BISC577 Assignment

(Snakemake and Genome Assembly)

by, Raktim Mitra and Jiawei Huang

1. Add a rule to map reads from the two species

   We modified the snakemake file and change the **all_sra=["SRR5762776", "SRR6765736"]** to map reads from the two species.
   We also changed the **rule all** command and change **bam** and **bamIndex** with expand command to use all_sra.(see Assignment.2.2.snakefile)

2. Add a rule and Call variants using freebayes.

   Use default command "**freebayes -f ref.fa aln.bam >var.vcf**" to call variants. where ref.fa is reference genome, aln.bam is two species needed to call variant. var.vcf is the output file.(see Assignment.2.2.snakefile)

3. Compare the variants between the two files.

   -How many SNV calls are there per drosophila sample?
   calculate the rows of each file(without comment rows) can give us the number of SNV calls for each file.(see q3_123.h)

   - How many homozygous SNVs?
   - How many heterozygous SNVs?

   counts for SRR5762776.freebayes.vcf
   homo: **192834** hetero: **551782** total: **744616**
   counts for SRR6765736.freebayes.vcf
   homo: **95386** hetero: **734331** total: **829717**

   We noticed that in the output vcf file, there are two fields, one is **FORMAT**, the other one is **sample** column. The short names of the sample-level annotations are recorded in the FORMAT field. The annotation values are then recorded in corresponding order in each sample column.
   The content of FORMAT is GT:DP:AD:RO:QR:AO:QA:GL. the GT flag is the genotype of this sample at this site. So we can judge the homozygous and heterozygous according to GT flag. if they are the same then it is homozygous otherwise it is heterozygous. This can be implemented via awk command, where we check for each row of the vcf file **if the type is SNP and if the two alleles are the same or not**, and accordingly increment homozygous or heterozygous counts and return the final count.(see q3_123.sh)

-How many SNVs are shared between the two species?
Number of SNVs shared between two species: **546120**

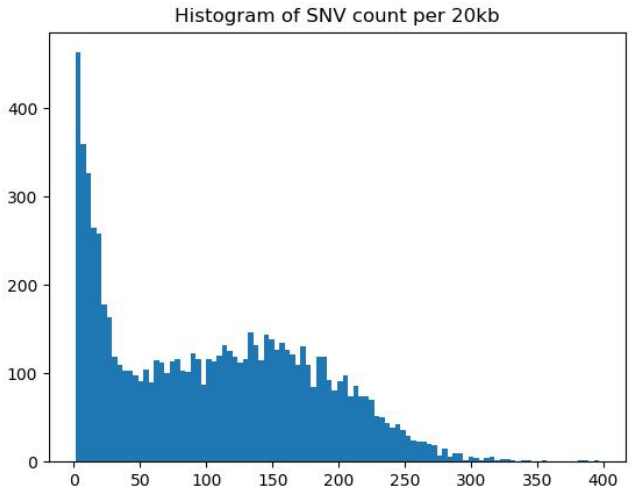
We use **bedtools intersect** command to calculate the shared SNVs of the two species. The output of bedtools intersect will give us a file containing all the overlaps between two sets of genomic features. Each row is a shared variant.
Just count the number of rows will give us the result of shared SNVs
(see q3_4.sh)

- Compute the SNV density in 20kbp bins, plot a histogram of SNV density.

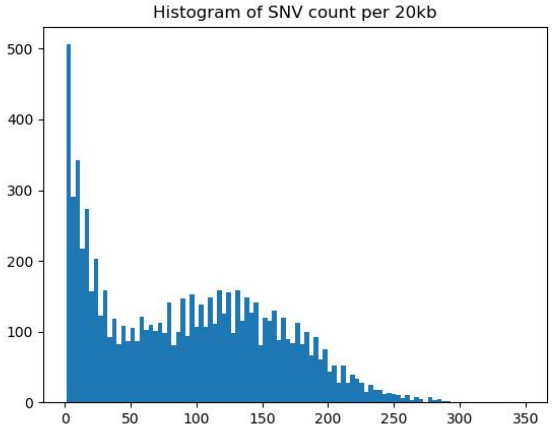
We first group the SNV by chromosome and then calculate SNVs every 20kbp on each chromosome. We used an awk script to generate a SQL script which runs positions of all the SNVS and then group, create 20kbp windows and returns required count of SNVs in each of the windows. We use these counts to plot the histograms with 100 bins. (see q3_5.sh).

reference: `https://www.biostars.org/p/218480/`

SRR6765736.freebayes.vcf:

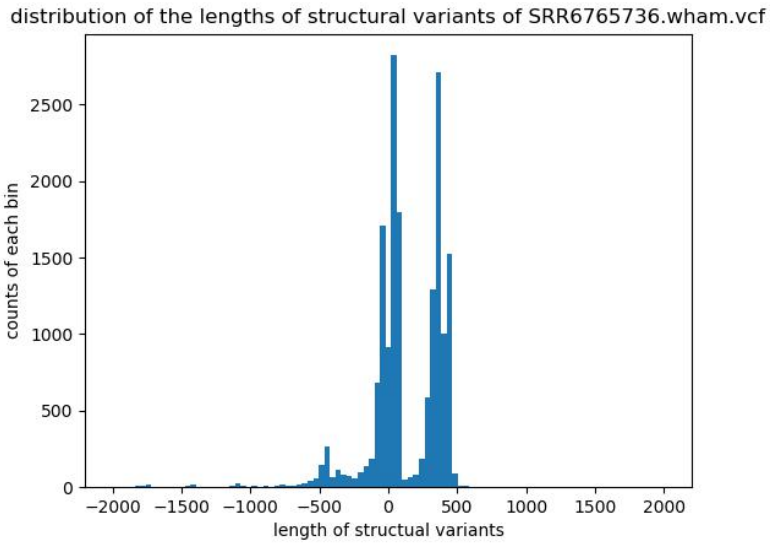


SRR5762776.freebayes.vcf :

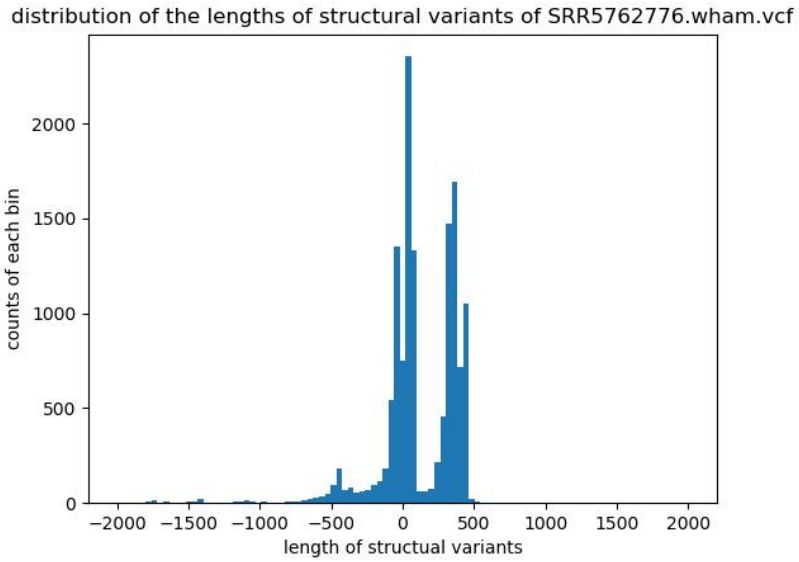


4. Call structural variants using wham. Plot a distribution of the lengths of structural variants in each species.
   We first add a rule called **CallStructuralVariant** to the snakemake file. The output is also a vcf file. The field SVLEN in the VCF file gives the expected length of the SV. So just pick it out and plot a histogram. (because there are lots of shorts length in the file but there is still some extreme large length in the vcf file which contributes to large x scale, we choose to cut it off to shorter scale)

   SRR6765736.wham.vcf:



   SRR5762776.wham.vcf:



6. Assemble each genome using minia

We first add a rule named CallMinia to assemble each genome.(k=31 for kmer size) and then we use samtools faidx to do the calculation(see Assignment.2.2.snakefile).

7. How much of the genome is assembled into contigs greater than 2kbp?

When we get the .fai index file. The second column of the file is contig length(bp). So we will only count those contigs whose length is greater than 2k.

Use command
”awk '{if ($2 >= 2000) a++};END{print "contigs greater than 2kbp in $FILENAME: " a}' SRR6765736_31.contigs.fa.fai” will get the result of SRR6765736_31. And the same for SRR5762776_31.

Result:
in SRR6765736 :
contigs greater than 2kbp : **16765**
in SRR5762776 :
contigs greater than 2kbp : **17268**

8. Create another rule to assemble with a different k-mer. Reassemble, and compute the N50 of the two assemblies.

ref:https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics
first sort the index file from large to small and get an ordered list of contigs C1>C2>...>Cn and then we sum the length of contigs up to find the 50% total length. Finally find the particular contig i which C1+C2+...+Ci > 50%total length.
make k = 22 to see the output.

So we first use **sort** command to sort the length from large to small and choose the specific row which reach half the total length. example code as following
One Example :

```
sort -k2 --numeric-sort --reverse  fai_file | awk '{t+=$2; n50=0; if (t>total_length/2 && n50==0) {n50 = $2; print n50}}' | head -n 1
```

A different runnable script for the above using awk and bash can be found in q8.sh

Results:
required N50 statistic for SRR6765736_22.contigs.fa.fai is **654**
required N50 statistic for SRR6765736_31.contigs.fa.fai is **2446**
required N50 statistic for SRR5762776_22.contigs.fa.fai is **767**
required N50 statistic for SRR5762776_31.contigs.fa.fai is **3130**

Note: All calculations and plots are done through calling the analysis.sh script from snakemake for each SRR file.
