You have to select a region of interest (ROI) in your image: the ROI includes a vessel or a part of a vesselwhere you see diagonal stripes. The ROI has to be saved in the tif image file provided to the code.

Then you have to compute the cross-correlation function among pairs of columns in the ROI. In the code you have to initialize 2 parameters: columnlimit and columnstep. Cross-correlation functions are computed for all the possibile distances from 0 to columnlimit, with step equal to columnstep. For example if you select columnstep=5 and columnlimit=20, you cross-correlate all the columns at distance 0, then all the columns at distance 5, then all the columns at distance 10 and finally all the columns at distance 15. The code outputs an average cross-correlation function for each column distance.

The other parameter you find in the code, corrlimit, should be equal to (or less than) half of the y dimension (in pixels) of your image (due to the fact that we compute the correlation functions with FFTs). The threshold parameter could be useful in case of low S/N: if you select a threshold>0, intensity is put equal to zero in all the pixels where the raw intensity is below the threshold.

Output txt files will contain both back and forth cross-correlations. You can select the correct CCFs file (from left to right or viceversa) from the direction of the diagonal stripes in your image.

We use the software Origin to globally fit the CCFs obtained at increasing column distance by using the function described in the paper.

The code is written in Python. Usually we use Canopy to edit and run the code: it already contains the necessary PIL, scipy and numpy libraries. If you use other editors (for example active state active python), just make sure you have the libraries or download them. In canopy, you have to specify the currently active folder containing the data. The code does not need to be in the same folder.

For the fitting procedure:

we use the following fitting function

y0+b\*exp(-(dist\*dx-v\*(cos(a))\*x)^2/(w2+0.5\*a2\*s+4\*s\*d\*x))\*exp(-x\*x\*(v\*sin(a)-dx/tau)\*(v\*sin(a)-dx/tau)/(w2+0.5\*a2\*s+4\*s\*d\*x))/(4\*s\*d\*x+w2+0.5\*s\*a2)/((4\*s\*D\*x+wz2+0.5\*s\*a2)^(0.5))

b amplitude (in the global fit it is a not shared parameter)

dist distance between columns (in the global fit it is a not shared parameter since you have multiple CCFs, one for each column distance, but you can fix the parameter)

dx pixel size (fix this parameter; in this case the parameter is shared among all CCFs)

a angle (rad) between the flow velocity vector and the scan path. It is fixed to the value measured from the orientation of the vessel with respect to the x-axis (assuming that fluorophores flow in a direction parallel to the vessel wall, as in our case with RBCs). Usually we measure it from ImageJ and fix it during the fit. It is also a shared parameter.

w2 square of the laser beam waist along the radial direction (it is a shared and fixed parameter)

a2 square of the radius of the flowing objects (it is a shared and fixed parameter)

D Diffusion coefficient (it is a shared parameter; do not fix it). It must be regarded as an

effective value for large flowing objects so you will obtain a value larger than the theoretical prediction.

v velocity (shared parameter)

tau Time to acquire a line (take into account the retracing time). It is a shared and fixed parameter

y0 baseline

s it is =1 if you use single photon excitation or =2 in case of two-photon excitation (it is a shared and fix parameter)

wz2 square of the beam waist along the axial direction (it is a shared and fixed parameter)