

# Analyzing *what* you sampled

## active (SampleSet)

### **samples (list of Sample)**

- trajectory (Trajectory)
- ensemble (Ensemble)
- replica (int)

“active”: state of the system after the step

```
sample = storage.steps[10].active[ensemble]
sample = storage.steps[10].active[0]  # replica ID
```

## (Trajectory)

### **(list of Snapshot)**

*Features depend on engine.*

*Common examples:*

- coordinates (numpy.array)
- velocities (numpy.array)
- box\_vectors (numpy.array)

Analyze trajectory by accessing “features”:

*# from a snapshot*

```
trajectory[5].coordinates
```

*# for the whole trajectory*

```
trajectory.coordinates
```

## Other “features”:

- masses
- instantaneous\_temperature
- xyz (unitless coordinates)

# Analyzing *what* you sampled

## active (SampleSet)

### **samples (list of Sample)**

- trajectory (Trajectory)
- ensemble (Ensemble)
- replica (int)

“active”: state of the system after the step

```
sample = storage.steps[10].active[ensemble]
sample = storage.steps[10].active[0]  # replica ID
```

## (Trajectory)

### **(list of Snapshot)**

*Features depend on engine.*

*Common examples:*

- coordinates (numpy.array)
- velocities (numpy.array)
- box\_vectors (numpy.array)

Analyze trajectory by accessing “features”:

```
# from a snapshot
trajectory[5].coordinates
# for the whole trajectory
trajectory.coordinates
```

Or, convert trajectory to MDTraj:

```
from openpathsampling.engines.openmm.tools \
    import trajectory_to_mdtraj

new_traj = trajectory_to_mdtraj(trajectory)
```

## Other “features”:

- masses
- instantaneous\_temperature
- xyz (unitless coordinates)