Adapt your dataset for visualization

Created	@May 24, 2022 5:15 PM
Tags	

The program uses an output file from molecular dynamics simulation software as input. The output file needs to be correctly converted for the program to process without error.

There can be many implementations to allow Paraview to process data. In this program, we convert the MD output files into dataframe and process them. Then vtk file is generated from the dataframe for visualization in Paraview.

Dataframe must contain a header about location of the center of an ellipsoid (rx, ry, rz) and a vector expressing its long axis (nx, ny, nz)

header = [”rx”, “ry”, “rz”, “nx”, “ny” ,“nz”]

dataframe

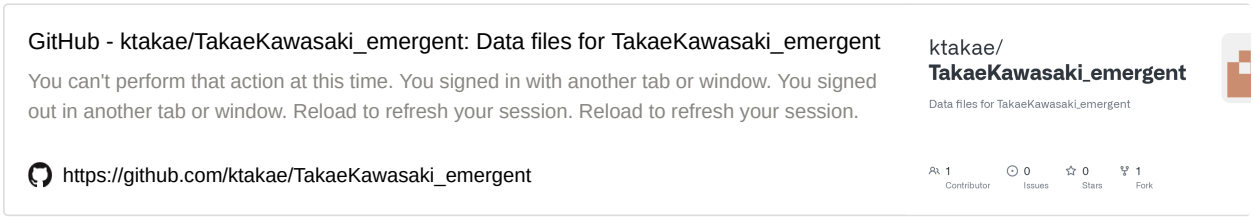
Based on the given data from MD output files, you can add more features into the dataframe.

Some examples include

- size and shape of an ellipsoid
- color of an ellipsoid
- extracting certain ellipsoids for visualization

Examples

1. examples of MD output:



Code

```python
def dat2df(file):

    # extract header for dataframe
    with open(file) as f:
        for i, line in enumerate(f):
            if i == 2: # 3rd line
                line = line.replace("#", "")
                line = line.replace("-", " ")
                header = line
            elif i > 3:
                break

    # read as dataframe
    df = pd.read_csv(file, skiprows=3, sep=" ", names = header.split())
    return df
```

dataframe.header() output

	plabel	pindex	rx	ry	rz	vx	vy	...	nz	dnx	dny	dnz	phix	phiy	phiz
0	1	3912	0.826915	0.978840	1.013283	0.388745	0.173847	...	-0.969527	0.969210	-1.565884	0.107548	2.442000	-4.152821	1.379263
1	1	9	1.352162	1.944742	1.007975	-0.515264	0.149467	...	-0.888020	-0.050139	-0.606095	-0.039387	1.890236	-2.497771	0.838141
2	1	72	1.966489	1.028953	0.992591	-0.235092	-0.140929	...	0.971935	-0.813748	0.526727	-0.228538	-1.723518	0.378447	-0.608038
3	1	73	2.479360	2.022747	1.009526	-0.401807	-0.009036	...	0.911767	-0.669125	-0.411702	-0.204359	-7.120127	-4.849189	-5.397039
4	1	136	0.770134	2.906395	0.997990	0.246304	0.390278	...	-0.205176	0.172403	-0.603892	0.377525	-53.989300	55.588884	-37.718142

2. example from Kawasaki-san 15/08/22

```
22.8692 26.8244 4.85187 0.569599 0.685136 0.454033
22.0803 3.49716 5.20238 0.470743 0.497648 0.728524
22.9539 2.84143 5.34281 0.563813 0.516151 0.64475
```

Code

```python
def dat2df(file):


    header = ['rx', 'ry', 'rz', 'nx', 'ny', 'nz']

    # read as dataframe
    df = pd.read_csv(file, sep=" ", names = header)
    return df
```

dataframe.header() output

rx	ry	rz	nx	ny	nz
22.8692	26.82440	4.85187	0.569599	0.685136	0.454033
22.0803	3.49716	5.20238	0.470743	0.497648	0.728524
22.9539	2.84143	5.34281	0.563813	0.516151	0.644750
24.7547	3.80540	3.21637	0.534581	0.363120	0.763130
25.1212	1.89007	3.44674	0.803090	0.145722	0.577765