# BME Chemical Toolkit 1.0

Created by Patrick Chirdon 2020

with the support of Dr.Douglas Goetz, Dr.Sumit Sharma, Dr.Colin Kruse, and Athena Ghiasi

at Ohio University

Requires Python

Spark

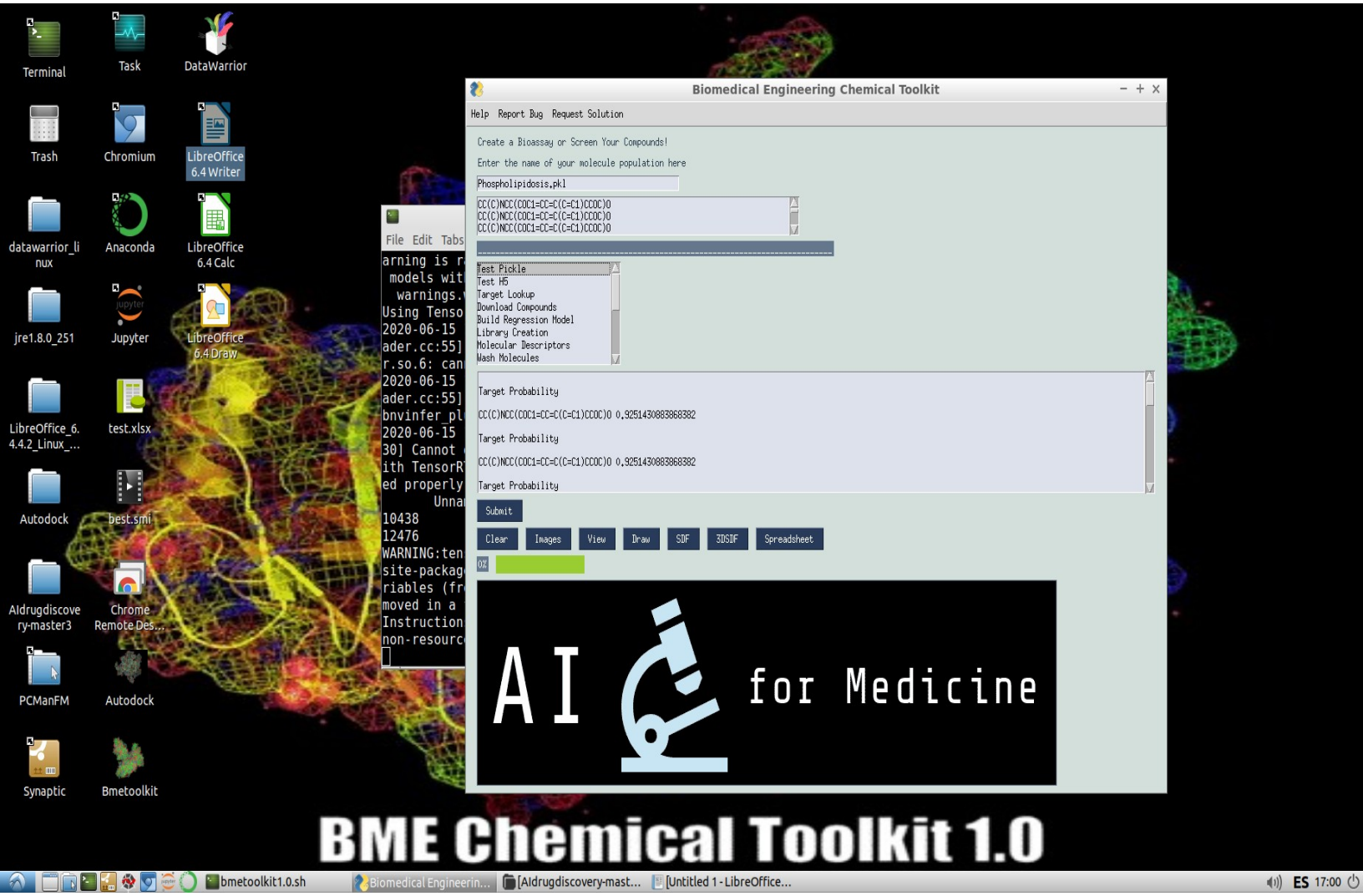
Java

Anaconda

Virtualbox

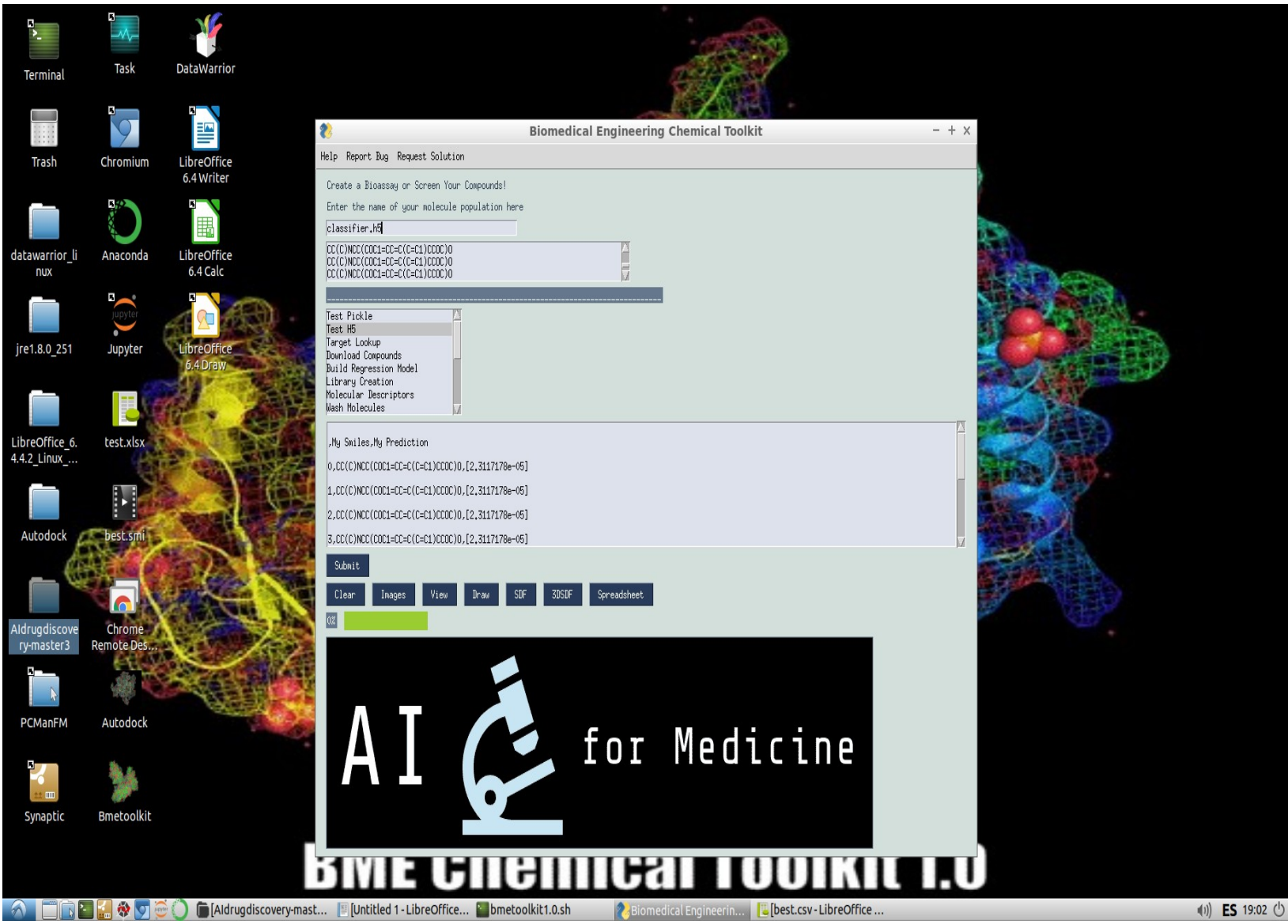
## 1.Test Pickle

Input the pickle name into the first box.  The file must end in .pkl.  Then input a list of smiles into the second box, click test pickle and then click submit.  The output is a target probability.


Terminal
Task
DataWarrior
Trash
Chromium
LibreOffice 6.4 Writer
datawarrior_linux
Anaconda
LibreOffice 6.4 Calc
jre1.8.0_251
Jupyter
LibreOffice 6.4 Draw
LibreOffice_6.4.4.2_Linux_...
test.xlsx
Autodock
best.smi
AIdrugdiscovery-master3
Chrome Remote Des...
PCManFM
Autodock
Synaptic
Bmetoolkit
Biomedical Engineering Chemical Toolkit
Help Report Bug Request Solution
Create a Bioassay or Screen Your Compounds!
Enter the name of your molecule population here
Phospholipidosis.pkl
Test Pickle
Test H5
Target Lookup
Download Compounds
Build Regression Model
Library Creation
Molecular Descriptors
Wash Molecules
Target Probability
Submit
Clear
Images
View
Draw
SDF
3DSDF
Spreadsheet
AI for Medicine
BME Chemical Toolkit 1.0
bmetoolkit1.0.sh
Biomedical Engineerin...
[AIdrugdiscovery-mast...
[Untitled 1 - LibreOffice...
ES 17:00

## 2. Test H5

Type the name of the .h5 in the first box. Type a list of SMILES into the second box and click submit. The output is a probability.




## 3. Target Lookup

Type in the name of the protein target into the first box.

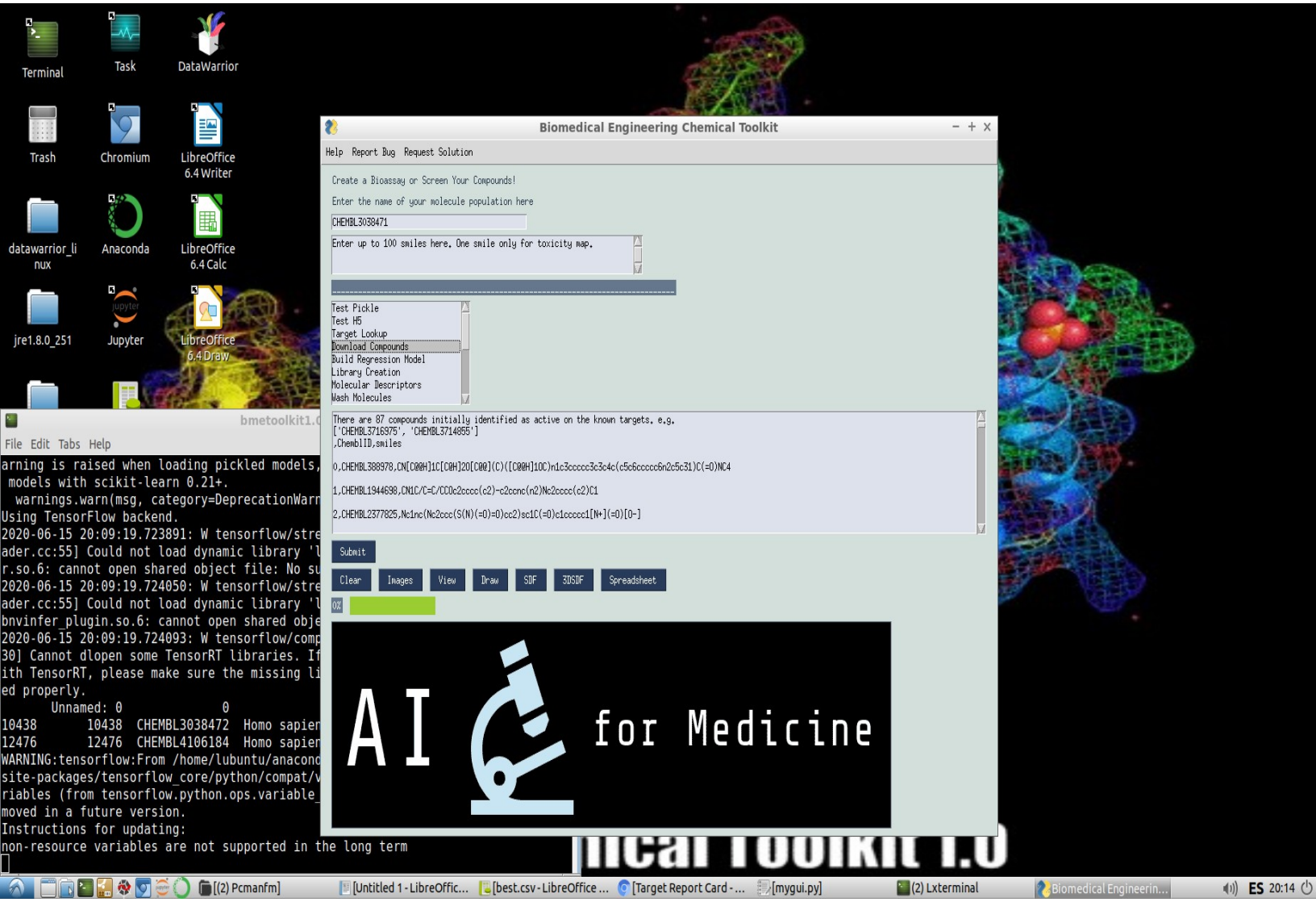
The output will be a CHEMBL ID. ie

C[C@@H](C(=O)N[C@@H](CO)C(=O)N1CCC[C@H]1C(=O)N[C@@H](CCCN=C(N)N)C(=O)N[C@@H](CCCCN)C(=O)O)NC(=O)[C@H](CC2=CN=CN2)NC(=O)[C@H](CC3=CN=CN3)N

| Unnamed: 0 | 0 | 1 | 2 | | |
|---|---|---|---|---|---|
| 10435 | 10435 | CHEMBL3038469 | Homo sapiens | CDK2/Cyclin A | |
| 10436 | 10436 | CHEMBL3038470 | Homo sapiens | CDK2/Cyclin A1 | |
| 10483 | 10483 | CHEMBL3038517 | Homo sapiens | CDK2/CDK4 | |
| 12444 | 12444 | CHEMBL4106152 | Homo sapiens | CDK2/Cyclin O | |
| 12445 | 12445 | CHEMBL4106153 | Homo sapiens | CDK2/Cyclin A | |
| 12477 | 12477 | CHEMBL4106185 | Mus musculus | CDK2/CDK4 | |
| 12478 | 12478 | CHEMBL4106186 | Mus musculus | CDK2/CDK9 | |

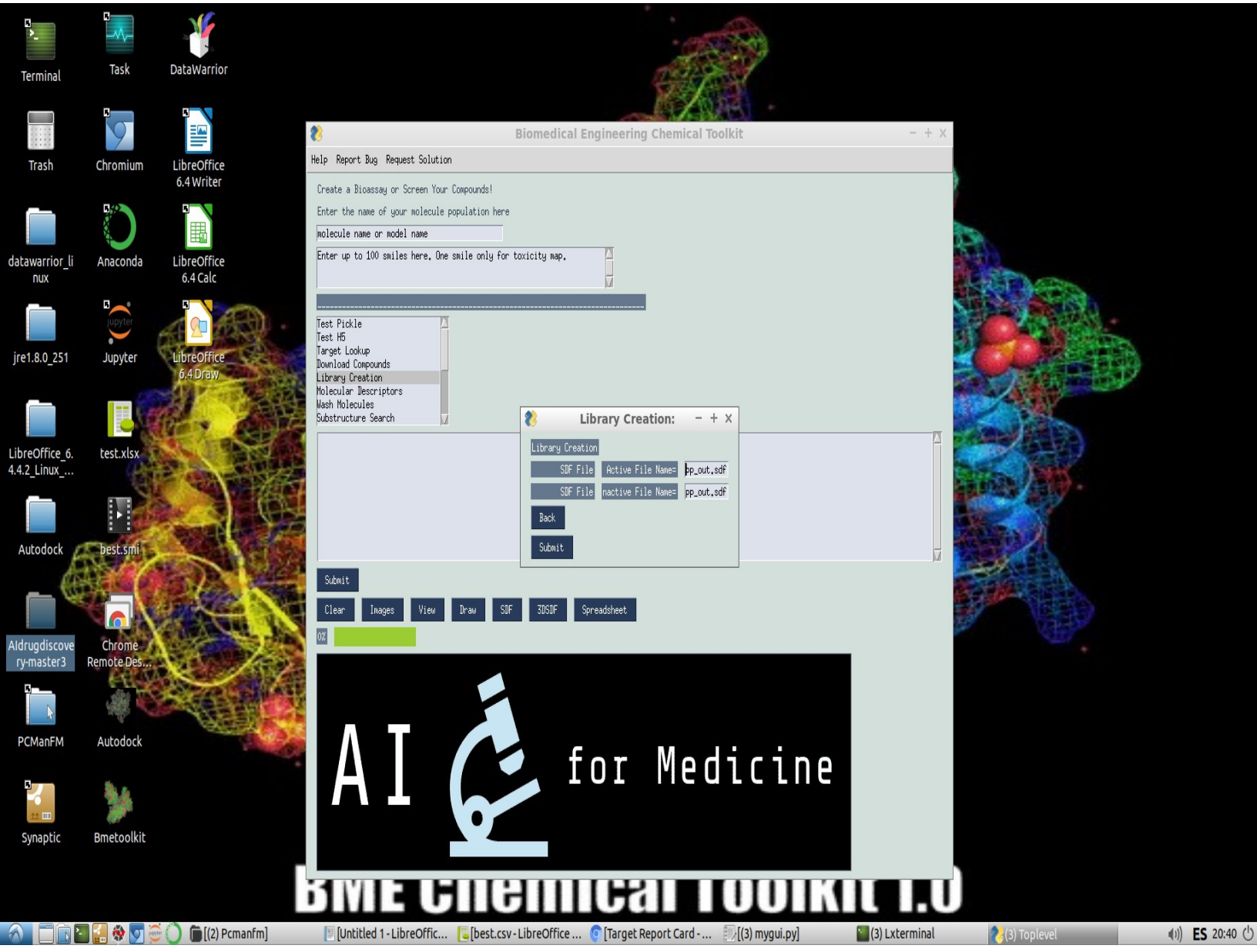

# 4. Download Compounds

Input the CHEMBL ID into the first box and click submit.  The output will be a list of smiles and their corresponding CHEMBLID for the compound



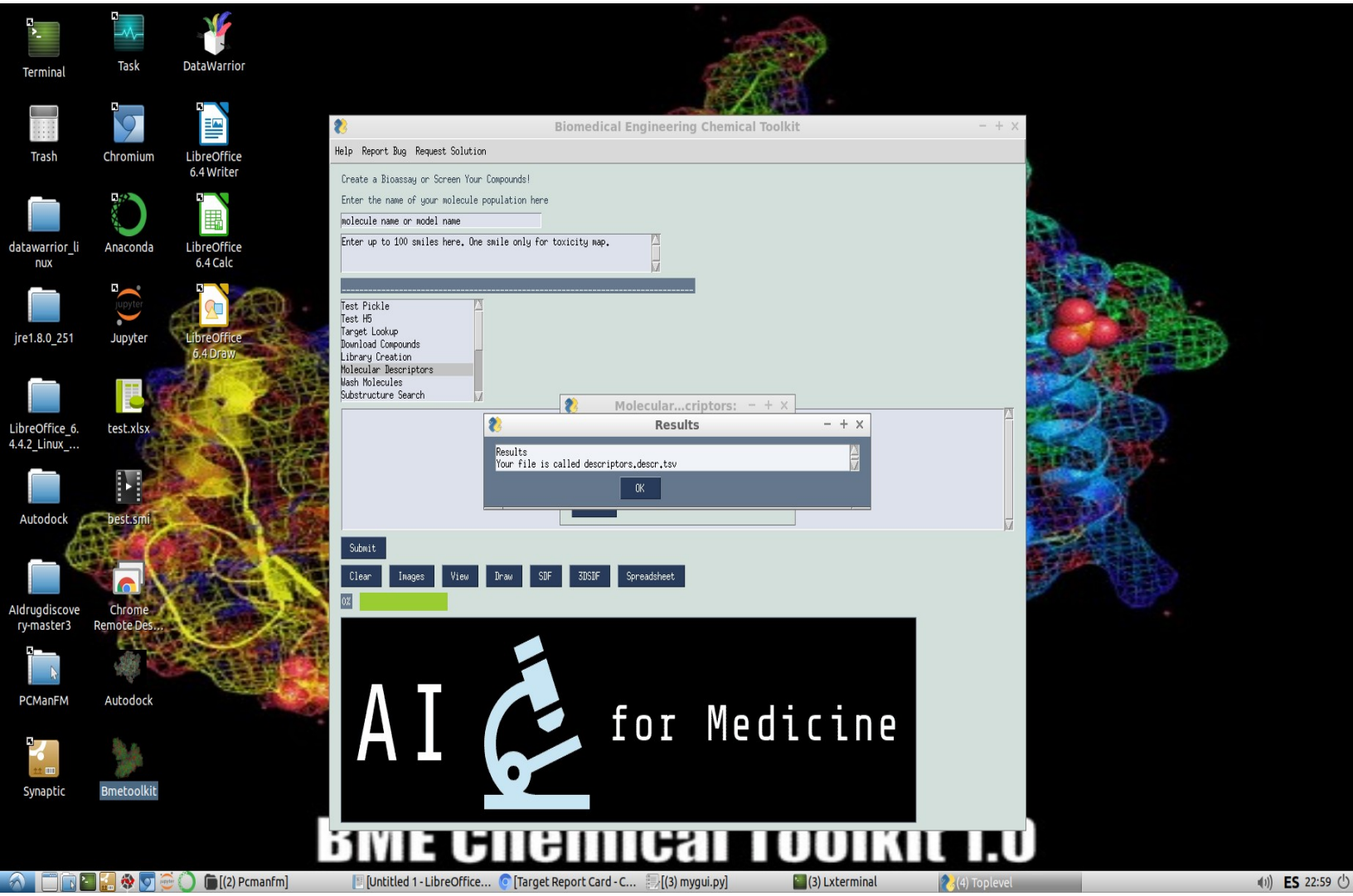
# 5. Library Creation

Create a library of compounds labeled active, 1, or inactive, 0 as a SDF file.  Input an active file name containing compounds that are known to bind a target and input an inactive file name containing compounds that do not bind (could also be just random compounds)



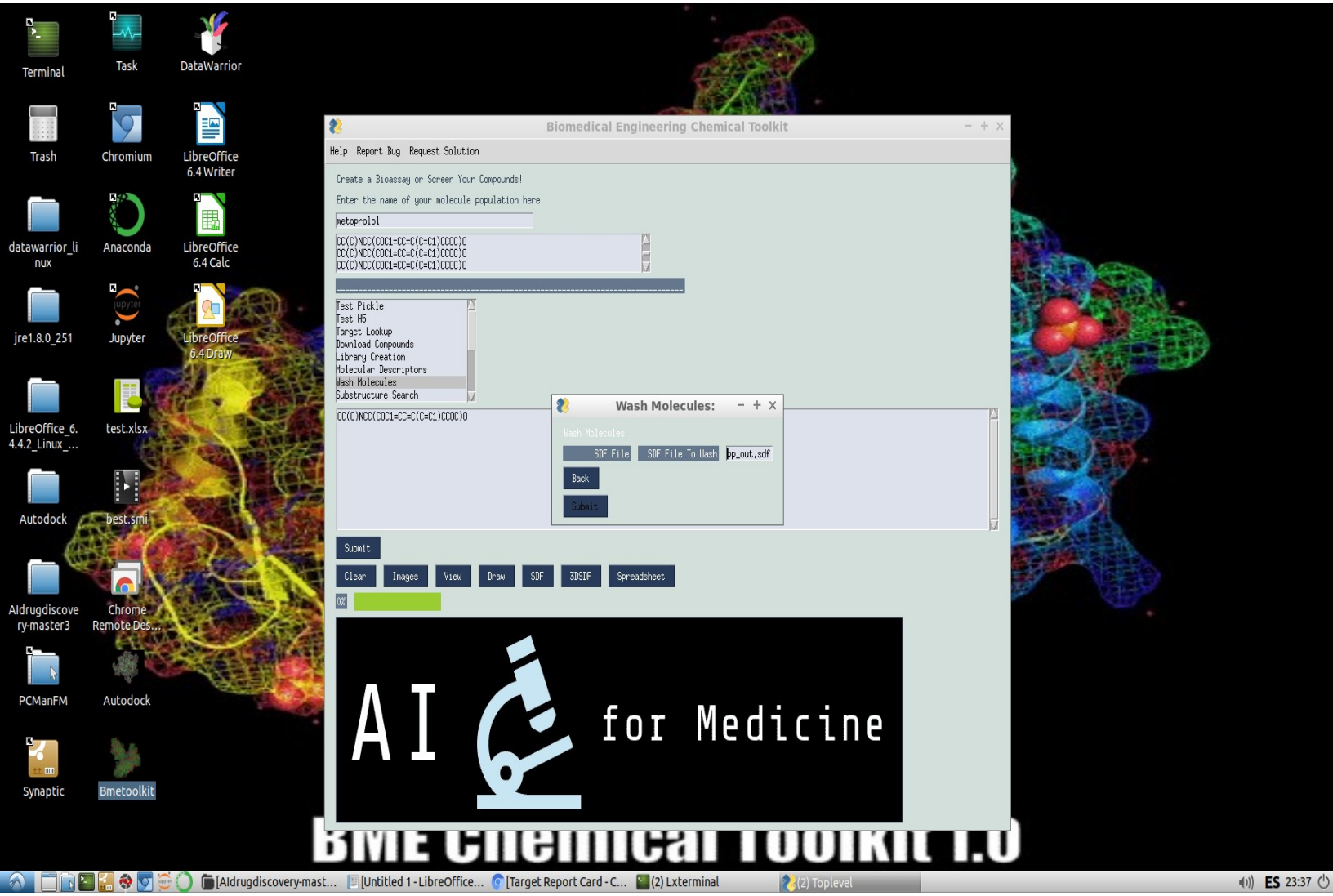
# 6. Molecular Descriptors

Input a SDF file. The output file will be called descriptors.descr.tsv. The output will be an excel file with chemical properties.



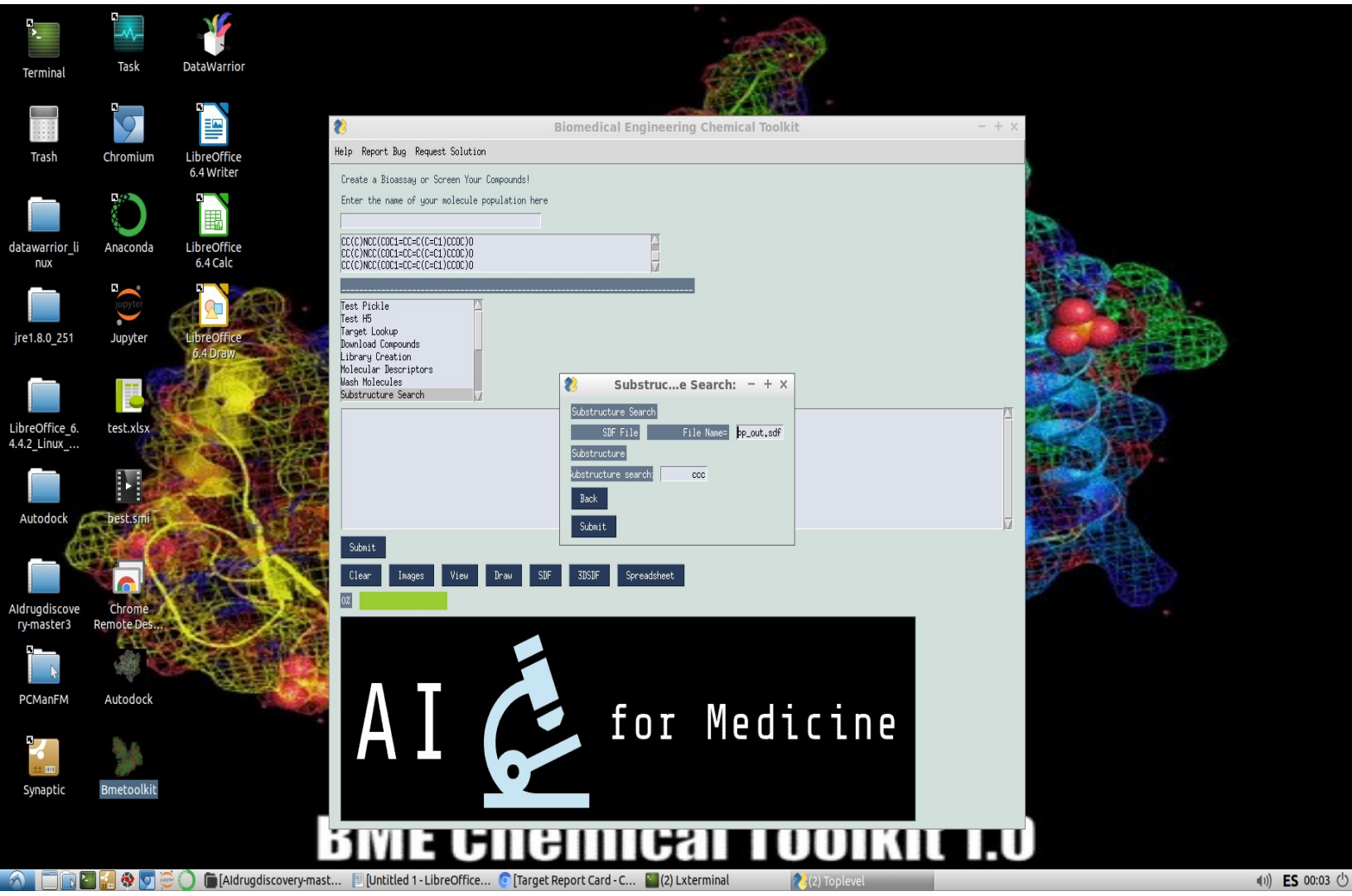
# 7. Wash Molecules

Input an SDF molecule.  The program will remove some metal atoms and ions, remove small fragments, adjust formal charges, and rebalance hydrogen counts.



# 8. Substructure Search

For a substructure search input a SDF file and search for a substructure.  The results output will be a list of molecules containing the substructures.



# 9. Pharmacophore

Input two smiles and click pharmacophore. The output will be the pharmacophore similarity. 1 means that the molecules are the same. The scale is from 0 to 1.

# 10. PAINS

PAN Assay Interference

The program will tell you what functional groups for each compound were responsible for a positive PAINFUL test result. The program also tells you the fraction of SP3 hybridized carbons. Compounds with scores > .47 are more selective binders. Note that double bonds reduce the fraction of sp3 hybridization, as they make the compound more flat. See Escape from flatland: increasing saturation as an approach to improving clinical success. Pains are defined as follows:

Doveston R, et al. A Unified Lead-oriented Synthesis of over Fifty Molecular Scaffolds. Org Biomol Chem 13 (2014) 859D65. doi:10.1039/C4OB02287D

Jadhav A, et al, Quantitative Analyses of Aggregation, Autofluorescence, and Reactivity Artifacts in a Screen for Inhibitors of Thiol Protease. J Med Chem 53 (2009) 37D51. doi:10.1021/jm901070c

# 11.Chembl Prediction

Chembl Target Prediction

Multiclass Classifier

Number of unique targets 560

Ion channel 5

kinase 96

nuclear receptor 21

GPCR 180

Others 258

accuracy .87

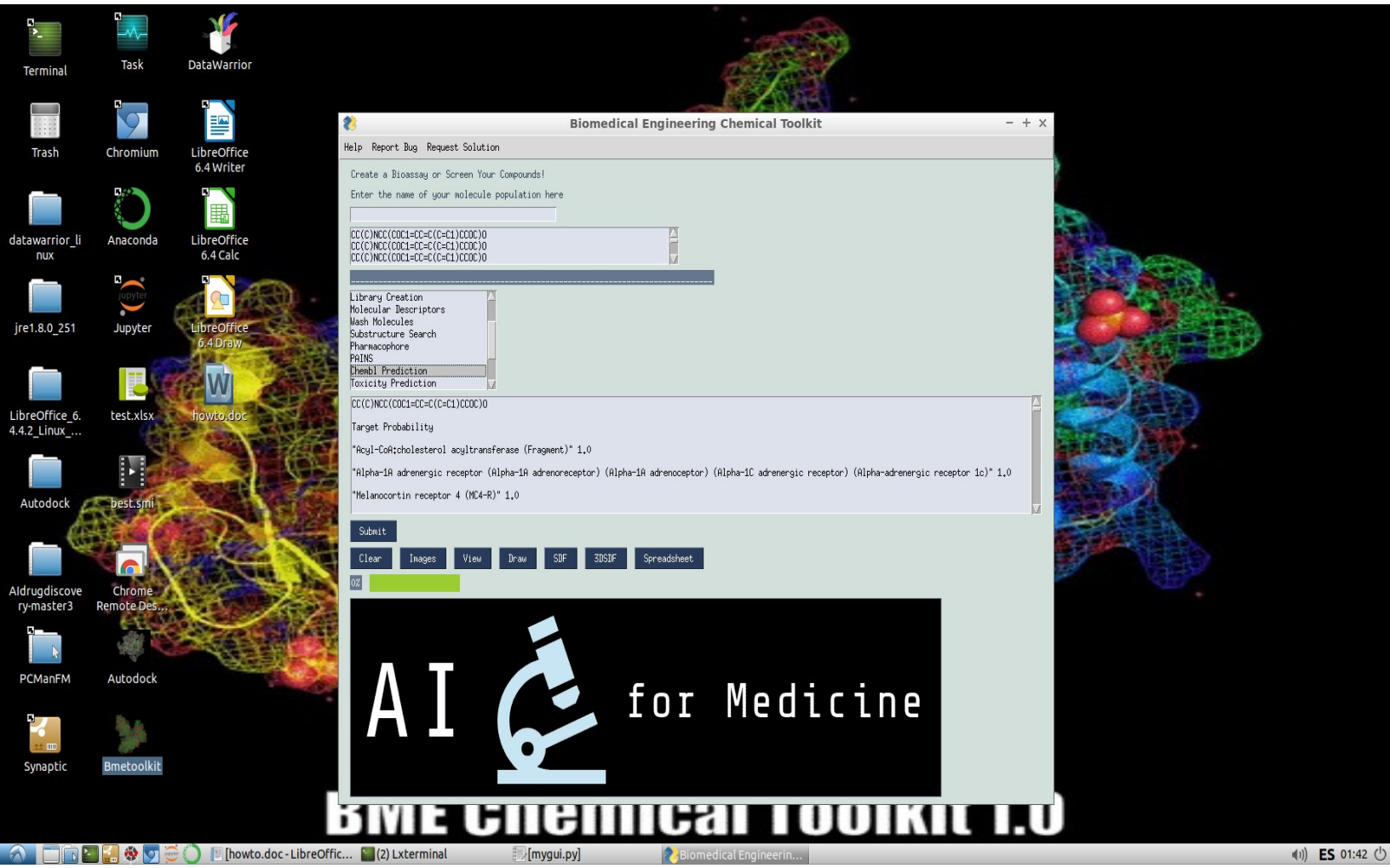
auc .92

sensitivity .76

specificity .92

precision .82

225/225 100% correct mechanism on independent test set. Note-- 1 is considered positive and zero is negative for a given target.



# 12. Target Prediction

Support Vector Machine

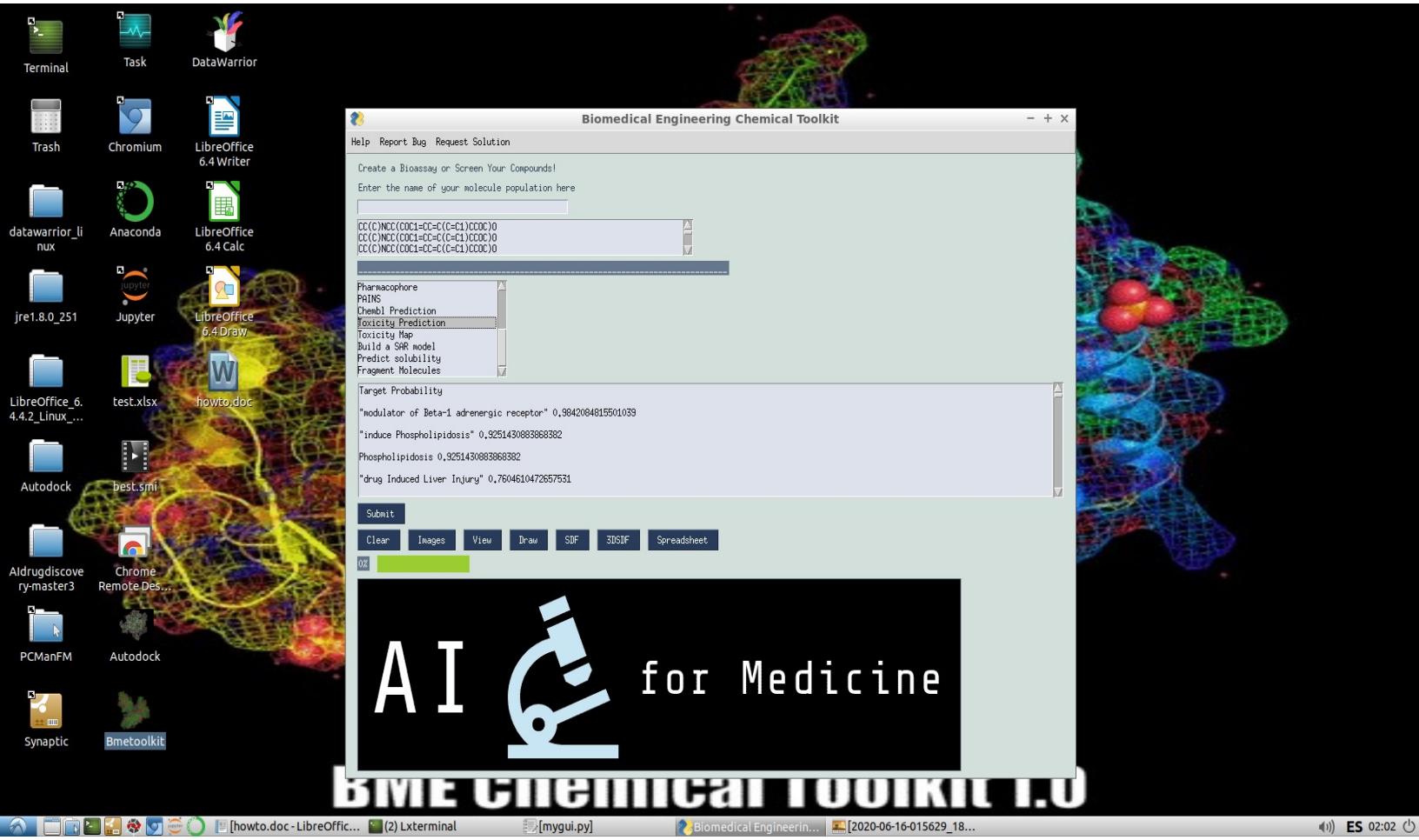
46 models. Metrics for test data:

accuracy 97.59 +/- 2.41

sensitivity 91.9 +/- 8.1

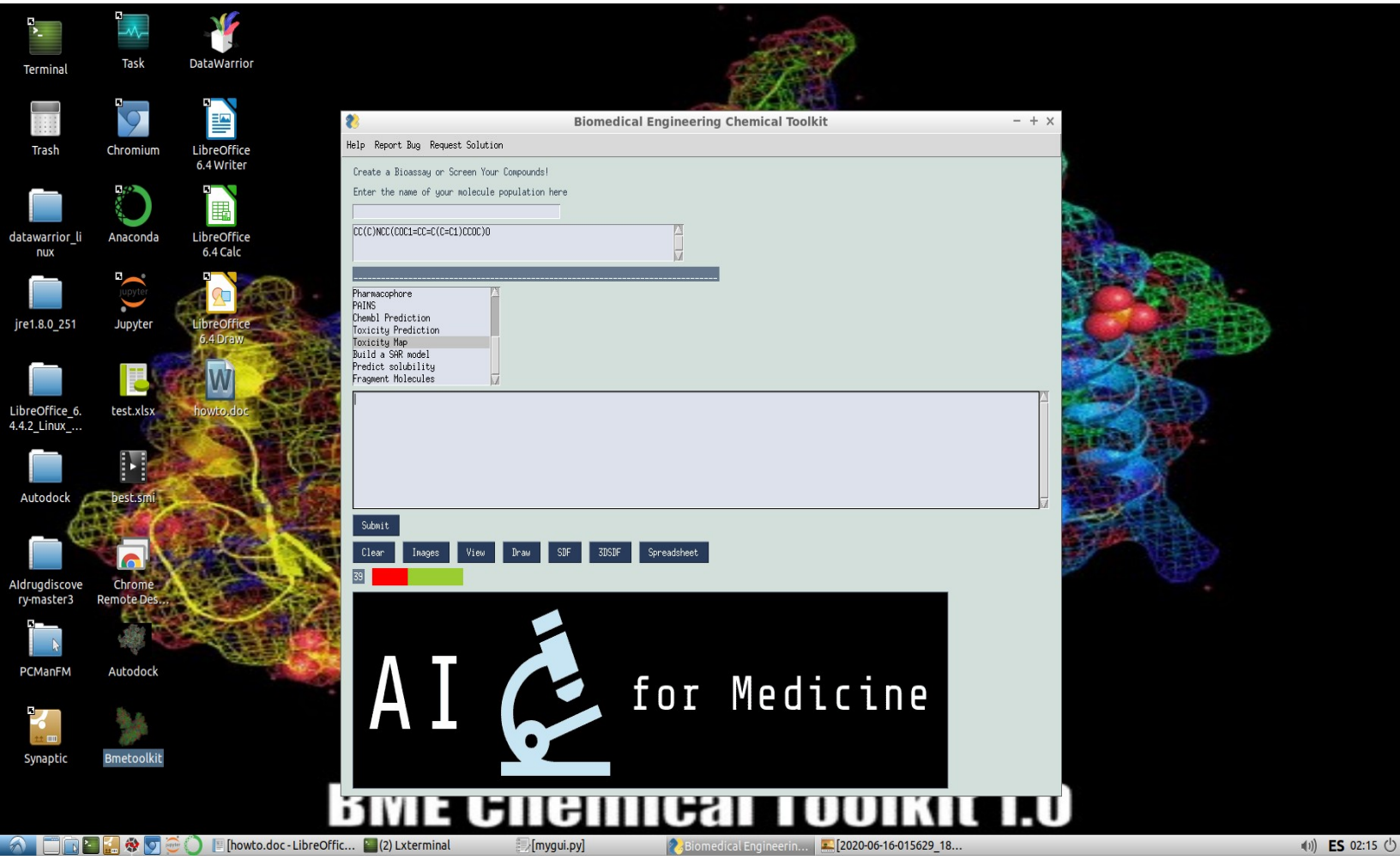
specificity 98.6 +/- 1.4

215/247 87% correct mechanism on independent test set.



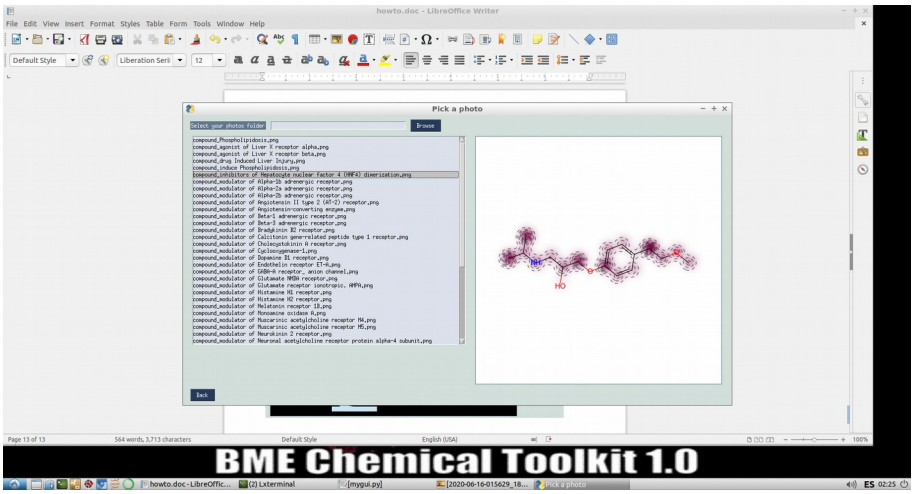
# 13. Toxicity Map

Input One SMILE. The output will be 46 toxicity maps for the 46 models available from toxicity prediction.

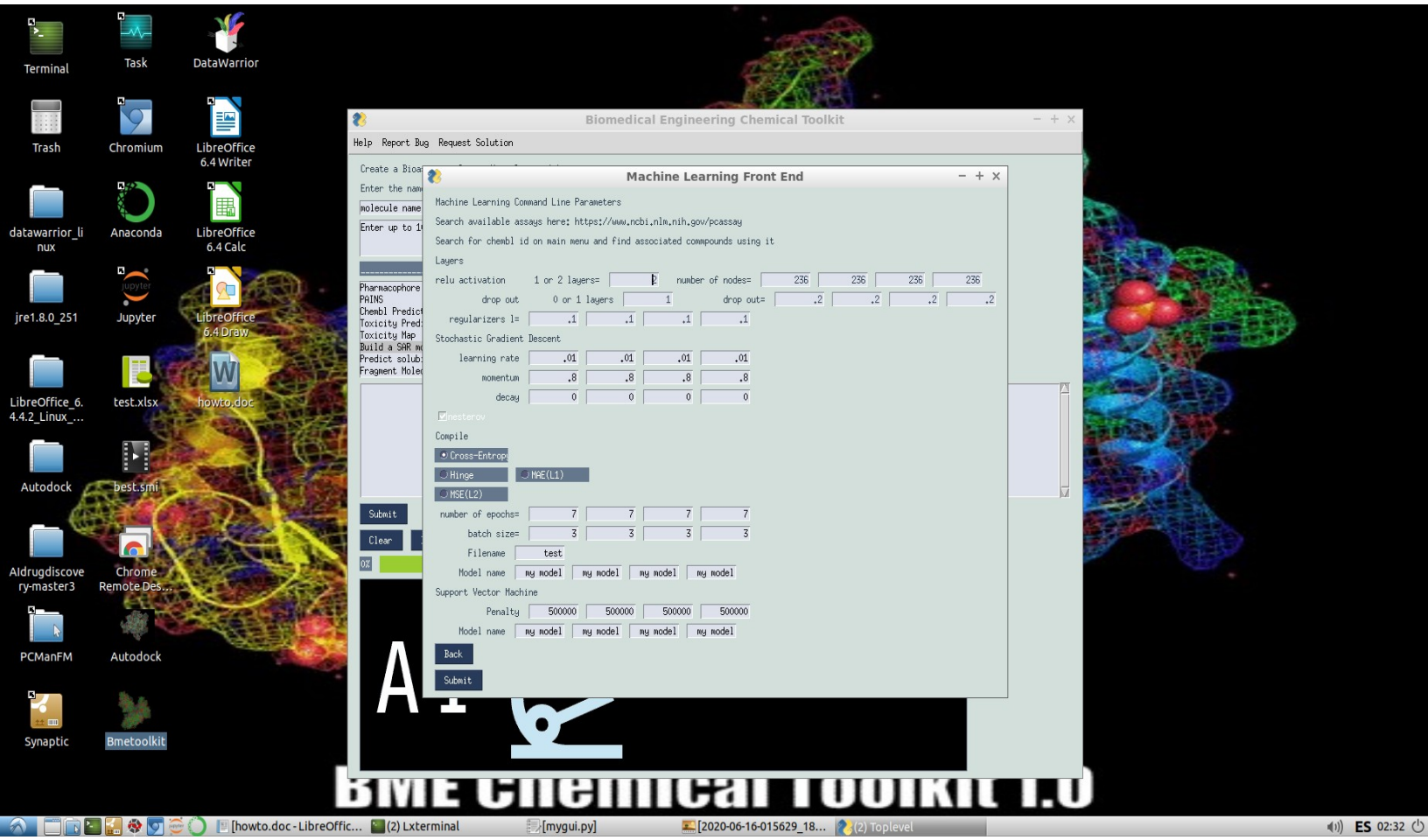


.

Click Images to see the output.  Pink marks inactive parts of the molecule, green marks active parts



# 14. Build a SAR Model



Toggle the parameters.  There are 4 models for the SVM and 4 models for the neural network

cross entropy- default loss function for binary classification problems. Summarizes the average difference between the actual and predicted probability.

hinge- alternative to cross entropy binary classification developed with SVM models used with support vector machine models

mse-default loss to use for regression problems. calculated as the average of the squared differences between the predicted and actual values

mae-for regression problems.  used in cases where there are outliers. average of the absolute difference between actual and predicted values

# 15. Solubility

Solubility

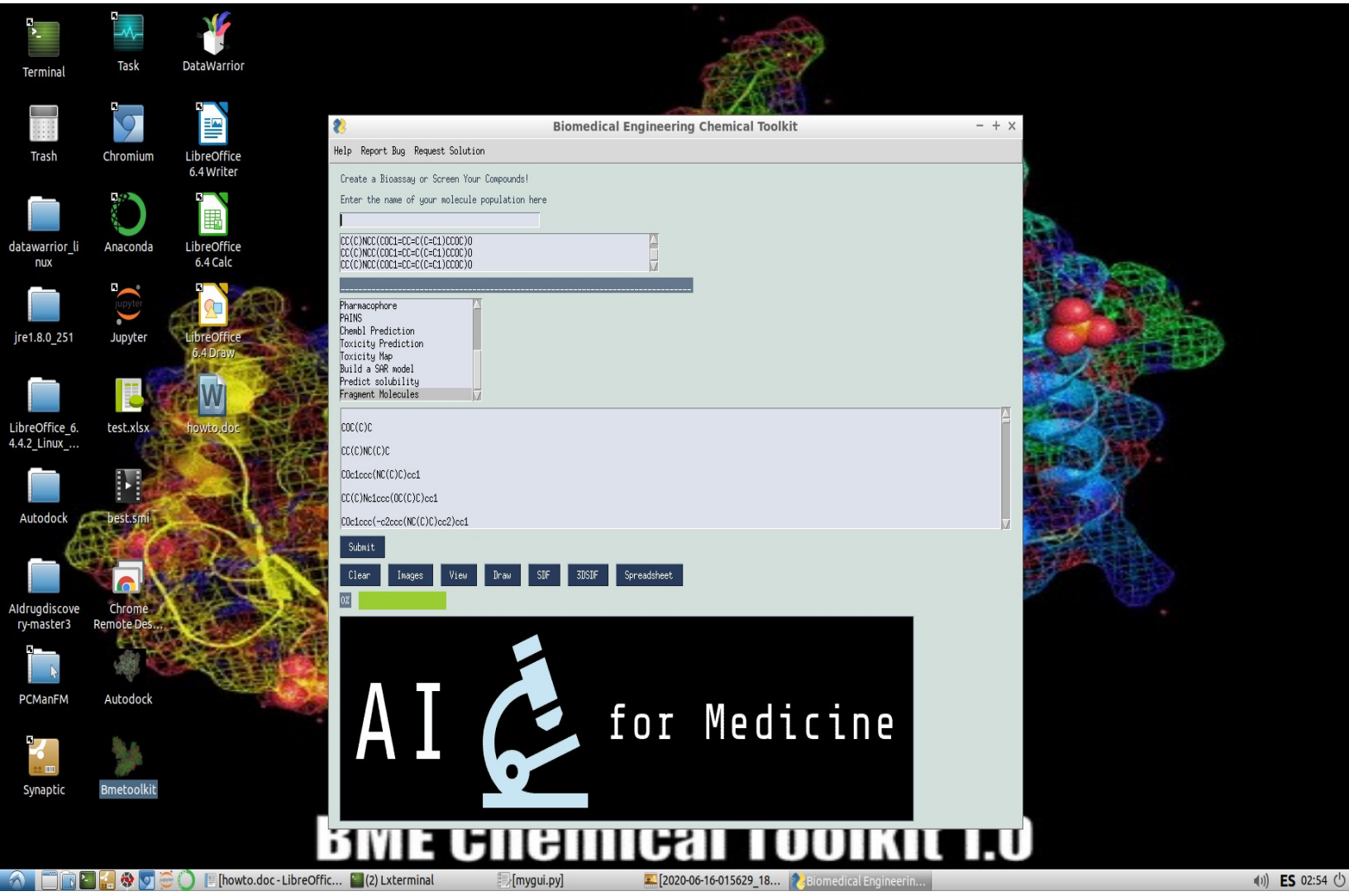
Predicts log S. Log S greater than -4 is soluble.

Root mean square error of 1.27 on a scale from -4 to 4.

linear regression

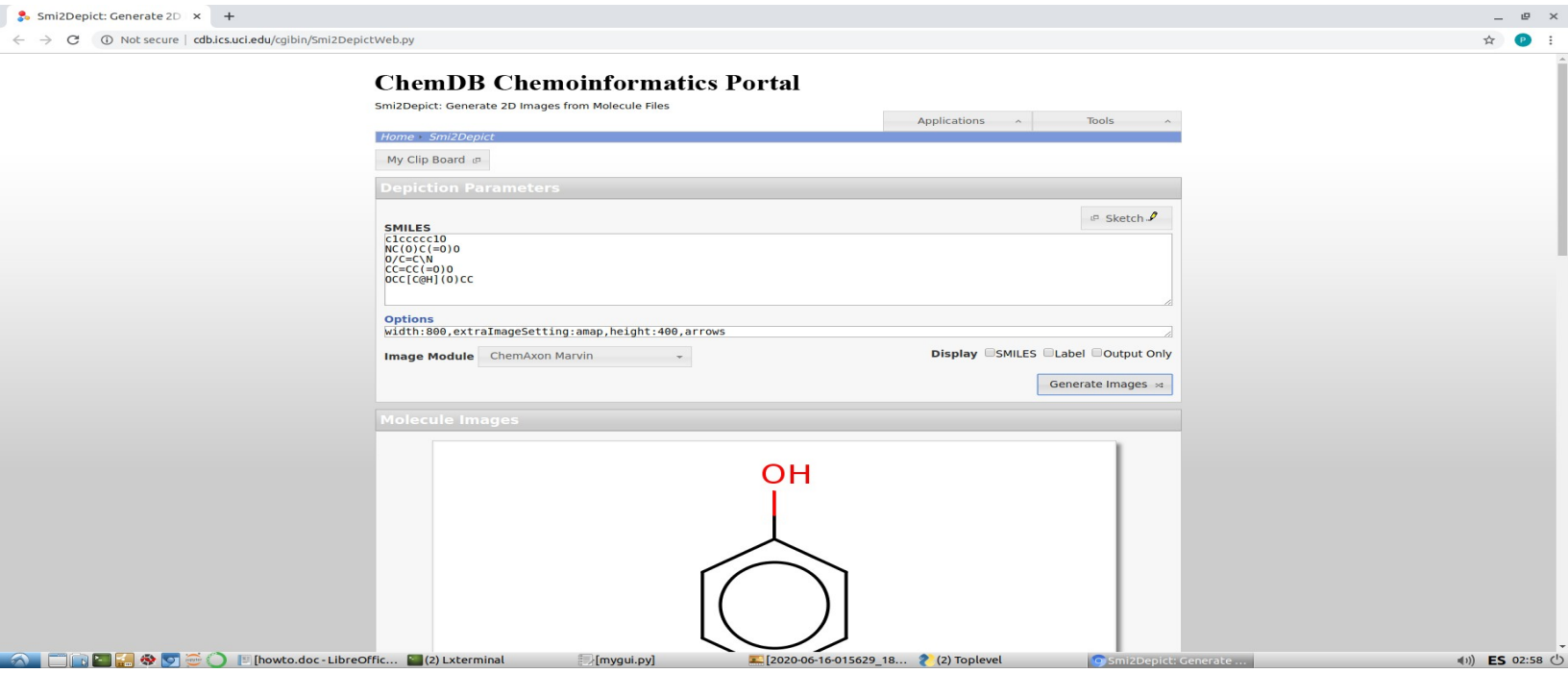
# 16. Fragmenter

Input a list of smiles. These will be recombined into new combinations. When you take the lowest energy ligands from a docking program and recombine these there may be some compounds that bind with lower energy than the original.

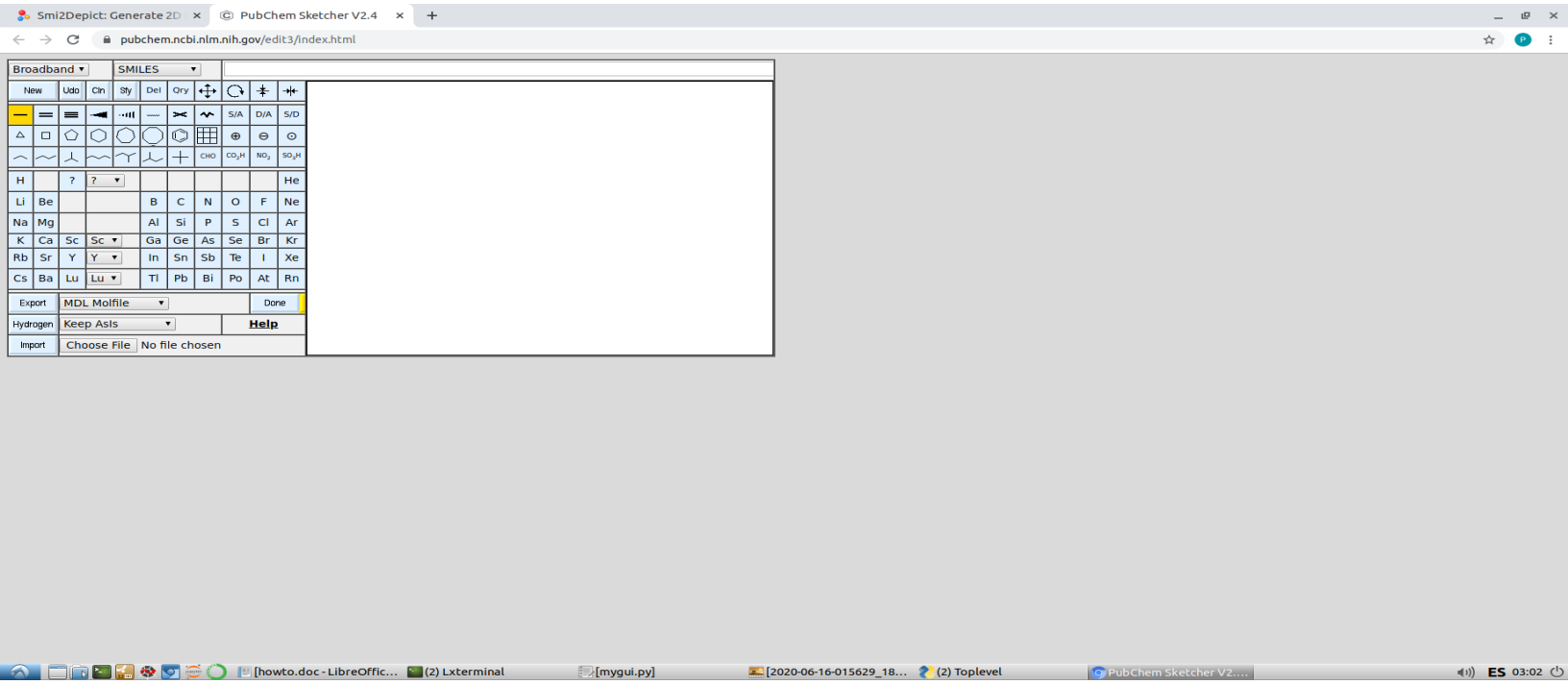



# 17. View

Input a list of SMILES and click Generate to view

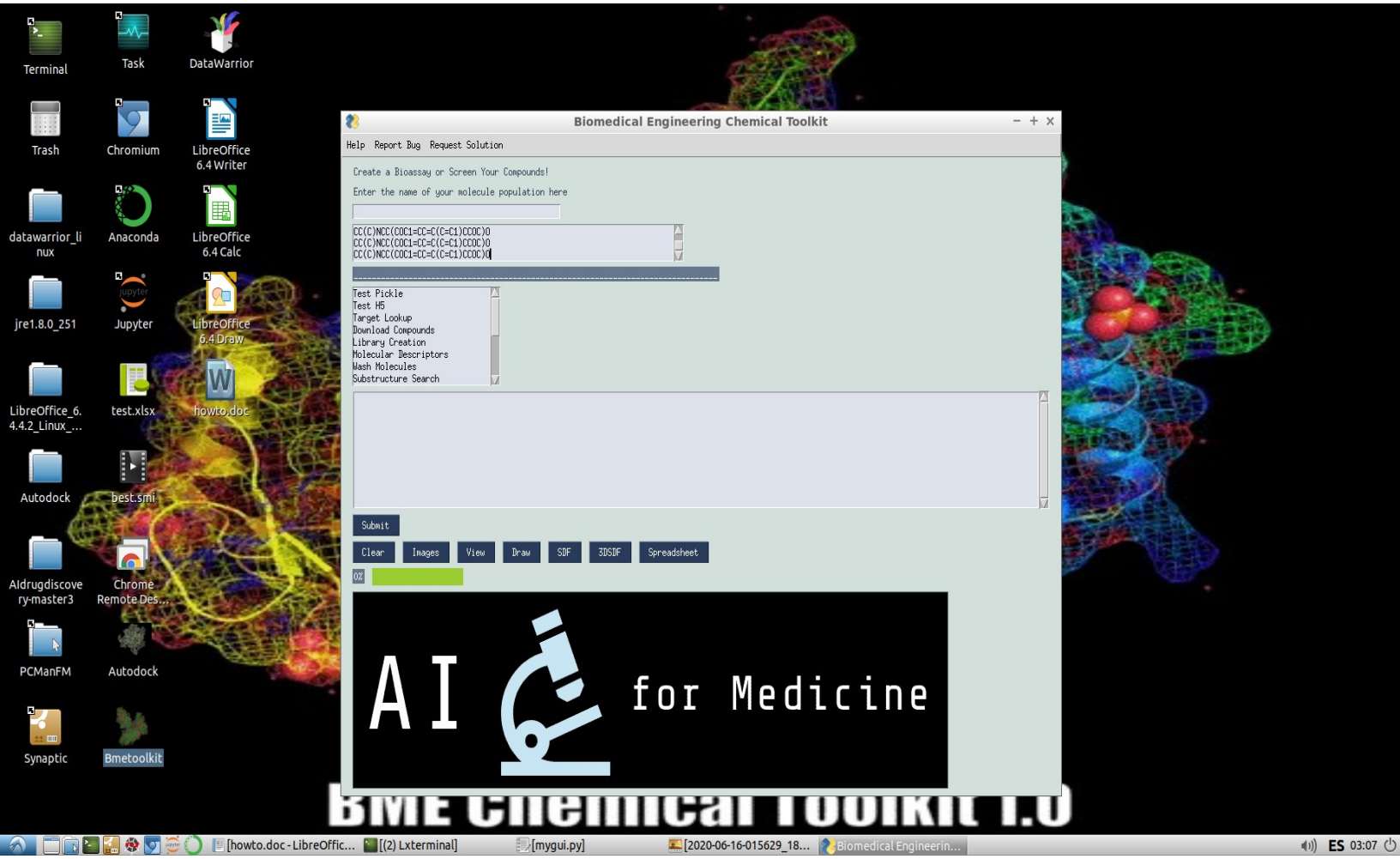



# 18. Draw



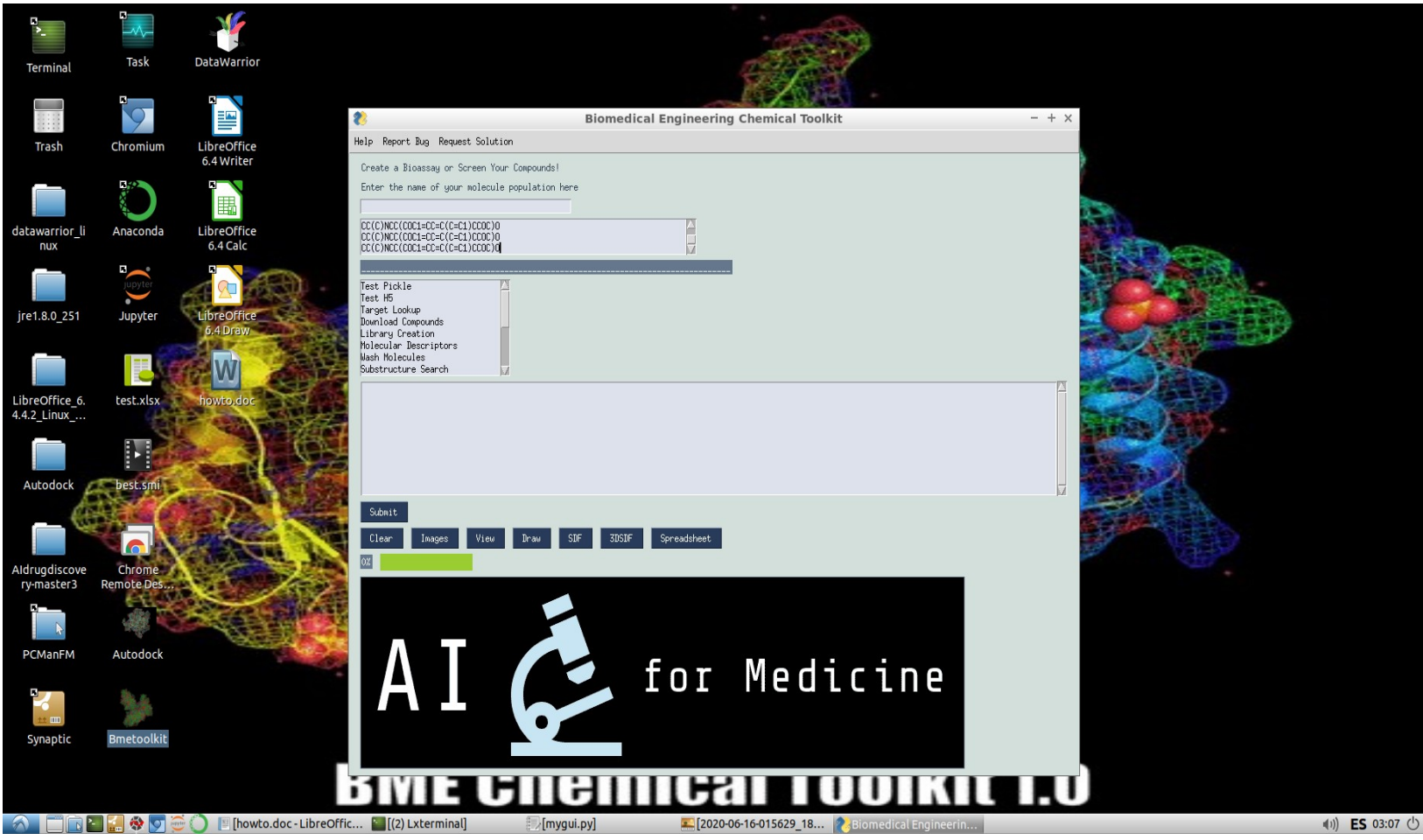

# 19. SDF

Input a list of molecules. Click SDF. The output will be a 2D SDF file called output.sdf

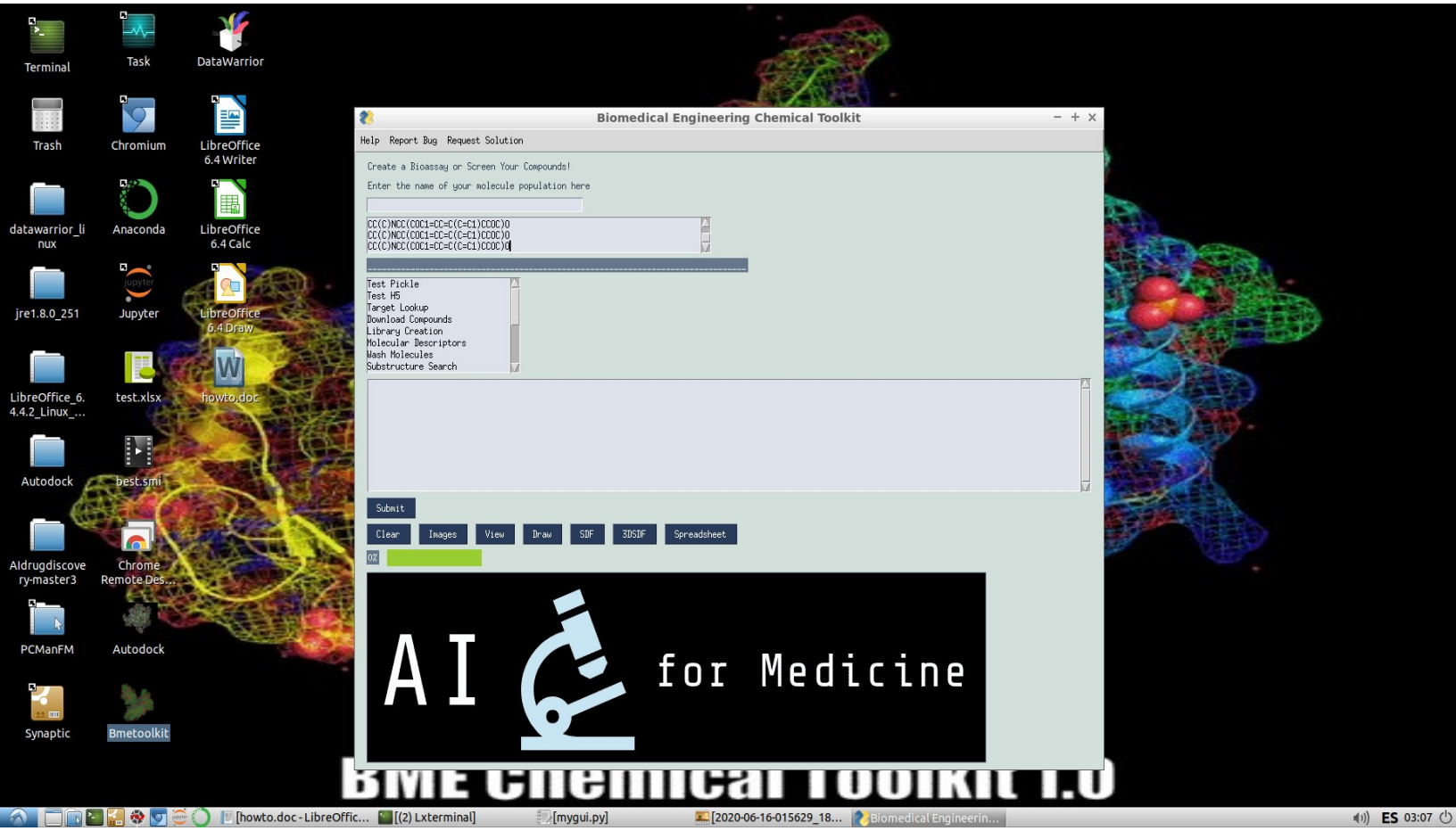

Terminal
Task
DataWarrior
Trash
Chromium
LibreOffice 6.4 Writer
datawarrior_linux
Anaconda
LibreOffice 6.4 Calc
jre1.8.0_251
Jupyter
LibreOffice 6.4 Draw
LibreOffice_6.4.4.2_Linux_...
test.xlsx
howto.doc
Autodock
best.smi
AIdrugdiscovery-master3
Chrome Remote Des...
PCManFM
Autodock
Synaptic
Bmetoolkit
Biomedical Engineering Chemical Toolkit
Help Report Bug Request Solution
Create a Bioassay or Screen Your Compounds!
Enter the name of your molecule population here
CC(C)NCC(COC1=CC=C(C=C1)CCOC)O
CC(C)NCC(COC1=CC=C(C=C1)CCOC)O
CC(C)NCC(COC1=CC=C(C=C1)CCOC)O
Test Pickle
Test H5
Target Lookup
Download Compounds
Library Creation
Molecular Descriptors
Wash Molecules
Substructure Search
Submit
Clear
Images
View
Draw
SDF
3DSDF
Spreadsheet
0%
AI
for Medicine
BME Chemical Toolkit 1.0
[howto.doc - LibreOffic...
[(2) Lxterminal]
[mygui.py]
[2020-06-16-015629_18...
Biomedical Engineerin...
ES
03:07

# 20. 3D SDF

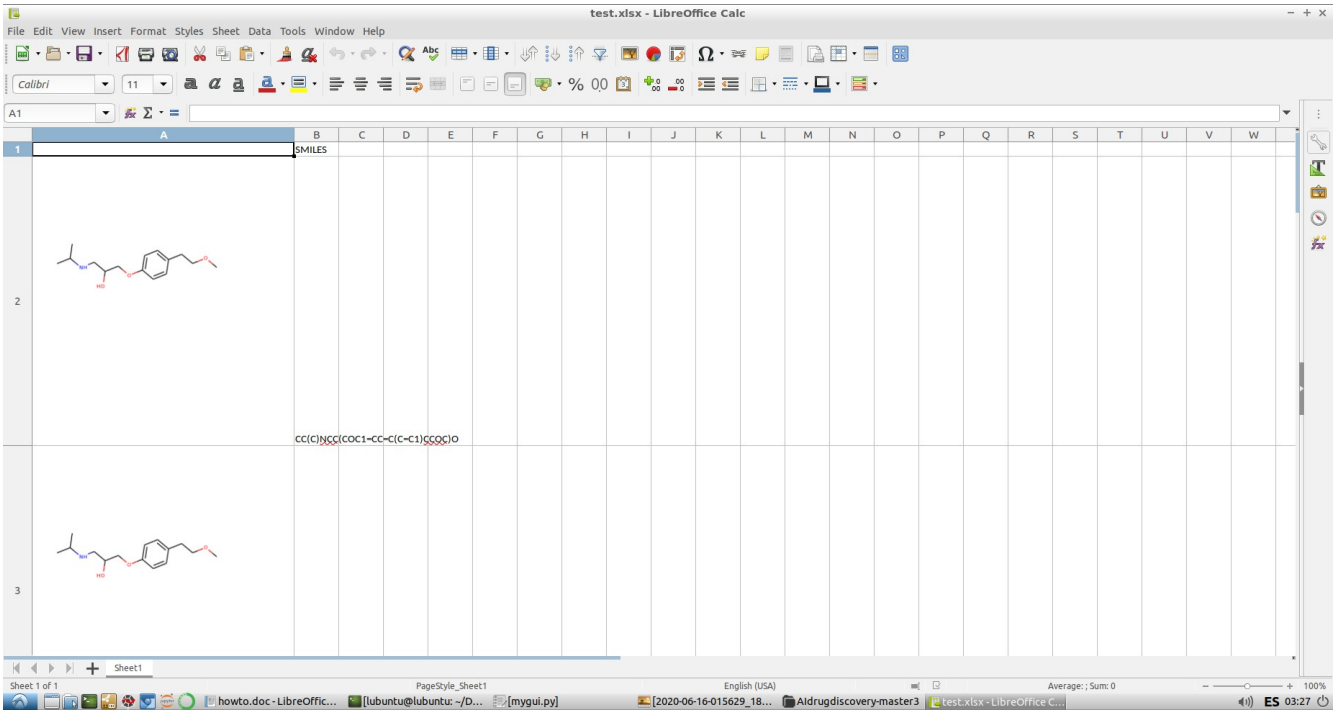
Input a list of SMILES and the output will be a 3D SDF file called output.sdf



# 21. Spreadsheet



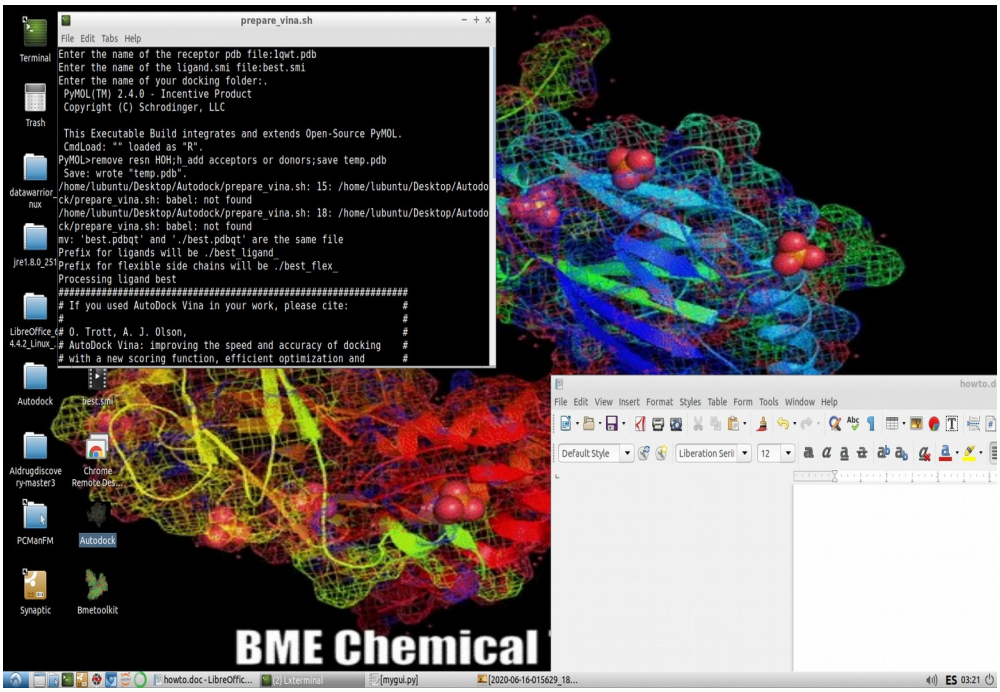
Input a list of SMILES.  The output will be a spreadsheet called test.xlsx with pictures and SMILES next to them.




# 22. Prepare Autodock VINA

1. Enter the name of the receptor pdb
2. Enter the name of the ligand.smi file
3. Enter the docking folder as

The output will be a separate directory for each receptor ligand complex with a log file.




There will be an output file called results table.csv in the autodock folder that will contain the docking energy in kJ for each of the compounds.  The lower the energy the better the binder.

Additional Pre Installed Packages

Python

pip install for new python packages

Anaconda

Conda install for new packages

Data Warrior for viewing and creating libraries.

Java

Spark

Open Office

Chrome Remote Desktop for sharing the virtualbox with others using the chromium web browser

Google Colab for sharing a jupyter notebook with others.  Also gives you the ability to use the GPU

……The password for lubuntu when installing new packages is ridiculous

Type sudo apt install package name

cd spark-3.0.0-preview2-bin-hadoop2.7

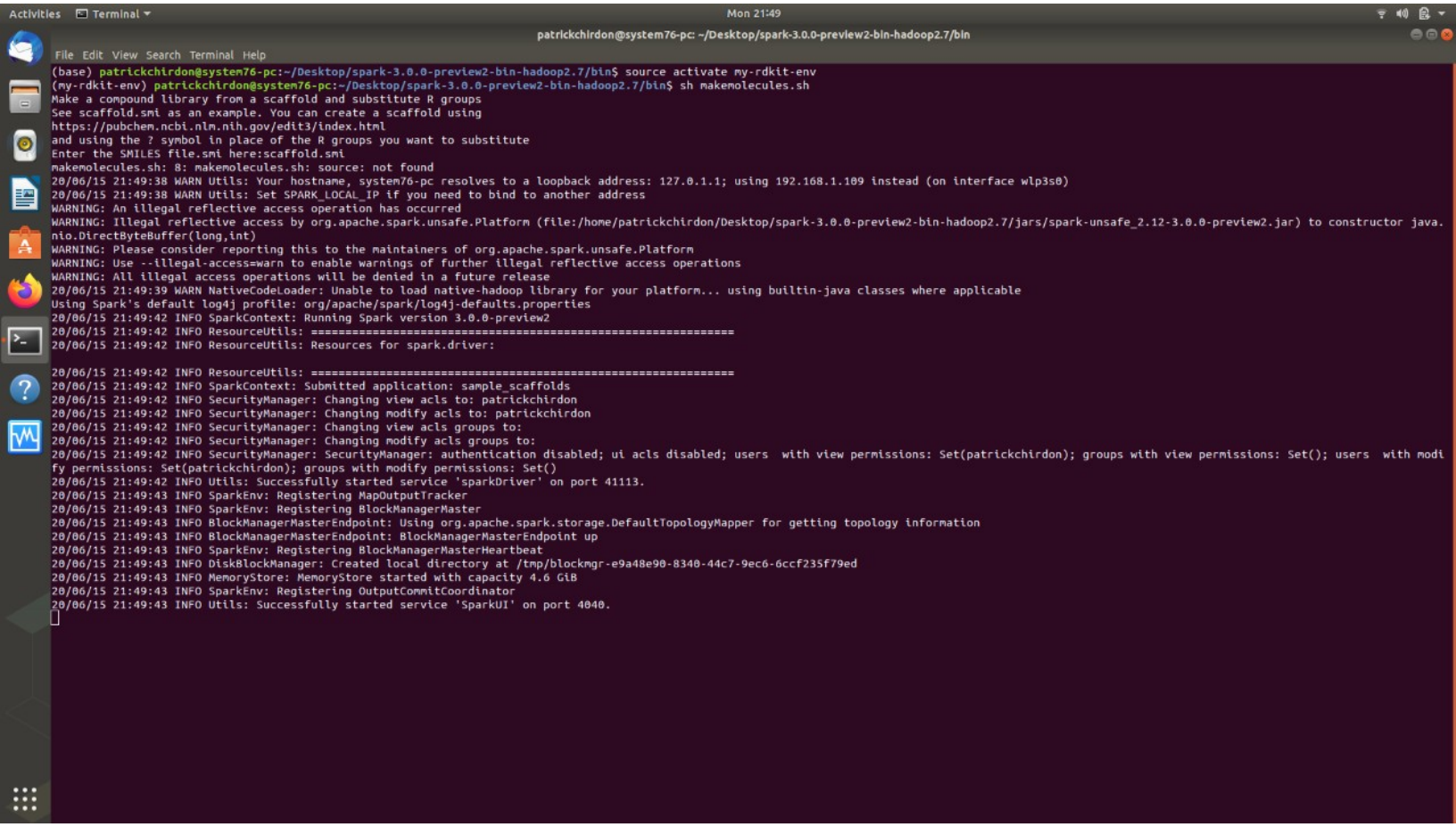
source activate my-rdkit-env

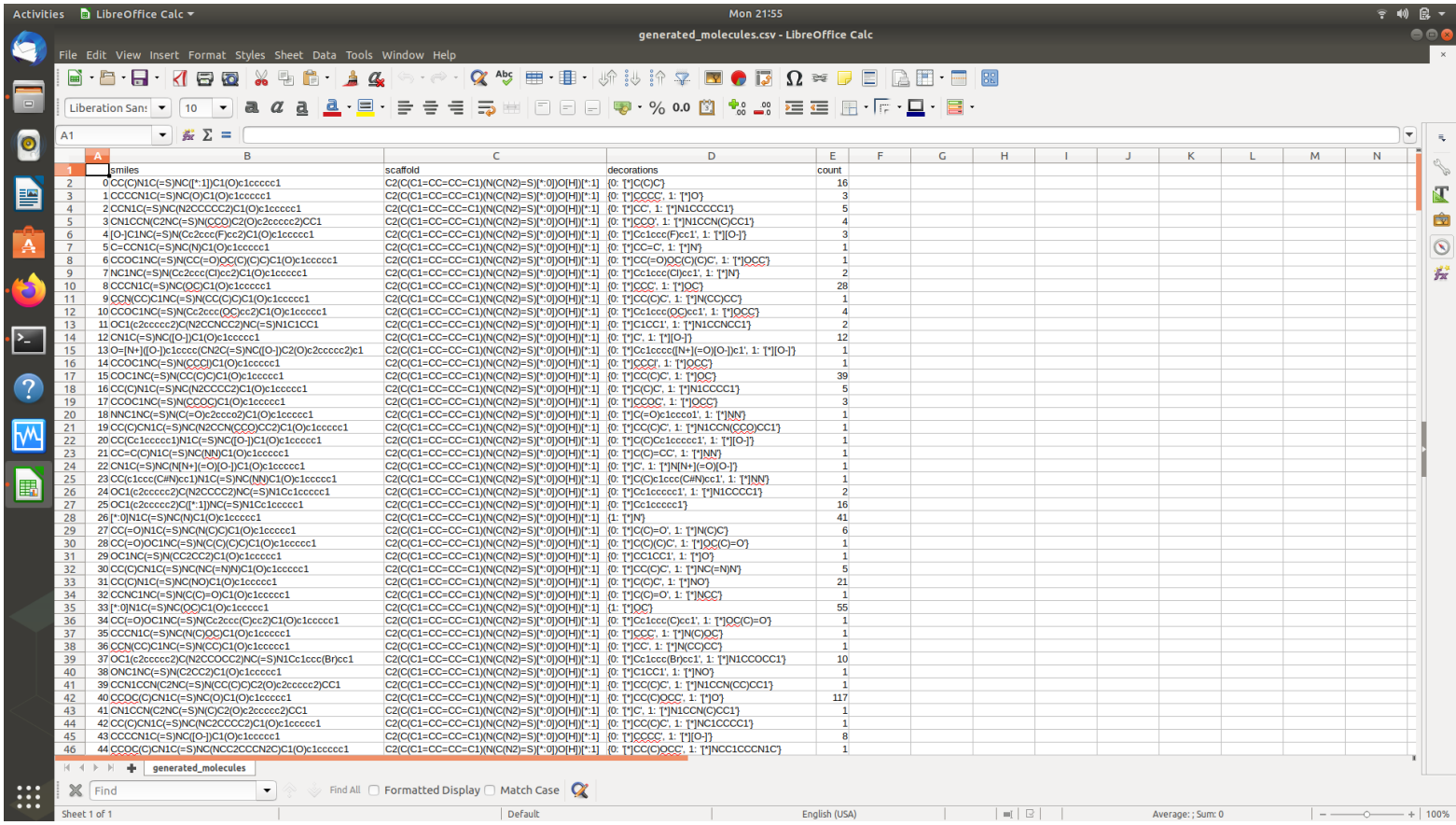
cd bin

sh makemolecules.sh

Follow the instructions.  You can edit scaffold.smi
using https://pubchem.ncbi.nlm.nih.gov/edit3/index.html

Use the ? symbol to mark R groups

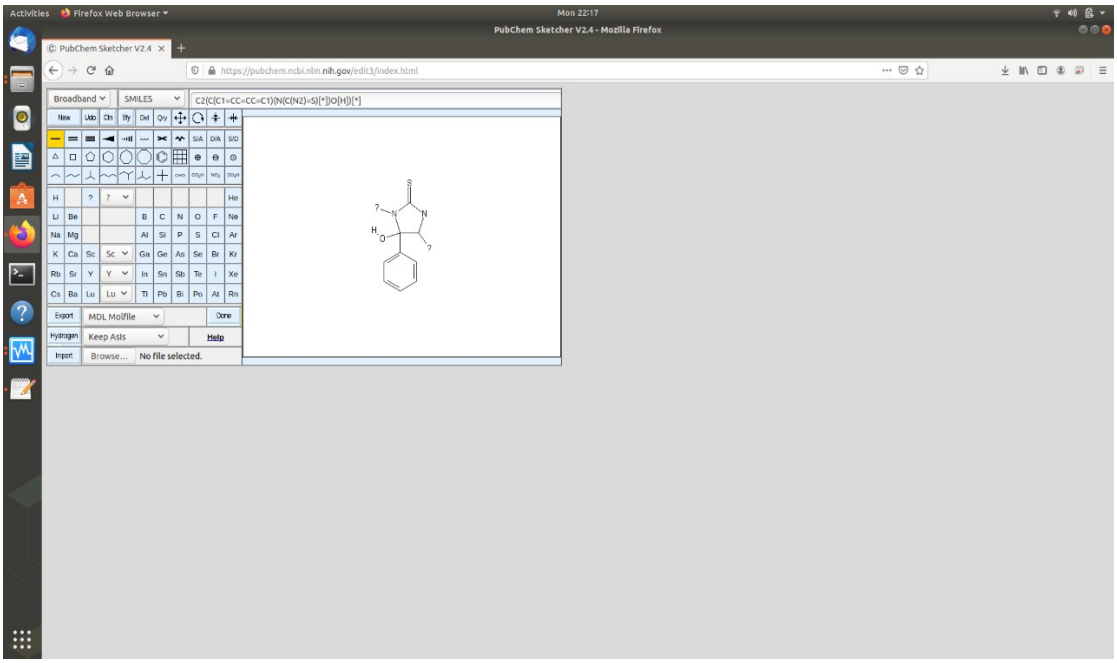

Activities
Terminal
Mon 21:49
patrickchirdon@system76-pc: ~/Desktop/spark-3.0.0-preview2-bin-hadoop2.7/bin
File Edit View Search Terminal Help
(base) patrickchirdon@system76-pc:~/Desktop/spark-3.0.0-preview2-bin-hadoop2.7/bin$ source activate my-rdkit-env
(my-rdkit-env) patrickchirdon@system76-pc:~/Desktop/spark-3.0.0-preview2-bin-hadoop2.7/bin$ sh makemolecules.sh
Make a compound library from a scaffold and substitute R groups
See scaffold.smi as an example. You can create a scaffold using
https://pubchem.ncbi.nlm.nih.gov/edit3/index.html
and using the ? symbol in place of the R groups you want to substitute
Enter the SMILES file.smi here:scaffold.smi
makemolecules.sh: 8: makemolecules.sh: source: not found
20/06/15 21:49:38 WARN Utils: Your hostname, system76-pc resolves to a loopback address: 127.0.1.1; using 192.168.1.109 instead (on interface wlp3s0)
20/06/15 21:49:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/patrickchirdon/Desktop/spark-3.0.0-preview2-bin-hadoop2.7/jars/spark-unsafe_2.12-3.0.0-preview2.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/06/15 21:49:39 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
20/06/15 21:49:42 INFO SparkContext: Running Spark version 3.0.0-preview2
20/06/15 21:49:42 INFO ResourceUtils: ==============================================================
20/06/15 21:49:42 INFO ResourceUtils: Resources for spark.driver:
20/06/15 21:49:42 INFO ResourceUtils: ==============================================================
20/06/15 21:49:42 INFO SparkContext: Submitted application: sample_scaffolds
20/06/15 21:49:42 INFO SecurityManager: Changing view acls to: patrickchirdon
20/06/15 21:49:42 INFO SecurityManager: Changing modify acls to: patrickchirdon
20/06/15 21:49:42 INFO SecurityManager: Changing view acls groups to:
20/06/15 21:49:42 INFO SecurityManager: Changing modify acls groups to:
20/06/15 21:49:42 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(patrickchirdon); groups with view permissions: Set(); users with modify permissions: Set(patrickchirdon); groups with modify permissions: Set()
20/06/15 21:49:42 INFO Utils: Successfully started service 'sparkDriver' on port 41113.
20/06/15 21:49:43 INFO SparkEnv: Registering MapOutputTracker
20/06/15 21:49:43 INFO SparkEnv: Registering BlockManagerMaster
20/06/15 21:49:43 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
20/06/15 21:49:43 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
20/06/15 21:49:43 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
20/06/15 21:49:43 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-e9a48e90-8340-44c7-9ec6-6ccf235f79ed
20/06/15 21:49:43 INFO MemoryStore: MemoryStore started with capacity 4.6 GiB
20/06/15 21:49:43 INFO SparkEnv: Registering OutputCommitCoordinator
20/06/15 21:49:43 INFO Utils: Successfully started service 'SparkUI' on port 4040.




| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | smiles | scaffold | decorations | count |
| 2 | 0 | CC(C)N1C(=S)NC([*:1])C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)C'} | 16 |
| 3 | 1 | CCCCN1C(=S)NC(O)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCCC', 1: [*]O'} | 3 |
| 4 | 2 | CCN1C(=S)NC(N2CCCCC2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC', 1: [*]N1CCCCC1'} | 5 |
| 5 | 3 | CN1CCN(C2NC(=S)N(CCO)C2(O)c2ccccc2)CC1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCO', 1: [*]N1CCN(C)CC1'} | 4 |
| 6 | 4 | [O-]C1NC(=S)N(Cc2ccc(F)cc2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccc(F)cc1', 1: [*][O-]'} | 3 |
| 7 | 5 | C=CCN1C(=S)NC(N)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC=C', 1: [*]N'} | 1 |
| 8 | 6 | CCOC1NC(=S)N(CC(=O)OC(C)(C)C)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(=O)OC(C)(C)C', 1: [*]OCC'} | 1 |
| 9 | 7 | NC1NC(=S)N(Cc2ccc(Cl)cc2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccc(Cl)cc1', 1: [*]N'} | 2 |
| 10 | 8 | CCCN1C(=S)NC(OC)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCC', 1: [*]OC'} | 28 |
| 11 | 9 | CCN(CC)C1NC(=S)N(CC(C)C)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)C', 1: [*]N(CC)CC'} | 1 |
| 12 | 10 | CCOC1NC(=S)N(Cc2ccc(OC)cc2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccc(OC)cc1', 1: [*]OCC'} | 4 |
| 13 | 11 | OC1(c2ccccc2)C(N2CCNCC2)NC(=S)N1C1CC1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C1CC1', 1: [*]N1CCNCC1'} | 2 |
| 14 | 12 | CN1C(=S)NC([O-])C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C', 1: [*][O-]'} | 12 |
| 15 | 13 | O=[N+]([O-])c1cccc(CN2C(=S)NC([O-])C2(O)c2ccccc2)c1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1cccc([N+](=O)[O-])c1', 1: [*][O-]'} | 1 |
| 16 | 14 | CCOC1NC(=S)N(CCCl)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCCl', 1: [*]OCC'} | 1 |
| 17 | 15 | COC1NC(=S)N(CC(C)C)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)C', 1: [*]OC'} | 39 |
| 18 | 16 | CC(C)N1C(=S)NC(N2CCCC2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)C', 1: [*]N1CCCC1'} | 5 |
| 19 | 17 | CCOC1NC(=S)N(CCOC)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCOC', 1: [*]OCC'} | 3 |
| 20 | 18 | NNC1NC(=S)N(C(=O)c2cccc2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(=O)c1cccc1', 1: [*]NN'} | 1 |
| 21 | 19 | CC(C)CN1C(=S)NC(N2CCN(CCO)CC2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)C', 1: [*]N1CCN(CCO)CC1'} | 1 |
| 22 | 20 | CC(Cc1ccccc1)N1C(=S)NC([O-])C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)Cc1ccccc1', 1: [*][O-]'} | 1 |
| 23 | 21 | CC=C(C)N1C(=S)NC(NN)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)=CC', 1: [*]NN'} | 1 |
| 24 | 22 | CN1C(=S)NC(N[N+](=O)[O-])C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C', 1: [*]N[N+](=O)[O-]'} | 1 |
| 25 | 23 | CC(c1ccc(C#N)cc1)N1C(=S)NC(NN)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)c1ccc(C#N)cc1', 1: [*]NN'} | 1 |
| 26 | 24 | OC1(c2ccccc2)C(N2CCCC2)NC(=S)N1Cc1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccccc1', 1: [*]N1CCCC1'} | 2 |
| 27 | 25 | OC1(c2ccccc2)C([*:1])NC(=S)N1Cc1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccccc1'} | 16 |
| 28 | 26 | [*:0]N1C(=S)NC(N)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {1: [*]N'} | 41 |
| 29 | 27 | CC(=O)N1C(=S)NC(N(C)C)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)=O', 1: [*]N(C)C'} | 6 |
| 30 | 28 | CC(=O)OC1NC(=S)N(C(C)(C)C)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)(C)C', 1: [*]OC(C)=O'} | 1 |
| 31 | 29 | OC1NC(=S)N(CC2CC2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC1CC1', 1: [*]O'} | 1 |
| 32 | 30 | CC(C)CN1C(=S)NC(NC(=N)N)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)C', 1: [*]NC(=N)N'} | 5 |
| 33 | 31 | CC(C)N1C(=S)NC(NO)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)C', 1: [*]NO'} | 21 |
| 34 | 32 | CCNC1NC(=S)N(C(C)=O)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C(C)=O', 1: [*]NCC'} | 1 |
| 35 | 33 | [*:0]N1C(=S)NC(OC)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {1: [*]OC'} | 55 |
| 36 | 34 | CC(=O)OC1NC(=S)N(Cc2ccc(C)cc2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccc(C)cc1', 1: [*]OC(C)=O'} | 1 |
| 37 | 35 | CCCN1C(=S)NC(N(C)OC)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCC', 1: [*]N(C)OC'} | 1 |
| 38 | 36 | CCN(CC)C1NC(=S)N(CC)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC', 1: [*]N(CC)CC'} | 1 |
| 39 | 37 | OC1(c2ccccc2)C(N2CCOCC2)NC(=S)N1Cc1ccc(Br)cc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]Cc1ccc(Br)cc1', 1: [*]N1CCOCC1'} | 10 |
| 40 | 38 | ONC1NC(=S)N(C2CC2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C1CC1', 1: [*]NO'} | 1 |
| 41 | 39 | CCN1CCN(C2NC(=S)N(CC(C)C)C2(O)c2ccccc2)CC1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)C', 1: [*]N1CCN(CC)CC1'} | 1 |
| 42 | 40 | CCOC(C)CN1C(=S)NC(O)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)OCC', 1: [*]O'} | 117 |
| 43 | 41 | CN1CCN(C2NC(=S)N(C)C2(O)c2ccccc2)CC1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]C', 1: [*]N1CCN(C)CC1'} | 1 |
| 44 | 42 | CC(C)CN1C(=S)NC(NC2CCCC2)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)C', 1: [*]NC1CCCC1'} | 1 |
| 45 | 43 | CCCCN1C(=S)NC([O-])C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CCCC', 1: [*][O-]'} | 8 |
| 46 | 44 | CCOC(C)CN1C(=S)NC(NCC2CCCN2C)C1(O)c1ccccc1 | C2(C(C1=CC=CC=C1)(N(C(N2)=S)[*:0])O[H])[*:1] | {0: [*]CC(C)OCC', 1: [*]NCC1CCCN1C'} | 1 |

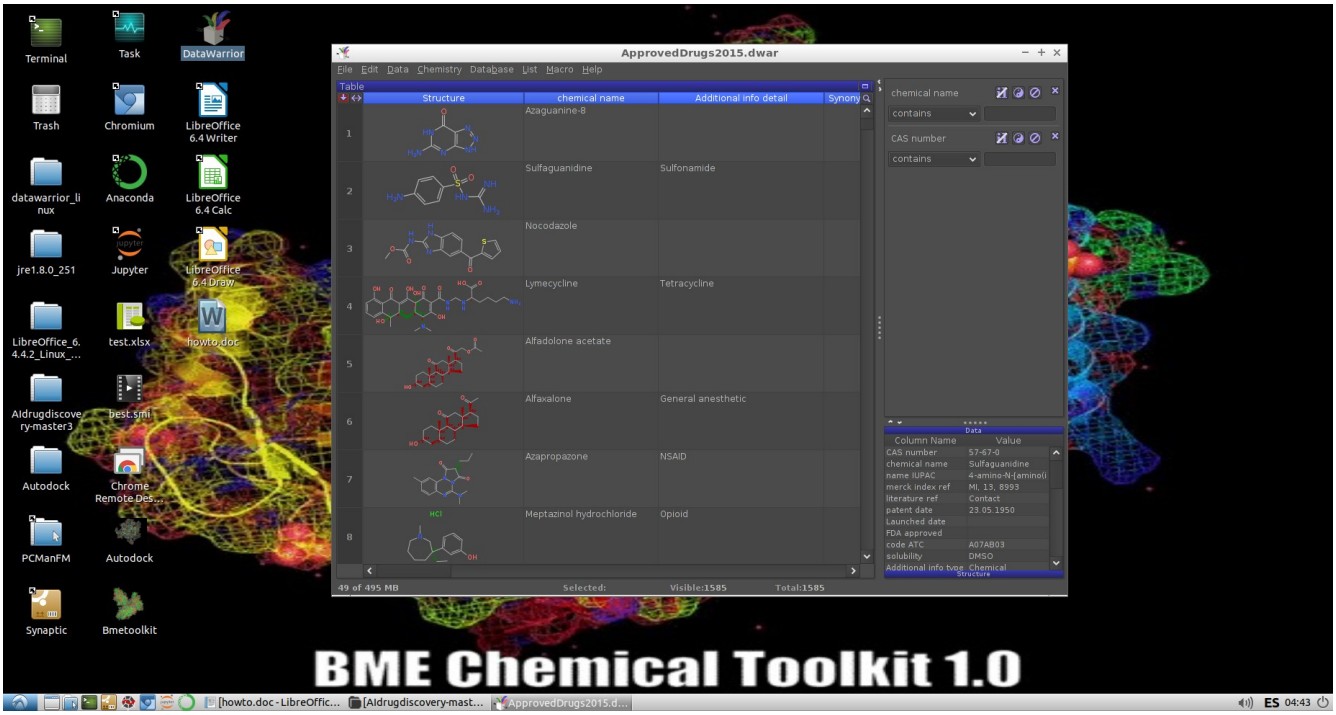
The output will be a csv file called generated molecules.csv that contains the generated SMILES and the scaffold it was generated from

Here is an example scaffold.

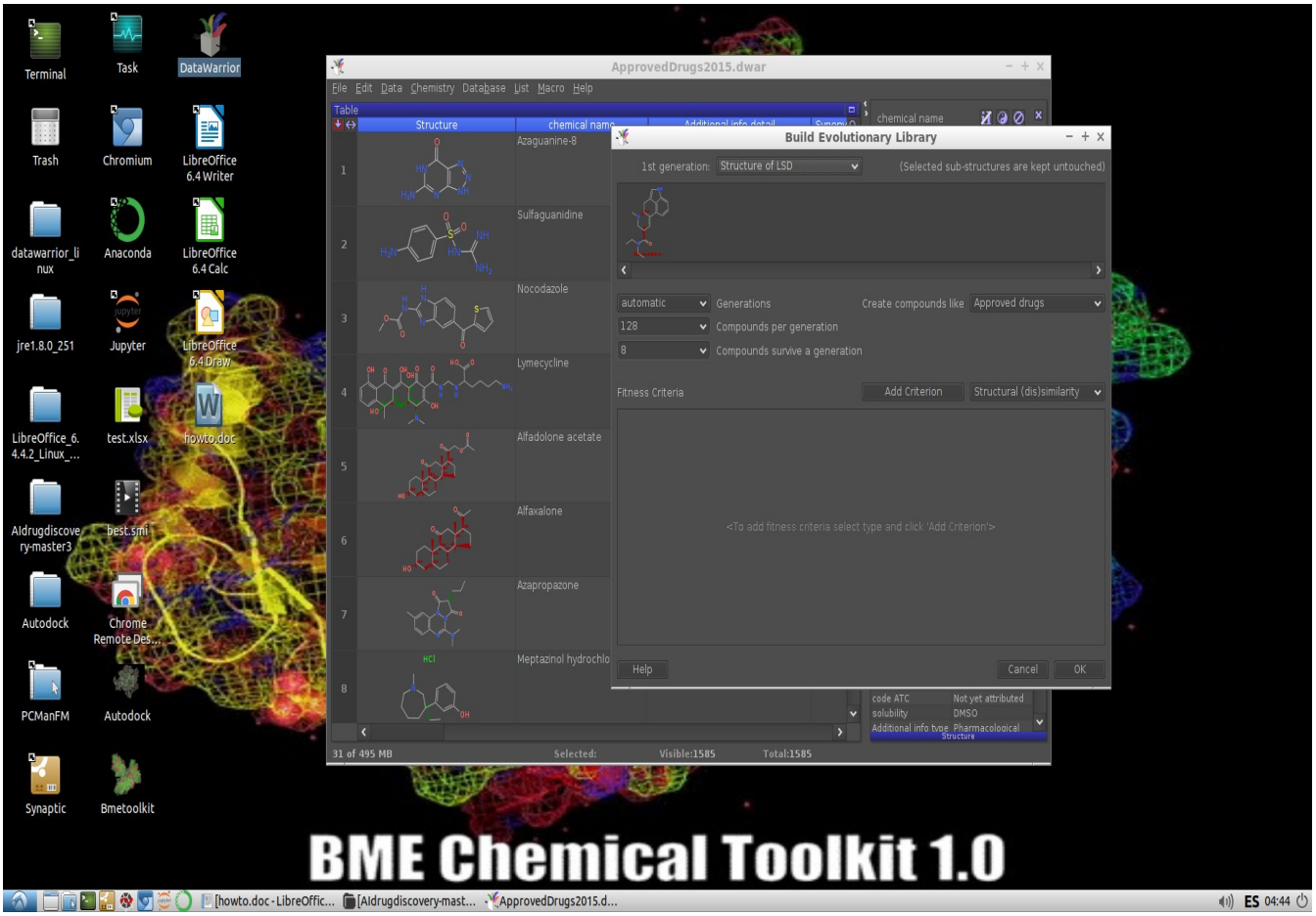



Other useful packages pre installed fpocket, openbabel, and pymol.  See documentation online

Data Warrior Library View allows you to store and edit libraries

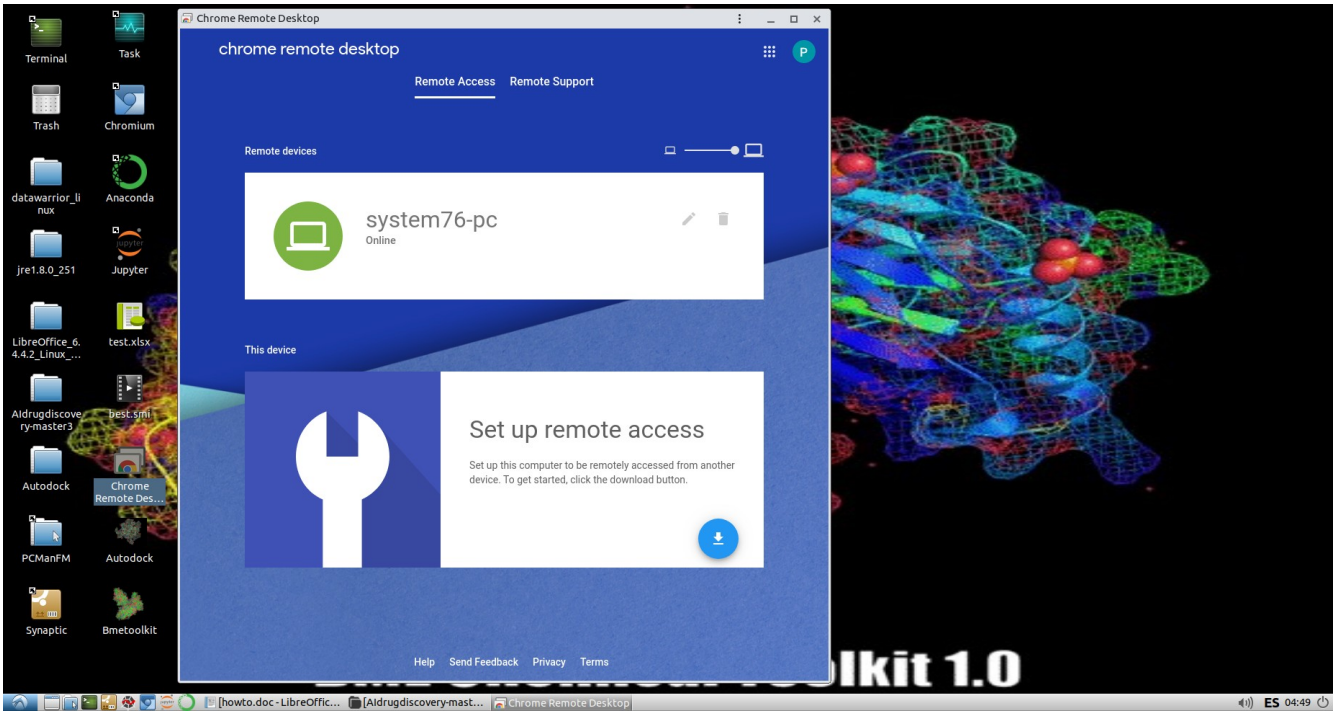



Generate an Evolutionary Library




Calculate a Drug Score under Macro, Calculate Properties

Chrome Remote Desktop allows you to share your screen remotely with someone through the browser




Open babel

open the shell type babel and you can convert between formats

fpocket

Allows you to find binding sites to use as docking coordinates for autodock

PyMOL

Allows you to view images