Bio-Join Phase 2 Presentation

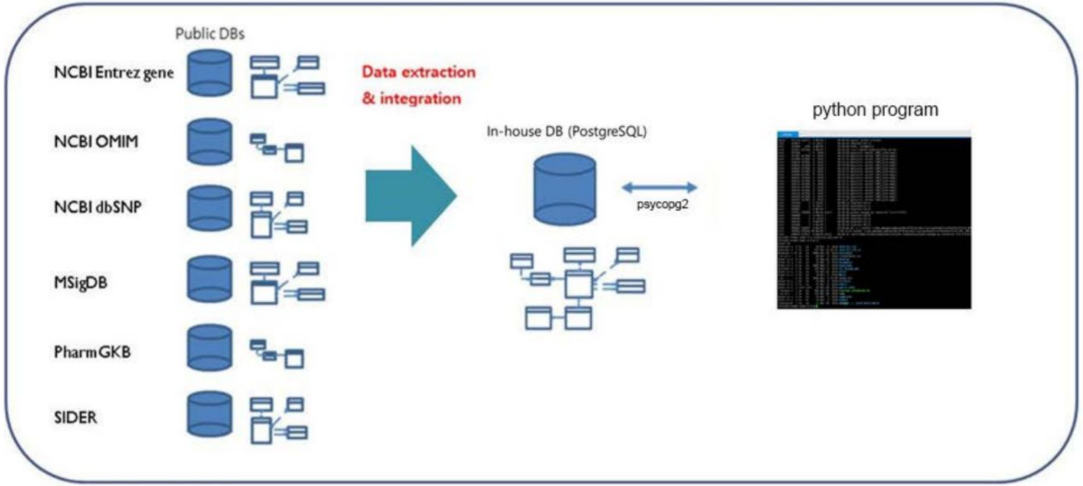
BiS332 course by Professor Doheon Lee

Surayouth Phuksawattanachai (20180813)

Alexander Semenov (20226159)

Lana Abu Hassan (20170842)

Overview

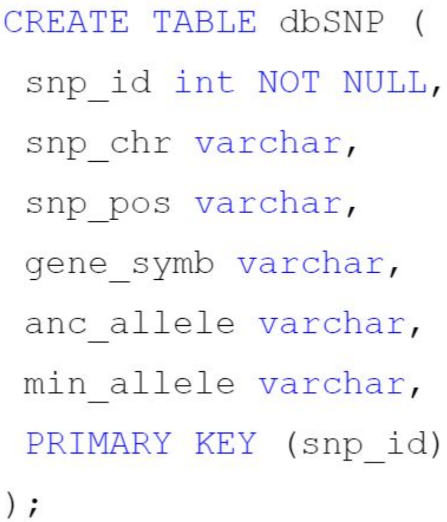


PostgreSQL

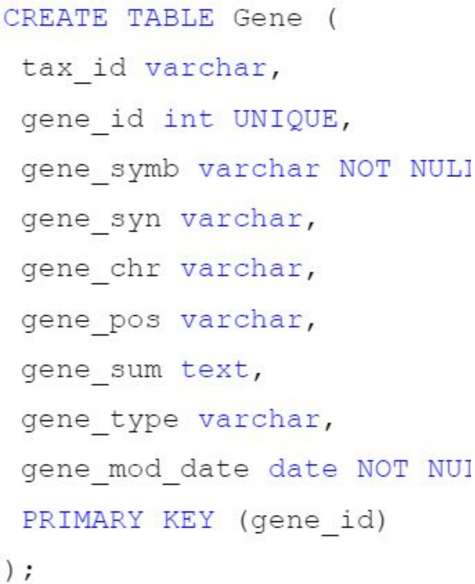
{JSON}

pandas

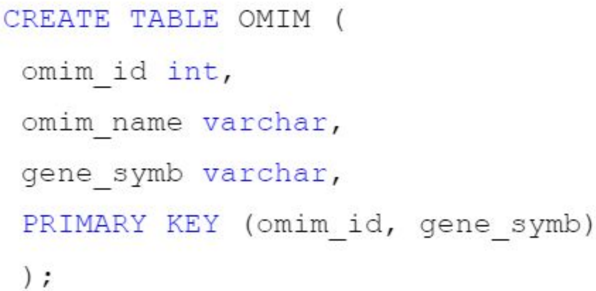
Table creation



```
CREATE TABLE dbSNP (
 snp_id int NOT NULL,
 snp_chr varchar,
 snp_pos varchar,
 gene_symb varchar,
 anc_allele varchar,
 min_allele varchar,
 PRIMARY KEY (snp_id)
);
```

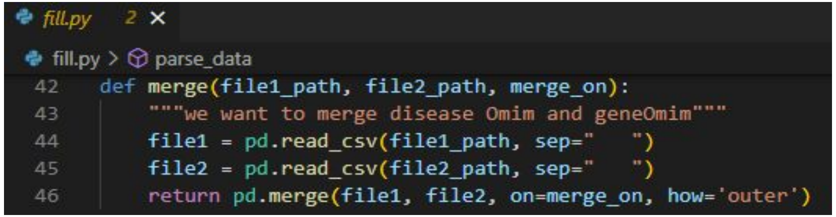


```
CREATE TABLE Gene (
 tax_id varchar,
 gene_id int UNIQUE,
 gene_symb varchar NOT NULI
 gene_syn varchar,
 gene_chr varchar,
 gene_pos varchar,
 gene_sum text,
 gene_type varchar,
 gene_mod_date date NOT NUI
 PRIMARY KEY (gene_id)
);
```

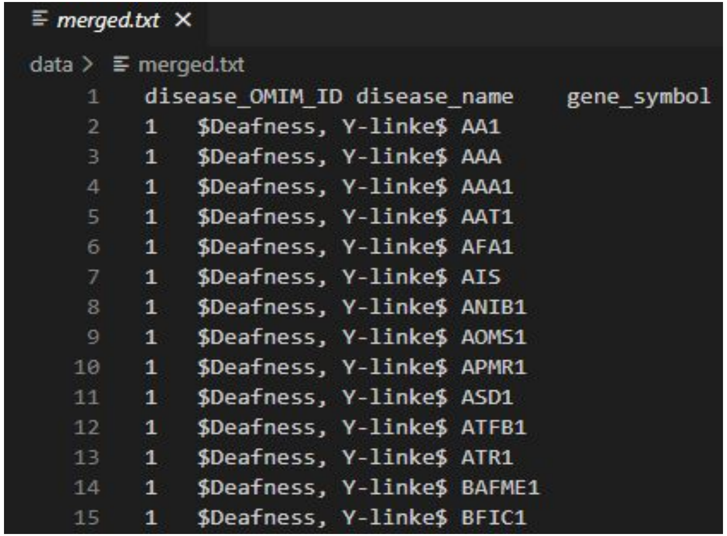


```
CREATE TABLE OMIM (
 omim_id int,
 omim_name varchar,
 gene_symb varchar,
 PRIMARY KEY (omim_id, gene_symb)
 );
```

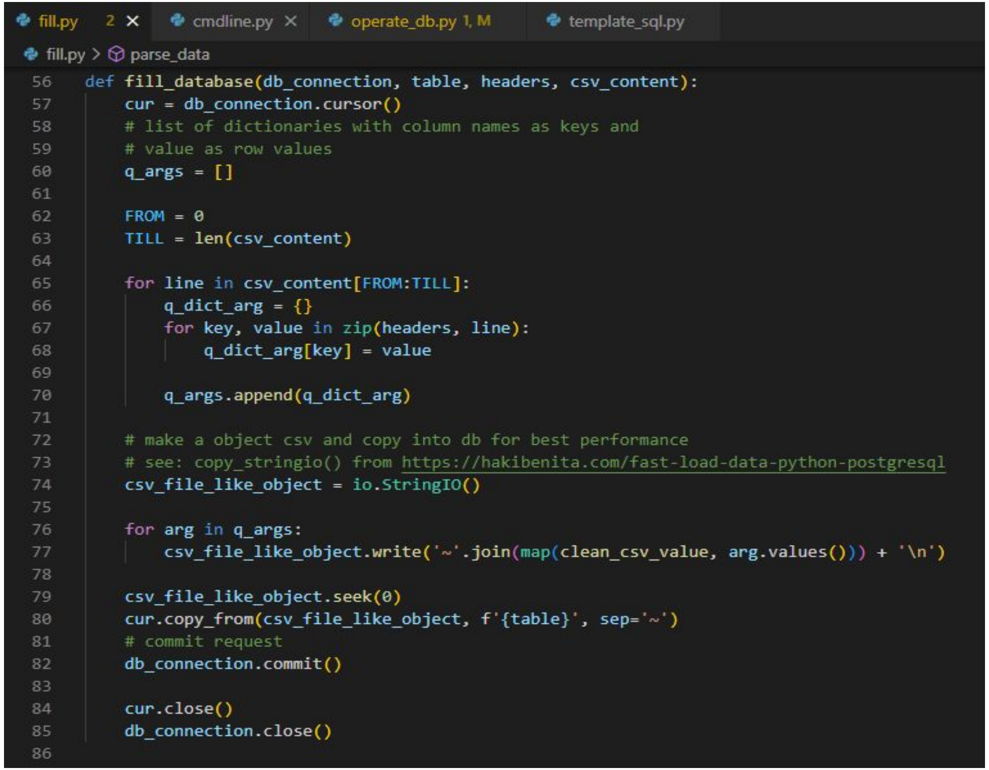
Database filling



```python
def merge(file1_path, file2_path, merge_on):
    """we want to merge disease Omim and geneOmim"""
    file1 = pd.read_csv(file1_path, sep="   ")
    file2 = pd.read_csv(file2_path, sep="   ")
    return pd.merge(file1, file2, on=merge_on, how='outer')
```

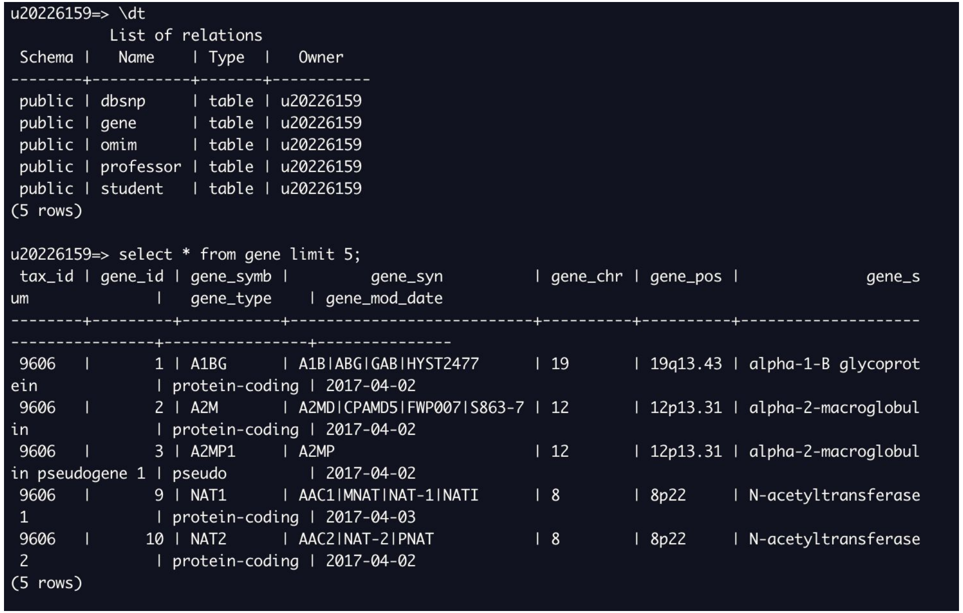


```
disease_OMIM_ID disease_name    gene_symbol
1    $Deafness, Y-linke$ AA1
1    $Deafness, Y-linke$ AAA
1    $Deafness, Y-linke$ AAA1
1    $Deafness, Y-linke$ AAT1
1    $Deafness, Y-linke$ AFA1
1    $Deafness, Y-linke$ AIS
1    $Deafness, Y-linke$ ANIB1
1    $Deafness, Y-linke$ AOMS1
1    $Deafness, Y-linke$ APMR1
1    $Deafness, Y-linke$ ASD1
1    $Deafness, Y-linke$ ATFB1
1    $Deafness, Y-linke$ ATR1
1    $Deafness, Y-linke$ BAFME1
1    $Deafness, Y-linke$ BFIC1
```

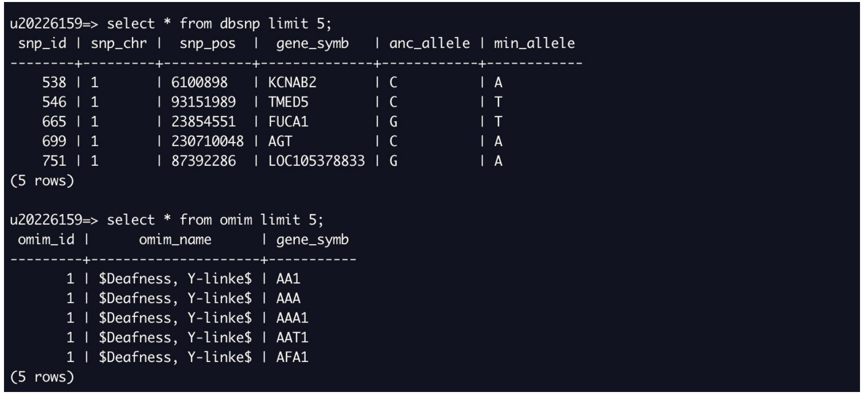


```python
def fill_database(db_connection, table, headers, csv_content):
    cur = db_connection.cursor()
    # list of dictionaries with column names as keys and
    # value as row values
    q_args = []

    FROM = 0
    TILL = len(csv_content)

    for line in csv_content[FROM:TILL]:
        q_dict_arg = {}
        for key, value in zip(headers, line):
            q_dict_arg[key] = value

        q_args.append(q_dict_arg)

    # make a object csv and copy into db for best performance
    # see: copy_stringio() from https://hakibenita.com/fast-load-data-python-postgresql
    csv_file_like_object = io.StringIO()

    for arg in q_args:
        csv_file_like_object.write('~'.join(map(clean_csv_value, arg.values())) + '\n')

    csv_file_like_object.seek(0)
    cur.copy_from(csv_file_like_object, f'{table}', sep='~')
    # commit request
    db_connection.commit()

    cur.close()
    db_connection.close()

```

Database filling



```
u20226159=> \dt
           List of relations
 Schema |   Name    | Type  |   Owner
--------+-----------+-------+-----------
 public | dbsnp     | table | u20226159
 public | gene      | table | u20226159
 public | omim      | table | u20226159
 public | professor | table | u20226159
 public | student   | table | u20226159
(5 rows)

u20226159=> select * from gene limit 5;
 tax_id | gene_id | gene_symb |         gene_syn          | gene_chr | gene_pos |              gene_s
um              |   gene_type    | gene_mod_date
--------+---------+-----------+---------------------------+----------+----------+--------------------
----------------+----------------+---------------
 9606   |       1 | A1BG      | A1B|ABG|GAB|HYST2477      | 19       | 19q13.43 | alpha-1-B glycoprot
ein             | protein-coding | 2017-04-02
 9606   |       2 | A2M       | A2MD|CPAMD5|FWP007|S863-7 | 12       | 12p13.31 | alpha-2-macroglobul
in              | protein-coding | 2017-04-02
 9606   |       3 | A2MP1     | A2MP                      | 12       | 12p13.31 | alpha-2-macroglobul
in pseudogene 1 | pseudo         | 2017-04-02
 9606   |       9 | NAT1      | AAC1|MNAT|NAT-1|NATI      | 8        | 8p22     | N-acetyltransferase
 1              | protein-coding | 2017-04-03
 9606   |      10 | NAT2      | AAC2|NAT-2|PNAT           | 8        | 8p22     | N-acetyltransferase
 2              | protein-coding | 2017-04-02
(5 rows)
```

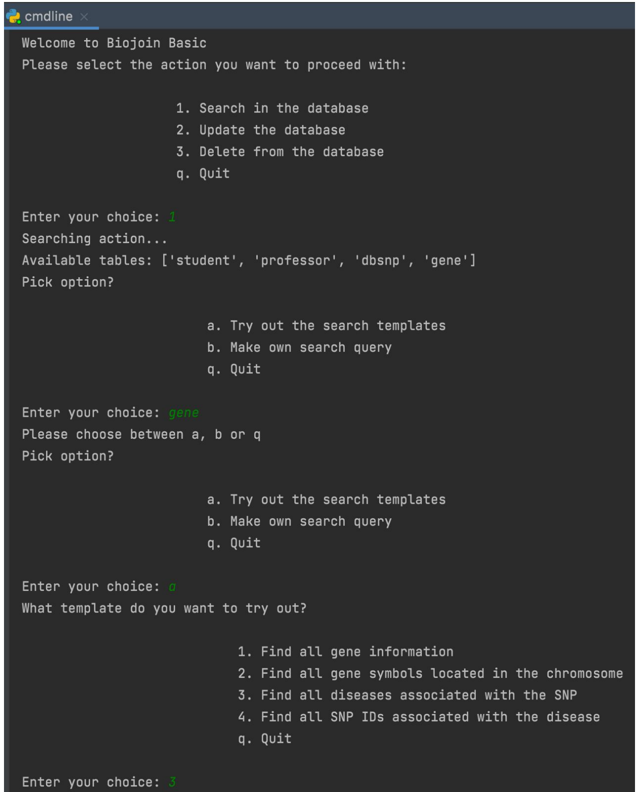


```
u20226159=> select * from dbsnp limit 5;
 snp_id | snp_chr |  snp_pos  |  gene_symb   | anc_allele | min_allele
--------+---------+-----------+--------------+------------+------------
    538 | 1       | 6100898   | KCNAB2       | C          | A
    546 | 1       | 93151989  | TMED5        | C          | T
    665 | 1       | 23854551  | FUCA1        | G          | T
    699 | 1       | 230710048 | AGT          | C          | A
    751 | 1       | 87392286  | LOC105378833 | G          | A
(5 rows)

u20226159=> select * from omim limit 5;
 omim_id |      omim_name      | gene_symb
---------+---------------------+-----------
       1 | $Deafness, Y-linke$ | AA1
       1 | $Deafness, Y-linke$ | AAA
       1 | $Deafness, Y-linke$ | AAA1
       1 | $Deafness, Y-linke$ | AAT1
       1 | $Deafness, Y-linke$ | AFA1
(5 rows)
```

Database manipulation software

Operations support:

1. Search
    a. Custom user search
    b. Template search (later explained)
2. Update of rows
3. Addition of rows
4. Deletion of row
5. Deletion of table



```
Welcome to Biojoin Basic
Please select the action you want to proceed with:

                    1. Search in the database
                    2. Update the database
                    3. Delete from the database
                    q. Quit

Enter your choice: 1
Searching action...
Available tables: ['student', 'professor', 'dbsnp', 'gene']
Pick option?

                        a. Try out the search templates
                        b. Make own search query
                        q. Quit

Enter your choice: gene
Please choose between a, b or q
Pick option?

                        a. Try out the search templates
                        b. Make own search query
                        q. Quit

Enter your choice: a
What template do you want to try out?

                            1. Find all gene information
                            2. Find all gene symbols located in the chromosome
                            3. Find all diseases associated with the SNP
                            4. Find all SNP IDs associated with the disease
                            q. Quit

Enter your choice: 3
```

Pre-defined search on the DB tables

1. Given a gene symbol, find all gene information stored in the gene table

   ```
   SELECT * FROM gene WHERE gene_symb = {gene_symbol};
   ```

2. Given a chromosome id, find all gene symbols located in the chromosome

   ```
   SELECT gene_symb FROM gene WHERE gene_chr = {chromosome};
   ```

3. Given an SNP ID, find all diseases associated with the SNP

   ```
   SELECT omim_name FROM omim WHERE omim.gene_symb
   IN (SELECT gene_symb FROM dbsnp WHERE snp_id = {snp_id});
   ```

4. Given a disease name, find all SNP IDs associated with the disease

   ```
   SELECT snp_id FROM dbsnp WHERE dbsnp.gene_symb
   IN SELECT gene_symb FROM omim WHERE omim_name = {disease_name});
   ```

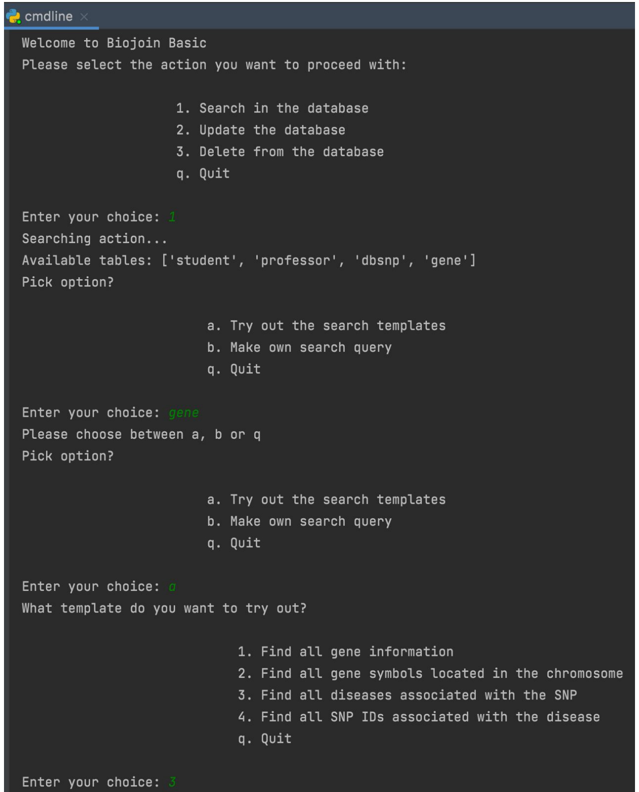
Thank you for you attention

Any questions?

Check out the source code on:
https://github.com/Tsatsch/Biojoin

Database manipulation software

Live demo :)



```
Welcome to Biojoin Basic
Please select the action you want to proceed with:

          1. Search in the database
          2. Update the database
          3. Delete from the database
          q. Quit

Enter your choice: 1
Searching action...
Available tables: ['student', 'professor', 'dbsnp', 'gene']
Pick option?

            a. Try out the search templates
            b. Make own search query
            q. Quit

Enter your choice: gene
Please choose between a, b or q
Pick option?

            a. Try out the search templates
            b. Make own search query
            q. Quit

Enter your choice: a
What template do you want to try out?

              1. Find all gene information
              2. Find all gene symbols located in the chromosome
              3. Find all diseases associated with the SNP
              4. Find all SNP IDs associated with the disease
              q. Quit

Enter your choice: 3
```