“FLIM-FLAM” Exponential Fitting Code User Manual

Time-correlated single photon counting (TCSPC) data are generally fit to exponential decay models, allowing results to be represented with summary parameters such as the weighted average lifetime. While both commercially available and open source software for this purpose exist, we found that no individual package implemented all or even most of our required functionalities. To remedy this, we developed a MATLAB pipeline (FLIM fluorescence lifetime analysis module, or FLIM-FLAM) capable of processing fluorescence lifetime imaging data in an automated fashion. This improvement to the analysis workflow was essential both for development of reproducible fitting models and for the advancements in throughput made by VF-FLIM.

The purpose of FLIM-FLAM is to process, analyze and visualize fluorescence lifetime imaging data. The specifics of this workflow were optimized towards membrane-localized fluorescence signal in mammalian cultured cell lines, but much of the package is generalizable to any application.

This workflow is broken down into three conceptual steps (below). Our modular design allows each of these steps to be called individually or as part of a sequential workflow.

1. Process: Identify the regions of interest (or pixels) within TCSPC data. Extract decays for fitting.
2. Analyze: Fit the fluorescence decays to the specified model. In this case, the model is a sum of 1, 2, 3, or 4 exponential decays.
3. Visualize: Rebuild the fluorescence lifetime data into image format or summarize it by time point and region of interest for plotting of summarized results.

The fitting is performed through iterative reconvolution of a guess model with the instrument response function (IRF). Here, we use a weighted least squares algorithm, together with the built-in fmincon optimization algorithm in Matlab. In addition to the fit output, the program will provide various estimates of goodness-of-fit, as well as inputs re-saved in .mat format for easy reprocessing. Most output files will by default save to the folder where the input data are stored.

The code requires raw photon counting histograms (.sdt or .bin exports), as well as metadata and configuration files. See below (and included examples) for documentation of the format of these additional files. The code requires installation of the BioFormats Matlab package for parsing the .sdt file format for raw photon count data from Becker & Hickl TCSPC systems. Additional documentation can be found within the code files and working examples.

Data Import

The wrapper code for the file parsers (importData.m) accepts four different import modes, which determine how the data are broken down. The below descriptions depict how an image of X pixels by Y pixels by N time channels would be handled by the software.

1. “single”: Combines all of the pixels X by Y into a single N by 1 decay for each image.
2. “pxwise”: Retains the native spatial resolution and returns an X by Y by N array.
3. “global_fiji”: Parses ROIs provided by the user as binary exports from ImageJ (FIJI). This mode uses these masks to define regions of interest for global analysis.

4. "global_btm": Uses thresholding code optimized for finding membranes in confocal images of cultured cells to identify ROIs for global analysis. The user has an opportunity to modify and verify ROIs generated by this "batch trace membranes" (BTM) routine.

Once the data have been imported, there are two exponential fitting functions easily available to the user. For global analysis, fitDataStruct.m will process the output of a global import and return a structure with the exponential decay fit results as well as any metadata. For pixelwise analysis, pxwiseAnalysis.m will fit the output of a "pxwise" import and render images of the components of interest, as well as overlays of these components on the photon count image. Average lifetimes for regions of interest in the images can be determined after pixelwise fitting using the provided script batchTraceFromStruct.m. This implements a similar routine to the segmentation performed in global_btm but acts on the fit images instead of the raw photon count histograms.

Region of Interest Identification

The scripts adapt the routines published previously in batchTraceMembranes2[6] to enable automated identification of contiguous membrane regions. These regions can be applied either before the fitting (global analysis) or after the fitting (pixelwise analysis). The user has the opportunity to remove debris, merge ROIs that were erroneously split, and approve the final ROIs within the batch tracing script. This processing mode works best if the thresholding is tweaked slightly to reflect the morphology and characteristics of the cell line, so the function takes the cell type as an input parameter. Currently supported cell lines are A431, CHO, HEK293T, MCF-7, and MDA-MB-231. In addition, the option HEK293T_red is available for settings with reduced autofluorescence; use this if HEK293T is causing oversegmentation. If the cell line name is not recognized, the defaults for HEK293T are used.

If no spatial resolution is desired, the global decay for the entire photon histogram can also be selected. More complex ROIs should be generated as TIFF masks (e.g. in FIJI). These can then be imported and applied as masks to the data (0 is background and values >1 are kept).

Function Usage Overview

The sample sequences of function calls (all MATLAB code files) for five common workflows are listed here for reference. Additional documentation on usage is available within each individual function file (*.m).

1. Pixelwise fitting and rendering of FLIM images: importData, pxwiseAnalysis
2. Pixelwise fitting of FLIM images, followed by averaging across automatically identified membrane regions of interest: importData, pxwiseAnalysis, batchTraceFromStruct
3. Global fitting, in any mode (single decay per image or decays defined by regions of interest): importData, fitDataStruct
4. Rendering TIFF photon count images for defining ROIs on data without processing in any other way: renderPhotonsImg
5. Rendering of TIFF images of parameters that were not previously processed from a pixelwise analysis: renderImage

Formatting of Input FLIM Data Files and Metadata Files

Filename Formats for Data Records

FLIM records should be imported as photon histograms generated either by the binary export in SymPhoTime (*.bin) or as the raw acquisition format in SPCM (*.sdt). Opening of sdt files relies on the BioFormats package for MATLAB. Data records can be uniquely identified by a combination of the following parameters:

1. Date recorded, generally YYYY-MM-DD
2. Coverslip or sample identifier (abbreviated 'cID', CC).
3. Image identifier (abbreviated as 'imageID', II). This is a unique field of view within the sample.
4. Replicate identifier (abbreviated 'repID', RR). This is the number of times a FLIM images was recorded from a particular field of view. It is almost always 1, but this parameter is included to allow for re-acquisition of images if technical difficulties arose or for comparisons of signal stability.
5. Frame identifier (abbreviated 'frameID', XX). This is the slice of a time series (in experiment time) that the image corresponds to. It is more generally used than replicate ID. For instance, if I recorded two time series of ten images each from the same field of view, they would have the (repID, frameID) combinations of (1, 1-10) and (2, 1-10).

We save our FLIM files to a regularized format so that the file name strings can be parsed to obtain information about the various IDs in the recordings. If filenames are not in the format below, they will need to be renamed for proper functioning of the multilevel ID system.

General format of data filename string, before the .bin or .sdt file extension:

YYYY-MM-DD_CC-II-RR_anything_else_you_wantXXX

	Parsing Results				
Input name	**Date**	**cID**	**imageID**	**repID**	**frameID**
2019-10-12_01-1_bob.bin	2019-10-12	1	1	1	N/A
2019-10-12_5-06-3_awesome.bin	2019-10-12	5	6	3	N/A
2019-10-12_5-06-3_ralph011.bin	2019-10-12	5	6	3	11
2019-10-12_010-02_ralph_1.bin	2019-10-12	10	2	1	1

Table A1-1: Sample parsing results for various filename strings.

This is not an exhaustive list of acceptable strings, but it is intended to give the user a sense of parser requirements. The software is flexible regarding the number of digits in CC, II, and RR as long as they are separated by dashes. 02-03-02 and 2-3-2 will give the same result. The RR is frequently not present; it will be assigned to 1 if it is missing. The XXX frameID is only processed if nFrames is set in the metadata (see below). The character before the start of the frameID does not matter as long as it is not another number. Not applicable (N/A) means that the field will not exist in the output data.

Configuration and Metadata File Formats

This code requires configuration and metadata files to select fit parameters and group data files into categories. In general, these files should be saved in a comma separated values (csv) format in which the first row is the categories and each subsequent row corresponds to a unique record. The analysis suite looks for exact matches for certain categories/fields in the input metadata to properly complete the analysis. Samples of these files are available within the software package, and the fields required are enumerated here:

The configuration file indicates standard fit settings and instrument properties. Most, if not all, of the metadata about acquisition settings is lost upon exporting to a histogrammed photon images (especially in the SymPhoTime software package from PicoQuant), so it needs to be re-entered here. Expected fields are below; order of the columns in the input file does not matter.

1. model: string or character array for the number of exponential components. Accepted values are 1exp, 2exp, 3exp, and 4exp.
2. fixedParam_X: A boolean indicating which parameters (taus and term weights) should be fixed to their starting values (fixedParam_X = 0) and which parameters should be allowed to vary (fixedParam_X = 1) during the fitting. The number _X at the end of the name specifies the parameter, with all of the coefficients numbered before all of the lifetimes. For example, for a biexponential fit, X = [1 2 3 4] for [a1 a2 tau1 tau2]. For a monoexponential fit, fixedParam is set to -1 by default in the software.
3. startParam_X: Floating point fields determining the starting value for each parameter, numbered as for fixedParam_X. If the parameter is fixed, it will be fixed to the input value of startParam_X. floating-point fields for each coefficient and tau to be fit. X is the index of the parameter. For a monoexponential fit, only provide the starting lifetime as startParam_1.
4. offsetMode: parameter indicating whether the offset (dark counts) should be fixed to zero or allowed to vary in the fitting. If offsetMode is 0, then the offset will be allowed to vary in the fitting. If offsetMode is 1, the offset will be fixed to 0. If offsetMode is 2, the offset will be set from the average of the time bins before the start of fitting. If the decay does not completely return to background levels within the period of the laser, offsetMode = 1 should be used (with the caveat that dark counts and stray light background must be low).
5. cShift: floating-point number corresponding to the shift between the IRF and the decay to be fit, in units of ADC time bins.
6. shiftFixed: boolean indicating whether the shift should be fixed to the starting value or used as a free parameter in the fit.
7. nsPeriod: The period of the laser cycle during the acquisition in nanoseconds. For 80 MHz laser repetition rate, the nsPeriod is 12.5.
8. irfStart: The beginning of the IRF (in ADC time bins). This enables cropping of a measured IRF to just the segment corresponding to the laser pulse.
9. irfEnd: The end of the IRF (in ADC time bins).
10. fitStart: The first ADC time bin to be used in the fit.
11. fitEnd: The final ADC time bin to be used in the fit. Note that the fit may stop before this point if a zero is encountered in an ADC time bin to be fit. If zeros are present in the time channels, the fit will stop at the first time bin with zero recorded photon counts.
12. threshold: number of photons minimum in a decay to be processed (sum across all time, not the peak value). This value is only used in pixelwise analysis; global analysis will be performed regardless of the number of photons.

13. irfMode: Accepted values are “single” or “paired.” This parameter indicates whether each IRF file will correspond to an IRF index or if the average of sequential IRF files should be used to generate an IRF for each IRF index.
14. adcRes: The expected number of time channels from the analog to digital converter (N in X by Y by N photon histograms). This value should be constant across all data and IRFs analyzed. It is provided as a fail-safe check in the configuration file to verify that nothing strange happened to the data upon exporting.
15. adcBin: The binning in the time dimension. For example, an adcBin of 1 does not bin, whereas an adcBin of 2 would sum paired adjacent time windows of a 256 bin decay into 128 resulting time windows.
16. viewDecay: Default value of 0 does not show the fit curve for each exponential decay. When set to 1, it will display the fit and residuals for each curve being processed during the fitting processing.
17. maxFunEval: Constraint on the number of function evaluations the fmincon fitting algorithm can perform. See Matlab documentation for more information.
18. stepTol: Convergence criterion in the Matlab built-in fmincon algorithm. See Matlab documentation for more information.
19. optimTol: Convergence criterion in the Matlab built-in fmincon algorithm. See Matlab documentation for more information.
20. constraintTol: Criterion in the Matlab built-in fmincon algorithm regarding bounds. Satisfying it does not stop the solver. See Matlab documentation for more information.
21. ccvColor: Color map for displaying the “color coded value,” usually the lifetime. Accepts Matlab build-in maps (e.g. ‘parula’ or ‘jet’).
22. spatialBin: The bin factor for pixelwise fitting; see binType below.
23. binType: Type of binning to be used with the spatialBin parameter. Acceptable values are (1) ‘std’: square binning (e.g. a *spatialBin* of 2 will lead to 2x2 square binning) or (2) ‘bh’: moving average binning, where each pixel is summed with all pixels *spatialBin* units away, thereby upsampling photons and nominally retaining the native spatial resolution of the image.

The metadata file provides information about the data being analyzed. It is intended to be a useful way to track categorical data and later bring output data with this information into any plotting software. Each file (defined by cID-imageID-repID) must map to a single, unique value in the metadata. The required files are below; again, order of the columns does not matter:

1. irf_index: The number of the IRF to be used in processing the data. If IRFs are imported as an average of a pair of two, this should count the pairs as a group, not as two individual files.
2. cID: The coverslip or sample ID of the record. If no other IDs are provided, all images

Certain special fields in the metadata file are not required, but their presence will change how the data are processed. Any fields beyond the required and special fields will be carried forward and assigned to the appropriate records, but they will not affect the analysis. Special fields are below:

1. imageID: If an imageID is provided, then all imageIDs in the input data must have a match in the metadata, and records will be matched to metadata based on both cID and imageID.

2. repID: Similar to imageID. It is only used for matching records with metadata if it is provided.
3. nFrames: The number of frames in the record. If this is specified, the file name parser will look for and assign the frameID based on the end of the filename.
4. cellType: This field must be provided if the user would like to access the automated membrane tracing in “BTM” modes either before or after fitting exponential decays. Supported values are “A431”, “CHO”, “HEK293T”, “MCF-7” and “MDA-MB-231,” which threshold the image slightly differently to best capture membrane given the typical morphology of that cell line. If cellType does not match one of these values, the code will use the settings for HEK293T cells.
5. smallestObject: This field must be provided if the user would like to access the automated membrane tracing in “BTM.” The units are in pixel areas. The script will assume that regions smaller than smallestObject are debris and will remove them automatically.

If the user is importing ROIs as TIFF masks (e.g. those generated in FIJI, see Materials and Methods), there are two options for matching the ROIs to the relevant images. This option is determined by a command prompt to the user, and it is specified within the function as the mode string (either “csv” or “parse”).

1. parse: Use the structure of the ROI name to determine which image the apply the ROI to. The ROI must be named in the format CC-II-RR_NN, where NN is the numerical identifier of that ROI (the number of ROIs for that image). CC, II, and RR are the cID, imageID, and repID as described above. Names in the format CC-II, CC-II-RR_NN, and CC-II_NN also are accepted. Missing fields are assumed to be equal to 1. Parse mode can be used for time series but the same ROIs will be applied to every frame of the time series. If changing ROIs are desired for a time series, they must be specified in the csv mode.
2. csv: Uses a separate ROI metadata file (*.csv) that contains the cID, imageID, repID, and frameID of the image as well as the exact name of the ROI (minus the file extension). cID and imageID must be present; if repID or frameID is missing, it will be filled in with the value 1.