

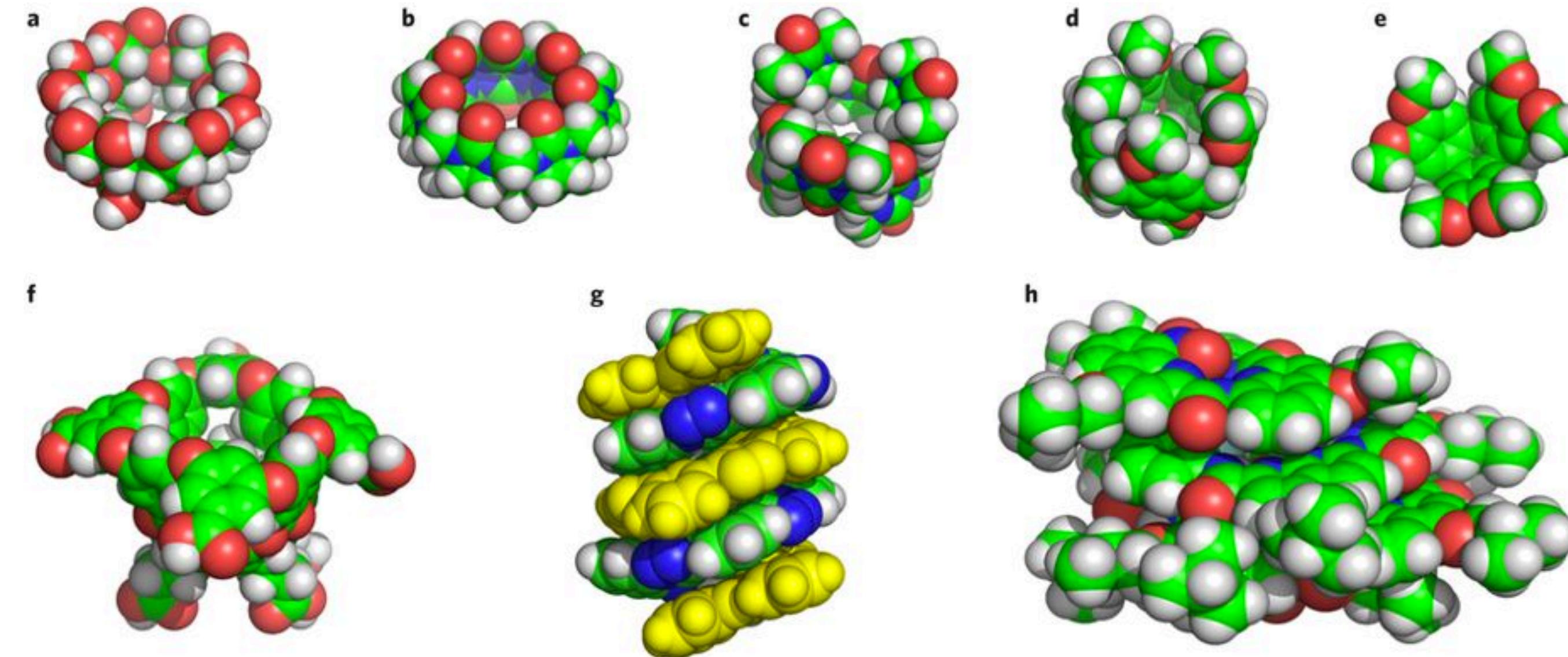


Binding free energy calculations using the attach-pull-release method

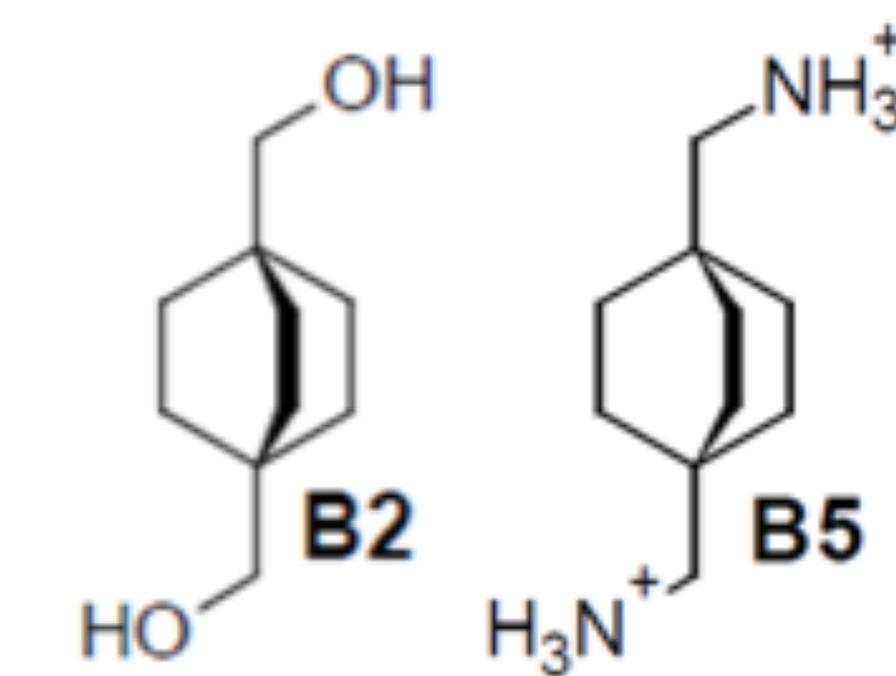
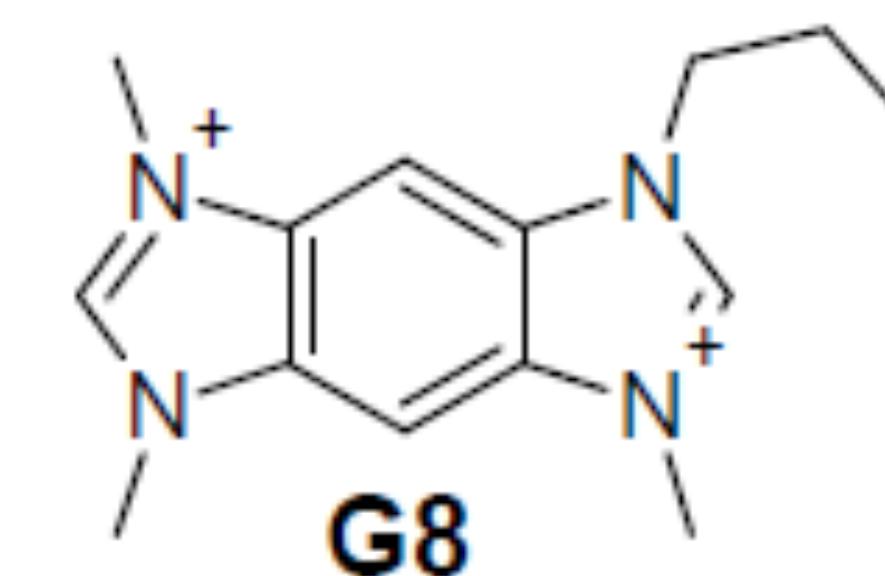
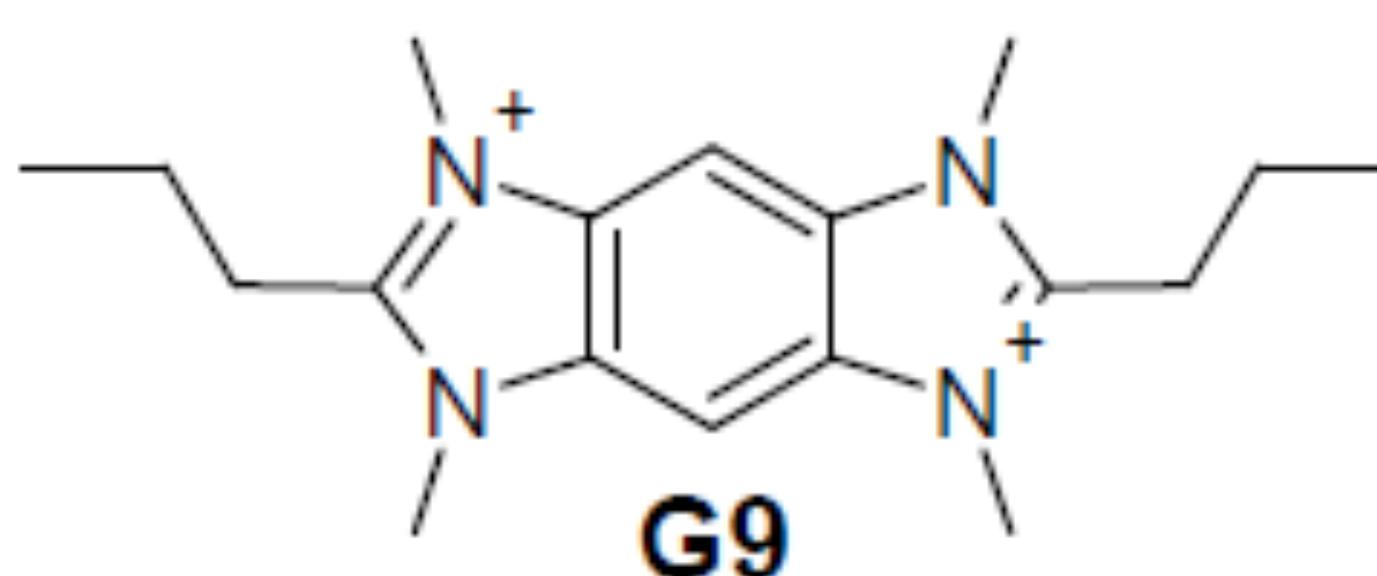
David Slochower, Niel Henriksen, and Michael K. Gilson
UCSD

Amber developers' meeting, March 28-30th 2019

Host-guest pairs are ideal binding model systems.

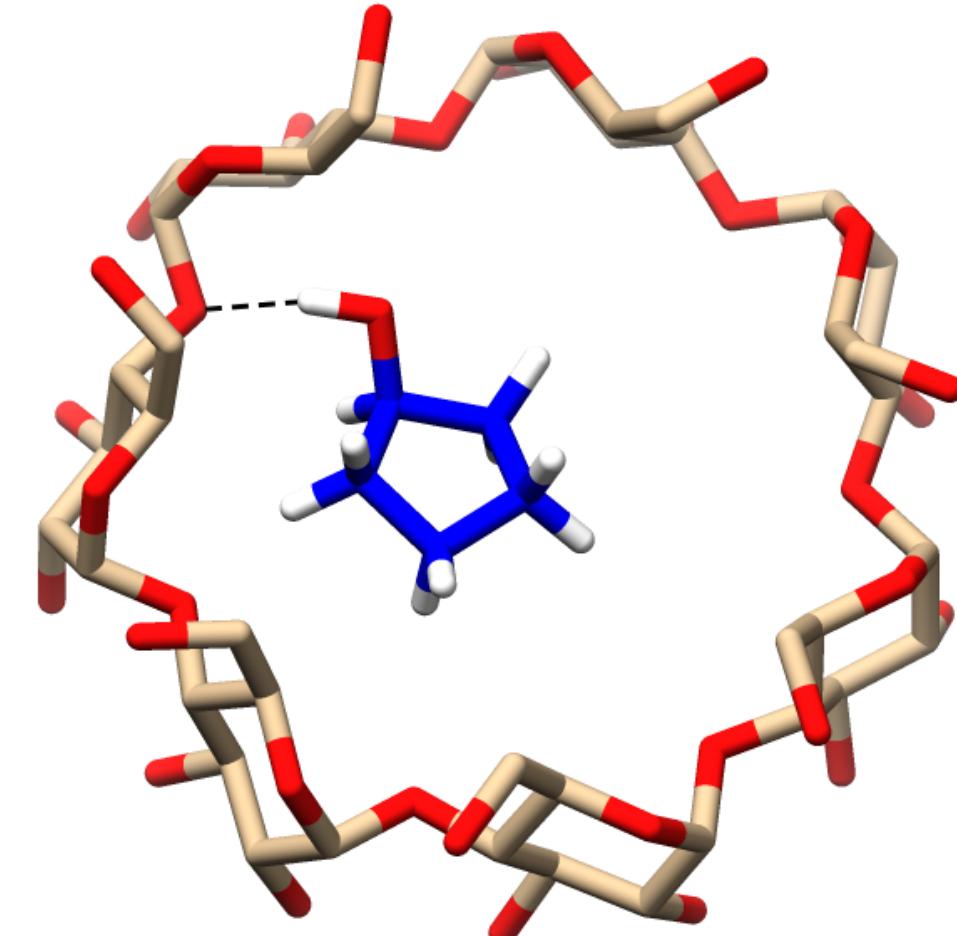


Hosts are cavity-containing molecules (Gibb's Deep Cavity Cavitands)

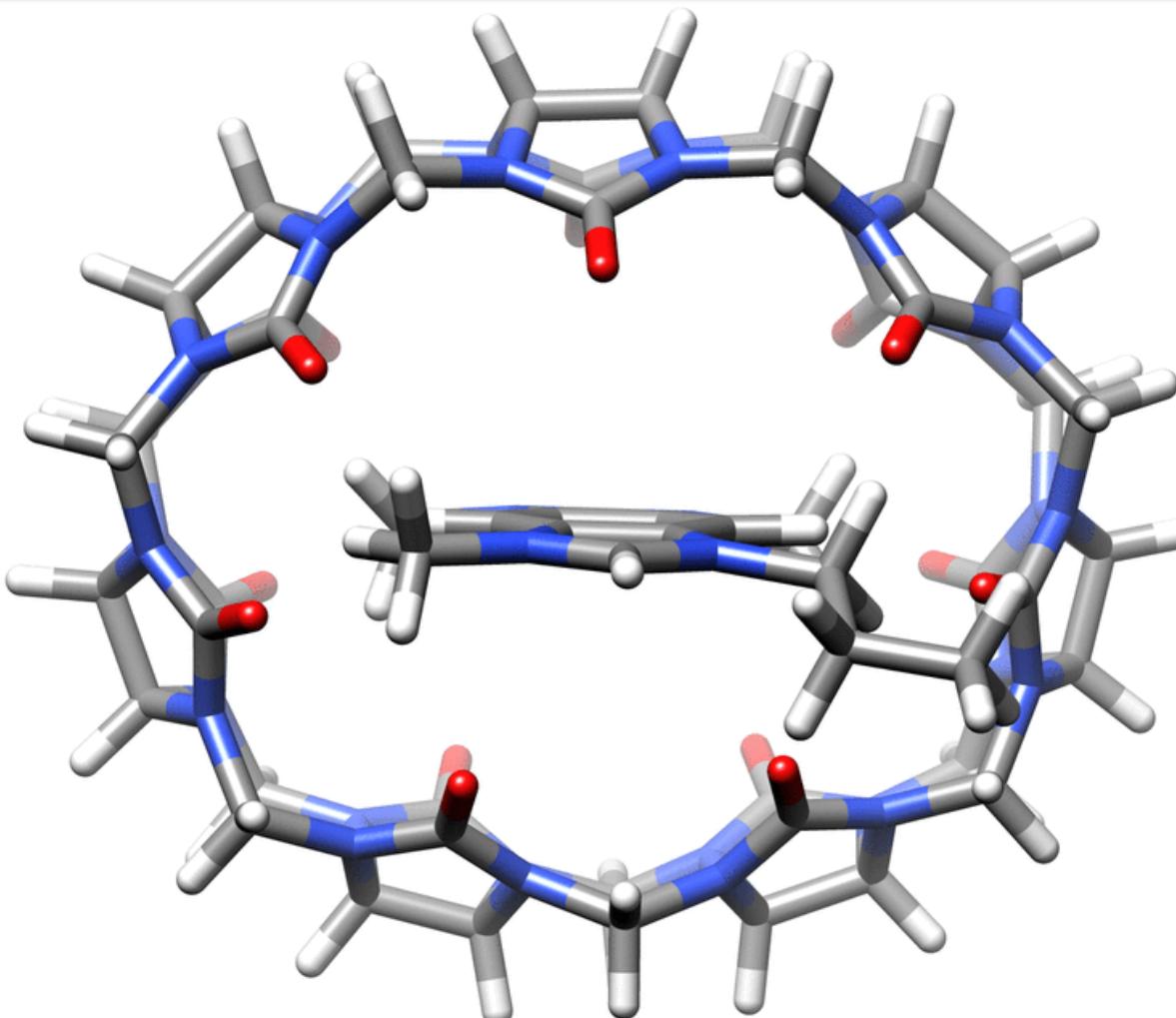


Guests are drug-like small molecules

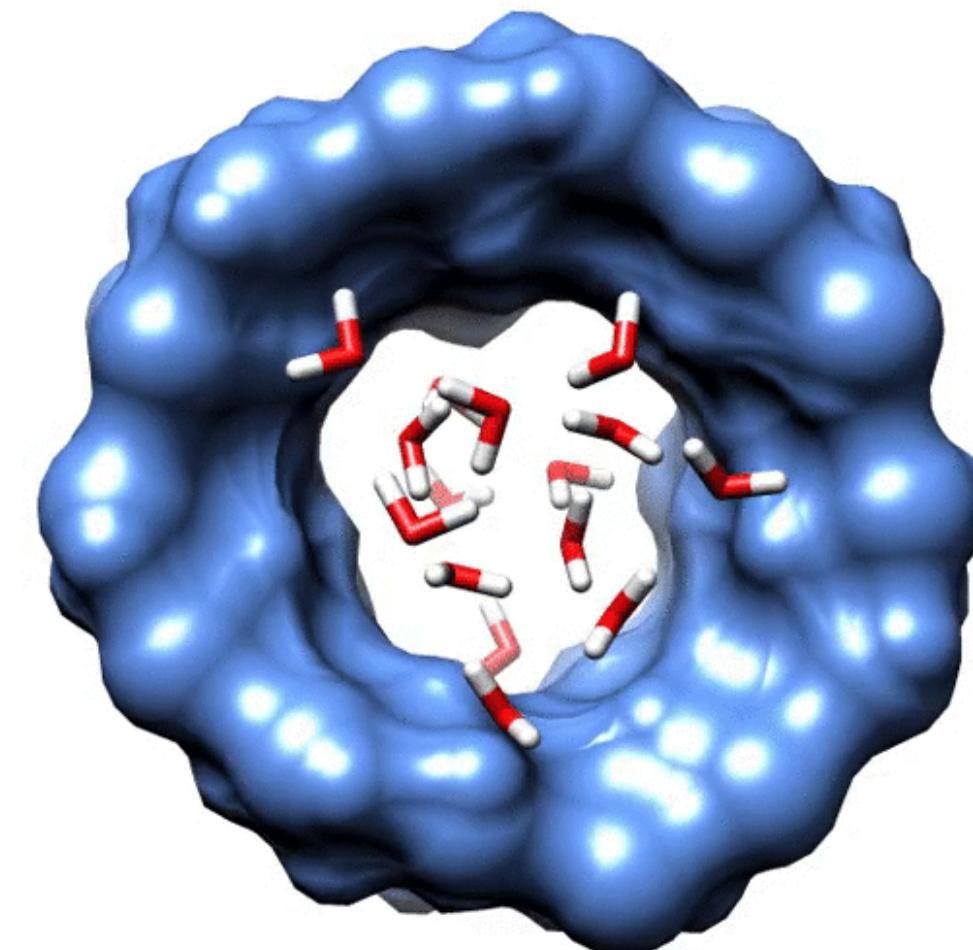
Host-guest pairs are ideal binding model systems.



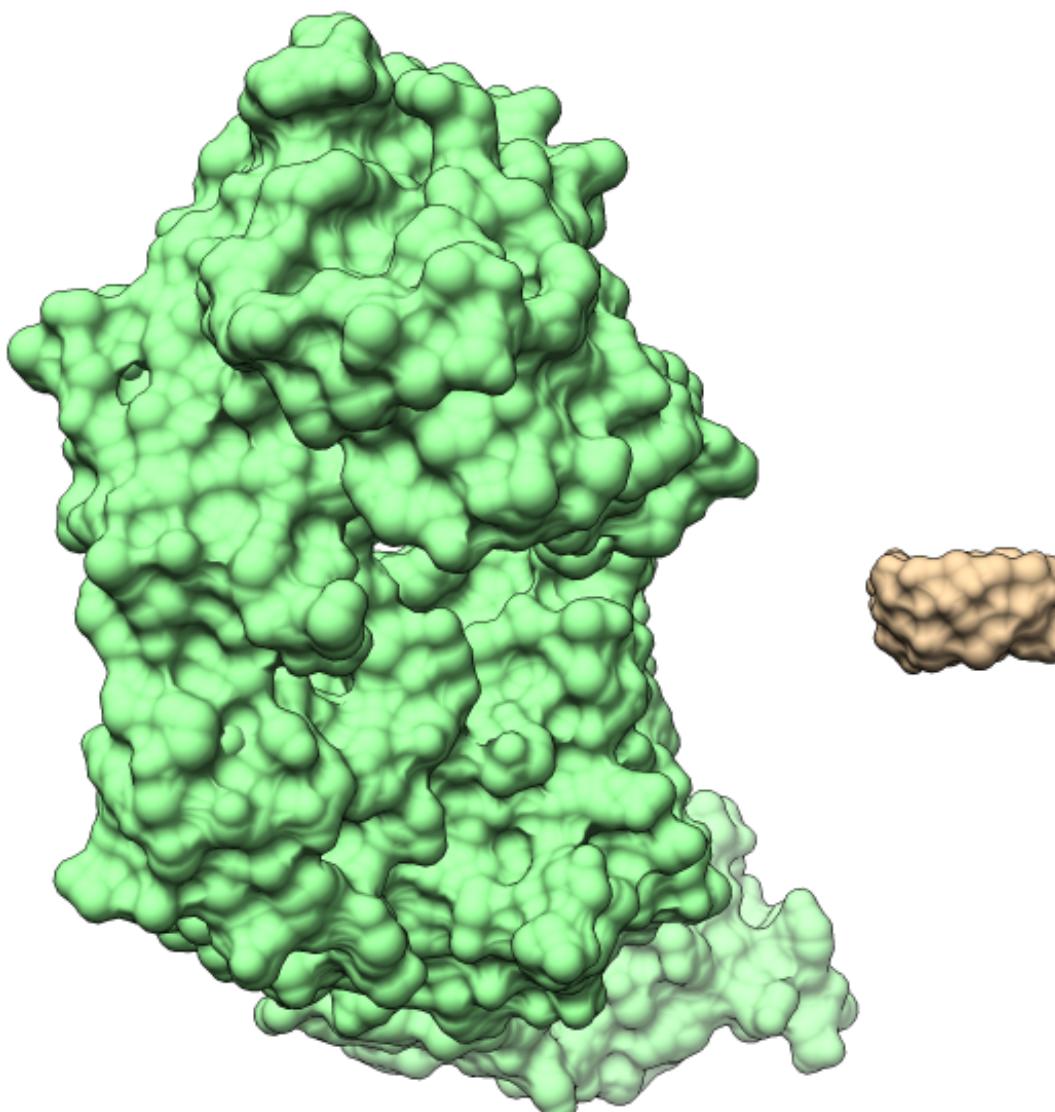
Hydrophilic & hydrophobic interactions



Conformational changes upon binding

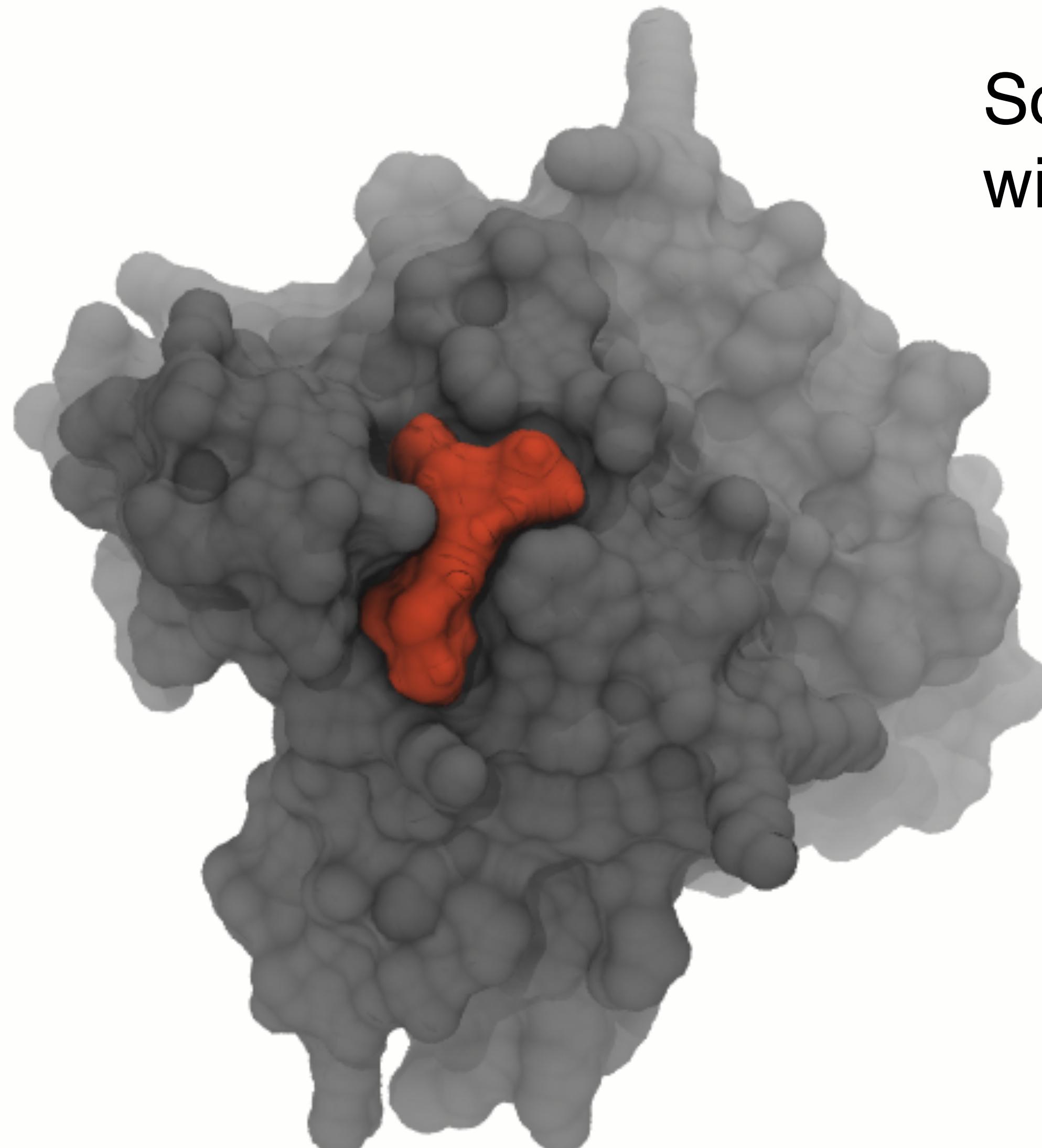


Regions of structured solvent



Simulations are quicker and can run longer

This approach works on proteins too, but it is trickier.

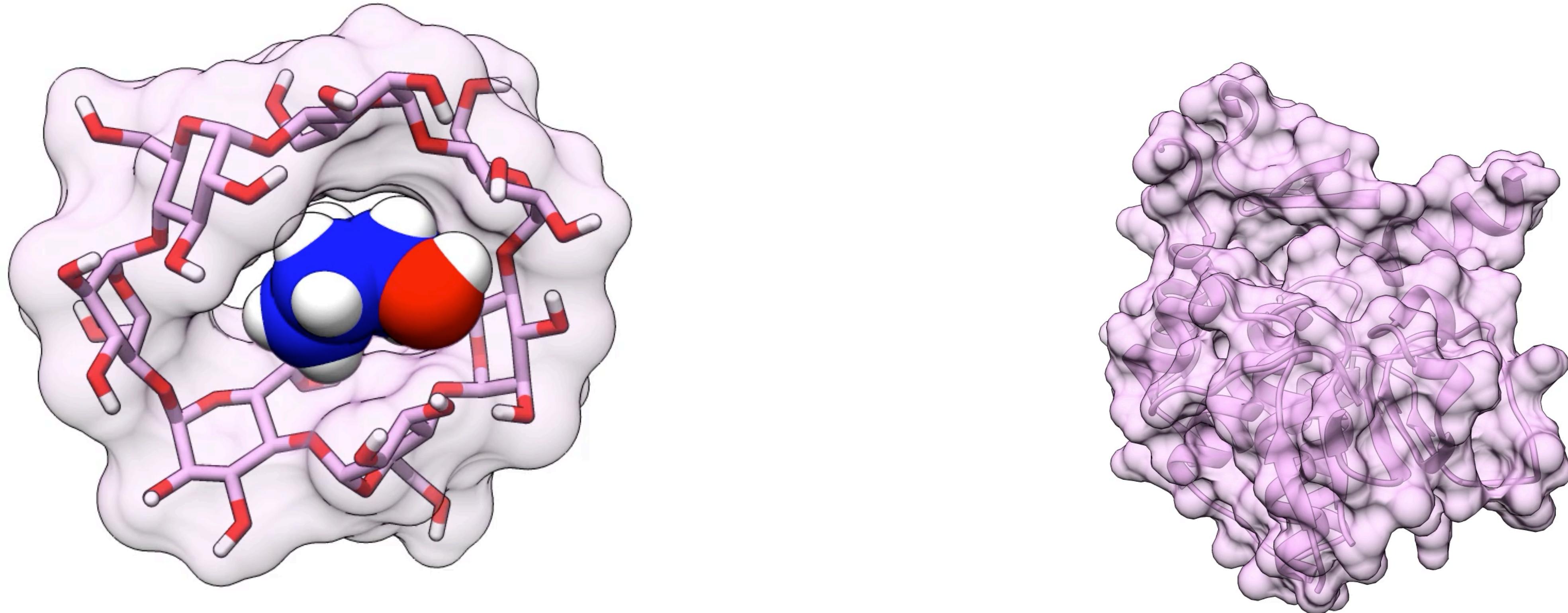


Sometimes it's difficult to access the binding site without moving something else out of the way first.

We develop new methods of computing binding free energies: computational calorimetry.

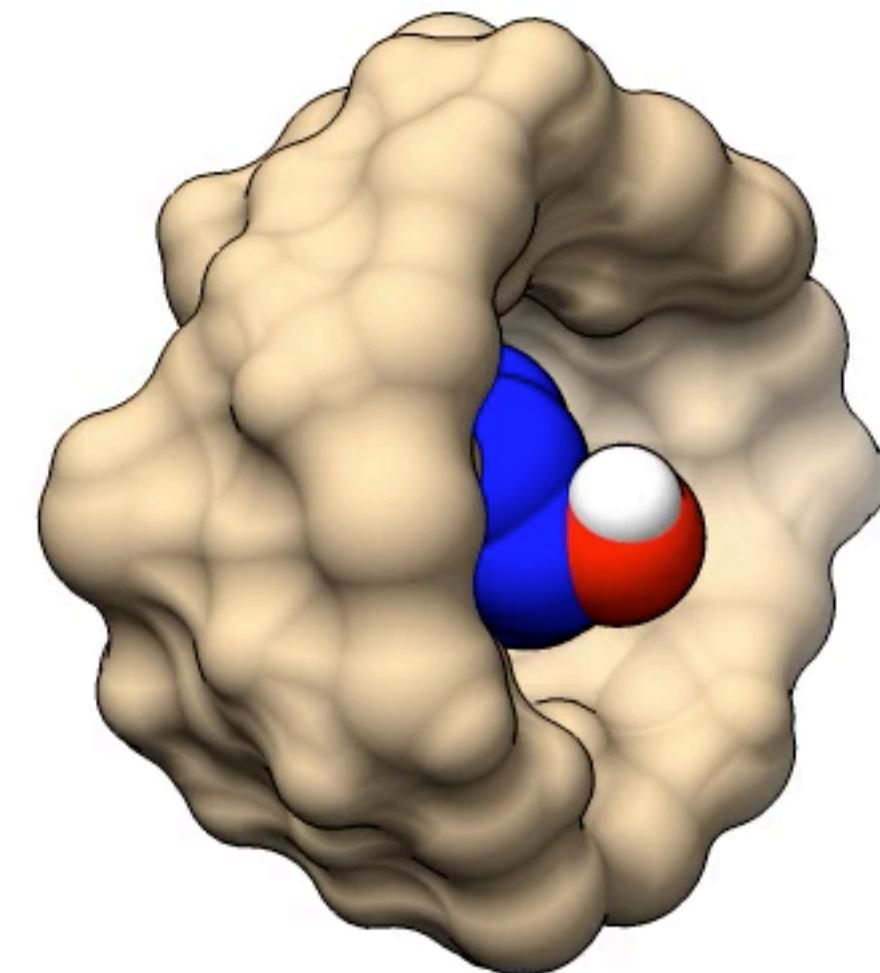
The binding affinity is the reversible work of pulling the ligand out of the host.

We compute binding free energies via a method called attach-pull-release.



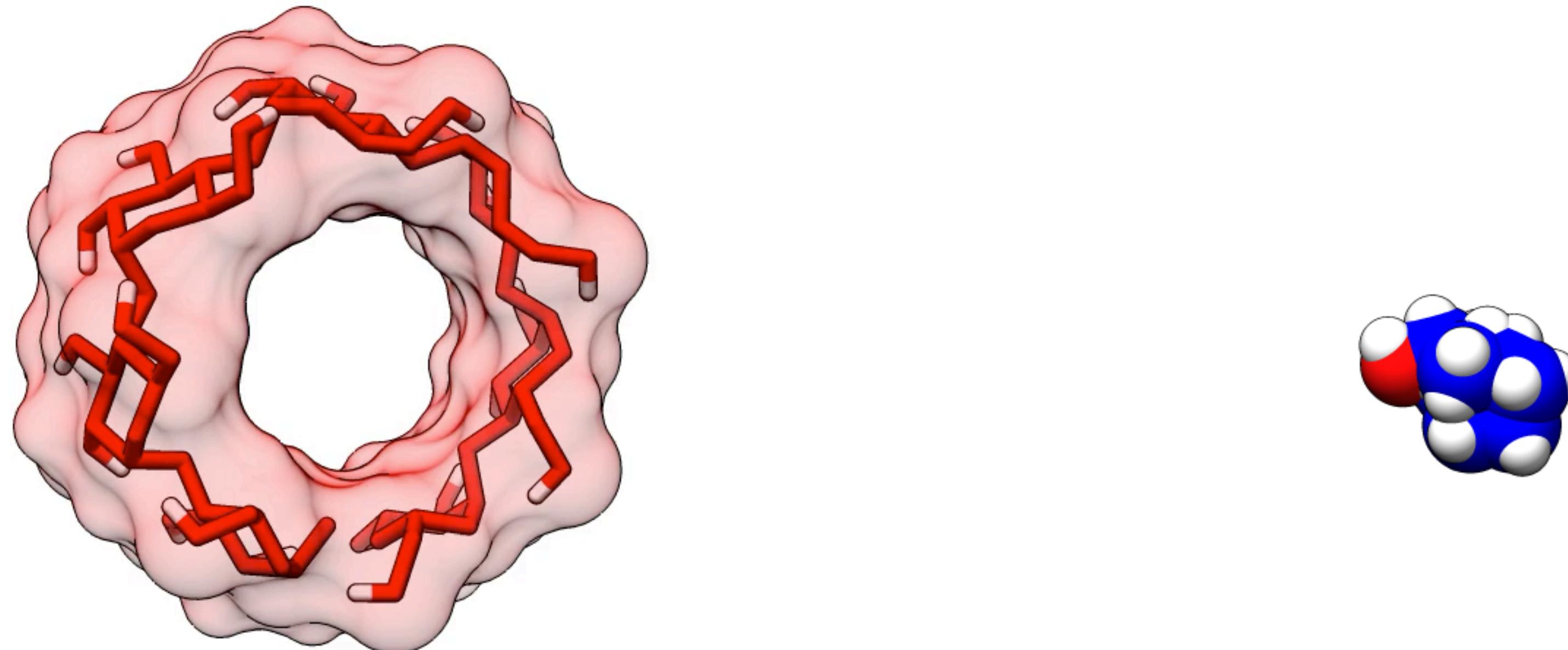
$$\Delta G^\circ = -(W_{\text{attach}} + W_{\text{pull}} + W_{\text{release}})$$

We compute binding free energies via a method called attach-pull-release.



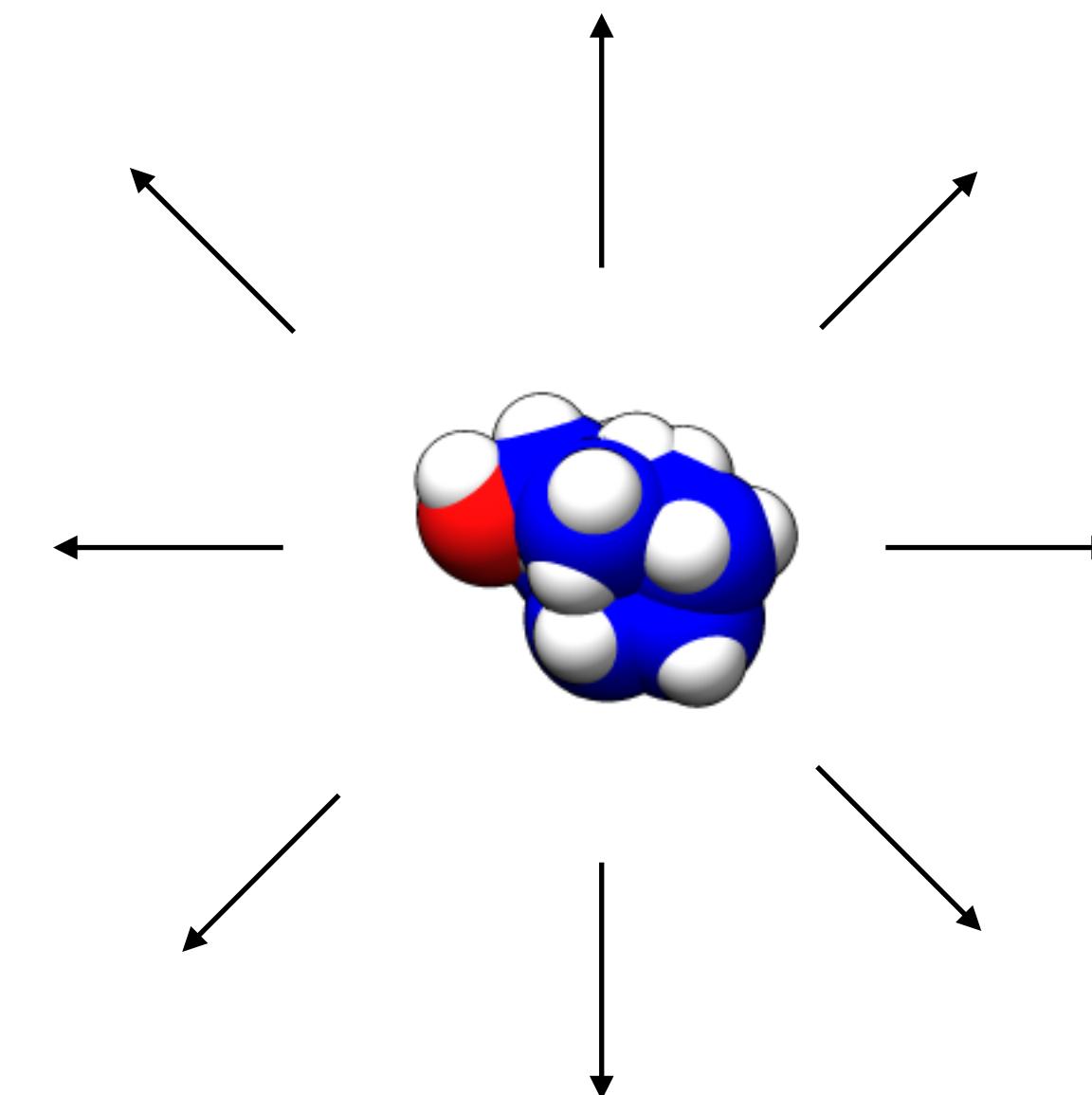
$$\Delta G^\circ = -(W_{\text{attach}} + W_{\text{pull}} + W_{\text{release}})$$

We compute binding free energies via a method called attach-pull-release.



$$\Delta G^\circ = -(W_{\text{attach}} + W_{\text{pull}} + W_{\text{release}})$$

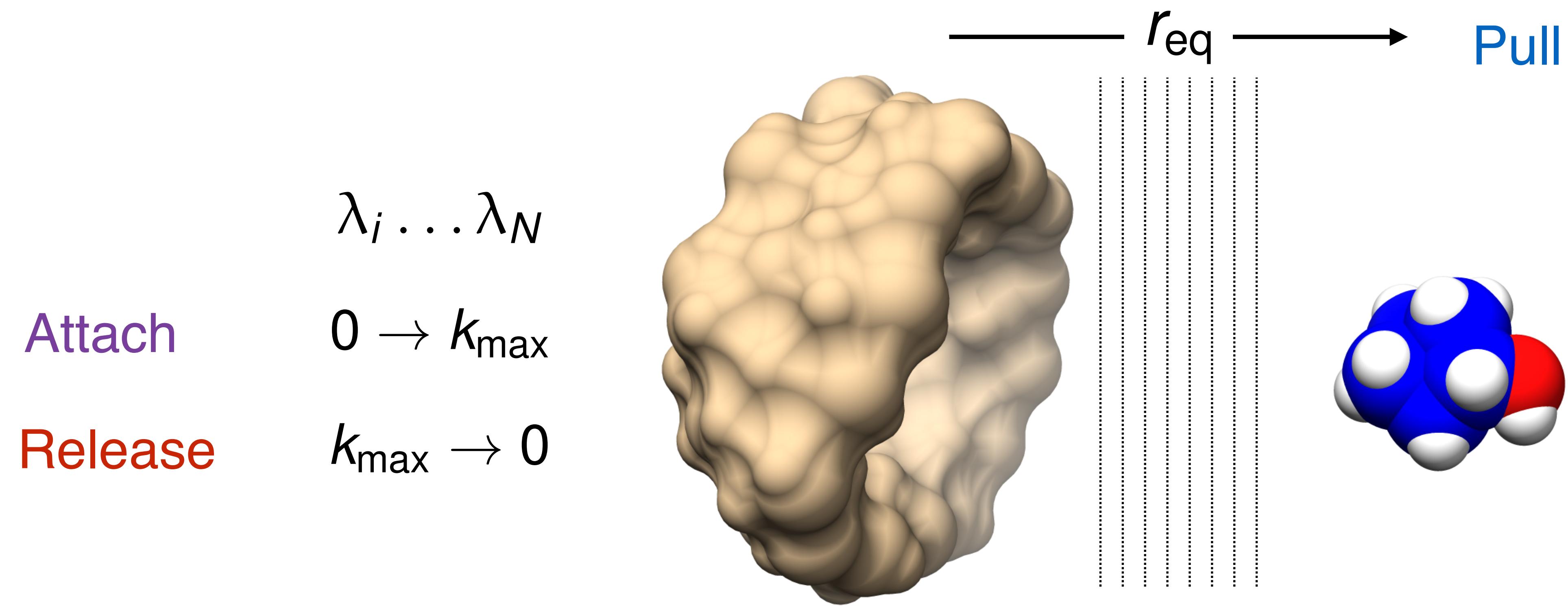
We compute binding free energies via a method called attach-pull-release.



$$W_{\text{release-std}}(r, \theta, \phi, a, b, c)$$

We compute binding free energies via a method called attach-pull-release.

Each window is independently built and simulated.



We use the work done by the restraints to derive a potential of mean force.

1. Estimate the restraint work in each window

$$\frac{\partial U}{\partial \lambda}$$

2. Sample off the work distributions

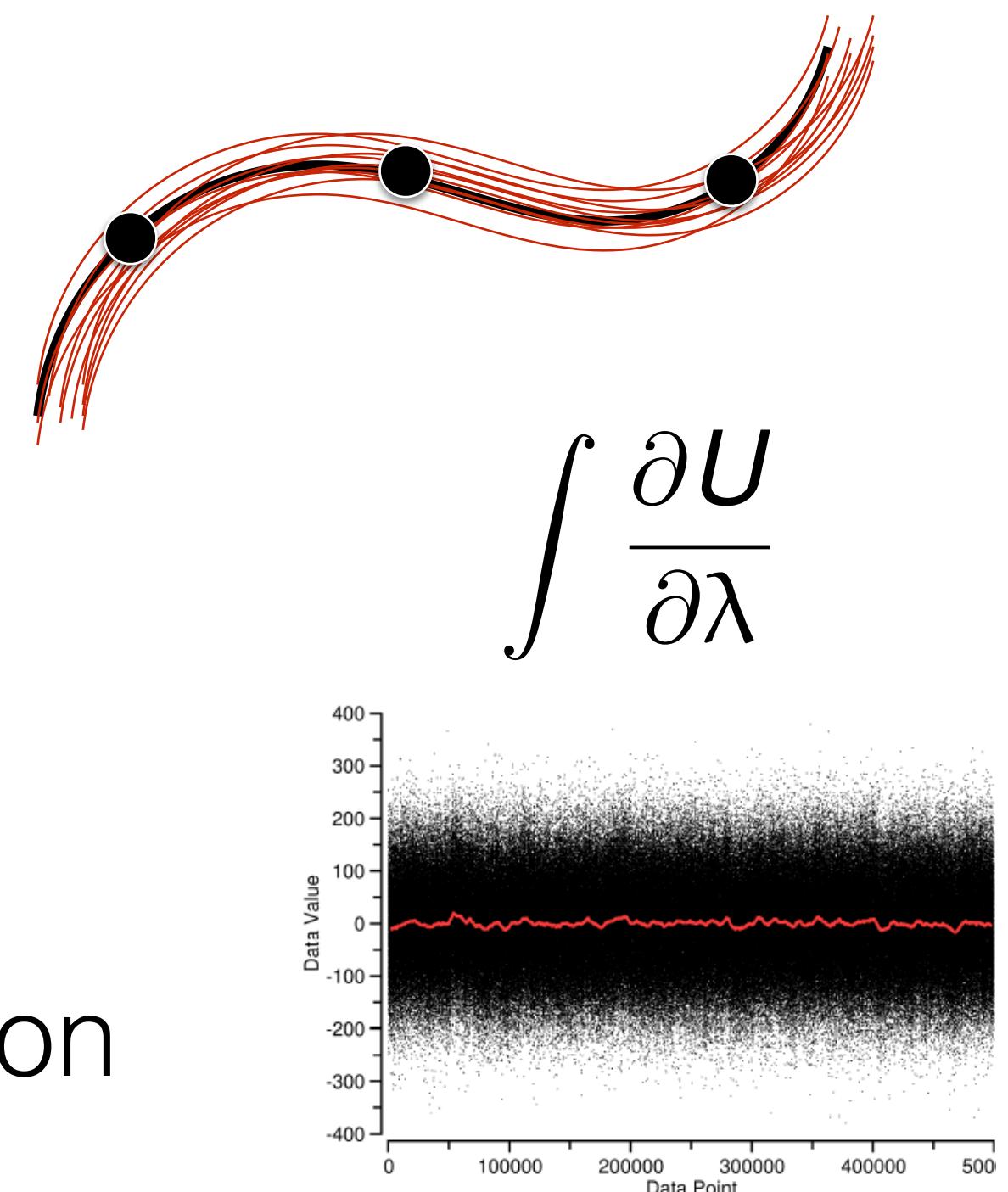
3. Create a spline through the sampled points

4. Integrate the bootstrapped curve

Thermodynamic integration or MBAR

5. Estimate the uncertainty using uncorrelated data

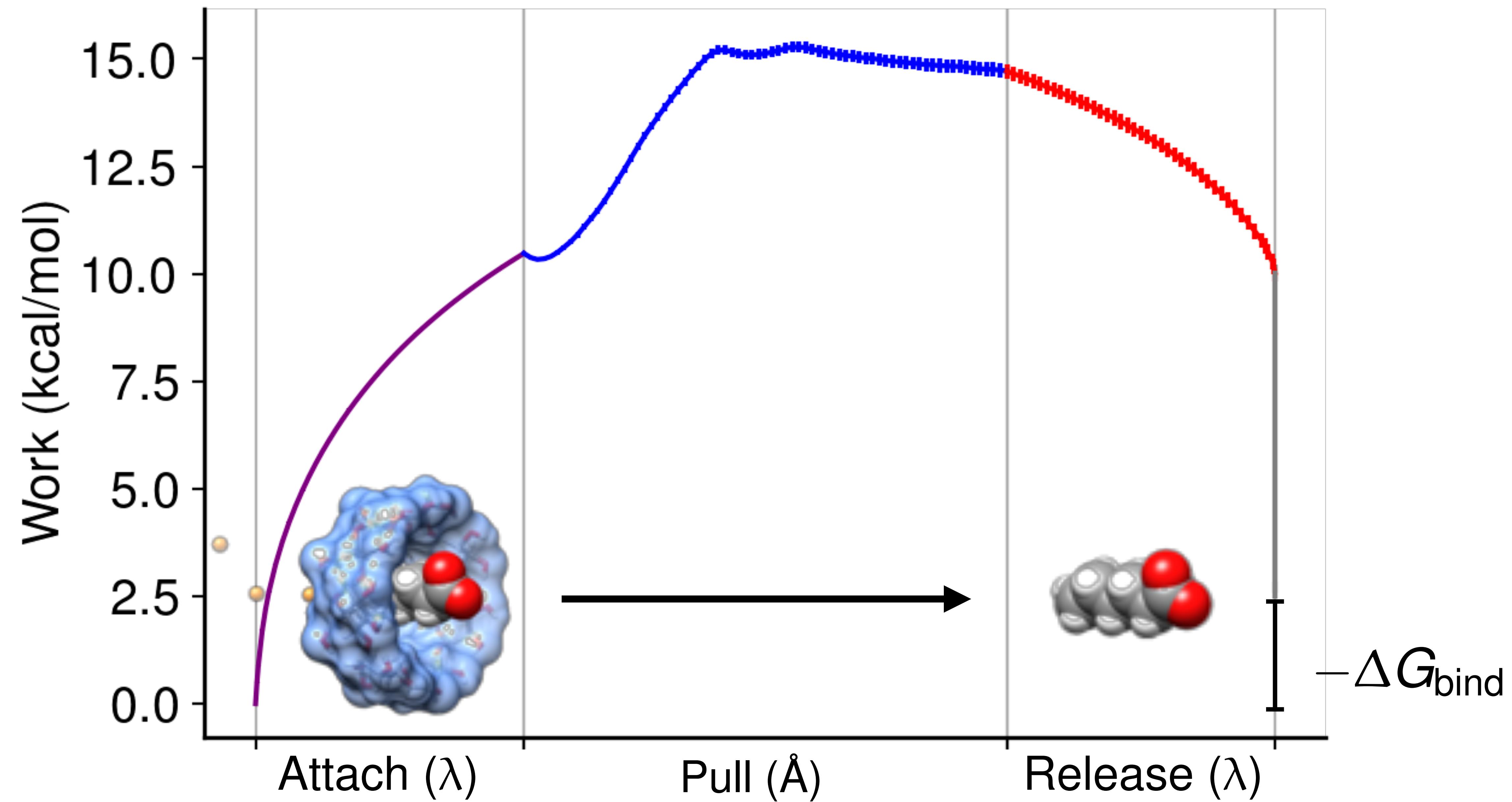
Blocking analysis or statistical inefficiency via autocorrelation



N. M. Henriksen, A. T. Fenley, M. K. Gilson, Computational Calorimetry: High-Precision Calculation of Host-Guest Binding Thermodynamics. *J. Chem. Theory Comput.* **11**, 4377–4394 (2015).

A. T. Fenley, N. M. Henriksen, H. S. Muddana, M. K. Gilson, Bridging Calorimetry and Simulation through Precise Calculations of Cucurbituril-Guest Binding Enthalpies. *J. Chem. Theory Comput.* **10**, 4069–4078 (2014).

We compute binding free energies via a method called attach-pull-release.





(Note: These tutorials are meant to provide illustrative examples of how to use AMBER to carry out molecular dynamics simulations. Use them only as a guide for your own simulations.)

Copyright Jian Yin, Niel M. Henriksen, David R. Slochower & Michael K. Gilson 2016

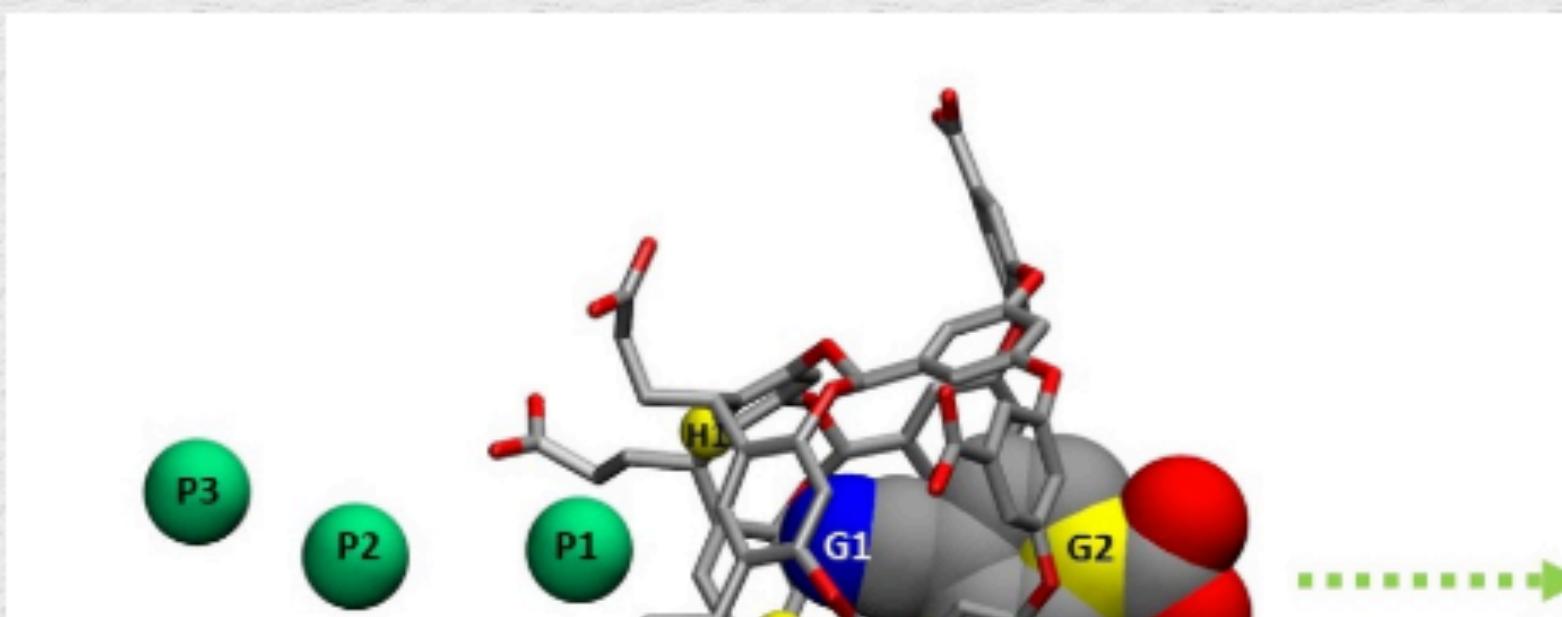
AMBER Advanced Tutorial 29

Computing Binding Free Energy and Binding Enthalpy Using the Attach-Pull-Release (APR) Method Example

By Jian Yin, Niel M. Henriksen, David R. Slochower and Michael K. Gilson

Introduction

This tutorial demonstrates how to compute the binding free energies and binding enthalpies of a host-guest system with explicit water using the attach-pull-release (APR) approach and AMBER suite of program. In the APR calculations, molecular dynamics (MD) simulations are used to sample a free energy path in a series of independent windows, in which the substrate is pulled from the binding cavity by imposed restraints. Compared to nonequilibrium pulling techniques such as steered MD, this approach avoids possible energy dissipation which may induce irreversibility. Another big advantage of using independent windows is that simulations can be carried out in parallel on heterogeneous computing architectures. To run the example in this tutorial, work stations that enable GPU acceleration of AMBER are highly recommended. Please remember to set the environment variable "CUDA_VISIBLE_DEVICES" to prevent multiple simulations from running on a single GPU. It is possible to run the tutorial with pmemd.MPI, pmemd, and even sander, but it will take much longer.



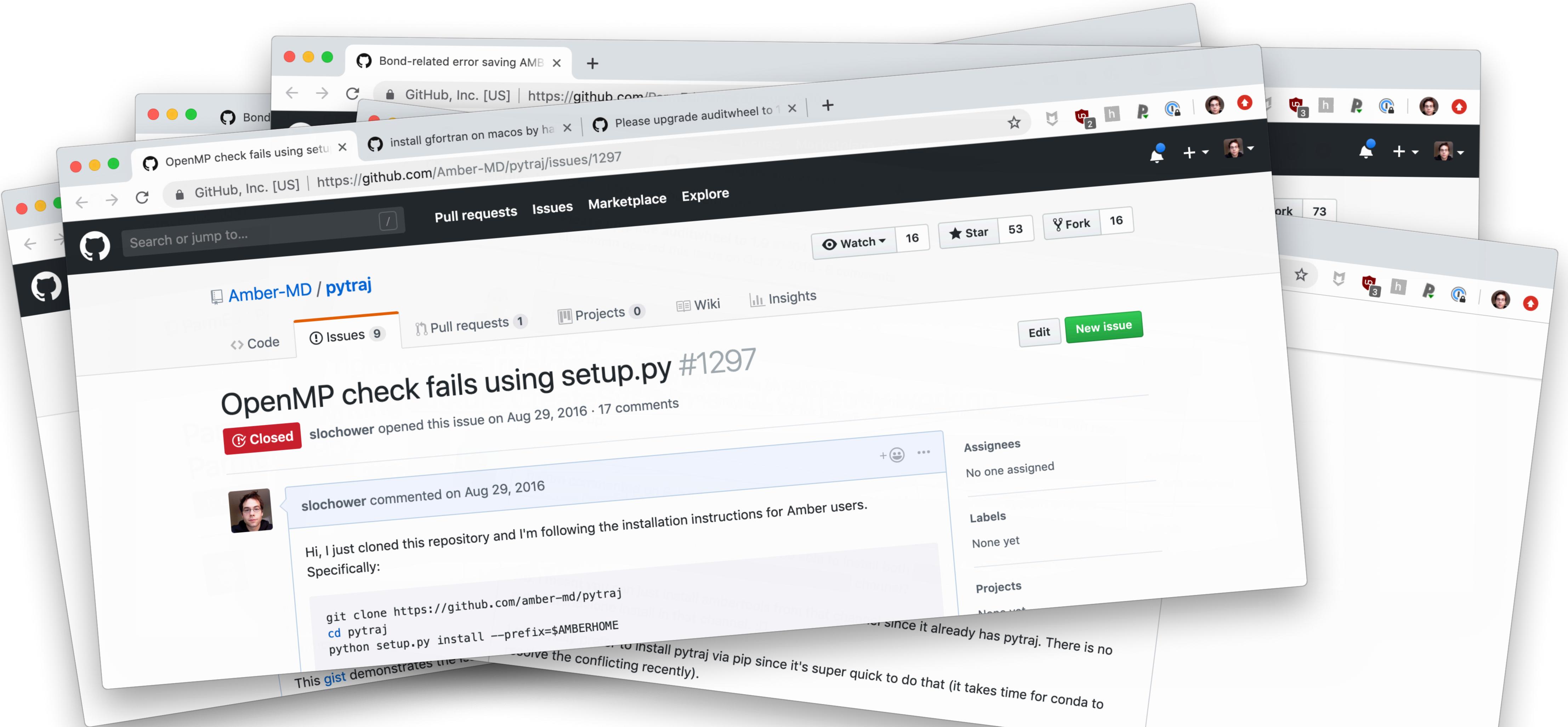
We have implemented attach-pull-release in a Python toolkit called pAPRika.

The screenshot shows the GitHub repository page for `slochower/pAPRika`. The page includes a navigation bar with links to Why GitHub?, Enterprise, Explore, Marketplace, Pricing, a search bar, and sign-in options. Below the header, there are buttons for Watch (2), Star (2), and Fork (3). The main content area features a "Join GitHub today" banner with a "Sign up" button. Below the banner, a description states: "Advanced toolkit for binding free energy calculations". Key statistics are displayed: 383 commits, 7 branches, 3 releases, 2 contributors, and BSD-3-Clause licensing. A "Find File" and "Clone or download" button are also present. The repository's history shows four recent commits:

Author	Commit Message	Date
slochower	Update formatting with isort, autoflake, and black (#99) ...	Latest commit 720187d 14 days ago
devtools	Update RTD (#91)	25 days ago
docs	Update RTD (#91)	25 days ago
paprika	Update formatting with isort, autoflake, and black (#99)	14 days ago

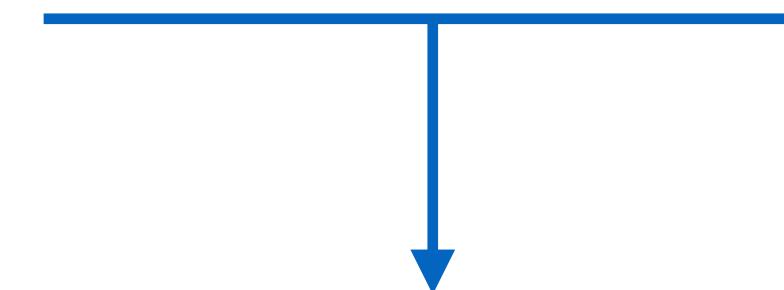
We have implemented attach-pull-release in a Python toolkit called pAPRika.

We wouldn't be able to do this without ParmEd and pytraj!



We have implemented attach-pull-release in a Python toolkit called pAPRika.

Available on conda-forge and working with AMBER or OpenMM (mostly).



Install AmberTools separately
tleap (required)
sander (optional)

<http://ambermd.org/downloads/ambertools/conda/> → ambermd

Thanks Hai Nguyen!

amber_omnia coming...?

We have implemented attach-pull-release in a Python toolkit called pAPRika.

Try me: <http://bit.ly/pAPRika>

(Case sensitive)

So what does it take to run until uncertainties are ~ 0.1 kcal/mol?

Attach and release: 15 windows

Pull: 45 windows

About 2000 waters

Counterions and buffers to match experiment

pmemd.cuda

NPT

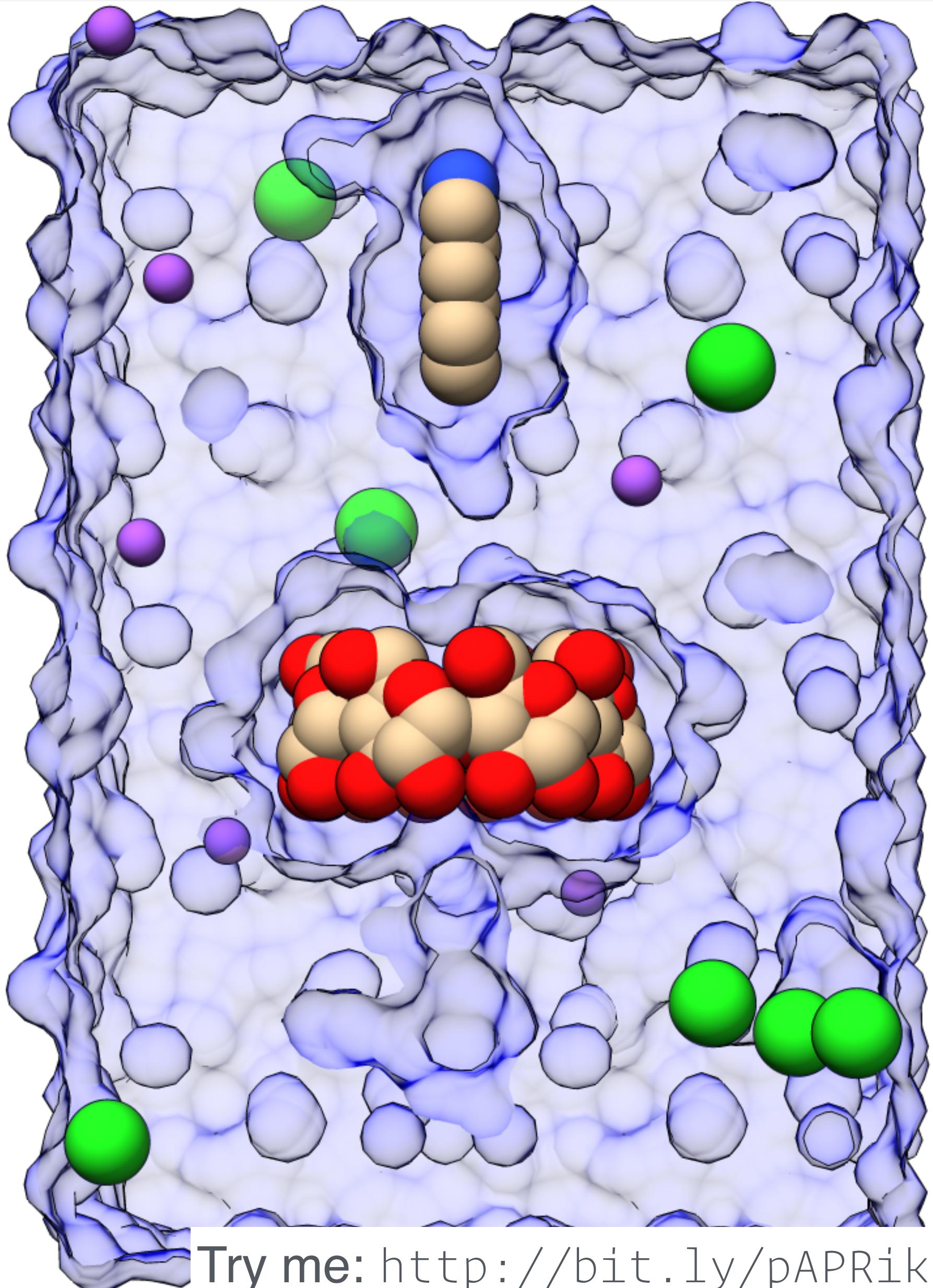
Langevin thermostat

Monte Carlo barostat

300 K

HMR with 4 fs timestep

9 Å nonbonded cutoff



Try me: <http://bit.ly/pAPRika>

So what does it take to run until uncertainties are ~ 0.1 kcal/mol?

5-50 ns simulations in each window (**~ 1 hour per window**).

1 μ s simulations for the **free** and **bound** states gives well-converged enthalpy results.

Ballpark 36 GPU-hours on a GTX TITAN X.

We will use binding free energies to benchmark & optimize new force fields.

Try me: <http://bit.ly/pAPRika>

Open source

Software available on GitHub and permissively licensed

Open science

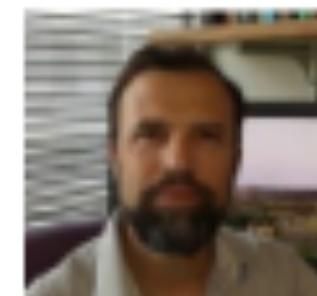
Papers available on bioRxiv and chemRxiv

Open data

Curated physical property and quantum chemical data sets



JOHN CHODERA
SLOAN KETTERING INSTITUTE



MICHAEL GILSON
UNIVERSITY OF CALIFORNIA, SAN DIEGO



CHRISTOPHER BAYLY
OPENEYE SCIENTIFIC



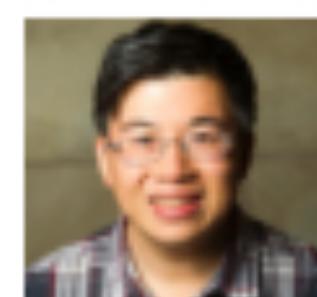
DAVID MOBLEY
UNIVERSITY OF CALIFORNIA, IRVINE



KENNETH KROENLEIN
**NIST THERMODYNAMICS
RESEARCH CENTER**



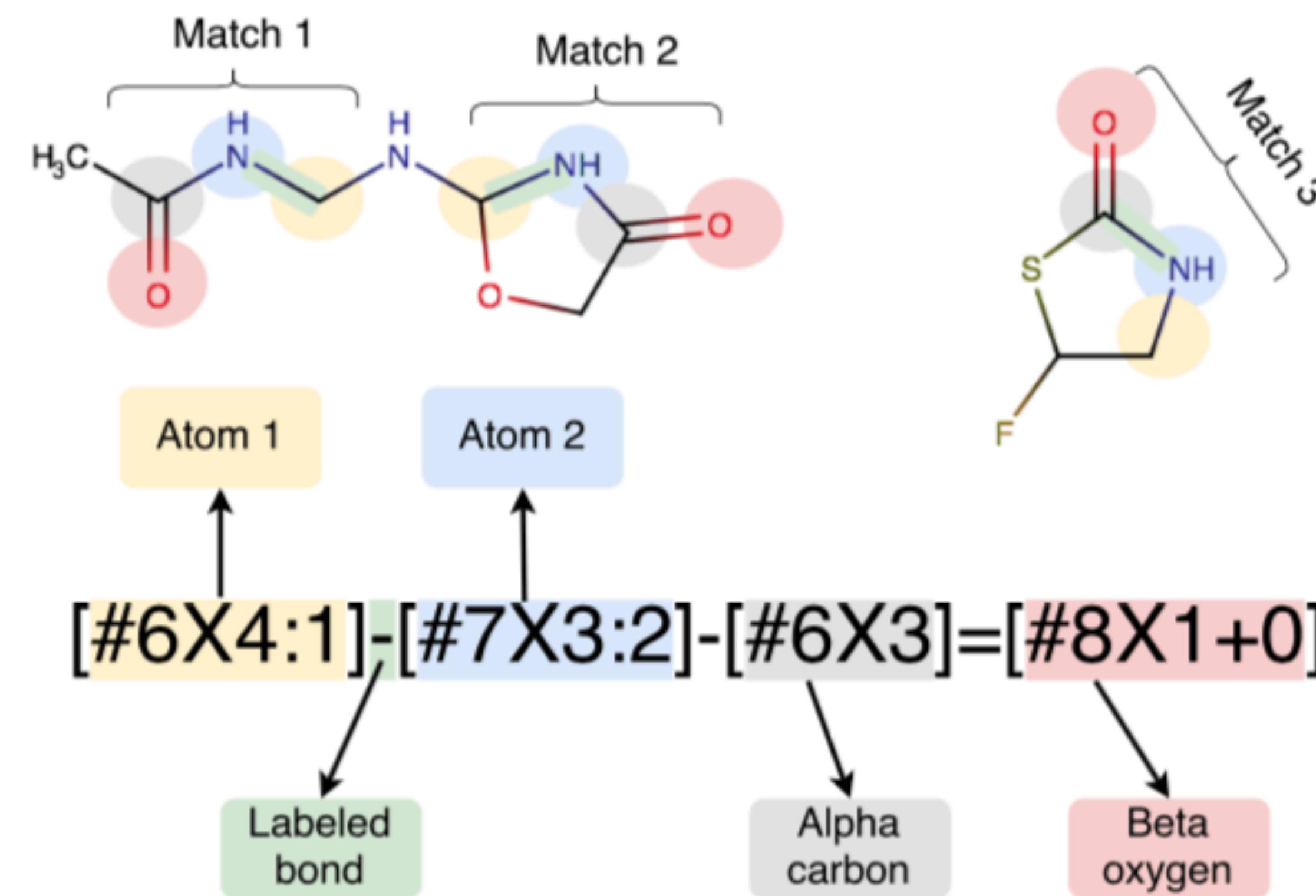
MICHAEL SHIRTS
UNIVERSITY OF COLORADO BOULDER



LEE-PING WANG
UNIVERSITY OF CALIFORNIA, DAVIS

New force fields will be built from SMIRKS strings, not atom types.

Direct chemical perception allows direct assignment of parameters via processing the full chemical graph of molecules.



Try me: <http://bit.ly/pAPRika>

SMIRKS Native Open Force Field – SMIRNOFF – format

gaff.dat

```
gaff.dat smirnoff99Frosst.offxml
141 c1-o 758.1 1.1724 SOURCE2_SOURCE5 31 0.0068
142 c1-oh 435.6 1.3260 SOURCE3 1
143 c1-os 447.5 1.3181 SOURCE3_SOURCE5 8 0.0079
144 c1-p2 289.3 1.7700 SOURCE3 1
145 c1-p3 275.1 1.7900 SOURCE3 1
146 c1-p4 275.1 1.7900 SOURCE3 1
147 c1-p5 302.2 1.7530 SOURCE3 2
148 c1-s2 410.0 1.5950 SOURCE3 1
149 c1-s 400.6 1.6032 SOURCE1_SOURCE5 37 0.0043
150 c1-s4 272.9 1.7460 SOURCE3 2
151 c1-s6 290.4 1.7220 SOURCE3 2
152 c1-sh 324.5 1.6800 SOURCE3 1
153 c1-ss 316.2 1.6898 SOURCE1_SOURCE5 49 0.0113
154 c2-c2 569.4 1.3343 SOURCE1_SOURCE5 3727 0.0053
155 c2-c3 326.8 1.5095 SOURCE1_SOURCE5 10204 0.0042
156 c2-ca 482.1 1.3846 SOURCE3_SOURCE5 9 0.0149
157 c2-cc 523.8 1.3593 SOURCE1_SOURCE5 882 0.0181
158 c2-cd 523.8 1.3593 SOURCE1_SOURCE5 882 0.0181
159 c2-ce 547.3 1.3461 SOURCE3_SOURCE5 3239 0.0058
160 c2-cf 547.3 1.3461 SOURCE3_SOURCE5 3177 0.0057
161 c2-cl 321.3 1.7308 SOURCE1_SOURCE5 290 0.0098
162 c2-cu 590.0 1.3240 5/2017 11 0.0010
163 c2-cx 352.0 1.4850 5/2017 76 0.0060
164 c2-cy 325.6 1.5110 5/2017 21 0.0067
165 c2-f 370.6 1.3385 SOURCE1_SOURCE5 35 0.0085
166 c2-h4 344.6 1.0868 SOURCE3_SOURCE5 517 0.0028
167 c2-h5 338.4 1.0912 SOURCE4_SOURCE5 116 0.0021
168 c2-ha 343.1 1.0879 SOURCE3_SOURCE5 5991 0.0019
169 c2-hc 344.3 1.0870 SOURCE3 789 0.0046
170 c2-hx 350.1 1.0830 SOURCE3 3 0.0008
171 c2-i 215.4 2.1701 SOURCE3_SOURCE5 7 0.0194
```

Ln 1, Col 1 Spaces: 3 UTF-8 LF Plain Text English Live Share

smirnoff99Frosst.offxml

```
?xml version='1.0' encoding='ASCII'?
<SMIRNOFF version="1.0" aromaticity_model="OEAroModel_MDL">
  <!-- SMIRNOFF (SMIRKS Native Open Force Field) template file -->
  <Date>2018-07-14</Date>
  <Author>C. I. Bayly, OpenEye/UC Irvine; C. C. Bannan, UC Irvine; D. L. Mobley, UC [REDACTED]
  <!-- This file is meant for processing via openforcefield.typing.engines.smirnoff -->
  <!-- WARNING: AMBER functional forms drop the factor of 2 in the bond energy term, so we do the same -->
  <Bonds length_unit="angstroms" k_unit="kilocalories_per_mole/angstrom**2">
    <Bond smirks="[#6X4:1]-[#6X4:2]" id="b1" k="620.0" length="1.526"/>
    <Bond smirks="[#6X4:1]-[#6X3:2]" id="b2" k="634.0" length="1.51"/>
    <Bond smirks="[#6X4:1]-[#6X3:2]=[#8X1+0]" id="b3" k="634.0" length="1.522"/>
    <Bond smirks="[#6X3:1]-[#6X3:2]" id="b4" k="820.0" length="1.45"/>
    <Bond smirks="[#6X3:1]:[#6X3:2]" id="b5" k="938.0" length="1.40"/>
    <Bond smirks="[#6X3:1]=[#6X3:2]" id="b6" k="1098.0" length="1.35"/>
    <Bond smirks="[#6:1]-[#7:2]" id="b7" k="734.0" length="1.47"/>
    <Bond smirks="[#6X3:1]-[#7X3:2]" id="b8" k="854.0" length="1.38"/>
    <Bond smirks="[#6X4:1]-[#7X3:2]-[#6X3]=[#8X1+0]" id="b9" k="674.0" length="1.449"/>
    <Bond smirks="[#6X3:1](=[#8X1+0])-[#7X3:2]" id="b10" k="980.0" length="1.335"/>
    <Bond smirks="[#6X3:1]-[#7X2:2]" id="b11" k="820.0" length="1.39"/>
    <Bond smirks="[#6X3:1]:[#7X2,#7X3+1:2]" id="b12" k="960.0" length="1.34"/>
    <Bond smirks="[#6X3:1]=[#7X2,#7X3+1:2]" id="b13" k="1060.0" length="1.30"/>
    <Bond smirks="[#6:1]-[#8:2]" id="b14" k="640.0" length="1.410"/>
    <Bond smirks="[#6X4:1]-[#8X2H0:2]" id="b15" k="640.0" length="1.370"/>
    <Bond smirks="[#6X3:1]-[#8X2:2]" id="b16" k="700.0" length="1.326"/>
    <Bond smirks="[#6X3:1]-[#8X2H1:2]" id="b17" k="900.0" length="1.364"/>
    <Bond smirks="[#6X3a:1]-[#8X2H0:2]" id="b18" k="900.0" length="1.323"/>
    <Bond smirks="[#6X3:1](=[#8X1])-[#8X2H0:2]" id="b19" k="640.0" length="1.340"/>
    <Bond smirks="[#6:1]=[#8X1+0,#8X2+1:2]" id="b20" k="1140.0" length="1.229"/>
    <Bond smirks="[#6X3:1](~[#8X1])~[#8X1:2]" id="b21" k="1312.0" length="1.250"/>
    <Bond smirks="[#6X3:1]:[#8X2+1:2]" id="b22" k="1140.0" length="1.28"/>
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF XML English Live Share

SMIRNOFF99Frosst is a new kind of force field.

Fewer numerical parameters than GAFF

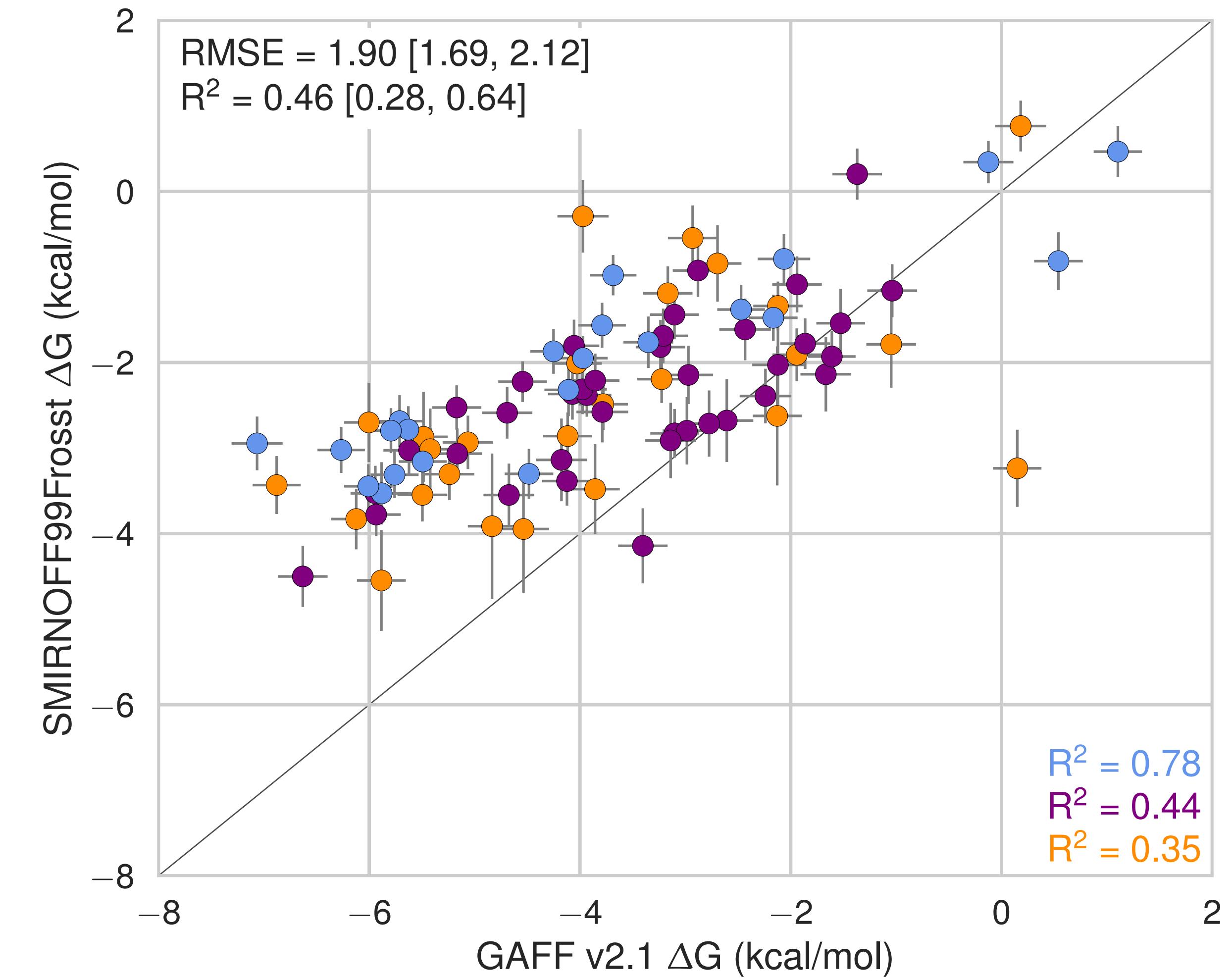
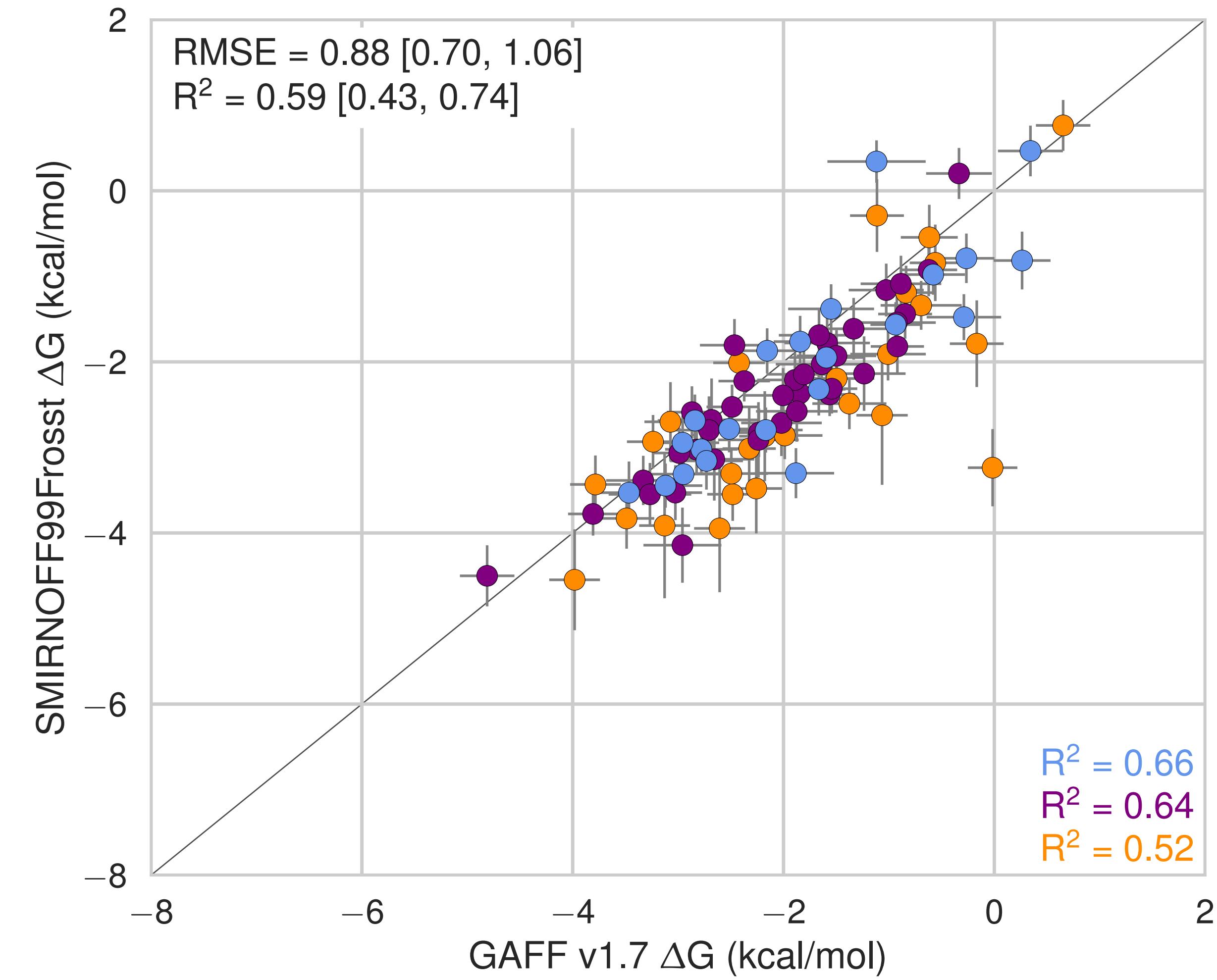
GAFF: 6,700

SMIRNOFF99Frosst: 330

Larger chemical space coverage than GAFF

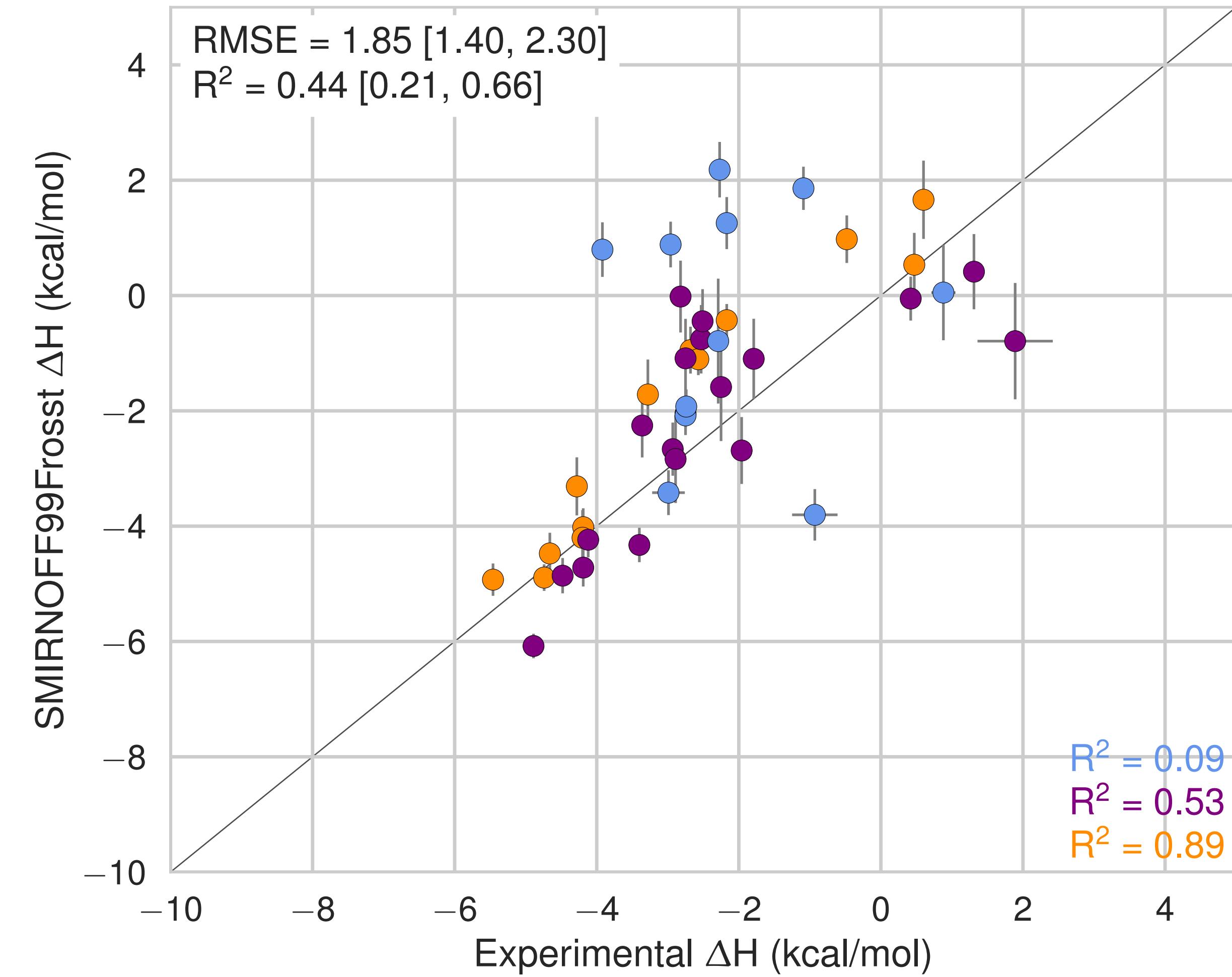
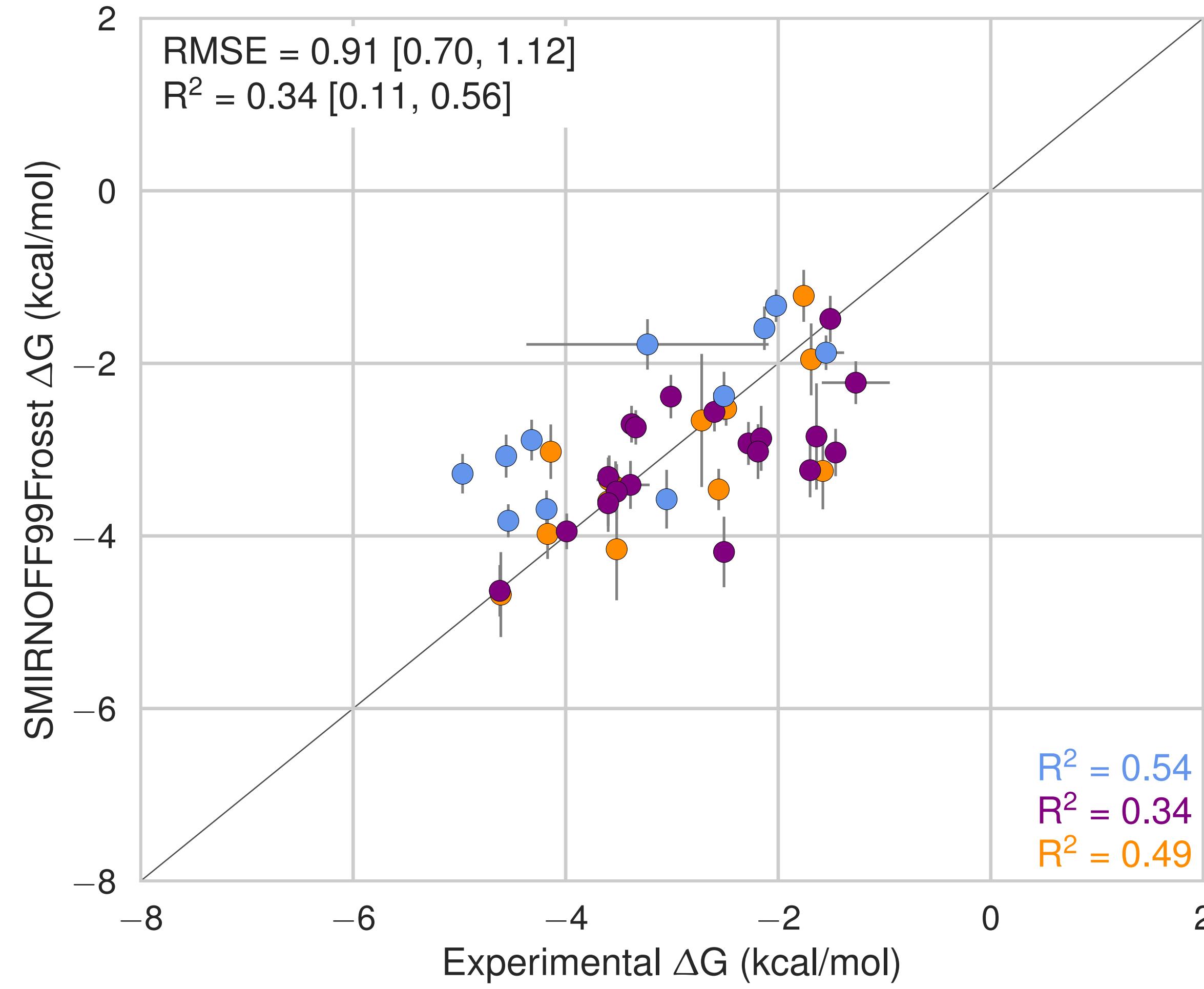
Usable in multiple simulation codes

SMIRNOFF and GAFF v1.7 perform similarly; the difference to GAFF v2.1 is bigger.



Try me: <http://bit.ly/pAPRika>

We have a ΔG RMSE around 1 kcal/mol and ΔH RMSE around 2 kcal/mol.



We hope to finalize the pAPRika public API within about 6 months.

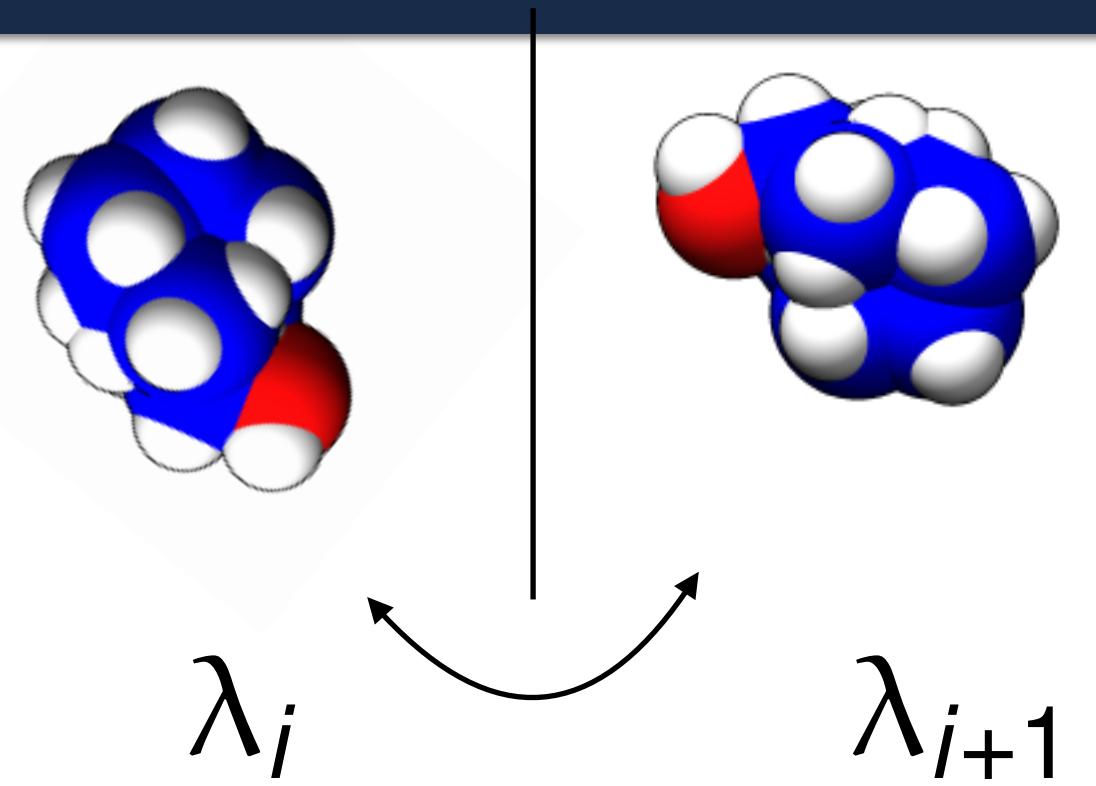
```
get_host_guest_system(binding_db_ID="BDBM197287",
                      pubmed="27658802",
                      guest_SMILES="CCCC")
```

```
setup_restraints_protocol(conformational_restraints=True,
                           pull_force_constant=10)
```

```
run_simulation(target_uncertainty=0.01,
                analysis_method="MBAR",
                uncertainty_method="autocorrelation")
```

We are trying to improve automation and the robustness of our approach.

Can we improve the sampling of the bound state by using Hamiltonian replica exchange?



What is the return on investment for running longer in this λ window?

$$\text{ROI} = \frac{\partial \Delta G_{\text{SEM}}}{\partial U / \partial \lambda_{\text{SEM}}} \frac{\partial U / \partial \lambda_{\text{SEM}}}{\partial N_{\text{frames}}}$$

How do we do compared to free energy perturbation techniques?

Fast and Flexible GPU Accelerated Binding Free Energy Calculations Within the AMBER Molecular Dynamics Package

Daniel J. Mermelstein ^[a] Charles Lin, ^[a,b] Gard Nelson, ^[c] Rachael Kretsch, ^[a,d] J. Andrew McCammon, ^[a] and Ross C. Walker ^[a,b]

Michael K. Gilson, M.D., Ph.D.



Gilson group

Ido Ben-Shalom, Ph.D.

Mudong (Winter) Feng

Samuel Kantonen

Katy Kellett, Ph.D.

Tiqing Liu, Ph.D.

John Lomibao

Michael Schauperl, Ph.D.

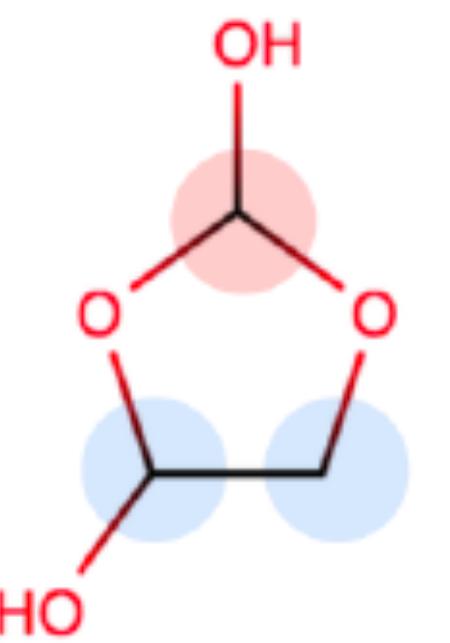
Felix Yang

Niel Henriksen, Ph.D. (Atomwise)

Jian Yin, Ph.D. (XtalPi)

UC San Diego
SKAGGS SCHOOL OF PHARMACY
AND PHARMACEUTICAL SCIENCES

NIH National Institutes of Health
Turning Discovery Into Health



Open Force Field Initiative

SDSC

SAN DIEGO SUPERCOMPUTER CENTER