ParmEd

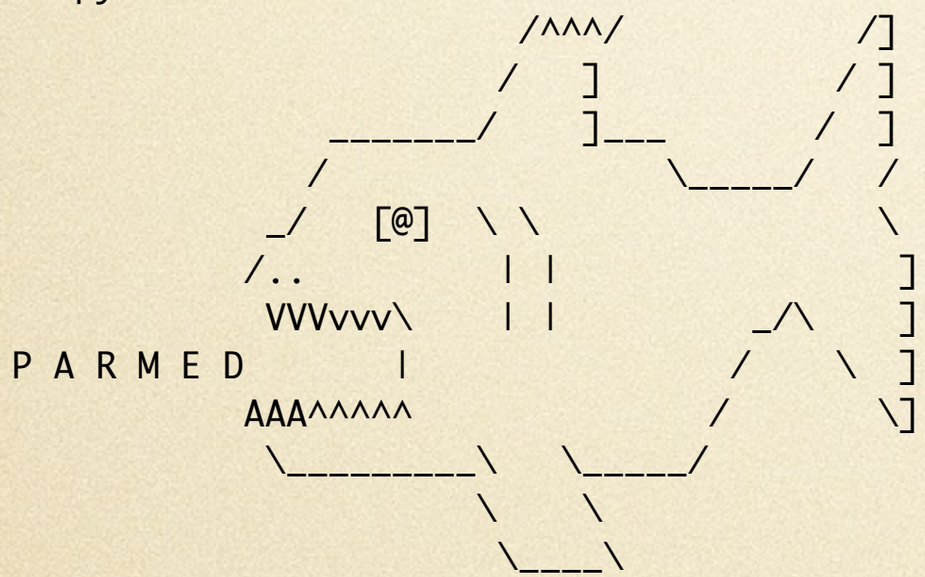
MSKCC, April 10, 2015

Jason Swails

ParmEd: What is it?

Standalone programs

```
bash$ parmed.py
```




```
ParmEd: a Parameter file Editor


Reading input from STDIN...
> help

Documented commands (type help <topic>):
========================================
EOF                 changeRadii      minimize        quit
HMassRepartition    checkValidity    netCharge       scale
OpenMM              defineSolvent    outCIF          scee
add12_6_4           deleteBond       outPDB          scnb
addAtomicNumber     deleteDihedral   outparm         setAngle
[snip]
```

API

Easily instantiate and manipulate full MM descriptions of chemical systems.

Support wide range of force fields.

OpenMM integration.

Robust format parsers.

ParmEd: API Design

Parsers / Writers

PDB File

PSF file
(and parameters)

prmtop

PDBFile

CharmmPsfFile

AmberParm

PDBx/
mmCIF

CIFFile

mol2

Mol2File

Structure base class

Planned /
in progress

...

System +
Topology

GromacsTopFile

OMM System and Topology

Gromacs top / itp

ParmEd: Structure

Structure

`residues`: list of `Residue`

`atoms`: list of `Atom`

`bonds`: list of `Bond`

`angles`: list of `Angle`

`dihedrals`: list of `Dihedral`

`impropers`: list of `Improper`

… etc.

`bond_types`: list of `BondType`

`angle_types`: list of `AngleType`

`dihedral_types`:
list of `DihedralType`

`improper_types`:
list of `ImproperType`

… etc.

→ Create OpenMM System / Topology

ParmEd: The Atom class

Atom

Atomic property attributes			
name	type	charge	atomic_number
mass	epsilon	rmin	radii
screen	bfactor	altloc	anisou

Topological attributes		
bond_partners	angle_partners	dihedral_partners
bonds	angles	dihedrals
residue	exclusion_partners	

ParmEd: The `Residue` class

Residue

Attributes		
`name`	`number`	`chain`
`insertion_code`	`atoms`	

Iterator / Container Behavior		
`__iter__`	`__len__`	`__contains__`
`__getitem__`		

Behave like containers of `Atom` instances

ParmEd: The Bond class

Bond

Attributes

`atom1`	`Atom` instance of first atom in bond
`atom2`	`Atom` instance of second atom in bond
`type`	`BondType` defining the parameter

`__contains__:atom in bond`

ParmEd: The `Angle` class

Angle

Attributes	
`atom1`	`Atom` instance of first atom in angle
`atom2`	`Atom` instance of middle atom in angle
`atom3`	`Atom` instance of third atom in angle
`type`	`AngleType` defining the parameter

`__contains__`: `atom in angle; bond in angle`

Other composite terms behave similarly (i.e., they have `atom1`, `atom2`, `atom3`, …, `type`, and `__contains__` both `Atom` and `Bond` instances

ParmEd: Automated Bookkeeping

- References to the same "thing" always point to the same instance

```
>>> parm.bonds[0].atom1
<Atom C [11]; In SER 0>
>>> parm.bonds[0].atom1 is parm.atoms[11]
True
```

  - When part of a list, all topology and parameter objects have an `idx` attribute that gives their exact index in that list, and it is always up-to-date, even if the underlying list is changed.

```
>>> atom = parm.atoms[10]
>>> atom.idx
10
```

```
>>> del parm.atoms[:2]
>>> atom.idx
8
```

ParmEd: Automatic Format Identification

```
import chemistry
obj = chemistry.load_file('your_file')
```

- Amber prmtop
- Charmm PSF
- Charmm Coordinate*
- Charmm Restart*
- PDB
- PDBx/mmCIF
- Mol2
- Amber OFF
- Amber mdcrd
- NetCDF (rst and traj)

*Read only

ParmEd: Gotchas

- Deleting `Atom`s from `Structure` does not delete valence terms including them — call `prune_empty_terms` (called automatically by functions using full `Structure` definition)

- Various terms *can* share parameter types, so modifying `bonds[n].type` *can* also modify `bonds[m].type`

`ParmEd`: `ParmedActions`

- All ParmEd `Actions` are available through an API
- It implements a series of high-level Structure modifications of common interest
- Primarily aimed at Amber topologies (support for general `Structure` classes is improving)
- Works around any “Gotchas”

ParmEd: Future Plans

- Gromacs parsers
- OpenMM `System`/`Topology` —> `Structure`
- Modeling and parametrization capabilities (like PDBFixer and OpenMM's modeling engines) supporting native Charmm, Amber, and Gromacs files.