

APPENDIX A  
ARTIFACT DESCRIPTION APPENDIX:  
ACCELERATING MICROSCOPE DATA ANALYSIS USING  
PARALLEL COMPUTING

### A. Abstract

This appendix aims to provide as much information as possible about using the software developed to accelerate the single-molecule localization microscopy technique, FPALM. The source code will be publicly available to download on a public github repository.

### B. Description

#### 1) Check-list (artifact meta information):

- **Algorithm:** Uses the FPALM algorithm described by Dr. Laplante and Dr. Fang in their paper "High-Speed Super-Resolution Imaging of Live Fission Yeast Cells". [https://link.springer.com/protocol/10.1007/978-1-4939-3145-3\\_4](https://link.springer.com/protocol/10.1007/978-1-4939-3145-3_4)
- **Program:** C++ binary and CUDA Code
- **Compilation:** Compiles using 'make' once all of the dependencies are installed. Uses the g++ and nvcc compilers.
- **Data set:** A dataset contains 8,000 256x256 uint16 images which will be provided as requested. I plan on hosting it readily available in a Google Drive link.
- **Execution:** There is a command line interface and a python gui. The python gui just takes the input and generates a terminal command.
- **Output:** The output generated are statistics plots from gnuplot, a super resolution image, and a matlab data file containing all quantitative data.
- **Publicly available?:** Currently, the optimized implementation is public on github; the original implementation can be provided when requested. Furthermore, additional test datasets can be provided when requested.

2) *How software can be obtained (if available):* Either by QR code or by url link: <https://github.com/laplante-lab/la-fpalm.git>

*Obligatory if the paper contains computational results.*

3) *Hardware dependencies:* The only exact hardware dependency is an Nvidia GPU to run the CUDA kernel; a Titan Xp (compute capability 6.1) was used.

Type	Name	Summary
CPU	Intel Xeon E5-2620 v4	any x86-64 cpu should work.
GPU	Nvidia Titan Xp	2 GPU were used; using only 1 GPU will work too.
RAM	128 GB	Less memory will limit the amount of CPU parallelism for larger dataset.

4) *Software dependencies:* Currently, the software has only been tested on ubuntu 16.04 LTS. I modified a version to work on Windows 10, and support windows 10 using cmake in the future. The following are the current software requirements:

Type	Name	Summary
OS	Ubuntu 16.04	Our lab server runs on this.
Library	DIPLib	Open source c++ quantitative image library. <a href="https://github.com/DIPLib/diplib">github.com/DIPLib/diplib</a>
Library	matio	Allows I/O of MATLAB data files <a href="https://github.com/tbeu/matio">github.com/tbeu/matio</a>
Library	nd2sdk	Enable input of raw Nikon data files (.nd2) <a href="https://nd2sdk.com">nd2sdk.com</a>
Compiler	g++	sudo apt-get install g++
Compiler	CUDA	nvcc is needed to compile CUDA code.

5) *Datasets:* The datasets were captured from an Nikon Microscope and saved to an .nd2 file. The .nd2 file contained all of the raw 256x256 images.

### C. Installation

Installation involves installing two open source libraries on Github. Following the instructions provided in the README.md in each repo should be sufficient. Also a compiled nd2sdk for ubuntu 64-bit is included in our github repo; however, the compiled binaries for different operating systems can be obtained from the nd2sdk website linked in the table above.

### D. Experiment workflow

Once data is obtained, you can the program as such:

```
make
./la-fpalm input-data.nd2 output_folder/
```

This will run the analysis using default parameters and a default noise map. A custom sCMOS cammer noise map will need to be generated for optimal analysis for different microscopes.

### E. Evaluation and expected result

The time spent and accuracy of the result can be compared with the Matlab implementation if requested, since the code was not originally open sourced.

At the least, the data should make sense looking at the super resolution reconstruction, and the generated plot should give a rough understanding of quality of data. More tools are being developed to further analyze the results of this FPALM technique.