* Bcftools was used to rename the samples from the current ‘sample ID’ names to the population-informed .
* bcftools reheader -s sample.txt -o BF\_65mb\_F4.filtered.vcf BF\_65mb\_F4\_filtered.vcf
  + Sample.txt - a text file containing old and new sample IDs in the same order as they appear in the VCF file
* Vcftools was used to to exclude the ‘new\_PA’ - PAU pop from the given data set
* vcftools --remove-indv new\_PA --vcf BF\_65mb\_F4.vcf --recode --out BF\_65mb\_F4\_filtered.vcf
  + --recode - This options generates a new file from the input after applying the filtering options specified.
* The Adgnet R script was used to run PCA, calculate K-means clustering, calculate matrix of genetic distances,run AMOVA and modify nexus file from Splitstree.
* Demographic relationships were inferred using two programmes -
  + Splitstree: Nei’s distance matrix generate with adgenet was input in Splitsree GUI, nexus file of demographic relationship from Splitstree was modified in the adgenet R script and reviewed in Splitstree.
  + fastStructure: source code: <https://github.com/rajanil/fastStructure/blob/master/README.md> was used to inferring population structure.
    - Plink2 was used to convert VCF to bed files.
      * plink2 --vcf BF\_65mb\_F4.4dg.filtered.vcf --make-bed --allow-extra-chr --out BF\_65mb\_F4.4dg
    - fastStructure:
      * python structure.py -K 6 --input=BF\_65mb\_F4.4dg --output=BF\_65mb\_F4\_faststructure.4dg
      * python distruct.py -K 6 --input=BF\_65mb\_F4\_faststructure.4dg --output=BF\_65mb\_F4.4dg.svg
      * Python /Users/DELL/miniconda3/pkgs/faststructure-1.0-py27h549429d\_0/bin/structure.py -K 6 --input=BF\_65mb\_F4.4dg --output=BF\_65mb\_F4\_faststructure.4dg
      * Python /Users/DELL/miniconda3/pkgs/faststructure-1.0-py27h549429d\_0/bin/distruct.py -K 6 --input=BF\_65mb\_F4\_faststructure.4dg --output=BF\_65mb\_F4.4dg.svg