Processing Steps

1. Convert 1D and 2D structures to 3D
2. Add all hydrogen atoms to satisfy atoms valencies
3. Generate all/desired tautomeric variants
4. Generate all/desired Ionic variants
5. Generate all/desired Stereochemical variants
6. Resolve atomic clashes (Energy minimization)
7. Minimize energies of the ligands
8. Assign desired/spesific protonation state
9. Assign partial charges
10. Add torsion information (which bonds need to be rotatable)
11. Merge non-polar Hydrogens in case of AutoDock (Use Autodocktools)

LigPrep from the Schrodinger Suite can be used to generate the 3D coordinates of all the curated compounds in addition to exact protonation and tautomeric states at biological relevant Ph

Format Conversion Step

1. For example from sdf to pdbqt
2. For AutoDock Vina we would need pdbqt format

Tools Required for These Jobs

Free Tools	Purpose
Open Babel	Ligands library preprocessing & filtering
DataWarrior	Conformers Generation
Gypsum-DL	Tautomers & Stereoisomers Generation
Anaconda	For running Gypsum-DL

Installation of OpenBabel

- The latest version of OpenBabel is 3.0.0
- But we will use stable version of OpenBabel which is 2.3.1
- https://openbabel.org/docs/dev/Installation/install.html

Installation of DataWarrior

- http://www.openmolecules.org/datawarrior/

Installation of Gypsum-DL

- https://durrantlab.pitt.edu/gypsum-dl/
- Exract zip file as “gpysum”

Installation of Anaconda

- **Install anaconda and required packages**
- **Run Anaconda as "administrator" or with "sudo"**
- **"conda install -c rdkit rdkit numpy scipy"**
- **"conda install -c intel mpi4py"**

ALL DONE !

PREPROCESS LIGANDS

➢ **Use of OpenBabel for Ligand Preprocessing**

- **You can use OpenBabel through GUI or Command line**
- **I recommend researchers to use it from the command line**

Command	Purpose
"obabel input1.sdf input2.sdf input3.sdf -O output.sdf"	To combine all the libraries into one
"obabel input_file.sdf -O output_file.sdf -h"	Adding all the hydrogens to the molecules
"obabel input_file.sdf -O output_file.sdf -d"	Delete all the hydrogens from the molecules
"obabel 1Dstruct.smi -O 2Dstruct.sdf --gen2D "	Conversion of 1D molecules to 2D molecules (smi to sdf) ➢ If you have downloaded only the smile IDs of the molecules, then you need to convert them to 2D sdf structures ➢ Because smile IDs are 1D structures ➢ Suppose the ligand library name is "1Dstruct.smi" ➢ You want to convert it to "2Dstruct.sdf" ➢ To convert these structures to 2D sdf formats use the command below
"obabel 2Dstruct.sdf -O 3Dstruct.sdf --gen3D"	Conversion of 1D smi or 2D sdf to 3D sdf ➢ Suppose our 2D structures are saved in "2Dstruct.sdf" ➢ You want to convert them to "3Dstruct.sdf" ➢ Use the following command to convert 2D to 3D
"obabel 1Dstruct.smi -O 3Dstruct.sdf --gen3D"	Conversion of 1D smi to 3D sdf
"obabel input.sdf -O output.sdf --partialcharge gasteiger" **Or** "obabel input.sdf -O output.sdf --partialcharge eem" **Or** "obabel input.sdf -O output.sdf --partialcharge mmff94"	Calculating partial charges
"obabel input.sdf -O output.sdf --unique"	Dublicate Removal ➢ Type the following command in command prompt for duplicate removal from the library file
"obabel input.sdf -O output.sdf --filter "MW<500""	Filter molecules with molecular weight less than 500 d

"obabel input.sdf -O output.sdf --filter "!ROTATABLE_BOND || ROTATABLE_BOND > 5""	Filter molecules with rotatable bonds more than 5
"obabel input.sdf -O output.sdf –filter "logP>5""	Filter molecules with logP greater than 5
"obabel input.sdf -O output.sdf --conformer --nconf 20 --score rmsd --writeconformers" **Or** "obabel input.sdf -O output.sdf --conformer --nconf 20 --weighted --writeconformers" **Or** "obabel input.sdf -O output.sdf --conformer --nconf 20 --random --writeconformers" **Or** "obabel input.sdf -O output.sdf --conformer --nconf 20 --systematic --writeconformers" **Or** "obabel input.sdf -O output.sdf --conformer --nconf 20 --ff UFF --writeconformers" **Or** "obabel input.sdf -O output.sdf --conformer --nconf 20 --ff MMFF94 --writeconformers"	Conformer generations →First Method (--conformer method) ➢ You can use any of the following commands to produce multiple conformers of your structures in open babel ➢ NOTE: Use only one command. The number 20 after –conf is max number of conformers you want to generate for each ligand. You can change this number to 30, 40 or 50.
"obabel input.sdf -O output.sdf --confab --conf 100000" Or "obabel input.sdf -O output.sdf --confab --rcutoff 0.5" Or "obabel input.sdf -O output.sdf --confab --ecutoff 50" Or "obabel input.sdf -O output.sdf --confab --rcutoff 100 -original"	Conformer generations →Second Method (--confab method) ➢ If method 1 works for you then you don't need to use this method ➢ If method 1 doesn't work, then run only one of the following commands

NOTE: If multiple conformers generation (First and Second Method) does not work on "OpenBabel", DataWarrior or Gypsum_Dl will be used.

- **Using Gypsum_DL to Generate 3D structures of Different Kinds of Isomers of The Ligands**
  - **Enantiomers**
  - **Tautomers**
  - **Ions**
  - **Cis trans isomers etc.**
  - **NOTE: Gypsum_DL is free, open-source program that converts 1D and 2D small-molecule representations (SMILES string or flat SDF files) into 3D models. It outputs models with alternate ionization, tautomeric, chiral, cis/trans isomeric, and ring-conformational states.**

    - **We need our ligand library in smi format**
      - **Use OpenBabel for file format conversion**
      - **To convert your library from sdf to smi use the following command in OpenBabel "obabel input.sdf -O 1Dstruct_out.smi"**
    - **Tautomers and Stereoisomers Generation**
      - **Put your library into Gypsum folder**
      - **Run Anaconda as administrator(for Windows) or sudors(for linux)**
      - **And type "python run_gypsum_dl.py --source 1Dstruct_out.smi"**
      - **Your output file will be created in the Gypsum folder in sdf file format**
      - **The name of the output file would be "gypsum_dl_success.sdf"**
      -

- **Conformers Generation Using DataWarrior**
  - **You need your library to be in the sdf format**
  - **If your library is not in sdf convert it using OpenBabel**
  - **File > Open > already_processed_using_obabel_file.sdf**
  - **Select with Ctrl+A all structures**

- Chemistry > Generate Conformers
  - Use default options for “Algorithm, Initial torsions, Minimize energy”
  - check “Write into file”
  - File Type “SD-File Version 3
  - Click “OK”
- Visualize Created sdf file on Discovery Studio Visualizer
  - Clicked one of the conformers arrow
    - View > check “Hierarchy” and see all conformers that you generated.
- Visualize Your Created Conformers in DataWarrior
  - On “Table” Widget Right clicked the structure
    - Select “Explore conformers of ‘Structure’”

SUMMARY

obabel 1Dstruct.smi -O 2Dstruct.sdf --gen2D
obabel 2Dstruct.sdf -O 3Dstruct.sdf --gen3D
obabel input1.sdf input2.sdf input3.sdf -O output.sdf
obabel input_file.sdf -O output_file.sdf -h
obabel input.sdf -O output.sdf --partialcharge gasteiger
obabel input.sdf -O output.sdf --filter “MW<500”
obabel input.sdf -O output.sdf --filter “!ROTATABLE_BOND || ROTATABLE_BOND > 5”
obabel input.sdf -O output.sdf –filter “logP>5”
obabel input.sdf -O output.sdf --conformer --nconf 30 --score rmsd –writeconformers
(This above step did not work for me, so I had to use DataWarrior for conformers generation)

DataWarrior and Gypsum-DL for Conformers generation
obabel input.sdf -O output.pdb
(Don’t run this command at this stage, we will use this later)

SOME IMPORTANT POINTS

1) Hydrogens can not be added by OpenBabel if your output file is in smi format
2) gen3D of obabel adds hydrogens and creates 3D structures
3) gen2D of obabel can not add hydrogens, it only creates 2D structures
4) Use smi format as input to Gypsum. It will give a sdf output
5) Gypsum adds hydrogen, creates 3D and conformers
6) DataWarrior can not use smi as input. Its output will be a sdf file
7) DataWarrior adds hydrogen, creates 3D and conformers
8) Don’t remove duplicates after conformer generation
9) In ligand flexible docking you don’t need to generate 3D and conformers