How to manage HGMD database in MySQL

Contents

Current maintenance

- Administrator: Yingying Zhang (yz2296@cornell.edu)

Raw data from HGMD (Jan 2021)

- Phenotypes:
  - HGMD_Phenbase_2021_1.sql
- Mutations:
  - hg38: 2021_1_hg38_fullinfo.vcf
  - hg19: 2021_1_hg19_fullinfo.vcf

Summary statistics after pre-processing

- Missense mutations (148,114):
  - Mapped to UniProt: 145,332
  - Unmapped: 2,782
- Nonsense mutations (34,405):
  - Mapped to UniProt: 33,168
  - Unmapped: 1,237
- Coding indels (64,392):
  - Mapped to UniProt: 61,491
  - Unmapped: 2,901
- Non-coding mutations (34,945)
- Dropped mutations with non-compliant format (5,288)

Link to processed data

- Indels in coding regions mapped to UniProt: File:2021 2 mapped codingindel.xlsx
- Missense mutations mapped to UniProt: File:2021 2 mapped missense.xlsx
- Nonsense mutations mapped to UniProt: File:2021 2 mapped nonsense.xlsx
- Non-coding mutations: File:2021 2 noncoding.xlsx
- Indels in coding regions which could not be mapped to UniProt: File:2021 2 unmapped codingindel.xlsx
- Missense mutations which could not be mapped to UniProt: File:2021 2 unmapped missense.xlsx
- Nonsense mutations which could not be mapped to UniProt: File:2021 2 unmapped nonsense.xlsx
- Raw data from HGMD: File:HGMD concept def.xlsx
- Raw data from HGMD: File:HGMD mutation.xlsx
- Raw data from HGMD: File:HGMD phenotype.xlsx

How to process HGMD raw data

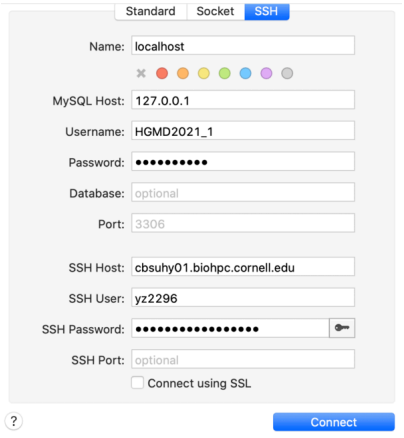
- Create a MySQL database called HGMD2021_1 (if not existed). Please find how to do so in #How to create a MySQL database.
- Import HGMD_Phenbase_2021_1.sql to HGMD2021_1. Please find how to do so in #How to import .sql file to an existing MySQL database. Add “HGMD_” as prefix to all imported tables. This is to distinguish original HGMD tables from the ones we generated.
- Process 2021_1_hg38_fullinfo.vcf:

1. The 7th column (index from 0) contains the information about each mutation. Parse the 7th column into 8 columns: Variant_class, Gene, Strand, dbSNP, Phenotype, Rank_score, RefSeq_DNA_record, and RefSeq_protein_record. If the 7th column does not contain RefSeq protein record (for example noncoding variants), set the value in RefSeq_protein_record to nan. For example, if the content in the 7th column is "CLASS=DM?;MUT=ALT;GENE=SAMD11;STRAND=+;DNA=NM_152486.4%3Ac.133A>G;PROT=NP_689699.3%3Ap.K45E;DB=rs903331232;PHEN="Retinit the values in the 8 columns following the order above should be: DM?, SAMD11, +, rs903331232, Retinitis_pigmentosa, 0.21, NM_152486.4:c.133A>G, NP_689699.3:p.K45E. Please notice that I replaced "%3A" with ":" for the values in RefSeq_DNA_record and RefSeq_protein_record. The values in RefSeq_protein_record stands for the amino acid change caused by the mutation. In this example, NP_689699.3:p.K45E stands for an amino acid substitution from K to E at position 45 in the protein NP_689699.3.
2. Add one more column POS_hg19 based on 2021_1_hg19_fullinfo.vcf. Map hg19 coordinates to hg38 coordinates using the HGMD ID of each variant.
3. Rename the columns and rearrange them so that you will get a dataframe with the following columns: CHR, POS, HGMD_ID, REF_N, ALT_N, POS_hg19, Variant_class, Gene, Strand, dbSNP, Phenotype, Rank_score, RefSeq_DNA_record, RefSeq_protein_record.

4. Pick out missense mutations from the parsed dataframe:
   - Missense mutations always have the format of r'.+:p.[A-Z]\d+[A-Z]' in RefSeq_protein_record (this is a python regular expression, please google to see what it means). Pick out all rows in the parsed dataframe where this pattern could be found in RefSeq_protein_record, and add two more columns REF_AA and ALT_AA to these rows, one for reference amino acid and one for altered amino acid. In the aforementioned example, the values in these two columns should be K and E.
   - Map missense mutations to protein positions using the pipeline written by Charles Liang: Genomics and protein coordinate conversion. After this step you should get two text files, **mapped_missense.txt** and **unmapped_missense.txt**. The columns in mapped_missense.txt should be: CHR, POS, HGMD_ID, REF_N, ALT_N, POS_hg19, REF_AA, ALT_AA, UNIPROT, UNIPROT_position, Variant_class, Gene, Strand, dbSNP, Phenotype, Rank_score, RefSeq_DNA_record, RefSeq_protein_record. These columns should also appear in unmapped_missense.txt except for UNIPROT and UNIPROT_position.
5. Pick out nonsense mutations from the parsed dataframe:
   - Nonsense mutations always have the format of r'.+:p.[A-Z]\d+*' in RefSeq_protein_record. Pick out all rows in the parsed dataframe where this pattern could be found in RefSeq_protein_record, and repeat the same procedures as what you did when picking out missense mutations. Finally you should get two text files, **mapped_nonsense.txt** and **unmapped_nonsense.txt**.
6. Pick out coding indels from the parsed dataframe:
   - From the parsed data frame, pick out variants whose REF_N and ALT_N have different length and RefSeq_protein_record is not nan. Put these variants into VEP(GRCh38): make sure to check the "Protein" and "UniProt" boxes under the "Identifiers" section, and also make sure to check "Transcript biotype" under the "Transcript annotation" section. In the "Filters" section, check the box saying "Return results for variants in coding regions only". In "Restrict results", select "Show one selected consequence per variant".
   - After getting VEP output, filter the output file and retain the rows where the SWISSPROT and TREMBL columns are not be both empty. There should be only one identifier in SWISSPROT and might be multiple identifiers in TREMBL (if so separated by ","). If the SWISSPROT or TREMBL identifiers are in the format of XXXXXX.XXX(for example P43489.17), only retain the content before "." (P43489). Then choose the best UniProt among SWISSPROT and TREMBL identifiers for each row, prioritizing SWISSPROT over TREMBL, and choosing the longest TREMBL identifier if there is no SWISSPROT. Naming the column containing the best UniProt as UNIPROT. Rename the Protein_position column as UNIPROT_position.
   - Parsing the Amino_acids column in the VEP output file into two columns: REF_AA and ALT_AA. For example, if the value in Amino_acids is "RLTQTV/X", the values in REF_AA and ALT_AA should be "RLTQTV" and "X" respectively.
   - Retain the six columns in VEP output: #Uploaded_variation, UNIPROT, UNIPROT_position, REF_AA, ALT_AA and Consequence, and map the VEP output to the coding indels using #Uploaded_variation as key. Finally you should get a text file named **mapped_coding_indel.txt** with 19 columns: CHR, POS, HGMD_ID, REF_N, ALT_N, POS_hg19, REF_AA, ALT_AA, UNIPROT, UNIPROT_position, Variant_class, Gene, Strand, dbSNP, Phenotype, Rank_score, RefSeq_DNA_record, RefSeq_protein_record, Consequence. Also you should get **unmapped_coding_indel.txt** with 14 columns (missing REF_AA, ALT_AA, UNIPROT, UNIPROT_position, and Consequence).
7. Pick out noncoding variants from the parsed dataframe:
   - From the parsed data frame, pick out variants whose RefSeq_protein_record is nan. Put the selected rows into **noncoding.txt**.
8. Put the remaining variants that do not belong to any of the aforementioned categories into **other_variants.txt**.
9. Import all text files you generated from the previous steps as tables into HGMD2021_1. Please find how to do so in #How to create and delete a table in an existing MySQL database.

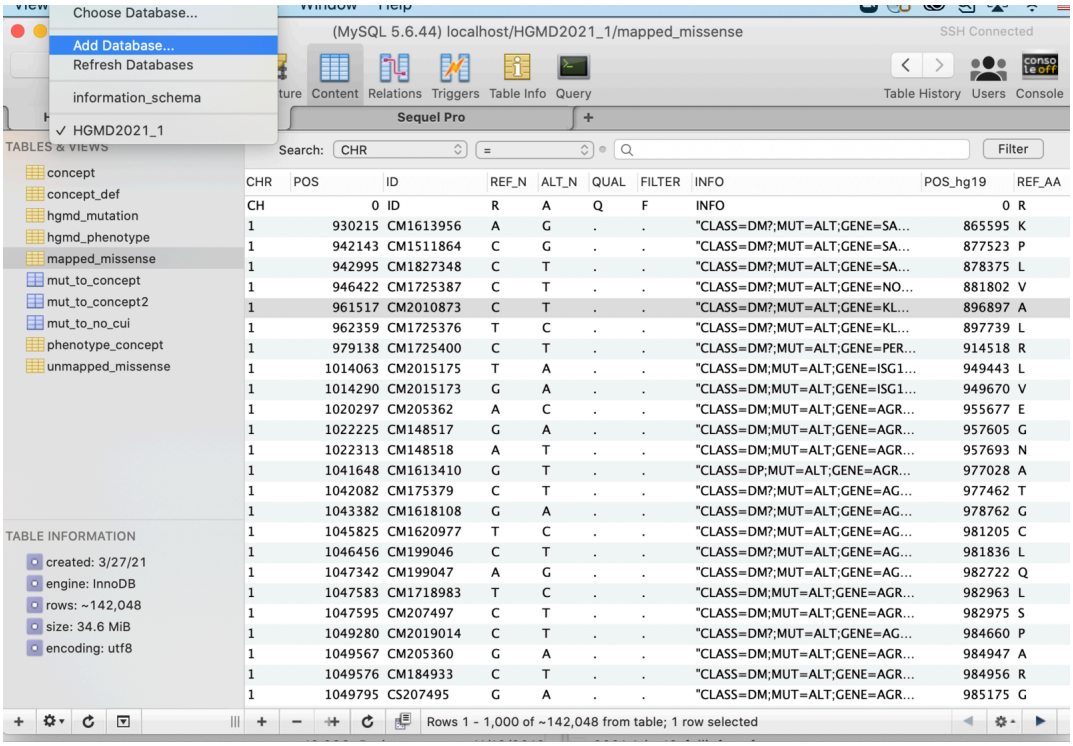
How to create and drop a MySQL database

- Please contact Yingying Zhang (yz2296@cornell.edu) for the MySQL account and password
- Download sequel pro to your computer: http://sequelpro.com/
- When the first time you launch the sequel pro application , the following window will pop up:



Type in "127.0.0.1" to "MySQL Host".
Type in the MySQL account username and password to "Username" and "Password".
Type in "cbsuhy01.biohpc.cornell.edu" to "SSH Host"
Type in your NetID to "SSH User"
Type in your ssh password to cbsuhy01 to "SSH Password"
Then you should have successfully logged in to MySQL on cbsuhy01

- If you want to create a new database, click the "Choose Database..." button at the upper left corner of the sequel pro interface, and you will see a drop-down list, among which there is an entry called "Add Database...". By clicking it you can create a new database and specify the name.



CHR	POS	ID	REF_N	ALT_N	QUAL	FILTER	INFO	POS_hg19	REF_AA
CH	0	ID	R	A	Q	F	INFO	0	R
1	930215	CM1613956	A	G	.	.	"CLASS=DM?;MUT=ALT;GENE=SA...	865595	K
1	942143	CM1511864	C	G	.	.	"CLASS=DM?;MUT=ALT;GENE=SA...	877523	P
1	942995	CM1827348	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=SA...	878375	L
1	946422	CM1725387	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=NO...	881802	V
1	961517	CM2010873	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=KL...	896897	A
1	962359	CM1725376	T	C	.	.	"CLASS=DM?;MUT=ALT;GENE=KL...	897739	L
1	979138	CM1725400	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=PER...	914518	R
1	1014063	CM2015175	T	A	.	.	"CLASS=DM;MUT=ALT;GENE=ISG1...	949443	L
1	1014290	CM2015173	G	A	.	.	"CLASS=DM;MUT=ALT;GENE=ISG1...	949670	V
1	1020297	CM205362	A	C	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	955677	E
1	1022225	CM148517	G	A	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	957605	G
1	1022313	CM148518	A	T	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	957693	N
1	1041648	CM1613410	G	T	.	.	"CLASS=DP;MUT=ALT;GENE=AGR...	977028	A
1	1042082	CM175379	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=AG...	977462	T
1	1043382	CM1618108	G	A	.	.	"CLASS=DM?;MUT=ALT;GENE=AG...	978762	G
1	1045825	CM1620977	T	C	.	.	"CLASS=DM?;MUT=ALT;GENE=AG...	981205	C
1	1046456	CM199046	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=AG...	981836	L
1	1047342	CM199047	A	G	.	.	"CLASS=DM?;MUT=ALT;GENE=AG...	982722	Q
1	1047583	CM1718983	T	C	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	982963	L
1	1047595	CM207497	C	T	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	982975	S
1	1049280	CM2019014	C	T	.	.	"CLASS=DM?;MUT=ALT;GENE=AG...	984660	P
1	1049567	CM205360	G	A	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	984947	A
1	1049576	CM184933	C	T	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	984956	R
1	1049795	CS207495	G	A	.	.	"CLASS=DM;MUT=ALT;GENE=AGR...	985175	G

- If you want to drop a database, use the following command:

```
DROP DATABASE database_name;
```

How to create and delete a table in an existing MySQL database

- If you want to create a new table in an existing database, for example a table "STUDENT_INFO" composed of two columns "NAME" and "AGE" in a database "STUDENTS_2021", use the following command:

```
mysql> use STUDENTS_2021;
mysql> CREATE TABLE STUDENT_INFO (NAME varchar(20), AGE Int);
```

- If you want to import a text file to an existing table, for to import student_info.txt to the table STUDENT_INFO, use the following command (replace /path/to/student_info.txt with the path to the text file that you want to import. If your text file does not have a header line, remove "IGNORE 1 LINES" from the following command):

```
mysql> LOAD DATA LOCAL INFILE '/path/to/student_info.txt' INTO TABLE STUDENT_INFO IGNORE 1 LINES;
```

- If you want to permanently delete a table, type the following command (Replace "table_name" with the real table name):

```
mysql> DROP TABLE table_name;
```

How to import .sql file to an existing MySQL database

- If you want to import a .sql file to an existing MySQL database, please login to cbsuhy01 through your terminal, go to the folder where the .sql file is stored, and then type in the following command:

```
sed 's/\sDEFINER=`[^`]*`@`[^`]*`//g' -i your_sql_file.sql
```

Please replace "your_sql_file.sql" with the name of the .sql file that you want to import. The command above is to remove the "DEFINER=.." statement from your .sql file, so that you won't run into any access problem.
Then type in the following command to import the .sql file to MySQL database:

```
mysql -u username -p database_name < your_sql_file.sql
```

Please replace "username", "database_name" with your MySQL account user name and the existing MySQL database name, respectively. Please also replace "your_sql_file.sql" with the name of the .sql file that you want to import. As you've already in the right folder, there is no need to include a file path.

How to export a MySQL table to a .csv file

```
mysql -B -u username -ppassword database_name -h localhost -e "SELECT * from table_name" |sed "s/'/\'/;s/\t/\",\"/g;s/^/\"/;s/$/\"/;s/\n//g" > /path/to/your/csv_file
```

Please replace "username" and "password" with the MySQL account username and password from Yingying Zhang. Please notice that there is no space between "-p" and your password [IMPORTANT]. Replace "database_name" and "table_name" with the real name of the database and table that you want to export. Replace "/path/to/your/csv_file" with the path to the output .csv file.

Basic commands to navigate MySQL on cbsuhy01

- ssh login to cbsuhy01:

```
net_id@cbsuhy01.biohpc.cornell.edu (please replace "net_id" with your own NetID)
```

- Type in your terminal:

```
mysql –u username –p
```

Please replace "username" with your MySQL account user name and then type in your MySQL account password. If your account exists, the following information will pop up:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.6.44 MySQL Community Server (GPL)
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

To get started on your own database, first check which databases currently exist:

```
mysql> show databases;
```

To the select a database, issue the use command (Replace "database_name" with the real database name that you want to use):

```
mysql> use database_name;
mysql> show tables;
```
