# Whole genome sequencing analysis of snow sheep

1 *Works for me*

Maulik U.

ABSTRACT

This protocol contains the codes associated with the Molecular Ecology-publication, "Whole genome sequencing reveals a complex introgression history and the basis of adaptation to subarctic climate in wild sheep". Note that the customized codes mentioned in this protocol can be accessed on this GitHub repository (https://github.com/BioInf2305). In case, some specific codes are not available in the repository, feel free to contact on: U.Maulik@gen.vetmed.uni-muenchen.de

PROTOCOL INFO

Maulik U.: Whole genome sequencing analysis of snow sheep. **protocols.io**
https://protocols.io/view/whole-genome-sequencing-analysis-of-snow-sheep-bqfsmtne

MANUSCRIPT CITATION please remember to cite the following publication along with this protocol

Whole genome sequencing reveals a complex introgression history and the basis of adaptation to subarctic climate in wild sheep

CREATED

Dec 04, 2020

LAST MODIFIED

Sep 13, 2021

PROTOCOL INTEGER ID

45266

COMMANDS

python script to generate slurm file to run the alignment

**python GenerateShFiles.py ERR157942.txt ERR157942**
linux

sbatch to submit the job on SLURM

**sbatch RunAlignmentERR157942.sh**
This command will run the bash file that will run the following steps: (1) Run sickle to trim the data (2) Run Fastqc and extract warning and failure concerning the data (3) Run bwa to align the data (4) Run samtools to sort the data (5) Run picard to mark and remove the duplicates (6) Run GATK for indel realignment (7) Calculate the average coverage
Linux

python

**python RunVariantCallingParallelSamtoolsForMapQ.py -b Argali.txt -r ~/data/Shared/References/Oar_v4.0/GCF_000298735.2_Oar_v4.0_genomic.fna -i SheepGenomeIndex.txt -c 30 -n 10000000**
This python script will call the consensus allele for each position in the genome in the bcf format.
Linux

python

**python makeConsensusBcf.py NC_019458.2_merged.bcf Sample_Depth.txt**

**NC_019458.2 NC_019458.2_OvisMus.consensus.fa**

This python script will make the consensus fasta file based on the consensus allele generated at each position in the previous step. The position at which the depth is less than third or greater than third of the average coverage will be replaced with missing genotypes.

Linux

cat

**cat NC_0194{58..83}.2_OvisAmn.consensus.fa > WholeGenome_OvisAmn.consensus.fa**

Combined the consensus for each chromosome generated in the previous step.

python

**python MergeCdsPerGene.py WholeGenome_OvisArs.consensus.fa FinalmRNACdsCord_1_to_1.bed OvisArs OvisArs.wholeGenomeCds.fa**

Extract coding sequences for each gene from the consensus references generated in the previous step.

Linux

bioawk

**bioawk -v seqCds="" -c fastx '{seqCds=seqCds$seq}END{printf ">OvisNiv""\n"seqCds"\n"}' WholeGenome_OvisNivCds.fa > WholeGenome_OvisNivCds.combined.fa**

Concatenate entire cds (super-matrix) method, remove header

linux

python

**python3 RunVariantCallingParallelSamtools.py -b AllBamFilesWithGoat.txt -r ~/data/Shared/References/Oar_v4.0/GCF_000298735.2_Oar_v4.0_genomic.fna -i SheepGenomeIndex.txt -c 40 -n 10000000**

This python script will carry out SNP calling in parallel using bcftools mpileup.

bcftools

**bcftools filter -g 3 NC_019458.2_merged.bcf -O b -o NC_019458.2.g3.bcf**
**bcftools view --exclude-types indels -e 'QUAL<40 || INFO/MQ <40' -O b -o NC_019458.2.g3NoIndelsSnpQMapQ40.bcf NC_019458.2.g3.bcf**
**python3 RemoveGoatOnlySnps.py NC_019458.2.g3NoIndelsSnpQMapQ40.bcf NC_019458.2_merged.RmGoatg3NoIndelsSnpQMapQ40.bcf NC_019458.GoatSpVar.bcf**
**python3 BcfIdentifyMissingSites.py NC_019458.2_merged.RmGoatg3NoIndelsSnpQMapQ40.bcf Sample_Depth.txt NC_019458.2_AllFiltered.bcf**

These commands will filter the SNPs

bcftools

**bcftools concat NC_0194{58..83}.2_AllFiltered.bcf -o WholeGenomeAllFiltered.vcf.gz -O z**

Command to concatenate the bcf files that were separated chromosome-wise

python

**python GetSummaryStats.py NC_019458.2_AllFiltered.bcf SamplePopIds.txt NC_019458 Stats**

calculate per sample and per population SNP count statistics
ubuntu

python

**python DrawBarPlots.py TotalSnpSharingStats.txt SpeciesNamesColorsUpdated.txt SnpStatistics**
Draw barplots of summary SNP statistics generated in the previous step.

python

**for z in {19458..19483};do python ExtractSpeciesSpecificSnps.py NC_0${z}.2_AllFiltered.bcf SampleIds1.tab NC_0${z}.2_SnowSheep_Specific SnowSheep;done**
Calculate snow sheep specific SNPs

python

**python CalcWattersonTheta.py NC_019458.2_AllFiltered.bcf 100000 SamplesIds1.tab  NC_019458_100kb_theta.txt**
Calculate watterson's theta in 100 kb window

mlRho

**mkdir mlRho_${1}**

**samtools view -b ${1}_rh.rmDuplIndelRealigned.bam | samtools mpileup -d $3 -Q 20 -q 20 -|~/software/MlRho_2.9/sam2pro -c 5 > ./mlRho_${1}/${1}.pro**

**cd mlRho_${1}**

**~/software/MlRho_2.9/formatPro -c $2 ${1}.pro**

**~/software/MlRho_2.9/mlRho -M 0 > ${1}.out**

**cd ..**
Calculate population mutation rate using mlRho

modeltest-ng

**/home/maulik/software/modeltest/modeltest-ng-static -d nt -i AllSpeciesCds.fa -o AllSpeciesCdsModeltest**
Run Modeltest ng to select best substitution model for the data

RAxML-ng

**raxml-ng --msa AllSpeciesCds.fa --model GTR+G4**
Run RAxML ng to infer phylogenetic tree

python

**for z in {19458..19483};do nohup python makeAncestralSeq2.py NC_0${z}.2_merged.bcf Sample_Depth.txt NC_0${z}.2 GoatAncestralSequence_NC_0${z}.fa & done**
Using python script on goat bcf files generate ancestral sequences

cat

**cat GoatAncestralSequence_NC_0194{58..83}.fa > WholeGenomeGoatAncestral.fa**
Concatenate fasta files (split by chromosomes)

samtools

**samtools faidx WholeGenomeGoatAncestral.fa**
index the ancestral fasta file

angsd

**nohup ~/software/populationGenomics/angsdtool/angsd/angsd -bam SnowSheep.txt -doSaf 1 -anc ./bcftools_Goat/WholeGenomeGoatAncestral.fa -GL 1 -P 26 -out outSnowSheep &**
unfolded site frequency spectrum using angsd

**~/software/populationGenomics/angsdtool/angsd/misc/realSFS outSnowSheep.saf.idx -P 26 > SnowSheep.sfs**
Global estimate of SFS

**~/software/populationGenomics/angsdtool/angsd/misc/realSFS saf2theta outSnowSheep.saf.idx -sfs SnowSheep.sfs -outname SnowSheepTheta**
theta for each site

**for z in {19458..19483};do nohup ~/software/populationGenomics/angsdtool/angsd/misc/thetaStat do_stat ArgaliTheta.thetas.idx -win 50000 -step 10000 -r NC_0${z}.2 -outnames NC_0${z}_50kb_10kb_theta & done**
Estimate FWH and other statistics

busco

**python ./scripts/run_BUSCO.py -c 8 -o $1 -i $2 -l /home/maulik/data/Shared/Database/Protein/BUSCO/mammalia_odb9 -m geno -sp human --long -z --augustus_parameters='--progress=true'**
busco to train augustus parameters using single copy orthologs

perl

**perl braker.pl --genome=SnowSheepGuppy351Racon2.Agouti.fasta --bam=SnowSheepGuppy351Racon2.HisatFrAligned.sorted.bam --species=BUSCO_SnowSheepTraining_2_703440901 --skipAllTraining --softmasking --cores 40**
BRAKER to predict gene structure using RNA-seq data as hints and above-trained parameters using busco

python

**for f in *fa ; do python ~/orthofinder_tutorial/OrthoFinder/tools/primary_transcript.py $f ; done**
First step is to extract the longest isoform per gene for ensembl annotation, for this purpose, script supplied with OrthoFinder will be used (reference: step 7 here: https://davidemms.github.io/orthofinder_tutorials/running-an-example-

orthofinder-analysis.html). The customized python script ('ExtractLongestIsoform.py") was also used to extract gene id and transcript id associated with these proteins of longest isoform.

python

**python ExtractLongNcbiProt.py GCF_000298735.2_Oar_v4.0_genomic.gff GCF_000298735.2_Oar_v4.0_protein.faa OvisAri.LongestIsoProt.fa**

Extract longest isoform (protein sequences) downloaded from NCBI (used this for Ovis Aries sequences)

bioawk

**bioawk -c fastx '{split($name,a,".");if(a[1] in seqDict){print a[1],$name;if(length(se {seqDict[a[1]]=$seq;seqName[a[1]]=$name};next}else;seqDict[a[1]]=$seq;seqNa in seqDict){print ">"seqName[head]"\n"seqDict[head]}}' augustus.hints.aa > Snow**

Extract longest isoform (protein sequences) downloaded from NCBI (used this for Ovis Aries sequences)

orthofinder

**./orthofinder -t 40 -a 20 -f MultiSpeciesLongestIsoform/**

Run the orthofinder tool to identify the orthologs across the 10 species (Supplementary Table S2) of the paper

linux

python

**python CdsAlignmentParallel.py -orthoF OrthoInfo1.txt -listF AllSpeciesF.txt -coreN 40 -outD 1_to_1_ortho_aligned**

linux

get4foldSites

**get4foldSites -infile in.txt -outfile res.fas -iupac 0/1 -verbose 0/1**

four-fold degenerate sites were extracted using the program: https://github.com/brunonevado/get4foldSites

raxml

**/home/maulik/software/RAxML_NG/raxml-ng --msa AllFilesFourFoldMotifs.parallel.phy --model GTR+G --prefix AllFilesFourFoldMotifs.parallel --threads 40 --seed 2 --outgroup MonoDel**

linux

awk, codeml, and python

**/home/maulik/software/paml4.9i/bin/codeml NullModelSnowSheep.ctl**
**/home/maulik/software/paml4.9i/bin/codeml AlternativeModelSnowSheep.ctl**
**awk '$0~/lnL/{print $5}' *AlternativeModel.mlc >> ENSBTAG00000013029.5.mlc**
**awk '$0~/lnL/{print $5}' *NullModel.mlc >> ENSBTAG00000013029.5.mlc**
**python3 CalcChiSquare.py ENSBTAG00000013029.5.mlc ENSBTAG00000013029.5**

linux

python and dh

**while read chrm start end theta taji fwh;do python VcfToFastaDh.py ${chrm}.2_AllFiltered_WithGoat.vcf.gz ${chrm}.annot.phased.vcf.gz GCF_000298735.2_Oar_v4.0_genomic.reheader.fna WholeGenomeGoatAncestral.fa SnowSheepIds.txt ${chrm}.2 $start $end ${chrm}_${start}.fa;java -cp**

**.:/home/maulik/software/populationGenomics/dh/dh.jar dh.ReadFasta ${chrm}_${start}.fa 13 > ${chrm}_${start}_dh.out;done<SnowSheepTop245_reheader.Seg.txt**

python script will make consensus of selected snow sheep samples and outgroup using following information: 1). phased file 2). Goat ancestral sequences 3). Sheep reference sequences 4). vcf file containing information about missing sites Subsequently, it will run dh program to check for P-value using simulated data for FWH.

python and bash

**while read chrom size;do mkdir ${chrom}; sed "s/NC_019483/${chrom}/g" config.json > ${chrom}/config.json;cd ./${chrom};sed -i "s/44047080/${size}/g" config.json; cp ../CalculateRNDminSnakemake.py .;cp ../RunRndMin.sh .;sed -i "s/NC_019483.2/${chrom}/g" RunRndMin.sh;cd ..;done<SheepGenomeIndex1.txt**

Estimate RNDmin

find

**find . -type f -name "*RndMin.txt" -exec cp {} ../minRndResults/Argali/ \;**

Collect the results and put outfiles in separate directory

awk

**awk '$3>0 && $3<=1{split($1,a,"/");split(a[9],b,"_");match(b[3],/([0-9]+)/,c);print b[1]"_"b[2],c[1]-50000,c[1],$2,$3}' NC_0{19458..19483}.2_RndMin.txt|sort -k5,5 > Argali_RndMinResult_Sorted.txt**

concatenate the results and sort based on minRND values

python

**while read chrom start end minD minRnd;do python CalcWeirFstatDrawTree.py ${chrom}_OnlyBiallelic.bcf SampleIds2.tab ${chrom} $start $end ${chrom}_${start}_Fst.tree;done < Argali_RndMinResult_Sorted_Top1Perc.txt**

make fst-based phylogenetic tree

bioawk

**while read chrm start end;do for z in $(cat SpeciesName.txt);do bioawk -c fastx -v ">"spe"\n"substr($seq,sp,50000)}' ${chrm}_${z}.consensus.fa>>${chrm}_${start}_${end}".fa";done;done<Argali_**

prepare fasta file for dfoil

**while read chrm start end;do python ~/software/populationGenomics/dfoil/fasta2df ${chrm}_${start}_${end}.fa --out ${chrm}_${start}_${end}.dfoil.in --names OvisNiv,OvisDal,OvisVgn,OvisAmn,CaprHcs;done<Argali_SnowSheep_Sisters_0.12_R**

fasta2dfoil

**while read chrm start end;do python ~/software/populationGenomics/dfoil/dfoil.py infile ${chrm}_${start}_${end}.dfoil.in --mode dfoil --out ${chrm}_${start}_${end}.dfoil.out;done<Argali_SnowSheep_Sisters_0.12_RndMin.**

dfoil calculation