DATE: 02 June 2021

Code versions:
gridfit.m
TentSim_v1p7.mlx
Tentsim_Preprocess_v1p0.mlx
Tentsim_BAS_v1p0.mlx

Running Matlab 2021a but built on 2020a

PRIMARY GOALS:
0. Get use to simulation and analysis software; Begin to understand how code can be modified for additional features
1. Assess how concentration and binning parameters affect the single BAS plots of a single population
2. Generate/document 5 sets of simulated data for later use between 1e-12 to 5e-10

TASKS
a. Try producing sim data with a single population at various concentration (change firebase name line 528 and value of Particle concentration on line 42)
b. Also, try different run files and repeatfiles values (line 103,104) to confirm similar BAS results
c. Try different nbinsA and nbinsB values (w/ nbinsA=nbinsB) along with different minthresh and maxthresh (lines168 - 184 in Tentsim_BAS_v1p0.mlx) so see affect on BAS plots .
d. Not all of the trial outputs need to be saved (will be too many files)!!! Save files that demonstrate interesting/important features only
e. Keep notes on ways to streamline/improve input and output
f. Keep notes on questions about code function

1. Confirm all dependancies/toolboxes

'gausswin' requires Signal Processing Toolbox.

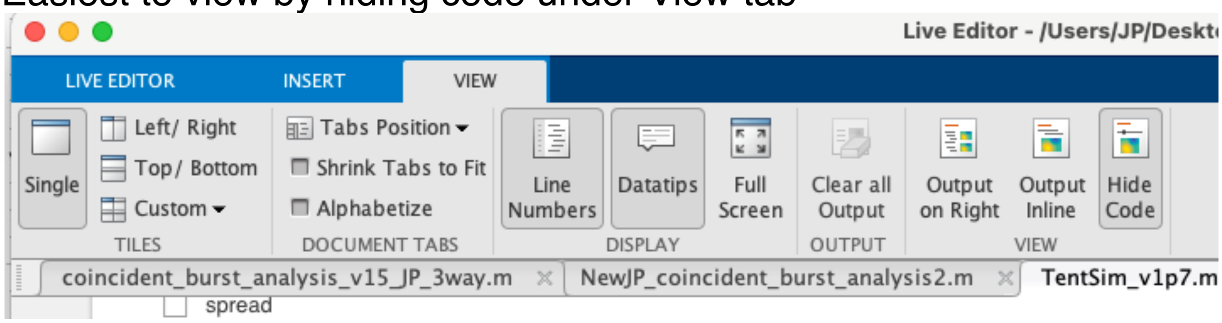
'std2' requires Image Processing Toolbox.

'gmdistribution' requires Statistics and Machine Learning Toolbox.

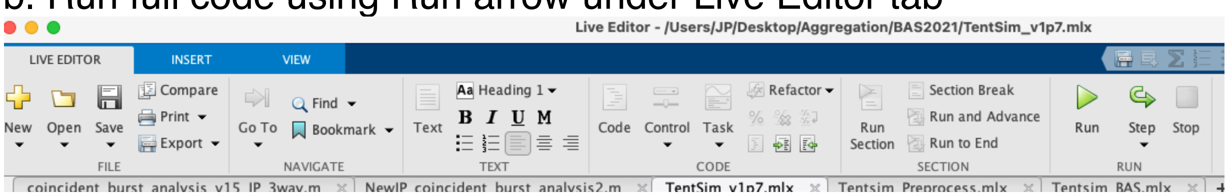
'fittype' requires Curve Fitting Toolbox.

2. Beginning with TentSim_v1p7.mlx

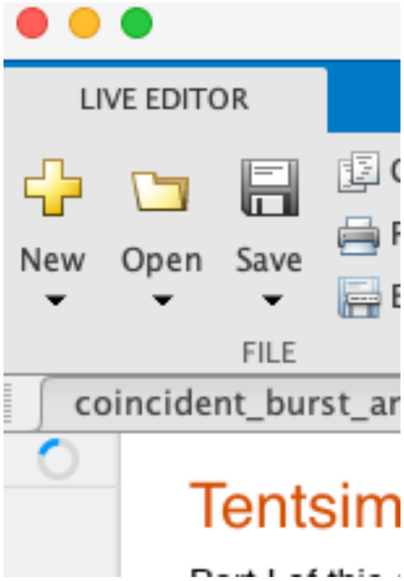
a. Set parameters consistent with those in TentSim_v1p7_DefaultValues.pdf
Easiest to view by hiding code under View tab



b. Run full code using Run arrow under Live Editor tab



c. Spinning blue circle next to top of code on left indicates run in progress



d. Inline plots will be generated and simulated data will be created in an automatically created subfolder of working directory ; Screen output can be saved using the Export pulldown on the Live Editor tab (see b above). SAVE THIS PDF FOR ANY SIM DATA TO BE USED IN ADDITIONAL ANALYSIS - USE THE SAME ROOT NAME AS THE CREATED DATA FILES. Output should look like TentSim_v1p7_DefaultOutput.pdf
Example names created:
i. Folder TentSim_v1p7_02Jun21-115808 created using this code version on June 2 at 11:58:08 am
ii. Multiple sim data files
Sim_200pM_56060002Jun21-153105_1_1.txt where -#-# indicated run and repeat numbers where runs are data collected from same sample placement from a prep and repeats are data collected from a fresh placement/slide but same overall prep
iii. A parameter file for preprocessing is created automatically :
Sim_200pM_56060002Jun21-153105-preparams.txt to be used in preprocessing
iv. The output file SimReport.pdf contains figures from *previous* run of code. Apparently, this utility is not officially supported. Instead, use the Export pulldown as discussed above to copy base sim data name and append OUTPUT :
Sim_200pM_56060002Jun21-153105_OUTPUT

!!!Sim datafile text headers are coded manually and NOT currently tied to Tentsim parameters. All internal parameters in code are passed though stored .mat files.

3. Run Tentsim_preprocess_v1p0.mlx

a. Output parameter file associated with data files will be read by this code (Example: Sim_200pM_56060002Jun21-153105-preparams.txt); this file contains names of all data files to be used in preprocessing - can modify file names included as desired. Header can have additional experiment info; the parameters associated with preprocessing include: headlines colA colB sampletime driftwinA driftwinB basethreshA basethreshB corrt dofit . See .mlx code for definitions of these parameters

b. After run, output data saved to .mat file (Example: Preprocess_v1p0Sim_200pM_56060002Jun21-153105-preparams.txt02Jun21-173045.mat)
These files are tagged with original input data file name and timestamp of this processing.

c. Save output of preprocessing using the Export button manually (Example: Preprocess_v1p0Sim_200pM_56060002Jun21-153105-preparams.txt02Jun21-173045_OUTPUT.pdf)

4. Run Tentsim_BAS_v1p0.mlx

a. Create a BAS parameters file (Example: BASparams_Example.txt) containing: OffsetA OffsetB smooth showfig (See .mlx file for definitions)

b. Program will read the BAS parameter file as well as the preprocessing .mat output file from Tentsim_preprocess

c. BAS results stored in .mat file (Example:

BAS_v1p0Sim_200pM_56060002Jun21-153105-preparams.txt03Jun21-010111RESULTS.mat)

d. Use Export button create pdf of screen output (Example: BAS_v1p0Sim_200pM_56060002Jun21-153105-preparams.txt03Jun21-010111OUTPUT.pdf)

——

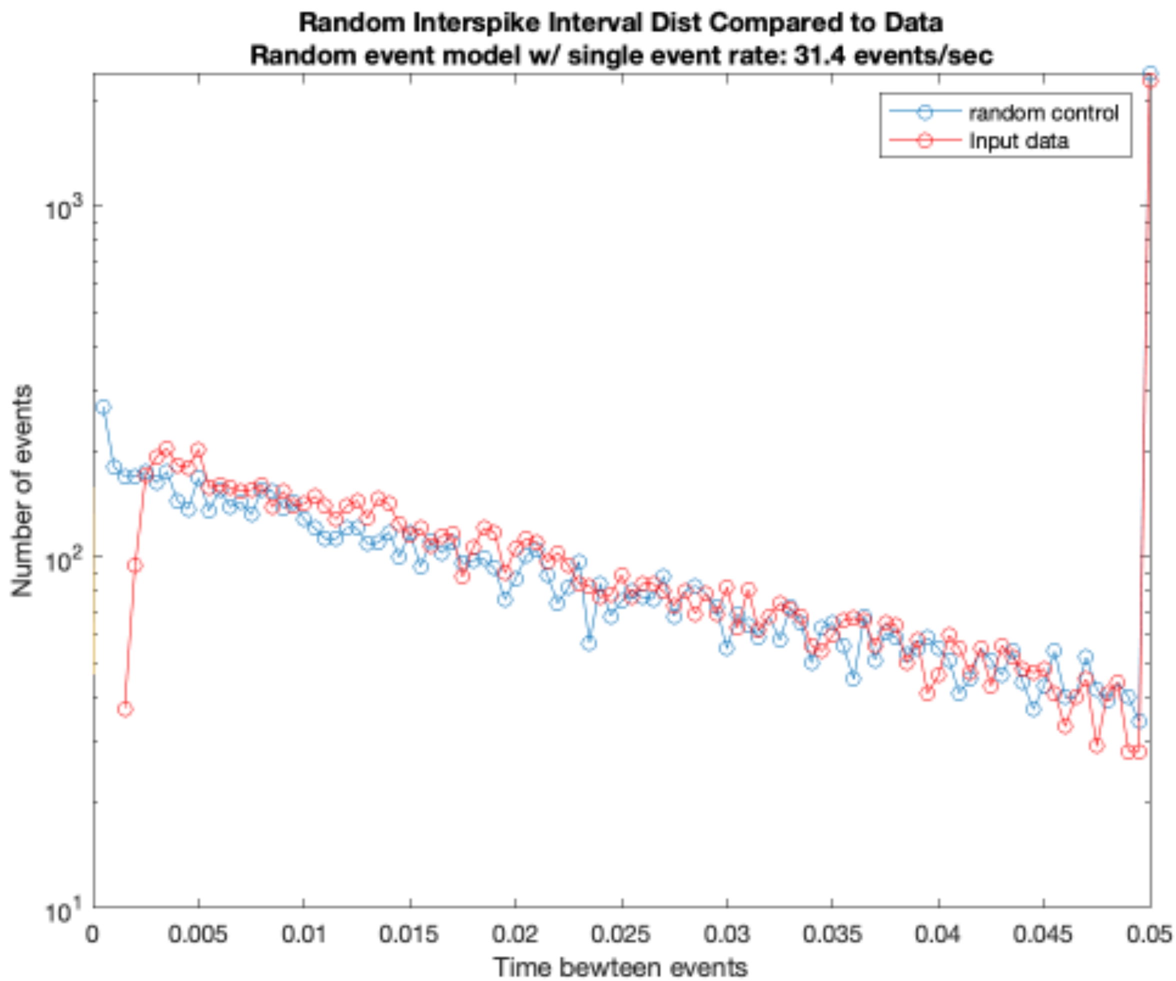
DATE: 07 June 2021

When running preprocess, the timestream rms and mean are measured. These numbers are first found in the “raw” form (complete dataset) then in a 5-sigma form where largest spikes are masked from the calculation so both values are determined from “non-spike” values. This is a heuristic approach to estimating if there is a fluorescent background component in the data. However, highly concentrated data will have a significant 5-sigma rms even w/o an extra background component. The output of the code will show you the 5-sigma rms and mean for each color channel:

```
handles = struct with fields:
            base: [1Ã—720000 double]
             rms: 26.3992
          median: 0
            mean: 4.9433
               d: [2Ã—720000 double]
    spthreshtype: "rms"
```

The parameter spthreshtype sets the value of a baseline cutoff in each channel. In the is example, all events with amplitude less than 26.3992 will be zeroed. Note that it is not clear whether `spthreshtype: "rms"` or `spthreshtype: “mean”` or `spthreshtype: “zero”` is better for the analysis (see line 265 in Preprocess code). Part of the summer simulation work will be to explore this threshold and the affect.

For now, let’s leave the type as “rms” as it is most conservative. Then, in concentrated samples or ones with a significant background, the rms will be > 1/100 of the maximum amplitude. Also, the this means events lower than rms will be zeroed so some of the preprocessing output plots are affected like this one below. On the left, there is a deficit of events that occur closest in time which are also the ones of lowest amplitude (see drop on the left for the red data curve)



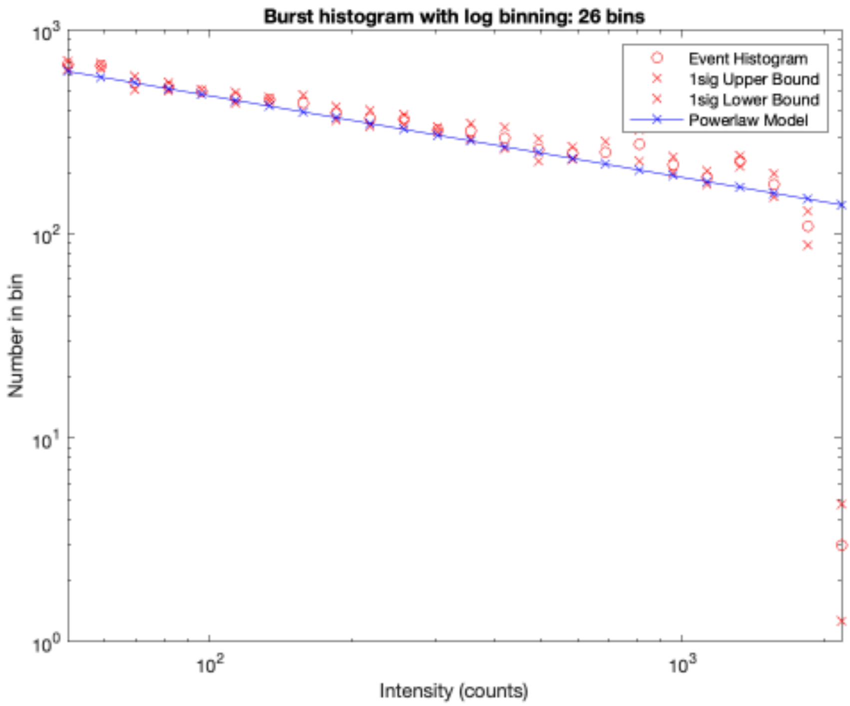
When running BAS code, we then need to set the minthresh > rms for each channel. In this case, below I set it to 50 (a bit conservative but just to be completely safe). If you set it too low then

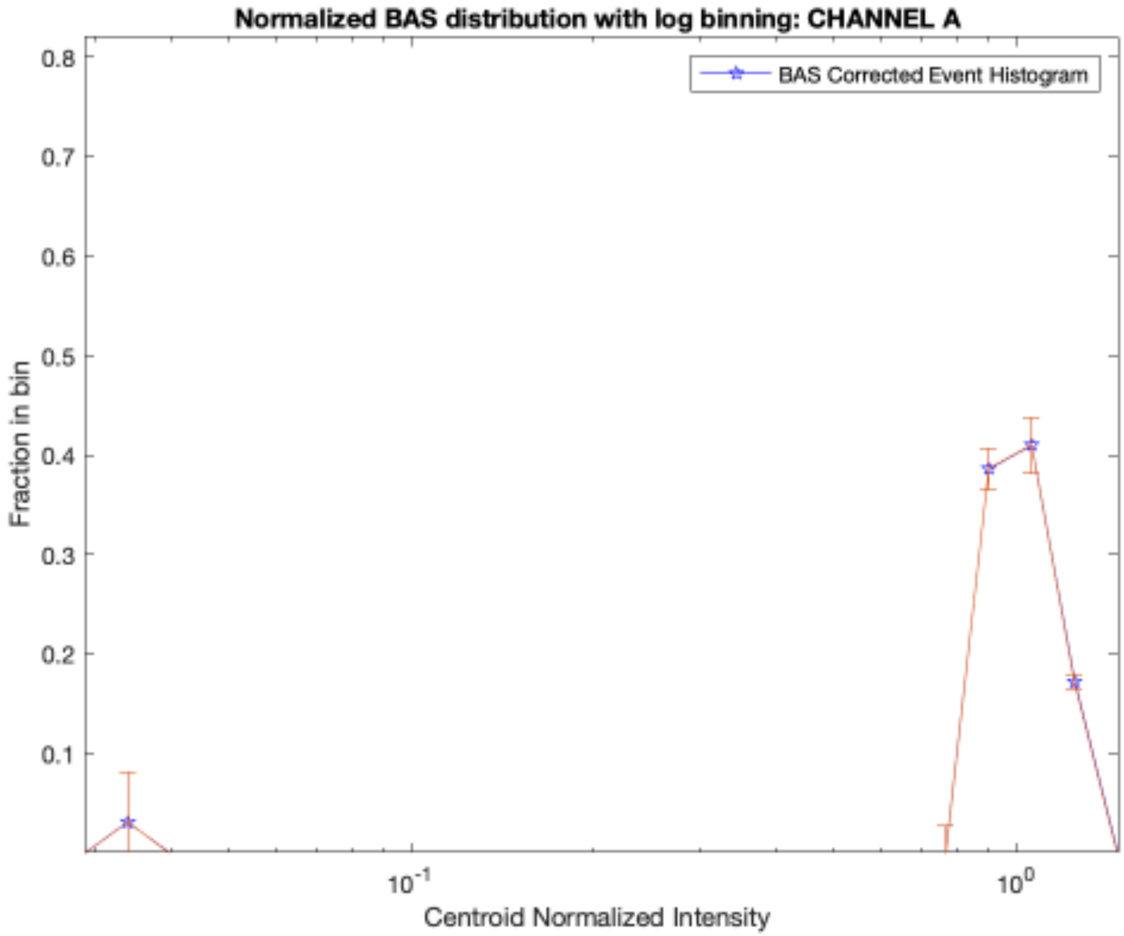
Find photon histogram

```
Calculate Event Histogram

%Initialize BAS results
logedgeA=0; logedgeB=0; uncertnA=0; uncertnB=0;
qA=0; qB=0; nA=0; nB=0; numA=0; numB=0; cummA=0;  cummB=0; nseg=1; part=1;

if colA
                    nbinsA = 26
                    ptypeA = Log
% See Minimum/Maximum non-zero event amplitude output to set values
                    minthreshA =  50
                    maxthreshA =  3010

 [logedgeA nA uncertnA]=photon_hist(handlesA,datastatA,nbinsA,minthreshA,maxthreshA,tbin,ptypeA,nseg,part);

end
if colB

                    nbinsB = 26
                    ptypeB = Log
% See Minimum/Maximum non-zero event amplitude output to set values
                    minthreshB =  50
                    maxthreshB =  3010


 [logedgeB nB uncertnB]=photon_hist(handlesB,datastatB,nbinsB,minthreshB,maxthreshB,tbin,ptypeB,nseg,part);


end
```

The output is now truncated at an amplitude of 50 and the histogram doesn’t have the low amplitude roll off:

```
nbinsA = 26
ptypeA = "Log"

minthreshA = 50
maxthreshA = 3010
Bin edges: 50      58.905231      69.396525      81.756367      96.317554      113.47215
```




Normalized BAS distribution with log binning: CHANNEL A
BAS Corrected Event Histogram
Fraction in bin
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
10^{-1}
10^{0}
Centroid Normalized Intensity