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_Pro`

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, 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

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 $OU

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
```

```python
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?