Project Overview

Title: Web-Based Tool for Population Genetic Structure Analysis

Team Members: Dhwani, Djinho,Haik, Nasibah

Version: Initial version (Prototype)

Date: 29.02.23

1. Introduction

Project Rationale

This project will help compare sites on chromosome 1 between several populations including that of Siberian populations. Not all 5 million snps in the original vcf were included in the analysis, 31,721 identified snps were compared across these populations in the app. Clinical annotations were also provided that state whether the variant the individual has stated to be disease benign or not. This can be used to help identify what disease related snps tend to be common in a specific ethnicity so could be useful for screening purposes. The PCA gives an idea of how much the genetic profiles of the individuals vary/ are similar based on ethnicity and shows whether individuals of the same ethnicity tend to cluster together on or not for the individual PCA. It also shows for the PCA by population which populations are closer to each other and which are further and whether any clusters form. The admixture analysis shows what similar that each population is to 5 hypothetical ancestral populations made from populations are grouped into 5 groups based on genetic similarity. This helps give a representation how genetically diverse these populations.

Objectives

This project aims to compare Siberian samples with other populations based on specific sites on chromosome 1. The raw data provided contains 726 Siberian individuals data at specific sites on chromosome 1 and smaller sample sizes for the other 26 human populations. The aim of this project is to compare via admixture analysis and clustering analysis to see similarities between genomic sites in all these available populations at both population level and superpopulation level and to be able to get clinical information,genotype and allele frequencies for snps queried by the user as well as to be able to get a matrix of pairwise population genetic differentiation and a file with the results of this analysis when different populations are selected.

3. System Architecture

The sql schema contains 6 tables:

**SNP\_Data** where each row represents a snp and which has the following columns:

1. SNPID:Row number
2. Chromosome: The chromosome that the snp is on
3. Position: The genomic coordinate of the snp
4. ID: The rsID of the snp
5. REF: The reference allele
6. ALT: The alternate allele
7. GeneName: The gene that the snp is located on or in between
8. GeneType: A description of the region the SNP is located on e.g. UTR3,intergenic,exonic
9. ClinicalSignificance: Whether the variant could result in a disease or not
10. DistanceToAdjacentGenes:How close the snp is to a neighbouring gene
11. Exonic function:Explains the effect of the variant e.g. Missense

12-38. Genotype and allele frequencies for each population. Data is arranged as follows: in each cell the frequency of homozygous reference:heterozygous:homozygous alternate ;allele frequency

39-43. Genotype and allele frequencies for each superpopulation. Data is arranged as follows: in each cell the frequency of homozygous reference:heterozygous:homozygous alternate ;allele frequency

**admixture\_results** which has the following columns:

1. ResultID: Row number
2. PopulationID: Randomly assigned codes associated with each population
3. Ancestry\_1:Proportion of ancestry 1 DNA in each population/superpopulation
4. Ancestry\_2: Proportion of ancestry 2 DNA in each population/superpopulation
5. Ancestry\_3: Proportion of ancestry 3 DNA in each population/superpopulation
6. Ancestry\_4: Proportion of ancestry 4 DNA in each population/superpopulation
7. Ancestry\_5: Proportion of ancestry 5 DNA in each population/superpopulation

**Individual\_pca\_coordinates** which has the following columns:

1.SampleID: Row number

2.PopulationID: The first half of the sampleIDs is the population code for each individual then the second half is the same individuals but using their superpopulation codes so the same data for principle components 1-10 is repeated in the same order at the second half of the table

3-12:pc1-pc10: The proportion of variance captured in that principle component for each individual

**pca\_coordinates** has the following columns:

1.CoordinateID:Row number

2.PopulationID: Number assigned to each population and superpopulation. 6-32 are for populations and 1-5 are for superpopulations

3-12:pc1-pc10: The proportion of variance captured in that principle component for each population/superpopulation

**populations** has the following columns:

1.PopulationID: All 32 population codes in numerical order

2.PopulationName:3 letter population code for each population

3.is\_Superpopulation: 0 means it’s a individual population and 1 means its superpopulation

3.Non-functional Requirements

Performance, usability, accessibility, and security considerations:

When using all snps for analysis, there tends to be a longer waiting time but the waiting time is usually short due to the reduced dataset and the data is set up in a way where there isn’t a reliance on manipulating the data to display it in a certain format. Sessions used to store information from one request to the other. Session data is stored as files on the server which is useful when dealing with storing a lot of information if cookies can’t. Its used to store and retrieve PCA results. Stores information server side with only with the client having the session ID. There are different routes on the snp analysis for the requirements 3 and 4 (displaying the relevant information about snps of interest accessible via the analyze button and the matrix of pairwise genetic differientation). This allows reduces the time for the results to be generated but also allows for the page to not be crammed too much information and displays it in a clearer format. In a non production environment, the database would need to be secured to avoid attacks where an individual could set themselves as admin and get all the information out of the database (sql injection). The site allows the user to easily go to home button if they are on the initial tab the site is on, if not they can switch to that tab and press home or the genomepop logo.

4. Software Requirements

1. The application should perform a suitable clustering analysis on the genetic data provided. The user should be able to select which populations or superpopulations (i.e. grouped at continental level) to include in this analysis.

2. The application should perform a suitable admixture analysis on the genetic data provided. The user should be able to select which populations or superpopulations (i.e. grouped at continental level) to include in this analysis.

3. The user should be able to retrieve sample allele and genotype frequencies, as well as information on clinical relevance, if available, for SNPs of interest by selecting either (i) a list with their id(s), (ii) a region as defined by genomic coordinates (chromosome, start and end), or (iii) a list with gene names. The user should be able to select which populations to include in this analysis.

4. If multiple populations are selected in requirement #3, a matrix of pairwise population genetic differentiation should be produced, alongside a suitable visual representation of it. The user should also be able to download a text file with the results obtained. Remember that results and illustrations should be presented in a manner that will help answer biological questions.

5.Functional Requirements

Werkzeug ensures files are safe to use.The site allows the user to look at the graphs for PCA using populations and PCA using individuals of selected populations. The site also allows user to get genotype and allele frequency for selected populations after user inputs snps that they want to get information about using positional information,gene names or rs ids as input. When the user presses the analyze button a new tab opens with the relevant database entries extracted. The user can press calculate FST to get a matrix of pairwise genetic differientation and heatmap representing that data using the same selected populations and same selected snps of interest.

5. Technologies Used

Technologies used, including programming languages, frameworks, database systems, and reasons for their selection.

The backend of the software uses a sql database which was created by extracting key information from the original vcf file which contained all the individuals data. The rs ids onto the snps in the info section of the original vcf and then a vcf was created that only included the data at snps with rs ids. This vcf was used for further annotation using clinvar for clinical annotations, ANNOVAR and SNPEff to identify the region that the snps were located in (ANNOVAR identified genes that snps were located in or between coding genes whereas SNPEff annotated identified the actual region that the snps were in and was used to validate the ANNOVAR information). Chromosome number,Position,REF,ALT information for those filtered snps as well as annotations were put into a csv which was then used to populate the SNP\_Data table. The steps used to create this csv were on linux.PCA and admixture analysis results were generated using the PLINK software and the data generated was put into the pca\_coordinates,individual\_pca\_coordinates and the admixture\_results table of the sql schema.

The sql database was created from the csv using the following statements: 1.CREATE TABLE SNP\_Data(

SNPID INT,

Chromosome TEXT,

Position INT,

ID TEXT,

REF TEXT,

ALT TEXT,

GeneName TEXT,

GeneType TEXT,

ClinicalSignificance TEXT,

DistanceToAdjacentGenes TEXT,

ExonicFunction TEXT,

ACB\_Frequency REAL,

ASW\_Frequency REAL,

BEB\_Frequency REAL,

CDX\_Frequency REAL,

CEU\_Frequency REAL,

CHB\_Frequency REAL,

CHS\_Frequency REAL,

CLM\_Frequency REAL,

ESN\_Frequency REAL,

FIN\_Frequency REAL,

GBR\_Frequency REAL,

GIH\_Frequency REAL,

GWD\_Frequency REAL,

IBS\_Frequency REAL,

ITU\_Frequency REAL,

JPT\_Frequency REAL,

KHV\_Frequency REAL,

LWK\_Frequency REAL,

MSL\_Frequency REAL,

MXL\_Frequency REAL,

PEL\_Frequency REAL,

PJL\_Frequency REAL,

PUR\_Frequency REAL,

SIB\_Frequency REAL,

STU\_Frequency REAL,

TSI\_Frequency REAL,

YRI\_Frequency REAL,

AFR\_Frequency REAL,

AMR\_Frequency REAL,

EAS\_Frequency REAL,

EUR\_Frequency REAL,

SAS\_Frequency REAL

)

2.CREATE TABLE admixture\_results (

ResultID INTEGER PRIMARY KEY AUTOINCREMENT,

PopulationID INTEGER,

ancestry\_1 REAL,

ancestry\_2 REAL,

ancestry\_3 REAL,

ancestry\_4 REAL,

ancestry\_5 REAL,

FOREIGN KEY (PopulationID) REFERENCES populations(PopulationID)

)

3.CREATE TABLE individual\_pca\_coordinates (

SampleID INTEGER PRIMARY KEY,

PopulationID INTEGER,

pc1 REAL,

pc2 REAL,

pc3 REAL,

pc4 REAL,

pc5 REAL,

pc6 REAL,

pc7 REAL,

pc8 REAL,

pc9 REAL,

pc10 REAl,

FOREIGN KEY (PopulationID) REFERENCES populations (PopulationID)

)

4.CREATE TABLE pca\_coordinates (

CoordinateID INTEGER PRIMARY KEY AUTOINCREMENT,

PopulationID INTEGER,

pc1 REAL,

pc2 REAL,

pc3 REAL,

pc4 REAL,

pc5 REAL,

pc6 REAL,

pc7 REAL,

pc8 REAL,

pc9 REAL,

pc10 REAL,

FOREIGN KEY (PopulationID) REFERENCES populations(PopulationID)

)

5.CREATE TABLE populations (

PopulationID INTEGER PRIMARY KEY AUTOINCREMENT,

PopulationName TEXT NOT NULL,

is\_Superpopulation INTEGER NOT NULL CHECK (is\_Superpopulation IN (0,1))

)

6.CREATE TABLE sqlite\_sequence(name,seq)

The web application should be able to provide the user with admixture results: The user can select to do a per sample analysis for the pca, per individual analysis for the pca, admixture analysis and snp analysis.

Os module helps find the database relative to the main.py.The modules that deal with the temporary images in memory that are made in binary format by matplotlib/plotly which can create plots, these binary images are stored in a byte object .base64 encodes those images into a UTF-8 string which can be embedded directly into html/css or transmitted via json.css was also used to help style the website. Database connection is established and stored in object g. The close\_db function registered to teardown\_appcontext, the database connection stored in g gets closed which stops connection leaks, when the connection is stopped. A function is defined to use pca\_form prepares pca data to be on the pca form page, 2 sql queries looks in the PopulationID and PopulationName columns from populations table in the sql schema: one query gets populations that are 0 as populations and another query gets superpopulations that are 1 in the is\_Superpopulation column. When this data is retrieved, the template for pca\_form is rendered and the population data and superpopulation are put into the template separately using jinja2 templating syntax which allows context variables to be passed to be passed to the form and iterated over and it put checkboxes next to them.

PCA using populations: A function called perform\_pca was created to perform pca between populations. get\_db that connects to the database, placeholders are set to prepare the query into a string format and then the string the user enters is assigned to the variable query. The selected\_populations has to match number of placeholders. There is a function that takes selected populations as input, it connects to the database and it links the 2 tables- the SampleID,PC1,PC2 columns from the individual\_pca\_coordinates and from the populations table, it takes the PopulationName and the 2 tables are linked using the PopulationID column which both tables have , 'WHERE' means only includes relevant information from the table.This information is then returned from the population\_name,SampleID ,pc1 and pc2 data and the return statement then passes this information to the plot\_pca function which uses the pc1 and pc2 are used to make a graph. Image saved as a PNG,encoded to base64 and returned to the results.html to be observed. User redirected to a results page,results should be displayed using the format in results.html (Population name, Coordinate ID, pc1 and pc2 will be shown as well as plot). If no results passed to the plot\_pca function then error message displayed.

PCA using individuals: If the user selects the option to perform the pca using individuals there is a function in the code which has placeholders set to prepare for the query into a string format. The selected\_populations has to match number of placeholders. There is a function that takes selected populations as input, it connects to the database and it links the 2 tables- the CoordinateID,PC1,PC2 columns from the pca\_coordinates and from the populations table, it takes the PopulationName and the 2 tables are linked using the PopulationID column which both tables have , 'WHERE' means only includes relevant information from the table

Admixture:

Connection to the database is established, gets a list of populations to be passed to the template for the admixture form which will display these populations with checkboxes.When the user goes to a admixture page, it shows the admixture form. Javascript is used to collect options user selects on the form and sends these to the server as a post request without the page refreshing. Analyze\_admixture function retrieves the admixture data for the user selected populations/superpopulations. This data is put into a format which plotly can use to plot the admixture results, this formatted data is sent as a json response to the Ajax request.

Improved explanation:When a user accesses the admixture analysis page, they are presented with a form populated with population options, derived from a database connection that fetches a list of populations. These are displayed with checkboxes through the admixture form template. JavaScript on the client side captures the user's selections and sends them as a POST request to the server without refreshing the page. The analyze\_admixture function on the server side then retrieves this JSON data, which includes the IDs of the selected populations. Using the get\_db function, it queries the database for admixture results specific to these populations. The SQL query is dynamically constructed with placeholders to secure precise ancestry proportions for the selected populations, such as ancestry\_1, ancestry\_2, etc., from the database. Once retrieved, this data is formatted to be compatible with Plotly's graphing requirements, assigning unique colors to each ancestry proportion and presenting them as individual bars in a stacked bar chart. This visualization technique enables a detailed representation of genetic ancestry by population. The prepared graph data is then sent back in JSON format as a response to the Ajax request initiated by the frontend. This end-to-end process, from form display, user selection, data fetching, to interactive chart rendering, offers a comprehensive insight into the genetic makeup of populations, facilitating an engaging user experience in exploring genetic admixture analysis.

Displaying SNP information based on user input:

A list of predefined population codes was made. When a user submits the SNP analysis form, the application checks the form's status and retrieves the selected action ("Analyze" or "Calculate FST"), along with the list of chosen SNPs and populations. An SQL query is then constructed to fetch relevant SNP data from the database based on the user's selections. The fetched data is organized into a list of dictionaries, each representing a single SNP. Initially, the application displays a form with basic SNP information, such as ID, position, and gene name. Upon user submission, a POST request triggers the database query to retrieve data corresponding to the user's selections. The user is then directed to a results page populated with detailed information about the selected SNPs, including chromosome position, RS ID, gene name, genotype and allele frequencies, and clinical annotations. Two routes are defined for the SNP analysis: one for displaying the initial form and another for presenting the results, including a pairwise genetic differentiation matrix if "Calculate FST" is selected. This approach ensures a seamless transition from initial data display to detailed analysis based on user inputs.

Matrix of pairwise genetic differientation: calculate\_fst\_from\_averages gets allele frequencies for the selected populations after the allele frequencies (the information after semicolon in the columns that include the word Frequency ) is extracted

Generate heatmap function uses imshow from plotly.express to show the matrix data

Calculate\_fst\_improved assigns weights based on sample size for each population. Both total and within population heterozygosity are calculated and then Fst formula is used which is (total heterozygosity minus within population)/total heterozygosity

Calculate\_fst\_from\_averages gets the data in the format of multiple dictionaries, one for each snp with allele frequencies across all populations in data\_dicts,selected populations is the list of populations and a population\_sample\_size dictionary that connects population names to sample sizes. These 3 are passed to the function and empty dictionary is created to store fst for the chosen populations. Combinatinations creates a tuple of each population pair. Dictionary created for allele frequencies of each population of the current pair, key is a population name and value is a list with nothing in it.A loop goes over each snp in the list of dictionaries and gets allele frequencies for them, if it’s not NAN. Average allele frequencies is calculated for each population, that entry is empty then sets fst for that pair as NAN. The FST matrix is created and both row and column names are population codes and fst\_results values fill the matrix. Diagonal of the matrix set to zero for self comparisions. This matrix data is formatted into a list of dictionaries so theres a dictionary per row to put on the html page.

Generate heatmap function is called to use the matrix as input and unique file name is generated for the csv with the matrix data. Then json data for the heatmap is returned, the csv filename and the dictionaries to be displayed as fst matrix on the html page.

6. Implementation Details

How each of the required analyses is implemented, the algorithms chosen, and how users can perform these analyses through the web interface.

7. User Guide

1.Go to the front page where you will see options of homes of different analysis

A screenshot of a computer

Description automatically generated

2.Home button as well as the GenomePopAnalytics logo can take user to front page(shown above)

3.Analysis tools takes user to this page

A screenshot of a computer

Description automatically generated

4.About takes user to this page

A screenshot of a computer

Description automatically generated

5.Contact takes user to this page

6. If the user goes to back to the front page using the home button, they will find

A screenshot of a computer

Description automatically generated

7.The info button under PCA Analysis takes user to this page:

A screenshot of a computer

Description automatically generated

1. The info button under Admixture analysis takes user to this page:

A screenshot of a computer

Description automatically generated

1. The info button under SNP Analysis takes user to this page:

A screenshot of a computer

Description automatically generated

1. If you go back to the home page, you can see the get started button which redirects user to the analysis tools page
2. Press on the name of what analysis you want to pick to be able pick it. You can also do this without pressing the get started button by pressing the name of the analysis you would like to pick which directs user to the page for that analysis.

A screenshot of a computer

Description automatically generated

12.This is the page the user will be taken to if they click on the PCA Analysis.The PCA analysis pages fives the user the option to select populations and/superpopulations to include in analysis. Theres also an option to do an PCA using the data of individuals of the each selected population rather than the values representative of that population.

A screenshot of a computer

Description automatically generated

13.An example of a PCA plot when the PCA is using population data:

A screenshot of a computer

Description automatically generated

An example of a PCA plot generated using individuals data (each member of the selected population)

A screenshot of a computer

Description automatically generated

14. If user goes to the presses the Admixture Analysis option by clicking on the name “Admixture Analysis”, the user will be directed to this page where they can select populations or superpopulations to include in the Admixture analysis.

A screenshot of a computer

Description automatically generated

15. This takes user to the page with an interactive barchart with results, on top of the barchart there are several options to adjust the view of the plot e.g. zoom in/out etc.

A screenshot of a computer

Description automatically generated

15. If user goes back to the home page and presses on the SNP Analysis name, it will take the user to the snp analysis page where they can select populations to include in the snp analysis (they can select all by pressing the Select All button and deselect by pressing Deselect All)and the user can select snps of interest by either typing in rs ids, gene names,chromosome positions. They can also select snps of interest using tick boxes next to table , this table contains the columns:Position,ID and GeneName columns from the SNP\_Data table. There are buttons to select and deselect all the boxes for the snps selected as there is a select all button next to gene name search box and the deselect under the search box that says search position.

A screenshot of a computer

Description automatically generated

16. After ticking the SNPs of interest and the populations, the user has the option to get information about sample genotype and allele frequencies

A screenshot of a computer

Description automatically generated

17.After pressing the Analyze button, the user should see the relevant snpid,rsid,ref,alt,gene name,gene function and relevant annotations as well as sample and allele frequencies for selected populations on a new tab.

A screenshot of a computer

Description automatically generated

18. If the user presses on the Calculate FST button, they will get directed to a page in a new tab with a heatmap, fst values in a matrix and a link to download the fst matrix in a csv and a link to download the heatmap, theres a link back to the snp analysis page:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

8. Developer Guide

Extending the software, including code structure and setup instructions:

Extend the software to include additional analyses e.g. can include more than one clustering method, use postgresql to improve security by using role-based access control, SSL/TLS encryption, and data encryption and postgresql supports more powerful analysis. Modify code to be compatible with postgresql.

9. Limitations and Future Work

Limitations of the prototype and potential areas for future development and improvement:

In terms of analysis, PCA components plotted (pc1 and pc2) didn’t capture a significant portion of the variance for both the individual and population level PCA. Using less snps for the analysis also means that the data could be misrepresenting the extent of genetic diversity between the populations.

10. Conclusion