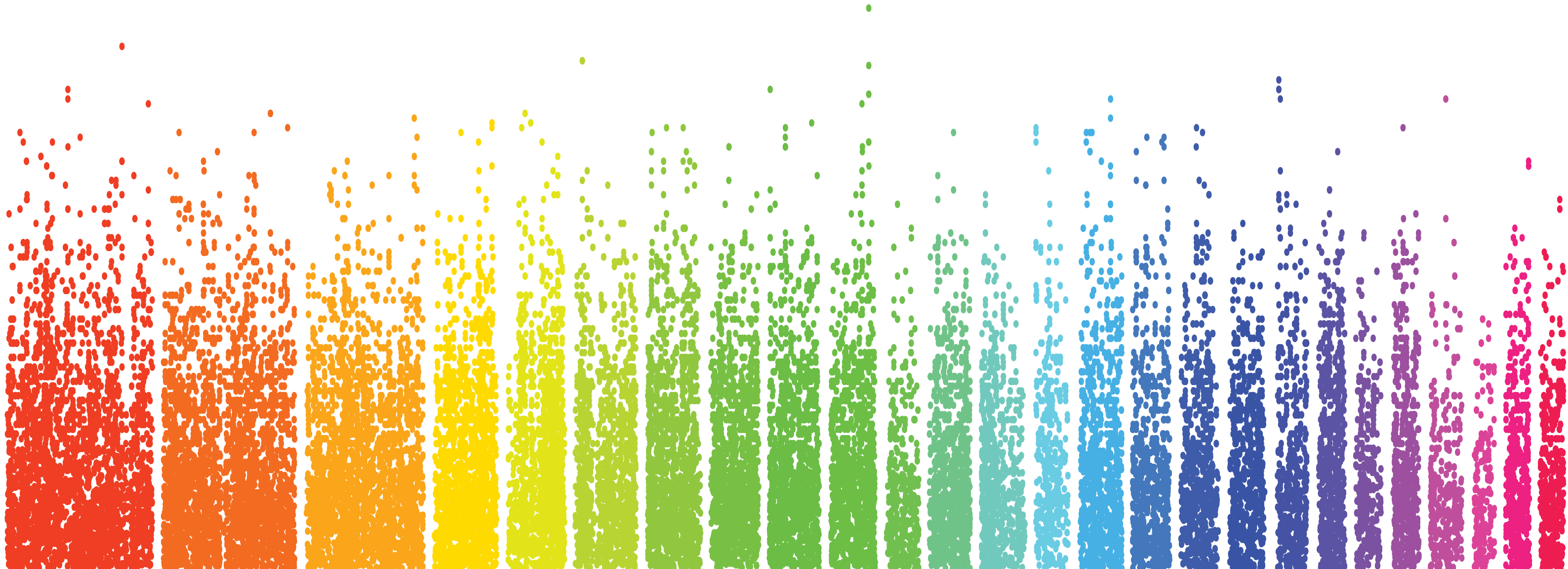
GENOVIS

Version 1.0.5

Θ(π)



Contents

1. Introduction

GENOVIS is a command-line Python-based package for visualizing different population genomics plots, with its first version released in August 2025. This package includes six modules: *mapden* (for visualizing SNP densities using PLINK map files), *relmap* (for visualizing heatmap relationship matrices), *pca3d* (for visualizing principal component analysis results in three dimensions), *admix* (for visualizing admixture analysis), *rohpainter* (for visualizing runs of homozygote regions), and *manplot* (for visualizing manhattan plots). We used different Python libraries, including matplotlib [1] (version 3.10.3), argparse (version 1.1), numpy [2] (version 2.2.5), pandas [3] (version 2.2.3), and seaborn [4] (version 0.13.2), to develop GENOVIS in the Python environment (version 3.11.9).

2. Installation

2.1. pip

GENOVIS is implemented entirely in Python; therefore, a Python installation (version ≥ 3.9) is required on the host machine. Once Python is available, GENOVIS can be installed directly from the Python Package Index (PyPI) using the standard pip install command via the Linux/Mac terminal or the Windows Command Prompt:

```
pip install genovis
```

2.2. Github

Alternatively, users may install GENOVIS directly from the GitHub repository. To do this, the source code can be cloned and installed in editable mode using pip as follows:

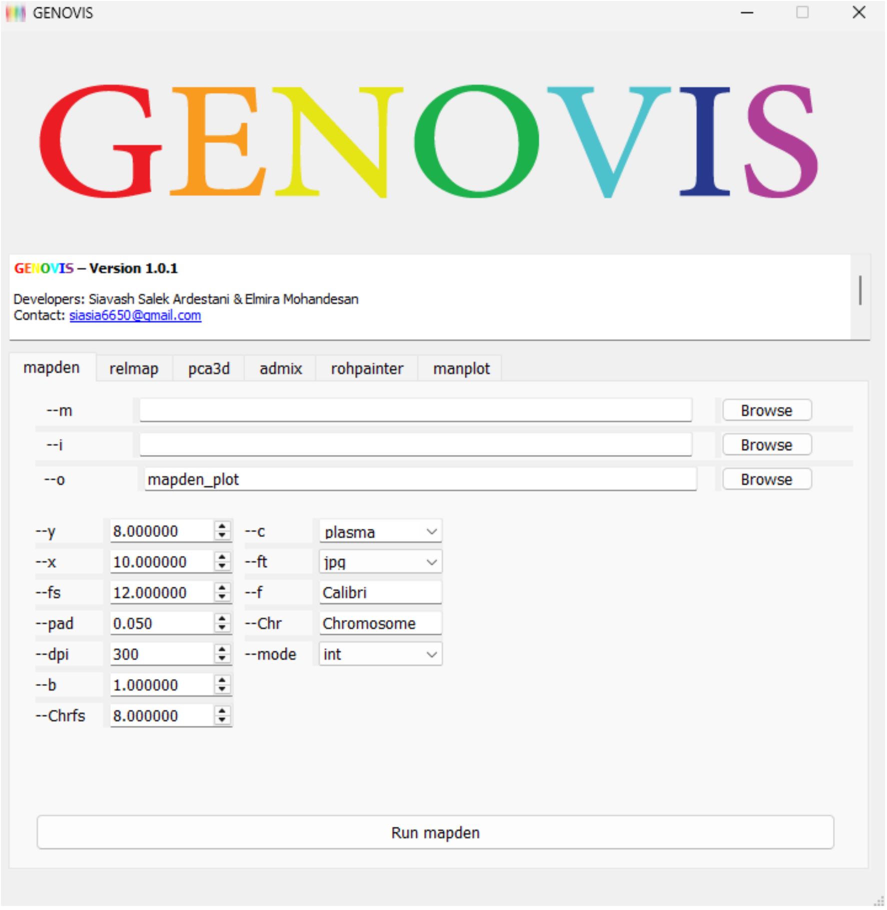
```
git clone https://github.com/Siavash-cloud/GENOVIS.git
cd GENOVIS
pip install -e .
```

3. Running GENOVIS

GENOVIS can be executed in two ways: 1. Graphical user interface (GUI) 2. Command-line interface (CLI). To launch the GENOVIS (GUI), users can simply run the following command in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
genovis-gui
```

Then, GENOVIS window will be opened, as shown in Figure 1.



GENOVIS – Version 1.0.1

Developers: Siavash Salek Ardestani & Elmira Mohandesan
Contact: siasia6650@gmail.com

--m	
--i	
--o	mapden_plot

--y	8.000000	--c	plasma
--x	10.000000	--ft	jpg
--fs	12.000000	--f	Calibri
--pad	0.050	--Chr	Chromosome
--dpi	300	--mode	int
--b	1.000000		
--Chrfs	8.000000		

Figure 1. GENOVIS interface launched via the "genovis-gui" command in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell).

To run GENOVIS in CLI mode, the tool can be invoked with:

```
genovis
```

in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell).

4. How to use GENOVIS (CLI)

4.1. *mapden*

mapden is a module for the visualization of single-nucleotide polymorphism (SNP) density by dividing each chromosome into non-overlapping, fixed-size bins. Options of this module are described in Table 1.

Table 1. Explanations of optional and required flags in *mapden* module.

#	Flag	Explanation	Optional/Required	Default values
1.	--m	Path to map file (Plink format: 1st column: Chromosome, 2nd column: SNPID, 3rd column: Genetic distance (morgans), 4th column: Base-pair position)	Required	-
2.	--i	Path to genome index file (1st column: chromosome, 2nd column: size)	Required	-
3.	--mode	Output mode: for this option, there are two choices to show plot interactively (--mode int) or directly save as a solid figure (--mode solid). By applying "--mode int", users can interactively change angles and consequently can save figure.	Optional	int
4.	--o	Path to output file	Optional	Mapden
5.	--x	Horizontal size of figure	Optional	8
6.	--y	Vertical size of figure	Optional	5
7.	--fs	Font size (except for chromosome labels)	Optional	12
8.	--ft	Format type of figure (PDF, JPG, etc.)	Optional	png
9.	--c	Colormaps	Optional	plasma
10.	--dpi	Dots per inch	Optional	300
11.	--f	Font type	Optional	Calibri
12.	--pad	Distance between legend and main figure	Optional	0.05
13.	--Chr	Chromosomal prefix(chr, chromosome, contig, or whatever the user wants)	Optional	Chr
14.	--b	Bin size (Mbp, default=1 Mbp)	Optional	1
15.	--Chrfs	Font size of chromosome labels	Optional	10

* The dataset we used in the "**Short instructions**" sections of this user manual is raw genotypes of different sheep breeds generated by Kijas et al. (2012) [5] and downloadable from (http://widde.toulouse.inra.fr). This dataset is also available at https://github.com/Siavash-cloud/GENOVIS/tree/main/example_data.

Short instructions to generate a map file from other common formats:

Users can generate a map (*.map) file from different common genotype formats, such as VCF and PLINK binary files, using the PLINK software in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
# Converting PLINK binary files to *.ped *.map
./plink --bfile raw_sheep_data --recode ped --sheep --out example_mapden
```

After running one of the above-mentioned commands, "*example_mapden.ped*" and "*example_mapden.map*" will be generated. Then, *mapden* module can use "example_mapden.map" file directly.

The input files (map and genome index) must be without headers and similar to the following:

1. Map file (--m, PLINK format) [6]: 1st column is chromosome number, 2nd column is SNPID, 3rd column is genetic distance (morgans) and, 4th column is base-pair position.

```
1    SNP1     0    81978
1    SNP2     0    315497
1    SNP3     0    357652
.    .        .    .
.    .        .    .
.    .        .    .
26   SNPn     0    44004281
```

2. Genome index (--i): 1st column is chromosome number and, 2nd column is size of chromosome.

```
1    275612895
2    248993846
.    .
.    .
.    .
26   44077779
```

Example 1. Command-line-based example for *mapden* module:

```
genovis mapden --m example_mapden.map --fs 12 --f "Times New Roman" --o Figure_mapden --ft
jpg --Chr "Chr" --i sheep_genome_index.txt --pad 0.1 --dpi 300 --mode solid
```

Example 1 produces the output file “Figure_mapden.jpg”, which is shown in Figure 2.

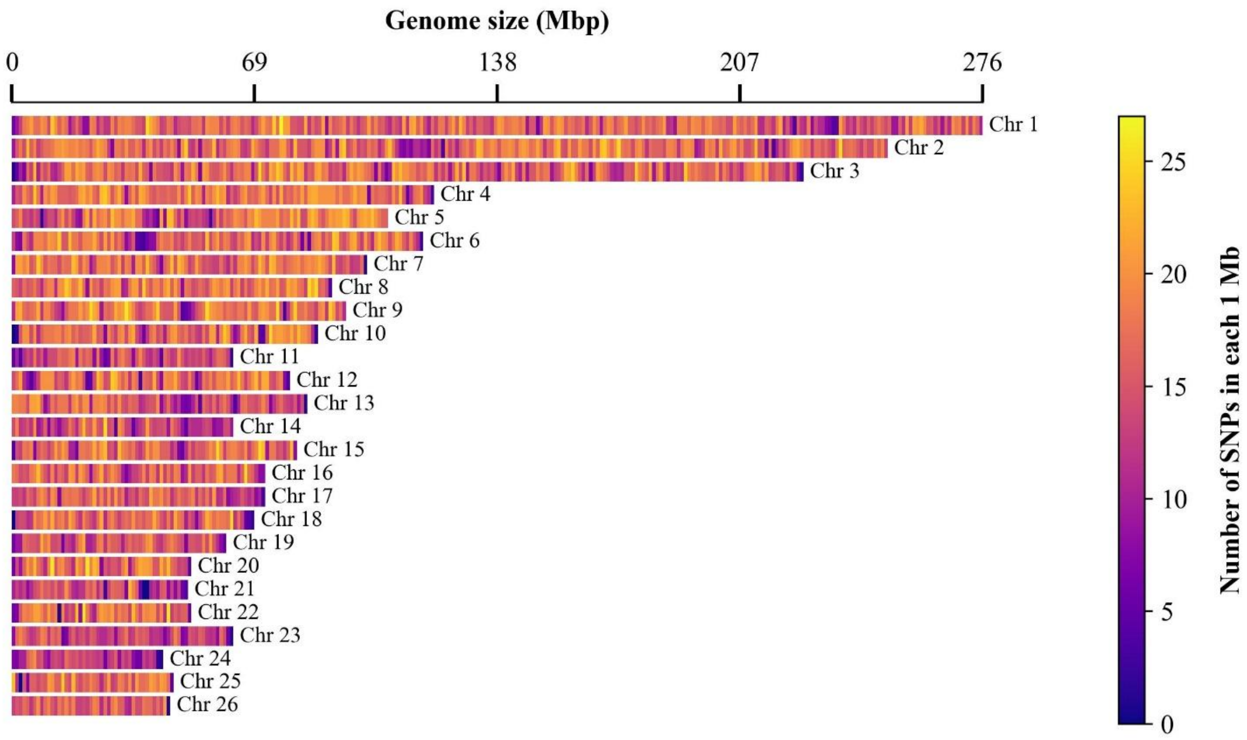


Figure 2. The SNP density plot produced by GENOVIS *mapden* module (`genovis mapden --m example_mapden.map --fs 12 --f "Times New Roman" --o Figure_mapden --ft jpg --Chr "Chr" --i sheep_genome_index.txt --pad 0.1 --dpi 300 --mode solid`)

4.2. *relmap*

The *relmap* module visualizes heatmap relationship matrix. This module can handle two types of relationship matrices (columnar and matrix+index). Moreover, the average of relationships between populations can be generated by applying "--av true" option (Table 2). Input of this module can be in two different types, one of which must be used in this module by specifying "--rf col/mat". If users want to use a columnar format type ("--rf col") of relationship matrix, a dataframe (without headers) is required, similar to the:

```
BAL    BAL_1          BAL    BAL_2          0.759347
BAL    BAL_1          BAL    BAL_3          0.761083
.      .              .      .
.      .              .      .
.      .              .      .
MAA    MAA_17081      MAA    MAA_64302      0.756056
```

If users want to use a matrix+index format in *relmap* module as input, the "--rf mat" must be set. Additionally, a relationship matrix (without headers) similar to the:

```
1           0.759347    0.761083    .           .           0.759347
0.759347    1           .           .           .           0.761083
0.761083    .           .           .           .           .
.           .           .           .           .           .
.           .           .           .           .           0.756056
0.700250    0.852551    0.751151    .           0.756056    1
```

and, an index file (without headers) similar to the one below is needed:

```
BAL    BAL_1
BAL    BAL_2
.      .
.      .
.      .
MAA    MAA_17081
```

The first and second columns in the index file are the population label and individual ID, respectively. A genomic relationship matrix can be constructed using PLINK2 [7] (https://www.cog-genomics.org/plink/1.9/distance).

Table 2. Explanations of optional and required flags in *relmap* module.

#	Flag	Explanation	Optional/Required	Default values
1.	--relmap	Path to relationship matrix file	Required	-
2.	--rf	Format of relationship matrix (--rf col/mat)	Required	Col
3.	--matindex	Index of relationship matrix (it is required if you are using --rf mat, matindex: A dataframe including two columns: Population labels and individual IDs)	Required if user sets “--rf mat”	-
4.	--mode	Output mode: for this option, there are two choices to show plot interactively (--mode int) or directly save as a solid figure (--mode solid). By applying “--mode int”, users can interactively change angles and consequently can save figure.	Optional	int
5.	--mask	Mask diagonal elements or not (true/false)	Optional	false
6.	--a	Annotation of relationship values in heatmap plot (true/false)	Optional	false
7.	--afs	Font size of annotations	Optional	6
8.	--av	Output for averages of relationships among populations (true/false)	Optional	false
9.	--sl	Show individual labels (true/false)	Optional	false
10.	--lc	Color of separator lines (black)	Optional	black
11.	--lws	Size of separator lines	Optional	0.45
12.	--pfs	Font size of population labels	Optional	12
13.	--t	Title of legend	Optional	Relationship
14.	--o	Path to output file	Optional	-
15.	--x	Horizontal size of figure	Optional	14
16.	--y	Vertical size of figure	Optional	10
17.	--xyfs	Font size of individual labels	Optional	1
18.	--ft	Format type of figure (PDF, JPG, etc.)	Optional	jpg
19.	--c	Colormaps	Optional	YlOrRd
20.	--dpi	Dots per inch	Optional	300
21.	--f	Font type	Optional	Calibri

Short instructions to generate a genomic relationship matrix:

To generate a genomic relationship matrix from a raw genotype dataset (PLINK binary file), users can run the following commands for quality control steps (using PLINK software):

```
# Step 1: filtering based on minor allele frequency (maf < 0.01)
./plink --bfile raw_sheep_data --maf 0.01 --make-bed --sheep --out maf
# Step 2: filtering based on missing genotypes (geno > than 0.1)
./plink --bfile maf --sheep --geno 0.1 --make-bed --out maf_geno
# Step 3: filtering based on Hardy-Weinberg P-values (hwe >than 0.000001)
./plink --bfile maf_geno --make-bed --sheep --hwe 0.000001 --out maf_geno_hwe
# Step 4: filtering based on missing genotype rates in individuals (mind > 0.1)
./plink --bfile maf_geno_hwe --make-bed --sheep --mind 0.1 --out maf_geno_hwe_mind
```

and then:

```
# Step 5: generating a relationship matrix using PLINK 2
./plink2 --bfile maf_geno_hwe_mind --make-rel square --out example_relmap_matrix
```

in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell). Finally, two files, including: 1. relationship matrix, 2. labels (IDs and populations), will be generated, which can be used in *relmap* module directly.

Example 2. Command-line-based example for *relmap* module:

```
genovis relmap --rf mat --relfile example_relmap_matrix.rel --matindex
example_relmap_matrix.rel.id --t "Relationship" --c viridis --sl false --x 7 --y 6 --xyfs 10
--pfs 6 --f "Times New Roman" --dpi 300 --a false --o relmap --mask false --av true --mode
solid
```

The produced figure by the Example 2 command is shown in Figure 3.

Hints for *relmap*:

1. Users can increase/decrease distances between population labels and the axes (x and y) by increasing/decreasing "--xyfs" value when "--sl false" (i.e., DO NOT SHOW INDIVIDUAL LABELS).

2. We do not suggest "--a true" (i.e., annotate relationship values) when users have a big dataset/matrix.

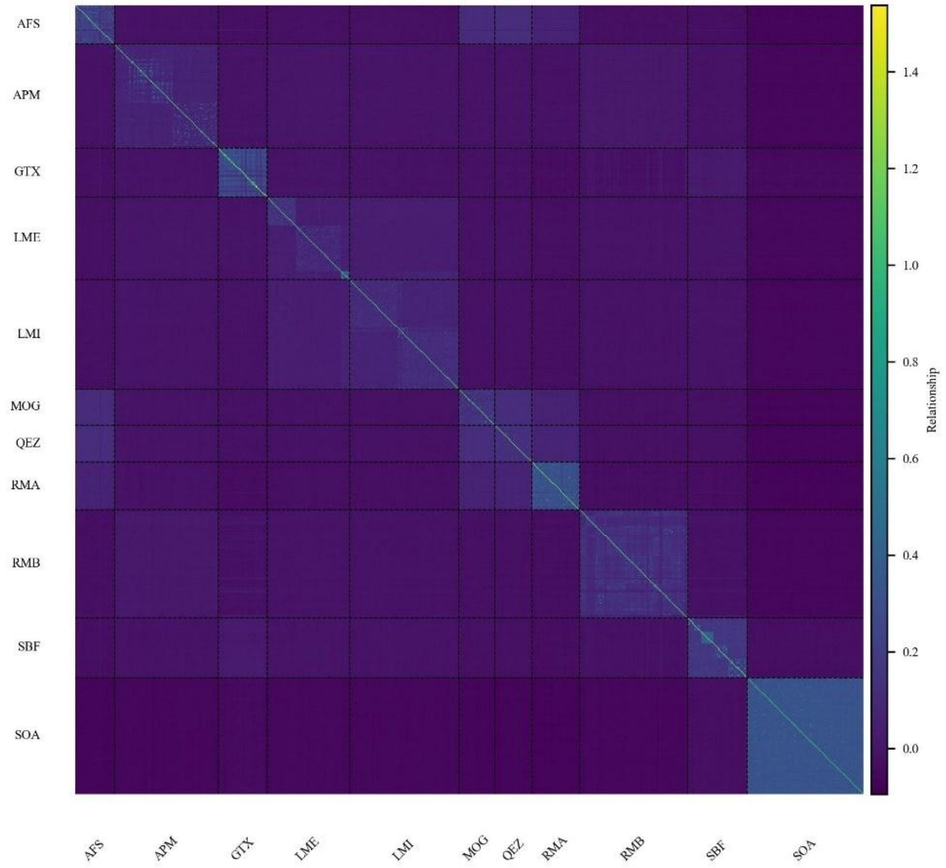


Figure 3. Generated relationship heatmap plot in the GENOVIS *relmap* (`genovis relmap --rf mat --relfile example_relmap_matrix.txt --matindex example_relmap_matindex.txt --t "Relationship" --c viridis --sl false --x 7 --y 6 --xyfs 10 --pfs 6 --f "Times New Roman" --dpi 300 --a false --o relmap --mask false --av true --mode solid`)

4.3. *pca3d*

The *pca3d* module is a tool for visualizing principal component analysis (PCA) in three dimensions. Users can use this module in an interactive mode (--mode "int") to easily change azimuth and elevation angles (Table 3). This module is adapted with PLINK software outputs for PCA ("--pca"). Therefore, users can directly use *.eigenvec and *.eigenval files from PLINK.

Structure of *.eigenvec file must be the same as below:

```
AFS    AFS_1       0.0213926        -0.0664315        0.00290785        .  .  .  -0.0251144
AFS    AFS_2       0.0215556        -0.070594         0.00253091        .  .  .  -0.041725
.      .           .                .                 .                 .  .  .  .
.      .           .                .                 .                 .  .  .  .
.      .           .                .                 .                 .  .  .  .
SOA    SOA_6386    -0.0870463       -0.00799303       -0.00622813       .  .  .  -0.00104679
```

In this dataframe, the first column is family ID or population label, the second column is the ID of individuals, and from the third column to the last one are PC_1 to PC_n. Structure of *.eigenval file must be the same as below in a column (explained variance by PC_1 to PC_n):

```
44.1208
22.8378
.
.
.
2.31125
```

Table 3. Explanations of optional and required flags in *pca3d* module.

#	Flag	Explanation	Optional/Required	Default values
1.	--evec	Path to eigenvec file	Required	-
2.	--eval	Path to eigenval file	Required	-
3.	--mode	Output mode: for this option, there are two choices to show plot interactively (--mode int) or directly save as a solid figure (--mode solid). By applying "--mode int", users can interactively change angles and consequently can save figure.	Optional	int
4.	--azim	Azimuth angle	Optional	65
5.	--elev	Elevation angle	Optional	20
6.	--fp	i^{th} PC on x axis (default=1, which means PC_1)	Optional	1
7.	--sp	j^{th} PC on y axis (default=2, which means PC_2)	Optional	2

8.	--tp	k^{th} PC on z axis (default=3, which means PC_3)	Optional	3
9.	--s	Size of scatter points	Optional	40
10.	--o	Path to output file	Optional	3D_PCA
11.	--x	Horizontal size of figure	Optional	10
12.	--y	Vertical size of figure	Optional	5
13.	--ft	Format type of figure (PDF, JPG, etc.)	Optional	jpg
14.	--c	Colormaps	Optional	brg
15.	--dpi	Dots per inch	Optional	300
16.	--f	Font type	Optional	Calibri
17.	--dim	Plot dimension	Optional	3d

Short instructions for PCA:

After running the following commands for quality control steps (using PLINK software) in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
# Step 1: filtering based on minor allele frequency (maf < 0.01)
./plink --bfile raw_sheep_data --maf 0.01 --make-bed --sheep --out maf
# Step 2: filtering based on missing genotypes (geno > than 0.1)
./plink --bfile maf --sheep --geno 0.1 --make-bed --out maf_geno
# Step 3: filtering based on Hardy-Weinberg P-values (hwe >than 0.000001)
./plink --bfile maf_geno --make-bed --sheep --hwe 0.000001 --out maf_geno_hwe
# Step 4: filtering based on missing genotype rates in individuals (mind > 0.1)
./plink --bfile maf_geno_hwe --make-bed --sheep --mind 0.1 --out maf_geno_hwe_mind
```

Users can run a PCA analysis using:

```
# Step 5: PCA in PLINK
./plink --bfile maf_geno_hwe_mind --pca --out example_3dPCA
```

in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell). Then, "*example_3dPCA.eigenvec*" and "*example_3PCA.eigenval*" can be used for visualizing a 2D/3D-PCA plot in *pca3d* module, directly.

Example 3. Command-line-based example for *pca3d* module (Figure 4):

```
genovis pca3d --evec example_3dPCA.eigenvec --eval example_3PCA.eigenval --fp 1 --sp 2 --tp 3

--s 70 --mode solid --f "Times New Roman" --azim 45 --elev 65 --o 3dpcaplot --ft jpg --y 8 -

-x 10 --dim 3d
```

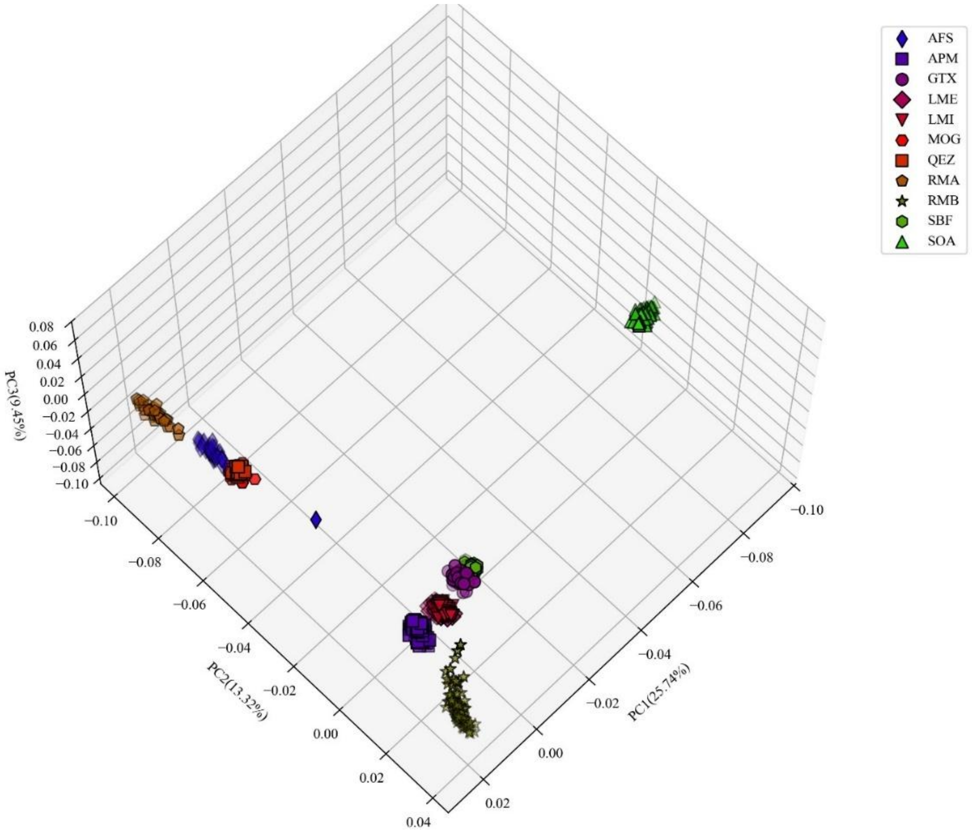


Figure 4. Generated 3D-PCA plot using GENOVIS *pca3d* module (`genovis pca3d --evec example_3dPCA.eigenvec --eval example_3PCA.eigenval --fp 1 --sp 2 --tp 3 --s 70 --mode solid --f "Times New Roman" --azim 45 --elev 65 --o 3dpcaplot --ft jpg --y 8 --x 10`)

Users can also generate 2D-PCA plots by "--dim 2d" as defined in Example 4 (Figure 5):

```
genovis pca3d --evec example_3dPCA.eigenvec --eval example_3PCA.eigenval --fp 1 --sp 2 --s 70
--mode solid --f "Times New Roman" --o 2dpcaplot --ft jpg --y 8 --x 10 --dim 2d
```

When users employ "--dim 2d", then GENOVIS only considers "--fp" (1stPC) and "--sp (2nd PC)". Therefore, there is no need to define "--tp" (3rd PC), "--elev", and "--azim" options.

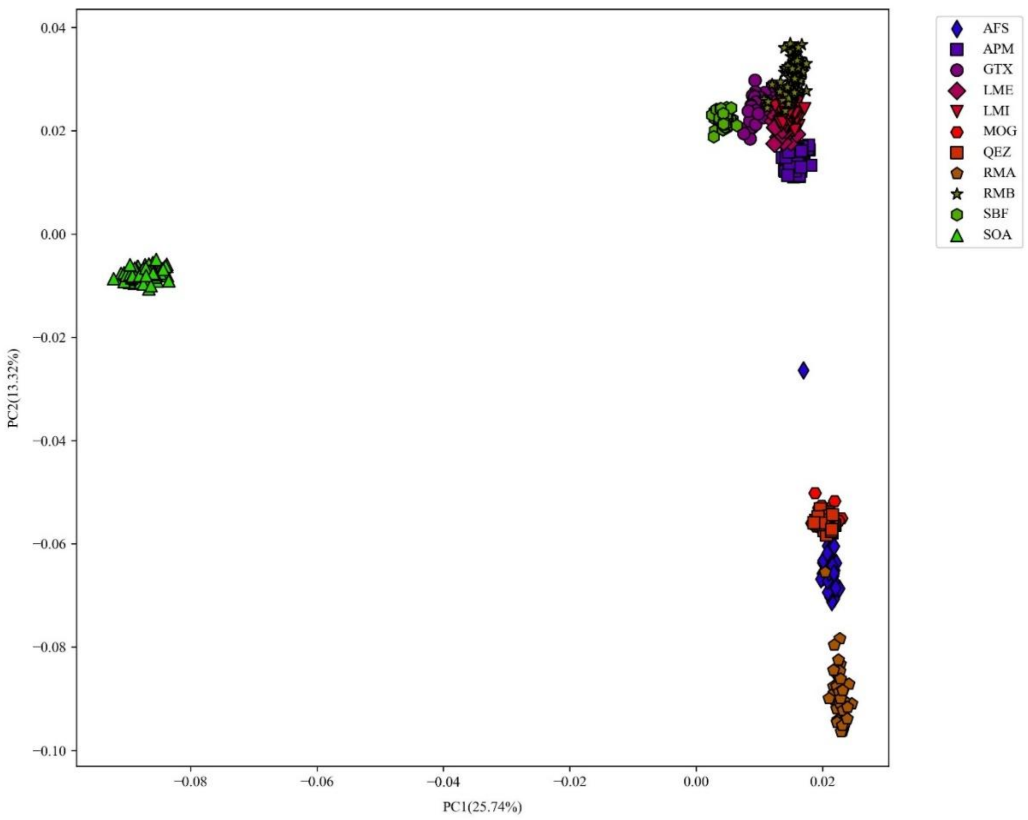


Figure 5. Generated 2D-PCA plot using GENOVIS *pca3d* module (`genovis pca3d --evec example_3dPCA.eigenvec --eval example_3PCA.eigenval --fp 1 --sp 2 --s 70 --mode solid --f "Times New Roman" --o 2dpcaplot --ft jpg --y 8 --x 10 --dim 2d`)

4.4. *admix*

Using *admix* module, users can visualize admixture proportions across individuals. The structure of the input data frame in *admix* module is described below:

```
AFS    AFS_1       0.994573     0.00001      .  .  .     0.005367
AFS    AFS_3       0.972354     0.00001      .  .  .     0.002364
.      .           .            .            .  .  .     .
.      .           .            .            .  .  .     .
.      .           .            .            .  .  .     .
SOA    SOA_6386    0.00001      0.12002      .  .  .     0.652544
```

The first and second columns are population labels and individual ID, respectively. Proportions K=1 to n are located in the third to n column. Options of *admix* module are described in Table 4.

Table 4. Explanations of optional and required flags in *admix* module.

#	Flag	Explanation	Optional/Required	Default values
1.	--d	Path to input file	Required	-
2.	--mode	Output mode: for this option, there are two choices to show plot interactively (--mode int) or directly save as a solid figure (--mode solid). By applying “--mode int”, users can interactively change angles and consequently can save figure.	Optional	int
3.	--o	Path to output file	Optional	admix_plot
4.	--sl	Show individual labels (true/false)	Optional	false
5.	--xt	Font size of individual labels	Optional	9
6.	--x	Horizontal size of figure	Optional	8.4
7.	--y	Vertical size of figure	Optional	4
8.	--lws	Size of separator lines	Optional	0.75
9.	--ft	Format type of figure (PDF, JPG, etc.)	Optional	png
10.	--c	Colormaps	Optional	hsv
11.	--dpi	Dots per inch	Optional	300
12.	--f	Font type	Optional	Calibri

Short instructions for admixture analysis:

After running the following commands for quality control steps (using PLINK software) in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
# Step 1: filtering based on minor allele frequency (maf < 0.01)
./plink --bfile raw_sheep_data --maf 0.01 --make-bed --sheep --out maf
# Step 2: filtering based on missing genotypes (geno > than 0.1)
./plink --bfile maf --sheep --geno 0.1 --make-bed --out maf_geno
# Step 3: filtering based on Hardy-Weinberg P-values (hwe >than 0.000001)
./plink --bfile maf_geno --make-bed --sheep --hwe 0.000001 --out maf_geno_hwe
# Step 4: filtering based on missing genotype rates in individuals (mind > 0.1)
./plink --bfile maf_geno_hwe --make-bed --sheep --mind 0.1 --out example_admix
```

Users can run an admixture analysis using:

```
# Step 5: admixture analysis (at K=8)
./admixture example_admix 8
```

in a Linux/Mac terminal. Then, “*example_admix.8.Q*” and “*example_admix.8.P*” files will be generated. The “*example_admix.8.Q*” and “*example_admix.fam*” (generated file from Step 4) can be used for producing an input file for *admix* module. The first two columns of “*example_admix.fam*” usually contain population IDs and individual IDs; by concatenating these columns into the “*example_admix.8.Q*” file (using Microsoft Excel or the “cbind” command in R), a compatible input for the *admix* module can be generated.

Example 5 Command-line-based example for *admix* module (Figure 6):

```
genovis admix --d example_admix.txt --c tab20 --fs 8 --lws 0.2 --x 8 --y 2 --o admix --mode solid --f "Times New Roman"
```

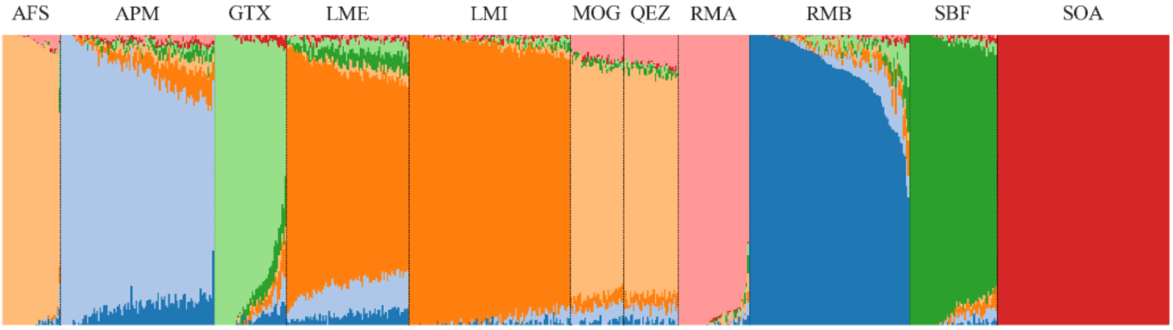


Figure 6. Generated admixture (K=7) plot using GENOVIS *admix* module (`genovis admix --d example_admix.txt --c tab20 --fs 8 --lws 0.2 --x 8 --y 2 --o admix --mode solid --f "Times New Roman"`)

4.5. *rohpainter*

By using *rohpainter* module, users can visualize runs of homozygote regions (ROH) across the genome. However, this module can also be used for visualizing the copy number of variant (CNV) regions. Inputs to this module include:

1. A dataframe including population label (1st column), individual ID (2nd column), chromosome (3rd column), start (4th column), and end (5th column) positions, same as below.

```
AFS    AFS_1        1     66312038         67889219
AFS    AFS_2        1     110928043        111691662
.      .            .     .
.      .            .     .
QEZ    QEZ_35       25    37661969         38926696
QEZ    QEZ_35       27    40620729         41561995
```

2. A genome index file, the same as what we mentioned for *mapden* module. The *rohpainter* flags are described in Table 5.

Table 5. Explanations of optional and required flags in *rohpainter* module.

#	Flag	Explanation	Optional/Required	Default values
1.	--d	Path to input file (1st column:population label, 2nd: Individual ID, 3rd: chromosome, 4th: start position, 5th: end position)	Required	-
2.	--i	Genome index (1st column:chromosome, 2nd: size)	Required	-
3.	--mode	Output mode: for this option, there are two choices to show plot interactively (--mode int) or directly save as a solid figure (--mode solid). By applying "--mode int", users can interactively change angles and consequently can save figure.	Optional	int
4.	--Chr	Chromosomal prefix(chr, chromosome, contig, or whatever the user wants)	Optional	Chromosome
5.	--xt	Font size of xticks	Optional	9

6.	--yt	Font size of yticks	Optional	9
7.	--t	Threshold for identifying common intervals(e.g., 0.8). Please use “--t false”, if you do not want to pinpoint common intervals.	Optional	false
8.	--tc	Threshold line color	Optional	red
9.	--tw	Threshold line width	Optional	0.75
10.	--sl	Show individual labels (true/false)	Optional	true
11.	--o	Path to output file	Optional	out
12.	--x	Horizontal size of figure	Optional	10
13.	--y	Vertical size of figure	Optional	4
14.	--fs	Font size	Optional	12
15.	--ft	Format type of figure (PDF, JPG, etc.)	Optional	jpg
16.	--c	Colormaps	Optional	Paired
17.	--dpi	Dots per inch	Optional	300
18.	--f	Font type	Optional	Calibri

Short instructions for ROH analysis:

After running the following commands for quality control steps (using PLINK software) in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
# Step 1: filtering based on minor allele frequency (maf < 0.01)
./plink --bfile raw_sheep_data --maf 0.01 --make-bed --sheep --out maf
# Step 2: filtering based on missing genotypes (geno > than 0.1)
./plink --bfile maf --sheep --geno 0.1 --make-bed --out maf_geno
# Step 3: filtering based on Hardy-Weinberg P-values (hwe >than 0.000001)
./plink --bfile maf_geno --make-bed --sheep --hwe 0.000001 --out maf_geno_hwe
# Step 4: filtering based on missing genotype rates in individuals (mind > 0.1)
./plink --bfile maf_geno_hwe --recode ped --sheep --mind 0.1 --out example_roh
```

Users can run a ROH analysis using the generated files from the last step of quality control (“*example_roh.ped*” and “*example_roh.map*”) and detectRUNS [8] package in R [9]:

```
# Step 5 (in R): ROH analysis using R and the detectRUNS package
#install.packages("detectRUNS")# detectRUNS installation
library(detectRUNS)
ROH<-
slidingRUNS.run("example_roh.ped","example_roh.map",minSNP=15,minLengthBps=500000,maxGap=10^
6,minDensity=1/100,maxMissWindow=1,maxOppWindow=1)
ROH<-ROH[,c(1,2,3,5,6)]
#Extraction of three Persian breeds to plot a subset of this dataframe
ROH_persian_breeds=ROH[ROH$group %in% c("QEZ","AFS","MOG"),]
write.table(ROH_persian_breeds,"example_rohpainter.txt",col.names=FALSE,row.names=FALSE,quot
e=FALSE)
```

The above-mentioned R script generates “*example_rohpainter.txt*” that can be used in *rohpainter* module, directly.

Example 6. Command-line-based example for *rohpainter* module (Figure 7):

```
genovis rohpainter --d example_rohpainter.txt --i sheep_genome_index.txt --fs 12 --t 0.4 --
ft png --o ROH_plot --y 6 --x 10 --yt 4 --f "Times New Roman" --tc "red" --tw 0.5 --sl true
--mode solid --c prism
```

After executing the abovementioned command, ROH plots files (according to the number of chromosomes) will be saved at the provided directory.

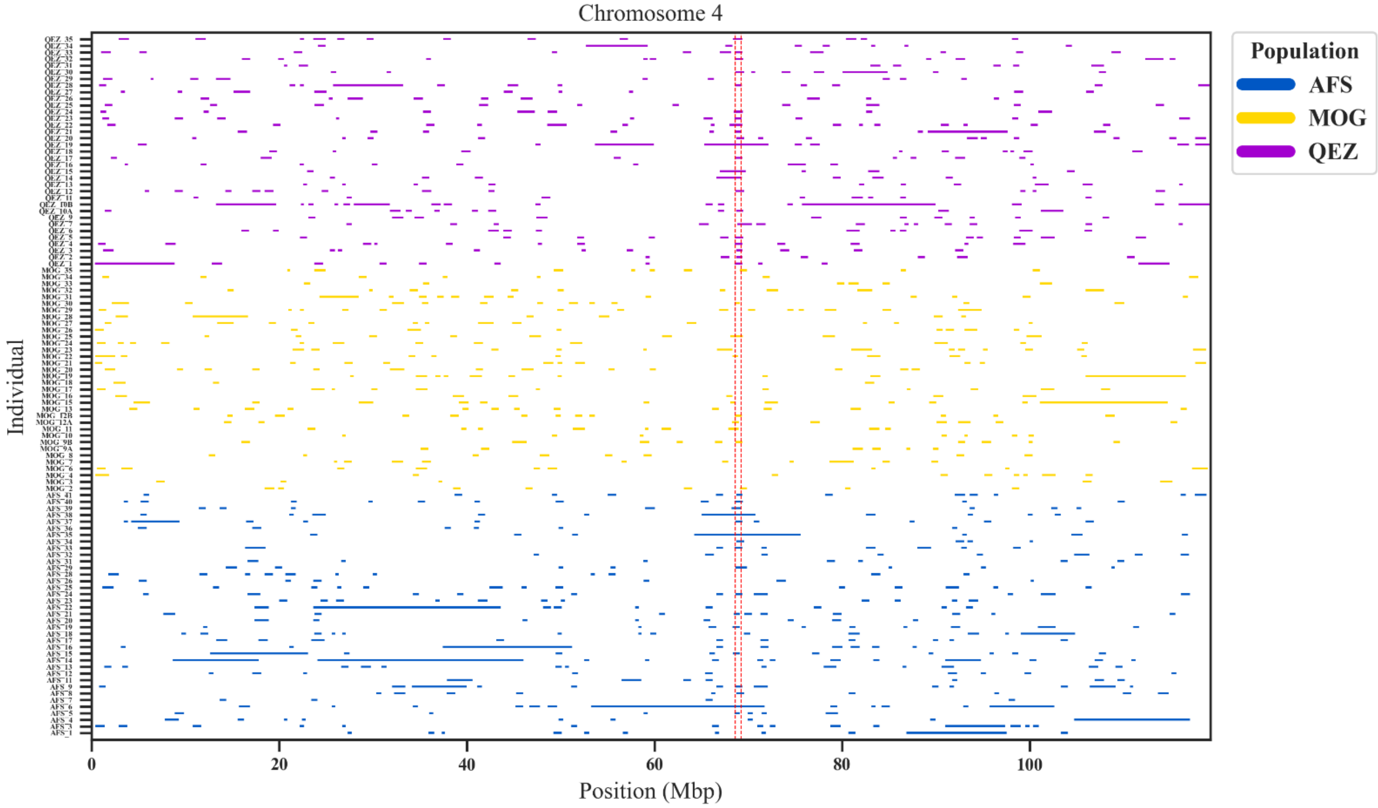


Figure 7. Generated ROH distribution plot (for chromosome 13) using GENOVIS *rohpainter* module (genovis rohpainter --d example_rohpainter.txt --i sheep_genome_index.txt --fs 12 --t 0.4 --ft png --o ROH_plot --y 6 --x 10 --yt 4 --f "Times New Roman" --tc "red" --tw 0.5 --sl true --mode solid --c prism)

Hints for *rohpainter*:

1. Users should choose a suitable value for "--y" regarding the number of individuals, as the width of lines is related to the number of individuals and the y-axis size ("--y").
2. Select a suitable font size for the y-axis ("--yt") to prevent labels on the y-axis from overlapping. This also depends on the number of individuals.

4.6. *manplot*

Using the *manplot* module, users can visualize a Manhattan plot to show values (e.g., GWAS-P-values, allelic substitution effect, nucleotide diversity, and selection pressures) for SNPs or different genomic regions across the genome. For plotting Manhattan plots, users need a data frame like:

```
1      81978       1.33E-06
1      315497      4.45E-06
1      357652      2.03E-06
.      .           .
.      .           .
.      .           .
26     44004281    1.91E-06
```

This data frame includes chromosome number (1st column), bp-position (2nd column), and values (3rd column - e.g, GWAS-P-values, allelic substitution effect, nucleotide diversity, or selection pressures). The *manplot* flags (optional/required) are explained in Table 6.

Table 6. Explanations of optional and required flags in *manplot* module.

#	Flag	Explanation	Optional/Required	Default values
1.	--d	Path to data frame	Required	-
2.	--nc	Number of colors	Optional	10

3.	--mode	Output mode: for this option, there are two choices to show plot interactively (--mode int) or directly save as a solid figure (--mode solid). By applying "--mode int", users can interactively change angles and consequently can save figure.	Optional	int
4.	--ylab	Label of y-axis	Optional	Value
5.	--xlab	Label of x-axis	Optional	Chromosome
6.	--a	The alpha blending value, between 0 (transparent) and 1 (opaque)	Optional	1
7.	--s	Scatter size	Optional	0.5
8.	--xt	Font size of xticks	Optional	9
9.	--yt	Font size of yticks	Optional	9
10.	--sug1	Suggestive line 1	Optional	-
11.	--sug2	Suggestive line 2	Optional	-
12.	--sug1lw	Width size of suggestive line 1	Optional	-
13.	--sug2lw	Width size of suggestive line 2	Optional	-
14.	--sug1c	Color of suggestive line 1	Optional	blue
15.	--sug2c	Color of suggestive line 2	Optional	red
16.	--o	Path to output	Optional	manplot_out
17.	--x	Horizontal size of figure	Optional	8.4
18.	--y	Vertical size of figure	Optional	4
19.	--fs	Font size	Optional	14
20.	--ft	Format type of figure (PDF, JPG, etc.)	Optional	png
21.	--c	Colormaps	Optional	plasma
22.	--dpi	Dots per inch	Optional	300
23.	--f	Font type	Optional	Calibri

Short instructions for F_{st} calculation:

After running the following commands for quality control steps (using PLINK software) in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
# Step 1: filtering based on minor allele frequency (maf < 0.01)
./plink --bfile raw_sheep_data --maf 0.01 --make-bed --sheep --out maf
# Step 2: filtering based on missing genotypes (geno > than 0.1)
./plink --bfile maf --sheep --geno 0.1 --make-bed --out maf_geno
# Step 3: filtering based on Hardy-Weinberg P-values (hwe >than 0.000001)
./plink --bfile maf_geno --make-bed --sheep --hwe 0.000001 --out maf_geno_hwe
# Step 4: filtering based on missing genotype rates in individuals (mind > 0.1)
./plink --bfile maf_geno_hwe --make-bed --sheep --mind 0.1 --out example_fst
```

To calculate Fst values (using PLINK) between two populations of Lacaune sheep (LMI: Lacaune sheep that bred for milk production and LME: Lacaune sheep that bred for meat production), we prepared a list ("*groups_Fst.txt*") including population labels (1st column), individual ID (2nd column), and production group (3rd column). Then, Fst values between populations can be calculated by running the following command in a Linux/Mac terminal or the Windows Command Prompt (or PowerShell):

```
# Step 5: Fst calculation using plink
./plink --bfile example_fst --within "groups_Fst.txt" --fst --out example_manplot
```

The "*example_manplot.fst*" file is not compatible with *manplot*. Therefore, by executing:

```
# Step 6 (in Linux/Mac terminal): Removing headers
sed 1d example_manplot.fst > No_header_fst.txt
# Step 7 (in Linux/Mac terminal): Extracting columns (1, 3, 5) are needed for manplot module
awk -F'\t' '{print $1, $3, $5}' No_header_fst.txt > example_manplot_fst.txt
```

in a Linux/Mac terminal, users can generate a compatible input for the *manplot* module.

Example 7. Command-line-based example for *manplot* module:

```
genovis manplot --d example_manplot_fst.txt --c nipy_spectral --nc 26  --ylab "F(st)" --f
"Times New Roman" --mode solid --o manplot --ft jpg --dpi 700
```

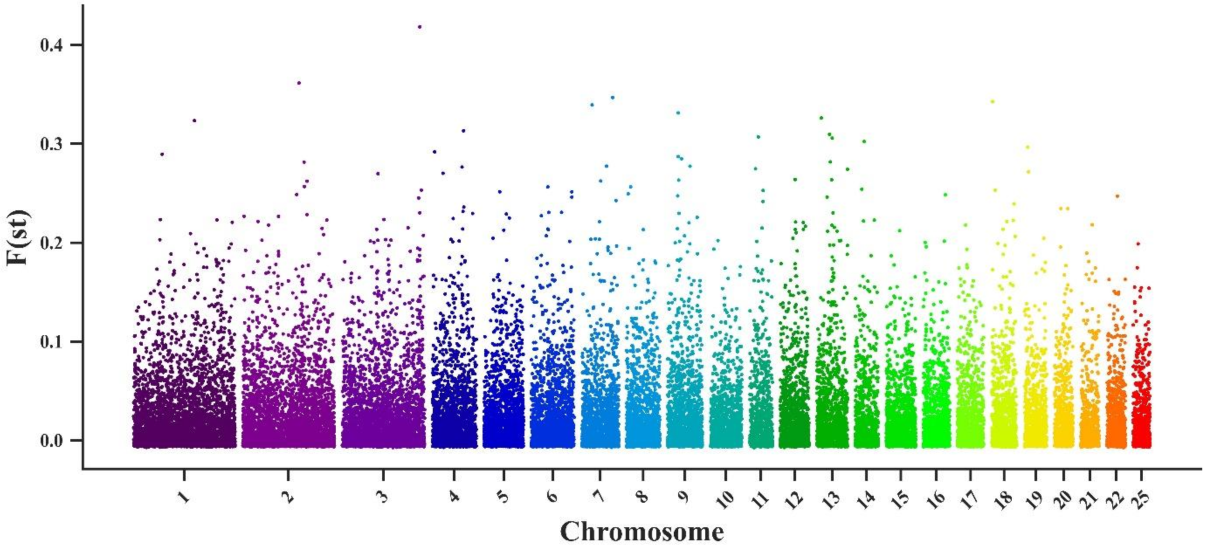


Figure 8. Generated Manhattan plot using GENOVIS *manplot* module (genovis manplot --d example_manplot_fst.txt --c nipy_spectral --nc 26 --ylab "F(st)" --f "Times New Roman" --mode solid --o manplot --ft jpg --dpi 700)

5. General hints

1. Color palettes are limited to matplotlib color palettes (https://matplotlib.org/stable/users/explain/colors/colormaps.html) and their reversed (*_r) versions.

2. Users must select font types based on the installed fonts on their systems.

3. Users can run the interface version of GENOVIS on Linux-based servers, provided that their SSH application supports X11 forwarding. We highly recommend using MobaXterm software, which is an SSH application that supports X11 forwarding.

4. GENOVIS can not work with dataframes that include headers. Therefore, users must make sure that their inputs do not have headers.

5. There is no need to define "--dpi" for generating vector image formats (e.g., SVG, EPS, and PDF). For raster image formats (e.g., TIFF, PNG, and JPG), users must consider the memory of the systems they use.

6. MIT License for GENOVIS

```
MIT License for GENOVIS

Copyright (c) 2025 Siavash Salek Ardestani, Elmira Mohandesan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy, modify, merge, publish,
```

7. References

1. Hunter JD: **Matplotlib: A 2D graphics environment**. *Computing in Science & Engineering* 2007, **9**(3):90-95.

2. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ *et al*: **Array programming with {NumPy}**. *Nature* 2020, **585**:357-362.

3. McKinney W: **Data structures for statistical computing in python**. In: *Proceedings of the 9th Python in Science Conference: 2010; Austin, TX*. 51-56.

4. Waskom M, Botvinnik O, O'Kane D, Hobson P, Lukauskas S, Gemperline DC, Augspurger T, Halchenko Y, Cole JB, Warmenhoven J *et al*: **mwaskom/seaborn: v0.8.1 (September 2017)**. In., vol. v0.8.1: Zenodo; 2017.

5. Kijas JW, Lenstra JA, Hayes B, Boitard S, Porto Neto LR, San Cristobal M, Servin B, McCulloch R, Whan V, Gietzen K *et al*: **Genome-Wide Analysis of the World's Sheep Breeds Reveals High Levels of Historic Mixture and Strong Recent Selection**. *PLOS Biology* 2012, **10**(2):e1001258.

6. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, Bender D, Maller J, Sklar P, de Bakker PI, Daly MJ *et al*: **PLINK: a tool set for whole-genome association and population-based linkage analyses**. *American journal of human genetics* 2007, **81**(3):559-575.

7. Chang CC, Chow CC, Tellier LCAM, Vattikuti S, Purcell SM, Lee JJ: **Second-generation PLINK: rising to the challenge of larger and richer datasets**. *GigaScience* 2015, **4**(1):s13742-13015-10047-13748.

8. Biscarini F, Cozzi P, Gaspa G, Marras G: **Detect runs of homozygosity and runs of heterozygosity in diploid genomes**. 2019.

9. Team RC: **R: A Language and Environment for Statistical Computing**. 2024.