CM3010 Databases and advanced data techniques

Midterm coursework report

Table of Content

Introduction

This report outlines the processes and methods applied in the coursework for the "Databases and Advanced Data Techniques (CM3010)" module. It begins with the rationale behind the dataset selection and data modeling, progressing through the stages of designing, developing a relational database and an associated web application. Each section of the report follows these steps in order, detailing the approach taken in design, implementation, and the integration of the database with the web application.

Dataset

Motivation and objectives

SNPedia has been selected as the primary data source for this project. SNPedia is a community-driven crowdsourcing initiative that collects and curates information about human genetics, with a particular focus on Single Nucleotide Polymorphisms (SNPs). SNPs are specific positions in DNA that vary among individuals and are known to correlate with various traits or diseases (phenotypes). For instance, rs1805007 with a (T;T) variant is linked to red hair, while rs4988235 with a (C;C) variant is associated with lactose intolerance.

The motivation to utilize SNPedia is rooted in an interest in bioinformatics and personal genomics, both of which are reshaping the future of healthcare and personalized medicine. Understanding the correlation between specific DNA positions and associated traits or diseases is invaluable. Such insights are crucial for predicting disease risk, determining drug response, unraveling the genetic basis of traits, and exploring population evolutionary history.

The aim of this project is to extract data from SNPedia and compile it into a dataset suitable for computational processes, using CSV files as medium. This dataset will then be used to establish a relational database, organizing the information in a structured, easily queryable form. A web application will also be developed, enabling specific, efficient queries and data presentation.

The idea of this project is inspired by personal genomics services, such as 23andMe, which provide information about a person's genetic variants and their associated traits or diseases. And especially by the Promethease, which uses SNPedia data to generate a report. However, the Promethease is proprietary and paid.

Dataset assessment

Data quality

Since SNPedia is a community-driven crowdsourcing project, the quality of the data is not guaranteed. The information provided by SNPedia is not peer-reviewed by experts, and there is no formal process for ensuring the accuracy of the data. As for any crowdsourcing project, the quality of the data is dependent on the number of contributors and their expertise. However, SNPedia provides a list of literature references for many SNPs, which can be used to verify the accuracy of the data.

Overall, the quality of the data is acceptable for the aims of this project. The dataset is not intended to be used for medical purposes, and the disclaimer information will be provided in the web application.

Details

SNPedia provides detailed information on over 100,000 SNPs. Each entry in SNPedia typically includes the location of the SNP in the DNA, associated traits or diseases, and the strength of these associations (magnitude). The dataset targeted for this project includes two main types of information: SNP data and genotype data. The SNP information covers aspects such as the SNP's name, its DNA location, related traits or diseases, and references to scientific papers. Each SNP has a set of associated PMIDs, and unique identifiers of PubMed records. SNPs are also categorized into various types, like 'Interesting', '23andMe SNP', or 'Y chromosome SNPs'. Genotype information encompasses the SNP's name, possible variations at that position, the magnitude of their effects, reputation (positive or negative), and the related traits or diseases. This dataset, with its multidimensional nature and complex relationships between different elements, is well-suited for modeling in a relational database.

Most of the interesting SNPs in the dataset have a detailed description of their effects, along with a list of literature references. However, some SNPs have only a brief description, while others have no description at all. The completeness of the information is not guaranteed, as is its quality. However, the dataset is still useful for the aims of this project, as it contains a large number of SNPs with detailed and useful information suitable for educational purposes.

Documentation

SNPedia is based on MediaWiki and uses the same markup language as Wikipedia. Therefore, all tools and documentation available for MediaWiki are applicable to SNPedia. SNPedia also provides a Bulk API for extracting data from SNPedia. This API is based on MediaWiki's API, which is well-documented and has a large community of users. Therefore, the documentation for SNPedia is sufficient for the aims of this project.

Interrelation

The format used in SNPedia, which is based on MediaWiki for ease of reading and editing, poses challenges for computer-based querying and processing. For instance, it's challenging to extract combined data about a specific SNP and its genotype variants, such as the (C;C) variant of rs1805007, because this information is spread across different pages.

Data licensing

SNPedia data is distributed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. This license permits the use and redistribution of the data, aligning well with the objectives of this project. Furthermore, SNPedia explicitly allows web scraping and provides access to a Bulk API through MediaWiki's API, specifically for this purpose. Therefore, extracting data from SNPedia for this project is both legally compliant and ethically sound.

Discoverability

The field of bioinformatics is rapidly evolving, and there are many open, freely accessible databases of genetic information. Most of them are computer-readable and have a well-defined schema. The most known databases are maintained by the National Center for Biotechnology Information (NCBI), including dbSNP, ClinVar.

However, SNPedia is unique in that it is a community-driven project that is not formally structured. This makes it challenging to extract data from SNPedia and organize it into a structured form. However, the dataset from SNPedia is still useful for the aims of this project, because it contains a curated extraction of particularly useful and interesting information.

In summary, the dataset from SNPedia offers real, openly accessible data that is complex enough for the aims of this project. It also provides a unique opportunity to explore the challenges of extracting data from a non-standardized source and organizing it into a structured, queryable form.

Research questions

The dataset from SNPedia is full of useful information that can help answer many research questions. This project focuses on the following key questions:

Genetic test interpretation: Customers of personal genomics services, such as 23andMe, are able to download their raw genetic data in a text file. This file contains a list of SNPs and their genotypes. Can the dataset be used to interpret this data and provide a report about the traits or diseases linked to these SNPs?

Most Important SNPs: Which SNPs have the greatest magnitude, indicating a strong impact on traits or diseases? This question seeks to identify SNPs that play a major role in human health and characteristics.

SNPs Related to Specific Traits or Diseases: Are there specific SNPs linked to certain traits or diseases? A keyword search function could be useful for finding these connections in the dataset.

Data scraping

The data scraping, processing, and cleaning code is implemented in the iPython notebook "build_dataset.ipynb". The contents of the notebook are provided in the Appendix. This process adheres to the guidelines specified on the Bulk API page of SNPedia. It consists of three primary steps: extracting raw data, transforming it into a structured format, and then saving this structured data as CSV files.

Initially, raw data is gathered from SNPedia using the MediaWiki API using the *requests* HTTP client library. This data is temporarily stored in a Pandas DataFrame for subsequent processing. The *mwparserfromhell* package is then used to parse the raw data, extracting relevant information from the Wikitext markup. Upon completion of this parsing step, the refined data is saved into CSV files using the 'to_csv' method of Pandas.

More details about the data extraction process can be found in the iPython notebook "build_dataset.ipynb" the contents of which are provided in the Appendix.

The structure of the dataset

As a result of the data extraction stage we have a set of CSV files, each containing a specific type of information. The most important files are:

snps.csv contains a list of SNPs, with their *ID* and a *Description*. The *ID* could be a name, like 'rs1805007', or a number, like 'i3003137'. The first one is a standard SNP ID, while the second one is a custom ID used by different personal genomics services as an alias for the standard ID or to represent a SNP that is not yet standardized and named officially. The `Description` is a short text briefly describing the SNP. Many SNPs do not have a description, but it is still included in the dataset for completeness. This list is complete and includes all SNPs extracted from SNPedia.

genotypes.csv contains a list of genotypes (variants) for each SNP. Each genotype has two alleles, one from each parent: *allele1* and *allele2*. The *magnitude* column indicates the strength of the association between the genotype and the related traits or diseases. The *repute* column indicates whether the genotype is considered 'good', 'bad', or 'mixed'. The *summary* column provides a brief description of the genotype. The *description* column provides a more detailed description of the genotype. *snp* is the ID of the SNP that the genotype is associated with. Here each SNP can have multiple genotypes, but each genotype is associated with only one SNP. This list is complete and includes all genotypes extracted from SNPedia.

Other files provide additional information about SNPs:

pmids.csv contains a list of PMIDs (PubMed IDs) for each SNP. PubMed is a widely used database of biomedical literature. Each PMID is a unique identifier of a scientific paper in the PubMed database. The *snp* column contains the ID of the SNP that the PMID is associated with. The *PMID* column contains the PMID, while the *Title* column contains the title of the paper. Here each SNP can have multiple PMIDs, but each PMID is associated with only one SNP.
rsnums.csv provides information about biological characteristics of each SNP. It includes the location of a SNP in DNA: *Gene*, *Chromosome*, *position*, *Orientation*, and *StabilizedOrientation*. The reference genomes: *Assembly*, *GenomeBuild*, and *dbSNPBuild*. A reference variant along with other known variants at this position: *ReferenceAllele*, *MissenseAllele*, *geno1*, *geno2*, *geno3*, etc.

clinvars.csv contains information from ClinVar, a widely-used public database of reports of the relationships among human variations and phenotypes. Most of the columns with a `CLN` prefix are VCF fields from ClinVar. VCF is a standard format for storing genetic data. The specific meaning of these fields can be found in the ClinVar VCF specification. But in general, these fields provide more advanced information about the SNP, that is beyond the scope of questions explored in this project.

categories.csv contains information about the categories that each SNP belongs to. These categories are defined by SNPedia editors and are not standardized. For example, 'Interesting' is a category that includes SNPs that are interesting to the editors of SNPedia. A `On chip 23andMe v5` category lists SNPs that are provided by the 23andMe v5 chip.

Data modeling

Entity–relationship Model

To address the research questions outlined in the previous section, a subset of the dataset is required. This subset includes the following files:

- *snps.csv* — a list of SNPs, with their `ID` and a `Description`.
- *genotypes.csv* — a list of possible genotypes (variants) for each SNP. This is the most crucial file, as it catalogs information about the meaning of a specific genotype, its magnitude, and the related traits or diseases.
- *pmids.csv* — a list of PMIDs (PubMed IDs) for each SNP. Would be useful for finding more information about a SNP in scientific papers.
- *rsnums.csv* — Providing biological characteristics of SNPs. Information from this file is valuable for more complex queries, such as identifying SNPs on a particular chromosome.

The *clinvars.csv* file contains information from the ClinVar database. This advanced information is not required for the research questions outlined in this project. But the model will be designed in an extensible way that allows for the integration of this information in the future.

From these files, the following entities can be identified:

SNP

The SNP entity stems from the *snps.csv* file. This file contains a complete list of SNPs extracted from SNPedia. Therefore, all related entities will be dependent on this entity. The SNP entity will have the following attributes:

- A primary key *id* that is a unique identifier of the SNP, such as 'rs1805007' or 'i3003137'. The string identifier is used instead of a number, as it is guaranteed to be unique, human-readable, not subject to change, and is meaningful as a foreign key in other tables.
- Non-key attributes *Description* from *snps.csv*. This is a general description of the SNP.
- Additional attributes from *rsnums.csv*: *gene*, *chromosome*, and *position*.

All non-key attributes should be optional, as some SNPs do not have this information. This entity will have a one-to-many relationship with *Genotype* and a many-to-many relationship with *Category* and *Literature*.

Genotype

The Genotype entity stems from the *genotypes.csv* file. This file contains a list of possible genotypes (variants) for each SNP. This entity will have a many-to-one relationship with the *SNP*. The *Genotype* entity is dependent on the *SNP*, as it doesn't make sense without a parent SNP. The *Genotype* entity will have the following attributes:

- A primary key *id* of type INT.
- A foreign key *snp_id* referencing the SNP it belongs to.
- Non-key attributes *allele1*, *allele2*, *magnitude*, *repute*, *summary*, *description*.

The *allele1* attribute is mandatory, but *allele2* is optional, as genotypes from X and Y chromosomes in men have only one allele. The *magnitude*, *summary*, *description* attributes are optional, as some genotypes do not have them.

Category

This entity stems from the *categories.csv* file. This file contains a list of snp-category pairs. The unique categories from this file will be extracted and used as Category names. This entity will have the following attributes:

- A primary key *id* of type INT.
- A non-key attribute *name*.

The *name* is a mandatory attribute. The name will have a unique constraint, as each category should have a unique name. However, name is not a primary key, as it is subject to change, that leads to a change of the foreign keys in all related entities that potentially leads to performance penalty and update anomalies.

This entity will have a many-to-many relationship with *SNP* via a junction table.

Literature

This entity stems from the *pmids.csv* file. This file contains a list of snp-pmid pairs. The unique PMIDs from this file will be extracted to populate this entity. This entity will have the following attributes:

- A primary key *id* of type INT.
- Non-key attributes *PMID*, *title*.

This entity will have a many-to-many relationship with *SNP* via a junction table.

Entity–relationship Diagram

The entity-relationship diagram below illustrates the relationships between these entities:

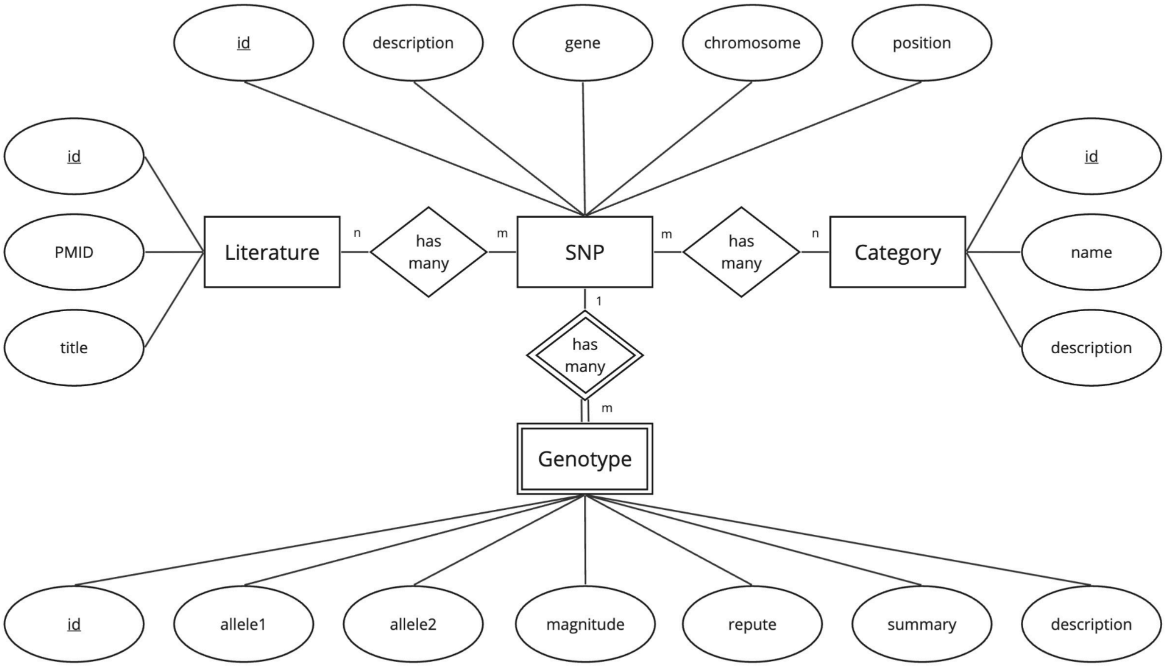


Figure 1: Entity–relationship diagram (ERD)

List of tables

The following tables will be created in the database:

SNP

Column	Type	Constraints	Description
id	VARCHAR(255)	PRIMARY KEY	Unique identifier of the SNP
description	TEXT		A text description of the SNP
gene	VARCHAR(255)		The gene that the SNP is located in
chromosome	VARCHAR(255)		The chromosome that the SNP is located on
position	INT		The position of the SNP on the chromosome

Genotype

Column	Type	Constraints	Description
id	INT	PRIMARY KEY	Unique identifier of the genotype
snp_id	VARCHAR(255)	FOREIGN KEY, NOT NULL	The ID of the SNP that the genotype belongs to
allele1	VARCHAR(255)	NOT NULL	The first allele of the genotype
allele2	VARCHAR(255)		The second allele of the genotype
magnitude	INT		The magnitude of the genotype
repute	ENUM('good', 'bad', 'mixed')		The reputation of the genotype
summary	TEXT		A brief description of the genotype
description	TEXT		A detailed description of the genotype

Category

Column	Type	Constraints	Description
id	INT	PRIMARY KEY	Unique identifier of the category
name	VARCHAR(255)	UNIQUE, NOT NULL	The name of the category

Literature

Column	Type	Constraints	Description
id	INT	PRIMARY KEY	Unique identifier of the literature
PMID	INT	UNIQUE, NOT NULL	The PubMed ID of the literature
title	VARCHAR(255)	NOT NULL	The title of the literature

SNP_Category

This is a junction table between *SNP* and *Category*. It does not have a separate primary key because the combination of *snp_id* and *category_id* is unique.

Column	Type	Constraints	Description
snp_id	VARCHAR(255)	FOREIGN KEY, NOT NULL	The ID of the SNP that the category belongs to
category_id	INT	FOREIGN KEY, NOT NULL	The ID of the category that the SNP belongs to

SNP_Literature

This is a junction table between *SNP* and *Literature*. It does not have a separate primary key because the combination of *snp_id* and *literature_id* is unique.

Column	Type	Constraints	Description
snp_id	VARCHAR(255)	FOREIGN KEY, NOT NULL	The ID of the SNP that the literature belongs to

literature_id	INT	FOREIGN KEY, NOT NULL	The ID of the literature that the SNP belongs to

Normal Form Analysis

Database normalization (Amato, 2023) is a process used in relational database design to organize data efficiently and reduce data redundancy while ensuring data integrity. It involves breaking down large tables into smaller,related tables and defining relationships between them. The main goals of database normalization are to eliminate data anomalies, reduce data duplication, and make the database more manageable.

Here is an assessment of the proposed database design against the normal forms:

1NF

Lewis (2016) states that a relation is in first normal form (1NF) if and only if all the domains in which its attributes are defined contain scalar values only. In the proposed model, all attributes in all relations are defined atomic values and do not use composite types, like arrays or JSON objects.

Therefore, the relations are in the first normal form.

2NF

Lewis (2016) states that a relation is in second normal form (2NF) if and only if:

1. it is in 1NF; and
2. every non-key attribute is irreducibly dependent on the primary key.

All relations in the proposed model satisfy these conditions:

- All relations are in 1NF.
- All non-key attributes are irreducibly dependent on the primary key. There are no non-key attributes that can be derived from a part of a primary key.

Therefore, the relations are in the second normal form.

3NF

Lewis (2016) states that a relation is in third normal form (3NF) if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key.

All relations in the proposed model satisfy these conditions:

- All tables are in 2NF.
- There are no transitive dependencies, as all non-key attributes depend exclusively on primary keys. For instance, the *magnitude* and *repute* attributes in the *Genotype* table are not dependent on each other and represent independent characteristics of the genotype. There are no non-key attributes that can be derived from other non-key attributes.

Therefore, the database design is in the third normal form.

BCNF

This is a more strict version of the third normal form and sometimes referred to as the "third and a half" normal form. Lewis (2016) states that a relation is in Boyce-Codd normal form (BCNF) if and only if every functional dependency has a candidate key as its determinant.

All relations in the proposed model satisfy these conditions as there are no attributes that are functionally dependent on a part of a candidate key.

4NF

Lewis (2016) states that a relation R is in fourth normal form (4NF) if and only if for any existing multi-valued dependency A ↠ B (where A and B are subsets of the attributes of R), A is a candidate key of R – so all the attributes of R are functionally dependent on A.

According to the Wikipedia article on multi-valued dependencies, a multivalued dependency exists when there are at least three attributes (like X,Y and Z) in a relation and for a value of X there is a well defined set of values of Y and a well defined set of values of Z. However, the set of values of Y is independent of set Z and vice versa.

In the *Genotype* relation of the proposed model, the *allele1* and *allele2* attributes are both dependent on the *id*, but independent of each other. However, the *id* is a candidate key of the *Genotype*. Therefore, the relation is in 4NF.

The *SNP_Category* and *SNP_Literature* resolve many-to-many relationships between *SNP* and *Category* and *Literature* respectively to avoid multi-valued dependencies. There are no other multivalued dependencies in the proposed model that would violate the 4NF.

Therefore, the database design is in the fourth normal form.

5NF

Lewis (2016) states that a relation is in fifth normal form (5NF) if and only if all its join dependencies are implied by its candidate keys.

In other words, to be in 5NF, every join dependency should have a superkey component. In the proposed model, there are joint dependencies between *SNP* and *Genotype*, *Category*, and *Literature*. All these join dependencies have a superkey component, as the primary key of *SNP* is a superkey. *Category* and *Literature* are dependent on *SNP* via a junction table, which has a composite primary key of *snp_id* and *category_id* or *literature_id*.

Therefore, the database design is in the fifth normal form.

Database implementation

Database Schema

The database is created using the following SQL statements:

```sql
CREATE DATABASE snpedia_db;
USE snpedia_db;

CREATE TABLE SNP (
    id VARCHAR(255) PRIMARY KEY,
    description TEXT,
    gene VARCHAR(255),
    chromosome VARCHAR(255),
    position INT,
    FULLTEXT KEY (id, description)
);

CREATE TABLE Genotype (
    id INT PRIMARY KEY AUTO_INCREMENT,
    snp_id VARCHAR(255) NOT NULL,
    allele1 VARCHAR(255) NOT NULL,
    allele2 VARCHAR(255),
    magnitude INT,
    repute ENUM('good', 'bad', 'mixed'),
    summary TEXT,
    description TEXT,
    FOREIGN KEY (snp_id) REFERENCES SNP(id)
);

CREATE TABLE Category (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) UNIQUE NOT NULL
);

CREATE TABLE Literature (
    id INT PRIMARY KEY AUTO_INCREMENT,
    PMID INT UNIQUE NOT NULL,
    title VARCHAR(255)
);

-- this is a junction table to link SNPs to categories as a
many-to-many relationship
CREATE TABLE SNP_Category (
    snp_id VARCHAR(255),
    category_id INT,
    PRIMARY KEY (snp_id, category_id),
    FOREIGN KEY (snp_id) REFERENCES SNP(id),
    FOREIGN KEY (category_id) REFERENCES Category(id)
);

-- this is a junction table to link SNPs to literature as a
many-to-many relationship
CREATE TABLE SNP_Literature (
    snp_id VARCHAR(255),
    literature_id INT,
    PRIMARY KEY (snp_id, literature_id),
    FOREIGN KEY (snp_id) REFERENCES SNP(id),
    FOREIGN KEY (literature_id) REFERENCES Literature(id)
);
```

Data Import

The data is imported into the database using the iPython notebook "data_import.ipynb". The contents of the notebook are provided in the Appendix with all details. To import the data, the Pandas `to_sql` method is used along with the SQLAlchemy library. This method generates a series of *INSERT* statements to import the data into the database. This approach is both simpler and safer than constructing and executing SQL statements manually.

Reflection

The dataset perfectly fits the relational database model. The dataset was designed to be fully compatible with the relational database model. However, it was not one-to-one mapping between the dataset and the database schema. For example, the `pmids.csv` file contains not normalized data, as the title of a paper is repeated for each PMID and each PMID is repeated for each SNP. The aforementioned import script normalized this data into two separate tables: *Literature* and *SNP_Literature*. It is a usual practice to have less normalized data in CSV files for simplicity, as CSV datasets are usually read-only and not subject to update anomalies.

The `clinvars.csv` file contains advanced information about SNPs. It was not included in the database schema, as this information is not required for the research questions outlined in this project. However, the database schema was designed in an extensible way that allows for the integration of this information in the future.

Queries

Query information about genotypes

The following query created to retrieve information about a set of genotypes, ordered by their magnitude (importance) and limited to 100 results:

```
SELECT SNP.id, SNP.description, Genotype.magnitude, Genotype.repute,
       Genotype.summary, Genotype.description, Genotype.allele1,
       Genotype.allele2
    FROM SNP
    JOIN Genotype ON SNP.id = Genotype.snp_id
    WHERE (SNP.id = 'rs333' AND Genotype.allele1 = '-' AND
              Genotype.allele2 = '-') OR
          (SNP.id = 'rs76361015' AND Genotype.allele1 = 'G' AND
              Genotype.allele2 = 'G') OR
          (SNP.id = 'rs6588505' AND Genotype.allele1 = 'C' AND
              Genotype.allele2 = 'T') OR
          (SNP.id = 'rs351855' AND Genotype.allele1 = 'C' AND
              Genotype.allele2 = 'C')
          -- here we can add more SNPs and genotypes to query
    ORDER BY Genotype.magnitude DESC
    LIMIT 100;
```

Explore SNPs

The following query created to explore all SNPs in the database, ordered by their magnitude (importance) and paginated:

```sql
SELECT SNP.id, SNP.description, Genotype.magnitude, Genotype.repute,
       Genotype.summary, Genotype.description, Genotype.allele1,
       Genotype.allele2
    FROM SNP
    JOIN Genotype ON SNP.id = Genotype.snp_id
    ORDER BY Genotype.magnitude DESC
    LIMIT ? OFFSET ?; -- Used for pagination
```

The *Genotype.magnitude* used to sort the results in descending order, so the most important SNPs are at the top of the list. Also, the *LIMIT* and *OFFSET* clauses are used for pagination, so the results can be displayed in pages. The COUNT aggregate function used to count the total number of SNPs in the database to calculate the total number of pages:

```sql
SELECT COUNT(*) FROM SNP JOIN Genotype ON SNP.id = Genotype.snp_id;
```

For SNP search functionality, the full text search feature of MySQL is used. The *MATCH* and *AGAINST* clauses used to search for a keyword in the *FULLTEXT* index of SNP:

```sql
SELECT SNP.id, SNP.description, Genotype.magnitude, Genotype.repute,
       Genotype.summary, Genotype.description, Genotype.allele1,
       Genotype.allele2
    FROM SNP
    JOIN Genotype ON SNP.id = Genotype.snp_id
    WHERE MATCH (SNP.id, SNP.description)
          AGAINST (? IN NATURAL LANGUAGE MODE)
    ORDER BY Genotype.magnitude DESC
    LIMIT ? OFFSET ?;
```

Query optimization

Initially, the query was not optimized and took minutes to return results for 10000 genotypes. The query plan generated by "EXPLAIN" showed that the query was not using any indexes on the Genotype table.

To improve the performance of the query, the following indexes were added:

```
CREATE UNIQUE INDEX idx_Genotype_snp_id_alel1_alel2 ON
Genotype(snp_id, allele1, allele2);
CREATE INDEX idx_genotype_magnitude ON Genotype(magnitude DESC);
```

The *idx_Genotype_snp_id_alel1_alel2* index aims to improve the performance of the *WHERE* clause, as it is used to filter the results. The *idx_genotype_magnitude* index aims to improve the performance of the *ORDER BY* clause, as it is used to sort the results.

After adding these indexes, the query plan generated by *EXPLAIN* showed that the query was using the *idx_Genotype_snp_id_alel1_alel2* index to filter the results and the *idx_genotype_magnitude* index to sort the results.

After adding these indexes, the query execution time was reduced from minutes to seconds.

Security

A separate user `snpedia_user` was created to access the database. This user has limited privileges and can only perform *SELECT* queries. This user is used by the web application to access the database. The principle of least privilege is followed, as the user has only the minimum privileges required to perform its function. The password for this user is stored in the `.env` file and is not committed to the repository:

```
CREATE USER 'snpedia_user' IDENTIFIED BY 'password';
GRANT SELECT ON snpedia_db.* TO 'snpedia_user';
```

To prevent SQL injection attacks, the web application uses prepared statements. Prepared statements are a feature of the MySQL client library that allows for the safe execution of SQL queries by separating the query string from the query parameters. The parameters are checked for validity automatically, before the query is executed.

Web application

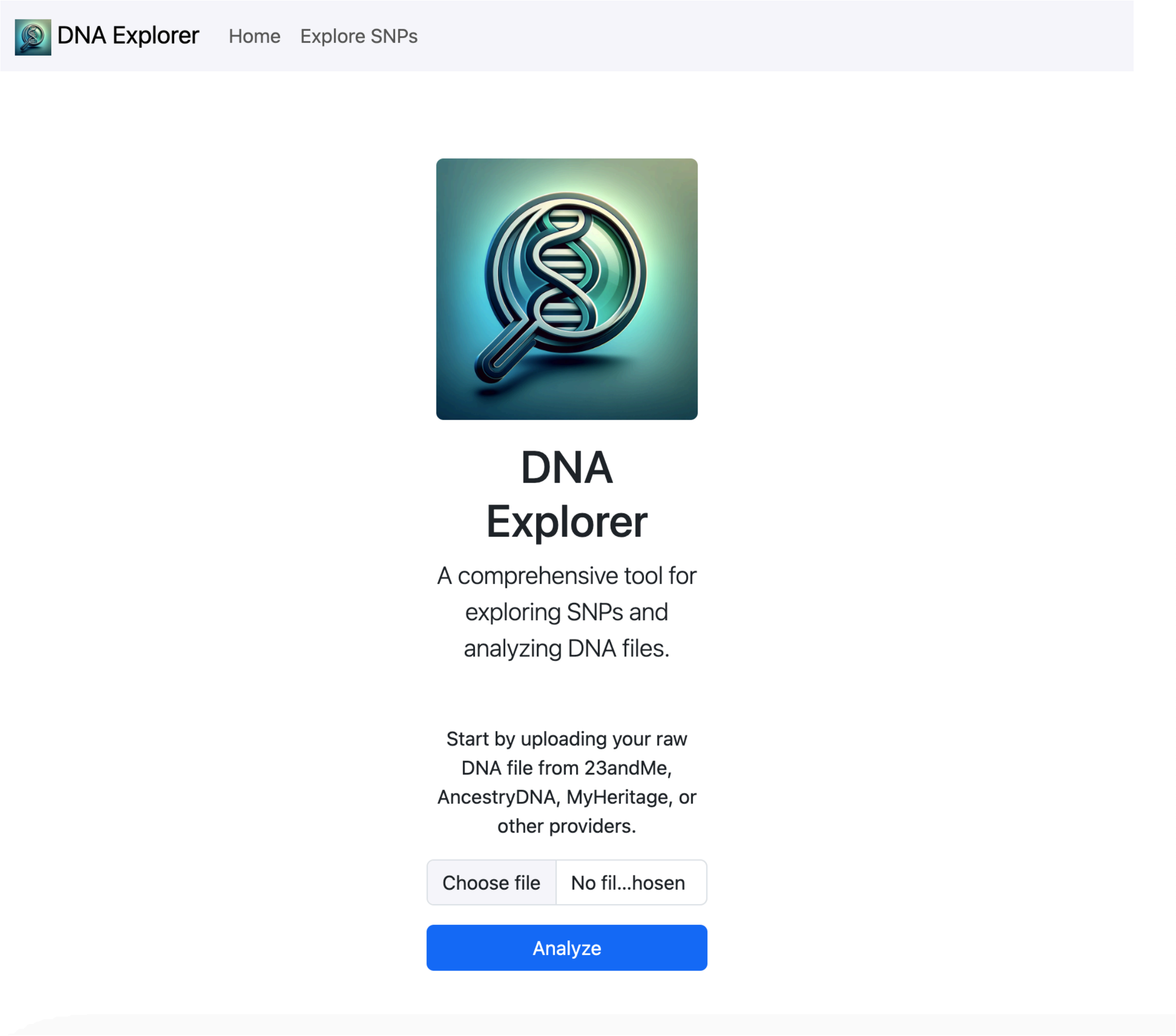
Application code is located in the `app` directory. It is implemented using the Express.js web application framework for Node.js. The application uses the ejs template engine to render HTML pages and *mysql2* library to connect to the MySQL database. The app implements two scenarios of use according to the research questions outlined in the previous sections:

1. Upload a raw genetic data file from a personal genomics service, such as 23andMe, and generate a report that lists the most important SNPs in the file and their interpretation. This scenario is implemented on the home page of the application. In files `app/routes/home.js` and `app/routes/analyze.js`. The `analyze.js` file contains the logic for processing the uploaded file and generating the report. The `app/views/results.ejs` file contains the HTML template for the report.

2. Explore all SNPs in the database and view their details. This scenario is implemented on the "Explore SNPs" page of the application. In files `app/routes/explore.js` and `app/views/explore.ejs`. The `explore.js` file contains the logic for querying the database, paginating the results, and full-text search. The `app/views/explore.ejs` file contains the HTML template for the list of SNPs.

Evaluation

The web application was evaluated using the following scenario:
1. Run the web application using `npm run dev`.
2. Open the web application in a browser at http://localhost:3000/.



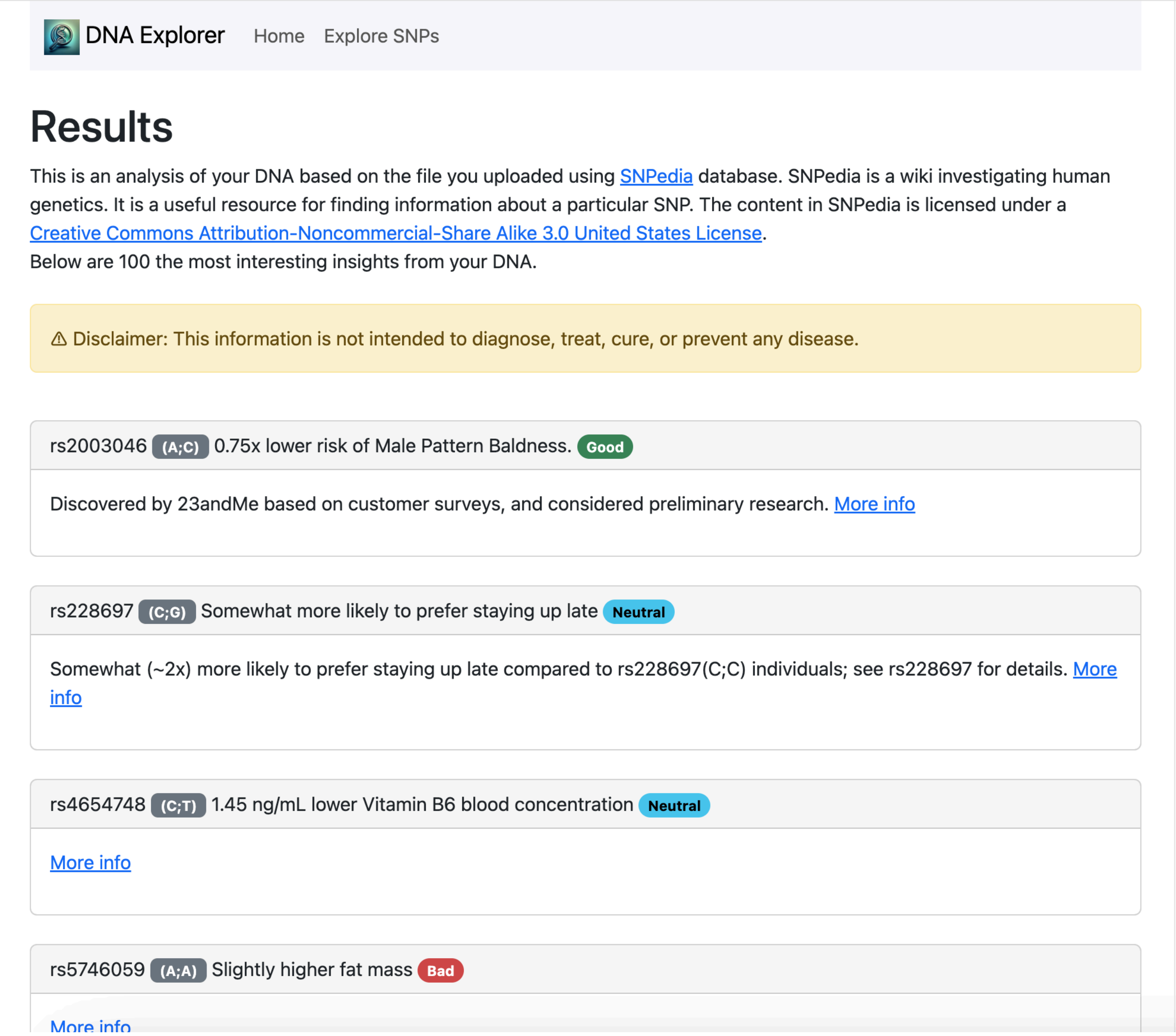
DNA Explorer

A comprehensive tool for exploring SNPs and analyzing DNA files.

Start by uploading your raw DNA file from 23andMe, AncestryDNA, MyHeritage, or other providers.

Figure 2: The home page with a form to upload a DNA test file.

3. On the home page (see Figure 2), click the "Choose file" and select the `test.txt` file from the `app/sample` directory. This is a sample 23andMe
raw genetic data file with a list of SNPs and their genotypes, truncated to 10000 SNPs to be processed faster.
4. Click the "Analyze" button.
5. Wait for the analysis to complete, 20-30 seconds on Apple M1.
6. A sample report should be displayed. The report contains a list of SNPs ordered by their magnitude, along with their interpretation, magnitude, reputation, and other information, see Figure 3:



Results

This is an analysis of your DNA based on the file you uploaded using SNPedia database. SNPedia is a wiki investigating human genetics. It is a useful resource for finding information about a particular SNP. The content in SNPedia is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.
Below are 100 the most interesting insights from your DNA.

⚠ Disclaimer: This information is not intended to diagnose, treat, cure, or prevent any disease.

rs2003046 (A;C) 0.75x lower risk of Male Pattern Baldness. Good

Discovered by 23andMe based on customer surveys, and considered preliminary research. More info

rs228697 (C;G) Somewhat more likely to prefer staying up late Neutral

Somewhat (~2x) more likely to prefer staying up late compared to rs228697(C;C) individuals; see rs228697 for details. More info

rs4654748 (C;T) 1.45 ng/mL lower Vitamin B6 blood concentration Neutral

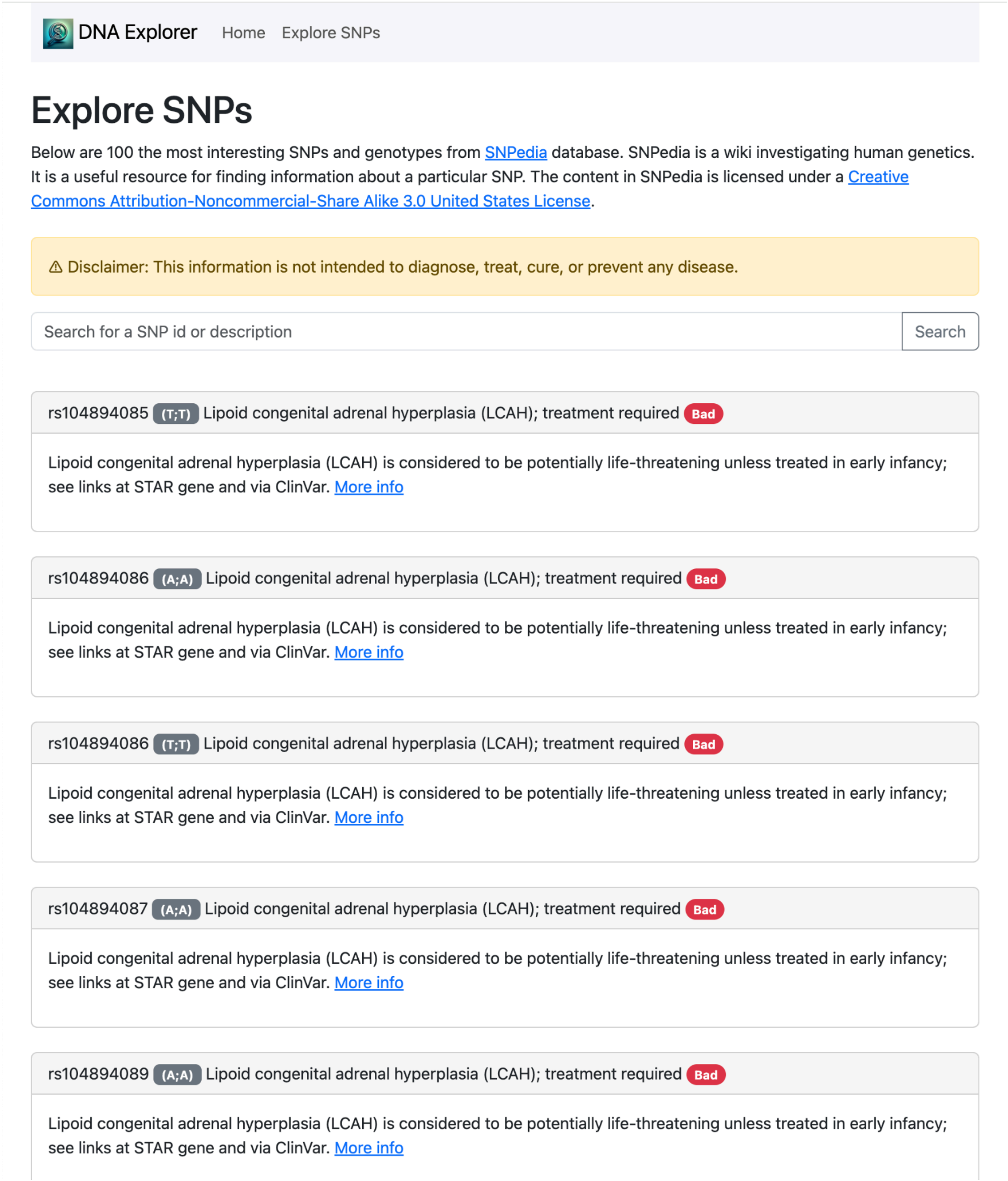
More info

rs5746059 (A;A) Slightly higher fat mass Bad

More info

Figure 3: A sample report

7. Click "Explore SNPs" link to view the list of all SNPs in the database, ordered by their magnitude, see Figure 4:



Explore SNPs

Below are 100 the most interesting SNPs and genotypes from SNPedia database. SNPedia is a wiki investigating human genetics. It is a useful resource for finding information about a particular SNP. The content in SNPedia is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.

⚠ Disclaimer: This information is not intended to diagnose, treat, cure, or prevent any disease.

rs104894085 (T;T) Lipoid congenital adrenal hyperplasia (LCAH); treatment required Bad

Lipoid congenital adrenal hyperplasia (LCAH) is considered to be potentially life-threatening unless treated in early infancy; see links at STAR gene and via ClinVar. More info

rs104894086 (A;A) Lipoid congenital adrenal hyperplasia (LCAH); treatment required Bad

Lipoid congenital adrenal hyperplasia (LCAH) is considered to be potentially life-threatening unless treated in early infancy; see links at STAR gene and via ClinVar. More info

rs104894086 (T;T) Lipoid congenital adrenal hyperplasia (LCAH); treatment required Bad

Lipoid congenital adrenal hyperplasia (LCAH) is considered to be potentially life-threatening unless treated in early infancy; see links at STAR gene and via ClinVar. More info

rs104894087 (A;A) Lipoid congenital adrenal hyperplasia (LCAH); treatment required Bad

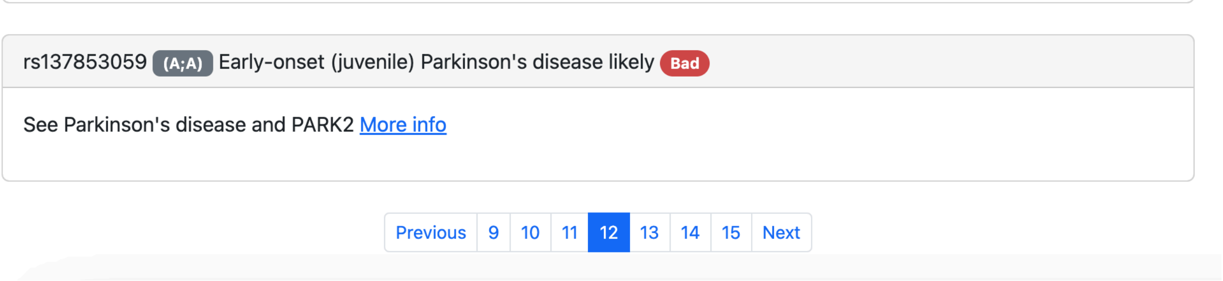
Lipoid congenital adrenal hyperplasia (LCAH) is considered to be potentially life-threatening unless treated in early infancy; see links at STAR gene and via ClinVar. More info

rs104894089 (A;A) Lipoid congenital adrenal hyperplasia (LCAH); treatment required Bad

Lipoid congenital adrenal hyperplasia (LCAH) is considered to be potentially life-threatening unless treated in early infancy; see links at STAR gene and via ClinVar. More info

Figure 4: A SNP Explorer with the most important SNPs.

8. Pagination at the bottom of the page allows navigating through the list of SNPs:

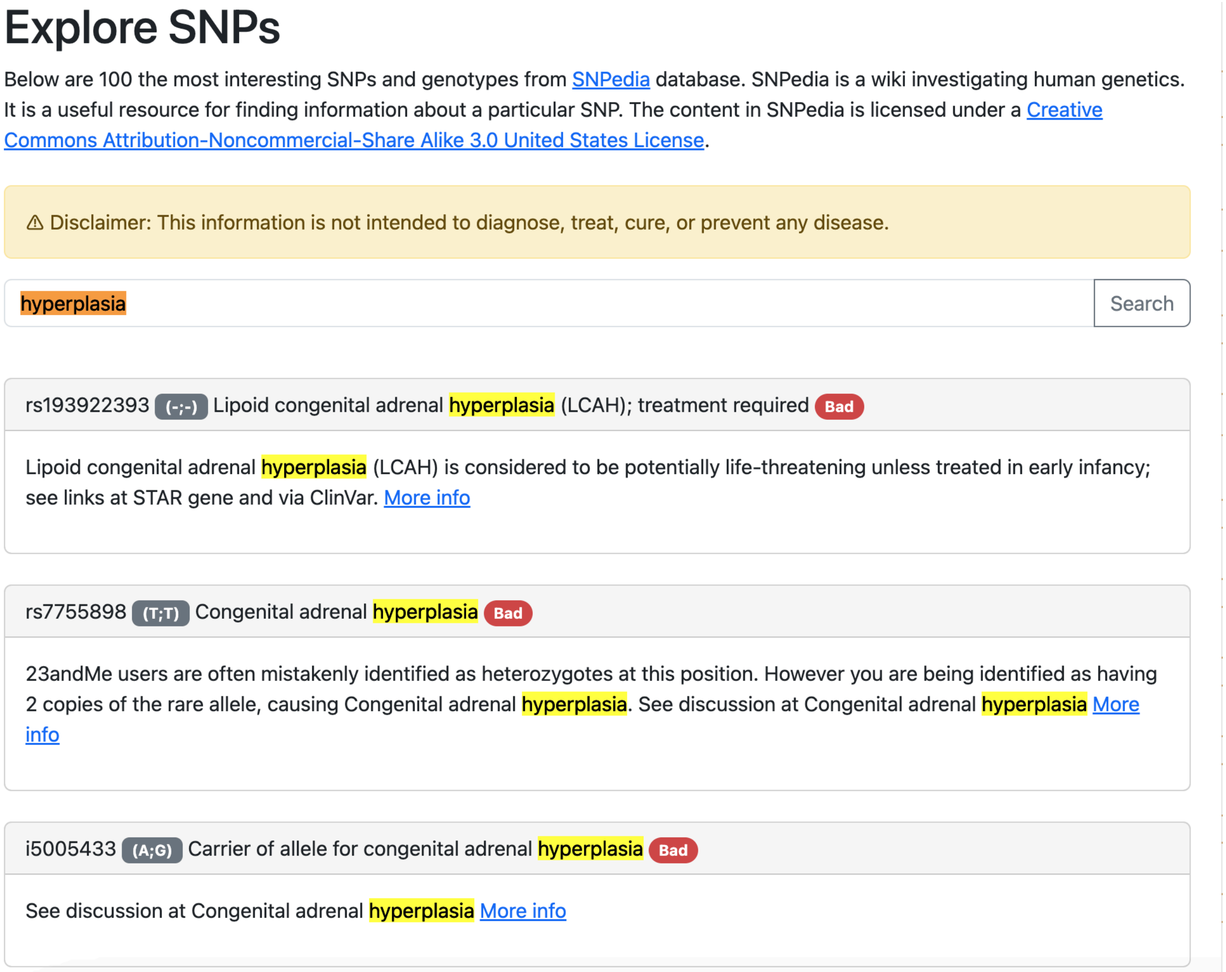


rs137853059 (A;A) Early-onset (juvenile) Parkinson's disease likely Bad

See Parkinson's disease and PARK2 More info

Figure 5: Pagination allows navigating through the list of SNPs.

9. Search for a specific term, such as "eye color" in the search box at the top of the page should return a list of SNPs related to this term:



Explore SNPs

Below are 100 the most interesting SNPs and genotypes from SNPedia database. SNPedia is a wiki investigating human genetics. It is a useful resource for finding information about a particular SNP. The content in SNPedia is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.

⚠ Disclaimer: This information is not intended to diagnose, treat, cure, or prevent any disease.

hyperplasia

rs193922393 (-;-) Lipoid congenital adrenal hyperplasia (LCAH); treatment required Bad

Lipoid congenital adrenal hyperplasia (LCAH) is considered to be potentially life-threatening unless treated in early infancy; see links at STAR gene and via ClinVar. More info

rs7755898 (T;T) Congenital adrenal hyperplasia Bad

23andMe users are often mistakenly identified as heterozygotes at this position. However you are being identified as having 2 copies of the rare allele, causing Congenital adrenal hyperplasia. See discussion at Congenital adrenal hyperplasia More info

i5005433 (A;G) Carrier of allele for congenital adrenal hyperplasia Bad

See discussion at Congenital adrenal hyperplasia More info

Reflection

A fully functional web application was developed with a useful set of features. The application allows users to upload a 23andMe raw data file and generate a report that lists the most important SNPs in the file and their interpretation. The application also allows users to explore all SNPs in the database and view their details. Some of the research questions outlined in the previous sections can be answered using this application.

The user interface of the application is sketchy as it was not the primary focus of this project. The application could be improved by adding a more user-friendly interface, such as a wizard that guides the user through the process of uploading a file and generating a report.

References

External code

All external code used in this project is labeled with the comments "START OF EXTERNAL CODE" and "END OF EXTERNAL CODE". It comes with an inline reference to the source of the code.

Used libraries

- requests — HTTP client library
- pandas — data analysis and manipulation tool
- mwparserfromhell — parser for MediaWiki wikicode
- sqlalchemy — SQL toolkit and Object Relational Mapper
- express.js — web application framework for Node.js
- ejs — template engine for Node.js
- dotenv — load configuration from a .env file.

Assets

- *app/public/images/logo.png* generated by a *OpenAI DALL-E 2* AI model.

Literature

Amato, N. (2023) _Mastering database normalization: A comprehensive exploration of normal forms_ [Online] Available from: https://www.researchgate.net/publication/374509386_Mastering_database_normalization_A_comprehensive_exploration_of_normal_forms [17 December 2023].

Lewis, D. (2016). _CO2209 Database systems._ London: University of London.

Wikipedia contributors. (2023) _Multivalued dependency_ [Online] Wikipedia, The Free Encyclopedia. Available from: https://en.wikipedia.org/wiki/Multivalued_dependency [17 December 2023].

The pandas development team. (2023) _Pandas 2.1.4 documentation_ [Online] NumFOCUS, Inc. Available from: https://pandas.pydata.org/docs/ [18 December 2023].

Oracle Corporation. (2023) _MySQL 8.0 Reference Manual_ [Online] Available from: https://dev.mysql.com/doc/refman/8.0/en/ [18 December 2023].

Appendix