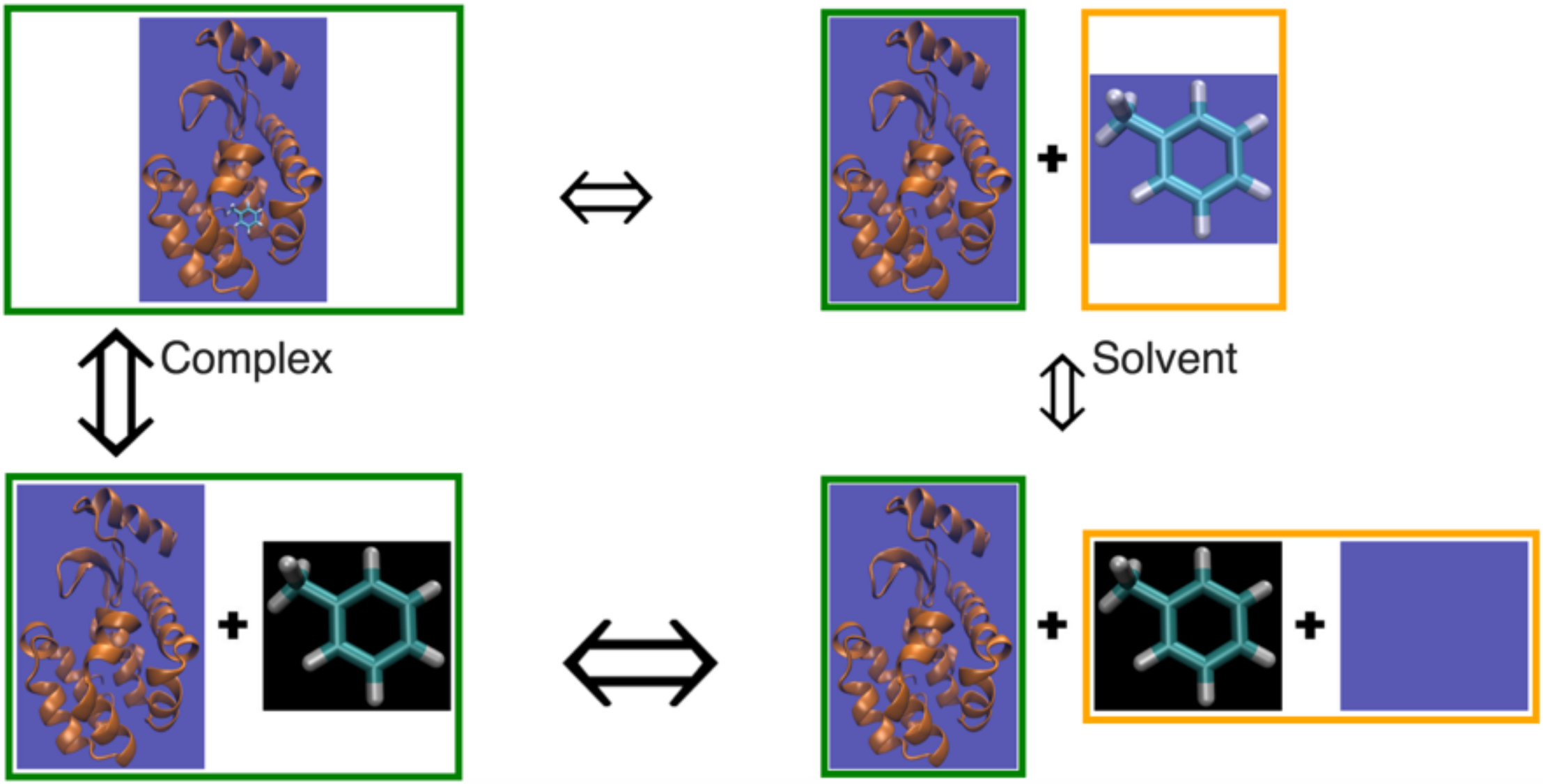
Run free energy calculations with YANK

MSM workshop @ Novartis, Boston - 5/27/2016

Andrea Rizzi

What does YANK do?



0) Install YANK

1. Install miniconda (or anaconda)

   http://conda.pydata.org/miniconda.html

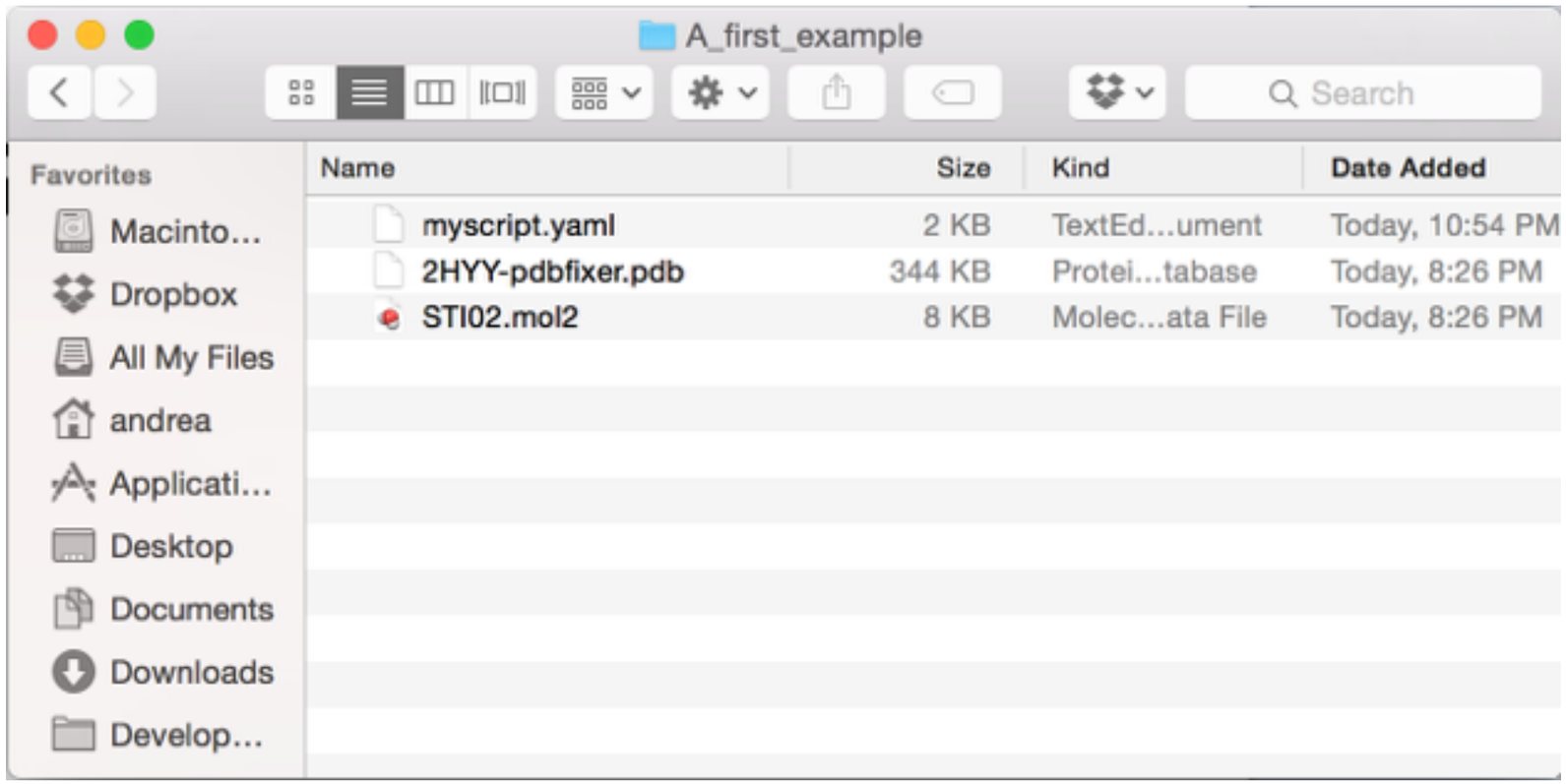
2. Install yank

```
conda config --add channels http://conda.binstar.org/omnia
conda install yank
```

0.a) Why miniconda?

- It's so easy
- Automatically resolve dependencies
- It doesn't mess up your system
  - Installs everything in your home folder, no admin rights required (i.e. works on clusters)
  - You can create parallel environments to avoid conflicts
- Can deploy essentially anything, not just Python

1) Get all the input files



- YANK configuration file (will see details later)
- Receptor structure
- Ligand structure

2) Run

```
yank script --yaml myscript.yaml
```

Automatically set up system
Hamiltonian Replica Exchange

3) Analyze results

```
yank analyze --store experiment/
```

MBAR

First example: YAML script

```yaml
---
molecules:
  Abl:
    filepath: 2HHY-pdbfixer.pdb
    parameters: leaprc.ff14SB
  imatinib:
    filepath: STI02.mol2
    parameters: openeye:am1bcc-gaff

solvents:
  RF: # reaction field
    nonbondedMethod: CutoffPeriodic
    nonbondedCutoff: 9*angstroms
    clearance: 9*angstroms
    positive_ion: Na+
    negative_ion: Cl-

experiment:
  components:
    receptor: Abl
    ligand: imatinib
    solvent: RF
  protocol: absolute-binding

protocols:
  absolute-binding:
    ... # more on this later

options:
  ... # more on this later
```

First example: YAML script

```yaml
---
molecules:
  Abl:
    filepath: 2HHY-pdbfixer.pdb
    parameters: leaprc.ff14SB
  imatinib:
    filepath: STI02.mol2
    parameters: openeye:am1bcc-gaff

solvents:
  RF: # reaction field
    nonbondedMethod: CutoffPeriodic
    nonbondedCutoff: 9*angstroms
    clearance: 9*angstroms
    positive_ion: Na+
    negative_ion: Cl-

experiment:
  components:
    receptor: Abl
    ligand: imatinib
    solvent: RF
  protocol: absolute-binding

protocols:
  absolute-binding:
    ... # more on this later

options:
  ... # more on this later
```

First example: YAML script

```yaml
---
molecules:
  Abl:
    filepath: 2HHY-pdbfixer.pdb
    parameters: leaprc.ff14SB
  imatinib:
    filepath: STI02.mol2
    parameters: openeye:am1bcc-gaff

solvents:
  RF: # reaction field
    nonbondedMethod: CutoffPeriodic
    nonbondedCutoff: 9*angstroms
    clearance: 9*angstroms
    positive_ion: Na+
    negative_ion: Cl-

experiment:
  components:
    receptor: Abl
    ligand: imatinib
    solvent: RF
  protocol: absolute-binding

protocols:
  absolute-binding:
    ... # more on this later

options:
  ... # more on this later
```

http://docs.eyesopen.com/toolkits/cookbook/python/modeling/am1-bcc.html

First example: YAML script

```yaml
---
molecules:
  Abl:
    filepath: 2HHY-pdbfixer.pdb
    parameters: leaprc.ff14SB
  imatinib:
    filepath: STI02.mol2
    parameters: antechamber

solvents:
  RF: # reaction field
    nonbondedMethod: CutoffPeriodic
    nonbondedCutoff: 9*angstroms
    clearance: 9*angstroms
    positive_ion: Na+
    negative_ion: Cl-

experiment:
  components:
    receptor: Abl
    ligand: imatinib
    solvent: RF
  protocol: absolute-binding

protocols:
  absolute-binding:
    ... # more on this later

options:
  ... # more on this later
```

```
# Parameterize ligand from Tripos mol2.
echo "Parameterizing ligand with GAFF and AM1-BCC charges..."
antechamber -fi mol2 -i pxylene.mol2 -fo mol2 -o pxylene.gaff.mol2 -c bcc
parmchk -i pxylene.gaff.mol2 -o pxylene.gaff.frcmod -f mol2
```

First example: YAML script

```yaml
---
molecules:
  Abl:
    filepath: 2HHY-pdbfixer.pdb
    parameters: leaprc.ff14SB
  imatinib:
    filepath: STI02.mol2
    parameters: openeye:am1bcc-gaff

solvents:
  RF: # reaction field
    nonbondedMethod: CutoffPeriodic
    nonbondedCutoff: 9*angstroms
    clearance: 9*angstroms
    positive_ion: Na+
    negative_ion: Cl-

experiment:
  components:
    receptor: Abl
    ligand: imatinib
    solvent: RF
  protocol: absolute-binding

protocols:
  absolute-binding:
    ... # more on this later

options:
  ... # more on this later
```

```
# Create AMBER prmtop/inpcrd files.
echo "Creating AMBER prmtop/inpcrd files..."
tleap -f setup.leap.in > setup.leap.out
```

setup.leap.in

```
# Load AMBER '14 forcefield for protein.
source leaprc.ff14SB

# Load GAFF parameters.
source leaprc.gaff

# Load parameters for ligand.
loadAmberParams STI02.gaff.frcmod

ligand = loadMol2 STI02.gaff.mol2

receptor = loadPdb 2HHY-pdbfixer.pdb

...
```

First example: YAML script

```yaml
---
molecules:
  Abl:
    filepath: 2HHY-pdbfixer.pdb
    parameters: leaprc.ff14SB
  imatinib:
    filepath: STI02.mol2
    parameters: openeye:am1bcc-gaff

solvents:
  RF: # reaction field
    nonbondedMethod: CutoffPeriodic
    nonbondedCutoff: 9*angstroms
    clearance: 9*angstroms
    positive_ion: Na+
    negative_ion: Cl-

experiment:
  components:
    receptor: Abl
    ligand: imatinib
    solvent: RF
  protocol: absolute-binding

protocols:
  absolute-binding:
    ... # more on this later

options:
  ... # more on this later
```

```
# Create AMBER prmtop/inpcrd files.
echo "Creating AMBER prmtop/inpcrd files..."
tleap -f setup.leap.in > setup.leap.out
```

setup.leap.in

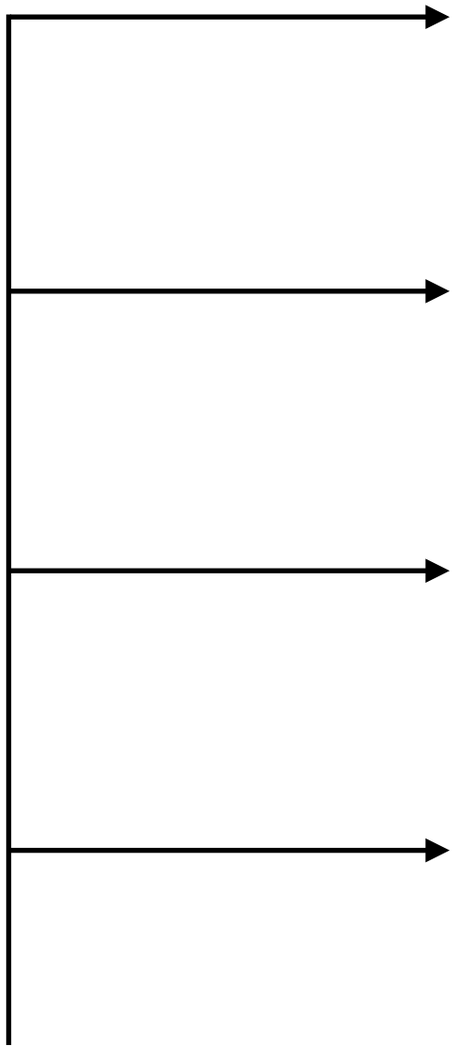
```
...

complex = combine { receptor ligand }

solvatebox complex TIP3PBOX 9.0 iso

solvatebox ligand TIP3PBOX 9.0 iso

saveamberparm complex complex.prmtop complex.inpcrd

saveamberparm ligand solvent.prmtop solvent.inpcrd

...
```

First example: YAML script

```yaml
---
molecules:
  ...

solvents:
  ...

experiment:
  components:
    ...
  protocol: absolute-binding

options:
  minimize: yes
  verbose: yes
  mpi: yes
  number_of_iterations: 1000
  output_dir: ./
  temperature: 300*kelvin
  pressure: 1*atmosphere
  softcore_beta: 0.0

protocols:
  absolute-binding:
    phases:
      complex:
        alchemical_path:
          lambda_electrostatics: [1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0, ...]
          lambda_sterics: [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9, 0.8, ...]
      solvent:
        alchemical_path:
          lambda_electrostatics: [1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0, ...]
          lambda_sterics: [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9, 0.8, ...]
```

Combinations of experiments

```yaml
---
molecules:
  abl:
    ...
  src:
    ...
  imatinib:
    ...
  bosutinib:
    ...

solvents:
  implicit:
    ...
  explicit:
    ...

experiment:
  components:
    receptor: [abl, src]
    ligand: [imatinib, bosutinib]
    solvent: [implicit, explicit]
  options:
    timestep: [1.0*femtosecond, 2.0*femtosecond]
```



```yaml
experiment:
  components:
    receptor: abl
    ligand: imatinib
    solvent: implicit
  options:
    timestep: 1.0*femtosecond
```

```yaml
experiment:
  components:
    receptor: abl
    ligand: imatinib
    solvent: explicit
  options:
    timestep: 1.0*femtosecond
```

```yaml
experiment:
  components:
    receptor: abl
    ligand: imatinib
    solvent: implicit
  options:
    timestep: 2.0*femtosecond
```

```yaml
experiment:
  components:
    receptor: src
    ligand: bosutinib
    solvent: explicit
  options:
    timestep: 2.0*femtosecond
```

...

More options

```yaml
options:
  verbose: true
  mpi: yes
  resume_setup: true
  resume_simulation: yes
  output_dir: path/to/outputdir
  temperature: 300*kelvins
  pressure: 1*atmosphere
  constraints: AllBonds
  hydrogen_mass: 2*amus                          # hydrogen mass repartitioning
  restraint_type: harmonic
  randomize_ligand: yes
  randomize_ligand_sigma_multiplier: 2.0         # ligand randomization maximum displacement
  randomize_ligand_close_cutoff: 1.5*angstrom    # ligand randomization threshold for overlapping atoms
 collision_rate: 5.0 / picosecond                # collision rate for Langevin dynamics
  constraint_tolerance: 1.0e-6
  timestep: 2.0*femtosecond
  nsteps_per_iteration: 2500
  number_of_iteration: 1000
  equilibration_timestep: 1.0*femtosecond
  number_of_equilibration_iterations: 100
  minimize: False
  minimize_tolerance: 1.0 * kilojoules_per_mole / nanometers
  minimize_max_iterations: 100
  replica_mixing_scheme: swap-all

  ...
```

https://github.com/choderalab/yank/blob/master/examples/yank-yaml-cookbook/all-options.yaml

More molecules

```
molecules:
  Abl_combinatorial:
    filepath: [2HHY-pdbfixer.pdb, 3CS9-pdbfixer.pdb] # We are working on a protein pipeline
    parameters: [leaprc.ff14SB, oldff/leaprc.ff99SBildn]
  benzene_sdf:
    filepath: benzene.sdf
    parameters: antechamber
benzene_smiles:
  smiles: c1ccccc1
  parameters: antechamber
benzene_name:
  name: benzene
  parameters: antechamber
benzene_protonation:
  name: benzene
  parameters: antechamber
  epik: 1
```

More solvents

```
solvents:
  vacuum:
    nonbondedMethod: NoCutoff
  GBSA-OBC2:
    implicitSolvent: OBC2
PME:
  nonbondedMethod: PME
  nonbondedCutoff: 1*nanometer
  clearance: 10*angstroms
ReactionField:
  nonbondedMethod: CutoffNonPeriodic
  nonbondedCutoff: 1*nanometer
  clearance: 10*angstroms
```

OpenMM 7.0 Benchmarks

Performance measured in ns/day. Details.

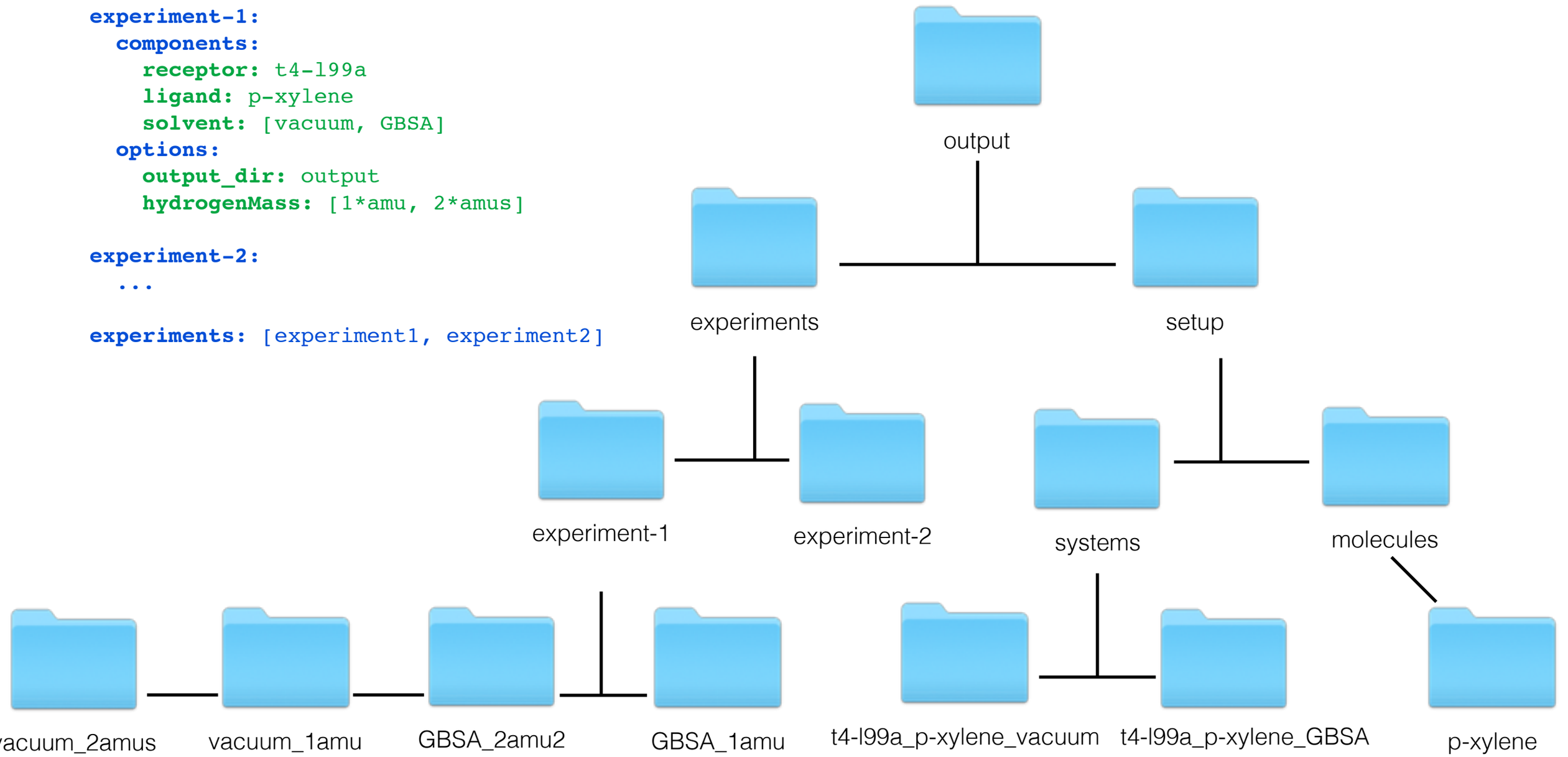
Simulation Type	CUDA (GTX Titan X)	OpenCL (GTX Titan X)	OpenCL (Radeon R9 Fury)	CPU (i7-2700K)
Implicit, 2 fs	482	361	195	8.5
Implicit, 5 fs HMR	756	592	258	20.4
Explicit-RF, 2 fs	309	266	154	16.5
Explicit-RF, 5 fs HMR	542	470	216	38.0
Explicit-PME, 2 fs	204	154	81	12.8
Explicit-PME, 5 fs HMR	415	330	144	30.2

Sequences of experiments

```yaml
experiment-abl:
  components:
    receptor: abl
    ligand: [imatinib, regorafenib, cabozantinib]
    solvent: [implicit, explicit]
  options:
    number_of_iteration: 1000

experiment-src:
  components:
    receptor: src
    ligand: [imatinib, regorafenib, cabozantinib]
    solvent: [implicit, explicit]
  options:
    number_of_iteration: 2000

experiments: [experiment-abl, experiment-src]
```

Output folder structure

```yaml
experiment-1:
  components:
    receptor: t4-l99a
    ligand: p-xylene
    solvent: [vacuum, GBSA]
  options:
    output_dir: output
    hydrogenMass: [1*amu, 2*amus]

experiment-2:
  ...

experiments: [experiment1, experiment2]
```



Resume

```
options:
  resume_setup: yes
  resume_simulation: no
```

- Initial dry run to check beforehand if the experiment will be interrupted but it still can happen

- YANK does not overwrite anything, you need to remove the files if you want them to be generated again

More docs

- Main docs: http://getyank.org/latest/
- Examples: https://github.com/choderalab/yank/tree/master/examples

```
yank script skeleton
```

- YAML cookbook: https://github.com/choderalab/yank/tree/master/examples/yank-yaml-cookbook
- This presentation: andrea.rizzi@choderalab.org

Near future

- CHARMM force field (Chaya Stern)
- Protein setup automatic pipeline
- Easy movie
- Solvation free energy
- Clustering binding modes (Rosa Luirink, Daan Geerke group in University of Amsterdam)
- Relative free energy calculations (*semi-near* future)