Simulating Biomolecules with Variable Protonation State: Constant-pH Molecular Dynamics Simulations with NAMD

Brian Radak

University of Illinois at Urbana–Champaign
Beckman Institute and Center for Macromolecular Modeling & Bioinformatics

Computational Biophysics Workshop –
Enhanced Sampling and Free Energy Calculations
September 11, 2018

Acknowledgements (other people to blame)

Univ of Chicago

- Benoît Roux
- Donghyuk Suh

ALCF

- Wei Jiang

UIUC

- Dave Hardy
- Jim Phillips
- Chris Chipot
- Abhi Singharoy (ASU)
- Shashank Pant
- Emad Tajkhorshid

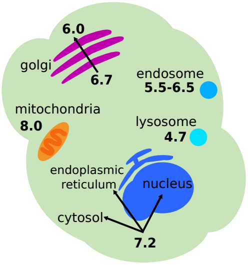
Argonne
NATIONAL LABORATORY

Theta Early Science Program

mpsdc MEMBRANE PROTEIN
STRUCTURAL DYNAMICS GATEWAY

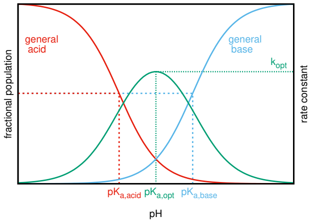
pH Effects in Biochemistry

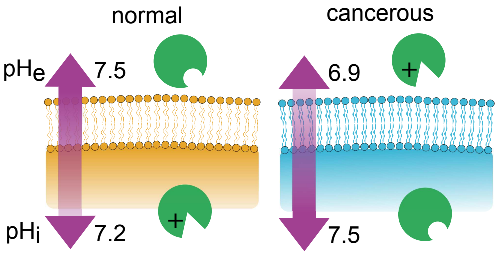
Casey, *et al Nat Rev Mol Cell Biol*, **2010**



variability of pH by region

enzyme rate vs. pH





pH gradients at cell surfaces

Webb, *et al Nat Rev Cancer*, **2011**

Constant pH and the semi-grand canonical ensemble

- Conventional MD samples a canonical ensemble:

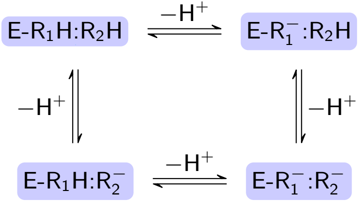
$$Q = \int d\boldsymbol{x}\, e^{-\beta U(\boldsymbol{x})}$$

- Constant-pH MD samples a semi-grand canonical ensemble:

$$\Xi(\mathrm{pH}) = \sum_{\boldsymbol{\lambda} \in \mathcal{S}} Q_{\boldsymbol{\lambda}} 10^{-n_{\boldsymbol{\lambda}} \mathrm{pH}}$$

The added interaction is between the number of protons, $n_{\boldsymbol{\lambda}}$, and a pH bath. $\boldsymbol{\lambda}$ is a new variable designating the protonation state.

Networks of protonation states

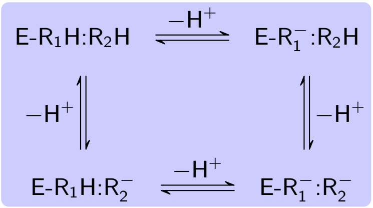


$E\text{-}R_1H{:}R_2H \xrightleftharpoons{-H^+} E\text{-}R_1^-{:}R_2H$

$-H^+$ $-H^+$

$E\text{-}R_1H{:}R_2^- \xrightleftharpoons{-H^+} E\text{-}R_1^-{:}R_2^-$

conventional MD



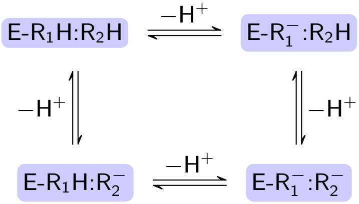
$E\text{-}R_1H{:}R_2H \xrightleftharpoons{-H^+} E\text{-}R_1^-{:}R_2H$

$-H^+$ $-H^+$

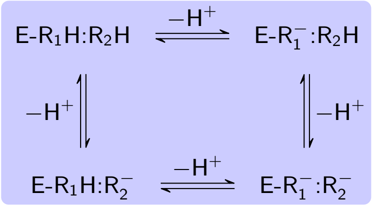
$E\text{-}R_1H{:}R_2^- \xrightleftharpoons{-H^+} E\text{-}R_1^-{:}R_2^-$

constant pH MD

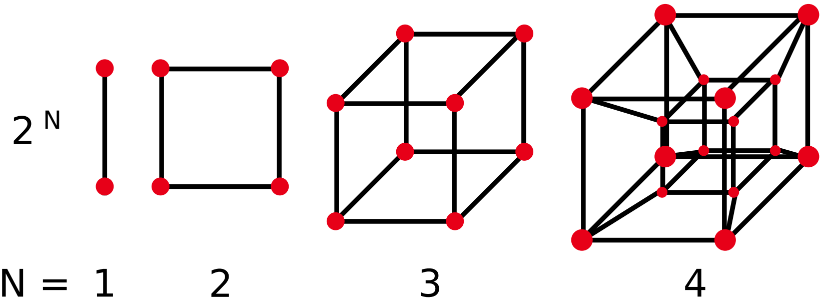
Networks of protonation states


$\text{E-R}_1\text{H:R}_2\text{H}$
$-\text{H}^+$
$\text{E-R}_1^-\text{:R}_2\text{H}$
$-\text{H}^+$
$-\text{H}^+$
$\text{E-R}_1\text{H:R}_2^-$
$-\text{H}^+$
$\text{E-R}_1^-\text{:R}_2^-$

conventional MD


$\text{E-R}_1\text{H:R}_2\text{H}$
$-\text{H}^+$
$\text{E-R}_1^-\text{:R}_2\text{H}$
$-\text{H}^+$
$-\text{H}^+$
$\text{E-R}_1\text{H:R}_2^-$
$-\text{H}^+$
$\text{E-R}_1^-\text{:R}_2^-$

constant pH MD



pH as a *thermodynamic* force

- Classical MD utilizes *mechanical* forces

$$\boldsymbol{F} = -\nabla U[\boldsymbol{x}(t)]$$

- pH may be regarded as a *thermodynamic* force

$$\mathrm{pH} = -\frac{1}{\ln 10}\frac{\partial \ln \Xi}{\partial n_{\boldsymbol{\lambda}}}$$

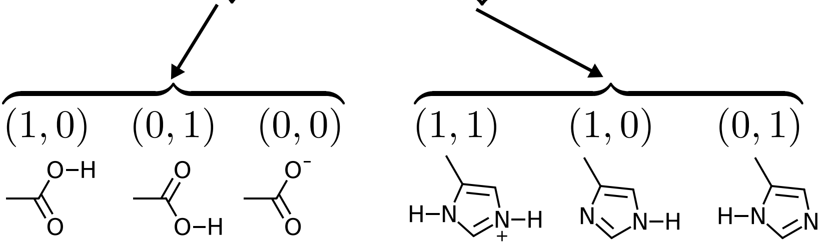
Mechanical forces – deterministic/stochastic dynamics
Thermodynamic forces – probabilistic “dynamics”

$$P_{\boldsymbol{\lambda}}(\mathrm{pH}) \propto Q_{\boldsymbol{\lambda}} 10^{-n_{\boldsymbol{\lambda}}\mathrm{pH}}$$

How do we define nodes in the network?

Consider a system with *m* sites:

$$\boldsymbol{\lambda} = \{\underbrace{\lambda_1, \lambda_2;}_{}\ldots\underbrace{\lambda_s, \lambda_{s+1};}_{}\ldots\lambda_m\}$$



Protonation state probabilities/populations

$$\langle A(\boldsymbol{x}, \boldsymbol{\lambda})\rangle_{\mathrm{pH}} = \frac{\sum_{\boldsymbol{\lambda}\in\mathcal{S}} \int d\boldsymbol{x}\, A(\boldsymbol{x}, \boldsymbol{\lambda}) e^{-\beta U(\boldsymbol{x};\boldsymbol{\lambda})} 10^{-n_{\boldsymbol{\lambda}}\mathrm{pH}}}{\Xi(\mathrm{pH})}$$

$P_{\lambda_s} = \langle \lambda_s \rangle_{\mathrm{pH}}$ – the probability that site s is occupied

There are two kinds of terms in the summation, $\lambda_s = 0/1$

$$\Xi(\mathrm{pH}) = \Xi_0(\mathrm{pH}) + \Xi_1(\mathrm{pH})10^{-\mathrm{pH}}$$

thus,

$$\langle \lambda_s \rangle_{\mathrm{pH}} = \frac{\Xi_1(\mathrm{pH})10^{-\mathrm{pH}}}{\Xi_0(\mathrm{pH}) + \Xi_1(\mathrm{pH})10^{-\mathrm{pH}}} = \frac{1}{1 + \frac{\Xi_0(\mathrm{pH})}{\Xi_1(pH)}10^{\mathrm{pH}}}$$

Connection to thermodynamics

$$\langle \lambda_s \rangle_{\mathrm{pH}} = \frac{1}{1 + \frac{\Xi_0(\mathrm{pH})}{\Xi_1(pH)} 10^{\mathrm{pH}}}$$

compares to the Henderson-Hasselbalch equation such that

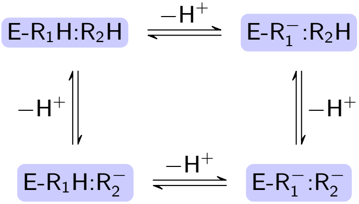
$$\mathrm{p}K_{\mathrm{a}}(\mathrm{pH}) = -\log \frac{\Xi_0(\mathrm{pH})}{\Xi_1(\mathrm{pH})},$$

except that now $\mathrm{p}K_{\mathrm{a}}(\mathrm{pH})$ is pH *dependent*. One often uses the approximation:

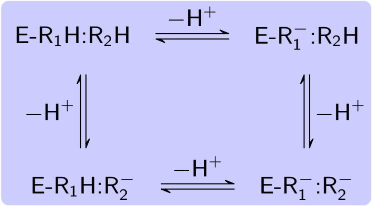
$$\mathrm{p}K_{\mathrm{a}}(pH) \approx \mathrm{p}K_{\mathrm{a}}^{(\mathrm{a})} + (1 - n)\left(\mathrm{pH} - \mathrm{p}K_{\mathrm{a}}^{(\mathrm{a})}\right),$$

where n is the Hill coefficient and $\mathrm{p}K_{\mathrm{a}}^{(\mathrm{a})}$ is the “apparent” $\mathrm{p}K_{\mathrm{a}}$.

Networks of protonation states



conventional MD



constant pH MD

We can now see that the fraction of simulation time spent in a given protonation state is directly impacted by the *difference* of the pK_a of a residue/site and the pH.

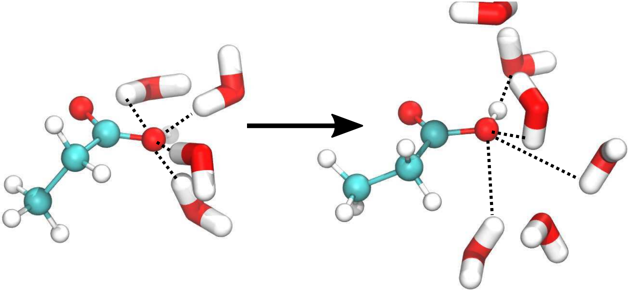
That's great – how do we sample the states?

1. Sample the configuration space of a given state
   (*i.e.*, sample $\boldsymbol{x}$ for a given $Q_{\boldsymbol{\lambda}}$)

2. Change between protonation states according to the number of protons and the given pH
   (*i.e.*, sample $\boldsymbol{\lambda}$ and choose a new $Q_{\boldsymbol{\lambda}}$)

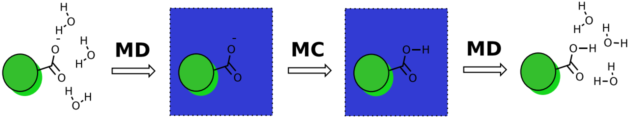
This may be regarded as a **Gibbs sampling**, whereby the configuration and state are sampled in an *alternating* fashion.

A problem! Environmental response



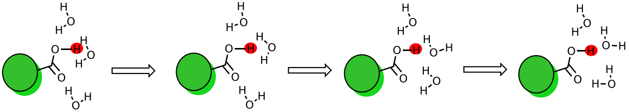
- (De)Protonation is a significant electrostatic event.
- Non-trivial reorganization of solvent, possibly solute.
- Naive sudden changes in protonation are likely to cause high energy configurations and/or steric clashes.

Possible solutions to the solvent clash problem



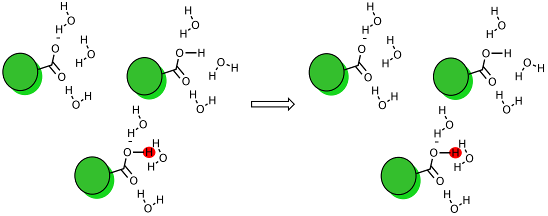
auxillary implicit solvent

Baptista, et al. **2002**.
Swails, et al. **2014**.



continuous fractional proton

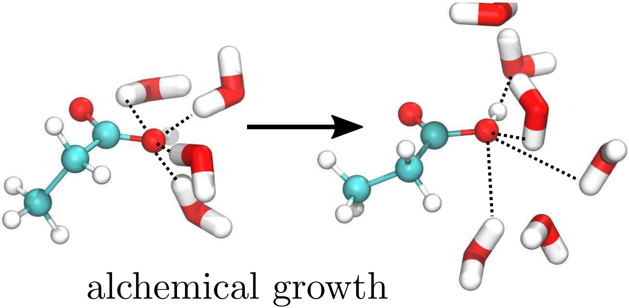
Lee, et al. **2004**,
Donnini, et al. **2011**.



discrete copy fractional proton

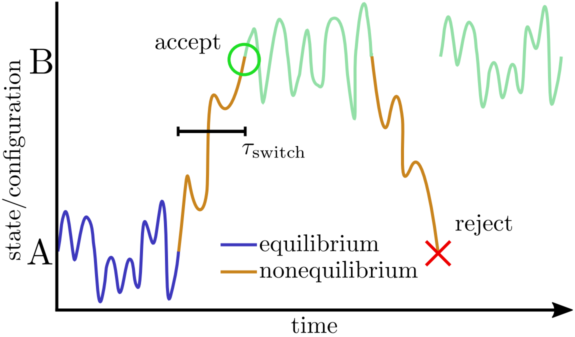
Lee, et al. **2014.**

“Fast” alchemical growth



- Swap the protonation state by using time-dependent interactions.
- Gradually stronger interactions will induce solvent response.
- Clashes are avoided by using the natural dynamics of the model.

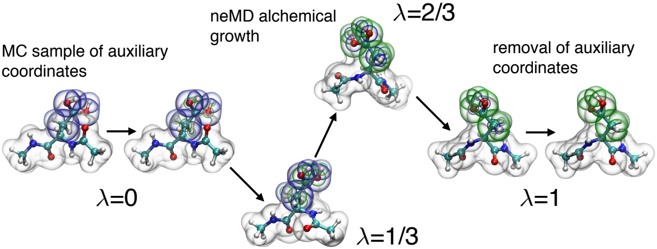
The neMD/MC constant pH paradigm



- Drive alchemical growth with *nonequilibrium* work
- Accept/reject with a generalized Metropolis criterion

Stern *J Chem Phys*, **2007**; Chen & Roux *J Chem Theory Comput*, **2015**;
Radak, *et al. J Chem Theory Comput*, **2017**

The neMD/MC constant pH paradigm



- Drive alchemical growth with *nonequilibrium* work
- Accept/reject with a generalized Metropolis criterion

Stern *J Chem Phys*, **2007**; Chen & Roux *J Chem Theory Comput*, **2015**;
Radak, *et al.* *J Chem Theory Comput*, **2017**

Beyond Gibbs sampling: Hybrid MD and neMD/MC

We now alternate conventional sampling with MD ($\boldsymbol{x}$) and Metropolis Monte Carlo sampling ($\boldsymbol{x}$ *and* $\boldsymbol{\lambda}$):

$$\rho(\boldsymbol{x}, \boldsymbol{\lambda}) T(\boldsymbol{x}, \boldsymbol{\lambda} \to \boldsymbol{x}', \boldsymbol{\lambda}') = \rho(\boldsymbol{x}', \boldsymbol{\lambda}') T(\boldsymbol{x}', \boldsymbol{\lambda}' \to \boldsymbol{x}, \boldsymbol{\lambda})$$

such that the neMD/MC transition probability is:

$$\begin{aligned} T(\boldsymbol{x}, \boldsymbol{\lambda} \to \boldsymbol{x}', \boldsymbol{\lambda}') &= \min\left[1, \frac{\rho(\boldsymbol{x}', \boldsymbol{\lambda}')}{\rho(\boldsymbol{x}, \boldsymbol{\lambda})}\right] \\ &= \min\left[1, e^{-\beta W} 10^{-\Delta n \mathrm{pH}}\right] \end{aligned}$$

(If you'd like, MD uses the probability $T(\boldsymbol{x} \to \boldsymbol{x}') = 1$.)

Important considerations

- How long should I sample the equilibrium stage?
- How long should I sample the nonequilibrium stage? (the "switch time," τ_{switch})
- Rejecting a nonequilibrum trajectory is expensive, how can we avoid doing that so much?

The two-step “inherent” pK_a algorithm

$$T(\boldsymbol{x}, \boldsymbol{\lambda} \rightarrow \boldsymbol{x}', \boldsymbol{\lambda}') = T^{(i)}(\boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}')\, T^{(s)}(\boldsymbol{x} \rightarrow \boldsymbol{x}' | \boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}')$$

$$T^{(i)}(\boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}') = \min\left[1, 10^{\mathrm{p}K_\mathrm{a}^{(\mathrm{i})}(\boldsymbol{\lambda}, \boldsymbol{\lambda}') - \Delta n \mathrm{pH}}\right]$$

- neMD/MC can be split into *two* parts
  1. $T^{(i)}$ – only depends on $\boldsymbol{\lambda}$ and the pH – CHEAP
  2. $T^{(s)}$ – depends on the switch (W) – COSTLY

Chen & Roux *J Chem Theory Comput*, **2015**; Radak, *et al.* *J Chem Theory Comput*, **2017**

The two-step "inherent" pK_a algorithm

$$T(\boldsymbol{x}, \boldsymbol{\lambda} \rightarrow \boldsymbol{x}', \boldsymbol{\lambda}') = T^{(i)}(\boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}') T^{(s)}(\boldsymbol{x} \rightarrow \boldsymbol{x}' | \boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}')$$

$$T^{(i)}(\boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}') = \min\left[1, 10^{\mathrm{p}K_\mathrm{a}^{(\mathrm{i})}(\boldsymbol{\lambda}, \boldsymbol{\lambda}') - \Delta n \mathrm{pH}}\right]$$

- neMD/MC can be split into *two* parts
  1. $T^{(i)}$ – only depends on $\boldsymbol{\lambda}$ and the pH – CHEAP
  2. $T^{(s)}$ – depends on the switch (W) – COSTLY
- Effort is shifted by estimating a parameter, p$K_\mathrm{a}^{(\mathrm{i})}$
- Optimal efficiency achieved for exact pK_a

Chen & Roux *J Chem Theory Comput*, **2015**; Radak, *et al. J Chem Theory Comput*, **2017**

The two-step "inherent" pK_a algorithm

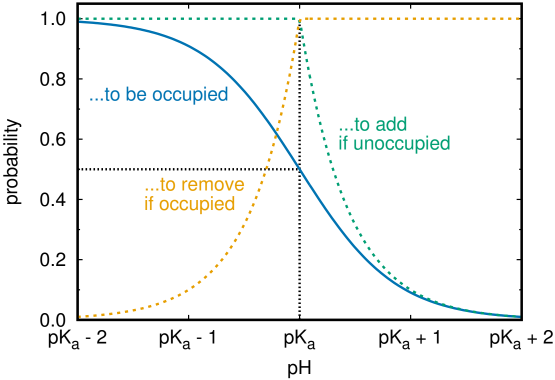
$$T(\boldsymbol{x}, \boldsymbol{\lambda} \to \boldsymbol{x}', \boldsymbol{\lambda}') = T^{(i)}(\boldsymbol{\lambda} \to \boldsymbol{\lambda}') T^{(s)}(\boldsymbol{x} \to \boldsymbol{x}' | \boldsymbol{\lambda} \to \boldsymbol{\lambda}')$$

$$T^{(i)}(\boldsymbol{\lambda} \to \boldsymbol{\lambda}') = \min\left[1, 10^{\mathrm{p}K_\mathrm{a}^{(\mathrm{i})}(\boldsymbol{\lambda},\boldsymbol{\lambda}') - \Delta n \mathrm{pH}}\right]$$

- neMD/MC can be split into *two* parts
  1. $T^{(i)}$ – only depends on $\boldsymbol{\lambda}$ and the pH – CHEAP
  2. $T^{(s)}$ – depends on the switch (W) – COSTLY
- Effort is shifted by estimating a parameter, p$K_\mathrm{a}^{(\mathrm{i})}$
- Optimal efficiency achieved for exact pK_a
- Dramatically improved performance on wide pH ranges!

Chen & Roux *J Chem Theory Comput*, **2015**; Radak, *et al.* *J Chem Theory Comput*, **2017**

A graphical view of the inherent pK_a algorithm

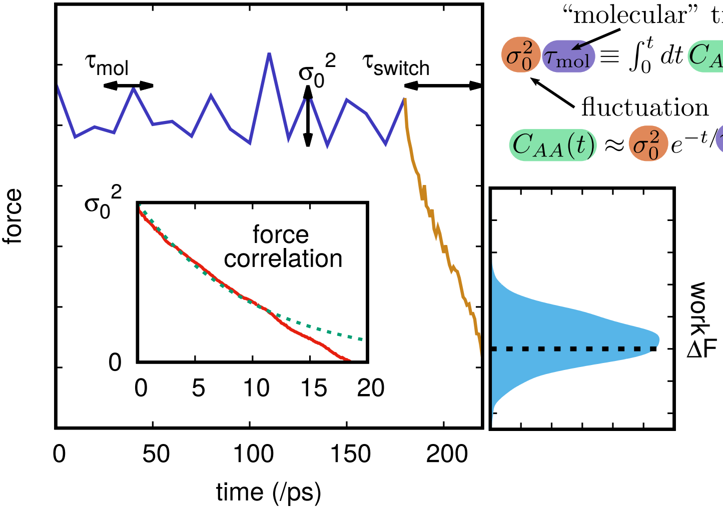


- It's silly to try to add/remove protons to/from acidic/basic residues at high/low pH
- Transitions are proposed in proportion to the estimated population.

What about after we've proposed a switch?

- A short switch will not change much and likely be rejected.
- A long switch is expensive (limit of a single switch – BAD).
- Since the switch success depends on the work, let's analyze that.

Work and force fluctuations – a typical neMD/MC cycle



“molecular” timescale

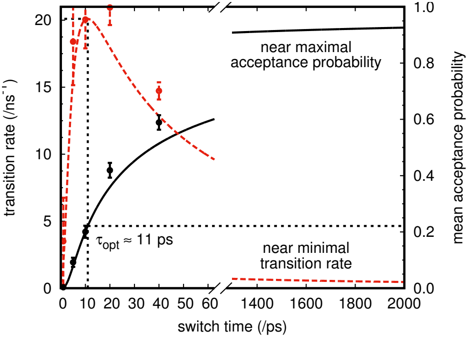
$$\sigma_0^2 \tau_{\mathrm{mol}} \equiv \int_0^t dt\, C_{AA}(t)$$

fluctuation

$$C_{AA}(t) \approx \sigma_0^2 e^{-t/\tau_{\mathrm{mol}}}$$

Radak & Roux *J Chem Phys*, **2016**

Theoretical and Empirical Performance Analysis



$$\tau_{\text{opt}} \leq \frac{\sigma_0^2 \tau_{\text{mol}}}{2.83475}$$

$$\overline{P_{\text{opt}}} \leq 23.4\%$$

$$k_{\text{opt}} \equiv \frac{\overline{P_{\text{opt}}}}{\tau_{\text{opt}}} \geq \frac{0.66318}{\sigma_0^2 \tau_{\text{mol}}}$$

- High acceptance is good, but not naively optimizable
- The transition rate can be optimized within constraints

Radak & Roux *J Chem Phys*, **2017**; Radak, *et al.* *J Chem Theory Comput*, **2017**

Main take-aways for the algorithm

- Estimating/updating the inherent pK_a is very helpful for efficiency.

- The best choice of switch time depends on the particular dynamics – values near 10–20 ps are reasonable. Look for acceptance rates $\sim$20%.

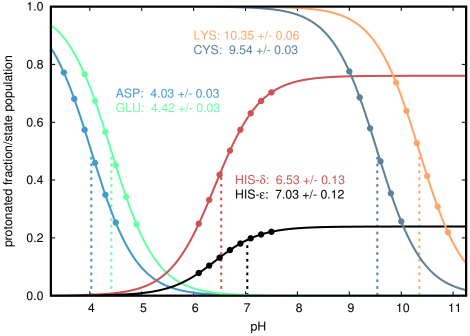
- The length of each cycle depends largely on the number of residues. Values near 0.1–1 ps should be reasonable.

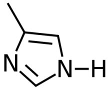
NAMD Constant pH: Features and Keywords

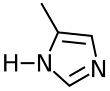
- Flexible Tcl interface `source lib/namdcph/namdcph.tcl`
- PSF build procedure is unchanged (automated `psfgen`)
- Implemented with PME and full electrostatics
- No GPU yet - depends on alchemy
- Companion analysis script `cphanalyze`

```
parameters    par_cph36_prot.prm
cphConfigFile conf_cph36_prot.json
topology      top_cph36_prot.rtf
pH 7.0
cphNumstepsPerSwitch 7500 ;# run 7500 steps per switch
cphRun 500 10 ;# run 10 cycles of 500 MD steps
```

CHARMM36: Reference amino acids are well-reproduced

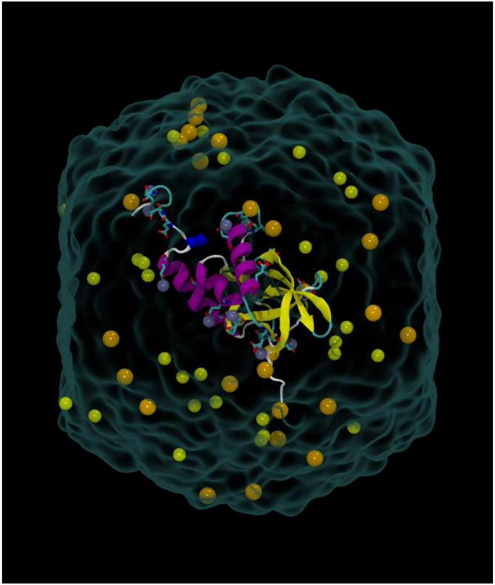




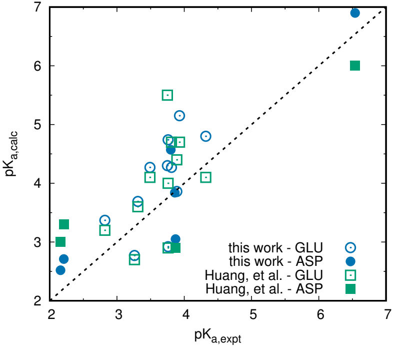


- Adjustments to force field enforce empirical reference values
- Implicitly model solvated proton and bond energy effects
- Bonus:
  accurate reproduction of tautomeric ratios!

Staph nuclease (SNase) - A constant pH benchmark



Benchmarking of SNase pK_a values

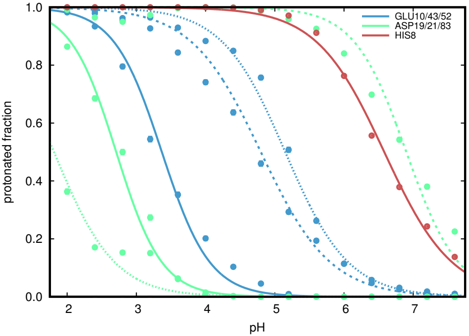


Radak, *et al. J Chem Theory Comput*, **2017**;
Huang, *et al. J Chem Theory Comput*, **2016**

- Good correlation with measured values for carboxylates
- Bonus:
  estimates for HIS

residue		this work
HIS	8	6.58 (0.29)
	121	5.19 (0.16)

Output and Analysis



- Normal usage requires multiple pH values (“titration curves”)
- `cphanalyze` can...
  - boost performance with WHAM
  - extract pK_a from Hill fitting

A Brief WHAM Primer

Consider $k = 1, \ldots, M$ pH values with N_k samples per value ($N = \sum_{k=1}^{M} N_k$) and site occupancies $\boldsymbol{\lambda}_t$ at each timestep.

$$P_\chi(\mathrm{pH}) = \frac{1}{N} \sum_{t=1}^{N} w_t(\mathrm{pH}) \chi(\boldsymbol{\lambda}_t),$$

$$w_t(\mathrm{pH}) \equiv \left[\sum_{k=1}^{M} \frac{N_k}{N} e^{f(\mathrm{pH}_k) - f(\mathrm{pH})} 10^{-(\mathrm{pH}_k - \mathrm{pH}) n_t} \right]^{-1}$$

- Energy difference only depends on the proton count, n_t
- Can compute probability for any indicator, $\chi(\boldsymbol{\lambda}_t)$
- Permits consistent interpolation/extrapolation

Output and Analysis

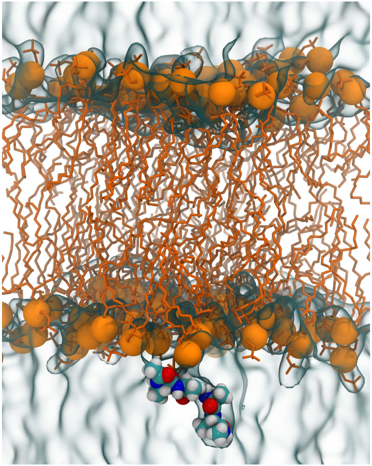
- New output: `cphlog`
- New checkpoint files: `psf/pdb`, `cphrst`

```
parameters    par_cph36_prot.prm
cphConfigFile conf_cph36_prot.json
topology      top_cph36_prot.rtf

structure      $oldOutputName.psf
coordinates    $oldOutputName.pdb
cphRestartFile $oldOutputName.cphrst

cphRun 500 10
```

Example `cphlog`:

```
#pH 4.0
#PROA:129:ASP PROA:141:GLU PROA:142:HIS
 PROA:145:ASP PROA:150:LYS PROA:161:GLU
 PROA:162:ASP
     1 0 0 1 0 1 1 0 0 1 1 1 0 0 0 0
     2 0 0 1 0 1 1 0 0 1 1 1 0 0 0 0
     3 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0
     4 0 0 0 0 1 1 0 0 1 1 1 1 0 0 0
```

Membranes... things get weird

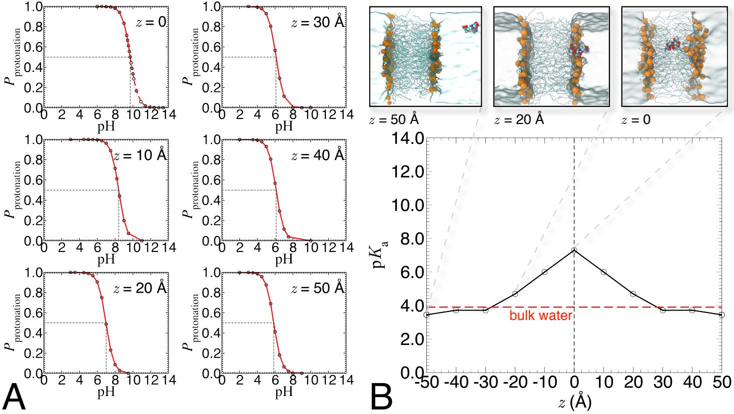


- A fluctuating net charge is tricky with PME.

$$E = E(\boldsymbol{x}) + \mathcal{O}\left(\frac{Q}{V\epsilon}\right)$$

- Membrane systems have a lower than usual mean dielectric and smaller aqueous volume.
- Multiple options to correct this, but all require care.

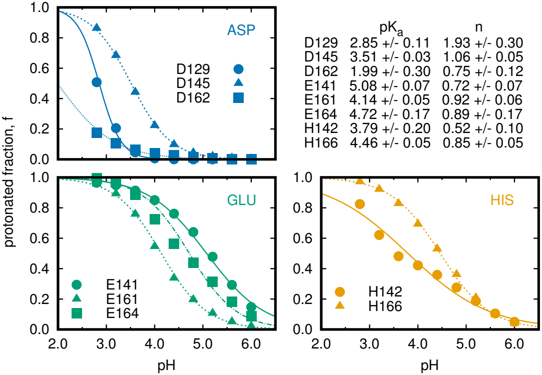
Membranes... things get weird



- Significant shifts due to low dielectric region.
- Effective pH changes by ∼2 units!

Other cautions: WHAM versus “naive” data analysis

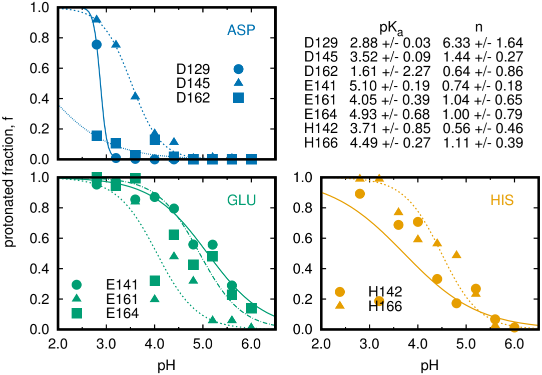
- WHAM is effectively a Bayesian framework with prior assumption that
  1. the data is i.i.d.
  2. the data is Boltzmann distributed
- This may be misleading when convergence is poor!



	pK_a	n
D129	2.85 +/- 0.11	1.93 +/- 0.30
D145	3.51 +/- 0.03	1.06 +/- 0.05
D162	1.99 +/- 0.30	0.75 +/- 0.12
E141	5.08 +/- 0.07	0.72 +/- 0.07
E161	4.14 +/- 0.05	0.92 +/- 0.06
E164	4.72 +/- 0.17	0.89 +/- 0.17
H142	3.79 +/- 0.20	0.52 +/- 0.10
H166	4.46 +/- 0.05	0.85 +/- 0.05

Other cautions: WHAM versus “naive” data analysis

- WHAM is effectively a Bayesian framework with prior assumption that
  1. the data is i.i.d.
  2. the data is Boltzmann distributed
- This may be misleading when convergence is poor!



	pK_a	n
D129	2.88 +/- 0.03	6.33 +/- 1.64
D145	3.52 +/- 0.09	1.44 +/- 0.27
D162	1.61 +/- 2.27	0.64 +/- 0.86
E141	5.10 +/- 0.19	0.74 +/- 0.18
E161	4.05 +/- 0.39	1.04 +/- 0.65
E164	4.93 +/- 0.68	1.00 +/- 0.79
H142	3.71 +/- 0.85	0.56 +/- 0.46
H166	4.49 +/- 0.27	1.11 +/- 0.39

Concluding Remarks/Future Directions

1. You can run constant-pH MD today on globular protein systems.
   - Consider using for systems with large numbers of (unknown) states
   - Can also use this as an alternative for structure based assignment
2. Things we are working on:
   - Performance improvements in alchemy – CUDA support
   - Better support for membrane systems
   - Better visualization support in VMD
   - More automated inherent *pKa* selection
   - pH replica exchange
3. Things we would like to work on:
   - `psfgen` improvements – support for Drude
   - Support for other force fields
   - More general small molecule support