Calculating molecular free energies using Parsl

Glen Hocky

ParslFest 2020

Department of Chemistry, NYU

Also working with Parallel Works, Inc

hockyg@nyu.edu / @glenhocky

ACKNOWLEDGEMENTS

Hocky Research Group, Summer 2020

Yuvraj, David, **Willmor**, Gaurav, Subarna,
Kangxin, **Daniela**, Bobby

FUNDING AND SUPPORT

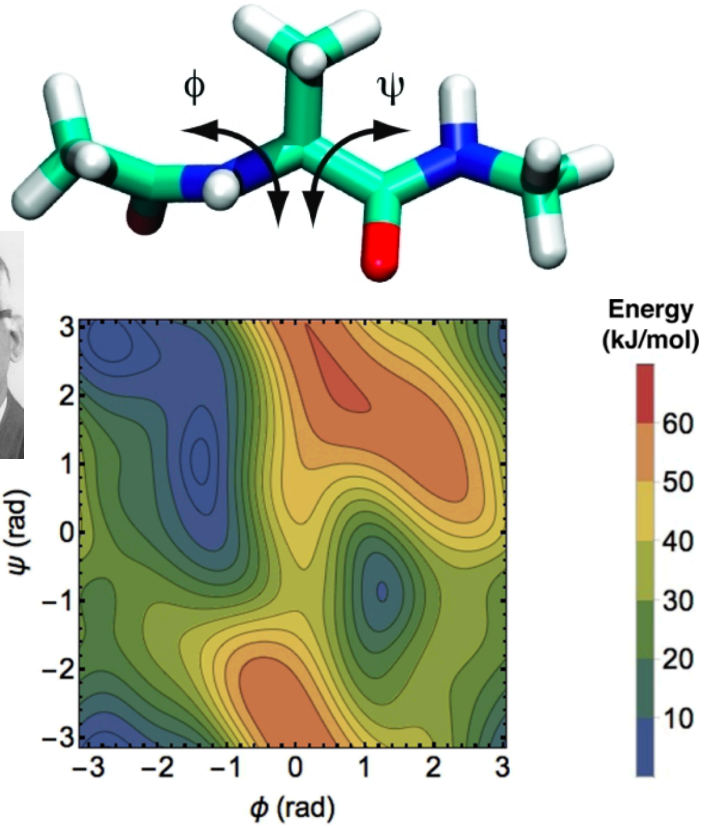
New York University
Faculty of Arts and Sciences

Department of Energy

Parsl Team
Mike Wilde + Parallel Works

Molecular Dynamics Basics

- Simple idea:
  F = m a → positions, velocities of atoms
  - Where do forces come from?
- Molecular mechanics "forcefield" built to reproduce experimental and quantum mechanical data
  - Atoms
    - Mass
    - Charge
    - Excluded volume
    - van der Waal's interactions
  - Bonds
    - Stretch
    - Bend
  - Torsion

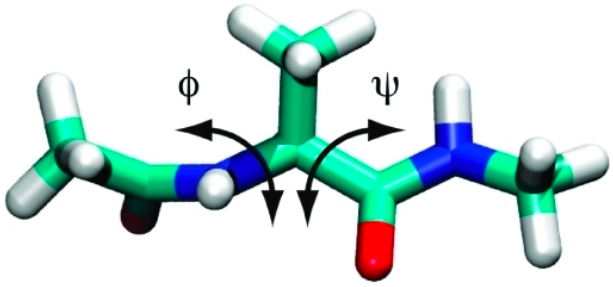


Put molecule into a "temperature bath", and get $P(X) \propto e^{-\frac{E(X)}{k_B T}}$

LAMMPS + PLUMED In Parsl (w/ Parallel Works)

- LAMMPS is a general purpose molecular dynamics package from Sandia National Lab (Steve Plimpton)
- PLUMED is an Open Source Software package that plugs in to MD codes such as LAMMPS
- Analyzes *collective variables* (CVs) and *biases* CVs

```
1  UNITS LENGTH=A TIME=fs ENERGY=kcal/mol
2  phi: TORSION ATOMS=5,7,9,15
3  psi: TORSION ATOMS=7,9,15,17
4
5  us: RESTRAINT ARG=phi,psi KAPPA=%(PHI_KAPPA)s,%(PSI_KAPPA)s AT=%(PHI_CENTER)s,%(PSI_CENTER)s
6
7  PRINT ARG=phi,psi,us.bias FILE=%(OUTPREFIX)s.colvars.txt STRIDE=100
```



Umbrella Sampling w/ Parsl (Parallel Works)

```
# example of running dialanine in lammps with parsl and docker, writing dihedral in plumed

@bash_app(cache=True)
def run_lammps_diala(stdout='parsl-outputs/diala_us.stdout', stderr='parsl-outputs/diala_us.stderr', inputs=[], outputs=[],
                     docker="sudo docker run --rm -u $(id -u $USER) -v '%s:/home/whitelab/scratch' whitelab/plumed-lammps",
                     ):
    import os
    run_dir = os.path.join(os.getcwd(),os.path.dirname(outputs[0]))
    return docker%run_dir + ' lammps -in %s -log %s '%(os.path.basename(inputs[0]),os.path.basename(outputs[0]))
```

```
pull_result_list = []
prod_result_list = []

for fileidx in range(len(pull_prefix_list)):
    pull_prefix = pull_prefix_list[fileidx]
    prod_prefix = prod_prefix_list[fileidx]
    #manual check for finish
    pull_result_list.append(
        run_lammps_diala(inputs=[pull_prefix+'.input'],
                     outputs=[pull_prefix+'.log',pull_prefix+'.colvars.txt',pull_prefix+'.data'])
    )

    prod_result_list.append(
        run_lammps_diala(inputs=[prod_prefix+'.input',pull_result_list[-1].outputs[2]],
                     outputs=[prod_prefix+'.log',prod_prefix+'.colvars.txt',prod_prefix+'.data'])
    )
```

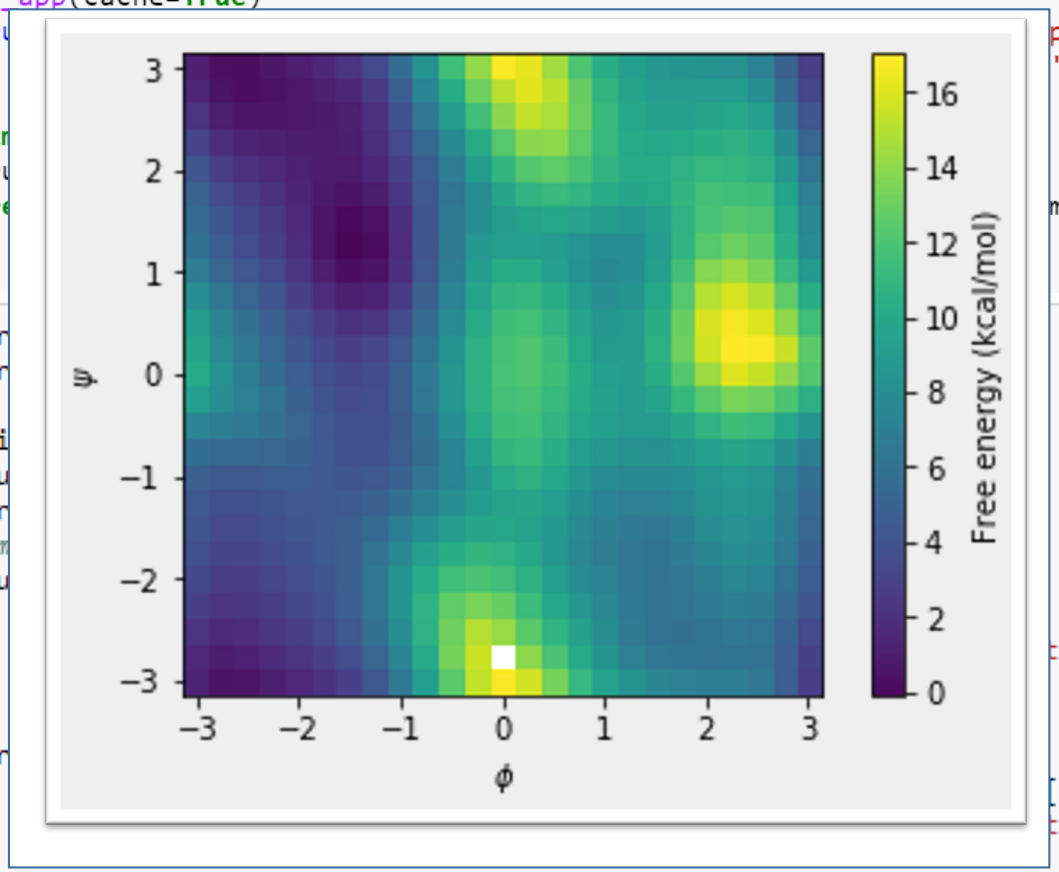
UMBRELLA SAMPLING W/ PARSL (PARALLEL WORKS)

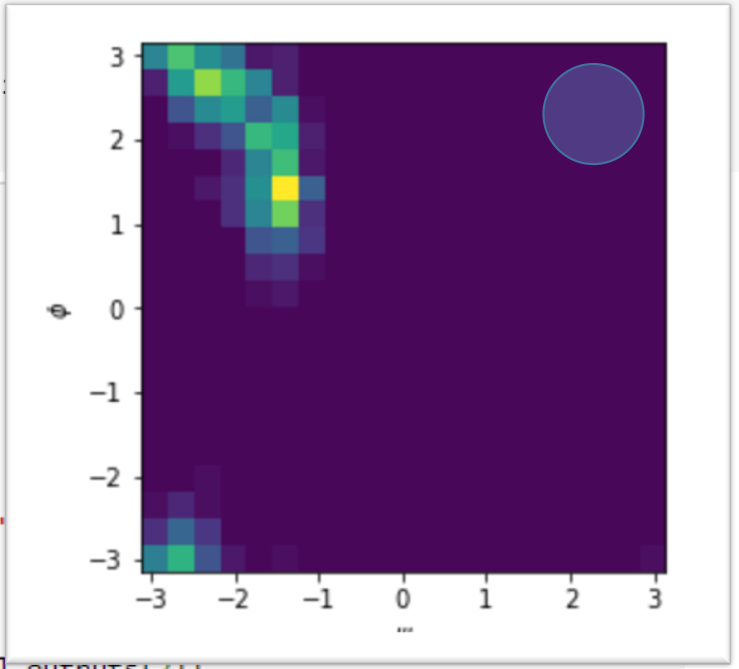
```
# example of running dialanine in lammps with parsl and docker, writing dihedral in plumed

@bash_app(cache=True)
def ru
                                                    parsl-outputs/diala_us.stderr', inputs=[], outputs=[],
                                                    '%s:/home/whitelab/scratch' whitelab/plumed-lammps",
    in
    ru
    re
                                                    me(
pull_r
prod_r

for fi
    pu
    pr
    #m
    pu
                                                    txt'
    )

    pr
                                                    [-1].outputs[2]],
                                                    txt',prod_prefix+'.data'])
    )
```





FREE ENERGIES FROM NON-EQUILIBRIUM QUENCHING

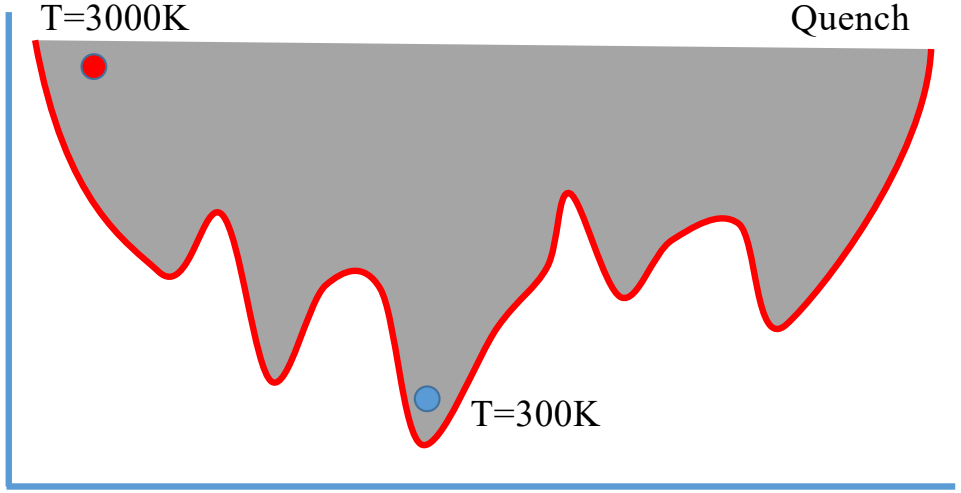
PHYSICAL REVIEW LETTERS **122,** 150602 (2019)

Dynamical Computation of the Density of States and Bayes Factors Using Nonequilibrium Importance Sampling

Grant M. Rotskoff* and Eric Vanden-Eijnden†
Courant Institute, New York University, 251 Mercer Street, New York, New York 10012, USA

(Received 1 October 2018; revised manuscript received 27 February 2019; published 16 April 2019)

$\mathcal{G}$



Algorithm

Step 1: Run several simulations at high temperature and collect initial configurations

Step 2: Run *independent* quench simulations from starting points and gather energy as a function of time $\begin{cases} \frac{dQ}{dt} = P \\ \frac{dP}{dt} = -\nabla U(Q) - \gamma P \end{cases}$

Step 3: Analyze quench data

Python App:

```
def run_quench(command_file, input_file,gamma=0.001, nsteps=2000, … ):
    from lammps import lammps
    import os
    lmp = lammps()
    …
    …
    lmp.command("thermo %d"%thermo_freq)
    lmp.command("fix 1 all quench_exponential %f"%gamma)
    lmp.command("run %d"%nsteps)

    return natoms,log_prefix
```

PARSL QUENCH WORKFLOW

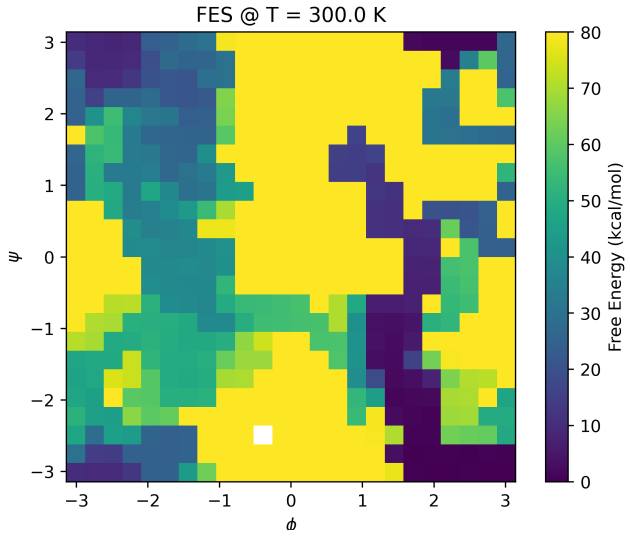
3 Step Workflow

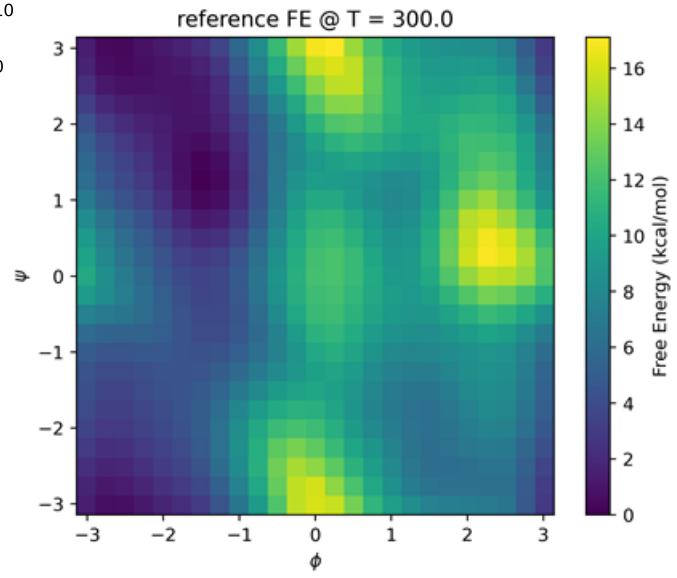
- **python run_highT_alanine.py input.yaml**
- **python run_quench_alanine.py input.yaml**
- **python analyze_quench_alanine.py input.yaml**

```
#YAML FILE
highT:
    input_file: "diala_start.data"
    command_file: "setup_diala_pylammps.lmp"
    nsims: !!int 10
    run_temp: !!float 3000
    eq_steps: !!int 50000
    run_steps: !!int 500000
    restart_freq: !!int 50000
    langevin_gamma: !!float 1.0

quench:
    quench_gamma : !!float 0.0005
    quench_steps : !!int 100000 #10000

analysis:
    target_temp : !!float 300.0
```





Results from Kangxin Liu

CONCLUSIONS AND FUTURE WORK

- Conclusions
  - Parsl enables parallel execution of independent MD simulations using LAMMPS python interface
  - Rapid prototyping of first molecular free energies from quench method
  - Easily implemented first ever combination of umbrella sampling with quenching

- Future work and possible needs
  - Better debugging for large workflows and catching LAMMPS errors
  - Better organization of input/output data sets + staging
  - Containerization for running on different machines