Shuffling through a billion molecule database using RDKit-based tools while looking for CNS-relevant compounds

Tuomo Kalliokoski
tuomo.kalliokoski@orionpharma.com

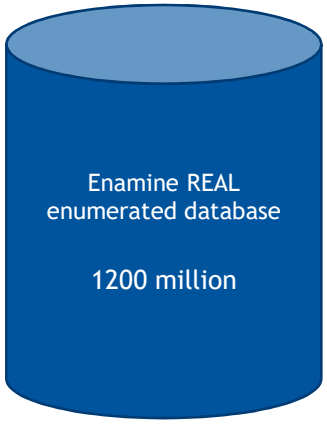
Computational Pharmacology and Medicine Design

ORION PHARMA
Building well-being

Background

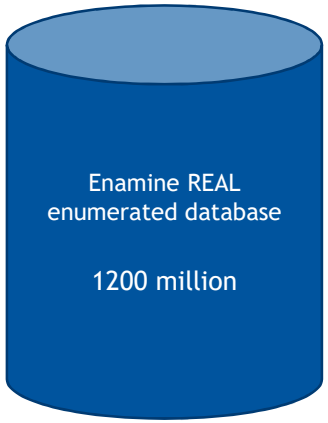
Ligand-based virtual screening with the throughput of 1,75 million compounds / hour

Per query 4 hours



Per query ~29 days

Background

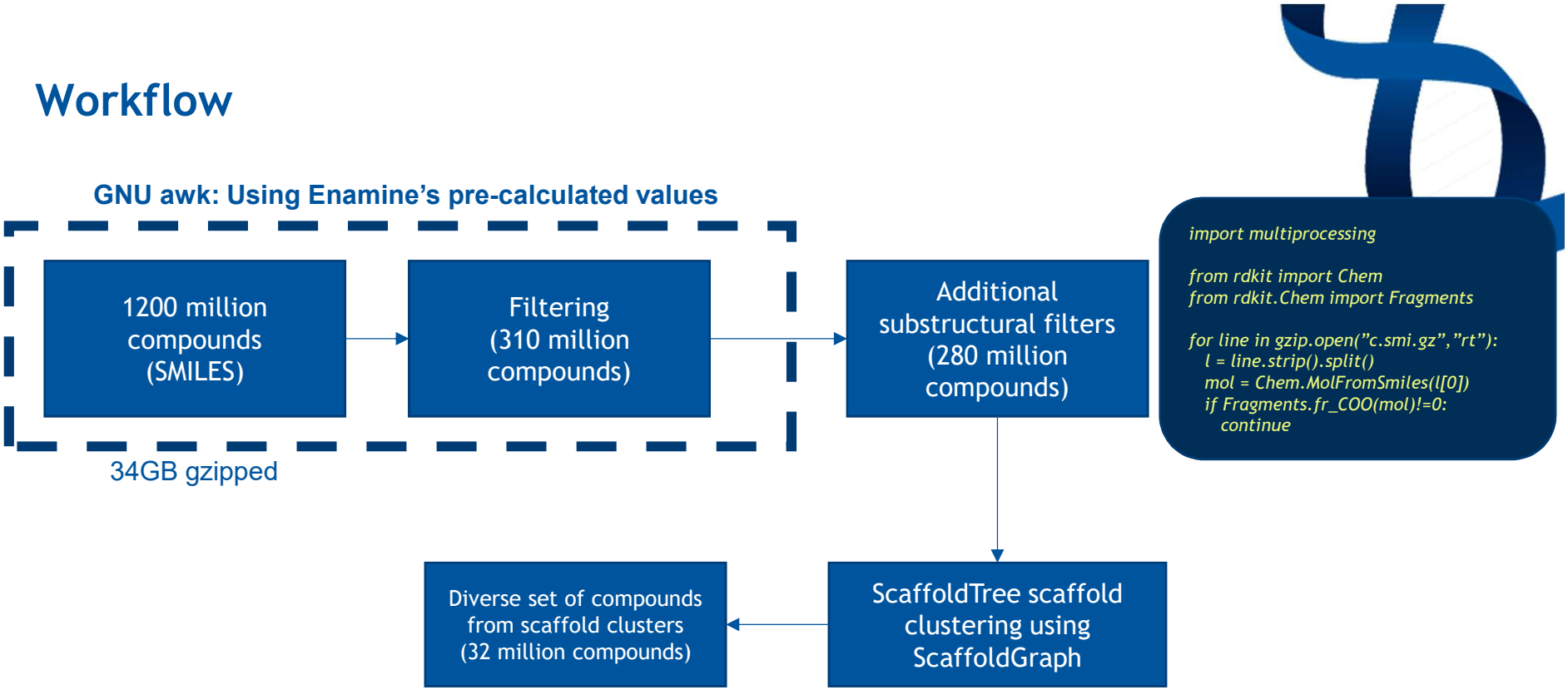


The current release of the *REAL* database comprises over 1.2 billion molecules which comply with "rule of 5" and Veber criteria: MW≤500, SlogP≤5, HBA≤10, HBD≤5, rotatable bonds≤10, and TPSA≤140.

CNS-relevant chemical space for hits is much smaller!

Pick a CNS subset that can be screened in a day (max ~40 million or so)

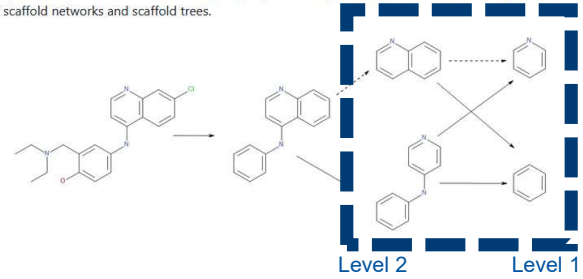
Workflow


GNU awk: Using Enamine's pre-calculated values
1200 million compounds (SMILES)
Filtering (310 million compounds)
Additional substructural filters (280 million compounds)
34GB gzipped
import multiprocessing
from rdkit import Chem
from rdkit.Chem import Fragments
for line in gzip.open("c.smi.gz","rt"):
l = line.strip().split()
mol = Chem.MolFromSmiles(l[0])
if Fragments.fr_COO(mol)!=0:
continue
ScaffoldTree scaffold clustering using ScaffoldGraph
Diverse set of compounds from scaffold clusters (32 million compounds)

ScaffoldTree clustering using ScaffoldGraph

ScaffoldGraph

ScaffoldGraph is an open-source cheminformatics library, built using RDKit and NetworkX, for the generation and analysis of scaffold networks and scaffold trees.



Scott and Chan. *Bioinformatics* **2020**,. 36 3930 DOI: 10.1093/bioinformatics/btaa219

https://github.com/UCLCheminformatics/ScaffoldGraph

280 million compounds → 773k scaffolds

With 50 cores in a reasonable time

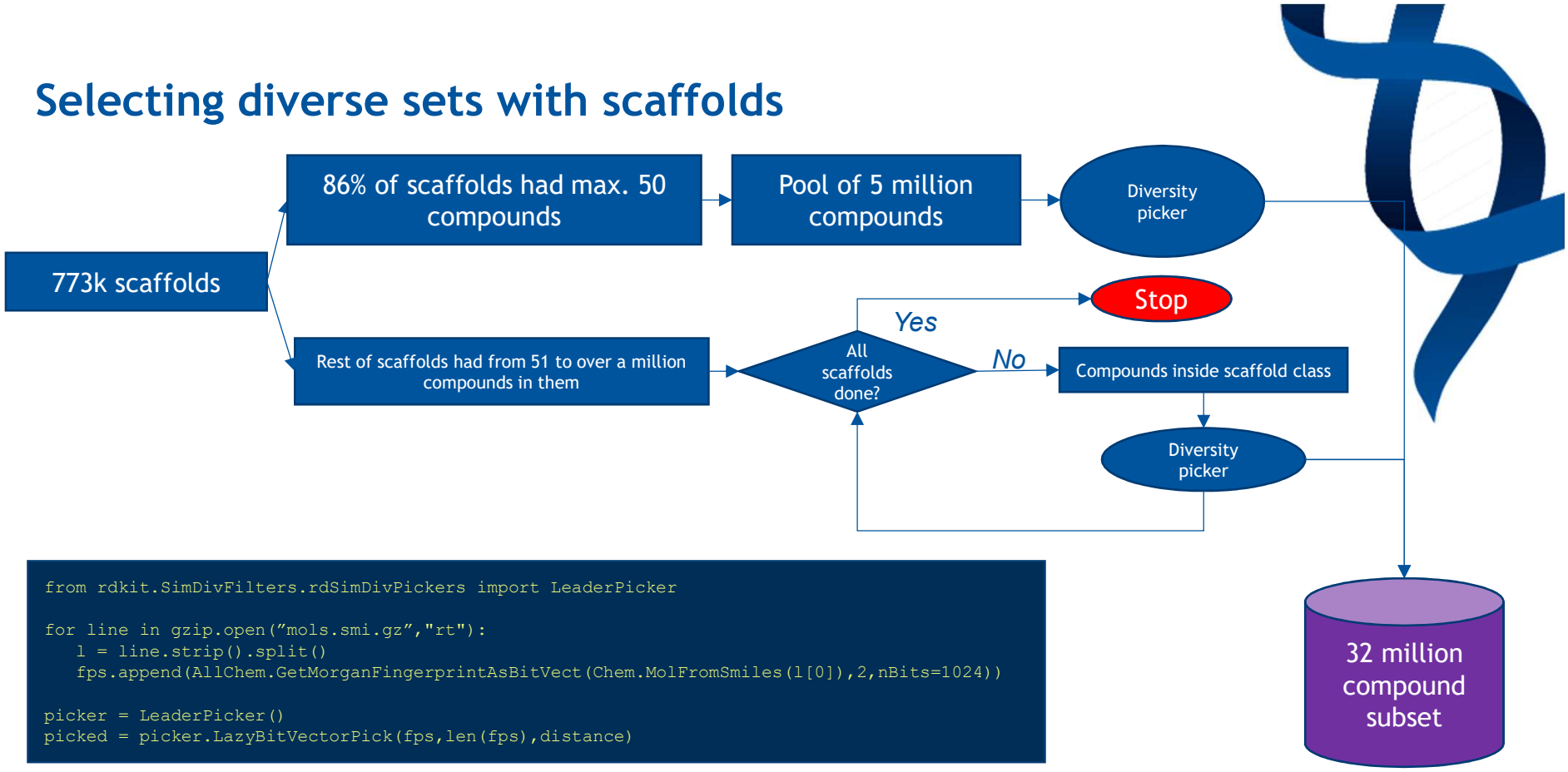
```
import scaffoldgraph as sg
import networkx as nx

def extract_scaffolds(input_filename,output_filename):
    tree = sg.ScaffoldTree.from_smiles_file(input_filename,progress=False)
    w = gzip.open(output_filename,"wt")
    for molecule in tree.get_molecule_nodes():
        scaffolds = list(nx.bfs_tree(tree,molecule,reverse=True))
        molecule_smiles = tree.nodes[scaffolds[0]]["smiles"]
        scaffold="NoRings!"
        if len(scaffolds)==2:
            scaffold = scaffolds[-1]
        else:
            scaffold = scaffolds[-2]
        w.write(molecule_smiles + " " +molecule+" "+scaffold+"\n")
    w.close()
```

Painless to install via conda:

```
conda config --add channels conda-forge
conda create --name scaffoldgraph python=3.7
conda activate scaffoldgraph
conda install -c uclcheminformatics scaffoldgraph
```

Selecting diverse sets with scaffolds




```
from rdkit.SimDivFilters.rdSimDivPickers import LeaderPicker

for line in gzip.open("mols.smi.gz","rt"):
   l = line.strip().split()
   fps.append(AllChem.GetMorganFingerprintAsBitVect(Chem.MolFromSmiles(l[0]),2,nBits=1024))

picker = LeaderPicker()
picked = picker.LazyBitVectorPick(fps,len(fps),distance)
```

See last year's RDKit UGM presentation by Roger: https://github.com/rdkit/UGM_2019/blob/master/Presentations/Sayle_Clustering.pdf