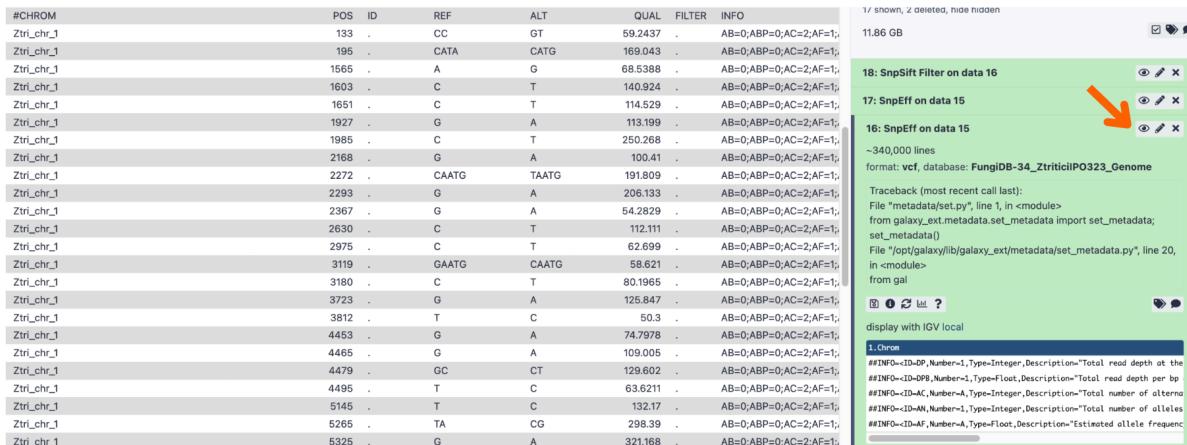
also contain a missense variant. SNPs in the splice sequences may affect intron splicing and lead to read through.

- Quality of reads is indicated in Phred's scale and is a good indicator of the quality of your datasets and results. Quality scores are normally represented by a bar graph where count = number of SNPs and X axis is quality score (higher score mean better p-values and high confidence of the results)
- Base changes: Reflects the frequency of base changes (purine-purine, purine-pyrimidine, pyrimidine-purine, pyrimidine-pyrimidine).
- Transition and transversion ratio help to identify if you may have a selective pressure on certain alleles (high ratio suggests that genes may be under selective pressure).
- Allele frequency statistics reports frequency of alleles and help to identify potential sequencing artifacts due to PCR enrichment step (generation of heterozygous counts in a haploid organism).

The vcf file generated by SnpEff contains information about SNPs and the genomic location. Post-processing of SNP data is normally required to make sense of thousands of SNPs and to decide which ones have biological and functional importance. Data processing can help you to extract SNP distribution and parse associated data including GeneIDs, protein-coding annotations, and effects in sequence ontology terms such as missense or synonymous variants, stop codon gain, etc. and also link changes to the genome model. SnpSift is among other programs that is often in SNP data post-processing. It can be installed and run locally to manipulate vcf files. Alternatively, you can also visualize vcf files in Artemis (additional steps are required to format the data).

## Examining SNP information.

You can view the SNP information by clicking on the “eye” icon within the SnpEff vcf file.



The screenshot shows a Galaxy tool interface for viewing a VCF file. The main area displays a table of SNP data with columns: #CHROM, POS, ID, REF, ALT, QUAL, FILTER, and INFO. The INFO column shows various allele frequency (AF) values. Above the table, a status bar indicates "1 / shown, 2 deleted, hide hidden" and "11.86 GB". Below the table, a list of panels is shown:

- 18: SnpSift Filter on data 16 (highlighted in green)
- 17: SnpEff on data 15
- 16: SnpEff on data 15

The bottom of the interface shows the command-line trace and the Galaxy configuration file (galaxy.yml) for this tool.

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
Ztri_chr_1	133	.	CC	GT	59.2437	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	195	.	CATA	CATG	169.043	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	1565	.	A	G	68.5388	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	1603	.	C	T	140.924	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	1651	.	C	T	114.529	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	1927	.	G	A	113.199	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	1985	.	C	T	250.268	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	2168	.	G	A	100.41	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	2272	.	CAATG	TAATG	191.809	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	2293	.	G	A	206.133	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	2367	.	G	A	54.2829	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	2630	.	C	T	112.111	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	2975	.	C	T	62.699	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	3119	.	GAATG	CAATG	58.621	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	3180	.	C	T	80.1965	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	3723	.	G	A	125.847	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	3812	.	T	C	50.3	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	4453	.	G	A	74.9798	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	4465	.	G	A	109.005	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	4479	.	GC	CT	129.602	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	4495	.	T	C	63.6211	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	5145	.	T	C	132.17	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	5265	.	TA	CG	298.39	.	AB=0;ABP=0;AC=2;AF=1;
Ztri_chr_1	5325	.	G	A	321.168	.	AB=0;ABP=0;AC=2;AF=1;

The vcf file generated by SnpEff contains information about SNPs and the genomic location. Here is an example of a file opened in Excel:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	unknown
CM001231	189057	.	AG	CT	787.449	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:143:0:0:143:5341:-207.887,-43.0473,0	
CM001231	483825	.	G	A	64.8756	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:4:0:0:4:146:-10.0999,-1.20412,0	
CM001231	518226	.	G	C	51.7908	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:8:0:0:7:276:-11.5007,-2.10721,0	
CM001231	574021	.	C	G	237.265	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:17:0:0:17:583:-39.079,-5.11751,0	
CM001231	609879	.	GAA	CAG	55.2785	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:32:8:277:22:861:-18.1711,-0.694735,0	
CM001231	1090073	.	G	T	79.4156	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:8:2:75:6:238:-11.5539,-1.36362,0	
CM001231	1090104	.	A	T	70.961	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:6:0:0:6:220:-12.5146,-1.80618,0	
CM001231	1153611	.	CCTC	GCTG	111.123	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:8:5:188:3:97:-9.30616,-6.1461,0	
CM001231	1159150	.	CT	GC	126.126	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:31:0:0:19:741:-29.7713,-5.71957,0	
CM001231	1159438	.	C	G	82.3312	.	AB=0;ABP=0;GT:DP:RO:QF	0/0:47:30:1092:17:640:0,-9.53002,-3.50705	
CM001231	1159465	.	G	C	249.656	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:126:47:1770:79:3013:-53.8644,-25.2134,0	
CM001231	1159499	.	T	C	124.95	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:143:32:1167:111:4248:-76.1575,-33.4865,0	
CM001231	1181576	.	CC	TG	191.675	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:27:0:0:25:924:-41.7448,-7.52575,0	
CM001231	1293309	.	C	G	51.22	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:2:0:0:2:78:-6.92763,-0.60206,0	
CM001231	1323058	.	TT	GC	71.3001	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:6:0:0:6:223:-12.5485,-1.80618,0	
CM001231	1485397	.	A	G	3558.42	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:499:0:0:497:18671:-804.678,-149.612,0	
CM001231	1485429	.	G	A	3783.33	.	AB=0;ABP=0;GT:DP:RO:QF	1/1:517:1:38:516:20010:-843.425,-151.978,0	

### Filtering VCF file data.

VCF files contain a lot of data about variants and their positions. SnpEff generates various analyses/summaries of VCF files (including GeneIDs that overlap variant positions). However, it is often necessary to filter VCF files further to obtain useful information for your specific question. For example, you may want to filter out SNP positions that have an impact on the coding sequence.

One tool that can be used is called SnpSift Filter (look at the last step of the pipeline you just ran). This tool allows you to write complex expressions to filter a VCF file. Your workflow is set up to use an expression that filters VCF files on moderate and high impact SNPs (this setting can be adjusted manually in the workflow editor). Here is the exact expression used:

```
((((ANN[*].IMPACT has 'HIGH') | (ANN[*].IMPACT has 'MODERATE')) & ((na FILTER) | (FILTER = 'PASS')))
```

- Extract filtered VCF file (SnpSift output) and convert into an Excel document.

For this exercise, two groups will be sharing data SnpSift outputs: group 1 & 2, group 3 & 4, and group 5 & 6. File manipulations should be performed on both SnpSift vcf files.

Look at the filtered vcf file in Galaxy. Notice that the Gene IDs are buried in the file, but the file has some structure which means you can extract them either programmatically or using a program like Excel.

```

9;SRF=6;SRP=26.4622;SRR=23;TYPE=snp;ANN=A|missense_variant|MODERATE|Afu1g0040|Afu1g00140|transcript|Afu1g00140-T|Coding|1;SRP=29.6108;SRR=14;TYPE=snp;ANN=A|missense_variant&splice_region_variant|MODERATE|Afu1g00140|Afu1g00140|transcript|Afu1g00140-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=C|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=0;SRR=0;TYPE=complex;ANN=G|GC|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=T|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=0;SRR=5.18177;SRR=0;TYPE=snp;ANN=A|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=G|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=G|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=0;SRR=0;TYPE=complex;ANN=GATCGGA|missense_variant|MODERATE|Afu1g00230|Afu1g00230|transcript|Afu1g00230-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=C|splice_acceptor_variant&intron_variant|HIGH|Afu1g00300|Afu1g00300|transcript|Afu1g00300-T|Coding|1;SRP=0;SRR=5.18177;SRR=0;TYPE=snp;ANN=C|splice_acceptor_variant|MODERATE|Afu1g00300|Afu1g00300|transcript|Afu1g00300-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=G|missense_variant|MODERATE|Afu1g00300|Afu1g00300|transcript|Afu1g00300-T|Coding|3/5;SRP=5.18177;SRR=1;TYPE=mnp;ANN=AGT|missense_variant|MODERATE|Afu1g00300|Afu1g00300|transcript|Afu1g00300-T|Coding|1;SRP=5.18177;SRR=0;TYPE=snp;ANN=A|stop_gained|HIGH|Afu1g00550|Afu1g00550|transcript|Afu1g00550-T|Coding|2.c.575G>A;SRP=0;SRR=5.18177;SRR=0;TYPE=snp;ANN=T|missense_variant|MODERATE|Afu1g00550|Afu1g00550|transcript|Afu1g00550-T|Coding|1;SRP=0;SRR=5.18177;SRR=0;TYPE=snp;ANN=T|missense_variant|MODERATE|Afu1g00550|Afu1g00550|transcript|Afu1g00550-T|Coding|1;SRP=0;SRR=5.18177;SRR=0;TYPE=snp;ANN=A|missense_variant|MODERATE|Afu1g00550|Afu1g00550|transcript|Afu1g00550-T|Coding|2/2.c.697G>A;TYPE=mnp;ANN=TT|missense_variant|MODERATE|Afu1g00570|Afu1g00570|transcript|Afu1g00570-T|Coding|1.c.910_911delGTinsA;SRP=0;SRR=5.18177;SRR=0;TYPE=snp;ANN=T|missense_variant|MODERATE|Afu1g00570|Afu1g00570|transcript|Afu1g00570-T|Coding|1;SRP=0;SRR=0;TYPE=complex;ANN=TATT|stop_gained|HIGH|Afu1g00570|Afu1g00570|transcript|Afu1g00570-T|Coding||c.892_896delGAAT

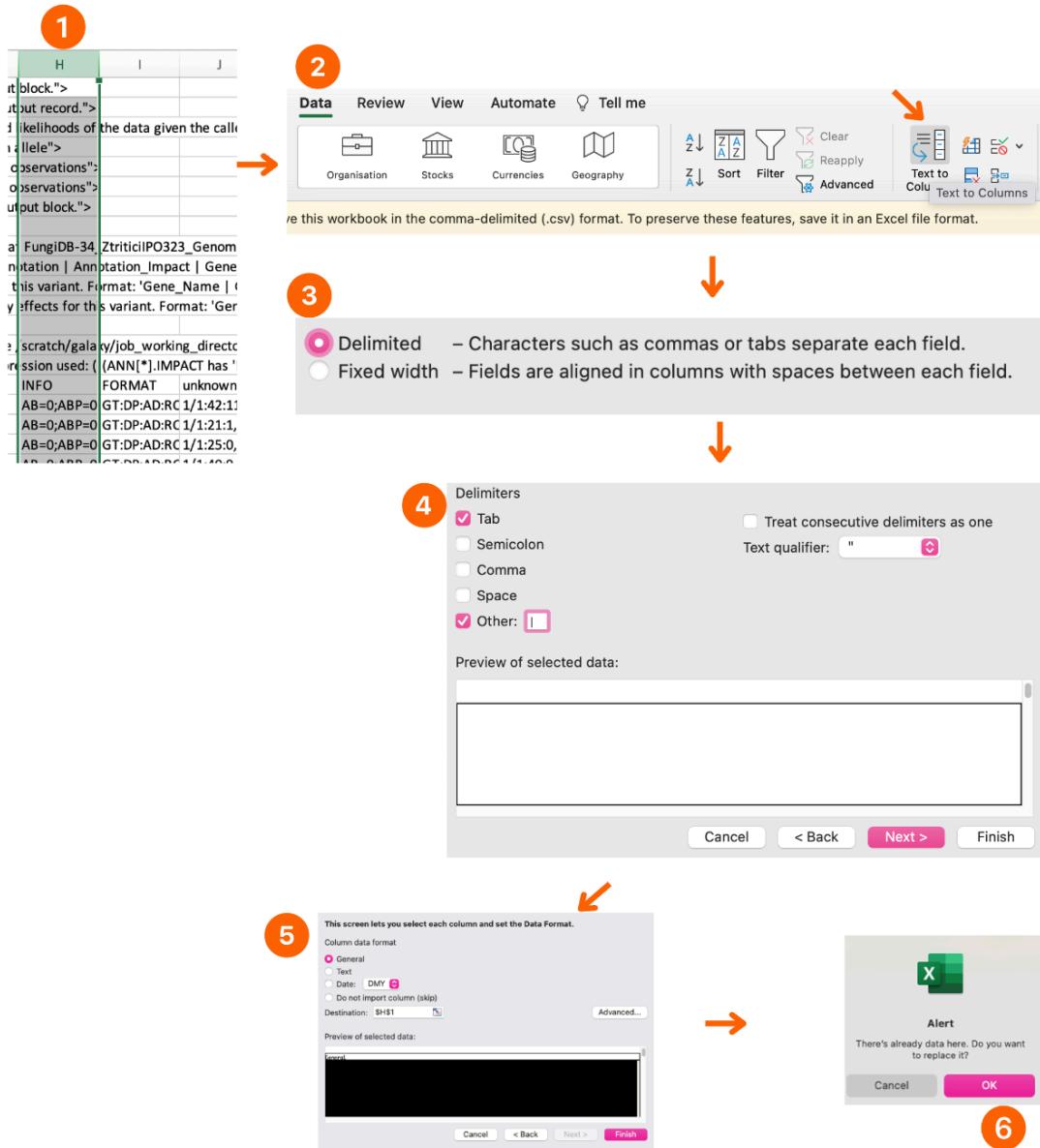
```

Here are some steps you can take to extract Gene IDs from two VCF files then compare them to identify genes that are in common or that distinguish the two files.

1. Download the SnpSift Filter output by clicking on the save icon.
2. Right click and open this file with Excel.

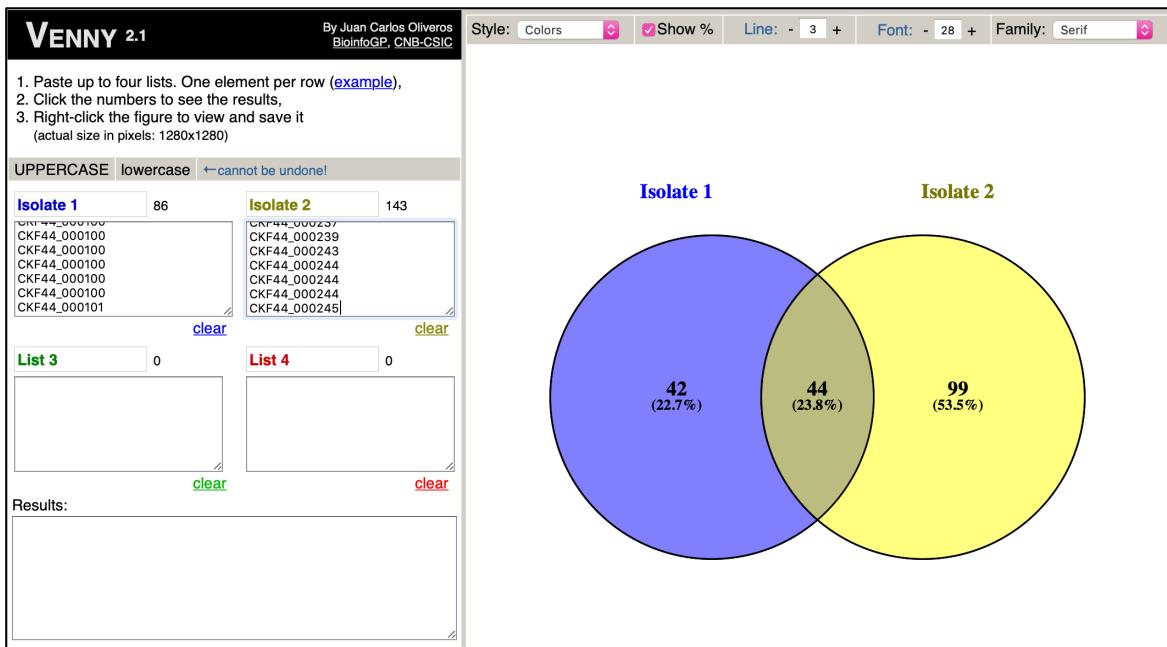
	e_ID	Feature_Typ	Feature_ID	Transcript_E_Rank	HGVS.c	HGVS.p	cDNA.pos / tCDN.pos / Cl_AA.pos / AA	Distance	ERRORS / WARNINGS / INFO'
49									genotype Quality, the Phred-scaled marginal (or unconditional) probability of the called genotype">
50									genotype Likelihood, log10-scaled likelihoods of the data given the called genotype for each possible genotype generated from the reference and alternate alleles given the sample ploidy">
51		"Read Depth">							
52		"Reference allele observation count">							
53		"Sum of quality of the reference observations">							
54		"Alternate allele observation count">							
55		"Sum of quality of the alternate observations">							
56		"an">							
57		les/008/dataset_8077.dat PlasmoDB-29_Pfalciparum3D7_Genome /scratch/galaxy/files/008/dataset_8075.dat "							
58	e_ID	Feature_Typ	Feature_ID	Transcript_E_Rank	HGVS.c	HGVS.p	cDNA.pos / tCDN.pos / Cl_AA.pos / AA	Distance	ERRORS / WARNINGS / INFO'
59		scriptsAffected">							
60		scriptsAffected">							
61		o Cingolani"							
62		008/dataset_8076.dat -e /scratch/galaxy/job_working_directory/004/4170/tmpAQDb8H"							
63									
64	QUAL	FILTER	INFO	FORMAT	unknown				
65	163.615..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	c.1729_1730 p.Asp577Pro>T229 6492	1729 6492	577/2163	
66	59.2743..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Jan c.1773A>T p.Lys591Asn>1773 6492	1773 6492	591/2163	
67	112.419..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	c.4420_4421 p.Thr1474Gln>A420 6492	4420 6492	1474/2163	
68	123.945..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Jan c.4432C>G p.Gln1478Gln>A432 6492	4432 6492	1478/2163	
69	70.7189..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Jan c.4466C>A p.Thr1489Ily>Arg4466 6492	4466 6492	1489/2163	
70	203.132..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Jan c.4655T>G p.Leu1552Asr>Arg4655 6492	4655 6492	1552/2163	
71	149.708..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	c.4733_4734 p.Asp1578Ala>I473 6492	4733 6492	1578/2163	
72	101.922..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Jan c.4741C>A p.Gln1581Ily>Arg474 6492	4741 6492	1581/2163	
73	106.751..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Feb c.5647A>G p.Asn1883Asn>Gly5647 6492	5647 6492	1883/2163	
74	68.702..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Feb c.5873C>G p.Thr1958Sle>Arg5873 6492	5873 6492	1958/2163	
75	599.479..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	c.6472_6474 p.Ala2158Ser>Leu6472 6492	6472 6492	2158/2163	
76	6.44607..	A B 0 AB =0;missense_va MODERATE	Pf3D7_0100100	Pf3D7_0100 transcript	Pf3D7_0100 Coding	2-Feb c.6490C>G p.Ile2159Asn>Leu6490 6492	6490 6492	2159/2163	

- Manipulate Excel file to display SNP info in columns.
1. Select the “INFO” column.
  2. Navigate to the “Data” tab in Excel and choose “Text to Columns”.
  3. Use the “Delimited” option.
  4. Set delimiters to the “Tab” and “|” in the “Other” and click “Next”
  5. Leave other criteria at default and click on the “Finish” button.
  6. Click “OK” on the Alert pop-up.



Now you can look for Gene IDs of interest in the excel file. For example, if this is a known drug resistant line you can sort and examine SNPs based on their characteristics.

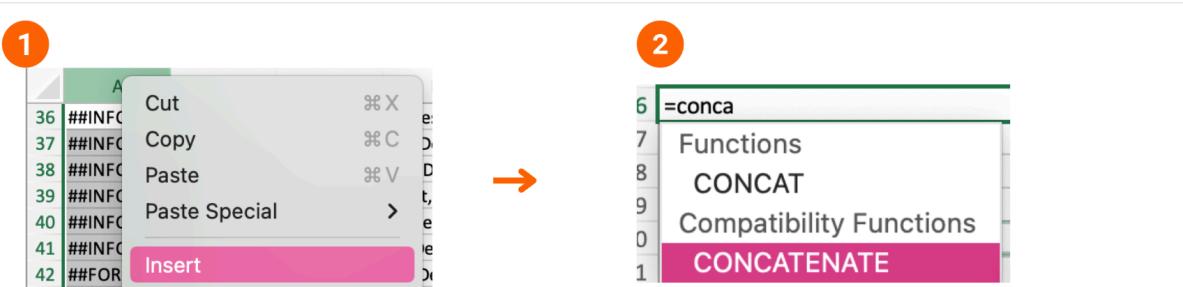
If you are comparing two or more strains, you may want to extract gene IDs from all VCF files and identify common signatures across isolates or strains. For this type of analysis, you can use <http://bioinfogp.cnb.csic.es/tools/venny/> to generate a Venn diagram:



The screenshot above is showing comparison of between lists of GeneIDs. Is it possible to miss some important polymorphisms using this method? Of course, the answer is yes😊  
For example, it is quite possible that a gene with a SNP in the WT and a SNP in the mutant that will be in the intersection of the two gene lists, contains different SNPs – you will miss this by doing the above steps. Below is a description of steps you can take to create a list of unique IDs for SNPs. This list of unique IDs can then be used in Venny.

- **Analyze your data in Venny.**

1. Start with the same excel files that you opened in the above section. Insert an empty column before the data.
2. Deploy the concatenate function in Excel.
3. Create a unique ID for SNPs by combining information from multiple columns to create something that looks like this: **chromosome:position:geneID**  
To do this you will use the concatenate function in Excel:  
`=concatenate(cell#1,":",cell#2,":",cell#3)`  
Cell#1 = cell with chromosome number  
Cell#2 = cell with position  
Cell#3 = cell with GeneID



3

SUM	A	B	C	D	E	F	G	H	I	J	K	L	M	N
50		#INFO<=ID=LOF,Number=.,Type=String,Description="Predicted loss of function effects for this variant. Format: 'Gene_Name   Gene_ID   Number_of_transcripts_in_gene   Percent_of_transcripts_in_gene   Description'"												
51		#INFO<=ID=NMD,Number=.,Type=String,Description="Predicted nonsense mediated decay effects for this variant. Format: 'Gene_Name   Gene_ID   Number_of_transcripts_in_gene   Percent_of_transcripts_in_gene   Description'"												
52		#SnpSiftVersion="SnpSift 4.1 (build 2015-10-03), by Pablo Cingolani"												
53		#SnpSiftCmd="SnpSift filter -f /scratch/galaxy/files/000/391/dataset_391071.dat -e /scratch/galaxy/job_working_directory/000/260/260223/configs/tmpufmf1sa3"												
54		#FILTER=<ID=SnpSift,Description="SnpSift 4.1 (build 2015-10-03), by Pablo Cingolani, Expression used: (((ANN*)>IMPACT has 'HIGH') & ((ANN*)>IMPACT has 'MODERATE')) & ((na FILTER)"												
55		#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO					
56	=CONCATENATE(B56,".",C56,".",M56)	Chr1_A_fumigatus_Af293	1314781		AATA	GATG	85.5736	.	AB=0;ABP=0;/ missense_va MODERATE	Afu1g00410	Afu1g00410	transcript	/	
57		Chr1_A_fumigatus_Af293	131514.		T	C	72.9308	.	AB=0;ABP=0;/ missense_va MODERATE	Afu1g00410	Afu1g00410	transcript	/	
58		Chr1_A_fumigatus_Af293	143640.		T	C	97.7793	.	AB=0;ABP=0;/ missense_va MODERATE	Afu1g00450	Afu1g00450	transcript	/	
59		Chr1_A_fumigatus_Af293	144396.		G	A	135.073	.	AB=0;ABP=0;/ missense_va MODERATE	Afu1g00450	Afu1g00450	transcript	/	

You should get unique SNP IDs that look like this (for example):

CP022321.1:15259:CKF44\_000003. Copy this function for other entries:

Chr1_A_fumigatus_Af293:185468:Afu1g00580	Chr1_A_fumigatus_Af293	185468	.	TTC
Chr1_A_fumigatus_Af293:185521:Afu1g00580	Chr1_A_fumigatus_Af293	185521	.	A
Chr1_A_fumigatus_Af293:401061:Afu1g01110	Chr1_A_fumigatus_Af293	401061	.	G
Chr1_A_fumigatus_Af293:402973:Afu1g01120	Chr1_A_fumigatus_Af293	402973	.	GG
Chr1_A_fumigatus_Af293:403260:Afu1g01120	Chr1_A_fumigatus_Af293	403260	.	A
Chr1_A_fumigatus_Af293:405284:Afu1g01130	Chr1_A_fumigatus_Af293	405284	.	T
Chr1_A_fumigatus_Af293:405434:Afu1g01130	Chr1_A_fumigatus_Af293	405434	.	A
Chr1_A_fumigatus_Af293:406035:Afu1g01140	Chr1_A_fumigatus_Af293	406035	.	G
Chr1_A_fumigatus_Af293:406481:Afu1g01140	Chr1_A_fumigatus_Af293	406481	.	G
Chr1_A_fumigatus_Af293:407398:Afu1g01160	Chr1_A_fumigatus_Af293	407398	.	A
	Chr1_A_fumigatus_Af293	407406	.	A
	Chr1_A_fumigatus_Af293	410505	.	C

4. Copy these newly generated unique IDs into List 1 and List 2 on Venny <http://bioinfogp.cnb.csic.es/tools/venny/> and examine the data.

4

Chr1\_A\_fumigatus\_Af293:145783:Afu1g00460  
Chr1\_A\_fumigatus\_Af293:148888:Afu1g00470  
Chr1\_A\_fumigatus\_Af293:148933:Afu1g00470  
Chr1\_A\_fumigatus\_Af293:148945:Afu1g00470  
Chr1\_A\_fumigatus\_Af293:185087:Afu1g00580  
Chr1\_A\_fumigatus\_Af293:185100:Afu1g00580  
Chr1\_A\_fumigatus\_Af293:185439:Afu1g00580  
Chr1\_A\_fumigatus\_Af293:185468:Afu1g00580  
Chr1\_A\_fumigatus\_Af293:185521:Afu1g00580  
Chr1\_A\_fumigatus\_Af293:401061:Afu1g01110  
Chr1\_A\_fumigatus\_Af293:402973:Afu1g01120  
Chr1\_A\_fumigatus\_Af293:403260:Afu1g01120  
Chr1\_A\_fumigatus\_Af293:405284:Afu1g01130  
Chr1\_A\_fumigatus\_Af293:405434:Afu1g01130  
Chr1\_A\_fumigatus\_Af293:406035:Afu1g01140  
Chr1\_A\_fumigatus\_Af293:406481:Afu1g01140

VENNY 2.1 By Juan Carlos Olvera (bioinfogp.cnb.csic.es)

1. Paste up to four lists. One element per row ([example](#)).  
2. Click the numbers to see the results.  
3. Right-click the figure to view and save it  
(actual size in pixels: 1280x1280)

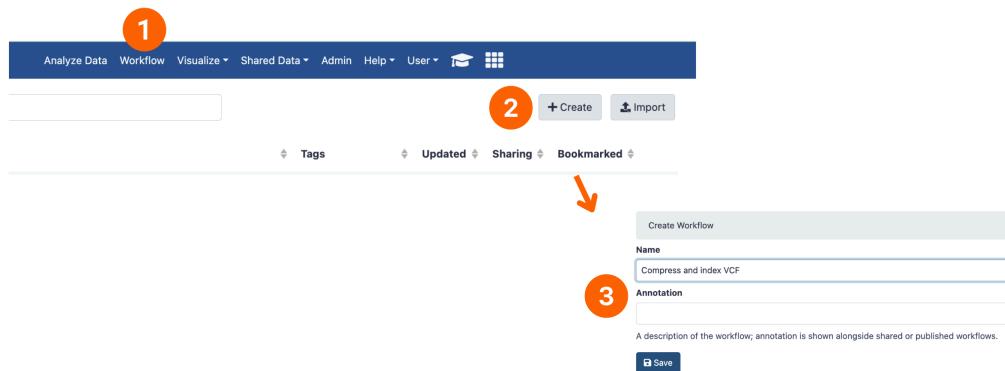
List 1	12	List 2	12
Chr1_A_fumigatus_Af293:145783:Afu1g00460		Chr1_A_fumigatus_Af293:185439:Afu1g00580	
Chr1_A_fumigatus_Af293:148888:Afu1g00470		Chr1_A_fumigatus_Af293:185468:Afu1g00580	
Chr1_A_fumigatus_Af293:148933:Afu1g00470		Chr1_A_fumigatus_Af293:185521:Afu1g00580	
Chr1_A_fumigatus_Af293:148945:Afu1g00470		Chr1_A_fumigatus_Af293:401061:Afu1g01110	
Chr1_A_fumigatus_Af293:185087:Afu1g00580		Chr1_A_fumigatus_Af293:402973:Afu1g01120	
Chr1_A_fumigatus_Af293:185100:Afu1g00580			
Chr1_A_fumigatus_Af293:185439:Afu1g00580			
Chr1_A_fumigatus_Af293:185468:Afu1g00580			
Chr1_A_fumigatus_Af293:185521:Afu1g00580			
Chr1_A_fumigatus_Af293:401061:Afu1g01110			
Chr1_A_fumigatus_Af293:402973:Afu1g01120			
Chr1_A_fumigatus_Af293:403260:Afu1g01120			
Chr1_A_fumigatus_Af293:405284:Afu1g01130			
Chr1_A_fumigatus_Af293:405434:Afu1g01130			
Chr1_A_fumigatus_Af293:406035:Afu1g01140			
Chr1_A_fumigatus_Af293:406481:Afu1g01140			

## Viewing VCF file results in the JBrowse genome browser.

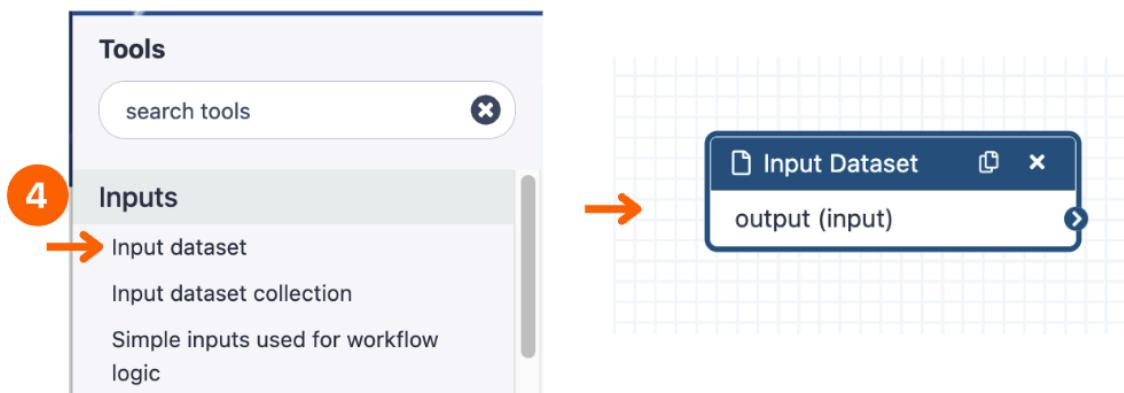
- **Create a workflow to generate a compressed vcf and index files for viewing your data in JBrowse.**

To view a VCF file in JBrowse, it first has to be indexed and compressed. This is done using two tools: bgzip and tabix, respectively. You can run these tools sequentially or you can set up a mini workflow and then run the workflow to generate the output files as follows:

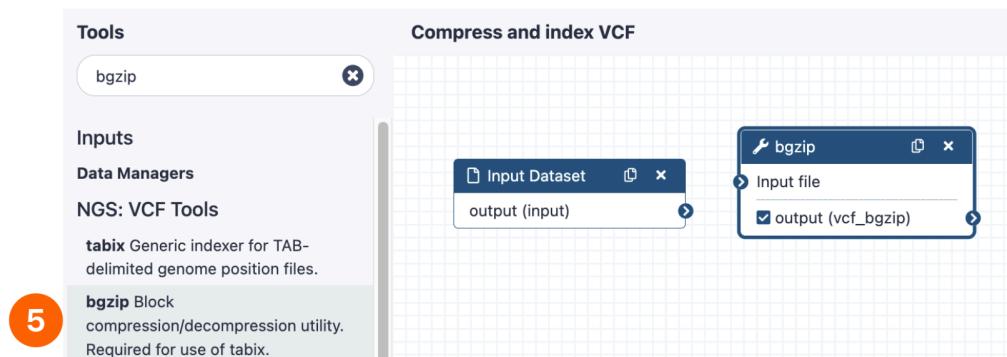
1. Click on the “Workflow” menu.
2. Click on the “Create” button to start a new workflow.
3. Give the workflow a name (e.g. Compress and index VCF) and click on the save button. This will open a workflow canvas.



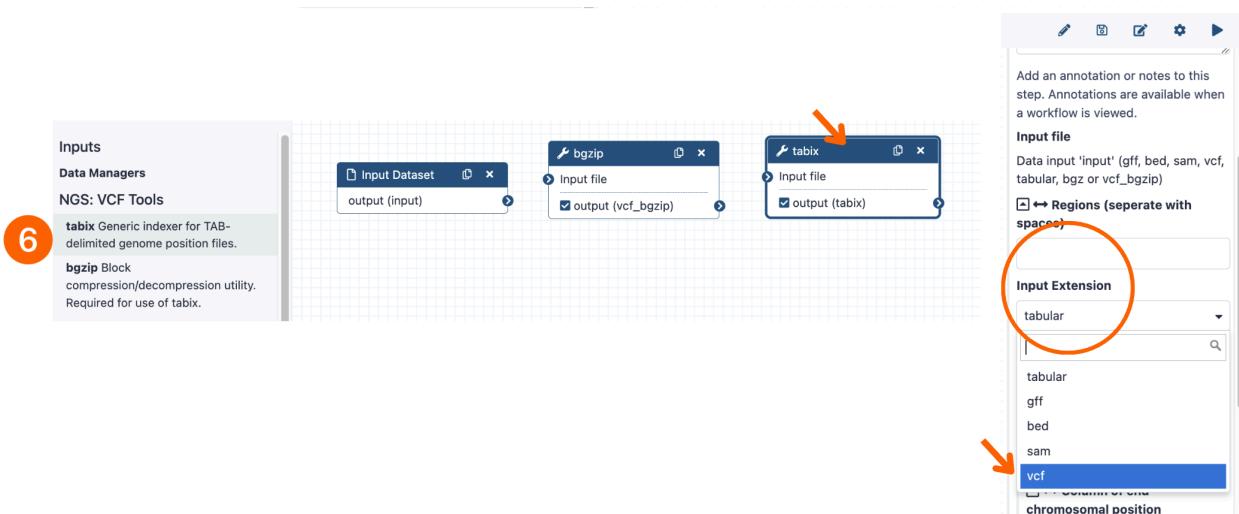
4. All workflows must start with an input file so add the “Input Dataset” step to the workflow using the menu on the left (you must click on the tool for it to appear in the workflow editor canvas).



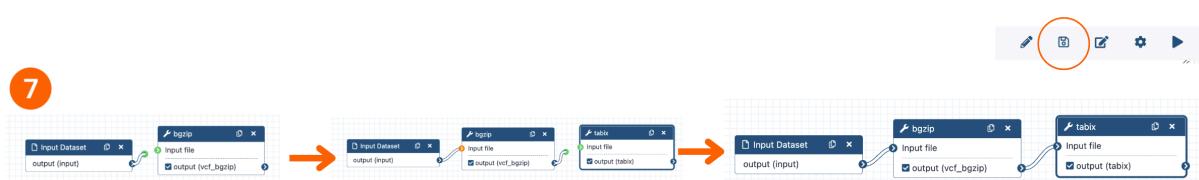
5. Using the menu on the left, search for and add the “bgzip” tool.



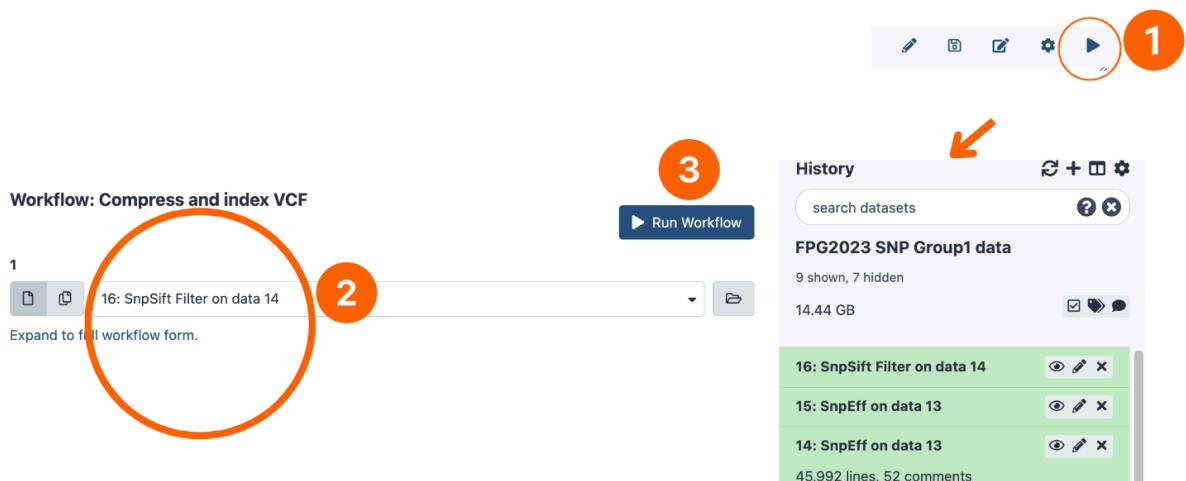
6. Using the menu on the left, search for and add the “tabix” tool. Left-click on the “tabix” icon and select “vcf” under “input selection” on the right (tool option section)



7. Connect each step/tool into a workflow and save it (the button is at the top of the screen)



- Run the newly created workflow to generate a compressed vcf and index files.
  - Click on the “Play” button to start your workflow.
  - Select the VCF file you want to process.
  - Click on the “Run Workflow” button.



After the workflow completed running, you should have 2 new files in the history on the right (tabix and bgzip).

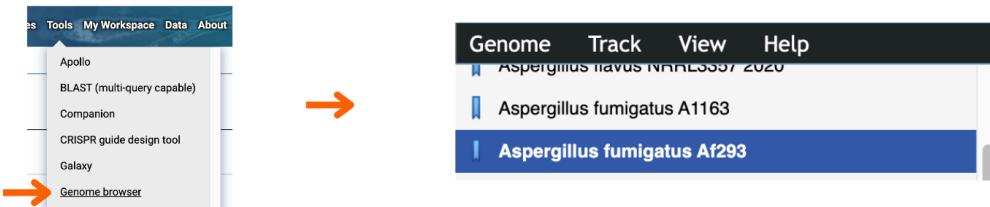
The screenshot shows the Galaxy interface after the workflow has completed. A green message box at the top left says "Successfully invoked workflow Compress and index VCF." Below it, a progress bar shows "3 of 3 steps successfully scheduled" and "2 of 2 jobs complete". To the right is a "History" panel. Two new entries are listed: "20: tabix on data 19" and "19: bgzip on data 14". Both of these entries are circled in orange.

- Download compressed vcf (vcf\_bgzip) and index (tabix) files and view them in JBrowse.
  - Download both files by clicking on the download icon. You will need both files.

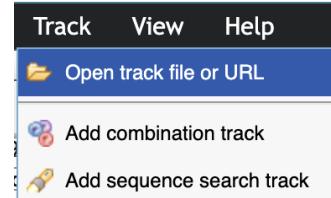
The screenshot shows the Galaxy interface displaying the details of two files from the history. On the left is "19: bgzip on data 14" and on the right is "20: tabix on data 19". Both files are circled in orange. The "19: bgzip on data 14" panel shows a file size of 7.2 MB and a "format: vcf\_bgzip, database: FungiDB-29\_AfumigatusAf293\_Genome". The "20: tabix on data 19" panel shows a file size of 14.2 KB and a "format: tabix, database: FungiDB-29\_AfumigatusAf293\_Genome". Both panels include a download icon (circled in orange) and other standard file metadata.

- After the files are downloaded, rename them as follows:
  - The **vcf\_bgzip** file to “**group#.vcf.gz**” (i.e. **group1.vcf.gz**)
  - The **tabix** file to “**group#.vcf.gz.tbi**” (i.e. **group1.vcf.gz.tbi**)

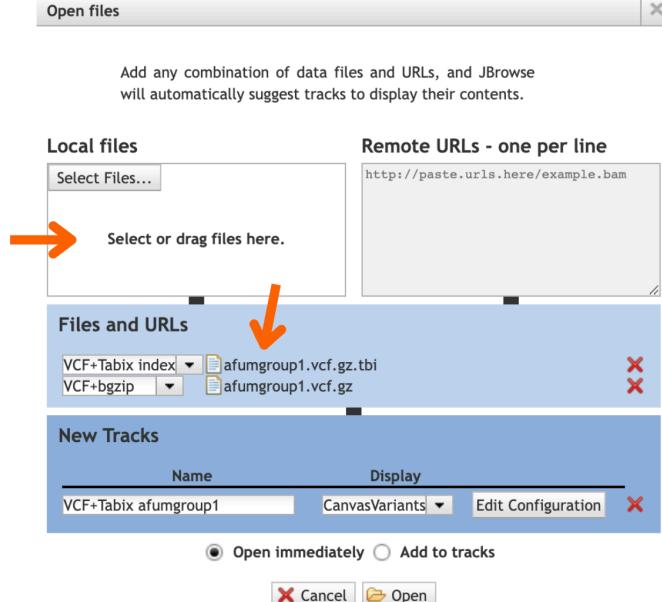
3. Navigate to JBrowse in FungiDB and select the correct genome from the Genome drop-down menu.



4. Click on the Track menu, select "Open track file or URL".



5. Drag and drop your files in the window that appears. Notice that the file formats are autodetected. Click on the “Open” button at the bottom of the pop-up.



You should now be able to view the SNPs in JBrowse.

