GATK Variant Calling Pipeline

This is starting from trimmed reads in FASTQ files. The scala script (called QScript by the GATK people) runs the following steps, facilitating parallelization to make things go a bit faster:

1. BWA alignment with reference genome

2. Mark duplicate reads using Picard

3. Run GATKs HaplotypeCaller, generating intermediate gVCF files for each sample. These are good to keep if you ever add more samples or something like that.

4. Run GATKs GenotypeGVCFs, doing the joint genotyping and generating ‘raw’ VCF (Variant Call Format) file with SNPs/INDELs for all samples

5. Run GATKs VariantFiltration, tagging sites and samples with low quality and/or low confidence calls. The final, filtered VCF file will be called 05\_variants\_filt.vcf.gz

Preparation:

1. Copy the scripts, config and filter files into your directory from

/home/gwynhn/Ahal/EastAsiaAhal/02\_CallSNPsGATK

You need VariantCallingPipeline.scala, GATKinputConfig.txt, filters.txt, and GtFilters.txt. Then, change sample/genotypes names, libraries, paths to fastq files etc in the configuration file GATKinputConfig.txt. Depending on what you’re doing, you might want to change the filters also. The current ones aren’t super strict since we didn’t have so many samples.

2. Make sure your directories on the server are set up how you want them, it will output everything into one directory.

3. To set up the environment, you need to run the following commands:

$ source /home/gwynhn/AhalSNP/analysis/etc/modules

$ module load seq/bwa/0.7.12

$ module load seq/samtools/1.3.1

$ module load seq/picard/2.6.0

$ export LD\_LIBRARY\_PATH=/usr/local/ge62/lib/lx24-amd64

You should symlink the reference from /srv/kenlab/genomes/ (or where ever) into which ever directory you’re running the pipeline in, something like:

$ ln -s /srv/kenlab/genomes/species.fa species.fa

Make sure the proper reference index files are present in the same directory as the reference itself. You need species.fa.fai and species.dict. You also need the index files for BWA, but it produces so many I honestly have no clue which one is actually required, so maybe just run the `bwa index` command regardless. If they don’t already exist in the genomes folder, you can create them as follows:

$ bwa index Alyr.fa

$ samtools faidx Alyr.fa

$ java -jar $picard\_jar CreateSequenceDictionary R=Alyr.fa O=Alyr.dict

The whole command for running the pipeline through GATK’s Queue is:

$ java -Djava.io.tmpdir=/scratch -jar /srv/kenlab/GwynHN/masters/ModifiedSeqTools/gatk3.6-protected/target/executable/Queue.jar -S VariantCallingPipeline.scala -runName "AlyrSNPcall" -runDir . -R Alyr.fa -libs "GATKinputConfig.txt" -filters "filters.txt" -gtfilters "GtFilters.txt" -t 8 -ct 8 -sj 12 -maxMem 256 -qsub -jobQueue "GT" -jobNative "-l h=fgcz-c-047" -memLimit 32 -jobParaEnv "smp" -run |& tee alyrNovi.log

All you should change is the runName, the directory you’re working in (-runDir) and the reference genome (-R). The last part I highlighted does two things: -run actually runs it, but if you want to test that the script compiled and the program can find and read everything in, leave out the `-run` argument. The `|& tee file.log` is so that the output that you see on the screen is also written into file.log. I do this because it’s easier for me to have all the messages in one place, while GATK’s built in logs are kind of scattered. That part is not required to run the pipeline, though, just user preference.