SNP calling for genome sequencing

2024-09-03

Background & uses

Prior to this, all sequences should have gone through the preprocessing pipeline (titled sequencing-preprocessing.pdf in the oyster-popgen repository). If that has not been completed, none of the SNP calling outlined in this tutorial will work. All scripts and analyses are run on UNH's computer cluster, premise, unless otherwise noted.

Creating an anaconda environment

For ease of analysis, you will need to create an anaconda environment on premise to run all of your analyses. To do this, start by running the following code:

```
module load anaconda
conda create --name snp-calling
```

When it asks to proceed, type in "y" to complete creating the anaconda environment. Once it is created, you will need to activate it and add in the necessary packages. Each time you activate your environment, make sure to load anaconda first with the code above (just the module load anaconda, not the conda create). To activate and add in the packages, run the following lines of code:

```
conda activate snp-calling
conda install -c bioconda freebayes
conda install -c bioconda vcftools
conda install -c bioconda admixture
conda install -c bioconda bwa
conda install -c bioconda samtools
conda install -c bioconda gatk
conda install -c bioconda plink
conda install -c bioconda admixture
git clone https://github.com/esrud/currentNe
cd currentNe
make
make static
```

Each time the system asks if you would like to proceed, make sure to type "y" and hit enter to continue installing the packages.

Once this has been completed, all necessary packages, as well as the currentNe directory with the appropriate scripts, will be installed.

SNP calling

First, you will need to copy the slurm script for SNP calling from the ecogen shared directory into your personal directory. To do so, run the following lines of code:

```
cp /mnt/home/ecogen/shared/scripts/snp-calling.slurm /path/to/your/directory/scripts/snp-calling.slurm
```

Once you have that copied into your scripts directory, open the script with vim editor and make sure to change all the path names and file names to your specific directories and file names. Next, navigate to the directory with all of your aligned bam files. FreeBayes requires a list of all files that are to be considered when SNP calling. To generate that list, in the directory with your aligned bam files, run the following code:

```
ls *align.bam > bam.fofn
```

Once completed, you can run the snp-calling.slurm script:

```
sbatch snp-calling.slurm
```

This script may take a while to run, as SNP calling is a very tedious process. This can take days depending on the size of your files and how many samples you have. You can look at the output file to get a sense of where the script is out. Just be patient!

SNP filtering

Once you have all SNPs called, you must filter the SNPs. The parameters in which you filter them may vary depending on the type of analysis you are doing, the population size, etc. Make sure to go through and adjust them for your project. For this example, I will be using the parameters used in the oyster population genomics paper. First, all indels should be removed. Then, the SNPs should be filtered based on how many individuals they are present in, minimum quality, minor allele count, and minimum depth. Once filtered, individuals with excessive missing data should be removed, and then more stringent SNP filtering should occur.

In the ecogen shared scripts directory, there is a slurm script named vcf-filtering.slurm. This script contains all of the filtering steps that are required for SNP filtering. Each code chunk in the script will need to be run one at a time, so make sure to follow the instructions for the next steps clearly. Copy the filtering script into your personal scripts directory using the following:

```
cp /mnt/home/ecogen/shared/scripts/vcf-filtering.slurm /path/to/your/directory/scripts/vcf-filtering.slu
```

Make sure to open the file and write in your file names and file paths. You will notice all of the vcf commands have a # in front of them, which comments them out so they do not run.

Step 1: remove indels from vcf file

Open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Remove the # from the lines under Step 1. The code should look like this:

```
# Step 1: remove indels from vcf file
vcftools --vcf your-vcf-file.vcf  --remove-indels --recode
--recode-INFO-all --out your-vcf-file-snps-only.vcf
```

The rest of the file below step 1 should have # before each and every line. Once that is complete, you can run the code with the following command:

```
sbatch vcf-filtering.slurm
```

Once completed, the output file with list how many variant sites you started with and how many are left after filtering. Make sure to write down these numbers for each step to see how the number of variant sites decreases with each filtering step.

Step 2: filter SNPs based on minor allele count, quality score, and how many individuals they are present in

Open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Put # in front of each line of the previous step. Remove the # from the lines under Step 2. The code should look like this:

```
# Step 1: remove indels from vcf file
#vcftools --vcf your-vcf-file.vcf  --remove-indels --recode
#--recode-INFO-all --out your-vcf-file-snps-only.vcf


# Step 2: filter SNPs based on minor allele count, quality score,
#and how many individuals they are present in
vcf=your-vcf-file-snps-only.vcf.recode.vcf
vcftools --vcf $vcf \
   --max-missing 0.5 \
   --mac 3 \
   --minQ 30 \
   --recode \
   --recode-INFO-all \
   --out snps-filtered-1
```

Make sure to change the filtering parameters based on your study. Once the vcf-filtering.slurm script is edited, save it and run with the following command:

```
sbatch vcf-filtering.slurm
```

Once completed, the output file with list how many variant sites you started with and how many are left after filtering. Make sure to write down these numbers for each step to see how the number of variant sites decreases with each filtering step.

Step 3: filter based on minimum depth

Open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Put # in front of each line of the previous step. Remove the # from the lines under Step 3. The code should look like this:

```
# Step 2: filter SNPs based on minor allele count, quality score,
#and how many individuals they are present in
#vcf=your-vcf-file-snps-only.vcf.recode.vcf
#vcftools --vcf $vcf \
#   --max-missing 0.5 \
#   --mac 3 \
```

```
#   --minQ 30 \
#   --recode \
#   --recode-INFO-all \
#   --out snps-filtered-1


# Step 3: filter based on minimum depth
vcftools --vcf snps-filtered-1.recode.vcf \
   --minDP 3 \
   --recode \
   --recode-INFO-all \
   --out snps-filtered-2
```

Make sure to change the filtering parameters based on your study. Once the vcf-filtering.slurm script is edited, save it and run with the following command:

```
sbatch vcf-filtering.slurm
```

Once completed, the output file with list how many variant sites you started with and how many are left after filtering. Make sure to write down these numbers for each step to see how the number of variant sites decreases with each filtering step.

Step 4: individuals with missing data

Step 4 requires 3 parts: make a list of all individuals and their percentage of missing data, create a list of individuals with excessive missing data, and remove those individuals from the analyses. To begin, open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Put # in front of each line of the previous step. Remove the # from the lines under Step 4.1. The code should look like this:

```
# Step 3: filter based on minimum depth
#vcftools --vcf snps-filtered-1.recode.vcf \
#   --minDP 3 \
#   --recode \
#   --recode-INFO-all \
#   --out snps-filtered-2


# Step 4.1: create list of individuals and how much data they are missing
vcftools --vcf snps-filtered-2.recode.vcf \
   --missing-indv
```

Once the vcf-filtering.slurm script is edited, save it and run with the following command:

```
sbatch vcf-filtering.slurm
```

The output of this will be a file titled out.imiss, which contains each individual's amount of missing data, which will be needed for the next step. For the next step, open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Put # in front of each line of the previous step. Remove the # from the lines under Step 4.2. The code should look like this:

```
# Step 4.1: create list of individuals and how much data they are missing
#vcftools --vcf snps-filtered-2.recode.vcf \
#   --missing-indv


# Step 4.2: make list of individuals missing more than 50% of data
mawk '$5 > 0.5' out.imiss | cut -f1 > lowDP.indv
```

Once the vcf-filtering.slurm script is edited, save it and run with the following command:

```
sbatch vcf-filtering.slurm
```

The output of this will be a file titled lowDP.indv, which contains all individuals that have excessive missing data. In this case, the cut off is 50% – if an individual is missing more than half the data, they are excluded from the study. Make sure to write down the list of individuals and their percent missing data for later. For the next step, open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Put # in front of each line of the previous step. Remove the # from the lines under Step 4.3. The code should look like this:

```
# Step 4.2: make list of individuals missing more than 50% of data
#mawk '$5 > 0.5' out.imiss | cut -f1 > lowDP.indv


# Step 4.3: remove individuals that have excessive missing data
vcftools --vcf snps-filtered-2.recode.vcf \
   --remove lowDP.indv \
   --recode --recode-INFO-all \
   --out snps-filtered-3
```

This last step will remove the individuals from the vcf file that have excessive missing data. Once the vcf-filtering.slurm script is edited, save it and run with the following command:

```
sbatch vcf-filtering.slurm
```

Step 5: filter SNPs based on how many individuals they are present in, minor allele frequency, and depth

Open the vcf-filtering.slurm file you copied into your scripts directory with the vim editor. Put # in front of each line of the previous step. Remove the # from the lines under Step 5. The code should look like this:

```
# Step 4.3: remove individuals that have excessive missing data
#vcftools --vcf snps-filtered-2.recode.vcf \
#   --remove lowDP.indv \
#   --recode --recode-INFO-all \
#   --out snps-filtered-3


# Step 5: filter SNPs based on how many individuals they are present in,
# minor allele frequency, and depth
vcftools --vcf snps-filtered-3.recode.vcf \
   --max-missing 0.95 \
```

```
  --maf 0.05 \
  --recode \
  --recode-INFO-all \
  --out snps-filtered-4 \
  --min-meanDP 20
```

Make sure to change the parameters based on the specifics of your study. Once the vcf-filtering.slurm script is edited, save it and run with the following command:

```
sbatch vcf-filtering.slurm
```

Once the code has run, you will have your final, filtered vcf file that only contains SNPs. The vcf file is now ready for downstream analyses. Have fun!