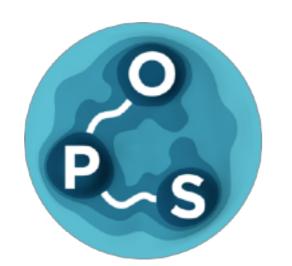
Path sampling: What's not great

- Still computationally intensive
 - 1000s of trajectories, ~10-100 ns each (better than normal MD!)
 - multiple walker parallelization helps somewhat
- Set-up can be difficult
 - identifying metastable states
 - getting initial trajectory
- No widely-used software for path sampling

OpenPathSampling



http://openpathsampling.org

Development at: http://github.org/openpathsampling/

Part of the Omnia consortium: http://omnia.md

A Python library for path sampling simulations

- ✓ Easy to use: Beginners can quickly learn to use it
- ✓ Easy to extend: Advanced users can use it to develop new methods
- ✓ Independent of dynamics engine: Useful in many fields and to the broadest audience

```
import openpathsampling as paths
in file = paths.AnalysisStorage("input file.nc")
init_traj = in_file.trajectories[0]
engine = in file.engines[0]
dist = in file.cvs['my distance']
stateA = paths.CVRangeVolume(dist, 0.0, 1.0)
stateB = paths.CVRangeVolume(dist, 3.0, float('inf'))
ensemble = paths.TPSEnsemble(stateA, stateB)
shooting mover = paths.OneWayShootingMover(ensemble)
init samp = paths.Sample(
   replica=0,
    trajectory=ensemble.split(init traj)[0],
    ensemble=ensemble)
out_file = paths.Storage("output.nc", "w", init_traj[0])
tps_calc = paths.PathSampling(
    storage=out_file,
    engine=engine,
   move scheme=paths.LockedMoveScheme(shooting mover),
    globalstate=paths.SampleSet([init samp]))
tps calc.run(1000)
```

DWHS, Prinz, Noé, Bolhuis, Chodera. In prep.