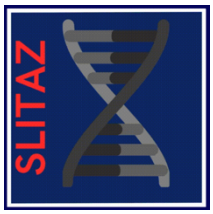

SLITAZ

GHATAORE. BHAVIKHDEEP

SALIM. SANJEEDAH

GODAVARTHI. BHOOMIJA

CHRISTI. XAVIA

Date: 29.02.2024

Contents:

1 Brief overview of the software:

This web-based software tool developed by our team is designed to support a clinician in investigating genetic data by means of population structure analyses. The tool handles molecular biological data, specifically, Single Nucleotide Polymorphism (SNP) data from Chromosome 1, collected from 27 populations including genetic data for several markers from Siberia presented in the form of a zipped Variant Call Format (VCF) file.

Population structure looks at the relatedness among the subgroups present in the whole population sample, usually occurring when samples are drawn from geographically or sociocultural distinct groups. Population structure analyses has been done through three statistical methods, Principal component analysis (PCA), Admixture analysis and Pairwise Fixation index (Fst). PCA allows recognition of patterns by highlighting similarities and differences in the genetic variation across individuals. By analysing the principal components users can infer population structures helping to understand admixture events and historical population migration. Admixture analysis helps understanding in genetic diversity by analysing contributions of ancestral populations to the current genetic makeup of modern populations. This can help in the study of migration patterns and mixing. Furthermore, Fst allows users to quantify genetic differentiation between two populations aiding in understanding population structure. Finally, users can view information for genomic regions, genes or SNPs of interest such as their allele frequency, genotype frequency, genomic coordinates, gene name and clinical relevance.

2 Purpose of the documentation:

This documentation serves as a comprehensive guide for utilising our web-based software tool, designed to assist clinicians and researchers in their exploration of genetic data using population structure analyses. It outlines the procedural steps for conducting analyses of Genotype and Allele frequencies within SNPs and the robust capabilities of Principal Component Analysis, Admixture Analysis, and the Pairwise Fixation Index. Beyond offering a guide on how to navigate our software, this document delves into the scientific underpinnings behind the analytical methods, providing a deeper understanding of their application and significance. We also highlight the limitations of software's/techniques being used so that users can make informed decisions regarding the output of the analyses. Our goal is to not only facilitate a seamless analytical experience but also to enrich the user's knowledge of genomic analysis, enhancing their ability to draw meaningful results from genetic data.

3 Scope and Audience:

This guide is designed to aid anyone interested in population genetics, whether you are a clinician, researcher or student to get the most out of our web-based software tool for analysing genetic data. This can be used by people who have minimal knowledge in the field of genomics to users who are experts in the field. We have aimed to produce a document that is clear and easy to understand, providing step-by-step instructions on how to use the software for various tasks, like analysing population structures and exploring genetic diversity among populations. Our goal is to make advanced genetic analysis techniques accessible to everyone. By guiding users through the process of analysing

genetic data, the documentation supports a range of applications from academic research in population genetics to practical application in medical genetics.

4 Overall Design Philosophy:

1. **Scalability**: The software was designed to accommodate large-scale genetic datasets efficiently. This involved utilizing data structures and algorithms optimized for handling thousands of genetic samples and millions of genetic variants. Additionally, the software was designed to be made easily scalable, allowing for the incorporation of additional genetic data sources and analysis of larger datasets in the future.
2. **Modularity**: The software was developed with a modular architecture to promote flexibility and reusability of components. Each analysis task, such as population structure analysis and genotype frequency comparison, was implemented as a separate module with well-defined interfaces. This modular design allows easy integration of new analysis methods and facilitates code maintenance and troubleshooting.
3. **Maintainability**: To ensure long-term maintainability and extensibility, the software was built with clean and well-documented code following best practices in software development. This includes adhering to coding standards, writing comprehensive documentation, and implementing version control using tools like Git. Additionally, regular code reviews and refactoring efforts were conducted to improve code and readability.
4. **Performance**: Performance optimization was a key consideration in the software design process, given the computational intensity of genetic data analysis tasks. This involved algorithms and data structures optimized for speed and memory efficiency, parallelizing computations where possible, and leveraging hardware resources effectively. As a result, the software is capable of efficiency processing large datasets within time limits.
5. **Usability**: User experience was prioritized in the design of the software interface to ensure ease of use and accessibility for researchers and clinicians with varying levels of expertise in genetics and bioinformatics. This involved designing intuitive user interfaces, providing clear instructions and feedback during analysis workflows, and offering comprehensive documentation and support resources. Additionally, efforts were made to incorporate visualization tools to facilitate data exploration and interpretation.
6. By considering these guiding principles in the software's architecture and design decisions, we aimed to develop a robust, scalable, and user-friendly platform for genetic data analysis, empowering researchers, and clinicians to gain valuable insights into population genetics and medical genetics applications.

5 Installation and Setup:

To install the software, users should have a python environment, such as Visual Code Studio (VSC), available on their system. VSC can be downloaded from the official website [https://code.visualstudio.com/Download]. Then download the database and the software folder uploaded on to our GitHub repository which can be found at [https://github.com/BhavikhSG/Population-Genomics-App/tree/master/PCAandADMIXandSNP%20app]. The database will be made available on OneDrive, the user must then paste the path of the downloaded database into Webapp.py. It should be pasted at the end of this line within the webapp.py file, **"app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///"**. Ensure all the modules are installed using the function "pip install", and then run the application.

6 Frontend:

Tools & Technologies Used:

- **Python and Flask** were used as a web framework to build the application. They facilitate efficient web route generation and HTTP request handling and are easy to integrate with other tools and technologies.
- **HTML** was used to structure the various webpages, in conjunction with CSS for web design to create an aesthetic user interface to enhance the user experience, making the information logically and coherently organized and enjoyable for the user. This focused on the visual presentation - styling and layout.
- **JavaScript** introduces dynamic elements to webpages and interactivity. JavaScript handles user interactions and processes client-side data, introducing dynamic updates, without requiring reloading of the entire webpage.
- **SQLite was used for relational database management, with the SQL toolkit SQL Alchemy which queries and retrieves data from the database as an Object-Relational Mapping (ORM) for Python.** This simplified database operations on the Population Genomics dataset. Database migration was done to define the SQL Alchemy database models within Flask.
- **Plotly** was used to generate interactive PCA plots of the genomic data, addressing the clustering analysis aspect of this Population Genomics Web Application. Features include zooming and values revealed upon hovering, as well as a color key differentiating populations and superpopulations.
- **Matplotlib** was used to create admixture graphs.
- **Jinja templates** were used to organize dynamic content layout, based on the user input and database queries. They dynamically render HTML pages, incorporating data retrieved from the backend database and integrating with Flask routes for user interactions.
- **Linux Command Line** was used as a command-line interface to interact with the operating system.
- Links to **Google Fonts Web API** to fetch font **Audiowide.**

7 Backend:

The backend of the software is powered by Python with Flask, which connects to the SQLite3 database and uses SQL Alchemy for easier and more efficient data querying and retrieval. HTML templates are incorporated for the front-end interface (webpages).

Flask Configuration was done to initialise, including a secret key for session management and the Uniform Resource Identifier (URI) for connecting to the SQLite database.

Database Migration was done, in which the database models were defined using SQL Alchemy to represent different tables in the SQLite database. The class models were:

- PCA – contains attributes: SAMPLE_ID, PC1 and PC2
- POP_GROUP – contains attributes: POPULATION_ID, POPULATION_NAME
- SUPERPOP – contains attributes: SUPERPOPULATION_ID, SUPERPOPULATION_NAME
- ADMIXTURE – contains attributes: SAMPLE_ID, V1, V2, V3, V4, V5
- SAMPLE_POP – contains attributes: SAMPLE_ID, POPULATION, SUPERPOPULATION
- VARIANTS – contains attributes: CHROMO, SNP_ID, POS, REF, ALT, GENE, CLINICAL_RELEVANCE

- Allele Frequency – contains attributes: SNP_ID, and the population columns: ACB, ASW, BEB, CDX, CEU, CHB, CHS, CLM, ESN, FIN, GBR, GIH, GWD, IBS, ITU, JPT, KHV, LWK, MSL, PEL, PJL, PUR, SIB, STU, TSI, YRI.
- Genotype Frequencies – contains attributes: SNP_ID, and the population columns: ACB, ASW, BEB, CDX, CEU, CHB, CHS, CLM, ESN, FIN, GBR, GIH, GWD, IBS, ITU, JPT, KHV, LWK, MSL, PEL, PJL, PUR, SIB, STU, TSI, YRI.

Flask Routes involve:

- Index route '/' - Renders the main page of the application.
- PCA plot route '/PCAplot' - Handles POST requests to generate PCA plots based on selected populations or superpopulations.
- Admixture Plot Route '/Admixture Plot' - Handles POST requests to generate admixture analysis plots based on selected populations or superpopulations.
- SNP Table Route (/SNPtable): Handles POST requests to retrieve and display SNP data and generate pairwise FST matrix plots.

8 Main Features of the Web Application:

- **Header** with transparent effect upon hovering.
- **Logo/Icon** which also functions as a home button. When clicked, redirects user back to homepage, or if already on homepage, refreshes any selections made e.g. populations or superpopulations.
- **Clustering Analysis** – population selection and superpopulation selection. Clicking the Start Analysis button triggers the plot generation.
- **Admixture Analysis** – population selection and superpopulation selection. Clicking the Start Analysis button triggers the plot generation.
- A **toggle function** means user can either select from populations or superpopulations during both clustering analysis and admixture analysis. For both analyses, Select All boxes are also applied which perform analysis on all options (either all populations or all superpopulations).
- **SNP Data Retrieval:**
    - **Dropdown** selection options to choose from the 3 search methods: SNP ID, Gene and Genomic Coordinates (within a range of POS START to POS END)
    - User inputs the **search criteria**
    - Population selection
    - **Start Retrieval** button

9 Database Schema:

The database schema is carefully designed to efficiently store and retrieve genomic data, with appropriate table structures, primary keys, and foreign keys. The schema is created using SQLite, with tables allocated for storing several types of data, including population information, SNP annotations, and clinical relevance data. Users are provided with insights into the database schema, enabling them to understand how data is organized and accessed within the software.

The database schema is meticulously crafted to efficiently manage genomic data, employing appropriate table structures, primary keys, and foreign keys. The "Variants" table encapsulates essential variant information, including chromosome number, SNP identifier, position, reference allele, alternate allele, associated gene, clinical significance, with SNP ID serving as the primary key. The "Super Pop" table houses overarching population groupings, featuring a unique super population ID and corresponding name, where the super population ID acts as the primary key. In the "Sample Pop" table, individual samples are linked to specific populations and superpopulations via foreign keys referencing population and superpopulation tables, with sample ID as the primary key. The "Population" table delineates distinct populations by their IDs and names, utilizing population ID as the primary key. The "PCA" table facilitates principal component analysis by storing sample IDs alongside PC1 and PC2 scores, with sample ID serving as both the primary and foreign key. Genotype and allele frequency tables store SNP IDs alongside population data, where SNP ID functions as the primary and foreign key, facilitating efficient data retrieval and analysis. Lastly, the "Admixture" table correlates sample IDs with admixture proportions (v1-v5), with sample ID acting as the primary and foreign key, providing insights into population ancestry dynamics. This well-organized schema empowers users to comprehend the data organization and retrieval mechanisms within the software.

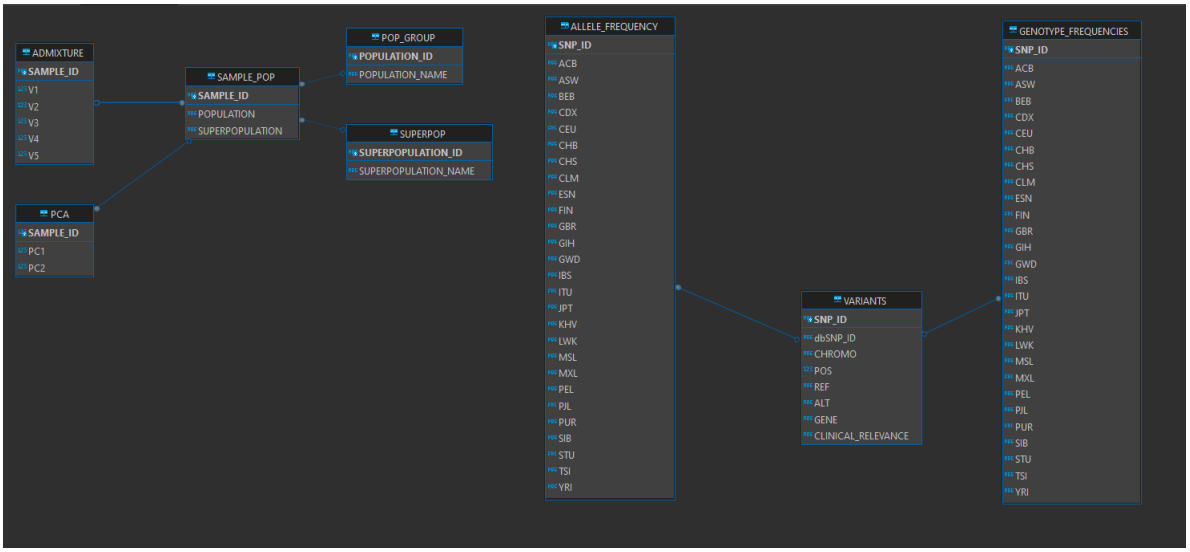

Figure 1: Displays SQLITE Database Schema. The diagram also displays foreign keys using arrows.

9.2 Data Importation:

To import data into the database, users need to typically employ CSV files and SQLite Studio. After connecting to the database, they access the Import and Export Wizard, specifying the CSV file as the data source and the destination as the SQL Server database. Configuration involves mapping columns, setting import options, and executing the import process. Once completed, users verify the imported data for accuracy. This process ensures efficient and accurate importing of data from CSV files into the SQL Server database using SSMS.

10 Principal Component Analysis (PCA):

10.1 Introduction to PCA:

Principal Component Analysis (PCA) is a statistical technique used to simplify complex data sets by reducing the dimensionality of the data while preserving its structure and variability. In the context of this population genomics web application, PCA is employed to visualize the genetic relationships among various populations based on single nucleotide polymorphism (SNP) data. The software incorporates PCA analysis, with preprocessing steps to remove indels and convert data into a suitable format for analysis. Users are guided through running PCA analysis and visualizing results using tools such as **PLINK** and the python Plotly package.

10.2 Justification for choosing PCA:

1. **Capturing Genetic Variation:** PCA captures the major axes of genetic variation within a dataset. In population genetics, genetic variation is the key component for understanding population differentiation and genetic diversity. PCA provides a concise summary of this variation by identifying the principal components that explain the largest proportion of genetic variance.
2. **Dimensionality Reduction:** Genetic data often involves many variables (e.g., SNP markers), which can make analysis and interpretation challenging. PCA mitigates this issue by reducing the dimensionality of the data while retaining most of its informative content. By projecting the data onto a lower-dimensional space defined by the principal components, PCA allows researchers to visualize and explore genetic relationships more effectively.
3. **Visualizing Population Structure:** PCA offers a graphical representation of genetic relationships among individuals or populations. By plotting individuals in the space defined by the principal components, patterns of genetic clustering and population structure can be visually discerned. This visualization aids in identifying groups of individuals with similar genetic backgrounds, detecting population admixture or migration events, and understanding genetic differentiation between populations.
4. **Interpretability:** PCA results are relatively easy to interpret compared to more complex multivariate techniques e.g. Multi-Dimensional Scaling (MDS). Each principal component represents a linear combination of each SNP to the component. This allows researchers to interpret the biological meaning of each principal component and identify which genetic variants drive population differentiation.
5. **Computational Efficiency:** PCA is computationally efficient and scalable to large datasets. This makes it suitable for our 5-million SNP database. It enables rapid exploration of genetic structure and facilitates comparisons across different populations or datasets.

10.3 Why didn't we choose other methods such as MDS and UMAP?

Alternative Clustering techniques such as Multidimensional Scaling (MDS) and Uniform Manifold Approximation and Projection (UMAP) offer distinct advantages, PCA emerged as the preferred method for this study based on its establishments mentioned above, PCA emerged as the preferred method fit this study based on its strengths and limitations, but PCA best aligned with the research objectives and dataset characteristics.

10.4 Implementation of PCA:

1. **Data Preprocessing:** Indels (Insertions and Deletions) were removed from the dataset using bcftools, ensuring that only SNP data was retained for analysis.
2. **Data Conversion:** The compressed Variant Call Format (VCF) file was converted into PLINK format (BED, BIM, FAM) using PLINK.

10.5 Data preparation with PLINK:

Before using ADMIXTURE, the vcf file containing SNP data for the various populations was converted to PLINK's binary file set (.bed, .bim, and .fam formats).

Ensure that PLINK is installed on your system, version's 1.9 and 2.0 support processing of **.vcf.gz** files directly. PLINK can be downloaded from its official website [https://www.cog-genomics.org/plink/].

Steps to producing PLINK's binary file set:

- Navigate to the folder containing the zipped VCF file on your terminal.
- Use the following PLINK command to convert the VCF file to PLINK's binary file set:
    - *plink --vcf chr1.vcf.gz --make-bed --out chr1*
        - **--vcf**: Specifies the input compressed VCF file.
        - **--make-bed**: Convert the VCF file into the binary file set formats.
        - **--out**: Defines the prefix for the output files. PLINK will generate files named **chr1.bed**, **chr1.bim**, and **chr1.fam**.

What does PLINK's binary file set contain?

- The **.bed** file stores the binary genotype data.
- The **.bim** file details the SNP markers.
- The **.fam** file provides sample information in the dataset.

Further information regarding the PLINK output file formats can be found on the official website [https://www.cog-genomics.org/plink/1.9/formats#fam].

PCA Calculation: PCA was conducted using PLINK, generating principal components that summarize the genetic variation in the dataset.

Visualization: Initially, the first two principal components were visualized using RStudio to provide an overview of the genetic structure.

10.6 Implementing PCA into Frontend:

The front-end implementation of Principal Component Analysis (PCA) involves creating a route in the Flask application to handle the visualization of PCA plots based on user-selected populations or superpopulations. When a POST request is received at the '/PCAplot' endpoint, the selected populations and superpopulations are retrieved from the request data. The Flask route then queries the database to fetch PCA data for the selected populations and superpopulations.

For each selected population, the PCA data is filtered, and the PC1 and PC2 values are extracted. A scatter plot trace is created using Plotly, with PC1 values on the x-axis and PC2 values on the y-axis. Each population is represented by a distinct color, allowing for easy differentiation in the plot. Similarly, for selected superpopulations, the process is repeated, and traces are added to the list of traces.

The Plotly layout is configured with appropriate axis titles, and the plot traces are combined into a Plotly Figure object. The Figure object is converted to JSON format, which is passed to the frontend template 'plot.html' using the render template function. In the front-end template, JavaScript code is used to render the Plotly plot based on the provided JSON data.

Overall, this implementation allows users to visualize PCA plots dynamically based on their selection of populations or superpopulations, providing an intuitive interface for exploring population genetic data.

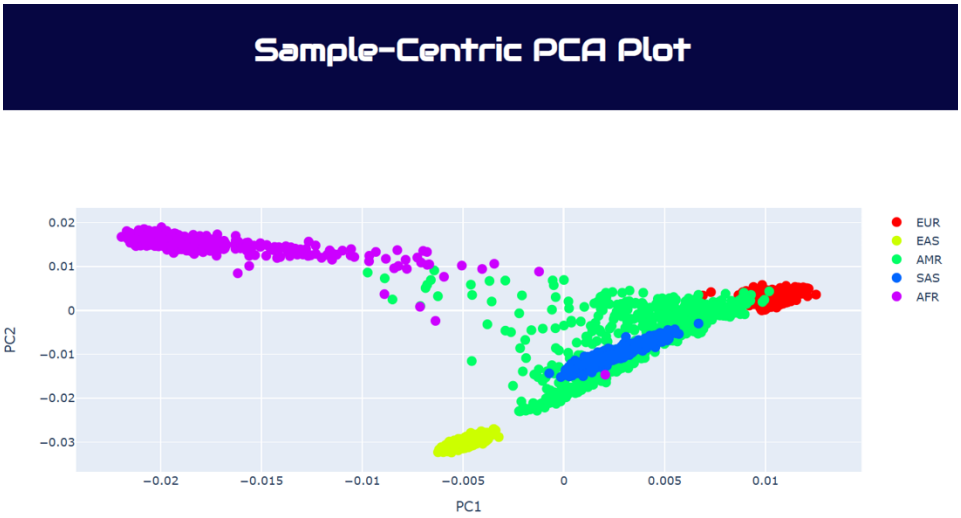

Figure 2: Sample-Centric Principal Component Analysis Plot, generated upon selecting all superpopulations. The data for Siberia is not present as it does not belong to a superpopulation.

10.7 Limitations of PCA:

While PCA is a powerful tool for visualizing genetic data, it has some limitations:

1. **Linear Assumption**: PCA assumes linear relationships between variables, which may not always hold true for complex genetic data.
2. **Population Stratification**: PCA may not fully capture population substructure, especially in cases with subtle genetic differences or admixture.
3. **Interpretation Challenges:** Interpreting the biological meaning of principal components can be challenging, as they represent abstract axes of genetic variation.

10.8 Opportunities for Future Development:

1. **Incorporating Additional Data Types:** Future development could involve integrating other types of genetic data, such as copy number variations or sequence data, to capture a broader spectrum of genetic variation.
2. Calculating the variance for the first two principal components.

11 Admixture analysis:

11.1 What is Admixture analysis?

Admixture analysis is the process of identifying the proportions of ancestry from multiple ancestral populations within groups of individuals. Admixture happens when isolated populations interbreed, and their offspring begin to represent alleles from different ancestral populations. Admixture analysis allows the researcher to classify individuals with unknown ancestry into discrete populations for comparison (Skotte *et al.,* 2013).

11.2 Why was ADMIXTURE chosen to carry out admixture analysis?

There are several software's that can carry out admixture analysis, such as STRUCTURE and ADMIXTURE. STRUCTURE uses a Bayesian clustering approach to assign individuals to populations based on their genotypes, without prior knowledge of the populations. The underlying model of STRUCTURE assumes that there are K populations (where K may be specified by the user or estimated from the data), each of which is characterized by a set of allele frequencies at each locus. Individuals in the dataset are then assigned (probabilistically) to populations, or to two or more populations if their genotypes indicate admixed ancestry. The goal is to find the value of K that best fits the data, and to assign individuals to populations or identify admixed individuals based on this model (Pritchard, Stephens, and Donnelly, 2000).

ADMIXTURE is a tool used for estimating ancestry in a mixed population. It calculates maximum likelihood estimation of individual ancestries from multilocus SNP genotype datasets. It uses the same statistical model as STRUCTURE but calculates estimates much more rapidly using a fast numerical optimization algorithm (Alexander, Novembre, and Lange, 2009). ADMIXTURE works by adjusting two key variables; how common certain alleles are and the proportion of ancestry from different populations. To make these adjustments, ADMIXTURE solves a series of mathematical problems that are designed to find the best values efficiently, using a method known as sequential quadratic programming. This approach quickly narrows down the best solution. The process is sped up even further by using a technique called quasi-Newton acceleration method that refines the search for the solution. Compared to other methods like EM algorithms and MCMC sampling, ADMIXTURE is much faster and more efficient in finding accurate ancestry estimates (Alexander, Novembre, and Lange, 2009).

11.3 Understanding ADMIXTURE file outputs:

Once the software has completed running there will be 3 output files for each K- value.

1. Q- matrix
    a. File extension: .Q
    b. The file, which can be seen in figure 3, contains the estimated ancestry proportions for each individual in the dataset. Each row represents one individual, and each column represents a different ancestral population. The values in the column are fractions which add up to one, representing the proportion of the individual's genome belonging to each ancestral population.

```
0.987760 0.000010 0.000010 0.000010 0.012210
0.999957 0.000013 0.000010 0.000010 0.000010
0.973244 0.023299 0.000010 0.000010 0.003437
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
0.999956 0.000014 0.000010 0.000010 0.000010
```

Figure 3: Q- matrix produced by ADMIXTURE software.

2. P- matrix
    a. File extension: .P
    b. This file, which can be seen in figure 4 provides the estimated allele frequencies in each ancestral population for every genetic marker analysed. Each row corresponds to a genetic marker (SNP), and each column represents an ancestral population. The values indicate the frequency of the allele being studied within each ancestral population.

```
0.999461 0.996669 0.999288 0.999348 0.999990
0.999981 0.999167 0.999439 0.999990 0.999990
0.999292 0.999167 0.998878 0.999388 0.999990
0.999460 0.999170 0.999990 0.999990 0.999990
0.994221 0.989844 0.991519 0.993456 0.994073
0.994503 0.994131 0.995429 0.992952 0.989402
0.996191 0.998384 0.999990 0.996807 0.994576
```

Figure 4: P- matrix produced by ADMIXTURE software.

3. Cross- validation error
    a. File extension: .log
    b. A log file is produced containing the cross- validation error for the specified k- value. Seen in figure 5, the file contains information about the ADMIXTURE run including the command line used, parameters, convergence information, and final log-likelihood of the model. This file is used for trouble- shooting.

```
Cross-validation will be performed.  Folds=5.
Parallel execution requested.  Will use 12 threads.
Random seed: 12345
Point estimation method: Block relaxation algorithm
Convergence acceleration algorithm: QuasiNewton, 3 secant conditions
Point estimation will terminate when objective function delta < 0.0001
Estimation of standard errors disabled; will compute point estimates only.
Size of G: 3928x5027081
Performing five EM steps to prime main algorithm
1 (EM)  Elapsed: 2077   Loglikelihood: -2.45769e+09     (delta): 2.20164e+10
2 (EM)  Elapsed: 2036.68        Loglikelihood: -2.30295e+09     (delta): 1.54741e+08
3 (EM)  Elapsed: 2049.71        Loglikelihood: -2.28432e+09     (delta): 1.86329e+07
4 (EM)  Elapsed: 2061.11        Loglikelihood: -2.27391e+09     (delta): 1.04072e+07
5 (EM)  Elapsed: 2101.27        Loglikelihood: -2.26639e+09     (delta): 7.52118e+06
Initial loglikelihood: -2.26639e+09
Starting main algorithm
1 (QN/Block)    Elapsed: 5592.27        Loglikelihood: -2.08716e+09     (delta): 1.79234e+08
2 (QN/Block)    Elapsed: 5578.44        Loglikelihood: -2.03765e+09     (delta): 4.95078e+07
3 (QN/Block)    Elapsed: 5689.76        Loglikelihood: -2.01719e+09     (delta): 2.0463e+07
4 (QN/Block)    Elapsed: 5545.6 Loglikelihood: -2.003e+09       (delta): 1.41853e+07
5 (QN/Block)    Elapsed: 5575.66        Loglikelihood: -1.99622e+09     (delta): 6.78309e+06
6 (QN/Block)    Elapsed: 5494.53        Loglikelihood: -1.99295e+09     (delta): 3.2632e+06
7 (QN/Block)    Elapsed: 5530.67        Loglikelihood: -1.99034e+09     (delta): 2.61599e+06
8 (QN/Block)    Elapsed: 5452.51        Loglikelihood: -1.98904e+09     (delta): 1.29937e+06
9 (QN/Block)    Elapsed: 5437.57        Loglikelihood: -1.98866e+09     (delta): 374047
10 (QN/Block)   Elapsed: 5388.57        Loglikelihood: -1.98858e+09     (delta): 82225
11 (QN/Block)   Elapsed: 5524.89        Loglikelihood: -1.98857e+09     (delta): 10023.6
12 (QN/Block)   Elapsed: 5603.77        Loglikelihood: -1.98857e+09     (delta): 1446.18
13 (QN/Block)   Elapsed: 5734.75        Loglikelihood: -1.98857e+09     (delta): 106.439
14 (QN/Block)   Elapsed: 5576.9 Loglikelihood: -1.98857e+09     (delta): 8.22272
15 (QN/Block)   Elapsed: 5650.27        Loglikelihood: -1.98857e+09     (delta): 0.263365
16 (QN/Block)   Elapsed: 5637.7 Loglikelihood: -1.98857e+09     (delta): 0.0837994
17 (QN/Block)   Elapsed: 5652.16        Loglikelihood: -1.98857e+09     (delta): 0.0427251
18 (QN/Block)   Elapsed: 5647.35        Loglikelihood: -1.98857e+09     (delta): 0.0200689
19 (QN/Block)   Elapsed: 5572.78        Loglikelihood: -1.98857e+09     (delta): 0.000762939
20 (QN/Block)   Elapsed: 5747.9 Loglikelihood: -1.98857e+09     (delta): 0.000763416
21 (QN/Block)   Elapsed: 5656.8 Loglikelihood: -1.98857e+09     (delta): 0.000148058
22 (QN/Block)   Elapsed: 5634.15        Loglikelihood: -1.98857e+09     (delta): 9.53674e-07
Summary:
Converged in 22 iterations (133754 sec)
Loglikelihood: -1988571169.602375
Fst divergences between estimated populations:
        Pop0    Pop1    Pop2    Pop3
Pop0
Pop1    0.101
Pop2    0.126   0.146
Pop3    0.039   0.065   0.115
Pop4    0.114   0.089   0.166   0.094
CV error (K=5): 0.04038
Writing output files.
```

Figure 5: Log file produced by ADMIXTURE software.

11.4 How was admixture analysis carried out?

Data preparation with PLINK:

Before using ADMIXTURE, the vcf file containing SNP data for the various populations was converted to PLINK's binary file set (.bed, .bim, and .fam formats). Further information regarding PLINK can be found in the PCA section above.

ADMIXTURE installation:

Make sure ADMIXTURE is installed and accessible on your system. For use in a script an environment variable (ADMIXTURE_BIN) the path of the ADMXITURE binary should be implemented. ADMIXTURE can be downloaded from the official website [https://dalexander.github.io/admixture/download.html]. Version 1.3.0 has been used to carry out the analysis.

Job script:

A script was produced to be submitted to a high-performance computing (HPC) environment. The script automates the process of running ADMIXTURE with various values of K. Figure 4 shows the script submitted to Queen Mary's Apocrita HPC facility.

Portable Batch System settings:

1. **#$ -pe smp 12**: Requests a parallel environment called "smp" (symmetric multiprocessing) and allocates 12 CPU cores for this job. Due to the large dataset ADMIXTURE can benefit from parallel processing as the analysis can be completed much quicker. 12 CPU cores is a standard configuration, thus choosing 12 cores fits well with the available hardware.
2. **#$ -l h_rt=240:0:0**: Requests a runtime limit of 240 hours (10 days). This tells the scheduler that the job should not run longer than 10 days and if it does, it will be terminated. 10 days was selected as it is the maximum run time and provides adequate time to carry out the analyses.
3. **#$ -l h_vmem=6G**: Requests 6 GB of virtual memory per core. As 12 cores are requested, this job will be allocated 72 GB of virtual memory. Allocating slightly more memory than required accounts for spikes in memory so the job doesn't fail out of unexpected memory errors. We chose this value through the QMUL HPC documentation site.

Variable settings:

1. **'ADMIXTURE_BIN'** Sets the variable **ADMIXTURE_BIN** to the path where the ADMIXTURE binary is located.
2. **'BEDFILE'** Sets the variable **BEDFILE** to the path of the **.bed** file that contains the genetic data.
3. **'SEED=12345'** Sets a random seed number for ADMIXTURE, which ensures reproducibility of the results.

Running ADMIXTURE:

1. The **for** loop sets up a sequence of K values ranging from 5 to 15. For each K, it runs the ADMIXTURE command with the following options:
2. **--cv**: Indicates that cross-validation should be performed.

3. **-jN**: Tells ADMIXTURE to use N parallel threads, where N is the number of CPU cores requested.
4. **--seed**: Sets the random seed for ADMIXTURE for reproducibility.
5. The output is piped to a log file specific for each K value (**tee log_$K.txt**), which allows the user to see the output on the screen and writes it to a file.

```
#!/bin/bash
#$ -S /bin/bash
#$ -cwd
#$ -pe smp 12         # Request 12 CPU cores
#$ -l h_rt=240:0:0    # Request 240 hour runtime
#$ -l h_vmem=6G       # Request 6GB RAM per core

# Email address for notifications
#$ -M bt23045@qmul.ac.uk

# Notify on job progress
#$ -m bea

# Path to the ADMIXTURE binary within the extracted 'dist' directory
ADMIXTURE_BIN="/data/home/bt23045/ADMIXTURE/dist/admixture_linux-1.3.0/admixture"

# Path to the BED file
BEDFILE="/data/home/bt23045/ADMIXTURE/filteredwoindels_bed.bed"

# Random seed
SEED=12345

# Check if ADMIXTURE binary is executable
if [ ! -x "$ADMIXTURE_BIN" ]; then
    echo "ADMIXTURE binary is not executable or not found at $ADMIXTURE_BIN. Please check the path and permissions."
    exit 1
fi

# Loop through K values from 5 to 15
for K in {5..15}
do
    echo "Running ADMIXTURE for K=$K"
    $ADMIXTURE_BIN --cv -j$NSLOTS --seed $SEED $BEDFILE $K | tee log_$K.txt
done
```

Figure 6: Job script submitted to Queen Mary's Apocrita HPC facility, testing K-values ranging from 5 to 15.

11.5 Choosing the correct K- value:

Choosing the right K-value is very important when plotting the results of ADMIXTURE. As we did not know the number of ancestral populations a range of K- values were tested. Each K- value produces a cross-validation error value which should be compared and the K- value with the lowest error value should be chosen, also consider biological relevance and prior knowledge about the samples. We decided to pick K = 5 which has a cross-validation error value of 0.04038.

11.6 Implementing ADMIXTURE into Frontend:

Admixture was the name of the table in the SQLITE database containing the preprocessed admixture data for the K-value five. This table was linked to the SAMPLE_POP table which contained the populations and superpopulations of each sample ID, which allowed querying of results by their population or superpopulation. The user can select to view the results for either all populations/superpopulations (using select-all button) or the ones that were selected. Dependent on whether the user's input (population or superpopulation), it will trigger the correct pipeline. The pipeline will plot a barplot for the selected user input using the plotting library matplotlib.

11.7 Limitations of ADMIXTURE:

Although ADMIXTURE was chosen over STRUCTURE, the software still has its limitations. This includes the model assuming the population alleles are well mixed and follow the Hardy-Weinberg equilibrium. This might not be true in real life populations where inbreeding and non- random mating may take place. The software also requires selecting the number of ancestral populations (K value), choosing the wrong K

value can lead to misleading results. There is also no single way to calculate the K value and different methods lead to different K values. If the individuals in the population are related and there are population structures, it can bias the results produced by ADMIXTURE. The best results are produced when the samples are not related, and populations are well defined. Although ADMIXTURE is efficient, analyzing large datasets is still computationally intensive, requiring significant computing and processing power which can limit some users. It is important to consider this limitation when carrying out admixture analysis.

11.8 Opportunities for future development:

To improve the admixture analysis methodologies, we would suggest enhancing methods for determining the optimal number of ancestral populations by incorporating more robust cross-validation techniques and account for biological relevance and prior knowledge. We would also expand the models for admixture analyses to produce more sophisticated models that include complex scenarios such as non-random mating, pre-existing population structures and continuous gene flow. This would improve the model's accuracy by considering all scenarios to provide deeper insights into evolution/population structures. We would also test more K- values and compare their cross-validation error values to pick the most suitable K-value, due to limitations in computational memory and time we were able to generate cross-validation error values for K- values 5, 6 and 7.

12 SNP Data Retrieval:

The SNP Data Retrieval section allows users to carry out a search using 3 different methods – by SNP ID, Gene and Genomic Coordinates (position range). This queries the database for single nucleotide polymorphisms (SNPs) based on the user-input search criteria. This method retrieves SNP data from the database, including allele frequencies and genotype frequencies for selected populations, and returns the results as a data frame with the following columns:

SNP ID	POS	ALT	REF	GENE	CLINICAL RELEVANCE	AF (ALT: REF)	GF (Homozygous ALT: Heterozygous: Homozygous REF)

12. 1 Clinical Significance:

In this study, the genetic data underwent rigorous preprocessing using bcftools to exclude any indels, ensuring that only single nucleotide polymorphism (SNP) data remained for downstream analysis. Subsequently, the SNP variant call format (vcf) file was meticulously annotated using ANNOVAR with the hg38 reference genome, enriching each variant with detailed information, including associated gene names. This annotation process provided valuable context by linking genetic variations to specific genes. To further enhance the clinical relevance of our findings, we leveraged data from the UCSC browser, focusing particularly on chromosome 1. The annotated dataset was efficiently managed and manipulated using the pandas library within the Visual Studio Code (VSCode) environment, facilitating comprehensive exploration and analysis of the genetic data. Through these meticulous steps and tools, we aimed to elucidate potential genetic associations and unravel insights into the clinical significance of variations on chromosome 1.

12.2 Functionality:

- **Input Parameters:** The method accepts POST requests containing search type (snp_id, gene, pos_range), search value/criteria, and selected populations.
- **Query Generation:** Based on the search type, appropriate filter conditions are generated to query the database for SNP data.
- **Database Query:** The method performs a database query to retrieve SNP data, including allele frequencies and genotype frequencies for selected populations.
- **Data Processing:** Relevant information is extracted from the query results to generate a table of SNP data for display. The method returns the SNP data as a list of tuples (variant data) containing SNP ID, position, reference allele, alternative allele, gene, clinical relevance, and allele/genotype frequencies for selected populations.
- For multi-population selections, a 'Show/Hide Matrix' button is also displayed, to reveal the pairwise FST matrix, and alongside it, a download link for the text file:

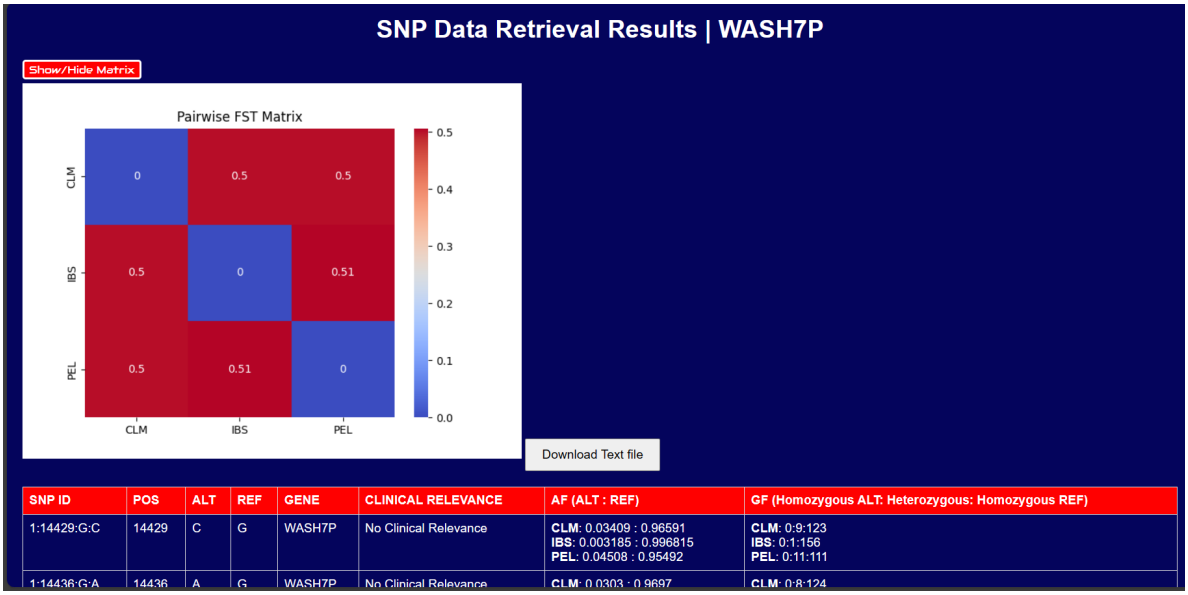

Figure 7: SNP Data Retrieval Results page for dropdown selection 'Gene' and Search Criteria 'WASH7P' for populations CLM, IBS and PEL. The pairwise FST matrix is also shown, controlled by the 'Show/Hide Matrix' button.

- For single-population selection, the same page is displayed but without the 'Show/Hide Matrix' button and its associated pairwise FST matrix and download link.

Currently, the query is limited to 1000 records to prevent excessive data retrieval.

12.3 Opportunities for future development:

Currently, the SNP Search functionality for our application is limited to displaying only values for the specific gene entered. For instance, if the user enters 'WASH7P' into the SNP Search, the system retrieves and displays values for those SNPs associated solely with 'WASH7P' and no other genes. This

means that only the 'GENE' columns (within the 'VARIANTS' table of the database) which contain 'WASH7P' exclusively, are included in the search results. Fields containing 'WASH7P' alongside another gene, such as 'DDX11L1; WASH7P', are currently excluded from our search. This aspect is crucial to address and further develop in the future to enhance the comprehensiveness and usability of our SNP search feature within our application, so that clinicians may retrieve all data pertaining to their search criteria.

13 Pairwise population genetic differentiation matrix:

13.1 What is pairwise population genetic differentiation?

Pairwise FST measures population structure, it is a way to quantify genetic relationships. There are many approaches to calculating Fst, Weir and Cockerham's approach has been used in this instance. The Weir and Cockerham estimator for FST is a statistical method developed by Bruce Weir and C.C. Cockerham to estimate the degree of genetic differentiation between populations. FST quantifies genetic variation among populations relative to genetic variation within populations, ranging from 0 (no differentiation) to 1 (complete differentiation) (Weir and Cockerham, 1984). The Weir and Cockerham approach is designed to provide an unbiased estimate of FST and is particularly suited for use with co-dominant genetic markers, such as SNPs. This method is commonly used in population genetics studies because it accounts for sample size differences and can be applied to multiple alleles and loci (Weir and Cockerham, 1984).

13.2 Why was the Weir and Cockerham approach used?

The Weir and Cockerham approach for estimating genetic differentiation, especially suited for SNP data was the chosen approach due to its ability to provide unbiased estimates of FST and its incorporation of corrections for sample size differences, addressing a common challenge in genetic studies where population samples may vary significantly. It is commonly used across population genetic literature which supports its reliability and accuracy.

13.3 How was the pairwise population genetic differentiation used?

The Weir and Cockerham FST estimator were used in flask to produce a pairwise FST matrix for the selected populations. The matrix was visualised as a heatmap using the visualisation library seaborn. If there is minimal genetic differentiation between the 2 populations, that quadrant will be colored blue and if there is high genetic differentiation the quadrant will be colored red.

13.4 Limitations:

The Weir and Cockerham estimator assume random mating within populations, very minor mutation and migration rates, and that the populations follow the Hardy-Weinberg equilibrium. Deviations from these assumptions can affect the accuracy of the FST estimates. Additionally, the method may not perform well with very small sample sizes or when there is a high rate of gene flow between populations. It is important to consider these assumptions when interpreting the Fst values produced.

13.5 Opportunities for future development:

To improve calculation of pairwise population genetic differentiation an improved algorithm could be developed that goes beyond Weir and Cockerham's assumptions, accounting for random mating, mutation and migration rates. This could be done by including machine learning techniques to correct for biases in the data.

14 How to use the web-based software:

1. **Clustering Analysis** - Select either populations or superpopulations that you want to use for clustering analysis (PCA). Upon selecting either a population or superpopulation, the toggle effect triggers the other option to disappear so that only either population data or superpopulation data is displayed on the PCA graph. Select specific populations then click 'Start Clustering Analysis' to be redirected to the page containing the PCA plot. Alternatively, click 'SELECT ALL' to trigger clustering analysis of all populations or superpopulations.

Figure 8: Population and superpopulation selection checkboxes for Clustering Analysis.

2. **Admixture Analysis -** Select either populations or superpopulations that you want to use for admixture analysis (ADMIXTURE). Upon selecting either a population or superpopulation, the toggle effect triggers the other option to disappear so that only either population data or superpopulation data is displayed in the Admixture graph. Select specific populations then click 'Start Clustering Analysis' to be redirected to the page containing the Admixture plot. Alternatively, click 'SELECT ALL' to trigger admixture analysis of all populations or superpopulations.

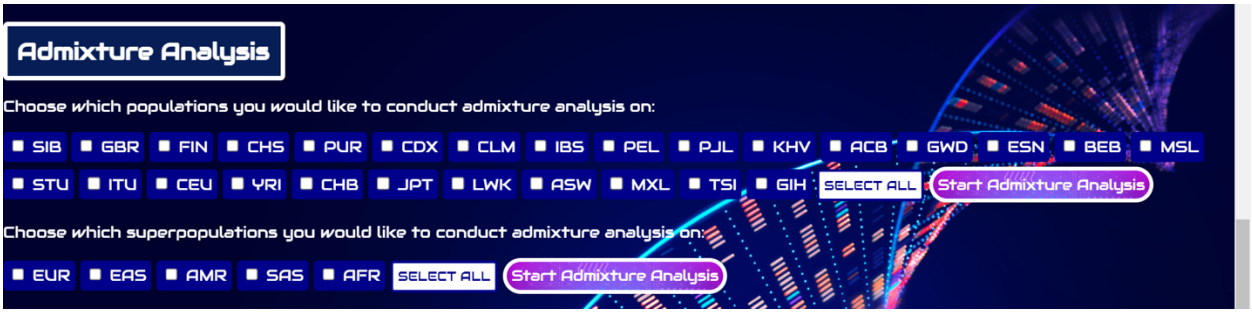

Figure 9: Population and superpopulation selection checkboxes for Admixture Analysis.

3. **SNP Data Retrieval:**

- Select a method that you want to search by from the 3 available methods in the dropdown selection – SNP ID, Gene and Genomic Coordinates.
- Input your search criteria into the search box
- Choose which populations you would like to include in SNP retrieval from the available population checkboxes
- Click 'Start Retrieval' to be redirected to the SNP Data Retrieval Results page.

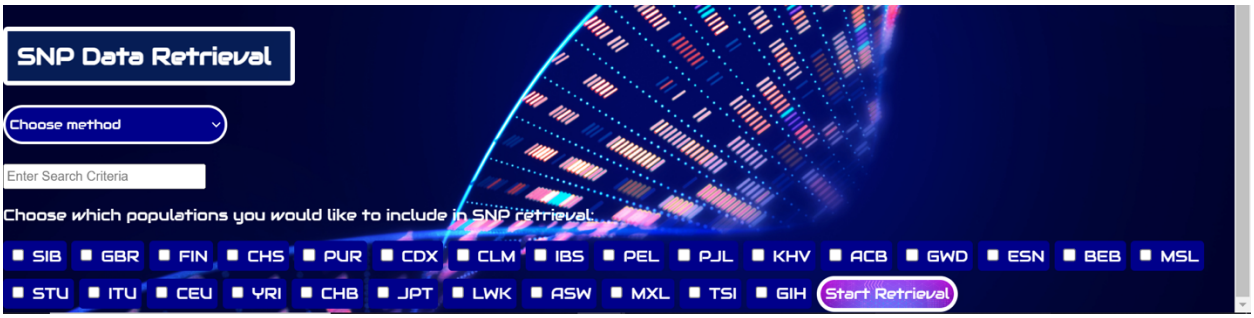

Figure 10: SNP Data Retrieval section showing dropdown selection (which contains the 3 methods SNP ID, Gene and Genomic coordinates), the Search Box for user input and the populations which the user will select.

For multi-population selections, a 'Show/Hide Matrix' button is also displayed, to reveal the pairwise FST matrix, and alongside it, a download link for the text file (see figure 7).

For single-population selection, the same page is displayed but without the 'Show/Hide Matrix' button and its associated pairwise FST matrix and download link.

- Home icon (SLITAZ logo) redirects to main webpage, or refreshes if already on it, resetting the selections made by the user. This is present on the main webpage itself, the PCA plot page and the SNP Data Retrieval page.

The Genomics Analysis Software is a powerful web-based tool for genomic data analysis, offering a wide range of functionalities for analysing population genetics data. This document serves as a comprehensive guide for users and developers, providing insights into installation, usage, architecture, and more. We hope that this software will empower researchers and scientists to gain deeper insights into genetic diversity and population structure.

15 References:

Alexander, D.H., Novembre, J. and Lange, K., 2009. Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, [online] 19(9), pp.1655-1664. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2752134/>

Pritchard, J.K., Stephens, M. and Donnelly, P., 2000. Inference of population structure using multilocus genotype data. *Genetics*, [online] 155(2), pp.945-959. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1461096/>

Skotte, L., Korneliussen, T.S. and Albrechtsen, A., 2013. Estimating Individual Admixture Proportions from Next Generation Sequencing Data. *Genetics*, [online] 195(3), pp.693-702. Available at: https://doi.org/10.1534/genetics.113.154138.

Weir, B.S. and Cockerham, C.C., 1984. Estimating F-Statistics for the Analysis of Population Structure. *Evolution*, 38(6), pp.1358-1370.