GOLD User Guide

1 GOLD User Guide

A Component of the CSD-Discovery Suite

2025.3 CSD Release

To access our new format tutorials please visit the GOLD https://www.ccdc.cam.ac.uk/solutions/software/gold/ web page

1.1 Conditions of Use

By using this software, you also agree to our standard licence agreement in the following link: https://www.ccdc.cam.ac.uk/licence-agreement/

The GOLD program, Hermes visualiser, associated documentation and software, 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 Portfolio 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 Program is 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.

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

2 Introduction and Overview

GOLD (**G**enetic **O**ptimisation for **L**igand **D**ocking) is a genetic algorithm for docking flexible ligands into protein binding sites. GOLD has been extensively tested and has shown excellent performance for pose prediction and good results for virtual screening.

GOLD is supplied as part of CSD-Discovery, which also includes Hermes. Hermes provides the graphical user interface for GOLD. It is designed to assist with the preparation of input information for docking with GOLD, visualisation of docking results and calculation of descriptors. Further details are provided in the Hermes documentation.

GOLD can be run in batch mode (see Running GOLD from the Command Line and GOLD Configuration File documentation) with:

- One or more prepared protein input structures.

- Prepared ligand input structures.
- A GOLD .conf file including all the settings for the docking, e.g. definition of the binding site, constraints.

GOLD will only produce reliable results if ligands have correct protonation states set. Hermes will automatically derive SYBYL atom types from the input provided (see Atom and Bond Types). Protein and ligand structures can be prepared using standard molecular modelling packages, as long as SYBYL atom types are set correctly. Ligand input structures very much benefit from minimisation using CSD Conformer Generation prior to docking with GOLD.

GOLD offers a wide range of customisation choices to ensure that it is tailored to your project and can make the most of your knowledge. This document will guide you through the wealth of options (e.g. scoring functions, constraints, flexiblity, speed) available.

Tutorials 1-8 (see Appendix A: Tutorials) of this user guide are designed for the novice GOLD user and do not require any previous knowledge. They are a very good starting point to get familiar with GOLD and provide plenty of references to more detailed information on the options available in GOLD.

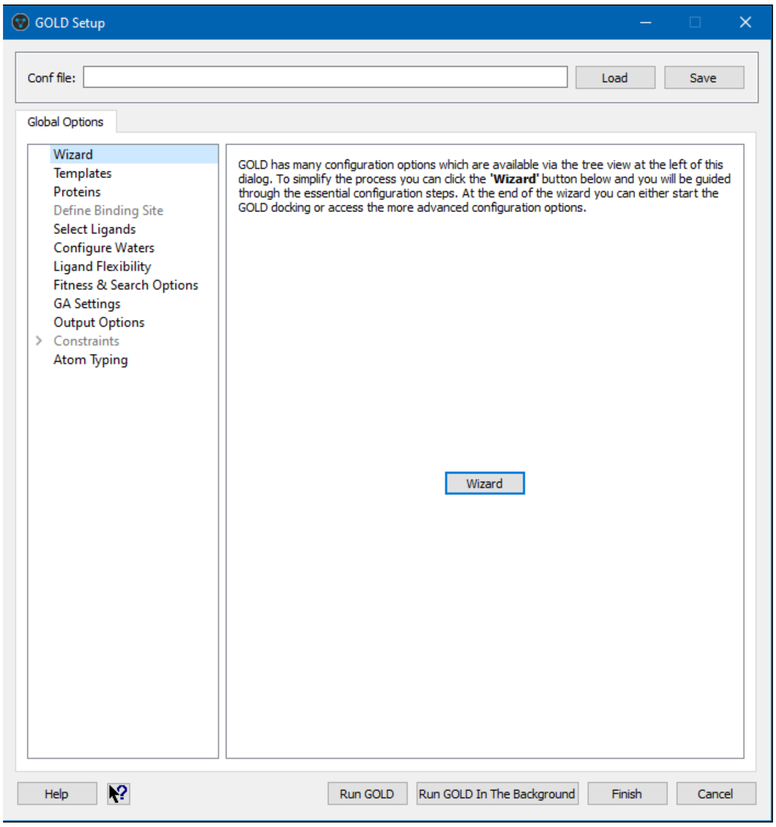
Please do not hesitate to contact support@ccdc.cam.ac.uk for more information and help.

3 Getting Started

3.1 Overview of the GOLD Interface

Select **GOLD** from the top-level menu in the Hermes visualiser, then **Setup and Run a Docking** from the resulting menu.

You will be asked whether you wish to create a new GOLD configuration file or to load an existing one. The configuration file is a text file which specifies the GOLD calculation that is to be run, including details of the ligand, the protein binding site, the fitness-function to be used, and the genetic algorithm parameters etc. Selecting an existing configuration file (e.g. from one of the tutorials) will result in the defined configuration options being read into the **GOLD Setup** window. The corresponding structure input files will also be opened within Hermes. Selecting **New** will open an empty **GOLD Setup** window, in which you will be required to specify all the configuration options required to define the docking job (see Saving and Re-using Program Settings in Configuration Files).



GOLD has many configuration options which are available via the tree view at the left of this dialog. To simplify the process you can click the **'Wizard'** button below and you will be guided through the essential configuration steps. At the end of the wizard you can either start the GOLD docking or access the more advanced configuration options.

A list of **Global Options** is given on the left of the **GOLD Setup** window. Note that there are a number of setup options that are specific to the protein file, thus some options will not be visible until a protein file is read either manually or via a `gold.conf`. Click on a configuration option in the list in order to specify the corresponding settings on the right of the **GOLD Setup** window.

The Hermes visualiser is an integral part of the GOLD interface. It is used alongside the **GOLD Setup** window to prepare input files and for interactive docking setup, e.g. for defining the binding site and the setting of constraints. For further information on using the Hermes visualiser, refer to the Hermes user guide.

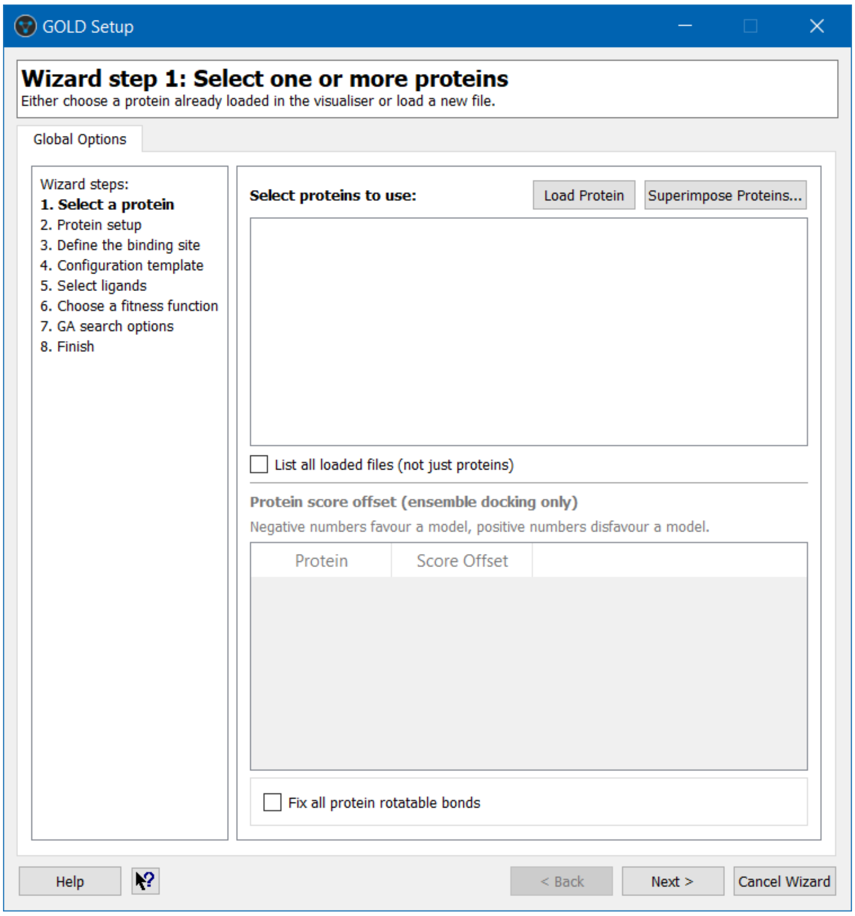
To simplify the process of setting up a docking, a wizard is available which will guide you through the essential configuration steps. The wizard can be opened at any stage from the **GOLD Setup** window by clicking on **Global Options** on the left of the window, then clicking on the **Wizard** button (see Using the GOLD Docking Wizard).

A number of configuration file templates are also available which contain recommended settings for particular docking protocols (see Using Configuration File Templates).

3.2 Using the GOLD Docking Wizard

GOLD has many configuration options. To simplify the process of setting up a docking a wizard is available which will guide you through the essential configuration steps.

Select **GOLD** from the top-level menu in the Hermes visualiser, then **Wizard** from the resulting menu. Alternatively, the wizard can be opened at any stage from the **GOLD Setup** window by clicking on **Global Options** on the left of the window, then clicking on the **Wizard** button.



GOLD Setup

Wizard step 1: Select one or more proteins

Either choose a protein already loaded in the visualiser or load a new file.

Global Options

Wizard steps:

1. **Select a protein**
2. Protein setup
3. Define the binding site
4. Configuration template
5. Select ligands
6. Choose a fitness function
7. GA search options
8. Finish

Select proteins to use: Load Protein | Superimpose Proteins...

□ List all loaded files (not just proteins)

Protein score offset (ensemble docking only)

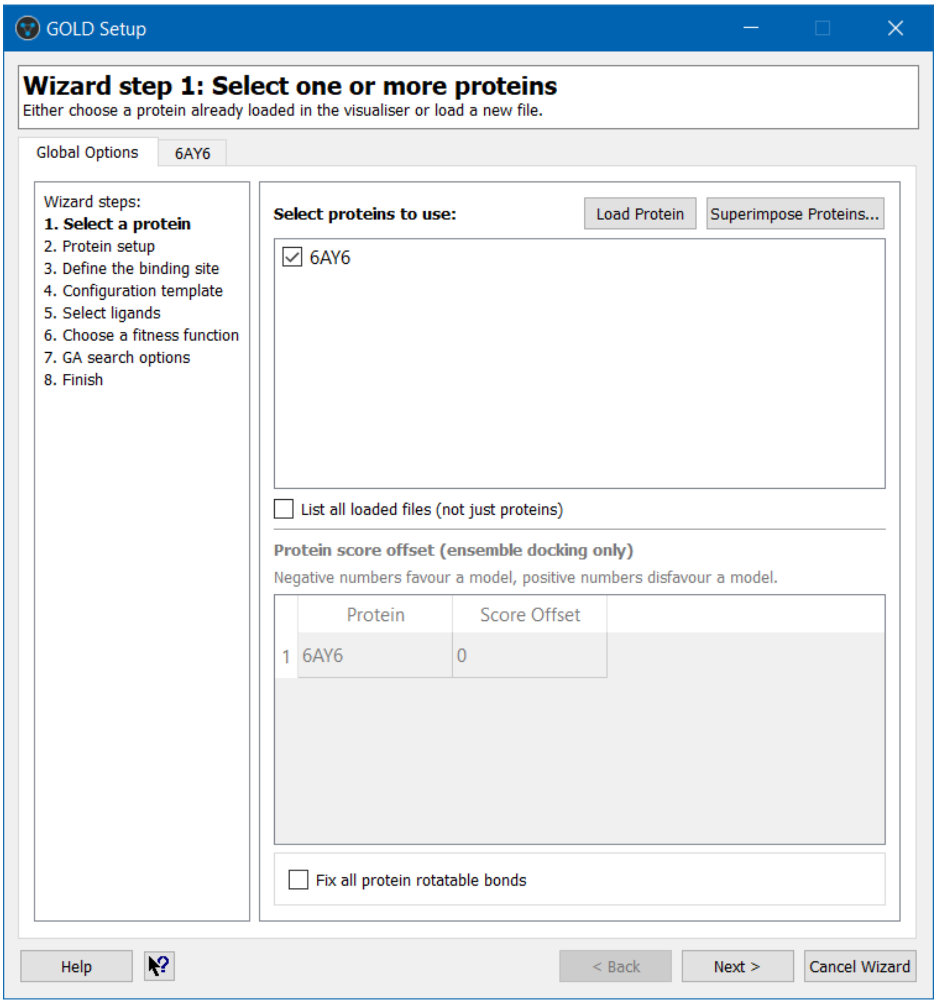
Negative numbers favour a model, positive numbers disfavour a model.

Protein	Score Offset

□ Fix all protein rotatable bonds

The appearance of the wizard will vary depending on whether a protein file has been read in or not. General docking settings are available from within the **Global Options** tab while protein-specific options only become available after a protein has been loaded into Hermes (either via **File**, **Open** or via a `gold.conf`). These protein-specific options can be found under an additional tab found next to the **Global Options** tab. The text on this tab is taken from the `_entry.id` field in an mmCIF file, or the `HEADER` record in a pdb file, or the `@<TRIPOS>MOLECULE` record in a mol2 file.

The number of tabs will vary depending on how many proteins are loaded.



Wizard step 1: Select one or more proteins

Either choose a protein already loaded in the visualiser or load a new file.

Wizard steps:
1. Select a protein
2. Protein setup
3. Define the binding site
4. Configuration template
5. Select ligands
6. Choose a fitness function
7. GA search options
8. Finish

Select proteins to use:

☑ 6AY6

□ List all loaded files (not just proteins)

Protein score offset (ensemble docking only)

Negative numbers favour a model, positive numbers disfavour a model.

	Protein	Score Offset
1	6AY6	0

□ Fix all protein rotatable bonds

The wizard will guide you through the steps required to configure a docking. At each step follow the instruction provided. Once a step has been completed click on the **Next** button to proceed to the next configuration step, or **Back** to return to the previous step. To cancel the wizard, click on the **Cancel Wizard** button.

Tutorial 1 describes, in detail, how to use the GOLD wizard (see Tutorial 1: A Step-By-Step Guide to Using GOLD).

4 Setting Up the Protein(s)

4.1 Essential Steps

Protein setup is the same whether an individual protein or an ensemble of proteins is being used:

- You can either input the whole protein structure to GOLD, or just those residues that are in the active site region. The latter leads to somewhat shorter run times, since both protein initialisation and cavity detection will be quicker.

- If you input only the region of interest around the binding site, you must ensure that all the residues you include are complete. You should also include all residues within a 5 Å radius from the solvent-accessible surface of the cavity.

- Add all hydrogen atoms, including those necessary to define the correct ionisation and tautomeric states of residues such as Asp, Glu and His (see Protonation and Tautomeric States).

- Ensure that all bond types are correct. They can be explicitely written into mmCIF and mol2. If they are, and hydrogen atoms have been placed on the correct atoms, GOLD will deduce atom types automatically (see Automatically Setting Atom and Bond Types). This also applies to pdb input files but only for known residues (i.e. there is no HET group library). If an mmCIF file doesn't contain bond information, the bonds will be generated based on a protonation pattern.

- GOLD connects atoms within residues on the basis of proximity. Double bonds are assigned as appropriate for the naturally occurring protein residues.

- Residues should be in sequence order, and correctly named.

- All atoms should be properly labelled (CA, CB, etc.).

- Any unusual bonds (disulfide bridges, etc.) should have CONECT records.

- If a metal ion is present, ensure that all bonds between the ion and coordinating protein or water atoms are deleted (GOLD will re-find them automatically). Metals should be within bonding distance of at least two protein and/or water atoms in the active site so that GOLD can infer likely coordination geometries (see Metal Ions).

- Save the protein in, e.g., mmCIF or mol2 format.

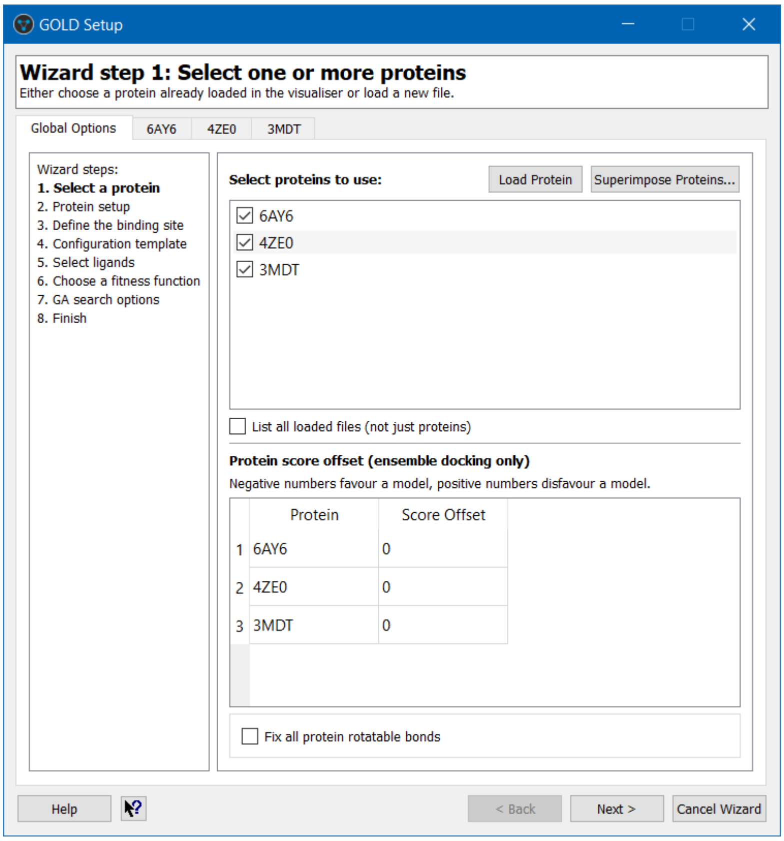
- GOLD assigns atom types from the information about element types and bond orders in the input structure file, so it is important that these are correct. However, if for any reason, GOLD is unable to deduce an atom type, then the atom in question will be replaced with a dummy atom type Du. If this is the case a warning message will be given in the `gold_protein.log` file.

- The presence of dummy atoms should not significantly affect the docking prediction since dummy atoms are neither considered as donors nor acceptors.

Note that the steps above are essential whether docking a ligand into a single protein or carrying out an ensemble docking (see Ensemble Docking).

4.2 Specifying the Protein File or Files

Click on **Proteins** from the list of **Global Options** given on the left of the **GOLD Setup** window.



Wizard step 1: Select one or more proteins

Either choose a protein already loaded in the visualiser or load a new file.

Wizard steps:
1. Select a protein
2. Protein setup
3. Define the binding site
4. Configuration template
5. Select ligands
6. Choose a fitness function
7. GA search options
8. Finish

Select proteins to use:

☑ 6AY6
☑ 4ZE0
☑ 3MDT

□ List all loaded files (not just proteins)

Protein score offset (ensemble docking only)

Negative numbers favour a model, positive numbers disfavour a model.

	Protein	Score Offset
1	6AY6	0
2	4ZE0	0
3	3MDT	0

□ Fix all protein rotatable bonds

A list of those proteins currently loaded in the Hermes visualiser is listed under **Select proteins to use**. Select the protein you wish to use from this list.

Alternatively, to specify a different protein file, click on the **Load Protein** button and use the file selection window to choose the protein data file. Once selected, the chosen protein will be loaded and displayed within the Hermes visualiser.

Use the protein tickboxes to determine which proteins are to be docked into, e.g. in the example above all three proteins (6AY6, 4EZ0 and 3MDT) will be docked into.

Acceptable protein file formats are mmCIF, pdb and mol2.

4.3 Protonation and Tautomeric States

4.3.1 Adding Hydrogen Atoms to the Protein Using Program Defaults

GOLD uses an all-atom model, so the protein must have all hydrogen atoms added.

To add missing hydrogen atoms to a particular protein, select the tab appropriate to the protein you wish to protonate and then select **Protonation & Tautomers** from the list of available options given on the left of the window.

Click on the **Add Hydrogens** button to protonate the protein.

The number of hydrogens added to each atom will be sufficient to satisfy the atom's unfilled valencies.

The hydrogen atom positions will be normalised, i.e. the X-H distance will be made equal to the average neutron diffraction value (hydrogen atoms are accurately located by neutron diffraction), e.g. C-H bond lengths will be set to 1.083 Å, N-H to 1.009 Å, and O-H to 0.983 Å. It is possible to customise the values for C-H, N-H and O-H H-normalisation within the Hermes visualiser. It is also possible to specify values to normalise the position of H atoms bonded to other elements. For further information, refer to the Hermes user guide.

The geometry of added hydrogen atoms will be chemically meaningful. However, the precise geometrical positions of Ser, Thr and Tyr hydroxyl hydrogen atoms or Lys NH_3 hydrogen atoms do not matter as their orientation will be optimised during the GOLD run (see Rotatable O-H and NH3 Groups).

GOLD deduces the hydrogen-bonding abilities of protein residues from the presence or absence of hydrogen atoms. For example, you can control the protonation and tautomeric state of Asp, Glu and His residues by adding or removing appropriate hydrogen atoms. If incorrect ionisation or tautomeric states are inferred by the program, it is unlikely that correct protein-ligand binding modes will be predicted. It is therefore important that you check protonation states of such residues before proceeding with the docking. Additional structure editing functionality is available within the Hermes visualiser.

4.3.2 Applying Protonation Rules

It is possible to protonate using SMARTS-based protonation rules contained in the `protonation_rules.txt` file.

A sample file is provided in:

4.3.2.1 Windows

```
<Installation folder>\ccdc-software\hermes
```

4.3.2.2 Linux

```
<Installation folder>/ccdc-software/hermes/Hermes
```

4.3.2.3 macOS

```
<Installation folder>/ccdc-software/hermes/Hermes.app/Contents/
        MacOS
```

The file contains SMARTS-based rules for protonation of the format `<query SMARTS> <rule SMARTS>`, e.g.

```
# Carboxylate (set the C-O bond type to aromatic)
*-C(\~[OD])\~[OD] *-C(:[OD]):[OD]
```

Load this file using the file selection window obtained by clicking on the **...** button adjacent to the **Protonation Rules** text box.

The file can be modified and supplemented to suit user preferences.

4.3.3 Flipping Asn and Gln Residues

Terminal CO-NH_2 groups in Asn and Gln residues can be flipped (i.e. rotated 180 degrees). This can be useful when dealing with poorly resolved protein structures in which you suspect the oxygen and nitrogen atoms may have been incorrectly determined, i.e. transposed.

As residues are protein-specific, click on the appropriate protein tab adjacent to the **Global Options** tab and select **Protonation & Tautomers** from the list of options provided.

A list of the Asn and Gln residues within the defined binding site will be displayed. Select the residue you wish to flip from this list (the selected residue will be highlighted in the Hermes visualiser) and click on the **Flip** button in order to rotate the CO-NH_2 group 180 degrees.

4.3.4 Specifying Histidine Tautomers

GOLD will not vary tautomeric states during docking.

To specify the tautomeric state of particular histidine residues within the binding site select the appropriate protein tab then select **Protonation & Tautomers** from the list of options given on the left of the **GOLD Setup** window.

A list of the His residues within the defined binding site will be displayed. Select the His residue you wish to edit from this list, the selected residue will be highlighted in the Hermes visualiser. To protonate the ND1 and/or NE2 atoms enable the corresponding check-box(es) and click on the **Set Protonation** button.

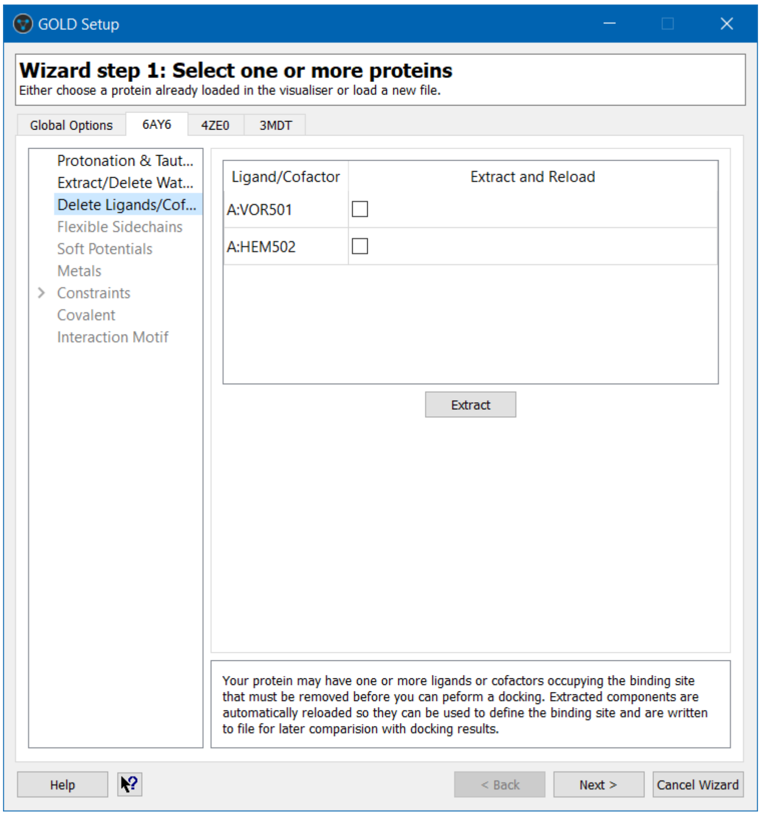
If you are unsure about the tautomeric state of a His residue, you should perform separate GOLD runs using the different possibilities.

4.4 Deleting Ligands, Cofactors and Metal Ions

The protein file may have one or more ligands/cofactors occupying the binding site. Cofactors are normally retained within the binding site for a docking so that they can make interactions with the docked ligand(s) in the same manner that the protein binding site makes interactions with the docked ligand(s). Ligands must be removed before you can perform a docking.

The removal of ligands is protein-specific thus first select the appropriate protein tab adjacent to the **Global Options** tab then click on **Delete Ligands/Cofactors** from the list of options given.

A list of the ligands/cofactors present in the protein file will be displayed. Each ligand/cofactor is assigned a unique identifier based on the protein chain and on the 3-character PDB chemical ID:



Wizard step 1: Select one or more proteins

Either choose a protein already loaded in the visualiser or load a new file.

Ligand/Cofactor	Extract and Reload
A:VOR501	□
A:HEM502	□

Your protein may have one or more ligands or cofactors occupying the binding site that must be removed before you can peform a docking. Extracted components are automatically reloaded so they can be used to define the binding site and are written to file for later comparision with docking results.

Clicking on a ligand/cofactor in this list will highlight it in the Hermes visualiser.

Select which ligands you wish to remove by switching on their corresponding **Extract and Reload** check-boxes, then click on the **Extract** button.

Extracted ligands are removed from the protein file and automatically reloaded into Hermes so that they can be used e.g. to define the binding site (see Defining a Binding Site from a Reference Ligand).

When extracting ligands, you will be asked if you want to write the ligand to a file. This can be useful for later comparison with docking results.

If the protein contains metal ions, then GOLD can automatically determine their coordination geometry. Virtual coordination points are then added at locations where GOLD is missing a coordination site and these coordination points are used as fitting points that can bind to acceptors (see Metal Ions). However, if you wish to delete a metal ion from the protein, select the appropriate protein tab then click on **Metals** from the list of available options. A list of the metal ions present in the protein file will be shown. To remove a metal select it in the list and click on the **Delete** button. Note that this is an expert GOLD option thus will be greyed out if you are using the wizard. To make use of this feature you will need to exit the wizard by clicking on the **Cancel Wizard** button.

4.5 Water Molecules

4.5.1 Methodology For Handling Waters

Water molecules often play key roles in protein-ligand recognition. Water molecules can either form mediating hydrogen bonds between protein and ligand, or they can be displaced by the ligand on binding.

GOLD allows waters to switch on and off (i.e. to be bound or displaced) and to rotate around their three principal axes (to optimise hydrogen bonding) during docking.

To predict whether a specific water molecule should be bound or displaced, GOLD estimates the free-energy change, ΔG_b, associated with transferring a water molecule from the bulk solvent to its binding site in a protein-ligand complex. ΔG_b for a given water molecule is defined as:

$\Delta G_b(W) = \Delta G_p(W) + \Delta G_i(W)$

$\Delta G_p(W)$ is a constant penalty added for each water molecule that is switched on and represents the loss of rigid-body entropy on binding to the target (hence rewarding water displacement). Note: ΔG_p values were optimised against a training set of 58 protein-ligand complexes for four targets (HIV-1 protease, factor Xa, thymidine kinase and the oligopeptide-binding protein Opp A) where water molecule play key roles in the recognition. Further details can be found in Modeling Water Molecules in Protein-Ligand Docking Using GOLD (see References).

$\Delta G_i(W)$ represents the intrinsic binding affinity of a water molecule and contains contributions resulting from interactions that the water forms with the protein and ligand (changes in the interactions between protein and ligand caused by introduction of the water are also accounted for).

Therefore, for a water molecule to be bound to a protein-ligand complex, its intrinsic binding affinity needs to outweigh the loss of rigid-body entropy on binding.

Where waters are specified in the `gold.conf`, an additional parameter, `S(bar)`, is added to the fitness score calculation:

Fitness = S(hb_ext) + 1.3750 * S(vdw_ext) + S(hb_int) + 1.0000 * S(int) + S(bar)

`S(bar)` is a barrier/penalty term associated with non-displacement of water.

4.5.2 Specifying Waters

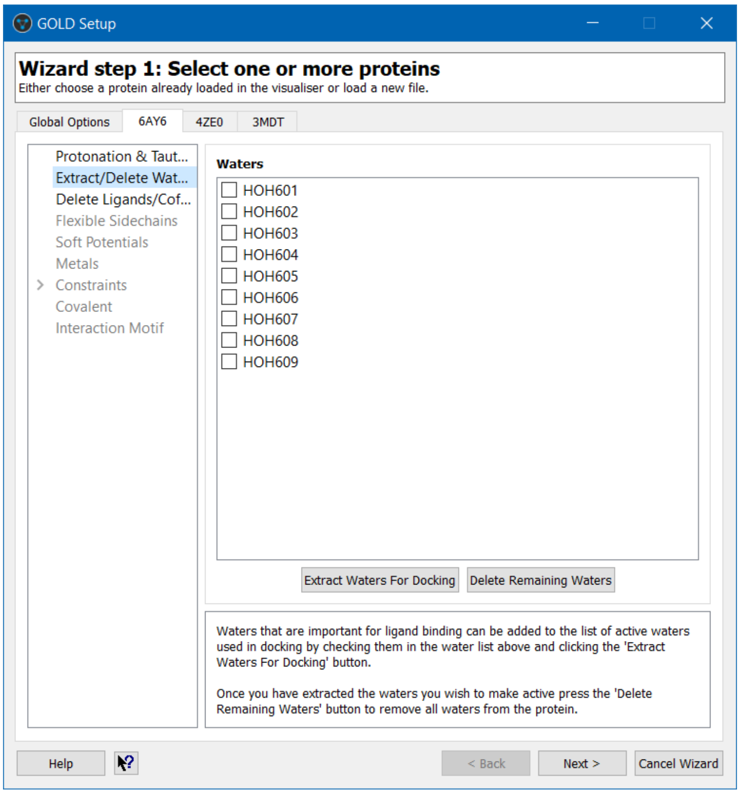
GOLD allows you to retain specific water molecules that are important to ligand binding (i.e. you can specify whether a particular water should be present or absent in the protein). Furthermore, for waters which are retained, GOLD can automatically determine whether a water should be bound or displaced by the ligand during docking (i.e. by toggling it on and off during the run). The orientation of the water hydrogen atoms can also be optimised by GOLD during docking. In addition, the location of each water molecule can be allowed to translate within a radius of 2 Å.

Click on **Configure Waters** from the list of **Global Options** given on the left of the **GOLD Setup** window.

Waters must be specified in separate files, i.e. one water per mol2 file. To specify the water files, select the **Add** button at the bottom of the **Configure Waters** dialogue. Use the file browser to locate the water files, select one or multiple files then hit **Open** to add them to the **Configure Waters** dialogue.

If the protein file contains all waters, i.e. active and non-active waters, the active waters must be extracted and the non-active waters deleted in the following way:

Click on the protein name tab, adjacent to the **Global Options** one (in the example below this is the **6AY6** tab), then select the **Extract/ Delete Waters** option.



Wizard step 1: Select one or more proteins

Either choose a protein already loaded in the visualiser or load a new file.

□ HOH601
□ HOH602
□ HOH603
□ HOH604
□ HOH605
□ HOH606
□ HOH607
□ HOH608
□ HOH609

Waters that are important for ligand binding can be added to the list of active waters used in docking by checking them in the water list above and clicking the 'Extract Waters For Docking' button.

Once you have extracted the waters you wish to make active press the 'Delete Remaining Waters' button to remove all waters from the protein.

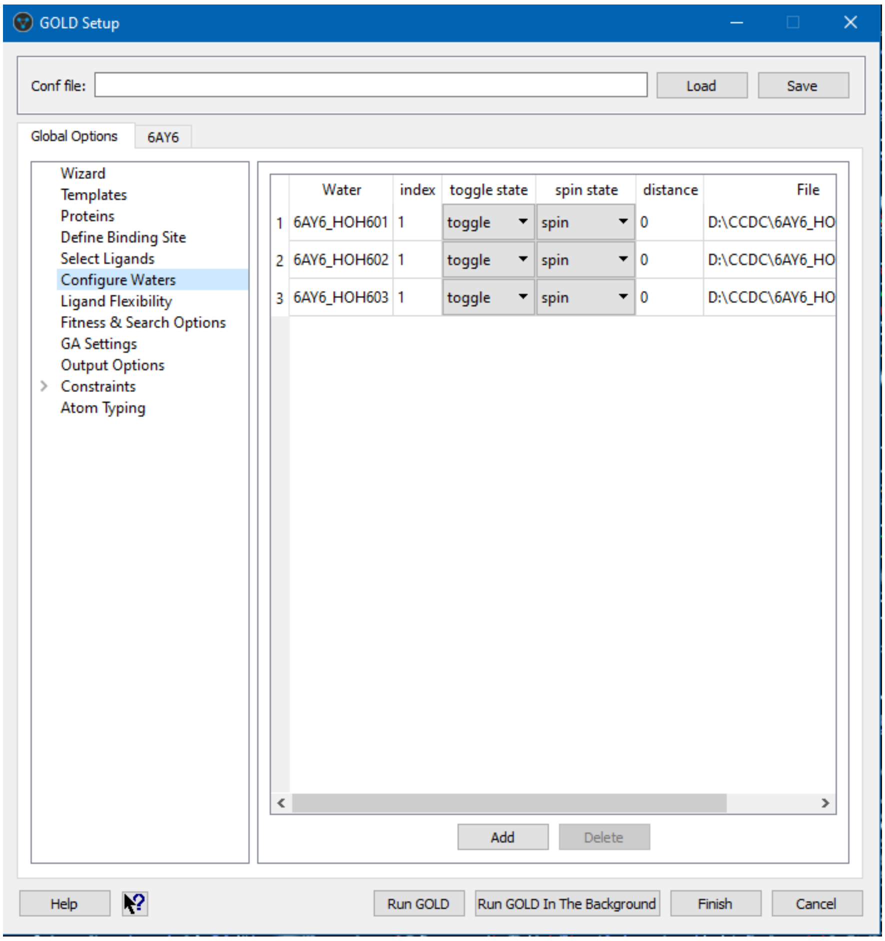
Select the waters you wish to keep either by selecting them in the Hermes 3D view or by activating their corresponding tick box. Selected waters can be unselected by deactivating their tickbox or by deselecting them in the Hermes 3D view.

Hit the **Extract Waters for Docking** button. This will write the waters to individual files in the working directory. The files will have names of the type `6AY6_HOH601.mol2`.

Once you have extracted the important waters, all other waters must be deleted from the protein file. This is done by hitting the **Delete Remaining Waters** button.

If the waters are extracted in this way, they are automatically added to the **Configure Waters** dialogue under the **Global Options** tab.

By default, each water molecule in the **Configure Waters** list will be allowed to toggle on and off in the binding site during docking and will be allowed to spin in order to optimise the orientation of the water hydrogen atoms. These settings can be customised for specific water molecules within this dialogue:



	Water	index	toggle state	spin state	distance	File
1	6AY6_HOH601	1	toggle	spin	0	D:\CCDC\6AY6_HO
2	6AY6_HOH602	1	toggle	spin	0	D:\CCDC\6AY6_HO
3	6AY6_HOH603	1	toggle	spin	0	D:\CCDC\6AY6_HO

For each water molecule listed the following can be specified:

The state of the water, available options are **On**: use the water for docking (i.e. present); **Off**: do not use the water for docking (i.e. absent); **Toggle**: have GOLD decide whether the water should be present or absent (i.e. bound or displaced by the ligand) during docking.

The orientation of the water hydrogen atoms, available options are **Spin:** have GOLD automatically optimise the orientation of the hydrogen atoms; **Trans_spin**: activate this option and input a translation value into the distance dialogue to make GOLD spin and translate the water molecule to optimise the orientation of the hydrogen atoms as well as the water molecule’s position within a user defined radius. Note that the distance value must be between 0 and 2 Å; **Fix:** use the orientation specified in the input file.

After docking a summary of which waters were retained or displaced and their contribution to the fitness score can be found in the **Analysis of active water placements** section of the `gold_ligand.log` file.

4.6 Defining the Binding Site

4.6.1 Overview

It is necessary to define the protein binding site. This can be done in several ways, e.g. by specifying the approximate centre of the binding site and taking all atoms that lie within a specified radius of this point (see Defining a Binding Site from a Point).

The binding site definition is detailed in the `Cavity atoms` section of the `gold_protein.log` file. The cavity atom selection can be saved as a protein atom subset and viewed within Hermes. To do this click on the **Add Definition as a Selection** button within the **Define Binding Site** section of the **GOLD Setup** window. You can then highlight the atoms belonging to the subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the Hermes visualiser display area.

Only those atoms specifically included in the binding site definition will be considered during docking. The binding site definition should therefore be large enough to contain any possible binding mode of the ligand and should include all atoms or residues that might be involved in ligand binding.

Since this binding site definition might include atoms that lie outside the cavity (i.e. on the surface of the protein) you can use cavity detection to restrict the binding site definition to concave parts of the binding site surface (see Cavity Detection).

Each atom in the defined binding site is tested for solvent accessibility, this is a two-step process:

First, the solvent accessible surface of each atom in the defined binding site is calculated. Potential donor and acceptor fitting points (used for ligand placement) are then generated for only those protein atoms that are accessible.

Second, the potential fitting points are themselves tested for solvent accessibility, and only those fitting points that are accessible are used.

Therefore, for a protein atom to be recognised as a donor or acceptor it must be included in the binding site definition, be solvent accessible and have at least one associated solvent accessible fitting point.

It is possible to remove the requirement for fitting points to be solvent accessible (see Solvent Accessibility). In this case fitting points would be generated for all solvent accessible donor and acceptor atoms within the binding site. Remember that these atoms are already deemed to be solvent accessible but it's their potential fitting points that may have been desolvated by neighbouring atoms.

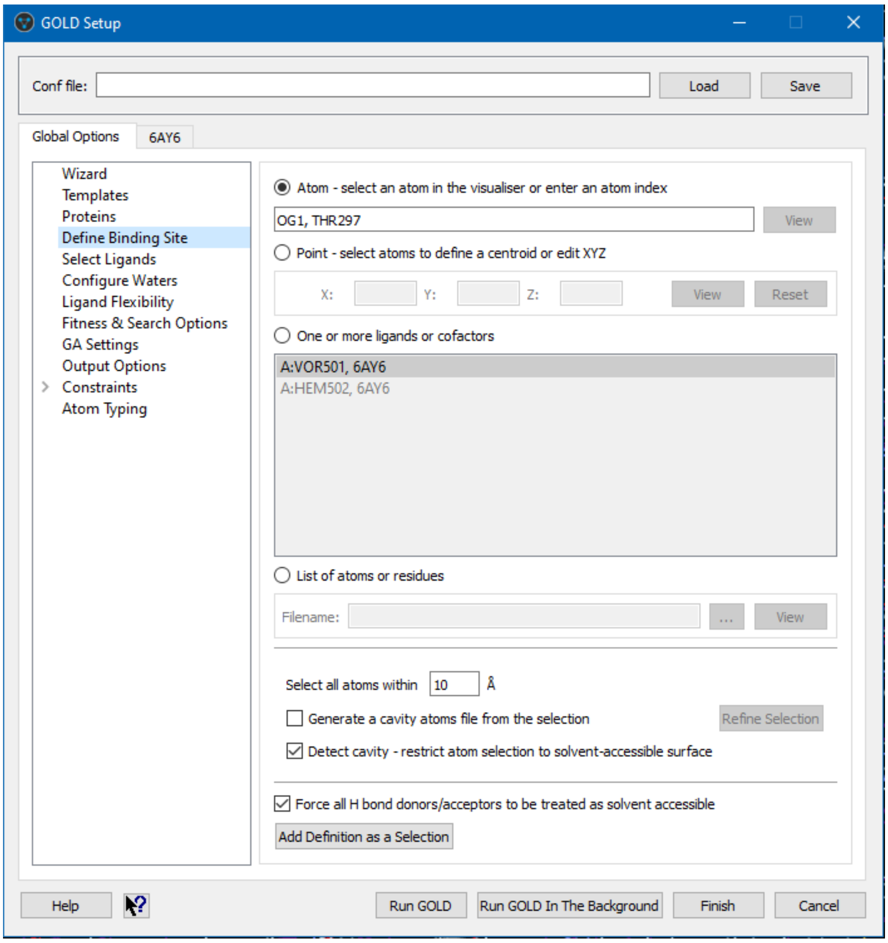
A `Fitting points summary` is provided in the `gold_protein.log` file. The polar fitting points used by GOLD are also saved as protein atom subsets within Hermes. Two subsets are saved, `donor hydrogens` and `lone pairs`. You can highlight the atoms belonging to any subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the Hermes visualiser display area.

Any water molecules and cofactors retained within the binding site will be included in what is termed the protein cavity atoms in the rest of this section.

4.6.2 Defining a Binding Site from an Atom

Click on **Define Binding Site** from the list of **Global Options** given on the left of the **GOLD Setup** window.

Switch on the radio button labelled **Atom**. Then, within the Hermes visualiser select a single solvent-accessible protein atom close to the centre of the active site of the protein.



Conf file:

● Atom - select an atom in the visualiser or enter an atom index

OG1, THR297

○ Point - select atoms to define a centroid or edit XYZ

X: Y: Z:

○ One or more ligands or cofactors

A:VOR501, 6AY6
A:HEM502, 6AY6

○ List of atoms or residues

Filename:

Select all atoms within 10 Å

☐ Generate a cavity atoms file from the selection

☑ Detect cavity - restrict atom selection to solvent-accessible surface

☑ Force all H bond donors/acceptors to be treated as solvent accessible

The approximate radius of the binding site must also be specified. If r is the radius, the binding site will be defined as all atoms that lie within r Å of the specified protein atom. By default, the binding site radius is set to 10.0 Å. This can be changed by entering a value in the box labelled **Select all atoms within**.

Residues, and cofactors if present, that have at least one of their atoms included in the binding site definition will be highlighted in the Hermes visualiser (carbon atoms of residues not included in the binding site definition will turn purple). When entering a new value in the **Select all atoms within** box it is necessary to hit the **Enter** key before the visualiser will update to reflect the changes made.

After visual inspection you may wish to manually refine the binding site definition. To do this, switch on the check-box labelled **Generate a cavity atoms file from the selection**. By enabling this option, the binding site definition will automatically be expanded to include all atoms in the existing definition plus all the atoms of their associated protein residues. To manually refine this selection, click on the **Refine Selection** button to open the **Refine Binding Site Selection** dialogue. All residues included in the binding site definition are listed. Residues can then be added or removed from the selection by clicking on atoms in the Hermes visualiser.

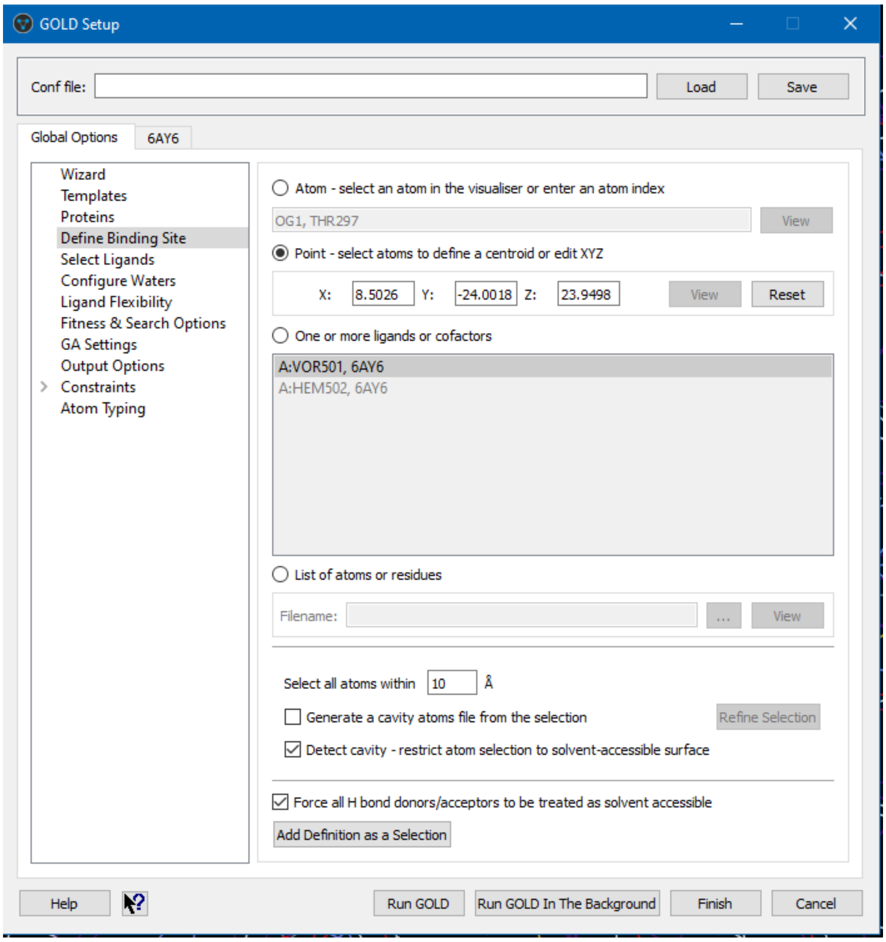
The cavity atom selection can be saved as a protein atom subset and viewed within Hermes. To do this click on the **Add Definition as a Selection** button. You can then highlight the atoms belonging to the subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the Hermes visualiser display area.

Note that it is not possible to define the binding site from an atom when performing an ensemble docking.

4.6.3 Defining a Binding Site from a Point

Click on **Define Binding Site** from the list of **Global Options** given on the left of the **GOLD Setup** window.

Switch on the radio button labelled **Point**. Then, within the Hermes visualiser click on one or more protein atoms in order to define a centroid close to the centre of the active site. Alternatively, the orthogonal x,y,z coordinates of a single solvent-accessible point approximately at the centre of the active site can be typed directly into the three boxes.



GOLD Setup

Conf file:

Global Options | 6AY6

○ Atom - select an atom in the visualiser or enter an atom index

OG1, THR297

● Point - select atoms to define a centroid or edit XYZ

X: 8.5026 Y: -24.0018 Z: 23.9498

○ One or more ligands or cofactors

A:VOR501, 6AY6
A:HEM502, 6AY6

○ List of atoms or residues

Filename:

Select all atoms within 10 Å

□ Generate a cavity atoms file from the selection

☑ Detect cavity - restrict atom selection to solvent-accessible surface

☑ Force all H bond donors/acceptors to be treated as solvent accessible

The approximate radius of the binding site must also be specified. If r is the radius, the binding site will be defined as all atoms that lie within r Å of the specified point. By default, the binding site radius is set to 10.0 Å. This can be changed by entering a value in the box labelled **Select all atoms within**.

Click on the **View** button to highlight, in the Hermes visualiser, those residues, and cofactors if present, that have at least one of their atoms included in the binding site definition (carbon atoms of residues not included in the binding site definition will turn purple). When entering a new value in the **Select all atoms within** box it is necessary to hit the **Enter** key before the visualiser will update to reflect the changes made.

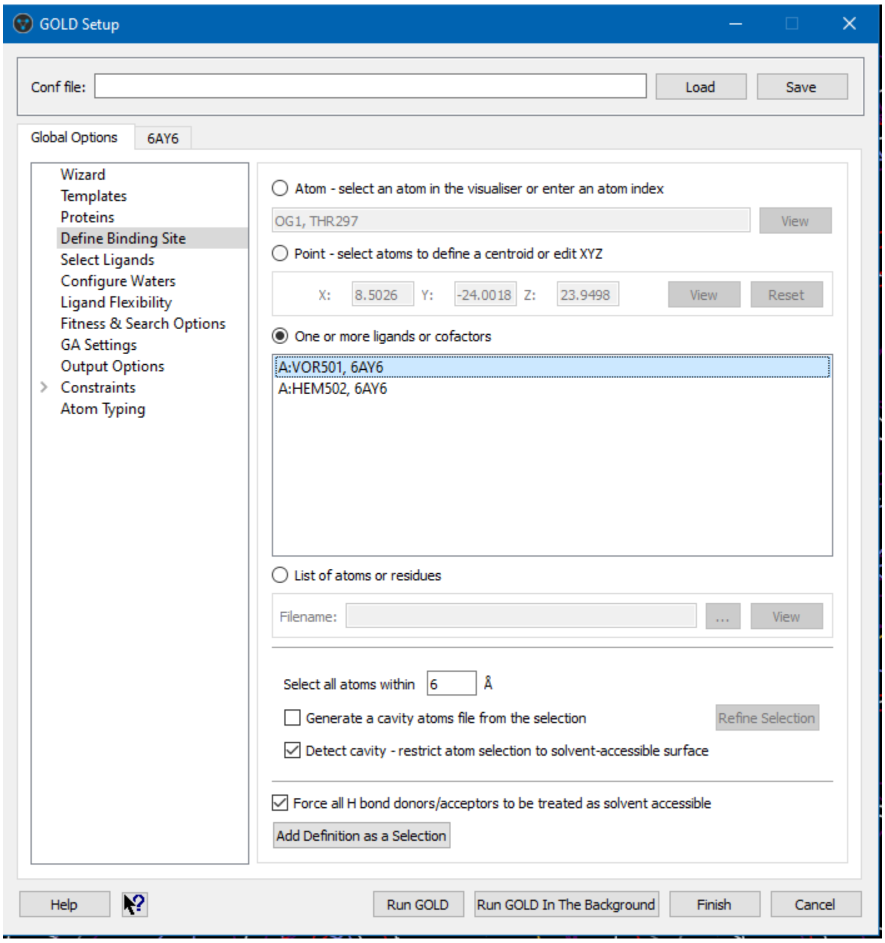
After visual inspection you may wish to manually refine the binding site definition. To do this, switch on the check-box labelled **Generate a cavity atoms file from the selection**. By enabling this option, the binding site definition will automatically be expanded to include all atoms in the existing definition plus all the atoms of their associated residues. To manually refine this selection, click on the **Refine Selection** button to open the **Refine Binding Site Selection** dialogue. All residues included in the binding site definition are listed. Residues can then be added or removed from the selection by clicking on atoms in the Hermes visualiser.

The cavity atom selection can be saved as a protein atom subset and viewed within Hermes. To do this click on the **Add Definition as a Selection** button. You can then highlight the atoms belonging to the subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the visualiser display area.

4.6.4 Defining a Binding Site from a Reference Ligand or Cofactor

Click on **Define Binding Site** from the list of **Global Options** given on the left of the **GOLD Setup** window.

Switch on the radio button labelled **One or more ligands or cofactors**. A list of those ligands and cofactors currently loaded in the Hermes visualiser will be shown. Loaded ligands and cofactors might typically include ligands in a known binding mode, or the co-crystallised ligand, or the co-crystallised cofactor. Select the reference ligand(s) and/or cofactor(s) you wish to use from this list. Multiple ligands/cofactors can be selected by left-clicking whilst holding down the **Shift** key. Only extracted ligands/cofactors can be used to define the binding site.



Conf file:

○ Atom - select an atom in the visualiser or enter an atom index

OG1, THR297

○ Point - select atoms to define a centroid or edit XYZ

X: 8.5026 Y: -24.0018 Z: 23.9498

● One or more ligands or cofactors

A:VOR501, 6AY6
A:HEM502, 6AY6

○ List of atoms or residues

Filename:

Select all atoms within 6 Å

☐ Generate a cavity atoms file from the selection

☑ Detect cavity - restrict atom selection to solvent-accessible surface

☑ Force all H bond donors/acceptors to be treated as solvent accessible

By default, all protein atoms within 6.0 Å of each selected ligand/ cofactor are used for the binding site definition. This can be changed by entering a new value in the box labelled **Select all atoms within**.

Residues that have at least one of their atoms included in the binding site definition will be highlighted in the Hermes visualiser (carbon atoms of residues not included in the binding site definition will turn purple). When entering a new value in the **Select all atoms within** box it is necessary to hit the **Enter** key before the visualiser will update to reflect the changes made.

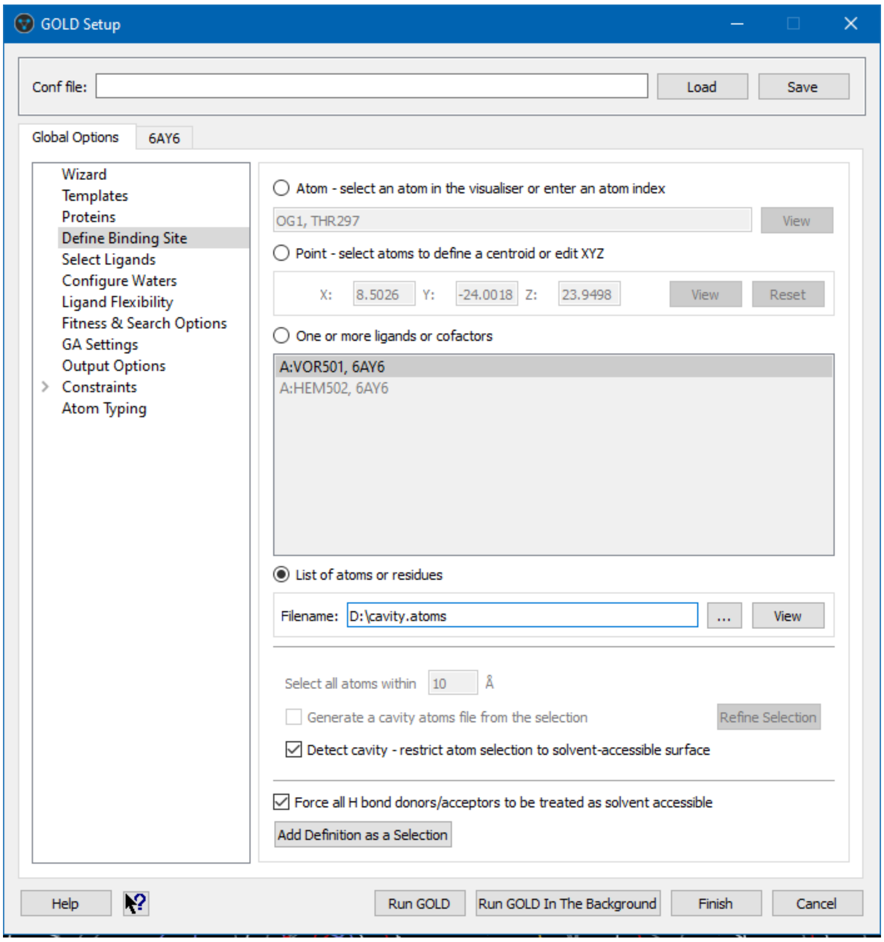
After visual inspection you may wish to manually refine the binding site definition. To do this, switch on the check-box labelled **Generate a cavity atoms file from the selection**. By enabling this option, the binding site definition will automatically be expanded to include all atoms in the existing definition plus all the atoms of their associated residues. To manually refine this selection, click on the **Refine Selection** button to open the **Refine Binding Site Selection** dialogue. All residues included in the binding site definition are listed. Residues can then be added or removed from the selection by clicking on atoms in the Hermes visualiser.

The cavity atom selection can be saved as a protein atom subset and viewed within Hermes. To do this click on the **Add Definition as a Selection** button. You can then highlight the atoms belonging to the subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the visualiser display area.

4.6.5 Defining a Binding Site from a List of Atoms or Residues

Click on **Define Binding Site** from the list of **Global Options** given on the left of the **GOLD Setup** window.

Switch on the radio button labelled **List of atoms or residues**. A file which contains a list of protein atom numbers or residues must be specified. Either enter the path and filename of the file, or click on the **...** button and use the file selection window to choose the file.



Conf file:

○ Atom - select an atom in the visualiser or enter an atom index

OG1, THR297

○ Point - select atoms to define a centroid or edit XYZ

X: 8.5026 Y: -24.0018 Z: 23.9498

○ One or more ligands or cofactors

A:VOR501, 6AY6
A:HEM502, 6AY6

● List of atoms or residues

Filename: D:\cavity.atoms

Select all atoms within 10 Å

□ Generate a cavity atoms file from the selection

☑ Detect cavity - restrict atom selection to solvent-accessible surface

☑ Force all H bond donors/acceptors to be treated as solvent accessible

When specifying a list of atoms, the atom numbers as they appear in the input protein must be provided. Multiple atom numbers are permitted on each line in the file. It is therefore possible to re-use an existing active site definition by using the list of active atoms printed in the `protein.log` file. Example file format is shown below:

```
17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36
37 38 39 40 402 403 404 405 406 407
408 409 410 411 412 926 927 928 929 930
931 932 933 934 935 936 937 938 939 947
948 949 950 951 952 953 954 955 956 957
958 959 960 961 962 963 964 965 966 967
968 969 970 971 972 973 974 1005 1006 1007
1008 1009 1010 1011 1012 1013 1014 1015 1016 1017
1018 1367 1368 1369 1370 1371 1372 1373 1374 1375
1376 1377 1378 1379 1380 1381 1382 1383 1384 1385
1386 1387 1388 1389 1390 1391 1392 1393 1394 1395
1396 1397 1398 1399 1400 1401 1402 1423 1424 1425
1426 1427 1428 1429 1430 1431 1432 1433 1434 1435
1436 1437 1438 1439 1460 1461 1462 1463 1464 1465
1466 1467 1468 1469 1470 1471 1472 1473 1474 1475
1476 1592 1593 1594 1595 1596 1597 1598 1599 1600
1601 1602 1603 1604 1605 1606 1808 1809 1810 1811
1812 1813 1814 1815 1816 1817 1818 1819 1820 1821
1822 1823 1824 1844 1845 1846 1847 1848 1849 1850
```

When specifying a list of residues, the residues can be extracted from any text file, including a standard GOLD solution file (GOLD writes the active site residues list to the solution files if output of rotatable hydrogens is turned on).

The following formatting restrictions apply:

- The list must begin with the following tag on its own line:

```
> <Gold.Protein.ActiveResidues>
```

- The list must end with a blank line (or the end of the text file).
- GOLD will read multiple residue names from one line, but lines must not exceed 250 characters in length.
- Residue names must be separated by a space, for example:

```
> <Gold.Protein.ActiveResidues>
    HIS69 ARG71 GLU72 ARG127 ASN144 ARG145 GLY155 ALA156 GLU163
        THR164
    HIS196 SER197 TYR198 SER199 LEU201 LEU203 ILE243 ILE244 ILE247
    TYR248 GLN249 ALA250 GLY253 SER254 ILE255 THR268 GLU270 PHE279
    ZN309
```

NB! If an mmCIF file is used as an input file, residues' IDs have to match the following fields: `_atom_site.label_comp_id`, `_atom_site.label_seq_id`.

All solvent-accessible protein acceptor and donor atoms available to the ligand are taken from the list. The file should contain all atoms or residues which are required to explicitly define the protein active site.

Click on the **View** button to highlight in the Hermes visualiser those residues that have at least one of their atoms included in the binding site definition (carbon atoms of residues not included in the binding site definition will turn purple).

The cavity atom selection can be saved as a protein atom subset and viewed within Hermes. To do this click on the **Add Definition as a Selection** button. You can then highlight the atoms belonging to the subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the Hermes visualiser display area.

4.6.6 Cavity Detection

The binding site can be defined in several ways, e.g. by specifying the approximate centre of the binding site and taking all atoms that lie within a specified radius of this point (see Defining a Binding Site from a Point).

This binding site definition might therefore include atoms that lie outside the cavity (i.e. on the surface of the protein).

You can use a cavity detection algorithm (LIGSITE: Automatic and efficient detection of potential small molecule binding sites in proteins. M. Hendlich, F. Rippmann, G. Barnickel. Journal of Chemical Information and Computer Sciences 1997, **37**, 774-778) to restrict the region of interest to concave parts of the binding site surface.

To enable cavity detection switch on the check-box labelled **Detect cavity - restrict atom selection to solvent-accessible surface**. This option is available by clicking on **Define Binding Site** from the list of **Global Options** given on the left of the **GOLD Setup** window.

After docking the atoms included in the binding site definition are listed in the `Cavity atoms` section of the `gold_protein.log` file. The cavity atom selection is also saved as a protein atom subset within Hermes. You can highlight the atoms belonging to any subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the Hermes visualiser display area.

It is possible to generate contour (`.acnt`) files of the cavity used by GOLD by editing `WRITE_CNT_FILES = 0` to `WRITE_CNT_FILES = 3` in the `gold.params` file (see Altering GOLD Parameters: the gold.params File). The `.acnt` files produced can be read into Hermes following the docking via **Display**, **Contour Surfaces**. Further details on how to read `.acnt` files into Hermes are provided in the Hermes User Guide.

4.6.7 Solvent Accessibility

Each atom in the defined binding site is tested for solvent accessibility, this is a two-step process:

First, the solvent accessible surface of each atom in the defined binding site is calculated. Potential donor and acceptor fitting points (used for ligand placement) are then generated for only those protein atoms that are accessible.

Second, the potential fitting points are themselves tested for solvent accessibility, and only those fitting points that are accessible are used.

It is possible to remove the requirement for fitting points to be solvent accessible. In this case fitting points would be generated for all solvent accessible donor and acceptor atoms within the binding site. Remember that these atoms are already deemed to be solvent accessible but it's their potential fitting points that may have been desolvated by neighbouring atoms.

This option can be used e.g. to avoid problems with solvent accessibility of backbone carbonyls in kinases where one of the carbonyl lone pairs is typically desolvated by a neighbouring atom.

To generate fitting points for all solvent accessible donor and acceptor atoms switch on the check-box labelled **Force all H bond donors/acceptors to be treated as solvent accessible**. This option is available by clicking on **Define Binding Site** from the list of **Global Options** given on the left of the **GOLD Setup** window.

A `Fitting points summary` is provided in the `gold_protein.log` file. The polar fitting points used by GOLD are also saved as protein atom subsets within Hermes. Two subsets are saved, `donor hydrogens` and `lone pairs`. You can highlight the atoms belonging to any subset by picking the required subset from the **Atom lists** pull-down menu, which is situated above the Hermes visualiser display area.

4.7 Rotatable O-H and NH_3 Groups

The torsion angles of Ser, Thr and Tyr hydroxyl groups will be optimised by GOLD, so their starting positions do not matter.

Specifically, each Ser, Thr and Tyr OH will be allowed to rotate to optimise its hydrogen-bonding to the ligand.

Lysine NH_3^+ groups are similarly optimised, unless they are held in place by strong H-bonds to neighbouring protein residues.

The optimised positions of polar protein hydrogen atoms generated during docking can be written to GOLD solution files (see Controlling the Information Written to Ligand Solution Files).

It is possible to run a docking keeping these rotatable bonds static, if required (see Docking into a Rigid Protein).

4.8 Docking into a Rigid Protein

Even when not using advanced protein flexibility (see Protein Flexibility), serine, threonine and tyrosine hydroxyl groups are optimised i.e. rotated during docking, as are lysine NH_3^+ groups.

In some cases, it might be necessary to dock into a rigid protein, i.e. to keep all the polar hydrogen atoms fixed during a docking.

To dock into a rigid protein, select **Proteins** from the list of **Global Options** given on the left of the **GOLD Setup** window and activate the **Fix all protein rotatable bonds** tick box.

4.9 Metal Ions

4.9.1 Preparing a Protein Input File which Contains a Metal Ion

There are some additional requirements when preparing a protein input file which contains a metal ion.

The metal ion must be coordinated to at least two protein atoms or water molecules so that GOLD can predict the coordination geometry (see Automatic Determination of Metal Coordination Geometries).

In the protein input file, the metal ion should not have any bonds to coordinating atoms. If these are present in the original structure file, they must be deleted.

Note: GOLD can only handle the hardcoded metal atom types (see Automatic Determination of Metal Coordination Geometries); it is not possible to add user defined metal atom types.

If a particular metal ion is not required, it can be removed from the protein (see Deleting Ligands, Cofactors and Metal Ions).

4.9.2 Automatic Determination of Metal Coordination Geometries

GOLD is able to recognise the following metal coordination geometries:

Template	Geometry	Coordination Number
TETR	Tetrahedral	n=4
TBP	Trigonal bipyramidal	n=5
OCT	Octahedral	n=6
CTP	Capped trigonal prism	n=7
PBP	Pentagonal bipyramidal	n=7
SQAP	Square prism	n=8
ICO	Icosahedral	n=10
DOD	Dodecahedral	n=12

In order to determine the coordination geometry of a particular metal atom, GOLD performs a permuted superimposition of coordination geometry templates onto the coordinating atoms found in the protein (e.g. if there are only two coordinating atoms in the protein, then every unique pair of coordinating template atoms are selected and superimposed on the system in the protein).

Coordination fitting points are then generated using the template that gives the best fit (based on RMSD).

The geometry templates used for given metals are defined in the `gold.params` file in the section headed `# Metals` (for explanation of parameters refer to comments in the `gold.params` file):

H-Bonding Type	**SYBYL Atom Type**	**Atom Type (default or elucidated)**	**Donor (D), Acceptor (A), or Metal (M)**	**Allowed Coordination Geometries**	**Coordination Distance**
MGD	Mg	DEF	M	4, 6	2.05
ZND	Zn	DEF	M	4, 5, 6	2.09
MND	Mn	DEF	M	4, 6	2.06
FED	Fe	DEF	M	4, 6	1.98
CAD	Ca	DEF	M	6, 7	2.44
COBD	Co.oh	DEF	M	6	2.09
GDD	Gd	DEF	M	6	2.44
CUD	Cu	DEF	M	4, 6	2.10
HGD	Hg	DEF	M	4, 6	2.40
CDD	Cd	DEF	M	4, 6	2.30
NID	Ni	DEF	M	4, 6	2.15
VD	V	DEF	M	4, 6	2.10

For example, for a Zn atom GOLD will attempt to match coordination geometries 4, 5 and 6 (tetrahedral, trigonal bipyramidal, and octahedral templates) onto the coordinating atoms found in the protein.

The template that gives the best match will then be used to generate coordination fitting points.

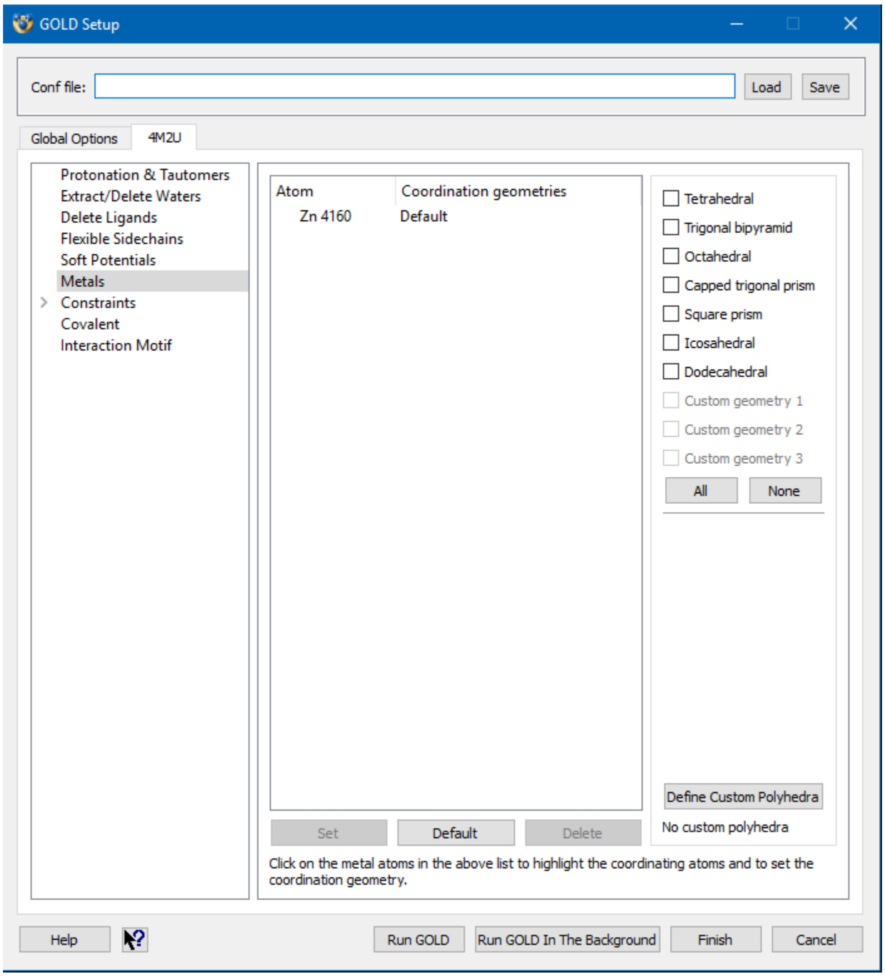
Details of the coordination geometry determination are given in the `gold_protein.log` file.

The output file `gold_protein.mol2` will contain a number of dummy atoms representing idealised coordination positions. These dummy atoms will be connected to the metal ion. Any unoccupied coordination points will then be available for ligand binding (see Metal-Ligand Interactions).

4.9.3 Specifying Metal Coordination Geometries Manually

It is possible to manually specify coordination geometries for particular metal atoms. This can be used to allow non-standard metal coordination geometries, or to limit the number of possible geometries that GOLD checks (i.e. it is possible to overrule the default geometries for the corresponding metal type defined in the `gold.params` file (see Automatic Determination of Metal Coordination Geometries).

Metal ions are protein-specific so first activate the relevant protein tab adjacent to the **Global Options** tab (e.g. **4M2U** in the example below). Click on **Metals** from the list of options given on the left of the **GOLD Setup** window.



Atom	Coordination geometries
Zn 4160	Default

□ Tetrahedral
□ Trigonal bipyramid
□ Octahedral
□ Capped trigonal prism
□ Square prism
□ Icosahedral
□ Dodecahedral
□ Custom geometry 1
□ Custom geometry 2
□ Custom geometry 3

No custom polyhedra

Click on the metal atoms in the above list to highlight the coordinating atoms and to set the coordination geometry.

Any metals in the currently loaded protein will be recognised and listed. By default, only the coordination geometries for the corresponding metal type defined in the `gold.params` file will be considered during docking. For example, for a Zn atom GOLD will attempt to match coordination geometries 4, 5 and 6 (tetrahedral, trigonal bipyramidal, and octahedral templates) onto the coordinating atoms found in the protein (see Automatic Determination of Metal Coordination Geometries).

If you wish to manually specify coordination geometries for particular metal atoms, then select the allowed coordination geometries by enabling the corresponding check box(es). Once the allowed geometries have been selected for a particular metal atom click on the **Set** button.

If the list of pre-defined coordination geometries does not contain a suitable geometry, then you can define a custom metal coordination geometry (see Defining Custom Metal Coordination Geometries).

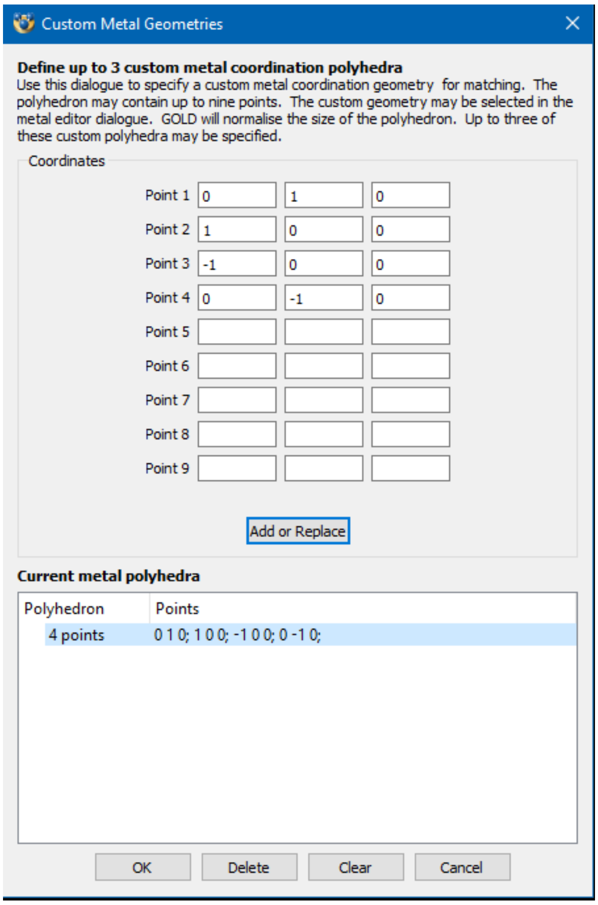
To return the allowed coordination geometries of a particular metal to the defaults defined in the `gold.params` file highlight the entry and hit the **Default** button.

4.9.4 Defining Custom Metal Coordination Geometries

It is possible to specify custom metal coordination geometries which can subsequently be used to derive ligand binding points around particular metal atoms.

GOLD will normalise the size of the custom polyhedron to the appropriate metal-chelator distance before matching it to the metal and the coordinating atoms found in the protein.

Click on the protein tab then select **Metals** from the list of options given on the left of the **GOLD Setup** window. Click on the **Define Custom Polyhedra** button. The **Custom Metal Geometries** window will appear:



Define up to 3 custom metal coordination polyhedra

Use this dialogue to specify a custom metal coordination geometry for matching. The polyhedron may contain up to nine points. The custom geometry may be selected in the metal editor dialogue. GOLD will normalise the size of the polyhedron. Up to three of these custom polyhedra may be specified.

Coordinates

Point 1	0	1	0
Point 2	1	0	0
Point 3	-1	0	0
Point 4	0	-1	0
Point 5			
Point 6			
Point 7			
Point 8			
Point 9			

Current metal polyhedra

Polyhedron	Points
4 points	0 1 0; 1 0 0; -1 0 0; 0 -1 0;

Custom metal polyhedron may contain up to nine points. Each point in the custom polyhedron must be specified using a vector (assuming the centre of your polyhedron is at the origin).

For example, to set up a custom square planar geometry you must specify four points using the following vectors:

0, 1, 0
1, 0, 0
-1, 0, 0
0, -1, 0

Assuming the metal is on the origin (0,0,0), GOLD will then attempt to match the specified vectors onto the metal-to-protein-atom vectors found in the protein (vectors are normalised to a metal-to-chelator distance of 2.0 Å).

Once vectors for each point in the polyhedron have been defined click on the **Add or Replace** button to add the custom definition to the list of coordination geometries available for selection (see Specifying Metal Coordination Geometries Manually).

Repeat the above procedure if you want to specify an additional custom polyhedron. It is possible to set up to three custom metal polyhedra.

To edit a custom polyhedron, highlight the corresponding entry in the **Custom Metal Geometries** window, make the required changes and then hit the **Add or Replace** button.

To remove a custom polyhedron, highlight the corresponding entry in the **Custom Metal Geometries** window and hit the **Delete** button, or to remove all entries hit the **Clear** button.

Once defined the custom geometries will be available for selection when manually specifying allowed coordination geometries (see Specifying Metal Coordination Geometries Manually).

4.9.5 Metal-Ligand Interactions

Metal coordination in GOLD is modelled as 'pseudo-hydrogen bonding'.

Metal-ligand interactions will typically involve the metal binding to, for example, carboxylate ions, deprotonated histidines (i.e. negatively charged), and phenolates. Therefore, metals can be considered to bind to H-bond acceptors and the metal will compete with H-bond donors for interaction.

Consequently, GOLD uses the following approach for handling metals:

Virtual coordination points are added at locations where GOLD is missing a coordination site.

These coordination points are then used as fitting points that can bind to acceptors.

4.9.6 Heme Containing Proteins

The paper by S. B. Kirton et al. (Prediction of binding modes for ligands in the cytochromes P450 and other heme-containing proteins in Proteins: Structure, Function, and Bioinformatics, **58**, 836-844, 2005) describes the use of ligand-specific iron parameters in the context of docking to heme-containing proteins. This extended metal parameterisation is available for the fine-tuning of metal interactions, so that e.g. metal-ligand interactions can specifically be addressed depending on the metal contact.

The protein does not need to be set up in a special way to make use of these parameters however the standard set-up should be followed (see Preparing a Protein Input File which Contains a Metal Ion).

Further information on setting up a GOLD run with these settings is available (see Heme Scoring Function).

5 Protein Flexibility

Protein flexibility can be handled in one of three ways in GOLD:

- By allowing side chains to rotate within user-defined bounds during docking (see Side Chain Flexibility);
- By docking into subtly different versions of the same protein (see Ensemble Docking);
- By docking using soft potentials (see Allowing for Localised Movements: Docking with Soft Potentials).

5.1 Side Chain Flexibility

5.1.1 Introduction

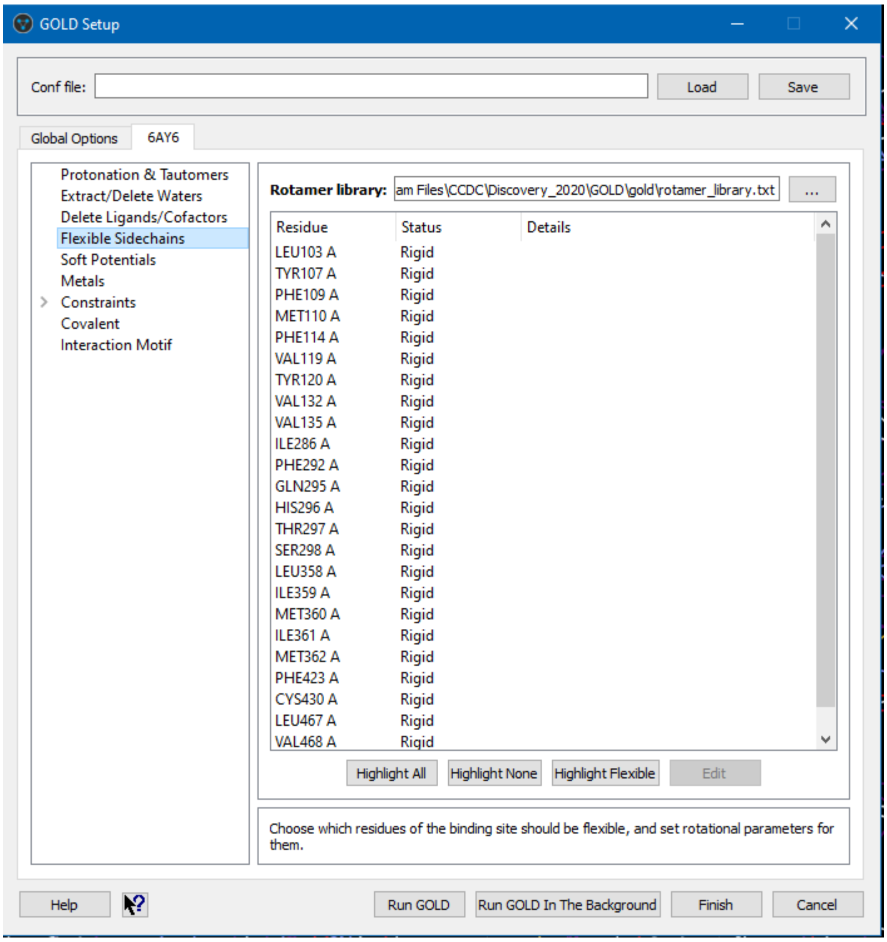
You may specify that one or more protein side chains are to be treated as flexible. Each flexible side chain will be allowed to undergo torsional rotation around one or more of its acyclic bonds during docking.

Making a side chain flexible can make docking more difficult because it increases the search space that must be explored. It may also increase the chance of false positives (i.e. ligands that appear to dock well but do not actually bind). Therefore, you should only make a side chain flexible if you have good reason to believe (e.g. from X-ray data) that it is likely to move in response to ligand binding.

5.1.2 Specifying Flexible Side Chains

You may specify that one or more protein side chains are to be treated as flexible during docking.

Flexible side chains are protein-specific, thus click on the protein tab adjacent to the **Global Options** tab (in the example below the protein tab is **6AY6**), then select **Flexible Sidechains** from the list of available options. A list of the side chains included within the binding site definition will be displayed.



Rotamer library: am Files\CCDC\Discovery_2020\GOLD\gold\rotamer_library.txt

Residue	Status	Details
LEU103 A	Rigid	
TYR107 A	Rigid	
PHE109 A	Rigid	
MET110 A	Rigid	
PHE114 A	Rigid	
VAL119 A	Rigid	
TYR120 A	Rigid	
VAL132 A	Rigid	
VAL135 A	Rigid	
ILE286 A	Rigid	
PHE292 A	Rigid	
GLN295 A	Rigid	
HIS296 A	Rigid	
THR297 A	Rigid	
SER298 A	Rigid	
LEU358 A	Rigid	
ILE359 A	Rigid	
MET360 A	Rigid	
ILE361 A	Rigid	
MET362 A	Rigid	
PHE423 A	Rigid	
CYS430 A	Rigid	
LEU467 A	Rigid	
VAL468 A	Rigid	

Choose which residues of the binding site should be flexible, and set rotational parameters for them.

By default, all side chains will be treated as rigid, i.e. they will be held fixed at their input conformation during the docking. To make a side chain flexible you can either:

- Select the side chain by clicking on it in the list. Once selected, a side chain will be highlighted in the Hermes visualiser. Once a side chain has been selected you will be required to define one or more allowed rotamers. Each rotamer specifies the torsion angles that are permitted to vary, and the allowed values or ranges of values for those torsion angles. Click on the **Edit** button. The resulting **Edit Rotamer Library** dialogue should then be used to set the rotational parameters for the selected side chain (see Defining Rotamers).

- Alternatively, select a side chain within Hermes by right-clicking on it and selecting **Set flexibility parameters** from the drop-down list. The resulting **Edit Rotamer Library** dialogue should then be used to set the rotational parameters for the selected side chain (see Defining Rotamers).

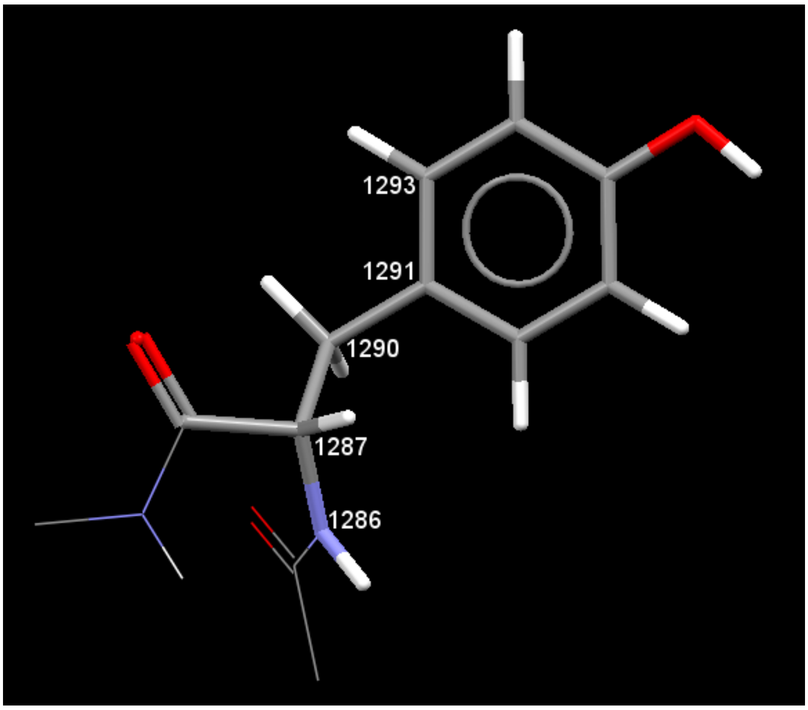
A maximum of 10 flexible side chains can be defined.

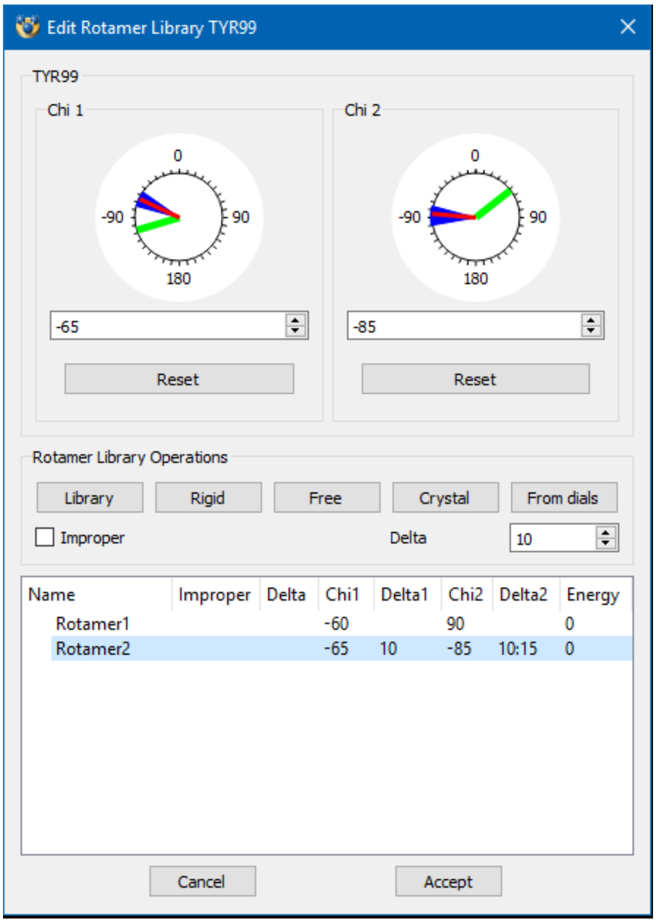
Once rotational parameters have been specified the Status of those side chains made flexible will be updated in the list. To highlight in the Hermes visualiser only those side chains that have been made flexible click on the **Highlight Flexible** button. To highlight all side chains in the defined binding site, click on **Highlight All**, and to remove all highlighting click on the **Highlight None** button.

5.1.3 Defining Rotamers

Once a side chain has been specified as flexible you will be required to define one or more allowed rotamers. Each rotamer specifies the torsion angles that are permitted to vary, and the allowed values or ranges of values for those torsion angles. Up to 50 rotamers can be defined for each flexible side chain.

Rotamers are defined using the **Edit Rotamer Library** dialogue which is opened when selecting a side chain (see Specifying Flexible Side Chains). For example, consider the side chain TYR99, shown below.





Chi 1: -65

Chi 2: -85

☐ Improper

Delta: 10

Name	Improper	Delta	Chi1	Delta1	Chi2	Delta2	Energy
Rotamer1			-60		90		0
Rotamer2			-65	10	-85	10:15	0

Chi1 is the first rotatable torsion in the side chain. In this example, it corresponds to rotation around Cα-Cβ, so the atoms will be the backbone N, (= atom 1286), CA (1287), CB (1290) and CG (1291).

Chi2 is the second rotatable torsion and corresponds to rotation around Cβ-Cγ, so the atoms are CA (1287), CB (1290), CG (1291) and CD1 (1293).

Thus, **Rotamer1** specifies the first set of allowed values for **chi1** and **chi2**. In this example, this is chi1 = -60, chi2 = 90.

Rotamer2 specifies the second set of allowed values. In this example delta1 =10 and specifies the allowed range (delta1 - chi1) to (delta1 + chi1), while delta2 = 10:15 and specifies the range (chi2 - 10) to (chi2 + 15).

In summary, the effect of these two rotamers is therefore to allow Tyr99 to adopt the conformation of precisely chi1 = -60, chi2 = 90, or any conformation in the range chi1 = -55 to -75, chi2 = -95 to -70.

Each rotamer therefore describes one allowed conformation of the side chain as defined by the torsion angles values (chi1, chi2, etc.) and their allowed ranges (delta1, delta2, etc.). Rotamers can be defined in the following ways:

- Setting a side chain to be rigid: To fix a particular side chain at its input conformation (i.e. to make it non-flexible during docking) click on the **Rigid** button. Any previously defined rotamers will be lost.

- Setting a side chain to be freely rotatable: To allow a side chain to rotate freely during docking click on the **Free** button. This will define a single rotamer where all rotatable torsions are permitted to vary over the range -180 to +180. Any previously defined rotamers will be lost.

- From a rotamer library: The file `<GOLD_DIR>/gold/rotamer_library.txt` contains information taken from the paper The Penultimate Rotamer Library, S. C. Lovell et al., Proteins, **40**, 389-408, 2000. It is a compilation of the most commonly observed side chain conformations for the naturally occurring amino acids. To define rotamers corresponding to these commonly observed side chain conformations click on the **Library** button. Note that the library settings are simply a starting point; users are encouraged to generate their own rotamers for optimal results.

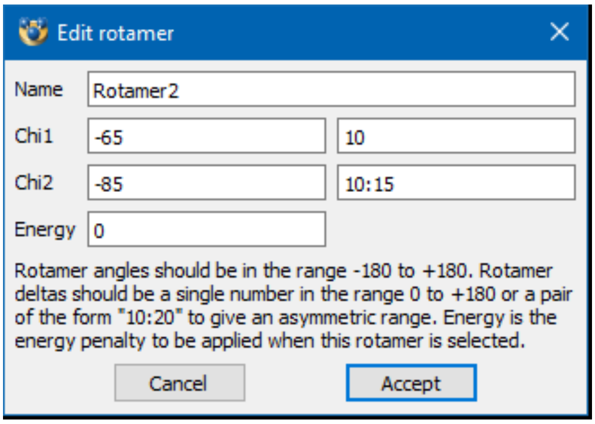
- From the protein input file: Click on the **Crystal** button to define a rotamer in which all rotatable torsions in the side chain will be allowed to vary over the range (delta - chi) to (delta + chi), where chi values are taken from the protein input file.

- From dials: Rotamers can be specified directly. To set a chi value click on the dial and while holding down the mouse button move the red indicator line to the required position. The corresponding torsion will rotate within the Hermes visualiser to show the current value. Alternatively, type the required chi value into the entry box directly under the dial. Once the chi and delta values have been set click on the **From Dials** button to add this rotamer definition.

5.1.4 Deleting and Editing Rotamer Definitions

To remove or copy a particular rotamer definition right click on the rotamer name in the **Edit Rotamer Library** dialogue and select either **Delete rotamer** or **Copy rotamer** from the resulting drop-down menu.

To edit a particular rotamer definition right click on the rotamer name in the **Edit Rotamer Library** dialogue and select **Edit this rotamer** from the resulting drop-down menu. This will open the **Edit rotamer** dialogue:



Name: Rotamer2
Chi1: -65 | 10
Chi2: -85 | 10:15
Energy: 0

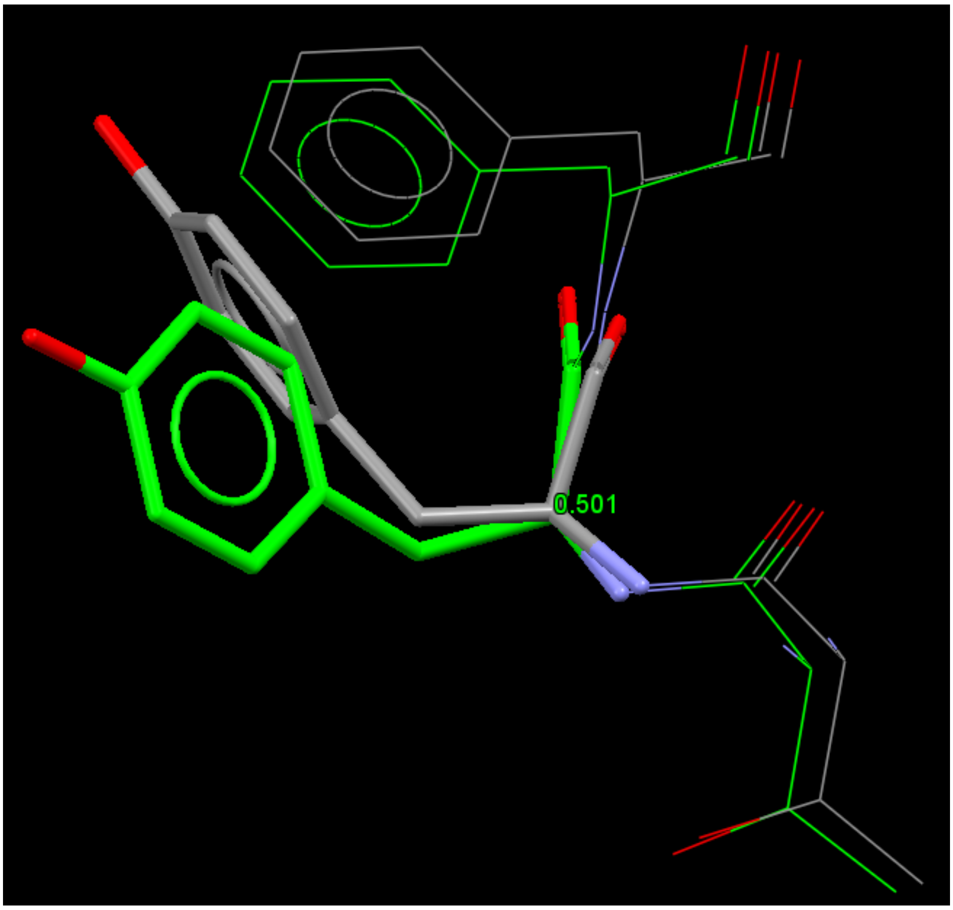
Rotamer angles should be in the range -180 to +180. Rotamer deltas should be a single number in the range 0 to +180 or a pair of the form "10:20" to give an asymmetric range. Energy is the energy penalty to be applied when this rotamer is selected.

The rotamer **Name**, **Chi** and corresponding **Delta** values can be changed by typing into the appropriate entry boxes. **Chi** values should be a single number in the range -180 to +180. **Delta** values should be a single number in the range 0 to -180, or a pair of numbers of the form x:y to specify an asymmetric range.

An **Energy** may be assigned to a given rotamer. This will penalise (i.e. reduce) the fitness by the value specified if the side chain is placed in the defined conformation. In other words, it makes this conformation less favourable. A negative **Energy** value can be entered, its effect would be to improve (i.e. increase) the fitness if the side chain is placed in the defined conformation.

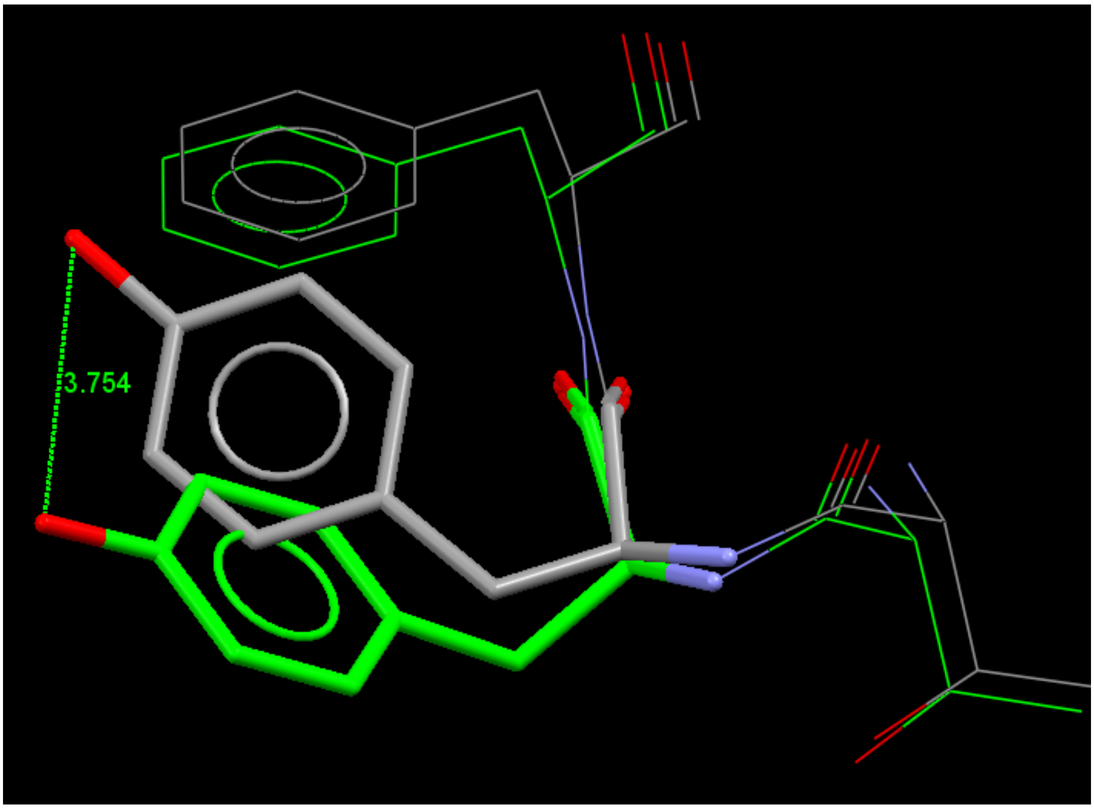
5.1.5 Allowing a Localised Backbone Movement

Quite often, a side chain rotation is accompanied by a small change in the local backbone conformation. For example, the figure below shows a detail from an overlay of two PDB structures (1qon, 1dx4, with green and grey carbon atoms, respectively) of the same enzyme:



Not only has the Tyr370 side chain rotated around Cα-Cβ and Cβ-Cγ, but there has also been a small backbone movement, primarily affecting the position of the Cα atom.

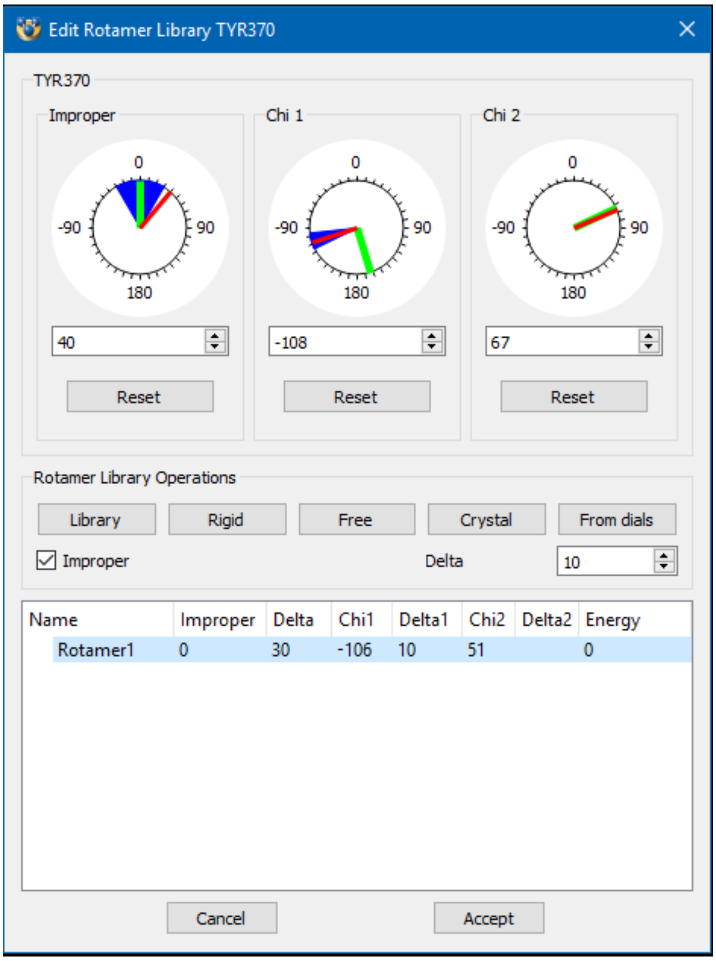
Although minor (the two Cα positions are only 0.5 Å apart), this movement is extremely important because it alters the vector direction Cα-Cβ, and this can have a big leverage effect on the positions of atoms further down the side chain. In this case, it is impossible to overlay the Tyr370 side chain of 1dx4 closely onto that of 1qon simply by rotating around the Cα-Cβ and Cβ-Cγ bonds. This is about as close as one can get:



The backbone movement can be mimicked by allowing the Cα atom and the attached side chain to rotate around the N-C vector, where N and C are the backbone atoms on either side of the Cα atom. This is defined as a rotation of the improper torsion defined by the atom sequence CA-N-C-CA.

To define an improper torsion enable the **Improper** check-box in the **Edit Rotamer Library** dialogue. An additional **Improper** torsion angle dial will become available for defining rotamers (see Defining Rotamers).

In the example shown below, an additional improper torsion has been specified. The specification for the improper torsion angle will allow a rotation of (+ or -)30 degrees around the N-C vector, the zero angle corresponding to the Cα position given in the protein input file.



Improper: 40

Chi 1: -108

Chi 2: 67

☑ Improper

Delta: 10

Name	Improper	Delta	Chi1	Delta1	Chi2	Delta2	Energy
Rotamer1	0	30	-106	10	51		0

It is not easy to decide on suitable rotation limits for improper torsions - a trial and error approach is normally required - but they often need to be quite large. For example, an improper rotation of about +40 degrees has to be applied to Tyr370 of 1dx4 for it to be possible to overlay the side chain closely onto the 1qon Tyr370 position.

5.1.6 Protein-Protein Clashes

By default, when a flexible side chain is moved during docking, GOLD checks whether any of its atoms clash with atoms in neighbouring residues. This gives rise to an extra **Protein Energy** term which contributes to the total fitness value.

The term is computed by summing the van der Waals interactions of all pairs of protein atoms which satisfy the following conditions: (a) at least one of the protein atoms is in a flexible side chain; (b) the van der Waals term for that pair of atoms is repulsive. The van der Waals interactions will be estimated using the same potential as is used for the protein-ligand vdW term (by default, this is a 4-8 potential).

The protein-protein clash term can be switched off by including the command `penalise_protein_clashes = 0` anywhere in a `rotamer_lib` block within the `gold.conf` file. For further instructions, refer to the GOLD configuration file documentation. Note that this will switch off calculation of the protein-protein clash term for all flexible side chains, not just the one corresponding to the `rotamer_lib` block in which you have placed the `penalise_protein_clashes = 0` command.

It is recommended that the protein-protein clash term be switched off in the following cases:

- When fixed conformers are used (i.e. Delta = 0 throughout): Fixed conformations will usually arise from having identified all such conformers in different crystal structures, and so will be comparable in energetics. You may therefore wish to treat each on equal merit. They may exhibit different protein-protein clash terms, however, and if this option is switched on, one may be significantly favoured over the other in a way you do not intend.

- When using improper torsions: These distort the protein structure in artefactual ways and will often introduce an artefactual protein-protein clash.

5.2 Large Backbone Movements

It is not possible for GOLD to make large backbone movements. This sort of problem can be dealt with by performing an ensemble docking (see Ensemble Docking).

Small backbone movements in the vicinity of a flexible side chain may be allowed by including the improper torsion angle CA-N-C-CA in a `rotamer_lib` command block (see Allowing a Localised Backbone Movement). Another option you can try is to apply a Localised Soft Potential to one or more residues in the loop (see Allowing for Localised Movements: Docking with Soft Potentials).

5.3 Ensemble Docking

5.3.1 Introduction

Sequential docking of individual ligands into a protein is computationally time-consuming.

Ensemble docking aims to address the issue of protein flexibility by adding multiple protein structures into a single GA run. The ultimate aim is to obtain higher enrichments in virtual screening experiments.

Multiple protein conformations can be searched concurrently when docking an ensemble, thus saving valuable time compared to a sequential docking approach.

Starting from a superimposed set of protein structures, GOLD evolves a separate population of individuals (representing ligand conformations) for each protein structure part of the ensemble.

The best ligand conformation found in any of the ensemble structures is returned, i.e. GOLD selects the best protein for a particular ligand based on the maximum fitness value of a ligand. For example, if a ligand gets the scores 10 in protein 1, 20 in protein 2 and 15 in protein 3, protein 2 will be selected.

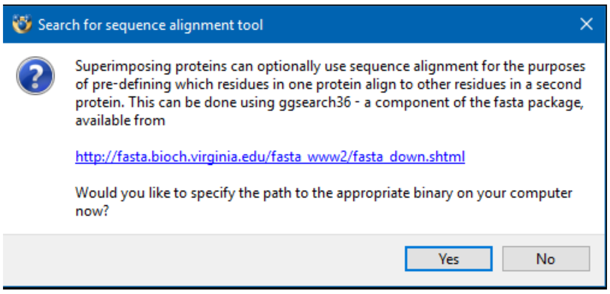
There should only be one binding site definition across the entire ensemble (hence the need to superimpose proteins) and this must be protein independent.

5.3.2 Setting up Proteins for Ensemble Docking

Proteins being specified in an ensemble should be set up in the usual way (see Setting Up the Protein(s)).

In addition, proteins that are to be used in an ensemble docking **must** be superimposed.

Proteins can be superimposed by using the **Superimpose Proteins...** button in the **Global Options** tab of the **GOLD Setup** window, or via **Calculate** and then **Superimpose Proteins** in Hermes. Brief details follow, complete details are provided in the Hermes documentation.



Superimposing proteins can optionally use sequence alignment for the purposes of pre-defining which residues in one protein align to other residues in a second protein. This can be done using ggsearch36 - a component of the fasta package, available from

http://fasta.bioch.virginia.edu/fasta_www2/fasta_down.shtml

Would you like to specify the path to the appropriate binary on your computer now?

A wizard is provided to facilitate protein superimposition. Proteins can be overlaid by matching residues based on label, matching residues based on sequence number or by matching residues based on sequence alignment.

Optionally, a component of fasta (called ggsearch36) can be used for sequence alignment of proteins to be superimposed. The package can be downloaded from http://fasta.bioch.virginia.edu/fasta_www2/fasta_down.shtml.

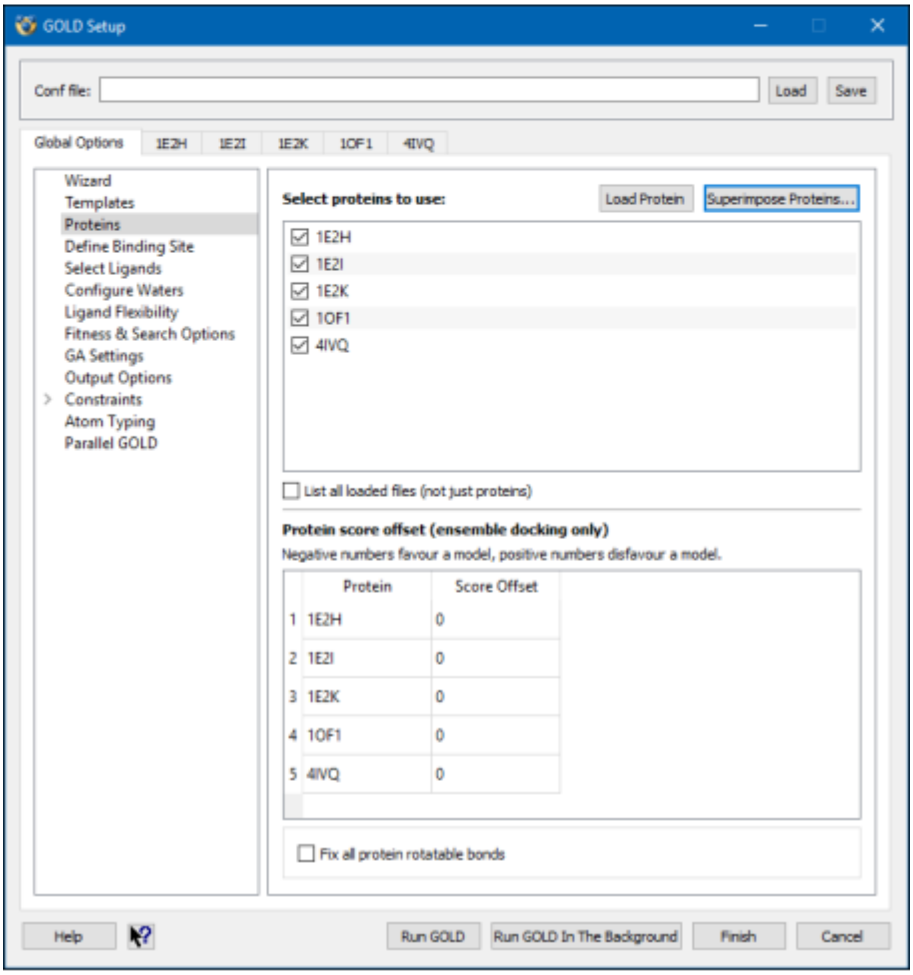
In both cases above the wizard guides you through the superimposition process.

5.3.3 Setting up an Ensemble Docking

All proteins currently loaded into Hermes are listed in the **Select proteins to use** section of **Global Options** tab in the **GOLD Setup** window and can be selected or deselected for use in the ensemble using their associated tick box.

Only proteins are listed in this dialogue. To view all other loaded files, activate the **List all loaded files (not just proteins)** tick box.

Each loaded protein has its own tab (adjacent to the **Global Options** tab) labelled with the name taken from the protein file, e.g. **1E2H** below.



Select proteins to use:

☑ 1E2H
☑ 1E2I
☑ 1E2K
☑ 1OF1
☑ 4IVQ

□ List all loaded files (not just proteins)

Protein score offset (ensemble docking only)

Negative numbers favour a model, positive numbers disfavour a model.

	Protein	Score Offset
1	1E2H	0
2	1E2I	0
3	1E2K	0
4	1OF1	0
5	4IVQ	0

□ Fix all protein rotatable bonds

Proteins must be set up in the usual way and superimposed before they are used in an ensemble docking (see Setting up Proteins for Ensemble Docking).

From within the **Proteins** option in **Global Options**, it is possible to apply a **Protein score offset**. This user-defined value will be subtracted from the overall fitness score if a ligand is docked into this protein structure. Both negative and positive values can be used; negative values favour the selection of a protein conformation, positive ones disfavor it. Thus, using the protein score offset, it is possible to bias which protein is selected. There are no limits for these values. If using this feature, these scores are reported as `DE (Protein)` in the GOLD log files.

When docking an ensemble, the binding site definition must describe the binding sites of all loaded and superimposed proteins. The binding site must therefore be defined using a method that isn't protein specific, i.e. a point (see Defining a Binding Site from a Point) or a reference ligand or cofactor (see Defining a Binding Site from a Reference Ligand). It is not possible to define the active site using an atom or a list of atoms or residues.

Ligands are specified for ensemble docking in the same way as when docking into an individual protein (see Specifying the Ligand File(s)). To achieve good results, ensembles of around 5 diverse protein conformers are recommended. Technically, up to 20 protein conformers can be used. The approximate memory requirements are 1GB per 10 protein conformers.

The following setup options can be applied across the entire ensemble:

- Specification of active waters (see Specifying Waters). Waters should not be considered to be associated with a specific protein, rather representative waters should be specified that will be used in all proteins.

- Similarity constraint (see Setting Up a Similarity Constraint).

- Scaffold constraint (see Setting Up Scaffold Match Constraints).

- Region constraint (see Setting Up Region (Hydrophobic) Constraints).

- Pharmacophore constraint (See Pharmacophore Constraints).

Settings specific to each protein are controlled from within the individual protein tabs, i.e.:

- Protein setup (addition of H atoms, extraction/deletion of water molecules, extraction of ligands).

- Specification of flexible side chains (see Side Chain Flexibility). Note that the active site definition must be set before flexible side chains can be setup.

- Special treatment of metal atoms (see Metal Ions).

- Distance constraint (see Setting Up a Distance Constraint).

- Substructure constraint (see Setting Up Substructure-Based Distance Constraints).
- HBond constraint (see Setting Up Hydrogen Bond Constraints).
- Protein HBond constraint (see Setting up Protein H Bond Constraints).
- Interaction motif (see Setting up an Interaction Motif Constraint).

Note: Protein-specific constraints are only evaluated if the respective protein structures are selected for scoring in the ensemble docking process. It is also possible to only specify a constraint for a single protein structure in the ensemble. There are caveats associated with definition of constraints when docking into an ensemble (see Caveats of Docking into Ensembles).

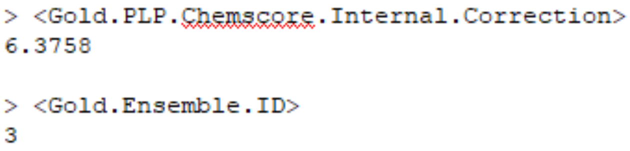
It is not possible to apply soft potentials or covalent constraints to a docking ensemble.

5.3.4 Interpreting Ensemble Docking Output

Standard docking output is detailed elsewhere (see Viewing and Analysing Results). The following details ensemble-specific output.

Each initialised protein is written to a file of the type: `gold_protein_<ensemble_index>.mol2`.

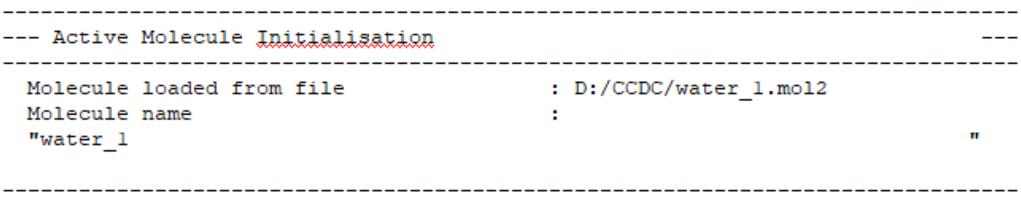
Each solution file will contain a `> <Gold.Ensemble.ID>` tag with the ensemble index, identifying the protein that GOLD has selected as the receptor for this solution.



```
> <Gold.PLP.Chemscore.Internal.Correction>
6.3758

> <Gold.Ensemble.ID>
3
```

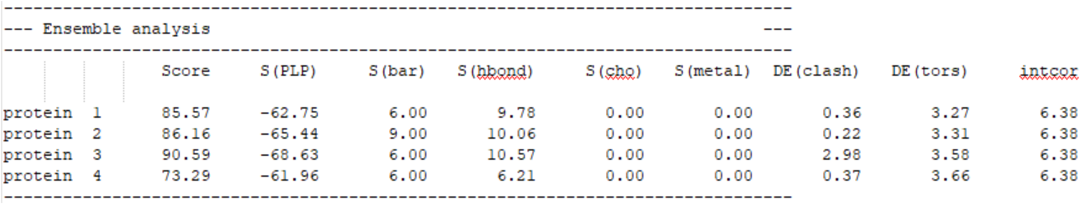
The `ligand.rnk` file and the `bestranking.lst` file have an additional `protein` column that details the ensemble index of the protein the ligand was docked into.

The `gold_protein.log` will contain a **Loading protein** section for each initialised protein (i.e. each protein in the ensemble) and there will be an **Active Molecule Initialisation** section for each of the initialised waters molecules in the ensemble.



```
--------------------------------------------------------------------------------
--- Active Molecule Initialisation                                           ---
--------------------------------------------------------------------------------
  Molecule loaded from file                   : D:/CCDC/water_1.mol2
  Molecule name                               :
  "water_1                                                                  "

--------------------------------------------------------------------------------
```

Protein scores for each ligand are contained with the ligand.log output file. You can see below that for this particular ligand, protein 3 scores highest.



```
--------------------------------------------------------------------------------
--- Ensemble analysis                                                        ---
--------------------------------------------------------------------------------
               Score      S(PLP)      S(bar)   S(hbond)      S(cho)    S(metal)  DE(clash)    DE(tors)      intcor

protein  1     85.57      -62.75        6.00       9.78        0.00        0.00       0.36        3.27        6.38
protein  2     86.16      -65.44        9.00      10.06        0.00        0.00       0.22        3.31        6.38
protein  3     90.59      -68.63        6.00      10.57        0.00        0.00       2.98        3.58        6.38
protein  4     73.29      -61.96        6.00       6.21        0.00        0.00       0.37        3.66        6.38
--------------------------------------------------------------------------------
```

5.3.5 Caveats of Docking into Ensembles

Although it is possible to specify rotatable side chains when docking an ensemble, these sorts of movements can be captured in an additional protein mode that can be added to the ensemble. This might be worth considering before setting any side chains as flexible.

Each protein is assigned an index number by GOLD when the ensemble docking is carried out. It is possible to rescore an ensemble docking however if a separate .conf file is used from the original docking, it is essential that the order the proteins are listed in is maintained in the rescore run. If the protein order is not retained, the rescore will not run.

In ensemble docking it is possible to define constraints on individual protein models or on all protein models. Constraints work by penalising poses that do not fulfil a specific constraint. As such, if a constraint is only set up on a subset of all models, the proteins without constraints could end up being favoured over those with the constraint. In addition, it is worth noting that docking the same ligand into different protein models can lead to differences in the scores. Thus, in order for the constraints to have a noticeable effect one might need to increase their weighting from the default values. **The combination of constraints and ensemble docking is not a straightforward problem; care should be taken in order to obtain results that are meaningful.**

5.4 Allowing for Localised Movements: Docking with Soft Potentials

GoldScore uses Lennard-Jones functional forms for both the External and Internal van der Waals contributions to the Fitness Function. By default, a 6-12 potential is applied to the Internal van der Waals contribution and a 4-8 potential is applied to the External van der Waals contribution. These defaults are defined in the `gold.params` file (see Altering GOLD Parameters: the gold.params File).

The 4-8 potential form for the External contribution is selected as being optimum for general use. However, there are cases where this potential form may be too severe in the short contact (i.e. the clash) component. This would arise, for instance, where part of the binding site is made up of a loop which it is known can move aside slightly to accommodate large ligands. In such cases it is possible to apply a softer 'Split Van der Waals Potential' for certain selected residues. Note that this is only applicable when using GoldScore.

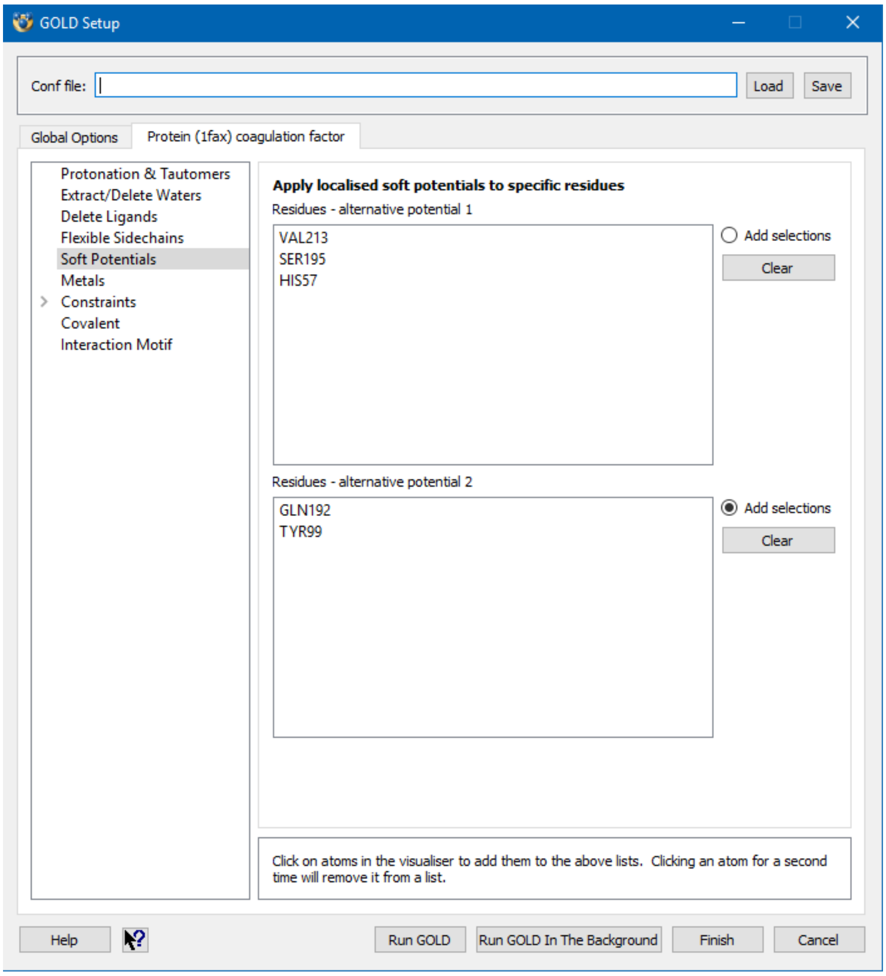
Two alternative soft 'Split Potential' forms are parameterised in the `gold.params` file:

```
EXTERNAL_POTENTIAL(1) = 4-8 2-4 - Potential 1
```

```
EXTERNAL_POTENTIAL(2) = 4-8 1-2 - Potential 2
```

The first term of each form describes long range interactions, the second term describes short range interactions. The point of change-over is at the 4-8 potential minimum and the second term is set such that both terms take the same value at this point. The function therefore remains continuous and the minimum point is the same as with the default 4-8 potential.

Soft potentials are protein-specific so to apply an alternative soft potential to specific residues you must first activate the protein tab (adjacent to the **Global Options** tab, e.g. **Protein (1fax) coagulation factor** in the example below) then click on **Soft Potentials** from the list of available options in the protein tab.



Apply localised soft potentials to specific residues

Residues - alternative potential 1

VAL213
SER195
HIS57

○ Add selections

Clear

Residues - alternative potential 2

GLN192
TYR99

● Add selections

Clear

Click on atoms in the visualiser to add them to the above lists. Clicking an atom for a second time will remove it from a list.

Only default residues' IDs are displayed here.

Select the alternative potential you wish to apply by switching on the corresponding **Add Selection** radio button, then specify those residues you want to apply the alternative potential to by clicking on them in the Hermes visualiser. Selected residues will be listed.

To remove a selected residue from the list, click on it again within the Hermes visualiser. To remove all selected residues, click on the **Clear** button.

More than one residue can be specified, and both alternative potential forms can be used in the same GOLD run as shown in the above example.

6 Setting Up Ligands

6.1 Essential Steps

Add all hydrogen atoms, including those necessary to define the correct ionisation and tautomeric states (see Ligand Hydrogen Atoms, Ionisation States and Tautomeric States).

Ensure that all bond types are correct. If they are, and hydrogen atoms have been placed on the correct atoms, GOLD will deduce atom types automatically when atom typing is turned on (see Automatically Setting Atom and Bond Types).

GOLD assigns atom types from the information about element types and bond orders in the input structure file, so it is important that these are correct. However, if for any reason, GOLD is unable to deduce an atom type, then the atom in question will be replaced with a dummy atom type Du. If this is the case a warning message will be given in the `gold_ligand.log` file.

The presence of dummy atoms should not significantly affect the docking prediction since dummy atoms are neither considered as donors nor acceptors.

There is usually a right and a wrong way to code groups which can be drawn in more than one way (i.e. have more than one canonical form), such as nitro, carboxylate and amidinium (see Atom and Bond Type Conventions for Difficult Groups).

The starting geometry of the ligand should be reasonably low in energy, since GOLD only samples torsion angles and will not alter bond lengths or angles, or rotate rigid bonds such as amide linkages, double bonds and certain bonds to trigonal nitrogens. Optimisation of the ligand using the CSD Conformer Generator is a good method to optimise your starting geometry.

Save the ligand as a `.mol2` file (i.e. Tripos format), a `.mmCIF` or a `.mol` file (i.e. MDL sd format). It is also possible (but not recommended) to use `.pdb` format. If using `.pdb` format, CONECT records should also be included (see Ligand File Formats).

6.2 Ligand Hydrogen Atoms, Ionisation States and Tautomeric States

GOLD uses an all-atom model, so the ligand must have all hydrogen atoms added. The precise geometrical positions of rotatable (e.g. hydroxyl and amino) hydrogen atoms do not matter, as they will be optimised during the GOLD run.

GOLD deduces hydrogen-bonding abilities from the presence or absence of hydrogen atoms. For example, you can control the protonation state of a carboxylic acid group by adding or removing the ionisable hydrogen atom. If incorrect ionisation or tautomeric states are inferred by the program, it is unlikely that correct protein-ligand binding modes will be predicted. If you are unsure about, e.g., the preferred ionisation state of the ligand, you should perform separate GOLD runs using the different possibilities.

GOLD ignores atom charges, both formal and partial. It deduces whether an atom is charged by counting the bond orders of the bonds that it forms and comparing the result with the atom's normal valency.

6.3 Ligand Geometry, Conformation and Stereochemistry

The ligand conformation will be varied by GOLD during docking. The starting conformation therefore does not matter. GOLD will not alter bond lengths or angles. These parameters should therefore be set to reasonably optimum values. A good practice is to build the ligand in an arbitrary conformation and then use CSD Conformer Generator to take it close to its local potential-energy minimum.

Non-fused ring conformations can be searched during docking using flipping of ring corners (see Flipping Ring Corners) and a library of ring templates (see Ring Conformations).

The torsion angles around rigid bonds such as amide linkages, double bonds and certain bonds to trigonal nitrogens will normally be fixed at their starting values. However, you can use the **Ligand Flexibility** option to enable some of these features to vary (see Ligand Flexibility).

GOLD will not alter stereochemistry. If you are unsure about the stereochemistry of the ligand, you must generate all alternatives and dock each separately. It is meaningful to make comparisons between fitness scores for dockings of different stereoisomers.

6.4 Ligand File Formats

Acceptable ligand file formats are mol2 (i.e. Tripos format), mol (i.e. MDL sd format), mmCIF, CCD and pdb (although we do not recommend the useof pdb format). Files in mol format may also have the extension `.mdl` or `.sdf`.

Only mol2 may be used if you wish to set ligand atom types manually (see Automatically Setting Atom and Bond Types).

An extension to the pdb file format is required if it is used for storing the ligand structure. Specifically, a bond specified twice in a single CONECT record is assumed to be a double bond, and a bond specified three times in a single CONECT record is assumed to be a triple bond. For example, the following CONECT records both specify a double bond between atoms with serial numbers 25 and 26:

```
CONECT 25 20 26 30 26
```

```
CONECT 26 25 27 52 25
```

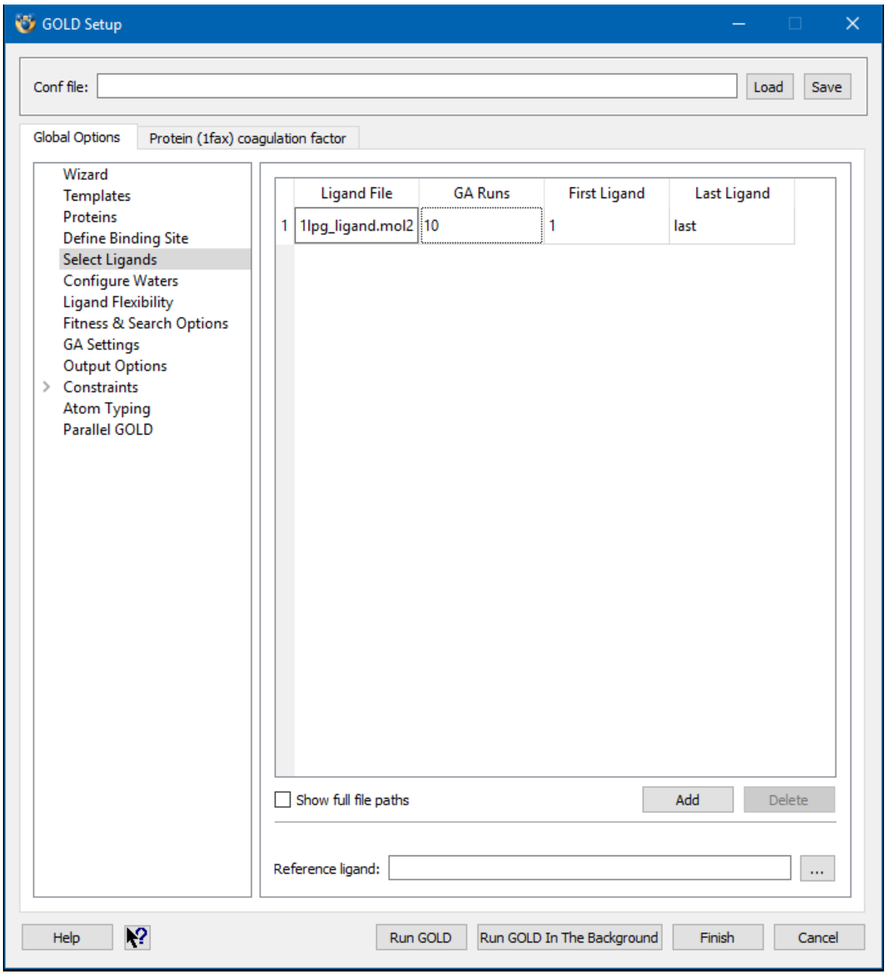
This mechanism for specifying bond orders is forced by the lack of a bond-order field in the standard pdb format and seems to offer lots of scope for users to commit errors. For that reason, we recommend that the pdb format is not used for ligands.

6.5 Specifying the Ligand File(s)

Any number of ligands can be specified, either by selecting several individual files, or by selecting a single file containing several ligands (i.e. a multi-mol2 or sd file). GOLD will dock each in turn.

Acceptable ligand file formats are mol2 (i.e. Tripos format), mol (i.e. MDL sd format), mmCIF and CCD (see Ligand File Formats).

Click on **Select Ligands** from the list of **Global Options** given on the left of the **GOLD Setup** window.



	Ligand File	GA Runs	First Ligand	Last Ligand
1	1lpg_ligand.mol2	10	1	last

□ Show full file paths

To specify a ligand file, click on the **Add** button and use the file selection window to choose the ligand data file(s).

Specify the number of times each ligand is to be docked by entering a value in the **GA runs** box (see Number of Dockings).

When using a single file containing several ligands (i.e. a multi-mol2 or sd file) it is possible to only dock specific ligands in that file. Specify which ligand you wish to start and finish docking at by entering, in the **First Ligand** and **Last Ligand** boxes, the numbers relating to the position of the ligands within the file. Unless specified otherwise, GOLD will, by default, start at the first ligand and finish at the last ligand in the file.

Repeat the above procedure if you want to select further ligands for docking.

To edit a specified ligand file (e.g. to change the number of times the ligand will be docked) highlight the ligand file with the mouse and make the required change.

To remove a specified file from those listed, highlight the ligand file with the mouse and hit the **Delete** button.

It is also possible to supply to GOLD with a file containing a reference ligand (e.g. a crystallographically observed ligand pose). The ligand reference file will be used to perform automated RMSD calculations against GOLD solution(s) (see Specifying a Ligand Reference File).

6.6 Setting Up Covalently Bound Ligands

GOLD is able to dock covalently bound inhibitors if you either specify which ligand atom is bonded to which protein atom or specify the reaction involved via a warhead-based workflow. GOLD supports three types of covalent link:

- A covalent link for use with individual ligands (see Setting Up a Single Covalent Link).
- A substructure-based covalent link for use with multiple ligands which have a common functional group (see Setting Up Substructure-Based Covalent Links).
- A warhead-based definition. Here, the ligand is scanned for known reactive warheads, and the protein and ligand are transformed together (see Setting up a Warhead-based Covalent Docking).

6.6.1 Method Used for Docking Covalently Bound Ligands

The program assumes that there is just one atom linking the ligand to the protein (e.g. the O in a serine residue). Both protein and ligand files are set up with the link atom included (so, if the serine O is the link atom, it will appear in both the protein and ligand input files). Ideally the link atom, in both the ligand and the protein, will have a free valence available through which the link can be made. If the link atom on the ligand does not have a free valence, having a hydrogen instead, then the docking will proceed, and the hydrogen will be ignored in terms of its contribution to the fitness score. It will however still be displayed when docking poses are visualised.

Inside the GOLD least-squares fitting routine, the link atom in the ligand will be forced to fit onto the link atom in the protein.

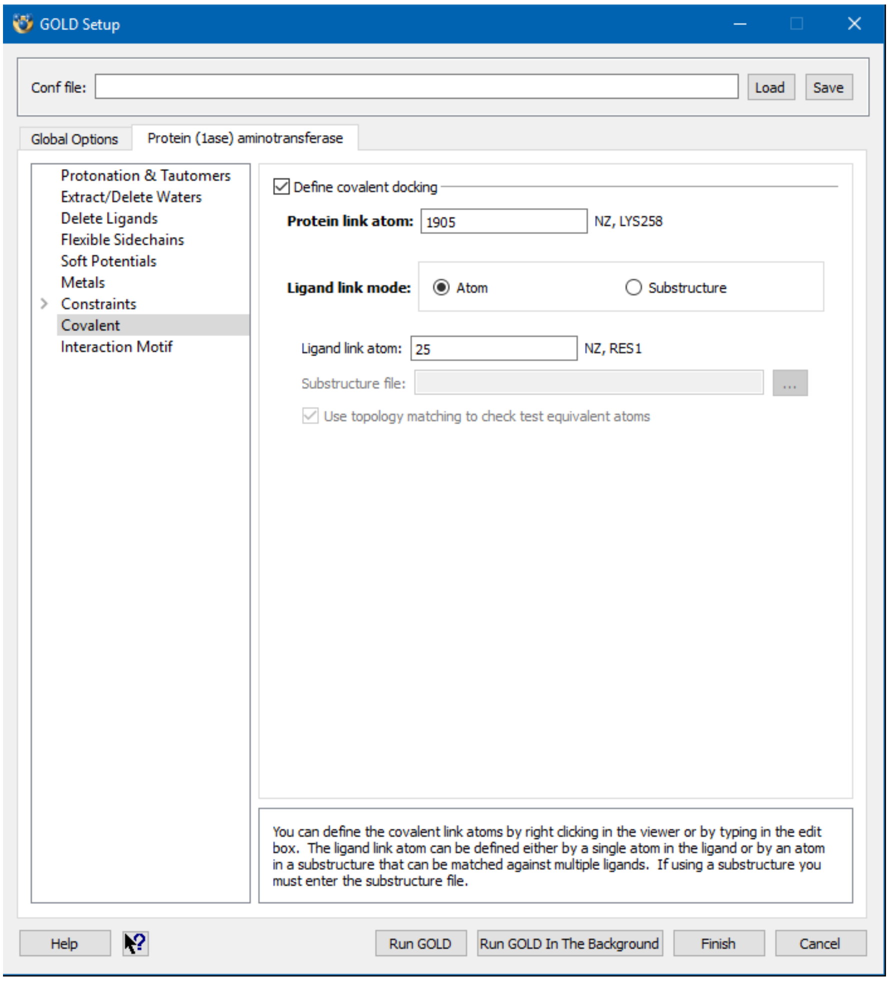
In order to make sure that the geometry of the bound ligand is correct, the angle-bending potential from the Tripos Force Field has been incorporated into the fitness function. On evaluating the score for the docked ligand, the angle-bending energy for the link atom is included in the calculation of the fitness score. The Tripos force-field is described in Validation of the General Purpose Tripos 5.2 Force Field, M. Clark, R.D. Cramer III & N. Van Opdenbosch, J. Comp. Chem., **10**, 982-1012, 1989.

This seems to work well in the systems on which GOLD was validated. However, since the protein is held rigid (apart from hydroxyl hydrogen atoms), it does require that the position of the link atom in the protein is sensible.

6.6.2 Setting Up a Single Covalent Link

Set up the protein and ligand structures so that they both contain the link atom (see Method Used for Docking Covalently Bound Ligands).

Covalent constraints are specific to the protein thus click on the protein tab (e.g. **Protein (1ase) aminotransferase** in the example below), select **Covalent** from the list of available options given on the left of the **GOLD Setup** window and enable the **Define covalent docking** check-box.



☑ Define covalent docking

Protein link atom: 1905 NZ, LYS258

Ligand link mode: ● Atom ○ Substructure

Ligand link atom: 25 NZ, RES1

Substructure file:

☑ Use topology matching to check test equivalent atoms

You can define the covalent link atoms by right clicking in the viewer or by typing in the edit box. The ligand link atom can be defined either by a single atom in the ligand or by an atom in a substructure that can be matched against multiple ligands. If using a substructure you must enter the substructure file.

Select **Atom** as the ligand link mode and define both the **Protein link atom** and **Ligand link atom**. This can be done by clicking on an atom in the visualiser. Alternatively, you can enter the atom number directly into the appropriate entry box.

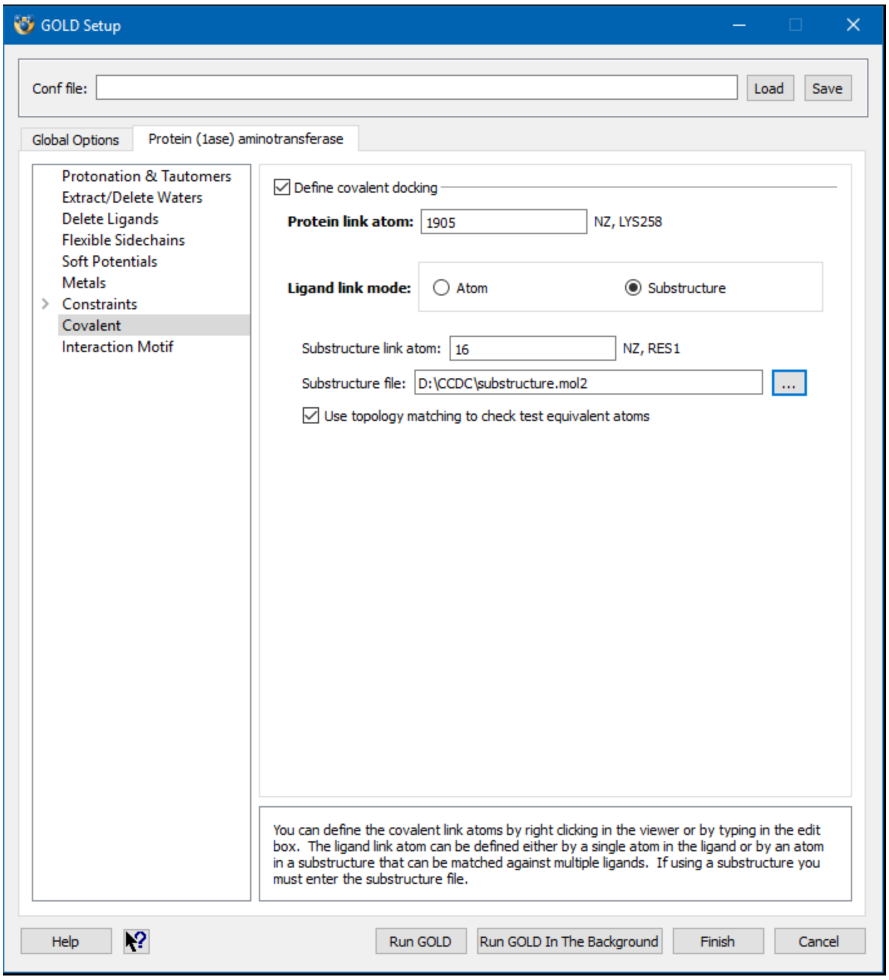
6.6.3 Setting Up Substructure-Based Covalent Links

It is possible to apply a covalent link to multiple ligands which have a common functional group. During docking the link will be applied to any ligands which contain a specified substructure (matching is performed on the basis of the atom types and 2D connectivity). Note: the substructure must be a sub-graph rather than a complete molecule.

To use a substructure-based covalent link, first create a file containing the substructure in mol2 format (e.g. `substructure.mol2`). It is recommended that you set atom types manually (see Manually Setting Atom and Bond Types) since an incomplete fragment can cause problems with automatic atom-typing. The actual conformation of the group in this file is not important, as only the atom types and 2D connectivity will be used.

Covalent constraints are specific to the protein thus click on the protein tab (e.g. **Protein (1ase) aminotransferase** in the example below), select **Covalent** from the list of available options given on the left of the **GOLD Setup** window and enable the **Define covalent docking** check-box.

Select **Substructure** as the ligand link mode.



Conf file:

Global Options | Protein (1ase) aminotransferase

☑ Define covalent docking

Protein link atom: 1905 NZ, LYS258

Ligand link mode: ○ Atom ● Substructure

Substructure link atom: 16 NZ, RES1

Substructure file: D:\CCDC\substructure.mol2

☑ Use topology matching to check test equivalent atoms

You can define the covalent link atoms by right clicking in the viewer or by typing in the edit box. The ligand link atom can be defined either by a single atom in the ligand or by an atom in a substructure that can be matched against multiple ligands. If using a substructure you must enter the substructure file.

To specify the **Substructure file** either enter the path and filename of the file or click on the **...** button and use the file selection window to choose the file.

Define both the **Protein link atom** and **Substructure link atom**. This can be done by clicking on an atom in the visualiser. Alternatively, you can enter the atom number directly into the appropriate entry box.

Enable the **Use topology matching to check test equivalent atoms** check-box if the constraint refers to a substructure atom (and therefore a ligand atom) which is topologically equivalent to other atoms (e.g. it is one of the oxygen atoms of an ionised carboxylate group); GOLD will then use whichever of the equivalent atoms gives the best result.

6.7 Setting up a Warhead-based Covalent Docking

Some covalent inhibitors undergo a reaction which is more complex than a single bond formation. In this context, a **warhead** is a functional group or moiety which reacts with a particular protein residue. Common examples are beta-lactam rings in antibiotics like Penicillin, or thiol groups in the antiplatelet medication Clopidogrel.

Using the warhead option in GOLD's covalent docking tab, you can automatically detect and transform warheads and receptors according to a reaction definition. As an example, the beta-lactam ring of a Penicillin molecule opens and binds to a cysteine residue. Using the warhead option, GOLD is able to detect the beta-lactam ring, and transform it as if it had reacted with the cysteine.

In practice, a covalent docking creates a single molecule from a reaction between a ligand and a protein. However, in GOLD, the two structures are kept separate. This makes reaction definitions much simpler, and allows clearer structure highlighting. We have chosen to make the arbitrary connecting point between the two structures the beta carbon of the residue. This means that any binding residue such as a cysteine or serine, is truncated to an alanine; the ligand and binding residue sidechain are transformed together. This means only a ligand transformation definition is required, not a protein transformation definition. The protein always undergoes the same change, while the ligand can undergo many different transformations, halving the work of defining reactions and generating conformers.

The ligand transformations can significantly alter the 3D structure. For this reason, the warhead option for covalent docking in GOLD requires a conformer generation step. Consider again the opening of a beta-lactam ring: the atom positions and torsion angles change completely. As a result, a licence for our conformer generator is required. If no such licence is present, the warhead option will show as locked; please reach out to your provider if you wish to upgrade.

6.7.1 List of Reactions

A complete list of the reactions which are natively available in GOLD is as follows. The reaction type is given, followed by the residues which may be involved. If you wish to add more reactions, please see Adding Reactions.

- Nucleophilic Substitution of Sulfonyl Fluoride: serine, threonine, tyrosine, lysine, cysteine, histidine
- Nucleophilic Substitution of Fluorosulfate: serine, threonine, tyrosine, lysine, cysteine, histidine

- Beta-Lactam Ring Opening: serine, cysteine
- Michael Addition to Alpha-Beta Unsaturated Carbonyl: cysteine
- Michael Addition to Vinylsulfone: cysteine
- Michael Addition to Vinylsulfonamide: cysteine
- Michael Addition to Ethyne: cysteine
- Addition to Nitrile: cysteine
- Nucleophilic Substitution to Chloromethylketone: cysteine
- Nucleophilic Substitution to Bromomethylketone: cysteine
- Addition to Aldehyde: cysteine
- Addition to Ketone: cysteine
- Disulfide Formation: cysteine
- Oxirane Ring Opening: cysteine
- Aziridine Ring Opening: cysteine
- Proton Pump Inhibitors: cysteine
- Double Condensation: Arginine

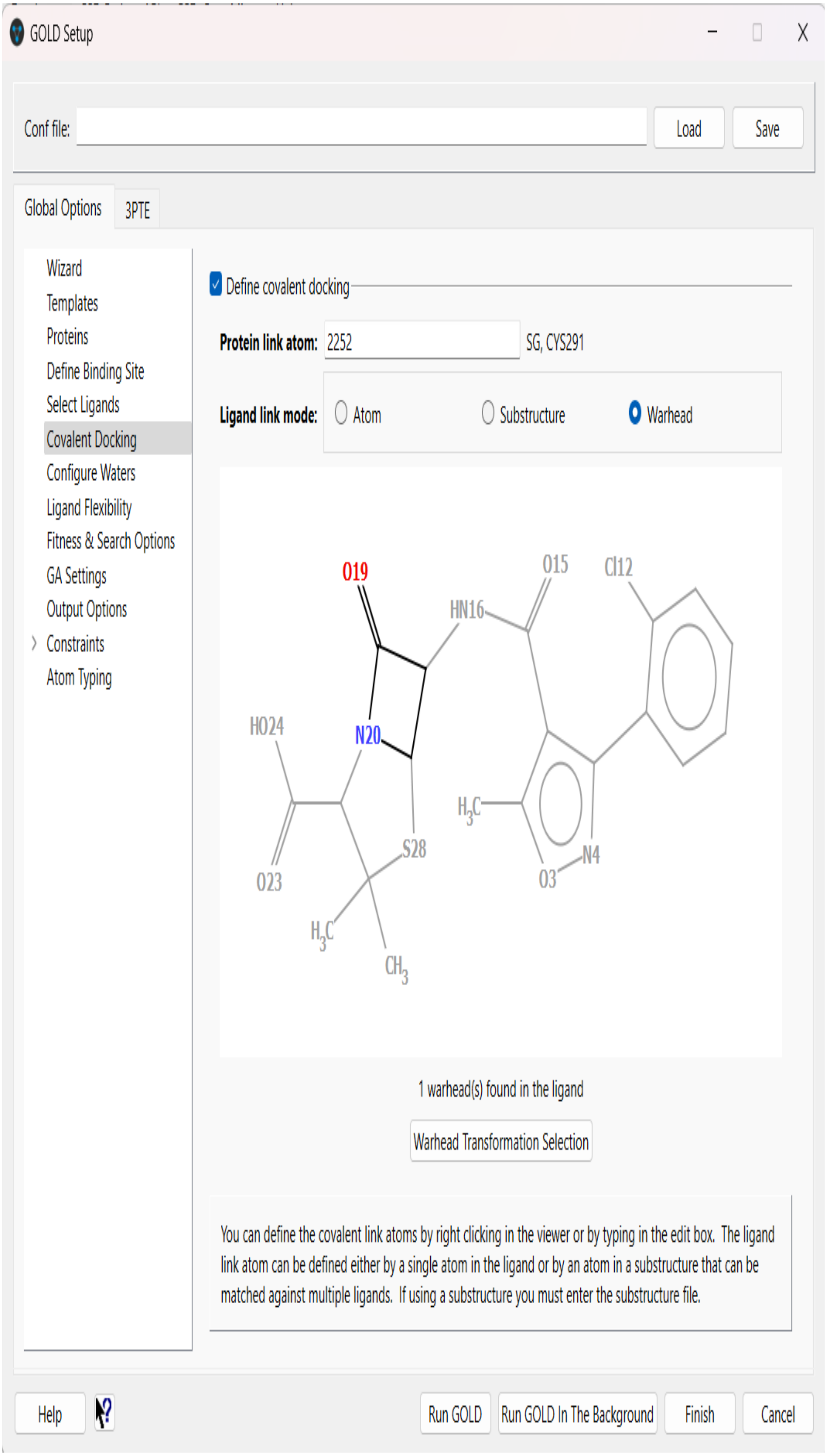
These warheads defined in a JSON file which contains the patterns to detect them, as well as their possible transformations when reacting with different residues. This file can be viewed in the installation location of Hermes, under CCDC/ccdc-software/hermes/covalent_docking/warheads.json.

6.7.2 Warhead Docking Requirements, Options and Execution

There are three key requirements for a warhead covalent docking: a single protein, a single ligand, and a connecting atom on the protein. This atom must belong to a residue which can covalently bind to your ligand in some way. Usually, this is an atom from a cysteine or serine residue. Make sure to input these three requirements before proceeding to the covalent tab in the wizard or setup window.

To perform a covalent docking with the warhead option, first input a binding atom on the protein; text will appear next to the input field indicating the atom and residue of interest, e.g. “SG, CYS291”. The radio button for warhead covalent docking can then be selected below. This will scan the input ligand for any “warheads”: reactive functional groups or moieties which may covalently bind to certain residues. This option requires that a protein link atom has been specified; any highlighted warheads will be based on the connecting residue. For example, a protein atom in a cysteine residue is selected, so warheads which react with cysteines are highlighted. If no warheads are found, please ensure you have selected the correct atom in the protein, and that the warhead is present in the warheads file. If your input ligand has already been through a warhead docking, it has already been transformed to its reacted state and will not show in the selection window.

If reacting warheads are found, they will be highlighted in a diagram of the ligand. Below this diagram is a button to select the desired warhead transformations: “Warhead Transformation Selection”. Clicking this will open a new selection window, for example:



☑ Define covalent docking

Protein link atom: 2252 SG, CYS291

Ligand link mode: ○ Atom ○ Substructure ● Warhead

1 warhead(s) found in the ligand

You can define the covalent link atoms by right clicking in the viewer or by typing in the edit box. The ligand link atom can be defined either by a single atom in the ligand or by an atom in a substructure that can be matched against multiple ligands. If using a substructure you must enter the substructure file.

Warhead Name: This is the warhead name or reaction class, as defined in the warheads.json file. Examples include “Addition to Ketone” or “Beta-Lactam Ring Opening”. This broadly describes how the warhead will react with the residue selected.

Match: In the rare circumstance where a ligand contains two (or more) warheads of the same type, they will each be given a number.

Warhead Structure: This is the warhead match, as found in the ligand. The warhead substructure is highlighted with some surrounding atoms presented for context.

Transformed Subtype: If a warhead can transform in multiple different ways (commonly, different stereoisomers), this is indicated here. This will usually be an R or S label, more rarely an E or Z. In the unlikely event that two chiral centres are created by the transformation, the labels are numbered, e.g. S1.

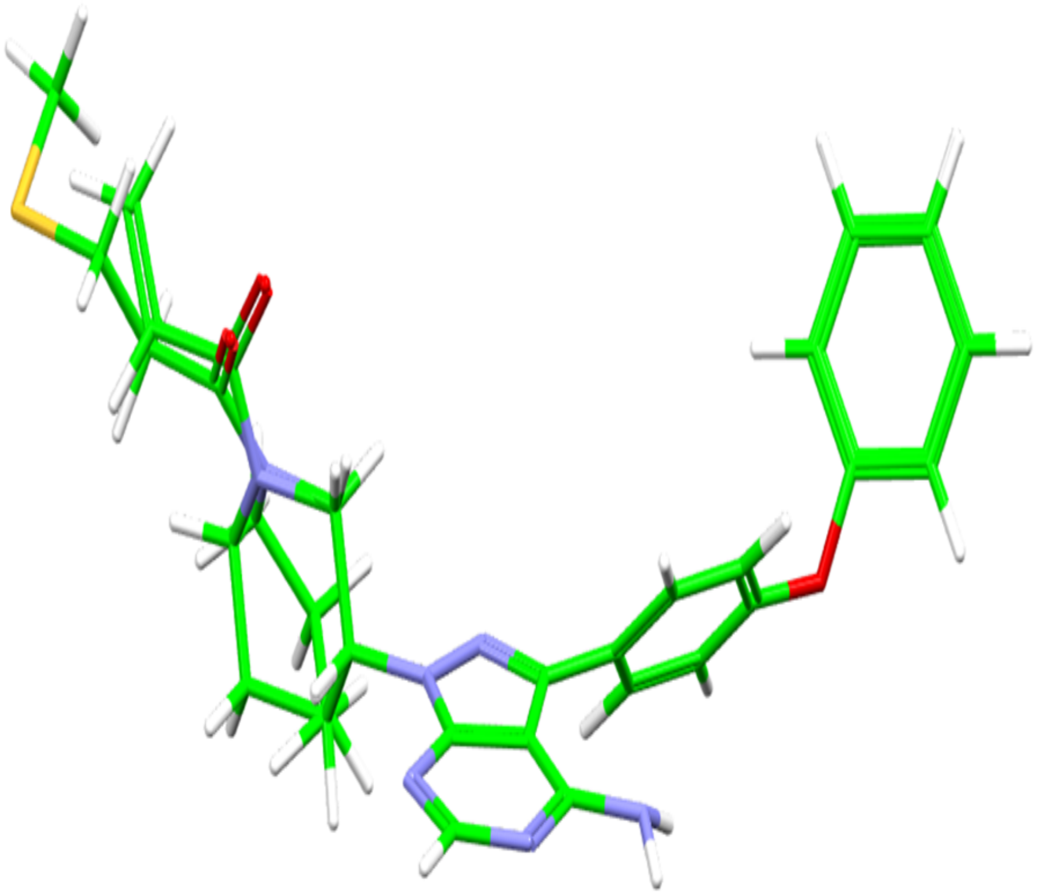
Transformed Structure: This is the outcome of the warhead being transformed with a particular residue. The transformed substructure is highlighted and matches the highlighted atoms in the warhead structure column. Some surrounding atoms are shown for context.

Select: This is where the desired warhead transformation(s) can be selected.

Upon selecting the desired warhead transformation(s) and confirming with “OK”, the text below the ligand diagram will show how many transformations have been chosen. Any other docking settings can then be altered in the other tabs before running the docking. Each selection made is treated as a separate docking with separate results. So, if producing 10 results is your default for a docking, each warhead reaction selection will produce 10 scored ligands.

New ligand and protein files will be created. The format of these files is `<abbreviated reaction name><match number>_<transformed subtype>.<file_extension>`, for example: `ABUnsat0_R.mol` is an Alpha-Beta Unsaturated Carbonyl warhead reaction, the zeroth match, with the R optical isomer.

An example of a transformed and untransformed structure are shown below, overlaid. This is ibrutinib, as it would be if bound to a cysteine residue (left).



The `gold.conf` file will also be updated to contain the edited protein and ligand files. This means that if running the docking independently of the setup—or simply re-running a docking—no further editing is required. In the GUI, this means you would select a single-atom covalent docking, rather than repeating a warhead covalent docking.

6.7.3 Adding Reactions

If you are interested in a covalent docking reaction that is not present in GOLD, it is possible to add your own reactions. Where Hermes is installed, there is a JSON data file which defines all the reactions: `covalent_docking/warheads.json`. It is recommended to make a backup of the file before changing or adding to it.

The structure of this file is as follows:

```
“<find pattern>”: {
 “name”: “<name of reaction class or warhead>”,
 “abbrev”: “<abbreviated name, used for naming results files>”,
 “receptors”: {
  “<residue 1>”: “<replace pattern>”,
  “<residue 2>”: {
   “R”: “<replace pattern>”,
   “S”: “<replace pattern>”
  },
  ...
 }
},
...
```

The "find" pattern is the SMARTS pattern of a particular warhead to be found in a ligand. For example, the pattern `[N:1]#[C:2][C:3]` would match to a nitrile group connected to an aliphatic carbon.

The "replace" pattern is a SMARTS pattern which will replace the find pattern in the ligand. This pattern represents the warhead after it has reacted with the protein and is in the bound state. The numbering on these patterns indicates which atoms map to which. Again using the nitrile example, the replace pattern `[C:3343]S[C:2](=[NH:1])[C:3]` shows that the nitrile has reacted with a cysteine to form an imine. Any atoms which are not numbered in the find pattern will be removed, any atoms which are not numbered in the replace pattern will be added.

The carbon atom numbered 3343 is the atom which will bond to the alanine-like truncated residue in the protein. No transformation definition is needed for the protein as it is transformed in the same way every time.

Any stereoisometric variants to the resulting ligand i.e. diastereoisomers or enantiomers can be specified for each residue. Commonly, cysteine residues give rise to chiral carbons, and need to be described. If only one isomer is of interest, the other transformation need not be defined.

6.7.4 Potential Pitfalls with Warhead Covalent Docking

While we have tried to make the process of warhead transformation simple and easy, it is by nature a complex problem. As a result, there are some complications worth bearing in mind when working with these inhibitors.

6.7.4.1 Comparing to Reference Ligands

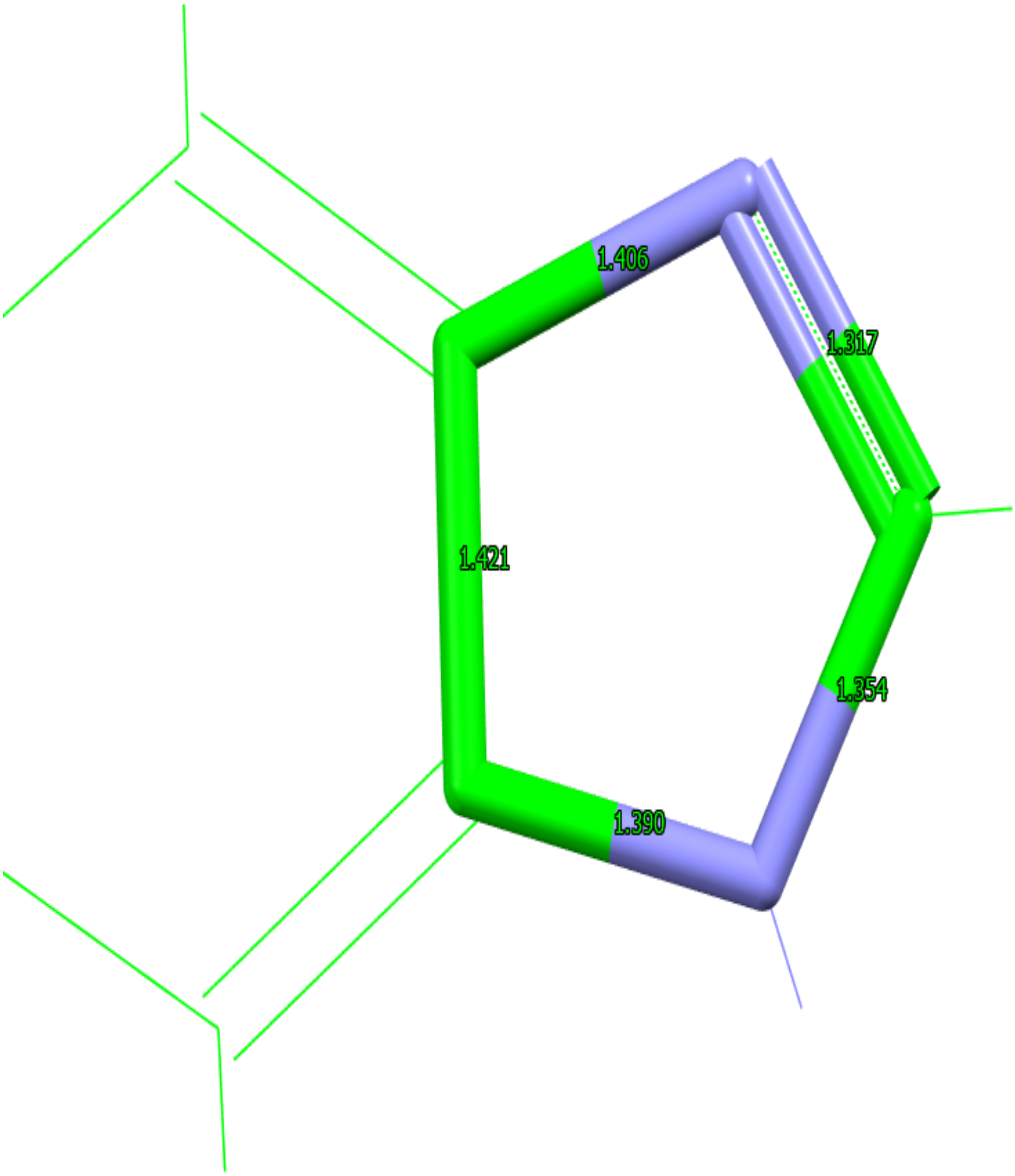
Be careful when comparing to a reference ligand. Because the atoms of the ligand have changed during the transformation process, RMSD values will likely be affected. We recommend comparing to reference ligands visually only.

6.7.4.2 Defining New Reactions

When defining new reactions in the warheads file, SMARTS patterns are used. SMARTS patterns can be more general than makes sense in this context. For example, a transformation could contain a wildcard (any) atom. This would not make sense for a transformation and should not be used. If you're unsure about acceptable patterns, the file itself has many examples.

It is common when creating ligand files to store aromatic rings as a series of single and double bonds. Unfortunately, this can have the effect of "incorrectly" matching a SMARTS pattern. For example, benzaldehyde could match as an unsaturated carbonyl, if the benzene ring is treated as a series of single and double bonds. It is therefore important to prepare ligand files such that aromatic rings are correctly labelled. We have tried to write SMARTS patterns which accommodate this issue, but it is a balance between making patterns open enough to not miss important reactions, but restrictive enough to not match excessively. We recommend aromatic rings are labelled as such in the input file, rather than using the kekulised form.

Aromaticity for fused five-membered rings containing nitrogen can be ill-defined. Even in the simple case of purine-like substructures, electron distributions can be considered aromatic or aliphatic in different contexts, as is often indicated by the bond lengths within the rings. If adding a new reaction with such a ring, be careful in the reaction definition to account for this. If in doubt, we recommend using the more general pattern `[#7]~[#6]` as opposed to `[n][c]`. This may also ameliorate certain issues with tautomerisation.



6.7.4.3 Disorder

If the protein supplied for warhead covalent docking is disordered, there may be ambiguity around which atoms to remove. For this reason, it is recommended to select the desired occupancies as part of protein preparation. This is only necessary if the binding residue is disordered.

6.8 Specifying a Ligand Reference File

It is possible to supply to GOLD with a file containing a reference ligand (e.g. a crystallographically observed ligand pose).

Please note that this is feature is really designed to facilitate analysis of self-docking experiments, i.e. where a crystallographic ligand is docked back into it's protein and the result compared with the experimentally-determined pose. This implies that the input ligand and the reference ligand be the same. While the facility can work in cases where input ligands are similar but not identical to the reference this can also cause problems. At best the RMS values can be and at worst GOLD can crash. The use of this feature other than for the case of self-docking should thus now be considered to be deprecated.

The ligand reference file will be used to perform automated RMSD calculations against GOLD solution(s). For each GOLD solution the resultant RMSD with respect to the reference ligand will be written to the files containing the fitness function rankings, i.e. the ligand rank file (`.rnk`) and `bestranking.lst` file.

Click on **Select Ligands** from the list of **Global Options** given on the left of the **GOLD Setup** window. To specify the ligand reference file, either enter the path and filename of the file in the **Reference ligand** box, or click on the **...** button and use the file selection window to choose the file.

7 Atom and Bond Types

7.1 Atom and Bond Type Overview

Each protein and ligand atom must be assigned an atom type which is used, for example, to determine whether the atom is capable of forming hydrogen bonds.

GOLD atom typing is based on SYBYL atom types. Internally, GOLD also uses some additional atom types (see Internal GOLD Atom Types).

SYBYL bond types are also used.

Correct assignment of atom and bond types is crucial.

GOLD assigns atom types from the information about element types and bond orders in the input structure file, so it is important that these are correct. However, if for any reason, GOLD is unable to deduce an atom type, then the atom in question will be replaced with a dummy atom type Du. If this is the case a warning message will be given in the `gold_protein.log` file.

The presence of dummy atoms should not significantly affect the docking prediction since dummy atoms are neither considered as donors nor acceptors

Dummy atoms may be visualised in Hermes by activating the **Show unknown atoms** tickbox in the Visualisation Options toolbar in Hermes.

Atom types may be set manually, provided you are using mol2 input files (see Manually Setting Atom and Bond Types).

Alternatively, they can be set automatically (see Automatically Setting Atom and Bond Types). Unless you are an expert GOLD user or are dealing with a very unusual ligand structure, you are recommended to use this option. However, you still need to input the ligand and protein structures correctly, i.e. with correct bond orders and appropriate protonation states.

7.2 Automatically Setting Atom and Bond Types

Unless you are an expert GOLD user or are dealing with a very unusual ligand structure, you are recommended to use the automatic atom-type assigner.

To automatically set atom and bond types click on **Atom Typing** from the list of **Global Options** given on the left of the **GOLD Setup** window. Atom and bond types can then be assigned automatically for the ligand and/or protein by switching on the appropriate check-box(es).

GOLD assigns atom types from the information about element types and bond orders in the input structure file, so it is important that these are correct (see Atom and Bond Type Conventions for Difficult Groups). However, if for any reason, GOLD is unable to deduce an atom type, then the atom in question will be replaced with a dummy atom type Du.

It does not matter whether the bonds in an aromatic ring are coded as aromatic (ar) or alternate single and double, as the GOLD atom-type assigner will automatically assign the special SYBYL bond type ar where appropriate.

The atom-type assigner will also detect amide linkages and assign them the SYBYL bond type am.

Care should be taken when using type-assignment on protein input files. In particular, the software is likely to be unreliable if protein residues have been partially deleted so that some atoms appear to have free valencies. This situation can be avoided by ensuring that all residues included in the input file are complete.

There is usually a right and a wrong way to code groups which can be drawn in more than one way (i.e. have more than one canonical form), such as nitro, carboxylate and amidinium. A list of correct bond types for some of the common, difficult groups is available (see Atom and Bond Type Conventions for Difficult Groups).

Because correct atom typing is so important, any messages from the type checker are logged in both the `gold_protein.log` file and the `gold.err` file. These errors will also be displayed in a separate window if GOLD is run through the front end.

7.3 Manually Setting Atom and Bond Types

If you do not want to use the automatic atom- and bond-type assignment available in GOLD, you can define the atom and bond types yourself, provided that you use mol2 format. This option is useful when you want to set unusual atom types or user-defined types.

GOLD atom typing is based on SYBYL atom types (see Appendix B: List of Atom and Bond Types).

SYBYL bond types are also used (see Appendix B: List of Atom and Bond Types).

Even if atom types are set manually, the automatic atom-type assignment software is still run to check the ligand structure for inconsistencies. Any errors will be recorded in both the log file and the error file. In most cases, input types will not be reset.

If for any reason GOLD is unable to deduce an atom type, then the atom in question will be replaced with a dummy atom type Du.

Bond types must be correctly set (see Atom and Bond Type Conventions for Difficult Groups). This is normally just a case of checking single and double bonds. However, the amide bond must be set to the am bond type. Also, the ar bond type is used for delocalised bonds (e.g. in carboxylate, phosphate and guanidinium ions) as well as for aromatic bonds.

Atom types should conform to those expected in SYBYL. In particular, sp2 oxygen is atom type O.2, sp3 oxygen is O.3, tetrahedral nitrogen is N.3 (or N.4 if protonated), planar (non-amide) nitrogen is N.pl3 and the planar amide nitrogen is N.am. The atom type O.co2 should be used for the oxygens of carboxylate and phosphate ions or the singly-charged oxygen of phenolates.

If an atom is mis-typed, it is possible that GOLD will assign it the wrong H-bond donor or acceptor properties. Therefore, correct atom-type assignment is crucial. An N.3 donor (tetrahedral nitrogen), is very different from an N.4 (protonated nitrogen) or an N.pl3 (planar trigonal nitrogen) donor. The assignment of rotatable bonds may also be affected. If a bond has the wrong type, it may be inappropriately allowed to rotate freely.

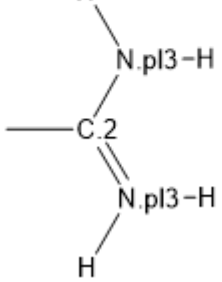
A list of atom and bond type conventions for some common, difficult groups is available (see Atom and Bond Type Conventions for Difficult Groups).

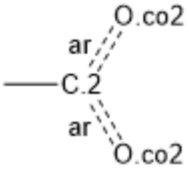
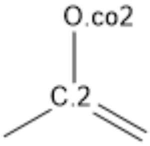
7.4 Atom and Bond Type Conventions for Difficult Groups

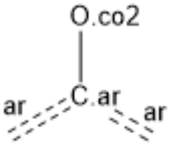
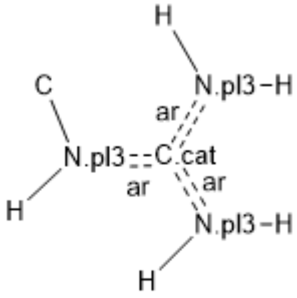
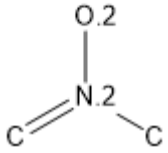
Use of correct atom and bond types in GOLD is important for producing good results.

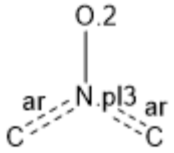
In order for the GOLD atom-type assigner to work correctly, it is necessary for the input structures to have correct bond orders. This can be difficult when a ligand contains a group that can be drawn in more than one way (i.e. a group which has more than one canonical form). In such cases, there is usually a right and a wrong way for GOLD, and you need to know which is the right one to use.

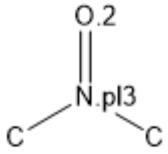
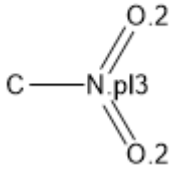
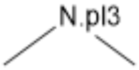
The following table explains how to set the bond orders of some common difficult groups. It also shows the atom types that GOLD will assign if bond types are set correctly (or that you must assign if you are setting atom types manually).

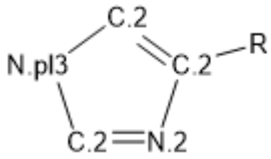
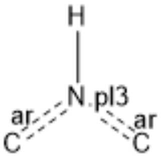
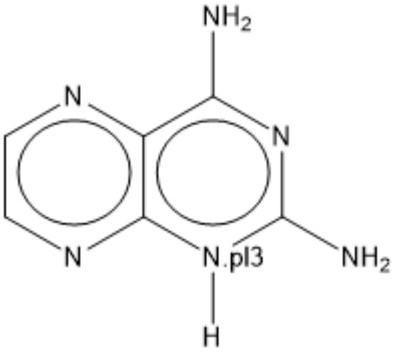
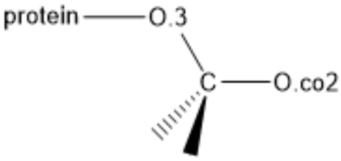
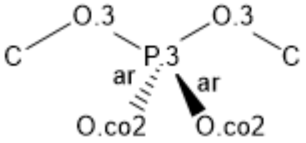
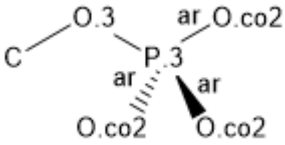
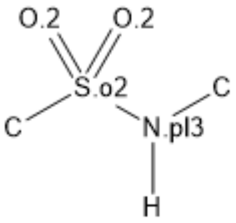
Functional Group	2D Diagram	Notes
Amidinium		
Carboxylate		

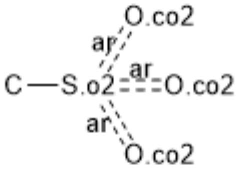
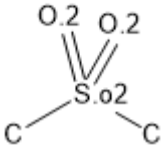
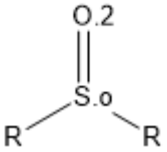
<table>
<tr><th>Functional Group</th><th>2D Diagram</th><th>Notes</th></tr>
<tr><td></td><td>
</td><td></td></tr>
<tr><td>Enolate/ phenolate oxygen</td><td>


</td><td></td></tr>
<tr><td>Guanidinium</td><td>
</td><td></td></tr>
<tr><td>N-oxide</td><td>




</td><td></td></tr>
<tr><td>Nitro</td><td>
</td><td></td></tr>
<tr><td>Nitrogen (anionic)</td><td>
</td><td></td></tr>
</table>

Functional Group	**2D Diagram**	**Notes**
		For example, an anionic imidazole ring would be: 
Nitrogen (cationic, aromatic)		For example, the pteridine ring system in methotrexate (PDB code 4DFR) would be: 
Oxygen (anionic)		A serine protease transition-state analogue example is shown
Phosphate (bridging)		
Phosphate (terminal)	 O.3 ar O.co2 C P.3 ar ar O.co2 O.co2	
Sulfonamide		GOLD will treat the nitrogen atom as a planar, trigonal nitrogen, i.e. not capable of accepting a hydrogen bond. However, pyramidal sulfonamide nitrogen atoms are now typed as N.3, if the

Functional Group	2D Diagram	Notes
		geometry read into GOLD is pyramidal rather than N.pl3, and are treated as H-bond acceptors (i.e. they have a fitting point) allowing them to coordinate metal groups.
Sulfonate		
Sulfone		
Sulfoxide (sulfinyl)		

7.5 Internal GOLD Atom Types

GOLD uses four internal atom types which are not recognised by SYBYL. These are N.plc (nitrogen donors in a protonated delocalised system, such as a guanidinium ion), N.acid (acidic nitrogen, e.g. in tetrazole or sulfonamide ions), S.a (sulfur acceptors) and S.m (charged sulfur atoms). You should not really need to know about these, but all assignments of the N.plc, N.acid, S.a and S.m atom types are logged in the `gold.log` file, so you can check to see if everything is working as you would expect.

8 Fitness Functions

8.1 Selecting a Fitness Function

GOLD offers a choice of fitness functions: GoldScore, ChemScore, Astex Statistical Potential (ASP)) and Piecewise Linear Potential (ChemPLP)).

ChemPLP has been found to give the highest average success rates for both pose prediction and virtual screening experiments against diverse validation test sets and is therefore the default scoring function in GOLD.

With respect to use of the GoldScore, ChemScore and ASP scoring functions, they are about equally reliable although, on any given problem, one may give a good prediction and the other not. Therefore, when screening large numbers of compounds, rescoring docking poses with alternative scoring functions and considering the best results from each can have a favourable impact on the overall rank ordering of ligands (see Rescoring).

To select a scoring function, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select the required scoring function from the drop-down menus for docking and/or rescoring.

8.2 Piecewise Linear Potential (ChemPLP)

8.2.1 Overview

For a more detailed description of the PLP and ChemPLP fitness functions as well as the derivation of their parameters, please see Empirical Scoring Functions for Advanced Protein-Ligand Docking with PLANTS (see References).

PLP and ChemPLP are empirical fitness functions optimised for pose prediction. ChemPLP is the default scoring function in GOLD.

In both cases, the Piecewise Linear Potential (f_{PLP}) is used to model the steric complementarity between protein and ligand, while for ChemPLP additionally the distance- and angle-dependent hydrogen and metal bonding terms from ChemScore are considered ($f_{chem-hb}$, $f_{chem-cho}$, $f_{chem-met}$).

The internal score of the ligand consists of the heavy-atom clash potential ($f_{lig-clash}$) (see References) as well as the torsional potential used within ChemScore ($f_{lig-tors}$).

Both fitness functions are capable of covalent docking ($f_{chem-cov}$), considering flexible side chains ($f_{chem-prot}$) and explicit water molecules as well as handling constraints (f_{cons}).

Parameters for both fitness functions can be altered by changing the files plp.params and chemplp.params for PLP and ChemPLP, respectively (see Altering PLP Fitness-Function parameters).

ChemPLP parameters are used by default as they show on average an improved performance in pose prediction and virtual screening applications.

$$\mathrm{fitness}_{PLP} = -(w_{PLP} \cdot f_{PLP} + W_{lig\text{-}clash} \cdot f_{lig\text{-}clash} + w_{lig\text{-}tors} \cdot f_{lig\text{-}tors} + f_{chem\text{-}cov} + w_{prot} \cdot f_{chem\text{-}prot} + w_{cons} \cdot f_{cons})$$

$$\mathrm{fitness}_{ChemPLP} = \mathrm{fitness}_{PLP} -(f_{chem\text{-}hb} + f_{chem\text{-}cho} + f_{chem\text{-}met})$$

8.2.2 PLP Interaction Types

The Piecewise Linear Potential (PLP) models the attraction as well as repulsion of protein and ligand heavy atoms. In the left-hand figure below the partially attractive potential using 6 parameters A to F is presented, and in the right-hand figure below the purely repulsive potential using 4 parameters A to D is presented.

All protein and ligand heavy atoms are typed as donor, acceptor, donor/acceptor or nonpolar. Additionally, metal ions in the protein binding site are assigned the metal type.

Depending on protein and ligand atom type, the appropriate potential from the table below is selected. Each potential H-bond, metal, buried and nonpolar is defined by a specific setting of parameters A to F. The same accounts for the repulsive potential, in which case parameters A to D are specified. See `plp.params` and `chemplp.params` for the default parameters used.

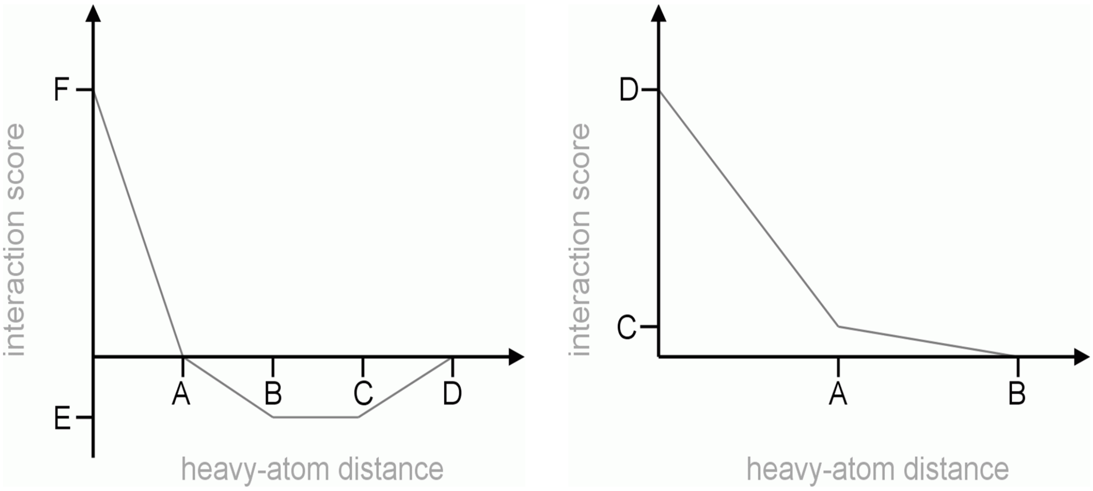


Figure: Piecewise Linear Potential. (left) Partially attractive potential, (right) Repulsive potential.

PLP interaction types selected depend on the protein and ligand atom type.

	Protein atom type				
Ligand atom type	Donor	Acceptor	Don./ Acc.	Nonpolar	Metal
Donor	Repulsive	H-bond	H-bond	Buried	Repulsive
Acceptor	H-bond	Repulsive	H-bond	Buried	Metal
Don./Acc.	H-bond	H-bond	H-bond	Buried	Metal
Nonpolar	Buried	Buried	Buried	Nonpolar	Buried

8.2.3 Altering PLP Fitness-Function parameters

The PLP and ChemPLP parameter files are stored in the `<Installation folder>/ccdc_software/gold/GOLD/gold/` directory. They contain all the parameters used by the GOLD implementation of PLP. A full description of the meaning of the PLP specific parameters are given below.

The PLP and ChemPLP files can be customised by copying them, editing the copy, and instructing GOLD to use the edited file.

To use a modified `plp.params` or `chemplp.params` file, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select ChemPLP from the **Scoring Function** drop-down menu. Then, either enter the path and filename of the **Scoring function parameter file** or click on the **...** button and use the file selection window to choose the file.

The format of the PLP and ChemPLP file is quite strict: incorrect editing may cause GOLD to behave in unexpected ways.

The following commands are recognised in PLP parameter files. For the default values used in PLP and ChemPLP, see the `plp.params` and `chemplp.params` parameter files, respectively.

Parameters used for PLP and ChemPLP (plp.params and chemplp.params)	
PLP_COEFFICIENT Weight of PLP contributions (W_{PLP})	<float value>

Parameters used for PLP and ChemPLP (plp.params and chemplp.params)	
PLP_LIGAND_CLASH_COEFFICIENT Weight of ligand clash potential ($W_{\text{lig-clash}}$)	<float value>
PLP_LIGAND_TORSION_COEFFICIENT Weight of ligand torsion potential ($W_{\text{lig-tors}}$)	<float value>
PLP_PROTEIN_ENERGY_COEFFICIENT Weight of ChemScore protein potential (W_{prot})	<float value>
PLP_CONSTRAINT_COEFFICIENT Weight of constraint contributions (W_{cons})	<float value>
PLP_GRID_SPACING Grid spacing used for PLP map	<float value>
PLP_HBOND_METAL_FUNCTION Additional hydrogen and metal bonding contributions. Use CHEMSCORE to activate ChemScore hydrogen and metal bonding contributions. If NONE is specified, only the PLP contributions will be considered (see the plp.params file)	<CHEMSCORE | NONE >
PLP_WATER_BARRIER Penalty value added for each explicit water molecule activated by the search algorithm (positive value)	<float value>
HBOND_A	<float value>
HBOND_B	<float value>
HBOND_C	<float value>

Parameters used for PLP and ChemPLP (plp.params and chemplp.params)	
HBOND_D	<float value>
HBOND_E	<float value>
HBOND_F Hydrogen bonding parameters for potential H-Bond (A to D are distance parameters, E and F are interaction scores)	<float value>
BURIED_A	<float value>
BURIED_B	<float value>
BURIED_C	<float value>
BURIED_D	<float value>
BURIED_E	<float value>
BURIED_F Parameters used for potential buried (A to D are distance parameters, E and F are interaction scores)	<float value>
METAL_A	<float value>
METAL_B	<float value>
METAL_C	<float value>
METAL_D	<float value>
METAL_E	<float value>
METAL_F Metal bonding parameters for potential metal (A to D are distance parameters, E and F are interaction scores)	<float value>
NONPOLAR_A	<float value>
NONPOLAR_B	<float value>
NONPOLAR_C	<float value>
NONPOLAR_D	<float value>
NONPOLAR_E	<float value>
NONPOLAR_F	<float value>

Parameters used for PLP and ChemPLP (plp.params and chemplp.params)	
Parameters for potential nonpolar (A to D are distance parameters, E and F are interaction scores)	
REPULSIVE_A	<float value>
REPULSIVE_B	<float value>
REPULSIVE_C	<float value>
REPULSIVE_D	
Parameters for potential repulsive (A and B are distance parameters, C and D are interaction scores)	<float value>
LINK_BEND_COEFFICIENT (see ChemScore)	<float value>

Parameters for additional ChemScore contributions used in ChemPLP (chemplp.params)	
HBOND_COEFFICIENT (see ChemScore)	<float value>
METAL_COEFFICIENT (see ChemScore)	<float value>
CHARGED_HBOND_FACTOR Scaling factor for charged hydrogen bonds (expected to be greater than or equal to one)	<float value>
CHARGED_METAL_FACTOR Scaling factor for charged acceptors coordinating to a metal ion (expected to be greater than or equal to one)	<float value>
	<float value>

Parameters for additional ChemScore contributions used in ChemPLP (chemplp.params)	
DELTA_BETA_IDEAL (see ChemScore)	
DELTA_BETA_MAX (see ChemScore)	<float value>
CHO_COEFFICIENT (see ChemScore)	<float value>
CHO_TYPE (see ChemScore)	<string>
CHO_R_IDEAL (see ChemScore)	<float value>
CHO_DELTA_R_IDEAL (see ChemScore)	<float value>
CHO_DELTA_R_MAX (see ChemScore)	<float value>
CHO_ALPHA_IDEAL (see ChemScore)	<float value>
CHO_DELTA_ALPHA_IDEAL (see ChemScore)	<float value>

Parameters for additional ChemScore contributions used in ChemPLP (chemplp.params)	
CHO_DELTA_ALPHA_MAX (see ChemScore)	<float value>
CHO_BETA_IDEAL (see ChemScore)	<float value>
CHO_DELTA_BETA_IDEAL (see ChemScore)	<float value>
CHO_DELTA_BETA_MAX (see ChemScore)	<float value>
HBOND_SCALING (see ChemScore)	<float value>

8.3 GoldScore

8.3.1 Overview

The GoldScore fitness function is made up of four components:

- Protein-ligand hydrogen bond energy (`external H-bond`).
- Protein-ligand van der Waals (`vdW`) energy (`external vdW`).
- Ligand internal vdW energy (`internal vdW`).
- Ligand torsional strain energy (`internal torsion`).

Optionally, a fifth component, `ligand intramolecular hydrogen bond energy` (`internal H-bond`), may be added.

If any constraints have been specified, then an additional constraint scoring contribution `S(con)` will be made to the final fitness score. Similarly, when docking covalently bound ligands a covalent term `S(cov)` will be present.

By default, output files will contain a single internal energy term `S(int)` which is the sum of the `internal torsion` and `internal vdW` terms. To write these component terms to output files you will need to edit the `gold.params` file (see Altering GOLD Parameters: the gold.params File) to replace the default `VERBOSE = 0` with `VERBOSE_SCORE = 1`

Empirical parameters used in the fitness function (hydrogen bond energies, atom radii and polarisabilities, torsion potentials, hydrogen bond directionalities, etc.) are taken from the GOLD parameter file. These parameters are independent of the scoring function being used. Parameters can be customised by copying the file, editing the copy, and instructing GOLD to use the edited file (see Altering GOLD Parameters: the gold.params File).

A scoring function specific parameters file is also used; for GoldScore this is called `goldscore.params`. Parameters within this file can also be modified (see Altering GoldScore Fitness-Function Parameters: the goldscore.params File).

The fitness score is taken as the negative of the sum of the component energy terms, so that larger fitness scores are better.

The external vdW score is multiplied by a factor of 1.375 when the total fitness score is computed. This is an empirical correction to encourage protein-ligand hydrophobic contact.

During a docking run, the fitness score may appear to get worse as the docking proceeds. This is due to the fact that the effects of poor H-bond geometry and close nonbonded contacts are artificially down-weighted at early stages of the docking (annealing). Only the final fitness score (i.e. from the completed docking) has any meaning.

The fitness function has been optimised for the prediction of ligand binding positions rather than the prediction of binding affinities, although some correlation with the latter has been found.

8.3.2 Van der Waals and Hydrogen Bonding Annealing Parameters

When GoldScore is being used, the annealing parameters, `van der Waals` and `Hydrogen Bonding`, allow poor hydrogen bonds to occur at the beginning of a genetic algorithm run, in the expectation that they will evolve to better solutions.

At the start of a GOLD run, external van der Waals (`vdW`) energies are cut off when E_{ij} > van der Waals k_{ij} , where k_{ij} is the depth of the vdW well between atoms i and j. At the end of the run, the cut-off value is `FINISH_VDW_LINEAR_CUTOFF`. This allows a few bad bumps to be tolerated at the beginning of the run.

Similarly, the parameters `Hydrogen Bonding` and `FINAL_VIRTUAL_PT_MATCH_MAX` are used to set starting and finishing values of `max_distance` (the distance between donor hydrogen and fitting point must be less than `max_distance` for the bond to count towards the fitness score). This allows poor hydrogen bonds to occur at the beginning of a GA run.

The `van der Waals` and `Hydrogen Bonding` annealing parameters can only be set manually when using user-defined GA parameters settings (see Using User-Defined Genetic Algorithm Parameter Settings). Changes to the genetic algorithm parameters should be made with care.

When **GA Settings** in **Global Options** given on the left of the **GOLD Setup** window are set to **User defined**, click on **Fitness & Search Options** and select **GoldScore** from the **Scoring Function** drop-down menu. The **Annealing parameters VdW and H bond** entry boxes can then be used to specify new values.

Both the vdW and H-bond annealing must be gradual, and the population allowed plenty of time to adapt to changes in the fitness function.

8.3.3 Altering GoldScore Fitness-Function Parameters: the goldscore.params File

A GoldScore parameter file, `goldscore.params`, is provided in the `$GOLD_DIR/gold` directory.

Parameters can be customised by copying the file, editing the copy, and instructing GOLD to use the edited file. Changes to the scoring function parameters file should be made with care.

To use a modified `goldscore.params` file click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select GoldScore from the **Scoring Function** drop-down menu. Then, either enter the path and filename of the **Scoring function parameter file** or click on the **...** button and use the file selection window to choose the file.

8.4 ChemScore

8.4.1 Overview

The ChemScore scoring function is published in:

M. D. Eldridge, C. W. Murray, T. R. Auton, G. V. Paolini and R. P. Mee, J. Comput.-Aided Mol. Des., **11**, 425-445 (1997) https://doi.org/10.1023/A:1007996124545

C. A. Baxter, C. W. Murray, D. E. Clark, D. R. Westhead and M. D. Eldridge, Proteins, **33**, 367-382 (1998) https://doi.org/10.1002/(SICI)1097-0134(19981115)33:3%3C367::AID-PROT6%3E3.0.CO;2-W

ChemScore was derived empirically from a set of 82 protein-ligand complexes for which measured binding affinities were available.

Unlike GoldScore, the ChemScore function was trained by regression against measured affinity data, although there is no clear indication that it is superior to GoldScore in predicting affinities.

ChemScore estimates the total free energy change that occurs on ligand binding as:

$$\Delta G_{binding} = \Delta G_0 + \Delta G_{hbond} + \Delta G_{metal} + \Delta G_{lipo} + \Delta G_{rot}$$

Each component of this equation is the product of a term dependent on the magnitude of a particular physical contribution to free energy (e.g. hydrogen bonding) and a scale factor determined by regression, i.e.

$$\begin{aligned}
\Delta G_{0} &= v_{0} \\
\Delta G_{hbond} &= v_{1} P_{hbond} \\
\Delta G_{metal} &= v_{2} P_{metal} \\
\Delta G_{lipo} &= v_{3} P_{lipo} \\
\Delta G_{rot} &= v_{4} P_{rot}
\end{aligned}$$

Here, the ν terms are the regression coefficients and the P terms represent the various types of physical contributions to binding.

The final ChemScore value is obtained by adding in a clash penalty and internal torsion terms, which mitigate against close contacts in docking and poor internal conformations. Covalent and constraint scores may also be included.

$$ChemScore = \Delta G_{binding} + P_{clash} + c_{internal} P_{internal} + \left(c_{covalent} P_{covalent} + P_{constraint} \right)$$

Empirical parameters used in the fitness function (hydrogen bond energies, atom radii and polarisabilities, torsion potentials, hydrogen bond directionalities, etc.) are taken from the GOLD parameter file. These parameters are independent of the scoring function being used. Parameters can be customised by copying the file, editing the copy, and instructing GOLD to use the edited file (see Altering GOLD Parameters: the gold.params File).

A scoring function specific parameters file is also used; for ChemScore this is called `chemscore.params`. Parameters within this file can also be modified (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

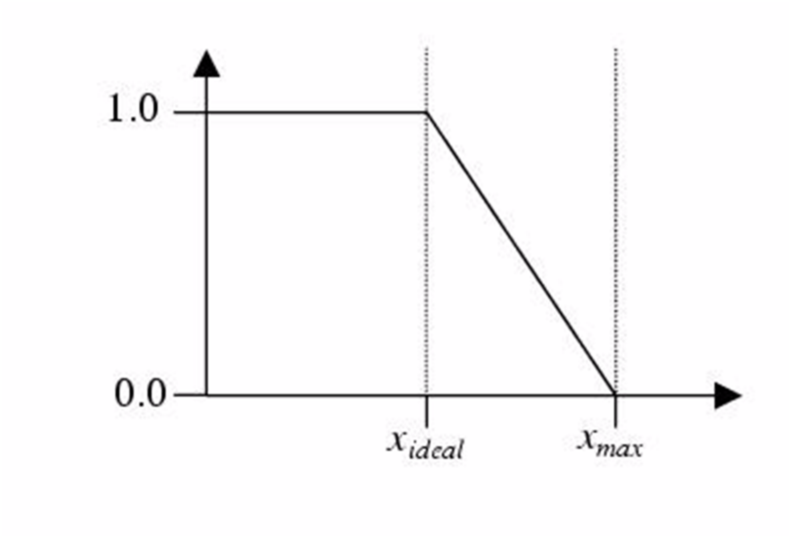
8.4.2 Block Functions in ChemScore

ChemScore uses block functions throughout its implementation to describe contact terms of various types.

A block function is of the following form:

$$B(x, x_{ideal}, x_{\max}) = \left\{ \begin{array}{l} 1 \text{ if } x \le x_{ideal} \\ 1.0 - \dfrac{x - x_{ideal}}{x_{max} - x_{ideal}} \text{ if } x_{ideal} \le x \le x_{max} \\ 0 \text{ if } x > x_{max} \end{array} \right\}$$

This functional form looks like:

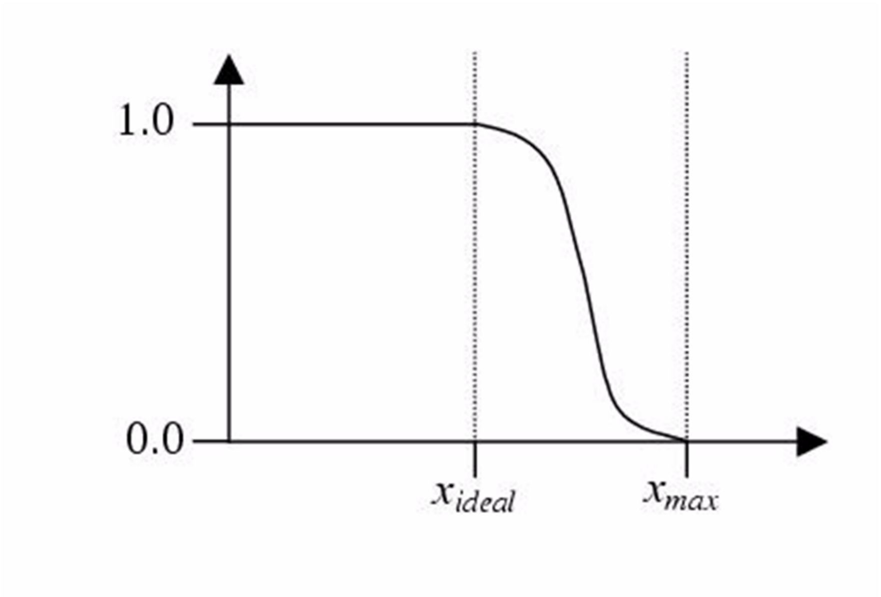


In the GOLD implementation of ChemScore, the block function is sometimes convoluted with a Gaussian function:

$$B'(x, x_{ideal}, x_{\max}, \sigma) = \frac{\int_{-\infty}^{+\infty} B(x-u, x_{ideal}, x_{\max})\, g(u,\sigma)\, du}{\int_{-\infty}^{+\infty} g(u,\sigma)\, du}$$

$$g(u,\sigma) = e^{-u^2/2\sigma^2}$$

The effect is to smooth the function, e.g.:



8.4.3 Hydrogen-Bond Terms

The hydrogen-bond term is computed as a sum over all possible donor-acceptor pairs, such that one atom belongs to the protein and the other to the ligand.

Each term in the summation is the product of three Gaussian-smoothed block functions (see Block Functions in ChemScore). The purpose of the block functions is to reduce the contribution of a hydrogen bond according to how much its geometry deviates from (a) ideal H...A distance, (b) ideal D-H...A angle and (c) ideal directionality with respect to the acceptor atom. The maximum contribution of a given donor-acceptor pair to the summation is 1; this will occur if the pair form a hydrogen bond of “ideal” geometry.

$$\Delta G_{hbond} = \sum_{\substack{\text{alldonor-acceptor}\\ \text{pairs}}} B'(\Delta r, \Delta r_{ideal}, \Delta r_{max}, \sigma_r).B'(\Delta\alpha, \Delta\alpha_{ideal}, \Delta\alpha_{max}, \sigma_\alpha).B'*(\Delta\beta, \Delta\beta_{ideal}, \Delta\beta_{max}, \sigma_\beta)$$

The tables below describe the various parameters in this equation, their meanings, and what they are called in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

D-H..A distance parameters (D= Donor, A = Acceptor)			
Term	Meaning	Name in ChemScore File	Default Value
r	The ideal hydrogen..acceptor (H...A) distance (in Å)	`R_IDEAL`	1.85
Δr	The absolute deviation of the actual H..A separation from r	Calculated for each H-bond	-
Δr_{ideal}	The tolerance window around the H..A distance, r, within which the H-bond is regarded as ideal	`DELTA_R_IDEAL`	0.25
Δr_{max}	The maximum possible deviation from the ideal distance; above this, the interaction is not regarded as an H-bond	`DELTA_R_MAX`	0.65
σ_r	The Gaussian smearing sigma associated with this term	`HBOND_R_SIGMA`	0.1

D-H..A angle parameters (D= Donor, A = Acceptor)			
Term	Meaning	Name in ChemScore File	Default Value
α	The ideal D-H..A angle (in degrees)	`ALPHA_IDEAL`	180.0

D-H..A angle parameters (D= Donor, A = Acceptor)			
$\Delta \alpha$	The absolute deviation of the actual D-H..A angle from α	Calculated for each H-bond	-
$\Delta \alpha_{\text{ideal}}$	The tolerance window around the D-H..A angle, α, within which the H-bond is regarded as ideal	`DELTA_ALPHA_IDEAL`	30.0
$\Delta \alpha_{\text{max}}$	The maximum possible deviation from the ideal D-H..A angle; above this, the interaction is not regarded as an H-bond	`DELTA_ALPHA_MAX`	80.0
σ_{α}	The Gaussian smearing sigma associated with this term	`HBOND_ALPHA_SIGMA`	10.0

DH..A-X acceptor-centred angle parameters (D= Donor, A = Acceptor, X = Heavy atom attached to A)			
Term	Meaning	Name in ChemScore File	Default Value
β	The ideal H..A-X angle (in degrees)	`BETA_IDEAL`	180.0
$\Delta \beta$	The absolute deviation of the actual H..A-X angle from β	Calculated for each H-bond	-
$\Delta \beta_{\text{ideal}}$	The tolerance window around	`DELTA_BETA_IDEAL`	70.0

DH..A-X acceptor-centred angle parameters (D= Donor, A = Acceptor, X = Heavy atom attached to A)			
	the H..A-X angle, β, within which the H-bond is regarded as ideal		
$\Delta \beta_{max}$	The maximum possible deviation from the ideal H..A-X angle; above this, the interaction is not regarded as an H-bond	`DELTA_BETA_MAX`	80.0
σ_{β}	The Gaussian smearing sigma associated with this term	`HBOND_BETA_SIGMA`	10.0

The third block function in the H-bond equation, B'^{*}, is the sum of all possible values for a given hydrogen bond. For example, a tertiary amine acceptor has three covalently-bound atoms that could be deemed as the “X” atom: in this case, the term added for an H-bond to the amine is the product of the block-function values for all three possible H..A-X angles.

Hydrogen bonds have a regression coefficient associated with them, v_1. By default, this is set to –3.34. The name of this coefficient in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File) is `HBOND_COEFFICIENT`.

8.4.4 Metal-Binding and Lipophilic Terms

The metal-binding term in ChemScore is computed as a sum over all possible metal-ion ... acceptor pairs, where the acceptor is an atom in the ligand that is capable of binding to a metal.

Each term in the summation is a Gaussian-smoothed block function (see Block Functions in ChemScore) whose purpose is to reduce the contribution of the metal-acceptor interaction if the geometry is not ideal.

$$P_{metal} = \sum_{\substack{\text{All ligand}\\ \text{acceptors}}} \sum_{\substack{\text{All protein}\\ \text{metals}}} B(r_{aM}, R_{ideal}, R_{max}, \sigma_{metal})$$

The table below describes the various parameters in this equation, their meanings, and what they are called in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

Metal-binding parameters in ChemScore

Term	Meaning	Name in ChemScore File	Default Value
r_{aM}	The actual acceptor-metal distance (in Å)	Calculated for each acceptor-metal pair	-
R_{ideal}	The ideal acceptor-metal distance	`METAL_R1`	2.6
R_{max}	The maximum acceptor-metal distance to be considered a binding interaction	`METAL_R2`	3.0
σ_{metal}	The Gaussian smearing sigma associated with this term	`METAL_R_SIGMA`	0.1

The metal-binding term has a regression coefficient associated with it, v_2. By default, this is set to –6.03. The name of this coefficient in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File) is `METAL_COEFFICIENT`.

The lipophilic term is defined in a similar way:

$$P_{lipo} = \sum_{\substack{\text{All ligand}\\ \text{lipophilic atoms}}} \sum_{\substack{\text{All protein}\\ \text{lipophilic atoms}}} B(r_{ll}, R_{ideal}, R_{max}, \sigma_{lipo})$$

The table below describes the various parameters in this equation, their meanings, and what they are called in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

Lipophilic parameters in ChemScore

Term	Meaning	Name in ChemScore File	Default Value
r_{ll}	The actual distance between the pair of lipophilic atoms (in Å)	Calculated for each atom-atom pair	-
R_{ideal}	The ideal atom...atom distance separation	`LIPO_R1`	4.1
R_{max}	The maximum separation, beyond which no interaction is deemed to occur	`LIPO_R2`	7.1
σ_{lipo}	The Gaussian smearing sigma associated with this term	`LIPO_R_SIGMA`	0.1

The difference between the metal and lipophilic parameterisation is that the lipophilic term is scored over a much longer range.

Lipophilic atoms are defined as non-accepting sulphurs, non-polar carbon atoms (polar carbon atoms are carbon atoms attached to two or more polar atoms), and non-ionic chlorine, bromine and iodine atoms.

The lipophilic term has a regression coefficient associated with it, v_3. By default, this is set to –0.117. The name of this coefficient in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File) is `LIPO_COEFFICIENT`.

8.4.5 Rotatable-Bond Freezing Term

The following formula is used to estimate the entropic loss that occurs when single, acyclic bonds in the ligand become non-rotatable upon binding:

$$p_{rot} = 1 + (1 - \frac{1}{N_{rot}}) \sum_{r} \frac{(P_{nl}(r) + P'_{nl}(r))}{2}$$

*N_{rot} *is the number of frozen rotatable bonds in the ligand (a bond is considered frozen if one or more atoms on both sides of the rotatable bond is in contact with the protein). The expression is deemed to have a value of zero if there are no rotatable bonds in the ligand.

$P_{nl}(r)$ and $P'_{nl}(r)$ are the percentages of non-hydrogen atoms on either side of the rotatable bond that are not lipophilic. For example, if there are 10 non-hydrogen atoms on one side of the bond, of which 3 are not lipophilic, and there are 20 non-hydrogen atoms on the other side, of which 2 are not lipophilic, then $P_{nl}(r)$ and $P'_{nl}(r)$ are 30% and 10%, respectively.

The regression coefficient associated with this term, v_4, has the default value 2.56. The name of this coefficient in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File) is `ROT_COEFFICIENT`.

8.4.6 Clash Penalty and Internal Torsion Terms

Clashes between protein and ligand atoms and ligand internal torsional strain are accommodated by penalty terms.

These terms are included to prevent poor geometries in docking.

The clash penalty terms in ChemScore differ on the nature of the contact, i.e. whether it is a hydrogen-bonding contact, a metal-binding contact or neither of these.

Any hydrogen bond with an H...A distance shorter than r_{hbond} Å contributes a clash term of:

$$P_{clash-hbond} = \frac{20.0\times\left(r_{hbond} - r\right)}{\Delta G_{hbond} \times r_{hbond}}$$

The value of r_{hbond} (default = 1.6 Å) can be changed by altering the parameter `CLASH_RADIUS_HBOND` in the ChemScore file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

Any metal coordination contact shorter than r_{metal} Å contributes a clash term of:

$$P_{clash-metal} = \frac{20.0 \times (r_{metal} - r)}{\Delta G_{metal} \times r_{metal}}$$

The value of r_{metal} (default = 1.3 Å) can be changed by altering the parameter `CLASH_RADIUS_METAL` in the ChemScore file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

All other ligand-protein interatomic contacts contribute clash terms of the following form:

$$P_{clash-other} = 1.0 + \frac{4.0(r_{clash} - r)}{r_{clash}}$$

r_{clash} varies with contact type: for contacts to protein sulfur atoms, it is set to 3.35 Å; for all other contacts, it is set to 3.10 Å. These settings correspond to the parameters `CLASH_RADIUS_SULPHUR` and `CLASH_RADIUS_GENERAL` in the ChemScore file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

Internal ligand strain is accommodated by clash terms in combination with torsional strain terms of the form:

$$P_{internal} = \sum_{\substack{\text{All rotatable} \\ \text{bonds}}} A_i \left(1 - \cos(n\Phi - \Phi_0)\right)$$

Bonds are deemed to be rotatable if they are single and acyclic and involve pairs of atoms with hybridisation states sp^3-sp^3, sp^3-sp^2 or sp^2-sp^2.

The parameters A, n and ***Φ*** in the above equation are set in the ChemScore file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File). The relevant lines are `SP3_SP3_BOND`, `SP3_SP2_BOND`, `SP2_SP2_BOND` and `UNKNOWN_BOND`. The syntax is of the form:

```
SP3_SP3_BOND A nΦ<sub>0</sub>
```

For example:

```
SP3_SP3_BOND 0.18750 3.0 3.1515926
```

The overall contribution of intramolecular strain to the scoring function is scaled by the coefficient called `INTRA_COEFFICIENT` in the ChemScore file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

8.4.7 Covalent Term

When covalent bonding is switched on (see Setting Up Covalently Bound Ligands the ChemScore function is modified in the following ways:

- The clash term (see Clash Penalty and Internal Torsion Terms) is reduced so that no clash is registered for 1-2 or 1-3 contacts around the link atoms in the protein and ligand.
- Torsion terms (see Clash Penalty and Internal Torsion Terms) are added for the rotatable parts of the linkage.
- A valence-angle bending term is added to the overall energy to penalise poor link geometries.
- The weight of the covalent link energy in the ChemScore function is controlled by the parameter called `LINK_BEND_COEFFICIENT` in the ChemScore parameter file (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

8.4.8 Constraint Terms

Constraints (see Setting Constraints) are implemented in ChemScore in the same way as they are in GoldScore.

8.4.9 Altering ChemScore Fitness-Function Parameters; the ChemScore File

The ChemScore parameter file is stored in the GOLD distribution directory. It contains all the parameters used by the GOLD implementation of ChemScore. A full description of the meaning of the various parameters is given elsewhere (see ChemScore).

The ChemScore file can be customised by copying it, editing the copy, and instructing GOLD to use the edited file.

To use a modified `chemscore.params` file click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select ChemScore from the **Scoring Function** drop-down menu. Then, either enter the path and filename of the **Scoring function parameter file** or click on the **...** button and use the file selection window to choose the file.

The format of the ChemScore file is quite strict: incorrect editing may cause GOLD to behave in unexpected ways or even to crash. Due to the large number of parameters, no guarantee can be given that the program will behave reliably with anything other than the default parameterisation.

8.5 Astex Statistical Potential (ASP)

8.5.1 Overview

For a more thorough discussion on the Astex Scoring Potential (ASP) fitness function, please see: W. T. M. Mooij and M. L. Verdonk, Proteins: Struct. Func. And Bioinf., **61**, 272-287 (2005).

ASP is an atom-atom potential derived from a database of protein-ligand complexes and can be compared to other such scoring potentials, e.g. PMF and Drugscore. ASP has comparable accuracy to the ChemScore and GoldScore fitness functions.

Traditional scoring functions are based on force fields or on regression, where parameters are derived from a set of experimental binding affinities and structures. ASP uses a different approach; information about the frequency of interaction between ligand and protein atoms is gathered by analysing existing ligand-protein structures in the PDB and this information is used to generate statistical potentials. Depending on the database where the atom-atom potentials are taken from, the scoring function created can be targeted to certain proteins (see Targeted Scoring Functions). A general scoring function (one that can be used for “all” types of proteins) would take its interactions from the ‘entire’ PDB while a more targeted function would be created from specific families of proteins. Atom-atom potentials for a general scoring function are included in the GOLD distribution.

Empirical parameters used in the fitness function (hydrogen bond energies, atom radii and polarisabilities, torsion potentials, hydrogen bond directionalities, etc.) are taken from the GOLD parameter file. These parameters are independent of the scoring function being used. Parameters can be customised by copying the file, editing the copy, and instructing GOLD to use the edited file (see Altering GOLD Parameters: the gold.params File).

A scoring function specific parameters file is also used; for ASP this is called `asp.params`. Parameters within this file can also be modified (see Altering ASP Fitness-Function Parameters: the asp.params File).

8.5.2 The Reference State

The ASP scoring function differs from other statistical potentials by the choice of the so-called reference state. The reference state is the expected number of contacts if there were no interaction between the atoms (i.e. at long distances), incorporating any corrections. The reference state determines how the raw distribution of observations is transformed into potentials.

Contacts between atoms are usually determined by radial distribution functions (RDFs). Given an atom at some position the RDF will tell us how many other atoms we can expect to find at a distance between r to r+dr, where dr is the bin width in the RDF and can be thought of as the ‘thickness’ of a spherical shell. A statistical potential between two atom types i and j is defined as:

$$\mathrm{StatScore}(i,j,r) = -\ln\frac{n_{\mathrm{obs}}(i,j,r)}{n_{\mathrm{exp}}^{\mathrm{StatScore}}(i,j,r)}$$

where the denominator is the reference state of the potential; for ASP the reference state is given by:

$$n_{\mathrm{exp}}^{\mathrm{ASP}} = \left\langle\frac{n_{\mathrm{obs}}(i,j,r')}{f_p(i,r')f_l(j,r')4\pi r'^2\Delta r}\right\rangle_{r'=6.0}^{r'=8.0}\cdot f_p(i,r')\cdot f_l(j,r')\cdot 4\pi r^2\Delta r$$

The average contact density is taken to be the average between 6.0 and 8.0 Å of the corrected RDF. At this long range, atoms are not considered to make any specific interactions and should ensure that the scores of the function are close to zero at this length. The two terms f_p and f_s denote the protein and ligand volume corrections to the contacts, respectively. These two terms are added to account for the difference in accessibility of different protein and ligand atoms; if no excluded volume corrections are included in the reference state the expected number of contacts is simply the product of the average contact density in the sphere with radius R_{max} and the volume of a spherical shell at distance r. The way that these corrections are defined differs from other potential scoring functions and the inclusion of a protein correction term is novel to the ASP fitness function. The ligand correction term can be compared to the corresponding term included in Drugscore.

8.5.3 The Generation of Potentials

To derive the pair potentials between ligand and protein atoms, a database of protein ligand complexes from the PDB is used. Bond types of the protein are assigned based on residue and atom names. Additional atom types are defined to separate backbone nitrogen and oxygen atoms from those in aspargine, glutamine, aspartic acid and glutamic acid side chains, and to distinguish serine and threonine hydroxyl oxygen from those in tyrosine. Ligands in the database were divided into three separate categories i) covalent ii) cofactor and iii) normal. Only binding sites with normal ligands with a heavy-atom count between 6 and 60 were included in the database (cofactors are treated as part of the protein).

Atom-atom potentials were calculated for each atom pair with an excess of 150 observations in the database using the ASP reference state, for atom types with fewer observation the potential was set to zero for all distances. For short distances there will be no observed contacts and the potential is set to 10. The atom-atom potentials for all atom types can be found in the GOLD installation directory, `<Installation folder>/Discovery_2022/GOLD/gold/asp_tables`.

The statistical potentials are augmented with the ChemScore clash term and internal energy term (see Clash Penalty and Internal Torsion Terms). The internal energy term is needed to prevent the docking of high-energy ligand conformations, while the clash term should prevent protein-ligand clashes where the supplied potential is too soft to provide sufficient repulsion between protein and ligand atoms and at the same time preventing overlap between atoms with no potential (i.e. too few observations for the generation of a non-zero potential). The final ASP fitness can be written:

$$\text{ASP Fitness} = -C_s \sum_p \sum_l \text{StatScore}(p, l, r_{pl}) - c_{int}E_{int} - c_{clash}E_{clash}$$

$$\sum_p \sum_l \text{StatScore}(p, l, r_{pl}) = \text{S(map)}$$

The total StatScore (written as S(map) in the ligand output file) is a summation over all combinations of protein atoms, p, and ligand atoms, l, within 6.0 Å, and r_{pl} is the distance between protein atom p and ligand atom l. C_s is a scaling factor and c_{int} and c_{clash} are the internal energy and clash coefficients, respectively. C_s is per default set to 0.2 and both c_{int} and c_{clash} are set to 1.0. To speed up scoring and docking, grids are precalculated for each atom type using a grid spacing of 0.3 Å. The max distance for interaction (6.0 Å), the scaling factor (C_s) and the respective weights for the internal (c_{int}) and clash energy (c_{clash}) together with the grid spacing can be altered in the `asp.params` file (see Altering ASP Fitness-Function Parameters: the asp.params File).

8.5.4 Metal and Hydrogen Bond Correction

In ASP you can add a correction when docking to metal-containing receptors. When adding the metal correction to the ASP score, a hydrogen bond correction is included by default. The hydrogen bond correction is similar to the one found in ChemScore (see Hydrogen-Bond Terms). The final score is calculated as:

$$\text{ASP Fitness} = -C_S(\text{S(map)} + \text{S(hbond)} + \text{S(metal)}) - c_{int}E_{int} - c_{clash}E_{clash}$$

The S(metal) is calculated for single metal-ligand atom interactions based on the actual distance between the metal and the ligand acceptor and is corrected for the metal-ligand score in S(map). This means that the S(map) contribution (based on grid points) is subtracted from S(metal) to offset the contribution that is already present in the ASP grid for the acceptor type. The S(metal) score more accurately reflects the metal-ligand interaction.

The S(hbond) correction corresponds to the metal correction as it calculates the score of hydrogen bonds from the actual distance as opposed to S(map) where the pre-calculated grid points are used. The hbond score is the multiplied with the `HBOND_CORRECTION_FACTOR`, making it possible to weigh the contribution of hydrogen bonds to the final score; this parameter can be changed by editing the `asp.params` file (see Altering ASP Fitness-Function Parameters: the asp.params File). The S(hbond) correction is similar to the same term found in ChemScore, since the deviation from ideal geometry is taken into account when calculating the score of the hydrogen bond (see Hydrogen-Bond Terms). The S(hbond) contribution is offset by the score already present in S(map).

8.5.5 Covalent Docking and Docking with Constraints

Covalent docking with ASP is handled by adding a covalent term to the calculated score. The implementation is the same as for ChemScore (see Covalent Term).

Using constraints in conjunction with ASP is carried out using the same principle as with GoldScore and ChemScore (see GoldScore).

8.5.6 Targeted Scoring Functions

The use of statistical potentials in a scoring function enables the creation of targeted fitness functions to certain proteins. This is done by using target-specific information when calculating the atom-atom potentials. Instead of using the information from a general database, such as the PDB, information can be taken from a smaller set of crystal structures, which could be comprised of only one family of proteins or if there is insufficient information in the specific-target database one can mix in information from the general PDB database.

You can store your customised potentials in a directory specified in the `asp.params` file (see Altering ASP Fitness-Function Parameters: the asp.params File).

8.5.7 Performance of the ASP fitness function

On the CCDC/Astex validation set, ASP has similar success rate as Goldscore and Chemscore; for a more complete discussion on the accuracy of the ASP please refer to the original publication (see Overview).

8.5.8 Altering ASP Fitness-Function Parameters: the asp.params File

The ASP parameter file is stored in the `<Installation folder>/ccdc-software/gold/GOLD/gold/` directory. It contains all the parameters used by the GOLD implementation of ASP. A full description of the meaning of the ASP specific parameters is given below.

The ASP file can be customised by copying it, editing the copy, and instructing GOLD to use the edited file.

To use a modified `asp.params` file click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select ASP from the **Scoring Function** drop-down menu. Then, either enter the path and filename of the **Scoring function parameter file** or click on the **...** button and use the file selection window to choose the file.

The format of the ASP file is quite strict: incorrect editing may cause GOLD to behave in unexpected ways.

The ASP fitness function shares many of its parameters with ChemScore, see the `chemscore.params` section (see ChemScore) for an explanation. However, please note that the default value of these may differ from the value used in ChemScore.

The `asp.params` file contains the following parameters:

`ASP_COEFFICIENT`: Default `0.2`. The total contribution to the score by the potential is scaled by a coefficient which has been optimised to 0.2 (see The Generation of Potentials).

CLASH_COEFFICIENT: Default 1.0. The CLASH_COEFFICIENT controls the weight of the clash term to the overall score (see The Generation of Potentials).

INTERNAL_COEFFICIENT: Default 1.0. The INTERNAL_COEFFICIENT controls the weight of the internal energy of the ligand to the overall score (see The Generation of Potentials).

HBOND_FUNCTION: Default NONE. The HBOND_FUNCTION can be turned on by replacing NONE by ASP. When switched on it will also include the metal correction (see Metal and Hydrogen Bond Correction).

HBOND_CORRECTION_FACTOR: Default 1.0 (see Metal and Hydrogen Bond Correction).

CLASH_FUNCTION: Default ASP. The CLASH_FUNCTION is used for the calculation of clashes between the ligand and the protein. The clash term is evaluated in the same way as for ChemScore (see Clash Penalty and Internal Torsion Terms).

ASP_GRID_SPACING: Default 0.3. Parameter to control the density of the pre-calculated grid used for evaluation of the atom-atom potentials (see The Generation of Potentials).

ASP_GRID_INTERPOLATE: Uncomment the ASP_GRID_INTERPOLATE to more accurately calculate the distance between atoms for the calculation of the score. Grid points surrounding atoms are used to interpolate a more exact atom location. This will give a similar effect as increasing the grid density.

ASP_MAX_DISTANCE: Default 6.0. The max distance set for interaction between ligand and protein atoms.

ASP_GRID_LOOKUP: This setting is on by default; the score is evaluated from the generated grid points.

ASP_DIRECTORY: Default DEFAULT. Sets the location of the ASP potentials, by default the location is $GOLD_DIR/gold/asp_tables.

TARGETED_ASP_DIRECTORY. Uncomment this parameter to specify the location of customised targeted ASP potentials (see Targeted Scoring Functions).

SAVE_ASP_MAPS: Default 0. If set to 1 the map generated for each ASP ligand atom type is printed out.

8.6 Altering GOLD Parameters: the gold.params File

The parameter file `gold.params` is stored in the GOLD distribution directory. It contains the parameters used by GOLD (e.g. hydrogen bond energies, atom radii and polarisabilities, torsion potentials, hydrogen bond directionalities, etc.) other than those which are specified in the configuration file (i.e. can be set via the GOLD front end) or those specific to the various scoring functions.

It also contains parameters that control the general behaviour of GOLD, e.g. whether the final solution from a genetic algorithm run is to be minimised via a Simplex procedure before being saved. Another paramerer that has been added recently is the 'Docking Duration Limit'. Occasionally, some combination of docking config and ligand can take a long time.

If required, a time limit may be imposed such that GOLD will exit the docking of a particularly slow ligand and move on to the next.

The parameter file can be customised by copying it, editing the copy, and instructing GOLD to use the edited file.

To use a modified `gold.params` file click on **Fitness and Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window. Then, either enter the path and filename of the **GOLD parameter file**, or click on the **...** button and use the file selection window to choose the file.

If the parameter file is set to **DEFAULT** then the standard GOLD distribution parameter file is copied to the current directory.

GOLD gets the location of the parameter file from the configuration file line `param_file = <parameter file location>`. This is most easily defined using the **Parameter File** button in the front end.

The format of the parameter file is quite strict: incorrect editing may cause GOLD to behave in unexpected ways or even to crash. Because of the large number of parameters, no guarantee can be given that the program will behave reliably with anything other than the default parameterisation.

For more information see the comments in the parameter file, `gold.params`.

8.7 Targeted Scoring Functions

8.7.1 Kinase Scoring Function

Weak CHO interactions can be accounted for by inclusion of a ChemScore term that calculates a contribution for weak hydrogen bonds. This term can be useful when dealing with particular proteins, e.g. most kinases contain weak N-heterocycle CH...O hydrogen bonds.

This term can be enabled by using the `chemscore.kinase.params` scoring function parameters file located within the `$GOLD_DIR/gold` directory.

To employ this file click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select ChemScore from the **Scoring Function** drop-down menu. Then, either enter the path and filename of the **Scoring function parameter file**, or click on the **...** button and use the file selection window to choose the `chemscore.kinase.params` file.

This will enable the recognition of activated CH groups for hydrogen bonding. Active CH groups are those in aromatic rings next to nitrogens (e.g. the CHs in an imidazole ring). These groups are recognised both in the ligand and protein active site.

For further details please refer to Virtual Screening Using Protein-Ligand Docking: Avoiding Artificial Enrichment (see References).

8.7.2 Heme Scoring Function

The heme scoring function is available for both GoldScore (see GoldScore) and ChemScore (see ChemScore).

By default, GOLD makes no distinction between different H-bond acceptors in terms of their strength of interaction with the metal. A publication by Kirton et al. (S. B. Kirton, C. W. Murray, M. L. Verdonk and R. D. Taylor, Proteins, **58**, 836-844, 2005, DOI: 10.1002/prot.20389) demonstrated how metal parameters can be set up in GOLD for both GoldScore and ChemScore, to take account of different H-bond acceptor types. They described the use of ligand-specific iron parameters in the context of docking to heme-containing proteins and demonstrated improved performance. It is possible in GOLD to optionally use these parameters.

The parameters are derived from contact statistics obtained from the CSD and PDB databases. Parameters were derived for both GoldScore and ChemScore.

These parameters can be used by choosing the appropriate scoring function `.params` file from those that have been supplied with the GOLD installation. The scoring function `.params` files that are available are:

- `goldscore.p450_csd.params`
- `goldscore.p450_pdb.params`
- `chemscore.p450_csd.params`
- `chemscore.p450.pdb.params`

The files are located within the `$GOLD_DIR/gold` directory. The content below shows the iron parameters for GoldScore, derived from the CSD, as displayed in the `goldscore.p450_csd.params` file:

```
# HEME SCORING FUNCTION

#======================

# Ref: Kirton et al. Proteins (2005) 58 pp836-844

MAKE_PLANAR_N_LIPO 1

# METAL COORDINATION CSD

METAL_COORD 2.00 0.9 Fe | N.2

METAL_COORD 2.00 0.9 Fe | N.ar

METAL_COORD 2.00 0.8 Fe | N.3

METAL_COORD 2.00 0.3 Fe | O.2

METAL_COORD 2.00 0.9 Fe | O.co2

METAL_COORD 2.00 0.3 Fe | O.3(1H)

METAL_COORD 2.00 0.0 Fe | O.3(0H C C)

METAL_COORD 2.00 0.9 Fe | S.3(0H C)

METAL_COORD 2.00 0.3 Fe | S(=C)
```

To employ one of these files, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and select GoldScore or ChemScore from the **Scoring Function** drop-down menu. Then, either enter the path and filename of the **Scoring function parameter file** or click on the **...** button and use the file selection window to choose the file.

It was found necessary by Kirton et al. to assign the planar nitrogens in the heme molecules as lipophilic when using the ChemScore scoring function. In order to bring this about, the `chemscore.p450` parameter files therefore contain the additional keyword:

```
MAKE_PLANAR_N_LIPO 1
```

Note: Use of this keyword has only been validated for nitrogen atoms within heme containing proteins. Improvements in docking performance when used with non-heme containing proteins are not guaranteed.

9 Ligand Flexibility

9.1 Ring Conformations

9.1.1 Flipping Ring Corners

To allow free corners of ligand rings to flip during docking click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window, activate the **flip ring corners** tick box in the top section of the window.

This will result in GOLD performing a limited conformational search of cyclic systems by allowing free corners of rings to flip above or below the plane of their neighbouring atoms.

The rules governing flipping of ring corners in GOLD are given in: A. W. R. Payne & R. C. Glen, J. Mol. Graphics, **10**, 74-91 (1993).

9.1.2 Using CSD Ring Conformation Templates

A library of ring conformations extracted from the Cambridge Structural Database (CSD) can be utilised by GOLD. This allows GOLD to vary ligand non-fused ring conformations during docking.

The use of ring conformation templates may improve the chances of GOLD finding the correct answer by allowing the algorithm to sample ring conformations that are commonly observed in crystal structures.

Rings in the ligand are matched against a template library. If a matching template is found, then the conformation of that ligand ring will be varied during docking using a set of supplied alternative conformations for that ring.

Each time an alternative conformation is sampled the ligand conformation is changed to match the new conformation by altering the bond lengths, internal ring angles and torsions. A ring torsional strain energy term is also added to the ligand internal energy.

To use ring conformation templates during docking click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window, activate the **Match template conformations** tick box in the top section of the window.

Information on the composition of the CSD ring conformation library and how ring templates are matched at run time is available (see The CSD Ring Conformation Library and Matching Templates at Run Time).

It is also possible to specify your own ring templates, and the allowed alternative conformations for those rings (see User-Defined Ring Conformations).

9.1.3 The CSD Ring Conformation Library and Matching Templates at Run Time

GOLD identifies each ring in the ligand and attempts to match it to a ring template in the following encrypted file:

```
$GOLD_DIR/gold/ring_conformations/template_library
```

Details of matched rings are written to the `gold_<ligand_name>_m<n>.log` file under the heading `Match Ring Templates`.

The encrypted `template_library` file contains 1274 ring templates. Each template represents a different ring identified within the Cambridge Structural Database (CSD). For each template the number of alternative conformations that will be explored during docking will vary depending on the abundance and suitability of data within the CSD.

The index number of the final ring conformation used in each docking solution is written to the `gold_<ligand_name>_m<n>.log` file under the heading `Chromosome decoded`.

9.1.4 User-Defined Ring Conformations

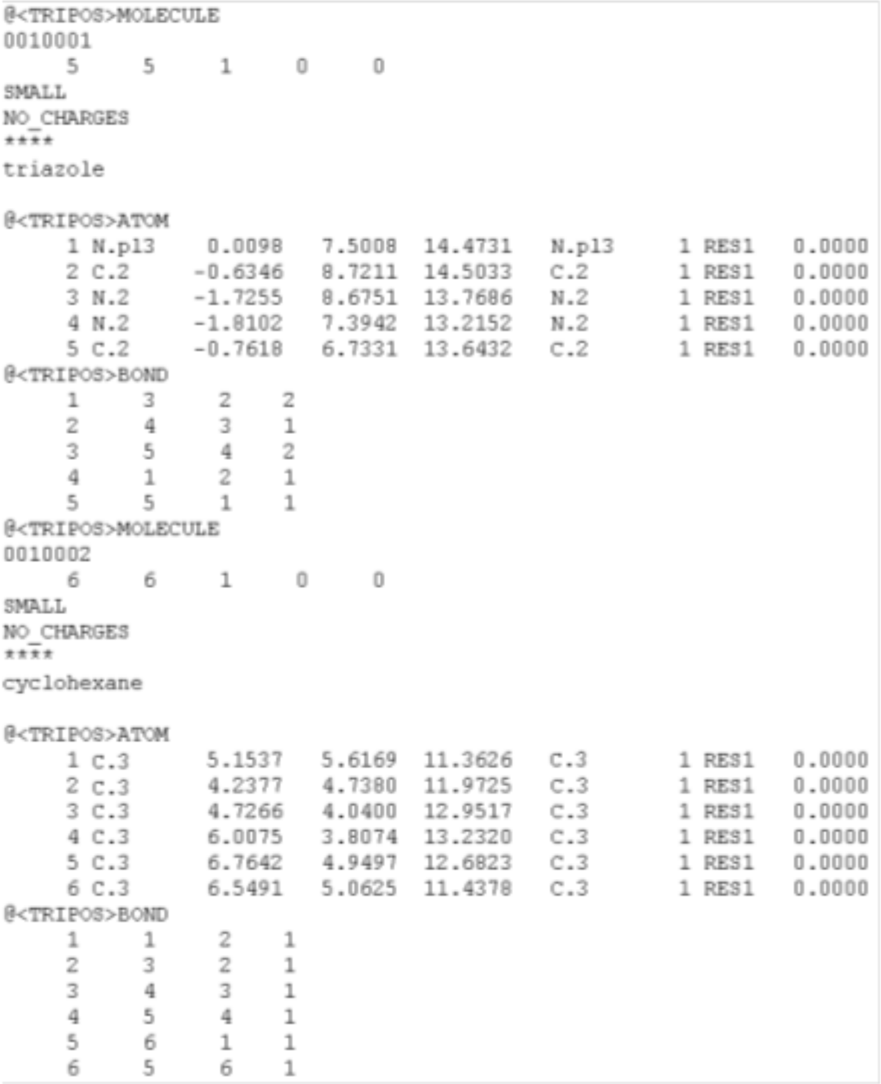
GOLD can vary ligand ring conformations during docking. A library of ring conformations extracted from the Cambridge Structural Database (CSD) is supplied with GOLD for this purpose (see The CSD Ring Conformation Library and Matching Templates at Run Time).

For non-fused rings, it is also possible to specify your own ring templates, and the allowed alternative conformations for those rings. This is useful if you wish to add a new ring template (not already present in the CSD derived library), or if you wish to override the CSD conformations for an existing template. When isolating ring templates from a file, it is recommended that you set atom types manually (see Manually Setting Atom and Bond Types) since an isolated ring without its excocyclic substituents can cause problems with automatic atom-typing.

First, you must create a ring template library file in the directory:

```
$GOLD_DIR/gold/user_ring_conformations/template_library.mol2
```

This mol2 file should contain all user-defined ring types. Rings in the ligand are matched against the rings in this template file. The atom types in the `template_library.mol2` file must therefore match the ligand atom types exactly, i.e. after any ligand atom typing has been performed (see Automatically Setting Atom and Bond Types). The molecule identifiers in the `template_library.mol2` file must start at 0010001. The identifier must be incremented by 1 for each successive ring in the file. For example, the following `template_library.mol2` file contains two templates, a triazole and a cyclohexane ring:



```
@<TRIPOS>MOLECULE
0010001
     5     5     1     0     0
SMALL
NO_CHARGES
****
triazole

@<TRIPOS>ATOM
     1 N.pl3     0.0098    7.5008  14.4731   N.pl3      1 RES1    0.0000
     2 C.2      -0.6346    8.7211  14.5033   C.2        1 RES1    0.0000
     3 N.2      -1.7255    8.6751  13.7686   N.2        1 RES1    0.0000
     4 N.2      -1.8102    7.3942  13.2152   N.2        1 RES1    0.0000
     5 C.2      -0.7618    6.7331  13.6432   C.2        1 RES1    0.0000
@<TRIPOS>BOND
     1     3     2    2
     2     4     3    1
     3     5     4    2
     4     1     2    1
     5     5     1    1
@<TRIPOS>MOLECULE
0010002
     6     6     1     0     0
SMALL
NO_CHARGES
****
cyclohexane

@<TRIPOS>ATOM
     1 C.3       5.1537    5.6169  11.3626   C.3        1 RES1    0.0000
     2 C.3       4.2377    4.7380  11.9725   C.3        1 RES1    0.0000
     3 C.3       4.7266    4.0400  12.9517   C.3        1 RES1    0.0000
     4 C.3       6.0075    3.8074  13.2320   C.3        1 RES1    0.0000
     5 C.3       6.7642    4.9497  12.6823   C.3        1 RES1    0.0000
     6 C.3       6.5491    5.0625  11.4378   C.3        1 RES1    0.0000
@<TRIPOS>BOND
     1     1     2    1
     2     3     2    1
     3     4     3    1
     4     5     4    1
     5     6     1    1
     6     5     6    1
```

Secondly, a set of allowed alternative ring conformations for each ring in `template_library.mol2` must be created and stored in the following files:

`$GOLD_DIR/gold/user_ring_conformations/0010001.mol2` `$GOLD_DIR/gold/user_ring_conformations/0010002.mol2` and so on...

The ring conformation filenames are based on the corresponding template identifier. In the above example the triazole ring template has the molecule identifier 0010001. The alternative conformations of that ring that will be used during docking must therefore be located in `$GOLD_DIR/gold/user_ring_conformations/0010001.mol2`.

Note that any user-defined templates, and their corresponding set of allowed conformations, will be used in preference to the supplied CSD conformations.

9.2 Flipping Amide Bonds

During ligand initialisation, ligand amide groups (including thioamides, ureas and thioureas) will be set to the trans conformation. Flattening to the trans conformation will also occur if the (O)C-N(H) torsion is greater than twenty degrees out of plane. Note: If the (O)C-N(H) torsion is greater than five but less than 20 degrees out of the trans plane, the bond will not be flattened and a warning message will be written to the `gold.err` file.

Click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and switch on the **Flip amide bonds** check box to allow amides, thioamides, ureas and thioureas in the ligand to flip between cis and trans conformations during docking.

In order to flip between cis and trans conformations the CO-NRR' torsion is first made planar (at the initialised trans conformation). Note: N,N-disubstituted amides are not made planar; CO-NH_2 will be set so that the NH_2 group is in plane with the CO (care must be taken that the input RNH_2 group itself is planar since GOLD will not change this).

On occasion this flattening of the CO-NRR' torsion may result in clashes in the initialised structure. If this occurs, it is advisable to turn off normalisation of amide bonds using the `FLATTEN_BONDS` keyword in the `gold.params` file. In this case it is recommended to fix the bond by switching off **Flip amide bonds**, or by explicitly specifying that the appropriate rotatable bonds are held at their input conformation (see Fixing Rotatable Bonds at Their Input Conformation).

If the use of torsion angle distribution has been enabled (see Using Torsion Angle Distributions) GOLD will attempt to match amide torsions against the torsion angle distributions file. If an amide torsion matches, this will override the **Flip amide bonds** flag **s**etting. Note: Data in the CSD show that both cis and trans conformations occur in ureas; it is therefore recommended that amide flipping be turned on in order to sample R-N-C(O)-N torsions of 0 degrees when docking ureas.

9.3 Flipping Pyramidal Nitrogens

Click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and switch on the **Flip pyramidal N** check box to allow pyramidal (i.e. non-planar sp^3) nitrogens to invert during docking (otherwise, they will be held fixed at the input geometry).

Given a non-planar group RR’R”N or tetrahedrally surrounded RR’R”NH, the **Flip pyramidal N** switch enables flipping of the local stereochemistry around the nitrogen (the energy barrier for this umbrella-like change of geometry around the nitrogen is low).

Flipping only changes the stereochemistry around RR’R”N and RR’R”NH nitrogens. It does not affect other chiral centres.

9.4 Intramolecular Hydrogen Bonds

Click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and switch on the **Detect internal H-bonds** check box to allow intramolecular hydrogen bonds in the ligand to be formed during docking.

Use this with care as it can make ligands like methotrexate curl up.

9.5 Flipping Planar Nitrogens

Click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window. In the section **Ring-NR1R2 group flexibility** it is possible to allow planar trigonal nitrogens in the ligand (bound to sp^2 carbons) to flip between cis and trans conformations during docking (otherwise, they will be held fixed at the input geometry).

The behaviour of both ring-NHR and ring-NR1R2 groups during docking can be independently controlled. The following options are available for each:

- **Fix**: this fixes ring-NHR or ring-NR1R2 bonds at their input conformation.
- **Flip**: allows ring-NHR and ring-NR1R2 to flip (i.e. rotate 180 deg.) during docking.
- **Rotate**: use this option to allow free rotation of ring-NHR or ring-NR1R2 groups during docking.

9.6 Protonated Carboxylic Acids

Protonated carboxylic acids can be held rigid at their input conformation or allowed to flip or rotate freely during docking.

Click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and select one of the options for **OH groups flexibility in carboxylic acids**:

- **Fix**: this fixes protonated carboxylic acids bonds at their input conformation.
- **Flip**: allows protonated carboxylic acids to flip (i.e. rotate 180 deg.) during docking.
- **Rotate**: use this option to allow free rotation of protonated carboxylic acids groups during docking.

9.7 Using Torsion Angle Distributions

9.7.1 Enabling Use of Torsion Angle Distributions

Torsion angle distributions extracted from the Cambridge Structural Database (CSD) can be utilised by GOLD. These distributions can be used to restrict the ligand conformational space sampled by the genetic algorithm.

Using torsion angle distributions in this way will not make GOLD go any faster. However, it may improve the chances of GOLD finding the correct answer by biasing the search towards ligand torsion-angle values that are commonly observed in crystal structures. It may also improve convergence and so make improve accuracy with faster settings (see Controlling Accuracy and Speed with Genetic Algorithm Parameter Settings).

To enable the use of torsion angle distributions, click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and switch on the **Use Torsion Angle Distributions** check box.

A torsion angle distribution file must be specified. Either enter the path and filename of the file or click on the **...** button and use the file selection window to choose the file. Two torsion angle distribution files are provided with GOLD:

`tor_lib_2020.tordist` - this is the default file since the 2023.3 release.

`mimumba.tordist` - this contains all the torsional distributions used in the MIMUMBA program (Klebe and Mietzner, J.Comput.-Aided Mol.Des., **8**, 583-606, 1994).

`gold.tordist` - this was the default file prior to the 2023.3 release.

Since the 2023.3 release, a new torsion distribution file has been used, with patterns partially adapted from this publication, which we acknowledge here.

The Torsion Library: Semi-automated Improvement of Torsion Rules with SMARTScompare. (Penner et al, Journal of Chemical Information and Modeling, , **62**, 7, 1644–1653, (2022) https://doi.org/10.1021/acs.jcim.2c00043)

If you wish to maintain pre-2023.3 behaviour, you can switch back to the `gold.tordist` file by changing your gold configuration file for docking.

Some additional patterns have been added to the list, and some more general patterns have been removed.

It is possible to customise torsion angle distribution information by editing one of the standard torsion angle distribution files (see Editing Torsion Angle Distribution Files).

9.7.2 Editing Torsion Angle Distribution Files

Copy the default file torsion angle distributions file that is provided in the `$GOLD_DIR/gold/` directory to the current directory. After making your changes instruct GOLD to use the edited file (see Enabling Use of Torsion Angle Distributions).

The format of entries in the file is quite strict: incorrect editing of the file may cause GOLD to behave in unexpected ways or even to crash.

For further information refer to Appendix E: The Torsion Angle Distribution File.

9.7.3 Matching Torsion Angle Distributions at Run Time

GOLD identifies each rotatable bond in the ligand and attempts to match it to a torsion angle distribution in the torsion angle distribution file. This includes bonds that are identified by GOLD as flippable (e.g. if torsions are switched on, then ligand carboxylic acids (O)C-OH will also use a torsion distribution).

Details of matched torsions are written to the `gold_<ligand_name>_m1.log` file, specifically:

- An itemised list of which torsions have been matched during ligand initialisation, including the torsion name, e.g.

```
Rotatable bond [40 41 61 63] matches torsion: ester
    C.3 | O.3 | C.2 ( =O.2 ) | C.3
    Rotatable bond [65 64 63 61] matches torsion: acid T1
    C.2 (O.co2 O.co2 ) | C.3 ( 2H ) | C.3 ( 2H ) | C
    Rotatable bond [67 65 64 63] matches torsion: acid T2
    O.co2 | C.2 ( O.co2 ) | C.3 ( 2H ) | C.3 ( 2H C)
```

- Matched torsion angles are now identified in the rotatable ligand bonds list, written out at the end of the docking run when the chromosome is decoded, e.g.

```
Chromosome decoded:
    Ligand Torsions
    [ 61 41 40 3 ] -51.44
    [ 40 41 61 63 ] 165.82 matched torsion: ester
    [ 26 57 48 38 ] 60.63
    [ 65 64 63 61 ] -71.33 matched torsion: acid T1
    [ 67 65 64 63 ] 161.95 matched torsion: acid T2
```

In some cases, a rotatable bond may match more than one torsion angle distribution. If this happens, a score is calculated for each torsion angle distribution and the distribution with the highest score is selected. Note: A weighting scheme is used when matching rotatable bonds in the ligand to a torsion angle distribution such that more specific torsion definitions are taken in preference to more generic ones.

Each portion of the torsion angle distribution contributes to the score as follows:

Torsion Angle Distribution	Score
Element atom type	1.5
SYBYL atom type	2.0
Fragment	3.0
Hydrogen count	2.0
Bond linkage	0.5

9.8 Overriding Automatic Bond Settings

When using ligand flexibility options, e.g. **Flip amide bonds** (see Flipping Amide Bonds) or **Ring-NR1R2 group flexibility** (see Flipping Planar Nitrogens), the bond in question is treated in a specific manner at ligand initialisation to prepare it for the docking run (in both the aforementioned cases, the bond is flattened at ligand initialisation prior to it being flipped during docking).

If a bond is e.g. desired to rotate freely rather than flip during docking, this fine-grained control can be achieved by using the `rotatable_bond_override.mol2` file, found in the `$GOLD_DIR/gold/` directory. Some fragments are already provided (which can be edited), however user-specific ones may also be added. Instructions on how to do this, as well as further information, can be found in the file itself.

To post process fragments via the `rotatable_bond_override.mol2` file click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and switch on the **Use Rotatable Bond Override File** check box. Then, either enter the path and filename of the file or click on the **...** button and use the file selection window to choose the file.

This option is particularly useful if further control is sought over more than one ligand with a common substructure in a ligand library file.

The new bond type(s) are specified in the `rotatable_bond_override.mol2` file, in the `@<TRIPOS>COMMENT` part of the molecule file. The following format should be used:

```
RESET_BOND_TYPE <bond_number> <fix | flip | 1 | am>
```

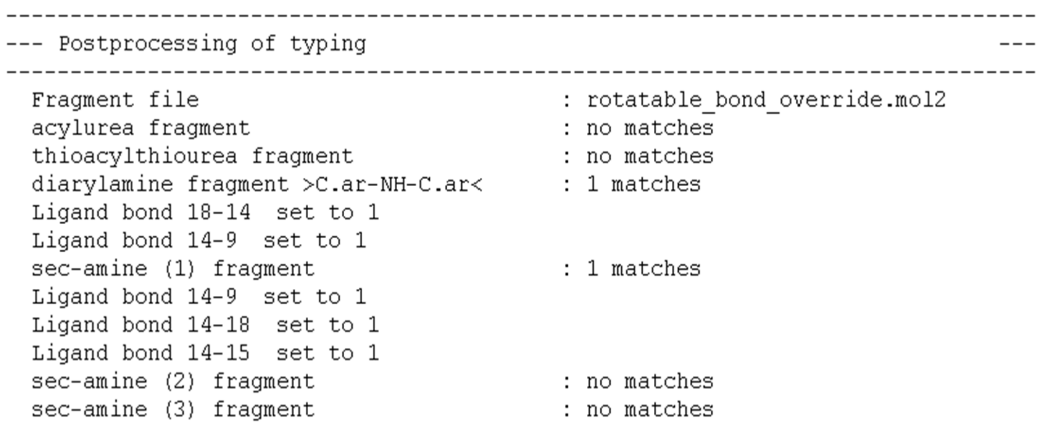
`fix` keeps the bond at its input angle. This option can also be specified for a single ligand docking via the `gold.conf` (see Fixing Rotatable Bonds at Their Input Conformation).

`flip` causes 180 degree turns of the input angle geometry.

`1` re-types the bond to a single bond, thus it is treated as fully rotatable.

`am` re-types the bond as an amide bond.

A report detailing what has been matched can be found in the `gold_ligand.log` file:



```
-------------------------------------------------------------------------------
--- Postprocessing of typing                                                 ---
-------------------------------------------------------------------------------
  Fragment file                           : rotatable_bond_override.mol2
  acylurea fragment                       : no matches
  thioacylthiourea fragment               : no matches
  diarylamine fragment >C.ar-NH-C.ar<     : 1 matches
  Ligand bond 18-14  set to 1
  Ligand bond 14-9  set to 1
  sec-amine (1) fragment                  : 1 matches
  Ligand bond 14-9  set to 1
  Ligand bond 14-18  set to 1
  Ligand bond 14-15  set to 1
  sec-amine (2) fragment                  : no matches
  sec-amine (3) fragment                  : no matches
```

If using the postprocess instruction and rotatable bond override file, the geometry is overruled whether the associated fitness flag is on or off.

If a torsion distribution can be found and matched, this will be used to bias the geometry of the re-typed bond.

Care should be taken to ensure the correct substructure is defined in the `rotatable_bonds_override.mol2` file. If a substructure cannot be matched, the bond override will not be used.

9.9 Fixing Rotatable Bonds at Their Input Conformation

GOLD was designed to dock flexible ligands into protein binding sites. However, sometimes it can be useful to fix the geometry of part or all of the ligand, e.g. in order to study the possible binding of a pre-determined ligand geometry.

To fix rotatable bonds at their input conformation click on **Ligand Flexibility** from the list of **Global Options** given on the left of the **GOLD Setup** window and switch on the **Fix Ligand Rotatable Bonds** check box. The following options are then available:

To fix all rotatable bonds in the ligand at their input conformation, select the **fix all** button.

To fix all non-terminal rotatable bonds (i.e. not -CH3, -OH, etc.), select the **fix all but terminal** button.

To fix the rotatable bond between two specified atoms select **fix specific**, then click on the **Specify Bonds** button. The resulting **Select Ligand Bonds To Fix** dialogue allows you to select the bond(s) you wish to fix. To select a bond, hit **Add** then either select the bond by clicking on it in the visualiser, or by entering the bond atom indices directly. Multiple rotatable bonds can be specified. Click on **Delete** to remove a bond from the list. Once you are satisfied with your selections click on **Close**.

The ability to fix rotatable bonds in the ligand at their input conformations is also available using the `rotatable_bond_override.mol2` file (see Overriding Automatic Bond Settings). This is particularly useful if docking a library of ligands that have a common substructure rather than the method above which is more suitable when docking an individual ligand.

Note: When fixing all rotatable bonds at their input conformation (i.e. performing a rigid ligand docking) GOLD will try to find the best orientation of the ligand in the binding site by mapping donor-acceptor (as well as hydrophobic-hydrophobic) fitting points. However, GOLD will not perform a local optimisation (simplex) on the final solution. This may lead to penalisation of near-optimal conformations. Minimising ligands using the CSD Conformer Generator before docking will help to take the ligand close to its local potential-energy minimum.

10 Ligand Search Options

10.1 Internal Energy Offset

Click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window. The **Use the internal ligand energy offset** check-box is switched on by default.

Enabling this option results in the internal energy terms (`internal torsion`, `internal vdw`, and `internal Hbond`) being corrected according to the best energy encountered for these terms during the run.

By applying this correction, the internal energy will be calculated with respect to that of a close to optimal non-bound structure, thereby taking into account any irreducible internal energy.

For each scoring function the ligand energy correction value is written to the docked solution files in the tag `<Gold.<scoring_function>.Internal.Correction>`. This is the best (i.e. minimum energy) value encountered.

For all scoring functions the best value encountered is subtracted from the ligand score (or energy) value before being passed to the final energy term.

The `.rnk` file is corrected at the end of a run with the best energy encountered after all docking attempts on a particular ligand (individual solution files are not). Therefore, you may observe small deviations for the best energy found between the solutions and rank file. Increasing the number of dockings or the number of GA operations in each docking will result in the discrepancy being less pronounced.

10.2 Hydrophobic Fitting Points

GOLD automatically calculates a list of hydrophobic fitting points in the binding site. These are used during the generation of trial docking solutions to map hydrophobic ligand atoms into favourable regions of the binding site.

GOLD generates its hydrophobic fitting points by placing a fine grid over the binding site. At each grid position, the van der Waals interaction energy between a bare carbon atom and the protein is evaluated. By default, positions at which the interaction energy is below -2.5 kcal/mol are added to the list of fitting points. The potential and threshold for selecting fitting points can be changed by editing the `gold.params` file and changing the values of `INTERNAL_POTENTIAL_FITPTS` and `E_FITPT_THRESHOLD`.

In this way, a map is constructed that contains positions onto which the placement of a hydrophobic ligand atom should be favourable.

The ligand fitting points are used for the matching of hydrophobic regions.

By default, only carbon atoms in the ligand are considered when identifying fitting points. The selection of suitable ligand atoms can be extended to include carbon, halogen and non-polar sulfur atoms by uncommenting the following line in the `gold.params` file:

```
#LIGAND_FITPTS_SELECTION EXTENDED_HAL_S
```

During docking, GOLD selects a list of lipophilic ligand atoms and matches them onto a subset of the hydrophobic fitting points.

It is possible to use customised hydrophobic fitting points. This might be appropriate if GOLD is not giving good results on a particular protein and you suspect that the fault may lie in the placement of hydrophobic ligand groups.

Customised fitting points must be supplied in a mol2 format file that contains a list of dummy atoms at the desired fitting-point locations. The supplied fitting points should sample all regions of interest in the cavity, so that the docking algorithm has sufficient alternatives for placement of hydrophobic ligand atoms within the cavity. GOLD uses gridded points that are spaced by 0.25 Å; for a speed-up in calculation, higher values could be used.

To make GOLD use a customised fitting-point file, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and enable the **Read hydrophobic fitting points** check-box. Then, either enter the path and filename of the mol2 file or click on the **...** button and use the file selection window to choose the file.

Customised fitting points can, for example, be generated by the CCDC program SuperStar, which offers the possibility of writing out a file of GOLD fitting points in the appropriate format (see SuperStar manual sections on `SAVE_GOLD_FITTING_POINTS` and `GOLD_MIN_PROPENSITY`).

10.3 Generating Diverse Solutions

There are occasions when GOLD obtains a number of docking solutions for a particular ligand which are very similar. If the **Allow early termination** option is activated, GOLD may obtain a user-defined number of ligands within the allowed RMSD very quickly.

Although this may not always be a problem there are occasions where it is apparent that none of the solutions are correct.

If this happens GOLD can be set up so that a number of different, diverse solutions can be generated.

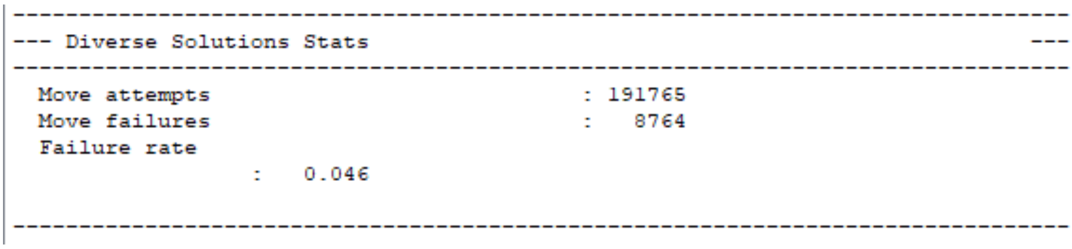
10.3.1 Method Used to Generate Diverse Solutions

Diversity is enforced during the ligand mapping stage. As the ligand is constructed and mapped into the binding site, GOLD checks the RMSD of the heavy atoms of the current solution against those that have already been generated.

If the RMSD is below the diversity threshold or the maximum number of solutions per cluster has been reached, the mapping is rejected and the process repeated until an acceptable solution is generated. GOLD keeps track of any failures; once the failure threshold has been reached the diverse solutions code is switched off.

The failure threshold is checked once the diverse solutions code has been called a thousand times. After that if the ratio of the number of failures to the number of times the code is called (i.e. the number of attempts) is greater than 0.2 then the diverse solutions code is switched off.

After each GA run the `Diverse Solutions Stats` are printed in the ligand log file, e.g.



```
--------------------------------------------------------------------------------
--- Diverse Solutions Stats                                                  ---
--------------------------------------------------------------------------------
  Move attempts                            : 191765
  Move failures                            :   8764
  Failure rate
                    :   0.046

--------------------------------------------------------------------------------
```

As the run progresses, the failure rate will (probably) increase for each subsequent solution as it becomes increasingly more difficult to generate diversity.

10.3.2 Setting Up GOLD to Generate Diverse Solutions

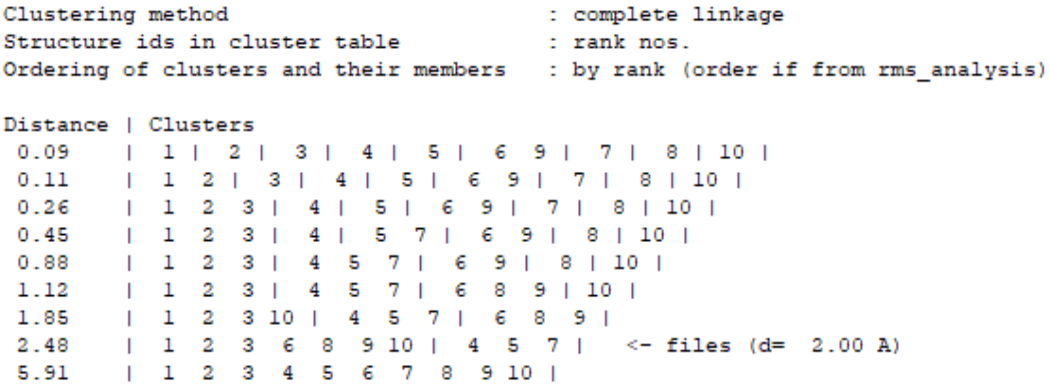
To generate diverse solutions for a docking run, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, then enable the **Generate diverse solutions** check box.

To specify the diversity criterion, click on the **Diverse Solution Options** button. The resulting **Diverse Solutions Options** dialogue enables the following two criteria to be specified:

Cluster size: the default is 1. Use this to specify how many ligand diverse solutions are contained in a cluster within a user-defined RMSD.

R.M.S.D.: the default is 1.5 Å. Use this setting to define the heavy atom RMSD cut-off (in Å) for determining if diverse solutions are in the same cluster or not.

The `ligand.log` output file contains information on which ligands are in which cluster at a particular RMSD cutoff. For example, the **Cluster size** was 3 and the **R.M.S.D.** setting was 1.5 Å in the docking below.



```
  Clustering method                         : complete linkage
  Structure ids in cluster table            : rank nos.
  Ordering of clusters and their members    : by rank (order if from rms_analysis)

  Distance | Clusters
   0.09    |  1 |  2 |  3 |  4 |  5 |  6  9 |  7 |  8 | 10 |
   0.11    |  1  2 |  3 |  4 |  5 |  6  9 |  7 |  8 | 10 |
   0.26    |  1  2  3 |  4 |  5 |  6  9 |  7 |  8 | 10 |
   0.45    |  1  2  3 |  4 |  5  7 |  6  9 |  8 | 10 |
   0.88    |  1  2  3 |  4  5  7 |  6  9 |  8 | 10 |
   1.12    |  1  2  3 |  4  5  7 |  6  8  9 | 10 |
   1.85    |  1  2  3 10 |  4  5  7 |  6  8  9 |
   2.48    |  1  2  3  6  8  9 10 |  4  5  7 |   <- files (d=  2.00 A)
   5.91    |  1  2  3  4  5  6  7  8  9 10 |
```

It is recommended that the **Allow early termination** tick box is disabled in the GOLD front end when generating diverse solutions (see Early Termination).

It is possible to generate links to the top ranked solution from each distinct cluster at a given RMSD cutoff (see Identification of Different Binding Modes (Clustering of Ligand Poses)).

11 Setting Constraints

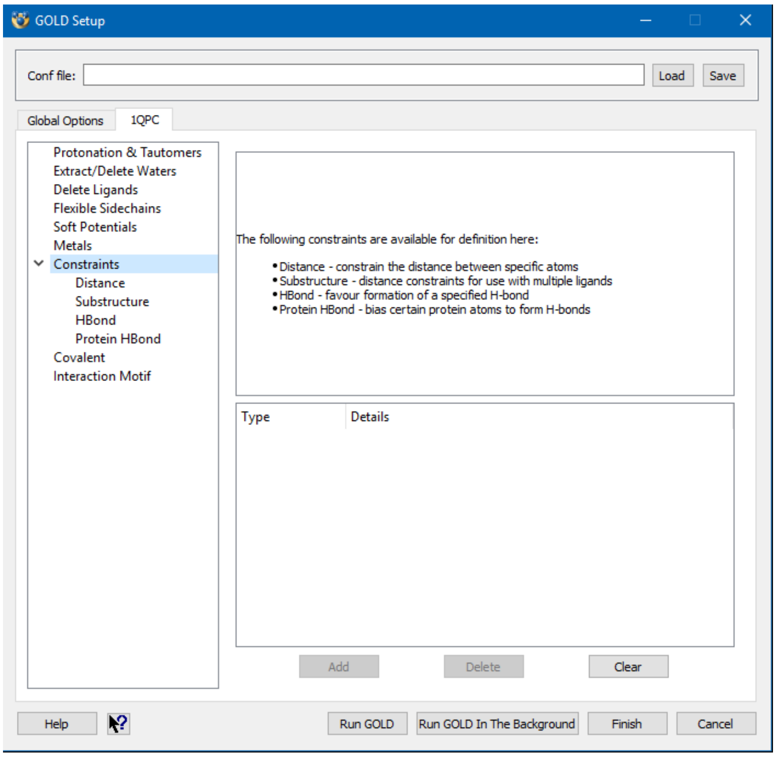
11.1 Using the Constraint Editor

Depending on what sort of contraint(s) are required they may be protein-specific or applied to a protein ensemble.

Protein-specific constraints are the following:

- Distance constraint, for use with individual ligands (see Distance Constraints).
- Substructure based distance constraint, for use with multiple ligands that have a common substructure or functional group (see Distance Constraints).
- Hydrogen bond constraint, for specifying a hydrogen bond between a particular ligand atom and a particular atom in the protein (see Hydrogen Bond Constraints).
- Protein hydrogen bond constraint, for specifying that a particular protein atom should be hydrogen-bonded to the ligand, but without specifying to which ligand atom (see Hydrogen Bond Constraints).

To apply one of the above constraints, it is necessary to first click on the appropriate protein tab (e.g. **1QPC** below) adjacent to the **Global Options** tab. To define a constraint, select a constraint type from those listed on the left of the **GOLD Setup** window. If individual constraint types are not visible, click on the “**>**” icon next to **Constraints** to expand the list of options.



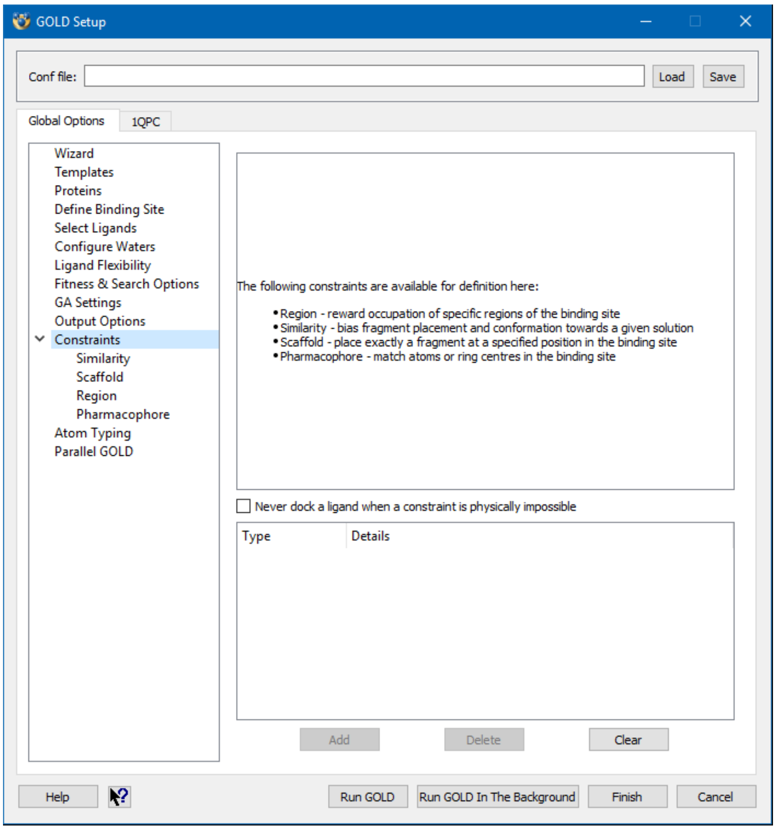
The following constraints are available for definition here:

- Distance - constrain the distance between specific atoms
- Substructure - distance constraints for use with multiple ligands
- HBond - favour formation of a specified H-bond
- Protein HBond - bias certain protein atoms to form H-bonds

Constraints that are applicable to an individual protein or protein ensemble are the following:

- **Region (hydrophobic) constraint**, for biasing the docking towards solutions in which particular regions of the binding site are occupied by specific ligand atoms or types of ligand atom (see Region (Hydrophobic) Constraints).
- **Template similarity constraint**, for biasing the conformation of docked ligands towards a given solution or template (see Similarity Constraints).
- **Scaffold constraint**, to place a ligand fragment at an exact specified position in the binding site (see [Scaffold Match Constraint](#scaffold-match-constraint)).
- **Pharmacophore constraint**, to match specific types of atoms or ring centres in the binding site (See Pharmacophore Constraints).

To define one of the above constraints, ensure the **Global Options** tab is selected then pick a constraint type from those listed on the left of the **GOLD Setup** window. If individual constraint types are not visible, click on the “**>**” icon next to **Constraints** to expand the list of options.



The following constraints are available for definition here:

- Region - reward occupation of specific regions of the binding site
- Similarity - bias fragment placement and conformation towards a given solution
- Scaffold - place exactly a fragment at a specified position in the binding site
- Pharmacophore - match atoms or ring centres in the binding site

☐ Never dock a ligand when a constraint is physically impossible

For all constraints the constraint editor is present at the bottom of each constraint setup page. Once the settings for a constraint have been specified click on the **Add** button to add the constraint definition to the list of defined constraints. Repeat this procedure if you want to specify additional constraints.

To edit a constraint, highlight the corresponding entry in the list, make the required change and then hit the **Add** button.

To remove a constraint from the list, highlight the entry and hit the **Delete** button, or to remove all defined constraints hit the **Clear** button.

It is possible to instruct GOLD not to dock ligands when the specified constraint is physically impossible to satisfy (e.g. if no suitable group is present in the ligand to form the required H-bond constraint). This is done by selecting the **Never dock a ligand when a constraint is physically impossible** check box.

When using constraints GOLD will be biased towards finding solutions in which the specified constraint is satisfied. However, it is important to remember that such a solution is not guaranteed (i.e. it is not possible to force a constraint to be satisfied in the final solution).

11.2 Distance Constraints

Distance constraints are applicable to individual protein-ligand complexes (i.e. these must be set up individually for each protein-ligand complex if performing ensemble docking).

Any distance between a ligand and protein atom (or between an extracted cofactor and protein atom, or a ligand and cofactor retained within the protein, or two ligand atoms, or two protein atoms, or two extracted cofactor atoms) can be constrained to lie between minimum and maximum distance bounds. Note that typically cofactors are not extracted and instead are retained within the protein structure, such that they are considered part of the protein for setting up constraints. GOLD features two types of distance constraint:

- A standard distance constraint for use with individual ligands (see Setting Up a Distance Constraint).
- A substructure-based distance constraint for use with multiple ligands which have a common functional group (see Setting Up Substructure-Based Distance Constraints).

11.2.1 Setting Up a Distance Constraint

A distance between a specified ligand and protein atom (or between an extracted cofactor and protein atom, or a ligand and cofactor retained within the protein, or two ligand atoms, or two protein atoms, or two extracted cofactor atoms) can be constrained to lie between minimum and maximum distance bounds.

During a GOLD run, if a constrained distance is found to lie outside its bounds, a spring energy term is used to reduce the fitness score, i.e.

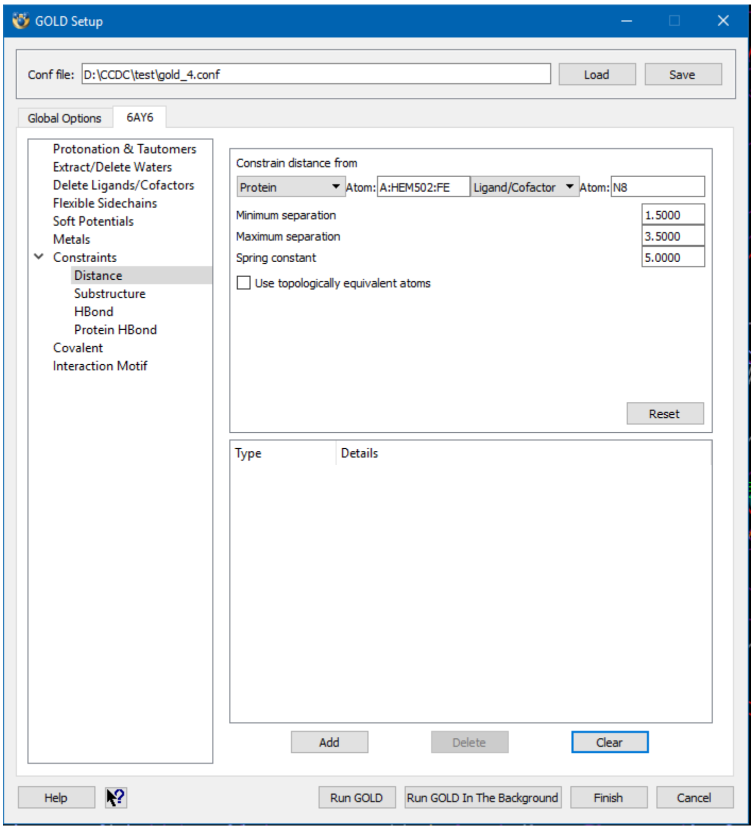
$E = kx^2$ where: x is the difference between the distance and the closest constraint bound; k is a user-defined spring constant.

To set up a distance constraint you must first select the appropriate protein tab, adjacent to the **Global Options** tab.

Select **Distance** from the list of **Global Options** given on the left of the **GOLD Setup** window. If this option is not visible, click on the "**>**" icon next to **Constraints** to expand the list of options.

Specify the atoms to be used in the distance constraint and whether they belong to the protein or ligand/cofactor. This can be done by clicking on an atom in the visualiser (the atom structure type, i.e. protein or ligand/cofactor will be updated automatically upon selection). Alternatively, you can enter the atom label directly into the appropriate entry box. When specifying protein atom labels it is necessary to also set the chain and the residue they belong to (e.g. A:CYS430:SG where the atom SG belongs to residue CYS430 in protein chain A; or A:HEM502:FE where the atom FE belongs to heme HEM502 not extracted and considered part of protein chain A). Note that typically cofactors are not extracted and instead are retained within the protein structure, such that they are considered part of the protein for setting up constraints.

The maximum and minimum separation of the constrained atoms must be entered (distances are in Å), and the spring constant must also be specified. For example:



Conf file: D:\CCDC\test\gold_4.conf

Constrain distance from

Protein Atom: A:HEM502:FE Ligand/Cofactor Atom: N8

Minimum separation 1.5000

Maximum separation 3.5000

Spring constant 5.0000

□ Use topologically equivalent atoms

If the specified ligand atom is topologically equivalent to other atoms in the ligand (e.g. it is one of the oxygen atoms of an ionised carboxylate group), then GOLD will compute the constraint term using whichever of the equivalent atoms gives the best value automatically, as long as the **Use topologically equivalent atoms** check-box is ticked.

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor).

11.2.2 Method Used for Substructure-Based Distance Constraints

Substructure-based constraints are applicable to individual protein-ligand complexes (i.e. must be set up individually for each protein-ligand if performing ensemble docking).

It is possible to apply a distance constraint to multiple ligands which have a common functional group.

The constraint forces GOLD to limit the distance between a protein atom and one atom of this functional group. Docking solutions will be biased towards the specified distance range. Note that typically cofactors are not extracted and instead are retained within the protein structure, such that they are considered part of the protein for setting up constraints.

During docking the constraint will be applied to any ligands which contain the specified substructure (matching is performed on the basis of the element types and 2D connectivity) and the resulting solutions will be biased towards the specified distance range. GOLD always accounts for topology in the substructure.

Note: The substructure must be a sub-graph rather than a complete molecule.

As with normal distance constraints (see Setting Up a Distance Constraint), the score is reduced for unfavourable ligand solutions. The amount of decrease in the score is determined by a weight term that the user must supply (i.e. the spring constant).

11.2.3 Setting Up Substructure-Based Distance Constraints

To use a substructure-based distance constraint, first create a file containing the substructure in mol2 format (e.g. `substructure.mol2`). The actual conformation of the group in this file is not important, as only the element types and 2D connectivity will be used.

To set up a distance constraint you must first select the appropriate protein tab, adjacent to the **Global Options** tab.

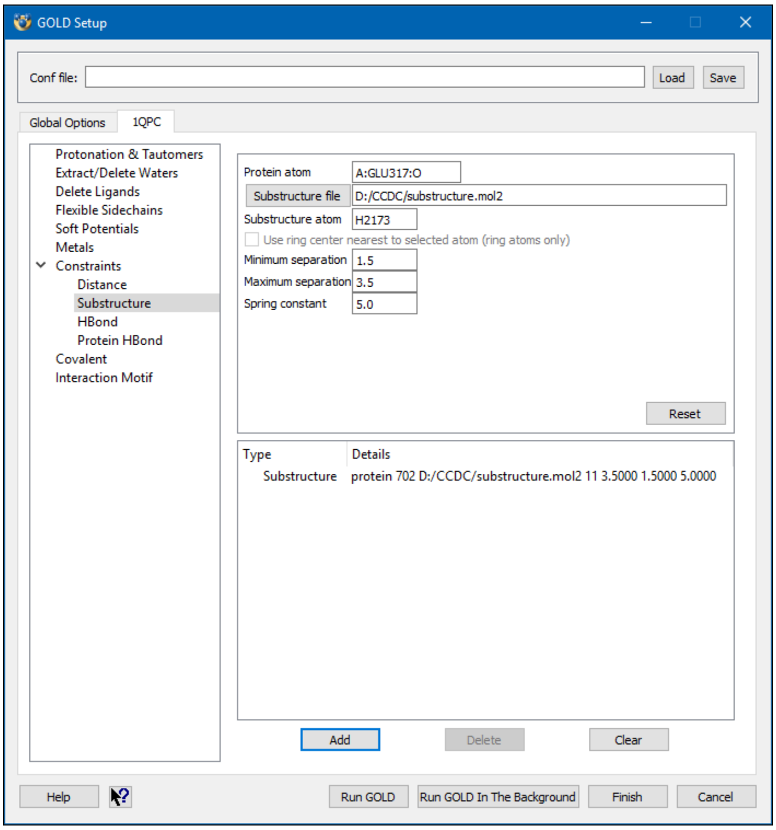
To constrain a distance, click on **Distance** from the list of **Global Options** given on the left of the **GOLD Setup** window. If this option is not visible, click on the “**>**” icon next to **Constraints** to expand the list of options.

To specify the **Substructure file** either enter the path and filename of the file or click on the **Substructure file** button and use the file selection window to choose the file.

Specify the **Protein atom** and **Substructure atom** to be used in the distance constraint. This can be done by clicking on an atom in the visualiser. Alternatively, you can enter the atom label directly into the appropriate entry box. For protein atoms, chains and residue identifiers should also be specified (e.g. A:CYS430:SG). Note that typically cofactors are not extracted and instead are retained within the protein structure, such that they are considered part of the protein for setting up constraints.

Specify the allowed range of separation by entering a **Maximum separation** and a **Minimum separation** (distances are in Å).

Enter the spring constant (i.e. the weight of the term). This causes a spring-based distance constraint to be added for the specified substructure atom and protein atom. The weight specifies the spring energy term; usually, a weight in the range of 5 to 10 will work well.



Protein atom: A:GLU317:O
Substructure file: D:/CCDC/substructure.mol2
Substructure atom: H2173
□ Use ring center nearest to selected atom (ring atoms only)
Minimum separation: 1.5
Maximum separation: 3.5
Spring constant: 5.0

Type	Details
Substructure	protein 702 D:/CCDC/substructure.mol2 11 3.5000 1.5000 5.0000

It is possible to define a distance constraint from a centroid of a ring in the ligand. To do this, specify an atom within the ring of interest and enable the **Use ring center nearest to selected atom (ring atoms only)** check-box. The closest ring centre to the selected atom will be used. Note: When defining a distance constraint involving a ring centre, ensure that the maximum and minimum separations are adjusted accordingly.

If the constraint refers to a substructure atom (and therefore a ligand atom) which is topologically equivalent to other atoms (e.g. it is one of the oxygen atoms of an ionised carboxylate group), GOLD will automatically compute the constraint term using whichever of the equivalent atoms gives the best value.

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor).

11.3 Hydrogen Bond Constraints

Two types of hydrogen bond constraints may be specified:

- A hydrogen bond constraint (see Setting Up Hydrogen Bond Constraints), which can be used to force a hydrogen bond between a particular protein atom and a particular ligand atom.
- A protein hydrogen bond constraint (see Setting up Protein H Bond Constraints), which can be used to specify that a particular protein atom should be hydrogen-bonded to the ligand, but without specifying to which ligand atom.
- Note that typically cofactors are not extracted and are retained within the protein structure, such that they are considered part of the protein for setting up constraints.

11.3.1 Setting Up Hydrogen Bond Constraints

Hydrogen bond constraints are applicable to individual protein-ligand complexes (i.e. must be set up individually for each protein-ligand complex if performing ensemble docking).

A ligand atom may be constrained to form a hydrogen bond to a particular protein atom. One atom should be a donatable hydrogen atom (you must specify the hydrogen atom, not the O or N atom to which it is attached) and the other should be an acceptor. The protein atom should be available for ligand binding (i.e. solvent accessible). Note that typically cofactors are not extracted and instead are retained within the protein structure, such that they are considered part of the protein for setting up constraints. Note that this constraint does not work with metals.

The constraint is incorporated into the least-squares fitting routine used by GOLD. Thus, when least-squares fitting is used to dock the ligand (by attempting to form hydrogen bonds encoded within the chromosome) the constraint is added to the least-squares mapping. The constraint has a weight of 5 relative to a normal hydrogen bond taken from the chromosome.

To set up a distance constraint you must first select the appropriate protein tab, adjacent to the **Global Options** tab.

To define a hydrogen bond constraint, click on **HBond** from the list of options given on the left of the **GOLD Setup** window. If this option is not visible, click on the "**>**" icon next to **Constraints** to expand the list of options.

Specify the atoms to be used in the constraint. This can be done by clicking on an atom in the visualiser. Alternatively, you can enter the atom label directly into the appropriate entry box. For protein atoms, the label must include information about the chain and the residue (e.g. A:HIS3839:ND1).

The hydrogen bond constraint weighting can be altered within the `# FITNESS FUNCTION` section of the GOLD parameters file by changing the value of the parameter `CONSTRAINT_WT`**.**

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor).

11.3.2 Method Used for Protein H Bond Constraints

Protein H bond constraints are applicable to individual protein-ligand complexes (i.e. must be set up individually for each protein-ligand complex if performing ensemble docking).

GOLD will be biased towards finding solutions in which the specified protein atoms form hydrogen bonds. The fitness score of a given docking will be penalised by a user-specified value c for every protein H-bond constraint that is not satisfied (i.e. for every protein atom that you have specified should form a hydrogen bond but does not).

GOLD assesses the geometry of each required hydrogen bond on a scale of 0 to 1, with 1 denoting perfect. If this geometry weight for the constrained Hbond falls below the **Minimum H-bond geometry weight** specified by the user, a penalty will be applied to the score for the unfulfilled hydrogen bond, i.e. it will not be considered to be an H-bond and will therefore contribute a penalty to the fitness score. The magnitude of this penalty is equal to the weight specified for the constraint.

Each trial ligand docking in a genetic algorithm run is generated by a least-squares fit of mapping points (H-bonding or hydrophobic binding points on the protein with complementary points on the ligand). The inclusion of a protein H-bond constraint will ensure that at least one of the specified protein atoms is included as one of the mapping points, i.e. use of the specified points is enforced at the mapping stage of the algorithm.

11.3.3 Setting up Protein H Bond Constraints

A protein hydrogen bond constraint can be used to specify that a particular protein atom should be hydrogen-bonded to the ligand, but without specifying to which ligand atom.

To set up a distance constraint you must first select the appropriate protein tab, adjacent to the **Global Options** tab.

To define a protein hydrogen bond constraint, click on **Protein HBond** from the list of options given on the left of the **GOLD Setup** window. If this option is not visible, click on the "**>**" icon next to **Constraints** to expand the list of options.

Specify the protein atoms to be used in the constraint. This can be done by clicking on an atom in the visualiser. Alternatively, you can enter the atom label including chain and residue identifier (e.g. A:HIS3839:ND1) directly into the appropriate box.

Either a donatable hydrogen atom (you must specify the hydrogen atom, not the O or N atom to which it is attached) or an acceptor can be specified. The protein atom should be available for ligand binding (i.e. solvent accessible). Note that typically cofactors are not extracted and instead are retained within the protein structure, such that they are considered part of the protein for setting up constraints. Note that this constraint does not work with metals.

The **Constraint weight** is the strength of bias applied to the formation of a specified hydrogen bond in the least squares mapping algorithm within GOLD. The **Constraint weight** is also the value of the penalty applied to the fitness score for each constrained H bond that is not formed.

The **Minimum H-bond geometry weight** is a user defined score that determines how good a hydrogen bonding interaction has to be in order for it to be considered a hydrogen bond by GOLD. The **Minimum H-bond geometry weight** takes a range of values from 0 to 1; by default this value is set at 0.005.

For a given protein H bond constraint more than one protein atom label can be entered in the **Protein atom(s) required to form H-bond** entry box. This will instruct GOLD to use an either-or type of constraint during docking. For example, specifying two protein atoms, acceptor m and acceptor n, will result in the constraint being satisfied if an H bond is formed to either m or n during docking. This is of use when defining constraints involving, for example, carboxylates where it is not important which oxygen atom forms an H bond, provided that one of them does.

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor). It is possible to specify several different protein H bond constraints, with different weights for each constraint.

11.4 Region (Hydrophobic) Constraints

This constraint can be used to bias the docking towards solutions in which particular regions of the binding site are occupied by specific ligand atoms (or types of ligand atom, e.g. hydrophobic atoms).

11.4.1 Method Used for Region (Hydrophobic) Constraints

This constraint can be used to bias the docking towards solutions in which particular regions of the binding site are occupied by specific ligand atoms (or types of ligand atom).

For each region (hydrophobic) constraint specified a sphere is placed at an explicitly-defined position within the binding site. Each sphere is assigned a user-defined radius, so a sphere can be adjusted if required, e.g. to fill an entire pocket in the binding-site. Minimum settable radius is 0.5 Å.

A contribution (determined according to a user-specified weighting) is then added to the score for each specified non-hydrogen ligand atom that lies within the designated sphere. A contribution is added to the score for each atom located within the sphere (i.e. the total contribution will depend on the number of atoms found in the region of interest and ultimately the ligand-accessible volume of the region).

The ligand atoms used in the constraint can be specified explicitly. Alternatively, it is possible to use all hydrophobic ligand atoms, or to use only those hydrophobic atoms in aromatic rings. Atoms considered to be hydrophobic include:

- Carbon atoms bound to at least two H or C atoms.
- Atoms typed C.cat.
- Atoms typed S.3 and bound to two carbons.
- H atoms bound to an sp^2, sp^3 or aromatic carbon (Note: Only heavy atoms found within the sphere will contribute to the score.)

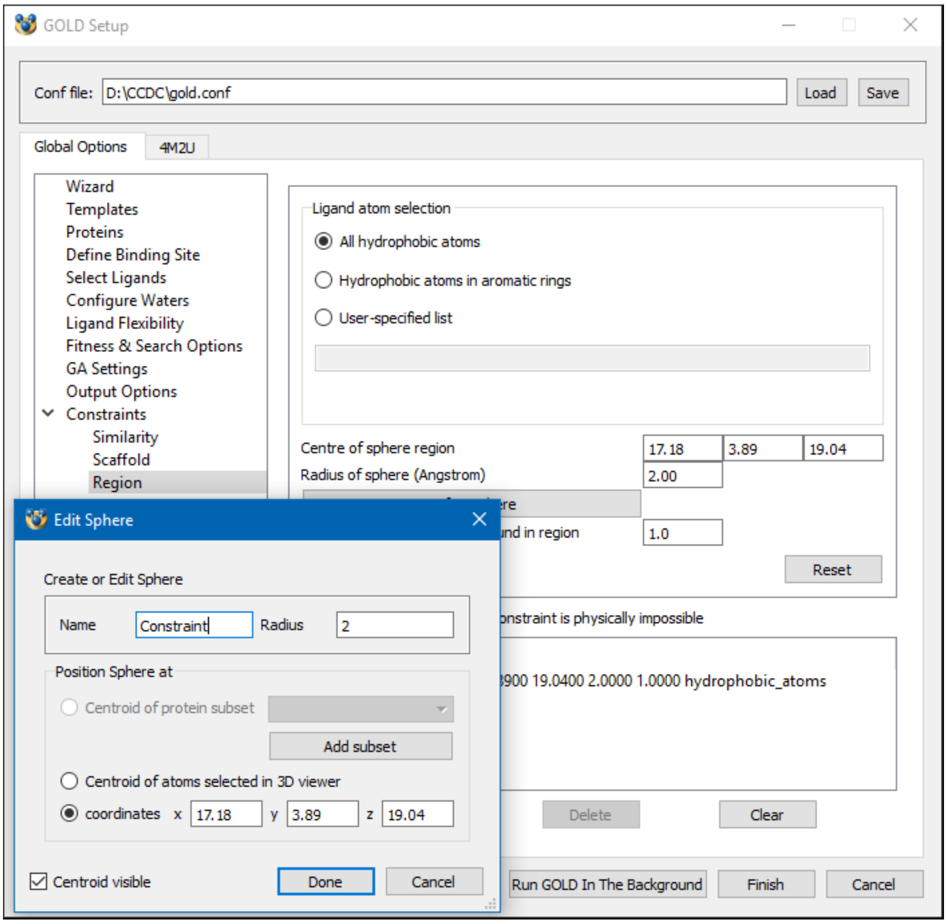
Details of the region (hydrophobic) constraint calculation, including the final contribution to the fitness score, are given in the ligand log file (see Ligand Log File).

11.4.2 Setting Up Region (Hydrophobic) Constraints

To define a region (hydrophobic) constraint, click on **Region** from the list of **Global Options** given on the left of the **GOLD Setup** window. If this option is not visible, click on the “**>**” icon next to **Constraints** to expand the list of options.

Specify the ligand atoms to be used in the constraint by selecting either **All hydrophobic atoms**, **Hydrophobic atoms in aromatic rings**, or **User-specified list**. If **User-specified list** is selected, then individual ligand atoms can be selected by clicking on them in the visualiser (you may need to first hide the sphere using the **Centroid visible** checkbox in the **Edit Sphere** dialogue). Alternatively, you can enter the atom numbers (as it appears in the input file) directly into entry box.

Next, specify the position and radius of the sphere. To do this, click on the **Define sphere** button, this will launch the **Edit Sphere** dialogue:



Conf file: D:\CCDC\gold.conf

Ligand atom selection

● All hydrophobic atoms

○ Hydrophobic atoms in aromatic rings

○ User-specified list

Centre of sphere region 17.18 3.89 19.04

Radius of sphere (Angstrom) 2.00

nd in region 1.0

onstraint is physically impossible

900 19.0400 2.0000 1.0000 hydrophobic_atoms

Create or Edit Sphere

Name Constraint Radius 2

Position Sphere at

○ Centroid of protein subset

○ Centroid of atoms selected in 3D viewer

● coordinates x 17.18 y 3.89 z 19.04

☑ Centroid visible

Enter a name and the radius of the sphere (distances are in Å).

The sphere must then be positioned within the binding site, this can be done in a number of ways:

- The sphere can be positioned on the centroid of an existing subset of protein atoms. Select **Centroid of protein subset** and select the protein subset from the drop-down list. To create a new subset of protein atoms, click on the **Add subset** button. Instructions on defining protein subsets can be found in the Hermes User Guide.
- The sphere can be positioned on the centroid of selected atoms. Select **Centroid of atoms selected in 3D viewer**, then within Hermes click on one or more protein atoms in order to define a centroid.
- Alternatively, select **Coordinates** and enter the orthogonal x,y,z coordinates of a single point upon which to position the sphere.

Click on **Done** in the **Edit Sphere** dialogue once the sphere has been defined.

A score contribution must also be specified. This is the value that will be added to the fitness score for each specified non-hydrogen ligand atom found within the sphere region. The total contribution added will therefore depend on the number of atoms located within the sphere.

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor). It is possible to define multiple region (hydrophobic) constraints.

11.5 Pharmacophore Constraints

As the region constraint, the pharmacophore constraint can be used to bias the docking towards solutions in which particular regions of the binding site are occupied by specific types of ligand atom (e.g. hydrogen-bond donor, hydrogen-bond acceptor, ring centre).

11.5.1 Method Used for Pharmacophore Constraints

For each pharmacophore point specified a sphere is placed at an explicitly-defined position within the binding site. Each sphere is assigned a user-defined radius, so it can be adjusted if required to better represent the feature’s type. By default, the radius is set to 0.7 Å.

Five types of pharmacophore point can be defined: H-bond acceptor, H-bond donor, H-bond donor or acceptor, aromatic ring centre and ring centre. The pharmacophore sphere will be placed on the atom coordinates of H-bond donor or acceptor atoms and at the centroid of a ring.

A contribution (determined according to a user-specified weighting) is then added to the score for each specified pharmacophore point. A contribution is added to the score if the distance between the pharmacophore point and an atom of the same type in the molecule is less than the specified radius of the sphere. If a pharmacophore constraint point is not matched in the molecule, it will not be used during scoring.

The point, by default, will also be used as a fitting point in the docking. You can select this in the GUI (it is on by default) and modify the weight used in fitting. A high weight will mean dockings will be strongly biased towards the pharmacophore point.

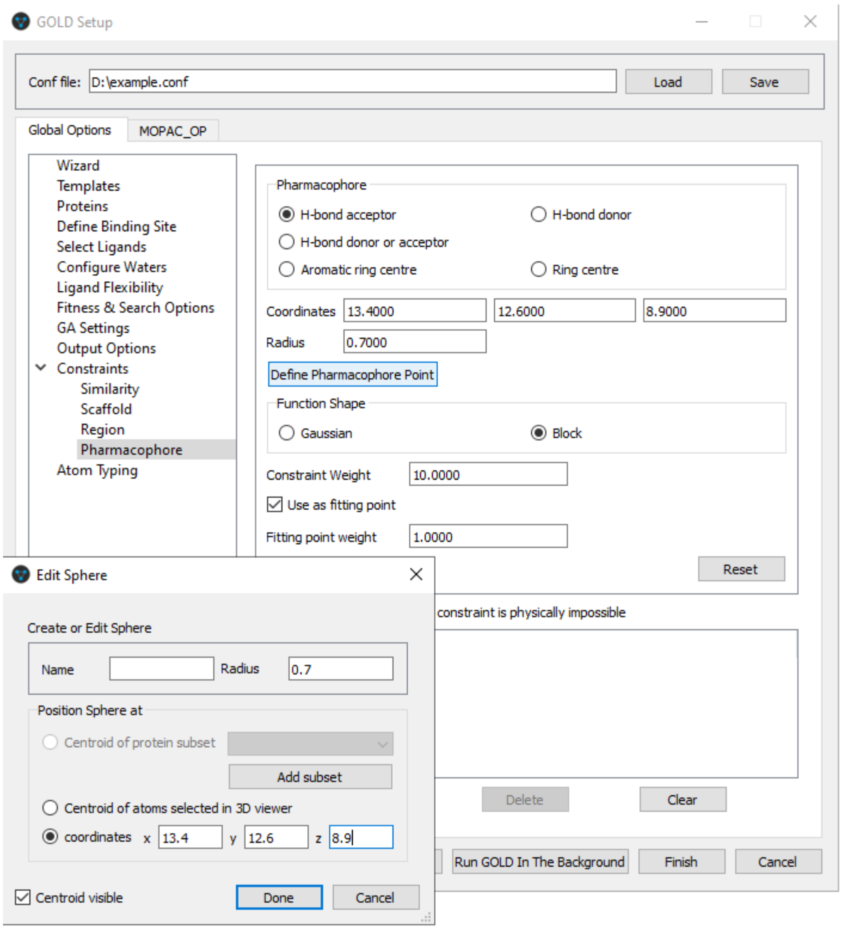
Details of the pharmacophore constraint calculation, including the final contribution to the fitness score, are given in the ligand log file (see Ligand Log File).

11.5.2 Setting Up a Pharmacophore Constraints

To define a pharmacophore constraint, click on **Pharmacophore** from the list of **Global Options** given on the left of the **GOLD Setup** window. If this option is not visible, click on the “**>**” icon next to **Constraints** to expand the list of options.

Specify the type of pharmacophore point to be used in the constraint by selected either **H-bond acceptor**, **H-bond donor**, **H-bond donor or acceptor**, **Aromatic ring centre** or **Ring Centre**.

Next, specify the position and radius of the sphere. To do this, click on the **Define Pharmacophore Point** button, this will launch the **Edit Sphere** dialogue:



Conf file: D:\example.conf

Pharmacophore

● H-bond acceptor ○ H-bond donor

○ H-bond donor or acceptor

○ Aromatic ring centre ○ Ring centre

Coordinates 13.4000 12.6000 8.9000

Radius 0.7000

Function Shape

○ Gaussian ● Block

Constraint Weight 10.0000

☑ Use as fitting point

Fitting point weight 1.0000

constraint is physically impossible

Name Radius 0.7

Position Sphere at

○ Centroid of protein subset

○ Centroid of atoms selected in 3D viewer

● coordinates x 13.4 y 12.6 z 8.9

☑ Centroid visible

Enter a name and the radius of the sphere (distances are in Å).

The sphere must then be positioned within the binding site, this can be done in a number of ways:

- The sphere can be positioned on the centroid of an existing subset of protein atoms. Select **Centroid of protein subset** and select the protein subset from the drop-down list. To create a new subset of protein atoms click on the **Add Subset** button. Instructions on defining protein subsets can be found in the Hermes user guide.

- The sphere can be positioned on the centroid of selected atoms. Select **Centroid of atoms selected in 3D viewer**, then within Hermes click on one or more protein atoms in order to define a centroid. This would be the recommended way of setting up a ring centre point.

- Alternatively, select **Coordinates** and enter the orthogonal x,y,z coordinates of a single point upon which to position the sphere.

Click on **Done** in the Edit Sphere dialogue once the sphere has been defined.

A pharmacophore weight must also be specified. This is the value that will be added to the fitness score for matched ligand atoms satisfying the constraint.

Finally, by default, GOLD will use your pharmacophore as afitting point. It is possible to alter the weight of the fitting point. The bigger this is the harder GOLD will try to find a pose that conforms to the point.

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor). It is possible to define multiple pharmacophore constraints.

11.6 Similarity Constraints

This constraint can be used to bias the conformation of docked ligands towards a given solution or template.

11.6.1 Method Used for Similarity Constraints

This constraint will bias the conformation of docked ligands towards a given solution. This solution, or template, can, for example, be another ligand in a known conformation, a common core (useful when docking ligands of a combinatorial set), or it may just be a large substructure that is expected, or known, to bind in a certain way.

The template must be supplied as a mol2 file.

Unlike the distance-based constraints, which reduce the score for ligands that adopt unfavourable orientations, this constraint will add an energy term to the score based on the similarity between the ligand being docked and the template provided. The similarity between the two is evaluated as a Gaussian overlap term.

The similarity constraint can be applied in three ways that differ in how the overlap between ligand and template is calculated. The similarity can be evaluated:

- Using the overlap between all donor atoms in the template and the ligand being docked.

- Using the overlap between all acceptor atoms in the template and the ligand being docked.

- Using the overlap of all atoms of the template (this can be regarded as a ligand-shape constraint).

The energy term to be added is calculated as similarity times weight (the similarity value is between 0 and 1, where 1 indicates identity of template and ligand).

If you wish to place a fragment at an exact specified position in the binding site, as opposed to biasing the docking, use the scaffold match constraint (see Scaffold Match Constraint).

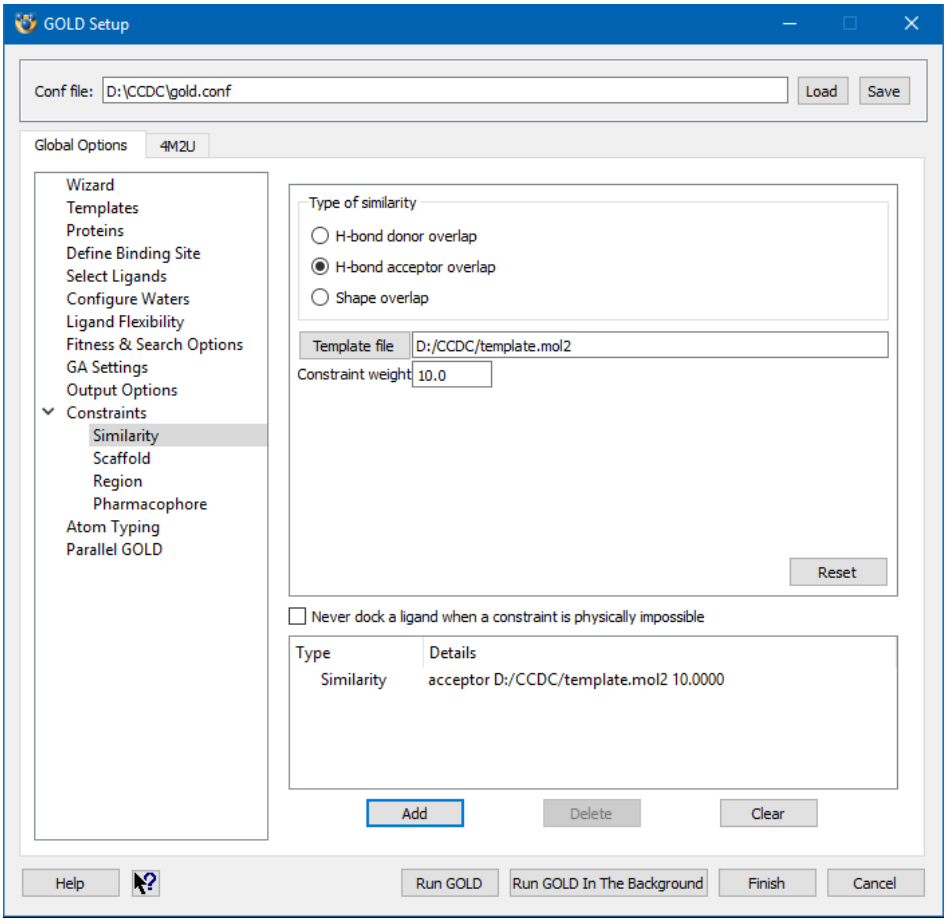
11.6.2 Setting Up a Similarity Constraint

To define a similarity constraint, click on **Similarity** from the list of **Global Options** given on the left of the **GOLD Setup** window. If this option is not visible, click on the **>** icon next to **Constraints** to expand the list of options.

Specify the similarity type to be used by selecting **H-bond donor overlap**, **H-bond-acceptor overlap**, or **Shape overlap** (see Method Used for Similarity Constraints).

The similarity template file should contain the template molecule or fragment in its docked position (i.e. expressed with respect to the same coordinate frame as the protein and with the coordinates required to place it in the correct pose). To specify the template file either enter the path and filename of the file or click on the **Template file** button and use the file selection window to choose the file.

The weight term determines the maximum energy term that would be added to the score in the case of perfect overlap between ligand and template. As an initial value for this term, we suggest a value between 5 and 30.



Conf file: D:\CCDC\gold.conf

Type of similarity

○ H-bond donor overlap

● H-bond acceptor overlap

○ Shape overlap

Template file: D:/CCDC/template.mol2

Constraint weight: 10.0

□ Never dock a ligand when a constraint is physically impossible

Type	Details
Similarity	acceptor D:/CCDC/template.mol2 10.0000

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor). It is possible to define multiple constraints, e.g. one for donors and one for acceptors.

11.7 Scaffold Match Constraint

The scaffold match constraint can be used to place a fragment at an exact specified position in the binding site. The geometry of the fragment will not be altered during docking.

11.7.1 Method Used for Scaffold Match Constraint

This constraint will attempt to a place a ligand onto a given scaffold location. The scaffold, can, for example, be a common core, a fragment (useful when docking ligands of a combinatorial set), or it may just be a substructure known to adopt a certain binding position.

It is advised that only those atoms required for scaffold matching are specified when using the scaffold constraint. Having a scaffold that almost exactly matches the docked ligand (and specifying a large number of atoms for matching) causes GOLD problems when it is generating random and unique individuals during docking.

The scaffold must be supplied as a mol2 file. The file should contain the scaffold fragment in its docked position (i.e. expressed in the same coordinate frame as the protein and with the coordinates required to place it in the correct pose). The element type is matched, not the atom type; thus, it is not essential for the SYBYL atom types to be correct in the scaffold mol2 file. It is recommended that the scaffold have hydrogens correctly placed on all appropriate atoms other than the unfulfilled valency at the substitution point, which must not be blocked by hydrogen.

Unlike the template similarity constraint, which will bias the docking by adding an energy term to the score based on the similarity between the ligand being docked and the template provided, this constraint is enforced at the mapping stage in GOLD. Ligand placements are generated using a best least-squares fit with the scaffold heavy atom positions, i.e. this constraint forces all atoms on the matching portion of the ligand to lie very close, or coincident, with the corresponding scaffold. There is no S(con) contribution to the fitness score to bias dockings.

How closely ligand atoms fit onto the scaffold is governed by a user-specified weight. Setting a higher weight will force the ligand to be placed onto the scaffold locations more strictly. A default weight of 5.0 is used. Setting a high weight can have a detrimental effect on the fitness score if the placement results in e.g. bad protein-ligand clashes. If desired, values below 1 can be used to achieve a more lenient overlay.

Symmetry effects (such as the flipping of a phenyl ring by 180 degrees) are not taken into account during matching of the ligand onto the scaffold. Therefore, a scaffold that will give a unique match should ideally be provided.

For a given ligand, it is not possible to match multiple scaffolds at the same time. Scaffolds are evaluated in the order supplied by the user and the scaffold that matches the ligand first will be used. This means that it is possible to specify two or more different scaffolds, and GOLD will use the scaffold that matches the ligand first. This can be useful when docking multiple different series of compounds.

11.7.2 Setting Up Scaffold Match Constraints

To define a scaffold constraint, click on **Scaffold** from the list of **Global Options** given on the left of the **GOLD Setup** window. If this option is not visible, click on the “**>**” icon next to **Constraints** to expand the list of options.

The scaffold template file should contain the scaffold molecule or fragment in its docked position (i.e. expressed with respect to the same coordinate frame as the protein and with the coordinates required to place it in the correct pose). To specify the scaffold file either enter the path and filename of the file or click on the **Scaffold file** button and use the file selection window to choose the file.

The **Constraint weight** determines how closely ligand atoms fit onto the scaffold. Setting a higher weight will force the ligand to be placed onto the scaffold locations more strictly.

By default, all heavy atoms in the supplied scaffold structure file will be used for matching. However, it is possible to specify only a subset of those atoms in the scaffold structure (these may include non-heavy atoms). Individual scaffold atoms can be specified by clicking on them in the visualiser. Alternatively, you can enter the atom numbers (as it appears in the scaffold file) directly into the entry box. Limiting the number of atoms to be matched can be useful for large, rigid scaffolds. In such a case, specifying only a few atoms distributed throughout the scaffold can be sufficient to obtain a good 3D superimposition.

Click on the **Add** button to add the constraint definition to the constraint editor (see Using the Constraint Editor).

11.8 Interaction Motif Constraint

This constraint can be used to bias the docking towards solutions that form particular protein-ligand binding motifs.

This constraint could be used e.g. when there is experimental evidence, such as a number of X-ray structures from a fragment screen, which show that certain combinations of interactions (motifs) are commonly formed by groups of fragment binders. Such motifs can therefore be considered as favourable and this information can be incorporated into the docking in order to bias the ligand poses that are generated.

For further information see: The Use of Protein-Ligand Interaction Fingerprints in Docking (see References).

11.8.1 Method Used for the Interaction Motif Constraint

Interaction motif constraints are applicable to individual protein-ligand complexes (i.e. must be set up individually for each protein-ligand complex if performing ensemble docking).

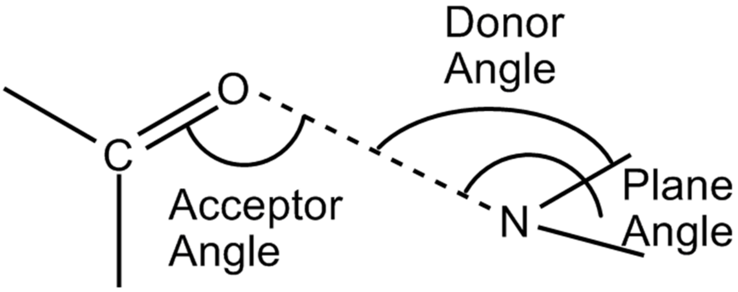
This constraint is used to bias the docking towards solutions that form particular protein-ligand binding motifs.

One or more motifs can be specified, and each motif will consist of a unique combination of interactions formed between the protein and the ligand.

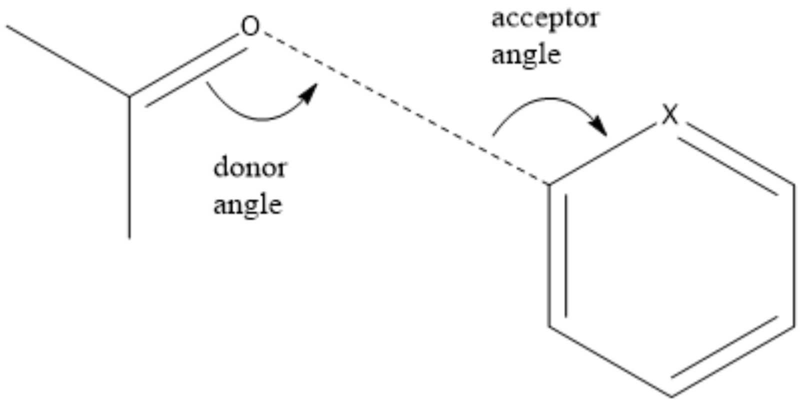
Individual interactions are described according to their protein atom interaction type (hydrogen bond acceptor, hydrogen bond donor, lipophilic interaction, or weak CH...O acceptor).

GOLD assesses whether or not specified interactions are satisfied as follows:

For H-bond acceptor and H-bond donor interaction types - A hydrogen bond is deemed to be present if the distance between the acceptor and the donor heavy atoms are within the range 2.85Å ± 0.45, the acceptor angle is within 145° ± 65 and the donor angle is within 115° ± 40. Furthermore, a planarity check is used to ensure that the hydrogen bond is not more than 30° out of the plane when the ligand donor is of atom type N.2 or N.pl3 or the protein donor is of atom type N.pl3 or N.am.



For a weak CH...O acceptor interaction type - An interaction is deemed to be present if the distance between the acceptor and the aromatic carbon is within the range 3.35Å ± 0.65, the acceptor angle is within 145° ± 65, the donor angle is within 115° ± 40 and the CH...O bond is not more than 30° out of the aromatic plane. Note the presence of a heteroatom (X in the figure below, where X is O (O.3), N (N.ar, N.2, N.am, N.pl3), S (S.3)) is required in the aromatic ring.



For a lipophilic interaction type - An interaction is deemed to be present if the sum of the protein and the ligand atom's van der Waals radii plus 0.4 is less than the distance between the protein and the ligand atoms.

During docking a contribution will be added to the fitness score of ligand poses in which a motif is matched (i.e. poses in which all the interactions defined as part of a motif are satisfied). This contribution is based upon the accumulated hydrogen-bonding and lipophilc interactions defined as part of that motif. Therefore, docking will be biased towards ligand poses which form interactions to the protein atoms of interest matching one of the uniquely defined motifs.

Please note that it is not possible to manipulate interaction motif constraints within the constraint editor (see Using the Constraint Editor).

11.8.2 Setting up an Interaction Motif Constraint

To set up an interaction motif constraint you must first select the appropriate protein tab, adjacent to the **Global Options** tab. Then, click on **Interaction Motif** from the list of options given on the left of the **GOLD Setup** window.

One or more motifs can be specified, and each motif will consist of a unique combination of interactions formed between the protein and the ligand.

To define an interaction:

- Click on the **Add Interaction** button.

Interactions are described according to their protein atom interaction type. Select the type of interaction (hydrogen bond acceptor, hydrogen bond donor, lipophilic interaction, or weak CH... O acceptor) using the drop-down menu under the column labelled **Type**.

Next, specify the protein atom that forms the interaction. This can be done by clicking on an atom in the visualiser. Alternatively, you can enter the residue, chain, name and atom identifiers directly into the appropriate entry boxes. Note that your protein input file must include chain identifiers in order to avoid problems when specifying the atom for use in this constraint.

A maximum of 10 interactions can be defined. Note that hydrogen bond acceptors that can form either a hydrogen bond or a weak CH...O interaction need to be added twice, one for each type of interaction.

To delete an interaction, click on the corresponding row number to highlight the interaction definition then hit the **Delete Interaction** button.

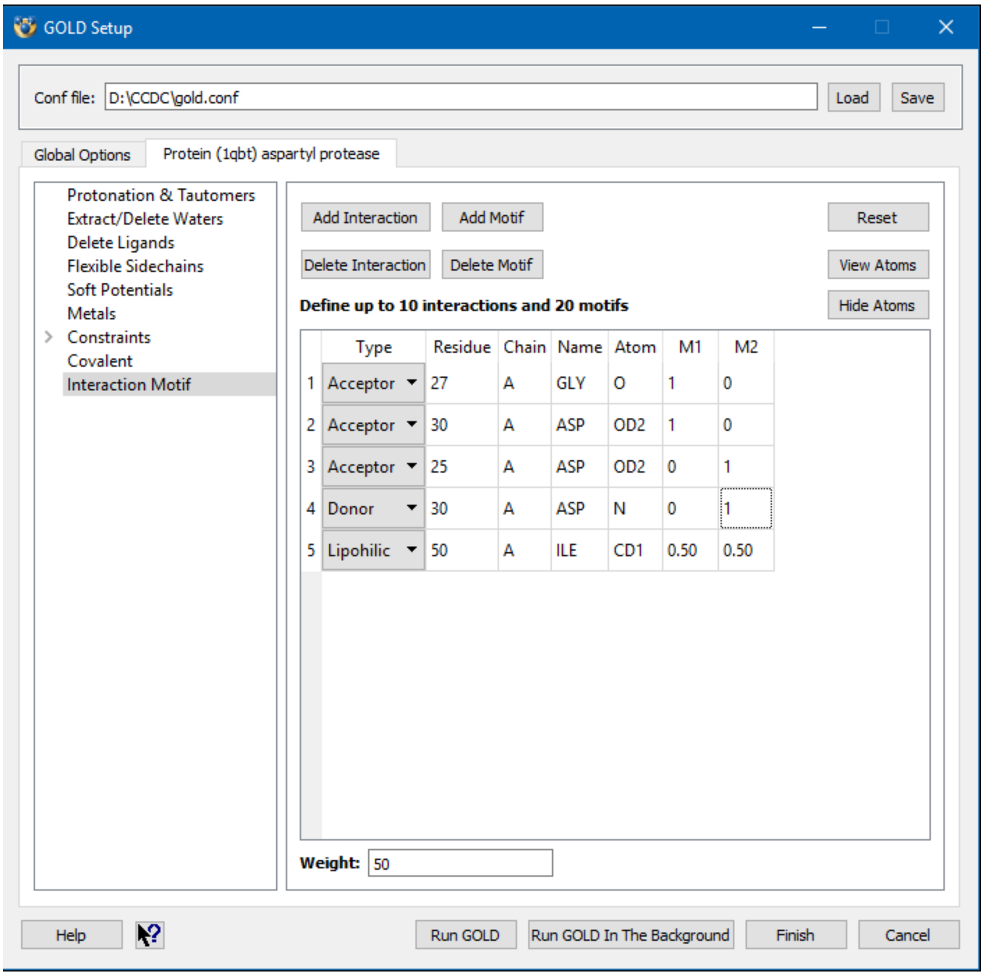
One or more binding motifs can now be defined. Each motif should consist of a unique combination of the interactions that you previously specified. To define a motif:

- Click the **Add Motif** button.

Specify each of the interactions that need to be included in the motif. The hydrogen bond interaction types (Acceptor, Donor, CHO) are set as either 1 or 0 depending on whether or not they are observed in a particular motif. The lipophilic interaction type is set as the frequency of that interaction as observed in the set of complexes used to originally identify the motifs. The frequency of a lipophilic interaction is added to all the interaction motifs.

To delete a motif, click on the corresponding motif number column then hit the **Delete Motif** button. The maximum number of motifs that can be defined is 20.

In the example below, two motifs have been defined. Motif **M1** features Gly27 O as an acceptor and Asp30 OD2 as an acceptor, but not Asp25 OD2 as an acceptor nor Asp30 N as a donor. Motif **M2** features Asp25 OD2 as an acceptor and Asp30 N as a donor, but not Gly27 O as an acceptor nor Asp30 OD2 as an acceptor. The lipophilic interaction is included in both motifs and is expressed as the frequency of that interaction as observed in the set of complexes originally used to identify the motifs.



Conf file: D:\CCDC\gold.conf

Define up to 10 interactions and 20 motifs

	Type	Residue	Chain	Name	Atom	M1	M2
1	Acceptor	27	A	GLY	O	1	0
2	Acceptor	30	A	ASP	OD2	1	0
3	Acceptor	25	A	ASP	OD2	0	1
4	Donor	30	A	ASP	N	0	1
5	Lipophilic	50	A	ILE	CD1	0.50	0.50

Weight: 50

Only default residues' labels are displayed here.

In the Hermes visualizer, to highlight the protein atoms involved in a motif:

- Highlight the appropriate column and click on the **View Atoms** button.

To switch off highlighting:

- Click the **Hide Atoms** button.

In order to remove all interaction and motif definitions:

- Click on **Reset**.

12 Balancing Docking Accuracy and Speed

12.1 Number of Dockings

GOLD will dock each ligand several times, starting each time from a different random population of ligand orientations. The results of the different docking runs are ranked by fitness score.

The number of dockings to be performed on each ligand is set when the ligand file is defined (see Specifying the Ligand File(s)).

By default the number of dockings to be performed on each ligand is 10.

The total time spent docking a ligand obviously depends on the number of docking runs, so you can make GOLD go faster by reducing this number. However, it is useful to perform at least a few docking runs on each ligand. This increases the chances of getting the right answer. Also, if the same answer is found in several different docking runs, it is usually a strong indicator that the answer is correct.

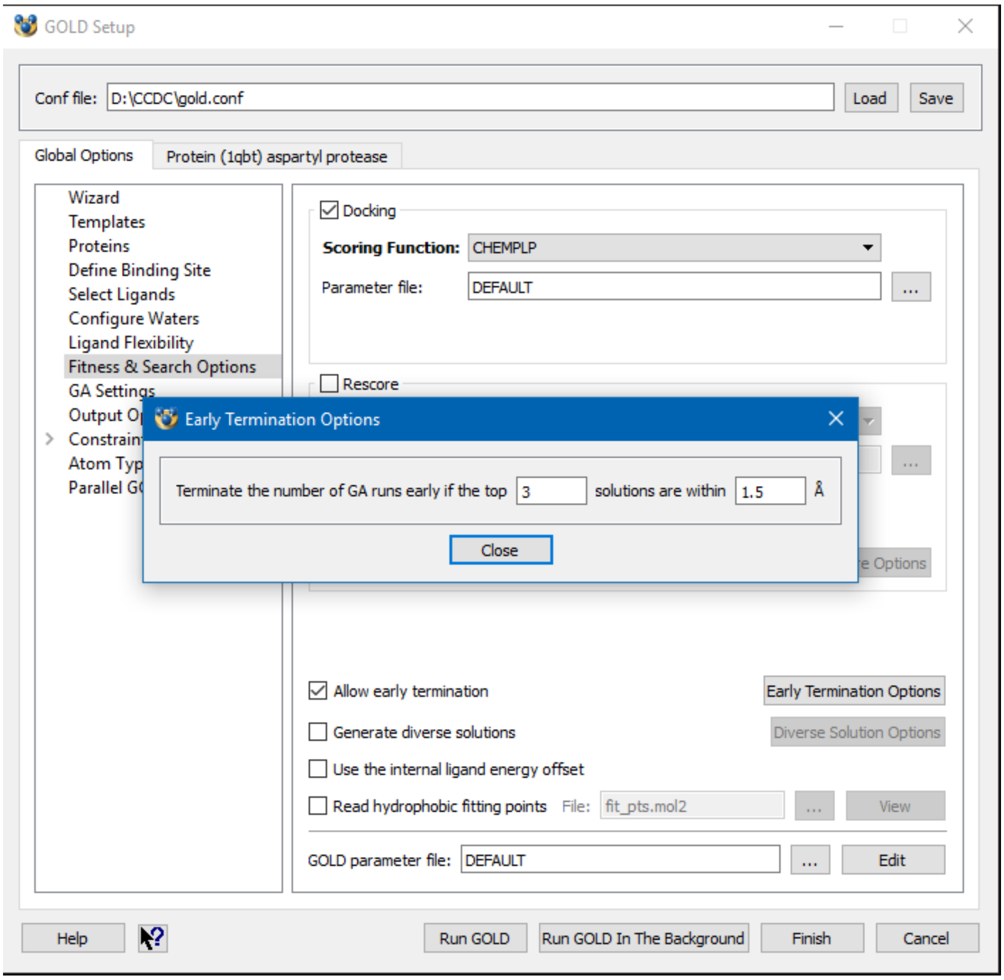
The **early termination** option (see Early Termination) can be used to prevent GOLD wasting time performing multiple docking runs on easy ligands.

12.2 Early Termination

The early termination option instructs GOLD to terminate docking runs on a given ligand as soon as a specified number of runs have given essentially the same answer. In this situation, it is probable that the answer is correct, and GOLD will just be wasting time if it performs more docking runs on that ligand.

To switch early termination on, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, then enable the **Allow early termination** check box.

To specify the early termination criterion, click on the **Early Termination Options** button. In the example below, GOLD has been instructed to stop docking a ligand if it reaches a state in which the best three solutions found so far are all within 1.5 Å heavy atom RMSD of each other:



Conf file: D:\CCDC\gold.conf

☑ Docking

Scoring Function: CHEMPLP

Parameter file: DEFAULT

☐ Rescore

Terminate the number of GA runs early if the top 3 solutions are within 1.5 Å

☑ Allow early termination

☐ Generate diverse solutions

☐ Use the internal ligand energy offset

☐ Read hydrophobic fitting points File: fit_pts.mol2

GOLD parameter file: DEFAULT

The RMS deviation takes account of any ligand symmetry.

Early termination does not always save as much time as you might think, because it tends to be invoked for easy (i.e. relatively rigid) ligands, which are quick to dock anyway.

12.3 Controlling Accuracy and Speed with Genetic Algorithm Parameter Settings

12.3.1 Genetic Algorithm Overview

GOLD optimises the fitness score by using a genetic algorithm.

A population of potential solutions (i.e. possible docked orientations of the ligand) is set up at random. Each member of the population is encoded as a chromosome, which contains information about the mapping of ligand H-bond atoms onto (complementary) protein H-bond atoms, mapping of hydrophobic points on the ligand onto protein hydrophobic points, and the conformation around flexible ligand bonds and protein OH groups.

Each chromosome is assigned a fitness score based on its predicted binding affinity and the chromosomes within the population are ranked according to fitness.

The population of chromosomes is iteratively optimised. At each step, a point mutation may occur in a chromosome, or two chromosomes may mate to give a child. The selection of parent chromosomes is biased towards fitter members of the population, i.e. chromosomes corresponding to ligand dockings with good fitness scores.

A number of parameters control the precise operation of the genetic algorithm:

- `Population size` (see Population Size)
- `Selection pressure` (see Selection Pressure)
- `Number of operations` (see Number of Operations)
- `Number of islands` (see Number of Islands)
- `Niche size` (see Niche Size)
- `Operator weights`: migrate, mutate, crossover (see Operator Weights: Migrate, Mutate, Crossover)
- Van der Walls and hydrogen bonding annealing parameters (see Van der Waals and Hydrogen Bonding Annealing Parameters)
- Changes to individual genetic algorithm parameters should be made with care (see Using User-Defined Genetic Algorithm Parameter Settings).

12.3.2 Relationship between Genetic Algorithm Parameters and Speed

The time taken by GOLD to dock ligands can be controlled by altering the values of the genetic algorithm (GA) parameters.

GOLD runs for a fixed number of genetic operations (crossover, migration, mutation). The easiest way to make GOLD go faster is to reduce the number of GA operations performed in the course of a run. This is done through the `Number of Operations` variable (this parameter is called `maxops` in the configuration file).

A reduction in `Number of Operations` is likely to change the optimum values of several other GA parameters, particularly `popsize`, `van der Waals` and `Hydrogen Bonding`.

GOLD manipulates a pool of chromosomes of size `popsize * Number of Islands`. The size of this pool should be such that the optimisation converges within the specified maximum number of operations, `Number of Operations`. If the pool size is too small for a given value of `Number of Operations`, the algorithm will converge prematurely. Conversely, if the pool size is too large the algorithm will terminate before it has converged.

The annealing parameters `van der Waals` and `Hydrogen Bonding` allow poor hydrogen bonds to occur at the beginning of a genetic algorithm run, in the expectation that they will evolve to better solutions. Both the vdw and H-bond annealing must be gradual, and the population allowed plenty of time to adapt to changes in the fitness function.

Because of these factors, it is difficult to set GA parameters by hand and you are recommended to use automatic (ligand-dependent) GA parameter settings (see Using Automatic (Ligand-Dependent) Genetic Algorithm Parameter Settings), or one of the default parameter sets offered (see Using Preset Genetic Algorithm Parameter Settings).

12.3.3 Using Automatic (Ligand-Dependent) Genetic Algorithm Parameter Settings

The number of genetic operations performed (crossover, migration, mutation) is the key parameter in determining how long a GOLD run will take (i.e. this parameter controls the coverage of the search space).

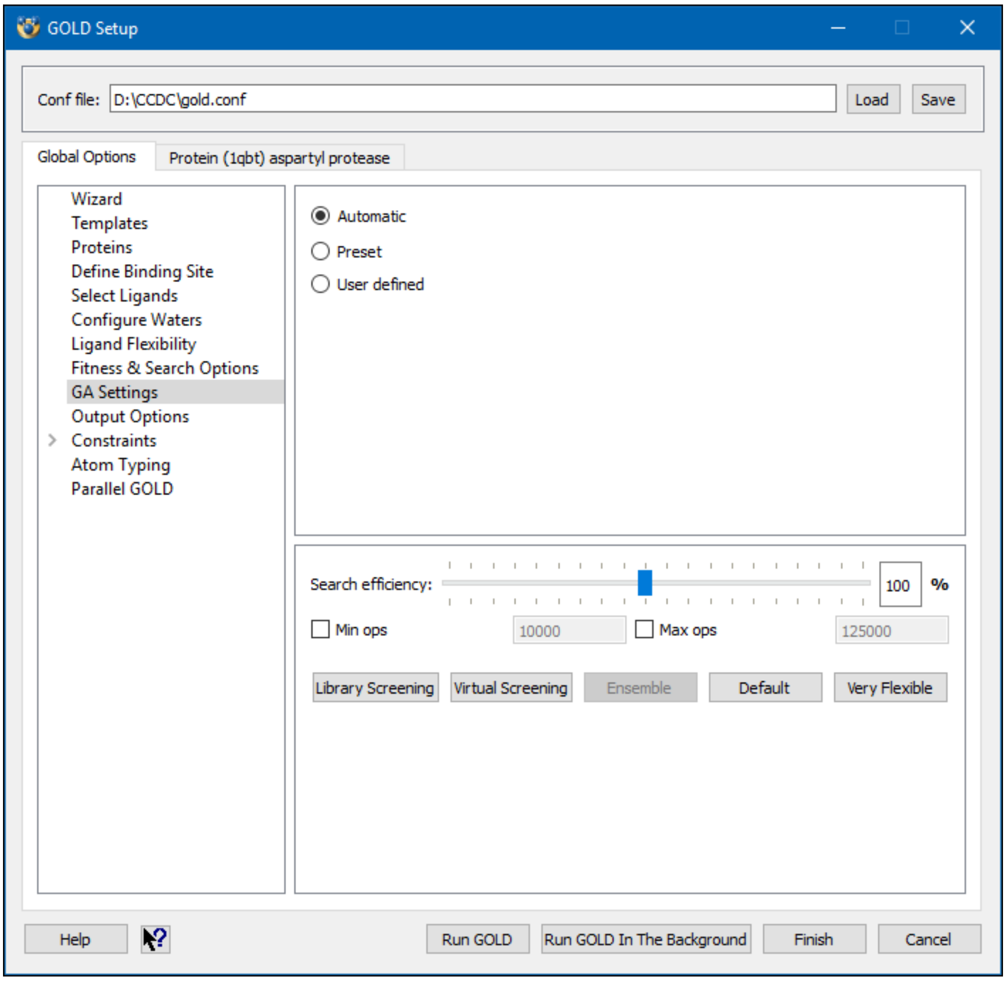
GOLD can automatically calculate an optimal number of operations for a given ligand, thereby making the most efficient use of search time, e.g. small ligands containing only one or two rotatable bonds will generally require fewer genetic operations than larger, highly flexible ligands.

The criteria used by GOLD to determine the optimal GA parameter settings for a given ligand include:

- The number of rotatable bonds in the ligand.
- Ligand flexibility, i.e. number of flexible ring corners, flippable nitrogens, etc. (see Ligand Flexibility).
- The volume of the protein binding site.
- The number of water molecules considered during docking (see Water Molecules).

The exact number of GA operations contributed, e.g. for each rotatable bond in the ligand, are defined in the `gold.params` file (see Altering GOLD Parameters: the gold.params File).

To enable automatic (i.e. ligand-dependent) GA settings, click on **GA Settings** from the list of **Global Options** given on the left of the **GOLD Setup** window, then switch on the button labelled **Automatic.**


Conf file: D:\CCDC\gold.conf

● Automatic
○ Preset
○ User defined

Search efficiency: 100 %

□ Min ops 10000
□ Max ops 125000

When using these ligand-dependent GA settings, the **Search efficiency** can be used to further control the speed of docking and the predictive accuracy (i.e. the reliability) of the results.

With the **Search efficiency** set at 100% GOLD will attempt to apply optimal settings for each ligand. For a ligand with five rotatable bonds this will be around 30,000 GA operations. If the **Search efficiency** were set to 50%, then GOLD would perform around 15,000 operations, thereby speeding up the docking by a factor of two, but the search space would be less well explored. Similarly, by setting a **Search efficiency** greater than 100%, it is possible to make the search more exhaustive (but slower).

The following search efficiency settings are available by clicking the corresponding button:

- **Very Flexible** - this sets the search efficiency at 200% and is recommended for large, highly flexible ligands. This setting delivers high predictive accuracy but is relatively slow.
- **Default** - this sets the search efficiency at 100%. GOLD will attempt to apply optimal settings for each ligand, see above.
- **Virtual Screening** - this sets the search efficiency at 30%. This setting is suitable for routine work and usually gives comparable predictive accuracy to the slower settings, unless the ligand has a large number of rotatable torsions.
- **Ensemble** – this sets the search efficiency at 75% and it is recommended for ensemble docking.
- **Library Screening** - this sets the search efficiency at 10%. This is the fastest setting and as a consequence is the least reliable.

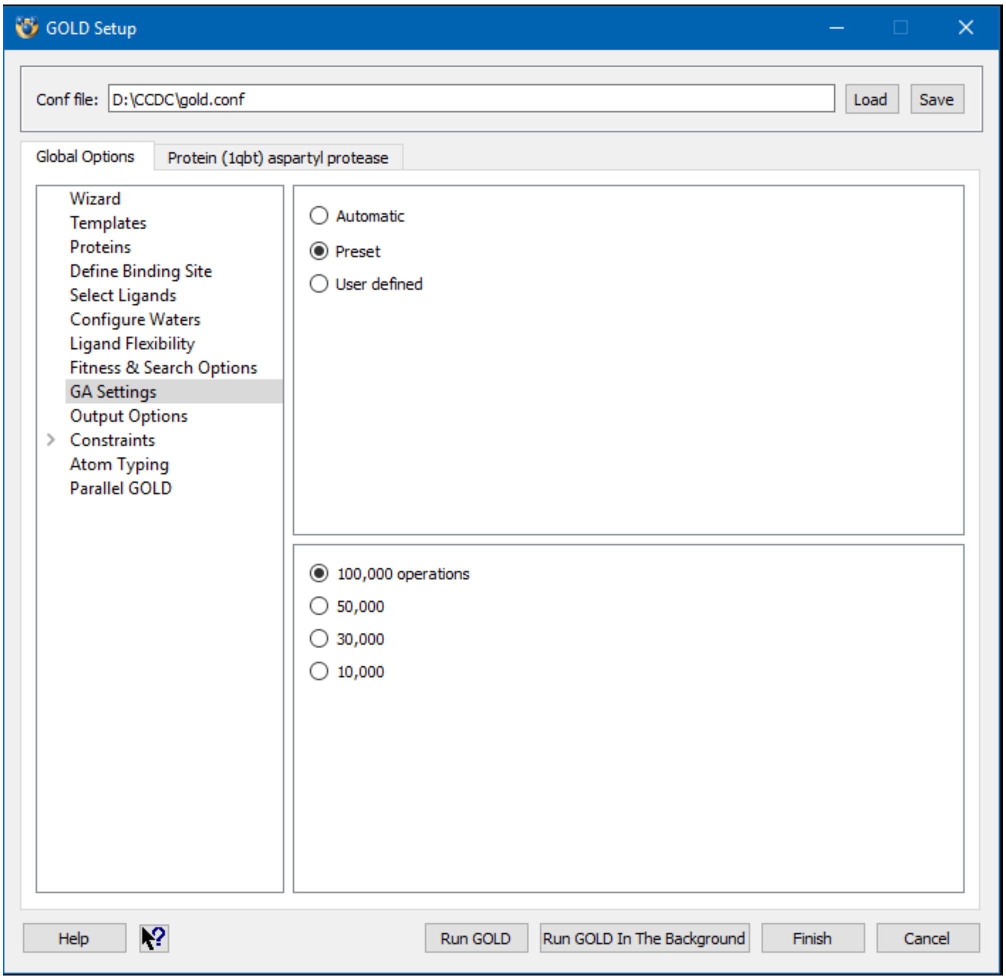
The Minimum number of operations performed during the run will therefore depend on the **Search efficiency** that is set. To ensure that every ligand is subjected to at least a user-specified number of operations enable the **Min ops** check-box and specify the minimum number of operations required for every ligand. Similarly, the maximum number of operations to be carried out can be set manually with the **Max ops** check-box.

12.3.4 Using Preset Genetic Algorithm Parameter Settings

The number of genetic operations performed (crossover, migration, mutation) is the key parameter in determining how long a GOLD run will take (i.e. this parameter controls the coverage of the search space).

When using pre-defined GA parameter settings every ligand, regardless of its size and flexibility, will be subjected to a specified number of genetic operations.

To use a pre-defined GA parameter set, click on **GA Settings** from the list of **Global Options** given on the left of the **GOLD Setup** window, then switch on the button labelled **Preset**.



Conf file: D:\CCDC\gold.conf

○ Automatic
● Preset
○ User defined

● 100,000 operations
○ 50,000
○ 30,000
○ 10,000

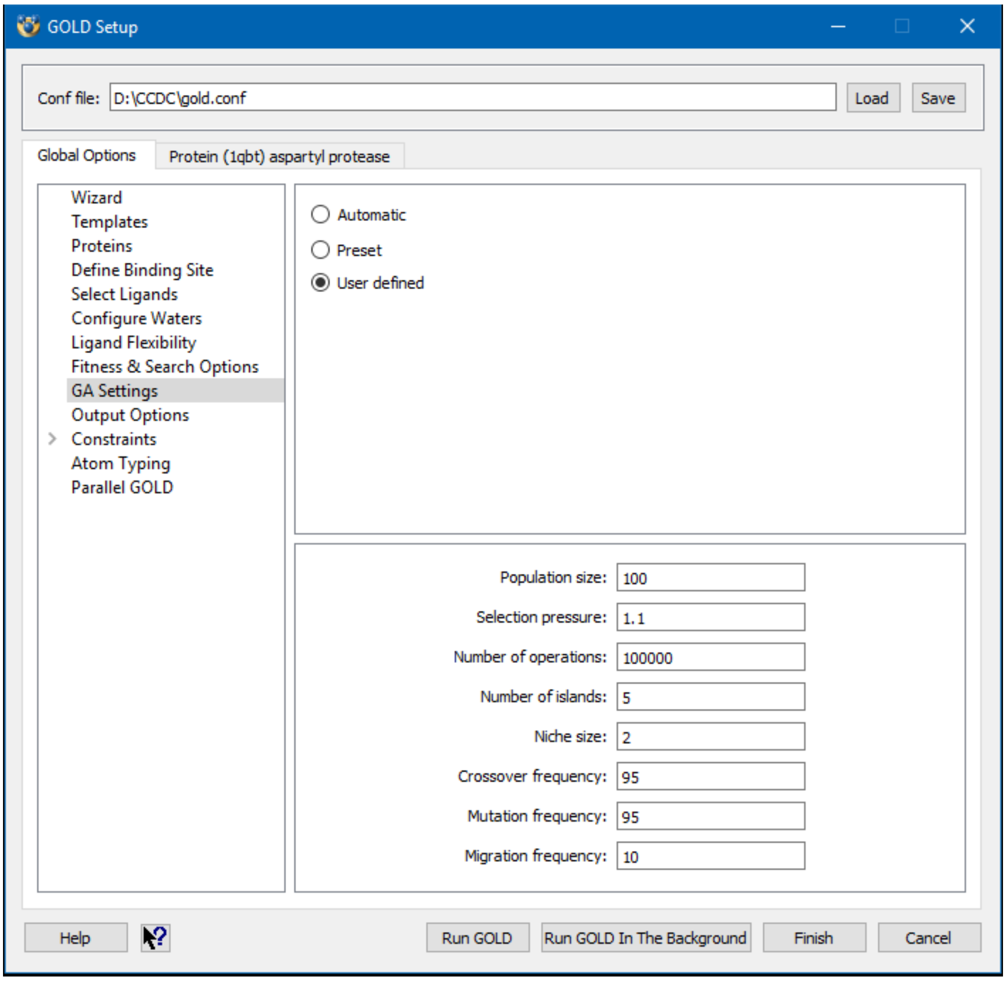
Select the required number of genetic operations from those listed:

- **100,000 operations** deliver high predictive accuracy but are relatively slow. These settings are recommended for use with large, highly flexible ligands, or for research applications where speed of docking is not an issue and optimal accuracy is required.
- **50,000 operations and 30,000 operations** are progressively quicker (predictive reliability will fall off, but quite slowly). These setting are recommended for use with compounds containing up to six flexible bonds and/or ring corners (see Ring Conformations).
- **10,000 operations** will give comparable predictive accuracy to the slow (100,000 operations) setting when docking small rigid ligands. These settings are recommended for use with ligands containing one or two rotatable torsions and for virtual screening work.

12.3.5 Using User-Defined Genetic Algorithm Parameter Settings

Individual GA parameters can be specified manually. However, it is recommended that you use the automatic (ligand-dependent) settings (see Using Automatic (Ligand-Dependent) Genetic Algorithm Parameter Settings), or one of the pre-defined GA parameter sets (see Using Preset Genetic Algorithm Parameter Settings) as opposed to altering individual parameters, because the optimum values of the parameters are highly correlated.

To manually specify individual GA parameter values, click on **GA Settings** from the list of **Global Options** given on the left of the **GOLD Setup** window, then switch on the button labelled **User defined**.



Conf file: D:\CCDC\gold.conf

○ Automatic
○ Preset
● User defined

Population size: 100
Selection pressure: 1.1
Number of operations: 100000
Number of islands: 5
Niche size: 2
Crossover frequency: 95
Mutation frequency: 95
Migration frequency: 10

The values for individual GA parameters can be specified using the appropriate entry box.

A definition of the individual genetic algorithm parameters is provided in (see Appendix D: Genetic Algorithm Parameter Definitions).

13 Rescoring

13.1 Overview

Different scoring functions may perform better for selected cases. You may find, for example, that ChemScore outperforms GoldScore in ranking actives or one protein class, whereas the reverse will apply for other classes.

Therefore, when screening large numbers of compounds, rescoring docking poses with alternative scoring functions and considering the best results from each (consensus scoring) can have a favourable impact on the overall rank ordering of ligands.

In GOLD it is possible to rescore a single ligand or a set of ligands in one or more files.

Typically, a user will rescore GOLD solution files with an alternative scoring function. However, it is also possible to score a ligand pose from an alternative source (for example, from a known crystal structure or a solution from another docking program). When rescoring from a source other than a GOLD solution file it will not be possible to use the optimised positions of polar protein hydrogen atoms (see Rescore settings).

Rescoring can be performed automatically after a docking run. This will result in the solutions from the docking being automatically scored with another scoring function. Alternatively, rescoring can be performed independently of the docking, e.g. against an existing set of GOLD solution files (or ligand poses from an alternative source) (see [Setting Up a Rescoring Run).

It is not possible to use the rescore feature if GOLD is being run in parallel (see Running GOLD in Parallel).

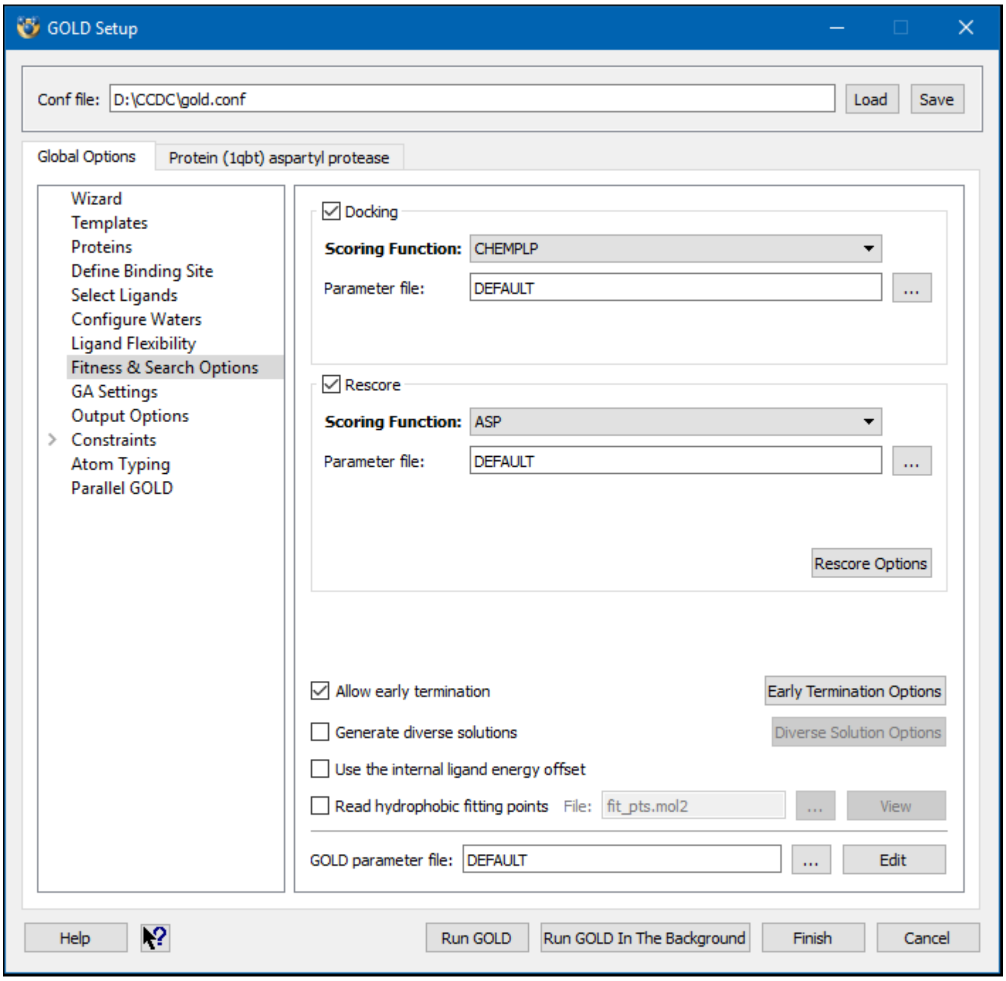
13.2 Setting Up a Rescoring Run

To automatically rescore the results of a docking run with another scoring function you will need to first set up the docking in the normal way. Then, click on **Fitness & Search Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and enable the **Rescore** check-box.

Select the required scoring function to be used for the rescore from the drop-down menu. To use a modified scoring function parameters file either enter the path and filename of the **Parameter file** or click on the **...** button and use the file selection window to choose the file.

Finally, specify the settings to be used for the rescoring run (see Rescore settings).

In the following example ChemPLP will be used for the docking and the resulting solutions will be rescored automatically using the Astex Statistical Potential (ASP) scoring function:



Conf file: D:\CCDC\gold.conf

☑ Docking

Scoring Function: CHEMPLP

Parameter file: DEFAULT

☑ Rescore

Scoring Function: ASP

Parameter file: DEFAULT

☑ Allow early termination

□ Generate diverse solutions

□ Use the internal ligand energy offset

□ Read hydrophobic fitting points File: fit_pts.mol2

GOLD parameter file: DEFAULT

To rescore an existing set of GOLD solution files or ligand poses from an alternative source (i.e. without first running a docking) enable the **Rescore** check-box only and select the required scoring function to be used for the rescore from the drop-down menu. To use a modified scoring function parameters file, either enter the path and filename of the **Parameter file** or click on the **...** button and use the file selection window to choose the file.

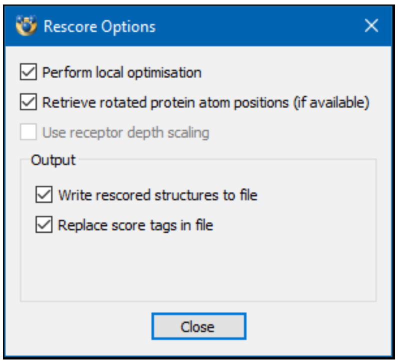
Rescoring in this way requires essentially the same information as a normal docking run. You will therefore need to:

- Provide a prepared protein input file (see Specifying the Protein File or Files).
- Define the binding site (preferably the same definition that was used for the original docking) (see Defining the Binding Site).
- Specify the ligand(s) you wish to rescore (see Specifying the Ligand File(s)).
- Specify the fitness function to be used for the rescoring (see Selecting a Fitness Function).

Finally, specify the settings to be used for the rescoring run (see Rescore settings).

13.3 Rescore settings

To specify the settings to be used for the rescoring run hit the **Rescore Options** button. This will open the **Rescore Options** dialogue:



Rescore Options

☑ Perform local optimisation

☑ Retrieve rotated protein atom positions (if available)

□ Use receptor depth scaling

Output

☑ Write rescored structures to file

☑ Replace score tags in file

The following rescore options are available:

- **Perform local optimisation** - Enable this check-box to minimise the docked ligand pose before rescoring. Simplexing is important if you are to obtain meaningful scores. Due to the nature of scoring functions, one finds that small changes in location or conformation of the pose can have large effects on the calculated score. Simplexing can also affect rotatable protein hydrogen atoms (see File Containing the Protein Binding-Site Geometry).
- **Retrieve rotatable protein atom positions (if available) -** When rescoring a GOLD solution file it is possible to use the optimised positions of the polar protein hydrogen atoms that were generated during the original docking (see File Containing the

Protein Binding-Site Geometry). If this option is not switched on (or no rotatable H positions are available) then the default hydrogen atoms positions specified in the protein input file will be used.

- **Use receptor depth scaling** - This option is only available when rescoring with ChemScore. When using receptor depth scaling the score attributed to hydrogen bonds is scaled depending on the depth in the pocket. Hydrogen bonds deep in the pocket are rewarded with an increased score, while the scores of those closer to the solvent-exposed surface are decreased (see Receptor Depth Scaling).

The following output options are available:

- **Write rescored structures to file -** Enable this check-box to write out docked ligand solutions after rescoring. Solutions will be written to the file `rescore.mol2` (to specify an alternative filename (see Rescore Solution File), mol2 or sd output can be specified (see Files Containing the Docked Ligand(s)). Solution files will contain the new scoring function terms. If writing of this file is switched off, only the `rescore.log` file will be written (see Rescore Log File).
- **Replace score tags in file** - When rescoring a GOLD solution file, enable this check-box to overwrite the list of active residues and the rotated protein hydrogen atom positions generated during the original docking with those resulting from the rescoring run. If you select not to replace relevant tags then `rescore.mol2` will contain both the binding site definition of the original docking and that of the subsequent rescoring run.

13.4 Receptor Depth Scaling

In many proteins, the cognate ligand forms hydrogen bonds deep in the active site. This feature of known binders can be exploited using receptor depth scaling, where the score attributed to hydrogen bonds is scaled depending on the depth in the pocket.

Hydrogen bonds deep in the pocket are rewarded with an increased score, while the scores of those closer to the solvent-exposed surface are decreased. Simultaneously, the scores attributed to lipophilic interactions are reduced.

This procedure has been shown to increase the relative scores of active molecules compared to inactive molecules across a diverse range of 85 proteins (Using Buriedness to Improve Discrimination Between Actives and Inactives in Docking, see References). While the default values for the parameters are suitable for the general case, for a particular protein it may be possible to gain better results by adjusting the scaling parameters (see the GOLD Configuration File documentation for further information).

Simplexing is turned on when rescoring with receptor depth scaling.

Receptor depth scaling is only available when rescoring with ChemScore (see Setting Up a Rescoring Run).

13.5 Rescore Output Files

Each rescored solution is written to the `rescore.log` file. This file contains the ligand identifiers, the final rescore fitness value and its component terms. To specify an alternative rescore log filename, see Rescore Log File.

Rescored structure solution files can be written out that will contain the new scoring function terms. Solutions will be written to the file `rescore.mol2` (see Rescore settings).

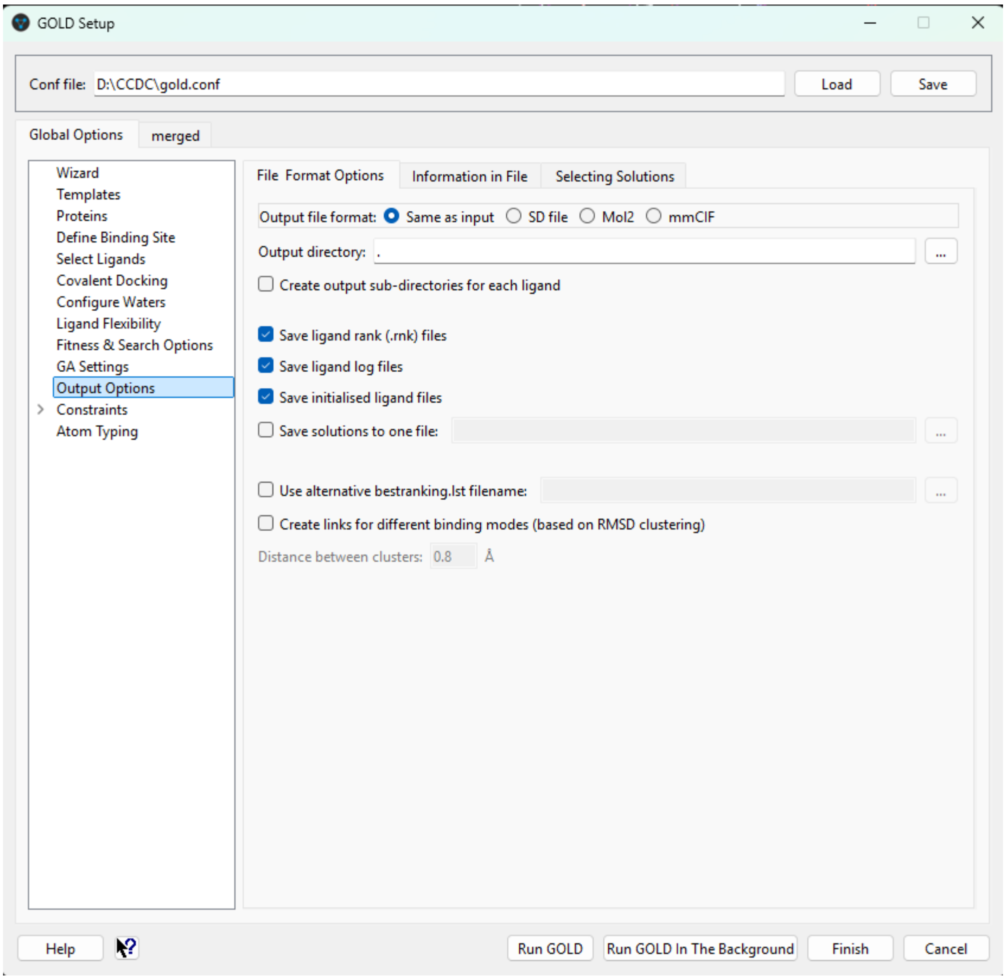
14 Docked Ligand Output Options

14.1 Specifying Ligand Solution File Formats and Directories

Click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, then select the **File Format Options** tab.

By default, docked ligands will be written out in the same format as was used for input. To change this, specify the required file format by selecting either **Same as input**, **SD file**, or **Mol2.**

Use the **Output directory** entry box to specify the directory to which output files will be written or click on the **...** button and use the directory selection window to choose the location. When more than one ligand is being docked, switch on the **Create output sub-directories** check box if you want results for each ligand to be written to a separate sub-directory.



Conf file: D:\CCDC\gold.conf

Output file format: ● Same as input ○ SD file ○ Mol2 ○ mmCIF

Output directory: .

□ Create output sub-directories for each ligand

☑ Save ligand rank (.rnk) files

☑ Save ligand log files

☑ Save initialised ligand files

□ Save solutions to one file:

□ Use alternative bestranking.lst filename:

□ Create links for different binding modes (based on RMSD clustering)

Distance between clusters: 0.8 Å

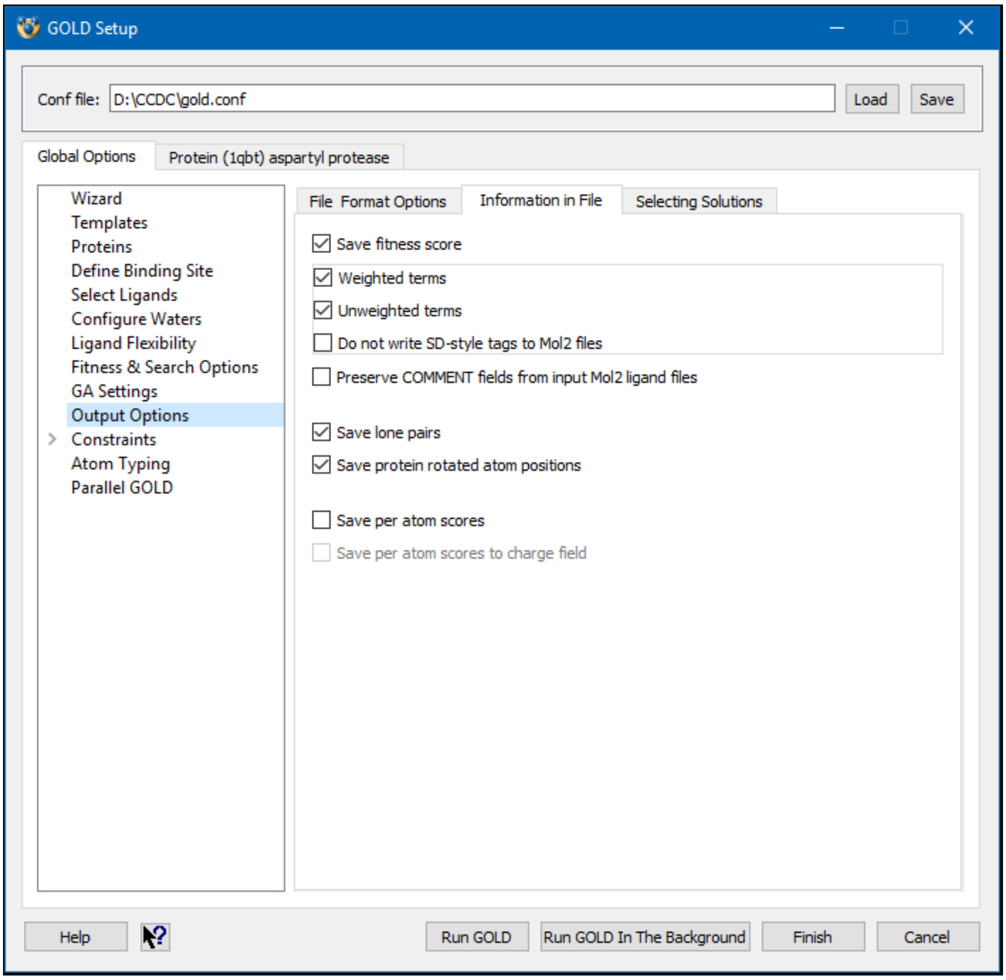
Each ligand will normally be docked several times, so a given input ligand will produce a set of files, each containing the results of a separate docking attempt. Alternatively, you can specify that all saved docking solutions for all ligands are to be concatenated and written to a single file. To do this, enable to **Save solutions to one file** check-box and either enter the path and filename of the file, or click on the **...** button and use the file selection window to choose the file.

14.2 Controlling the Information Written to Ligand Solution Files

It is possible to write additional information to docked solution files. This information is written to sd file tags; for mol2 files, these tags are written to comment blocks.

For post-processing docking results, it is particularly important that the scoring function terms and the rotated protein positions are saved.

Click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, then select the **Information in File** tab.



☑ Save fitness score

☑ Weighted terms

☑ Unweighted terms

□ Do not write SD-style tags to Mol2 files

□ Preserve COMMENT fields from input Mol2 ligand files

☑ Save lone pairs

☑ Save protein rotated atom positions

□ Save per atom scores

□ Save per atom scores to charge field

The following options are available:

- **Save fitness score -** Enable this check-box if you want the docked solution files to include the docking-score terms, e.g. the total GoldScore fitness value for each docking and its components such as protein-ligand H-bond energy, internal ligand strain energy, etc.

- **Weighted terms -** Certain docking scoring function terms are the product of a term dependent on the magnitude of a particular physical contribution (e.g. hydrogen bonding) and a scale factor determined e.g. by a regression coefficient. The docking scoring function terms included in the output file can therefore consist of weighted terms, non-weighted terms or both. To include weighted terms, enable this check-box.

- **Unweighted terms -** Enable this check-box to include non-weighted scoring function terms in the output file.

- **Do not write SD-style tags to Mol2 files** - Enable this check-box to prevent SD-style tags being written to comment blocks in mol2 solution files.

- **Preserve COMMENT fields from input Mol2 ligand files** - Enable this check-box to retain the COMMENT fields from input mol2 ligand files in the docked solution files.

- **Save lone pairs -** Some third-party programs have difficulty reading files which contain lone pairs. You can stop GOLD including lone pairs when it writes docked solution files by switching off this check-box.

- **Save protein rotated atom positions -** Enable this check-box to save the optimised positions of rotated protein atoms. These include the optimised positions of polar protein hydrogen atoms and also final positions of any protein side chains that have been defined as being flexible. Protein atom positions that are generated during docking will usually be different for each docked ligand pose and are therefore written to the individual ligand solution files. Rotated atom positions are utilised by **Hermes**.

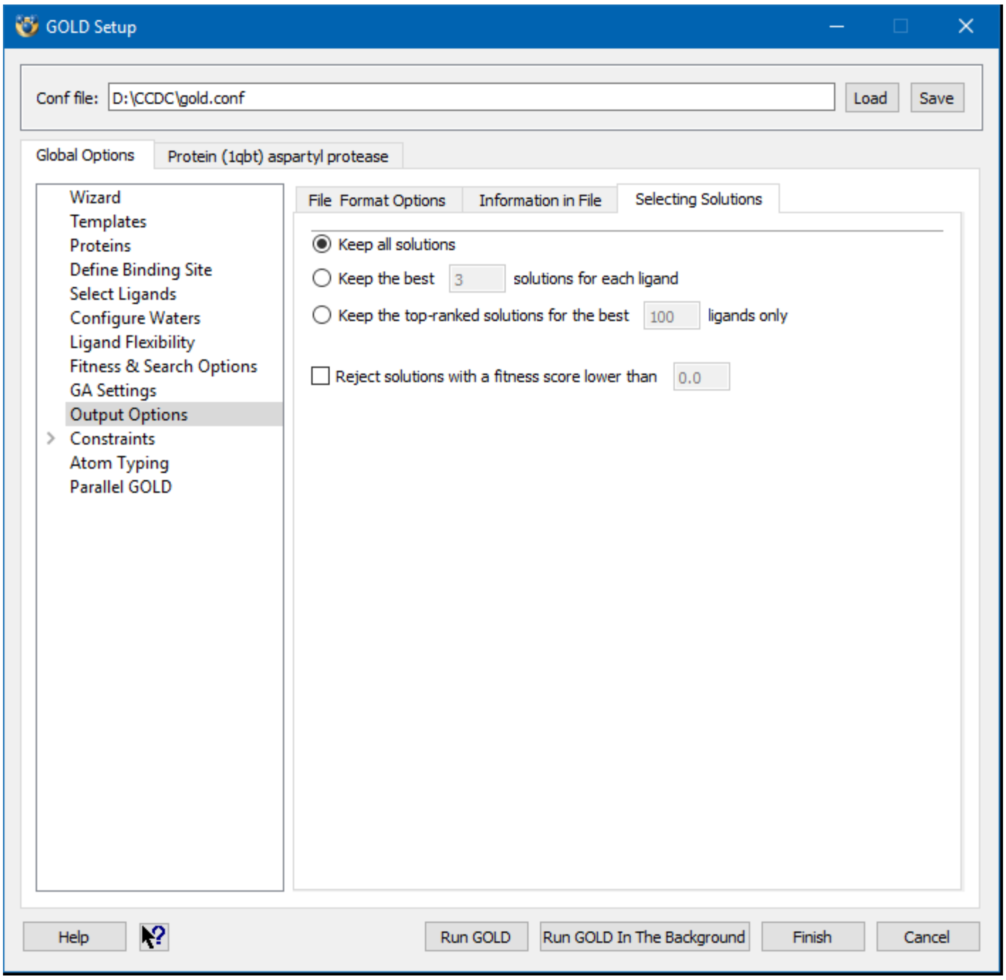
- **Save per atom scores -** Enable this check-box to include the scoring contributions of individual ligand and protein atoms to be written to docked solution output files. For each atom its contribution to the total fitness score and also the constituent scoring terms will be written.

- **Save per atom scores to charge field -** Enable this check-box to write to the mol2 file charge field of docked solution files the scoring contribution of individual ligand atoms.

14.3 Selecting Which Ligand Solutions to Keep

By default, GOLD will dock each ligand 10 times starting each time from a different random population of ligand orientations (see Number of Dockings). This can produce a lot of output and you may therefore wish to reduce the number of docking solutions that are retained.

Click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, then select the **Selecting Solutions** tab.



Conf file: D:\CCDC\gold.conf

● Keep all solutions

○ Keep the best 3 solutions for each ligand

○ Keep the top-ranked solutions for the best 100 ligands only

□ Reject solutions with a fitness score lower than 0.0

By selecting the appropriate option, it is possible to either:

- **Keep all solutions**.
- **Keep the best n solutions for each ligand**, where n is a user-specified number (e.g. n = 3).
- **Keep the top-ranked solutions for the best m ligands only**, i.e. retain just the best solution for only those m ligands with the best fitness scores, where m is user-specified (e.g. m = 100).
- In addition, you can filter out all solutions with fitness scores lower than a specified value by switching on the button labelled **Reject solutions with a fitness score lower than x**. This filter can be used in combination with the options listed above. For example, you could save 3 solutions for each ligand and not keep any solution with a fitness lower than 50.

15 Running GOLD

15.1 Required Input Files

The following files must be available before a GOLD job can be run:

- One or more files containing the ligand(s) to be docked, in mol2, mol, sd, mmCIF or pdb format (pdb format is not recommended for ligand files) (see Setting Up Ligands).
- A file (or files) containing the protein(s) (or the part of a protein) into which the ligand is to be docked. This may be in mmCIF, pdb, or mol2 format (see Setting Up the Protein(s)).
- GOLD also needs a configuration file, which contains the names of the protein and ligand files, and all the user-defined parameters such as genetic algorithm parameter settings, fitness flags, etc. The configuration file can be created manually, but it is usually easier and preferable to create it with the GOLD graphical front end (the file can be written out when the **Run GOLD**, **Run GOLD in The Background** and **Finish** buttons are hit). A number of configuration file templates are also available (see Using Configuration File Templates).
- In addition, GOLD uses a parameter file (see Altering GOLD Parameters: the `gold.params` File), a scoring function specific parameter file (see Fitness Functions), and (optionally) a torsion distribution file (see Using Torsion Angle Distributions). All these files are supplied in the GOLD distribution and, by default, will be found automatically by the program. If required, any of the files can be copied to a user's directory and edited, and GOLD can then be directed to use the edited file(s).

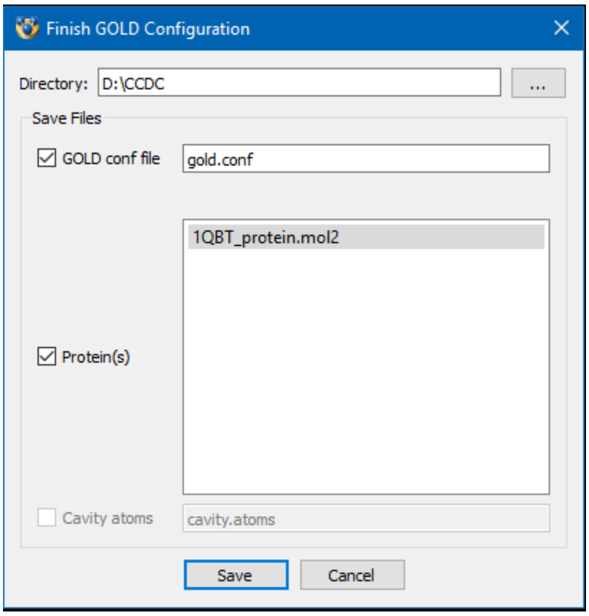
15.2 Running GOLD Interactively

GOLD can be run interactively by hitting the **Run GOLD** button in the **GOLD Setup** window.

Before the job is started you will be prompted to save a configuration file. The configuration file is a text file which specifies the GOLD calculation that is to be run, including details of the ligand, the protein binding site, the fitness-function parameter file to be used, the torsion distribution file to be used, and the genetic algorithm parameters (see Saving and Re-using Program Settings in Configuration Files).

To save a configuration file, specify the **Directory**, or click on the **...** button and use the browse for folder window to choose the directory. Then, enable the **GOLD conf file** check box, and specify a filename `<filename.conf>`.

By enabling the appropriate check boxes it is also possible to save out the initialised protein structure file (see Files Containing the Initialised Protein and Ligand) and the cavity atoms file (see Defining the Binding Site).



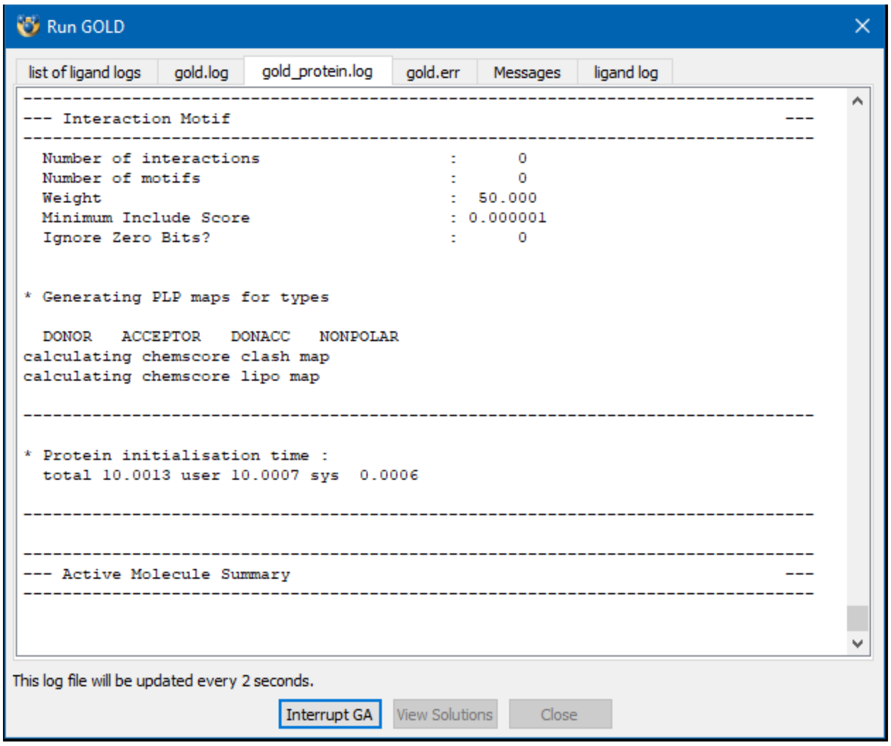
Directory: D:\CCDC

☑ GOLD conf file: gold.conf

☑ Protein(s): 1QBT_protein.mol2

□ Cavity atoms: cavity.atoms

If GOLD is run interactively, output that is written to the log files are displayed:



```
--------------------------------------------------------------------------------
--- Interaction Motif                                                        ---
--------------------------------------------------------------------------------
  Number of interactions                    :       0
  Number of motifs                          :       0
  Weight                                    :   50.000
  Minimum Include Score                     :  0.000001
  Ignore Zero Bits?                         :       0


* Generating PLP maps for types

  DONOR   ACCEPTOR   DONACC   NONPOLAR
calculating chemscore clash map
calculating chemscore lipo map

--------------------------------------------------------------------------------

* Protein initialisation time :
  total 10.0013 user 10.0007 sys  0.0006

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
--- Active Molecule Summary                                                  ---
--------------------------------------------------------------------------------
```

This log file will be updated every 2 seconds.

A full description of the output files produced by GOLD is available elsewhere (see Description of Output Files).

The parallel version only gives a summary as it is not possible to track multiple files.

If any error conditions are encountered, they will be displayed under the `gold.err` tab. Note that only fatal errors are reported for the parallel version.

You can use the **Interrupt GA** button in the **Run GOLD** window to interrupt and terminate the docking run.

Once the job is complete the docked ligand solutions can be viewed in the Hermes visualiser. To do this, click on the **View Solutions** button in the **Run GOLD** window.

15.3 Submitting a GOLD job to the Background

You can submit a GOLD job the background by using the **Run GOLD in The Background** button in the **GOLD Setup** window, having first specified all the required information, such as protein and ligand file names, parameter settings, etc.

15.4 Running GOLD from the Command Line

15.4.1 Windows platforms

GOLD can be run by starting a command prompt, navigating to the directory containing the `gold.conf` file and using a simple command available in `$GOLD_DIR\GOLD\gold\d_win32\bin`:

```
% gold_win32.exe gold.conf
```

where `gold.conf` is the name of a configuration file and `$GOLD_DIR=<Installation folder>\ccdc-software\gold\`

The `<Installation folder>` is most likely to be: `C:\Users\username\CCDC\`

Please use quotes to enclose the overall command, e.g.

```
“C:\Users\username\CCDC\ccdc-
        software\gold\GOLD\gold\d_win32\bin\gold_win32.exe"
gold.conf
```

15.4.2 Unix platforms

GOLD can be run directly in the background by using a simple command available in `$GOLD_DIR/bin`:

```
% gold_auto gold.conf &
```

where `gold.conf` is the name of a configuration file and `$GOLD_DIR=<Installation folder>/ccdc-software/gold/GOLD/`

The `<Installation folder>` is most likely to be: `/home/username/CCDC/`

15.4.3 macOS platforms

Please set `$GOLD_DIR` to e.g. `/<Installation folder>/ccdc-software/gold/GOLD` where `<Installation folder>` is most likely to be: `/Users/username/CCDC/`

Please make sure `$CSDHOME` is set to e.g. `/<Installation folder>/`

You can then run the gold_auto script, located in `$GOLD_DIR/bin`:

```
% gold_auto gold.conf &
```

15.5 Running GOLD in Parallel

For very large virtual-screening tasks, we provide the **GOLD Cloud** and **GOLD Cluster** tools. For smaller-scale parallelization, such as on a computational chemist's workstation or a local compute server, we do not have a specific product, but several options are available.

15.5.1 GOLD Cloud

This tool runs GOLD, inside a Docker container, on a Kubernetes cluster and is designed for use on commercial cloud platforms. Our experience is with Azure, but we are aware of customers who have used AWS and GCP. The GOLD Cloud tool and its User Guide are available to customers via our Downloads https://www.ccdc.cam.ac.uk/support-and-resources/downloads/ site under 'CSD-Discovery > GOLD'. A White Paper describing a practical application of the GOLD Cloud is also available, https://info.ccdc.cam.ac.uk/whitepaper-ultra-large-gold-docking-on-cloud-resources.

15.5.2 GOLD Cluster

This tool runs GOLD, inside a Singularity container, on an HPC compute cluster managed by Slurm. The GOLD Cluster tool and its User Guide are available to customers via our Downloads site, https://www.ccdc.cam.ac.uk/support-and-resources/downloads/, under 'CSD-Discovery > GOLD'.

15.5.3 Smaller-scale parallelization

GOLD Cloud can also be run locally using MiniKube https://kubernetes.io/docs/setup/learning-environment/minikube/ or Docker Desktop https://www.docker.com/products/docker-desktop. Thus, in principle, this tool may be used to parallelize GOLD docking. However, the setup involved isn't trivial and might not be straightforward on all platforms.

A simpler solution is to use the CSD Docking API, https://downloads.ccdc.cam.ac.uk/documentation/API/descriptive_docs/docking.html, and the Python standard library multiprocessing module, https://docs.python.org/3.7/library/multiprocessing.html, to parallelize GOLD docking. Example scripts using this approach are available at https://downloads.ccdc.cam.ac.uk/GOLD/HPC/gold_multiprocessing.zip; these should be suitable for docking some hundreds of ligands, depending on the compute resource available.

15.5.4 Platforms no longer supported

We no longer support using GOLD with Parallel Virtual Machine (PVM). We also do not currently provide tooling or support for running GOLD on Grid Engines, although there is nothing in principle that should prevent this being done.

16 Viewing and Analysing Results

16.1 Description of Output Files

16.1.1 Files Containing the Initialised Protein and Ligand

`gold_ligand.mol2` is the initialised ligand datafile with lone pairs added and the sets `DONOR_HYDROGENS` and `LONE_PAIRS` defined. If you do not wish to save this file, click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, select the **File Format Options** tab, then disable the **Save initialised ligand files** check-box.

`gold_protein.mol2` is the initialised protein datafile with lone pairs added to binding site atoms and the sets `DONOR_HYDROGENS` and `LONE_PAIRS` defined. The binding site is defined in the set `CAVITY_ATOMS`. These set-definitions in the `gold_protein.mol2` file are available for visualisation, as protein subsets, in Hermes.

16.1.2 Files Containing the Docked Ligand(s)

Each ligand will normally be docked several times, so a given input ligand will produce a set of files, each containing the results of a separate docking attempt.

Suppose that the original ligand file is `structure.mol2`. (this can contain more than one ligand, in which case each will be docked sequentially). As the GOLD job progresses, the result of each docking attempt is written out as `gold_soln_structure_m#_n.mol2`, where n is the solution number 1,2,3 ... and m# is the number of the ligand, i.e. m1 for the first ligand, m2 for the second, etc.

Note that the file `gold_soln_structure_m1_1.mol2` is not the best GOLD prediction, it is just the solution found in the first docking attempt. However, as GOLD proceeds, symbolic links are created: `ranked_structure_m#_1.mol2` will always point to the current top-ranked solution, `ranked_structure_m#_2.mol2` will point to the second-best solution, and so on.

Alternatively, you can specify that all saved docking solutions for all ligands are to be concatenated and written to a single file (see Specifying Ligand Solution File Formats and Directories).

Output files for the docked ligand(s) may also contain additional information such as the scoring function terms and the rotated protein hydrogen atom positions specific to that solution. This information can be written to sd file tags; for mol2 files, these tags are written to comment blocks. It is possible to control the information written to solution files (see Controlling the Information Written to Ligand Solution Files). A description of the various other tags available can be found in (see Appendix C: Additional Tags in Output Files).

Solution file title strings take the form

```
<file_basename>|<p>|[cov<r>]|dock<q>
```

where

`<file_basename>` is the base name of the ligand input file

<p> is the molecule number in the file

<q> is the number of the docking

<r> is the covalent attachment atom. This part is only printed for covalent dockings.

For example (mol2 file):

```
ligand|mol2|1|dock4
```

where the ligand filename is `ligand.mol2`, the structure is number 1 in the molecule input file, and the solution is from the fourth docking (dock4). The format for the output of the equivalent sd input file would be the following:

```
ligand|sd|1|dock4
```

To revert to the historic output, i.e. to output only the structure name, e.g.

```
N-phosphonacetyl-L-aspartate
```

the line `SET_UNIQUE_SOLN_TITLES = 1` in the `gold.params` file should be changed to read `SET_UNIQUE_SOLN_TITLES = 0`.

16.1.3 File Containing the Protein Binding-Site Geometry

During docking, GOLD will optimise hydrogen-bond geometries by rotating groups such as serine OH and lysine NH_3. It is also possible to allow specific protein sidechains to be treated as flexible during docking (see Side Chain Flexibility). This means that the coordinates of certain protein atoms such as these will change.

Protein atom positions that are generated during docking will usually be different for each docked ligand pose and are therefore written to the individual ligand solution files. This information can be written to sd file tags; for mol2 files, these tags are written to comment blocks (see Controlling the Information Written to Ligand Solution Files).

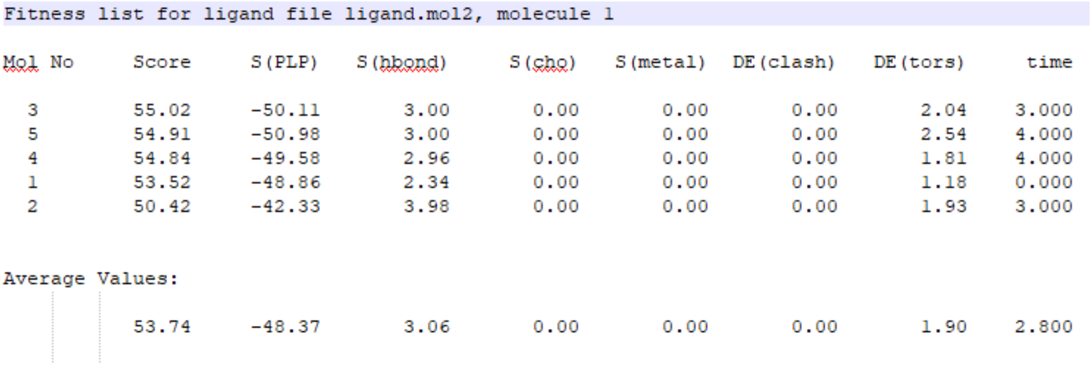
Structure files containing the optimised protein binding site geometry can be written out from the Hermes visualiser.

16.1.4 File Containing Ranked Fitness Scores for an Individual Ligand

A file called `<ligand_file_name>_m#.rnk` is written for each ligand (m# refers to the position of the ligand in the input file - remember that a given ligand input file may contain more than one ligand). This file contains a summary of the fitness scores for all the docking attempts on that ligand. The docking attempts are listed in decreasing order of fitness score, so the best solution is placed first.

The file gives total fitness scores and a breakdown of the fitness into its constituent energy terms.

The example file below corresponds to the first ligand in the input file `ligand.mol2` and is therefore called `ligand_m5.rnk`. The solution Mol No 5 corresponds to the file `gold_soln_ligand_m1_5.mol2`, which is symbolically linked to `ranked_ligand_m1_2.mol2`, since it is the second best of the docking attempts for this molecule:



```
Fitness list for ligand file ligand.mol2, molecule 1

Mol No      Score      S(PLP)    S(hbond)     S(cho)    S(metal)   DE(clash)    DE(tors)      time

  3         55.02      -50.11       3.00        0.00        0.00        0.00        2.04     3.000
  5         54.91      -50.98       3.00        0.00        0.00        0.00        2.54     4.000
  4         54.84      -49.58       2.96        0.00        0.00        0.00        1.81     4.000
  1         53.52      -48.86       2.34        0.00        0.00        0.00        1.18     0.000
  2         50.42      -42.33       3.98        0.00        0.00        0.00        1.93     3.000


Average Values:

            53.74      -48.37       3.06        0.00        0.00        0.00        1.90     2.800
```

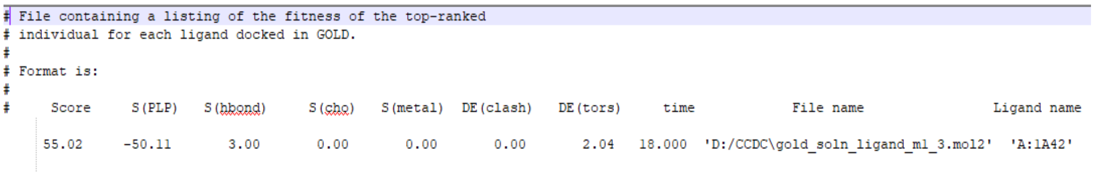
If you do not wish to save ligand rank files, click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, select the **File Format Options** tab, then disable the **Save ligand rank (.rnk) files** check-box.

16.1.5 File Containing Ranked Fitness Scores for a Set of Ligands

A file called `bestranking.lst` is written for batch jobs on multiple ligands. This gives a continuous summary of the best solution that has been obtained for each completed ligand.

To specify an alternative filename, click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, then select the **File Format Options** tab. Enable to **Use alternative bestranking.lst filename** check box and either enter the new path and filename, or click on the **...** button and use the file selection window to choose the file.

The file gives total fitness scores and a breakdown of the fitness into its constituent energy terms. The example file below was generated from a ligand input file containing one ligand. The listed file name corresponds to the name of the file containing the best solution found for each ligand, e.g. `gold_soln_ligand_m1_3.mol2` contains the best answer found for the first ligand in the input file.



```
# File containing a listing of the fitness of the top-ranked
# individual for each ligand docked in GOLD.
#
# Format is:
#
#     Score     S(PLP)   S(hbond)      S(cho)   S(metal)  DE(clash)   DE(tors)      time            File name                   Ligand name

    55.02     -50.11       3.00        0.00       0.00       0.00       2.04    18.000  'D:/CCDC\gold_soln_ligand_m1_3.mol2'  'A:1A42'
```

16.1.6 Rescore Solution File

A file containing the docked ligand solution(s) after rescoring can be written. You can control whether or not this file is written (see Rescore Output Files).

If specified, solutions will be written with the default filename `rescore.mol2`. To specify an alternative filename (for both the rescore solution and log files), add the following line to the `gold.conf` file:

```
concatenated_output = <filename.mol2>
```

For example, if `concatenated_output = Myfile.mol2` then the rescore mol2 file will be named `Myfile.mol2`.

Solution files will contain the new scoring function terms and the positions of rotatable protein hydrogen atoms generated during rescoring (see Rescore settings).

A full description of the additional tags written to solution output files is available in (see Appendix C: Additional Tags in Output Files).

16.1.7 Rescore Log File

The rescore log file `rescore.log` summarises the outcome of the rescoring run. To specify an alternative filename (for both the rescore solution and log files), add the following line to the `gold.conf` file:

```
concatenated_output = <filename.mol2>
```

For example, if `concatenated_output = Myfile.mol2` the log file will be named `Myfile.rescore.log`.

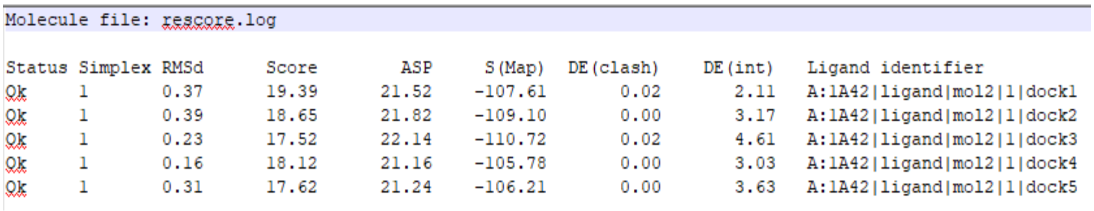
For each rescored ligand a total fitness score and the component scoring terms are listed.

`Status` gives an indication of whether or not there were any errors during the rescoring run.

`Simplex` indicates whether or not a locally optimised ligand pose was used for the rescoring. "**1**" indicates that the minimised pose was used, "**0**" indicates that the minimised pose was not used and "**-**" indicates that simplexing was not switched on (see Setting Up a Rescoring Run). Note: When **Perform local optimisation** (simplexing) is switched on, the minimised conformation will only be used for the rescoring if this results in an improvement to the fitness score.

When a minimised ligand pose is used for the rescoring an `RMSd` measure is given of the final minimised orientation with respect to the input ligand conformation.

The example file below was generated by rescoring all five solutions found for the first ligand in the solution files `gold_soln_ligand_m1_x.mol2`:



```
Molecule file: rescore.log

Status Simplex RMSd       Score       ASP      S(Map)  DE(clash)     DE(int)    Ligand identifier
Ok        1     0.37      19.39      21.52    -107.61       0.02        2.11    A:1A42|ligand|mol2|1|dock1
Ok        1     0.39      18.65      21.82    -109.10       0.00        3.17    A:1A42|ligand|mol2|1|dock2
Ok        1     0.23      17.52      22.14    -110.72       0.02        4.61    A:1A42|ligand|mol2|1|dock3
Ok        1     0.16      18.12      21.16    -105.78       0.00        3.03    A:1A42|ligand|mol2|1|dock4
Ok        1     0.31      17.62      21.24    -106.21       0.00        3.63    A:1A42|ligand|mol2|1|dock5
```

16.1.8 Protein Log File

The protein log file `gold_protein.log` details the parameterisation of the protein and the determination of the binding site.

The file is line buffered, so you can see how the algorithm is progressing even when GOLD is run in the background.

16.1.9 Ligand Log File

The progress of each genetic algorithm run is listed in the ligand log file `gold_<ligand_file_name>_m#.log`. Here, `m#` is an index to the number of the ligand in the input file, e.g. `m3` indicates that the log file refers to the third ligand in the input ligand file (remember that an input file may contain more than one ligand).

The log files are line buffered, so you can see how the algorithm is progressing even when GOLD is run in the background.

The parallel version of GOLD creates several temporary log files for each ligand, named `gold_soln_<ligand_file_name>_m#_<N>.log` where <N> is a docking-run number. Once all the docking runs for the ligand have been completed, these files are concatenated together into the single log file `gold_<ligand_file_name>_m#.log`.

The ligand log file contains information on:

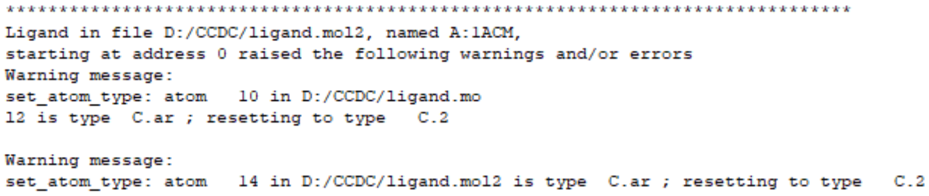
- The progress of each docking run (see Information on the Progress of Docking Runs).
- A comparison of the various docking solutions found (see Comparison of Docking Solutions).
- Clustering of ligand poses, for identification of solutions with different binding modes (see Identification of Different Binding Modes (Clustering of Ligand Poses)).

If you do not wish to save ligand log files, click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window, select the **File Format Options** tab, then disable the **Save ligand log files** check-box.

16.1.10 File Containing Error Messages

The file `gold.err` lists any errors found by the program. These are generally fatal and cause the program to stop. It is a good idea to check `gold.err` if something goes wrong.

Errors and warnings generated by the atom-type checker are also written to `gold.err`. If you are unsure about your atom typing you should therefore check this file. For example:



```
**********************************************************************************
Ligand in file D:/CCDC/ligand.mol2, named A:1ACM,
starting at address 0 raised the following warnings and/or errors
Warning message:
set_atom_type: atom   10 in D:/CCDC/ligand.mo
12 is type  C.ar ; resetting to type   C.2

Warning message:
set_atom_type: atom   14 in D:/CCDC/ligand.mol2 is type  C.ar ; resetting to type   C.2
```

In the parallel version, warning messages are logged in individual error files - one for each process. They are not sent back to the central parallel scheduling process.

`gold.err` is line buffered so errors are logged immediately.

16.1.11 Process File

The file `gold.pid` records the user, host and process number of the GOLD job. It is deleted when GOLD exits. Its purpose is to stop the user running two GOLD jobs in the same directory.

If the machine goes down, or GOLD crashes or is killed with signal 9, you will need to remove `gold.pid` before you can run another GOLD job in the same directory.

16.1.12 Seed Log File

A file called `gold.seed_log` is written to the output directory for each docking run.

GOLD uses a random number generator for some operations, e.g. choosing which genetic operator to use next or when to create the starting population of random individuals. The random number generator is normally initialised with random seeds; it is these seeds that are printed to the `gold.seed_log` file at the end of each docking run.

The seed file can be used to reproduce identical docking results for repeat runs (as long as all other settings are equal). To make use of the seed file in this way:

Copy the required `gold.seed_log` from the original output directory to an alternative location, e.g. the new docking directory.

Specify the location of the seed file in the `gold.params` file, i.e. open the `gold.params` via the **Edit** button next to **GOLD parameter file** under the **Fitness & Search Options** in the **GOLD Setup** dialogue. Find the lines that read

```
# Read seeds from SEED_FILE if not equal to none. Used for
debugging.
SEED_FILE = none
```

Change the `SEED_FILE = none` setting to include the full path to your seed file, e.g.

```
SEED_FILE = /home/username/new_docking_dir/gold.seed_log
```

Then run the docking using the modified `gold.params` file.

16.2 Information on the Progress of Docking Runs

As each docking run is performed on a ligand, the progress of the genetic algorithm is recorded in the ligand log file (see Ligand Log File).

The best (most fit) individual at any time is listed. The total fitness and its component terms are also displayed.

During a docking run, the fitness score may appear to get worse as the docking proceeds. This is due to the fact that the effects of poor H-bond geometry and close nonbonded contacts are artificially down-weighted at early stages of the docking (annealing). Only the final fitness score (i.e. from the completed docking) has any meaning.

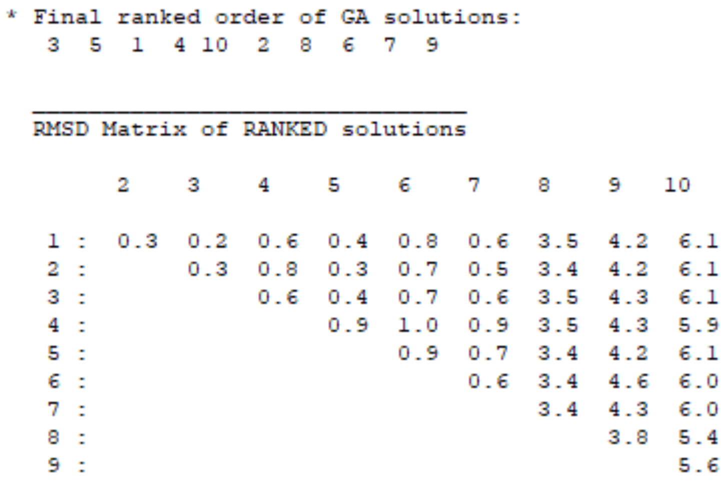
The message Reordering... refers to a re-ranking of the GA populations caused by the annealing process.

At the end of the GA run, the solution is output and summarised.

16.3 Comparison of Docking Solutions

Following the completion of all docking runs on a ligand, the results from the different runs are compared in the ligand log file.

The file will include a matrix of RMS deviations between the various docked ligand positions. The rms deviation algorithm takes account of symmetry effects, using a graph isomorphism algorithm. For example:



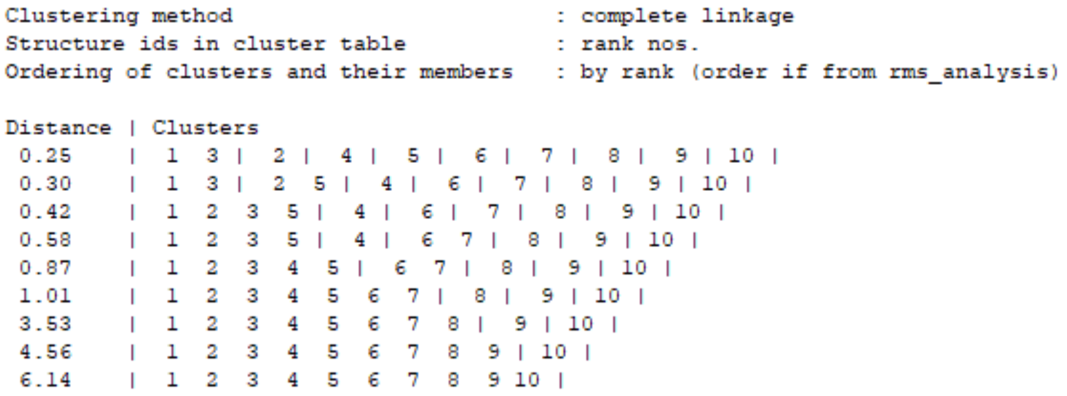
```
* Final ranked order of GA solutions:
   3  5  1  4 10  2  8  6  7  9

  ________________________________
  RMSD Matrix of RANKED solutions

        2     3     4     5     6     7     8     9    10

   1 :  0.3   0.2   0.6   0.4   0.8   0.6   3.5   4.2   6.1
   2 :        0.3   0.8   0.3   0.7   0.5   3.4   4.2   6.1
   3 :              0.6   0.4   0.7   0.6   3.5   4.3   6.1
   4 :                    0.9   1.0   0.9   3.5   4.3   5.9
   5 :                          0.9   0.7   3.4   4.2   6.1
   6 :                                0.6   3.4   4.6   6.0
   7 :                                      3.4   4.3   6.0
   8 :                                            3.8   5.4
   9 :                                                  5.6
```

In this case, solution number 3 had the largest fitness score (this solution will be in `gold_soln_ligand_m#_3.mol2`, which will be symbolically linked to `ranked_ligand_m#_1.mol2`), while solution number 9 had the worst fitness.

The numbers in the matrix of rms deviations refer to the rankings, not the run numbers (e.g. row 1 of the above matrix refers to the solution with the best fitness score, contained in `ranked_ligand_m#_1.mol2`).

Finally, the rms deviations are used as input to a hierarchical cluster analysis, using the complete linkage algorithm. Each line shows one iteration of the clustering algorithm, the distance between the clusters that were merged at that step, and the contents of the current set of clusters.

Clusters are separated by the '|' symbol and rankings are used rather than run numbers. For example, the solutions `ranked_ligand_m#_1.mol2` and `ranked_ligand_m#_3.mol2` were merged in the first step of the following cluster analysis:



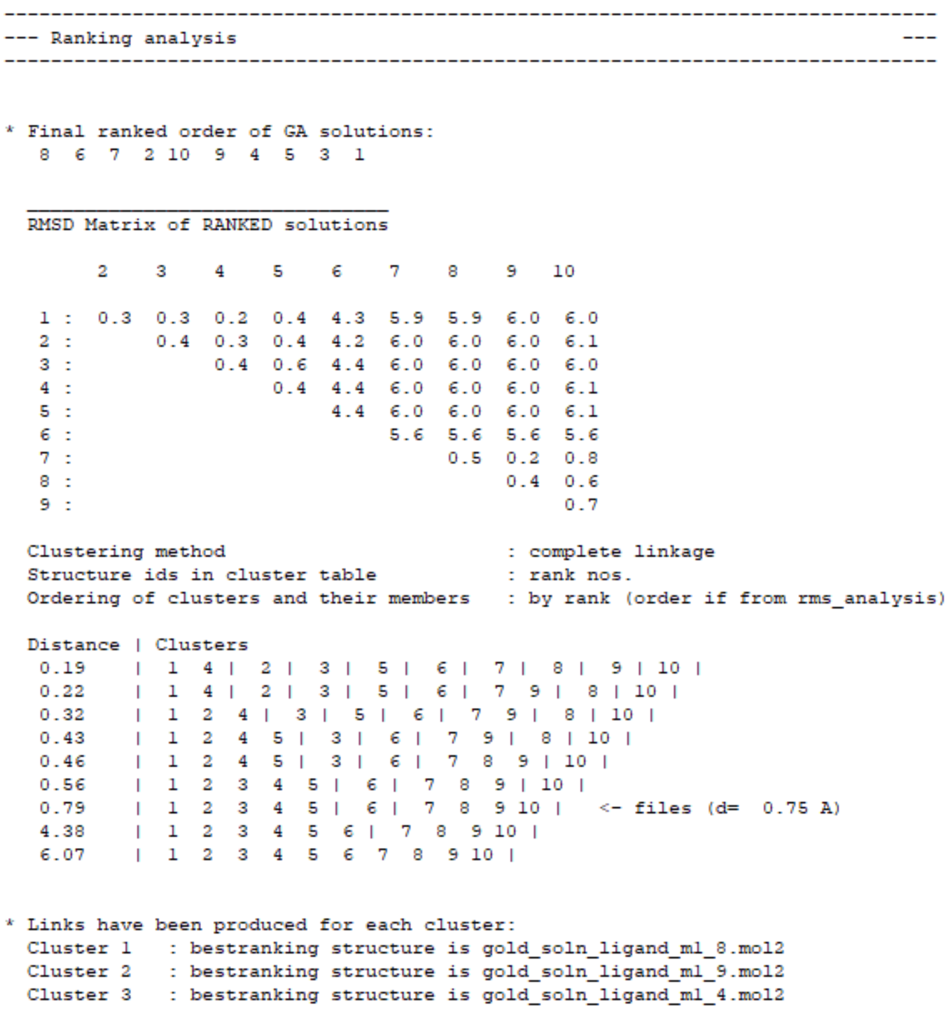
```
 Clustering method                         : complete linkage
 Structure ids in cluster table            : rank nos.
 Ordering of clusters and their members    : by rank (order if from rms_analysis)

 Distance  | Clusters
  0.25     |  1  3 |  2 |  4 |  5 |  6 |  7 |  8 |  9 | 10 |
  0.30     |  1  3 |  2  5 |  4 |  6 |  7 |  8 |  9 | 10 |
  0.42     |  1  2  3  5 |  4 |  6 |  7 |  8 |  9 | 10 |
  0.58     |  1  2  3  5 |  4 |  6  7 |  8 |  9 | 10 |
  0.87     |  1  2  3  4  5 |  6  7 |  8 |  9 | 10 |
  1.01     |  1  2  3  4  5  6  7 |  8 |  9 | 10 |
  3.53     |  1  2  3  4  5  6  7  8 |  9 | 10 |
  4.56     |  1  2  3  4  5  6  7  8  9 | 10 |
  6.14     |  1  2  3  4  5  6  7  8  9 10 |
```

16.4 Identification of Different Binding Modes (Clustering of Ligand Poses)

GOLD clusters docked solutions according to how similar the poses are in terms of their RMSd (see Comparison of Docking Solutions). A link can be generated to the top ranked solution from each distinct cluster. This can be useful in identifying different ligand binding modes. Considering solutions from different clusters is often more relevant than taking the top n ranked poses since these will often be very similar (i.e. all from the same cluster of solutions).

Click on **Output Options** from the list of **Global Options** given on the left of the **GOLD Setup** window and enable the **Create links for different binding modes (based on RMSD clustering)** check box, and specify the **Distance between clusters** (this determines how similar the poses are in each cluster of solutions). By default the clustering distance is 0.75 Å.

A clustering report is given at the end of the ligand log file (see Ligand Log File). The clusters themselves and the individual solutions within each cluster are in ranked order (i.e. the first member of the first cluster is always the top-ranked solution). For example, output from a run of 10 GA dockings may look like:



```
--------------------------------------------------------------------------------
--- Ranking analysis                                                         ---
--------------------------------------------------------------------------------


* Final ranked order of GA solutions:
   8   6   7   2  10   9   4   5   3   1

  ______________________________
  RMSD Matrix of RANKED solutions

        2    3    4    5    6    7    8    9   10

   1 :  0.3  0.3  0.2  0.4  4.3  5.9  5.9  6.0  6.0
   2 :       0.4  0.3  0.4  4.2  6.0  6.0  6.0  6.1
   3 :            0.4  0.6  4.4  6.0  6.0  6.0  6.0
   4 :                 0.4  4.4  6.0  6.0  6.0  6.1
   5 :                      4.4  6.0  6.0  6.0  6.1
   6 :                           5.6  5.6  5.6  5.6
   7 :                                0.5  0.2  0.8
   8 :                                     0.4  0.6
   9 :                                          0.7

  Clustering method                         : complete linkage
  Structure ids in cluster table            : rank nos.
  Ordering of clusters and their members    : by rank (order if from rms_analysis)

  Distance | Clusters
   0.19    |  1  4 |  2 |  3 |  5 |  6 |  7 |  8 |  9 | 10 |
   0.22    |  1  4 |  2 |  3 |  5 |  6 |  7  9 |  8 | 10 |
   0.32    |  1  2  4 |  3 |  5 |  6 |  7  9 |  8 | 10 |
   0.43    |  1  2  4  5 |  3 |  6 |  7  9 |  8 | 10 |
   0.46    |  1  2  4  5 |  3 |  6 |  7  8  9 | 10 |
   0.56    |  1  2  3  4  5 |  6 |  7  8  9 | 10 |
   0.79    |  1  2  3  4  5 |  6 |  7  8  9 10 |   <- files (d=  0.75 A)
   4.38    |  1  2  3  4  5  6 |  7  8  9 10 |
   6.07    |  1  2  3  4  5  6  7  8  9 10 |


* Links have been produced for each cluster:
  Cluster 1   : bestranking structure is gold_soln_ligand_ml_8.mol2
  Cluster 2   : bestranking structure is gold_soln_ligand_ml_9.mol2
  Cluster 3   : bestranking structure is gold_soln_ligand_ml_4.mol2
```

In the above example, at a clustering distance of 0.75 Å, there are three different clusters of solutions:

```
0.79 | 1 2 3 4 5 | 6 | 7 8 9 10 | <- files (d = 0.75
Å)
```

Clusters are separated by the '|' symbol and rankings are used rather than run numbers (see Files Containing the Docked Ligand(s)).

The first cluster contains five solutions ranked numbers 1, 2, 3, 4 and 5, the bestranking structure in this cluster is `ranked_structure_m#_1.mol2` which corresponds to the docked solution `gold_soln_ligand_m1_8.mol2`. Likewise, the second cluster contains one solution ranked number 6, the bestranking structure in this cluster is `ranked_structure_m#_6.mol2` which corresponds to the docked solution `gold_soln_ligand_m1_9.mol2`, and so on for the third cluster.

Symbolic links will be generated in the output directory which will link to the top-ranked solution in each cluster:

Cluster 1: bestranking structure is `gold_soln_ligand_m1_8.mol` Cluster 2: bestranking structure is `gold_soln_ligand_m1_9.mol2` Cluster 3: bestranking structure is `gold_soln_ligand_m1_4.mol2`

GOLD can be set up to generate diverse solutions, based on cluster size and RMSD (see Generating Diverse Solutions).

16.5 Viewing Docked Solutions in Hermes

Once the job is complete, to visualise docked solutions in the Hermes visualiser, click on the **View Solutions** button in the **Run GOLD** window.

Within Hermes, the docking poses from GOLD docking jobs can be navigated and visualised alongside the associated protein model using the **Docking Solutions** pane of the **Molecule Explorer** window. It is possible to display only the ligand and the cavity atoms that were used during the docking (including active waters if present) by enabling the tickbox adjacent to **Show only cavity and ligand** at the bottom of this window. A Numerical data associated with the solutions, such as the fitness score and its components, are tabulated as columns within the **Docking Solutions** pane. The data can be sorted according to selected data columns. Poses can be grouped. Poses may also be manually selected and can then be re-exported with a tailorable number of fields of associated data.

It is also possible in Hermes to further describe the docking poses by calculating additional descriptors for them. The descriptors quantify, amongst other things:

- The hydrogen-bonding interactions that occur between protein and docked ligand.
- H-bond interactions that do not occur, e.g. a protein H-bond donor that is prevented from forming a hydrogen bond by a ligand hydrophobic group.
- Other close contacts between protein and ligand.
- The buried surface area of the ligand, or of certain types of atoms in the ligand (e.g. hydrophobic atoms).
- Whether particular regions of the binding site are occupied by the ligand.
- Simple properties such as the number of H-bonding ligand atoms, molecular weight of ligand, number of rotatable bonds.

For further information on Visualising and Refining Selections of Docking Poses and on Defining and Calculating Descriptors please refer to the Hermes User Guide.

17 Saving and Reusing Docking Settings

17.1 Saving and Re-using Program Settings in Configuration Files

The configuration file is a text file which specifies the GOLD calculation that is to be run, including details of the ligand, the protein binding site, the fitness-function parameter file to be used, the torsion distribution file to be used, and the genetic algorithm parameters. Although the file can be generated with a standard text editor, the easiest way to create it is to use the GOLD front end.

Any settings that have been defined in the GOLD interface can be saved as a configuration file by selecting the **Save** button located next to the **Conf file** entry box at the top of the **GOLD Setup** window. Alternatively, you will be prompted to save the file if you start a GOLD job from the interface by selecting either **Run GOLD** or **Run GOLD in The Background**.

By default, the configuration file will be saved in the directory from which GOLD was opened and will be called `gold.conf`. Use the **Conf file** entry box at the top of the **GOLD Setup** window to change the file name and/or directory (any file name can be used).

Once a configuration file has been created, it can be re-used, either as a quick way of reading program settings into the GOLD front end or to run GOLD from the command line (see Running GOLD).

To load a previously created configuration file into the GOLD interface, enter the file name into the **Conf file** entry box at the top of the **GOLD Setup** window. Alternatively, click on the **Load** button and use the file selection window to choose the file. The parameters read in from the configuration file will overwrite any parameters that have already been set in the GOLD front end.

If you have a valid configuration file (i.e. one that completely specifies a GOLD job), you can run GOLD from the command line by using a simple command available in `$GOLD_DIR/bin`. For example, if the configuration file is `gold.conf`, the command is:

```
% gold_auto gold.conf &
```

17.2 Using Configuration File Templates

The configuration file is a text file which fully specifies the GOLD job that is to be run.

Once a configuration file has been created, it can be saved and re-used, either as a quick way of reading program settings into the GOLD front end or to run GOLD from the command line (see Saving and Re-using Program Settings in Configuration Files).

Configuration file templates can be used. These contain recommended setting for a number of different docking protocols.

As reported in Pose Prediction and Virtual Screening Performance of GOLD Scoring Functions in a Standardised Test (see References), validation experiments carried out on the DUD sets of actives/decoys have suggested preferred virtual screening protocols for several protein target classes. These protocols show good early enrichment statistics but also have the best trade-off between speed and accuracy, as far as can be ascertained. Protocols for nuclear hormone receptors, kinases, metallo-proteases and folate containing enzymes have been created out of this work. The protocol for serine proteases has been changed to allow use of a faster scoring function for docking.

Although these protocols are currently believed to be optimum they should be used with care as the datasets used to derive these protocols are small. Other protocols may work better for individual target proteins. It is recommended that ChemPLP be used for target classes not mentioned here.

To load a template configuration file, click on **Templates** from the list of **Global Options** given on the left of the **GOLD Setup** window. Select the template you wish to use from the list of available templates, then click on the **Load Template** button.

Note that configuration file templates are independent of the protein and ligand input files, so these will need to be specified in the usual way before running the docking.

17.3 Customising Scoring Function Parameters

Empirical parameters used in the fitness function (hydrogen bond energies, atom radii and polarisabilities, torsion potentials, hydrogen bond directionalities, etc.) are taken from the GOLD parameter file. These parameters are independent of the scoring function being used. Parameters can be customised by copying the file, editing the copy, and instructing GOLD to use the edited file (see Altering GOLD Parameters: the gold.params File).

A scoring function specific parameters file is also used. For GoldScore this is called `goldscore.params`. Parameters within this file can also be modified (see Altering GoldScore Fitness-Function Parameters: the goldscore.params File).

The ChemScore fitness-function parameters are stored in the `chemscore.params` file, which can also be customised (see Altering ChemScore Fitness-Function Parameters; the ChemScore File).

The Astex Statistical Potential (ASP) fitness-function parameters are stored in the `asp.params` file; this can also be customised (see Altering ASP Fitness-Function Parameters: the asp.params File).

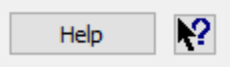
The ChemPLP fitness-function parameters are stored in the `chemplp.params` file; this can also be customised respectively (see Altering PLP Fitness-Function parameters).

17.4 Customising the Torsion Angle Distribution File

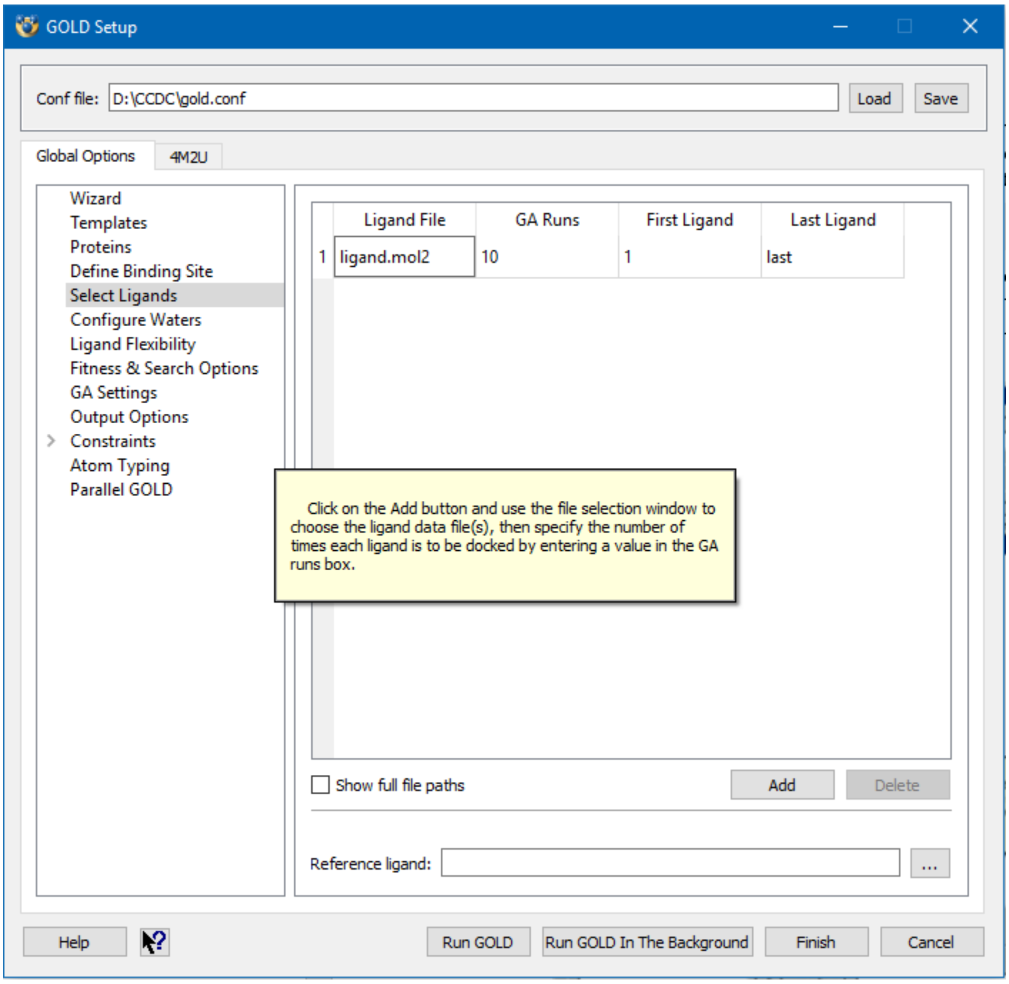
It is possible to customise torsion distribution information by copying one of the standard torsion distribution files, editing it, and instructing GOLD to use the edited file (see Editing Torsion Angle Distribution Files).

18 Help and Balloon Help

Help is available by clicking the **Help** button located at the bottom-left corner of the **GOLD Setup** window. Clicking this button will result in the GOLD User Guide being opened.



Balloon help is also available by clicking on the **?** icon located at the bottom-left corner of the **GOLD Setup** window then clicking on an option within the interface. For example, selecting **?** and clicking within the **Select Ligands** page under **Global Options** brings up the following help window:



Conf file: D:\CCDC\gold.conf

	Ligand File	GA Runs	First Ligand	Last Ligand
1	ligand.mol2	10	1	last

☐ Show full file paths

Reference ligand:

Click on the Add button and use the file selection window to choose the ligand data file(s), then specify the number of times each ligand is to be docked by entering a value in the GA runs box.

Balloon help can also be viewed by right-clicking at certain locations in the interface and selecting **What's this?** from the resulting menu.

19 Utility Programs To Set Up Docking Runs and Analyse Their Results in Batch Mode

These utility programs can be found in the following locations:

Windows:

`rms_analysis_win32.exe`, `check_mol2_win32.exe`, `smart_rms_win32.exe` and `gold_utils.exe` are located in:

```
<Installation folder>\ccdc-software\gold\GOLD\gold\d_win32\bin
```

where the `<Installation folder>` is likely to be: `C:\Users\username\CCDC\`

`gold_utils.exe` is located in:

```
<Installation folder>\ccdc-software\gold\GOLD\gold\d_win32\bin
```

where the `<Installation folder>` is likely to be: `C:\Users\username\CCDC\`

Linux:

`rms_analysis`, `check_mol2`, `smart_rms` are located in:

```
<Installation folder>/ccdc-software/gold/GOLD/utilities/
```

with gold_utils located in:

```
<Installation folder>/ccdc-software/gold/GOLD/
```

where the `<Installation folder>` is likely to be: `/home/username/CCDC/`

macOS:

`rms_analysis`, `check_mol2`, `smart_rms`and `gold_utils` are located in:

```
<Installation folder>/ccdc-software/gold/GOLD/utilities/
```

where the `<Installation folder>` is likely to be: `/Users/username/CCDC/`

Note: On macOS, before using any of these utility programs, you will need to set the GOLD_DIR environment variable:

```
GOLD_DIR=/<Installation folder>/ccdc-software/gold/GOLD/
```

where the `<Installation folder>` is likely to be: `/Users/username/CCDC/`

Note: When calling a utility program please make sure your system finds the paths to the program location specified here. This may be done by either explicitly specifying the path or setting an appropriate systems variable. On Windows, use quotes “” to enclose the full paths, e.g:

```
“C:\Users\username\CCDC\ccdc-
        software\gold\GOLD\gold\d_win32\bin\smart_rms_win32.exe”
```

19.1 gold_utils

The `gold_utils.exe` (Windows) or `gold_utils` (Linux and macOS) executable includes:

- `-protonate`: addition of hydrogen atoms outside the Hermes graphical interface.
- `-convert`: conversion of PDB-files to the .mol2 format required for docking.
- `-print_rotamer`: print rotamer library parameters required to set up docking with flexible side chains.
- `-write_complexes`: write out the protein coordinates for each docked ligand including protein movements. This facilitates post-processing in third party software.

19.1.1 gold_utils -protonate

This utility is used for protonating molecule file(s).

Usage:

```
gold_utils/gold_utils.exe -protonate -i <filename> -o
<filename> [-rules <filename>]
```

Use `gold_utils` for Linux and macOS, and `gold_utils.exe` for Windows.

Details of the arguments above are:

- `-protonate`: required argument that instructs the script to protonate the supplied molecule file.
- `-i`: required argument that specifies the input molecule file.
- `-o`: required argument that specifies the output molecule file.
- `-rules`: optional argument that forces the script to use the specified protonation rules file (see Applying Protonation Rules).

19.1.2 gold_utils -print_rotamer

This utility is used for printing the rotamer block for the specified amino acid residue.

Usage:

```
gold_utils/gold_utils.exe -print_rotamer -i <filename> [-o
        <filename>] -residue <residue id> [-ignore_library] [-
        reference_ligand <filename>]
```

Use `gold_utils` for Linux and macOS, and `gold_utils.exe` for Windows.

Details of the arguments above are:

- `-print_rotamer`: required argument that instructs the script to print he rotamer block for the specified amino acid residue (for inclusion in a GOLD .conf file).
- `-i`: required argument that specifies the input molecule file.
- `-o`: optional argument that specifies the output molecule file (which will contain the rotamer block).
- `-residue`: required argument that specifies the amino acid residue to use, including the chain ID e.g. ASP102A.
- `-ignore_library`: optional argument that instructs the script to ignore the rotamer library and set all torsions to fully rotate.
- `-reference_ligand`: optional argument that instructs the script to use the specified reference ligand file to identify which chain to use (in case the unique residue cannot be identified from the chain ID).

19.1.3 gold_utils -convert

This utility enables file conversion, e.g. `.pdb` to `.mol2`.

Usage:

```
gold_utils/gold_utils.exe -convert -i <filename> -o
<filename>
```

Use `gold_utils` for Linux and macOS, and `gold_utils.exe` for Windows.

Details of the arguments above are:

- `-convert`: required argument that converts the file based on the formats given by the `-i` and `-o` name extensions.
- `-i`: required argument that specifies the input molecule file.
- `-o`: required argument that specifies the output molecule file.

19.1.4 gold_utils -write_complexes -conf

This file enables a set of protein-ligand complex files to be written out for a set of docking solutions.

Usage:

```
gold_utils/gold_utils.exe -write_complexes -conf <conf_filename>
        [-o <output_directory>] [-format <output_format>]
```

Use `gold_utils` for Linux and macOS, and `gold_utils.exe` for Windows.

Details of the arguments above are:

- `-write_complexes`: required argument that writes all protein-ligand complexes for a set of docking solutions.
- `-o`: optional argument that specifies the output directory (which must already exist).
- `-format`: optional argument that specifies the output molecule file format (if different from the default format, .mol2).

19.2 rms_analysis

`rms_analysis` can be used from the command line:

```
rms_analysis/rms_analysis_win32.exe -method [simple|complete|
        group_average] <file1>.mol2 <file2>.mol2 <file3>.mol2
        <file4>.mol2...
```

Use `rms_analysis` for Linux and macOS, and `rms_analysis_win32.exe` for Windows.

The utility calculates an RMS difference matrix for a set of docked ligands (as mol2 files) and performs hierarchical cluster analysis. A graph isomorphism algorithm is used to determine optimal RMSD values. Note that the utility is strictly for use with docked ligand poses only and not for protein-ligand complexes.

Choose `simple` for single linkage cluster analysis, `complete` for complete linkage, `group_average` for group average.

For example, the table of RMS deviations below for nine dockings of a molecule produces the following clustering with the complete linkage method:

	2	3	4	5	6	7	8	9
1	0.8	1.1	1.0	1.0	1.4	2.3	5.0	4.6

2		0.9	1.1	1.1	1.2	2.3	5.2	4.6
3			0.4	0.8	0.9	2.3	5.0	4.5
4				0.6	1.1	2.3	4.9	4.5
5					1.3	2.0	4.9	4.5
6						1.8	5.1	4.4
7							5.3	4.5
8								2.4

Step	Distance between clusters being merged	Clusters
1	0.40	1 / 2 / 3, 4 / 9 / 5 / 6 / 7 / 8
2	0.84	1 / 2 / 3, 4, 5 / 9 / 8 / 6
3	0.84	1, 2 / 7 / 3, 4, 5 / 9 / 8 / 6
4	1.13	1, 2, 3, 4, 5 / 7 / 6 / 9 / 8
5	1.42	1, 2, 3, 4, 5, 6 / 7 / 8 / 9
6	2.35	1, 2, 3, 4, 5, 6, 7 / 9 / 8
7	2.38	1, 2, 3, 4, 5, 6, 7 / 8, 9
8	5.28	1, 2, 3, 4, 5, 6, 7, 8, 9

19.3 check_mol2

check_mol2 uses the same algorithms as the main GOLD program to check the quality of the input ligand structure. It can either be used to simply generate output on the ligand of interest, or to create a corrected version of the ligand that can be used by GOLD. Note that check_mol2 only works on mol2 files of individual ligands. In other words, it will not work on a multi-mol2 file.

Usage:

```
check_mol2/check_mol2_win32.exe -i ligand.mol2 [-o
        corrected_ligand.mol2]
```

Use check_mol2 for Linux and macOS, and check_mol2_win32.exe for Windows.

Details of the arguments above:

- -i: required argument that specifies the original input ligand.

- `-o`: optional argument that specifies the corrected output ligand.

19.4 smart_rms

`smart_rms` calculates the RMSD of the overlap of two molecules in their current positions. It generates the outputs that GOLD generates for the `gold_ligand.mol2` file. The RMSD is output to the terminal.

Usage:

```
smart_rms/smart_rms_win32.exe [-a|hvcf[s solvacc_file <threshold>|
        gradual]] molecule1 molecule2
```

Use `smart_rms` for Linux and macOS, and `smart_rms_win32.exe` for Windows.

Details of the arguments above:

- `-a`: all atoms (the default)
- `-h`: heavy atoms only
- `-v`: verbose
- `-c`: atom type checking
- `-f`: force no atom matching (assumes atoms are in the same order in both molecule files, useful when comparing large molecules such as protein-ligand complexes)
- `-s`: requires a solvent-accessibility file and a threshold.

All atoms with solvent-accessibility less than `threshold` will be disregarded for the calculation of the solvent-accessible RMSD. If keyword `gradual` is specified after the solvent accessibility file (`solvacc_file`) instead of a `threshold`, the solvent accessiblities will be used to determine the weight of each atom in the solvent-accessible RMSD calculation.

`molecule1` and `molecule2`: required arguments that specify the two molecule input files whose overlap RMSD will be calculated.

20 References

20.1 GOLD references

Molecular Recognition of Receptor Sites Using a Genetic Algorithm with a Description of Desolvation G. Jones, P. Willett and R. C. Glen J. Mol. Biol., **245**, 43-53, 1995 [DOI: 10.1016/S0022-2836(95)80037-9].

Development and Validation of a Genetic Algorithm for Flexible Docking G. Jones, P. Willett, R. C. Glen, A. R. Leach and R. Taylor J. Mol. Biol., **267**, 727-748, 1997 [DOI: 10.1006/jmbi.1996.0897].

Protein-Ligand Docking and Virtual Screening with GOLD J. C. Cole, J. W. M. Nissink and R. Taylor in Virtual Screening in Drug Discovery (Eds. J. Alvarez, B. Shoichet), Taylor & Francis CRC Press, Boca Raton, Florida, USA (2005).

Modeling Water Molecules in Protein-Ligand Docking Using GOLD M. L. Verdonk, G. Chessari, J. C. Cole, M. J. Hartshorn, C. W. Murray, J. W. M. Nissink, R. D. Taylor, and R. Taylor J. Med. Chem., **48**, 6504-6515, 2005 [DOI: 10.1021/jm050543p].

Using Buriedness to Improve Discrimination Between Actives and Inactives in Docking N. M. O'Boyle, S. C. Brewerton and R. Taylor J. Chem. Inf. Model., **48**, 1269-1278, 2008 [DOI: 10.1021/ci8000452].

The Use of Protein-Ligand Interaction Fingerprints in Docking S. C. Brewerton Curr. Opin. Drug Discov. Devel. **11**, 356-364, 2008.

Empirical Scoring Functions for Advanced Protein-Ligand Docking with PLANTS O. Korb, T. Stützle and T. E. Exner J. Chem. Inf. Model., **49**, 84-96, 2009 [DOI: 10.1021/ci800298z].

20.2 References dealing with GOLD validation

A New Test Set for Validating Predictions of Protein-Ligand Interactions J. W. M. Nissink, C. Murray, M. Hartshorn, M. L. Verdonk, J. C. Cole and R. Taylor Proteins, **49**, 457-471, 2002 [DOI: 10.1002/prot.10232].

Improved Protein-Ligand Docking using GOLD M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray and R. D. Taylor Proteins, **52**, 609-623, 2003 [DOI: 10.1002/prot.10465].

Diverse, High-Quality Test Set for the Validation of Protein-Ligand Docking Performance M. J. Hartshorn, M. L. Verdonk, G. Chessari, S. C. Brewerton, W. T. M. Mooij, P. N. Mortenson and C. W. Murray J. Med. Chem., **50**, 726-741, 2007 [DOI: 10.1021/jm061277y].

Pose Prediction and Virtual Screening Performance of GOLD Scoring Functions in a Standardised Test J. W. Liebeschuetz, J. C. Cole and O. Korb. J. Comput. Aided Mol. Des, **50**, 737-748, 2012 [DOI: 10.1007/s10822-012-9551-4].

20.3 General GOLD/docking/virtual screening papers that may be of interest

Life-Science Applications of the Cambridge Structural Database R. Taylor Acta Cryst., **D58**, 879-888, 2002 [DOI: 10.1107/S090744490200358X].

Virtual Screening Using Protein-Ligand Docking: Avoiding Artificial Enrichment M. L. Verdonk, V. Berdini, M. J. Hartshorn, W. T. M. Mooij, C. W. Murray, R. D. Taylor and P. Watson J. Chem. Inf. Comput. Sci., **44**, 793-806, 2004 [DOI: 10.1021/ci034289q].

Comparing Protein-Ligand Docking Programs is Difficult J. C. Cole, C. W. Murray, J. W. M. Nissink, R. D. Taylor and R. Taylor Proteins, **60**, 325-332, 2005 [DOI: 10.1002/prot.20497].

Evaluating Docking Programs: Keeping the Playing Field Level J. W. Liebeschuetz J. Comput. Aided Mol., **22**, 229-238, 2008 [DOI: 10.1007/s10822-008-9169-8].

Testing Assumptions and Hypotheses for Rescoring Success in Protein-Ligand Docking. N. M. O’Boyle, J. W. Liebeschuetz and J. C. Cole J. Chem. Inf. Model., **49**, 1871-1878, 2009 [DOI:10.1021/ci900164f].

Docking Performance of Fragments and Drug-like Compounds M. L. Verdonk, I. Giangreco, R. J. Hall, O. Korb, P. N. Mortensen and C. W. Murray J. Med. Chem., **54**, 5422-5431, 2011 [DOI: 10.1021/jm200558u].

21 Acknowledgements

GOLD was written by Gareth Jones (University of Sheffield, UK) in a DTI LINK collaboration with GlaxoWellcome and the Cambridge Crystallographic Data Centre (CCDC).

Funding was provided by the Biotechnology and Biological Sciences Research Council, the Department of Trade and Industry, the Medical Research Council, GlaxoWellcome Ltd and the CCDC.

Peter Willett (University of Sheffield), Robert Glen (Wellcome), Andrew Leach (GlaxoWellcome) and Jacques Barbanton (Lipha Pharmaceuticals) are also thanked for significant contributions to the development of GOLD.

Implementation of the ChemScore, Heme, Kinase and Astex Statistical Potential scoring functions and the Diverse Solutions code within GOLD is copyright 2001-2007 Astex Therapeutics Ltd. We also thank Astex Technology Ltd for their contribution to the water modelling code. We also thank Astex Pharmaceuticals for their contribution to the pharmacophore constraint code.

One of the torsion libraries supplied with GOLD was developed by Gerhard Klebe and Thomas Mietzner (BASF).

22 Appendix A: Tutorials

In order to familiarise yourself with GOLD it is recommended that you work through the tutorial examples provided. Tutorial 1 will go through the process of setting up and running an example docking using the Docking Wizard in some detail, subsequent tutorials will be more concise but will introduce other, more advanced, aspects of the program.

Tutorial 1 illustrates how to set-up protein and ligand files simply using the Hermes visualiser. For more comprehensive functionality for setting up protein and ligand files (see Essential Steps for setting up the protein file and Essential Steps for setting up ligand files) we recommend you use a molecular modelling program. Full details of the software requirements needed in order to use GOLD are given elsewhere (see Introduction).

Please note: Due to the non-deterministic nature of GOLD, results may vary from those described in the tutorials.

Note also that extra tutorial material is available from our Documentation and Resources site, https://www.ccdc.cam.ac.uk/support-and-resources/ccdcresources/: search there for Resource Type ‘Tutorials’ and Product ‘GOLD’.

22.1 Tutorial 1: A Step-By-Step Guide to Using GOLD

22.1.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial1` to a directory to which you have write permissions.

GOLD features a **Wizard** for docking setup and an **Advanced** interface for users who are more familiar with using GOLD.

The Wizard guides the user through the key steps involved in setting up protein and ligand files, as well as the components that are key to running a successful docking.

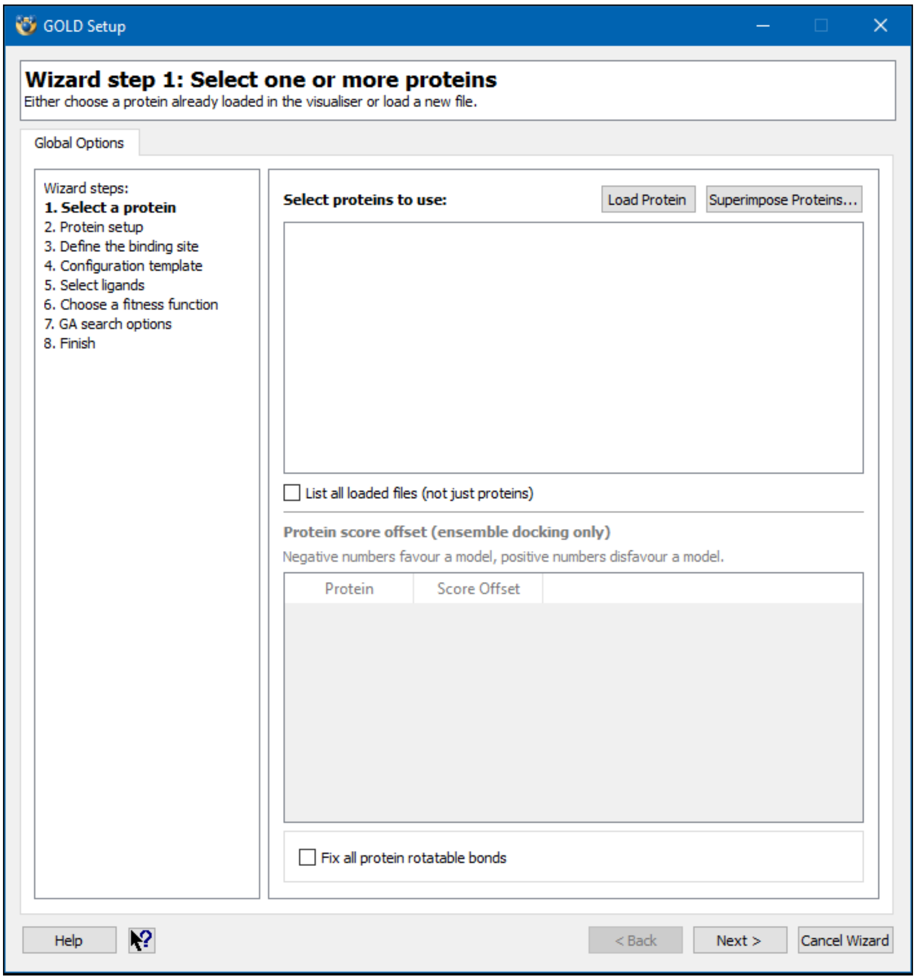
GOLD will only produce reliable results if the protein and ligand input files are set up correctly. It is therefore essential that a number of key steps are followed when preparing any input structure for use in GOLD (see Essential Steps for setting up the protein file and Essential Steps for setting up the ligand files).

This tutorial aims to provide a step-by-step guide to making the most of the GOLD Wizard. To illustrate this, the procedure for setting up a protein and ligand for use with GOLD and then the subsequent docking will be explained, and additional information will be provided on related issues.

In this example GOLD will be used to determine the binding mode of N-phosphonacetyl-L-aspartate with the aspartate carbamoyltransferase, PDB entry code 1acm.

22.1.2 Using the GOLD Wizard to Prepare the Protein File

Open Hermes. Open the GOLD setup wizard by clicking on the main menu option **GOLD**, then by picking **Wizard** from the resultant pull-down menu. The steps required to setup files for docking are listed down the left-hand side.



Wizard step 1: Select one or more proteins
Either choose a protein already loaded in the visualiser or load a new file.

Wizard steps:
1. Select a protein
2. Protein setup
3. Define the binding site
4. Configuration template
5. Select ligands
6. Choose a fitness function
7. GA search options
8. Finish

Select proteins to use:

□ List all loaded files (not just proteins)

Protein score offset (ensemble docking only)
Negative numbers favour a model, positive numbers disfavour a model.

□ Fix all protein rotatable bonds

22.1.3 Selecting a Protein

In the **Select a protein** step, read in the protein file, `1ACM.pdb`, by hitting the **Load Protein** button, navigating to the folder to where you copied the tutorial1 files, select the file and then clicking on **Open**. The protein file will be loaded into Hermes 3D view. You will notice that the protein C atoms are coloured grey (i.e. coloured by atom type) while the C atoms of any ligands are coloured green.

In Hermes you will notice the **Molecule Explorer** to the left-hand side of the Hermes 3D view. If the **Molecule Explorer** is not visible, click on **Display** in the Hermes main menu and select **Toolbars** then **Molecule Explorer...**.

Click on the "**>**" adjacent to **1ACM** and underneath **All Entries**. Protein structures can consist of the following components: **Chains**, **Nucleic Acids**, **Ligands**, **Cofactors**, **Metals** and **Waters**. When present in a given protein structure, each of these has a corresponding ">" adjacent to it. Each successive time the ">" is clicked on, the component it corresponds to is broken down further. In this way it is possible e.g. to identify specific protein residues or atoms in a ligand. Display styles, colours and labels and selection options are available by right-clicking on any of these components.

You will notice that this protein structure consists of 4 chains: A, B, C and D. The chain we will be focusing on is chain A.

In the GOLD Wizard click on the **Next** button to proceed to the **Protein setup** step. Click on the **1ACM** tab adjacent to the **Global Options** tab. From under the **1ACM** tab we can make some essential modifications to the protein, specifically we can:

- **Add hydrogens** (see Adding Hydrogen Atoms): all hydrogen atoms must be present in the protein input file (see Protonation and Tautomeric States). The hydrogen atoms are placed on the protein in order to ensure that ionisation and tautomeric states are defined unambiguously. Advanced options within this **Protein setup** step allow for switching between different tautomeric states for histidine residues, and flipping Asn and Gln side chains as required.

- **Delete waters** (see Deleting Waters): water molecules often play key roles in protein-ligand recognition. Water molecules can either form mediating hydrogen bonds between protein and ligand, or be displaced by the ligand on binding. Water molecules within the active site can be retained and allowed to toggle (i.e. switch on and off during docking), rotate and translate within a radius of 2 Å to optimise their H-bonding positions. Those outside of the binding site can be removed from the protein altogether.

- **Delete ligands** (see Extracting and Deleting Ligands): the `1ACM.pdb` protein is the raw PDB file which is the original protein-ligand complex. For GOLD to effectively dock a ligand back into the active site, the co-crystallised ligand must first be removed (i.e. the binding site must be empty).

22.1.4 Adding Hydrogen Atoms

From within the **1ACM** tab, add hydrogens to the protein by selecting the **Add Hydrogens** button from the first **Protonation & Tautomers** option. Note that this may take a little time depending on how many other processes are running on your computer.

Once the H atoms have been added a pop-up window will inform you that 7192 hydrogens were added. Click **OK** to close the pop-up window.

22.1.5 Deleting Waters

Still in the **1ACM** window, hit the **Extract/Delete Waters** option underneath **Protonation & Tautomers**. From within this dialogue it is possible to specify water molecules that mediate protein-ligand interactions (i.e. active waters), and to delete those that are not required (see Water Molecules). There are not many water molecules co-crystallised with the protein and they are not needed for the purposes of this tutorial, thus they can be deleted.

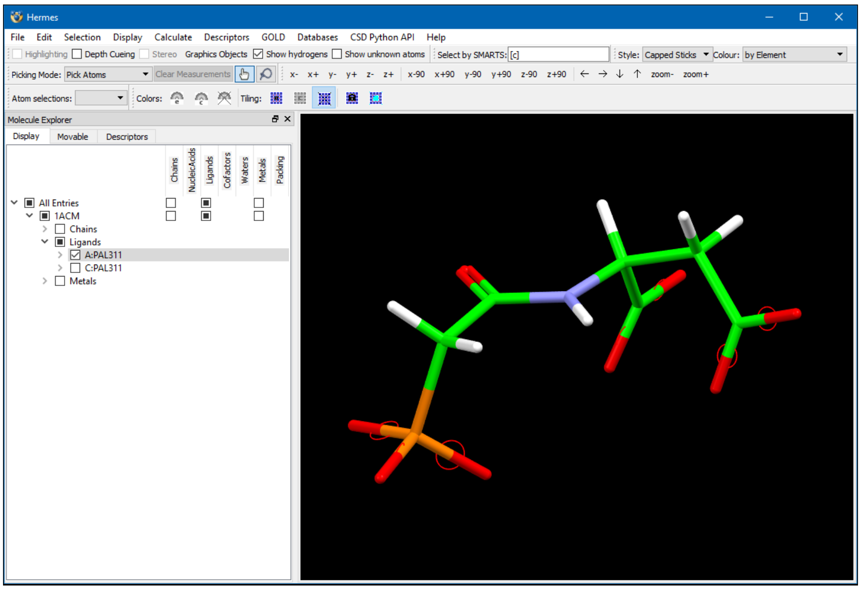
Hit the **Delete Remaining Waters** button. When prompted Are you sure you want to delete all the waters? hit **OK**. You will be informed that 15 waters have been deleted.

22.1.6 Extracting and Deleting Ligands/Cofactors

Before extracting ligands/cofactors, it is important to ensure the protonation states are correct. Hermes makes a best guess at protonation states, however as PDB files don't contain atom type information it is not unexpected that Hermes occasionally gets it wrong.

The protein structure in this example contains only ligands, not cofactors.

Minimise the **GOLD Setup** window and return to the 3D view. The most convenient way of editing the ligand structure is to display only the ligand. We are going to be docking into chain A, so hide all **Chains**, **Ligands** and **Metals** apart from Ligand **A:PAL311** by expanding the **Ligands** tree, then deactivating all tickboxes apart from that adjacent to **A:PAL311.** You may need to right-click on it and choose **Center & Zoom 3D View**:



☐ Highlighting ☐ Depth Cueing ☐ Stereo Graphics Objects ☑ Show hydrogens ☐ Show unknown atoms

Select by SMARTS: [c]

Style: Capped Sticks

Colour: by Element

Picking Mode: Pick Atoms

Atom selections:

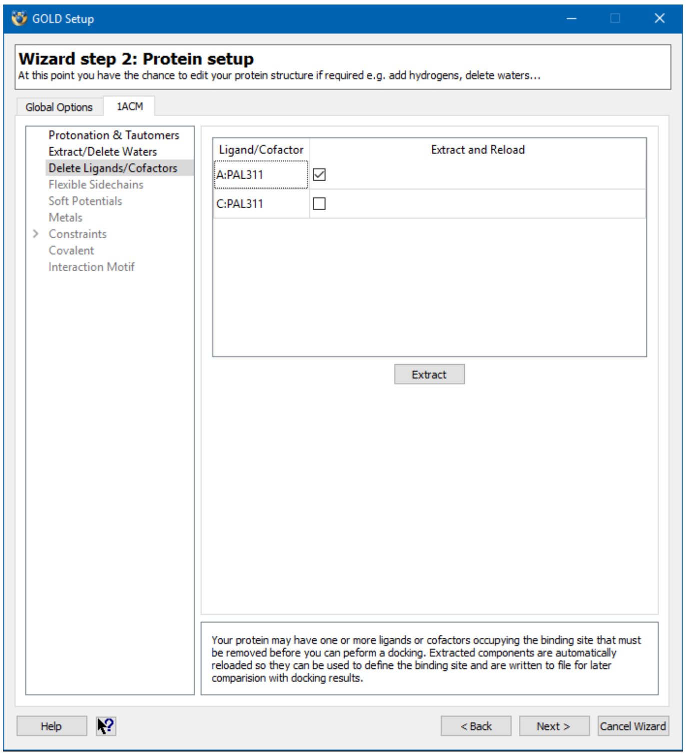
- ■ All Entries
  - ■ 1ACM
    - ☐ Chains
    - ■ Ligands
      - ☑ A:PAL311
      - ☐ C:PAL311
    - ☐ Metals

Certain groups can be represented in more than one way (i.e. have more than one canonical form), such as nitro, carboxylate and amidinium. In such cases, there is usually a right and a wrong representation for use in GOLD. The conventions used for some common difficult groups and further help on setting up the ligand is provided (see Setting Up Ligands).

The phosphate and carboxylate groups are unprotonated and in each case the bond to the O atom has been assigned as aromatic (displayed with the red circle across these bonds, when the option in **Display** > **Styles** > **Display Aromatic Rings** is ticked). In this case these groups have been correctly handled by Hermes and so no editing is required. Had modifications been required, editing functionality is available under **Edit**, **Edit Structure**.

Re-display all the **Chains**, **Ligands** and **Metals** tickboxes so that the 3D view displays the entire protein structure.

In the Protein setup window, click on the **Delete Ligands/Cofactors** option beneath the **Extract/Delete Waters** option. This window enables us to extract and delete a ligand from the protein in order to set it up for docking. As we are going to be docking into chain A we need to first remove the co-crystallised ligand from its active site: activate the tickbox adjacent to **A:PAL311**, underneath the **Extract and Reload** header:



Wizard step 2: Protein setup
At this point you have the chance to edit your protein structure if required e.g. add hydrogens, delete waters...

Ligand/Cofactor	Extract and Reload
A:PAL311	☑
C:PAL311	☐

Your protein may have one or more ligands or cofactors occupying the binding site that must be removed before you can peform a docking. Extracted components are automatically reloaded so they can be used to define the binding site and are written to file for later comparision with docking results.

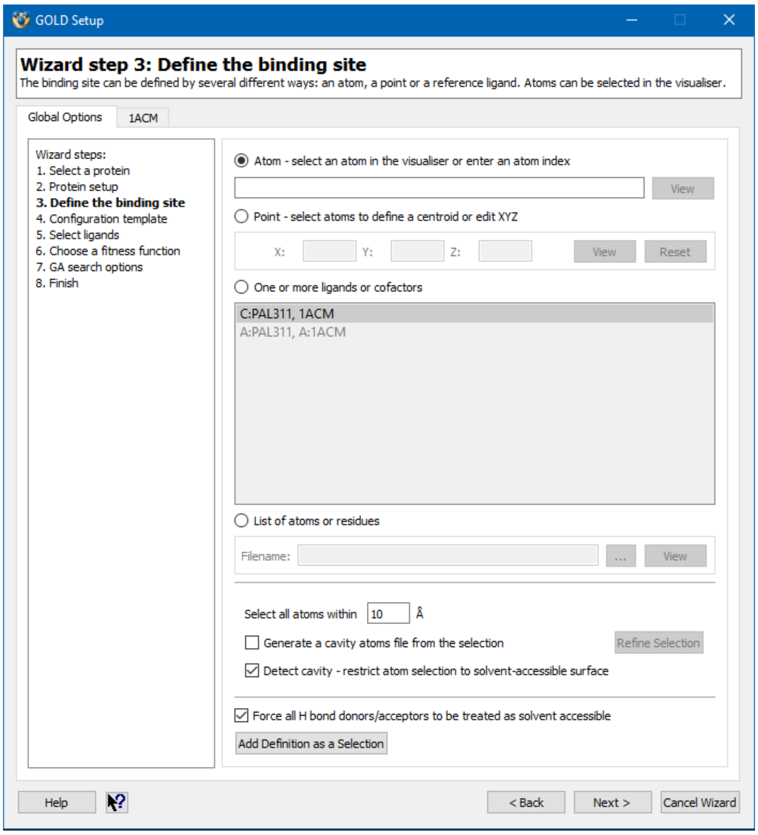
Now hit the **Extract** button. When prompted save the ligand file e.g. as `ligand.mol2` in the folder to which you copied the tutorial1 files. The reason for this is that all the files listed in the section [Analysis of Output](#) will therefore have the correct name, and in all sections below it too.

Return to Hermes 3D view: you will notice the ligand file has been re-loaded separately as the **A:1ACM** entry, alongside the protein entry in the **Molecule Explorer**.

Return to the GOLD Wizard and click on the **Global Options** tab.

Hit **Next** to proceed to the **Define the binding site** step.

22.1.7 Defining the Protein Binding Site



GOLD Setup

Wizard step 3: Define the binding site

The binding site can be defined by several different ways: an atom, a point or a reference ligand. Atoms can be selected in the visualiser.

Wizard steps:
1. Select a protein
2. Protein setup
3. **Define the binding site**
4. Configuration template
5. Select ligands
6. Choose a fitness function
7. GA search options
8. Finish

● Atom - select an atom in the visualiser or enter an atom index

○ Point - select atoms to define a centroid or edit XYZ

X: Y: Z:

○ One or more ligands or cofactors

C:PAL311, 1ACM
A:PAL311, A:1ACM

○ List of atoms or residues

Filename:

Select all atoms within 10 Å

□ Generate a cavity atoms file from the selection

☑ Detect cavity - restrict atom selection to solvent-accessible surface

☑ Force all H bond donors/acceptors to be treated as solvent accessible

It is necessary to specify the approximate centre and extent of the protein binding site; this can be done in a number of ways from within the **Define the binding site** window, including:

- From a protein atom (see [Defining a Binding Site from an Atom](#)).
- From a point (see [Defining a Binding Site from a Point](#)).
- From a reference ligand or cofactor (see [Defining a Binding Site from a Reference Ligand](#)).

- From a file containing a list of atoms or residues (see Defining a Binding Site from a List of Atoms or Residues).

For this example, we are going to use a set of coordinates to define the binding site. The coordinates can either be input as x,y,z values, or the centroid of a user-selected group of atoms can be used. We will use the former.

Activate the radio button that reads **Point - select atoms to define a centroid or edit XYZ**. The centre of the binding site in 1acm can be found at 42.409, 29.242, 16.869, so enter these coordinates into the corresponding x,y,z boxes. Click the **View** button in order to visualise those atoms included in the binding site definition.

The approximate radius of the binding site must also be specified. By default, the binding site radius is set to 10.0 Å; ensure that this is the case. This radius should be large enough to contain any possible binding mode of the N-phosphonacetyl-L-aspartate ligand.

A cavity detection algorithm, LIGSITE, is used to restrict the region of interest to concave, solvent-accessible surfaces. Ensure that cavity detection is enabled by activating the **Detect Cavity - restrict atom selection to solvent-accessible surface** tickbox.

Click **Next** to proceed to the **Configuration template** dialogue.

22.1.8 Specifying a Configuration File Template

At this point you are given the option to load a configuration file template. Configuration templates can be used to load recommended settings for a number of different types of docking protocols (see Using Configuration File Templates).

In this example we will specify all docking settings manually. Click **Next** to proceed to the **Select ligands** step.

22.1.9 Specifying the Ligand File

As with the protein file, all hydrogen atoms must be present in the ligand input file (see Ligand Hydrogen Atoms, Ionisation States and Tautomeric States). We have already added H atoms to our ligand, extracted it from the protein binding site and saved it.

From within the **Select ligands** window it is possible to:

- Add single ligands,

- Select a complete directory of ligand files,

- Specify a single file containing several ligands (i.e. a multi-mol2 or sdf file).

Specify the ligand you saved earlier by hitting the **Add** button at the bottom of the GOLD Wizard. Navigate to the folder in which you copied the tutorial1 files, select `ligand.mol2` then click **Open**. The `ligand.mol2` will be listed under **Ligand File**.

The number of dockings to be performed on each ligand is specified under **GA runs.** By default, this value is 10. The value can be edited by clicking on this number and re-entering another value; however 10 GA runs are sufficient for this docking.

Click **Next** to proceed to the **Choose a fitness function** window.

22.1.10 Selecting a Fitness Function

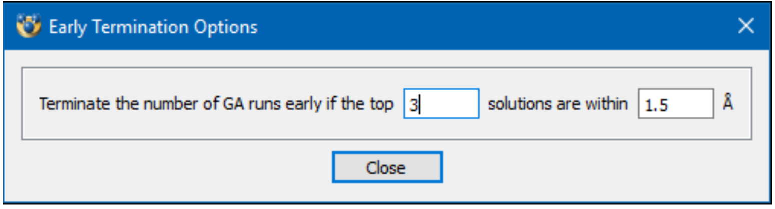
During a docking run, the solutions found by GOLD are scored according to a fitness function (see Fitness Functions).

GOLD offers a choice of fitness functions, Piecewise Linear Potential (ChemPLP) (see Piecewise Linear Potential (ChemPLP), GoldScore (see GoldScore), ChemScore (see ChemScore), Astex Statistical Potential (ASP) (see Astex Statistical Potential (ASP), and **User Defined Score**.

Ensure that the default **CHEMPLP** scoring function is selected.

A number of additional options are available by clicking on the **More** button.

Allow early termination: by default, the **Allow early termination** check box should be switched on. Click on the **Early Termination Options** button to inspect the settings.



Early Termination Options

Terminate the number of GA runs early if the top 3 solutions are within 1.5 Å

This will instruct GOLD to terminate the docking if, at any point, the best three solutions found are all within 1.5 Å rmsd of each other. In this case, it is probable that the answer is correct and further docking runs will not be required. Keep the settings as they are and hit **Close**.

For the purposes of this tutorial all other settings should be left at their default values.

Hit the **Next** button to proceed to the **Genetic Algorithm search options** window.

22.1.11 Selecting Docking Speed

GOLD optimises the fitness score using a genetic algorithm (GA) (see Genetic Algorithm Overview). A number of parameters control the precise operation of the genetic algorithm. The settings are encapsulated into three speeds:

- Slow (most accurate).
- Medium.
- Fast (least accurate).

Further options for each speed setting are available by clicking on the **More** button:

- **Automatic**: enable this to make GOLD automatically calculate an optimal number of operations for a given ligand, thereby making the most efficient use of search time, e.g. small ligands containing only one or two rotatable bonds will generally require fewer genetic operations than larger, highly flexible ligands (see Using Automatic (Ligand-Dependent) Genetic Algorithm Parameter Settings).
- **Preset**: choose from four preset options with varying numbers of operations. The larger the number of operations, the slower and thus more accurate the docking (see Using Preset Genetic Algorithm Parameter Settings).
- **User defined**: this option allows you to tailor your GA settings. Care should be taken when altering these parameter settings (see Using User-Defined Genetic Algorithm Parameter Settings) and you are recommended to use one of the presets offered.

Ensure the radio button is set to **Slow (most accurate)**. Then, click on the **More** button and enable automatic GA settings by clicking on the **Automatic** radio button. Ensure the **Search efficiency** is set to 100%.

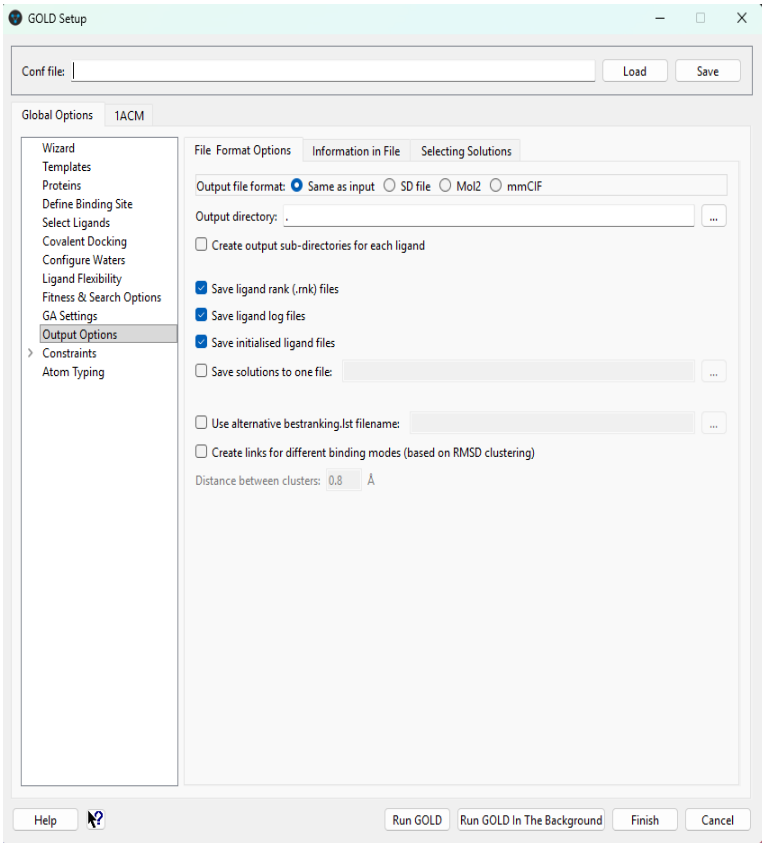
The criteria used by GOLD to determine the optimal GA parameter settings for a given ligand include: the ligand flexibility (i.e. number of rotatable bonds in the ligand, number of flexible ring corners, flippable nitrogens, etc.), the volume of the protein binding site, and the number of water molecules considered during docking. Details of the exact settings used will be given in the ligand log file `gold_ligand_m1.log` (see Ligand Log File).

Hit **Next** to proceed to the **Finish** window.

22.1.12 Specifying a Directory for GOLD Output

We are now finished our docking setup and presented with a **Run GOLD** button with which we can start the docking. If we were to click on **Run GOLD** now, the output would be written to the directory the `1ACM.pdb` file is stored in. It is generally preferable to write output to a separate directory. This option is available as part of the advanced options, so rather that clicking on the **Run GOLD** button, click on the **Advanced** button at the bottom right of the interface. This takes us to the standard GOLD interface.

Select **Output Options** under **Global Options**. This page is separated into three tabbed views: **File Format Options**, **Information in File** and **Selecting Solutions**, all of which allow control of which files are output and what information is written to the files.



Conf file:

Output file format: ● Same as input ○ SD file ○ Mol2 ○ mmCIF

Output directory: .

□ Create output sub-directories for each ligand

☑ Save ligand rank (.rnk) files

☑ Save ligand log files

☑ Save initialised ligand files

□ Save solutions to one file:

□ Use alternative bestranking.lst filename:

□ Create links for different binding modes (based on RMSD clustering)

Distance between clusters: 0.8 Å

In the **File Format Options** tab, ensure that the **Same as input** radio button is activated, adjacent to **Output file format**. This means the docking solutions will be written out in the same format as was used for input (we saved our ligand out in mol2 format; thus this is the format our ligands will be written out in).

Click on the **...** button next to **Output directory** and specify a directory to which you have write permission; this is where the GOLD output files will be written.

Ensure that the **Save ligand rank (.rnk) files**, **Save ligand log files** and **Save initialised ligand files** check boxes are switched on; this will instruct GOLD to retain output files listing fitness-function rankings and ligand log files. The content of these files is discussed later (see Analysis of Output).

Click on the **Information in File** tab.

It is possible to write additional information to docked solution files. This information is written to sd file tags; for mol2 files, these tags are written to comment blocks. This information is particularly important for post-processing docking results. For the purposes of this tutorial, the **Information in File** settings can be left at their default settings.

Now click on the **Selecting Solutions** tab.

GOLD can produce a large amount of output. However, it is possible to cut this down by applying output filter options. These options can be used to:

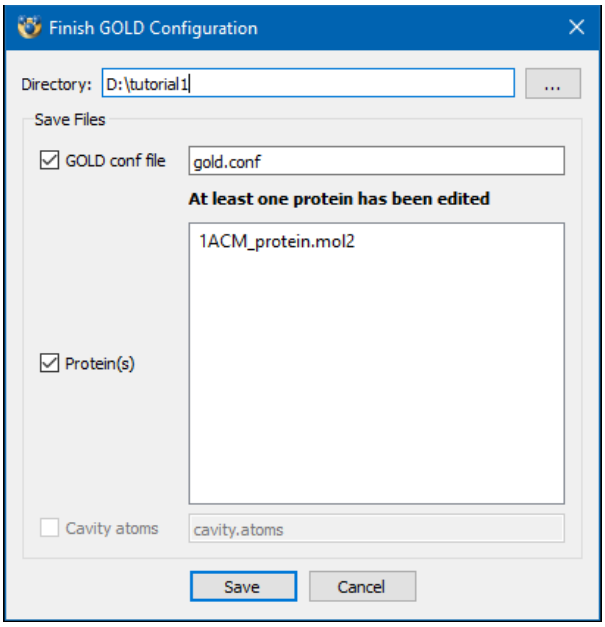
- Specify that all docking solutions are saved
- Retain only the n best docking solutions for each ligand
- Save the top-ranked solution for the best m ligands only
- Filter out all solutions with fitness scores lower than a specified value

By default, the **Keep all solutions** option from the **Selecting Solutions** tab in the **Output Options** window will be selected.

22.1.13 Starting the Docking Run

We are now finished setting up our docking, so click on the **Run GOLD** button at the bottom of the GOLD interface.

You will be presented with a **Finish GOLD Configuration** window containing three **Save Files** options:



Directory: D:\tutorial1

Save Files

☑ GOLD conf file: gold.conf

At least one protein has been edited

☑ Protein(s): 1ACM_protein.mol2

□ Cavity atoms: cavity.atoms

- **GOLD conf file**: if the gold.conf has changed in any way, or if there is currently no gold.conf for the docking (as is the case with this tutorial), you will be provided with the option of saving out a gold.conf file and/or modifying its name.

- **Protein(s)**: in this case we have started from a raw PDB file that was not correctly set up for use with GOLD. The modifications that we have made (note we have been prompted that At least one protein has been edited) mean that the protein(s) can now be used with GOLD, but only if the modifications we have made are saved out as a mol2 file.

- **Cavity atoms**: this option will be greyed out.

Ensure the **GOLD conf file** and **Protein(s)** tickboxes are activated and that the filenames are as you want them, then hit **Save** to start the docking.

22.1.14 Things to Consider While the Docking is Running

Protein and ligand initialisation:

- Both the protein and ligand files are initialised before the docking commences. At this step, GOLD deduces atom types from the information about element types and bond orders in the input structure files. It is therefore crucial that both the protein and ligand input files are prepared according to the guidelines provided (see Atom and Bond Type Overview).

- If automatic atom typing is switched on, GOLD will re-type any atoms or bonds it considers to be incorrect and will issue a warning concerning this in the `gold.err` file. However, if for any reason GOLD is unable to deduce an atom or bond type, then the atom or bond in question will be replaced with a dummy atom type Du or an unknown bond type Un respectively. By default, automatic atom typing is carried out on the ligand file but not the protein file; however, this can be enabled under the **Atom Typing Advanced** option.

Protein flexibility:

- By default, the torsion angles of Ser, Thr and Tyr hydroxyl groups will be allowed to rotate during docking in order to optimise their hydrogen-bonding to the ligand. Lysine NH_3^+ groups are similarly optimised.
- GOLD can allow side chains to rotate within user-defined bounds during docking (see Side Chain Flexibility).
- GOLD can dock into multiple conformations of the same protein (see Ensemble Docking).
- Note that the final positions of any movable protein atoms that are generated during docking (these will usually be different for each docked ligand pose) can be saved to the docked solution file (see Controlling the Information Written to Ligand Solution Files).

Ligand flexibility:

- Only the torsions around the ligand's flexible bonds will be optimised during docking. Bond distances and valence angles must be optimised before using GOLD.
- Torsion angle distributions, extracted from the Cambridge Structural Database (CSD), can be used to restrict the ligand conformational space sampled by the genetic algorithm. Using torsion angle distributions in this way may improve the chances of GOLD finding the correct answer by biasing the search towards ligand torsion-angle values that are commonly observed in crystal structures. It may also improve convergence and so make GOLD usable with faster settings (see Enabling Use of Torsion Angle Distributions).
- The use of torsion angle distributions is enabled by default. The torsion distribution files `tor_lib_2020.tordist` (the default since 2023.3) and `gold.tordist` are provided in `<Installation folder>/ccdc-software/gold/GOLD/gold/` and can be manually edited to include specific, user-defined distributions. The file to be used is specified in the **Ligand Flexibility** window, under **Global Options** in **Advanced** options.

22.1.15 During the Docking

As the job progresses output will be displayed in the **Run GOLD** window.

This is a tabbed view that allows inspection of several files: list of ligand logs, `gold.log`, `gold_protein.log`, `gold.err`, Messages and ligand log.

Any error or warning messages produced will be displayed under the `gold.err` tab. This may contain a number of warning messages relating to the GOLD atom type assigner. These messages can be safely ignored.

Once the job is complete, the docking results can be loaded into Hermes by clicking on the **View Solutions** button at the bottom of the **Run GOLD** window. Keep the **Run GOLD** window open.

22.1.16 Analysis of Output

In addition to the files that can be inspected in the **Run GOLD** window, the specified output directory (see Specifying Ligand Solution File Formats and Directories) will contain a number of files including:

Files containing the initialised protein and ligand (`gold_protein.mol2` and `gold_ligand_m1.mol2`)

Files containing the docked ligand (`gold_soln_ligand_m1_n.mol2`)

Files containing fitness function rankings (`ligand_m1.rnk` and `bestranking.lst`)

Protein and ligand log files (`gold_protein.log` and `gold_ligand_m1.log`)

A file containing error messages (`gold.err`); this file will be empty if no errors are found.

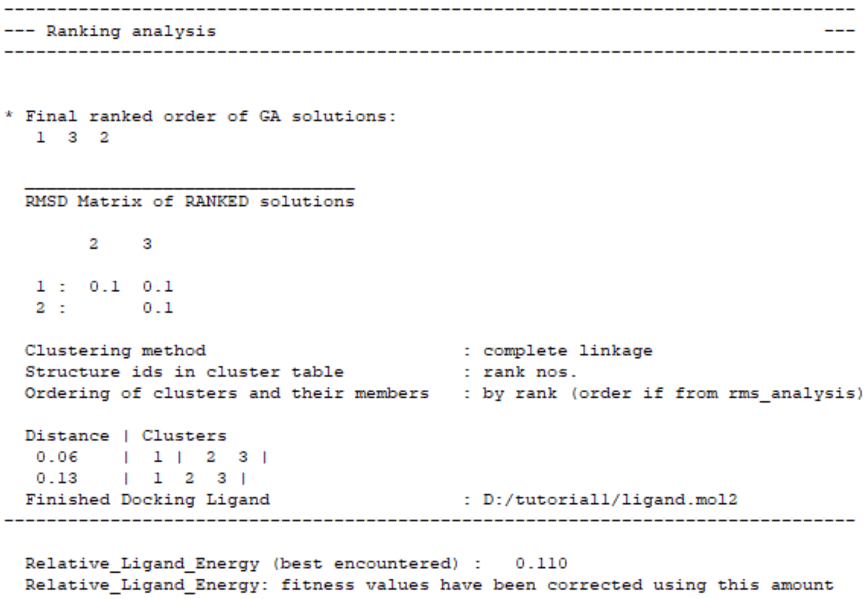
Some of these output files will be dealt with in detail below. Further information on the content of all these output files is available (see Description of Output Files).

22.1.17 The Ligand Log File (gold_ligand_m1.log)

Ten docking runs were set up for this ligand and, for each of these docking runs, the progress of the genetic algorithm is displayed in the `gold_ligand_m1.log` file. This file can be displayed in its own tab in the **Run GOLD** window upon clicking onto the `gold_ligand_m1.log` line listed in the **list of ligand logs** tab. The ligand log file is also saved to the specified output directory (where `m1` is the index or number of the ligand in the input file).

Inspect the `gold_ligand_m1.log` in the **Run GOLD** window. If you have closed this window by accident you can read the file from your output directory into a text editor and view it this way.

Following the completion of all docking runs on the ligand, the results from the different runs are compared. The end of the `gold_ligand_m1.log` file will include a matrix of root mean square deviations (RMSD) between the various docked ligand positions (see Comparison of Docking Solutions). A clustering report is also given which can be used to identify different binding modes (see Identification of Different Binding Modes (Clustering of Ligand Poses)). It is possible that fewer than the specified ten dockings were completed due to the **Allow early termination** option being selected (see Early Termination). In the example output shown below, the solution found for docking attempt number 1 has the best fitness score, while the solution found for docking attempt number 2 has the worst fitness:



```
--------------------------------------------------------------------------------
--- Ranking analysis                                                         ---
--------------------------------------------------------------------------------


* Final ranked order of GA solutions:
   1  3  2

  _______________________________
  RMSD Matrix of RANKED solutions

        2    3

   1 :  0.1  0.1
   2 :       0.1

  Clustering method                      : complete linkage
  Structure ids in cluster table         : rank nos.
  Ordering of clusters and their members : by rank (order if from rms_analysis)

  Distance | Clusters
   0.06    |  1 |  2  3 |
   0.13    |  1  2  3 |
  Finished Docking Ligand                : D:/tutorial1/ligand.mol2
--------------------------------------------------------------------------------

  Relative_Ligand_Energy (best encountered) :   0.110
  Relative_Ligand_Energy: fitness values have been corrected using this amount
```

We have finished with the **Run GOLD** window now so close the window by clicking on the **Close** button.

22.1.18 Fitness Function Rankings Files (ligand_m1.rnk and bestranking.lst)

The `ligand_m1.rnk` file is stored in the specified output directory: open and inspect the file in a text editor. This file contains a summary of the fitness scores for all the docking attempts on the N-phosphonacetyl-L-aspartate ligand.

The docking attempts are listed according to fitness score, so the best solution is placed first.

The file gives total fitness scores and a breakdown of the fitness into its constituent energy terms.

A file called `bestranking.lst` is written and gives a continuous summary of the best solution that has been obtained for each completed ligand. The file gives total fitness scores and a breakdown of the fitness into its constituent energy terms.

22.1.19 Files Containing the Docked Ligand (gold_soln_ligand_m#_n.mol2)

The N-phosphonacetyl-L-aspartate ligand will have been docked a number of times so a set of files will have been written to the output directory, each containing the results of a separate docking attempt.

The result of each docking attempt is written out as `gold_soln_ligand_m1_n.mol2`, where `n` is the number of the docking solution 1,2,3 ... and `m1` is an index to the ligand (in this example, only one ligand was docked).

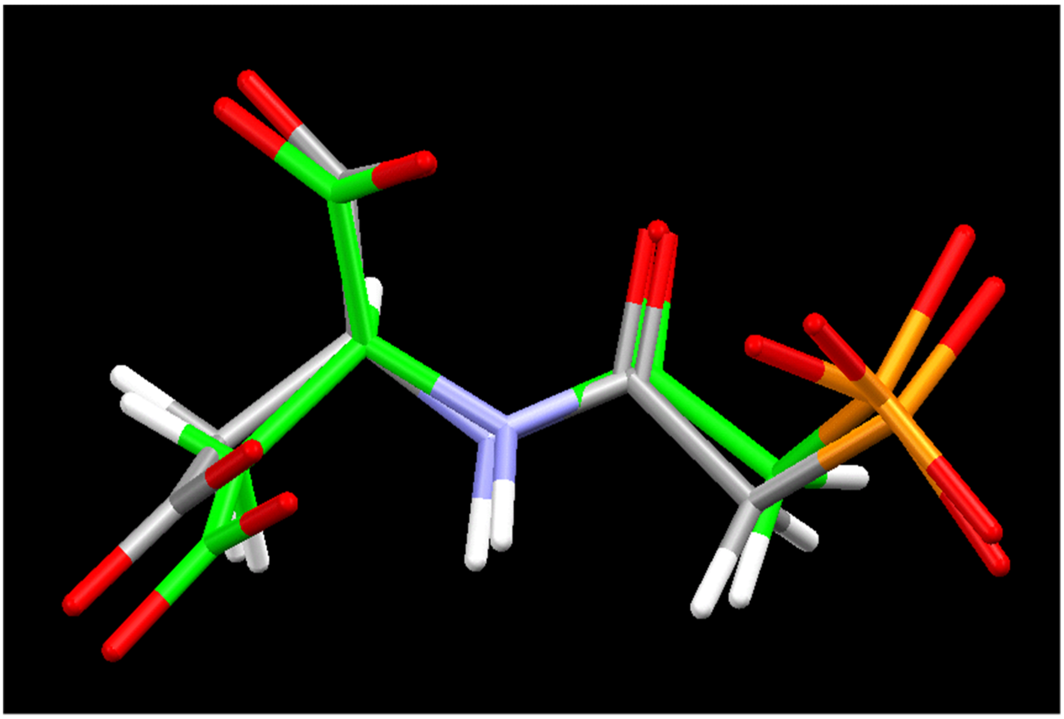
Note that the file `gold_soln_ligand_m1_1.mol2` is not necessarily the best GOLD prediction, it is just the solution found in the first docking attempt. However, as GOLD proceeds, symbolic links are created: `ranked_ligand_m1_1.mol2` will point to the current top-ranked solution, `ranked_ligand_m1_2.mol2` will point to the second-best solution, and so on.

Return to the Hermes 3D view and inspect the top-ranked solution predicted by GOLD. Note that the original protein we edited is still loaded: to make the display less complicated you may wish to disable one of the proteins by deactivating the tickbox adjacent to 1ACM or 1ACM_2 under the **Display** tab in the **Molecule Explorer**. If you do this, return to the **Docking Solutions** tab once you have finished.

The docking solutions are given in their docked order with their corresponding fitness score listed under the column headed **PLP.Fitness**. If required, the solutions can be ordered by clicking on **PLP.Fitness** to determine which is the highest scoring.

A simple test of the effectiveness of a docking program is to take a protein-ligand complex from the PDB and extract the ligand. The docking program can then be used to predict the binding mode of the ligand and a comparison made with the crystallographically observed position. The crystallographically observed conformation of the docked N-phosphonacetyl-L-aspartate ligand is stored in the ligand we extracted from the protein and that was subsequently re-loaded (**A:1ACM** in the **Molecule Explorer**). Compare this with the solution predicted by GOLD.

In the figure below the crystallographically observed reference structure **A:1ACM** (shown in green) is compared with the top-ranked solution predicted by GOLD (shown coloured by element):



Using this methodology, GOLD has been validated against a large number of protein-ligand complexes taken from the PDB. Further details and the entire validation test set are available for download.

This ends the tutorial.

22.2 Tutorial 2: Handling of Metals in GOLD

22.2.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial2` to a directory to which you have write permissions.

The object of this tutorial is to investigate the binding mode of dorzolamide, an inhibitor of carbonic anhydrase II, PDB entry code 4m2u. In this example, the dorzolamide molecule is known to coordinate to a zinc atom within the binding site of the protein.

This tutorial will illustrate the requirements for setting up and running a docking in which the protein binding site features a metal ion. Additional information will also be provided on the handling and parameterisation of metals in GOLD.

22.2.2 Preparation of Input Files

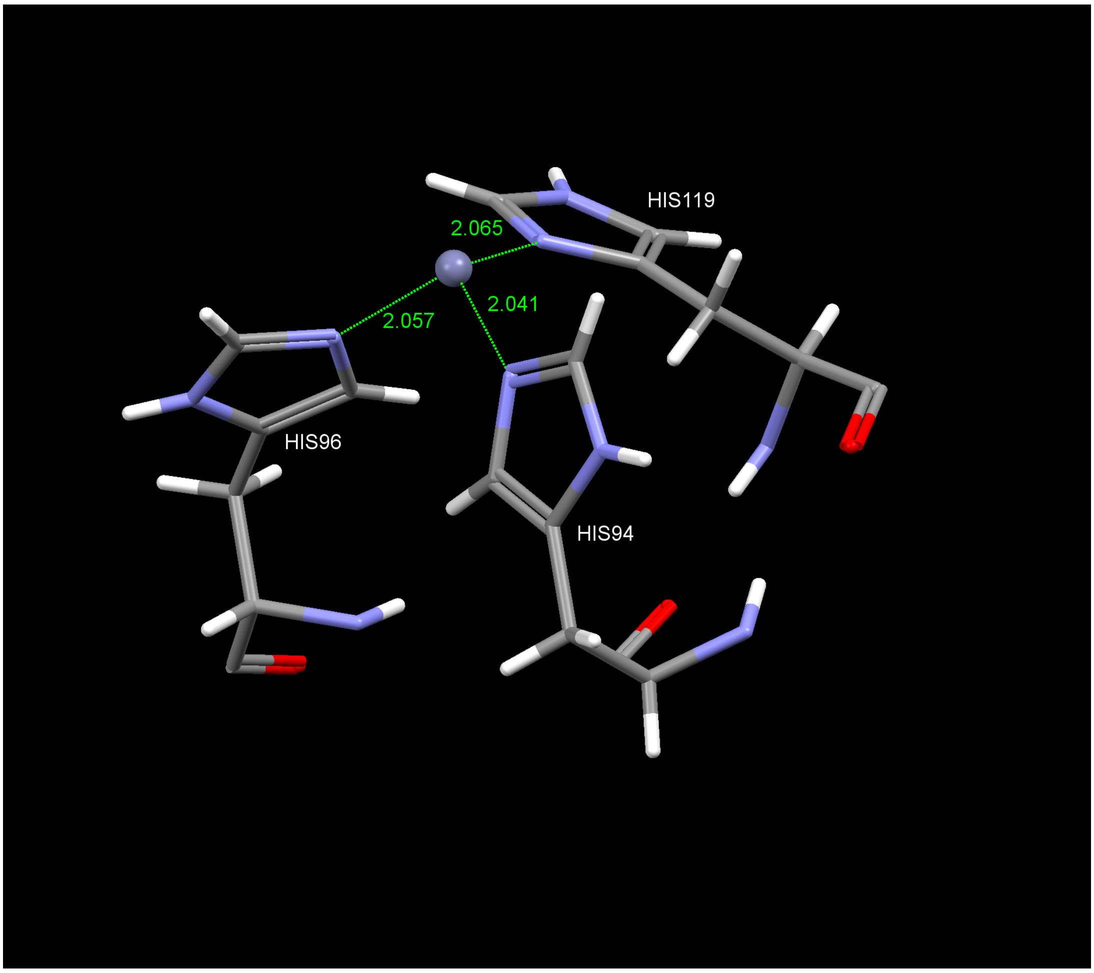
Open Hermes and read in the file `protein.mol2` from the folder to which you copied the tutorial2 files. The original PDB file `4M2U.pdb` has also been provided should you wish to set up the protein for yourself.

The protein has been set up in accordance with the guidelines for the preparation of protein input files (see Setting Up the Protein(s)). All water molecules have been deleted and hydrogen atoms have been placed on the protein to define correct ionisation and tautomeric states (see Protonation and Tautomeric States).

In addition, the following requirements specific to proteins with metal ions are met:

- The metal ion is coordinated to at least two protein atoms or water molecules to allow GOLD to determine the correct coordination geometry.
- The metal ion must not have any bonds to coordinating atoms. Protein-zinc bonds present in **4M2U.pdb** have been deleted.

In `protein.mol2`, the zinc atom is coordinated to three histidine residues via non-protonated azole nitrogen atoms.



The dorzolamide inhibitor molecule has been set up in accordance with the guidelines for the preparation of ligands (see Essential Steps). Additional care has been taken to ensure correct recognition of the deprotonated sulfonamide group and the quarternary nitrogen in dorzolamide. In accordance with GOLD's conventions (see Atom and Bond Type Conventions for Difficult Groups), N13 in the sulfonamide group is deprotonated and its atom type set to N.pl3, and N14 is protonated and set to N.4.

Open and inspect the file `ligand.mol2` from the folder to which you copied the tutorial2 files.

Once you have finished looking at the protein and ligand files, close them by selecting **File**, then **Close All Files**.

22.2.3 The GOLD Configuration File

All of the parameters and settings required to define a particular GOLD job are saved in the GOLD configuration file (`gold.conf`) (see Saving and Reusing Docking Settings). This text file includes the specification of the ligand, the protein binding site, the fitness function, the torsion distributions, and the genetic algorithm. The `protein.mol2` and `ligand.mol2` input files will be loaded into Hermes upon opening the `gold.conf`.

A configuration file has been provided for this tutorial. The `gold.conf` is loaded by clicking on the main menu option **GOLD**, then **Setup and Run a Docking**. From the resultant pop-up window select the **Load Existing** button: you should then browse to the folder to which you copied the tutorial2 files, select the file `gold.conf`, then hit **Open**. This will automatically load the settings and parameter values for this tutorial into the GOLD Setup window.

From the **Global Options** on the left-hand side of the GOLD Setup window, click on **Output Options**. Specify or browse to a directory (using the **...** button adjacent to **Output directory**) for which you have write permissions. This is where the GOLD output files will be written.

22.2.4 The Handling and Parameterisation of Metals in GOLD

GOLD is able to predict binding to twelve metal ions: Mg, Zn, Fe, Mn, Ca, Co, Gd, Cu, Hg, Cd, Ni and V.

No special instructions are needed to dock to metal ions; they will be handled automatically when present in the protein binding site.

22.2.5 Automatic Determination of Metal Coordination Geometries

GOLD will automatically recognise the following metal coordination geometries:

Template	Geometry	Coordination Number
TETR	Tetrahedral	n=4
TBP	Trigonal bipyramidal	n=5
OCT	Octahedral	n=6
CTP	Capped trigonal prism	n=7
PBP	Pentagonal bipyramidal	n=7
SQAP	Square prism	n=8
ICO	Icosahedral	n=10
DOD	Dodecahedral	n=12

In order to determine the coordination geometry of a particular metal atom, GOLD performs a permuted superimposition of coordination geometry templates onto the coordinating atoms found in the protein. Coordination fitting points are then generated using the template that gives the best fit (based on RMSD).

The geometry templates used for a given metal are defined in the `gold.params` file in the section headed `# Metals`:

H-Bonding type	SYBYL atom type	Atom type (default or elucidated)	Donor (D), Acceptor (A), or Metal (M).	Allowed Coordination geometries	Coordination distance
MGD	Mg	DEF	M	4, 6	2.05
ZND	Zn	DEF	M	4, 5, 6	2.09
MND	Mn	DEF	M	4, 6	2.06
FED	Fe	DEF	M	4, 6	1.98
CAD	Ca	DEF	M	6, 7	2.44
COBD	Co.oh	DEF	M	6	2.09
GDD	Gd	DEF	M	6	2.44
CUD	Cu	DEF	M	4, 6	2.10
HGD	Hg	DEF	M	4, 6	2.40
CDD	Cd	DEF	M	4, 6	2.30
NID	Ni	DEF	M	4, 6	2.15
VD	V	DEF	M	4, 6	2.10

The `gold.params` file is stored in `<Installation_folder>/ccdc-software/gold/GOLD/gold/`; go to this directory and open the parameters file using a text editor.

The parameters used by GOLD for each metal are listed; for an explanation of these parameters, please refer to comments in the `gold.params` file. Additional metal parameterisation can also be found within the `H_BOND TABLE`.

For our Zn atom, GOLD will therefore attempt to match coordination geometries 4, 5 and 6 (tetrahedral, trigonal bipyramidal, and octahedral templates) onto the coordinating atoms found in the protein. The template that gives the best match will then be used to generate coordination fitting points.

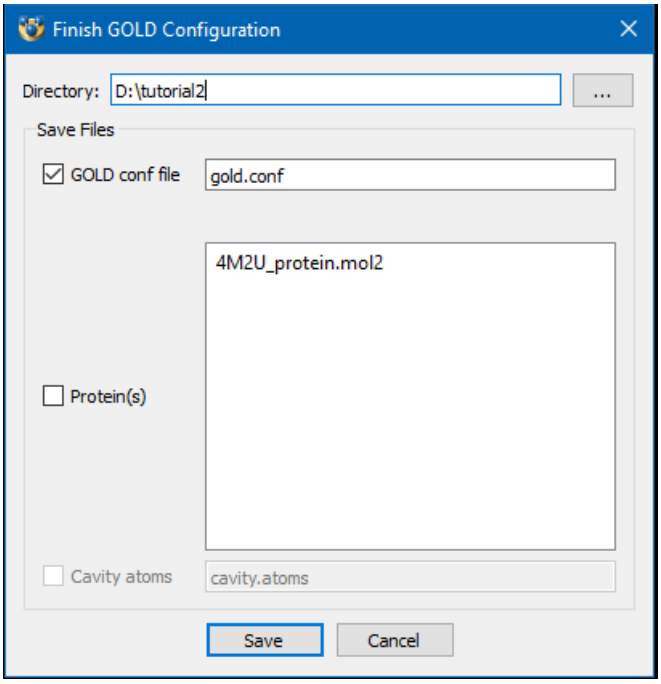
Once you have finished viewing the file, close it.

22.2.6 Manually Specifying Metal Coordination Geometries

It is possible to manually specify coordination geometries for particular metal atoms (see Defining Custom Metal Coordination Geometries). This can be useful in allowing non-standard metal coordination geometries, or to limit the number of possible geometries that GOLD checks (i.e. to overrule the default geometries for the corresponding metal type defined in the `gold.params` file).

22.2.7 Running GOLD and Analysis of Output

Click on the **Run GOLD** button at the bottom of the GOLD front end. If you have changed the output directory, you will be prompted that the GOLD configuration has been updated and needs to be saved; in this case, click **Save** to proceed to the **Finish GOLD Configuration** window.



Directory: D:\tutorial2

Save Files

☑ GOLD conf file: gold.conf

☐ Protein(s): 4M2U_protein.mol2

☐ Cavity atoms: cavity.atoms

From within this window we can:

- Specify the directory the `gold.conf` is to be saved to: we will leave this as the default working directory.
- Save a Protein mol2 file (and specify its name): this is only necessary if the protein file has been modified. We have not modified the `protein.mol2` file so ensure the **Protein(s)** tickbox is deactivated.
- Save a GOLD conf file (and specify its name). If you have changed some of the configuration options such as the results output directory, ensure that the **GOLD conf file** tick box is activated.

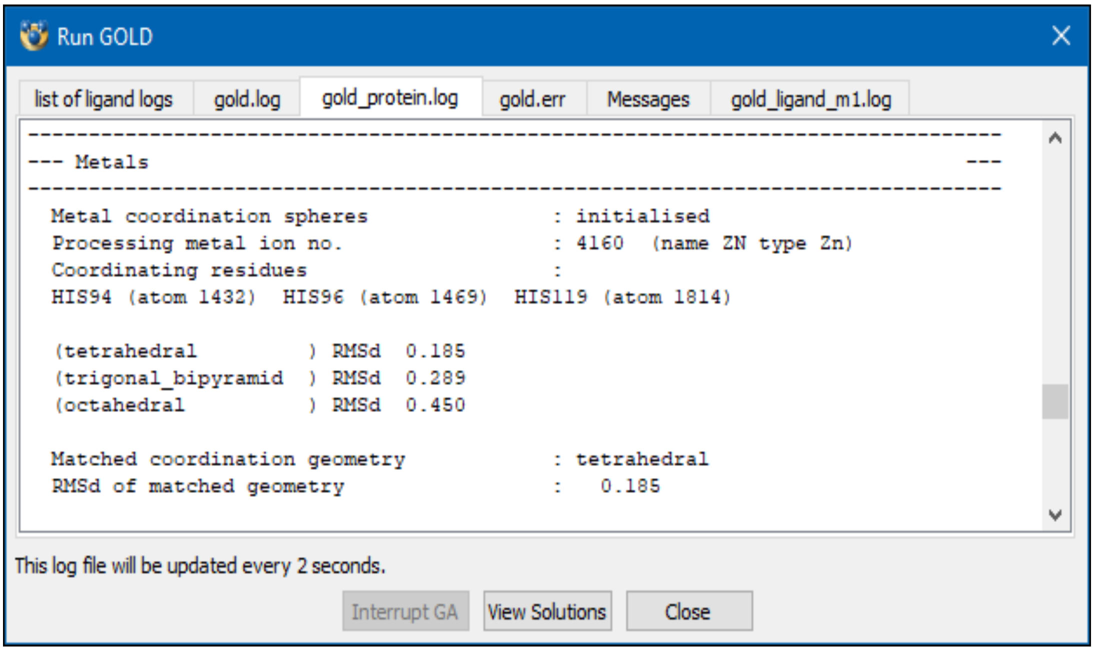
Hit **Save** then **OK** to overwrite the existing `gold.conf` file. GOLD will then start running interactively.

The GOLD output window is a tabbed view that allows you to inspect various files that are written while the docking proceeds. Once the job is complete, the message Finished Docking Ligand: ligand.mol2 will appear in the **gold_ligand_m1.log** tab.

22.2.8 Protein Log File

Inspect the `gold_protein.log` file by hitting the `gold_protein.log` tab in the **Run GOLD** window. If you have already closed the Run GOLD window this file can be found in the output directory specified (see The GOLD Configuration File) and can be read using a text editor.

The `gold_protein.log` file contains details of the parameterisation of the protein and the determination of the binding site. Information relating to the metal and the determination of the coordination geometry will also be given:



```
--------------------------------------------------------------------------------
--- Metals                                                                   ---
--------------------------------------------------------------------------------
  Metal coordination spheres                  : initialised
  Processing metal ion no.                    : 4160  (name ZN type Zn)
  Coordinating residues                       :
  HIS94 (atom 1432)  HIS96 (atom 1469)  HIS119 (atom 1814)

  (tetrahedral          ) RMSd  0.185
  (trigonal_bipyramid   ) RMSd  0.289
  (octahedral           ) RMSd  0.450

  Matched coordination geometry               : tetrahedral
  RMSd of matched geometry                    :   0.185
```

This log file will be updated every 2 seconds.

After evaluating RMSD values for fitting metal coordination states from its library, GOLD has assigned a tetrahedral geometry.

Further information about the contents of the `gold_protein.log` file is given elsewhere, (see Protein Log File).

22.2.9 Files Containing the Protein and Docked Ligands

Load the docking results into Hermes by clicking on the **View Solutions** button in the **Run GOLD** window. We no longer need the **Run GOLD** window so close it by hitting the **Close** button.

The docking results will be loaded into the Hermes visualiser.

The protein file now contains dummy atoms connected to the metal which represent idealised metal coordination positions. These can be visualised by activating the **Show unknown atoms** tick box at the top of the **Hermes** window.

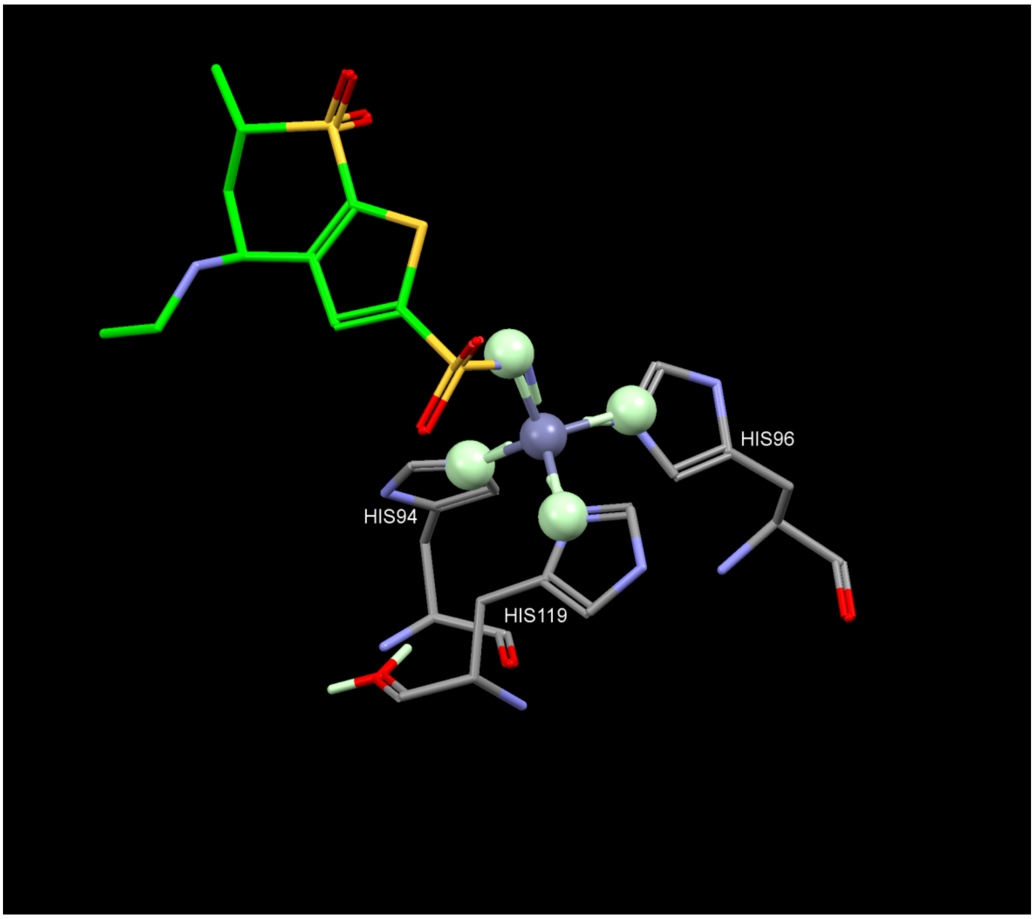
At locations where GOLD is missing a coordination site (i.e. coordination points not bound to the protein) virtual coordination points are added. These coordination points are then used as fitting points that can bind to acceptors.

Inspect how well the docked inhibitor fits within the protein binding site.

The docking solutions are listed in the **Docking Solutions** tab of the **Molecule Explorer**, their overall Goldscore.Fitness score is listed alongside. Click on the top solution and, holding the left mouse button depressed, drag with your mouse to the final solution in the list, thus selecting all solutions.

The ligands coordinate to the metal either via the sulfonamide moiety or via the sulfonyl group (it may be that you only see one coordination pattern, due to the stochastic nature of GOLD your results may differ from those presented here).

The zinc (shown in blue) is coordinated to the protein via three histidine residues. In the example shown below, the remaining zinc coordination site is used to bind the inhibitor (shown with C atoms coloured in green) via interaction with the electron lone pair of the deprotonated sulfonamide N atom.



Metal coordination in GOLD is modelled as 'pseudo-hydrogen bonding'. Metal-ligand interactions will typically involve the metal binding to, for example, carboxylate ions, deprotonated histidines (i.e. negatively charged), and phenolates. In our example, zinc is coordinated to a sulfonamide anion. Metals can be considered to bind to H-bond acceptors and the metal will compete with H-bond donors for interaction.

This ends the tutorial.

22.3 Tutorial 3: Use of Hydrogen Bonding Constraints

22.3.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial3` to a directory to which you have write permissions.

The design of new and more potent antiretroviral agents for the human immunodeficiency virus (HIV) continues to be the focus of much attention. The crystal structures of HIV-1 protease in complex with a number of cyclic urea inhibitors have been determined in order to identify the key interactions responsible for the high potency of this class of inhibitor (see P. K. Jadhav et al., J. Med. Chem., **40**, 181-191, 1997, DOI: 10.1021/jm960586t). The C_2 symmetric cyclic urea scaffold is well suited to interact with the viral protease. It has been observed that these inhibitors are anchored in the active site of the protease by six key hydrogen bonds.

The object of this tutorial is to investigate the binding mode of a cyclic urea inhibitor with HIV-1 protease, PDB entry code 1qbt. The use of hydrogen bonding constraints in order to reproduce these key interactions will also be illustrated.

22.3.2 Input Files

Open Hermes and read in and inspect the file `protein.mol2` from the folder to which you copied the tutorial3 files. The original PDB file 1QBT.pdb has also been provided, should you wish to set up the protein for yourself.

HIV-1 protease, `protein.mol2`, has already been set up in accordance with the guidelines for the preparation of protein input files (see Setting Up the Protein(s)).

An important feature of cyclic urea inhibitors is their ability, upon binding, to displace a structural water molecule present within the active site of the protein. In this example, all water molecules have been deleted from `protein.mol2`. However, in other complexes you may not know whether water molecules should form mediating hydrogen bonds or be displaced by the ligand on binding. GOLD allows waters to switch on and off (i.e. to be bound or displaced), to rotate and to translate within a radius of 2 Å (to optimise hydrogen bonding) during docking (see Water Molecules).

The cyclic urea inhibitor has already been prepared in accordance with the requirements for setting up the ligand (see Setting Up Ligands).

Open the file `ligand.mol2` from the folder to which you copied the tutorial3 files within Hermes and inspect the structure. Keep the file open once you have finished.

A configuration file has been provided for this tutorial. Load the `gold.conf` by clicking on the main menu option **GOLD**, then **Setup and Run a Docking**. From the resultant pop-up window select the **Load Existing** button: you should then browse to the directory to which you copied the tutorial3 files, select the file `gold.conf` then hit **Open**. This will automatically load the settings and parameter values for this tutorial into the GOLD Setup window in addition to the specified protein file.

The GOLD interface contains two tabbed views; the default is **Global Options** which allows you to specify particulars of the docking in general; the other displays the protein name, in this case **1QBT**, and allows you to edit the protein and set up parameters specific to the protein such as constraints. Click on the **1QBT** tab.

22.3.3 Hydrogen Bonding Constraints

GOLD features two types of hydrogen bonding constraints:

- **A standard hydrogen bond constraint**, which can be used to force a hydrogen bond between a specific protein atom and a specific ligand atom (see Standard Hydrogen Bond Constraints).
- **A protein hydrogen bond constraint**, which can be used to specify that a particular protein atom should be hydrogen-bonded to the ligand, but without specifying to which ligand atom (see Protein Hydrogen Bond Constraints).

22.3.4 Standard Hydrogen Bond Constraints

A standard hydrogen bond constraint allows a particular ligand atom to be constrained to form a hydrogen bond to a particular protein atom.

Click on the triangle next to **Contraints** in the **1QBT** tab: this will expand the tree to list all constraint options available. Select **HBond** from the list of constraint types.

When specifying a hydrogen bond constraint, the ligand and protein atoms involved in the constraint need to be selected in the 3D view. Clicking on the atoms simultaneously selects them (selected atoms will be surrounded by a cyan sphere) and enters the relevant atom IDs into the constraints dialogue.

One of the atoms must be an H-bond donor and the other should be an acceptor. The protein atom must also be available for ligand binding (i.e. solvent accessible).

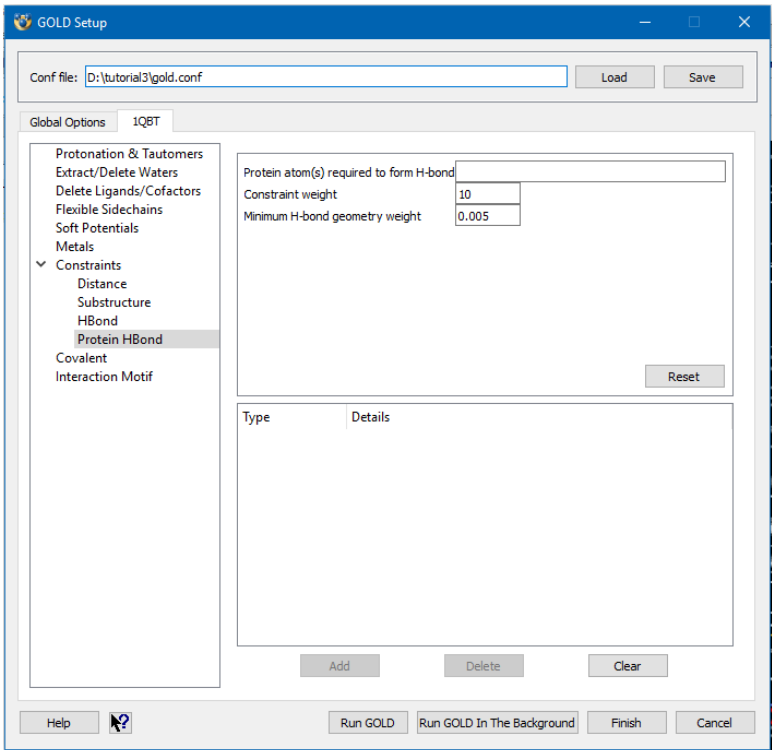
Once defined, an H-bond constraint is incorporated into the least-squares fitting routine used by GOLD to dock the ligand. The constraint has a weight of 5 relative to a normal hydrogen bond. Thus, the docking will be biased towards solutions which include the specified hydrogen bond.

The hydrogen bond constraint weighting can be altered within the `Fitness Function` section of the GOLD parameters file by changing the value of the parameter `CONSTRAINT_WT`**.**

22.3.5 Protein Hydrogen Bond Constraints

A protein hydrogen bond constraint can be used to specify that a particular protein atom should be hydrogen-bonded to the ligand, but without specifying to which ligand atom (see Setting up Protein H Bond Constraints).

Click on the triangle next to **Contraints** in the **1QBT** tab and select **Protein HBond** from the list of constraint types:



Conf file: D:\tutorial3\gold.conf

Protein atom(s) required to form H-bond

Constraint weight 10

Minimum H-bond geometry weight 0.005

When specifying a protein hydrogen bond constraint, the protein atom must be selected in the 3D view.

GOLD will then be biased towards finding solutions in which the specified protein atom forms hydrogen bonds. However, as with standard hydrogen bond constraints, such a solution is not guaranteed.

During the GOLD run the fitness score of a given docking will be penalised for every protein H-bond constraint that is not satisfied.

The **Constraint weight** is the strength of bias applied to the formation of a specified hydrogen bond in the least squares mapping algorithm within GOLD. The **Constraint weight** is also the value of the penalty applied to the fitness score for each constrained H-bond that is not formed.

The **Minimum H-bond geometry weight** is a user-defined score that determines how good a hydrogen bonding interaction has to be in order for it to be considered a hydrogen bond by GOLD. The **Minimum H-bond geometry weight** takes a range of values from 0 to 1; by default this value is set at 0.005.

22.3.6 Specifying Multiple Constraints

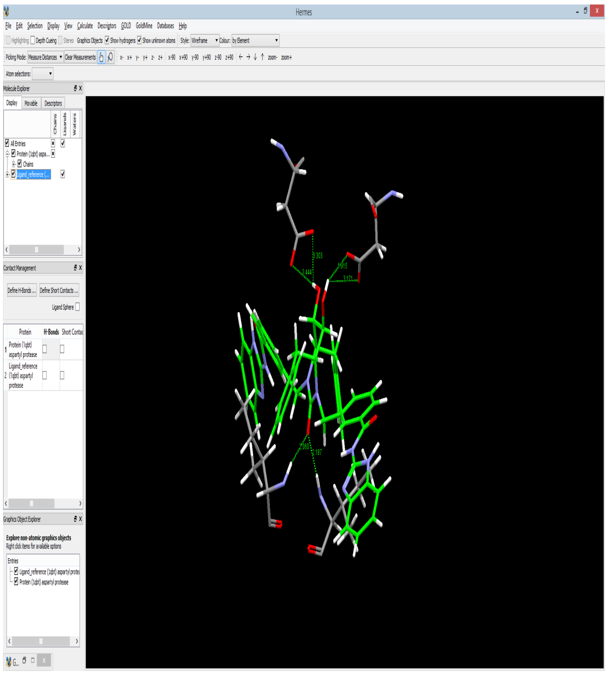
It is possible to specify several different protein H-bond constraints, with different weights for each constraint. Simply select each protein atom required to form an H-bond with the ligand as well as the required weight, then click on the **Add** button to add the constraint definition to the constraints dialogue at the bottom of the window. Repeat the procedure to set up further constraints; each constraint will be displayed on a separate line in the constraints dialogue.

For a given protein H-bond constraint more than one protein atom can be selected and added to the **Protein atom(s) required to form H-bond** input box. This will instruct GOLD to use an 'either-or' type of constraint during docking. For example, specifying two protein atoms m and n, separated by a space, will result in the constraint being satisfied if an H bond is formed to either m or n during docking. This is of particular use when defining constraints involving, for example, carboxylates where it is not important which oxygen atom forms an H-bond, provided one does.

22.3.7 Defining the Protein H-Bond Constraints

The crystal structures of HIV-1 protease in complex with a number of cyclic urea inhibitors have been determined. It has been observed that the central urea moiety is anchored in the active site of the protease by six key hydrogen bonds:

- Two hydrogen bonds between the urea oxygen atom and the protein backbone peptide groups of Ile50 and Ile50' (shown below).
- Four hydrogen bonds between the cyclic urea diol and the carboxylates of the catalytic aspartate of the protein residues (ASP25') (shown below).



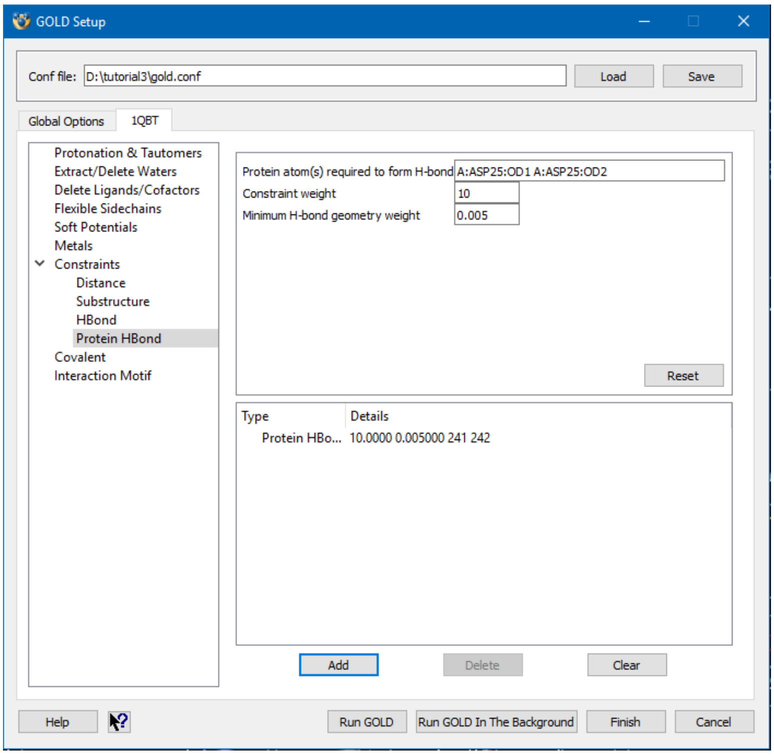
	Protein	H-Bonds	Short Contac
1	Protein (1qbt) aspartyl protease	□	□
2	Ligand_reference (1qbt) aspartyl protease	□	□

Protein H-bond constraints can be used in order to attempt to reproduce these key interactions during docking.

Specify that either oxygen atom of the carboxylate group of Asp25 in chain A should form a hydrogen bond to the ligand by clicking on one O atom, then the other. Note: If you have problems locating Asp25 in chain A you can expand the protein tree in the **Molecule Explorer,** i.e. by clicking on the “**>**” adjacent to the 1QBT entry. Expanding the **Chains** tree then **A** will give a breakdown of all residues in chain A. You can then right-click on **Asp25** to modify the display settings and make it stand out.

Click on both carboxylate atoms in the 3D view. The selected atoms will be highlighted with cyan spheres.

In the constraints window, the default settings for **Constraint weight** and **Minimum H-bond geometry weight** are given (10.0 and 0.005 respectively). Select **Add** to accept these values. The specified constraint will appear in the list at the bottom:



Conf file: D:\tutorial3\gold.conf

Protein atom(s) required to form H-bond: A:ASP25:OD1 A:ASP25:OD2

Constraint weight: 10

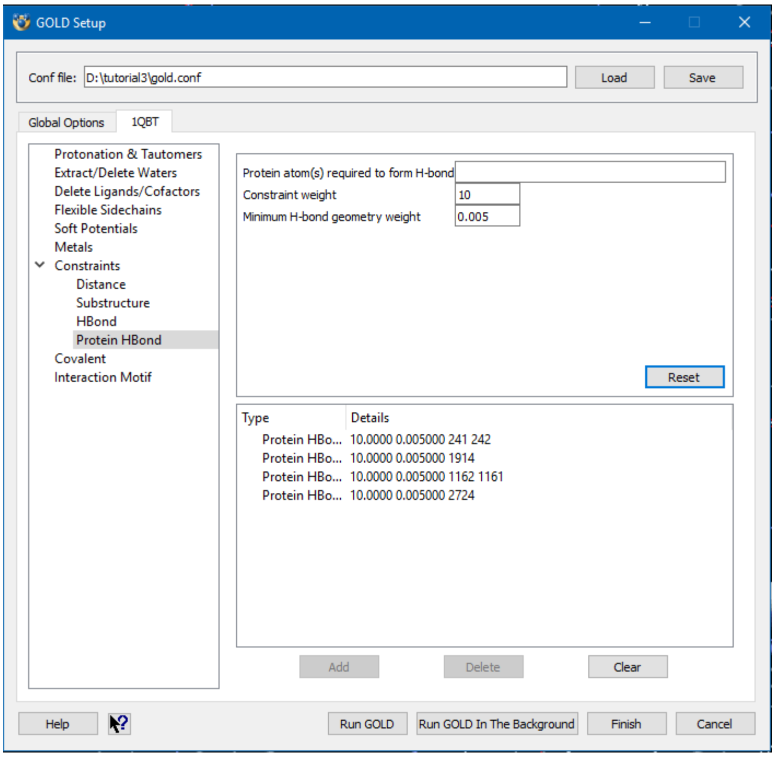
Minimum H-bond geometry weight: 0.005

Type	Details
Protein HBo...	10.0000 0.005000 241 242

Specify protein H-bond constraints for the three remaining key hydrogen bonding interactions as outlined in the table below (note that you may have to hit the **Reset** button to clear the **Protein atom(s) required to form H-bond** textbox). Note: It is necessary to specify the hydrogen atom to define the donor partner in the H-bond constraint.

Protein H bonding group	**Atom label(s)**	**Atom number(s)**	**Constraint weight**	**Minimum H bond geometry weight**
Ile50 (chain A)	A:ILE50:H	1914	10.0	0.005
Ile50 (chain B)	B:ILE50:H	2724	10.0	0.005
Asp25 (chain B)	B:ASP25:OD2 B:ASP25:OD1	1161 or 1162	10.0	0.005

Once all of these protein H-bond constraints have been set up, the list at the bottom of the constraints window should contain four individual constraints:



Conf file: D:\tutorial3\gold.conf

Protein atom(s) required to form H-bond:

Constraint weight: 10

Minimum H-bond geometry weight: 0.005

Type	Details
Protein HBo...	10.0000 0.005000 241 242
Protein HBo...	10.0000 0.005000 1914
Protein HBo...	10.0000 0.005000 1162 1161
Protein HBo...	10.0000 0.005000 2724

22.3.8 Running GOLD

Return to the general docking setup by clicking on the **Global Options** tab.

Click on **Output Options**. Either type an output directory name in the **Output directory** window or browse to a directory using the **...** button adjacent to this window. This is where the GOLD output files will be written.

Click on the **Run GOLD** button at the bottom of the GOLD front end.

All settings can remain as they are, so hit **Save** to start the GOLD run. You will be prompted that a file called gold.conf already exists and asked if you want to overwrite it. Click **OK** to agree to overwrite the existing `gold.conf` file. Alternatively, choose **Cancel** to go back to the **Finish GOLD Configuration** window, enter a new file name for the GOLD conf file and press **Save**. The GOLD job will now start interactively. As the job progresses output will be displayed in the **Run GOLD** window.

The **Run GOLD** output window is a tabbed view that allows you to inspect various files that are written while the docking proceeds. Once the job is complete, the message Finished Docking Ligand: ligand.mol2 will appear in the **gold_ligand_m1.log** tabbed view.

Once the GOLD job is complete, load the results into Hermes using the **View Solutions** button.

22.3.9 Analysis of Output

Inspect the `gold_protein.log` file by hitting the **gold_protein.log** tab in the **Run GOLD** window. If you have already closed the **Run GOLD** window this file can be found in the output directory specified (see The GOLD Configuration File) and can be read using a text editor. This file contains details of the protein initialisation.

Now return to the **list of ligand logs** window and click on **gold_ligand_m1.log** (again if you have closed the **Run GOLD** window, this information can be found in the `gold_ligand_m1.log` file stored in your specified output directory). This file gives a total fitness score and a breakdown of the fitness into its constituent energy terms for each docking performed on the ligand.

A constraint scoring term `DE(con)` is listed for each docking. If a solution predicted by GOLD satisfies all of the protein H-bond constraints, then the contribution from this scoring term will be 0.00. However, for solutions in which not all of the constraints are satisfied, a penalty will be applied to the fitness score for each constrained H-bond that is not formed. The value of this penalty is the **Constraint weight** previously specified.

The details of each specified protein H-bond constraint satisfied in the solution are listed and an overall constraint score is given. A list of all hydrogen bonds formed between ligand and protein is also provided in the ligand log file.

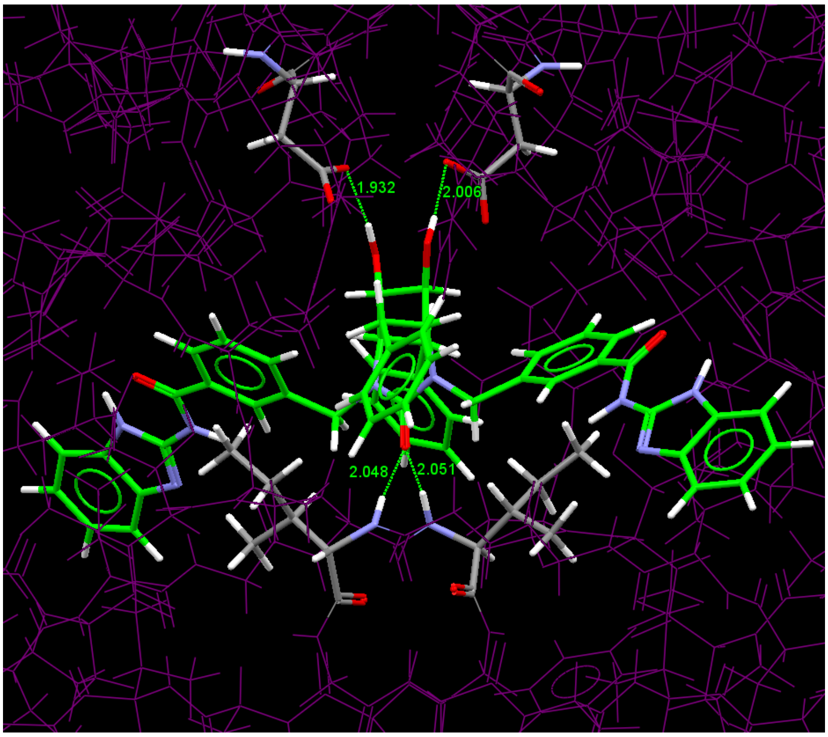
Go to the end of the `gold_ligand_m1.log` file, then scroll up slightly until you see text similar to the following:

```sh
* Final ranked order of GA solutions
3 4 6 2 5 1
```

This text tells you how the docking attempts rank in terms of fitness score. Here, the third docking attempt is the top-ranked, while the first docking attempt is the lowest-ranked of all the solutions.

Go to Hermes 3D view and display the top-ranked solution (note that it may not be docking attempt number 3 for your results). If you are still unsure which is the top-ranked solution, the docking results can be ordered based on their fitness score in the **Molecule Explorer** window, using the **PLP.Fitness** header in the **Docking Solutions** view.

Inspect how well the docked inhibitor fits within the protein binding site as predicted by GOLD:



Interactions between the cyclic urea inhibitor and HIV-1 protease can be divided into two groups: those that anchor the scaffold in the active site and those that fix the substituents in the target sub-sites.

Confirm that the hydrogen bonds specified in the constraints are formed as expected to the cyclic urea scaffold by measuring the relevant contact distances. Identify any additional hydrogen bonding interactions between the benzimidazole substituents and the target sub-sites within the protein.

This ends the tutorial.

22.4 Tutorial 4: Use of Substructure Based Distance Constraints

22.4.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial4` to a directory to which you have write permissions.

The object of this tutorial is to assess the binding of a small number of structurally related ligands with the carbonic anhydrase II, PDB entry code 1cil. In the ETS inhibitor a terminal sulfonamide nitrogen atom is observed to coordinate to a zinc atom within the protein binding site.

This tutorial will illustrate how GOLD can be used to screen a number of compounds in order to identify ligands with potential activity. The use of constraints in order to bias solutions towards the observed binding mode of the inhibitor will also be demonstrated, as well as the use of automatic speed settings.

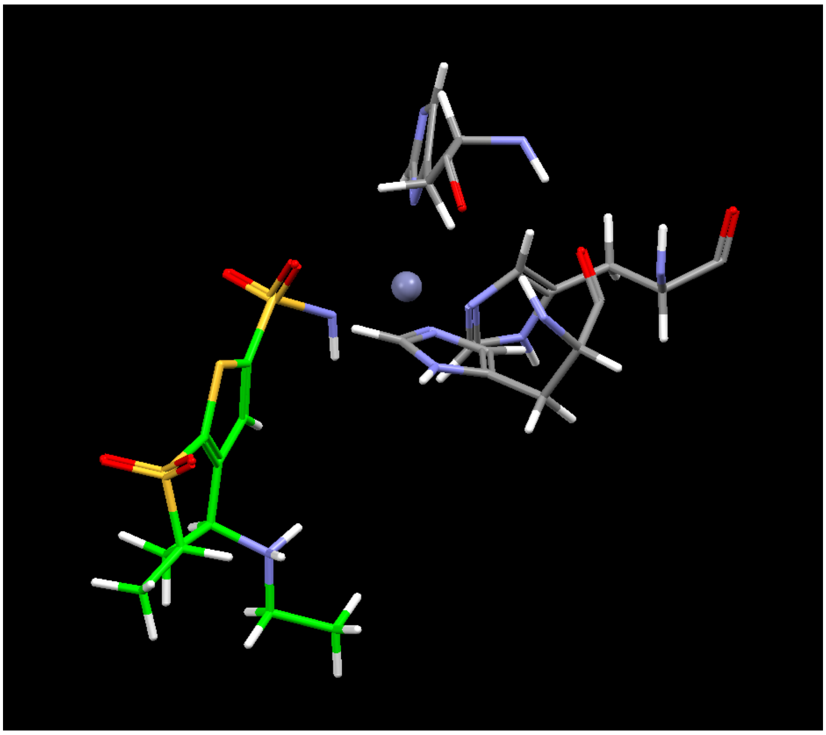
22.4.2 Input Files

Open Hermes and read in the file `protein.mol2` from the folder to which you copied the tutorial4 files. The original protein PDB file `1CIL.pdb` has also been provided should you wish to set up the protein for yourself.

The carbonic anhydrase II 1cil protein, `protein.mol2`, has already been set up in accordance with the guidelines for the preparation of protein input files (see Protonation and Tautomeric States).

Upon inspection of the protein you will see that the zinc atom is coordinated to three histidine groups, such that the one remaining zinc coordination site is available for binding to the ligand.

Read in the file `ligand_reference.mol2` from the folder to which you copied the tutorial4 files. Inspect the crystallographically observed position of the ETS inhibitor (shown in green) within the protein binding site:



The terminal sulfonamide nitrogen atom of the ligand clearly coordinates to the zinc. We can attempt to reproduce this known binding mode within GOLD with the introduction of a distance constraint during docking.

Ten ligands, each structurally similar to the ETS inhibitor and each featuring a terminal sulfonamide group, will be screened using GOLD. These ligands, `ligand.mol2`, are available from the folder to which you copied the tutorial4 files.

If you have opened all of the files above, close them by going to **File**, **Close All Files**.

A configuration file (`gold.conf`) has been provided for this tutorial which will automatically load the settings and parameter values for this tutorial into the GOLD Setup window.

In Hermes click on **GOLD**, then **Setup and Run a Docking** in the top-level menu. Load the `gold.conf` for tutorial 4 by selecting **Load Existing** from the resultant pop-up window and navigating to the directory where the `gold.conf` is stored and clicking **Open**.

22.4.3 Distance Constraints

Any distance between a ligand atom and a protein atom can be constrained, or restrained, to lie between minimum and maximum distance bounds.

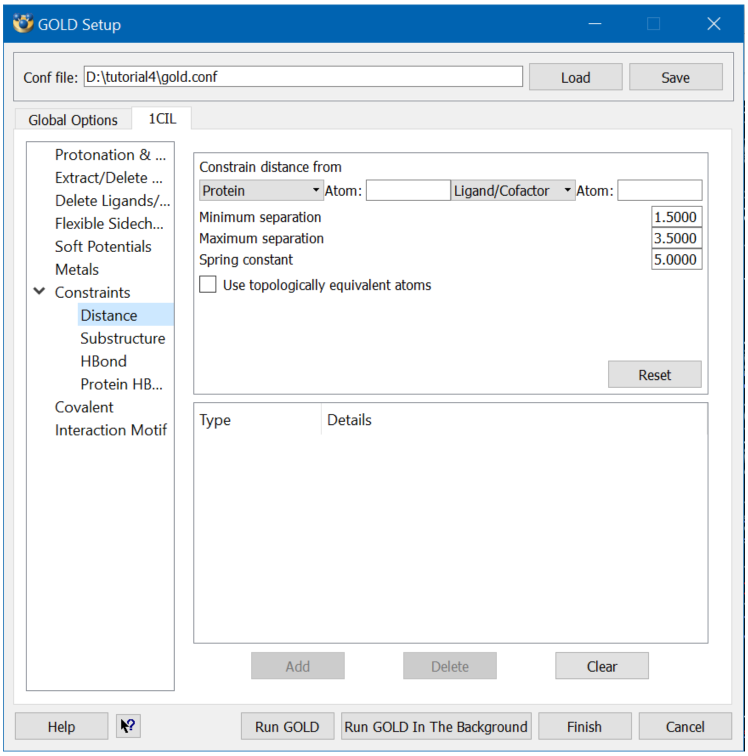
GOLD features two types of distance constraint:

- A standard distance constraint for use with individual ligands (see Standard Distance Constraints).
- A substructure-based distance constraint for use with multiple ligands which have a common functional group (see Substructure-Based Distance Constraints).

22.4.4 Standard Distance Constraints

Distance-based constraints are specific to each protein, thus click on the **1CIL** tab to access all protein-specific aspects of the docking setup.

Hit the triangle adjacent to **Constraints** in the list of available options to expand the Constraints tree, then select **Distance**.



Conf file: D:\tutorial4\gold.conf

Minimum separation 1.5000

Maximum separation 3.5000

Spring constant 5.0000

☐ Use topologically equivalent atoms

When setting up a distance constraint it is necessary to select both atoms involved in the constraint within the 3D view. Alternatively, the protein and ligand/cofactor atom labels can be typed into the **Protein** and **Ligand/Cofactor** windows (note you will have to hit return to update the 3D view). The maximum and minimum separation of the constrained atoms must also be entered (distances are in Å).

During a GOLD run, if a constrained distance is found to lie outside the specified bounds, a spring energy term is used to reduce the fitness score. This spring energy term has the form (E) = kx^2, where x is the difference between the distance and the closest constraint bound and k is a user-defined spring constant.

22.4.5 Substructure-Based Distance Constraints

It is possible to apply a distance constraint to multiple ligands which have a common substructure or functional group.

In order to use a substructure-based distance constraint, it is first necessary to create a file containing the common substructure in mol2 format.

The substructure-based constraint forces GOLD to limit the distance between a protein atom and one atom of this substructure.

During docking the constraint will be applied to any ligands which contain the specified substructure (matching is performed on the basis of the atom types and 2D connectivity) and the resulting solutions will be biased towards the specified distance range.

Click on **Substructure** within the **Constraints** tree to open the substructure constraint set-up window.

We now need to select the atoms involved in the constraint, specifically the zinc atom and the N atom in the `substructure.mol2` file.

The zinc atom is coloured grey in the 3D view. Click on the metal atom: the metal atom will be highlighted with a yellow sphere and the atom A:ZN261:ZN will be entered into the **Protein atom** dialogue.

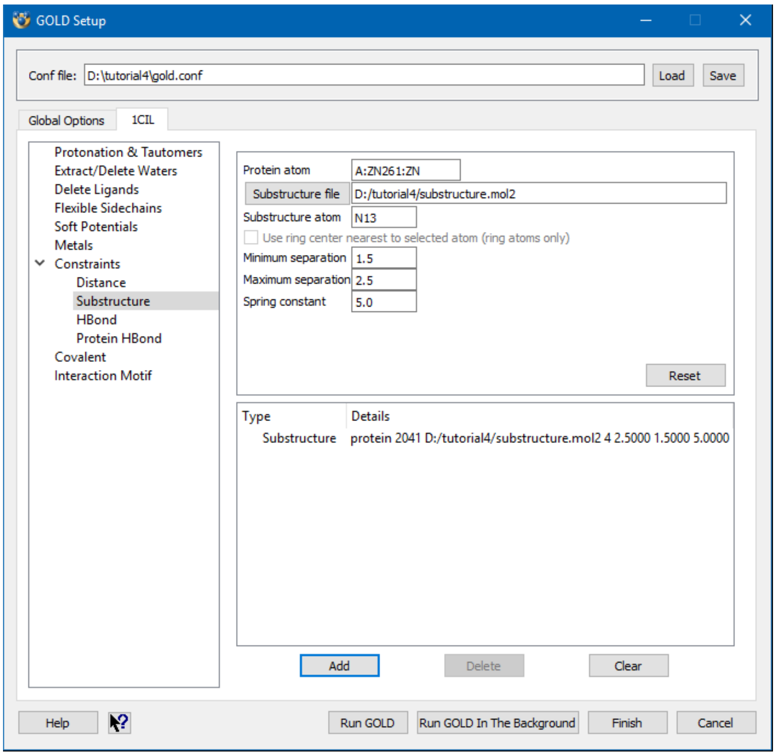
A substructure file (`substructure.mol2`) containing a sulfonamide group has been provided for this tutorial and can be found in the folder to which you copied the tutorial4 files. When creating your own substructure files, it is recommended that you set atom types manually (see Setting Up Substructure-Based Distance Constraints) since an incomplete fragment can cause problems with automatic atom-typing.

Click on the **Substructure file** button, then select the file `substructure.mol2` from the folder to which you copied the tutorial4 files and hit **Open**. This automatically loads the substructure file into the 3D view. If the substructure is not clear, you can suppress the protein atoms via the tickbox adjacent to **1CIL** in the **Molecule Explorer**.

Select the N atom in the substructure. This will enter the atom label **N13** into the **Substructure atom** dialogue. The N atom will also be surrounded by a cyan sphere in the 3D view.

Specify the allowed range of separation by entering a **Minimum separation** of 1.50 and a **Maximum separation** of 2.50 (distances are in Å).

As with standard distance constraints, the fitness score is reduced for solutions which do not satisfy the constraint. The amount by which the score is reduced is determined by a user-defined weight term. Set the value of the **Spring constant** to 5.0, then click on the **Add** button to add the constraint to the constraints list.



Conf file: D:\tutorial4\gold.conf

Protein atom: A:ZN261:ZN

Substructure file: D:/tutorial4/substructure.mol2

Substructure atom: N13

□ Use ring center nearest to selected atom (ring atoms only)

Minimum separation: 1.5

Maximum separation: 2.5

Spring constant: 5.0

Type	Details
Substructure	protein 2041 D:/tutorial4/substructure.mol2 4 2.5000 1.5000 5.0000

Click on the **Global Options** tab to return to the general docking setup window.

22.4.6 Running GOLD

The time taken by GOLD to dock ligands can be controlled by altering the values of the genetic algorithm (GA) parameters (see Controlling Accuracy and Speed with Genetic Algorithm Parameter Settings). GOLD runs for a fixed number of genetic operations (crossover, migration, mutation). Therefore, reducing the number of GA operations performed during the course of a run will result in GOLD running faster, however the search will be less exhaustive.

GOLD can decide on the optimal settings to use for a given ligand (see Controlling Accuracy and Speed with Genetic Algorithm Parameter Settings).

To enable automatic GA settings, click on the **GA Settings** option in the list of available options, then activate the **Automatic** radio button. The **Search efficiency** will by default be set to 100%: we will use the default settings.

We now need to specify an output directory. Click on **Output Options** and specify a directory to which you have write permission. This is where the GOLD output files will be written.

Now click on the **Run GOLD** button at the bottom of the interface. In the **Finish GOLD Configuration** window, you will be prompted that the GOLD configuration has been updated and needs to be saved; click **Save** to proceed. The configuration file name can remain as it is, so hit **OK** to overwrite the existing `gold.conf`. This will start the GOLD job interactively. As the job progresses output will be displayed in the **Run GOLD** window.

Any warning messages produced will be displayed under the **gold.err** tab.

Once the job is complete, the message Finished Docking Ligand: ligand.mol2 will appear in the **gold_ligand_m1.log** tabbed view. Once you view this message, load the results into Hermes using the **View Solutions** button then close the **Run GOLD** window using the **Close** button.

22.4.7 Analysis of Output

A file called `bestranking.lst` is written to the specified output directory for batch jobs. Open this file and inspect it using a text editor; it gives a continuous summary of the best solution that has been obtained for each docked ligand.

The listed file names correspond to the names of the files containing the best solution found for each ligand.

The file gives total fitness scores and a breakdown of the fitness into its constituent energy terms.

An additional constraint scoring term `S(con)` is also listed. For docking solutions which satisfy the specified distance constraint the contribution from this scoring term will be 0.00. However, for solutions in which the constrained distances lie outside the specified bounds a negative `S(con)` score will be applied, thus reducing the overall fitness.

Further details relating to substructure-based constraints are given within individual ligand log files. Your output directory should contain ten ligand log files `gold_ligands_m#.log`, one for each ligand.

Open and inspect the ligand log file corresponding to the first ligand in the input file, i.e. `gold_ligands_m1.log`. This file will contain the distance bounds as specified in the constraint and the actual distance observed in the docked solution.

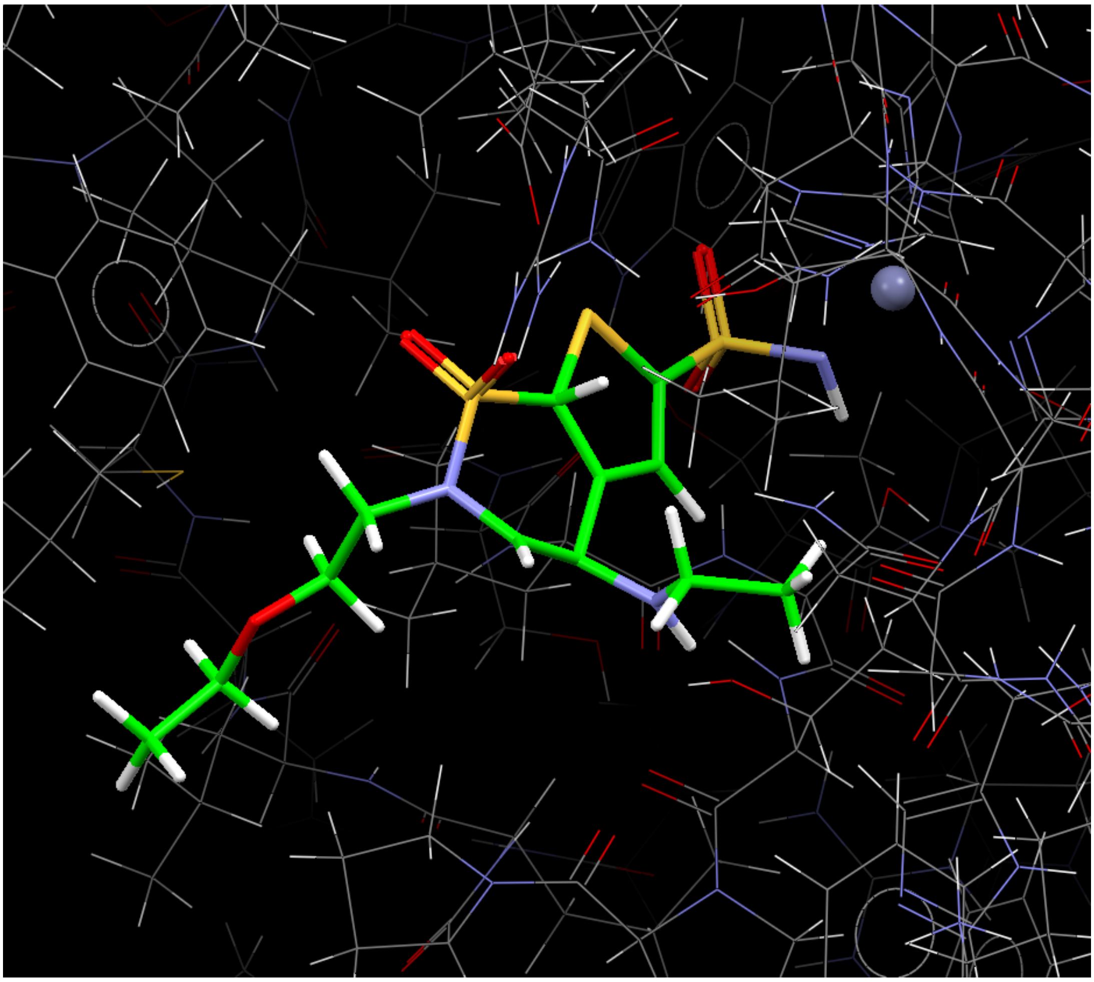
From your `bestranking.lst` file identify GOLD’s top-ranked solution for the ligand with the best total fitness score.

Go to the Hermes 3D view. The overall top-ranking ligand can be viewed by ordering the ligands based on their fitness score. To do this, go to the **Molecule Explorer** and find the column labelled

Goldscore.Fitness in the **Docking Solutions** view. Click on this column either once or twice until you have the best fitness score (i.e. the highest value) listed at the top of the column.

The position and orientation of the terminal sulfonamide groups in the docked solutions should be similar to that observed in the co-crystallised ETS inhibitor (i.e. coordinated to the zinc within the protein via the sulfonamide nitrogen).

In the example below the terminal sulfonamide group of GOLD’s top-ranked solution can be seen to satisfy the specified constraint and reproduces the known binding mode of the co-crystallised ETS inhibitor:



This ends the tutorial.

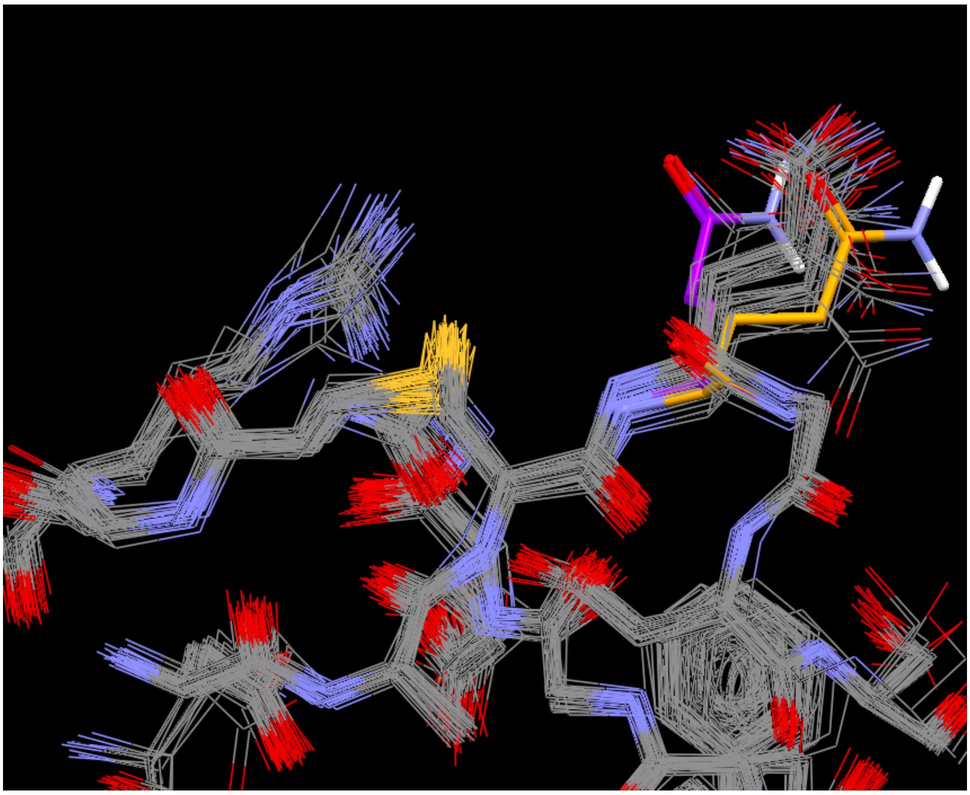
22.5 Tutorial 5: Docking with a Flexible Side Chain

22.5.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial5` to a directory to which you have write permissions.

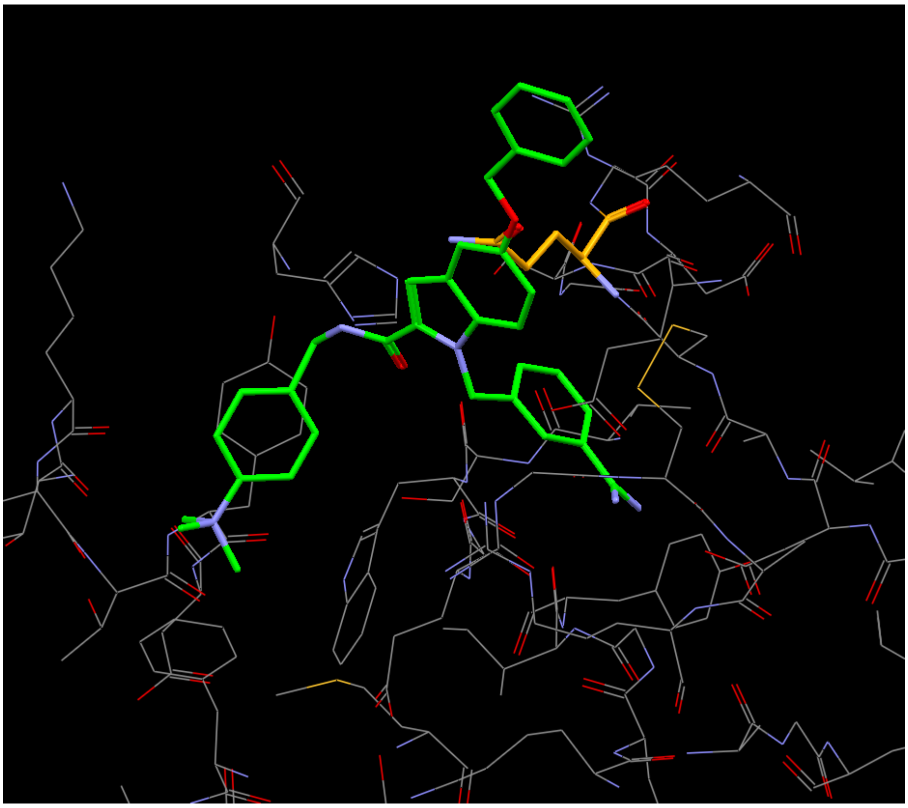
The object of this tutorial is to demonstrate how to dock a ligand into a binding site which is known to contain a flexible side chain. The example will involve docking the ligand from PDB entry 1lpg into the protein binding site taken from 1fax. These structures are of blood coagulation factor Xa, complexed with two different ligands.

The figure below shows a superposition of several experimental determinations of the factor Xa binding site, complexed with a variety of different ligands (not shown), Only a small part of the binding site is displayed.



While it is clear that parts of the binding site are rigid, their positions hardly moving from one structure to the next, other parts are more inclined to move. In particular, the residue at the top right-hand corner of the plot, Gln192, adopts a variety of positions according to which ligand is bound. The Gln192 position highlighted in orange is taken from 1lpg, the one shown in purple is taken from 1fax.

The next figure was produced by superimposing 1lpg and 1fax. It shows the 1fax binding site (with grey C atoms) and the 1lpg ligand (with green C atoms). Gln192 is highlighted with orange C atoms. It is immediately clear that the 1lpg ligand cannot be docked accurately into the 1fax binding site if Gln192 is not allowed to move, since there is a severe steric clash between these two.



To see this more clearly, you can open Hermes and read in the file `1fax_1lpg_super.mol2` from the folder containing the tutorial5 files via **File**, **Open**. This is the superposition from which the above figure was generated.

22.5.2 Preparation of Input Files

The file `1fax_protein.mol2` contains the binding site from 1fax. It has been set up for docking in the normal way. Hydrogen atoms have been placed on the protein in order to ensure that ionisation and tautomeric states are defined unambiguously (see Essential Steps).

The ligand from 1lpg has also been set up for docking (see Essential Steps). It is stored in `1lpg_ligand.mol2`. Again, attention has been given to protonation states (e.g. the benzamidine group has been built in its protonated form) and the bond types have been set in accordance with GOLD conventions.

These two files may be viewed in Hermes if desired.

22.5.3 Example conf Files and Output Files

Two GOLD configuration files have been prepared:

- `non_flexible.conf`: this file was set up in the normal way using the GOLD front end. It corresponds to a standard docking of the 1lgp ligand into the 1fax binding site, using slow search settings (100,000 GA operations) and allowing no side chain flexibility. The considerations outlined in the preceding part of this tutorial suggest that this docking protocol is unlikely to give good results. The corresponding output can be found in the `non_flexible` subdirectory.
- `flexible.conf`: this file defines a docking in which the Gln192 side chain is allowed to move. It was set up using the **Flexible Sidechains** option in the GOLD Setup window in Advanced mode. The corresponding output can be found in the `flexible` subdirectory.

  The processes used to setup and run these dockings are covered in the sections that follow. If you wish, you can run the two GOLD jobs using the configuration files provided. Alternatively, you can view the results that we have generated. Since GOLD is non-deterministic, any results that you get might differ from ours, but the general trends are likely to be the same.

22.5.4 Running the Non-flexible Docking and Analysing the Results

Open Hermes and specify the `non_flexible.conf` via **GOLD**, **Setup and run a Docking**, **Load Existing**.

Click on the **1FAX** tab then click on the **Flexible Sidechains** option. The defined active site in the protein has been broken down into its constituent residues which are provided in a scrolling list. Scan through the list: you will notice that the **Status** of all the residues is listed as **Rigid**.

Return to the **Global Options** tab (where we can define general docking settings) and click on **Output Options**. Change the output directory name to e.g. `non_flexible2` then click on **Run GOLD**. Change the GOLD conf file name to e.g. `non_flexible2.conf`. Start the docking by clicking on **Save**.

Once the docking has completed, load the solutions into Hermes using the **View Solutions** button, then **Close** to close the **Run GOLD** window and then click on **Cancel** in the **GOLD Setup** window.

Load the `1fax_1lpg_super.mol2` superposition file into Hermes via **File**, **Open**.

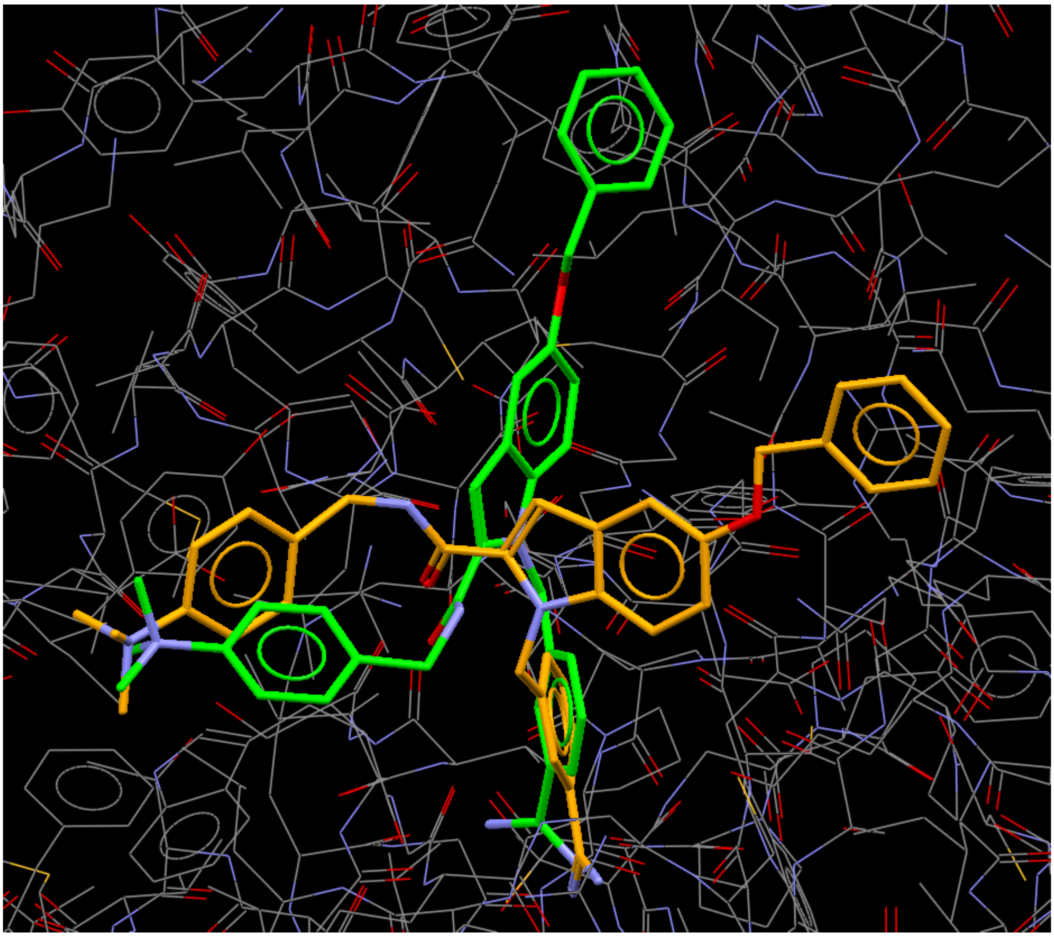
Hit the **Display** tab in the **Molecule Explorer** window of the interface. This tabbed window lists all component parts of the visualiser display e.g. **1FAX** corresponds to the docked protein and ligands.

Deselect all radio buttons in the list apart from **1FAX** and **1lpg_ligand_reference** (the experimental position of the 1lpg ligand). You may wish to hide the H atoms by deactivating the **Show hydrogens** tickbox at the top of the Hermes interface if the display is not clear.

Return to the **Docking Solutions** tab and select the ligand with the highest fitness score. Scores are tabulated under the **Goldscore.Fitness** header in the **Molecule Explorer** window. Solutions can be ordered on fitness score by clicking on the **Goldscore.Fitness** header. You can scroll through the other docking solutions simply by clicking on them.

Note that, everytime you click on one of the docking solution, the experimental position the 1lpg ligand is disabled. To display the structure again in the 3D view, tick the **1lpg_ligand_reference** tickbox in the **Display** tab of the **Molecule Explore** window.

As expected, none of the solutions produced in our non-flexible run is correct; all have the benzyloxy side chain seriously misplaced. The top-ranked docking has a GoldScore of 63.4848 and is shown below with the true ligand position (C atoms coloured orange) for reference:



22.5.5 Running the Flexible Docking and Analysing the Results

Clear the display by going to **File**, **Close All Files**.

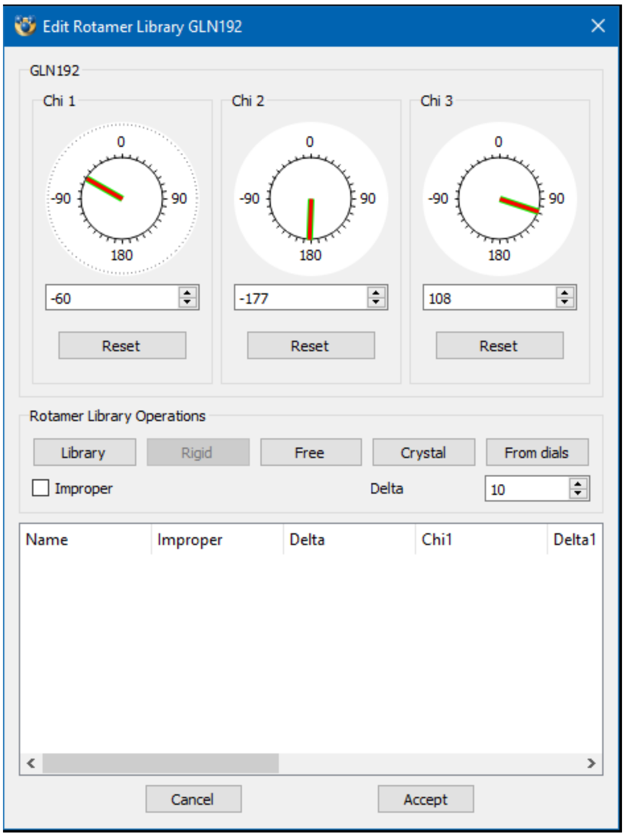
At this point you can choose to load the existing `flexible.conf` to view the results of the flexible docking. Alternatively, continue with the tutorial to see how to set up a flexible docking.

Load the `non_flexible.conf` via **GOLD**, **Setup and run a Docking**, **Load Existing**.

The settings in this file are for the non-flexible docking; click on the **1FAX** tab and click on the **Flexible Sidechains** option where we will be able to specify sidechains that will be allowed to rotate during docking.

Scroll down the list of residues until you get to **GLN192**. Click on **GLN192** then hit the **Edit** button (alternatively simply double-click on **GLN192**).

This launches a dialogue where we can add, edit or delete rotamers.



Chi 1: -60

Chi 2: -177

Chi 3: 108

□ Improper

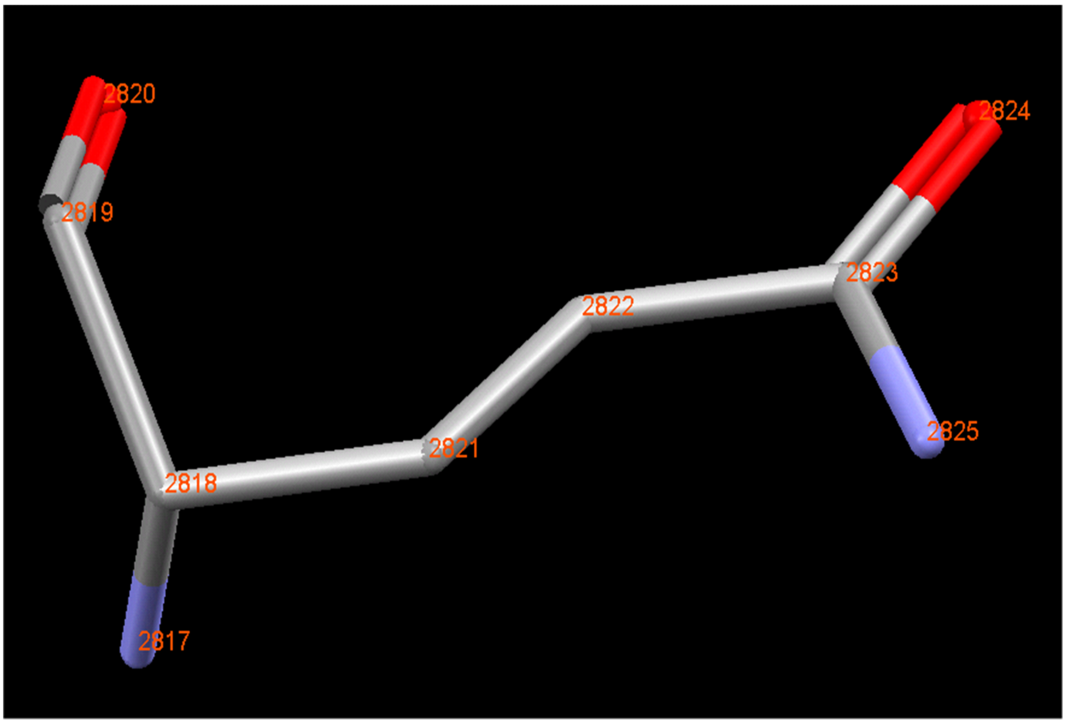
Delta: 10

Name	Improper	Delta	Chi1	Delta1

The dialogue is blank as we have yet to define any rotamers for GLN192.

Click on the **Library** button. The dialogue has been updated to reflect rotamers taken from the rotamer library information in The Penultimate Rotamer Library, S. C. Lovell et al., Proteins, **40**, 389-408, 2000. It is a compilation of the most commonly observed side chain conformations for the naturally occurring amino acids. The `rotamer_library.txt` file can be viewed in txt format in `<Installation folder>/ccdc-software/gold/GOLD/gold`. Note that the library settings are simply a starting point; users are encouraged to generate their own rotamers with the **Edit Rotamer Library** dialogue for optimal results.

The (nine) allowed rotamers are listed at the bottom of the **Edit Rotamer Library GLN192** window. Collectively these lines define the torsional flexibility that the Gln192 side chain will be allowed to have during docking.



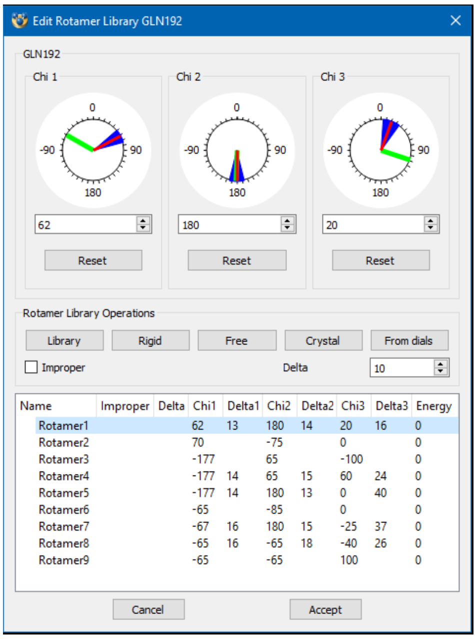
There are a number of parameters listed alongside each rotamer:

Chi1 is the first rotatable torsion in the side chain. In the case of GLN192 this corresponds to rotation around Cα-Cβ, so the atoms will be the backbone N, (= atom 2817), CA (2818), CB (2821) and CG (2822).

Chi2 is the second rotatable torsion and corresponds to rotation around Cβ-Cγ, so the atoms are CA (2818), CB (2821), CG (2822) and CD (2823).

Chi3 is the third rotatable torsion, corresponding to rotation around Cγ-Cδ so the atoms are CB (2821), CG (2822), CD (2823) and terminal N (2825).

Associated with each numbered Chi value is a Delta value.



Chi 1: 62

Chi 2: 180

Chi 3: 20

□ Improper

Delta: 10

Name	Improper	Delta	Chi1	Delta1	Chi2	Delta2	Chi3	Delta3	Energy
Rotamer1			62	13	180	14	20	16	0
Rotamer2			70		-75		0		0
Rotamer3			-177		65		-100		0
Rotamer4			-177	14	65	15	60	24	0
Rotamer5			-177	14	180	13	0	40	0
Rotamer6			-65		-85		0		0
Rotamer7			-67	16	180	15	-25	37	0
Rotamer8			-65	16	-65	18	-40	26	0
Rotamer9			-65		-65		100		0

Click on the **Rotamer1** in the rotamer list:

Rotamer1 specifies the first set of allowed values for **Chi1**, **Chi2** and **Chi3**, i.e. Chi1 = 62, Chi2 = 180, Chi3 = 20.

The Delta values associated with the Chi values are Delta1 = 13, Delta2 = 14 and Delta3 = 16. These Delta values specify the allowed range e.g. (Delta1 - Chi1) to (Delta1 + Chi1).

Each rotamer therefore describes one allowed conformation of the side chain as defined by the torsion angles values (Chi1, Chi2, Chi3) and their allowed ranges (Delta1, Delta2, Delta3).

The dials at the top of the window reflect the rotamer information for the currently loaded rotamer, **Rotamer1** in this case. Allowed rotation values for other rotamers can be viewed in the dials by clicking on each Rotamer line in turn. The settings on the dials describe the following:

- Green: the observed torsion in the protein
- Red: the defined rotamer (Chi value) that will be used during docking
- Blue wedge: the tolerance allowed (Delta value) for the defined rotamer
- There are a number of other options for setting rotamers other than the library settings:
- **Rigid**: this fixes a particular side chain at its input conformation, i.e. makes it non-flexible during docking.

- **Free**: this allows a side chain to rotate freely during docking, i.e. the defined rotatable torsion will be permitted to vary over the range -180 to +180.

- **Crystal**: this setting will define a rotamer in which all rotatable torsions in the side chain will be allowed to vary over the range (Delta - Chi) to (Delta + Chi); where Chi values are taken from the protein input file.

- **From dials**: this allows rotamers to be specified directly. Start by setting each Chi value: click on the dial and while holding down the mouse button move the red indicator line to the required position. The corresponding torsion will rotate within the Hermes visualiser to show the current value. Alternatively, type the required Chi value into the entry box directly under the dial. When all Chi values are as required, press the **From dials** button to accept the rotamer definition.

We will use the **Library** settings for this docking so if you have modified any of the rotamer settings from those initially loaded, hit **Rigid** to reset the rotamer settings then hit **Library** again. Hit **Accept** to close the rotamer definition window.

GLN192 is now listed as being **Constrained** to **9 rotamers** in the **Flexible Sidechains** window.

We only require GLN192 to be flexible for the purposes of this example, however using this method we can specify up to 10 rotatable side chains if we wish.

Click on the **Global Options** tab and within the **Output Options** window, change the output directory to e.g. `flexible2` then click the **Run GOLD** button.

Change the name of the GOLD conf file to e.g. `flexible2.conf` then click **Save**. Once the docking has finished, load the results into Hermes using the **View Solutions** button, then click **Close** in the **Run GOLD** window and click **Cancel** in the **GOLD Setup** window.

The following describes the output in the `flexible` directory provided with the tutorial. If you have set-up and run your own flexible docking using the instructions above, your output may vary slightly however the general trends should be the same.

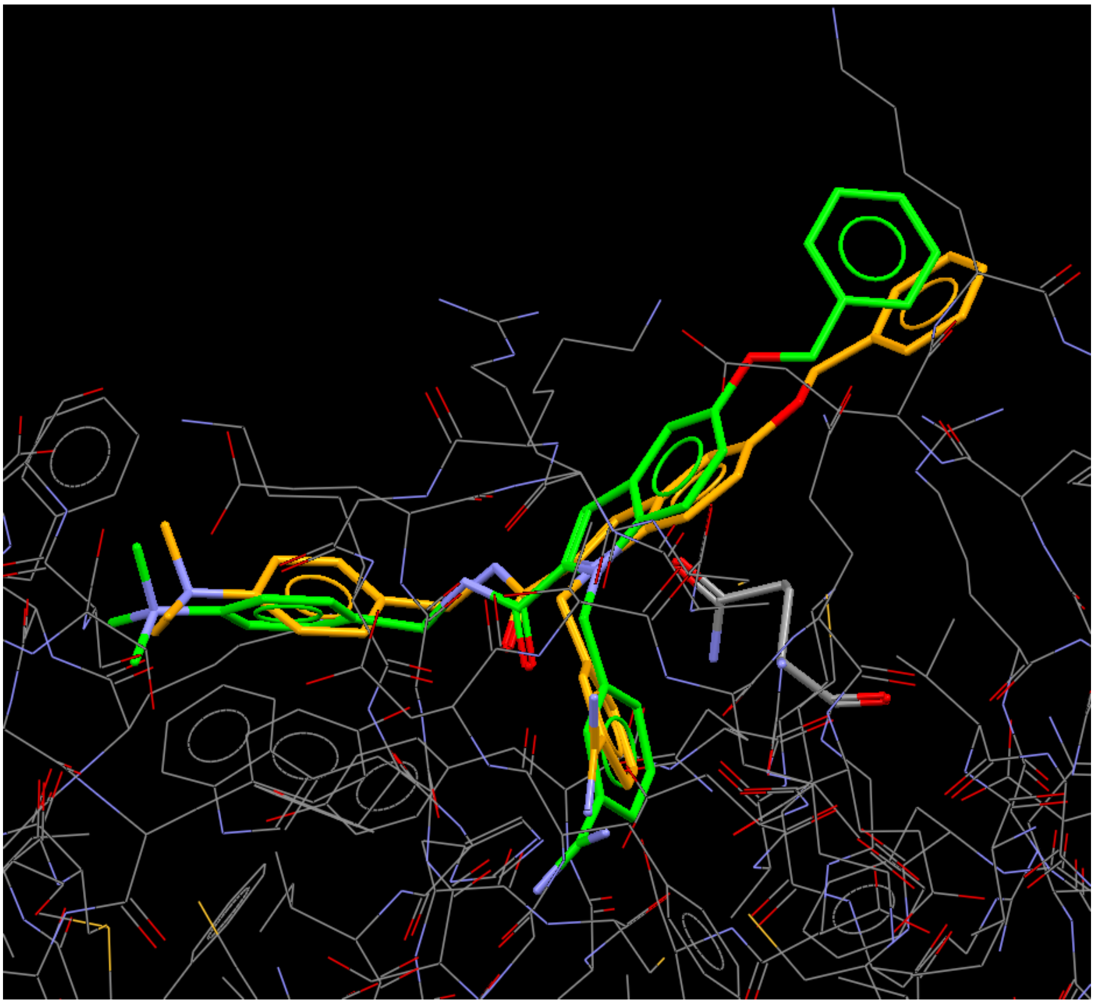
To compare the top-ranked solution with the experimental position of the 1lpg ligand, load the `1fax_1lpg_super.mol2` superposition file into Hermes via **File**, **Open**.

Hit the **Display** tab in the **Molecule Explorer** window of the interface and deselect all radio buttons in the list apart from **1FAX** and **1lpg_ligand_reference** (the experimental position of the 1lpg ligand).

Return to the **Docking Solutions** tab and order the solutions on the fitness score by clicking on the **Goldscore.Fitness** header.

Click on the top-ranked solution of the flexible run to display it in the 3D view. From the **Display** tab select the radio button of **1lpg_ligand_reference** to display the experimental position of the 1lpg ligand in the 3D view.

The top-ranked solution of the flexible run is much better than the top-ranked solution of the rigid run, it is not perfect - in particular, the benzamidine moiety is somewhat displaced - but the benzyloxy side chain is now roughly in the right position because the Gln192 side chain having moved out the way (reference ligand C atoms coloured orange, docking result C atoms coloured green):



Also, the best solution from the flexible run has a higher GoldScore value (74.6248) than the best solution that was obtained from the rigid run (62.1221).

The movements of the flexible side chain Gln192 can be seen more effectively if the representation style of the Gln192 residue is changed.

From the **Display** tab of the **Molecule Explorer**, open the **1FAX** tree to see its components; open up chain **A** to look at the residues making up this protein chain.

Right-click on **GLN192** in the **Molecule Explorer** list and choose **Styles**, **Capped Sticks**.

22.5.6 Choosing Side chain Rotamers

Two decisions must be made when using the flexible side chain facility: (a) which side chains are made flexible; (b) how flexible is each side chain made? It is important to recognise that the more flexibility is introduced, the larger the search space becomes. Particularly with high-throughput runs, when relatively little time can be allowed per ligand, this may seriously decrease the chance of finding the global minimum.

A sensible strategy is therefore to make a side chain flexible only if you have some a priori reason to suppose that it will move, as we have (from X-ray structures) in the tutorial example.

On the other hand, we probably allowed Gln192 more movement than necessary in the above experiments. As long as it can adopt the native 1fax position and one other position in which it is folded away from the binding site, that might well have been enough.

One problem is that, in some conformations, Gln192 tends to clash with Arg143. At first sight, this means we have to be careful to pick a Gln192 rotamer that is folded away from the binding region but also does not clash with this arginine residue. A way round this is to add the command `penalise_protein_clashes = 0` to the `rotamer_lib` command block (place it anywhere between `rotamer_lib` and `end_rotamer_lib`). This will switch off calculation of clashes between flexible side chain atoms and neighbouring protein atoms, allowing Gln192 to approach nearby residues closely. While physically unrealistic, this is a pragmatic tactic that might well work (and is not as egregious as it sounds, since, in reality, Arg143 can probably move away from Gln192 if it needs to).

You can experiment with these options if you wish.

This ends the tutorial.

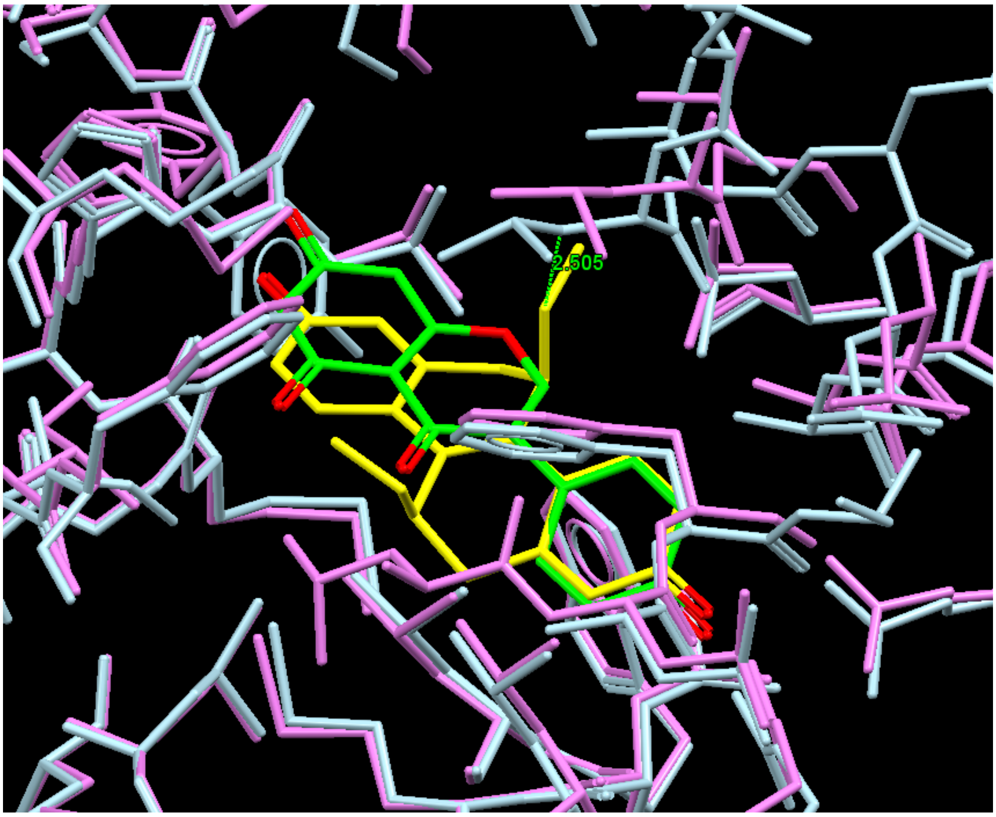
22.6 Tutorial 6: Docking using Localised Soft Potentials

22.6.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial6`to a directory to which you have write permissions.

The object of this tutorial is to demonstrate how to employ the Localised Soft Potential option that is available when using GoldScore. This option allows you to soften the vdW clash component of the GoldScore for one or more residues in the protein. We will examine the docking of a ligand to two different crystal structures of Oestrogen Receptor Alpha. The structures differ in that a small loop movement constrains the binding site of one of the structures (PDB code 1x7r) slightly more than for the other structure (PDB code 1l2i).

The figure below shows the superposition of both protein structures, where 1x7r corresponds to the protein coloured in light blue and the ligand with green C atoms, and 1l2i corresponds to the protein coloured in purple and the ligand with yellow C atoms.



Most of the binding site is well superimposed however above the ligands you can see that there is movement of a protein loop that brings Leu346 closer in to the ligand in 1x7r than in 1l2i. This superposition suggests that a clash would exist if the ligand from 1l2i were docked into 1x7r. This might prevent the correct binding mode being rated highly if using a scoring function such as GoldScore, with a clash term that increases sharply with proximity to the protein. Other residues such as Met343 also do not superimpose well as a consequence of this loop movement. However, these residue shifts appear to have less of an impact on the size of the active site than does that of Leu346.

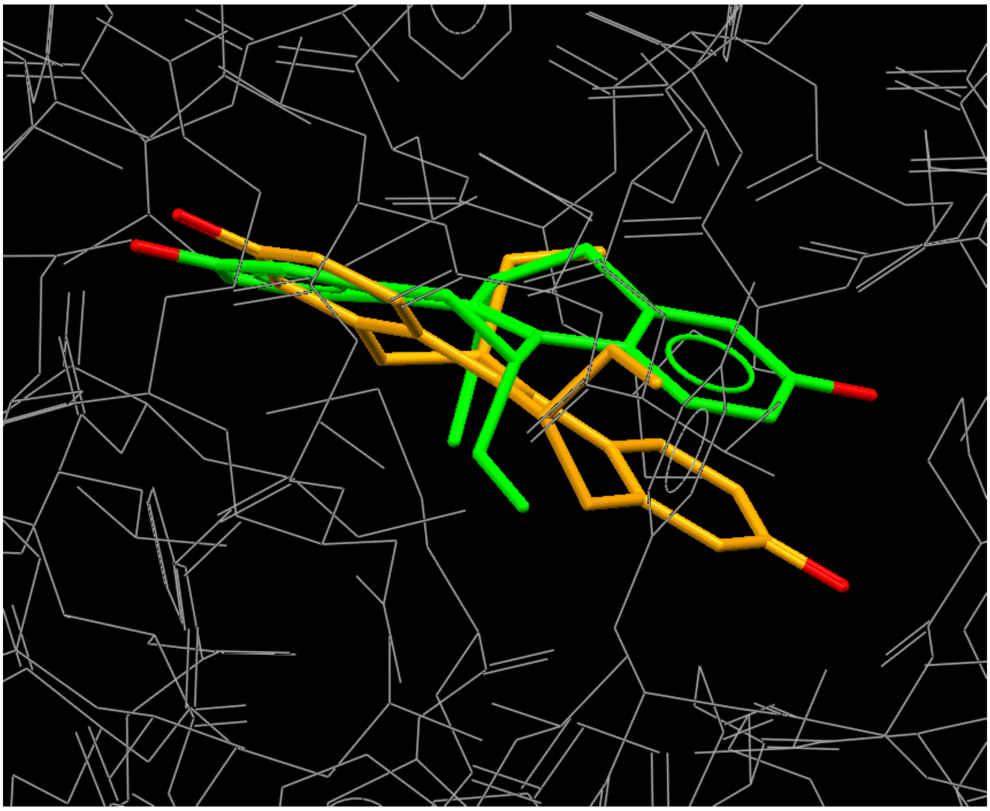
You can view this superposition yourself by opening Hermes and reading in the file `1x7r_1l2i_sup.mol2`.

22.6.2 Docking With No Soft Potential Applied

The files `1l2i_prot.mol2` and `1x7r_prot.mol2` are the protein models derived from the pdb entries 1l2i and 1x7r respectively. The file `1l2i_lig.mol2` is the ligand structure obtained from 1l2i and in the same frame of reference as 1l2i.

The GOLD configuration file `gold_1l2i_1l2i.conf` is set up to dock the 1l2i ligand back into the 1l2i protein structure. Load the file into GOLD via **GOLD**, **Setup and Run a Docking, Load Existing**, then navigate to the folder containing the tutorial6 files, select `gold_1l2i_1l2i.conf` and click **Open**. Run this GOLD job. Once it has finished, the docking results can be read directly into Hermes by clicking on the **View Solutions** button. Analyse the results in Hermes to check that the crystallographic binding mode is indeed retrieved, reading in the file `1l2i_lig.mol2` to make the comparison.

The GOLD configuration file `gold_1x7r_1l2i.conf` is set up to dock the 1l2i ligand into the 1x7r protein structure. Load the `gold_1x7r_1l2i.conf` as described above, run this GOLD job then analyse the results in Hermes. Read in the file `1x7r_1l2i_sup.mol2` to compare the docked poses with the binding mode found in 1l2i (i.e. with the ligand component of the `1l2i_binding_site` entry in **Molecule Explorer**). You may find that there are some solutions which have approximately the right binding mode which return scores of between 23 and 25. However there should also exist higher ranking poses with scores of between 28 and 33. These poses have the ligand rotated through 180 degrees along the long axis as shown in the superposition below (crystallographic binding mode with orange C atoms, GOLD docking pose with green C atoms for the ligand and grey atoms for the protein chain).



22.6.3 Cross-Docking into 1x7r with a Soft Potential applied to Leu346

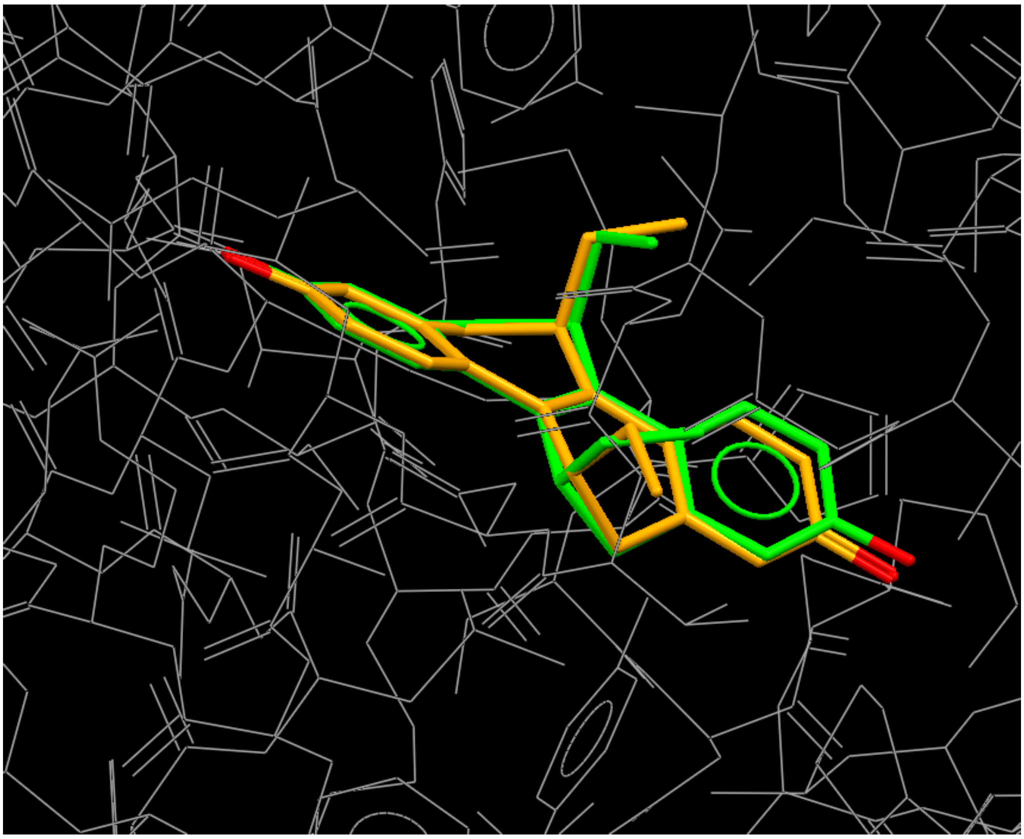
Load the file `gold_1x7r_1l2i_SP.conf` into Hermes. The definition of soft potentials is specific to the protein thus click on the **1x7r** tab (adjacent to the **Global Options** tab). Click on **Soft Potentials**.

You will notice that **LEU346** is already in the **Residues - alternative potential 1** box. This means that a soft vdW potential with 2-4 functional form has been applied to one residue only, Leu346. This replaces the default 4-8 functional form that applies to the rest of the protein. Note: Potentials are applied simply by choosing the appropriate potential (i.e. either 1 or 2) and activating the **Add selections** radio button adjacent. Residues are then added to the appropriate box by selecting them in the 3D view.

Because **LEU346** is in the **Residues - alternative potential 1** box, this means a 2-4 soft potential is applied to the residue. If **LEU346** had been entered into the **Residues - alternative potential 2**, this would mean a softer 1-2 functional form would be applied. Further information is available (see Allowing for Localised Movements: Docking with Soft Potentials).

Run the docking job `gold_1x7r_1l21_SP.conf` and analyse the results using Hermes.

This time you should find that the highest scoring solutions correspond very closely with the 1l2i binding mode (see below). These solutions will have scores above 40. The reversed binding mode may still appear in some solutions, but these invariably have much lower scores close to 30 (if they appear at all).



This ends the tutorial.

22.7 Tutorial 7: Generating Diverse Solutions

22.7.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial7` to a directory to which you have write permissions.

The object of this tutorial is to investigate PDB code 3MTH, pig hormone complexed with methylparaben insulin. The binding site is large and primarily hydrophobic in nature with a small number of acceptor regions. The ligand is small, thus there is potential for obtaining an incorrect docking pose or poses. Consequently, GOLD does not perform well when attempting to replicate the binding mode of the ligand.

3MTH is an entry in the CCDC/Astex validation test set which is available to download as CCDC Astex Validation Set in the Validation Test Set section from: https://www.ccdc.cam.ac.uk/support-and-resources/downloads/

A water molecule which mediates protein-ligand binding in the native crystal structure has been reinstated in the `protein.mol2` file provided with this tutorial (all water molecules were removed from the protein files for validation so is not present in the structure in the CCDC/Astex validation test set). Treating this water molecule explicitly does not improve the standard GOLD docking results significantly (i.e. only using the diverse solutions feature improves the docking results).

This tutorial will illustrate how to use GOLD to generate diverse solutions and show how this feature can be used to improve the outcome of docking methylparaben insulin back into its native protein.

22.7.2 Preparation of Input Files

The original PDB file (`3MTH.pdb`) has been provided should you wish to set up the protein and ligand files yourself.

Protein and ligand files are also provided and have been set up in accordance with guidelines for the preparation of input files (see Setting Up the Protein(s) and Setting Up Ligands respectively).

These files can be opened in Hermes and inspected. You will be able to see the extent of the protein active site compared to the size of the ligand.

22.7.3 GOLD Configuration Files

Two GOLD configuration files are provided in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial7`. The settings in these files and how to run the files is covered in the sections that follow:

- `standard.conf` (see Running standard.conf and Viewing the Results).
- `diverse.conf` (see Running a Diverse Solutions Docking and Viewing the Results).

22.7.4 Running standard.conf and Viewing the Results

The `standard.conf` configuration file contains settings for carrying out a standard docking, i.e. without generating diverse solutions. Output files have already been generated for this docking and are provided in the standard directory. These results can be viewed directly by opening Hermes and reading in the `standard.conf` via **Load GOLD results**. Alternatively, GOLD can be run again following the instructions below.

Load the `standard.conf` into GOLD via **GOLD**, **Setup and Run a Docking, Load Existing**, then navigate to the folder containing the tutorial7 files, select `standard.conf` and click **Open**. This automatically loads the settings and parameter values for this tutorial into the GOLD Setup window.

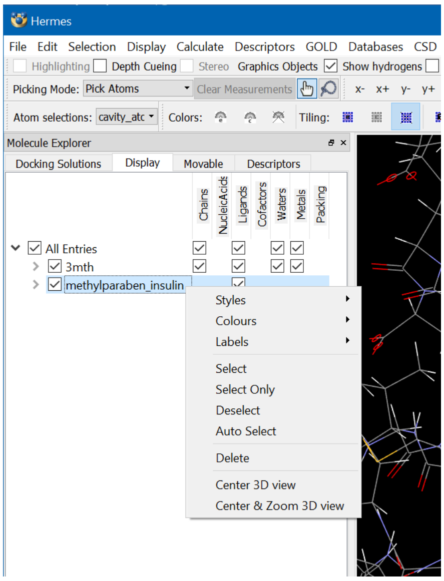
No settings need to be changed for the purposes of this docking however you may also wish to change the output directory. To do this, click on **Output Options** and either type the path to or browse to an appropriate output folder where the GOLD output files can be saved, e.g. naming it `standard_solutions`.

The GOLD run is started by hitting the **Run GOLD** button at the bottom of the interface. If you have changed the output directory, then in the **Finish GOLD Configuration** window, you will be prompted that the GOLD configuration has been updated and needs to be saved. We have not modified the `protein.mol2` file so we do not need to save this file, thus ensure the tick box adjacent to `protein.mol2` is deactivated. Change the configuration file name to e.g. `standard_solutions.conf` then hit **Save** to start the GOLD run.

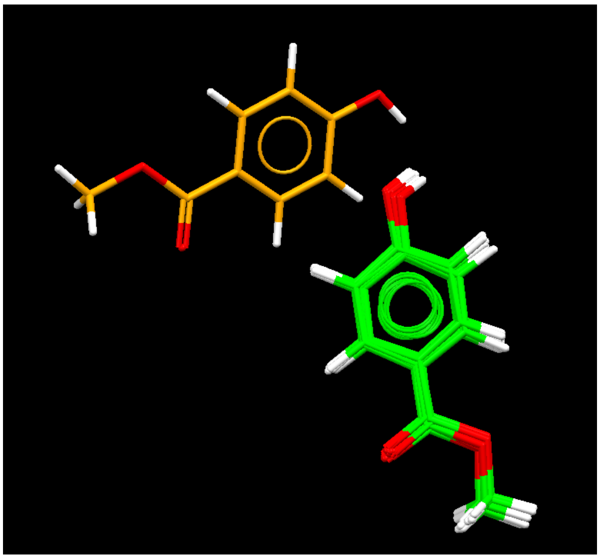
Once the docking is complete, the message Finished Docking Ligand: ligand.mol2 will appear in the **gold_ligand_m1.log** tabbed view of the **Run GOLD** window. The GOLD results can be read into Hermes via the **View Solutions** button in the **Run GOLD** window.

The input ligand can also be used as a reference ligand (i.e. the original pose of the ligand in the crystal structure) so read `ligand.mol2` in via **File**, **Open**.

The 3D display can be controlled by using the settings under the **Display** tab of the **Molecule Explorer**. Further options are available on right-clicking. For example, you can change the colour of the reference ligand so that C atoms are coloured orange.



All docked solutions can be viewed by returning to the **Docking Solutions** tab of **Molecule Explorer** and pressing the **Shift** keyboard key whilst using the mouse to select first the top then the bottom solution. You will notice that all the solutions are very similar.



None of the solutions replicate the original binding mode. The ligand OH group is H-bonding to the same CYS6 carbonyl group; however, the size of the protein active site means it is possible for the ligand to occupy an alternative area of the cavity than in the original crystal structure. Also, the water molecule known to mediate the protein-ligand interaction would not be used by the pose above.

The heavy atom RMSD of the top-ranked pose when compared to that of the co-crystallised ligand is 7.306 Å (as can been seen in the `ligand_m1.rnk` file in the output folder).

Clear up the 3D view by selecting **File** then **Close All Files**.

22.7.5 Running a Diverse Solutions Docking and Viewing the Results

The docking set-up is provided in `diverse.conf` and the corresponding output is in the diverse directory in the folder to which you copied the tutorial7 files. The docking results can be read into Hermes via **File**, **Load GOLD results**. Alternatively, the diverse solutions docking can be set up by modifying the conf file from the previous exercise; instructions on how to do this follow.

The GOLD interface will still be open. Click on the **Load** button to read the `standard.conf` from above.

Click on **Fitness & Search Options** then activate the **Generate diverse solutions** tickbox. The settings for generating diverse solutions can be modified; click on the **Diverse Solutions Options** button to view these settings.

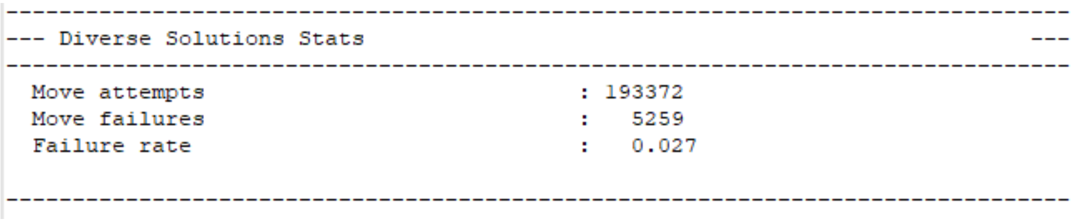
Change the **Cluster size** to 2 and the **R.M.S.D.** to 2.0 Angstroms. This means that each diverse solutions cluster will contain 2 ligands and that the clusters will differ by an RMSD of 2 Å (see Setting Up GOLD to Generate Diverse Solutions). Hit **Close** to close the window.

Now click on **Output Options** and change the directory from what it was previously to e.g. `diverse_solutions`. At the bottom of the window, activate the **Create links for different binding modes (based on RMSD clustering)** tick box, then enter 2.0 into the box next to **Distance between clusters**. Because we have defined the same value for RMSD as with our diverse solutions settings, the cluster shortcuts will point to the top-ranked solution in each of our diverse clusters (see Identification of Different Binding Modes (Clustering of Ligand Poses)).

Now hit the **Run GOLD** button to start the docking. Change the GOLD conf file name to e.g. `diverse_solutions.conf` (this will ensure a new conf file is saved rather than overwriting the original file), then hit **Save** to start the docking.

Once the docking has completed, load the results into Hermes by hitting the **View Solutions** button.

Before closing the Run GOLD window, inspect the `gold_ligand_m1.log` file. Diverse solution information is given for each docked ligand under the heading `Diverse Solutions Stats`.



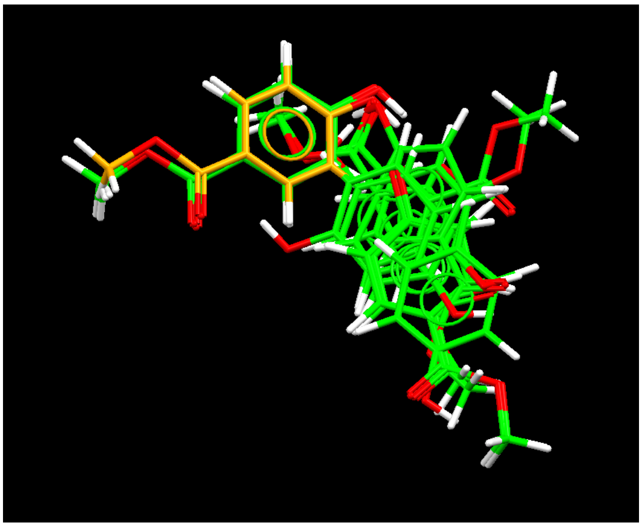
```
--------------------------------------------------------------------------------
--- Diverse Solutions Stats                                                  ---
--------------------------------------------------------------------------------
  Move attempts                         : 193372
  Move failures                         :   5259
  Failure rate                          :   0.027

--------------------------------------------------------------------------------
```

These stats are explained elsewhere (see Method Used to Generate Diverse Solutions).

Cluster information can be found at the end of the file.

In the provided results, at the 2.34 Å cut-off there are 5 clusters.

Close the **Run GOLD** window by hitting the **Close** button. As before, load the reference file `ligand.mol2` via **File**, **Open** so that the docked poses can be compared to the crystallographic pose (which has its C atoms coloured orange for easier visualisation).



You should see something similar to the above. Two of the solutions are close to the native binding pose.

As can been seen in the `ligand_m1.rnk` file in the output folder, the RMSDs for the top ranked pose in each cluster compared to the native ligand pose in the example above are:

`cluster 1`: 7.287 Å (GoldScore fitness 29.45, ranked 1st)

`cluster 2`: 5.117 Å (GoldScore fitness 26.25, ranked 3rd)

`cluster 3`: 0.803 Å (GoldScore fitness 25.33, ranked 7th)

`cluster 4`: 7.725 Å (GoldScore fitness 24.80, ranked 9th)

`cluster 5`: 6.354 Å (GoldScore fitness 24.78, ranked 10th)

You will observe something similar in the docking you have carried out.

22.7.6 Conclusions

The binding site of 3MTH is large. In addition, there are a relatively small number of donor and/or acceptor points in the active site where a ligand might bind. Furthermore, the co-crystallised ligand (methylparaben insulin) also contains few functional groups. All of these factors mean the docking of the native ligand back into 3MTH is a complex problem for GOLD.

Enabling the diverse solutions feature produces a number of different docked poses, two of which are found to be close to the native ligand pose.

This ends the tutorial.

22.8 Tutorial 8: Running a Covalent Docking

22.8.1 Introduction

First, copy the files in `<Installation folder>/ccdc-software/gold/GOLD/examples/tutorial8` to a directory to which you have write permissions.

The object of this tutorial is to perform a covalent docking using 5L6O, a receptor tyrosine kinase EphB3. These are involved in the regulation of dynamic cellular events such as cell proliferation and migration.

The ligand is an irreversible inhibitor and is bound covalently through a thioether linkage to an active-site cysteine (CYS717), located in the hinge region of the EphB3 kinase domain.

This tutorial will illustrate how to carry out a covalent docking by docking the quinazolin-containing ligand back into its native binding site.

22.8.2 Preparation of Input files

The original PDB file (`5l6o.pdb`) has been provided should you wish to set up the protein and ligand files yourself.

Protein and ligand files are also provided and have been set up in accordance with guidelines for the preparation of input files (Setting Up the Protein(s) and Setting Up Ligands) respectively.

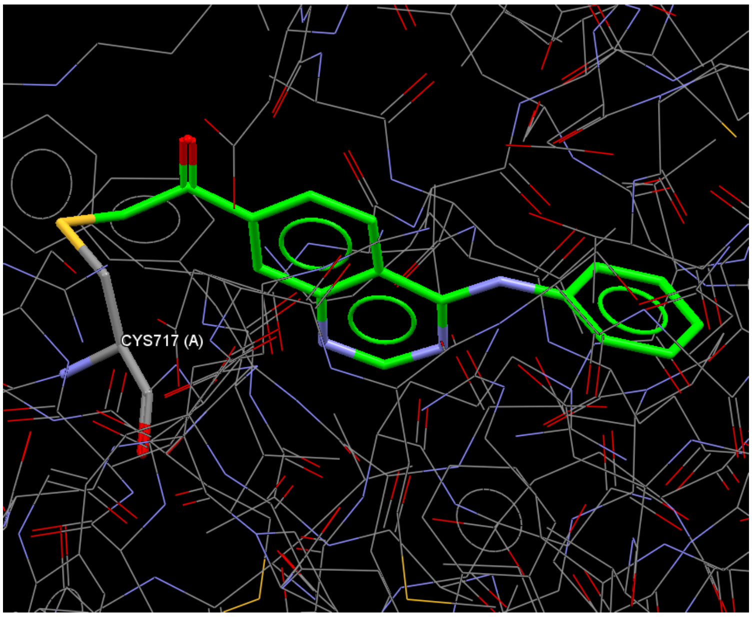
22.8.3 Running a Covalent Docking

GOLD makes a few assumptions when docking covalently (see Method Used for Docking Covalently Bound Ligands):

- It is assumed that there is just one atom linking the ligand to the protein.
- The link atom must be present in both the protein and ligand files.
- Ideally in both files the link atom will have a free valence available through which the link can be made.

It is possible to dock covalently to a single ligand (see Setting Up a Single Covalent Link) or a ligand substructure (see Setting Up Substructure-Based Covalent Links). In this case we are docking only to a single ligand. Note that mol2 files must be used when running covalent dockings.

Load both the protein and ligand files into Hermes. You will see that the ligand is indeed covalently bonded to CYS717. Both moieties have been highlighted in capped sticks in the image below (note that the hydrogen atoms have been hidden for clarity).



If you view the protein and ligand files separately (toggling either off using the **Molecule Explorer**) you will notice the S atom is present in both files. The presence of this atom in both files is extremely important as will be illustrated when we set up the docking later on.

Close the protein and ligand files by clicking on **File** and then **Close All Files** in the Hermes main menu.

22.8.4 Setting up a covalent docking

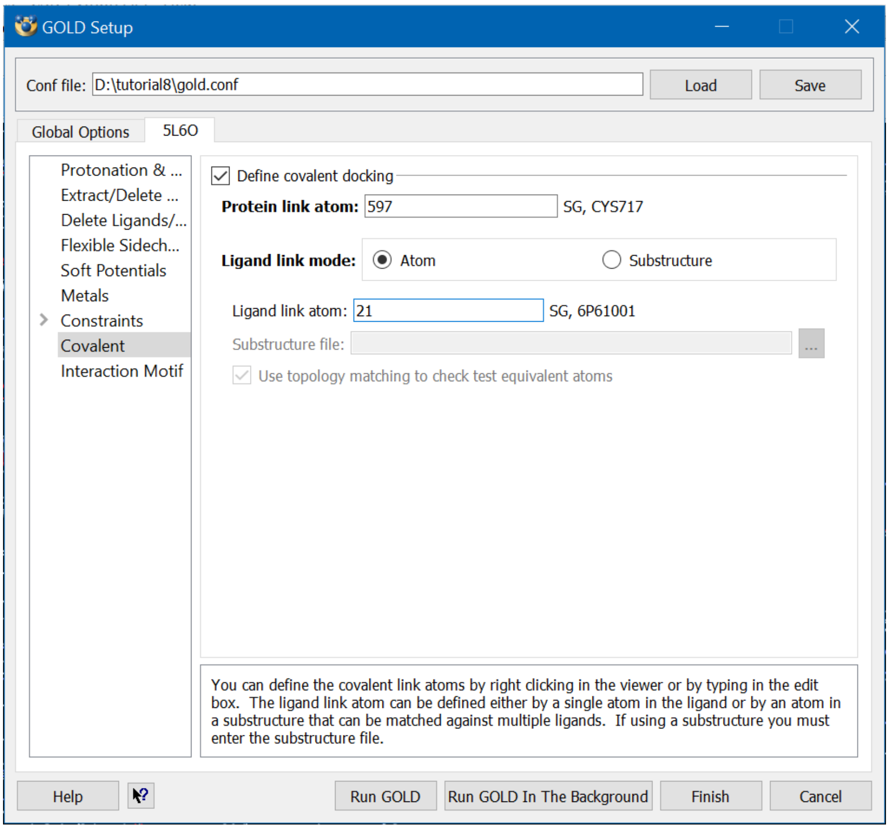
Open Hermes. Load the `gold.conf` provided into GOLD via **GOLD**, **Setup and Run a Docking, Load Existing**, then navigate to the folder containing the tutorial8 files, select `gold.conf` and click **Open**.

Open the `ligand.mol2` file via **File**, **Open**.

The settings for a covalent docking are specific to the protein so click on the **5L6O** tab to access the protein settings, then click on **Covalent**.

Activate the **Define covalent docking** tickbox. As we are docking a single ligand and not a substructure, ensure the **Ligand link mode: Atom** radio button is selected. Now we must either enter the atom IDs manually into the **Protein link atom** and **Ligand link atom** boxes or else select the atoms in the 3D view.

Both the ligand and protein files are open in the 3D view, so return to Hermes and select first the protein link S atom (atom ID 597) and then the ligand link S atom (atom ID 21). Note that you can toggle the ligand/protein on and off via the **Molecule Explorer** to help make the selection of both covalent constraint atoms easier.



Conf file: D:\tutorial8\gold.conf

☑ Define covalent docking

Protein link atom: 597 SG, CYS717

Ligand link mode: ● Atom ○ Substructure

Ligand link atom: 21 SG, 6P61001

Substructure file:

☑ Use topology matching to check test equivalent atoms

You can define the covalent link atoms by right clicking in the viewer or by typing in the edit box. The ligand link atom can be defined either by a single atom in the ligand or by an atom in a substructure that can be matched against multiple ligands. If using a substructure you must enter the substructure file.

Once you have finished defining the covalent atoms, click on the **Global Options** tab to return to the general docking setup.

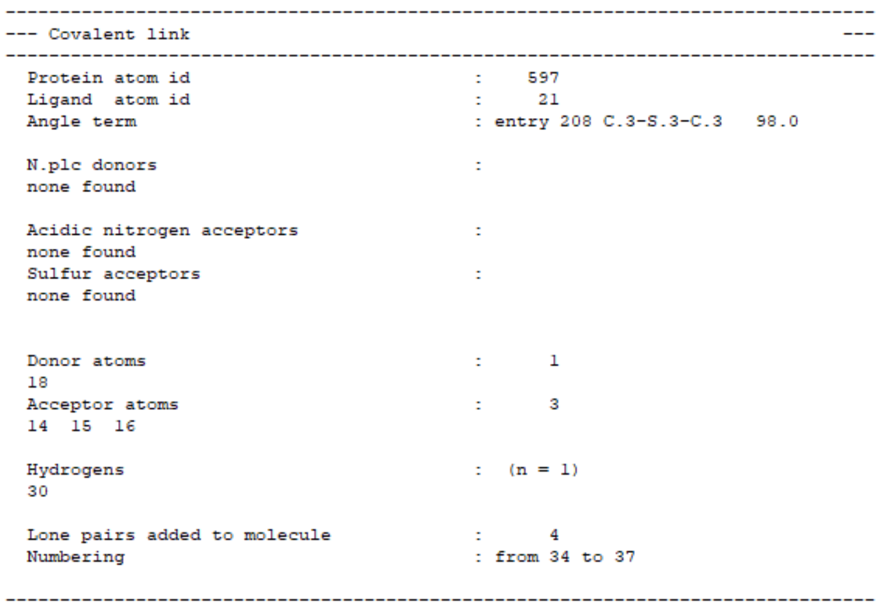
Click on **Output Options** and specify a directory to which you have write permission. This will be where the output files are written.

Click on the **Run GOLD** button. In the **Finish GOLD Configuration** window, you will be prompted that the GOLD configuration has been updated and needs to be saved. Change the configuration file name to e.g. `gold_covalent.conf` then hit **Save** to start the GOLD run.

22.8.5 Analysis of output

Once the docking has completed, click on the `gold_ligand_m1.log` text in the **Run GOLD** window. This will load the ligand.log file into the Run GOLD window, enabling us to view the progress of each genetic algorithm run.

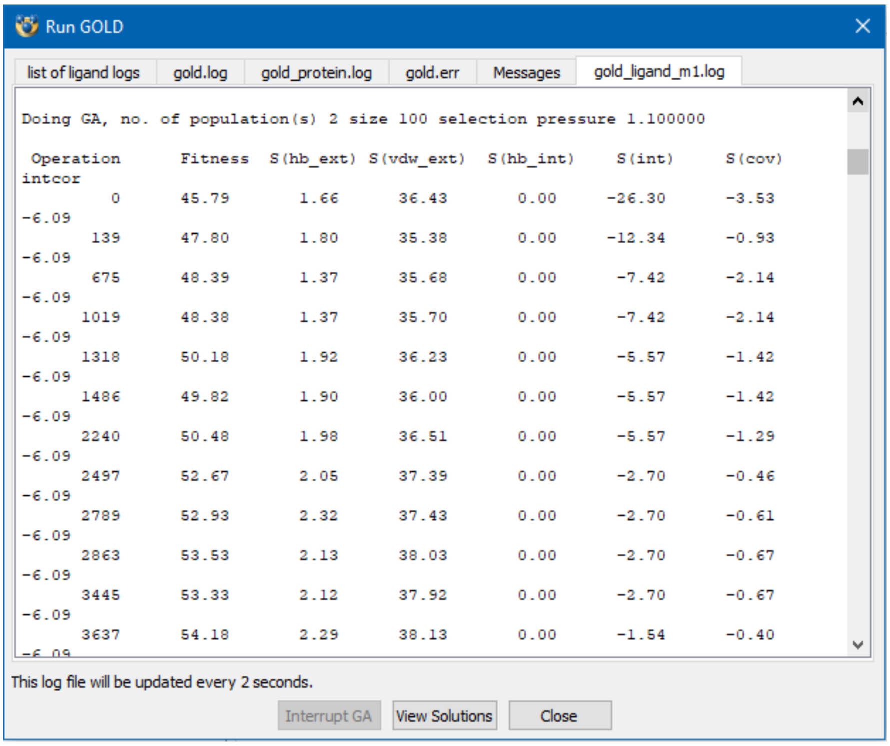
Scroll through this file. You will notice that as the ligand was being initialised, the covalent constraint was analysed:



```
--------------------------------------------------------------------------------
--- Covalent link                                                            ---
--------------------------------------------------------------------------------
  Protein atom id                         :    597
  Ligand  atom id                         :     21
  Angle term                              : entry 208 C.3-S.3-C.3   98.0

  N.plc donors                            :
  none found

  Acidic nitrogen acceptors               :
  none found
  Sulfur acceptors                        :
  none found


  Donor atoms                             :      1
  18
  Acceptor atoms                          :      3
  14  15  16

  Hydrogens                               :  (n = 1)
  30

  Lone pairs added to molecule            :      4
  Numbering                               : from 34 to 37

--------------------------------------------------------------------------------
```

You will notice an `Angle term`: this corresponds to the geometry around the link S atom. During docking GOLD refers to angle and torsion data in its parameter file to ensure the pose(s) adopted by the ligand are chemically sensible. This is also done for the covalent link to ensure the geometry around the link is sensible. The `entry 208` text corresponds to the line number of the angle entry in the `BOND_ANGLE_TABLE` in the `gold.params` file.

If you scroll down further you will notice that during the algorithm run, the fitness score is broken down into its constituent parts, specifically `S(hb_ext)`, `S(vdw_ext)`, `S(hb_int)`, `S(int)`. In addition to these default scoring terms there is an additional term, `S(cov)`, that is added only when docking covalently. Within the `BOND_ANGLE_TABLE` in the `gold.params` there are also energy terms and the `S(cov)` contribution to the score is calculated from this.



```
Doing GA, no. of population(s) 2 size 100 selection pressure 1.100000

 Operation      Fitness  S(hb_ext) S(vdw_ext)  S(hb_int)     S(int)      S(cov)
intcor
        0       45.79       1.66       36.43       0.00     -26.30      -3.53
-6.09
      139       47.80       1.80       35.38       0.00     -12.34      -0.93
-6.09
      675       48.39       1.37       35.68       0.00      -7.42      -2.14
-6.09
     1019       48.38       1.37       35.70       0.00      -7.42      -2.14
-6.09
     1318       50.18       1.92       36.23       0.00      -5.57      -1.42
-6.09
     1486       49.82       1.90       36.00       0.00      -5.57      -1.42
-6.09
     2240       50.48       1.98       36.51       0.00      -5.57      -1.29
-6.09
     2497       52.67       2.05       37.39       0.00      -2.70      -0.46
-6.09
     2789       52.93       2.32       37.43       0.00      -2.70      -0.61
-6.09
     2863       53.53       2.13       38.03       0.00      -2.70      -0.67
-6.09
     3445       53.33       2.12       37.92       0.00      -2.70      -0.67
-6.09
     3637       54.18       2.29       38.13       0.00      -1.54      -0.40
-6.09
```

This log file will be updated every 2 seconds.

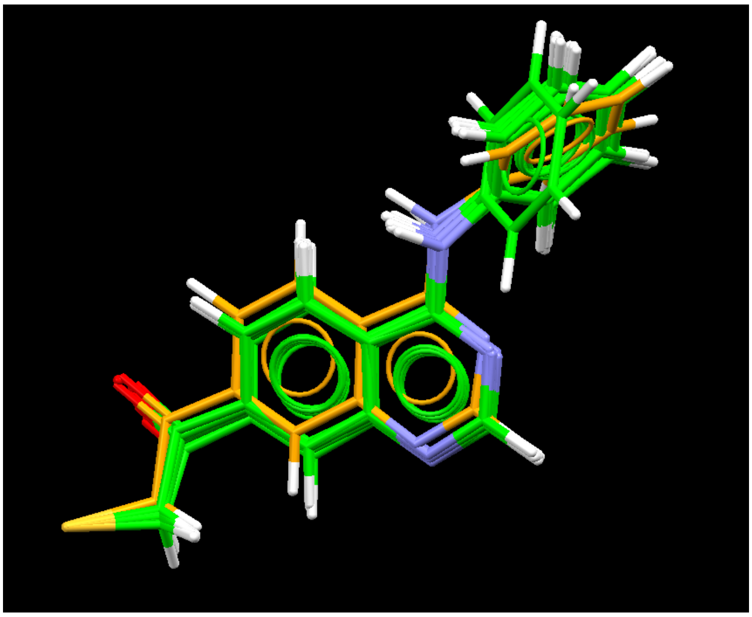
Click on the **View Solutions** button in the **Run GOLD** window to load the docking results into the 3D view. Then hit **Close** to close the window.

Scroll through the docking results. You will notice the ligand is indeed bound to the protein.

The docked poses can be compared to the pose of the native ligand structure by superimposing the docking solutions with the `ligand.mol2` file. Load this file in Hermes via **File**, **Open**.

In the **Docking Solutions** tab of the **Molecule Explorer**, select all docking solutions. Then select the **Display** tab, ensure that the native ligand is in view with the **5l6o_ligand** tickbox active, right-click on this entry labelled **5l6o_ligand** and colour the C atoms in orange.

To make the display clearer, you can activate in the **Molecule Explorer** only the tickbox corresponding to the native ligand and that to the docked solutions.



This ends the tutorial.

23 Appendix B: List of Atom and Bond Types

GOLD uses SYBYL atom and bond types as follows:

23.1 Atom types

Hydrogen	H
Carbon sp^3	C.3
Carbon sp^2	C.2
Carbon sp	C.1
Carbon aromatic	C.ar
Carbocation (guanadinium)	C.cat
Nitrogen sp^3	N.3
Nitrogen sp^2	N.2
Nitrogen sp	N.1
Nitrogen aromatic, e.g. in pyridine	N.ar
Nitrogen amide	N.am
Nitrogen trigonal planar, e.g. in nitro, pyrrole	N.pl3
	N.4

Nitrogen sp^3 positively charged, e.g. in lysine	
Oxygen sp^3	O.3
Oxygen sp^2	O.2
Oxygen in carboxylates and phosphates	O.co2
Sulfur sp^3	S.3
Sulfur sp^2	S.2
Sulfoxide sulfur	S.o
Sulfone sulfur	S.o2
Phosphorus sp^3	P.3
Halogens, metals	normal element symbols, e.g. F, Cl, Ca, Zn

23.2 Bond types

Single	1
Double	2
Triple	3
Aromatic	ar
Amide	am
Delocalised, e.g. in carboxylate, guanidinium	ar

24 Appendix C: Additional Tags in Output Files

Solution output files for the docked ligand(s) can contain additional information such as the scoring function terms and the rotated protein hydrogen atom positions that were generated during the docking.

This information can be written to sd file tags; for mol2 files, these tags are written to comment blocks. For mmCIF files they are written into the _ccdc.docking_solution table. This additional information is particularly important when post-processing docking results. It is possible to control the information written to solution files (see Controlling the Information Written to Ligand Solution Files).

The table below lists the tag names that you are likely to see in GOLD solution files.

Name	Explanation	See
`Gold.Protein.ActiveResidues`	List of protein residues used to define the binding site	(see Defining a Binding Site from a List of Atoms or Residues)
`Gold.Protein.RotatedAtoms`	Optimised positions of polar protein hydrogen atoms that are generated during docking	(see File Containing the Protein Binding-Site Geometry)
`Gold.Protein.RotatedWaterAtoms`	Optimised positions of water hydrogen atoms generated during docking	(see Water Molecules)
`Gold.Protein.RotatedTorsions`	Optimised torsions for rotatable bonds in the ligand. Also for protein side chain torsions which have been specified as being allowed to rotate during docking	(see Side Chain Flexibility)
`Gold.Id.Protein`	Enabling the association of a solution with its protein	
`Gold.Id.Ligand`	Ligand identifier	
`Gold.Rescore.Rmsd`	RMSD of rescored solutions	(see Rescoring)
Scoring terms		
`Gold.Fitness.Score`	Total fitness value of docked ligand	
`Gold.Covalent.Energy`	Covalent bonding contribution to the fitness score	
`Gold.Constraint.Score`	Constraint contribution to the fitness score	

Name	Explanation	See
`Ligand.Score.Contributions`	For individual ligand atom: its scoring contribution to the total fitness score and also the constituent scoring terms	(see Controlling the Information Written to Ligand Solution Files)
`Protein.Score.Contributions`	For individual protein atom: its scoring contribution to the total fitness score and also the constituent scoring terms	(see Controlling the Information Written to Ligand Solution Files)
`Gold.Ensemble.ID`	When docking into protein ensembles, this is a numerical identifier given to each initialised protein. The Gold.Ensemble.ID corresponds to the number in the output protein file, i.e. gold_protein_1.mol2, gold_protein_2.mol2	(see Interpreting Ensemble Docking Output)
[GoldScore		
`Gold.Goldscore.Fitness`	Total GoldScore fitness value of docked ligand	(see GoldScore)
`Gold.Goldscore.External.Hbond`	Protein-ligand H-bond contribution to GoldScore value	(see GoldScore)
`Gold.Goldscore.External.Vdw`	Protein-ligand vdW contribution to GoldScore value	(see GoldScore)
`Gold.Goldscore.Internal.Hbond`	Internal ligand intramolecular H-bond contribution to GoldScore value	(see GoldScore)
`Gold.Goldscore.Internal.Vdw`		(see GoldScore)

Name	Explanation	See
	Internal ligand vdW contribution to GoldScore value	
`Gold.Goldscore.Internal.Torsion`	Internal ligand torsion-strain contribution to GoldScore value	(see GoldScore)
`Gold.Goldscore.Covalent.Energy`	Covalent bonding contribution to GoldScore value	(see GoldScore)
`Gold.Goldscore.Constraint.Score`	Constraint contribution to GoldScore value	(see GoldScore)
`Gold.Goldscore.Internal.Correction`	Internal ligand energy offset	(see Internal Energy Offset)
`Gold.Goldscore.Protein.Energy`	Protein energy term to penalise clashes when using flexible side chains	(see Protein-Protein Clashes)
`Gold.Goldscore.Sbar`	Penalty term for non-displacement of active site waters	(see Water Molecules)
`Gold.Goldscore.Reference.RMSD`	RMSD of solution against reference ligand	(see Specifying a Ligand Reference File)
ChemScore		
`Gold.Chemscore.ZeroCoef`	The ChemScore zero coefficient	(see Overview)
`Gold.Chemscore.Rot`	Rotatable-bond freezing term contribution to ChemScore value	(see Rotatable-Bond Freezing Term)
`Gold.Chemscore.Fitness`	Total ChemScore fitness value of docked ligand	(see Overview)
`Gold.Chemscore.Hbond`	Protein-ligand H-bond contribution to ChemScore value	(see Hydrogen-Bond Terms)

Name	Explanation	See
`Gold.Chemscore.Lipo`	Protein-ligand lipophilic contribution to the ChemScore value	(see Metal-Binding and Lipophilic Terms)
`Gold.Chemscore.Metal`	Metal-binding contribution to ChemScore value	(see Metal-Binding and Lipophilic Terms)
`Gold.Chemscore.internal_Hbond`	Internal ligand intramolecular H-bond contribution to ChemScore value	(see Hydrogen-Bond Terms)
`Gold.Chemscore.DEClash`	Protein-ligand clash penalty to the ChemScore value	(see Clash Penalty and Internal Torsion Terms)
`Gold.Chemscore.DEInternal`	Internal ligand torsional strain penalty to the ChemScore value	(see Clash Penalty and Internal Torsion Terms)
`Gold.Chemscore.DG`	Free energy change (that occurs on ligand binding) contribution to ChemScore value	(see Overview)
`Gold.Chemscore.Covalent`	Covalent bonding contribution to ChemScore value	(see Covalent Term)
`Gold.Chemscore.Constraint`	Constraint contribution to ChemScore value	(see Constraint Terms)
`Gold.Chemscore.CHOScore`	Contribution for weak CH...O H-bonds	(see Kinase Scoring Function)
`Gold.Chemscore.Internal.Correction`	Internal ligand energy offset	(see Internal Energy Offset)
`Gold.Chemscore.Protein.Energy`	Protein energy term to penalise clashes	(see Protein-

Name	Explanation	See
	when using flexible side chains	Protein Clashes)
Gold.Chemscore.Sbar	Penalty term for non-displacement of active site waters	(see Water Molecules)
Gold.Chemscore.Reference.RMSD	RMSD of solution against reference ligand	(see Specifying a Ligand Reference File)
Astex Statistical Potential (ASP)		
Gold.ASP.Fitness	Total ASP fitness value of docked ligand	(see Astex Statistical Potential (ASP))
Gold.ASP.ASP	Calculated statistical potential plus the ChemScore clash term and internal energy term	(see Astex Statistical Potential (ASP))
Gold.ASP.Map	The total calculated statistical potential is a summation over all combinations of protein and ligand atoms	(see Astex Statistical Potential (ASP))
Gold.ASP.Hbond	Protein-ligand H-bond contribution to ASP value	(see Astex Statistical Potential (ASP))
Gold.ASP.Metal	Metal-binding contribution to ASP value	(see Astex Statistical Potential (ASP))
Gold.ASP.DEClash	Protein-ligand clash penalty to the ASP value	(see Astex Statistical Potential (ASP))
Gold.ASP.DEInternal	Internal ligand intramolecular H-bond contribution to ASP value	(see Astex Statistical Potential (ASP))
Gold.ASP.Rot	Rotatable-bond freezing term	(see Astex Statistical

Name	Explanation	See
	contribution to ASP value	Potential (ASP))
`Gold.ASP.Covalent`	Covalent bonding contribution to ASP value	(see Covalent Docking and Docking with Constraints)
`Gold.ASP.Constraint`	Constraint contribution to ASP value	(see Covalent Docking and Docking with Constraints)
`Gold.ASP.Protein.Energy`	Protein energy term to penalise clashes when using flexible side chains	(see Protein-Protein Clashes)
`Gold.ASP.Sbar`	Penalty term for non-displacement of active site waters	(see Water Molecules)
`Gold.ASP.Internal.Correction`	Internal ligand energy offset	(see Internal Energy Offset)
`Gold.ASP.Reference.RMSD`	RMSD of solution against reference ligand	(see Specifying a Ligand Reference File)
Piecewise Linear Potential (ChemPLP)		
`Gold.PLP.Fitness`	Total ChemPLP fitness value of docked ligand	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.PLP`	Calculated potentials plus the ChemScore clash term and internal energy term	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.part.hbond`		

Name	Explanation	See
	Protein-ligand H-bond contribution to PLP value	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.part.metal`	Metal-binding contribution to PLP value	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.part.buried`	Scoring contribution from buried interaction types	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.part.nonpolar`	Scoring contribution from nonpolar interaction types	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.part.repulsive`	Scoring contribution from repulsive interaction types	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.ligand.clash`	Protein-ligand clash penalty to the PLP value	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.ligand.torsion`	Internal ligand torsional strain penalty to the PLP value	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.Chemscore.hbond`	ChemScore Protein-ligand H-bond contribution	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.Chemscore.CHOscore`	Contribution for weak CH...O H-bonds	(see Piecewise Linear Potential (ChemPLP))

Name	Explanation	See
`Gold.PLP.Chemscore.metal`	ChemScore Metal-binding contribution	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.Goldscore.Hbond`	GoldScore Protein-ligand H-bond contribution	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.DEclash`	Protein-ligand clash penalty to the PLP value	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.Chemscore.protein.energy`	Protein energy term to penalise clashes when using flexible side chains	(see Protein-Protein Clashe)
`Gold.PLP.Sbar`	Penalty term for non-displacement of active site waters	(see Water Molecules)
`Gold.PLP.Chemscore.internal.correction`	Internal ligand energy offset	(see Internal Energy Offset)
`Gold.PLP.Chemscore.covalent`	Covalent bonding contribution to PLP value	(see Piecewise Linear Potential (ChemPLP))
`Gold.PLP.constraint`	Constraint contribution to PLP value	(see Piecewise Linear Potential (ChemPLP))

Certain docking-score terms are the product of a term dependent on the magnitude of a particular physical contribution (e.g. hydrogen bonding) and a scale factor determined e.g. by a regression coefficient.

The docking-score term descriptors included in the output file can therefore consist of weighted terms, non-weighted terms or both (see Controlling the Information Written to Ligand Solution Files).

Weighted terms will be indicated as such in the tag name, e.g. `Gold.Chemscore.Hbond.Weighted`.

25 Appendix D: Genetic Algorithm Parameter Definitions

Changes to genetic algorithm parameters should be made with care (see Controlling Accuracy and Speed with Genetic Algorithm Parameter Settings).

25.1 Population Size

The genetic algorithm maintains a set of possible solutions to the problem. Each possible solution is known as a chromosome and the set of solutions is termed a population.

The variable `Population Size` (or `popsize`) is the number of chromosomes in the population. If `n_islands`is greater than one (i.e. the genetic algorithm is split over two or more islands), `popsize` is the population on each island.

25.2 Selection Pressure

Each of the genetic operations (crossover, migration, mutation) (see Operator Weights: Migrate, Mutate, Crossover) takes information from parent chromosomes and assembles this information in child chromosomes. The child chromosomes then replace the worst members of the population.

The selection of parent chromosomes is biased towards those of high fitness, i.e. a fit chromosome is more likely to be a parent than an unfit one.

The selection pressure is defined as the ratio between the probability that the most fit member of the population is selected as a parent to the probability that an average member is selected as a parent. Too high a selection pressure will result in the population converging too early.

For the GOLD docking algorithm, a selection pressure of 1.1 seems appropriate, although 1.125 may be better for library screening where the aim is faster convergence.

25.3 Number of Operations

The genetic algorithm starts off with a random population (each value in every chromosome is set to a random number). Genetic operations (crossover, migration, mutation) (see Operator Weights: Migrate, Mutate, Crossover) are then applied iteratively to the population. The parameter `Number of Operations` (or `maxops`) is the number of operators that are applied over the course of a GA run.

It is the key parameter in determining how long a GOLD run will take.

25.4 Number of Islands

Rather than maintaining a single population, the genetic algorithm can maintain a number of populations that are arranged as a ring of islands. Specifically, the algorithm maintains `n_islands` populations, each of size `popsize`.

Individuals can migrate between adjacent islands using the migration operator.

The effect of `n_islands` on the efficiency of the genetic algorithm is uncertain.

25.5 Niche Size

Niching is a common technique used in genetic algorithms to preserve diversity within the population.

In GOLD, two individuals share the same niche if the RMSD between the coordinates of their donor and acceptor atoms is less than 1.0 Å.

When adding a new individual to the population, a count is made of the number of individuals in the population that inhabit the same niche as the new chromosome. If there are more than `NicheSize` individuals in the niche, then the new individual replaces the worst member of the niche rather than the worst member of the total population.

25.6 Operator Weights: Migrate, Mutate, Crossover

The operator weights are the parameters `Mutate`, `Migrate` and `Crossover` (or `pt_cross`).

They govern the relative frequencies of the three types of operations that can occur during a genetic optimisation: point mutation of the chromosome, migration of a population member from one island to another, and crossover (sexual mating) of two chromosomes.

Each time the genetic algorithm selects an operator, it does so at random. Any bias in this choice is determined by the operator weights. For example, if `Mutate` is 40 and `Crossover` is 10 then, on average, four mutations will be applied for every crossover.

The migrate weight should be zero if there is only one island; otherwise, migration should occur about 5% of the time.

26 Appendix E: The Torsion Angle Distribution File

26.1 Format of Torsion Angle Distribution File Header

The first section of the torsion angle distribution file sets parameters and tells GOLD what to do with the distributions.

`N_BINS` is the number of bins used in the torsion histogram.

`REMOVE_HIGH_ENERGY` and `DELTA_E` are parameters that can be used to control the filtering out of high-energy torsion angles.

If torsion angle distributions are used, GOLD will no longer sample over 360 degrees but will constrain the torsion to values contained in the histogram. However, if a histogram contains a large number of entries, there may be some high-energy torsions within the histogram. GOLD therefore provides a method for filtering out such high-energy torsions: set `REMOVE_HIGH_ENERGY = 1` and `DELTA_E = E` to remove those bars in the histogram that correspond to torsions that are `E` kcal/mol higher in energy than the most populated state. The ground state of the torsion is assumed to correspond to the maximum peak in the torsional histogram. The energy difference between this ground state and any other peak in the torsion angle histogram is then assumed to be approximately given by the partition function.

The following table indicates the relationship between the value of `DELTA_E` and the ratio `high/low`, where `high` is the height of the biggest bar in the histogram and `low` is the height below which bars will be removed from the histogram:

DELTA_E	Ratio
3.0	161
2.5	69
2.0	30

For example, if `REMOVE_HIGH_ENERGY=1` and `DELTA_E = 2.5`, those bars which are 1/69th or less of the height of the largest bar will be removed from the histogram and torsion angles corresponding to these bars will never be sampled by the genetic algorithm.

The relationship between `DELTA_E` and `ratio`, based on the partition function, is: ratio = exp (DELTA_E/0.5898)

An alternative method for pruning torsion angle distributions also exists. `REMOVE_LOW_AREA_BINS` and `MIN_RELATIVE_AREA` can be used to remove bins in the distributions that have a low relative area (where the area of a bin is bin count x bin width). So, for example, if `MIN_RELATIVE_AREA` is set to 2.0, and a distribution was built from 200 observations, any bin with less than 4 counts would be excluded from the search space.

When using this method additional heuristics are also applied. If the total number of observations is < 100 in the distribution, the value of `MIN_RELATIVE_AREA` is reduced for the distribution, meaning that sparse distributions are pruned less. Further, if the consequence of a cutoff would be that all bins would be removed, the relative areas is reduced so that only the 1/2 of the distribution is retained in the sampling.

26.2 Format of Torsion Angle Distributions

Each torsion angle distribution entry comprises three lines: the first line is the name of the torsion angle; the second line is the definition of the torsion angle; the third line is the histogram.

The histogram should be a list of space-separated integers. The i^{th} integer should be the number of observations in the torsion-angle range of the i^{th} bin. There should be `N_BINS` integers in all. The first bin starts at -180 degrees and the last bin ends at +180 degrees.

Torsion angle distributions are defined using Backus-Naur Form (BNF) grammar, as follows (all the symbols in the table are part of the grammar except for ||, which is used to indicate alternative fields):

Torsion Angle	**Backus-Naur Form (BNF) Grammar**
TORSION	NODE | NODE | NODE | NODE | || NODE | NODE | NODE | NODE | DIRECTIVE || NODE | NODE | NODE | NODE | DIRECTIVE | DIRECTIVE
DIRECTIVE	expand <min> <max> || period <min> <max>
NODE	ATOM || ATOM (NEIGHBOURS)
NEIGHBOURS	NEIGHBOUR_NODE || NEIGHBOUR_NODE NEIGHBOURS
NEIGHBOUR_NODE	NODE || HYDROGENS
HYDROGENS	0H || 1H || 2H || 3H
ATOM	ATOM_DEF || ATOM_DEF [FRAGMENT]
FRAGMENT	ribose || adenine || uracil || benzene
ATOM_DEF	TYPE_DEF || LINKAGE&ltno space&gtTYPE_DEF
TYPE_DEF	SYB_TYPE || EL_TYPE
LINKAGE	~ || = || -
SYB_TYPE	C.3 || C.2 || C.1 || C.ar || C.cat || N.3 || N.2 || N. 1 || N.ar || N.am || N.pl3 || N.4 || O.3 || O.2 || O.co2 || S.3 || S.2 || S.o || S.o2 || P.3 || H || F || Cl || Br || I
EL_TYPE	C || N || O || S || P

This grammar allows torsions to be specified as four fragment nodes. Each node defines an atom type and, optionally, a set of neighbours to which the atom is connected. Each of the neighbours is a node or an exact count of the number of hydrogen atoms to which the atom is bonded. Atom types are defined using SYBYL atom types or elemental atom types. The atom can also be required to be part of a pre-defined fragment.

Bonding environments can also be specified, using the symbols ~,=,-, which indicate, respectively, that an atom forms an aromatic, double or single bond to its parent node. Note: ~,=, and - should therefore not be used on the first atoms specified, these bond types are specified for substituents only.

A node is a parent of all its neighbours and a top-level node in the torsion definition is a parent of subsequent nodes in the torsion.

There are currently four fragments available, one of which (the uracil fragment) matches both thymine and uracil. More fragments can easily be added. The Ullman algorithm is used to determine if an atom belongs to a fragment. Fragments are defined through SYBYL atom types and connectivity (exact bond types are not used). Only heavy atoms are considered. Currently, fragments are precompiled, but they could be read in at run-time if required.

As of the 2023.3 Release of GOLD, an alternative SMARTS based approach has been supported. For example, a torsion pattern could be generated using a keyword such as below:

```
SMARTS [CH2:1][C:2](=O)!@[OX2:3][CH2:4]
```

This would express a four atom torsion pattern. The labels 1,2,3 and 4 mark the 4 reference nodes of the torsion. SMARTS are less easily readable than the Backus-Naur Form (BNF) Grammar, but are far more widely supported and more easily suited for use for generation of patterns by automated processes using standard cheminformatic toolkits.

Note that the format of SMARTS patterns used for CCDC products is documented in the CSD Python API documentation.

Directives are allowed to take account of special circumstances. There are two directives: `expand` and `period`.

The `expand` directive has the form `expand <min> <max>` where `<max> - <min> = 180.0` or `<min> = 0`. This directive is used for torsions where the CSD query has symmetry and torsions are only measured over `<min>` to `<max>` degrees. However, although the CSD query may have two-fold symmetry, often the matched structure does not. The `expand` directive fills out the rest of the histogram with the correct values.

The `period` directive takes account of those torsional distributions for which the matched structure has symmetry. This directive has the form `period <pmin> <pmax>`. The distribution will only be expanded between angles `<pmin>` and `<pmax>`.

26.3 Example Torsion Angle Distributions

Here are some examples of torsion angle distributions extracted from the Cambridge Structural Database and in the correct format:

```
acid T1
C.2 (O.co2 O.co2) | C.3 (2H) | C.3 (2H) | C
41 8 0 0 0 0 0 0 0 0 1 8 7 2 0 0 0 0 1 1 0 0 0 1 0 4 1 0 1 0 0 0 0 0
2 2 41

acid T2
O.co2 | C.2 (O.co2) | C.3 (2H) | C.3 (2H C)
8 5 1 3 2 1 3 2 3 2 3 3 4 0 3 2 7 11 15 9 1 4 1 0 2 1 4 4 1 3 3 6
0 3 5 7

amide nh T2
C.2 (=O.2 N.am (1H)) | C.3 (1H C.3) | N.am (1H) | C.2 (=O.2)
1 1 14 16 29 25 23 38 35 50 82 156 53 6 1 0 0 0 0 0 0 1 1 14 17 15
4 4 2 1 2 5 2 2 0 0

uracil
O.3 [ribose] | C.3 [ribose] | N.am [uracil] (C.2 (1H))| C.2
[uracil] (=O.2)
24 73 85 44 59 60 40 14 8 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 7 5 3 0 0
1 4 3 3 5 10 6

benzyl sub
C | C.3 (2H) | C.ar (~C.ar (0H)) | ~C.ar (0H) | expand 0.0 180.0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 9 27 76 64 15 7 4
2 0 0 0 0

CCH2C(CH3)(CH3)C
SMARTS [#6:1][#6X4H2:2]!@[#6X4:3]([#6H3])([#6H3])[#1:4] | weight
100
33 9 1 1 2 0 1 4 10 57 128 899 1784 750 137 50 32 14 14 32 50 137
750 1784 899 128 57 10 4 1 0 2 1 1 9 33
```

26.4 Creating your own patterns

As of GOLD Release 2023.3 it is possible for CSD Python API users to use a script to create their own patterns. These can then be included into your own torsion distribution file.

The API script `create_pattern_file.py` in the `create_gold_torsion_patterns` utility allows creation of torsion distributions from the CSD. See the CSD Python API Utilities documentation for more information.