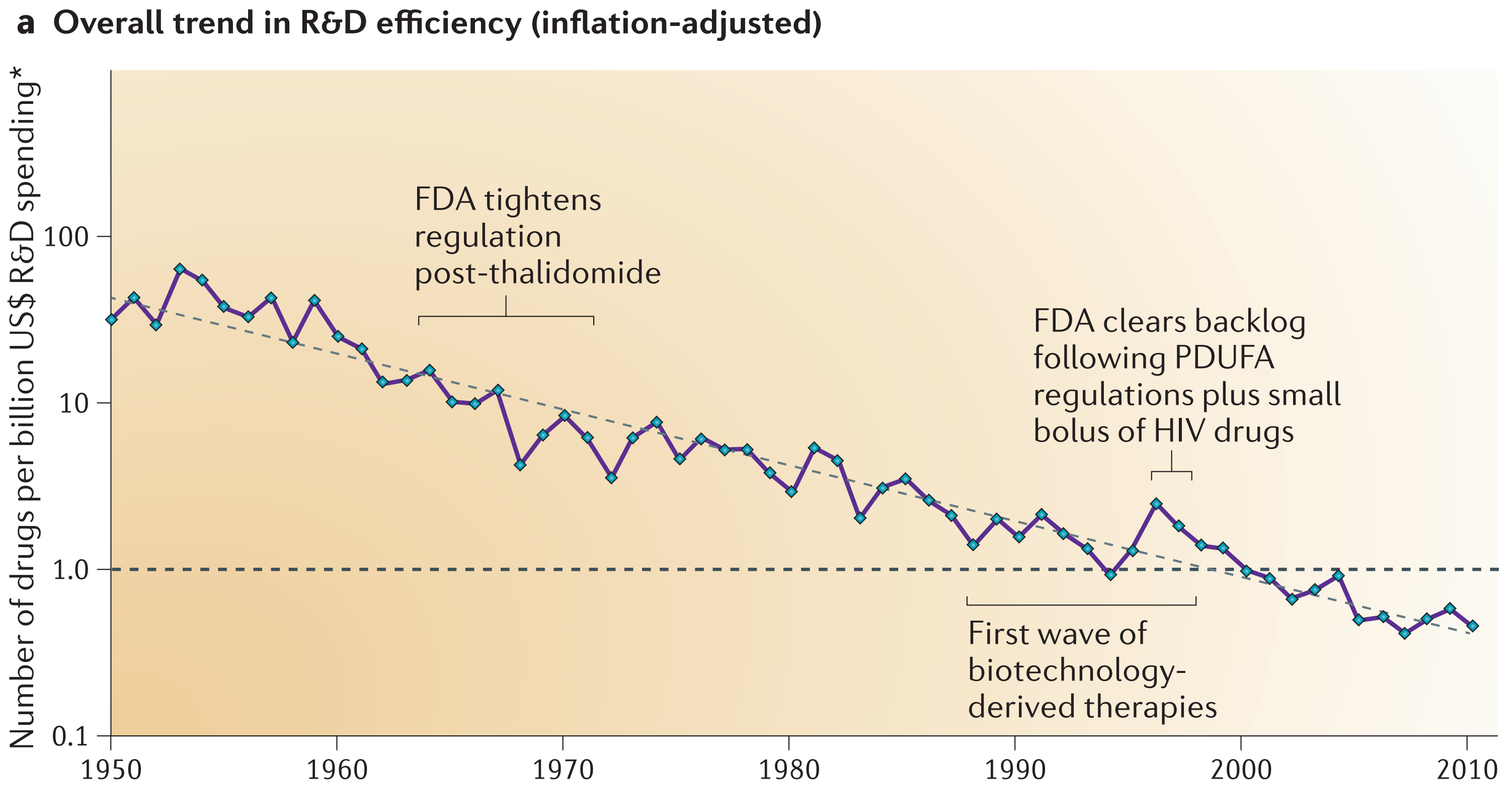
We want to automatically build new, accurate force fields for diverse applications including drug discovery, materials design, etc.

Take drug discovery as an example: It's very expensive and usually fails



Total pharma R&D spending **doubled** to $65B over 2000-2010

FDA approvals of new molecular entities **went down by half**

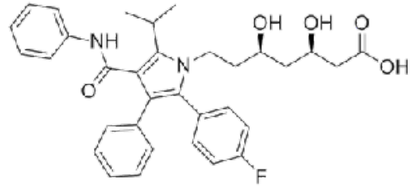
Number of truly innovative new molecules **remained constant at 5-6/year**

2010-2015 saw large reductions in pharma R&D in the US

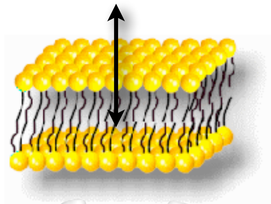
Scannell et al. Nature Rev Drug Disc 11:191, 2012

Alchemical techniques can compute binding affinities, but also many other useful properties

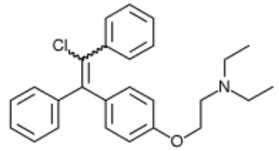
partition coefficients (logP, logD) and permeabilities



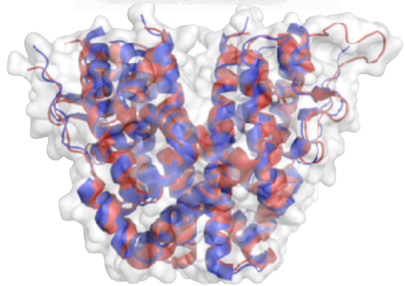
lipitor



selectivity for subtypes or related targets/off-targets



clomifene



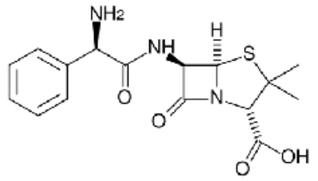
ERα/β

lead optimization of affinity and selectivity

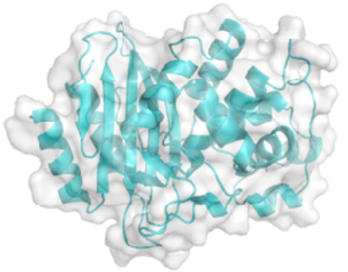
Imatinib

Dasatinib

susceptibility to resistance mutations



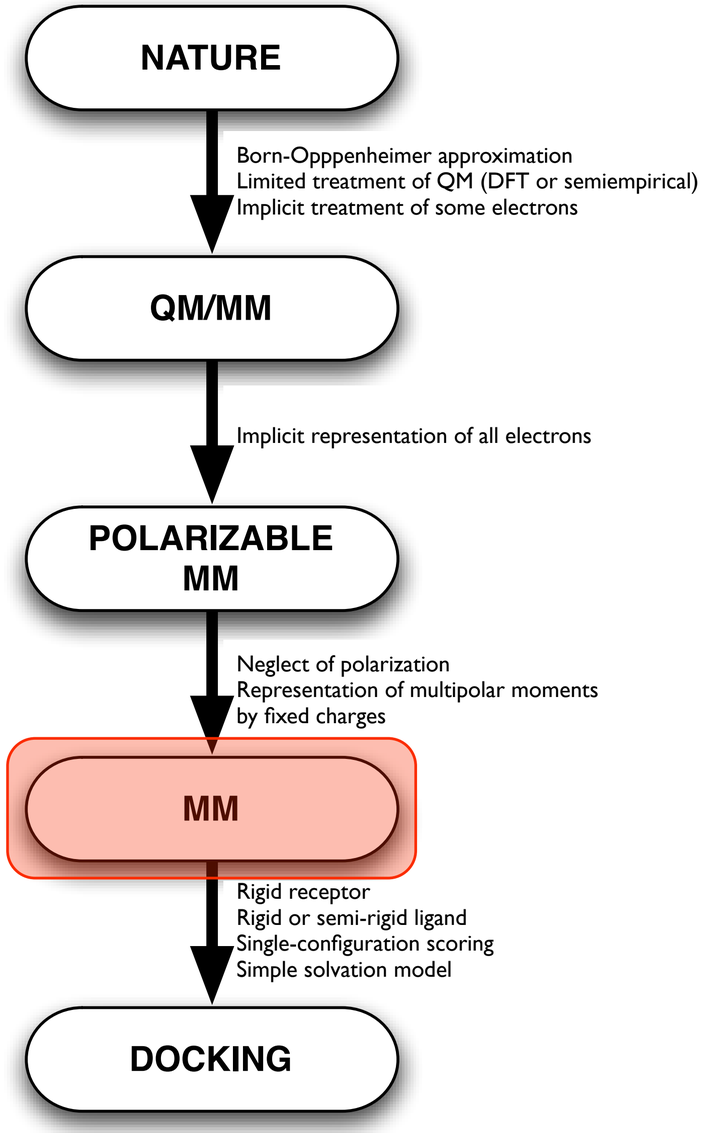
ampicillin



β-lactamase

also solubilities, polymorphs, etc.

What details are crucial for accuracy?



if insufficiently accurate, systematically add detail

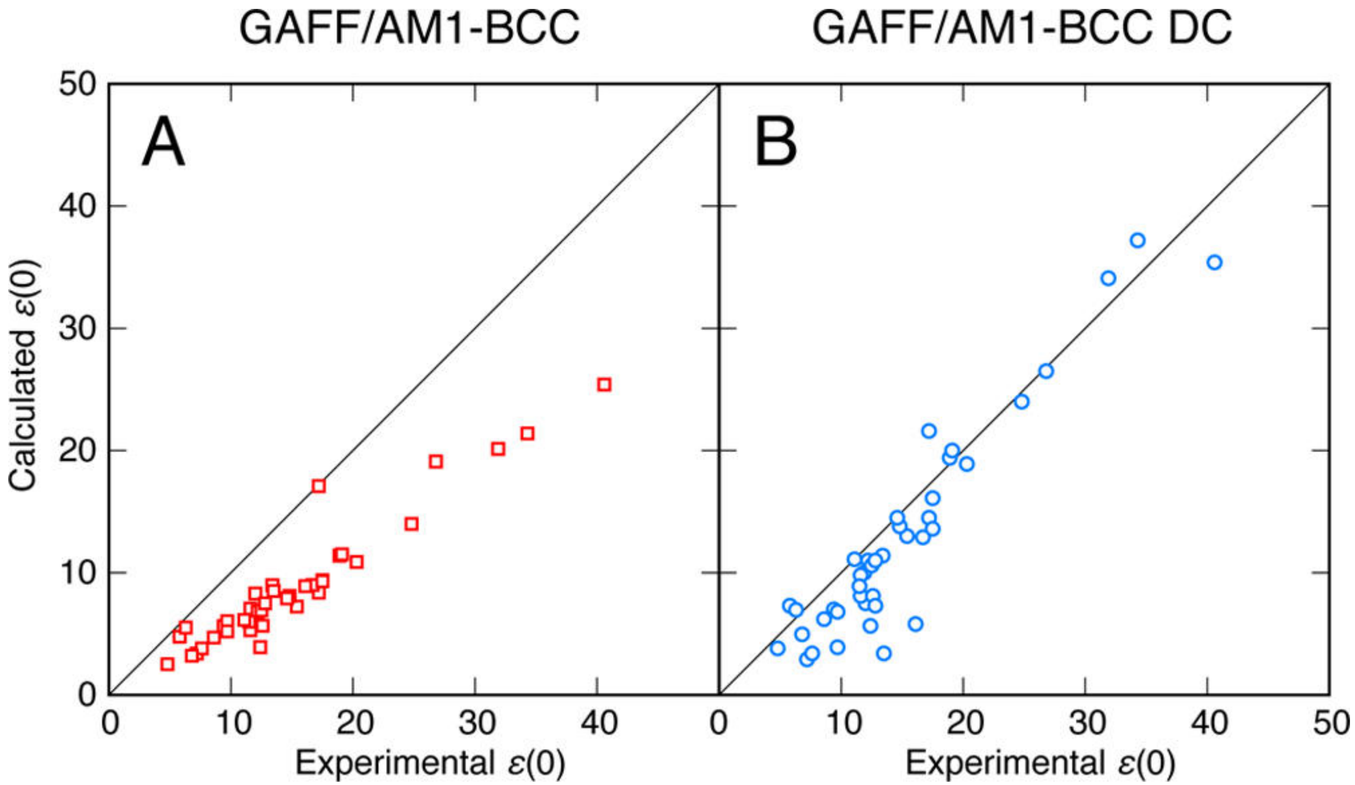
if accurate enough, systematically

$$V(\mathbf{q}) = \sum_{\text{bonds}} K_r(r - r_{eq})^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_{eq})^2$$

$$+ \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right]$$

molecular mechanics potential energy

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



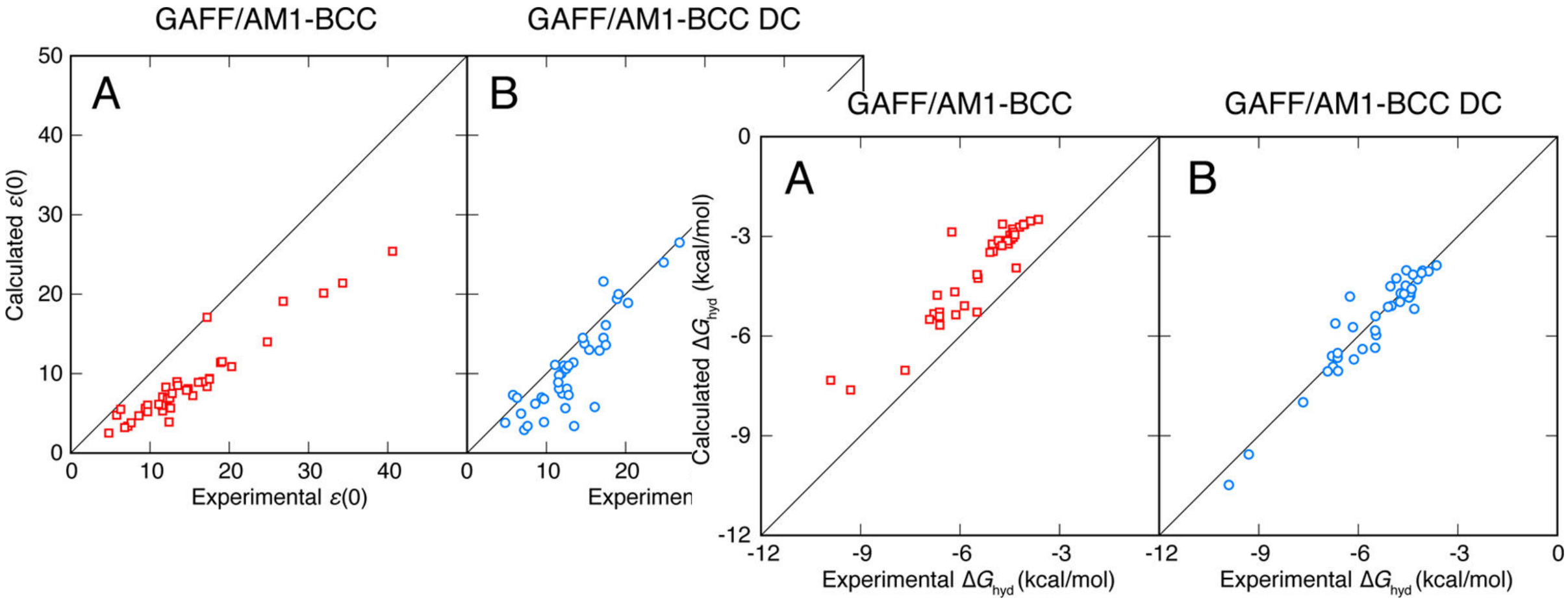
e.g. hydroxyl dielectrics
and hydration free energies

but many other functional groups too; see e.g. 10.1021/ct800409d

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

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

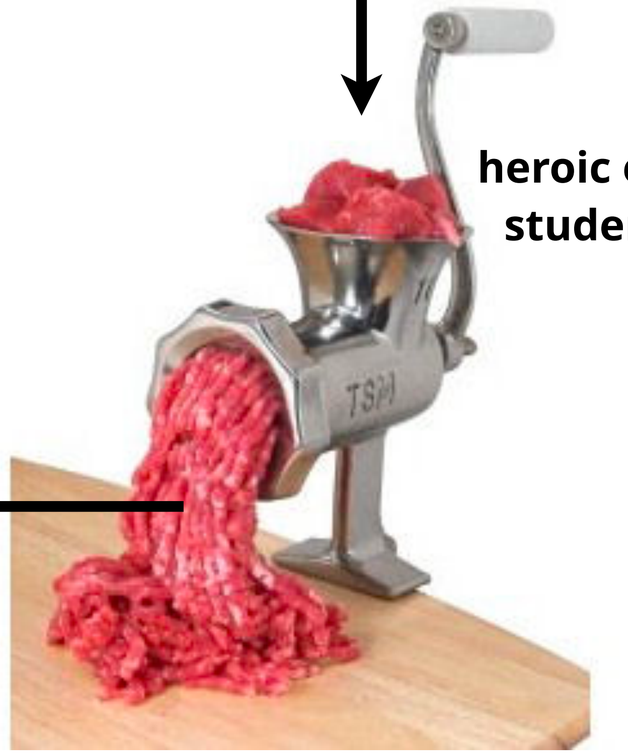


Many functional groups in today's force fields have serious systematic errors that await a general fix, because refitting is too hard

DOI: 10.1021/jp411529h

Force field development traditionally takes a heroic, monumental effort and many person-years

experimental data
quantum chemistry
keen chemical intuition



heroic effort by graduate students and postdocs

a parameter set we desperately hope someone uses

We want to change that, AND produce better FFs

Force field development traditionally takes a heroic, monumental effort and many person-years

experimental data
quantum chemistry
keen chemical intuition

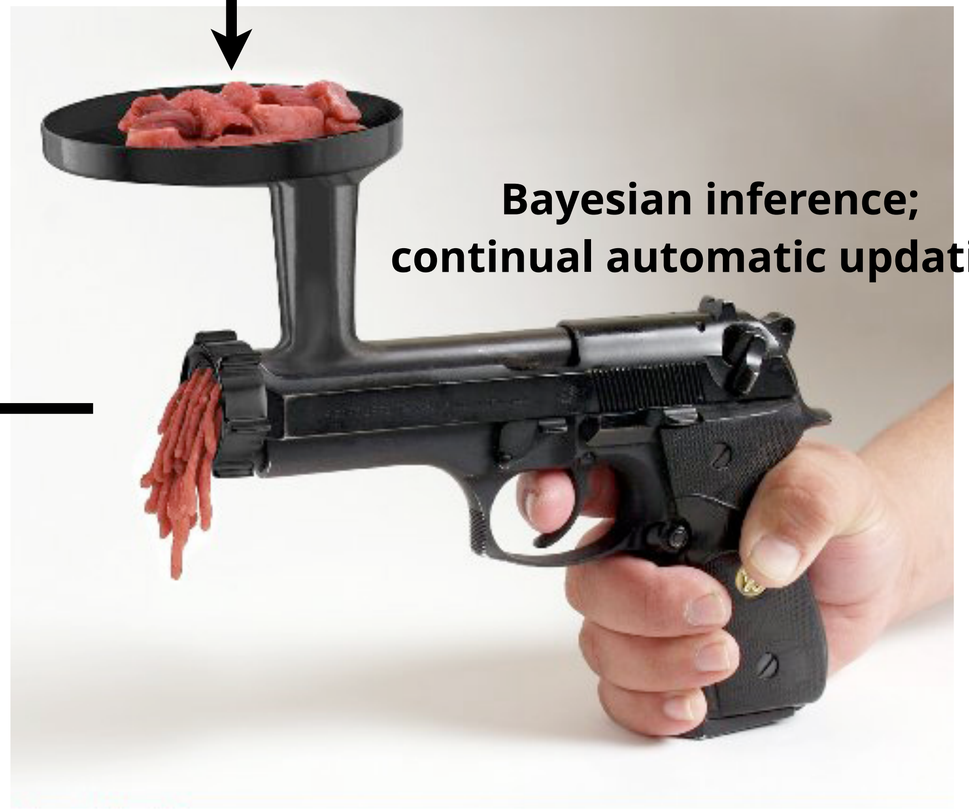
heroic effort by graduate students and postdocs

a parameter set we desperately hope someone uses

We want to change that, AND produce better FFs

We want to move to a new world where force field development is automated and based on data

functional form
experimental data
quantum chemistry
uncertainties



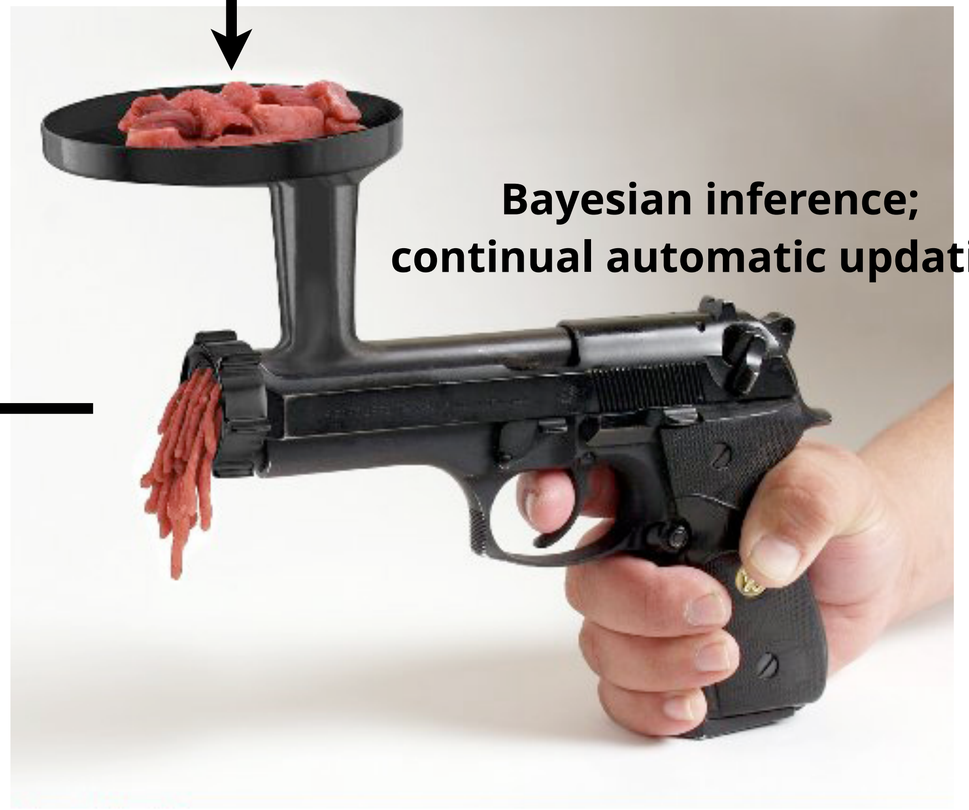
Bayesian inference;
continual automatic updating

ensemble of
parameter sets

pull the trigger and go!

We want to move to a new world where force field development is automated and based on data

functional form
experimental data
quantum chemistry
uncertainties



Bayesian inference;
continual automatic updating

ensemble of
parameter sets

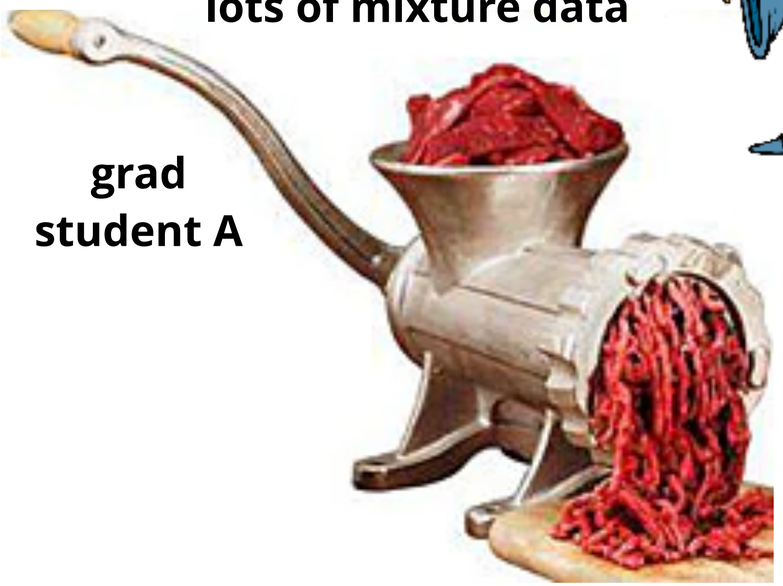
pull the trigger and go!

The old approach advances force field science relatively slowly

fixed charge FF
protein structures
lots of mixture data

polarizable FF
lots of QM data
pure solvent data

grad
student A



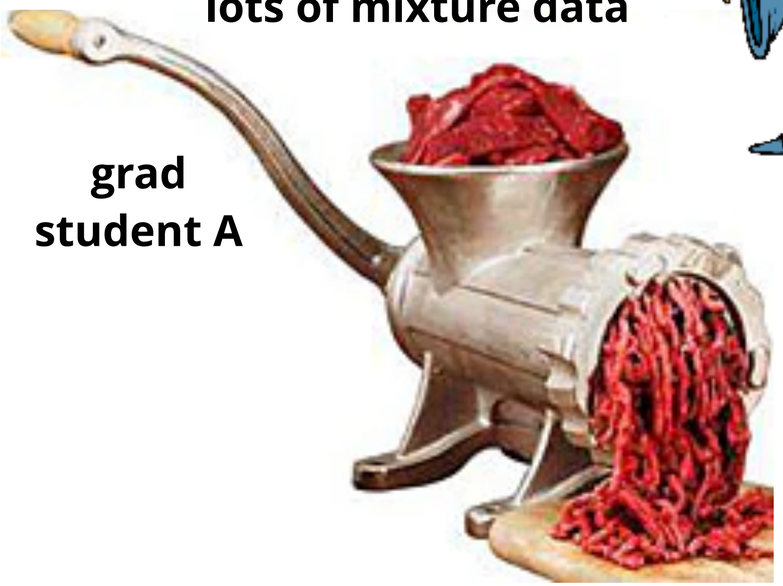
grad
student B

The old approach advances force field science relatively slowly

fixed charge FF
protein structures
lots of mixture data

polarizable FF
lots of QM data
pure solvent data

grad
student A



grad
student B

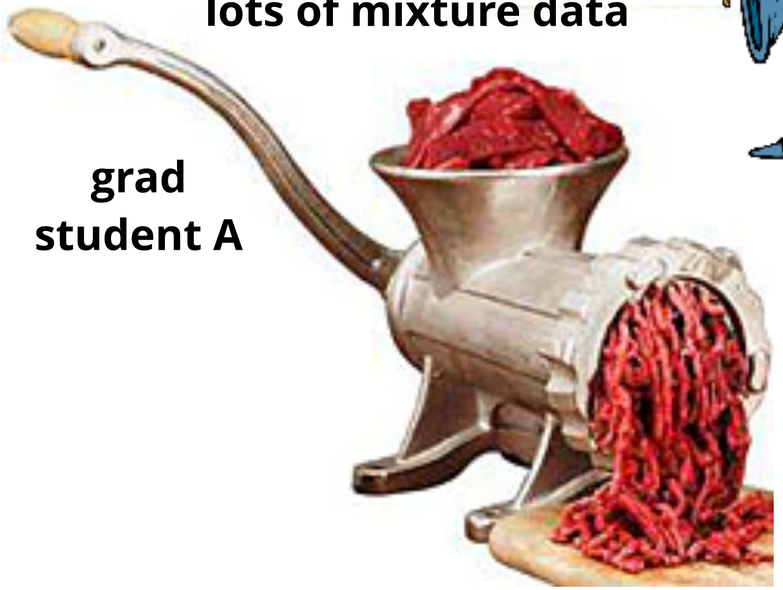
Wow! We get different force fields back!

This problem doesn’t really go away even with automatic parameterization

fixed charge FF
protein structures
lots of mixture data

polarizable FF
lots of QM data
pure solvent data

grad
student A



We’ve only removed the students from the equation

To rapidly advance force field science, we need a shared framework for parameterization

fixed charge FF
protein structures
lots of mixture data

polarizable FF
lots of QM data
pure solvent data

To rapidly advance force field science, we need a shared framework for parameterization

fixed charge FF
protein structures
lots of mixture data

polarizable FF
lots of QM data
pure solvent data

To rapidly advance force field science, we need a shared framework for parameterization

fixed charge FF
protein structures
lots of mixture data

polarizable FF
lots of QM data
pure solvent data

Then we can isolate the effects of the choice of functional form or the type of input data

What do we want out of a force field parameterization scheme?

Everything is automatic; don’t need to tweak things by hand.

Stupendous feats of chemical insight are not required.

Automatically chooses optimal functional forms.

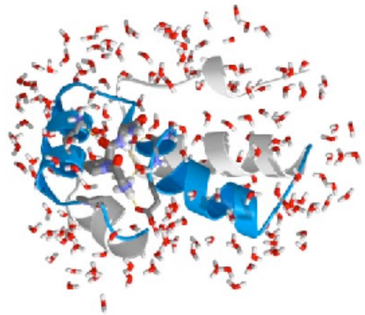
We can can add more data when we reach uncharted parts of chemical space.

Would give us an idea of how reliable it new predictions are expected to be.

We can build a map of what data we should try to collect to improve accuracy.

Is there a procedure that could fit these criteria?

Force field parameterization typically strives for a single set of parameters/results

θ^* ⟶ **MD** ⟶  ⟶ $\langle A \rangle_*$

A Bayesian approach to parameterization can automate and provide uncertainties

Bayes' rule provides a probability measure over unknown parameters given data and an automated way to update parameters given new experimental data

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$

$\mathcal{D}$ data

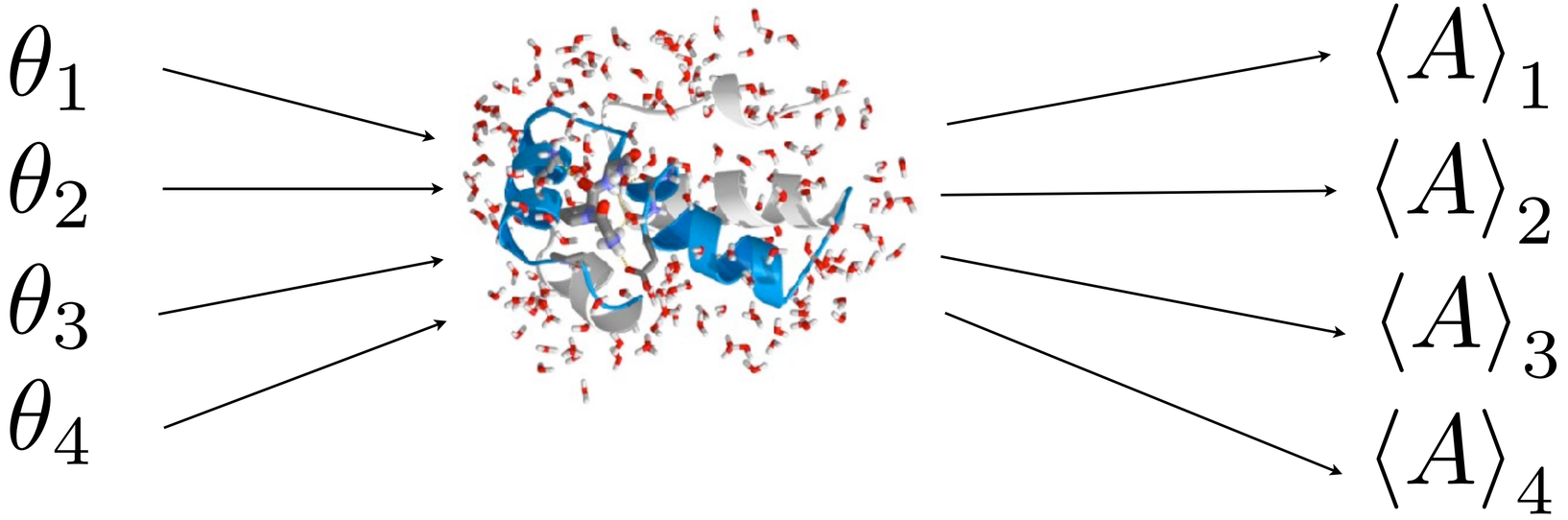
θ forcefield

$p(\theta|\mathcal{D})$ posterior

$p(\mathcal{D}|\theta)$ data model

$p(\theta)$ prior on forcefield parameters

In the Bayesian framework, we aren't after a single parameter set/result



Multiple parameter sets in, multiple estimates out

Multiple results provide insight into chemistry where parameters (and thus predictions) are uncertain

Having multiple predicted values might seem odd, but in fact it will be extremely valuable

Compound prioritization suddenly becomes
much easier in given systematic error predictions

better worse

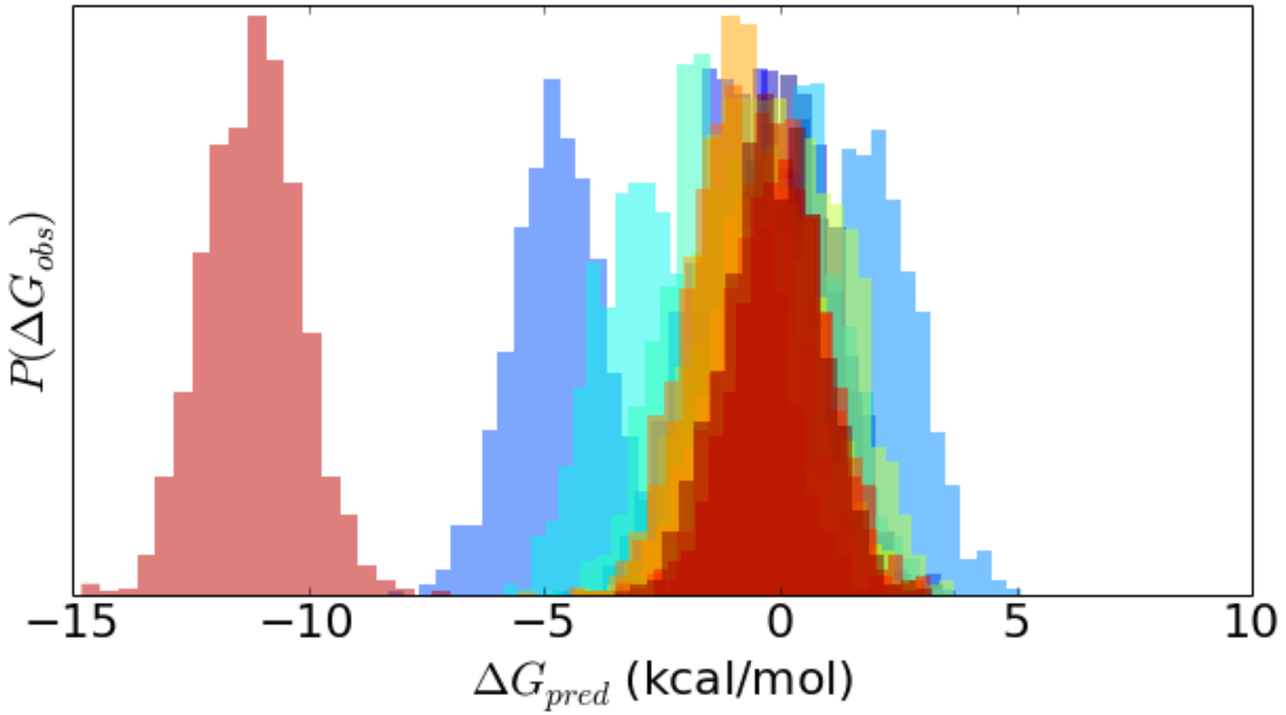
Multiple parameter sets in, multiple estimates out

We can estimate both statistical and systematic components of computed results

Having multiple predicted values might seem odd, but in fact it will be extremely valuable

Compound prioritization suddenly becomes
much easier in given systematic error predictions

better



worse

Multiple parameter sets in, multiple estimates out

We can estimate both statistical and systematic components of computed results

Thus there are several key steps towards this new world

Input data: We need lots of high quality data

Pipeline needs to be independent of chemical expertise

Bayesian parameterization framework

We'll need to rely on lots of high quality data

ThermoML Archive
densities and enthalpies of mixing for neat liquids and liquid mixtures
dielectric constants, speed of sound

Host-guest binding free energies and enthalpies (Gilson lab)
Calorimetry

Excess densities and mixing enthalpies for complex mixtures (Chodera lab)
Automated densitometry and calorimetry

Distribution coefficients for small molecules
Automated Sirius T3 measurements, pharma datasets

Quantum chemical calculations
Automated QM calculations with psi4; key advantage: can use *any* molecule

There are lots of types of data we can use

EXPERIMENTAL DATA

- densities of neat liquids and miscible liquid mixtures
- enthalpies of mixing of miscible molecular liquids
- transfer free energies (partition and distribution coefficients, hydration free energies)
- host-guest binding thermodynamics (free energies and enthalpies)
- small molecule 1D/2D NMR data (chemical shifts, J-coupling constants, NOE/ROEs)
- dielectric constants of neat liquids (and possibly mixtures)
- speed of sound data
- small molecule crystal structures and primary reflection data (CCSD)
- protein-ligand binding free energies

QM DATA

- QM electrostatic potentials near molecular surface
- QM equilibrium geometries and force constant matrices (Hessians)
- QM single-point energies for 1- and 2-torsion drives
- C6 dispersion coefficients
- statistic atomic and molecular polarizabilities

primarily valence terms
primarily Lennard-Jones
primarily electrostatics

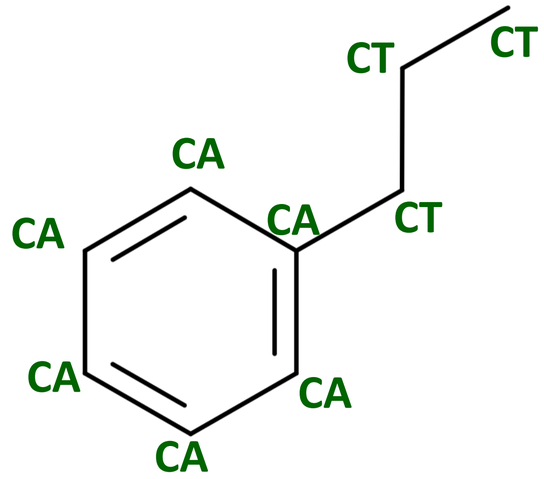
Beyond data, we need to remove the need for wizardry (tremendous chemical expertise)

Wizards are responsible for chemical perception/chemical intuition

To get rid of them, we need:

- A language to express chemical perception
- A way to sample over chemical perception automatically

Chemical perception defines what chemistry is “similar”, usually via atom types

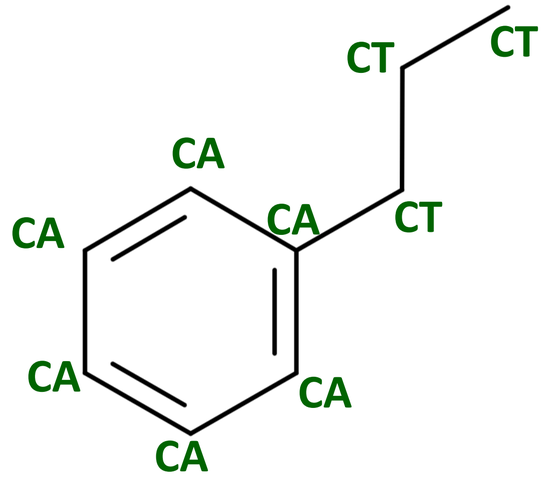


CT: aliphatic (sp3) carbon

CA: aromatic sp2 carbon

```
X -CA-CA-X     4    14.50          180.0              2.
…
CT-CT-CT-CT    1     0.18            0.0             -3.
CT-CT-CT-CT    1     0.25          180.0             -2.
CT-CT-CT-CT    1     0.20          180.0              1.
```

Today’s force fields mostly use indirect chemical perception



CT: aliphatic (sp3) carbon
CA: aromatic sp2 carbon

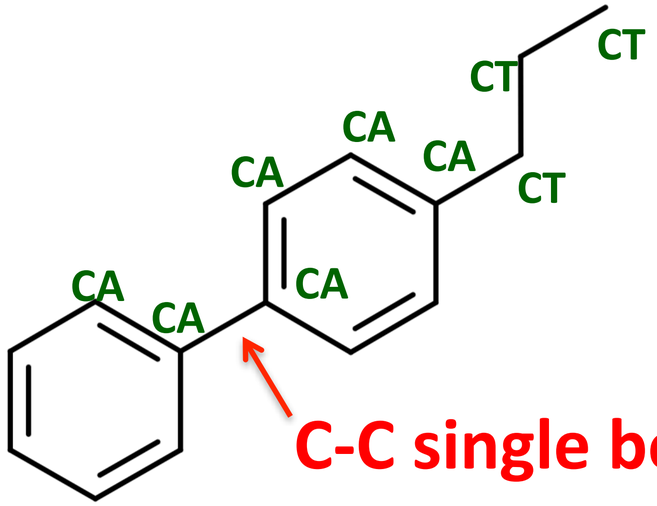
Some tool (or human) assigns atom types

From the atom types, parameters are assigned

Thus, atom types must encode all requisite chemistry

Catch 22 for us: We can’t encode requisite chemistry until we sample over it to find out what it is

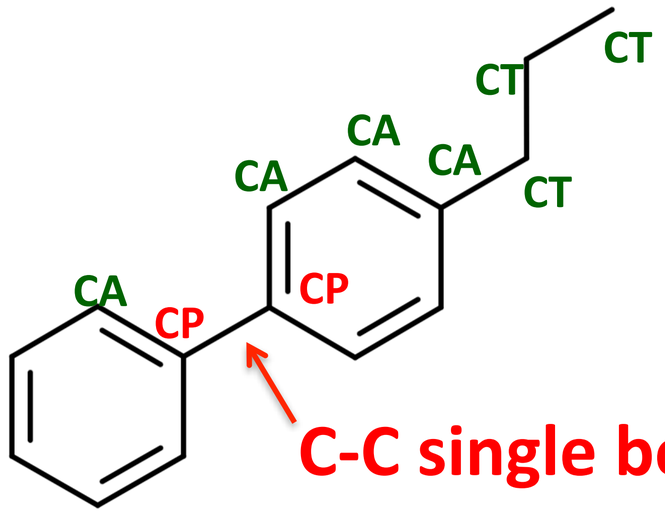
This is a vital issue: Failing to capture the requisite chemistry leads to disaster



CT: aliphatic (sp3) carbon
CA: aromatic sp2 carbon

```
X -CA-CA-X      4    14.50          180.0              2.
…
CT-CT-CT-CT     1     0.18            0.0             -3.
CT-CT-CT-CT     1     0.25          180.0             -2.
CT-CT-CT-CT     1     0.20          180.0              1.
```

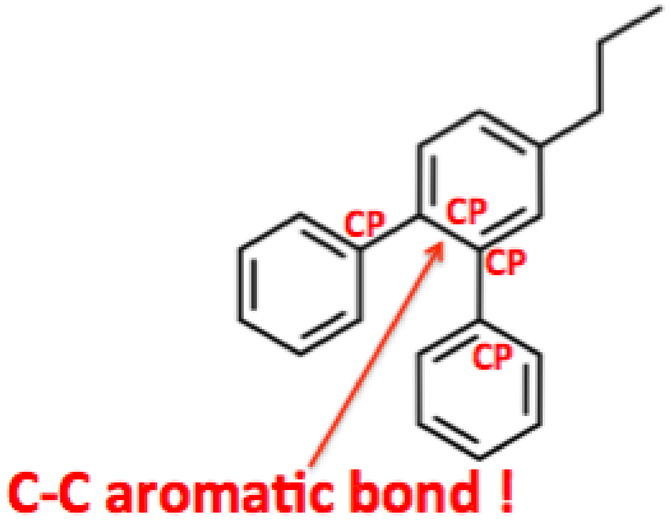
One can fix this with more complex atom typing



CT: aliphatic (sp3) carbon
CA: aromatic sp2 carbon
CP: aromatic sp2 carbon joined to another aromatic ring

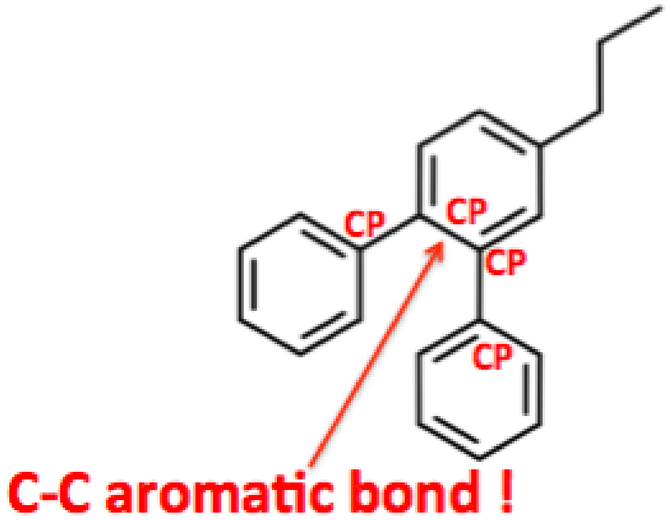
```
X -CA-CA-X      4    14.50          180.0              2
X -CP-CP-X      4     2.50          180.0              2
…
CT-CT-CT-CT     1     0.18            0.0             -3
CT-CT-CT-CT     1     0.25          180.0             -2
CT-CT-CT-CT     1     0.20          180.0              1
```

Though, new chemistry can still pose challenges



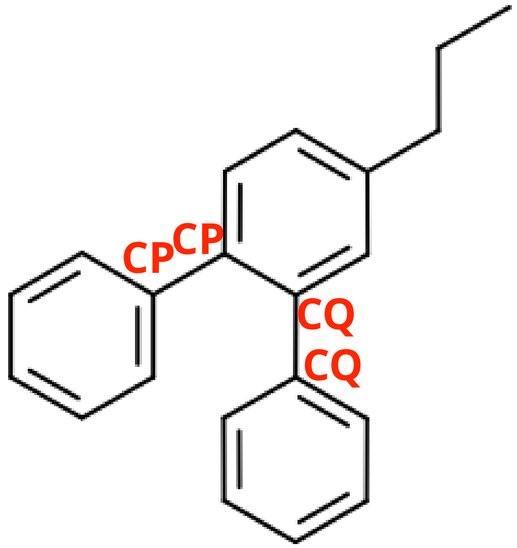
CT: aliphatic (sp3) carbon
CA: aromatic sp2 carbon
CP: aromatic sp2 carbon joined to another aromatic ring

Though, new chemistry can still pose challenges

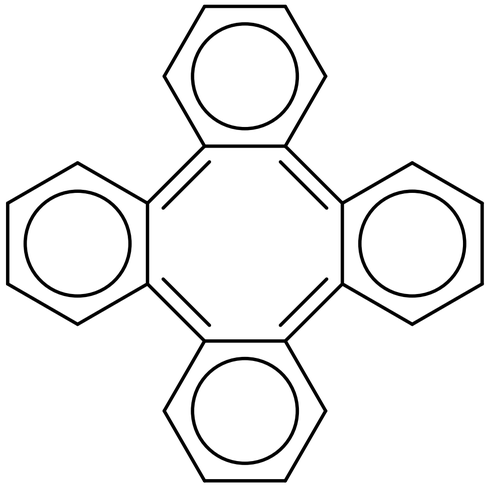


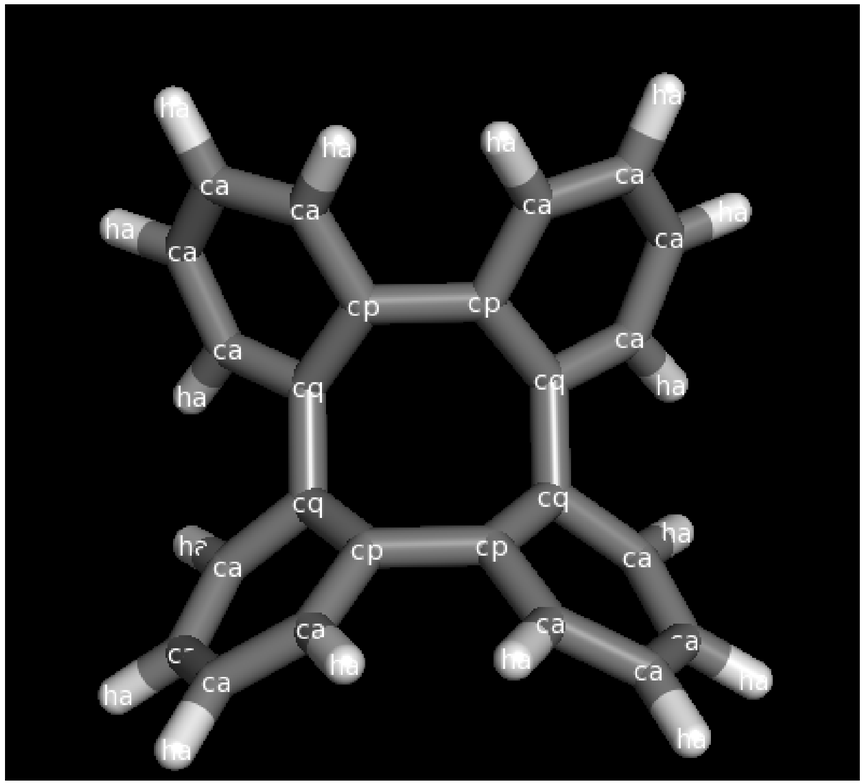
CT: aliphatic (sp3) carbon
CA: aromatic sp2 carbon
CP: aromatic sp2 carbon joined to another aromatic ring

Not to worry, we can fix that too:
Let's introduce a new type CQ which is the same as CP except that CP-CQ bonds are aromatic



Superficially that’s fine, though it does result in a proliferation of parameters





Torsions:

ca-cp-cp-ca

ca-cp-cq-ca

ca-cp-cq-cq

cp-cp-cq-cq

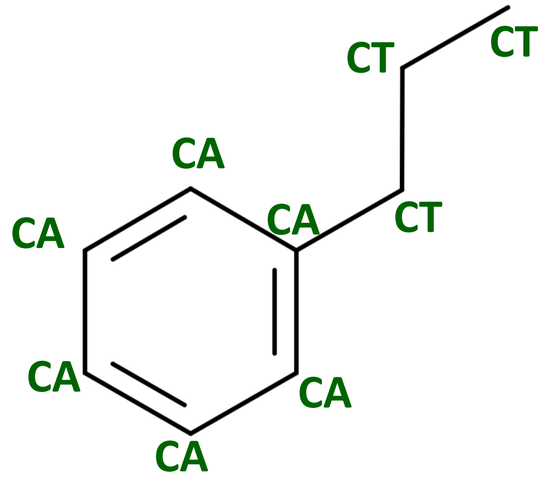
cp-cq-cq-cp

ca-ca-cq-cp

etc.

GAFF2 only knows a small fraction of these; parmchk2 generally makes them all equal

The problem, here, is indirect chemical perception

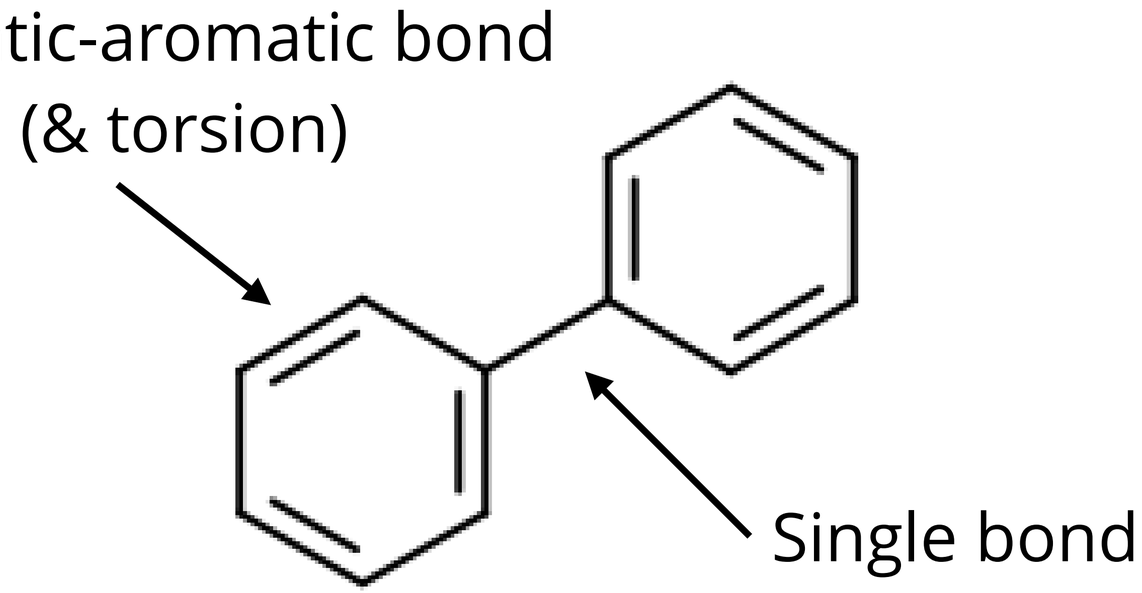


CT: aliphatic (sp3) carbon
CA: aromatic sp2 carbon

Atom types -> parameters

Thus atom types must have great complexity

So, why not just do direct chemical perception?



Direct chemical perception: Assign parameters based on analysis of the molecule rather than atom types

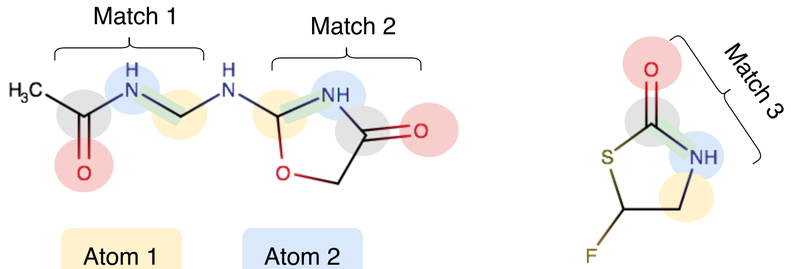
Answer: No good reason, apparently; hard to change legacy code

SMIRNOFF:

A modern starting point for forcefield parameterization

https://github.com/openforcefield/openforcefield

The SMIRKS Native Open Force Field (SMIRNOFF) avoids the complexities of atom typing



MATCH BONDS DIRECTLY: [#6X4:1]-[#7X3:2]-[#6X3]=[#8X1+0]

Labeled bond | Alpha carbon | Beta oxygen

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

We'll discuss our force field format and efforts in this direction a bit later; for now, know that we have a new force field format using direct chemical perception

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

SMIRKS provide a complete language for chemical perception

Example SMIRKS patterns	
`[#1:1]`	hydrogen
`[#8X2H0+0:1]`	neutral divalent oxygen with no connected hydrogens
`[#1:1]-[#6X4]~[*+1,*+2]`	hydrogen attached to tetravalent carbon with a +1 or +2 attached atom
`[#6:1]#[#7:2]`	carbon triple bonded to a nitrogen
Basic ingredients	
`[*]`	An atom – here, any atom
`[Cl]`	A chlorine atom
`[Cl]-[#6]`	Chlorine singly bonded to any carbon
`[#6:1]-[#7]([#1])-[#6]`	A labeled carbon atom attached to a nitrogen which is attached to carbon and hydrogen

Decorators for atoms and bonds		
	Symbol	**Definition**
Atoms	#n	Atomic Number
	*	any atom
	A	Aliphatic
	a	Aromatic
	Hn	Hydrogen count
	Xn	Connectivity
	±n	charge
	rn	in ring of (smallest size) n
	`$(*~[a])`	the preceding atom is attached to an aliphatic
Bonds	~	any bond
	@	ring bond
	–	single bond
	=	double bond
	:	aromatic bond
Booleans for both	,	or
	&	high priority `and`
	;	low priority `and`
	!	Not

Removing the need for wizards also requires a way to sample over chemical perception automatically

Wizards are responsible for chemical perception/chemical intuition

To get rid of them, we need:

- A language to express chemical perception
- A way to sample over chemical perception automatically

We can combine Monte Carlo in chemical perception space with Bayesian inference to automate this process

parent types

```
% atom types
[#1]    hydrogen
[#6]    carbon
[#7]    nitrogen
[#8]    oxygen
[#9]    fluorine
[#15]   phosphorous
[#16]   sulfur
[#17]   chlorine
[#35]   bromine
[#53]   iodine
```

x

decorators

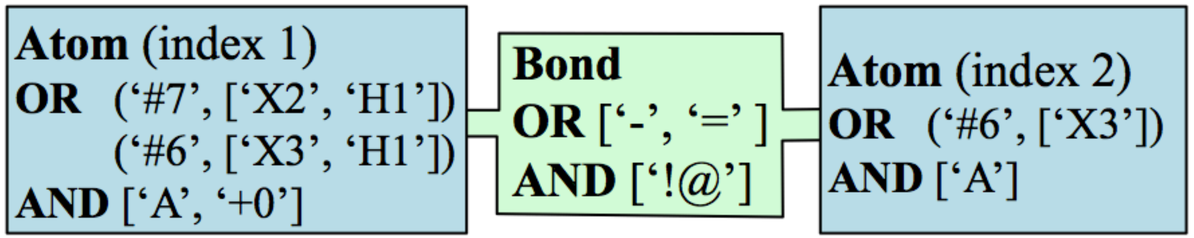
```
% total connectivity
X1          connections-1
X2          connections-2
X3          connections-3
X4          connections-4
% total-h-count
H0          total-h-count-0
H1          total-h-count-1
H2          total-h-count-2
H3          total-h-count-3
% formal charge
+0          neutral
+1          cationic+1
-1          anionic-1
% aromatic/aliphatic
a           aromatic
A           aliphatic
```

=

proposed child types

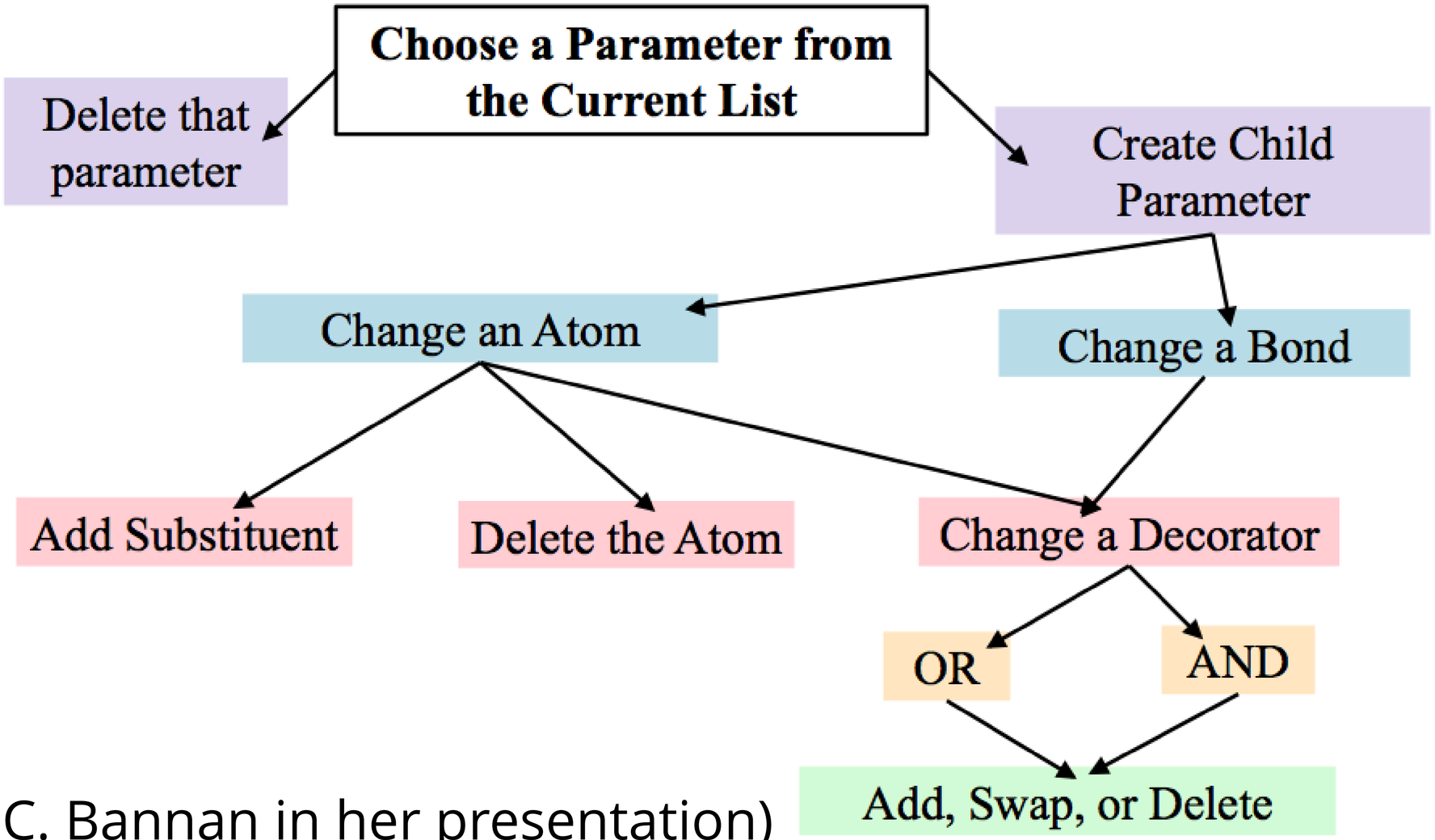
```
[#6X4:1]      tetrahedral carbon
[#6:1]~[#7]   carbon nitrogen-adjacent
```

“[#7X2H1,#6X3H1;A;+0:1] -,= ;!@ [#6X3;A:2]”



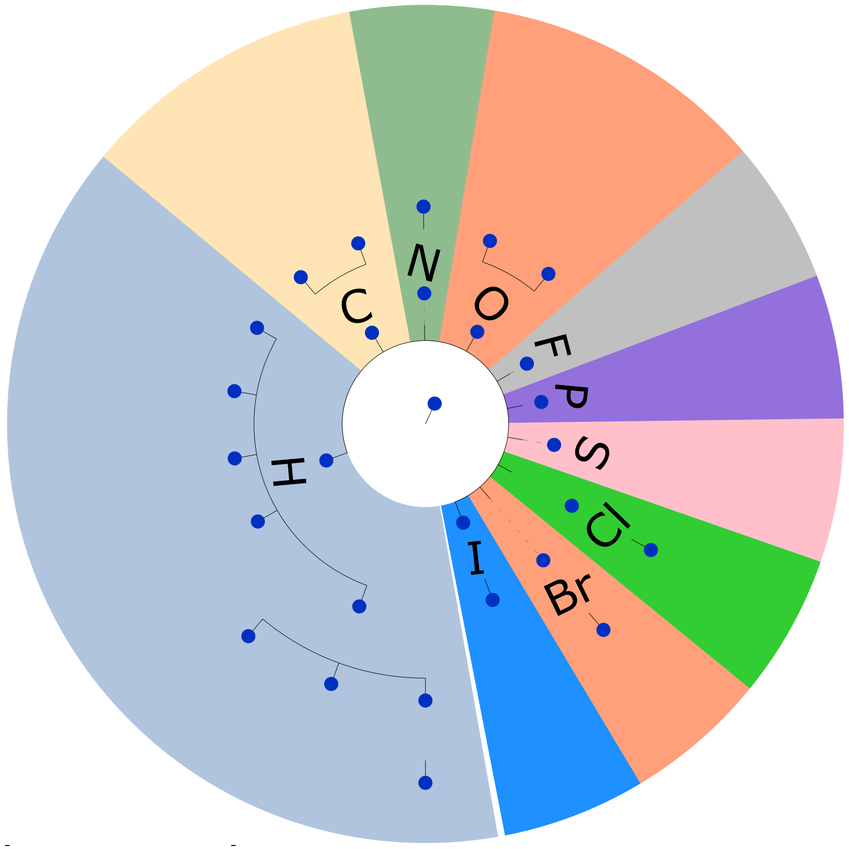
https://github.com/open-forcefield-group/smarty

MC Moves in SMIRKS space can suggest and evaluate new chemical environments



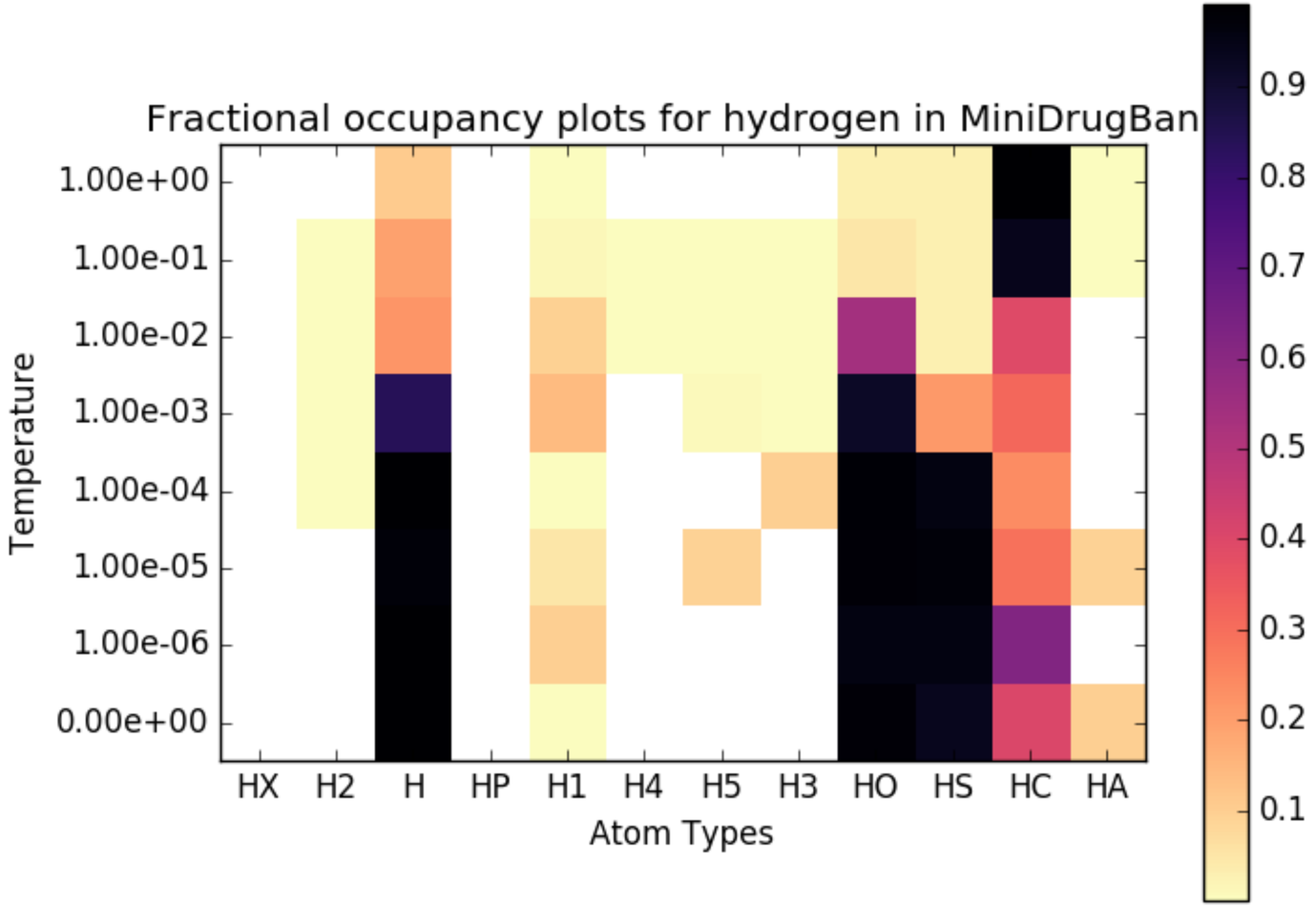
(More from C. Bannan in her presentation)

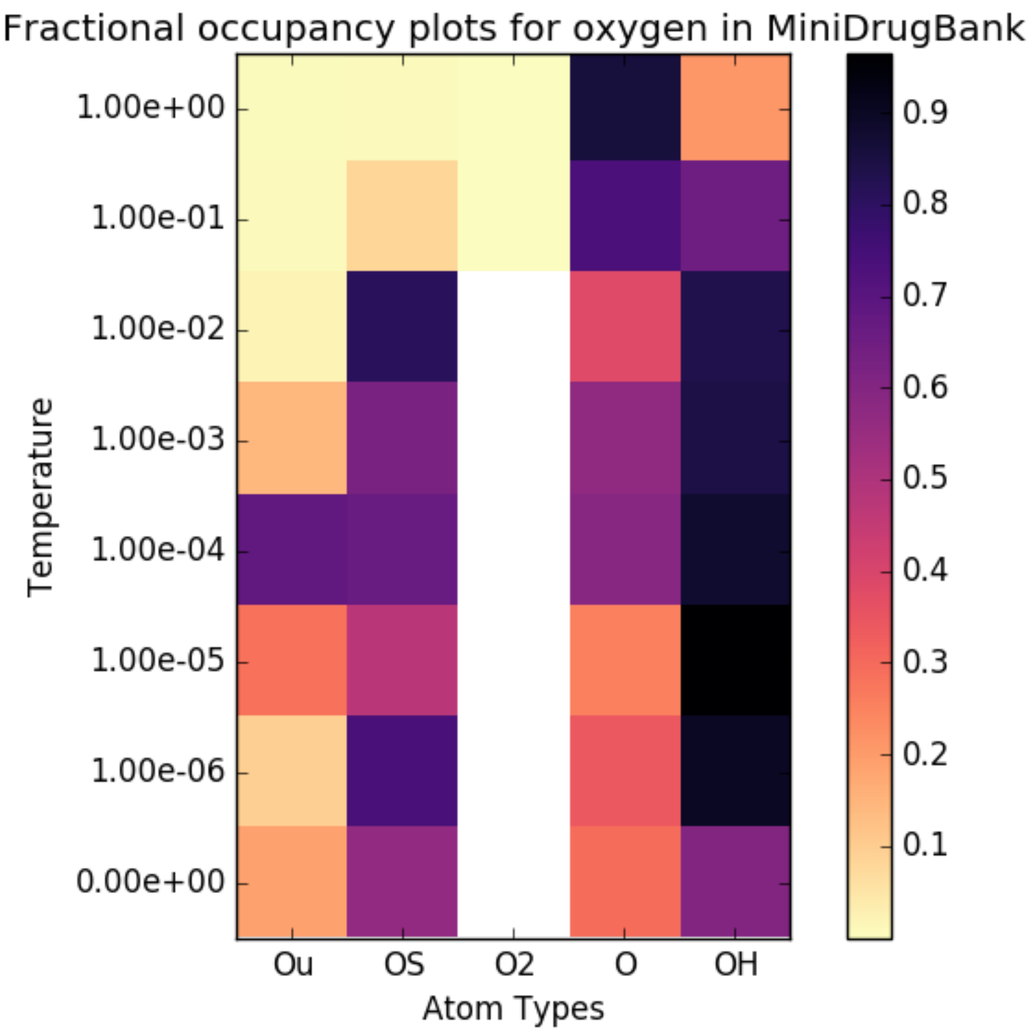
We’ve tested this approach by seeing how it can learn the chemical perception corresponding to existing atom typing




```
H ([#1:1])
    5262 ([#1:1]~[#6])
        4148 ([#1:1]~[#6!X3])
            8668 ([#1:1]~[#6!X3]~[$ewg2])
                1874 ([#1:1]~[#6!X3](~[$ewg2])~[$ewg2])
                    4596 ([#1:1]~[#6!X3](~[#7])(~[$ewg2])~[$ewg2])
            2356 ([#1:1]~[#6!X3X2])
            5962 ([#1:1]~[#8X2])
            2012 ([#1:1]~[#6!X3]~[#17])
        4227 ([#1:1]~[#6]~[#17])
        1674 ([#1:1]~[#6H1X3]~[#7!X4])
            6955 ([#1:1]~[#6H1X3](~[#6])~[#7!X4])
    1945 ([#1:1]~[#16])
C ([#6:1])
    4016 ([#6X4:1])
    3620 ([#6;X3:1])
N ([#7:1])
O ([#8:1])
    3664 ([#8H0:1])
    1964 ([#8!X2;R0:1])
F ([#9:1])
    2860 ([#9!R:1])
P ([#15:1])
    5153 ([#15:1]~[$ewg2])
S ([#16:1])
    7194 ([#16:1]~[*])
Cl ([#17:1])
    4081 ([#17:1]~[#6])
Br ([#35:1])
I ([#53:1])
```

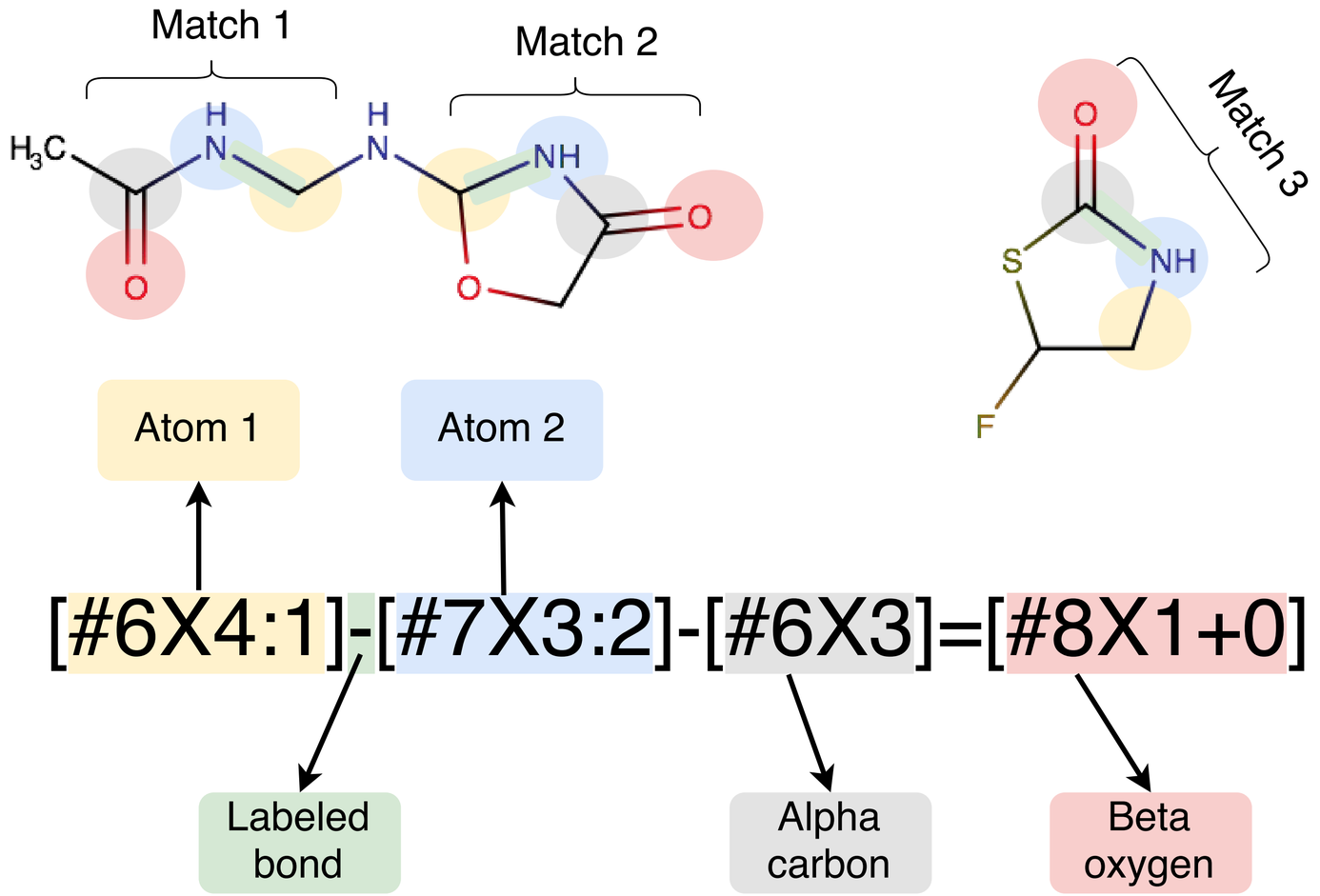
On MiniDrugBank, we are sampling the right types of chemistry and can recover existing atom typing





The SMIRKS Native Open Force Field (SMIRNOFF) avoids the complexities of atom typing

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



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

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

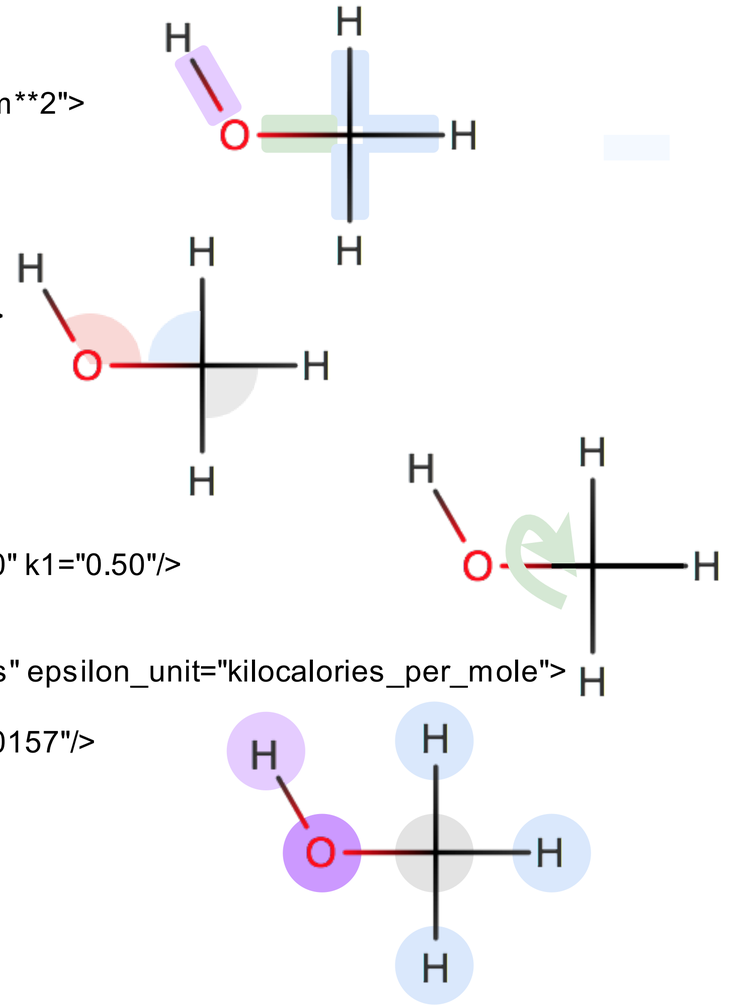
[#6:1]-[#6:2], length=1.526 angstroms, force constant=620.0 kcal/(mol angstrom^2)

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

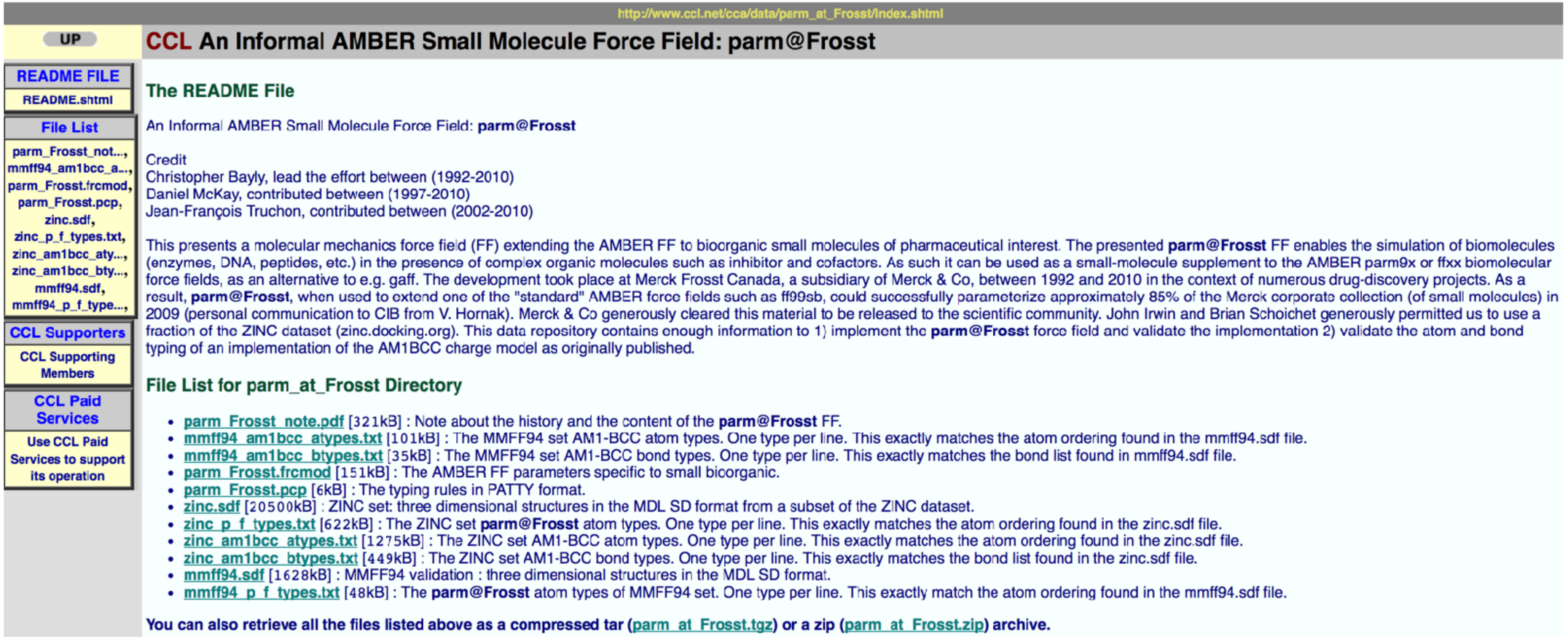
Perhaps a more specialized bond?
[#6X3:1]=[#6X3:2] with different parameters

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>
```



parm@frosst is the starting point for an improved GAFF-like small molecule forcefield



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

UP

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

README FILE
README.shtml

File List
parm_Frosst_not..., mmff94_am1bcc_a..., parm_Frosst.frcmod, parm_Frosst.pcp, zinc.sdf, zinc_p_f_types.txt, zinc_am1bcc_aty..., zinc_am1bcc_bty..., mmff94.sdf, mmff94_p_f_type...,

CCL Supporters
CCL Supporting Members

CCL Paid Services
Use CCL Paid Services to support its operation

The README File

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

Credit
Christopher Bayly, lead the effort between (1992-2010)
Daniel McKay, contributed between (1997-2010)
Jean-François Truchon, contributed between (2002-2010)

This presents a molecular mechanics force field (FF) extending the AMBER FF to biorganic small molecules of pharmaceutical interest. The presented **parm@Frosst** FF enables the simulation of biomolecules (enzymes, DNA, peptides, etc.) in the presence of complex organic molecules such as inhibitor and cofactors. As such it can be used as a small-molecule supplement to the AMBER parm9x or ffxx biomolecular force fields, as an alternative to e.g. gaff. The development took place at Merck Frosst Canada, a subsidiary of Merck & Co, between 1992 and 2010 in the context of numerous drug-discovery projects. As a result, **parm@Frosst**, when used to extend one of the "standard" AMBER force fields such as ff99sb, could successfully parameterize approximately 85% of the Merck corporate collection (of small molecules) in 2009 (personal communication to CIB from V. Hornak). Merck & Co generously cleared this material to be released to the scientific community. John Irwin and Brian Schoichet generously permitted us to use a fraction of the ZINC dataset (zinc.docking.org). This data repository contains enough information to 1) implement the **parm@Frosst** force field and validate the implementation 2) validate the atom and bond typing of an implementation of the AM1BCC charge model as originally published.

File List for parm_at_Frosst Directory

- parm_Frosst_note.pdf [321kB] : Note about the history and the content of the **parm@Frosst** FF.
- mmff94_am1bcc_atypes.txt [101kB] : The MMFF94 set AM1-BCC atom types. One type per line. This exactly matches the atom ordering found in the mmff94.sdf file.
- mmff94_am1bcc_btypes.txt [35kB] : The MMFF94 set AM1-BCC bond types. One type per line. This exactly matches the bond list found in mmff94.sdf file.
- parm_Frosst.frcmod [151kB] : The AMBER FF parameters specific to small bicorganic.
- parm_Frosst.pcp [6kB] : The typing rules in PATTY format.
- zinc.sdf [20500kB] : ZINC set: three dimensional structures in the MDL SD format from a subset of the ZINC dataset.
- zinc_p_f_types.txt [622kB] : The ZINC set **parm@Frosst** atom types. One type per line. This exactly matches the atom ordering found in the zinc.sdf file.
- zinc_am1bcc_atypes.txt [1275kB] : The ZINC set AM1-BCC atom types. One type per line. This exactly matches the atom ordering found in the zinc.sdf file.
- zinc_am1bcc_btypes.txt [449kB] : The ZINC set AM1-BCC bond types. One type per line. This exactly matches the bond list found in the zinc.sdf file.
- mmff94.sdf [1628kB] : MMFF94 validation : three dimensional structures in the MDL SD format.
- mmff94_p_f_types.txt [48kB] : The **parm@Frosst** atom types of MMFF94 set. One type per line. This exactly match the atom ordering found in the mmff94.sdf file.

You can also retrieve all the files listed above as a compressed tar (parm_at_Frosst.tgz) or a zip (parm_at_Frosst.zip) archive.

From parm@frosst (originally based on PATTY types), Bayly/Mobley/Bannan produced SMIRNOFF version (smirnoff99frosst) with same parameters but enormously simplified typing rules.

Ditching “atom types” for SMIRKS (“parameter types”) allows parameter compression

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)

This has larger implications for angles

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

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

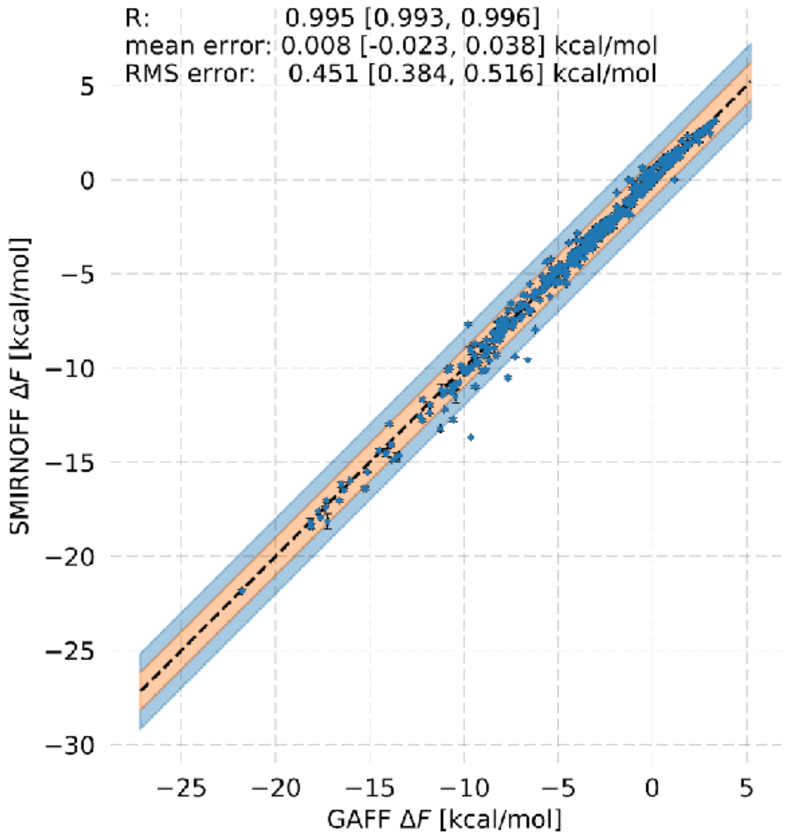
...and for torsions

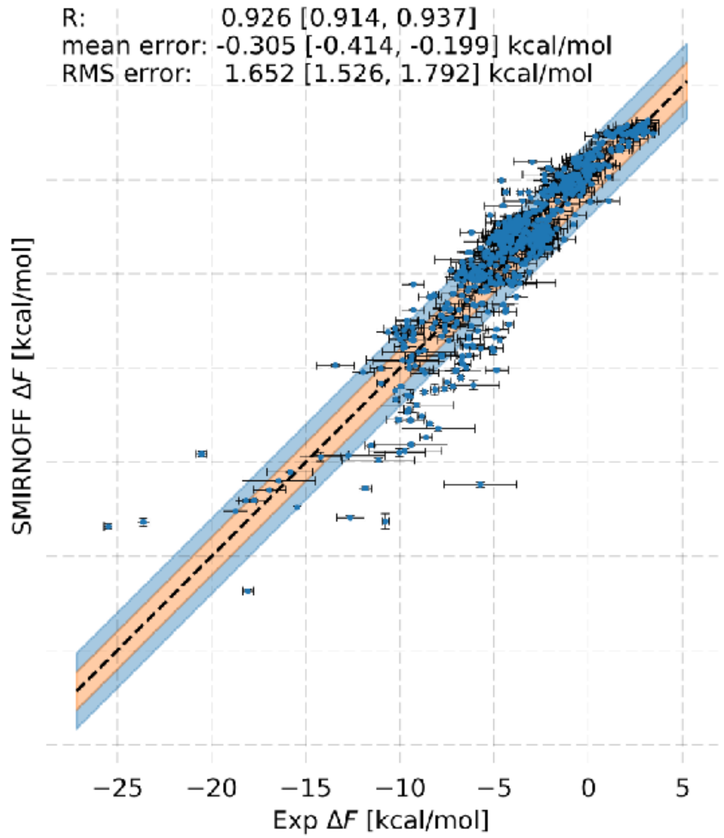
```
X -n1-c2-X    2    0.000       180.000         2.000
X -n1-c3-X    3    0.000       180.000         2.000
X -n1-ca-X    2    0.000       180.000         2.000
X -n1-cc-X    2    0.000       180.000         2.000
X -n1-cd-X    2    0.000       180.000         2.000
X -n1-ce-X    2    0.000       180.000         2.000
X -n1-cf-X    2    0.000       180.000         2.000
X -n1-cu-X    2    0.000       180.000         2.000
X -n1-cv-X    2    0.000       180.000         2.000
X -n1-cx-X    3    0.000       180.000         2.000
X -n1-cy-X    3    0.000       180.000         2.000
X -n1-n -X    2    0.000       180.000         2.000
X -n1-n1-X    1    0.000       180.000         2.000
X -n1-n2-X    1    0.000       180.000         2.000
X -n1-n3-X    2    0.000       180.000         2.000
X -n1-n4-X    3    0.000       180.000         2.000
X -n1-na-X    2    0.000       180.000         2.000
X -n1-nb-X    2    0.000       180.000         2.000
X -n1-nc-X    2    0.000       180.000         2.000
X -n1-nd-X    2    0.000       180.000         2.000
X -n1-ne-X    2    0.000       180.000         2.000
X -n1-nf-X    2    0.000       180.000         2.000
X -n1-nh-X    2    0.000       180.000         2.000
```

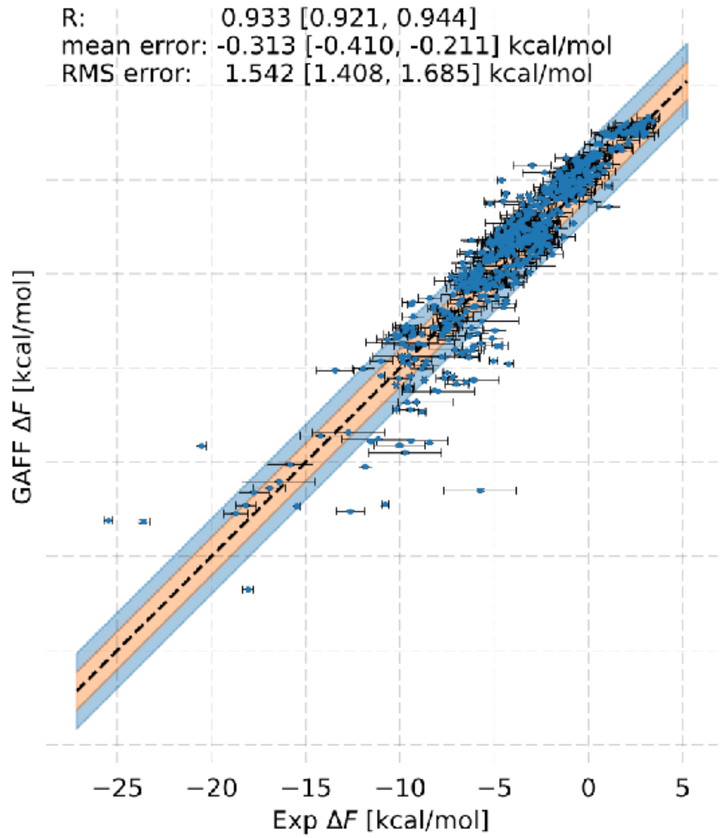
GAFF has a lot of this

smirnoff99frosst performs as well as GAFF but is 300 lines instead of 6000+ lines

FreeSolv hydration free energy benchmark





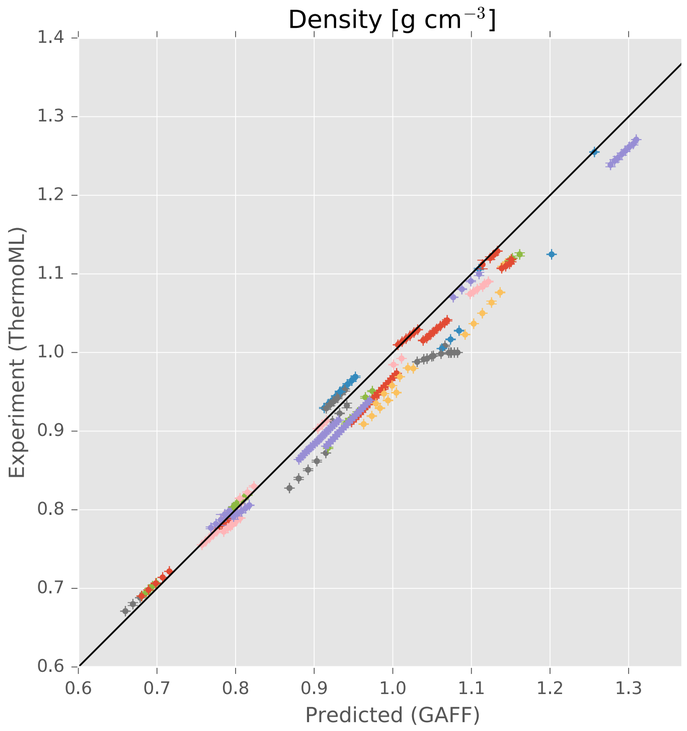


http://github.com/openforcefield/smirnoff99Frosst

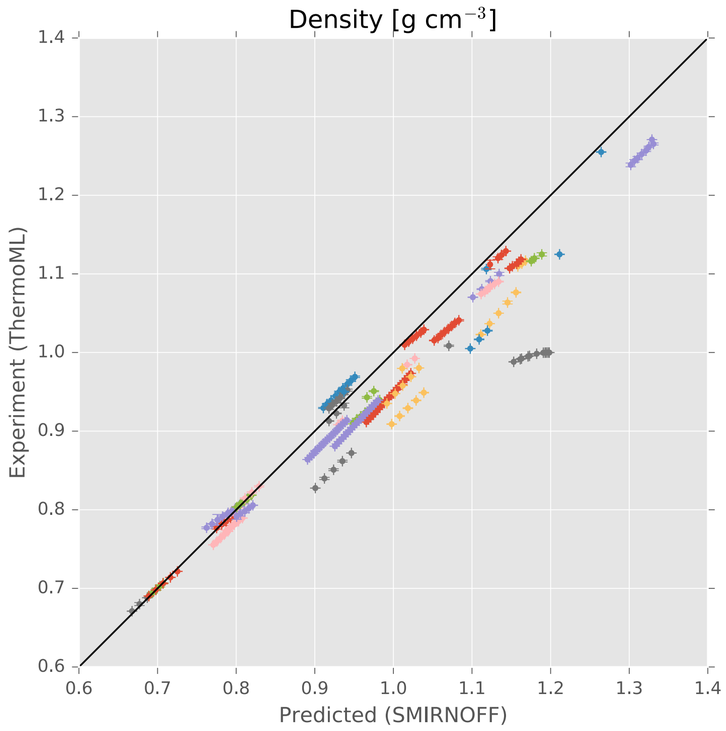
smirnoff99frosst performs as well as GAFF but is 300 lines instead of 6000+ lines

ThermoML Archive density/dielectric benchmark set

GAFF



smirnoff99frosst

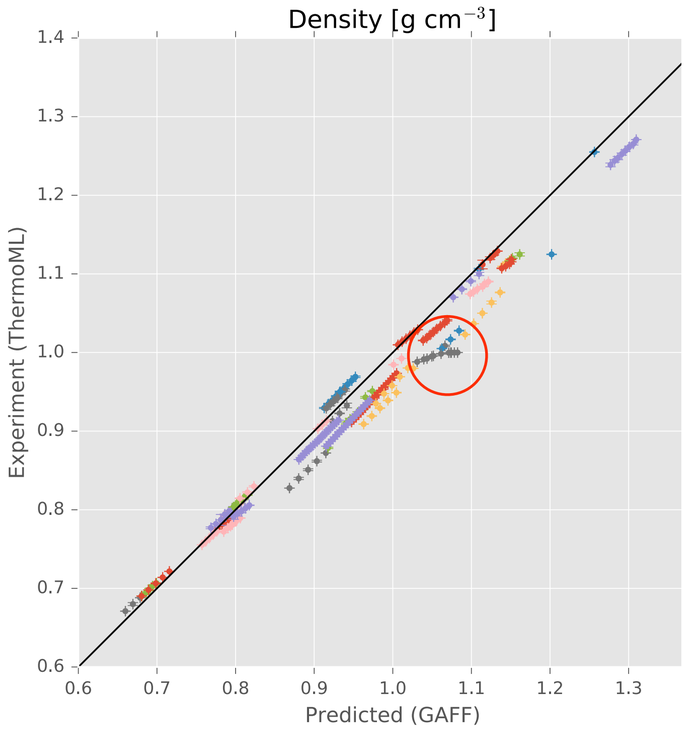


http://github.com/openforcefield/smirnoff99Frosst

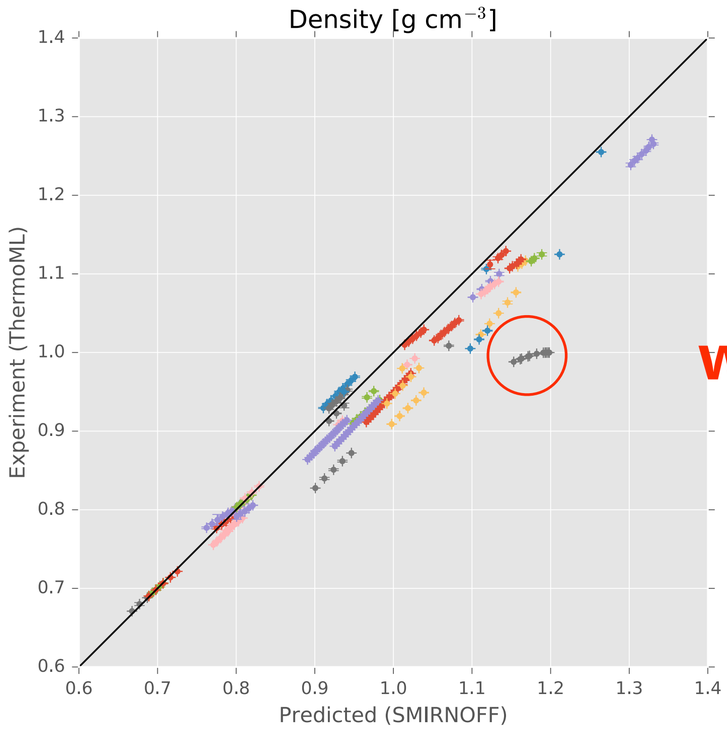
smirnoff99frosst performs as well as GAFF but is 300 lines instead of 6000+ lines

ThermoML Archive density/dielectric benchmark set

GAFF



smirnoff99frosst



water

http://github.com/openforcefield/smirnoff99Frosst

smirnoff99frosst has good coverage of chemistry

No metals, metaloids

Can type most of DrugBank (excluding metals and boron)

Can type most of eMolecules (exact numbers forthcoming)

Compatible with AMBER99 and derivative biomolecular forcefields

Using SMIRNOFF is easy

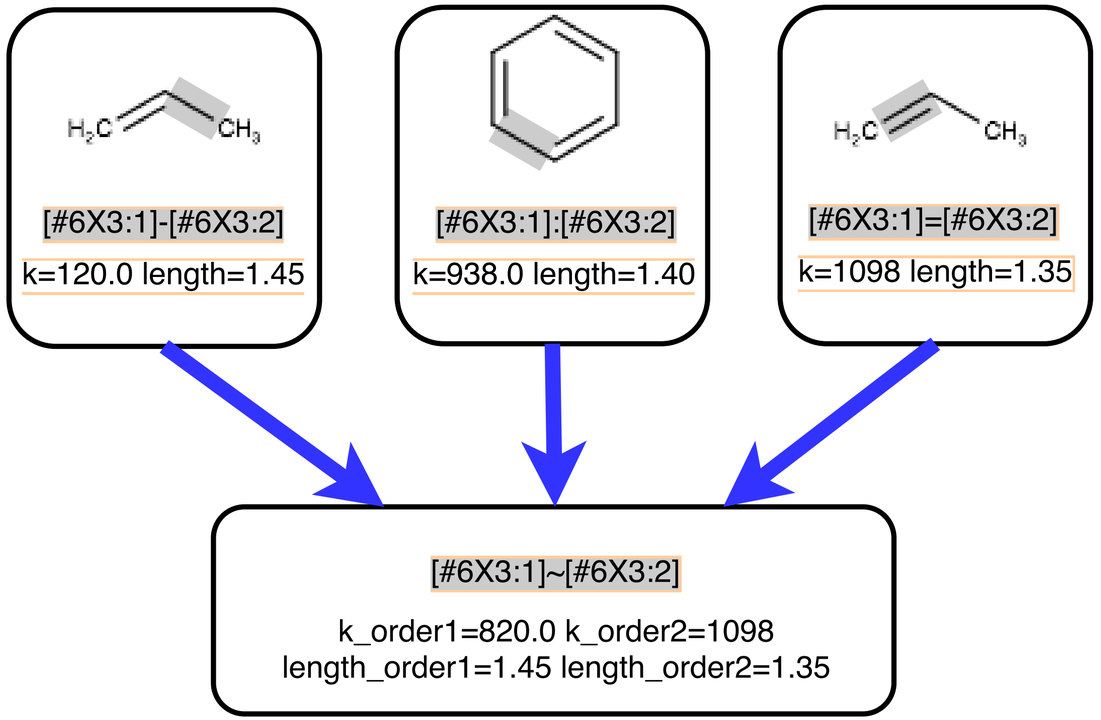
```
# Load the smirnoff99frosst forcefield
from openforcefield.typing.engines.smirnoff import ForceField
forcefield = ForceField('smirnoff99frosst.ffxml')
# Create an OpenMM System object to simulate
system = forcefield.createSystem(topology)
```

Currently based on oechem, but now adding rdkit support

Can **export to all major simulation codes** (OpenMM, Amber, CHARMM, gromacs, NAMD, LAMMPS, Desmond) using ParmEd and InterMol conversion tools

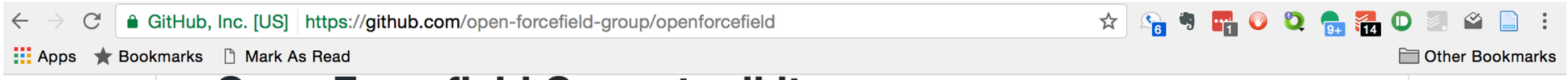
https://github.com/openforcefield/openforcefield

We can also simplify further by building in more chemistry, like partial bond orders



We can use partial bond orders (Wiberg bond orders) to get bond orders specific to each molecule for interpolation

Tools for this are available for free online



Open Forcefield Group toolkit

This repository contains a number of tools from the Open Force Field Group for the development and use of modern molecular mechanics forcefields based on direct chemical perception and parameterized with rigorous statistical methods.

This repository hosts tools that we have committed to stably maintain throughout their lifetimes:

- The SMIRks Native Open Force Field (SMIRNOFF) direct chemical perception forcefield specification language
- Tools for direct chemical environment perception and manipulation

Installation

We recommend the miniconda Python distribution. To install `miniconda` on `osx` with `bash`, this is:

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-MacOSX-x86_64.sh
bash Miniconda3-latest-MacOSX-x86_64.sh -b -p $HOME/miniconda3
export PATH="$HOME/miniconda3/bin:${PATH}"
```

These tools currently require the OpenEye toolkit (which requires a license that is free for academics indenting to rapidly release results into the public domain):

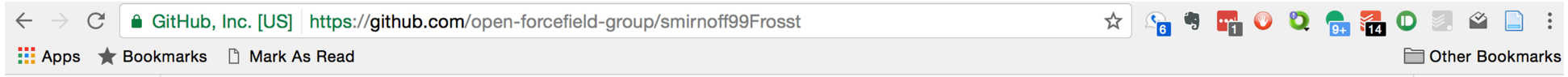
```
pip install -i https://pypi.anaconda.org/OpenEye/simple OpenEye-toolkits
```

Install `openforcefield` tools via conda:

```
conda install --yes -c omnia openforcefield
```

github.com/openforcefield/openforcefield

We have a basic small molecule force field using these ideas, smirnoff99Frosst



smirnoff99Frosst

This provides the first general-purpose implementation of a Smirks Native Open Force Field (SMIRNOFF) as implemented by SMARTY and its ForceField class (in smarty.forcefield) for parameterizing small molecules for OpenMM.

Latest release: DOI 10.5281/zenodo.348165

Installation

```
conda install -c mobleylab smiff99frosst=1.0.4
```

(smirnoff99frosst was formerly known as smirff99frosst)

Covers all of our filtered DrugBank set

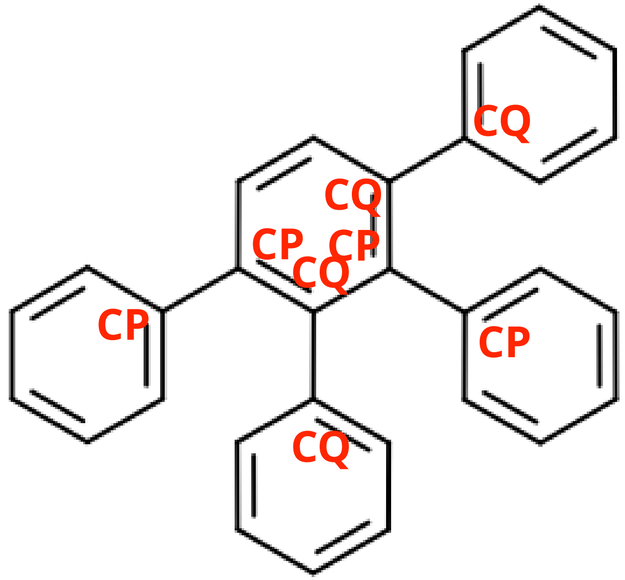
What it is

The provided smirff99Frosst.xml (forcefield) is a starting point for a general-purpose small molecule force field in the SMIRFF format; it should cover all or almost all of drug-like chemical space, and illustrates some of the major functionality of the SMIRFF format as well as how it simplifies the specification of force field parameters in a compact and chemically sensible way.

HOWEVER, this is not expected to be (at present) an especially accurate small molecule force field. Its authors (see History, below) expect that while coverage will initially be good, additional refinements will be required (and possibly some expansion of the number of parameters) before it can rival current force fields such as GAFF or OPLS in accuracy. However, we are optimistic that it already rivals them in extensibility, and potentially with relatively minimal work can be extended to be a compelling present-day small molecule force field.

github.com/openforcefield/smirnoff99Frosst

smirnoff99Frosst also fixes issues with GAFF/GAFF2



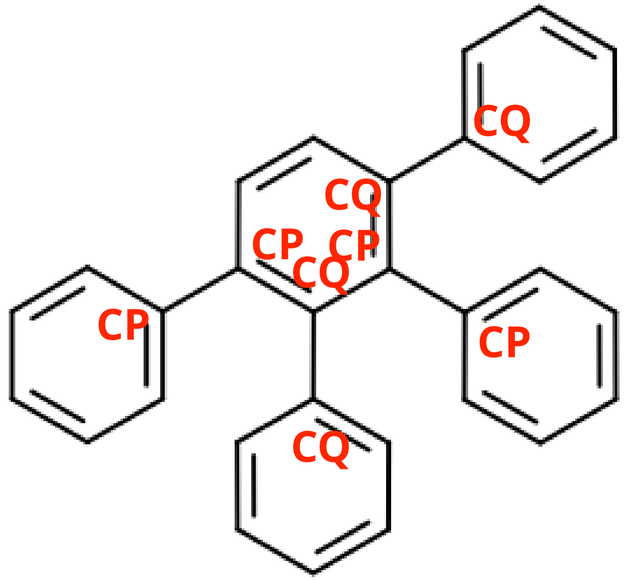
CA: aromatic sp2 carbon

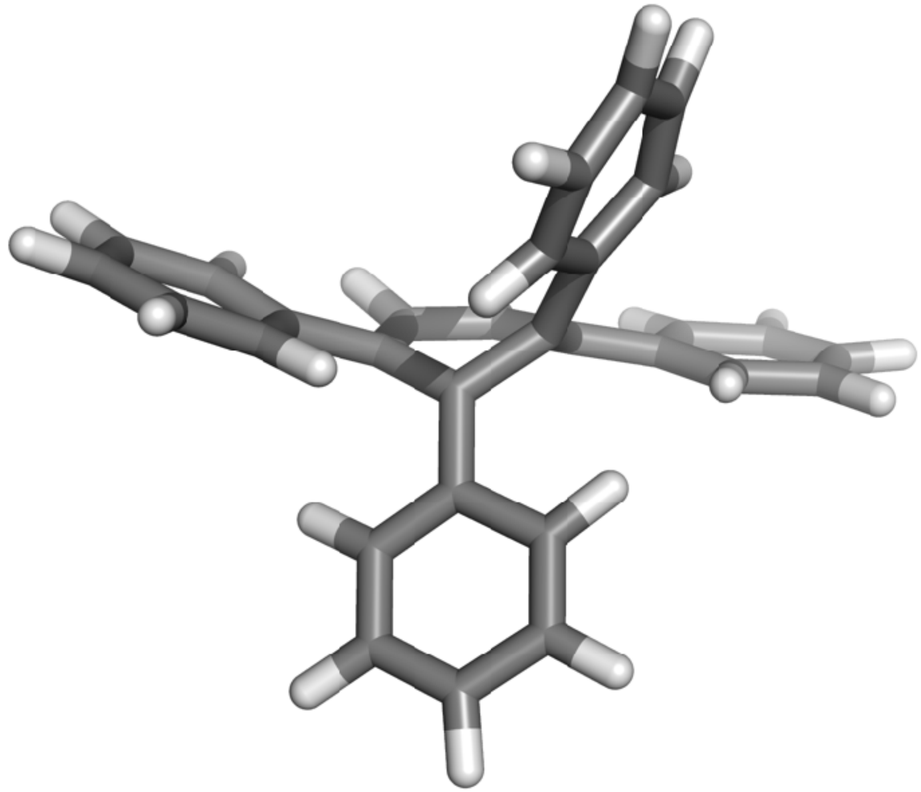
CP: aromatic sp2 carbon joined to another aromatic ring

CQ: Same as CP except CP-CQ bonds are aromatic

- In GAFF/GAFF2, these are handled by cp and cq, and all of the torsional combinations of ca, cp, and cq must be correct for correct conformational preferences (roughly 3*3*3*3/2 combinations, or around 40)
- Human error can easily creep in

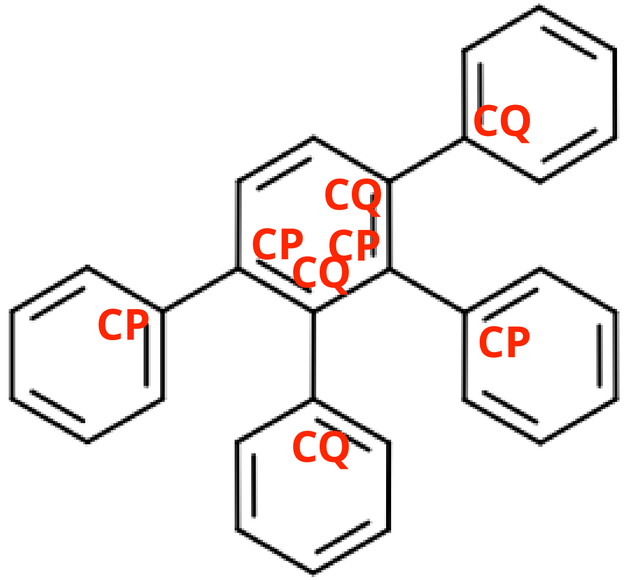
GAFF/GAFF2 mis-assigns some torsions resulting in buckling of the central ring





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

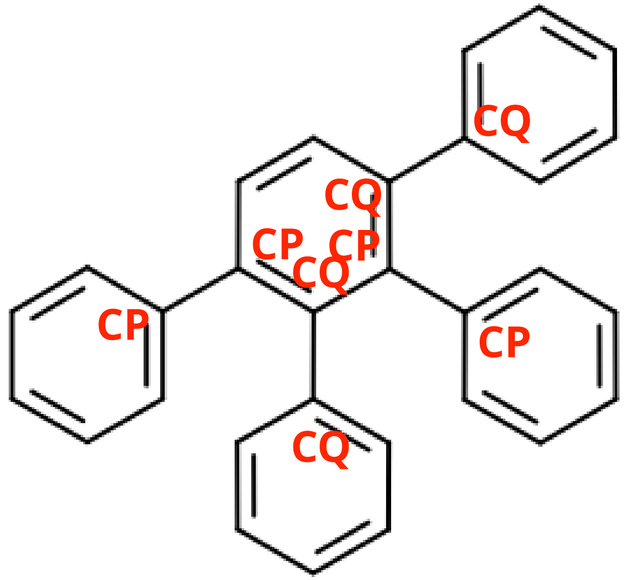
Human error is far too easy in such cases

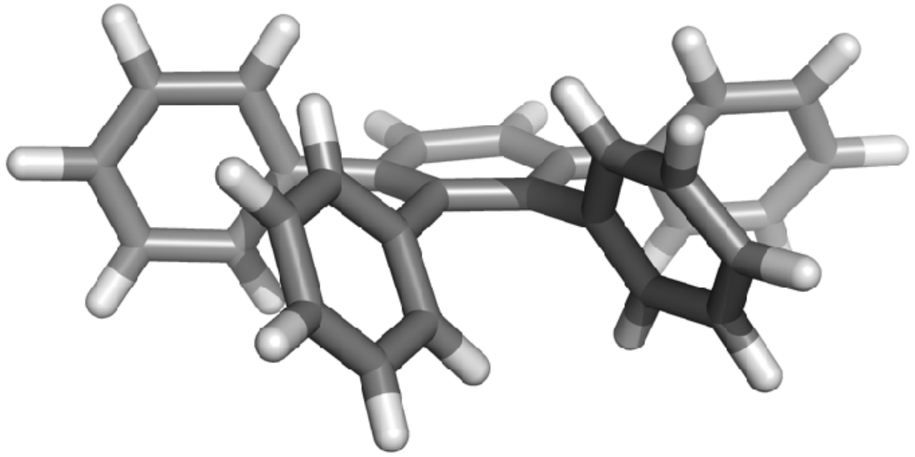


- Aromatics (partial list):
  - cq-cq-cp-cp
  - ca-cq-cp-cp
  - cq-cp-cq-cq
  - cp-cq-cp-cq
- Singles (partial list):
  - cq-cp-cp-ca
  - cq-cp-cp-ca
  - ca-cq-cq-ca

- parmchk2 for GAFF/GAFF2 assign many of these incorrectly
- Yet GAFF used biphenyls in parameterization! Logic errors

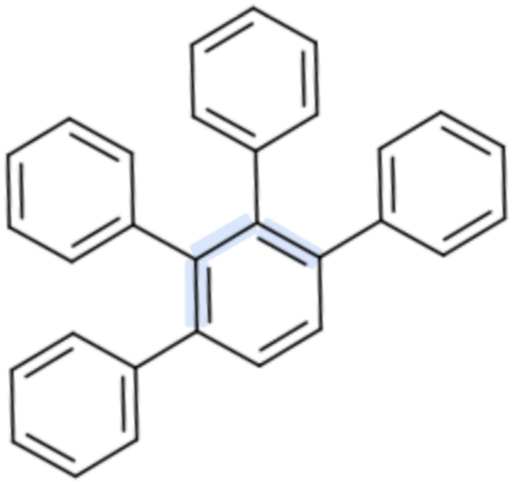
smirff99Frosst gets it right without any particular attention to this specific case



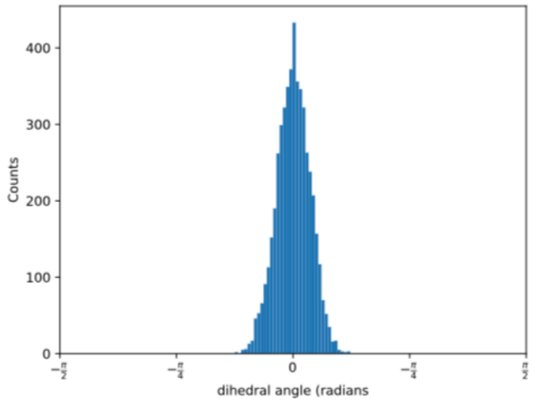


- We have a [*:1]~[#6X3:2]:[#6X3:3]~[*:4] generic for aromatics that treats the ring bonds separately

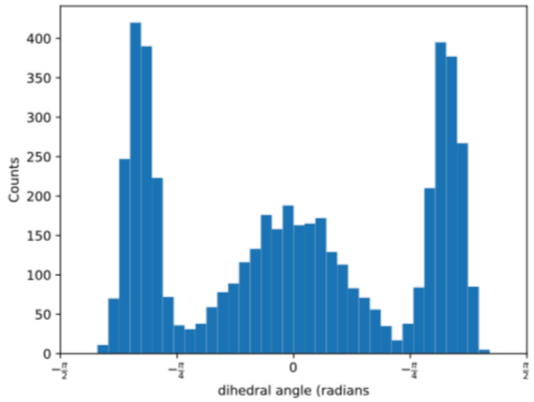
smirff99Frosst gets it right without any particular attention to this specific case



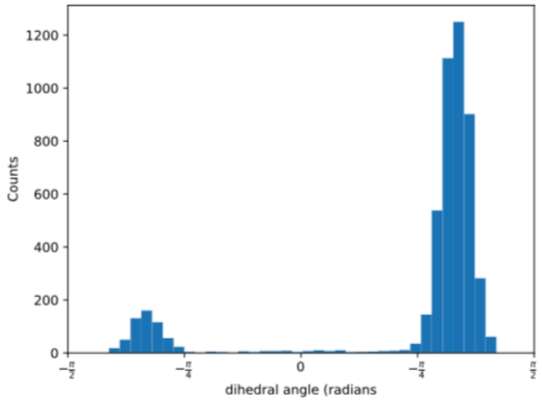
(a) 1,2,3,4-tetraphenylbenzene torsion



(b) smirnoff99frosst

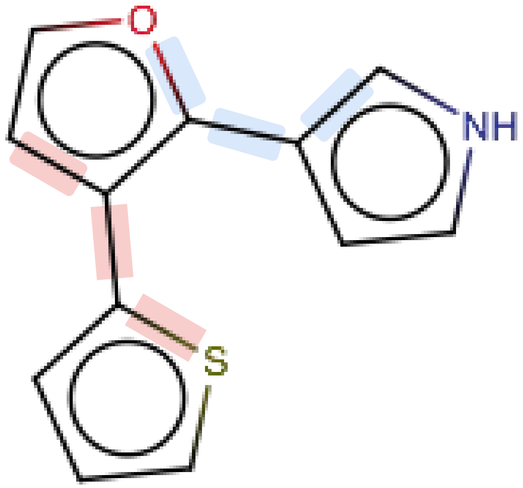


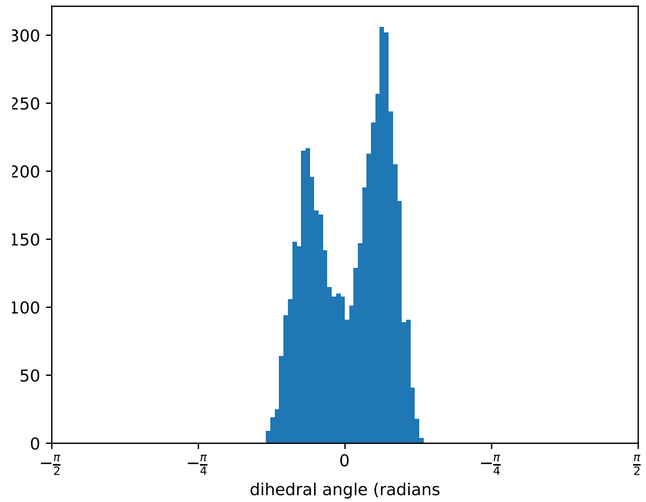
(c) GAFF

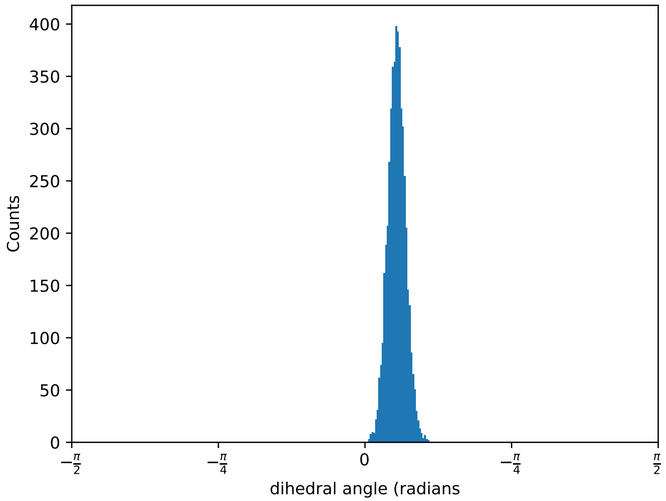


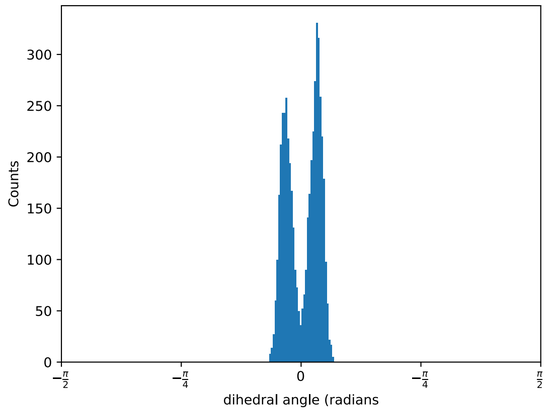
(d) GAFF2

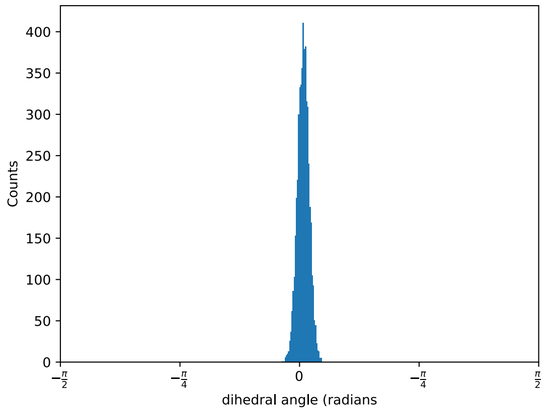
The “biphenyl problem” was “dealt with” in GAFF, but it has siblings which were overlooked





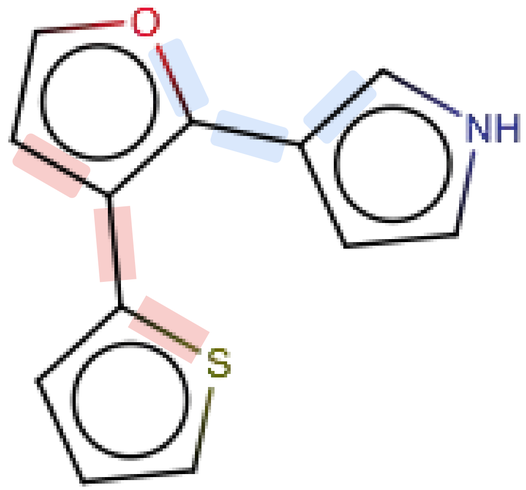


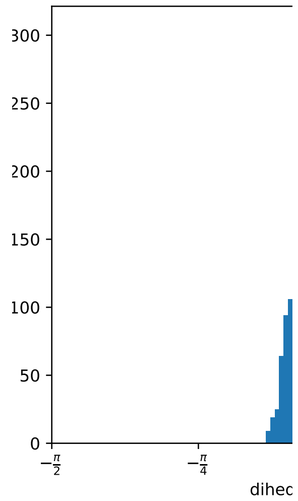


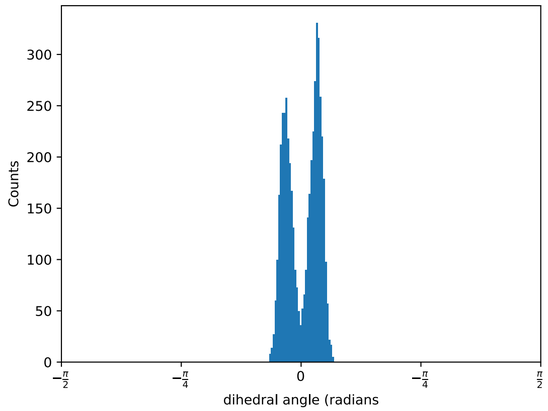


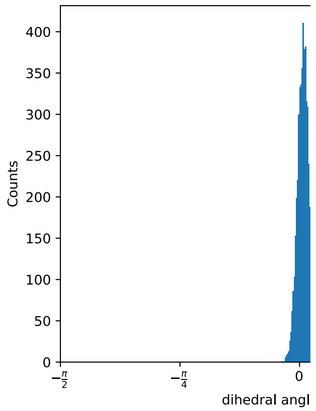
- SMIRNOFF makes these rotatable single bonds
- GAFF/GAFF2 make them essentially aromatic
- Central ring buckles due to steric strain

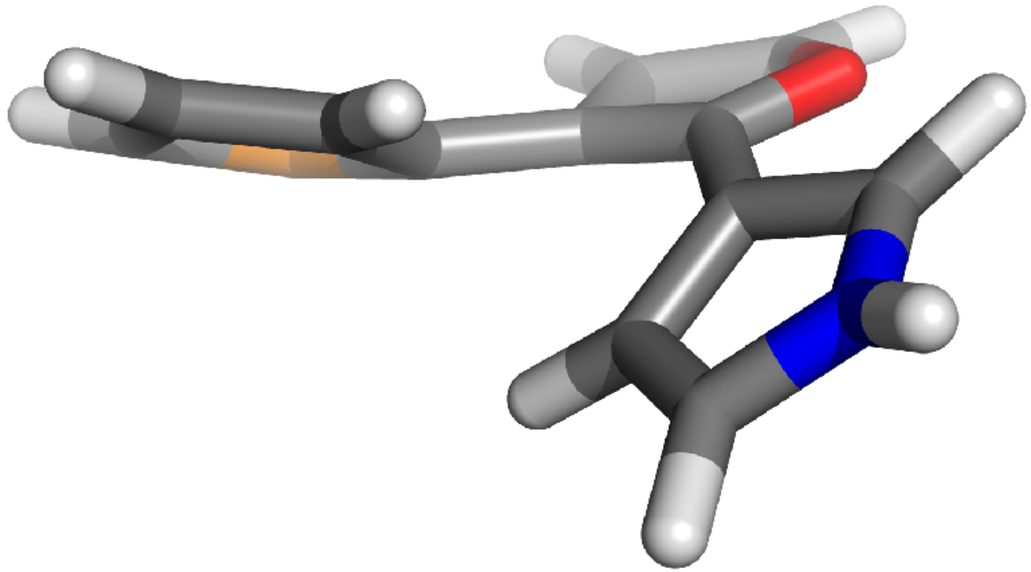
The “biphenyl problem” was “dealt with” in GAFF, but it has siblings which were overlooked



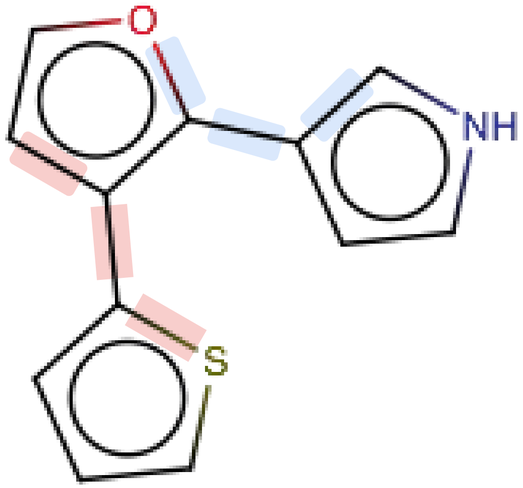


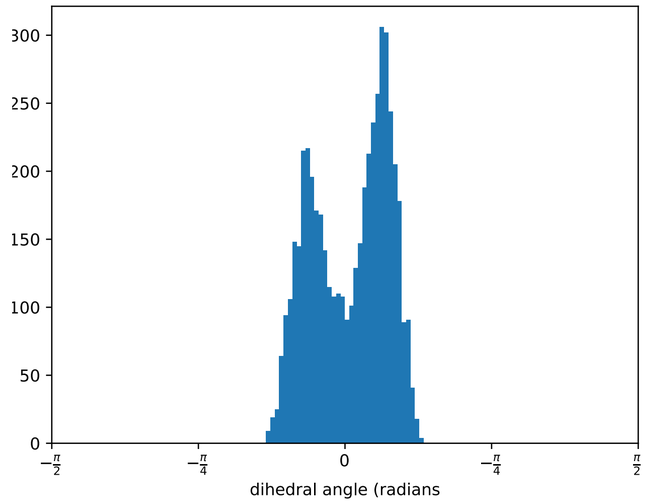


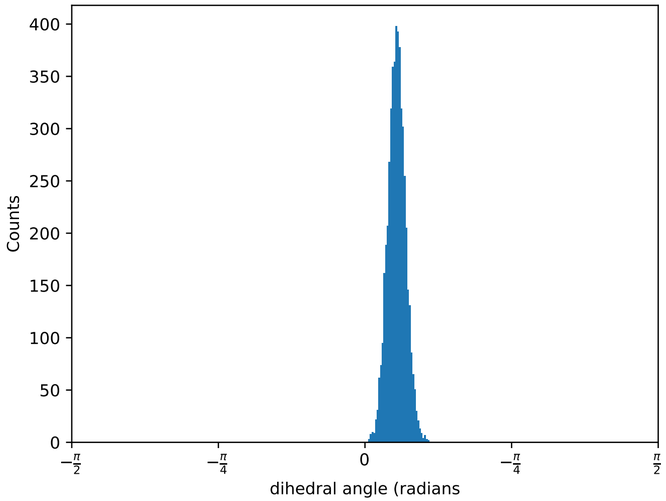


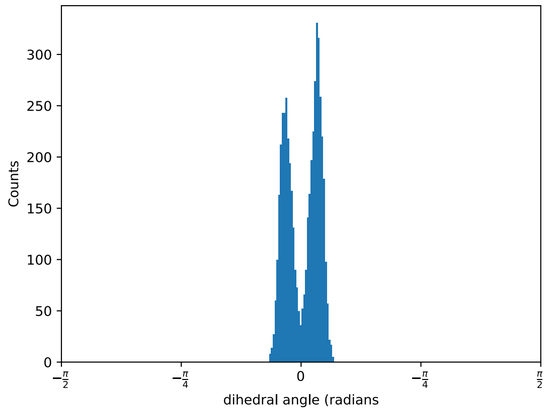


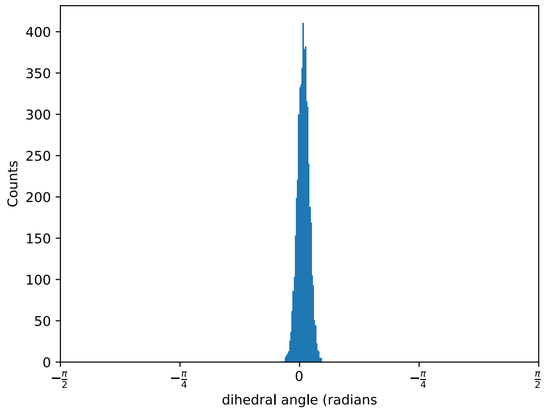
The “biphenyl problem” was “dealt with” in GAFF, but it has siblings which were overlooked





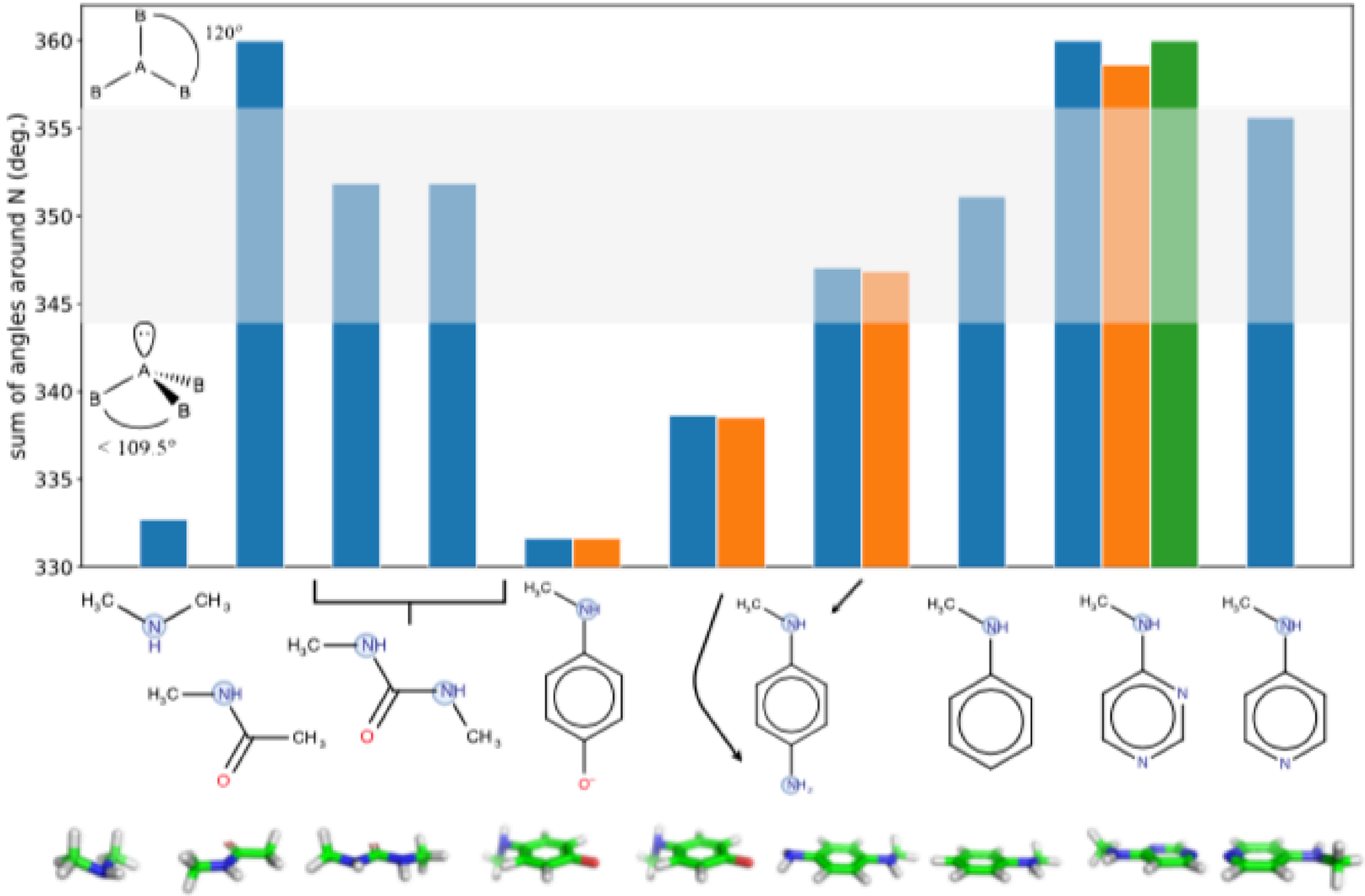






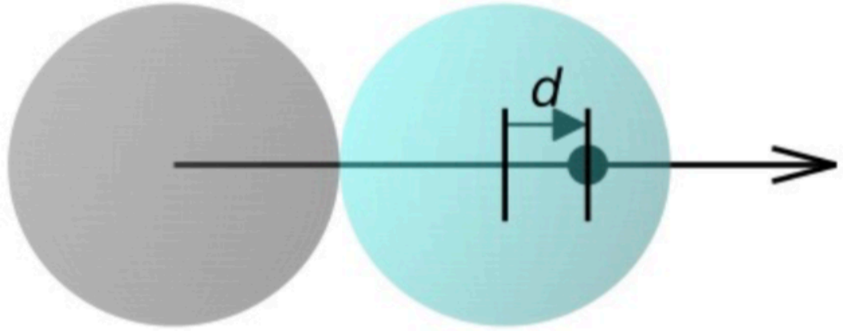
- SMIRNOFF makes these rotatable single bonds
- GAFF/GAFF2 make them essentially aromatic
- Central ring buckles due to steric strain

SMIRNOFF also opens up interesting avenues for new force field science, such as partial planarity

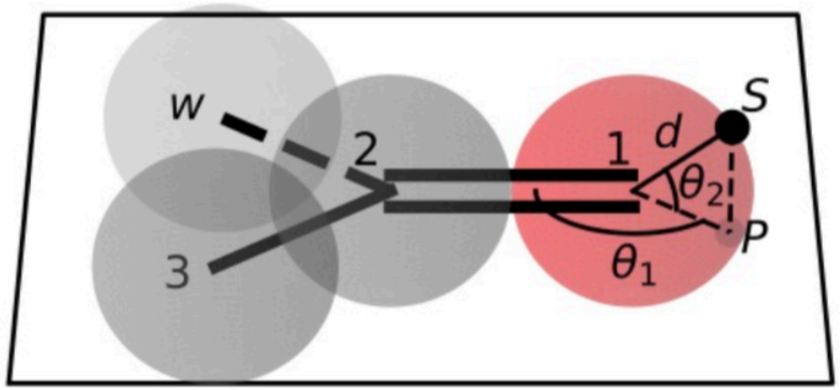


- We can interpolate based on partial bond order around the nitrogen center
- Connecting this to impropers should allow partially planar nitrogens

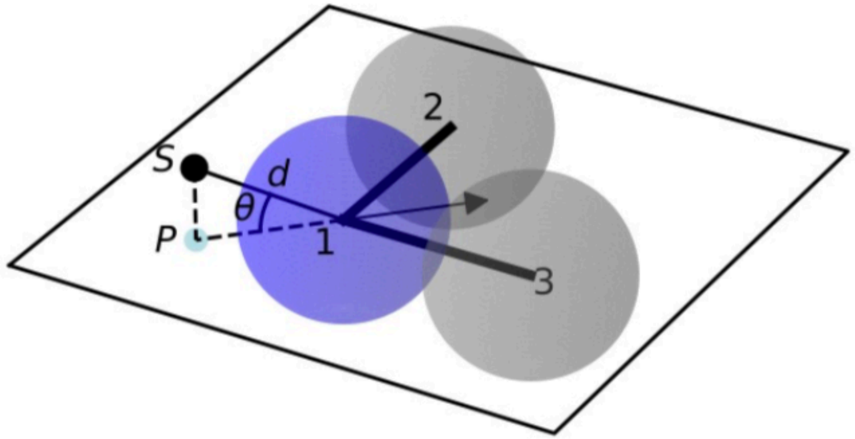
We also plan a variety of virtual sites for off-center charges



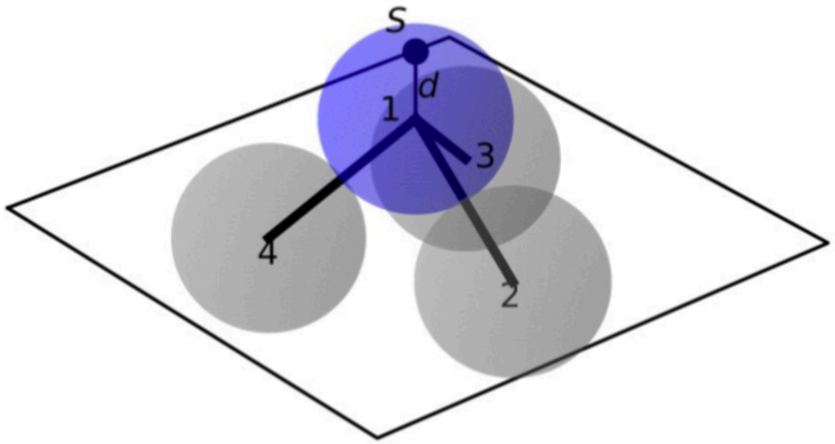
(a) `BondCharge` virtual site



(b) `MonovalentLonePair` virtual site



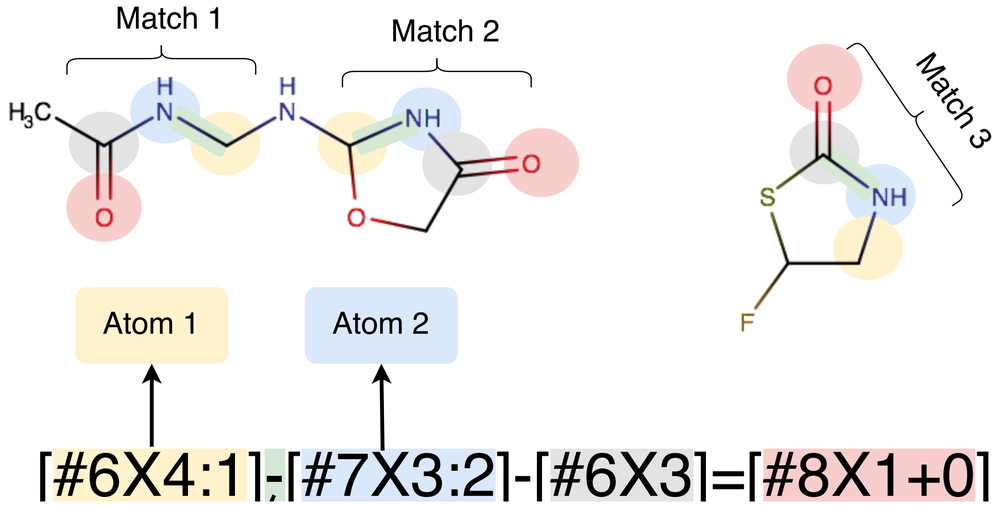
(c) `DivalentLonePair` virtual site



(d) `TrivalentLonePair` virtual site

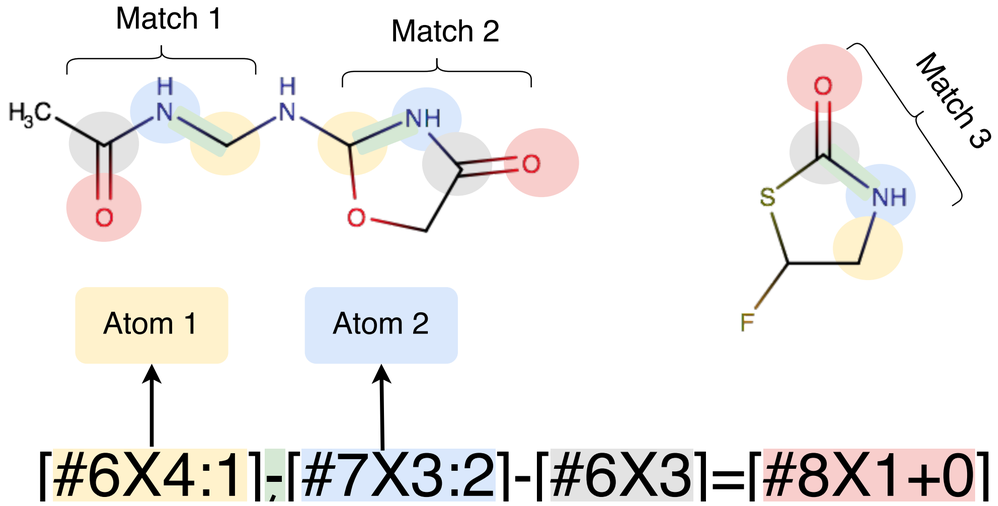
You can use SMIRNOFF now, but it’s an exciting new avenue for force fields

You can use SMIRNOFF now, but it’s an exciting new avenue for force fields



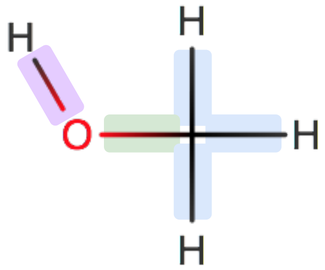
SMIRKS provide a language for direct chemical perception

You can use SMIRNOFF now, but it's an exciting new avenue for force fields



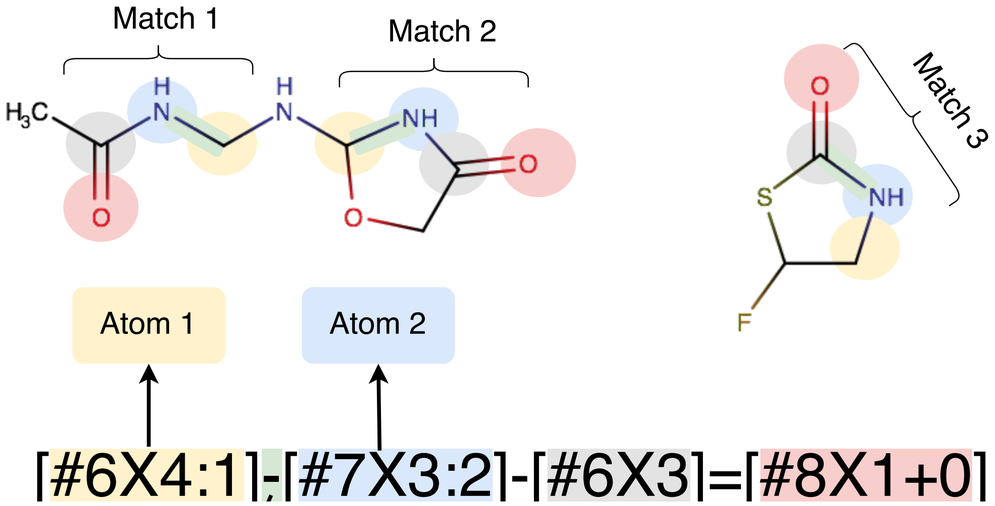
SMIRKS provide a language for direct chemical perception

```
<?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>
```



Thus we have a new force field format

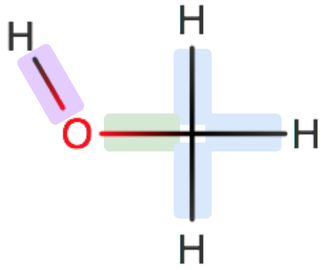
You can use SMIRNOFF now, but it's an exciting new avenue for force fields

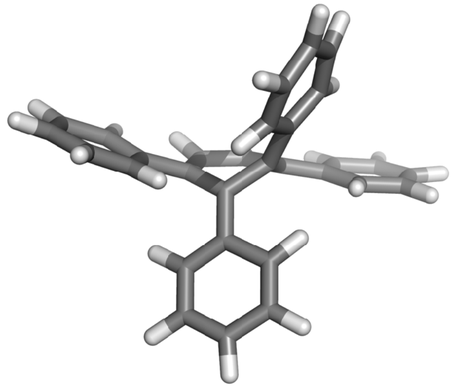


SMIRKS provide a language for direct chemical perception

```
<?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>
```

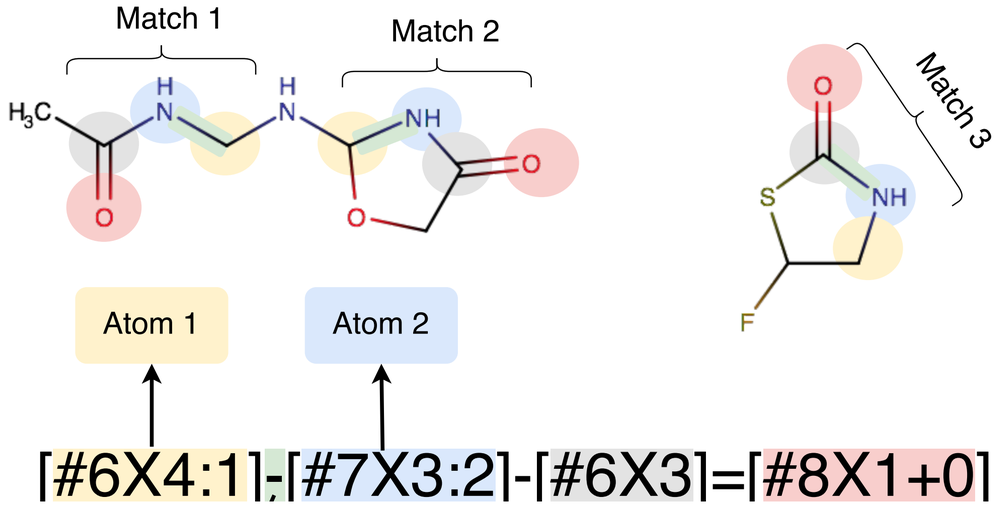
Thus we have a new force field format



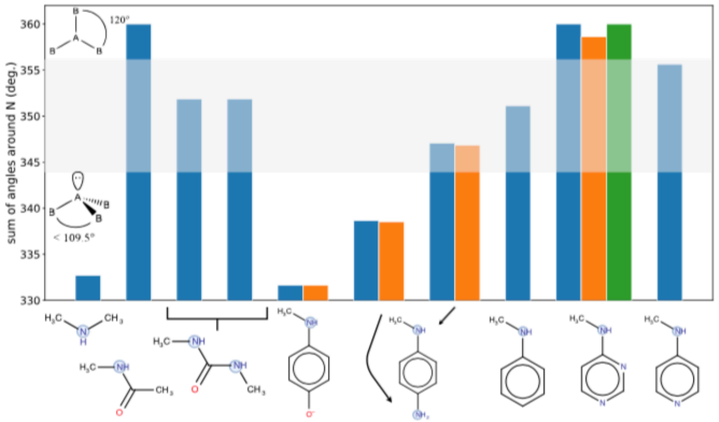


And a new FF which fixes some problems with GAFF

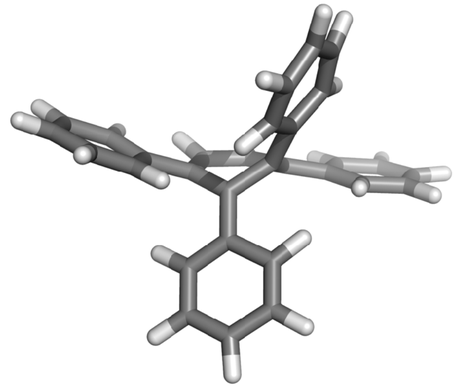
You can use SMIRNOFF now, but it's an exciting new avenue for force fields



SMIRKS provide a language for direct chemical perception

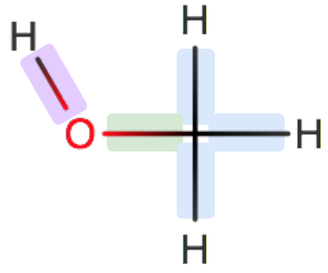


...while opening up possibilities for new science



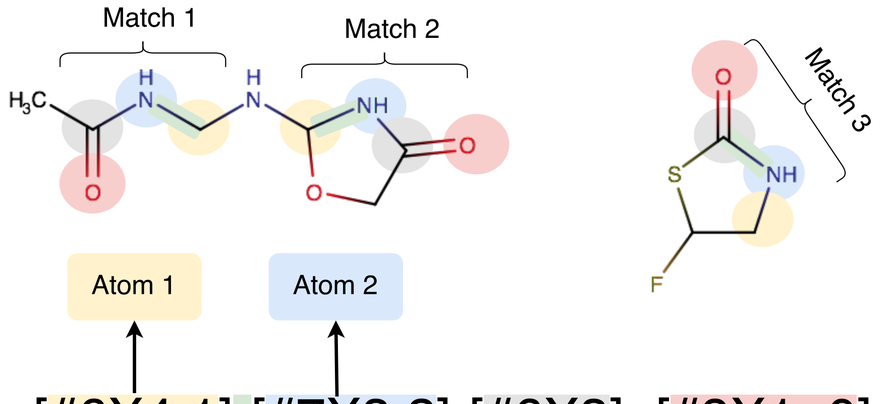
```
<?xml version="1.0"?>
<SMIRNOFF>

<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>
```

Thus we have a new force field format



And a new FF which fixes some problems with GAFF

We want to automate force field development



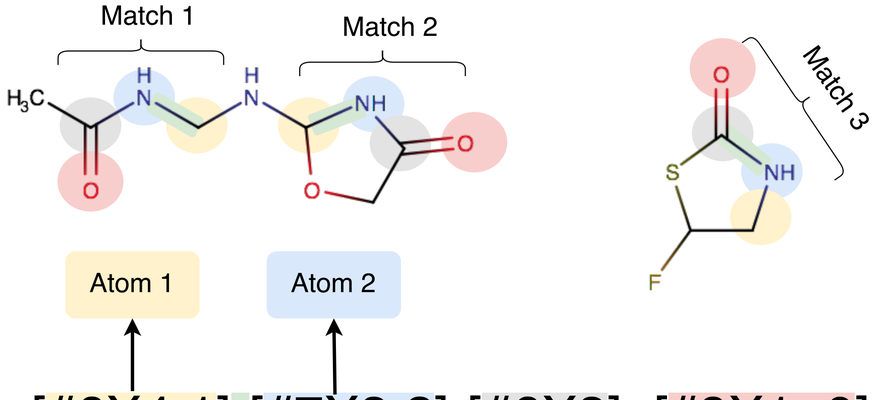
[#6X4:1]-[#7X3:2]-[#6X3]=[#8X1+0]

SMIRKS provide a language for this

We'll rely extensively on NIST's ThermoML, and on QM data

We want to automate force field development

We want to automate FF development



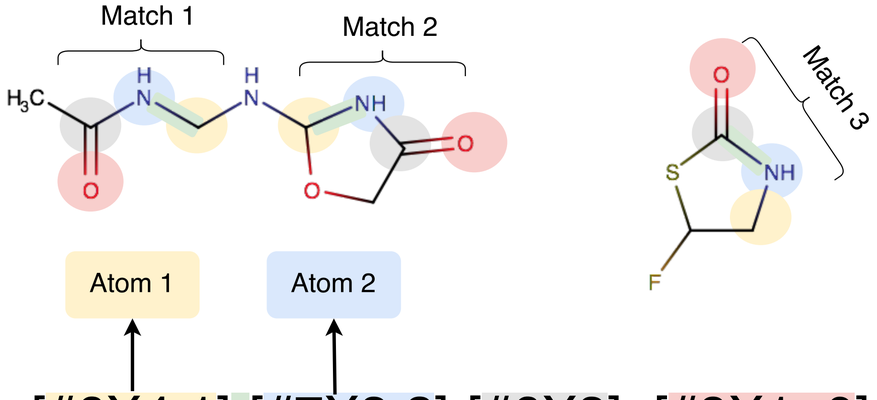
SMIRKS provide a language for this

We'll rely extensively on NIST's ThermoML, and on QM data

We want to automate force field development

We want to automate FF development

Part of this involves learning chemical perception



SMIRKS provide a language for this

We'll rely extensively on NIST's ThermoML, and on QM data

Acknowledgments

- NSF, NIH funding for issues directly or peripherally related
- Collaborators in the effort:
  - John Chodera and his lab (Josh Fass, Chaya Stern, Andrea Rizzi, …)
  - Michael Shirts and his lab (Bryce Manubay)
  - Christopher Bayly
  - Michael K. Gilson
  - Mobley group: Caitlin Bannan, Camila Zanette, Victoria Lim, Gaetano Calabro (now OE), Guilherme Matos
  - Peter Eastman
  - Kyle Beauchamp