Tutorial 2: First *in-silico* microscopy video with different exposure time

Subhamoy Mahajan

27 Dec, 2020

The PSF generated in tutorial 1 is used as the PSF.

1. Extract the structure files.

```
term$ tar -xzf Struct.tar.gz
```

This will extract structure files “dp10.gro” to “dp110.gro”. The number after “dp” refers to the timestep of the molecular dynamics simulation.

2. Generate monochrome image data files.

```
term$ bash gen_imgdat.sh
```

gen_imgdat.sh:

```
for i in {10..110}
do
    ../../gen_mono -p parameters.dat -f dp${i}.gro -o img${i}
done
```

This generates image data file for timesteps 10 to 110. For each timestep two image data files associated with the wavelength 670 nm and 518 nm is generated. The files created are “img10_lam518_fs800.dat” to “img110_lam518_fs800.dat”, and “img10_lam670_fs800.dat” to “img110_lam670_fs800.dat”.

3. Generate colored *in-silico* microsocpy image.

```
term$ bash gen_color.sh
```

gen_color.sh:

```
for i in {10..100..10}
do
    python ../../mono2color.py -f img -p png_param.dat -t ${i}
done

for i in {10..100..10}
do
    python ../../mono2color.py -f img -p png_param_T10.dat -t ${i}
done
```

The first for loop generates colored microscopy images for every 10^{th} timestep with $T = 1$ (no time averaging). Similarly, the second loop generates colored microscopy images for every 10^{th} timestep with $T = 10$. Using $T = 10$ implies image data from 10 consecutive timesteps are used to create an averaged color image.

This creates PNG image files "img10_fs800_T1_I_0.13_0.25.png", "img20_fs800_T1_I_0.13_0.25.png", $\cdots$, "img110_fs800_T1_I_0.13_0.25.png", "img10_fs800_T10_I_0.13_0.25.png", "img20_fs800_T10_I_0.13_0.25.png", $\cdots$, "img110_fs800_T10_I_0.13_0.25.png".

4. Generate *in-silico* microscopy video

```
term$ python ../../create_vid.py -f img -p png_param.dat -t0 10 -tmax 100
      -tdiff 10 -fps 5
term$ python ../../create_vid.py -f img -p png_param_T10.dat -t0 10
      -tmax 100 -tdiff 10 -fps 5
```

Videos are generated starting from colored microscopy images using images from timestep 10, 20, $\cdots$, 100. The image is generate with 5 fps. Therefore, the entire video will be 2 seconds long.

This creates two video files "img_fs800_T10_I_0.13_0.25.avi" and "img_fs800_T1_I_0.13_0.25.avi".

5. Generate monochrome *in-silico* microscopy images.

```
term$ bash gen_mono.sh
```

gen_mono.sh:

```
for i in {10..100..10}
do
    python ../../render_mono.py -f img -p png_param.dat -t ${i}
done

for i in {10..100..10}
do
    python ../../render_mono.py -f img -p png_param_T10.dat -t ${i}
done
```

The first for loop generates monochrome microscopy images for every 10^{th} timestep with $T = 1$ (no time averaging). Similarly, the second loop generates monochrome microscopy images for every 10^{th} timestep with $T = 10$. Using $T = 10$ implies image data from 10 consecutive timesteps are used to create an averaged monochrome image.

This creates PNG image files:

- "mono_img10_lam670_fs800_T1_I0.13.png", "mono_img20_lam670_fs800_T1_I0.13.png", $\cdots$, "mono_img100_lam670_fs800_T1_I0.13.png"
- "mono_img10_lam518_fs800_T1_I0.25.png", "mono_img20_lam518_fs800_T1_I0.25.png", $\cdots$, "mono_img100_lam518_fs800_T1_I0.25.png"
- "mono_img10_lam670_fs800_T10_I0.13.png", "mono_img20_lam670_fs800_T10_I0.13.png", $\cdots$, "mono_img100_lam670_fs800_T10_I0.13.png"
- "mono_img10_lam518_fs800_T10_I0.25.png", "mono_img20_lam518_fs800_T10_I0.25.png", $\cdots$, "mono_img100_lam518_fs800_T10_I0.25.png".

6. Generate monochrome *in-silico* microscopy videos.

```
term$ python ../../create_vid.py -f img -p png_param.dat -t0 10 -tmax 100
      -tdiff 10 -fps 5 -type mono
term$ python ../../create_vid.py -f img -p png_param_T10.dat -t0 10
      -tmax 100 -tdiff 10 -fps 5 -type mono
```

This creates four video files "mono_img_lam518_fs800_T1_I0.25.avi", "mono_img_lam670_fs800_T1_I0.13.avi", "mono_img_lam518_fs800_T10_I0.25.avi" and "mono_img_lam670_fs800_T10_I0.13.avi". Each video is 2 seconds long similar to the colored microscopy videos.