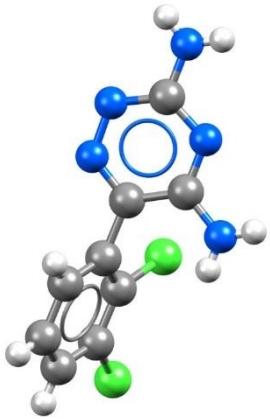
Ligand-Based Virtual Screening

2020.0 CSD Release



CCDC
advancing structural science

Table of Contents

Introduction

In the absence of three-dimensional (3D) structures of potential drug targets, ligand-based drug design is one of the most popular approaches for drug discovery and lead optimisation. Pharmacophore modelling is a widely-used tool in ligand-based drug design and can provide predictive models suitable for lead compound optimisation.

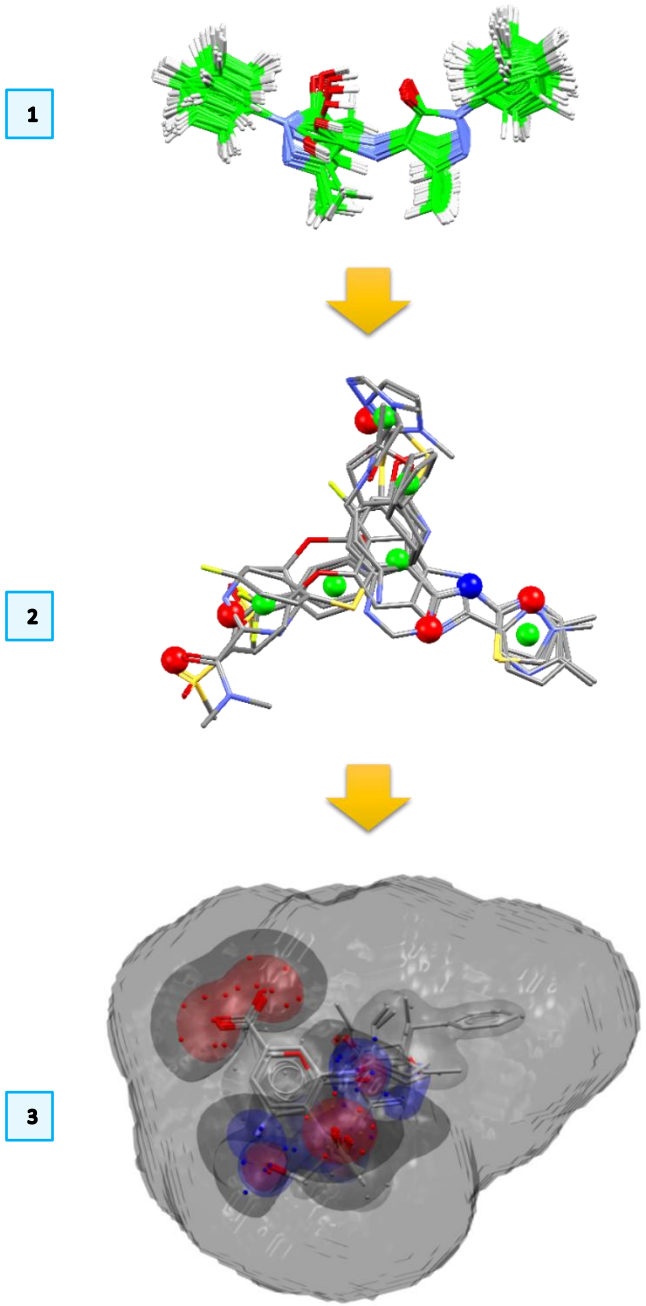
According to the IUPAC definition, a pharmacophore model is “an ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target and to trigger (or block) its biological response”.

A pharmacophore model can be generated by superposing a set of active molecules that are assumed to bind to the same target with the same binding mode, and extracting common chemical features that are essential for their bioactivity. Such a model can then be used to virtually screen libraries of compounds with the aim of identifying potential new binders of the target of interest.

In this tutorial, you will learn how to perform a ligand-based virtual screening using a suite of knowledge-based tools.

1. First, ensembles of conformers will be generated for a set of known CDK2 inhibitors.
2. Overlay hypotheses for these ligands will be produced using the CSD-Ligand Overlay program and a pharmacophore model will be selected for step 3.
3. To validate the predictive power of our model, a library of active and inactive (or decoy) compounds will be screened against the selected overlay solution and enrichment metrics will be calculated.

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



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.

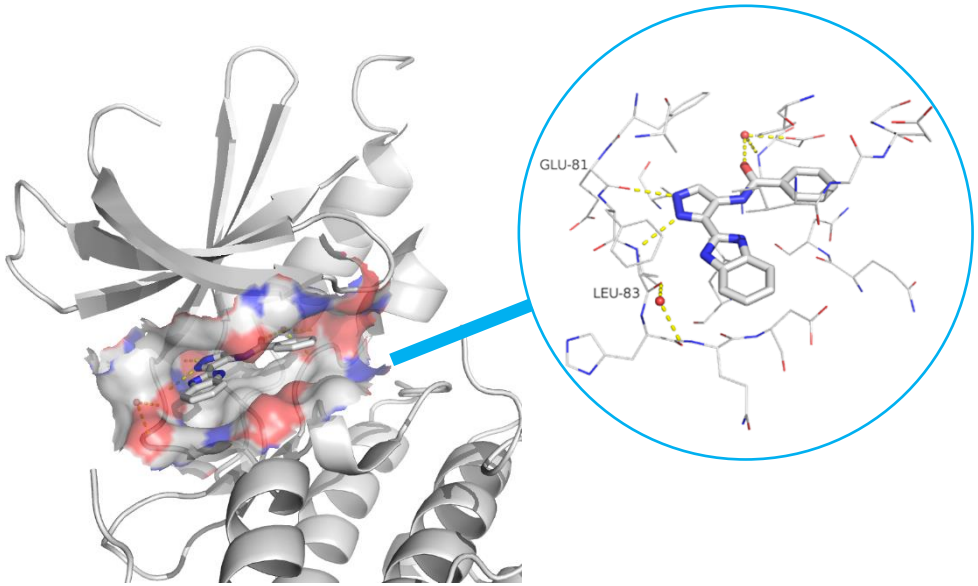
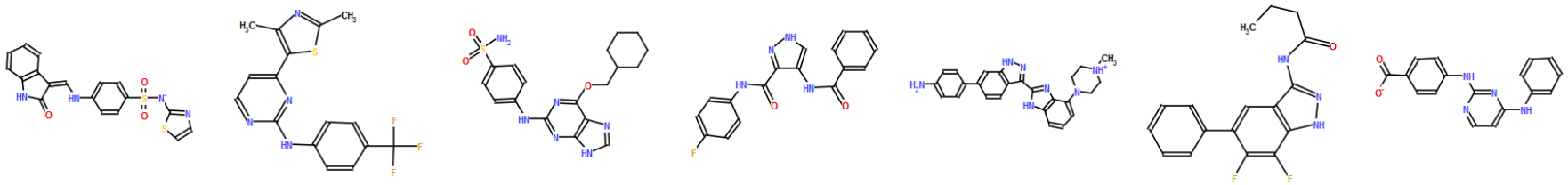
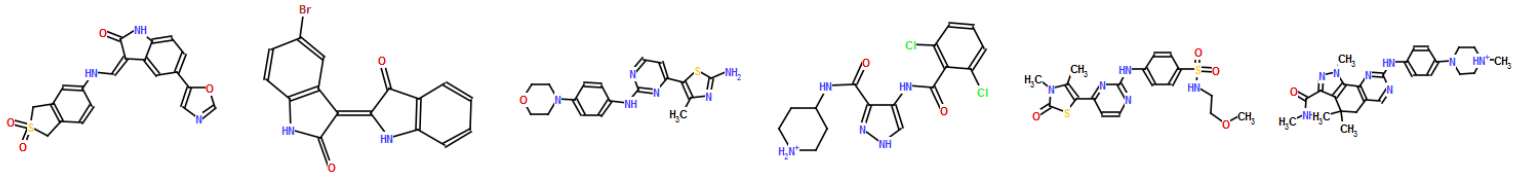


Figure1. CDK2 in complex with a fragment-like molecule. Hydrogen bond interactions are shown as dashed yellow lines.



1ke8 1pxl 2iw9 2vto 3ezv 3lfq 3unj



1ke7 2bhe 2uue 2vu3 2xmy 2wih

1. Generating conformers

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.

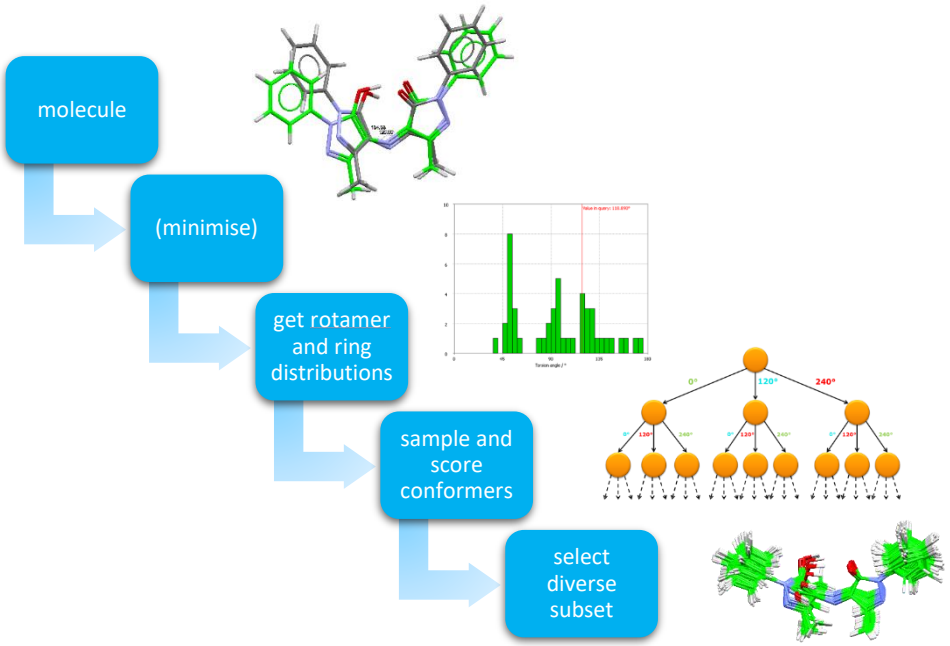
To avoid any bias from the experimentally observed conformation for this example, we regenerated 3D coordinates for the 13 molecules using the program Corina.

The CSD Conformer Generator requires as input 3D structures with all hydrogen atoms present. The ensembles of high probability knowledge-driven conformations can then be generated:

1) Through the command line utility of the CSD Conformer Generator program,
2) Using the Mercury interface,
3) Using the CSD Python API.

In this tutorial we provide a brief overview of how to use these three methods for conformer generation. Choose only <u>ONE</u> of these methodologies to generate the conformers of the 13 CDK2 inhibitors.

Generating Conformers Workflow



1.1 Command line utility

The command line utility of the CSD Conformer Generator is used from a command prompt.

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) By default, on **Windows**, the conformer generator.exe is located in `"C:\Program Files\CCDC\CSD_2020\Conformer Generator\"`

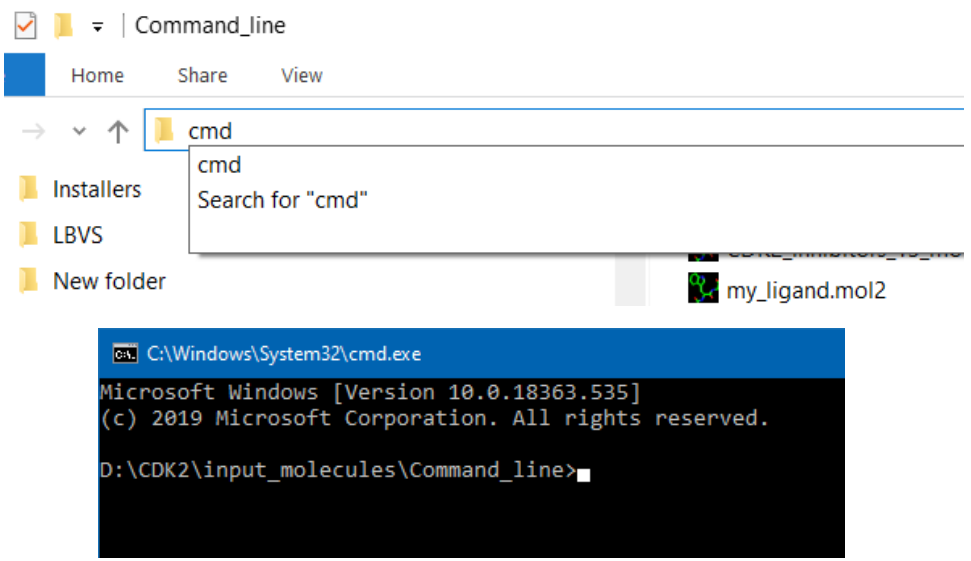
   b) By default, on **macOS**, the conformer generator executable is located in `/Applications/CCDC/CSD_2020/Conformer Generator/bin/`

2. **If you are a Windows user:**
   Type `"C:\Program Files\CCDC\CSD_2020\Conformer Generator\conformer_generator" -h`

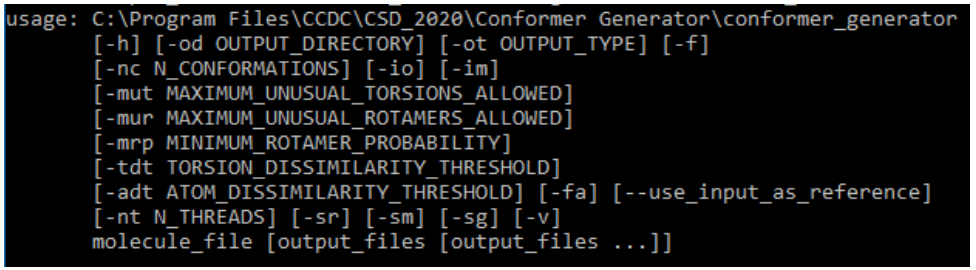
   If you are a macOS user:
   Type `/Applications/CCDC/CSD_2020/Conformer Generator/bin/conformer_generator -h`

   `-h` command provides help instructions on all the supported commands of the conformer generator tool.

1



2



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:

   ```
   "C:\Program Files\CCDC\CSD_2020\Conformer Generator\conformer_generator" my_ligand.mol2 my_conformers.mol2
   ```

   If you are a macOS user:

   Type the command:

   ```
   Applications/CCDC/CSD_2020/Conformer Generator/bin/conformer_generator 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:

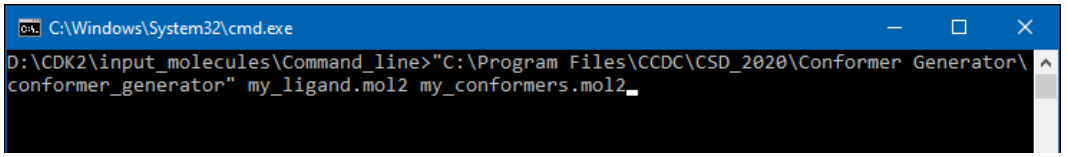
   ```
   "C:\Program Files\CCDC\CSD_2020\Conformer Generator\conformer_generator" my_ligand.mol2 my_conformers.mol2 my_conformers.csv -f
   ```

   If you are a macOS user:

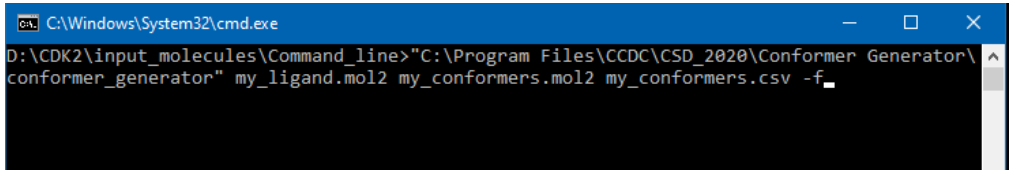
   Type the command:

   ```
   Applications/CCDC/CSD_2020/Conformer Generator/bin/conformer_generator my_ligand.mol2 my_conformers.mol2 my_conformers.csv -f
   ```

3



4



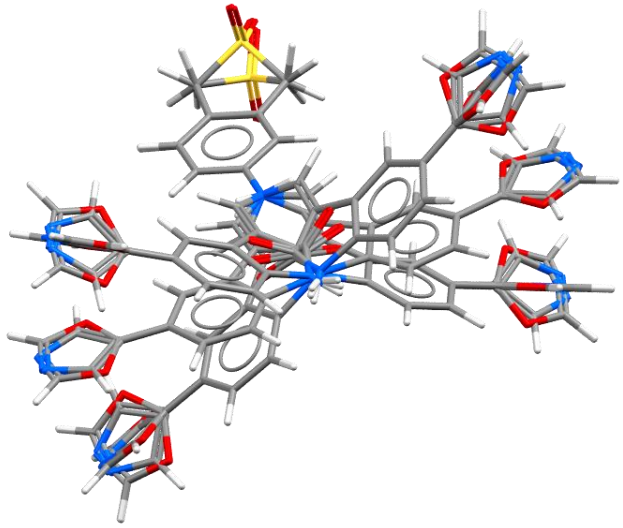
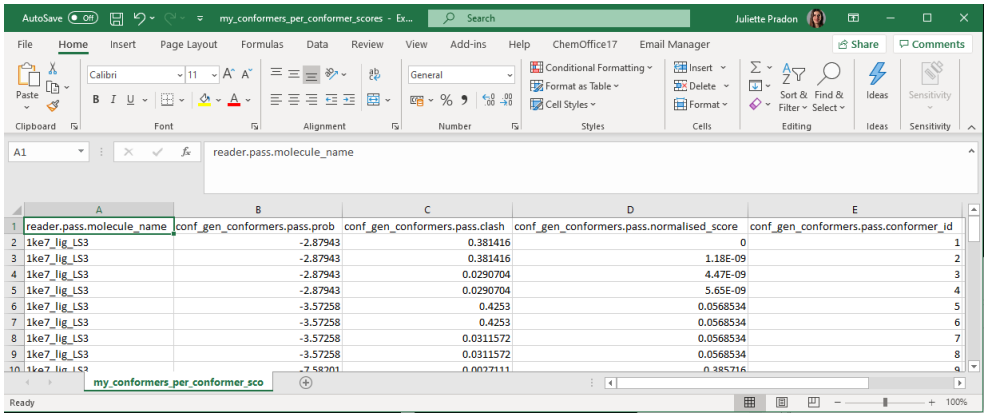


Figure 2: *Conformers generated for the molecule 1ke7_lig_LS3.mol2. By default, they are not superimposed on any particular substructure.*

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:

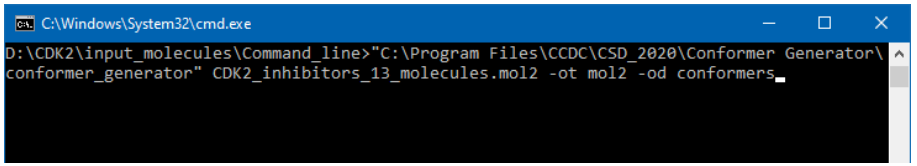
- `my_conformers.csv` containing data for each ligand run through the conformer generator, and
- `my_conformers_per_conformer_scores.csv` containing data for each retained conformer per ligand run through the conformer generator.
  This file, shown on the right, includes several data items for each conformer 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.



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
1ke7_lig_LS3	-2.87943	0.381416	0	1
1ke7_lig_LS3	-2.87943	0.381416	1.18E-09	2
1ke7_lig_LS3	-2.87943	0.0290704	4.47E-09	3
1ke7_lig_LS3	-2.87943	0.0290704	5.65E-09	4
1ke7_lig_LS3	-3.57258	0.4253	0.0568534	5
1ke7_lig_LS3	-3.57258	0.4253	0.0568534	6
1ke7_lig_LS3	-3.57258	0.0311572	0.0568534	7
1ke7_lig_LS3	-3.57258	0.0311572	0.0568534	8

5

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:\Program Files\CCDC\CSD_2020\Conformer
Generator\conformer_generator"
CDK2_inhibitors_13_molecules.mol2 -ot mol2 -od conformers
```

If you are a macOS user:

Type the command:

```
Applications/CCDC/CSD_2020/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.

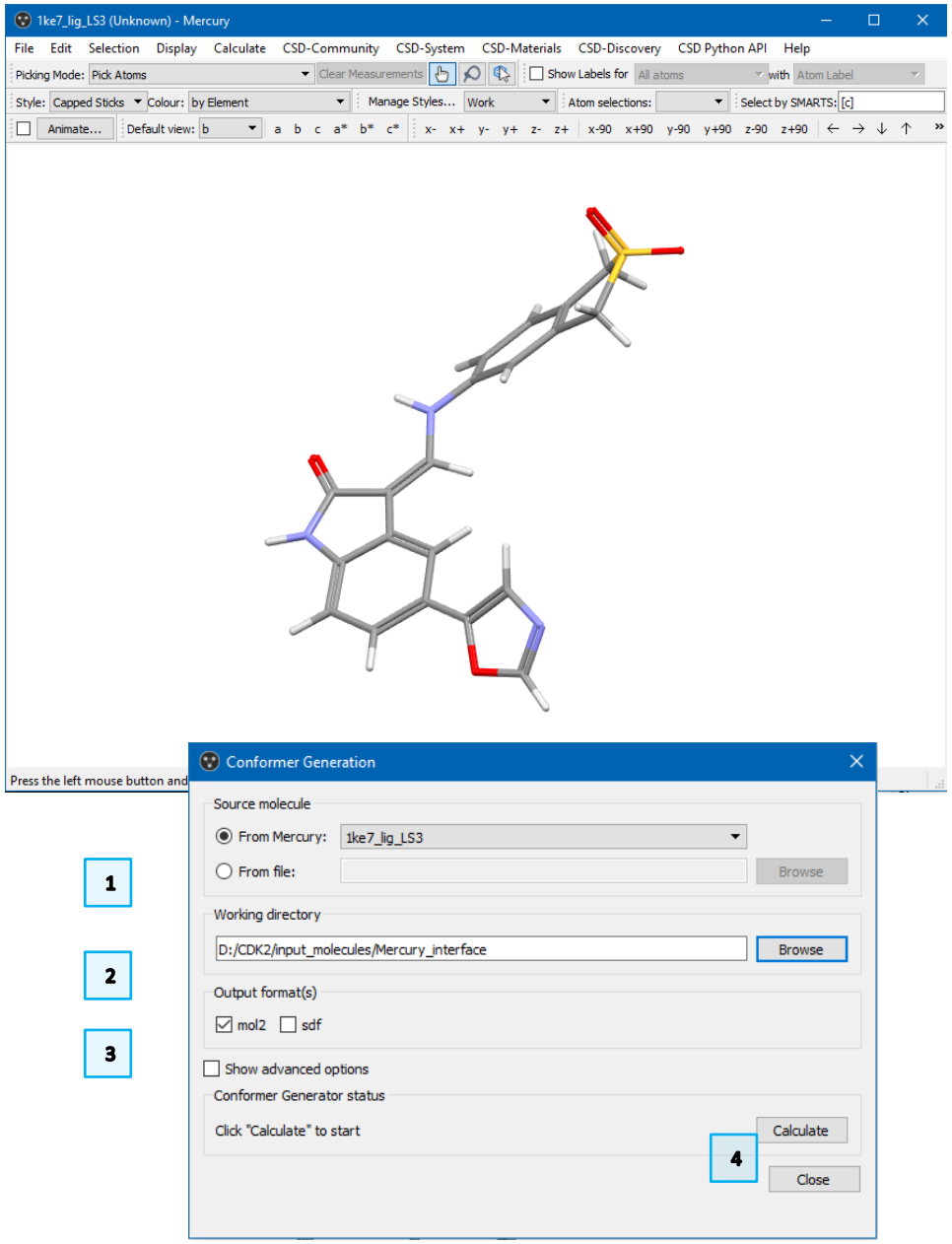
If less than 200 conformers are generated (e.g. for ligand *1pxl_lig_CK4*) it means that, according to the CSD Conformer Generator clustering algorithm, the search space may be exhausted before 200 distinct conformers are generated.

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.

1.2 The Mercury interface

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

1. In the *Conformer Generation* window in the *Source molecule* session 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`
2. 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.



3. 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.

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

5. A multi-file containing 128 conformers (Figure 3) will be saved in `my_path_to\CDK2\input_molecules\Mercury_interface\1ke7_lig_LS3\CSD_Conformer_Generator\2019_12_18_10_15_09\conformers.mol2`.

Please note that `2019_12_18_10_15_09` is the date (18/12/2019) and time (10:15:09) when the calculation was run. You will have a different folder name corresponding to your own calculation date and time.

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`

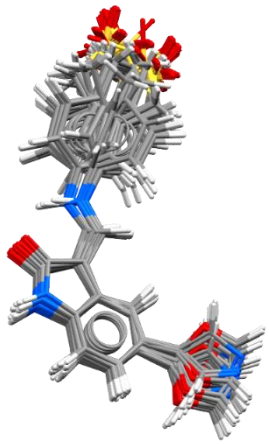
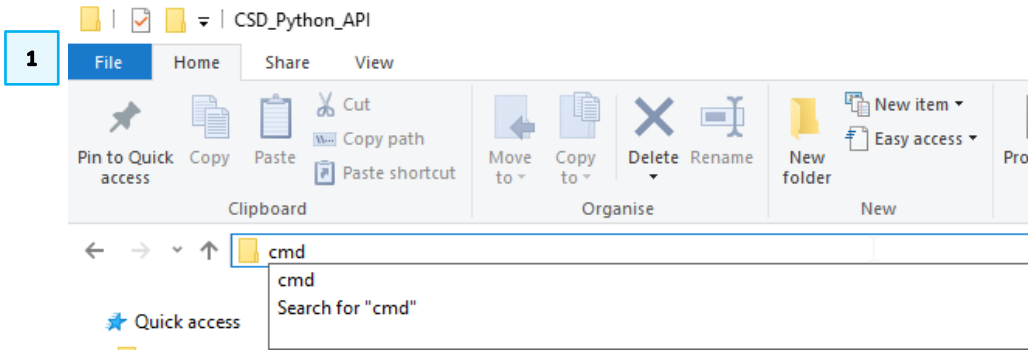


Figure 3. Ensemble of conformers generated for the molecule *1ke7_lig_LS3.mol2*. the input molecule was used as reference to superimpose the resulting conformations on all heavy atoms.



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`).

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:\Program Files\CCDC\Python_API_2020\miniconda\Scripts\activate.bat"
```

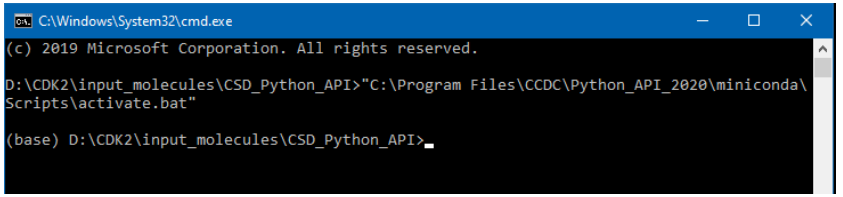
If you are a macOS user:

In the terminal type:

```
source Applications/CCDC/Python_API_2020/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)` .

2

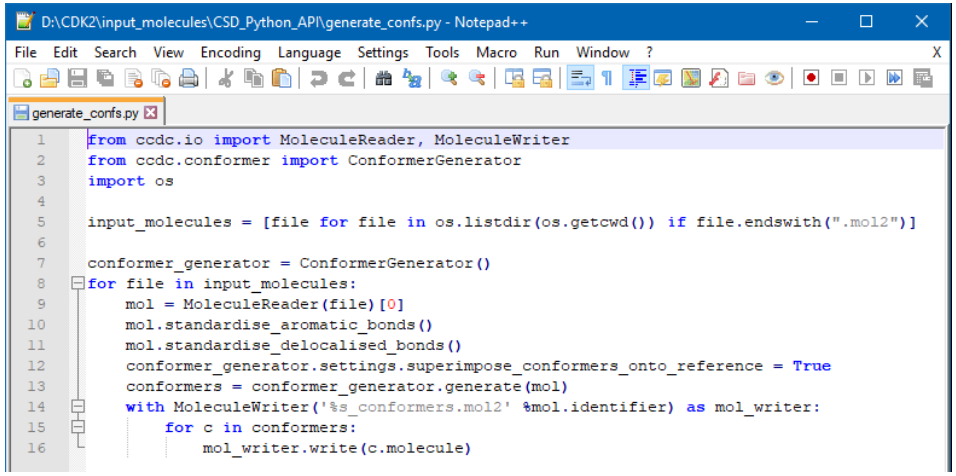


Any subsequent use of Python in that terminal will use the version installed with the CSD-System. This step can be reversed by using the `deactivate` command.

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.

3



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.

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.

2. Overlaying ligands

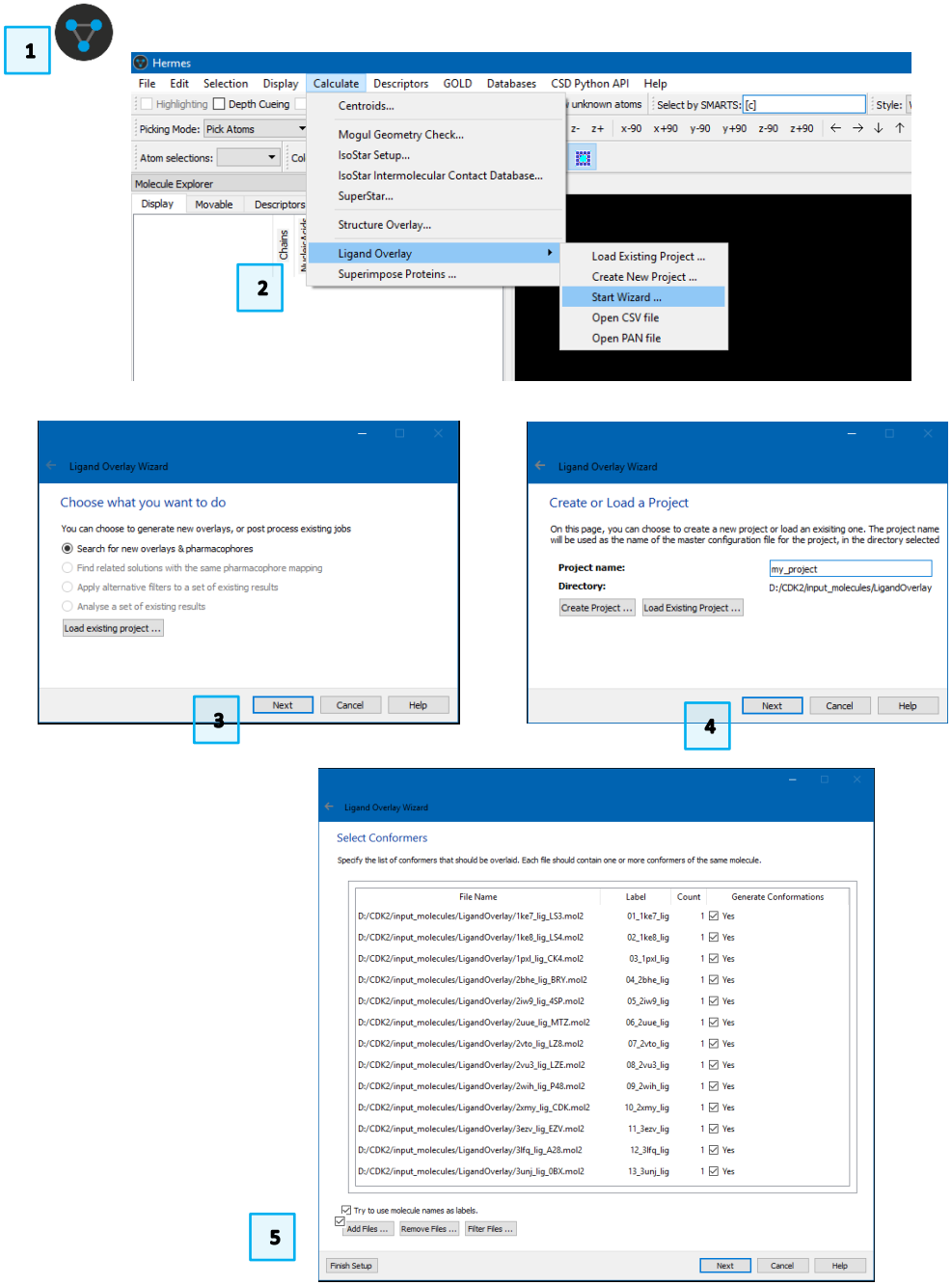
The CSD Ligand Overlay program aims to overlay sets of flexible molecules that are assumed to be ligands of the same protein. The molecules to be overlaid are divided into *features* such as hydrogen-bond donors and acceptors, and hydrophobic groups.

In the absence of any protein-structure information, there may be multiple overlays that present legitimate hypotheses for binding.

In order to address this problem, the algorithm used by the overlay program returns multiple diverse overlays that all give potential pharmacophore hypotheses to test.

1. Launch **Hermes** by clicking the Hermes Icon on your desktop or if you are a macOS user launching it from the Applications menu.
2. From the top-level menu choose **Calculate > Ligand Overlay > Start Wizard…** It is possible to load an existing project, but as no calculations have been run so far, this is not appropriate. Here we will generate some possible overlays of the 13 CDK2 inhibitors, as discussed in the Introduction.
3. In the *Choose what you want to do* dialogue, choose option to **Search for new overlays & pharmacophores** and click **Next** (or **Continue** on MacOS).
4. In the *Create or Load a Project* dialogue, click **Create Project...**. This allows you to navigate to an existing directory or create a new directory for the results to be saved. You can also edit the project name in the *Project name:* textbox. Once you have done this, click **Next**.
5. In the *Select Conformers* dialogue, you will be prompted to add the conformer files for all ligands that you wish to overlay.

   Note that you can either load pre-generated conformer files (e.g. as the ones generated in Section 1 of this tutorial) or supply each ligand as a single conformer file (in which case by default the program will generate conformers



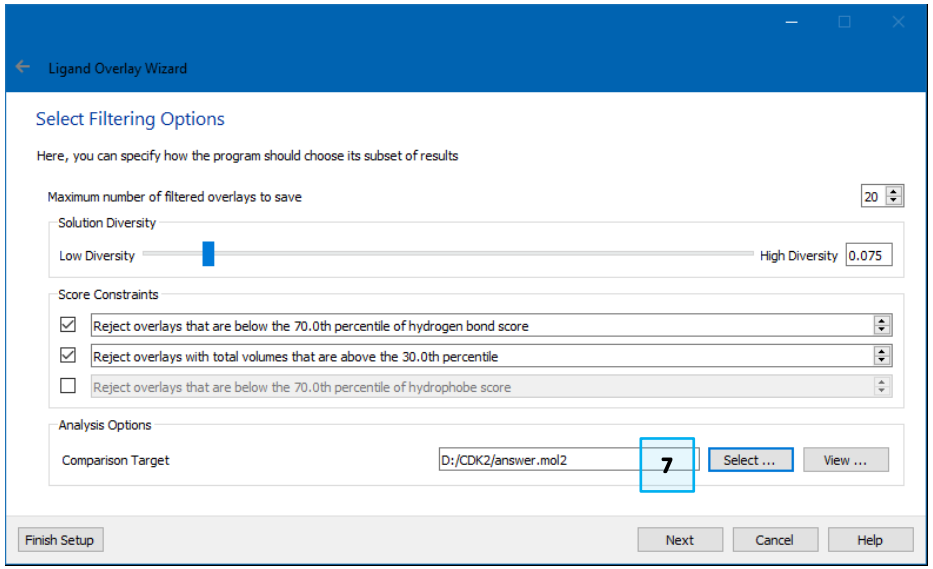
for each ligand). Here we are going to use the second option by clicking **Add Files…** and selecting all ligands in `my_path_to\CDK2\input_molecules\LigandOverlay`. Ensure the **Try to use molecule names as labels** check-box is ticked.

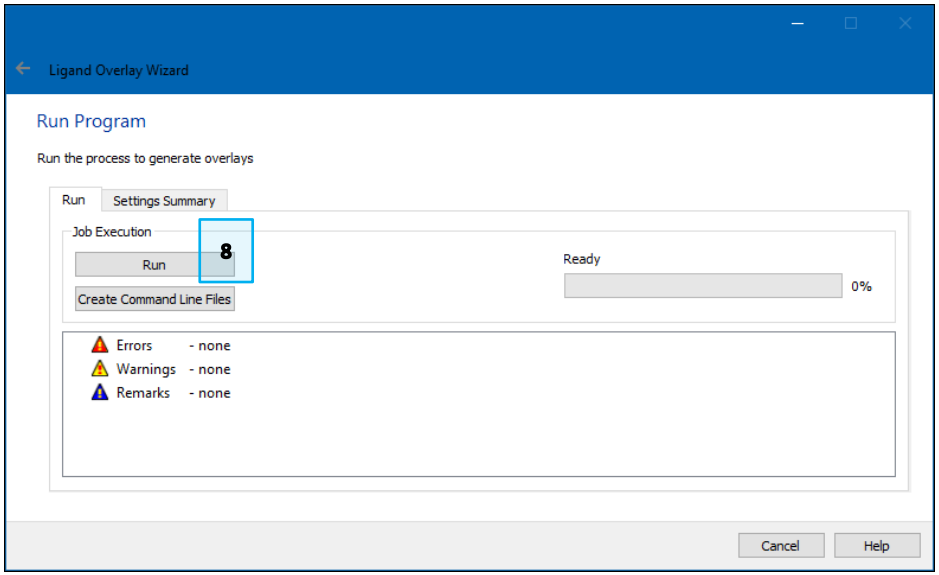
6. Click **Next** to go to the *Select Parameter Settings* dialog. As the default parameters are appropriate for this tutorial we could just click **Finish Setup** and start the calculation. However, as we are performing a retrospective study of known CDK2 inhibitors for which the crystal structures of the protein-ligand complexes are available, we will use this information to assess the quality of the results.

7. Click **More Options** and then **Next** until you get to the *Select Filtering Options* dialog. In the *Analysis Options* section, include the **Comparison Target** by clicking **Select...** and navigate to the true overlay *answer.mol2*, which can be found in `my_path_to\CDK2\`.

8. Click **Next** and finally **Run** to start the calculation.

The progress bar helps to monitor the progress of the calculation as the program goes through several stages: *Generating Conformers, Initializing, Generating Overlays, Scoring Overlays, Filtering Overlays, Optimising Overlays and Analysing Overlays.*





Analysis of the results

Once the calculation has completed, it is easy to browse through the solutions using the Hermes *Data Analysis* window that is automatically generated.

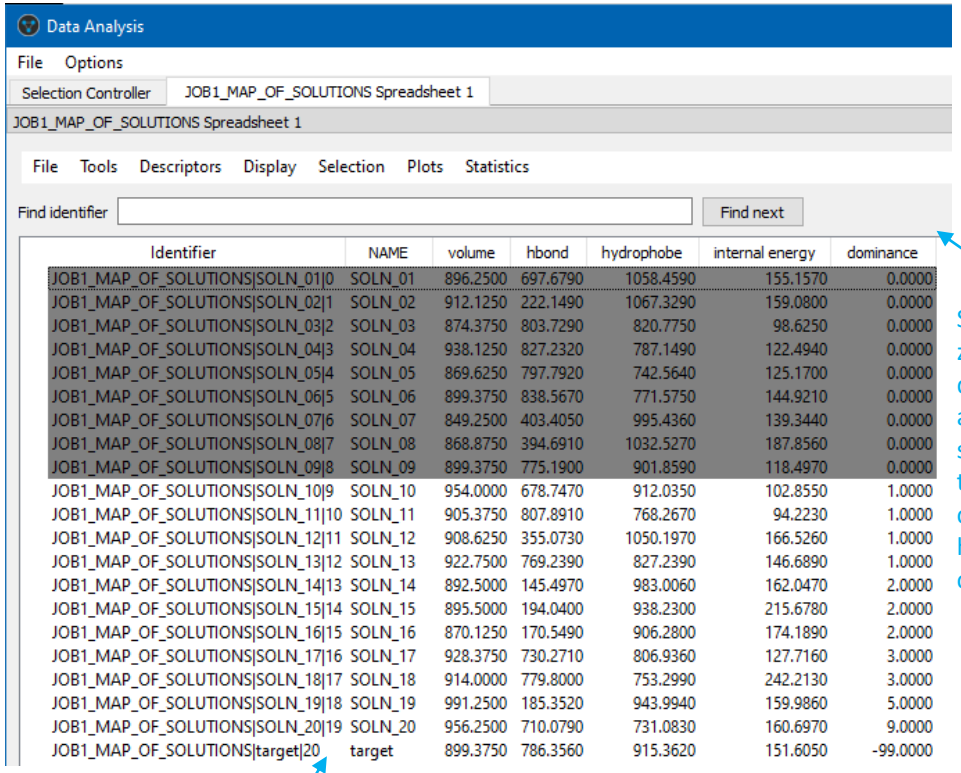
Note that, due to the stochastic nature of the algorithm, results are not reproducible and therefore, here we will discuss an example run.

The solutions can be analysed using the data contained in *JOB1_MAP_OF_SOLUTIONS Spreadsheet 1* in the *Data Analysis* window. The volume, H-bond, hydrophobe coplanarity and internal energy (internal strain) scores of each solution and of the true overlay (target) are given in the four columns of the map file headed *volume*, *hbond*, *hydrophobe* and *internal energy*, respectively.

These scores can be useful, particularly for comparison with the scores that could be obtained for the true crystallographic overlay (i.e. target).

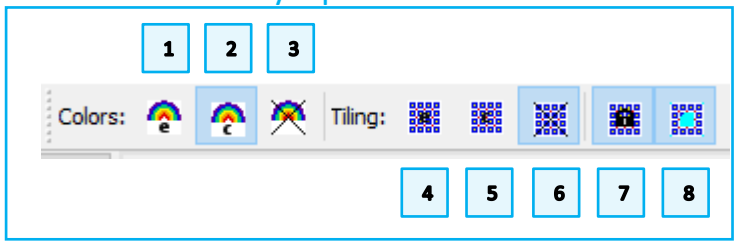
The dominance corresponds to the final Pareto rank of the overlays. Solutions with dominance equal to zero are called non-dominated or equivalent, as they all represent the best compromise in optimising the parameters used to score the solutions. Good solutions can usually be identified by considering first the overlays with zero dominance, and in particular focusing on those that also exhibit low internal strain scores.

Clicking on each solution in the *JOB1_MAP_OF_SOLUTIONS Spreadsheet 1* will display it in the Hermes molecule display area.



The Overlay Options Toolbar in Hermes can be used to help in this analysis.

Colouring options:



1. *Colour entries by rainbow*, where an entry is a given overlay solution.
2. *Colour components by rainbow*, where a component is a given molecule in the overlay.
3. Turn *Rainbow colouring off*.

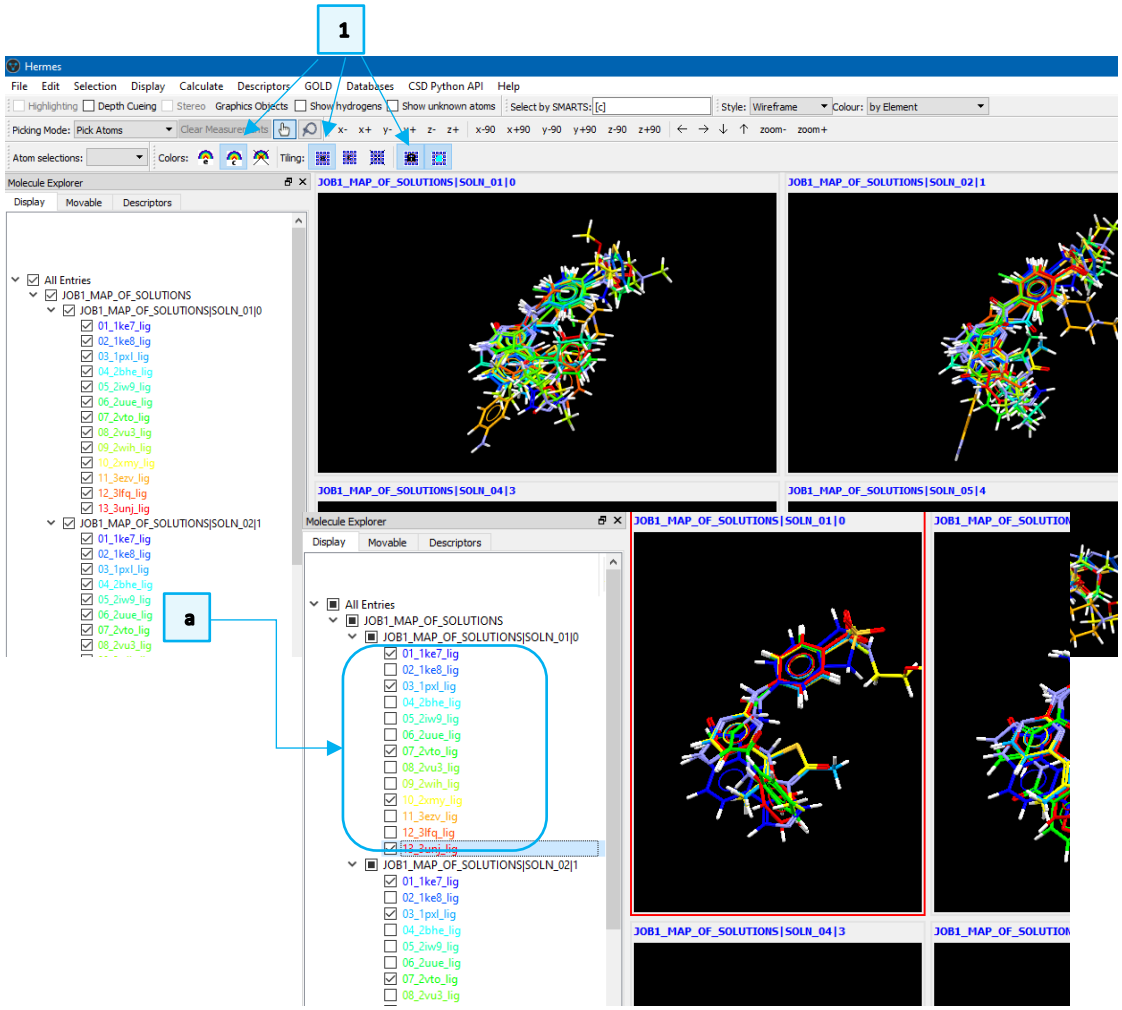
Tiling options:

4. *Tile by entry*, to display one overlay solution per tile.
5. *Tile by component*, to display the same molecule from all selected solutions in each tile.
6. *Remove Tiling*.
7. *Link rotations and translations across all tiles*.
8. *Label tiles with entry identifier*.

1. In order to compare different overlay solutions, click **Tile by entry** and **Link rotations and translations across all tiles**. In addition, be sure that the **Label tiles entry identifier** is activated.

   Each tile will now be populated with one of the selected solutions, and by default, each molecule will be coloured by rainbow. This display helps understanding and comparing the overall shape of the different overlays.

   a. It is possible to clear the display by clicking the tick box next to *JOB1_MAP_OF_SOLUTIONS* under *All Entries* in the *Molecule Explorer* tab. After that, individual molecules can be inspected one after the other to see what feature matches have been found.



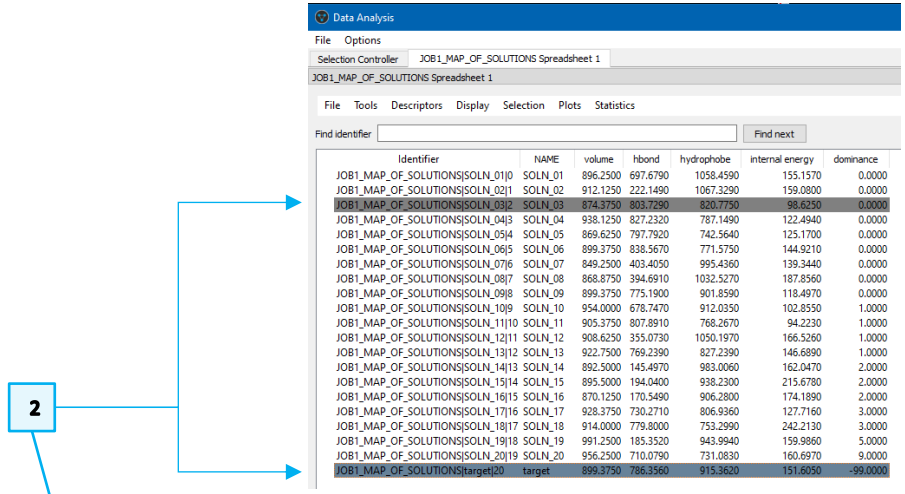
2. In order to assess the performance of the CSD Ligand Overlay to reproduce the superimposition of the 13 input molecules observed experimentally, select one of the non-dominated solutions (i.e. with zero dominance) and the target from the *Data Analysis* table.

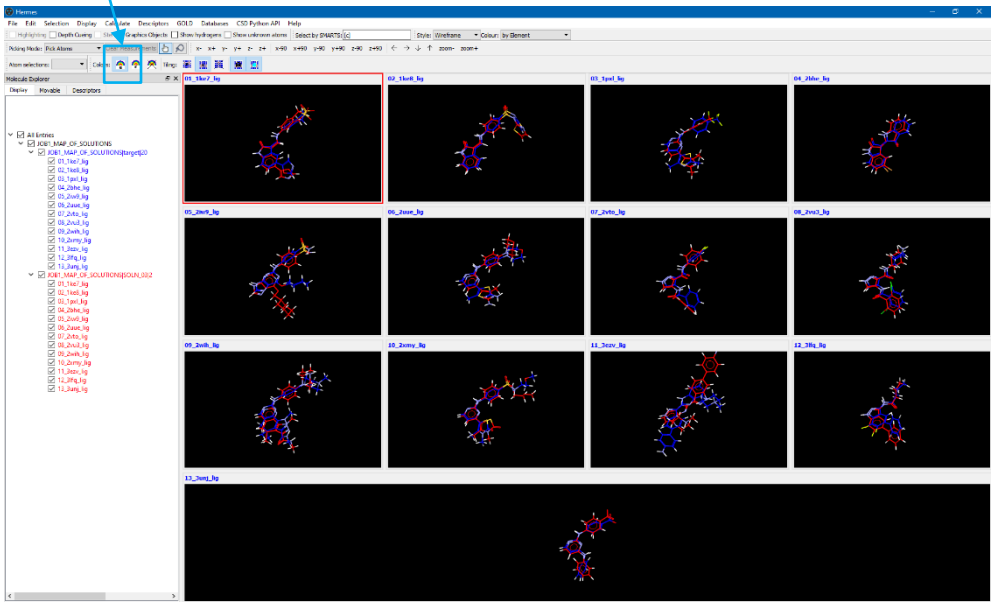
Here we show the comparison of solution_03 with the experimental overlay. Click **Tile by component** to see how the conformation picked by the overlay program compares with the binding conformation of the equivalent ligand in the associated protein crystal structure.

As you can see, the predicted conformations are all close to the equivalent experimental ones. Small deviations are observed in the flexible terminal groups, but this is expected from a ligand-based tool when there is not enough information to support a specific orientation.

We recommend looking at a few diverse overlay solutions, especially in a fully ligand-based approach. They represent alternative, plausible pharmacophore hypotheses that can be used as a query during library screening.

Please bear in mind that here we have done a retrospective analysis where binding site and binding mode information about the protein are available. This can also be a valuable approach to prospectively combine ligand- and structure-based techniques. Therefore, the same concepts and analyses can be applied when using the tool in a real drug design project.





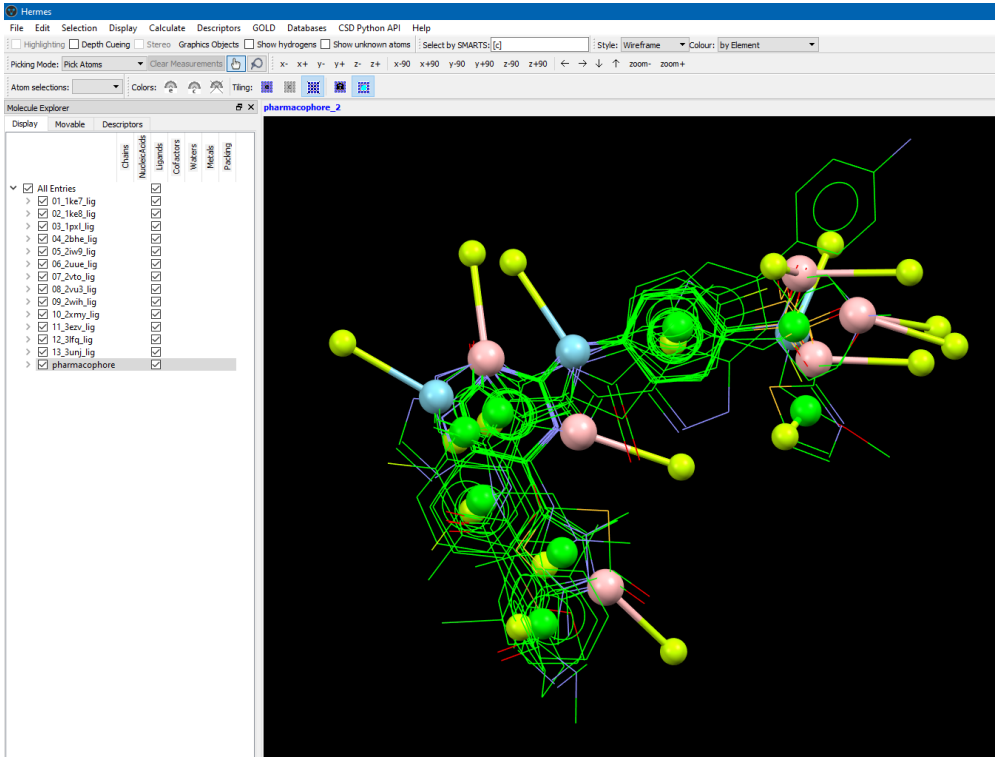
Pharmacophore model

In your output directory `my_path_to\CKD2\job1` you should find these:

Files	**Explanation**
`all_generated.chrm`	File containing five mandatory header lines followed by all 10,000 generated overlays (pre-filtering), stored in chromosome form.
`solutions.chrm`	File containing five mandatory header lines followed by the 20 overlays chosen by the filtering step, stored in chromosome form.
`solution_XX.mol2  solution_XX.sdf`	The 20 solutions chosen by the filtering step and optimised to bring groups into tighter alignment – as a multi mol2 or sdf file.
`pharmacophores\solution_pharm_XX.mol2`	The pharmacophore representations for the 20 solutions.
`solution_target.sdf` `solution_target.mol2`	The true overlay, if indicated in the overlay generation.
`target_overlay_XX.mol2` `target_overlay_XX.sdf`	The 20 optimised solutions, superimposed onto the true overlay if indicated in the overlay generation – as a multi mol2 or sdf file.
`job1_map_of_solutions.csv`	The dominance, consensus coefficients (*x_consensus*, *y_consensus* and *z_consensus*), the pharmacophore coefficients (*x_pharm*, *y_pharm* and *z_pharm*), the superposition coefficients (*x_super*, *y_super* and *z_super*), the three scores (*volume*, *hbond* and *hydrophobe*) and the internal energy (*e*) for all 20 chosen solutions after filtering and optimisation.
`job1_dissimilary_matrix.csv`	The dissimilarity matrices of the 20 chosen solutions based on all three coefficients (consensus, superposition and pharmacophore).
`job1_dissimilarity_matrix_consensus.csv`	The dissimilarity matrices of the 20 chosen solutions based on the consensus coefficients (*x_consensus*, *y_consensus* and *z_consensus*) only.
`job1_dissimilarity_matrix_pharm.csv`	The dissimilarity matrices of the 20 chosen solutions based on the pharmacophore coefficients (*x_pharm*, *y_pharm* and *z_pharm*) only.
`job1_dissimilarity_matrix_super.csv`	The dissimilarity matrices of the 20 chosen solutions based on the superposition coefficients (*x_super*, *y_super* and *z_super*) only.
`scores.csv`	The final *volume*, *hbond* and *hydrophobe* scores for the 20 chosen overlays after filtering and optimisation.
`log.txt`	The log file for the calculation, including the values of all parameters (those left as default and those set) and a summary of all steps and of their duration.

Open a new **Hermes** session and load one of the overlay solutions you are interested in, by clicking on **File > Open** from the main menu.

Also load the corresponding pharmacophore for the chosen solution from `my_path_to\CKD2\job1\pharmacophores`. An example is shown here where the overlay is rendered as a green wireframe and the pharmacophore features are rendered as ball and stick and color-coded based on their type. Acceptor groups are pink, donor groups are cyan and hydrophobic points are green. The projected points are all green.



Conclusion

A number of overlay hypotheses have been produced which in turn can be used as a query for a ligand-based virtual screening experiment.

You should now be familiar with the essential steps of the CSD Ligand Overlay wizard; how to interpret the results with the aid of the tiling options; and how to view the pharmacophore model associated with each solution.

For more information on the advanced settings of the CSD Ligand Overlay please see the main documentation at https://www.ccdc.cam.ac.uk/support-and-resources/ccdcresources/CSD_Ligand_Overlay_User_Guide.pdf

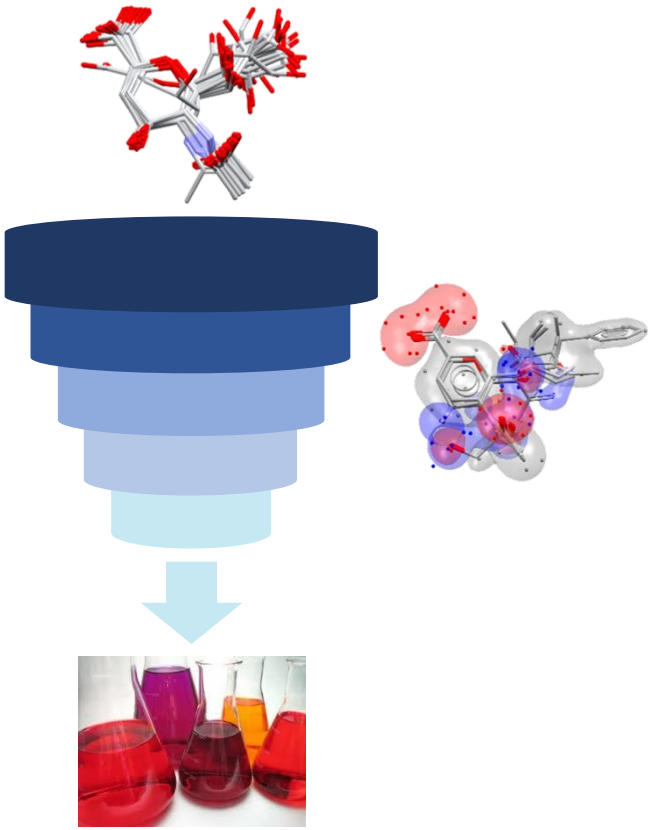
3. Field-based virtual screening

Virtual screening is the computational equivalent of biological screening and is used to score, rank, and/or filter a set of structures by using one or more computational procedures. The CSD Ligand Screener can be used to screen a library of compounds against a pharmacophore query obtained from one or multiple overlaid ligands. The algorithm generalises the 3D pharmacophore definition using atom property fields that are created around the query based on user-defined atom types and potentials. Three main steps are performed:

- Generation of a field potential from the query and creation of fitting points in hotspots.
- Global optimisation of the translation and rotation of each ligand by generating conformer libraries and then fitting the ligand atoms to pre-calculated fitting points.
- Scoring of each screened ligand with numerical gradients.

The methodology requires two input files (*i.e.* a query and the molecules to screen) where all atoms have 3D coordinates.

This tutorial will take you through the steps needed to set up a screening calculation using the CSD Python API. You will learn how to screen a library of known active and decoy molecules against a pharmacophore model generated through the CSD Ligand Overlay. You will calculate several enrichment metrics and plot a ROC curve to assess the success of the model in scoring and ranking the active molecules earlier than the decoys.



Retrieval of actives and decoys

A benchmarking data set for ligand-based virtual screening is publicly available (i.e. DUD LIB VS 1.0) and contains 13 targets including CDK2. The collection of 47 actives and 2070 decoys for this target have been downloaded and are provided with this tutorial in the `my_path_to\CKD2\Screening` subdirectory:

- Actives: *cdk2_clustered_3D_MM.sdf*
- Decoys: *DUD_cdk2_decoys_ID_pass_MWPass_I_MM.sdf*

DUD SUBSETS FOR LIGAND-BASED VIRTUAL SCREENING.
A. Jahn, G. Hinselmann, N. Fechner and A. Zell, "Optimal assignment methods for ligand-based virtual screening", *Journal of Cheminformatics*, 2009, **1**, 14. DOI: 10.1186/1758-2946-1-14.

Library screening

The CSD Python API provides programmatic access to the CSD data and the CSD-System functionality, as well as to features that have never been exposed within an interface. One of these features is the CSD Ligand Screener.
Through Python scripting it is possible to build highly tailored custom applications to help you answer detailed research questions, or to automate frequently performed analysis steps.
Here we will use an example script (i.e. *ligand_based_VS.py*) that performs the three steps described above:

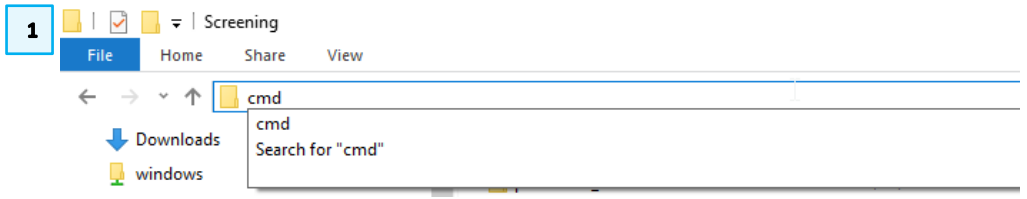
- We will use a solution obtained from the CSD Ligand Overlay to generate our query.
- We will generate knowledge-driven conformers for the library of actives and decoys taken from DUD LIB VS.
- The program will optimise the placement of every conformer onto the fields, but only the top-scoring will be retained and used for ranking.
- The ranked list of scores will be saved to a .csv file (the more negative the scores the better).

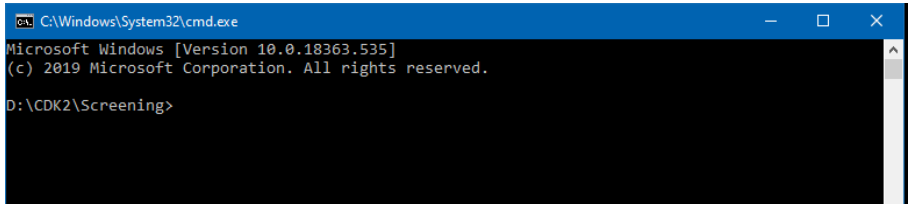
1. **For Windows users:** open File Explorer, go to the directory you are working in (e.g. `D:\CDK2\Screening`) and click on the address bar. Now simply type `cmd` in the address bar. It will open the command prompt with the path to your current folder already set.

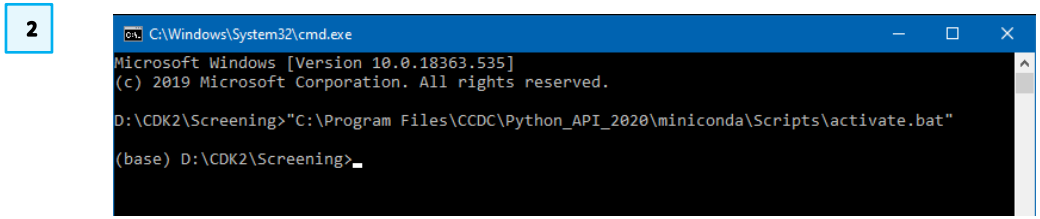
   If you are a macOS user: use a `terminal` to go to your working directory (e.g. `D:/CDK2/Screening`).

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:\Program Files\CCDC\Python_API_2020\miniconda\Scripts\activate.bat`"

If you are a macOS user:

In the terminal type

`source Applications/CCDC/Python_API_2020/miniconda/bin/activate`

As you can see you are now working in a different environment called `(base)`.

3. We are now ready to use the *ligand_based_VS.py* script.

   Type `python ligand_based_VS.py -h` to see all the options supported by the script, what are the required input files and which parameters can be changed by the user.

4. The command to run the script is:

   ```
   python ligand_based_VS.py -q solution_02.sdf -a
   cdk2_clustered_3D_MM.sdf -d
   DUD_cdk2_decoys_ID_pass_MWPass_I_MM.sdf -n 200 -t 3
   ```

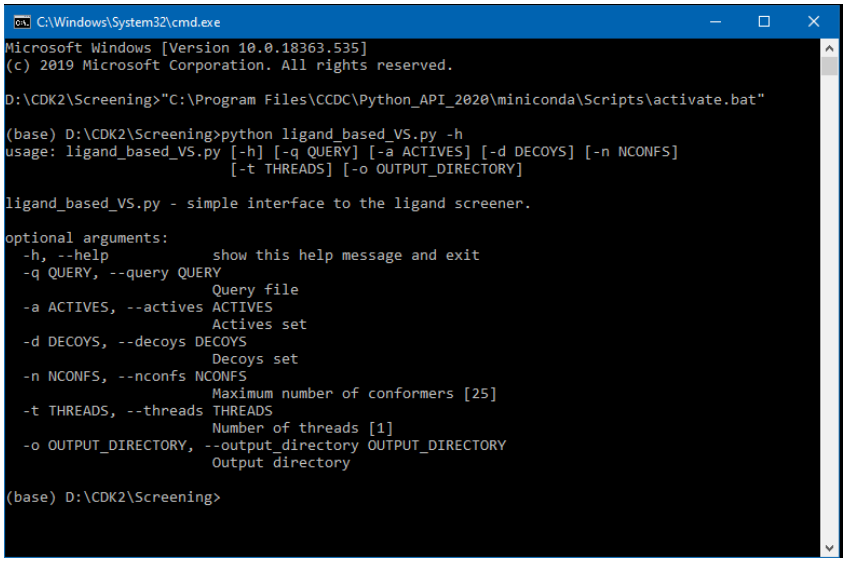
   where:

   - *solution_02.sdf* is an overlay hypothesis we've selected from a previous CSD Ligand Overlay calculation; you may use it or replace it with another promising solution from the CSD Ligand Overlay calculation you have just run;
   - up to 200 conformations will be generated for each active and decoy molecule;
   - the calculation will be distributed over 3 threads.

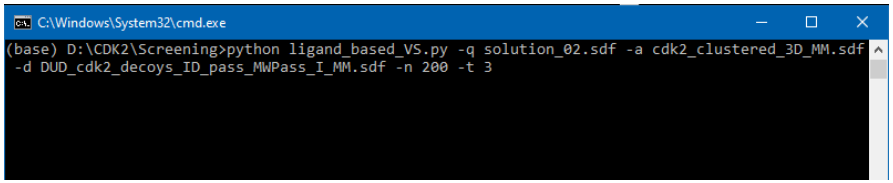
A basic Python script showing the simple lines required for screening is provided in the tutorial folder (*example.py)* and shown on the right.

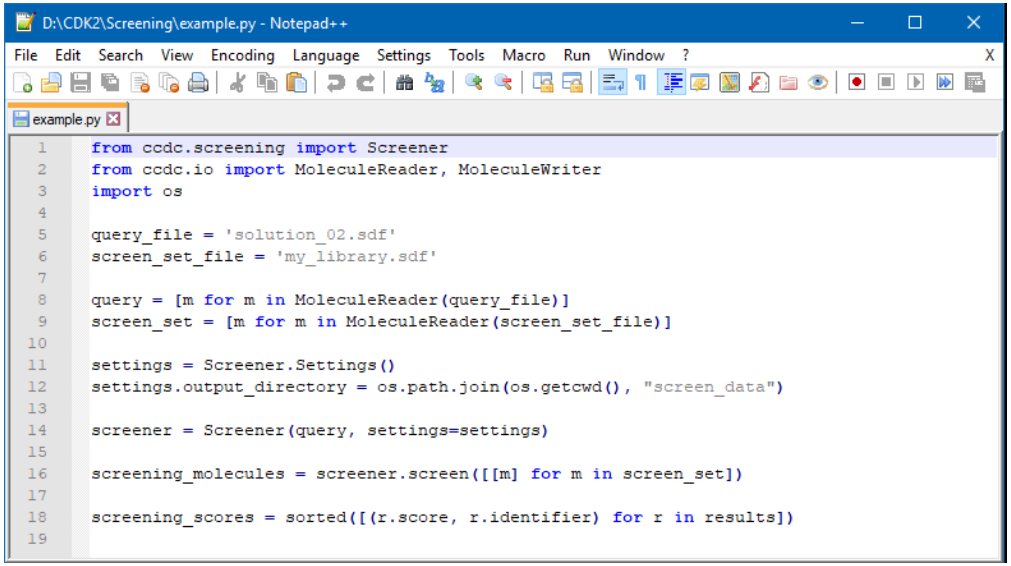
This script shows the basic line for screening the input conformation of the molecules in *my_library.sdf* against an overlay of ligands *solution_02.sdf* as the

3



4





query and write out the resulting *screening_molecules* and *screening_scores* files for further investigation.

Data analysis

The `ligand_based_VS.py` script returns the results into the output directory called *screen_data*. It contains:

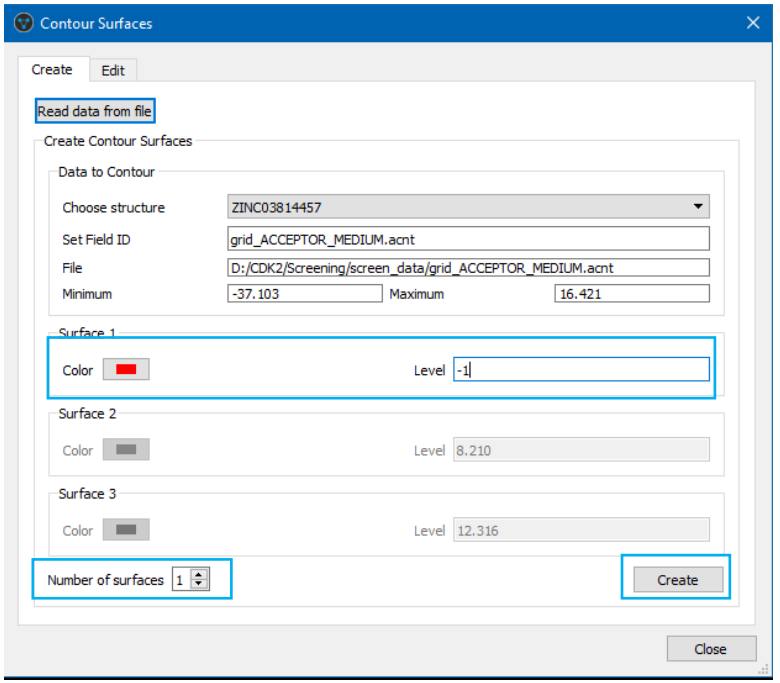
- The grid maps for the donor, acceptor and non-polar fields in .acnt format (e.g. *grid_ACCEPTOR_MEDIUM.acnt*).
- The corresponding fitting points in .mol2 format (e.g. *fitpts_ACCEPTOR_MEDIUM.mol2*).
- The screened libraries (i.e. *actives_screened.mol2* and *decoys_screened.mol2*).
- The ranked list of scores (i.e. *screening_scores.csv*).

Please note that, after launching the script, the grid maps and fitting points will be created almost immediately. It may take up to 15 minutes for the *actives_screened.mol2* file (with the top scoring conformer per screened active molecule) to be returned, and up to 2 hours for the *decoys_screened.mol2* (with the top scoring conformer per screened decoy molecule) and the *screening_scores.csv* to be returned, with times depending on your machine. Please do not close the command prompt when the screening has finished.

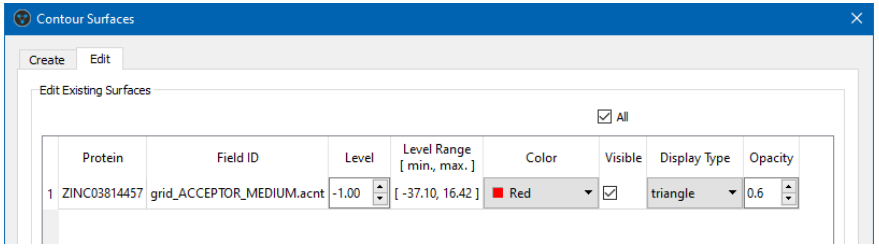
To visualise these fields, launch a new instance of Hermes and load *screen_data/actives_screened.mol2* using **File > Open** menu. Deactivate the tick-box next to *All Entries* in the *Molecule Explorer* window to clear the 3D view in Hermes.

1. Click **Display** from the main menu, then **Contour Surfaces…** In the *Create* tab click **Read data from file** and select *grid_ACCEPTOR_MEDIUM.acnt* from the output directory. Reduce *the Number of surfaces* to 1 and set the Level of *Surface 1* to -1. Click **Create**.

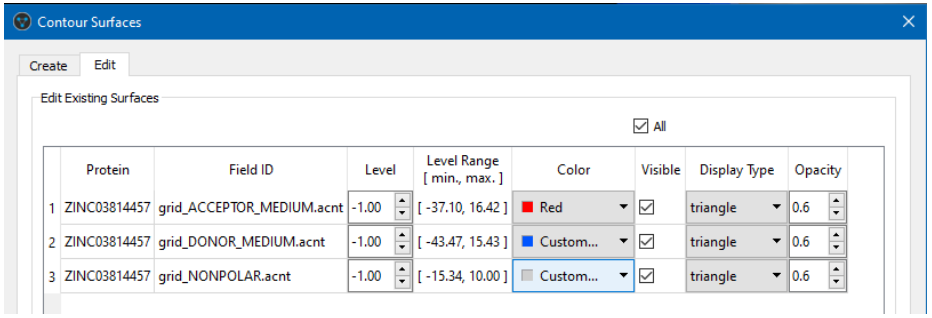
1



2



3



2. In the *Edit* tab change the *Display Type* to triangle and the *Opacity* to 0.6.

3. Repeat the same steps for *grid_DONOR_MEDIUM.acnt* and *grid_NONPOLAR.acnt*, but set the colour to blue and light grey, respectively.

4. The field potentials and the superimposition of the active ligands that have been ranked among the top 1% of the dataset are shown. For this dataset of screened 2117 molecules, this corresponds to the superimposition of the actives found among the best scoring 21 molecules.

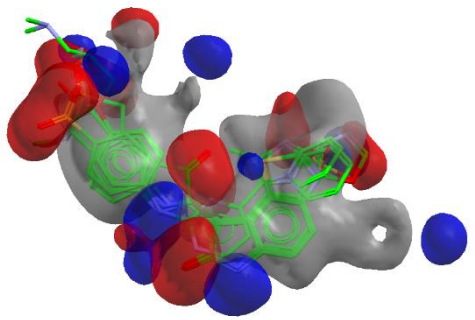
We can calculate the arithmetic weighted ROC enrichment at 1% of false positive fraction as introduced by Jain, A.N. & Nicholls, A. (*J Comput Aided Mol Des* (2008) **22**: 133. doi:10.1007/s10822-008-9196-5). It represents the fraction of actives seen along with the top 1% of known decoys (multiplied by 100). Similarly, awROC enrichments at 0.5%, 2%, and 5% can be calculated, along with the arithmetic weighted version of the AUC (awAUC). We will use the *calc_enrichment_metrics.py* script to calculate these. We will then use the *generate_roc_plot.py* script to generate the ROC plot for this virtual screen study.

5. In the command prompt, please type the command to run the *calc_enrichment_metrics.py* script:
   `python calc_enrichment_metrics.py`

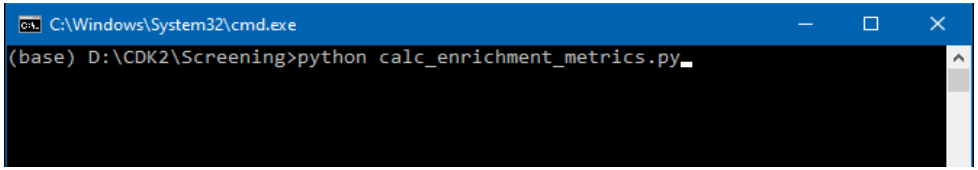
   Please note that you may need to use a text editor such as Notepad++ to edit this *calc_enrichment_metrics.py* script if the path to your *screening_scores.csv* file differs from `D:\CDK2\Screening\screen_data\screening_scores.csv` and if you wish to write the output to a different location than `D:\CDK2\Screening\screen_data\VS_enrichment_metrics.csv`.

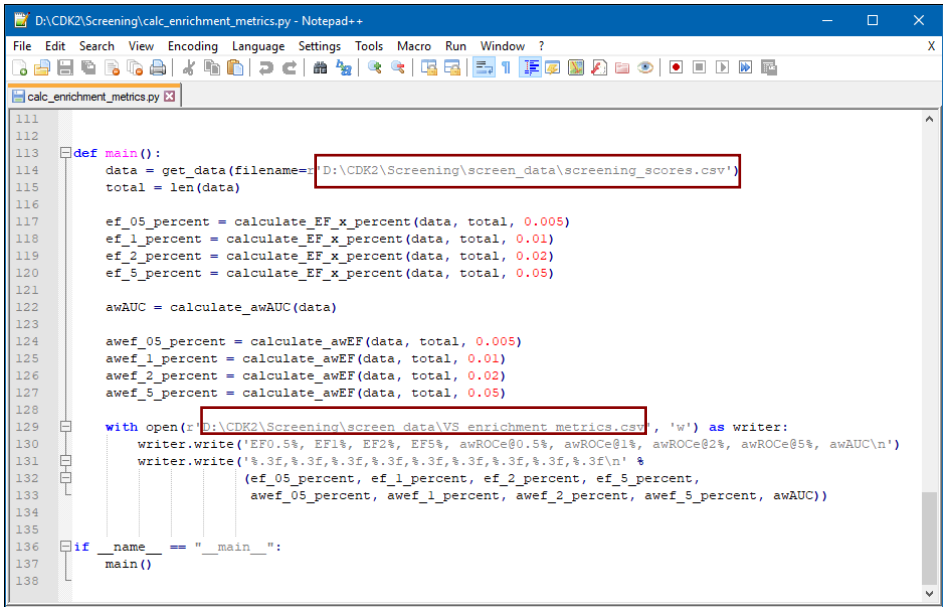
6. In the command prompt, please type the command to run the *generate_roc_plot.py* script:
   `python generate_roc_plot.py screen_data\screening_scores.csv`

4



5

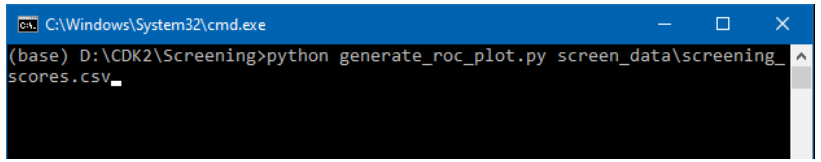




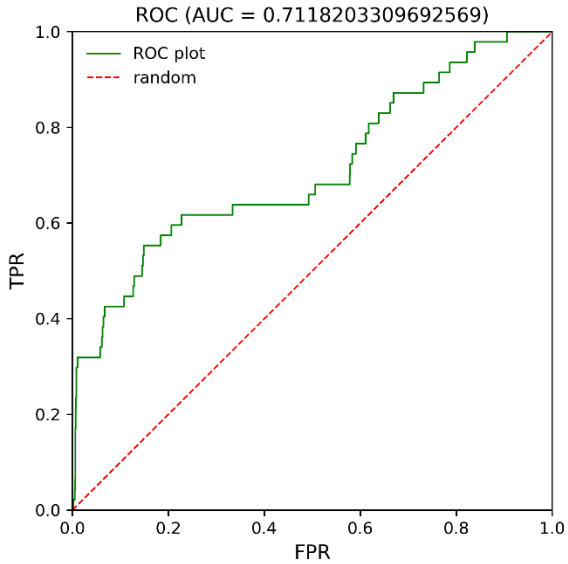
where

`screen_data\screening_scores.csv` is the path to the *screening_scores.csv* output file for this virtual screen study.

6



The ROC curve will be saved as *rocs.png*.



The arithmetic weighted enrichment metrics (which by default were output to `D:\CDK2\Screening\screen_data\VS_enrichment_metrics.csv`) are shown in Table 1, which also contains comparative data collected from the reference paper where the dataset was originally described. The only difference is the query used in all other seven approaches was defined by a single ligand (i.e. PDB entry: 1cpk). This is also possible with the CSD Ligand Screener, but the size and the shape of the molecule will impact the results. In fact, a default distance of 3 Å from the overlay (i.e. the shortest distance to any atom in the overlay) is used to define an excluded volume envelope after which a penalty value is applied if an atom is placed there.

Here we show that exploiting the knowledge from several ligands, if available, can lead to improved performance as they better resemble the shape of the binding pocket.

Table 1. Results of the CSD Ligand Screener compared with other methods. Data taken from DOI: 10.1186/1758-2946-1-14. The best value of each metric is highlighted in blue.

VS metrics	CSD Ligand Screener	DOCK	FieldScreen	MACCS	OAK	OAKFLEX	2SHA	OAAP
awAUC	**0.721**	0.53 ± 0.03	0.44	0.55 ± 0.02	0.55 ± 0.02	0.46 ± 0.02	0.48 ± 0.03	0.53 ± 0.03
awROC enrichment@0.5%	**15.094**	4.0 ± 6.1	7.5	9.4 ± 1.8	9.4 ± 1.8	9.4 ± 1.8	9.4 ± 3.7	24.1 ± 4.7
awROC enrichment@1%	**26.183**	9.5 ± 2.7	3.8	4.9 ± 1.0	6.5 ± 1.5	4.9 ± 1.5	11.1 ± 2.4	15.7 ± 2.8
awROC enrichment@2%	**16.942**	7.0 ± 1.4	1.9	2.5 ± 0.5	5.5 ± 1.2	4.8 ± 1.1	7.9 ± 1.4	8.6 ± 1.3
awROC enrichment@5%	**6.777**	2.8 ± 0.6	0.8	2.6 ± 0.6	2.6 ± 0.4	2.6 ± 0.4	3.5 ± 0.5	3.5 ± 0.7

Additional notes

- Tautomerism is crucial in protein-ligand interactions. Thus, virtual screening libraries should include multiple tautomeric forms of all molecules to lower the risk of losing an important hit. Different protonation states should also be enumerated. However, the final list of ranked scores should only contain the protomer or tautomer with the better score.
- For the sake of transparency and comparison, here we've used the publicly available dataset, but we strongly recommend that in real drug discovery projects all tautomers and protomers are enumerated before performing any structure- or ligand-based virtual screening study.

Conclusion

A retrospective virtual screening study of known active and decoy molecules has been carried out. Considering the high enrichment values, although CDK2 is still a challenging case for ligand-based methods (*J. Chem. Inf. Model.*, 2008, **48** (11), 2108–2117, DOI: 10.1021/ci800110p), we can be more confident in using the selected overlay hypothesis and the CSD Ligand Screener for a prospective screening of unknown binders.

You should now be familiar with how to setup a ligand-based virtual screening study using the CSD Python API, how to visualise the potential fields in Hermes, and how to analyse the ranked list of scores.