Protocol to analyze population structure and migration history based on human genome variation data

Zhao, Zicheng; Wang, Yinan; Zhang, Zhe; Li, Shuai Cheng

Published in:
STAR Protocols

Published: 17/03/2023

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY-NC-ND

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1016/j.xpro.2022.101928

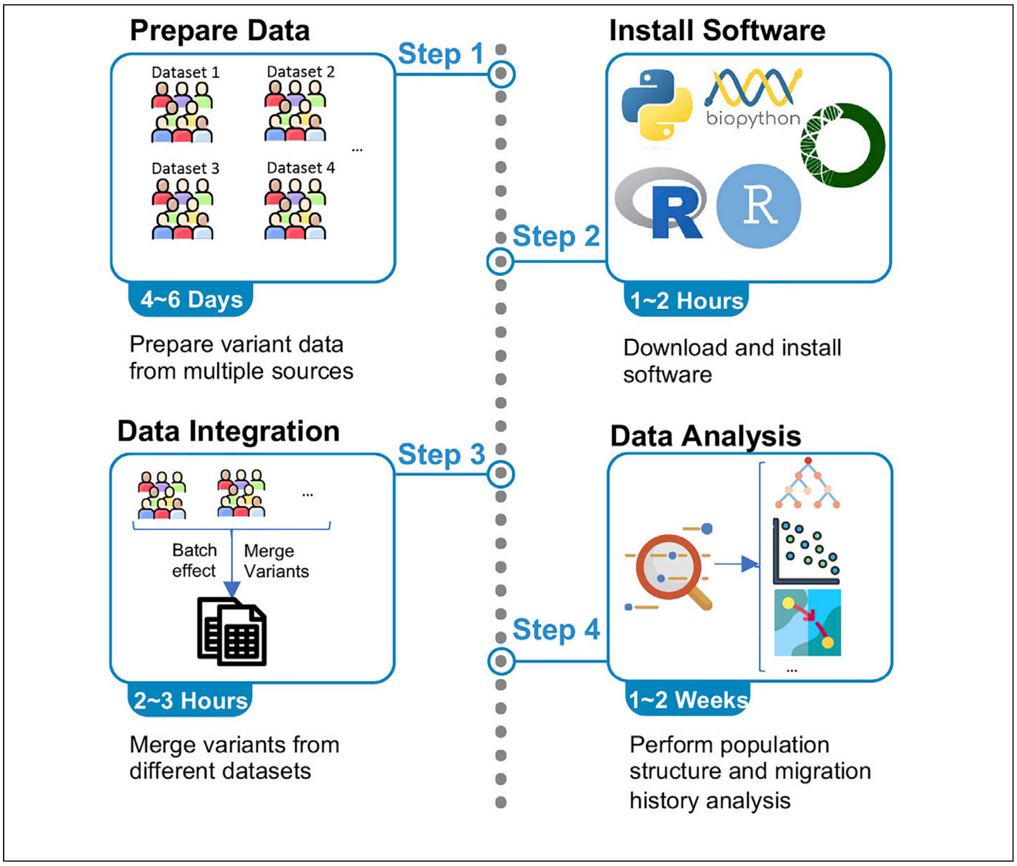
Publication details:
Zhao, Z., Wang, Y., Zhang, Z., & Li, S. C. (2023). Protocol to analyze population structure and migration history based on human genome variation data. *STAR Protocols*, *4*(1), Article 101928. https://doi.org/10.1016/j.xpro.2022.101928

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol to analyze population structure and migration history based on human genome variation data



Zicheng Zhao, Yinan Wang, Zhe Zhang, Shuai Cheng Li

shuaicli@cityu.edu.hk

Highlights

Population structure and migration history analysis with variation data

Semi-automated scripts for variant filtering and eliminating batch effects

Multi-source human variant data integration

We describe a protocol to integrate genome variation data from different datasets and explore the population structure and migration history of human populations. This protocol provides semi-automated scripts to perform and visualize the effect of variant filtering strategy on eliminating batch effects, principal component analysis, ancestry component analysis, historical population effective size inference, and migration and isolation analysis based on independent biallelic SNPs, genotype likelihoods, and haplotypes. The protocol can be adapted to variation data from other sources.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Zhao et al., STAR Protocols 4, 101928
March 17, 2023 © 2022 The Authors.
https://doi.org/10.1016/j.xpro.2022.101928

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol to analyze population structure and migration history based on human genome variation data

Zicheng Zhao,[1,2,6] Yinan Wang,[3,4] Zhe Zhang,[5] and Shuai Cheng Li[1,7,*]

[1]Department of Computer Science, City University of Hong Kong, Kowloon 999077, Hong Kong

[2]Shenzhen Byoryn Technology Co., Ltd., Shenzhen 518000, China

[3]Peking University Shenzhen Hospital, 1120 Lianhua Road, Shenzhen 518036, China

[4]School of Medicine, Southern University of Science and Technology, 1088 Xueyuan Avenue, Shenzhen 518055, China

[5]Department of Cardiology, Zhuhai People's Hospital (Zhuhai Hospital Affiliated with Jinan University), No. 79 Kanning Road, Zhuhai 519000, China

[6]Technical contact: zhaozicheng@byoryn.com

[7]Lead contact

*Correspondence: shuaicli@cityu.edu.hk

https://doi.org/10.1016/j.xpro.2022.101928

SUMMARY

We describe a protocol to integrate genome variation data from different datasets and explore the population structure and migration history of human populations. This protocol provides semi-automated scripts to perform and visualize the effect of variant filtering strategy on eliminating batch effects, principal component analysis, ancestry component analysis, historical population effective size inference, and migration and isolation analysis based on independent biallelic SNPs, genotype likelihoods, and haplotypes. The protocol can be adapted to variation data from other sources.

For complete details on the use and execution of this protocol, please refer to Zhang et al. (2022).[1]

BEFORE YOU BEGIN

This protocol describes how to analyze population structure and migration history based on genome variation data from the Tibetan-Yi Corridor,[1] Tibetan Highlanders Project,[2] Simons Genome Diversity Project,[3] and 1000 Genomes Project (1KGP) Phase 3.[4] It can also be applied to population studies using other variation data resources. If this is desired, reliable variations should be called following GATK[5] best practice with proper filtering strategy in each data resource. The code described in this protocol was solely tested under the Linux operating system. If you are using another operating system, please check compatibility.

Institutional permissions

The variation data from the Tibetan-Yi Corridor Project are retrieved in the Genome Variation Map (GVM) in the Big Data Center, Beijing Institute of Genomics (BIG), Chinese Academy of Science. Users should contact the data submitter (anyzh69@gmail.com) to apply for permission before accessing the data.

Download variation data

⏱ Timing: 4–7 days

The data sets required to implement the code in this protocol include variation data from the Tibetan-Yi Corridor Project, the Tibetan Highlanders Project, the Simons Genome Diversity Project, and the 1KGP Phase 3. The downloaded data should be preprocessed to a variable call format (VCF) for subsequent analyses.

1. Download and pre-processing variation data from the Tibetan-Yi Corridor Project.
   a. The variation data for this project have been deposited in the GVM in the Big Data Center (accession number GVM000100) at http://bigd.big.ac.cn/gvm/getProjectDetail?project=GVM000100. The data can be downloaded by contacting the data submitter (amyzh69@gmail.com) to apply for permission and forwarding the approved email to gvm@big.ac.cn to get the data link. The data downloaded is a *vcf* file containing 248 individuals from 16 populations.
2. Download and pre-processing variation data from the Tibetan Highlanders Project.
   a. The Tibetan Highlanders Project has only published raw next generation sequencing data in *fastq* format. The data have been deposited in the Genome Sequence Archive (GSA) and the National Omics Data Encyclopedia (NODE) under accession numbers GSA: PRJCA000246 and NODE: ND00000013EP, respectively.
   b. Run the "filterfq" software to remove adaptors and low-quality reads by command:

```
> filterfq -f <sample i>_1.fq.gz <sample i>_2.fq.gz -O <work_dir> -o <sample i> -T 8
```

Note: Users should run the command individually by sample number.

Note: This command is for pair-end sequencing data with forward sequencing reads in <sample i>_1.fq.gz file and reverse sequencing reads in <sample i>_2.fq.gz.

Note: The "-T" parameter is the thread number for running this program and can be manually set according to the computation resource.

   c. Run the "BWA" software to align the clean reads obtained by "filterfq" to the human reference genome by the following commands:

```
> bwa mem -t 8 <ref.fa> \
    <sample i>_1.clean.fq.gz \
    <sample i>_2.clean.fq.gz | \
  samtools view -F 0x800 -b -T <ref.fa> | \
  samtools sort -thread 8 -n | \
  samtools fixmate -O bam - <sample i>.fixmate.bam
> samtools sort -thread 8 <sample i>.fixmate.bam -o <sample i>.sort.bam
```

Note: This command is for the alignment of pair-end sequencing reads.

Note: The "Fixmate" sub-command will fill in mate coordinates and insert size field for pair-end reads. Before running the "Fixmate" subcommand, the alignment file should be sorted by read name (-n option in the "samtools sort" command).

Note: The reference file denoted by <ref.fa> is a human reference. The candidate reference versions for this protocol are hg19 or hg38. Subsequent analyses should use the same reference version.

   d. Run "Picard" software to remove duplicate reads by command:

```
> java -Xmx5g -jar picard.jar MarkDuplicates \
   I=<sample i>.sort.bam \
   O=<sample i>.markdup.bam \
   M=<sample i>.markdup.matrics
```

e. This step includes calling variants per sample in *gvcf* format, consolidating *gvcf* files, joint calling cohort in *vcf* format, and performing variant filtering by variant quality score. Users can perform the analysis following the GATK best practice of germline short variant discovery (SNPs+indels) available at https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels-.

Note: After running the "GATK" pipeline, the variants of the samples in this cohort will be merged into one variant file in *vcf* format.

f. Further filtering variants in the *vcf* file by cohort statistics typically follows the rules below:
   i. Only bases with the Illumina base quality of at least 20 are included;
   ii. Only reads with BWA mapping quality of at least 13 are included;
   iii. Sites with less than two reads were set to missing value; Sites with missing value in more than 15% of the samples were filtered out;
   iv. Hardy-Weinberg proportions: The expected genotype frequencies are calculated for each variable site based on allele frequencies. Variants with "ExcHet = 1" annotated by bcftools 1.9 are filtered out;
   v. Potentially paralogous variants are excluded based on the strict accessibility mask from the 1000 Genome Project and the 100-mer mappability track in the UCSC Genome Browser.

Note: The filtering strategy aims to address the possible batch effects between different datasets and obtain a set of genomic loci deemed reliable for population genetic analysis.

Note: For in-house data, users can try this filtering strategy and check the batch effects using the module "SiteEval" (see Section "data integration and batch effect evaluation"). Users can set a more stringent variant filtering strategy if the variants fail the batch-effect evaluation.

3. Download and preprocess variation data from the Simons Genome Diversity Project.
   a. The variation data of this project can be downloaded at https://sharehost.hms.harvard.edu/genetics/reich_lab/sgdp/vcf_variants/vcfs.variants.public_samples.279samples.tar. The downloaded data is a *vcf* file containing filtered variants and thus can be used directly.
4. Download and pre-processing variation data from 1KGP Phase 3.
   a. The variation data for this project can be downloaded at https://www.internationalgenome.org/category/phase-3/. The downloaded data is a *vcf* file containing filtered variants and thus can be used directly.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
1000 Genomes Project Phase 3	1000 Genomes Project Consortium, 2015 (Auton et al.[4])	https://www.internationalgenome.org/category/phase-3/
The Simons Genome Diversity Project	(Mallick et al.[3])	EBI-ENA: PRJEB9586, ERP010710

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Tibetan Highlanders Project	(Lu et al.[2])	GSA: PRJCA000246 NODE: ND00000013EP
Tibetan-Yi Corridor population variation database	(Zhang et al.[1])	GVA: GVM000100
Software and algorithms		
filterfq (v1.2.0)	(Chen et al.[6])	https://github.com/bowentan/filterfq
BWA (v0.7.13)	(Li and Durbin[7]) http://bio-bwa.sourceforge.net/	RRID: SCR_010910
Samtools (v1.9)	(Li et al.[8]) http://samtools.sourceforge.net/	RRID: SCR_002105
Picard (v2.1.0)	BroadInstitute http://broadinstitute.github.io/picard/	RRID: SCR_006525
GATK (v3.8)	(Manichaikul et al.[9]) https://software.broadinstitute.org/gatk/	RRID: SCR_001876
bcftools (v1.9)	(Li[10]) https://samtools.github.io/bcftools/	RRID: SCR_005227
PLINK (v1.90)	(Purcell et al.[11]) https://www.cog-genomics.org/plink2/	RRID: SCR_001757
ANNOVAR (v2018Apr16)	(Wang et al.[12]) http://annovar.openbioinformatics.org/en/latest/	RRID: SCR_012821
ANGSD (v0.931)	(Korneliussen et al.[13]) http://www.popgen.dk/angsd/index.php/ANGSD/	RRID: SCR_021865
Beagle (v4.0)	(Browning and Browning[14]) https://faculty.washington.edu/browning/beagle/b4_0.html	RRID: SCR_001789
SHAPEIT (v2.0)	(Delaneau et al.[15])	https://mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.html
KING (v2.0.1)	(Manichaikul et al.[9]) http://people.virginia.edu/~wc9c/KING/	RRID: SCR_009251
EIGENSOFT (v4.2)	(Price et al.[16]) https://data.broadinstitute.org/alkesgroup/EIGENSOFT/	RRID: SCR_004965
fineSTRUCTURE (v2.0)	(Lawson et al.[17]) http://www.paintmychromosomes.com/	RRID: SCR_018170
ADMIXTURE (v1.3.0)	(Alexander et al.[18]) http://software.genetics.ucla.edu/admixture/	RRID: SCR_001263
CLUMPAK (v1.1)	(Kopelman et al.[19])	http://clumpak.tau.ac.il
Ohana (v1.0)	(Cheng et al.[20])	https://github.com/jade-cheng/ohana
EEMS (v1.0)	(Petkova et al.[21])	https://github.com/dipetkov/eems
MSMC (v1.0)	(Schiffels and Durbin[22])	https://github.com/stschiff/msmc
AdmixTools (v6.0)	(Patterson et al.[23]) https://github.com/DReichLab/AdmixTools/	RRID: SCR_018495

STEP-BY-STEP METHOD DETAILS

Here, we describe how to combine variant files from multiple datasets for downstream population structure analysis and historical migration analysis. To illustrate the data process steps, we show the analysis process and results of four different datasets used by Zhang et al. (2022)[1] as an example.

Install software

Timing: 1–2 h

Several analysis toolkits are required to perform computational tasks such as batch-effect estimation, variant calling and filtering, and downstream statistical analyses of population genetics.

1. Install Python and related packages.
   a. Install Python (version 3.8 or current version). The package and documentation are available at https://www.python.org/.
   b. Install NumPy (version 1.19.2 or current version). Download and documentation are available from https://numpy.org/. To install NumPy, you can type:

```
>pip install numpy==1.19.2
```

   c. Install scikit-learn (version 0.23.2 or current version). The package and documentation are available from https://scikit-learn.org/stable/. To install scikit-learn, you can type:

```
>pip install sklearn==0.23.2

sk
```

   d. Install PyVCF (version 0.6.8 or current version). The package and documentation are available at https://pyvcf.readthedocs.io/en/latest/. To install PyVCF, you can type:

```
>pip install pyvcf==0.6.8

sk
```

   e. Install pandas (version 1.4.1 or current version). The download and documentation are available from https://pandas.pydata.org/. To install pandas, you can type:

```
>pip install pandas==1.4.1
```

2. Install R and related packages.
   a. Install R (version 3.6.3 and version 4.2.0). The download and documentation are available from https://www.r-project.org/.
   b. Install Bioconductor (version 3.10 or current version).[24] Installation instructions and documentation are available from https://www.bioconductor.org/. To install the Bioconductor, start R and type:

```
>if (!require(‘‘BiocManager’’, quietly = TRUE))

  install.packages(‘‘BiocManager’’)

>BiocManager::install(version = ‘‘3.10’’)
```

   c. Install genotypeevel (version 3.15 or current version). The required R version for this package is 4.2.0. Installation instructions and documentation are available at http://bioconductor.org/packages/release/bioc/html/genotypeeval.html. To install genotypeevel, start R and type:

```
>BiocManager::install(‘‘genotypeeval’’)
```

3. Install filterfq.
   a. Install filterfq (version v1.2.0).[6] The download and documentation are available from https://github.com/bowentan/filterfq.
4. Install BWA.
   a. Install BWA (version v0.7.13).[7] The download and documentation are available from http://bio-bwa.sourceforge.net/.
5. Install Samtools.
   a. Install Samtools (version v1.9).[8] Download and documentation are available from http://samtools.sourceforge.net/.

6. Install Picard.
   a. Install Picard (version v2.1.0).[25] The download and documentation are available from http://broadinstitute.github.io/picard/.
7. Install GATK.
   a. Install GATK (version v3.8). The download and documentation are available at https://software.broadinstitute.org/gatk/.
8. Install ANNOVAR.
   a. Install ANNOVAR (version v2018Apr16). The download and documentation are available from https://annovar.openbioinformatics.org/en/latest/.
   b. The dbSNP database is required for downstream analyses. The database can be downloaded and configured using the following command line:

```
> perl annotate_variation.pl -downdb -webfrom annovar -buildver hg19 snp138 humandb/
```

9. Install bcftools.
   a. Install bcftools (version v1.9).[10] The download and documentation are available at https://samtools.github.io/bcftools/.
10. Install PLINK.
    a. Install PLINK (version v1.90).[26] Download and documentation are available from https://www.cog-genomics.org/plink2.
11. Install ANGSD.
    a. Install ANGSD (version v0.931).[13] Download and documentation are available at http://www.popgen.dk/angsd/index.php/ANGSD/.
12. Install Beagle and phasing bundle resources.
    a. Install Beagle (version v4.0).[14] The download and documentation are available from https://faculty.washington.edu/browning/beagle/b4_0.html.
    b. Download the Beagle phasing bundle from https://bochet.gcc.biostat.washington.edu/beagle/genetic_maps/.
    c. Download the 1KGP phase 3 reference panel from https://bochet.gcc.biostat.washington.edu/beagle/1000_Genomes_phase3_v5a/.
13. Install SHAPEIT2.
    a. Install SHAPEIT2 (version 2.0).[15] Download and documentation are available from https://mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.html.
14. Install KING.
    a. Install KING (version v2.0.1).[9] Download and documentation are available from http://people.virginia.edu/~wc9c/KING/.
15. Install EIGENSOFT.
    a. Install EIGENSOFT (version v4.2).[16] The download and documentation are available at https://data.broadinstitute.org/alkesgroup/EIGENSOFT/.
16. Install fineSTRUCTURE.
    a. Install fineSTRUCTURE (version v2.0).[16] Download and documentation are available at http://www.paintmychromosomes.com/.
17. Install ADMIXTURE.
    a. Install ADMIXTURE (version v1.3.0).[18] Download and documentation are available at https://dalexander.github.io/admixture/download.html/.
18. Install CLUMPAK.
    a. Install CLUMPAK (version v1.1).[19] The download and documentation are available from http://clumpak.tau.ac.il.
19. Install Ohana.
    a. Install Ohana (version v1.0).[20] The download and documentation are available at https://github.com/jade-cheng/ohana.

20. Install EEMS.
    a. Install EEMS (version v1.0).[21] The download and documentation are available from https://github.com/dipetkov/eems.
21. Install MSMC.
    a. Install MSMC (version v1.0).[27] Download and documentation are available from https://github.com/stschiff/msmc.
22. Install AdmixTools.
    a. Install AdmixTools (version v6.0).[23] The download and documentation are available from https://github.com/DReichLab/AdmixTools/.

⚠ CRITICAL: The indicated software and package versions were used by Zhang et al.[1] Other versions of the software packages than those indicated here were not tested. If you intend to use different versions of software or packages, please check compatibility and be aware that the steps described in this protocol might not work as expected.

Data integration and batch effect evaluation

⏱ Timing: 2–3 h

The data-combination step merges the variation data from the four datasets above into a single *vcf* file. The non-random associated variants described by linkage disequilibrium (LD) add redundancy in subsequential analyses; thus, it is essential to remove linkage variants based on pairwise LD. The remaining variants are then annotated by the dbSNP database and used to evaluate the batch effect. Users can perform batch-effect elimination with a more stringent variant filtering strategy (compared to the aforementioned filtering strategy in the Section "download variation data") if the variants fail the batch-effect evaluation. For example, the lowest base quality can be increased to 30 and/or the lowest BWA mapping quality can be increased to 20.

23. Data integration.

Adopt the "merge" subcommand of the software "bcftools" to combine the variants vcf files from different resources. Adopt the "view" subcommand to filter the variants in the *vcf* files.

```
> bcftools merge -O z \
    -o MergedVariants.vcf.gz \
    <source_1>.vcf.gz <source_2>.vcf.gz ...
> bcftools view -v snps -m2 -M2 MergedVariants.vcf.gz
```

Note: "bcftools" "merge" subcommand has the option (-0, –missing-to-ref) to use reference allele (0/0) instead of the default missing genotype. In our study, any variant is absent in one of the *vcf* files to be merged will be treated as missing information in the merging process.

Note: We keep only biallelic SNPs for subsequent analyses.

24. LD pruning.

LD pruning by the software "plink" use the "–indep-pairwise" option with three parameters: window size (kb), step size, and r2 threshold. Blog "https://blog.goldenhelix.com/determining-best-ld-pruning-options" describes the details of the parameters and how to determine the best parameter values for LD pruning. It is common practice to set the parameters to 500, 50, and 0.2 for window size, step size, and r2 threshold, respectively, in studies of homo species populations. The commands for filtering linkage SNPs with PLINK are:

```
> plink –vcf MergedVariants.vcf.gz –indep-pairwise 500 50 0.2 -out prune

> plink –vcf MergedVariants.vcf.gz –extract prune.in –recode vcf –out MergedVaraints.prune.vcf
```

Note: We compressed and moved the pruned *vcf* file to "MergedVariants.vcf.gz", which contains independent biallelic SNPs, and we used this filename in the subsequent analyses.

Note: For the merged variants file used for haplotype phasing, skip this step.

25. Site annotation.

The command to annotate variants by "ANNOVAR" with the dbSNP database is:

```
> perl annotate_variation.pl -downdb -webfrom annovar -buildver hg19 snp138 humandb/
```

Note: Reference hg19 is used here as the human reference genome. If you choose hg38 as the reference from the beginning, please use "-buildver hg38".

26. Batch effect evaluation and variant statistics calculation.

We developed an integrated module "SiteEval" for batch effect evaluation and variant statistics calculation on the merged *vcf* file. SiteEval uses the R package "genotypeeval" to identify batch effects by some key quality metrics, such as percent of variants confirmed in dbSNP, mean genotype quality, median read depth, transition (Ti) transversion (Tv) ratio in noncoding and coding regions, and percent heterozygotes. Genotypeeval requires the coordinate cds regions, dbsnp, and known-gene regions for the corresponding reference genome version in *bed* format. These information are loaded by including R libraries "TxDb.Hsapiens.UCSC.hg38.knownGene" for hg38, and "TxDb.Hsapiens.UCSC.hg19.knownGene" for hg19, respectively. "SiteEval" also calculates variant statistics, including the number of Ti variants, the number of Tv variants, the Ti / Tv ratio, and the ratio of called variants shared with dbSNP or 1KGP. "SiteEval" is one of the subfolders in the Github repository PopBoost. The command line for running SiteEval is:

```
> python get_vcf_stats.py –in MergedVariants.vcf.gz –out MergedVariants.statusV
```

27. Prepare the variant file in *bed* format for downstream analyses.

The command line for transforming variant file in *vcf* format to *bed* format is:

```
> plink –vcf MergedVariants.vcf.gz –make-bed –out MergedVariants
```

Note: This command will generate four output files, namely MergedVariants.bed, MergedVariants.fam, MergedVariants.bim, and MergedVariants.log. The *bed* file contains variant genotypes in binary format. The *fam* file indicates the sample information and the *bim* file indicates the variant information. Detailed documentation of the *bed* format is available at https://www.cog-genomics.org/plink/1.9/formats.

Population structure and migration history analyses

Timing: 1–2 weeks

Population structure and migration history analyses require the merged variants as input. The merged variant file containing linkage SNPs is used for haplotype phasing to perform linked principal component analysis (PCA) and historical effective population size estimation. The merged variants file containing independent biallelic SNPs is used to perform PCA, ADMIXTURE analysis, homozygosity scan runs (ROH), EEMS analysis, and F_{st}, F_3 statistics, and D statistics estimation.

28. Kinship analysis.

Most downstream analyzes require unrelated individuals in the cohort as the close inheritance distances of related individuals provide redundant variants. We use the "KING" software to test cryptic relatedness between individuals. In our research, the threshold for the kinship coefficient of unrelated individuals was empirically set to <0.2. Individuals with pairwise coefficients higher than the threshold are excluded. The command line for running "KING" software is:

```
> king -b MergedVariants.bed \
   -fam MergedVariants.fam \
   -bim ex.bim -related
```

29. Haplotype phasing.

In our method, we integrate "Beagle" and "SHAPEIT2" to obtain accurate phasing haplotypes due to the low coverage samples in the Tibetan-Yi Corridor Project (5×) and 1KGP (7.4×).

a. Run "Beagle" with the default parameters to get an initial set of genotypes. BEAGLE-called genotypes with posterior probability greater than 0.995 can be considered known genotypes.

Note: Users could also run a smaller number of iterations (e.g., 5) to reduce the runtime of this step. However, the phasing switch error could increase with fewer iterations if the result does not converge.

b. Run "SHAPEIT2" to phase the initial genotype set into haplotypes with ten burning-in iterations and ten pruning iterations. The process detail is available at https://mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.html#prephasing in the "Genotype calling from low coverage sequencing" subsection.

The phasing process is integrated into the PopBoost repository with a Python program in the "phase" subfolder. Users could run the script using the following command:

```
> python phase.py -mode low \
      -vcf MergedVariants.vcf.gz \
      -out MergedVariants.phased.vcf.gz \
      -thread 8 -burn 10 -prune 10 \
      -shapeit-path <shapeit2 program path> \
      -beagle-path <beagle java file path>
```

Note: The "MergedVariants.vcf.gz" file here is not processed by LD pruning and contains linkage SNPs.

Note: Users can ignore the –shapeit-path and –beagle-path parameters if the programs are on their system path.

Note: The mode parameter provides two options, namely "normal" and "lowcov". "Normal" is the default mode for the script. In "normal" mode, the script will run SHAPEIT2 directly.

30. PCA.

PCA could demonstrate population stratification. Users can perform analyses based on genotype and haplotype, respectively. Both methods are integrated into the PopBoost repository PCA subfolder. Users can perform PCA with the following steps.

   a. The command line for running the integrated script "pca.py" for the PCA genotype is:

```
> python pca.py -mode lowcov -vcf MergedVariants.vcf.gz -out out.pca
```

Note: The script provides a "low coverage" mode and a "normal" mode. "ANGSD" and "smartPCA" will be adopted in low-coverage mode and normal mode, respectively. "ANGSD" will use beagle genotype likelihood (BEAGLE-BL) information when calculating sample PCA, and thus is more suitable for studies with low coverage samples. The *vcf* file with BEAGLE-BL information can be obtained in the former haplotype phasing step.

   b. Adopt the software "fineSTRUCTURE" to perform linked PCA and investigate population structure based on haplotype data. "fineSTRUCTURE" requires phased variation data as input. The working flow is available at https://people.maths.bris.ac.uk/~madjl/finestructure/manualse2.html#x5-40002. The official tutorial recommends running "SHAPEIT2" to obtain phasing variants. Before running "fineSTRUCTURE", users should transform the "SHAPEIT2" output in ".haps" format to chromopainter format ending with ".phase" by "fineSTRUCTURE" util script "impute2chromopainter.pl" with the following command:

```
> perl impute2chromopainter.pl impute.haps output_prefix
```

31. ROH.

ROH provides evidence in the demographic history of the population ancestors. The software "bcftools" can calculate the value with fixed sliding window size and the minimum length of an ROH. The ROH analysis requires that each input file have individuals from the same population. Thus, users should separate the merged *vcf* file by population and run ROH separately. The command line for ROH analysis is as follows:

```
> bcftools roh -G30 -AF-dflt 0.4 <population.vcf.gz>
```

32. Historical population effective size inference.

The historical effective population size quantifies the rate of genetic drift and inbreeding in the demographic history of the population.[28] Software Multiple sequentially Markovian coalescent (MSMC) takes both phased and unphased data of multiple samples within a population to infer the population's historical effective size. The mutation rate and generation time are typically set at 1.25e-8 per base pair and 25 years per generation, respectively.

The "MSMC" tutorial and pipeline are available at "https://github.com/stschiff/msmc/blob/master/guide.md". We integrate the "MSMC" pipeline into the "msmc" subfolder of the PopBoost repository. The "MSMC" program requires either a bam file or a "masterVarBeta" file as input. The command line for running the integrated "MSMC" pipeline is:

```
> python msmc.py -sample-file <sample.list> -bam-folder <bam_folder> -out <outfolder> -ref
hg19.fa -thread 8
```

Note: Users can manually set the generation time and mutation rate by the "–mu" and "–g" parameters, respectively.

Note: The script requires "SHAPEIT2" and "samtools" in the system path. Users can also assign the program path by the parameters "–shapeit-path" and "–samtools-path".

33. Genetic diversity between populations (Fixation Index F_{st}).

F_{st} measures the genetic differences between populations. The analysis requires a population list file as input to compute pairwise F_{st} values. The command line for calculating F_{st} with software "vcftools" is:

```
> vcftools -gzvcf MergedVariants.vcf.gz -weir-fst-pop <pop1.indv.lst> -weir-fst-pop
<pop2.indv.lst>
```

Note: <pop1.indv.lst> and <pop2.indv.lst> are lists with sample IDs in population 1 and population 2.

34. F_3 statistics and D statistics analyzes.

Statistics F_3 and D provide evidence of historical genetic events between populations. The additive-F_3 statistic and the outgroup-F_3 statistic f_3(X, Y; Z) require three populations X, Y, and Z. For the additive-F_3, X and Y are regarded as background populations. Z is tested to check if it has ancestry from populations related to X and Y. A significantly negative value provides strong evidence of a mixture in the test population Z with X and Y. These three populations are selected according to research hypothesis. Outgroup-F_3 estimates the shared genetic drift between X and Y. Population Z is considered an outgroup population. The outgroup population is typically set to African populations like Mbuti. D statistics *D*(X, Y; Z, W) is used to test a tree-like relatedness between four populations. The blocked jackknife value is used to assess the statistical significance of the result. "Admixtools" is an integrated toolkit to calculate F_3 statistics and D statistics. For each calculation, a population list representing X, Y, Z in F_3 or X, Y, Z, W in D statistics should be provided.

a. The command line to calculate the admixture-F_3 statistic f_3(X, Y; test) is:

```
> qp3Pop -p <param.par>
```

Note: <param.par> is the config file for admixture-F_3 statistic. The config file is a description of each parameter in a par file. Here is an example from our study:

```
genotypename: MergedVaraints.bed

snpname: MergedVariants.bim

indivname: MergedVariants.ind

popfilename: <population list>

outgroupmode: NO
```

Note: The "MergedVariants.ind" file can be extracted from "MergedVariants.fam". The first column in the "MergedVariants.ind" file is the sample ID, the second column is the sample gender, and the third column is the sample population. The population list file is a tab-separated file, in which each line represents a three-population group. The input files have the same format in the subsequential outgroup-F_3 and D statistics analyses, despite the population list having a four-population group for D statistics analysis.

b. The command line to calculate the outgroup-F_3 statistic f_3(X, Y; outgroup) is:

```
> qp3Pop -p <param.par>
```

Note: <param.par> is the config file for outgroup-F_3 statistic. The config file is a description of each parameter in a par file. Here is an example from our study:

```
genotypename: MergedVariants.bed

snpname: MergedVaraints.bim

indivname: MergedVariants.ind

popfilename: <population list>

outgroupmode: YES
```

c. The command line for calculating the D statistics *D*(X, Y; Z, W) is:

```
> qpDstat -p <param.par>
```

Note: <param.par> is the config file for D statistics. The config file is a description of each parameter in a par file. Here is an example from our study:

```
genotypename: MergedVariants.bed

snpname: MergedVariants.bim

indivname: MergedVaraints.ind

popfilename: <population list>

f4mode: NO
```

35. Model-based clustering and ancestry component analyses.
We apply the "ADMIXTURE" software to perform model-based clustering analysis. The software takes variants in *bed* format as input. Users can set the number of ancestry components *k* (represented by the cluster number) according to the scale of their dataset. Our study set the k-value varying from 2 to 13 for the worldwide population dataset and from 2 to 8 for the regional population dataset. "ADMIXTURE" software uses the maximum-likelihood approach. Thus, it requires multiple experiments to obtain the best output clustering result. We set the number of repetitive experiments to 10 in our study. Users can custom set the number of repetitions. A more significant number of experimental repeats (≥ 10) will result in a more robust result. The best output can be identified and visualized by "CLUMPAK". "Ohana" takes the output components as input to conduct the population tree for the ancestry components.

a. Run the "ADMIXTURE" program to perform model-based clustering.

```
> admixture -j8 –cv MergedVariants.bed <k>
```

Note: The <k> parameter dedicates the number of ancestry components.

b. "CLUMPAK" provides an online website available at http://clumpak.tau.ac.il/. Users can perform the analysis following the instructions on the website http://clumpak.tau.ac.il/help.html. Note that the Q-matrixes obtained from "ADMIXTURE" should be compressed to a zip file before uploading. The "CLUMPAK" server will send the results to the user's email after the analysis is accomplished.

c. "Ohana" requires a *ped* format file as input. The command lines for conducting the population tree from ancestry components are as follows:

```
> convert ped2dgm MergedVariants.ped MergedVariants.dgm

> qpas MergedVariants.dgm -k <k> -qo q.matrix -fo f.matrix -mi 5

> nemeco MergedVariants.dgm f.matrix -co c.matrix -mi 5

> convert cov2nwk c.matrix tree.nwk

> convert nwk2svg tree.nwk tree.svg
```

Note: The <k> parameter in "qpas" command is the number of ancestry components, similar to "ADMIXTURE" analysis.

Note: In our study, we performed "Ohana" with only high-coverage samples in the Tibetan-Yi Corridor Project. If low coverage samples are included, users can adopt genotype likelihood data as input.

d. The whole process has been integrated in the "admixture" subfolder of the PopBoost repository. Users can perform "ADMIXTURE", "clumpak", and "ohana" analyses by one command line:

```
> bash run_admixture.sh MergedVariants.bed \

<output> \

<startk> <endk> \

<reptime>
```

Note: The parameters start-k and end-k define the minimum number of ancestry components and maximum number of ancestry components in the analysis, respectively. The "reptime" parameter defines the number of experiments in each k-value while running the "ADMIXTURE" program. "Ohana" requires fewer experiments for small k values. The number of experiments in "Ohana" will automatically be determined by the script "run_admixture.py".

36. Migration and isolation by distance.

Adopt "EEMS" for performing migration and isolation analysis. Before running "EEMS" for geolocation-based analysis, variants should be formatted in genetic dissimilarities by the "bed2diffs" script in the EEMS repository. "EEMS" takes one million iterations for each run, followed by an additional one thousand iterations for each posterior sample. The migration surfaces vary with the EEMS running threshold. The threshold value ranges from 0 to 1. A more significant threshold generates more stringent migration surfaces. The R script "rEEMSplots.R" is used to visualize the results on the map. The interpolation analysis is to visualize the ancestry coefficients (Q matrix) on a geographic map. The Q matrix is calculated by ADMIXTURE and selected by CLUMPAK in the previous step. The R script "POPSutilities.R" is used to visualize the result.

a. The command line for generating genetic dissimilarity by the "bed2diffs" script is:

```
> bed2diffs_v1 –bfile MergedVariants –nthreads 8
```

Note: This command will generate an output file named "MergedVariants.diffs".

b. The "EEMS" program requires three input files: i) the "diffs" file generated by the previous step; ii) the "coord" file with sample location represented by longitude and latitude (one sample per line); iii) the "outer" file represents habitat coordinates. Users should prepare the second and the third input files according to their project. The command line for running "EEMS" is:

```
> runeems_snps –params <eems_param>
```

Note: The "eems_param" is the parameter file for running EEMS. Below is an example:

```
datapath = MergedVaraints

mcmcpath = <output directory>

nIndiv = <individual number>

nSites = <SNP site number>

nDemes = 200

diploid = false

numMCMCIter = 2000000

numBurIter = 1000000

numThinIter = 9999
```

Note: The three input files should have the same prefix, e.g., MergedVariants.diffs, MergedVariants.coord, and MergedVaraints.outer. EEMS will read the input files by the "datapath" parameter in the config file.

c. User can visualize the results of the previous step by custom R script with "rEEMSplots" package. In the "eems" folder of PopBoost repository, we provide a visualization R script "plotEEMS.R". The command line for running "plotEEMS.R" is:

```
> Rscript plotEEMS.R <output directory> <figure name>
```

d. Before performing an interpolation analysis, users should prepare an asc-format geographical roster map in the target area. The script also requires a Q matrix file and a coordinate file as input. The coordinate file is the same as the EEMS "coord" file, and the Q matrix is the output of ADMIXTURE analysis. In the "interpolate" folder of the PopBoost repository, we provide an R script "interpolate.R" to perform interpolate analysis and result visualization. The command line for running "interpolate.R" is:

```
> Rscript interpolate.R <asc map> <Q-matrix> <coord file> <outfile>
```

EXPECTED OUTCOMES

The protocol steps described above each yield output files and/or visualizations of the respective results. In the following subsections, the expected outputs and results of every step are described.

Data combination and batch effect evaluation

Data combination and batch effect produce one annotated <.vcf.gz> file and two batch effect measurement <.tsv> table files in *tsv* format. The *vcf* file stores the SNP sites merged from different data sources with site information such as chromosome, position, dbSNP number, and genotypes at each individual. The detailed documentation of the *vcf* format is available at https://samtools.github.io/hts-specs/VCFv4.2.pdf. The first *tsv* file with filename "batch_eval.tsv" contains the evaluation result

Table 1. Batch evaluation and statistic example

Batch evaluation		Site statistics	
Description	Status	Description	Value
# Homozygous Calls	Pass	# Bi-allele SNPs	
# Total Calls	Pass	# Ti sites	
% Heterozygous Calls	Pass	# Tv site	
# Heterozygous Calls	Pass	Ti/Tv ratio	
Gap size in Chrs	Pass	# Calls in dbSNP	
# Multi Calls	Pass	# Calls not in dbSNP	
		% sites in dbSNP	

of the batch effect, in which the first column is key quality metrics, and the second column is measurement results. The key quality metric with no batch effect in the merged *vcf* file will be marked as "PASS". Otherwise, the quality metric will be marked as "Fail". Users could conclude that there are no batch effects if all key quality metrics pass the evaluation. The second *tsv* file "site_statistic.tsv" contains the site statistics of the CDS region, the non-CDS region, and the whole chromosome. Table 1 displays the "batch_eval.tsv" and "site_statictic.tsv" of batch effect evaluation results.

Population structure and migration history analyzes

The analyses take the variation data in *vcf* format and an individual information file as input. The individual information file provides the property, such as population, geographical location, and gender of each individual. An example individual information file is in the root directory of the PopBoost Github Repository folder. Population structure analyses include PCA, ADMIXTURE analysis, and Fst measures. Migration history analyzes include ROH scanning, EEMS, and F_3 statistics, and D statistics estimating. We describe the expected outputs and results for each analysis below.

Kinship analysis generates a coefficient matrix file in the "output/Kinship" folder. The coefficient matrix is a symmetric matrix with both rows and columns representing samples. Each element in the matrix represents the kinship relationship between the two corresponding samples. Note that users should manually remove the related individuals in the input *vcf* file according to the kinship coefficient to ensure that all the samples in the subsequential analyses are unrelated.

Haplotype phasing generates a phased *vcf.gz* file in the "output/Phase" folder. This file has the same format as the input *vcf* file. The difference is that the phased file has a genotype separated by "|", while the token is "/" in the unphased file. The allele type on the same side of "|" belongs to the same haplotype in the phased *vcf* file. A detailed description of the phased *vcf* format is available at https://support.10xgenomics.com/genome-exome/software/pipelines/latest/output/vcf.

PCA generates a separate tabular formatted (*tsv*) file and a set of visualization *pdf* files in the "output/PCA" folder. Each row represents a sample in the *tsv* file, and each column represents a principal component, namely PC1, PC2, and PC3. The visualization of PCA results takes the combination of all two possible principles in PC1, PC2, and PC3 as a two-dimensional coordinate axis. Then we allocate the samples on the axis by their coordinate. Figure 1 is an example of the output of PCA visualization in a worldwide range of populations (Tibetan-Yi Corridor Project, Tibetan Highlanders Project, 1KGP Phase 3, and Simons Genome Diversity Project).

The output of the ADMIXTURE analysis is the Q matrix files in the folder "output/Admixture/<k>/rep<n>.Q". The <k> value represents the different numbers of the ancestry components, and the <n> value represents the number of repetitive experiments. The Q matrix is in tabular format, in which each row represents a sample and each column represents an ancestry component. The column number depends on the <k> value for running ADMIXTURE. Each element represents the ratio of the corresponding sample and the ancestry component. CLUMPAK will select the best Q matrix at each k and provide a visualization of the ancestry components. The outputs are stored in the folder

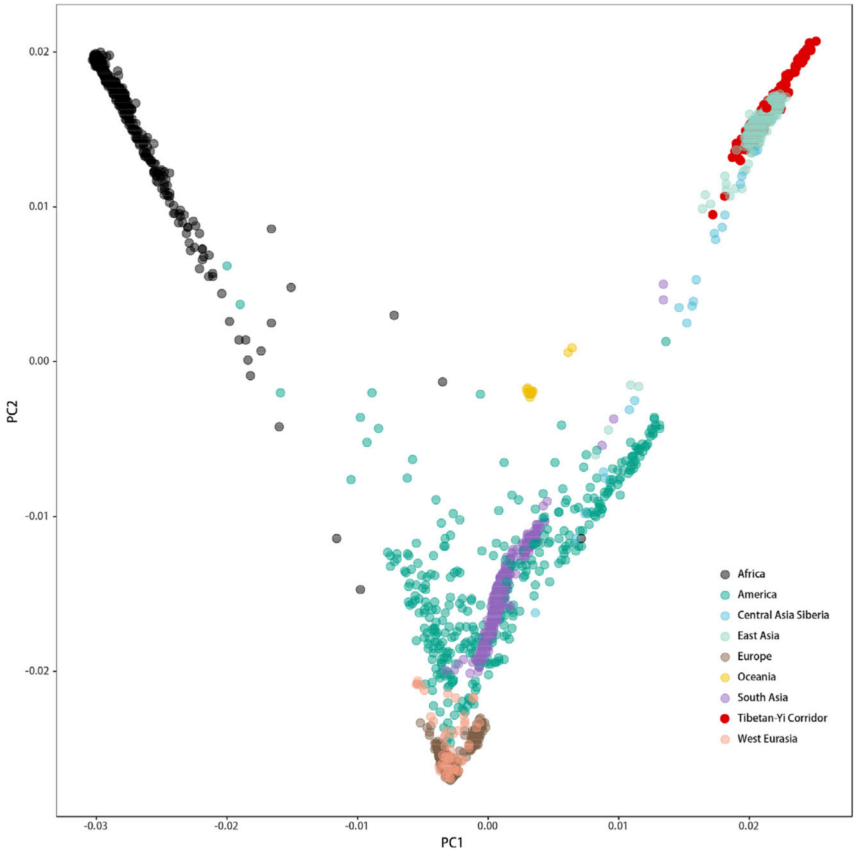


Figure 1. Worldwide PCA from four datasets

PCA shows the patterns of tested populations along the first principle component (PC1) and second principle component (PC2). Each dot in the graph represents an individual and is colored according to ethnic groups. Black is used for African populations, dark green for American populations, blue for Central Asian Siberia populations, light green for East-Asian populations, brown for European populations, yellow for Oceanian populations, purple for South-Asian populations, pink for West-Eurasian populations, and red for Tibetan-Yi corridor populations.

"output/CLUMPAK." The selected Q matrix is the input of Ohana and generates a visualized evolutionary tree for the components of the ancestry. The visualization is in *pdf* format and stored in "output/Ohana." Figure 2 shows the visualization result of high-report individuals worldwide at k=11.

The pairwise F_{st} output is stored in the "output/Fst" folder. The folder has one matrix file in *tsv* format and another visualization file in *pdf* format. The matrix is symmetric in that rows and columns are the input populations. The element in the matrix represents the F_{st} value between the populations represented by the corresponding column and row. The visualization is a heatmap with a hierarchical tree on the top of the heatmap. The tree measures the distance between populations. Figure 3 is an example of the F_{st} visualization in worldwide populations.

The F_3 analysis outputs are stored in the folder "output/F3" with two *txt* files named "outgroup-f3.txt" and "admix-f3.txt". The D statistics analyses output is stored in the "output/Dstatistics" folder with one txt file named "Dstat.txt." The three files are in the same format. Each row represents a three-population test result for F_3 and a four-population test result in D statistics.

The ROH outputs are stored in the folder "output/ROH" with one *txt* file and one visualization file. The *txt* file is in tabular format, in which the first column is the population name.

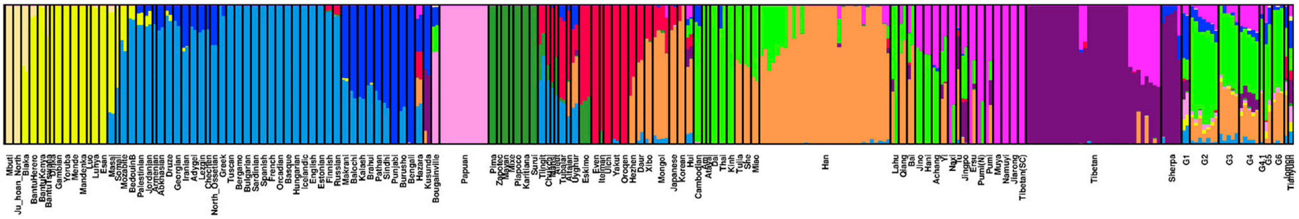


Figure 2. Visualization of ancestral components generated by ADMIXTURE and CLUMPAK with K=11

Different colors represent the identified ancestral components. The figure demonstrated the proportion of ancestral components in each individual.

As each population runs the MSMC program separately, the outputs are in the folders "output/MSMC/<population>." MSMC provides a visualization module that generates a curve plot, in which the x-axis represents the timeline in generation time and the y-axis represents the estimated historical effective population size. Figure 4 shows the historical effective population size of Achang as an example.

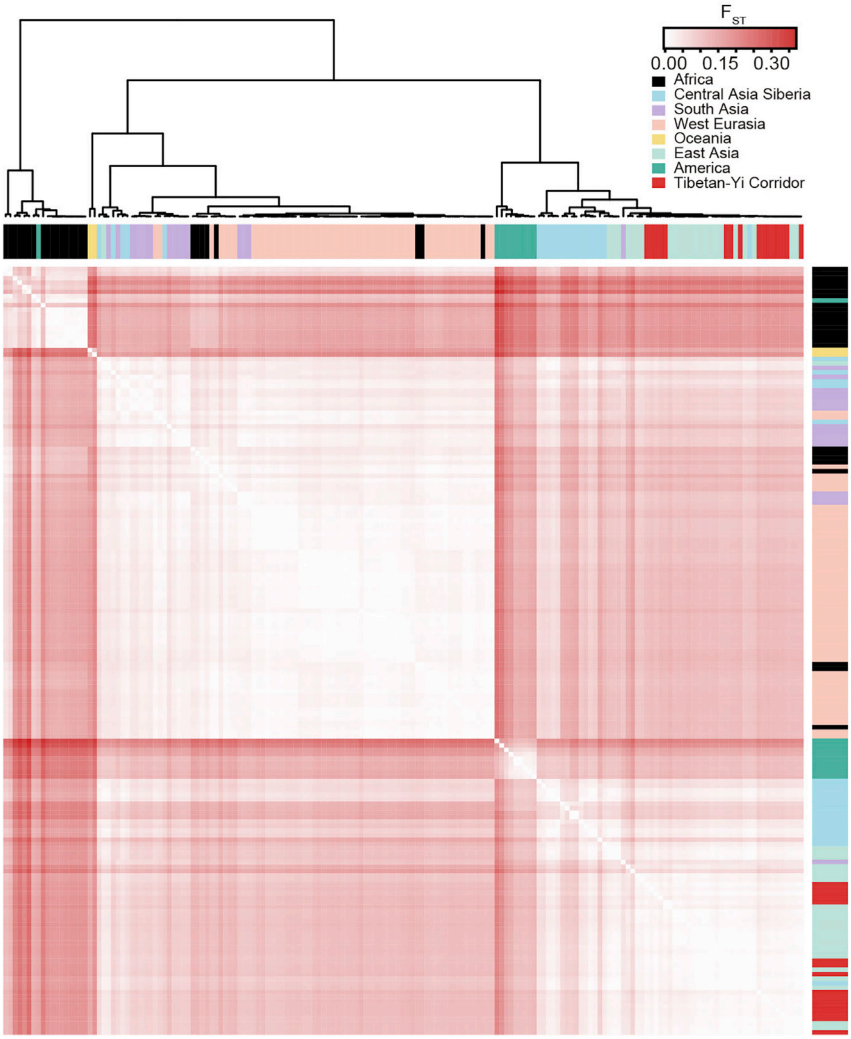


Figure 3. F_{st} heatmap in a worldwide population

The colors in the heatmap represent the pairwise F_{st} values between tested populations. The color bars above and on the right side of the heatmap indicate the individual populations (color same as in Figure 1). The tree on the top of the graph is generated by hierarchical clustering.

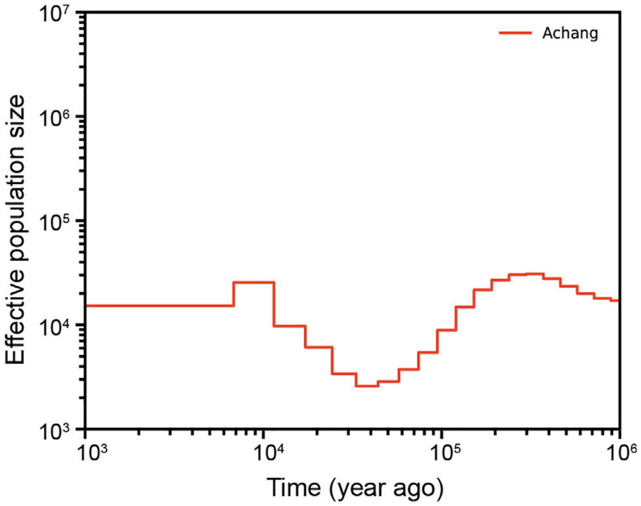


Figure 4. Estimated population size for Achang population in Tibetan-Yi corridor

The x-axis is the historical time and the y-axis is the estimated population size. The red line is the estimated population size along the historical time slots.

The R package "PopGPlot" visualizes the EEMS results, and the visualization output is in the "output/EEMS" folder. The visualization provides effective migration surfaces represented by different colors on the map. Isolation analysis visualizes the ancestor components on a geographic map.

LIMITATIONS

When batch effects are removed from different datasets, the filtering parameters provided in our method are not necessarily suitable for these datasets. We recommend that the filtering parameters provided in the protocol be considered initial conditions when implementing specific projects. Then, the filtering parameters can be adjusted according to the removal of batch effects evaluated by the R package "genotypeeval".

Our method also applies to data from the SNP microarray. However, we do not recommend merging datasets that differ by orders of magnitude. At this point, the SNPs that were ultimately used for analysis depend on the dataset with the fewest number of SNPs. A lot of information will be lost when a whole genome sequencing dataset is integrated with a small SNP array dataset.

TROUBLESHOOTING

Problem 1

No root privileges when installing Python modules (see "before you begin" step 1).

Potential solution

In most situations, the best solution is to rely on the so-called "user site" location by running:

```
> pip install –user module_name
```

Problem 2

Twenty-three genome data in the SGDP cannot be downloaded directly (see "before you begin" step 25).

Potential solution

Please send Swapan Mallick (ude.dravrah.dem.sciteneg@pohs) and David Reich (ude.dravrah.dem.sciteneg@hcier) a signed letter to acquire a password protected link to download these 23 genomes.

Problem 3

The code provided in this method produces error(s) (see "step-by-step method details").

Potential solution

This may be caused by the software and package version. Other versions than the ones indicated in the "key resources table" were not tested. Please check for compatibility between different versions of software and packages. We recommend upgrading or downgrading the required software and packages to the indicated version.

Problem 4

The uploaded ADMIXTURE result cannot be parsed by CLUMPAK server (see step 8).

Potential solution

The CLUMPAK server accept input data from STRUCTURE analysis or ADMIXTURE analysis. Users should indicate ADMIXTURE format before analysis. The uploaded data should in zip format follow the instructions in CLUMPAK server. The Qmatrix files generated by different K values should be separated in different folders in the uploaded zip file.

Problem 5

The interpolate output do not contain all the samples in the visualization (see step 9).

Potential solution

The interpolate analysis visualization use the roster map as the background. The missing sample should be outside the map boundary. The user can check the longitude and latitude of the sample, and consider revise the roster map to a broader region.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and fulfilled by the lead contact, Dr. Shuai Cheng Li (shuaicli@cityu.edu.hk).

Materials availability

This study does not use any materials.

Data and code availability

In this protocal, data from four projects are used: Tibetan-Yi Corridor Project, Tibetan Highlanders Project, Simon Genome Diversity Project, and 1KGP Phase 3. They can be download from GVM: GVM000100 (http://bigd.big.ac.cn/gvm/getProjectDetail?project=GVM000100), GSA: PRJCA000246, Reich Lab (https://sharehost.hms.harvard.edu/genetics/reich_lab/sgdp/vcf_variants/vcfs.variants.public_samples.279samples.tar), and The International Genome Sample Resource (https://www.internationalgenome.org/category/phase-33/), respectively. The accession number for the data reported in this paper is GVM: GVM000100. The code generated during this study is available at https://github.com/zachary-zzc/PopBoost. A DOI can be found at https://doi.org/10.5281/zenodo.7323771.

ACKNOWLEDGMENTS

We wish to thank the participants and their families for their contributions to valuable data.

AUTHOR CONTRIBUTIONS

Conceptualization, S.L.; methodology, Z.Z.C.; software, Z.Z.C., Y.W., Z.Z.; writing – original draft, Z.Z.C., Y.W., S.L.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Zhang, Z., Zhang, Y., Wang, Y., Zhao, Z., Yang, M., Zhang, L., Zhou, B., Xu, B., Zhang, H., Chen, T., et al. (2022). The Tibetan-Yi region is both a corridor and a barrier for human gene flow. Cell Rep. *39*, 110720.

2. Lu, D., Lou, H., Yuan, K., Wang, X., Wang, Y., Zhang, C., Lu, Y., Yang, X., Deng, L., Zhou, Y., et al. (2016). Ancestral origins and genetic history of Tibetan Highlanders. Am. J. Hum. Genet. *99*, 580–594.

3. Mallick, S., Li, H., Lipson, M., Mathieson, I., Gymrek, M., Racimo, F., Zhao, M., Chennagiri, N., Nordenfelt, S., Tandon, A., et al. (2016). The Simons genome diversity project: 300 genomes from 142 diverse populations. Nature *538*, 201–206.

4. 1000 Genomes Project Consortium, Auton, A., Brooks, L.D., Durbin, R.M., Garrison, E.P., Kang, H.M., Korbel, J.O., Marchini, J.L., McCarthy, S., McVean, G.A., and Abecasis, G.R. (2015). A global reference for human genetic variation. Nature *526*, 68–74.

5. DePristo, M.A., Banks, E., Poplin, R., Garimella, K.V., Maguire, J.R., Hartl, C., Philippakis, A.A., del Angel, G., Rivas, M.A., Hanna, M., et al. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat. Genet. *43*, 491–498.

6. Chen, Y., Chen, Y., Shi, C., Huang, Z., Zhang, Y., Li, S., Li, Y., Ye, J., Yu, C., Li, Z., et al. (2018). SOAPnuke: a MapReduce acceleration-supported software for integrated quality control and preprocessing of high-throughput sequencing data. GigaScience *7*, 1–6.

7. Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics *25*, 1754–1760.

8. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R.; 1000 Genome Project Data Processing Subgroup (2009). The sequence alignment/map format and SAMtools. Bioinformatics *25*, 2078–2079.

9. Manichaikul, A., Mychaleckyj, J.C., Rich, S.S., Daly, K., Sale, M., and Chen, W.M. (2010). Robust relationship inference in genome-wide association studies. Bioinformatics *26*, 2867–2873.

10. Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics *27*, 2987–2993.

11. Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A.R., Bender, D., Maller, J., Sklar, P., de Bakker, P.I.W., Daly, M.J., and Sham, P.C. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. Am. J. Hum. Genet. *81*, 559–575.

12. Wang, K., Li, M., and Hakonarson, H. (2010). ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res. *38*, e164.

13. Korneliussen, T.S., Albrechtsen, A., and Nielsen, R. (2014). ANGSD: analysis of next generation sequencing data. BMC Bioinf. *15*, 356.

14. Browning, S.R., and Browning, B.L. (2007). Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. Am. J. Hum. Genet. *81*, 1084–1097.

15. Delaneau, O., Zagury, J.F., and Marchini, J. (2013). Improved whole-chromosome phasing for disease and population genetic studies. Nat. Methods *10*, 5–6.

16. Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. Nat. Genet. *38*, 904–909.

17. Lawson, D.J., Hellenthal, G., Myers, S., and Falush, D. (2012). Inference of population structure using dense haplotype data. PLoS Genet. *8*, e1002453.

18. Alexander, D.H., Novembre, J., and Lange, K. (2009). Fast model-based estimation of ancestry in unrelated individuals. Genome Res. *19*, 1655–1664.

19. Kopelman, N.M., Mayzel, J., Jakobsson, M., Rosenberg, N.A., and Mayrose, I. (2015). Clumpak: a program for identifying clustering modes and packaging population structure inferences across K. Mol. Ecol. Resour. *15*, 1179–1191.

20. Cheng, J.Y., Mailund, T., and Nielsen, R. (2017). Fast admixture analysis and population tree estimation for SNP and NGS data. Bioinformatics *33*, 2148–2155.

21. Petkova, D., Novembre, J., and Stephens, M. (2016). Visualizing spatial population structure with estimated effective migration surfaces. Nat. Genet. *48*, 94–100.

22. Schiffels, S., and Durbin, R. (2014). Inferring human population size and separation history from multiple genome sequences. Nat. Genet. *46*, 919–925.

23. Patterson, N., Moorjani, P., Luo, Y., Mallick, S., Rohland, N., Zhan, Y., Genschoreck, T., Webster, T., and Reich, D. (2012). Ancient admixture in human history. Genetics *192*, 1065–1093.

24. Huber, W., Carey, V.J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B.S., Bravo, H.C., Davis, S., Gatto, L., Girke, T., et al. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nat. Methods *12*, 115–121.

25. Baroud, G., and Steffen, T. (2005). A new cannula to ease cement injection during vertebroplasty. Euro. Spine J. *14*, 474–479.

26. Chang, C.C., Chow, C.C., Tellier, L.C., Vattikuti, S., Purcell, S.M., and Lee, J.J. (2015). Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience *4*, 7.

27. Schiffels, S., and Wang, K. (2020). MSMC and MSMC2: the multiple sequentially markovian coalescent. Methods Mol. Biol. *2090*, 147–166.

28. Wang, J., Santiago, E., and Caballero, A. (2016). Prediction and estimation of effective population size. Heredity *117*, 193–206.