Order Documentation

Release 0.0.1

Pu Du

Jul 10, 2018

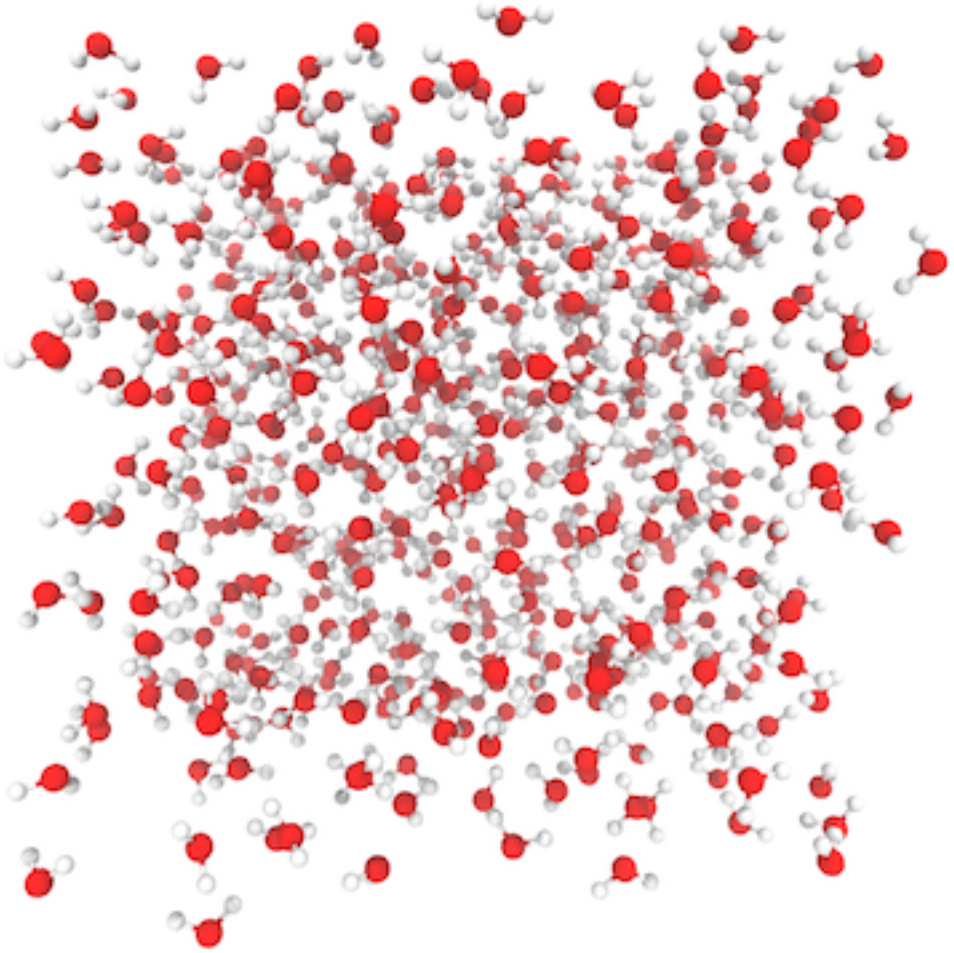
Contents

CHAPTER 1

System

1. Obtain the trajectory from the molecular dynamics production run.

Note: Example system : 550 TIP3P water molecules with box length of 25.5 Å * 25.5 Å * 25.5 Å



2. Generate the **xyz file** (input for the order package) from the trajectory with the atoms (O or center of mass) necessary to calculate the local structure.

Note: This xyz file should include all the water Oxygen atoms and all the other atoms that are cosidered as the closest neighbors. **Make sure to follow the following file format for the xyz file.** In this example the central Oxygen atoms of water are named as "OW".

Line #1 : Number of atoms

Line #2 : BoxlenghtX(Space)BoxlenghtY(Space)BoxlenghtZ

Line #3 & onwards : Symbol assigned by the user for the central atom and the other atoms that can be considered as the closest neighbors

```
550
25.5 25.5 25.5
OW   16.243700000000000       12.386200000000001       4.0693500000000000
OW   21.858300000000000       1.8386800000000001       9.2767199999999992
OW   25.076799999999999       18.696899999999999       25.118400000000001
OW   19.804200000000002       9.5165299999999995       23.707200000000000
OW   13.436999999999999      0.12200600000000000       16.233899999999998
OW   3.4941599999999999       22.202700000000000       5.6509000000000000
OW   25.118600000000001       22.973400000000002       18.201899999999998
OW   8.1970100000000006       15.085699999999999       22.419899999999998
OW   19.969000000000001       3.7206100000000002       10.606999999999999
OW   20.281199999999998       22.139399999999998       12.428599999999999
OW   15.610200000000001       11.665600000000000       24.473099999999999
OW   23.413200000000000       10.349000000000000       10.814600000000000
OW   16.514600000000002       2.6051199999999999       4.7354799999999999
OW   12.232400000000000       24.456700000000001       14.069900000000001
OW   15.659900000000000       3.0214200000000000       24.070699999999999
OW   18.790800000000001       22.306000000000001       1.6701800000000000
OW   12.759200000000000       23.092900000000000       18.335999999999999
OW   7.1571400000000001       17.076100000000000       2.7649300000000001
```

3. Commands available in the package

```
$order [input] [-h] [-t TASK] [-c CENTER] [-b BINS] [-f FREQUENCY] [-p PLOT]

input

File name of the xyz file

-h

Show this help message and exit

-t TASK

Specification of the type of task that you need to perform. TASK can
be oto or tto or avc (default: oto).
If you need to perform multiple tasks simultaneously use a "," to
separate the tasks. (Ex : tto,oto)

-c CENTER

Type of center atom (default: 'O')
(In the example mentioned above the center is 'OW')

-b BINS

Number of bins for the parameter (default: 100)

-f FREQUENCY
```

(continues on next page)

(continued from previous page)

```
Compute the parameter every n frame(s) in the xyz file(default: 1)

-p PLOT

Turn on / off of plotting (default: on)
```

CHAPTER 2

Orientational Tetrahedral Order **(OTO)**

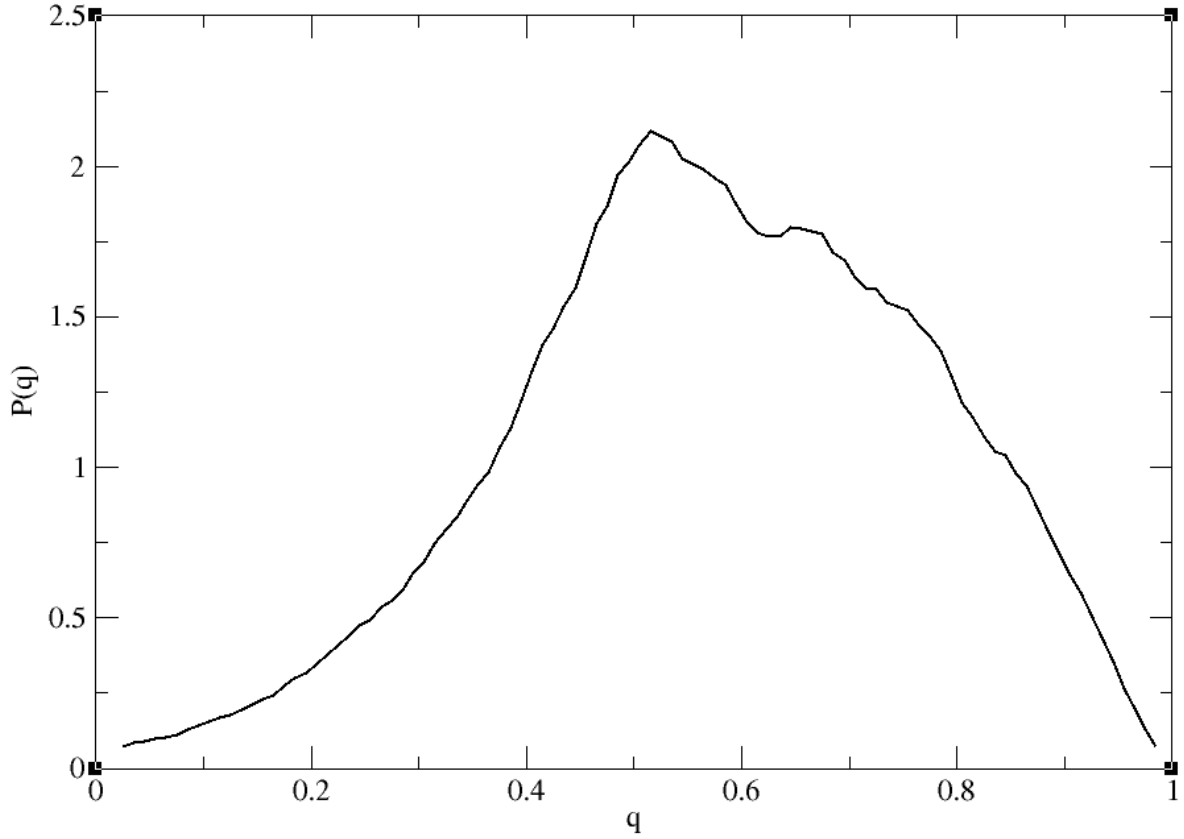
This is the most common type of tetrahedral ordr paramater that is being used. This order parameter uses the four closest water Oxygen neighbors for the calculation. The value of q can rang from 0 to 1, where 0 is for an ideal gas and 1 is for a regular tetrahedron.

$$q = 1 - \frac{3}{8}\sum_{j=1}^{3}\sum_{k=j+1}^{4}\left(\cos\psi_{jk} + \frac{1}{3}\right)^2$$

q = Orientational tetrahedral order parameter

ψ_{jk} = Angle formed by the Oxygen atom under consideration & the two nearest neighbor atoms *j* & *k*

```
$order test.xyz -t oto -c 'OW' -f 5
```


2.5
2
1.5
1
0.5
0
P(q)
0
0.2
0.4
0.6
0.8
1
q

CHAPTER 3

Translational Tetrahedral Order **(TTO)**

Similar to orientational tetrahedral order. But here, the variance of the radial distnace between the central water Oxygen atom and the four nearest neighbors are calculated. THe value of $S_k$is close to 1 and quals 1 for the perfect tetrahedron. As the local tetrahedral nature increases, $S_k$becomes more closer to 1.

$$S_k = 1 - \frac{1}{3}\sum_{k=1}^{4}\frac{(r_k - \bar{r})^2}{4\bar{r}^2}$$

S_k = Translational tetrahedral order parameter

r_k = Radial distance from the cental Oxygen atom to the *k* th peripheral closest neighbor

$\bar{r}$ = Arithmatic mean of the four radial distances

```
$order test.xyz -t tto -c 'OW' -f 5
```

You can get the average $S_k$value for your system by getting the average value of the **raw_data** output file.

Average value for $S_k$for the example is 0.998892128

CHAPTER 4

Asphericity of the Voronoi Cell **(AVC)**

Asphericity parameter (η) can be used to characterize the shape of the Voronoi polyhedron. This value is independent of the size of the polyhedron. The value of η for a perfect sphere the is 1, for ice is 2.25 and for a regular tetrahedron it is 3.31. *[Duboué-Dijon2015]*

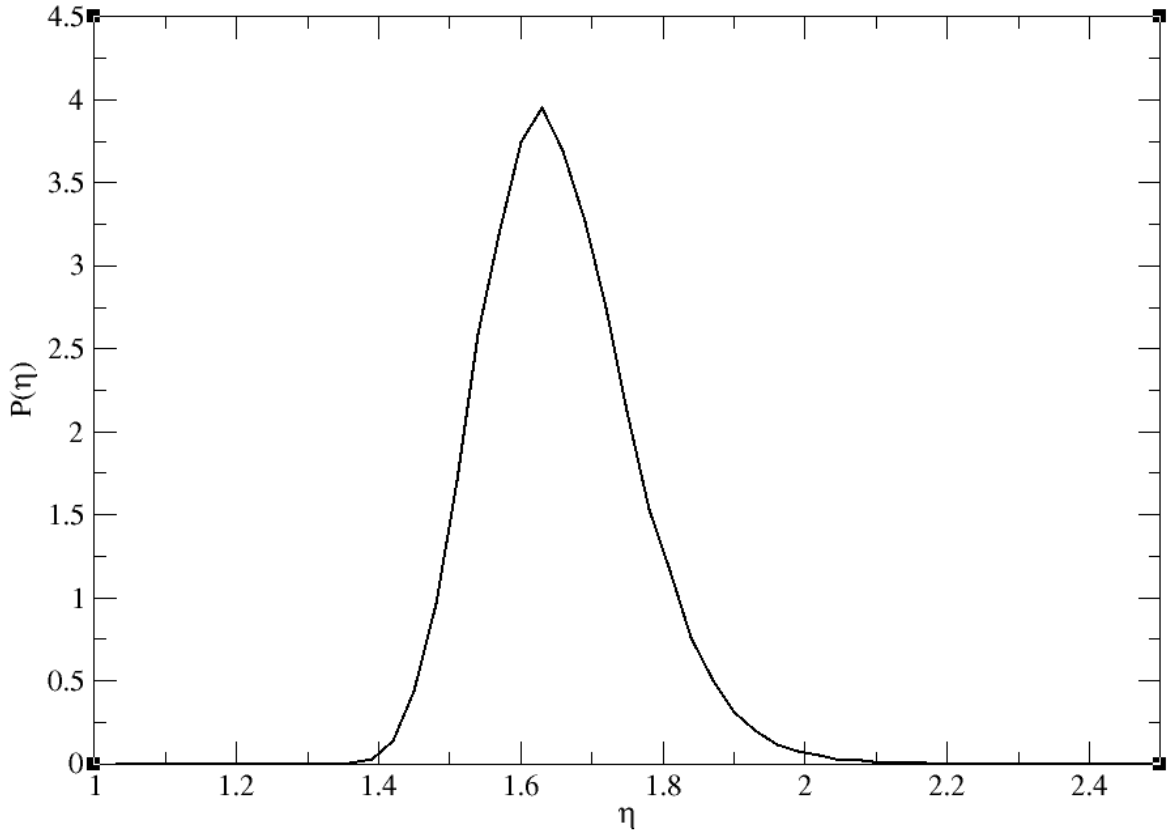
$$\eta = \frac{A^3}{36\pi V^2}$$

η = Asphericity parameter

A = Area of the polyhedron

V = Volume of the polyhedron

```
$order test.xyz -t avc -c 'OW' -f 5
```


4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
P(η)
1
1.2
1.4
1.6
1.8
2
2.2
2.4
η

CHAPTER 5

API Reference

5.1 order package

5.1.1 Submodules

5.1.2 order.XYZ module

XYZ trajectory reader

class order.XYZ.**XYZLoader**(*filename*)

Bases: object

close()
close XYZ file if it was open

read_all_frames()
read all frames of XYZ trajectory

read_n_frames()
get the starting position of each frame

read_next_frame(*frame*)
read a frame of XYZ trajectory

5.1.3 order.avc module

class order.avc.**VoronoiCell**(*filename*, *center*, *bins=100*)

Bases: *order.oto.Orientational*

asphericity of the Voronoi cell

asphericity(*freq=1*)
compute asphericity of the Voronoi cell

compute_vc(*points*)
compute the Voronoi cell

polyhedron(*coords*, *j*, *L*)
find the polyhedron for center molecule

wrap_box(*c_coord*, *coords*, *L*)
wrap the simulation box

5.1.4 order.lsi module

5.1.5 order.order module

5.1.6 order.oto module

Orientational Tetrahedral Order q

class order.oto.**Orientational**(*trajectory*, *center*, *bins=100*)
Bases: object

orientational tetrahedral order parameter

four_neighbors(*coords*, *L*)
compute four nearest water oxygen neighbors

orientational_param(*freq=1*)
compute orientational order parameter

out_put(*taskname='oto'*, *param_name='Q'*)
output raw data and distribution

5.1.7 order.plot module

class order.plot.**plot**(*filename*, *taskname*)
Bases: object

plotting

plot_distribution()
plot distribution

plot_ionic()
plot distribution

5.1.8 order.tto module

Translational Tetrahedral Order Sk

class order.tto.**Translational**(*filename*, *center*, *bins=100*)
Bases: *order.oto.Orientational*

translational tetrahedral order parameter

translational_param(*freq=1*)
compute translational order parameter

5.1.9 order.util module

order.util.**cos_angle**(*v1*, *v2*)
compute the cos angle of two giving vectors

order.util.**output_end**(*t_start*, *t_end*)
print total running time

order.util.**output_interface_info**(*input_info*)
print interface task information

order.util.**output_ionic_info**(*input_info*)
print ionic conductivity task information

order.util.**output_system_info**(*filename*, *n_atoms*, *n_frames*)
print system information

order.util.**output_task**(*name*, *freq*, *bins*, *center*)
print task information

order.util.**output_welcome**()
print welcome information

order.util.**pbc**(*dx*, *dy*, *dz*, *L*)
periodic boundary conditions

5.1.10 Module contents

CHAPTER 6

Order

6.1 Getting started

6.1.1 Installation:

```
$ pip install iorder
```

or

```
$ git clone https://github.com/ipudu/order
$ cd order
$ python setup.py install
```

6.1.2 Running:

```
# calculate bar parmeter for center atom of O
$ order foo.xyz -t bar -c 'O' -f 100 -b 100
```

6.2 Feature Support

Order is ready to calculate several geometric order parameters:

- Orientational Tetrahedral Order
- Translational Tetrahedral Order
- Asphericity of the Voronoi Cell

- Ionic Conductivity
- and more

6.3 Contributing

Contributions to this library are always welcome and highly appreciated.

6.4 License

MIT - See the LICENSE for more information.

Bibliography

[Duboué-Dijon2015] DOI: 10.1021/acs.jpcb.5b02936

Python Module Index

o

Index