Library preparation: few comments shared here and there

- Pay attention to your initial library: does it contain counter salts for ionic species? Racemic mixtures of chiral compounds? Correct molecule naming/identifier?
- If it's a PDB format, check that you have the chemical compound you want! PDB => SDF/mol2 is rarely a safe conversion (almost never)
- Work as much as possible with SDF files. Convert to MOL2 only at the final stage
- Convert to MOL2 files with corina: RDKit MOL2 parser was coded with corina as reference. RDKit being the main/most complete/best coded cheminformatic tool, this will prove useful
- To my knowledge, there's nothing obabel does that RDKit doesn't do better and faster, apart from mol2 generation (on purpose). And corina is much faster and more reliable than obabel for that task

Library preparation: few comments shared here and there

- Try to fit your screening library to your task: what do you want as a result? Fragments? Charged molecules? Reduce noise before running things!
- Examples: logP <= 3, rotatable bonds <= 5, N ring > 1, HAC > 5 and < 35, CGenFF penalty exclusion…
- Don’t bother with PAINS/aggregator. Check them at purchase
- Only include/exclude certain substructures (example of targeted interaction with the outter glutamate of B2A or exclusion of nitro)
- Pay attention to the molecule names in files (unique, that let you identify the molecule properly)

Short description of the meta script

7 main steps

1. ***Generate tautomers*** *(including protomers) => cxcalc (**ChemAxon**), and select those with predicted occupancy > 25% (**python**)*
2. ***Generate conformers*** *=> RDKit ETKDG algorithm. Important to use this algo. Default parameters in the script are 100 runs of conformer generation, with a RMSD clustering (doesn't include symmetry consideration) of 0.75A, and a UFF vacuo minimization. Subsequent clustering to remove identical molecules around axis of symmetry. => **python + RDKit > 2017***
3. ***Convert to mol2*** *=> **corina***
4. ***Generate CGenFF parameters*** *for the first conformer of each tautomer => **CGenFF**. We don't need to lose time generating parameters for each conformer, they are the same molecule! But tautomers are not. In general, I name the molecules MOLNAME_tauto_X_conf_Y.sdf. Filter out molecules that generated .str files without parameters (non empty)*

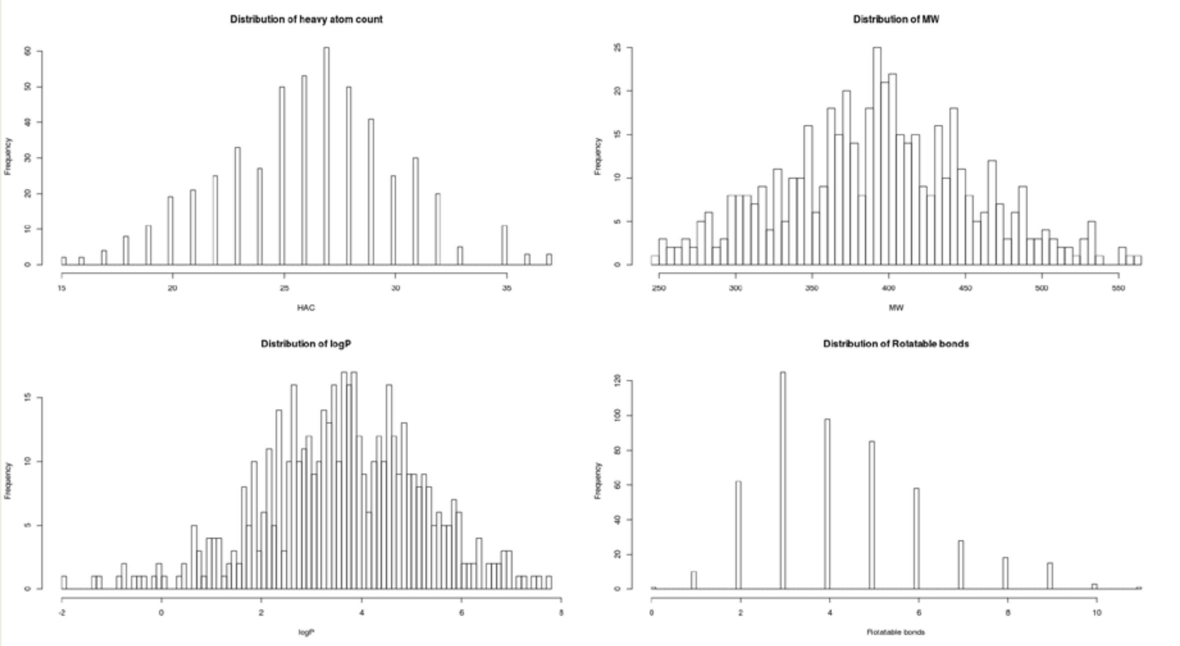
Short description of the meta script

5. ***Create MOL2 files for SEED*** *(replace partial charge field with CGenFF charge, assign ALT_TYPES with CGenFF atom types, calculate coordinates of LP particles and place them in the file =>* ***python***
6. ***Generate ABSINTH parameters*** *=>* ***python, RDKit***
7. ***Generate custom*** *CAMPARI/ABSINTH MOL2 files*

Benchmark of the meta script: how much time to expect to prepare a library?

Remark: there’s quite some I/O writing on the disk, working on an SSD speeds up the process. There will also be some issues with the huge number of files that can be generated in some cases, although everything should be concatenated or tared at the end

Benchmark set, n=504, «exhaustive conformer generation»



Benchmark

1. Tautomer generation: 1m17.368s (576 tautomers)
2. Conformer generation: 69m20.140s (38 558 structures)
3. Conversion to mol2 with corina: 4m16.956s
4. Generate CGenFF parameters: 0m24.255s (541 tautomers)
5. Build SEED MOL2: 8m10.150s (36 782 structures)

~ 10 seconds per compound in the original library (1h30 on 1 core)

6. Generate parameters for ABSINTH: 3m58.043s (537 tautomers)

7. Generate ABSINTH files: 0m5.749s (537 tautomers [of 36 382 structures]) (note these are just reference files to be updated with docked coordinates!)

~ 11 seconds per compounds in the original library

Note that with 504 initial molecules (assuming there's no need to generate isomers), we generate 40 000 files at steps 2 and 5.

Most of the time is spent generating conformers, but I wouldn't use a simpler function.