NPRC Project - Data Preparation

Shaurita Hutchins

2020-01-14

## Data Preparation

In order to use our data in downstream applications such as Truffle/fastStructure and the pedigree reconstruction, vcf files from each primate research center were combined into one file (for faster data annotation - bcftools merge was used). That combined file is named all\_nprcs.vcf.gz. The next step was to remove unneded annotation from the files. This step was completed using bcftools.

DATA=/ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/mGAP  
VCF=${DATA}/all\_nprcs.vcf.gz  
OUTFILE=${DATA}/all\_nprcs\_rminfo.vcf.gz  
  
IDS=INFO/ANN,INFO/CADD\_PH,INFO/CADD\_RS,INFO/CCDS,INFO/ENC,INFO/ENCDNA\_CT,INFO/ENCDNA\_SC,  
INFO/ENCSEG\_CT,INFO/ENCSEG\_NM,INFO/ENCTFBS\_CL,INFO/ENCTFBS\_SC,INFO/ENCTFBS\_TF,INFO/ENN,  
INFO/ERBCTA\_CT,INFO/ERBCTA\_NM,INFO/ERBCTA\_SC,INFO/ERBSEG\_CT,INFO/ERBSEG\_NM,INFO/ERBSEG\_SC,  
INFO/ERBSUM\_NM,INFO/ERBSUM\_SC,INFO/ERBTFBS\_PB,INFO/ERBTFBS\_TF,INFO/FC,INFO/FE,INFO/FS\_EN,  
INFO/FS\_NS,INFO/FS\_SC,INFO/FS\_SN,INFO/FS\_TG,INFO/FS\_US,INFO/FS\_WS,INFO/GRASP\_AN,INFO/GRASP\_P,  
INFO/GRASP\_PH,INFO/GRASP\_PL,INFO/GRASP\_PMID,INFO/GRASP\_RS,INFO/LF,INFO/LOF,INFO/NC,INFO/NE,  
INFO/NF,INFO/NG,INFO/NH,INFO/NJ,INFO/NK,INFO/NL,INFO/NM,INFO/NMD,INFO/OMIMC,INFO/OMIMD,  
INFO/OMIMM,INFO/OMIMMUS,INFO/OMIMN,INFO/OMIMS,INFO/OMIMT,INFO/OREGANNO\_PMID,INFO/OREGANNO\_TYPE,I  
NFO/PC\_PL,INFO/PC\_PR,INFO/PC\_VB,INFO/PP\_PL,INFO/PP\_PR,INFO/PP\_VB,INFO/RDB\_MF,INFO/RDB\_WS,  
INFO/RFG,INFO/RSID,INFO/SCSNV\_ADA,INFO/SCSNV\_RS,INFO/SD,INFO/SF,INFO/SM,INFO/SP\_SC,  
INFO/SX,INFO/TMAF,INFO/CLN\_ALLELE,INFO/CLN\_ALLELEID,INFO/CLN\_DBVARID,INFO/CLN\_DISDB,  
INFO/CLN\_DISDBINCL,INFO/CLN\_DN,INFO/CLN\_DNINCL,INFO/CLN\_GENEINFO,INFO/CLN\_HGVS,INFO/CLN\_MC,  
INFO/CLN\_ORIGIN,INFO/CLN\_REVSTAT,INFO/CLN\_RS,INFO/CLN\_SIG,INFO/CLN\_SIGINCL,INFO/CLN\_SSR,  
INFO/CLN\_VC,INFO/CLN\_VCSO,INFO/CLN\_VI  
  
module use /ddn/home3/vallender/software/module\_files  
module load nprc\_project  
  
bcftools annotate $VCF -x $IDS -O z -o $OUTFILE --threads 40  
  
bcftools index $OUTFILE --threads 40

After unnecessary information (INFO/ID) was removed, the file shrunk by 6 GB which has aided in manageability. The resulting file was all\_nprcs\_rminfo.vcf.gz. It is located in /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/mGAP.

### Generating Input for Pedigree Reconstruction

Based on Tulane’s paper on pedigree reconstruction, Chinese-origin animals and only WGS data should be included in later analyses. Also, according to [Sasha Kusev](https://github.com/gusevlab/germline/issues/5), the creator of germline, our data should be biallelic. That information isn’t present in the current literature, but it’s not a shocking revelation.

160 of our samples were exome only samples across the data sets. That means that 578 samples have at least their whole genome sequenced.

# Set directory variables  
DATA\_DIR=/ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data  
MGAP\_DIR=${DATA\_DIR}/mGAP  
INFILE1=${MGAP\_DIR}/all\_nprcs\_rminfo\_norm.vcf.gz  
FASTA=${DATA\_DIR}/ref\_genome/1\_Mmul\_8.0.1.fasta  
OUTDIR=${DATA\_DIR}/pedigree\_reconstruction\_input  
OUTFILE1=${OUTDIR}/all\_nprcs\_rminfo\_auto\_wgs.vcf.gz  
OUTFILE2=${OUTDIR}/all\_nprcs\_rminfo\_auto\_wgs\_norm.vcf.gz  
OUTFILE3=${OUTDIR}/all\_nprcs\_rminfo\_auto\_wgs\_norm\_bia.vcf.gz  
CHROMOSOMES=chr01,chr02,chr03,chr04,chr05,chr06,chr07,chr08,chr09,chr10,chr11,  
chr12,chr13,chr14,chr15,chr16,chr17,chr18,chr19,chr20,chrUn,MT  
SAMPLES\_FILE=${DATA\_DIR}/metadata/wgs\_india\_ids.txt  
  
# Load module file  
module use /ddn/home3/vallender/software/module\_files  
module load nprc\_project  
  
mkdir $OUTDIR  
  
# Subset the vcf. Keep autosomal chromosomes. Keep only WG, INDIAN origin samples.  
bcftools view $INFILE1 -r $CHROMOSOMES -S $SAMPLES\_FILE -O z -o $OUTFILE1 --threads 75  
  
bcftools index $OUTFILE1 --threads 75  
  
# Normalize the vcf file by checking snps with the ref genome  
# and removing both snp & indel duplicates  
bcftools norm $OUTFILE1 -c s -d both -f $FASTA -O z -o $OUTFILE2 --threads 75  
  
bcftools index $OUTFILE2 --threads 75  
rm -rf $OUTFILE1  
rm -rf $OUTFILE1.csi  
  
# -m2 and -M2 indicate making the file biallelic  
# -v selects both snps and indels  
bcftools view $OUTFILE2 -O z -o $OUTFILE3 -m2 -M2 -v snps,indels --threads 75  
  
bcftools index $OUTFILE3 --threads 75  
rm -rf $OUTFILE2  
rm -rf $OUTFILE2.csi  
  
chmod 0770 $OUTDIR  
chmod 0770 $OUTFILE3

The script for this can be found at /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/src/pedigree\_prep on the MCSR. The resulting file is all\_nprcs\_rminfo\_auto\_wgs\_norm\_bia.vcf.gz and is located in /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/pedigree\_reconstruction\_input.

### Creating Input for Truffle & fastStructure

Both truffle and fastStructure require biallelic data and WGS only data. For Truffle, if exome data is being used, parameters need to be tweaked. Initially, when the data included exome data, I attempted to tweak the parameters to no avail.

#### Truffle Input Data

The output file for Truffle is very similar to the input file for the pedigree reconstruction pipeline except it has not been normalized. The normalization step does not seem to be necessary for Truffle. There was no need to filter or subset data by the minor allele frequency given that Truffle can filter data based on it.

DATA\_DIR=/ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data  
  
INFILE=${DATA\_DIR}/mGAP/all\_nprcs\_rminfo.vcf.gz  
OUTFILE=${DATA\_DIR}/truffle\_input/all\_nprcs\_rminfo\_autosomal\_wgs\_biallelic\_mt.vcf.gz  
CHROMOSOMES=chr01,chr02,chr03,chr04,chr05,chr06,chr07,chr08,chr09,chr10,chr11,  
chr12,chr13,chr14,chr15,chr16,chr17,chr18,chr19,chr20,chrUn,MT  
SAMPLES\_FILE=/ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/metadata/wgs\_indian\_ids.txt  
  
module use /ddn/home3/vallender/software/module\_files  
module load nprc\_project  
  
  
bcftools view $INFILE -r $CHROMOSOMES -S $SAMPLES\_FILE -O z -o $OUTFILE -m2 -M2 -v snps,indels --threads 80  
  
bcftools index $OUTFILE --threads 80

The script for this can be found at /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/src/truffle\_prep on the MCSR.

There are two data files available for truffle: all\_nprcs\_rminfo\_autosomal\_wgs\_biallelic\_mt.vcf.gz and all\_nprcs\_rminfo\_autosomal\_wgs\_biallelic\_un\_mt.vcf.gz. They are both located in /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/truffle\_input. The two data files were generated in order to test whether chrUn (unknown snps) has an impact on the data.

#### fastStructure Input Data

fastStructure poses some complications in that the input for the program needs to be in a very specific format that is poorly documented. Also, given the computational intensiveness of fastStructure, the size of the resulting data file should be limited if possible.

To prepare the vcf for fastStructure, only WGS, Indian, autosomal chromosomes, biallelic snps and indels, and a minor allele frequency of >0.05 were used.

DATA\_DIR=/ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data  
  
INFILE=${DATA\_DIR}/mGAP/all\_nprcs\_rminfo.vcf.gz  
OUTFILE=${DATA\_DIR}/structure\_input/all\_auto\_wgs\_biallelic\_maf05.vcf.gz  
CHROMOSOMES=chr01,chr02,chr03,chr04,chr05,chr06,chr07,chr08,chr09,chr10,chr11,  
chr12,chr13,chr14,chr15,chr16,chr17,chr18,chr19,chr20,chrUn,MT  
SAMPLES\_FILE=/ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/metadata/wgs\_india\_ids.txt  
  
module use /ddn/home3/vallender/software/module\_files  
module load nprc\_project  
  
# Biallelic, autosomes only, only wgs samples, maf > .05  
bcftools view $INFILE -r $CHROMOSOMES -S $SAMPLES\_FILE -m2 -M2 -v snps,indels -i 'MAF>0.05' -O z -o $OUTFILE --threads 80  
  
bcftools index $OUTFILE --threads 80

In addition to preparing the vcf file, a file formatted for fastStructure must be created. Initially, this was being done using a perl script that was written by Dr. Vallender, but I have adapted (and am adapting) a python script (below) that works and will generate the input for fastStructure.

#!/usr/bin/env python3  
from \_\_future\_\_ import print\_function  
import argparse  
import sys  
import os  
import vcf # pip install pyVCF  
  
  
def errprint(\*args, \*\*kwargs):  
 """print to stderr not stdout"""  
 print(\*args, file=sys.stderr, \*\*kwargs)  
  
  
# parser  
parser = argparse.ArgumentParser() # add the parser  
parser.add\_argument("--input", help="input VCF file") # add the parser  
parser.add\_argument(  
 "--output", help="output STRUCTURE DATA file") # add the parser  
  
args = parser.parse\_args()  
  
  
def write\_structure\_file(outfile, snps, genotype\_dict):  
 print("Writing %s..." % outfile)  
 with open(outfile, "w") as output:  
 header = "#\t#\t#\t#\t#\t\Sample\_ID"  
 output.write("\t".join(snps) + "\n")  
 for ind in genotype\_dict.keys():  
 output.write("\t".join([ind]+genotype\_dict[ind]) + "\n")  
 print("%s has been written." % outfile)  
  
  
def import\_vcf(infile):  
  
 # open the vcf parser  
 input\_vcf = vcf.Reader(filename=infile, compressed=True,  
 prepend\_chr="False", strict\_whitespace=False)  
 print('%s has been imported.' % infile)  
  
 return input\_vcf  
  
  
def parse(vcf\_obj):  
 dict\_alleles = {"0/0": "11", "0/1": "12",  
 "1/0": "12", "1/1": "22", "./.": "-9"}  
 list\_snps = []  
 nsites = 0  
 gen\_dict = {ind: [] for ind in vcf\_obj.samples}  
  
 # store all the genotypes and loci names  
 print("Creating genotype dictionary...")  
 for site in vcf\_obj:  
 list\_snps.append(site.CHROM + "\_" + str(site.POS))  
 for i in range(len(gen\_dict.keys())):  
 gen\_dict[site.samples[i].sample].append(  
 dict\_alleles[site.samples[i]["GT"]])  
  
 return list\_snps, gen\_dict  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 vcf = import\_vcf(infile=args.input)  
  
 snps, genes = parse(vcf\_obj=vcf)  
  
 write\_structure\_file(outfile=args.output,  
 snps=snps, genotype\_dict=genes)

Both scripts are located in /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/src/fast\_structure. The output is located in /ddn/home3/vallender/projects/NPRC\_Pedigree\_Project/data/structure\_input.

## Current Status

At the moment, input for the pedigree reconstruction pipleline is being generated. The vcf for fastStructure input has been generated. The truffle input vcfs have also been generated.

## Questions

* In the Tulane paper, it is mentioned that data is further filtered for ERSA, but there are INFO fields unavailable to us (even prior to removing some). Was a different version of the vcf file used or was the data run with a different/newer version of the GATK pipeline?
* Should the ids removed from the subsetted plink PCA analysis also be removed for our input data files for the pedigree reconstruction, truffle, and fastStructure?