

# OpenMM, tools, and more

----

## Setting up and running your own system

Gert Kiss

OpenMM Workshop

Stanford University

March 2014

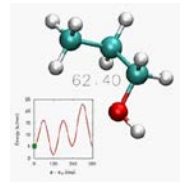
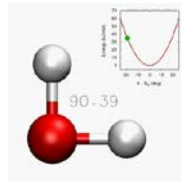
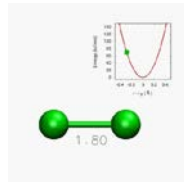
# Things to consider

- Format of your files:

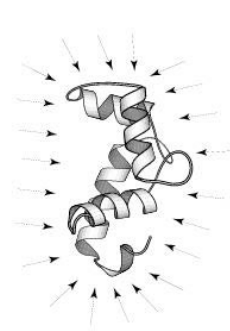
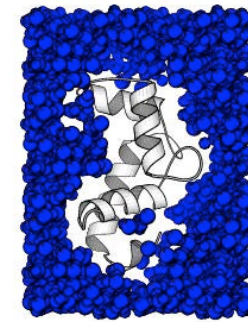


**GROMACS**  
Groningen Machine for Chemical Simulations

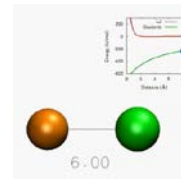
- Force field



- Solvent: implicit **v** explicit? => which model?



- Non-bonded interactions



- Integrators

- Constraints vs. restraints

# Overview of the logic behind OpenMM

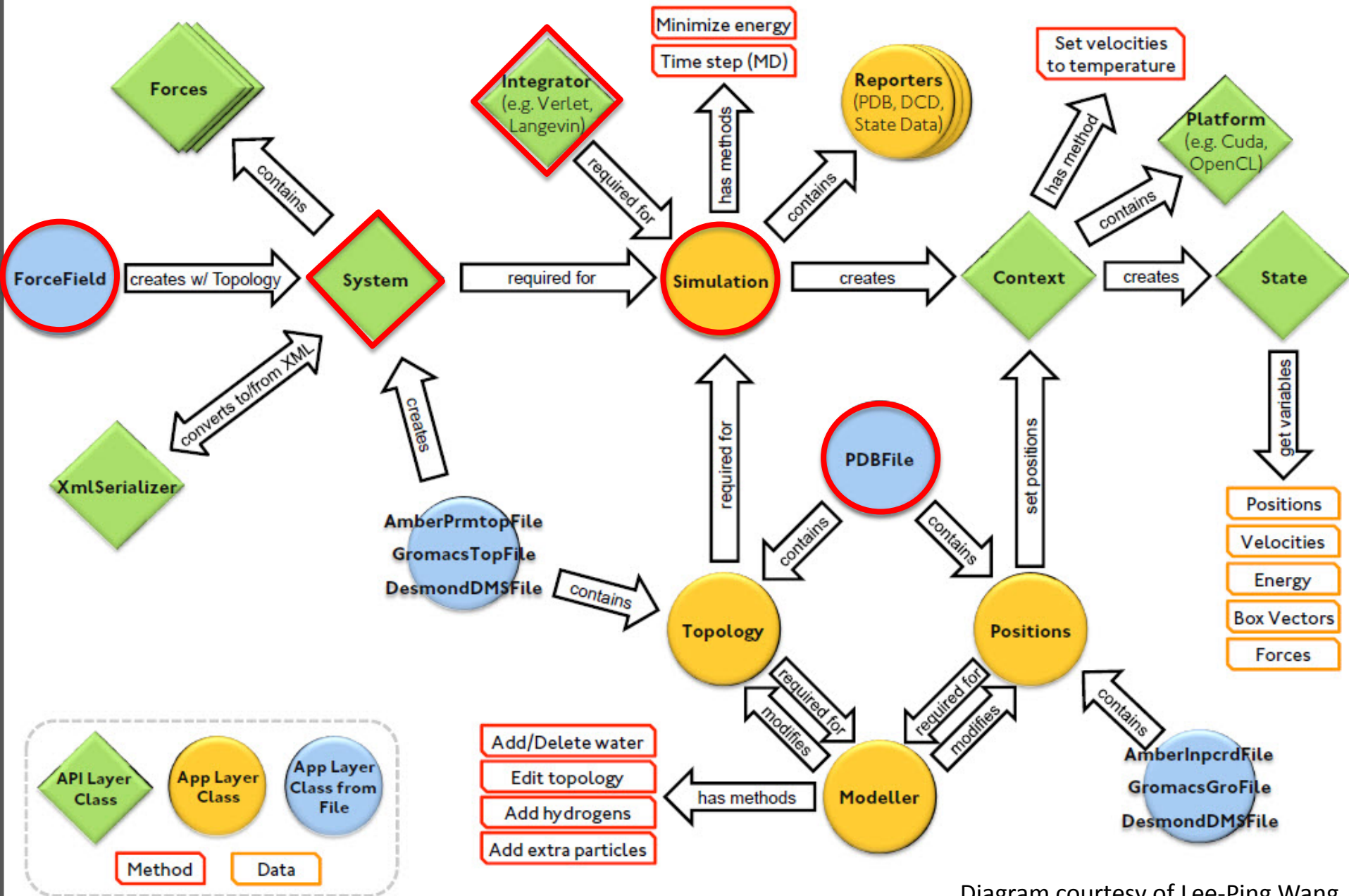
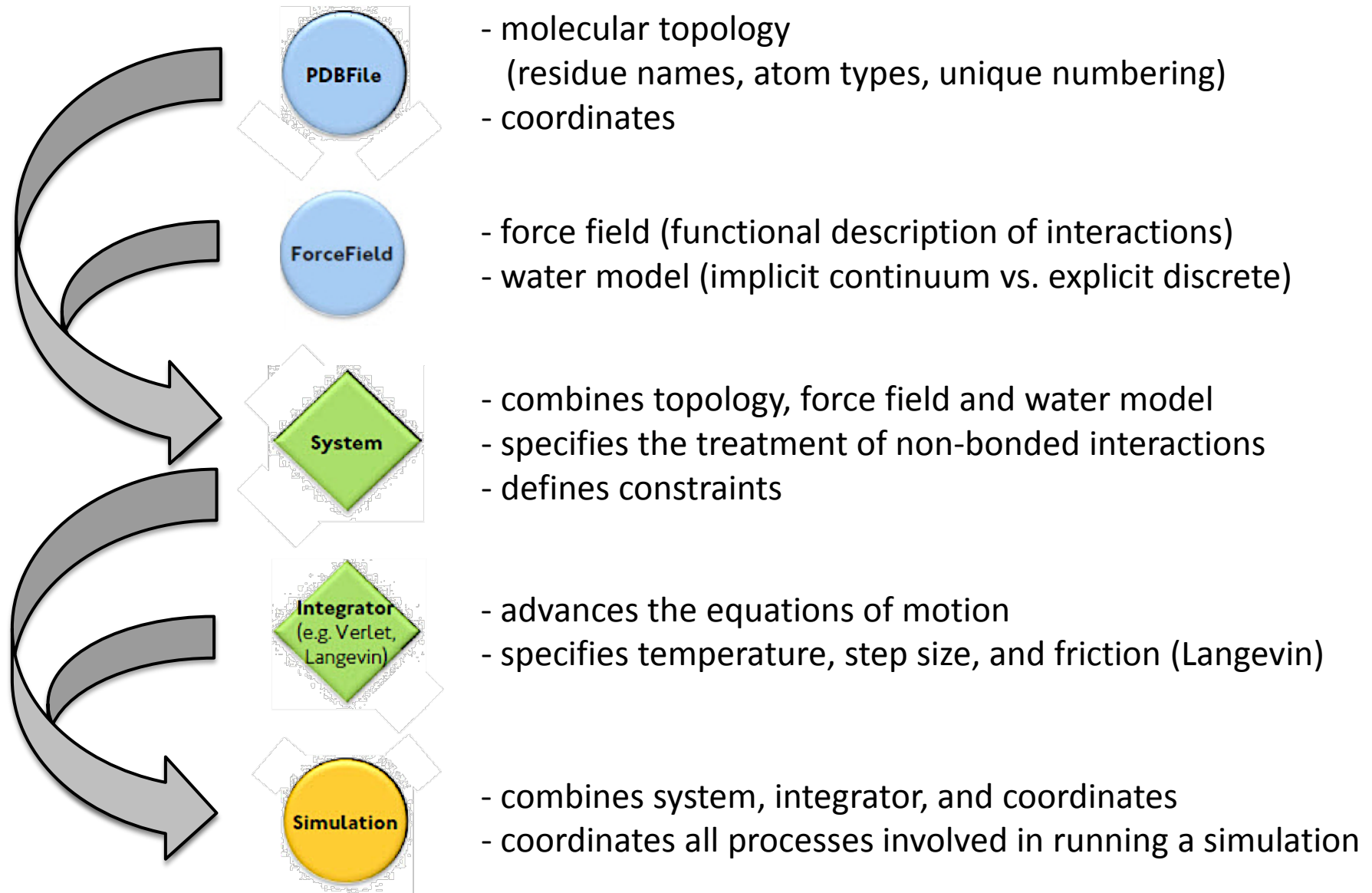


Diagram courtesy of Lee-Ping Wang

# Essential components that we will encounter



# Force fields available in OpenMM

Set with:

```
forcefield = ForceField('amber99sb.xml', 'tip3p.xml')
```

File	Force Field
amber96.xml	AMBER96 <sup>1</sup>
amber99sb.xml	AMBER99 <sup>2</sup> with modified backbone torsions <sup>3</sup>
amber99sbildn.xml	AMBER99SB plus improved side chain torsions <sup>4</sup>
amber99sbnmr.xml	AMBER99SB with modifications to fit NMR data <sup>5</sup>
amber03.xml	AMBER03 <sup>6</sup>
amber10.xml	AMBER10
amoeba2009.xml	AMOEBA <sup>7</sup> (AMOEBA includes its own water model)

New, currently unavailable and custom force fields can be added.  
(check out the docs on how)

# Solvent models

Set with:

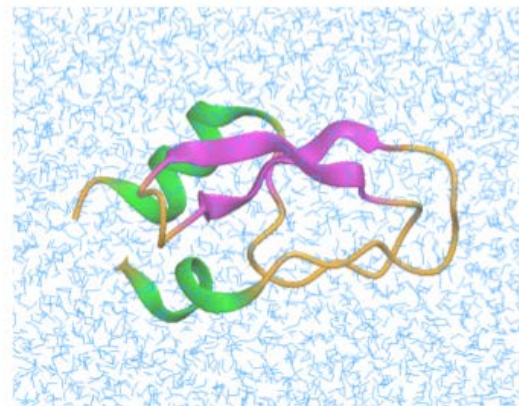
```
forcefield = ForceField('amber99sb.xml', 'tip3p.xml')
```



$$H = H_{\text{prot}} + \Delta G_{\text{solv}}$$

**Implicit water model**

- Fast (fewer atoms to track and less friction)
- Gets bulk properties 'right'



$$H = H_{\text{prot}} + H_{\text{prot-wat}} + H_{\text{wat}}$$

**Explicit water model**

- Slower (water atoms make up most of the system)
- Gets atomistic properties 'right'



# Explicit solvent models ...

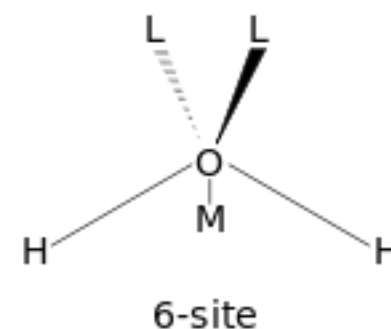
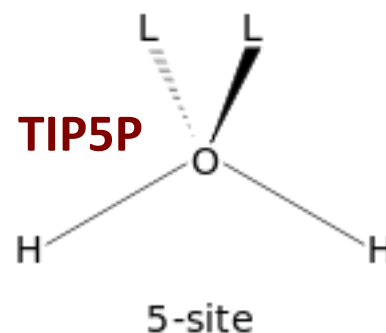
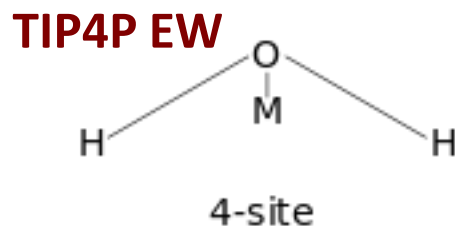
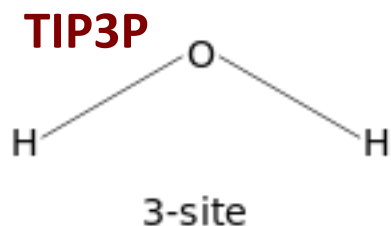
... parameterized to compensate for their simplified description of reality.

## Most models have incomplete physics:

- Fixed point charges (no electronic polarization)
- Classical mechanics (no isotope effects)
- Fixed bond topology (no chemistry)

## However, much can be recovered through parameterization:

- Increase the partial charges, tune vdW parameters, etc.
- In many cases, force fields exceed the accuracy of quantum methods!



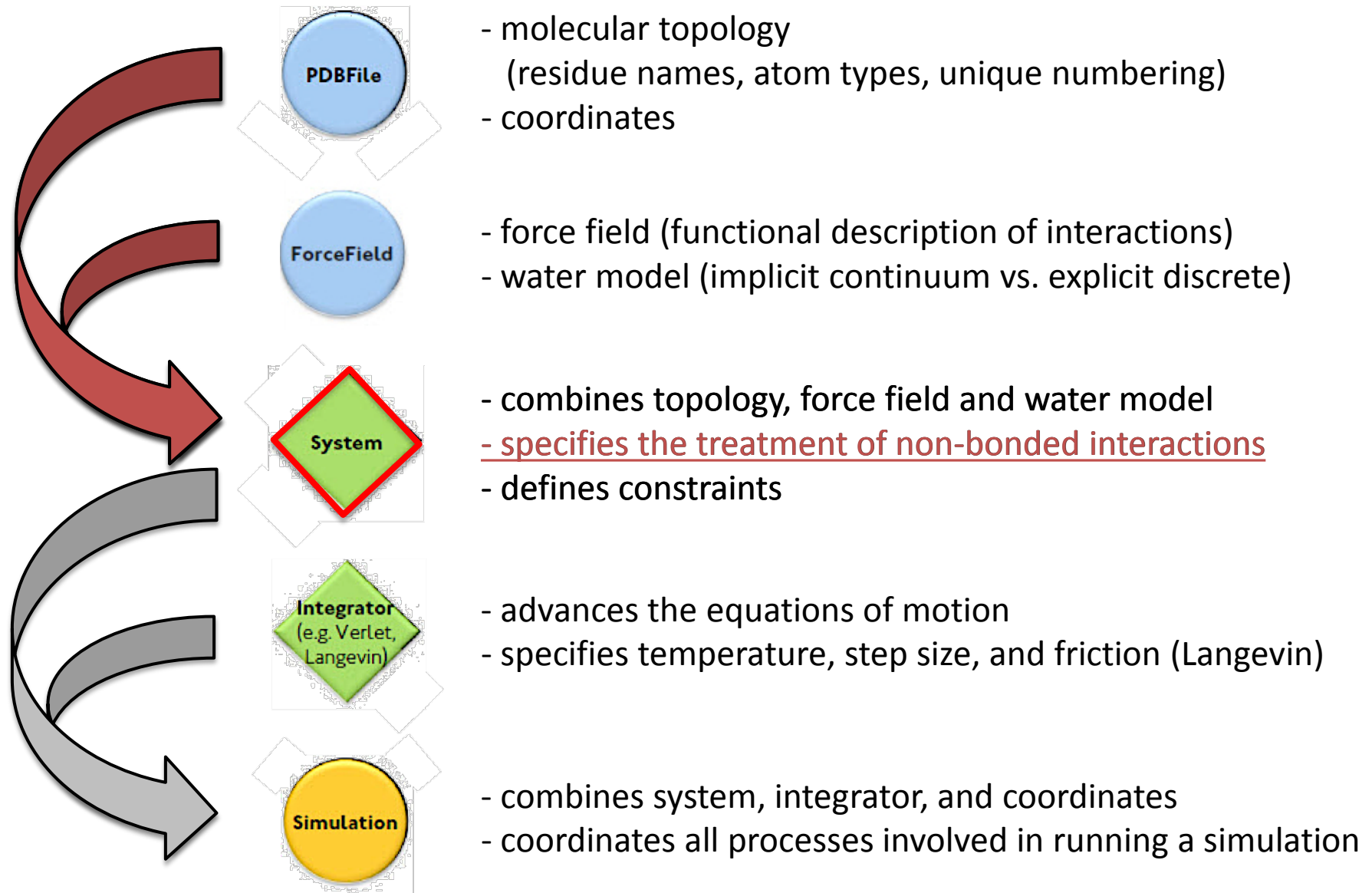
# Explicit and implicit water models in OpenMM

File	Water Model
tip3p.xml	TIP3P water model <sup>8</sup>
tip4pew.xml	TIP4P-Ew water model <sup>9</sup>
tip5p.xml	TIP5P water model <sup>10</sup>
spce.xml	SPC/E water model <sup>11</sup>
swm4ndp.xml	SWM4-NDP water model <sup>12</sup>

File	Implicit Solvation Model
amber96_obc.xml	GBSA-OBC solvation model <sup>13</sup> for use with AMBER96 force field
amber99_obc.xml	GBSA-OBC solvation model for use with AMBER99 force fields
amber03_obc.xml	GBSA-OBC solvation model for use with AMBER03 force field
amber10_obc.xml	GBSA-OBC solvation model for use with AMBER10 force field
amoeba2009_gk.xml	Generalized Kirkwood solvation model <sup>14</sup> for use with AMOEBA force field

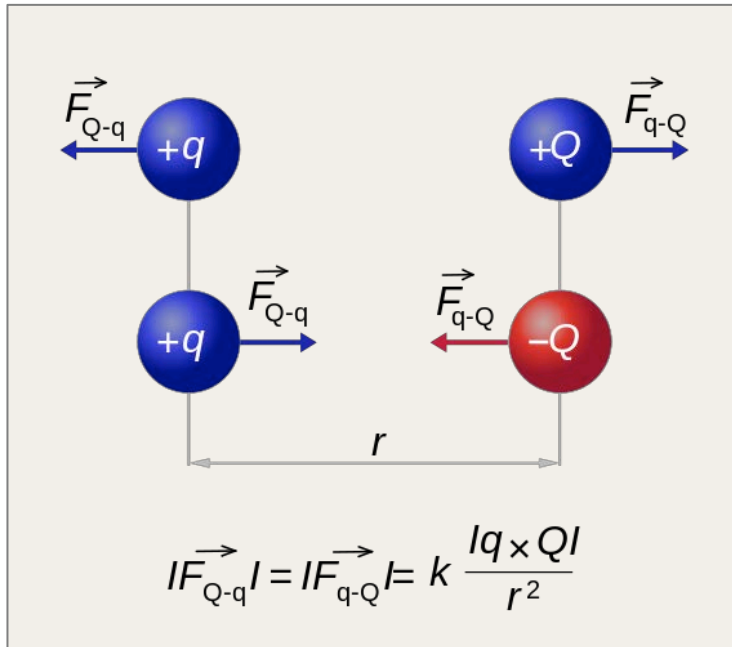


# Next up: Creating the 'system' object



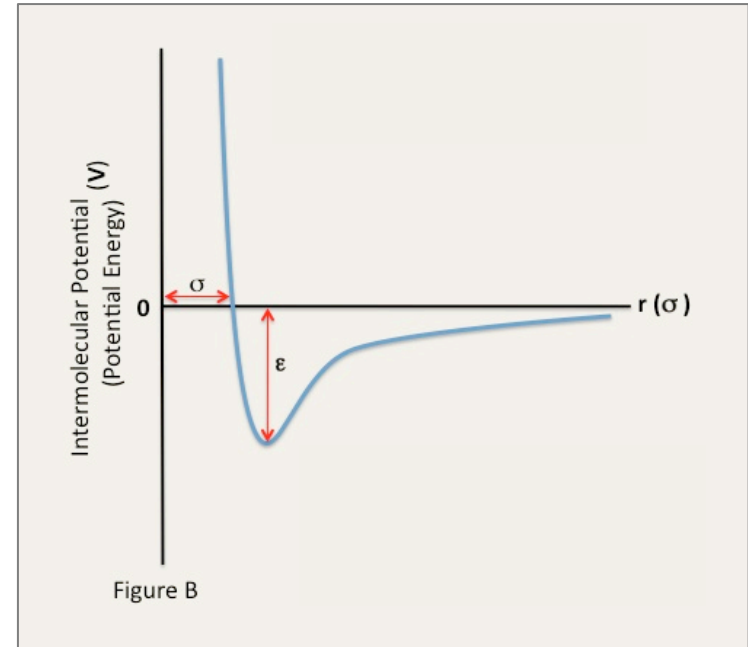
# But first: what are non-bonded interactions?

## Coulomb's Law



for the treatment of point charges

## Lennard Jones Potential



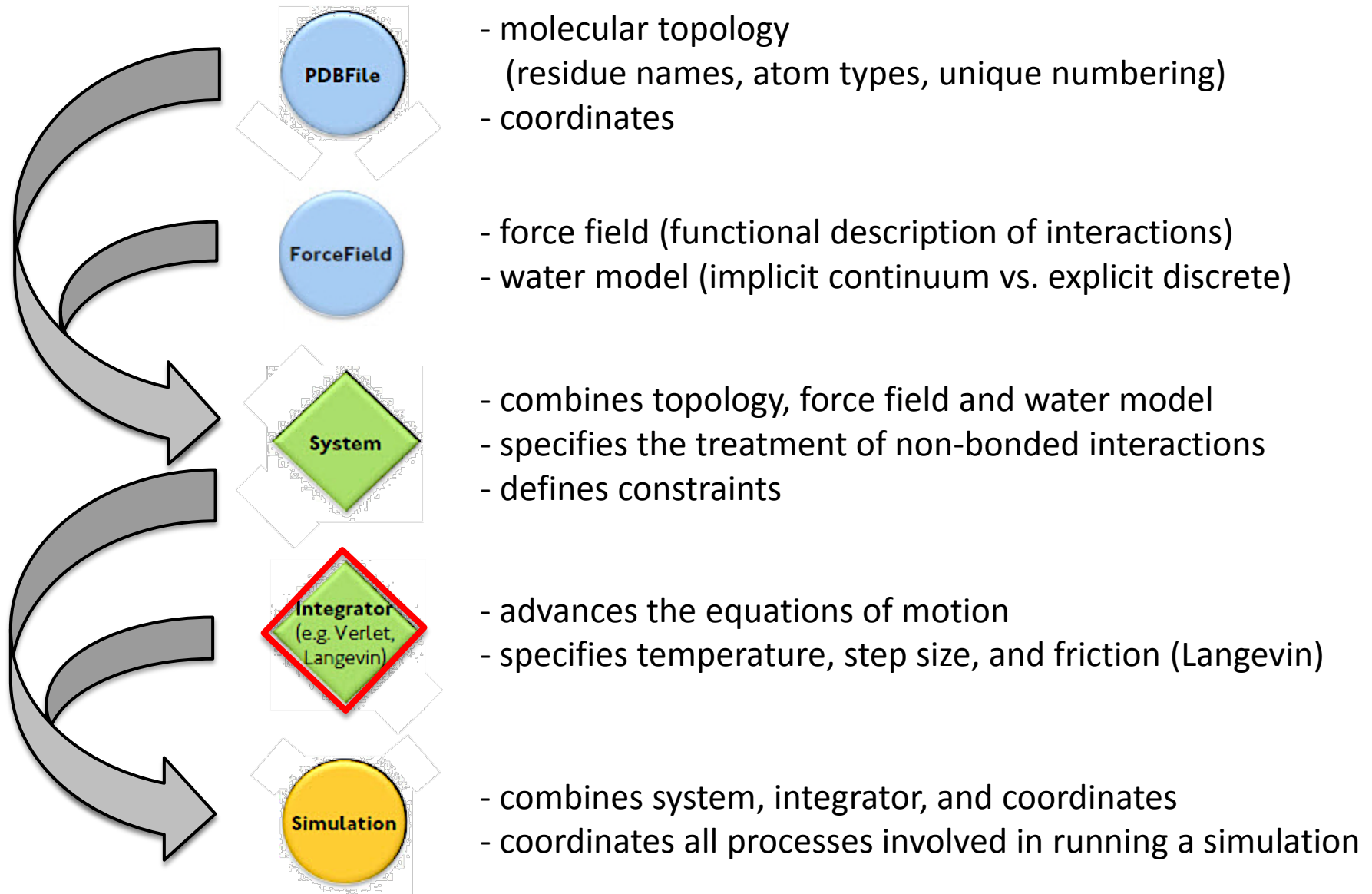
for short range (Pauli) repulsion  
and long range (VdW) attraction

# Treatment of non-bonded interactions

```
system = forcefield.createSystem(pdb.topology, nonbondedMethod=NoCutoff,  
                                constraints=HBonds)
```

Value	Meaning
NoCutoff	No cutoff is applied.
CutoffNonPeriodic	The reaction field method is used to eliminate all interactions beyond a cutoff distance. Not valid for AMOEBA.
CutoffPeriodic	The reaction field method is used to eliminate all interactions beyond a cutoff distance. Periodic boundary conditions are applied, so each atom interacts only with the nearest periodic copy of every other atom. Not valid for AMOEBA.
Ewald	Periodic boundary conditions are applied. Ewald summation is used to compute long range interactions. (This option is rarely used, since PME is much faster for all but the smallest systems.) Not valid for AMOEBA.
PME	Periodic boundary conditions are applied. The Particle Mesh Ewald method is used to compute long range interactions.

# Next up: Creating the integrator



# Statistical mechanical ensembles ...

... allow our simulation to exchange energy with an external environment.

Ensemble menu: Choose one from each row	
Particle number $N$	Chemical potential $m$
Volume $V$	Pressure $P$
Energy $E$	Temperature $T$

$$P_{NVT}(\mathbf{r}) \propto e^{-\frac{E(\mathbf{r})}{k_b T}}$$

Probability of a microstate  
in the *canonical (NVT) ensemble*.

- An *ensemble* represents all of the microstates (i.e. geometries) that are accessible to the simulation, and provides the probability of each microstate.
- An ideal MD simulation conserves the total energy and entropy, and samples the *microcanonical (NVE) ensemble*.
- More realistic systems may exchange energy, volume or particles with external reservoirs
- However, this could make the algorithms more **difficult**

# Integrators available in OpenMM

## Constant temperature

```
integrator = LangevinIntegrator(300*kelvin, 1/picosecond, 0.002*picoseconds)
```

## Constant energy

```
integrator = VerletIntegrator(0.002*picoseconds)
```

## Brownian dynamics

```
integrator = BrownianIntegrator(300*kelvin, 1/picosecond, 0.002*picoseconds)
```

## Variable time step ( continuous adjustment of step size to stay within error tolerance )

```
integrator = VariableLangevinIntegrator(300*kelvin, 1/picosecond, 0.001)
```

```
integrator = VariableVerletIntegrator(0.001)
```

## Temperature coupling ( as an alternative to the Langevin integrator )

```
system.addForce(AndersenThermostat(300*kelvin, 1/picosecond))
```

## Pressure coupling ( for NPT simulations; use with Langevin or Andersen thermostat )

```
system.addForce(MonteCarloBarostat(1*bar, 300*kelvin))
```

# Constraining certain types of bonds and angles

```
system = forcefield.createSystem(pdb.topology, nonbondedMethod=NoCutoff, constraints=HBonds)
```

Value	Meaning	Time step
None	No constraints are applied. This is the default value.	$\leq 1$ fs
HBonds	The lengths of all bonds that involve a hydrogen atom are constrained.	2 fs
AllBonds	The lengths of all bonds are constrained.	
HAngles	The lengths of all bonds are constrained. In addition, all angles of the form H-X-H or H-O-X (where X is an arbitrary atom) are constrained.	Up to 4 fs

**Constraints? => Larger integration time steps => Greater speedup.**

(Note: Be aware of the added level of approximation and apply constraints with care.)

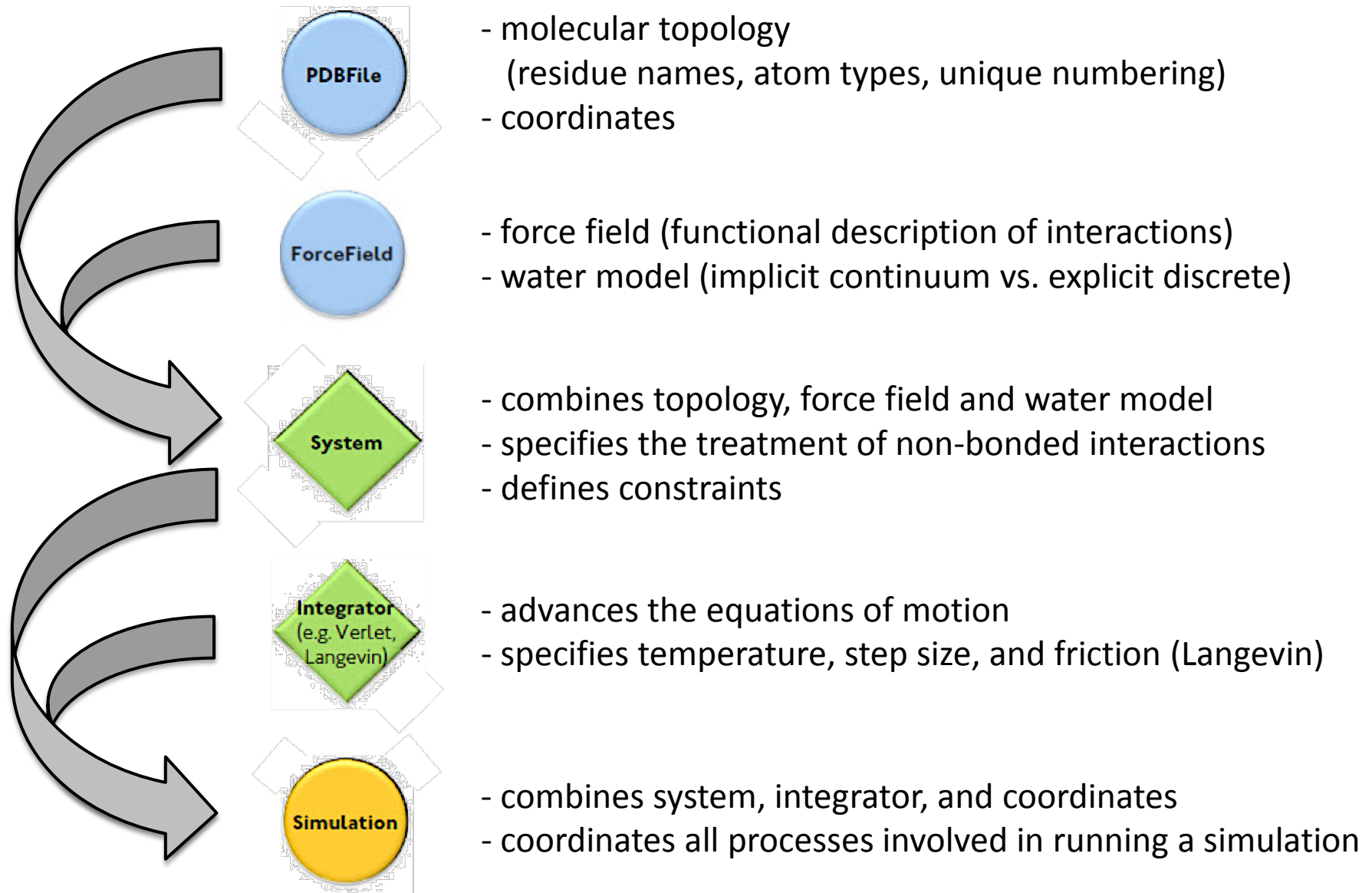
**Hint:** It's good practice to choose 'None' while heating a system. Once equilibrated, one can safely choose 'Hbonds' for production runs.

**By default,** bonds and angles of water molecules are constrained (accessible through rigidWater parameter):

```
system = forcefield.createSystem(pdb.topology, nonbondedMethod=NoCutoff, constraints=None, rigidWater=False)
```

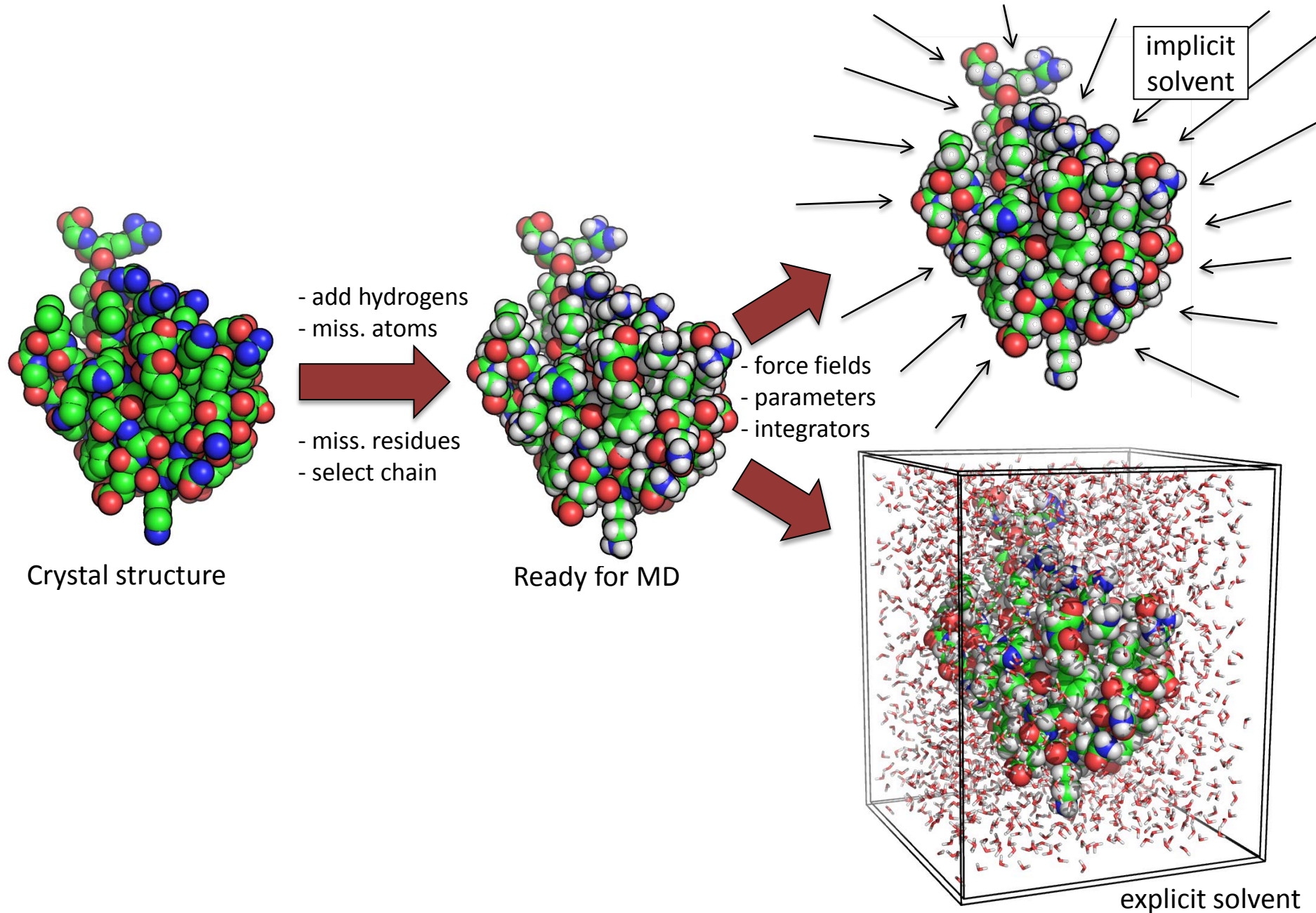


# Summing up the dry stuff



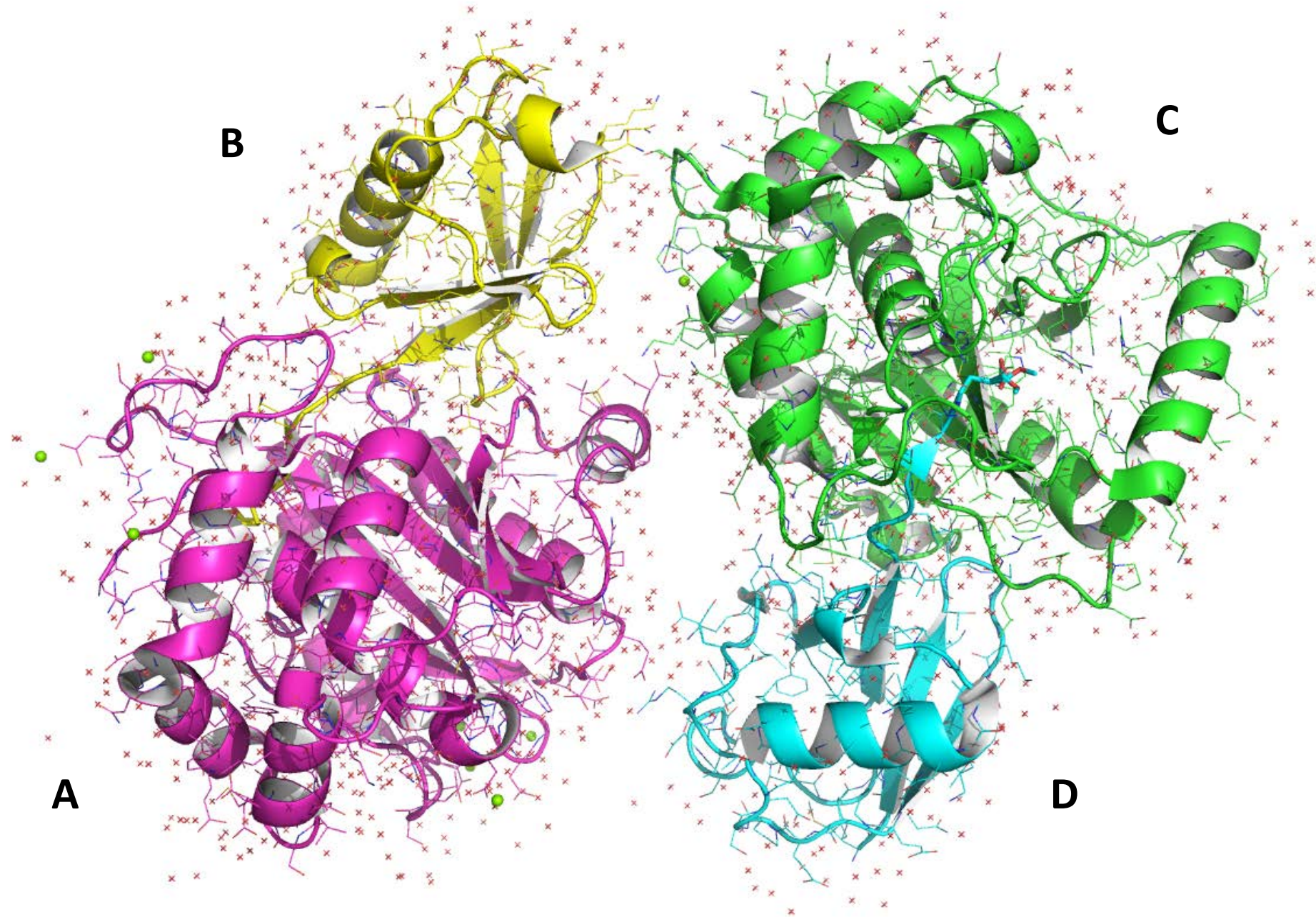
**But now, let's get you started with OpenMM**

# General considerations in the setup of MD sims



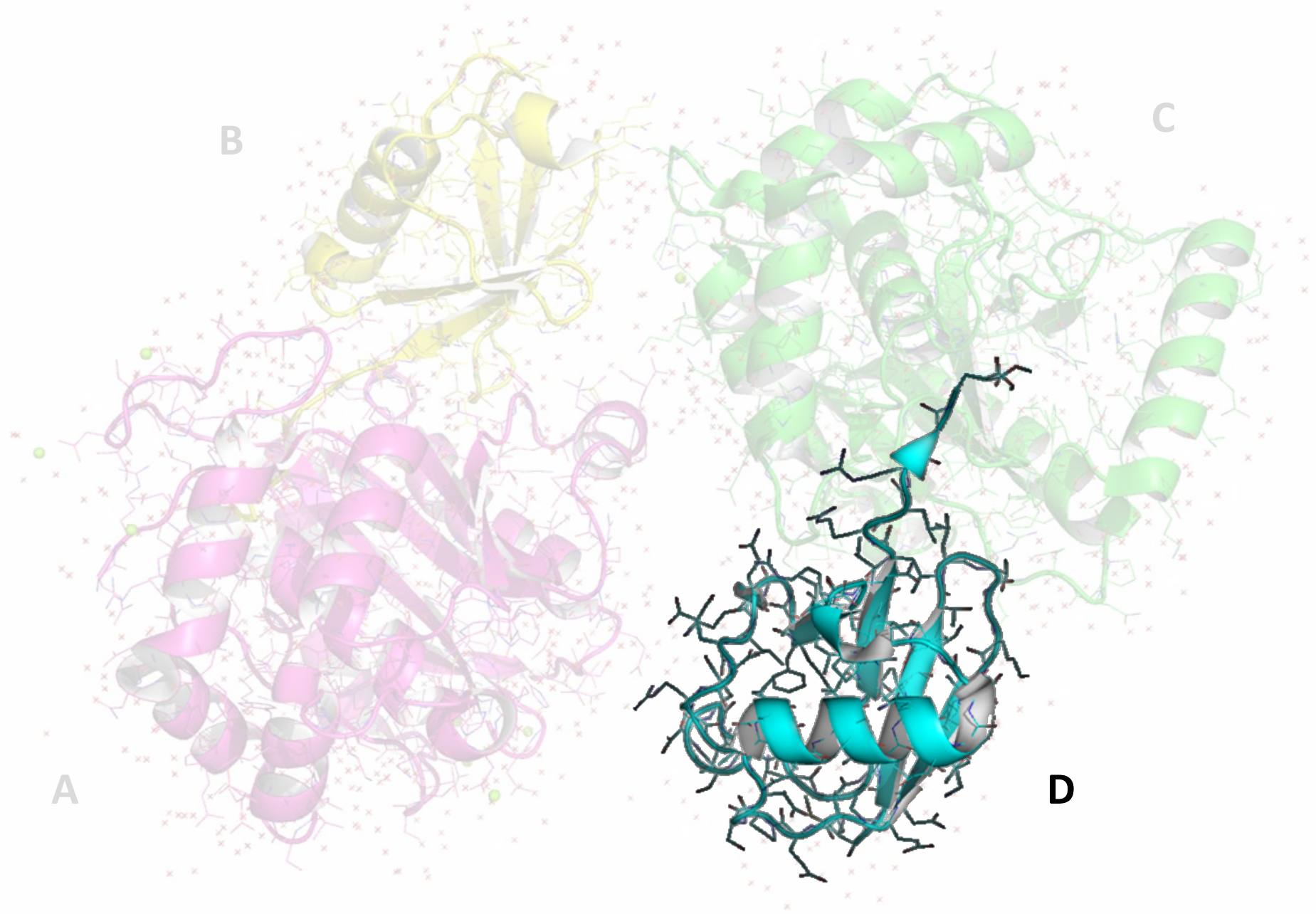


Let's begin with [ 1xd3 ] from the RCSB ([rcsb.org](https://rcsb.org))





# Pull one of the chains from the crystal structure



# Inspect the header info of [ 1xd3 ] ...

... for common problems with pdb files

## REMARK 465 MISSING RESIDUES

REMARK 465 THE FOLLOWING RESIDUES WERE NOT LOCATED IN THE  
REMARK 465 EXPERIMENT. (M=MODEL NUMBER; RES=RESIDUE  
NAME; C=CHAIN

REMARK 465 IDENTIFIER; SSSEQ=SEQUENCE NUMBER; I=INSERTION  
CODE.)

REMARK 465

REMARK 465 M RES C SSSEQI

REMARK 465 MET A 1

REMARK 465 MET C 1

REMARK 465 GLU C 2

REMARK 465 GLY C 3

REMARK 470

## REMARK 470 MISSING ATOM

REMARK 470 THE FOLLOWING RESIDUES HAVE MISSING  
ATOMS(M=MODEL NUMBER;

REMARK 470 RES=RESIDUE NAME; C=CHAIN IDENTIFIER;  
SSEQ=SEQUENCE NUMBER;

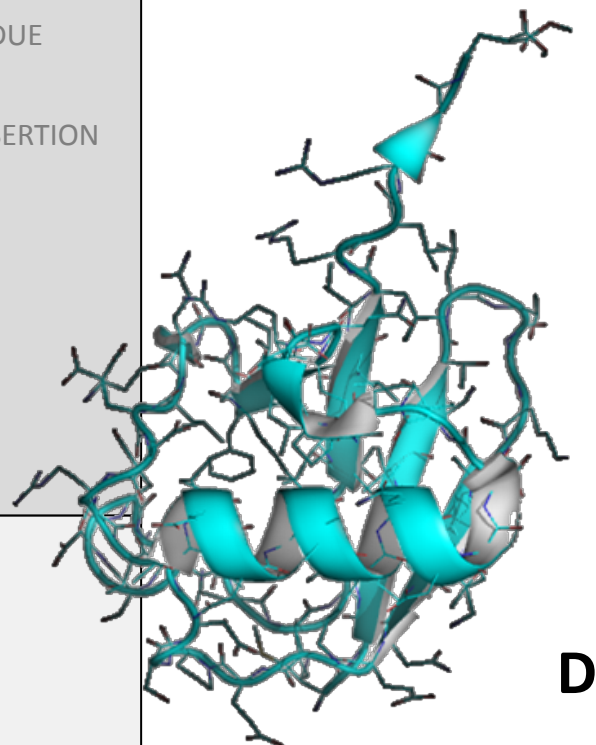
REMARK 470 I=INSERTION CODE):

REMARK 470 M RES CSSEQI ATOMS

REMARK 470 GLU A 158 CG CD OE1 OE2

REMARK 470 ARG C 136 CG CD NE CZ NH1 NH2

**REMARK 470 GLU D 24 CG CD OE1 OE2**



**D**

# Good solution: Clean up the structure with PDBFixer

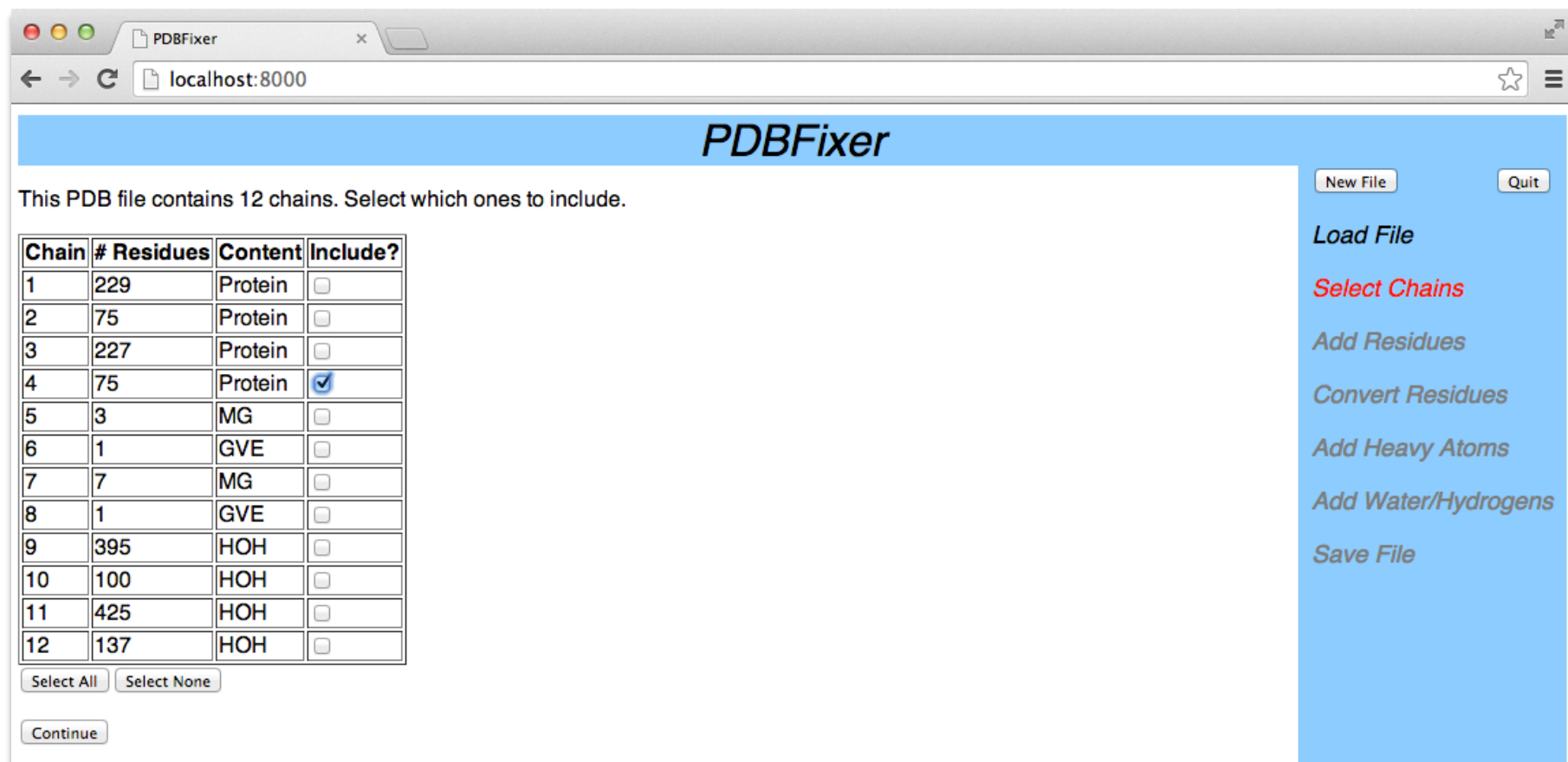


**Get it at:**

<https://github.com/peastman/pdbfixer/releases>



# PDBFixer: Select chains



The screenshot shows a web browser window with the address bar at localhost:8000. The page title is "PDBFixer". Below the title, a message states: "This PDB file contains 12 chains. Select which ones to include." A table with 4 columns (Chain, # Residues, Content, Include?) lists 12 chains. Chain 4 is selected. Below the table are buttons for "Select All", "Select None", and "Continue". On the right, a sidebar contains buttons for "New File", "Quit", "Load File", "Select Chains", "Add Residues", "Convert Residues", "Add Heavy Atoms", "Add Water/Hydrogens", and "Save File".

Chain	# Residues	Content	Include?
1	229	Protein	<input type="checkbox"/>
2	75	Protein	<input type="checkbox"/>
3	227	Protein	<input type="checkbox"/>
4	75	Protein	<input checked="" type="checkbox"/>
5	3	MG	<input type="checkbox"/>
6	1	GVE	<input type="checkbox"/>
7	7	MG	<input type="checkbox"/>
8	1	GVE	<input type="checkbox"/>
9	395	HOH	<input type="checkbox"/>
10	100	HOH	<input type="checkbox"/>
11	425	HOH	<input type="checkbox"/>
12	137	HOH	<input type="checkbox"/>

Select All Select None

Continue

**PDBFixer**

New File Quit

Load File

Select Chains

Add Residues

Convert Residues

Add Heavy Atoms

Add Water/Hydrogens

Save File

# PDBFixer: Add missing atoms



The screenshot shows a web browser window with the address bar set to `localhost:8000`. The page title is *PDBFixer*. The main content area displays a message: "The following residues are missing heavy atoms, which will be added." Below this message is a table with three columns: Chain, Residue, and Missing Atoms. The table contains two rows of data. A "Continue" button is located below the table. On the right side of the page, there is a vertical menu with several options: "New File", "Quit", "Load File", "Select Chains", "Add Residues", "Convert Residues", "Add Heavy Atoms" (highlighted in red), "Add Water/Hydrogens", and "Save File".

*PDBFixer*

The following residues are missing heavy atoms, which will be added.

Chain	Residue	Missing Atoms
1	GLU 24	CG, CD, OE1, OE2
1	GLY 75	OXT

[Continue](#)

- [New File](#)
- [Quit](#)
- [Load File](#)
- [Select Chains](#)
- [Add Residues](#)
- [Convert Residues](#)
- [Add Heavy Atoms](#)
- [Add Water/Hydrogens](#)
- [Save File](#)

# PDBFixer: heterogens, hydrogens, water molecules

PDBFixer

localhost:8000

PDBFixer

New File

Quit

Load File

Select Chains

Add Residues

Convert Residues

Add Heavy Atoms

Add Water/Hydrogens

Save File

## Delete Heterogens

A heterogen is any residue other than a standard amino acid or nucleotide. Do you want to delete heterogens?

Delete all heterogens

## Add Missing Hydrogens

Add missing hydrogen atoms?

☒ Add hydrogens appropriate for pH 

7.0

## Add Water

Add a water box surrounding the model?

☐ Add water

Box dimensions (in nm):

Crystallographic unit cell:    4.611    4.929    6.762

Box containing all atoms:    2.833    3.699    2.926

Ions will be added to neutralize the model. You can optionally add more ions based on a desired bulk ionic strength.

Ionic strength (molar): 

0.0

Positive ion: 

Na+

Negative ion: 

Cl-

Continue

# Hands on: OpenMM implicit solvent exercise

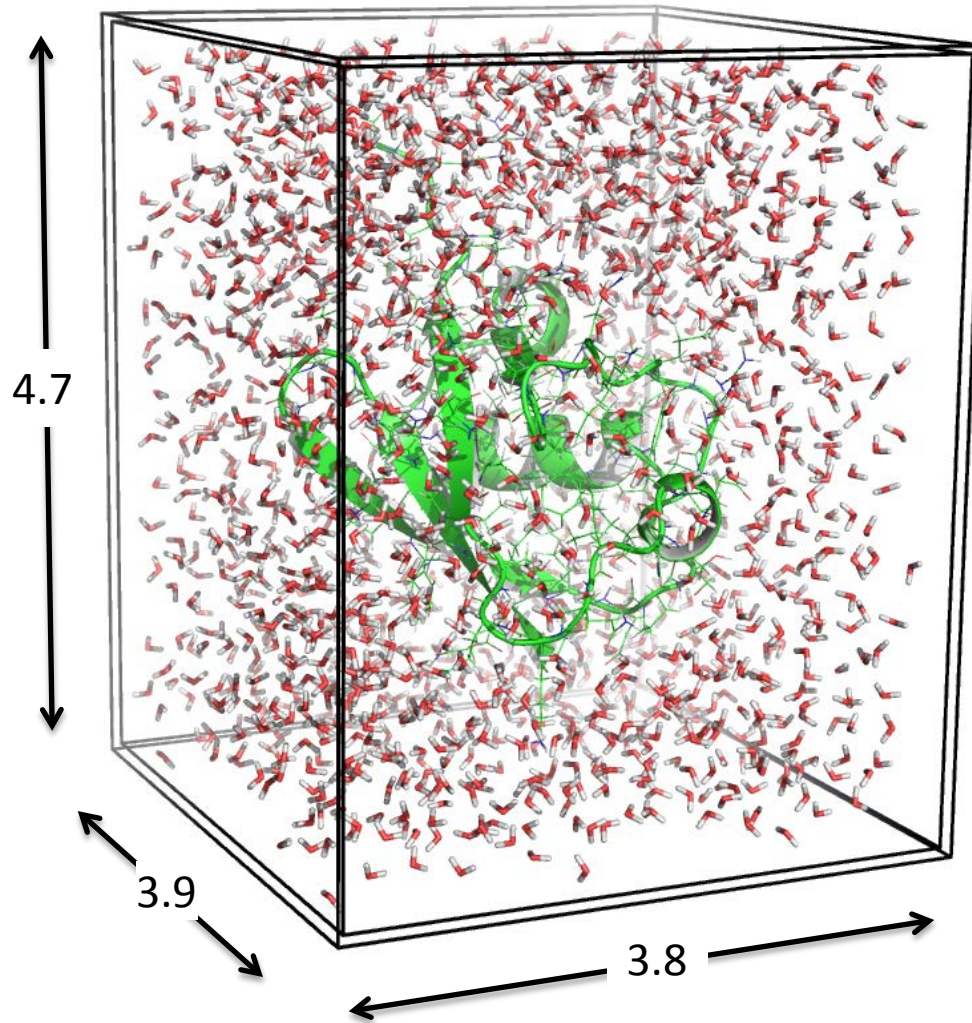
## PDBFixer steps:

- Start PDBFixer: [ `cd /Path-to-pdbfixer` ]; then type [ `python pdbfixer.py` ]
- Load 1xd3 from the PDB with PDBFixer
- Select the 4<sup>th</sup> chain
- Add missing heavy atoms
- Delete all heterogens (any molecules that are not standard amino acids)
- Add hydrogen atoms for pH=7
- Save the .pdb file **without** adding a water box (look in your home directory for the file 'output.pdb')

## OpenMM steps:

- Copy 'output.pdb' from your home directory to the directory of the exercise. Rename to your liking.
- Start iPyNotebook: [ `cd /Path-to-exercise` ]; then type [ `ipython notebook` ]
- Select the implicit solvent exercise from the list.
- In the code section '**Load the protein coordinates**', enter the name of your input .pdb file.
- Choose a name for the output file in the section '**Create a reporter ...**'.
- Click on the first panel and execute each section by pressing: **< shift > + < return >**
- The calculations will take a couple of minutes.
- Open the output file in the visualization program of your choice (i.e. VMD, PyMol, etc.).

Next, let's use PDBFixer to place 1xd3 in a water box



units in nanometer

# Hands on: OpenMM explicit solvent exercise

## PDBFixer steps:

- Start PDBFixer: [ `cd /Path-to-pdbfixer` ]; then type [ `python pdbfixer.py` ]
- If PDBFixer is already running, click on **'New File'**
- Load 1xd3 from the PDB with PDBFixer.
- Select the 4<sup>th</sup> chain.
- Add missing heavy atoms.
- Delete all heterogens (any molecules that are not standard amino acids).
- Add hydrogen atoms for pH=7.
- Select 'Add water' and specify the size of your water box ( ~1nm larger than the box containing all atoms)
- Save the .pdb file **after** adding a water box (look in your home directory for the file 'output.pdb').

## OpenMM steps:

- Copy 'output.pdb' from your home directory to the directory of the exercise. Rename to your liking.
- Start iPyNotebook: [ `cd /Path-to-exercise` ]; then type [ `ipython notebook` ]
- Select the explicit solvent exercise from the list.
- In the code section **'Load the protein coordinates'**, enter the name of your input .pdb file.
- Choose a name for the output file in the section **'Create a reporter ...'**.
- Click on the first panel and execute each section by pressing: **< shift > + < return >**
- The calculations will take a couple of minutes.
- Open the output file in the visualization program of your choice (i.e. VMD, PyMol, etc.).

# Now to some more advanced exercises

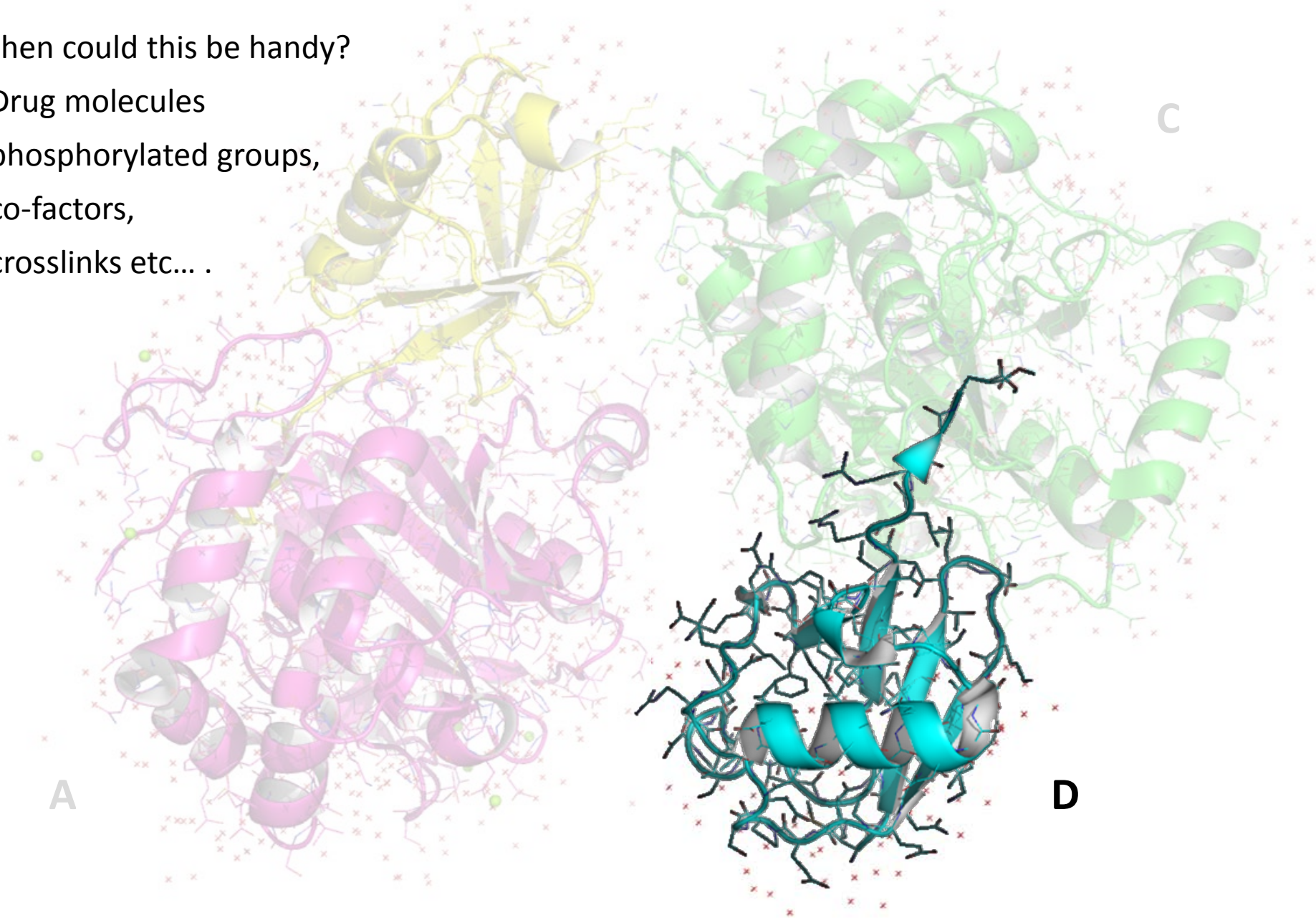
- Use AMBER (or GROMACS) files as an input
- Simulate a hydrogen-bonded system and keep track of the interactions
- Heating your system
- Constraints and restraints



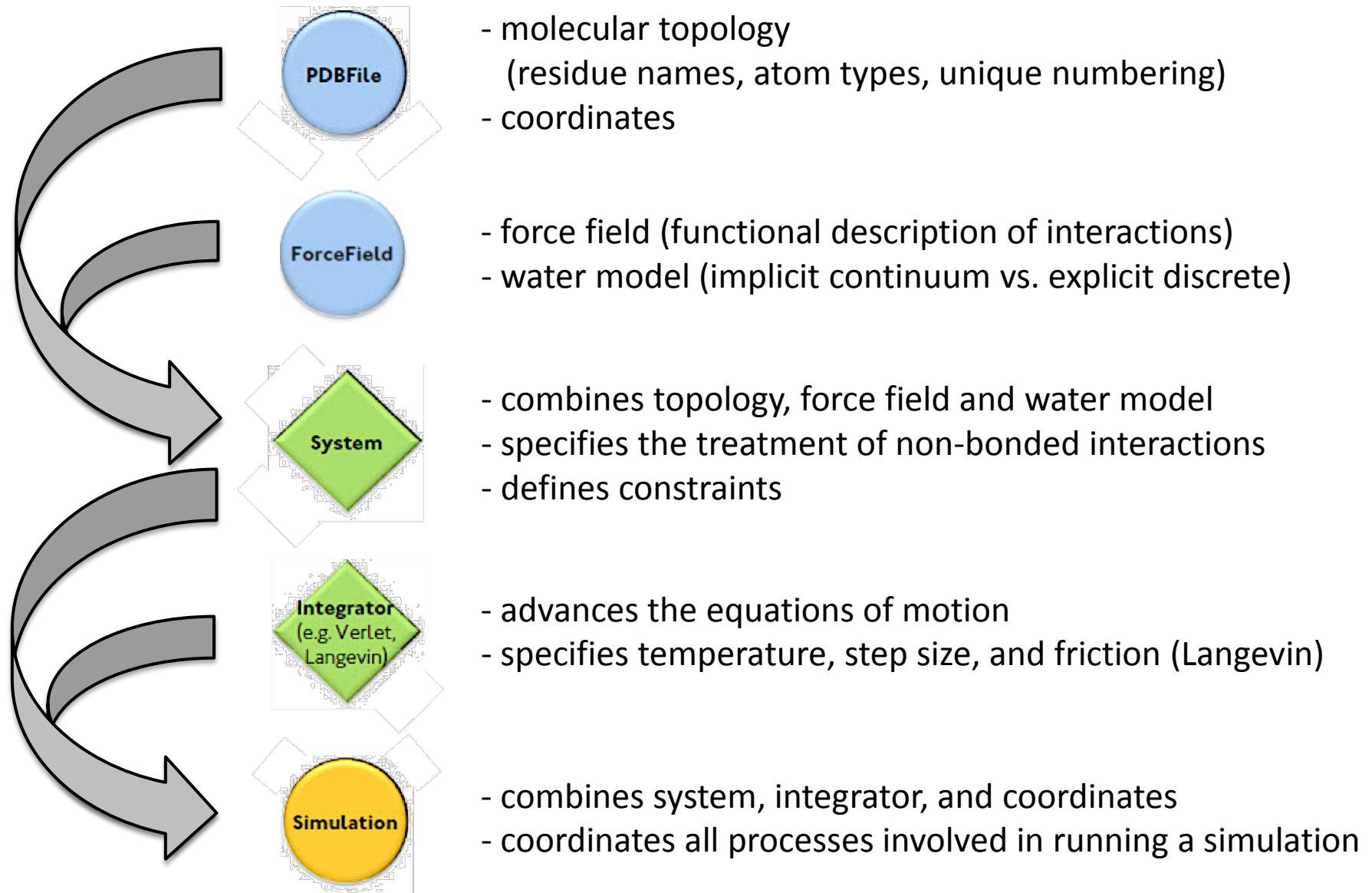
# AMBER or GROMACS input vs. PDB files

When could this be handy?

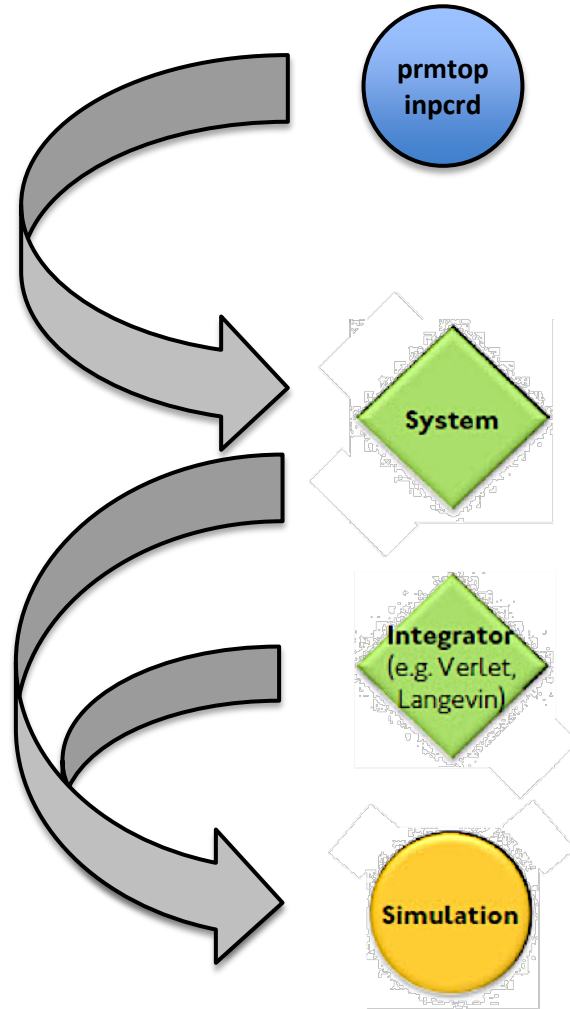
- Drug molecules
- phosphorylated groups,
- co-factors,
- crosslinks etc... .



# Let's quickly revisit the components



# Let's quickly revisit the components



- molecular topology  
(residue names, atom types, unique numbering)
  - coordinates
  - force field (functional description of interactions)
  - water model (implicit continuum vs. explicit discrete)
- 
- combines topology, force field and water model
  - specifies the treatment of non-bonded interactions
  - defines constraints
- 
- advances the equations of motion
  - specifies temperature, step size, and friction (Langevin)
- 
- combines system, integrator, and coordinates
  - coordinates all processes involved in running a simulation



# Simulate a hydrogen-bonded system

iPyNotebook exercise

# Heating

iPyNotebook exercise

# Constraints and Restraints

iPyNotebook exercise



**Now you are ready to play with your own systems**