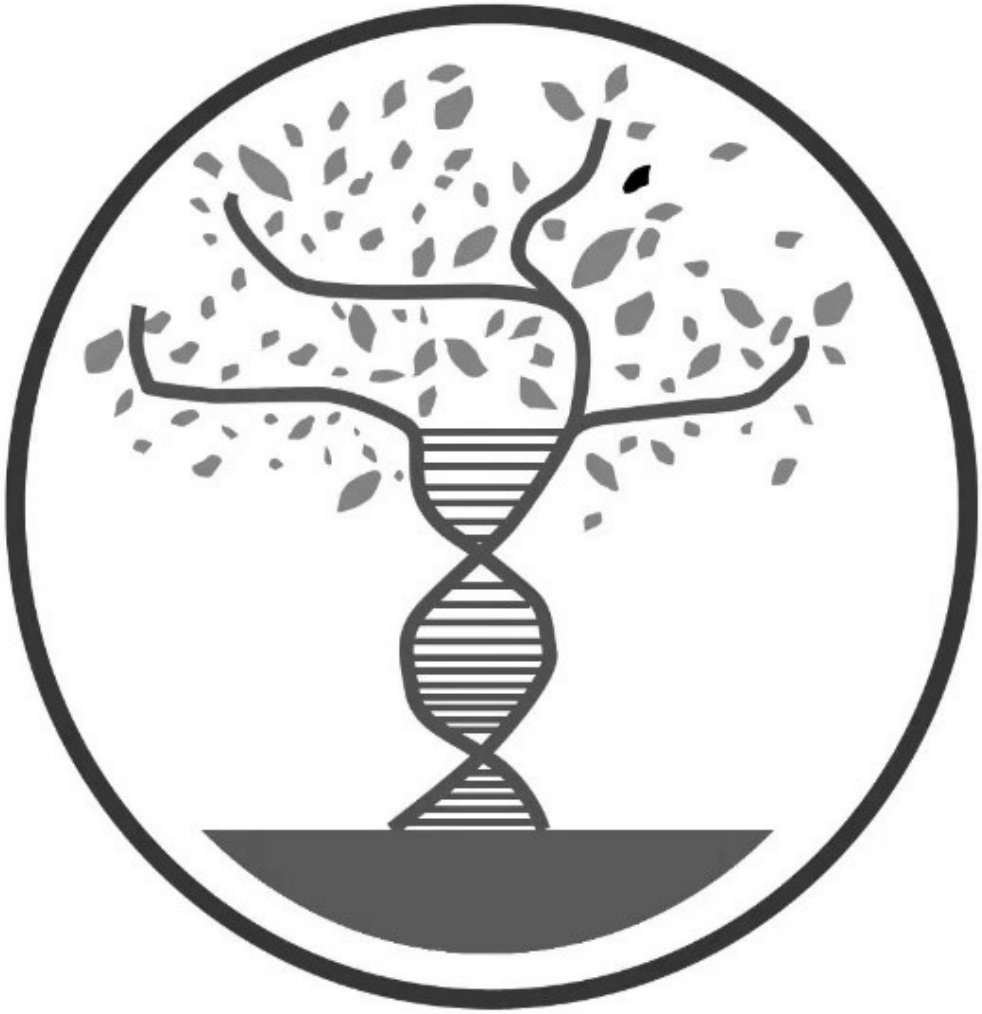
SNPGLOBAL



By: Gabriel Marengo, Ervin Rexhepi, Zainab Haybe, and Zibo Cong

Table of Contents

Aim of SNPGlobal

Single nucleotide polymorphisms are the most abundant form of DNA variation in humans (Marth et al., 1999). SNPs are used as genetic markers for ancestry and to understand gene functions related to disease (Kwok and Gu, 1999). The advancement of high throughput sequencing resulted in vast amount of genomic data available to analyze. The aim of the SNPGlobal website is to provide biologists and researchers a resource to obtain information on single nucleotide polymorphism in humans. SNPGlobal allows users to obtain basic information and summary statistics for SNPs by searching based on gene name, gene ID, rsID, or genomic position.

Running SNPGlobal

In order to run the website locally, please use the following script to download the required packages on your terminal

```
$ pip install Flask==2.0.2
$ pip install Flask-SQLAlchemy==2.5.1
$ pip install Flask-WTF==1.0.0
$ pip install kaleido==0.2.1
$ pip install numpy==1.22.2
$ pip install pandas==1.4.0
$ pip install plotly==5.6.0
$ pip install plotnine==0.8.0
$ pip install scikit-allel==1.3.5
$ pip install WTForms==3.0.1
```

Then use the following code to obtain the files from the GitHub page that can run the website.

```
$ git clone https://github.com/ErvinRex/TheBestGroup.git
$ cd Front_End
$ python3 main.py
```

Design Philosophy

Software Architecture

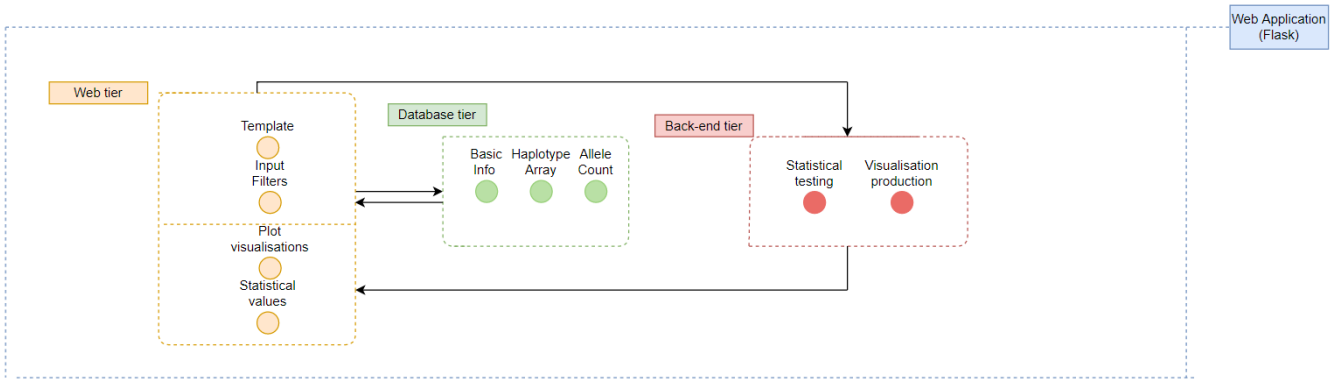


Figure 1: Software architecture visualised with three specific tiers and their respective layers that form the Web Application.

The software architecture follows an N-tier model architecture, consisting of tiers to separate functionalities across the application. There are three tiers, the web, database, and back-end. Each with multiple layers hosted on the same tier. For each layer, responsibilities and dependencies are separated to provide functionality to one tier. The highest tier in this model is the web, with the two lower tiers unable to call to one another, as the web is responsible for calling on both. The web is responsible for acquiring the input filters from the user through the HTML templates navigated by the user and calling them to the database to query for specific genes, regions or rsIDs. The database holds the necessary data and was populated using SQLite3. Querying is done with SQLAlchemy, which then feeds back the queried data to the web to display on a HTML template. When summary statistics are requested, the web further calls on the database which queries respective database layers to feed back to the web. The web then calls on the back-end tier to carry out appropriate statistical tests through Python 3.10 with the data queried for. The back-end provides interactive visualisations of the query that are then output onto the web in the form of HTMLs alongside appropriate statistical results.

Website Structure

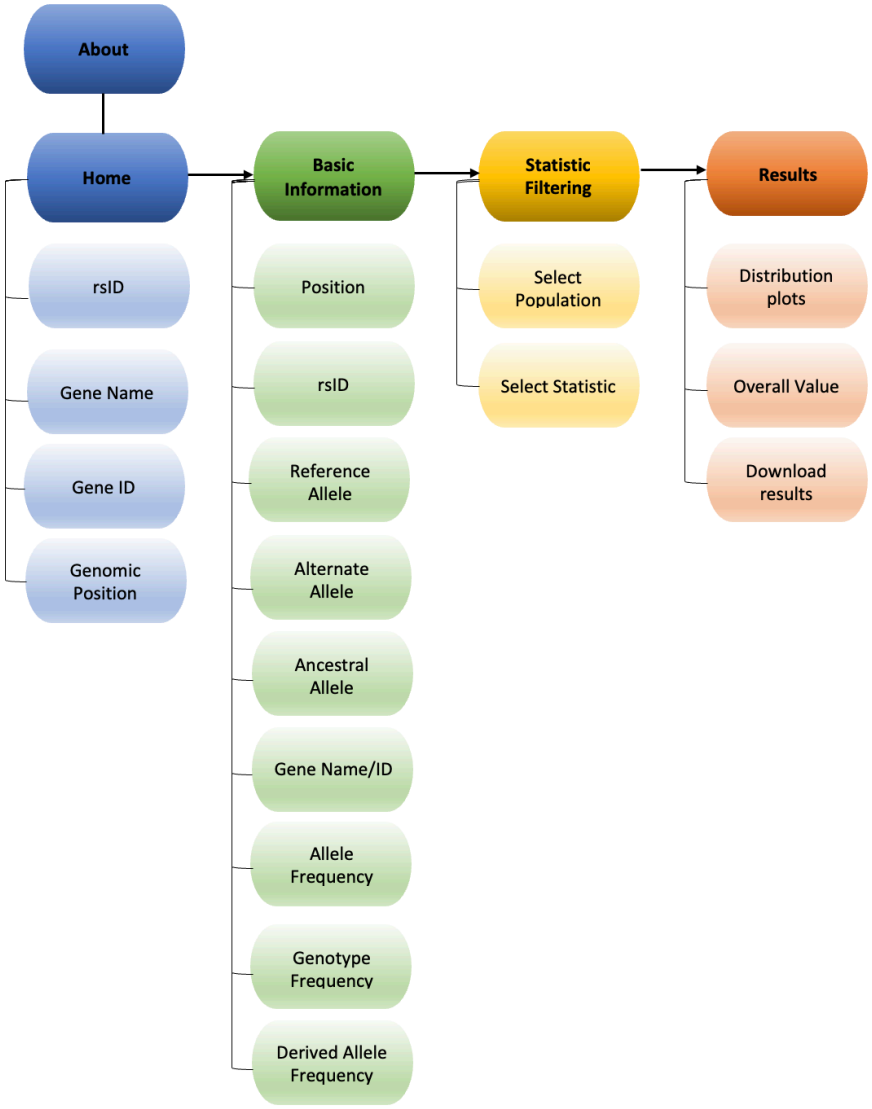


Figure 2: A schematic diagram of the website with the pages and the information on each page. The arrows indicate directional relationship from one page to the next. The line connects the information on each page.

The home page allows users to search based on the gene name, gene ID, rsID, and genomic position. Once the user submits their search criteria, they are directed to the basic information page which contains a list of SNPs that match the user's input. The basic information page contains the position, rsID, gene name, gene ID, reference allele, alternate allele, ancestral allele, allele frequency, genotype frequency, and the

derived allele frequency. The frequencies are provided for each of the 5 populations the database contains. The user clicks the 'Go' buttons to be directed to a selection page where they can choose the population and summary statistic of interest. Once the user clicks the 'Go' button, they will be directed to the final page which contains the summary statistics plotted using sliding windows and they are able to download the data as a csv file. To begin a new search, the user needs to select the home button in the top bar of the website. The About page contains some basic information regarding the SNPGlobal website and information on the summary statistics.

Database

Data Collection

The SNP information for chromosome 22 was obtained from the 1000 Genomes 30x on GRCh38 dataset from the phased vcf files, genotypes annotation text file, and the sample information csv (https://www.internationalgenome.org/data-portal/data-collection/30x-grch38). The phased vcf contained the position, chromosome, reference allele, alternate allele, and genotypes for each sample. The sample information file contains the population code for each sample name. The annotation text contained the gene name and gene ID for some of the SNPs. (Byrska-Bishop et al., 2021)

The rsID and the ancestral allele was obtained from Ensembl FTP download using the homo_sapien_chr22.vcf.gz file (http://ftp.ensembl.org/pub/release-105/variation/vcf/homo_sapiens/). The missing genes and the gene alias' for chromosome 22 were obtained using the ensemble Biomart database (https://www.ensembl.org/biomart/martview/3b0ca5dd705d8c4fc8f777f108b14b8d). (Howe et al., 2021)

Database Schema

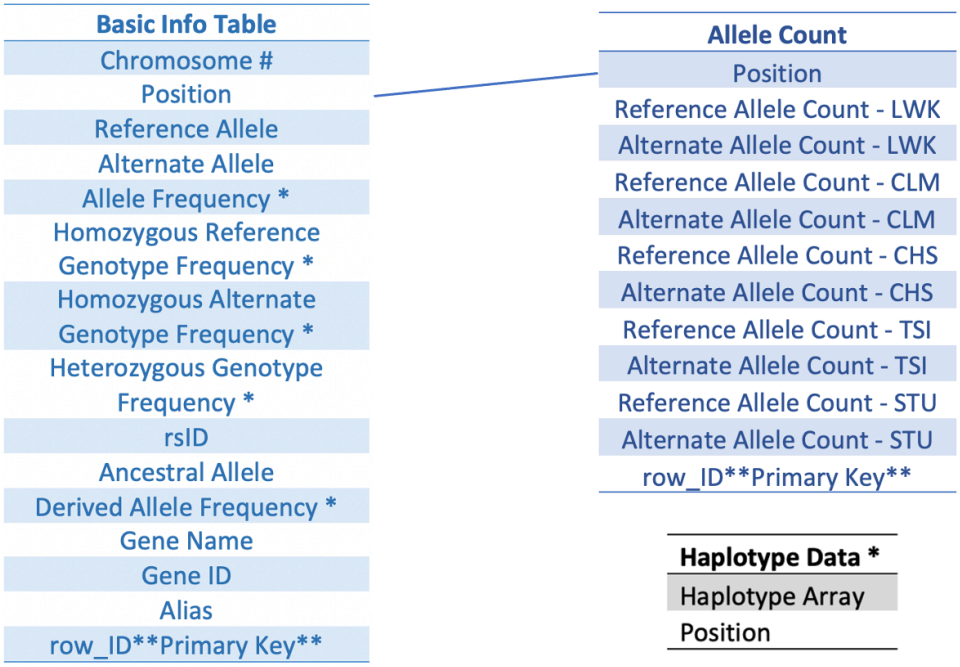


Figure 3: Database schema showing the Basic Information sql table, Allele count sql table, and the haplotype csv. The line connects the basic Information and Allele count table based on position. The * indicates the value was repeated for each of the 5 populations.

The VCF was converted to a pandas dataframe which contained the chromosome, position, reference allele, alternate allele, and if variant is SNP using Scikit-allel package. The dataframe was filtered to only contain SNP by removing indel and other variants using pandas. The allele counts for each population were obtained as a dataframe and used to calculate the genotype frequency and allele frequency. The reference and alternate allele count array was stored for each of the populations using Scikit-allel package. The basic information and the allele count dataframes were converted an SQL table and then stored in the database. The haplotype array for each population was also stored as a csv file as it was large and easier to access for the statistics.

The populations were selected to contain one population from each of the 5 superpopulations. The African ancestry population was the Luhya in Webuye, Kenya (LWK). The East Asian ancestry population is the southern Han Chinese, China (CHS). The American ancestry population in the Colombian in Medellin, Colombia (CLM). The European ancestry is the Toscani in Italy (TSI). The South Asian ancestry population is Sri Lankan Tamil in the UK (STU). The sample information csv was used to obtain a list of sample names for each of the 5 populations. The vcf annotation csv and the Biomart csv were filtered using pandas to contain the gene names, gene ID's and the alias'.

Statistics Information

The summary statistics carried out by this application include: F_{ST} , Tajima's D, Watterson's estimator of Diversity and Haplotype diversity.

F_{ST}

The variance of allele frequencies between populations is calculated for a given locus (Holsinger & Weir 2009). This calculation is carried out using the scikit-allel package which takes advantage of numpy arrays extrapolated from the "Allele Count" database. The average F_{ST} is estimated in moving windows across the queried region using the method of Hudson (1992). The value of FST ranges from 0 to 1. A value of zero indicates full panmixis (random mating), which means that the two populations are freely interbreeding. A value of 1 means that the two populations do not share any genetic diversity. The FST value largely depends on the distance of the populations that are chosen for comparison.

Tajima's D

Tajima's D is the difference between two measures of genetic diversity: the average number of pairwise differences and the number of segregating loci (Tajima 1989). Both are scaled in medium-sized constant-size populations, so their expectations are the same. Results of Tajima's D can be used to explain selection experienced by the population. This calculation is carried out using the scikit-allel package which takes

advantage of numpy arrays extrapolated from the “Allele Count” database. The Tajima’s D value is calculated over a queried region in sliding windows. The positive and negative values of Tajima’s D respectively represent the abundance of rare alleles, which can infer the recent history of the population. A very rough significance rule is that a value greater than +2 or less than -2 may be significant, but it does not actually represent a critical value for a significance test (Simonsen, Churchill & Aquadro 1995).

Tajima’s D = 0: average heterozygosity equal number of polymorphic loci. The observed variation is like the expected variation, and the population evolves according to the mutation-drift equilibrium. There is no evidence of selection. Tajima’s D < 0: Number of haplotypes less than number of polymorphic loci (lower mean heterozygosity). Rare alleles are present at high frequencies. The recent selective sweep, the population expansion after the recent bottleneck. Tajima’s D > 0: Number of haplotypes more than number of polymorphic loci (more average heterozygosity). Rare alleles exist at low frequencies, balanced selection, or sudden population shrinkage.

Watterson’s estimator of Diversity

Watterson Estimator of diversity estimates the genetic diversity of a population by counting the number of polymorphic loci and was developed by Margaret Wu and GA Watterson in the 1970s (Rohlfs 2019). It uses population nucleotide diversity as a measure of "population mutation rate" (Ferretti & Ramos-Onsins 2015). This calculation is carried out using the scikit-allel package which takes advantage of allele count & positional numpy arrays extrapolated from the “Allele Count” database. The Watterson’s estimator of Diversity value is calculated in windows over a queried region. Watterson's estimator is commonly used because of its simplicity. The variance of the estimator decreases with increasing sample size or recombination rate. The introduction of nucleotide diversity for comparison with the value of the Watterson estimator is the basis for Tajima's D, which can be used to infer evolutionary mechanisms. The Watterson estimator can be influenced by population structure and is downward biased in exponentially growing populations or where multiple mutations can cover each other.

Haplotype diversity

Haplotype diversity is an important indicator to measure the degree of variation in a population. Haplotype diversity refers to the frequency at which two different haplotypes are randomly selected from a sample (Jong et al. 2011). A population with high haplotype diversity indicates high genetic diversity and rich genetic resources. This calculation is carried out using the scikit-allel package which takes advantage of numpy arrays extrapolated from the “Haplotype Array” database. The Haplotype diversity value is calculated in windows over a queried region. If the haplotype diversity of a selected region is relatively low, it can be considered that this region with relatively conserved variability is associated with certain diseases, which is very helpful for disease research (Manolio & Collins 2009).

Derived Allele Frequency

Derived allele is a new allele created by a mutation, which frequency is at least 5% in population frequency (Gorlova et al., 2012; Yan et al., 2021). The opposite of derived allele is ancestral allele, which is an allele inherited from an ancestor. Ancestral allele can often be confirmed by aligning the genetic sequences of closely related species, such as humans and gorillas. Derived allele is usually assessed by derived allele frequency (DAF), that is, how high is the proportion of derived allele in the population. DAF can be used as the basis for many studies, such as disease risk, population history and environmental influences, and differences between different subpopulations (Keinan et al., 2007; Gorlova et al., 2012).

Gorlova et al. (2012) noted that derived alleles or rarer alleles were more likely to be associated with risk than ancestral alleles. When the ancestral allele frequency is very high, this locus can be highly conserved and highly associated with certain diseases. However, the derived allele does not necessarily have to be low frequency, and when it is high frequency, it indicates that the compilation of this locus may be neutral (late-onset diseases) or protective allele (Gorlova et al., 2012).

Technologies Utilized

Scikit-Allele

Scikit-allel is a comprehensive package that can extract information from vcf files, conduct statistical tests, and plot the results. The extensive functions of the package and the ability to easily integrate it with the flask-app code is the main reason scikit-allel was used. The *allel.vcf_to_dataframe()* converts the information from the vcf into a dataframe. The *allel.read_vcf()* converts the vcf into a dictionary which includes the sample names, position, and genotype for the samples of your choice. The *allel.GenotypeArray()* obtains the genotype array which was then used to obtained the allele counts and haplotype arrays. (Alistair, M. & N. Harding, 2017)

The Scikit-Allel package was also used for each of the summary statistics on SNP global. The *allel.watterson_theta()* was used to calculate the overall Watterson's estimator value of the region and the *allel.windowed_watterson_theta()* was used to calculate the value using sliding windows. The *allel.tajima_d()* was used to calculate the overall Tajima's D value of the region and the *allel.windowed_tajima_d()* was used to calculate the value using sliding windows. The *allel.hudson_fst()* was used to calculate the overall Fst value of the region and the *allel.windowed_hudson_fst()* was used to calculate the value using sliding windows. Finally, the *allel.haplotype_diversity()* was used to calculate the overall haplotype diversity value of the region and the *allel.moving_haplotype_diversity ()* was used to calculate the value using sliding windows. (Alistair, M. & N. Harding, 2017)

Flask

The website makes extensive use of Flask. Flask is a microframework for web development that allows for a simple core application to be built relatively quickly compared to other frameworks such as Django. However, Flask is also capable for developing extensive applications, offering flexibility to developers, for example if developing a prototype application in a short period of time. Flask therefore was a suitable framework for this project. Flask uses views to associate functionality with

routes on the website. The application was designed such that most of the views corresponded to a webpage. (Grinberg, 2018)

The following were the views used:

- **Home:** Renders home.html which displays the forms for the user to search for rsID, Gene, Gene ID and Positions on the chromosome. Each search form is associated with the Thread view.
- **About:** renders the about.html script which displays information about SNPglobal.
- **Documentation:** Renders the documentation.html script which displays the documentation for SNPglobal.
- **Thread:** Renders the basic_search.html script which displays the positions, rsID, gene name, reference allele, alternate allele, alternate allele frequency, genotype frequency and derived allele frequencies for each population.
- **Stats:** Gets the Allele counts associated with the users search query for the statistical tests; Tajima's D, Wattersons estimator Theta and Fst. Renders the pops_stats_selection.html script which displays the options for the user to select the statistical tests and populations of interest.
- **Stats_2:** Runs the statistical tests; Tajima's D, Wattersons estimator Theta and Fst using the allele counts data from the stats view. Runs the statistical test Haplotype diversity using the haplotype array dataframes from the haplodb file. Creates objects for the overall statistical test values, visualizations, and dataframes for plots on front end.
- **Stats_3**: Renders the stats_output.html script which displays the Plotly plots, overall statistical values for the users search query and the ability to download the dataframes used for the graph plotting.
- **Download views:** convert statistical results from dataframes to csv for user download.

Flask: WTForms

Flask supports numerous extensions, the extension Flask WTF, which integrates WTForms with Flask, was utilised for registering user inputs on the 'home' and 'select population and statistical tests' pages. On the 'home' page, the StringField field handles rsID, Gene name, Gene ID, Gene Alias searches and the IntegerField field handles positional searches. On the 'select population and statistical tests' page, the SelectMultiple field was used for both the populations selection and statistical tests selection. The SubmitField was used to check for user submission on the front-end of the application. The information registered by WTForms formed the basis of the database search queries and the application logic related to the statistical tests.

Flask: SQL Alchemy

The Flask extension Flask-SQLAlchemy was used as an object-relational mapper in conjunction with SQLite3 database to query the BasicInfo and AlleleCount tables. Interfacing with the SQLite3 database was enabled by SQLAlchemy models. SQLAlchemy models allow for SQL querying through pythonic scripting, which due to the time constraints of the project was a more appealing option compared to SQL scripting. SQLAlchemy also supports pagination which increased the speed at which the basic search results are returned. (Bayer, 2012)

Jinja2

The Jinja2 templating engine was also chosen for this project due to its similarities with Python scripting. Jinja2 provides template inheritance and has the ability to transfer portions of the applications logic to the HTML templates. These features reduce repetitive coding and the complexity of the template design process respectively. As an example, using template inheritance, a navigation bar was implemented in the base.html file, which was inherited by the rest of the applications HTML templates, making the navigation bar visible throughout the website.

Example of Jinja in HTML template

```
{% if message_for_user  %}

        <p>{{ message_for_user }}<p>

{% endif %}
```

Blueprints were registered with the application factory to provide modularity to the Flask view functions. The application does not currently spread views across modules, however future development would be able to capitalise on this feature.

CSS

CSS was used to style the website, which provides a more intuitive user experience. It also provides functionality, for example by highlighting rows in the table returned on the ‘basic search outputs’ page. This is useful because the table has many columns, allowing row elements to be more easily compared, for instance, across populations. (Duckett, 2011)

Plotly

Visualisations created for each individual statistical test are designed to allow precise positional values for a queried rsID/region/gene. The interactive viewer is produced using the plotly package and allows for multiple populations or multiple population combinations to be plotted simultaneously. This allows for accurate comparisons between populations in each statistical test, with the ability to also hover over each position to obtain the respective statistical value for that SNP. Plots can also be zoomed in and out of to see differences more accurately in statistical values per SNP per population. Static images of plots can also be downloaded as a JPEG to be used for review and other research purposes. (Inc, 2015)

Opportunities for future development

Expanding the database

Considering the limitation of time and computer computing power, our application database only contains the chromosome 22 information of several populations. So, our statistics must be biased and cannot fully cover the entire human genome. However, when developing this application, we did not use tools such as VCFTools for auxiliary processing, we directly used Python to process VCF files. This makes sense for subsequent developers, as they only need to find VCF files that cover more complete information to expand the database with existing code (of course a better computer is required).

Increasing statistical Tests

In this application, we only show four statistical results such as Tajima's D and perform simple data visualization. However, we did not perform an overall statistical analysis such as comparing exons and introns. In subsequent application development, developers can add some comparisons such as the above. In addition, more statistical methods can also be added to the application, such as Fu & Li's D, etc. More statistical methods help users to obtain more detailed and useful data. The website could also provide additional statistical features such as comparing across genes or chromosomal positions.

Links to external websites

Our purpose is to provide a user-friendly information for querying specific SNPs or specific genome coordinates. This application is designed to provide frequency information and statistical results. Detailed information about genes, including products, related diseases, etc., are not provided in this application. Therefore, in the future development, corresponding detailed explanations from other websites can be provided for this. For example, when the application returns a SNP, the user can also get the link of the corresponding SNP of the SNP in databases such as NCBI (www.ncbi.nlm.nih.gov) or GeneCards (www.genecards.org), so that they can know

more about the information of the related SNP and the information of the gene to which the SNP belongs.

Option for the sliding window size

In our current version of the app, the size of the sliding window is fixed. This makes some statistical methods not completely in accordance with the user's wishes. Subsequent software development can focus on developing variable-sized sliding windows and presenting options to the user.

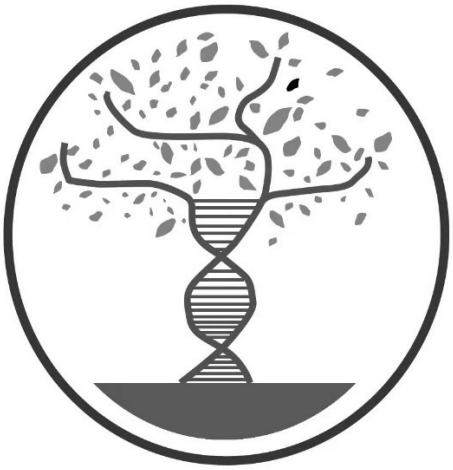
Flow of the website

The website has a linear flow and uses global variables, meaning that the user must return to the home page and progress through each page in order to view the statistical outputs. This was a consequence of time constraints; future development could implement Flask Sessions and Redis to resolve this aspect of the website. The website could include the ability to search for genes, gene IDs and chromosome positions within the ‘statistical output’ page to streamline this part of the application.

Database: Haplotype Array

The code for getting the alleles counts was inefficient due to issues with accessing items from the SQLAlchemy query objects. This would be problematic when scaling up from five populations to, for example 26, as it would be very repetitive. Getting the haplotype arrays is also inefficient as the data is fragmented into separate data frames for each population, resulting in repeated code.

Logo



The design of our logo is inspired by the differences and similarities between different subpopulations. Each subpopulation has a different statistical value, but the subpopulations are quite close. The DNA tree in the logo represents human migration and evolution. The base and the trunk respectively represent that we all come from the same starting point and have a common ancestor. The branches represent different subpopulations, we are separated by migration and live in different environments. Each leaf represents a different individual in the database, on the same branch (in a subpopulation), they are more like each other. The outermost circle represents that, despite so many differences between subpopulations, we are still one family, with the same home, the Earth.

References

Alistair, M. & N. Harding. 2017. cggh/scikit-allel: v1.1.8 (Version v1.1.8). Zenodo. http://doi.org/10.5281/zenodo.822784

Bayer, M., 2012. SQLAlchemy. In A. Brown & G. Wilson, eds. The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks. aosabook.org. Available at: "http://aosabook.org/en/sqlalchemy.html"

Beerli, P. (1998) 'Estimation of migration rates and population sizes in geographically structured populations', p. 15.

Byrska-Bishop, M. et al (2021). High Coverage Whole Genome Sequencing of the Expanded 1000 Genomes Project Cohort Including 602 Trios. SSRN Electronic Journal,.

Duckett, J., 2011. HTML & CSS: design and build websites, Wiley Indianapolis, IN.

Dung, S.K. *et al.* (2019) 'Illuminating Women's Hidden Contribution to Historical Theoretical Population Genetics', *Genetics*, 211(2), pp. 363–366. doi:10.1534/genetics.118.301277.

Elhaik, E. (2012) 'Empirical Distributions of F ST from Large-Scale Human Polymorphism Data', *PLoS ONE*, 7(11), p. e49837. doi:10.1371/journal.pone.0049837.

Fan, P. *et al.* (2021) 'An approach for estimating haplotype diversity from sequences with unequal lengths', *Methods in Ecology and Evolution*, 12(9), pp. 1658–1667. doi:10.1111/2041-210X.13643.

Ferretti, L. and Ramos-Onsins, S.E. (2015) 'A generalized Watterson estimator for next-generation sequencing: From trios to autopolyploids', *Theoretical Population Biology*, 100, pp. 79–87. doi:10.1016/j.tpb.2015.01.001.

GeneCards - Human Genes | Gene Database | Gene Search (no date). Available at: https://www.genecards.org/

GORLOVA, O.Y. *et al.* (2012) 'Derived SNP Alleles are used More Frequently than Ancestral Alleles as Risk-Associated Variants in Common Human Diseases', *Journal of bioinformatics and computational biology*, 10(2), p. 1241008. doi:10.1142/S0219720012410089.

Grinberg, M., 2018. *Flask web development: developing web applications with python*, " O'Reilly Media, Inc."

Holsinger, K.E. and Weir, B.S. (2009) 'Genetics in geographically structured populations: defining, estimating and interpreting FST', *Nature reviews. Genetics*, 10(9), pp. 639–650. doi:10.1038/nrg2611.

Howe, K., Achuthan, P., Allen, J., Allen, J. and Bennet, R., 2021. Ensembl 2021. Nucleic Acids Research, [online] 49(D1), pp.D884–D891. Available at: <https://academic.oup.com/nar/article/49/D1/D884/5952199.

Hudson, R.R., Slatkin, M. and Maddison, W.P. (1992) 'Estimation of Levels of Gene Flow from DNA Sequence Data', *Genetics*, 132(2), pp. 583–589.

Inc., P.T., 2015. Collaborative data science. Available at: https://plot.ly.

Information, N.C. for B. *et al.* (no date) *National Center for Biotechnology Information*. Available at: https://www.ncbi.nlm.nih.gov/.

Jong, M.A. de *et al.* (2011) 'Mitochondrial DNA Signature for Range-Wide Populations of Bicyclus anynana Suggests a Rapid Expansion from Recent Refugia', *PLOS ONE*, 6(6), p. e21385. doi:10.1371/journal.pone.0021385.

Keinan, A. *et al.* (2007) 'Measurement of the human allele frequency spectrum demonstrates greater genetic drift in East Asians than in Europeans', *Nature genetics*, 39(10), pp. 1251–1255. doi:10.1038/ng2116.

Kwok, P. and Gu, Z., 1999. Single nucleotide polymorphism libraries: why and how are we building them?. Molecular Medicine Today, [online] 5(12), pp.538-543. Available at: https://www.sciencedirect.com/science/article/pii/S1357431099016019.

Manolio, T.A. and Collins, F.S. (2009) 'The HapMap and Genome-Wide Association Studies in Diagnosis and Therapy', *Annual review of medicine*, 60, pp. 443–456. doi:10.1146/annurev.med.60.061907.093117.

Marth, G., Korf, I., Yandell, M., Yeh, R., Gu, Z., Zakeri, H., Stitziel, N., Hillier, L., Kwok, P. and Gish, W., 1999. A general approach to single-nucleotide polymorphism discovery. Nature Genetics, [online] 23(4), pp.452-456. Available at: http://clavius.bc.edu/~marth/archive/BI820-4/files/Marth-Polybayes-NG-1999.pdf.

Nelson, M.R. *et al.* (2012) 'An abundance of rare functional variants in 202 drug target genes sequenced in 14,002 people', *Science (New York, N.Y.)*, 337(6090), pp. 100–104. doi:10.1126/science.1217876.

Purfield, D.C. *et al.* (2012) 'Runs of homozygosity and population history in cattle', *BMC Genetics*, 13, p. 70. doi:10.1186/1471-2156-13-70.

Simonsen, K.L., Churchill, G.A. and Aquadro, C.F. (1995) 'Properties of Statistical Tests of Neutrality for DNA Polymorphism Data', *Genetics*, 141(1), pp. 413–429.

Subramanian, S. (2016) 'The effects of sample size on population genomic analyses – implications for the tests of neutrality', *BMC Genomics*, 17, p. 123. doi:10.1186/s12864-016-2441-8.

Tajima, F. (1989) 'Statistical Method for Testing the Neutral Mutation Hypothesis by DNA Polymorphism', *Genetics*, 123(3), pp. 585–595.

Thorisson, G.A. *et al.* (2005) 'The International HapMap Project Web site', *Genome Research*, 15(11), pp. 1592–1593. doi:10.1101/gr.4413105.

Yan, Y.H. *et al.* (2021) 'Confirming putative variants at ≤ 5% allele frequency using allele enrichment and Sanger sequencing', *Scientific Reports*, 11(1), p. 11640. doi:10.1038/s41598-021-91142-1.