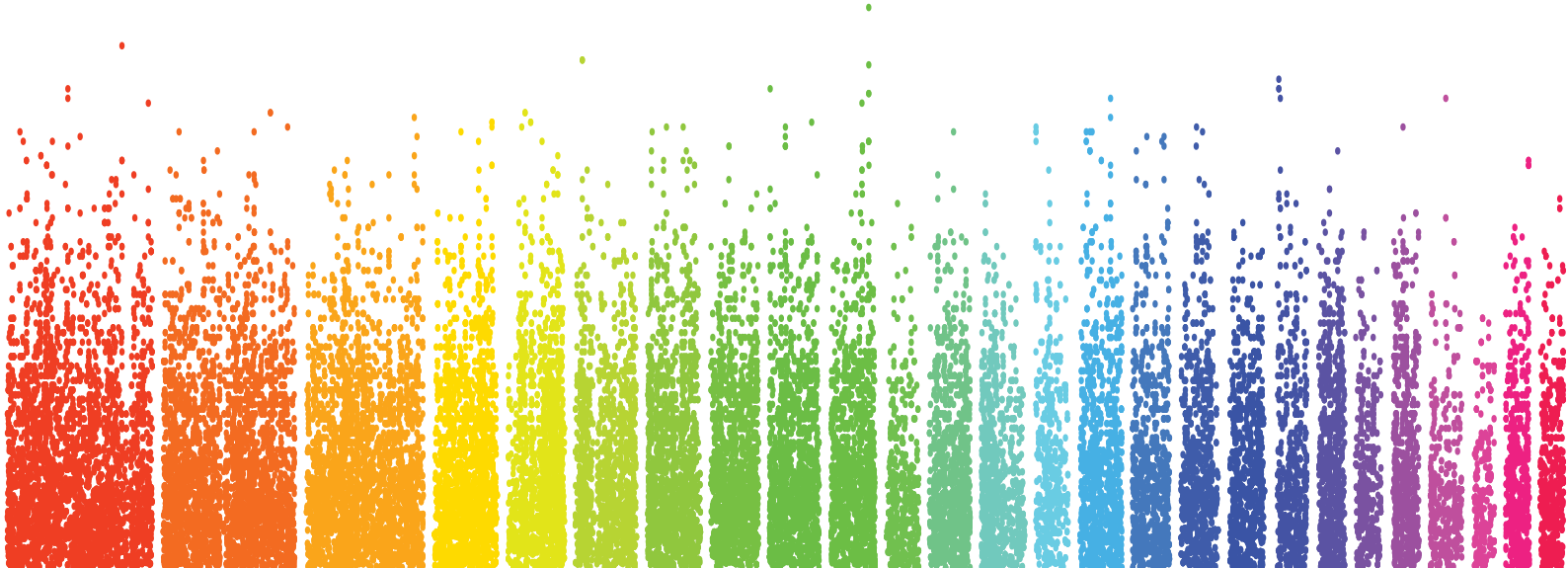
GENOVIS

Version 1.0



GENOVIS

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), *3dpca* (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 (Hunter, 2007) (version 3.10.3), argparse (version 1.1), numpy (Harris *et al.*, 2020) (version 2.2.5), pandas (McKinney, 2010) (version 2.2.3), and seaborn (Waskom *et al.*, 2017) (version 0.13.2), to develop GENOVIS in the Python environment (version 3.11.12).

2. Interface-portable version

2.1. Windows (portable)

Users can directly download the interface-portable version from the GitHub page (https://github.com/Siavash-cloud/GENOVIS). Then, by changing to the "Windows" folder and clicking on "GENOVIS-GUI.exe" (Figure 1), users can employ the graphical interface portable version of GENOVIS.

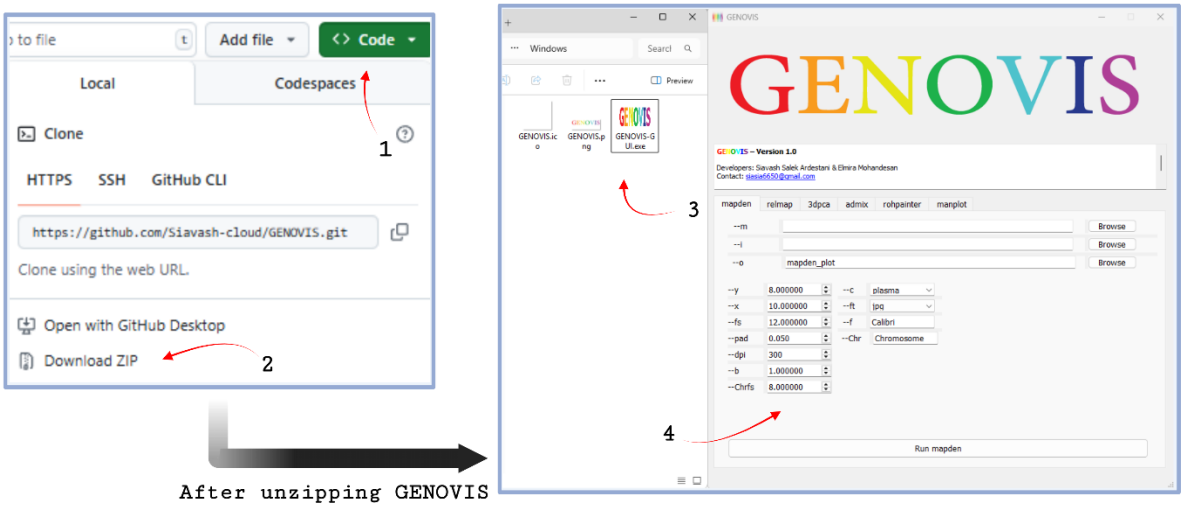


Figure 1. Downloading and executing GENOVIS (graphical interface portable version) without any installation.

2.2. Linux (portable)

Users can easily download from the GitHub page and, consequently, execute the Linux version of GENOVIS-GUI (graphical interface portable version) using the following command in the Linux terminal:

```
git clone https://github.com/Siavash-cloud/GENOVIS.git
cd GENOVIS/Linux
./GENOVIS-GUI
```

3. Command-line-based version

To use the command-line-based version, users must install Python (version 3.11) and some Python libraries (matplotlib 3.10.3, numpy 2.2.5, pandas 2.2.3, seaborn 0.13.2, argparse 1.1) on their machines to prevent errors and conflicts.

3.1. Linux (command-line-based version)

We highly recommend using the aforementioned versions of Python libraries for users who prefer to use the command-line-based version. Downloading main files from GitHub page:

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

Installation of Python libraries (dependencies):

```
pip install matplotlib==3.10.3 numpy==2.2.5 pandas==2.2.3 seaborn==0.13.2 argparse==1.1
```

Alternatively, users can make a virtual Python environment using conda:

```
conda create -n GENOVIS python=3.11.12
conda activate GENOVIS
conda install matplotlib=3.10.3 numpy=2.2.5 pandas=2.2.3 seaborn=0.13.2
```

Instead of using conda to install Python libraries, users can also use “pip” command after installation and activation of the aforementioned virtual Python environment:

```
pip install matplotlib==3.10.3 numpy==2.2.5 pandas==2.2.3 seaborn==0.13.2
```

Finally, users can execute "GENOVIS.py" in the terminal:

```
python  GENOVIS.py [options]
```

4. How to use GENOVIS

4.1. *mapden*

The 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
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.	--o	Path to output file	Optional
4.	--x	Horizontal size of figure	Optional
5.	--y	Vertical size of figure	Optional
6.	--fs	Font size (except for chromosome labels)	Optional
7.	--ft	Format type of figure (PDF, JPG, etc.)	Optional
8.	--c	Colormaps	Optional
9.	--dpi	Dots per inch	Optional
10.	--f	Font type	Optional
11.	--pad	Distance between legend and main figure	Optional
12.	--Chr	Chromosomal prefix(chr, chromosome, contig, or whatever the user wants)	Optional
13.	--b	Bin size (Mbp, default=1 Mbp)	Optional
14.	--Chrfs	Font size of chromosome labels	Optional

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

1. Map file (--m, PLINK format) (Purcell *et al.*, 2007): 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:

```
python GENOVIS.py mapden --m sheep_map.map --fs 12 --f Arial --o Figure_mapden --ft jpg --
Chr "Chr" --i sheep_genome_index.txt --pad 0.01 --dpi 300
```

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 (Chang *et al.*, 2015) (https://www.cog-genomics.org/plink/1.9/distance).

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

#	Flag	Explanation	Optional/Required
1.	--relmap	Path to relationship matrix file	Required
2.	--rf	Format of relationship matrix (--rf col/mat)	Required
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.	--mask	Mask diagonal elements or not (true/false)	Optional
5.	--a	Annotation of relationship values in heatmap plot (true/false)	Optional
6.	--afs	Font size of annotations	Optional
7.	--av	Output for averages of relationships among populations (true/false)	Optional
8.	--sl	Show individual labels (true/false)	Optional
9.	--lc	Color of separator lines (black)	Optional
10.	--lws	Size of separator lines	Optional
11.	--pfs	Font size of population labels	Optional
12.	--t	Title of legend	Optional
13.	--o	Path to output file	Optional
14.	--x	Horizontal size of figure	Optional
15.	--y	Vertical size of figure	Optional
16.	--xyfs	Font size of individual labels	Optional
17.	--ft	Format type of figure (PDF, JPG, etc.)	Optional
18.	--c	Colormaps	Optional
19.	--dpi	Dots per inch	Optional
20.	--f	Font type	Optional

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

```
python GENOVIS.py relmap --rf mat --relfile GRM.txt --matindex GRM_index.txt --t
"Relationship" --c viridis --sl false --x 7 --y 6 --xyf 10 --pfs 6 --f "Times New Roman" --
dpi 300 --a false --o relmap --mask false --av true
```

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.

4.3. *3dpca*

The 3dpca 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 *relmap* module.

#	Flag	Explanation	Optional/Required
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
5.	--azim	Azimuth angle	Optional
6.	--elev	Elevation angle	Optional
7.	--fp	i^{th} PC on x axis (default=1, which means PC_1)	Optional
8.	--sp	j^{th} PC on y axis (default=2, which means PC_2)	Optional
9.	--tp	k^{th} PC on z axis (default=3, which means PC_3)	Optional
10.	--s	Size of scatter points	Optional
11.	--o	Path to output file	Optional
12.	--x	Horizontal size of figure	Optional
13.	--y	Vertical size of figure	Optional
14.	--ft	Format type of figure (PDF, JPG, etc.)	Optional
15.	--c	Colormaps	Optional
16.	--dpi	Dots per inch	Optional
17.	--f	Font type	Optional

Example 3. Command-line-based example for *3dpca* module:

```
python GENOVIS.py 3dpca --evec plink.eigenvec --eval plink.eigenval --fp 1 --sp 2 --tp 3 --s
70 --mod int --f "Times New Roman" --azim 45 --elev 65 --o 3dpcaplot --ft svg --y 8 --x 10
```

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
1.	--d	Path to input file	Required
2.	--o	Path to output file	Optional
3.	--sl	Show individual labels (true/false)	Optional
5.	--xt	Font size of individual labels	Optional
6.	--x	Horizontal size of figure	Optional
7.	--y	Vertical size of figure	Optional
8.	--lws	Size of separator lines	Optional
9.	--ft	Format type of figure (PDF, JPG, etc.)	Optional
10.	--c	Colormaps	Optional
11.	--dpi	Dots per inch	Optional
12.	--f	Font type	Optional

Example 4. Command-line-based example for *admix* module:

```
python GENOVIS.py admix --d K8.txt --c tab20 --f "Times New Roman"  --sl "true" --xt 1 --o K8
--x 12 --y 4 --fs 8 --ft jpg --dpi 600
```

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
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.	--Chr	Chromosomal prefix(chr, chromosome, contig, or whatever the user wants)	Optional
4.	--xt	Font size of xticks	Optional
5.	--yt	Font size of yticks	Optional
6.	--t	Threshold for identifying common intervals(e.g., 0.8). Please use "--t false", if you do not want to pinpoint common intervals.	Optional
7.	--tc	Threshold line color	Optional
8.	--tw	Threshold line width	Optional
9.	--sl	Show individual labels (true/false)	Optional
10.	--o	Path to output file	Optional
11.	--x	Horizontal size of figure	Optional
12.	--y	Vertical size of figure	Optional
13.	--fs	Font size	Optional
14.	--ft	Format type of figure (PDF, JPG, etc.)	Optional
15.	--c	Colormaps	Optional
16.	--dpi	Dots per inch	Optional
17.	--f	Font type	Optional

Example 5. Command-line-based example for *rohpainter* module:

```
python GENOVIS.py rohpainter --d ROHs_two_breeds.txt --i sheep_genome_index.txt --x 10 --y 4
--f "Times New Roman" --t 0.4 --sl true --tc blue --tw 0.25 --ft pdf --yt 3 --c hsv --o
ROHs_plot
```

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 mahattan plot, 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
1.	--d	Path to data frame	Required
2.	--nc	Number of colors	Optional
3.	--ylab	Label of y-axis	Optional
4.	--xlab	Label of x-axis	Optional
5.	--a	The alpha blending value, between 0 (transparent) and 1 (opaque)	Optional
6.	--s	Scatter size	Optional
7.	--xt	Font size of xticks	Optional
8.	--yt	Font size of yticks	Optional
9.	--sug1	Suggestive line 1	Optional
10.	-- sug2	Suggestive line 2	Optional
11.	--sug1lw	Width size of suggestive line 1	Optional
12.	--sug2lw	Width size of suggestive line 2	Optional
13.	--sug1c	color of suggestive line 1	Optional
14.	--sug2c	color of suggestive line 2	Optional
15.	--o	Path to output	Optional
16.	--x	Horizontal size of figure	Optional
17.	--y	Vertical size of figure	Optional
18.	--fs	Font size	Optional
19.	--ft	Format type of figure (PDF, JPG, etc.)	Optional
20.	--c	Colormaps	Optional
21.	--dpi	Dots per inch	Optional
22.	--f	Font type	Optional

Example 6. Command-line-based example for *manplot* module:

```
python GENOVIS.py manplot --d Pi.txt --nc 26 --ylab π --xlab Chromosome --f “Times New Roman”
--c gnuplot2 --dpi 600 --sug1 0.000015 --o man_plot
```

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

7. References

Chang, C. C., C. C. Chow, L. C. A. M. Tellier, S. Vattikuti, S. M. Purcell and J. J. Lee, 2015: Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience,* **4,** s13742-13015-10047-13748.

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, 2020: Array programming with {NumPy}. *Nature,* **585,** 357-362.

Hunter, J. D., 2007: Matplotlib: A 2D graphics environment. *Computing in Science & Engineering,* **9,** 90-95.

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

Purcell, S., B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. de Bakker, M. J. Daly and P. C. Sham, 2007: PLINK: a tool set for whole-genome association and population-based linkage analyses. *American journal of human genetics*, **81,** *559-575*.

Waskom, M., O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee and A. Qalieh, 2017: mwaskom/seaborn: v0.8.1 (September 2017). Zenodo.