

Description

These programs implement software multi-tau correlator that acts efficiently on sparse photon records obtained in a fluorescence correlation spectroscopy (FCS) experiment. Software correlators offer several advantages over hardware correlators since they allow more sophisticated data processing such as calculation of high order correlation functions and incorporation of lifetime information (lifetime FCS and gated FCS).

Supported data types:

Flex 8-bit
Confocor2 raw
Confocor3 raw
PicoHarp 300 - *.pt3

If your data type is not supported, please submit an issue on this page, and I will incorporate your data type into FCS_calc.

Installation and operation:

1. Make sure that ImageJ is installed on your computer.
2. Download FCS_calc.java and Gmn.java source files and copy them into the ImageJ plugins folder.
3. Compile Gmn by running it from Plugins->Compile and Run... in ImageJ.
4. Compile FCS_calc by running it from Plugins->Compile and Run... in ImageJ.
5. The correlator is ready for use. To launch your calculation, select "FCS calc" under Plugins (becomes available after ImageJ is restarted).

Once you run the plugin, you will have to provide values for several parameters:

"Time base" is the lowest time resolution used for calculating correlation functions, and must always be an integer. The value of 1 results in the highest time resolution of the correlation function and the slowest calculation.

"N of cascades" and "N of points per cascade" are parameters used for constructing the semi-logarithmic X axis.

The calculation mode

'auto' (AxA);

'cross' (AxA, BxB, AxB, and BxA);

'autoHOMlt' high order correlation functions calculated by direct multiplication and NOT corrected for dead time (dead time correction will be implemented in a future release) calculation will be supported in future releases.

Other parameters are self-explanatory.

Once you've made these selections, you will be prompted to locate you data files. Multiple files can be selected for sequential calculation of the correlation function; for data types that store channels A and B separately, select channel A files and click "Open"; then do the same for channel B files.

FCS_calc.java:

FCS_calc is a plugin for ImageJ that provides user interface for the software correlator realized in Gmn.java. This user interface is minimal but should be sufficient for demonstrating utility of Gmn.java. You may want to optimize FCS_calc for your application by removing data types you do not use, changing default values, etc. You may also want to make sure that the time resolution (dt, in seconds) is defined correctly for your type of data. This value does not affect the calculation in any way but changes the scale of the X axis.

Gmn.java:

Gmn.java is an implementation of a fast algorithm for calculating cross-correlation functions. The Gmn class has an initialization method, several utility methods, methods for converting input data files into a universal format with inter-photon arrival times and photon weights, and methods for calculating auto and cross correlations.

I believe this program is sufficiently fast to be useful for on-the-fly calculations. While it is not interfaced with any specific hardware, it should be relatively straightforward to incorporate Gmn into a custom program that controls your data collection hardware. An example program (**online_auto.java**) is provided to illustrate this application. If you are writing a custom program to control your hardware and would like to incorporate Gmn, you are welcome to contact me with any questions.

All calculations operate on an array photon interarrival times. For example, if you have the following record of photon arrival times

```
ch.A  18
ch.A  25
ch.B  39
ch.A  55
ch.B  81
```

the following arrays should be provided to Gmn:

```
photonsIat  [18, 7, 14, 16, 26]
wAint  [1, 1, 0, 1, 0]
wBint  [0, 0, 1, 0, 1]
```

Simplification to a single channel is straightforward, and in this case all photon weights will be equal to 1.

Algorithm:

The correlation algorithm realized by Gmn is based on Yang et al. (2009) Real-time data acquisition incorporating high-speed software correlator for single molecule spectroscopy. J of Microscopy 234, 302-310.

Symmetric normalization and the multi-tau scheme of time coarsening are described in Schatzel et al. (1988) Photon correlation measurement at large lag times: improving statistical accuracy. J of Modern Optics 35, 711-718.

Wahl et al. (2003) Fast calculation of fluorescence correlation data with asynchronous time-correlated single-photon counting. Optics Express 11, 3583-3591.

Current implementation of high order correlation functions ('autoHOMlt') is described in Melnykov and Hall (2009) Revival of high-order fluorescence correlation analysis: generalized theory

and biochemical applications. J Phys Chem B 113, 15629-15638.

Software version:

These instructions are for **version 1.1**. The main differences from 1.0 are:

1. Calculation of intensity traces.
2. Support for *.pt3 files.
3. Calculation of high order correlation functions.

Future additions:

1. Correction of the high order correlation functions for dead time effects.
2. Calculation of higher order auto-correlations using sub-binning approach.
3. Masking of intensity traces to eliminate spikes from calculations.
4. Support for other data formats.
5. C code corresponding to all Java methods.

Edited on 08/29/17