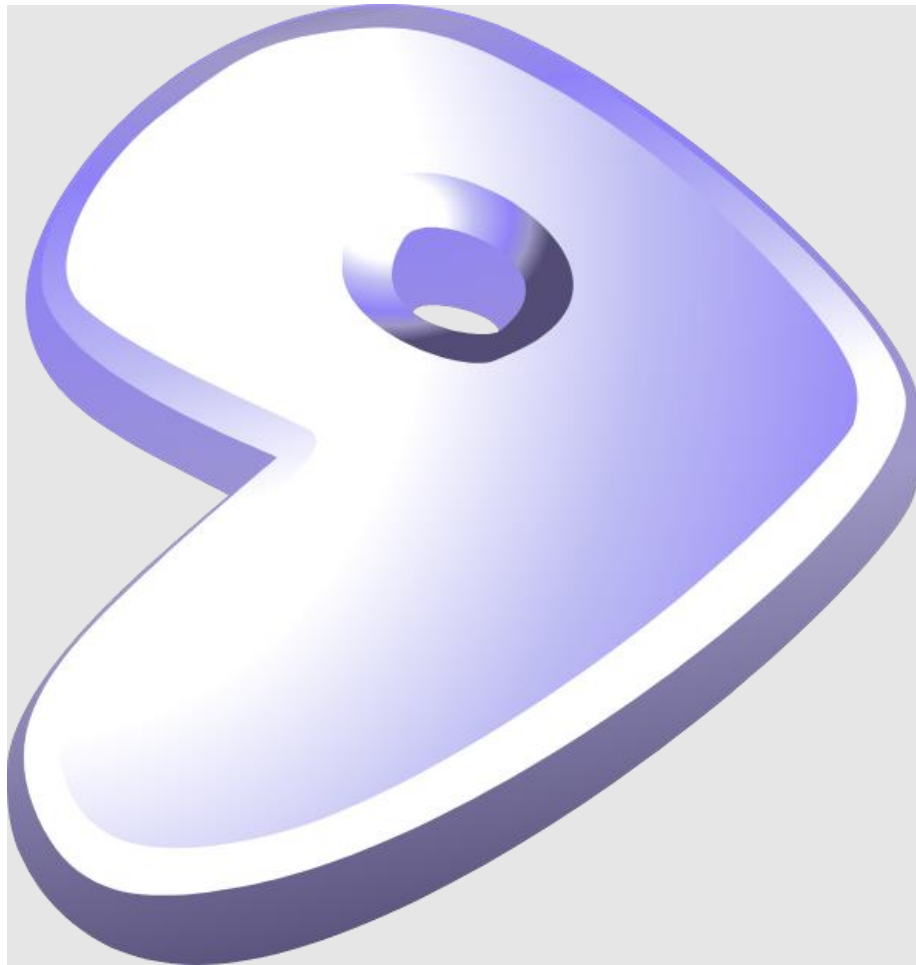


Software Documentation for Gentoo App



Overview

Introduction

In response to the need for a robust tool for population genetic structure analyses, we have embarked on the development of a web application. This endeavor aims to provide researchers and practitioners with a comprehensive platform for exploring genetic data, conducting analyses, and deriving meaningful insights. The overarching goal of the project is to create a versatile application capable of performing diverse analyses on genetic datasets while ensuring ease of use and interpretability of results. In addition, the web application is designed to meet specific requirements aimed at enhancing its analytical capabilities and usability:

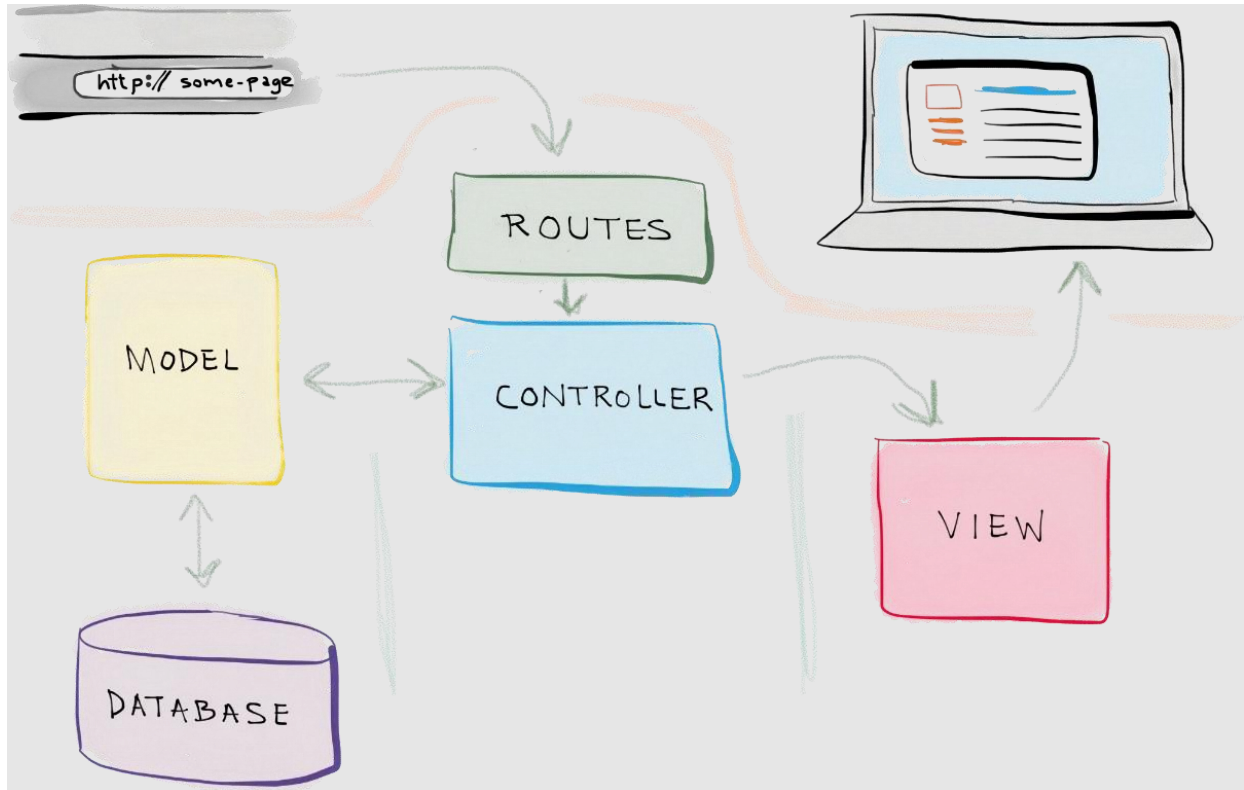
- Clustering and Admixture Analyses:** The application will facilitate clustering and admixture analyses on the provided genetic data via PCA methodology, allowing users to select populations or superpopulations for inclusion. These analyses provide insights into genetic relationships, population structure, and ancestry proportions.
- Data Retrieval and Analysis:** Users can retrieve sample allele and genotype frequencies, and clinical information for SNPs of interest. This functionality enables targeted analysis based on SNP IDs, genomic coordinates, or gene names, with options to specify populations for analysis.
- Pairwise Population Genetic Differentiation:** For analyses involving multiple populations, the application will generate a matrix of pairwise population genetic differentiation. Visual representations of these results will be provided, alongside the option to download data for further analysis.

By integrating these features into our web application, we aim to empower researchers and practitioners in the field of population genetics with a versatile and user-friendly tool for exploring genetic diversity, understanding population dynamics, and conducting insightful analyses. Through the seamless integration of technology and biology, we strive to advance genetic research and contribute to our understanding of human diversity and evolution.

Tools Overview

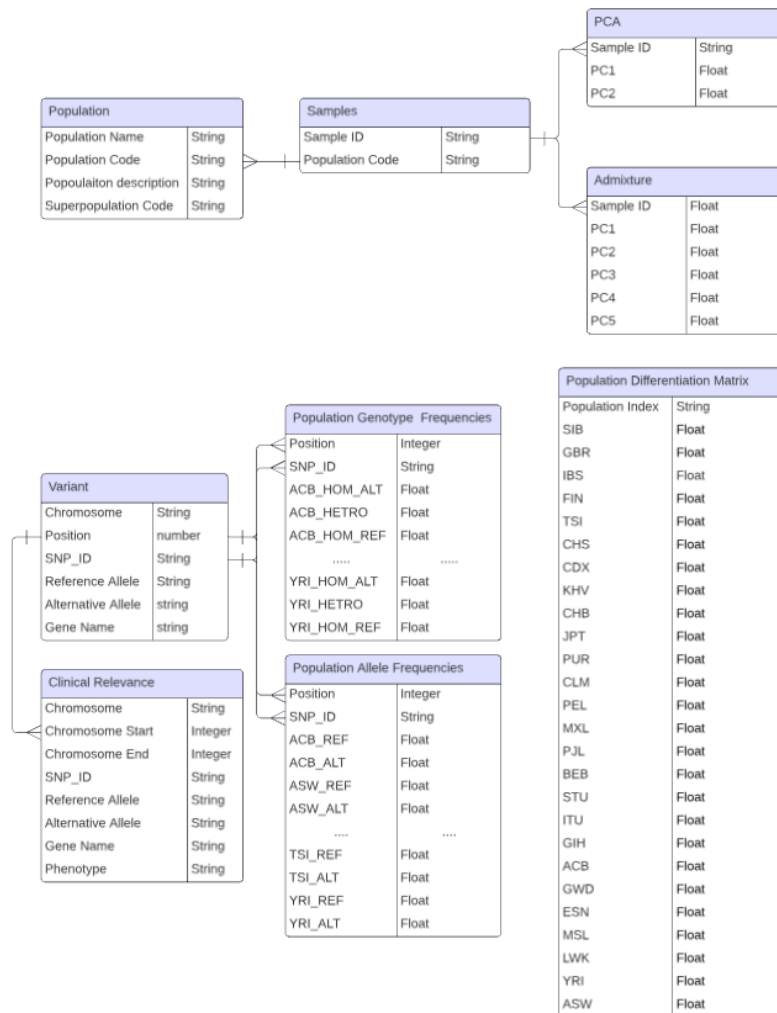
Technology	Version
Python	Python 3.11.5
SQLite3	3.41.2
Flask	Flask 3.0.2
BCFtools	bcftools 1.19
PLINK	plink 1.90
Pysam	N/A
Pandas	N/A
Matplotlib	N/A
Seaborn	N/A
Numpy	N/A

System Architecture



We have adopted the Model-View-Controller (MVC) software design pattern, leveraging Flask, a lightweight yet powerful web framework for Python, to structure and organise our application. The MVC architecture divides the application into five interconnected components: Models, Views, Controllers, Routes and Services. Models represent the data and its associated logic, serving as the core of the application. Controllers act as intermediaries between the views and models, orchestrating the flow of data and business logic. Views are responsible for presenting the data to users in a clear and accessible manner. Routes define URL endpoints and handle incoming HTTP requests, directing them to corresponding controller actions for processing and Services encapsulate reusable business logic and operations, providing functionalities across multiple components for enhanced code reusability and maintainability. In tandem with the MVC structure, we utilise Flask blueprints to modularize our application, enabling us to divide it into smaller, manageable pieces. Blueprints allow us to register multiple components at distinct URL prefixes, facilitating efficient routing and resource management.

Database Schema



This visual representation of the schema shows nine tables. There are essentially three blocks of tables:

1. The first block contains three tables: population, samples, PCA and admixture. The population table is connected to the samples table via population code and the PCA and Admixture tables via sample IDs.
2. The second block has four tables: variant, clinical relevance, population genotype frequencies and population allele frequencies. The variant table connects to all other tables. The variant table is connected by chromosome position and SNP_ID to both frequency tables. The clinical relevance table is connected to the variant table by chromosome start.
3. The third block only has one table containing FST values for each population in the pairwise population differentiation matrix.

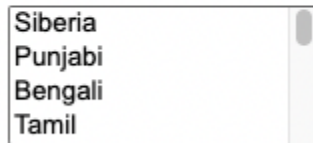
User Guide

On the homepage of our app, the user has the option of clicking on four tabs:

Clustering Analysis

Select groups for clustering analysis:

☒ Populations ☐ Superpopulations



Perform Clustering Analysis

Clustering Analysis: Upon clicking on this option, the user will be rerouted to a new page that will allow them to first choose either to perform analysis on population or superpopulation. After selecting, the user can choose between five superpopulation or twenty-six standard populations. Once the user has selected their superpopulations or population, they click on *"Perform Clustering Analysis"* to be redirected to a new page that will display a PCA plot showing the clustering of the populations along with a coloured key to identify each population in the plot.

Admixture Analysis

Select groups for admixture analysis:

☐ Populations ☒ Superpopulations



Perform Admixture Analysis

Admixture Analysis: When selecting this option, users will be directed to a new page where they can choose between analyzing either population or superpopulation admixture. After making their selection, users can then choose from five superpopulations or twenty-six standard populations. Once the user has made their choice of superpopulations or populations, they can click *"Perform Admixture Analysis"* to be redirected to another page displaying an admixture plot illustrating the population clustering, along with a color legend to identify each population on the plot.

Genotype Frequency Analysis

SNP ID:

Gene Name:

Genomic Region - Start Position:

Genomic Region - End Position:

Select Populations to include:

☐ ACB ☐ ASW ☐ BEB ☐ CDX ☐ CHB ☐ CHS ☐ CLM ☐ ESN ☐ FIN ☐ GBR ☐ GIH ☐ GWD ☐ IBS ☐ ITU ☐ JPT ☐ KHV ☐ LWK ☐ MSL ☐ MXL ☐ PEL ☐ PJL ☐ PUR ☐ SIB ☐ STU ☐ TSI ☐ YRI

Genetic Information: Selecting this option will reroute the user to a new page where they can select their search criteria: *SNP ID*, *Genomic Coordinates* or *Gene Names*. If the user selects SNP ID, an empty field will appear where the user can enter their SNP IDs separated by commas. If the user selects Genomic Coordinates, three fields will appear where the user can enter chromosome number, start position on the chromosome and end position on the chromosome. If the user selects Gene Names, an empty field will appear where the user can enter their genes of interest separated by commas. After specifying the requirements, the user will click on "Retrieve *Information*" to be redirected to a new results page.

About

Developed by:

Osman Mohamed

Jeremiah Mushtaq

Harshan Mehra

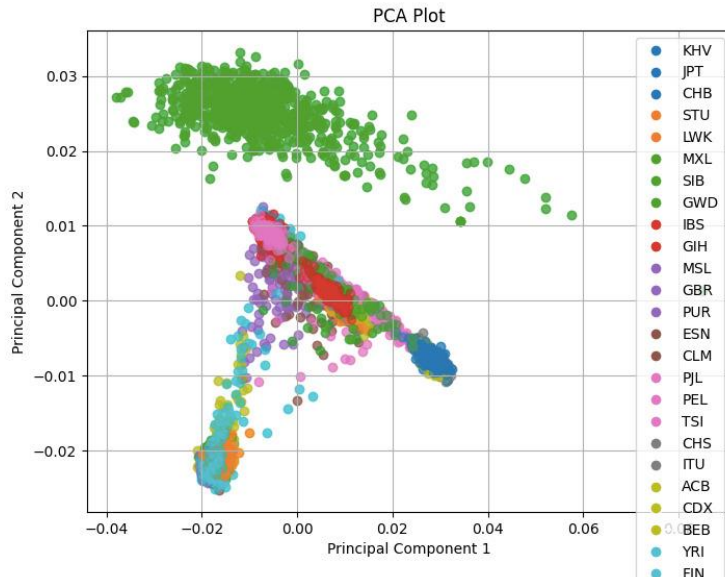
Documentation:

Download the app documentation [here](#).

About: This page has information on the developers of the application and an option to download app documentation.

Technology Stack

Principle Component Analysis



Principal Component Analysis (PCA) is a pivotal technique within our web application's clustering analysis framework. The technique is a powerful dimensionality reduction method commonly used in genetics to unravel underlying patterns and structures within high-dimensional datasets. By transforming the original variables into a new set of orthogonal variables (principal components), PCA simplifies the data while preserving its variance, thus aiding in visualizing and understanding complex relationships. We used PCA for

clustering and admixture analysis due to the following factors:

- **Dimensionality Reduction:** Our data comprises numerous variables, posing challenges for traditional clustering algorithms. PCA reduces the dimensionality of the data by identifying the most informative features, thereby facilitating efficient clustering analysis.
- **Visualisation:** PCA transforms high-dimensional data into a lower-dimensional space, making it easier to visualise and interpret. We found that variance in our data was best explained in principle components 1 and 2. Therefore, we decided to use those two components to display insights into genetic structure and population relationships.
- **Linearity:** PCA assumes linearity in data relationships, making it suitable for linearly separable clusters. Our research indicated that in genetic analyses relationships between variables are typically linear hence it was decided that PCA would provide an effective means of clustering.
- **Computational Efficiency:** Compared to some other clustering algorithms, such as hierarchical clustering or k-means clustering, PCA is computationally efficient, making it suitable for large-scale genetic datasets.

PLINK

PLINK is a software that offers a wide array of functionalities tailored for population genetic analysis, ranging from data formatting and quality control to various statistical analyses. Its versatility makes it a one-stop solution for processing genetic data, aligning with the diverse needs of our web application. There were two main reasons for using PLINK:

- **Scalability:** PLINK's scalability and performance can handle large-scale genetic datasets efficiently. This scalability aligns with the demands of our web application due to the size and

BCFtools

BCFtools is a collection of command-line tools designed for efficiently working with VCF and BCF files. It provides a wide range of functionalities for tasks such as filtering variants, merging or comparing VCF files, calculating genotype statistics, and performing various analyses on genomic variation data. The reasons for using BCFtools were:

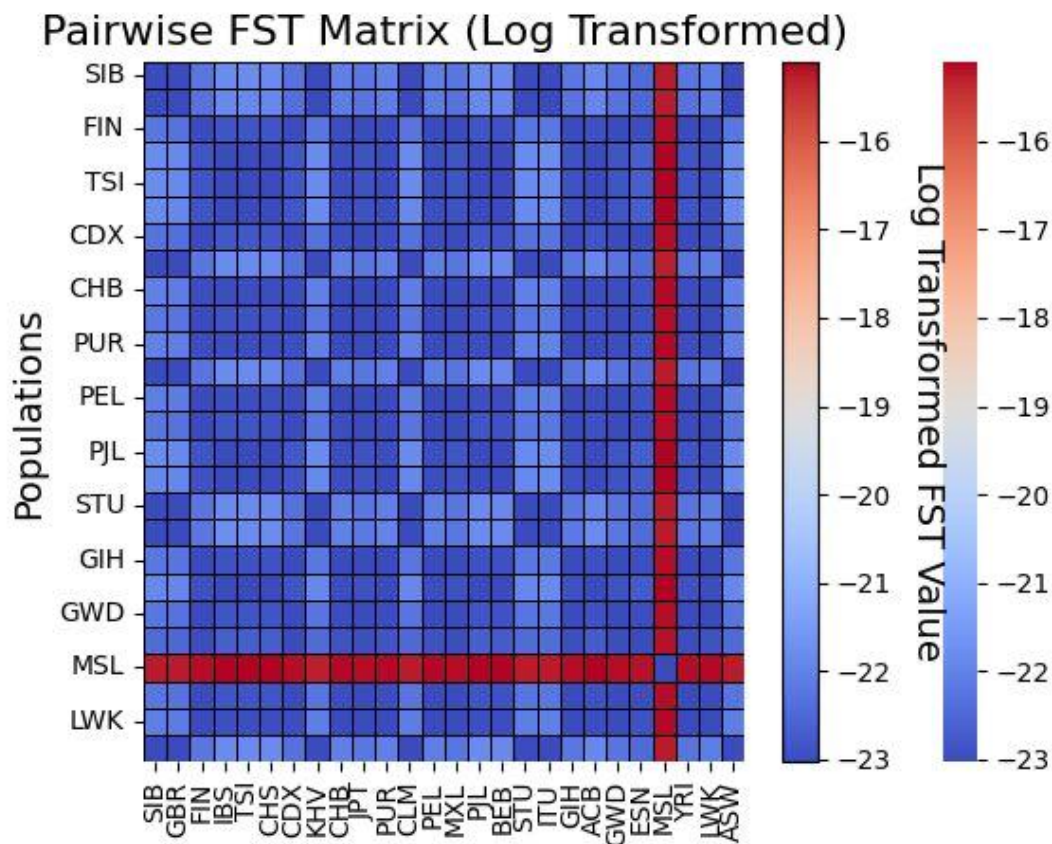
- **Efficiency:** BCFtools is highly optimized for working with VCF and BCF files, allowing for fast and efficient processing of large genomic datasets. It can handle large VCF files with millions of variants and thousands of samples efficiently, making it suitable for population-scale studies. Our VCF file had approximately 5 million variants for ~4000 individuals making the processing via conventional python scripts time consuming hence BCFtools made it possible to process our main VCF file in an efficient manner.
- **Flexibility:** BCFtools provides flexible options for filtering and subsetting VCF files based on various criteria, including sample IDs, population labels, genotype information, and variant annotations. This allowed us to customise the filtering criteria to match the specific populations of interest and extract data for those populations.
- **Integration with Bioinformatics Pipelines:** BCFtools is commonly used as part of bioinformatics pipelines and workflows for genomic data analysis. Its compatibility with other bioinformatics tools and formats makes it easy to integrate into existing analysis pipelines for population genetics studies. In our analysis, we integrated BCFtools with PLINK as the former tool was used to segregate the primary VCF file according to multiple populations and then the resulting population specific VCF files were used by PLINK to obtain allele and genotype frequencies.

Clinical Relevance

Data regarding clinical relevance of SNPs was obtained from the ClinVar Database. ClinVar serves as a centralised repository for storing and sharing clinical interpretations of genetic variants submitted by researchers, laboratories, clinicians, and expert panels from around the world. The database contains much information on the association between genetic variants and human diseases, including Mendelian disorders and complex traits. We used ClinVar for the following reasons:

- **Data Sharing and Accessibility:** ClinVar promotes data sharing and accessibility by providing open access to its database and data submission tools. Researchers, clinicians, and patients can easily access, search, and retrieve variant information through the ClinVar website, APIs, and data downloads, fostering collaboration and knowledge exchange in the genetics community. This allowed us to programmatically obtain clinical information for a large number of SNPs.
 - **Structured Annotations:** ClinVar provides structured annotations for each variant, including information on variant type, genomic location, allele frequencies in different populations, associated diseases or phenotypes, mode of inheritance, and supporting evidence from clinical studies and functional assays. These structured annotations facilitate data retrieval, interpretation, and comparison across different variants and studies. This was important because our analysis required information of allele frequencies along with clinical data.
-

Fixation Indexing



FST (Fixation Index) is a measure of genetic differentiation between populations. It quantifies the extent of genetic variation among populations relative to the total genetic variation in the entire population. FST values range from 0 to 1, with higher values indicating greater genetic differentiation between populations. An FST value of zero indicates no genetic differentiation between populations, meaning that all populations have the same allele frequencies whereas an FST value of 1 indicates complete genetic differentiation between populations, meaning that each population has fixed differences in allele frequencies and no gene flow occurs between them. FST is used in population genetics to study genetic diversity, population structure, and evolutionary relationships. It plays a crucial role in understanding the genetic dynamics of populations, including patterns of migration, genetic drift, and natural selection. We used FST to determine genetic differentiation between populations in the form of a matrix. The matrix is based on Wright's F-statistics performed on alternative allele frequencies for each population. Most resulting values equated to approximately 10^{-10} therefore visual representation of the data showed little variation. To better interpret the data, all FST values were log transformed with an offset of 10^{-10} to account for values equaling zero. This allowed for better interpretation of data in the resulting heatmap for users.

Project Evaluations

Limitations

Flask & SQLite3

Despite their strengths in providing a flexible framework for web development, Flask and SQLite3 present certain constraints in terms of scalability of the application. By default, Flask runs in a single-threaded mode. While this simplifies development and ensures thread safety, it limits concurrency and can lead to performance bottlenecks, especially under heavy load or when handling multiple concurrent requests. Likewise, SQLite3 stores data in a single disk file, making it unsuitable for high-concurrency or distributed environments where multiple clients need simultaneous access to the database. Concurrent writing operations can lead to file locking and contention issues, reducing performance and scalability. However, the prototype nature of the app means it was built with a focus on functionality rather than scalability.

Fixation indexing

Another limitation of this application is related to analysis of genetic differentiation between populations. The algorithm our app uses to calculate F_{ST} for each population takes an average of all SNPs present within that population thereby reducing the amount of genetic diversity that could possibly be captured. There are also assumptions and limitations to using Wright's F-statistics to calculate F_{ST} values. The accurate estimation and interpretation of F-statistics depends on several assumptions, including Hardy-Weinberg equilibrium (HWE) within populations, absence of migration, and random mating. Violations of these assumptions can lead to biased estimates of F-statistics. For example, departures from HWE due to inbreeding or selection can inflate or deflate estimates of F_{IS} (inbreeding coefficient), impacting the overall interpretation of population structure.

Clinical Relevance Database

ClinVar is a valuable resource for obtaining clinical relevance information for genetic variants but it has several limitations:

- **Incomplete Coverage:** ClinVar may not include annotations for all genetic variants, particularly rare or recently discovered variants. Variants that have not been submitted to ClinVar or have not undergone expert curation may not have associated clinical interpretations available in the database.
 - **Subjectivity of Interpretations:** Clinical interpretations in ClinVar are based on submissions from various sources, including laboratories, researchers, and clinicians. Interpretations may vary depending on the expertise and criteria used by the submitter, leading to inconsistencies or discrepancies in clinical significance classifications.
 - **Population Representation:** ClinVar may have biases in the representation of certain populations or ethnic groups, potentially leading to disparities in variant interpretations and clinical relevance assessments. Variants that are more prevalent or well-studied in certain populations may be overrepresented, while variants in underrepresented populations may be underreported or overlooked.
-

Alternatives

PCA

Principal Component Analysis (PCA) was chosen over Uniform Manifold Approximation and Projection (UMAP) and Multidimensional Scaling (MDS) in our genetic analysis framework due to the limitations associated with UMAP and MDS. While both UMAP and MDS offer valuable dimensionality reduction and visualization techniques, they were not selected primarily because of their drawbacks. UMAP, for instance, may not always capture linear relationships effectively, which are common in genetic datasets. This limitation could hinder its ability to identify linearly separable clusters, a crucial aspect in genetic analysis. Similarly, MDS can be computationally intensive, especially for large-scale genetic datasets, which might impede swift processing and analysis. In contrast, PCA's computational efficiency, simplicity of interpretation, and ability to handle linear relationships align well with the characteristics of genetic data. By addressing these limitations of UMAP and MDS, PCA emerges as the preferred choice for uncovering meaningful genetic patterns and structures within our framework.

ADMIXTURE

The decision to utilize admixture analysis over STRUCTURE within our web application was driven by several factors, primarily centered on the computational demands and limitations of STRUCTURE. While STRUCTURE is a widely respected tool for population structure analysis, its computational intensity poses challenges for large-scale datasets or real-time analysis within a web application context. The extensive computational resources and time required for running STRUCTURE may hinder the responsiveness and scalability of our application, potentially impacting user experience and efficiency. Additionally, STRUCTURE relies on Bayesian clustering algorithms, which come with inherent assumptions about the underlying genetic model and population structure. These assumptions may not always align with the complexities of real-world genetic data, potentially leading to biased or inaccurate results. Furthermore, the comprehensive ancestry estimates provided by ADMIXTURE offer a more user-friendly and interpretable approach compared to the potentially complex outputs of STRUCTURE. By opting for ADMIXTURE over STRUCTURE, we prioritize efficiency, scalability, and the ability to provide accurate and actionable insights into genetic ancestry while mitigating the computational burden and potential limitations associated with STRUCTURE.

Clinical Relevance Database

While ClinVar is a widely used and comprehensive database for accessing clinical relevance information for genetic variants, there are several alternative resources and databases that provide similar information. Some of these alternatives include:

- **dbSNP:** This SNP database is maintained by the NCBI and contains information on genetic variations, including SNPs, indels, and structural variants. While dbSNP primarily serves as a repository for genetic variation data, it may include annotations related to variant clinical significance from sources such as ClinVar.
 - **gnomAD:** This database provides variant frequency data from large-scale sequencing projects, including exome and genome sequencing data from diverse populations. While gnomAD primarily focuses on variant frequency information, it also includes annotations related to variant pathogenicity and clinical relevance.
-

- **OMIM:** OMIM is a comprehensive database of genes and genetic disorders curated from biomedical literature. OMIM provides detailed information on the molecular basis of genetic diseases, including descriptions of disease-associated variants and their clinical significance.

Python Libraries

Matplotlib

Matplotlib was selected for its comprehensive plotting capabilities, particularly for creating static visualizations like heatmaps in the code. Its pyplot interface provides a flexible and customizable framework for generating a wide range of plots, from simple line charts to complex, publication-quality figures. Matplotlib's extensive customization options allow for precise control over every aspect of the plot, including colors, labels, annotations, and layouts, enabling users to create highly tailored visualizations to convey their data effectively. Furthermore, Matplotlib's integration with Jupyter notebooks facilitates interactive exploration and analysis of data through dynamic plotting. Despite the availability of alternative plotting libraries like Seaborn or Plotly, Matplotlib remains a popular choice due to its maturity, versatility, and extensive documentation, making it suitable for a wide range of plotting tasks in scientific computing and data visualization.

NumPY

While alternatives like TensorFlow or PyTorch excel in specific domains such as deep learning, they are not suitable replacements for NumPy in general numerical computing tasks like those required in this application. NumPy is purpose-built for efficient array manipulation and mathematical operations, making it the preferred choice for tasks such as calculating mean allele frequencies and pairwise FST values. Unlike TensorFlow and PyTorch, which are primarily designed for building and training neural networks, NumPy provides a simpler and more lightweight solution for performing fundamental numerical computations on arrays and matrices. Additionally, NumPy's rich ecosystem of scientific computing tools and libraries, along with its seamless integration with other Python packages, further solidifies its position as the standard choice for numerical computing in Python. While TensorFlow and PyTorch offer advanced capabilities for deep learning tasks, they lack the versatility and simplicity of NumPy for general numerical computations, making them less suitable alternatives for the requirements of this code.

Pandas

Pandas was chosen to handle the tabular data in the code due to its intuitive and efficient data structures for structured data manipulation. Its DataFrame object provides a powerful and flexible tool for indexing, selecting, filtering, and transforming datasets, making it well-suited for tasks involving data exploration, cleaning, and analysis. Pandas' rich set of functions simplifies common data operations, such as merging, grouping, and reshaping, thus streamlining the data preprocessing workflow. Additionally, Pandas seamlessly integrates with NumPy, allowing for efficient data interchange between numerical computations and tabular data processing. Another alternative to Pandas for handling tabular data in Python is the datatable library which supports integration with NumPy and Pandas, allowing for seamless interoperability with existing codebases and workflows. However, despite datatable provides a compelling alternative to Pandas for applications requiring efficient processing of large tabular datasets, Pandas was chosen over datatable due to its widespread adoption, extensive documentation, and ease of use, which may be more suitable for users who prioritize simplicity and familiarity in their workflow.

Future Developments

While our current web application provides a solid foundation for population genetic analysis, there exist several avenues for future development that can significantly enhance its functionality and value. One promising direction involves expanding the range of analysis modules to cover a broader spectrum of population genetic analyses. For instance, integrating tools for haplotype-based analyses, such as haplotype phasing and identification of haplotype blocks, could yield deeper insights into genetic variation and population structure. Additionally, incorporating advanced statistical methods for detecting signals of natural selection and identifying candidate adaptive loci would enable researchers to explore the evolutionary forces shaping genetic diversity in greater detail. Improving the scalability and efficiency of the application to handle even larger datasets would extend its utility to studies involving extensive genomic data, such as whole-genome sequencing datasets. Furthermore, enhancing the accessibility and usability of the application through the integration of interactive data visualization tools and user-friendly interfaces for result interpretation would benefit researchers and practitioners with varying levels of expertise in population genetics. Lastly, fostering collaboration with experts in bioinformatics, genetics, and computational biology to continuously update and refine the application's functionalities based on emerging research trends and user feedback will be pivotal in ensuring its ongoing relevance and effectiveness in advancing population genetic studies. By pursuing these avenues for future development, our web application can evolve into an indispensable tool for researchers exploring genetic diversity, population dynamics, and evolutionary processes.
