Using Mongo db for Chemical Structure Search

Abhik Seal

August 20, 2014

1 Using MongoDB Automated Scripts

MongoDB version 2.6 introduced some new aggregation features that may have better performance. Of particular interest is the $setIntersection operator, which is exactly what is needed to calculate the number of bits in common between two fingerprints. Installation of mongodb is simple and it can be installed using the steps given here `http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/`After installation create a database path and start the mongod daemon process. Mongod is the primary daemon process for the MongoDB system. It handles data requests, manages data format, and performs background management operations. In mongod daemon can be started at port 27017 and with –dbpath set to /home/abhik/data/db. –dbpath shows the database directory to . Before running mongod command make sure to create a /data/db directory and port 27017 is open . Pymongo is required which is the python driver for mongo db . If you have setuptools installed you should be able to do easy_install pymongo to install PyMongo. Otherwise you can download the project source and do python setup.py install to install.

```
# In one shell
#create a directory under /home/abhik
unichemvm:~ mkdir -p /data/db
unichemvm:~ mongod --dbpath /home/abhik/data/db
#In another shell
unichemvm:~ sudo mongo
```

This shows the mongo is up is running. To build a chemical database of fingerprints use the db_build.py program in the codes folder. db_build.py is a command line argument program where you can submit sdf,smi format, the pattern of fingerprint, its length fingerprint and fingerprint tag name, it generates fingerprints in mongodb . Currently fingerprints include morgan type, RDKFingerprint and rdkit maccs keys are supported. The working is show below in the code snippet.

```
chembl@unichemvm:~$ python db_build.py -h
usage: db_build.py [-h] --i I --db DB [--tag TAG] [--fpSize FPSIZE]
```

```
                   [--fpname FPNAME]
                   {morgan,rdkfp,rdmaccs} ...

Build a Database of fingerprints in MongoDB

optional arguments:
  -h, --help            show this help message and exit
  --i I                 input the structure file
  --db DB               Input Database Name
  --tag TAG             Give tag name. Must be present in structure file. Eg
                        'chembl_id'
  --fpSize FPSIZE       Length of the fingerprints
  --fpname FPNAME       Name of the fp Eg: mfp1,mfp2 .. etc

subcommands:
  valid subcommands

  {morgan,rdkfp,rdmaccs}
                        additional help
    morgan              Generate Morgan type fingerprints
    rdkfp               Generate RDKFingerprint
    rdmaccs             Generate MACCS Keys

# Parameters for rdkfp fingerprint
chembl@unichemvm:~$  python db_build.py rdkfp -h

usage: dbbuild.py morgan [-h] [--radius RADIUS]

optional arguments:
  -h, --help       show this help message and exit
  --radius RADIUS  Radius for morgan fingerprints
chembl@unichemvm:~$ python dbbuild.py rdkfp -h
usage: dbbuild.py rdkfp [-h] [--minPath MINPATH] [--maxPath MAXPATH]
                        [--nBitsPerHash NBITSPERHASH]

optional arguments:
  -h, --help            show this help message and exit
  --minPath MINPATH     minimum number of bonds to include in the subgraphs
                        Default 1.
  --maxPath MAXPATH     maximum number of bonds to include in the subgraphs
                        Default 7.
  --nBitsPerHash NBITSPERHASH
```

```
                      number of bits to set per path Defaults 2.

# To generate morgan type fingerprints with default parameters
chembl@unichemvm:~$ python db_build.py --i benzodiazepine.smi --tag chembl_id
                    --fpname mfp1 morgan

# To generate morgan type fingerprints with default parameters and fpSize 1024
chembl@unichemvm:~$ python db_build.py --i benzodiazepine.smi --tag chembl_id
                    --fpSize 1024 --fpname mfp2 morgan

# To generate morgan type fingerprints with fpSize 1024 and radius of 4
chembl@unichemvm:~$ python db_build.py --i benzodiazepine.smi --tag chembl_id
                    --fpSize 1024 --fpname mfp3 morgan --radius 4
```

Once the database is built with one fingerprint addfps.py can be called to generate more fingerprints over the molecular data. This way multiple fingerprints can be generated and stored. The code for addfps.py is stored in codes folder. The code is almost similar to db_buildy but here you need to specify the database for which you will generate the fingerprint. To execute it,

```
python addfps.py --db moltest --fpSize 1024 --fpname mfp2 morgan
```

To see everything working fine a sample smi file benzodiazepine.smi is given and is explained below.

```
# Generating maccs keys
chembl@unichemvm:~$ python db_build.py --i benzodiazepine.smi --db chemtest
                    --tag chembl_id --fpname mfp1 rdmaccs
fingerprints mfp1 done ...
Building Indices...

# Go to the mongo terminal and check moltest is created
> show dbs
admin       (empty)
chemtest    0.078GB
local       0.078GB
> use chemtest
# Shows the mfp_1 counts are generated.
> show collections
mfp1_counts
molecules
system.indexes

# Adding morgan fingerprints with length 1024 bits to the existing collection
```

```
chembl@unichemvm:~$ python addfps.py --db chemtest --fpSize 1024
                    --fpname mfp2 morgan
fingerprints mfp2 done ...
Building Indices...

# In mongo terminal shows mfp_2 counts added

> show collections
mfp1_counts
mfp2_counts
molecules
system.indexes
```

The query to the mongo database can be done using the monQuery.py script in the codes folder. In monQuery script is a command line program where user should give smiles string(smi) , database name to search(db) , the tag name of ids which is given for generation of original database(tag) , size of the fingerprint(fpsize) and its parameters for generation of similar type of fingerprint and fingperprint name (fpname) as given in the search database (ex: mfp1). Below shows the script how it is executed.

```
#-h open the help file
chembl@unichemvm:~$ python monQuery.py -h
usage: monQuery.py [-h] --db DB --smi SMI [--fpSize FPSIZE] --fpname FPNAME
                   [--t T] [--tag TAG]
                   {morgan,rdkfp,rdmaccs} ...

Search MongoDB database

optional arguments:
  -h, --help            show this help message and exit
  --db DB               Input Database Name
  --smi SMI             Enter the smiles string
  --fpSize FPSIZE       Length of the fingerprints
  --fpname FPNAME       Name of the fp ex:mfp1,mfp2 .. etc
  --t T                 Similarity threshold
  --tag TAG             tag name in the original database Ex:chembl_id

subcommands:
  valid subcommands

  {morgan,rdkfp,rdmaccs}
                        additional help
    morgan              Generate Morgan type fingerprints
    rdkfp               Generate RDKFingerprint
    rdmaccs             Generate MACCS Keys


# Searching the database
chembl@unichemvm:~$ python monQuery.py --db chemtest --smi
'CC1=NN=C2N1C3=C(C=C(C=C3)Cl)C(=NC2)C4=CC=CC=C4' --fpSize 512
--fpname mfp1 --t 0.8 --tag chembl_id morgan --radius 2
fingerprints mfp1 done ...
start Aggregate ..
response done ..
Hits:  6 Time :  0.0532460212708
1.0: 450819
0.952380952381: 19002642
1.0: 2118
0.952380952381: 178274
0.813953488372: 12562523
0.818181818182: 21489341
```