Fitting and benchmarking condensed-phase properties

Open Force Field Initiative
Consortium meeting, 7 Jan 2019

John Chodera (MSKCC), **Michael Shirts** (UC Boulder), **Simon Boothroyd** (MSKCC), **Jeff Wagner** (OFF), **Owen Madin** (UC Boulder), **Richard Messerly** (NIST), **Josh Fass** (MSKCC), **David Slochower** (UCSD), **David Mobley** (UCI)

#propertycalculator #benchmarks #datasets on Slack

Condensed-phase data is critical for force field parameterization and benchmarking

For fixed-charge force fields, difficult (or impossible) to reproduce properties we care about (affinities) without incorporating neglected effects (polarizability) in some way.

Even with polarizability, fitting to only QM neglects nuclear quantum effects that may be important in properties we care about (such as hydrogen bonding).

What we care **most** about (binding affinities, ligand occupancies, protein conformations) are most directly related to condensed-phase experimental data.

Protein:ligand binding data difficult to use in parameterization and benchmarking: data quality, diversity and dynamic range, and difficulties in convergence

Properties of organic liquids in the condensed phase are the closest analogs that can be computed rapidly and reliably enough to be used in parameterization, and there is an abundance of high-quality thermodynamic data that has yet to be exploited

Different physical properties inform different parameters

EXPERIMENTAL DATA

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

QM DATA

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

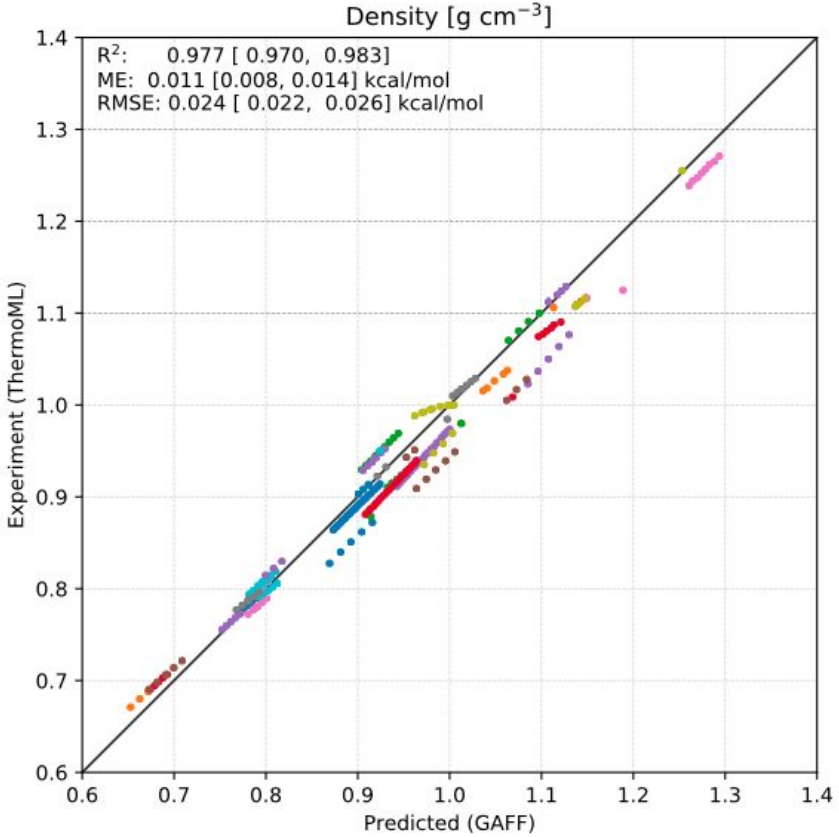
Many physical properties to choose from! Prioritization is important.

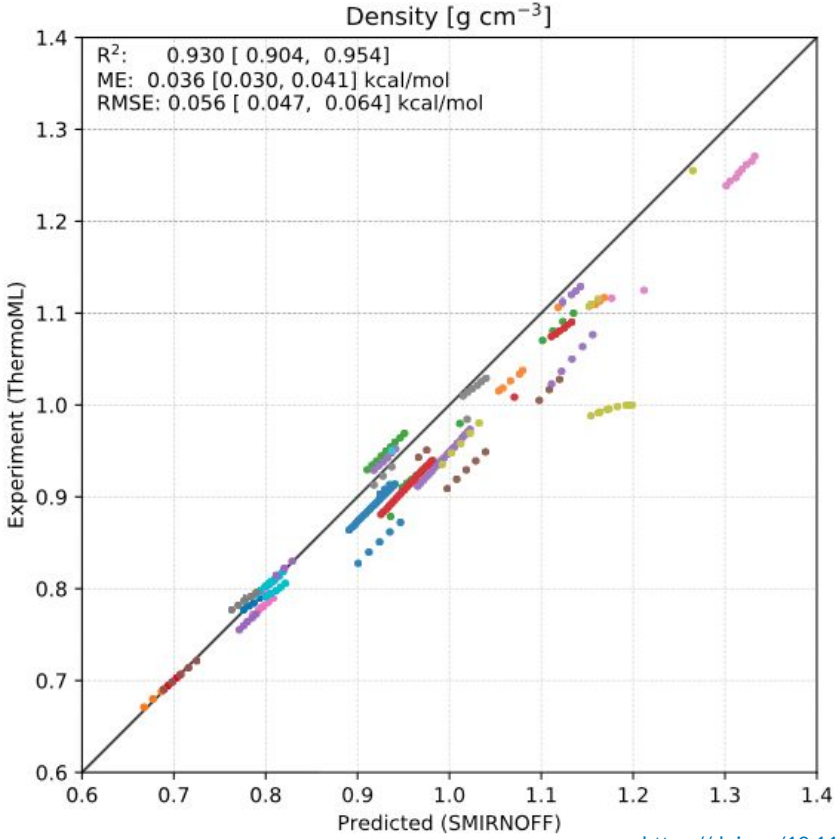
primarily valence terms
primarily Lennard-Jones
primarily electrostatics

Benchmarking against condensed-phase data can identify systematic issues

neat liquid densities from the **NIST ThermoML Archive**

https://trc.nist.gov/ThermoML.html

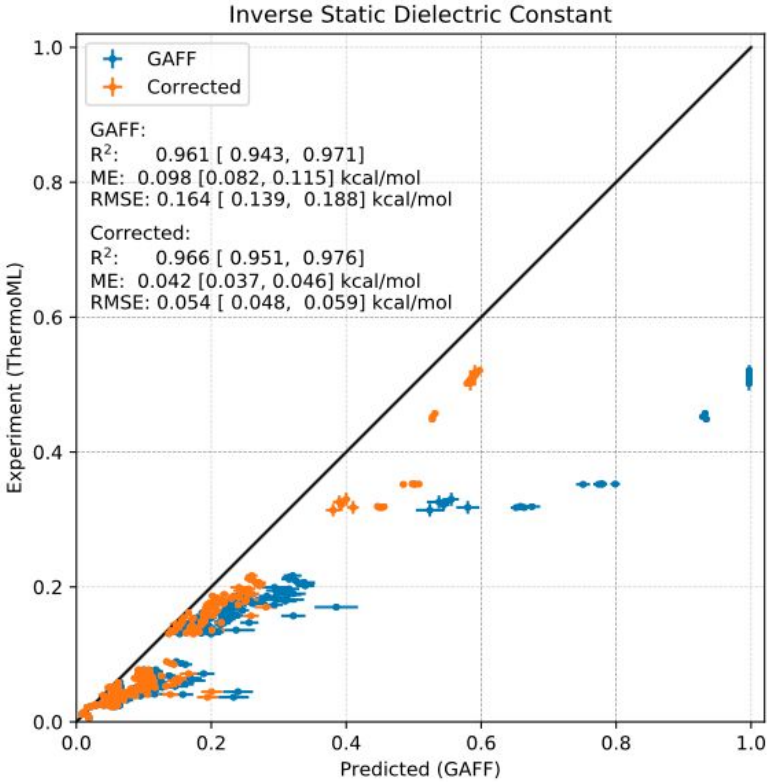


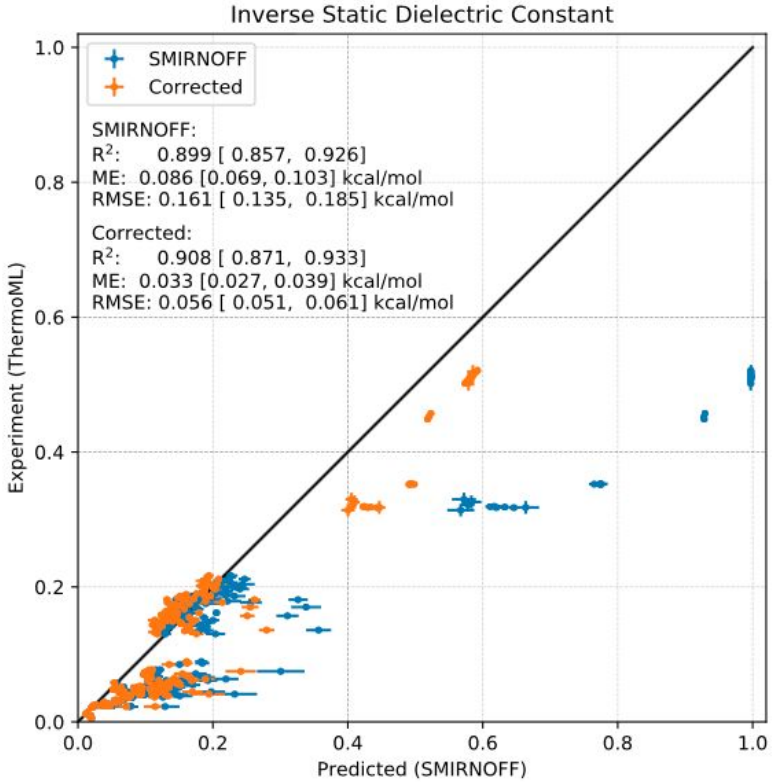


https://doi.org/10.1101/286542

Benchmarking against condensed-phase data can identify systematic issues

neat liquid static dielectric constants from the **NIST ThermoML Archive**
https://trc.nist.gov/ThermoML.html



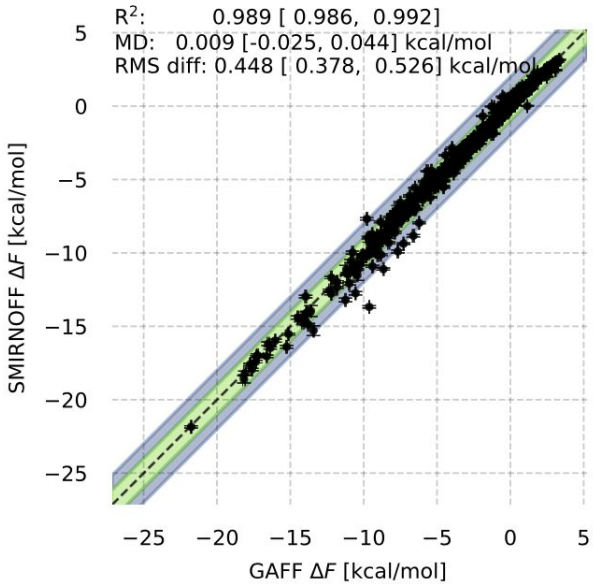


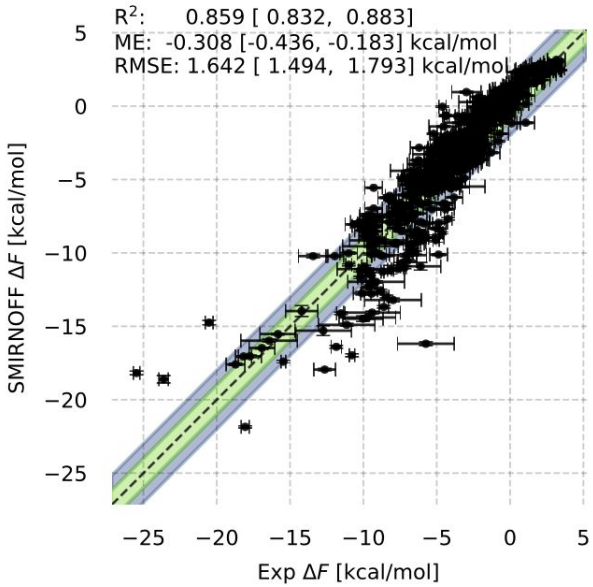
https://doi.org/10.1101/286542

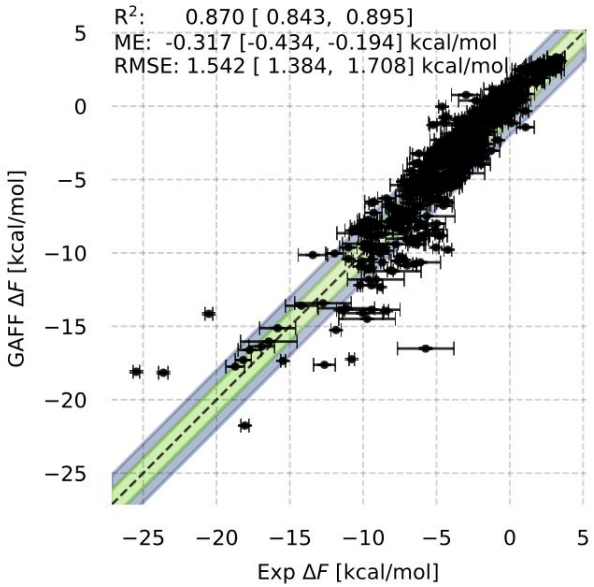
Benchmarking free energies can facilitate force field comparison

hydration free energies from the **FreeSolv 0.52**

https://github.com/MobleyLab/FreeSolv





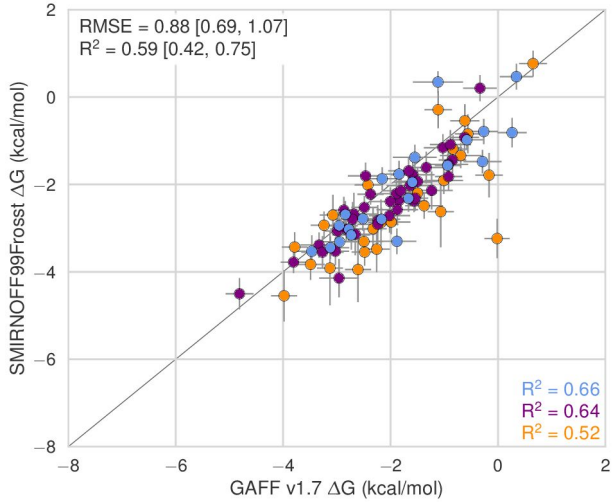


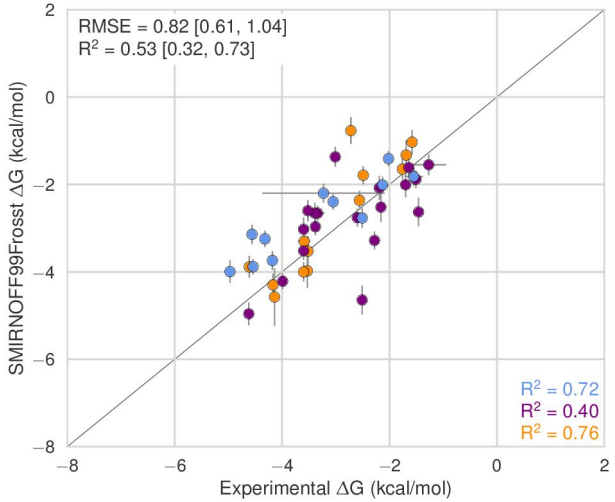
Computed using YANK 0.16.0

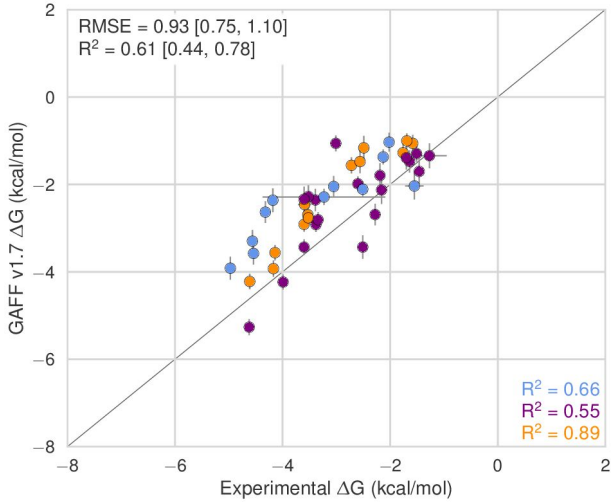
https://doi.org/10.1101/286542

Benchmarking host-guest binding free energies is feasible; could provide a lower bound on protein-ligand error

binding free energies of α-cyclodextrin and β-cyclodextrin with small molecule guests





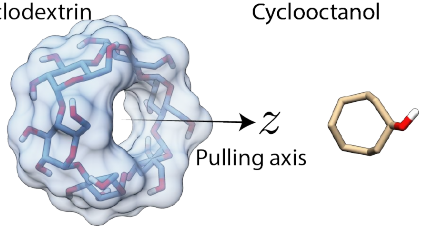


Points colored by guest functional group

Alcohol
Carboxylate
Ammonium

Host
Cyclodextrin

Guest
Cyclooctanol



Computed using pAPRika 0.0.4

Lennard-Jones parameters represent an opportunity for significant improvement

- OPLS v2 has not really changed, and OPLS v3 is not publicly available.
- GAFF2 has many changes and optimizations, but not documented.
- CHARMM GAMMP recently updated (2018) with neat liquid data
- GROMOS has a recent (chemically limited) small molecule update water and cyclohexane ΔG_{solv}
- In most cases, protein and DNA parameters inherit from small molecules (though some refitting)

In virtually all cases, only densities and heats of vaporization of PURE FLUIDS used. **No** mixture properties used.**

Philosophy: prioritizing high utility / low effort tasks informs order in which physical properties integrated

Generation 1

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

Generation 2

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

Our physical property estimation framework will support both automated benchmarking and parameterization

Our **design goals** focus on automation, modularity, and scalability:

- **plug-in architecture** for adding new physical properties
- **clear standards** documenting best practices for physical property computation
- newly added properties may be slow, and will first be used for **benchmarking**, then **parameterization** once their computation is fast enough to support it

Modularity of code and collaboration achieved via **simple, extensible APIs:**

- **Property calculation API**: parameters in, estimated properties out
- **Property plugin API:** define new physical properties and how to compute them
- **Property surrogate model API:** learn property response to parameters

Striving for simplicity in APIs and object models

```
# Define the input datasets from ThermoML
thermoml_keys = ['10.1016/j.jct.2005.03.012', ...]
dataset = ThermoMLDataset(thermoml_keys)
# Filter the dataset to include only molar heat capacities measured between 280-350 K
dataset.filter(ePropName='Excess molar enthalpy (molar enthalpy of mixing), kJ/mol')
dataset.filter(VariableType='eTemperature', min=280*unit.kelvin, max=350*kelvin)
# Load an initial parameter set
parameter_set = [ ForceField('smirnoff99frosst.offxml') ]
# Compute physical properties for these measurements
estimator = PropertyEstimator()
computed_properties = estimator.computeProperties(dataset, parameter_set)
# Write out statistics about errors in computed properties
for (computed, measured) in (computed_properties, dataset):
   print('%24s : experiment %s +- %s | calculated %s +- %s' % (measured.value,
measured.uncertainty, computed.value, computed.uncertainty)
```

https://github.com/openforcefield/openforcefield/blob/property-estimator/The-PropertyEstimator-format.md

Life cycle of a new physical property

Our **modular framework** will allow new properties to be implemented independently using a simple stable API, so that multiple researchers can work on different properties independently, and properties can be merged into benchmarking and parameterization sprints when ready.

1. Define an **object model** for representing data from source database (e.g. ThermoML)
2. **Curate** a subset of data of interest for use in initial testing and benchmarking
3. Develop a **forward simulation model** for computing the property from molecular simulations
4. Document **best practices for property computation** alongside the forward model
5. Develop an **error likelihood model** for the experimental measurements in terms of computed properties
6. Integrate the new property into the next **benchmarking** epoch
7. Optimize property calculation to enable inclusion in the next **parameterization** epoch
8. **Curate** successively larger datasets for improved parameterization accuracy

Roadmap for physical property integration

- All thermodynamics determined by G, which is explicitly a function of T, P, and composition.
  - If we have G(T,P,x) data, we will get all thermodynamics right.
- However, won't always tell us why we get the thermodynamics right
  - Static dielectrics involved dG/dE, should provide some information about electrostatic energy terms

- We also need to **balance:**
  - Availability of experimental data,
  - Information content of experimental data
  - Ease of producing or obtaining new experimental data
  - Simulation expense

- In some cases, we will want to generate new data for parts of chemical space without enough information

Roadmap for physical property integration

- **Low hanging fruit**
  - Molar volumes/density, static dielectrics
  - Temperature dependence of the above
- **Problematic, but might be useful at first**
  - Enthalpies of vaporization, hydration free energies / enthalpies
- **Nearly as low hanging**
  - Densities and dielectric for mixtures
  - Compressibility, thermal expansion, speed of sound
  - Partial molar volumes, heats of mixing, residual heat capacity
- **More computationally expensive, but probe intramolecular properties**
  - Partition coefficients
  - Infinite dilution activity coefficients in mixtures (basically, dG/dx)
  - Relative solubilities
- **More complex, but closer to what we want to predict**
  - Host-guest binding free energies with chemically-tuned hosts and guest
  - Ligand-binding free energies for calculations we are confident about converging?
- **Others**:
  - Small molecule crystals, liquid structure factors, others?

We will draw on (and develop) high-quality datasets

Draw on (and develop) **high-quality computer-readable datasets** with **experimental uncertainties** and **unambiguous definition of systems**:

- **NIST ThermoML Archive:** https://trc.nist.gov/ThermoML.html
- **BindingDB** subsets**:** http://bindingdb.org
- **FreeSolv** (for benchmarking only?)**:** https://github.com/MobleyLab/FreeSolv
- **Partition coefficients, relative solubilities,** and **activity coefficients** from other sources or experiment
  - Can industry provide some datasets that can be made public?
  - NIST can archive it.

Initial physical properties will focus on maximizing improvement at minimal computational cost

- **densities** of neat liquids from ThermoML Archive
- **static dielectric constants** of neat liquids from ThermoML Archive
- **hydration free energies** from FreeSolv
- **host-guest binding free energies** from BindingDB

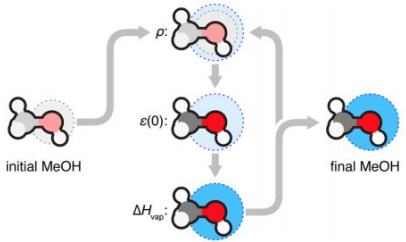
First epoch will integrate density and static dielectric properties for parameterization and benchmarking

ThermoML Archive contains dozens of **density** and **static dielectric** measurements of neat liquids (and mixtures) near ambient pressure and temperature

(from 2015) Filter step	Number of measurements remaining	
	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

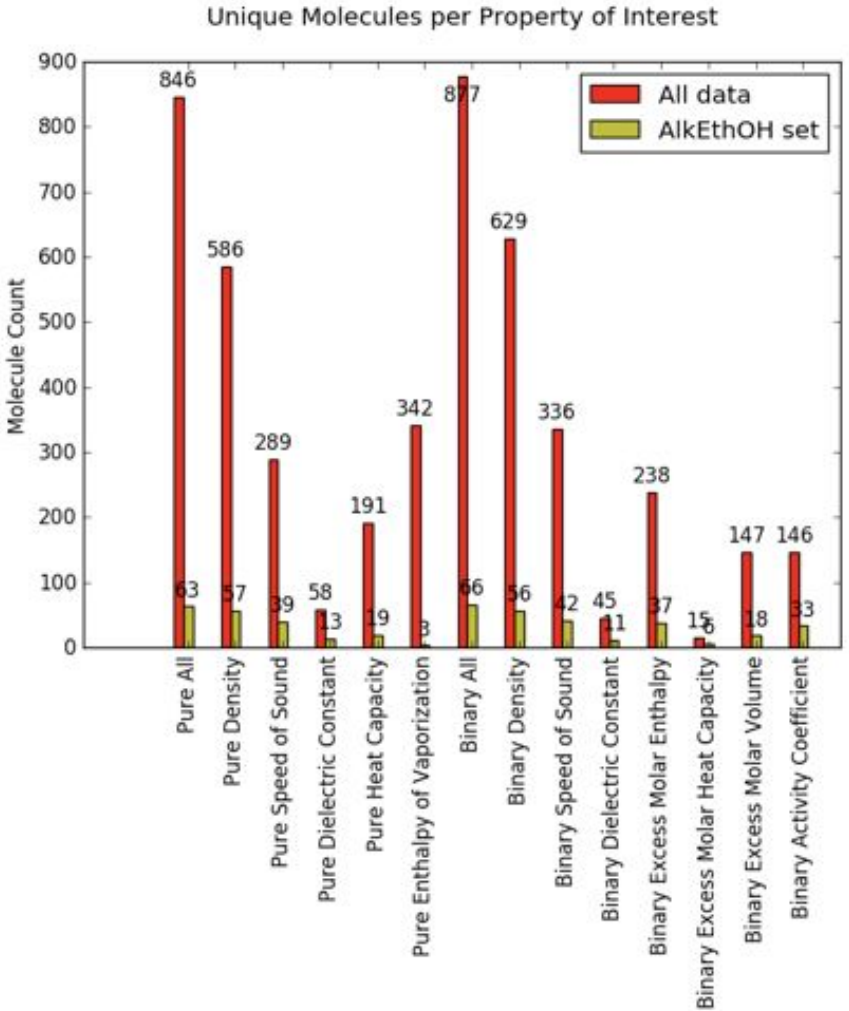
Incorporating minimal **neat liquid static dielectric constants** significantly improves physical properties like hydration free energies



http://doi.org/10.1021/jp411529h

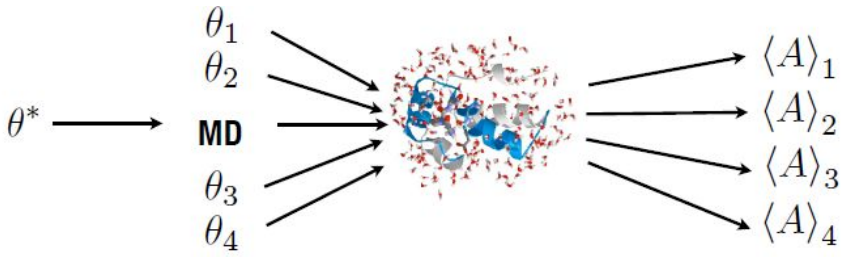
Liquid thermodynamic data

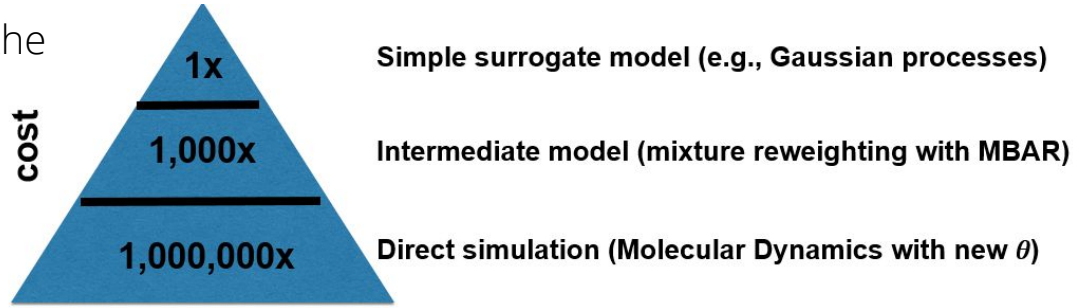
- NIST ThermoML Archive: https://trc.nist.gov/ThermoML.html
- Has lots of data (usually a lot of T, P, and composition data) at 1 atm and biological T.
- But chemical coverage is somewhat sparse
- Will eventually need more experimental data to cover chemical space
  - Cheap: densities, heats of mixing for binary compounds
- Working with NIST and Chodera lab for automating experiments, ideally driven by informatics coming from simulations



How can we use statistical evidence to choose non-bonded functionals and parameters?

- Bayesian inference provides a extensible way evaluate fitness of parameters and functional forms based on statistical evidence
- Problem: There are dozens to hundreds of non-bonded parameters in a force field.
- MCMC sampling of likelihoods VERY EXPENSIVE for observables requiring simulation.
- Even if not doing Bayesian, exploring the space can be expensive
- Problem: How do we make it sufficiently cheap for MCMC?
- **Answer: Multifidelity sampling**

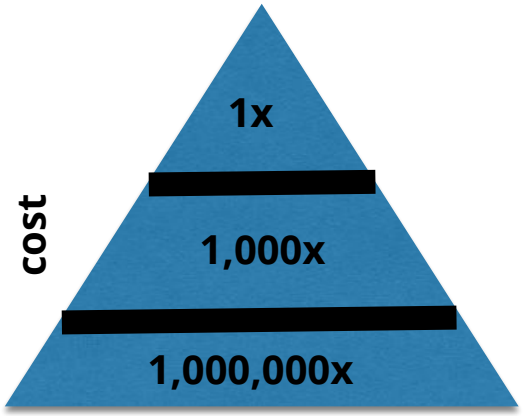




Computing physical properties is expensive

- Evaluating the least-squares error penalty or data likelihood for each trial parameter set θ requires computing either an expectation or free energy
- Both are 1) **expensive** and 2) **have statistical error**.

- We will need a hierarchy of schemes to evaluate likelihood (even multidimesional optimization)
- More expensive levels are only used as needed, such that inference cost is amortized.
- When converged, remaining evaluations will be carried out by fast models.



fast surrogate model (Gaussian processes, neural network)

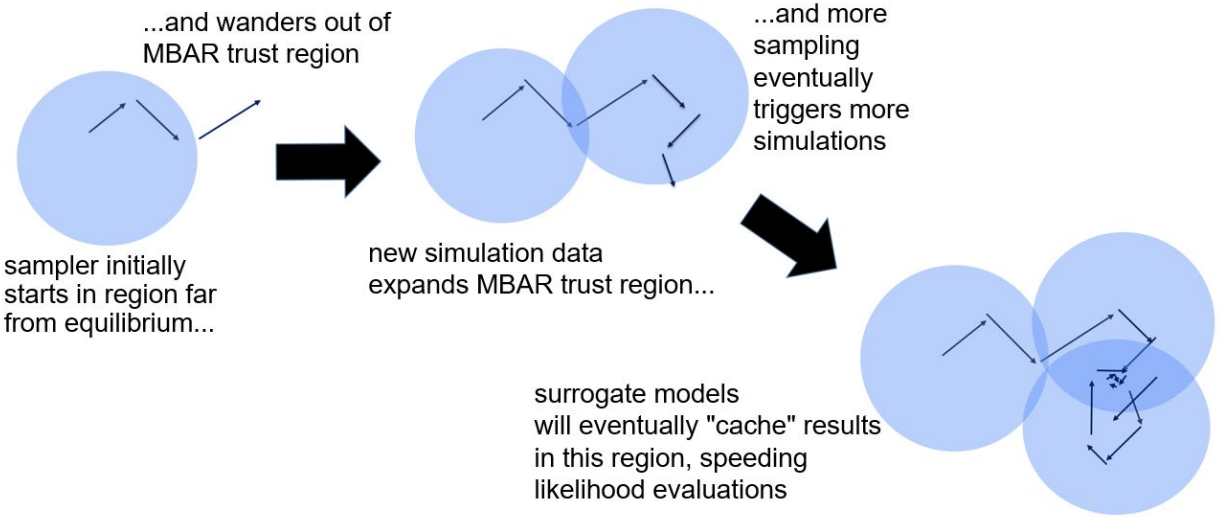
MBAR reweighting to estimate new properties at θ

Performing a simulation at new parameter set θ

Multifidelity sampling allows for cheap and accurate evaluation of Bayesian likelihoods

Three* levels of fidelity used to calculate likelihoods

1. MD simulation (most expensive)
2. Reweighting with MBAR (intermediate)
3. Surrogate models



Reweighting is cheap and can be made arbitrarily good

Reweighting to one state:

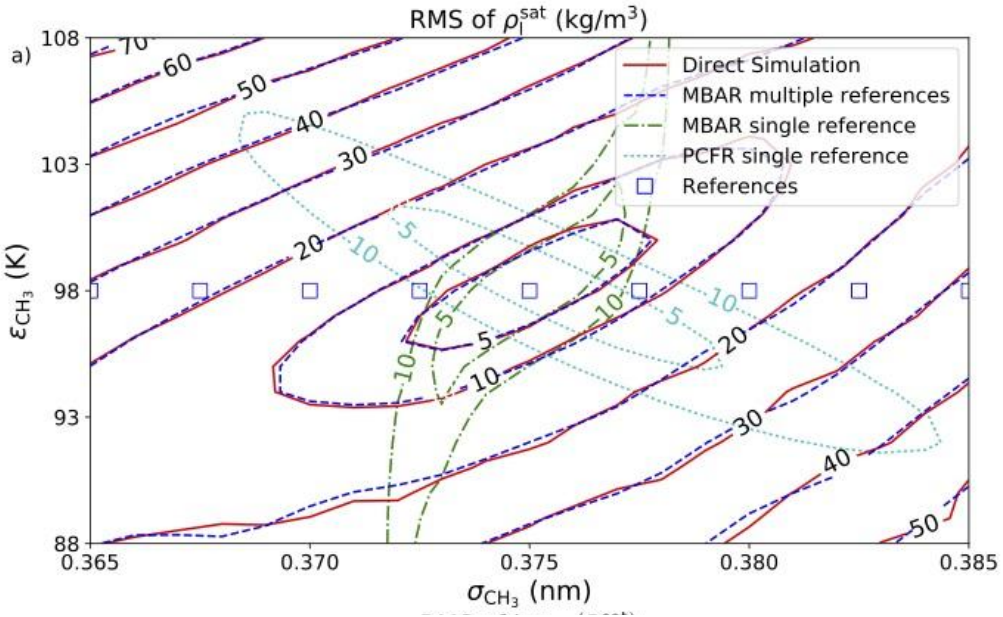
$$\langle A \rangle_i = \left\langle A(x) e^{\beta \Delta G_i - \beta \Delta U_i(x)} \right\rangle_j$$

Problem: poor convergence if ΔU varies too much (poor phase space overlap)

Solution: Reweighting to multiple different states

$$\langle A \rangle_i = \left\langle A(x) \frac{e^{\beta G_i - \beta U_i(x)}}{\sum_k e^{\beta G_k - \beta U_k(x)}} \right\rangle_{mixture}$$

Mixture distribution: throw all the simulations into the same pot



Example: using reweighting to calculate RMSDs of liquid saturation densities of ethane as a function of LJ σ and ε.

RA Messerly, SM Razavi, MR Shirts. *J. Chem. Theory Comput.* 14 (6), 3144-3162 (2018)

We can detect where reweighting fails

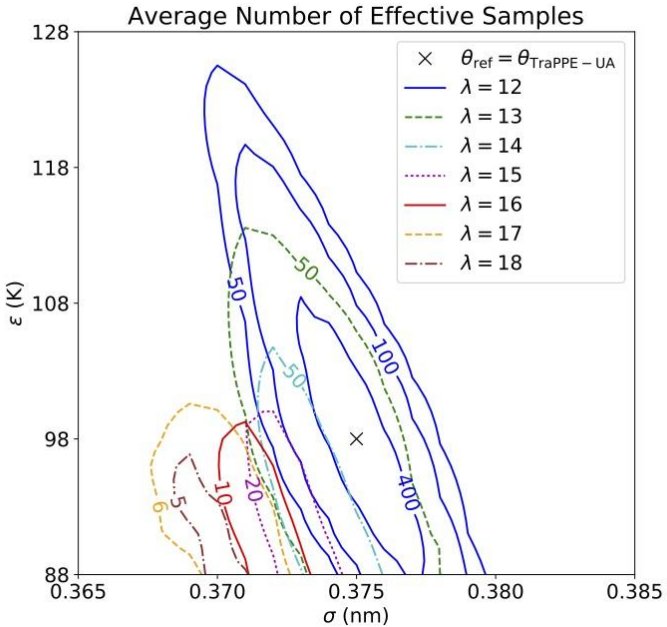
$$N_{eff} = \sum_i \frac{1}{w_i^2}$$

where

$$w_i = \frac{e^{\beta G_i - \beta U_i(x)}}{\sum_k e^{\beta G_k - \beta U_k(x)}}$$

Over a large range of predicted observables, If $N_{eff} > 50$, our error estimates are fairly reliable.

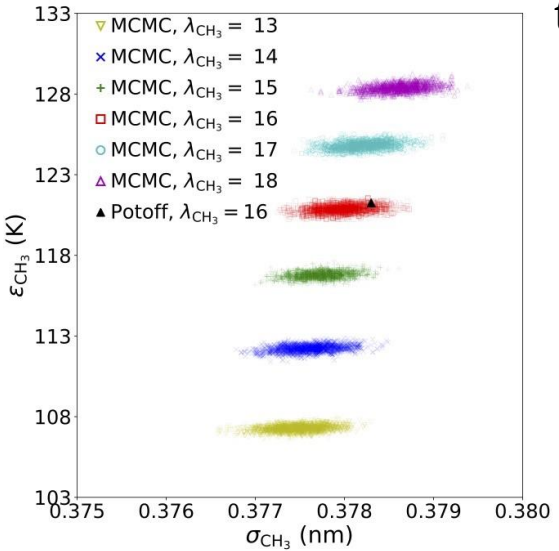
RA Messerly, SM Razavi, MR Shirts. *J. Chem. Theory Comput.* 14 (6), 3144-3162 (2018)



$$U(r) = C(\lambda)\epsilon\left[\left(\frac{\sigma}{r}\right)^{\lambda} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

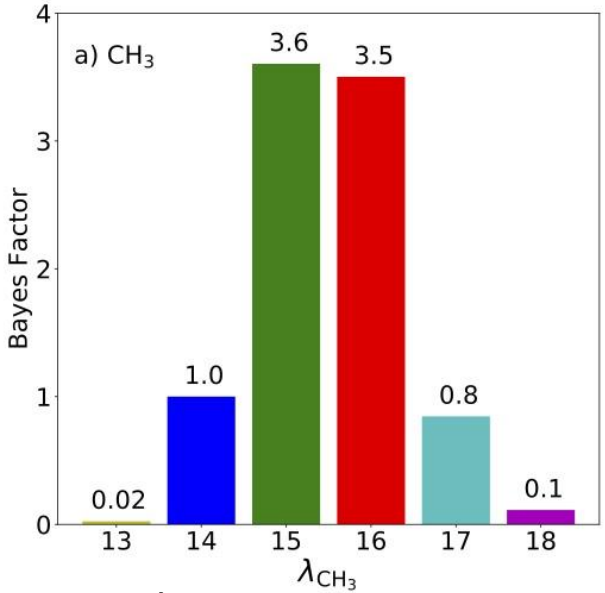
Example: Number effective samples while changing σ, ε, and λ from a single reference state

With fast models, we can use full machinery of Bayesian inference



Ethane using Mie potential fit to phase equilibrium data

$$U(r) = C(\lambda)\epsilon\left[\left(\frac{\sigma}{r}\right)^{\lambda} - \left(\frac{\sigma}{r}\right)^{6}\right]$$



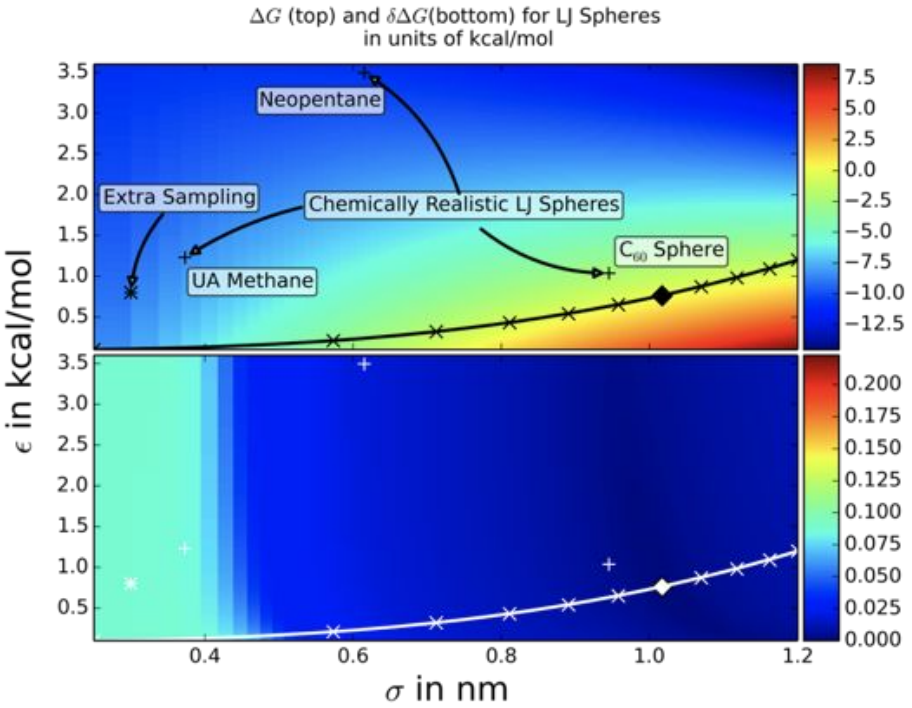
Use MCMC to sample posterior of probability of ethane models as a function of σ, ε, and (integer) λ.

From this, extract Bayes factors for parameters!

RA Messerly, RA Messerly, MR Shirts, AF Kazakov *J. Chem. Phys.* 149 (11), 114109 (2018)

Even cheaper; use a surrogate model or metamodel

Properties generally change smoothly with parameter

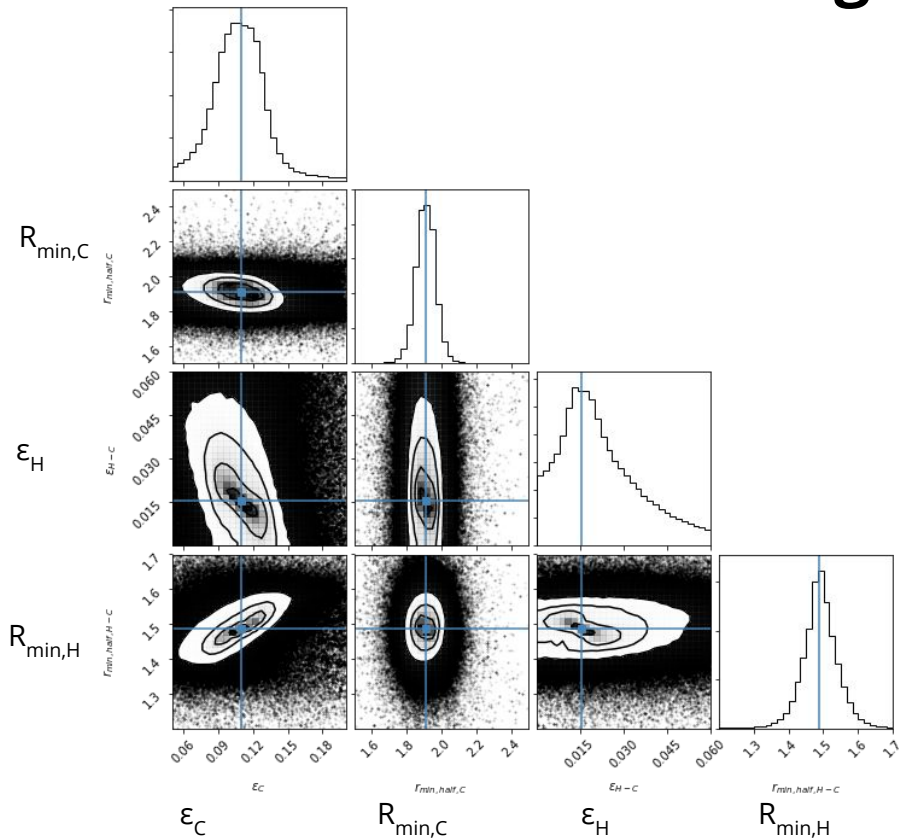


Replace the smooth function of property(parameter) from calculations by some multidimensional fitting within the range of uncertainty

Right now, using Gaussian processes, but are looking at other options for higher dimensions

- Polynomial chaos expansion
- Neural nets or other machine learning?
- Criteria: we have **smooth** functions with **good local data** through reweighting

We can perform sampling on higher dimensional surrogate models

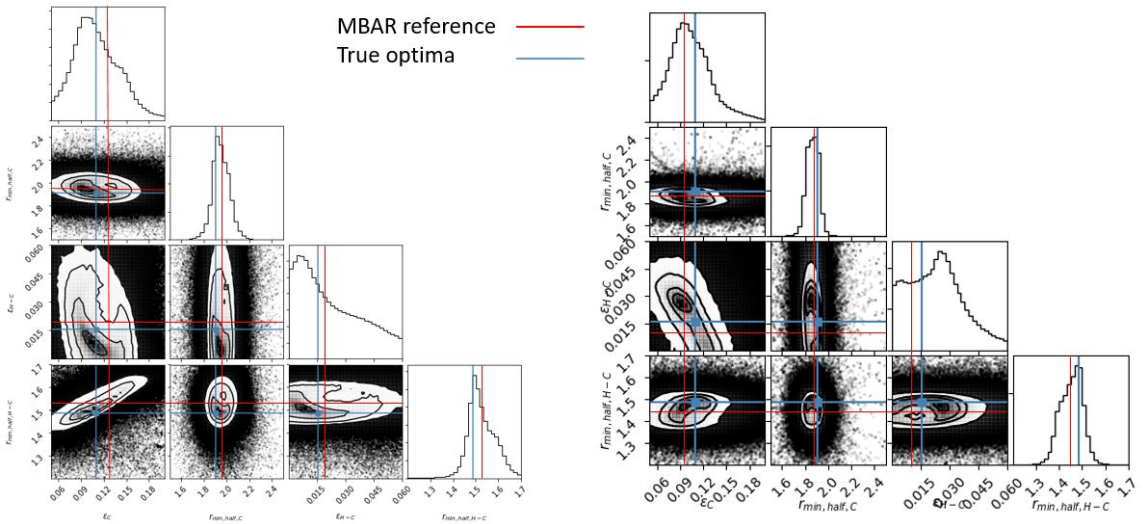


Fit MBAR results, including uncertainties, with a Gaussian process

Hard to represent 4 D surfaces visually!

Instead: shown are marginal distributions sampled from surrogate model determined from MBAR in the 4 cyclohexane J parameters

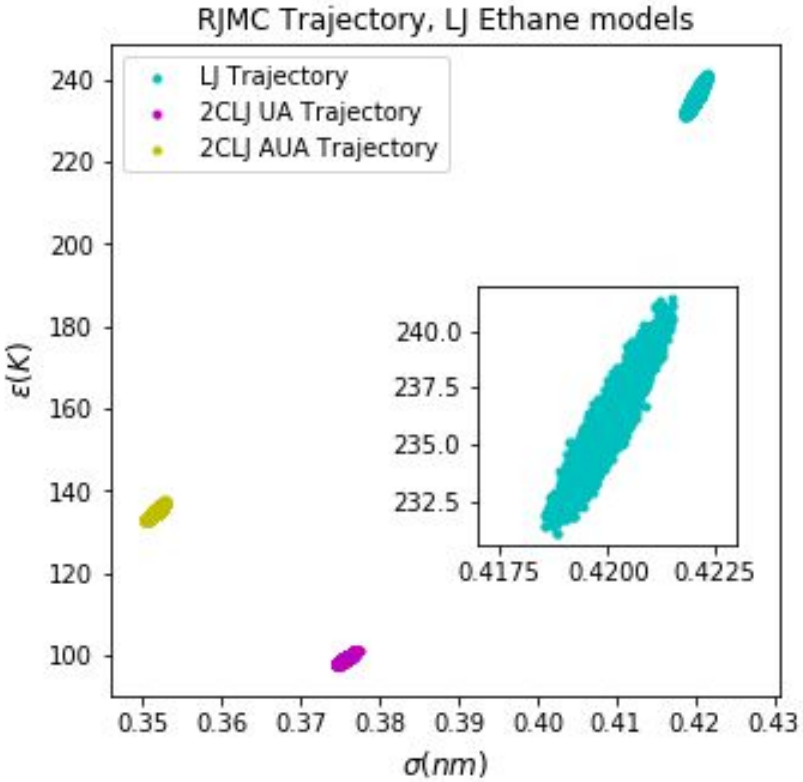
Multifidelity sampling allows for cheap and accurate evaluation of Bayesian likelihoods



Multifidelity sampling provides accurate enough model to sample distribution of L-J parameters of cyclohexane

Future directions and challenges

- **Second problem**: sampling between models with different numbers of parameters
- Splitting Lennard-Jones types
- Testing which functional or combination rule is optimal
- Reversible jump Monte Carlo extends traditional MCMC to moves across models
- Can sample across model space and determine which models are preferable based on sampling ratio



Problems and future directions

- **How does it scale with number of parameters?**
  - Gaussian process may only scale to 10-15 dimensions: investigating other options (NN, polynomial chaos)
  - Optimize in parameter “eigenspace” (i.e. learn how certain parameters are coupled in some way): Many sets of parameters have low correlations
  - Properties are generally smooth functions of parameters
- **Other parameters?**
  - Can use for BCCs, valence terms as well
- **Different functional forms?**
  - Use r*eversible jump MC (RJMC)* to build evidence for different forms
  - Choice of combining rules, functional form, types
- **AUTOMATE**
  - Will be building a PropertyCalculator API to make this easily swappable for property and easily automated

Benchmarks discussed from last time

- Each generation of force field will be evaluated using **benchmarks with increasing coverage**
- Benchmarks will be **publicly posted** along with each force field generation to track progress
- Benchmarks will be run with Property Calculator framework
- Some set of reserved testing data from training datasets
- Others:
  - Surface tension, self-diffusion coefficient and shear viscosity
  - Binding free energies (host:guest, benchmark protein:ligand sets that can be converged)
  - Small molecule crystal structures
  - Liquid structure factors (neutron or X-ray)
  - Phase change data (Melting?)
  - Lipid partitioning or surfactant partitioning?

Continue conversion in **#datasets** and **#benchmarks!**

Acknowledgements

GitHub:

- github.com/openforcefield/openforcefield

Slack:

- #propertycalculator
- #benchmarks