**Frequency Multiplexed DNA-Paint Demodulation Software Guide**

This code is related to the publication:

"Excitation-multiplexed multicolor superresolution imaging with fm-STORM and fm-DNA-PAINT", Pablo A. Gómez-García, Erik T. Garbacik, Jason J. Otterstrom, Maria F. Garcia-Parajo, Melike Lakadamyali, PNAS, 2018.

The workflow explains how to use the software. In our case, we used DAX images from an Andor EM-CCD Camera and Insight3 (from Bo Huang) for the localization of the single molecules. The software could be adapted to accept other file types for the images and localization lists.

The DAX files should contain an info file (.inf) with the same name, where the number of frames, the number of lasers used, the laser modulation frequencies, the pixel size, the camera frame rate and exposure time, and the image size are specified. An example of this file is also attached.

**1. Parameters definition:**

In the main.py file introduce the working directory that should be the with the top-level directory of the data sets. If there are several folders with several files contained, the code would be executed in batch through all the DAX files. The directory must be a string with a “/” at the end, like the example below:

dir = ‘W:/fm-DNA-PAINT/Cell1/’

Provide a value to the “chunk\_size” variable that is the maximum number of frames the software will take to process. If it is smaller than the total number of frames of the image, the code will make groups and process them sequentially.

In Analysis.py provide a value to the variable “centers” by modifying the line 43 of the code. This variable specifies the frequency bins that you want to demodulate (see Figure below for the available frequency bins depending on your experimental conditions). “centers” is a vector with length equal to the number of channels to demodulate (the number of channels should be equal or smaller than the number of available frequency bins).

**C:\Users\pgomez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure 2 (V2).tif**

The rest of the variables, like F, fi, image size, pixel size, etc, will be automatically obtained from the DAX info file. The frame window size (m) will be calculated depending of the number of lasers used in the experiment. By default:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nº of lasers** | 1 | 2 | 3 | 4 | 5 |
| **m** | 6 | 6 | 6 | 8 | 10 |

Those default values can be modified in lines 9-31 of main.py.

In order to illustrate how to choose the appropriate values for the “centers” variable, see the examples below:

1. F=60Hz, 2-lasers at f1=30Hz and f2=20Hz and frame window size m=6:

The output from DFT will contain values corresponding to the following frequencies:

[0 10 20 -30 -20 -10] Hz.

The “centers” corresponding to 30Hz and 20Hz, respectively would be:

[3, 2] (it is the order that the frequencies have in the frequency bins vector, in this case 2nd and 3rd). (Note that python starts counting at 0).

1. F=90Hz, 3-lasers with f1=45Hz, f2=30Hz and f3=15Hz, and frame window size m=6:

The output from DFT will contain values corresponding to the following frequencies:

[0 15 30 -45 -30 -15] Hz.

The “centers” corresponding to 45Hz, 30Hz and 15Hz, respectively would be:

[3, 2, 1].

1. F=60Hz, and 2-lasers with f1=30Hz, f2=15Hz, and frame window size m=4:

The output from DFT will contain values corresponding to the following frequencies:

[0 15 -30 -15] Hz.

the “centers” corresponding to 30Hz and 15Hz respectively would be:

[3, 2].

The frequency bins and the selected centers will be printed in the python console so that the user can check if the values are correct. Note that the DFT is symmetrical so that -30Hz corresponds to 30Hz.

**2. Execute the main.py file:**







The code will run in batch mode for all the image stacks contained in the subfolder and automatically safe the demodulated stacks for each image (one per channel) on the same folder of the Raw data.