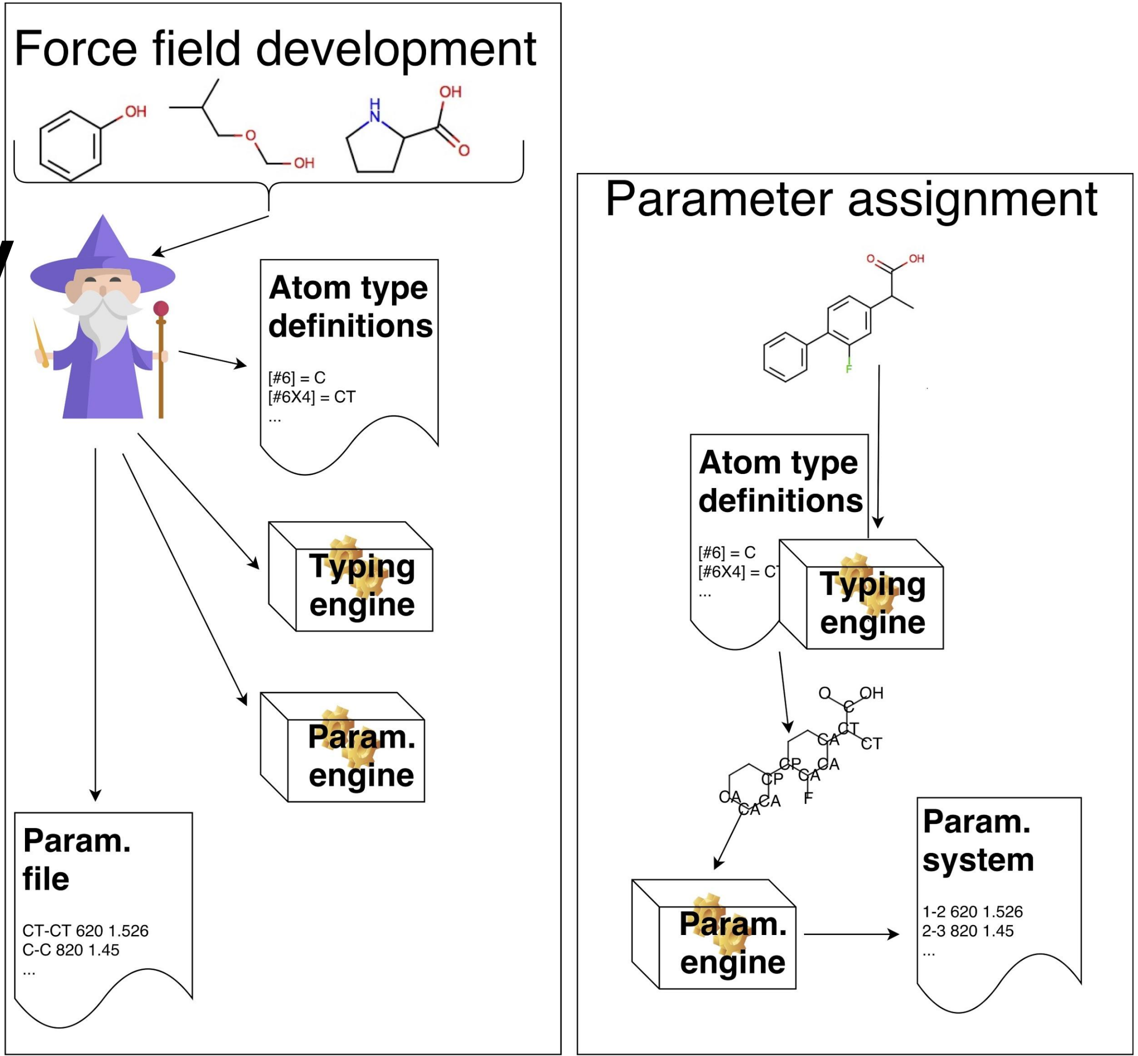
# Chemical perception and SMIRNOFF typing

**Caitlin C. Bannan and David L. Mobley** (UC Irvine)

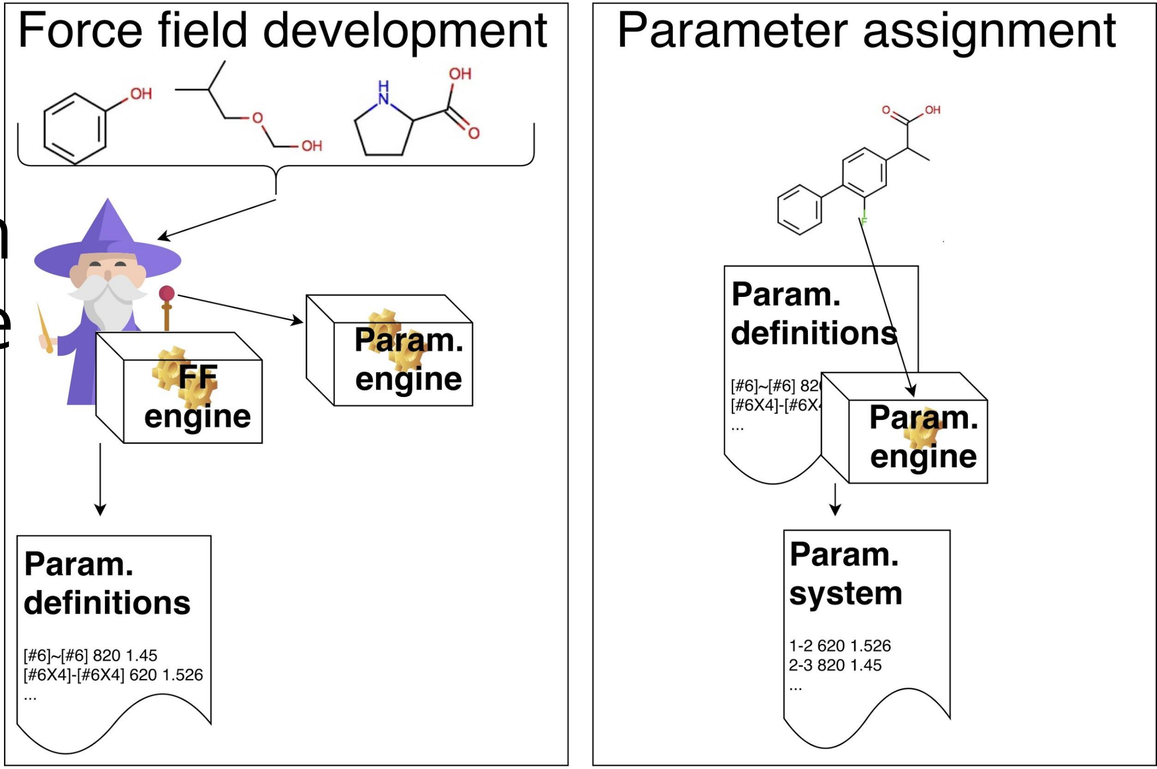
**#smirnoff** and **#smarty** on Slack

## Force fields typically rely on "indirect chemical perception", where atom typing is a design decision

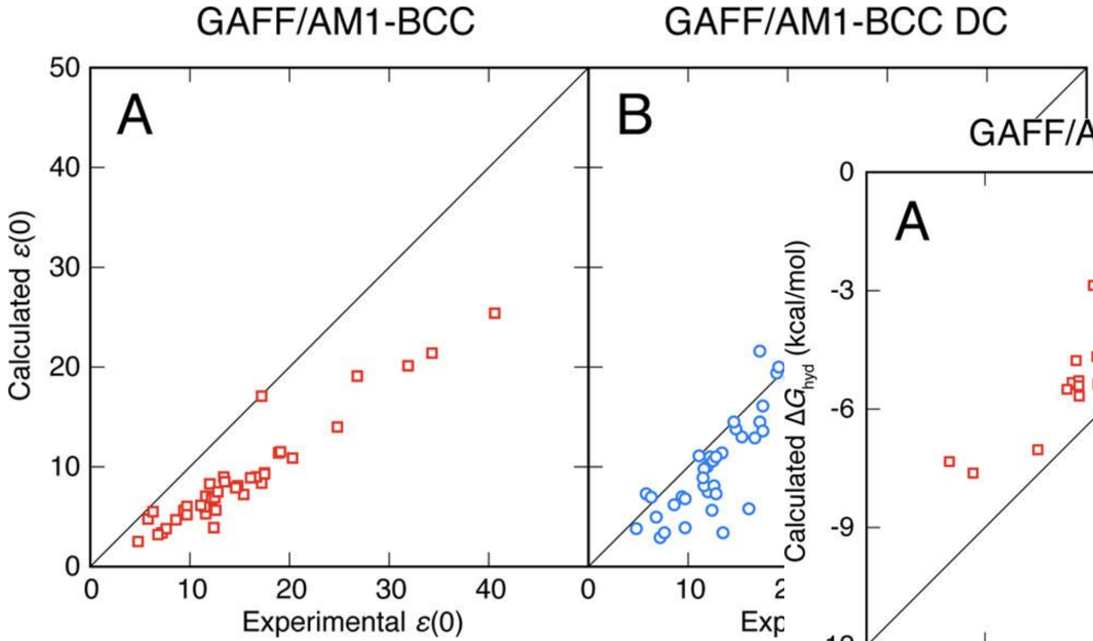


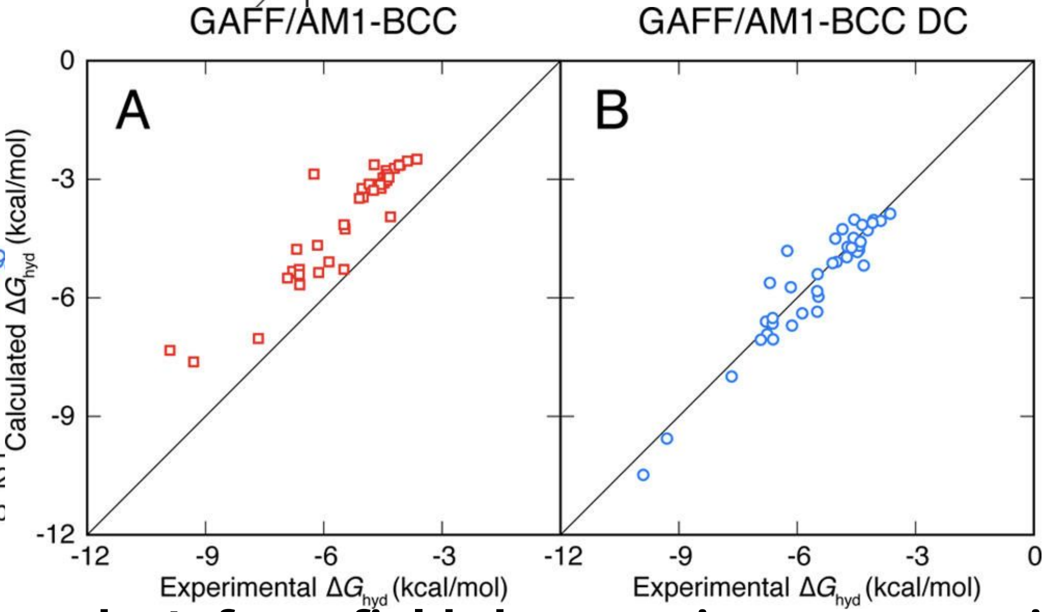
## We employ direct chemical perception

The distinction is important: All chemical information is still available to the parameterization engine



## Why new force fields? We already know we can do better than today's force fields without new physics

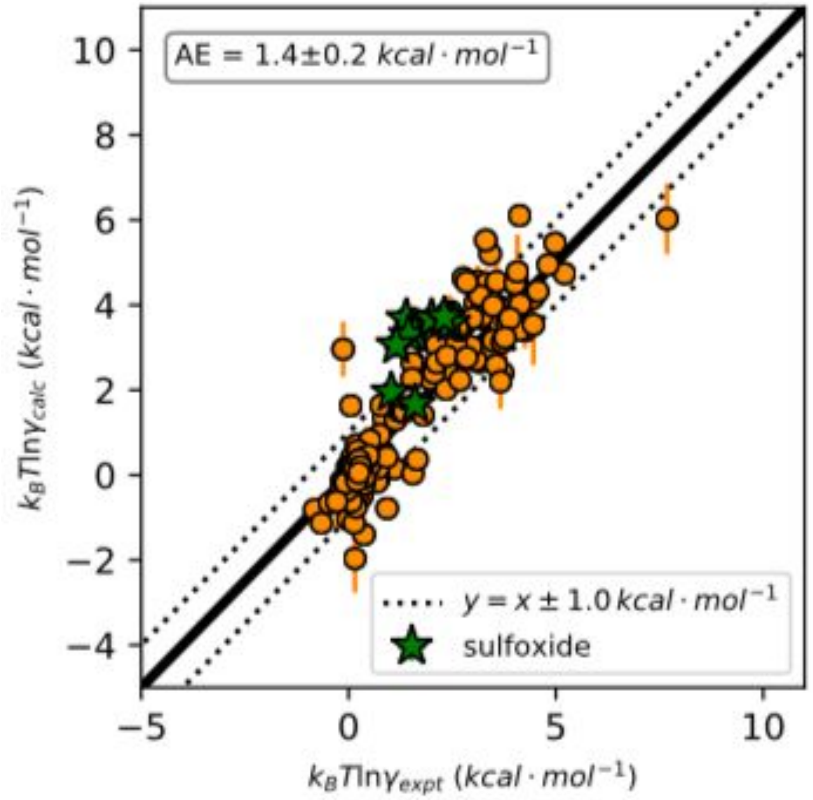




**Many functional groups in today's force fields have serious systematic errors that await a general fix, because refitting is too hard**

DOI: 10.1021/jp411529h

# Similar issues stare at us everywhere we look



https://doi.org/10.26434/chemrxiv.7199786.v2

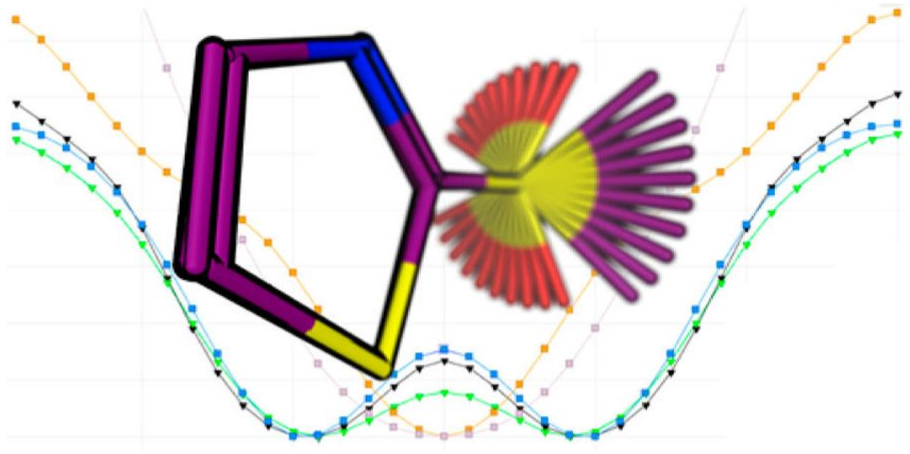
**Infinite dilution activity coefficients are untapped for force field development and show clear systematic errors**

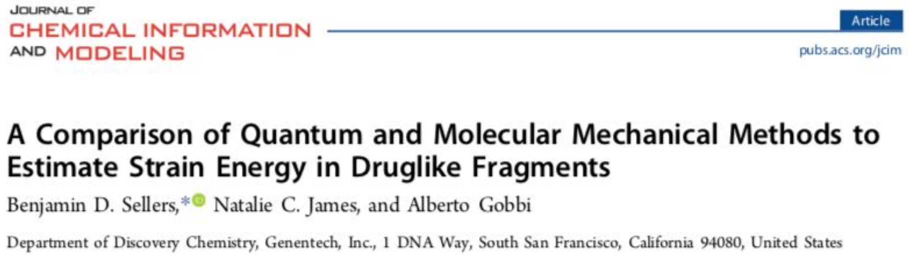
They inform on relative solvation in different solvents

Here, DMSO is poorly represented as a solvent

(Calculations of all the suitable activity coefficients from NIST's ThermoML done overnight on Orion)

# We all have our own war stories, too





JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

A Comparison of Quantum and Molecular Mechanical Methods to Estimate Strain Energy in Druglike Fragments

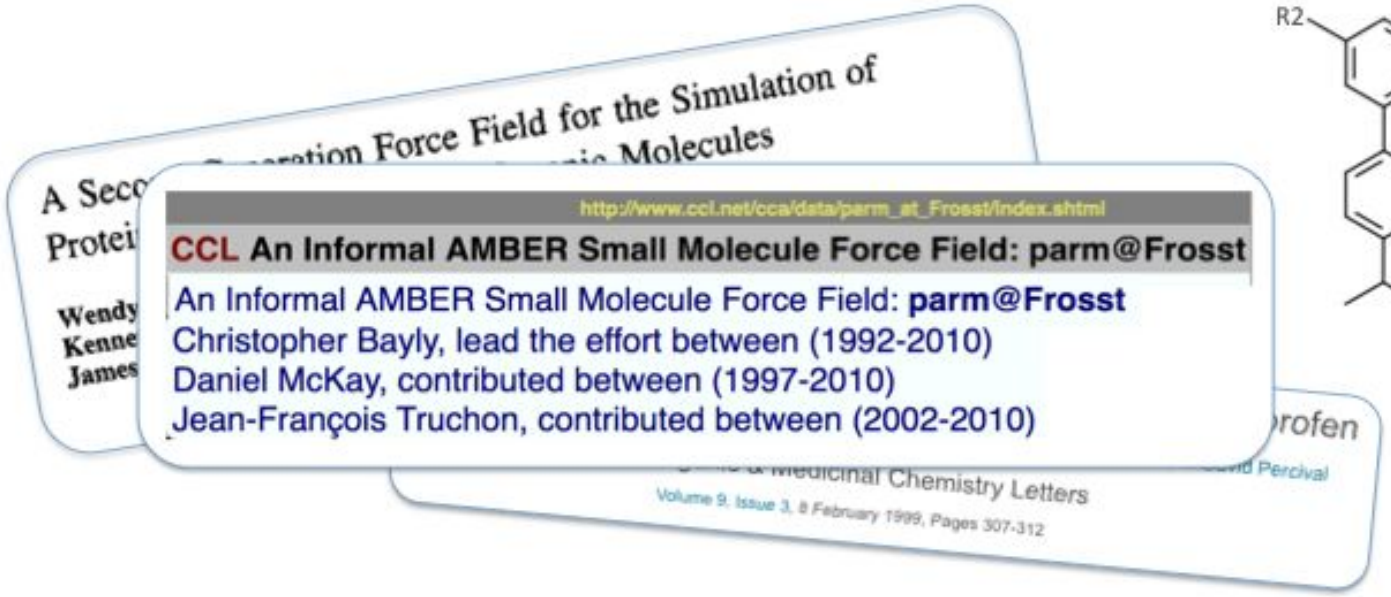
Benjamin D. Sellers,* Natalie C. James, and Alberto Gobbi

Department of Discovery Chemistry, Genentech, Inc., 1 DNA Way, South San Francisco, California 94080, United States

**Different force fields might not even agree on the location of a minimum**

DOI: 10.1021/acs.jcim.6b00614

# parm@frosst is the starting point for a GAFF-like small molecule force field using SMARTS/SMIRKS



A Seco ... Force Field for the Simulation of
Protei ... Molecules

Wendy
Kenne
James

http://www.ccl.net/cca/data/parm_at_Frosst/index.shtml

CCL An Informal AMBER Small Molecule Force Field: parm@Frosst

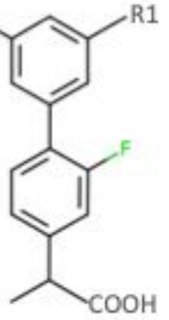
An Informal AMBER Small Molecule Force Field: **parm@Frosst**
Christopher Bayly, lead the effort between (1992-2010)
Daniel McKay, contributed between (1997-2010)
Jean-François Truchon, contributed between (2002-2010)

rofen

avid Percival

Medicinal Chemistry Letters

Volume 9, Issue 3, 8 February 1999, Pages 307-312



Bayly et al.’s parm@Frosst is an AMBER-family small molecule force field and sibling of GAFF

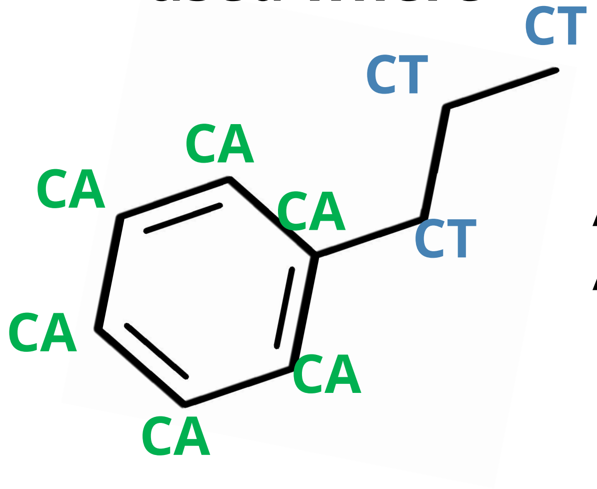
## So, how would we use SMIRKS for a force field? Let’s think of a carbon-carbon single bond

**[#6:1]-[#6:2]**, length=1.526 angstroms, force constant=620.0 kcal/(mol angstrom^2)

**Or, maybe we’d want a generic carbon-carbon bond:**
**[#6:1]~[#6:2]** with its own parameters

Perhaps a more specialized bond?
**[#6X3:1]=[#6X3:2]** with different parameters

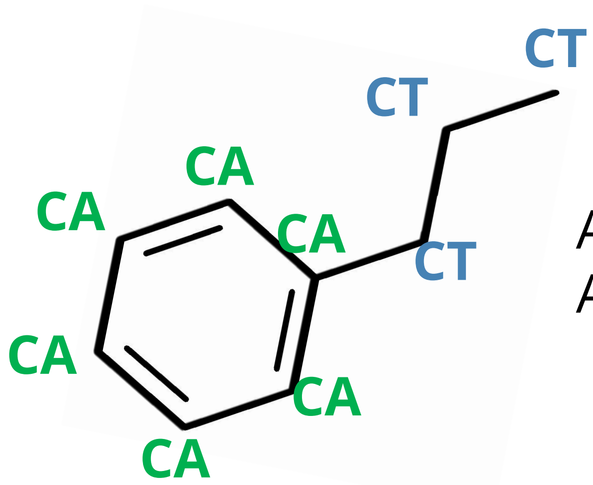
## Why is this a good thing? Let's think of atom typing or "chemical perception" which defines which parameters are used where



Aliphatic sp3 carbon (CT)
Aromatic sp2 carbon (CA)

| | |
|---|---|
| X -CT-CT-X | Low Barrier Torsion |
| X-CT-CA-X | Low Barrier Torsion |
| X -CA-CA-X | High Barrier Torsion |

## Today's force fields mostly use indirect chemical perception



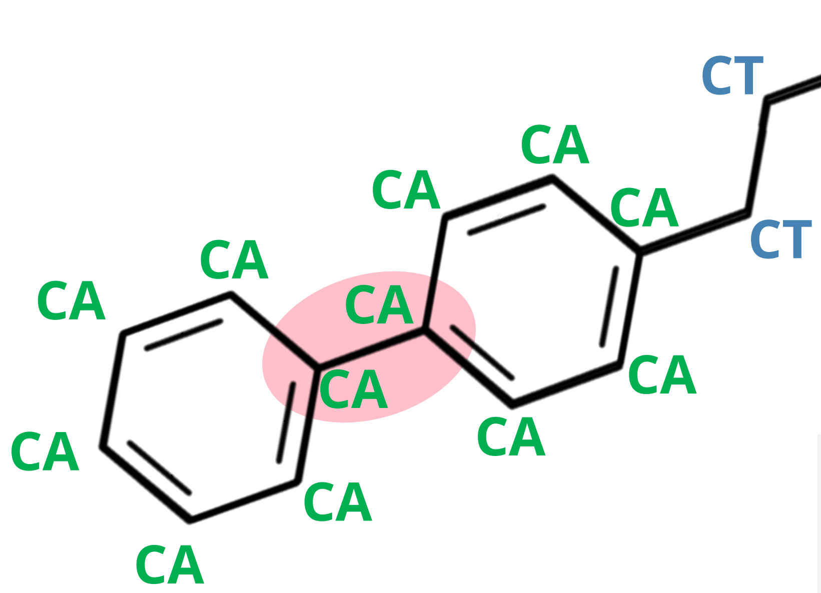
Aliphatic sp3 carbon (**CT**)
Aromatic sp2 carbon (**CA**)

Some tool (or human) assigns atom types

From the atom types, parameters are assigned

Thus, atom types must encode all requisite chemistry and can't be fitted as part of the process

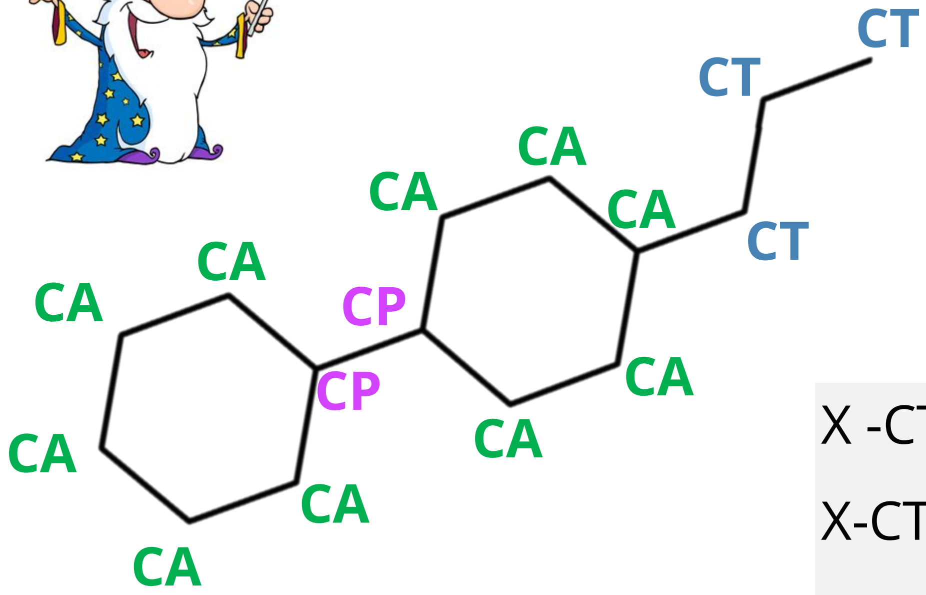
# This is a vital issue: Failing to capture the requisite chemistry leads to disaster



Aliphatic sp3 carbon (CT)
Aromatic sp2 carbon (CA)

| | |
|---|---|
| X -CT-CT-X | Low Barrier Torsion |
| X-CT-CA-X | Low Barrier Torsion |
| X -CA-CA-X | High Barrier Torsion |

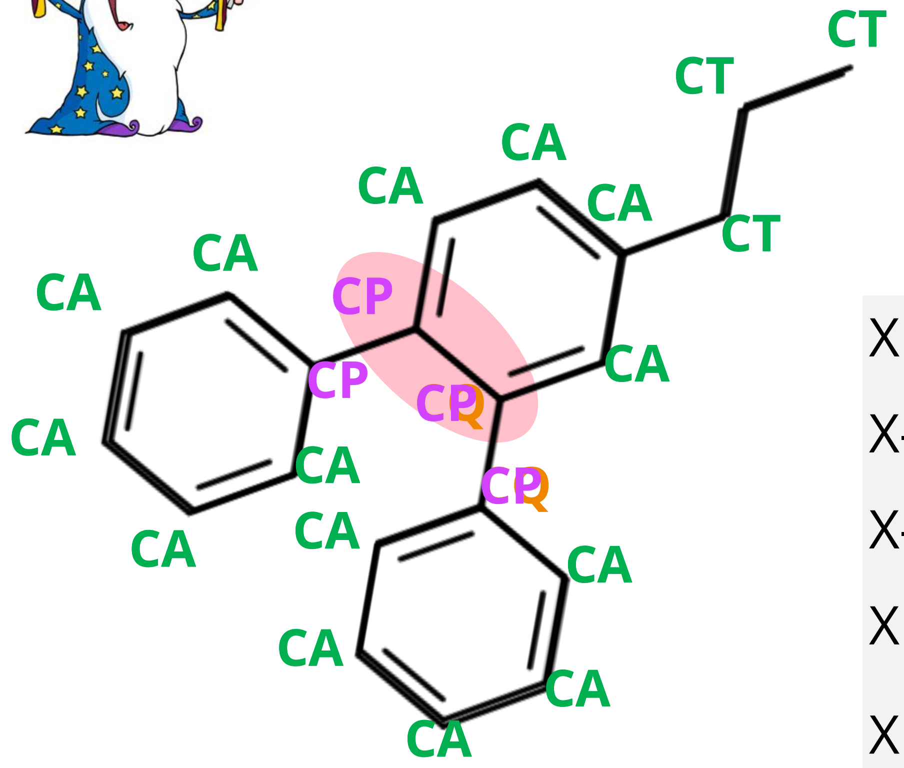
# One can fix this with more complex atom typing



Aliphatic sp3 carbon (CT)
Aromatic sp2 carbon (CA)
Aromatic carbon bridging phenyl rings (CP)

| | |
|---|---|
| X -CT-CT-X | Low Barrier Torsion |
| X-CT-CA-X | Low Barrier Torsion |
| X-CP-CP-X | Low Barrier Torsion |
| X -CA-CA-X | High Barrier Torsion |

# Though, new chemistry can still pose challenges so you need an atom type for every special case



Aliphatic sp3 carbon (**CT**)
Aromatic sp2 carbon (**CA**)
Aromatic carbon bridging phenyl rings (**CP**)

Same as CP for multiple bridges (**CQ**)

| | |
|---|---|
| X -CT-CT-X | Low Barrier Torsion |
| X-CT-CA-X | Low Barrier Torsion |
| X-CP-CP-X | Low Barrier Torsion |
| X -CA-CA-X | High Barrier Torsion |
| X -CA-CA-X | High Barrier Torsion |

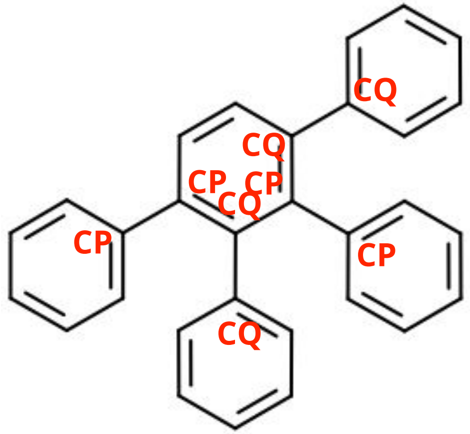
# This produces many redundant parameters

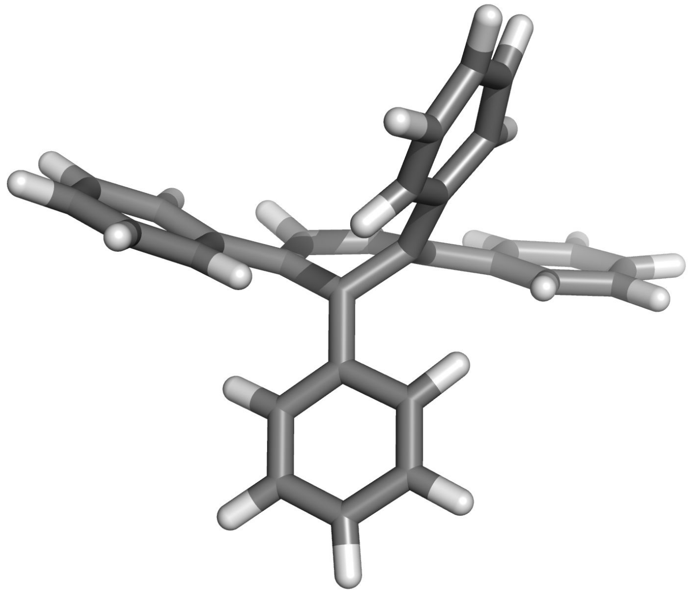
```
CA-CA-SO    70.000 120.000 force ff94 CA-CA-CT
CA-CA-SH    70.000 120.000 std aromatic
CA-CA-SD    70.000 120.000 std aromatic
CA-CA-S     70.000 120.000 std aromatic
CA-CA-P     70.000 120.000 std aromatic
CA-CA-OS    70.000 120.000 ** Gro, JACS,V111,2152('89)
CA-CA-OH    70.000 120.000 ff94 CA-C-OH
CA-CA-O2    70.000 120.000 guess March 5 2009 anionic O
CA-CA-NL    70.000 120.000 guess
CA-CA-ND    70.000 120.000 calc B3PW91/6-31+G** Jan 30 2002
CA-CA-NC    70.000 120.000 quinoline, ff94 CA-CA-CT
CA-CA-NB    70.000 120.000 guess april 11 2000
CA-CA-NA    70.000 120.000 ff94 CA-CA-CT
CA-CA-N3    70.000 120.000 guess april 11 2000
CA-CA-N2    70.000 120.000 ff94 CA-CA-CT
CA-CA-N*    70.000 120.000 ff94 std aromatic
CA-CA-N     70.000 120.000 ff94 CA-CA-CT cb 6jan97
CA-CA-I     70.000 120.000 std sp2 carbon aug 15 2001
CA-CA-F     70.000 120.000 ff94 CA-C-OH
CA-CA-Cl    70.000 120.000 ff94 CA-C-OH
CA-CA-CW    70.000 120.000 amidopyridine, ff94 CA-CA-CT
CA-CA-CR    70.000 120.000 amidopyridine, ff94 CA-CA-CT
```

parm@frosst has a few hundred lines of this type of redundancy

More than 60 identical parameters for CP alone

# Downstream problems persist to this day, even in GAFF and GAFF2





- Torsions within the ring end up getting X-CP-CP-X values (rotatable single bond) rather than X-CA-CA-X

## Ditching “atom types” for SMIRKS (“parameter types”) allows considerable simplification

For example, GAFF2 has 16 vdW types for carbon

| | | |
|---|---|---|
| c | 1.8606 | 0.0988 |
| cs | 1.8606 | 0.0988 |
| ca | 1.8606 | 0.0988 |
| cc | 1.8606 | 0.0988 |
| cd | 1.8606 | 0.0988 |
| ce | 1.8606 | 0.0988 |
| cf | 1.8606 | 0.0988 |
| cp | 1.8606 | 0.0988 |
| cq | 1.8606 | 0.0988 |
| cz | 1.8606 | 0.0988 |
| cu | 1.8606 | 0.0988 |
| cv | 1.8606 | 0.0988 |
| cg | 1.9525 | 0.1596 |
| ch | 1.9525 | 0.1596 |
| cx | 1.9069 | 0.1078 |
| cy | 1.9069 | 0.1078 |

But this should be three SMIRKS strings

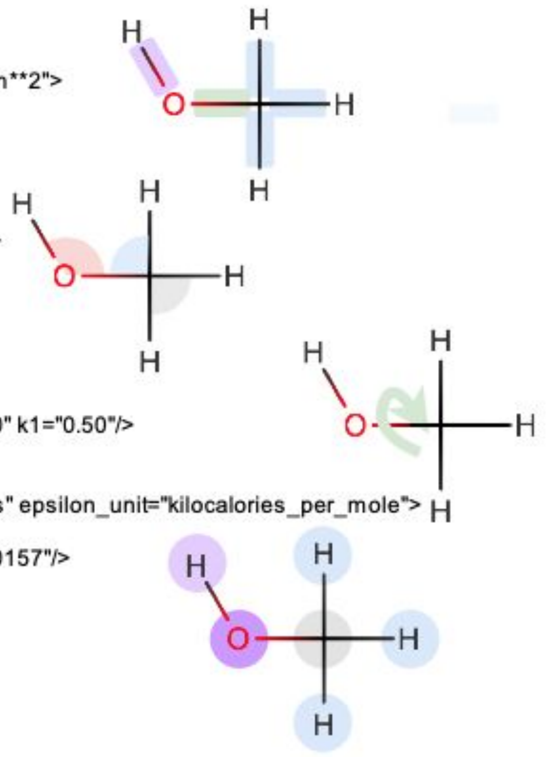
| | | |
|---|---|---|
| [#6:1] | 1.8606 | 0.0988 |
| [#6X1:1] | 1.9525 | 0.1596 |
| [#6X3r3,#6X3r4:1] | 1.9069 | 0.1078 |

Very relevant when attempting to automatically fit parameters — are there 32 parameters here, or 6?

(We would argue 6 — the atom types were introduced because of the need for angle or torsional complexity, usually)

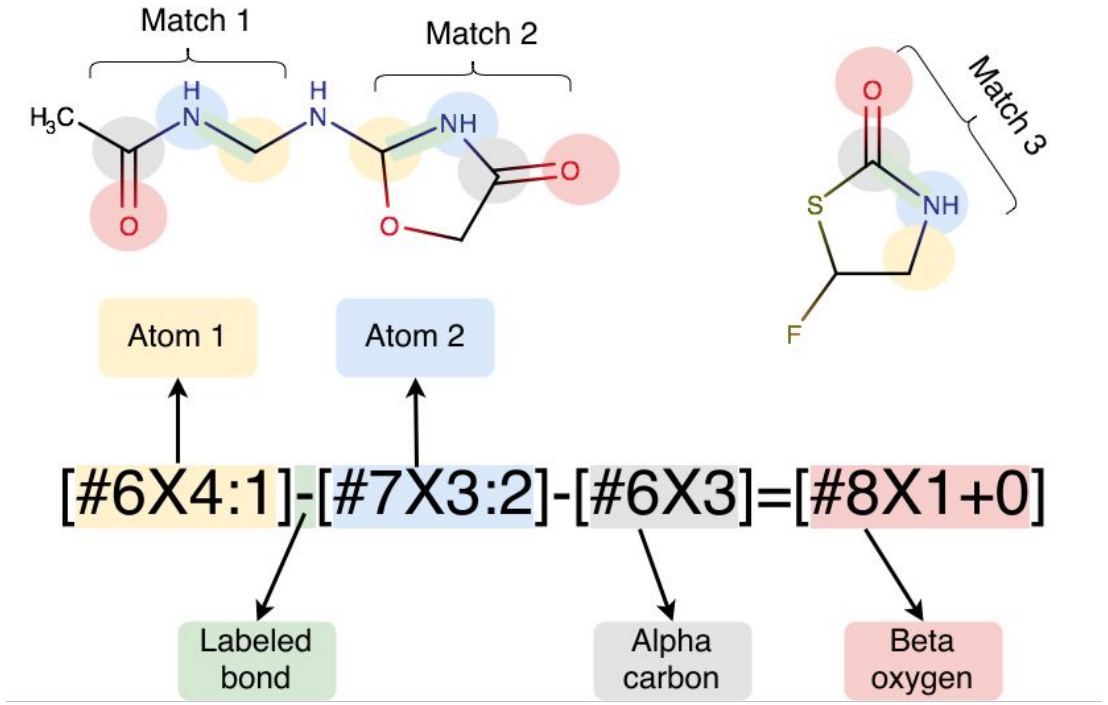
# SMIRNOFF parameters for methanol are simple

```
<?xml version="1.0"?>
<SMIRNOFF>

<HarmonicBondForce length_unit="angstroms" k_unit="kilocalories_per_mole/angstrom**2">
  <Bond smirks="[#6X4:1]-[#1:2]" length="1.090" k="680.0"/>
  <Bond smirks="[#6X4:1]-[#8&amp;X2&amp;H1:2]" length="1.410" k="640.0"/>
  <Bond smirks="[#8X2:1]-[#1:2]" length="0.960" k="1106.0"/>
</HarmonicBondForce>

<HarmonicAngleForce angle_unit="degrees" k_unit="kilocalories_per_mole/radian**2">
  <Angle smirks="[a,A:1]-[#6X4:2]-[a,A:3]" angle="109.50" k="100.0"/>
  <Angle smirks="[#1:1]-[#6X4:2]-[#1:3]" angle="109.50" k="70.0"/>
  <Angle smirks="[#6X4:1]-[#8X2:2]-[#1:3]" angle="108.50" k="110.0"/>
</HarmonicAngleForce>

<PeriodicTorsionForce phase_unit="degrees" k_unit="kilocalories_per_mole">
  <Proper smirks="[a,A:1]-[#6X4:2]-[#8X2:3]-[#1:4]" idivf1="3" periodicity1="3" phase1="0.0" k1="0.50"/>
</PeriodicTorsionForce>

<NonbondedForce coulomb14scale="0.833333" lj14scale="0.5" sigma_unit="angstroms" epsilon_unit="kilocalories_per_mole">
  <Atom smirks="[#1:1]" rmin_half="1.4870" epsilon="0.0157"/>
  <Atom smirks="[$([#1]-[#6]-[#7,#8,#9,#16,#17,#35]):1]" rmin_half="1.3870" epsilon="0.0157"/>
  <Atom smirks="[#1$(*-[#8]):1]" rmin_half="0.0000" epsilon="0.0000"/>
  <Atom smirks="[#6:1]" rmin_half="1.9080" epsilon="0.1094"/>
  <Atom smirks="[#8:1]" rmin_half="1.6837" epsilon="0.1700"/>
  <Atom smirks="[#8X2+0$(*-[#1]):1]" rmin_half="1.7210" epsilon="0.2104"/>
</NonbondedForce>

</SMIRNOFF>
```



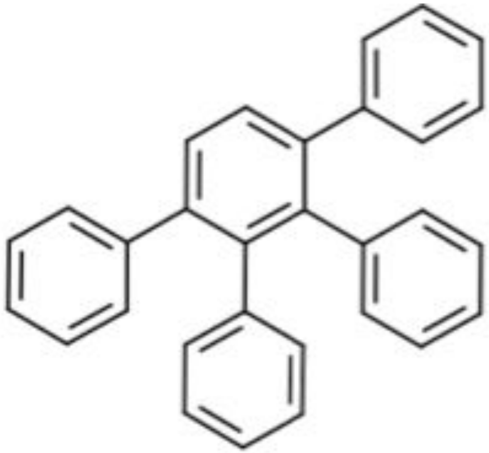
# Using SMARTS/SMIRKS allows us to escape atom typing

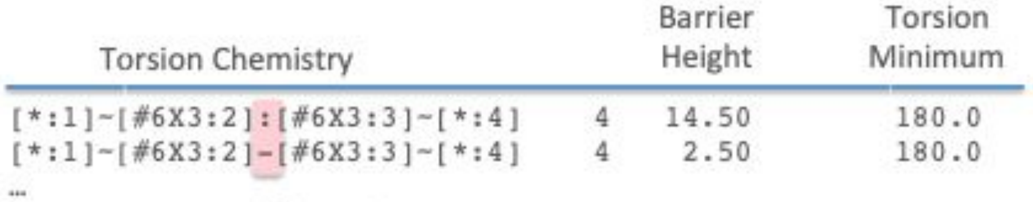
**We use substructure searches on the molecule to assign parameters, rather than atom typing**



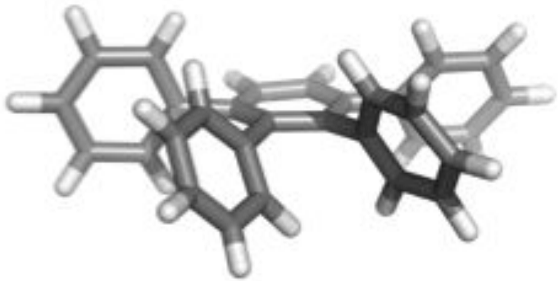
https://github.com/openforcefield/openforcefield/blob/master/The-SMIRNOFF-force-field-format.md

# Direct chemical perception utilizes bond order



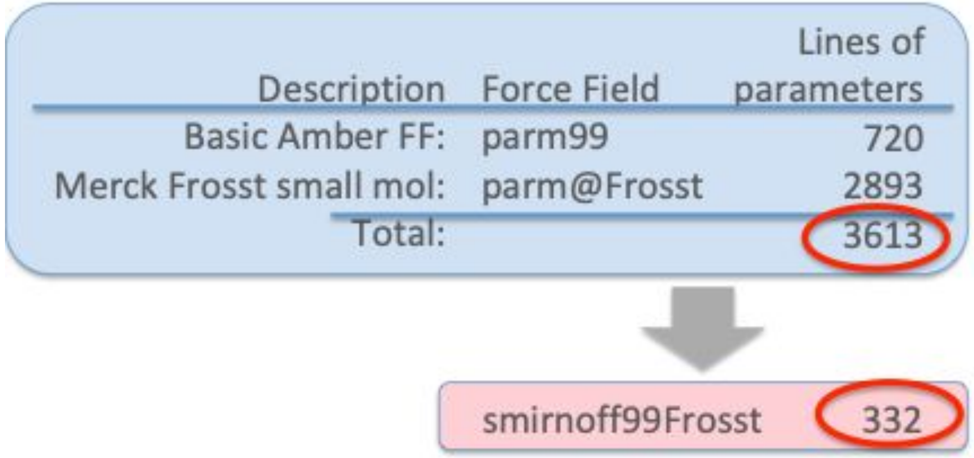


| Torsion Chemistry | | Barrier Height | Torsion Minimum |
|---|---|---|---|
| [*:1]~[#6X3:2]:[#6X3:3]~[*:4] | 4 | 14.50 | 180.0 |
| [*:1]~[#6X3:2]-[#6X3:3]~[*:4] | 4 | 2.50 | 180.0 |
| … | | | |



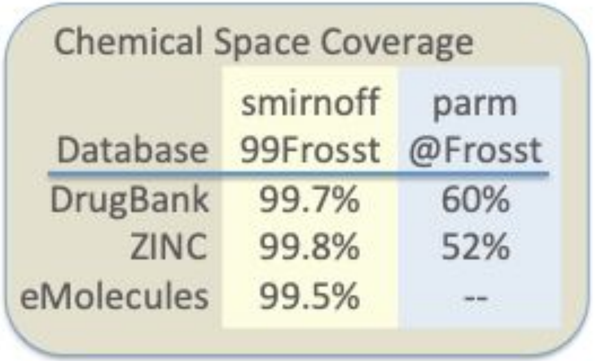
**We get the geometry right with no special treatment and far fewer parameters**

## smirnoff99Frosst is our adaptation of parm99+parm@Frosst into this format



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

| | smirnoff99Frosst | 332 |
|---|---|---|

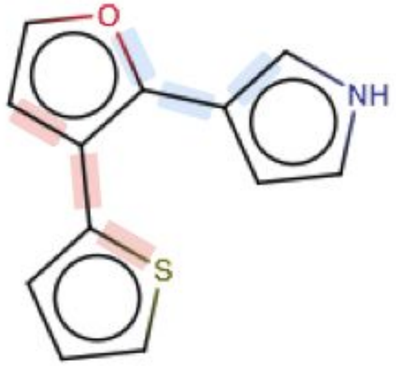


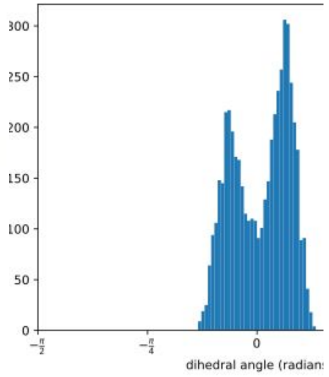
Chemical Space Coverage

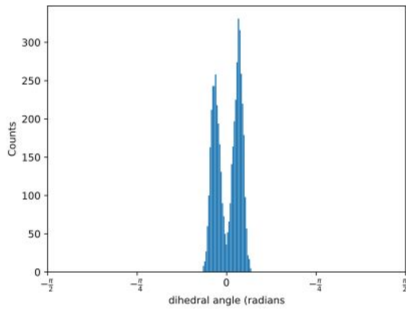
| Database | smirnoff 99Frosst | parm @Frosst |
|---|---|---|
| DrugBank | 99.7% | 60% |
| ZINC | 99.8% | 52% |
| eMolecules | 99.5% | -- |

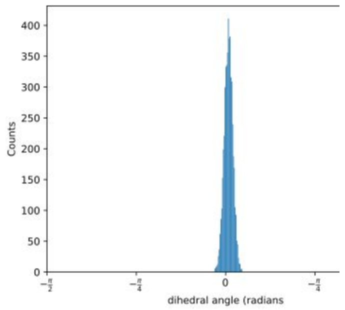
- Less than 1/10 the size of the original force field
- Removes redundancy
- Almost completely covers pharmaceutical chemical space

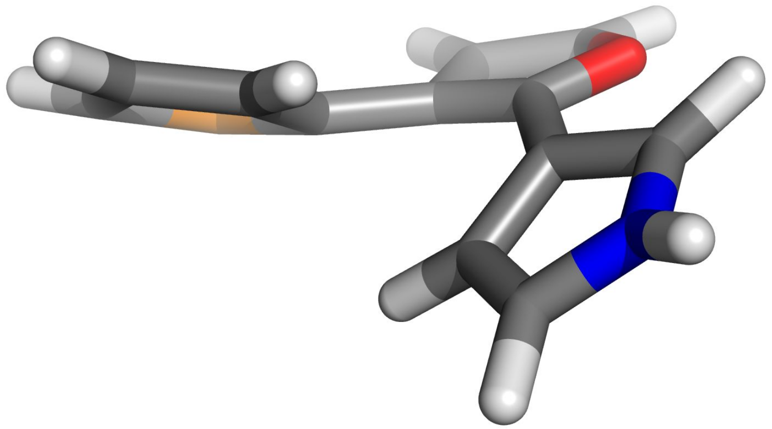
# Out of the box it fixes a variety of problems, including siblings of the biphenyl problem





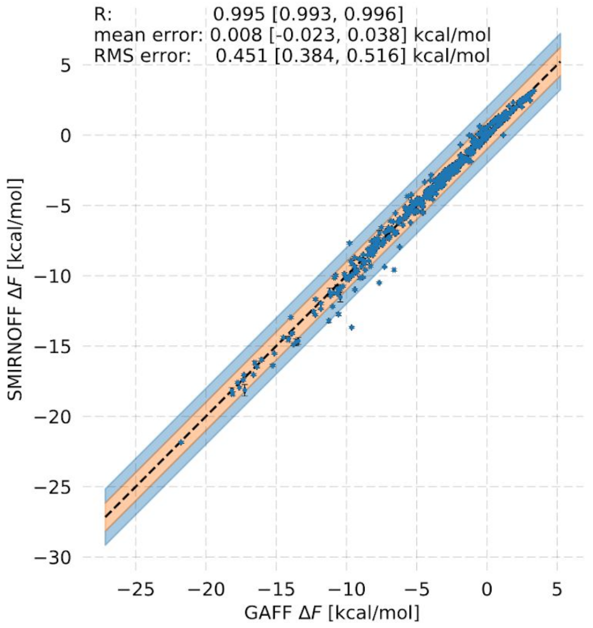


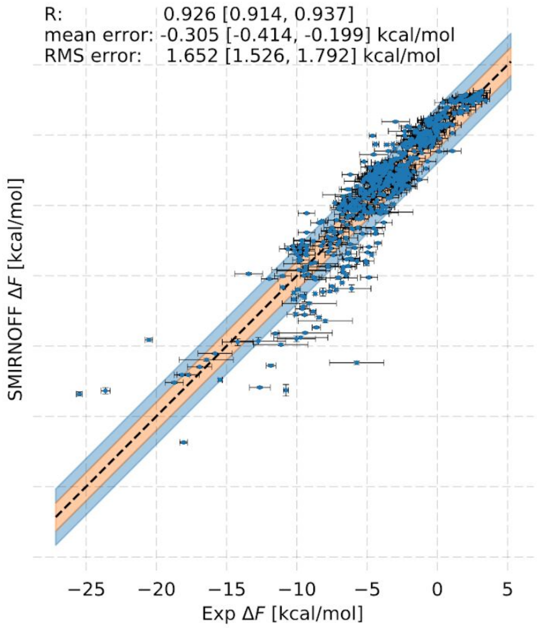


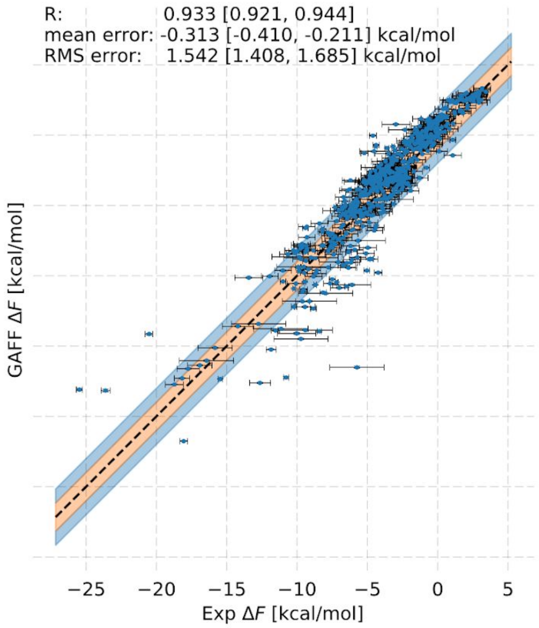


# We think smirnoff99Frosst is a great starting point for parameterization work

FreeSolv hydration free energy benchmark





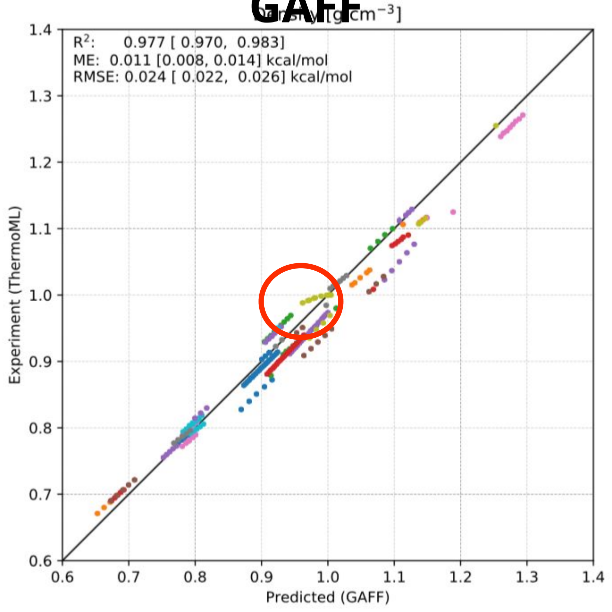


http://github.com/openforcefield/smirnoff99Frosst

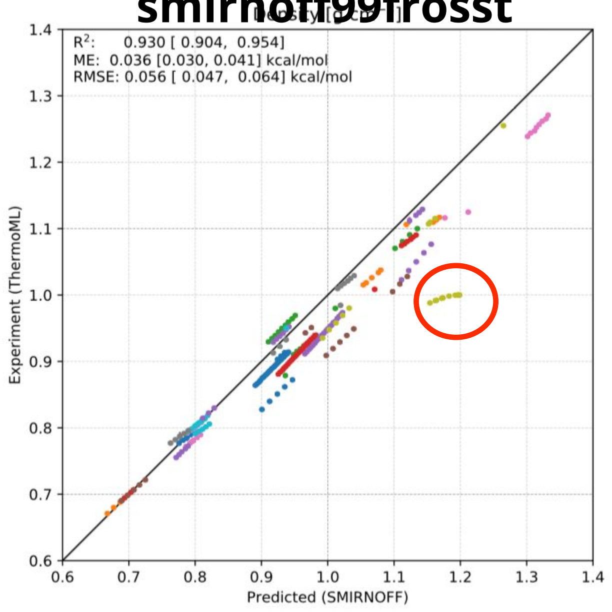
# It is competitive with GAFF but with far fewer parameters

**ThermoML Archive density benchmark set**

**GAFF**



**smirnoff99frosst**



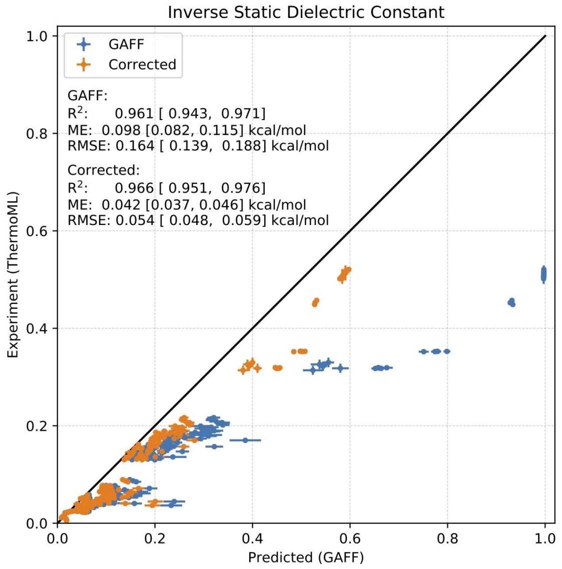
water

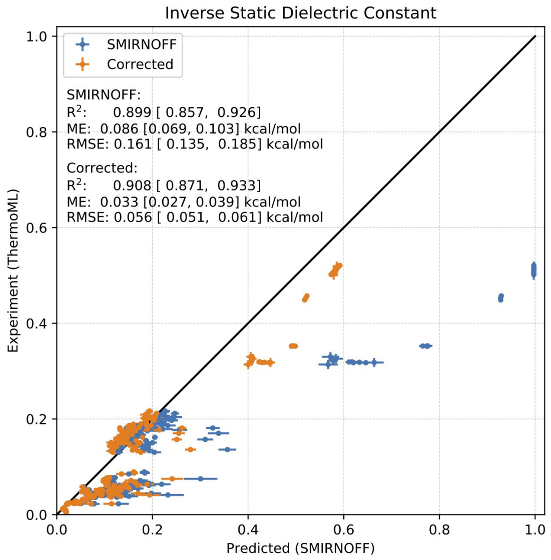
http://github.com/openforcefield/smirnoff99Frosst

# It is competitive with GAFF but with far fewer parameters

**ThermoML Archive dielectric benchmark set**

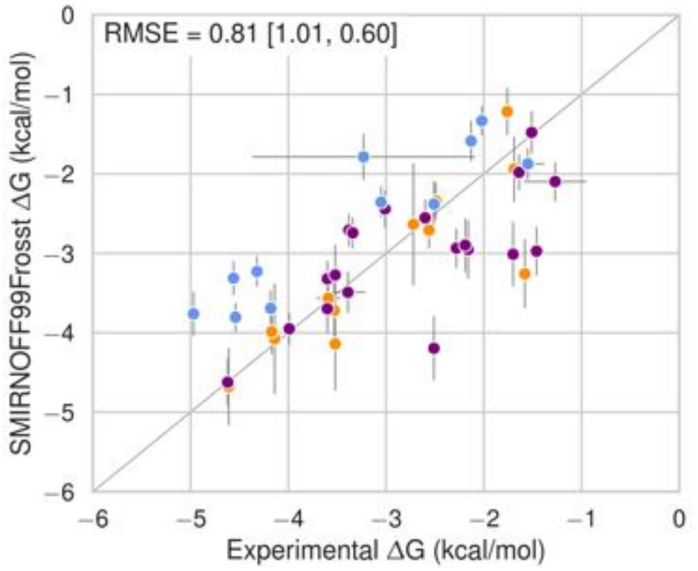
**GAFF** **smirnoff99frosst**

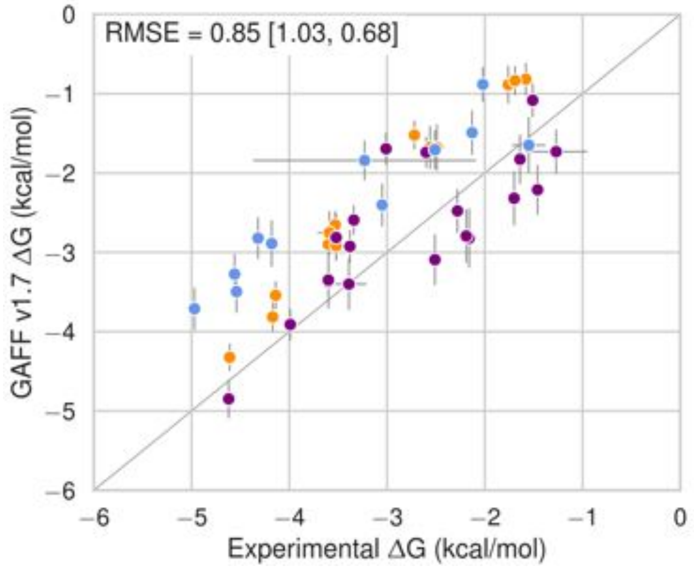




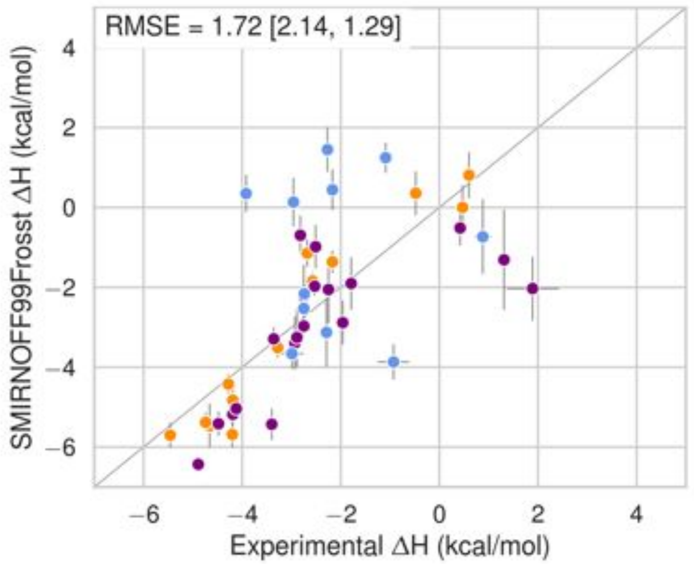
[http://github.com/openforcefield/smirnoff99Frosst](http://github.com/openforcefield/smirnoff99Frosst)

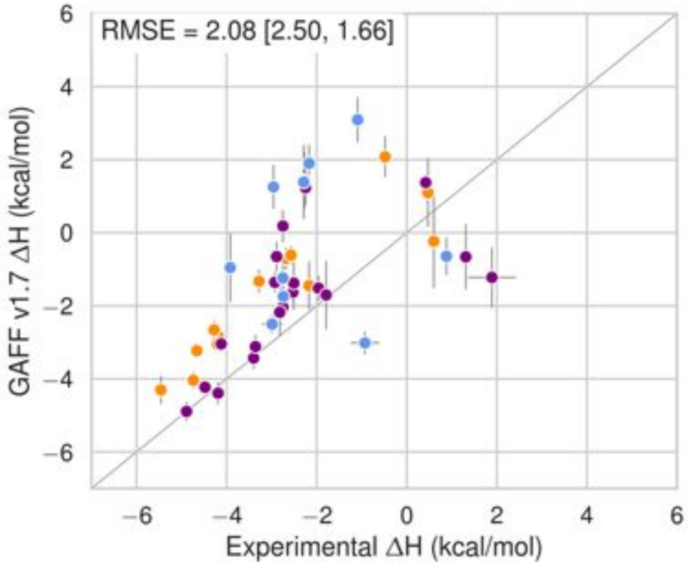
## We’re also interested in using binding data for fitting, and it turns out to do fairly well on host-guest binding also



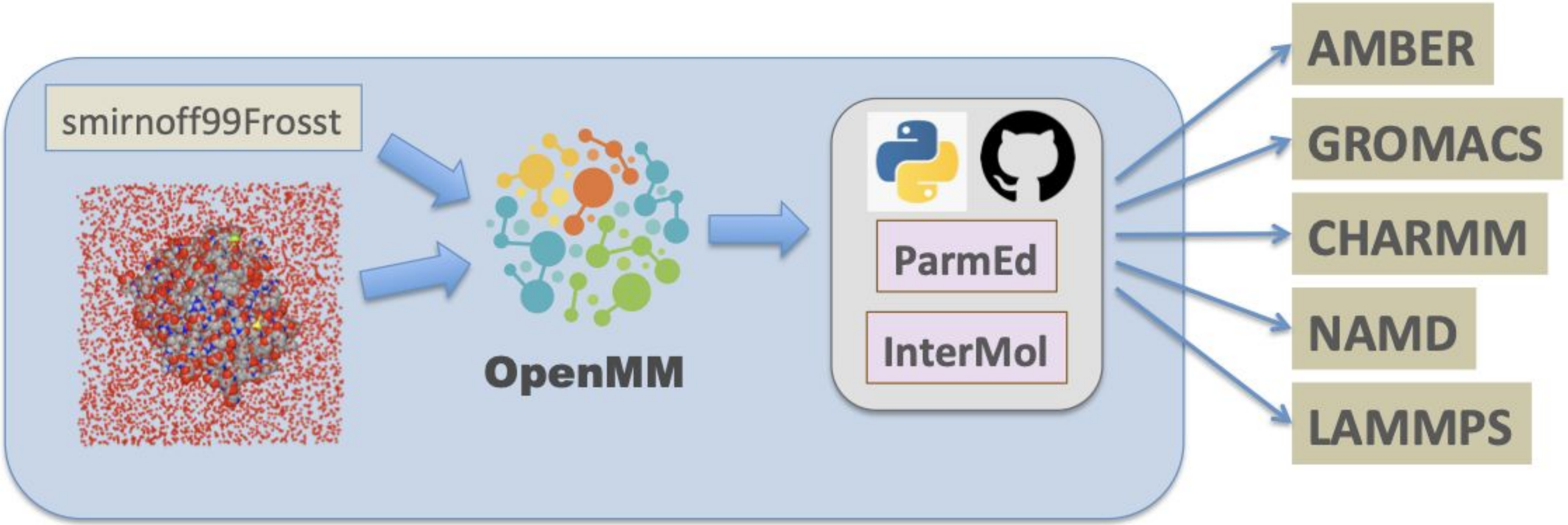


# It even works fairly well on enthalpies of binding





# You can use smirnoff99Frosst today!



https://github.com/openforcefield/smirnoff99Frosst

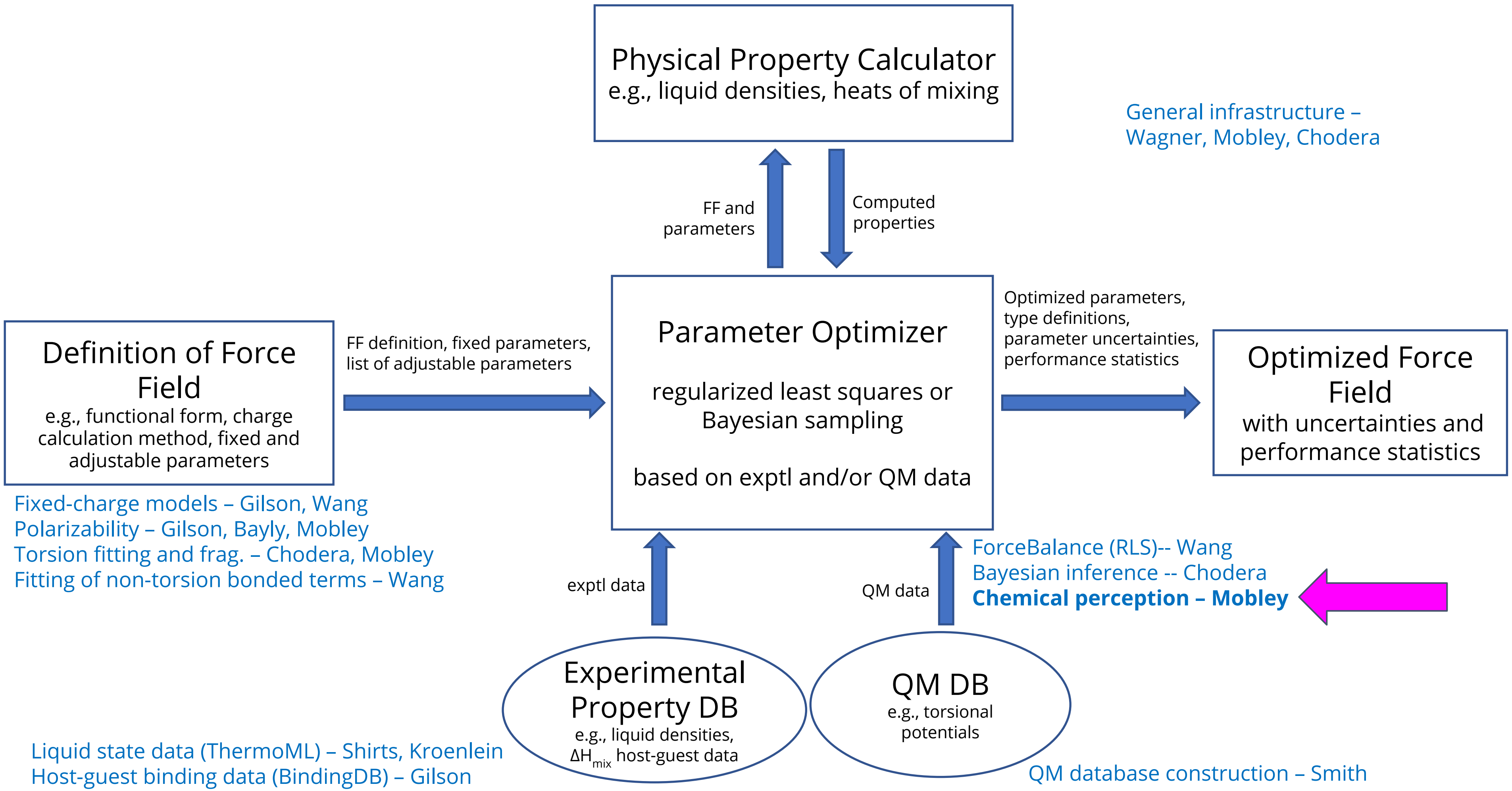
https://github.com/openforcefield/openforcefield

**See Slack #onboarding** and examples in github.com/openforcefield/openforcefield

## Currently, the SMIRNOFF SMIRKS are written by hand

```
"[*:1]~[#6:2]-[#6:3]~[*:4]”       Low Barrier Torsion
"[*:1]~[#6X3:2]:[#6X3:3]~[*:4]”  High Barrier Torsion
```

Chodera/Shirts/Gilson/Boothroyd


Physical Property Calculator
e.g., liquid densities, heats of mixing
FF and parameters
Computed properties
Definition of Force Field
e.g., functional form, charge calculation method, fixed and adjustable parameters
FF definition, fixed parameters, list of adjustable parameters
Parameter Optimizer
regularized least squares or Bayesian sampling
based on exptl and/or QM data
Optimized parameters, type definitions, parameter uncertainties, performance statistics
Optimized Force Field
with uncertainties and performance statistics
exptl data
QM data
Experimental Property DB
e.g., liquid densities, $\Delta H_{mix}$ host-guest data
QM DB
e.g., torsional potentials

General infrastructure – Wagner, Mobley, Chodera

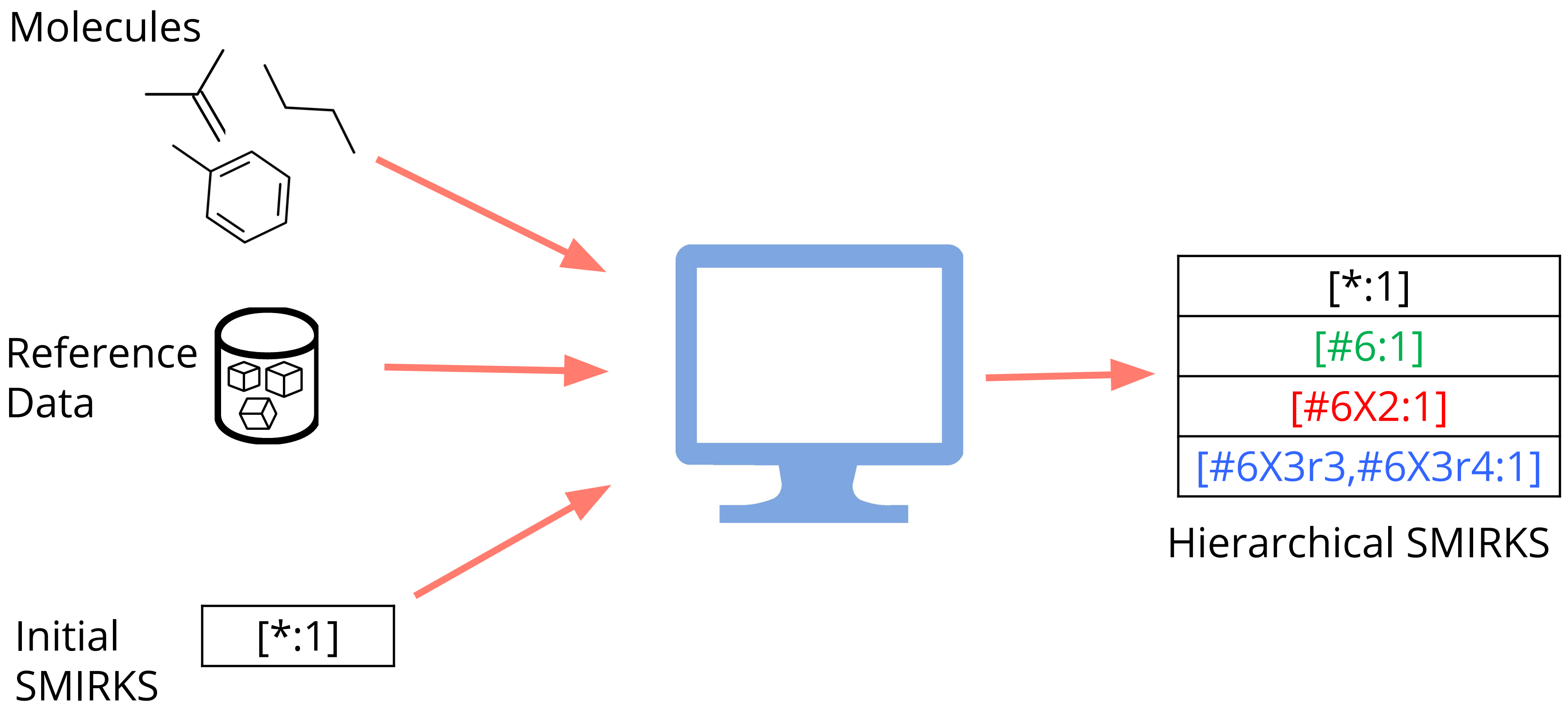
Fixed-charge models – Gilson, Wang
Polarizability – Gilson, Bayly, Mobley
Torsion fitting and frag. – Chodera, Mobley
Fitting of non-torsion bonded terms – Wang

ForceBalance (RLS)-- Wang
Bayesian inference -- Chodera
**Chemical perception – Mobley**

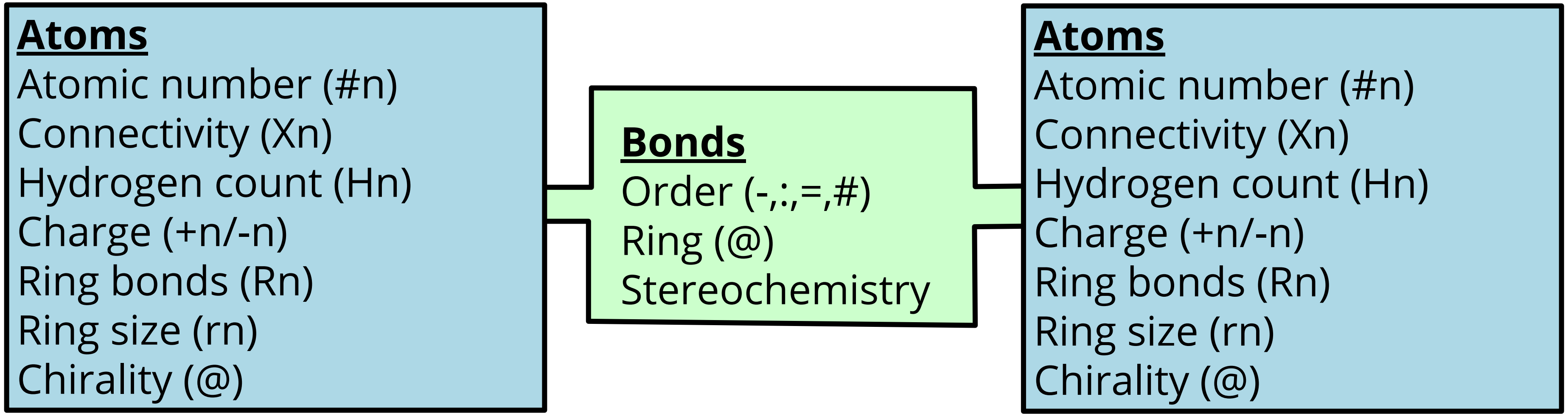
Liquid state data (ThermoML) – Shirts, Kroenlein
Host-guest binding data (BindingDB) – Gilson

QM database construction – Smith

## Chemical perception choices should be data-driven



# SMIRKS have simple components, but are a complex language

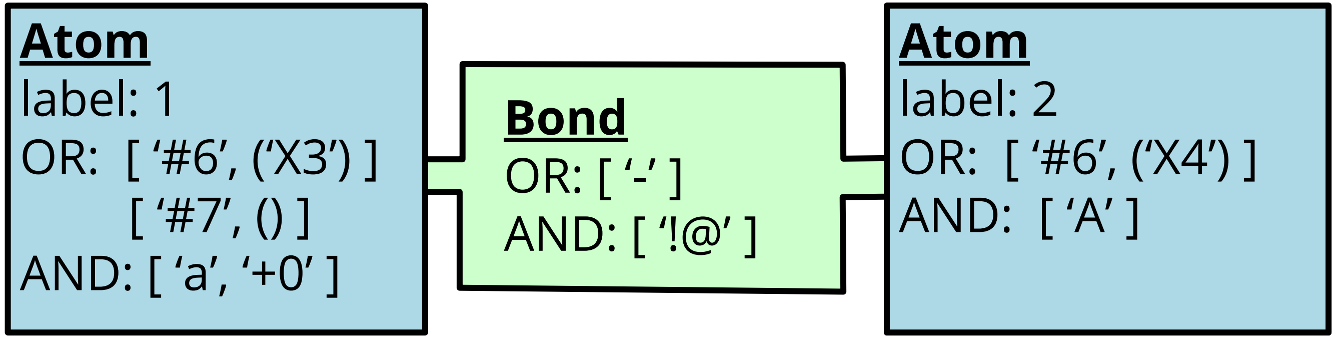


```
“[#6,#“[C,N]-;@[C,N]”;;A:2]”
```

# We needed a tool to edit SMIRKS by component

“[#6X3,#7;a;+0:1]-;!@[#6;A:2]” → “[#6X3,#7;a;+0:1]-;!@[#6X4;A:2]”

ChemicalEnvironment objects

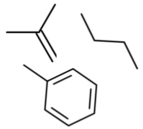


- store SMIRKS information by atom and bond
- allow for decoration changes
- are input and output as SMIRKS patterns

# SMIRKY: a Monte Carlo algorithm for learning SMIRKS in a SMIRNOFF

1. Chose a random SMIRKS
2. (a) Create a new SMIRKS by randomly changing this or (b) Delete this SMIRKS
3. Compute a change in “score” based on similarity to an existing FF
4. Use the Metropolis criterion to accept or reject the proposed move

Reference Data: SMIRNOFF99Frosst



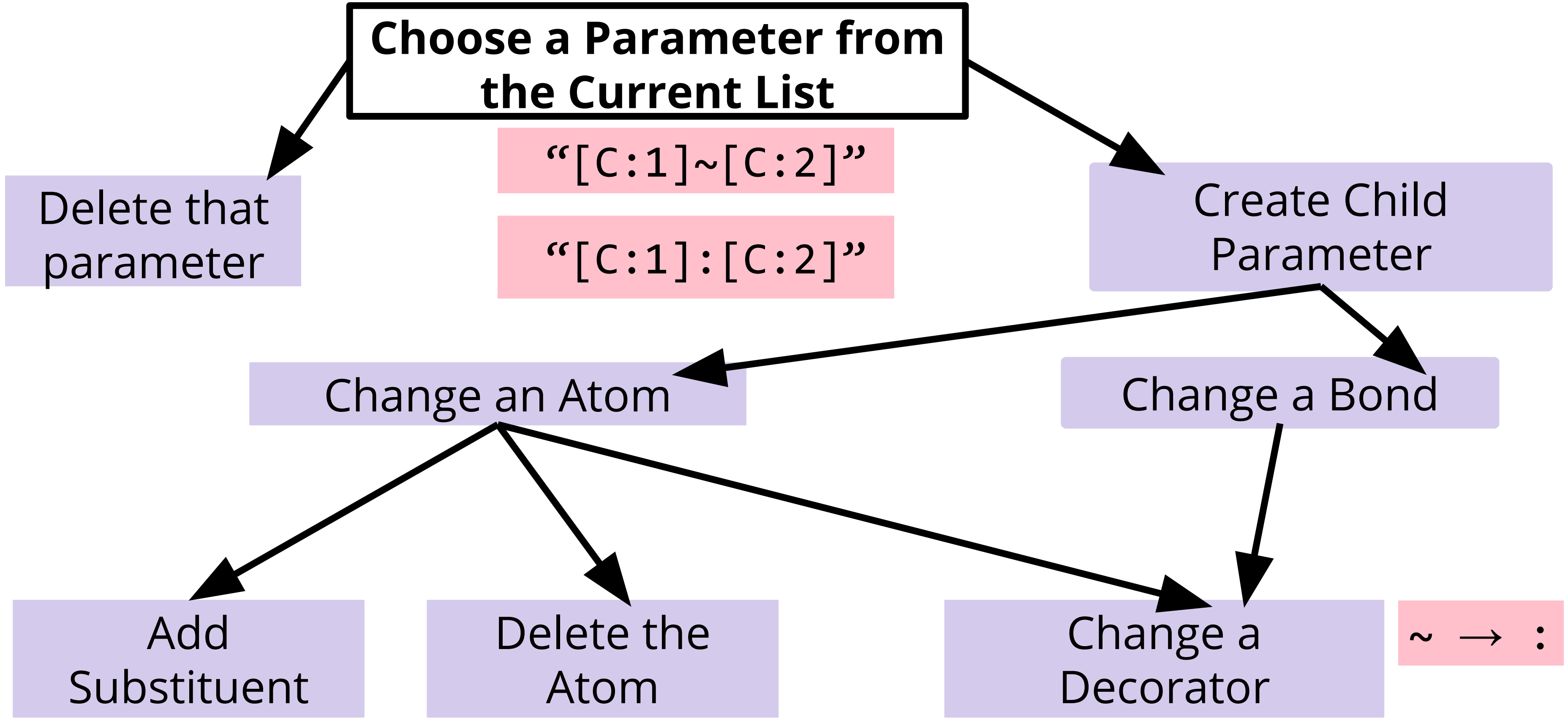
Molecules: 3 sets of increasing chemical complexity

[*:1]

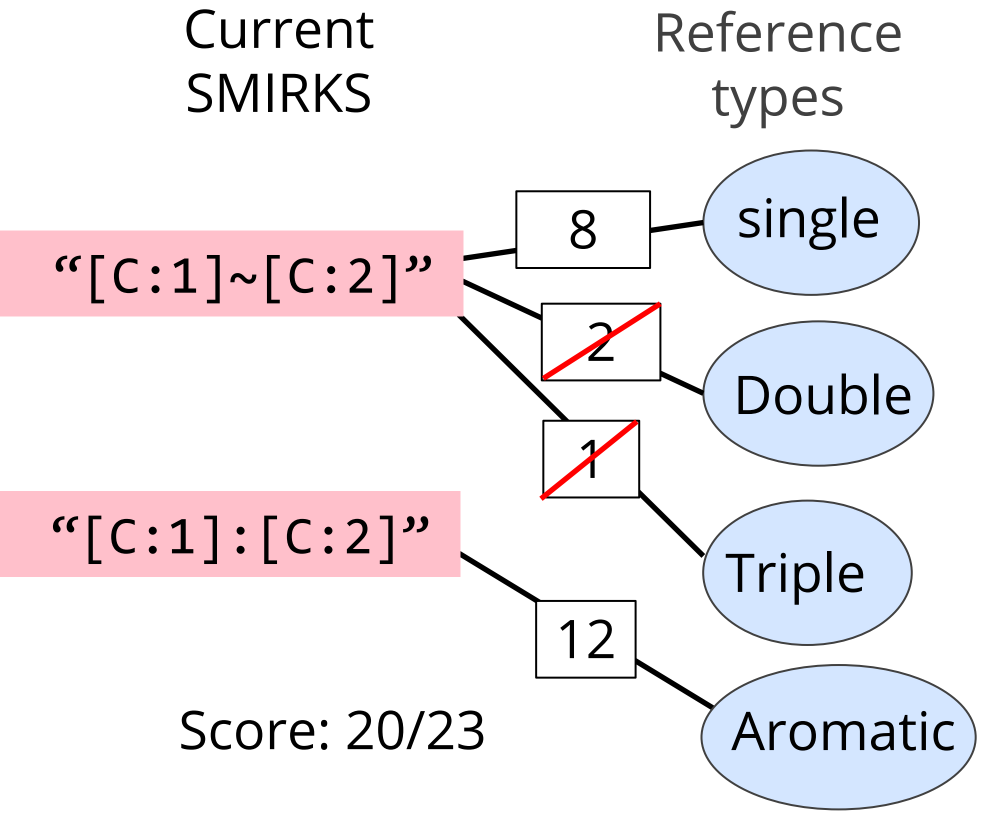
Initial SMIRKS: Simple based on parameter type

Zanette, et al. doi:10.1021/acs.jctc.8b00821

# We created a move set to create SMIRKS for chemical environment



# SMIRKS sets are scored based on typed molecules

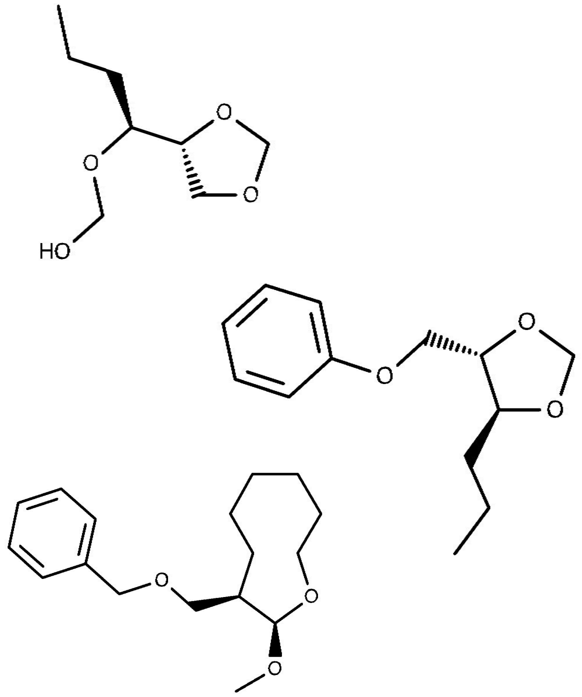


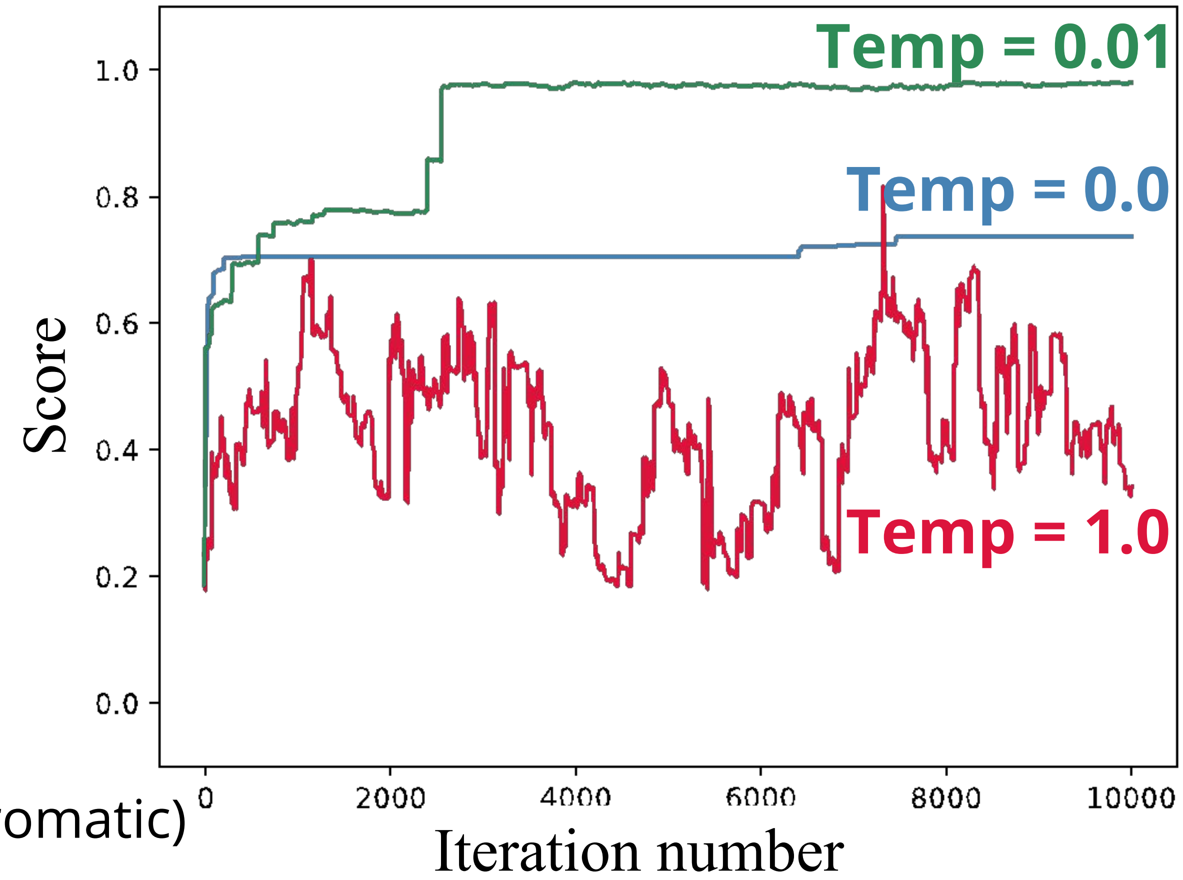
Score: 20/23

Accepted if $R < \exp\left[\frac{S_{curr} - S_{prev}}{T}\right]$

R = Random[0,1]
S = Score
T = Temperature

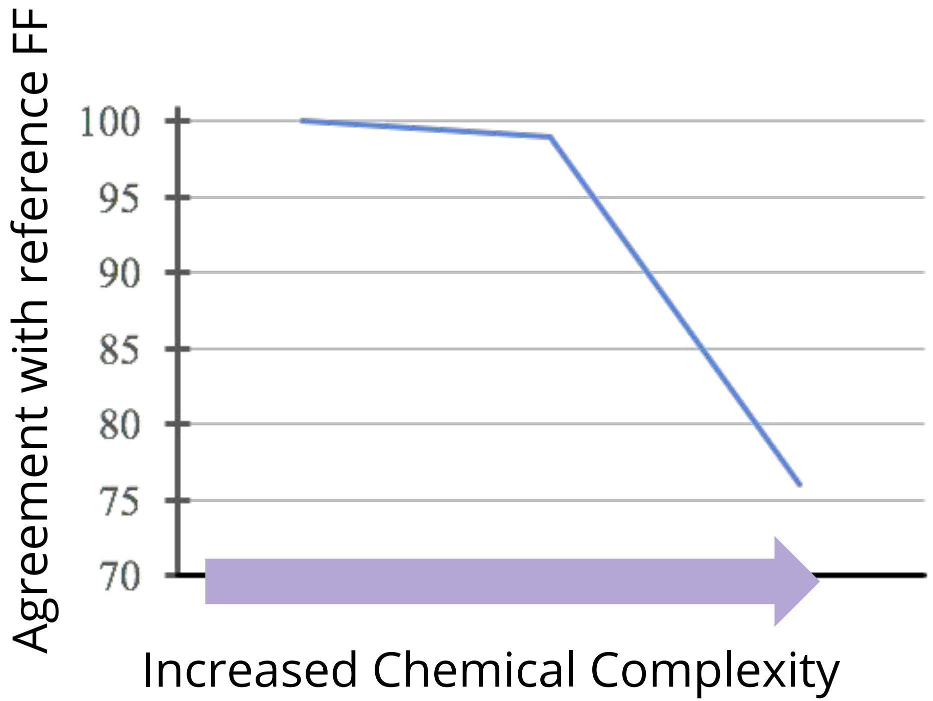
# We can find SMIRNOFF parameters in a simple molecule sets





200 molecules with oxygen, hydrogen, and carbon (aliphatic and aromatic)

# Realistic molecule sets were less successful due to the large combinatorial space



How long to make this Target SMIRKS pattern

[*:1]-,:[#6X3:2]=[#7X2:3]-[*:4]

1. Pick the right starting SMIRKS

   [*:1]~[#6:2]~[#7:3]~[*:4]

   Probability ~ 0.02

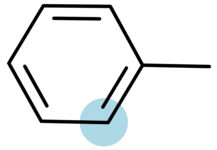
2. Right decorator to the right atom

   Probability ~ 0.01

**Approximately 1,000,000,000 iterations**

# Important SMIRKY lessons for moving forward

- Naive moves in SMIRKS space waste time and are unnecessary



#6,X3,H1,+0,R2,r6

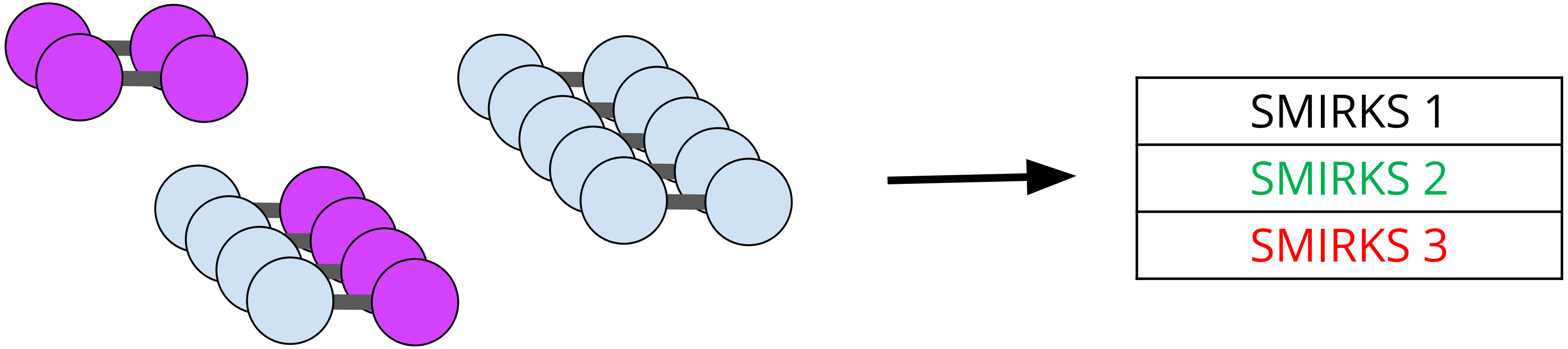
- SMIRKY was built exclusively for comparison with reference force fields

Reference Data and scoring

SMIRKS Generation

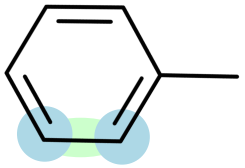
Input Molecules

# ChemPer extracts decorators to make SMIRKS from input molecular fragment

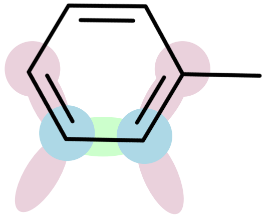


- How does ChemPer create SMIRKS patterns
- Challenges addressed in this package
- How to generate SMIRKS from reference QM data

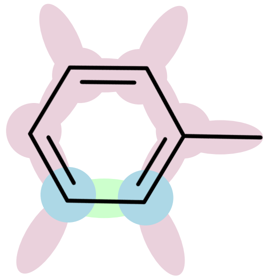
# SMIRKS are generated starting with the indexed atoms



```
'[#6aH1X3x2r6+0:1]:@[#6aH1X3x2r6+0:2]'
```



```
'[#6aH1X3x2r6+0:1](-!@[#1AH0X1x0!r+0])(:@[#6aH1X3x2r6+0]):@
[#6aH1X3x2r6+0:2](-!@[#1AH0X1x0!r+0]):@[#6aH1X3x2r6+0]'
```



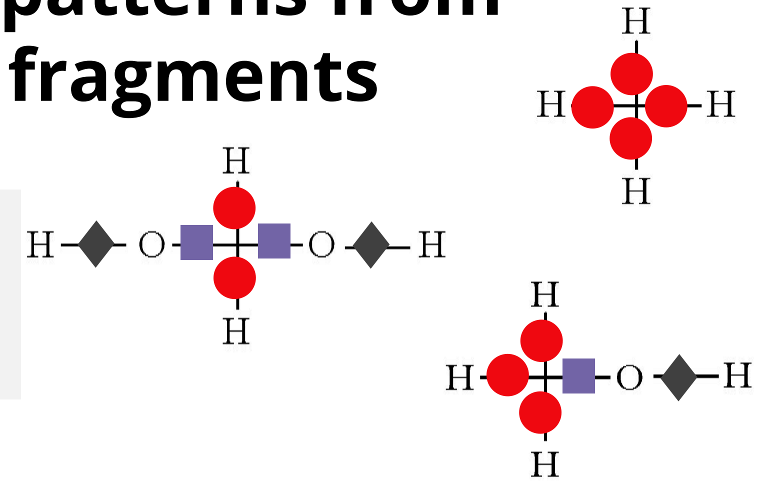
```
'[#6aH1X3x2r6+0:1](-!@[#1AH0X1x0!r+0])(:@[#6aH1X3x2r6+0]
(-!@[#1AH0X1x0!r+0]):@[#6aH0X3x2r6+0](-!@[#6AH3X4x0!r+0]
(-!@[#1AH0X1x0!r+0])(-!@[#1AH0X1x0!r+0])-!@[#1AH0X1x0!r+0]):@
[#6aH1X3x2r6+0](-!@[#1AH0X1x0!r+0]):@[#6aH1X3x2r6+0]-!@
[#1AH0X1x0!r+0]):@[#6aH1X3x2r6+0:2]-!@[#1AH0X1x0!r+0]'
```

# ChemPer: Builds SMIRKS patterns from clusters of molecular fragments

● <Bond smirks="[#6X4:1]-[#6X4:2]" k="620.0" length="1.526"/>

■ <Bond smirks="[#6:1]-[#8:2]" k="640.0" length="1.410"/>

◆ <Bond smirks="[#8:1]-[#1:2]" k="1106.0" length="0.960"/>



● [#6AH2X4x0!r+0,#6AH3X4x0!r+0,#6AH4X4x0!r+0:1]-;!@[#1AH0X1x0!r+0:2]

■ [#6AH2X4x0!r+0,#6AH3X4x0!r+0:1]-;!@[#8AH1X2x0!r+0:2]

◆ [#8AH1X2x0!r+0:1]-;!@[#1AH0X1x0!r+0:2]

## These SMIRKS are generalized by removing unnecessary decorators

● [#6AH2X4x0!r+0,#6AH3X4x0!r+0,#6AH4X4x0!r+0:1]-;!@[#1AH0X1x0!r+0:2]

■ [#6AH2X4x0!r+0,#6AH3X4x0!r+0:1]-;!@[#8AH1X2x0!r+0:2]

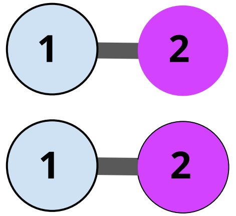
◆ [#8AH1X2x0!r+0:1]-;!@[#1AH0X1x0!r+0:2]

**Iteratively remove unnecessary decorators while maintaining clustering**

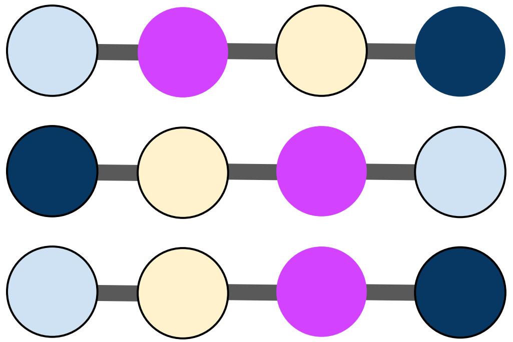
| ● | ■ | ◆ |
|---|---|---|
| [#6:1]-[#6:2] | [#6:1]-[#8:2] | [#8:1]-[#1:2] |

# Fragments need to be properly aligned

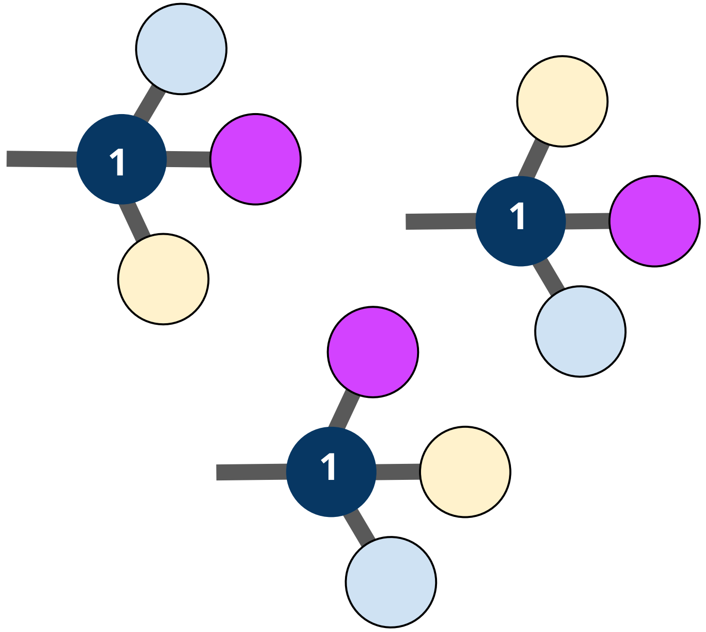
SMIRKS need to be created from aligned indexed atoms



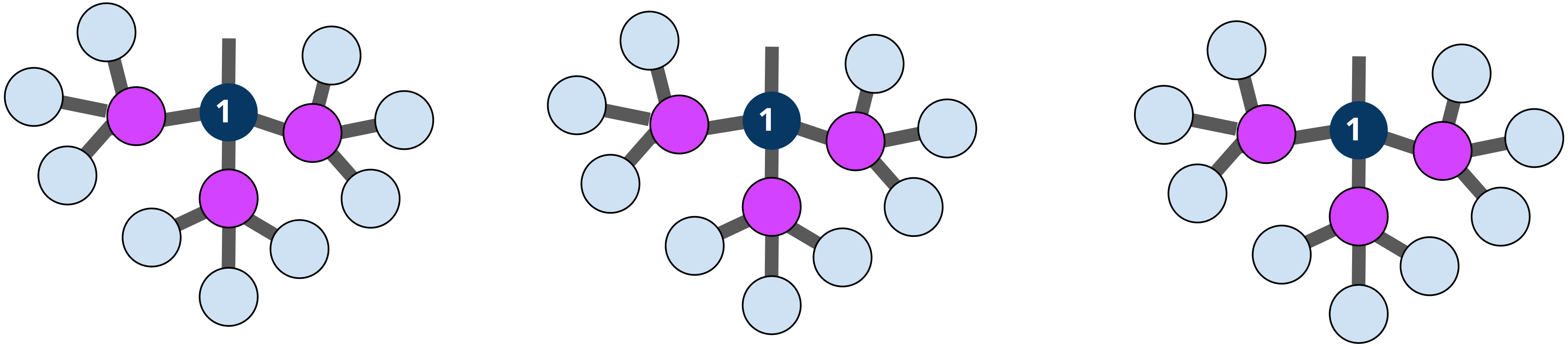
This is slightly more complex for torsions



Just as important, unindexed atoms need to agree when they are added



## Size of SMIRKS is determined automatically



- Layers of atoms are added until SMIRKS match the input clustering
- Non-indexed atoms can be removed during SMIRKS reduction

# Order is important when creating hierarchical SMIRKS

Handwritten SMIRKS increase in complexity

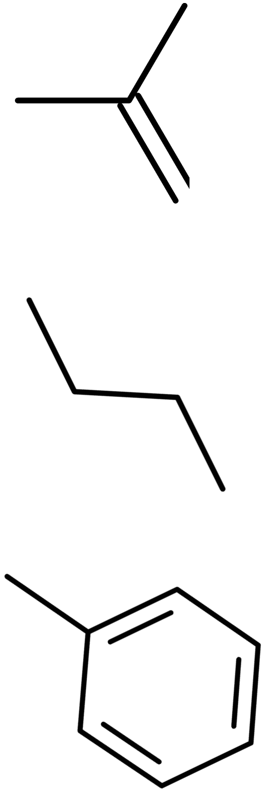
| |
|---|
| [*:1] |
| [#6:1] |
| [#6X2:1] |
| [#6X3r3,#6X3r4:1] |

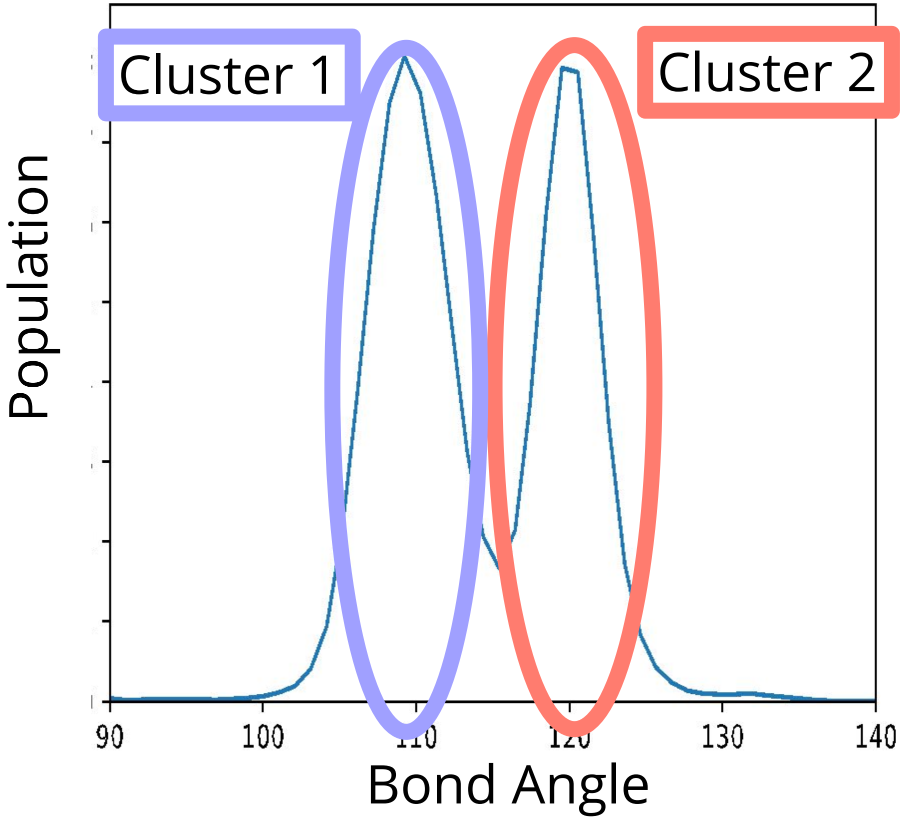
When automatically generating SMIRKS, the “correct” order should be determined by input molecules and reference data.

Sort clusters of fragments by

- Number of fragments in the cluster
- Those with more decorators per atom
- Random, checking multiple options

# Toy Example: Consider finding parameters for angles around carbon

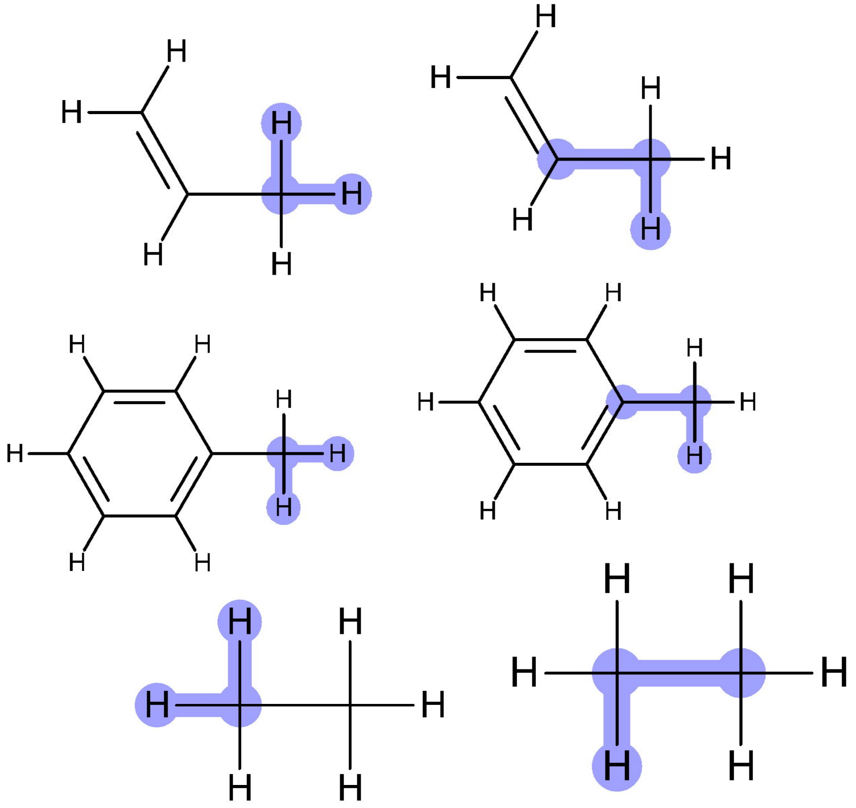




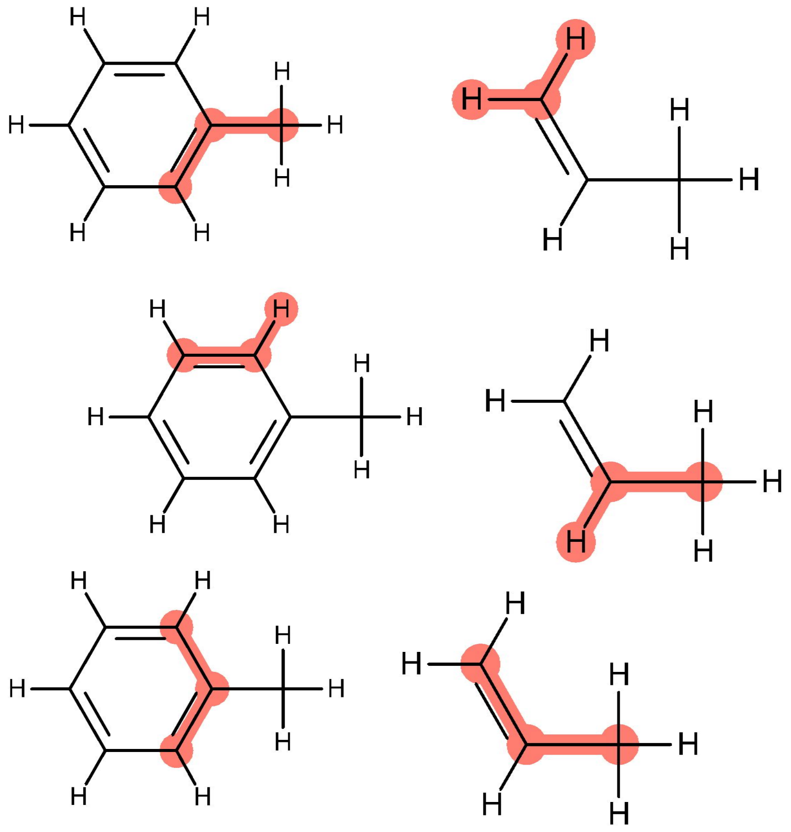
Initially identified two clusters.

## Step 1: cluster atoms by which parameter they will be assigned

Cluster 1

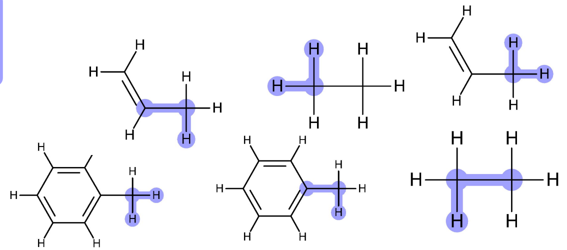


Cluster 2



## Step 2: Extract all possible decorator combinations for each cluster

Cluster 1

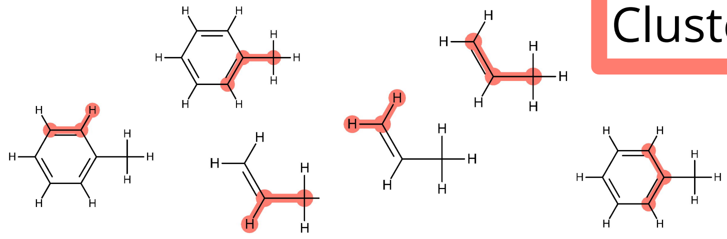


[#1AH0X1x0!r+0,#6AH3X4x0!r+0,#6aH0X3x

[*:1]-[#6X4]-[*:3]

,#6AH0X3x0!r+0,#6AH2X4x0!r+0:3]

Cluster 2



[#1AH0X1x0!r+0,#6AH3X4x0!r+0,#6aH1X3x2r6+
0:1]-,:
x2r6+

[*:1]-[#6X3]-[*:3]

6AH0X3x0!r+0,#6AH2X3x0!r+0,#6AH3X4x0!r+0,
#6aH1X3x2r6+0:3]

## Step 3: Systematically reduce SMIRKS to only the essentials

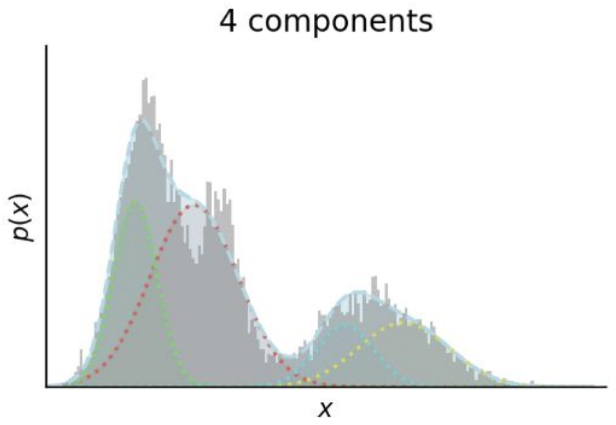
- Compare decorators in each cluster
- Remove all unnecessary decorators while maintaining clustering

## QM data for valence terms should drive chemical perception choices

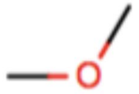
- Bonds and angles
  - Cluster based on bond lengths/angles and/or force constants
  - #Valence
- Proper torsions
  - Fragment molecules and then cluster torsions using force constants
  - #Torsions
- Improper torsion
  - Cluster using phase angles and force constants
  - Starting with nitrogen centers
  - #Improper-torsion

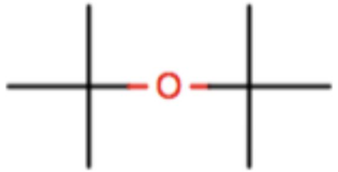
## The best types of data and clustering algorithms are still being considered

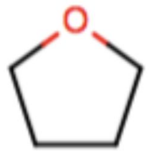
- ChemPer is modular so we can easily try new clustering algorithms



- Concerns about multiple contributions to energy changes in valence terms

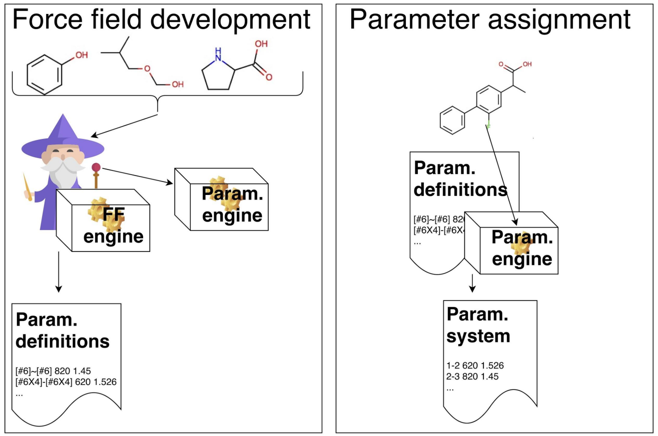




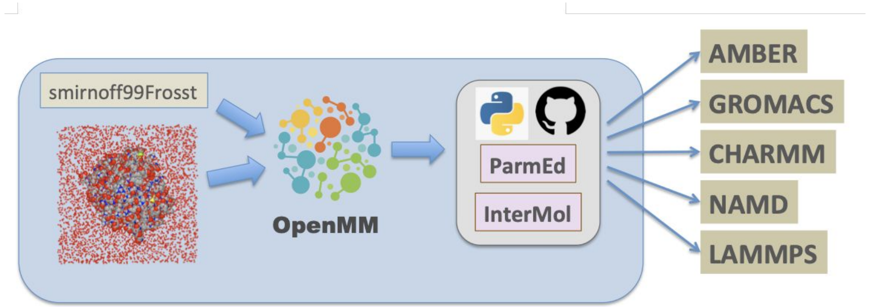


|  |  |  |  |
|---|---|---|---|
| $\Theta_0$: | 111.6 | 125.9 | 108.5 |
| k: | small | medium | large |

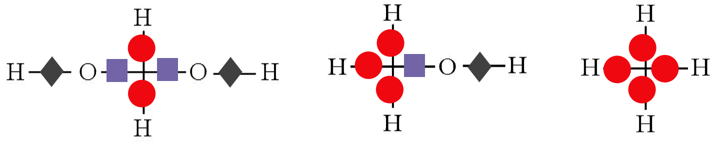
## Direct chemical perception reduces redundancy



## SMIRNOFF99Frosst is available now



## ChemPer makes SMIRKS from clustered fragments

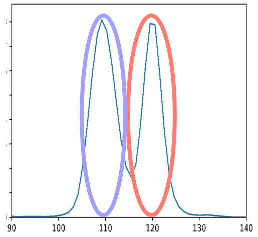


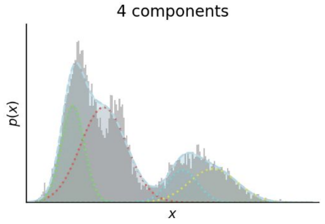
● [#6AH2X4x0!r+0,#6AH3X4x0!r+0,#6AH4X4x0!r+0:1]-;!@[#1AH0X1x0!r+0:2]

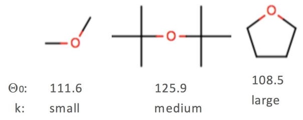
■ [#6AH2X4x0!r+0,#6AH3X4x0!r+0:1]-;!@[#8AH1X2x0!r+0:2]

◆ [#8AH1X2x0!r+0:1]-;!@[#1AH0X1x0!r+0:2]

## Further research into types of clustering data and algorithms is required







# Acknowledgements

GitHub:

- github.com/openforcefield/openforcefield
- github.com/openforcefield/smirnoff99Frosst
- github.com/mobleylab/chemper

Slack:

- #smirnoff
- #smarty