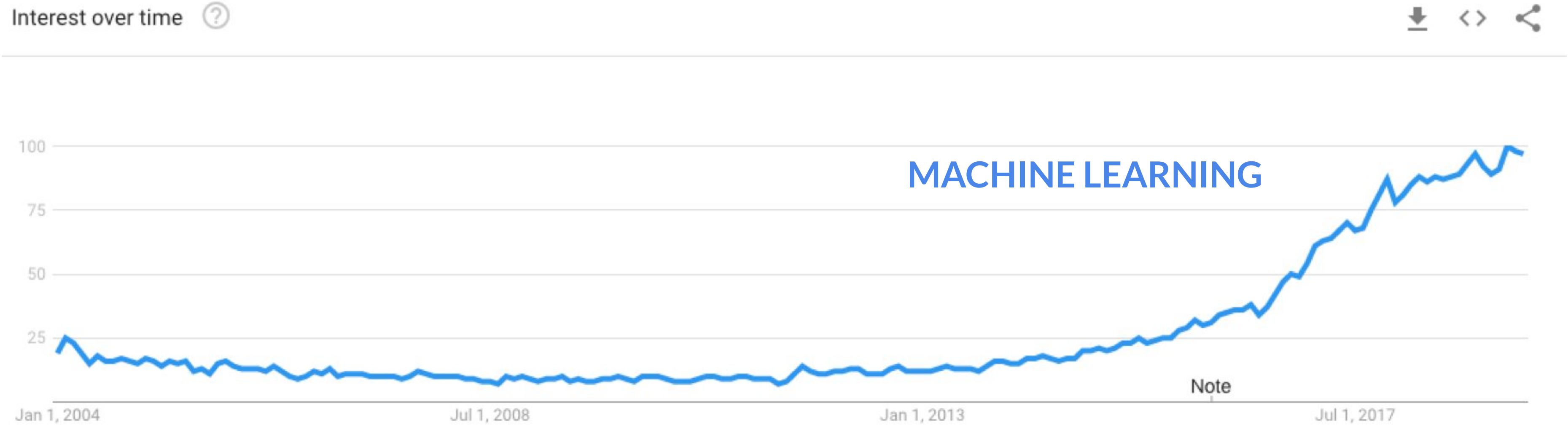
The future of the Open Force Field Initiative: Year Two and Beyond

Aug 2019 | **#bayesian-inference** on Slack

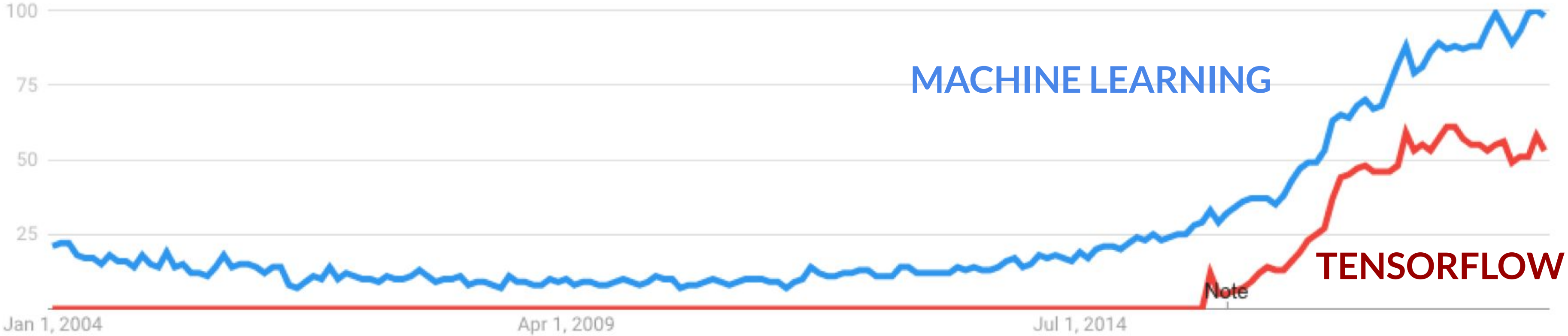
WHAT COULD THE FUTURE LOOK LIKE FOR FORCE FIELD SCIENCE?

WE CAN LOOK TO THE MACHINE LEARNING FIELD FOR INSPIRATION



MACHINE LEARNING

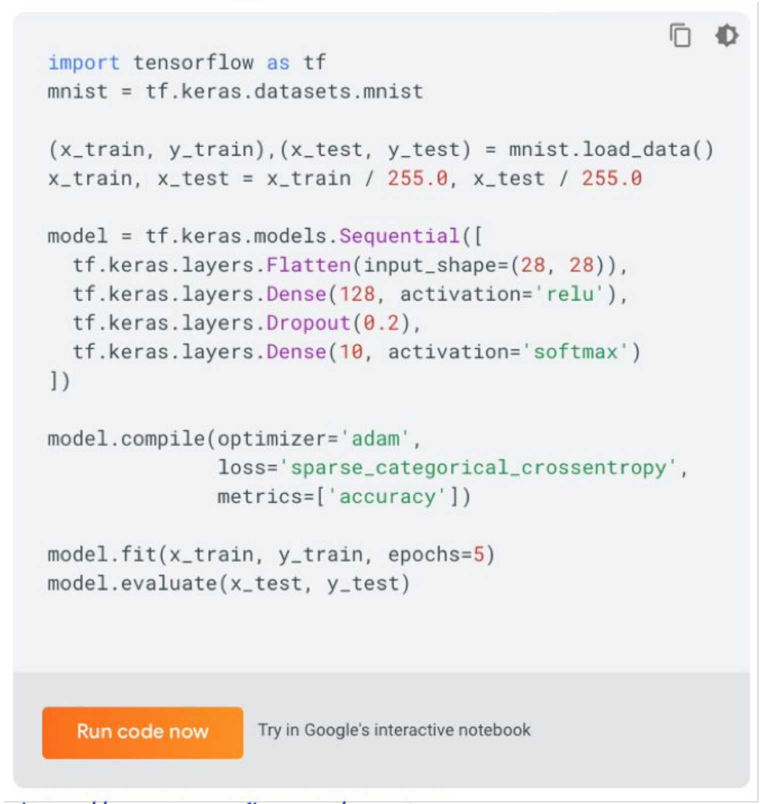
WE CAN LOOK TO THE MACHINE LEARNING FIELD FOR INSPIRATION



- Open software ecosystems have the potential to **accelerate progress**
- Providing useful levels of abstraction **enhances productivity**
- Easy-to-use tools **find new uses everywhere**

CAN WE MAKE BUILDING NEW FORCE FIELDS AS EASY AS TRAINING A MACHINE LEARNING MODEL?

training a neural network



```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

https://www.tensorflow.org/overview

import your tools

grab a standard, curated dataset

define a novel model architecture

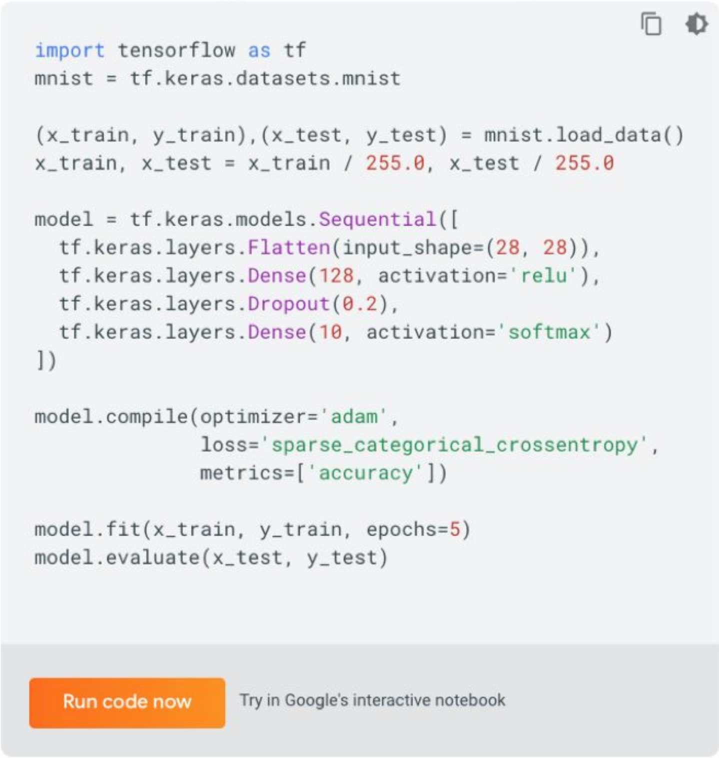
declare your objectives in training it

fit it

use it

CAN WE MAKE BUILDING NEW FORCE FIELDS AS EASY AS TRAINING A MACHINE LEARNING MODEL?

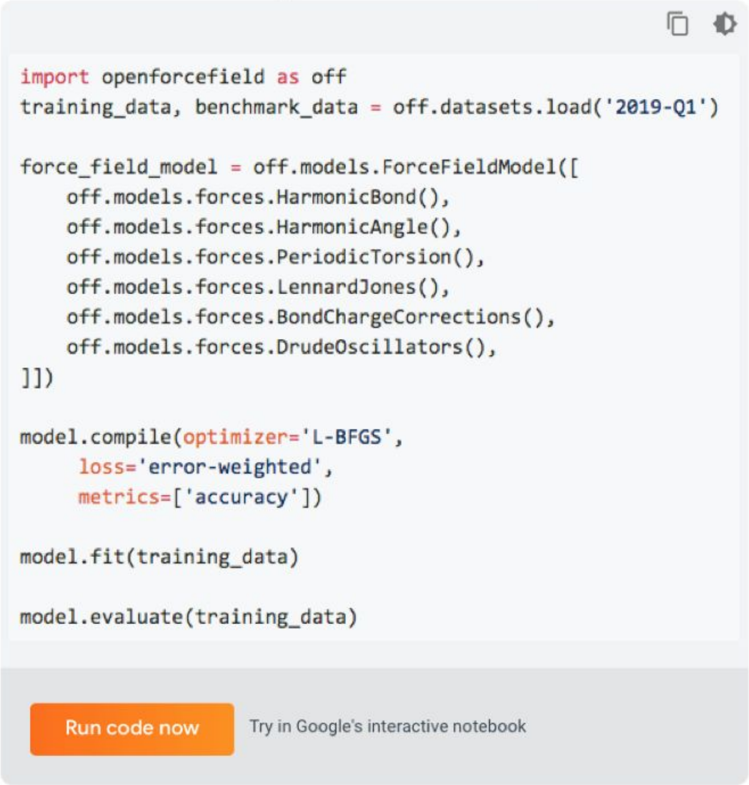
training a neural network



```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

https://www.tensorflow.org/overview

fitting a force field



```
import openforcefield as off
training_data, benchmark_data = off.datasets.load('2019-Q1')

force_field_model = off.models.ForceFieldModel([
    off.models.forces.HarmonicBond(),
    off.models.forces.HarmonicAngle(),
    off.models.forces.PeriodicTorsion(),
    off.models.forces.LennardJones(),
    off.models.forces.BondChargeCorrections(),
    off.models.forces.DrudeOscillators(),
]])

model.compile(optimizer='L-BFGS',
     loss='error-weighted',
     metrics=['accuracy'])

model.fit(training_data)

model.evaluate(training_data)
```

Would this be useful to the community?

AUTOMATION IS KEY TO SCALABILITY

EASY

Gen 1

- ⚙ Bonds/angle refitting to high-level QM
- ⚙ Refit torsions to high-level QM for drug-like molecules
- ⚙ Small molecule Lennard-Jones improvements based on liquid property data

Gen 2

- Valence type expansion
- Lennard-Jones type expansion
- Inclusion of host-guest thermodynamics in fitting
- Refit BCCs to high-quality QM and liquid-phase data
- Use partial bond orders in fitting process to simplify valence type complexity
- Introduce off-site charges and BCCs to support them

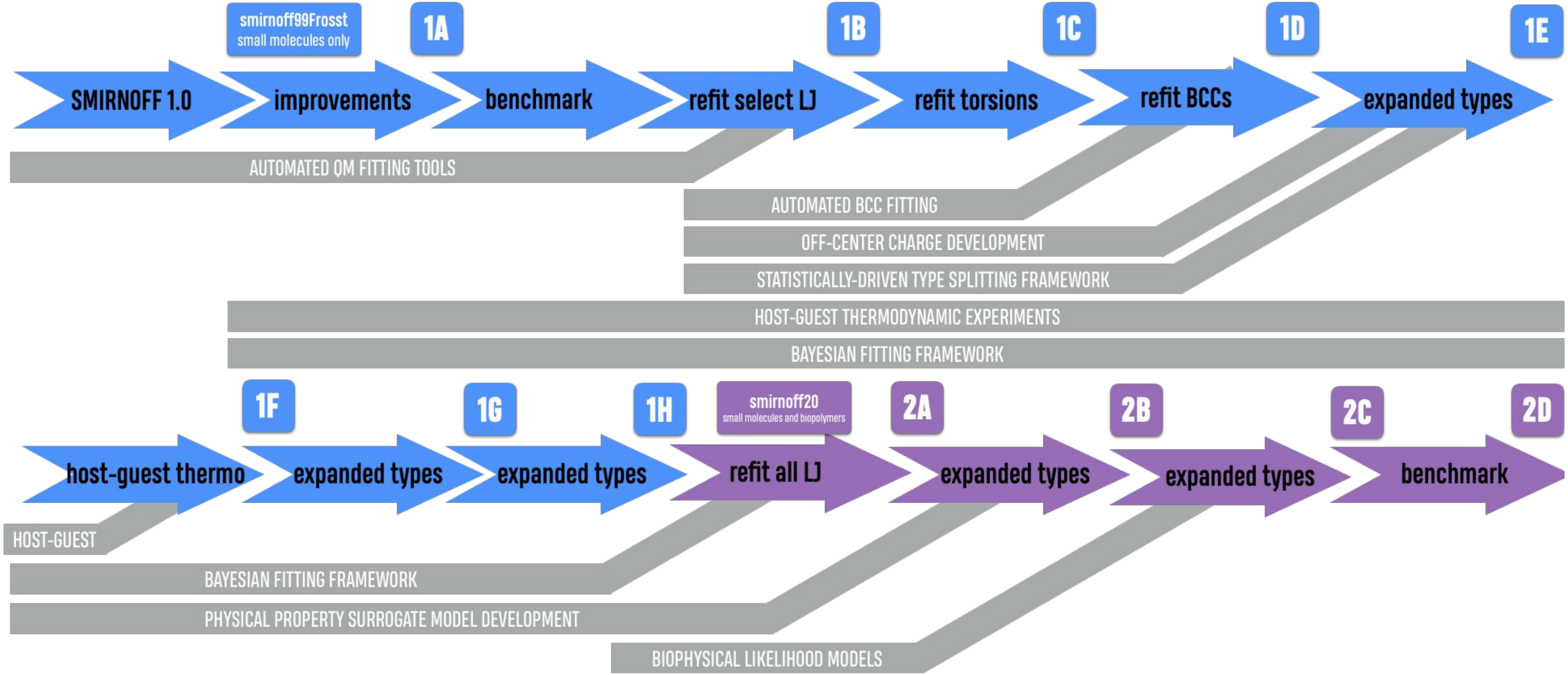
Gen 3

- Complete Lennard-Jones refit (requires breaking AMBER compatibility)
- Bayesian parameter uncertainty propagation to quantify systematic error
- Surrogate thermodynamic models to accelerate forcefield parameterization
- Automated type refinement to penalize complexity
- Selective polarizability

HARD

⚙ Automated and ready to scale to more data to achieve increased accuracy/coverage

AUTOMATION ALLOWS US TO FOCUS ON SCIENCE AND RAPIDLY BENEFIT FROM MATURE RESEARCH EFFORTS



smirnoff99Frosst
small molecules only
1A
1B
1C
1D
1E
SMIRNOFF 1.0
improvements
benchmark
refit select LJ
refit torsions
refit BCCs
expanded types
AUTOMATED QM FITTING TOOLS
AUTOMATED BCC FITTING
OFF-CENTER CHARGE DEVELOPMENT
STATISTICALLY-DRIVEN TYPE SPLITTING FRAMEWORK
HOST-GUEST THERMODYNAMIC EXPERIMENTS
BAYESIAN FITTING FRAMEWORK
1F
1G
1H
smirnoff20
small molecules and biopolymers
2A
2B
2C
2D
host-guest thermo
expanded types
expanded types
refit all LJ
expanded types
expanded types
benchmark
HOST-GUEST
BAYESIAN FITTING FRAMEWORK
PHYSICAL PROPERTY SURROGATE MODEL DEVELOPMENT
BIOPHYSICAL LIKELIHOOD MODELS

WE CAN EXPLORE MANY PHYSICAL AND QUANTUM CHEMICAL PROPERTIES

EXPERIMENTAL DATA

- 🟦 densities of neat liquids and misicble 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

AUTOMATION ALLOWS US TO KEEP INCREASING DATASET SIZES TO ACHIEVE INCREASINGLY GREATER COVERAGE AND ACCURACY

Generation of large quantum chemical datasets

- PDB Ligand Expo (80K)
- Enamine REAL (11B)
- Partner patent datasets (>1M)
- Partner-submitted datasets: (???)

  ```
  qcfractal-submit --api-key <key> molecules.smi
  ```

Exploiting large physical property datasets

- ThermoML Archive has a **huge** amount of data we can exploit

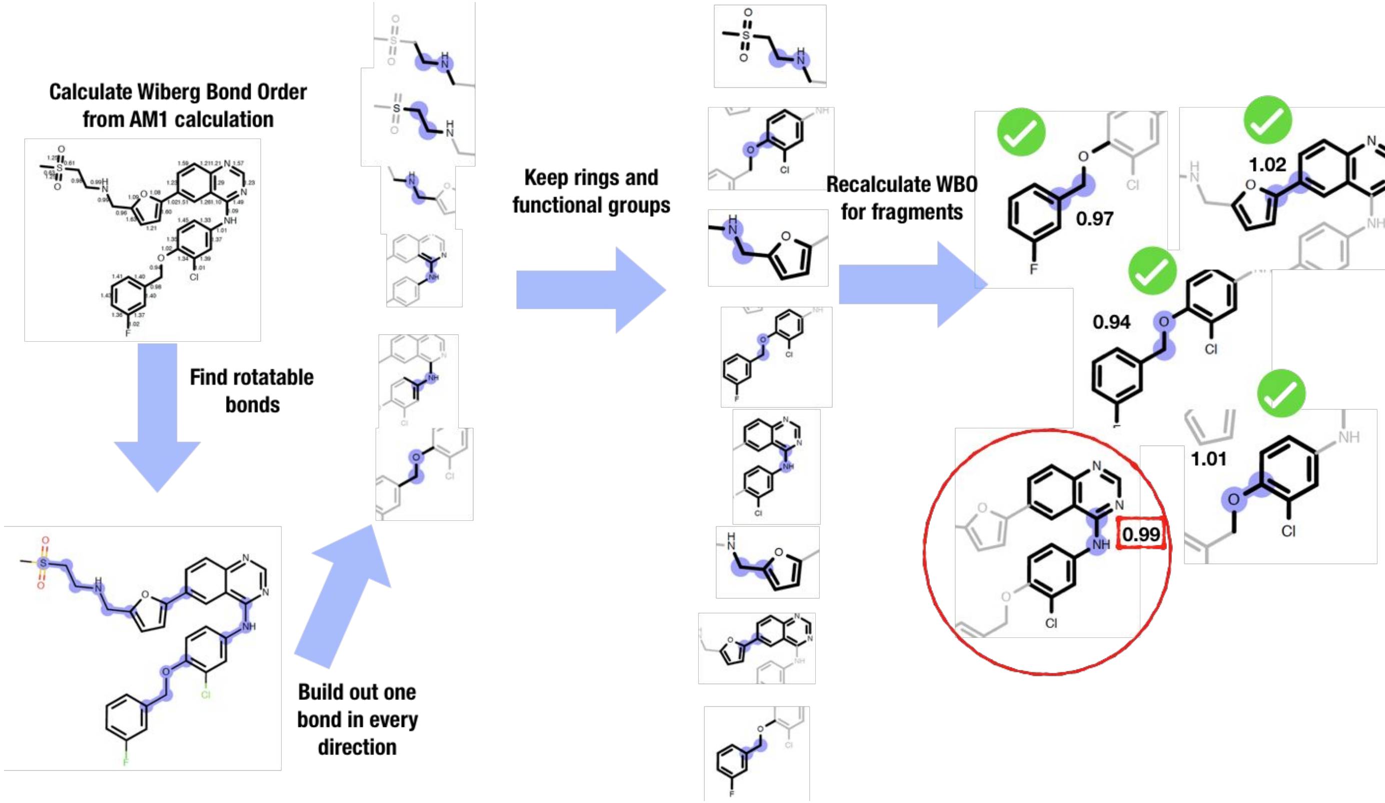
Increasing efficiency

- psi4 and the QCFractal ecosystem
- OpenMM and PropertyEstimator

Increased access to computational resources

SMALL MOLECULE FRAGMENTATION IS REACHING MATURITY

Will soon be generating very large quantum chemical datasets



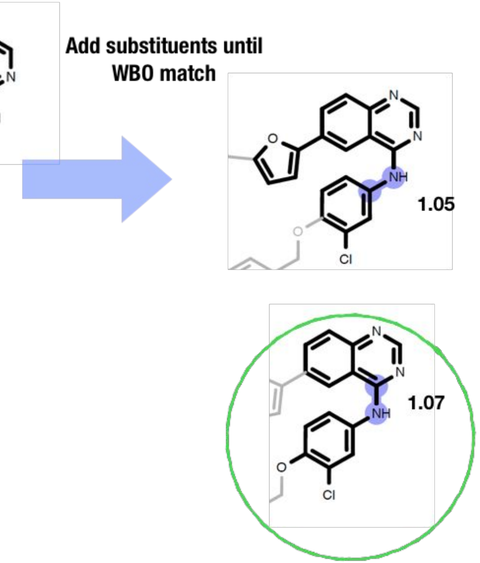
Calculate Wiberg Bond Order from AM1 calculation

Find rotatable bonds

Build out one bond in every direction

Keep rings and functional groups

Recalculate WBO for fragments



Add substituents until WBO match

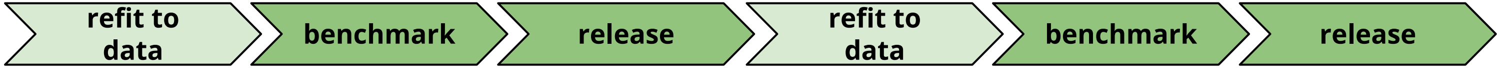
Inspired by Xinjun Hou and Visnu Sresth (Pfizer)

CHAYA STERN

AUTOMATION AND MODULARITY IS KEY TO RAPID ITERATIVE PROGRESS

Deliverables:

- Regular versioned improvements of small molecule force fields:



refit to data → benchmark → release → refit to data → benchmark → release

  - Integrating additional experimental data each cycle
    - More complex mixture data
    - Host-guest binding thermodynamics
    - Benchmark sets become training set for next generation force field
  - Aim to transition to Bayesian inference
    - Sample parameter space broadly, escaping local minima
    - Refine/expand atom types with reversible-jump
    - Identify BCCs for next-generation charge models
    - Select optimal mixing rules, off-site charges, etc.

Research questions:

- Which new functional forms are justified by the data?

MODULARITY MAKES IT EASY TO EXPLORE DIFFERENT DATA SOURCES AND PHYSICAL MODELS IN PARALLEL

Quantum chemical datasets

- **QCFractal** can define new distributed quantum chemical workflows

Physical properties

- **PropertyDataset**/**PhysicalProperty** can integrate new physical properties and data sources
  (speed of sound, X-ray scattering, BindingDB, CCSD)

Assessment sets

- **PropertyEstimator** can integrate distributed free energy workflows
  (pAPRika, gmx/gromacs, YANK, perses)

Open force field toolkit:

- **ParameterType** plugin can integrate new physical models
  (virtual sites, point polarizable dipoles, Drude oscillators, ML models)

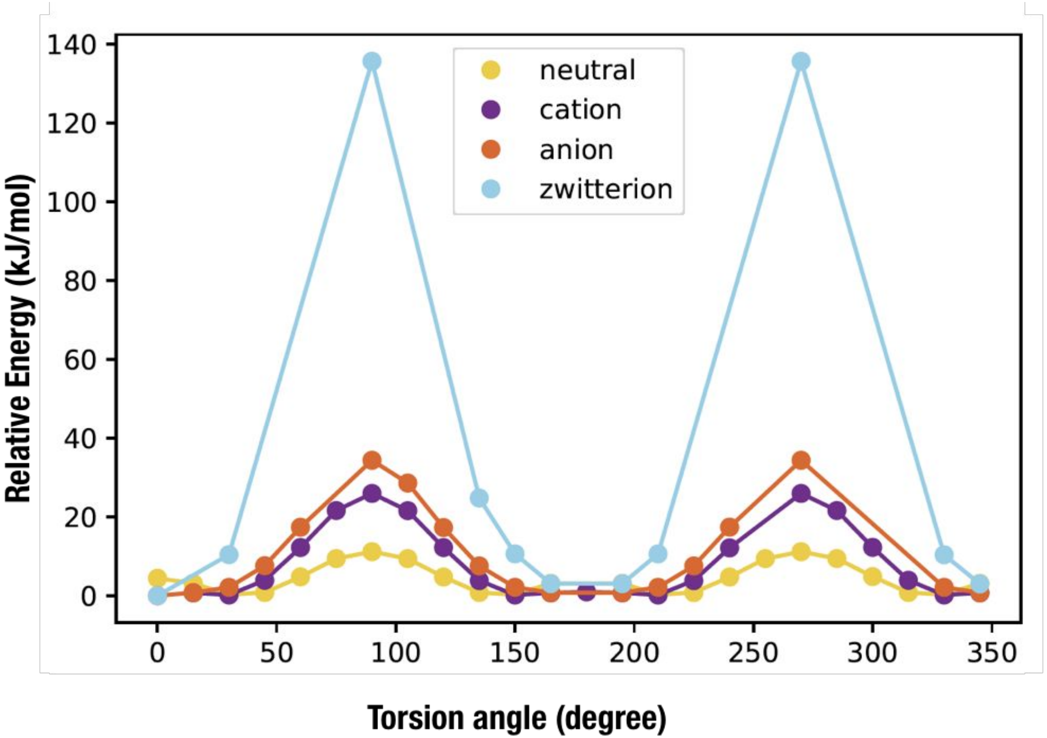
SMIRNOFF EXTENSIONS WILL ENABLE MORE SCIENTIFIC RESEARCH

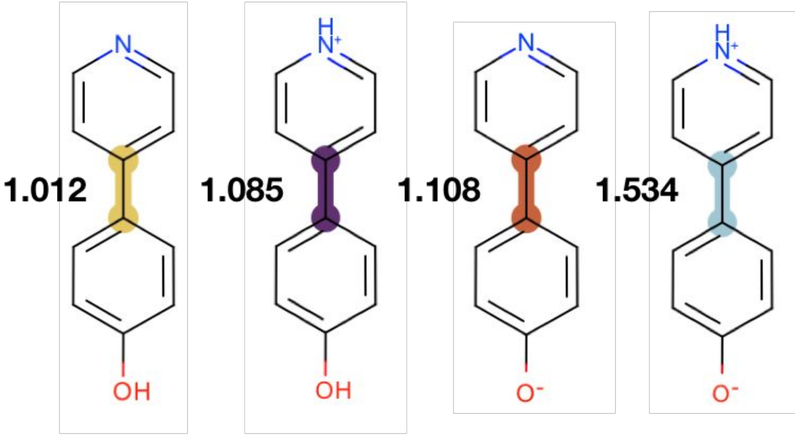
- Wiberg bond order (WBO) interpolated valence terms
- Virtual sites
- Alternative Lennard-Jones mixing rules / pairwise LJ parameters
- Coupled torsions (CMAP)
- Experimental support for alternative vdW functional forms
- Experimental support for point polarizable dipoles / Drude oscillators
- Experimental support for ML potentials

WIBERG BOND ORDER (WBO) INTERPOLATED VALENCE TERMS COULD GREATLY REDUCE NUMBER OF DISTINCT TYPES NEEDED

$$W_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} |D_{\mu\nu}|^2.$$

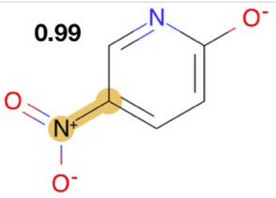
$$D_{\mu\nu} = 2 \sum_{i}^{\text{occ.}} C_{\mu i} C_{\nu i}^*,$$

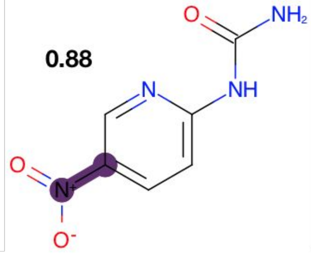


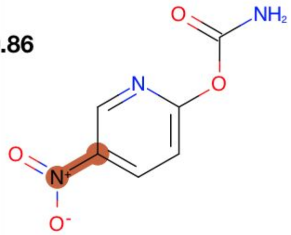


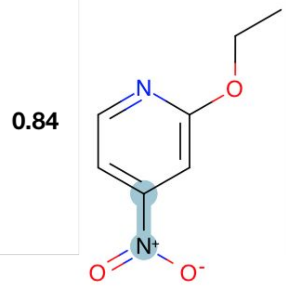
CHAYA STERN

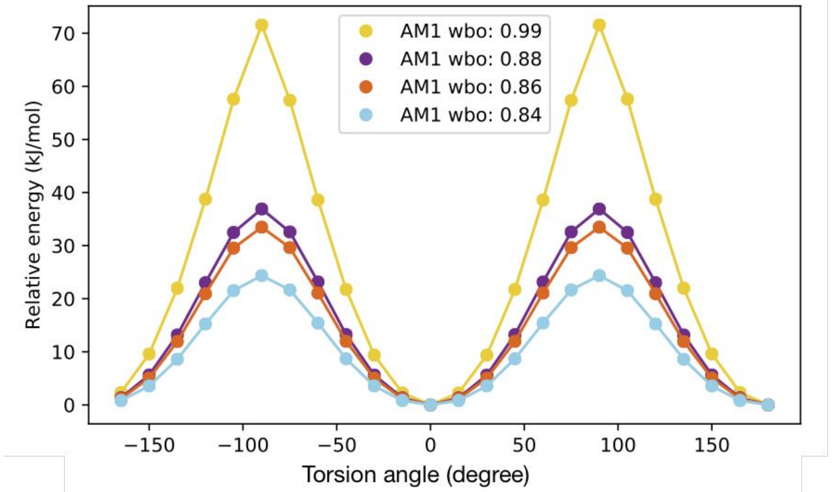
WIBERG BOND ORDER (WBO) INTERPOLATED VALENCE TERMS COULD GREATLY REDUCE NUMBER OF DISTINCT TYPES NEEDED

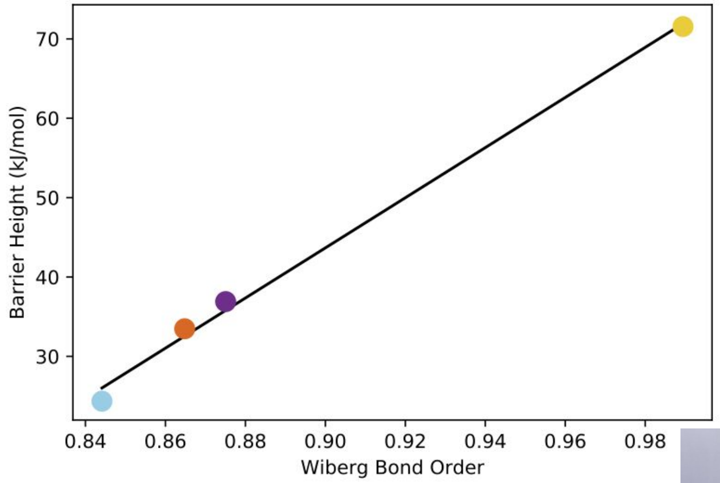






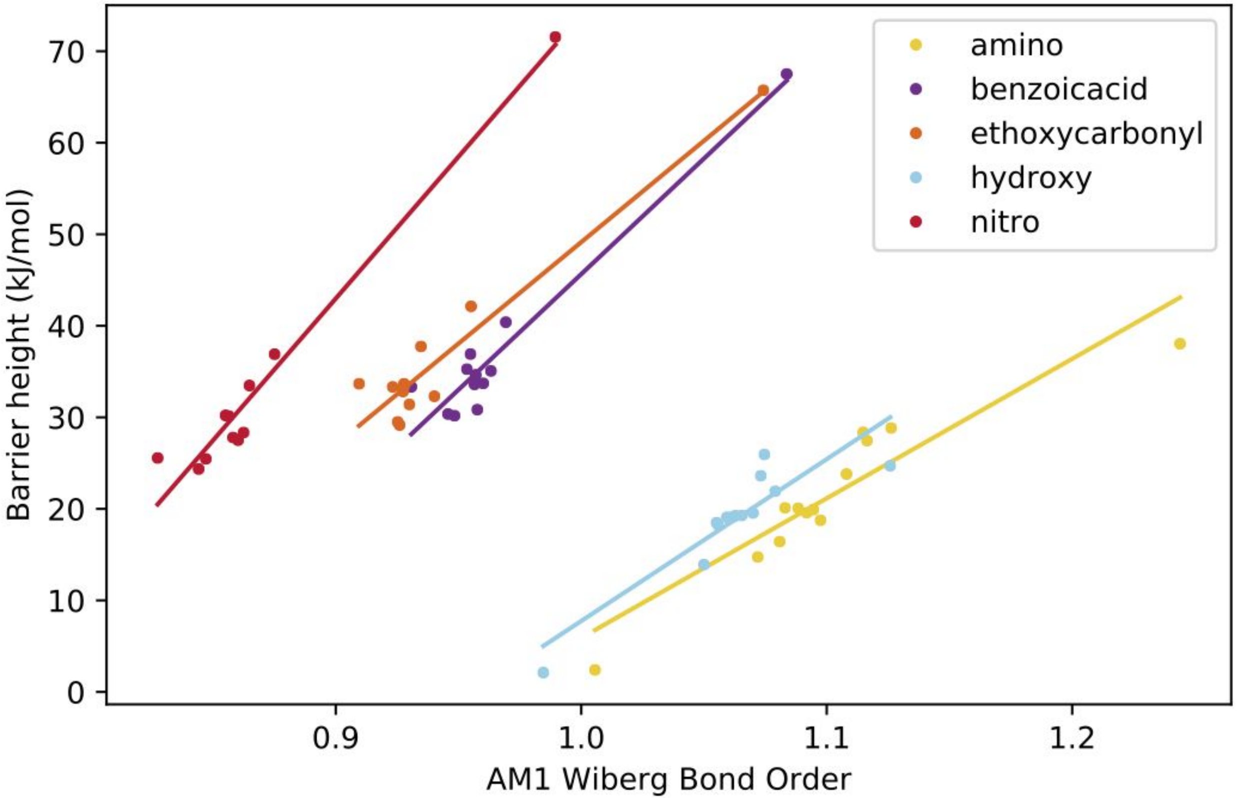






CHAYA STERN

WIBERG BOND ORDER (WBO) INTERPOLATED VALENCE TERMS COULD GREATLY REDUCE NUMBER OF DISTINCT TYPES NEEDED



Slope is nearly universal!

CHAYA STERN

DISTRIBUTED COMPUTING NEEDS AND RESOURCES WILL CONTINUE TO GROW

Academic clusters:

- MSKCC: ~1000 cores sustained, up to 200 GPUs in bursts
- UCD: ~1000 cores, ~100 GPUs; leading refits to quantum chemical data
- Other sites will continue to be integrated

XSEDE (due 15 Oct 2019)**:**

- Frontera opening soon (CPUs+GPUs)
  (Olexandr ran 100K DLPNO-CCSD(T)/CBS in one day!)

Cloud computing

- Would need to reserve budget for AWS/GCE costs

Folding@home?

- Investigating running QCFractal on **Folding@home** (requires dev $)

Industry resources?

- Contribute to QCFractal quantum chemical calculations?

HOW DO WE AUTOMATE AWAY HUMAN-DRIVEN TASKS?
Force field optimization has evolved over time

year	**forcefield**	**parameter fitting scheme**
1990s	1990s	lots of hand tweaking
early 2000s	early 2000s	genetic algorithm
mid 2000s	mid 2000s	least squares
2019	OFF 1st gen	regularized least squares (ForceBalance)
2020	**OFF 2nd gen**	**Bayesian inference**

WHAT DO WE WANT FROM AUTOMATED PARAMETERIZATION?

Everything is **automatic**; no hand-tweaking necessary

We aren’t just seeking **local brittle optima**, but instead good, generalizable parameter sets

We don’t need to arbitrarily **assign data weights** to different data sources

Feats of chemical insight are not required; decisions guided by statistically sound methodology

Toolkit **automatically selects appropriate functional forms** given the data

We can **rapidly and systematically improve forcefields** with more data

Forcefield provides an **assessment of the reliability of its predictions**

The forcefield can **tell us what new data is most valuable** for improving accuracy

A Bayesian inference approach fulfills our desiderata

Parameter space may have many local minima
Bayesian methods can find good, broad, low free energy basins that are likely to generalize

Parameter optimization problems generally feature nonlinear nearly-degenerate solutions
Bayesian methods can cope well with nonlinear regions of high probability

We have good ways to characterize statistical error, but no way to assess systematic error
Bayesan methods can predict chemistry-specific uncertainties (e.g., sulfonyl, sulfonamide groups)

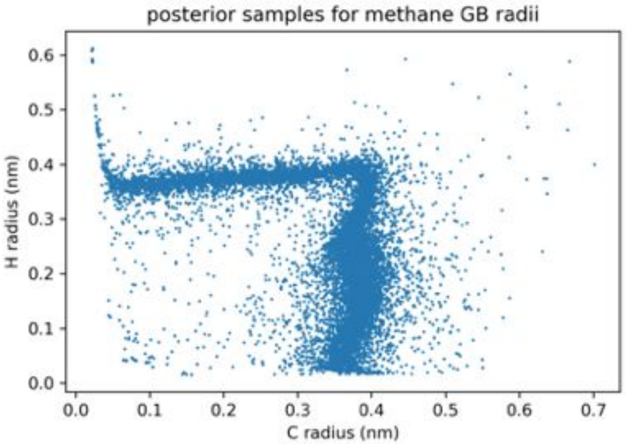
We want to make data-driven decisions about which choices are best justified by data
The data will tell us which functional forms or mixing rules are sufficiently well-justified

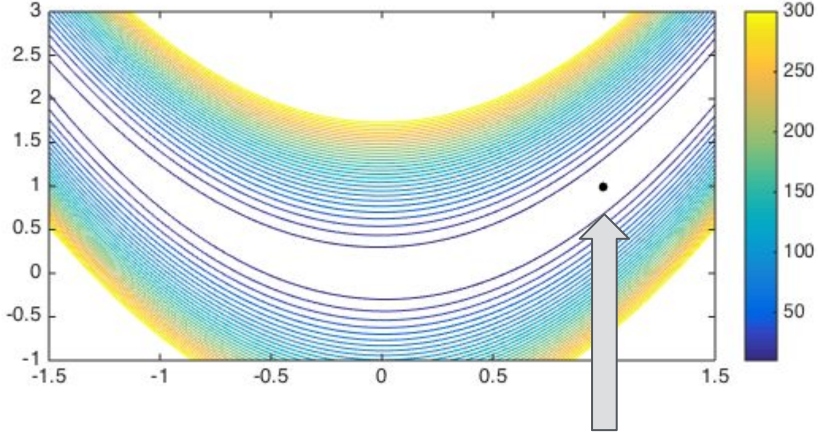
Data is automatically weighted by its measurement error
No need for humans to specify weights for different measurements or regularization schemes

We can balance the proliferation of parameters with their gain in accuracy
Polarizability, multipoles more parameters, risk overfitting; Bayesian methods penalize complexity

We want to identify which experiments will give us the most new information at least cost
The whole field of Bayesian experimental design can be harnessed

Nonlinear optimization problems generally feature nearly-degenerate solution spaces



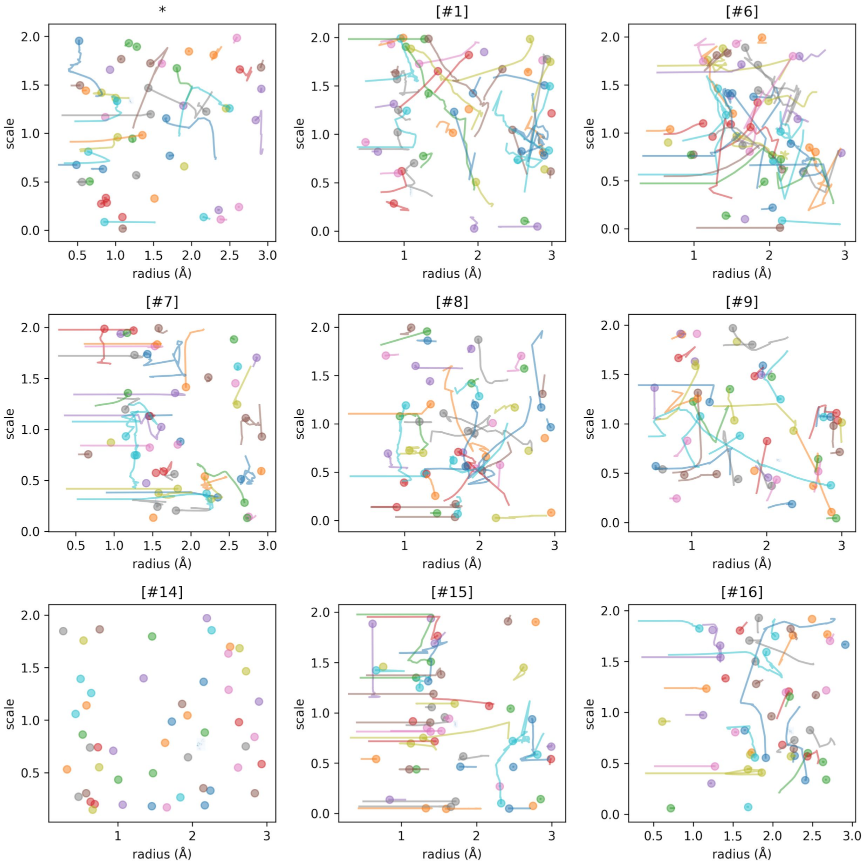


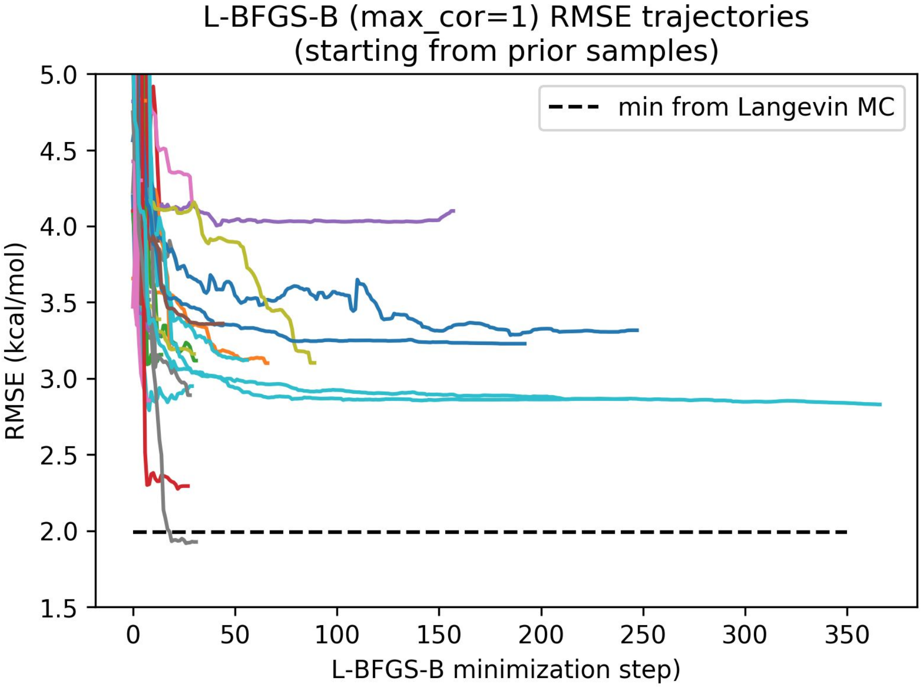
why just this set of parameters?

Some predicted properties may be **insensitive** to different choices of parameters in this set, but other predicted properties by be very **sensitive**

PARAMETER LANDSCAPES ARE MULTI-MODAL

Optimizing GBSA parameters to FreeSolv gets stuck in local minima

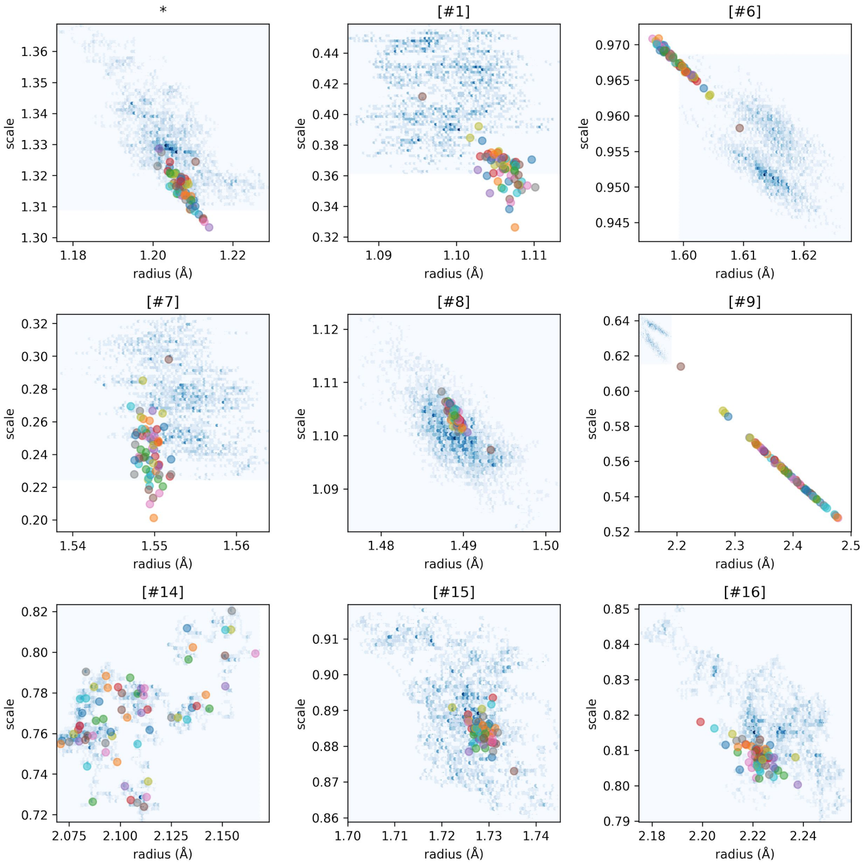


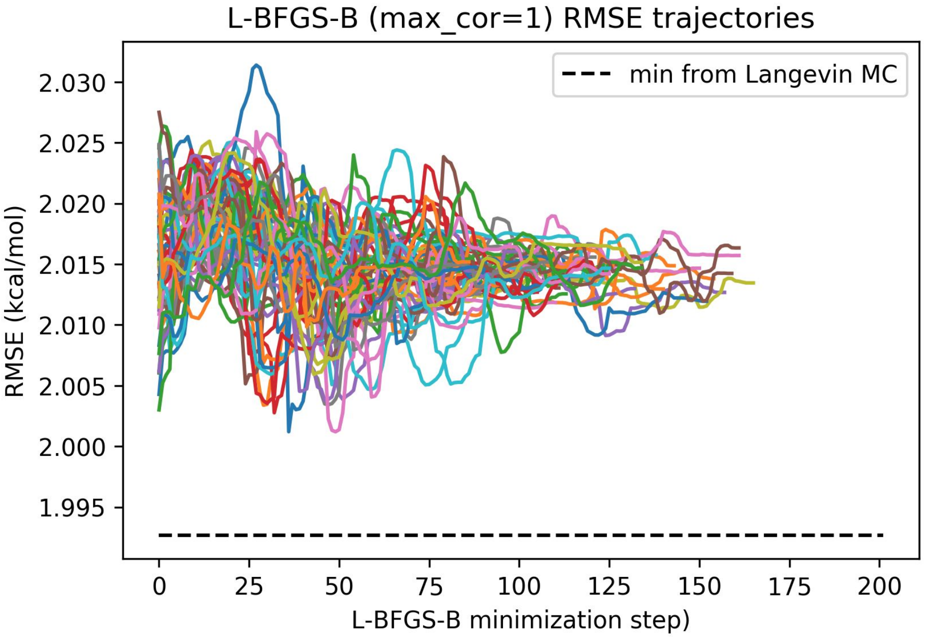


JOSH FASS

PARAMETER LANDSCAPES ARE MULTI-MODAL

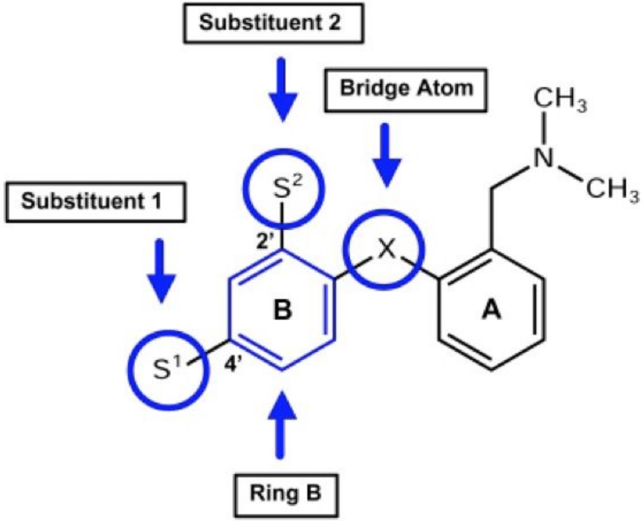
Even near the global basin, there are tons of minima!

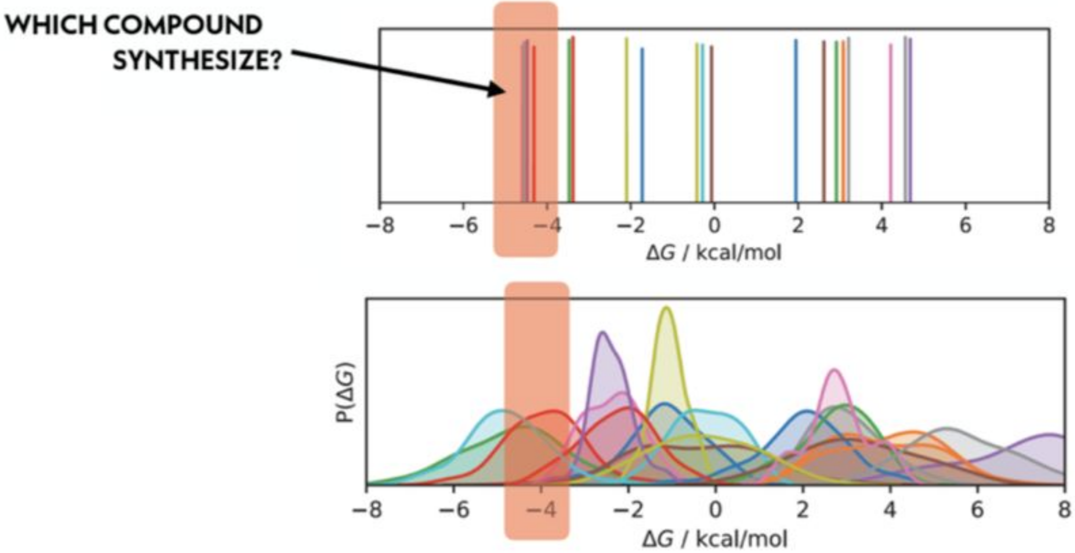




JOSH FASS

Predictive uncertainties are essential for good decisions





Existing approaches for quantifying **statistical error** are mature, but **systematic error** dominates error.
We need force fields that **know when they will provide poor estimates of predicted properties** because training data is too limited or parameters may be overfit.

The Bayesian Way

Bayes rule allows us to assign **how confident we are** in a specific force field parameter set, and provides an automated, statistically motivated way to **update** parameters given new data

$$\underbrace{p(\theta|\mathcal{D})}_{\text{posterior}} \propto \underbrace{p(\mathcal{D}|\theta)}_{\text{likelihood}}\underbrace{p(\theta)}_{\text{prior}}$$

$\mathcal{D}$	data
θ	forcefield parameters
$p(\theta\|\mathcal{D})$	posterior
$p(\mathcal{D}\|\theta)$	data model
$p(\theta)$	prior on forcefield parameters

Data likelihood can be factorized into contributions from independent measurements

$$p(\theta|\mathcal{D}) \propto \underbrace{p(\mathcal{D}|\theta)}_{\text{likelihood}}p(\theta)$$

$\mathcal{D}$ data

θ forcefield parameters

$$p(\mathcal{D}|\theta) = \prod_{n=1}^{N} p_n(\mathcal{D}_n|\theta)$$

Computing the likelihood requires two components

Likelihood function requires a **forward model** and an **experimental error model**:
* **Forward model** gives the true error-free property A(θ) given parameters θ
* **Experimental error model** is probability data A' was observed given error-free A(θ)

Both can also be used in regularized least-squares fitting!

$$\underbrace{p(\theta|\mathcal{D})}_{\textbf{posterior}} \propto \underbrace{p(\mathcal{D}|\theta)}_{\textbf{likelihood}}\underbrace{p(\theta)}_{\textbf{prior}}$$

$\mathcal{D}$ data

θ forcefield parameters

Forward models come from best practices in computing experimental observables from molecular simulations, and often involve expectations or free energy differences:

e.g. density calculation

$$\rho_*(\theta) \equiv \left\langle \frac{N}{V} \right\rangle_\theta = E_\theta\left[\frac{N}{V}\right]$$

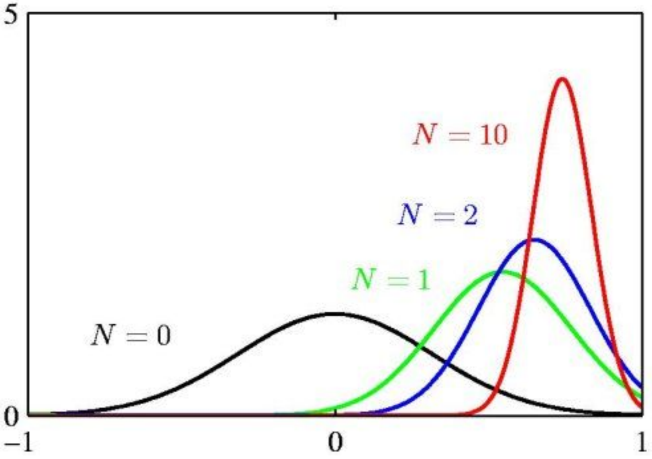
Experimental error models come from models of the experimental measurement process and may involve unknown parameters corresponding to instrumental reliability:

e.g. density measurement

$$p(\rho_{\text{obs}}|\rho_*) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(\rho-\rho_*)^2}{\sigma^2}}$$

Conditioning on more data reduces uncertainty

As we collect more data, the posterior uncertainty in parameters given data decreases



We can quantify uncertainty in an **information-theoretic** manner $S[p(x)] = -\int dx\, p(x) \ln p(x)$

Interactive illustration: http://rpsychologist.com/d3/bayes/

Priors express our current state of knowledge

Informationless priors attempt to minimally bias parameters

$$\underbrace{p(\theta|\mathcal{D})}_{\textbf{posterior}} \propto \underbrace{p(\mathcal{D}|\theta)}_{\textbf{likelihood}}\underbrace{p(\theta)}_{\textbf{prior}}$$

$\mathcal{D}$ data
θ forcefield parameters

Often we strive to make posterior probabilities independent of parameterization;
e.g., a prior for equilibrium angle θ and cos(θ) should give the same posterior distribution

Nuisance parameters can be introduced to model unknown experimental details like instrument noise, but can be marginalized out (but still contribute to overall uncertainty)

Prior rounds of inference can be used as priors:

* Posterior parameter sets can be rapidly reweighted using likelihood functions for new data
* Posterior parameter sets will be close to equilibrium for seeding new posterior sampling

Statistical mechanics and Bayesian inference are isomorphic

If you know stat mech already, you know Bayesian inference!

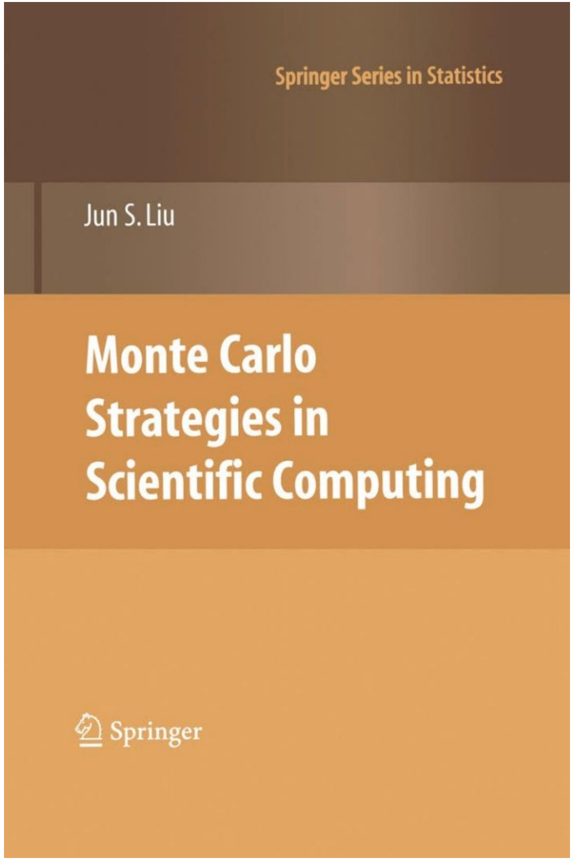
statistical mechanics		statistical inference
potential energy	$u(x) = -\ln q(x)$	potential (negative log unnormalized density)
partition functions	$Z_i = \int dx\, q_i(x)$	normalizing constants
binding affinities/ partition coefficients	Z_i / Z_j	bayes factors/ model evidences
physical properties	$E_i[A] = Z_i^{-1} \int dx\, q_i(x)\, A(x)$	expectations
entropy	$S = -\int dx\, \pi(x) \ln \pi(x)$	entropy/uncertainty/ information

The **same algorithms** can be used in both fields

This book unifies methods from both fields if you want to learn more

statistical mechanics

parallel tempering
simulated tempering
wang-landau
nonequilibrium candidate monte Carlo
semigrand-canonical monte carlo
(metropolized) molecular dynamics
configurational bias monte carlo
pruned/enriched rosenbluth methods


Springer Series in Statistics
Jun S. Liu
Monte Carlo Strategies in Scientific Computing
Springer

statistical inference

sequential Monte carlo
self-adjusted mixture sampling
annealed importance sampling
reversible-jump monte carlo
hybrid monte carlo
particle filtering

https://www.springer.com/gp/book/9780387763699

Familiar algorithms can be used to sample from Bayesian posteriors and potential energy functions

Metropolis Monte Carlo can make updates to individual dimensions or combinations of dimensions using only the potential

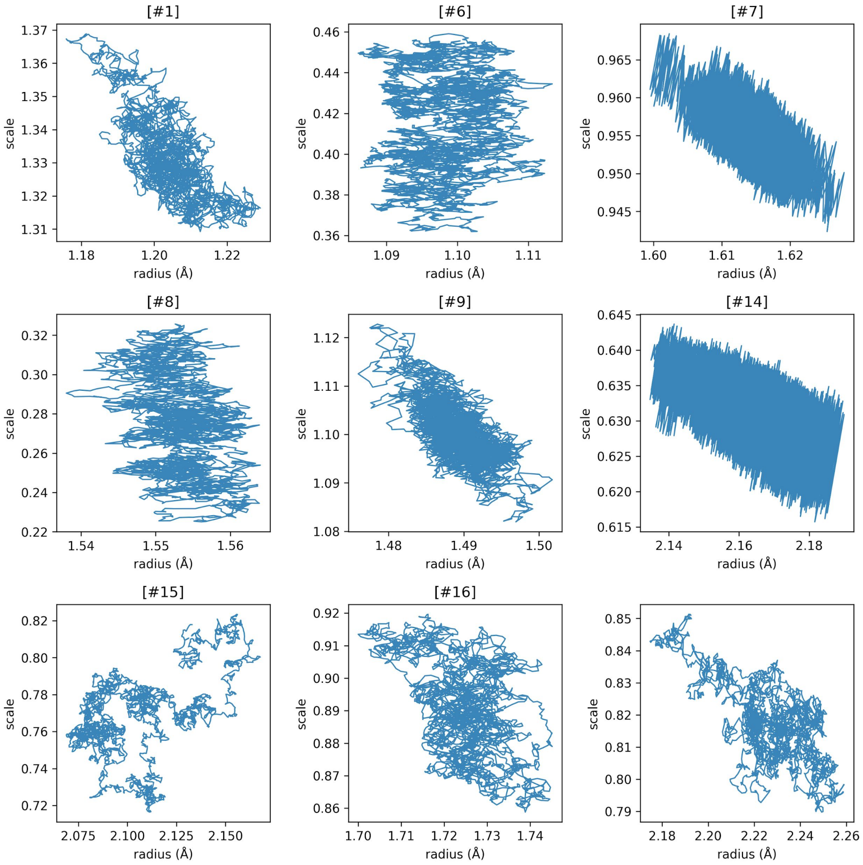
Hybrid Monte Carlo can exactly sample from the posterior using gradient information, but becomes inefficient in high dimension

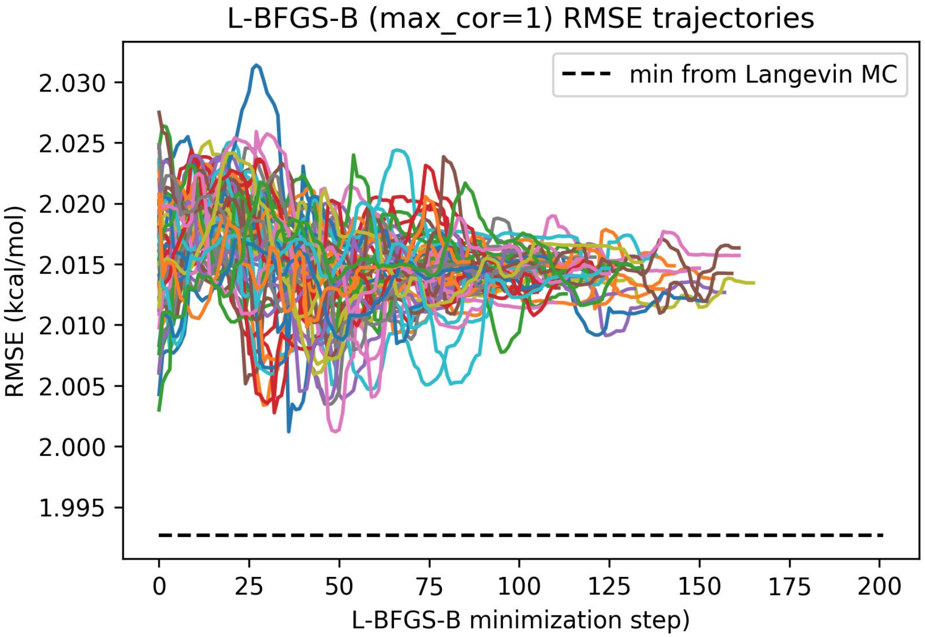
Langevin integrators can approximately sample from very highly multidimensional problems using gradient information, and good Langevin integrators (BAOAB) are accurate and efficient

Gibbs sampling strategies (like replica exchange or expanded ensemble) allow alternation between updating different subsets of parameters, or even discrete and continuous parameters

PARAMETER LANDSCAPES ARE MULTI-MODAL

Langevin methods effectively sample parameter space

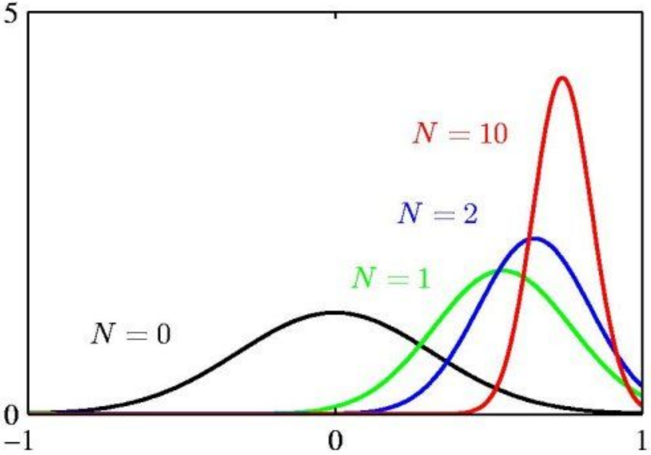



L-BFGS-B (max_cor=1) RMSE trajectories

JOSH FASS

Conditioning on more data reduces uncertainty

As we include more data (*N*), parameter uncertainty decreases

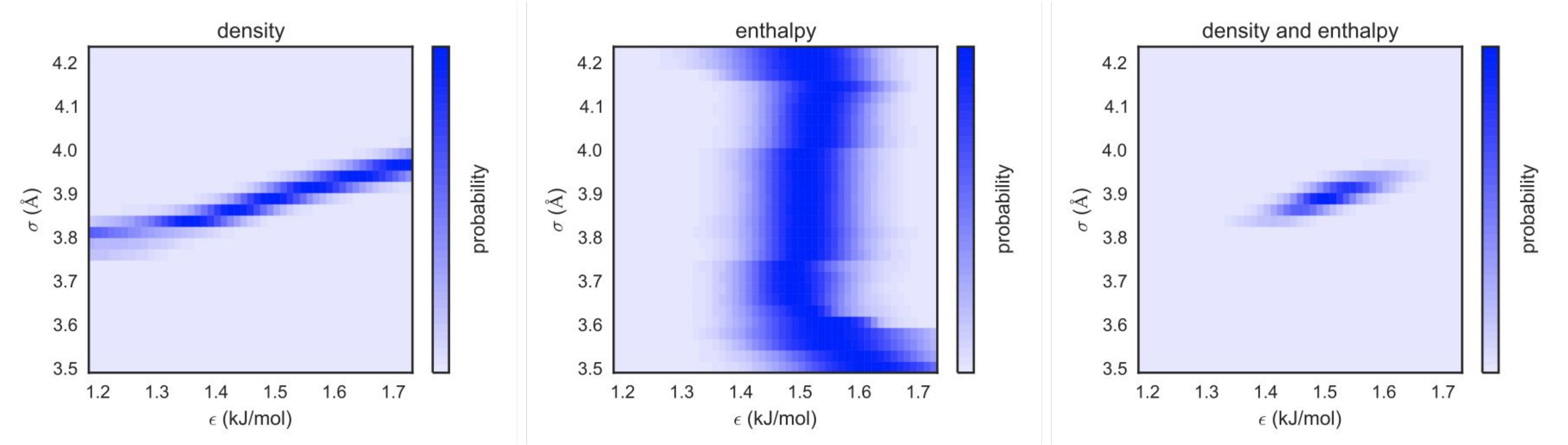


We can quantify uncertainty in an **information-theoretic** manner $S[p(x)] = -\int dx\, p(x) \ln p(x)$

Interactive illustration: http://rpsychologist.com/d3/bayes/

Conditioning on more data reduces uncertainty

A real example: **United-atom methane** (from Michael Shirts and Levi Naden)
Combining density and enthalpy greatly reduces region over which posterior is large



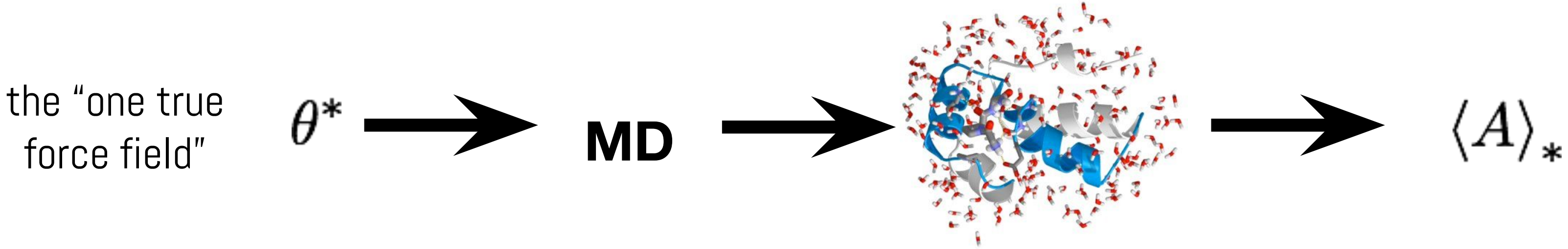
Bayesian models provide a direct way to estimate systematic error in predictions

The **marginal posterior probability** for an expectation describes our confidence in a prediction:

$$p(A'|\mathcal{D}) = \int d\theta \, \delta(A' - \langle A \rangle_\theta) \, p(\theta|\mathcal{D})$$

We can also predict the **joint uncertainty** in two computed properties,
which can **exploit favorable cancellation of error** in predictions

Predicting properties: The old way



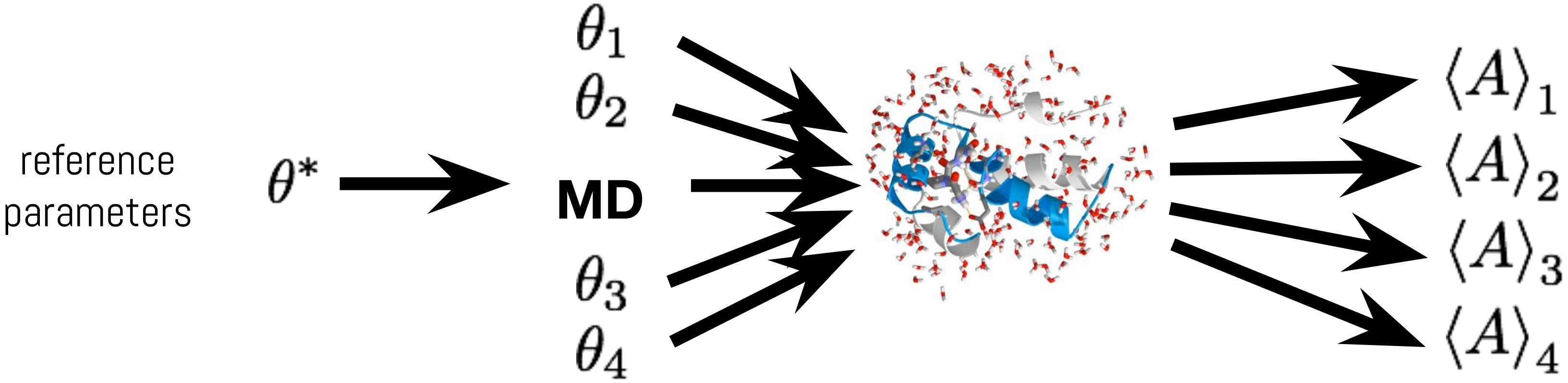
the "one true force field"

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

One set of parameters in, one computed result out

Only **statistical** error can be assessed

Predicting properties: The Bayesian way



reference parameters

θ^*

MD

θ_1

θ_2

θ_3

θ_4

$\langle A \rangle_1$

$\langle A \rangle_2$

$\langle A \rangle_3$

$\langle A \rangle_4$

Multiple parameter sets in, multiple estimates out

We can estimate both statistical and systematic components of computed results
Simulations are performed with a reference parameter set, and **fast reweighting** assesses systematic error

Constructing a force field requires addressing many questions where data should drive decisions

Lennard-Jones mixing rules: Which Lennard-Jones mixing rules (Lorentz-Berthelot, geometric, arithmetic, others) best fits liquid-phase data?

Functional forms: Which nonbonded sterics model best fits the data?

Atom types: How many atom types do I need to fit the data well? Which ones?

Bond charge corrections (BCCs): How many BCCs (and of what types) do I need to reproduce experimental properties well?

Off-atom charges: Do off-atom partial charges provide a sufficient increase in accuracy to warrant the additional parameters? Where do they belong?

Polarizable sites: Is polarizability worth the increase in parameters? Which atoms or sites should have polarizability? How many distinct polarizability parameters are needed?

Bayesian model selection lets data drive decisions

If discrete model choices are available, **Bayes factors** provide ratio of evidence for one model over another in a manner that is directly interpretable as break-even gambling odds.

model evidence $$\mathcal{E}(\mathcal{M}_i|\mathcal{D}) \quad = \quad p(\mathcal{D}|\mathcal{M}_i) = \int d\theta \, p(\mathcal{D}|\theta, \mathcal{M}_i) \, p(\theta|\mathcal{M}_i) \, p(\mathcal{M}_i)$$

Computation is **isomorphic** with an absolute or relative free energy calculation

Bayesian model selection lets data drive decisions

We can include multiple discrete model choices in the same posterior sampling scheme with **reversible-jump Monte Carlo (RJMC)**, *even if the models differ in dimension!*

Only need to ensure detailed-balance is satisfied in proposed jumps between models:

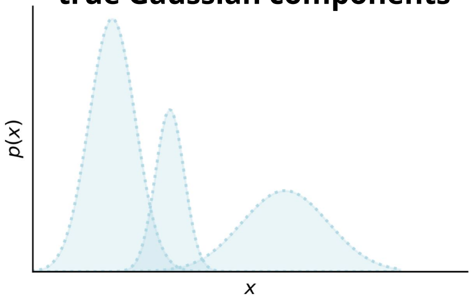
$$P_{\text{accept}} \quad = \quad \min\left\{1, \frac{p(\theta_j|\mathcal{D})}{p(\theta_i|\mathcal{D})} \frac{P(\mathcal{M}_j)}{P(\mathcal{M}_i)} \frac{P(\mathcal{M}_i|\mathcal{M}_j)}{P(\mathcal{M}_i|\mathcal{M}_j)} \frac{T_{ji}(\theta_i|\theta_j)}{T_{ij}(\theta_j|\theta_i)}\right\}$$

Computation is isomorphic with **grand-canonical Monte Carlo** simulation

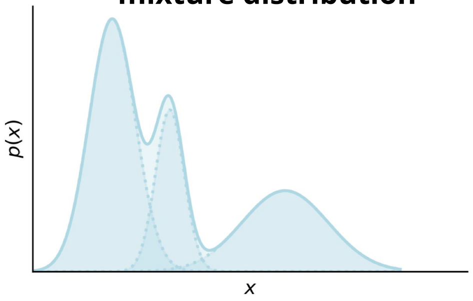
Reversible-jump Monte Carlo (RJMC) is a statistically principled way to sample an unknown number of types

Illustrative example: Fitting **mixture of unknown number of Gaussians** with RJMC

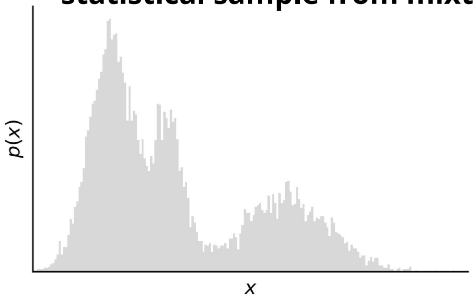
true Gaussian components



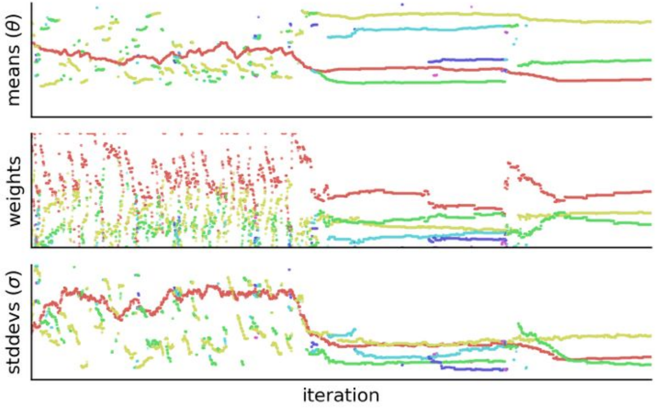
mixture distribution

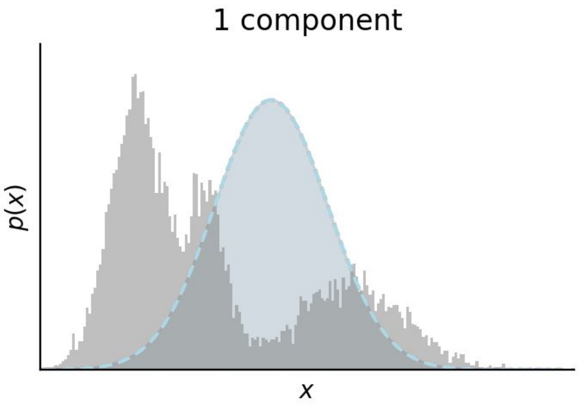


statistical sample from mixture



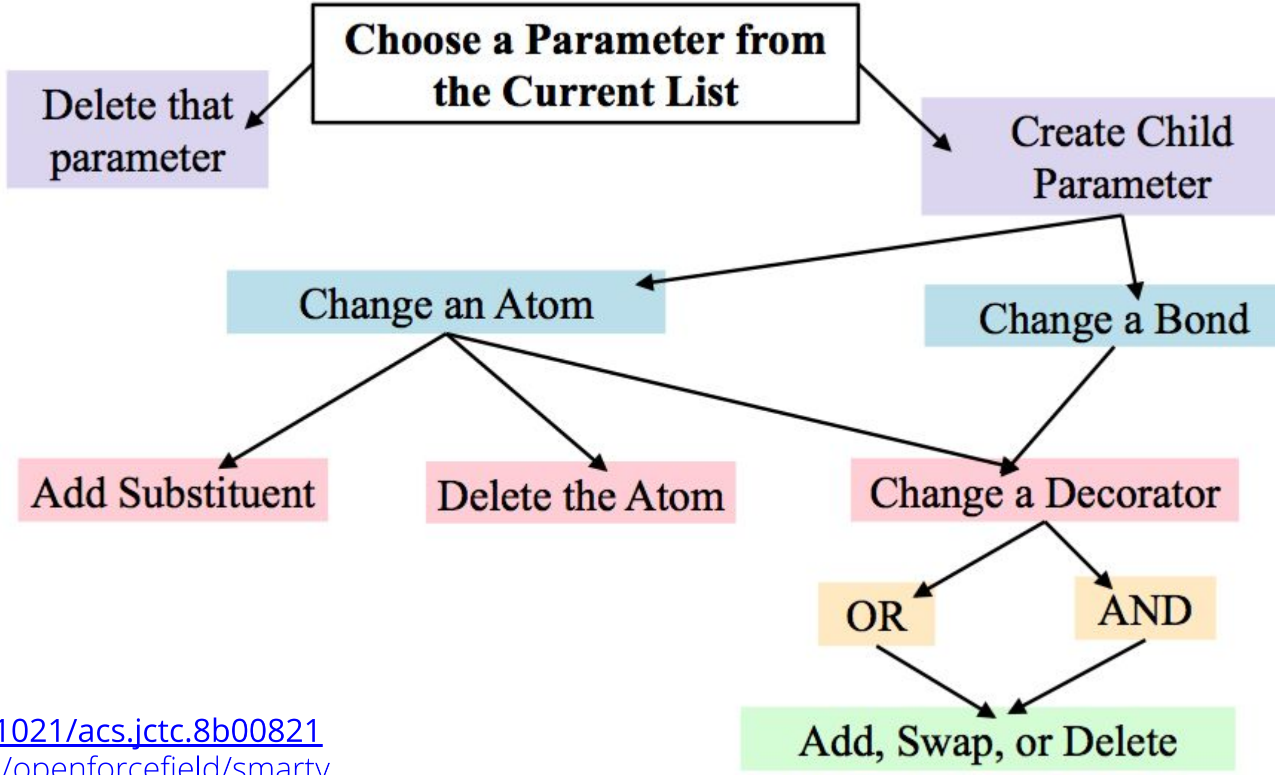
RJMC inference simulation from statistical sample





JOSH FASS

Reversible-jump Monte Carlo (RJMC) could sample over atom types (penalizing complexity) in an automated way



CAMILA ZANETTE

CAITLIN BANNAN

http://doi.org/10.1021/acs.jctc.8b00821
https://github.com/openforcefield/smarty

A simple scheme using SMARTS “decorators” can sample new child types with increased complexity

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

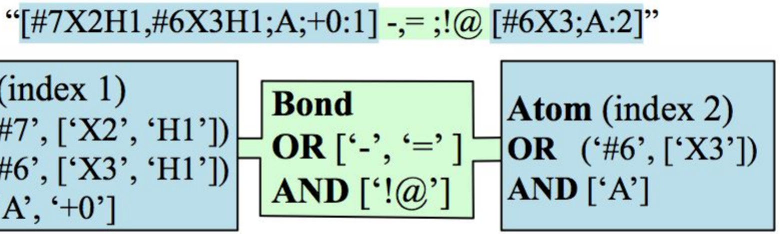
decorators

```
% 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
```

=

proposed child types

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



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

Atom (index 1)
OR (‘#7’, [‘X2’, ‘H1’])
(‘#6’, [‘X3’, ‘H1’])
AND [‘A’, ‘+0’]

Bond
OR [‘-’, ‘=’]
AND [‘!@’]

Atom (index 2)
OR (‘#6’, [‘X3’])
AND [‘A’]

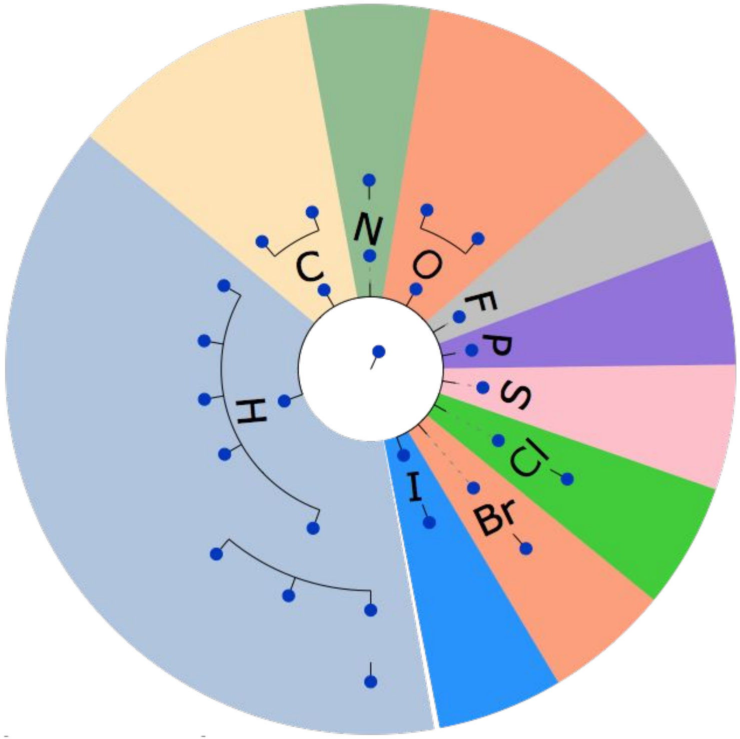
CAMILA ZANETTE

CAITLIN BANNAN

http://doi.org/10.1021/acs.jctc.8b00821

https://github.com/openforcefield/smarty

Sampling with this scheme can generate SMARTS-based typing trees with interesting complexity

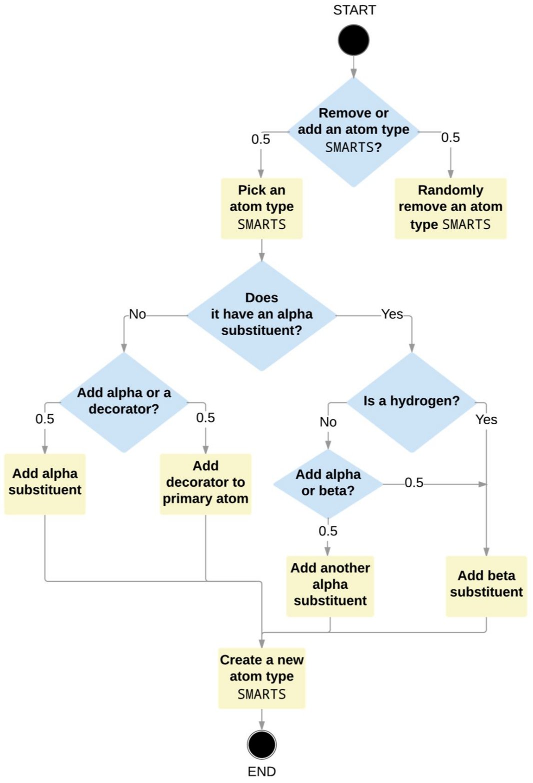


```
    H ([#1:1])
        5262 ([#1:1]~[#6])
            4148 ([#1:1]~[#6!X3])
                8668 ([#1:1]~[#6!X3]~[$ewg2])
                    1874
([#1:1]~[#6!X3](~[$ewg2])~[$ewg2])
                        4596
([#1:1]~[#6!X3](~[#7])(~[$ewg2])~[$ewg2])
                2356 ([#1:1]~[#6!X3X2])
   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

CAMILA ZANETTE

CAITLIN BANNAN

A simple RJMC scheme can recover human-generated atom types over large typed molecule datasets



START

Remove or add an atom type SMARTS?

0.5

0.5

Pick an atom type SMARTS

Randomly remove an atom type SMARTS

Does it have an alpha substituent?

No

Yes

Add alpha or a decorator?

0.5

0.5

Is a hydrogen?

No

Yes

Add alpha substituent

Add decorator to primary atom

Add alpha or beta?

0.5

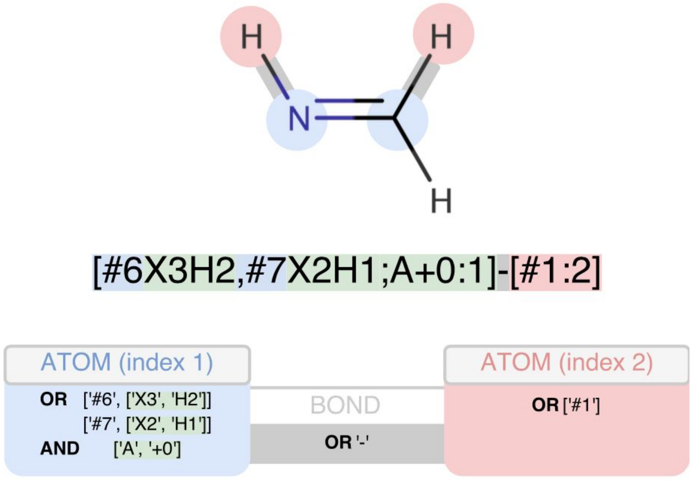
0.5

Add another alpha substituent

Add beta substituent

Create a new atom type SMARTS

END



[#6X3H2,#7X2H1;A+0:1]-[#1:2]

ATOM (index 1)

OR ['#6', ['X3', 'H2']]
['#7', ['X2', 'H1']]
AND ['A', '+0']

BOND

OR '-'

ATOM (index 2)

OR ['#1']

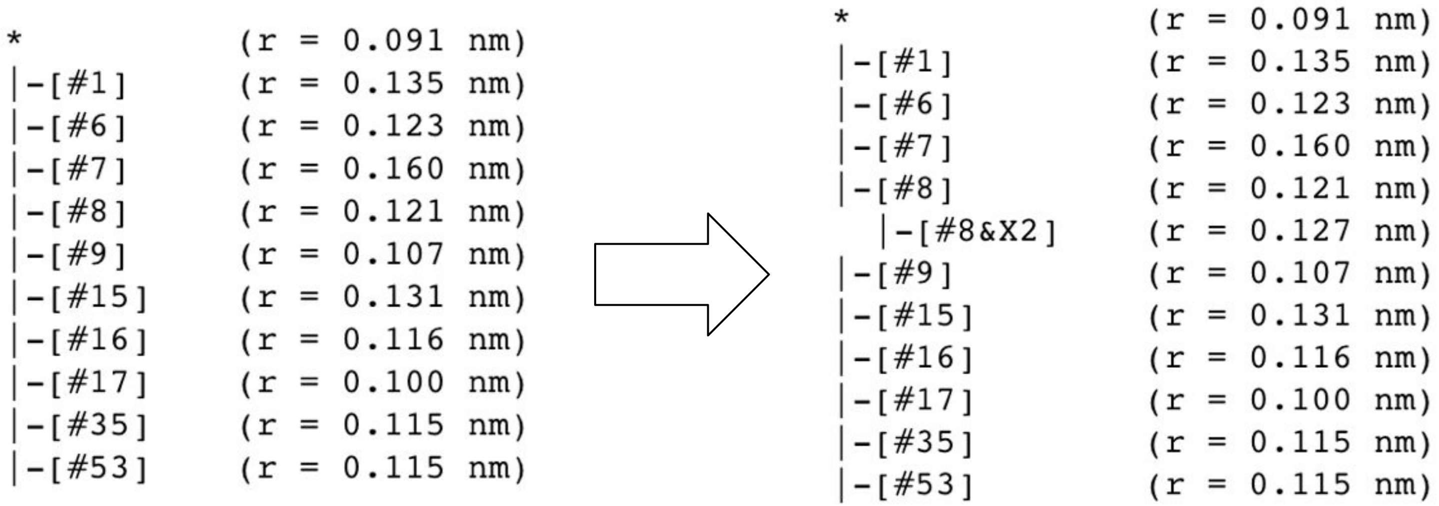
	AlkEthOH (%)		PhEthOH (%)		MiniDrugBank (%)	
	Initial	Maximum	Initial	Maximum	Initial	Maximum
All	67.8	100.0	54.5	100.0	40.5	93.0
Hydrogen	52.6	100	39.0	100	35.9	97.0
Carbon	100	100	71.0	100	39.0	95.7
Oxygen	63.6	100	84.1	100	38.3	98.0
Nitrogen	n/a	n/a	n/a	n/a	33.1	84.0
Sulfur	n/a	n/a	n/a	n/a	52.2	100

http://doi.org/10.1021/acs.jctc.8b00821

https://github.com/openforcefield/smarty

Initial experiment: Sampling over GBSA atom types fit to small molecule hydration free energies

Example of a GBSA type creation proposal



```
*               (r = 0.091 nm)
|-[#1]          (r = 0.135 nm)
|-[#6]          (r = 0.123 nm)
|-[#7]          (r = 0.160 nm)
|-[#8]          (r = 0.121 nm)
|-[#9]          (r = 0.107 nm)
|-[#15]         (r = 0.131 nm)
|-[#16]         (r = 0.116 nm)
|-[#17]         (r = 0.100 nm)
|-[#35]         (r = 0.115 nm)
|-[#53]         (r = 0.115 nm)
```

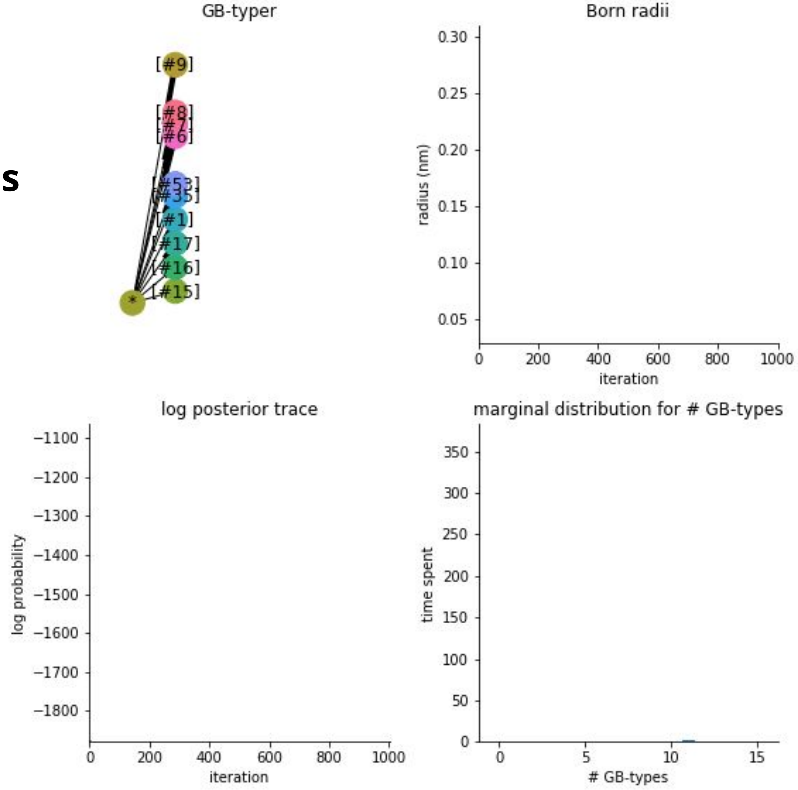
```
*               (r = 0.091 nm)
|-[#1]          (r = 0.135 nm)
|-[#6]          (r = 0.123 nm)
|-[#7]          (r = 0.160 nm)
|-[#8]          (r = 0.121 nm)
  |-[#8&X2]     (r = 0.127 nm)
|-[#9]          (r = 0.107 nm)
|-[#15]         (r = 0.131 nm)
|-[#16]         (r = 0.116 nm)
|-[#17]         (r = 0.100 nm)
|-[#35]         (r = 0.115 nm)
|-[#53]         (r = 0.115 nm)
```

JOSH FASS

Initial experiment: Sampling over GBSA atom types fit to small molecule hydration free energies

Hierarchical SMIRNOFF types

log posterior

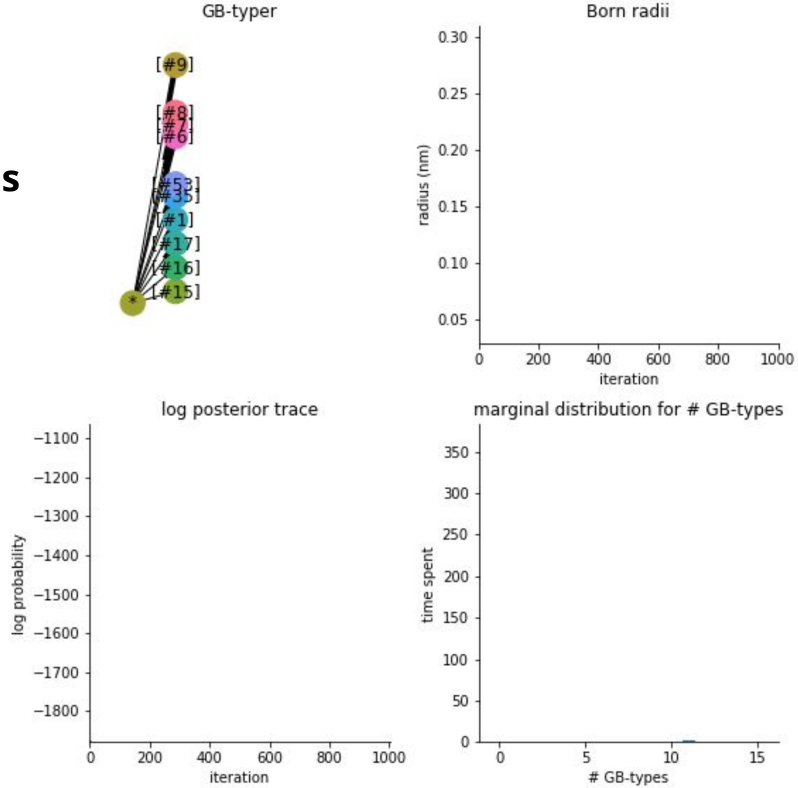


JOSH FASS

The RJMC approach can discover interesting chemistry!

Hierarchical SMIRNOFF types

log posterior



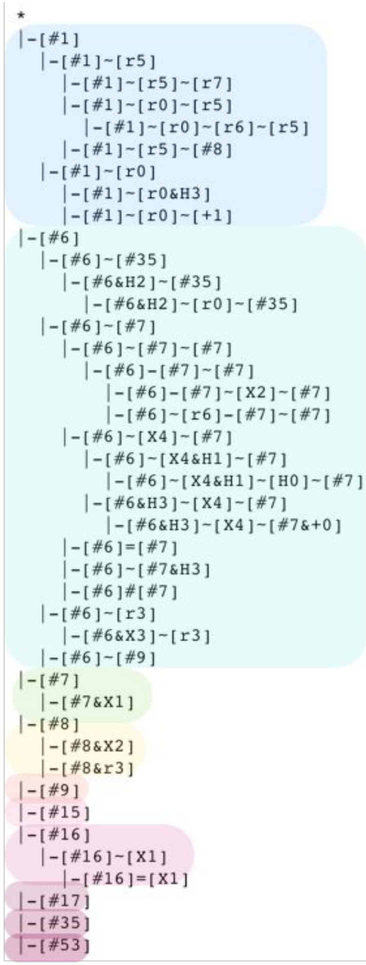
JOSH FASS

9 hydrogen types

20 carbon types

distinguishes nitriles!

and sulfonyls!



```
*
|-[#1]
  |-[#1]~[r5]
    |-[#1]~[r5]~[r7]
    |-[#1]~[r0]~[r5]
      |-[#1]~[r0]~[r6]~[r5]
    |-[#1]~[r5]~[#8]
  |-[#1]~[r0]
    |-[#1]~[r0&H3]
    |-[#1]~[r0]~[+1]
|-[#6]
  |-[#6]~[#35]
    |-[#6&H2]~[#35]
      |-[#6&H2]~[r0]~[#35]
  |-[#6]~[#7]
    |-[#6]~[#7]~[#7]
      |-[#6]-[#7]~[#7]
        |-[#6]-[#7]~[X2]~[#7]
        |-[#6]~[r6]-[#7]~[#7]
    |-[#6]~[X4]~[#7]
      |-[#6]~[X4&H1]~[#7]
        |-[#6]~[X4&H1]~[H0]~[#7]
      |-[#6&H3]~[X4]~[#7]
        |-[#6&H3]~[X4]~[#7&+0]
    |-[#6]=[#7]
    |-[#6]~[#7&H3]
    |-[#6]#[#7]
  |-[#6]~[r3]
    |-[#6&X3]~[r3]
  |-[#6]~[#9]
|-[#7]
  |-[#7&X1]
|-[#8]
  |-[#8&X2]
  |-[#8&r3]
|-[#9]
|-[#15]
|-[#16]
  |-[#16]~[X1]
    |-[#16]=[X1]
|-[#17]
|-[#35]
|-[#53]
```

Our second-generation fitting approach will use Bayesian inference, reusing all our existing software components

Automated atom and valence type determination

- **Published**: SMIRKY Monte Carlo moves can rediscover existing types (Zanette, Bannan, Mobley)
- **Current**: Sampling over GBSA typing rules and parameters (Josh Fass)
- **Next**: Automated Lennard-Jones type determination to fit ThermoML Archive data
- **Beyond**: Automated mixing rule and functional form determination

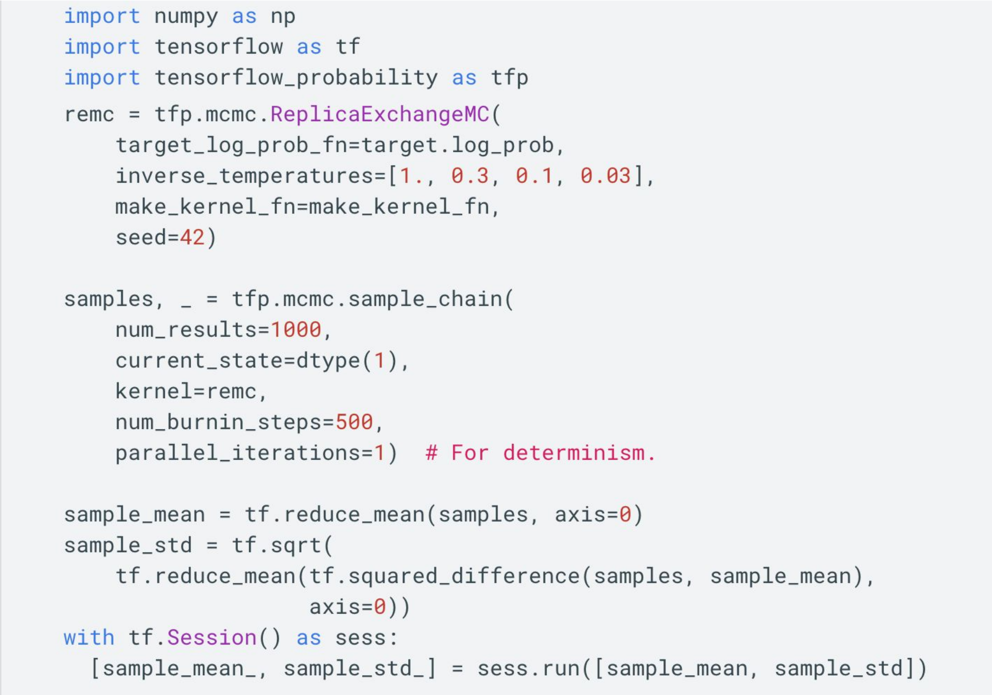
Automated parameter fitting with MCMC avoids local minima

- Currently exploring efficient parallel parameter searching/sampling schemes that utilize gradients and can make use of distributed computing resources

Uncertainty quantification via rapid reweighting

- “killer app” is binding free energy calculations

BAYESIAN FITTING WITH TENSORFLOW PROBABILITY?



```
import numpy as np
import tensorflow as tf
import tensorflow_probability as tfp
remc = tfp.mcmc.ReplicaExchangeMC(
    target_log_prob_fn=target.log_prob,
    inverse_temperatures=[1., 0.3, 0.1, 0.03],
    make_kernel_fn=make_kernel_fn,
    seed=42)

samples, _ = tfp.mcmc.sample_chain(
    num_results=1000,
    current_state=dtype(1),
    kernel=remc,
    num_burnin_steps=500,
    parallel_iterations=1)  # For determinism.

sample_mean = tf.reduce_mean(samples, axis=0)
sample_std = tf.sqrt(
    tf.reduce_mean(tf.squared_difference(samples, sample_mean),
                   axis=0))
with tf.Session() as sess:
  [sample_mean_, sample_std_] = sess.run([sample_mean, sample_std])
```

- Well-supported **ecosystem**
- Support for **distributed computation**
- **Training** advantages for students
- **Synergies** with machine learning potentials: Easier to mix-and-match model components

Would another framework (PyTorch, Python/JAX) be a better choice for our community?

https://www.tensorflow.org/probability

HOW DO WE WANT TO SUPPORT MACHINE LEARNING POTENTIALS?

ANI

Deep Tensor Networks

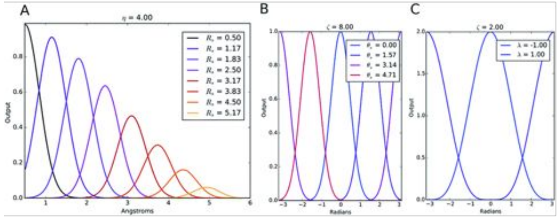
Tensor Field Networks

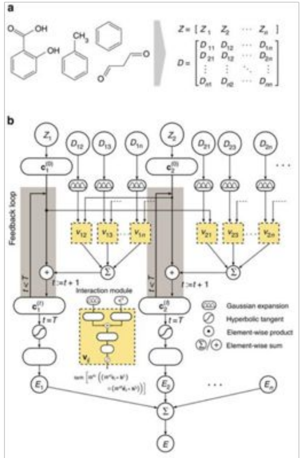
PotentialNet

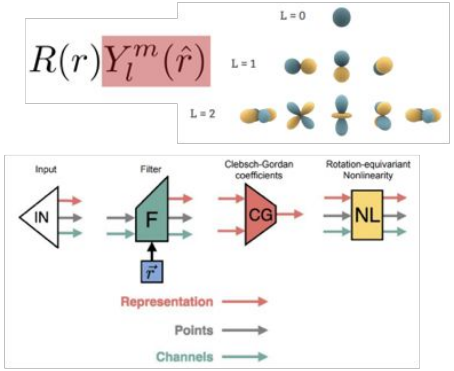
$$f_C(R_{ij}) = \begin{cases} 0.5\times\cos\left(\frac{\pi R_{ij}}{R_C}\right)+0.5 & \text{for } R_{ij} \le R_C \\ 0.0 & \text{for } R_{ij} > R_C \end{cases}$$

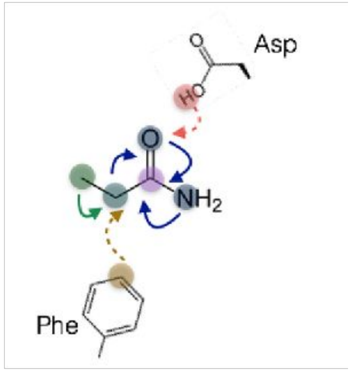
$$G_m^{R} = \sum^{\text{all atoms}} e^{-\eta(R_{ij}-R_s)^2} f_C(R_{ij})$$

$$G_m^{A_{\text{mod}}} = 2^{1-\zeta} \sum_{j,k\neq i}^{\text{all atoms}} \left(1+\cos(\theta_{ijk}-\theta_s)\right)^{\zeta} \exp\left[-\eta\left(\frac{R_{ij}+R_{ik}}{2}-R_s\right)^2\right] f_C(R_{ij}) f_C(R_{ik})$$









Olexandr Isayev will tell us about ANI and friends next
Could we extend SMIRNOFF to include ANI-like models?

Killer app: Fit hybrid models (ML for valence + physical long-range interactions) to both QM and physical property data.

Need to standardize how simulation packages will support ML potentials:
MolSSI Interoperability Workshop 3-5 Nov 2019 in Brooklyn NY
https://molssi.org/2019/07/29/molssi-workshop-molecular-dynamics-software-interoperability/

MACHINE LEARNING WILL COME TO REPLACE PAIN POINTS

AM1-BCC presents significant challenges to growth:

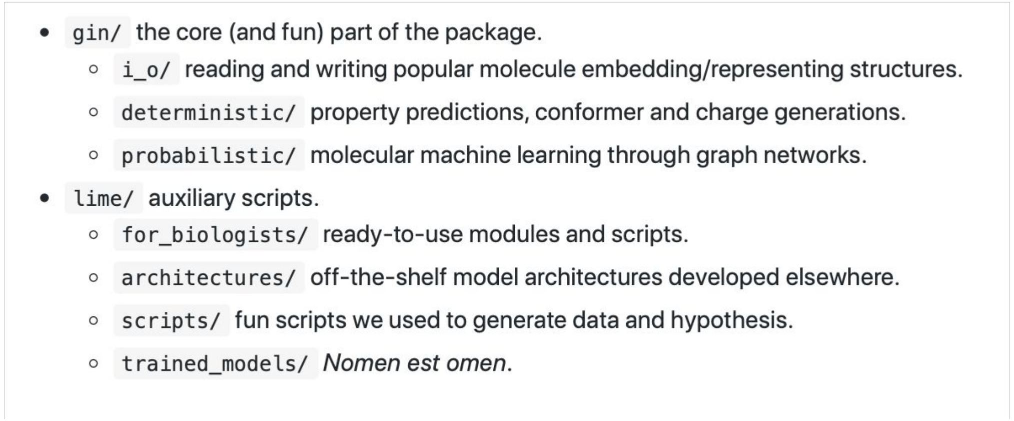
- AM1 ~15 seconds (with wide variability) per small molecule (and still isn’t deven particularly good)
- Cannot scale to biopolymers to provide consistent charge model
- Requires conformer generator, which is toolkit-dependent
- Even ELF10 (conformer selection/averaging) still produces toolkit-dependent charges
- Few good choices for AM1 software

Wiberg bond orders (WBOs) present same challenges

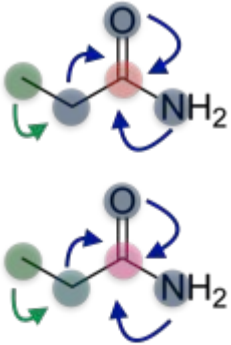
Can we replace this with a simple machine learning model that will produce conformer-independent charges and scale to biopolymers?

GRAPH CONVOLUTIONAL NETWORKS FOR PARTIAL CHARGES

Graph Inference on MoLEcular Topology github.com/choderalab/gimlet



- `gin/` the core (and fun) part of the package.
    - `i_o/` reading and writing popular molecule embedding/representing structures.
    - `deterministic/` property predictions, conformer and charge generations.
    - `probabilistic/` molecular machine learning through graph networks.
- `lime/` auxiliary scripts.
    - `for_biologists/` ready-to-use modules and scripts.
    - `architectures/` off-the-shelf model architectures developed elsewhere.
    - `scripts/` fun scripts we used to generate data and hypothesis.
    - `trained_models/` *Nomen est omen.*



https://doi.org/10.1021/acscentsci.8b00507

Yuanqing Wang (MSKCC)

https://github.com/choderalab/gimlet

COMBINING PHYSICS WITH MACHINE LEARNING IS POWERFUL

Define the contribution of potential energy by atomic charge as $E_A(Q)$. It has been shown that the second-order Taylor expansion is sufficient to approximate.

$$E_A(Q) \approx E_{A0} + Q_A\left(\frac{\partial E}{\partial Q}\right)_{A0} + \frac{1}{2}Q_A^2\left(\frac{\partial^2 E}{\partial Q^2}\right)_{A0}$$

the first- and second-order derivates are termed *electronegativity* and *hardness*.

$$e_A \equiv \left(\frac{\partial E}{\partial Q}\right)_{A0} \approx \frac{1}{2}(E_A(+1) - E_A(-1)) = \frac{1}{2}(\mathrm{IP} + \mathrm{EA})$$

$$s_A \equiv J_{AA}^0 \equiv \left(\frac{\partial^2 E}{\partial Q^2}\right)_{A0} \approx E_A(+1) + E_A(-1) - 2E_A(0) = \mathrm{IP} - \mathrm{EA}$$

where IP and EA are **ionization potential** and **electron affinity**.

Rappe and Goddard (1991) doi:10.1021/j100161a070
Gilson, GIlson, and Potter (2003) doi:10.1021/ci034148o

COMBINING PHYSICS WITH MACHINE LEARNING IS POWERFUL

Adapting the clever trick by Gilson et al., we predict the first- and second- order derivative of $E_A(Q)$, and form this problem as a double optimization, where,

$$\{\hat{e}_i, \hat{s}_i\} = \underset{e_i, s_i}{\operatorname{argmin}}(\underset{q_i}{\operatorname{argmin}} \sum_i e_i q_i + \frac{1}{2} s_i q_i^{\,2})$$

subject to:

$$\sum_i q_i = \sum_i q_{i0}$$

For the second minimization, i.e. solving {qi} with given {ei} and {si}, it could be solved analytically using Lagrange multipliers,

$$\hat{q}_i = -e_i s_i^{-1} + s_i^{-1} \frac{Q + \sum_i e_i s_i^{-1}}{\sum_j s_j^{-1}}$$

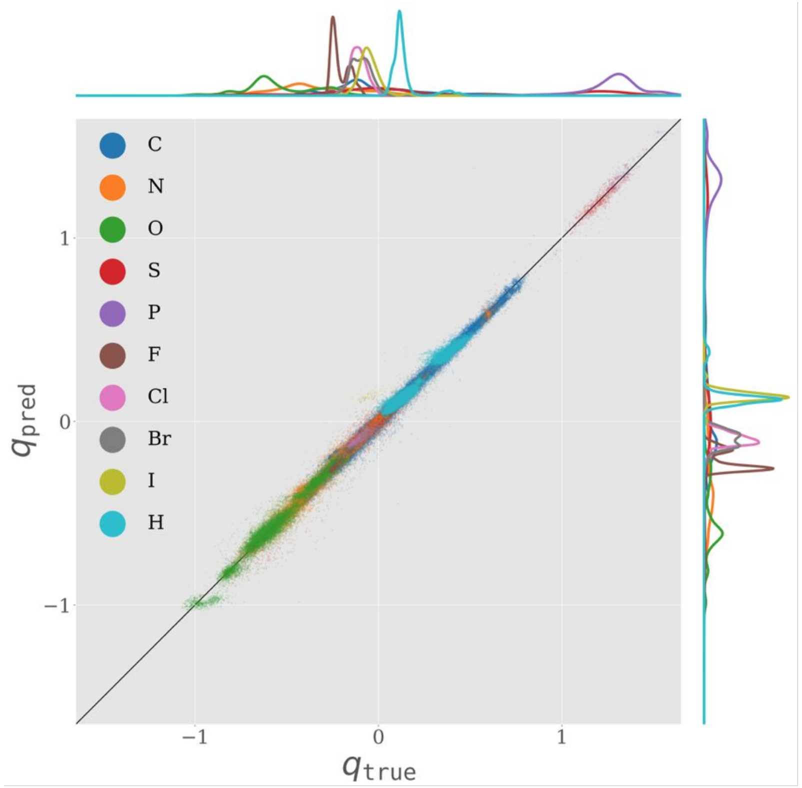
whose Jacobian and Hessian are trivially easy to calculate.

Rappe and Goddard (1991) doi:10.1021/j100161a070

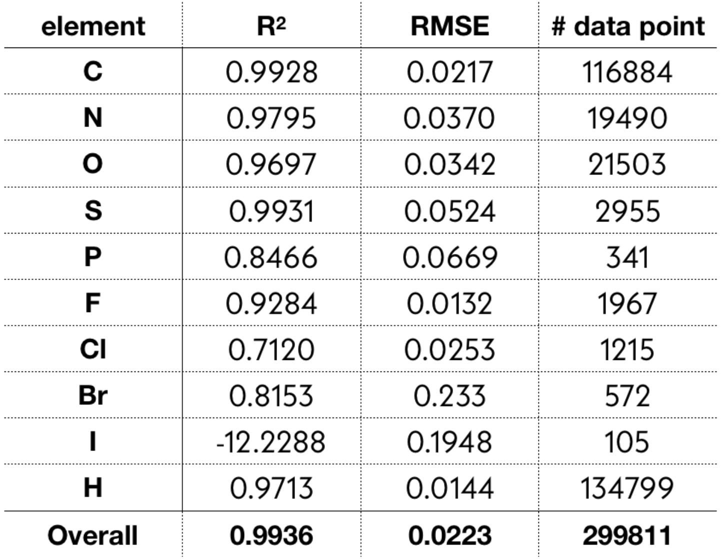
Gilson, GIlson, and Potter (2003) doi:10.1021/ci034148o

Yuanqing Wang (MSKCC)

COMBINING PHYSICS WITH MACHINE LEARNING IS POWERFUL



Predicted versus reference charge on held-out test set.

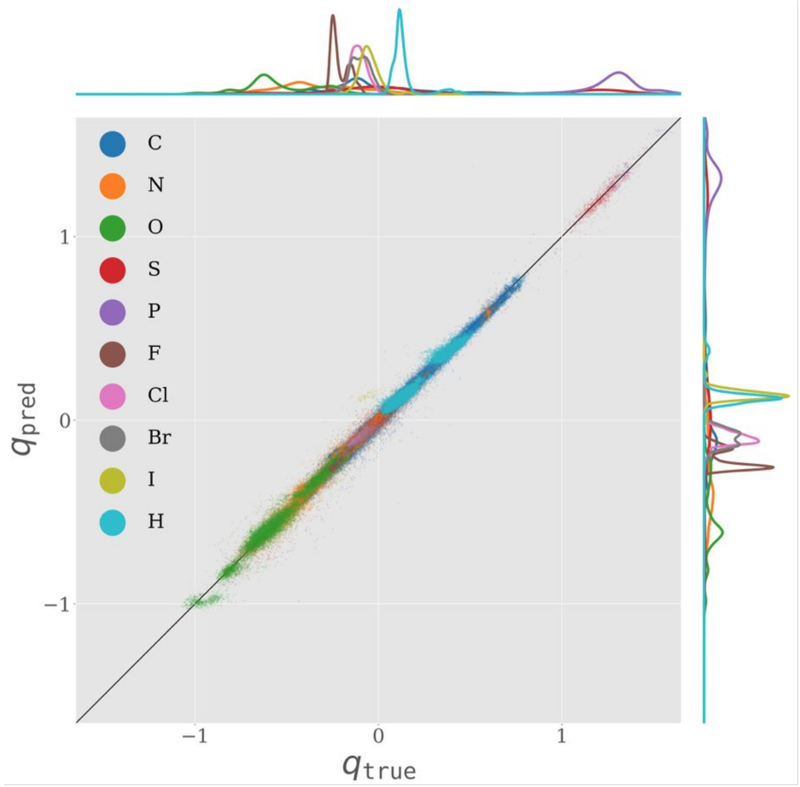


element	R²	RMSE	# data point
C	0.9928	0.0217	116884
N	0.9795	0.0370	19490
O	0.9697	0.0342	21503
S	0.9931	0.0524	2955
P	0.8466	0.0669	341
F	0.9284	0.0132	1967
Cl	0.7120	0.0253	1215
Br	0.8153	0.233	572
I	-12.2288	0.1948	105
H	0.9713	0.0144	134799
Overall	**0.9936**	**0.0223**	**299811**

R^2, RMSE, and number of data points in held-out test set grouped by element type.

DFT/COSMO-RS charges from Bleiziffer, Schaller, and Riniker ChEMBL set: doi:10.1021/acs.jcim.7b00663

COMBINING PHYSICS WITH MACHINE LEARNING IS POWERFUL



Predicted versus reference charge on held-out test set.

500x faster than AM1-BCC

Portable (currently TensorFlow)

<35 ms/molecule on CPU or GPU

Scalable to biopolymers

Yuanqing Wang (MSKCC)

BESPOKE MOLECULE FITTING PIPELINE WILL REUSE COMPONENTS

Reuse existing components in local workflow:

- **QCFractal / TorsionDrive / GeomeTRIC / QCEngine**
  - **psi4** quantum chemical calculations
  - **ANI1-ccx** fast approximate coupled-cluster
- Optional **QCArchive instance** for local caching of QC calcs
- **RESP/RESP2** multi-conformer electrostatics

SUSTAINABILITY IS KEY TO LONG-TERM SUCCESS

Funding

- Open Force Field Consortium: Continue to focus on near-term industry needs
- NSF CHE: Exploration of Bayesian inference for parameterization
- Institutional funding
- MolSSI Software Fellowships
- NIH Focused Technology R&D R01: Biopolymers and heterogeneous systems
- Chan Zuckerberg Essential Open Source Software for Science
- Other funding sources?

Community

- Formed Scientific Advisory Board to help maximize impact in force field communities
  - We want force field developers to be more productive with our tools
- Working with package developers to better integrate OFF tools and force fields
- Aiming to nucleate a supportive, active community of developers and users