Integrated Database

Documentation Manual

Thahmina Ali, Lei Xie*
∗Corresponding Author: lxie@iscb.org

Contents

1 Data Initial Input

Open up terminal and create a directory: 'database_initial_input'. Inside the directory obtain the starting data via their ftp links in the following. Note: wget and unzip if necessary all starting data.

1.1 UniProtKB fasta sequences:

wget ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fa
gunzip uniprot_sprot.fasta.gz

1.2 RGD (disease ontology, chemical ontology, phenotype ontology, orthologs):

1.2.1 Disease ontology:

Human: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/homo_terms_do
Mouse: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/mus_terms_rdo
Rat: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/rattus_terms_rdo

1.2.2 Drug ontology:

Human: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/homo_terms_chebi
Mouse: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/mus_terms_chebi
Rat: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/rattus_terms_chebi

1.2.3 Phenotype ontology:

Human: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/homo_terms_mp
Mouse: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/mus_terms_mp
Rat: wget ftp://ftp.rgd.mcw.edu/pub/data_release/with_terms/rattus_terms_mp

1.2.4 Rat gene orthologs to Human genes:

wget ftp://ftp.rgd.mcw.edu/pub/data_release/RGD_ORTHOLOGS_HomoloGene.txt

1.3 DisGeNET:

Human:
wget www.disgenet.org/ds/DisGeNET/results/all_gene_disease_associations.tsv.gz
gunzip all_gene_disease_associations.tsv.gz

1.4 MGI:

Mouse gene orthologs to Human genes:
wget http://www.informatics.jax.org/downloads/reports/HMD_HumanPhenotype.rpt

1.5 IMPC:

Mouse gene orthologs to Human genes:

```
wget ftp://ftp.ebi.ac.uk/pub/databases/impc/latest/csv/ALL_genotype_phenotype.csv.gz
gunzip ALL_genotype_phenotype.csv.gz
```

1.6 Chebi:

Chebi IDs:

```
wget ftp://ftp.ebi.ac.uk/pub/databases/chebi/Flat_file_tab_delimited/chebiId_inchi_3star.tsv
gunzip chebiId_inchi_3star.tsv
```

1.7 Chembl:

Chembl IDs, smiles notation:

```
wget ftp://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBLdb/latest/chembl_23_chemreps.txt.gz
gunzip chembl_23_chemreps.txt.gz
```

1.8 MedGen:

Concept IDs (CUI), MESH, OMIM:

```
wget ftp://ftp.ncbi.nlm.nih.gov/pub/medgen/csv/MGSAT_1.csv.gz
gunzip MGSAT_1.csv.gz
wget ftp://ftp.ncbi.nlm.nih.gov/pub/medgen/csv/MGSAT_2.csv.gz
gunzip MGSAT_2.csv.gz
```

2 Data Process

In order to make the various associations, all data is required to be in tabular form in columns which will require some preprocessing steps to format.

2.1 Convert Uniprot data from fasta format to tabular:

```
cat uniprot_sprot.fasta | sed ’s/->/[]/g’| perl -pe ’s/>(.*)/> \1 \t/g’ | perl -pe ’s/ \n//g’
| perl -pe ’s/>/\n>/g’ | grep -E -v ’∧$’| sed ’s/|/\t/g’ | sed ’s/_/\t/g’ | cut -f 2,3,4,5 |
sed ’s/ /_/1’ | sed ’s/_/\t/g’ > uniprot_sprot.tab
```

2.2 Remove beginning comments of the data obtained from RGD:

2.2.1 Mouse Orthologs:

Remove extra spaces, column, commented lines:

```
cut -f 1,2,3,4,5,6 HMD_HumanPhenotype.rpt | sed ’s/ /\t/g’> HMD_HumanPhenotype.tab
```

2.2.2 Disease Ontology:

```
sed ’1,28d’ rattus_terms_rdo > rattus_terms_rdo.tab
sed ’1,28d’ homo_terms_rdo >homo_terms_rdo.tab
sed ’1,28d’ mus_terms_rdo > mus_terms_rdo.tab
```

2.2.3 Drug Ontology:

```
sed ’1,28d’ rattus_terms_chebi > rattus_terms_chebi.tab
sed ’1,28d’ homo_terms_chebi > homo_terms_chebi.tab
sed ’1,28d’ mus_terms_chebi > mus_terms_chebi.tab
```

2.2.4 Phenotype Ontology:

```
sed ’1,28d’ rattus_terms_mp > rattus_terms_mp.tab
sed ’1,28d’ homo_terms_mp > homo_terms_mp.tab
sed ’1,28d’ mus_terms_mp > mus_terms_mp.tab
```

2.2.5 RGD Rat orthologs to Human genes:

```
sed ’1,54d’ RGD_ORTHOLOGS_HomoloGene.txt > rgd_orthologs.tab
```

2.2.6 Chembl data:

```
sed -i ’1d’ chembl_23_chemreps.txt
```

2.3 Obtain Medgen CUI terms with respective MESH/OMIM (for use in human gene-disease DisGeNET step):

```
sed ’s/,/\t/g’ MGSAT_1.csv > MGSAT_1.tab
sed ’s/,/\t/g’ MGSAT_2.csv > MGSAT_2.tab
cat MGSAT_1.tab MGSAT_2.tab > MGSAT_all.tab
sed ’s/’ ’//g’ MGSAT_all.tab | awk ’$5=="NCBI_OMIM" {print $0}’ > MGSAT_omim
sed ’s/’ ’//g’ MGSAT_all.tab | awk ’$7=="MSH" print $0’ > MGSAT_mesh
awk ’{print $1"\t""MESH:"$4}’ MGSAT_mesh > MGSAT_mesh2
awk ’{print $1"\t""OMIM:"$7}’ MGSAT_mesh > MGSAT_omim2
cat MGSAT_omim2 MGSAT_mesh2 > MGSAT_mesh_omim
```

2.3.1 Remove duplicates:

```
awk ’!seen[$0]++’ MGSAT_mesh_omim > MGSAT_mesh_omim_final
```

2.4 Label chebi ID to match labeling in RGD ontology files:

```
awk ’{print "CHEBI:"$1"\t"$2}’ chebiId_inchi_3star.tsv> chebi_ids
```

2.5 Convert IMPC mouse phenotype ontology csv file to tabular file:

```
sed ’s/,/\t/g’ ALL_genotype_phenotype.csv > mouse_phenotype_impc.tab
```

3 Data Final Input:

Create a directory: ’database_final_input’. Inside the directory move all the following processed 17 inputs located in the database_initial_input into the database_final_input folder:

```
  mv /path/to/database_initial_input/all_gene_disease_associations.tsv /path/to/database_final_input
  mv /path/to/database_initial_input/chebi_ids /path/to/database_final_input
  mv /path/to/database_initial_input/chembl_23_chemreps.txt /path/to/database_final_input
  mv /path/to/database_initial_input/HMD_HumanPhenotype.tab /path/to/database_final_input
  mv /path/to/database_initial_input/homo_terms_chebi.tab /path/to/database_final_input
  mv /path/to/database_initial_input/homo_terms_mp.tab /path/to/database_final_input
  mv /path/to/database_initial_input/homo_terms_rdo.tab /path/to/database_final_input
  mv /path/to/database_initial_input/MGSAT_mesh_omim_final /path/to/database_final_input
  mv /path/to/database_initial_input/mouse_phenotype_impc.tab /path/to/database_final_input
  mv /path/to/database_initial_input/mus_terms_chebi.tab /path/to/database_final_input
  mv /path/to/database_initial_input/mus_terms_mp.tab /path/to/database_final_input
  mv /path/to/database_initial_input/mus_terms_rdo.tab /path/to/database_final_input
  mv /path/to/database_initial_input/rattus_terms_chebi.tab /path/to/database_final_input
  mv /path/to/database_initial_input/rattus_terms_mp.tab /path/to/database_final_input
  mv /path/to/database_initial_input/rattus_terms_rdo.tab /path/to/database_final_input
  mv /path/to/database_initial_input/rgd_orthologs.tab /path/to/database_final_input
  mv /path/to/database_initial_input/uniprot_sprot.tab /path/to/database_final_input
```

4 Database Tables

4.1 Creating database of 17 tables:

```
mysql> CREATE DATABASE integrated_database;
mysql> USE integrated_database;
```

4.1.1 Rat orthologs:

```
mysql> CREATE TABLE rat_orthologs ( rat_gene_symbol VARCHAR(30), rat_rgd
LONGTEXT, rat_ncbi LONGTEXT, human_orth_rat VARCHAR(30), human_orth_rgd
LONGTEXT, human_orth_rat_ncbi LONGTEXT, human_orth_rat_source LONGTEXT,
mouse_orth_rat LONGTEXT, mouse_orth_rat_rgd LONGTEXT, mouse_orth_rat_ncbi
LONGTEXT, mouse_orth_rat_mgi LONGTEXT, mouse_orth_rat_source LONGTEXT,
human_orth_rat_hgnc LONGTEXT);
```

4.1.2 Mouse orthologs:

```
mysql> CREATE TABLE mouse_orthologs ( human_gene_symbol VARCHAR(30), entrez_id LONGTEXT, homolog_id LONGTEXT, hgnc_assoc VARCHAR(30), mouse_gene_symbol LONGTEXT, mouse_mgi_id LONGTEXT);
```

4.1.3 Uniprot data:

```
mysql> CREATE TABLE uniprot_data (uniprot_id VARCHAR(30), gene_symbol VARCHAR(30), organism_symbol VARCHAR(30), gene_desc LONGTEXT, sequence LONGTEXT, PRIMARY KEY(uniprot_id));
```

4.1.4 Chebi data:

```
mysql> CREATE TABLE chebi_data (chebi_id VARCHAR(30), inchi LONGTEXT);
```

4.1.5 Chembl data:

```
mysql> CREATE TABLE chembl_data (chembl_id VARCHAR(30), smiles LONGTEXT, inchi LONGTEXT, inchi_key LONGTEXT);
```

4.1.6 MedGen data:

```
mysql> CREATE TABLE medgen_to_mesh_omim (medgen_id VARCHAR(30), mesh_omim_id VARCHAR(30));
```

4.1.7 DisGeNET data:

```
mysql> CREATE TABLE human_disgenet(gene_id LONGTEXT, gene_symbol VARCHAR(30), disease_id VARCHAR(30), disease_name LONGTEXT, score LONGTEXT, NofPmids LONGTEXT, NofSnps LONGTEXT, source LONGTEXT);
```

4.1.8 IMPC Mouse Phenotype Ontology:

```
mysql> CREATE TABLE mouse_phene_impc(mgi_id VARCHAR(30), gene_symbol VARCHAR(30), phenotyping LONGTEXT, colony_id LONGTEXT, sex VARCHAR(30), zygosity VARCHAR(30), allele_accession_id VARCHAR(30), allele_symbol LONGTEXT, allele_name LONGTEXT, strain_accession_id VARCHAR(30), strain_name VARCHAR(30), project_name VARCHAR(30), project_fullname LONGTEXT, pipeline_name LONGTEXT, pipeline_stable_id VARCHAR(30), procedure_stable_id LONGTEXT, procedure_name LONGTEXT, parameter_stable_id LONGTEXT, parameter_name LONGTEXT, top_level_mp_term_id VARCHAR(30), top_level_mp_term_name LONGTEXT, mp_term_id LONGTEXT, mp_term_name LONGTEXT, p_value LONGTEXT, percentage_change LONGTEXT, effect_size LONGTEXT, statistical_method LONGTEXT, resource_name LONGTEXT);
```

4.1.9 Disease Ontology-

Human:

mysql> CREATE TABLE human_disease_rgd (rgd_id INT(6), gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), rdo_id VARCHAR(30), disease_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);

Mouse:

mysql> CREATE TABLE mouse_disease_rgd (rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), rdo_id VARCHAR(30), disease_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);

Rat:

mysql> CREATE TABLE rat_disease_rgd (rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), rdo_id VARCHAR(30), disease_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);

4.1.10 Chemical Ontology-

Human:

mysql> CREATE TABLE human_chemical_rgd (rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), chebi_id VARCHAR(30), chemical_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);

Rat:

mysql> CREATE TABLE rat_chemical_rgd (rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), chebi_id VARCHAR(30), chemical_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT,

```
association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);
```

Mouse:

```
mysql> CREATE TABLE mouse_chemical_rgd ( rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), chebi_id VARCHAR(30), chemical_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);
```

4.1.11 Phenotype Ontology-

Human:

```
mysql> CREATE TABLE human_phene_rgd ( rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), phene_id VARCHAR(30), phene_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);
```

Rat:

```
mysql> CREATE TABLE rat_phene_rgd ( rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), phene_id VARCHAR(30), phene_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT, assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);
```

Mouse:

```
mysql> CREATE TABLE mouse_phene_rgd ( rgd_id LONGTEXT, gene_symbol VARCHAR(30), gene_desc LONGTEXT, gene_type VARCHAR(30), phene_id VARCHAR(30), phene_name LONGTEXT, qualifier LONGTEXT, evidence LONGTEXT, association VARCHAR(30), aspect LONGTEXT, ref LONGTEXT, created_date LONGTEXT assigned_by LONGTEXT, mesh_omim_id VARCHAR(30), curation_notes LONGTEXT, original_reference LONGTEXT);
```

5 Database Tables Input Load

Load the mysql tables with the input files either using a script which has accumulated the commands that uses the ’insert’ option to load the data from the input files into the tables, or via the ’load data infile’ option:

5.1 Insert Option:

Inside the data_final_input directory with all the inputs located copy and paste the following code into an editor and save the file as ’insert_mysql.py’. Change the username (user), password (passwd) and database name (db) to respective information. Run the following command inside the data_final_input:

```
$ python insert_mysql.py
```

5.1.1 insert_mysql.py code:

```
#!/usr/bin/env python
import MySQLdb, csv, sys
conn = MySQLdb.connect (host = "127.0.0.1",user = "testuser", passwd = "testuser",db = "integrated_database")
c = conn.cursor()
rat_orthologs=csv.reader(file("rgd_orthologs.tab"), delimiter=’\t’)
for row in rat_orthologs:
    c.execute("INSERT INTO rat_orthologs (rat_gene_symbol, rat_rgd, rat_ncbi, human_orth_rat, human_orth_rgd, human_orth_rat_ncbi, human_orth_rat_source, mouse_orth_rat, mouse_orth_rat_rgd, mouse_orth_rat_ncbi, mouse_orth_rat_mgi, mouse_orth_rat_source, human_orth_rat_hgnc) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s )", row)
mouse_orthologs=csv.reader(file("HMD_HumanPhenotype.rpt"), delimiter=’\t’)
for row in mouse_orthologs:
    c.execute("INSERT INTO mouse_orthologs (human_gene_symbol, entrez_id, homolog_id, hgnc_assoc, mouse_gene_symbol, mouse_mgi_id) VALUES (%s, %s, %s, %s, %s, %s)", row)
uniprot_data=csv.reader(file("uniprot_sprot.tab"),delimiter=’\t’)
for row in uniprot_data:
    c.execute("INSERT INTO uniprot_data (uniprot_id, gene_symbol, organism_symbol, gene_desc, sequence) VALUES (%s, %s, %s, %s, %s)", row)
chebi_data=csv.reader(file("chebi_ids"), delimiter=’\t’)
for row in chebi_data:
    c.execute("INSERT INTO chebi_data (chebi_id, inchi) VALUES (%s, %s)", row)
chembl_data=csv.reader(file("chembl_23_chemreps.txt"), delimiter=’\t’)
for row in chembl_data:
    c.execute("INSERT INTO chembl_data (chembl_id, smiles, inchi, inchi_key) VALUES
(%s, %s, %s, %s)", row)
medgen_to_mesh_omim=csv.reader(file("MGSAT_mesh_omim_final"), delimiter='\t')
for row in medgen_to_mesh_omim:
   c.execute("INSERT INTO medgen_to_mesh_omim (medgen_id, mesh_omim_id)
VALUES (%s, %s)", row) human_disgenet=csv.reader(file("all_gene_disease_associations.tsv"),
delimiter='\t')
for row in human_disgenet:
   c.execute("INSERT INTO human_disgenet (gene_id, gene_symbol, disease_id,
disease_name, score, NofPmids, NofSnps, source) VALUES (%s, %s, %s, %s, %s, %s, %s,
%s),", row)
mouse_phene_impc=csv.reader(file("mouse_phenotype_impc.tab"), delimiter='\t')
for row in mouse_phene_impc:
   c.execute("INSERT INTO mouse_phene_impc (mgi_id, gene_symbol, phenotyping,
colony_id, sex, zygosity, allele_acession_id, allele_symbol, allele_name, strain_accession_id,
strain_name, project_name, project_fullname, pipeline_name, pipeline_stable_id,
procedure_stable_id, procedure_name, parameter_stable_id, parameter_name, top_level_mp_term_id,
top_level_mp_term_name, mp_term_id, mp_term_name, p_value, percentage_change,
effect_size, statistical_method, resource_name) VALUES (%s, %s, %s, %s, %s, %s, %s,
%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)",
row)
human_disease_rgd=csv.reader(file("homo_terms_rdo.tab"), delimiter='\t')
for row in human_disease_rgd:
   c.execute("INSERT INTO human_disease_rgd (rgd_id, gene_symbol, gene_desc,
gene_type, rdo_id, disease_name, qualifier, evidence, association, aspect, ref,
created_date, assigned_by, mesh_omim_id, curation_notes, original_reference) VALUES
(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
mouse_disease_rgd=csv.reader(file("mouse_terms_rdo.tab"), delimiter='\t')
for row in mouse_disease_rgd:
   c.execute("INSERT INTO mouse_disease_rgd (rgd_id, gene_symbol, gene_desc,
gene_type, rdo_id, disease_name, qualifier, evidence, association, aspect, ref,
created_date, assigned_by, mesh_omim_id, curation_notes, original_reference) VALUES
(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
rat_disease_rgd=csv.reader(file("rattus_terms_rdo.tab"), delimiter='\t')
for row in rat_disease_rgd:
   c.execute("INSERT INTO rat_disease_rgd (rgd_id, gene_symbol, gene_desc, gene_type,
rdo_id, disease_name, qualifier, evidence, association, aspect, ref, created_date,
assigned_by, mesh_omim_id, curation_notes, original_reference) VALUES (%s, %s, %s,
%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
human_chemical_rgd=csv.reader(file("homo_terms_chebi.tab"), delimiter='\t')
for row in human_chemical_rgd:
   c.execute("INSERT INTO human_chemical_rgd (rgd_id, gene_symbol, gene_desc,
gene_type, chebi_id, chemical_name, qualifier, evidence, association, aspect, ref,
created_date, assigned_by, mesh_omim_id,curation_notes, original_reference) VALUES
```

```
(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
rat_chemical_rgd=csv.reader(file("rattus_terms_chebi.tab"), delimiter=’\t’)
for row in rat_chemical_rgd:
   c.execute("INSERT INTO rat_chemical_rgd (rgd_id, gene_symbol, gene_desc, gene_type, chebi_id, chemical_name, qualifier, evidence, association, aspect, ref, created_date, assigned_by, mesh_omim_id,curation_notes, original_reference) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)" , row)
mouse_chemical_rgd=csv.reader(file("mus_terms_chebi.tab"), delimiter=’\t’)
for row in mouse_chemical_rgd:
   c.execute("INSERT INTO mouse_chemical_rgd (rgd_id, gene_symbol, gene_desc, gene_type, chebi_id, chemical_name, qualifier, evidence, association, aspect, ref, created_date, assigned_by, mesh_omim_id,curation_notes, original_reference) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)" , row)
human_phene_rgd=csv.reader(file("homo_terms_mp.tab"), delimiter=’\t’)
for row in human_phene_rgd:
   c.execute("INSERT INTO human_phene_rgd (rgd_id, gene_symbol, gene_desc, gene_type, phene_id, phene_name, qualifier, evidence, association, aspect, ref, created_date, assigned_by, mesh_omim_id, curation_notes, original_reference) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
rat_phene_rgd=csv.reader(file("rattus_terms_mp.tab"), delimiter=’\t’)
for row in rat_phene_rgd:
   c.execute("INSERT INTO rat_phene_rgd (rgd_id, gene_symbol, gene_desc, gene_type, phene_id, phene_name, qualifier, evidence, association, aspect, ref, created_date, assigned_by, mesh_omim_id, curation_notes, original_reference) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
mouse_phene_rgd=csv.reader(file("mouse_terms_mp.tab"), delimiter=’\t’)
for row in mouse_phene_rgd:
   c.execute("INSERT INTO mouse_terms_mp.tab (rgd_id, gene_symbol, gene_desc, gene_type, phene_id, phene_name, qualifier, evidence, association, aspect, ref, created_date, assigned_by, mesh_omim_id, curation_notes, original_reference) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", row)
conn.commit()
c.close()
```

5.1.2 Load Data Infile option

1. Rat orthologs:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/rgd_ortho
INTO TABLE rgd_orthologs;
```

2. Mouse orthologs:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/HMD_Hu
INTO TABLE mouse_orthologs;
```

3. Uniprot data:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/uniprot_sp
INTO TABLE uniprot_data;
```

4. Chebi data:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/chebi_idsâ
INTO TABLE chebi_data;
```

5. Chembl data:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/chembl_23
INTO TABLE chembl_data;
```

6. MedGen data:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/MGSAT_r
INTO TABLE medgen_to_mesh_omim;
```

7. DisGeNET data:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/all_gene_
INTO TABLE human_disgenet;
```

8. IMPC Mouse Phenotype Ontology:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/mouse_ph
INTO TABLE mouse_phene_impc;
```

Disease Ontology-

9. Human:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/homo_term
INTO TABLE human_disease_rgd;
```

10. Mouse:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/mouse_ter
INTO TABLE mouse_disease_rgd;
```

11. Rat:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/rattus_ter
INTO TABLE rat_disease_rgd;
```

Chemical Ontology-

12. Human:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/homo_term
INTO TABLE human_chemical_rgd;
```

13. Rat:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/rattus_ter
INTO TABLE rat_chemical_rgd;
```

14. Mouse:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/mus_terma
INTO TABLE mouse_chemical_rgd;
```

Phenotype Ontology-

15. Human:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/homo_term
INTO TABLE human_phene_rgd;
```

16. Rat:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/rattus_ter
INTO TABLE rat_phene_rgd;
```

17. Mouse:

```
mysql> LOAD DATA INFILE ’/var/lib/mysql-files/integrated_database/database_final_input/mouse_ter
INTO TABLE mouse_phene_rgd;
```

6 Queries

6.1 Join Queries:

6.1.1 Join on common human genes to obtain rat and mouse orthologs:

```
mysql> CREATE TABLE mouse_rat_association SELECT DISTINCT rat_orthologs.human_orth_rat
as ’human_gene_symbol’, rat_orthologs.rat_gene_symbol, mouse_orthologs.mouse_gene_symbol
from rat_orthologs,mouse_orthologs WHERE rat_orthologs.human_orth_rat=mouse_orthologs.human_ger
```

6.1.2 Join on human, mouse and rat gene symbol on uniprot file to obtain uniprot id:

Obtain human uniprot ids:

```
mysql> CREATE TABLE mouse_rat_association_human SELECT DISTINCT mouse_rat_association.huma
uniprot_data.uniprot_id, ’NULL’ as human_ortholog, ’HUMAN’ as organism from
mouse_rat_association, uniprot_data WHERE mouse_rat_association.human_gene_symbol=uniprot_data
AND uniprot_data.organism_symbol=’HUMAN’;
```

Obtain mouse uniprot ids:

```
mysql> CREATE TABLE mouse_rat_association_mouse SELECT DISTINCT mouse_rat_association.mous
uniprot_data.uniprot_id, mouse_rat_association.human_gene_symbol as human_ortholog,
’MOUSE’ as organism from mouse_rat_association, uniprot_data WHERE mouse_rat_association.mouse_ge
AND uniprot_data.organism_symbol=’MOUSE’;
```

Obtain rat uniprot ids:

```
mysql> CREATE TABLE mouse_rat_association_rat SELECT DISTINCT mouse_rat_association.rat_gene
uniprot_data.uniprot_id, mouse_rat_association.human_gene_symbol as human_ortholog,
’RAT’ as organism from mouse_rat_association, uniprot_data WHERE mouse_rat_association.rat_gene_syr
AND uniprot_data.organism_symbol=’RAT’;
```

6.1.3 Join and combine all chebi ids with respective chembl ids based on its common chemical representation (InchI) and obtain smiles notation:

```
mysql> CREATE TABLE chebi_to_chembl SELECT chembl_data.chembl_id, chembl_data.inchi,
chebi_data.chebi_id, chembl_data.smiles from chembl_data, chebi_data WHERE
chembl_data.inchi= chebi_data.inchi;
```

6.1.4 Join all chebi ids of human, mouse and rat files from RGD against the mapped chembl_to_chebi file to obtain the chembl version of the chebi ids and smiles notation for each of gene-chebi association in respective organisms:

Human:

```
mysql> CREATE TABLE human_chebi_chembl SELECT human_chemical_rgd.gene_symbol,
human_chemical_rgd.chemical_name, chebi_to_chembl.chembl_id, chebi_to_chembl.smiles
from human_chemical_rgd, chebi_to_chembl WHERE chebi_to_chembl.chebi_id=
human_chemical_rgd.chebi_id;
```

Mouse:

```
mysql> CREATE TABLE mouse_chebi_chembl SELECT mouse_chemical_rgd.gene_symbol,
mouse_chemical_rgd.chemical_name, chebi_to_chembl.chembl_id, chebi_to_chembl.smiles
from mouse_chemical_rgd, chebi_to_chembl WHERE chebi_to_chembl.chebi_id=
mouse_chemical_rgd.chebi_id;
```

Rat:

```
mysql> CREATE TABLE rat_chebi_chembl SELECT rat_chemical_rgd.gene_symbol,
rat_chemical_rgd.chemical_name, chebi_to_chembl.chembl_id, chebi_to_chembl.smiles
from rat_chemical_rgd, chebi_to_chembl WHERE chebi_to_chembl.chebi_id= rat_chemical_rgd.chebi_id
```

7 Entity Association Queries:

7.1 Gene-Disease:

7.1.1 Human:

Use the human genes from the mouse_rat_association_human table and associate it with the disgenet genes:

```
mysql> CREATE TABLE human_disgenet_genes SELECT DISTINCT mouse_rat_association_human.hum
mouse_rat_association_human.uniprot_id, human_disgenet.disease_id, human_disgenet.score
as disgenet_score from mouse_rat_association_human, human_disgenet WHERE
mouse_rat_association_human.human_gene_symbol=human_disgenet.gene_symbol;
```

Convert cui from the disgenet genes ids to mesh/omim using the medgen_to_mesh_omim table:

```
mysql> CREATE TABLE human_disgenet_m_o SELECT DISTINCT human_disgenet_genes.human_gene
human_disgenet_genes.uniprot_id, medgen_to_mesh_omim.mesh_omim_id, 'MedGen'
as source_id, human_disgenet_genes.disgenet_score, 'DisGeNET' as source_attr
from medgen_to_mesh_omim, human_disgenet_genes WHERE medgen_to_mesh_omim.medgen_id=hum
```

Compare the human_disease_rgd against the mouse_rat_association_final to obtain uniprot ids of the human genes presented in the human_disease_rgd.

```
mysql> CREATE TABLE human_disease_rgd_uniprot SELECT human_disease_rgd.gene_symbol
as human_gene_symbol, mouse_rat_association_human.uniprot_id, human_disease_rgd.mesh_omim_id,
'NULL' as source_id, 'NULL' as disease_score, 'RGD' as source_attr from human_disease_rgd,
mouse_rat_association_human WHERE human_disease_rgd.gene_symbol= mouse_rat_association_huma
```

Compare the human_disgenet_m_o genes with the human_disease_rgd_uniprot genes to capture any missing associations from the RGD database:

```
mysql> CREATE TABLE human_disease_rgd_disgenet SELECT DISTINCT * from human_disgenet_m_o
UNION ALL SELECT DISTINCT * from human_disease_rgd_uniprot;
```

```
mysql> CREATE TABLE human_gene_disease_final SELECT human_gene_symbol,
uniprot_id, mesh_omim_id, source_id, ' NULL ' as human_ortholog, ' NULL' as
mouse_rat_assoc, disgenet_score, source_attr from human_disease_rgd_disgenet GROUP
BY human_gene_symbol, mesh_omim_id;
```

7.1.2 Mouse:

Use the mouse genes from the mouse_rat_association_final table and join on common mouse genes against the RGD disease ontology file and obtain the mesh/omim ids associations:

```
mysql> CREATE TABLE mouse_gene_disease_final SELECT DISTINCT mouse_rat_association_mouse.m
mouse_rat_association_mouse.uniprot_id, mouse_disease_rgd.mesh_omim_id, 'NULL'
as source_id, mouse_rat_association_mouse.human_ortholog, ' ' as mouse_rat_assoc,
'NULL' as disgenet_score, ' RGD' as source_attr from mouse_rat_association_mouse,
mouse_disease_rgd WHERE mouse_rat_association_mouse.mouse_gene_symbol =
mouse_disease_rgd.gene_symbol;
```

Same Mouse genes with Human:

```
mysql> CREATE TABLE human_mouse_same_disease SELECT * from mouse_gene_disease_final
WHERE EXISTS (SELECT * from human_gene_disease_final WHERE mouse_gene_disease_final.mesh_o
AND human_gene_disease_final.human_gene_symbol=mouse_gene_disease_final.human_ortholog);
```

Distinct Mouse genes from Human:

```
mysql> CREATE TABLE human_mouse_distinct_disease SELECT * from mouse_gene_disease_final
WHERE NOT EXISTS (SELECT * from human_gene_disease_final WHERE mouse_gene_disease_final.me
AND human_gene_disease_final.human_gene_symbol=mouse_gene_disease_final.human_ortholog);
```

7.1.3 Rat:

Use the rat genes from the mouse_rat_association_final table and join on common rat genes against the rdo file and obtain the mesh and omim ids associations:

```
mysql> CREATE TABLE rat_gene_disease_final SELECT DISTINCT mouse_rat_association_rat.rat_gen
mouse_rat_association_rat.uniprot_id, rat_disease_rgd.mesh_omim_id, 'NULL' as
source_id, mouse_rat_association_rat.human_ortholog , ' ' as mouse_rat_assoc,
'NULL' as disgenet_score, 'RGD' as source_attr from mouse_rat_association_rat,
rat_disease_rgd WHERE mouse_rat_association_rat.rat_gene_symbol = rat_disease_rgd.gene_symbol;
```

Same Rat genes with Human:

```
mysql> CREATE TABLE human_rat_same_disease SELECT * from rat_gene_disease_final
WHERE EXISTS (SELECT * from human_gene_disease_final WHERE rat_gene_disease_final.mesh_omim
AND human_gene_disease_final.human_gene_symbol=rat_gene_disease_final.human_ortholog);
```

Distinct Rat genes from Human:

```
mysql> CREATE TABLE human_rat_distinct_disease SELECT * from rat_gene_disease_final
WHERE NOT EXISTS (SELECT * from human_gene_disease_final WHERE rat_gene_disease_final.mesh_
AND human_gene_disease_final.human_gene_symbol=rat_gene_disease_final.human_ortholog);
```

Inserting values for columns that will distinguish matched and unmatched mouse and rat genes against human genes:

```
mysql> UPDATE human_rat_same_disease SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE human_rat_distinct_disease SET mouse_rat_assoc =' indirect_assoc
'; mysql> UPDATE human_mouse_same_disease SET mouse_rat_assoc ='direct_assoc
'; mysql> UPDATE human_mouse_distinct_disease SET mouse_rat_assoc =' indirect_assoc ';
```

Unify all of the tables of the mouse and rat that have direct and indirect disease associations with humans , along with the human gene-disease association table: human_rat_same_disease, human_rat_distinct_disease, human_mouse_same_disease, human_mouse_distinct_disease, human_gene_disease_final:

```
mysql> CREATE TABLE gene_disease_final (SELECT uniprot_id, mesh_omim_id,
source_id, human_ortholog, mouse_rat_assoc, disgenet_score, source_attr, 'human'
as organism from human_gene_disease_final) UNION ALL (SELECT uniprot_id,
mesh_omim_id, source_id, human_ortholog, mouse_rat_assoc, disgenet_score, source_attr,
'rat' as organism from human_rat_same_disease) UNION ALL (SELECT uniprot_id,
mesh_omim_id, source_id, human_ortholog, mouse_rat_assoc, disgenet_score, source_attr,
'rat' as organism from human_rat_distinct_disease) UNION ALL (SELECT uniprot_id,
mesh_omim_id, source_id, human_ortholog, mouse_rat_assoc, disgenet_score, source_attr,
'mouse' as organism from human_mouse_distinct_disease) UNION ALL (SELECT
uniprot_id, mesh_omim_id, source_id, human_ortholog, mouse_rat_assoc, disgenet_score,
source_attr, 'mouse' as organism from human_mouse_same_disease);
```

7.2 Gene-Drug:

7.2.1 Human

Use the human genes from the mouse_rat_association_final table and join on common human genes against the RGD chebi file and obtain the chembl chemical associations:

```
mysql> CREATE TABLE human_gene_drug_final SELECT DISTINCT mouse_rat_association_human.hu
mouse_rat_association_human.uniprot_id, human_chebi_chembl.chembl_id, 'chembl'
as source_id, 'NULL' as human_ortholog, 'NULL' as mouse_rat_assoc, 'RGD' as
source_attr from mouse_rat_association_human, human_chebi_chembl WHERE mouse_rat_association_h
= human_chebi_chembl.gene_symbol;
```

7.2.2 Mouse

Use the mouse genes from the mouse_rat_association_final table and join on common mouse genes against the RGD chebi file and obtain the chembl chemical associations:

```
mysql> CREATE TABLE mouse_gene_drug_final SELECT DISTINCT mouse_rat_association_mouse.mou
, mouse_rat_association_mouse.uniprot_id, mouse_chebi_chembl.chembl_id, 'chembl'
as source_id, mouse_rat_association_mouse.human_ortholog, ' ' as mouse_rat_assoc,
'RGD' as source_attr from mouse_rat_association_mouse, mouse_chebi_chembl WHERE
mouse_rat_association_mouse.mouse_gene_symbol = mouse_chebi_chembl.gene_symbol;
```

Same Mouse genes with Human:

```
mysql> CREATE TABLE human_mouse_same_drug SELECT * from mouse_gene_drug_final
WHERE EXISTS (SELECT * from human_gene_drug_final WHERE mouse_gene_drug_final.chembl_id=h
AND human_gene_drug_final.human_gene_symbol=mouse_gene_drug_final.human_ortholog);
```

Distinct Mouse genes from Human:

```
mysql> CREATE TABLE human_mouse_distinct_drug SELECT * from mouse_gene_drug_final
WHERE NOT EXISTS (SELECT * from human_gene_drug_final WHERE mouse_gene_drug_final.chembl_
AND human_gene_drug_final.human_gene_symbol=mouse_gene_drug_final.human_ortholog);
```

7.2.3 Rat

Use the rat genes from the mouse_rat_association_final table and join on common rat genes against the RGD chebi file and obtain the chembl chemical associations:

```
mysql> CREATE TABLE rat_gene_drug_final SELECT DISTINCT mouse_rat_association_rat.rat_gene_
mouse_rat_association_final.uniprot_id, rat_chebi_chembl.chembl_id, âĂŸchemblâĂŹ
as source_id, mouse_rat_association_rat.human_ortholog, âĂŸ âĂŹ as mouse_rat_assoc,
âĂŸRGDâĂŹ as source_attr from mouse_rat_association_rat, rat_chebi_chembl WHERE
mouse_rat_association_rat.rat_gene_symbol = rat_chebi_chembl.gene_symbol;
```

Same Rat genes with Human:

```
mysql> CREATE TABLE human_rat_same_drug SELECT * from rat_gene_drug_final
WHERE EXISTS (SELECT * from human_gene_drug_final WHERE rat_gene_drug_final.chembl_id=hum
AND human_gene_drug_final.human_gene=rat_gene_drug_final.human_ortholog);
```

Distinct Rat genes from Human:

```
mysql> CREATE TABLE human_rat_distinct_drug SELECT * from rat_gene_drug_final
WHERE NOT EXISTS (SELECT * from human_gene_drug_final WHERE rat_gene_drug_final.chembl_id
AND human_gene_drug_final.human_gene=rat_gene_drug_final.human_ortholog);
```

Inserting values for columns that will distinguish matched and unmatched mouse and rat genes against human genes:

```
mysql> UPDATE human_rat_same_drug SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE human_rat_distinct_drug SET mouse_rat_assoc =' indirect_assoc';
mysql> UPDATE human_mouse_same_drug SET mouse_rat_assoc =' direct_assoc
'; mysql> UPDATE human_mouse_distinct_drug SET mouse_rat_assoc =' indirect_assoc';
```

Unify all of the tables of the human, mouse and rat (direct and indirect disease associations with humans): human_gene_drug_final, human_rat_same_drug, human_rat_distinct_drug, human_mouse_same_drug, human_mouse_distinct_drug:

```
mysql> CREATE TABLE gene_drug_final (SELECT human_gene_drug_final.* ,
âĂŸhumanâĂŹ as organism from human_gene_drug_final) UNION ALL (SELECT
human_rat_same_disease.* , âĂŸratâĂŹ as organism from human_rat_same_drug
) UNION ALL (SELECT human_rat_distinct_drug.*, âĂŸratâĂŹ as organism from
human_rat_distinct_drug) UNION ALL (SELECT human_mouse_same_drug.* , âĂŸmouseâĂŹ
as organism from human_mouse_same_drug) UNION ALL (SELECT human_mouse_distinct_drug.*
, âĂŸmouseâĂŹ as organism human_mouse_distinct_drug);
```

Drug-Disease:

For the human, mouse and rat, join based on common gene from the gene_disease_final and gene_drug_final tables to grab the mesh/omim and chebi ids which will show drug-disease associations.

7.2.4 Human

mysql> CREATE TABLE human_drug_disease_final SELECT DISTINCT human_gene_disease_final.mesh human_gene_disease_final.source_id as disease_source, human_gene_drug_final.chembl_id, human_gene_drug_final.source_id as chemical_source, 'NULL' from human_gene_disease_final, human_gene_drug_final WHERE human_gene_drug_final.uniprot_id = human_gene_disease_final.unipro

7.2.5 Mouse

mysql> CREATE TABLE mouse_drug_disease_same SELECT DISTINCT human_mouse_same_disease.me human_mouse_same_disease.source_id, human_mouse_same_drug.chembl_id, human_mouse_same_drug.source_id, human_mouse_same_disease.human_ortholog, âĂŸ âĂŸ as mouse_rat_assoc from human_mouse_same_disease, human_mouse_same_drug WHERE human_mouse_same_disease.uniprot_id = human_mouse_same_drug.uniprot_id AND human_mouse_same_disease.human_ortholog = human_mouse_same_drug.human_ortholog;

mysql> CREATE TABLE mouse_drug_disease_distinct SELECT DISTINCT human_mouse_distinct_disease.mesh_omim_id, human_mouse_distinct_disease.source_id, human_mouse_distinct_drug.chembl_id, human_mouse_distinct_drug.source_id human_mouse_distinct_disease.human_ortholog, âĂŸ âĂŸ as mouse_rat_assoc, from human_mouse_distinct_disease, human_mouse_distinct_drug WHERE human_mouse_distinct_disease.unip = human_mouse_distinct_drug.uniprot_id AND human_mouse_distinct_disease.human_ortholog = human_mouse_distinct_drug.human_ortholog;

7.2.6 Rat

mysql> CREATE TABLE rat_drug_disease_same SELECT DISTINCT human_rat_same_disease.mesh_om human_rat_same_disease.source_id, human_rat_same_drug.chembl_id, human_rat_same_drug.source_i human_rat_same_disease.human_ortholog, âĂŸ âĂŸ as mouse_rat_assoc, from human_rat_same_disease, human_rat_same_drug WHERE human_rat_same_disease.uniprot_id = human_rat_same_drug.uniprot_id AND human_rat_same_disease.human_ortholog = human_rat_same_drug.human_ortholog;

mysql> CREATE TABLE rat_drug_disease_distinct SELECT DISTINCT human_rat_distinct_disease.m human_rat_distinct_disease.source_id, human_rat_distinct_drug.chembl_id, human_mouse_distinct_dru human_rat_distinct_disease.human_ortholog, âĂŸ âĂŸ as mouse_rat_assoc, from human_rat_distinct_disease, human_rat_distinct_drug WHERE human_rat_distinct_disease.uniprot_id = human_rat_distinct_drug.uniprot_id AND human_rat_distinct_disease.human_ortholog = human_rat_distinct_drug.human_ortholog;

Inserting values for columns that will distinguish matched and unmatched mouse genes against human genes:

```
mysql> UPDATE rat_drug_disease_same SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE rat_drug_disease_distinct SET mouse_rat_assoc =' indirect_assoc';
mysql> UPDATE mouse_drug_disease_distinct SET mouse_rat_assoc =' direct_assoc
'; mysql> UPDATE mouse_drug_disease_same SET mouse_rat_assoc =' indirect_assoc';
```

unify-

```
mysql> CREATE TABLE drug_disease_final (SELECT human_drug_disease_final.* ,
âĂŸhumanâĂŹ as organism from human_drug_disease_final) UNION ALL (SELECT
rat_drug_disease_same.* , âĂŸratâĂŹ as organism from rat_drug_disease_same)
UNION ALL (SELECT rat_drug_disease_distinct.*, âĂŸratâĂŹ as organism from
rat_drug_disease_distinct) UNION ALL (SELECT mouse_drug_disease_distinct.* ,
âĂŸmouseâĂŹ as organism from mouse_drug_disease_distinct) UNION ALL (SELECT
mouse_drug_disease_same.* , âĂŸmouseâĂŹ as organism mouse_drug_disease_same);
```

Phenotype-Gene:

7.2.7 Human

Use the human genes from the mouse_rat_association_final table and join on common human genes against the RGD phenotype file and obtain the phenotype associations:

```
mysql> CREATE TABLE human_gene_phene_final SELECT DISTINCT mouse_rat_association_human.h
mouse_rat_association_human.uniprot_id, human_phene_rgd.phene_id, âĂŸNUL-
LâĂŹ as human_ortholog, âĂŸNULLâĂŹ as mouse_rat_assoc, âĂŸRGDâĂŹ as source_attr
from mouse_rat_association_human, human_phene_rgd WHERE mouse_rat_association_human.human_
= human_phene_rgd.gene_symbol;
```

7.2.8 Mouse

Use the mouse genes from the mouse_rat_association_final table and join on common mouse genes against the RGD phenotype file and obtain the phenotype associations:

```
mysql> CREATE TABLE mouse_gene_phene_rgd SELECT DISTINCT mouse_rat_association_mouse.mo
mouse_rat_association_mouse.uniprot_id, mouse_phene_rgd.phene_id, mouse_rat_association_mouse.h
âĂŸ âĂŹ as mouse_rat_assoc, âĂŸRGDâĂŹ as source_attr from mouse_rat_association_mouse,
mouse_phene_rgd WHERE mouse_rat_association_mouse.mouse_gene_symbol =
mouse_phene_rgd.gene_symbol;
```

impc

```
mysql> CREATE TABLE mouse_gene_phene_impc SELECT DISTINCT mouse_rat_association_mouse.m
mouse_rat_association_mouse.uniprot_id, mouse_phene_impc.mp_term_name as
âĂŸphene_idâĂŹ, mouse_rat_association_mouse.human_ortholog, âĂŸ âĂŹ as mouse_rat_assoc,
âĂŸIMPCâĂŹ as source_attr from mouse_rat_association_mouse, mouse_phene_impc
WHERE mouse_rat_association_mouse.mouse_gene_symbol= mouse_phene_impc.gene_symbol;
```

distinct impc

```
mysql> CREATE TABLE mouse_gene_phene_impc_distinct SELECT * from mouse_gene_phene_impc
WHERE NOT EXISTS (SELECT * from mouse_gene_phene_rgd WHERE mouse_gene_phene_rgd.phene_
mouse_gene_phene_impc.phene_id AND mouse_gene_phene_impc.human_ortholog=
mouse_gene_phene_rgd.human_ortholog);
```

combine rgd and impc mouse gene-phenes:

```
mysql> CREATE TABLE mouse_gene_phene_final (SELECT * from mouse_gene_phene_rgd)
UNION ALL (SELECT * from mouse_gene_phene_impc_distinct)
```

same

```
mysql> CREATE TABLE human_mouse_same_phene SELECT * from mouse_gene_phene_final
WHERE EXISTS (SELECT * from human_gene_phene_final WHERE mouse_gene_phene_final.phene_id=
human_gene_phene_final.phene_id AND mouse_gene_phene_final.human_ortholog=
human_gene_phene_final.human_ortholog);
```

distinct

```
mysql> CREATE TABLE human_mouse_distinct_phene SELECT * from mouse_gene_phene_final
WHERE NOT EXISTS (SELECT * from human_gene_phene_final WHERE mouse_gene_phene_final.phen
human_gene_phene_final.phene_id AND mouse_gene_phene_final.human_ortholog=
human_gene_phene_final.human_ortholog);
```

7.2.9 Rat

Use the rat genes from the mouse_rat_association_final table and join on common rat genes against the RGD phenotype file and obtain the phenotype associations:

```
mysql> CREATE TABLE rat_gene_phene_final SELECT DISTINCT mouse_rat_association_rat.rat_gene
mouse_rat_association_rat.uniprot_id, rat_phene_rgd.phene_id, mouse_rat_association_rat.human_orth
âĂŸ âĂŹ as mouse_rat_assoc, âĂŸRGDâĂŹ as source_attr from mouse_rat_association_rat,
rat_phene_rgd WHERE mouse_rat_association_rat.rat_gene_symbol = rat_phene_rgd.gene_symbol;
```

same-

```
mysql> CREATE TABLE human_rat_same_phene SELECT * from rat_gene_phene_final
WHERE EXISTS (SELECT * from human_gene_phene_final WHERE rat_gene_phene_final.phene_id=
human_gene_phene_final.phene_id AND rat_gene_phene_final.human_ortholog=
human_gene_phene_final.human_ortholog);
```

distinct-

```
mysql> CREATE TABLE human_rat_distinct_phene SELECT * from rat_gene_phene_final
WHERE NOT EXISTS (SELECT * from human_gene_phene_final WHERE rat_gene_phene_final.phene_i
human_gene_phene_final.phene_id AND rat_gene_phene_final.human_ortholog=
human_gene_phene_final.human_ortholog);
```

Inserting values for columns that will distinguish matched and unmatched mouse genes against human genes:

```
mysql> UPDATE human_rat_phene_same SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE human_rat_phene_distinct SET mouse_rat_assoc =' indirect_assoc';
mysql> UPDATE human_mouse_phene_same SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE human_mouse_phene_same SET mouse_rat_assoc =' indirect_assoc';
```

unify-

```
mysql> CREATE TABLE gene_phene_final (SELECT * from human_phene_gene_final)
UNION ALL (SELECT * from human_rat_phene_same) UNION ALL (SELECT * from
human_rat_phene_distinct) UNION ALL (SELECT * from human_mouse_phene_same)
UNION ALL (SELECT * from human_mouse_phene_same);
```

Phenotype-Disease:

For the human, mouse and rat, join based on common gene from the gene_disease_final and phene_gene_final tables to grab the mesh/omim and phene ids which will show drug-disease associations.

7.2.10 Human

```
mysql> CREATE TABLE human_phene_disease_final SELECT DISTINCT human_gene_disease_final.mes
human_gene_disease_final.source_id as âĂŸsource_id_diseaseâĂŹ, human_phene_gene_final.phene_id,
human_phene_gene_final.source_attr as âĂŸsource_id_pheneâĂŹ, âĂŸNULLâĂŹ as
human_ortholog, âĂŸNULLâĂŹ as mouse_rat_assoc from human_gene_disease_final,
human_phene_gene_final WHERE human_phene_gene_final.uniprot_id = human_gene_disease_final.uni
```

7.2.11 Mouse

```
mysql> CREATE TABLE mouse_phene_disease_same SELECT DISTINCT human_mouse_same_disease.n
human_mouse_same_disease.source_id as âĂŸsource_id_diseaseâĂŹ, human_mouse_phene_same.phene_
```

human_mouse_phene_same.source_attri as âĂŸsource_id_pheneâĂŹ, human_mouse_same_disease.human
âĂŸ âĂŸ as mouse_rat_assoc from human_mouse_same_disease, human_mouse_phene_same
WHERE , human_mouse_same_disease.uniprot_id = human_mouse_phene_same.uniprot_id
AND human_mouse_same_disease.human_ortholog = human_mouse_phene_same.human_ortholog;

mysql> CREATE TABLE mouse_phene_disease_distinct SELECT DISTINCT human_mouse_distinct_disease.mesh_omim_id, human_mouse_distinct_disease.source_id
as âĂŸsource_id_diseaseâĂŹ, human_mouse_phene_distinct.phene_id, human_mouse_phene_distinct.sou
as âĂŸsource_id_pheneâĂŹ, human_mouse_distinct_disease.human_ortholog, âĂŸ
âĂŸ as mouse_rat_assoc from human_mouse_distinct_disease, human_mouse_phene_same
WHERE , human_mouse_distinct_disease.uniprot_id = human_mouse_phene_ distinct.uniprot_id AND human_mouse_distinct_disease.human_ortholog = human_mouse_phene_distinct.hu

7.2.12 Rat

mysql> CREATE TABLE rat_phene_disease_same SELECT DISTINCT human_rat_same_disease.mesh_o
human_rat_same_disease.source_id as âĂŸsource_id_diseaseâĂŹ, human_rat_phene_same.phene_id,
human_rat_phene_same.source_attr as âĂŸsource_id_pheneâĂŹ, human_rat_same_disease.human_ortho
âĂŸ âĂŸ as mouse_rat_assoc from human_rat_same_disease, human_rat_phene_same
WHERE , human_rat_same_disease.uniprot_id = human_rat_phene_same.uniprot_id
AND human_rat_same_disease.human_ortholog = human_rat_phene_same.human_ortholog;

mysql> CREATE TABLE rat_phene_disease_distinct SELECT DISTINCT human_rat_distinct_disease.mesh_omim_id, human_rat_distinct_disease.source_id as
âĂŸsource_id_diseaseâĂŹ, human_rat_phene_distinct.phene_id, human_rat_phene_distinct.source_attr
as âĂŸsource_id_pheneâĂŹ, human_rat_distinct_disease.human_ortholog, âĂŸ âĂŸ
as mouse_rat_assoc from human_rat_distinct_disease, human_rat_phene_distinct
WHERE , human_rat_distinct_disease.uniprot_id = human_rat_phene_distinct.uniprot_id
AND human_rat_distinct_disease.human_ortholog = human_rat_phene_distinct.human_ortholog;

mysql> CREATE TABLE rat_phene_disease_final SELECT DISTINCT human_rat_same_disease.mesh_
human_rat_same_disease.source_id as âĂŸsource_id_diseaseâĂŹ, human_rat_phene_same.phene_id,
human_rat_phene_same.source_attri as âĂŸsource_id_pheneâĂŹ, human_rat_same_disease.human_orth
âĂŸ âĂŸ as mouse_rat_assoc from human_rat_same_disease, human_rat_phene_same
WHERE , human_rat_same_disease.uniprot_id = human_rat_phene_same.uniprot_id
AND human_rat_same_disease.human_ortholog = human_rat_phene_same.human_ortholog;

Inserting values for columns that will distinguish matched and unmatched mouse and rat genes against human genes:

mysql> UPDATE rat_phene_disease_same SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE human_rat_phene_distinct SET mouse_rat_assoc =' indirect_assoc';
mysql> UPDATE human_mouse_phene_same SET mouse_rat_assoc =' direct_assoc ';
mysql> UPDATE human_mouse_phene_same SET mouse_rat_assoc =' indirect_assoc';

unify-

```
mysql> CREATE TABLE drug_gene_final (SELECT * from human_drug_disease_final)
UNION ALL (SELECT * from rat_drug_disease_same) UNION ALL (SELECT * from
rat_drug_disease_distinct) UNION ALL (SELECT * from mouse_drug_disease_distinct)
UNION ALL (SELECT * from mouse_drug_disease_same);
```

8