Brad Olinger

Quant Bio

Week 2

9/17/21

Step 1: Index the sacCer3 genome with bwa index

Command: bwa index -p sacCer3.fa -a is sacCer3.fa

### Step 2: Alignment with bwa mem

Note: use -R flag to add a line as a header for each file to keep track of for step 4

Command: bwa mem -t 4 -R "@RG\tID:A01\_63\tSM:A01\_63" sacCer3.fa A01\_63.fastq > Sam63.sam

Note: replace the R tag, fastq file name, and output file for each of the 10 samples.

Note: -t changes the number of threads used

2nd Corrected command #2: bwa mem -t 4 -R "@RG\tID:A01\_63\tSM:A01\_63" -o output/63.sam sacCer3.fa A01\_63.fastq

### Step 3: Create a sorted bam file with samtools, for input to variant callers

Perhaps consider the -O and -o flags.

-o defines the out file name

-O defines the output file type

Command: samtools sort -o out63.bam -O bam Sam63.sam

Note: rerun this line for each sample

### Corrected command: samtools sort -O BAM -o out63.bam Sam63.sam

**2nd corrected command:** samtools sort -O BAM -o output/63.bam output/63.sam

### Step 4: Variant calling with freebayes

Use freebayes to identify genetic variants in all of your yeast strains concurrently. It will output results in Variant Call Format (VCF). You should consider using the -f, --genotype-qualities, and -p flags.

Note: set -p (ploidy) to 1 for haploid samples

Command: samtools index out09.bam

Use for each bam file

#This indexes each bam file

Command: freebayes -f sacCer3.fa -p 1 -= out\*.bam > varcall.vcf

Corrected command: freebayes -f sacCer3.fa -L bamnames.txt --genotype-qualities -p 1 > varcall.vcf

#checking order of chromosomes

Samtools idxstats out09.bam

### Step 5: Filter variants based on genotype quality using vcffilter

Filter your VCF so that you only keep variants whose estimated probability of being polymorphic is greater than 0.99. You should consider how to do this with the -f flag. The freebayes documentation will be helpful here, as well as [this vcffilter info](https://github.com/vcflib/vcflib#vcffilter).

Command: vcffilter varcall.vcf -f "QUAL > 20" > results.vcf

Note: Quality score >20 indicates only a p=0.01 of the null hypothesis (not being polymorphic).

### Step 6: Decompose complex haplotypes using vcfallelicprimitives

We suggest using the -k and -g flags to keep annotations for the variant sites and sample genotypes in your VCF.

Command: vcfallelicprimitives results.vcf -k --keep-info -g --keep-geno > decom.vcf

### Step 7: Variant effect prediction with snpeff ann

First, fetch the appropriate yeast reference database:

snpeff download R64-1-1.99

Then, use snpeff ann to annotate your VCF with the predicted functional effects that these genetic variants may have.

We recommend not Googling the snpeff documenation. It will tell you to use java -jar snpEff.jar, which you should not. The help option for snpeff ann’s command-line tool is 100 times better.

Command: snpeff ann R64-1-1.99 decom.vcf > prediction.vcf

### Step 8: Exploratory data analysis through plotting

In Python, produce a nicely formatted and labeled multi-panel plot describing your variants.  
  
Explore each of the following characteristics of the variant genotypes called across all ten yeast samples. (Each characteristic will be a subplot in the multi-panel plot).

* The read depth distribution of variant genotypes (histogram)
  + This information can be found in the sample specific FORMAT field for each variant/line. Check the file header to decide which ID is appropriate.
* The quality distribution of variant genotypes (histogram)
  + This information can be found in the sample specific FORMAT field for each variant/line. Check the file header to decide which ID is appropriate.
* The allele frequency spectrum of your identified variants (histogram)
  + This information is pre-calculated for you and can be found in the variant specific INFO field. Check the file header to decide which ID is appropriate.
* A summary of the predicted effect(s) of each variant as determed by snpEff (barplot)
  + This information was added to the VCF by snpEff and can be found in the variant specific INFO field. Check the file header to decide which ID is appropriate and how to parse the information.
  + We encourage you to consider every possible effect for each variant, but feel free to just grab the first one.

You may find it helpful to reference [this page](https://pcingola.github.io/SnpEff/se_inputoutput/) of the snpeff manual, which describes the format of its output VCF.

All on python script