Assignment 2 Fst calculation of between maize and mexicana

Niranjan Pokhrel

2025-03-24

Information about article and data

Since the data collected in Assignment_1 for the study was around 1 TB, it was very large file to do analysis. So, for this study another article titled "Selective dorting of ancestral introgression in maize and teosinte along an elevational cline." by Calfee et.al., 2021 was selected. This study used 14 maize and 14 mexicana accessions. The data used in this study is stored in NCBI Short Read Archive with accession number "PRJNA657016".

For this assignment purposed, only 10 accessions of each maize and mexicana were used. Whole genome sequence dataset were downloaded and extracted. The codes to download and unzip the data is shared in Assignment_1.

The detailed steps to calculate Fst value for two population is explained below:

Step 1: The "bwa" and "samtools" was used to align raw sequence file to the referene genome and process, process manage and filter alignment files respectively. The detailed code is in the chunk below:

```
#!/bin/bash
#SBATCH -D /work/agro932/niranjan27/
#SBATCH -o /work/agro932/niranjan27/logs/align-wg-stdout-%A_%a.txt
#SBATCH -e /work/agro932/niranjan27/logs/align-wg-stderr-%A_%a.txt
#SBATCH -J align_wholegenome
#SBATCH -t 96:00:00
#SBATCH -p schnablelab
#SBATCH --array=1-20
#SBATCH -c 16
#SBATCH --mem=64G
#SBATCH --mail-user=npokhrel3@huskers.unl.edu
#SBATCH --mail-type=BEGIN
#SBATCH --mail-type=END
#SBATCH --mail-type=FAIL

# Exit if any command fails
set -euo pipefail

# Load required modules
module load bwa samtools
```

```
# Define paths
REF_WG="/work/agro932/niranjan27/Whole_reference_genome/Zea_mays.Zm-B73-REFERENCE-NAM-5.0.dna.toplevel.
MAIZE_DIR="/work/agro932/niranjan27/maize"
MEXICANA_DIR="/work/agro932/niranjan27/mexicana"
OUTPUT_DIR="/work/agro932/niranjan27/aligned_wholegenome"
SAMPLE_LIST="/work/agro932/niranjan27/sample_list.txt"

# Create output directory if it doesn't exist
mkdir -p "$OUTPUT_DIR"

# Get current sample information from sample list
SAMPLE_INFO=$(sed -n "${SLURM_ARRAY_TASK_ID}p" "$SAMPLE_LIST")
PREFIX=$(echo "$SAMPLE_INFO" | awk '{print $1}')
SAMPLE=$(echo "$SAMPLE_INFO" | awk '{print $2}')

# Set the read directory based on species prefix
if [[ "$PREFIX" == "maize" ]]; then
    READ_DIR="$MAIZE_DIR"
else
    READ_DIR="$MEXICANA_DIR"
fi

# Define FASTQ input file paths
R1="${READ_DIR}/${SAMPLE}.lite.1_1.fastq"
R2="${READ_DIR}/${SAMPLE}.lite.1_2.fastq"

# Check if FASTQ files exist
if [[ ! -f "$R1" || ! -f "$R2" ]]; then
    echo "Missing FASTQ files for $SAMPLE"
    exit 1
fi

# Define BAM output filenames
OUT_BAM="${OUTPUT_DIR}/${PREFIX}_${SAMPLE}_wg.bam"
SORTED_BAM="${OUTPUT_DIR}/${PREFIX}_${SAMPLE}_wg_sorted.bam"

# Step 1: Align reads to reference genome
bwa mem -t 16 "$REF_WG" "$R1" "$R2" | samtools view -bSh - > "$OUT_BAM"

# Step 2: Sort BAM file
samtools sort -@ 8 "$OUT_BAM" -o "$SORTED_BAM"

# Step 3: Index sorted BAM
samtools index "$SORTED_BAM"

# Step 4: Clean up intermediate BAM
rm "$OUT_BAM"
```

Step 2: The step 1 created the BAM files which is aligned sequences files. The aligned sequence files are then used for snp calling using "bcftools". The detailed steps is in chunk below:

```
#!/bin/bash
#SBATCH -D /work/agro932/niranjan27/
#SBATCH -o /work/agro932/niranjan27/snp_calls_whole_genome/logs/snp-stdout-%A_%a.txt
#SBATCH -e /work/agro932/niranjan27/snp_calls_whole_genome/logs/snp-stderr-%A_%a.txt
#SBATCH -J maize_mexicana_snp_wg
#SBATCH -t 72:00:00
#SBATCH --array=1-20
#SBATCH -c 16
#SBATCH --mem=64G
#SBATCH --mail-user=npokhrel3@huskers.unl.edu
#SBATCH --mail-type=BEGIN,END,FAIL

set -euo pipefail

# === Load required modules ===
module load samtools bcftools

# === Define input BAMs and get sample for this array task ===
BAMS=(/work/agro932/niranjan27/aligned_wholegenome/*_wg_sorted.bam)
BAM=${BAMS[$SLURM_ARRAY_TASK_ID-1]}
SAMPLE=$(basename "$BAM" _wg_sorted.bam)

# === Reference genome (whole genome) ===
REF=/work/agro932/niranjan27/Whole_reference_genome/Zea_mays.Zm-B73-REFERENCE-NAM-5.0.dna.toplevel.fa

# === Output directory ===
OUTDIR=/work/agro932/niranjan27/snp_calls_whole_genome
mkdir -p "${OUTDIR}/logs"

# === Index BAM if index is missing ===
if [[ ! -f "${BAM}.bai" ]]; then
    samtools index "$BAM"
fi

# === SNP calling with bcftools ===
bcftools mpileup -Ou -f "$REF" "$BAM" -a AD,DP | \
bcftools call -mv -Ob -o "${OUTDIR}/${SAMPLE}.bcf"

# === Index the BCF file ===
bcftools index "${OUTDIR}/${SAMPLE}.bcf"
```

Step 3: ALl snp called bcf files of 10 maize and 10 mexicanan were then merged and then converted to tabular format for analysis in R.

```
#Merge all bcf file

module load bcftools
```

```
bcftools merge -Ob -o merged.bcf *.bcf

#Index merged file
bcftools index merged.bcf

# Filter biallelic snps
bcftools view merged.bcf -m2 -M2 -v snps -Ob -o merged_biallelic_snps.bcf
bcftools index merged_biallelic_snps.bcf

#Convert to tabular format for R

bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%QUAL\t%DP[\t%GT]\n' merged_biallelic_snps.bcf > snp_calls
```

Step 3: Since the final text file of "snp__calls.txt" was large file R could not load and extract alleles and allele frequency, so the snp data was seperated to small sized chunks for easy analysis. The code below seperates snp data to small chunks and extract Fst value for all and create Manhatten plot for each chunk.

```
# === Load required package ===
library(data.table)

# === Read arguments from SLURM (chunk input/output filenames) ===
args <- commandArgs(trailingOnly = TRUE)
input_file <- args[1]
output_csv <- args[2]
output_plot <- args[3]

# === Read SNP chunk ===
geno <- fread(input_file, header = FALSE)

# === Assign column names ===
names(geno) <- c("chr", "pos", "ref", "alt", "quality", "depth", paste0("l", 1:20))

# === Extract alleles for each individual ===
geno <- as.data.frame(geno)
for(i in 7:26){
  allele1 <- gsub("/.*", "", geno[, i])
  allele2 <- gsub(".*/", "", geno[, i])
  nm <- names(geno)[i]
  geno[[paste0(nm, "_a1")]] <- allele1
  geno[[paste0(nm, "_a2")]] <- allele2
}

# === Replace missing values ===
geno[geno == "."] <- NA

# === Calculate allele frequencies ===
geno$p <- apply(geno[, 27:66], 1, function(x) sum(x == 0, na.rm=TRUE)) / 40
geno$p1 <- apply(geno[, 27:46], 1, function(x) sum(x == 0, na.rm=TRUE)) / 20
geno$p2 <- apply(geno[, 47:66], 1, function(x) sum(x == 0, na.rm=TRUE)) / 20

# === Compute Fst ===
geno$fst <- with(geno, ((p1 - p)^2 + (p2 - p)^2) / (2 * p * (1 - p)))
geno$fst[is.nan(geno$fst)] <- NA  # clean up invalid values

# === Save Fst values to CSV ===
write.csv(geno[, c("chr", "pos", "fst")], output_csv, row.names = FALSE)

# === Plot Fst ===
png(output_plot, width=1200, height=600)
plot(geno$pos, geno$fst, pch=20, col="blue", xlab="Position", ylab="Fst", main=paste("Fst -", basename(
dev.off()
```

Step 4: Combining the allele frequency data and plotting a single manhatten plot

```
# Combine all the csv file and plot the data

# Load package
library(data.table)

# === Define your folder ===
fst_dir <- "/work/agro932/niranjan27/fst_output/"

# === List all Fst CSV files ===
fst_files <- list.files(path = fst_dir, pattern = "snp_chunk_.*_fst\\.csv$", full.names = TRUE)

# === Combine all files into one data.table ===
fst_all <- rbindlist(lapply(fst_files, fread))

# === Optional: Sort by chromosome and position ===
fst_all <- fst_all[order(chr, pos)]

# === Save combined Fst CSV ===
fwrite(fst_all, file = file.path(fst_dir, "fst_all_combined.csv"))

# === Plot combined Fst ===
png(file.path(fst_dir, "fst_combined_plot.png"), width = 1600, height = 600)
plot(fst_all$pos, fst_all$fst, pch = 20, col = "darkblue",
     xlab = "Genomic Position", ylab = "Fst",
     main = "Genome-wide Fst Values")
dev.off()
```

The code below is to run above step 4 code in slurm

```
#!/bin/bash
#SBATCH -D /work/agro932/niranjan27/
#SBATCH -J fst_chunk
#SBATCH -o /work/agro932/niranjan27/fst_output/fst-stdout-%A_%a.txt
#SBATCH -e /work/agro932/niranjan27/fst_output/fst-stderr-%A_%a.txt
#SBATCH -t 24:00:00
#SBATCH -p schnablelab
#SBATCH -c 8
```

```
#SBATCH --mem=64G
#SBATCH --array=1-90
#SBATCH --mail-type=FAIL

# === Load R ===
module load R

# === Define chunk filenames ===
CHUNK_DIR="/work/agro932/niranjan27/snp_calls_whole_genome/chunks"
CHUNK_LIST=($(ls $CHUNK_DIR/snp_chunk_*.txt))
INPUT_FILE=${CHUNK_LIST[$SLURM_ARRAY_TASK_ID-1]}

# === Output filenames ===
OUT_DIR="/work/agro932/niranjan27/fst_output"
BASENAME=$(basename "$INPUT_FILE" .txt)
CSV_OUT="${OUT_DIR}/${BASENAME}_fst.csv"
PLOT_OUT="${OUT_DIR}/${BASENAME}_fst.png"

# === Run R script ===
Rscript /work/agro932/niranjan27/scripts/calculate_fst_chunk.R "$INPUT_FILE" "$CSV_OUT" "$PLOT_OUT"
```