wilding_paper

Josquin Daron

9 juillet 2021

1. Samples

- **AG1000g phase 2: 1142 genomes and 16 populations**

GNcol (4), GQgam (9), GHgam (12), FRgam (24), GNgam (40), KE (48), GHcol (55), GM (65), GAgam (69), CIcol (71), BFcol (75), AOcol (78), GW (91), BFgam (92), UGgam (112), CMgam (297)

- **Wilding genomes: 96 genomes and 3 populations**

32 LVBdom (Libreville, Gabon domestic)
32 LPdom (La lope, Gabon domestic)
32 LPfor (La lope, Gabon forest)

---

2. Dataset creation: reads mapping, SNP calling and filtering

2.1 Reads mapping

2.1.1 Bash script to perform: FASTQC, cutadapt, bwa mem, gatk realigner, bam report

- FASTQC report:

-> Wilding fastqc report: *not available because we've got mapped reads.*

- cutadapt:

```
cutadapt -a AGATCGGAAGAGCACACGTCTGAA -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT  -q 3
```

- bwa mem:

```
header=$(zcat $sampleId.R1.fastq.gz | head -n 1)
id=$(echo $header | head -n 1 | cut -f 1-4 -d":" | sed 's/@//' | sed 's/:/_/g')
sm=$(echo $header | head -n 1 | grep -Eo "[ATGCN]+$")
echo "Read Group @RG\tID:$id\tSM:$id"_"$sm\tLB:$id"_"$sm\tPL:ILLUMINA"

bwa mem -t 1 Anopheles_gambiae.AgamP4.dna.chr.fna $sampleId.R1.fastq.gz $sampleId.R2.fastq.gz -R $(echo
m\tPL:ILLUMINA") | samtools view -F 4 -b - | samtools sort - -o $sampleId.map.sort.bam
```

- gatk realigner:

```
java -jar ~/bioInf/bin/GenomeAnalysisTK-3.8-0-ge9d806836/GenomeAnalysisTK.jar -T RealignerTargetCreator
 -I $sampleId.map.sort.bam -o $sampleId.realignertargetcreator.intervals

java -Xmx8G -Djava.io.tmpdir=/tmp -jar ~/bioInf/bin/GenomeAnalysisTK-3.8-0-ge9d806836/GenomeAnalysisTK.j
ae.AgamP4.dna.chr.fna -targetIntervals $sampleId.realignertargetcreator.intervals -I $sampleId.map.sort
```

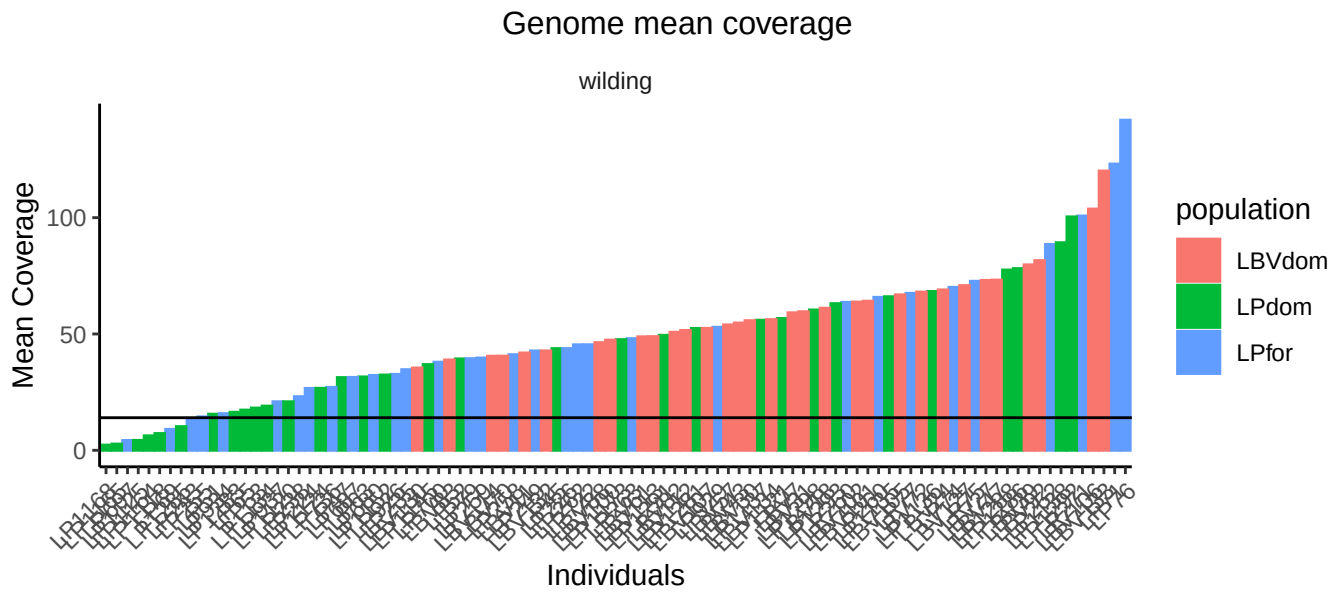
- bam file report (qualimap):

```
qualimap bamqc -bam /scratch/daron_anopheles/bam/$inputFile.indelrealigner.bam -c --java-mem-size=8G -ou
es/fastqBamInfo/bamInfo/jesus_qualimap/$inputFile.outqualimap -nt 2 -outformat HTML
```

-> wilding bam report: qualimap report git hub link.

2.1.2 Bam files analysis (genome depth, sex determination)

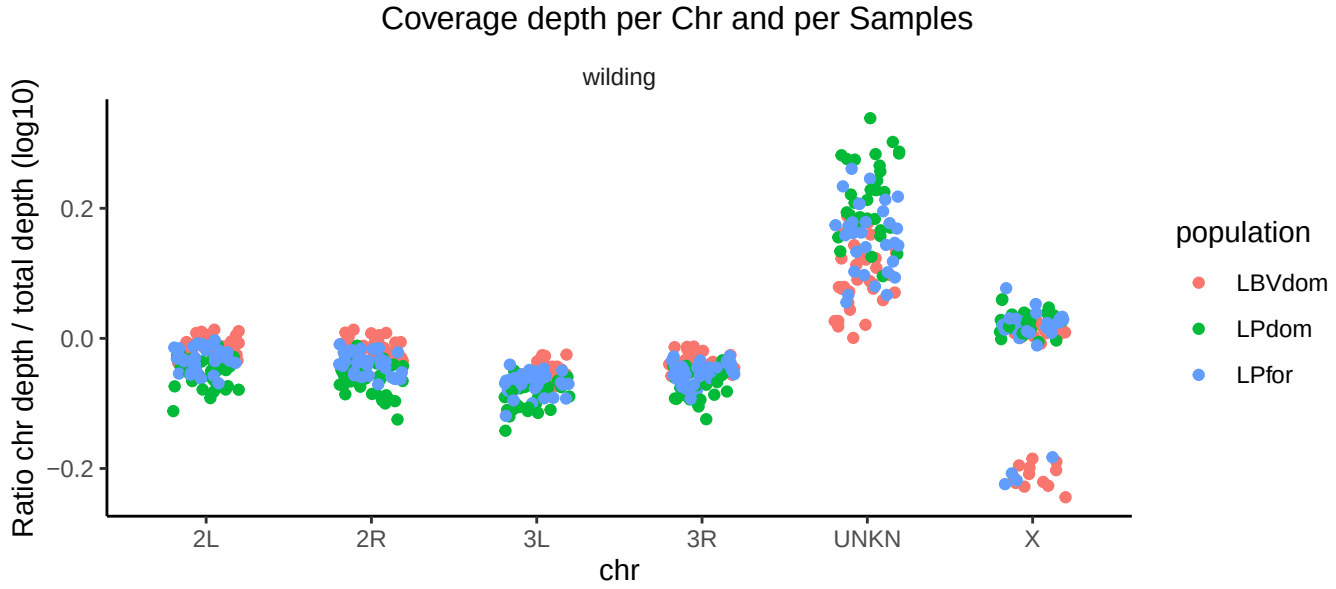
- Genome mean coverage:



–> **Filtering individuals: Remove individuals with mean coverage lower than 14x**

9 individuals from Wilding: LP69, LP243, LP697, LP1118, LP1125, LP1164, LP1165, LP1168, LP1285

- Determining sex of each samples:



Result:

- wilding: 16/96 Males

2.2 SNPs calling and filtering

2.2.1 SNPs calling script

- gatk unifiedGenotyper:

```
java -jar ~/bin/GenomeAnalysisTK-3.8-0-ge9d806836/GenomeAnalysisTK.jar -T UnifiedGenotyper -R Anopheles_
ist -L $interval --genotyping_mode DISCOVERY --downsampling_type BY_SAMPLE -dcov 250 --output_mode EMIT_
17 --genotype_likelihoods_model BOTH --heterozygosity 0.01 --indel_heterozygosity 0.001 -stand_call_con
rose.$out.unifiedGenotyper.vcf
```

2.2.2 SNP filtering:

- Jupyter-notebook script to generate html report on the newly created VCF file: Jupyter notebook VCF stat report code.

```
launch_ipynb.py -i vcfStats_slurm.ipynb -o wilding.chr3R.vcfStats_slurm.html
```

-> Wilding samples only SNPs stat report: Wilding SNP stat report.

Table 1: Final number of SNPs per chr after filtration

chr	NbSNP
2L	1228916
2R	1605477
3L	1159765
3R	1610164
X	295618

- 9 inds removed because imiss > 10%: LP1120 LP1134 LP1283 LP255 LP51 LP53 LP65 LP934 LP937
- Bash script to perform SNP and ind filtering (because scikit-allel is only outputing stats and cannot create a VCF or zarr file)

```
#!/bin/bash

# input file list
IN_VCF=$1          # input VCF file
AG_VCF_ACCESS=$2   # AG1000G VCF for genome accessibility, downloaded at ftp://ngs.sanger.ac.uk/producti
ssibility.X.vcf.gz
REF=$3
IND=$4

IN_PREFIX=`echo $IN_VCF | sed 's,\(.*\).vcf.gz,\1,'`

# Step 1: Select variants using GATK
echo "--> Step 1: Select variant based on GATK metrics `date`"
echo "Filter Expression QD < 5.00 || FS > 60.000 || ReadPosRankSum < -8.000"

java -jar ~/bioInf/bin/GenomeAnalysisTK-3.8-0-ge9d806836/GenomeAnalysisTK.jar -T SelectVariants -R $REF
 -selectType SNP

tabix $IN_PREFIX.snponly.vcf.gz

java -jar ~/bioInf/bin/GenomeAnalysisTK-3.8-0-ge9d806836/GenomeAnalysisTK.jar -T VariantFiltration -R $
```

```
PREFIX.annot.vcf.gz --filterExpression "QD < 5.00 || FS > 60.000 || ReadPosRankSum < -8.000 " --filterNa

zcat $IN_PREFIX.annot.vcf.gz | egrep -v "LOW_QUAL" | bgzip > $IN_PREFIX.passQC.vcf.gz

# Step 2: Select variants from inputed VCF based on genome accessibility
echo "--> Step 2: Choose variant based on genome accessibility `date`"
echo "minGQ 20 min DP 10"

zcat $AG_VCF_ACCESS | awk '{if($7=="PASS"){print $1"\t"$2}}' > $IN_PREFIX.pos

vcftools --gzvcf $IN_PREFIX.passQC.vcf.gz --positions $IN_PREFIX.pos --minGQ 20 --non-ref-ac-any 1 --re
> $IN_PREFIX.snpPassQC.vcf.gz

rm $IN_PREFIX.annot.vcf.gz $IN_PREFIX.snponly.vcf.gz $IN_PREFIX.passQC.vcf.gz

vcftools --gzvcf $IN_PREFIX.snpPassQC.vcf.gz --missing-site --stdout | awk '{if($6<0.05){print $0}}' |

vcftools --gzvcf $IN_PREFIX.snpPassQC.vcf.gz --positions $IN_PREFIX.lmiss --remove $IND --non-ref-ac-any
| bgzip > $IN_PREFIX.passQC.vcf.gz

rm $IN_PREFIX.snpPassQC.vcf.gz
```

- Summary of filtering step:

1. Remove individual with mean coverage lower than 14x
2. Discard SNPs present in none accessible area (defined in ag1000g), QD $<$ 5.00, FS $>$ 60.000 and ReadPosRankSum $<$ -8.000
3. Replace by NA genotypes with low call confidence (GQ$<$20)
4. Remove SNPs with $>$5% lmiss
5. Remove Inds with $>$10% imiss

---

3. Structure of genetic variation

Global genetic structure

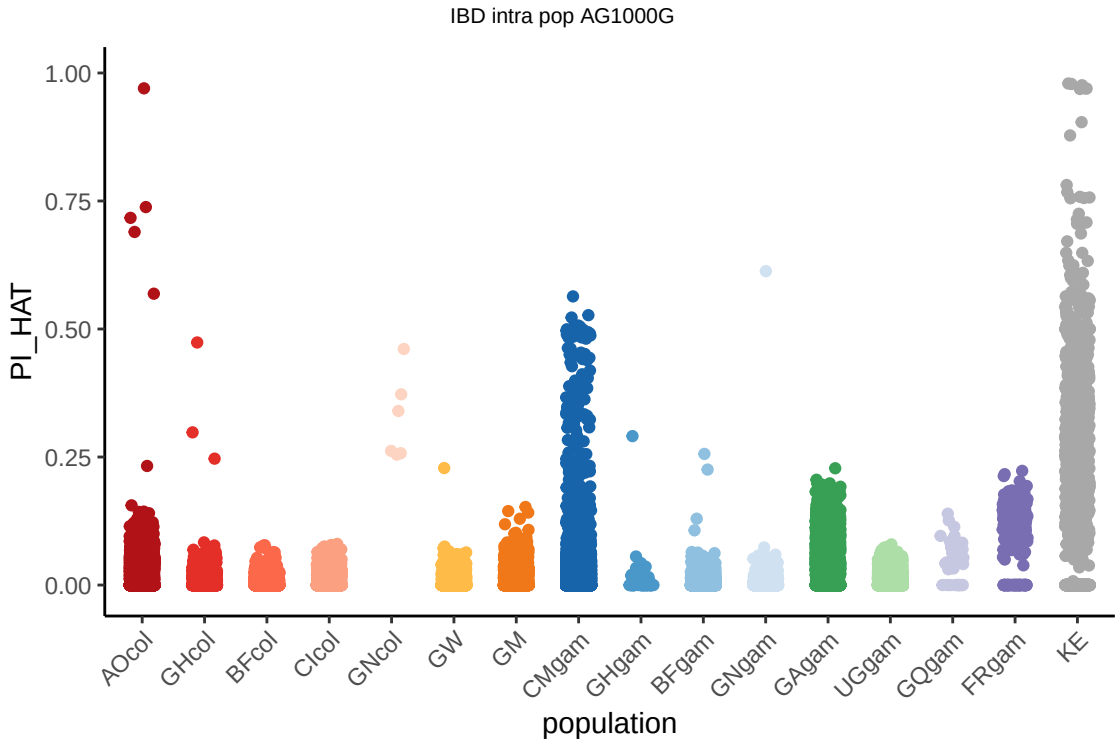
IBD anlaysis to identify closely related samples

Context: Downstream analysis required the absence of high related samples (i.e. more than we would expect by chance in a random sample). Goal: Identify putative closely related pair of sample using plink –genome

Bellow is the script for the IBD analysis:

```
# 1. Prune VCF using customized script. The program output the list of the coordinate of unlinked SNPs
prune_SNPs.py --snp wilding_urbano.3L.unifiedGenotyper.cov14x.passQC.vcf.gz --pop all --meta wilding_url

prune_SNPs.py --snp wilding_urbano.3R.unifiedGenotyper.cov14x.passQC.vcf.gz --pop all --meta wilding_url

# 2. Fetch pruned SNPs using VCFtools
vcftools --gzvcf wilding_urbano.3L.unifiedGenotyper.cov14x.passQC.vcf.gz --positions wilding_urbano.3L.p
 --recode --stdout | bgzip > wilding_urbano.3L.pruned.vcf.gz

vcftools --gzvcf wilding_urbano.3R.unifiedGenotyper.cov14x.passQC.vcf.gz --positions wilding_urbano.3R.p
 --recode --stdout | bgzip > wilding_urbano.3R.pruned.vcf.gz

vcf-concat wilding_urbano.3L.unifiedGenotyper.cov14x.passQC.vcf.gz wilding_urbano.3R.unifiedGenotyper.co

# 3. IBD with plink --genome
plink --vcf wilding_urbano.3.pruned.vcf.gz --allow-extra-chr --genome --out wilding_urbano.chr3.ibd
```

1. AG1000g



2. Wilding

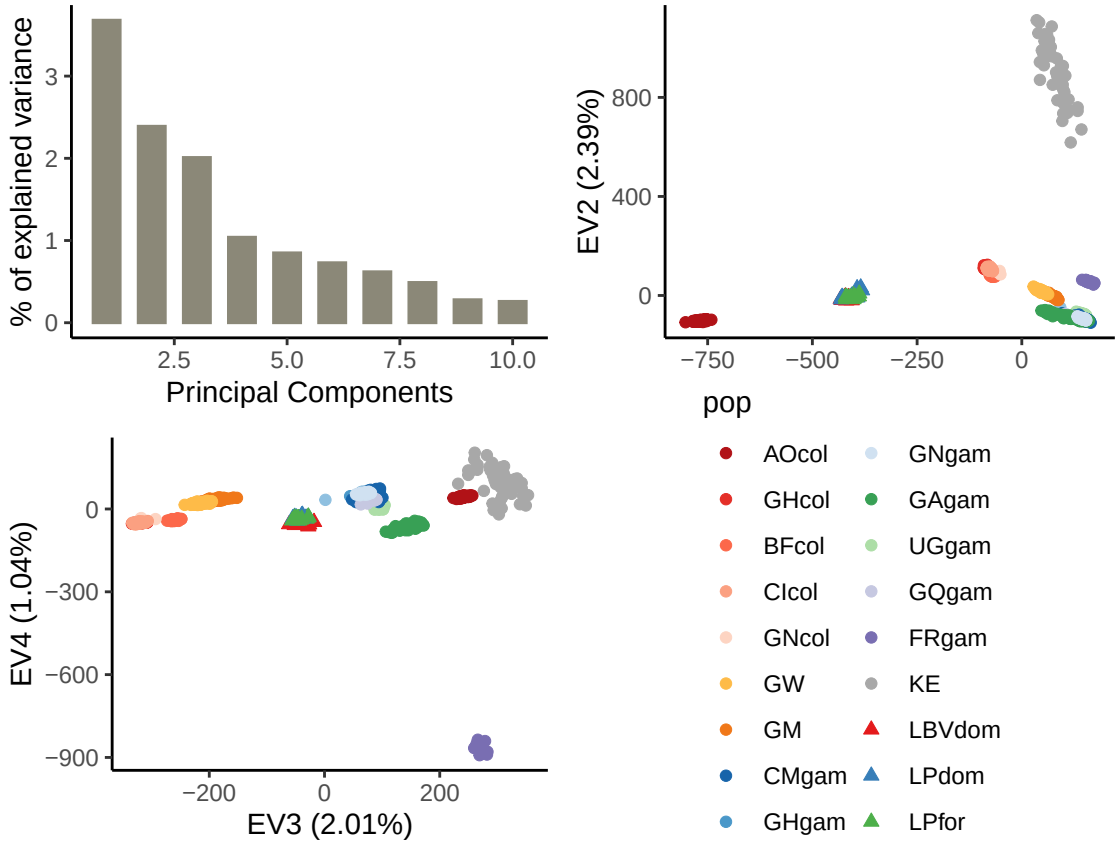
- Conclusion: Most sample pairs have low IBD value, meaning they are not related.

PCA

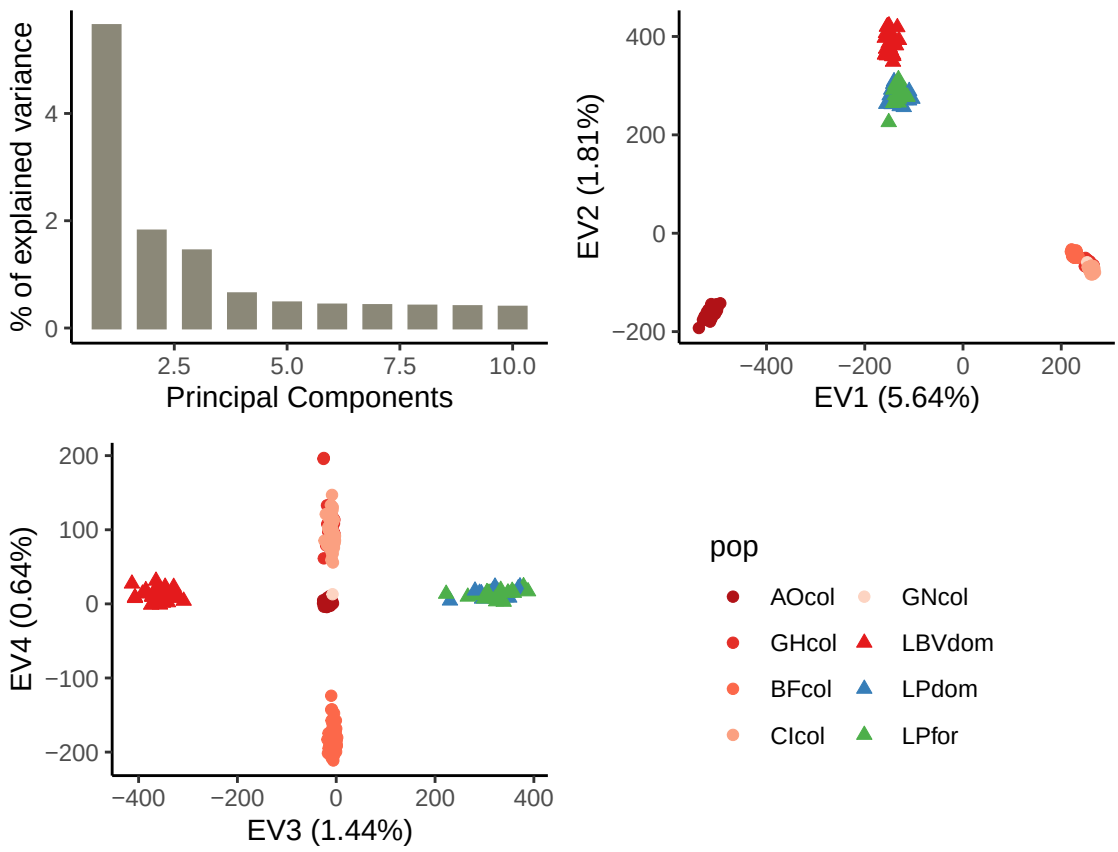
```
# PCA analysis was made using the jupyter-notebook script pca.ipynb
```

Question: How does the genetics variation of wilding samples is structured alone, or within the AG1000G samples?

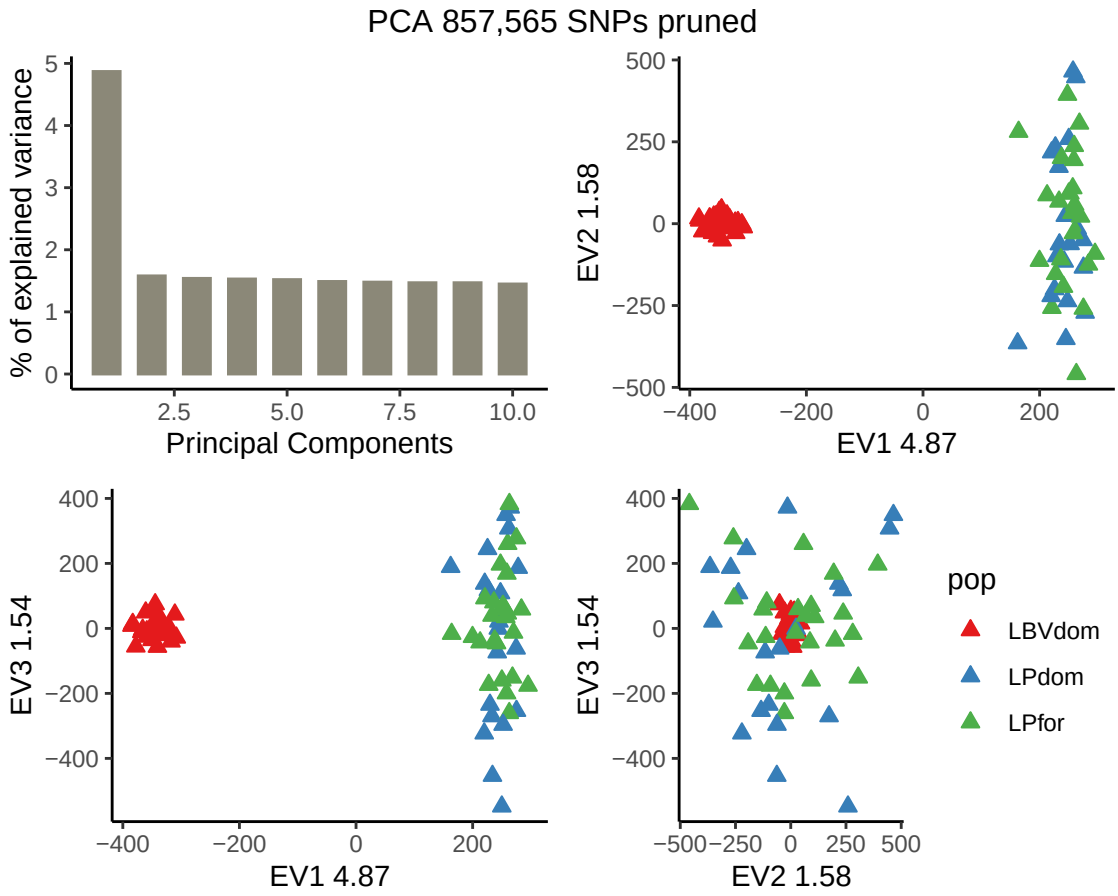
1. Wilding within AG1000G



2. Wilding within AG1000G colluzzi


% of explained variance
Principal Components
EV2 (1.81%)
EV1 (5.64%)
EV4 (0.64%)
EV3 (1.44%)
pop
AOcol
GNcol
GHcol
LBVdom
BFcol
LPdom
CIcol
LPfor

3. Wilding



- Conclusion:

1. Wilding genome are located between the AOcol and GHcol population, as expected.
2. Based on the wilding pca, we can distinguish clearly two pop: (i) LBVdom and (ii) LPdom-LPfor. LPdom and LPfor even they have been samples at 15km away from each other they represent one single population

Admixture

Question: Same question than above adding the admixture component.

- lauch admixture on the cluster, 1 job per k, 10 time a random set of 100kSNPs and 10 seed per samples set (N=100 per k):

```
path_to_tmp="/scratch/daron_$SLURM_JOB_ID"
IN_PREFIX="ag1000g.phase2.anopheles-rose.merged.biallelic.3.pruned"
K=${1}

# download vcf file
scp nas3:/data3/projects/plasmodium/anopheles/vcf_store/$IN_PREFIX.vcf.gz $path_to_tmp

# run admixture 10 resamples of 100kSNPs with 10 different seed; N=100
N=0
for i in {1..10}; do
    zcat $IN_PREFIX.vcf.gz | egrep -v "#" | cut -f 1,2 | shuf -n 100000 | sort -k 1n,1n -k 2n,2n > pos
    vcftools --gzvcf $IN_PREFIX.vcf.gz --positions pos --stdout --recode | sed 's,^3L,1,' | sed 's,^3R,
```

```
    tabix samp.vcf.gz
    plink --vcf samp.vcf.gz --make-bed  --allow-extra-chr --out samp

    for j in {1..10}; do
        N=$(( $N + 1 ))
        admixture --cv --seed=$RANDOM -j9 samp.bed $K | tee $IN_PREFIX.n$N.log${K}.out
        mv samp.$K.P $IN_PREFIX.n$N.$K.P
        mv samp.$K.Q $IN_PREFIX.n$N.$K.Q
        scp $IN_PREFIX.n$N.$K.P $path_to_dir/
        scp $IN_PREFIX.n$N.$K.Q $path_to_dir/
        scp $IN_PREFIX.n$N.log${K}.out $path_to_dir/
    done
done
```

- reorder ind based on a customized order:

```
R --vanilla --args $PWD/'admixture output file' $PWD/'ind ordered as in the admix' $PWD/'ind new order'
```

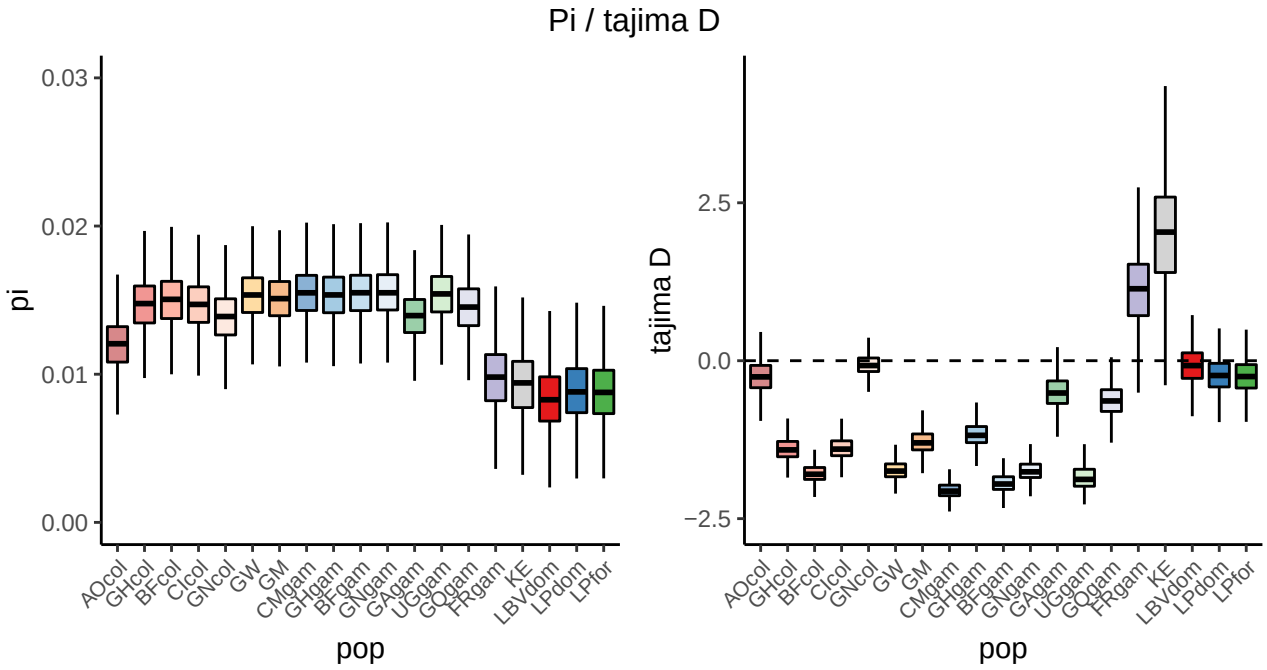
- run clumpack

```
CLUMPAK.pl -id 10 -dir ./ -file out_ordered_ag1000gCol_wilding.zip -inputtype Admixture -indtopop ag1000

option help:
-id Job id (defaut random number generated)
-inputtype Input file format [Admixture|Structure]
-indtopop population label (only pop name in the same order than input file)
```

Stat descriptives

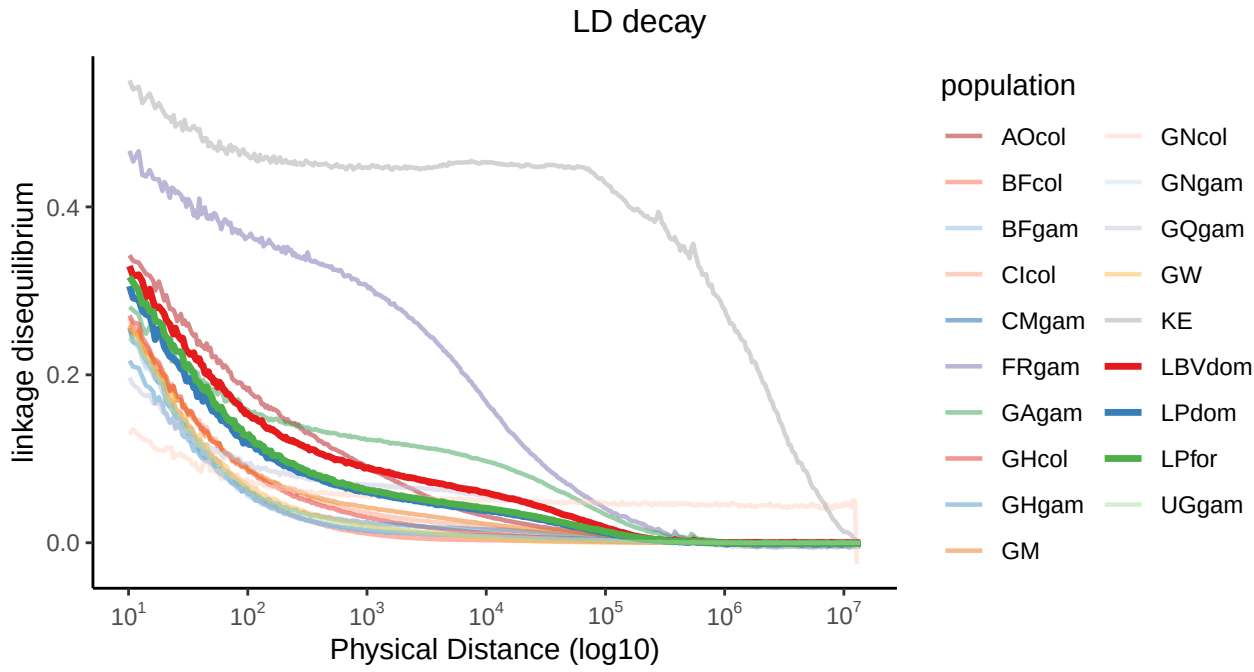
- Pi and tajima D

```
popStatsDescWindows.py --snp anopheles-rose.unifiedGenotyper.cov14x.passQC.zarr --keep LPdom.ind --bed a
```



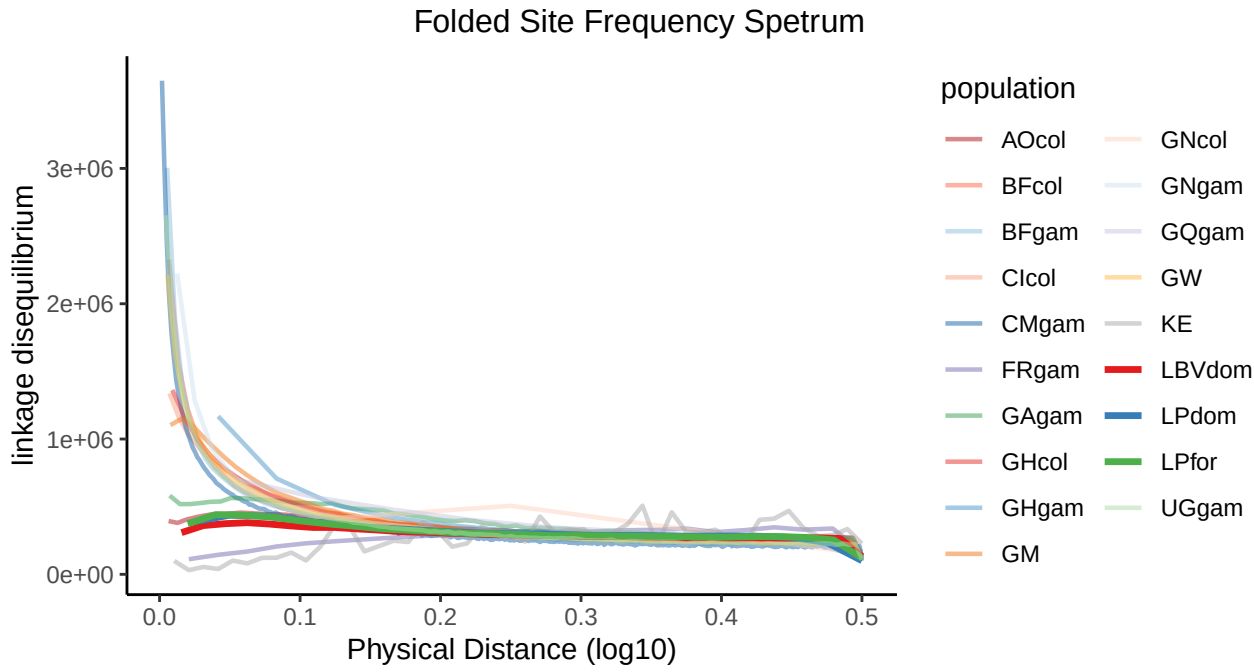
- LDdecay Here is the link to the jupyter-notebook script: LdDecay.

```
ldDecay.ipynb
```



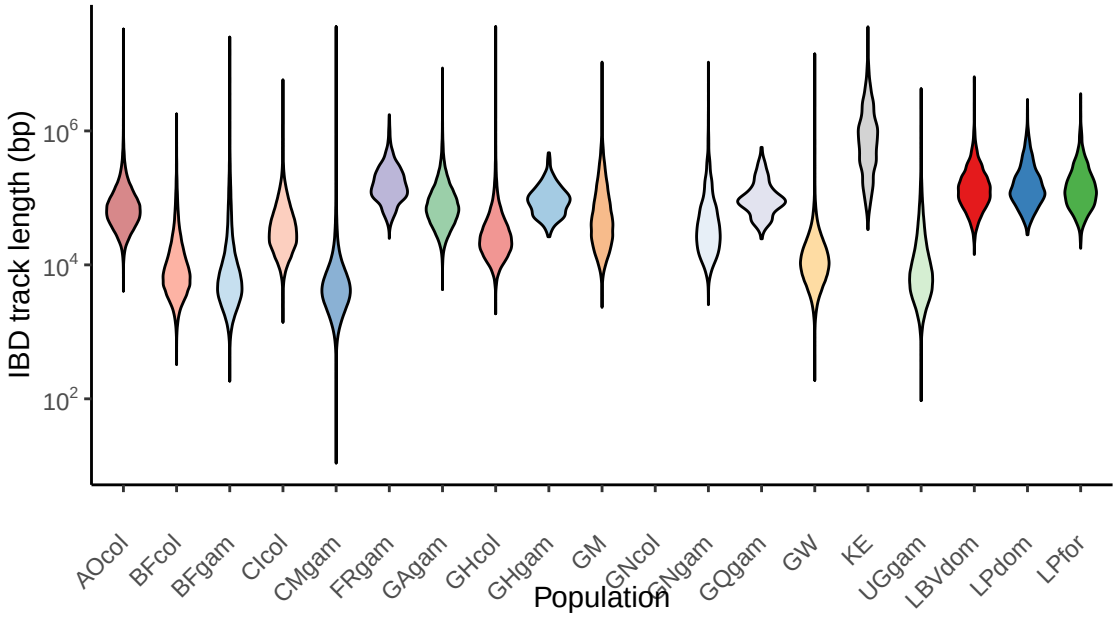
- Folde Site Frequency spetrum

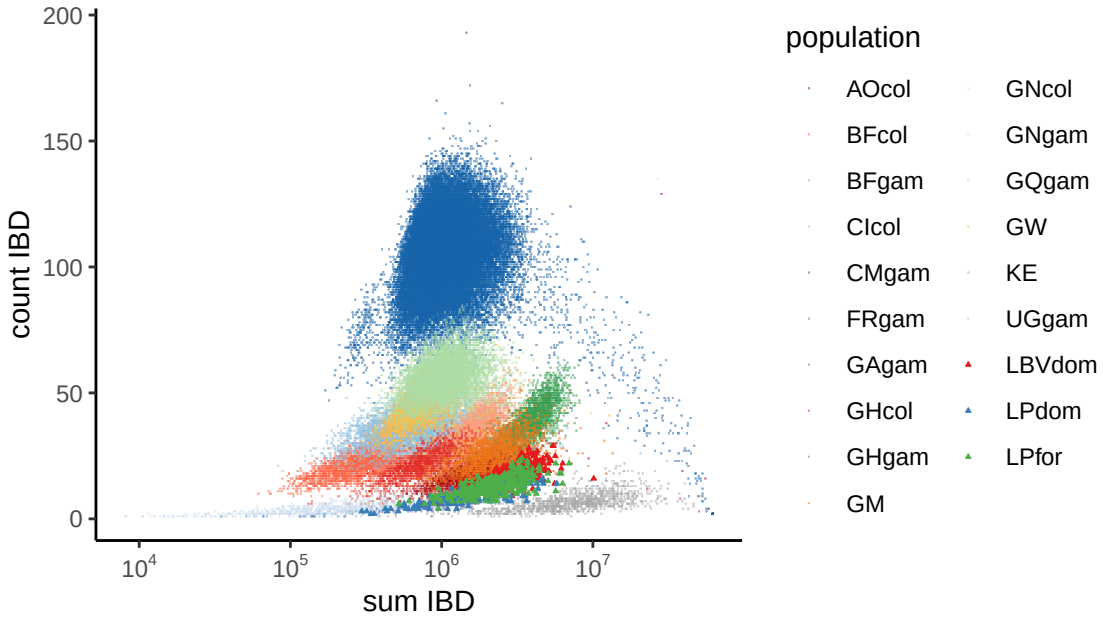
```
sfs.py --zarr ag1000g.phase2.anopheles-rose.merged.biallelic.zarr --keep LPdom.ind --bed ag1000g.chr3pe
```



- IBD

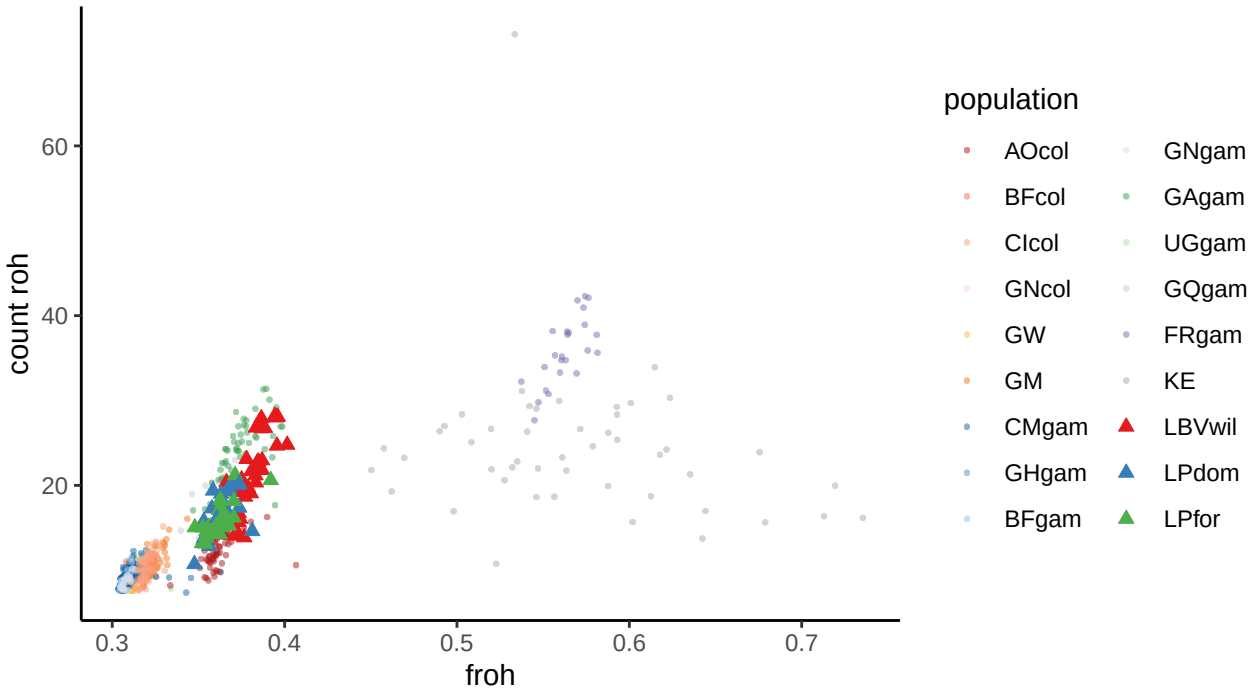
```
egrep -w $pop wilding.samples.meta.txt | cut -f 1 > k

vcftools --gzvcf anopheles-rose.3L.unifiedGenotyper.cov14x.passQC.vcf.gz --chr 3L --from-bp 15000000 --

vcftools --gzvcf anopheles-rose.3R.unifiedGenotyper.cov14x.passQC.vcf.gz --chr 3R --from-bp 1000000 --t

java -Xmx8g -jar ~/bioInf/bin/ibdseq.r1206.jar gt=$pop.3L.vcf.gz out=$pop.ibdseq.3L
java -Xmx8g -jar ~/bioInf/bin/ibdseq.r1206.jar gt=$pop.3R.vcf.gz out=$pop.ibdseq.3R
```





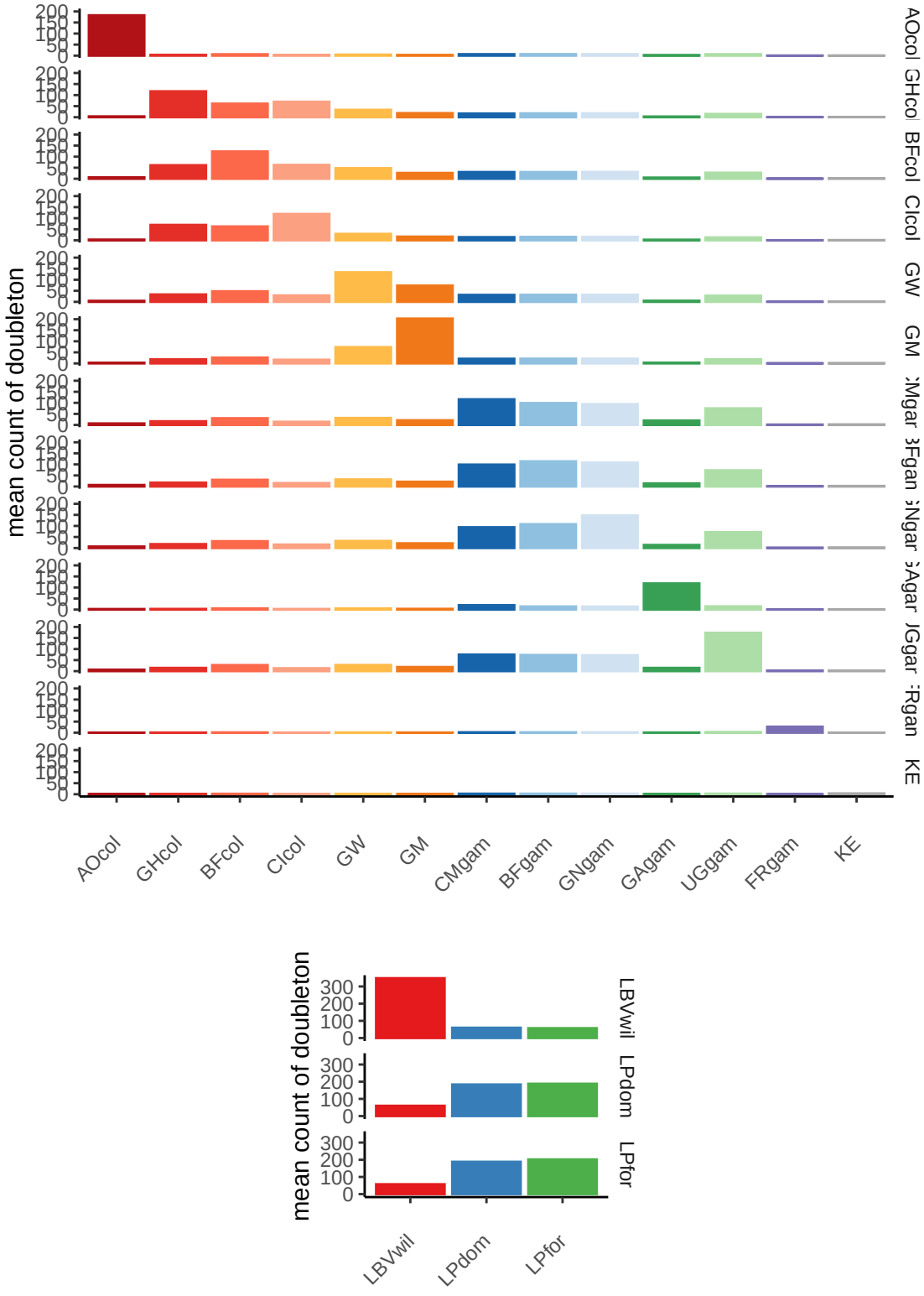
- roh

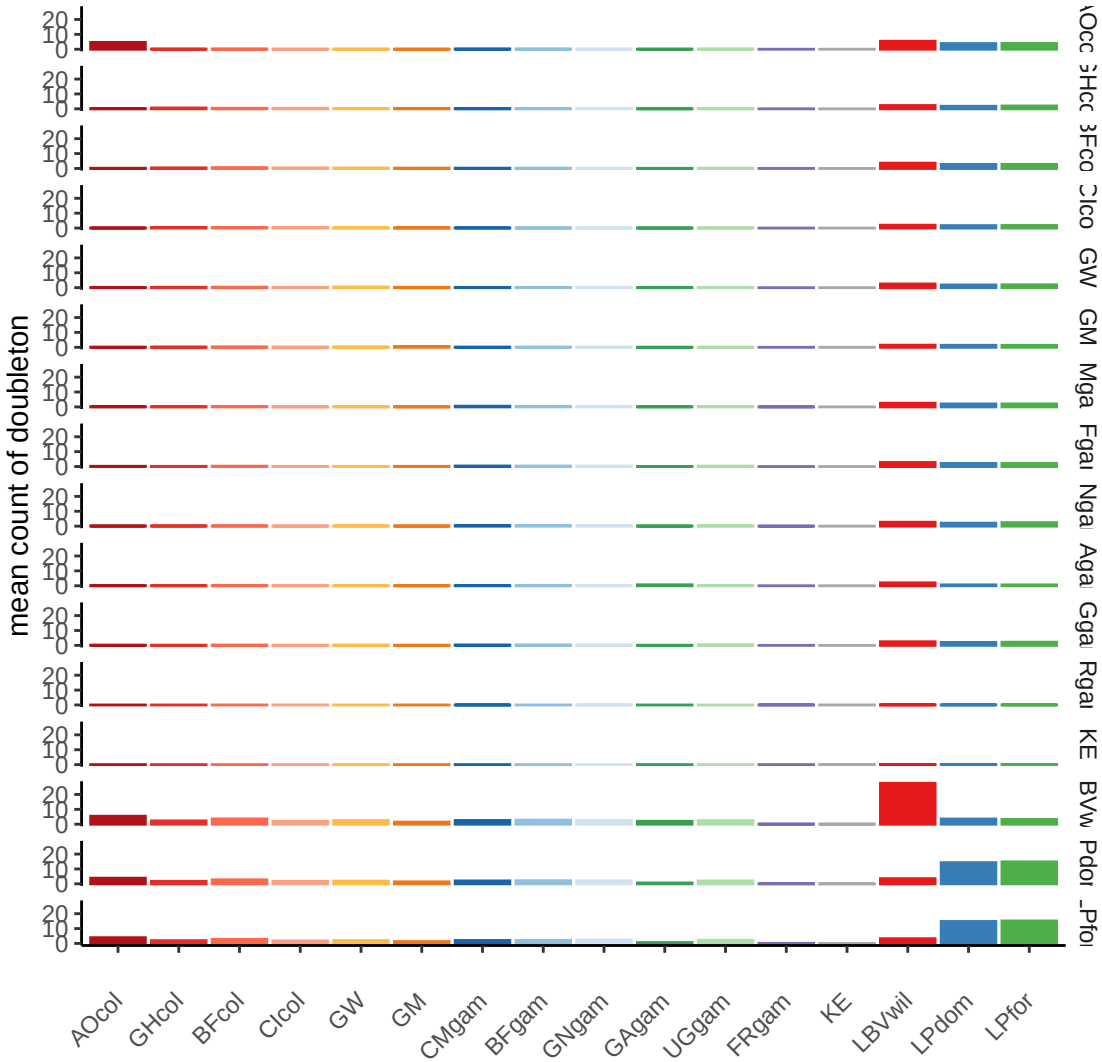
```
get_roh.py --snp ag1000g.phase2.anopheles-rose.merged.biallelic.zarr --keep LPdom.ind --bed ag1000g.chr
```



- doubleton sharing

```
doubleton_sharing.py --meta wilding.meta --bed ag1000g.chr3pericentroRegion.bed -z wilding.unifiedGenoty
```


mean count of doubleton
AOcol
GHcol
BFcol
CIcol
GW
GM
CMgam
BFgam
GNgam
GAgam
UGgam
FRgam
KE
LBVwil
LPdom
LPfor



Deomgraphic history:

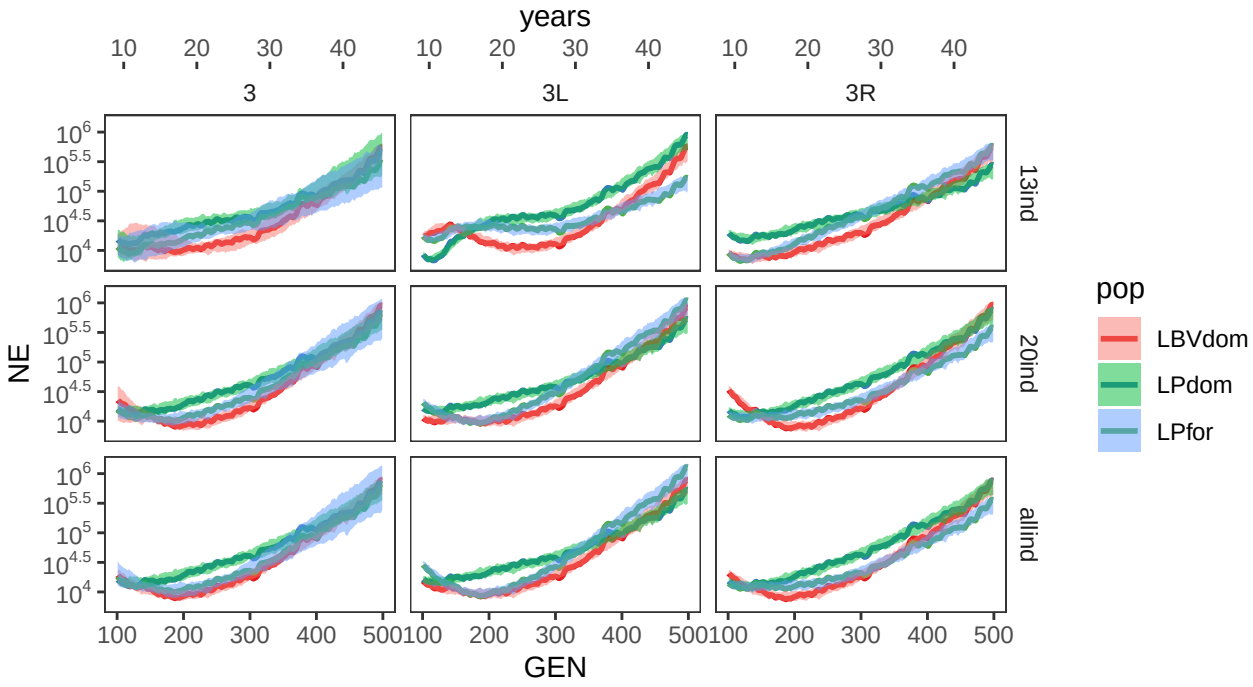
Investigate recent changes in population size over time.

IBDne

Context: IBDne infer recent population history (last 500 generation). Based on the population structure we observed, with two distinct population, one in LBVdom et one in LPdom et LPfor, looking at recent demographic change in each sample group is the most likely way to observe differences. Question: Does LPdom et LPfor have exhibit recent demography changes ?

```
cat $pop.ibdseq.3*.ibd | java -jar ibdne.04Sep15.e78.jar map=ag.chr3.map out=$pop.3.ne nthreads=7 minib
```

Recombination map can be find here: AG1000G recombination map.

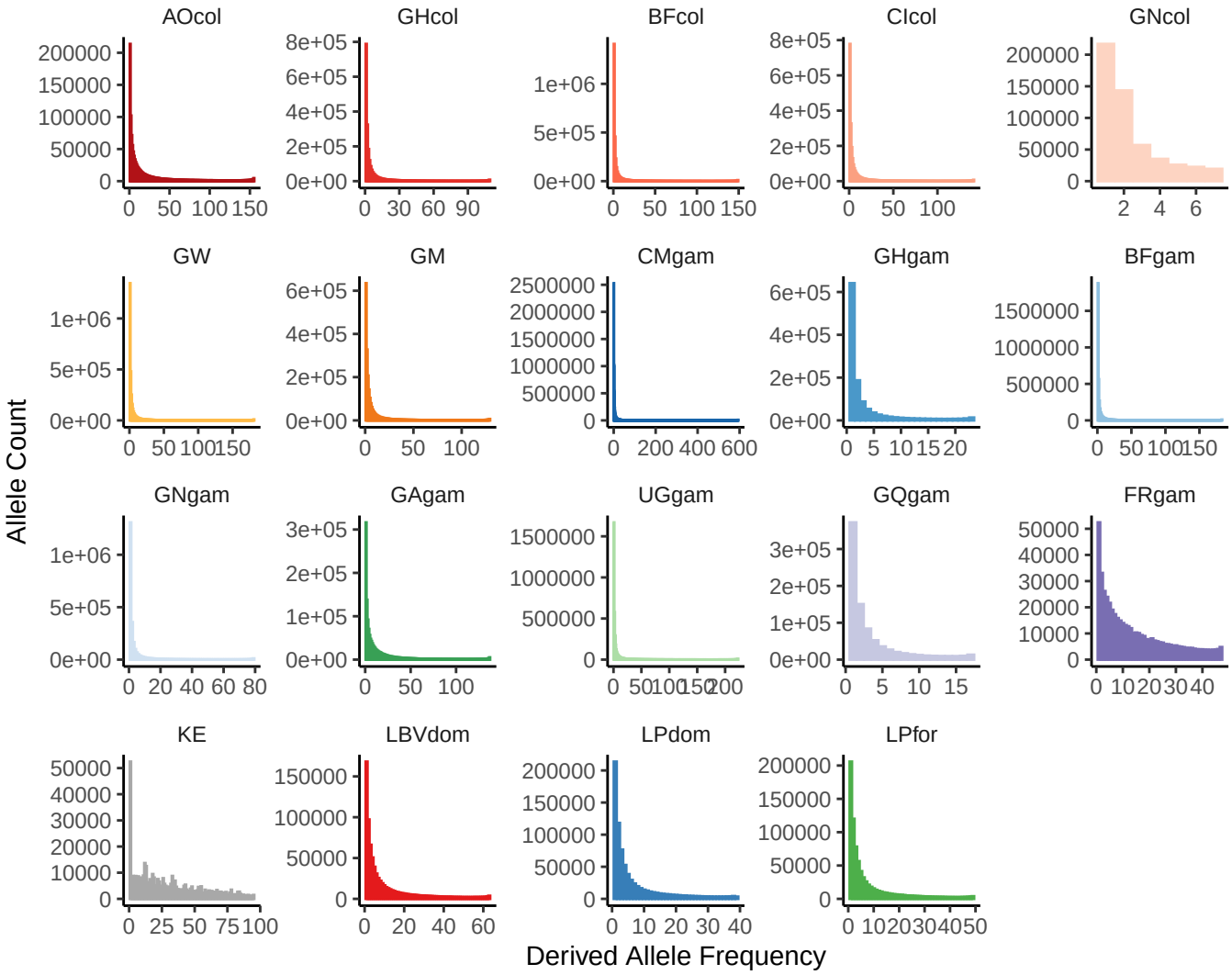


IBDne was run with 3 samples size: -allind = all the ind of each pop
-20ind = the same number of ind accross populations N=20
- 13ind = considering only individual collected during the mission Nov-16 N=13

Stairway plot

SNP polarization: In order to generate a unfolded site frequency spetrum we need to polarize the SNPs using an outgroup. Here we used A. merus as outgroup with N=10 individuals.

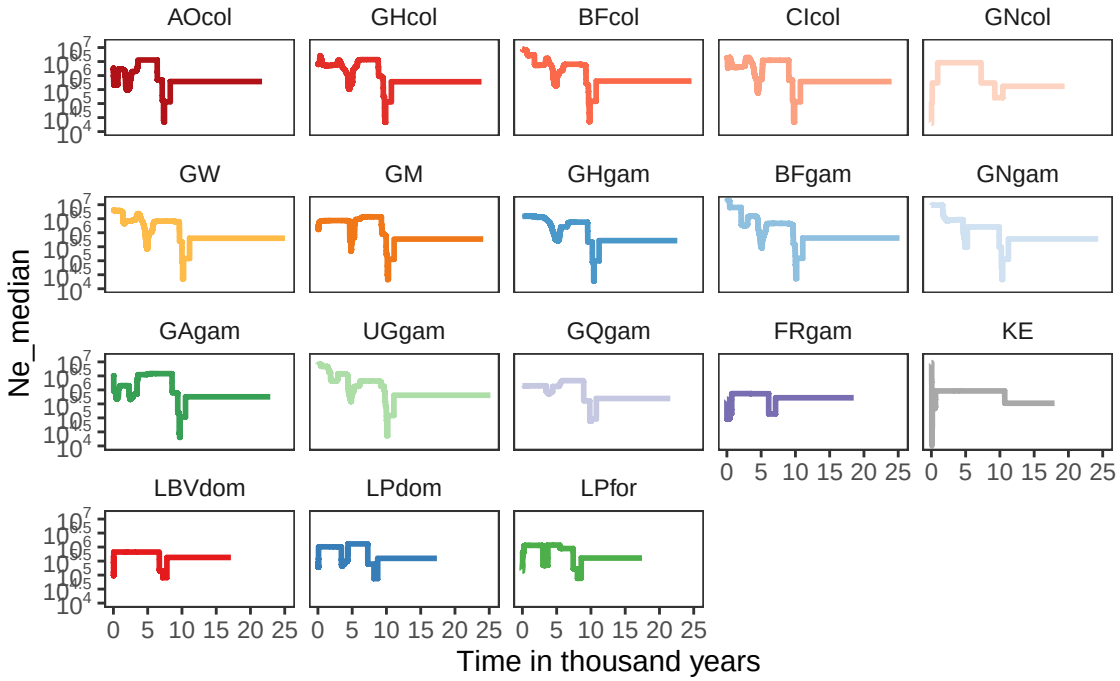
```
### merge wilding VCF with merus VCF
join -1 1 -2 1 anopheles-rose.$chr.unifiedGenotyper.cov14x.passQC.pos chri_meru.merged.pos | sed 's,_,\t

bcftools merge -i - -m snps anopheles-rose.$chr.unifiedGenotyper.cov14x.passQC.vcf.gz chri_meru.merged.v

### polarize REF allele using outgroup
# samll header printing issue in the polarizeVCF.py scrit so header need to be adding separatly
# use major allel as ancestral allel, discar 0.5:0.5 SNPs, site need a MAF of 0.1 to be considered
polarizeVCF.py --vcf anopheles-rose.$chr.unifiedGenotyper.cov14x.passQC.merge.vcf.gz --keep chri_meru.ic


sfs.py --zarr ag1000g.phase2.anopheles-rose.merged.biallelic.zarr --keep LPdom.ind --bed ag1000g.chr3pei
```



b. Stairway plot

```
pop=${1}
inputSFS=${2}
sfs=$(cat /data3/projects/plasmodium/anopheles/popstructure/statDesc/usfs/$inputSFS | cut -f 3 | sed '1d
nbseq=$(cat /data3/projects/plasmodium/anopheles/popstructure/statDesc/usfs/$inputSFS | cut -f 3 | sed
nbSite=30484401
nrand_1=$(( ($nbseq-2)/4 ))
nrand_2=$(( ($nbseq-2)/2 ))
nrand_3=$(( (($nbseq-2)*3)/4 ))
nrand_4=$(($nbseq-2))

### create blueprint file
echo "popid: $pop # id of the population (no white space)" > run.blueprint
echo "nseq: $nbseq # number of sequences" >> run.blueprint
echo "L: $nbSite # total number of observed nucleic sites, including polymorphic and monomorphic"  >> ru
echo "whether_folded: false # whethr the SFS is folded (true or false)" >> run.blueprint
```

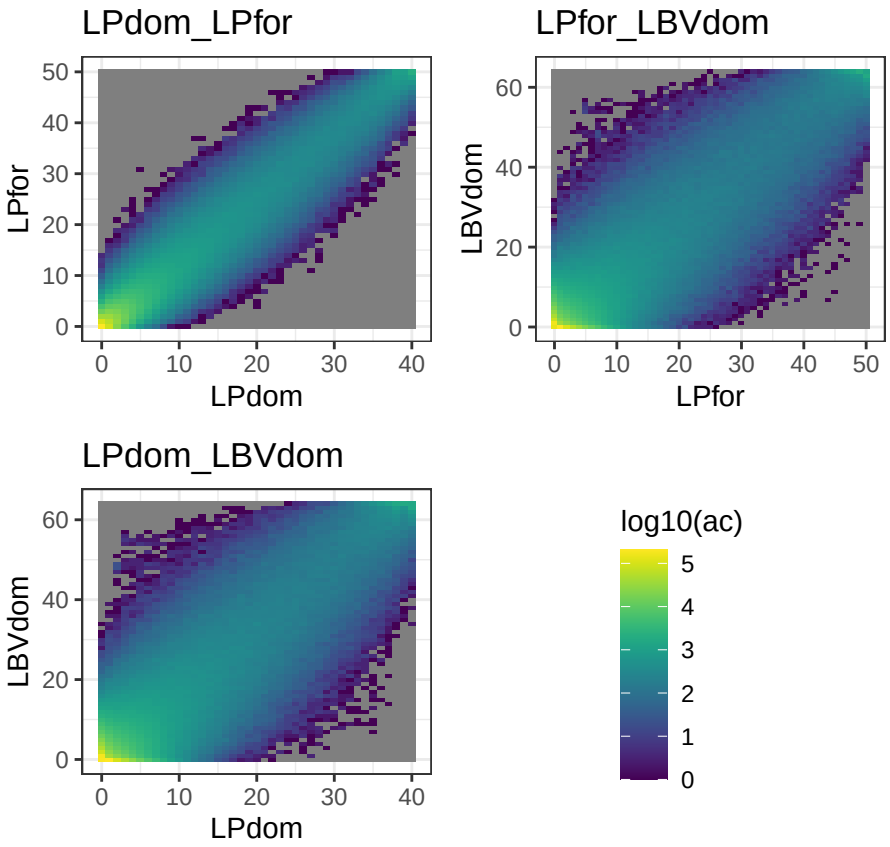
```
echo "SFS: $sfs # snp frequency spectrum: number of singleton, number of doubleton, etc. (separated by
echo "#smallest_size_of_SFS_bin_used_for_estimation: 1 # default is 1; to ignore singletons, uncomment
> run.blueprint
echo "#largest_size_of_SFS_bin_used_for_estimation: 29 # default is n-1; to ignore singletons, uncommen
eq-2" >> run.blueprint
echo "nrand: $nrand_1   $nrand_2    $nrand_3    $nrand_4 # number of random break points for each try (
n.blueprint
echo "project_dir: $pop # project directory" >> run.blueprint
echo "stairway_plot_dir: /home/daron/bioInf/bin/stairway_plot_v2.1.1/stairway_plot_es # directory to th
echo "ninput: 200 # number of input files to be created for each estimation" >> run.blueprint
echo "#random_seed: $RANDOM" >> run.blueprint
echo "#output setting" >> run.blueprint
echo "mu: 2.8e-9 # assumed mutation rate per site per generation" >> run.blueprint
echo "year_per_generation: 11 # assumed generation time (in years)" >> run.blueprint
echo "#plot setting" >> run.blueprint
echo "plot_title: $pop.plot # title of the plot" >> run.blueprint
echo "xrange: 0.1,10000 # Time (1k year) range; format: xmin,xmax; "0,0" for default" >> run.blueprint
echo "yrange: 0,0 # Ne (1k individual) range; format: xmin,xmax; "0,0" for default" >> run.blueprint
echo "xspacing: 2 # X axis spacing" >> run.blueprint
echo "yspacing: 2 # Y axis spacing" >> run.blueprint
echo "fontsize: 12 # Font size" >> run.blueprint

### Run stairwayplot2
java -cp /home/daron/bioInf/bin/stairway_plot_v2.1.1/stairway_plot_es Stairbuilder run.blueprint
bash run.blueprint.sh
```



Population differenciation and gene flow inference

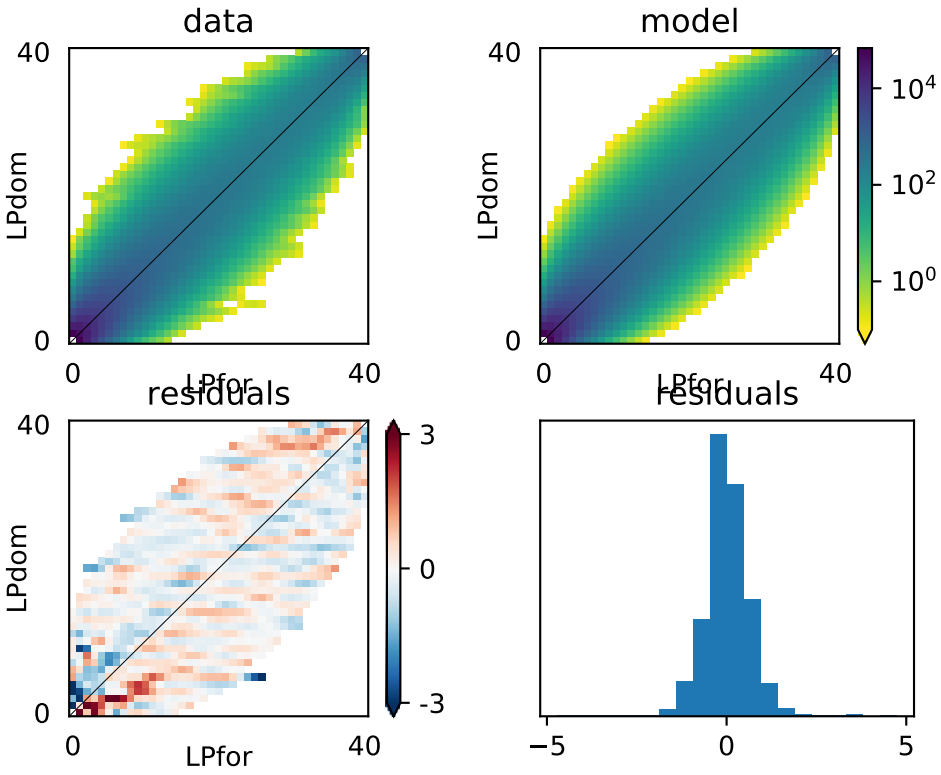
a. Calculate the join site frequency spectrum (jsfs) of the three pair of pops.

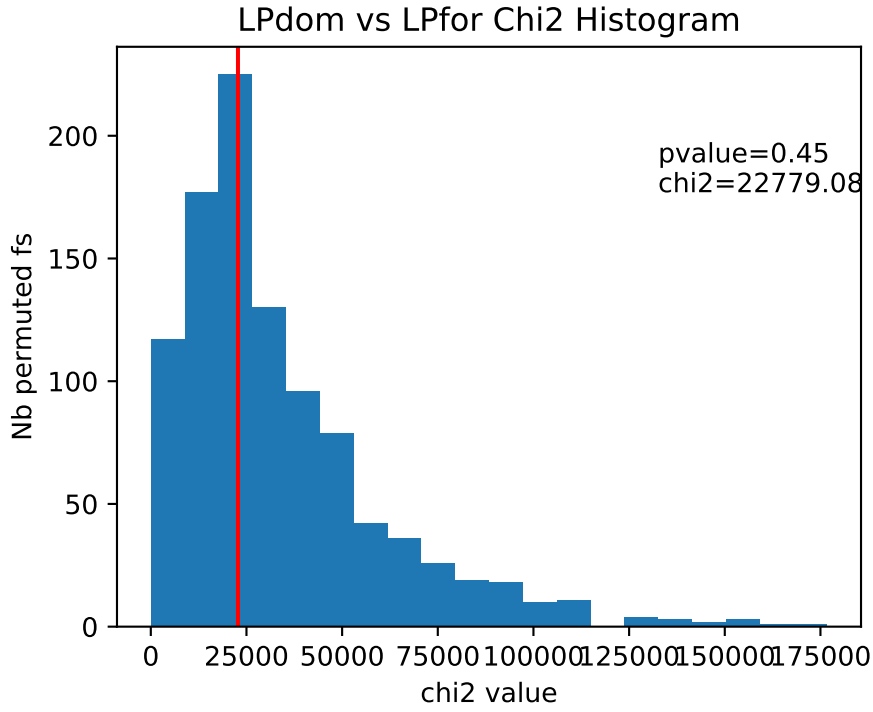
```
sfs.py --zarr ag1000g.phase2.anopheles-rose.merged.biallelic.zarr --keep LPdom.ind --keep2 LPfor.ind --
```



b. Formaly test whether LPdom and LPfor are two separate population or form one single panmictic population

Panmixia test has been done using a jupyter-notebook script available here : Test panmixia.



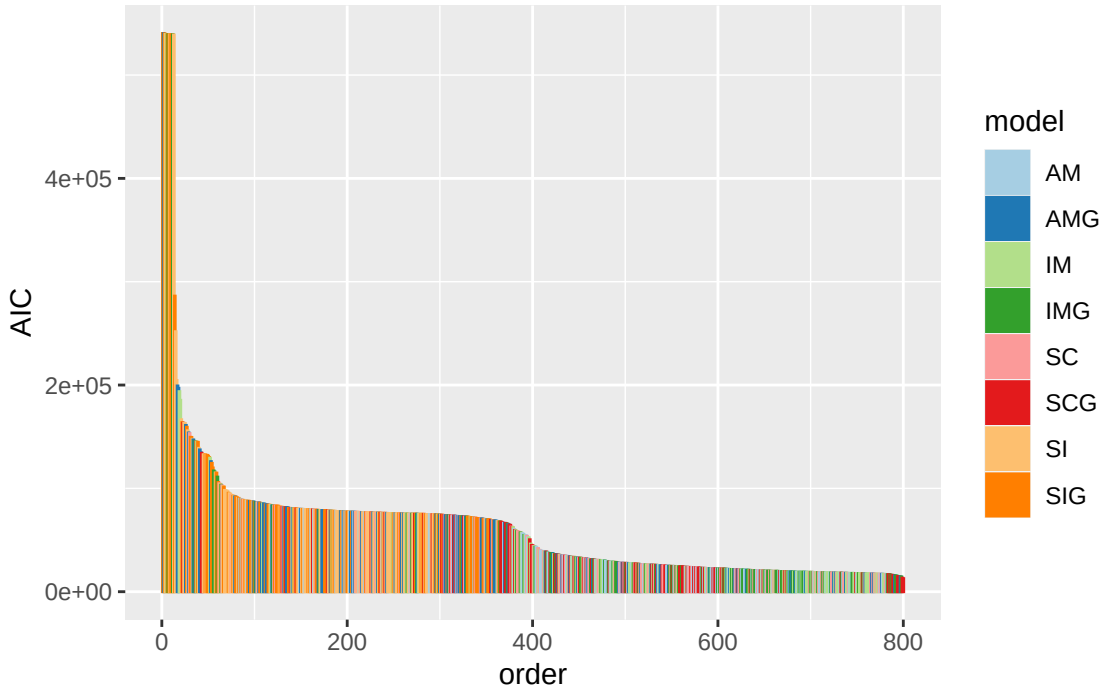


Results: - Chi2 value of obs-scrambled jsfs LPdom-LPfor falls within the null distribution (build from 1000 permutation of pop labels). H0 of no difference between the two pop is accepted.

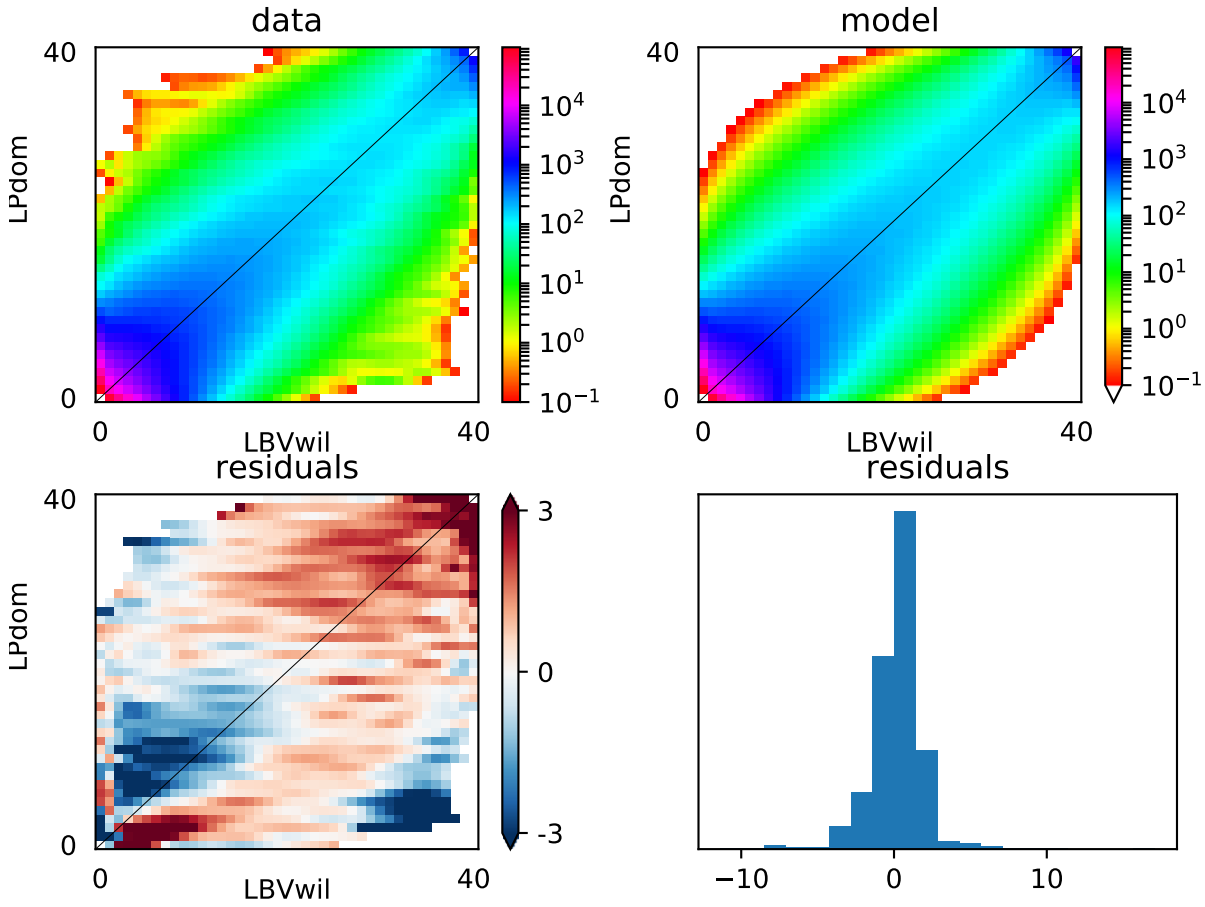
Dadi 2D

Run 100 each of the 8model and plot AIC

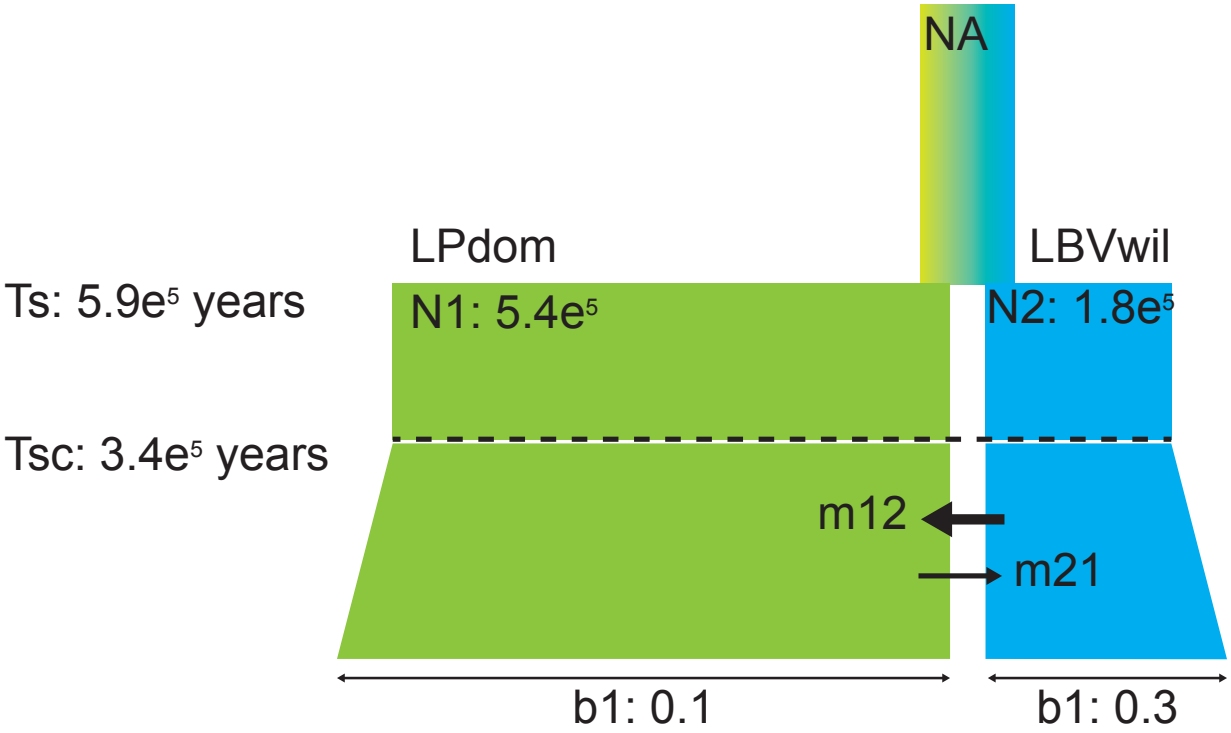
```
for i in {1..100}
do
    echo $i
    script_inference_demo_new_models.py  -x LBVwil -y LPdom -o $outdir.${i} --fs_file_name anopheles-ros
done
```



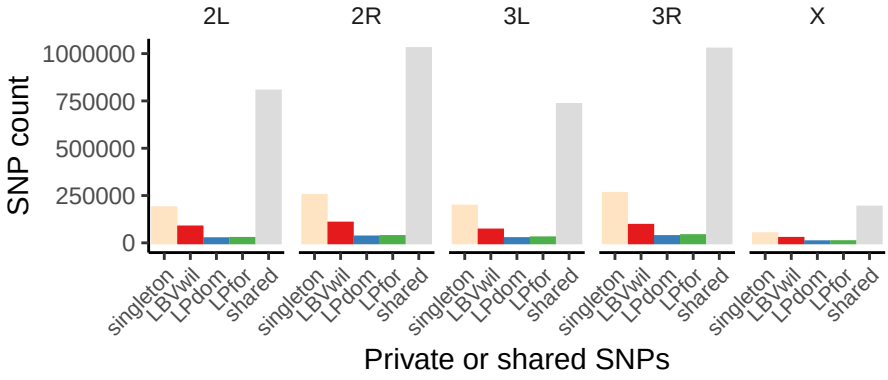
SCG best model

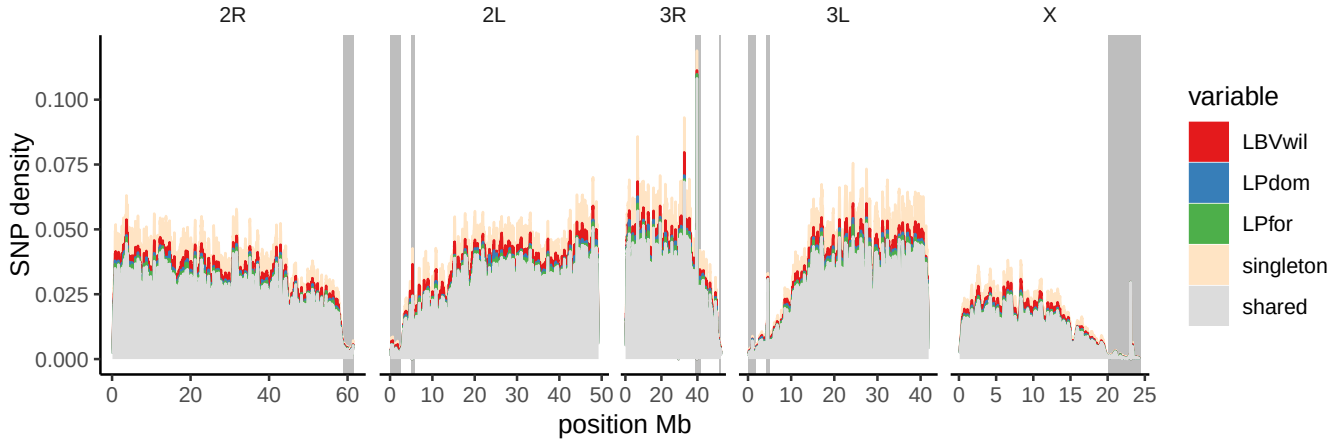


SCG draw best model


NA
LPdom
LBVwil
Ts: 5.9e5 years
N1: 5.4e5
N2: 1.8e5
Tsc: 3.4e5 years
m12
m21
b1: 0.1
b1: 0.3

4. Selection

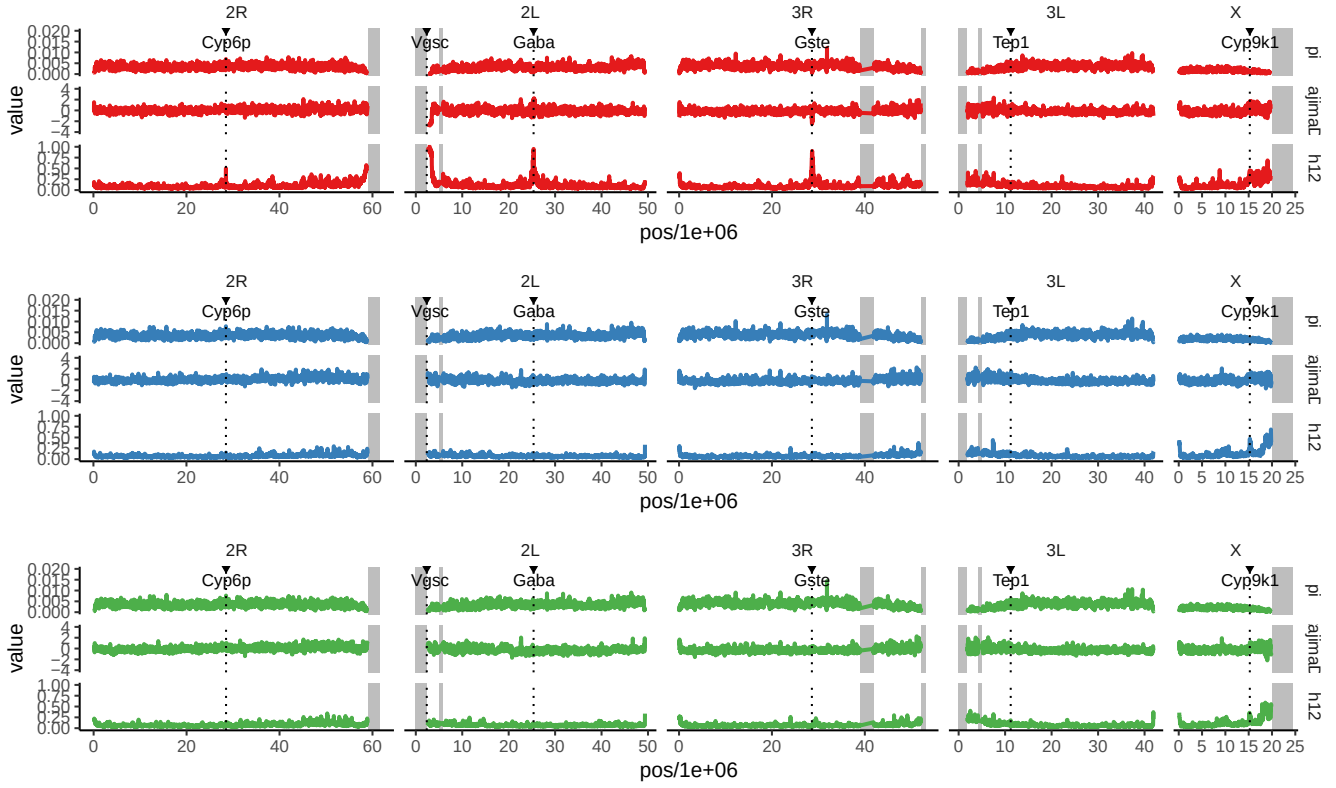

2L
2R
3L
3R
X
SNP count
1000000
750000
500000
250000
0
singleton
LBVwil
LPdom
LPfor
shared
Private or shared SNPs

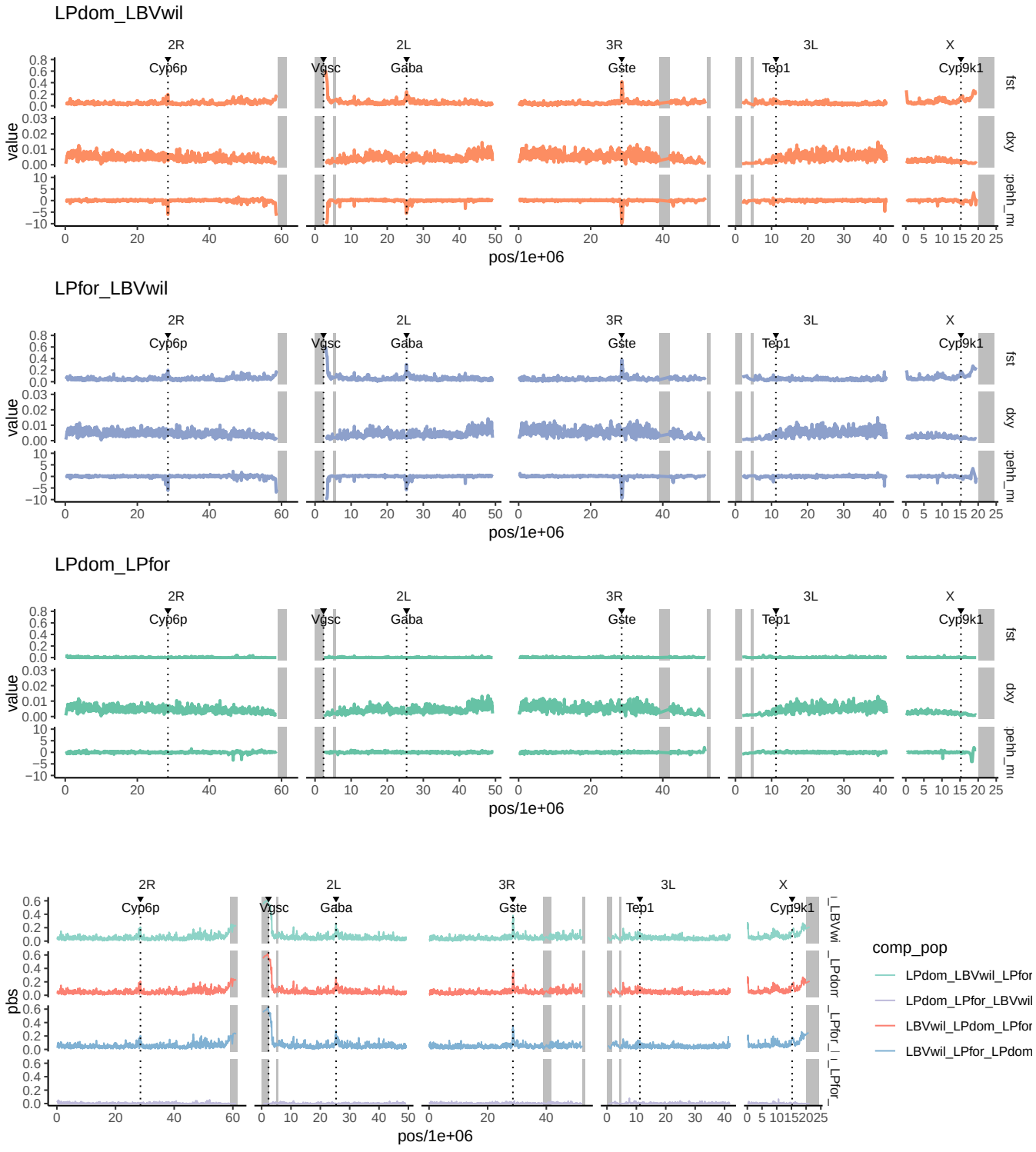



```
popStatsDescWindows.py --snp wilding.unifiedGenotyper.cov14x.passQC.phased.zarr/ -b bed  -o LBVwil --ke
popStatsDescWindows.py --snp wilding.unifiedGenotyper.cov14x.passQC.phased.zarr/ -b bed  -o LPdom --kee
popStatsDescWindows.py --snp wilding.unifiedGenotyper.cov14x.passQC.phased.zarr/ -b bed  -o LPfor --kee
```

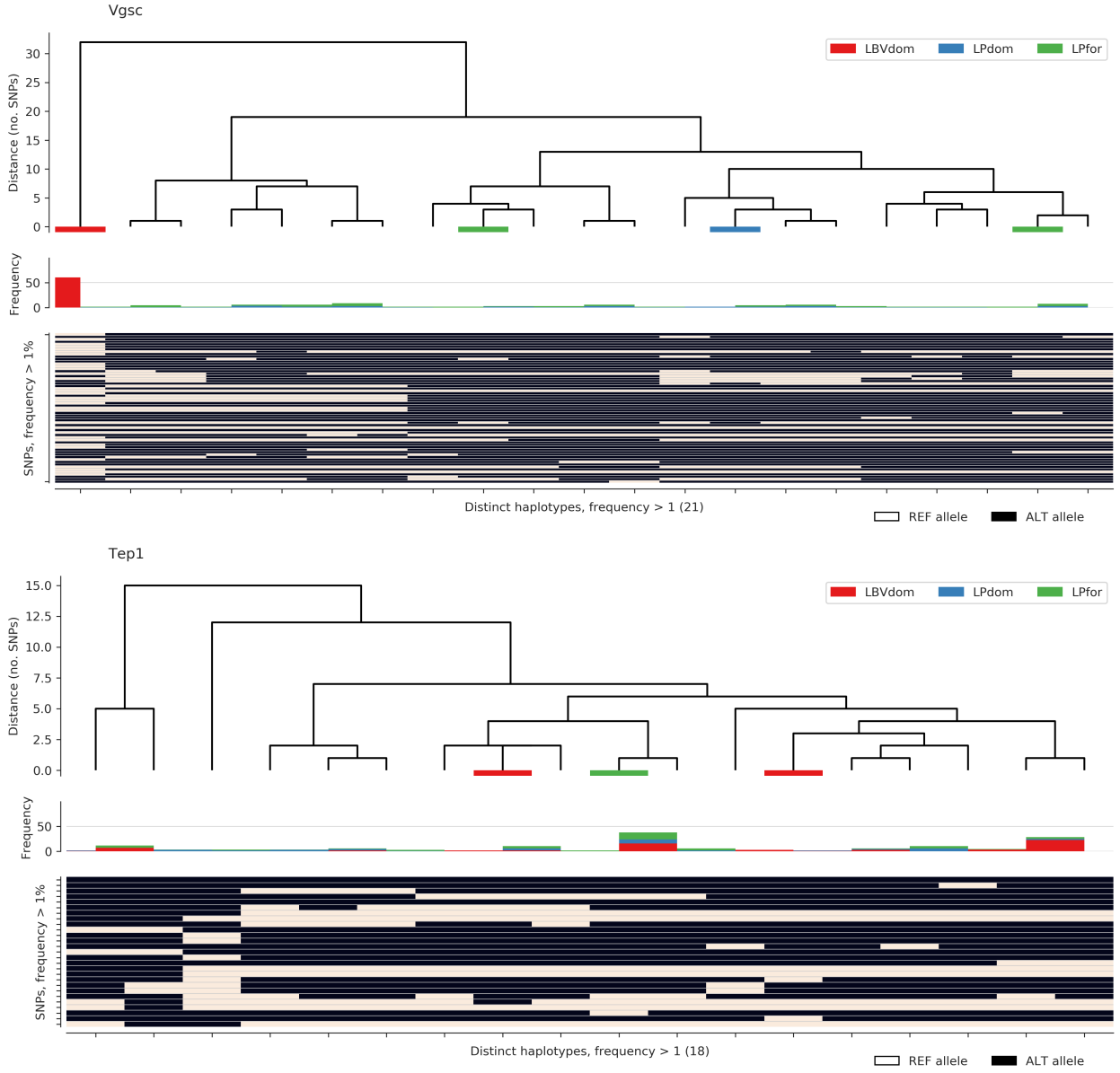
```
fst_scan.py --zarr wilding.unifiedGenotyper.cov14x.passQC.phased.zarr/ --pop1 LPdom.ind --pop2 LPfor.in
fst_scan.py --zarr wilding.unifiedGenotyper.cov14x.passQC.phased.zarr/ --pop1 LPdom.ind --pop2 LBVwil.i
fst_scan.py --zarr wilding.unifiedGenotyper.cov14x.passQC.phased.zarr/ --pop1 LPfor.ind --pop2 LBVwil.i
```

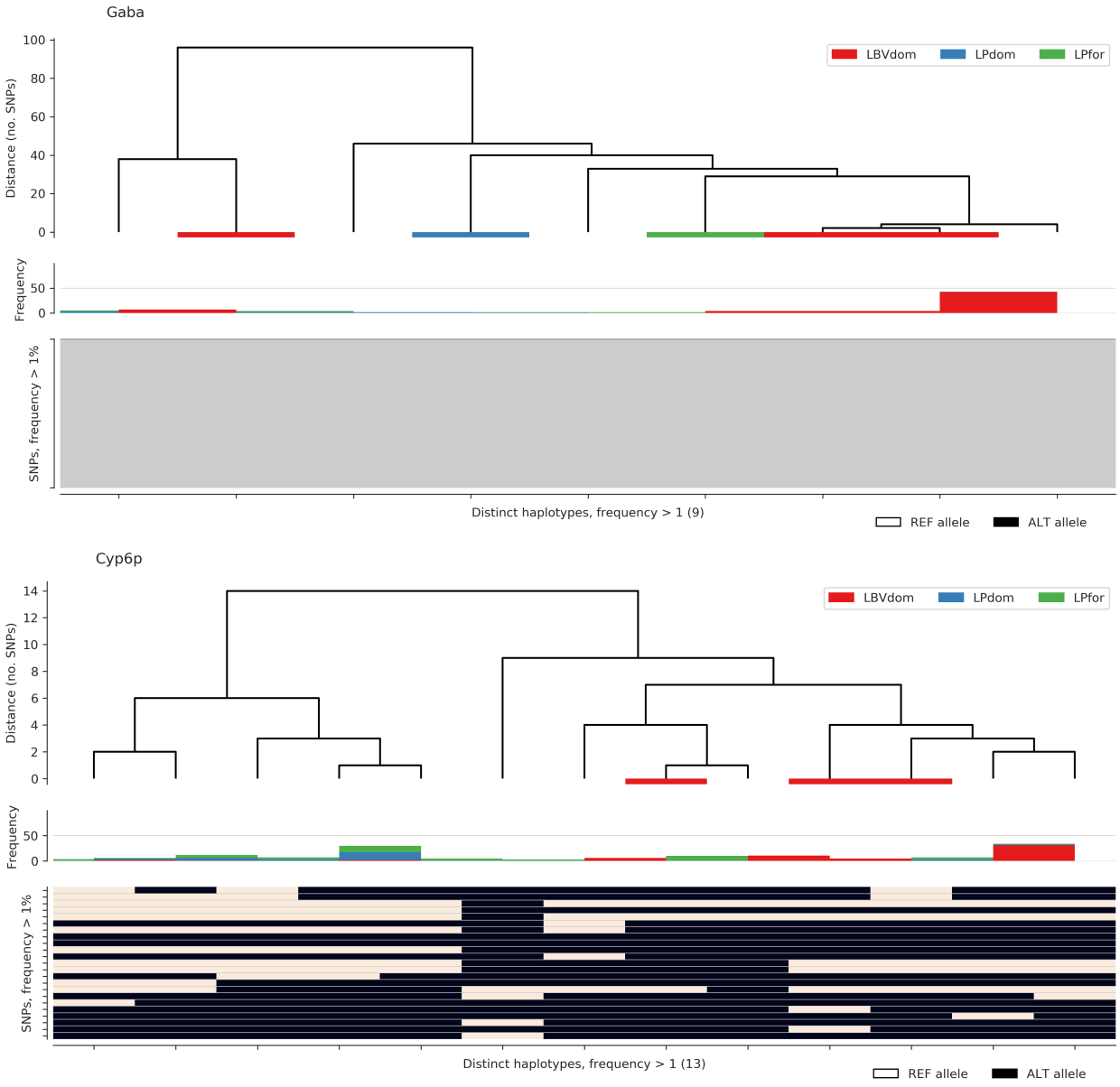
```
getPBS_scan.py --pop1 LPdom.ind --pop2 LPfor.ind --pop3 LBVwil.ind --zarr wilding.unifiedGenotyper.cov1
getPBS_scan.py --pop1 LBVwil.ind --pop2 LPdom.ind --pop3 LPfor.ind --zarr wilding.unifiedGenotyper.cov1
getPBS_scan.py --pop1 LBVwil.ind --pop2 LPfor.ind --pop3 LPdom.ind --zarr wilding.unifiedGenotyper.cov1
```

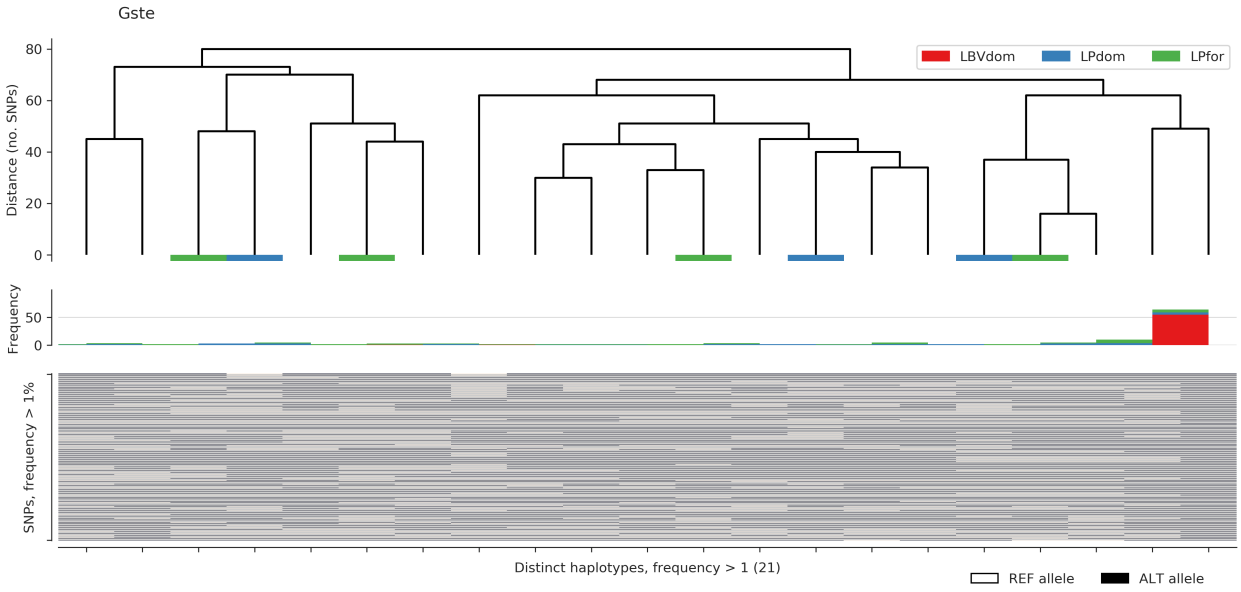



LPdom_LBVwil
LPfor_LBVwil
LPdom_LPfor
2R
2L
3R
3L
X
Cyp6p
Vgsc
Gaba
Gste
Tep1
Cyp9k1
fst
dxy
value
pos/1e+06
pbs
comp_pop
LPdom_LBVwil_LPfor
LPdom_LPfor_LBVwil
LBVwil_LPdom_LPfor
LBVwil_LPfor_LPdom

Haplotype clustering analysis




Gaba
LBVdom
LPdom
LPfor
Distance (no. SNPs)
100
80
60
40
20
0
Frequency
50
0
SNPs, frequency > 1%
Distinct haplotypes, frequency > 1 (9)
REF allele
ALT allele
Cyp6p
LBVdom
LPdom
LPfor
Distance (no. SNPs)
14
12
10
8
6
4
2
0
Frequency
50
0
SNPs, frequency > 1%
Distinct haplotypes, frequency > 1 (13)
REF allele
ALT allele


Gste
LBVdom
LPdom
LPfor
Distance (no. SNPs)
80
60
40
20
0
Frequency
50
0
SNPs, frequency > 1%
Distinct haplotypes, frequency > 1 (21)
REF allele
ALT allele