General Properties

ID	Weight			Rot. bonds
12	423 g/mol			4
363	423 g/mol			6

```
##################################################
    Step 01: SELECT MOLECULES FROM DATA TABLE
##################################################
```

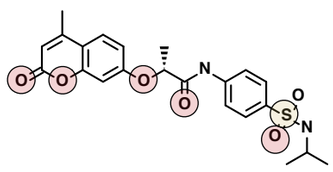
input mol2 directory + datatable path → 01_selected-mol2s

Given a directory of mol2 files
and a data table with general information
about molecules, select a subset of molecules that have fewer than 8 rotatable bonds and are heavier than 200 g/mol.

This step uses a column filter that can be customized
in the config file:

```
column filter: (NRB <= 7) & (MWT > 200)
```

Atom Type Filter



```
##################################################
Step 02: PREFILTER BY FUNCTIONAL GROUP PRESENCE
##################################################
```

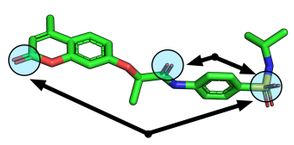
01_selected-mol2s → 02_fgroup-presence_mol2s

Using the subset of previously selected
MOL2 structures, this step will select only those molecules
that satisfy the following 2 criteria:
1) Contain a S.3 or S.o2 sulfur atom (sp3-hybridized sulfur)
2) Contain a O.2 atom (sp2-hybridized oxygen)

This is accomplished by the following
`functional group presence filter settings`
in the configuration file:

```
((atom_type == 'S.3') | (atom_type == 'S.o2'))
--> (atom_type == 'O.2')
```

Functional Group Distances



```
##################################################
Step 03: PREFILTER BY FUNCTIONAL GROUP DISTANCE
##################################################
```

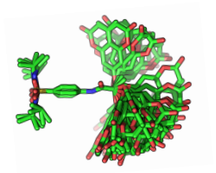
02_fgroup-presence_mol2s → 03_fgroup_distance_mol2s

Now, based on the subset of molecules that
contain at least an O.2 atom and a S.3 or S.o2
atom, this pipeline step selects only those molecules where
both an oxygen and a sulfur atom are within a specified
distance with each other (here: 13-20 angstroms).

This is accomplished by the following
`functional group distance filter settings`
in the configuration file:

```
selection key: ((atom_type == 'S.3') | (atom_type == 'S.o2'))
                --> (atom_type == 'O.2')
distance: 13-20
```

Database Conformers

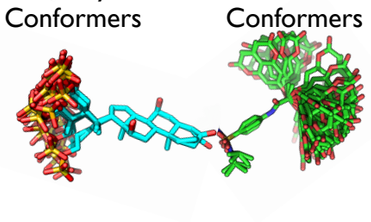


```
##################################################
          Step 04: OMEGA conformers
##################################################
```

03_fgroup_distance_mol2s → 04_omega_conformers

This step utilizes OpenEye OMEGA to
generate multiple low-energy conformations (here: up to 200)
of the selected database molecules. These database molecules will then
be overlaid with conformers of the query molecule in the next step.

Query Conformers / Database Conformers



```
##################################################
          Step 05: ROCS OVERLAYS
##################################################
```

04_omega_conformers + 3kpzs_conf_subset_nowarts.mol2 → 05_rocs_overlays

This pipeline step assumes that you have already a MOL2 file with low-energy conformers of the query molecule availlable. The conformers in this input data file (here: `3kpzs_conf_subset_nowarts.mol2`) are then overlaid with the database conformers that were created in the previous step. In the previous step, we used a wrapper function within the screenlamp toolkit to generate the database molecule conformers for multiple partitions, but you can use a tool like OpenEye OMEGA directly to generate such conformers without using screenlamp.

Also note that performing ROCS overlays using multi-conformer query molecule input is recommended but computationally more intensive than using a single-conformer query. For instance, by using a 20-conformer input query, the ROCS overlay step will take approximately 20x longer compared to a single-conformer query.

Database-Query Conformer Overlays

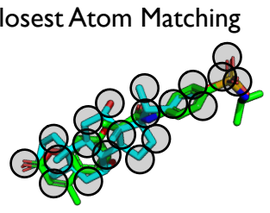

```
##################################################
          Step 06: SORT ROCS OVERLAYS
##################################################
```

05_rocs_overlays → 06_rocs_overlays_sorted

In this step are processed for the further pipeline steps. That is,
a separate file for database molecules and query molecules — with
respect to the best-scoring match — will be generated. In addition, the molecules
are sorted by one or more ROCS scoring metric, and a minimum-score threshold
can be set. The sort columns and score threshold can be specified in the
configuration file in the `ROCS settings` section:

```
ROCS results sort by: TanimotoCombo,ColorTanimoto
ROCS score threshold: (TanimotoCombo >= 0.75) &
                      (ColorTanimoto >= 0.1)
```

Closest Atom Matching



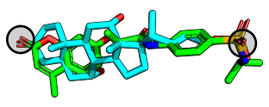
```
##################################################
      Step 07: MATCHING FUNCTIONAL GROUPS
##################################################
```

06_rocs_overlays_sorted → 07_funcgroup_matching

After the sorted database-query molecule overlays were generated
in the previous pipeline step, the data frames with functional group
matching information can be generated. More specifically, this pipeline step will
generate tables that list the closest (in terms of distance) atom match for each
query atom. By default, a pair of atoms within 1.3 angstroms is considered an
overlay match. This maximum-distance threshold can be changed in the
configuration file in the following section:

```
functional group matching selection settings:
  maximum pairwise atom distance: 1.3 # in angstrom
```

Functional Group Matching Selection



```
##################################################
   Step 08: SELECTING FUNCTIONAL GROUP MATCHES
##################################################
```

07_funcgroup_matching → 08_funcgroup_selection

Based on the data tables generated in the previous pipeline step,
this final step selects a subset of molecules that have a atom or
functional group matching pattern of interest. Using the unmodified
default settings in the configuration file, we are selecting only those molecules
that meet the following two criteria:

a) Have a keto-group, with negative partial charge, that overlays with the keto-group
in the query molecule.
b) Have a sulfate atom (positively charged) that overlays with the sulfur atom
in the query molecule.

```
functional group match selection settings:

  atomtype selection keys: ((S1 == 'S.3') | (S1 == 'S.o2'))
                           --> (O2 == 'O.2')
  charge selection keys: ((S1 >= 1.0)) --> (O2 <= -0.5)
```