Ligand-Based Virtual Screening

(PYAPI-003)

2025.1 CSD Release

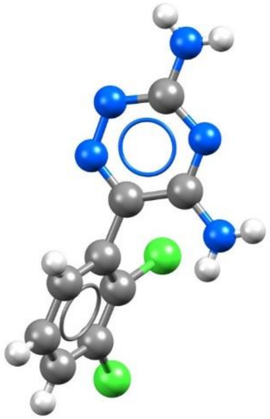


Table of Contents

CCDC
advancing structural science

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”.

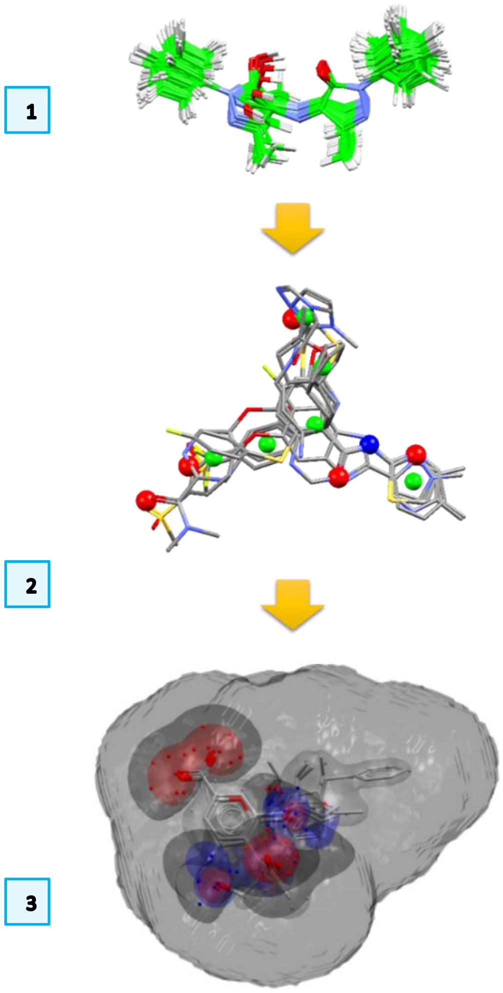
A pharmacophore model can be generated by superposing a set of active molecules that are assumed to bind to the same target with the same binding mode and extracting common chemical features that are essential for their bioactivity. Such a model can then be used to virtually screen libraries of compounds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

Learning Outcomes

In this tutorial, you will learn how to perform a ligand-based virtual screening using a suite of knowledge-based tools. After completing this workshop, you should be able to:

1. Generate ensembles of conformers for a set of known CDK2 inhibitors.
2. Use the CSD-Ligand Overlay program and a pharmacophore model will be selected for step 3.
3. Screen a library of active and inactive (or decoy) compounds against the selected overlay solution and calculate enrichment metrics.

This workshop will take approximately **2** hours to be completed. The words In *Blue Italic* in the text are reported in the Glossary at the end of this handout.



Pre-required Skills

Familiarity with the Hermes interface is important; you can access the Visualization in Hermes 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

In the Screening folder, you will find several scripts used in this handout, as well as input files required to run the scripts.

`generate_confs.py` is written using the CSD Python API, documented here:

https://downloads.ccdc.cam.ac.uk/documentation/API/modules/conformer_api.html

`ligand_based_vs.py` is taken from the CSD Python API cookbook:

https://downloads.ccdc.cam.ac.uk/documentation/API/cookbook_examples/virtual_screening_examples.html

`calc_enrichment_metrics.py` has been custom written for this workshop based on *J Comput Aided Mol Des* (2008) **22**: 133. doi:10.1007/s10822-008-9196-5.

`generate_roc_plot.py` has been written using the StatisticalDescriptors module in the CSD Python API, documented here:

https://downloads.ccdc.cam.ac.uk/documentation/API/modules/descriptors_api.html#ccdc.descriptors.StatisticalDescriptors

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.

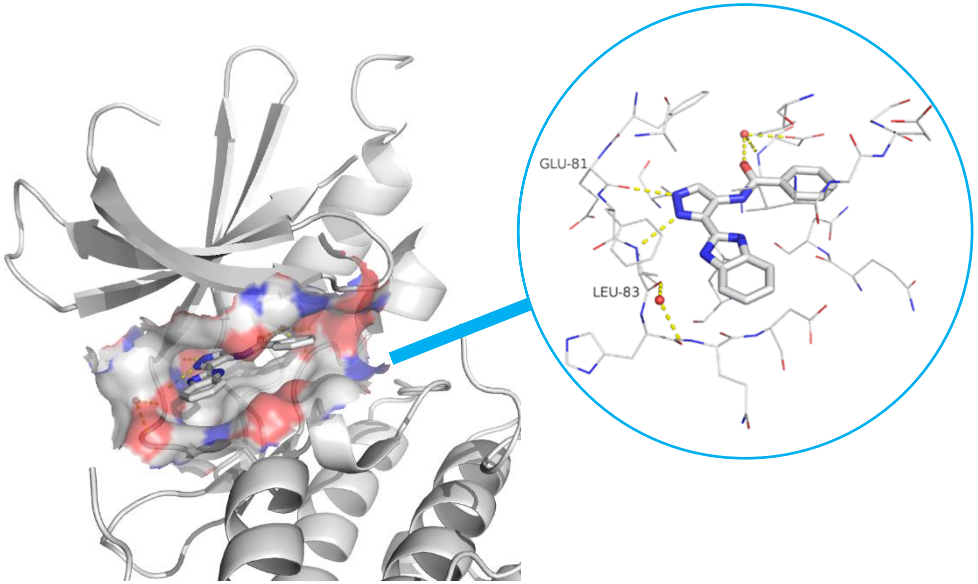
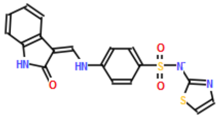
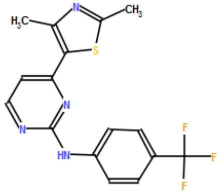


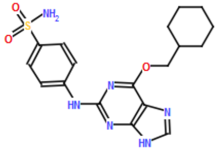
Figure1. CDK2 in complex with a fragment-like molecule. Hydrogen bond interactions are shown as dashed yellow lines.



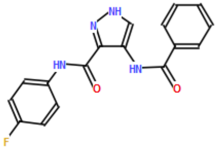
1ke8



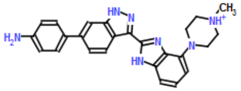
1pxl



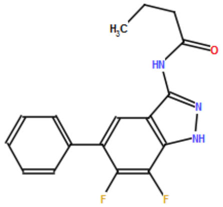
2iw9



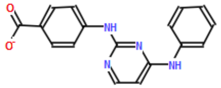
2vto



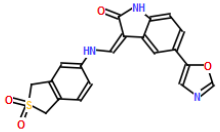
3ezv



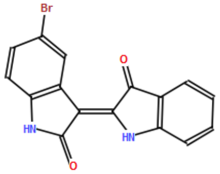
3lfq



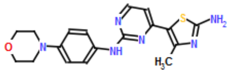
3unj



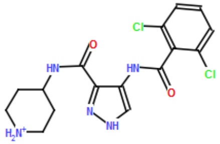
1ke7



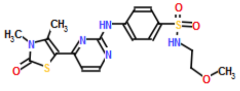
2bhe



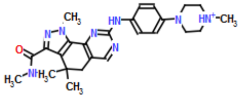
2uue



2vu3



2xmy



2wih

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.

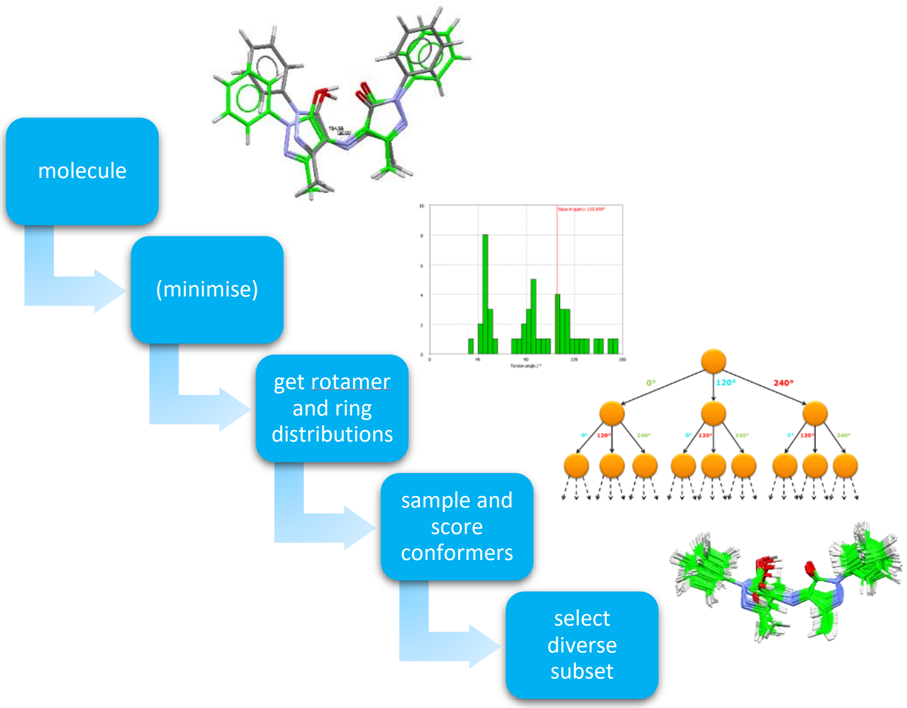
To avoid any bias from the experimentally observed conformation for this example, we regenerated 3D coordinates for the 13 molecules using the program Corina.

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 the CSD Python API method to generate the conformers of the 13 CDK2 inhibitors. Tutorials for the command line utility and Mercury interface can be found in self-guided workshop CONF-002 which can be found here: https://www.ccdc.cam.ac.uk/community/training-and-learning/workshop-materials/csd-discovery-workshops/ .

Generating Conformers Workflow



molecule

(minimise)

get rotamer and ring distributions

sample and score conformers

select diverse subset

Generating conformers with the CSD Python API

The CSD Conformer Generator functionality is 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 Section 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`).

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.

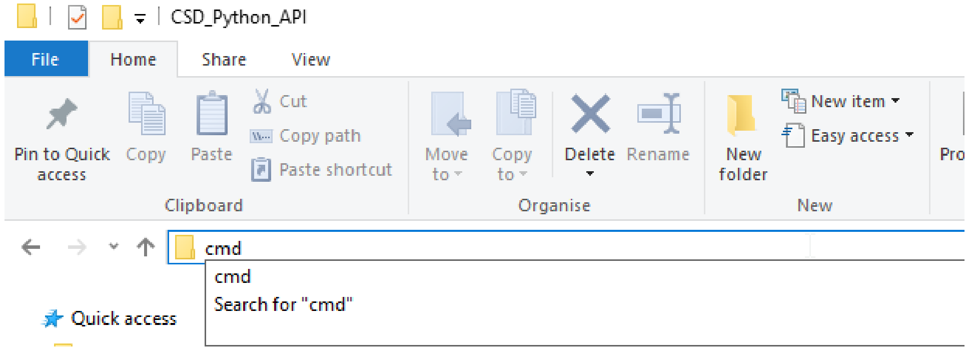
   If you are a Windows user:
   In the command prompt type
   `“C:\path\to\CCDC\ccdc-software\csd-python-api\miniconda\Scripts\activate”`

   If you are a macOS user:
   In the terminal type:
   `source Applications/CCDC/ccdc-software/csd-python-api/miniconda/bin/activate`

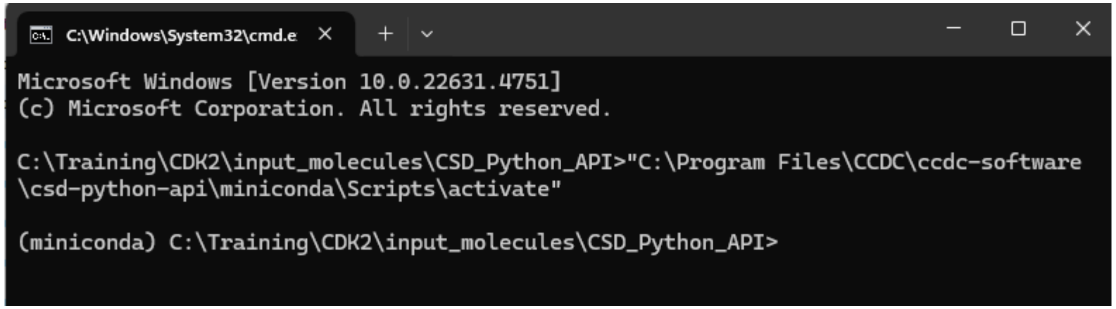
   The above command will run the CSD Python API activation script. As you can see you are now working in a different environment called `(miniconda)`.

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.

1



2



```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Training\CDK2\input_molecules\CSD_Python_API>"C:\Program Files\CCDC\ccdc-software
\csd-python-api\miniconda\Scripts\activate"

(miniconda) C:\Training\CDK2\input_molecules\CSD_Python_API>
```

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:

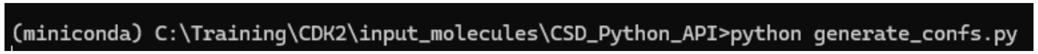
- **`max_conformers`**
  Maximum number of conformers to generate.
- **`max_unusual_torsions`**
  Number of unusual torsions allowed per conformer.
- **`normalised_score_threshold`**
  Maximum deviation from the theoretically achievable normalised conformer probability (0=”best”, 1=”worst”).
- **`superimpose_conformers_onto_reference`**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

The generated conformers will be placed in your working folder once the script concludes, Fig 4.

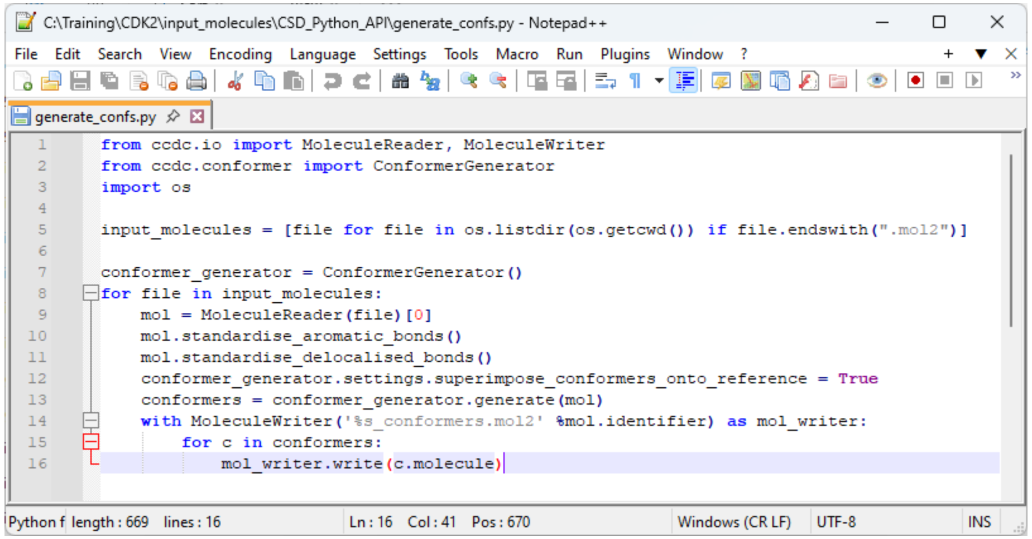
Conclusion

Using 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.

3



```
(miniconda) C:\Training\CDK2\input_molecules\CSD_Python_API>python generate_confs.py
```


C:\Training\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 f length : 669 lines : 16 Ln : 16 Col : 41 Pos : 670 Windows (CR LF) UTF-8 INS

```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)
```

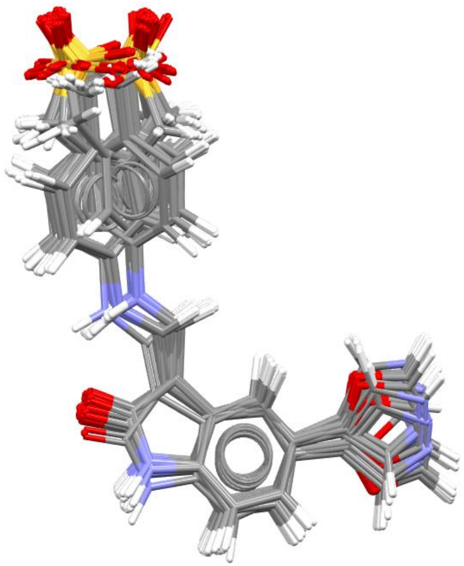


Figure 4: Ensemble of conformers generated for the molecule *1ke7_lig_LS3.mol2* using the CSD Python API. The input molecule was used as reference to superimpose the resulting conformations on all heavy atoms.

Overlaying ligands

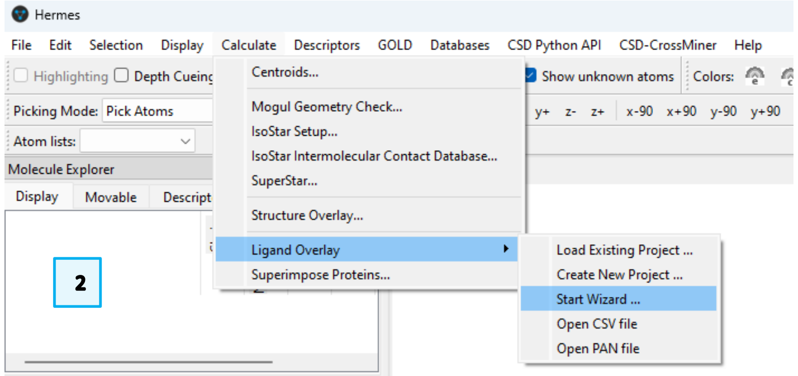
The CSD Ligand Overlay program aims to overlay sets of flexible molecules that are assumed to be *ligands* of the same protein. The molecules to be overlaid are divided into *features* such as hydrogen-bond donors and acceptors, and hydrophobic groups.

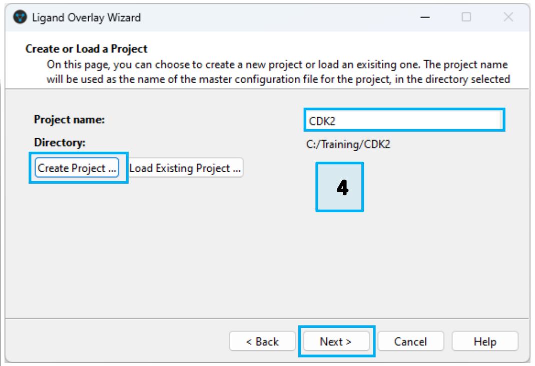
In the absence of any protein-structure information, there may be multiple overlays that present legitimate hypotheses for binding.

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.



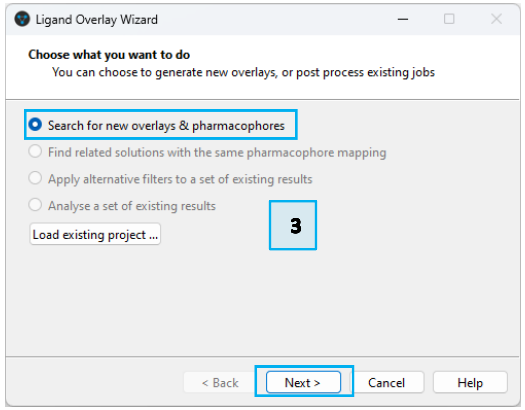


Create or Load a Project

On this page, you can choose to create a new project or load an existing one. The project name will be used as the name of the master configuration file for the project, in the directory selected

Project name: CDK2

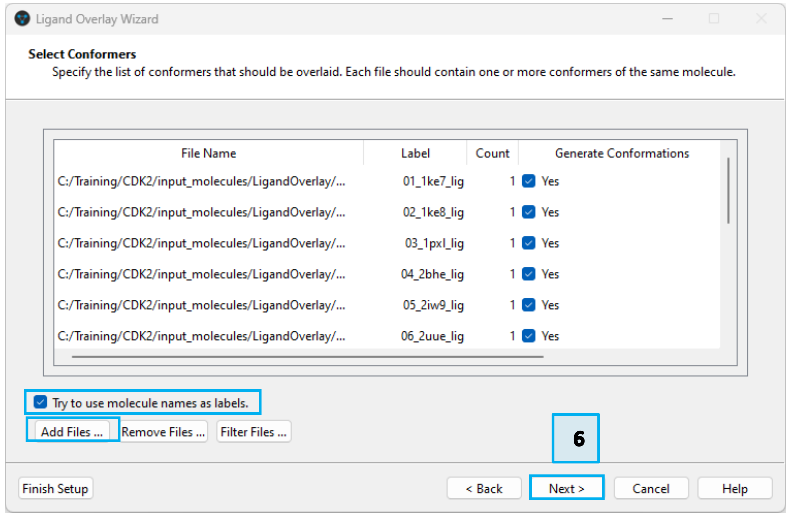
Directory: C:/Training/CDK2



Choose what you want to do

You can choose to generate new overlays, or post process existing jobs

- ● Search for new overlays & pharmacophores
- ○ Find related solutions with the same pharmacophore mapping
- ○ Apply alternative filters to a set of existing results
- ○ Analyse a set of existing results



Select Conformers

Specify the list of conformers that should be overlaid. Each file should contain one or more conformers of the same molecule.

File Name	Label	Count	Generate Conformations
C:/Training/CDK2/input_molecules/LigandOverlay/...	01_1ke7_lig	1	☑ Yes
C:/Training/CDK2/input_molecules/LigandOverlay/...	02_1ke8_lig	1	☑ Yes
C:/Training/CDK2/input_molecules/LigandOverlay/...	03_1pxl_lig	1	☑ Yes
C:/Training/CDK2/input_molecules/LigandOverlay/...	04_2bhe_lig	1	☑ Yes
C:/Training/CDK2/input_molecules/LigandOverlay/...	05_2iw9_lig	1	☑ Yes
C:/Training/CDK2/input_molecules/LigandOverlay/...	06_2uue_lig	1	☑ Yes

☑ Try to use molecule names as labels.

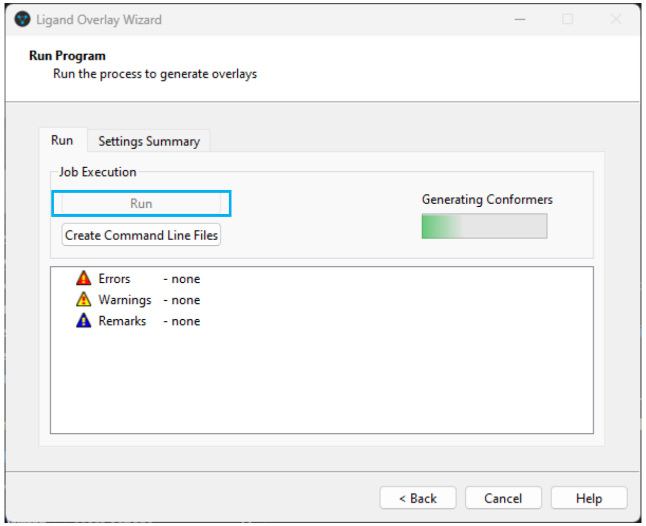
6. Click **Next** to go to the *Select Parameter Settings* dialog. As the default parameters are appropriate for this tutorial we could just click **Finish Setup** and start the calculation. However, as we are performing a retrospective study of known CDK2 inhibitors for which the crystal structures 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.*

8



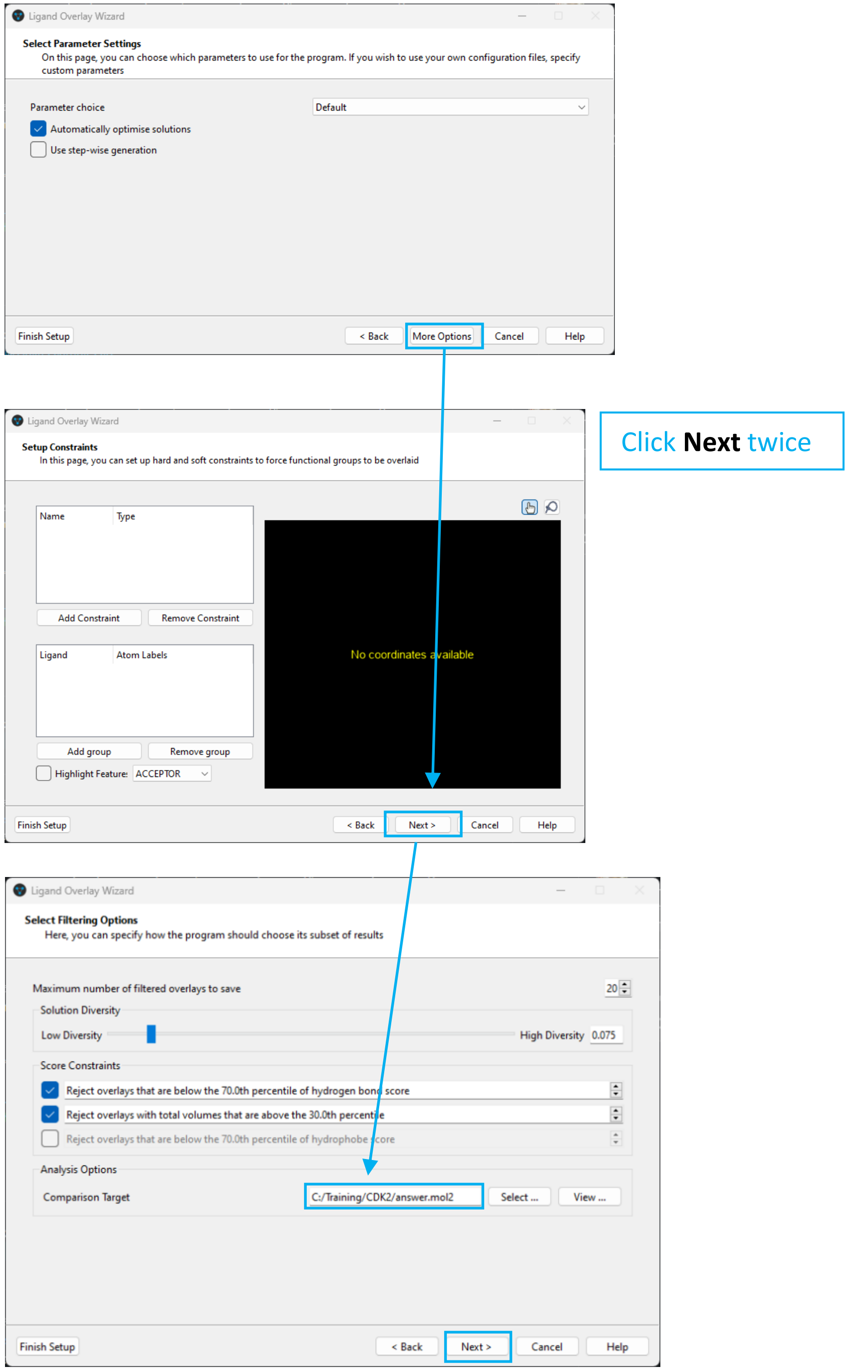
Run Program
Run the process to generate overlays

Job Execution

Generating Conformers

Errors - none
Warnings - none
Remarks - none

7



Select Parameter Settings
On this page, you can choose which parameters to use for the program. If you wish to use your own configuration files, specify custom parameters

Parameter choice: Default
☑ Automatically optimise solutions
□ Use step-wise generation

Click **Next** twice

Setup Constraints
In this page, you can set up hard and soft constraints to force functional groups to be overlaid

□ Highlight Feature: ACCEPTOR

Select Filtering Options
Here, you can specify how the program should choose its subset of results

Maximum number of filtered overlays to save: 20

Solution Diversity
Low Diversity — High Diversity 0.075

Score Constraints
☑ Reject overlays that are below the 70.0th percentile of hydrogen bond score
☑ Reject overlays with total volumes that are above the 30.0th percentile
□ Reject overlays that are below the 70.0th percentile of hydrophobe score

Analysis Options
Comparison Target: C:/Training/CDK2/answer.mol2

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.

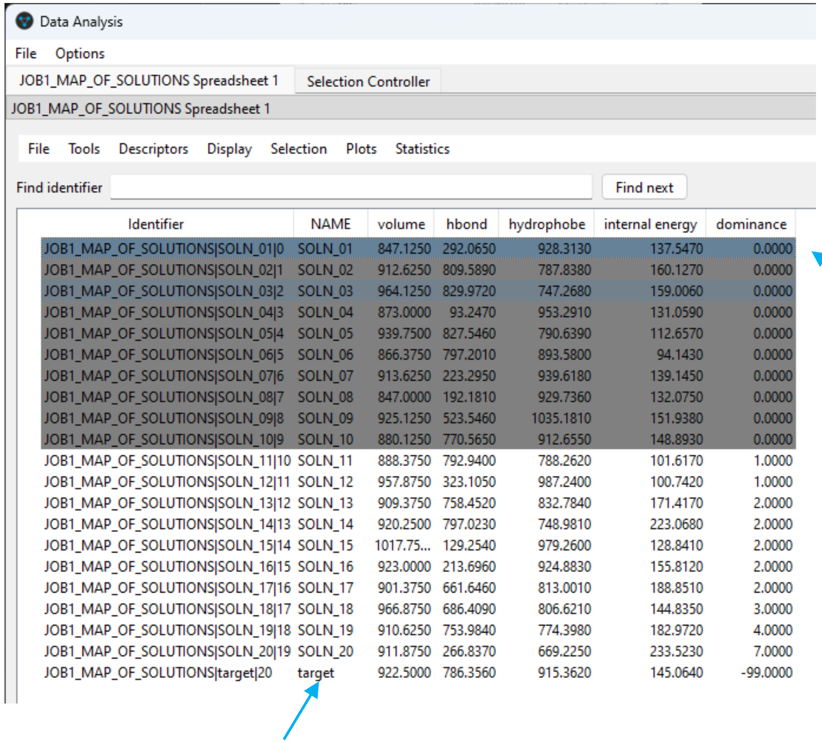
Note: due to the stochastic nature of the algorithm, exact results are not reproducible and therefore, here we will discuss an example run.

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.


Data Analysis
File Options
JOB1_MAP_OF_SOLUTIONS Spreadsheet 1
Selection Controller
JOB1_MAP_OF_SOLUTIONS Spreadsheet 1
File Tools Descriptors Display Selection Plots Statistics
Find identifier
Find next

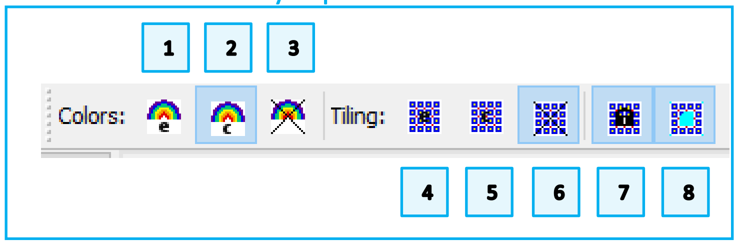
Identifier	NAME	volume	hbond	hydrophobe	internal energy	dominance
JOB1_MAP_OF_SOLUTIONS|SOLN_01|0	SOLN_01	847.1250	292.0650	928.3130	137.5470	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_02|1	SOLN_02	912.6250	809.5890	787.8380	160.1270	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_03|2	SOLN_03	964.1250	829.9720	747.2680	159.0060	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_04|3	SOLN_04	873.0000	93.2470	953.2910	131.0590	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_05|4	SOLN_05	939.7500	827.5460	790.6390	112.6570	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_06|5	SOLN_06	866.3750	797.2010	893.5800	94.1430	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_07|6	SOLN_07	913.6250	223.2950	939.6180	139.1450	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_08|7	SOLN_08	847.0000	192.1810	929.7360	132.0750	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_09|8	SOLN_09	925.1250	523.5460	1035.1810	151.9380	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_10|9	SOLN_10	880.1250	770.5650	912.6550	148.8930	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_11|10	SOLN_11	888.3750	792.9400	788.2620	101.6170	1.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_12|11	SOLN_12	957.8750	323.1050	987.2400	100.7420	1.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_13|12	SOLN_13	909.3750	758.4520	832.7840	171.4170	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_14|13	SOLN_14	920.2500	797.0230	748.9810	223.0680	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_15|14	SOLN_15	1017.75...	129.2540	979.2600	128.8410	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_16|15	SOLN_16	923.0000	213.6960	924.8830	155.8120	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_17|16	SOLN_17	901.3750	661.6460	813.0010	188.8510	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_18|17	SOLN_18	966.8750	686.4090	806.6210	144.8350	3.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_19|18	SOLN_19	910.6250	753.9840	774.3980	182.9720	4.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_20|19	SOLN_20	911.8750	266.8370	669.2250	233.5230	7.0000
JOB1_MAP_OF_SOLUTIONS|target|20	target	922.5000	786.3560	915.3620	145.0640	-99.0000

Solutions with zero dominance automatically selected when the calculation has completed

True crystallographic overlay

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

Overlay Options Toolbar



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:

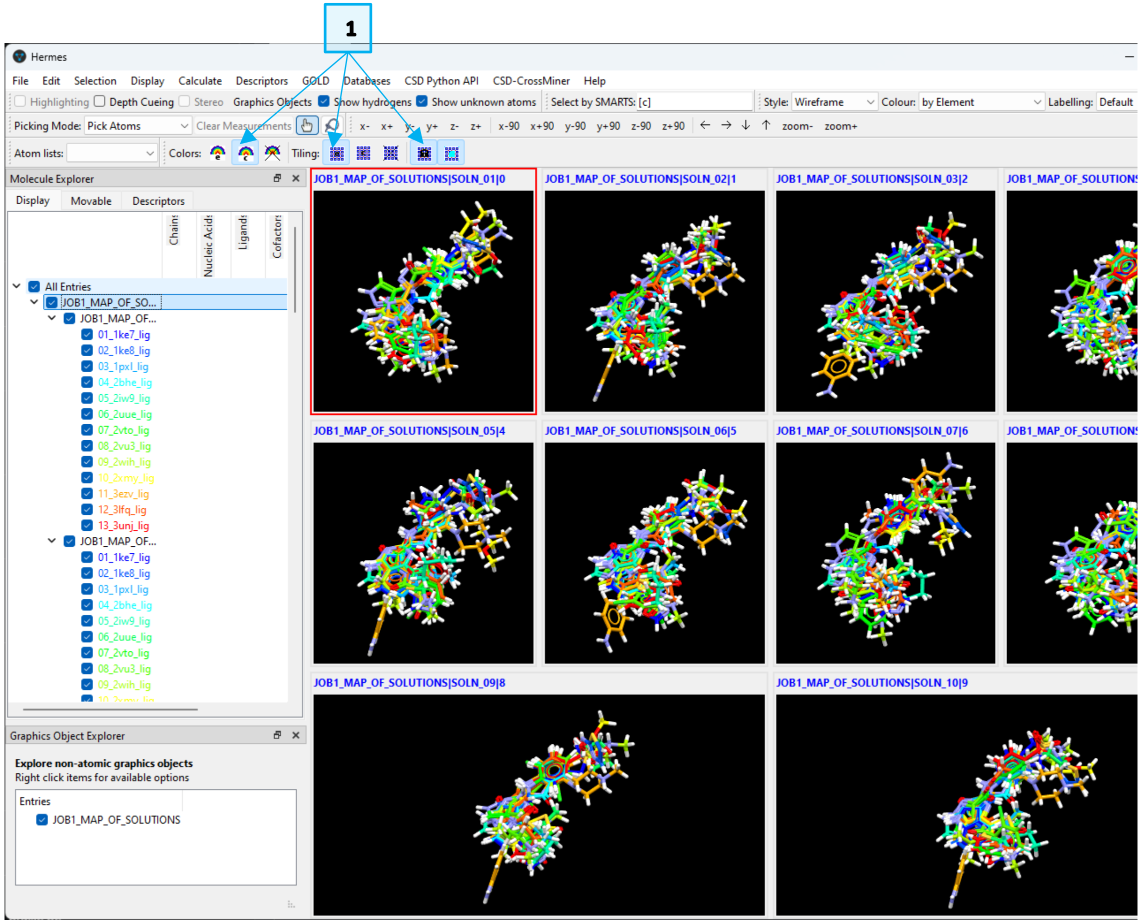
4. *Tile by entry*, to display one overlay solution per tile.
5. *Tile by component*, to display the same molecule from all selected solutions in each tile.

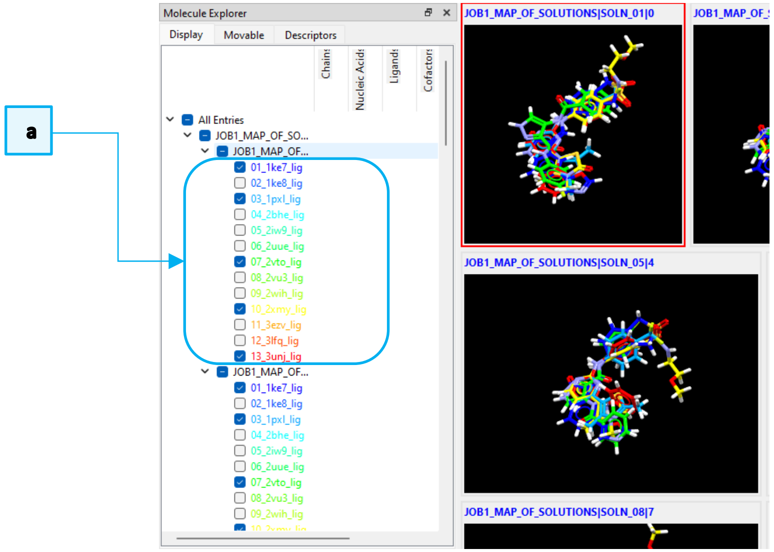
6. *Remove Tiling*.
7. *Link rotations and translations across all tiles*.
8. *Label tiles with entry identifier*.

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 **Label tiles with 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.


1
Hermes
File Edit Selection Display Calculate Descriptors GOLD Databases CSD Python API CSD-CrossMiner Help
Tiling:
Molecule Explorer
Display Movable Descriptors
All Entries
JOB1_MAP_OF_SO...
Graphics Object Explorer
Explore non-atomic graphics objects
Right click items for available options
Entries
JOB1_MAP_OF_SOLUTIONS
JOB1_MAP_OF_SOLUTIONS|SOLN_01|0
JOB1_MAP_OF_SOLUTIONS|SOLN_02|1
JOB1_MAP_OF_SOLUTIONS|SOLN_03|2
JOB1_MAP_OF_SOLUTIONS|SOLN_05|4
JOB1_MAP_OF_SOLUTIONS|SOLN_06|5
JOB1_MAP_OF_SOLUTIONS|SOLN_07|6
JOB1_MAP_OF_SOLUTIONS|SOLN_09|8
JOB1_MAP_OF_SOLUTIONS|SOLN_10|9



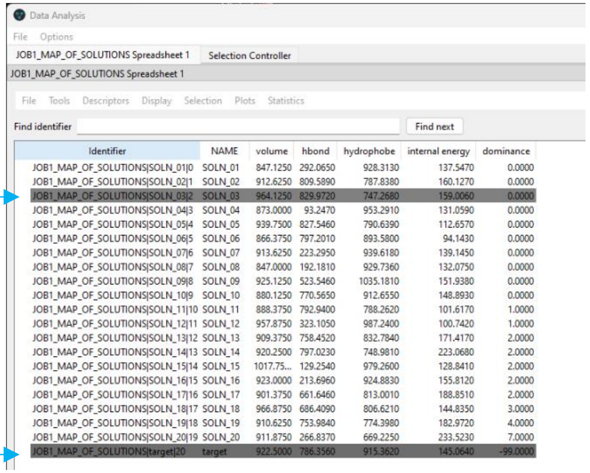
2. In order to assess the performance of the CSD Ligand Overlay to reproduce the superimposition of the 13 input molecules observed experimentally, select one of the non-dominated solutions (i.e. with zero dominance) and the target from the *Data Analysis* table.

   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.

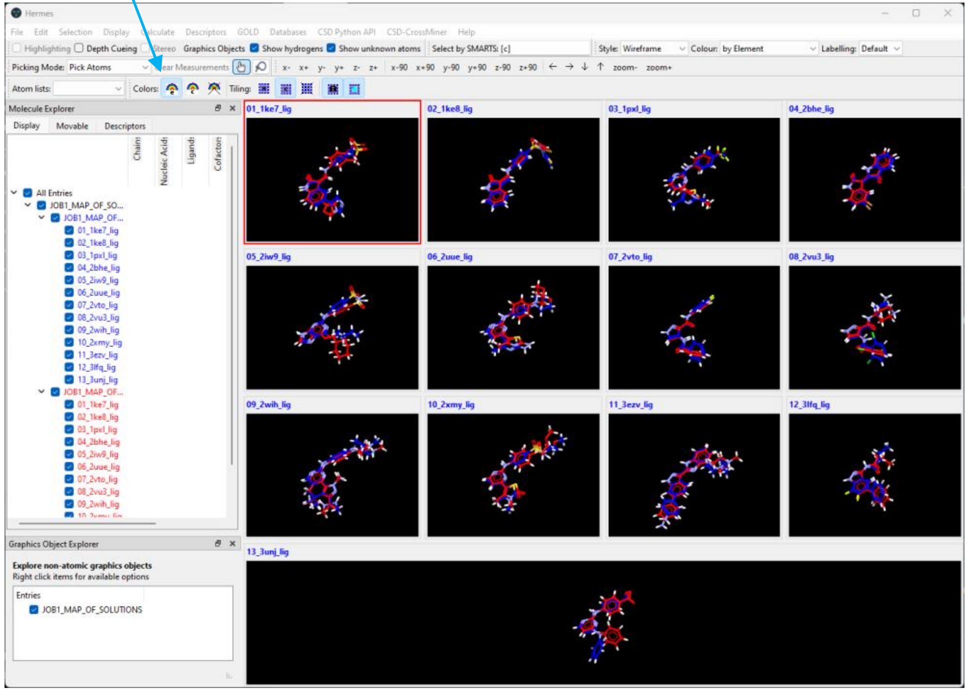
We recommend looking at a few diverse overlay solutions, especially in a fully ligand-based approach. They represent alternative, plausible pharmacophore hypotheses that can be used as a query during library screening.

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.


Data Analysis
File Options
JOB1_MAP_OF_SOLUTIONS Spreadsheet 1
Selection Controller
JOB1_MAP_OF_SOLUTIONS Spreadsheet 1
File Tools Descriptors Display Selection Plots Statistics
Find identifier
Find next

Identifier	NAME	volume	hbond	hydrophobe	internal energy	dominance
JOB1_MAP_OF_SOLUTIONS|SOLN_01|0	SOLN_01	847.1250	292.0650	928.3130	137.5470	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_02|1	SOLN_02	912.6250	809.5890	787.8380	160.1270	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_03|2	SOLN_03	964.1250	829.9720	747.2680	159.0060	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_04|3	SOLN_04	873.0000	93.2470	953.2910	131.0590	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_05|4	SOLN_05	939.7500	827.5460	790.6390	112.6570	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_06|5	SOLN_06	866.3750	797.2010	893.5800	94.1430	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_07|6	SOLN_07	913.6250	223.2950	939.6180	139.1450	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_08|7	SOLN_08	847.0000	192.1810	929.7360	132.0750	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_09|8	SOLN_09	925.1250	523.5460	1035.1810	151.9380	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_10|9	SOLN_10	880.1250	770.5650	912.6550	148.8930	0.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_11|10	SOLN_11	888.3750	792.9400	788.2620	101.6170	1.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_12|11	SOLN_12	957.8750	323.1050	987.2400	100.7420	1.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_13|12	SOLN_13	909.3750	758.4520	832.7840	171.4170	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_14|13	SOLN_14	920.2500	797.0230	748.9810	223.0680	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_15|14	SOLN_15	1017.75...	129.2540	979.2600	128.8410	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_16|15	SOLN_16	923.0000	213.6960	924.8830	155.8120	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_17|16	SOLN_17	901.3750	661.6460	813.0010	188.8510	2.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_18|17	SOLN_18	966.8750	686.4090	806.6210	144.8350	3.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_19|18	SOLN_19	910.6250	753.9840	774.3980	182.9720	4.0000
JOB1_MAP_OF_SOLUTIONS|SOLN_20|19	SOLN_20	911.8750	266.8370	669.2250	233.5230	7.0000
JOB1_MAP_OF_SOLUTIONS|target|20	target	922.5000	786.3560	915.3620	145.0640	-99.0000

2

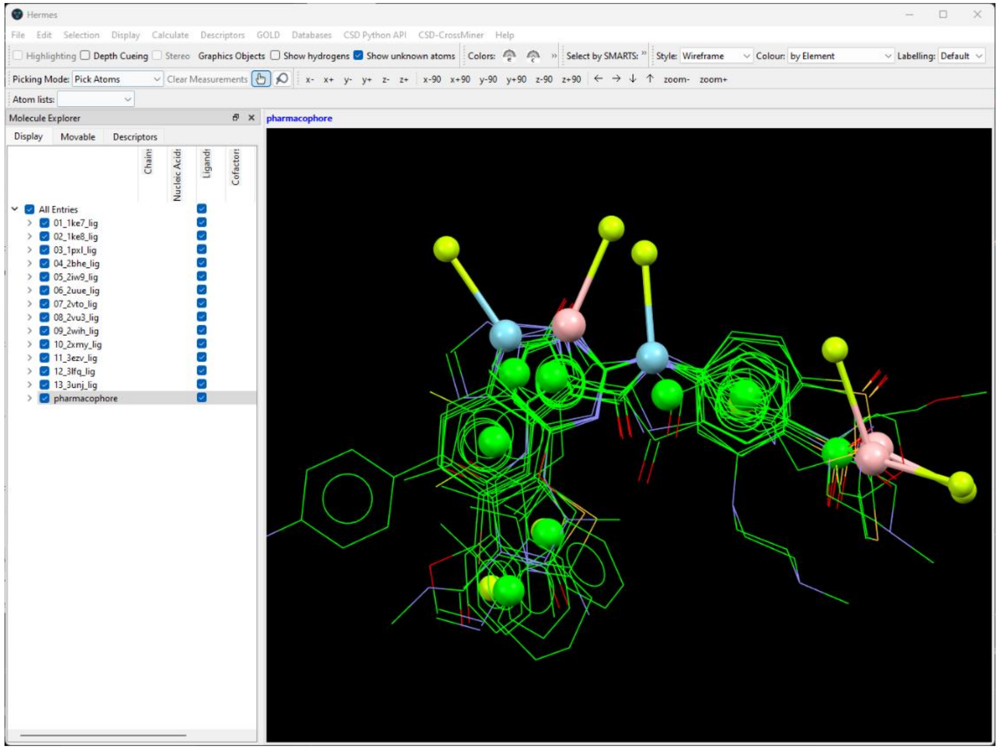


Pharmacophore model

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.

3. 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 light green.


Hermes
File Edit Selection Display Calculate Descriptors GOLD Databases CSD Python API CSD-CrossMiner Help
Graphics Objects
Colors:
Select by SMARTS:
Clear Measurements
x- x+ y- y+ z- z+ x-90 x+90 y-90 y+90 z-90 z+90 ← → ↓ ↑ zoom- zoom+
Molecule Explorer
Display Movable Descriptors
Chains Nucleic Acids Ligands Cofactors
pharmacophore

□ Highlighting □ Depth Cueing □ Stereo □ Show hydrogens ☑ Show unknown atoms

Style: Wireframe Colour: by Element Labelling: Default

Picking Mode: Pick Atoms

Atom lists:

☑ All Entries ☑
☑ 01_1ke7_lig ☑
☑ 02_1ke8_lig ☑
☑ 03_1pxl_lig ☑
☑ 04_2bhe_lig ☑
☑ 05_2iw9_lig ☑
☑ 06_2uue_lig ☑
☑ 07_2vto_lig ☑
☑ 08_2vu3_lig ☑
☑ 09_2wih_lig ☑
☑ 10_2xmy_lig ☑
☑ 11_3ezv_lig ☑
☑ 12_3lfq_lig ☑
☑ 13_3unj_lig ☑
☑ pharmacophore ☑

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/media/ligand-overlay-1.pdf

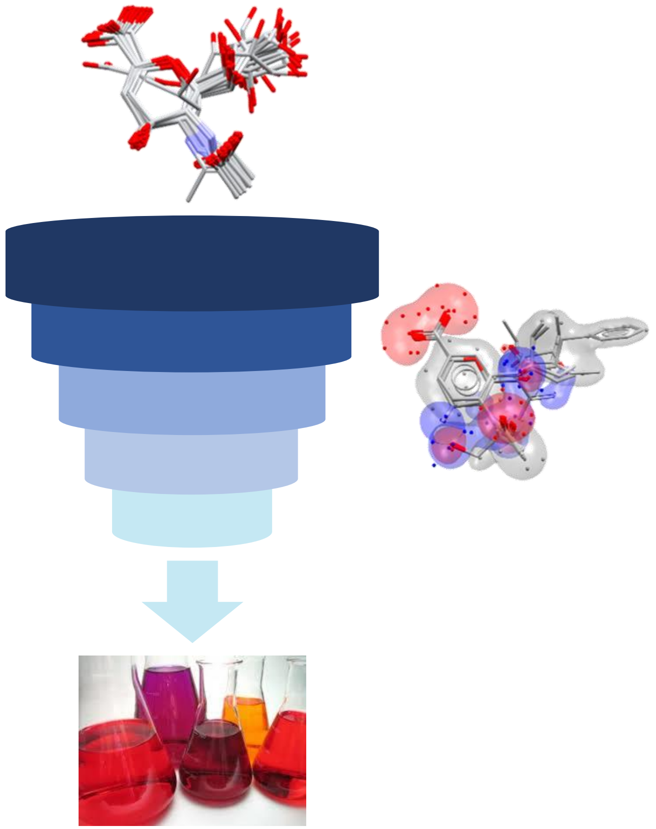
Field-based virtual screening

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 *atomic 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

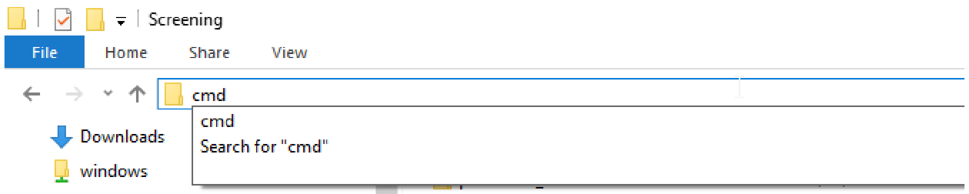
The CSD Python API provides programmatic access to the CSD data and the CSD-portfolio functionality, as well as to features that have never been exposed within an interface. One of these features is the CSD Ligand Screener.
Through Python scripting it is possible to build highly tailored custom applications 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:

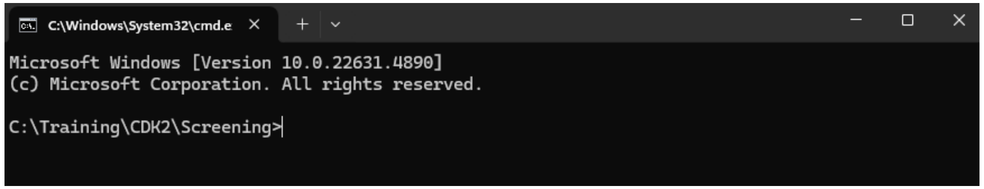
- We will use a solution obtained from the CSD Ligand Overlay to generate our query.
- We will generate knowledge-driven conformers for the library of actives and decoys taken from DUD LIB VS.
- The program will optimise the placement of every conformer onto the fields, but only the top-scoring will 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. `my_path_to\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. `my_path_to/CDK2/Screening`).

1



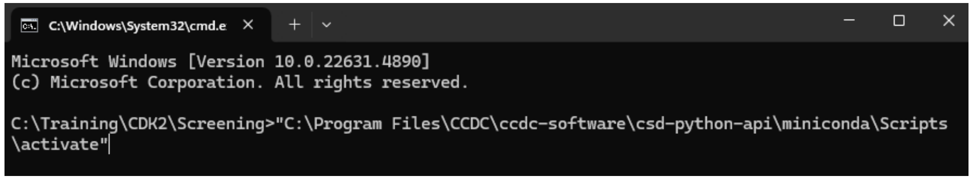


```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Training\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:\path\to\CCDC\ccdc-software\csd-python-api\miniconda\Scripts\activate`"
   If you are a macOS user:
   In the terminal type
   `source Applications/CCDC/ccdc-software/csd-python-api/miniconda/bin/activate`

2



```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Training\CDK2\Screening>"C:\Program Files\CCDC\ccdc-software\csd-python-api\miniconda\Scripts
\activate"
```

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

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
```

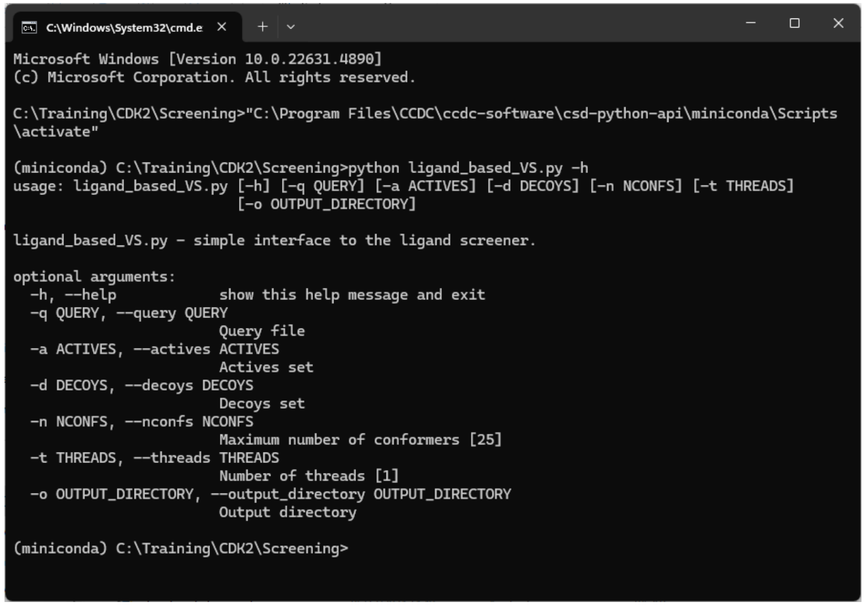
   where:

   - *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.

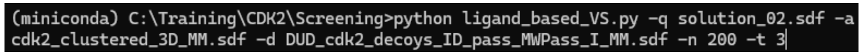
This script shows the basic line for screening the input conformation of the molecules in *my_library.sdf* against an overlay of ligands *solution_02.sdf* as the query and write out the resulting *screening_molecules* and *screening_scores* files for further investigation.

3

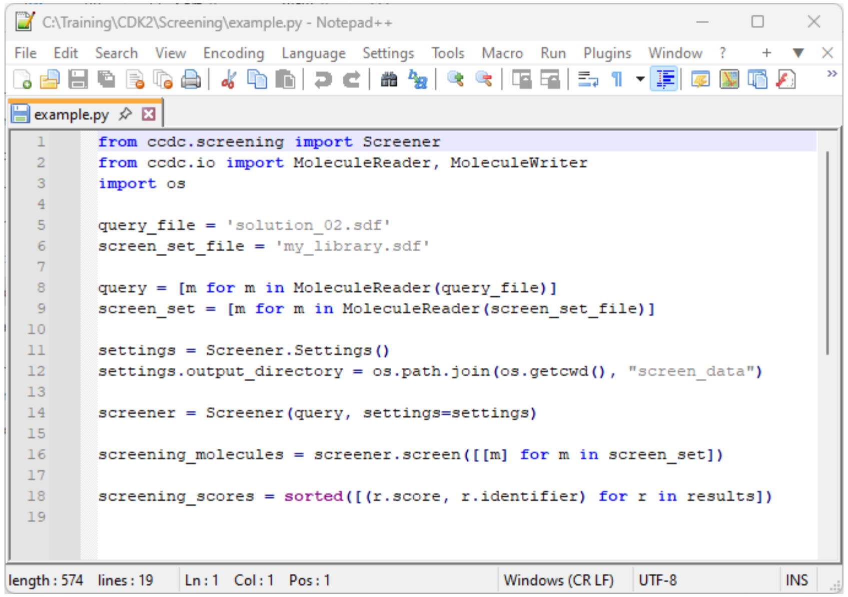


```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Training\CDK2\Screening>"C:\Program Files\CCDC\ccdc-software\csd-python-api\miniconda\Scripts
\activate"

(miniconda) C:\Training\CDK2\Screening>python ligand_based_VS.py -h
usage: ligand_based_VS.py [-h] [-q QUERY] [-a ACTIVES] [-d DECOYS] [-n NCONFS] [-t THREADS]
                          [-o OUTPUT_DIRECTORY]

ligand_based_VS.py - simple interface to the ligand screener.

optional arguments:
  -h, --help            show this help message and exit
  -q QUERY, --query QUERY
                        Query file
  -a ACTIVES, --actives ACTIVES
                        Actives set
  -d DECOYS, --decoys DECOYS
                        Decoys set
  -n NCONFS, --nconfs NCONFS
                        Maximum number of conformers [25]
  -t THREADS, --threads THREADS
                        Number of threads [1]
  -o OUTPUT_DIRECTORY, --output_directory OUTPUT_DIRECTORY
                        Output directory

(miniconda) C:\Training\CDK2\Screening>
```

4



```
(miniconda) C:\Training\CDK2\Screening>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
```



```python
from ccdc.screening import Screener
from ccdc.io import MoleculeReader, MoleculeWriter
import os

query_file = 'solution_02.sdf'
screen_set_file = 'my_library.sdf'

query = [m for m in MoleculeReader(query_file)]
screen_set = [m for m in MoleculeReader(screen_set_file)]

settings = Screener.Settings()
settings.output_directory = os.path.join(os.getcwd(), "screen_data")

screener = Screener(query, settings=settings)

screening_molecules = screener.screen([[m] for m in screen_set])

screening_scores = sorted([(r.score, r.identifier) for r in results])

```

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*).

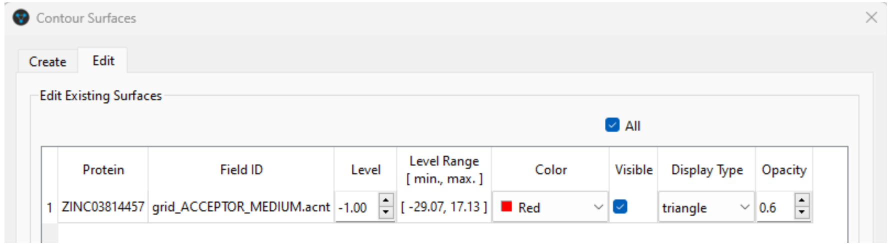
Please note that, after launching the script, the grid maps and fitting points will be created almost immediately. It may take up to **15 minutes** for the *actives_screened.mol2* file (with the top scoring conformer per screened active molecule) to be returned, and up to **2 hours** for the *decoys_screened.mol2* (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.

1. 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.

2. 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**.

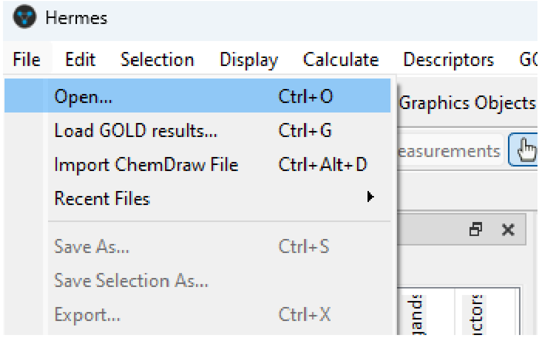
3. In the *Edit* tab change the *Display Type* to triangle and the *Opacity* to 0.6.

3

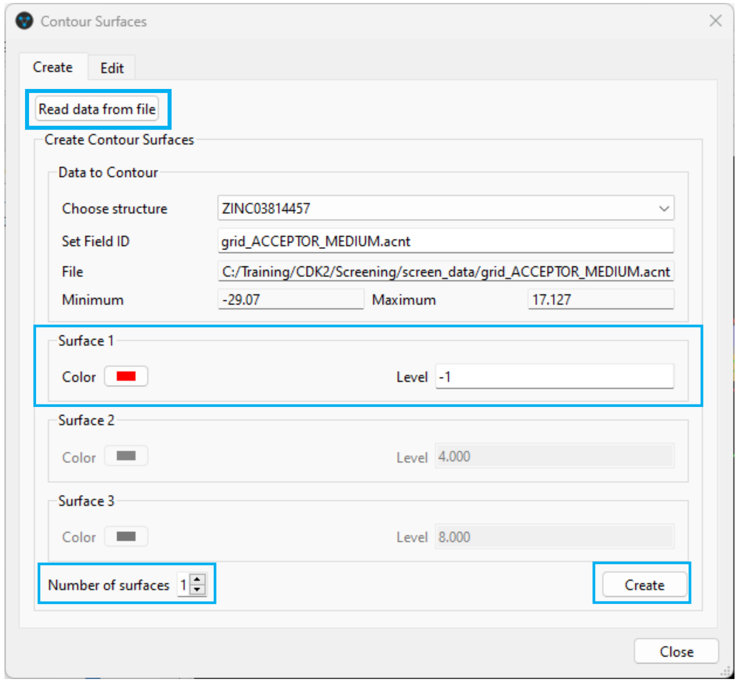


	Protein	Field ID	Level	Level Range [min., max.]	Color	Visible	Display Type	Opacity
1	ZINC03814457	grid_ACCEPTOR_MEDIUM.acnt	-1.00	[-29.07, 17.13]	🟥 Red	☑	triangle	0.6

1



2



Choose structure: ZINC03814457
Set Field ID: grid_ACCEPTOR_MEDIUM.acnt
File: C:/Training/CDK2/Screening/screen_data/grid_ACCEPTOR_MEDIUM.acnt
Minimum: -29.07 Maximum: 17.127

Surface 1: Color 🟥 Level -1
Surface 2: Color Level 4.000
Surface 3: Color Level 8.000

Number of surfaces: 1

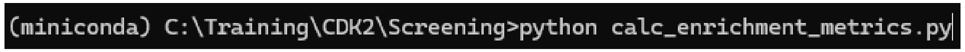
4. Repeat the same steps for *grid_DONOR_MEDIUM.acnt* and *grid_NONPOLAR.acnt*, but set the colour to blue and light grey, respectively.

5. 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.

6. Before running the calc_enrichment_metrics.py script, we need to provide the path to the screening_scores.csv file and a location to write the output. Using a text editor such as Notepad++, open the calc_enrichment_metrics.py script and replace the path to the screening_scores.csv with the location on your machine. Then, specify a location for the output file, this can be any folder for which you have write permissions and save the file. In the command prompt, please type the command to run the *calc_enrichment_metrics.py* script:

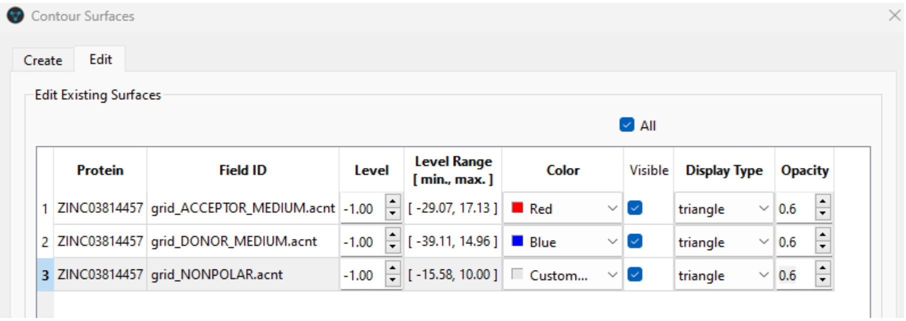
```
python calc_enrichment_metrics.py
```



```
(miniconda) C:\Training\CDK2\Screening>python calc_enrichment_metrics.py
```

6

4



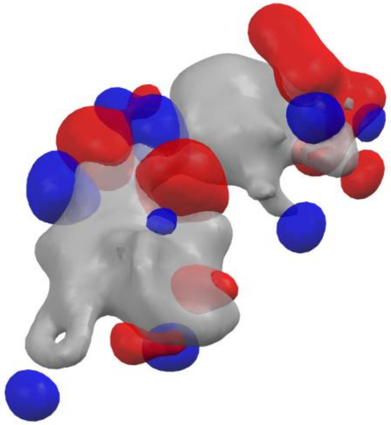
Contour Surfaces

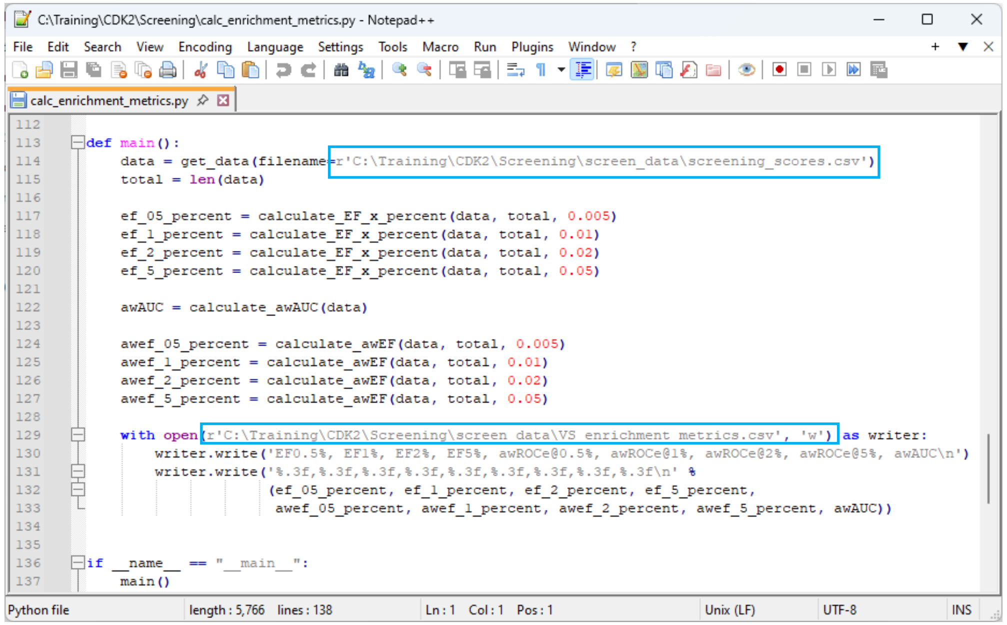
Edit Existing Surfaces

☑ All

	Protein	Field ID	Level	Level Range [min., max.]	Color	Visible	Display Type	Opacity
1	ZINC03814457	grid_ACCEPTOR_MEDIUM.acnt	-1.00	[-29.07, 17.13]	🟥 Red	☑	triangle	0.6
2	ZINC03814457	grid_DONOR_MEDIUM.acnt	-1.00	[-39.11, 14.96]	🟦 Blue	☑	triangle	0.6
3	ZINC03814457	grid_NONPOLAR.acnt	-1.00	[-15.58, 10.00]	□ Custom...	☑	triangle	0.6

5





C:\Training\CDK2\Screening\calc_enrichment_metrics.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
calc_enrichment_metrics.py

```
def main():
    data = get_data(filename=r'C:\Training\CDK2\Screening\screen_data\screening_scores.csv')
    total = len(data)

    ef_05_percent = calculate_EF_x_percent(data, total, 0.005)
    ef_1_percent = calculate_EF_x_percent(data, total, 0.01)
    ef_2_percent = calculate_EF_x_percent(data, total, 0.02)
    ef_5_percent = calculate_EF_x_percent(data, total, 0.05)

    awAUC = calculate_awAUC(data)

    awef_05_percent = calculate_awEF(data, total, 0.005)
    awef_1_percent = calculate_awEF(data, total, 0.01)
    awef_2_percent = calculate_awEF(data, total, 0.02)
    awef_5_percent = calculate_awEF(data, total, 0.05)

    with open(r'C:\Training\CDK2\Screening\screen_data\VS_enrichment_metrics.csv', 'w') as writer:
        writer.write('EF0.5%, EF1%, EF2%, EF5%, awROCe@0.5%, awROCe@1%, awROCe@2%, awROCe@5%, awAUC\n')
        writer.write('%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f,%.3f\n' %
                     (ef_05_percent, ef_1_percent, ef_2_percent, ef_5_percent,
                      awef_05_percent, awef_1_percent, awef_2_percent, awef_5_percent, awAUC))


if __name__ == "__main__":
    main()
```

Python file   length : 5,766   lines : 138   Ln : 1   Col : 1   Pos : 1   Unix (LF)   UTF-8   INS

7. 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
   ```

   where

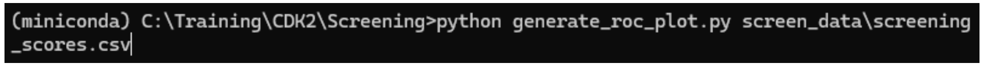
   `screen_data\screening_scores.csv` is the path to the *screening_scores.csv* output file for this virtual screen study.

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

The arithmetic weighted enrichment metrics (which by default were output to `C:\Training\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.

7



```
(miniconda) C:\Training\CDK2\Screening>python generate_roc_plot.py screen_data\screening
_scores.csv
```

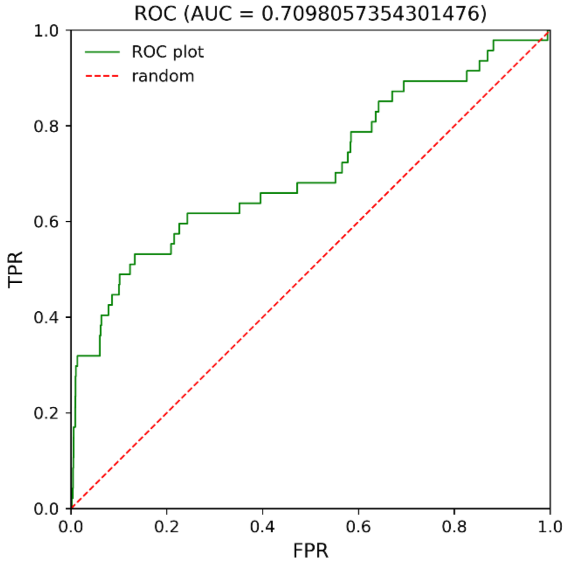


Table 1. Results of the CSD Ligand Screener compared with other methods. Data taken from DOI: 10.1186/1758-2946-1-14. The best value of each metric is highlighted in blue.

VS metrics	CSD Ligand Screener	DOCK	FieldScreen	MACCS	OAK	OAKFLEX	2SHA	OAAP
awAUC	**0.725**	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%	18.328	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%	**27.723**	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

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.

Tips for virtual screening

- 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.

Summary

In this workshop, we have seen a typical workflow for ligand based virtual screening, which involved three stages. You should now be confident in generating molecular conformations using the CSD Conformer Generator, generating pharmacophore hypotheses using the Ligand Overlay tool and screening a library of compounds against a pharmacophore query.

For further details on the programs used in this workshop, please consult the CSD Python API, CSD Conformer Generator and Ligand Overlay user guides.

Next Steps

If you have enjoyed this workshop you might like to try some of the other virtual screening tutorials from the CSD-Discovery self-guided workshop pages. You might also like to explore some of the scripts available from our ccdc-opensource GitHub repository

Feedback

We hope this workshop improved your understanding of *ligand-based virtual screening* and you found it useful for your work. As we aim to continuously improve our training materials, we would love to hear your feedback. Follow the link on the workshop homepage and insert the workshop code, which for this self-guided workshop is *CODE*. It will only take 5 minutes and your feedback is anonymous. Thank you!

Glossary

Actives and decoys (virtual screening)

In virtual screening, actives are molecules which are known to bind to a target experimentally. Decoys are molecules which look similar to actives in some way (such as shape or functional groups) but are known *not* to bind to the target.

Apoenzyme

An apoenzyme is the inactive form of an enzyme that requires a non-protein component (a cofactor) to become active.

Atomic Property Fields

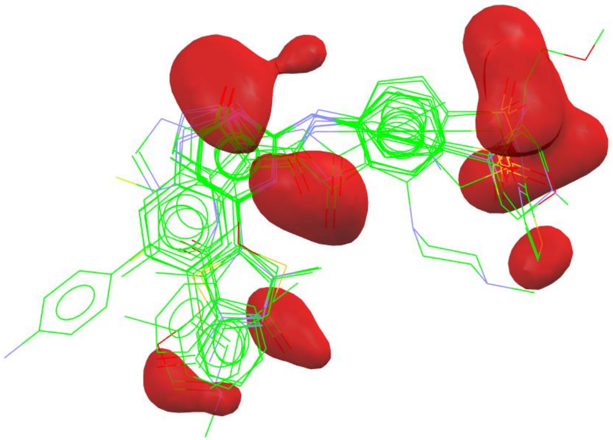
The atomic property fields concept is a continuous, multi-component 3D potential that reflects preferences for various atomic properties at each point in space. Examples of property fields include hydrogen bond donors, hydrogen bond acceptors and non-polar groups.

Conformation

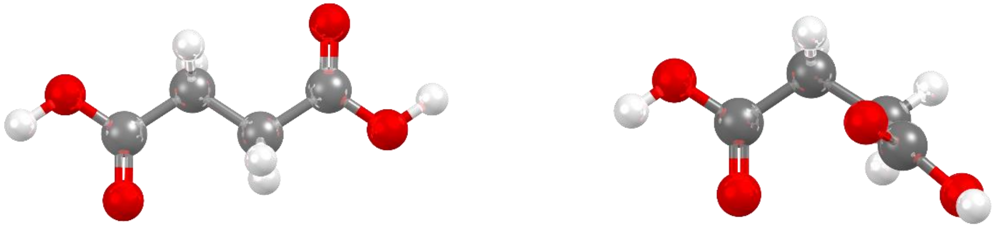
The spatial arrangement of the atoms affording distinction between stereoisomers which can be interconverted by rotations about formally single bonds. Some authorities extend the term to include inversion at trigonal pyramidal centres and other polytopal rearrangements. *Sources*: PAC, 1994, 66, 1077. (Glossary of terms used in physical organic chemistry (IUPAC Recommendations 1994)) on page 1099. If a conformation corresponds to a distict potential energy minimum, it is a *conformer*.

Hydrogen Bonds

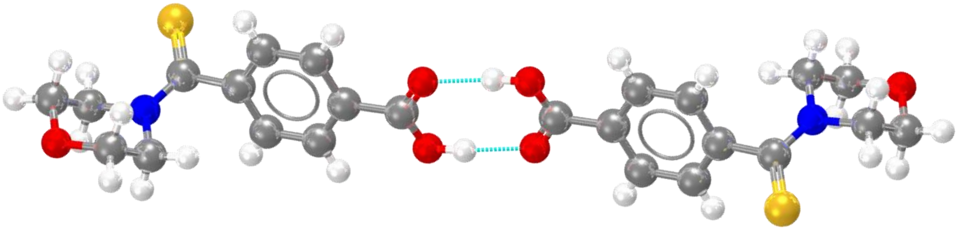
Hydrogen bonding occurs between donor-acceptor interactions precisely involving hydrogen atoms. The H-bonds interactions are classified as: strong (mostly covalent), moderate (mostly electrostatic) and weak (electrostatic). Their strength is observed to be between 12 and 30 kJ/mol.



A hydrogen bond acceptor field generated from a ligand overlay



Two conformations of succinic acid molecules, shown on refcodes SUCACB02 (left) and SUCACB19 (right)



In light blue, example of hydrogen bonds for refcode MULWIC.

Hydrophobic

Hydrophobic molecules effectively “repel” water and thus have a tendency to self-aggregate in aqueous media, excluding water in so doing. On a structural level, these are non-polar groups such as alkyl or aryl moieties. If these functional groups or molecular fragments are also pharmacophore features, then they are called *hydrophobes*.

Ligand (protein)

ligands are small molecules that bind to the protein and can change the protein function.

Ligand-based drug design

A drug discovery method which uses the properties of known active molecules, which are ligands of a target protein, to discover new active molecules without knowledge of the target protein structure.

Pareto rank

Pareto ranking is a method used to rank solutions in multi-objective optimisation based on Pareto dominance. A rank is assigned to a solution based on how many other solutions dominate it. Solution **A** is said to dominate solution **B** if A is as good as **B** in all objectives and **A** is strictly better than **B** in one objective. The set of solution which are not dominated by any others is sometimes called the Pareto front.

Pharmacophore

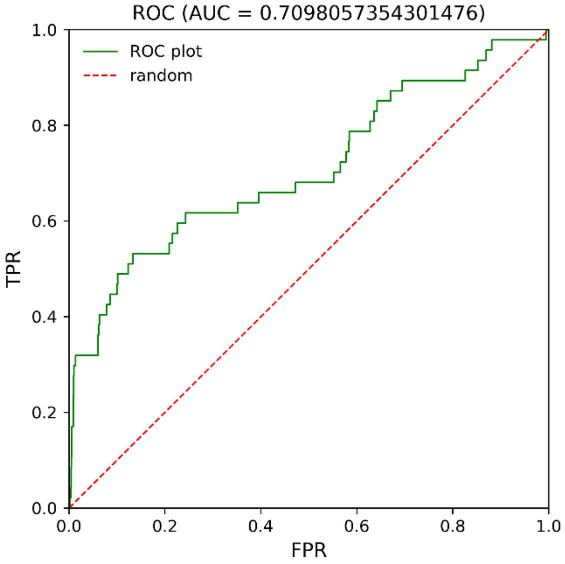
A pharmacophore is the ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or block) its biological response. *Source*: PAC, 1998, 70, 1129. (Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998)) on page 1140.

Protein Data Bank (PDB)

The global archive of 3D structure data for large biomolecules including proteins, DNA and RNA https://www.rcsb.org/.

Receiver Operating Characteristic

The ROC curve (receiver operating characteristic) gives a measure of how well the screening process distinguishes between actives and decoys. The plot shows a trade-off between sensitivity and specificity. The False Positive Rate (FPR), which is the proportion of decoys incorrectly ranked as hits, is plotted on the *x*-axis and the True Positive Rate (TPR), which is the proportion of actives correctly ranked as hits, in plotted on the *y*-axis. The FPR and TPR are calculated at different thresholds in the rank (top 1%, 2% and so on) and plotted cumulatively. The ideal curve shape hugs the top left side of the graph; if the curve rises steeply at first, it shows that the screening is good at ranking actives high up the list (early enrichment). In contrast, a diagonal line indicates no discrimination. The Area Under the ROC Curve (AUC) is used as a convenient metric; an AUC of 1 indicates perfect screening, an AUC of 0.5 corresponds to random behaviour and an AUC of less than 0.5 (rare) means the screening preferentially picks decoys.



Rotamer

Rotamers are conformers that differ only by rotation about one single bond.

Basics of Hermes Visualisation

Hermes is the CCDC's visualization software to study protein-ligands interaction, and it is the interface for protein docking with GOLD.

In the following we will see some of the basics of navigation and visualization in Hermes that you will find helpful to support your analysis.

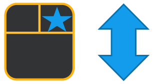
In the **Hermes interface** we find:

- **At the top**: list of menus from which we can access visualization and analysis options, and other CSD-Discovery components.
- **On the left-hand side**: the **Molecule Explorer**, where the proteins, ligands, water molecules, and other molecules present in the structures are listed in the *Display* tab. You can expand each entry to see the components, and tick or untick the corresponding boxes to display / not display them. The solutions from a GOLD run will also be displayed in the Molecule Explorer.
- **On the left-hand side**, below the Molecule Explorer: the **Contact Management** box, which allows to display H-bonds and short contacts for selected molecules by ticking the corresponding box.

Using the **mouse to enhance visualization**:

- Left mouse button and move – rotate molecules

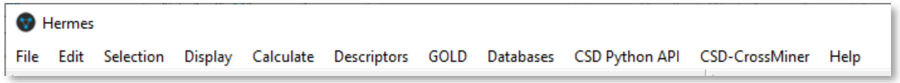
- Middle Mouse wheel – move molecules up and down

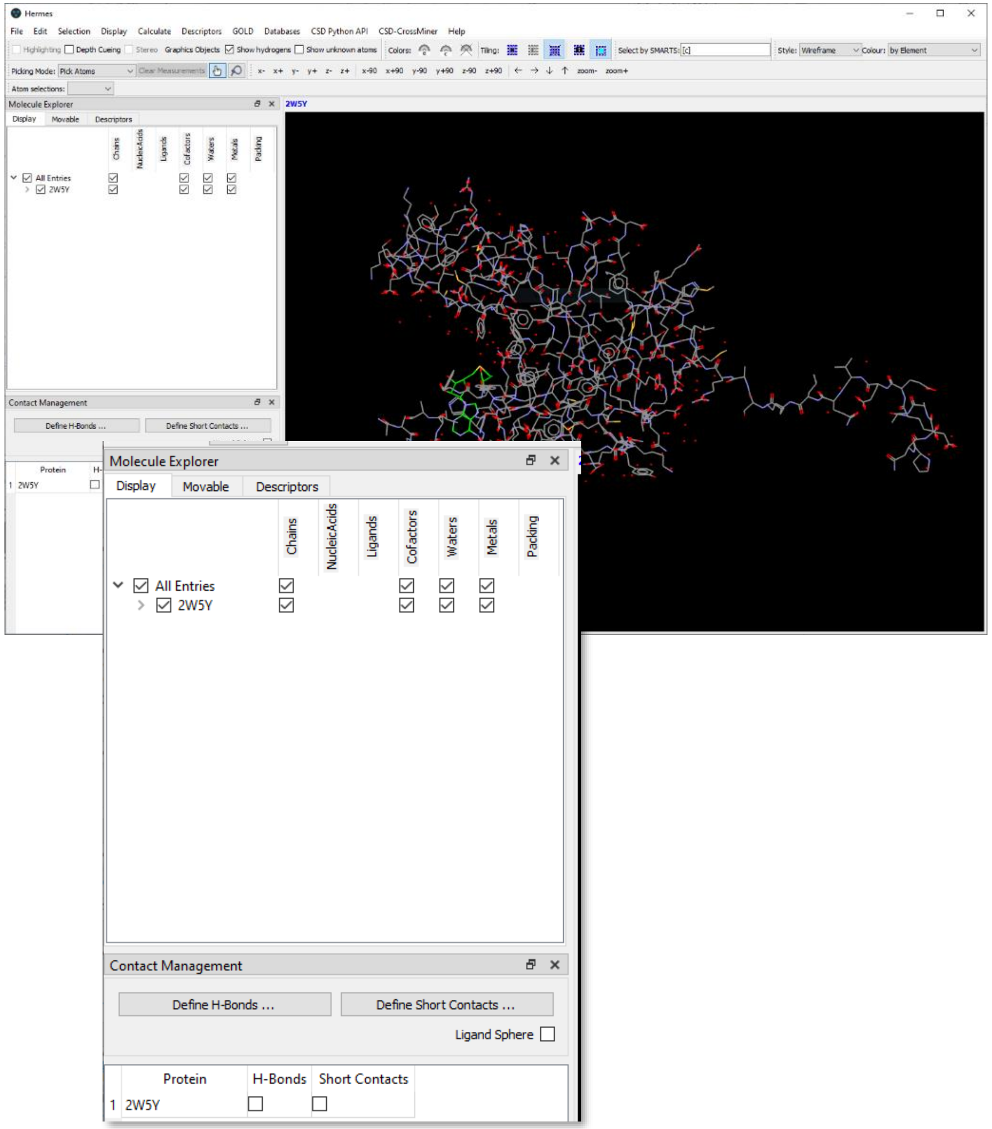


- Right mouse button and move up and down – zoom in and out of molecules

- Shift + Left mouse button and move - rotate in the plane molecules

- Ctrl + Left mouse button and move - translate molecules





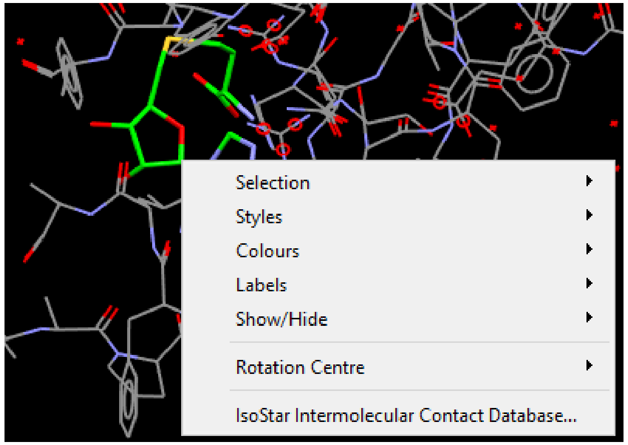
	Protein	H-Bonds	Short Contacts
1	2W5Y	☐	☐

Right click:

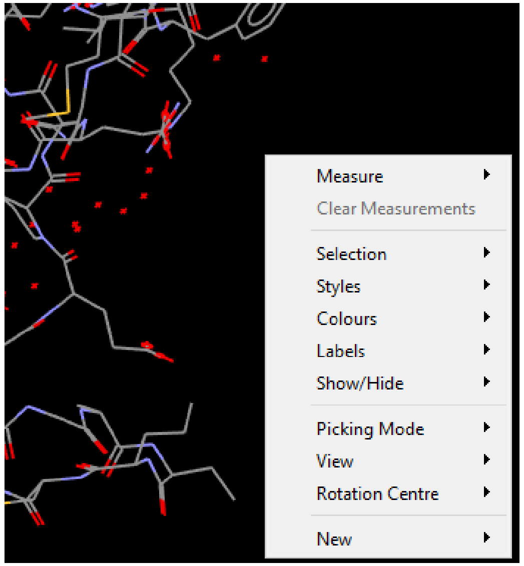
a) On a feature

b) Away from a feature

a



b



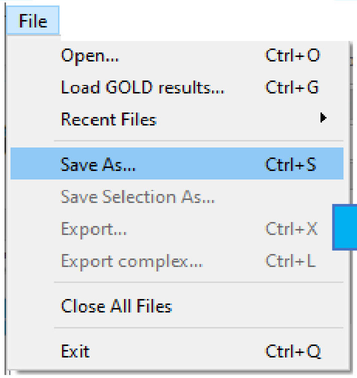
Creating high-resolution images

1. Go to the top menus and click on **File**, then **Save A**s. Alternatively you can use the keyboard and do **Ctrl + S**.

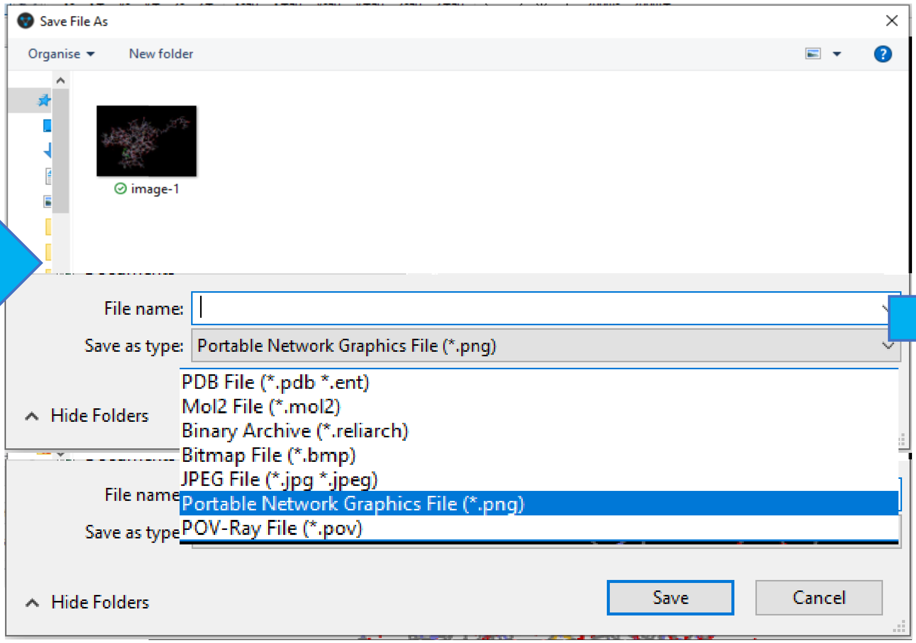
2. In the *Save File As* window, select destination folder, file type and choose a name. Then click **Save**.

3. This will bring up the *Save Image* menu. Here you can select the resolution and tick whether you want a transparent background. Click **Ok**.

1



2



File name:

Save as type: Portable Network Graphics File (*.png)

3

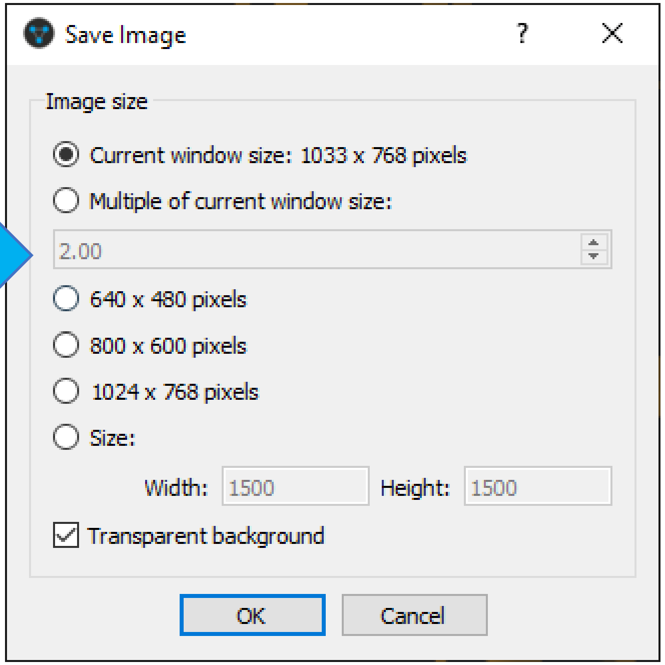


Image size

● Current window size: 1033 x 768 pixels

○ Multiple of current window size:

2.00

○ 640 x 480 pixels

○ 800 x 600 pixels

○ 1024 x 768 pixels

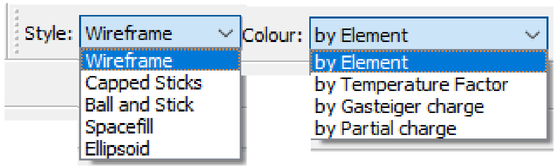
○ Size:

Width: 1500 Height: 1500

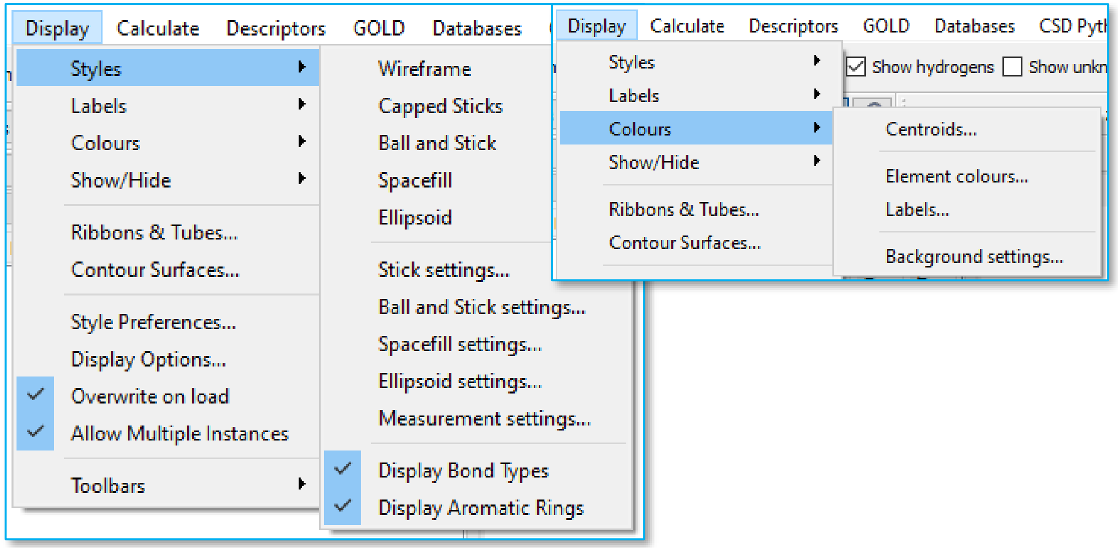
☑ Transparent background

You can access visualization options from the **Display** menu.

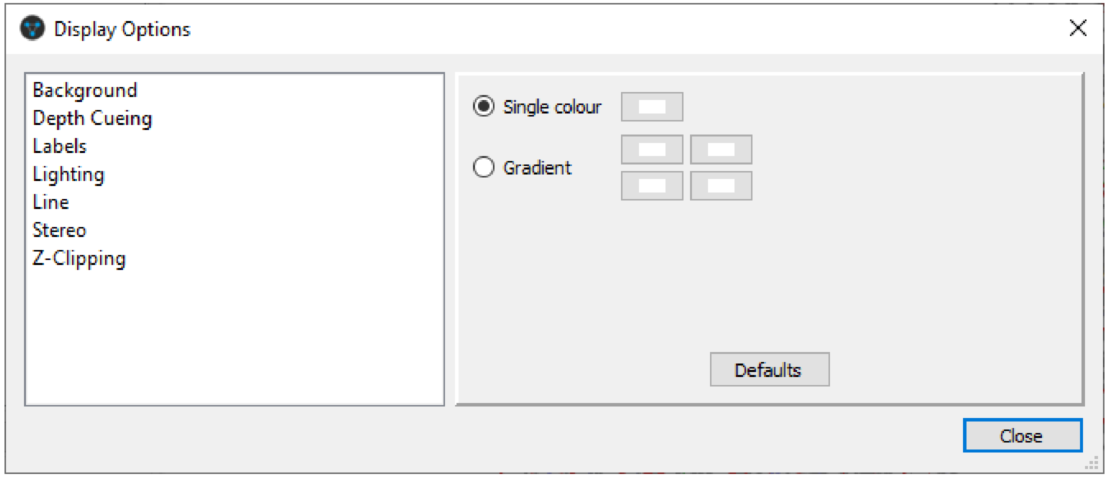
- From **Display > Styles**, you can pick the representation style of atoms and bonds and change settings. **Note**: you can also access this feature from the dropdown menu at the top right of the 3D visualizer window.



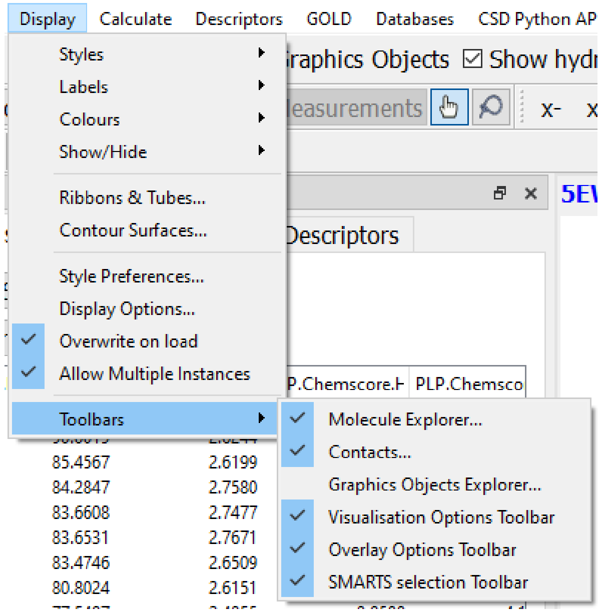
- From **Display > Colours**, you can edit the colour for elements and object. **Note**: you can change the colouring style for the structure from the dropdown menu at the top right of the 3D visualizer window.



- Clicking on Display > Display Options brings up the Display Options menu, where you can edit settings for the 3D visualizer.



- If any of the toolbars at the top of the Hermes interface is not displayed and you wish to have it shown, or vice versa, you can do so from **Display > Toolbars**, where you can ensure to tick the ones you wish to have on.


Display
Calculate
Descriptors
GOLD
Databases
CSD Python AP
Styles
Labels
Colours
Show/Hide
Ribbons & Tubes...
Contour Surfaces...
Style Preferences...
Display Options...
✓ Overwrite on load
✓ Allow Multiple Instances
Toolbars
✓ Molecule Explorer...
✓ Contacts...
Graphics Objects Explorer...
✓ Visualisation Options Toolbar
✓ Overlay Options Toolbar
✓ SMARTS selection Toolbar
raphics Objects
☑ Show hydr
easurements
5EV
Descriptors
P.Chemscore.H
PLP.Chemsco
85.4567
84.2847
83.6608
83.6531
83.4746
80.8024
2.6199
2.7580
2.7477
2.7671
2.6509
2.6151