Oct 25, 2022

GATK Nuclear variant discovery and consensus assembly

Graham Etherington[1]

[1]The Earlham Institute

dx.doi.org/10.17504/protocols.io.bqzgmx3w

Graham Etherington
The Earlham Institute

ABSTRACT

The European polecat (*Mustela putorius*) is a mammalian predator which breeds across much of Europe east to central Asia. In Great Britain, following years of persecution the European polecat has recently undergone a population increase due to legal protection and its range now overlaps that of feral domestic ferrets (*Mustela putorius furo*). During this range expansion, European polecats hybridised with feral domestic ferrets producing viable offspring. Here we carry out population-level whole genome sequencing on domestic ferrets, British European polecats, and European polecats from the European mainland and find high degrees of genome introgression in British polecats outside their previous stronghold, even in those individuals phenotyped as 'pure' polecats.

DOI

dx.doi.org/10.17504/protocols.io.bqzgmx3w

EXTERNAL LINK

https://doi.org/10.1093/jhered/esac038

PROTOCOL CITATION

Graham Etherington 2022. GATK Nuclear variant discovery and consensus assembly. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

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

Etherington GJ, Ciezarek A, Shaw R, Michaux J, Croose E, Haerty W, Palma FD, Extensive genome introgression between domestic ferret and European polecat during population recovery in Great Britain. Journal of Heredity 113(5). doi: 10.1093/jhered/esac038

CREATED

Dec 23, 2020

LAST MODIFIED

Oct 25, 2022

PROTOCOL INTEGER ID

45832

BWA mapping

1 Here we map our reads to the domestic ferret genome, run the GATK pipeline to identify variants.

1.1 Map the reads to the reference and sort the output

```
#R1 and R2 refer to the forward and reverse reads
R1=$1
R2=$2
dir="$(dirname $R1)"
fpath="$(basename $R1)"
samplename="$(cut -d'_' -f1 <<< $fpath)"
fname=../bams/$dir/$samplename

source bwa-0.7.7
source samtools-1.7

bwa mem -t 8 -M  ../reference/MusPutFur1.0_bionano.fasta
$R1 $R2 | samtools sort -@ 8 -o $fname\_sorted.bam -
```

2 GATK Pipeline

GATK pipeline

2.1 Mark Duplicates, add read groups, call variants, calculate GATK parameters,

Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

filter variants, create base quality recalibration table, apply it, and re-run variant calling.
The GenomeHelper program can be found here:
https://github.com/ethering/GenomeHelper
The mean value of SNP quality, read depth, and mapping quality (QUAL, DP, and MQ values respectively from the VCF file) were calculated using GenomeHelper,

```
#R1 refer to the forward and reverse reads used in the
mapping. It's used as the sample name.
R1=$1

dir="$(dirname $R1)"
fpath="$(basename $R1)"
samplename="$(cut -d'_' -f1 <<< $fpath)"
fname=../bams/$dir/$samplename
metrics=../metrics/$samplename\_metrics.txt
BAM=$fname\_rg.bam
VCF=../vcfs/$dir/$samplename\.vcf
PARAMS=../vcfs/$dir/$samplename\.vcf.params
FILTERED=../vcfs/$dir/$samplename\_filtered.vcf
RECTABLE=../bams/$dir/$samplename\_recal.table
RECBAM=../bams/$dir/$samplename\_recal.bam
GVCF=../vcfs/$dir/$samplename\.g.vcf.gz


source jre-8u92
source samtools-1.7
source gatk-4.1.3.0_spark

srun java -jar -XX:ParallelGCThreads=2 -Xmx100G
/ei/software/testing/picardtools/2.21.4/x86_64/bin/picar
d.jar MarkDuplicates I=$fname\_sorted.bam
O=$fname\_mkdups.bam M=$metrics
TMP_DIR=/ei/scratch/ethering/tmp
srun java -jar -XX:ParallelGCThreads=2 -Xmx100G
/ei/software/testing/picardtools/2.21.4/x86_64/bin/picar
d.jar AddOrReplaceReadGroups I=$fname\_mkdups.bam O=$BAM
RGID=$samplename RGLB=lib1 RGPL=illumina
RGSM=$samplename RGPU=$samplename
srun samtools index $BAM
srun gatk --java-options "-Xmx100G -
XX:ParallelGCThreads=2" HaplotypeCaller  -R
../reference/MusPutFur1.0_bionano.fasta -I $BAM -O $VCF
```


Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

```
#FOR PCR-FREE
#gatk --java-options "-Xmx50G -XX:ParallelGCThreads=4"
HaplotypeCaller --pcr-indel-model NONE  -R
../reference/MusPutFur1.0_bionano.fasta -I $BAM -O $VCF

#Calculate the mean value of SNP quality, read depth,
and mapping quality (QUAL, DP, and MQ values
respectively from the VCF file)
srun java -Xmx100G -XX:ParallelGCThreads=2 -jar
~/NetBeansProjects/uk.ac.tsl.etherington.genomehelper/di
st/GenomeHelper.jar CalculateGATKparams -in $VCF

LOWQUAL=$(sed -ne 's/^lowQUAL://p' $PARAMS)
LOWDP=$(sed -ne 's/^lowDP://p' $PARAMS)
LOWMQ=$(sed -ne 's/^lowMQ://p' $PARAMS)

srun gatk --java-options "-Xmx100G -
XX:ParallelGCThreads=2" VariantFiltration  -R
../reference/MusPutFur1.0_bionano.fasta -V $VCF -O
$FILTERED -filter "MQ < '${LOWMQ}'" --filter-name
"LowMQ" -filter "QUAL < '${LOWQUAL}'"  --filter-name
'LowQual' -filter "DP < '${LOWDP}'"  --filter-name
'LowCov'

srun gatk --java-options "-Xmx100G -
XX:ParallelGCThreads=2" BaseRecalibrator -R
../reference/MusPutFur1.0_bionano.fasta -I $BAM --known-
sites $FILTERED -O $RECTABLE
srun gatk --java-options "-Xmx100G -
XX:ParallelGCThreads=2" ApplyBQSR -R
../reference/MusPutFur1.0_bionano.fasta -I $BAM --bqsr-
recal-file $RECTABLE -O $RECBAM
srun gatk --java-options "-Xmx100G -
XX:ParallelGCThreads=2" HaplotypeCaller  -R
../reference/MusPutFur1.0_bionano.fasta -I $RECBAM -O
$GVCF -ERC GVCF
##FOR PCR-FREE
#gatk --java-options "-Xmx50G -XX:ParallelGCThreads=4"
HaplotypeCaller  --pcr-indel-model NONE -R
../reference/MusPutFur1.0_bionano.fasta -I $RECBAM -O
$GVCF -ERC GVCF
```

2.2 Identify SNPs

Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

Next, we want to create a GATK GenomicsDB and then joint genotype all of the samples to get a multi-sample VCF file of SNPs

Firstly, In our 'vcfs' directory, we need a tab-delimited sample map, linking the sample-specifice GVCF file to a sample name, and an intervals file.

cohort.sample_map

```
bff_SRR1508214  ./m_nigripes/SRR1508214.g.vcf.gz
bff_SRR1508215  ./m_nigripes/SRR1508215.g.vcf.gz
bff_SRR1508749  ./m_nigripes/SRR1508749.g.vcf.gz
bff_SRR1508750  ./m_nigripes/SRR1508750.g.vcf.gz
domestic_LIB8733
./m_putorius/furo/LIB8733.g.vcf.gz
domestic_LIB8734
./m_putorius/furo/LIB8734.g.vcf.gz
domestic_LIB8735
./m_putorius/furo/LIB8735.g.vcf.gz
domestic_LIB8736
./m_putorius/furo/LIB8736.g.vcf.gz
domestic_LIB8737
./m_putorius/furo/LIB8737.g.vcf.gz
domestic_LIB8738
./m_putorius/furo/LIB8738.g.vcf.gz
domestic_LIB8739
./m_putorius/furo/LIB8739.g.vcf.gz
domestic_LIB8740
./m_putorius/furo/LIB8740.g.vcf.gz
euro_LIB18989   ./m_putorius/putorius/LIB18989.g.vcf.gz
euro_LIB18990   ./m_putorius/putorius/LIB18990.g.vcf.gz
euro_LIB18991   ./m_putorius/putorius/LIB18991.g.vcf.gz
euro_LIB18992   ./m_putorius/putorius/LIB18992.g.vcf.gz
euro_LIB18993   ./m_putorius/putorius/LIB18993.g.vcf.gz
euro_LIB18994   ./m_putorius/putorius/LIB18994.g.vcf.gz
euro_LIB18995   ./m_putorius/putorius/LIB18995.g.vcf.gz
euro_LIB21977   ./m_putorius/putorius/LIB21977.g.vcf.gz
euro_S01        ./m_putorius/putorius/S01.g.vcf.gz
euro_S02        ./m_putorius/putorius/S02.g.vcf.gz
euro_S04        ./m_putorius/putorius/S04.g.vcf.gz
euro_S05        ./m_putorius/putorius/S05.g.vcf.gz
euro_S06        ./m_putorius/putorius/S06.g.vcf.gz
euro_S19        ./m_putorius/putorius/S19.g.vcf.gz
euro_S20        ./m_putorius/putorius/S20.g.vcf.gz
hybrid_LIB21971 ./m_putorius/hybrids/LIB21971.g.vcf.gz
hybrid_LIB21972 ./m_putorius/hybrids/LIB21972.g.vcf.gz
hybrid_LIB21973 ./m_putorius/hybrids/LIB21973.g.vcf.gz
steppe_LIB18996 ./m_eversmanii/LIB18996.g.vcf.gz
```


Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

```
steppe_LIB22031 ./m_eversmanii/LIB22031.g.vcf.gz
uk_euro_LIB21974
./m_putorius/putorius/LIB21974.g.vcf.gz
uk_euro_LIB21975
./m_putorius/putorius/LIB21975.g.vcf.gz
uk_euro_LIB22032
./m_putorius/putorius/LIB22032.g.vcf.gz
uk_euro_LIB23764
./m_putorius/putorius/LIB23764.g.vcf.gz
uk_euro_S07     ./m_putorius/putorius/S07.g.vcf.gz
uk_euro_S08     ./m_putorius/putorius/S08.g.vcf.gz
uk_euro_S09     ./m_putorius/putorius/S09.g.vcf.gz
uk_euro_S10     ./m_putorius/putorius/S10.g.vcf.gz
uk_euro_S11     ./m_putorius/putorius/S11.g.vcf.gz
uk_euro_S12     ./m_putorius/putorius/S12.g.vcf.gz
uk_euro_S13     ./m_putorius/putorius/S13.g.vcf.gz
uk_euro_S14     ./m_putorius/putorius/S14.g.vcf.gz
uk_euro_S15     ./m_putorius/putorius/S15.g.vcf.gz
uk_euro_S16     ./m_putorius/putorius/S16.g.vcf.gz
uk_euro_S17     ./m_putorius/putorius/S17.g.vcf.gz
uk_euro_S18     ./m_putorius/putorius/S18.g.vcf.gz
weasel_LIB20027 ./m_nivalis/LIB20027.g.vcf.gz
```

We also need to to create an intervals file, which will just be a list of scaffolds in the reference sequence

```
grep ">" MusPutFur1.0_bionano.fasta | sed 's/>//g' >
MusPutFur1.0_bionano.intervals
```

2.3 Create the GenomicsDB, Genotype each sample and select only SNPs

```
REF=../reference/MusPutFur1.0_bionano.fasta

source jre-8u92
source gatk-4.1.3.0_spark

srun gatk --java-options "-Xmx550g -
XX:ParallelGCThreads=2"  GenomicsDBImport -R $REF  --
intervals MusPutFur1.0_bionano.intervals --genomicsdb-
workspace-path all_samples_GenomicsDB --sample-name-map
cohort.sample_map --tmp-dir=/ei/scratch/ethering/tmp/ -
max-num-intervals-to-import-in-parallel 100 --merge-
input-intervals --overwrite-existing-genomicsdb-
```


Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

```
workspace
gatk --java-options "-Xmx550g -XX:ParallelGCThreads=2"
GenotypeGVCFs -R $REF  -V gendb://all_samples_GenomicsDB
--new-qual -O all_samples_genotyped.vcf
gatk --java-options "-Xmx550g -XX:ParallelGCThreads=2"
SelectVariants  -R $REF  -V all_samples_genotyped.vcf -O
all_samples_genotyped_snps.vcf -select-type SNP
```

2.4 Create a consensus genome sequence for each sample

gatk_farm.sh

```
#provide the sample name (as provided in
cohort.sample_map above)
SN=$1
#Provide the path to the directory
DIR_PATH=$2
FASTA_PATH="$(dirname $DIR_PATH)"

#The multi-sample VCF file with SNPS
VCF=../vcfs/all_samples_genotyped_snps.vcf
#Create a temporary vcf file
SN_VCF=../temp/$SN\_temp.vcf
#The reference sequence
REF=../reference/MusPutFur1.0_bionano.fasta
#The path and filename for the output consensus sequence
FASTA=$FASTA_PATH/$SN.fasta

source jre-8u92
source gatk-4.1.3.0_spark

#Exclude non-variants from the sample to avoid calling
REF calls as ALTS
gatk --java-options "-Xmx20g -XX:ParallelGCThreads=4"
SelectVariants -sn $SN -R $REF -V $VCF -O $SN_VCF --
exclude-non-variants

#Create the consensus sequence
gatk --java-options "-Xmx20g -XX:ParallelGCThreads=4"
FastaAlternateReferenceMaker -R $REF -V $SN_VCF -O
$FASTA

#See 'Note 1' below
sed -i -e 's/>[[:digit:]]\s/>/g' -e 's/:.*//g' $FASTA
```


Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w

```
#delete the temporary VCF file
rm $SN_VCF*
```

Note 1.
FastaAlternateReferenceMaker puts an incremented number after the ">" and then the co-ordinates of the sequence after the scaffold name, so:

>Super-Scaffold_5
becomes
>5 Super-Scaffold_5:1-33914829

I use the following sed at the end of the script to go back to the original fasta header:

```
sed -i -e 's/>[[:digit:]]\s/>/g' -e 's/:.*//g' $FASTA
```

As a helper script, if using SLURM, you can use the following script to submit each job.

```
while read -ra array
do
        sbatch -J ${array[0]} gatk_farm.sh ${array[0]}
${array[1]}
done < cohort.sample_map
```


Citation: Graham Etherington GATK Nuclear variant discovery and consensus assembly
https://dx.doi.org/10.17504/protocols.io.bqzgmx3w