**BIO727P - Bioinformatics Software Development Group Project**

**SMART T1D Variant Catalogue**

\* **S**hukri, **M**ohamed, **A**hmad, **R**ashmi & **T**eodora

**Correspondence:**

**E-mail:** enquiry@qmul.ac.uk.

**Address:** Queen Mary University London, Mile End Road, London E1 4NS

Table of Contents

[Table of contents: 2](#_Toc128669061)

[1. Project Philosophy: 3](#_Toc128669062)

[2. Software: 4](#_Toc128669063)

[2.1: Software Architecture 4](#_Toc128669064)

[2.2 Data gathering 5](#_Toc128669065)

[2.3 SQLite Database Creation and Management 5](#_Toc128669066)

[2.4 DATABASE SCHEMA 6](#_Toc128669067)

[3. Website Design and Features 6](#_Toc128669068)

[3.1: Website user guide 6](#_Toc128669069)

[3.2 Data presentation and visualisation 7](#_Toc128669070)

[3.2 Website features: 8](#_Toc128669071)

[4 Integration Layer: 8](#_Toc128669072)

[4.1 Functions to connect user input to database: 8](#_Toc128669073)

[4.2 Statistics 9](#_Toc128669074)

[4.2.1 Linkage Disequilibrium: 9](#_Toc128669075)

[4.2.2 Manhattan plot: 10](#_Toc128669076)

[5. Limitations 11](#_Toc128669077)

[6. Opportunities for development: 12](#_Toc128669078)

[7. References 12](#_Toc128669079)

# Project Philosophy:

The SMART T1D Variant Catalogue is a web application that allows researchers to retrieve Single Nucleotide Polymorphism (SNPs) data from chromosome 6, associated to type 1 diabetes (T1D). There are four search parameters the user can query to retrieve the data: rs-value, gene name, genomic location and/or co-ordinates, or region. Once queried this will return seven data points which are presented on the website which are: SNP ID (rs:ID), genomic positions and/or location, p-value(s), mapped genes(s), variant allele frequency in 5 populations - African, American, East Asian, European and South Asian - functional impact in the form of Sorting Intolerant From Tolerant (SIFT) and Combined Annotation Dependent Depletion (CADD) scores and gene ontology.

To ensure the information is accessible to the user all the data is presented in an interactive table, in which the user can select and download part or the whole table. Additionally, if data for multiple SNPs are returned the user can select the SNPs of interest and calculate the r2 Linkage Disequilbrium (LD) between the selected SNPs and one of three populations. This data is presented in a heatmap for ease of understanding and visualisation. Furthermore, to aid the user in answering biological questions, another feature of this the web application is production of a Manhattan plot of p-values for three populations, African, Asian and European if genomic co-ordinates are queried.

The SMART T1D Variant Catalogue web application is comprised of three main layers, the website which was constructed in Streamlit outlined in more detail in se. The integrated which was done using SQLite3 and standard python programming and the database. The data sourced from GWAS and Ensemble underwent data wrangling was put into a SQLite database, more about this process can be found in the next section.

To access the website, click here: <https://moabdul123-team-safari-smart-web-application-home-99fsig.streamlit.app/>

# 2. Software:

## 2.1: Software Architecture

Diagram

Description automatically generated

*Figure 1: The Software Architecture Diagram shows the components used to build the SMART web application and the interaction between the components when a user inputs a search query. Created with BioRender.com* *(Aoki, 2017)*

## 2.2 Data gathering

The construction of the database used the European Bioinformatic Institute’s (EBI’s) Genome Wide Association (GWAS) Catalog which is a catalog that contain SNPs that are associated with different pathologies and the accompanying meta data (Morales et al., 2018). Ensembl’s Variant Effect Predictor (VEP), which assess the impact of variants (McLaren et al., 2016).

The GWAS Catalog was used to obtain genomic information of variants from all know association of T1D. The search term *type 1 diabetes mellitus* was queried and the MONDO\_0005147 entry was used as it also contained data from three population and had a large sample size of 59,527. The different populations were selected so the genetic variant between the populations could be explored and compared with each other. This data set contained information such as rs-IDs which were then used to obtain needed information such as Gene Ontology (GO) terms, as well as SIFT, polyphenn, and CADD scores. This was done through Ensembl’s Variant Effect Predictor (VEP) (McLaren et al., 2016)whereby the user can input RS values and select certain filters to acquire downloadable tabulated information on the input values. The data was then downloaded as a TSV file. The different populations were selected so the genetic variant between the populations could be explored and compared with each other.

Different populations were selected is because it would be a better representation of T1D worldwide. This is because having populations selected within a certain region in the world to represent T1D does not mean that the information about the T1D would be the same in a population raised in a different region. Additionally, it would also provide better access to users who are interested in retrieving genomic, functional, and clinical information from certain populations in different regions of the world. As a result, researchers can carry out several studies of how T1D would develop based on the different genetic factors affected by the environment the population were raised in.

## 2.3 SQLite Database Creation and Management

An empty database was created using the sqlite 3 package in python, which was subsequently imported into SQLite studio v3.4.3, a standalone application used to edit SQLite databases, using a graphical user interface (GUI).

Once all the relevant datasets had been obtained, they were then imported into the database as separate tables through SQLite studio’s built in management functions.  The tables were then renamed to *‘GWAS\_CHR\_6’* for the GWAS table, and *‘CHR\_6\_Disease Predictors’* for the dataset downloaded from the VEP. The datasets were then assigned primary and foreign keys through SQLite studio; the primary key being the *‘SNPs’* column from the GWAS table, and the foreign key being the *‘SNPs’* column from the disease predictors table. Following the designation of the keys, the superfluous columns were removed. The database was then saved and ready to use for the integration layer.

## 2.4 DATABASE SCHEMA

**Diagram

Description automatically generated with low confidence**

*Figure 2: The Database schema shows the structure of the relational database in SQLite that was used in the construction of the SMART web application. Created with BioRender.com* (Aoki, 2017)

# 3. Website Design and Features

## 3.1: Website user guide

When the user first visits the website, they will see a brief description of the website. Immediately below that, they are prompted to select the type of genomic information they would like to query by clicking on the toggle bar. After the toggle bar is clicked they have four options to choose from which can be seen in figure 4. Below the toggle bar the user is able to enter the relevant data point they would like to query related to the genomic option they choose.

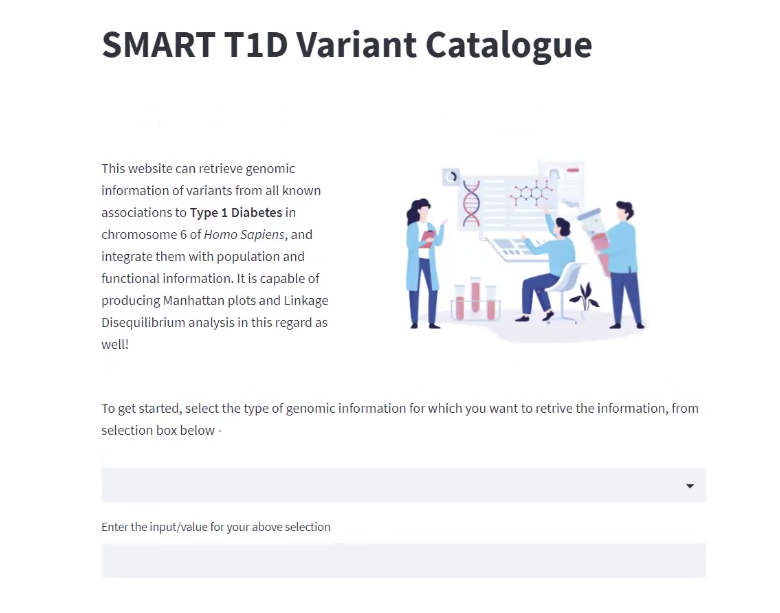


Figure 3: Screengrab of the SMART T1D Variant Catalogue homepage

Graphical user interface, application, Teams

Description automatically generated

Figure 4: Screenshot of dropdown bar that has four genomic search options, rs values, gene name, genomic location/co-ordinate and region. Once an option has been selected the user can input the related data.

## 3.2 Data presentation and visualisation

To improve the user experience, when data is returned from the server it is presented in an interactive table which is easy to navigate. Furthermore, the retrieved data can be downloaded in a CSV file format.

We then created a framework for collecting user input and streamlined the process by enabling direct input from the table when the user selects rows. The integration layer was utilized to fetch data from the database and display it in the interactive table on the website.

**Graphical user interface, table

Description automatically generated**

Figure 5: Example of interactive table of produced from TRIM31 gene name query.

## 3.2 Website features:

To create the SMART T1D Variant Catalogue web application a python-based framework called Streamlit was used, which is specifically tailored for use in the data science and machine learning fields. The initial aspects of the website that was constructed was the website title, heading and description, as well as, incorporating a Graphic Interchange Format (GIF) to enhance the user experience. After, additional features to help the user submit their queries were added such as search bars and side bars. As an added feature current date and time were integrated into the side bar, and a chatbot feature similar to Chat-GPT was incorporated.

Finally, the "Contact Us" page was designed with a functional form that is linked to our email, enabling prompt response to user queries. Social media badges with links to relevant accounts were also added to this page, and a chatbot feature was included, enabling users to communicate with us in real-time.

Multiple pages were added to the site, including an "About Us" page which incorporates the website name, a description of the group members, and interactive pictures of each member. The "Documentation" page included a hyperlink to this document, and a GIF was added to this page.

# 4 Integration Layer:

To connect the SQLite database and the website interface sqlite 3 and standard python programming was used. For each of the four genomic search options a function was created. The all four functions follow the basic structure for that of the rs\_value. To view the code please see file name.py, lines in the Team\_Safari Git Hub repository:

## 4.1 Functions to connect user input to database:

1. ***show\_rsvalue (rsvalue****)* – connects to the SQL database and returns the region, chromosome position, mapped genes (if applicable), p-values, allele frequency in five populations, CADD Phred score and gene ontology. A data frame is then created to store the data, then Steamlit commands are used to present the data frame in an interactive table on the home web page.
2. ***gene\_name(gene)*** – works in a similar way to *rsvalue,* however the user can enter one or more characters and all the rows of SNPs that have mapped genes will be returned. This is made possible with this SQL statement: WHERE MAPPED\_GENE LIKE ? """, ('%' + gene + '%',)). In this function if multiple rows are produced then a Manhattan plot is produced.
3. **show\_region(region1, region2=None**) – allows the user to enter one region and reterive data for that specific query. Or the user can enter two regions and all the SNPs between those regions – inclusive- will be returned and a Manhattan plot produced.
4. **chrpos(chrpos1, chrpos2=None)**– allows the user to enter one SNP location and reterive data for that specific query. Or the user can enter co-ordinates and all the SNPs between those co-ordinates – inclusive- will be returned with a Manhattan plot produced. The user can also select SNPs from the table and LD will be calculated for the population of there choice. This is then presented in a heatmap.

## 4.2 Statistics

### 4.2.1 Linkage Disequilibrium:

Linkage disequilibrium (LD) is used to explore non-random association of alleles at different loci within a population and can be calculated using either r^2 or D’ statistical methods (Slatkin, 2008). The r2  calculation returned data in a range from 0 to 1 where as D’ presents data from -1 to 1. If the user enters genomic co-ordinates, then Linkage Disequilibrium (LD) will be calculated for both r^2 and D’. However, the heatmap produced presents values calculated for r2. One of the reasons why r2 is used in the heat map it because the r2 calculation is less effected by allele frequency when compared to D’.

To calculate LD three functions were created:

1. **map\_population\_to\_ensembl\_populations (population)** - stores population data
2. **calculate\_ld\_for\_pair (rs\_values:, population) -**  accepts two SNPs and the population for which it should calculate their LD. The function returns the D` and r^2 values, if they can be calculated and “NaN” otherwise. Under the hood we are using ensembl’s API for LD calculation. In order to increase the likelihood for a successful calculation for each population(European, Asian and African) we go through all available population sets in ensembl that correspond to our requested population. To ensure the best user experience and to not waste resources, once we successfully calculate the LD values we stop iterating through the datasets and return the result.
3. **calculate\_pairwise\_ld\_for\_list (rs\_values, population)** - generates all possible combinations of SNPs for a given list, for the specified population(s), then gathering the results from calculate\_pairwise\_ld and recording the input parameters and the results in a Pandas data frame (rs1, rs2, population, D`, r2). These results can then be easily used to create a plot (figure 6 ) or saved as a tsv.
4. The request library was used for the http request to ensemble. A *get request* was performed to obtain the necessary LD values as specified by their API documentation(Yates et al., 2015). The response was decoded as json and the necessary information extracted.

Chart, bar chart

Description automatically generated

Figure 6: Example heatmap created.

## 4.2.2 Manhattan plot:

Instances where the user query returned multiple SNPs, a Manhattan plot was utilised to represent the p-value of each SNP, with -log10pvalue being used for the y-axis and the ordered SNP position for the x-axis. The purpose of the Manhattan plot is to represent many genetic variants in a single plot, so that the user could easily spot the regions of the chromosome that pass a significance threshold. Consistent with best practices, the -log10pvalue was preferred over the raw p-value as it enhances the visualisation of outliers. Furthermore, to enable the distinction of different population-based SNP categories, each population was assigned a different colour.

The function created to produce the Manhattan Plots used the matplotlib and pandas libraries. The Manhattan plot component consists of the below three functions:

1. **map\_populations\_for\_manhattan\_plot**(populations): converts a population string from our database to a corresponding simple one word string. This is necessary because the format of the population information in our database cannot be nicely displayed in a plot. (e.g. “3,561 European ancestry cases, 4,646 European ancestry controls”).
2. **create\_manhattan\_plot(df):** creates and displays a manhattan plot for a given data frame. The data frame should contain pvalue, position and population columns. The data frame is plotted using the -log10pvalue for the y-axis and the position for the x-axis. The data points are also colored according to population(ancestry).
3. **fetch\_data\_and\_create\_manhattan\_plot(genomic\_coordinates)**: fetches the necessary snp data for given genomic coordinates from the database. Then creates and displays a manhattan plot with the returned snp data.

Chart, scatter chart

Description automatically generated

Figure 7: Example of Manhattan plot produced. Blue represents African population, pink Asian and Green represents European.

# 5. Limitations

The data based on population from the GWAS Catalog is has more allele frequency data for European population compared with the smaller samples and studies on African and Asian populations. This is also a similar issue when calculating LD, for example data from the Asian population where not all from continental Asian countries but from people that migrated to specific cities in European countries. Another limitation of this web application is that the database only contains SNPs data for chromosome 6 in Homo Sapiens.

To further improve the user, experience the heatmap and Manhattan plot can include options for greater interactivity, such as generating a heatmap to show d prime values and changing colours of specific datapoints to highlight regions of interest.

For future development, the heatmap and the Manhattan plot can be made more interactive. The LD heatmap only shows r^2 values, an option can be included to generate a heatmap to also show d prime values. The Manhattan plot can also include options to change to change colors of specific datapoints to highlight a region of SNPs of interest and can be edited to show SNPS found in other chromosomes.

To present data on LD an Ensembl API was used. A limitation of this is that the Ensembl database does not have data for all the pairs of rs values in SMART database. Thus for some pairs, calculation of LD will not be possible. In that case the user will see a NaN entry in the resulting table. In order to improve the success rate, a different source of data has to be found. For example, in the International Genome Sample Resource website there were several VCF files that contain relevant data. However, the vcf file approach took a very long time to calculate LD.

# 6. Opportunities for development:

There are many opportunities for developments that can be included in future iterations of this web application. Firstly, the database can be expanded to cover SNP data for all Homo Sapiens as will as for other genetic pathologies. Once that has been updated the database can then be extended to other animal models.

One of the many additional features to the web application was the chatbot. Currently this is a standard chatbot but can be trained on information from the documentation and the website, so that it can answer more website and data specific questions.

To allow users to keep a record of the data they have search for in addition to being able to download the tables and plots, future updates should include a log-in feature that saves their search history and plots produced from their queries.

# 7. References

Aoki, S. (2017). *BIORENDER*. Biorender.

McLaren, W., Gil, L., Hunt, S. E., Riat, H. S., Ritchie, G. R. S., Thormann, A., Flicek, P., & Cunningham, F. (2016). The Ensembl Variant Effect Predictor. *Genome Biology*, *17*(1). https://doi.org/10.1186/s13059-016-0974-4

Morales, J., Welter, D., Bowler, E. H., Cerezo, M., Harris, L. W., McMahon, A. C., Hall, P., Junkins, H. A., Milano, A., Hastings, E., Malangone, C., Buniello, A., Burdett, T., Flicek, P., Parkinson, H., Cunningham, F., Hindorff, L. A., & MacArthur, J. A. L. (2018). A standardized framework for representation of ancestry data in genomics studies, with application to the NHGRI-EBI GWAS Catalog. In *Genome Biology* (Vol. 19, Issue 1). https://doi.org/10.1186/s13059-018-1396-2

Slatkin, M. (2008). Linkage disequilibrium - Understanding the evolutionary past and mapping the medical future. In *Nature Reviews Genetics* (Vol. 9, Issue 6). https://doi.org/10.1038/nrg2361

Yates, A., Beal, K., Keenan, S., McLaren, W., Pignatelli, M., Ritchie, G. R. S., Ruffier, M., Taylor, K., Vullo, A., & Flicek, P. (2015). The Ensembl REST API: Ensembl Data for Any Language. *Bioinformatics*, *31*(1). https://doi.org/10.1093/bioinformatics/btu613