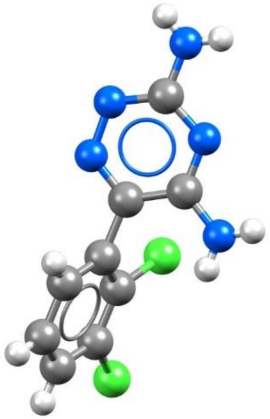
How to Generate Ensembles of Conformers using the CSD Conformer Generator (CONF-002)

2024.1 CSD Release



CCDC
advancing structural science

Table of Contents

Introduction

The CSD Conformer Generator provides the ability to both minimise molecular conformations and generate diverse conformer subsets based on CSD data. The methodology starts from an input 3D molecular structure with all hydrogen atoms present, which is optionally minimised in the first step. Subsequently, conformations are sampled based on CSD-derived rotamer distributions and ring templates. A final diverse set of conformers, clustered according to conformer similarity, is returned. Each conformer is locally optimised in torsion space. There are three methods for generating conformers with the CSD portfolio: via the Mercury interface, the command line utility and the CSD Python API.

Before beginning this workshop, ensure that you have a registered copy of CSD-Materials or CSD-Discovery or CSD-Enterprise installed on your computer.

Learning Outcomes

In this tutorial, you will learn how to generate ensembles of conformers for a set of known CDK2 inhibitors using the three methods. After completing this workshop, you will be able to generate conformers using:

- The command line utility of the CSD Conformer Generator program,
- The Mercury interface.
- The CSD Python API.

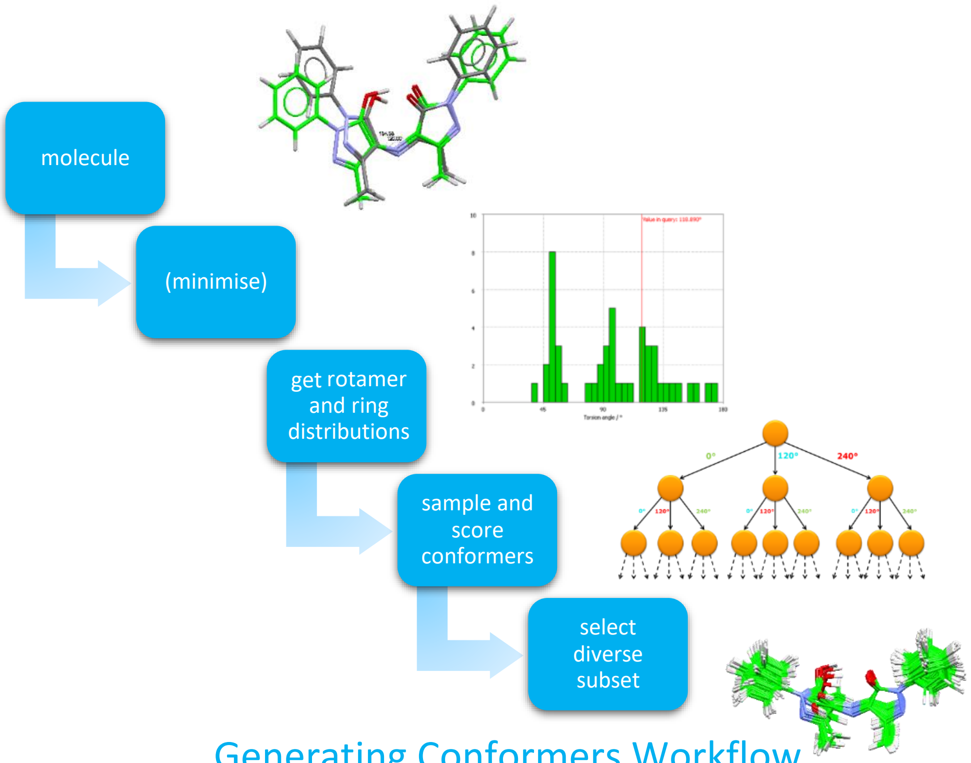
Note: This handout should take approximately 25 **minutes** to be completed.

Pre-required Skills

Familiarity with the Mercury interface is important; you can access the Visualization in Mercury self-guided workshop here. Familiarity with the command line interface and the CSD Python API is recommended. You can find introduction to the CSD Python API on this page.

Materials

Please download the example files here.
https://downloads.ccdc.cam.ac.uk/tutorials/CDK2.zip


molecule
(minimise)
get rotamer and ring distributions
sample and score conformers
select diverse subset
0°
120°
240°

Generating Conformers Workflow

Case study

Cyclin-dependent kinase 2 (CDK2) is an important protein kinase required for promoting the cell division cycle and for successful progression through S and G2 phases. This role in the cell cycle progression has led to an active search of small molecule compounds inhibiting this enzyme as potential anticancer drugs.

Many studies have been published that describe various CDK inhibitors which target the ATP pocket of CDK2. They bind by hydrophobic interactions and by forming hydrogen bonds with the kinase, especially with the backbone of Glu81 and Leu83 in the structure of the apoenzyme (Figure 1).

Many structures of CDK2 in complex with different types of inhibitors have been deposited in the Protein Data Bank (PDB).

In 2013, researchers at AstraZeneca published an extensive and diverse set of molecular overlays for the validation of pharmacophore programs (DOI: 10.1021/ci400020a). This benchmarking data set contains the experimental overlay of 24 CDK2 inhibitors taken from high resolution structures available from the PDB. A subset of 13 molecules (see below) arbitrarily chosen but covering all different chemotypes represented in the whole set, will be used as a case study in this tutorial.

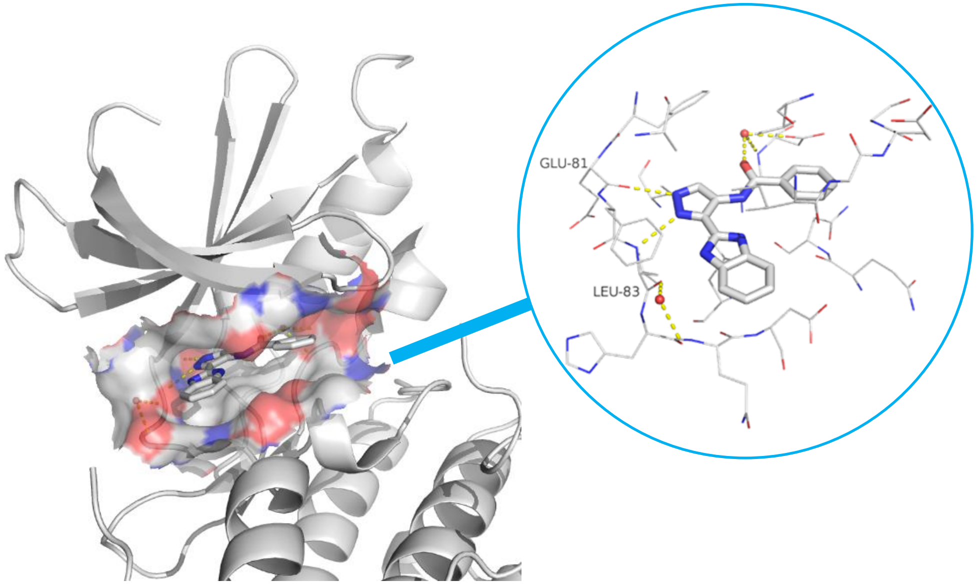
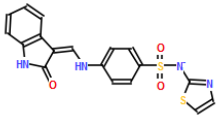
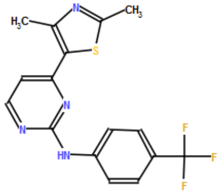
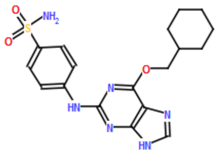
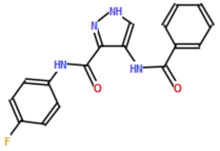


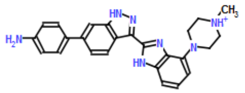
Figure1. CDK2 in complex with a fragment-like molecule. Hydrogen bond interactions are shown as dashed yellow lines.

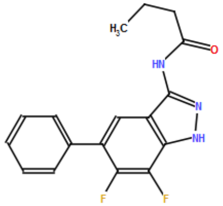

1ke8

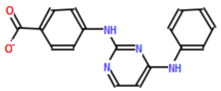

1pxl

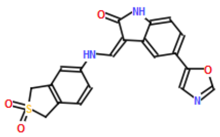

2iw9

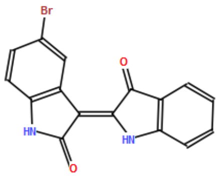

2vto

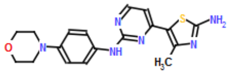

3ezv

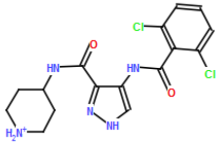

3lfq

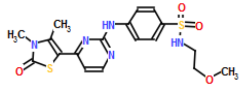

3unj

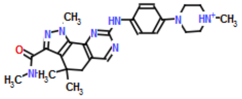

1ke7


2bhe


2uue


2vu3


2xmy


2wih

1.1 Command line utility

The command line utility of the CSD Conformer Generator is used from a command prompt and requires a local install of python.

1. To open a command prompt on **Windows**, open File Explorer, go to the directory you are working in (e.g. `my_path_to\CDK2\input_molecules\Command_line`).
   Click on the address bar and type `cmd`, this will open the command prompt with the path to your current folder already set.

   If you are a **macOS** user, use a `terminal` to go to your working directory (e.g. `my_path_to/CDK2/input_molecules/Command_line`).

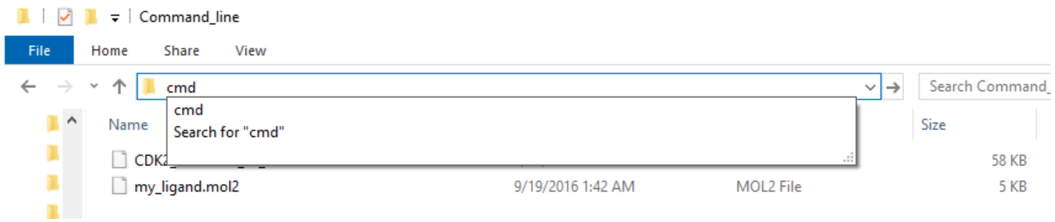
   Then, to launch the CSD Conformer Generator program, you have to specify the path to its executable*:

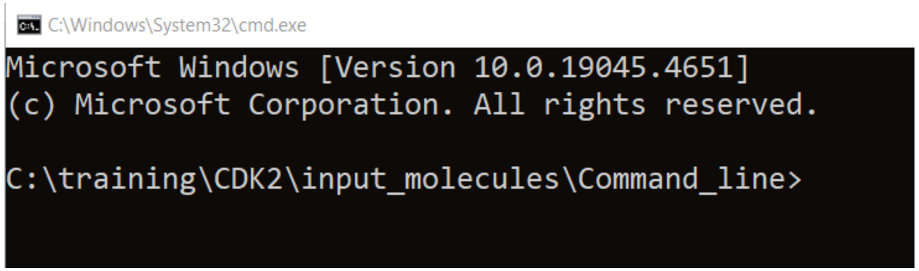
   a) on **Windows**, the conformer generator script is in `"path\to\CCDC\ccdc-software\conformer-generator\conformer_generator"`

   b) on **macOS**, the conformer generator script is in `“/path/to/CCDC/ccdc-software/conformer-generator/Conformer_Generator/bin/conformer_generator”`

   c) on Linux, the conformer generator script is in `“/path/to/CCDC/ccdc-software/conformer-generator/bin”`

1





```
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\training\CDK2\input_molecules\Command_line>
```

* The default install location for the CSD portfolio is the user’s home area but can be installed in a separate location. “path\to\” should be replaced with the path to your CCDC folder. On MacOS, the Applications folder can be specified during installation.

2. **If you are a Windows user, type:**

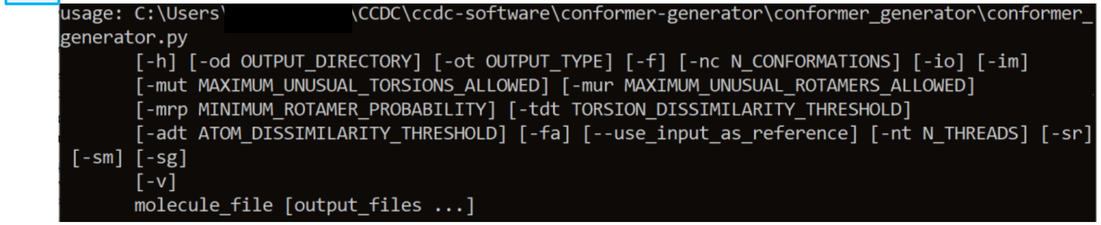
   ```
   python "C:\path\to\CCDC\ccdc-software\conformer-generator\conformer_generator\conformer_generator.py " -h
   ```

   If you are a macOS user, type:

   ```
   /Applications/CCDC/ccdc-software/conformer-generator/Conformer_Generator/bin/conformer_generator -h
   ```

   `-h` command provides help instructions on all the supported commands of the conformer generator tool. Linux users can run the same command as macOS but provide the correct path for their install.

2



```
usage: C:\Users\[redacted]\CCDC\ccdc-software\conformer-generator\conformer_generator\conformer_
generator.py
       [-h] [-od OUTPUT_DIRECTORY] [-ot OUTPUT_TYPE] [-f] [-nc N_CONFORMATIONS] [-io] [-im]
       [-mut MAXIMUM_UNUSUAL_TORSIONS_ALLOWED] [-mur MAXIMUM_UNUSUAL_ROTAMERS_ALLOWED]
       [-mrp MINIMUM_ROTAMER_PROBABILITY] [-tdt TORSION_DISSIMILARITY_THRESHOLD]
       [-adt ATOM_DISSIMILARITY_THRESHOLD] [-fa] [--use_input_as_reference] [-nt N_THREADS] [-sr]
 [-sm] [-sg]
       [-v]
       molecule_file [output_files ...]
```

3. At the simplest level, the CSD Conformer Generator can be used with an input file and an output file, with no further options.

   If you are a Windows user:

   Type the command:

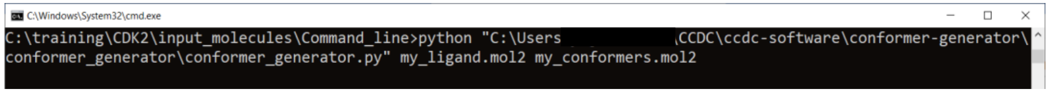
   ```
   python "C:\path\to\CCDC\ccdc-software\conformer-generator\conformer_generator\conformer_generator.py" my_ligand.mol2 my_conformers.mol2
   ```

   If you are a macOS user:

   Type the command:

   ```
   Applications/CCDC/ccdc-software/conformer-generator/Conformer_Generator/bin/conformer_generator my_ligand.mol2 my_conformers.mol2
   ```

3



```
C:\training\CDK2\input_molecules\Command_line>python "C:\Users[redacted]CCDC\ccdc-software\conformer-generator\
conformer_generator\conformer_generator.py" my_ligand.mol2 my_conformers.mol2
```

Note that supported formats for the input and output files are *.sdf* and *.mol2*.

By default, the program will generate up to 200 conformations for each molecule in the input file, sorted by their normalised score which is their likelihood based on the knowledge driven by similar molecules in the CSD.

4. If you wish, the normalised scores can be written out as a comma separated file (*.csv*) by extending the command

   If you are a Windows user

   Type the command:

   ```
   python "C:\path\to\CCDC\ccdc-software\conformer-
   generator\conformer_generator\conformer_generator.py"
   my_ligand.mol2 my_conformers.mol2 my_conformers.csv -f
   ```

   If you are a macOS user:

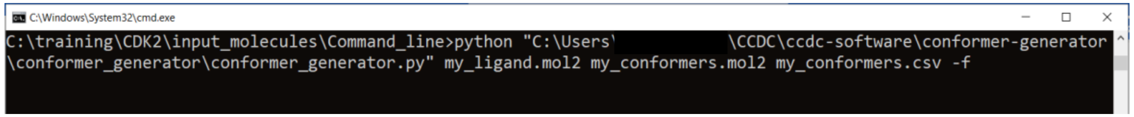
   Type the command:

   ```
   Applications/CCDC/ccdc-software/conformer-
   generator/Conformer_Generator/bin/conformer_generator
   my_ligand.mol2 my_conformers.mol2 my_conformers.csv -f
   ```

The `-f` command is required to overwrite any output files if already present in the output folder. In addition to the `my_conformers.mol2` file (Figure 2), which includes up to 200 conformers of `my_ligand.mol2`. The command above will automatically produce two other files:

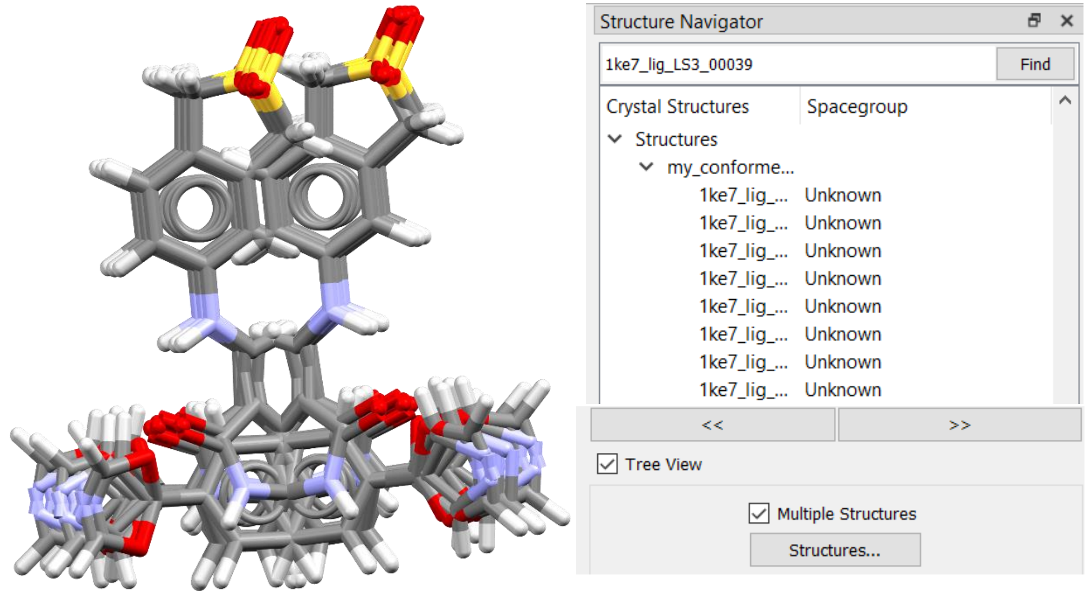
- `my_conformers.csv` containing data for each ligand run through the conformer generator, and
- `my_conformers_per_conformer_scores.csv` containing data for each retained conformer per ligand run through the conformer generator.
  This file, shown on the right, includes several data items for each conformer such as the conformer probability, the clash score measured for this conformer, and a score that normalises the probability into the range of [0.0,1.0], where a zero score indicates the conformer with the highest likelihood and a score of 1.0 would indicate the conformer with the least likelihood.

4



C:\Windows\System32\cmd.exe

```
C:\training\CDK2\input_molecules\Command_line>python "C:\Users\[redacted]\CCDC\ccdc-software\conformer-generator
\conformer_generator\conformer_generator.py" my_ligand.mol2 my_conformers.mol2 my_conformers.csv -f
```



Structure Navigator

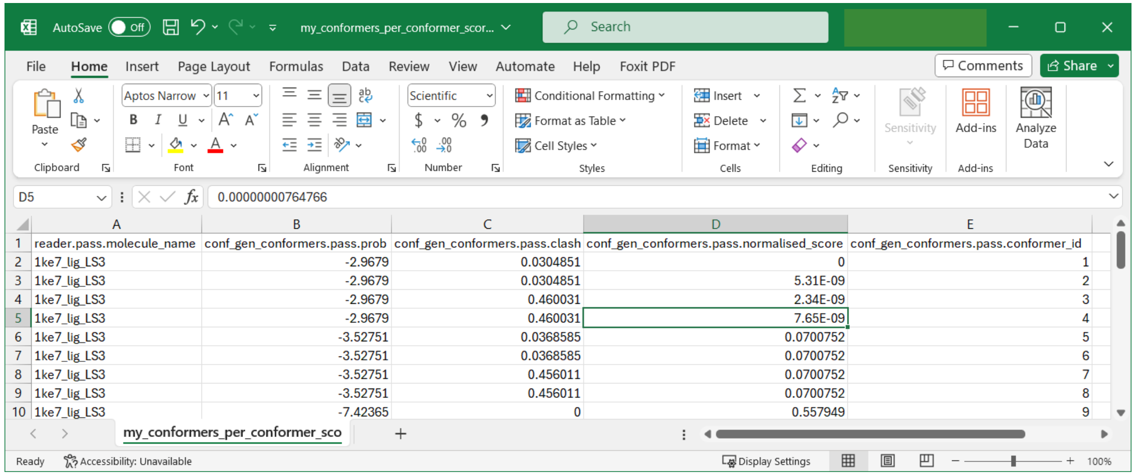
1ke7_lig_LS3_00039 Find

Crystal Structures	Spacegroup
Structures	
my_conforme...	
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown
1ke7_lig_...	Unknown

☑ Tree View

☑ Multiple Structures

Figure 2: *Conformers generated for the molecule 1ke7_lig_LS3.mol2. By default, they are not superimposed on any substructure.* To create this view in Mercury, open the my_conformers.mol2 file. One structure will be loaded, tick the Multiple Structures box in the lower right and click to select the other structures to superimpose them.



D5 0.00000000764766

	A	B	C	D	E
1	reader.pass.molecule_name	conf_gen_conformers.pass.prob	conf_gen_conformers.pass.clash	conf_gen_conformers.pass.normalised_score	conf_gen_conformers.pass.conformer_id
2	1ke7_lig_LS3	-2.9679	0.0304851	0	1
3	1ke7_lig_LS3	-2.9679	0.0304851	5.31E-09	2
4	1ke7_lig_LS3	-2.9679	0.460031	2.34E-09	3
5	1ke7_lig_LS3	-2.9679	0.460031	7.65E-09	4
6	1ke7_lig_LS3	-3.52751	0.0368585	0.0700752	5
7	1ke7_lig_LS3	-3.52751	0.0368585	0.0700752	6
8	1ke7_lig_LS3	-3.52751	0.456011	0.0700752	7
9	1ke7_lig_LS3	-3.52751	0.456011	0.0700752	8
10	1ke7_lig_LS3	-7.42365	0	0.557949	9

5. Finally, we are going to generate a separate output file of conformers for each of the 13 CDK2 inhibitors. This is performed by specifying a subdirectory to contain the output files, rather than specifying a single output file.

   If you are a Windows user

   Type the command:

   ```
   "C:\path\to\CCDC\ ccdc-software\conformer-
   generator\conformer_generator\conformer_generator.py "
   CDK2_inhibitors_13_molecules.mol2 -ot mol2 -od conformers
   ```

   If you are a macOS user:

   Type the command:

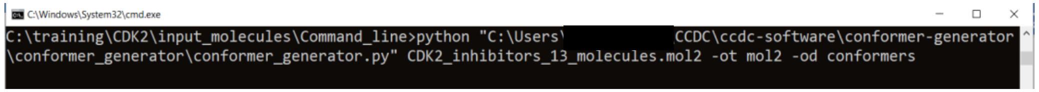
   ```
   Applications/CCDC/ccdc-software/conformer-
   generator/Conformer_Generator/bin/conformer_generator
   CDK2_inhibitors_13_molecules.mol2 -ot mol2 -od conformers
   ```

   This will create a new directory called `conformers` containing all the generated conformers in *.mol2* format. Note that molecule *2bhe_lig_BRY* does not contain any rotatable bond; therefore, it will be minimised but no conformers will be generated for it. You can retrieve information about the number of conformers generated by looking at the *conformer_generator.log* file in your `my_path_to/CDK2/input_molecules/Command_line` folder.

If less than 200 conformers are generated (e.g. for ligand *1pxl_lig_CK4*) it means that, according to the CSD Conformer Generator clustering algorithm, the search space may be exhausted before 200 distinct conformers are generated.

Note that it is possible to run the CSD Conformer Generator using multiple threads on a given machine using the `–nt` command. Please note that all threads will share the memory available to the conformer generator process.

5



```
C:\training\CDK2\input_molecules\Command_line>python "C:\Users\[redacted]CCDC\ccdc-software\conformer-generator
\conformer_generator\conformer_generator.py" CDK2_inhibitors_13_molecules.mol2 -ot mol2 -od conformers
```

1.2 The Mercury interface

The CSD Conformer Generator can also be accessed through the **CSD-Materials** and **CSD-Discovery** menus within Mercury .

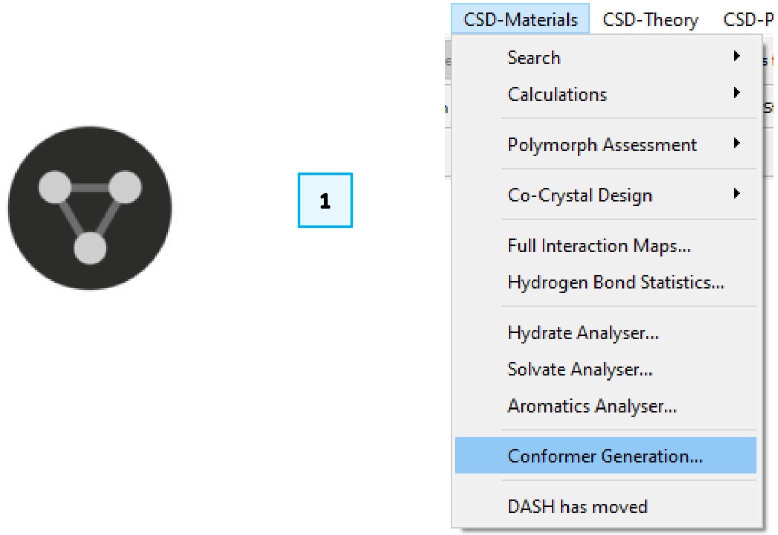
1. Start Mercury by double-clicking on the icon. From the CSD-Materials menu, click on *Conformer Generator*.

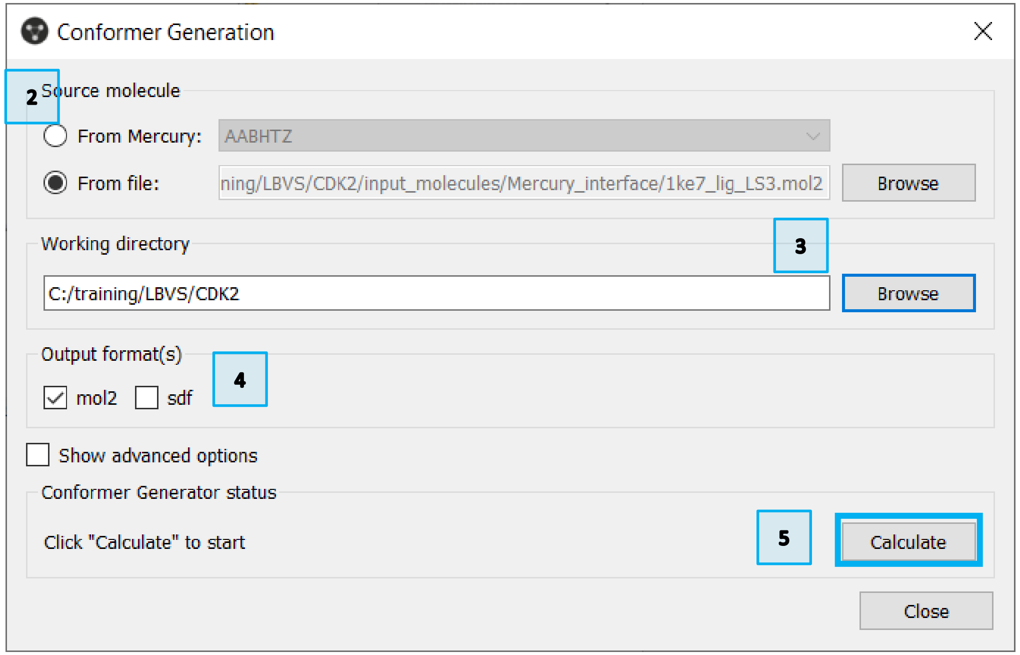
2. In the **Conformer Generation** window in the *Source molecule* section you can specify a CSD entry, an in-house database entry, or another structure as the input molecule for conformer generation. Alternatively, you can select a file on the disk as the input by selecting *From file***:** option and clicking on the **Browse** button. In order to load the file *1ke7_lig_LS3.*mol2, navigate to `my_path_to\CDK2\input_molecules\Mercury_interface`

3. The working directory can be selected under *Working Directory* section of the *Conformer Generation* window. Click the **Browse** button and navigate to `my_path_to\CDK2\input_molecules\Mercury_interface`. A folder will be created in this directory named after the identifier of the input molecule used. Both a copy of the input molecule and the generated conformers will be written to the output folder.

4. Conformers may be generated in *.mol2* or *.sdf* format by choosing the relevant checkboxes under *Output format(s)*. The default *.mol2* format will be used in this tutorial.

5. Click the **Calculate** button to generate conformers.





Conformer Generation

Source molecule

○ From Mercury: AABHTZ

● From file: ning/LBVS/CDK2/input_molecules/Mercury_interface/1ke7_lig_LS3.mol2 Browse

Working directory

C:/training/LBVS/CDK2 Browse

Output format(s)

☑ mol2 □ sdf

□ Show advanced options

Conformer Generator status

Click "Calculate" to start Calculate

Close

6. A multi-file containing 128 conformers will be saved in `my_path_to\CDK2\i1ke7_lig_LS3\CSD_Conformer_Generator\YYYY_MM_DD_hh_mm_ss\conformers.mol2`.

Please note that YYYY_MM_DD_hh_mm_ss is the date and time when the calculation was run. You will have a folder name corresponding to your own calculation date and time. The conformers are also populated in the Structure Navigator as new entries.

You can use the same procedure to create the conformer libraries of the other 12 CDK2 inhibitors.

1.3 The CSD Python API

The CSD Conformer Generator functionality is also fully available via the CSD Python API, allowing you to link conformer generation seamlessly to other applications and enabling complex workflows and analyses.

To generate conformer libraries for all 13 CDK2 inhibitors, via the CSD Python API, all you need are a few very simple Python statements. The script on the right of the next page shows how to generate conformers for all *.mol2* files stored in a given working directory. Prior to generation, molecules are standardised according to CSD conventions, and the resulting conformers overlaid using all heavy atoms as shown in Session 3 of this tutorial.

1. **If you are a Windows user,** open File Explorer, go to the directory you are working in. This may be:
   `my_path_to\CDK2\input_molecules\CSD_Python_API`
   Click on the address bar and type `cmd`, this will open the command prompt with the path to your current folder already set.

   If you are a macOS user, use a `terminal` to go to your working directory (e.g. `my_path_to/CDK2/input_molecules/CSD_Python_API`).

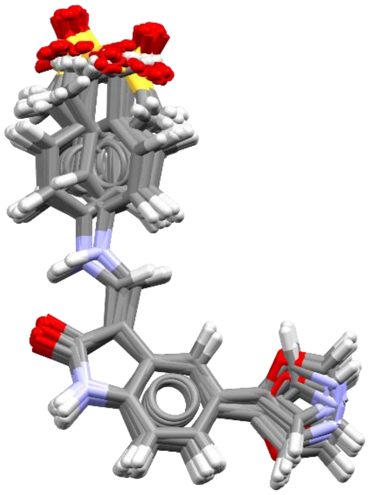
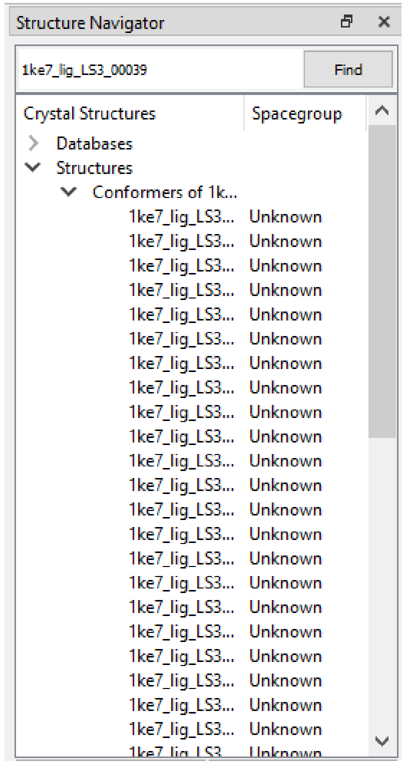


Figure 3: Ensemble of conformers generated for the molecule *1ke7_lig_LS3.mol2*. the input molecule was used as reference to superimpose the resulting conformations on all heavy atoms.

6

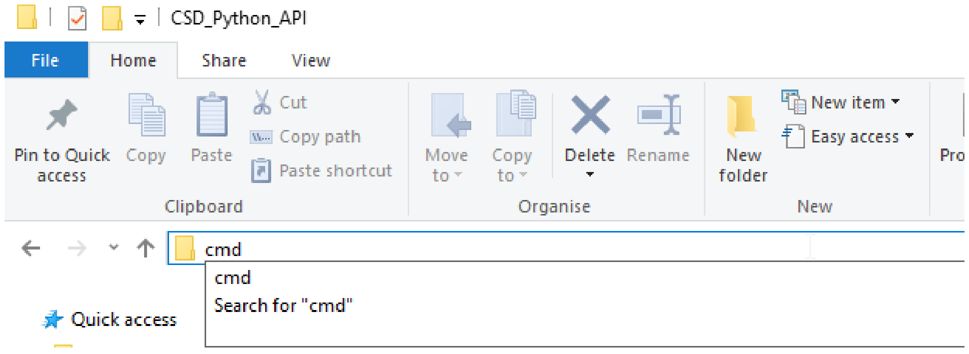


Structure Navigator

1ke7_lig_LS3_00039 Find

Crystal Structures	Spacegroup
Databases	
Structures	
Conformers of 1k...	
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3...	Unknown
1ke7_lig_LS3	Unknown

1



cmd

cmd

Search for "cmd"

2. In order to use the CSD python API, we need to activate the Python environment where it has been installed together with its dependencies.

If you are a Windows user:

In the command prompt type
`C:\path\to\CCDC\ccdc-software\csd-python-api\miniconda\Scripts\activate`

If you are a macOS user:

In the terminal type:
`source /path/to/CCDC/ccdc-software/csd-python-api/miniconda/bin/activate`

The above command will run the CSD Python API activation script. As you can see you are now working in a different environment called `(base)`.

Any subsequent use of Python in that terminal will use the version installed with the CSD. This step can be reversed by using the `deactivate` command.

3. We are now ready to use our Python script to generate the conformers of the 13 CDK2 inhibitors.

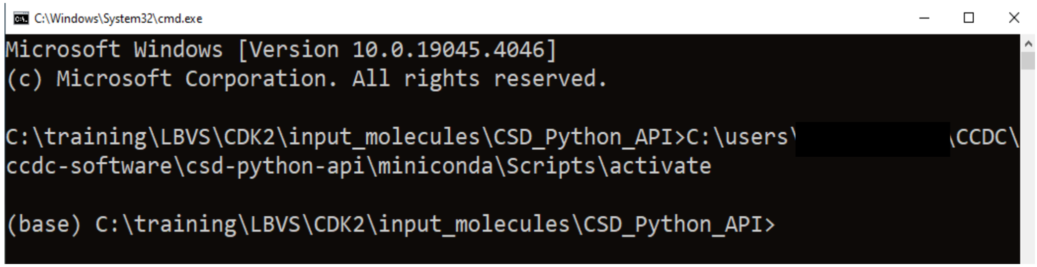
Type `python generate_confs.py` to run it.

Note that parameters can be changed via the `settings` object:

- **`max_conformers`**
  Maximum number of conformers to generate.
- **`max_unusual_torsions`**
  Number of unusual torsions allowed per conformer.
- **`normalised_score_threshold`**
  Maximum deviation from the theoretically achievable normalised conformer probability (0=”best”, 1=”worst”).
- **`superimpose_conformers_onto_reference`**
  Whether or not to superimpose to a common reference.

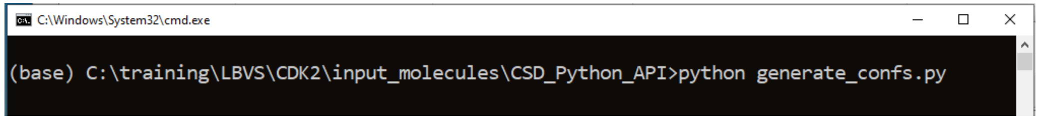
The generated conformers will be placed in your working folder once the script concludes.

2

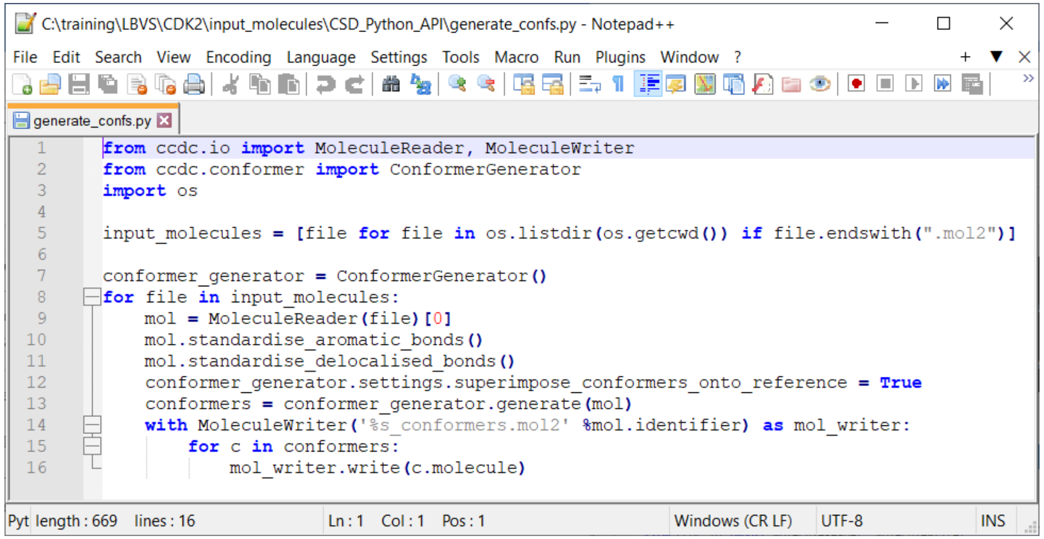


```
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\training\LBVS\CDK2\input_molecules\CSD_Python_API>C:\users\[redacted]\CCDC\
ccdc-software\csd-python-api\miniconda\Scripts\activate

(base) C:\training\LBVS\CDK2\input_molecules\CSD_Python_API>
```

3



```
(base) C:\training\LBVS\CDK2\input_molecules\CSD_Python_API>python generate_confs.py
```


C:\training\LBVS\CDK2\input_molecules\CSD_Python_API\generate_confs.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
generate_confs.py

```python
from ccdc.io import MoleculeReader, MoleculeWriter
from ccdc.conformer import ConformerGenerator
import os

input_molecules = [file for file in os.listdir(os.getcwd()) if file.endswith(".mol2")]

conformer_generator = ConformerGenerator()
for file in input_molecules:
    mol = MoleculeReader(file)[0]
    mol.standardise_aromatic_bonds()
    mol.standardise_delocalised_bonds()
    conformer_generator.settings.superimpose_conformers_onto_reference = True
    conformers = conformer_generator.generate(mol)
    with MoleculeWriter('%s_conformers.mol2' %mol.identifier) as mol_writer:
        for c in conformers:
            mol_writer.write(c.molecule)
```

Conclusion

Using either the command line utility of the CSD Conformer Generator, or the Mercury interface, or the CSD Python API, you were able to generate a number of high-probability conformations for the dataset of CDK2 inhibitors. These libraries can now be passed on to other tools for post-processing. For example, they can be used as input for the CSD Ligand Overlay to generate overlay hypotheses for pharmacophore modelling.

Summary

In this workshop, you have learnt how to use the three available methods for generating conformers in the CSD portfolio. You should now be able to:

- Generate an ensemble of conformers.
- Analyse and visualize an ensemble of conformers.

Next Steps

The CSD Conformer Generator User Guide can be found here or from the Documentation and Resources section of our website. There is a separate workshop on running the Conformer Generator in Mercury on a single molecule that can be found on this page.

https://www.ccdc.cam.ac.uk/community/training-and-learning/workshop-materials/csd-materials-workshops/

Feedback

We hope this workshop improved your understanding of the *CSD Conformer Generator* and you found it useful for your work. As we aim to continuously improve our training materials, we would love to hear your feedback. Click on this link to a survey (link also available from workshops webpage), it will take less than 5 minutes to complete. The feedback is anonymous. You will be asked to insert the workshop code, which for this self-guided workshop is CONF-002. Thank you!