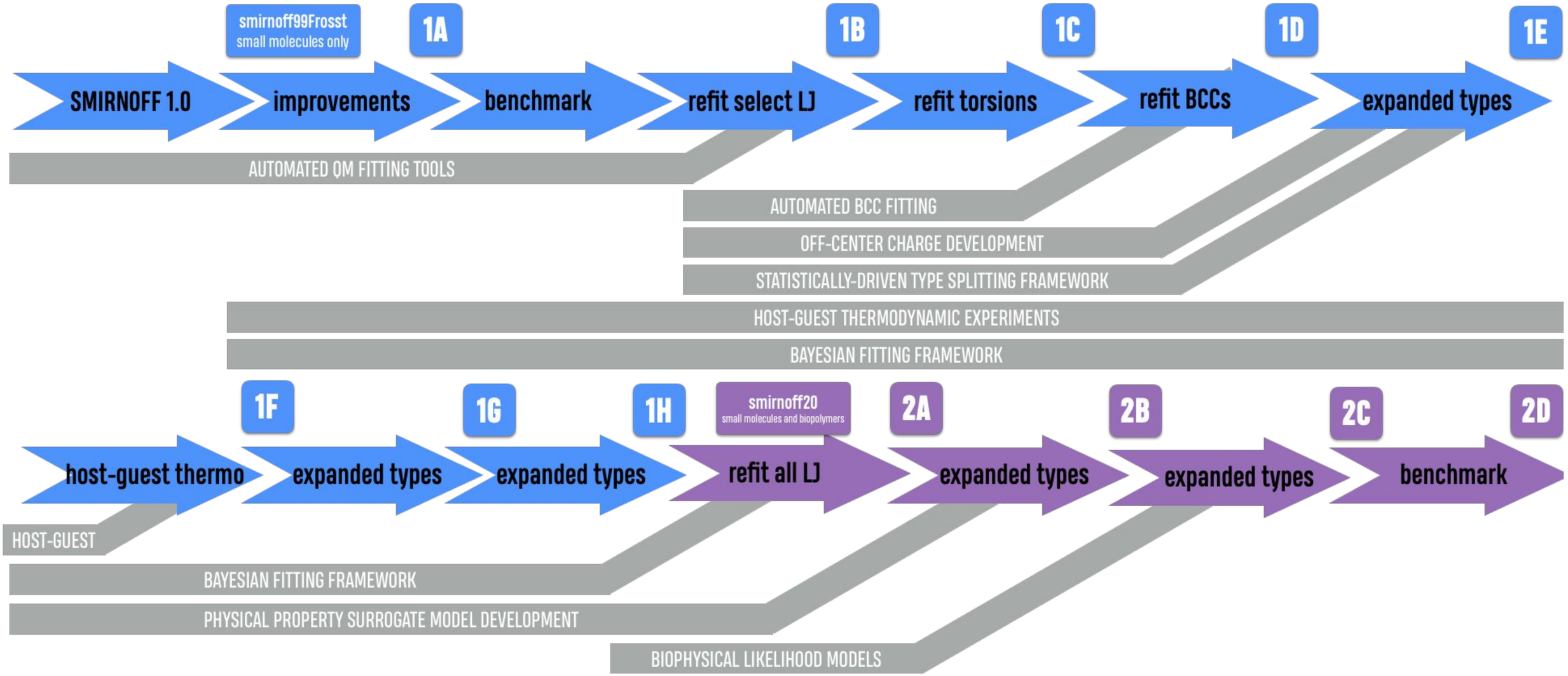
Bayesian inference of force fields

Open Force Field Initiative
Consortium meeting, 8 Jan 2019

John Chodera (MSKCC), **Michael Shirts** (UC Boulder), **Josh Fass** (MSKCC), **Owen Madin** (UC Boulder), **Chaya Stern** (MSKCC), **Richard Messerly** (NIST)

#bayesian-inference on Slack

Bayesian inference will power the second generation of OFFI force fields



Evolution of parameterization strategies

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

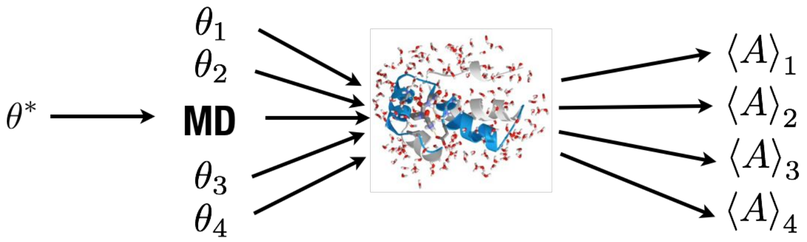
Bayesian methods resolve major force field problems

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

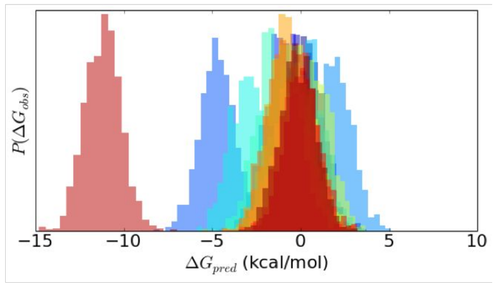
posterior | likelihood | prior

$\mathcal{D}$ data

θ forcefield parameters



Multiple parameter sets in, multiple estimates out



Markov chain Monte Carlo can avoid getting trapped in local minima and find “lowest free energy” basin in parameter space to provide **better generalizability** while automating parameter fitting

Bayesian methods automatically **penalize unnecessary complexity** to avoid overfitting (via too many atom types or functional forms with unwarranted complexity)

By reweighting predictions with multiple posterior parameter samples, **predictions of force field error** due to uncertainty in parameters or models can provide systematic error estimate essentially for free

What do we want in a force field parameterization approach?

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)

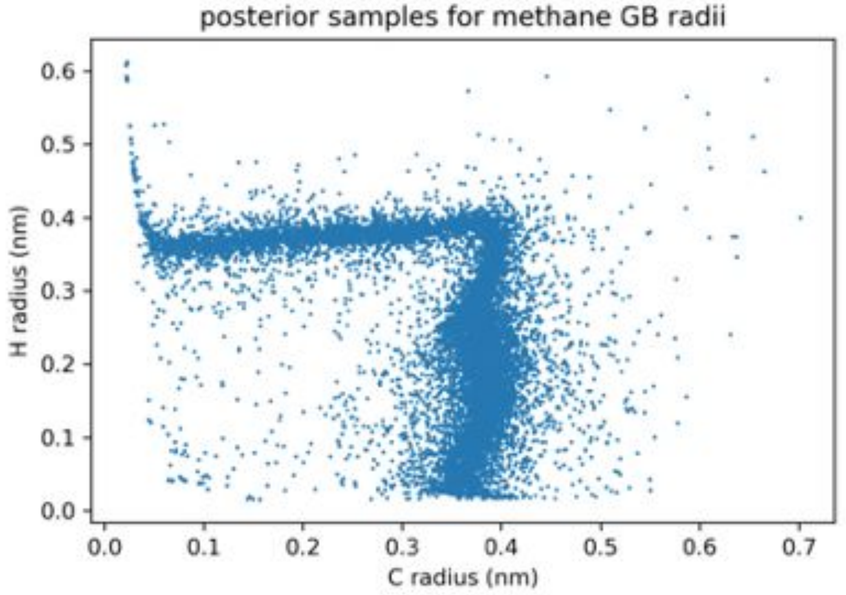
We want to make data-driven decisions about which choices are best justified by data
The data will tell us which functional forms or mixing rules are sufficiently well-justified

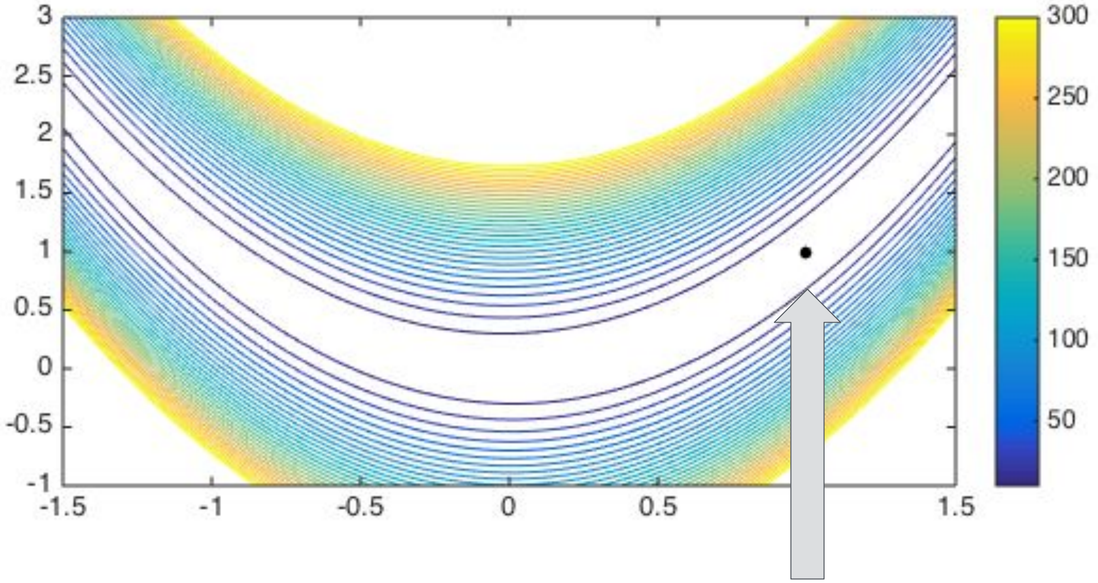
Data is automatically weighted by its measurement error
No need for humans to specify weights for different measurements or regularization schemes

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

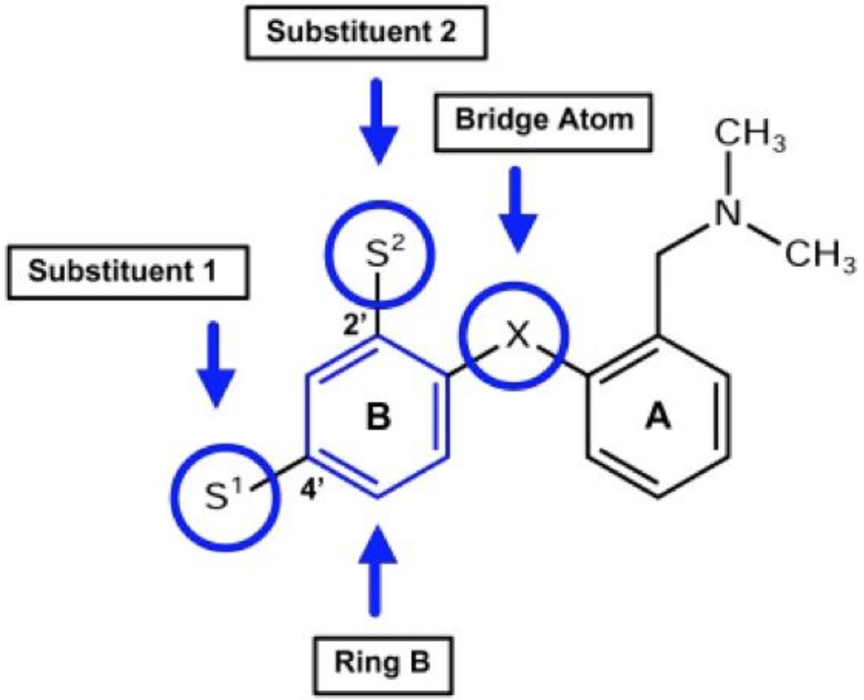


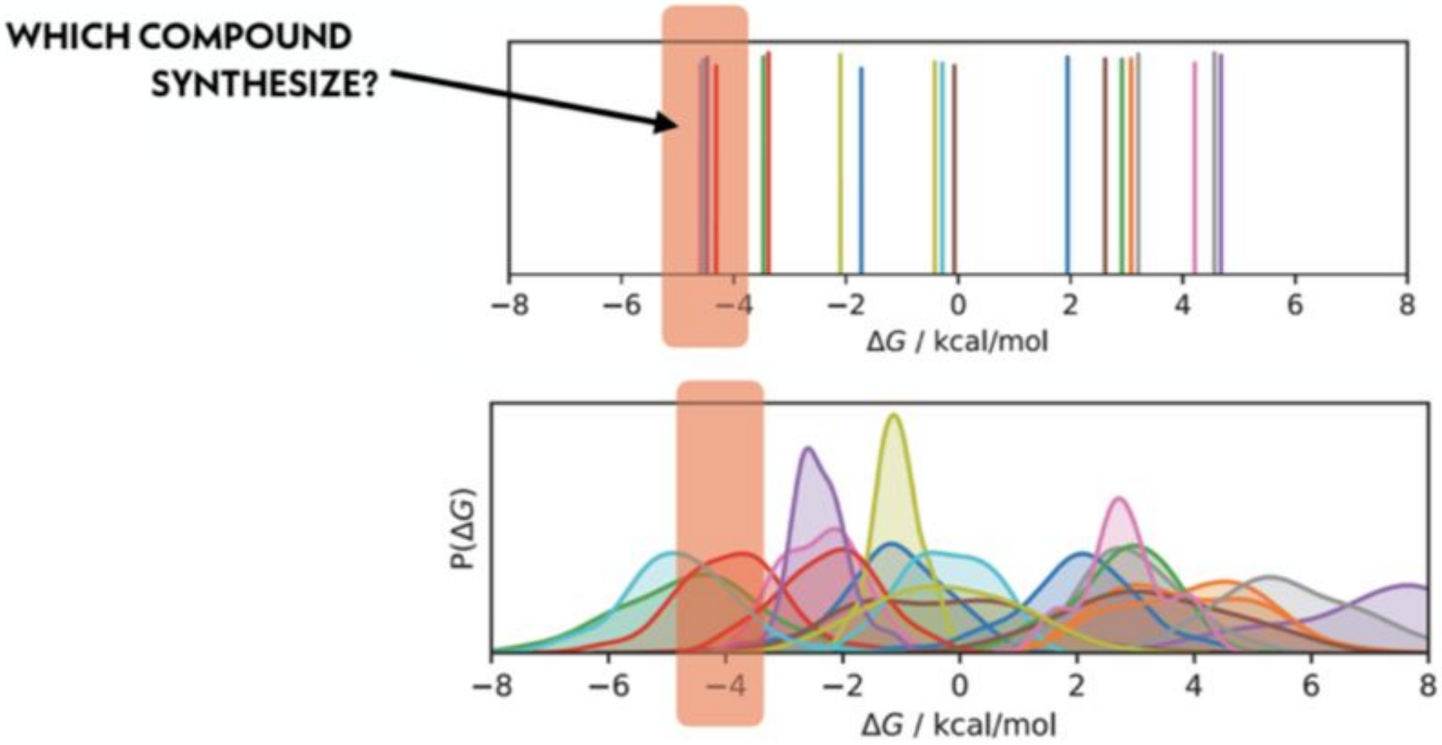


why just this set of parameters?

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

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

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

likelihood

$\mathcal{D}$ 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!

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

posterior **likelihood** **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

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

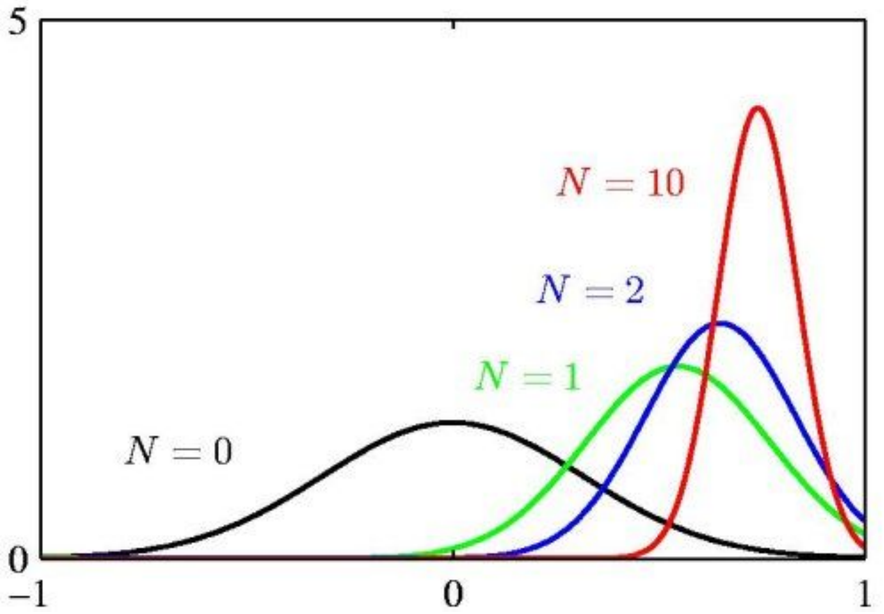
Experimental error models come from models of the experimental measurement process and may involve unknown parameters corresponding to instrumental reliability:

e.g. density measurement

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

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

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

posterior **likelihood** **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

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

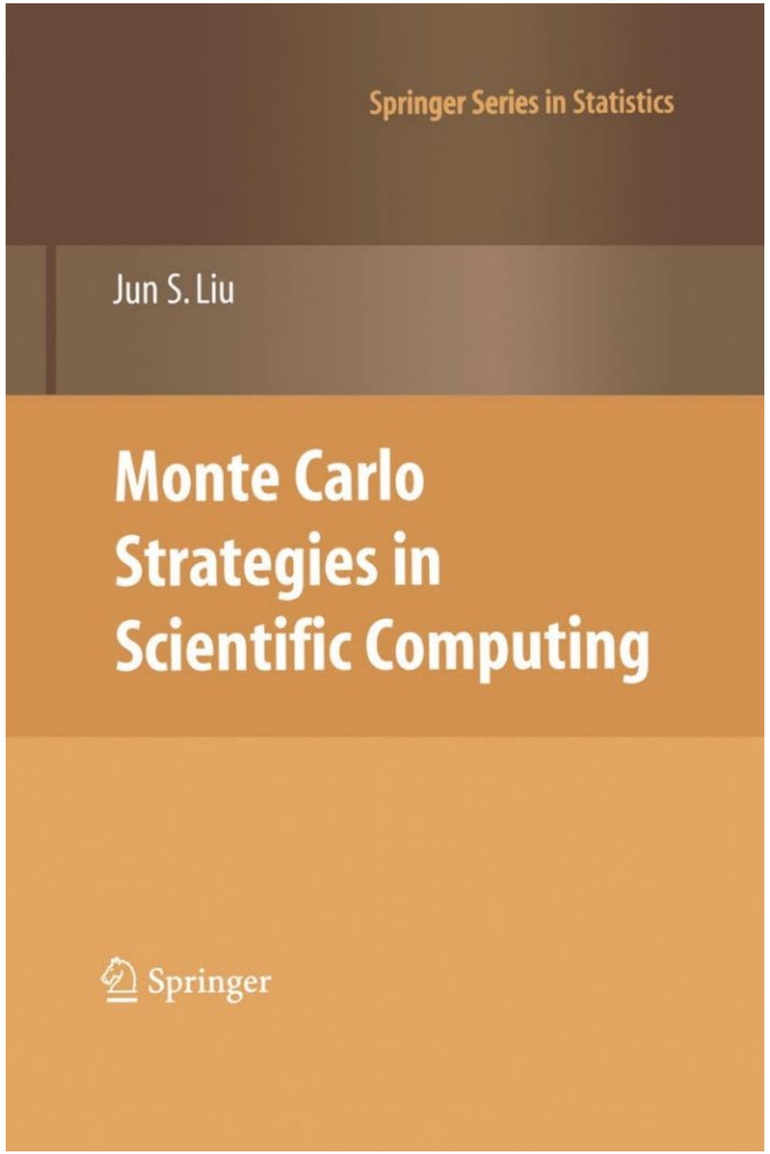
Langevin integrators can approximately sample from very highly multidimensional problems using gradient information, and good Langevin integrators (BAOAB) are accurate and efficient

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

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



statistical inference

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

We can utilize a large variety of experimental data

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

The NIST ThermoML Archive stores physical property data using an IUPAC standard

Applied Chemicals and Materials Division
TRC Group

NIST National Institute of Standards and Technology

ThermoML

AN XML-BASED IUPAC STANDARD FOR STORAGE AND EXCHANGE OF EXPERIMENTAL THERMOPHYSICAL AND THERMOCHEMICAL PROPERTY DATA

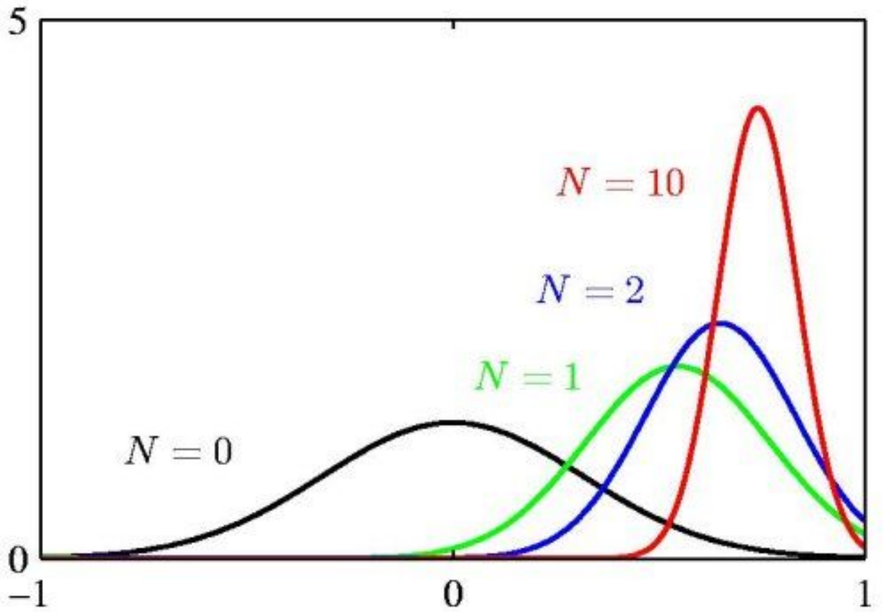
with Kenneth Kroenlein, NIST TRC

(From 2015)	Number of measurements remaining	
Filter step	Mass density	Static dielectric
1. Single Component	136212	1651
2. Druglike Elements	125953	1651
3. Heavy Atoms	71595	1569
4. Temperature	38821	964
5. Pressure	14103	461
6. Liquid state	14033	461
7. Aggregate T, P	3592	432
8. Density+Dielectric	246	246

http://doi.org/10.1021/acs.jpcb.5b06703

Conditioning on more data reduces uncertainty

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

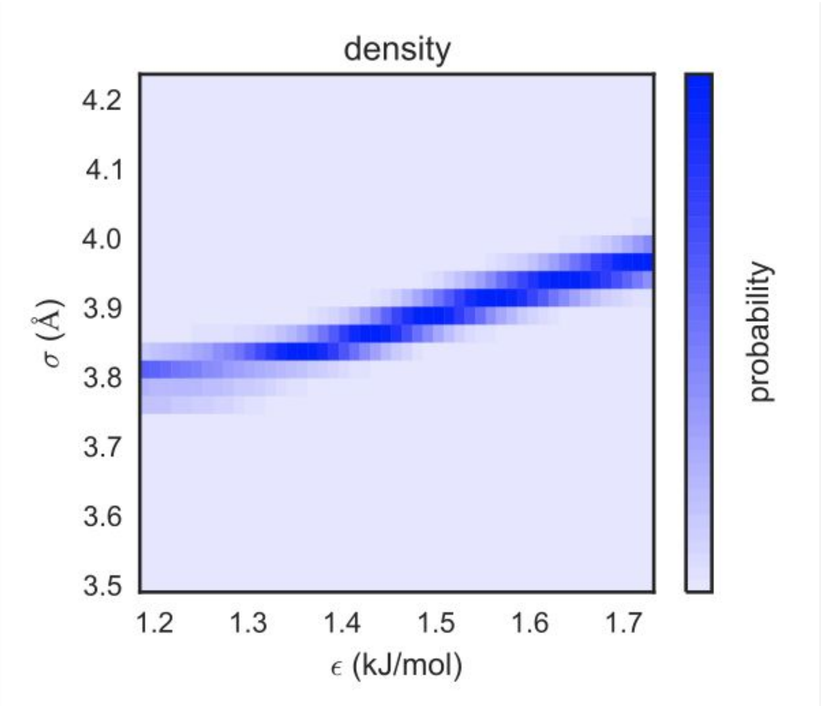


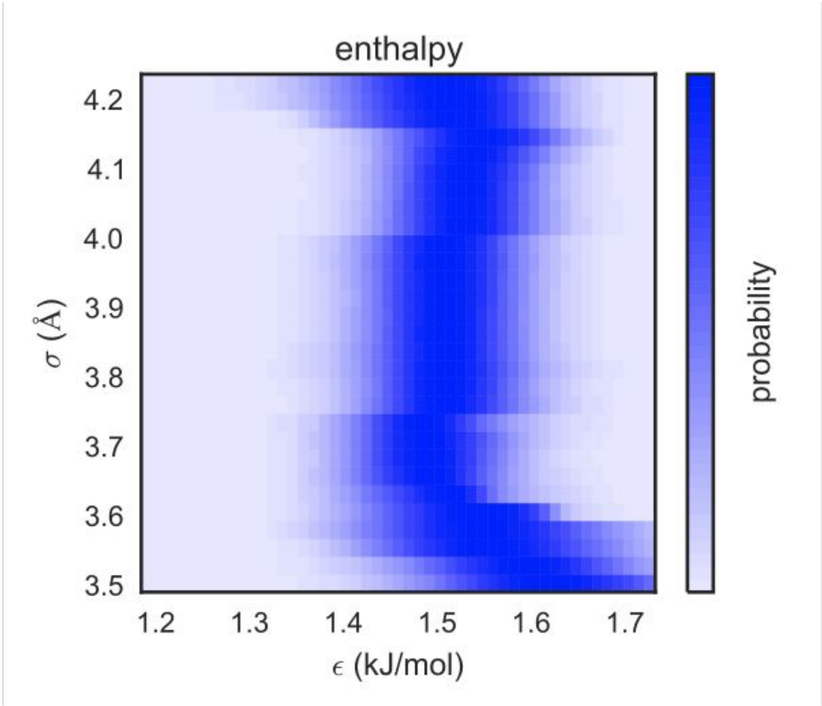
We can quantify uncertainty in an **information-theoretic** manner $S[p(x)] = -\int dx\, p(x) \ln p(x)$

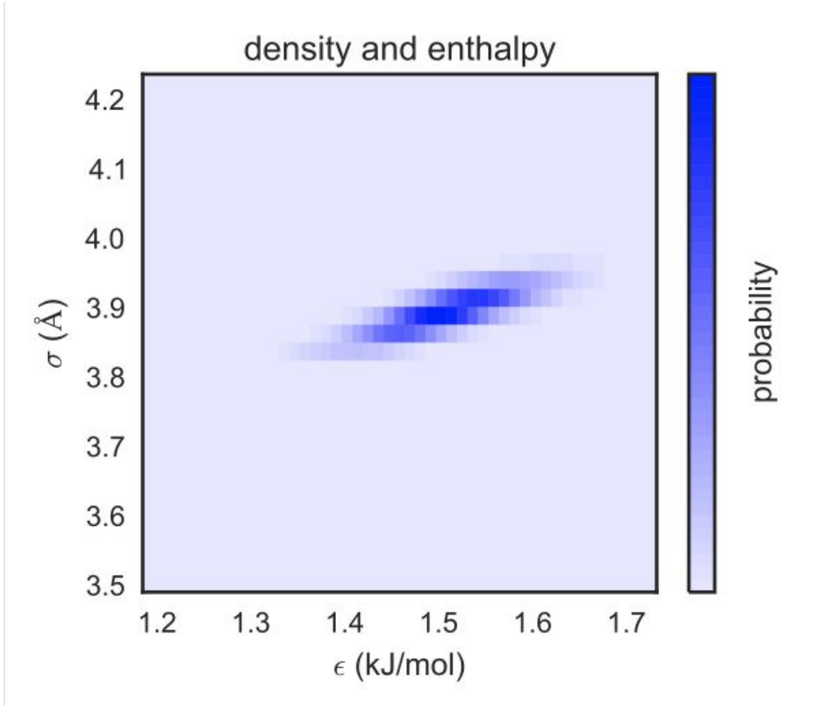
Interactive illustration: http://rpsychologist.com/d3/bayes/

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

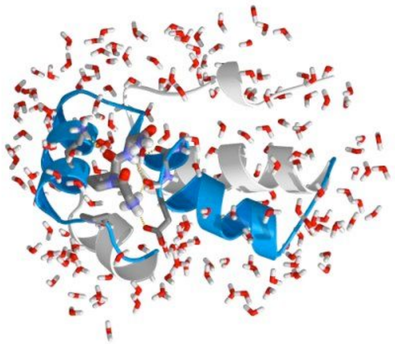
The **marginal posterior probability** for an expectation describes our confidence in a prediction:

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

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

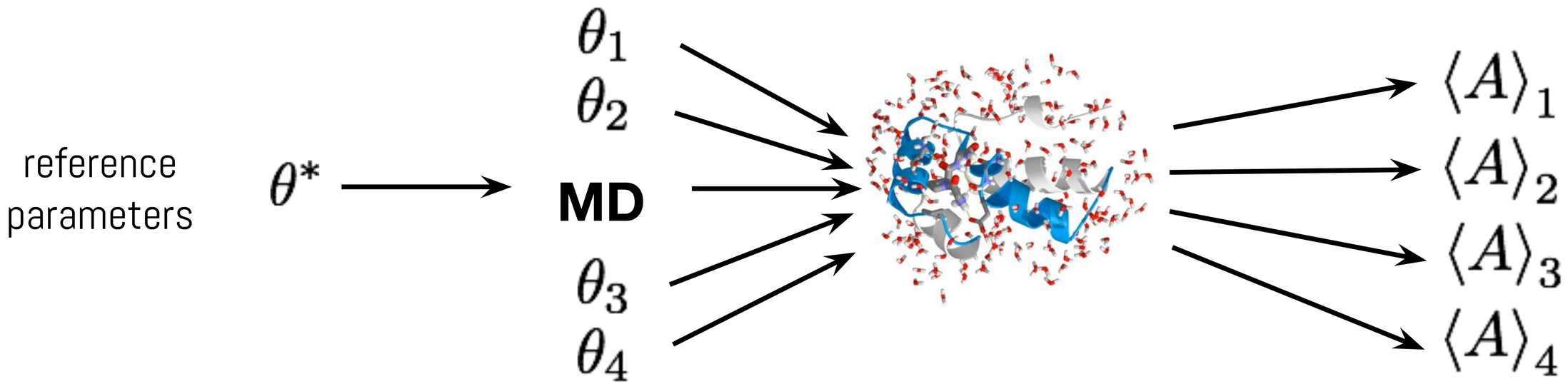
the “one true force field”

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

One set of parameters in, one computed result out

Only **statistical** error can be assessed

Predicting properties: The Bayesian way



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:

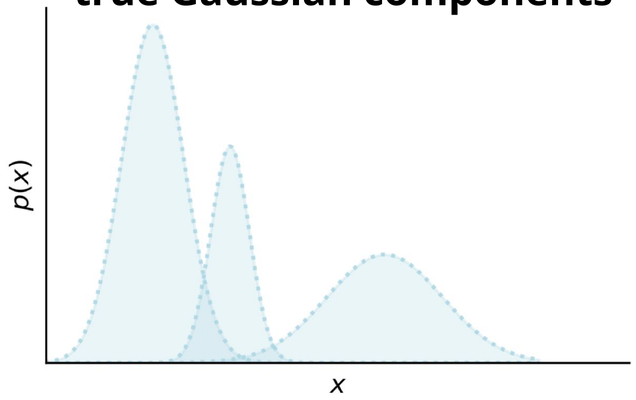
$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta_j|\mathcal{D})}{p(\theta_i|\mathcal{D})}\frac{P(\mathcal{M}_j)}{P(\mathcal{M}_i)}\frac{P(\mathcal{M}_i|\mathcal{M}_j)}{P(\mathcal{M}_i|\mathcal{M}_j)}\frac{T_{ji}(\theta_i|\theta_j)}{T_{ij}(\theta_j|\theta_i)}\right\}$$

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

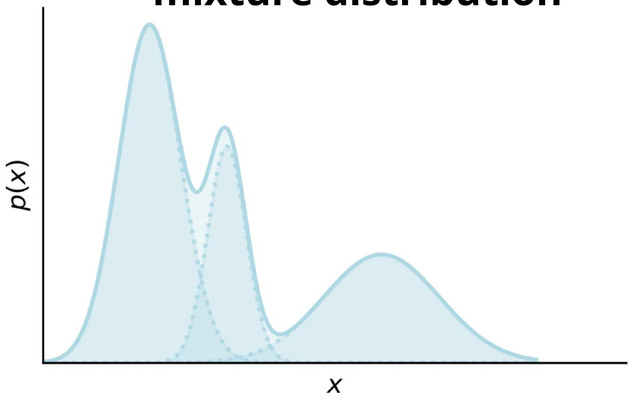
Reversible-jump Monte Carlo (RJMC) is a statistically principled way to sample an unknown number of types

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

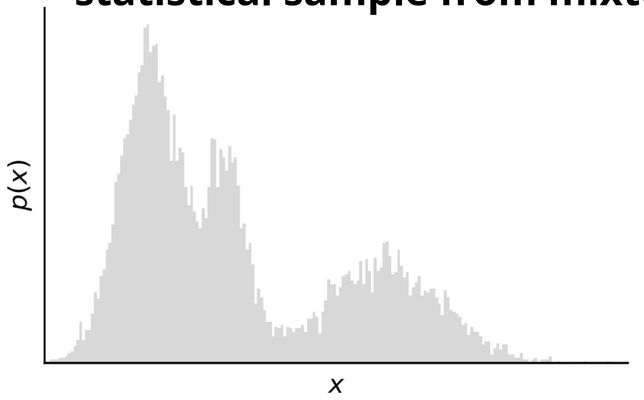
true Gaussian components



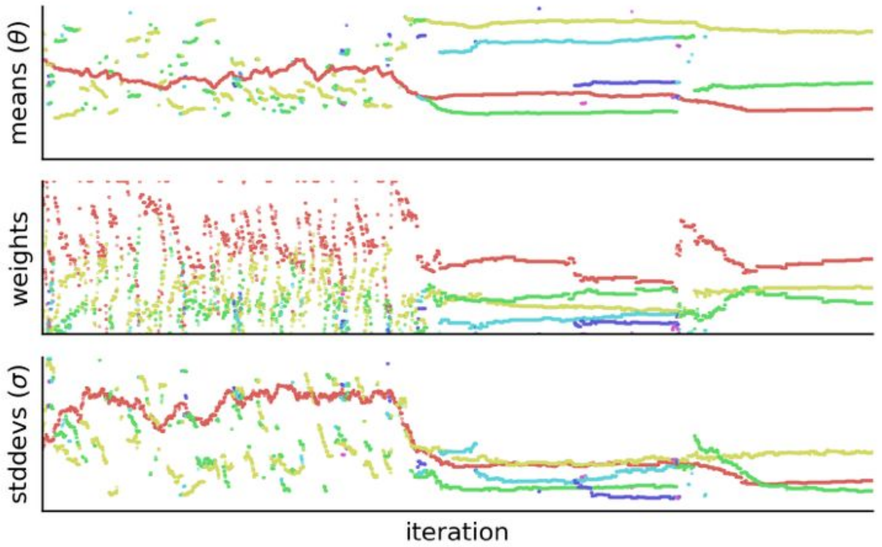
mixture distribution

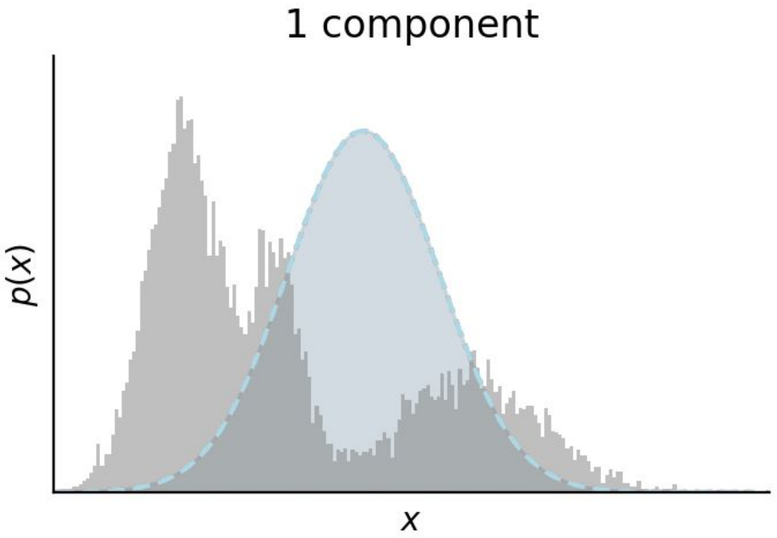


statistical sample from mixture



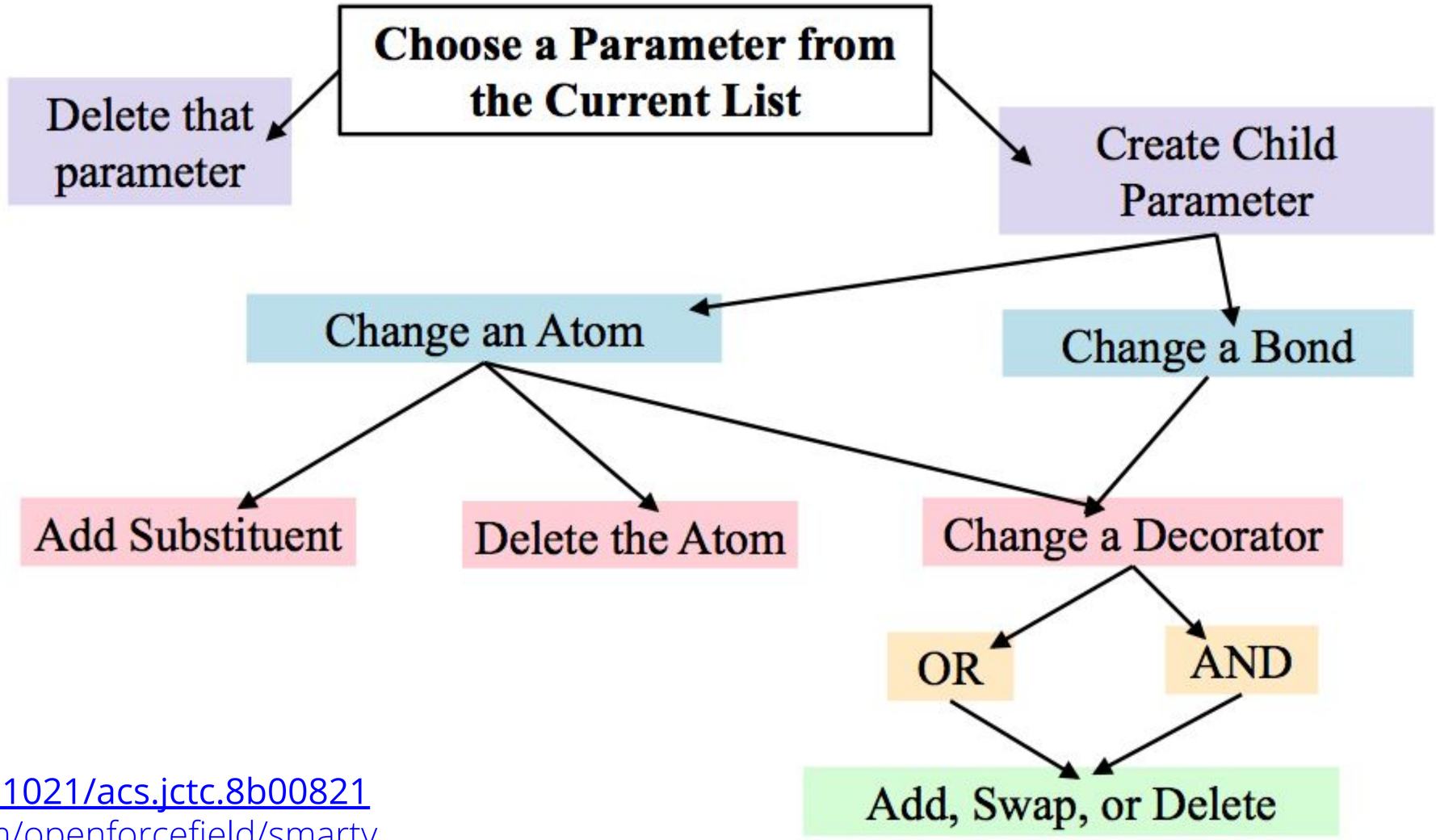
RJMC inference simulation from statistical sample





JOSH FASS

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



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

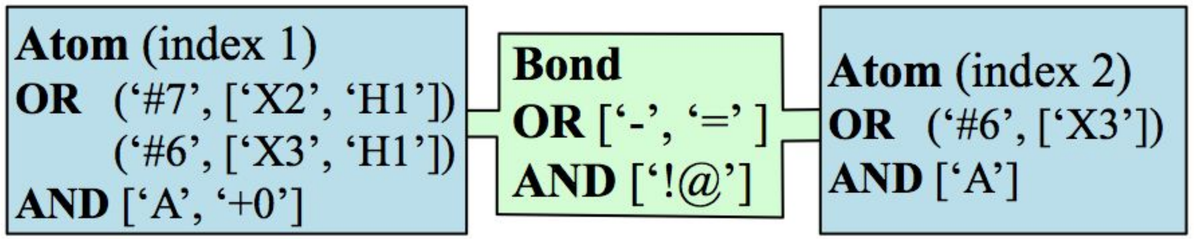
```
% total connectivity
 X1             connections-1
 X2             connections-2
 X3             connections-3
 X4             connections-4
 % total-h-count
 H0             total-h-count-0
 H1             total-h-count-1
 H2             total-h-count-2
 H3             total-h-count-3
 % formal charge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

=

proposed child types

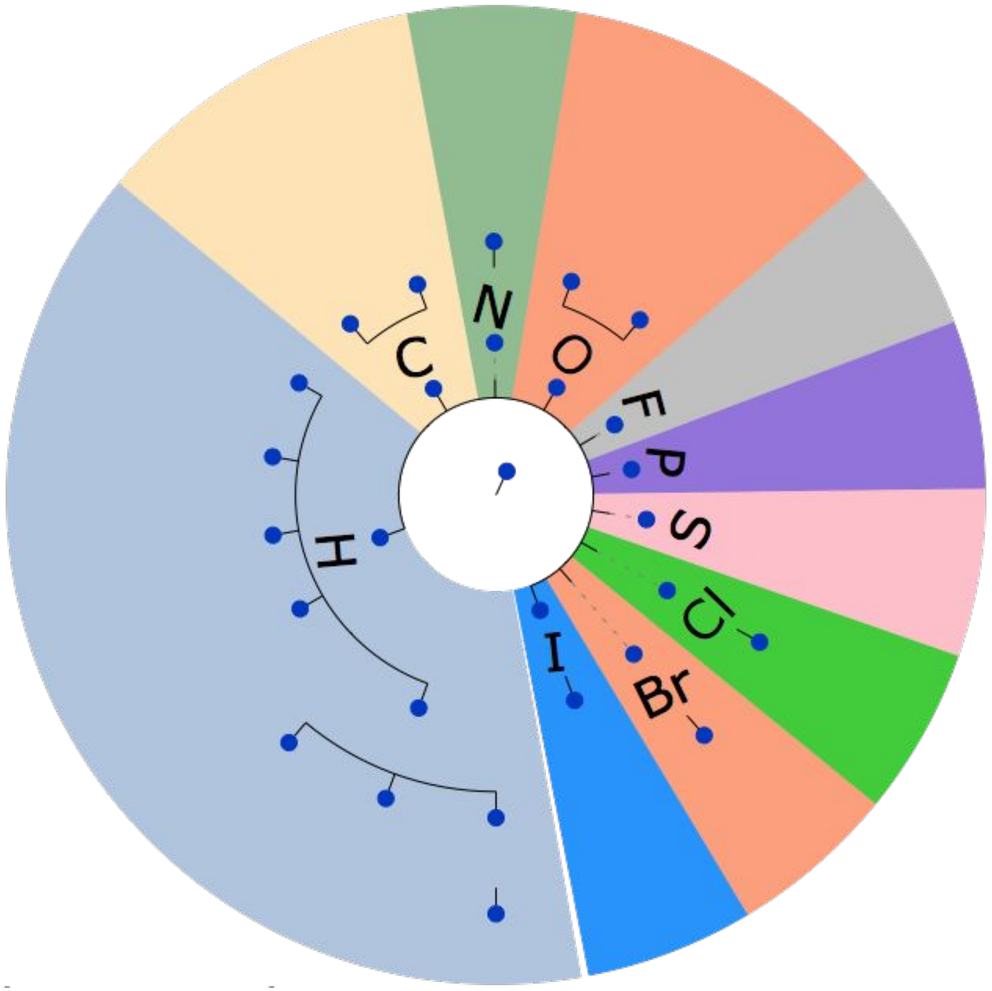
```
[#6X4:1]        tetrahedral carbon
[#6:1]~[#7]     carbon nitrogen-adjacent
```

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



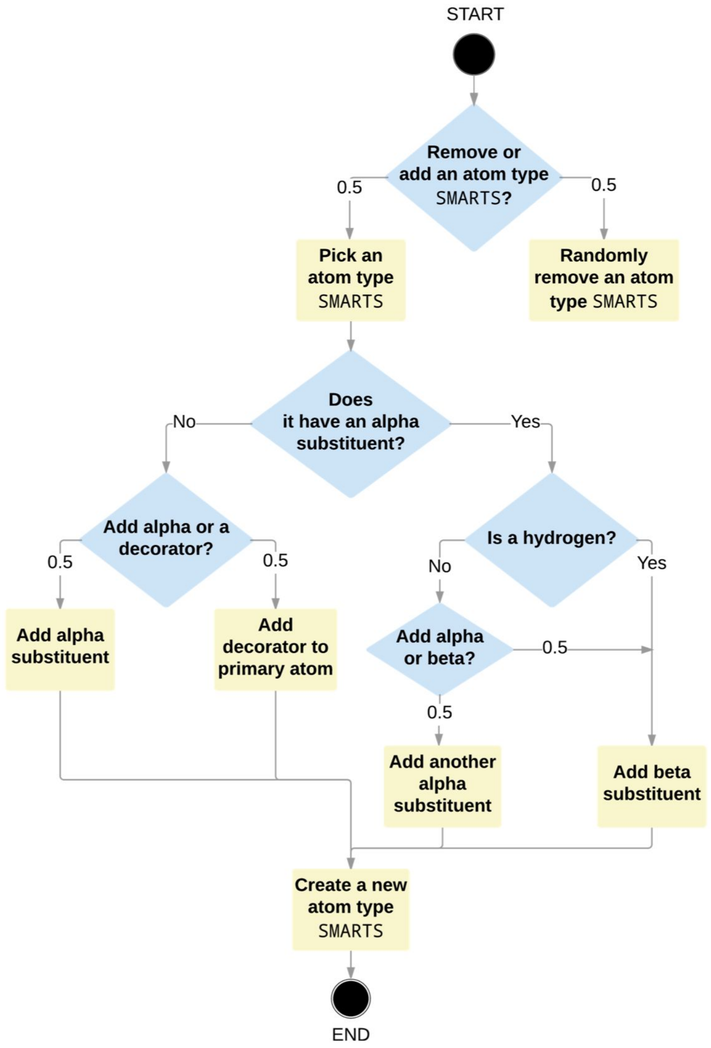
http://doi.org/10.1021/acs.jctc.8b00821
https://github.com/openforcefield/smarty

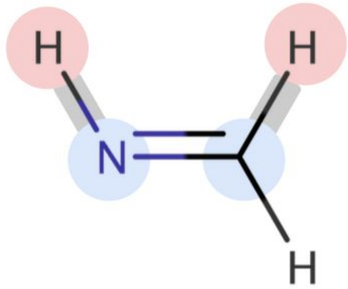
Sampling with this scheme can generate SMARTS-based typing trees with interesting complexity



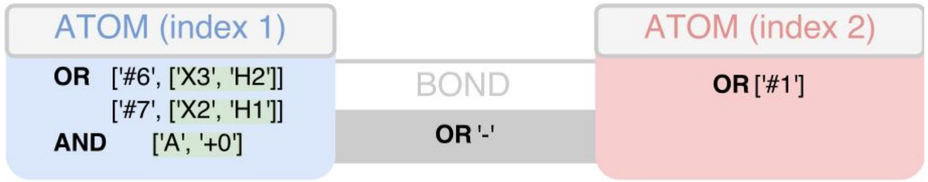

```
H ([#1:1])
     5262 ([#1:1]~[#6])
          4148 ([#1:1]~[#6!X3])
               8668 ([#1:1]~[#6!X3]~[$ewg2])
                    1874 ([#1:1]~[#6!X3](~[$ewg2])~[$ewg2])
                         4596
([#1:1]~[#6!X3](~[#7])(~[$ewg2])~[$ewg2])
               2356 ([#1:1]~[#6!X3X2])
             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

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





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



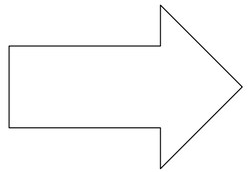
	AlkEthOH (%)		PhEthOH (%)		MiniDrugBank (%)	
	Initial	Maximum	Initial	Maximum	Initial	Maximum
All	67.8	100.0	54.5	100.0	40.5	93.0
Hydrogen	52.6	100	39.0	100	35.9	97.0
Carbon	100	100	71.0	100	39.0	95.7
Oxygen	63.6	100	84.1	100	38.3	98.0
Nitrogen	n/a	n/a	n/a	n/a	33.1	84.0
Sulfur	n/a	n/a	n/a	n/a	52.2	100

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)
```

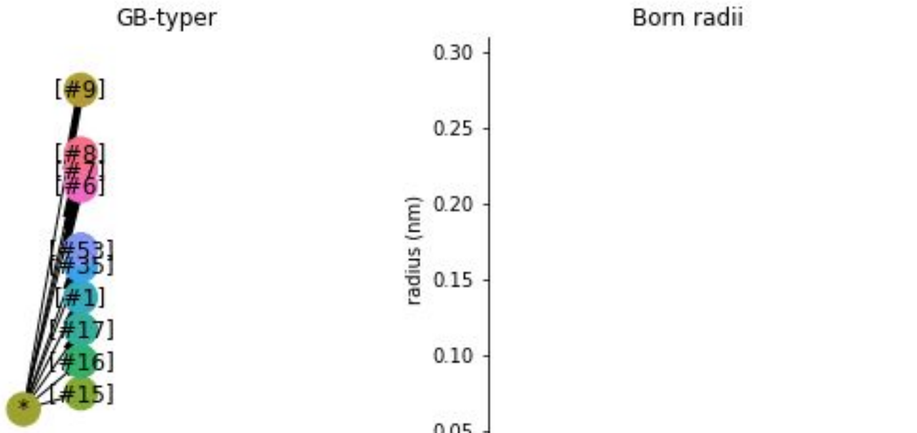


```
*             (r = 0.091 nm)
|-[#1]        (r = 0.135 nm)
|-[#6]        (r = 0.123 nm)
|-[#7]        (r = 0.160 nm)
|-[#8]        (r = 0.121 nm)
  |-[#8&X2]   (r = 0.127 nm)
|-[#9]        (r = 0.107 nm)
|-[#15]       (r = 0.131 nm)
|-[#16]       (r = 0.116 nm)
|-[#17]       (r = 0.100 nm)
|-[#35]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

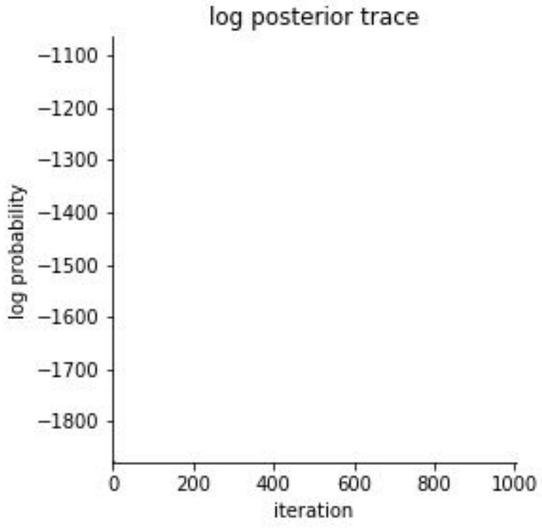
JOSH FASS

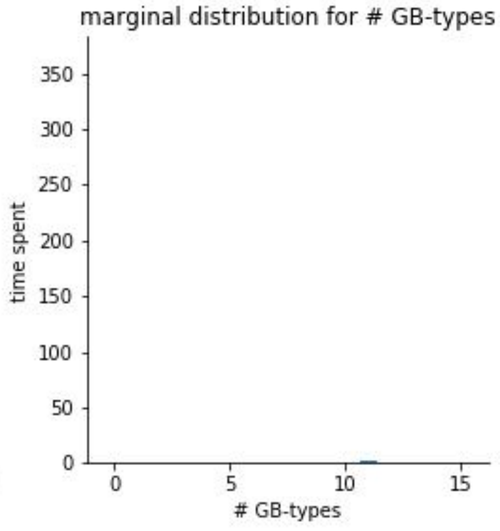
Initial experiment: Sampling over GBSA atom types fit to small molecule hydration free energies

Hierarchical SMIRNOFF types



log posterior

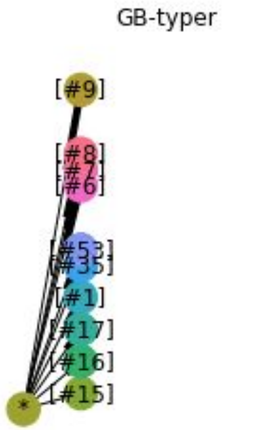


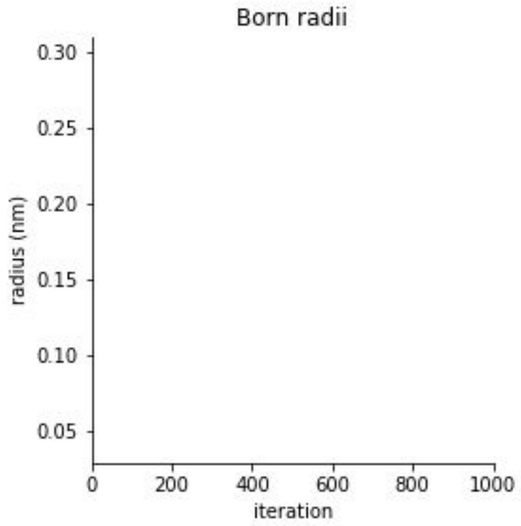


JOSH FASS

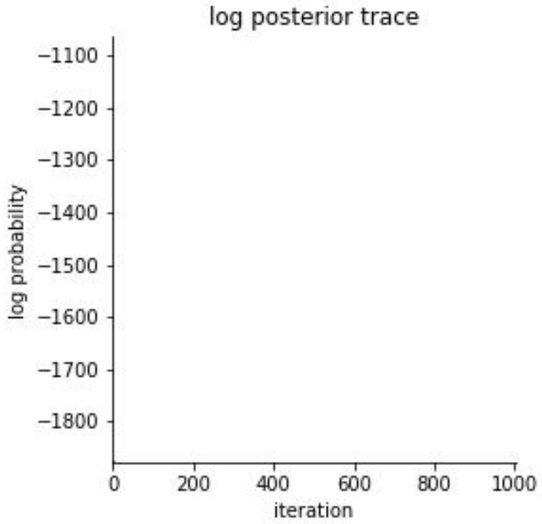
The RJMC approach can discover interesting chemistry!

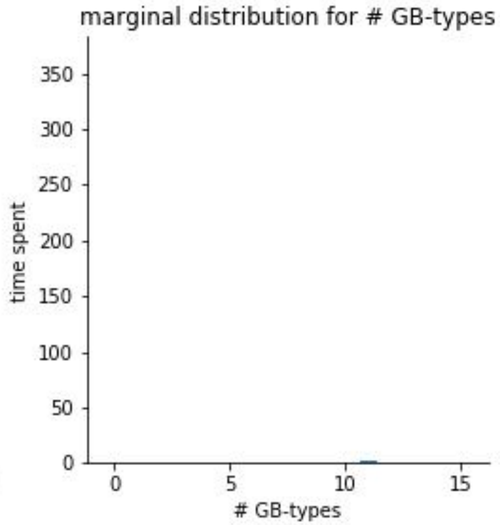
Hierarchical SMIRNOFF types





log posterior





JOSH FASS

9 hydrogen types

20 carbon types

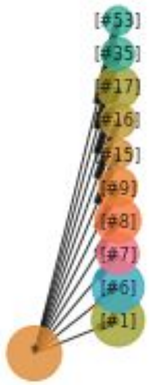
distinguishes nitriles!

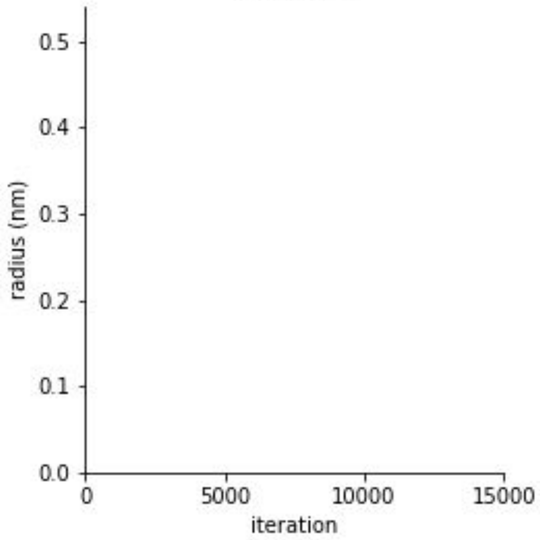
and sulfonyls!

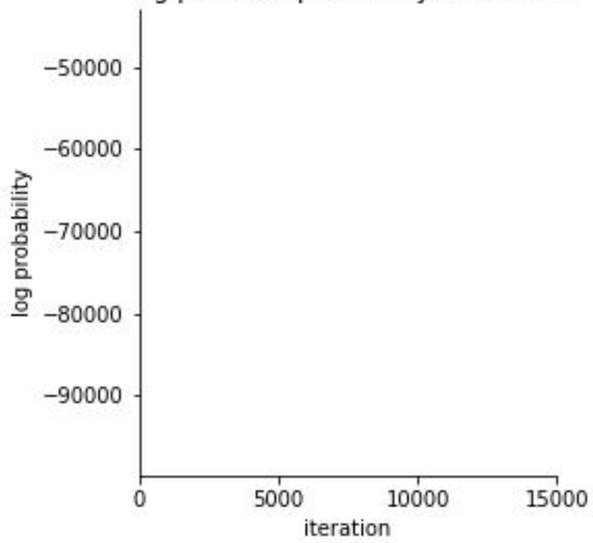
```
*
|-[#1]
  |-[#1]~[r5]
    |-[#1]~[r5]~[r7]
    |-[#1]~[r0]~[r5]
      |-[#1]~[r0]~[r6]~[r5]
    |-[#1]~[r5]~[#8]
  |-[#1]~[r0]
    |-[#1]~[r0&H3]
    |-[#1]~[r0]~[+1]
|-[#6]
  |-[#6]~[#35]
    |-[#6&H2]~[#35]
      |-[#6&H2]~[r0]~[#35]
  |-[#6]~[#7]
    |-[#6]~[#7]~[#7]
      |-[#6]~[#7]~[#7]
        |-[#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]
```

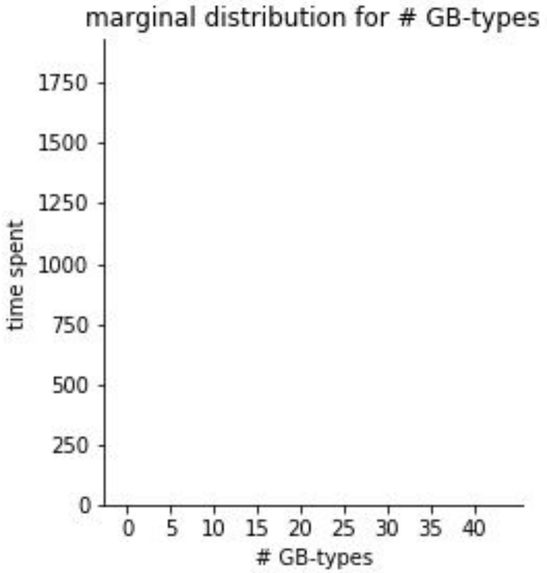
Initial experiment: Sampling over GBSA atom types fit to small molecule hydration free energies

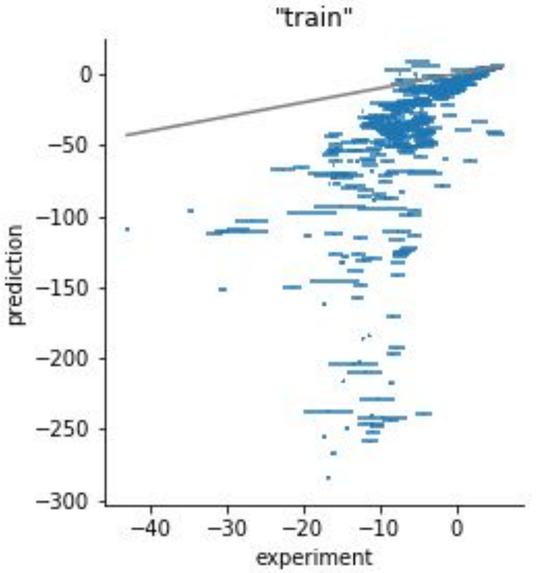
JOSH FASS

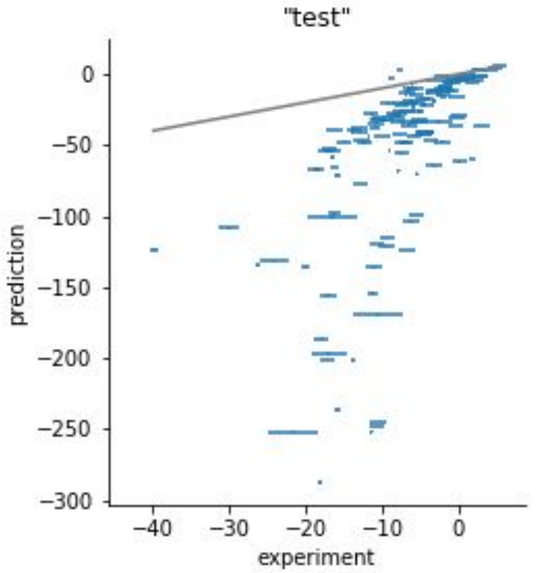












Our second-generation fitting approach will use Bayesian inference, reusing all our existing software components

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

Some challenges remain to be solved

Model space: If model space is too small, the posterior distribution may be misleadingly narrow, and predicted errors will be underestimated.

Our current approaches to handling this are:

* Ensure potential function model space is big enough to include realistic models
* Introduce an additional **model error** to capture model limitations, assigning Jeffries prior

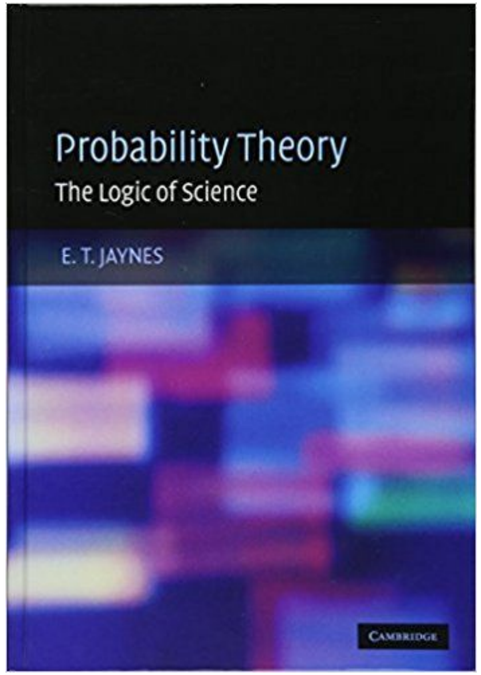
$$p(A_{\mathrm{obs}}|A(\theta)) \propto \exp\left[-\frac{1}{2}\frac{(A_{\mathrm{obs}}-A(\theta))^2}{(\sigma^2_{\mathrm{exp}}+\sigma^2_{\mathrm{model}})}\right]$$

$$p(\sigma_{\mathrm{model}}) \propto \sigma^{-1}_{\mathrm{model}}$$

Efficiency: RJMC is notoriously difficult to make efficient unless we can propose clever jumps that map high-probability regions between models. Nonequilibrium candidate Monte Carlo (NCMC) can help with this.

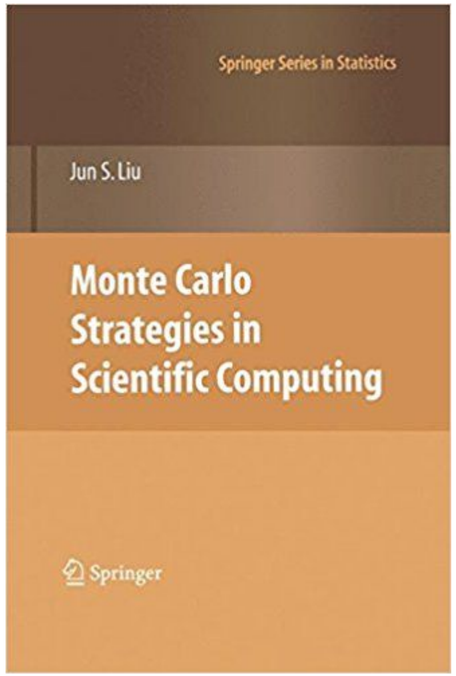
Where to learn more

theory and philosophy



https://www.amazon.com/Probability-Theory-Science-T-Jaynes/dp/0521592712

algorithms and numerics



https://www.amazon.com/Strategies-Scientific-Computing-Springer-Statistics/dp/0387763694

Acknowledgements

GitHub:

- github.com/openforcefield/openforcefield
- github.com/openforcefield/bayes-implicit-solvent

Slack:

- #bayesian-inference
- #bayes-implicit