

PIPE user manual

Photo-converted Intensity Profile Expansion (PIPE) is a fluorescence microscopy method for measuring effective diffusion coefficients of proteins in cells. PIPE consists of an experimental part and an analytical part. The experimental part can be executed in various ways on an array of imaging systems. See the England lab website (englandlab.com) for the suggested experimental protocol, which uses the Nikon A1R confocal microscope for photo-conversion and signal acquisition. The analytical part described in this manual analyzes the microscopy movies generated by the experimental protocol, and provides the users with the effective diffusion coefficient that describes the expanding intensity profile, as well as graphical reports on the raw data and quality of the fitted models. The code for the analytical part can be downloaded from englandlab.com and run from within MATLAB (2014b or newer). The code can be run as-is, and does not require setup or modifications. This user manual provides instruction for running the code with default parameters and also describes parameters that more advanced users can change to improve their results.

Run with default parameters

After downloading the .zip file from englandlab.com, extract all the .m files from the .zip file into a single folder. Open MATLAB and add that folder to the executable search path using the command ***addpath(<folder>)***, for example by going to that folder in the Current Folder window in MATLAB and typing the command ***addpath(pwd)***.

Execute PIPE with the default parameters by typing

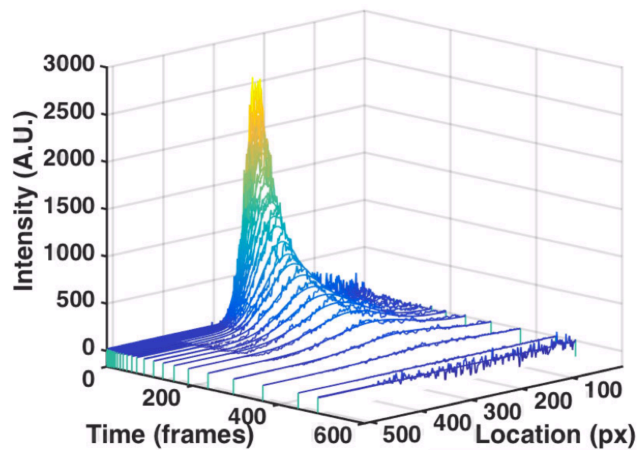
PIPE('path',<myfolder>)

where <myfolder> is the path of the folder that contains the microscopy images to be analyzed. See the following part *Input data format* for explanation on the file format and file names required for proper data input.

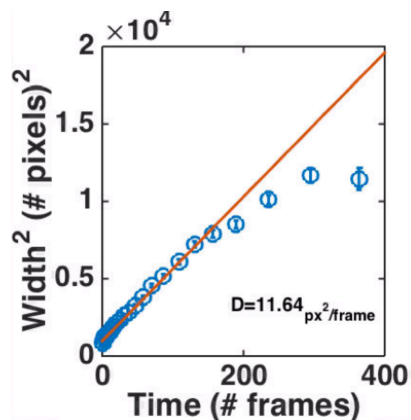
If all goes well, the analysis will finish in about one minute, and MATLAB will return a data structure that contains the diffusion coefficient and other information. It will look like this:

```
>> PIPE('path', '/Volumes/Rotem/photoconversion_data/20130813/5/')
ans =
    widths: [1x26 double]
    err_widths: [1x26 double]
    time_points: [1x26 double]
    mean_loc: [1x28 double]
    sum_r: [1x28 double]
    background: [1x28 double]
    err_background: [1x28 double]
    immobile_fraction: 1
    immobile_fraction_correction_attempted: 1
    D: 11.6395
    err_D: 2.0930
    units_of_D: 'pixels^2/frame'
```

In addition, you will see two graphical reports showing the raw data and quality of fitted models. The first report shows the raw intensity profile at different time points and the Gaussian models fitted to each time point. You can use this report to assess whether the signal indeed resembles an expanding Gaussian (see more elaborate instruction in *Quality assessment*).



The second report shows the widths of the fitted Gaussians from the first report as a function of time, and the linear fit that expresses the assumption that the observed motion follows the Fickian diffusion equation. This report can help assessing whether the temporal scaling behaves as expected. For example, in the figure below the fit describes the data well up to time frame ~200.

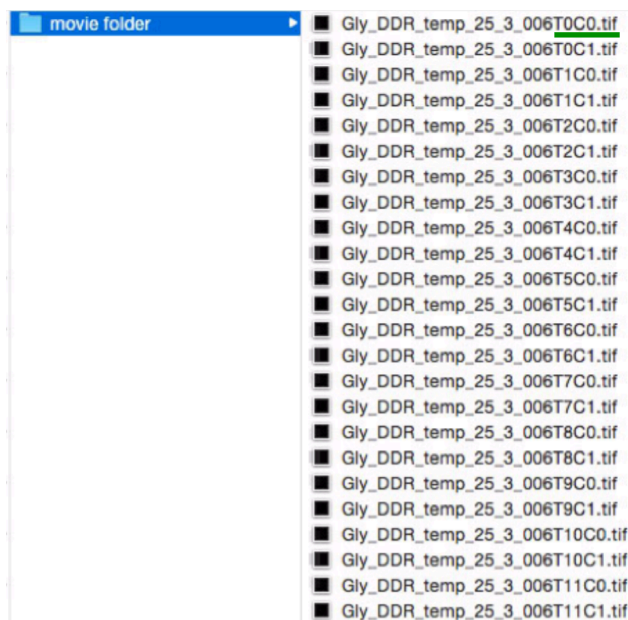


Input data format

PIPE reads microscopy movies in the form of .tif image files. All the files that belong to a single movie need to be saved under one folder. The file names should obey the following convention:

- 1) Indication of time sequence should appear as the character ‘T’ immediately followed by any number of digits, which represent the image time point. File names of consecutive images should contain consecutive time point signatures, i.e. “T0”, “T1”, “T2” and so forth. It does not matter what the first time signature is, although we have always worked with the value 0.
- 2) If the data include more than one channel (e.g. a green channel for pre-photo-conversion fluorophores and a red channel for post-photo-conversion fluorophores), the file name should contain the character ‘C’ immediately followed by a single digit representing the channel number.

An example for a valid data format:



Some software applications that control the microscope hardware (e.g. Nikon’s NIS Elements) allow exporting a movie to a folder of .tif files with names following these conventions by default.

PIPE output fields

In addition to the diffusion coefficient, PIPE returns several fields in the output data structure. Following is a description of these fields:

widths – a vector containing the widths of the Gaussian fits at all time points. Please remember that the MSD is the square widths, so the temporal scaling of this vector should not be linear, but more like the square root of time.

err_widths – a vector containing the 1σ confidence interval of the widths.

time_points – a vector containing the time points of the analyzed profiles, where $t=0$ is the point immediately after the end of the photo-conversion pulse.

mean_loc – a vector containing the locations of the Gaussian peak at every time points.

sum_r – a vector containing the sum of residuals of the Gaussian fit at every time point.

background – a vector containing the offset of the Gaussian from zero intensity.

err_background – a vector containing the 1σ confidence interval of the background.

immobile_fraction – whether or not an immobile fraction was detected in this movie.

immobile_fraction_correction_attempted – whether or not PIPE tried to correct for the immobile fraction by subtracting the mean of the last few images from the rest of the images.

D – the diffusion coefficient.

err_D – the 1σ confidence interval of D .

units_of_D – the units of D , either in $\text{pixels}^2/\text{frame}$ or in $\mu\text{m}^2/\text{sec}$.

Enhance performance by modifying input parameters

While running PIPE with default parameters should execute high quality analysis under a wide range of conditions, changing some of the parameters may help you achieve better results or give you tools to better assess the quality of the analysis. Following is a list of the parameters and how changing them may help you in analyzing your data. To change the value of a parameter, follow the standard MATLAB syntax for optional input arguments by entering the

parameter name followed by its desired value, separated by a comma (i.e. *PIPE('path', <myfolder>, <par1 name>, <par1 value>, <par2 name>, <par2 values>...)*).

Plotting

to_plot – takes either 0 or 1 (default 1). Determines whether the graphical reports will be presented to the users.

to_plot_extended – takes either 0 or 1 (default 0). Determines whether to plot the Gaussian fitting parameters, i.e. the peak location, amplitude, background and R^2 . This can help in determining how fast the signal decays, whether there is drift in the sample or in the imaging system, whether the background fluorescence photo-bleaches and how the fit quality changes with time.

plot_Gaussians_3D – takes either 0 or 1 (default 1). Determines whether to plot the fitted Gaussians on the same plot and in 3D, as shown above, or whether to plot each Gaussian in a separate subplot in 2D. The former shows more clearly the overall time evolution of the intensity profile, while the latter enables examining each Gaussian fit more closely together with the raw data. See the part *Quality assessment* for more instruction on using these graphical reports to assess the quality of analysis.

Data sectioning

max_time_point – takes an integer (default empty). Determines the maximal time point to be analyzed, counting from the end of the photo-conversion pulse. This can help limiting the analysis to a certain part of the movie, e.g. to avoid analyzing late time points with very noisy data that seems to affect the results.

max_n_data_points – takes an integer (default 30). Limits the number of data points to-be-analyzed, usually to keep the results easily presentable. When the value of this parameter is smaller than the number of time points in the movie (which is usually the case), the points are uniformly distributed on a log scale. This choice provides more sampling at short time scales, where the signal expands faster than at long time scales.

Background corrections

estimate_baseline_from – takes either 'beginning' or 'end' (default 'beginning'). Determines whether PIPE will estimate the baseline fluorescence from the beginning or end of

the movie. Baseline fluorescence is any fluorescence that has the same wavelength as the signal but is not a part of the signal. Examples include autofluorescence or pre-existing photo-converted fluorophores. PIPE subtracts the baseline from the signal. It is usually best to estimate the baseline from the beginning, before the photo-conversion pulse is applied, but only in case the fluorescence field is static. If the fluorescence changes much (due to e.g. motion of bright fluorescent objects through the field of view), it might be better to change this parameter to 'end'.

baseline_estimation_window – takes an integer (default 100). Determines how many frames will be used for estimating the baseline. In principle, more frames lead to a more accurate estimate, but if the fluorescence changes fast (e.g. due to photo-bleaching), fewer frames may do better. If PIPE estimates the baseline from the beginning of the movie, it automatically stops the estimation when the photo-conversion pulse begins. (For example, if the pulse starts at frame 80, PIPE will estimate the baseline based on 79 frames even if the value of *baseline_estimation_window* is 100.)

correct_baseline – takes either 0 or 1 (default 1). Determines whether the baseline will be subtracted from the signal. See previous two parameters for explanation on the baseline. In some cases you might not want the baseline to be corrected, for example if the fluorescence changes a lot both in the beginning and at the end of the movie, leading to an incorrect estimate that skews the results.

smoothing_window – takes an integer (default 25). Determines the amount of smoothing applied to the data when estimating the baseline. After averaging the baseline images, PIPE uses a 2D Gaussian kernel to smooth out the averaged image before subtracting the result from the rest of the data. The width of the kernel in pixels is determined by this argument. You may want to use smaller values if the baseline contains sharp and short-scale spatial fluctuations that do not seem to be aptly corrected by the default analysis.

correct_immobile_fraction – takes either 0 or 1 (default 1). Determines whether to automatically correct for an immobile fraction of the fluorophores, if such a fraction is detected. Correction involves subtracting from the signal the average fluorescence field at the end of the analyzed part of the movie.

Manual processing

movie – takes a 3D matrix of integers. This special parameter allows an alternative way to read the data. Use it instead of the parameter ‘path’. You may read the data in any way you choose, and feed it to PIPE in the form of a 3D matrix, where the third dimension must be time. For example, you may read the data using the function *PIPE_read_2d_movie*, which is included in the code package. You can use this function to read only some of the image files in the data folder (since by default PIPE reads all of them). This function returns the data in the right form, so you can feed it to PIPE directly, for example:

data=PIPE_read_2d_movie(<myfolder>,100,400,0); %reads .tif files from <myfolder>, starting from index 100 and until index 400, from channel 0.

*PIPE('movie',data) %runs PIPE on the 3D matrix **data** that was created above*

name – takes a string (default empty). This provides PIPE with a name to write as the title of the plots. When reading the data using the ‘path’ parameter, the name is automatically taken as the last two folders in the path.

pulse_coords – takes a list of four integers (default empty). Determines the coordinates of the center of the photo-conversion pulse in the format (t0,t_start,y0,x0). t0, y0 and x0 denote the temporal and spatial coordinates of the center of the pulse at the last time point where it is still on. t_start denotes the time point in which the pulse starts. When this parameter is empty, PIPE calculates these parameters automatically using image processing techniques.

channel_number – takes an integer (default 1). Determines the channel number the analyzed files need to contain in their name after the letter C (see *Input data format* above). Notice that in some versions of Nikon’s software NIS Elements the first channel in the file names is 0 by default, while in other versions it is 1. For PIPE, it does not matter what the number is, as long as it matches the value of this argument.

direction – takes an integer (default 0). Specifies the angle of the intensity profile relative to the horizontal axis that passes through the center of the initial photo-converted fluorescence distribution. While the analysis can work with any value of this parameter between -90 and 90, we mainly worked with the horizontal direction, because the profiles in this direction tend to contain smaller spatial fluctuations than in other directions. This is probably because the

horizontal direction is the direction our microscope scans the sample, so spatially adjacent pixels in this direction are also adjacent in time, and so less motion takes place between their acquisition times. If you do change this parameter, make sure to adjust the units of the diffusion coefficient by multiplying the pixel size by the cosine of this angle, if it is between -45 and 45, or by the sine if it is between -90 and -45 or between 45 and 90.

thickness_of_profile – takes an integer (default 30). Specifies the number of adjacent lines that are averaged over when segmenting the 1D profile from a 2D image. The lines are averaged pixel-by-pixel in order to decrease measurement noise. We did not observe a significant difference in performance for values between 10 and 30. Values above 30 should decrease noise even further, but values that are significantly bigger than the initial width of the photo-converted fluorescence distribution may lead to a distortion of the Gaussian shapes and to incorrect results.

time_points – takes a vector of integers (default empty). Specifies which time points will be analyzed, starting from the last frame of the photo-conversion pulse as 0. Please note that specifying the time points also determines the time averaging scheme, as all the profiles between two specified time points are averaged together to decrease measurement noise. By default, PIPE automatically chooses the time points based on optimizing the quality of the Gaussian fits. However, in some cases PIPE might make a suboptimal choice, for example by averaging too many time points together, distorting the Gaussian shapes. You can see which time points were automatically chosen by looking at the field `time_points` in PIPE's output (see above). In case you just want to cap the maximal time point to-be-analyzed, use the argument *max_time_point* (see above).

exclude_extra_cellular – takes either 0 or 1 (default 1). Determines whether to exclude the areas outside the cell from the analysis by trimming the intensity profiles, in case the field of view contains a cell. PIPE automatically detects if the whole field of view is roughly uniformly fluorescent and then does not trim the profiles. However, if PIPE incorrectly excludes some parts of the images from the analysis, changes this argument to 0.

Units

pixel_size – takes a double (default 0). Determines the edge size of a single pixel in μm . 0 means size is undetermined.

imaging_rate – takes a double (default 0). Determines the imaging rate that characterizes that data in frames per second. 0 means rate is undetermined.

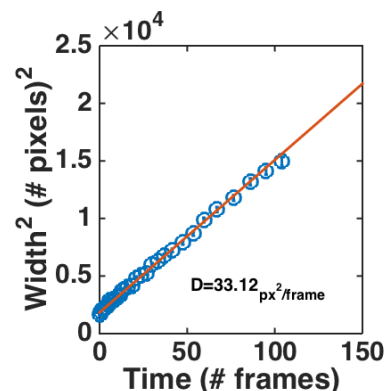
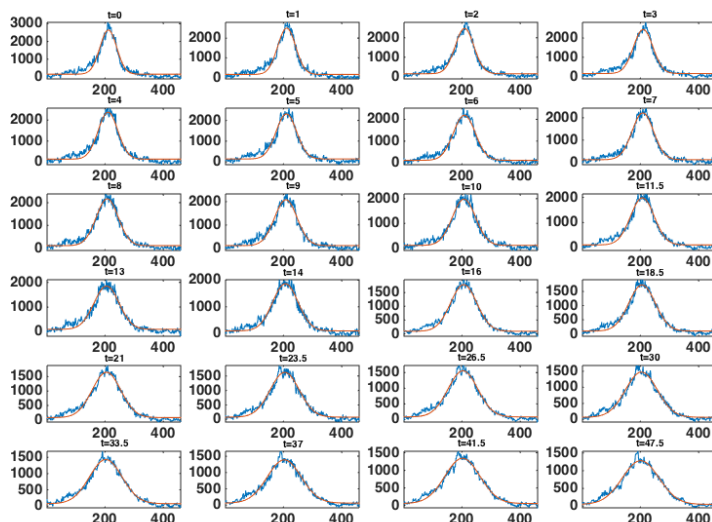
When both `pixel_size` and `imaging_rate` are finite, the output diffusion coefficient is presented in units of $\mu\text{m}^2/\text{sec}$. Otherwise, the diffusion coefficient is presented in units of $\text{pixels}^2/\text{frame}$.

Quality assessment

PIPE's graphical reports provide the user with tools to assess the quality of the output. Following are a few examples for common problems that affect the output. We are using a 2D representation of the Gaussian fits by choosing the value 0 for the argument `plot_Gaussians_3D`.

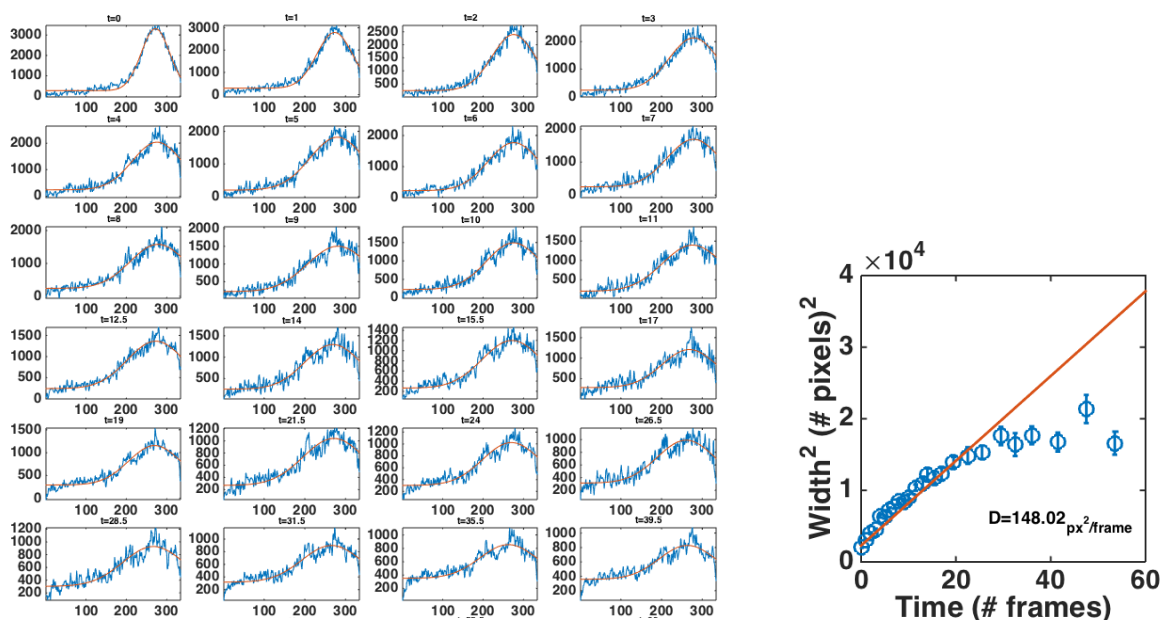
Good signal

Before showing low quality output, following is an example for how a high quality output looks like. Notice the flat tails of the profiles and the way the fits match the raw data. Also notice how the linear fit passes through all of the square width data points on the second plot.



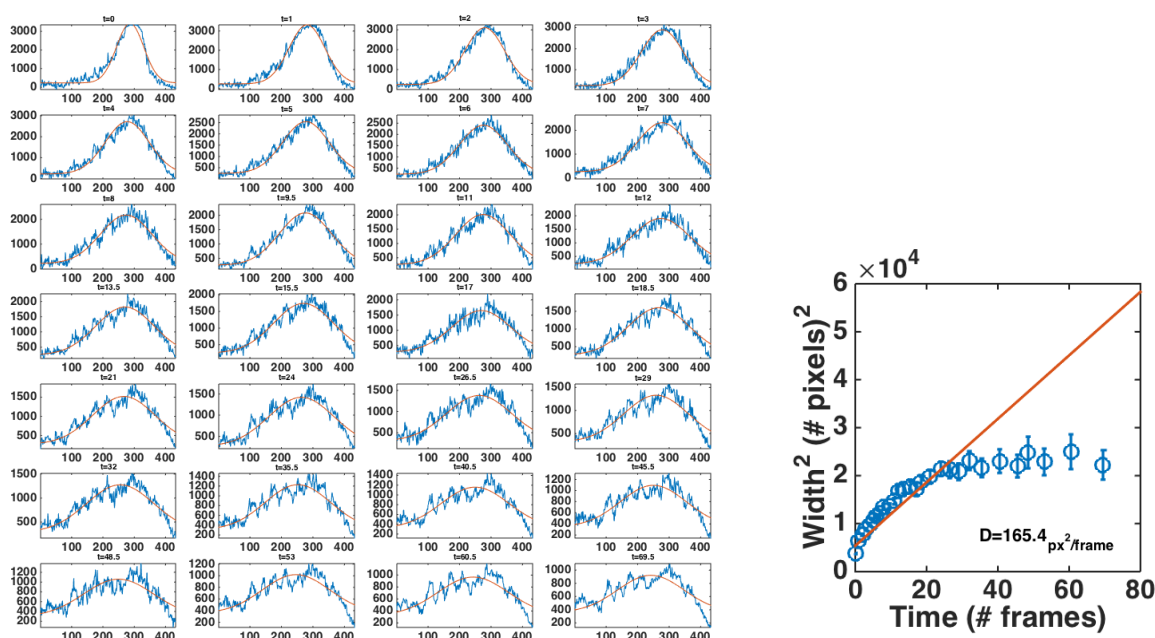
Hitting cell boundary

In this example, the photo-converted fluorophores start close to the cell boundary, and the MSD does not grow linearly with time. While the Gaussian fits seem good, notice how the profiles are asymmetric and are much shorter than the full frame length (512 pixels in our imaging system), which means the rest was probably excluded as being outside the cell. Also notice the flattening out of the square width data points compared to the linear fit in the second plot.



Signal saturation and small cell

In this example we imaged a small cell and chose a high value for the signal amplification (or gain) done by the detector. Notice the flat peak of the Gaussians at short times, indicating the saturation in the signal due to too high amplification. Notice also the low quality Gaussian fits at long times. The steep decrease of the right tail of the profiles, especially evident at long times in this example, is indicative of the fluorescence at the cell boundary. Like in the example above, hitting the cell boundary stops the linear temporal scaling of the MSD, making all the time points from then onward unanalyzable. Lastly, notice the sublinear scaling of the square widths in the plot on the right.



Source file description

Executables

PIPE.m -- main program. Input data either as a folder path or as a matlab matrix read by PIPE_read_2d_movie.m. Run from within Matlab: see examples in file header.

PIPE_read_2d_movie.m -- create a data matrix from raw data files. By default, users are not required to run this script, but they may do so for more advanced usage.

Core of analysis

PIPE_calc_gaussian_widths.m -- fitting each intensity profile to a Gaussian and extracting its width.

PIPE_calc_D_of_expanding_gaussians.m -- fit widths vs. time to a normal diffusion model.

Data characterization and pre-processing:

PIPE_identify_cell.m -- distinguish cell from cell exterior in image

PIPE_find_min_distance_to_boundary.m -- calculate distance from photo-conversion pulse to cell boundary

PIPE_correct_baseline.m -- subtract fluorescence baseline from signal

PIPE_find_phoc_peak.m -- find maximum of noisy time series

PIPE_find_pulse_center.m -- find center of 2D peak

PIPE_detect_pulse_end.m -- detect the last frame where the photo-conversion pulse is still on

PIPE_define_1d_profile_mask.m -- define a straight line passing through a given point within a 2D image

PIPE_apply_profile_mask.m -- get intensity profile from a 2D image

PIPE_estimate_width0.m -- estimate the width of the initial photo-converted protein ensemble

PIPE_calc_time_points_by_noise_analysis.m -- determine optimal averaging scheme to reduce noise

PIPE_reduce_n_time_points.m -- reduce number of analyzed time points to a user-provided number

PIPE_correct_immobile_fraction.m -- subtract late-times intensity profile from rest of signal

PIPE_smooth_with_gaussian_kernel.m -- customized 1D or 2D smoothing procedure