Vernalis
RESEARCH
part of Vernalis plc

RDKit UGM 2015

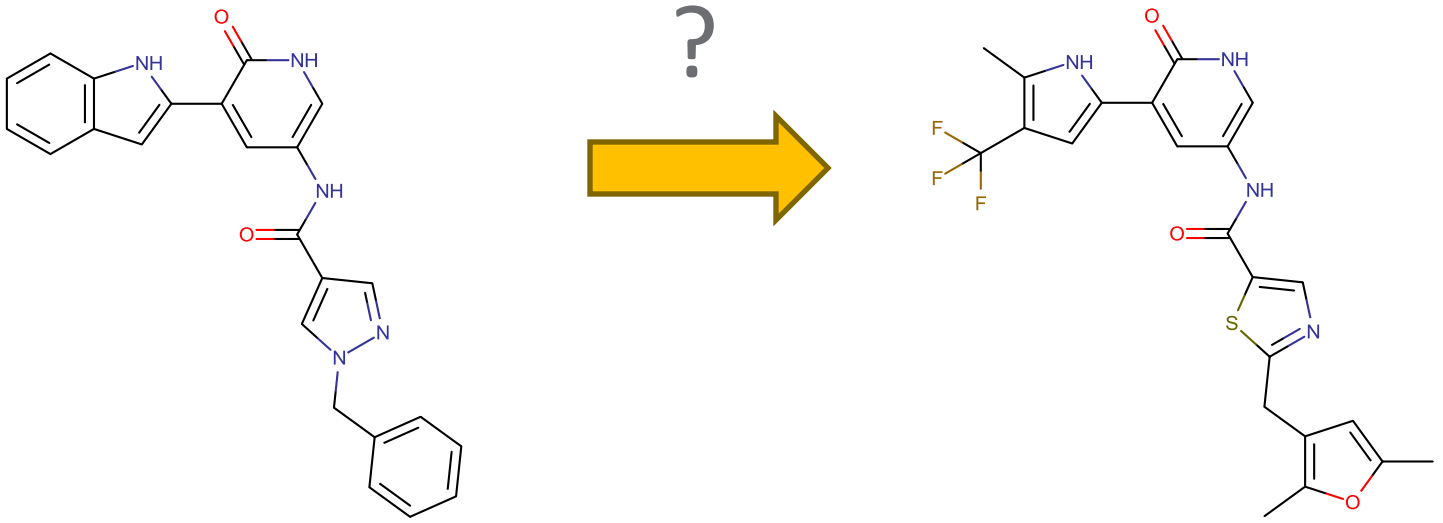
Aligning ideas to bound ligands in PyMOL

THE QUEEN'S AWARDS
FOR ENTERPRISE:
INTERNATIONAL TRADE
2014

Helping chemists be (even more) creative

Vernalis RESEARCH part of Vernalis plc

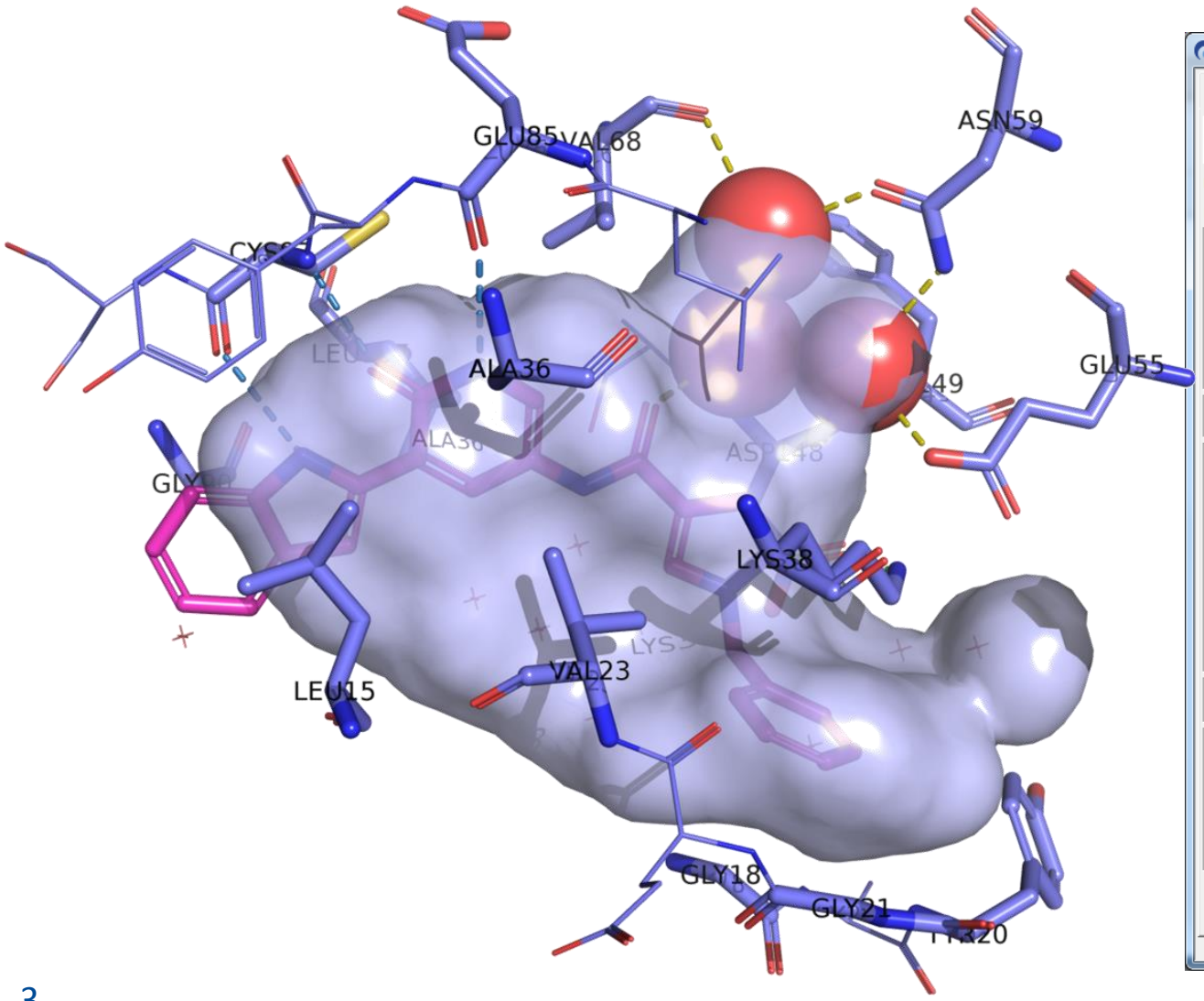
- Goal – all our chemists should be able to:
  - Explore details of **ligand-bound crystal structures**
  - Assess and prioritise their **own** structure-based hypotheses
  - **Share ideas** with expert computational chemists for higher-level calculations



Automatically annotate binding sites

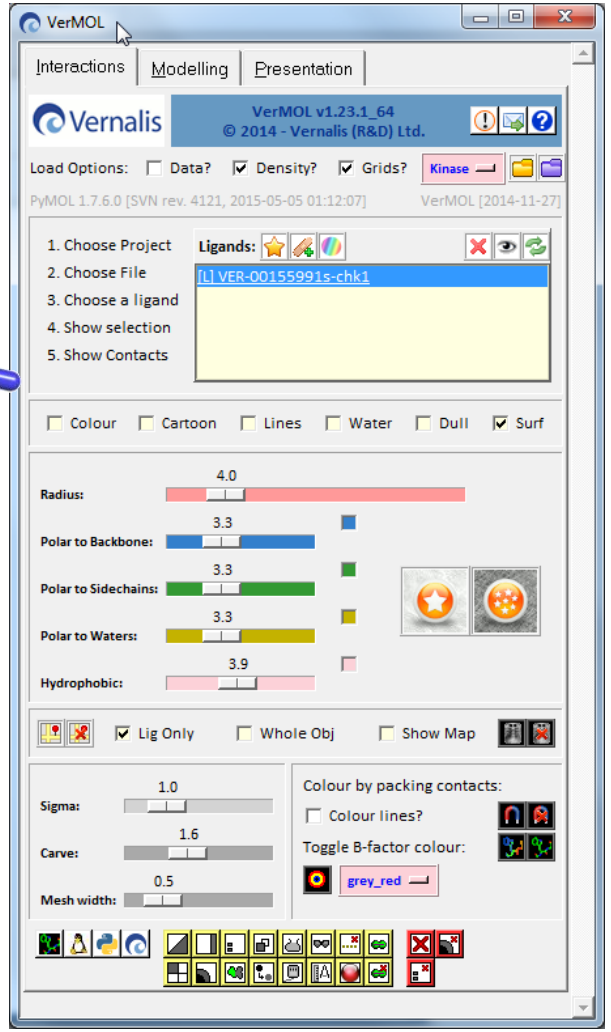
VerMOL *– PyMOL plugin to help our chemists*

Vernalis RESEARCH
part of Vernalis plc



PyMOL

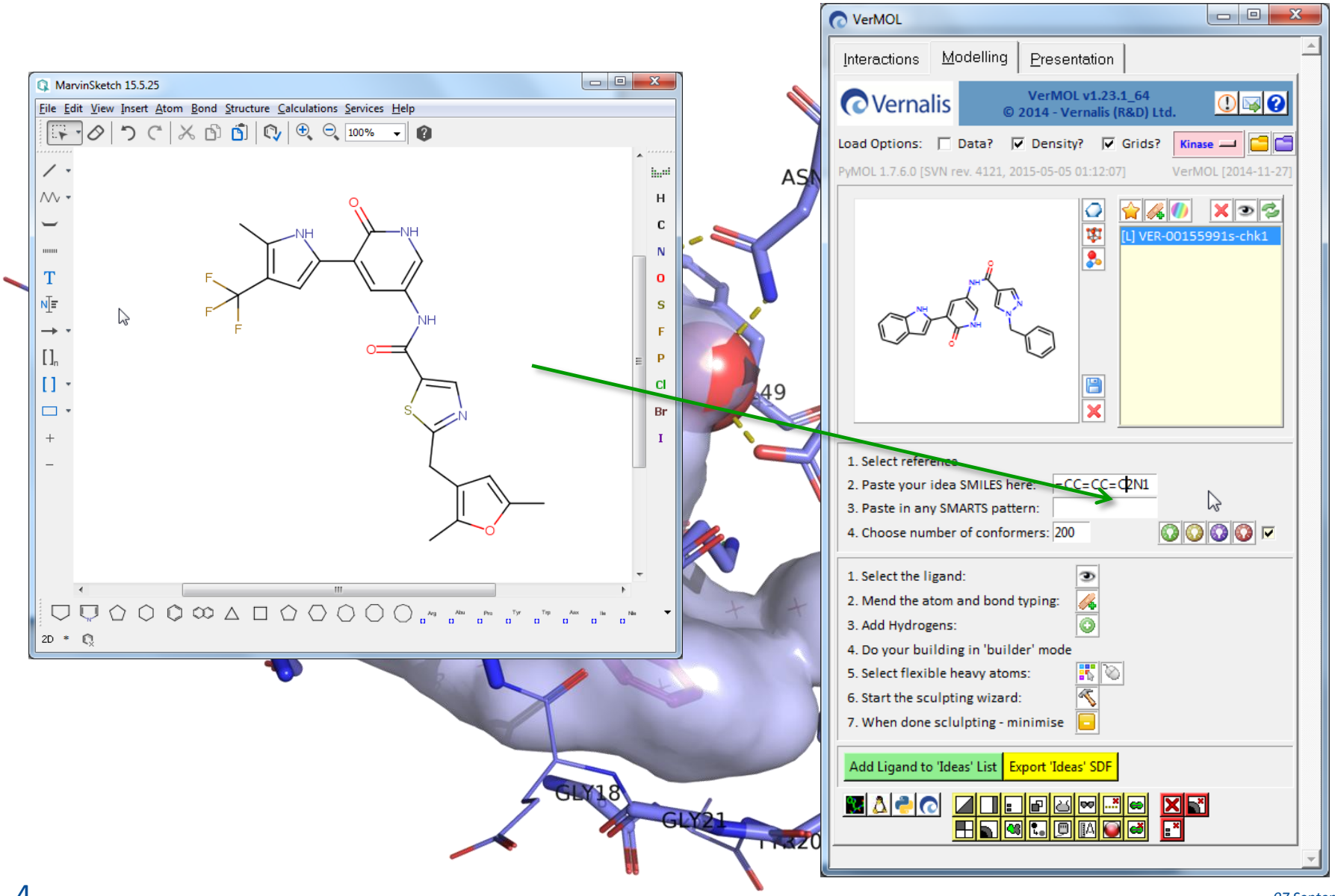
RDKit



Assessing a new idea

Sketch, copy, paste, score

Vernalis RESEARCH part of Vernalis plc



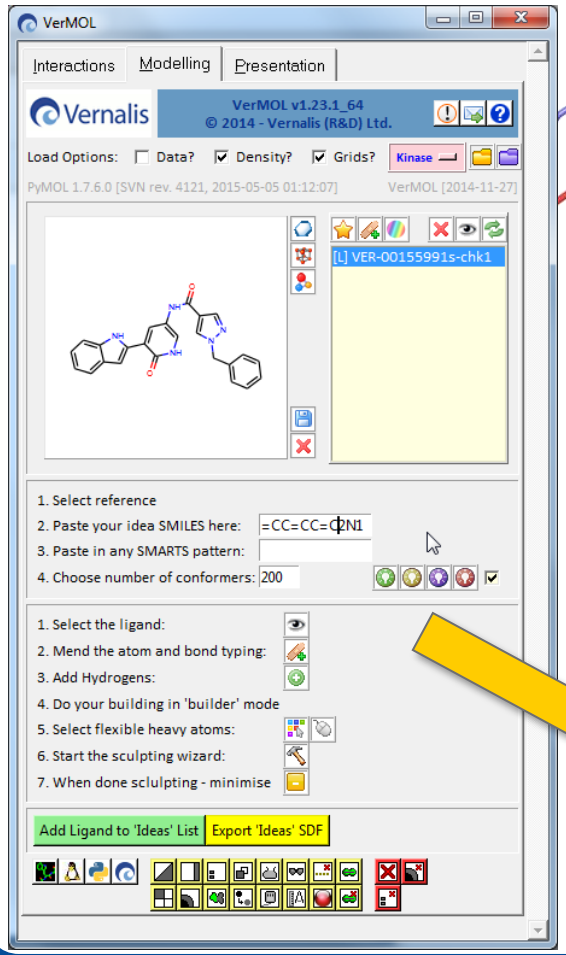
Scored idea

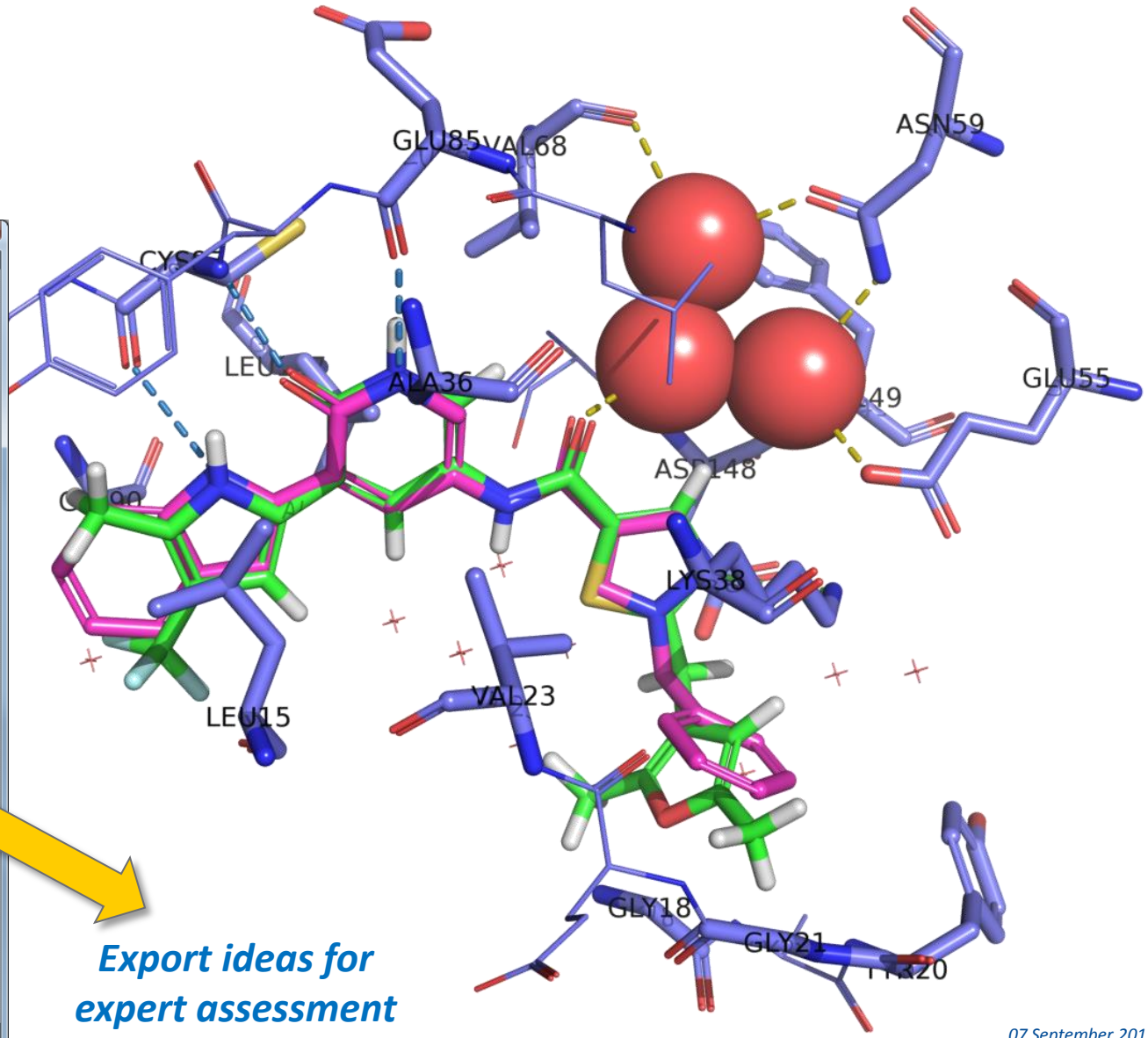
Based on best ligand overlay

Vernalis RESEARCH
part of Vernalis plc

```
idea_13 has the following properties:

Best Score          : 201.13
Conformers          : 166/200 converged
Max iterations      : 500
Energy (kcal/mol)   : 77.96
RMSD (A)            : 0.43
Forcefield          : MMFF94s
Synthetic Access    : 3.3
MW                  : 476
SlogP               : 5.2
TPSA                : 104
Fsp3                : 0.23
```

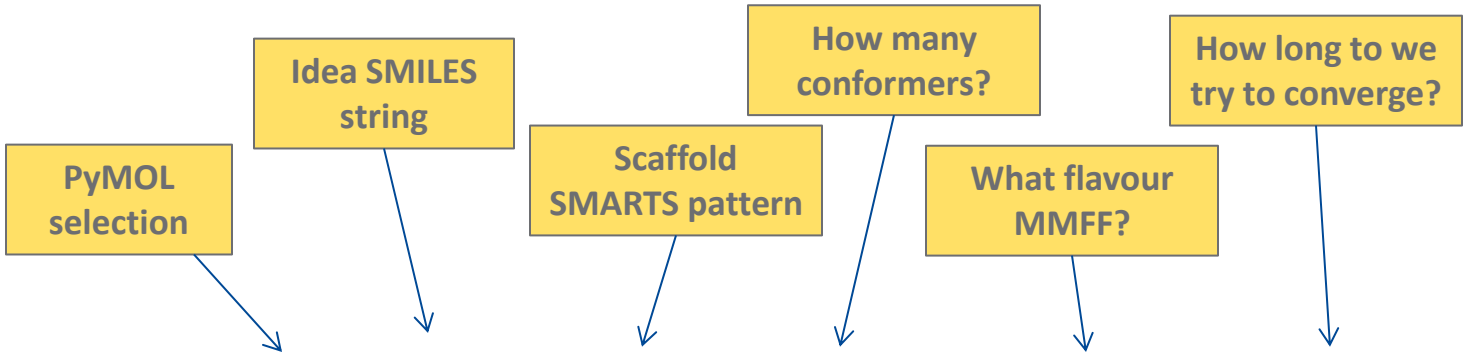




Export ideas for expert assessment

Implementation steps (and demo)

*Actually, for the live demo we'll use the **2C3K** PDB structure*



1. `generateIdeaConformers(sele, smiles, smarts, numConfs, mmFFVariant, maxIters, numThreads)`

   This is the 'orchestrating' function

2. `pymolSelectionToRDKit(sele)`

   Returns RDKit molecule for the reference ligand

3. `multiConfFromSmiles(smiles, nConfs, maxIters, mmFFVariant, numThreads)`

   `AllChem.EmbedMultipleConfs(idea, numConfs, numThreads)`

   `AllChem.MMFFOptimizeMoleculeConfs(idea, numThreads, maxIters, mmffVariant)`

   Returns multi-conformer RDKit molecule

4. `alignMultiConfToRef(multiConfMol, ref)`

   `AllChem.GetO3AForProbeConfs(multiConfMol, ref, prbPyMMFFMolProperties, refPyMMFFMolProperties, constraintMap, numThreads)`

   Returns pandas dataframe

5. `dataframeToPyMOLloadSDF(dataframe)`

   Loads the object back into PyMOL as an SDF (so that we can step through, etc)