SLIMER LONG WRITEUP

MICHAEL RONQUEST
ERIK SHAW
MARY KIDD
STEVE ELLIOTT

1. Introduction

This document outlines the main findings of the SLIMER experiment.

2. Description of the Apparatus

2.1. **Microscope Body.**

2.2. **Lenses.**

2.3. **EM CCD Camera.**

2.4. **Fluorescence System.**

3. Recommended Procedure for Taking Data

3.1. **Power-up procedure.** Describe the order in which things should be powered up and/or started.

3.2. **Recommended EMCCD settings.** Describe things like EM gain, BERT settings, etc. here

3.3. **Focusing Procedure.**

3.4. **Data Taking Settings.**

4. Data Analysis

The analysis code is written in Python, and utilizies a number of external libraries to enable reasonably fast analysis of data. These libraries include Numpy, Scipy, PANDAS as well as PIL. These may all be obtained within the Anaconda Python package.

4.1. **Locations on WIT.** The copy of Anaconda, which includes the python distribution and libraries used in this analysis, can be setup by sourcing this file: XXXXXXXX Note that this will also setup a copy of ROOT which can interact with the newer version of python.

We now use a seperate program on WIT to convert from nd2 files to tiff files. The location is here: proj/Software/ImageProcessing/bftools/bfconvert

4.2. **Analysis Chain.** The data flow, from data taking to final histograms is:

- Nikon Elements Data Acquisition → 1 `.nd2` file per run
- 1 `.nd2` file per run → `bfconvert` → N `.tiff` files, 1 per exposure
- Background `.tiff` files → `background_average.py` → 1 `.npz` (NumPy Zip archive) file, which contains 2 `ndarray`s representing the mean and variance for each pixel in the EMCCD camera
- Background `.npz` file, data `.tiff` files → `image_analysis.py` → crunched data (energy, position, etc..) in ASCII (`.dat`) format as well as HDF5 (`.h5`) format
- Derived observable (`.h5`) files → `slimer_ana.py` (cuts defined in python script) → crunched data with cuts in HDF5 (`.h5`) format as well as histograms in ROOT (`.root`) format

The main part of the data reduction takes place within `image_analysis.py`, where clustering and other analysis is performed. This is computationally expensive, however this stage can be done in parallel on wit, with the output concatanted afterward.

5. Results

6. 14C spectra and efficiency

show some plots, and estimate the DETECTION efficiency. Also estimate the energy threshold.

7. Observed Problems, Recommended Solutions

7.1. **Light Leaks.**

7.2. **Communication Problems.** Detail the Elements to Micrscope comm problems which would modify the filter settings here

7.3. **Focusing Problems.** Erik found that the focus at the 20x setting changes when the source is placed/removed. Not touching the scintillator slide seems to help, but this may be a weight issue. He has confirmed that there is NOT a time-dependent drift in focus. NEED TO CLAMP THE SCINTILLATOR BETTER.

7.4. **Varying Background Level.** Erik has found a difference in the variance on the background level (the raw pixel counts) between two background runs taken during different days. The shift in variance is from 7 to 10 counts/pixel.

He has also seen a variance in the mean pixel count on an image by image basis. This agrees with the behavior seen in the pixel count histogram in the Nikon elements DAQ software

7.5. **Optimum Magnification.** We have consistenly been able to produce good observations at 20x magnification, with few background events satisfying our analysis criteria. However, to check if this magnification was optimum, we also ran at 10x magnification, with all other DAQ settings the same. For these data, a number of things were observed:

- The energy spectrum, in units of pixel counts, shifted to lower values relative to the 20x magnification

BUT WE NEED TO ENSURE WE UNDERTAND THE CHANGES REQUIRED TO SWITCH TO A DIFFERENT MANGIFICATION!

Different lenses have different numerical apertures.