Multicalling pipeline

Repository of scripts for generating a high-quality WES callset

The pipeline consists of three parts: part 1 covers the pre-processing of each sample in isolation; part 2 covers the steps to combine samples and part 3 shows how to apply adequate QC filters.

Part 1 - Sample pre-proccessing

1. First step is aligning FASTQ data against reference genome using the `ALIGN.sh` script. In the first lines of the script, replace the paths to the desired reference and the dbsnp dataset. Remember, dbsnp has to be un-compressed to work. Alignment would take approximately 3 hours with 40 processors. Packages required include: *biobuilds/2017.05*, *picard* and *GATK v 4.0*+. GATK 4 base recalibration works better than previous versions.

2. Second step is generating the g.vcf file. This step is done using the `CALL.sh` script. The script is optimised for 40 processors and uses the HaplotypeCaller algorithm. An additional parameter *--dontUseSoftClippedBases* forces the caller to ignore soft clipped bases during the local realignment for calling. This underpowers the detection of indels, but increases the confidence when calling SNVs.

Part 2 - Combine samples

Once all the samples were processed with these two scripts, it's time to combine them all together. Because of the computationally intensive process, samples have to:

1. Combine samples in batches (~20 to 30 samples per batch). This step is performed with the `combiner.sh` script. The optimal strategy is to create a job array and a set of batch_XX.list imput file. E.g. From a list of all g.vcf run `split -d -l 20 list_of_vcf batch_` this command will split the file "list_of_vcf" into n files each containing 20 lines and will name them "batch_0, batch_1, ..." When running the array, each job will have a TASK_ID and the combiner.sh scrip will look for the inputs in the batch_0 file and generate the batch_0.g.vcf.gz file. Remember to rename all the batches to end in ".list" (e.g. batch_0.list) otherwise GATK will fail.

2. Genotype all the batches together by chromosome (~2h at 40 proc per chr1). This step is pefrormed with the `gtyper.sh` script. In this case the input is the list of all batches and the job array iterates through the chromosomes (array from 1 to 22). Chromosomes X and Y must be sbmitted individually. The result is 22+X+Y vcf files containing all the samples

3. Concatenate (CombineVariants) all chromosomes. This step is performed with the `catvars.sh` script and simply take all the chromosome vcf (provided in the all_chr.list file) and concatenates all the callset in a single vcf.

‼️At this stage the multicalling VCF file is unfiltered and not restricted to any capture kit bed file. The restriction to desired bed file can be achieved using `GATK SelectVariants` and the `-L xxx.bed` parameter.

Part 3 - QC and filtering

This parts describes the steps and script for filtering variants from WES as described by [Carson et al. BMC Bioinformatics 2014, 15:125] (http://www.biomedcentral.com/1471-2105/15/125). As recommended, filtering has to be applied batch-wise (e.g. different capture or sequencing technology). The scripts here provided assumes that Part 2 generated a single VCF containing IBD cases and Controls.

1. Follow the instruction in the `Filtering.sh` script. Requires also the `meanGQ_filter.sh` script
2. Run the `Recalibrate.sh` script
3. Output of the recalibration script has to be filtered to retain only PASS variants according to chosen tranche. This can be done with vcftools (--remove-filtered-all)

👏👏 The output VCF file is now ready for analysis/annotations with your favourite tools

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3


EMossotto Enrico Mossotto


isstafford

Garyl01 Gary Leggatt

Languages

- **Shell** 100.0%