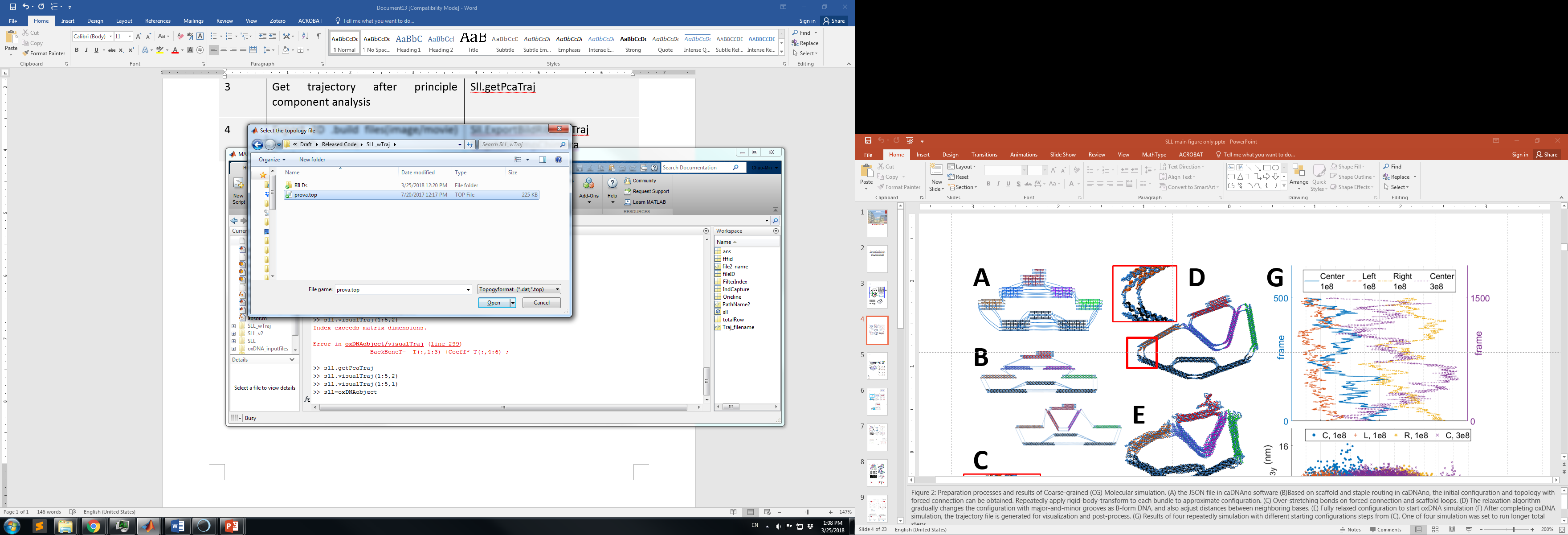
Tutorial of Object-oriented programming (OOP) class oxDNAobject to visualize oxDNA configuration or trajectory

|  |  |  |
| --- | --- | --- |
| **Steps** | **Goal** | **Functions** |
| 1 | Create the object | sll=oxDNAobject |
| 2 | Visualize configuration/trajectory in MATLAB | Sll.visualConf  Sll.visualTraj |
| 3 | Get trajectory after principle component analysis | Sll.visualConfAndAssignRef  Sll.getPcaTraj |
| 4 | Export 3D .build files(image/movie) for UCSD chimera | Sll.ExportBildRibbonTraj  Sll.printCompChimera |

1. **Create the object**

This step requires three input files in this order to gather information:

>> sll=oxDNAobject



1. oxDNA topology file
2. oxDNA configuration/trajectory file

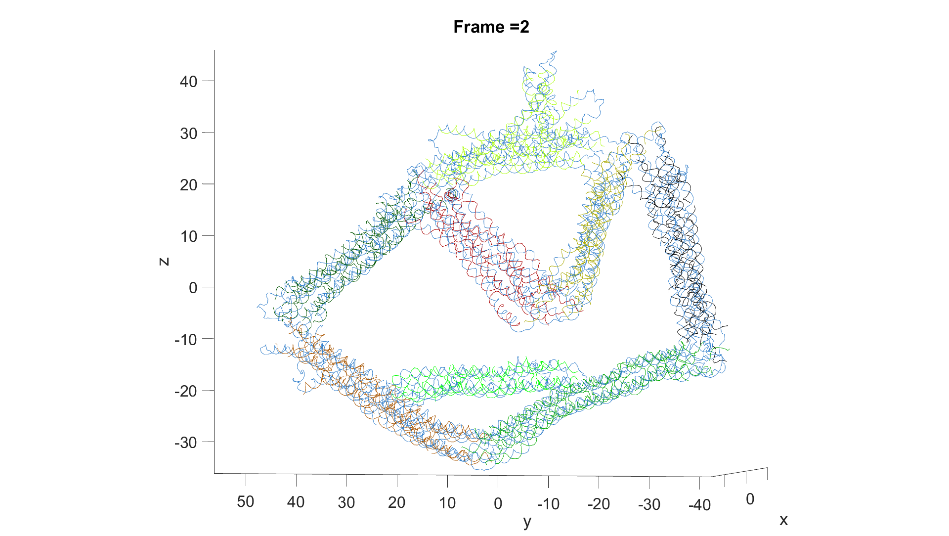
**Note**: Dependent on the number of configurations in a trajectory, the searching line by line may take a while. To make it easier, once a trajectory had been loaded, it may save as .mat file for loading faster later.

1. cadnano json file

This file includes the color codes for each bundle. To change the colors, it requires not only changing colors in cadnano but also executing RBT\_Conf.m to update the presaved variable, BM, in the folder.

1. **Visualize configuration/trajectory in MATLAB**
2. Sll.visualConf(*frame\_index*)

>> sll.visualConf(2)



1. Sll.visualTraj(*frame\_indexs* , *mode*)

*frame\_indexs:* indexes of interested frames, array.

*mode:* if ==1 trajectory with shifting to center

else trajectory after PCA’s orientation

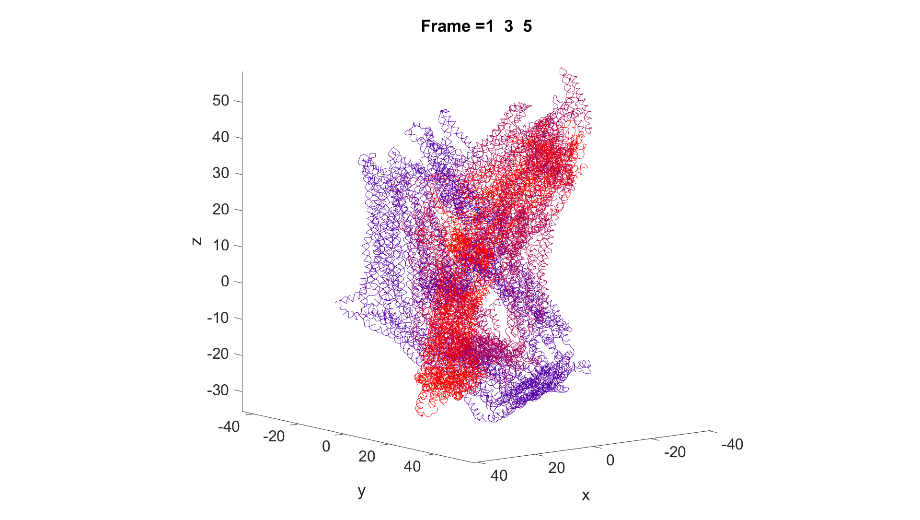
Note:

1. To get the trajectory with similar orientation, *getPcaTraj* need to run earlier.
2. Colors from blue to red show the configurations order. i.e. The blue configuration is the first one.

.

. 1>> sll.visualTraj(1:2:5,1)

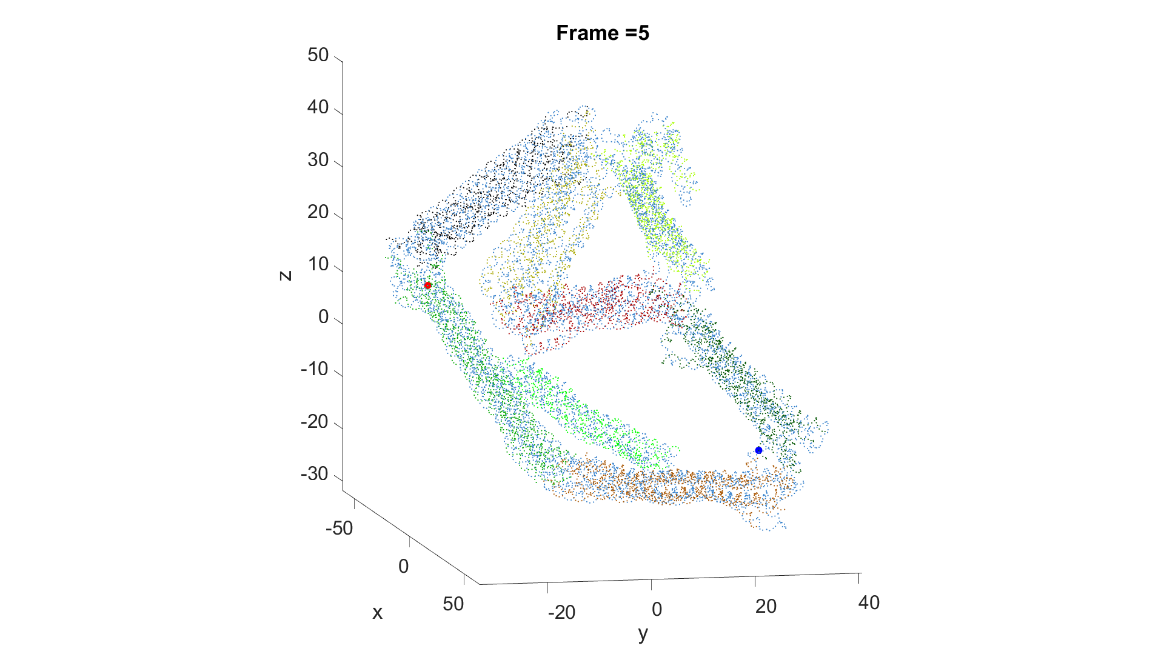
. 2>>sll.visualTraj(1:2:5,2)

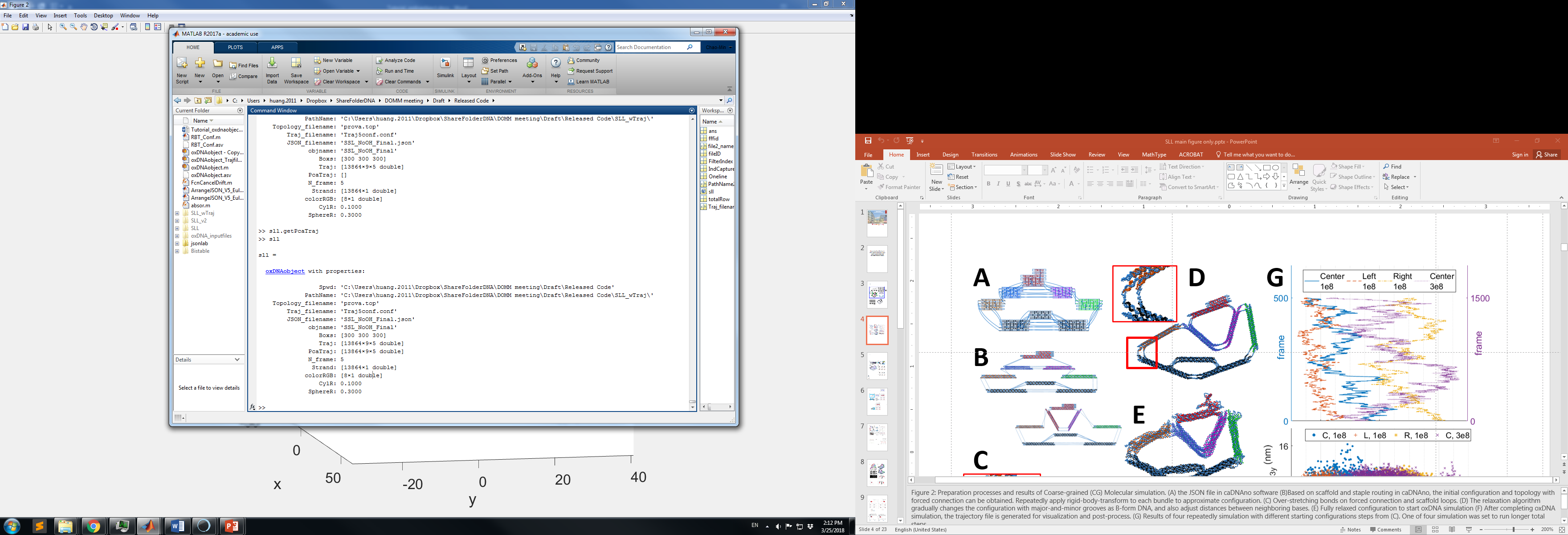


1. **get trajectory after principle component analysis**

Sll.visualConfAndAssignRef(*frame\_index*) & Sll.getPcaTraj

During simulations, a structure may float around the simulation box if without any constrain force. However, we are interested in the configuration changes, which means the structural fluctuation in local coordinate. In addition, the experiment usually relies on the transmission electron microscopy(TEM) whose results are planar images. Here we assume the third eigenvector of the principle component analysis(PCA) as the projection direction(Z direction). To determine a local coordinate, one still need to assign another vector to build the local coordinate. Use the right and left clicks of the mouse to assign two points (blue and red) as reference to obtain the vector. The projection of this vector will be considered as the X direction.

. 3 >> sll.visualConfAndAssignRef(5)

. 4 

. 5>> sll.getPcaTraj

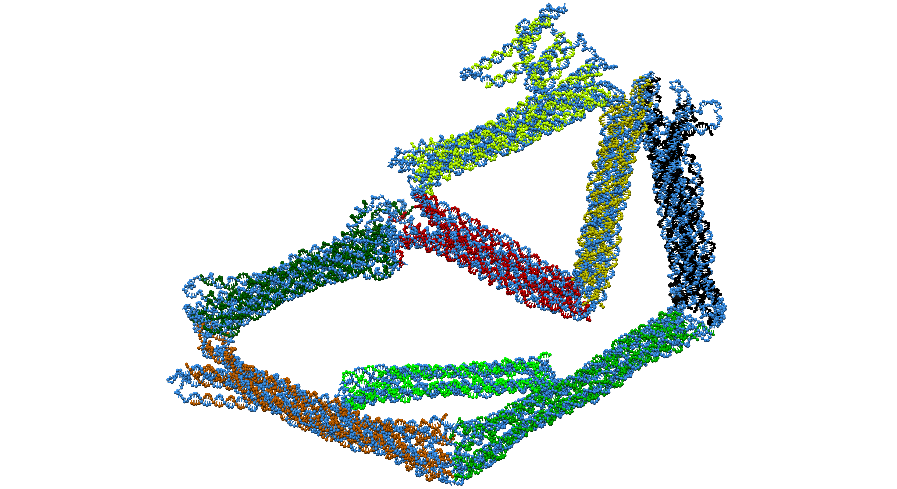
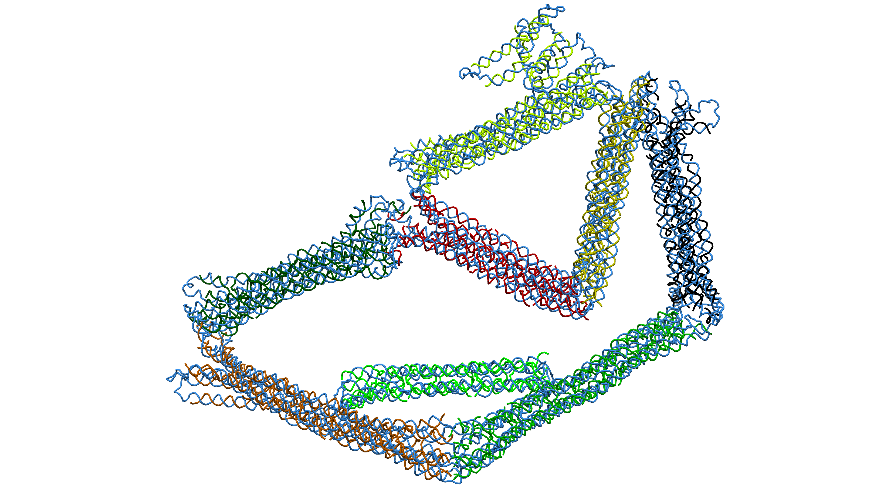
1. Export 3D .build files(image/movie) for UCSD chimera
2. Sll.ExportBildRibbonTraj(*mode*)

*mode*: if ==1 export trajectory as ribbon model

else export trajectory as CG model



. 7CG model



. 6 Ribbon model

1. Sll.printCompChimera(*mode*)

*mode*: if ==1 export command script for ribbon model

else export command script for CG model

Now we should have all 3D BILD files for all configurations no matter in CG or ribbon models in the “BILD” subfolder. To make a video, we strongly suggest to use the trajectory after PCA otherwise the video may fluctuate in a fixed view. The function “printCompChimera” generate a command script for Chimera as the following figure. **Make sure the path without any space.** This script will automatically and repeatedly load those 3D BILD files (from 4a), save as png files, and rotate to different views. The resolution of each image can be assigned in MATLAB or in this script.

