Toward more accurate force fields through direct chemical perception: How to get rid of wizards

David L. Mobley, Caitlin C. Bannan, Camila Zanette, Christopher I. Bayly, John D. Chodera, Michael R. Shirts, Josh Fass, Bryce Manubay, Peter Eastman, and Michael K. Gilson
Force field development traditionally takes a heroic, monumental effort and many person-years of experimental data, quantum chemistry, and keen chemical intuition. A parameter set we desperately hope someone uses comes from heroic effort by graduate students and postdocs.
Force field development traditionally takes a heroic, monumental effort and many person-years. Experimental data, quantum chemistry, and keen chemical intuition are combined through a heroic effort by graduate students and postdocs. A parameter set we desperately hope someone uses.
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!

Slide: J. Chodera
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!

Slide: J. Chodera
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

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.

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 **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.

\[ \theta^* \rightarrow \text{MD} \rightarrow \langle A \rangle_* \]
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
- \( \theta \): 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 results provide insight into chemistry where parameters (and thus predictions) are uncertain.
Thus there are several key steps towards this new world

Pipeline needs to be independent of chemical expertise

Bayesian parameterization framework

Input data: That’s your decision/our debate (but note NIST’s ThermoML)
Today, we’re going to talk about why we have wizards, and how to get rid of them.

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

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

C-C single bond!

<table>
<thead>
<tr>
<th>Bond Configuration</th>
<th>Length</th>
<th>Angle</th>
<th>Bend</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-CA-CA-X</td>
<td>14.50</td>
<td>180.0</td>
<td>2.</td>
</tr>
<tr>
<td>CT-CT-CT-CT</td>
<td>0.18</td>
<td>0.0</td>
<td>-3.</td>
</tr>
<tr>
<td>CT-CT-CT-CT</td>
<td>0.25</td>
<td>180.0</td>
<td>-2.</td>
</tr>
<tr>
<td>CT-CT-CT-CT</td>
<td>0.20</td>
<td>180.0</td>
<td>1.</td>
</tr>
</tbody>
</table>
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

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
cca-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?

Aromatic-aromatic bond (& torsion)

Direct chemical perception: Assign parameters based on analysis of the molecule rather than atom types
We just need a language for direct chemical perception. SMARTS/SMIRKS provide this

SMARTS are similar to SMILES but for fragments
Atoms are enclosed in brackets [ ], such as [#6] for carbon

os-c3-c3-oh
[#8]-[#6]-[#6]-[#8]
SMIRKS, for our purposes, are the same as SMARTS but with atom labels

\[
\text{os-c3-c3-oh} \\
[#8:1]-[#6:2]-[#6:3]-[#8:4]
\]
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
The new SMIRKS native open force field (SMIRNOFF) format assigns parameters via SMIRKS

<?xml version="1.0"?>
<SMIRFF>
  <HarmonicBondForce length_unit="angstroms" k_unit="kilocalories_per_mole/angstrom**2">
    <Bond smirks="[#6X4:1]-[#6X4:2]" length="1.526" k="620.0"/>
    <Bond smirks="[#6X4:1]-[#1:2]" length="1.090" k="680.0"/>
    <Bond smirks="[#8:1]~[#1:2]" length="1.410" k="640.0"/>
    <Bond smirks="[#6X4:1]-[#8&X2&H1:2]" length="1.410" k="640.0"/>
    <Bond smirks="[#6X4:1]-[#8&X2&H0:2]" length="1.370" 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"/>
    ...
  </HarmonicAngleForce>
</SMIRFF>
The new SMIRKS native open force field (SMIRNOFF) format assigns parameters via SMIRKS...

<PeriodicTorsionForce phase_unit="degrees" k_unit="kilocalories_per_mole">
  <Proper smirks="[a,A:1]-[#6X4:2]-[#6X4:3]-[a,A:4]" idivf1="9" periodicity1="3" phase1="0.0" k1="1.40"/>
  <Proper smirks="[a,A:1]-[#6X4:2]-[#8X2:3]-[#1:4]" idivf1="3" periodicity1="3" phase1="0.0" k1="0.50"/>
  <Proper smirks="[a,A:1]-[#6X4:2]-[#8X2:3]-![#1:4]" idivf1="3" periodicity1="3" phase1="0.0" k1="1.15"/>
  ...
  <Improper smirks="[a,A:1]~[#6X3:2]([a,A:3])~[#8X1:4]" periodicity1="2" phase1="180.0" k1="10.5"/>
</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="[#8X2+0$(*-[#1]):1]" rmin_half="1.7210" epsilon="0.2104"/>
</NonbondedForce>
A hand-constructed version reproduces FF on our AlkEthOH test set with just 35 lines of parameters.
Ditching “atom types” for SMIRKS (“parameter types”) allows parameter compression.

<table>
<thead>
<tr>
<th>Atom Type</th>
<th>Radius 1</th>
<th>Sigma 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cs</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>ca</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cc</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cd</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>ce</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cf</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cp</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cq</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cz</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cu</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cv</td>
<td>1.8606</td>
<td>0.0988</td>
</tr>
<tr>
<td>cg</td>
<td>1.9525</td>
<td>0.1596</td>
</tr>
<tr>
<td>ch</td>
<td>1.9525</td>
<td>0.1596</td>
</tr>
<tr>
<td>cx</td>
<td>1.9069</td>
<td>0.1078</td>
</tr>
<tr>
<td>cy</td>
<td>1.9069</td>
<td>0.1078</td>
</tr>
</tbody>
</table>

For example, GAFF2 has 16 vdW types for carbon. 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.)
We can also simplify further by building in more chemistry, like partial bond orders.

Three carbon-carbon bonds:

\[
\begin{align*}
[#6X3:1]-[#6X3:2] & : 820.0 \quad 1.45 \\
[#6X3:1]:[#6X3:2] & : 938.0 \quad 1.40 \\
[#6X3:1]=[#6X3:2] & : 1098.0 \quad 1.35
\end{align*}
\]

Can become one bond via interpolation:

\[
\text{<Bond smirks="[#6X3:1]!#[6X3:2]" id="b5" k_order1="820.0" k_order2="1098" length_order1="1.45" length_order2="1.35"/>}
\]

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 intending 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/open-forcefield-group/openforcefield
We have a basic small molecule force field using these ideas, smirnoff99Frosst

Covers all of our filtered DrugBank set


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.


Installation

conda install -c mobleylab smirff99frosst=1.0.4

(smirnoff99frosst was formerly known as smirff99frosst)

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/open-forcefield-group/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 $3^3*3^3*3/2$ 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
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
We also have comparable accuracy to GAFF on densities and dielectrics of 45 neat liquids

Beauchamp et al., J Phys Chem B 2015, 10.1021/acs.jpcb.5b06703
We also have comparable accuracy to GAFF on densities and dielectrics of 45 neat liquids.

Beauchamp et al., J Phys Chem B 2015, 10.1021/acs.jpcb.5b06703
We also have comparable accuracy to GAFF on densities and dielectrics of 45 neat liquids.
Ultimately, getting away from atom typing makes parameter assignment simpler

- With knowledge of which bonds are single, double, aromatic, etc., parameter assignment is straightforward
- Without it, atom types must “carry” bond order information, which is almost impossible to do in general
- SMIRNOFF force fields handle this seamlessly
So, we have a language that can express chemical perception (even if it does look a bit like profanity)

To get rid of wizards, we need:
• A language to express chemical perception
• A way to sample over chemical perception automatically

[#6X4:1]-[#8&X2&H1:2]
We begin with a graph of a chemical environment with decorated nodes connected by edges (bonds).
We make Monte Carlo moves in this chemical environment space.
We test this by checking whether we can learn chemical perception of existing force fields on DrugBank? Yes

Here, vdW types in smirnoff99Frosst
We test this by checking whether we can learn chemical perception of existing force fields. Can we learn the chemical perception used by existing force fields on DrugBank? Yes.
We hope these ideas can help advance force field science
We hope these ideas can help advance force field science.

We want to automate parameter sampling given input data.
We hope these ideas can help advance force field science.

We want to automate parameter sampling given input data.

This requires learning chemical perception.
We hope these ideas can help advance force field science.

We want to automate parameter sampling given input data. This requires learning chemical perception.

SMIRKS provide a language for this.
We hope these ideas can help advance force field science.

We want to automate parameter sampling given input data. This requires learning chemical perception.

SMIRKS provide a language for this. They also easily fix some challenges of atom typing.
Acknowledgments

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