TEAM UBUNTU

Software Documentation

29 February 2024

Camila Ballenghien, Shathavi Krishnan, Haotian Zheng, Aiza Safdar

Introduction

Welcome to the official documentation for the Ubuntu Analysis Tool. This comprehensive guide is designed to provide users with a detailed understanding of the functionalities of our software. The Ubuntu Analysis Tool is a platform tailored to facilitate the analysis of genetic data sourced from diverse human populations. This tool allows users to explore the genetic landscape of these populations, getting insightful information into population structure and genetic diversity.

Our tool is organized into three parts:

(1) A Clustering Analysis section where the user can look at clustering patterns between populations (or superpopulations) of their choice.

(2) An ADMIXTURE analysis where the user can look at ancestral relatedness between populations (or superpopulations).

(3) A section for retrieval of allele and genotype frequencies (and clinical relevance when available) for Single Nucleotide Polymorphisms (SNPs) by selecting either SNP IDs, chromosome coordinates or gene names.

The software is based on a database containing more than 5 million SNPs for chromosome 1. It also holds information on the prevalence of these SNPs for 27 population groups coming from different parts of the world. 26 of these human populations come from whole-genome sequencing data samples from the 1000 Genomes Project Database. The 27th population group (Siberian group) comes from our collaborator, who integrated their genetic data into the samples from the 1000 Genomes Project Database.

The incentive to create this software was to allow our collaborator to compare these Siberian samples to other human populations in their genetics through population structure analyses.

The results of all the different analyses performed by the Ubuntu team are stored in the database which should allow efficient retrieval of information by the users.

Table of Contents

1. Software Overview

1.1 Software Architecture

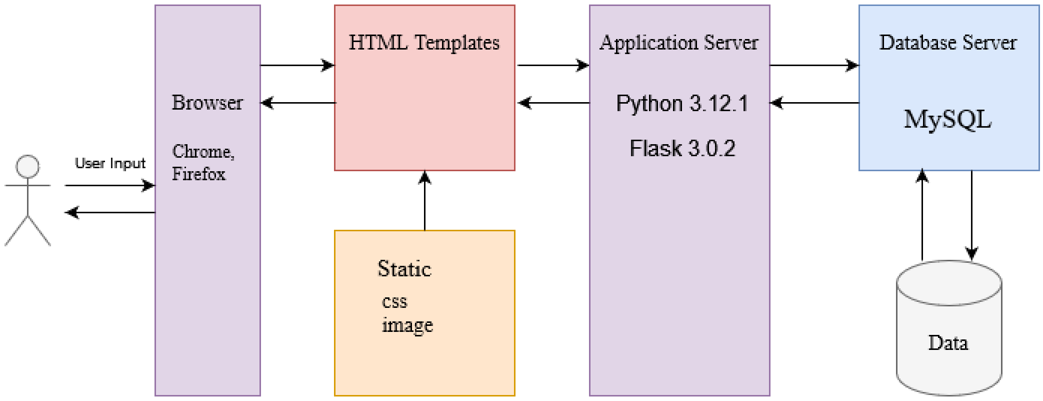


Figure 1. Illustration of the Ubuntu Analysis software architecture.

This software is based on a database built using MySQL 8.0. The software was developed using the Flask package on Python, which communicates with the database server using the MySQL connector, executes the MySQL code, and presents the results on the web using the HTML templates.

1.2 Website Structure

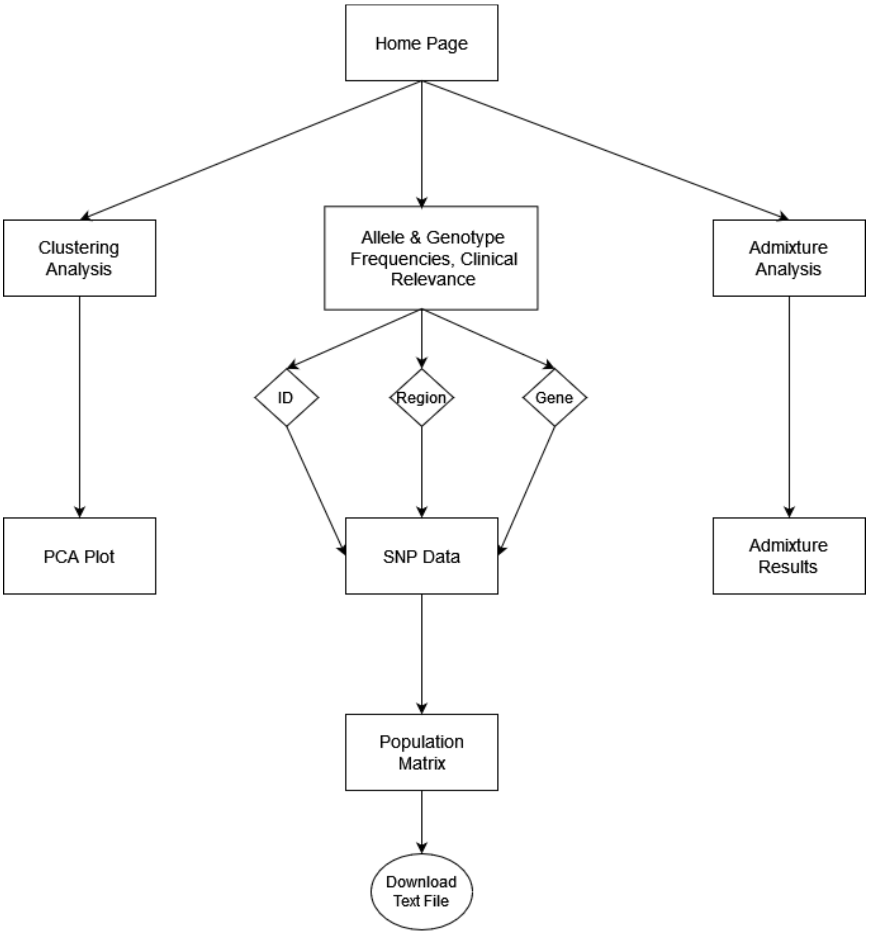


Figure 2. Website structure of Ubuntu Analysis software

The design of the website structure of the Ubuntu Analysis software is based on the functions that the software performs. For each function, there is an information input page and a results display page. Users can navigate to the desired pages by simply clicking on the buttons.

1.3 Running the Ubuntu web application

- Use Python version 3.12.1
- Go to the Ubuntu GitHub (https://github.com/ml22826/Ubuntu) and install the requirements.txt file
- Install MySQL server 8.0
- Open the MySQL command line and create a database called UBUNTU:

```
CREATE DATABASE ubuntu;
```

- Import the SQL database script file (ubuntu.sql) and import this to your database using this command:

```
mysql -u username -p ubuntu < /path/to/your/ubuntu.sql;
```

- Import the Flask directory from the Ubuntu GitHub https://github.com/ml22826/Ubuntu/tree/main/Front%20end, and the value of 'user', 'password', 'host', 'database' in the read_config function in the app.py file has been changed to modify the local machine.
- Running flask:

  After completing the necessary preparations, run the Flask script by going to the Ubuntu_Flask directory on the terminal and executing it using the commands corresponding to the operating system.

  On the Windows System:

```
set FLASK_APP=app.py
set FLASK_DEBUG=1
flask run
```

  On the Linux System:

```
export FLASK_APP=app.py
export FLASK_DEBUG=1
flask run
```

1.4 Software/Package used

1.4.1 Plink v.1.9

Plink is an open-source bioinformatics software tailored for the comprehensive analysis of genetic data (Purcell et al., 2007). Its primary application is in analyzing genotype and phenotype data (Purcell, 2014) Plink supports various data formats, including VCFs, which made it an ideal choice for our project (PLINK 1.90 beta, 2023). There are multiple versions of this software, Plink v.2.0 being the most recent version (PLINK 2.00 alpha, 2024). For this project, Plink v.1.9 was used to calculate the allele frequencies as Plink v.2 0 faced download issues on our machines.

With Plink v.1.9, the MAF (Minor Allele Frequency) was calculated, this indicates at which frequency the less common allele occurs in a population (Chanock & Ostrander, 2014). Therefore, the minor allele can be the alternate allele or the reference allele in the case of SNPs. As a result, more processing was needed to determine the alternate allele frequency and the reference allele frequency (more detail about this in the database section). Plink was further utilised for clustering and admixture analysis due to its ability to handle large VCF files (see analysis section).

1.4.2 MySQL 8.0

MySQL is an open-source relational database management system. It was used for querying and processing data within the Ubuntu database. The choice of MySQL was driven by its speed, reliability and its ability to handle large datasets (Oracle, 2021) These factors made it particularly suitable for our project given that the Ubuntu database includes tables with as many as 150 million rows. Additionally, the integration of MySQL with Flask package was made very easy and efficient using the MySQL connector package on Python. This enabled very straightforward database querying within the Flask environment. Both HeidiSQL and MySQL Workbench were used as graphical interface for the creation and testing of the database.

1.4.3 SnpEff

SnpEff is an open-source Bioinformatics tools that performs annotation on variants and prediction of the effects of genetic variants, such as SNPs (Cingolani, 2023). It provides insight into how these variants can affect gene function. It’s designed to handle large scale genomic datasets, which makes it very suitable for the Ubuntu project. Moreover, SnpEff is user-friendly and has a straightforward download process which set it apart from other annotation tools.

For proper use of SnpEff, certain prerequisites must be met, such as:

- A computer with a Unix-based operating system, such as Ubuntu.
- Java Runtime Environment (JRE) to execute the Java-based software.

SnpEff offers a wide array of databases where the annotation can be made from (Cingolani, 2023). For the Ubuntu project the database chosen was the GRCh38.p7.RefSeq focusing on functional annotations and the genes affected.

1.4.4 PyVCF

PyVCF is a Pyton library designed for parsing and handling VCF files (Casbon, 2012). In this project it was primarily employed to process ClinVar VCF files, connecting various SNPs found in the chr1.vcf.gz with their clinical impact. This was done to get a comprehensive snp_info table containing essential clinical information.

One key feature of PyVCF which was useful in this project was its ability to distinguish between SNPs and non-SNP variants in the ClinVar VCF files (Casbon, 2012).

Incorporating PyVCF was very practical as it integrated well with the Python framework and its various libraries such as pandas, os, etc.

1.4.5 ADMIXTURE

ADMIXTURE is a software tool which is specifically designed for the maximum likelihood estimation for individual ancestries (Alexander et al, 2009). In this project we used the ADMIXTURE software to run the admixture analysis. This analytical process was carried out to obtain the ancestral proportions for each population.

1.4.6 Flask

Flask is a module in Python that provides a lightweight web framework for developing web applications based on the Werkzeg WSGI toolkit and the Jinja2 template engine. The reason for choosing Flask is that learning Flask and starting a web application project with Flask should be a straightforward process for developers with prior experience in Python.

The Ubuntu web application is developed in Python 3.12.1 using Flask 3.0.2 (available at https://flask.palletsprojects.com/en/3.0.x/). The following packages are used in the Flask file: Flask-WTF 1.2.1 (https://flask-wtf.readthedocs.io/en/1.2.x/), mysql-connector-python 8.3.0 (https://dev.mysql.com/downloads/connector/python/?os=31), Pandas 2.2.0 (https://pandas.pydata.org/), Matplotlib 3.8.2 (https://matplotlib.org/), NumPy 1.26.3 (https://numpy.org/), Seaborn 0.13.2 (https://seaborn.pydata.org/), and Beautifulsoup4 4.12.2 (https://pypi.org/project/beautifulsoup4/). Flask-WTF is used to construct a secure form using a Cross Site Request Forgery (CSRF) token that prevents CSRF attacks. The web application uses the form to pass the information entered by the user. The Flask connects to the MySQL database using mysql-connector-python. Pandas is used to manipulate the data retrieved from the database conveniently. The PCA plot and admixture results are generated using Matplotlib, while the HeatMap for the population matrix is visualised using Seaborn. The population matrix is generated using NumPy, while the text for the population matrix is generated using Beautifulsoup4. Standard Python library packages are also used to build this application, such as io and base64, which pass images from Flask to HTML.

1.5 HTML/CSS

Hypertext Markup Language (HTML) defines the structure and content of the web, while Cascading Style Sheets (CSS) control the appearance, formatting and layout of web pages. The Ubuntu team created a design CSS file and 8 HTML templates for each page to optimise performance.

1.6 Ubuntu database

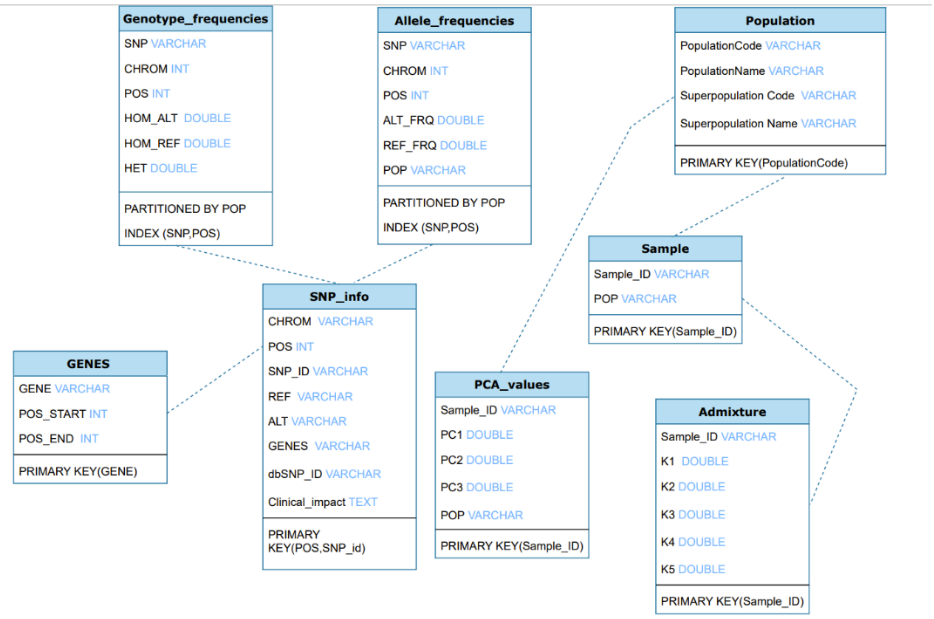


Figure 3: Database schema showing the relationships between tables

The schema depicted by Figure 3 indicates relationships between the tables of the database. The ‘SNP_info’ table is the central hub for SNP related information. It is linked to the genes table, genotype and allele frequencies table. The ‘sample’ table is also central when it comes to linking analysis information (PCA and admixture) tables with the corresponding superpopulation group. It is linked to the ‘PCA_values’, ’Admixture’ and ‘Population’ table.

Database schema

1 - Information on the different tables and sourcing the data

Table 1. Summarising the tables of the database

This table below summarizes each table present in the database and how the data was sourced.

Table name	Description	Source
SNP_info	Contains information for each SNP including position, clinical impact, and affected genes.	SNP information sourced from chr1.vcf.gz. Clinical impact data obtained from GWAS and ClinVar and gene names obtained using SnpEff
Sample	Identifier for each sample among 3928 individuals and their associated population code.	Internal file (sample_pop.tsv)
Population	Information about each population code given in the sample_pop.tsv such as population name, superpopulation code and superpopulation name.	Superpopulation information was sourced from the 1000 genomes project
Genes	Information on start and end position for each gene.	Sourced from the SNP_info table
Allele frequencies	Allele frequencies for each SNP calculated across all the populations	Created using plink v.1.9
Genotype frequencies	Genotype frequencies for each SNP calculated across all the populations	Genotype frequencies were calculated using the Hardy Weinberg principle (ref)
PCA_values	Principal component analysis (PCA) values for each sample.	
Admixture	Ancestral information for each population group	Created using the ADMIXTURE software

This table above summarizes each table present in the database and how the data was sourced.

The ‘Sample’ and ‘Population’ table sourcing was quite straightforward. The ‘Sample’ table was imported from the SharePoint provided by Dr. Fumagalli, and it was directly uploaded onto the database.

The ‘Population’ table’s data was sourced from the 1000 Genomes Project database (The 1000 Genomes Project Consortium, 2015). Minor preprocessing was performed on the data before uploading it onto the database. The ‘PCA_values’ and ‘Admixture’ data came directly from our own analyses.

On the other hand, the ‘SNP_info table’ required several processing steps before obtaining the final version of the table. This table originally came from the chr1.vcf.gz file obtained on Dr. Fumagalli’s Sharepoint. This file was annotated using SnpEff software which added the corresponding gene names to each SNP. Afterwards, the dbSNP IDs were aligned with their respective SNPs from the initial file by coordinating them based on position, reference allele, and alternate allele (www.ncbi.nlm.nih.gov, n.d.). Finally, information about clinical relevance was also integrated using GWAS dabatase (www.ebi.ac.uk, n.d.) and using the ClinVar database (ftp.ncbi.nlm.nih.gov, n.d.).

For the allele frequencies table, the first step was to generate a basic allele frequency report using Plink v.1.9. One report per population group was generated, therefore a total of 27 basic allele frequency report were obtained. Each frequency report gave the Minor Allele Frequency score.

The organization of the basic allele frequency report is displayed below (Table 2)

Table 2: Demonstration of the different columns in the basic allele frequency report generated by Plink.v.1.9. (Chang, 2024)

CHR	**SNP**	**A1**	**A2**	**MAF**
Chromosome code	SNP ID	Allele 1 (usually minor)	Allele 2 (usually major)	Allele 1 frequency

As demonstrated on Table 2, the MAF score represents the frequency of allele 1 which is usually the minor allele frequency but there are no specifications whether A1 is the alternate allele or reference allele frequency. Therefore, for this project the basic allele frequency report wasn’t sufficient, more processing was necessary to get the alternate and reference allele frequency. As a result, a Python script was developed which iterated over each of the 27 basic allele frequency report. This script checked for each SNP if A1 represented the alternate allele or the reference allele, and according to that information, it generated an alternate allele frequency score (it took into account if a score was NA). This allowed to generate a final file containing the alternate allele and reference allele frequencies for each SNP per population group.

This final file was also used to calculate the genotype frequencies. The Hardy-Weinberg equation (depicted blow) was applied to the latter which resulted in predicted genotype frequencies for each SNP per population group.

$$p^2 + 2pq + q^2 = 1$$

Hardy-Weinberg equation (Nature Education, 2014)

p^2 ___~~i~~s the dominant homozygous frequency

q^2 ___is the recessive homozygous frequency.

$2pq$ _____is _the heterozygous frequency.

$p\ and\ q$ both representing the allele frequencies.

Using the Hardy-Weinberg principle to calculate the genotype frequencies offered multiple advantages. First of all, it's a straightforward equation to apply, therefore it wasn't too computationally demanding. Moreover, it provides theoretical information about genotype frequencies under certain conditions.

However, there are limitations to using this equation. There are multiple requirements that need to be fulfilled, such as the assumptions of random mating, infinite population size, no mutation, no migration, and no natural selection (Andrews, 2010). As a result, using Hardy-Weinberg equation might not offer the full picture of the genotype frequencies for our project. Indeed, these conditions are rarely (if ever) met in natural populations (Nature Education, 2014b), making It challenging to apply this equation without considering the potential for discrepancy between predicted frequencies and observed frequencies.

An interesting approach would be to calculate observed genotype frequencies and compare them to the predicted frequencies. This comparative analysis could offer valuable insights into any divergence from expected patterns.

2- Sorting the data/Handling the data (as there is a lot of it)

The Ubuntu database contains a very large amount of data, therefore optimizing the design of the tables and the database was crucial.

For rapid and efficient querying of the data multiple approaches were taken such as adding primary keys in most tables, indexing some columns and partitioning.

The 3 largest tables are SNP_info, allele_frequencies and genotype_frequencies.

SNP_info contains 5 million rows, therefore, to ensure efficient querying of the data, two columns were primary keyed, and one column was indexed.

Allele_frequencies and Genotype_frequencies both contain more than 130 million rows. Therefore, to improve query performance, a list partitioning approach was used for these two tables (Oracle, 2024). They were partitioned according to their population group (population code). This method facilitated the division of these tables into smaller and more manageable sub-tables, which significantly increased the speed of queries.

2. Ubuntu Website Features

2.1 Homepage

Type http://127.0.0.1:5000 to go to a new page in a web browser once the script has been successfully executed. The page created is the home page of the web application.

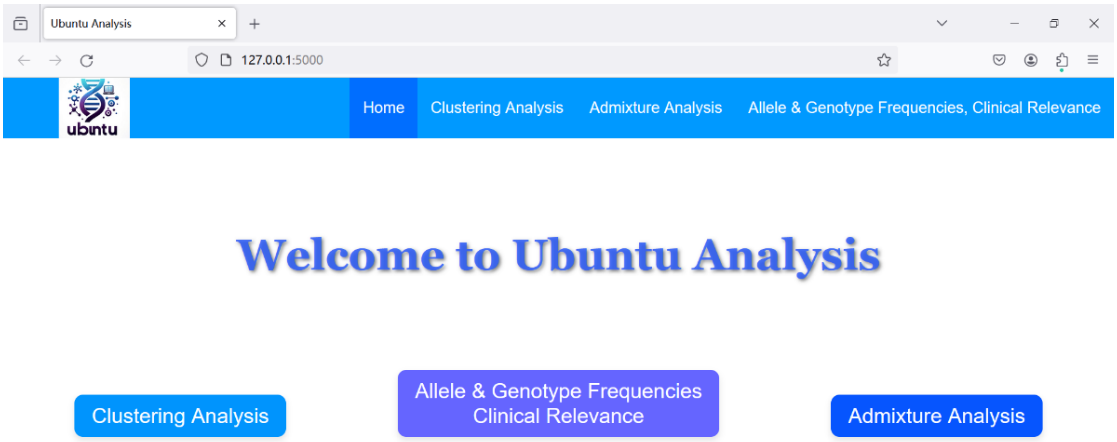
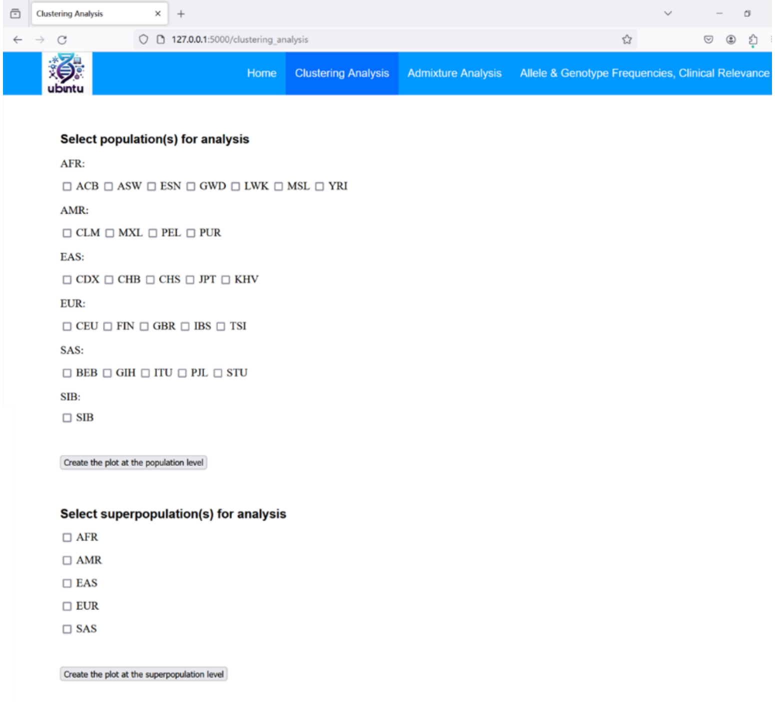

Ubuntu Analysis
127.0.0.1:5000
ubuntu
Home
Clustering Analysis
Admixture Analysis
Allele & Genotype Frequencies, Clinical Relevance
Welcome to Ubuntu Analysis
Clustering Analysis
Allele & Genotype Frequencies Clinical Relevance
Admixture Analysis

Figure 4. Homepage

As illustrated in Figure 4, the homepage is designed with a navigation bar and three buttons, each corresponding to one of the provided functions. Users can navigate the functional pages by clicking the button or the navigation bar.

2.2 Clustering Analysis page


Clustering Analysis
127.0.0.1:5000/clustering_analysis
ubuntu
Home
Clustering Analysis
Admixture Analysis
Allele & Genotype Frequencies, Clinical Relevance

Select population(s) for analysis

AFR:

□ ACB □ ASW □ ESN □ GWD □ LWK □ MSL □ YRI

AMR:

□ CLM □ MXL □ PEL □ PUR

EAS:

□ CDX □ CHB □ CHS □ JPT □ KHV

EUR:

□ CEU □ FIN □ GBR □ IBS □ TSI

SAS:

□ BEB □ GIH □ ITU □ PJL □ STU

SIB:

□ SIB

Create the plot at the population level

Select superpopulation(s) for analysis

□ AFR

□ AMR

□ EAS

□ EUR

□ SAS

Create the plot at the superpopulation level

Figure 5. Clustering Analysis page

The clustering analysis page will be navigated by clicking the “Clustering Analysis” button, as shown in Figure 5. Users can choose which populations or superpopulations that they are interested in by checking on the box in front of the relative population or superpopulation names. Once selected, by simply clicking on the button corresponding to the level at which the plot will be generated, the users will be taken to the results display page, as shown in Figure 6.

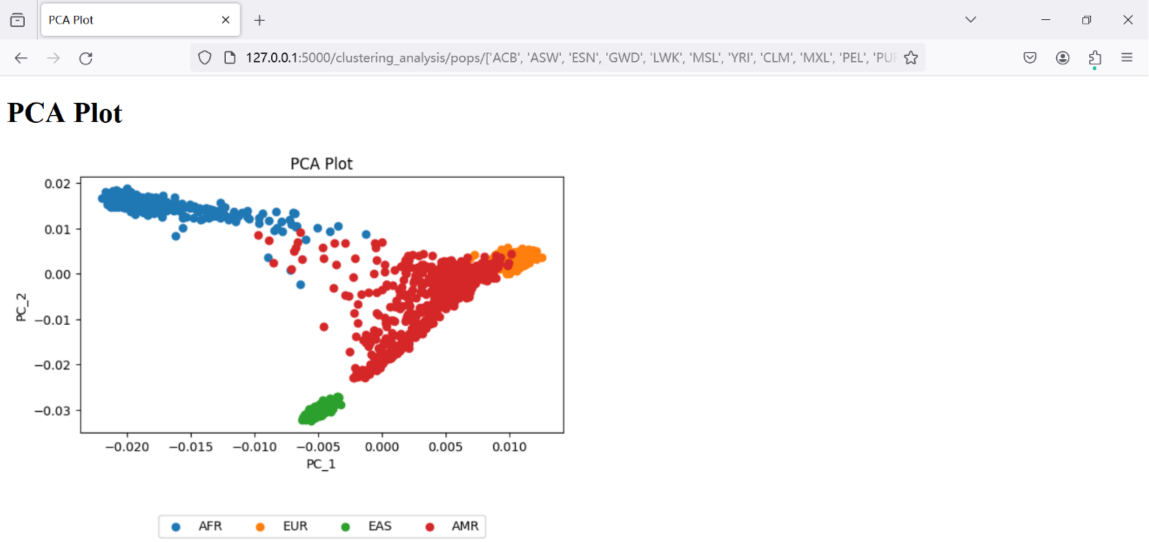
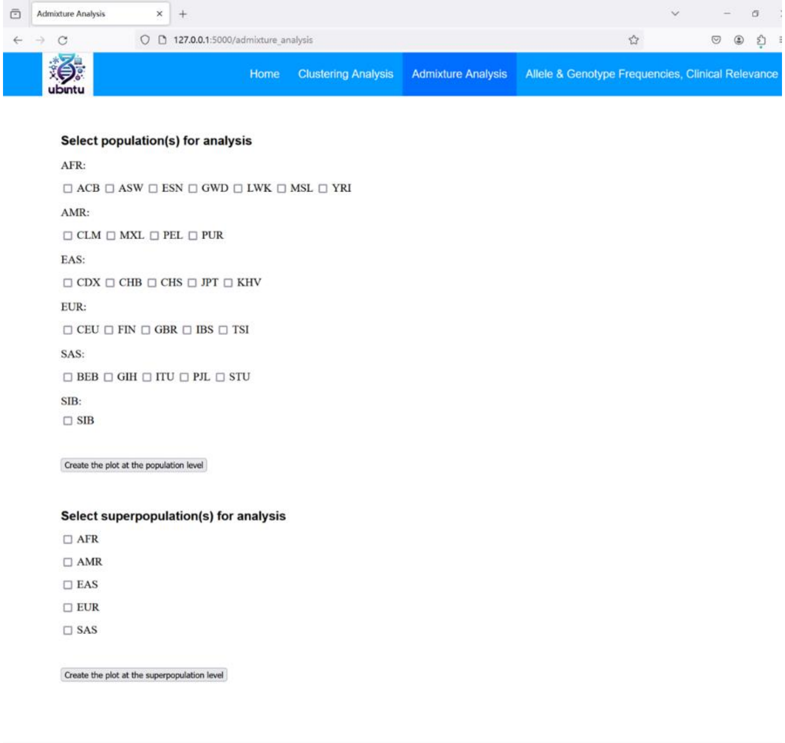

PCA Plot
127.0.0.1:5000/clustering_analysis/pops/['ACB', 'ASW', 'ESN', 'GWD', 'LWK', 'MSL', 'YRI', 'CLM', 'MXL', 'PEL', 'PU
PCA Plot
PCA Plot
0.02
0.01
0.00
−0.01
−0.02
−0.03
PC_2
−0.020
−0.015
−0.010
−0.005
0.000
0.005
0.010
PC_1
AFR
EUR
EAS
AMR

Figure 6. PCA Plot page

2.3 Admixture Analysis page



Admixture Analysis
127.0.0.1:5000/admixture_analysis
ubuntu
Home
Clustering Analysis
Admixture Analysis
Allele & Genotype Frequencies, Clinical Relevance

Select population(s) for analysis

AFR:

□ ACB □ ASW □ ESN □ GWD □ LWK □ MSL □ YRI

AMR:

□ CLM □ MXL □ PEL □ PUR

EAS:

□ CDX □ CHB □ CHS □ JPT □ KHV

EUR:

□ CEU □ FIN □ GBR □ IBS □ TSI

SAS:

□ BEB □ GIH □ ITU □ PJL □ STU

SIB:

□ SIB

Create the plot at the population level

Select superpopulation(s) for analysis

□ AFR

□ AMR

□ EAS

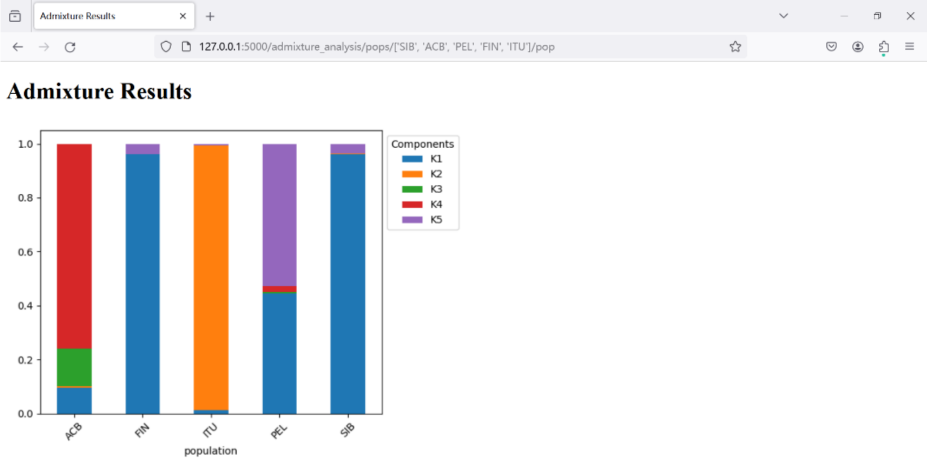
□ EUR

□ SAS

Create the plot at the superpopulation level

Figure 7. Admixture Analysis page

Similar to the clustering analysis page, the admixture analysis page allows the user to choose which populations or superpopulations to incorporate into the results. To initiate the admixture analysis, simply click the corresponding icon. It will be redirected to the admixture results page. When the populations ACB, FIN, ITU, PEL, and SIB are selected and the population level plot button is chosen, the results appear on the page depicted in Figure 8.

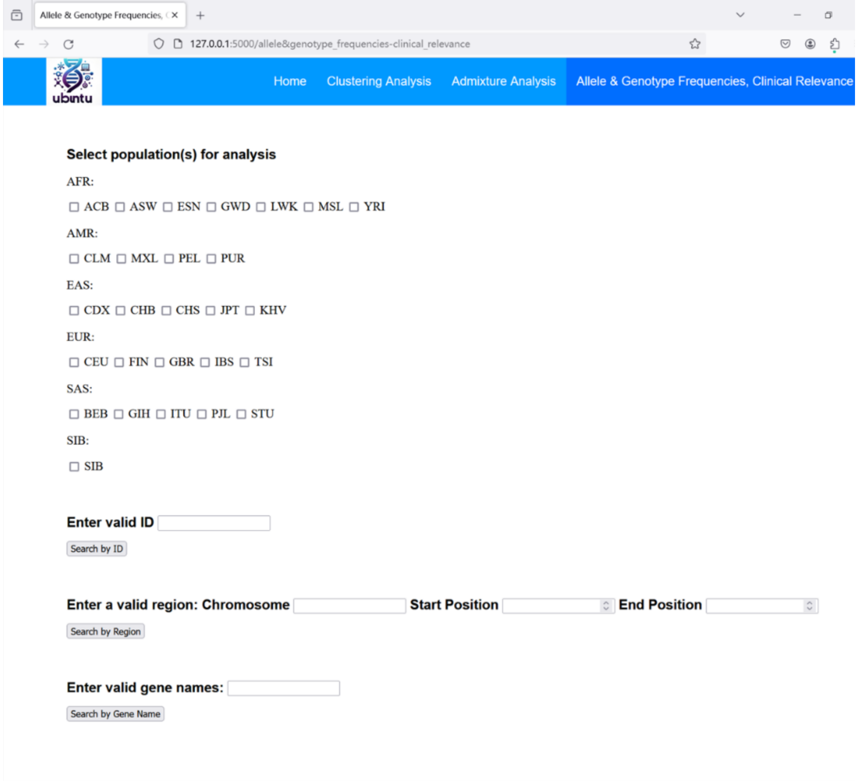


Admixture Results
127.0.0.1:5000/admixture_analysis/pops/['SIB', 'ACB', 'PEL', 'FIN', 'ITU']/pop

Admixture Results

Figure 8. Admixture Analysis page for selecting populations ACB, FIN, ITU, PEL, and SIB

2.4 Allele & Genotype Frequencies, Clinical Relevance Retrieve page


Allele & Genotype Frequencies, C
127.0.0.1:5000/allele&genotype_frequencies-clinical_relevance
ubıntu
Home
Clustering Analysis
Admixture Analysis
Allele & Genotype Frequencies, Clinical Relevance

Select population(s) for analysis

AFR:

□ ACB □ ASW □ ESN □ GWD □ LWK □ MSL □ YRI

AMR:

□ CLM □ MXL □ PEL □ PUR

EAS:

□ CDX □ CHB □ CHS □ JPT □ KHV

EUR:

□ CEU □ FIN □ GBR □ IBS □ TSI

SAS:

□ BEB □ GIH □ ITU □ PJL □ STU

SIB:

□ SIB

Enter valid ID

Search by ID

Enter a valid region: Chromosome **Start Position** **End Position**

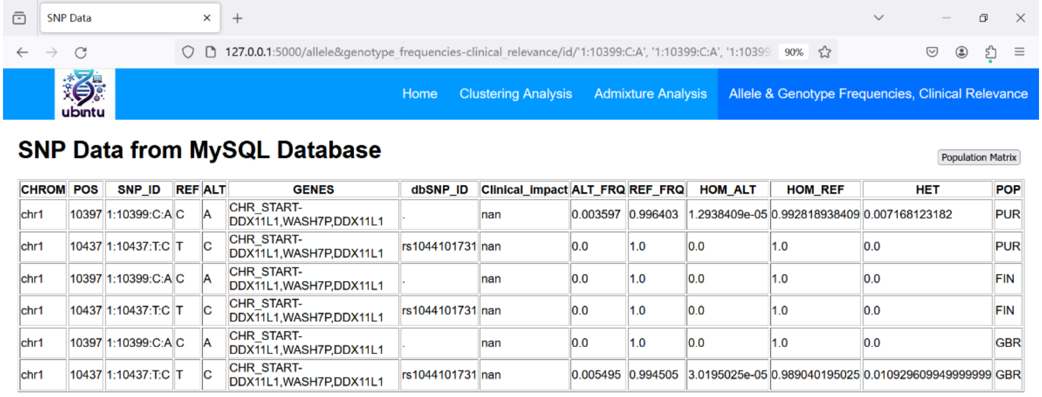
Search by Region

Enter valid gene names:

Search by Gene Name

Figure 9. Allele & Genotype Frequencies, Clinical Relevance Retrieve page

As depicted in Figure 9, the Allele & Genotype Frequencies, Clinical Relevance Retrieve page allows users to select which populations to include in the retrieval process. The three search bars are designed for three approaches to data retrieval. In the Region search bar, users are needed to input the chromosome name (in the format 'chr1') and two valid integers indicating the starting and ending points of the region they wish to seek. Users can also retrieve results by entering multiple valid ID or gene names in the ID or Gene Name search bars, which should be separated by any special character except ':'. For example, By selecting population GBR, FIN, and PUR and inputting the following values into the ID search bar: "'1:10399:C:A', '1:10399:C:A', '1:10399:C:A', '1:10437:T:C', '1:10437:T:C', '1:10437:T:C'" will result in the display of the page illustrated in Figure 10.

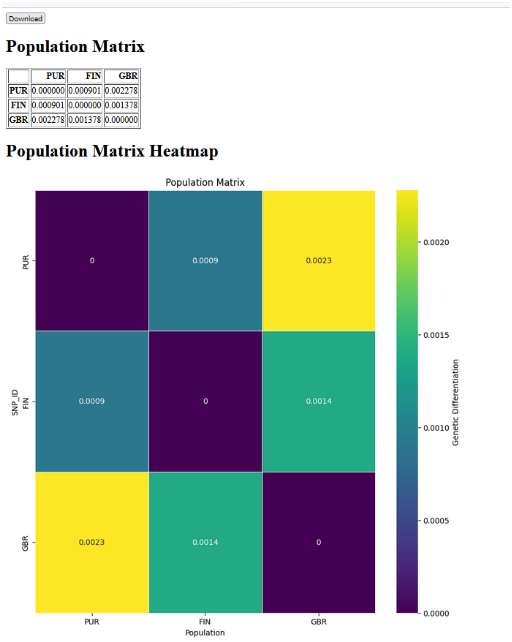

SNP Data
127.0.0.1:5000/allele&genotype_frequencies-clinical_relevance/id/'1:10399:C:A', '1:10399:C:A', '1:10399
90%
ubıntu
Home
Clustering Analysis
Admixture Analysis
Allele & Genotype Frequencies, Clinical Relevance
Population Matrix

SNP Data from MySQL Database

CHROM	POS	SNP_ID	REF	ALT	GENES	dbSNP_ID	Clinical_impact	ALT_FRQ	REF_FRQ	HOM_ALT	HOM_REF	HET	POP
chr1	10397	1:10399:C:A	C	A	CHR_START-DDX11L1,WASH7P,DDX11L1	.	nan	0.003597	0.996403	1.2938409e-05	0.992818938409	0.007168123182	PUR
chr1	10437	1:10437:T:C	T	C	CHR_START-DDX11L1,WASH7P,DDX11L1	rs1044101731	nan	0.0	1.0	0.0	1.0	0.0	PUR
chr1	10397	1:10399:C:A	C	A	CHR_START-DDX11L1,WASH7P,DDX11L1	.	nan	0.0	1.0	0.0	1.0	0.0	FIN
chr1	10437	1:10437:T:C	T	C	CHR_START-DDX11L1,WASH7P,DDX11L1	rs1044101731	nan	0.0	1.0	0.0	1.0	0.0	FIN
chr1	10397	1:10399:C:A	C	A	CHR_START-DDX11L1,WASH7P,DDX11L1	.	nan	0.0	1.0	0.0	1.0	0.0	GBR
chr1	10437	1:10437:T:C	T	C	CHR_START-DDX11L1,WASH7P,DDX11L1	rs1044101731	nan	0.005495	0.994505	3.0195025e-05	0.989040195025	0.010929609949999999	GBR

Figure 10. Example for searching by ID

Then, clicking on the population matrix button will display a heatmap and population matrix as shown in Figure 11.



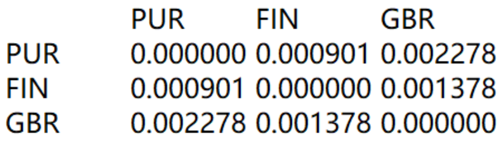
Population Matrix

	PUR	FIN	GBR
PUR	0.000000	0.000901	0.002278
FIN	0.000901	0.000000	0.001378
GBR	0.002278	0.001378	0.000000

Population Matrix Heatmap

Figure 11. Example for population matrix and heatmap

Finally, clicking the Download button will download a text file containing the population matrix to the user's local computer.



```
          PUR      FIN      GBR
PUR       0.000000 0.000901 0.002278
FIN       0.000901 0.000000 0.001378
GBR       0.002278 0.001378 0.000000
```

Figure 12. Example of text file for population matrix

3. Clustering

For our web application, we used clustering analysis to explore population structure among our populations. Principle component analysis (PCA) was our primary technique for this. PCA is a statistical technique that is primarily used to reduce the dimensionality of large datasets while preserving as much variability as possible (Ding et al, 2004). PCA does this by transforming large datasets into new coordinate systems, in which the first coordinate is aligned in the direction of the greatest variance of the data. The subsequent coordinates are then orthogonally placed to the previous ones and are aligned with the next highest variance (Follow S. 2022).

The process provides a way to catch the most significant patterns in the data with fewer dimensions.

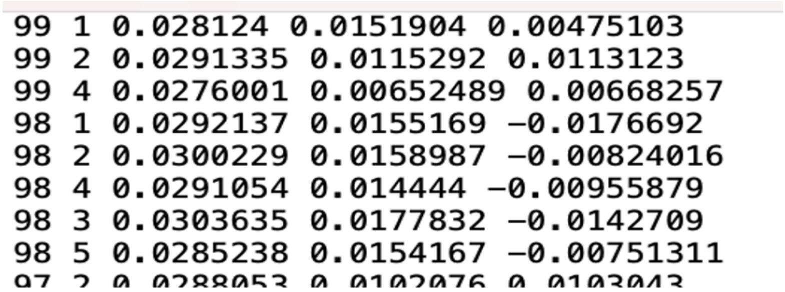
We choose PCA analysis to explore population structure because it is a powerful tool that allows the visualisation of high dimensional data. The use of PCA can reveal patterns in the data which corresponds to population structure. The principle component scores can be used to determine how the samples or populations are related. If two samples or populations have a similar PC scores it may suggest that they share similar genetic makeup and therefore, are more closely related.

3.1 Methodology

Before the PC scores were calculated the VCF file was converted into binary files using Plink 1.9v. This conversion was required because we intended to generate PC scores for our data using one of Plink’s functionality. Plink specifically requires the input file to be in binary format in order to utilise the -PC function for PCA (Dutheil et al,2021).

We used Plink to compute 3 PC scores for each sample. This is because choosing 3 PCs can make the visualisation of the PCA plot more manageable and clear. Including more PC scores could complicate the interpretation and therefore, lead to a lower variance in the plotted data, making it less informative and harder to understand.

After using plink to calculate the PC scores for each sample, an eigenvec file was obtained.

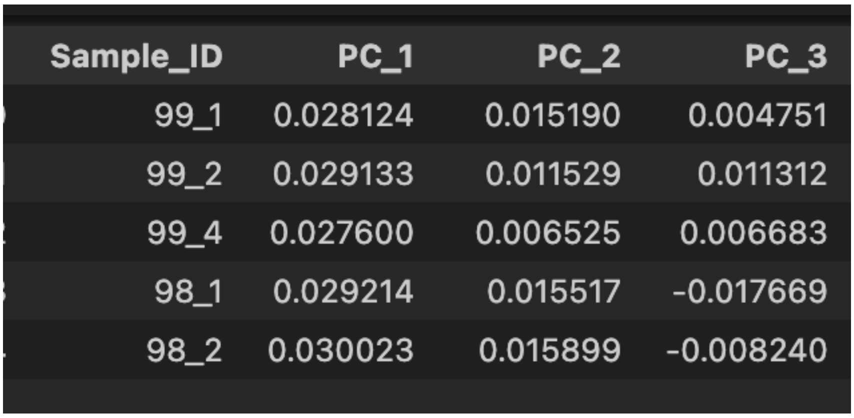


```
99 1 0.028124 0.0151904 0.00475103
99 2 0.0291335 0.0115292 0.0113123
99 4 0.0276001 0.00652489 0.00668257
98 1 0.0292137 0.0155169 -0.0176692
98 2 0.0300229 0.0158987 -0.00824016
98 4 0.0291054 0.014444 -0.00955879
98 3 0.0303635 0.0177832 -0.0142709
98 5 0.0285238 0.0154167 -0.00751311
97 2 0.0288053 0.0102076 0.0103043
```

Figure 13. The Eigenvec file

This file doesn't have any columns and the sample IDs for the Siberian population do not align with those found in the VCF file and the database (Chang C. 2024). This discrepancy is likely due to Plink’s handling of character delimiters. This could present an issue when the table is added to the database as the Siberian ids from the PC table may not link correctly to the rest of the database.

To resolve this, we used python's pandas library to convert the sample ids found in the eigenvec file to match the sample ids present in the database. The apply function was used to concatenate the 1st and 2nd column with an underscore for each row of the Siberian population and the 2nd column was dropped. This created the sample Ids which could be linked to the database.



Sample_ID	PC_1	PC_2	PC_3
99_1	0.028124	0.015190	0.004751
99_2	0.029133	0.011529	0.011312
99_4	0.027600	0.006525	0.006683
98_1	0.029214	0.015517	-0.017669
98_2	0.030023	0.015899	-0.008240

Figure 14. The modified PC scores table

The Matplotlib package was used to generate a PCA plot. A scatter plot of PC1 and PC2 was produced. We decided to only use PC 1 and 2 because the third PC didn't represent much of the data and a plot that only plotted PC1 and PC2 would be easier to interpret compared to three dimensional plot. In this plot each sample was individually represented. To view population structure, we colour coded each sample according to its population. This approach enables us to observe potential clustering patterns and observe population structure.

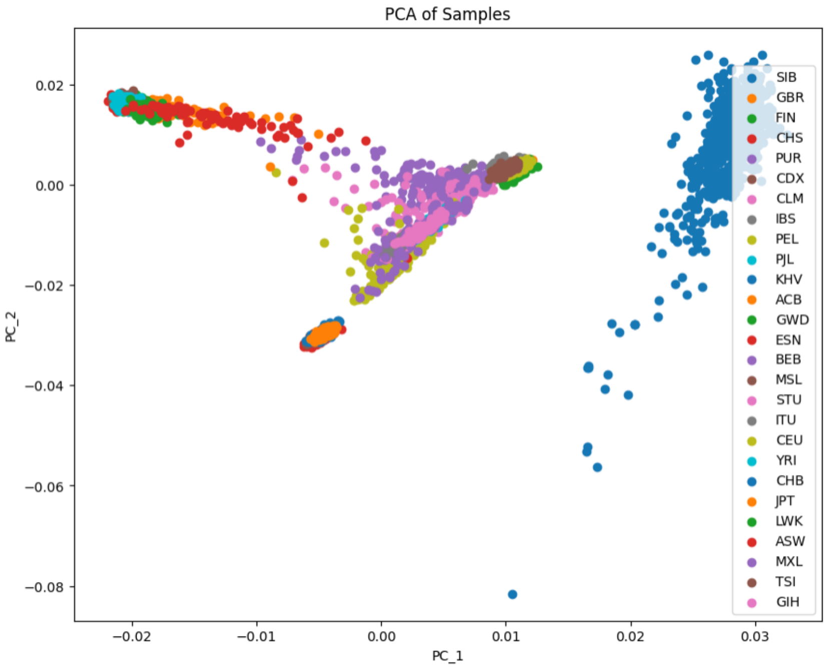


Figure 15. The PCA Plot.

The plot shows the different populations clustering together, suggesting that they may have similar genetic makeup and may be related together.

3.2 Limitations:

In our analysis we did not prune SNPs with significant linkage disequilibrium (LD). The VCF file used for this analysis could have obtained a high linkage disequilibrium. This risks the PCA analysis capturing LD patterns over the population structure, which could suggest we have misinterpreted the data. To improve our PCA analysis we could prune SNPs with high LD using the plink software (malomane et al, 2021). In addition, the PC scores capture only little bit of the data perhaps, other clustering techniques such as UMAP can be used in combination with the PCA to give a more comparative analysis.

4. Admixture analysis:

Admixture occurs when isolated populations start to inbreed resulting in their offspring obtaining a mixture of alleles from different ancestral populations (Albrechtsen et al, 2013). Admixture analysis can be used to determine the ancestry proportions from various ancestral population in groups of individuals. The ubuntu database aims to allow the user to view the proportions of ancestry across different populations.

4.1 Admixture Software

There are several software tools are available to conduct admixture analysis, which include ADMIXTURE, STRUCTURE and fastSTRUCTURE. In our analysis for the UBUNTU website, we opted to use the ADMIXTURE software due to it's notable computational efficiency and speed, which are essential for handling a large dataset that contains over 5 million SNPs. While ADMIXTURE and STURUCTRE share a similar statistical model, ADMIXTURE calculates the estimates more efficiently. This is due to the use of the block relaxation approach, which alternates between updating allele frequency and ancestry fraction parameters (Alexander et al, 2009). Each block update involves solving numerous independent convex problems with the use fast sequential quadratic programming methods, that are further accelerated by a novel quasi-Newton acceleration method (Pritchard et al, 2000). This approach significantly outperforms other methods such as Expectation-Maximisation (EM) algorithms and Markoc Chain Monte Carlo sampling that are used by STRUCUTRE and fastSTURUCTRE (Pritchard et al, 2000). Given the size of our dataset, the efficiency of ADMIXTURE will be beneficial in running our admixture analysis.

4.2 Methodology

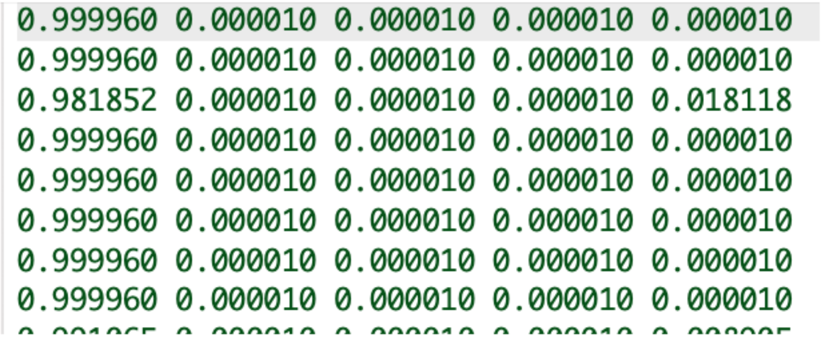
Before we could run the admixture analysis, we first converted the VCF file to a bed file using Plink 1.9v. We did this because the ADMIXTURE software was only able to accept three formats of files (Alexander et al, 2011): binary (.bed), ordinary (.ped) and EIGENSTRAT (.geno). Out of these file formats we decided to use the binary (.bed) file as compared to the other text files, binary files are more space efficient and therefore, are able to represent data in a more compact and optimised format (Cog-genomics.org). This is beneficial for handling large datasets due to the reduced storage requirements and faster data processing. In addition, since admixture is a time-consuming process opting for a binary file instead of the text file proved to be more advantageous as binary files can be read and written faster than text files, which is especially beneficial when having a large file. Furthermore, utilising a bed file was beneficial as these files are compatible with a large range of software tools and therefore, could be reused for other analysis downstream.

We ran ADMIXTURE on a high-performance computer called apocrita. This is because we initially faced difficulties running AMIXTURE on our local machines due to insufficient amount of memory and CPUs on our local machine. To run ADMIXTURE on apocrita we first had to install the ADMIXTURE software on to the apocrita space. This was done by downloading the software form the ADMIXTURE website (Alexander et al, 2011) and transferring the software from the local machine to the appropriate space using the secure copy protocol command.

We then created a job script to run the ADMIXTURE analysis, in the job script we requested 16 CPU cores, 96GB of memory and a runtime of 10 days. These choices were made considering the computational intensity of the ADMIXTURE process. Increasing the number of CPU cores can enhance the speed of the analysis which can be beneficial for large datasets as they require more time. Furthermore, using 96GB of memory ensured that there was enough space to efficiently handle and manipulate the large input file with the ADMIXTURE software. Additionally, a runtime of 10 days was chosen to allow ADMIXTURE to run successfully without any interruptions or premature terminations.

We decided to run the admixture analysis in a loop where K values ranged from 3 to 5. We decided to start off using a smaller K values as the analysis would be faster. In addition, having a higher K value can sometimes lead to overfitting, where the model can become more complex and potentially start fitting noise in the data rather than capturing population structure (Alexander, Lange, 2011). After the analysis completed, we decided to use the results from K5 to continue the admixture analysis. This decision was made based on the observation that the cross-validation error was lowest at K5 which suggests that K5 had the best predictive accuracy (Alexander et al, 2011). Furthermore, the VCF file mentioned that there were 5 superpopulations in the data, therefore selecting K5 would be consistent with the underlying structure we expected.

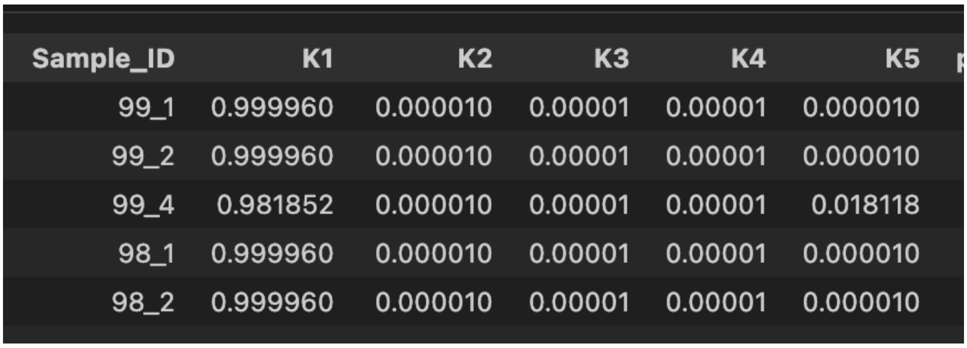
Once the Q file was generated by K=5 admixture analysis, the file was processed using Python's pandas library to construct a suitable table for or database.



```
0.999960 0.000010 0.000010 0.000010 0.000010
0.999960 0.000010 0.000010 0.000010 0.000010
0.981852 0.000010 0.000010 0.000010 0.018118
0.999960 0.000010 0.000010 0.000010 0.000010
0.999960 0.000010 0.000010 0.000010 0.000010
0.999960 0.000010 0.000010 0.000010 0.000010
0.999960 0.000010 0.000010 0.000010 0.000010
0.999960 0.000010 0.000010 0.000010 0.000010
```

Figure 16. Raw Q file produced after the ADMIXTURE analysis.

The raw output file of the admixture analysis only contains K values and does not contain the sample ids that are associated with them. Therefore, we had to make some modifications on the table to make it suitable to for the database.



Sample_ID	K1	K2	K3	K4	K5
99_1	0.999960	0.000010	0.00001	0.00001	0.000010
99_2	0.999960	0.000010	0.00001	0.00001	0.000010
99_4	0.981852	0.000010	0.00001	0.00001	0.018118
98_1	0.999960	0.000010	0.00001	0.00001	0.000010
98_2	0.999960	0.000010	0.00001	0.00001	0.000010

Figure 17. The modified table.

The modified table now has the sample id and therefore can be integrated onto the database.

We used Matplotlib to generate bar plots to visualise the results of the admixture analysis. To enhance the clarity of the plot we decided to organise individuals in their respective populations using the information from the population table in the database. Each population was represented by a single vertical bar. To assign K values for each population the groupby function in pandas was implemented, the average K values of the individuals within the sample population were calculated. Each segment in the bar chart corresponds to the proportion of ancestry from different ancestral population.

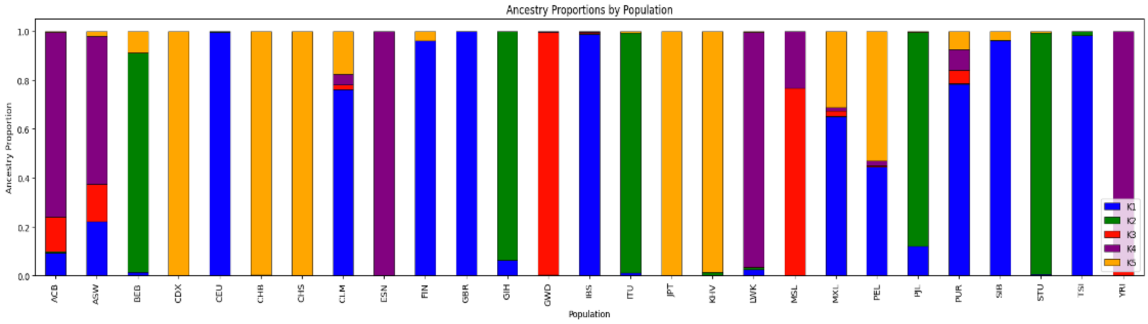


Figure 18. The plot for the different population within the database and their portions of ancestry.

4.3 Limitations

We only ran the admixture analysis for K values ranging from K3 to K5. The decision to choose K5 was based on it having the lowest cross validation value and having the best predictive accuracy. However, it is worth noting that if we had extended the analysis to include higher K values such as K6 or K7 they might have been an even had lower cross-validation values and potentially offered a better fit.

It also could be argued that whilst the admixture plot is informative, it might be perceived as being too generalised as it only took the average K value data. An alternative approach could be considered by plotting all the samples and their K values. This can be done by a python package called Geneview (Pypi). Geneview is a package specifically designed to plot admixture results and can be used to plot individual samples without making the plot look overcrowded. This could, therefore, provide a more detailed plot about ancestries. It's worth noting that the reason why we choose not to use Geneview for our admixture plot was due it's software documentation not providing us detail about which colour map the package use. Therefore, when plotting the bar graph, we weren't able to match the colours of the bars to the keys which contained the K values, therefore it was not clear which segment represented which K.

5. Matrix of Pairwise Population Genetic Differentiation

Weir and Cockerham's Fixation index (F_{ST}) estimator was employed to quantify the genetic differentiation among populations (Weir & Cockerham, 1984). This involved calculating within-population and among-population variance components based on allele frequencies at each locus. Weir and Cockerham's F_{ST} estimator are chosen based on its ability to handle variation in sample size and multiple loci simultaneously incorporate heterozygosity information. The formulation is represented by:

$$F_{ST} = \frac{H_T - H_S}{H_T}$$

To calculate H_T, average allele frequency across two populations was calculated and subtracted by one to obtain the alternative allele frequency. Doubling this provided the product of average allele frequency and its complement under Hardy-Weinburg equilibrium. H_S was calculated separately for each population and then averaged.

Its assumption of the island model of population structure, where populations are assumed to be panmictic (random mating) with equal migration rates and population sizes, facilitates straightforward interpretation of genetic differentiation. However, this assumption may not always reflect the complex reality of natural populations, particularly in scenarios involving hierarchical or isolation-by-distance structures. Additionally, the method assumes populations are in HW Equilibrium within each locus, which is a limitation of this analysis.

5.1 Visualise the population matrix

A heatmap was used for the visualisation of the population matrix as it provided a clear and thoughtful way to represent complex data. The population matrix may contain a vast amount of information, depending on what the user wants to analyse and a heatmap allows us to display all of this information in an easy and concise manner. The colours in the heatmap can represent different values in the matrix, making it easy to see patterns and differences across different populations. The heatmap can provide a quick comparison between different population and can enable the user to identify any interesting patterns or trends between populations. The UBUNTU website has chosen to use the virdis colour scheme for the heatmap as it will make the heatmap easier to visualise for colourblind people (Rudis et al 2024).

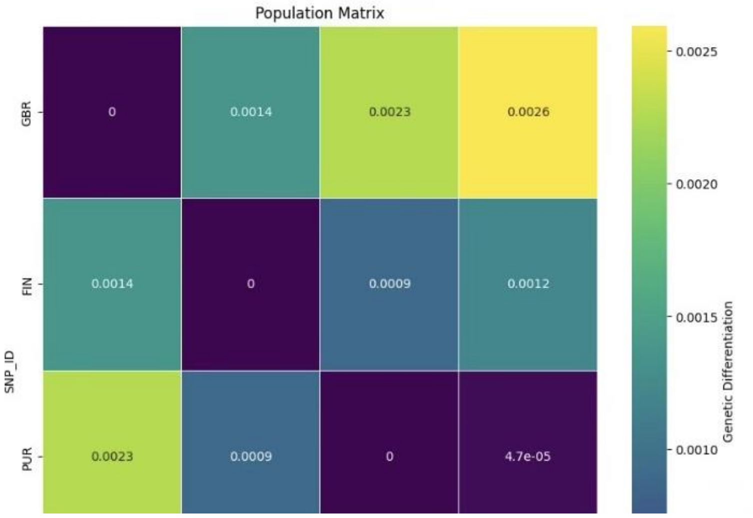


Figure 19. Heatmap of population matrix.

This is an example of what the heatmap may look like once the user has selected their populations of interest. The values in the heatmap represent matrix scores.

Conclusion

UbuntuAnalysis provides a user-friendly, comprehensive solution for inferring patterns of population genetic structure using multiple analytical techniques (PCA, ADMIXTURE, and Fst) in addition to SNP information and clinical implications.

Next, we aim to overcome the challenges of testing and integration by involving multiple-end users, metamorphic tests to detect subtle faults, and version control tools. To facilitate the exploration and communication of complex population genetic patterns, we aim to incorporate new features such as plot scalability and additional summary statistics (Mita & Siol, 2012) to complement the analysis, as well as to include whether the results indicate ancestral-driven outcomes, results of recent migration, or effects of purifying selection (Yurchenko et al., 2019, Potapova et al., 2020) and expand to include information on all chromosomes. This will require and be optimised to efficiently handle larger datasets, possibly incorporating cloud-based technology and neural networks. Similarly, adding extra sections in documentation such as loading times and memory usage.

Reference

1. Alexander, D. H., Novembre, J. and Lange, K. (2009) “Fast model-based estimation of ancestry in unrelated individuals,” Genome research, 19(9), pp. 1655–1664. doi: 10.1101/gr.094052.109.
2. Andrews, C.A. (2010). The Hardy-Weinberg Principle. [online] Nature.com. Available at: https://www.nature.com/scitable/knowledge/library/the-hardy-weinberg-principle-13235724/.
3. Cingolani, P. (2023) Snpeff & SnpSift, Home - SnpEff & SnpSift. Available at: https://pcingola.github.io/SnpEff/ (Accessed: 27 February 2024).
4. Chanock, S.J. and Ostrander, E.A. (2014) Minor allele frequency, Minor Allele Frequency - an overview | ScienceDirect Topics. Available at: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/minor-allele-frequency (Accessed: 27 February 2024).
5. Chang, C. (2024). File format reference - PLINK 1.9. [online] www.cog-genomics.org. Available at: https://www.cog-genomics.org/plink/1.9/formats.
6. Ding, C. (no date) K -means Clustering via Principal Component Analysis, Icml.cc. Available at: https://icml.cc/Conferences/2004/proceedings/papers/262.pdf (Accessed: February 29, 2024).
7. Dutheil, J. Y. (2021) “Correction to: Statistical Population Genomics,” in Methods in Molecular Biology. New York, NY: Springer US, pp. C1–C1.
8. Follow, S. (2022) Reduce Data Dimensionality using PCA - Python, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/reduce-data-dimentionality-using-pca-python/ (Accessed: February 29, 2024).
9. ftp.ncbi.nlm.nih.gov. (n.d.). Index of /pub/clinvar/vcf_GRCh38. [online] Available at: https://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/ [Accessed 28 Feb. 2024].
10. Nature Education (2014). Hardy-Weinberg equation | Learn Science at Scitable. [online] Nature.com. Available at: https://www.nature.com/scitable/definition/hardy-weinberg-equation-299/.
11. Nature Education (2014b). Hardy-Weinberg equilibrium | World Library of Science. [online] www.nature.com. Available at: https://www.nature.com/wls/definition/hardy-weinberg-equilibrium-122/.
12. Oracle (2024) MySQL 8.0 Reference Manual :: 26.2.2 list partitioning, MySQL. Available at: https://dev.mysql.com/doc/refman/8.0/en/partitioning-list.html#:~:text=This%20is%20done%20by%20using,comma%2Dseparated%20list%20of%20integers. (Accessed: 27 February 2024).
13. Oracle (2021). What is MySQL? [online] Oracle.com. Available at: https://www.oracle.com/mysql/what-is-mysql/.
14. Purcell, S. (2014) Whole genome association analysis toolset, PLINK: Whole genome data analysis toolset. Available at: https://zzz.bwh.harvard.edu/plink/ (Accessed: 27 February 2024).
15. Purcell, S. et al. (2007) ‘PLINK: A tool set for whole-genome association and population-based linkage analyses’, The American Journal of Human Genetics, 81(3), pp. 559–575. doi:10.1086/519795.
16. What is mysql? (2024a) Oracle. Available at: https://www.oracle.com/mysql/what-is-mysql/ (Accessed: 27 February 2024).
17. www.ebi.ac.uk. (n.d.). GWAS Catalog. [online] Available at: https://www.ebi.ac.uk/gwas/docs/file-downloads.
18. www.ncbi.nlm.nih.gov. (n.d.). Human Variation Sets in VCF Format. [online] Available at:

https://www.ncbi.nlm.nih.gov/variation/docs/human_variation_vcf/ [Accessed 28 Feb. 2024].

19. The 1000 Genomes Project Consortium (2015). A Global Reference for Human Genetic Variation. Nature, 526(7571), pp.68–74. doi:https://doi.org/10.1038/nature15393.

20. A;, N. (2022) Improving bioinformatics software quality through incorporation of software engineering practices, PeerJ. Computer science. Available at: https://pubmed.ncbi.nlm.nih.gov/35111923/ (Accessed: 29 February 2024).

21. Mita, S.D. and Siol, M. (2012) Egglib: Processing, analysis and simulation tools for population genetics and Genomics - BMC Genomic Data, BioMed Central. Available at: https://bmcgenomdata.biomedcentral.com/articles/10.1186/1471-2156-13-27 (Accessed: 29 February 2024).