# Tutorial 2: Calculating Principal Transition Diagrams

Subhamoy Mahajan

April 6, 2021

This tutorial demonstrates the use of **NPanalysis v1.2** module for the calculation of some additional analysis performed in supporting information of Mahajan and Tang [1]. Namely, total number of PEI role conversions, moment of inertia of a nanoparticle, shape descriptors of a nanoparticle, and evaluation of minimum distance between DNA and PEI by only considering the charged beads. The files `connected.pickle`, `cluster.pickle`, `constants.pickle`, `molndx.pickle`, and `NPwhole.xtc` from Tutorial 1 is provided.

To make the `NPwhole.xtc` file readable it is separated into several `.gro` files using the following commands,

```
Tut1$ mkdir NPwhole
Tut1$ echo "0" | gmx trjconv -f NPwhole.xtc -s md_1.tpr -o NPwhole/
    DP.gro -sep
```

The aforementioned properties can then be calculated by running `python calc.py`. The detailed explanation of each step is provided in the following sections.

## 1. PEI role conversion

PEI can perform three roles (i) unbound (u), (ii) peripheral (p), or (iii) bridging (b). By checking the role of each PEI in two consecutive timesteps, their change (or lack thereof) in role can be determined. The total number of role conversions is calculated using the command,

```
NPa.connMat.get_role_conversion(connected_pickle='connected.pickle', mol=1, \
    outname='role_conv.dat', sep=',')
```

Here, the connection matrix (`connected.pickle`)determined in Tutorial 1 is used to calculate the role conversions. The output is stored in `role_conv.dat` and the data is separated by ','. `mol=1` is used to count the role conversion of PEIs.

**role_conv.dat**

```
# Bridging molecule is PEI
#u->p   p->b    u->b
576,444,6
#p->u   b->p    b->u
413,398,4
```

## 2. Moment of Inertia

The moment of inertia of nanoparticle is calculated using Eq 1, where $(x_i, y_i, z_i)$ are position coordinate of $i^{th}$ particle in the nanoparticle with respect to its center of mass, $m_i$ is mass of the $i^{th}$ particle, and $N$ is the total number of particles.

$$I = \sum_{i=1}^{N} m_i \begin{bmatrix} (y_i^2 + z_i^2) & -x_i y_i & -x_i z_i \\ -y_i x_i & (x_i^2 + z_i^2) & -y_i z_i \\ -z_i x_i & -z_i y_i & (x_i^2 + y_i^2) \end{bmatrix} \tag{1}$$

This is calculated using the command below for the largest NP at the last timestep.

```
MOI=NPa.radius.calc_MOI(pos, atoms, mass)
```

`pos` contains position of all particles at the last timestep,

```
pos,box,text=NPa.gmx.read_gro('NPwhole/DP250.gro')
```

`atoms` contains the atom IDs of all beads present in the largest NP at the last timestep,

```
clusters=nx.read_gpickle('cluster.pickle')
constants=nx.read_gpickle('constants.pickle')
dname=constants['dna_name']
pname=constants['pei_name']
ndx=nx.read_gpickle('molndx.pickle')
atoms=NPa.gmx.get_NPatomIDs(clusters[-1],ndx,dname,pname, main_mol=0, NP_ID=1)
```

and `mass` contains the mass of all beads in the simulation,

```
mass=nx.read_gpickle('mass.pickle')
```

Finally the moment of inertia (in amu nm$^2$) is printed out as,

```
[[ 3119083.69998314 -1284933.78606724  1239199.99084473]
 [-1284933.78606724  3678059.07061965   878460.37583514]
 [ 1239199.99084473   878460.37583514  3636473.93409995]]
```

## 3. Gyration Tensor and Shape descriptors

The gyration tensor of the NP is calculated using Eq 2, where $(x_i, y_i, z_i)$ are position coordinate of $i^{th}$ particle in the nanoparticle with respect to its geometric center and $N$ is the total number of particles.

$$S = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} x_i^2 & x_i y_i & x_i z_i \\ y_i x_i & z_i^2 & y_i z_i \\ z_i x_i & z_i y_i & z_i^2 \end{bmatrix} \tag{2}$$

Three shape desciptors asphericity ($b$), acylindricity ($c$) and relative shape anisotropy ($\kappa^2$) is calculated using Eq 3-5, where $\lambda_1 < \lambda_2 < \lambda_3$ are eigen values of $S$.[2]

$$b = \lambda_3 - \frac{1}{2}(\lambda_2 + \lambda_1) \tag{3}$$

$$c = \lambda_2 - \lambda_1 \tag{4}$$

$$\kappa^2 = \frac{3}{2}\frac{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}{(\lambda_1 + \lambda_2 + \lambda_3)^2} - \frac{1}{2} \tag{5}$$

For a short summary of shape descriptors, see this wiki. The shape decriptors are evaluated using the following command. The result is shown below.

```
NPa.radius.NP_shape(clusters[-1][1],inGRO='NPwhole/DP250.gro', \
                    ndx_pickle='molndx.pickle')
```

```
asphericity: 23.607
acylindricity: 1.4791
relative shape anisotropy: 0.4453
```

## 4. Minimum distance using only charged beads

To calculate the minimum distance using only charged beads, first an index file (pickled) is created only for charged beads using the following command, where `Qndx.pickle` is the output pickled file and the prefix `'Q'` is added to DNA and PEI names saved in the `constants.pickle` file.

```
NPa.gmx.gen_index_charge('Qndx.pickle',prefix='Q')
```

The index file can be saved in a readable format using the following command, where `Qndx.ndx` is the output index file in Gromacs format. This command is optional and is not necessary for calculating the minimum distance.

```
NPa.gmx.write_index_mol('Qndx.ndx',ndx_pickle='Qndx.pickle',prefix='Q')
```

Finally the minimum distances for each DNA and PEI pair can be calculated using `gro2connected()` function like Tutorial 1.

```
NPa.connMat.gro2connected(inGRO='NPwhole/DP',connected_pickle='Qconn.pickle',
    ndx_pickle='Qndx.pickle', time_shift=0, time_fac=4E-6, prefix='Q',
    time_pickle='time.pickle', mindist_pickle='Qmind.pickle')
```

The minimum distance for DNA5 and PEI58 is printed out as,

```
[6.4092828  6.82187753 7.61701707 7.47597278 8.57964481 6.40781179
 7.59644726 8.50762828 8.86821966 8.36644972 8.62780743 8.29687676
 7.11541102 7.58601173 7.07632454 6.66340311 6.48184063 6.35721299
 5.19205836 5.01282485 5.27732764 5.34715083 5.63868407 4.72269521
 4.09918175 2.64368947 3.34924842 2.88802718 2.28130314 1.67515313
 1.39295406 0.79239195 0.52044884 0.51480579 0.50042082 0.49016732
 0.49323625 0.47298309 0.47800732 0.47341948 0.46195346 0.54124024
 0.52806439 0.47662669 0.51508931 0.47737302 0.50591007 0.52177581
...
```

## References


[1] S. Mahajan and T. Tang, Polyethylenimine–DNA Ratio Strongly Affects Their Nanoparticle Formation: A Large-Scale Coarse-Grained Molecular Dynamics Study, *J. Phys. B* **2019**, 123, 45, 9629-9640.

D. N. Theodorou and U. W. Suter, Shape of Unperturbed Linear Polymers: Polypropylene, *Macromolecules* **1985**, 18, 1206-1214