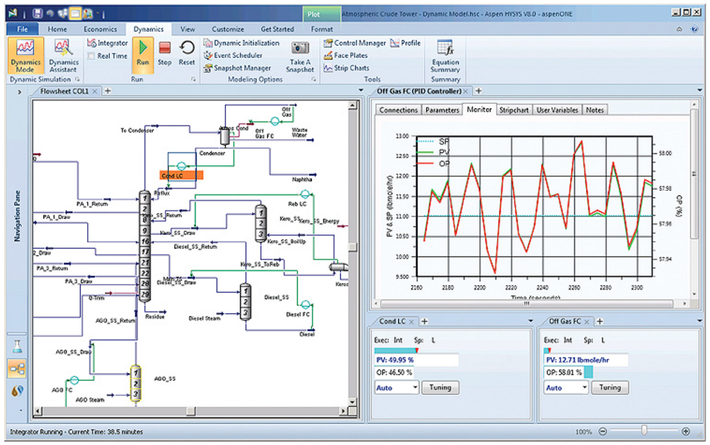
Towards best practices in calculating protein-ligand binding free energies

Michael R. Shirts
Department of Chemical Engineering
University of Virginia

2014 Workshop on Free Energy Methods in Drug Design
May 21, 2014

Computational modeling and design is central to engineering



Exec: Int Sp: L

PV: 49.95 %

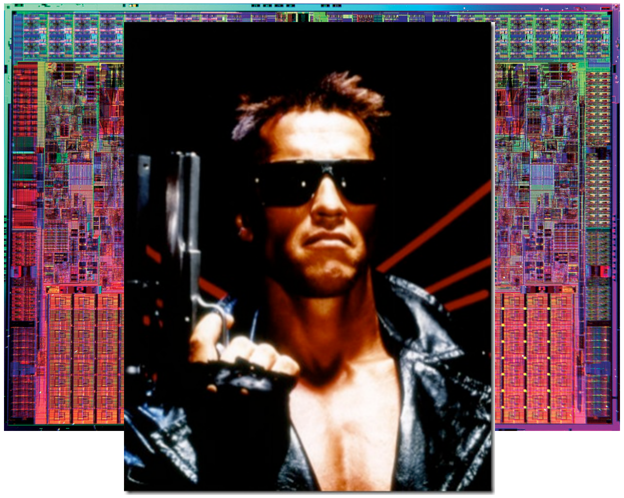
OP: 46.50 %

Exec: Int Sp: L

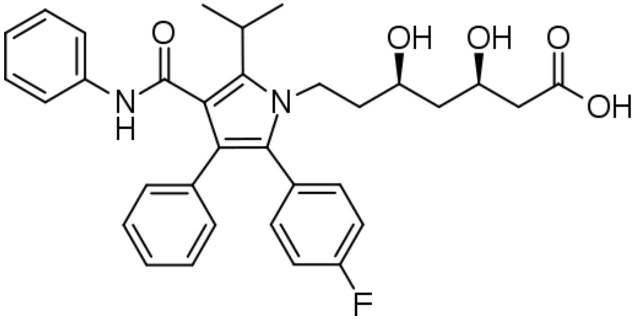
PV: 12.71 lbmole/hr

OP: 58.01 %

Integrator Running - Current Time: 38.5 minutes



Why can't we design drugs on a computer?

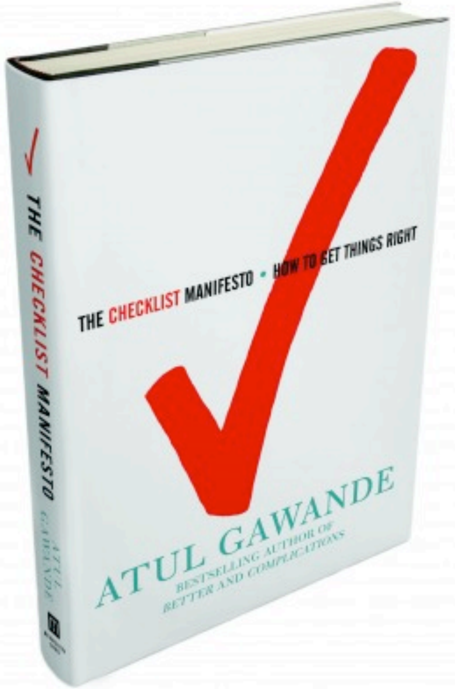


What is preventing free energy calculations from being more powerful and useful in drug design?

- Claim: not <u>primarily</u> lack of computational power anymore (at least for many of the calculations here)
- Other more relevant bottlenecks
  - Time new researchers take to learn methods
  - Sorting through the jungle of different methods to choose
  - Easily avoidably errors in running calculations
  - Lack of understanding how various methods affect free energies
  - Lack of common test systems to benchmark new methods
  - Time required to wrangle files in bunch of different formats
  - Lack of testing in code leading to errors only found later

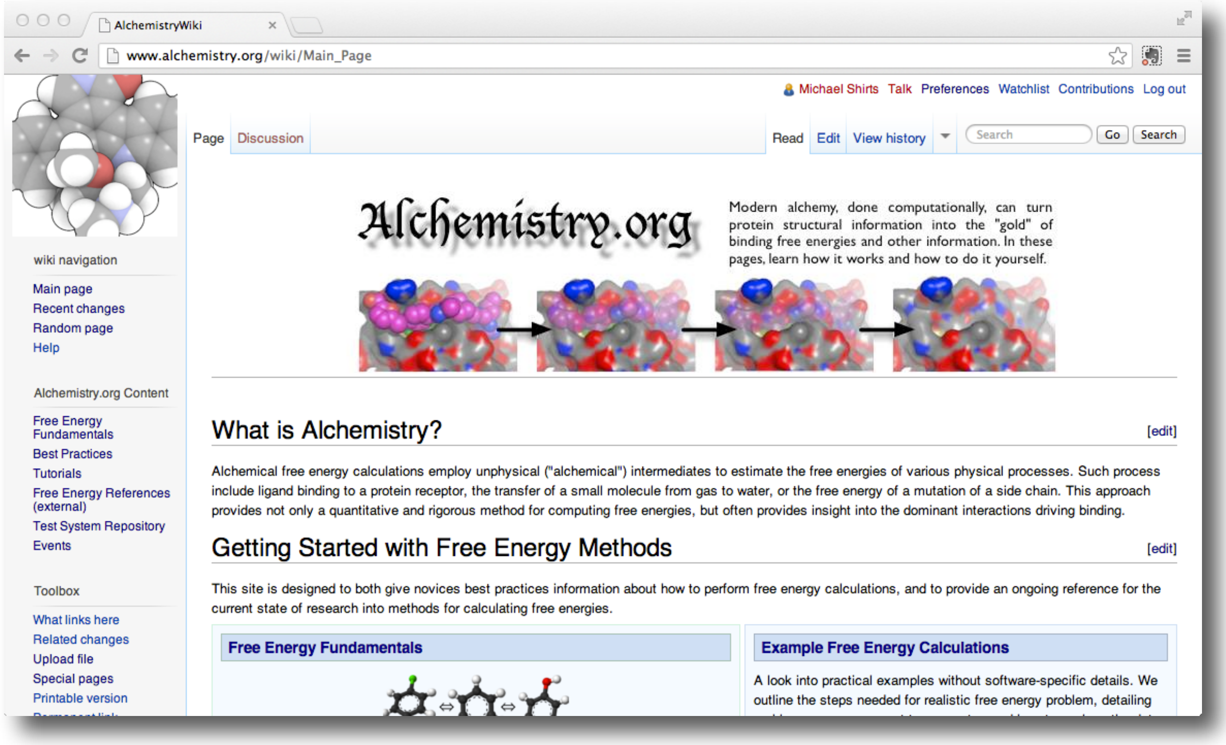
I haven’t got to force fields yet . . .

‘The Checklist Manifesto’



- Formalize informal knowledge for complex systems
- Read from file, don’t store in volatile memory
- Examples
  - Airline pilots
  - Guidelines for central line bloodstream infections
- What is the equivalent for molecular simulations?
- How do we create the culture?

Alchemistry.org: An experimental community site for learning about free energy calculations



Modern alchemy, done computationally, can turn protein structural information into the "gold" of binding free energies and other information. In these pages, learn how it works and how to do it yourself.

What is Alchemistry?

Alchemical free energy calculations employ unphysical ("alchemical") intermediates to estimate the free energies of various physical processes. Such process include ligand binding to a protein receptor, the transfer of a small molecule from gas to water, or the free energy of a mutation of a side chain. This approach provides not only a quantitative and rigorous method for computing free energies, but often provides insight into the dominant interactions driving binding.

Getting Started with Free Energy Methods

This site is designed to both give novices best practices information about how to perform free energy calculations, and to provide an ongoing reference for the current state of research into methods for calculating free energies.

Free Energy Fundamentals

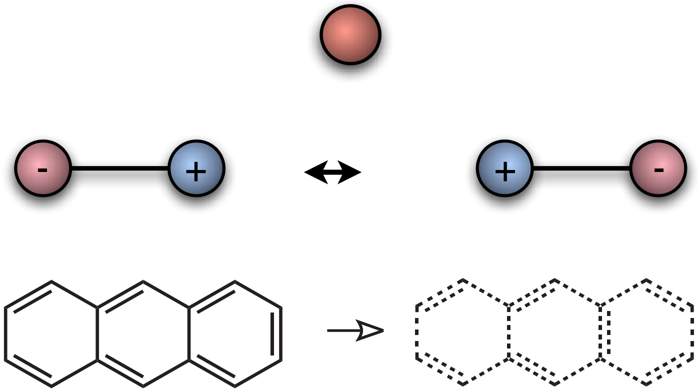
Example Free Energy Calculations

A look into practical examples without software-specific details. We outline the steps needed for realistic free energy problem, detailing

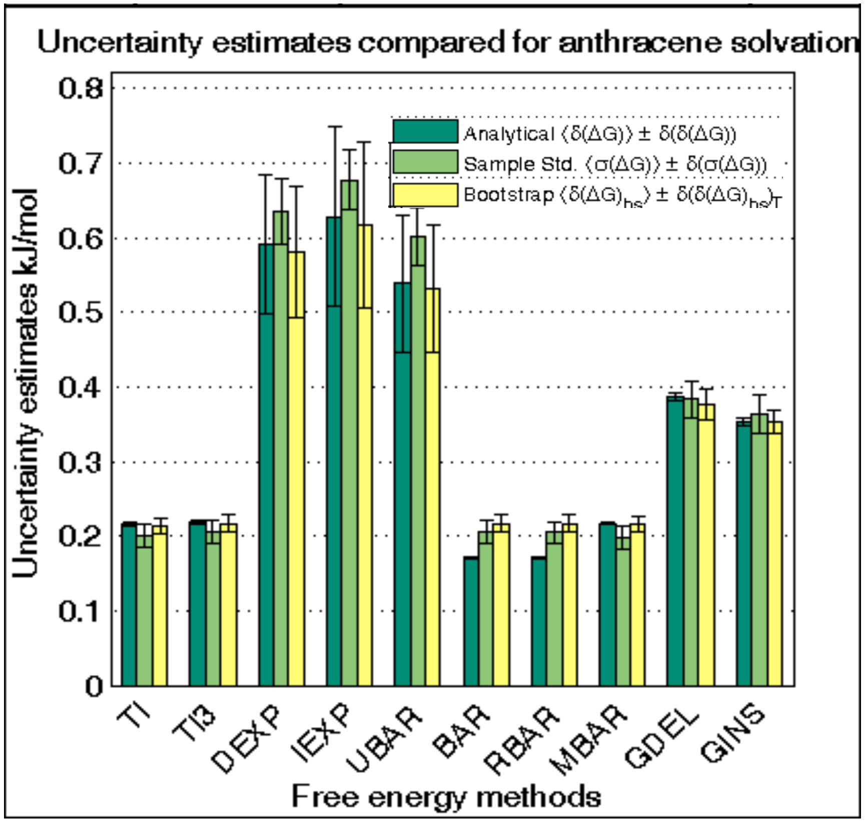
- A continually updated review of theory and best practices
- Versioned best practices and checklists
- A place to post tutorials
- A place to post benchmark files
- Annotatable database of citations

We have validated the statistical error estimates for free energy calculations

- Benchmark test set for free energy calculations
- Paliwal and Shirts, *J. Chem. Theory Comput*, 7, 4115 (2011)



- Repeat calculations 100 times
- Compare analytical uncertainties with actual sample variance
- 10 different free energy methods



We can use multistate reweighting to validate simulation parameters for free energies

Sets of K simulations with M parameter sets

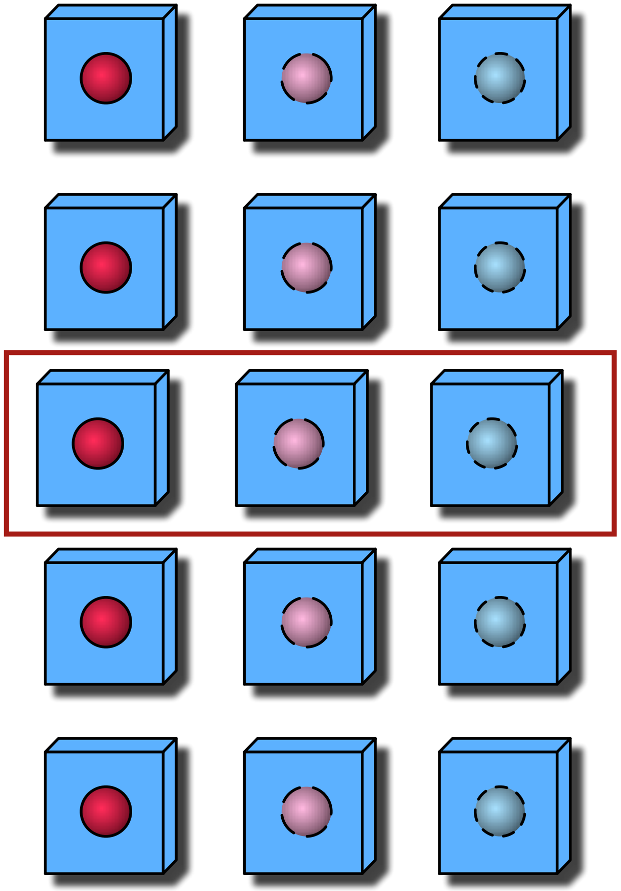
LJ_{cut}= 1.0 nm

LJ_{cut}= 0.95 nm

LJ_{cut}= 0.9 nm

LJ_{cut}= 0.85 nm

LJ_{cut}= 0.80 nm



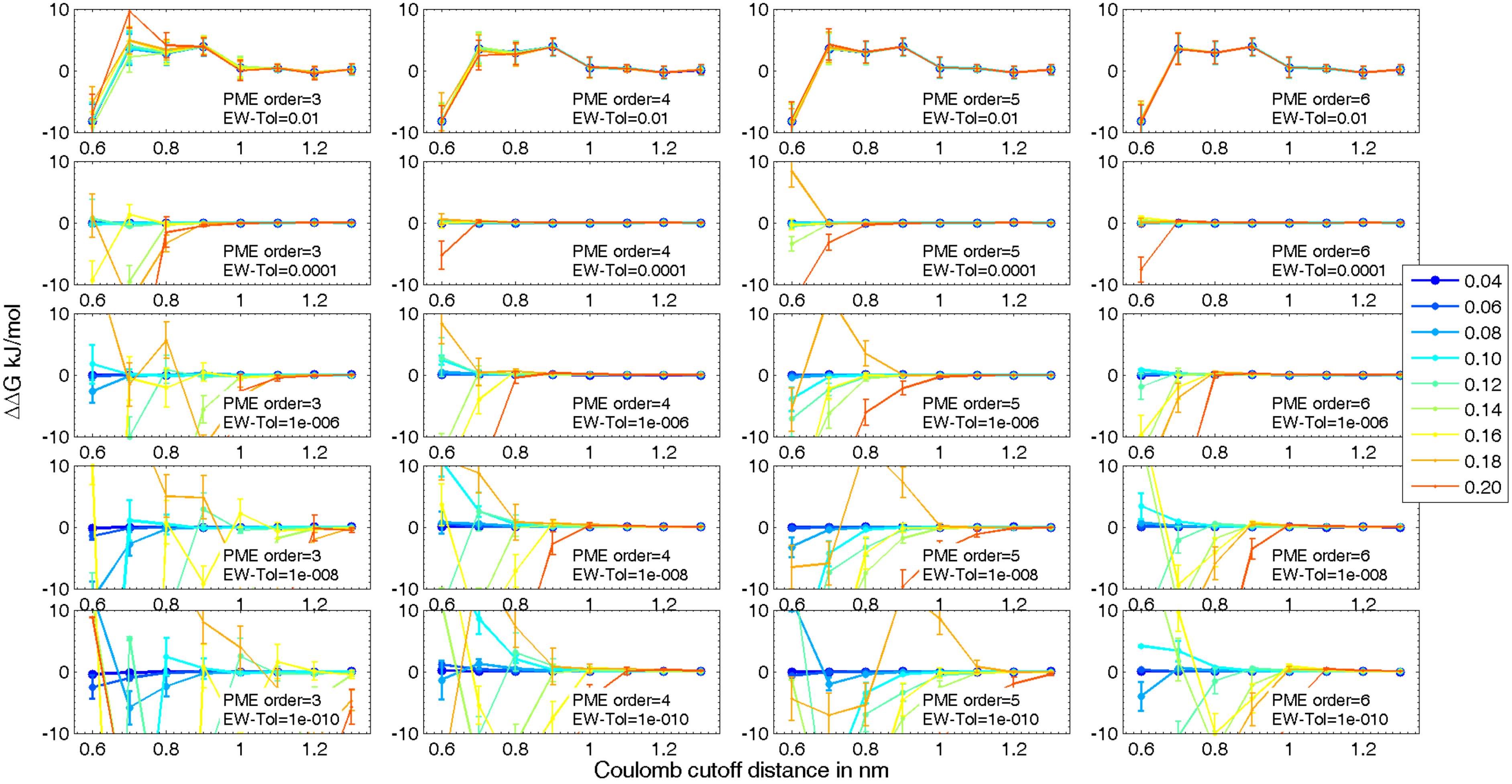
From energies of MK states evaluated at samples from K simulations

↓

Compute free energies of MK states

Takes about a minute to reevaluate one set of parameters
540 CPU years →1 CPU month

We can rapidly scan free energy differences as a function of parameters affecting free energies

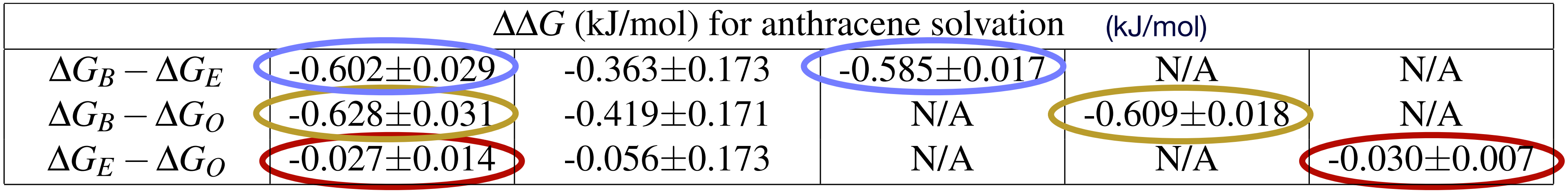


Reminder: Predicting $\Delta\Delta G_{E \rightarrow i}$ using only samples from a single initial set of parameters

H. Paliwal and M. R. Shirts, *J. Chem. Theory Comput.*, 9 (11), 4700–4717 (2013)

We identify simulation parameter choices with negligible difference from full energies

ΔG_B = Benchmark set
ΔG_E = Expensive set
ΔG_O = Optimized set



$\Delta\Delta G$ (kJ/mol) for anthracene solvation (kJ/mol)					
$\Delta G_B - \Delta G_E$	-0.602±0.029	-0.363±0.173	-0.585±0.017	N/A	N/A
$\Delta G_B - \Delta G_O$	-0.628±0.031	-0.419±0.171	N/A	-0.609±0.018	N/A
$\Delta G_E - \Delta G_O$	-0.027±0.014	-0.056±0.173	N/A	N/A	-0.030±0.007

Predictions using only benchmark set

Direct differences in free energy

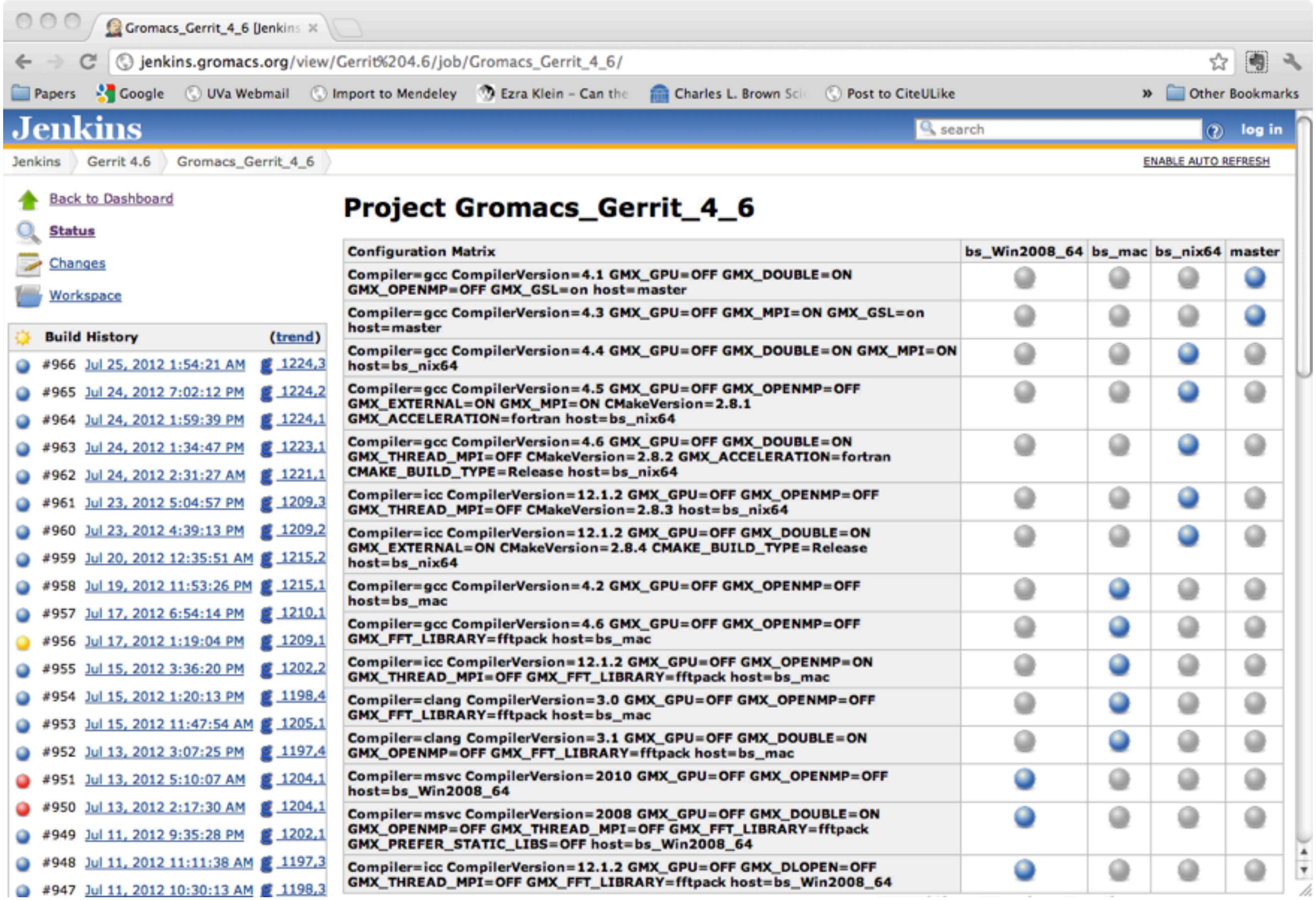
Tested results using MBAR on both parameter sets

Key: $\mathrm{var}(\Delta G_2\text{-}\Delta G_1) = \mathrm{var}(\Delta G_1) + \mathrm{var}(\Delta G_1) - \mathrm{cov}(\Delta G_1, \Delta G_2)$

What sort of code validation can be done? Code validation for molecular simulation

- How can we minimize the introduction of coding errors?
  - Regression tests
  - Automated builds and verification
  - Unit tests
- Example: New GROMACS strategy
  - Automated regression tests and builds
  - All code reviewed and signed off on by multiple developers
  - Code review tied to bug reports

What sort of code validation can be done? Code validation for molecular simulation



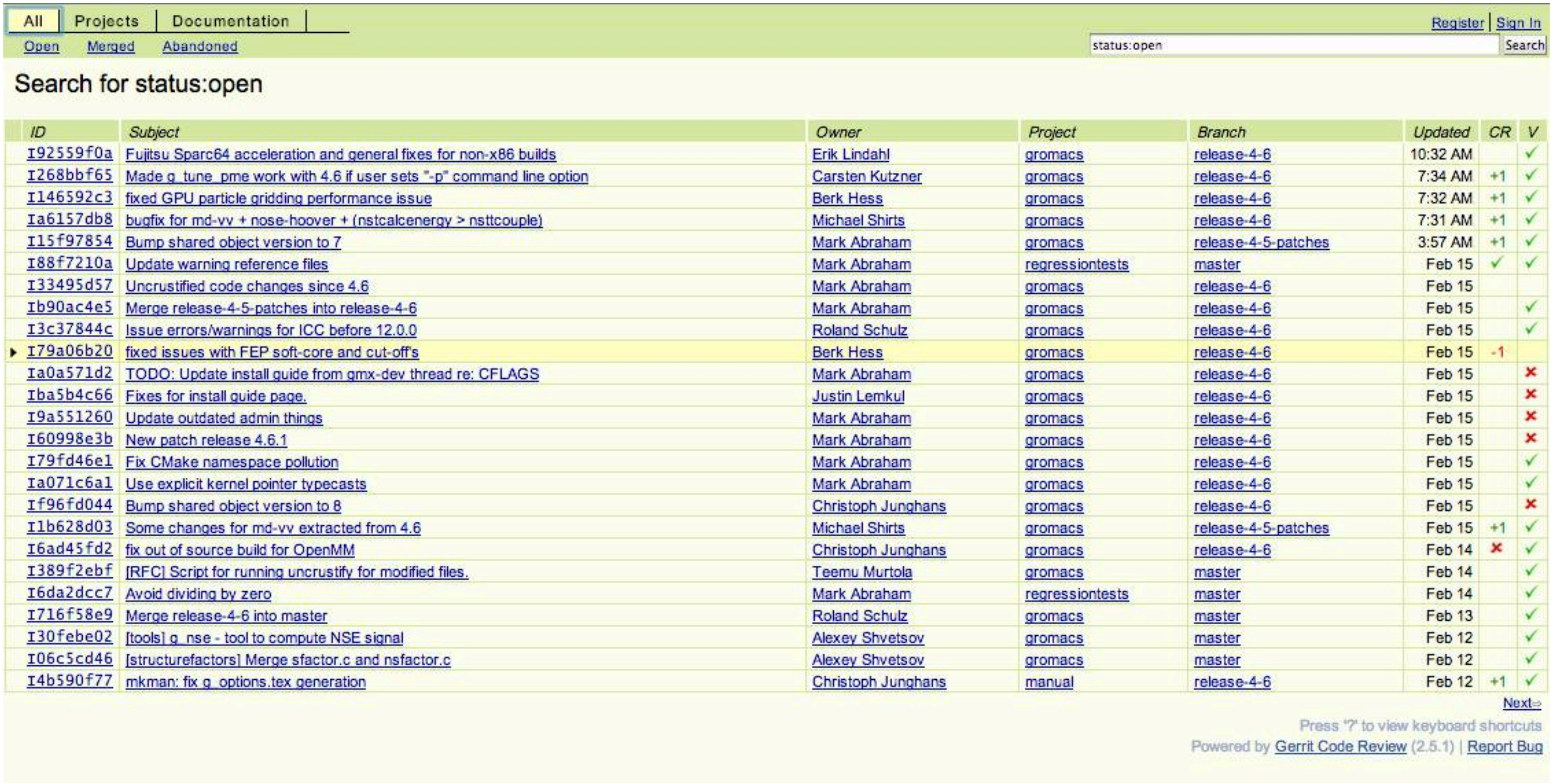
Build History (trend)

🔵	#966	Jul 25, 2012 1:54:21 AM	1224,3
🔵	#965	Jul 24, 2012 7:02:12 PM	1224,2
🔵	#964	Jul 24, 2012 1:59:39 PM	1224,1
🔵	#963	Jul 24, 2012 1:34:47 PM	1223,1
🔵	#962	Jul 24, 2012 2:31:27 AM	1221,1
🔵	#961	Jul 23, 2012 5:04:57 PM	1209,3
🔵	#960	Jul 23, 2012 4:39:13 PM	1209,2
🔵	#959	Jul 20, 2012 12:35:51 AM	1215,2
🔵	#958	Jul 19, 2012 11:53:26 PM	1215,1
🔵	#957	Jul 17, 2012 6:54:14 PM	1210,1
🟡	#956	Jul 17, 2012 1:19:04 PM	1209,1
🔵	#955	Jul 15, 2012 3:36:20 PM	1202,2
🔵	#954	Jul 15, 2012 1:20:13 PM	1198,4
🔵	#953	Jul 15, 2012 11:47:54 AM	1205,1
🔵	#952	Jul 13, 2012 3:07:25 PM	1197,4
🔴	#951	Jul 13, 2012 5:10:07 AM	1204,1
🔴	#950	Jul 13, 2012 2:17:30 AM	1204,1
🔵	#949	Jul 11, 2012 9:35:28 PM	1202,1
🔵	#948	Jul 11, 2012 11:11:38 AM	1197,3
🔵	#947	Jul 11, 2012 10:30:13 AM	1198,3

Project Gromacs_Gerrit_4_6

Configuration Matrix	bs_Win2008_64	bs_mac	bs_nix64	master
Compiler=gcc CompilerVersion=4.1 GMX_GPU=OFF GMX_DOUBLE=ON GMX_OPENMP=OFF GMX_GSL=on host=master	⚪	⚪	⚪	🔵
Compiler=gcc CompilerVersion=4.3 GMX_GPU=OFF GMX_MPI=ON GMX_GSL=on host=master	⚪	⚪	⚪	🔵
Compiler=gcc CompilerVersion=4.4 GMX_GPU=OFF GMX_DOUBLE=ON GMX_MPI=ON host=bs_nix64	⚪	⚪	🔵	⚪
Compiler=gcc CompilerVersion=4.5 GMX_GPU=OFF GMX_OPENMP=OFF GMX_EXTERNAL=ON GMX_MPI=ON CMakeVersion=2.8.1 GMX_ACCELERATION=fortran host=bs_nix64	⚪	⚪	🔵	⚪
Compiler=gcc CompilerVersion=4.6 GMX_GPU=OFF GMX_DOUBLE=ON GMX_THREAD_MPI=OFF CMakeVersion=2.8.2 GMX_ACCELERATION=fortran CMAKE_BUILD_TYPE=Release host=bs_nix64	⚪	⚪	🔵	⚪
Compiler=icc CompilerVersion=12.1.2 GMX_GPU=OFF GMX_OPENMP=OFF GMX_THREAD_MPI=OFF CMakeVersion=2.8.3 host=bs_nix64	⚪	⚪	🔵	⚪
Compiler=icc CompilerVersion=12.1.2 GMX_GPU=OFF GMX_DOUBLE=ON GMX_EXTERNAL=ON CMakeVersion=2.8.4 CMAKE_BUILD_TYPE=Release host=bs_nix64	⚪	⚪	🔵	⚪
Compiler=gcc CompilerVersion=4.2 GMX_GPU=OFF GMX_OPENMP=OFF host=bs_mac	⚪	🔵	⚪	⚪
Compiler=gcc CompilerVersion=4.6 GMX_GPU=OFF GMX_OPENMP=OFF GMX_FFT_LIBRARY=fftpack host=bs_mac	⚪	🔵	⚪	⚪
Compiler=icc CompilerVersion=12.1.2 GMX_GPU=OFF GMX_OPENMP=ON GMX_THREAD_MPI=OFF GMX_FFT_LIBRARY=fftpack host=bs_mac	⚪	🔵	⚪	⚪
Compiler=clang CompilerVersion=3.0 GMX_GPU=OFF GMX_OPENMP=OFF GMX_FFT_LIBRARY=fftpack host=bs_mac	⚪	🔵	⚪	⚪
Compiler=clang CompilerVersion=3.1 GMX_GPU=OFF GMX_DOUBLE=ON GMX_OPENMP=OFF GMX_FFT_LIBRARY=fftpack host=bs_mac	⚪	🔵	⚪	⚪
Compiler=msvc CompilerVersion=2010 GMX_GPU=OFF GMX_OPENMP=OFF host=bs_Win2008_64	🔵	⚪	⚪	⚪
Compiler=msvc CompilerVersion=2008 GMX_GPU=OFF GMX_DOUBLE=ON GMX_OPENMP=OFF GMX_THREAD_MPI=OFF GMX_FFT_LIBRARY=fftpack GMX_PREFER_STATIC_LIBS=OFF host=bs_Win2008_64	🔵	⚪	⚪	⚪
Compiler=icc CompilerVersion=12.1.2 GMX_GPU=OFF GMX_DLOPEN=OFF GMX_THREAD_MPI=OFF GMX_FFT_LIBRARY=fftpack host=bs_Win2008_64	🔵	⚪	⚪	⚪

What sort of code validation can be done?
Code validation for molecular simulation



status:open

Search for status:open

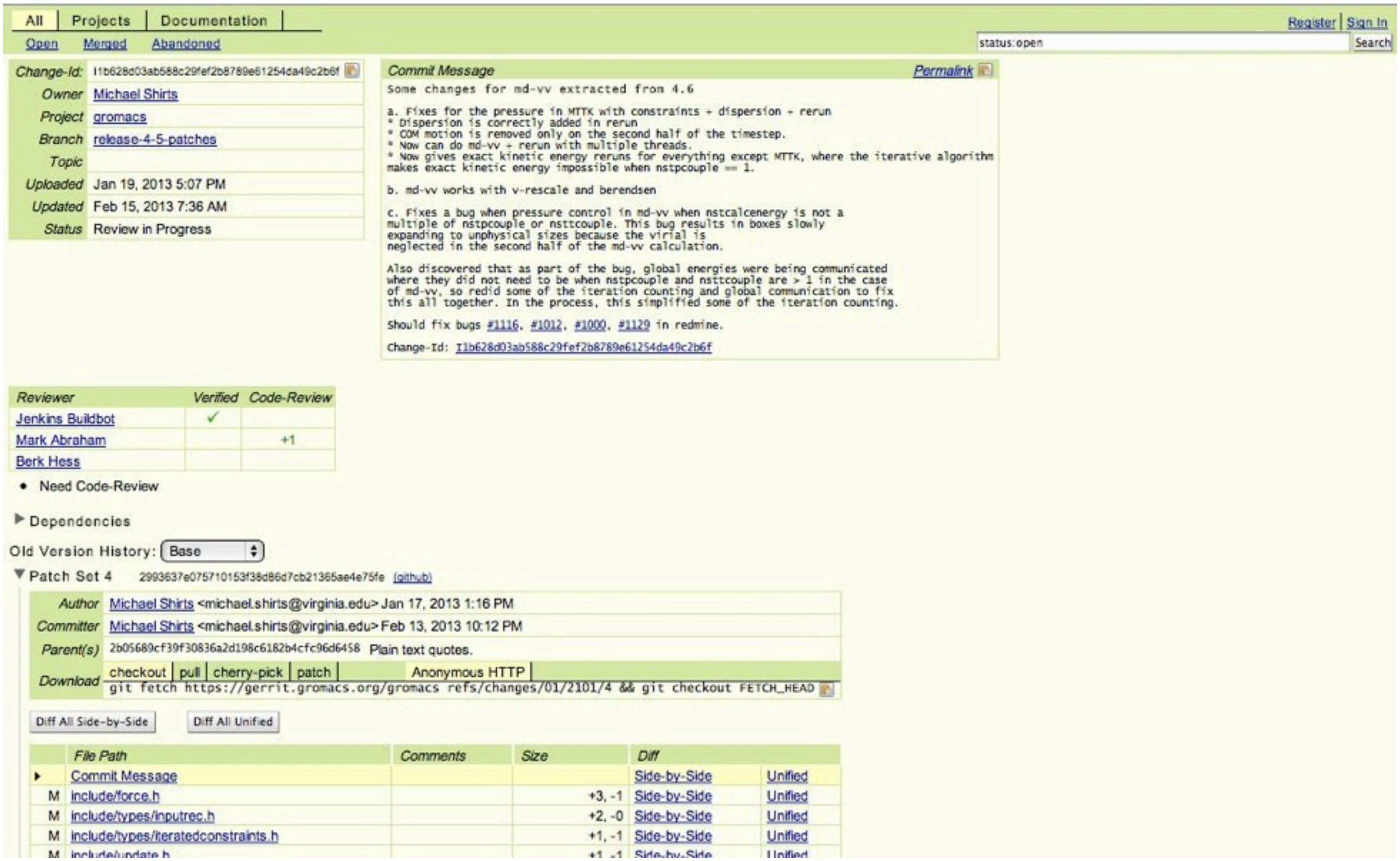
ID	Subject	Owner	Project	Branch	Updated	CR	V
I92559f0a	Fujitsu Sparc64 acceleration and general fixes for non-x86 builds	Erik Lindahl	gromacs	release-4-6	10:32 AM		✓
I268bbf65	Made g_tune_pme work with 4.6 if user sets "-p" command line option	Carsten Kutzner	gromacs	release-4-6	7:34 AM	+1	✓
I146592c3	fixed GPU particle gridding performance issue	Berk Hess	gromacs	release-4-6	7:32 AM	+1	✓
Ia6157db8	bugfix for md-vv + nose-hoover + (nstcalcenergy > nsttcouple)	Michael Shirts	gromacs	release-4-6	7:31 AM	+1	✓
I15f97854	Bump shared object version to 7	Mark Abraham	gromacs	release-4-5-patches	3:57 AM	+1	✓
I88f7210a	Update warning reference files	Mark Abraham	regressiontests	master	Feb 15	✓	✓
I33495d57	Uncrustified code changes since 4.6	Mark Abraham	gromacs	release-4-6	Feb 15		
Ib90ac4e5	Merge release-4-5-patches into release-4-6	Mark Abraham	gromacs	release-4-6	Feb 15		✓
I3c37844c	Issue errors/warnings for ICC before 12.0.0	Roland Schulz	gromacs	release-4-6	Feb 15		✓
▸ I79a06b20	fixed issues with FEP soft-core and cut-off's	Berk Hess	gromacs	release-4-6	Feb 15	-1	
Ia0a571d2	TODO: Update install guide from gmx-dev thread re: CFLAGS	Mark Abraham	gromacs	release-4-6	Feb 15		✗
Iba5b4c66	Fixes for install guide page.	Justin Lemkul	gromacs	release-4-6	Feb 15		✗
I9a551260	Update outdated admin things	Mark Abraham	gromacs	release-4-6	Feb 15		✗
I60998e3b	New patch release 4.6.1	Mark Abraham	gromacs	release-4-6	Feb 15		✗
I79fd46e1	Fix CMake namespace pollution	Mark Abraham	gromacs	release-4-6	Feb 15		✓
Ia071c6a1	Use explicit kernel pointer typecasts	Mark Abraham	gromacs	release-4-6	Feb 15		✓
If96fd044	Bump shared object version to 8	Christoph Junghans	gromacs	release-4-6	Feb 15		✗
I1b628d03	Some changes for md-vv extracted from 4.6	Michael Shirts	gromacs	release-4-5-patches	Feb 15	+1	✓
I6ad45fd2	fix out of source build for OpenMM	Christoph Junghans	gromacs	release-4-6	Feb 14	✗	✓
I389f2ebf	[RFC] Script for running uncrustify for modified files.	Teemu Murtola	gromacs	master	Feb 14		✓
I6da2dcc7	Avoid dividing by zero	Mark Abraham	regressiontests	master	Feb 14		✓
I716f58e9	Merge release-4-6 into master	Roland Schulz	gromacs	master	Feb 13		✓
I30febe02	[tools] g_nse - tool to compute NSE signal	Alexey Shvetsov	gromacs	master	Feb 12		✓
I06c5cd46	[structurefactors] Merge sfactor.c and nsfactor.c	Alexey Shvetsov	gromacs	master	Feb 12		✓
I4b590f77	mkman: fix g_options.tex generation	Christoph Junghans	manual	release-4-6	Feb 12	+1	✓

Next⇨

Press '?' to view keyboard shortcuts

Powered by Gerrit Code Review (2.5.1) | Report Bug

What sort of code validation can be done?
Code validation for molecular simulation



Change-Id:	I1b628d03ab588c29fef2b8789e61254da49c2b6f
Owner	Michael Shirts
Project	gromacs
Branch	release-4-5-patches
Topic	
Uploaded	Jan 19, 2013 5:07 PM
Updated	Feb 15, 2013 7:36 AM
Status	Review in Progress

Commit Message

```
Some changes for md-vv extracted from 4.6

a. Fixes for the pressure in MTTK with constraints + dispersion + rerun
* Dispersion is correctly added in rerun
* COM motion is removed only on the second half of the timestep.
* Now can do md-vv + rerun with multiple threads.
* Now gives exact kinetic energy reruns for everything except MTTK, where the iterative algorithm
makes exact kinetic energy impossible when nstpcouple == 1.

b. md-vv works with v-rescale and berendsen

c. Fixes a bug when pressure control in md-vv when nstcalcenergy is not a
multiple of nstpcouple or nsttcouple. This bug results in boxes slowly
expanding to unphysical sizes because the virial is
neglected in the second half of the md-vv calculation.

Also discovered that as part of the bug, global energies were being communicated
where they did not need to be when nstpcouple and nsttcouple are > 1 in the case
of md-vv, so redid some of the iteration counting and global communication to fix
this all together. In the process, this simplified some of the iteration counting.

Should fix bugs #1116, #1012, #1000, #1129 in redmine.

Change-Id: I1b628d03ab588c29fef2b8789e61254da49c2b6f
```

Reviewer	Verified	Code-Review
Jenkins Buildbot	✓	
Mark Abraham		+1
Berk Hess		

- Need Code-Review

▶ Dependencies

Old Version History: Base

▼ Patch Set 4 2993637e075710153f38d86d7cb21365ae4e75fe (github)

Author	Michael Shirts <michael.shirts@virginia.edu> Jan 17, 2013 1:16 PM
Committer	Michael Shirts <michael.shirts@virginia.edu> Feb 13, 2013 10:12 PM
Parent(s)	2b05680cf39f30836a2d198c6182b4cfc96d6458 Plain text quotes.
Download	checkout | pull | cherry-pick | patch Anonymous HTTP git fetch https://gerrit.gromacs.org/gromacs refs/changes/01/2101/4 && git checkout FETCH_HEAD

	File Path	Comments	Size	Diff	
▸	Commit Message			Side-by-Side	Unified
M	include/force.h		+3, -1	Side-by-Side	Unified
M	include/types/inputrec.h		+2, -0	Side-by-Side	Unified
M	include/types/iteratedconstraints.h		+1, -1	Side-by-Side	Unified
M	include/update.h		+1, -1	Side-by-Side	Unified

How do I know if I’m sampling from the correct distribution?

Run the same system, same options,
but two different temperatures

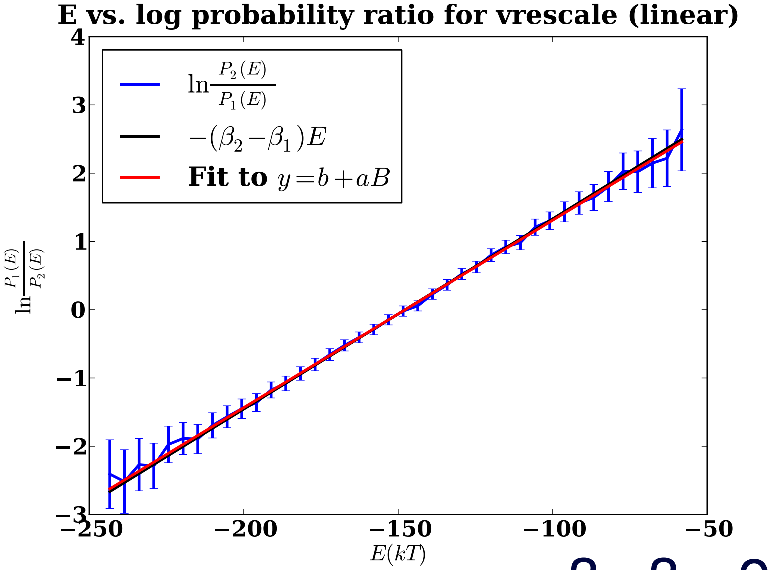
$$P_1(E) = Q_1^{-1}\Omega(E)e^{-\beta_1 E}$$

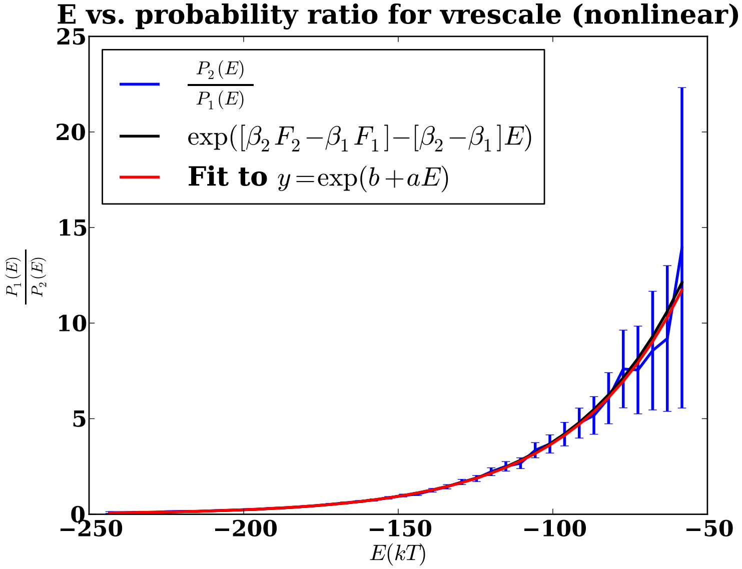
$$P_2(E) = Q_2^{-1}\Omega(E)e^{-\beta_2 E}$$

$$\frac{P_1(E)}{P_2(E)} = \frac{Q_2}{Q_1}e^{(\beta_2-\beta_1)E}$$

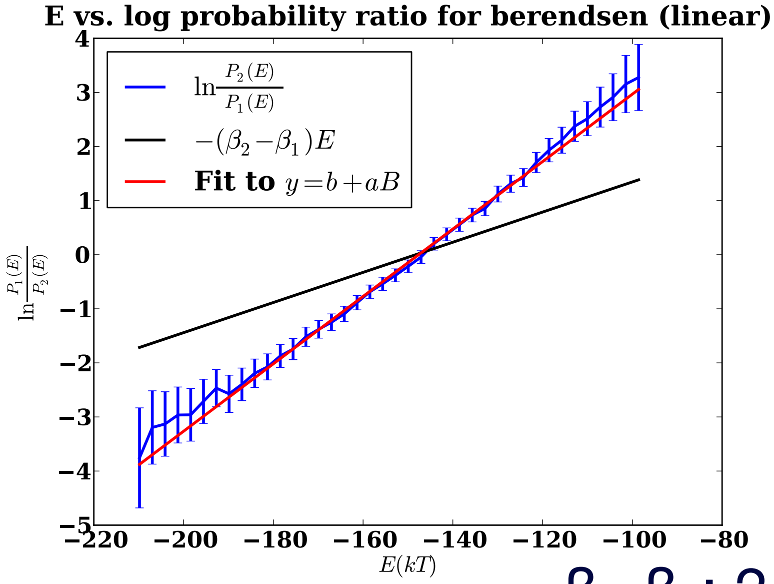
$$\ln\frac{P_1(E)}{P_2(E)} = \ln\frac{Q_2}{Q_1} + (\beta_2 - \beta_1)E$$

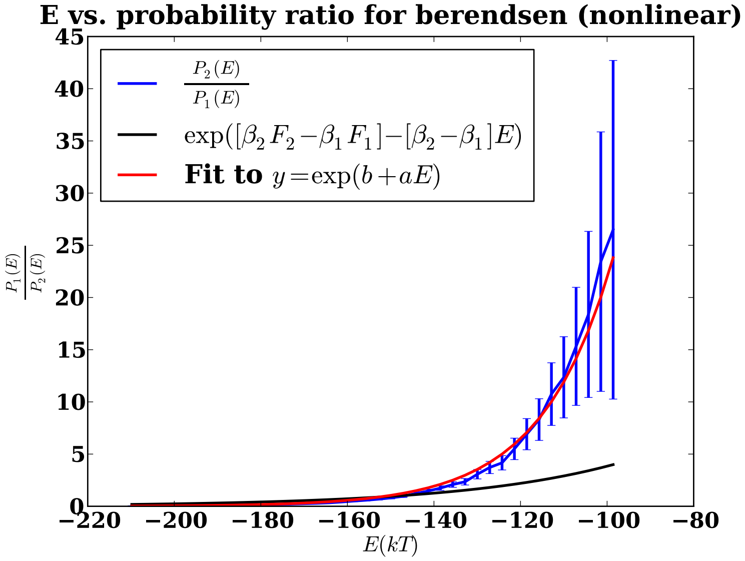
We can visually observe deviations from the correct energy distribution


E vs. log probability ratio for vrescale (linear)


E vs. probability ratio for vrescale (nonlinear)

β_2-β_1: 0.66 σ from true value


E vs. log probability ratio for berendsen (linear)

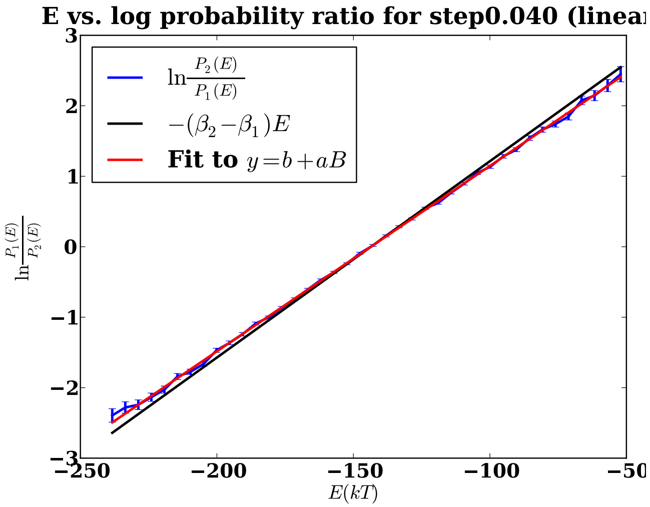

E vs. probability ratio for berendsen (nonlinear)

β_2-β_1: 24 σ from true value

These tests can validate simulation parameters in an automated, quantitative way

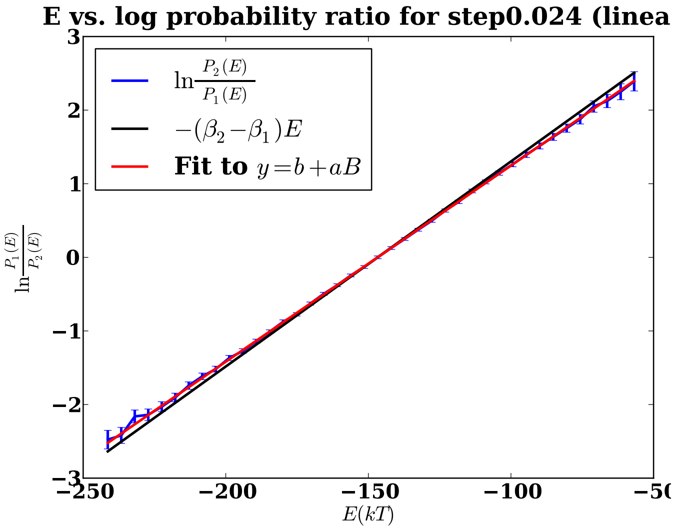
- Example: Validating the molecular dynamics time steps for argon

40 fs time step



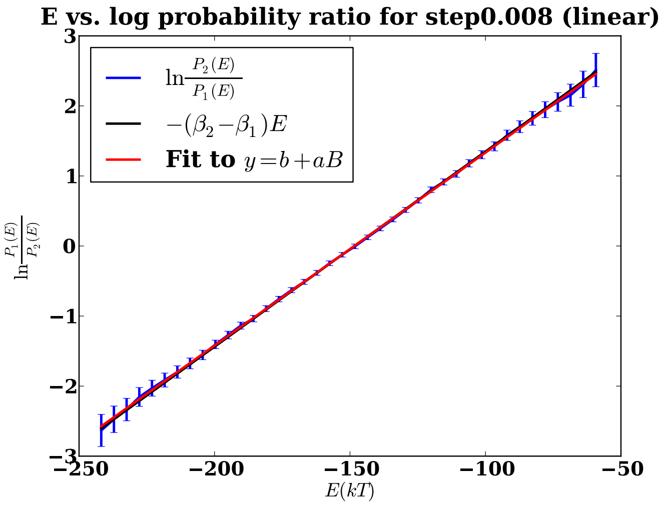
β2-β1: 14 σ from true

24 fs time step



β2-β1: 8 σ from true

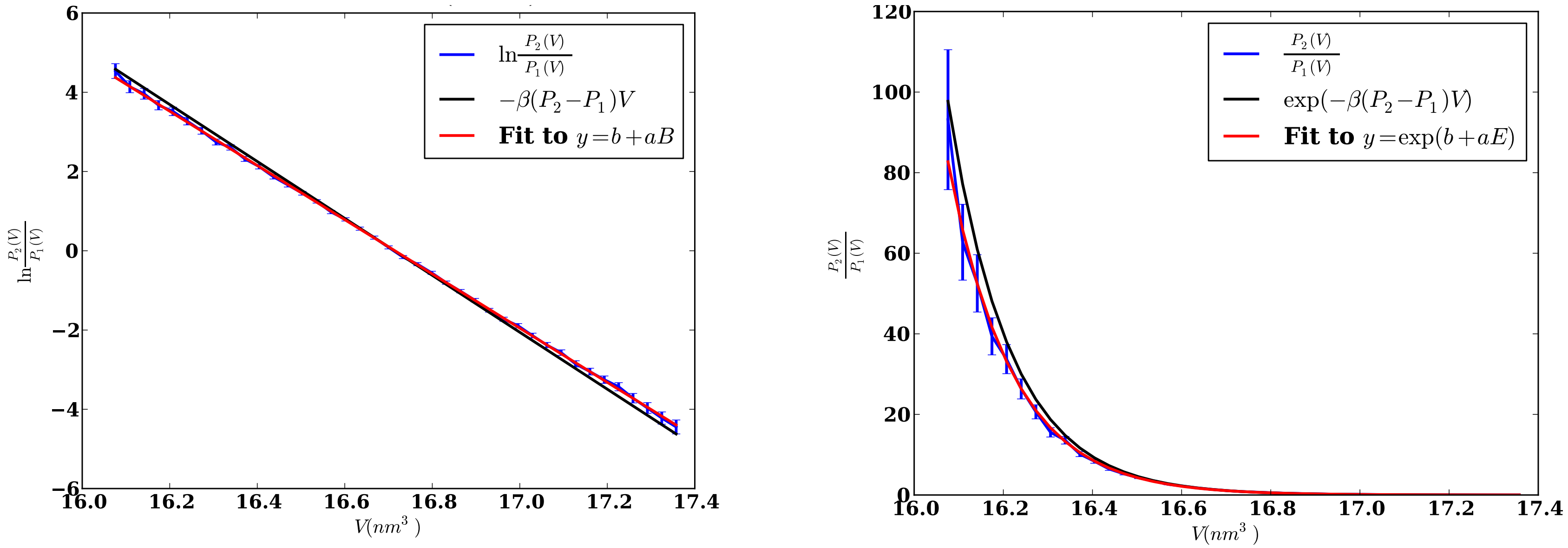
8 fs time step



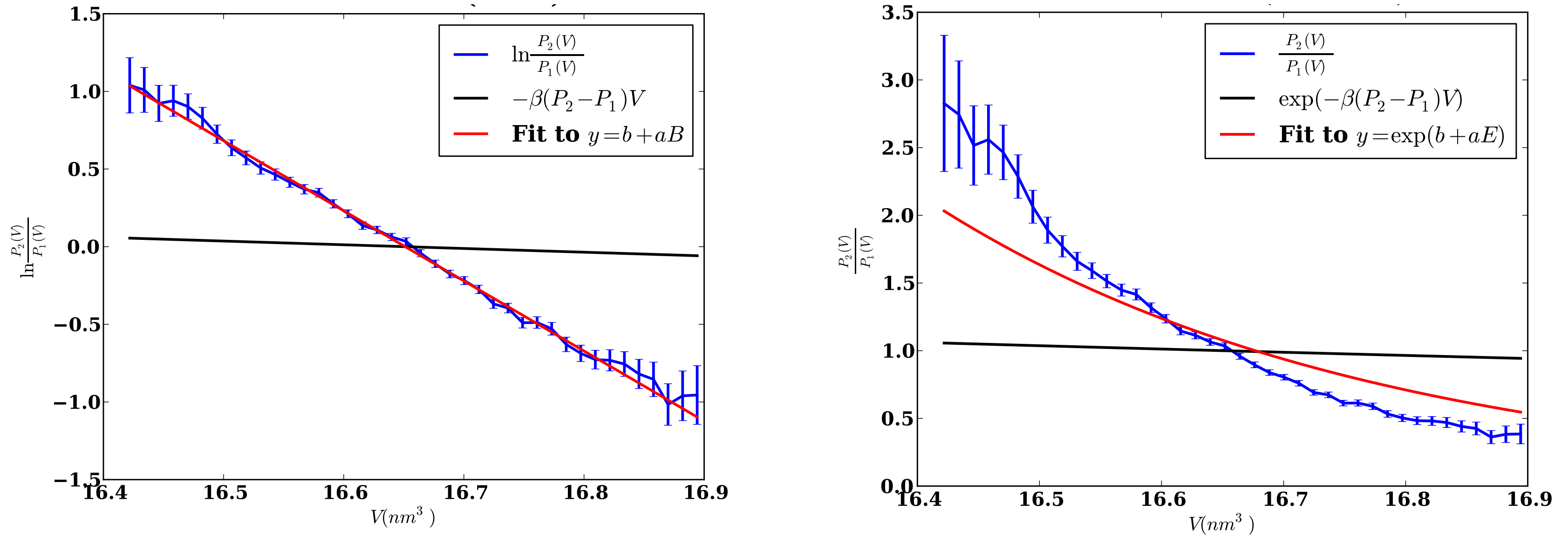
β2-β1: 1.0 σ from true

Validation of Volume Fluctuations in NPT

Parrinello-Rahman

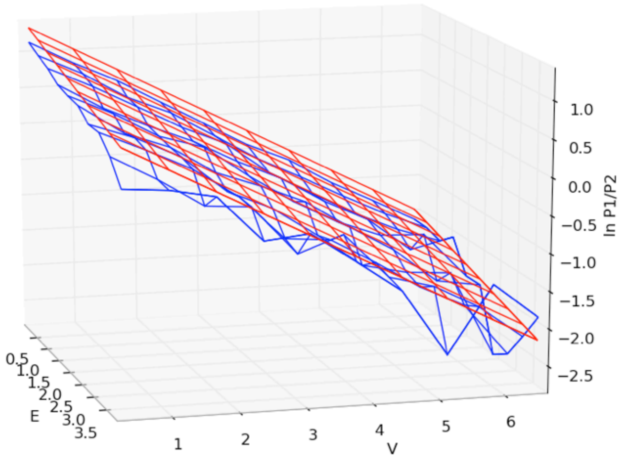


Berendsen



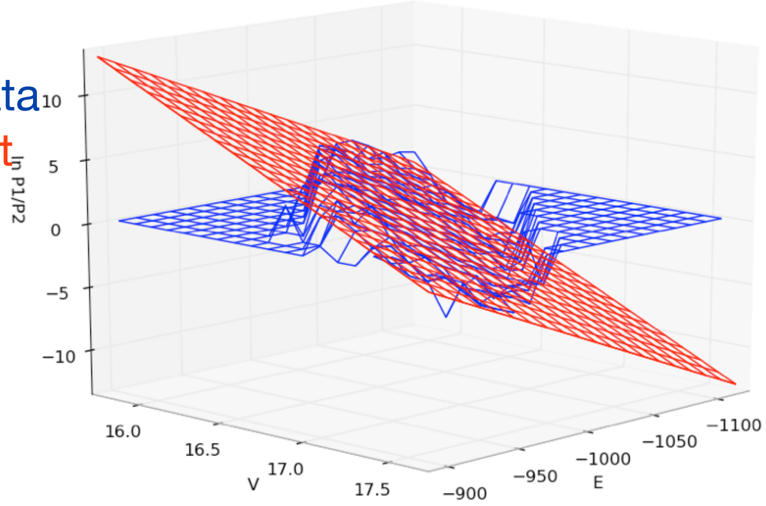
Other variations on a theme

- Can separate kinetic and potential energies
  - Can use for MC algorithms as well
- NPT simulations
  - Can look at distribution of E + PV
  - Can look at distribution of V alone
  - Can look at joint distribution of E and V
  - Grand canonical simulations



Model with K(V)

Blue = Data
Red = Fit



Lennard-Jones fluid

- Quantitative measurement as well
- Python implementation: https://simtk.org/home/checkensemble

Validation tools

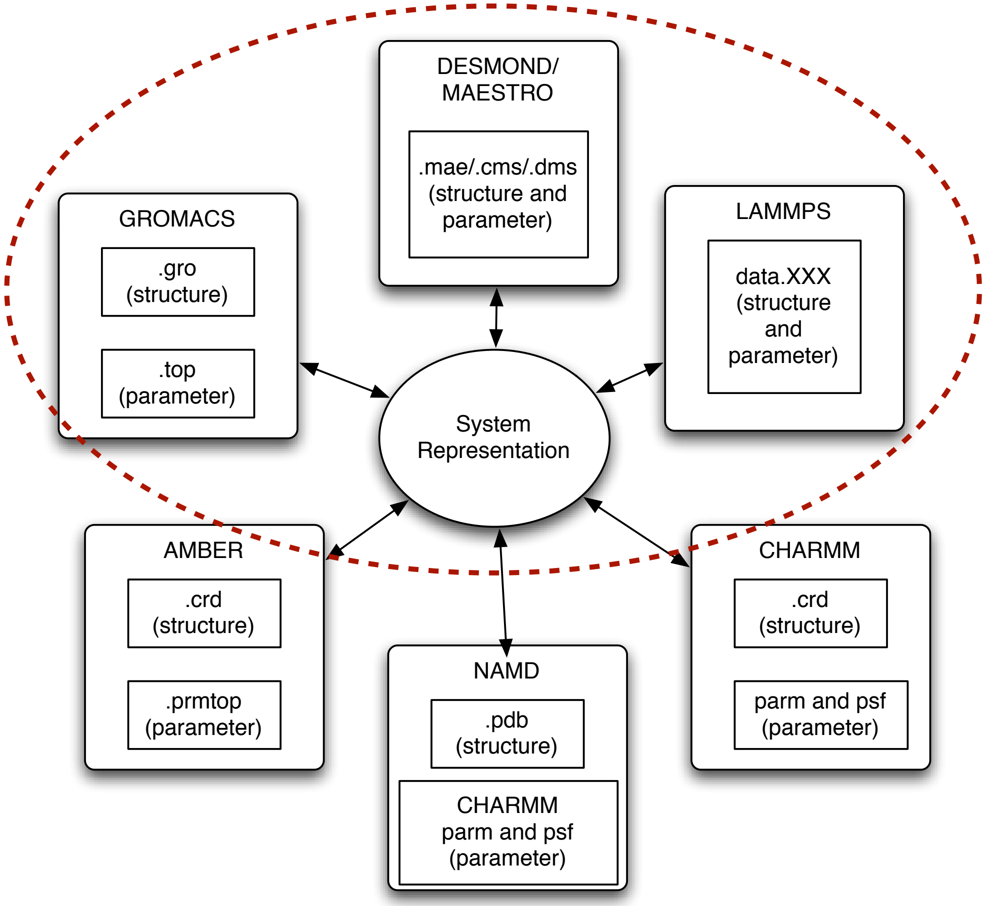
M. R. Shirts, *J. Chem. Theory Comput.*, 9, 909 (2013)

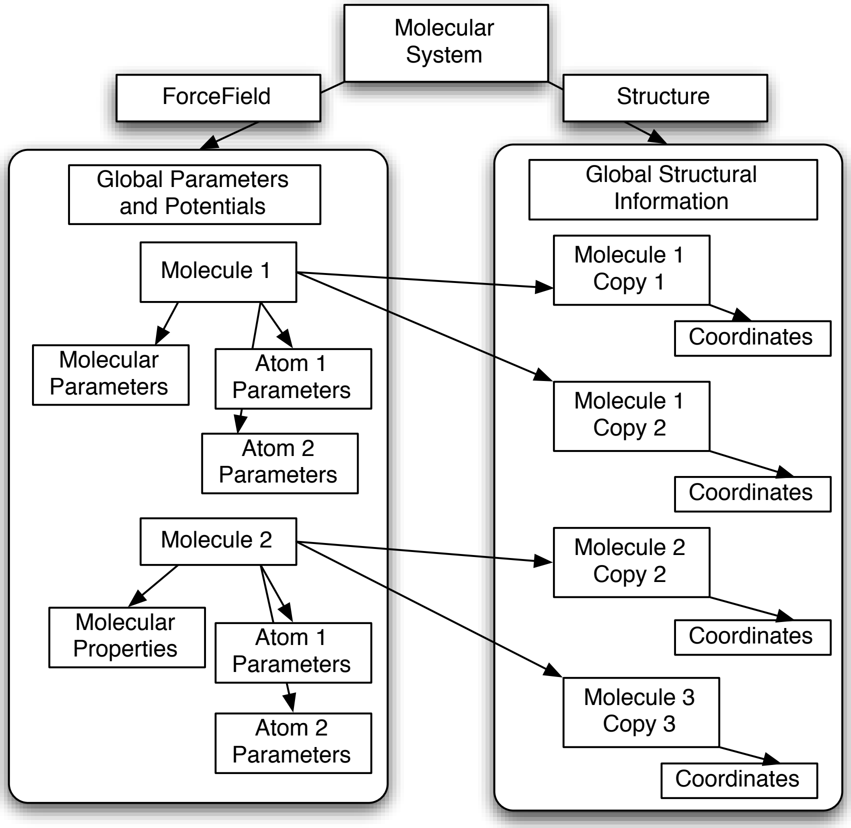
- Python implementation
  - https://github.org/shirtsgroup/checkensemble

- Quantitative, not just visual: weighted linear, nonlinear, and maximum likelihood fits
- NPT, NVT, and μVT supported
- Supports multiple MD formats (CHARMM, GROMACS, Desmond)
- Incorporates autocorrelation
- Automated graphing
- Replica exchange analysis

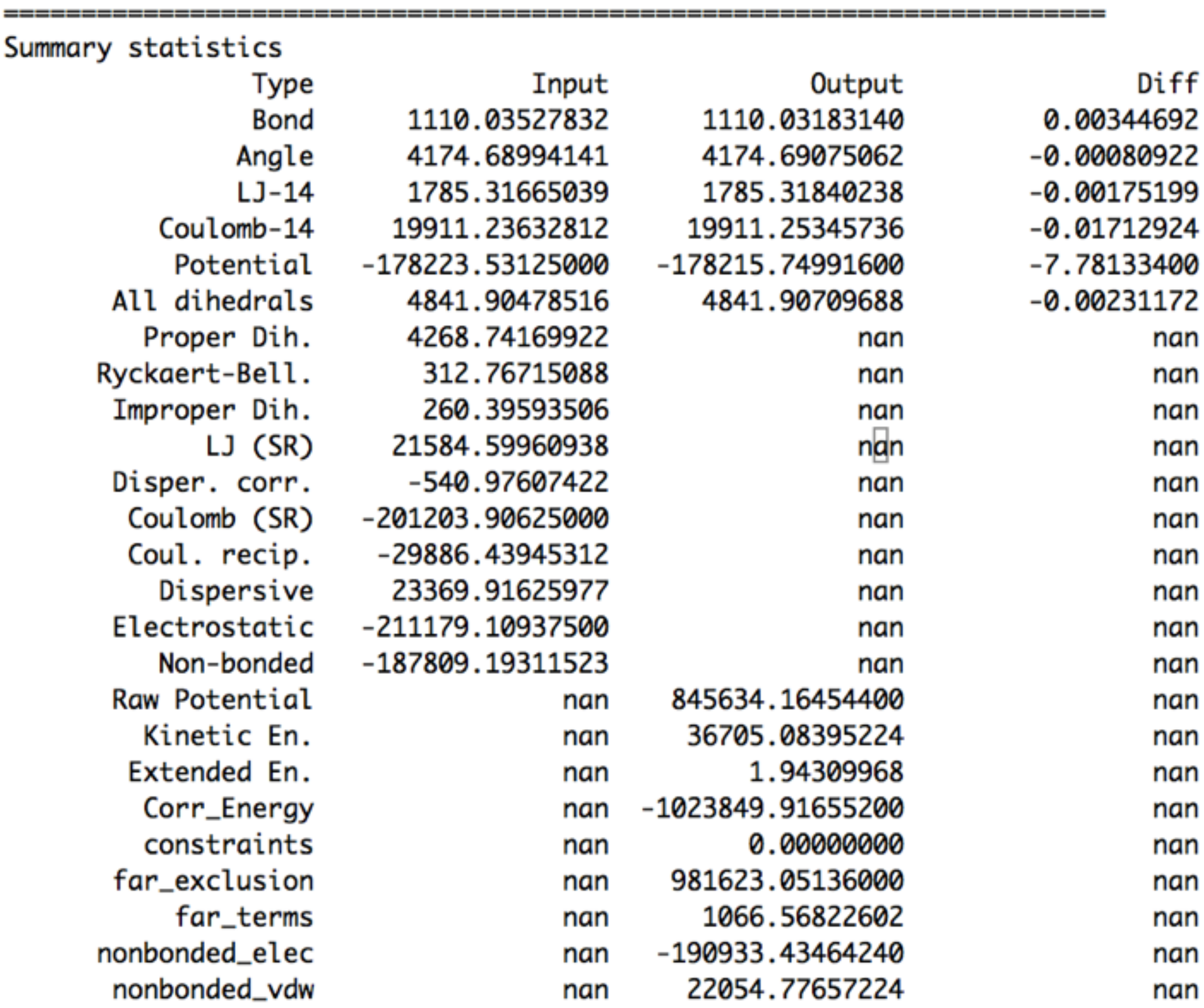
We can simplify conversion between simulation input files by automation: InterMol

Avoid N^2
different scripts





InterMol in practice (single precision comparison)



```
==================================================================================
Summary statistics
                 Type              Input             Output               Diff
                 Bond      1110.03527832      1110.03183140         0.00344692
                Angle      4174.68994141      4174.69075062        -0.00080922
                LJ-14      1785.31665039      1785.31840238        -0.00175199
           Coulomb-14     19911.23632812     19911.25345736        -0.01712924
            Potential   -178223.53125000   -178215.74991600        -7.78133400
        All dihedrals      4841.90478516      4841.90709688        -0.00231172
          Proper Dih.      4268.74169922                nan                nan
       Ryckaert-Bell.       312.76715088                nan                nan
        Improper Dih.       260.39593506                nan                nan
             LJ (SR)      21584.59960938                nan                nan
        Disper. corr.      -540.97607422                nan                nan
         Coulomb (SR)   -201203.90625000                nan                nan
          Coul. recip.   -29886.43945312                nan                nan
           Dispersive     23369.91625977                nan                nan
        Electrostatic   -211179.10937500                nan                nan
           Non-bonded   -187809.19311523                nan                nan
        Raw Potential                nan    845634.16454400                nan
          Kinetic En.                nan     36705.08395224                nan
         Extended En.                nan         1.94309968                nan
          Corr_Energy                nan  -1023849.91655200                nan
          constraints                nan         0.00000000                nan
        far_exclusion                nan    981623.05136000                nan
            far_terms                nan      1066.56822602                nan
       nonbonded_elec                nan   -190933.43464240                nan
        nonbonded_vdw                nan     22054.77657224                nan
```

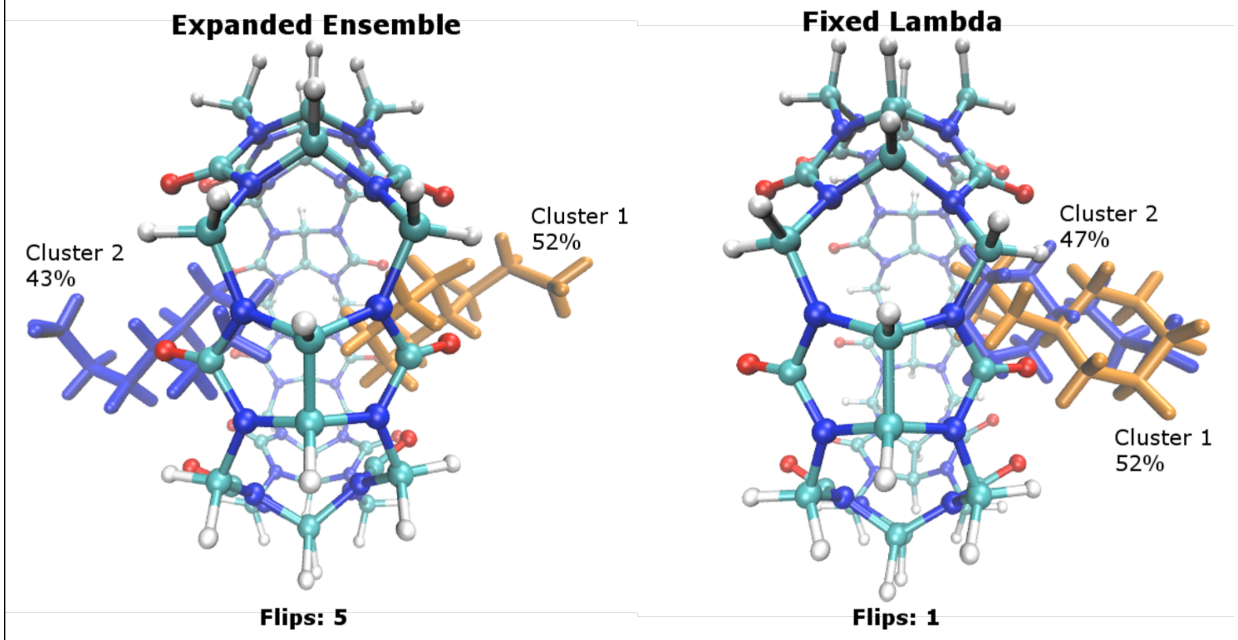
alpha code at https://github.com/shirtsgroup/intermol

What are best practices ways to overcome sampling issues?

- Convergence error for shoving waters out of the way around rigid stuff
  - Easy if a good soft core alchemical pathway used and moderate sampling
- Convergence error due to the protein moving around
  - Hard

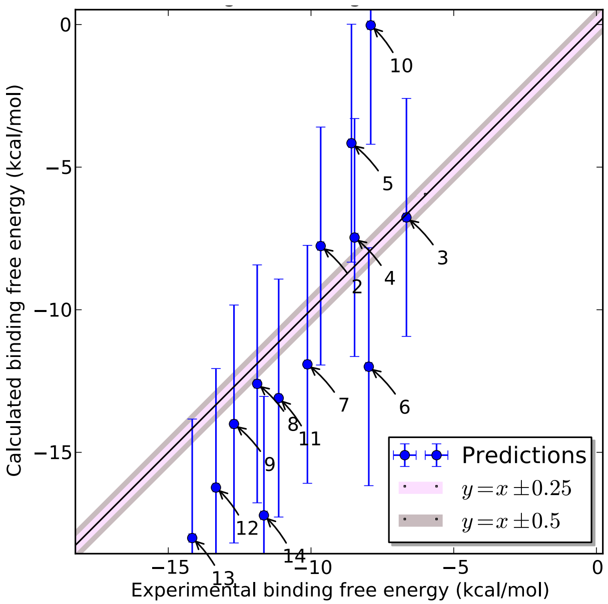
It’s not the size, it’s . . .

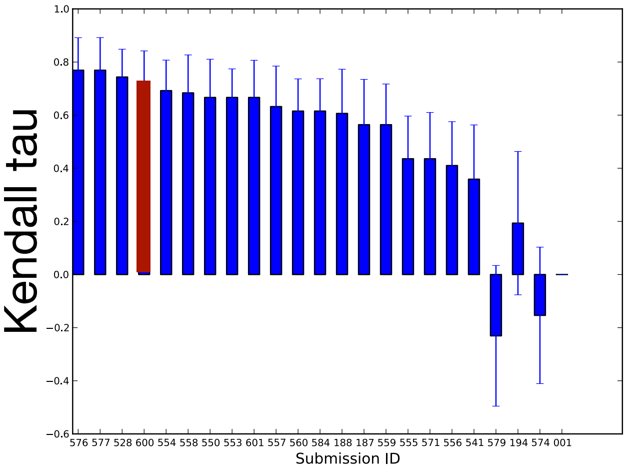
SAMPL4 blind prediction exercise
Model host-guest systems
Explicit solvent
Expanded ensemble
(serial replica exchange)
GROMACS 4.6.5



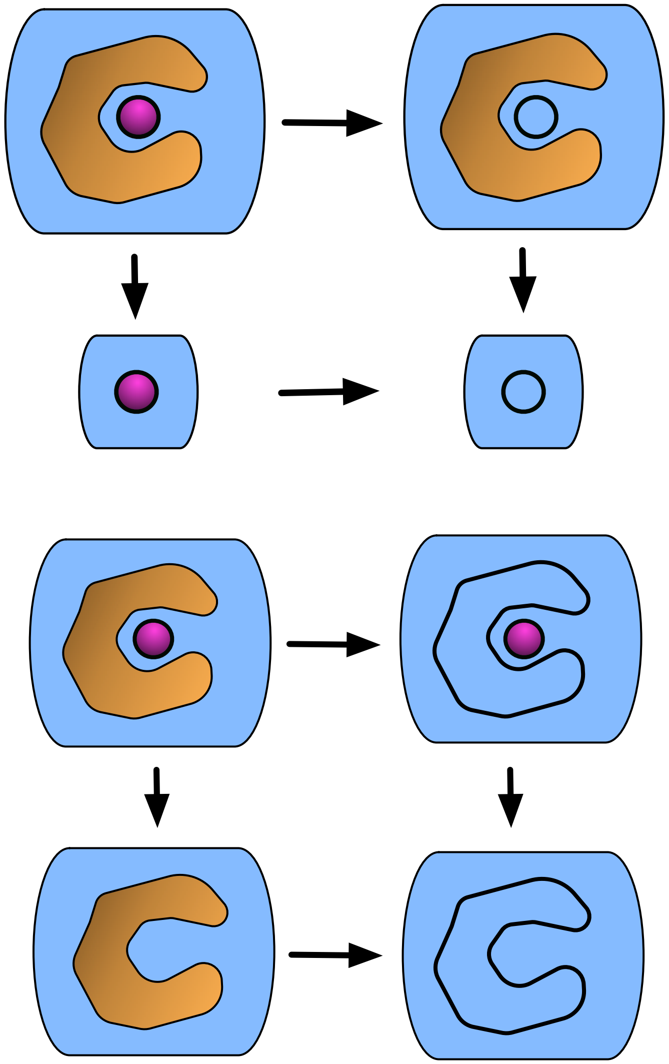
J. I. Monroe and M. R. Shirts. *J. Comput. Aid. Mol. Design.* (2014)

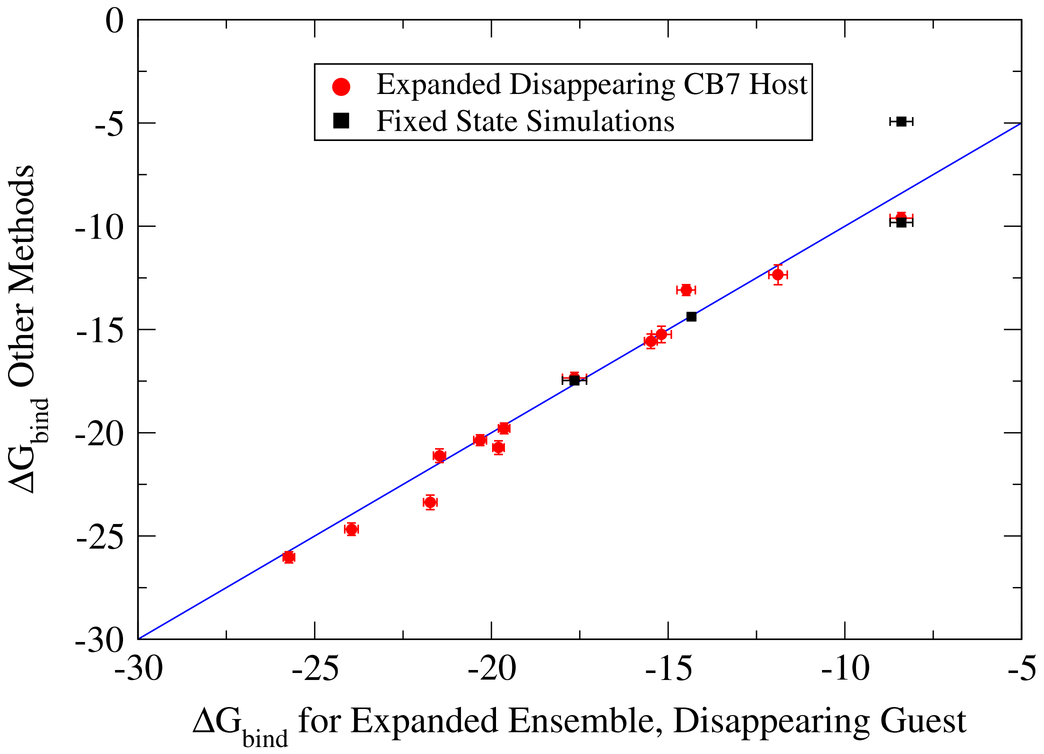
SAMPL4 blind prediction





With expanded ensemble, consistency in disappearing host and disappearing guest




Expanded Disappearing CB7 Host
Fixed State Simulations
ΔG_bind Other Methods
ΔG_bind for Expanded Ensemble, Disappearing Guest
0
-5
-10
-15
-20
-25
-30
-30
-25
-20
-15
-10
-5

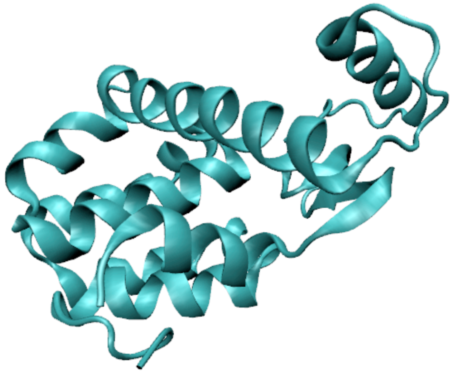
Differences are between two numbers ranging between150 kJ and 900 kJ/mol

Synergy: Use alchemical path to improve sampling

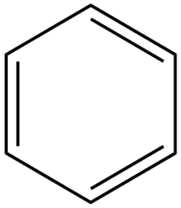
- Need enough local simulation to converge the free energy of displacing molecules in dense fluids
  - Move water out the way, move side chains
- Need enough long time scale simulation to sample the protein configurations
- Need a method to accelerate sampling
  - A bunch of acceleration methods
  - Using the alchemical pathway to increase sampling
- KEY: Swapping between states
  - Replica exchange
  - Expanded ensemble approaches: all states in one simulation
- Example: FEP/REST
  - One lambda to connect end state
  - One lambda to ‘floppify’ the binding site and ligand
  - Move through both lambdas simultaneously

Applications of sampling and free energy to the T4 lysozyme L99A model system

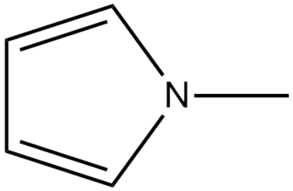
T4 Lysozyme



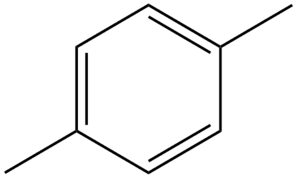
+



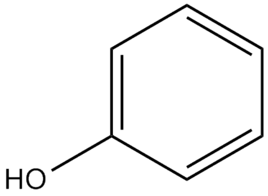
Original



Binder



Binds,
but doesn't fit
apo hole

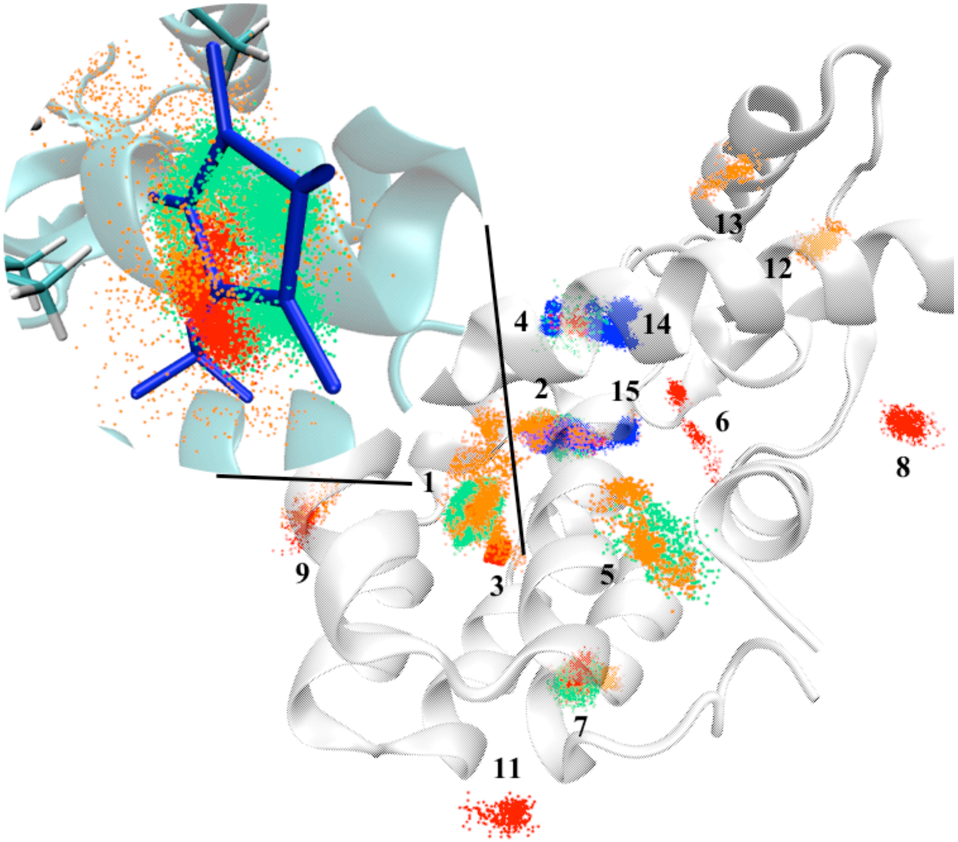


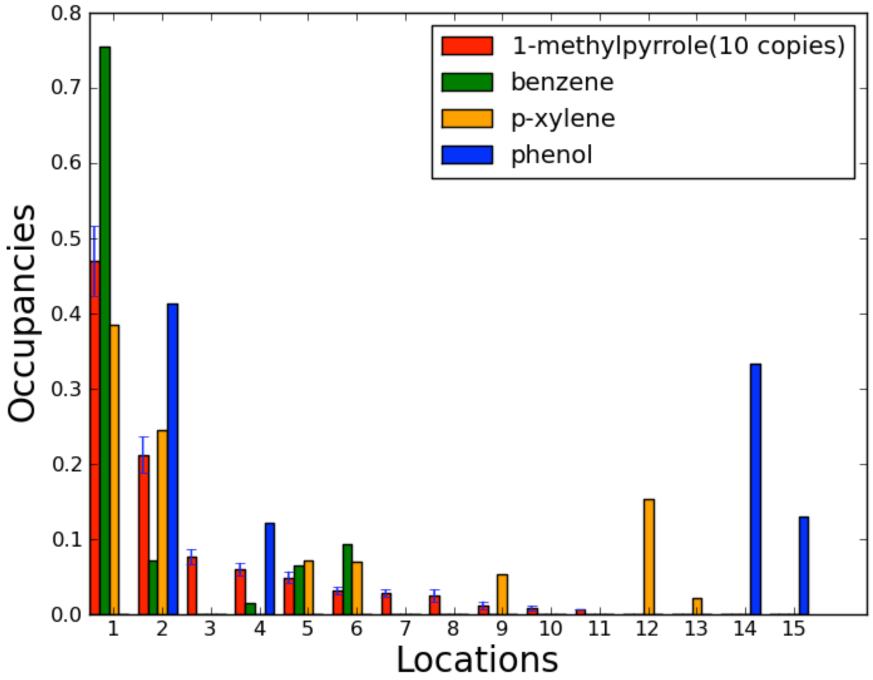
Doesn't bind

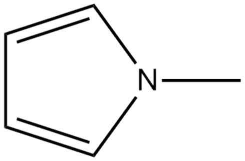
- Hamiltonian replica exchange between coupled and uncoupled states: absolute free energy calculation
- Linear Coulomb + soft-core van der Waals
- GPU accelerated implicit solvent dynamics via OpenMM
- 15 ns at each of 24 intermediates
- Restrain ligand near protein, but not to specific site

- Question: Can we sample well enough to know what ligand binding distribution looks like?

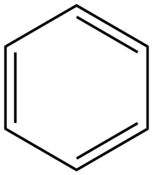
We get a consistent ensemble of small molecule binding locations



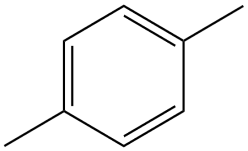




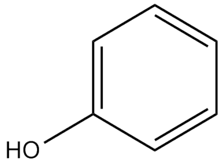
Binder



Original



Binder

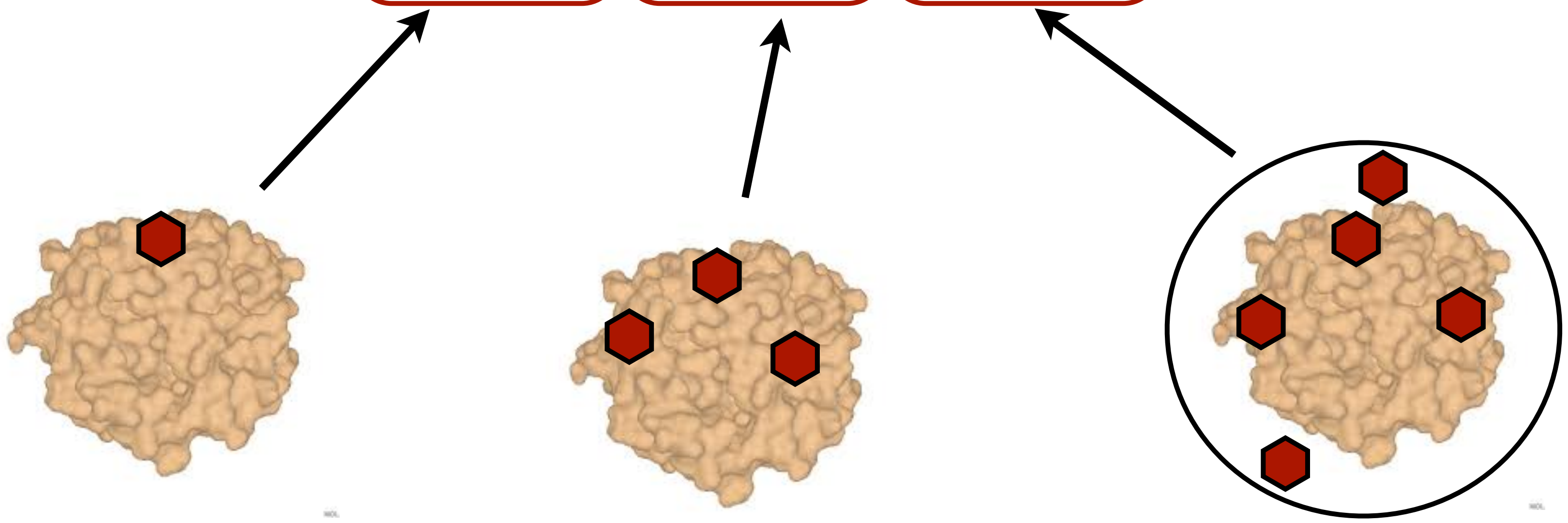


Nonbinder

K. Wang, J. D. Chodera, Y. Yang and M. R. Shirts. *J. Comput. Aid. Mol. Design.* **27**, 989 (2013)

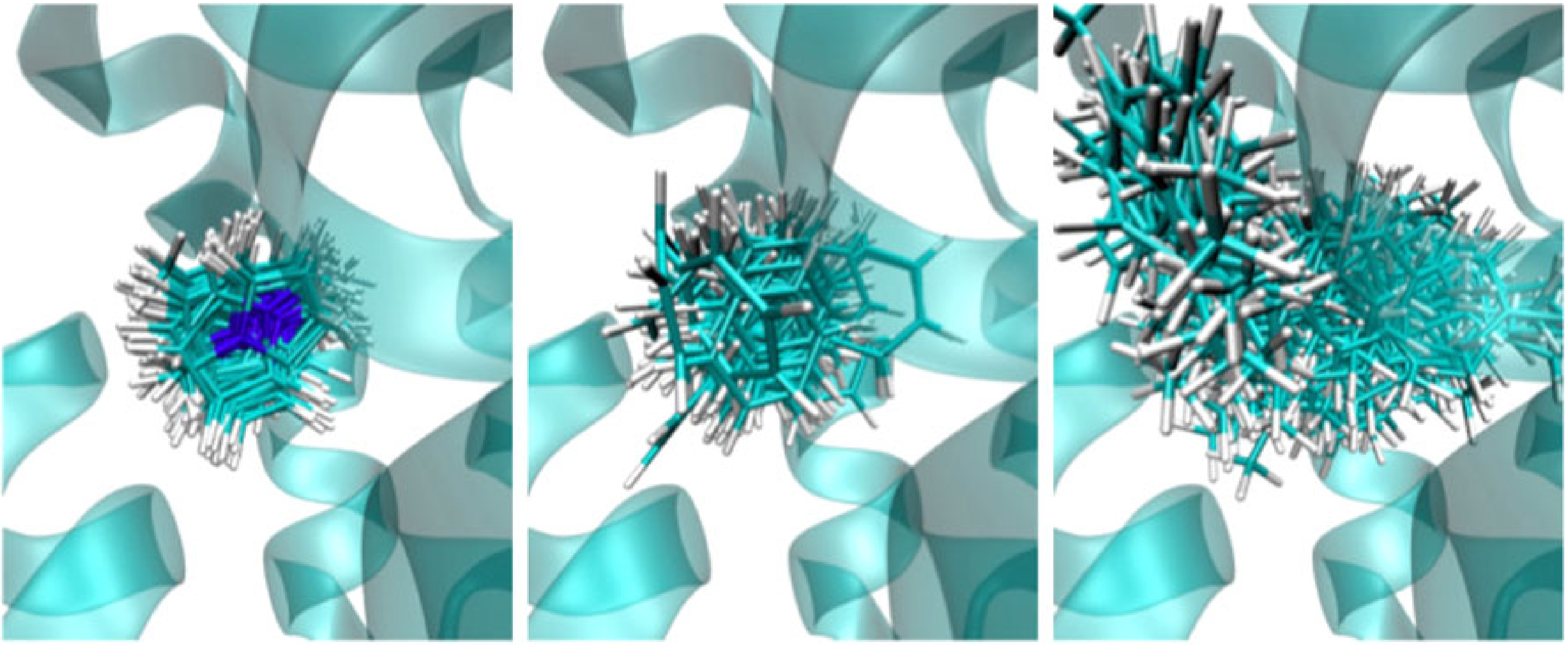
We can decompose free energies according to different definitions of binding

Molecules	ΔG_{site}	$\Delta G_{all\ sites}$	$\Delta G_{overall}$	$\Delta G_{explicit}$	$\Delta G_{experimental}$
1-methylpyrrole	-3.48 ± 0.26	-4.15 ± 0.25	-5.05 ± 0.21	-4.32 ± 0.08	-4.44
benzene	-4.26 ± 0.71	-5.15 ± 0.80	-6.01 ± 0.81	-4.56 ± 0.20	-5.19
p-xylene	-4.01 ± 0.89	-4.94 ± 0.85	-5.72 ± 0.95	-3.54± 0.17	-4.67
phenol	-1.03± 0.32	-1.78 ± 0.47	-2.32 ± 0.58	-1.26 ± 0.09	> -2.74



K. Wang, J. D. Chodera, Y. Yang and M. R. Shirts. *J. Comput. Aid. Mol. Design.* **27**, 989 (2013)

We can capture structural heterogeneity and reorganization in the binding site



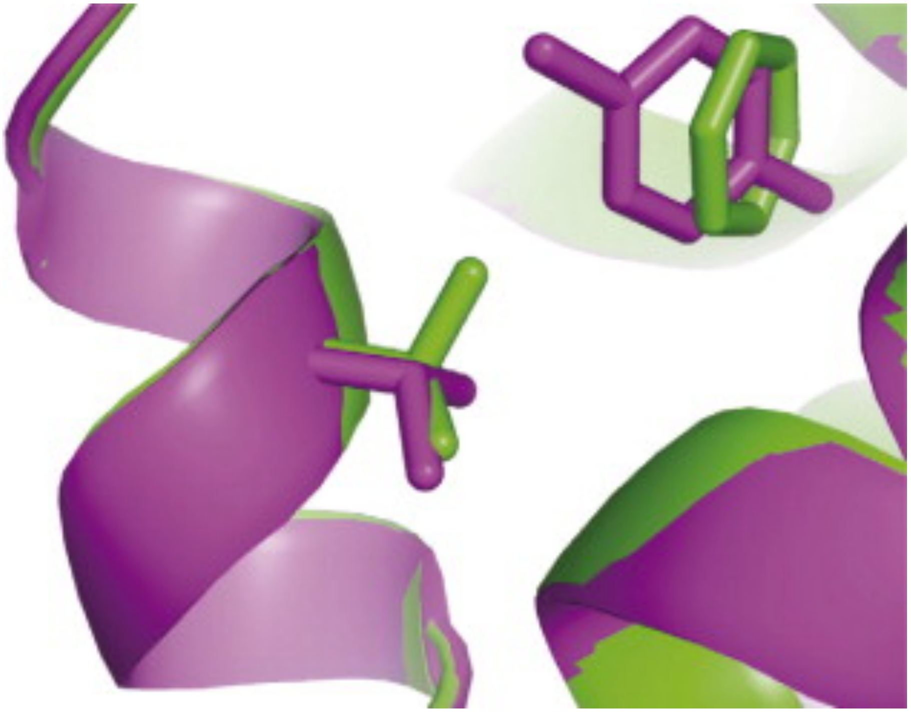
1-methylpyrrole

benzene

p-xylene

K. Wang, J. D. Chodera, Y. Yang and M. R. Shirts. *J. Comput. Aid. Mol. Design.* **27**, 989 (2013)

A known problem:
Val111 movement is required for *p*-xylene to bind

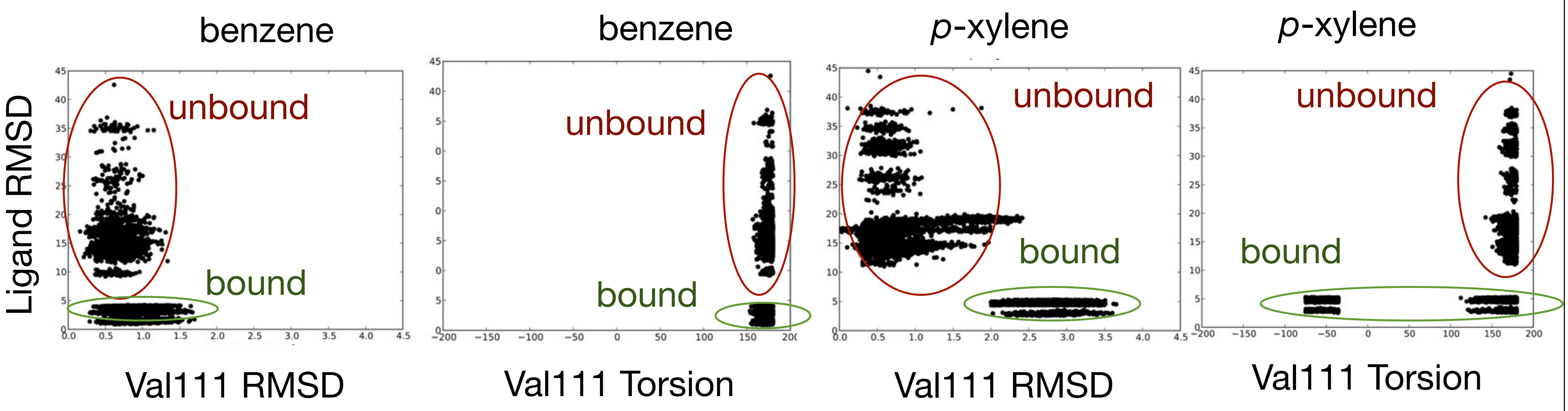


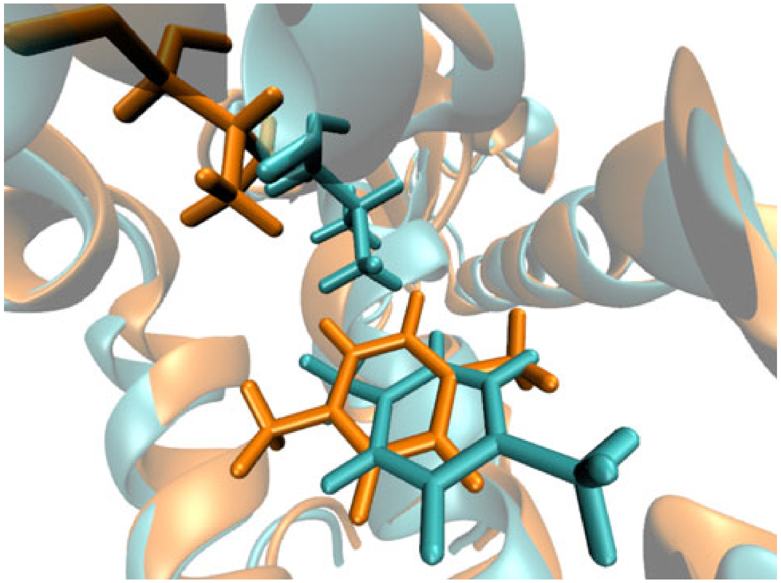
Benzene co-crystal

p-xylene co-crystal

Mobley et al. *J. Mol. Biol.,* **317**, 1118 (2007)

We can capture known conformational rearrangement





crystal-like

RMSD 0.3Å
Rotation of Val111
53% of site

alternate

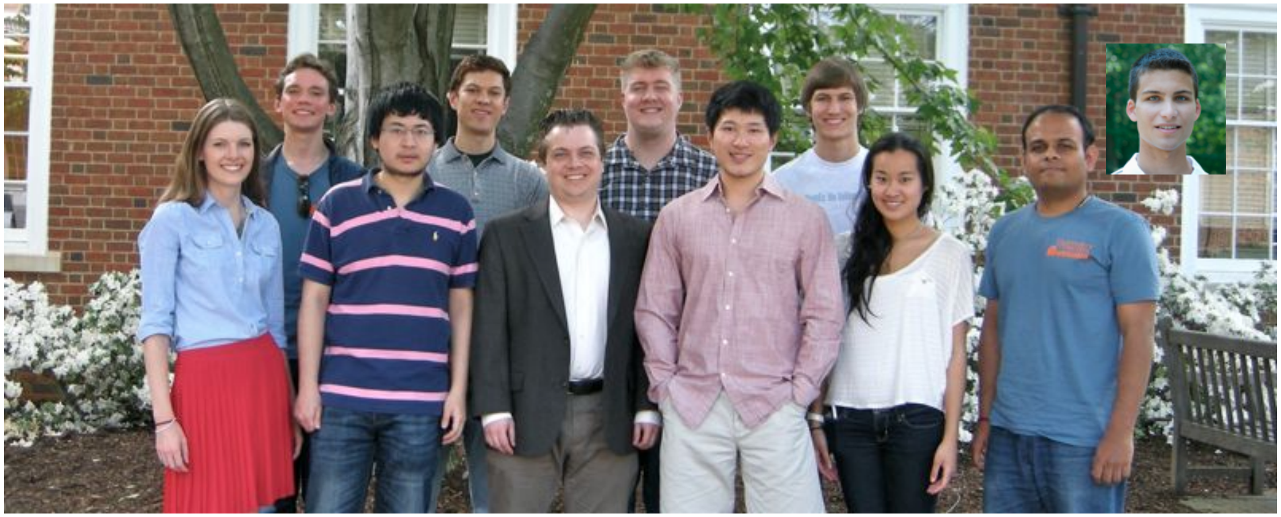
RMSD 2.7Å
Shift of backbone
moving Val111
32% of site

K. Wang, J. D. Chodera, Y. Yang and M. R. Shirts. *J. Comput. Aid. Mol. Design.* **27**, 989 (2013)

Concluding Remarks?

- Not all of the roadblocks to better simulation are merely about efficiency, or even accuracy some are about useability and robustness
- We should be moving towards community best practices to change this
- Can we develop community knowledge to improve useability?
- Can we develop and compare the tools and methods as a community to increase robustness?
- Will this all decrease confusion and improve simplicity?

Shirts Group



Past members

Tri Pham

Himanshu Paliwal
Jon Fuller
Christoph Klein
Ellen Zhong
Yanzhi Yang
Jacob Monroe
Chris Lee

Alchemistry.org Coupled sampling InterMol Reweighting

Graduate

Kai Wang
Joe Basconi
Levi Naden
Brittany Zimmerman

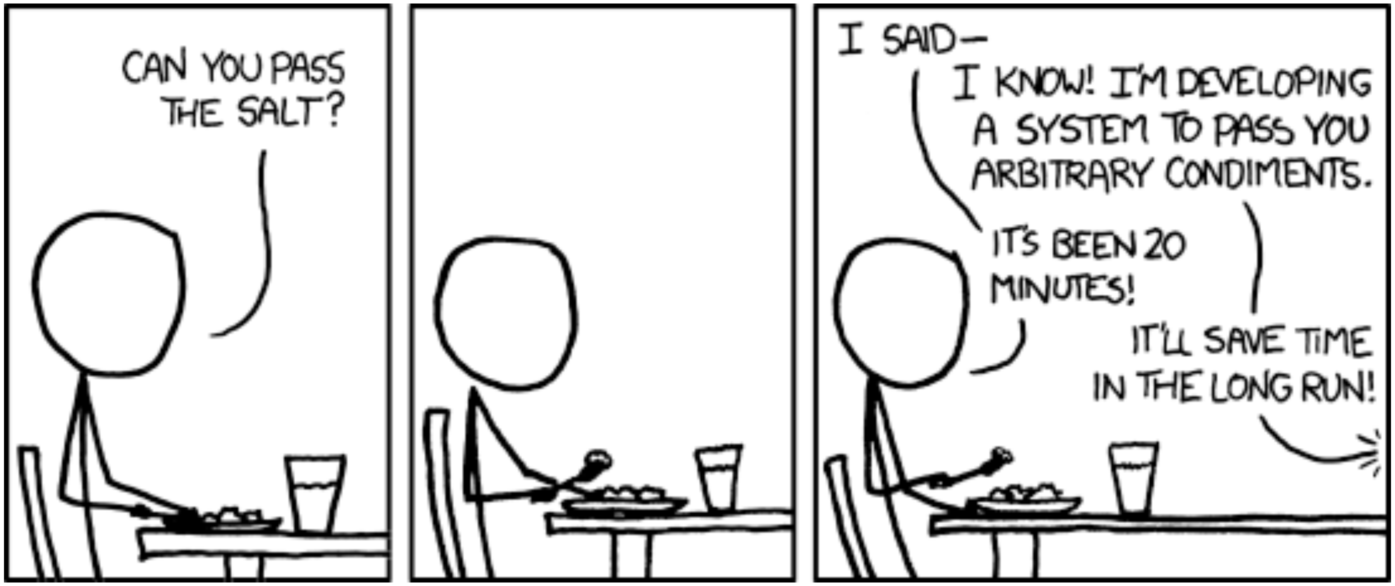
Undergraduate

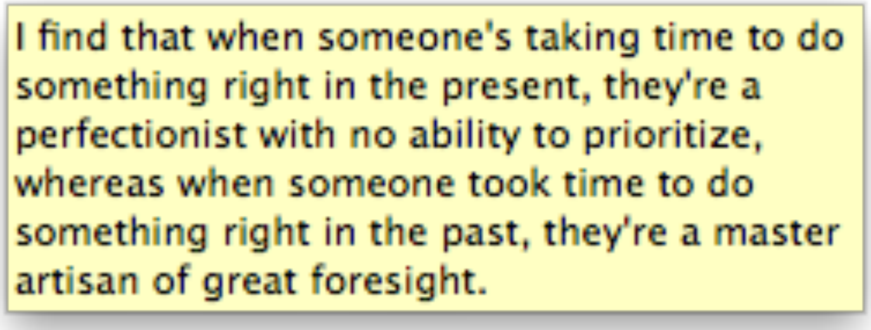
Jacob Rosenthal
Mitch Slovin
Alex Yang

External:

John Chodera (MSKCC)
David Mobley (UC-Irvine)

Let’s work to make free energy calculations methods more <u>powerful</u> AND <u>easier</u>


CAN YOU PASS THE SALT?
I SAID—
I KNOW! I'M DEVELOPING A SYSTEM TO PASS YOU ARBITRARY CONDIMENTS.
IT'S BEEN 20 MINUTES!
IT'LL SAVE TIME IN THE LONG RUN!


I find that when someone's taking time to do something right in the present, they're a perfectionist with no ability to prioritize, whereas when someone took time to do something right in the past, they're a master artisan of great foresight.