open
forcefield

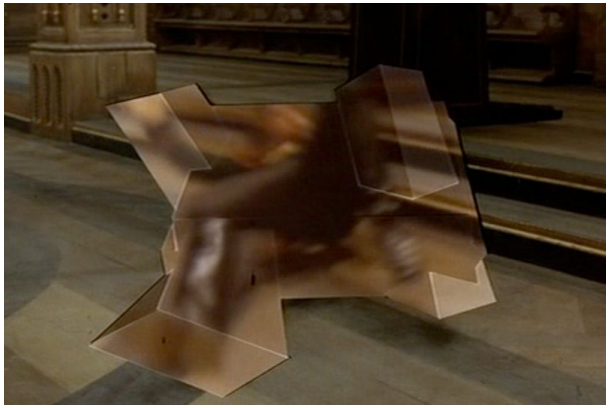
@openforcefield

www.openforcefield.org

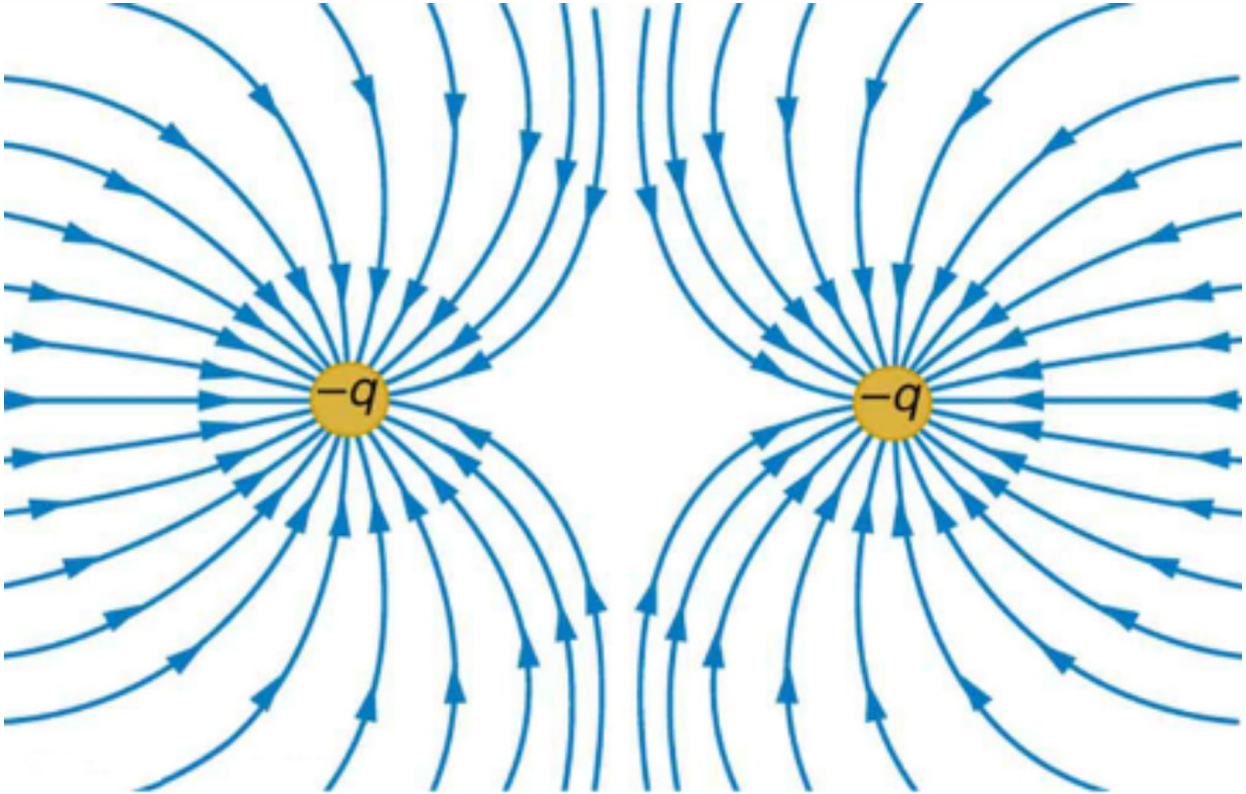
Open Force Field: next-generation force fields with open data, open software, and open science

Jeff Wagner, Technical Lead

What is a force field?



What is a force field?



What is a force field?

$$U(R) = \sum_{bonds} k_r (r - r_{eq})^2 \quad \text{bond}$$

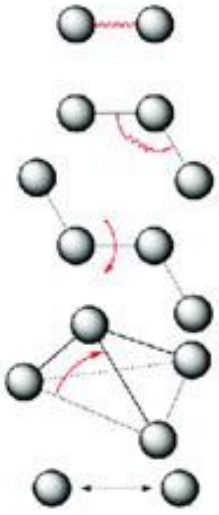
$$+ \sum_{angles} k_\theta (\theta - \theta_{eq})^2 \quad \text{angle}$$

$$+ \sum_{dihedrals} k_\phi (1 + \cos[n\phi - \gamma]) \quad \text{dihedral}$$

$$+ \sum_{impropers} k_\omega (\omega - \omega_{eq})^2 \quad \text{improper}$$

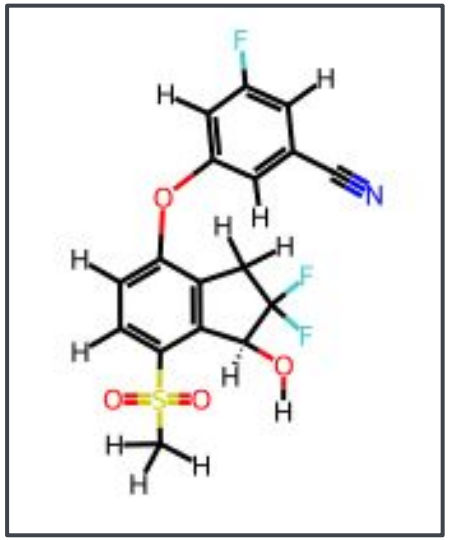
$$+ \sum_{i<j}^{atoms} \varepsilon_{ij} \left[\left(\frac{r_m}{r_{ij}} \right)^{12} - 2 \left(\frac{r_m}{r_{ij}} \right)^{6} \right] \quad \text{van der Waals}$$

$$+ \sum_{i<j}^{atoms} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \quad \text{electrostatic}$$

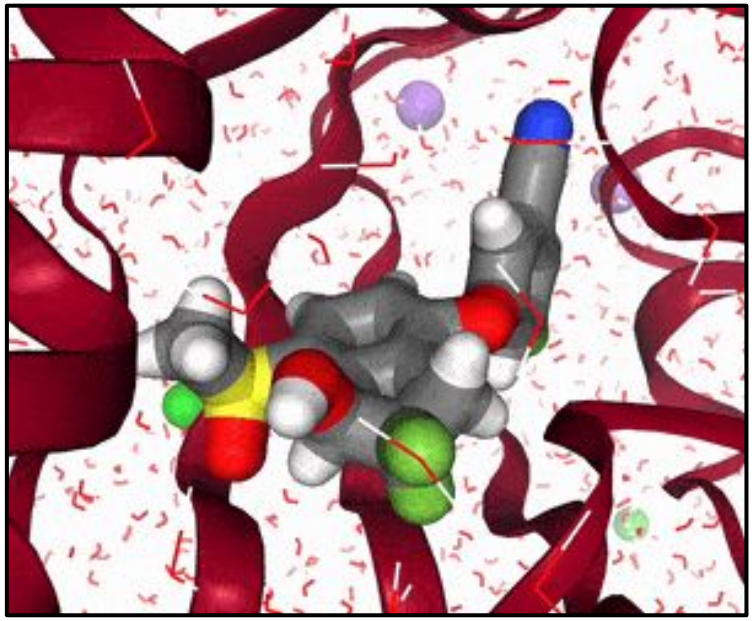


DOI: 10.3390/catal6060082

What is a force field?



Force field



The Open Force Field Consortium

INDUSTRY

AbbVie
Amgen
Bayer
Cresset
Eli Lilly
Janssen
OpenEye
Pfizer
Redesign
Roche
Ventus
Vertex
... and others

Open Molecular Software Foundation

PROJECT STAFF

Jeff Wagner
Technical Lead

Lily Wang
Science Lead

...

ACADEMIC

John Chodera (MSKCC)

Michael Gilson (UC San Diego)

David Mobley (UC Irvine)

Michael Shirts (CU Boulder)

Plus affiliates:

- Danny Cole (Newcastle)
- Lee-Ping Wang (UCD)
- Dennis Della Corte (BYU)
- MolSSI (Virginia Tech)

How is the Open Force Field Initiative open?

Open source Python Toolkit: use the parameters in most simulation packages

Open curated QM / physical property datasets: build your own force fields

Open infrastructure: Run your own benchmarks; fit your own FFs

Open science: Everything done in the open/everyone can get involved

What is the Open Force Field Initiative producing?

Toolkits: Modern toolkits for rapid development, application, and evaluation of force fields

Parameters: Parameterized datasets for different model resolutions

Datasets: Curated collections of physical property measurements

Community: Bringing together top developers & users and working together to solve our problems in the open so everyone benefits

Best Practices: Measurement and calculation of physical properties

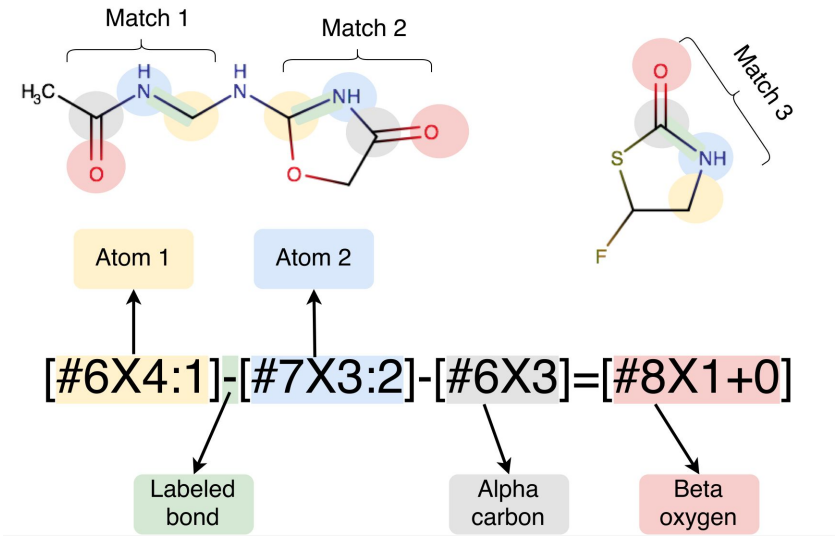
Standards: Representation of molecular systems; forcefield descriptions

Documentation: Theory; toolkit documentation; tutorials and training materials

Publications: Communicating the ideas behind our work to the scientific community

The SMIRKS Native Open Force Field spec. (SMIRNOFF) avoids atom typing, simplifies parameter assignment

match bonds directly:



Use of industry-standard SMARTS/SMIRKS chemical perception greatly simplifies tooling for parameter assignment while solving issues with extensibility and flexibility

SMIRNOFF allowed significant compression of smirnoff99Frosst, our AMBER-lineage starting point

Description	Force Field	Lines of parameters
Basic Amber FF:	parm99	720
Merck Frosst small mol:	parm@Frosst	2893
Total:		3613

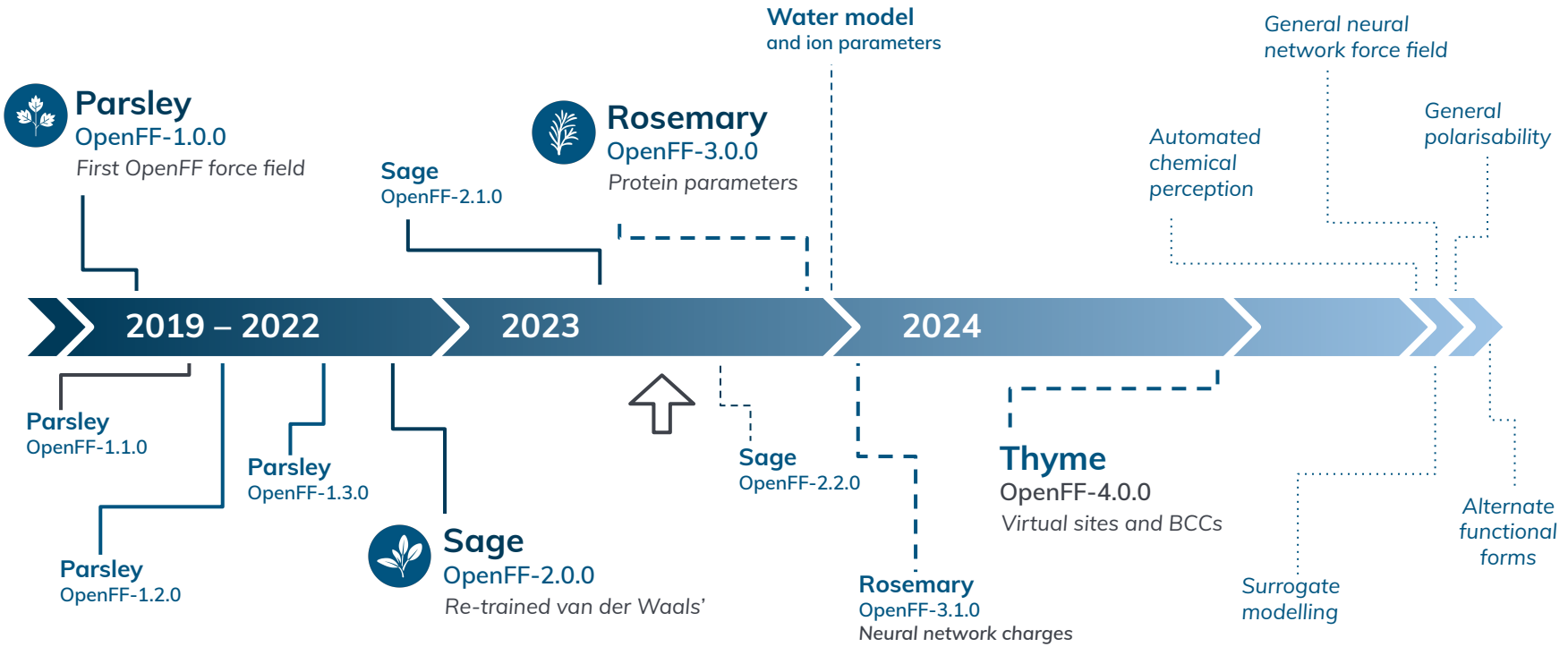
smirnoff99Frosst	332

Chemical Space Coverage

Database	smirnoff 99Frosst	parm @Frosst
DrugBank	99.7%	60%
ZINC	99.8%	52%
eMolecules	99.5%	--

- Less than 1/10 the size of the original force field
- Removes redundancy
- Almost completely covers pharmaceutical chemical space

Force Field Roadmap




DO YOU WANT TO KNOW THE TERRIFYING TRUTH... OR DO YOU WANNA SEE ME SOCK A FEW DINGERS?

Vignettes

Code and input files available at
https://github.com/openforcefield/2023-workshop-vignettes

Some of this is experimental: no promises, run your own benchmarks!

- Production ready, we stand by this!
- In the public API, but expect bugs
- Prototype, experimental

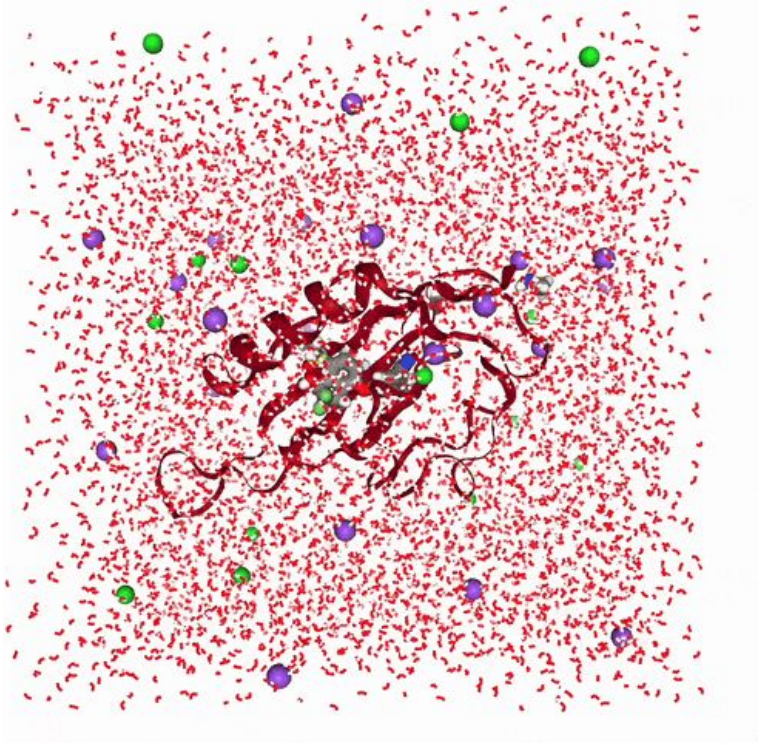
Run an simulation with 7 lines of OpenFF code

OpenFF Code

```
from openff.toolkit import Molecule, Topology, ForceField
ligand = Molecule.from_file('inputs/PT2385.sdf')
top = Topology.from_pdb('inputs/solvated_complex.pdb',
                        unique_molecules=[ligand]
                        )
ff = ForceField("openff-2.0.0.offxml", "ff14sb_off_impropers_0.0.3.offxml")
sys = ff.create_openmm_system(top)
```

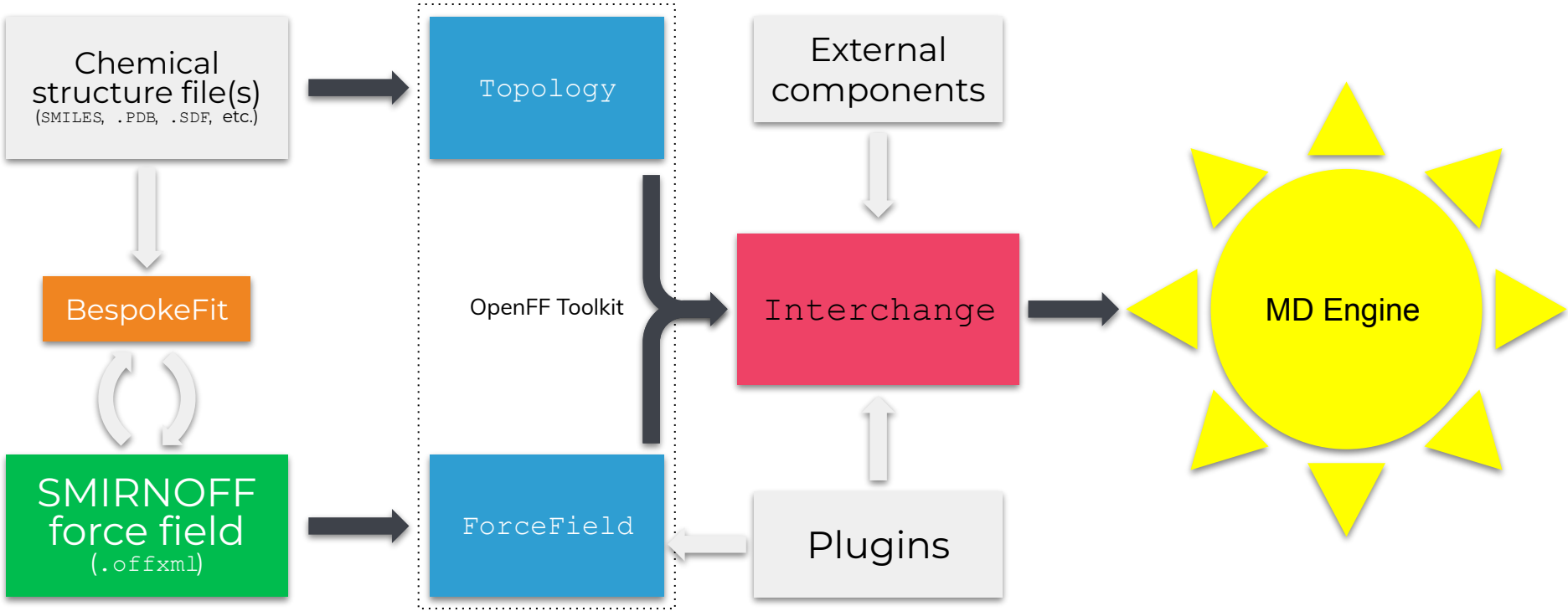
OpenMM Code

```
import openmm
from openff.units import Quantity, unit
from openmm import unit as openmm_unit

# Construct and configure a Langevin integrator at 300 K with an appropriate friction consta
integrator = openmm.LangevinIntegrator(
    300 * openmm_unit.kelvin,
    1 / openmm_unit.picosecond,
    0.002 * openmm_unit.picoseconds,
)

# Combine the topology, system, integrator and initial positions into a simulation
simulation = openmm.app.Simulation(top.to_openmm(), sys, integrator)
simulation.context.setPositions(top.get_positions().to_openmm())

# Add a reporter to record the structure every 10 steps
dcd_reporter = openmm.app.DCDReporter("trajectory.dcd", 250)
simulation.reporters.append(dcd_reporter)
simulation.context.setVelocitiesToTemperature(300 * openmm_unit.kelvin)
simulation.runForClockTime(1.5 * openmm_unit.minute)
```



- PDB-to-simulation

The OpenFF Workflow



Dependencies/Interoperability

Upstream

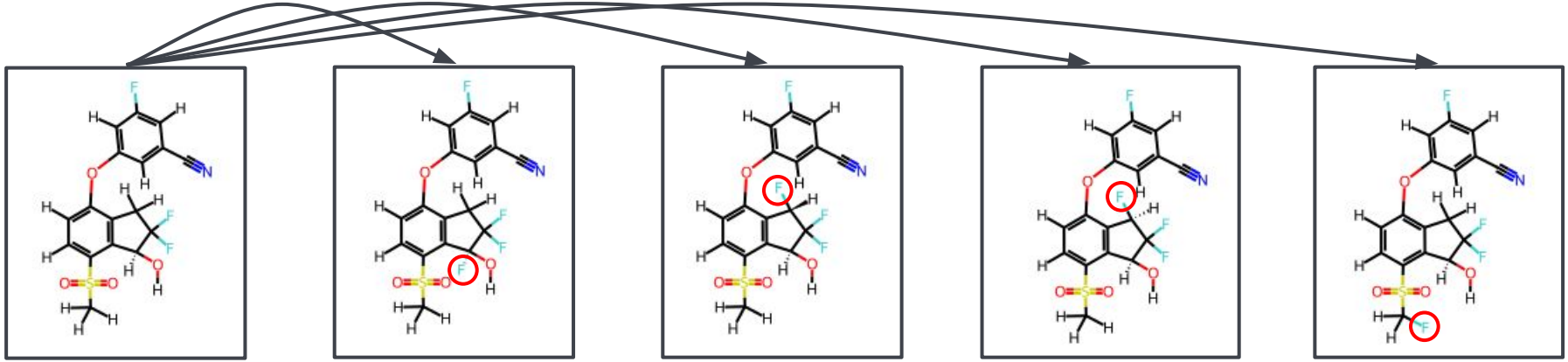
- Numpy
- Pint
- OpenMM (and thereby MDAnalysis/MDTraj)
- RDKit/OpenEye
- QCFractal

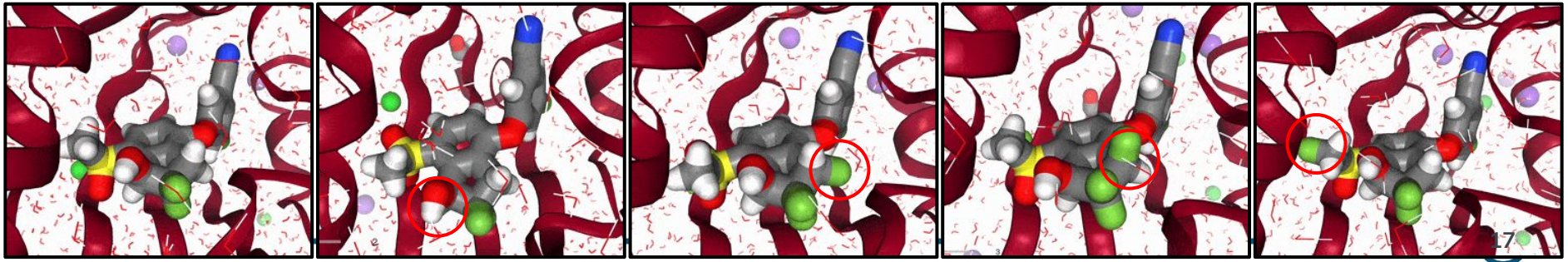
Downstream

- OpenFE
- OpenMM (and thereby NglView/MDAnalysis/MDTraj)
- ParmEd
- AMBER/GROMACS/other MD engines
- QCFractal

OpenFF is highly interoperable with RDKit

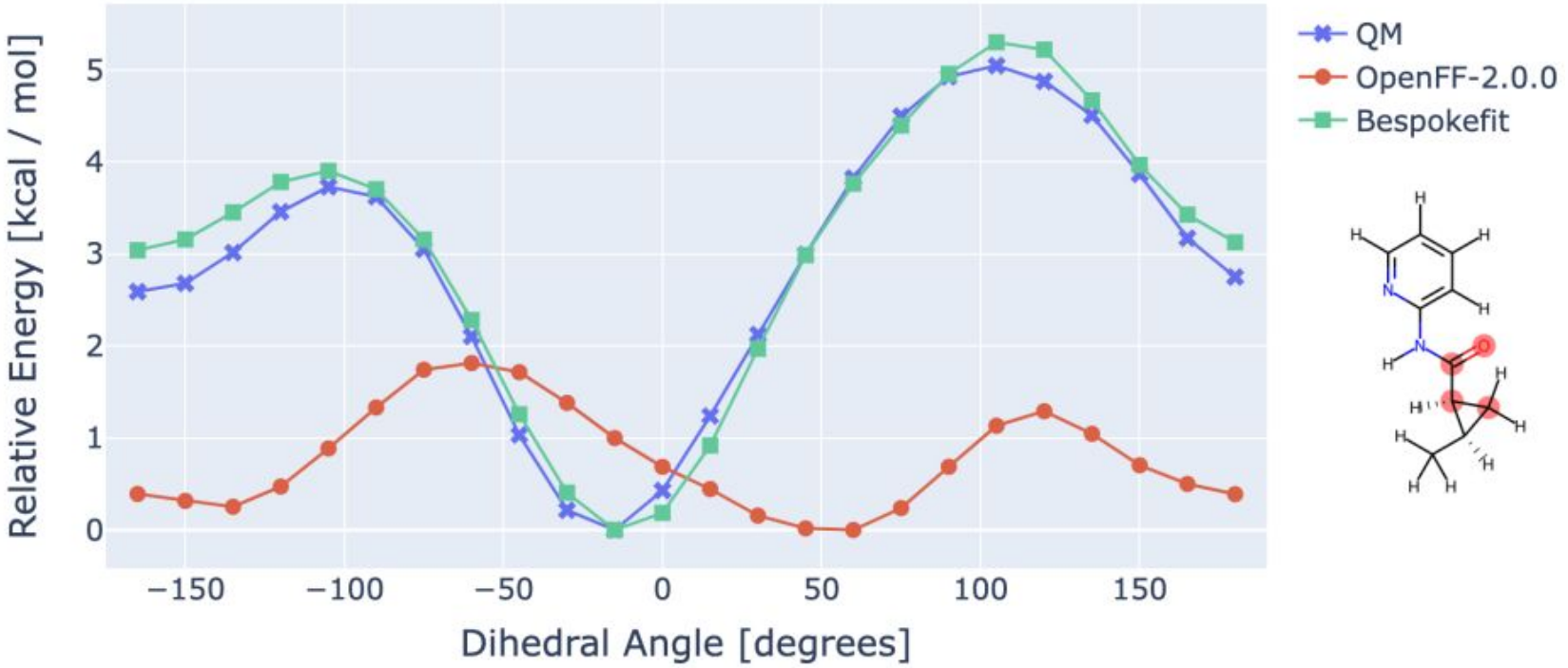
```
rxn = rdkit.Chem.rdChemReactions.ReactionFromSmarts("[C:1][H:2] >> [C:1][F:2]")
```



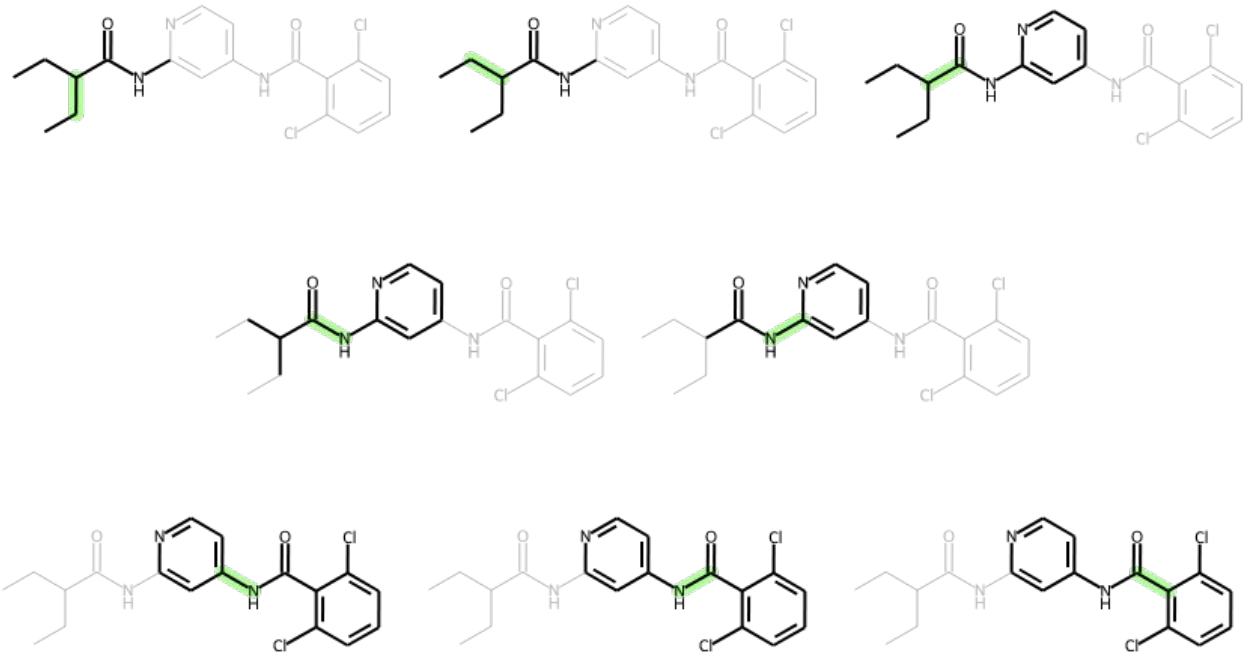


- rdkit-ligand-modification (~70 LoC, ~10 minute runtime)

BespokeFit

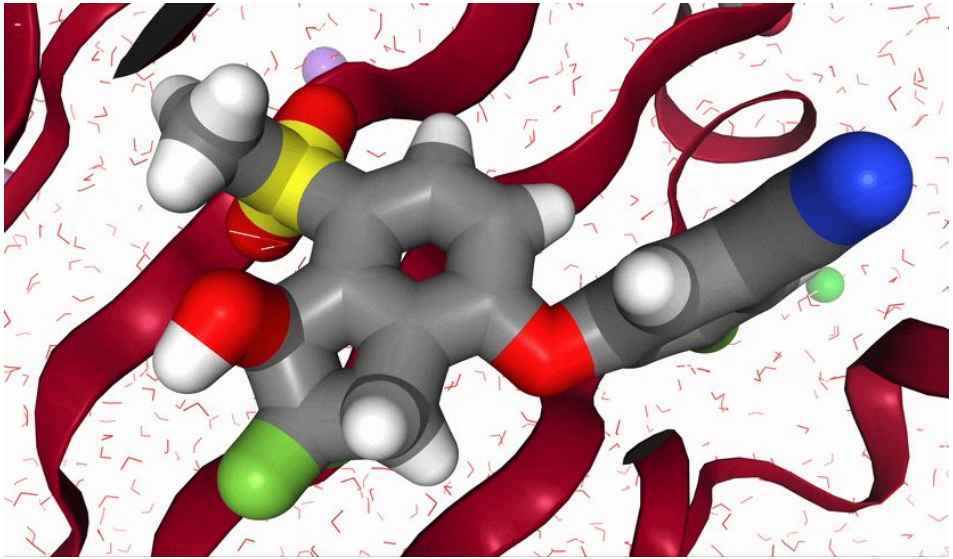

Relative Energy [kcal / mol]
5
4
3
2
1
0
−150
−100
−50
0
50
100
150
Dihedral Angle [degrees]
QM
OpenFF-2.0.0
Bespokefit

BespokeFit



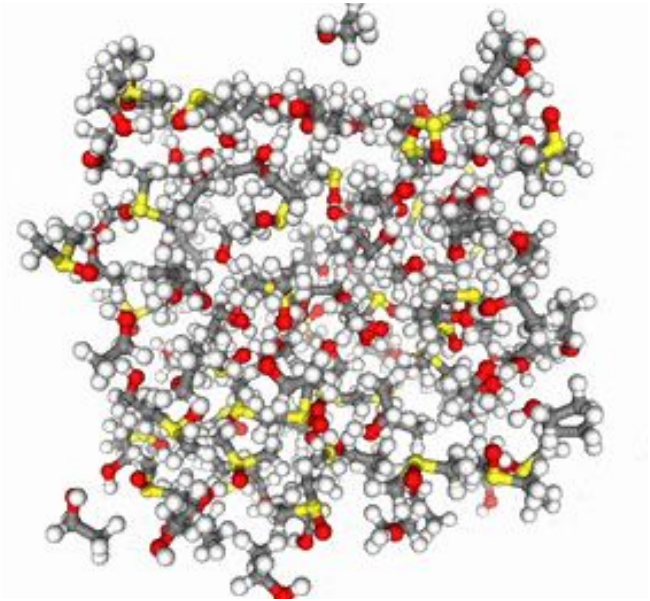
BespokeFit

```
!openff-bespoke executor run                                  \
    --file                "inputs/PT2385.sdf"                 \
    --workflow            "default"                           \
    --output              "PT2385.json"                       \
    --output-force-field "openff-2.0.0-PT2385.offxml" \
    --n-qc-compute-workers 1                                  \
    --qc-compute-n-cores   10                                 \
    --default-qc-spec xtb gfn2xtb none

from openff.toolkit import ForceField

ff = ForceField(
    "openff-2.0.0-PT2385.offxml",
    "ff14sb_off_impropers_0.0.3.offxml",
)
```

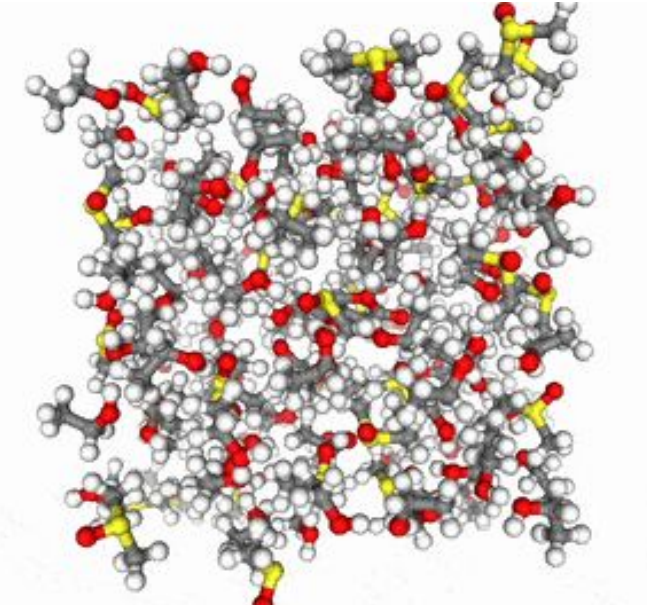


Interchange can export to different MD engines

```
system = interchange.to_openmm()
```

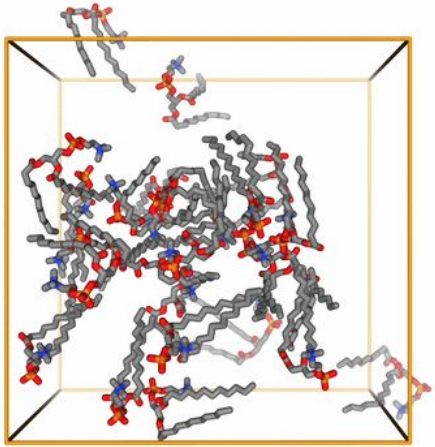


```
interchange.to_gromacs("gromacs_input")
```



Micelle self assembly

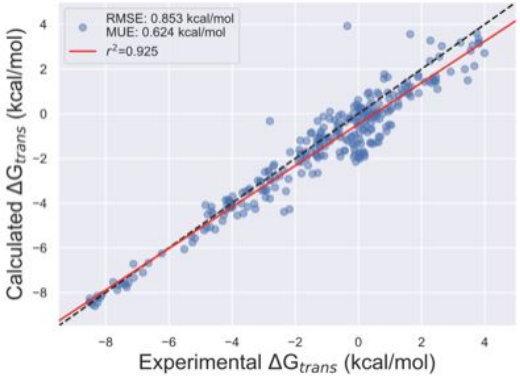
```python
from openff.interchange.components._packmol import pack_box
from openff.toolkit import Molecule
from openff.units import unit

dlpc_smiles = "CCCCCCCCCCCC(=O)OC[C@H](CO[P@](=O)([O-])OCC[N+](C)(C)C)OC(=O)CCCCCCCCCCC"
molecules = [
    Molecule.from_smiles(dlpc_smiles),
    Molecule.from_smiles("[H]O[H]"),
]

n_copies = [4000, 25]

target_density = 1.0 * unit.gram/unit.liter
total_mass = sum([
    sum([atom.mass for atom in molecule.atoms]) * n
    for molecule, n in zip(molecules, n_copies)
])
target_volume = total_mass / target_density
box_size = np.ones(3) * np.cbrt(target_volume)

topology = pack_box(
    molecules,
    n_copies,
    box_size=box_size,
    tolerance=0.05 * unit.nanometer,
)
```

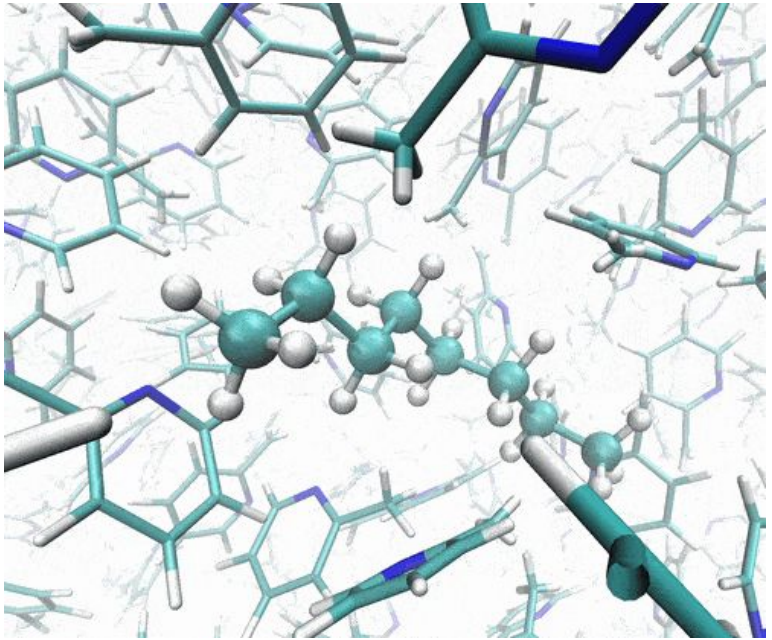


Modified functional forms and plugins

$$U^{B68}(r) = A\exp(-br) - f_{damp,6}(r)\frac{C_6}{r^6} - f_{damp,8}(r)\frac{C_8}{r^8}$$

```
from openff.toolkit import ForceField
ff = ForceField(
    "de-force-1.0.0.offxml",
    load_plugins=True
)
```

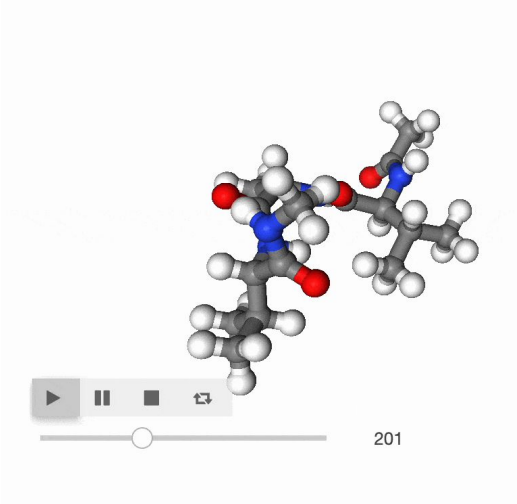


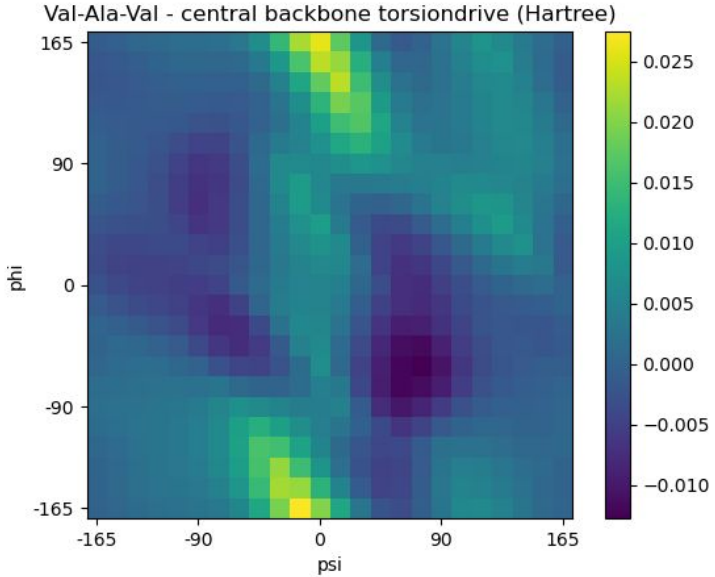


A transfer free energy simulation of octane in 2-methylpyridine

https://doi.org/10.26434/chemrxiv-2023-28r9s / https://github.com/jthorton/de-forcefields

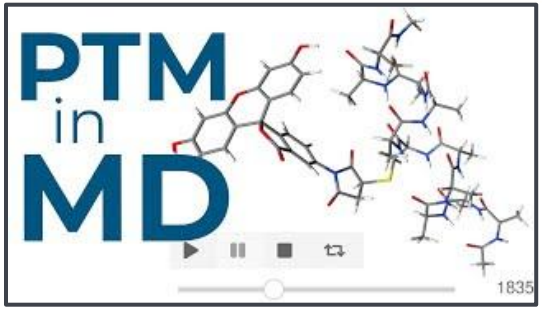
An MM-person's guide in QC-land, pull down our datasets as graph molecules with QM energies, filter for bad cases like connectivity rearrangements

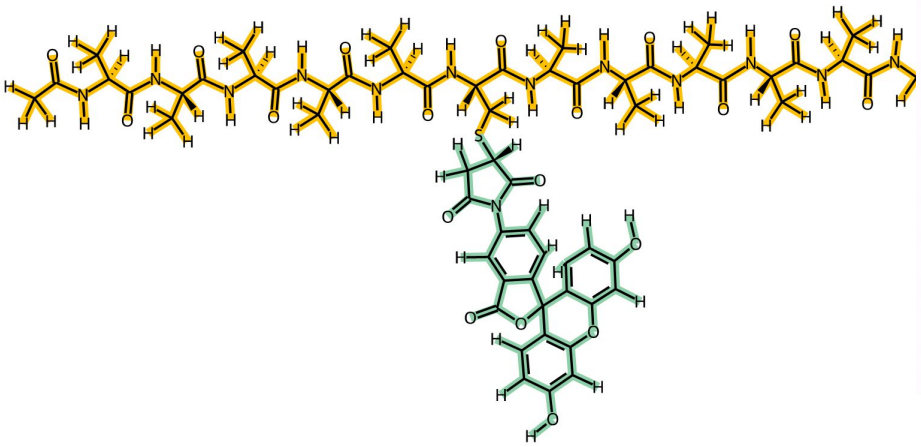


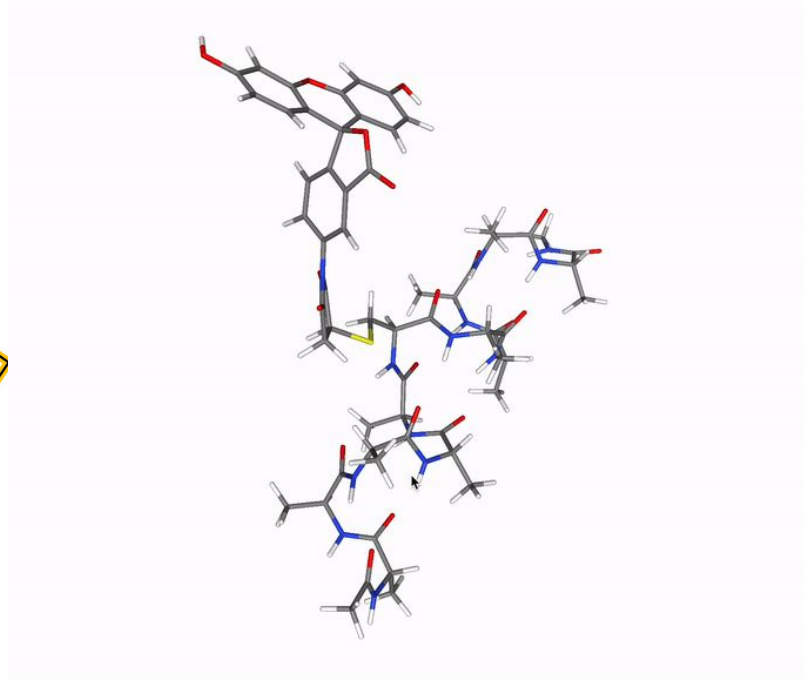


QC Archive
A MolSSI Project

SMIRNOFF format allows blending force fields







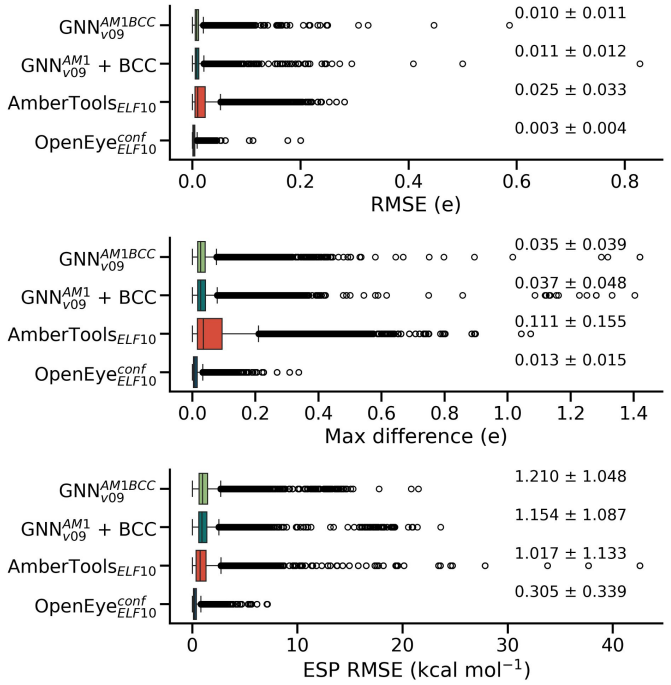
Expanding electrostatics with neural network charges

Rosemary 3.1.0 will use neural network charges

- We now have working models that perform similarly to existing backends

```
In [1]: from openff.toolkit import Molecule
        from openff.toolkit.utils._nagl_wrapper import _NAGLToolkitWrapper
        from rdkit import Chem

        rd_gb3 = Chem.MolFromFASTA(
            "MQYKLVINGKTLKGETTTKAVDAETAE"
            "KAFKQYANDNGVDGVWTYDDATKTFTVTE"
        )
        off_gb3 = Molecule.from_rdkit(rd_gb3)
        off_gb3.n_atoms

Out[1]: 864

In [2]: %%time

        off_gb3.assign_partial_charges(
            "_nagl_am1bccelf10",
            toolkit_registry=_NAGLToolkitWrapper()
        )

CPU times: user 45.8 s, sys: 2.1 s, total: 47.9 s
Wall time: 38.4 s
```

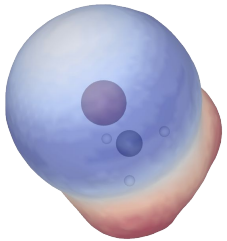
Prototype now available (**not the most up-to-date model!*)



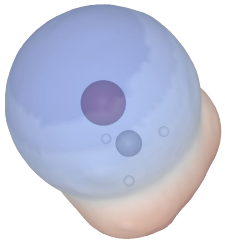
OpenFF-Recharge improves fits to electrostatics with virtual sites

Virtual sites are now a top priority for small molecule improvement

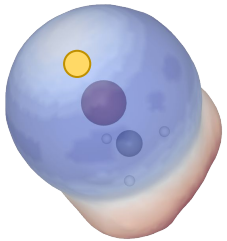
HF/6-31G*

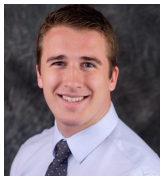


Sage



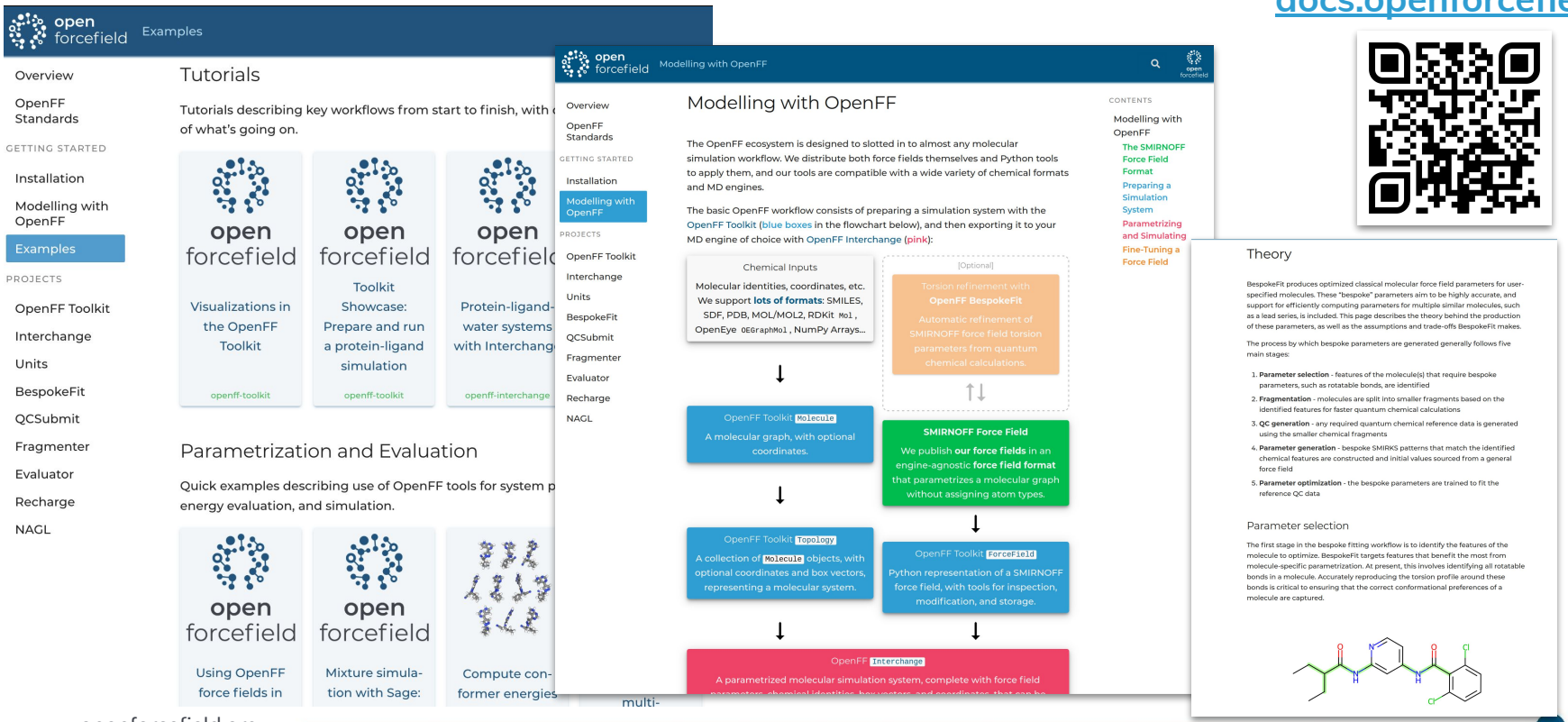
With virtual sites





Docs

docs.openforcefield.org



Improving infrastructure usability

Interchange expansion

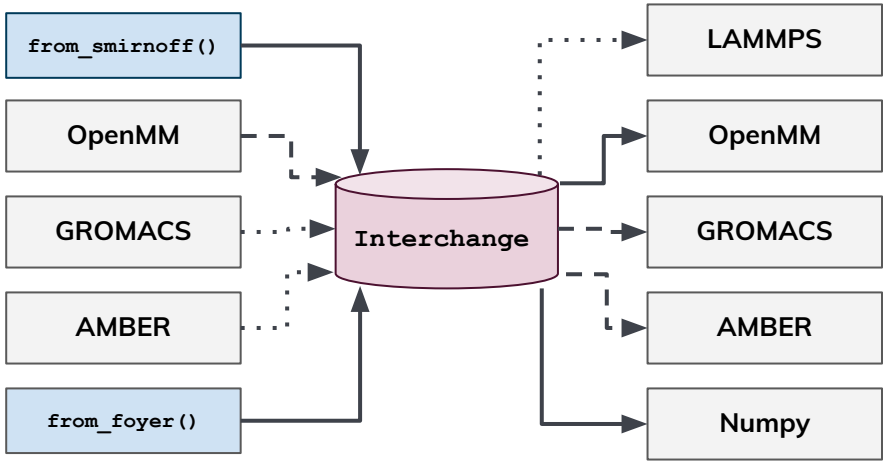
- “Vanilla” importers for OpenMM, AMBER, GROMACS, etc.
- Usability for protein-ligand workflows

Expanded polymer loading

- Multi-component protein, DNA, RNA, custom monomer units
- PDBx/mmcif

Centralized documentation

- Ecosystem wayfinding
- Additional, centralised workflow examples

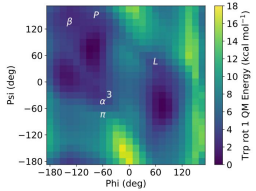


Force field Roadmap

Rosemary (template charges)

OpenFF-3.0.0

- Self-consistent force field for biopolymers and small molecules
- Incorporating protein-specific torsions
- Benchmarked against QM and physical property data
- Includes template partial charges for standard residues



Thyme

OpenFF-4.0.0

- Re-optimized BCCs to HF/6-31G* (or higher) electric field
- Virtual sites fit to HF/6-31G* (or higher) electric field

2023

Sage minor release

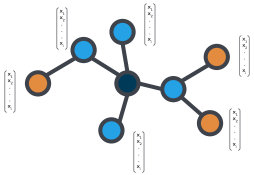
OpenFF-2.1.0

- New training targets
- Broader parameter coverage
- New parameters
- Targeted improvements for specific functional groups, e.g. sulfonamides

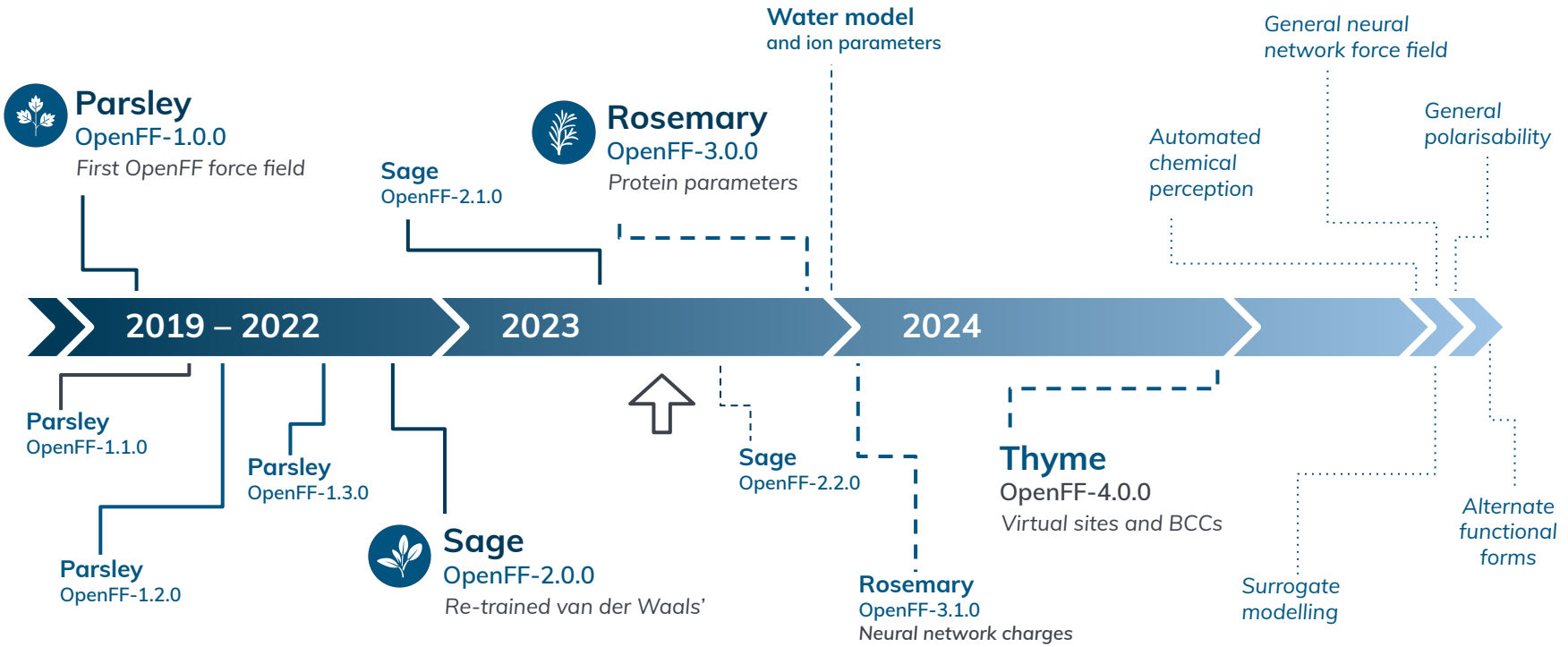
Rosemary (Graph charges)

OpenFF-3.1.0

- Use convolutional graph neural networks (CGNNs) to generate AM1-BCC-like charges
- Charges are fast and conformer-independent
- Enables custom macromolecule support



Force field Roadmap



Open Free Energy

Open Free Energy is funded and governed by a pre-competitive collaboration of industry partners.

The project is hosted by the Open Molecular Software Foundation (OMSF) a 501(c)(3) nonprofit.

https://omsf.io

Our software helps researchers performing free energy calculations by providing tools for preparing, defining, executing and analysing calculations.

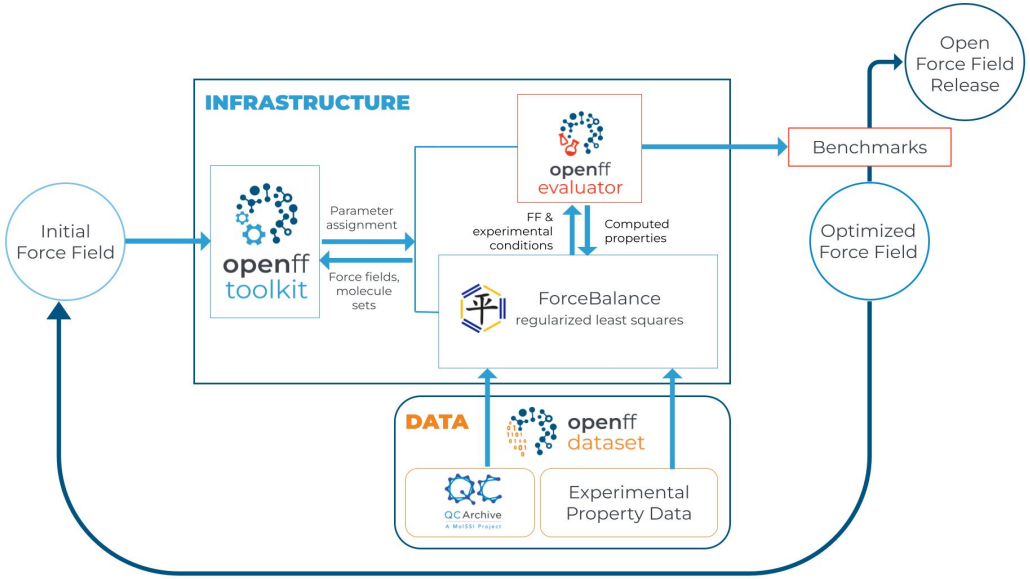
We are maintaining and building an ecosystem of tools for free energy calculations which are:

- ➔ Free to commercially use and extend
  - Permissive open source (MIT)
- ➔ Interoperable
  - Compatible with existing workflows
  - No lock in to any ecosystem / engine
- ➔ Robust
- ➔ Automated and scalable
  - Designed for simple large scale deployment
- ➔ Modular and extensible
  - Clear API to allow reuse and adaptation of components

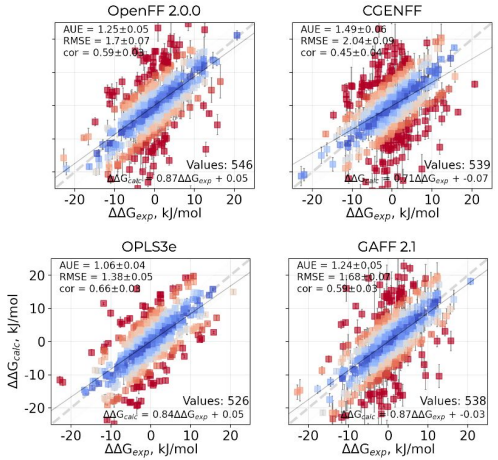
https://github.com/OpenFreeEnergy

OpenFF partners with industry to build accurate, pre-competitive force fields for general use

Our infrastructure automates fitting, allowing force field science and driving innovation



OpenFF 2.0.0 (Sage) outperforms other public small molecule force fields



The OpenFF community shares insights, experiences, code and workflows
More than a dozen industry partners participate. **Contact**: info@openforcefield.org

doi: 10.1021/acs.jcim.1c01445
doi: 10.26434/chemrxiv-2022-n2z1c-v2

OPEN Software, OPEN Data, OPEN Science is rapidly facilitating force field science!

OPEN SOFTWARE

Automated infrastructure enables rapid experimentation with minimum human intervention

OPEN DATA

Access to large, high quality experimental and quantum chemical data facilitates easy curation of balanced train / test sets

OPEN SCIENCE

Exploring new force field science:

hypothesis - build software - train - test - iterate

is now almost routine

open
forcefield

docs.openforcefield.org

Website
https://openforcefield.org/

GitHub
https://github.com/openforcefield/

Zenodo
https://zenodo.org/communities/openforcefield/

Twitter
https://twitter.com/openforcefield

YouTube
https://www.youtube.com/channel/UCh0qJSUm_sYr7nuTzhW806g/videos

LinkedIn
https://www.linkedin.com/company/openforcefield

Email
info@openforcefield.org

1. What is OpenFF?
    a. Mission
    b. People + Industry partners
    c. Standard workflow
    d. Docs.openff.org
2. What are our current capabilities?
    a. Toolkit Showcase (Toolkit+Interchange)
    b. Data from QCArchive (QCSubmit+QCF)
    c. DEXP sim (plugins+Interchange+Evaluator)
    d. BespokeFit Workshop (BespokeFit)
    e. NCAA parameterization
    f. Custom substructures + NAGL (NAGL)
3. Where are we going?
    a. Roadmap - Particularly biopolymer FFs and vsites
    b. How to join
    c.