Aug 22, 2018

BSA-seq in Maize

G3: Genes|Genomes|Genetics

DOI
dx.doi.org/10.17504/protocols.io.qyedxte

Harry Klein[1], Yuguo Xiao[2], Phillip A. Conklin[3], Rajanikanth Govindarajulu[4], Jacob A. Kelly[2], Michael J. Scanlon[3], Clinton J. Whipple[2], Madelaine Bartlett[1]

[1]University of Massachusetts at Amherst; [2]Brigham Young University; [3]Cornell University; [4]West Virginia University

Harry Klein

DOI: dx.doi.org/10.17504/protocols.io.qyedxte

External link: https://doi.org/10.1534/g3.118.200499

Protocol Citation: Harry Klein, Yuguo Xiao, Phillip A. Conklin, Rajanikanth Govindarajulu, Jacob A. Kelly, Michael J. Scanlon, Clinton J. Whipple, Madelaine Bartlett 2018. BSA-seq in Maize. **protocols.io https://dx.doi.org/10.17504/protocols.io.qyedxte**

Manuscript citation:
Klein H, Xiao Y, Conklin PA, Govindarajulu R, Kelly JA, Scanlon MJ, Whipple CJ, Bartlett M, Bulked-Segregant Analysis Coupled to Whole Genome Sequencing (BSA-Seq) for Rapid Gene Cloning in Maize. G3: Genes|Genomes|Genetics 8(11). doi: 10.1534/g3.118.200499

Protocol status: Working

Created: June 13, 2018

Last Modified: August 22, 2018

Protocol Integer ID: 13030

Abstract

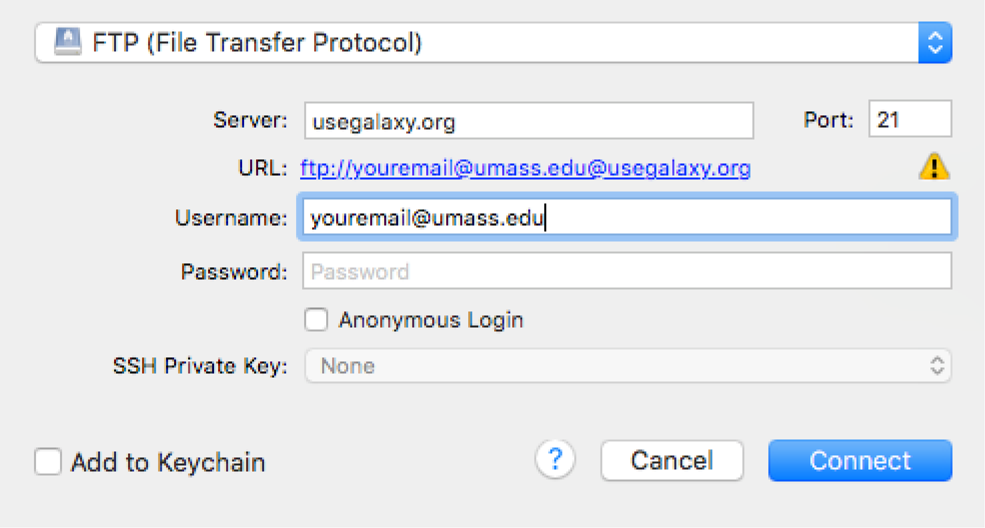
In the past, identifying a gene of interest for a mutant phenotype in maize has proven to be a laborious process. Now, with the aid of next generation sequencing technology, the researcher can quickly identify the region containing a mutant gene and in some cases identify the causative lesion. In our BSA-seq protocol, we use the Galaxy platform for file processing and manipulation. This platform is intended to provide a user friendly experience, so that researchers of any level can clone maize genes. We start with raw sequencing reads from our F2 mapping population and end with plots of SNPs on each chromosome and a list of candidate SNPs. The Galaxy platform generates the files needed for plotting SNPs with the R package ggplot2 and for identifying candidate SNPs with SnpEff. Our protocol will guide the user step-by-step through the Galaxy platform, chromosome plotting, and SnpEff.

Phase 1: Map your reads, generate a filtered pileup file, and a Varscan file

1 **Upload data**

1. https://usegalaxy.org/ has its own uploading format, but it is limited to 2GB of data uploaded. Chances are, the data is much larger than 2GB, so use a FTP to upload your data. The maximum amount of memory available on the Galaxy server for a given user account is 250Gb. If very large sequencing files are being used, we recommend using another solution. We found that 250Gb was plenty of space to run our protocol, however, some files may need to be deleted to remain under the 250Gb limit. For mapping to B73, upload the reference genome downloaded from maizesequence.org.

2. We used CyberDuck, but there is a large list of free FTP uploaders to choose from. CyberDuck has an option to set the server, which means that it sets the location of where your data is going (Fig. 1). After connecting to Galaxy (https://galaxyproject.org/ftp-upload/), simply drag your raw sequencing files into the Cyberduck program and they will begin uploading.


FTP (File Transfer Protocol)

Server: usegalaxy.org Port: 21

URL: ftp://youremail@umass.edu@usegalaxy.org

Username: youremail@umass.edu

Password: Password

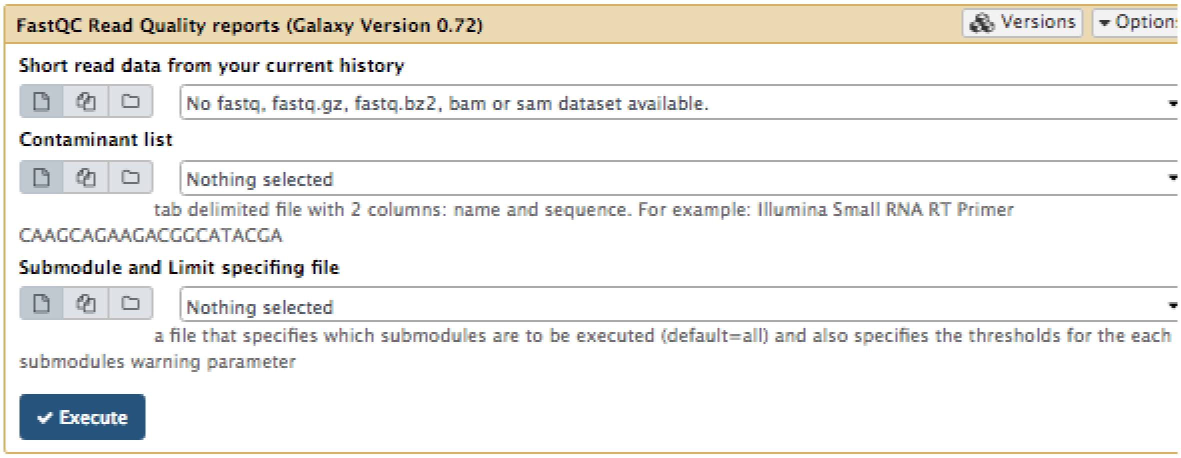
□ Anonymous Login

SSH Private Key: None

3. Once your FTP uploader has finished uploading the data, click the tab on Galaxy that says “choose FTP file” in Get Data>Upload File. In your instance, your data files should show and you can have them be uploaded to the history (Fig 2.).

2 **Check the quality of your sequencing data**

Check the quality of your sequencing data with FastQC. We use this step to assess our sequencing run and determine a good quality cutoff. Contaminant list and submodules are optional.



FastQC Read Quality reports (Galaxy Version 0.72)

Short read data from your current history

No fastq, fastq.gz, fastq.bz2, bam or sam dataset available.

Contaminant list

Nothing selected

tab delimited file with 2 columns: name and sequence. For example: Illumina Small RNA RT Primer CAAGCAGAAGACGGCATACGA

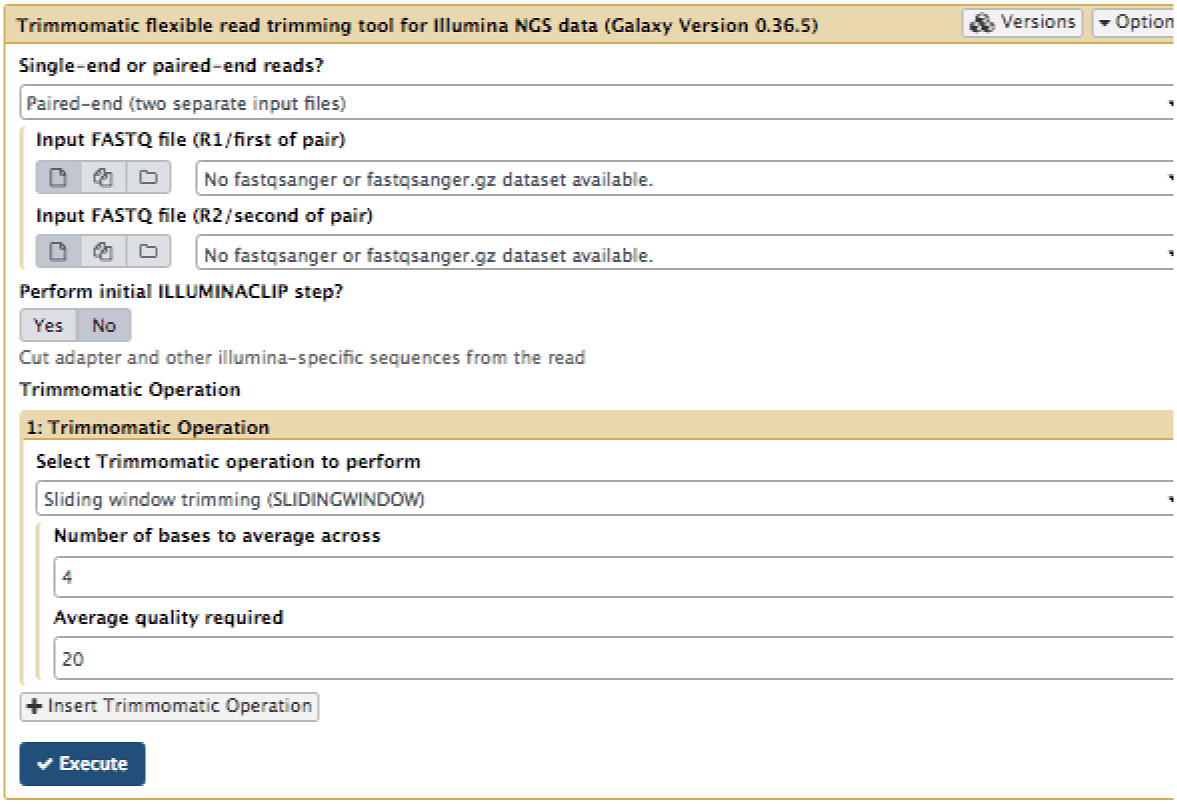
Submodule and Limit specifing file

Nothing selected

a file that specifies which submodules are to be executed (default=all) and also specifies the thresholds for the each submodules warning parameter

3 Trim reads with Trimmomatic

Run Trimmomatic on your data. Typically, we trim reads with <20 Phred quality scores and use a sliding window of 4. This is the default setting for the Trimmomatic tool on Galaxy. Select your files and chug away.



Trimmomatic flexible read trimming tool for Illumina NGS data (Galaxy Version 0.36.5)

Single-end or paired-end reads?
Paired-end (two separate input files)

Input FASTQ file (R1/first of pair)
No fastqsanger or fastqsanger.gz dataset available.

Input FASTQ file (R2/second of pair)
No fastqsanger or fastqsanger.gz dataset available.

Perform initial ILLUMINACLIP step?
Yes No
Cut adapter and other illumina-specific sequences from the read

Trimmomatic Operation

1: Trimmomatic Operation

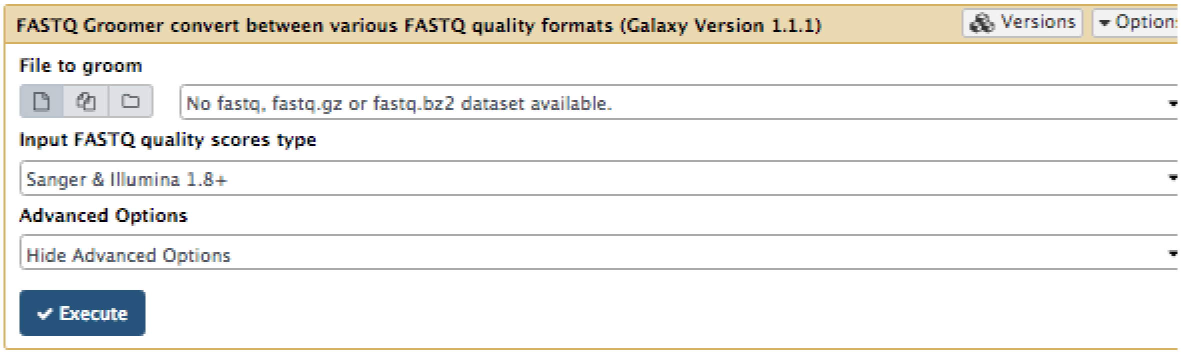
Select Trimmomatic operation to perform
Sliding window trimming (SLIDINGWINDOW)

Number of bases to average across
4

Average quality required
20

4 **Check the format of your sequencing files (Optional step depending on the aligner used)**

If you map your sequencing data using Bowtie2, it requires your data to be in the ***fastqsanger format***. In some instances, the format of the original sequencing data is in the fastq format, which means that it needs to be changed in order for https://usegalaxy.org/ to recognize and align data. Use the program FASTQ Groomer to convert your files.



FASTQ Groomer convert between various FASTQ quality formats (Galaxy Version 1.1.1)

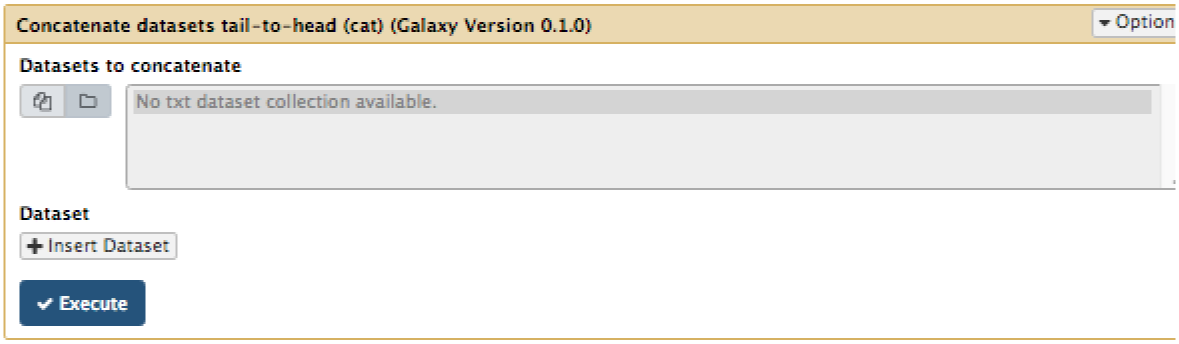
File to groom
No fastq, fastq.gz or fastq.bz2 dataset available.

Input FASTQ quality scores type
Sanger & Illumina 1.8+

Advanced Options
Hide Advanced Options

5 **Optional step: Concatenate multiple datasets**

Use 'cat' to merge multiple sequencing files into one file. Often, sequencing files will be broken up by the Illumina machines.



Concatenate datasets tail-to-head (cat) (Galaxy Version 0.1.0)

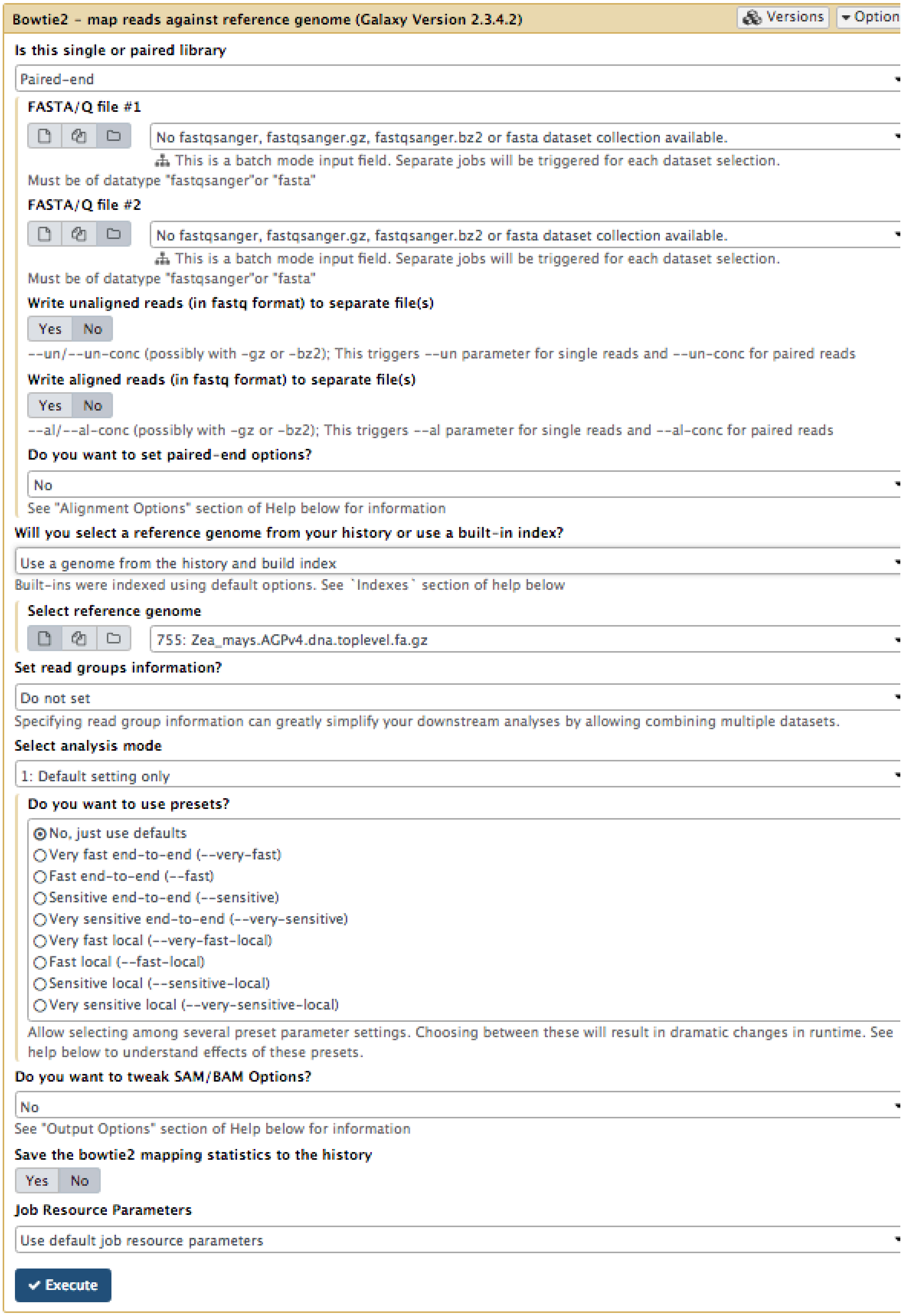
Datasets to concatenate

No txt dataset collection available.

Dataset

6 **Map your reads**

We used Bowtie2 to map reads to version 4 of the B73 reference genome. Select paired end or single end reads. Build a genome from your history and select the B73 reference genome. All other options are optional. Typically this step takes 4-24 hours depending on how big your sequencing data files are.



Bowtie2 - map reads against reference genome (Galaxy Version 2.3.4.2)

Is this single or paired library

Paired-end

FASTA/Q file #1

No fastqsanger, fastqsanger.gz, fastqsanger.bz2 or fasta dataset collection available.

This is a batch mode input field. Separate jobs will be triggered for each dataset selection.

Must be of datatype "fastqsanger"or "fasta"

FASTA/Q file #2

No fastqsanger, fastqsanger.gz, fastqsanger.bz2 or fasta dataset collection available.

This is a batch mode input field. Separate jobs will be triggered for each dataset selection.

Must be of datatype "fastqsanger"or "fasta"

Write unaligned reads (in fastq format) to separate file(s)

Yes No

--un/--un-conc (possibly with -gz or -bz2); This triggers --un parameter for single reads and --un-conc for paired reads

Write aligned reads (in fastq format) to separate file(s)

Yes No

--al/--al-conc (possibly with -gz or -bz2); This triggers --al parameter for single reads and --al-conc for paired reads

Do you want to set paired-end options?

No

See "Alignment Options" section of Help below for information

Will you select a reference genome from your history or use a built-in index?

Use a genome from the history and build index

Built-ins were indexed using default options. See `Indexes` section of help below

Select reference genome

755: Zea_mays.AGPv4.dna.toplevel.fa.gz

Set read groups information?

Do not set

Specifying read group information can greatly simplify your downstream analyses by allowing combining multiple datasets.

Select analysis mode

1: Default setting only

Do you want to use presets?

● No, just use defaults
○ Very fast end-to-end (--very-fast)
○ Fast end-to-end (--fast)
○ Sensitive end-to-end (--sensitive)
○ Very sensitive end-to-end (--very-sensitive)
○ Very fast local (--very-fast-local)
○ Fast local (--fast-local)
○ Sensitive local (--sensitive-local)
○ Very sensitive local (--very-sensitive-local)

Allow selecting among several preset parameter settings. Choosing between these will result in dramatic changes in runtime. See help below to understand effects of these presets.

Do you want to tweak SAM/BAM Options?

No

See "Output Options" section of Help below for information

Save the bowtie2 mapping statistics to the history

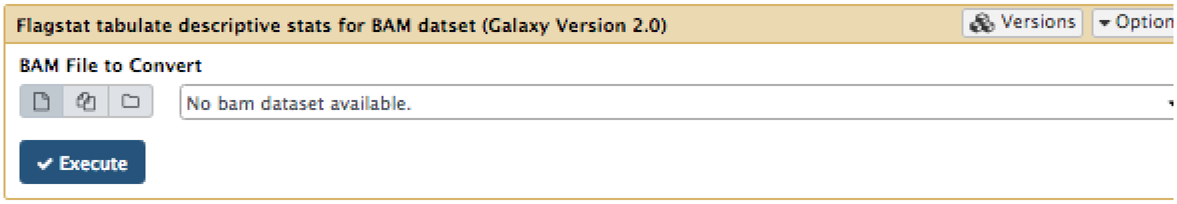
Yes No

Job Resource Parameters

Use default job resource parameters

7 **Calculate coverage from your data**

Run Samtools Flagstat on your alignment file. The flagstat results will tell you how many reads mapped to the reference genome, handy for calculating sequencing coverage.

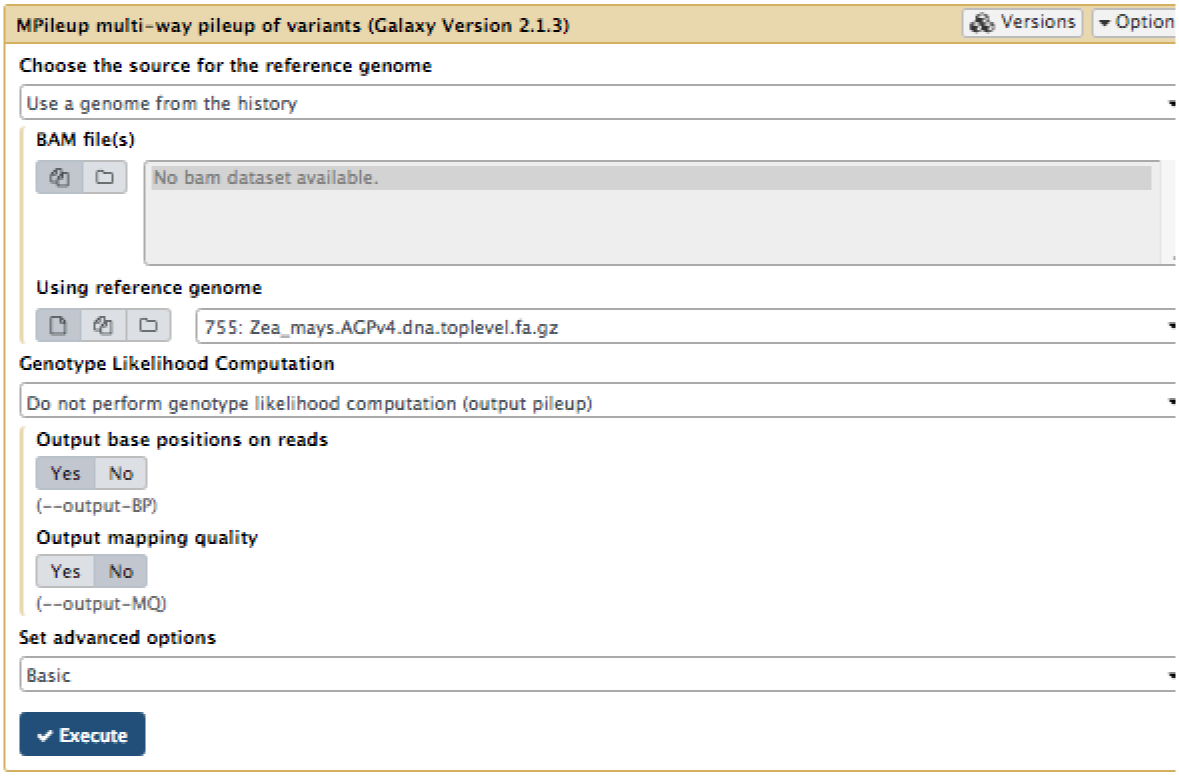

Flagstat tabulate descriptive stats for BAM datset (Galaxy Version 2.0)

BAM File to Convert

No bam dataset available.

8 **Call variants from your alignment**

Run Samtools Mpileup, choose your genome from your history, choose Do not perform genotype likelihood computation (output pileup), and output base positions on reads.


MPileup multi-way pileup of variants (Galaxy Version 2.1.3)

Choose the source for the reference genome

Use a genome from the history

BAM file(s)

No bam dataset available.

Using reference genome

755: Zea_mays.AGPv4.dna.toplevel.fa.gz

Genotype Likelihood Computation

Do not perform genotype likelihood computation (output pileup)

Output base positions on reads

Yes No

(--output-BP)

Output mapping quality

Yes No

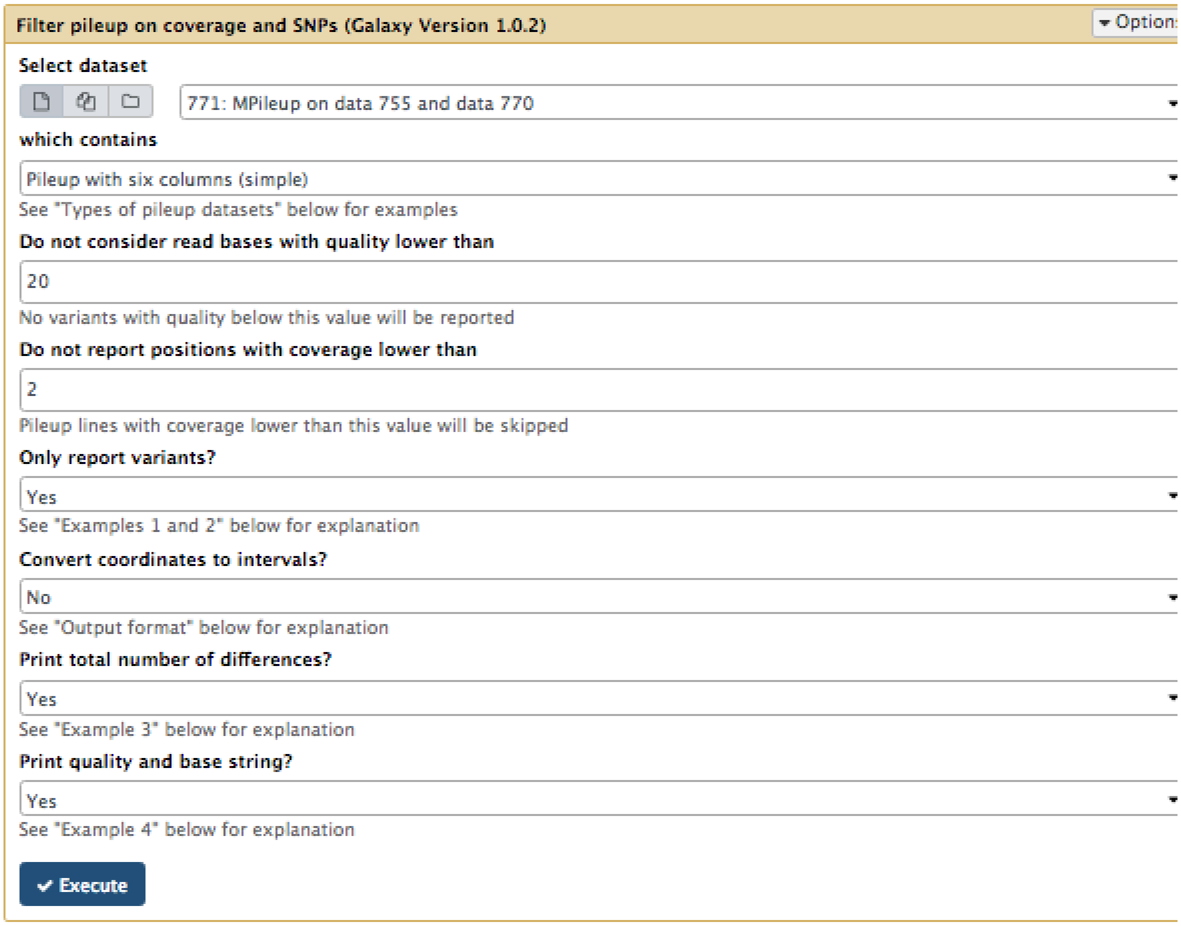
(--output-MQ)

Set advanced options

Basic

9 **Filter your Mpileup file**

Use Samtools filter pileup plugin to filter SNPs on coverage. Assign a coverage cutoff based on the average cutoff of your data. If you have low average coverage (<15) call all reads with coverage >2. Reads with a coverage of 1 can be sequencing error, so a coverage of 2 can give you more confidence. For high coverage data (>15X), a coverage cutoff of 8 should be sufficient. Additionally, print total number of differences.



Filter pileup on coverage and SNPs (Galaxy Version 1.0.2)

Select dataset: 771: MPileup on data 755 and data 770

which contains: Pileup with six columns (simple)

See "Types of pileup datasets" below for examples

Do not consider read bases with quality lower than: 20

No variants with quality below this value will be reported

Do not report positions with coverage lower than: 2

Pileup lines with coverage lower than this value will be skipped

Only report variants?: Yes

See "Examples 1 and 2" below for explanation

Convert coordinates to intervals?: No

See "Output format" below for explanation

Print total number of differences?: Yes

See "Example 3" below for explanation

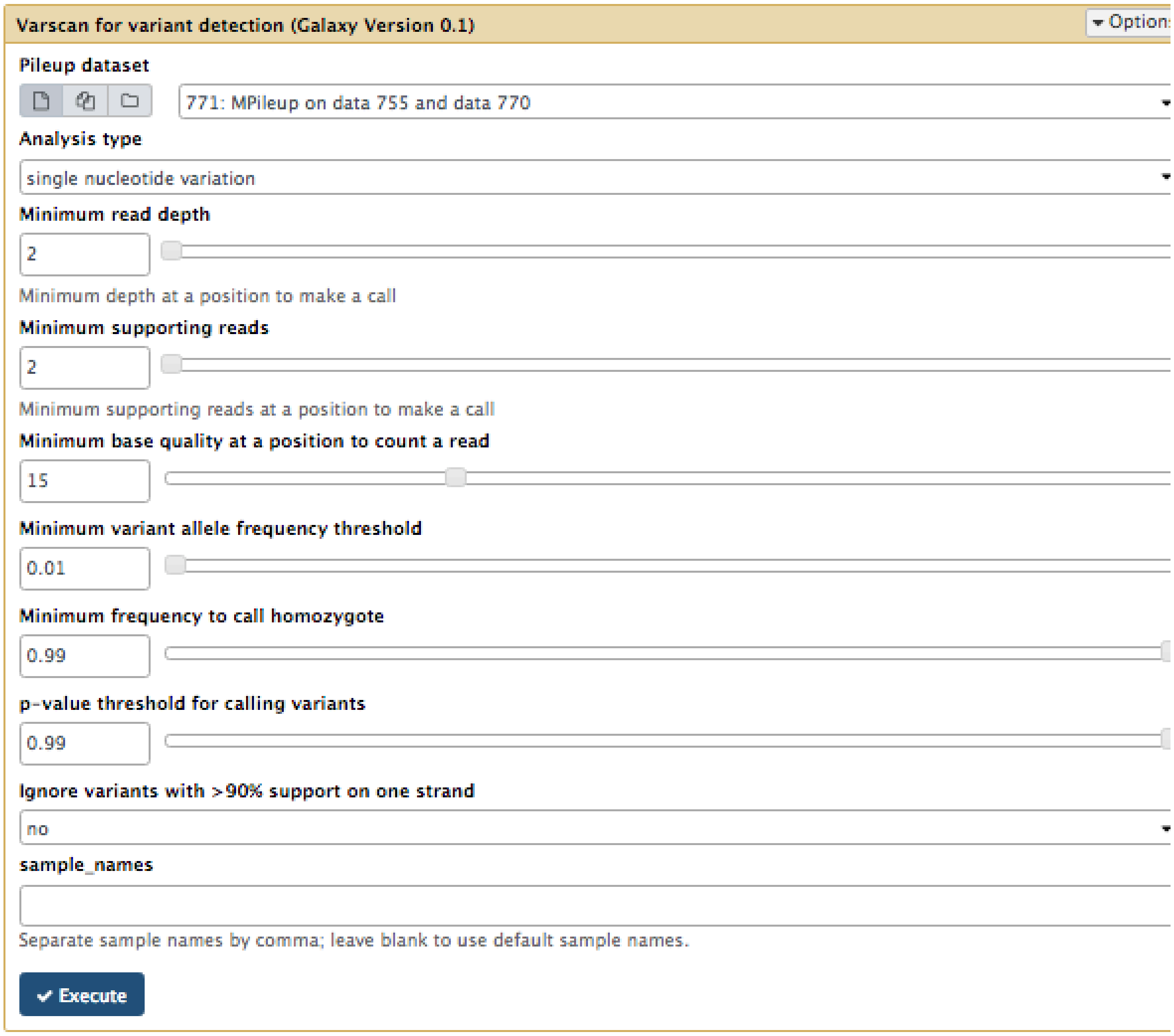
Print quality and base string?: Yes

See "Example 4" below for explanation

10 **Generate a Varscan SNP and Indel vcf file from your Mpileup file**

Use the Varscan plugin to filter SNPs on coverage and output SNPs in vcf format. These Varscan files contain the same SNP positions as the filtered pileup file, but they are in vcf format. The vcf format is necessary for input into SnpEff (candidate SNP identification) later on. Run Varscan with a coverage cutoff of 2 for low coverage data (<10) and 8 for high coverage data (>15). We also typically set the minimum frequency to call a homozygote to 0.99 as that provides a stringest cutoff for allele frequency and aids in calling homozygous SNPs later on.

Additionally, output a Varscan file calling indels. This file will be useful for verifying your mapping region and further fine mapping using large indels (>=10bp) as markers.



Varscan for variant detection (Galaxy Version 0.1)

Pileup dataset
771: MPileup on data 755 and data 770

Analysis type
single nucleotide variation

Minimum read depth
2
Minimum depth at a position to make a call

Minimum supporting reads
2
Minimum supporting reads at a position to make a call

Minimum base quality at a position to count a read
15

Minimum variant allele frequency threshold
0.01

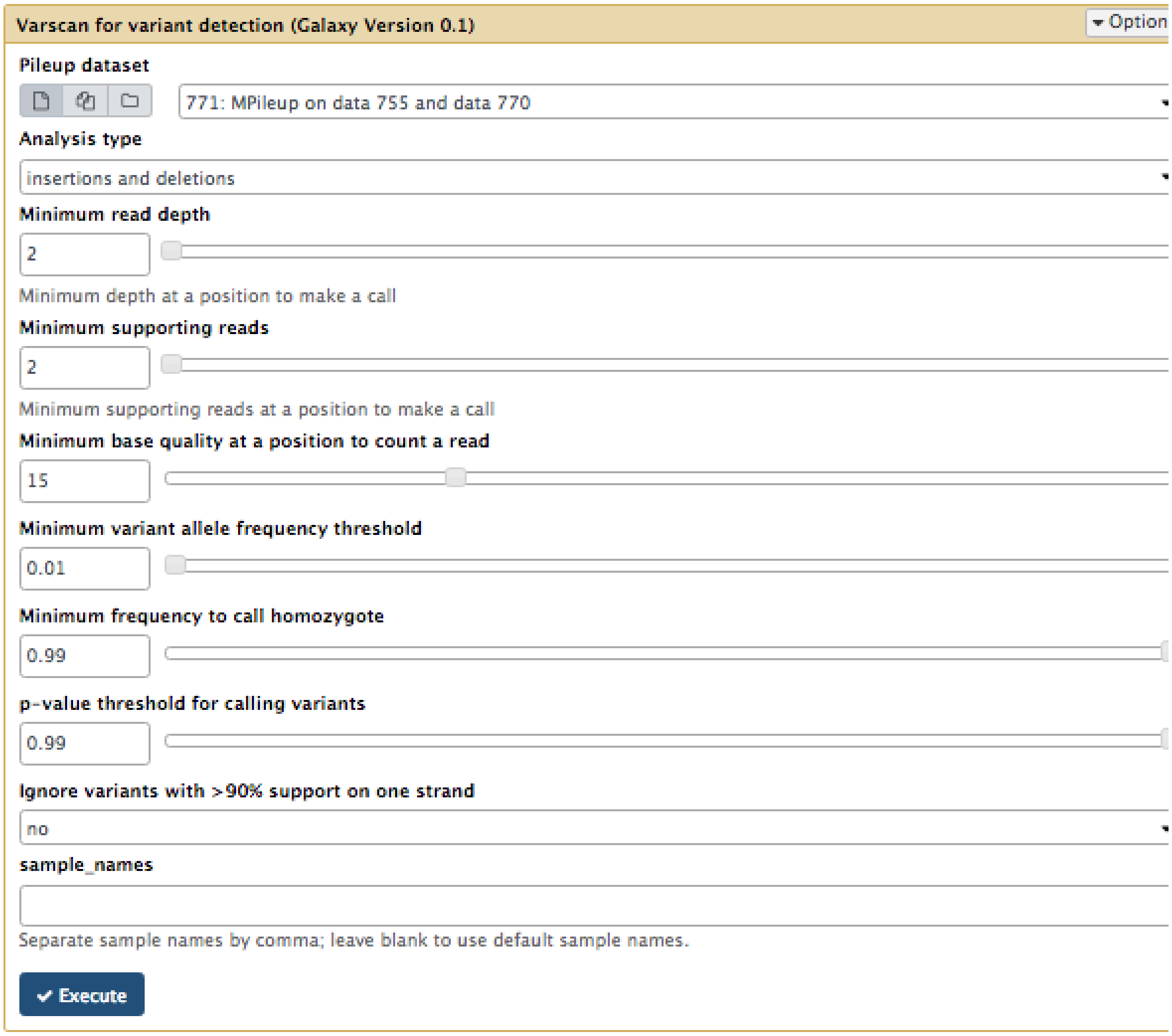
Minimum frequency to call homozygote
0.99

p-value threshold for calling variants
0.99

Ignore variants with >90% support on one strand
no

sample_names

Separate sample names by comma; leave blank to use default sample names.



Varscan for variant detection (Galaxy Version 0.1)

Pileup dataset: 771: MPileup on data 755 and data 770

Analysis type: insertions and deletions

Minimum read depth: 2

Minimum depth at a position to make a call

Minimum supporting reads: 2

Minimum supporting reads at a position to make a call

Minimum base quality at a position to count a read: 15

Minimum variant allele frequency threshold: 0.01

Minimum frequency to call homozygote: 0.99

p-value threshold for calling variants: 0.99

Ignore variants with >90% support on one strand: no

sample_names:

Separate sample names by comma; leave blank to use default sample names.

Phase 2: Plot homozygous SNPs in the filtered pileup file

11 **Phase 2 Overview**

To plot homozygous SNPs in our filtered pileup file, we use the R packages dplyr and ggplot2 in R studio. Before plotting, we first change the format of our filtered pileup file to make it easy to manipulate in R studio. Then we use dplyr to remove background SNPs in the data. And finally, we plot our data as binned homozygous SNPs along each chromosome.

12 **Change format of the filtered pileup file before importing into R**

1. In a terminal or command prompt, change your working directory to the directory with your filtered pileup file.

```
cd working_directory
```

2. Extract chromosomes from the filtered pileup file and divide the total number of deviants by the quality adjusted coverage (output to new column in the file) to get your allele frequency (f_variant). This step and all following steps will need the line of code run for each chromosome.

```
awk '{if ($1== Chromosome number e.g. 1) print $0}' filtered.pileup | awk '{$14=$13/$12; print $0}'  > chr1.freq.pileup
```

3. Filter for f_variant >0.5, to filter SNP variants that are not homozygous

```
awk '{if($14>0.5)print$0}' chr1.freq.pileup > chr1_0.5.pileup
```

4. Print position, ref, alternate, fvariant, coverage from pileup chr file

```
awk '{print$2' '$3' '$5' '$14' '$11}' chr1_0.5.pileup > chr1_filter.pileup
```

5. Convert tttt..ttTT and AaaAA..,, and GGgg..,, and CcccC..,, in 2nd column to T,A,G,C

```
awk '$3~/[Aa]/{$3="A"} $3~/[Tt]/{$3="T"} $3~/[Gg]/{$3="G"} $3~/[Cc]/{$3="C"} 1' chr1_filter1.pileup > chr1_filter2.pileup
```

Optional step 5a: Code to extract only EMS variants, may be needed if a clear mapping region is not identified.

```
awk '{if ($2=="C" && $3~/[Tt]/) print $0; else if ($2=="G" && $3~/[Aa]/) print $0 }' chr1_filter.pileup > chr1_filter2.pileup
```

Optional step 5b: Convert tttt..ttTT and AaaAA..,, in 2nd column to T and A

```
awk '$3~/[Aa]/{$3='A'} $3~/[Tt]/{$3='T'} 1' chr1_filter2.pileup > chr1_filter3.pileup
```

6. Covert to .csv

```
awk '{print$1','$2','$3','$4','$5}' chr1_filter3.pileup > chr1_EMS.csv
```

7. Add a header to your csv by first creating this csv header in a text editor and then copying into your working directory: position,ref,alt,f_variant,coverage \n

```
cat header.csv chr1.csv > chr1_final.csv
```

13 **Load R packages dplyr and ggplot2**

Start up R studio and begin a new R script:

```
install.packages('dplyr')
library(dplyr)
library(ggplot2)
```

14 **Read your files into R studio (All commands are run the same for each chromosome)**

chr1_final ← read.csv('path to chr1_final.csv')

If you have files with SNP positions from the WT background or from the maize Hapmap, read these in also.

WT_chr1 ← read.table('path to WT_chr1')
Hapmap_chr1 ← read.table('path to Hapmap_chr1')

If these files are a single column of positions, then you can simply add a header with:
colnames(WT_chr1) ← c("position")

The "position" header is essential for later steps.

15 **Optional step: Filter for coverage +- 1SD of the mean**

chr1_coverage100 ← filter(chr1_180502, coverage<=100)
chr1_summary ← summarise(chr1_coverage100,mean=mean(coverage), sd=sd(coverage))
chr1_filtercov ← filter(chr1_coverage100, coverage>="lowerlimit" & coverage<="upperlimit")

16 **Remove background SNPs**

chr1_final_NoWT ← anti_join(chr1_final, WT_chr1, by='position')
chr1_final_NoWTHapmap ← anti_join(chr1_final_NoWT, Hapmap_chr1, by='position')

17 **Plot homozygous SNPs with ggplot2**

Initial plot to figure out which chromosome has the most SNPs.

ggplot(subset(chr1_final_NoWTHapmap, f_variant>=0.99), aes(x=position)) + geom_histogram(binwidth=1000000) + expand_limits(y=c(0,chr1_final_NoWTHapmap)) +scale_x_continuous(breaks = round(seq(min(0), max(chr1_final_NoWTHapmap$position), by=20000000), 10000000)) +theme(axis.text.x = element_text(angle = 70, hjust = 1)) + theme_bw()

Adjust the y-axis for the chromosome with the highest amount of SNPs (Varies from data to data).

ggplot(subset(chr1_final_NoWTHapmap, f_variant>=0.99), aes(x=position)) + geom_histogram(binwidth=1000000) + expand_limits(y=c(0,**1000**)) +scale_x_continuous(breaks = round(seq(min(0),

```
max(chr1_final_NoWTHapmap$position), by=20000000), 10000000)) +theme(axis.text.x = element_text(angle = 70, hjust = 1)) + theme_bw()
```

The mapping region will be a region of abundant SNPs in your data.

18 **Optional step: Confirm mapping region and narrow down candidate SNPs**

Once a mapping region has been identified from your plotting, you can use the Varscan indel file to design markers and verify your mapping region. Additionally, if there are several candidate SNPs in your interval, you can use these markers to narrow down the mapping interval.

1. Extract the mapping region

```
awk ‘{if($2>=40000000 && $2<=70000000)print$0}’ chr1_varscan.vcf > chr1_mapping_region.vcf
```

2. Call homozygous Indels

```
grep HOM=1 chr1_mapping_region.vcf > chr1_mapping_region_HOM.vcf
```

Pick indels that are >=10bp for best results. Design primers over the indel, perform PCR, and run the product on a 3.5% agarose gel. Differences in band size will indicate the genetic background at that indel.

Phase 3: Candidate SNP identification with SnpEff

19 **Processing the Varscan file for input into SnpEff**

We ran these commands on a high performance computing cluster. Regular expression commands can take a long time if they are run locally.

1. Divide Varscan by chromosome

```
awk ‘{if($1==”Chromosome number”)print$0}’ Varscan.vcf > chr1.vcf
```

2. Extract mapping region (vns here)

```
awk ‘{if($2>=40000000 && $2<=70000000)print$0}’ chr1.vcf > chr1_mapping_region.vcf
```

3. Remove background SNPs (Hapmap and WT) (analyzed on MGHPCC)

```
bsub -o chr1_mapping_region_NoWT.vcf -e background.err -n 1 -R rusage[mem=4096] -W 120 -q short grep –wvFf background_SNPs.txt chr1_mapping_region.vcf
```

Alternatively use:

```
awk 'FNR==NR {hash[$0]; next} !($2 in hash)' Hapmap_chr1.txt chr1_mapping_region.vcf
> chr1_NoHapmap.vcf
```

4. Filter for EMS variants

```
awk '{if ($4=='C' && $5=='T') print $0; else if ($4=='G' && $5=='A') print $0 }'
chr1_NoHapmap.vcf > chr1_NoHapmap_EMS.vcf
```

20 **Run SnpEff on your filtered EMS SNPs**

You can build a database in SnpEff for version 4 of the maize genome using instructions from the manual (http://snpeff.sourceforge.net/SnpEff_manual.html).
Once your database is built, navigate to the SnpEff folder, move your filtered EMS SNP file there, and run the following:

```
java -jar snpEff.jar Zmv4 chr1_NoHapmap_EMS.vcf
```

We submitted this command to our computing cluster with additional flags.

```
bsub -o vns_SNPeff.vcf -e vns_SNPeff.err -n 4 -R rusage[mem=4096] -W 120 -q short
```

bsub = command to submit a job
-o = output file
-e = error log file
-n = number of computing cores
-R = amount of memory per core
-W = run for a max of 120 minutes
-q = submit to short queue

21 **Filter your SnpEff file**

1. Filter for Moderate effect SNPs

```
grep MODERATE vns_SNPeff.vcf > MODERATE.vcf
```

2. Filter for High effect SNPs

```
grep HIGH vns_SNPeff.vcf  > HIGH.vcf
```

3. Filter for homozygous SNPs in both your Moderate and High effect files.

```
grep HOM=1 MODERATE.vcf > MODERATE_HOM.vcf
grep HOM=1 HIGH.vcf > HIGH_HOM.vcf
```

22 **Annotate Moderate and High effect genes**

Download the version 4 gene annotations from maizesequence.org (ftp://ftp.gramene.org/pub/gramene/CURRENT_RELEASE/gff3/zea_mays/gene_function).

Use this file and your Moderate and High effect vcf to annotate your genes.

First, extract gene IDs from your Moderate and High effect vcf files

1. Cut out "gene:Zm" from your vcf

```
awk -F'|' '{print$5}' MODERATE.vcf > MODERATE_filter1.vcf
```

2. Remove 'gene:'

```
awk -F':' '{print$2}' MODERATE_filter1.vcf > MODERATE_gene_IDs.txt
```

Then extract your Moderate IDs from the annotation file.

```
grep -wf MODERATE_gene_IDs.txt B73v4.gene_function.txt
```

23 **Optional step: Use Provean to assess Moderate SNPs**

You can use Provean to determine if a Moderate SNP is likely to be deleterious (http://provean.jcvi.org/seq_submit.php). The accuracy of Provean is about 77.9% and may miss your candidate SNP. Additionally, Provean is low throughput for non-human genomes, so for large numbers of Moderate SNPs an alternative program or narrowing down the mapping interval may be more successful for candidate SNP identification.