Alchemical Free Energy Methods with BioSimSpace

Julien Michel, Anna Herz, Finlay Clark, Lester Hedges, Christopher Woods

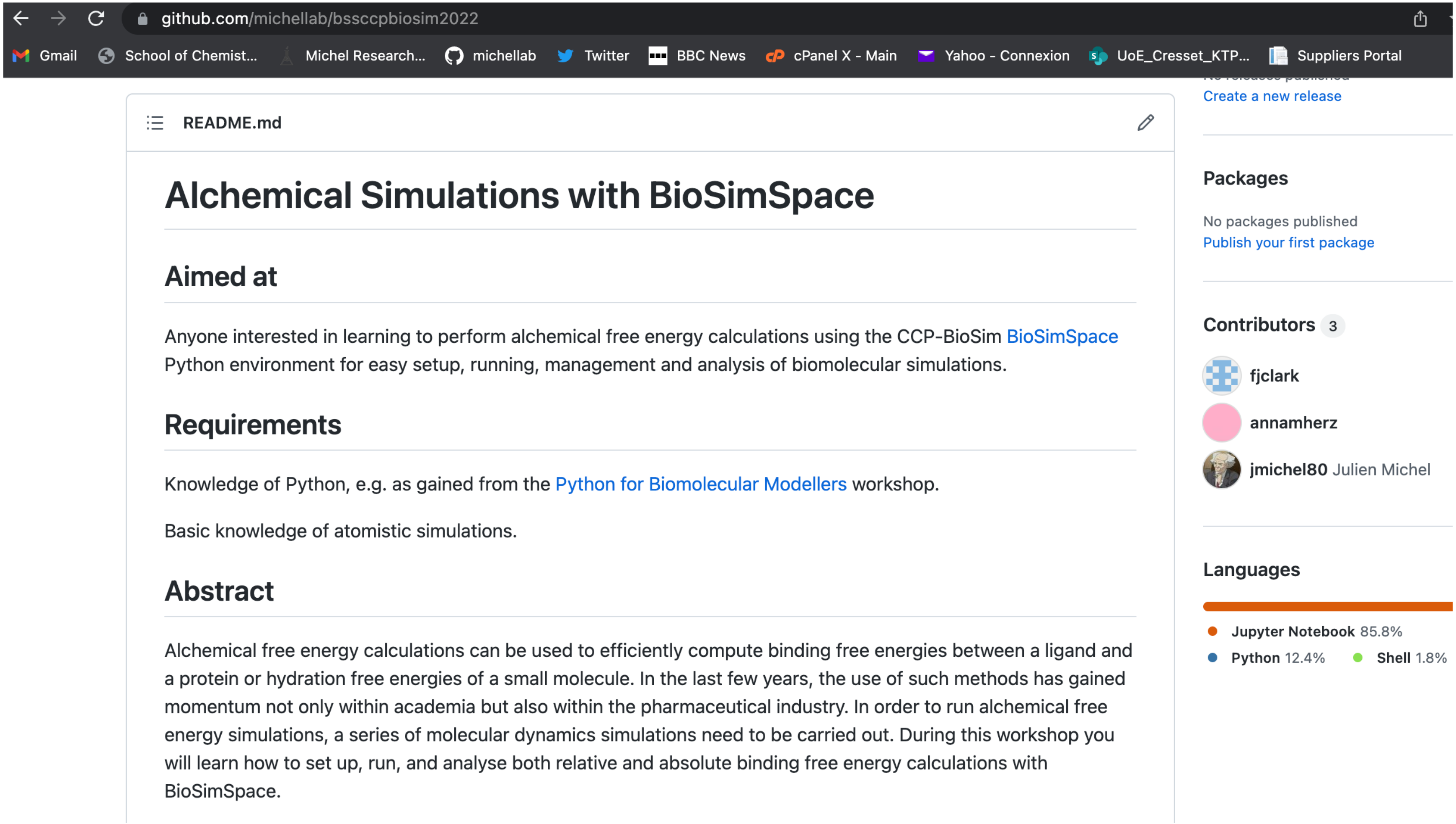
CCPBioSim Training Week – Leeds - 20/09/2022

Schedule

- 9.00 – 9.30 Set up & Lecture
- 9.30 – 10.15 Introduction to Alchemistry with BioSimSpace
- 10.15 - 10.30 Coffee Break
- 10.30 – 11.15 Relative Binding Free Energies with BioSimSpace
- 11.15 – 12.00 Absolute Binding Free Energies with BioSimSpace

Workshop materials

github.com/michellab/bssccpbiosim2022



Alchemical Simulations with BioSimSpace

Aimed at

Anyone interested in learning to perform alchemical free energy calculations using the CCP-BioSim BioSimSpace Python environment for easy setup, running, management and analysis of biomolecular simulations.

Requirements

Knowledge of Python, e.g. as gained from the Python for Biomolecular Modellers workshop.

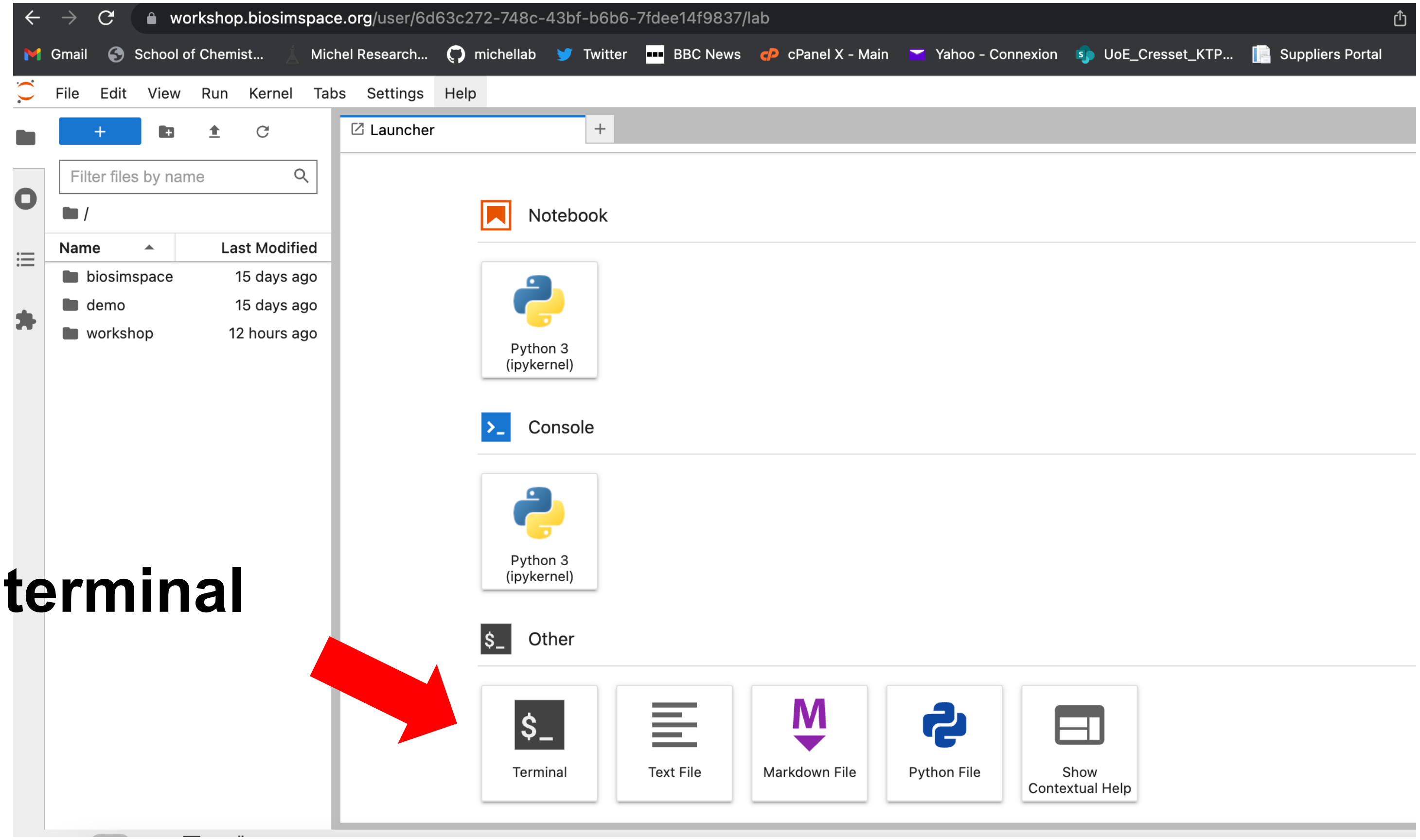
Basic knowledge of atomistic simulations.

Abstract

Alchemical free energy calculations can be used to efficiently compute binding free energies between a ligand and a protein or hydration free energies of a small molecule. In the last few years, the use of such methods has gained momentum not only within academia but also within the pharmaceutical industry. In order to run alchemical free energy simulations, a series of molecular dynamics simulations need to be carried out. During this workshop you will learn how to set up, run, and analyse both relative and absolute binding free energy calculations with BioSimSpace.

Accessing the workshop server

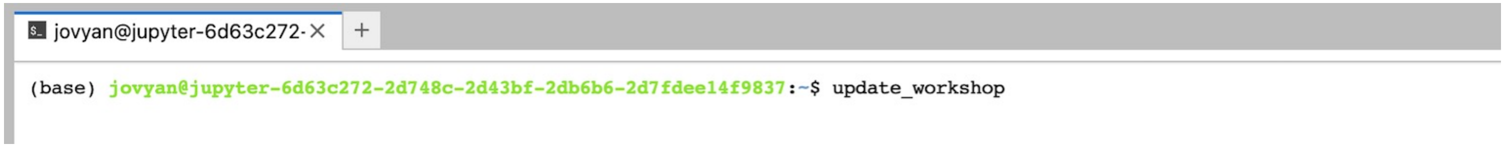
workshop.biosimspace.org



workshop.biosimspace.org/user/6d63c272-748c-43bf-b6b6-7fdee14f9837/lab

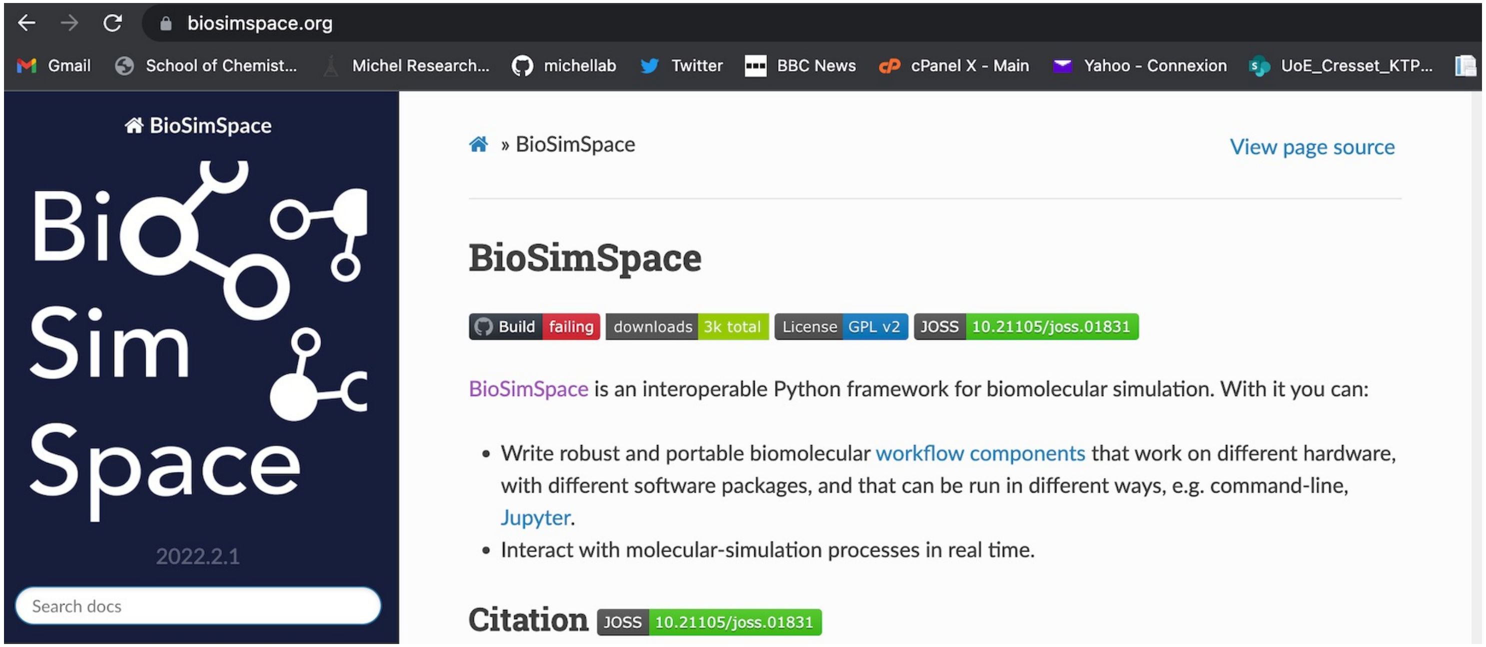
Name	Last Modified
biosimspace	15 days ago
demo	15 days ago
workshop	12 hours ago

open terminal



```
(base) jovyan@jupyter-6d63c272-2d748c-2d43bf-2db6b6-2d7fdee14f9837:~$ update_workshop
```

What is BioSimSpace?



BioSimSpace

BioSimSpace is an interoperable Python framework for biomolecular simulation. With it you can:

- Write robust and portable biomolecular workflow components that work on different hardware, with different software packages, and that can be run in different ways, e.g. command-line, Jupyter.
- Interact with molecular-simulation processes in real time.

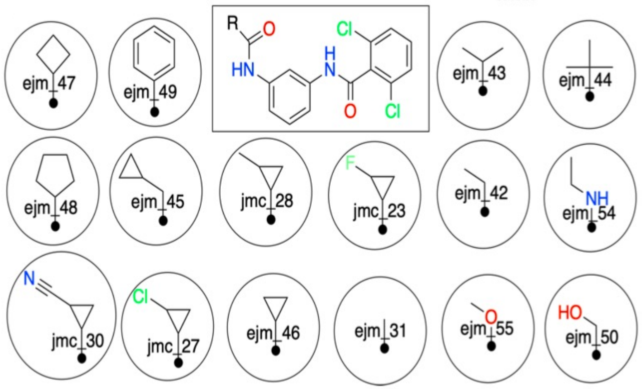
Citation JOSS 10.21105/joss.01831

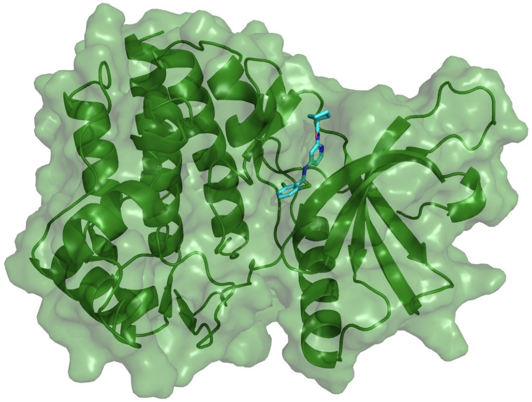
JOSS
The Journal of Open Source Software

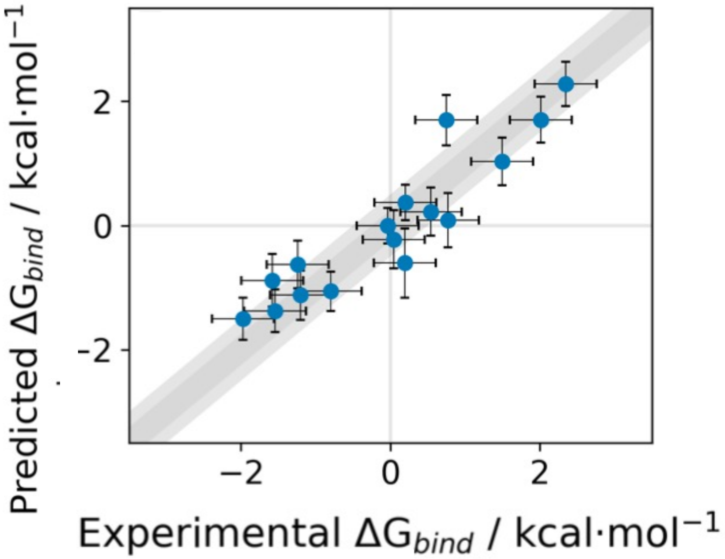
BioSimSpace: An interoperable Python framework for biomolecular simulation

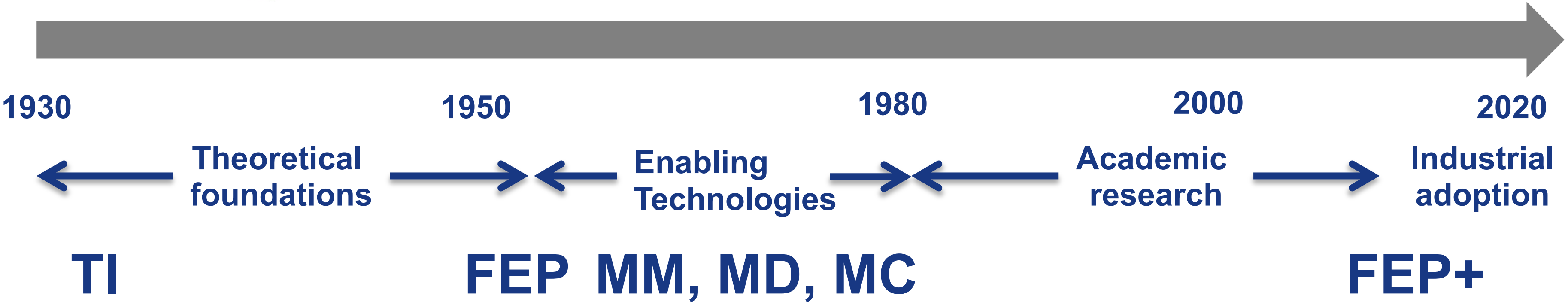
Lester O. Hedges[1], Antonia S.J.S. Mey[2], Charles A. Laughton[3], Francesco L. Gervasio[4], Adrian J. Mulholland[5], Christopher J. Woods[1], and Julien Michel[2]

Alchemical free energy methods in drug design






Predicted ΔG_bind / kcal·mol⁻¹
2
0
-2
−2
0
2
Experimental ΔG_bind / kcal·mol⁻¹



1930

1950

1980

2000

2020

Theoretical foundations

Enabling Technologies

Academic research

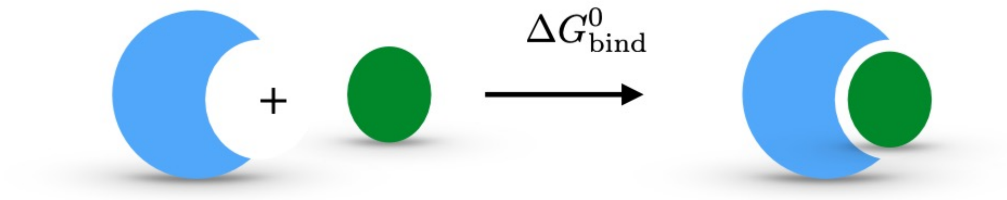
Industrial adoption

TI

FEP MM, MD, MC

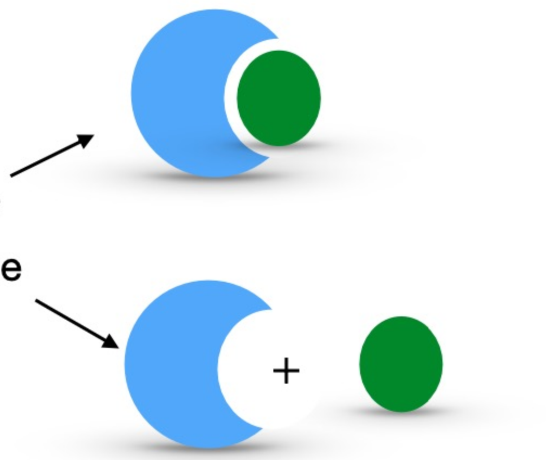
FEP+

Free energy of binding



$$\Delta G_{\text{bind}} = -k_B T \ln K_b^\circ$$

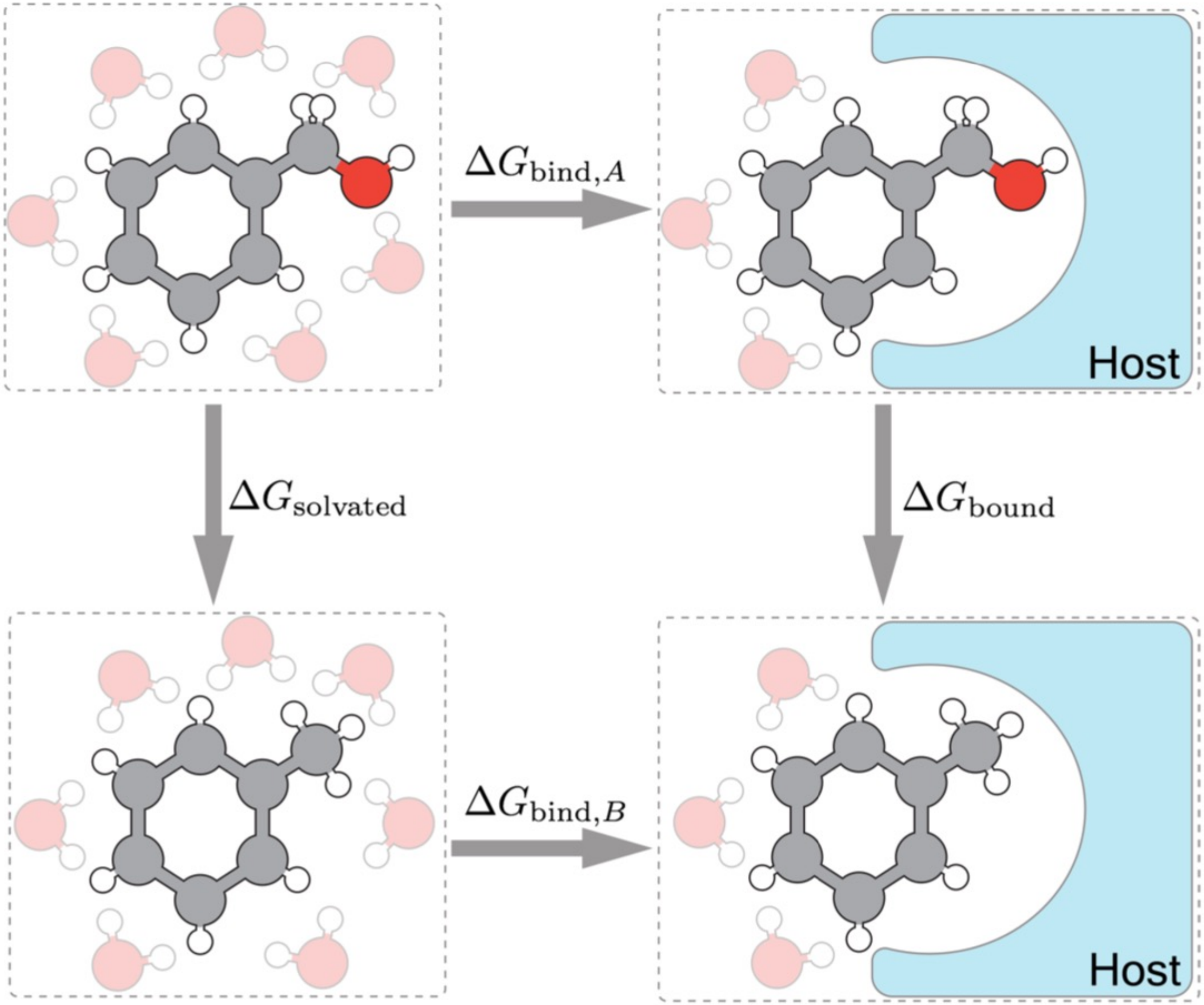
$$K_b^\circ = \frac{\text{probability in bound state}}{\text{probability in unbound state}}$$



Sample this using
Molecular dynamics?

slide courtesy of Antonia Mey

The alchemical pathway

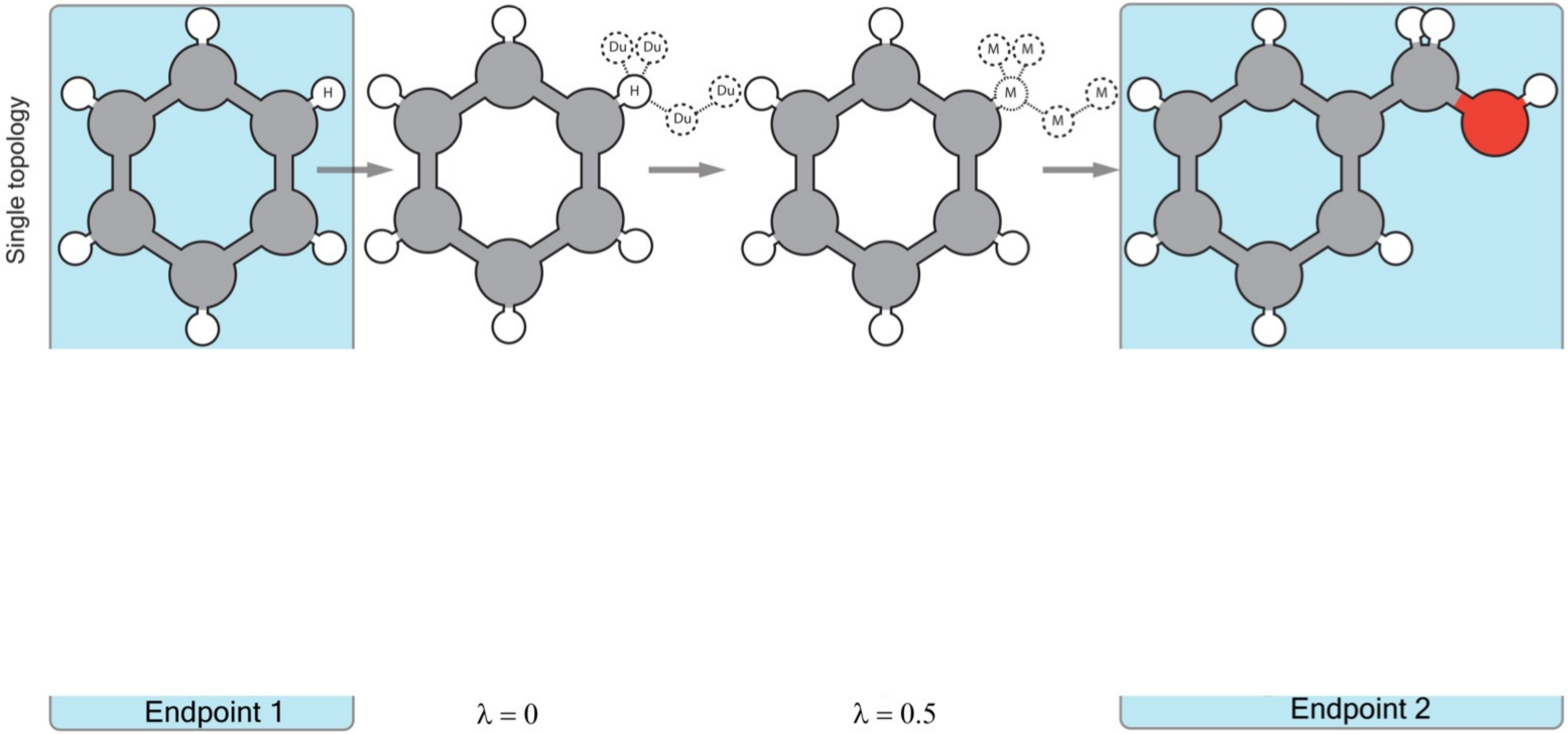


$\Delta G_{\mathrm{bind},A}$
Host
$\Delta G_{\mathrm{solvated}}$
$\Delta G_{\mathrm{bound}}$
$\Delta G_{\mathrm{bind},B}$
Host

$$\Delta\Delta G_{\mathrm{bind}} = \Delta G_{\mathrm{bound}} - \Delta G_{\mathrm{solv}}$$

slide courtesy of Antonia Mey

The lambda coordinate



$$U(\lambda, \mathbf{x}) = (1-\lambda)U_0(\mathbf{x}) + \lambda U_1(\mathbf{x}) + U_{\text{unaffected}}$$

slide courtesy of Antonia Mey

Potential energy functions

$$U = \sum_{\text{bonds}} \frac{1}{2} k_b (r - r_0)^2 + \sum_{\text{angles}} \frac{1}{2} k_a (\theta - \theta_0)^2 + \sum_{\text{torsions}} \frac{V_n}{2} [1 + \cos(n\phi - \delta)]$$

$$+ \sum_{\text{improper}} V_{imp} + \sum_{\text{LJ}} 4\epsilon_{ij} \left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right) + \sum_{\text{elec}} \frac{q_i q_j}{r_{ij}},$$

- The functional form and parameter set define a force field.
- Commonly used force fields include:
  CHARMM (Chemistry at Harvard Molecular Mechanics)
  AMBER (Assisted Model Building with Energy Refinement)
  OPLS (Optimised Potentials for Liquid Simulations)

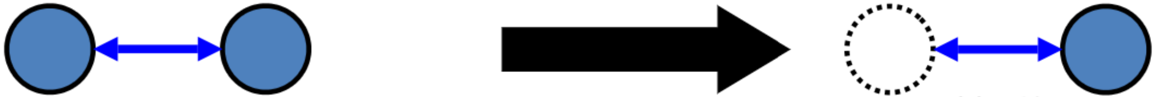
slide courtesy of Antonia Mey

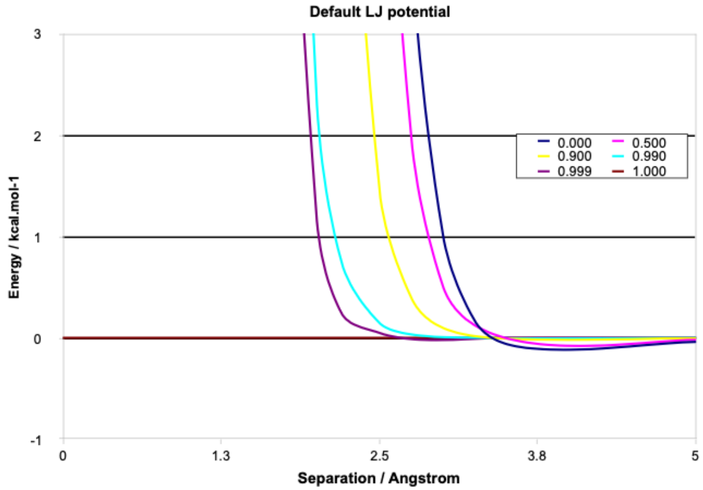
Softcore potentials

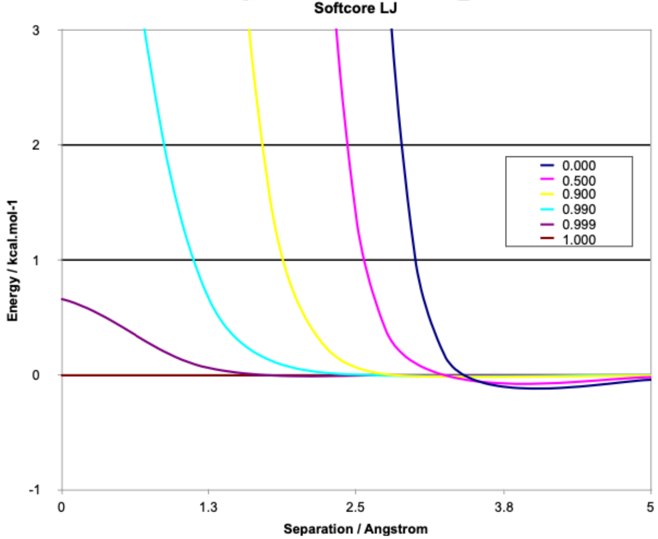
Zacharias et al. *J. Chem. Phys.* 100, 9025-, **1994**

$$U_{\text{nonbonded},\lambda} = (1-\lambda)4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}^{12}}{\left(\lambda\delta\sigma_{ij}+r_{ij}^2\right)^6}\right) - \left(\frac{\sigma_{ij}^{6}}{\left(\lambda\delta\sigma_{ij}+r_{ij}^2\right)^3}\right)\right] + \frac{(1-\lambda)^n q_i q_j}{4\pi\varepsilon_0\sqrt{\left(\lambda+r_{ij}^2\right)}}$$

Dummy atoms!

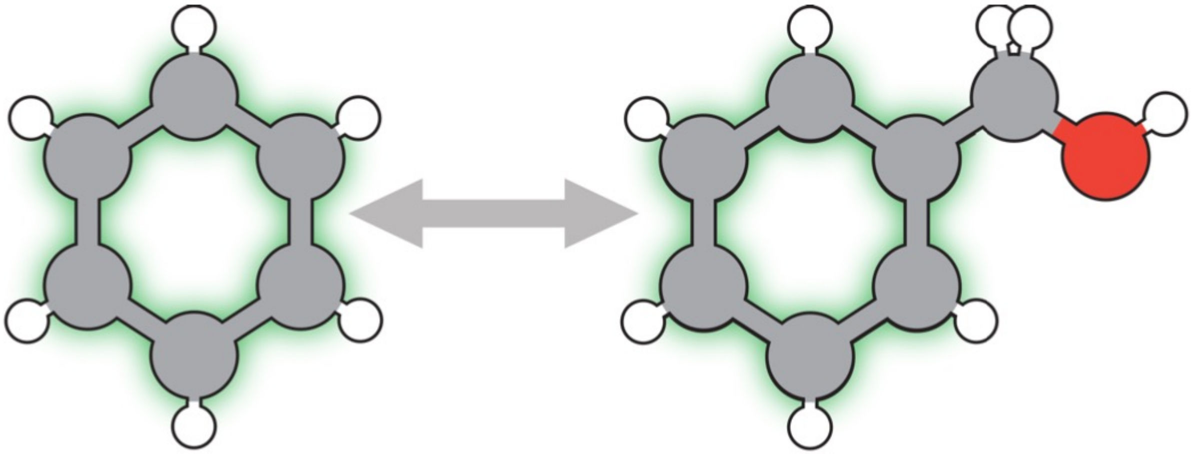


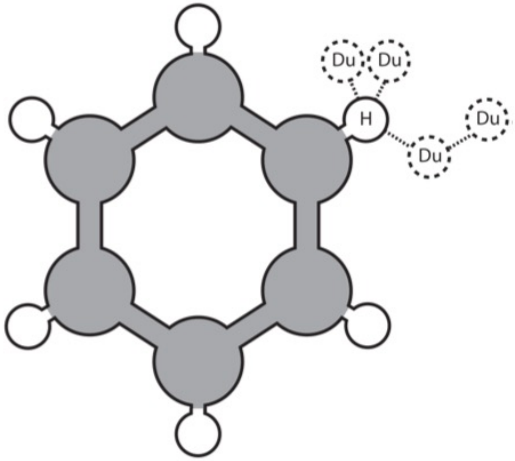




slide courtesy of Antonia Mey

BioSimSpace merged molecules

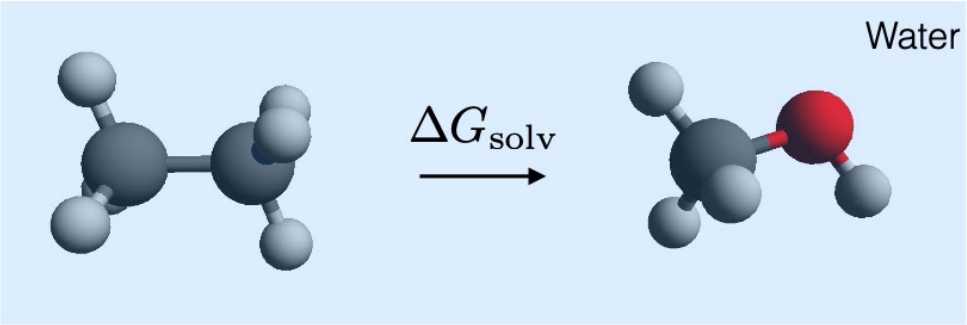


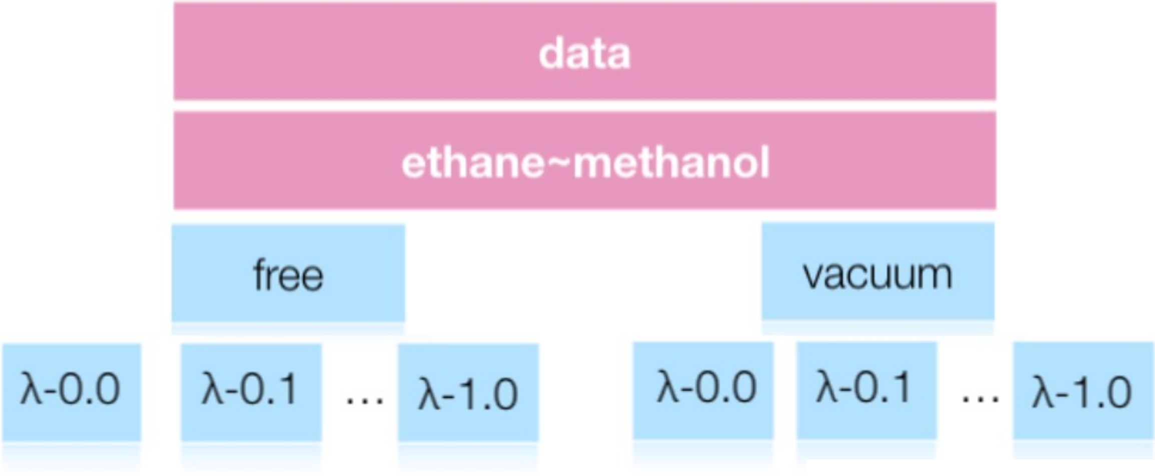


BioSimSpace holds a merged molecule and can write intermediates

slide courtesy of Antonia Mey

BioSimSpace setups inputs for different simulation engines




data

ethane~methanol

free

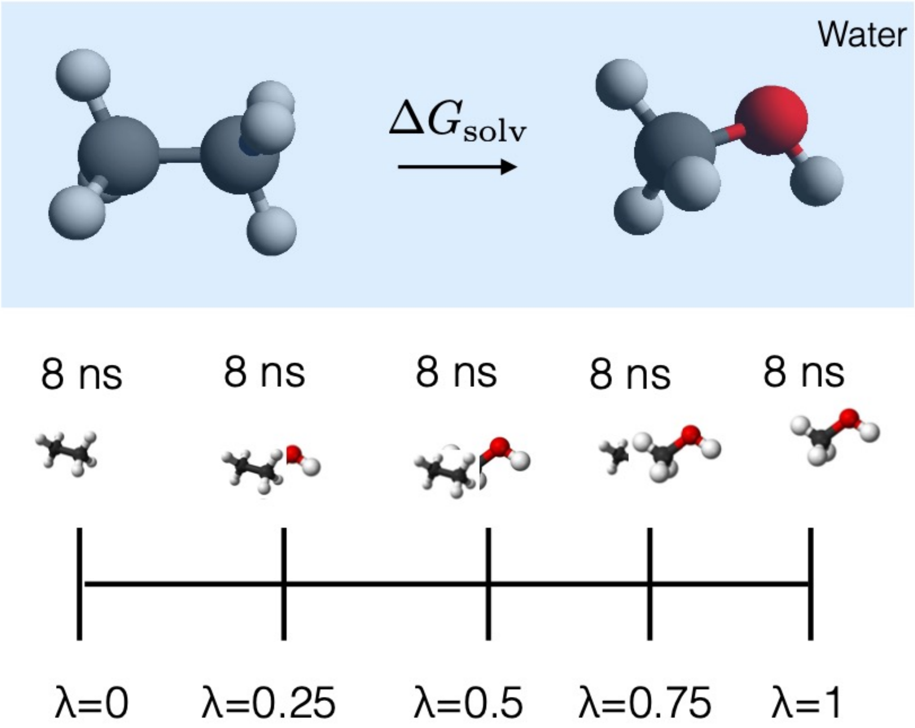
vacuum

λ-0.0 λ-0.1 … λ-1.0

λ-0.0 λ-0.1 … λ-1.0

SOMD AMBER MD GROMACS FAST. FLEXIBLE. FREE.

BioSimSpace analyses simulation outputs



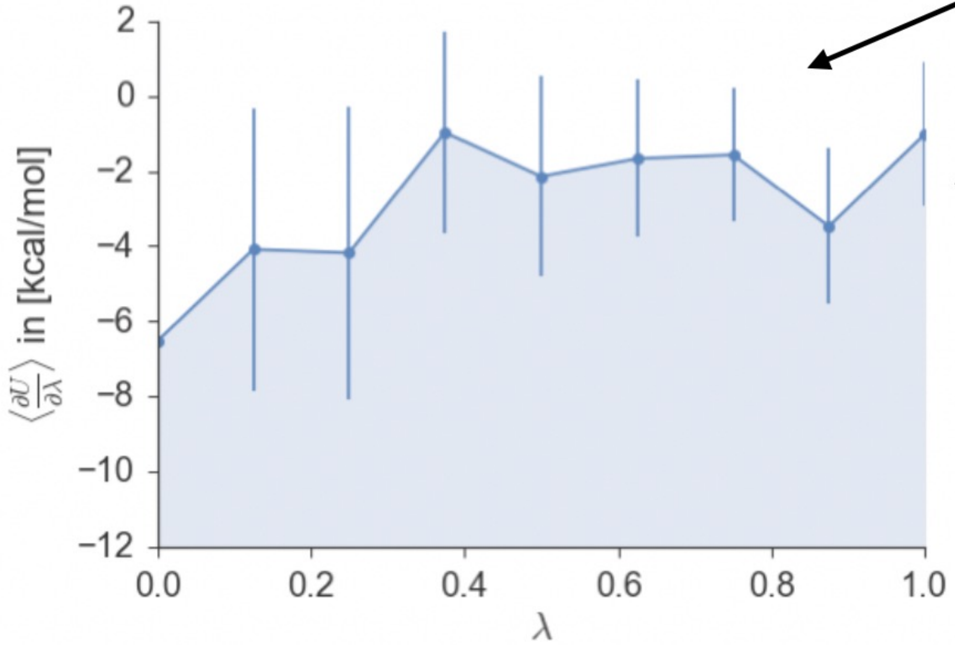
Zwanzig equation

$$\Delta G_{AB} = G(A) - G(B) = -k_B T \ln\langle \exp(-\frac{U_B - U_A}{k_B T})\rangle_A$$

slide courtesy of Antonia Mey

Thermodynamic integration

$$\Delta\Delta G_{\text{bind}} = \Delta G_{\text{bound}} - \Delta G_{\text{solv}}$$



$$\Delta G = \int_{\lambda_0}^{\lambda_1} \frac{\partial G(\lambda)}{\partial \lambda} d\lambda$$

$$\Delta G = \int_0^1 \langle \frac{\partial U}{\partial \lambda} \rangle d\lambda$$

The integral represents the area under the curve.
There are different ways in which one can numerically integrate.

slide courtesy of Antonia Mey

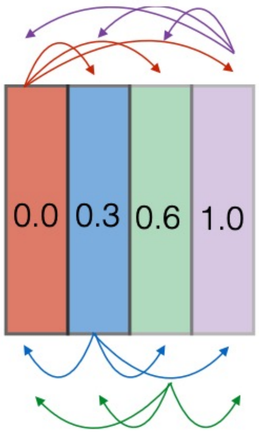
Multistate Bennett Acceptance Ratio (MBAR)

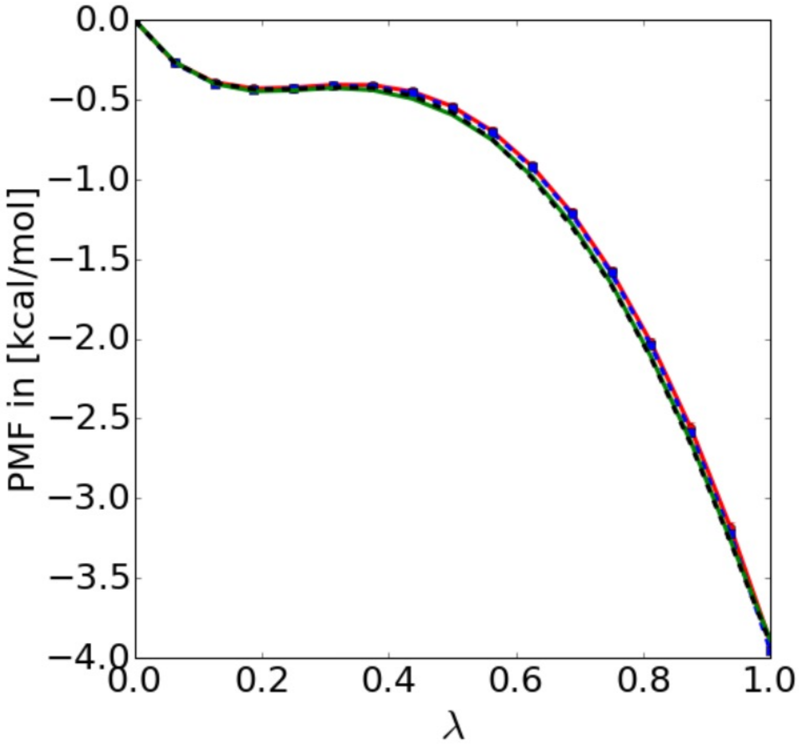
$$\mathcal{G}(\lambda_1) = -k_B T \ln \sum_{j=1}^{N} \sum_{n=1}^{K_j} \frac{\exp(-U(\lambda_i)/kT)}{\sum_{k=1}^{N} N_k \exp((\mathcal{G}(\lambda_k) - U(\lambda_k))/kT)}$$

MBAR is a reweighting estimator

The free energy is correct up to an additive constant, which will cancel out when evaluating free energy differences.

$$\Delta G_{AB} = G(\lambda_1) - G(\lambda_0)$$

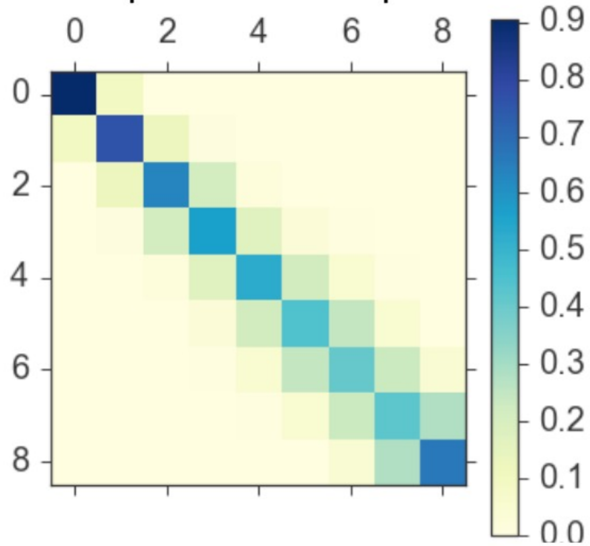


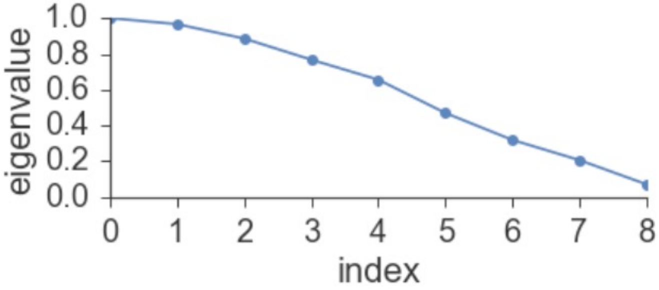


slide courtesy of Antonia Mey

Overlap matrix

Overlap matrix estimates the phase space overlap





Define a weight matrix W containing the weight of each sample x_n.

$$W_{n,i}(x_n) = \frac{\boxed{\exp(\beta F_i - U_i(x_n))}}{\sum_{k=1}^{K} N_K \exp(\beta F_k - \beta U_k(x_n))}$$

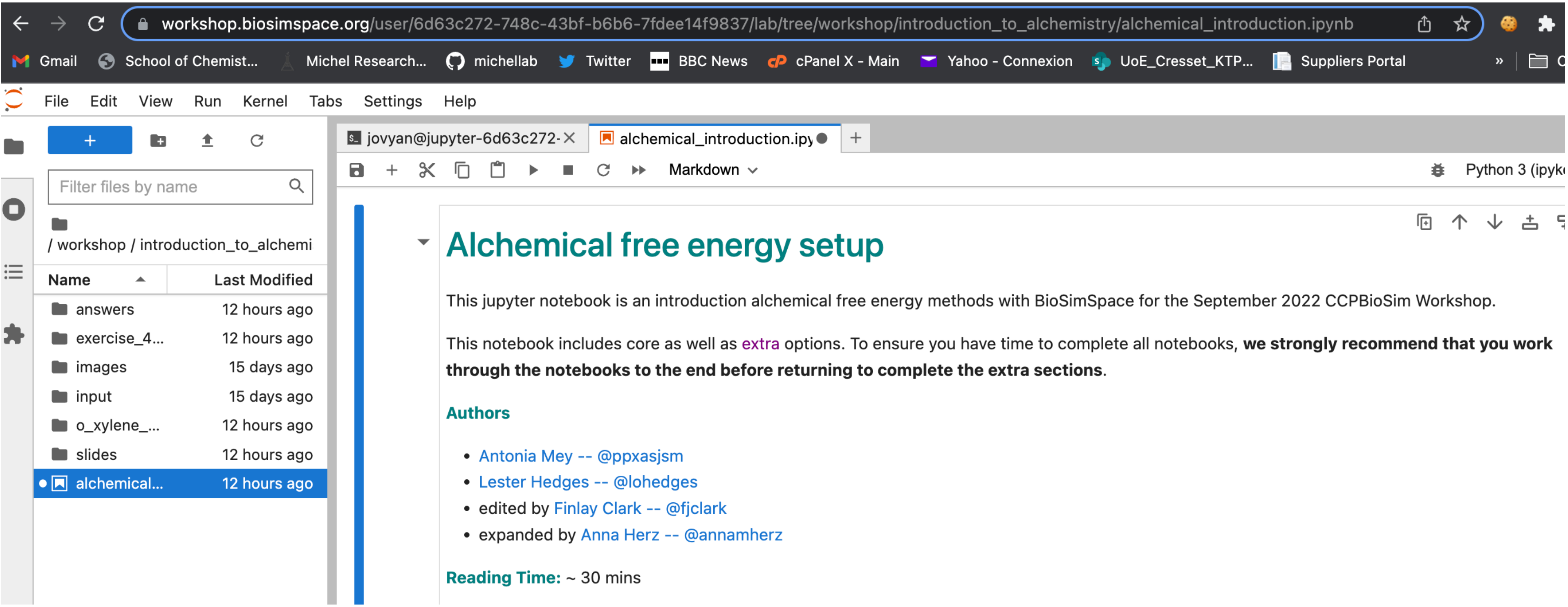
Probability of $p_i(x_n)$ of x_n occurring at lambda window i.

$$\mathbf{O} = \mathbf{W}^{\mathbf{T}}\mathbf{W}\mathbf{N}$$

N is a diagonal matrix with the number of samples collected at each lambda window.

slide courtesy of Antonia Mey

Let’s get started



/ workshop / introduction_to_alchemi

Name	Last Modified
answers	12 hours ago
exercise_4...	12 hours ago
images	15 days ago
input	15 days ago
o_xylene_...	12 hours ago
slides	12 hours ago
alchemical...	12 hours ago

Alchemical free energy setup

This jupyter notebook is an introduction alchemical free energy methods with BioSimSpace for the September 2022 CCPBioSim Workshop.

This notebook includes core as well as extra options. To ensure you have time to complete all notebooks, **we strongly recommend that you work through the notebooks to the end before returning to complete the extra sections.**

Authors

- Antonia Mey -- @ppxasjsm
- Lester Hedges -- @lohedges
- edited by Finlay Clark -- @fjclark
- expanded by Anna Herz -- @annamherz

Reading Time: ~ 30 mins