P3 Report: BioJoin Creative

Team 5

20180061 Daewon Kim 20200283 Haejoon Park 20210351 Gyuchan An

Contents

A. Introduction

1. Reflection on Genetic Distance

Genetic Distance is a measure of the genetic divergence between species or between populations within a species. Populations with many similar alleles have small genetic distances; which indicates that they are closely related and have a recent common ancestor.

Calculating genetic distance is useful in a variety of biomedical applications, especially for the problems which are in need to reconstruct the history of populations. Many genetic research projects rely on calculating the genetic distance between populations, and a variety of formulae have been developed for this purpose.

In this project, we focus on developing a tool that can aid academic research by calculating region-based genetic distance using SNP data.

2. Reflection on Visualization

Visualization is a technique that uses images or diagrams to convey information. The technique is especially useful in delivering information and data in an abstract way.

One branch of data visualization is statistical graphics, which is a way of presenting the result of data analysis by displaying it in pictorial form. The most renowned statistical graphic is plots, including scatter plots, probability plots etc. Statistical graphics benefit greatly in relationship identification and outlier detection.

In this project, we will exploit statistical graphics to represent our research results.

B. Methods

1. Cavalli-Sforza Genetic Distance

Cavalli-Sforza genetic distance is one of the most commonly used measures to calculate genetic distance between two populations. In the formula, ***X*** and ***Y*** represent two different populations for which ***L*** loci have been studied. $\boldsymbol{X_u}$ represents the ***u***th allele frequency at the ***l***th locus.

$$\boldsymbol{D_{CH}}' = \sqrt{1 - \sum_{l} \sum_{u} \sqrt{X_u Y_u}}$$

According to Ondrej Libiger, Caroline Nievergelt and Nicholas Schork, "*Comparison of Genetic Distance Measures Using Human SNP Genotype Data"* (BioOne, 2009), p. 389., "Cavalli-Sforza and Edwards distance is relatively more sensitive in distinguishing genetically similar populations". We decided to use this measure in our program as suggested in the research paper.

2. Visualization of SNP sequences

We intend to visualize SNP in the sense of gene sequences. The visualization will be done by displaying SNP sequences within a range of 21 amino acids, centering the ancestor or the minor allele in the 11th index. Two SNP sequences will be shown in a parallel form, upper sequence containing ancestor allele, and downstream sequence containing minor allele.

C. Database Design

1. ERD

Conceptual database design is illustrated in the following entity-relationship diagram. SNP, gene, and disease have already existed in P2.

SNP_genotype_550 is a database containing the genotype data of 550 individuals from different counties of the United States. The database is used to calculate the allele frequencies of each county.

human_genome_reference_sequence contains the whole genomic sequence of each chromosome, which is used to extract the sequence of each SNP using the chromo, pos column of SNP db as an index.

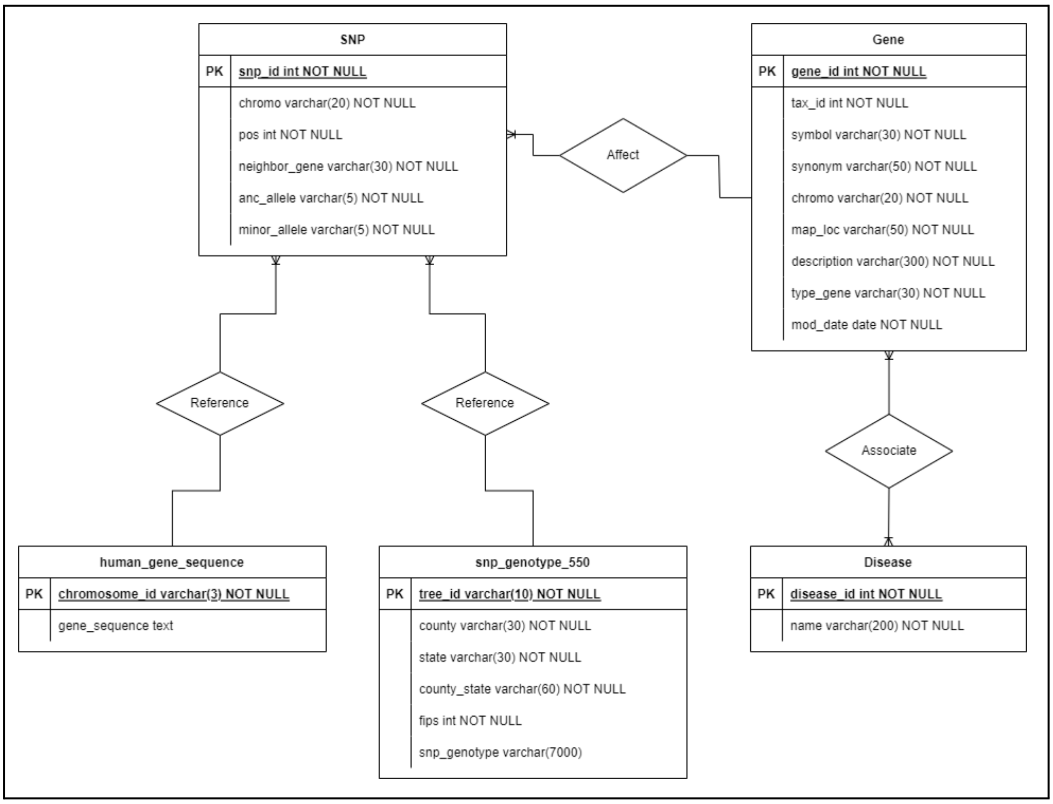

SNP
PK
snp_id int NOT NULL
chromo varchar(20) NOT NULL
pos int NOT NULL
neighbor_gene varchar(30) NOT NULL
anc_allele varchar(5) NOT NULL
minor_allele varchar(5) NOT NULL
Affect
Gene
PK
gene_id int NOT NULL
tax_id int NOT NULL
symbol varchar(30) NOT NULL
synonym varchar(50) NOT NULL
chromo varchar(20) NOT NULL
map_loc varchar(50) NOT NULL
description varchar(300) NOT NULL
type_gene varchar(30) NOT NULL
mod_date date NOT NULL
Reference
Reference
Associate
human_gene_sequence
PK
chromosome_id varchar(3) NOT NULL
gene_sequence text
snp_genotype_550
PK
tree_id varchar(10) NOT NULL
county varchar(30) NOT NULL
state varchar(30) NOT NULL
county_state varchar(60) NOT NULL
fips int NOT NULL
snp_genotype varchar(7000)
Disease
PK
disease_id int NOT NULL
name varchar(200) NOT NULL

Figure 1. Entity-Relationship Diagram (ERD) of the project.

2. Relational schema

SNP	(**snp_id,** chromo, pos, neighbor_gene, anc_allele, minor_allele)
Gene	(tax_id, **gene_id**, symbol, synonym, chromo, map_loc, description, type, mod_date)
Disease	(**disease_id**, name)
SNP_genotype_550	(**tree_id**, county, state, county_state, fips, SNP genotype)
human_genome_reference_sequence	(**chromosome_id**, gene_sequence)

Table 1. Logical database design (relational schema) of the project. Bold, underlined keywords are primary keys.

D. SQL Implementation

1. SQL DDL

Database	SQL Query
snp	CREATE TABLE snp (snp_id integer primary key, chromo varchar(20), pos integer, neighbor_gene varchar(30), anc_allele varchar(5), minor_allele varchar(5));
gene	CREATE TABLE gene (tax_id integer, gene_id integer primary key, symbol varchar(30), synonym varchar(50), chromo varchar(20), map_loc varchar(50), description varchar(300), type_gene varchar(30), mod_date date);

disease	CREATE TABLE disease (
	disease_id integer primary key,
	name varchar(200)
	);

snp_genotype_550	CREATE TABLE snp_genotype_550(tree_id varchar(10) primary key, county varchar(30) NOT NULL, state varchar(30) NOT NULL, county_state varchar(60) NOT NULL, fips int NOT NULL, snp_genotype varchar(7000));
human_genome_reference_sequence	CREATE TABLE human_genome_reference_sequence (chromosome_id varchar(3) primary key, genomic_sequence text NOT NULL)

Table 2. SQL queries to define each table. Description for each column, which could be inferred by its name, is omitted.

2. SQL DML

Function	SQL Query
Record insertion	INSERT INTO *table_name* VALUES ([*val1*], [*val2*], …);

Record update	UPDATE *table_name* SET *column_name* = [*new_value*] WHERE *primary_key* = [*primary_key_condition*]
Record deletion	DELETE FROM *table_name* WHERE *primary_key* = [*primary_key_condition*];
Record search† (within a table)	SELECT * FROM *table_name* WHERE *column* = [*value*];
Record search† (substring)	SELECT * FROM *table_name* WHERE *column* LIKE %[*value*]%;

Record search (diseases by SNP ID)	SELECT *snp_id*, *disease_id*, *name*
	FROM *snp_gene* LEFT OUTER JOIN *gene_disease* USING (*gene_id*)
	RIGHT OUTER JOIN *disease* USING (*disease_id*)
	RIGHT OUTER JOIN *snp* USING (*snp_id*)
	WHERE *snp_id* = [SNP ID];

Record search (SNPs by disease)	SELECT *disease_id*, *name*, *snp_id*
	FROM *snp_gene* RIGHT OUTER JOIN *gene_disease* USING (*gene_id*)
	RIGHT OUTER JOIN *disease* USING (*disease_id*)
	WHERE *name* = *disease_name*;

Substring SNP sequence	SELECT SUBSTRING(*genomic_sequence*, *pos*-11, 21)
	FROM *human_genome_reference_sequence*
	LEFT OUTER JOIN *snp* ON *chromosome_id* = *chromo*
	WHERE *snp_id* = [SNP_ID];

Table 3. BioJoin's function and corresponding SQL query statements. Square brackets imply user input.
† Multiple search conditions could be combined by AND operators.

E. Results and Conclusions

1. Results

Below is an example figure representing the genetic distances between Columbus county and the other counties. The more red a county is represented, the closer its genetic distance to Columbus is. In our program, such figures are visible for each county when clicking the 'Show figure' button. In addition, a SNP genotype data of an individual, provided as a text file "gen_dist_input.txt", is used to calculate and make the figure of genetic distance between counties and the individual.

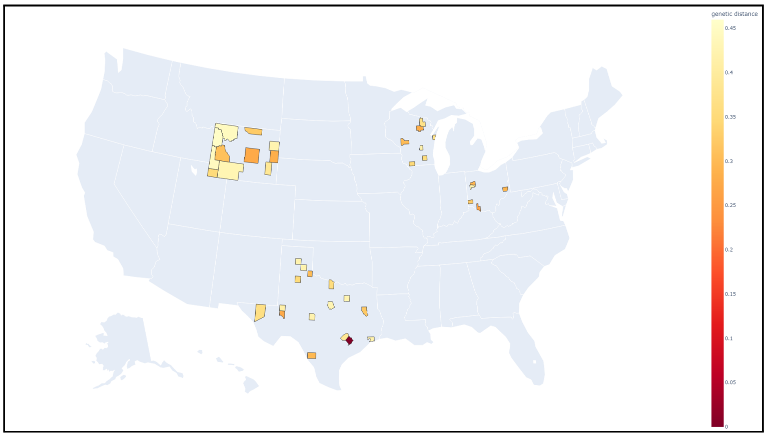


Figure 2. Distribution of Genetic Distance of Columbus County

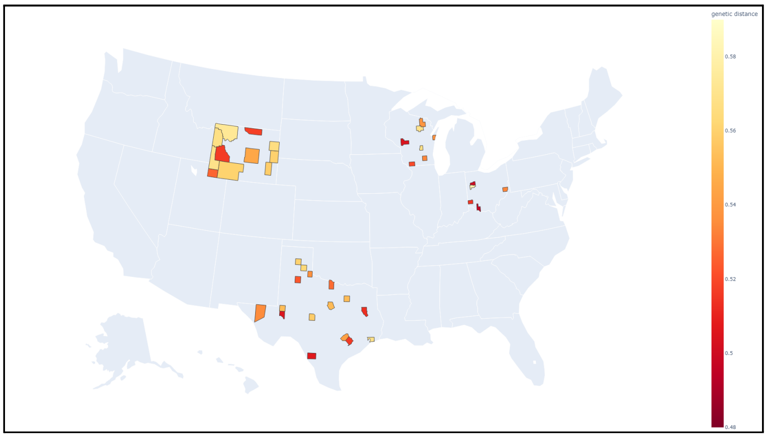
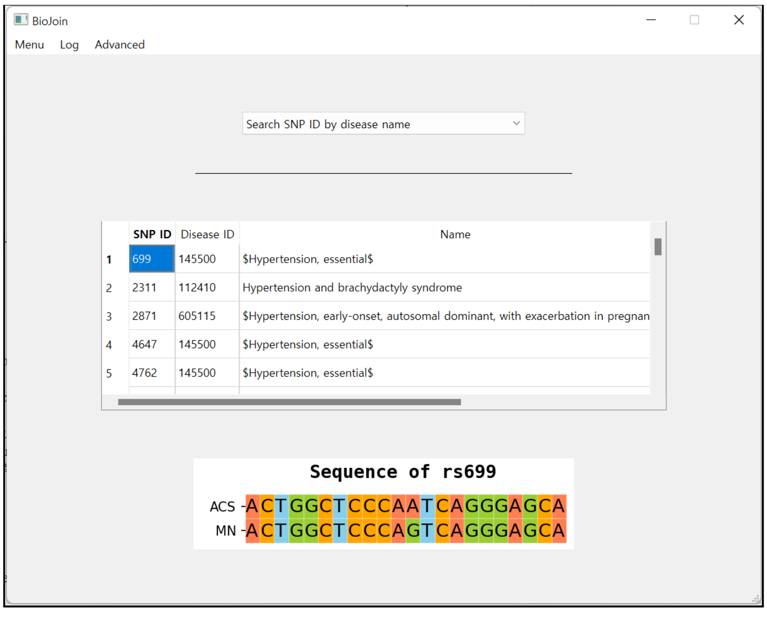


Figure 3. Distribution of Genetic Distance Relative to a Sample Individual

The following figure is an SNP visualization in the searching page. When the SNP ID cell is double-clicked, the DNA sequence around the single nucleotide polymorphism appears as a colorized figure at the bottom of the window.



	SNP ID	Disease ID	Name
1	699	145500	$Hypertension, essential$
2	2311	112410	Hypertension and brachydactyly syndrome
3	2871	605115	$Hypertension, early-onset, autosomal dominant, with exacerbation in pregnan
4	4647	145500	$Hypertension, essential$
5	4762	145500	$Hypertension, essential$

Sequence of rs699

```
ACS -ACTGGCTCCCAATCAGGGAGCA
 MN -ACTGGCTCCCAGTCAGGGAGCA
```

Figure 4. Distribution of Genetic Distance Relative to a Sample Individual

2. Conclusions

The resulting data of genetic distances between counties showed a mean of μ = 0.3928 and a standard deviation of σ = 0.07094. According to this data, we carried out a clustering with the k-means algorithm. To find the optimal number of clusters, we made use of the elbow method, whose figure is shown below. Obviously the optimal number is determined to be 3, and the resulting clustered figure is shown below as well. Counties colored the same are clustered together.

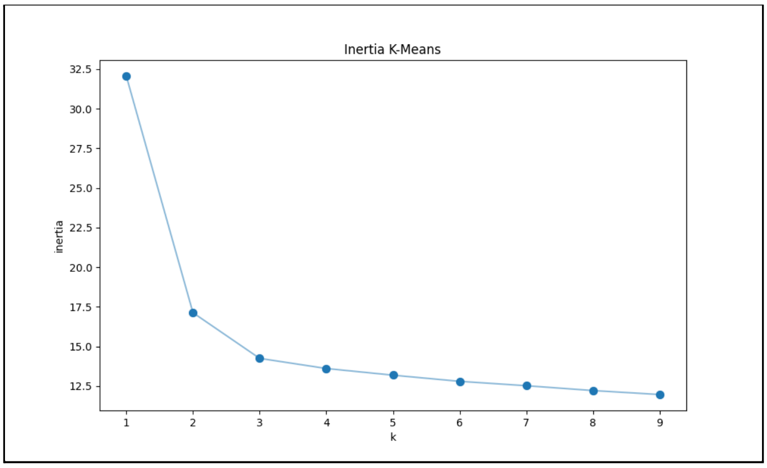


Figure 5. The Elbow Method used to find the Optimal Number of Clusters

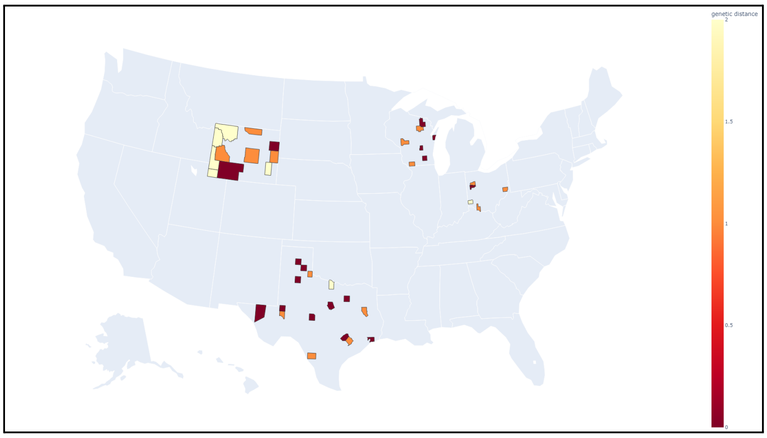


Figure 6. The Resulting Figure for Clustering - three clusters are visible.

Unfortunately, we could not find a high correlation between the genetic distances and physical distances between counties. This problem seems to be due to the heterogeneous characteristics of the United States, whose ethnicity has been often represented as “the melting pot”. We expect that if the database is expanded to that of individuals throughout the world, this algorithm will draw a more meaningful conclusion.

F. Appendix

1. Code Hierarchy

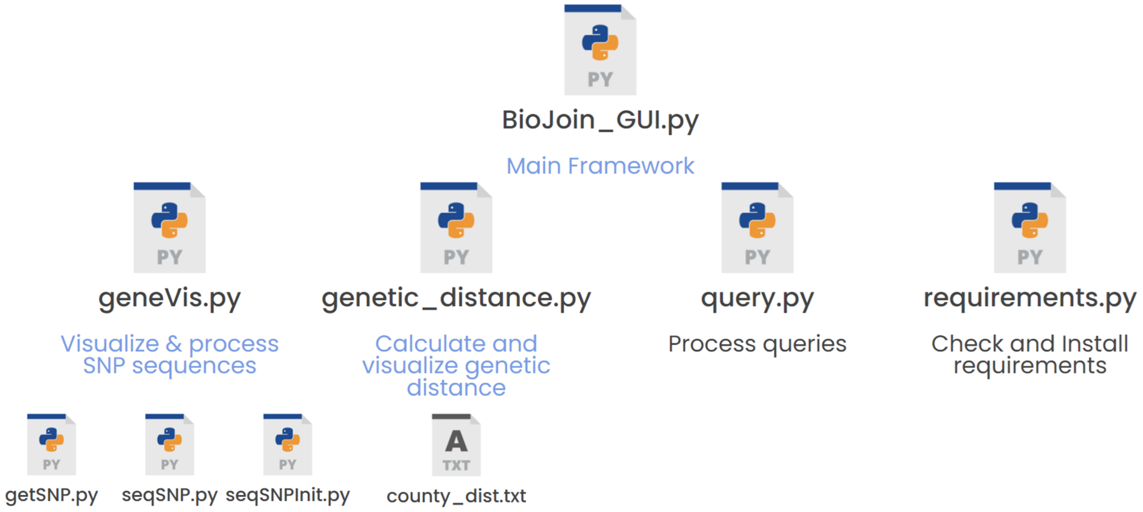
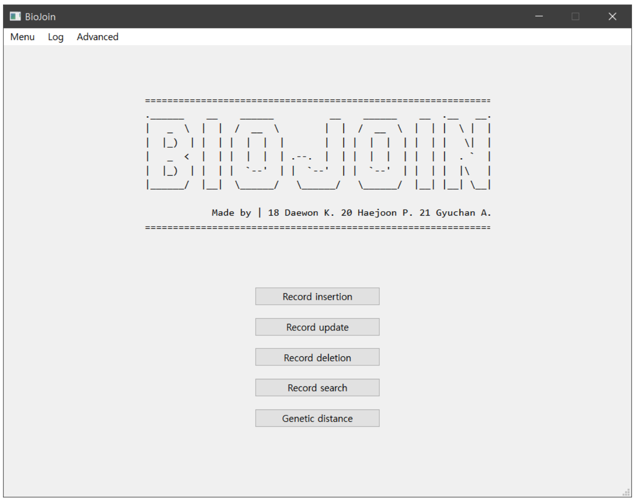


Figure 5. Code hierarchy of BioJoin program.

2. GUI

Our project applied PySide6 to build GUI. PySide6 is the official Python bindings for the Qt, which is a cross-platform application development framework. BioJoin_GUI.py imports all the required functions and classes from PySide6.

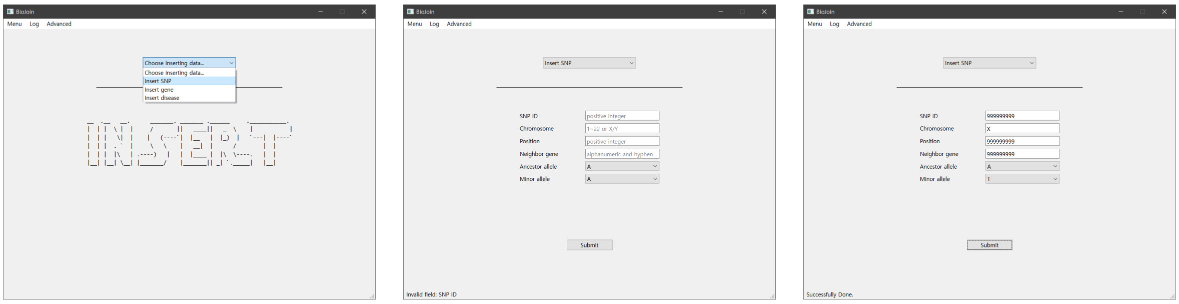


```
Made by | 18 Daewon K. 20 Haejoon P. 21 Gyuchan A.
```

Figure 6. Main menu of BioJoin GUI. There are five buttons to access corresponding functions.

Each operation internally transforms user input into appropriate SQL in table 3.

Insertion consists of three steps: selecting data to insert, filling out a form, and submitting. Record update, deletion, and search also starts by selecting data type. Each step is validated properly. Users cannot select data other than snp, gene, and disease. A line in a form prohibits invalid input according to its label. For example, a character ‘a’ cannot be typed in a field ‘SNP ID’. Pushing the submit button uploads given data to the database, which results in failure if one of the fields is invalid or the key is duplicated. Figure 7 shows steps of inserting SNP.



SNP ID: positive integer
Chromosome: 1~22 or X/Y
Position: positive integer
Neighbor gene: alphanumeric and hyphen
Ancestor allele: A
Minor allele: A

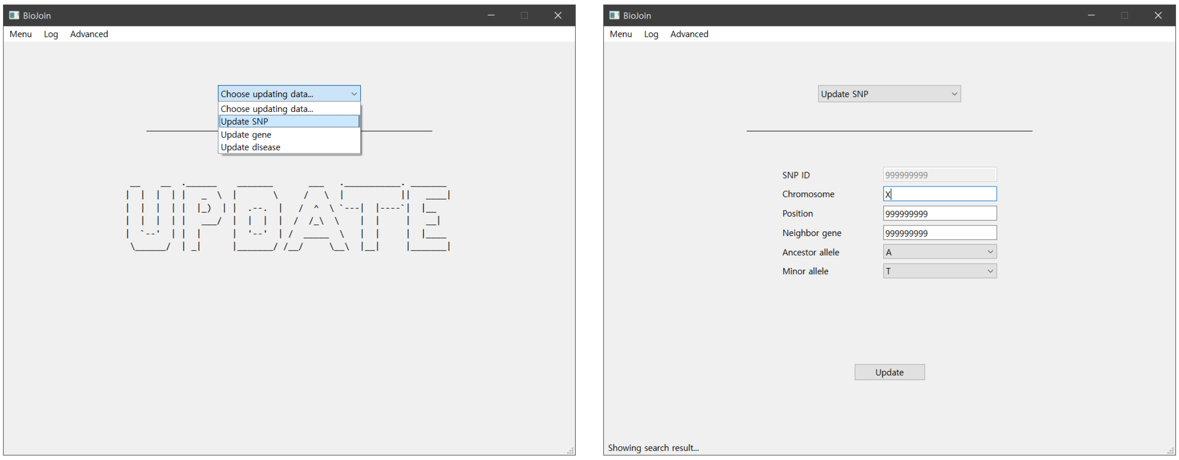
Invalid field: SNP ID

SNP ID: 999999999
Chromosome: X
Position: 999999999
Neighbor gene: 999999999
Ancestor allele: A
Minor allele: T

Successfully Done.

Figure 7. Record insertion screen. (7-1, left) User selects inserting data at first. (7-2, center) There is a form to fill out. Fields are determined by relational schema. (7-3, right) Pushing submit button leads to uploading filled data, which fails if invalid.

Record update is similar to insert, but there are two additional prior steps. User searches a record before he/she fills the form by primary key. BioJoin GUI then searches the record and automatically fills out the form with the current data. Users could modify the data and push the submit button like record insertion. Primary key field is exceptional; it cannot be modified (since the user selected the record to update by designating it).



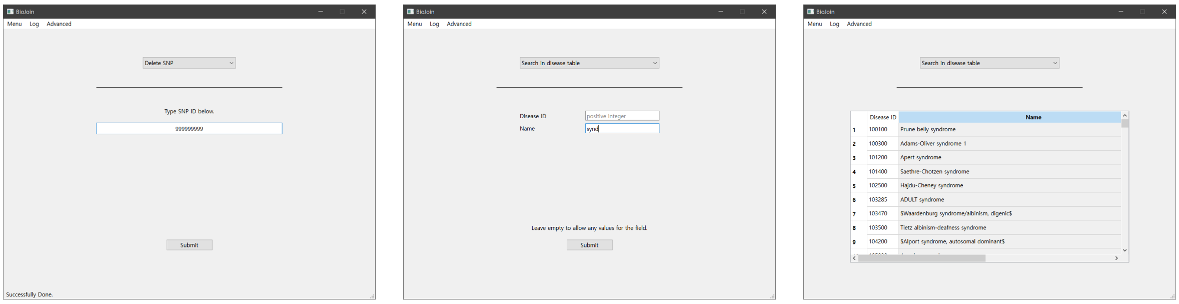
SNP ID: 999999999
Chromosome: X
Position: 999999999
Neighbor gene: 999999999
Ancestor allele: A
Minor allele: T

Showing search result...

Figure 8. Record update screen. (8-1, left) User selects updating data at first. (8-2, right) After the user inputs the primary key (SNP ID here), input form appears with original data. Modifying the fields and pushing the submit button would update the database.

Record deletion is simple; it simply accepts the primary key and deletes the connected record. Figure 9-1 shows a screen for record deletion.

Record search is similar to record insertion, but it allows empty input for some fields. If a field is empty, the field is considered as allowing any data. Also, it allows substring search – BioJoin will bring all records with the field containing the given string. For example, if the searching disease name is ‘synd’, the search result would include ‘adult syndrome’ (figure 9-2, 9-3). Results are tabulated as figure 9-3.



Delete SNP

Type SNP ID below.

999999999

Submit

Successfully Done.

Search in disease table

Disease ID: positive integer

Name: synd

Leave empty to allow any values for the field.

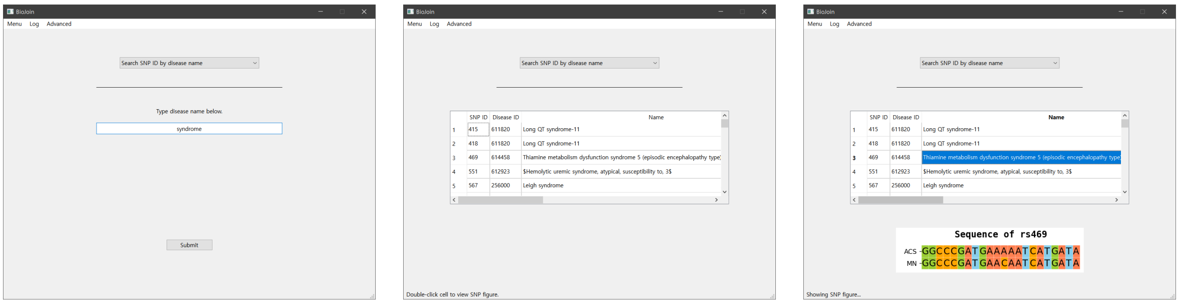
Submit

Search in disease table

	Disease ID	Name
1	100100	Prune belly syndrome
2	100300	Adams-Oliver syndrome 1
3	101200	Apert syndrome
4	101400	Saethre-Chotzen syndrome
5	102500	Hajdu-Cheney syndrome
6	103285	ADULT syndrome
7	103470	$Waardenburg syndrome/albinism, digenic$
8	103500	Tietz albinism-deafness syndrome
9	104200	$Alport syndrome, autosomal dominant$

Figure 9. Record deletion and search screen. (9-1, left) User selects deleting data. If successful, it shows the message ‘successfully done’. (9-2, center) Searching screen for disease table. User selects searching data at first. (9-3, right) Result is shown in a table. Note that it contains records whose disease names are not exactly the same as input in figure 9-2.

Searching data associated with more than one entity (which applies SQL JOIN keyword) will show additional information, SNP visualization, in its result screen. Double clicking a cell would display its visualization of SNP in the same row. (Figure 10) BioJoin supports searching SNP ID by disease name and searching diseases by SNP ID.



Search SNP ID by disease name

Type disease name below.

syndrome

Submit

Search SNP ID by disease name

	SNP ID	Disease ID	Name
1	415	611820	Long QT syndrome-11
2	418	611820	Long QT syndrome-11
3	469	614458	Thiamine metabolism dysfunction syndrome 5 (episodic encephalopathy type)
4	551	612923	$Hemolytic uremic syndrome, atypical, susceptibility to, 3$
5	567	256000	Leigh syndrome

Double-click cell to view SNP figure.

Showing SNP figure...

Figure 10. Record search screen. (10-1, left) In the figure, the user is trying to SNPs by disease name. (10-2, center) Results are shown in a table. (10-3, right) If a cell is double-clicked, its SNP would be visualized below the table.

Genetic distance menu could be accessed by clicking the last button in the main menu. There are two options: genetic distance between two counties; genetic distance between an individual and a county. The former one priorly selects a reference county. Then the ‘show figure’ button is activated. Clicking the button links the user to a web browser, showing the genetic distance in the US map by gradation of color. Genetic distance between an individual and a county is similar except that the user inputs an individual's SNP genotype data rather than a reference county.

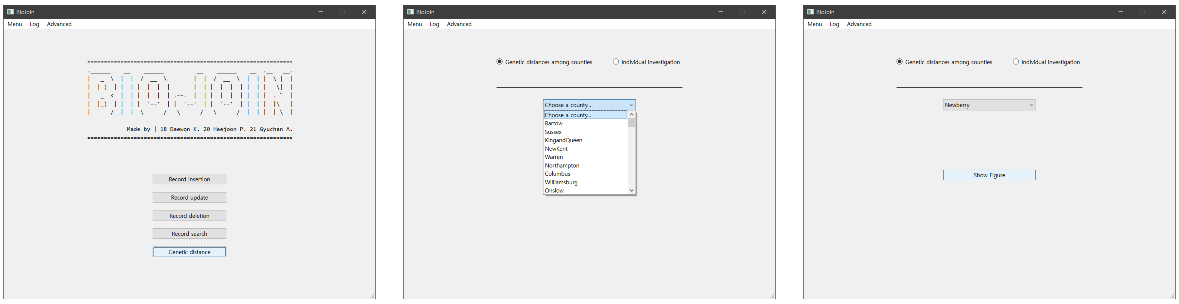
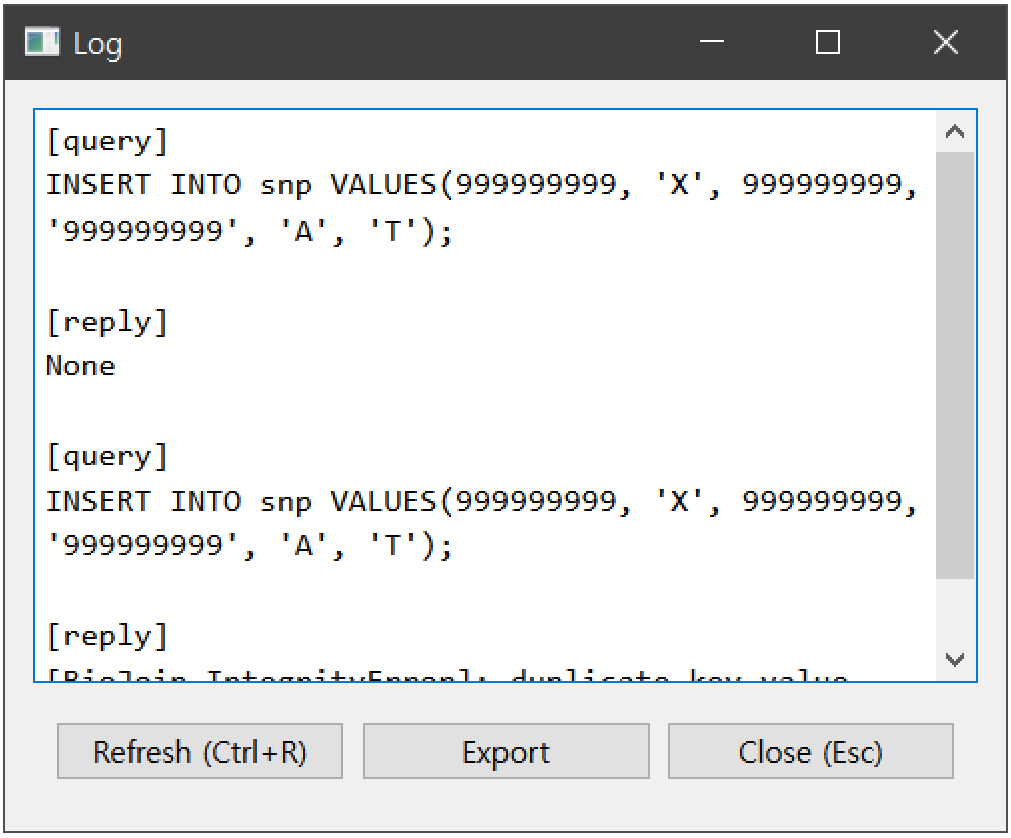


Figure 11. Genetic distance screen. (11-1, left) Genetic distance screen could be accessed by the last button in the main menu. (11-2, center) User selects genetic distance type at first. Here, the former one is selected. Then a reference county is selected by the dropdown list. (11-3, right) 'Show Figure' button would link the user to a web viewer and display the figure (Figure 2, 3).

BioJoin GUI supports a lot of convenient features like 'new window', 'go to home', keyboard shortcuts, etc.. It also contains a log system. SQL queries to the base and responses from the database (which might be 'None') are automatically recorded. The log could be exported to an external file (BioJoin.log).



```
[query]
INSERT INTO snp VALUES(999999999, 'X', 999999999,
'999999999', 'A', 'T');

[reply]
None

[query]
INSERT INTO snp VALUES(999999999, 'X', 999999999,
'999999999', 'A', 'T');

[reply]
[BioJoin IntegrityError]: duplicate key value
```

Figure 11. Log screen. Refresh button would reload the log list. Export button exports the log to an external file in the same directory.

3. References

[1] Ondrej Libiger, Caroline Nievergelt and Nicholas Schork, Comparison of Genetic Distance Measures Using Human SNP Genotype Data (BioOne, 2009), 389-406.

[2] SNP data API Base URL: https://clinicaltables.nlm.nih.gov/api/snps/v3/search
[3] Individual Based Genetic Distance for SNP Data
https://popgen.nescent.org/2015-05-18-Dist-SNP.html
[4] Human Genome Reference Sequence
https://www.ncbi.nlm.nih.gov/genome/guide/human/
[5] Genetic distance, Wikipedia https://en.wikipedia.org/wiki/Genetic_distance
[6] Statistical Graphics, Wikipedia https://en.wikipedia.org/wiki/Statistical_graphics