CSD-CrossMiner User Guide

A Component of the CSD-Discovery Suite

CSD-CrossMiner version 1.5.3

To access our new format tutorials please visit the CSD-CrossMiner web page

Conditions of Use

The CSD-CrossMiner program, the provided databases and associated documentation are copyright works of CCDC Software Limited and its licensors and all rights are protected. Use of the Program is permitted solely in accordance with a valid Software Licence Agreement or a valid Licence and Support Agreement with CCDC Software Limited or a valid Licence of Access to the CSD System with CCDC and the Program is proprietary. All persons accessing the Program should make themselves aware of the conditions contained in the Software Licence Agreement or Licence and Support Agreement or Licence of Access Agreement.

In particular:

- The CSD-Discovery and its CSD-CrossMiner Component are licensed subject to a time limit for use by a specified organisation at a specified location.

- The CSD-Discovery and its CSD-CrossMiner Component are to be treated as confidential and may NOT be disclosed or re-distributed in any form, in whole or in part, to any third party.

- Software or data derived from or developed using the CSD-Discovery and its CSD-CrossMiner Component may not be distributed without prior written approval of the CCDC. Such prior approval is also needed for joint projects between academic and for-profit organisations involving use of the CSD-Discovery.

- The CSD-Discovery and its CSD-CrossMiner Component may be used for scientific research, including the design of novel compounds. Results may be published in the scientific literature, but each such publication must include an appropriate citation as indicated in the Schedule to the Licence of Access Agreement or Products Licence and Support Agreement and on the CCDC website.

- No representations, warranties, or liabilities are expressed or implied in the supply of the CSD-Discovery or its CSD-CrossMiner Component by CCDC, its servants or agents, except where such exclusion or limitation is prohibited, void or unenforceable under governing law.

Licences may be obtained from:

CCDC Software Ltd.
12 Union Road
Cambridge CB2 1EZ
United Kingdom

Web: www.ccdc.cam.ac.uk
Telephone: +44-1223-336408
Email: admin@ccdc.cam.ac.uk

CCDC
advancing structural science

CSD-CrossMiner

Contents

1 Installation Notes

1.1 Minimum system requirements

CSD-CrossMiner can run on **64-bit** Windows, Linux and macOS systems. The minimum recommended RAM is 8GB. The minimum free disk-space for installation is 5GB.

This release is supported on the following platforms and operating systems:

- Windows - Intel compatible, 64-bit versions of Windows:
  - 7, 8 and 10
- Linux - Intel compatible, 64-bit distributions of Linux:
  - RedHat Enterprise 6 and 7
  - CentOS 6 and 7
  - Ubuntu 12, 14 and 16
- macOS - Intel compatible, 64-bit version of MacOS:
  - 10.12, 10.13 and 10.14

If you choose to use a version other than those listed above we cannot guarantee that CSD-CrossMiner will work correctly, although we will attempt to assist you with any problems you may encounter. If you do encounter any difficulties, please contact us at support@ccdc.cam.ac.uk to discuss possible solutions.

The software update mechanism can require more than 8GB to be available during the update activity. We advise that additional swap space or RAM is available at that moment to accommodate a further 4GB.

1.2 Windows 64-bit

Windows users should ensure that they have the 64-bit version of the Microsoft Visual C++ Redistributable Package for Visual Studio 2013 installed. This can be installed via the vcredist_x64.exe obtainable from this link: https://www.microsoft.com/en-GB/download/details.aspx?id=40784

1. Save the crossminer-2019-windows-x64.zip to your Desktop (or any local folder in your computer)
2. Unzip this folder (it is not possible to run the installer from the zip file)
3. Double-click on the setup-windows.exe file and follow the onscreen instructions
4. By default, the installer will install the program at `C:\Program Files\CCDC\CSD_CrossMiner_1.5.3`
5. The CSD-CrossMiner 1.5.3 icon will be created on the Desktop; this points to the executable in the CSD-CrossMiner installation folder.
6. The installer file can be deleted.

1.3 Linux 64-bit

1. Save the crossminer-2019-linux-x64.tar to your Desktop (or any local directory in your computer)
2. Unpack the contents of this file with the command:
   `tar -xvf crossminer-2019-linux-x64.tar`
3. Ensure the setup-linux-x64.run file is executable with the command:
   `chmod a+x setup-linux-x64.run`
4. Execute the run file and follow the onscreen instructions
5. By default, the installer will install the program at
   `/home/username/CCDC/CSD_CrossMiner_1.5.3`
6. The CSD-CrossMiner 1.5.3 icon will be created on the Desktop; this points to the executable in the bin directory of CSD-CrossMiner
   `/home/username/CCDC/CSD_CrossMiner_1.5.3/bin/`
7. The installer file can be deleted.

1.4 macOS

1. Save the crossminer-2019-osx-x64.dmg to your Desktop (or any local folder in your computer)
2. Open the DMG file and double-click on crossminer.osx-installer and follow the installation instructions on the screen.
3. By default, the installer will install the program at
   `/Applications/CCDC/CSD_CrossMiner_1.5.3`
4. The executable to open CSD-CrossMiner can be found in that location
   `/Applications/CCDC/CSD-CrossMiner.app`
5. The installer file can be deleted.

2 Introduction

CSD-CrossMiner is a novel tool that allows crystal structure databases such as the Cambridge Structural Database (CSD) and the Protein Data Bank (PDB) to be searched in terms of pharmacophore queries.

Intuitive pharmacophore queries describing, among others, protein–ligand interaction patterns, ligand scaffolds, or protein environments can be built and modified interactively. Matching crystal structures are overlaid onto the pharmacophore query and visualised as soon as they are available, enabling the user to quickly modify a hypothesis on the fly.

This delivers an overall interactive search experience with application in the areas of interaction searching, scaffold hopping or the identification of novel fragments for specific protein environments. For example use cases, please see:

Korb O, Kuhn B, Hert J, Taylor N, Cole J, Groom C & Stahl M "*Interactive and Versatile Navigation of Structural Databases*" J Med Chem, **2016**, 59(9):4257, **DOI:** 10.1021/acs.jmedchem.5b01756.

3 CSD-CrossMiner Terminology

CSD-CrossMiner uses several specific terms, some common to the field of drug discovery, and some not. For reference, these terms are defined as below:

Features: can be defined as an ensemble of steric and electronic features that characterise a protein and/or a small molecule. In CSD-CrossMiner a feature is defined as point(s), centroid or vector which represent a SMARTS query and, in the case of a vector, this includes geometric rules.

Pharmacophore point: is a feature that has been selected to be part of a pharmacophore because its presence is necessary to ensure the optimal supramolecular interactions with a specific biological target and to trigger or block its biological response.

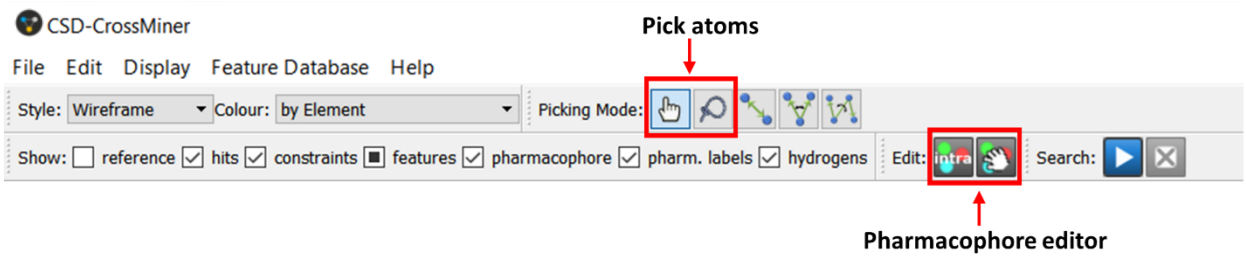
Excluded volume: is a special feature that defines the occupational volume, where no solute molecule in a solution can be present. Excluded volume can be set to be a protein and/or small molecule.

Exit vector: is a two-point feature that represents a single, non-ring bond between two heavy atoms. In CSD-CrossMiner an exit vector is bi-directional, therefore the directionality of the bond is not accounted for.

Structure database: is a database containing the 3D coordinates of small molecule structures and/or protein-ligand binding sites (See Databases in CSD-CrossMiner). This database is used to create a feature database.

Feature database: is a database containing the structures from the structure database, indexed with a set of feature definitions provided by CSD-CrossMiner and any additional features defined by the user. This is the database that CSD-CrossMiner uses to perform the actual 3D search against a pharmacophore query (See Databases in CSD-CrossMiner).

4 Overview of the User Interface

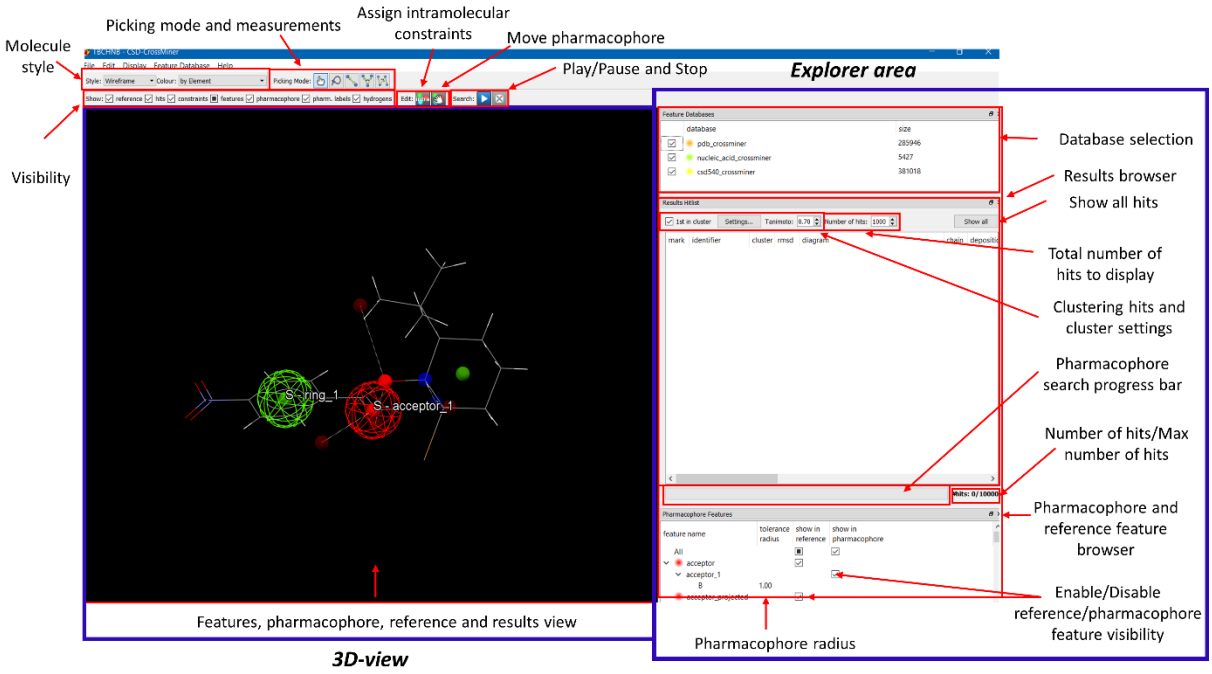


CSD-CrossMiner is a powerful tool with a simple user interface where the mouse function will depend on the selected picking mode: pick atoms and pharmacophore editor.

The CSD-CrossMiner user interface consists of the following:

- *Top-level menu* (See CSD-CrossMiner Top-Level Menu).
- *Style and Picking Mode toolbar* containing common, basic options, *e.g.*, *Style:* for setting global display styles; *Colours:* for setting the colour mode*; Picking mode:* for picking or lassoing atoms and for measuring distances, angles and torsions.
- *Show, Edit and Search toolbar,* containing *Show:* to select what is displayed in the 3D view; *Search:* to *Start/Pause* and *Stop* the pharmacophore search; and *Edit:* to have access to some pharmacophore edit options (See CSD-CrossMiner Toolbars).
- *Display area* (*3D view*) for showing 3D structures, features and pharmacophore points.
- *Explorer area* composed of the following windows:
    - *Feature Databases*, containing the name and the total number of structures contained in the loaded feature database.
    - *Results Hitlist*, containing information about the hits derived from the pharmacophore search (See Results Hitlist and Results Hitlist Browser).
    - *Pharmacophore Features*, containing all feature definitions assigned to the loaded feature database (See APPENDIX B. Feature Definitions in CSD-CrossMiner).

In CSD-CrossMiner, the different toolbars and windows are dockable; therefore, it is possible to hide and move any of these windows that may obscure the user view or they can be kept as entirely separate windows. To do this, move the mouse cursor to the top of one of the window *(e.g.*, *Feature Database*) then drag the window with the mouse, keeping the left-button depressed, and put it where you want by releasing the mouse button.



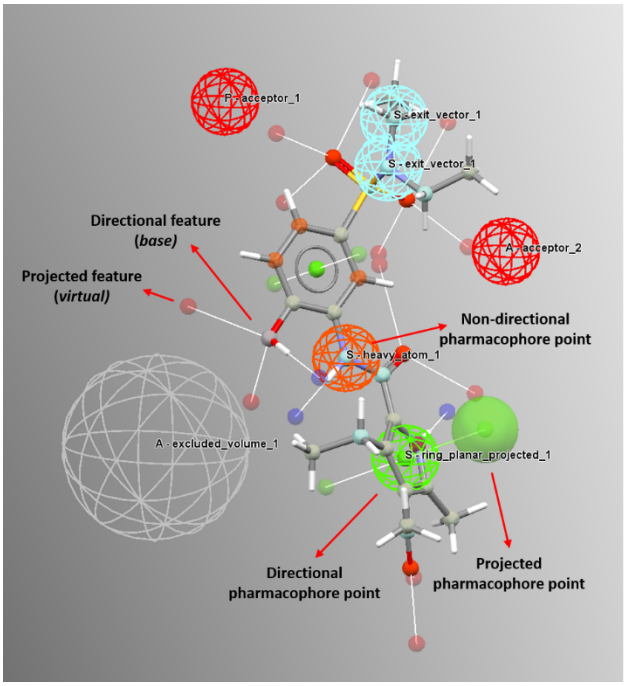
All toolbars and windows can also be switched on or off by right-clicking in the toolbar area and then enabling or disabling the desired toolbar, or by clicking on **Display** in the top-level menu, selecting **Toolbars** and then enabling or disabling the desired toolbar or window.

4.1 Feature and Pharmacophore Representation in CSD-CrossMiner

In the CSD-CrossMiner 3D view, a single-point feature (*e.g.* a heavy atom feature) is represented as a single small translucent sphere coloured as defined in the *Pharmacophore Feature* browser. A directional feature (*e.g.* a projected donor feature) is represented by two types of sphere, *base* and *virtual*, displayed as small translucent spheres. The base feature represents the feature itself while the virtual feature point(s) represent the directionality of the feature. Directional features can have more than one virtual sphere to represent the different directionality.

A pharmacophore point (*e.g.* a heavy atom pharmacophore point) is represented as a mesh sphere. The sphere radius of each pharmacophore point represents the tolerance radius and reflects the uncertainty in the position of the pharmacophore point. The radius of each pharmacophore point can be varied and thus be used to control the specificity of a pharmacophore query. A directional pharmacophore point (*e.g.* a projected donor pharmacophore point) is represented by two types of sphere, *base* and *virtual*, displayed as a mesh sphere and a solid sphere, respectively. A directional pharmacophore point is represented by one base pharmacophore point and virtual pharmacophore point(s), where directionality is defined by the virtual pharmacophore point(s). A pharmacophore point can be set to belong to a protein (P), small molecule (S) or to any molecule (A).

An excluded volume pharmacophore point is displayed in the CSD-CrossMiner 3D view as a single mesh sphere.



P: Protein pharmacophore point
S: Small molecule pharmacophore point
A: Any molecule pharmacophore point

Translucent sphere: Feature point
Mesh sphere: Pharmacophore point itself and pharmacophore *base* in directional feature
Solid sphere: *Virtual* pharmacophore point

An exit vector pharmacophore point is displayed as two mesh spheres.

5 Databases in CSD-CrossMiner

In CSD-CrossMiner there are two distinct types of database: structure database and feature database, where the structure database contains the 3D coordinates of the molecules and the feature database contains the structure database indexed with the feature definitions (See CSD-CrossMiner Terminology).

5.1 Structure and Feature Databases Supplied with CSD-CrossMiner

CSD-CrossMiner is supplied with the structure and feature databases containing the molecular structures of small molecules stored in the CSD and protein-ligand binding sites extracted from the PDB.

For CSD structures (version 5.40) a subset has been created containing structures that (a) are organic plus a small list of transition metals, i.e. Mn, Fe, Co, Ni, Cu, Zn (b) have an *R*-factor of at maximum 10%, (c) for which 3D coordinates have been determined, (d) have no disorder, and (e) are not polymeric. This resulted in a structure database containing over 380 000 structures.

For PDB structures, database entries were generated for each HET group (non-standard protein residues) in protein-ligand complexes, except metals, water molecules and commonly found small ions. Two subsets of the PDB structures have been created, the *pdb_crossminer* subset consists of PDB structures that a) do not contain nucleic acids b) have a resolution < 3Å, c) have ligands with more than 5 or less than 100 atoms, d) contain only the first model in NMR structures, e) does not contain unknown ligands. A second *nucleic_acid _crossminer* subset consists of PDB structures that a) contain

nucleic acids in addition to protein-ligand complexes b) have a resolution < 3.5Å, c) have ligands with more than 5 or less than 100 atoms, d) contain only the first model in NMR structures, e) does not contain unknown ligands. The Hydrogens were added using the Protein `add_hydrogens` function in the CSD Python API (See CSD Python API Documentation).

All molecules and atoms in the protein-ligand binding site were included in the respective database entry in *mol2* format, where the binding site is defined within a cut-off radius of 6 Å around the selected HET group. Note that for *nucleic_acids_crossminer* subset, the protein-ligand binding site may not contain DNA/RNA. The resulting SQLite structure databases (*pdb_crossminer* and *nucleic_acid_crossminer*) consist of more than 285 000 binding sites derived from more than 60 000 PDB entries.

The supplied feature database holds the structures from the structure database described above, indexed with a set of feature definitions, to define the ensemble of steric and electronic features that characterise a protein, nucleic acids and/or a small molecule. This feature database is used to perform the actual 3D search. The feature definitions used to create the supplied feature database are stored in the `feature_definitions` folder in the CSD-CrossMiner installation directory (APPENDIX B. Feature Definitions in CSD-CrossMiner).

In addition to the provided structure and feature databases, CSD-CrossMiner allows the user to create both structure and feature databases starting with their own structures (See Creating Databases).

5.1.1 Entry Identifiers

All entries in the structure and feature databases need to have an identifier that is unique in that database, that is generally defined during the creation of the structure database. In the provided feature database, the indentifier used for a small molecule structure is the Refcode used in the CSD. For the PDB structures in the supplied feature database, the unique identifier contains detailed information about the entry such as: PDB structure ID, model number, the protein chain(s) involved in the protein-ligand interactions and the ligand ID.

For example:

- `1A9U_m1_A_bs_SB2_A_800` corresponds to the binding site from 1A9U, single model structure with only one ligand: SB2 in chain A with residue number 800.
- `1A29_m1_A_bs_TFP_A_153` and `1A29_m1_A_bs_TFP_A_154` correspond to the binding sites for structure 1A29, single model structure with two TFP ligands.
- `3D5Q_m1_A-B_bs_T30_A_293`, `3D5Q_m1_A-B_bs_T30_B_1`, `3D5Q_m1_A_bs_NAP_A_1`, `3D5Q_m1_B_bs_NAP_B_2`, `3D5Q_m1_C-D_bs_T30_C_1_2`, `3D5Q_m1_C-D_bs_T30_D_1`, `3D5Q_m1_C_bs_NAP_C_3`, `3D5Q_m1_C_bs_NAP_D_4`, and `3D5Q_m2_C-D_bs_T30_C_1_2`, correspond to the binding sites of different ligands and binding sites with more than one protein chain forming the binding site.

The identifiers used in the feature database can be saved by clicking on **File** from the top-level menu in CSD-CrossMiner and then **Export Identifiers**. The saved identifiers can be used to create a *csv* file containing annotations for the database entries (See Annotating a Feature Database).

5.1.2 Annotations

The annotations in a feature database contain information about each database entry such as: the deposition date, the resolution, the PDB code, the CSD refcode. The data takes the form of *key-value* pairs of text and is displayed in the *Results Hitlist* browser for all hits matching the pharmacophore query.

The list of all annotations used in the supplied feature database are listed below:

`chain, deposition_date, ec_number, molecule, is_covalent, molecule_fragment, molecule_synonym, organism, organism_taxid, pdb, pdb_class, pdb_title, resolution, structure_method, CSD Refcode, formula` and `r factor`.

These annotations can be used to filter the database from which hits can be found (See Using Annotations as Filter).

Feature database can be annotated during a CSD-CrossMiner session with additional user-defined data (See Annotating a Feature Database). In addition, entries can be annotated during the process of creating the structure database (See APPENDIX D. Create a Feature Database with In-House Data).

6 Loading a Feature Database

The first step required in CSD-CrossMiner is to load the feature database and to initialise the associated structure database(s).

The first time CSD-CrossMiner is launched, a *Load Feature Database* pop-up window allows the user to select the desired feature database to load. Note that the location of the selected feature database will be automatically remembered between separate CSD-CrossMiner sessions, where a pop-up window will provide information about the name and path to the database being loaded. The user can interrupt the loading process by clicking on the **Cancel** button in the pop-up window and then load a new feature database by clicking on **File** from the top-level menu in CSD-CrossMiner, and **Load Feature Database…**. Note that an alternative feature database can be loaded at any point during a CSD-CrossMiner session, but that doing so will clear the current session of any existing pharmacophores or results.

The feature database supplied with the CSD-CrossMiner installation is called "*csd_pdb_crossminer.feat*" and is located in `crossminer_data` folder in the `CSD_2019` directory. After the loading is complete, the name of the structure database(s) and the number of included entries will be displayed in the *Feature Databases* window. The *Pharmacophore Features* window in the bottom right corner displays the feature definitions that were used to create the feature database.

7 Creating, Modifying and Saving Pharmacophore Queries

To start a pharmacophore search, a pharmacophore query has to be created or loaded. This activates the pharmacophore search buttons in CSD-CrossMiner. Instructions on how to load, create and modify a pharmacophore query are discussed below. Instructions on how to start, pause and stop a pharmacophore search are discussed later (See Pharmacophore Search).

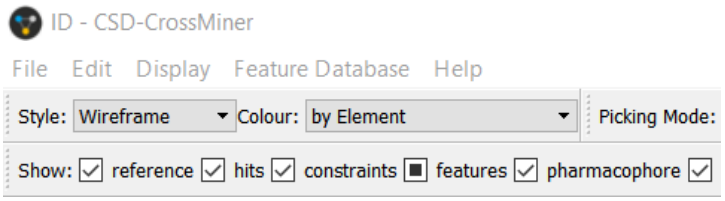
7.1 Loading an Existing Pharmacophore Query

An existing pharmacophore query can be loaded by selecting **File** from the top-level menu in CSD-CrossMiner, then **Load Pharmacophore…**. Several pharmacophore query examples can be found in the `example_pharmacophores` folder of the CSD-CrossMiner installation. Once a pharmacophore is loaded, it is displayed in the 3D view and the corresponding feature definitions are shown in the *Pharmacophore Features* window.

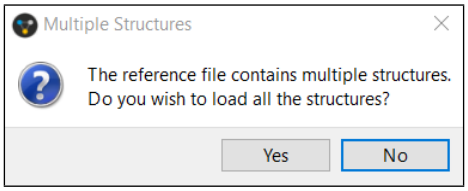
A pharmacophore search can then be started for this pharmacophore query by clicking on  (See Pharmacophore Search). The pharmacophore query can be cleared via **File** > **Close Pharmacophore**.

7.2 Creating a Pharmacophore Query from a Reference Structure

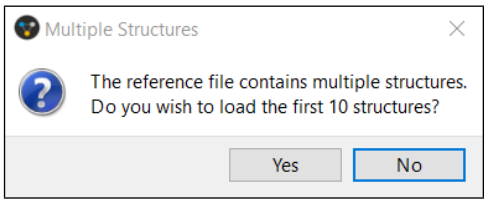
A pharmacophore query can be created from a reference structure that can be loaded via **File** > **Load Reference** (common molecule file formats are supported). The name of the loaded reference structure is then included in the title bar of the CSD-CrossMiner window (ID in the screenshot below).



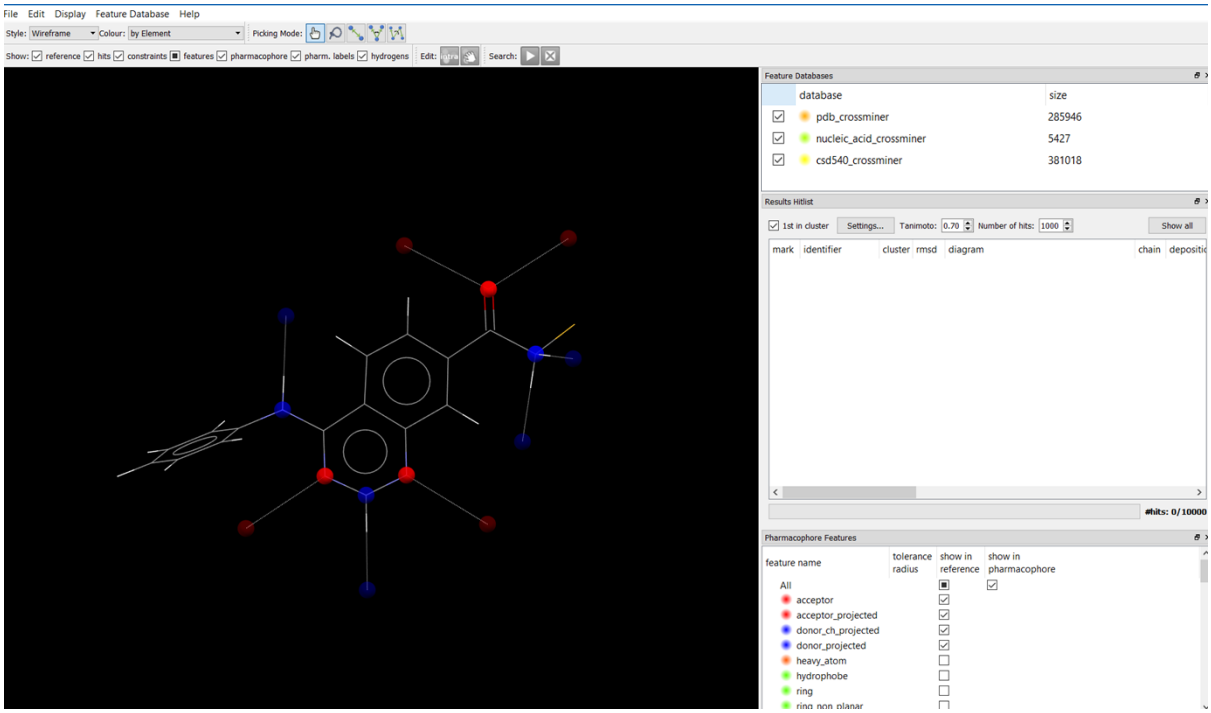
In CSD-CrossMiner it is possible to use a multi structures file as reference (e.g. overlaid ligands). Note that, the maximum number of reference structures that can be loaded and displayed in CSD-CrossMiner is ten. If the loaded reference file contains up to ten structures, by clicking on **Yes** in the *Multiple Structures* pop-up window, all the structures included in the reference file will be loaded and displayed in the 3D view; by clicking on **No** only the first structure in the reference file will be loaded.



If a reference file contains more than ten structures, clicking on **Yes** in the *Multiple Structures* pop-up window will load and display only the first ten structures of the reference file in the 3D view. Otherwise, by clicking on **No**, only the first structure of the reference file will be loaded.



When a new molecule from a reference structure is loaded in CSD-CrossMiner, by default only donor and acceptor features associated with the reference structure are displayed in the 3D view. The features are represented in the 3D view as small translucent spheres, whose identity and associated colour is shown in the *Pharmacophore Features* window.

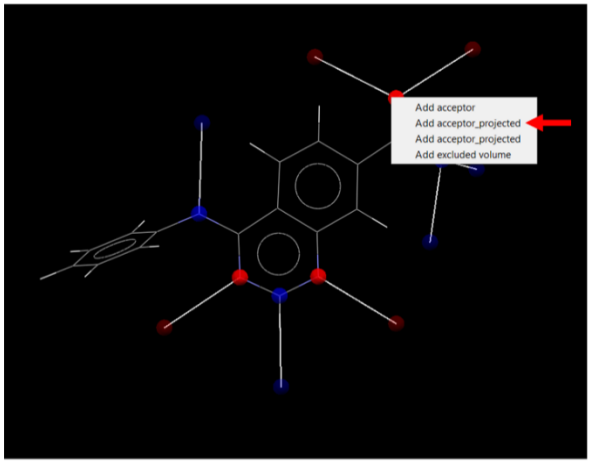


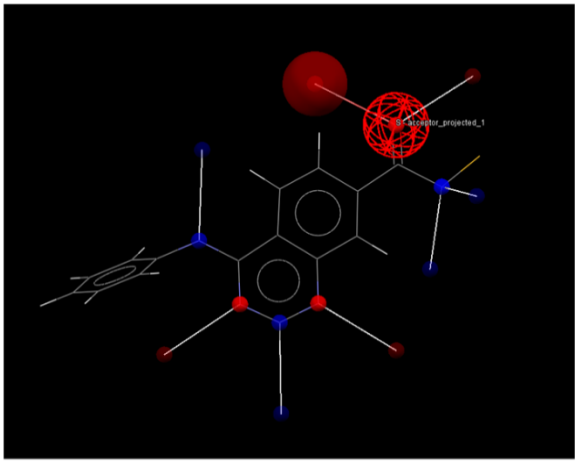
The displayed features of the reference structure are ticked in the *show in reference* column in the *Pharmacophore Features* window. Features can be displayed/hidden in the 3D view by ticking/unticking the relative tick-box in the *Pharmacophore Features* window. When only some of the features of the reference structure are displayed, the *features* tick-box in the CSD-CrossMiner *Show:* toolbar and the *All* features tick-box in the *Pharmacophore Features* window are shown as ▣.

All features associated with the reference structure can be displayed by clicking on the *feature* ▣ tick-box in the CSD-CrossMiner *Show:* toolbar (which will turn to ☑) or alternatively by ticking the tick-box for *All* features in the *show in reference* column in the *Pharmacophore Features* window. All features can be hidden by clicking on *features* ☑ (which will turn to ☐) or alternatively by unticking the *All* tick-box.

Note that if a different choice of displayed features is made (*e.g.* all features displayed), the new settings will be remembered if a new reference molecule is loaded (*e.g.* in this case all features of the new reference molecule would be displayed).

Right-clicking on a feature of the reference structure allows a pharmacophore point of the type(s) available for this feature to be created.





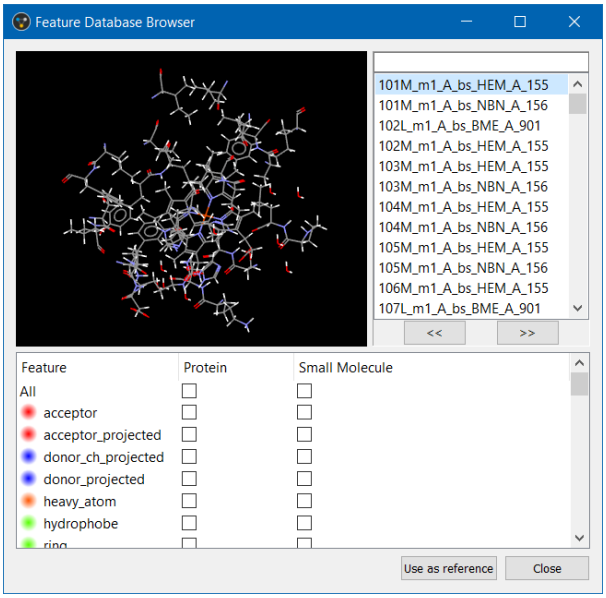
Note that for some features (*e.g.* acceptor) it is possible to define the directionality of the feature by choosing to create a projected pharmacophore point (*e.g.* acceptor_projected). In doing so, the pharmacophore search will lead to hits where both the base and the virtual feature(s) of the pharmacophore point have to be satisfied (See Editing and Creating Feature Definitions).

A pharmacophore search can then be started for this pharmacophore query by clicking on  (See Pharmacophore Search). The entire pharmacophore query and the reference molecule can be cleared via **File** > **Close Pharmacophore** and the **File** > **Close Reference** respectively.

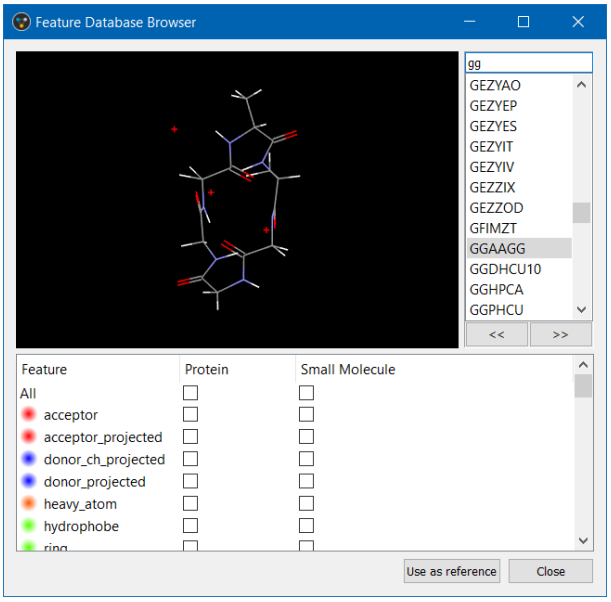
7.3 Creating a Pharmacophore Query from a Feature Database Entry

The feature database browser can be accessed by clicking on **Feature Database** in the CSD-CrossMiner top-level menu and then **Browse**. The *Feature Database Browser* pop-up window displays the list of all entries stored in the feature database and the feature definitions divided by *Protein* and *Small Molecule*. By default, the first molecule in the list is shown in the 3D Display of the *Feature Database Browser* window**.**

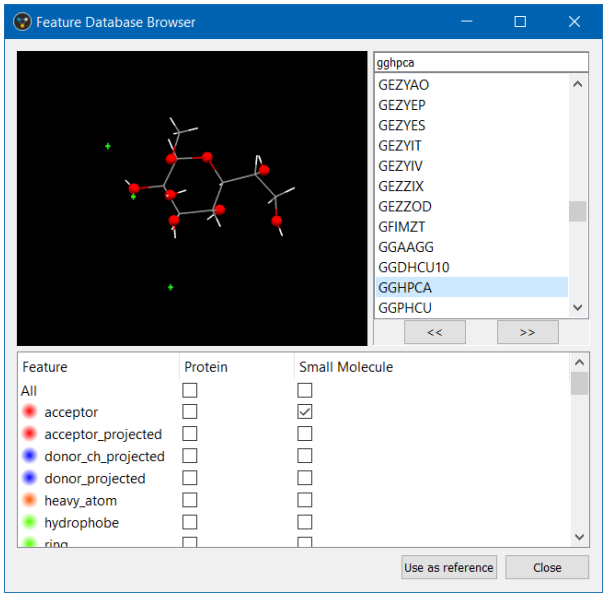
The user can navigate through the different database entries by clicking on a structure name in the right-hand panel, or by clicking on the **<<** and **>>** buttons at the bottom of it, or by using the up and down keyboard arrows to scroll through the list.



It is also possible to search for a specific structure in the database by typing the entry in the top-right text box of the *Feature Database Browser* window. Note that the search is not case sensitive and the list will be sorted by relevance while typing, with the most relevant result appearing at the top of the list.



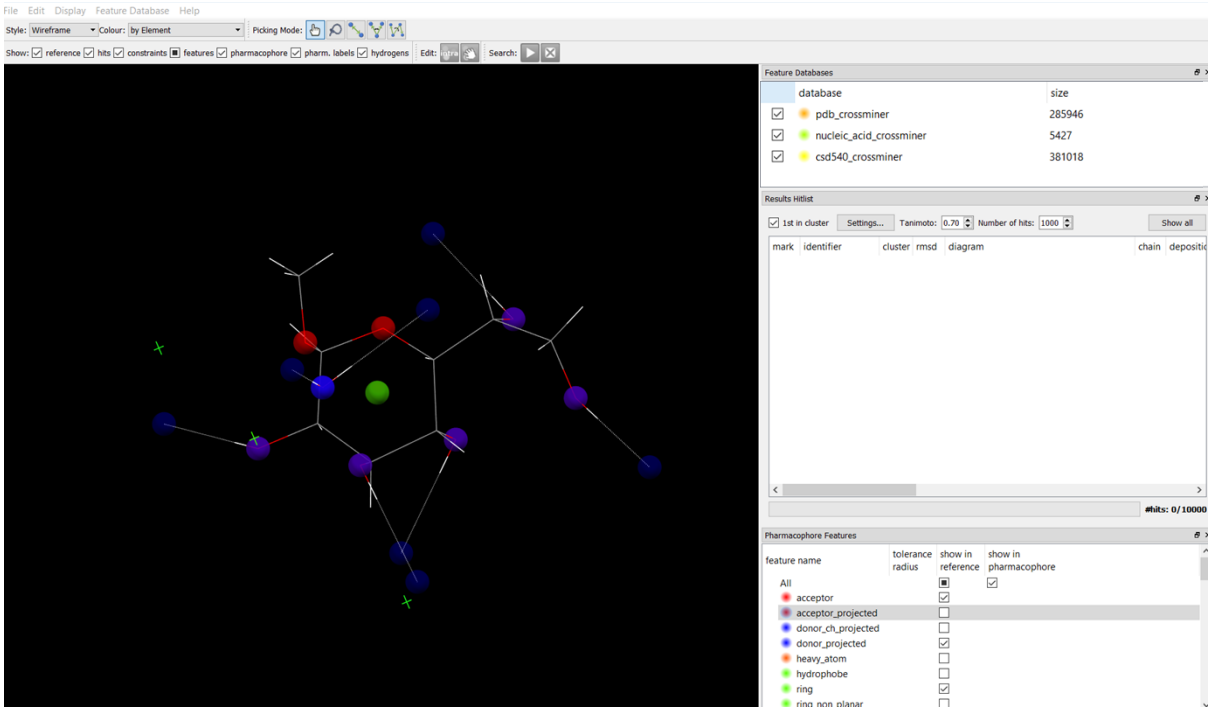
Selecting a feature database entry in the feature database list in the right-hand panel also displays it in the 3D display of the *Feature Database Browser* window (note that multi-selection is not allowed). Additionally, features associated with the selected structure can be displayed by ticking the *Protein* and/or *Small Molecule* tick-box corresponding to the desired feature types. Note that nucleic acids features are associate with the *Small Molecule* component.



Once a feature database entry is selected, it can then be loaded in the main 3D view of CSD-CrossMiner by clicking on the **Use as reference** button in the *Feature Database Browser* window.

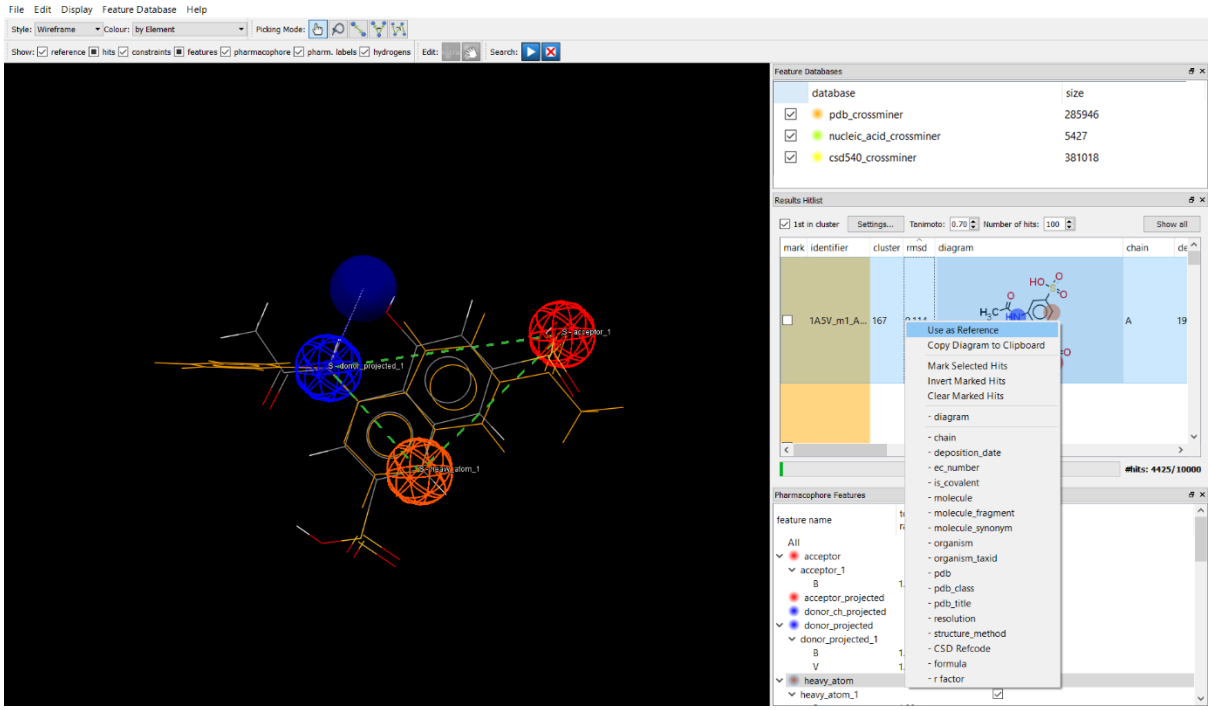
Note that the displayed features of the new reference structure will correspond to the choice of displayed features made during the CSD-CrossMiner session (features with ticked tick-box in *show in reference* column in the *Pharmacophore Features* window).

Features can be displayed, and pharmacophore points can be created from the features of the selected database entry that is now used as a reference structure (See Creating a Pharmacophore Query from a Reference Structure).



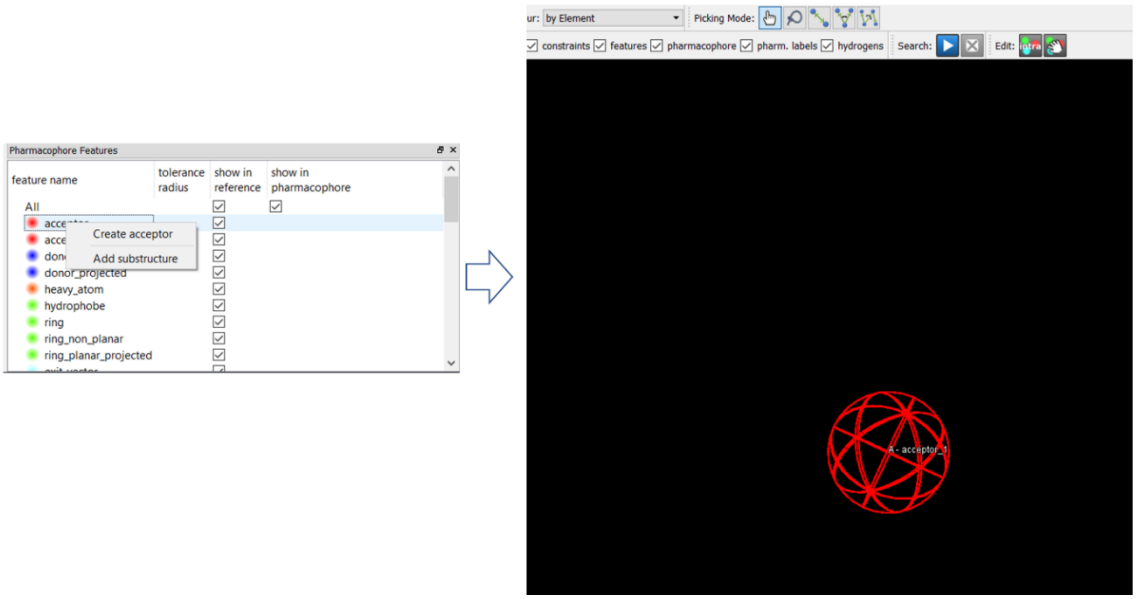
8 Creating a Pharmacophore Query from a Hit

A pharmacophore query can also be created based on any hit identified during a pharmacophore search, by right-clicking on the hit of interest in the *Results Hitlist* browser and selecting **Use as reference** from the resulting context menu. The pharmacophore query can then be defined from the features of this reference hit structure (See Creating a Pharmacophore Query from a Reference Structure).



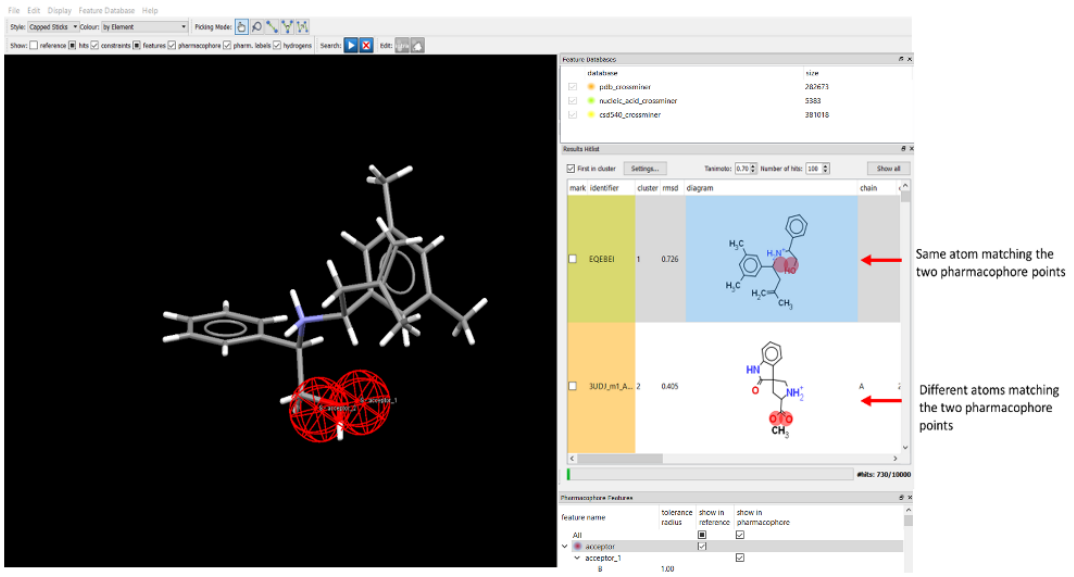
8.1 Creating a New Pharmacophore Query

Starting from an empty 3D view (if necessary, delete an existing pharmacophore query using **File** > **Close Pharmacophore** and/or delete any existing reference structure using **File** > **Close Reference**), pharmacophore points can be added from scratch by right-clicking on a desired feature type in the *Pharmacophore Features* window and then choosing **Create** from the resulting context menu. The corresponding pharmacophore point will appear in the 3D view:



By default, this will create an any molecule (A), pharmacophore point. The molecule type of the pharmacophore point can be modified to be protein (P) or small molecule (S) (See Modifying a Pharmacophore Query). If multiple pharmacophore points are created in this manner, they may be overlaid on top of each other in the 3D view. It is possible to translate each pharmacophore point by clicking on the interactive pharmacophore editing mode  in the *Edit:* toolbar. This will turn the mouse cursor to a small hand, allowing the pharmacophore point to be translated, even during a pharmacophore search. Pressing the left mouse button (LMB) and dragging a pharmacophore sphere to a given position will move it to this position.

Note that if pharmacophore points overlap, it is possible to obtain hits where the same atom in a structure can match multiple pharmacophore points.



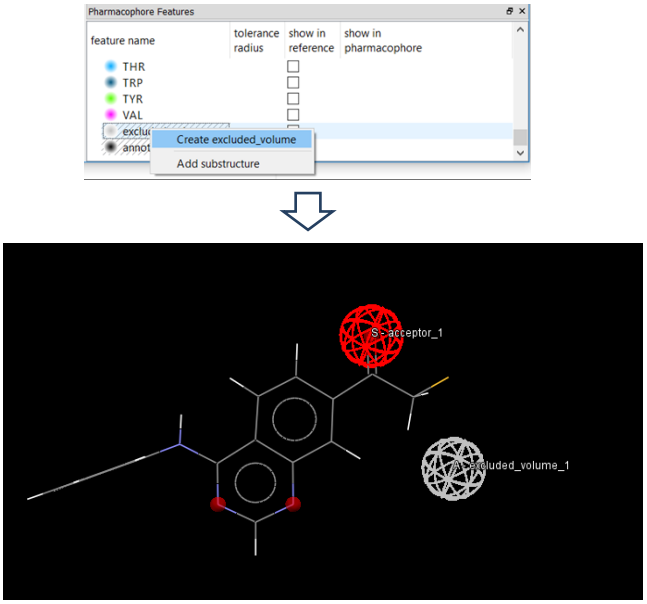
8.2 Adding an Excluded Volume to a Pharmacophore Query

An excluded volume is a special type of pharmacophore point that can be added to a pharmacophore query. This feature can be defined as a volume of occupation that can be set to be any of the three molecule types, *i.e.* Protein, Small Molecule or Any Molecule. There is no limitation on the number of excluded volume features that can be used in a pharmacophore query.

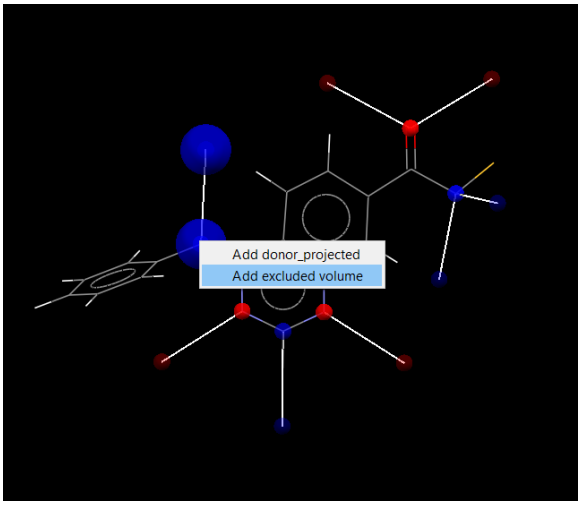
Because an excluded volume feature is never indexed in the feature database, it is represented with diagonal hatching in the *Pharmacophore Features* window.

An excluded volume feature can be added:

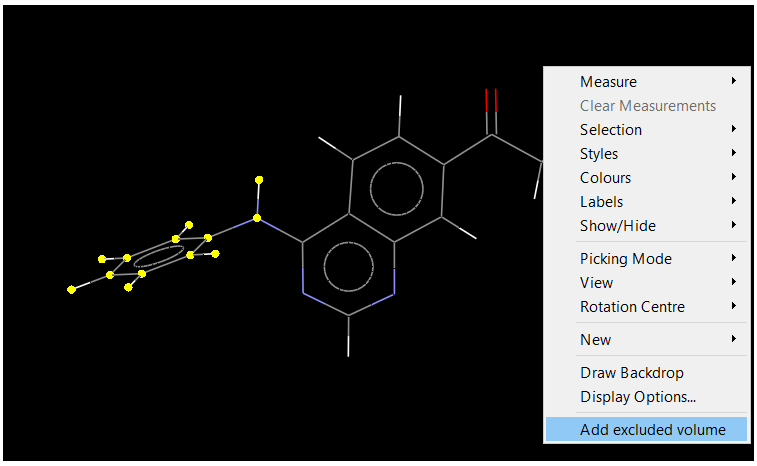
- From scratch: through the *Pharmacophore Features* window, by right-clicking on *excluded_volume* feature and then selecting **Create excluded_volume.** Note that it is not possible to perform a pharmacophore search with only an excluded volume pharmacophore point defined.



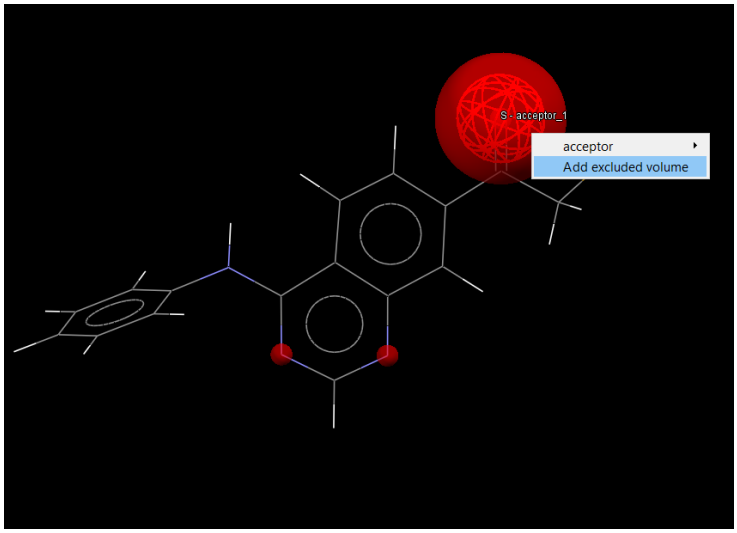
- From the features of a reference structure: by right-clicking on a feature and then selecting **Add excluded volume**.



- From the atom(s) of a reference structure: by selecting the atom(s) of a structure loaded in the 3D view (click on each atom or select multiple atoms when in lasso mode), then right-clicking anywhere in the 3D view and selecting **Add excluded volume** from the context menu. Note the excluded volume pharmacophore sphere will include all atoms thus selected.



- From an existing pharmacophore: by right-clicking on a pharmacophore point and then selecting **Add excluded volume**.



Regardless of how an excluded volume pharmacophore point is created, its tolerance radius can be changed (See Changing the Pharmacophore Tolerance) and its position can be changed (See Creating a New Pharmacophore Query).

An excluded volume pharmacophore feature point will result in rejection of any hit with atoms located within the specified volume.

8.3 Modifying a Pharmacophore Query

In CSD-CrossMiner, a pharmacophore point can be modified:

- Before starting the pharmacophore search
- When the pharmacophore search is complete or stopped (by clicking on the *Stop* button)
- During the pharmacophore search itself.

Note that it is not possible to modify a pharmacophore point when the search is paused (by clicking on *Pause* button).

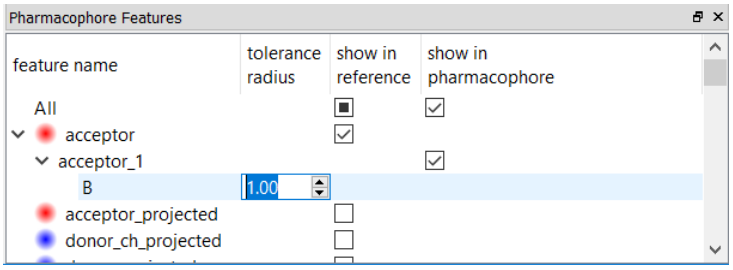
8.3.1 Translating a Pharmacophore Point

A pharmacophore point can be translated by clicking on button in the *Edit:* section of CSD-CrossMiner lower toolbar.

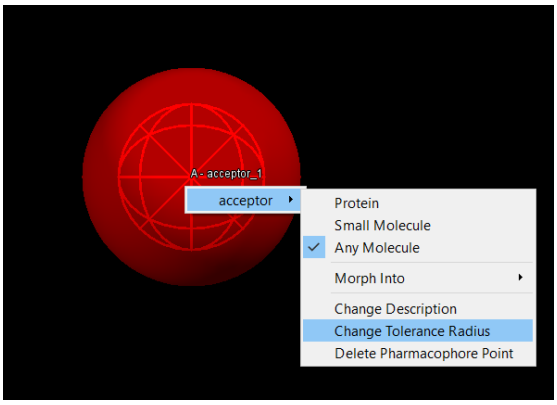
8.3.2 Changing the Pharmacophore Tolerance

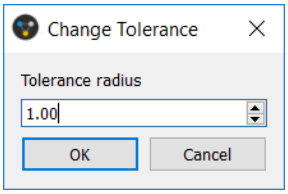
A pharmacophore point by default has a tolerance radius of 1.00 Å. This tolerance radius can be modified in order to increase or decrease the uncertainty in the position of this pharmacophore point in the overall pharmacophore query. The tolerance radius of a pharmacophore point can be changed in three ways:

- By double-clicking on the desired tolerance radius in the *Pharmacophore Features* window to have access to the spin-box and either using the up-down control to change the radius by 0.1 Å increment or entering a desired value in the text box.



- By right-clicking on the pharmacophore point, to have access to the pharmacophore context menu, and double clicking on **Change Tolerance Radius.** The *Change Tolerance* pop-up window allows to change the tolerance sphere of the pharmacophore point by 0.1 Å increment using the spin-box and either using the up-down control or entering a desired value in the text box.

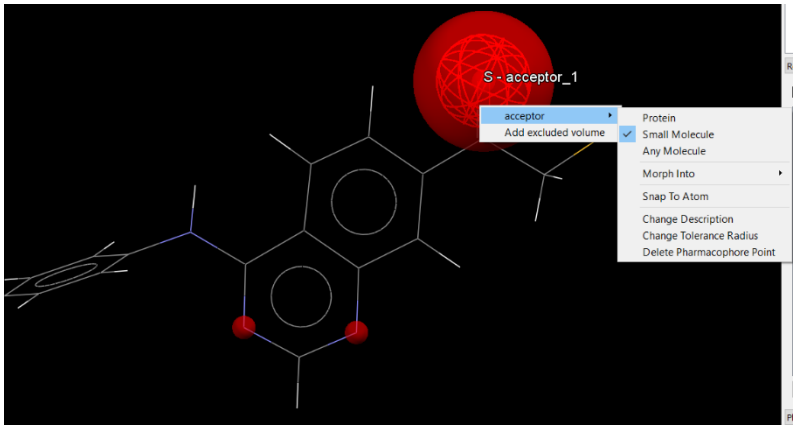




- When in interactive pharmacophore editing mode (having pressed the  button), pressing the middle mouse button (MMB) whilst above a pharmacophore sphere and moving the mouse will increase/decrease the tolerance radius for the pharmacophore point.

8.3.3 Changing the Molecule Type

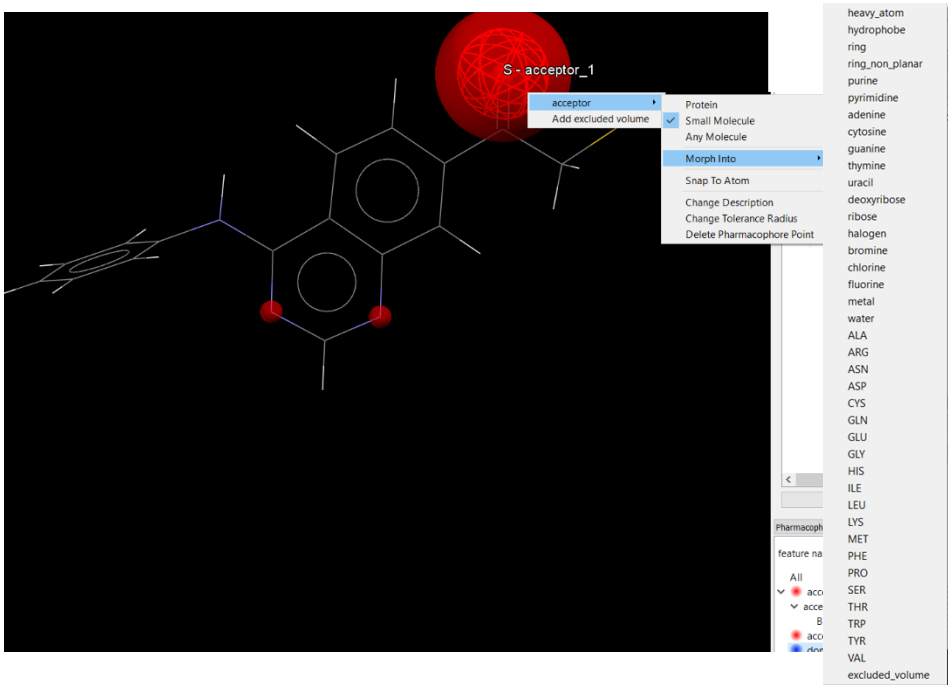
A pharmacophore point can be modified by accessing the pharmacophore context menu, available by right-clicking on a pharmacophore point. The molecule type of a pharmacophore can be changed to be *Protein (P)*, *Small Molecule (S)* or *Any molecule (A*) by selecting **Protein**, **Small Molecule** or **Any Molecule** from the right-click pharmacophore menu for the pharmacophore point. The pharmacophore point labelling in the 3D view is then updated to indicate the chosen molecule type as *P*, *S*, or *A*.



8.3.4 Changing the Pharmacophore Type

A pharmacophore point can be changed into another pharmacophore type by selecting **Morph Into** from the right-click pharmacophore context menu and choosing a new pharmacophore type from the list of available pharmacophore point types.

Directional pharmacophore point(s) can only be morphed into another directional pharmacophore point(s), and equally a one-point pharmacophore point can only be morphed into another one-point pharmacophore point. (See APPENDIX B. Feature Definitions in CSD-CrossMiner for a list of all one-point and directional pharmacophore feature definitions).

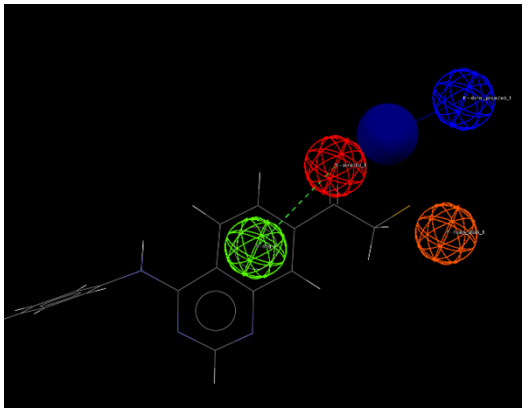


Note that an excluded volume pharmacophore point can be additionally created from an existing one-point pharmacophore: by right-clicking on a pharmacophore point and then selecting **Morph Into** and then **excluded_volume**.

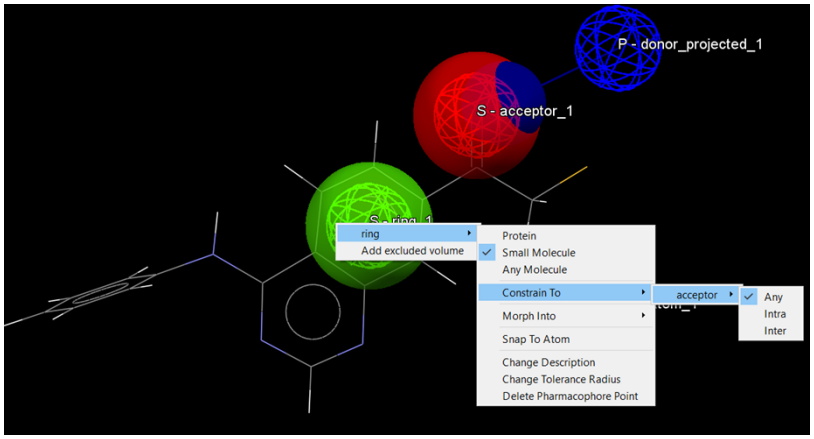
8.3.5 Setting Intramolecular and Intermolecular Constraints

In CSD-CrossMiner it is possible to set intramolecular constraints between all *Small Molecule* and *Any Molecule* pharmacophore points displayed in the 3D view by clicking on 

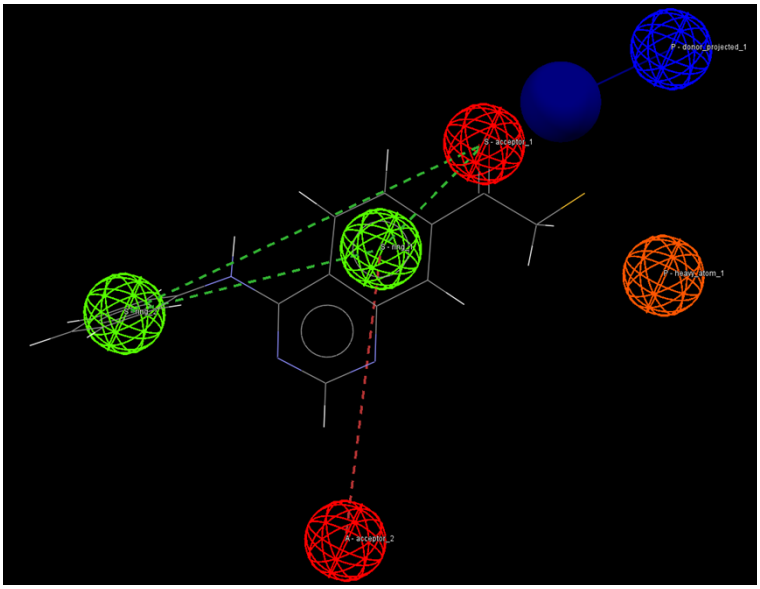
button. The intramolecular constraints are displayed as green dashed lines in the 3D view. It is also possible to create individual intramolecular constraints, intermolecular constraints, as well as 'Any' constraints (where the two pharmacophore points can either belong to the same molecule or not).



To access to these constraint options, right-click on a pharmacophore point and then select **Constrain to…**. This will list all pharmacophore points present in your pharmacophore query. Pharmacophores points of the same molecule type can be constrained to be part of the same molecule (**intra**) or part of different molecules (**inter**). If **Any** is selected both intra- and intermolecular hits may be found when searching with such a pharmacophore query. Intermolecular constraints are represented as red dashed lines in the 3D view.



Note that if a pharmacophore point is set to *Any Molecule (A)* type, both protein and small molecule molecules (including nucleic acids) are considered as a match; therefore, **intra**molecular constraints involving an *Any Molecule* pharmacophore point can only be set with any other *Any Molecule* pharmacophore point, and **inter**molecular constraints can be set with any other *Any Molecule*, *Small Molecule* or *Protein* pharmacophore point.



The multitude of possible combinations of intra- and intermolecular constraints allows for highly tailored interrogations of nonbonded interactions.

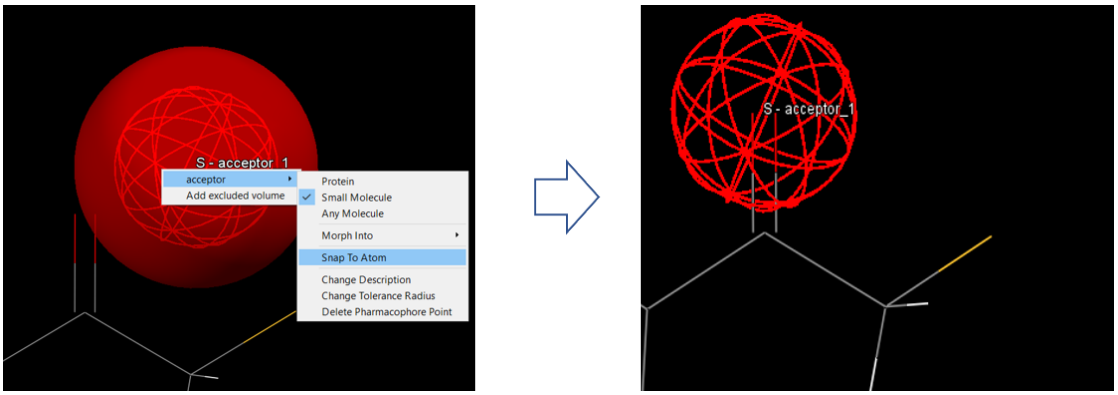
Note that constraints can be enabled/disabled before starting a search, interactively during the search, or when the search is terminated (whether stopped or complete), but not when the pharmacophore search is simply paused.

Changing the molecule type or morphing a pharmacophore point will automatically delete all intra- and intermolecular constraints that this pharmacophore point was previously involved in.

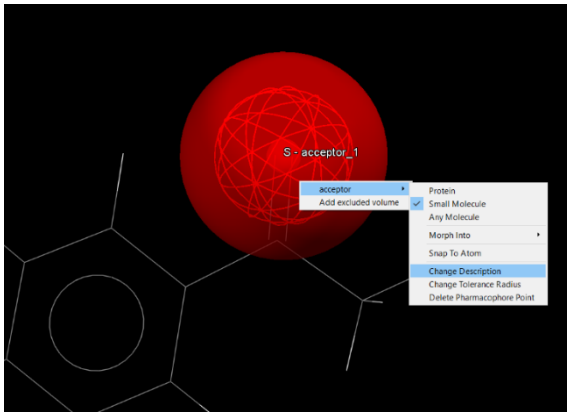
Note that it is not possible to constrain excluded volume pharmacophore points.

8.3.6 Further Editing of the Pharmacophore Point

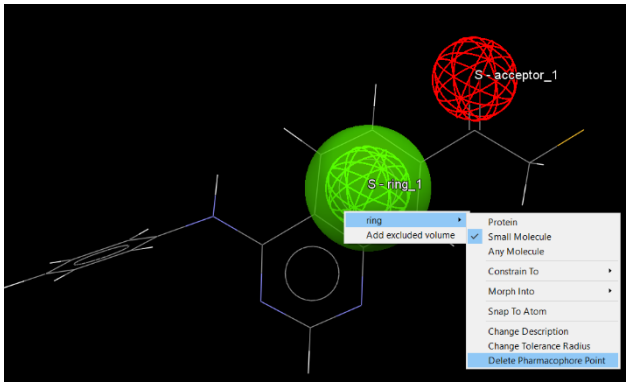
If a pharmacophore point is supposed to coincide with an atom position, it is possible to drag the pharmacophore sphere close to the atom and select **Snap To Atom** from the right-click context menu of the pharmacophore point.



Upon accessing the right-click context menu of a pharmacophore point, it is also possible to change its labelling in the 3D view by selecting **Change Description**. The *Change Feature Description* pop-up window invites the user to enter a new description.



Finally, it is possible to delete a pharmacophore point by selecting **Delete Pharmacophore Point** in the right-click context menu.



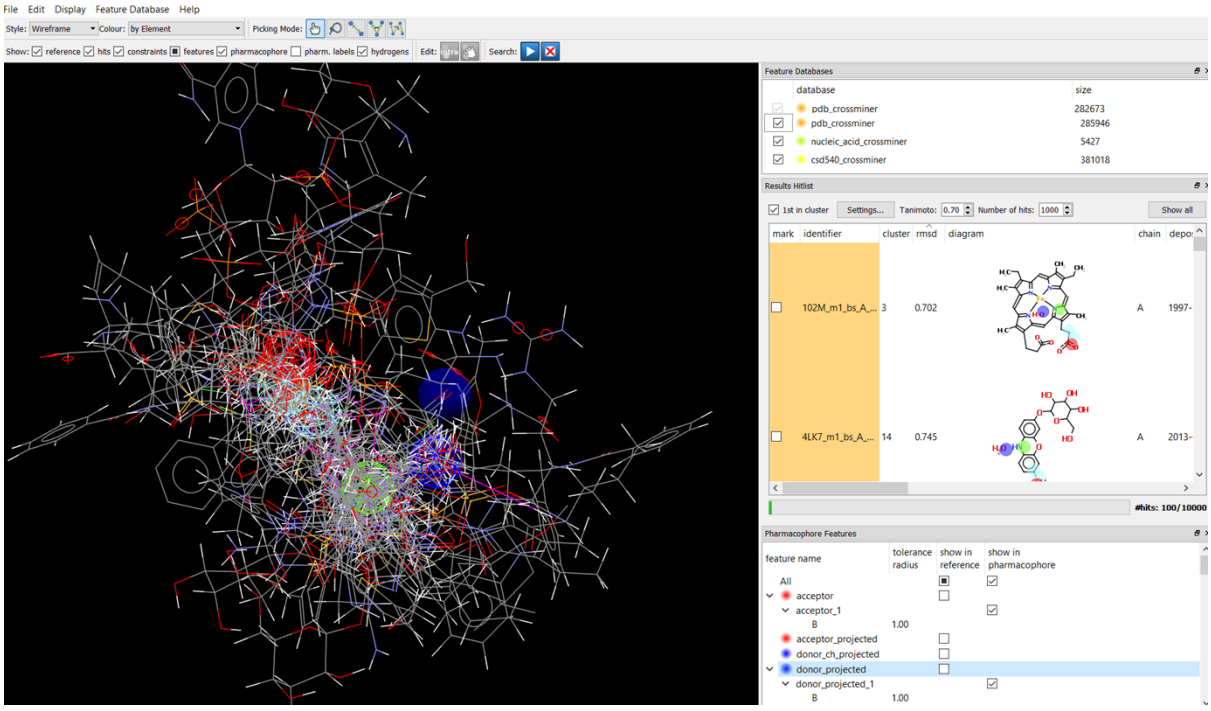
8.4 Saving a Pharmacophore Query

After creating and/or editing a pharmacophore, the resulting pharmacophore query can be saved, either in the CSD-CrossMiner pharmacophore format (*.cm*) by clicking on **File** in the CSD-CrossMiner top-level menu and then selecting **Save Pharmacophore**, or in PyMOL pharmacophore format (*.py*) by clicking on **File** and then selecting **Save PyMOL Pharmacophore**.

A pharmacophore query in the CSD-CrossMiner pharmacophore format can be loaded into CSD-CrossMiner using **File** > **Load Pharmacophore**. Note that a pharmacophore query in the PyMOL format cannot be loaded into CSD-CrossMiner but is compatible with third party software such as PyMOL.

9 Pharmacophore Search

Once the pharmacophore query has been created, a pharmacophore search can be started by pressing the *Start* button in the CSD-CrossMiner toolbar. The loaded feature database will be investigated to find hits matching the pharmacophore query, and both the 3D view and the *Results Hitlist* browser will be updated with the identified hits:



Note that the number of hits in the progress bar will update every 5 hits. During the pharmacophore search, any change in the topology or geometry of the pharmacophore query triggers a new database search and consequently both the 3D view and the *Results Hitlists* browser update, giving the user instant feedback on the kind of hits to be expected from the current query setup. During the search, the *Start* button turns to a *Pause*  button that can be used to pause the pharmacophore search ( is then restored and can be used to continue the search). Note that in pause mode, the pharmacophore query cannot be interactively changed.

The hits matching the pharmacophore query can be selected in the *Results Hitlist* browser and visualised in the 3D view, marked and/or sorted anytime during the pharmacophore search or when the search is paused (See Results Hitlist and Results Hitlist Browser).

The pharmacophore search can be stopped by pressing the *Stop*  button; this will clear the *Results Hitlist* browser and restore the *Start* button. When the pharmacophore search is stopped or is complete, the pharmacophore query can be edited again and a new search can be started.

The pharmacophore search starts with the conversion of the pharmacophore query, defined in Cartesian space by the tolerance spheres, into a distance space representation (fingerprint). For each pair of tolerance spheres, a distance constraint is derived by measuring the distance between the two sphere centres and adding, as well as subtracting, the sum of the sphere radii to obtain upper and lower bounds, respectively. Additionally, each distance constraint stores whether an intra- or intermolecular constraint has been defined for each pair of pharmacophore points, and whether a pharmacophore point is constrained to be part of a small molecule, a protein and/or any component.

The resulting pharmacophore query fingerprint can then be compared to the respective fingerprint of any database entry. Each bit set in the query fingerprint is required to be present in the database entry fingerprint. Therefore, for a database structure to be a hit against the query, it must contain all pharmacophore points of the pharmacophore query. It is only if the fingerprint comparison passes that a database entry is subjected to a 3D search using the pre-calculated feature points.

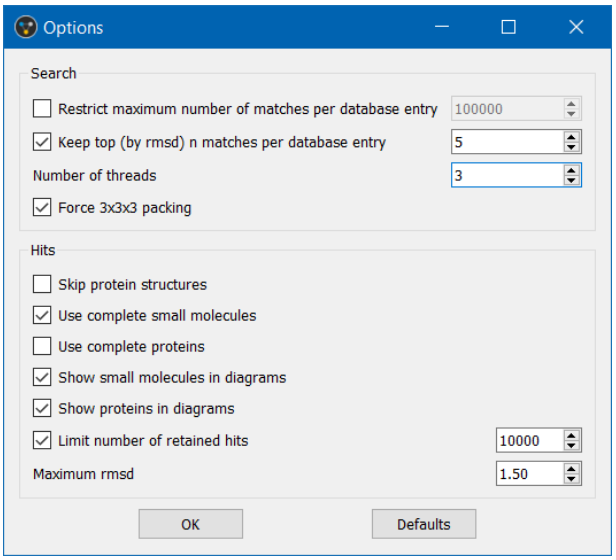
It is necessary to perform a final Cartesian space overlay check on full matches, since not all matches in distance space between a pharmacophore query and a database entry correspond to matches in Cartesian space.

Since for a single database entry multiple matches may have been identified, for each hit, CSD-CrossMiner calculates the minimum overlay root-mean-square deviation (rmsd) of the point coordinates in the match with respect to the pharmacophore sphere centres. This is used to obtain a unique ranking of the matches. The number of matches per entry can be customised by the user (See Pharmacophore Search Options).

For all feature database entries that match the pharmacophore query, the molecular structures are loaded from the structure database, and matching hits overlaid onto the pharmacophore query are shown in the 3D view.

9.1 Pharmacophore Search Options

The pharmacophore search options can be accessed by clicking on **Edit** in the CSD-CrossMiner top-level menu and selecting **Options**. Note that these are not available to be changed (and **Options** is greyed-out) when the pharmacophore search is running or paused.



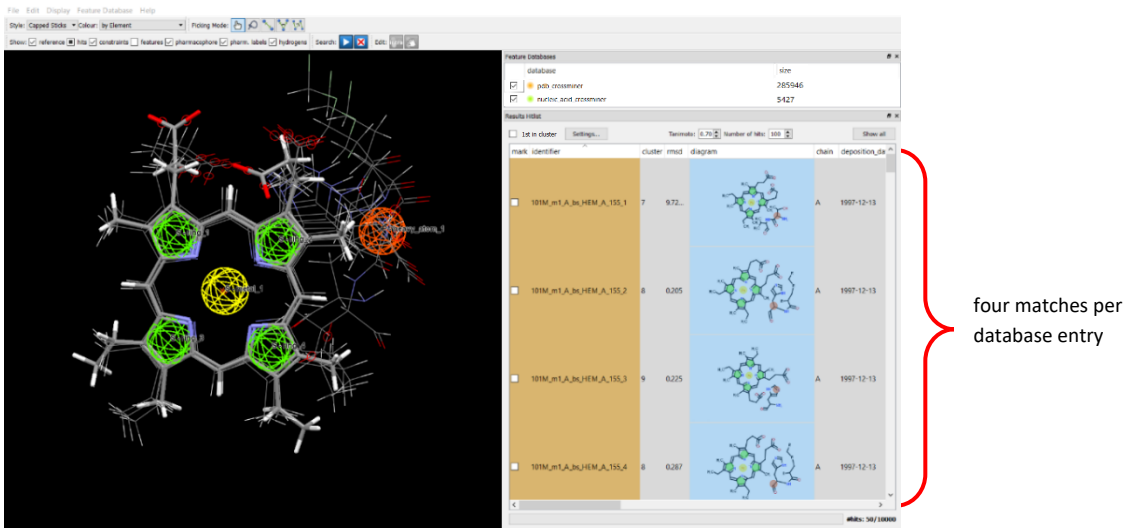
The *Search* section of the *Options* window contains modifiable settings for the pharmacophore search itself. The *Hits* section of this window contains modifiable settings for the processing of hits found by the pharmacophore search (*i.e.* clustering of hits and display of hits in the 2D diagram of the *Results Hitlist* window and in the main 3D view of CSD-CrossMiner).

In the *Search* section of the *Options* window the user has access to **Restrict maximum number of matches per database entry** and **Keep top (by rmsd) n matches per database entry**, two options which allow the user control over the number of matches returned per database entry in a search.

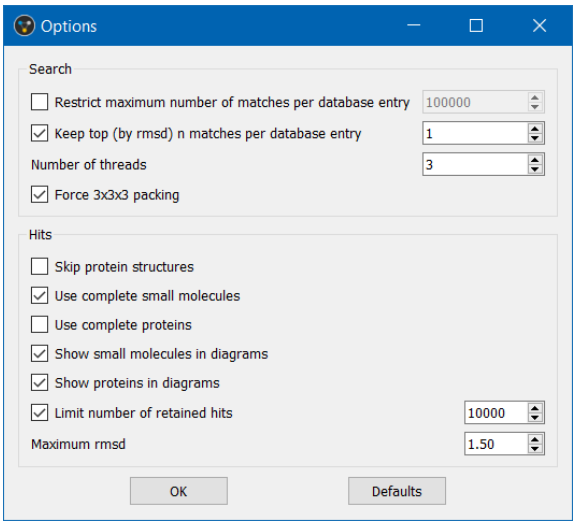
The first option sets after how many hits per database entry the pharmacophore search is terminated (by default, no restriction is applied, and all possible hits are generated for each matching database entry).

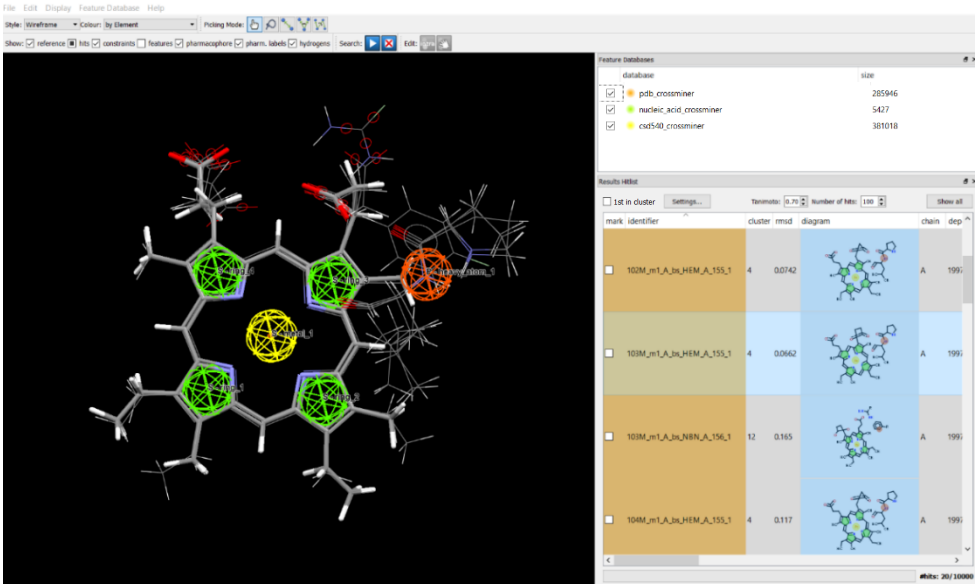
The second option sorts all hits per database entry according to rmsd and returns the top n.

By default, CSD-CrossMiner will return a maximum of five matches per database entry. Tweaking this option is useful when multiple matches in the same entry are returned. For example, in the case of symmetrical queries, such as a heme group with a pharmacophore query defined by four rings, a metal atom and one protein heavy atom, leaving the default value in **Keep top (by rmsd) n matches per database entry** would lead to four matches per entry (one for each ring).



Reducing the number of matches per database entry to one for example would reduce the redundancy of the solutions and would provide only one match per database entry.



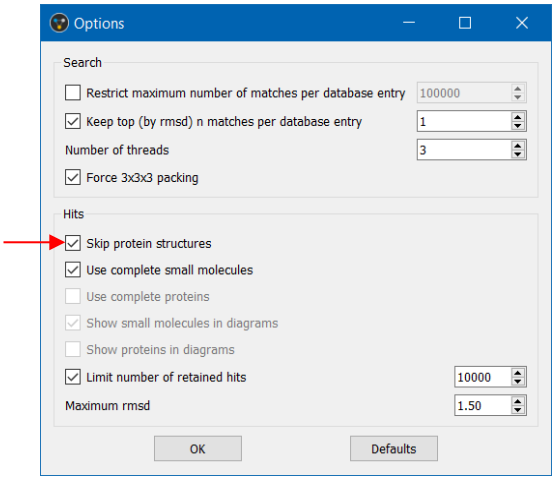


Via **Number of threads** it is possible to specify the computational resources to dedicate to the pharmacophore search, by defining that the database search be distributed to the specified number of CPU cores, if available.

By default, the **Force 3x3x3 packing** tick-box is ticked, which restricts the search to 26 unit cells around the central unit cell. This allows symmetry-related copies of the feature points to be considered for a small molecule crystal structure database entry that matches.

The *Hits* section of the *Options* window contains all settings for the processing and clustering of hits during a pharmacophore search.

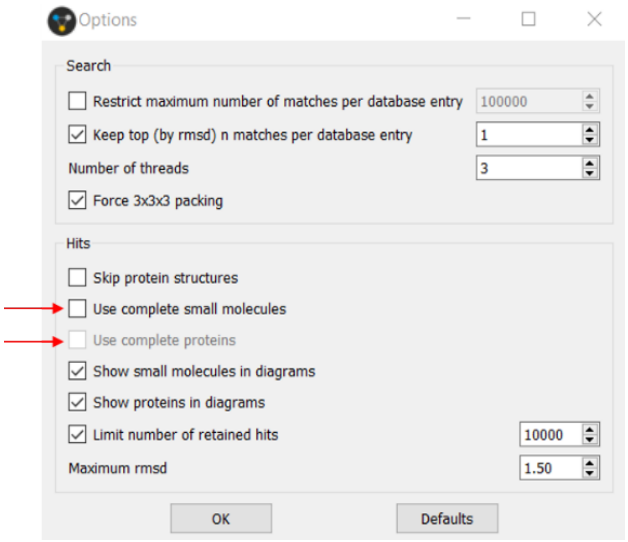
Ticking the **Skip protein structures** option (unticked by default) results in the protein components of a matching database entry (although used for the pharmacophore search itself) being omitted in the hit clustering and not being displayed in the 2D diagram or in the 3D view. Only the small molecule components would thus be used for the hit clustering and would be displayed in the 2D diagram and 3D view.



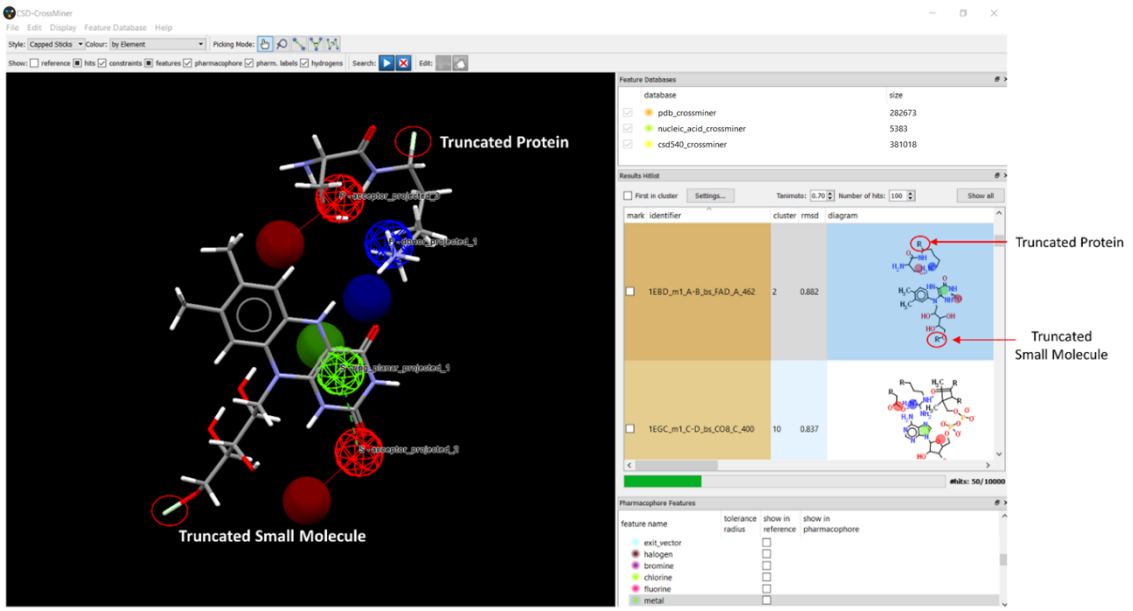
For a given pharmacophore search query, CSD-CrossMiner calculates a bounding sphere around the entire pharmacophore, which is an approximation of the overall smallest sphere that encompasses all pharmacophore points of the query and with a sphere radius at least equal to the largest sphere radius among the pharmacophore points plus 1.5 Å. When matched structures are overlaid onto the pharmacophore search query, there may be protein and/or small molecule atoms of the hit that are outside this bounding sphere. This is relevant for the **Use complete small molecules** and **Use complete proteins** options.

Ticking the **Use complete small molecules** option (ticked by default) results in the entire small molecule being used in clustering and displayed in 2D/3D, even if some small molecule parts are outside the pharmacophore bounding sphere. Similarly, the **Use complete proteins** option (unticked by default) can be ticked so that the entire protein binding site, as obtained from the structure database, is used for clustering and in the 3D display, even if some protein parts are outside the pharmacophore bounding sphere. Note that, for ease of visualisation, by ticking this option the protein component will not be shown in the 2D diagram.

Note that it is not possible to tick only **Use complete proteins** tick-box if the **Use complete small molecules** tick-box is unticked.



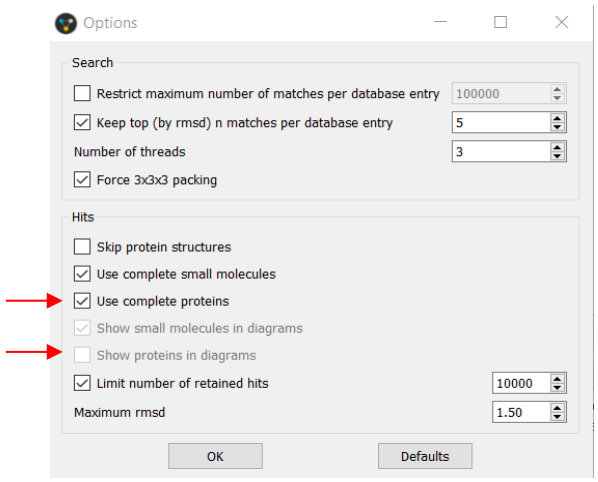
If the **Use complete small molecules** and/or **Use complete small proteins** tick-boxes are unticked, the part of the small molecule and/or protein component that is outside the pharmacophore bounding sphere will be truncated in both the 2D diagram and in the 3D view.



Ring systems (plus attached substituents) are always kept as intact units if at least one ring atom is inside the bounding sphere. For bonds which have one atom located inside and the other one located outside the bounding sphere, the latter atom will be replaced by an R-group symbol in the 2D diagram and light green R atom in the 3D view.

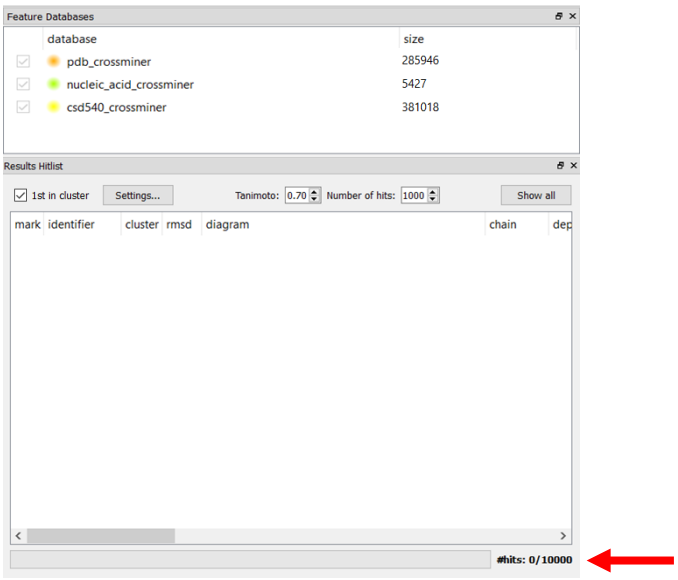
In addition, unticking the **Use complete small molecules** and/or **Use complete proteins** tick-boxes will affect the way CSD-CrossMiner performs clustering (See Clustering Algorithm and Clustering Settings).

The **Show small molecules in diagrams** and **Show proteins in diagrams** options are ticked by default, allowing the user to see both small molecules and proteins in the 2D diagram in the *Results Hitlist* browser. For easy visualisation, when the **Use complete proteins** tick-box is ticked, the **Show proteins in diagrams** option will automatically be disabled.

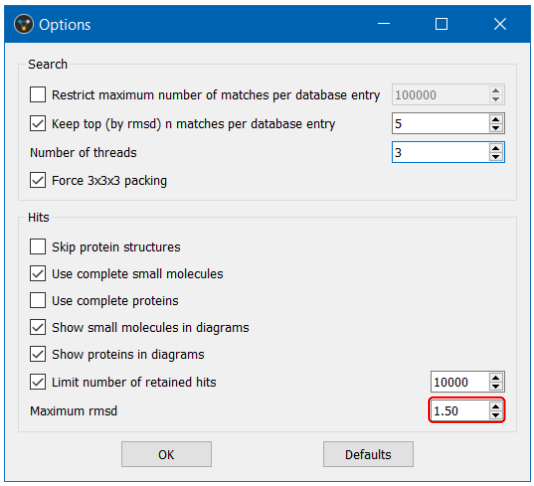


The **Show small molecules in diagrams** and **Show proteins in diagrams** options can be individually disabled by unticking the desired tick-box, such that the 2D diagram can contain only the small molecule component or only the protein component.

The **Limit number of retained hits** option defines how many hits will be retained in the 3D view and in the *Results Hitlist* browser. The default number of retained hits is 10 000 and is shown in the pharmacophore search progress bar. Changes in the number of retained hits will automatically update the upper limit in the pharmacophore search progress bar.



The rmsd upper limit for a pharmacophore match to be added to the hit list can be also edited by modifying the **Maximum rmsd** option. If tolerance spheres with large radii are used, this value may need to be modified accordingly.



By clicking **OK**, the new pharmacophore search option settings are saved and will be retained between separate CSD-CrossMiner sessions. However, it is possible to restore the default settings by clicking on the **Defaults** button in the *Options* window.

10 Clustering Algorithm and Clustering Settings

Although a pharmacophore search may provide the means for screening many compounds, this may be undesirable because resources may be wasted if this large-scale effort results in the production of redundant information. Clustering the solution space in real time provides a powerful help to remove such redundancy, thus allowing the user to quickly grasp the diversity in ligand topology and protein-ligand interactions found across the searched database.

The clustering algorithm in CSD-CrossMiner generates two similarity fingerprints: a small molecule fingerprint and a protein fingerprint. The small molecule fingerprint enumerates any small molecule substructure features that are present in the hit substructure. Similarly, the protein substructure features present in the hit are hashed in the protein fingerprint.

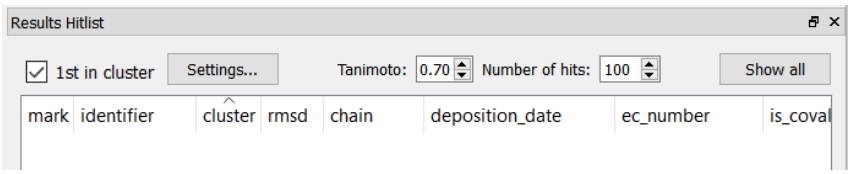
The hits matching the pharmacophore query are subject to two different clusterings: on-the-fly clustering (during the pharmacophore search itself) and post-search clustering (when the search is complete).

During the on-the-fly clustering, for each new hit the algorithm will loop through all current cluster representatives. If there are no cluster representatives within the user-defined Tanimoto threshold, the new hit will be added to the set of cluster representatives (*i.e.* this will create a new cluster, with cluster number *n+1* if there are already *n* clusters); otherwise it will be discarded (*i.e.* it will not create a new cluster and not become a cluster representative; however, it will be saved as a hit). During the search, this clustering method is heuristic, as it is possible that a new hit may have a lower rmsd than its cluster member; this will not be accounted for by on-the-fly clustering.

Note that the 3D view is constantly updated during the search by adding any new hit that has become a cluster representative (*i.e.* any new hit that has been assigned a new cluster number in the *Results Hitlist* browser). The result is that if selected hit changes its cluster representative status (*i.e.* discarded as a cluster representative), then all hits will be displayed in the 3D view.

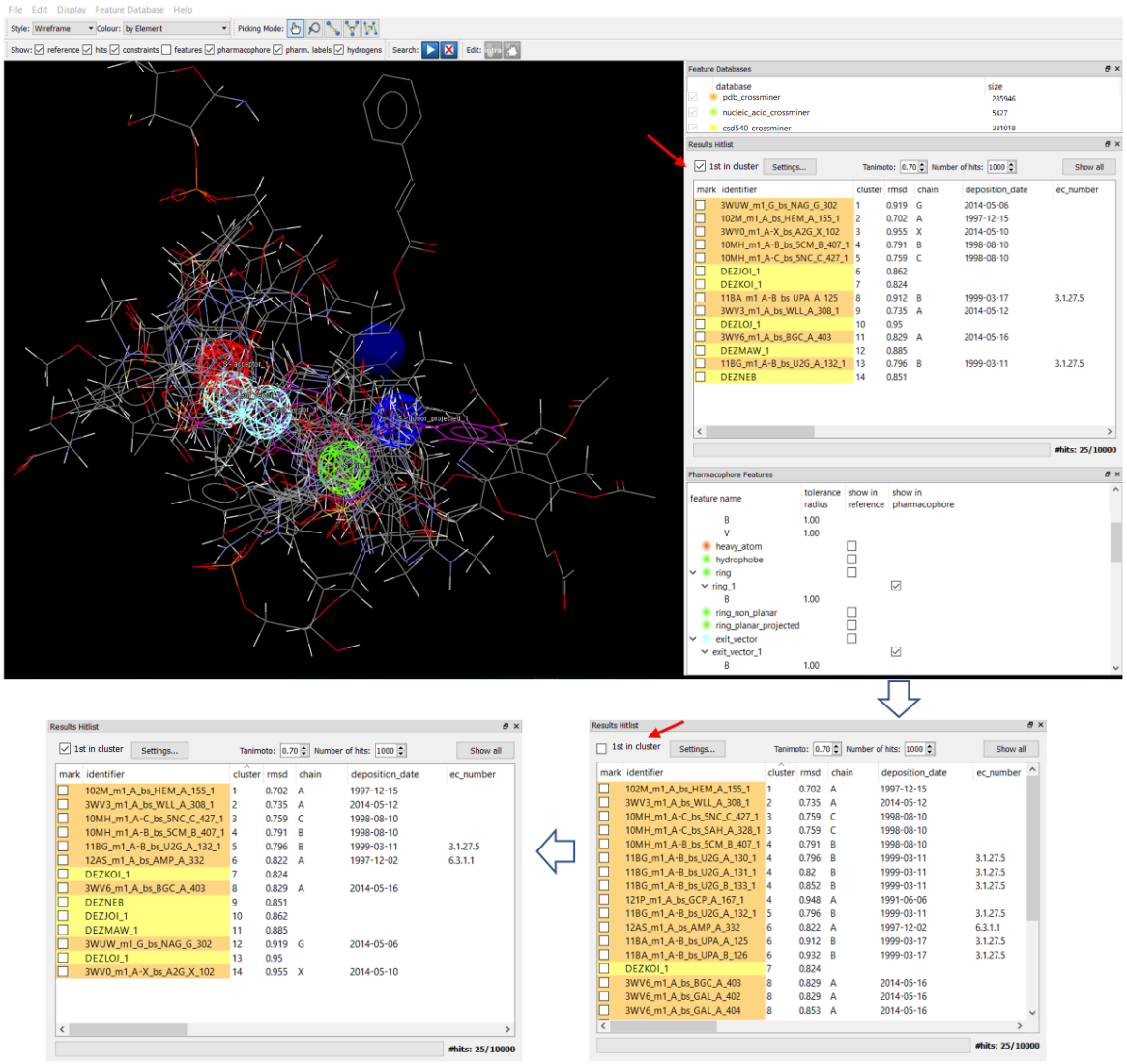
The post-search clustering will instead sort all hits by rmsd once the search is complete. The algorithm will search in all hits matching the pharmacophore query for the cluster representatives, within the user-defined Tanimoto threshold. This time the cluster representative with the best rmsd will be selected.

The clustering settings are displayed in the *Result Hitlist* window and can be edited at any time during a pharmacophore search.



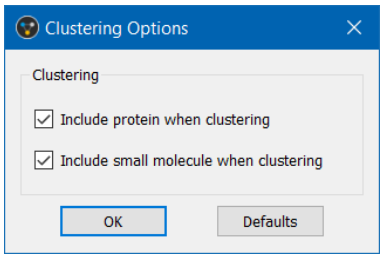
By default, the **1st in cluster** tick-box is ticked, such that it is only the cluster representative of each cluster that is shown in the *Results Hitlist* browser and in the 3D view. All cluster members can be

visualised by unticking the **1st in cluster** tick-box any time during and after the pharmacophore search. Note that this will increase the number of hits listed in the *Results Hitlist* browser and displayed in the 3D view.



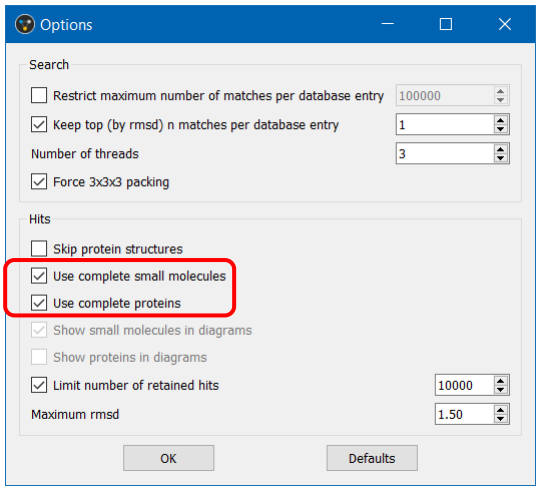
Note that unticking and then re-ticking this **1st in cluster** tick-box whilst the search is running, will activate the post-search clustering and the cluster representative with the best rmsd out of all hits found by this point of the search in progress will be selected. This may result in a different cluster representative being listed in the *Results Hitlist* browser and displayed in the 3D view. By default, two structures are deemed 'similar' if both the small molecule and protein fingerprints have a Tanimoto threshold within 0.7 Å. The user can tailor this threshold by entering a new value in the **Tanimoto** spin-box in the header of the *Results Hitlist* window (See Results Hitlist and Results Hitlist Browser). Please note that the default Tanimoto threshold of 0.7 Å will be restored between separate CSD-CrossMiner sessions. Additionally, the **Settings…** button in the header of the *Results Hitlist* window allows the user to configure the clustering settings by choosing whether to include the protein and/or

the small molecule fingerprint in the *Cluster Options* pop-up window. Please note that any modification of these clustering settings will be retained between separate CSD-CrossMiner sessions.



If the pharmacophore search options are set to the Defaults (See Pharmacophore Search Options), then the small molecule fingerprint and the protein fingerprint will be created by enumerating the pharmacophore features for all small molecule atoms and for those protein atoms within the bounding sphere of the pharmacophore, respectively. However, the user can change this clustering behaviour using the **Options** dialogue available through the CSD-CrossMiner top-level **Edit** menu.

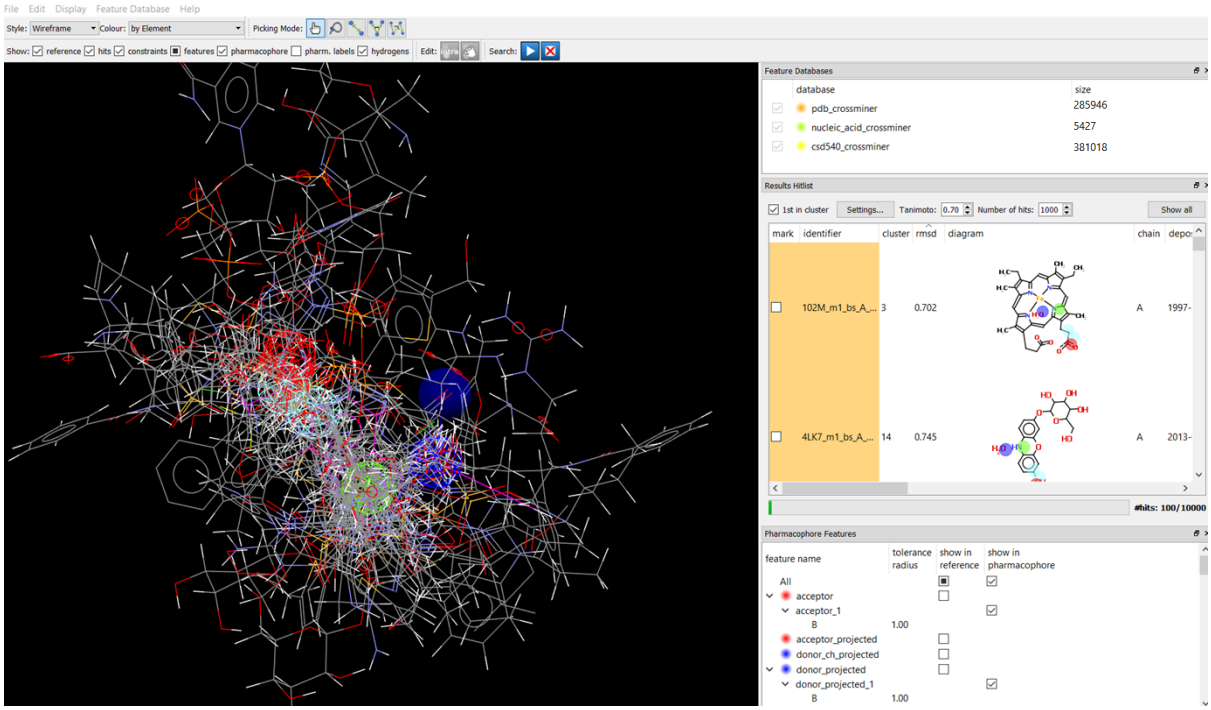
By additionally ticking the **Use complete protein** tick-box, the clustering algorithm will include all atoms in the protein binding site into the protein fingerprint (See Pharmacophore Search Options), instead of only those within the pharmacophore bounding sphere if this tick-box remained unticked as per default.



11 Results Hitlist and Results Hitlist Browser

In CSD-CrossMiner, all hits matching the pharmacophore query (up to a user-defined rmsd limit, see Pharmacophore Search Options) are collected and (up to a user-defined number of hits) are displayed in the 3D view as well as listed in the *Results Hitlist* browser. The first five columns in the *Results Hitlist* window contain: the ability to mark an entry (by putting a tick in the column *mark*), the entry name of each database structure (*e.g.* `102M_m1_A_bs_HEM_A_155` in the example below) followed by the number of time the entry matches the pharmacophore query as underscore (*identifier*), (up to the top number of matches per database entry defined in the **Options**, See Pharmacophore Search Options)

(*e.g.* `102M_m1_A_bs_HEM_A_155_1` in the example below), the number of the cluster (*cluster*), the rmsd (*rmsd*), and the 2D diagram of the pharmacophore overlay match (*diagram)*.



Further columns in the *Results Hitlist* window contain any other information stored in the feature database as annotations (See Annotating a Feature Database).

The hits can be sorted according to any text column in the *Results Hitlist* window at any point during or after the pharmacophore search. For example, hits can be sorted according to rmsd values or according to the number of the cluster by clicking on the *rmsd* or *cluster* column label in the *Results Hitlist* browser. A hit can be selected by clicking on it (multi-selection is enabled, with CTRL+LMB to select individual hits or with SHIFT+LMB to select a continuous list), which will automatically display the selected hit(s) in the 3D view. Note that when selecting hits during the pharmacophore search, it is possible that selected hits (even if retained as matched hit) will disappear from the 3D view and from the list in *Results Hitlist* browser. This can happen because the *Results Hitlist* window shows only a limited number of hits and it updates during the pharmacophore search.

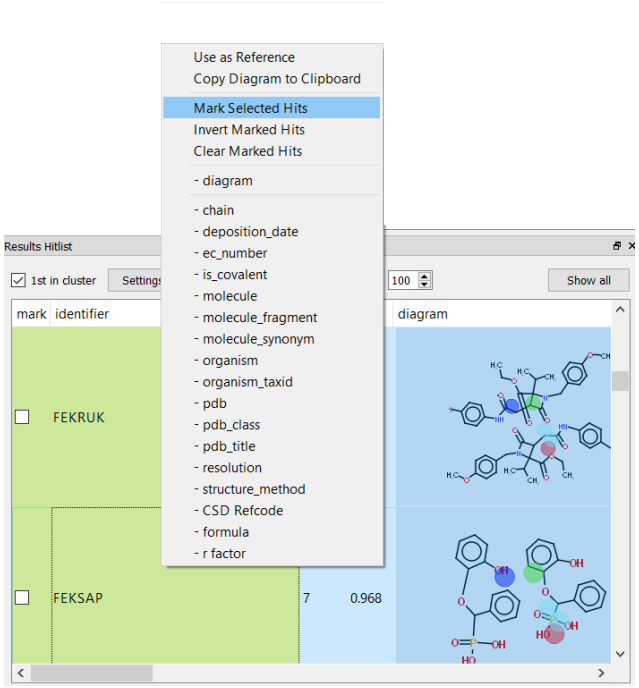
The default number of displayed hits is set to 100; however, this number can be changed (up to 1000) at any point before, during or after the search by using the **Number of hits** spin-box in the *Results Hitlist* window. The edited **Number of hits** value will then be retained between separate CSD-CrossMiner sessions.

Note that whilst the maximum number of hits displayed in the *Results Hitlists* browser can be varied an overall list of all hits (up to the number of matches per database entry defined in **Edit** > **Options**) is maintained at all times.

Once a hit is selected in the *Results Hitlist* browser, the **up** and **down arrow** keys, **Page Up** and **Page Down** keys and **Home** and **End** keys can be used to browse through the list.

Interesting hits (*i.e.* clusters of particular interest) can be marked by ticking the respective tick-boxes in the *mark* column and saved for later inspection using **Save Marked Hits** in the CSD-CrossMiner top-level **File** Menu (See Exporting Hits).

Selected hits can be also marked by right-clicking in the *Results Hitlist* browser and select **Mark Selected Hits.** Note that, the new marked hits will be added to the previously marked hits, if present.



Through the right-click menu it is also possible to invert the marked hits using the **Invert Marked Hits** option and clear the marked hits by using the **Clear Marked Hits** option.

All hits listed in the *Results Hitlist* browser can be displayed in the 3D view by clicking the **Show all** button in the header of the *Results Hitlist* window.

When multiple hits are displayed in the 3D view (but not selected in the *Results Hitlist* browser), it is possible to select a specific hit by clicking on any of its atoms in the 3D view. This will hide all other hits from the 3D view.

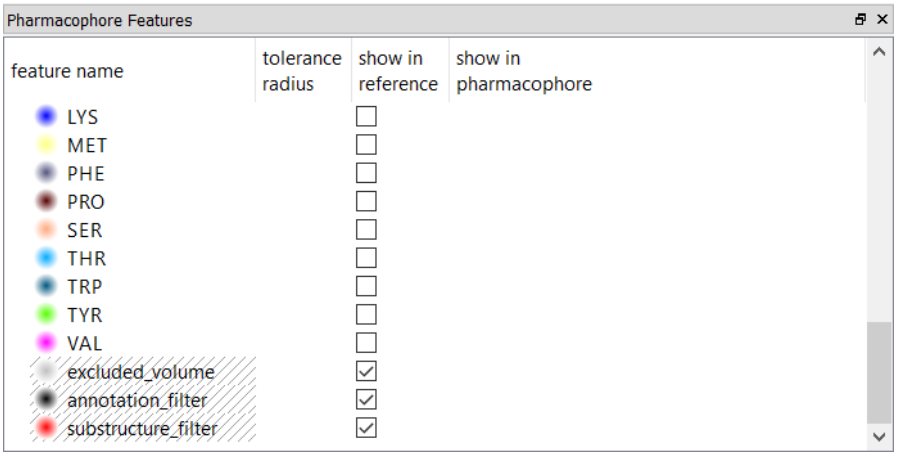
By default, the 2D diagram of any hit matching the pharmacophore query is shown in the *Results Hitlist* browser; however, it can be hidden by right-clicking in the *Results Hitlist* browser and selecting the -*diagram* option of the right-click menu. The size of the 2D diagram can be also changed by increasing or decreasing the *diagram* column while left-clicking on the *diagram* column delimiter in the *Results Hitlist* window.

In addition, by accessing the **Options** window from the CSD-CrossMiner top-level **Edit** menu (See Pharmacophore Search Options) it is possible to control what is contained in the 2D diagram: the small molecule, protein or both components by ticking/unticking the **Show small molecules in diagrams** and/or **Show proteins in diagrams** tick-boxes.

By default, all information (defined as annotations) stored in the loaded feature database are shown in the *Results Hitlist* browser. As for the 2D diagram, these annotations can be disabled individually by right-clicking in the *Results Hitlist* browser and selecting the desired annotation. Furthermore, the annotations can be used to filter the matched hits (See Using Annotations as Filter).

12 Filtering in CSD-CrossMiner

In CSD-CrossMiner is possible to filter the matching hits based on the presence/absence of specific substructure(s) and/or by annotations. The *substructure_filter* and *annotation_filter* features are available at the bottom of the *Pharmacophore Features* window. Because both *substructure_filter* and *annotation_filter* are not indexed in the feature database, they are represented with diagonal hatching in the *Pharmacophore Features* window.



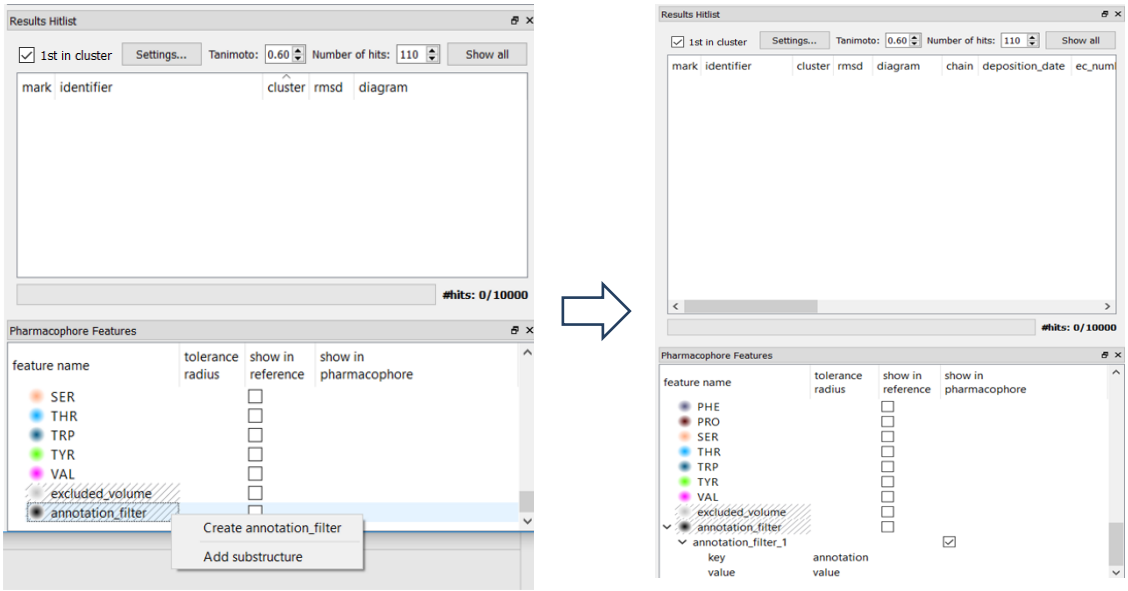
Note that the *substructure_filter* and *annotation_filter* can only be added before starting a pharmacophore search and/or after a pharmacophore search is stopped or completed.

12.1 Using Annotations as Filter

If the searched feature database contains annotations (as does the supplied feature database), these can be used to filter the database from which hits can be found using the *annotation_filter* listed in the *Pharmacophore Features* window.

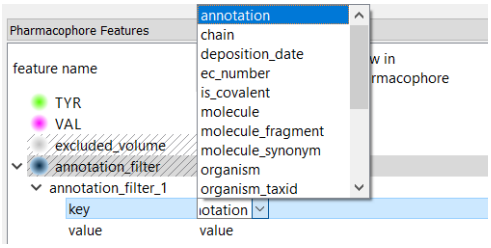
An *annotation_filter* is a specialised feature type that defines a textual filtering rule instead of pharmacophore feature spheres and thus is represented with diagonal hatching to differentiate it from indexed pharmacophore feature types.

An annotation filter can be created by right-clicking on the *annotation_filter* feature listed in the *Pharmacophore Features* window and then selecting on **Create annotation_filter**. Note that it is not possible to create and/or edit an annotation filter when the pharmacophore search has been paused, but only when it has not yet started or has been stopped.



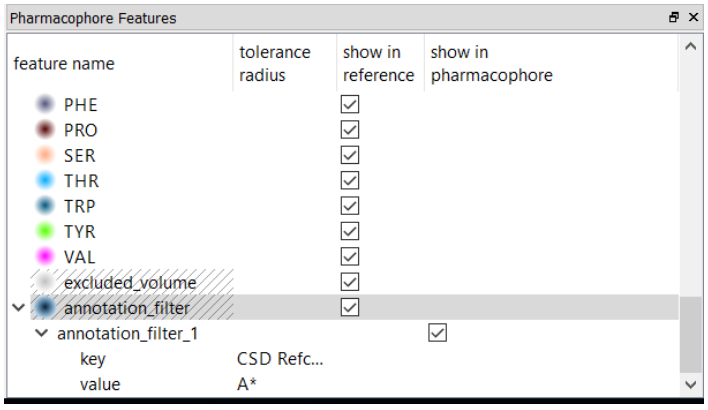
An *annotation_filter* is composed of two parts: a *key* and a *value,* where the key corresponds to one of the column headers listed in the *Results Hitlist* window and the value corresponds to the content in that specific column.

The list of all annotation keys for the loaded feature database is accessible by double clicking on the *annotation* key.

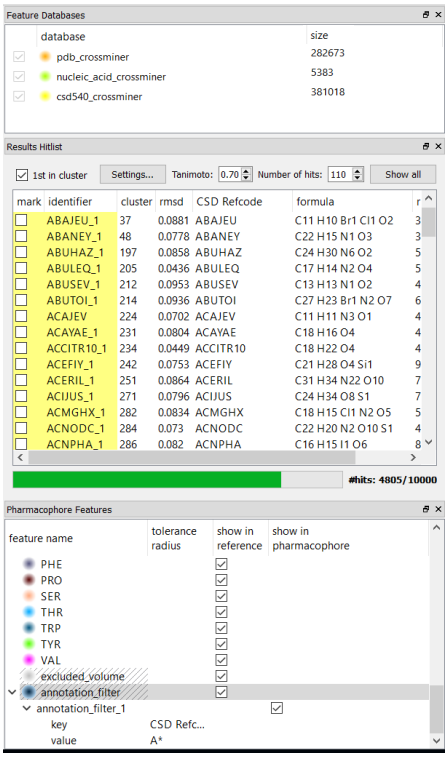


Note that this list will change if further or different annotations are added to the feature database (See Annotating a Feature Database).

A mismatch between a specified *annotation_filter* and the annotation associated with an entry will result in rejection of any putative hits from that entry.



In the above example, the key labelled *CSD Refcode* will be compared with the value rule *A**. This would result in showing only hits that belong to entries with a CSD REFCODE that begins with the letter 'A' (matching A*) and all other hits being rejected:



If an annotation is numeric, e.g. resolution for a PDB entry, the annotation is filtered as numeric value. Numeric operators such as *<, <=, >=, =* can be additionally used. Numeric annotations can be filtered by ranges using the numeric operator - (*e.g.* 1.5-2.5).

Note that multiple annotation filters can be added in a pharmacophore search query.

12.1.1 Filtering Matching Rules

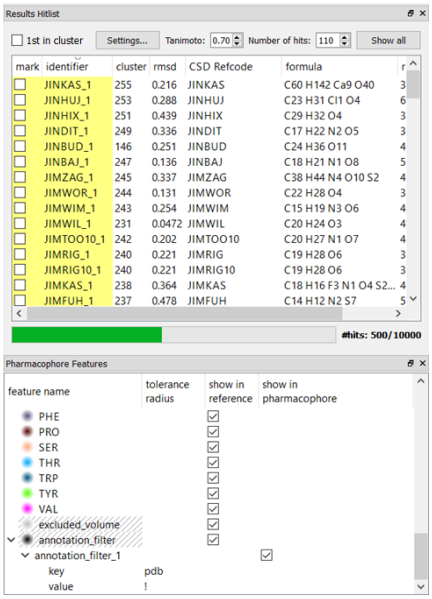
Filtering matching understands a few wildcard rules. These rules resemble the UNIX shell wildcards:

- **?** - Matches any one single character.
- ***** - Matches zero or more of any characters.
- **[...]** - Matches any one of the set of characters listed within the square brackets; *e.g.*, "[0123456789]" will match a numeric character only.
- **** - Escape character to treat the special character that follows as a literal character; *e.g.*, "\?" will match a "?" character only.

Any other character represents itself apart from those described above; *e.g.* "a" matches the character "a".

In addition to UNIX shell wildcards, an initial character of "!" will negate the comparison. For instance, "A*" will select every value beginning with "A" while "!A*" will select every entry not beginning with "A".

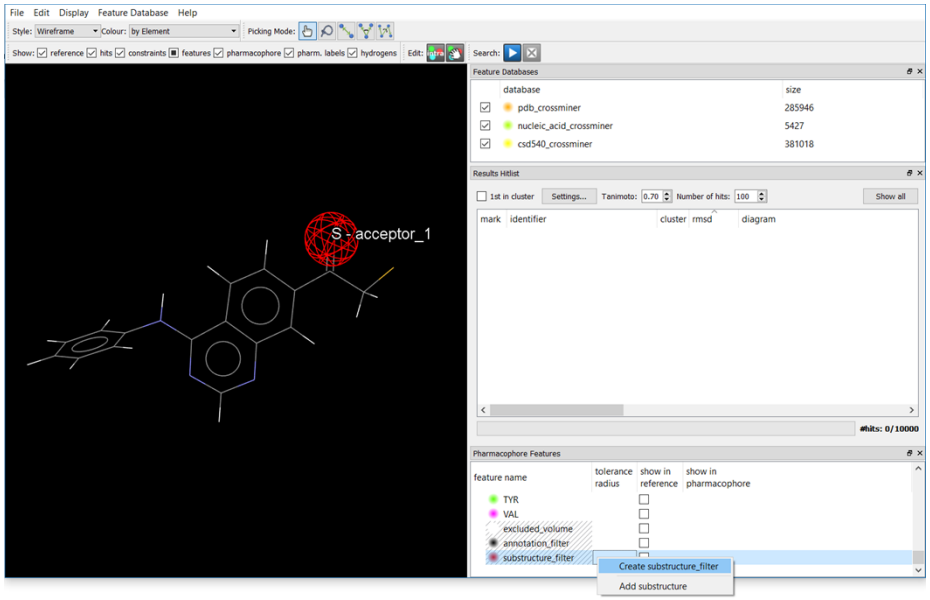
An empty value in the *value* text-box is treated as a check for existence of the annotation. This can also be negated, so the rule "!" will select every entry for which the specified annotation does not exist. In the example below, the key labelled *pdb* will be compared with the negation value rule *!*. This would result in showing only hits that do not have a PDB code.

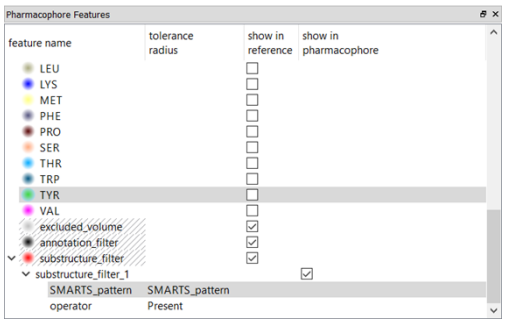


12.2 Substructure Filter

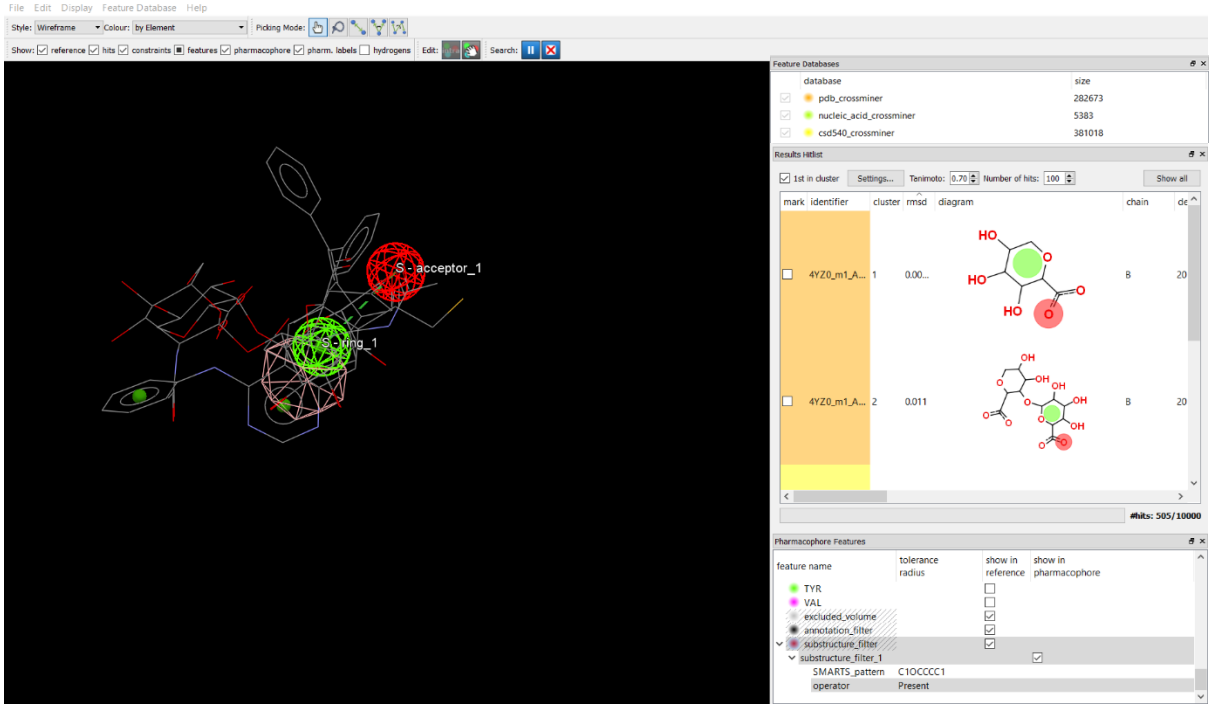
A *substructure_filter* is a specialised feature type that defines a SMARTS format filtering rule instead of pharmacophore feature spheres and thus is represented with diagonal hatching to differentiate it from indexed pharmacophore feature types. This filter can be used to constrain the pharmacophore search by the presence or absence of a specific substructure. The SMARTS format is simply a language for describing substructure patterns in molecules as a simple string of ASCII characters.

A substructure filter can be created by right-clicking on the *substructure_filter* feature listed in the *Pharmacophore Features* window and then selecting on **Create substructure_filter**. Note that it is not possible to create and/or edit a substructure filter when the pharmacophore search has been paused, but only when it has not yet started or has been stopped.

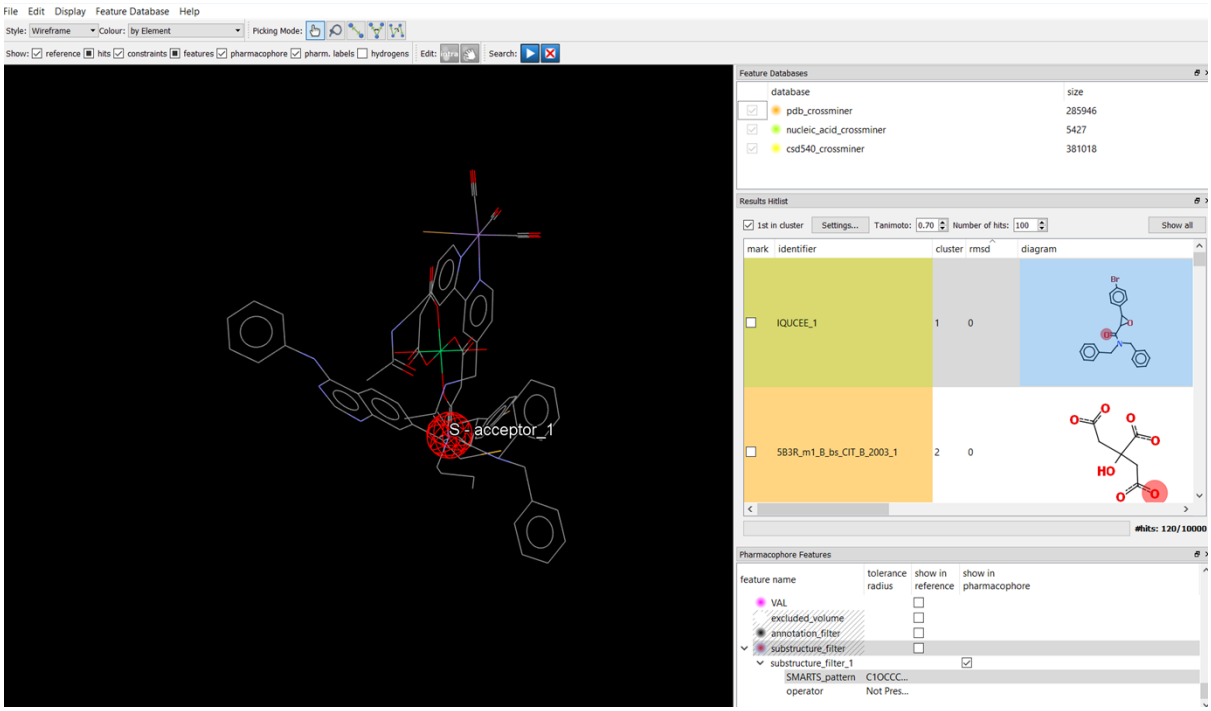




A *substructure_filter* is composed of two parts: a *SMARTS_pattern* and an *operator,* where the SMARTS pattern corresponds to the desired substructure (e.g. C1OCCCC1 in the example below) and the operator can be set to *Present* or *Not Present,* this will result in matched hits that contain or not contain the specified SMARTS pattern, respectively. In the example below, the set substructure filter will retrieve only hits matching the pharmacophore query that contain a C1OCCCC1 substructure.



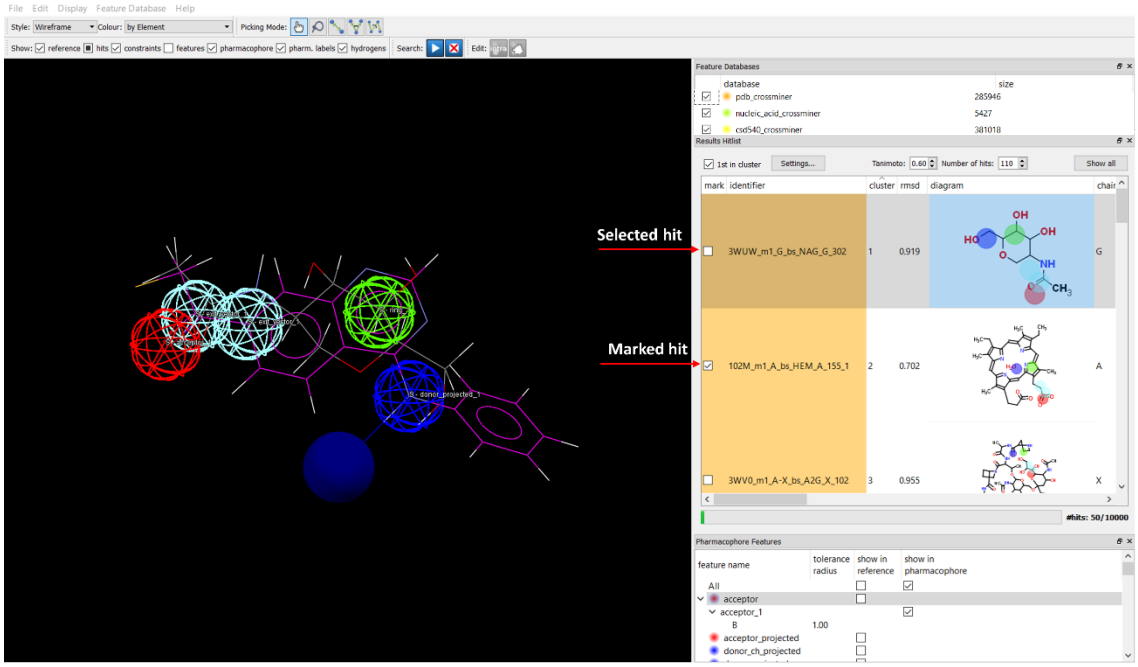
By changing the operator to *No Present,* the pharmacophore search will retrieve only hits matching the pharmacophore query that do not contain the C1OCCCC1 substructure.



Note that multiple substructure filters can be added in a pharmacophore search query.

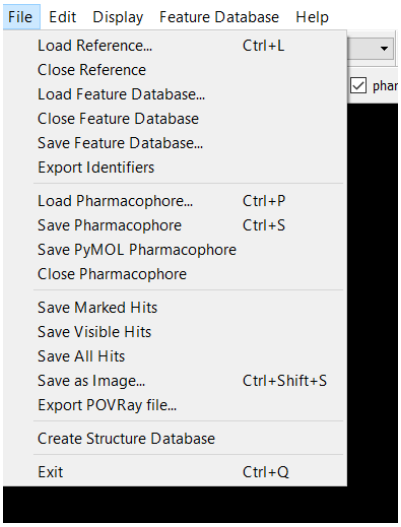
13 Exporting Hits

Interesting hits in the *Results Hitlist* browser can be marked (by ticking the *mark* tick-box for each hit of interest) and/or selected to be visualised in the 3D view (by clicking in the row for one hit of interest, multi selection is available by using CTRL+LMB and SHIFT+LMB).



Note that it is additionally possible to use the right-click menu to mark hits, see Results Hitlist and Results Hitlist Browser.

The 3D coordinates of marked and/or visible hits can be saved to disk as *mol2* or *sdf* files by selecting **Save Marked Hits** and/or **Save Visible Hits** from the CSD-CrossMiner **File** top-level menu and saving in *mol2* or *sdf* file format. Otherwise, all hits can be saved by clicking on **File** and then selecting **Save All Hits.**



As for the marked and visible hits, the 3D coordinates of all the hits matching the pharmacophore query can be saved as *mol2* or *sdf* files. The rmsd, cluster number are stored in the *mol2* or *sdf* files (in *mol2* under `@<TRIPOS>COMMENT` and in *sdf* under `> <cluster_number>`, `> <rmsd>`). Moreover, if the searched feature database has annotations (See Annotating a Feature Database), then all annotations will be stored as well in the *mol2* or *sdf* files.

In addition to *mol2* and *sdf* format, all the hits and/or the marked and/or the visible hits, can be saved in *csv* file format. This results in a table of the saved hits that include the SMILES of the small molecule matching the pharmacophore query in addition to the rmsd, cluster number and other annotations stored with the searched feature database.

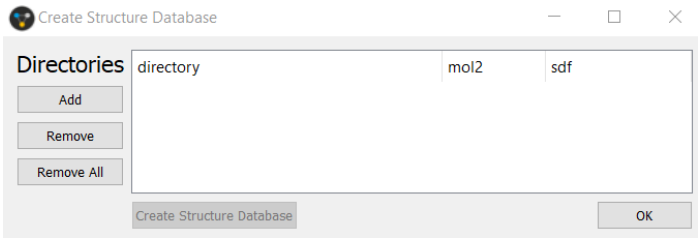
Note that hits cannot be saved during a pharmacophore search, only when the search is paused or complete.

14 Creating Databases

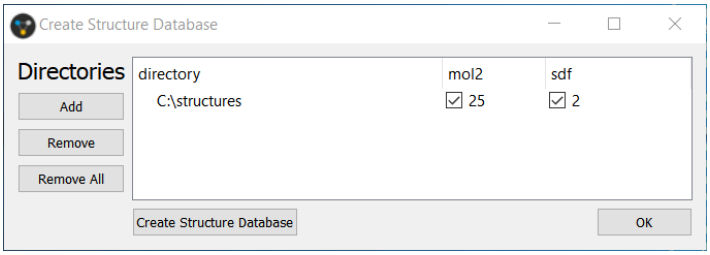
In CSD-CrossMiner there are two different types of database: structure database and feature database (See Databases in CSD-CrossMiner). Both databases can be created through the CSD-CrossMiner interface in the manner described below.

14.1 Creating a Structure Database

A new structure database can be created via the CSD-CrossMiner top-level menu **File** > **Create Structure Database**.



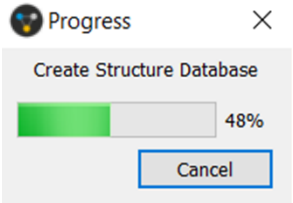
Single or multiple directories containing the files of the structures (in *mol2* and/or *sdf* format) can be included in the new structure database by clicking on **Add** in the *Create Structure Database* pop-up window. The respective number of files with both file types (*mol2 and sdf)* in the directories will be displayed in the respective *mol2* and *sdf* column with a tick-box. Any of these tick-boxes can be unticked to ignore those structures.



CSD-CrossMiner uses the residue information stored in *mol2* files to distinguish between protein and small molecule components (note that nucleic acids are treated as small molecule component). Thus, *mol2* is the preferred format if protein-ligand binding sites are to be added into a new structure database. For the associated collaborators, a python script in CSD Python API is available to convert protein-ligand and protein-ligand nucleic acids structures in PDB format into protein-ligand binding sites in *mol2* (See APPENDIX F: Example Scripts Available for Associated Collaborators).

Once the directories have been added, clicking on the **Create Structure Database** button will prompt the user to specify an output filename and will start the creation process of the structure database in CSD SQL FastBinary database file format (*.csdsqlx*).

A *Progress* pop-up window indicates the progress in creating the structure database and gives the ability to cancel the process.

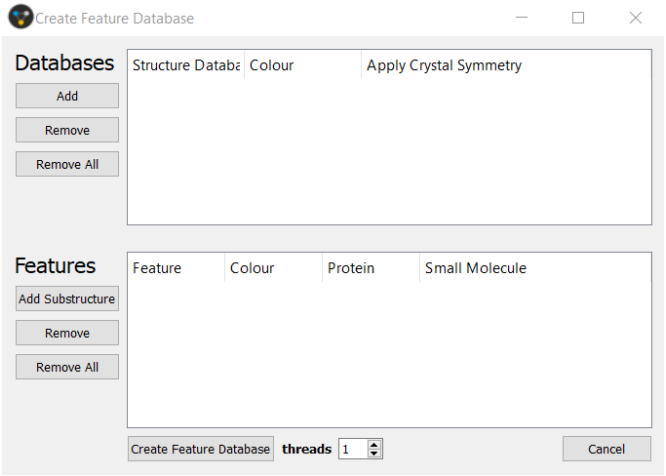


The creation of the database will be complete when the *Progress* pop-up window disappears. Clicking **OK** in the *Create Structure Database* window will then close the window.

Note that, it is important that the structure database does not contain duplicate identifiers. All entries in the structure databases need to have an identifier that is unique in that database.

14.2 Creating a Feature Database

A new feature database can be created by choosing **Feature Database** from the CSD-CrossMiner top-level menu and then **Create**. In the *Create Feature Database* dialog, structure databases (*Databases)* and feature definitions (*Features*) can be specified to create a new feature database.

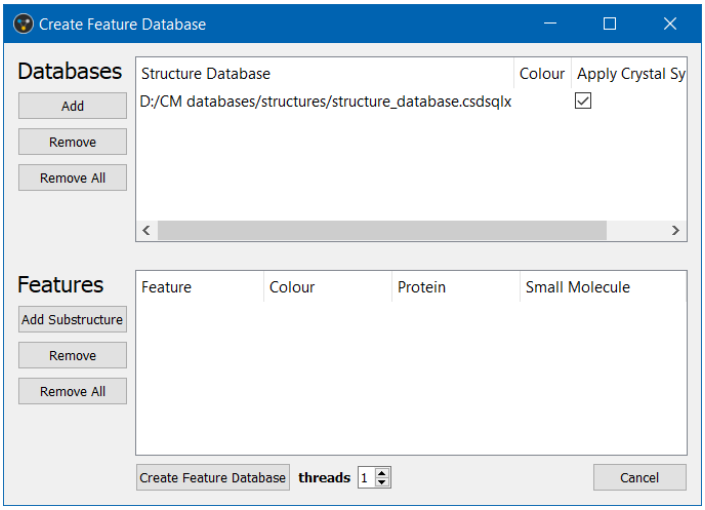


The following structure database input file formats are currently supported:

Format	File extension(s)	Comment
ASER	**.ind*, **.dbl*	Recommended for small molecule crystal structures (CSD-System databases).
CSD SQL FastBinary	**.csdsqlx*	Recommended for protein-ligand and for protein-ligand-nucleic acids binding sites (can be created from *mol2* and/or *sdf* files via CSD-CrossMiner, see Creating a Structure Database).
MOL2 SQLite	**.sqlmol2*	It is supported but it is not the recommended file format protein-ligand and protein-ligand-nucleic acids binding site. Please use **.csdsqlx* instead.
Tripos MOL2	**.mol2*	It is not recommended to use **large** multi-*mol2* files; please convert to *csdsqlx* first in these situations.
SDF	**.sdf*	It is not recommended to use **large** multi-*sdf* files; please convert to *csdsqlx* first in these situations.

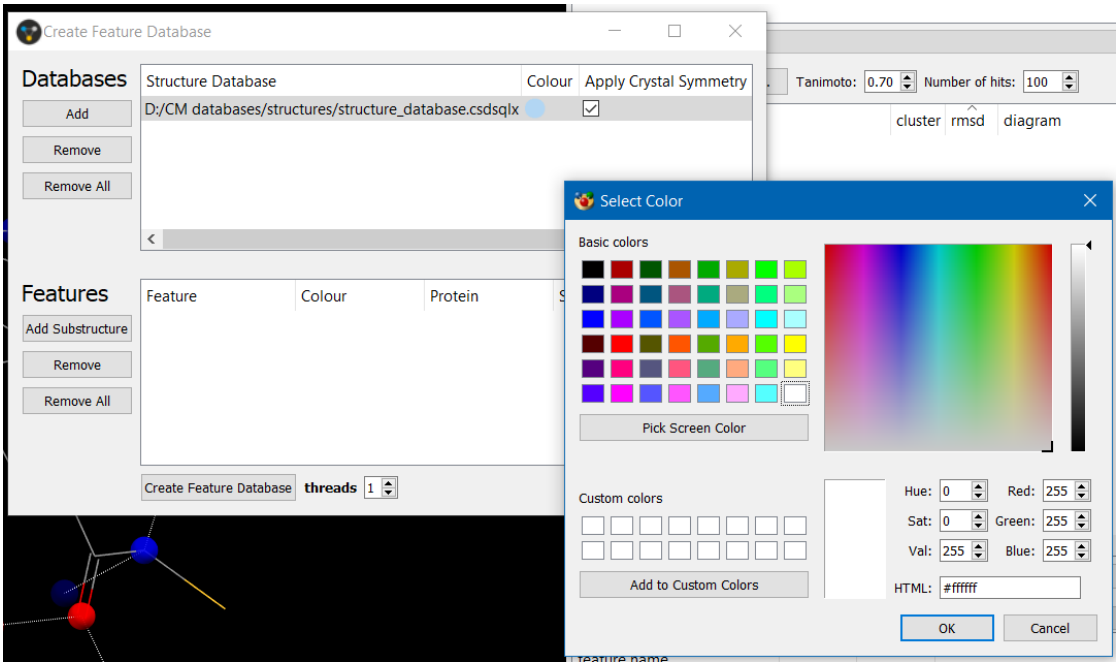
When creating a feature database, feature definitions are applied to the structure database(s); therefore, a feature database can contain different structure databases (*e.g.*, CSD and PDB structures as well as in-house small molecule and/or in-house protein-ligand binding site structures). All structure databases must thus be converted simultaneously into a single feature database in order to ensure homogeneity in the feature definitions (See APPENDIX B. Feature Definitions in CSD-CrossMiner).

Structure database(s) containing the molecular structures for which the features will be created can be added via the **Add** button in the *Databases* section of the *Create Feature Database* dialog. For each database two settings can be specified: *Colour* and *Apply Crystal Symmetry*.



The user can control these options individually for each database by highlighting each database one at a time and setting these options, which correspond to:

- *Colour:* The default colour is white; however, it can be customised by clicking on the sphere adjacent to the selected structure database, located in the *Colour* column and selecting the desired colour. When the created feature database is loaded in CSD-CrossMiner, this colour will be shown in the *Feature Databases* window and it will be used for the hit colouring in the *Results Hitlist* browser when a search is performed against this feature database. This is useful if multiple structure databases have been used to generate a single feature database.



- *Apply Crystal Symmetry:* This tick-box controls the symmetry generation behaviour; when it is ticked, CSD-CrossMiner will create symmetry-related copies based on the spacegroup information supplied for each structure. This option, therefore, should only be ticked for small molecule crystal structures (not all small molecule structures) and not for protein-ligand binding sites.

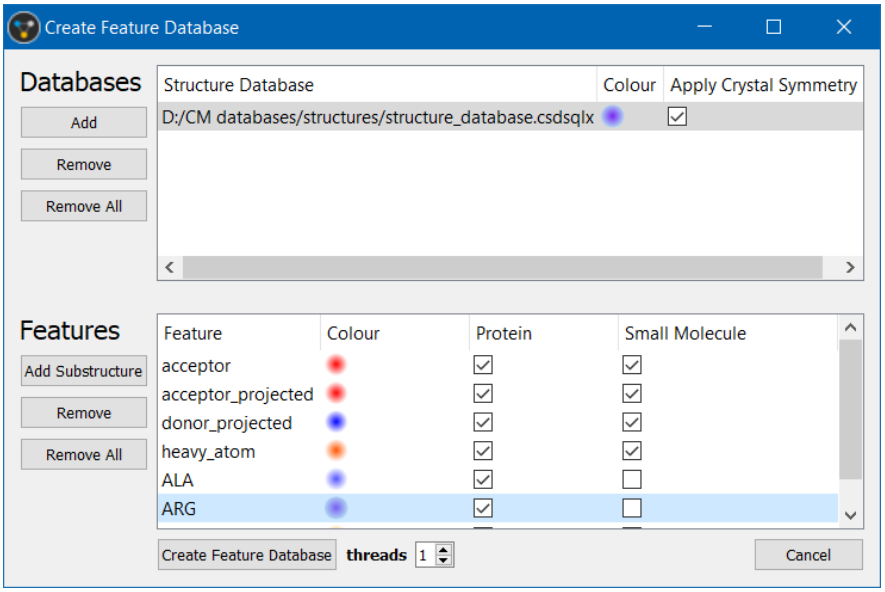
Once the structure databases have been specified, the features to be assigned to all structures contained in the database need to be loaded in the *Features* section of the *Create Feature Database* dialog.

The feature definitions used to create the feature database are derived by applying point generation rules to sets of atomic coordinates extracted from molecular structures by substructure definitions (See Editing and Creating Feature Definitions). The substructure-based features used to generate the supplied feature database are available for the user to use in the `feature_definitions` folder of the CSD-CrossMiner installation (See APPENDIX B. Feature Definitions in CSD-CrossMiner).

Note that excluded volume features are not assigned in the feature database (See Adding an Excluded Volume to a Pharmacophore Query).

These feature definition files (and any additional/updated feature created by the user) can be selected by clicking on the **Add Substructure** button in the *Features* section of the *Create Feature Database* dialog. Multiple feature definition files can be selected at the same time (using SHIFT+LMB to select a continuous list, or CTRL+LMB to select individual files). Pressing **Open** results in the selected features appearing in the *Features* list in the *Create Feature Database* window, along with their name and colour.

Since not all features may need to be created for protein and small molecule components, the feature creation should be turned *on or off* for individual components using the *Protein* and *Small Molecule* tick-boxes for each feature type that is loaded in the *Features* list.



Depending on the number of structures in the structure database(s) and the number of feature definitions loaded, the creation of a feature database can be computationally expensive; therefore, the feature database creation can be distributed across multiple CPU cores by specifying the desired number of cores in the **threads** spin-box.

The creation process can be started by clicking the **Create Feature Database** button and specifying a filename for the new feature database, which will be saved in the *.feat* file format.

The general workflow on how to create a feature database, including how to generate the input files is show in APPENDIX D. Create a Feature Database with In-House Data. Additionally, it is possible to fully automate the creation of the structure and feature by using the CSD Python API (See APPENDIX E: CrossMiner through the CSD Python API and CSD Python API Documentation).

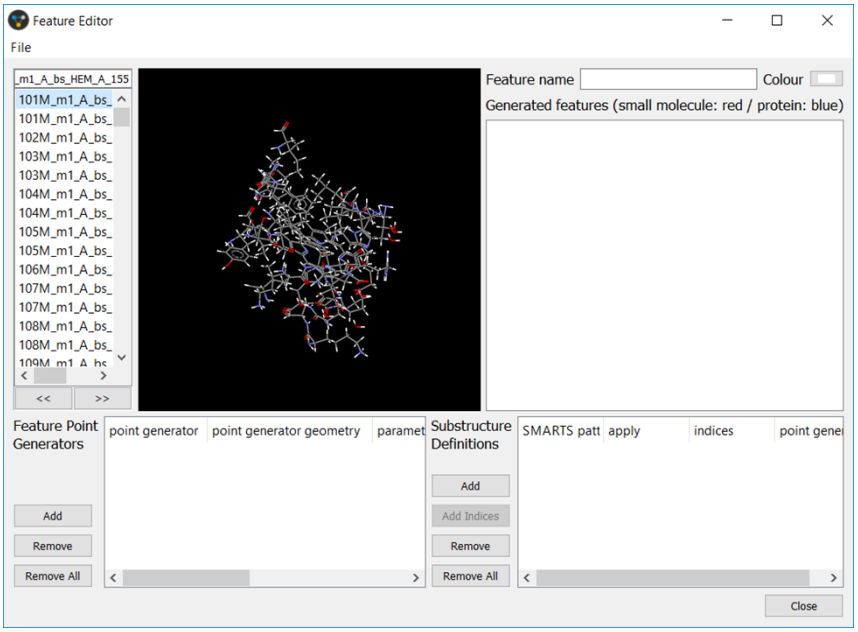
15 Editing and Creating Feature Definitions

Feature definitions can be created and edited using the *Feature Editor* accessible by clicking on **Feature Database** in the CSD-CrossMiner top-level menu and then selecting **Edit Features**.

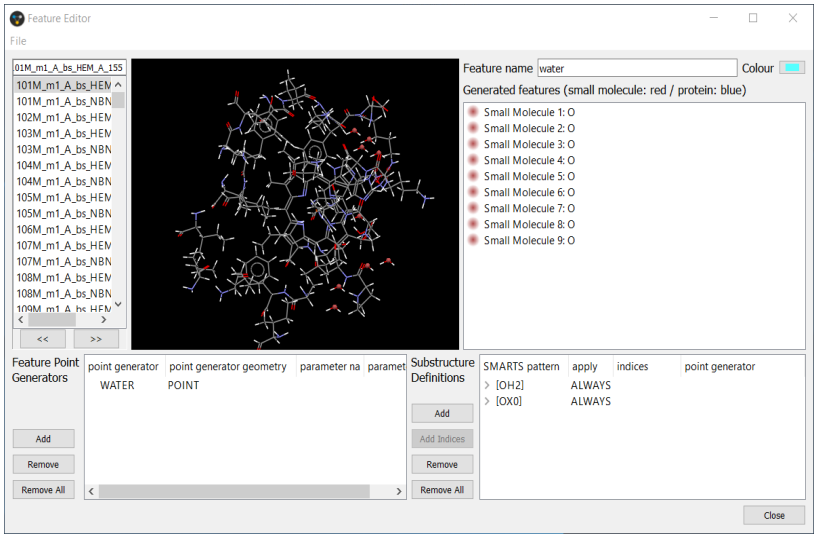
In general, the *Feature Editor* allows the generation of new features by applying point generators to sets of atomic coordinates extracted from molecular structures by substructure definitions.

It is recommended to first load a structure database by clicking on **File** in the *Feature Editor* window and then **Load Structure Database** (for example, choosing the structure database supplied in the `crossminer_data` folder in the `CSD_2019` directory).

Once a structure database has been loaded, the contained structure(s) will be displayed in the 3D display and listed in the upper left panel of the *Feature Editor* window. If the database contains multiple structures, these can be browsed by using the slider, by clicking on the **<<** and **>>** buttons at the bottom of the list, by entering an identifier in the text box, or by using the **up** and **down arrow** keys, **Page Up** and **Page Down** keys and **Home** and **End** keys.



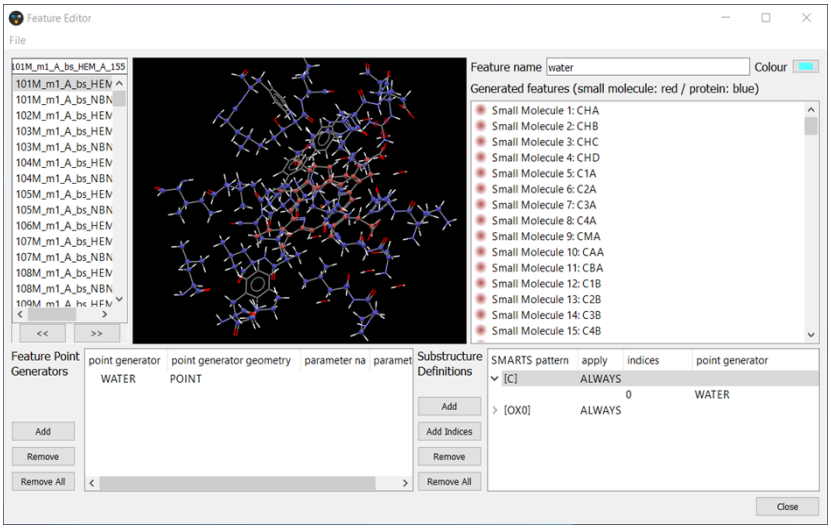
Existing feature definitions can be loaded by clicking on **File** in the *Feature Editor* window and then choosing **Load Feature Definition**. Note that only one feature definition can be loaded. In the example below, the water feature definition used to create the provided feature database is loaded. You can find the water feature together with all the other substructure-based features used to generate the supplied feature database in the `feature_definitions` folder of the CSD-CrossMiner installation folder.



The *Feature Point Generators* section includes the point generators associated with the loaded feature, while the *Substructure Definitions* section stores the substructure definitions corresponding to the loaded feature definition.

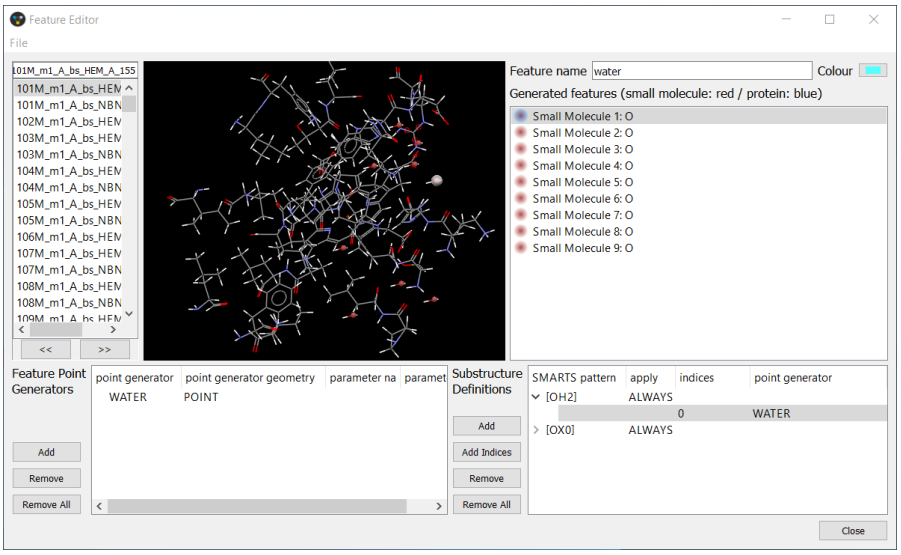
The loaded feature definition is displayed in the 3D display of the *Feature Editor* window as red (*Small Molecule*) and blue (*Protein*) feature points and listed in the right-hand panel under *Generated features* (*small molecule: red / protein: blue*) of the *Feature Editor* window. Note that this list is associated with the structure displayed in the 3D display. Clicking on any of the features in the list will highlight the respective feature in the 3D display and will select the SMARTS substructure definition that has been used to create this particular feature point (note that it can be multiple points for more complex feature definitions).

The *Feature Editor* will respond with immediate feedback on any changes to the feature definitions; *e.g.*, selecting the "[OH2]" SMARTS under *SMARTS pattern* in the *Substructure Definitions* list, changing it to "[C]" in the text box and pressing the enter key will immediately update the newly matched feature points (in this case, only aliphatic carbons) in the 3D display and the *Generated features* list.

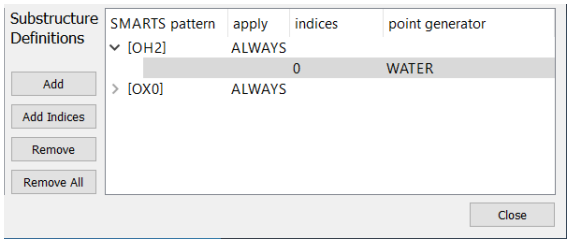


It is also possible to add a new SMARTS pattern to the list of those provided for the loaded feature or to remove one from the list by clicking on the **Add** or **Remove** button, respectively, in the *Substructure Definitions* panel.

Below the SMARTS pattern definition, there is another row in which the *indices* column is populated with a "0" index. In the example below, the 0 index selects the first atom in the SMARTS substructure it is assigned to (which is "[OH2]"); thus, it selects the oxygen atom and applies the *WATER* point generator, where the *WATER* point generator is defined in the *Feature Point Generators* panel. This point generator is simply used to create a feature point at the respective atom position.



It is also possible to add new indices in a selected SMARTS pattern by clicking on **Add Indices** in the *Substructure Definitions* panel.



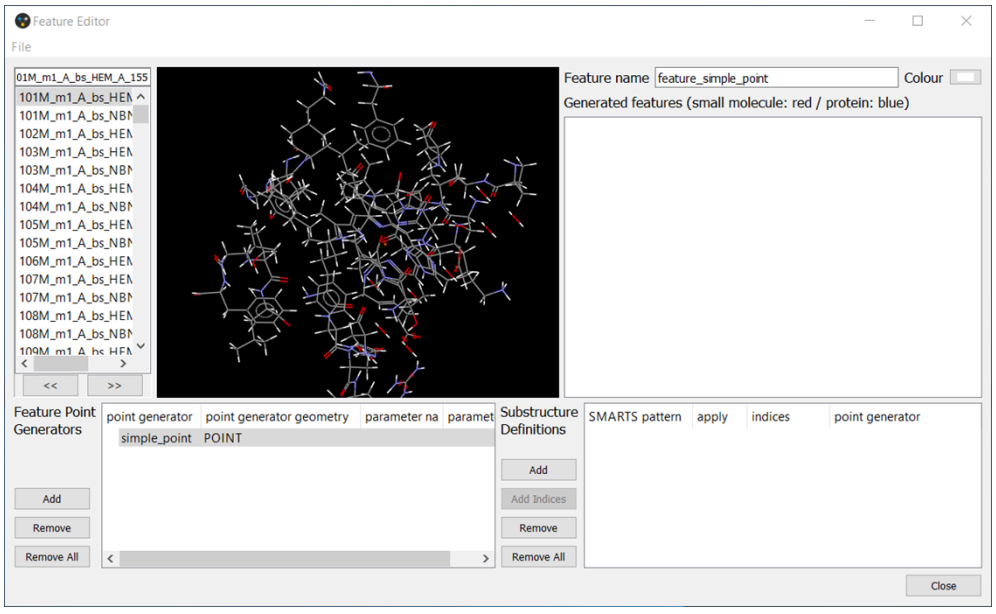
Multiple *SMARTS pattern* (along with their respective indices and point generator definitions) can be arranged in a hierarchy in which the substructures are matched in the specified order (substructure definitions can be dragged within the list box to change their priority). The substructure with the highest priority will be matched first and all selected indices will be used to mark the respective atoms of the structure as used. If a second substructure in the hierarchy with a lower priority is matched and all atoms in the structure selected by the indices have already been marked as used, then no feature

points will be generated in this case (See APPENDIX C. SMARTS Implementation and SMARTS Description.

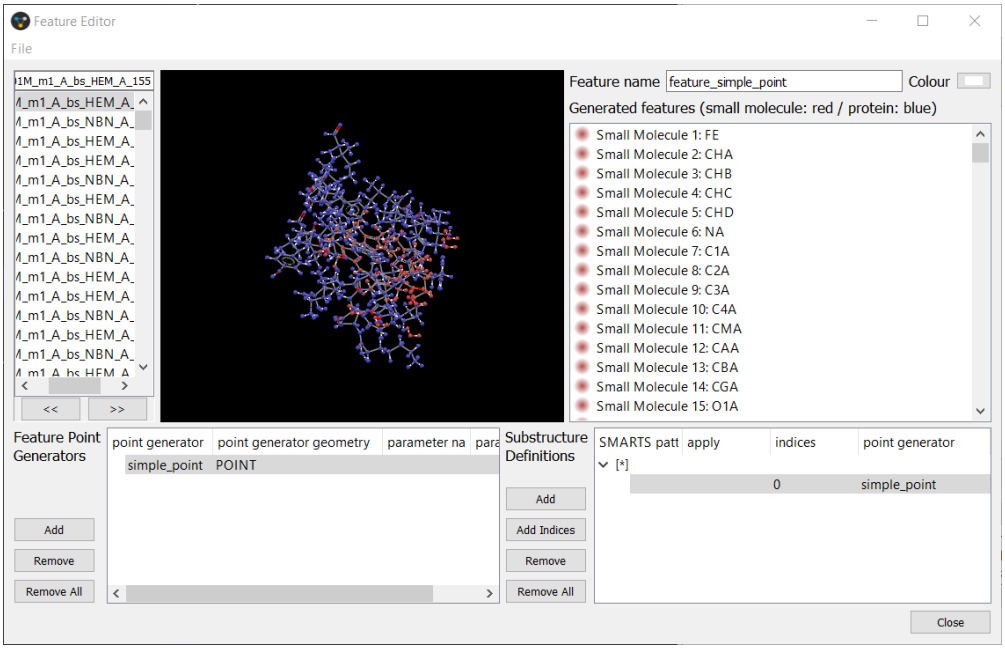
The name and the colour associated with the loaded feature is specified in upper right corner of the *Feature Editor* window. These can be edited by typing in the **Feature name** text box the new name associated with the feature and by clicking on the **Colour** box to select the desired colour.

The edited feature definition can be saved to disk in *.cpf* file format by clicking on **File** and then **Save Feature Definition** from the *Feature Editor* window and specifying a filename. This will make the new feature definition available to be used to create a new feature database (See Creating a Feature Database). It is possible to clear the *Feature Editor* window by clicking on **File** and then **Clear Feature Definition**.

A feature definition can also be created from scratch, by clicking on **Add** in the *Feature Point Generators* panel of the *Feature Editor* window.



By default, a point generator of type POINT is created, named *simple_point*, and then a SMARTS substructure definition can be created by clicking **Add** in the *Substructure Definitions* panel. By default, a SMARTS pattern of [*] that selects every atom is created.



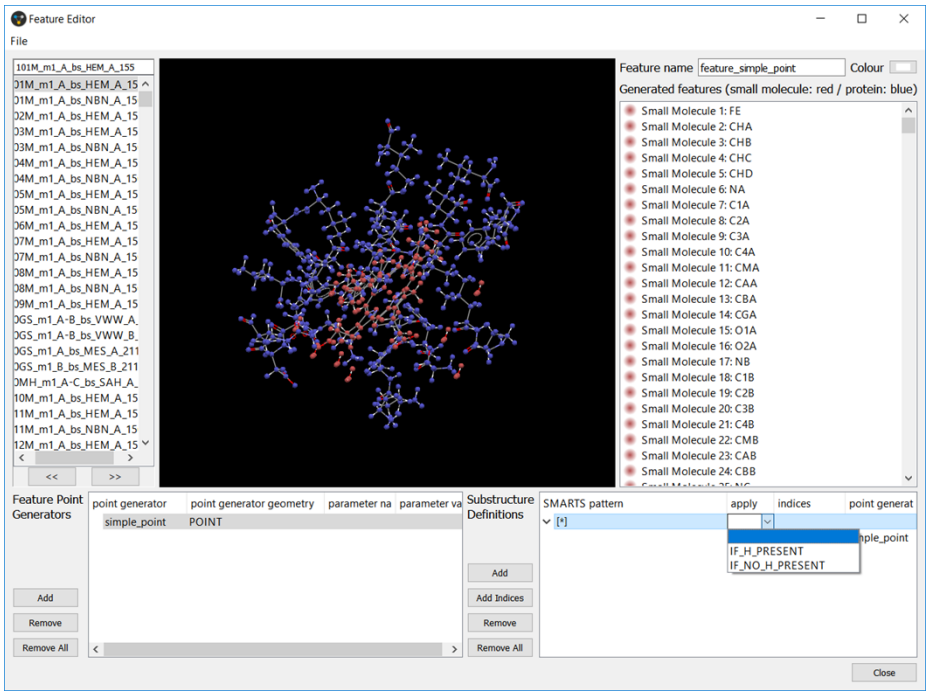
Both the point generator geometry type and SMARTS pattern can be changed; the list of all feature point generators available is provided in the following table. Details of SMARTS definition are provided in APPENDIX C. SMARTS Implementation and SMARTS Description.

Table 1: List of point generators

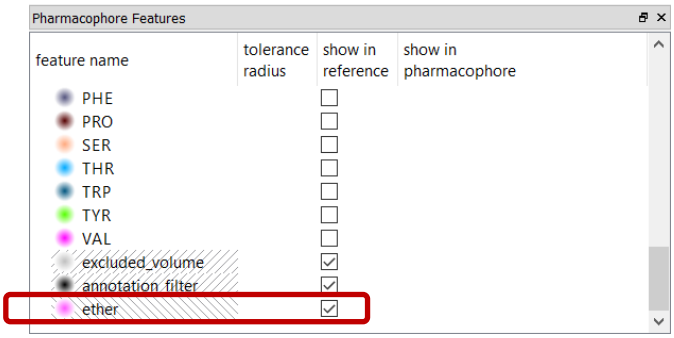
Name	Parameters	Description
DUMMY		marks the selected atoms as match although they are not used to generate any feature points. Practically, it works as 'NOT'; if an atom matches DUMMY, will not generate a feature point
POINT		creates a base feature point at the selected atom position
CENTROID		creates a base feature point at the centroid of all selected atom positions
CENTROID_PLANAR	planarity threshold: 0.1 Å	creates a base feature point at the centroid of all selected atom positions if the maximum distance of any selected atom positions to the least-squares fitted plane of the whole atom set is less than the planarity threshold.
CENTROID_NONPLANAR	planarity threshold: 0.1 Å	creates a base feature point at the centroid of all selected atom positions if the maximum distance of at least one selected atom position to the least-squares fitted plane of the whole atom set is higher than the planarity threshold.
LINEAR	virtual point distance: 2.8 Å	creates a base feature point at the selected atom position and a virtual one along the negative direction vector formed with the neighbouring atom (distance specified as a parameter).
LINEAR_NB	virtual point distance: 2.8 Å	creates a base feature point at the neighbour of the selected atom and a virtual one along the direction vector formed with the selected atom (distance specified as a parameter).
NORMAL	virtual point distance: 2.8 Å	creates a base feature point at the centroid of all selected atom positions and the virtual ones along

		the normal (N and -N) of the least-squares fitted plane of the whole atom (distance specified as a parameter).
NORMAL_PLANAR	virtual point distance: 2.8 Å planarity threshold: 0.1 Å	creates a base feature point at the centroid of all selected atom positions and the virtual ones along the normal (N and -N) of the least-squares fitted plane of the whole atom (distance specified as a parameter) if the selected atom set is classified as planar with respect to the specified planarity threshold.
TRIGONAL	virtual point distance: 2.8 Å	places a base feature point at the selected trigonal planar acceptor atom and a virtual one at the virtual lone pair position (distance specified as a parameter). Multiple pairs may be generated; *e.g.* two for a carbonyl oxygen.
TETRAHEDRAL	virtual point distance: 2.8 Å	places a base feature point at the selected tetrahedral acceptor atom and a virtual one at the virtual lone pair position (distance specified as a parameter). Multiple pairs may be generated; *e.g.* two for a hydroxyl oxygen.

In addition, it is possible to constrain when to apply the created definition in dependence of the protonation state of the structure.



The new feature definition can be saved to disk by clicking on **File** > **Save Feature Definition** in the *Feature Editor* window. In addition, it is possible to search for this feature on-the-fly by clicking on **File** and then **Add Feature to Current Feature Database** in the *Feature Editor* window. This will make the newly created features available in the *Pharmacophore Features* browser of CSD-CrossMiner with diagonal hatching to indicate that this feature has not been pre-calculated and therefore that the loaded database has not been indexed with this feature definition (e.g. ether in the example below).



Note that performing a pharmacophore search with a pharmacophore query generated with a non-indexed feature definition can affect the time performance of the pharmacophore search.

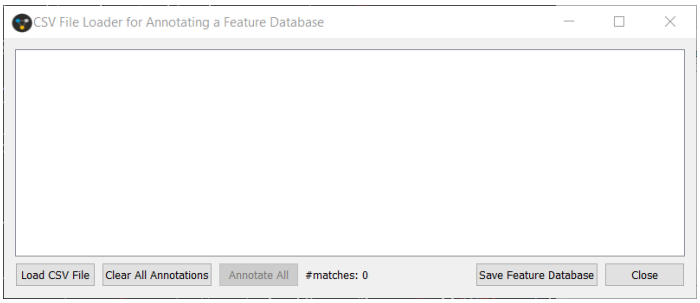
It is possible to permanently index the database with the new feature by recreating the feature database with all features including the new feature (See Creating a Feature Database).

More details on the feature definitions and SMARTS implementation are provided in the APPENDICES.

16 Annotating a Feature Database

Feature database entries in an existing feature database can be annotated with additional user-defined data. This data takes the form of key-value pairs of text and they will be displayed in the *Results Hitlist* window.

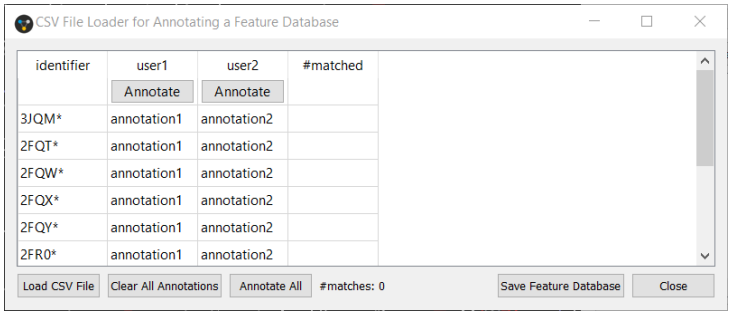
The annotations can be added to a loaded feature database by clicking on **Feature Database** from CSD-CrossMiner top-level menu and then choosing **Annotate** from the pull-down menu.



If the feature database already contains annotations (as is the case for the supplied feature database), these can be cleared by clicking **Clear All Annotations** in the *CSV File Loader for Annotating a Feature Database* pop-up window.

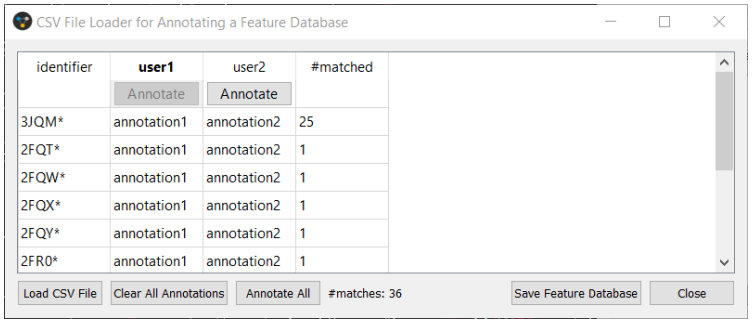
A *csv* file containing annotations for the database entries can be loaded by clicking **Load CSV File**. This *csv* file should list the annotation names in the first row, and the identifiers should be given in the first column; *e.g.*

```
identifier, user1, user2, …
AABHTZ, one, two, …
```



The first column of the *csv* file will be used for matching the identifiers stored in the feature database (See Identifier Matching Rules). Note that only the first 10 lines of the *csv* file will be displayed in the *CSV File Loader for Annotating a Feature Database*.

A single annotation can be loaded by clicking on the **Annotate** button in the respective annotation table. After the **Annotate** button is clicked, the *#matched* column will display the count of database entries from the feature database that has matched the corresponding identifier. Additionally, the number of overall matches produced will be displayed next to **"#matches**" at the bottom of the *CSV File Loader for Annotating a Feature Database* window.

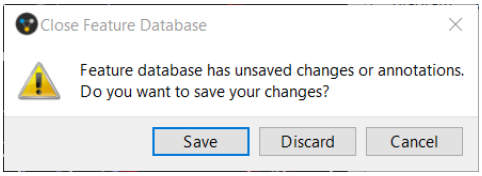


Note that the added annotation will then be immediately available in the *Results Hitlist* window.

All columns from the *csv* file may be annotated and matched by clicking the **Annotate All** button.

Please note that to permanently store the annotation in the feature database, it is necessary to save the database to disk. This can be done by clicking on the **Save Feature Database** button in the *CSV File Loader for Annotating a Feature Database* window or via **File** > **Save Feature Database** in the CSD-CrossMiner top-level menu.

If the current database is closed before annotations are saved (via **File** > **Close feature database** or **File** > **Exit** from the top-level menu), a *Close Feature Database* pop-up window will allow to **Save** or **Discard** the annotation before closing the database, or to **Cancel** the close feature database action.



16.1 Identifier Matching Rules

Identifier matching rules are similar to UNIX shell wildcards used to match the *key* and *value* during the annotation filtering process (See Filtering Matching Rules).

- **?** - Matches any one single character.
- ***** - Matches zero or more of any characters.
- **[...]** - Matches any one of the set of characters listed within the square brackets; *e.g.* "[0123456789]" will match a numeric character only.
- **** - Escape character to treat the following special character as a literal character; *e.g.* "\?" will match a "?" character only.

Any other character represents itself apart from those described above; *e.g.* "a" matches the character "a".

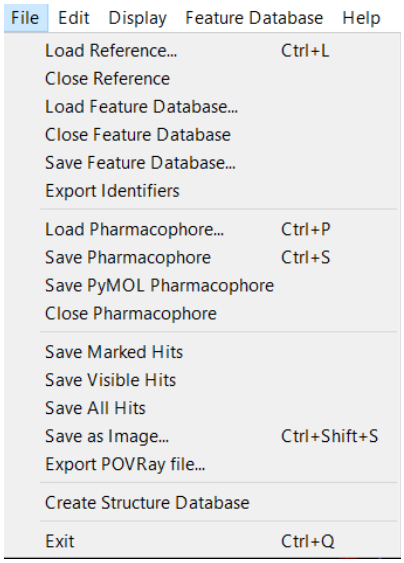
Each identifier in the *csv* file may match multiple entries in the feature database. For example, the identifier "AACANI*" would match both `AACANI10` and `AACANI11` from the CSD. Subsequent identifier matches to the same entry will overwrite any previously defined annotations for that entry.

Note that using UNIX shell wildcards for applying many annotations to a large database (*i.e.* a database with many entries) can be very slow. To speed up this process, it is recommended that simple exact string matching is used when the number of identifiers specified for matching is greater than a few tens of thousands.

17 Descriptive Menu Documentation

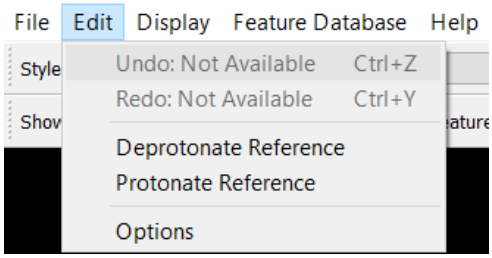
17.1 CSD-CrossMiner Top-Level Menu

17.1.1 File Menu



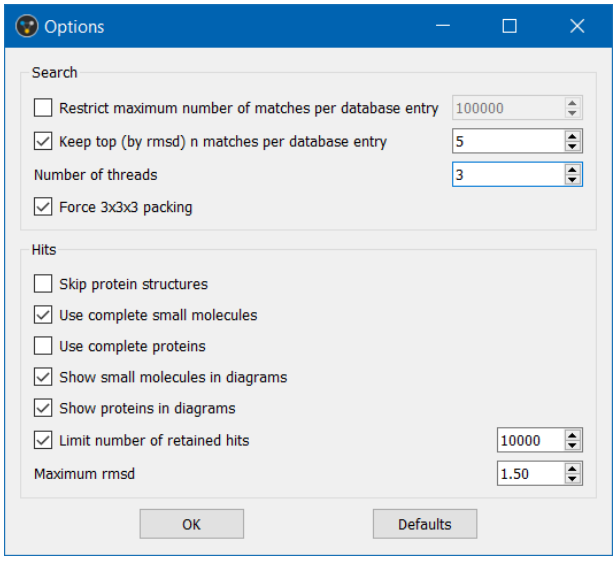
File options	Description
Load Reference	Load reference structure(s) which can be used as a template to define a pharmacophore.
Close Reference	Close the loaded reference structure(s).
Load Feature Database	Loads a feature database.
Close Feature Database	Close the loaded feature database.
Save Feature Database	Saves the current feature database.
Export Identifiers	Save the identifier of the loaded feature database. These can be used to create a *cvs* file
Load Pharmacophore	Load an existing pharmacophore.
Save Pharmacophore	Save the current pharmacophore.
Save PyMOL Pharmacophore	Save the pharmacophore in a ".py" file which can be loaded into PyMOL.
Close Pharmacophore	Clear a pharmacophore query.
Save Marked Hits	Save all hits marked in the *Results Hitlist* browser:
Save Visible Hits	Save all hits currently visible in the 3D view.
Save All Hits	Save all hits (the number of hits will correspond to the hit count shown in the progress toolbar in the *Results Hitlist*).
Save as Image	Save the 3D view as an image.
Export POVRay file	Store the scene in a POV-Ray file.
Create Structure Database	Create a structure database from *mol2*/*sdf* files (See Creating a Structure Database).
Exit	Exit the program.

17.1.2 Edit Menu



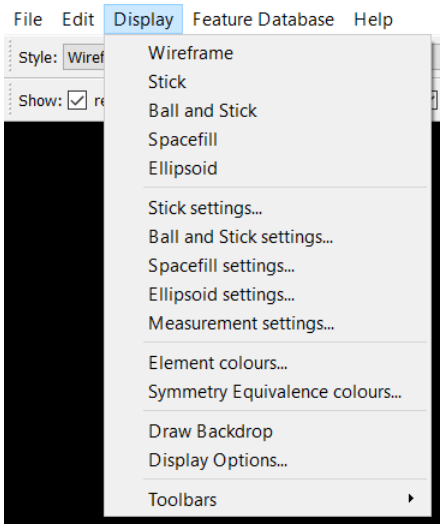
Edit options	Description
Undo:	Undo the last command.
Redo:	Redo the last command.
Deprotonate Reference	Remove all hydrogens from the current reference structure. This can have an impact on features that rely on hydrogen absence/presence.
Protonate Reference	Add a default protonation to the current reference structure.
Options	Show the *Options* dialog. The dialog can only be shown if no pharmacophore search is running (including paused searches).

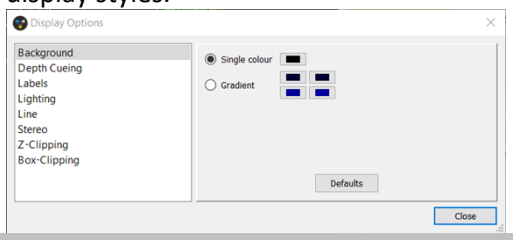
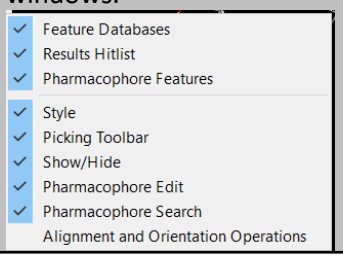
Note that for macOS users the **Options** dialogue is located in CSD-CrossMiner main task bar **>> Preferences…**



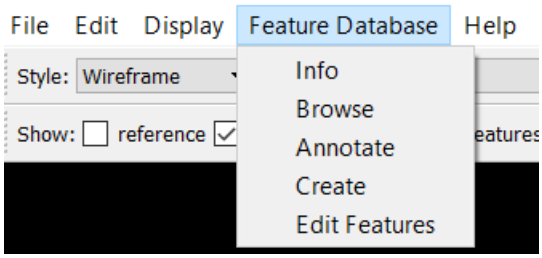
Options content	Description
Restrict maximum number of matches per database entry	Restrict the maximum number of hits per structure to the specified value. If this option is unticked no restriction is applied.
Keep top n solutions (by rmsd) n matches per database entry	Only keep the top *n* hits with respect to the Kabsch overlay *rmsd* (relative to the set of hits returned which has at most as many hits as specified in the previous option). If this option is unticked no restriction is applied.
Force 3x3x3 packing	Restrict the packing to 3x3x3 unit cells.
Skip protein structures	Don't display protein structures (only applies to features which have been set to be part of a *protein* component).
Use complete small molecules	Don't restrict a small molecule fingerprint to the bounding sphere (only applies to features which have been set to be part of a *small molecule* component).
Use complete proteins	Don't restrict the protein fingerprint to the bounding sphere.
Show small molecules in diagram	Display the small molecule 2D diagram pharmacophore overlay match.
Show proteins in diagram	Display the protein 2D diagram pharmacophore overlay match.
Limit number of retained hits	The maximum number of hits displayed in the 3D view and in the *Results Hitlist* browser.
Maximum rmsd	The maximum Kabsch overlay *rmsd* allowed for any hit.

17.1.3 Display Menu



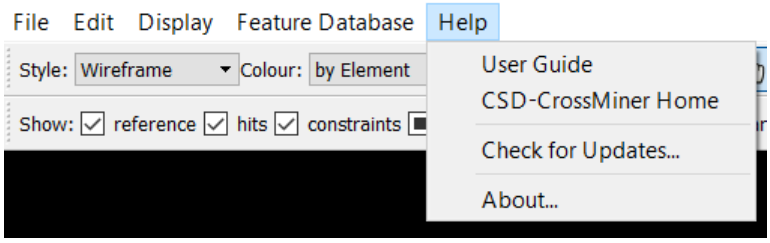
Display options	Description
Wireframe, Stick, Ball and Stick, Spacefill, Ellipsoid	Change the representation style of the molecule displayed in the 3D view.
Wireframe settings, Stick settings, Ball and Stick settings, Spacefill settings, Ellipsoid settings, Measurement settings…	Modify the current settings of the representation and measurement style.
Element colours	Change the colour of the chemical element.
Symmetry Equivalence colours	Change the colour used for *Colour by Symmetry equivalence*.
Draw Backdrop	Switch between the default black background and an alternative colour by right-clicking in the background area and hitting **Draw Backdrop**. The alternative colour will be a blue gradient.
Display Options	Create and modify sets of CSD-CrossMiner display styles. 
Toolbars	Switch on and off CSD-CrossMiner toolbars and windows. 

17.1.4 Feature Database Menu



Menu options	Description
Info	Display some information on the type of features and the number of structures stored in the database.
Browse	Open the feature database browser (See Creating a Pharmacophore Query from a Reference Structure).
Annotate	Open the annotation dialog (See Annotating a Feature Database).
Create	Open the feature database creator (See Creating a Feature Database).
Edit Features	Open the substructure feature editor (See Editing and Creating Feature Definitions).

17.1.5 Help Menu

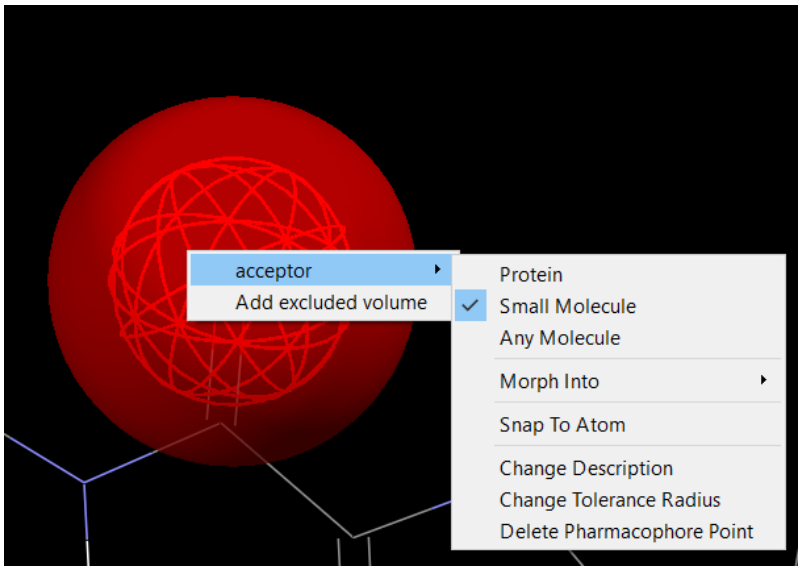


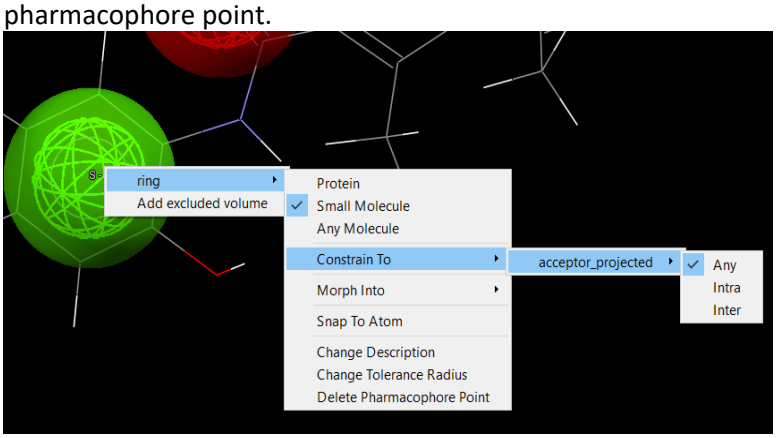
Menu options	Description
User Guide	Provide access to CSD-CrossMiner User Guide.
CSD-CrossMiner Home	Provide access to the CSD-CrossMiner web page. From there you can access to User Guide, tutorials and workshop examples.
Check for Updates…	Download available updates of CSD-CrossMiner.
About…	Display details about your version of CSD-CrossMiner including the current licensing status.

17.2 Context Right-Click Menu

17.2.1 Pharmacophore Context Right-Click Menu

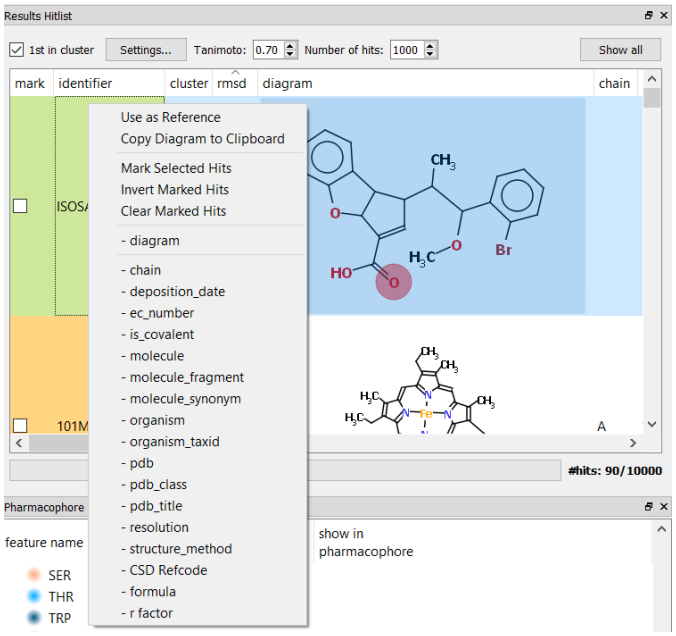
Right-click on a pharmacophore feature (See Modifying a Pharmacophore Query).



Menu options	Description
Protein	Require pharmacophore point to be part of a protein. A "**P**" label will be displayed for the pharmacophore point.
Small Molecule	Require pharmacophore point to be part of a small molecule (including nucleic acids). An "**S**" label will be displayed for the pharmacophore point.
Any Molecule	Pharmacophore point is part of a protein or a small molecule (including nucleic acids). An "**A**" label will be displayed for the pharmacophore point.
Constrain To	Allow the specification of a constraint to another pharmacophore point.  **Any:** Both features can be part of the same or different molecules. No constraint will be visualised. **Intra:** Pharmacophore points must be part of the same molecule. A dashed green line will be drawn between the

	pharmacophore points. **Inter:** Pharmacophore points must be part of different molecules. A dashed red line will be drawn between the pharmacophore points.
Morph Into	Change pharmacophore type into another one that has the same number of feature points and feature point types.
Snap To Atom	Move the pharmacophore point to the nearest atom.
Change Description	Change the description displayed next to the molecule type label. If no description is supplied by the user, the feature type will be displayed instead.
Change Tolerance Radius	Change the tolerance radius of a selected pharmacophore point.
Delete Pharmacophore Point	Delete the pharmacophore point.

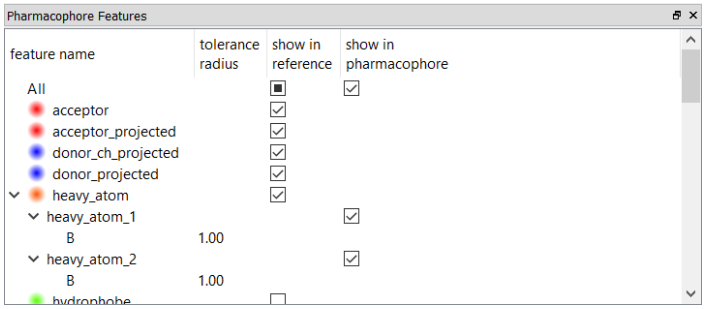
17.2.2 Results Hitlist Context Right-Click Menu



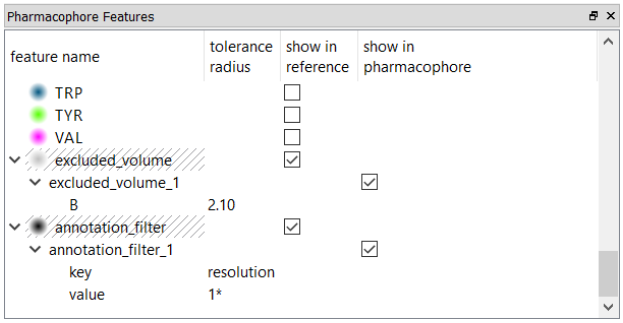
Menu options	Description
Use as reference	The selected hit will be shown in the 3D view and annotated with the features available in the current feature database (See Creating a Pharmacophore Query from a Hit).
Copy Diagram to Clipboard	Copy the diagram to the clipboard.
Mark Selected Hits	Mark all the hits selected in the *Results Hitlist* browser.
Invert Marked Hits	Invert the previously marked hits.
Clear Marked Hits	Unmark the marked hits.
"diagram"	Hide/show diagram in the *Results Hitlist* window.
"chain, deposition_date, ec_number…r_factor"	List of annotations registered in the feature database. It is possible to hide/display an annotation in the *Results Hitlist* window by clicking on it (See Results Hitlist and Results Hitlist Browser).

17.2.3 Feature and Pharmacophore Window Context Right-Click Menu

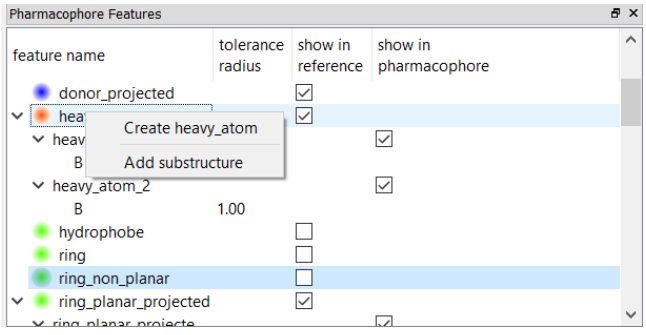
The feature types available in the feature database as well as the feature spheres used in the current pharmacophore are shown in this window. If a reference structure is shown in the 3D view, the feature points of the respective types can be shown/hidden by ticking/unticking the corresponding tick-boxes, or the *All* tick-box can be used to show/hide all feature types. The same holds for the pharmacophore points. The radii of individual pharmacophore points can be modified using the respective spin-boxes.



The *excluded volume*, *annotation_filter* and *substructure_filter* features are never indexed in the feature database. This is represented with diagonal hatching in the *Pharmacophore Features* window.



Right-clicking on a feature type will show this (or a similar) context menu:



Menu options	Description
Create …	Create a pharmacophore point of this type.
Add substructure	Add a new substructure feature type to the feature database. This will open the *Feature Editor* (See Editing and Creating Feature Definitions).

17.3 CSD-CrossMiner Toolbars

17.3.1 Style & Colour and Picking Mode Toolbars

This toolbar contains common, basic options; *e.g. Style:* for setting global display styles; *Colours:* for setting the colour mode*; Picking Mode:* for picking or lassoing atoms and for measuring distances, angles and torsions.

Style: controls the display style of molecules (wireframe, capped sticks, ball and stick, spacefill or ellipsoid).

Colour: controls the global colouring scheme. The following global colouring scheme are available:
- Colour by Element
- Colour by Symmetry equivalence
- Colour by Atomic displacement
- Colour by Symmetry operation
- Colour by Gasteiger charge
- Colour by Partial charge
- Colour by Element or Suppression

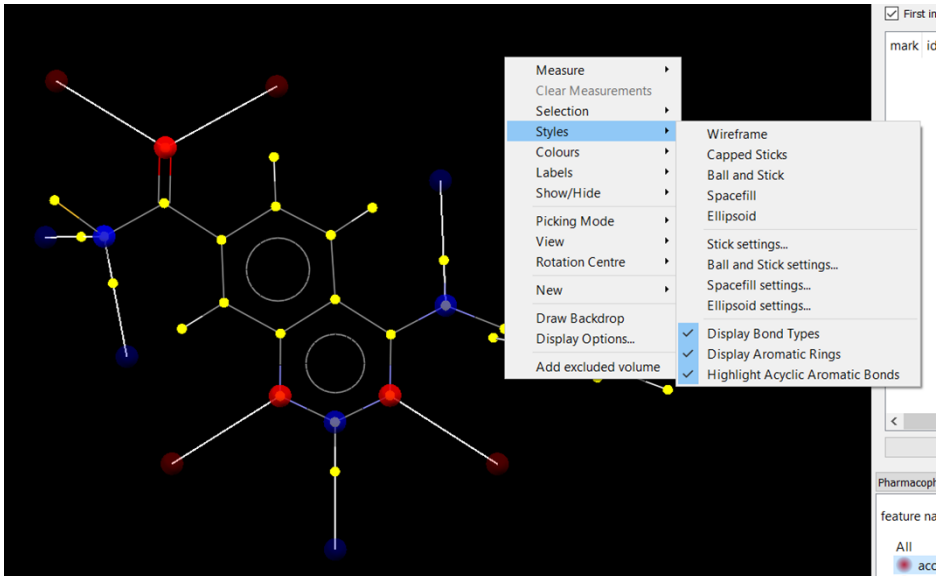
Picking Mode: controls what happens when you left-click on items in the display area:

- Use of **Pick Atoms** mode  allows the selection of atoms by clicking on them in the display area (the selection is represented as a small yellow sphere). It is possible to deselect an atom by clicking on it.

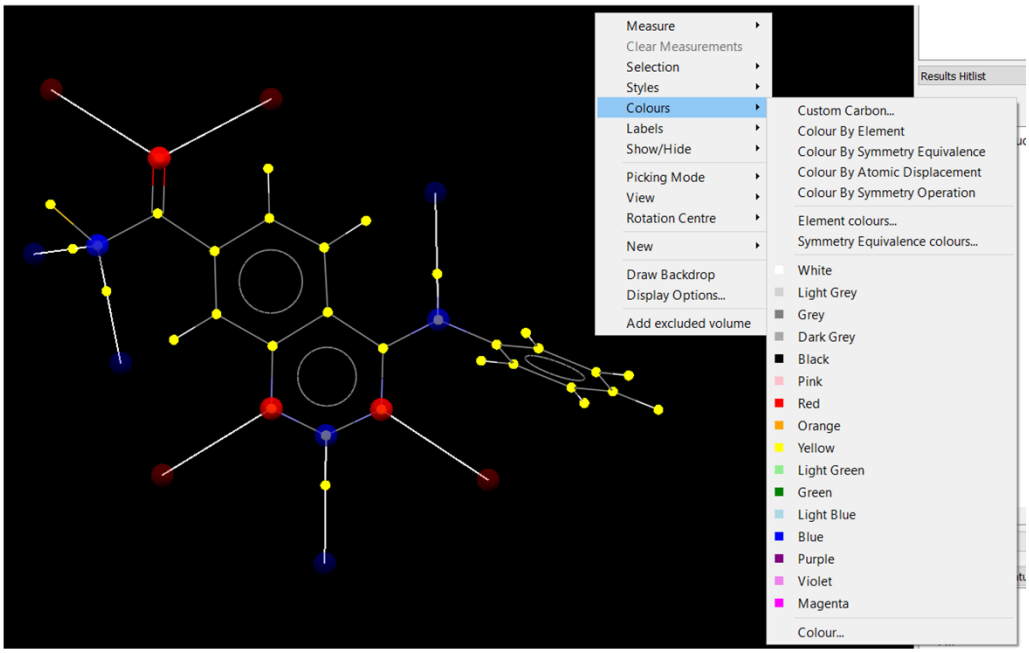
- The **Lasso Atoms** mode  allows the selection of a range of atoms by drawing a perimeter around those atoms.

Once selected, a set of atoms can be subjected to an operation; *e.g.*, changing the style and/or colour.

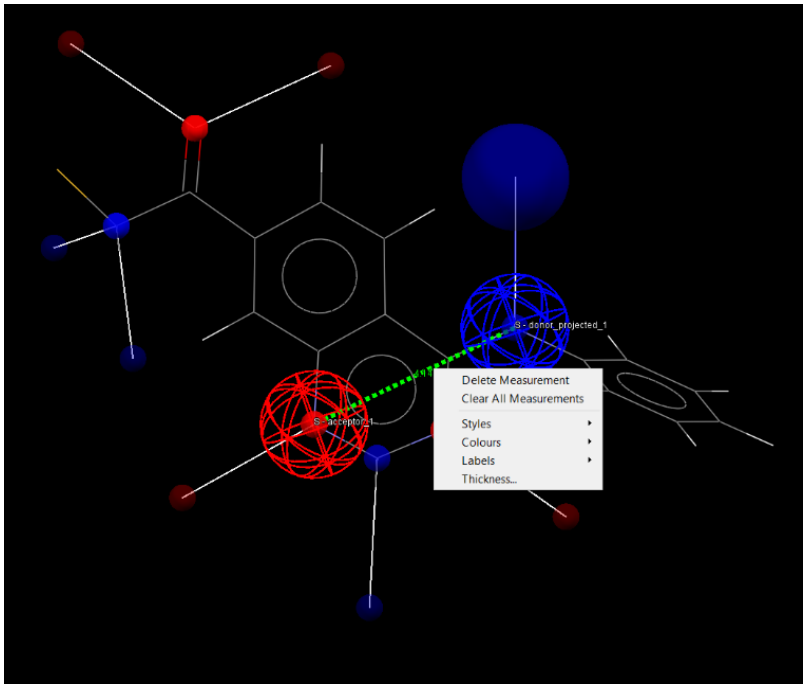
To change the style: right-click in the 3D view background, pick **Styles** from the pull-down menu and select the required style (**Wireframe**, **Capped Sticks**, **Ball and Stick**, **Spacefill, Ellipsoid**); or click **Display** in the CSD-CrossMiner top-level menu and then choose the required display style.



To change the colour: right-click in the 3D view background, pick **Colours** from the pull-down menu and select the required option (**Colour by Element**, **Colour by Symmetry Equivalence**, **Colour by Atomic Displacement**, **Colour by Symmetry Operation**).

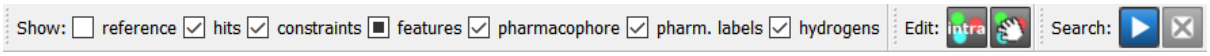


Use of **Measure Distance**, **Measure Angle** or **Measure Torsion** modes  permits the measurement of geometrical parameters by picking two, three or four objects (*e.g.* atoms, centroids), respectively. To remove all distance, angle and torsion angle measurements select **Clear Measurements** when right-clicking in the 3D view. It is also possible to remove a single bond, angle or torsion angle measurement by right-clicking on it and selecting **Delete Measurement** from the resulting pull-down menu.



The colour and style of a single measurement can be changed by right-clicking on the measurement to have access to the **Style** and **Colours** from the pull-down menu. The settings of all the distances, angles and torsions can be edited through **Measurement settings…**from the **Display** top-level menu.

17.3.2 Show, Edit and Search Toolbars



Show: Show/hide the reference structure, the hits, the pharmacophore constraints, the features, the pharmacophore, the pharmacophore labels and the hydrogen atoms.

Edit: control two pharmacophore edit options, which are

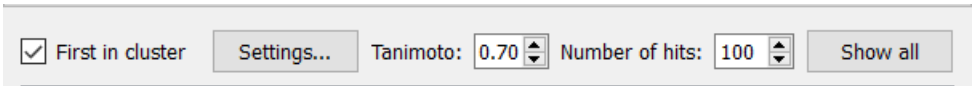
Assign intramolecular constraints between all pharmacophore points of the same type.

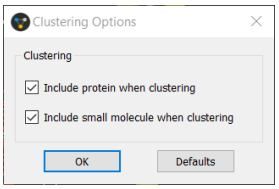

Enabls interactive editing mode. It allows an individual pharmacophore point to be translated. Any change in the position of a pharmacophore point will trigger a new pharmacophore search.

Search: Play/Pause and Stop the pharmacophore search (See

Pharmacophore Search).

17.3.3 Results Hitlist Toolbar



Toolbar options	Description
First in cluster	Enable/disable clustering of hits using the *Tanimoto* threshold specified in the spin box.
Settings…	 Allow to include/ exclude the protein and/or the small molecule fingerprint when clustering.
Number of hits	Number of displayed hits in the 3D view and in the *Results Hitlist* browser.
Show all	Visualise all hits in the 3D view.

APPENDICES

APPENDIX A. Command Line Interface

When CSD-CrossMiner is launched from a command-line interface, such as a Linux shell, there are some command-line options. The usage details for the executable will be reported if the "`-help`" argument is added to the command-line.

Option	Description
-help	Display the usage message and quit.
-feature_db	The feature database to load automatically on start-up. This will override reloading of the previously used feature database.
-pharmacophore	The pharmacophore model to load automatically on start-up.
-reference	The reference structure to load automatically on start-up.

APPENDIX B. Feature Definitions in CSD-CrossMiner

Feature definitions are used to create a feature database. They are derived from the substructures by applying point generation rules, which define how the feature points are determined from a set of atomic coordinates. The feature definitions are divided into one-point, directional and non-indexed features.

List of Feature Definitions

Feature definitions	List
One-point	Acceptor, heavy atom, hydrophobe, ring, ring non planar, adenine, cytosine, guanine, thymine, uracil, purine, pyrimidine, deoxyribose, ribose, halogen, bromine, chlorine, fluorine, metal, water, ALA, ARG, ASN, APS, CYS, GLN, GLU, GLY, HIS, ILE, LEU, LYS, MET, PHE, PRO, SER, THR, TRP, TYR, VAL.
Directional	Acceptor projected, donor projected, donor ch projected, ring planar projected, exit vector.
Non-indexed	Excluded volume, annotation_filter and substructure_filter.

The feature definitions used to create the supplied feature database are included in the `feature_definitions` folder of CSD-CrossMiner. Here, the features definitions are grouped in three different folders based on the molecule type: `small_molecule`, `protein` and `any`. Note that the `any` folder includes the feature definitions for nucleic acids.

Molecule type	Specific features
Small molecule	Exit vector, halogen, bromine, chlorine, fluorine, metal, water.
Protein	ALA, ARG, ASN, APS, CYS, GLN, GLU, GLY, HIS, ILE, LEU, LYS, MET, PHE, PRO, SER, THR, TRP, TYR, VAL.
Any molecule	Acceptor, acceptor projected, donor ch projected, donor projected, heavy atom, hydrophobe, ring, ring non-planar, ring planar projected, adenine, cytosine, guanine, thymine, uracil, purine, pyrimidine, deoxyribose, ribose.

The substructure-based features are defined by a hierarchy of SMARTS patterns. The list of SMARTS patterns defined in CSD-CrossMiner and used to generate the supplied feature database (created from CSD and PDB structures) is provided in APPENDIX C. SMARTS Implementation and SMARTS Description. These SMARTS patterns can be tailored and/or extended by the user, and new substructure-based features can be created and saved to disk (See Editing and Creating Feature Definitions).

APPENDIX C. SMARTS Implementation and SMARTS Description

The SMARTS language allows you to specify substructures using rules that are extensions of SMILES (Simplified Molecular Input Line Entry System). The current CSD-CrossMiner implementation of SMARTS is a subset of the SMARTS functionality described at http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html.

substructure_filter feature and the *Feature Editor* give immediate feedback on the validity the SMARTS string; if the SMARTS pattern defined in the *substructure_filter* is invalid, an *Invalid Search* pop-up window will invite to correct the SMARTS pattern or delete the filter. Likewise, the *Feature Editor* will give immediate feedback on the SMARTS used to define a new feature by displaying matching patterns of the loaded feature database in the 3D display. The following should be taken into consideration when using the CSD-CrossMiner implementation of SMARTS:

UNSUPPORTED FEATURES (GENERAL)

- Dot for "not necessarily connected" fragments or atoms; *e.g.* C.C
- Recursive SMARTS; *e.g.* `[$(CC);$(CCC)]`
- Reaction SMARTS; *e.g.* `CC>>CC`

UNSUPPORTED FEATURES (ATOM PROPERTIES)

- Some atom constraints (where n is an integer):
  - h<n>: implicit hydrogens
  - R<n>: ring membership

- <n>: atomic mass
- Stereochemical descriptors
- Constraints of different types combined with OR operator; *e.g.* `[#7X1,#7D2]`. However, a single feature definition can contain several SMARTS strings that are technically 'OR'
- High precedence AND in OR subexpression; *e.g.* `[C,N&H1]`. However, these SMARTS can be defined as separate `[C]` and `[N&H1]` in a single feature definition that will consider them as 'OR'.

UNSUPPORTED FEATURES (BOND PROPERTIES)

- Stereochemical descriptors for double bonds: these are treated as single bonds with unspecified stereochemistry
- High-precedence AND in OR subexpression; *e.g.* `=&@,-` (cyclic double or single and unspecified cyclicity)
- The following constructs are not supported:
  - NOT any bond; *e.g.* `!~`
  - different bond types combined with AND operator; *e.g.* `-&=` (single and double)
  - different NOT bond types combined with OR operator; *e.g.* `!-,!=` (not single or not double)

APPENDIX D. Create a Feature Database with In-House Data

Input Files

Due to the interactive quality of CSD-CrossMiner, it is recommended that protein files are truncated to the region of interest, such as the binding site(s) for protein-ligand and protein-ligand-nucleic acids complexes. It is possible to include full-length proteins and search across these with CSD-CrossMiner, but there will be an impact on the performance due to holding whole proteins with their associated features in memory.

General Workflow

1. Prepare the data
   a. 3D molecules with hydrogen atoms and appropriate atom and bond types are needed. CSD-CrossMiner feature definitions can assign HBD and HBA with no hydrogens based on generic rules about predominant protonation and tautomeric states; however, a more accurate description of the molecules in the database will also result in more accurate results.
   b. Recommended format is *mol2* for protein-ligand complexes and small molecules. For small molecules *sdf* format can also be used.
   c. If only binding sites of ligands in proteins are needed, reduce the protein files to the region surrounding the ligand only.
2. Convert the in-house data into structure database(s) using CSD-CrossMiner
   a. See Creating a Structure Database for details.

   b. Subsets of the CSD database or in-house small molecule crystal structure datasets are readily structure databases.
3. Convert these structure databases into the feature database using CSD-CrossMiner and the feature definitions
   a. See Creating a Feature Database for details.
   b. Please note that if one wants several structure databases included in a feature database, these structure databases needs to be converted simultaneously into a single feature database following instructions in Creating a Feature Database. This is to ensure homogeneity in the feature definitions.

Note that, structure and/or feature databases can be created using the CSD-CrossMiner interface (See Creating Databases) or using the CSD Python API (See CSD Python API documentation).

APPENDIX E: Pharmacophore search through the CSD Python API

CSD-CrossMiner is fully implemented in the CSD Python API allowing to automate the feature database generation, the pharmacophore query generation and the pharmacophore search workflow. Several cookbook examples are provided in the `examples` folder of CSD Python API downloadable from CCDC website. For a descriptive documentation and available cookbook examples see the CSD Python API documentation.

APPENDIX F: Example Scripts Available for Associated Collaborators

Prepare Input Files for the Structure Database

If only binding sites of ligands in proteins are needed in the feature database, associated collaborators can use *extract_binding_sites_to_mol2.py* example script to reduce the protein files to the region surrounding the ligand only. *extract_binding_sites_to_mol2.py* is available in the `utilities` folder of the `ccdc_rp` CSD Python API package, downloadable from CCDC website, see CCDC RP Utilities section of the CSD Python RP API documentation for more details.

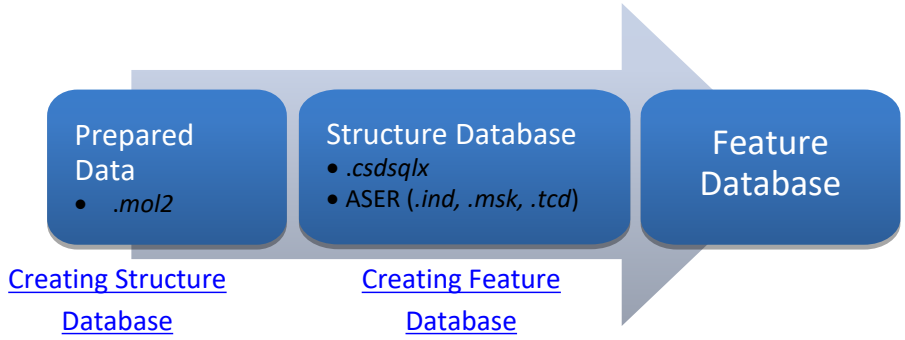
This script can be used to prepare the input data for the creation of the feature database (See APPENDIX D. Create a Feature Database with In-House Data).

The *extract_binding_sites_to_mol2.py* script uses the CSD Python API and Biopython[1] packages to extract the protein-ligand binding sites from protein-ligand complexes in PDB format and convert them to *mol2* format.
For each ligand with more than 5 atoms and fewer than 100 atoms, included in PDB protein-ligand and protein-ligand-nucleic acids complexes, the *extract_binding_sites_to_mol2.py* script will:

   a. Define the protein-ligand binding site as all residues within 6Å of the ligand.

[1] Biopython is a set of freely available tools for biological computation written in Python. See http://biopython.org/

b. Generate a *csv* file containing information about each generated protein-ligand binding site. This file can be used to annotate the feature database.
c. Add hydrogens using the `add_hydrogens` function in Python API (See CSD Python API Documentation).



d. Write the result out to a *mol2* file.

These *mol2* files can then be used to create the structure and feature database as described in APPENDIX D. Create a Feature Database with In-House Data.