Automated Bayesian Pipeline for Rapid Analysis of Single-Molecule Binding Data

Version 1.1

USER MANUAL

Table of Contents

INTRODUCTION

The pipeline was designed for rapid processing of co-localization single-molecule spectroscopy (CoSMoS) images and quantitatively assessing experimental data quality. The software requires MatLab, DIPImage and CUDA, as well as a computer that meets the requirements as outlined in the next section.

SYSTEM REQUIREMENTS

To function properly, the pipeline requires a computer that meets the following specifications:

- A CUDA-enabled graphics card
- Any Intel or AMD x86-64 processor (min. 4 cores recommended)
- 4 GB of RAM or more
- 4-8 GB of space for typical installation

The performance of the pipeline depends critically on the dataset size and the available computational power. For details on the hardware configurations that have been tested, refer to the Table 1.

INSTALLATION

All the software packages required for the pipeline support multiple operating systems. As a result, the pipeline should in principle work on Windows, Linux and MacOS. This manual currently includes installation instructions only for Windows. Adventurous users are however encouraged to test the pipeline on their operating system of choice.

System configuration				the pipeline performance	
CPU	**Memory**	**GPU**	**OS/MatLab/DIP**	**Version**	**Runtime**
Intel Xeon E5 8 cores@2.59GHz	DDR3 64GB@2400MHz	Quadro K2200 36GB@80GB/s	Windows 10 prof. R2017b / DIP 2.8.1	1.0	454 s

Table 1. Specifications of tested systems.

Windows installation

1. Clone repository or download software from https://github.com/quantitativenanoscopy/cosmos_pipeline

2. Install MatLab using the installation materials and procedures provided by your institution or acquired with the MatLab license.

3. Install DIPimage for MatLab

    a. Download the latest version from http://www.diplib.org/download. To maximize stability, make sure that the version you download has been tested with the MatLab version you use.

    b. Install DIPimage for MatLab (refer to DIPimage User Manual for assistance)

    c. Add the following lines to the MatLab startup file `startup.m` (usually found in Documents\MATLAB; create file if it does not exist yet):

    ```
    run('C:\Program Files\DIPimage\dip_initialize');
    ```

    Note: if DIPimage was not installed in the default location, the line above needs to be changed. In some DIPimage distributions, the DIPimage folder in Program Files is given a name that includes the version number (e.g. `DIPimage 2.8.1`).

    d. Confirm that DIPimage was installed successfully by executing the command `dipimage` in MatLab. Refer to the *DIPimage User Manual* for assistance with any unresolved errors.

4. Install CUDA

    a. Download the latest version of CUDA from https://developer.nvidia.com/cuda-toolkit

b. Run the installation as administrator and

   i. Choose *standard* or *express* installation;

   ii. Acknowledge the warnings regarding Visual Studio (not needed to run the image processing pipeline).

c. Restart the computer

5. Adjust GPU time-out settings using either of the following ways:

   a. Add the keys in keyfile (found in installation aids folder) to the windows registry

   b. Manually add or change the two following registry keys using *regedit*:

      i. Key 1
      KeyPath :
      HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\GraphicsDrivers
      KeyValue : TdrLevel
      ValueType : REG_DWORD
      ValueData : 0
      TdrLevelOff (0) - Detection disabled

      ii. Key 2
      KeyPath :
      HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\GraphicsDrivers
      KeyValue : TdrDelay
      ValueType : REG_DWORD
      ValueData : 1000
      TdrDelay - Detection disabled for number of Value seconds

6. Verify functionality of the pipeline by typing pipeline_interface in the MatLab command line. This command opens a graphical user interface. If everything was successful, you should see *"The execution was successful"* in the command window of MatLab, and you can continue to the next section.

UNDERSTANDING THE DATA

A clear understanding of the various components that compose a complete dataset will help you get the most out of your analysis.

Description of a complete dataset

A complete dataset consists of:

1. a gain calibration dataset,
2. a camera registration dataset,
3. a dataset of the binding experiment.

This section describes each component, using the example dataset available at https://figshare.com/collections/An_Automated_Bayesian_Pipeline_for_Rapid_Analysis_of_Single-Molecule_Binding_Data/4294421/1.

Gain calibration data

The gain calibration dataset, used to estimate the gain of Electron Multiplying Charge Coupled Device (EMCCD) cameras, consists of 2 files: Dark.tiff and Grid.tiff. The datasets, 100 frames each, are recorded in widefield mode (cell^TIRF – Widefield) with the same EM gain to be use for the binding experiment. Dark.tiff is recorded with no light collection (Metamorph – Illumination = none) (Fig. 1a). Grid.tiff is recorded from a grid slide (Metamorph – Illumination = Transmitted widefield) (Fig. 1b). The grid provides a convenient way to generate a range of pixel-intensities within a single image; other diffraction patterns or imaging standards may provide equally suitable gain calibration data.

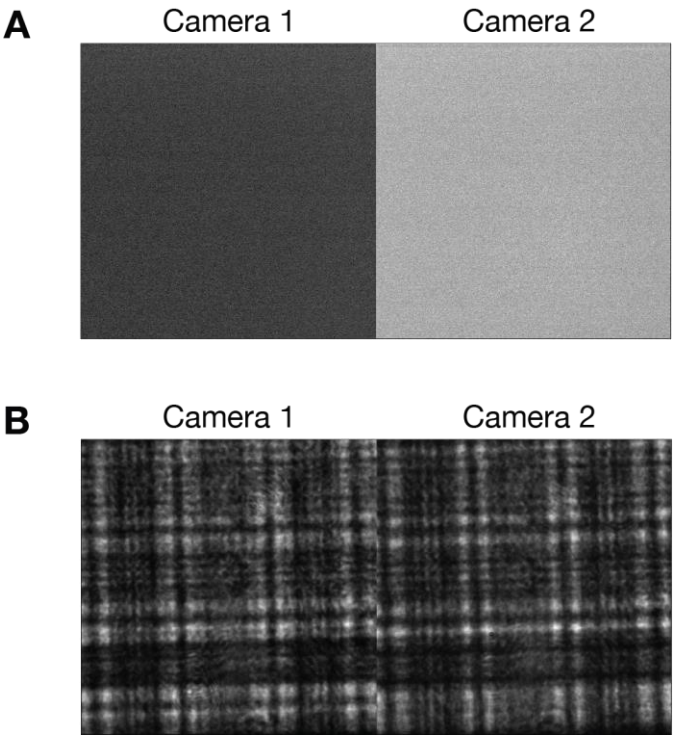


Figure 1. The gain calibration dataset. The first frame of each Dark.tiff (A) and Grid.tiff (B) is shown as an example.

Procedure

For optimal detection and localization, measurements from the camera need to be converted from digital units contained in the image to number of photons recorded by the camera using Poisson noise characteristics [1]. This conversion is accomplished by adjusting the estimated gain and pixel offset [2].

Camera registration data

The camera registration dataset, Beads.tiff, is used to determine the alignment of images from different cameras. A set of 10 images is recorded from a bead slide in TIRF mode (cell^TIRF – Critical angle) (Fig. 2). Note: for accurate alignment, the diameter of the beads used for camera registration should not exceed the diffraction-limited resolution.

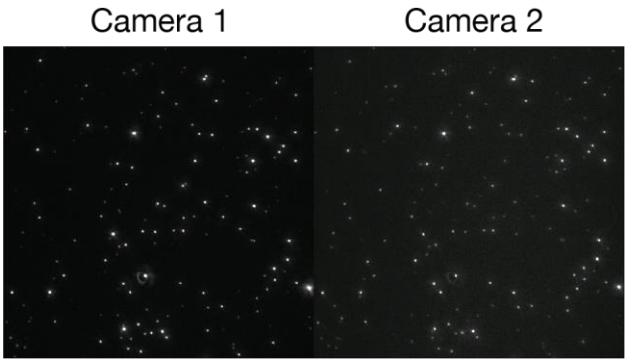


Figure 2. The camera registration dataset. The first frame of Beads.tiff is shown.

Experimental Data

In the example, the binding experiment measures the kinetics of TtAgo:guide complex binding a target DNA containing two binding sites. The target was immobilized on the slide via a biotin-streptavidin interaction. Continuous acquisition of frames was started when the TtAgo:guide solution was flowed-in. In this experiment, 1,500 frames were collected at 5 frames/s. Images were recorded as uncompressed TIFF files and merged into a stacked TIFF file: Experiment.tiff.

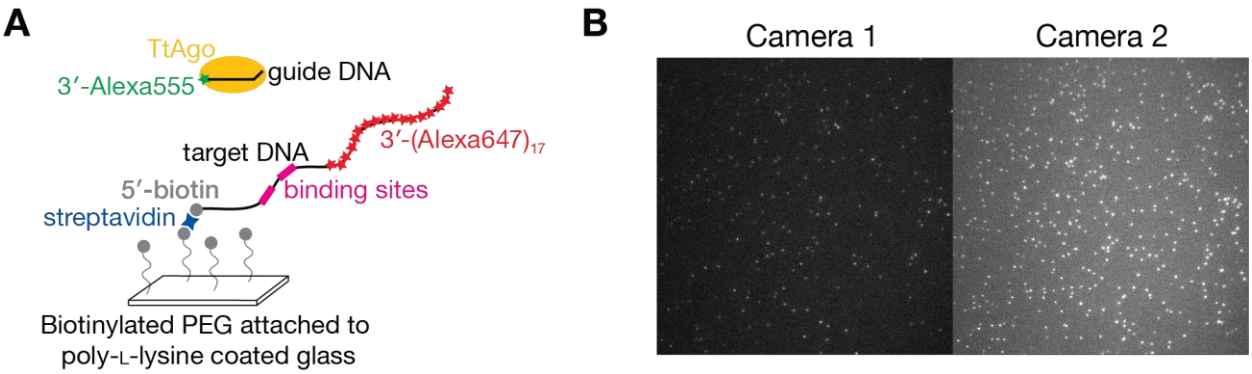


Figure 3. Example binding experiment. (A) Experimental setup. (B) TtAgo:guide complex (camera 1) binds a DNA target containing two binding sites (camera 2). Frame 567 is shown as an example.

Note: In the example experiment TtAgo binds on a target with long dwell times, therefore imaging speed of 5 frames/s was fast enough to capture these binding events. If protein of user's interest displays faster dissociation rates, the user needs to decrease the exposure time, which might result in lower fluorescent signal. An example of such lower-signal dataset is presented in Fig. 45b.

DATA PROCESSING AND ANALYSIS

Here we will walk you through the steps to analyze the example dataset:

1. Data pre-processing:
    a. gain calibration,
    b. alignment of cameras,
    c. drift correction.
2. Signal detection and localization:
    a. identification of target locations and of the binding complexes,
    b. co-localization of the diffusible molecules at each immobilized target.
3. Data analysis:
    a. calculation of association and dissociation rates,
    b. correction for non-specific binding of the mobile component to the slide,
    c. detection of multiple binding sites.

Note: To ensure your installation was successful, execute the pipeline by typing pipeline_interface in the MatLab command line. You should see *"The execution was successful"* in the command window of MatLab.

The pipeline Interface

Type pipeline_interface in the MatLab command line. This command opens a graphical user interface. If you do not see this window appear, revisit the installation instructions and check the command window for error messages.

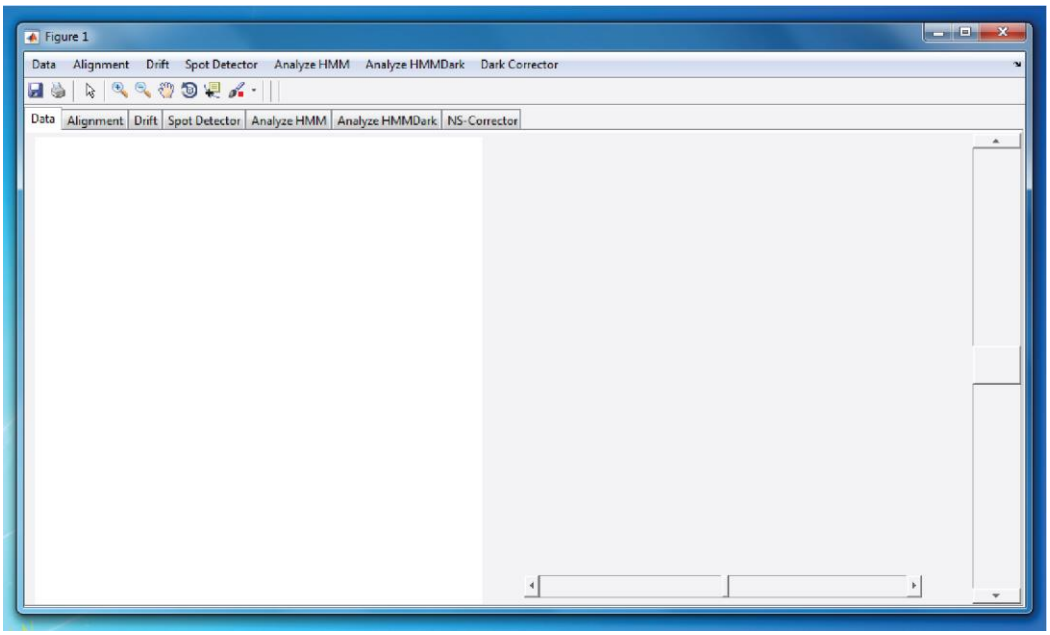


Figure 4. The Graphical User Interface (GUI) before loading data.

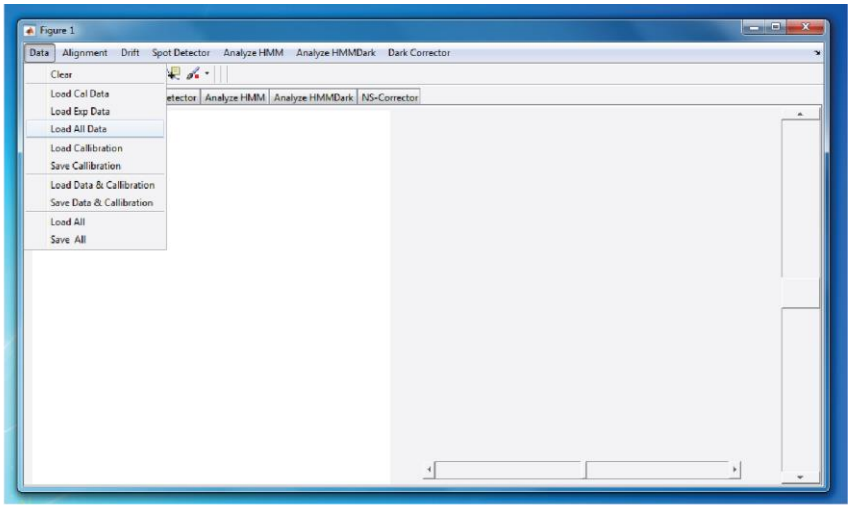
Loading data

To load the complete dataset, go to: Data —> load all data (Fig. 5a), select the correct files and parameters of data acquisition (Fig. 5b) and press OK.

Parameters that you are asked to provide:

- Intensity Range: values of pixel intensities used for camera calibration. By default, all values, i.e. [0 Inf], are used. To change the values refer to Troubleshooting – Camera Calibration; Fig. 32; values should be provided in [min max] format.
- Left wavelength and Right Wavelength: wavelength (in nm) of the left and right camera, as displayed on the computer screen). These values define the color code for displaying fluorescence intensity time traces of target molecules and mobile components.
- Frames per second: frame rate used for data acquisition. This parameter defines the time scale for plotting fluorescence intensity time traces and rastergrams and for calculating kinetic parameters.
- Number of frames to group: number of frames to sum in order to computationally increase the exposure time.
- FlipCams: if unchecked (the default), the pipeline assumes that target molecules are detected by the right camera; when checked, the left camera is presumed to detect the target molecules.

A



B

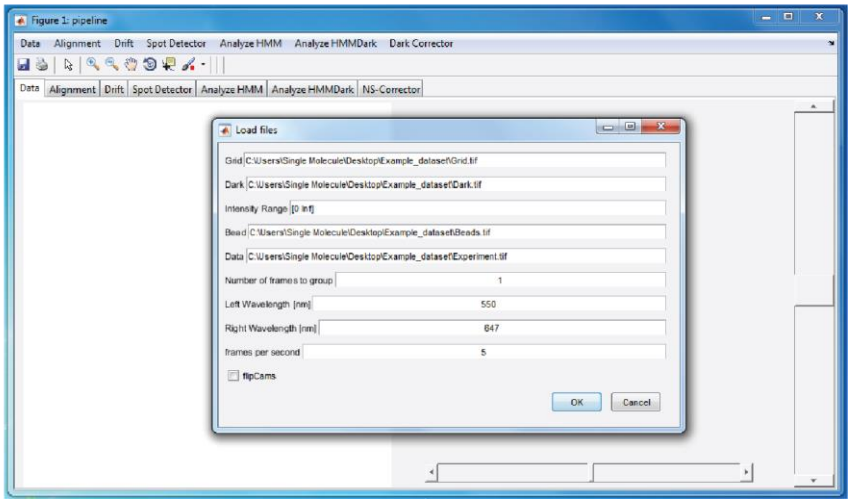


Figure 5. Loading a complete dataset. (A) Menu item "Load all data" is selected. (B) The correct files and settings are depicted.

Preprocessing steps

The first module, *preprocessing*, consists of Electron Multiplying Charge Coupled Device (EMCCD) camera gain calibration, multicamera alignment, and drift correction.

Gain calibration

The pipeline automatically estimates the gain and offset of the cameras, which determine the conversion between the number of photons recorded by the camera and the number of digital units contained in the image. If this process is successful, the calibration curves (Fig. 6) are linear. If this is not the case, check Troubleshooting – Camera calibration section. The gain and offset values are displayed (Fig. 6) and automatically stored by the pipeline.

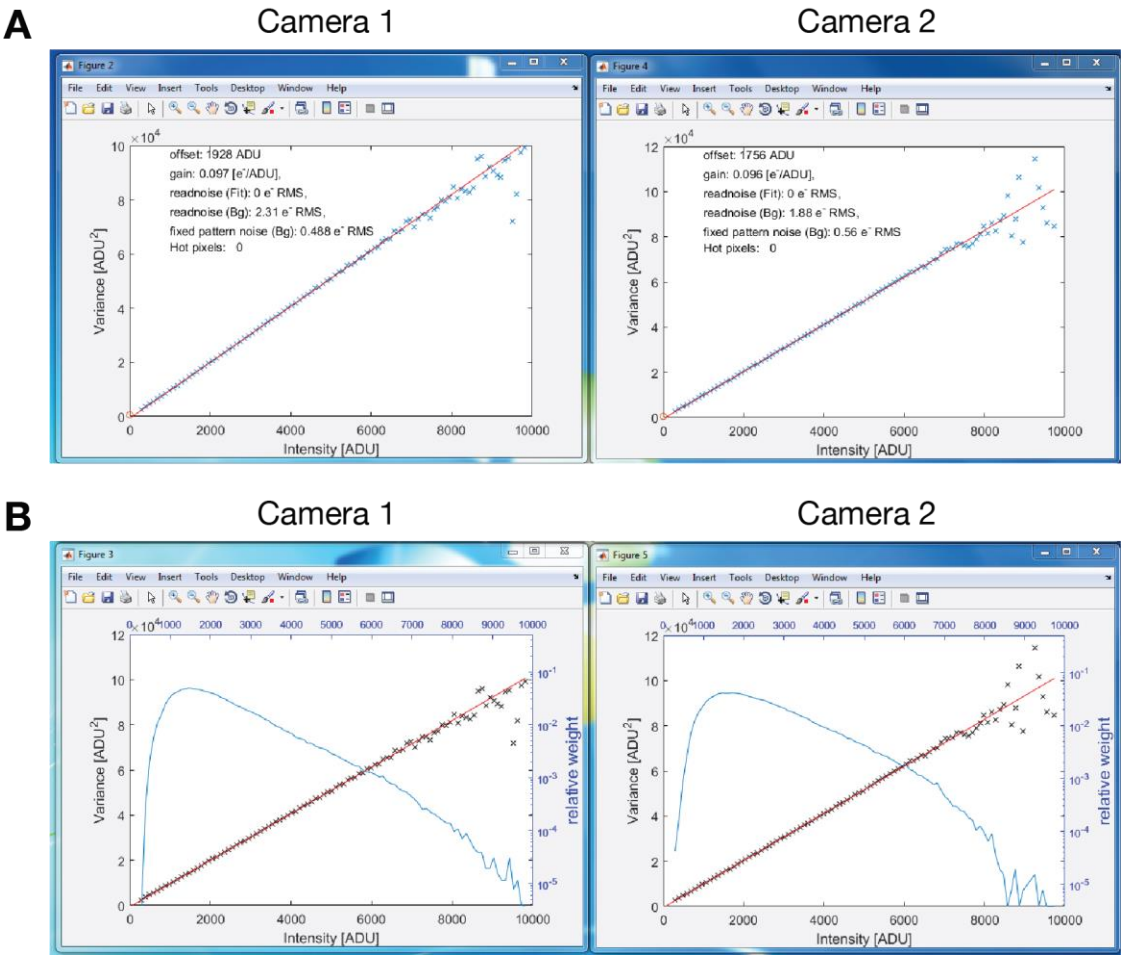


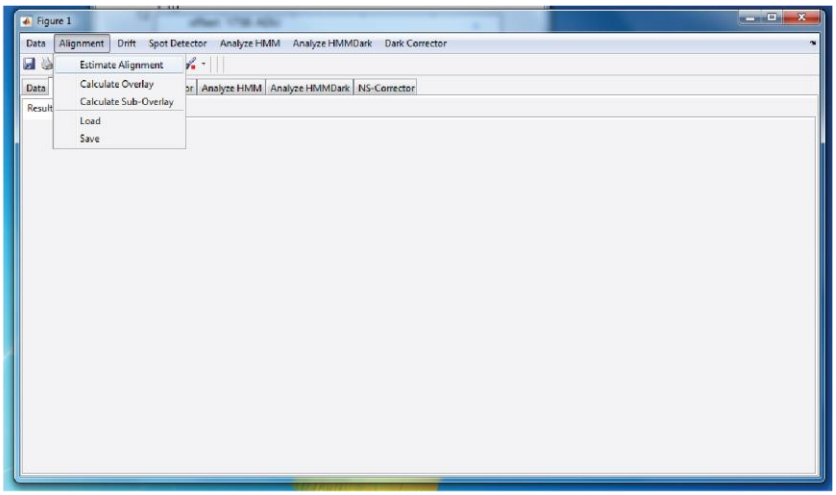
Figure 6. Camera calibration. (A) The pixel variance is plotted as function of pixel intensity for each camera. The gain (the slope of the calibration curve) and the offset of each camera are displayed. (B) The same calibration curves as in (A), displaying number of events for each intensity value (plotted in blue).

Alignment of the cameras

The second step aligns the fields of view from the cameras used detecting to the different fluorophores in the experiment: Alignment —> Estimate alignment (Fig. 7a).

The mapping is obtained by the estimation of an affine transformation from images of fluorescent beads that are detectable by both cameras. When the process has been completed, a superimposed image is displayed, allowing visual inspection of any remaining offset between the wavelength cameras (Fig. 7b, right panel). The same image is also displayed with a color code indicating an alignment score for each bead (Fig. 7b, left panel). The alignment score corresponds to the experimental mean square error, which measures the difference between the estimate of aligned positions and positions actually observed. Values closer to zero indicate better alignment.

A



B

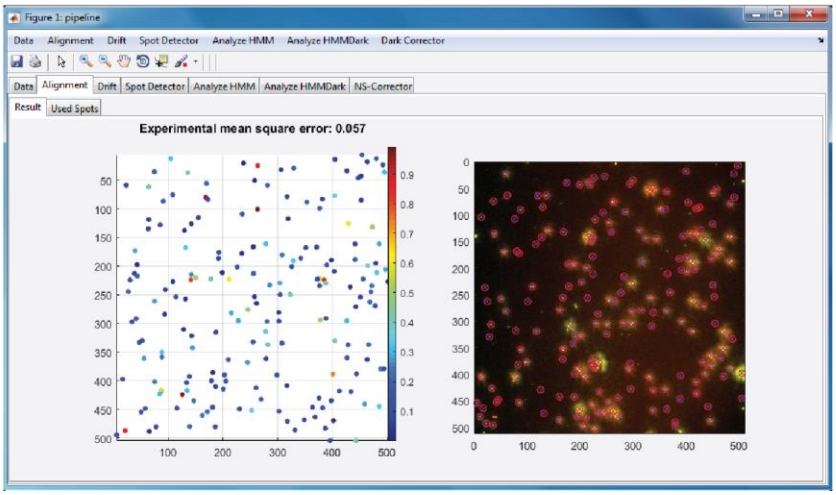


Figure 7. Alignment of images from different cameras. (A) Menu item "Estimate alignment" is selected. (B) Result of the initial alignment. Right panel displays a superimposed image of beads from both cameras. Left panel shows the alignment score for each bead. Blue, low score; red, high score.

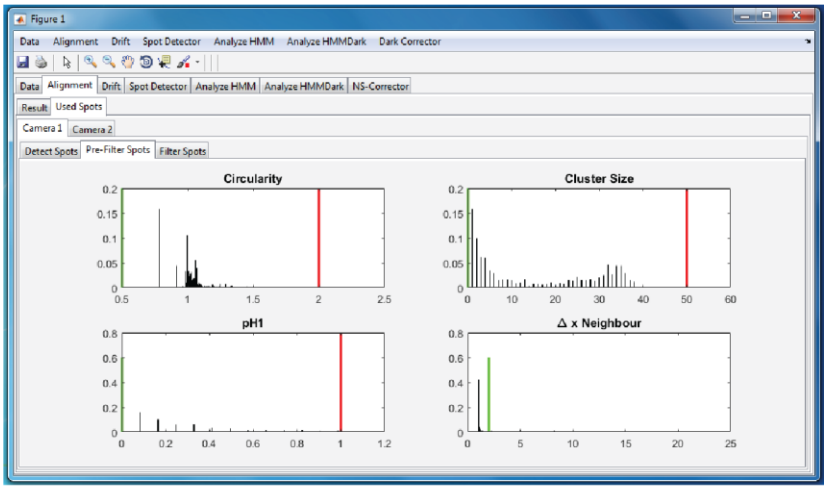
Refine alignment

Some beads may bias the alignment and need to be excluded. For example, because accurate alignment requires that the bead diameter does not exceed the diffraction-limited resolution, clusters of beads are excluded.

Removal of undesired beads can be achieved by changing the lower and upper thresholds of key parameters (Fig. 8, green and red lines, respectively) in Alignment —> Used spots —> Camera 1 or 2 tabs. The key parameters include:

- circularity, defined by ratio of two perpendicular diameters,
- cluster size,
- distance to a neighboring bead,
- PH1, probability of true detection,
- PFA, probability of false detection.

A



B

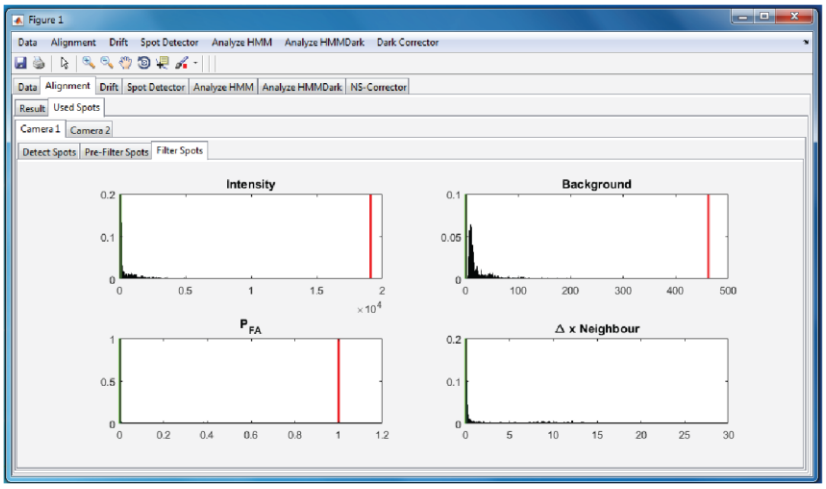


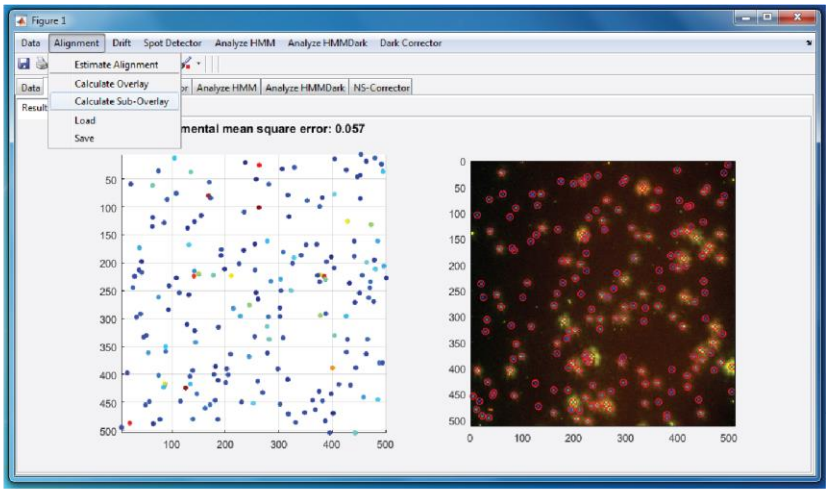
Figure 8. Filtering the beads used for the alignment. (A) Beads can be pre-filtered according to circularity, cluster size, probability of true detection (PH1), and distance to a neighboring bead. (B) Beads can be further filtered based by intensity of bead signal and of the background, as well as probability of false detection (PFA).

Quality check

When successfully aligned, **beads in the superimposed image appear yellow**. A lower alignment score indicates that the transformation is more precise between the two cameras (mean square error in unit pixels). Finally, the degree of alignment of both cameras can be visualized in experimental data by creating an overlay (or sub-overlay, if only a sub-region is considered) sequence of images (Fig. 9), which can be saved in TIF format (move to the last frame (frame = 1,500 in the example) —> right click —> save as TIFF) and visualized later by importing in Fiji as Bio-Formats.

Note: the current version of Fiji does not read properties correctly. To fix this: 1) load the file (File —> Import —> as Bio-Formats), 2) Image —> Properties —> Channels (c) = 3 and Frames (t) = displayed number divided by 3 (in the example 4500/3 = 1500), and 3) Image —> Color —> Make composite.

A



B

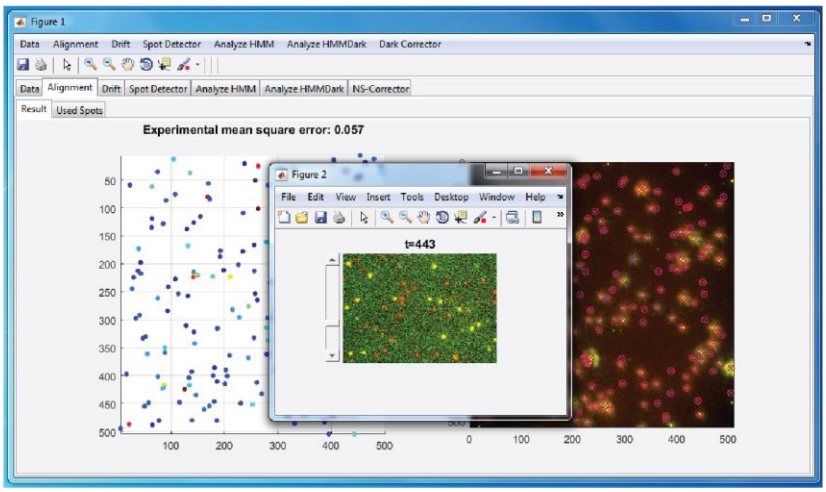


Figure 9. Creating an overlay movie for the experimental dataset. (A) Menu item "Calculate Sub-Overlay" is selected. A window corresponding to the camera of the mobile component (here TtAgo:guide complex) will appear. The user is invited to select a sub region. Note that the contrast is adjusted using the linear contrast stretch, i.e., a simple enhancement technique, which improves the contrast in an image by stretching the range of original intensity values to a desired range of values. The user can navigate through different frames of the data by pressing N for next frame and P for previous frame. (B) An overlay of the two cameras is created for the experimental dataset. For example, shown here, Frame 743 shows both unbound DNA target molecules (red) and DNA target molecules bound to the TtAgo:guide complex (yellow).

If the alignment does not pass quality control, check Troubleshooting – Camera alignment.

Procedure

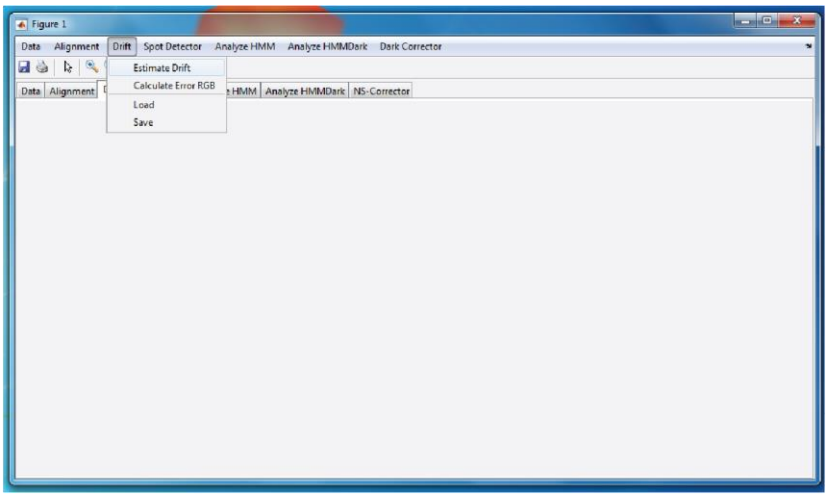
In the above steps the pipeline performed a set of calculations for the alignment. Assumed is that each object is characterized by a set of points $(\mathrm{P} = (\mathrm{p}_1, .. \mathrm{p}_\mathrm{n}))$. These points may correspond to, e.g., a descriptor or the center of mass of a bead that are obtained at high signal-to-noise ratios ($\sigma_{\mathrm{loc}} \approx 0$). These points are viewed from multiple different perspectives due to imperfections in camera alignment and/or optics $(\mathrm{P}^1, \ldots, \mathrm{P}^\mathrm{n})$, which results in a difference in rotation, scaling, shear and translation [3]. Under this assumption the relation between the different perspectives can be expressed though an affine transformation $\mathrm{P}^\mathrm{j} = \mathrm{R}_{\mathrm{i,j}}\mathrm{P}^\mathrm{i} + \mathrm{t}_{\mathrm{i,j}}$, where $R_{i,j}$ is a non-singular matrix and $t_{i,j}$ is a vector determining the transformation from perspective P^i to P^j. Furthermore, $R_{i,j}$ determines the rotation, scaling, and shear, and $t_{i,j}$ determines the translation.

Correction for lateral drift

To follow individual target molecules over time, it is necessary to correct for drift caused by experimental manipulations and by movements of the stage (e.g., when introducing a new solution into the sample chamber). In the example experiment, DNA target molecules are immobilized on the glass surface and thus can be used as fiducial markers in each frame. After clicking on the menu item “estimate drift” (Fig. 10a), a window appears (Fig. 10b) that allows the user to select a small area used for drift correction. To achieve an accurate estimation, we suggest selecting an area with several bright and individual spots.

Note: It is also possible to include on the sample chamber surface bright fiducial markers, e.g., fluorescent beads similar to those that are used in our camera alignment procedure. This allows for general experimental designs and in particular enables the analysis of data where signal from target molecules has been lost, e.g., due to photobleaching of single-dye fluorescent molecules or cleavage of the target accompanied by the departure of the labelled portion of the target.

A



B

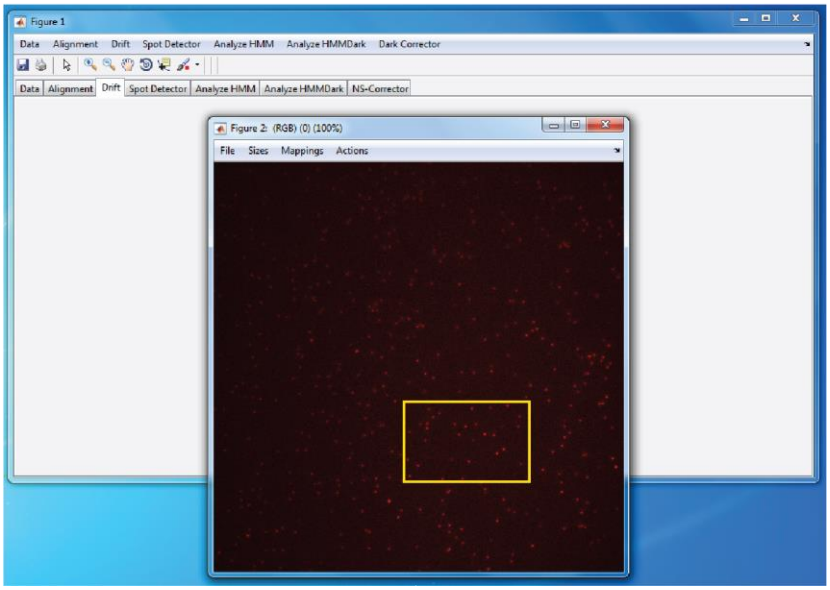
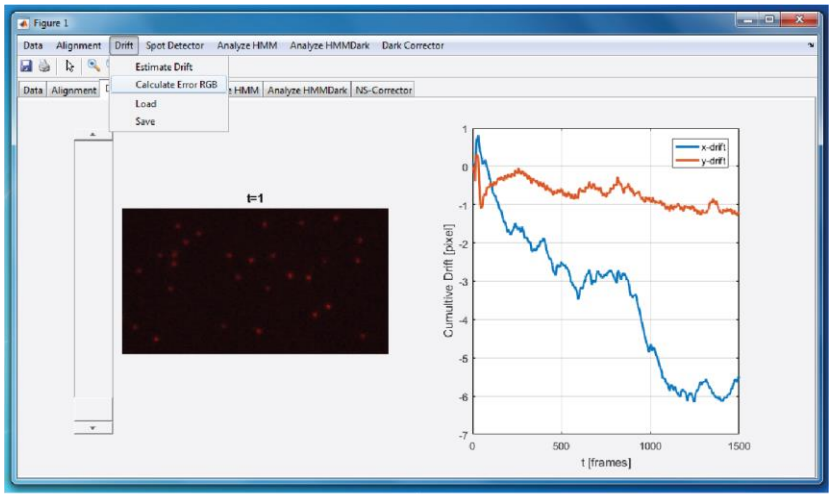


Figure 10. Drift correction. (A) Menu item "Estimate drift" is selected. (B) A window allowing selection of area used in this process (delimited by yellow line) appears.

Quality check

It is important to check **if the drift correction is performed correctly**. First, we suggest calculating the RGB error over a stable intensity region, for example over the region used for drift correction (Fig. 11a). The quality assessment is visualized by plotting frame *t* in red and *t+1* in green. Successful drift correction produces a yellow signal (Fig. 11b). Second, it is crucial to determine whether the selected spots move along with the underlying spots in the analysis tab (after the analysis is performed; see Figure 17). An example when the drift correction was not performed correctly is given in Fig. 39.

A



B

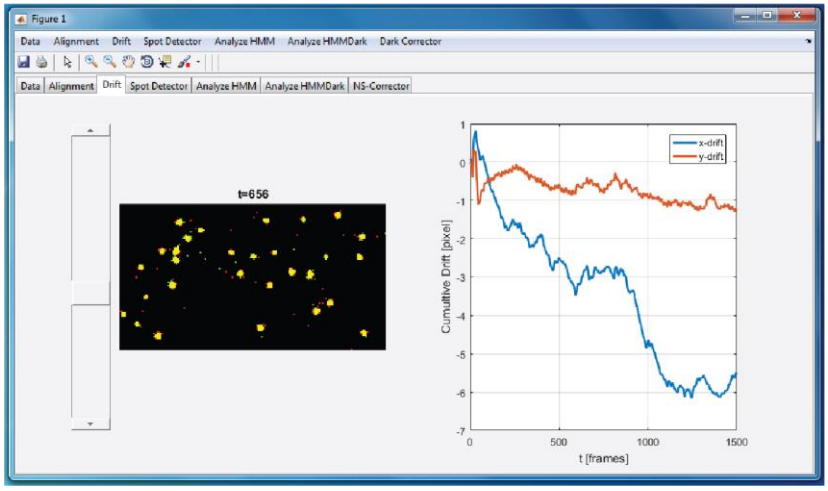


Figure 11. Quality control for drift correction. (A) Menu item "Calculate error RGB" is selected. (B) Successful drift correction from frame *t* to frame *t+1* results in yellow signal.

Procedure

We use the correlation between frames to correct for drift. Therefore, it is important that the selected region contains enough stationary signal over the whole experiment. If this is not the case, the estimation of the drift can be imprecise and biased. The drift is calculated using the target camera as this most often contains the most stationary signal. The integer drift between the sub-region in frame *t* and frame *t+1* is calculated by using the Fourier shift property, i.e., consider two frames $\mathrm{f_t}$ and $\mathrm{f_{t+1}}$, with the Fourier transforms $\mathrm{F_t} = \mathcal{F}(\mathrm{f_t})$ and $\mathrm{F_{t+1}} = \mathcal{F}(\mathrm{f_{t+1}})$, then the integer shift is computed as the position of maximum intensity of the phase correlation: $\{\Delta \mathrm{x_{\delta t}}, \Delta \mathrm{y_{\delta t}}\} = \max_{\mathrm{x,y}} \left[\mathcal{F}^{-1} \left(\frac{\mathrm{F_t o F_{t+1}^*}}{|\mathrm{F_t o F_{t+1}^*}|} \right) \right]$

Where o is the entry wise product, $\mathcal{F}^{-1}(.)$ is the inverse Fourier transformation, $|.|$ is the absolute value and $\Delta \mathrm{x_{\delta t}}, \Delta \mathrm{y_{\delta t}}$ is the estimated integer shift between the two images. To refine this estimate, an iterative gradient-based estimator is used to obtain the registration estimate with minimal bias and maximal precision [4].

Signal detection and localization

Target spot detection

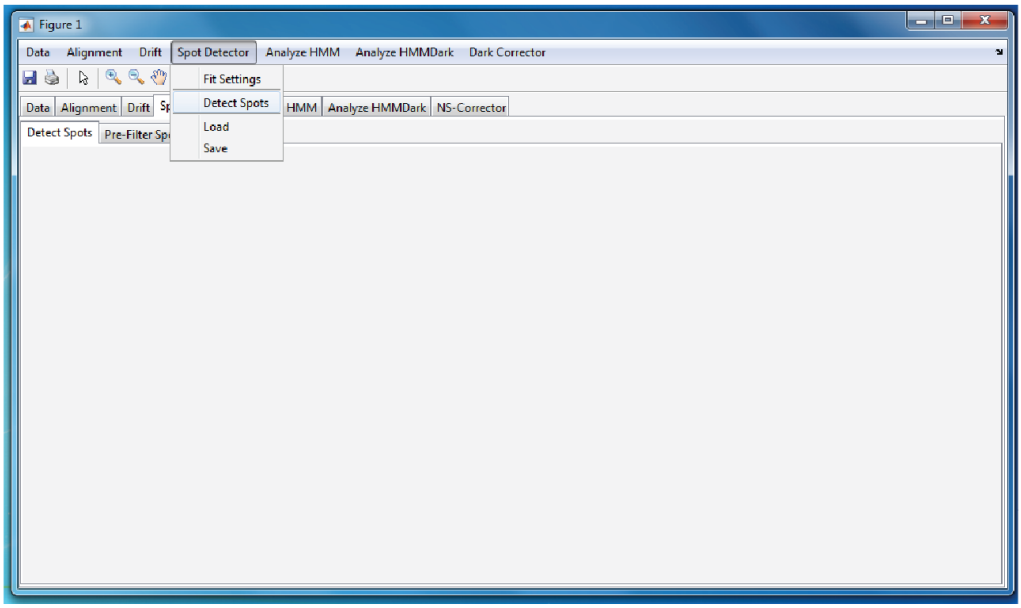
The first step in the analysis of the experimental data identifies the positions of the target molecules. In this step, the pipeline also detects dark locations, which do not contain target molecules and therefore provide a control for non-specific binding of the mobile component to the slide surface later in the analysis (see section "Analyzing the non-specific binding traces and the dark correction").

First, set parameters used to fit the position of the spots (Fig. 12). Next, click on "Detect spots" in Spot Detector tab (Fig. 12a). The result is displayed in Fig. 13. Parameters used to fit positions of the spots include:

- number of iterations: often 10 iterations are enough for convergence; however, this depends on the noisiness of the data,
- maximum number of Cuda Fits, which depends on the capacity of the graphics cards,
- PSF Sigma: width of the Gaussian point spread function of the imaging system,
- box size: the rule of thumb is that the box size should be $3(2\sigma_{\mathrm{PSF}} + 1)$.

Note: To detect spots, the pipeline also uses the same filter parameters to set thresholds for the quality of spots to be included. These parameters were described for detection of beads: circularity, cluster size, PH1, distance to neighbor, intensity, background, and PFA (Figure 8). The default thresholds for these parameters were leniently set in order to accommodate the majority of users. Consequently, some few locations of poor quality might be included. Therefore, we advice to perform a quick quality check by visually inspecting detected spots. The user will have a possibility to filter undesired spots by modifying the filter parameters to set thresholds (Figure 16) or by manually discarding those locations.

A



B

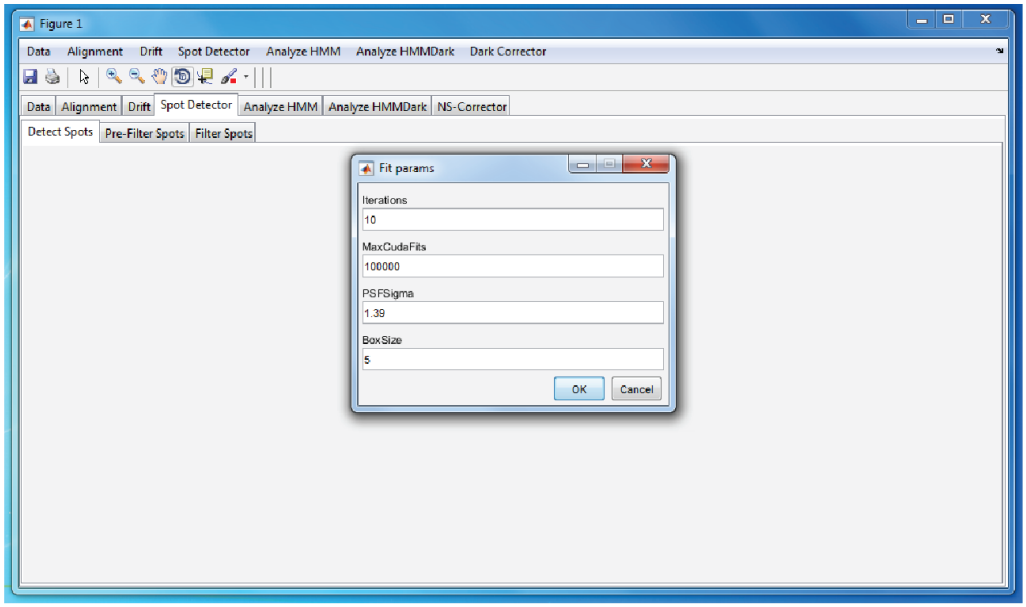


Figure 12. Detection of target and TtAgo locations. (A) Menu item “Spot Detector” allows setting fitting parameters and performs the spot detection. (B) Fitting parameters used in the example dataset.

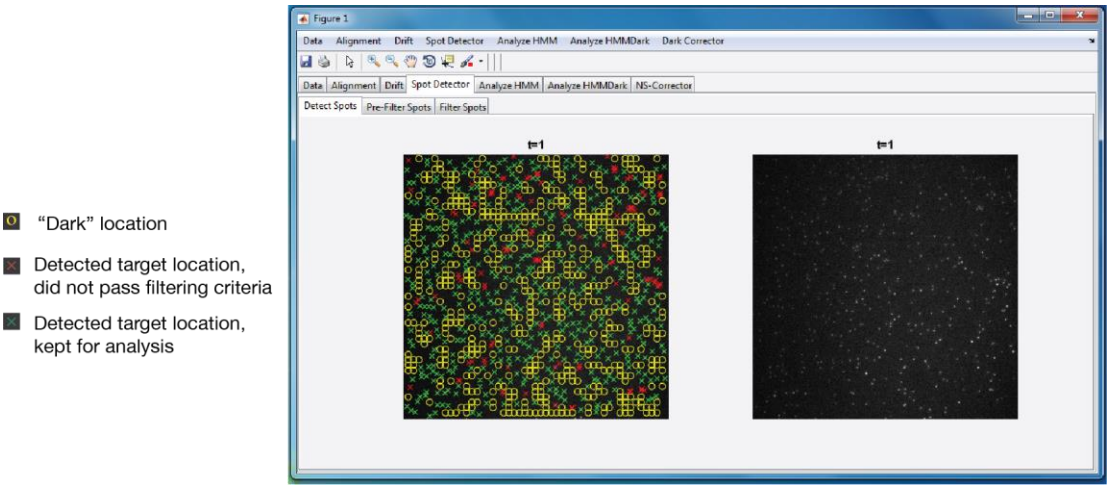


Figure 13. Target and Dark locations detected by the pipeline.

Quality check

It is important to check that the spot detection yielded individual target locations. Using the zoom (magnifying glass) in the tool menu, visually inspect the detected spots (Fig. 14). Some target locations of poor quality can be wrongly detected and potentially bias the analysis. This is the case of multiple clustering spots (Fig. 15a), two spots detected as one (Fig. 15b), and spots of low intensity (Fig. 15c). They can be manually removed by double-clicking on them (circle around the location turns from green to red) (Fig. 14d), or they can be automatically removed by modifying threshold parameters (Fig. 16).

If target locations were not detected successfully, check Troubleshooting – Spot Detection.

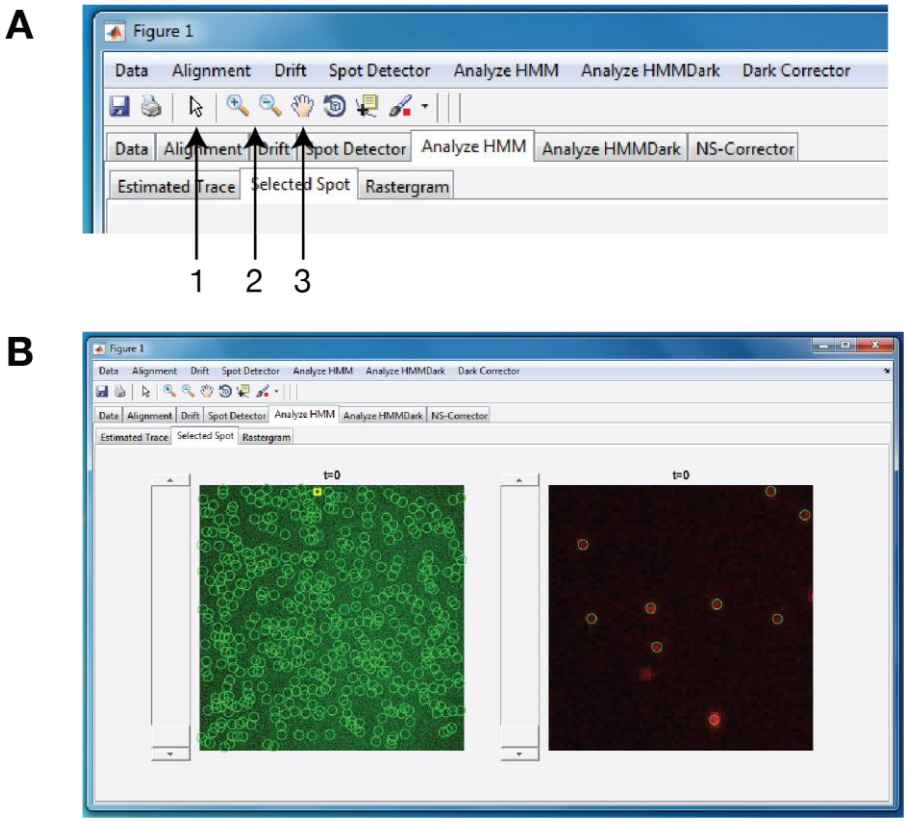


Figure 14. Visual inspection of detected target locations. (A) Selecting (1), zoom (2) and pan (3) tools are provided. (B) Example of a zoomed-in region of interest.

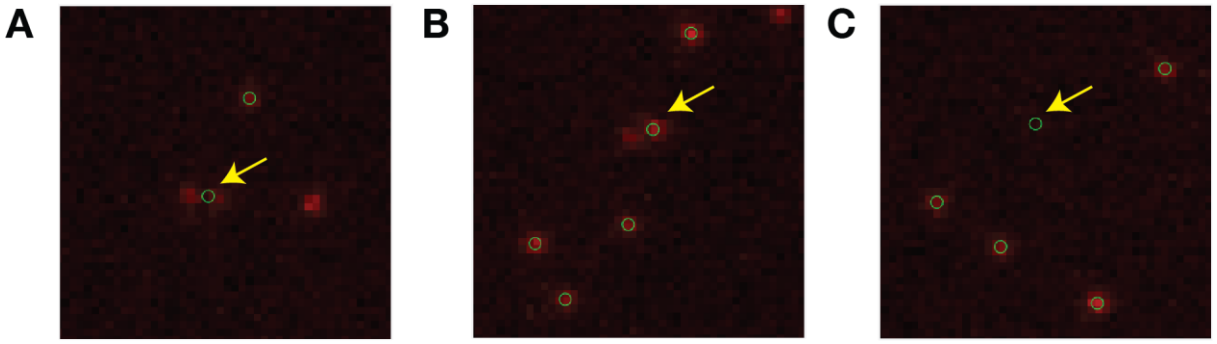
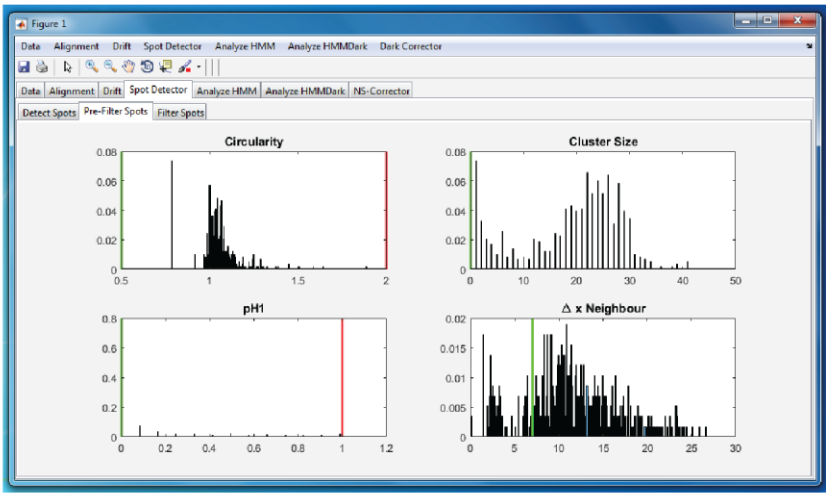


Figure 15. Examples of target locations that need to be excluded. (A) Detected location (green circle) does not correspond to a target molecule. (B) Target location was correctly detected, but may bias the analysis due to close proximity to another target molecule. (C) The pipeline allows full control of your data. Here are spots with low intensity that may correspond to a false positive spots easily excluded by use of automatic filtering.

A



B

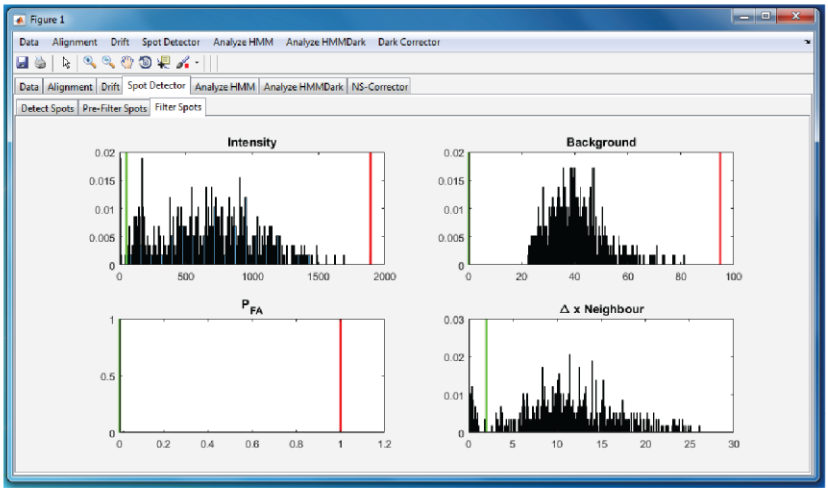


Figure 16. Filtering target locations. (A) Target locations can be pre-filtered by circularity, cluster size, probability of detection (PH1), and distance to a neighboring target molecule. (B) Target locations can be further filtered by the intensity of target molecules and of the background, as well as by probability of false detection (PFA). For more details on filtering parameters see Alignment of cameras – Refine alignment.

Before proceeding to the next step, **verify that drift correction was correctly performed**. To do this, check that target spots are positioned within green circles throughout the entire experiment (Fig. 17).

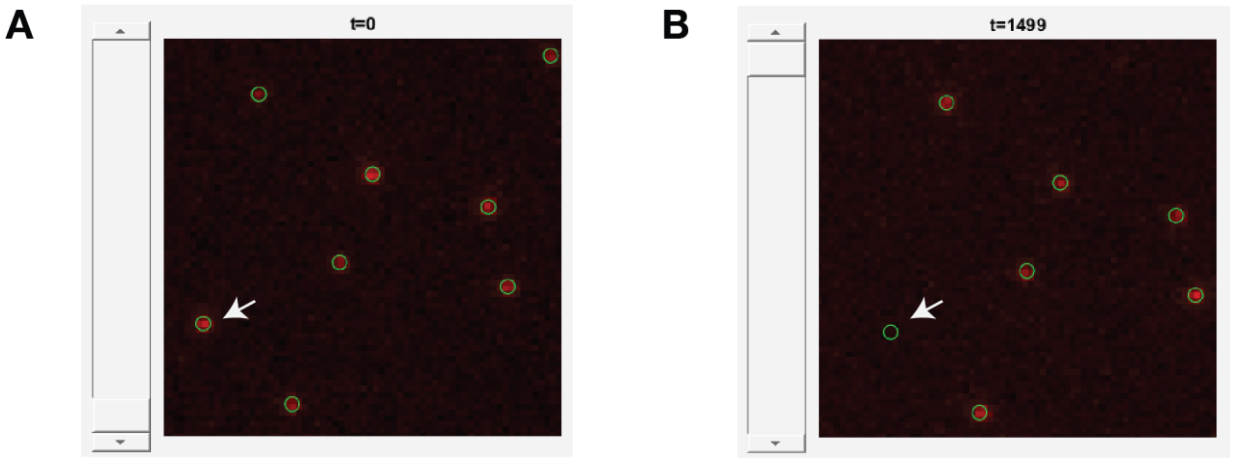


Figure 17. Quality control of drift correction. Target locations are within the green circles in the first frame (A), as well as in the last frame (B), indicating that the drift correction was correctly performed. Of note, one target molecule (white arrow) departed from the slide upon data acquisition. The user has a possibility to exclude this location from further analysis by double-clicking on the spot.

Procedure

In the above steps the pipeline used the optimal Generalized likehood ratio test (GLRT) detector at a fixed false-positive rate [5]. Briefly, we perform a test on the presence of a single molecule (H_1), i.e., the intensity of the single molecule being non-zero ($\theta_I \neq 0$), and on the absence of a single molecule (H_0), i.e., the intensity being equal to zero ($\theta_I = 0$). The motivation for the GLRT is that both the intensity θ_I and background θ_{bg} are unknown and have to be estimated. Both values, intensity and background, are found by Maximum Likelihood Estimation (MLE), one estimate for each of the two hypotheses.

The pipeline determines the location of a single molecule with sub-pixel precision. The most appropriate algorithm in localization microscopy for fitting an imaging model to the data is the Maximum Likelihood Estimator (MLE), where the MLE gives optimal and fast results when an accurate noise and imaging model are applied [6]. Here we chose to approximate the Point Spread Function (PSF) by a Gaussian distribution, which is known to be a valid approach in the context of 2D single emitter localization [7].

Visualization of fluorescence intensity time traces

Click on “Analyze” in Analysis HMM tab (Fig. 18) in order to obtain binding traces.

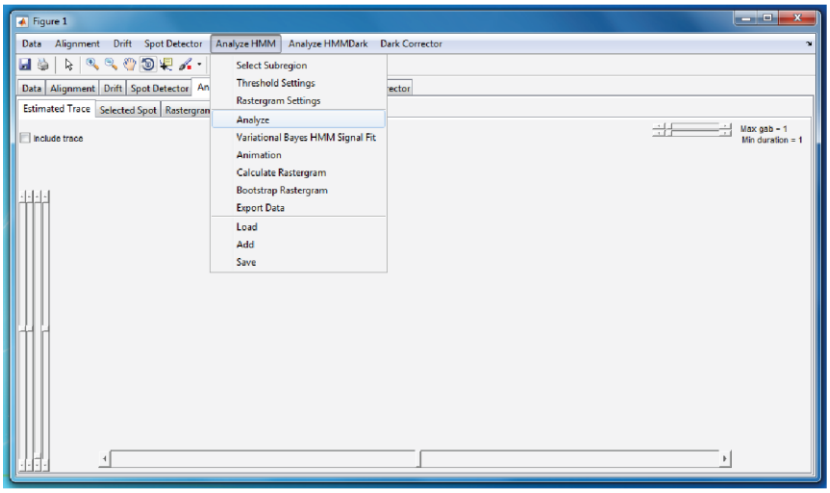


Figure 18. Obtaining fluorescence intensity time traces. Menu tab “Analyze HMM” includes “Analyze” and “Threshold Settings” items.

The pipeline displays fluorescence intensity time traces for each target location (Fig. 19). In the example experiment, TtAgo:guide intensity is shown in green and DNA target molecules in red. This color code can be changed by providing different values for left and right wavelengths upon data loading (Fig. 5b). Light brown indicates background levels of green fluorescence, whereas the black line denotes binding events detected by the pipeline allowing the user to visually inspect the algorithm’s interpretation of the data.

The pipeline displays fluorescence intensity time traces in interactive mode. The pipeline provides information on the mobile component (intensity of the signal, intensity of the background and spot width) and on its relative position to the target molecule at time *t* (Fig. 19, #1). Images of the mobile component and of the target molecule are shown at the same time *t*. By moving the black dashed line (Fig. 19, #2) the user can navigate from the beginning of the trace to its end; information on the mobile component and its image are updated in real time.

To display fluorescence intensity time traces for another target molecule, the user can use the horizontal scroll bar (Fig. 19, #3).

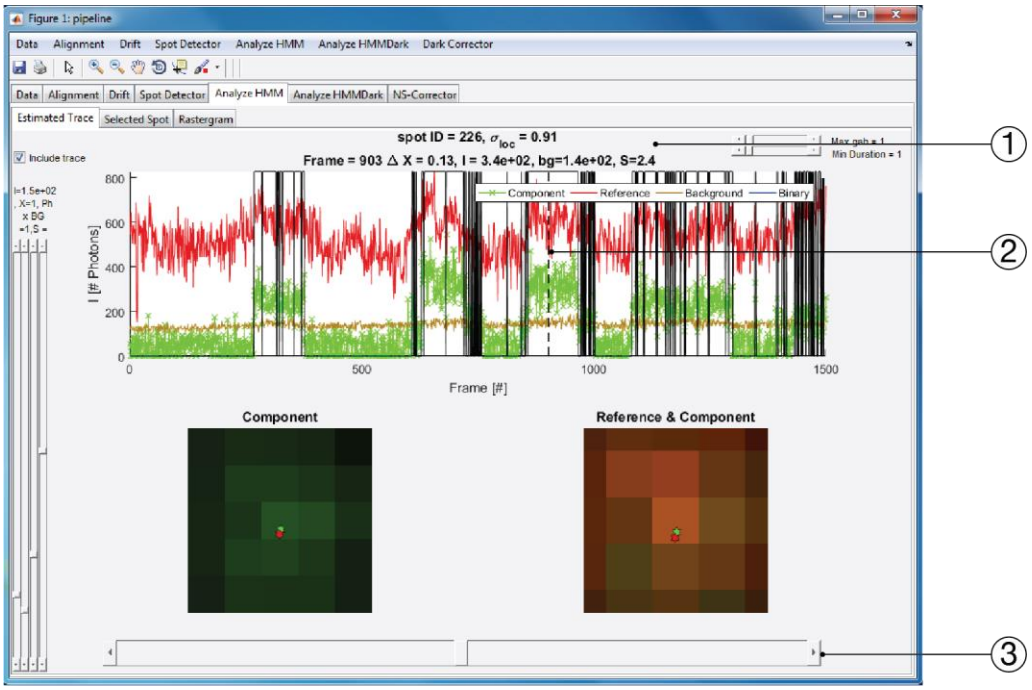


Figure 19. The pipeline displays fluorescence intensity time traces. TtAgo (green) binds DNA target (red); molecule #226 is displayed as an example. Light brown indicates background levels of green fluorescence, whereas the black line denotes binding events detected by the pipeline. Information on the mobile component binding to the target at time *t* (frame 903) is shown in (1). Images of the mobile component and of the target molecule at the same time *t* are shown under the fluorescence intensity time traces. The black dashed line (2) is used to navigate in time. The horizontal scroll bar allows navigation from one target molecule to another.

Co-localization analysis

The user needs to modify threshold parameters to correctly detect co-localization events (Fig. 20). Threshold parameters are:

- intensity of the mobile component (in photons),
- signal-to-background ratio of fluorescence of the mobile component,
- spot width of the mobile component (in pixels),
- relative distance of the mobile component to the target molecule (in pixels).

The user is free to set thresholds as he or she judges best. However, the pipeline provides best-practice values:

- signal-to-background ratio > 1,
- spot width of the mobile component < width of PSF,
- relative distance of the mobile component to the target molecule < average estimated localization precision.

Notes:

1. Average estimated localization precision depends on the width of PSF and on intensities of the signal and the background. It is calculated by the pipeline and is displayed as in Fig. 19 (#1).
2. Width of PSF is specific to each imaging system.
3. Thresholding the intensity of the mobile component is optional. We don't advise using it alone, because this approach does not account for variations in field illumination, which typically are caused by the relay optics delivering light to the sample.

In the example, TtAgo binding to target molecules was scored when the following criteria were met: (1) intensity of TtAgo complex > 150 photons, (2) signal-to-background ratio > 1, (3) the mobile component must be detected within 1 pixel of the target molecule, defined according to the average estimated localization precision, and (4) the spot width must be smaller or equal to 4.6, defined according to the width of the PSF of the microscope (Fig. 20).

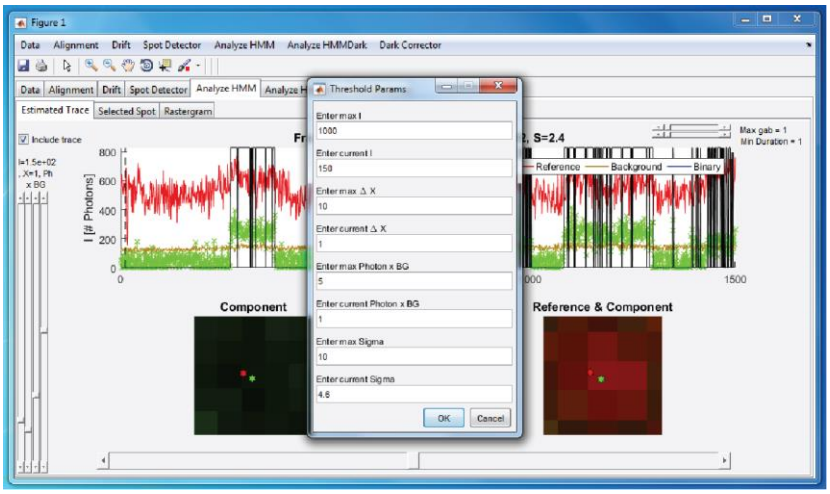
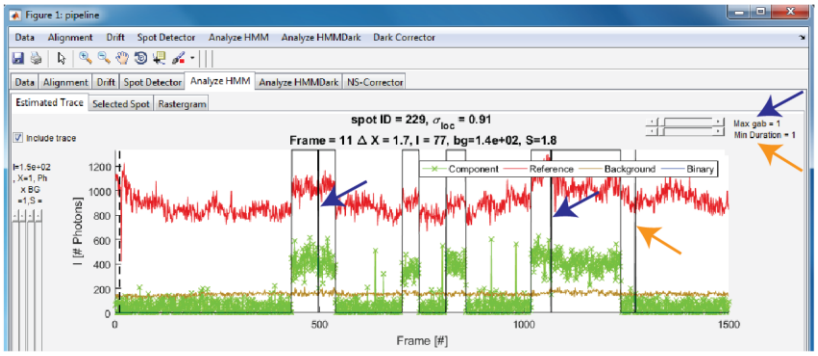


Figure 20. Threshold parameters used in the example. "Threshold Settings" item can be found in "Analyze HMM" menu tab (Fig. 18).

Fluorescent signal can be momentarily lost (for example due to dye blinking) (Fig. 21a). The user may decide to compensate for this phenomenon by increasing the gap closing (expressed in number of frames). In addition, the user has a possibility to eliminate short sporadic binding events by increasing the minimum duration of binding events (Fig. 21b).

A



B

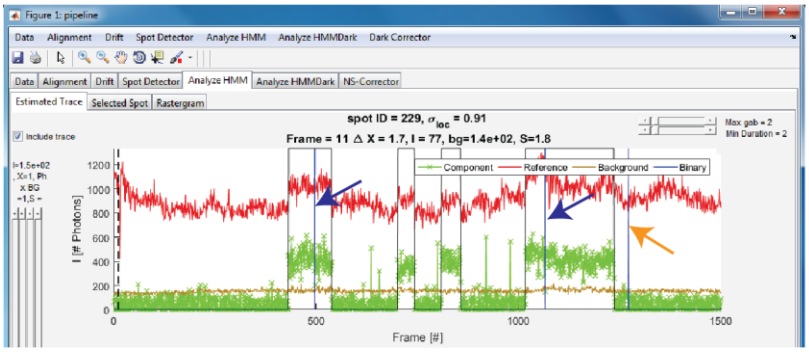


Figure 21. Fluorescence intensity time trace. Target location 229 is shown as an example. (A) Fluorescent signal from the TtAgo:guide complex was momentarily lost (blue arrows), leading to incorrect segmentation of binding events. On another hand, a short binding event is detected (orange arrow) and may correspond to a false positive event. Adjusting the gap closing (blue arrow) and minimum duration of a binding event (orange arrow) yields more accurate trace segmentation (B). The black line denotes binding events detected by the pipeline after event filtering, whereas the blue line indicates the initial co-localization intervals.

If binding events are not detected successfully, check Troubleshooting – Fluorescence intensity time traces.

Procedure

Individual molecules are fitted to a Gaussian with a constant background offset. The background is expressed in photons per pixel; single molecule intensity is the number of photons per the point spread function (PSF). Therefore, we multiply the background (bg) by the area of the PSF and obtain a quantity that has the same units $\mathrm{bg_{tot}} = 4\mathrm{bg}\sigma_{\mathrm{PSF}}^2$, where σ_{PSF} is the width of the Gaussian of the PSF. The distance between the target and the binding complex is calculated as $\Delta r = \sqrt{\Delta x^2 + \Delta y^2}$.

The co-localization criteria that can be set evaluate the intensity (I), the distance (Δr), the width of the PSF (σ_{PSF}), and the signal-to-background ratio ($\mathrm{SNB} = I/(4\mathrm{bg}\sigma_{\mathrm{PSF}}^2)$). The colocalization criterion is set as $\mathrm{coloc} = \mathrm{SNB} > \mathrm{crit_{SNB}}\ \&\ I > \mathrm{crit_I}\ \&\ \Delta r < \mathrm{crit_{\Delta r}}\ \&\ \sigma_{\mathrm{PSF}} < \mathrm{crit_{\sigma_{PSF}}}$, where $\mathrm{crit_i}$ is determined by a user-set slider in the GUI and coloc is a binary value where 0 corresponds to no co-localization and 1 equals co-localization.

Data analysis

This section describes how to use Hidden Markov Models to estimate the number of complexes bound to target molecules with multiple binding sites, how to calculate association and dissociation rates, and to correct for non-specific binding of the mobile component to the glass surface.

Hidden Markov Models

If your experiment is designed so that only one mobile component can bind a single target molecule (i.e., only one binding site is present), skip this section and proceed with the section entitled “Analyzing binding kinetics”.

In the example here, the DNA target molecule contains two identical binding sites that TtAgo can bind simultaneously. The pipeline applies Multivariate Hidden Markov Models to distinguish intervals when one site is bound from intervals when TtAgo binds both sites. The user should provide the prior parameters, which can be estimated from fluorescence intensity time traces. We suggest first performing HHM analysis using estimated mean intensity (Fig. 22) and leaving the remaining parameters as the default values (Fig. 23). Then, other parameters (κ, ν, $W^{1/2}$) should be adjusted to obtain optimal trace segmentation.

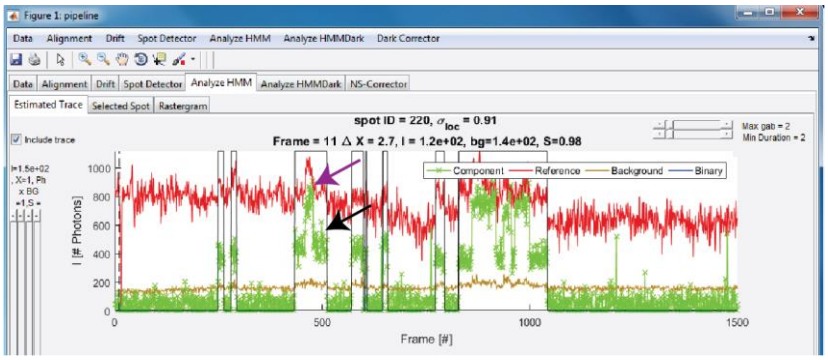
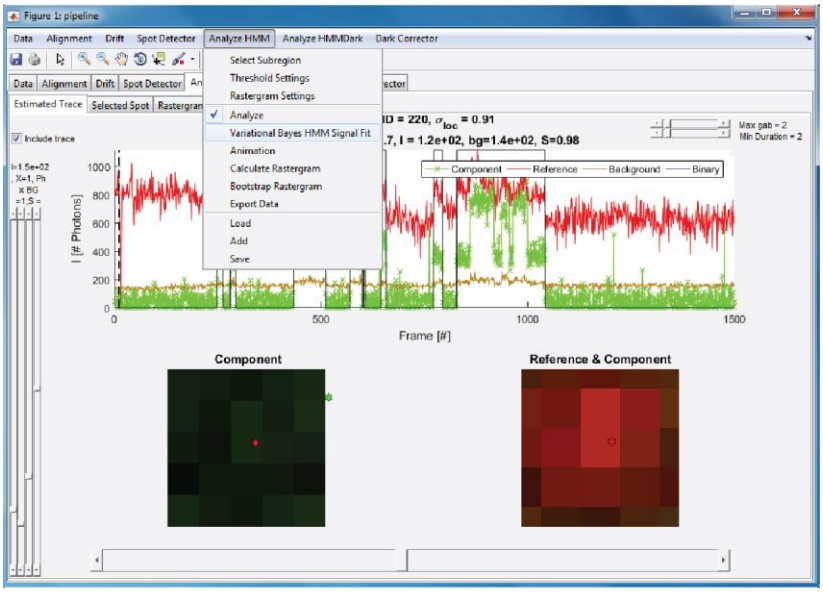


Figure 22. Estimation of mean intensity values from fluorescence intensity time trace. Intensity of one TtAgo:guide complex binding DNA target corresponds to ≈ 500 photons (black arrow), and intensity of two TtAgo:guide complexes corresponds to ≈ 1,000 photons (purple arrow).

A



B

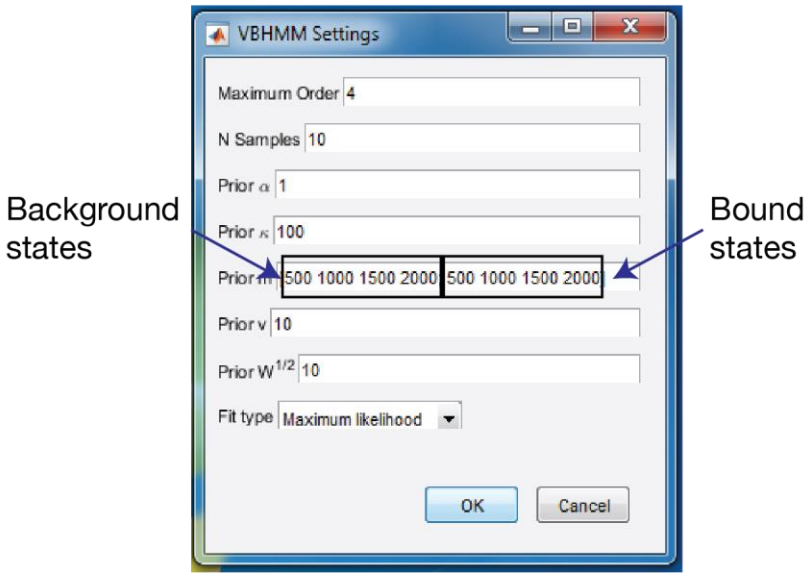


Figure 23. Running HMM analysis. (A) Menu item "Variational Bayes HMM Signal Fit" is selected. (B) Analysis is performed with default parameters. Mean background and intensity of bound states was estimated from fluorescence intensity time trace (Fig. 22), from which an increment of 500 photons was chosen. The maximum likelihood probability of an event is the probability of the event that is most likely given data. The posterior probability of an event is the conditional probability that is assigned after the relevant evidence or background is taken into account as a prior.

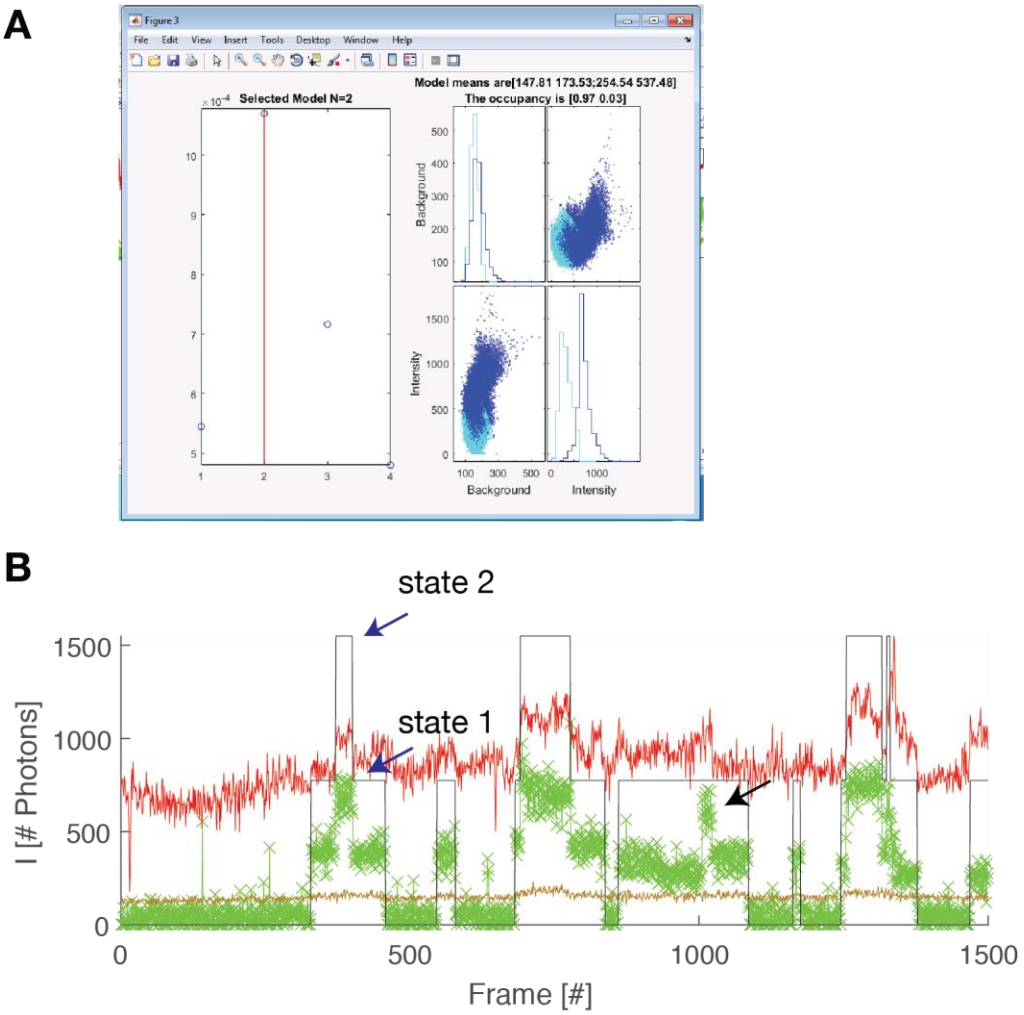


Figure 24. HMM estimates number of bound states. (A) The most probable model corresponds to two bound states. Model means are indicated for both background and bound states. Occupancy of each state is also reported. (B) Fluorescence intensity time trace of spot 126 is shown as an example. Black line denotes binding events detected by the pipeline after HMM analysis. Black arrow points to a 2-state binding event that was not correctly detected.

Analysis performed with default parameters, where only values of mean intensity were estimated from fluorescence intensity time trace, allowed correct estimation of the number of bound states and yielded accurate segmentation of intensity time traces (Fig. 24). However, some intervals in the plot were not correctly defined (black arrow in Fig. 24b); thus prior parameters need to be further adjusted. In short:

m	mean value of the Gaussian observations for each state (Gaussian distribution).
κ	variance of the Gaussian variance of observations values (prior for how easy a transition from one state to another can happen; Gaussian distribution)
$W^{1/2}$	mean of the prior on the variance of the Gaussian observations (Wishart distribution)
α	prior on initial state probability (prior for the system starting state; Dirichlet distribution)
v	variance of the prior on the variance of the Gaussian observations (Wishart distribution)

These parameters are parameters from the assumed priors. A single observation from state k is assumed to be multivariate Gaussian distributed (x_t):

$$p(x_t|\mu_k, \lambda_k) = N\left(\mu_k, \lambda_k^{-1}\right)$$

This distribution has a mean μ_k and a covariance λ_k. To ensure that we are not fitting noise, prior knowledge of μ_k and λ_k is included. This is done by assuming that the mean μ_k again comes from a multivariate Gaussian $N\left(\mu_k|m_k, (\kappa_k\lambda_k)^{-1}\right)$, which has a known mean m_k and a covariance equal to $\kappa_0\lambda_k$. We know that the covariance matrix is always positive (definite) and therefore we assume that the variance comes from a Wishart distribution $W(\lambda_k|W_k, v_k)$. The parameters that determine the prior information are m_k, κ_k, W_k, v_k.

Default value of α is set to 1, i.e., probabilities of the binding event to start in state 1, or 2, or … n are equal for all states. In the example, this situation was suitable for all datasets tested. On the other hand, *κ* indicates how likely it is for a state transition to occur. This is the key parameter to adjust.

To determine optimal value of *κ*, the user needs to visually examine a small number of fluorescence intensity time traces and verify whether the traces were segmented into intervals with correctly detected number of bound states. We suggest running the first round of analysis with a small value of *κ*, for example *κ* = 30 as in Fig. 25A. If *κ* is small, the transition is more likely to occur. Therefore, if *κ* is inferior to its optimal value, even a small fluctuation in the fluorescent signal may be considered as a transition. In the example in Fig. 25A, the black arrow points to an increase in fluorescence, which is in reality experimental fluctuation, but is detected as an additional bound state. To rule this possibility out, we suggest analyzing a control dataset, i.e. binding of the mobile component on a target with a single binding site (not shown here), with the same value of *κ*. If these fluctuations in the control dataset are identified as transitions, the value of *κ* needs to be increased. Setting the value of *κ* to 100 yields in correct segmentation of the trace (Fig. 25B). Finally, if *κ* is superior to the optimal value, transitions become unlikely to occur and some states can be missed in the analysis. The black arrow in Fig. 25C points to an interval, in which only the second half was identified correctly as state “2 bound sites” and the first half mistakenly considered as state “one bound site”.

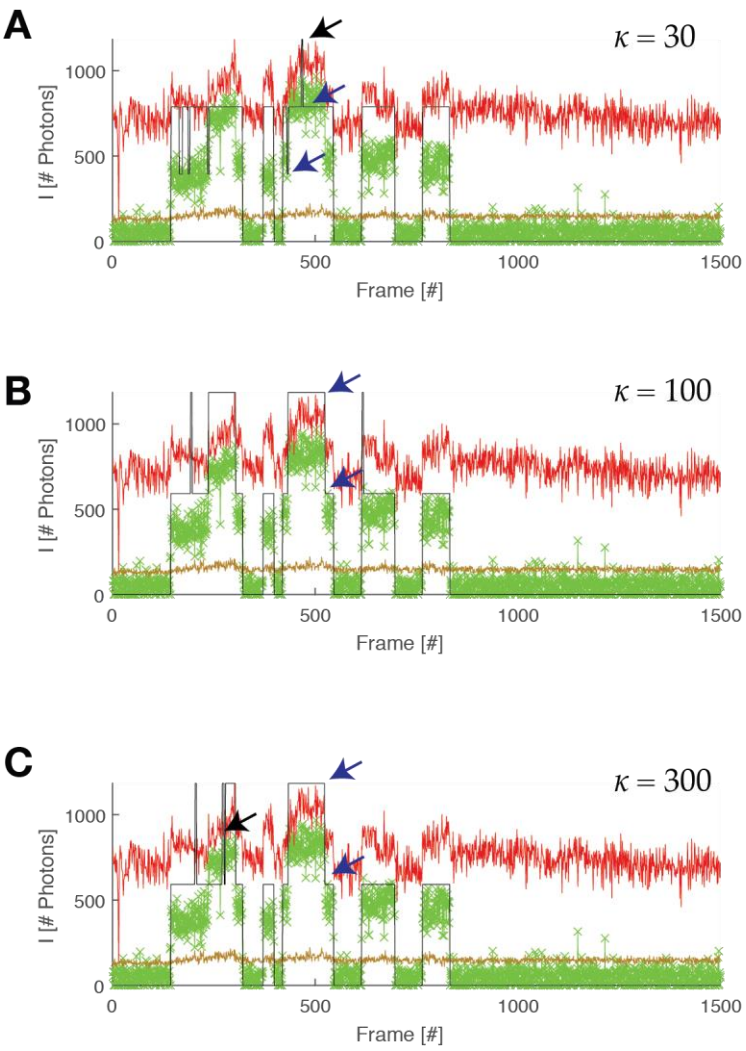


Figure 25. Optimizing κ. Spot 167 is shown as an example. (A) HMM analysis with κ = 30 yielded three bound states. While blue arrows point at expected bound states, the black arrow indicates the third, wrongly detected state. (B) Increasing κ to 100 yielded two states (blue arrows). The trace is segmented correctly. (C) Increasing κ to 300 decreased the likelihood of a transition. In particular, the black arrow indicates the first half of 2-state event was not detected as such.

Procedure

The first step is to estimate the number of binding states from the single molecule co-localization data and thereby characterize kinetic mechanisms not measurable by conventional population methods that aggregate kinetic properties. The second step is to quantify these binding states in terms of the stoichiometric and kinetic properties. Details and mathematical derivations of these steps are provided in Appendix 1.

Analyzing binding kinetics

A rastergram is a commonly used representation of binding events. Rastergrams summarize arrivals and departures for many individual target molecules, each in a single row (Fig. 26).

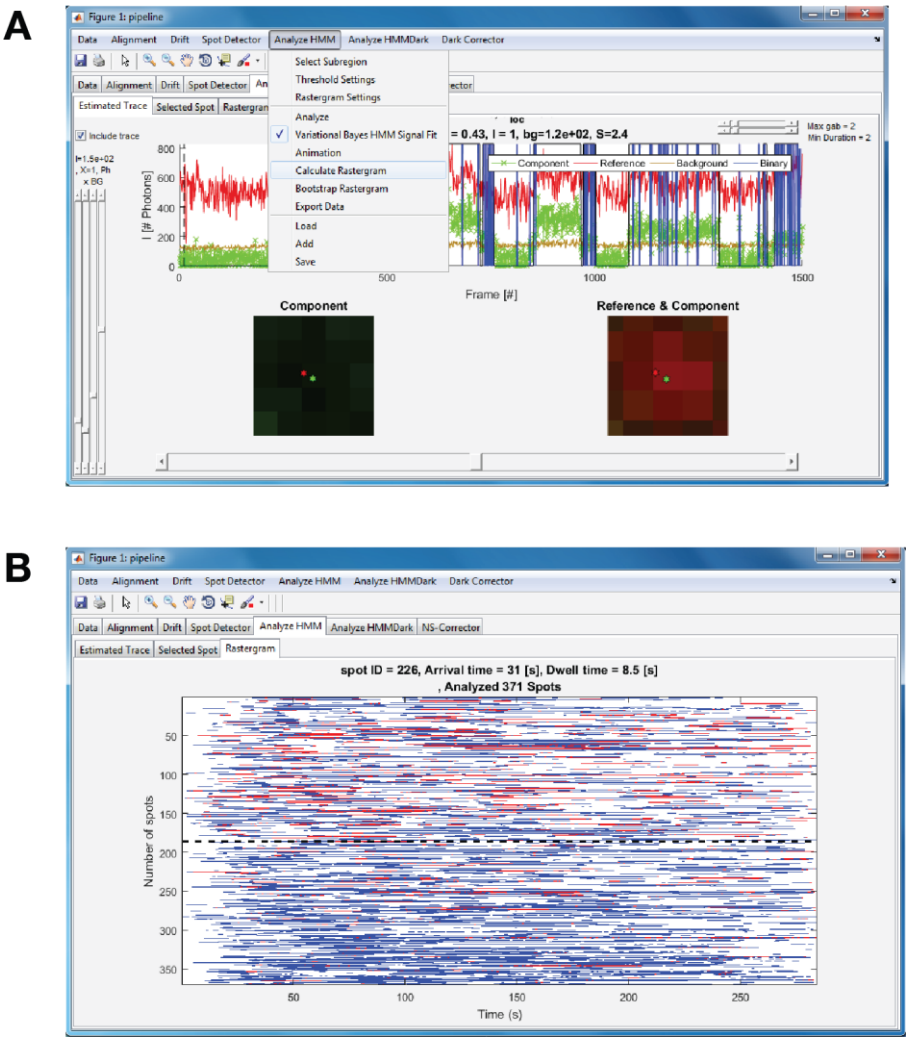


Figure 26. Creating a rastergram and calculating binding properties. (A) Menu item "Calculate Rastergram" is selected. (B) Rastergram representation of 371 target molecules, each in single row. Binding events with one or two TtAgo:guide complexes bound are shown in blue and red, respectively. Dashed line corresponds to row of the target molecule, whose location is indicated as spot ID. Mean arrival time and mean dwell time are calculated for all the target molecules analyzed and are reported in seconds.

The rastergram representation can be modified (Fig. 27): the color code can be changed; rows can be sorted by binding event arrival time or departure time or bound state. Also, if there is a lag between initiating the experiment and acquiring data, a specific frame can be set as $t = 0$.

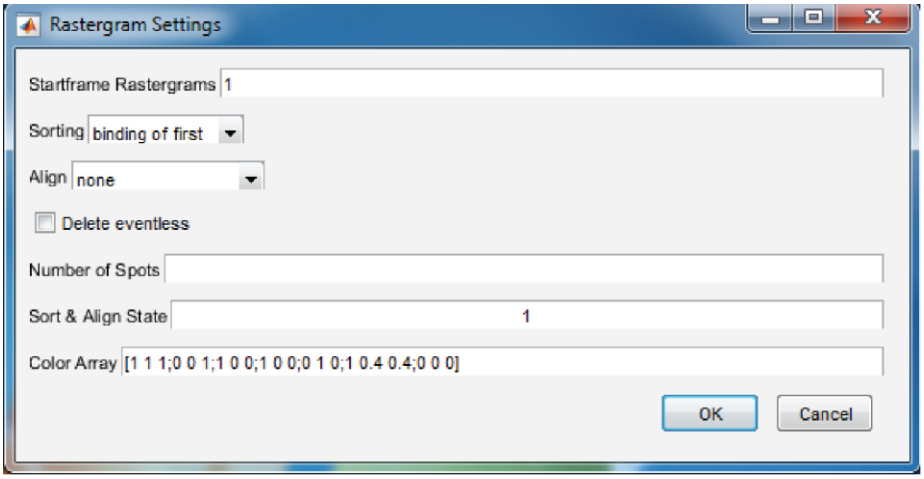


Figure 27. Rastergram settings. Select "Rastergram Settings" in "Analyze HMM" tab (Fig. 26a). Startframe is number of the frame taken to be the starting time of the experiment. Rows can be sorted by duration of the first or last event or arrival time of the first event. Rows can be aligned by the first binding event or last departure event. Finally, rows can be sorted by bound state. By default, a rastergram is plotted for all target molecules. Target locations without binding events can be removed from rastergram plot by checking "Delete eventless." Also, rastergrams can be plotted for a specific number of randomly chosen spots (for example, the user may want to generate a rastergram for the same number of molecules in different experiments). The color code of the rastergram is in RGB format; default set is: background, white (1 1 1); state "1 bound," blue (0 0 1); state "2 bound," red (1 0 0); state "3 bound," green (0 1 0); state "4 bound," in pink (1 0.4 0.4); and state "5 bound," black (0 0 0).

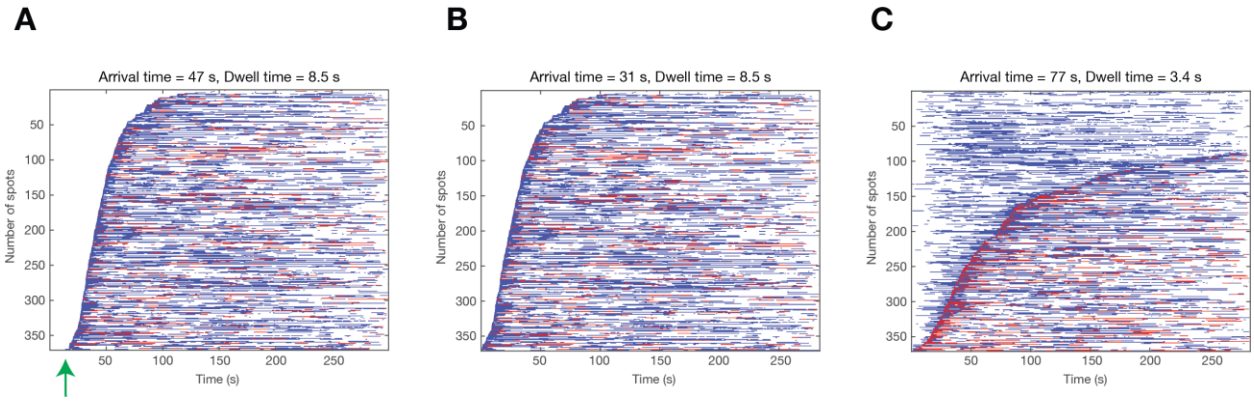


Figure 28. Examples of rastergrams sorted differently. The rastergram from Fig. 26b sorted by arrival time and by state 1 with Startframe = 1 (A) or Startframe = 80; note that the mean arrival time decreases (frame 80 is now considered to be t = 0). (C) The same rastergram sorted by arrival time and by state 2 (Startframe = 80). Mean arrival dwell times are reported for state 2.

Finally, the user can "bootstrap the rastergram" by selecting "Bootstrap Rastergram" in the "Analyze HMM" tab (Fig. 26a). This process corresponds to random sampling with replacement and allows estimation of sample distribution. Average values of bootstrapping are reported for mean arrival and dwell times ± error of bootstrapping (Fig. 29).

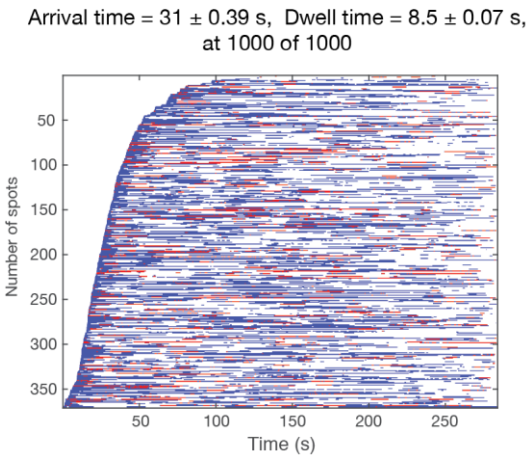


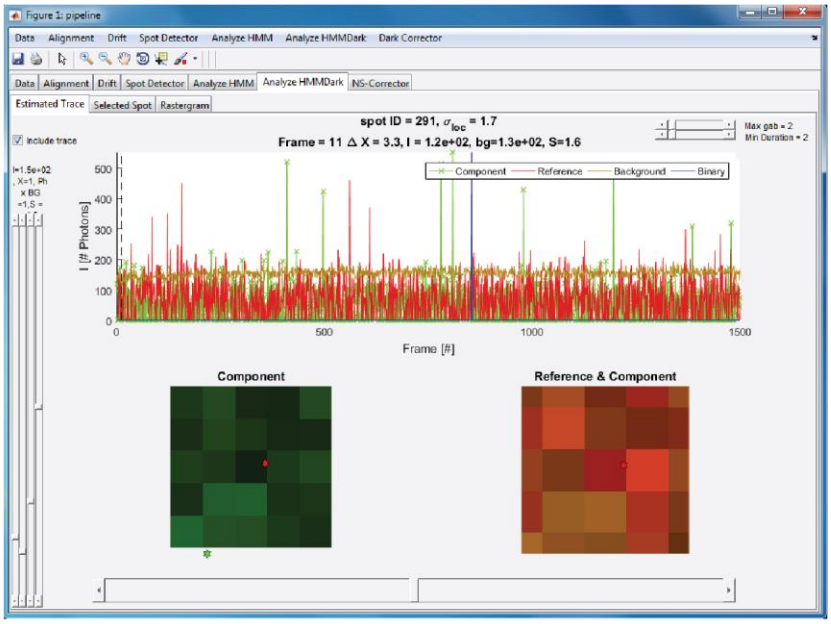
Figure 29. Bootstrapping a rastergram. Number of bootstraps performed = 1000. Values are reported as mean ± standard error from bootstrapping in seconds.

Correction for non-specific binding

Despite efforts to minimize non-specific binding, spurious binding is unavoidable and varies with the nature and concentration of fluorescent molecules. Both on-rate and off-rate measurements are affected by non-specific binding. Correcting for non-specific binding is achieved by measuring binding at "dark" locations, sites that contain no target molecules (detected in Fig. 13).

Perform the analysis on "dark locations" as for target locations (Fig. 30). The thresholds are automatically coupled to the threshold set in the analysis of target locations.

A



B

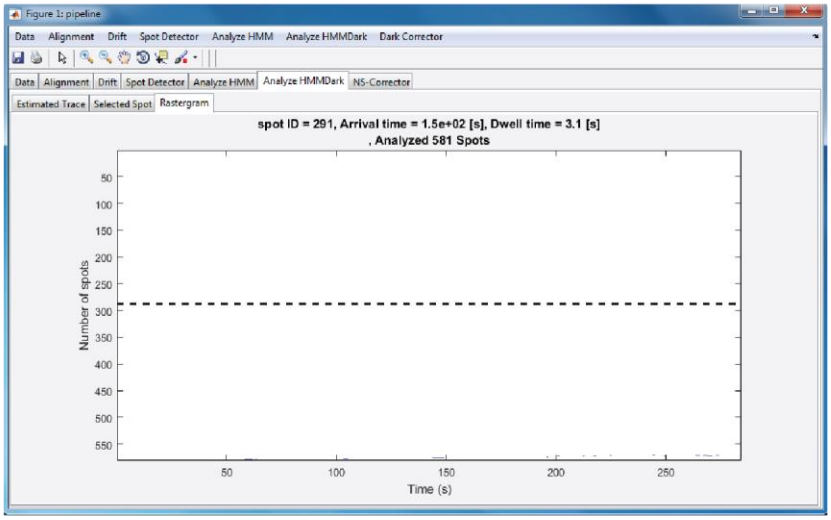


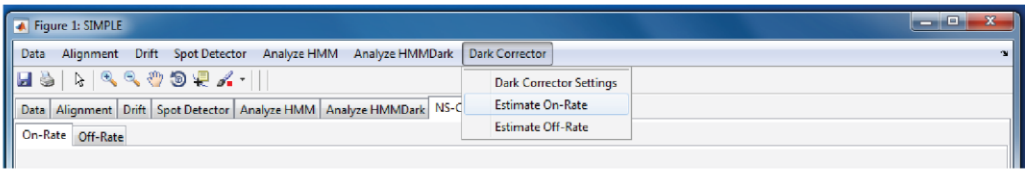
Figure 30. Non-specific binding measured at "dark" locations. (A) Representative fluorescence intensity time trace. No binding event was detected. (B) Rastergram summary of traces of 581 "dark" locations, each in a single row and sorted according to their arrival time. Only few locations display binding of TtAgo:guide, suggesting that non-specific binding is low.

On- and off- rates at "dark" locations can be estimated by fitting either the probability density function, or the cumulative distribution function (Fig. 31). The analysis corrects the on- and off-rates estimated in Fig. 29 for non-specific binding (Fig. 32). When non-specific binding is low and only little data exist, estimation of non-specific binding is more robust by use of the Cumulative Density Function (CDF). The drawback is that no asymptotic optimality condition exists for CFD-based estimation. Therefore, in the presence of enough data it is better to correct for non-specific binding by using the Probability Density Function (PDF) (see also Troubleshooting – Correction for Non-Specific Binding).

Note: When non-specific binding is very low, the small number of "dark" locations bound by the mobile component is insufficient for robust correction. If the correction results in an error message, check Troubleshooting – Correction for Non-specific binding. The source of this problem is usually fixed by merging data from multiple experiments (see next section).

> *Procedure*
>
> In the above example, both the corrected on- and off-rates were estimated. The estimation process, as well as a mathematical derivation is described in detail in Appendix 1.

A



B

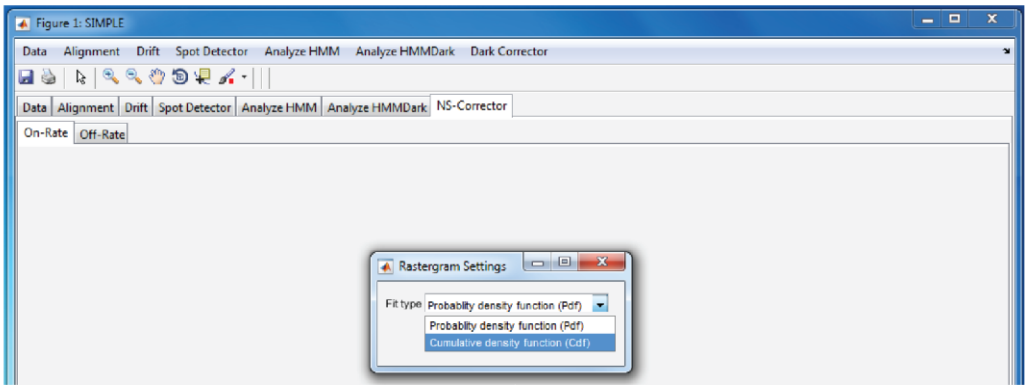
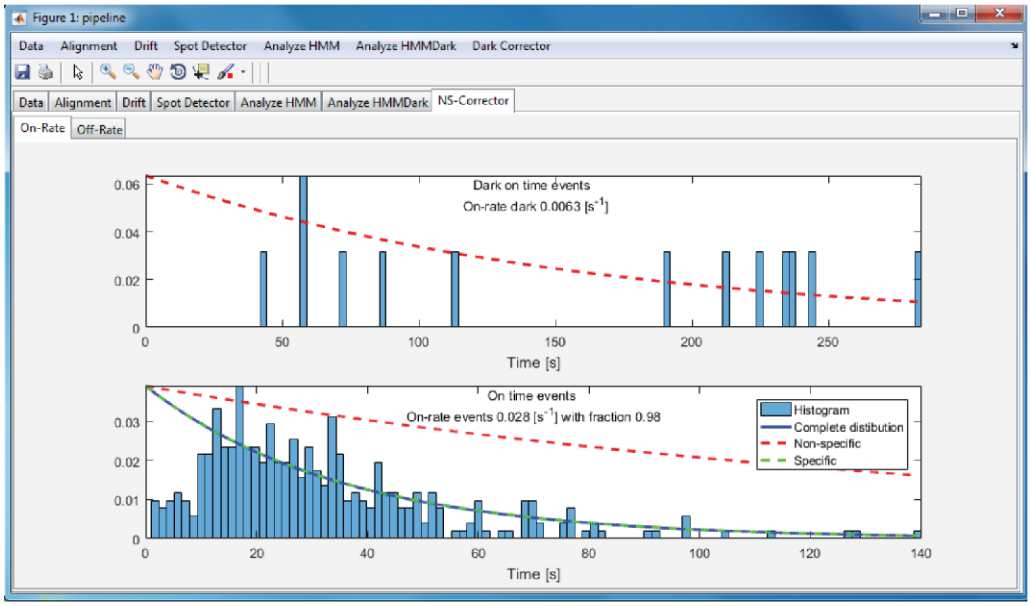


Figure 31. Correction for non-specific binding. (A) Menu tab "Dark Corrector" includes "Dark Corrector Settings", "Estimate On-rate" and "Off-Rate" items. (B) Correction for non-specific binding of on- and off-rates can be performed by fitting the Probability Density function or the Cumulative Distribution function.

A



B

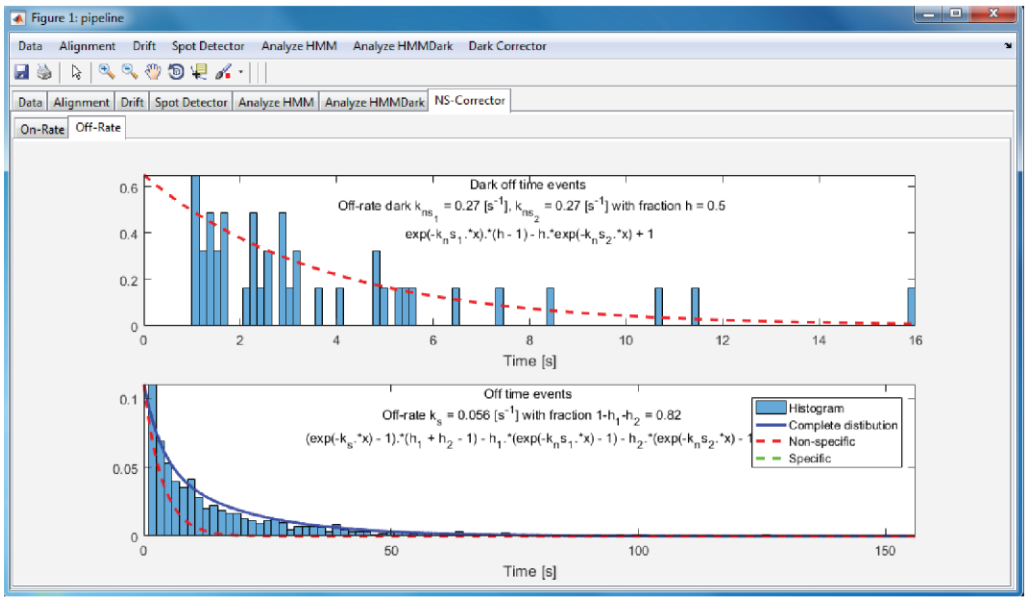


Figure 32. Correction of on- (A) and off- (B) rates for non-specific binding. Fitting the Cumulative distribution function was performed.

SAVE AND EXPORT ANALYSIS RESULTS

Save your analysis

The pipeline allows merging the processed data from several individual experiments in order to estimate the kinetic behavior of the mobile component using a larger number of molecules (see *Pooling multiple experimental replicates*, below). This requires that the analysis from each experiment be saved. The feature "Save" in "Analyze HMM" tab (Fig. 33) ensures saving of target location data (alignment of cameras, drift correction, fluorescence intensity time traces, and segmentation into binding events) as a .mat file. The same should be performed for the dark locations ("Save" item in "AnalyzeHMMDark" Menu tab).

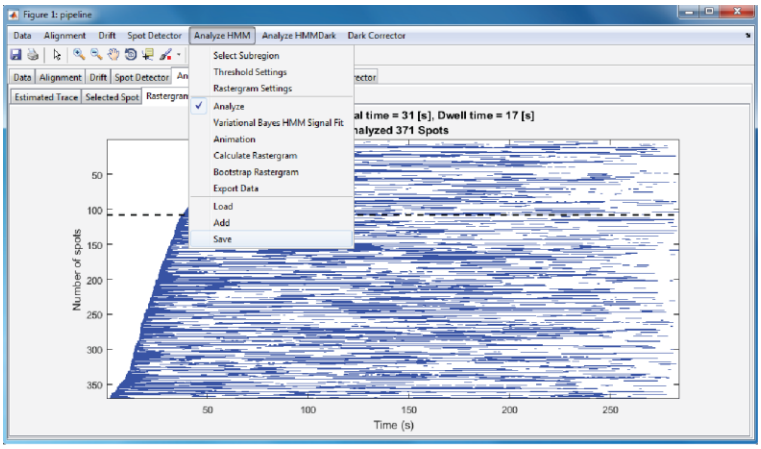


Figure 33. Save the target location object in MAT format (.mat file). Choose "Save" item, and then name your file.

Open your saved analysis

To go back to a previously saved analysis, open the GUI interface (run pipeline_interface.m), but instead of loading new dataset, click on "Load" item in the "Analyze HMM" menu tab (Fig. 34) and select the .mat file of target locations. Similarly, click on "Load" item in "Analyze HMMDark" menu tab and select the .mat file of dark locations.

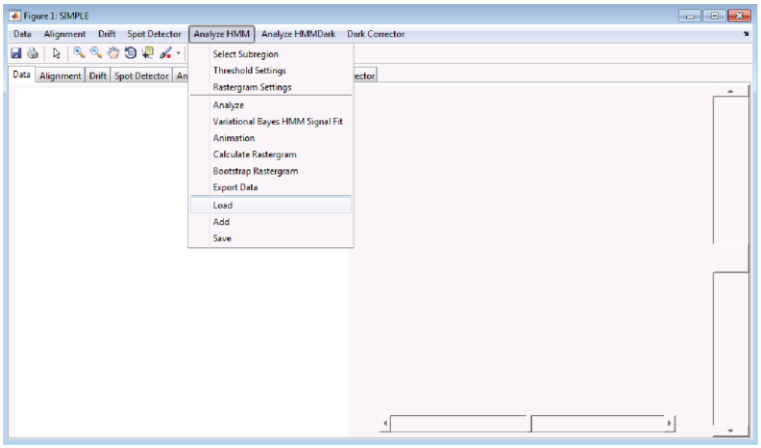


Figure 34. Open previously saved target locations. Click on "Load" item, and then choose the corresponding file.

Exporting figures

The pipeline supports easy sharing of results (fluorescence intensity time traces and rastergrams) as PDF files (Fig. 35), which can be further edited in vector graphics software.

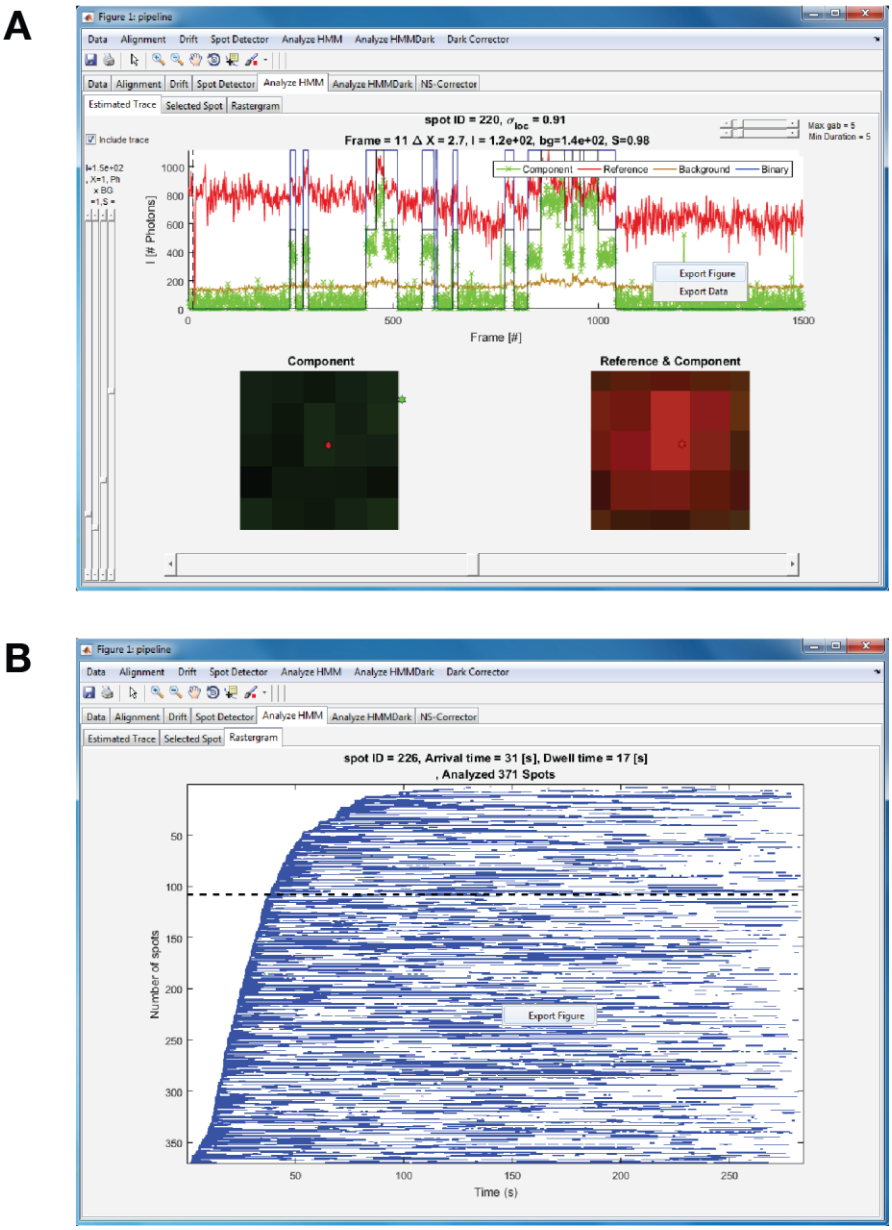


Figure 35. Export figures and data. A right click on fluorescence intensity time traces (A) or rastergrams (B) reveals the “Export Figure” menu item.

Exporting data for further analysis

If the user wants to perform the fundamental steps of data analysis using the pipeline (including mapping, drift correction, spot discrimination, and segmentation of binding events) and continue with other processes that the pipeline doesn't offer, data can be exported as .mat file (Fig. 36; Table 2).

Variable	Data	Type
Trace	Binary binding traces	Logic Array
spotIDs	Unique identification number of target	Integer Array
Background	Estimated background	Float Array
DxSq	Estimated colocalization distance	Float Array
Sigma	Estimated spot width	Float Array

Table 2. List of variables that can be exported for further analysis.

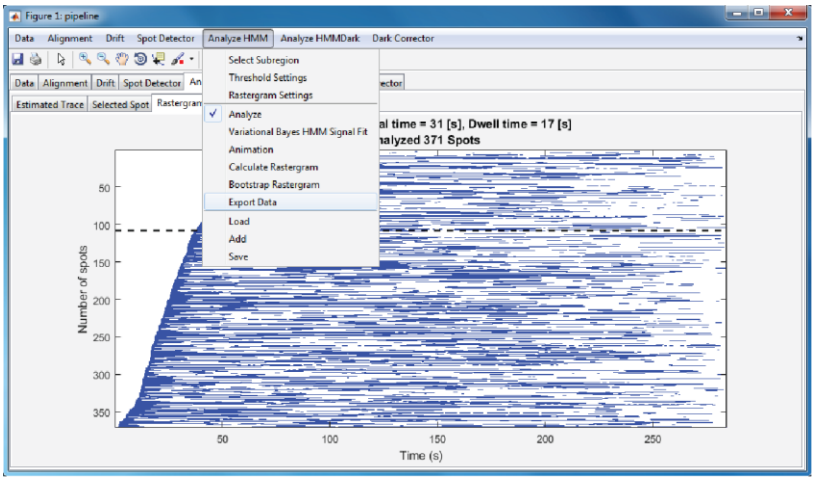


Figure 36. Export the target location data in MAT format (.mat file). Choose "Export Data" item, and then name your file.

Pooling multiple experimental replicates

The pipeline allows merging the processed data from several individual experiments in order to estimate the kinetic behavior of the mobile component using a larger number of molecules. Each experiment should be analyzed independently and saved as ".mat" files (both target and dark locations).

Open the GUI interface (run pipeline_interface.m) and load one of the saved replicates (Fig. 34). Then, add target locations of another saved replicate by clicking "Add" item in "Analyze HMM" menu tab (Fig. 37). Repeat this step for dark locations.

Now, the user can perform the same analysis on data from multiple experiments: plot rastergrams, calculate mean arrival and dwell times and correct them for non-specific binding. Note that threshold parameters for detection of binding events (Fig.20), as well as the gap closing and the minimum duration parameters (Fig. 21), should be the same in all the replicates merged. However, the start frame for each experiment can vary (Fig. 38).

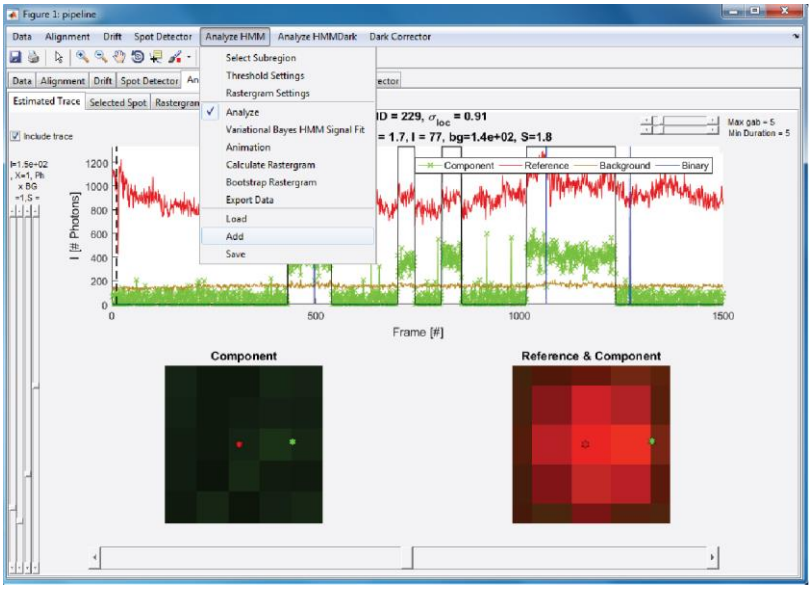


Figure 37. Add previously saved target locations to on-going analysis. Choose “Add” item, and then choose the corresponding file.

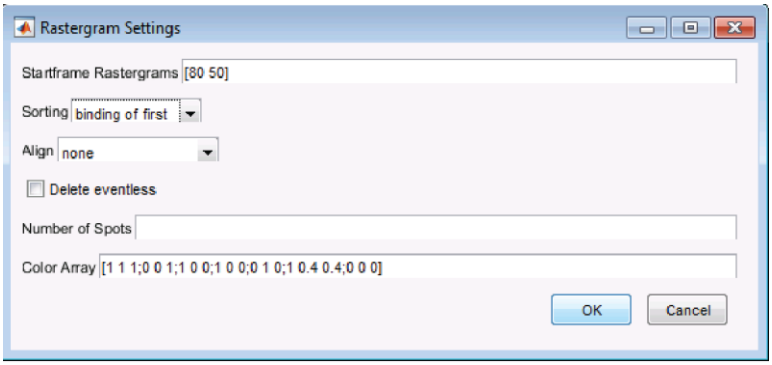


Figure 38. Setting the beginning of each replicate. Select “Rastergram Settings” in “Analyze HMM” tab (Fig. 26a). Indicate Startframe for each replicate in format [Startframe1 Strartframe2 etc]. In this example, experiments 1 and 2 start at frames 80 and 50, respectively.

BATCH PROCESSING

To ensure reproducibility of user settings and to analyze the same or a similar experimental dataset, we suggest scripting all the steps and the parameters.

This section contains two script templates. The user can copy-paste them to create his owns. Comments in green provide explanations for script commands. To help the user in filling in the commands, we have included in the package scripts used for the Example dataset (Appendix 2).

- **"Single replicate template"** will help to script analysis of an experimental dataset. Run your analysis via GUI interface first. While the interface is still open, create a new .m file in the Editor and copy-paste "**Single replicate template**". Follow instructions in green below to identify important parameters to save. You can also take inspiration from two scripts Example_dataset_analysis.m and Example_dataset_analysis_with_thresholds.m (Appendix 2), which encode the analysis of the Example dataset presented in this tutorial.
- **"Pooling template"** will help to script pooling of several analyzed and saved replicates. Create a new .m file in the Editor and copy-paste "**Pooling template**". Follow instructions in green below to identify important parameters to save. You can also take inspiration from the script Example_dataset_pooling_processed_data.m (Appendix 2).

Single replicate template

```
% Make sure that this function is set as you working directory, because ...
%the script will add the necessary dependencies.
close all
clearvars

% Add to path the folder containing required functions
addpath(genpath('.\helperfunctions'))

% Here the user indicates where are located the dataset to be analyzed and ...
%how the experiment was done (number of frames per second, wavelengths of ...
%the two cameras etc).
% Check gdlObj.fileStructure to get all single variables. For example, ...
%type in the command line "gdlObj.fileStructure.Grid" (without quotation ...
%marks) and copy-paste the answer.

fileStructure.Grid = ''; % write the answer between ''
fileStructure.Dark = ''; % write the answer between ''
fileStructure.BeadData = ''; % write the answer between ''
fileStructure.Data = ''; % write the answer between ''
fileStructure.rg = ; % write the answer before ; mark
fileStructure.NFramesGroup = ; % write the answer before semicolon (;)
fileStructure.lambdal = ; % write the answer before semicolon (;)
fileStructure.lambdar = ; % write the answer before semicolon (;)
fileStructure.intT = ; % write the answer before semicolon (;)
fileStructure.flipCams = ; % write the answer before semicolon (;)
```

```
numberOfSummedFrames= ; % write the answer before semicolon (;)

% Now the pipeline will load the datasets for the analysis. The user does ...
%not need to change anything in the code.

h = createParentFigure;
cameraIndexDriftEst=1;

htabgroup = uitabgroup(h);
gdlObj = guiDataLoader(htabgroup,fileStructure,[],numberOfSummedFrames,1);
set(0, 'CurrentFigure', h)
gaObj = guiAlignmentw(htabgroup,gdlObj);
gdcObj = guiDriftCorr(htabgroup,gdlObj,cameraIndexDriftEst);
gsdObj = guiSpotDetector(htabgroup,gdlObj,[],[],[],[],true);
gadObj = guiHMMAnalyse(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdrObj = guiDarkROIsw(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdcorObj = guiDarkCorrector(htabgroup,gadObj,gdrObj);
gocObj = guiDataCollector(gdlObj,gaObj,gdcObj,gsdObj,gadObj,gdrObj);

% Set parameters for choosing beads used in the alignment.

gaObj.spotDetectorCam1.paramsFilterFits.minPixelDist = ; % write the answer...
    %before semicolon (;) Default = 2.
gaObj.spotDetectorCam2.paramsFilterFits.minPixelDist = ; % write the answer...
    %before semicolon (;) Default = 2.
gaObj.spotDetectorCam1.paramsPreFilterFits.clusterSizeMax = ; % write the ...
    %answer before semicolon (;) Default = 50.
gaObj.spotDetectorCam2.paramsPreFilterFits.clusterSizeMax = ; % write the ...
    %answer before semicolon (;) Default = 50.

gaObj.alignmentEst; % Now the pipeline will perform alignment.

%%

% Now the pipeline will perform drift correction.
% Type gdcObj.C to know which region was used for drift correction in the ...
%ongoing analysis and set region.

C = [...
;... % Copy the first row of the answer before semicolon (;)
]; % Copy the second row of the answer before semicolon (;)

gdcObj.driftEst(C);

% Now the pipeline will detect locations (called here spots) of target ...
%molecules and mobile components.

% Set parameters used for pre-filtering spots

gsdObj.paramsPreFilterFits.circularityMin = ; % write the answer before semicolon
gsdObj.paramsPreFilterFits.circularityMax = ; % write the answer before semicolon
gsdObj.paramsPreFilterFits.PH1Min = ; % write the answer before semicolon
gsdObj.paramsPreFilterFits.PH1Max = ; % write the answer before semicolon
gsdObj.paramsPreFilterFits.minPixelDist = ; % write the answer before semicolon
```

```
gsdObj.paramsPreFilterFits.clusterSizeMin = ; % write the answer before semicolon
gsdObj.paramsPreFilterFits.clusterSizeMax = ; % write the answer before semicolon

% Set parameters used for filtering spots

gsdObj.paramsFilterFits.MinPhotons = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MaxPhotons = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MinBg = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MaxBg = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MinPValue = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MaxPValue = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MinPixelDist = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MinCRLBSTD = ; % write the answer before semicolon (;)
gsdObj.paramsFilterFits.MaxCRLBSTD = ; % write the answer before semicolon (;)

gsdObj.detectSpots;

% Now the pipeline will plot the fluorescence intensity time traces of ...
%target molecules and mobile components and detect co-localization events.

% Set the criteria for co-localization. First, type "gadObj.getThreshold" ...
%(without quotation marks) in the command line to get the thresholds used ...
%in the ongoing analysis. Second, type "gadObj.getGabDurSliders" (without ...
%quotation marks) in the command line to get the values of gap closing and ...
%minimum value of a co-localization event.
gadObj.analyze

gadObj.setThreshold([X X X X]); % replace Xs by the values from gadObj.getThreshold
gabDurVec = gadObj.getGabDurSliders;
gadObj.setGabDurSliders([X X]); % replace Xs by the values from ...
    %gadObj.getGabDurSliders

% To find indices of spots, which the user decided to exclude from the analysis,...
%type in the command line "a=find(~gadObj.spotsIncluded)" (without the ...
quotation marks). Then transpose the vertical vector to horizontal vector by ...
typing "b=a.'" (without the quotation marks).

gadObj.spotsIncluded([]) = 0; % Enter the values from b between []. If no ...
spot was excluded from the analysis, leave this line without modification.

% Now the pipeline will plot a rastergram summarizing the co-localization events.

% Enter rastergram settings
gadObj.sortType = ; % 0 if none; 1 if duration of first; 2 if duration of last;
    % 3 binding of first;
gadObj.sortState = 1; %state to sort, default is 1
gadObj.align = ; % 0 if none; 1 if first binding event; 2 if last departure event;
gadObj.delete = ; % 1 if true; 0 if false
gadObj.rastergramStartframe = ; %frame to start rastergram
gadObj.colorArray = ; %rgb values of the color of each state, ...
    %e.g [1 1 1;0 0 1;1 0 0;1 0 0;0 1 0;1 0.4 0.4;0 0 0]
gadObj.numberOfSpotsForRastergram = []; % empty if using all states

gadObj.runRastergram
```

```
% Now the pipeline will analyze non-specific binding at dark locations.
%All the parameters (the criteria for co-localization and the values of gap ...
%closing and minimum value of a co-localization event) are automatically set ...
%at the same values than for analysis of binding at the target molecules.

gdrObj.analyze

% To find indices of spots, which the user decided to exclude from the ...
%analysis, type in the command line "c=find(~gdrObj.spotsIncluded)" ...
%(without the quotation marks). Then transpose the vertical vector to ...
%horizontal vector by typing "d=c.'" (without the quotation marks).

gdrObj.spotsIncluded([]) = 0; % Enter the values from d between [].
    %If no spot was excluded from the analysis, leave this line without modification.

gabDurVec = gdrObj.getGabDurSliders;
gdrObj.setGabDurSliders([X X]); % replace Xs by the values from ...
    % gdrObj.getGabDurSliders

% Now the pipeline will plot a rastergram summarizing the co-localization events.
%All rastergram settings are automatically set at the same values than for ...
%analysis of binding at the target molecules.

% Copy rastergarm settings from the analysis module (gadObj)
gdrObj.copyRastergramSettings(gadObj);

gdrObj.sortType = ; % 0 if none; 1 if duration of first; 2 if duration of last;
    % 3 binding of first;
gdrObj.sortState = ; %state to sort, default is 1
gdrObj.align = ; % 0 if none; 1 if first binding event; 2 if last departure event;
gdrObj.delete = ; % 1 if true; 0 if false
gdrObj.rastergramStartframe = ; %frame to start rastergram
gdrObj.colorArray = ; %rgb values of the color of each state, ...
    %e.g [1 1 1;0 0 1;1 0 0;1 0 0;0 1 0;1 0.4 0.4;0 0 0]
gdrObj.numberOfSpotsForRastergram = []; % empty if using all states

gdrObj.runRastergram

% Now the pipeline saves all processed data (alignment of cameras, ...
%drift correction, fluorescence intensity time traces, and segmentation ...
%into binding events) as .mat files. This is done for target molecules ...
%and dark locations separately.

gdrObj.savevars([mfilename date 'DarkROIs.mat']);
gadObj.savevars([mfilename date 'Target.mat']);

% Set HMM settings
gadObj.prior.alpha = ; % write the answer before semicolon (;)
gadObj.prior.kappa = ; % write the answer before semicolon (;)
gadObj.prior.m = ''; % write the values in [X X; X X] format between ''
gadObj.prior.v = ; % write the answer before semicolon (;)
gadObj.prior.W = ; % write the answer before semicolon (;)
gadObj.NTimes = ; % write the answer before semicolon (;)
```

```
gadObj.NOrder= ; % write the answer before semicolon (;)
gadObj.bayesSignalFitGMM % this command runs VBHMM

% Botstrap the rastergrams
Nsignal=; % write before semicolon (;) how many times you want to bootstrap...
    %the rastergram of target locations
sortStateSignal=; %write before semicolon (;) bound state which was used to ...
    %sort the rastergram that you want to bootstrap
gadObj.bootStrapRastergram(Nsignal,sortStateSignal); % this command runs the ...
    %bootstrapping, i.e. it takes Nsignal times randomly 90% of the data ...
    %and sorts by state specified by the user

Ndark=; % write before semicolon (;) how many times you want to bootstrap...
    %the rastergram of dark locations
sortStateDark=; %write before semicolon (;) bound state which was used to ...
    %sort the rastergram that you want to bootstrap
gdrObj.bootStrapRastergram(Ndark,sortStateDark); % this command runs the ...
    %bootstrapping, i.e. it takes Ndark times randomly 90% of the data ...
    %and sorts by state specified by the user

% Perform correction of on- and off rates for non-specific binding.

gdcorObj.getCorrectedOnrate('X'); % this command runs correction of on-rate.
    %Replace 'X' by 'cdf' or by 'pdf' if you want to use the Cumulative ...
    %Density Function (CDF) or the Probability Density Function (PDF), ...
    %respectively.
gdcorObj.getCorrectedOffrate('X'); % this command runs correction of off-rate.
    %Replace 'X' by 'cdf' or by 'pdf' if you want to use the Cumulative ...
    %Density Function (CDF) or the Probability Density Function (PDF), ...
    %respectively.
```

Pooling template

```
% Make sure that this function is set as you working directory, ...
    %because the script will add the necessary dependencies.
clearvars
close all

gpuDevice;
affine2d;

addpath(genpath('./helperfunctions'))
addpath(genpath('./helperfunctions/ext'))

% If the user didn't save the processed data in mat format but has the ...
    %scripts containing all the parameters, he can automatically run all ...
    %the scripts corresponding to datasets to pool here.

fileNamesSciptToPool{1} = 'script1.m'; % Replace script1 by the name of ...
    %your first file
fileNamesSciptToPool{2} = 'script2.m'; % Replace script2 by the name of ...
    %your second file
```

```
% If you have more scripts corresponding to data to pool, add them here.

for i = 1:length(fileNamesSciptToPool)
    evalScript(fileNamesSciptToPool{i});
end

%%
% Now the pipeline will pool the replicates.
clearvars
close all

path = 'X'; % Enter instead of X the path, where all the files to pool ...
   %are located.

fileNameBase = [path 'X'] % Enter instead of X the name common to all ...
   %the files to pool (the user might have called the files ...
   %for example: experiment_replicate1_Target.mat, ...
   %experiment_replicate2_Target.mat, experiment_replicate1_DarkROIs.mat, ...
   %experiment_replicate2_DarkROIs.mat. ...
   %In this case fileNameBase = 'experiment')

% Now all the files with Target.mat at the end of their names will be ...
   %pooled as data for target locations, and all the files with ...
   %DarkROIs.mat at the end of their names will be pooled as data for ...
   %dark locations. If the user didn't use the same procedure in naming ...
   %files, he should change the two following lines accordingly.

filesDark = dir([fileNameBase '*DarkROIs.mat']);
filesExp = dir([fileNameBase '*Target.mat']);

for i = 1:length(filesExp)
   experimentFiles{i} = [path filesExp(i).name];
   experimentFilesDarkROIs{i} = [path filesDark(i).name];
end

h2 = createParentFigure;
htabgroup2 = uitabgroup(h2);
gepObj2 = guiExperimentPooling(htabgroup2,experimentFiles);
gepdrObj2 = guiExperimentPoolingDarkROI(htabgroup2,experimentFilesDarkROIs);
gdcorObj = guiDarkCorrector(htabgroup2,gepObj2,gepdrObj2);

% At this point, the user can look at fluorescence intensity time traces, ...
%plot rastergrams and perform correction for non-specific binding via the ...
%interface of the pipeline.
```

TROUBLESHOOTING

In this section, we provide examples of what may go wrong at different steps of the analysis flow and how errors may be fixed.

Camera calibration

The Curve of Variance plotted as function of intensity should be linear (Fig. 6). If the curve is saturated at high intensity values (Fig. 39a), the camera gains can still be estimated from the linear portion of the curve by indicating the linear range upon data loading (Fig. 39b).

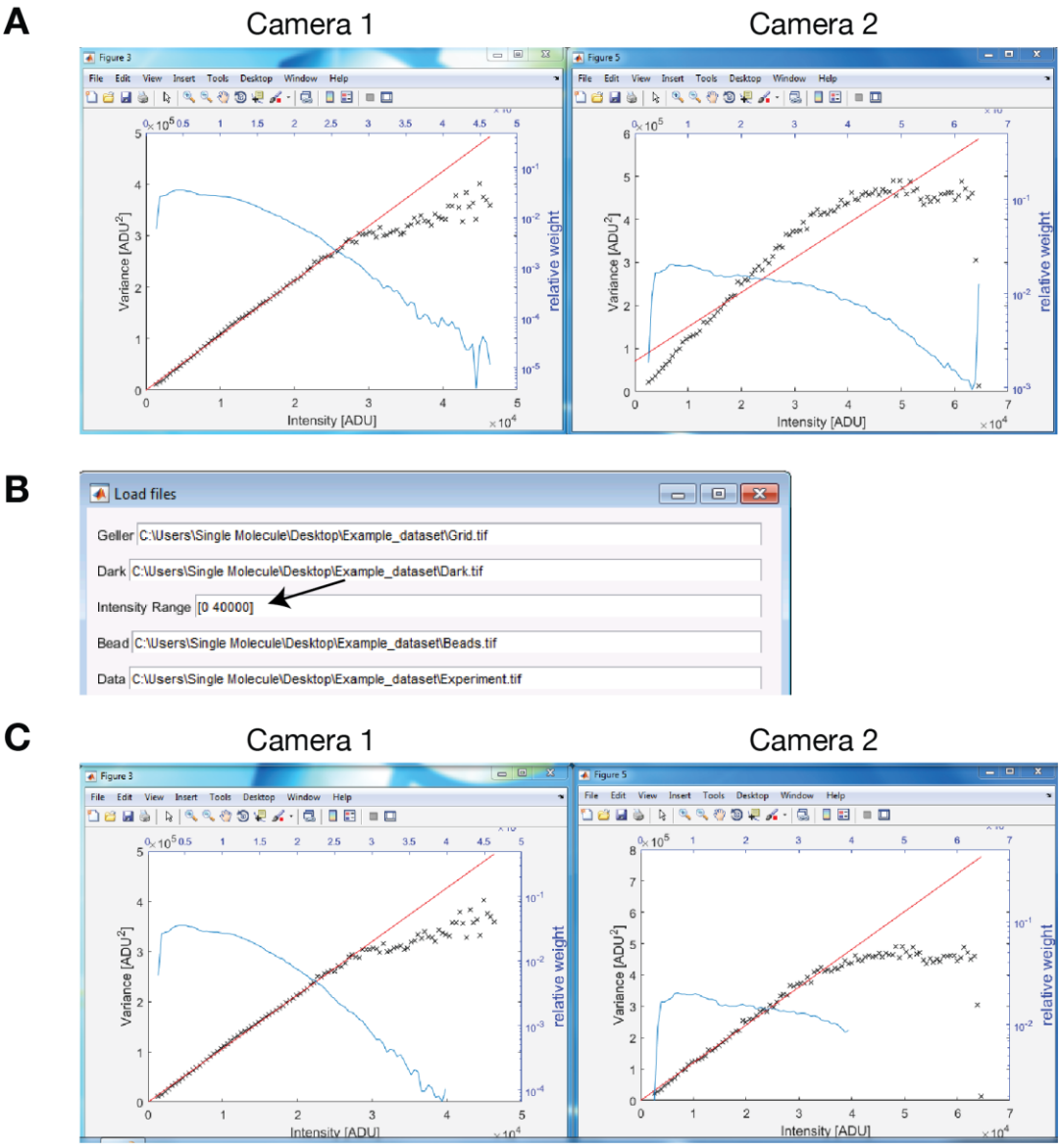


Figure 39. Camera calibration. (A) Variance is plotted as function of intensity for each camera. Slope of this curve is the gain of each camera. Number of events for each intensity value is plotted in blue. Curves are not linear at high values of intensities, and gain of the camera 2 was not correctly estimated. (B) Setting the intensity range for gain estimation at min = 0 and max = 40,000. (C) Curves were fitted using new intensity range, and the gain for camera 2 could be estimated.

If the curve does not have the expected shape (Fig. 40), you should acquire a new calibration dataset.

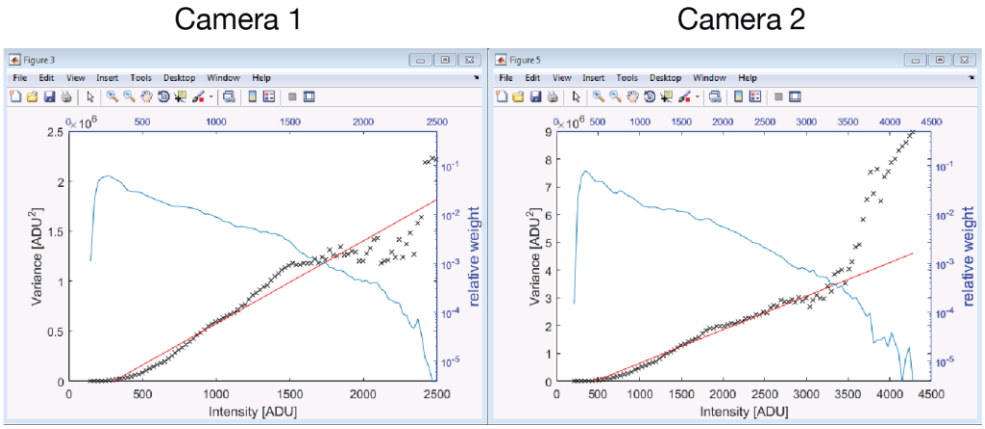


Figure 40. Example of an unsuitable calibration dataset. Variance is plotted as function of intensity for each camera. The number of events for each intensity value is plotted in blue. The curves are non-linear at both low and high intensity values.

Camera alignment

Even if the camera alignment appears to be successful, as evidenced by a superimposed image of beads from both cameras and low alignment score (Fig. 41a), a sub-overlay movie of the experimental dataset may clearly show the contrary (Fig. 41b). This is most likely due to a high number of large beads with a high intensity, and we suggest using another camera registration dataset.

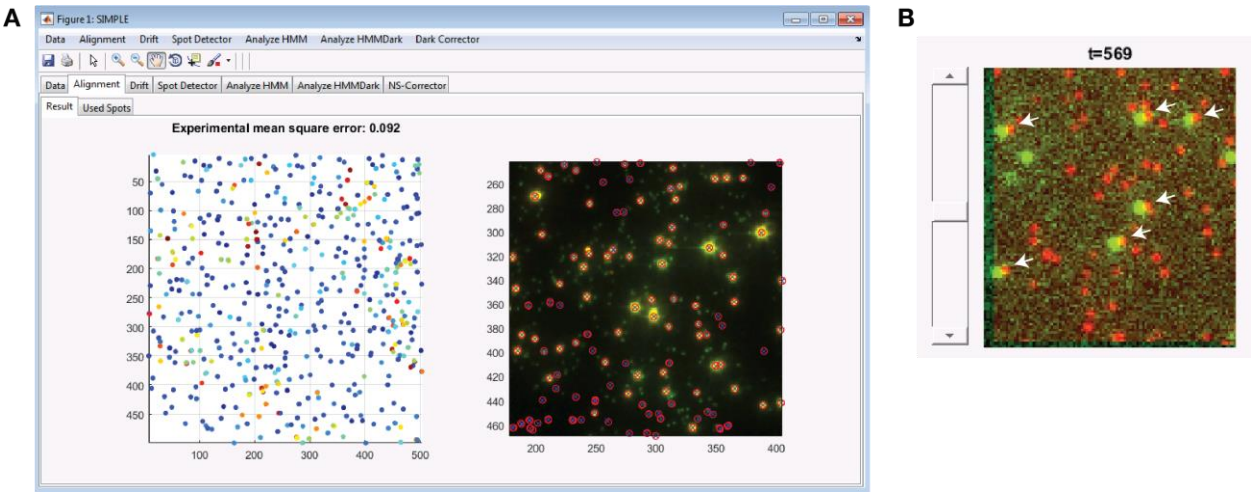


Figure 41. Alignment of images from multiple wavelength cameras. (A) Right panel displays a superimposed image of beads from both cameras. Left panel indicates alignment score for each bead. Blue denotes lower values, whereas red corresponds to higher values. (B) An overlay of the two cameras is created for the experimental dataset. Frame 569 is displayed as an example. White arrows point at target locations where TtAgo-guide complex binds but both cameras do not appear to co-localize, suggesting that camera alignment was not performed correctly.

Drift correction

Even if drift correction appears to be successful, as evidenced by overlapping frames *t* and *t+1* (yellow signal) (Fig. 42a), the second quality control step at target locations may show the contrary (Fig. 42b). For a more accurate estimation, we suggest selecting a bigger area with several bright individual spots.

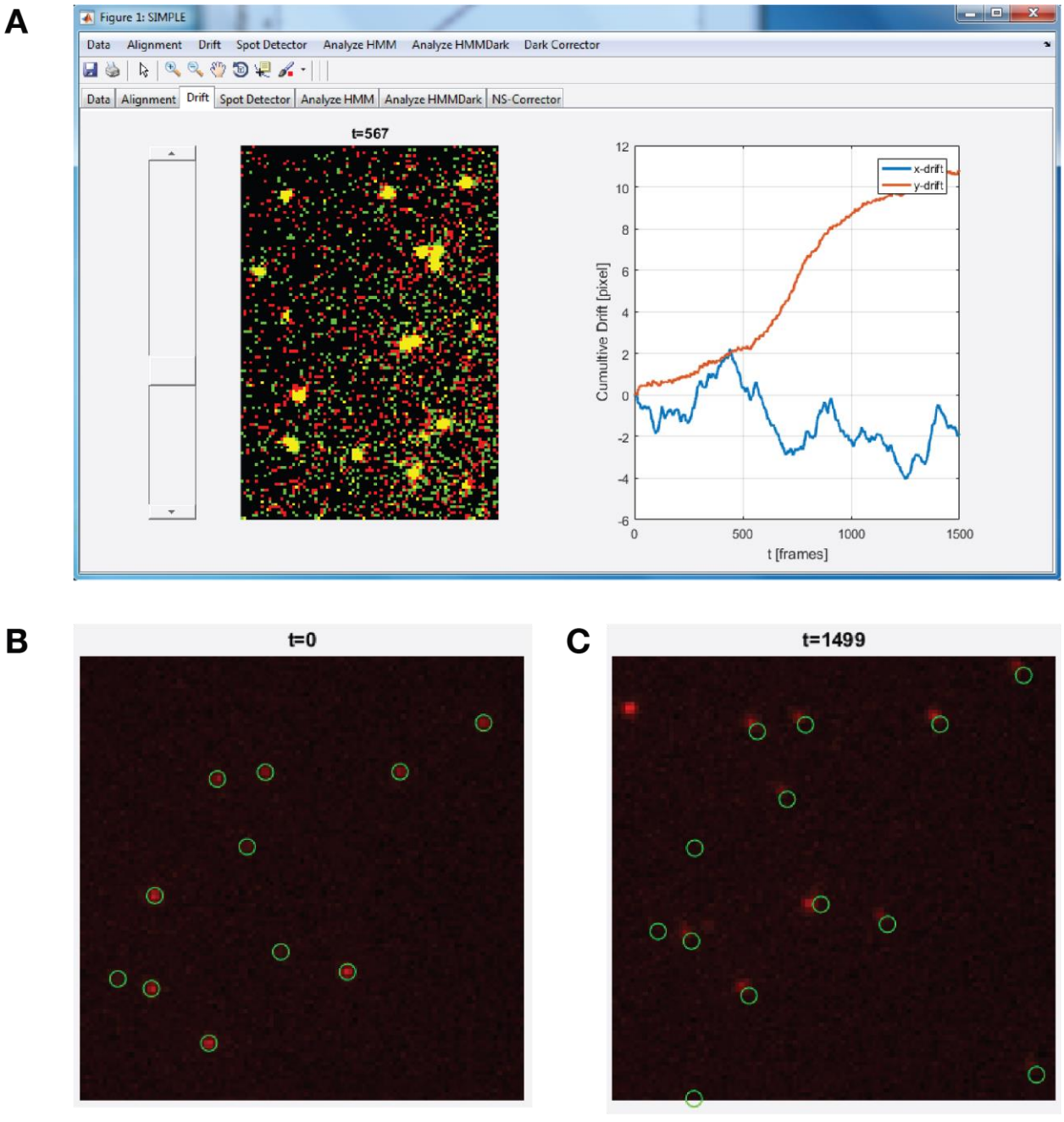


Figure 42. Quality control for drift correction. (A) Drift correction from frame t to frame t+1 resulted in yellow signal, suggesting that it was successful; however, the target locations were within the green circles in the first frame (B) but not in the last frame (C), indicating that the drift correction was underestimated.

Spot Detection

Incorrect detection of target locations essentially results from a high density of target molecules on the slide (Fig. 43). We advise performing experiments with a lower density of molecules. However, misinterpreted target locations can be successfully filtered, in particular according to their circularity (defined by the ratio of two perpendicular diameters) and cluster size (Fig. 44).

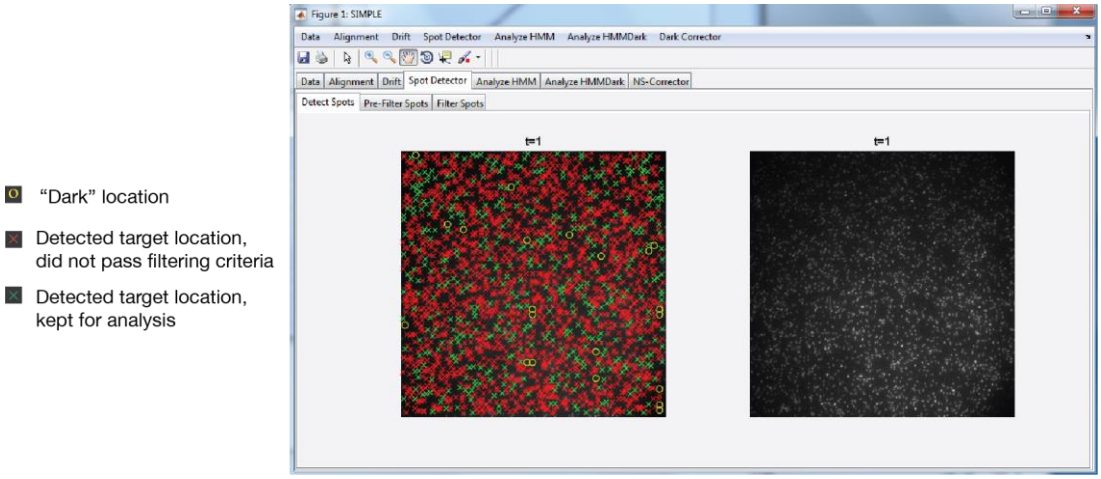


Figure 43. Example of an experimental dataset with high density of target locations.

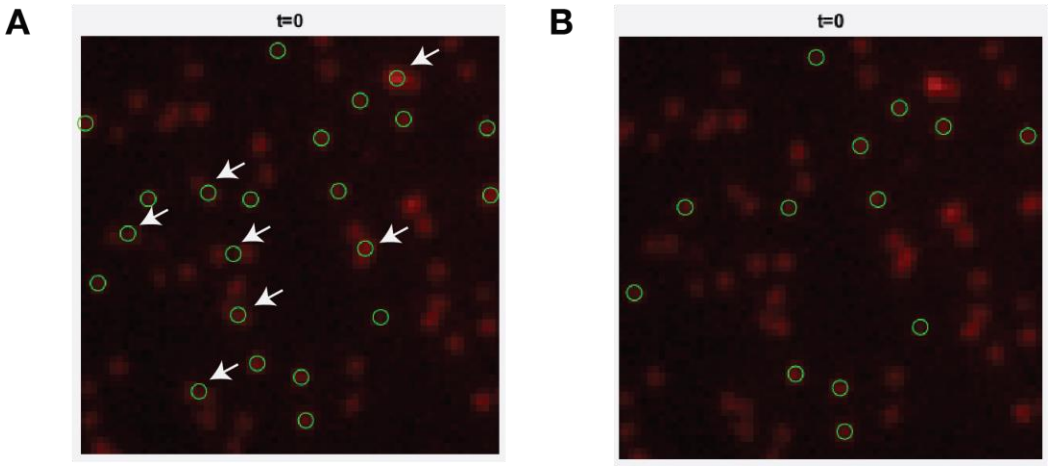


Figure 44. Example of misinterpreted target locations and their removal from the analysis. (A) Some detected locations (green circles) do not correspond to target molecules (white arrows). (B) Filtering by circularity (min = 0.8 and max = 1.45) and cluster size (min = 0 and max = 20) removed the misinterpreted target locations.

No binding detected

If no binding events were detected and the user has the impression from fluorescence intensity time traces that there should be binding, three checks can be done. First, verify that default threshold parameters for binding events were changed to parameters appropriate for your dataset. Second, verify that the Drift correction was performed correctly (Troubleshooting – Drift correction). Finally, verify that multiple cameras were aligned successfully (Troubleshooting – Camera alignment).

Low signal

In the example experiment TtAgo binds on a target with long dwell times, therefore imaging speed of 5 frames/s was fast enough to capture these binding events. If protein of user's interest displays faster dissociation rates, the user needs to decrease the exposure time, which might result in lower fluorescent signal.

We imaged TtAgo:guide binding a fully complementary DNA target with different exposure times, while maintaining the same laser power (Fig. 45). The pipeline successfully identified the locations of target molecules and of mobile components and detected co-localization events.

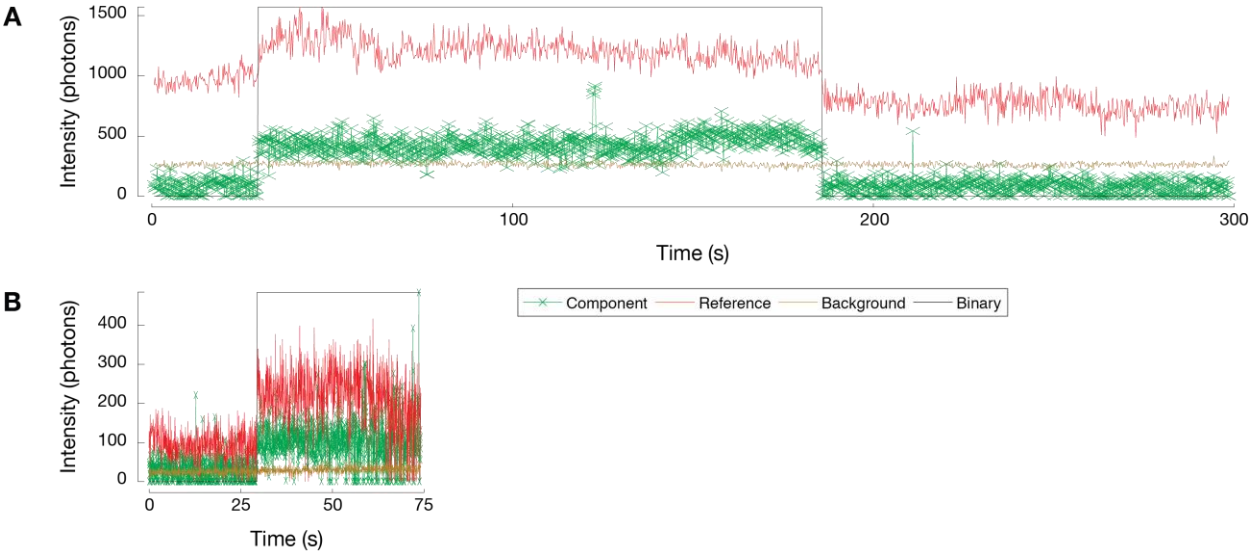


Figure 45. Analyzing datasets, which were acquired with different imaging speed. Exposure time was decreased from 200 ms (A) to 50 ms (B), while laser power was maintained the same. This yielded lower photon counts for both red and green fluorescent signals, as shown by representative fluorescence intensity time traces. Acquisition of the second dataset was stopped after 75 s (B).

High background

If the user has chosen to use the "Number of frames to group" feature, the signal-to-noise ratio might decrease (Fig. 46), and some binding events may no longer pass the threshold signal-to-background ratio > 1 used in the example. If this is the case, the user may choose to decrease the signal-to-background ratio threshold, or not use "Number of frames to group" feature.

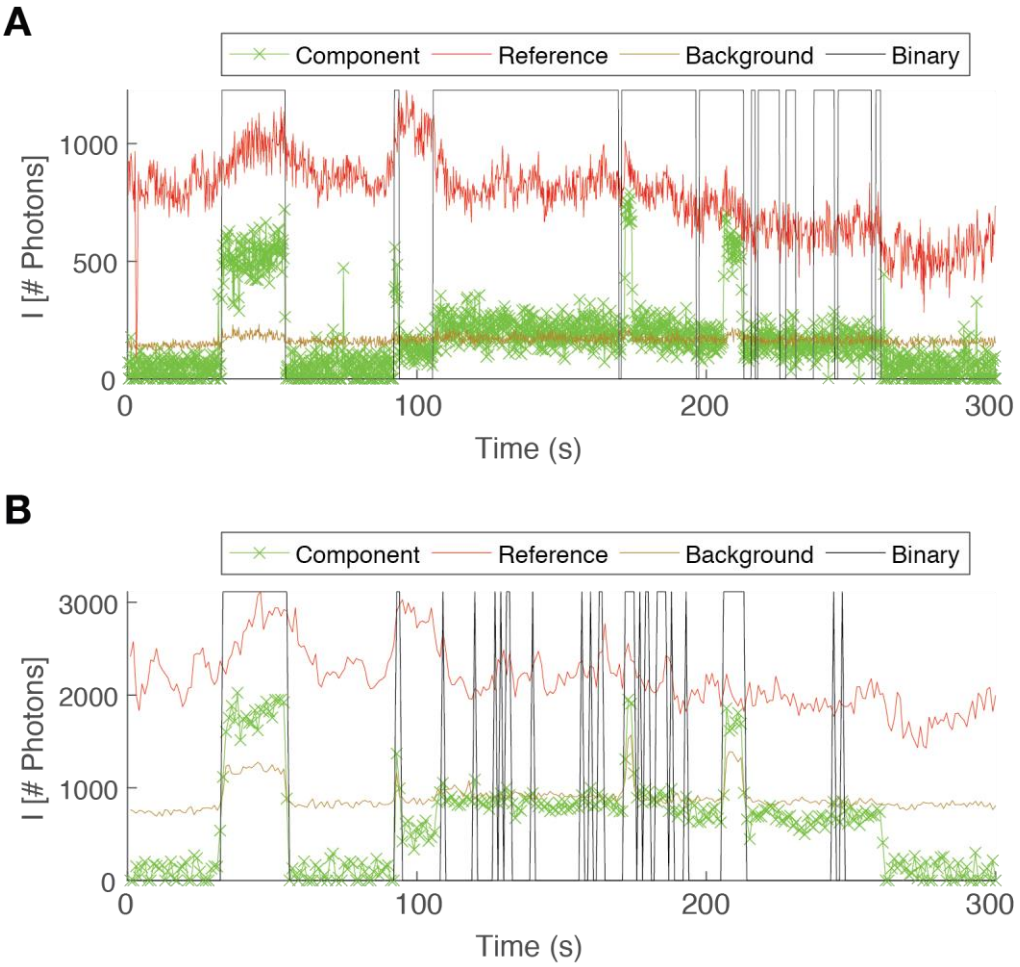


Figure 46. Using "Number of frames to group" feature tends to decrease the signal-to-background ratio. The experimental dataset was analyzed with the same parameters, except "Number of frames to group" was set to 1 in (A) and to 5 in (B). Fluorescence intensity time trace is displayed for the same target location.

Some experimental conditions, e.g., using higher concentration of the mobile component, might result in intrinsically high background (Fig. 47). In this case, the user might need to adjust the signal-to-background ratio in order to detect co-localization events.

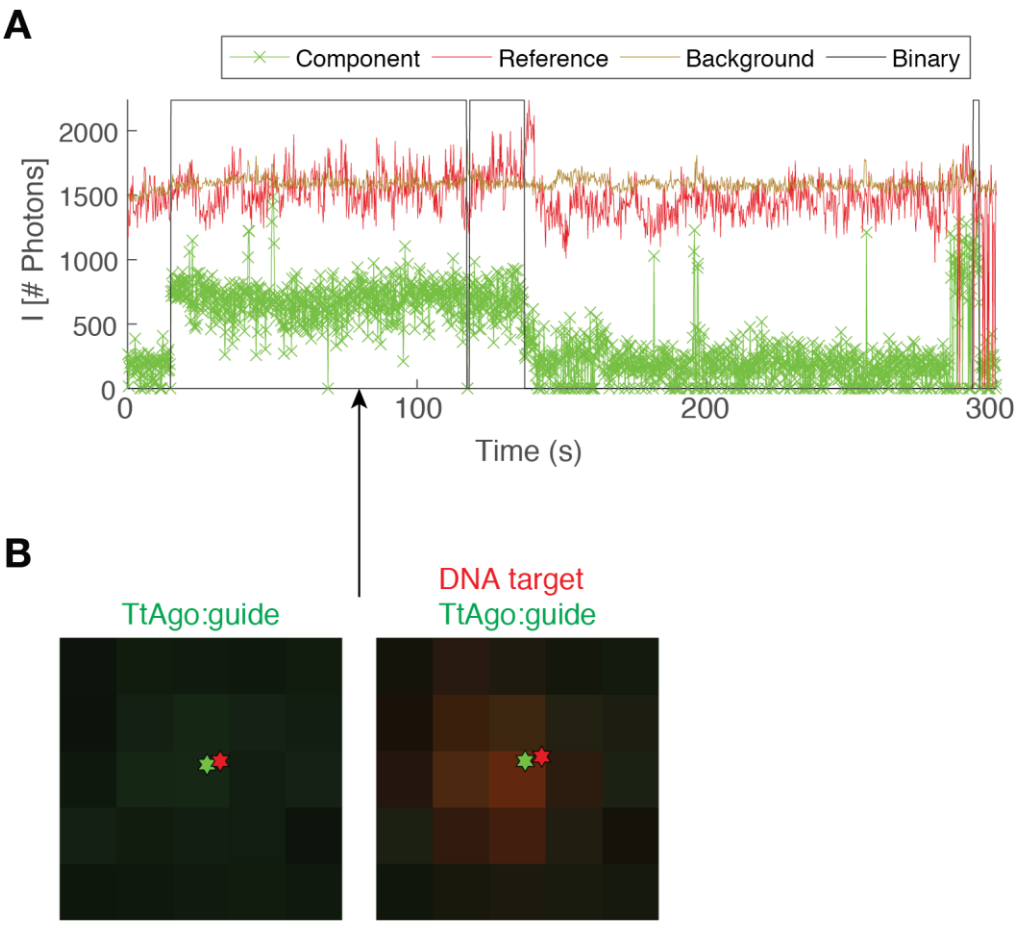


Figure 47. Analyzing datasets with high intrinsic background. (A) Fluorescence intensity time trace displaying high background. The signal-to-background ratio was lowered to 0.3 in order to detect binding events. (B) Images of TtAgo:guide and DNA target molecules observed at t = 80 s, as indicated by a black arrow in the fluorescence intensity time trace.

Correction for non-specific binding

When non-specific binding is very low, the small number of "dark" locations bound by the mobile component is insufficient to obtain a robust correction (Fig. 48). When only little data exist, estimation of non-specific binding is more robust by use of the Cumulative Density Function (CDF). The drawback is that no asymptotic optimality condition exists for CFD-based estimation. Therefore, in the presence of enough data it is better to correct for non-specific binding by using the Probability Density Function (PDF).

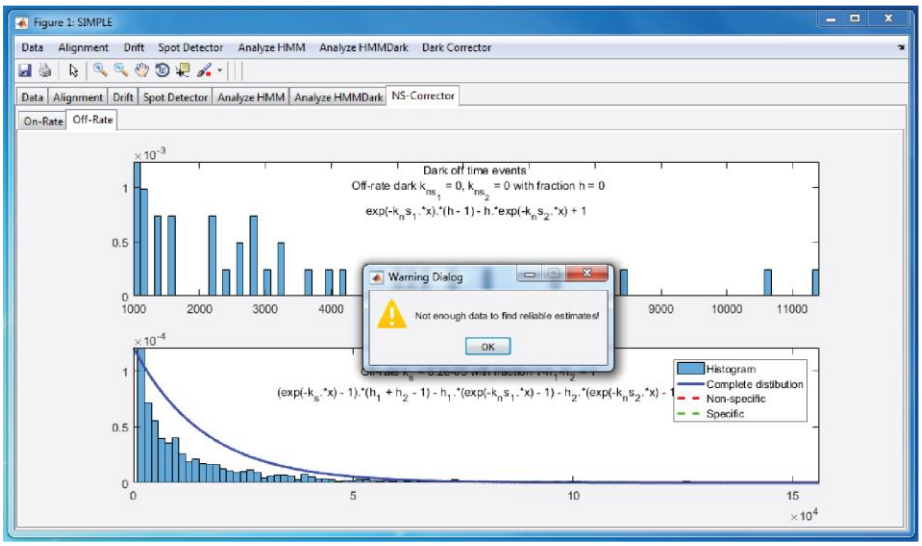


Figure 48. Correction of off-rate for non-specific binding by fitting the probability density function. Low number of “hot” locations was insufficient and lead to an error message.

APPENDIX 1: Derivation and implementation of mathematical concepts

For the next general derivation, we assume that θ is a vector that contains all the parameters that need to be estimated from a matrix with the measured data x. In the context of this paper x is a matrix that contains signal fluorescence and background fluorescence for all the targets at all their moments in time. To stabilize the estimation problem and to perform robust selection of the model order we will work in the framework of evidence maximization.

Maximum evidence estimation

The aim is to maximize the evidence towards θ

$$\ln\big(p(x)\big) = \ln\left(\frac{p(x,\theta)}{p(\theta|x)}\right).$$

Using $\int q(\theta)d\theta = 1$, we can write that

$$\begin{aligned}
&\int q(\theta)\ln\big(p(x)\big)\,d\theta = \int q(\theta)\ln\left(\frac{p(x,\theta)}{p(\theta|x)}\right)d\theta,\\
&= \int q(\theta)\ln\left(\frac{p(x,\theta)}{p(\theta)}\frac{q(\theta)}{q(\theta)}\right)d\theta\,,\\
&= \int q(\theta)\ln\left(\frac{q(\theta)}{p(\theta|x)}\right)d\theta + \int q(\theta)\ln\left(\frac{p(x,\theta)}{q(\theta)}\right)d\theta\,,\\
&\geq \int q(\theta)\ln\left(\frac{p(x,\theta)}{q(\theta)}\right)d\theta\,,\\
&\equiv L[q,p].
\end{aligned}$$

For ease of later notation we assume that for all random variables we can write that $q(\theta) = \prod_i q(\theta_i)$, and that $p(x,\theta) = p(x|\theta)\prod_j p\big(\theta_j|\theta_{\setminus j}\big)$. This enables us to rewrite $L[q,p]$ in the following way:

$$\begin{aligned}
&L[q,p] = \int q(\theta)\ln\left(\frac{p(x,\theta)}{q(\theta)}\right)d\theta\,,\\
&= \int \prod_{i\in N} q(\theta_i)\left(\sum_{j\in N}\ln\Big(p\big(x,\theta_j|\theta_{\setminus j}\big)\Big) - \sum_{k\in N}\ln\big(q(\theta_k)\big)\right)d\theta,\\
&= \sum_{j\in N}\int \prod_{i\in N} q(\theta_i)\Big(\ln\Big(p\big(x,\theta_j|\theta_{\setminus j}\big)\Big) - \ln\Big(q\big(\theta_j\big)\Big)\Big)\,d\theta,\\
&= \sum_{j\in N}\int q\big(\theta_j\big)\left(\int \prod_{i\in N\setminus j} q(\theta_i)\ln\Big(p\big(x,\theta_j|\theta_{\setminus j}\big)\Big)\,d\theta_i - \ln\Big(q\big(\theta_j\big)\Big)\right)d\theta_j,\\
&= \sum_{j\in N}\int q\big(\theta_j\big)\Big(\mathbb{E}_{q(\theta_{\setminus j})}\Big[\ln\Big(p\big(x,\theta_j|\theta_{\setminus j}\big)\Big)\Big] - \ln\Big(q\big(\theta_j\big)\Big)\Big)\,d\theta_j\quad.
\end{aligned}$$

The find a solution $q^*(\theta_j)$ that maximizes $L[q,p]$ we take the derivative

$$\begin{aligned}
&\nabla_{q(\theta_j)} L[q,p] = 0, \\
&= \nabla_{q(\theta_j)} \sum_{j \in N} \int q(\theta_j) \left(\mathbb{E}_{q(\theta_{\setminus j})} \left[\ln \left(p(x, \theta_j | \theta_{\setminus j}) \right) \right] - \ln \left(q(\theta_j) \right) \right) d\theta_j, \\
&= \mathbb{E}_{q(\theta_{\setminus j})} \left[\ln \left(p(x, \theta_j | \theta_{\setminus j}) \right) \right] - \ln \left(q(\theta_j) \right) + const.
\end{aligned}$$

This gives us the solution

$$q^*(\theta_j) = \frac{1}{Z_{\theta_j}} \exp \mathbb{E}_{q(\theta_{\setminus j})} \left[\ln \left(p(x, \theta_j | \theta_{\setminus j}) \right) \right],$$

where the integration constant is given by $Z_{\theta_j} = \int \exp \mathbb{E}_{q(\theta_{\setminus j})} \left[\ln \left(p(x, \theta_j | \theta_{\setminus j}) \right) \right] d\theta_j$ and $const = \ln(Z_{\theta_j})$.

Gaussian mixture model

We assume that this signal and background fluorescence of the $k-th$ binding state (i.e. k number of bound molecules) can be modelled by multi-variate Gaussian distribution $p(x_n|\mu_k, \lambda_k) = N(\mu_k, \lambda_k)$, where x_n is the $n-th$ measurement, and the μ_k is the mean and λ_k is the variance. The complete data will therefore be a mixture of K different multi-variate Gaussians. The goal is to estimate the set of multi-variates Gaussian, while we do not know how many multi-variate Gaussians there are [8]. The transition between each state, which can be modelled with a Hidden Markov Model (HMM), is left outside of the derivation in this section. This will be added in the next section with the aim to incrementally build up the complexity.

The probability distribution of several observations from a Gaussian mixture is given by

$$P(x, z, \pi, \lambda, \mu) = \; p(x|z, \mu, \lambda) p(z|\pi) p(\pi) p(\mu|\lambda) p(\lambda),$$

with the observation of the k mixtures as

$$p(x|z, \mu, \lambda) = \; \prod_k \prod_n p(x_n|\mu_k, \lambda_k)^{z_{nk}},$$

the probability of the observation being from each mixture

$$p(z|\pi) = \prod_k \prod_n p(z_{nk} = 1|\pi_k)^{z_{nk}},$$

the prior distributions on the parameters π μ, and λ to enable us to calculate the evidence

$$\begin{aligned}
p(\pi) &= p(\pi_1, \dots, \pi_K), \\
p(\mu|\lambda) &= \prod_k p(\mu_k \, | \mathrm{m}_0, (\beta_0 \lambda_\mathrm{k})^{-1}),
\end{aligned}$$

$$p(\lambda) = \prod_k p(\lambda_k | W_0, v_0).$$

where $n \in N$ is an observation a $k \in K$ mixture. Here for the Gaussian mixture model we chose a multi-variate Gaussian for the observations

$$p(x_n | \mu_k, \lambda_k) = N(\mu_k, \lambda_k),$$

and a discrete mixture probability

$$p(z_{nk} = 1 | \pi_k) = \pi_k.$$

To enable the calculation of the posterior probabilities we chose the corresponding conjugate priors Dirichlet distribution, multivariate Gaussian and a Wishart distribution, respectively

$$p(\pi_1, \dots, \pi_K) = Dir(\pi | \alpha_0),$$

$$\mathrm{p}(\mu_\mathrm{k} | \mathrm{m}_0, (\beta_0 \lambda_\mathrm{k})^{-1}) = N(\mu_\mathrm{k} | \mathrm{m}_0, (\beta_0 \lambda_\mathrm{k})^{-1}),$$

$$p(\lambda_k | W_0, v_0) = W(\lambda_k | W_0, v_0).$$

Given these equations we can find the update rules for our VBEM algorithm

$$\ln\left(q^*(\theta_j)\right) = \mathbb{E}_{q(\theta_{\setminus j})}\left[\ln\left(p(x, \theta_j | \theta_{\setminus j})\right)\right] + \ln\left(Z_{\theta_j}\right).$$

For

$$p(x, z, \theta) = p(x | z, \theta) p(z | \theta) \prod_j p(\theta_j | \theta_{\setminus j}),$$

we find

$$\ln\big(p(x, z, \theta)\big) = \ln\big(p(x | z, \mu, \lambda)\big) + \ln\big(p(z | \pi)\big) + \ln\big(p(\pi)\big) + \ln\big(p(\mu | \lambda)\big) + \ln\big(p(\lambda)\big),$$

and therefore, we write the following factorization

$$q^*(z, \theta) = q^*(z) q^*(\theta_\pi) q^*\big(\theta_{\mu|\lambda}\big) q^*(\theta_\lambda),$$

which gives us the following expectations to calculate

$$\ln\big(q^*(z)\big) \propto \mathbb{E}_{q(\theta_{\setminus z})}\big[\ln\big(p(x | z, \mu, \lambda)\big) + \ln\big(p(z | \pi)\big)\big],$$
$$\ln\big(q^*(\theta_\pi)\big) \propto \mathbb{E}_{q(\theta_{\setminus \pi})}\big[\ln\big(p(z | \pi)\big) + \ln\big(p(\pi)\big)\big],$$
$$\ln\left(q^*\big(\theta_{\mu|\lambda}\big)\right) \propto \mathbb{E}_{q(\theta_{\setminus \mu|\lambda})}\big[\ln\big(p(x | z, \mu, \lambda)\big) + \ln\big(p(\mu | \lambda)\big)\big].$$

To obtain $\ln\big(q^*(\theta_\lambda)\big)$ the problem needs to formulated in terms of $\ln\left(q^*\big(\theta_{\mu,\lambda}\big)\right)$

$$\ln\big(q^*(\theta_\lambda)\big) = \ln\left(q^*\big(\theta_{\mu,\lambda}\big)\right) - \ln\left(q^*\big(\theta_{\mu|\lambda}\big)\right),$$

where

$$\ln\left(q^*(\theta_{\mu,\lambda})\right) \propto \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\left[\ln\left(p(x|z,\mu,\lambda)\right) + \ln\left(p(\mu|\lambda)\right) + \ln\left(p(\lambda)\right)\right].$$

The first term

$$\begin{aligned}
&\ln\left(q^*(\theta_z)\right) \propto \mathbb{E}_{q(\theta_{\setminus z})}\left[\ln\left(p(x|z,\mu,\lambda)\right) + \ln\left(p(z|\pi)\right)\right],\\
&= \mathbb{E}_{q(\theta_{\setminus z})}\left[\ln\left(\prod_k\prod_n p(x_n|\mu_k,\lambda_k)^{z_{nk}}\right) + \ln\left(\prod_k\prod_n p(z_{nk}=1|\pi_k)^{z_{nk}}\right)\right],\\
&= \mathbb{E}_{q(\theta_{\setminus z})}\left[\sum_k\sum_n z_{nk}\ln\left(p(x_n|\mu_k,\lambda_k)\right) + \sum_k\sum_n z_{nk}\ln\left(p(z_{nk}=1|\pi_k)\right)\right],\\
&= \mathbb{E}_{q(\theta_{\setminus z})}\left[\sum_k\sum_n z_{nk}\left\{\frac{1}{2}\ln(|\lambda_k|) - \frac{D}{2}\ln(2\pi) - \frac{1}{2}(x_n-\mu_k)^T\lambda_k\ (x_n-\mu_k)\right\} + \sum_k\sum_n z_{nk}\ln(\pi_k)\right].
\end{aligned}$$

We find that

$$\ln\left(q^*(\theta_z)\right) = \sum_k\sum_n z_{nk}\left(\ln\left(\rho_{n,k}\right) - \ln(\sum_k \rho_{n,k})\right),$$

with

$$\rho_{n,k} = \exp\left(\mathbb{E}_{q(\theta_{\setminus z})}\left[\frac{1}{2}\ln(|\lambda_k|) - \frac{D}{2}\ln(2\pi) - \frac{1}{2}(x_n-\mu_k)^T\lambda_k\ (x_n-\mu_k) + \ln(\pi_k)\right]\right).$$

Next,

$$\begin{aligned}
&\ln\left(q^*(\theta_\pi)\right) \propto \mathbb{E}_{q(\theta\ \setminus\pi)}\left[\ln\left(p(z|\pi)\right) + \ln\left(p(\pi)\right)\right],\\
&= \mathbb{E}_{q(\theta\ \setminus\pi)}\left[\ln\left(\prod_k\prod_n p(z_{nk}=1|\pi_k)^{z_{nk}}\right) + \ln\left(\prod_k p(\pi_k|\alpha_0)\right)\right],\\
&= \mathbb{E}_{q(\theta\ \setminus\pi)}\left[\sum_k\sum_n z_{nk}\ln\left(p(z_{nk}=1|\pi_k)\right) + \sum_k \ln\left(p(\pi_k|\alpha_0)\right)\right],\\
&= \mathbb{E}_{q(\theta\ \setminus\pi)}\left[\sum_k\sum_n z_{nk}\ln(\pi_k) - C(\alpha_0) + (\alpha_0-1)\sum_k\ln(\pi_k)\right],
\end{aligned}$$

where $C(\alpha_0)$ is the normalization constant for the Dirichlet distribution of π. We find

$$\ln\left(q^*(\theta_\pi)\right) \propto \mathbb{E}_{q(\theta\ \setminus\pi)}\left[\sum_k(\alpha_k-1)\ln(\pi_k)\right],$$

with

$$\alpha_k = \alpha_0 + \sum_n \rho_{nk}.$$

As expected since Categorical is conjugate to Dirichlet, which is again a Dirichlet.

Next,

$$
\begin{aligned}
&\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\big[\ln\big(p(x|z,\mu,\lambda)\big)+\ln\big(p(\mu|\lambda)\big)\big]\\
&=\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\ln\left(\prod_k\prod_n p(x_n|\mu_k,\lambda_k)^{z_{nk}}\right)+\ln\left(\prod_k p(\mu_k\,|\mathrm{m}_0,(\beta_0\lambda_\mathrm{k})^{-1})\right)\right]\\
&=\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_n\sum_k z_{n,k}\left\{\frac{1}{2}\ln(|\lambda_\mathrm{k}|)-\frac{D}{2}\ln 2\pi-\frac{1}{2}(x_n-\mu_k)^T\lambda_k\ (x_n-\mu_k)\right\}\right.\\
&\qquad\left.+\sum_k\left\{\frac{1}{2}\ln(|\beta_0\lambda_\mathrm{k}|)-\frac{D}{2}\ln 2\pi-\frac{1}{2}(\mu_\mathrm{k}-\mathrm{m}_0)^T\lambda_k\beta_0\ (\mu_\mathrm{k}-m_0)\right\}\right]\\
&=\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_k\left\{-\frac{1}{2}(\mu_\mathrm{k}-\mathrm{m}_0)^T\lambda_k\beta_0\ (\mu_\mathrm{k}-m_0)-\sum_n z_{n,k}\frac{1}{2}(x_n-\mu_k)^T\lambda_k\ (x_n-\mu_k)\right\}\right]+c\\
&=\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_k\left\{\frac{1}{2}\left(\beta_0+\sum_n z_{n,k}\right)\mu_k^T\lambda_k\mu_k+\mu_k^T\lambda_k\left(\beta_0 m_0+\sum_n z_{n,k}\,x_n\right)\right\}\right]+c\\
&=\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_k\frac{1}{2}\left(\beta_0+\sum_n z_{n,k}\right)\left\{\ \mu_k^T\lambda_k\mu_k+2\mu_k^T\lambda_k\left(\beta_0 m_0+\sum_n z_{n,k}\,x_n\right)\left(\beta_0+\sum_n z_{n,k}\right)^{-1}\right\}\right]+c\\
&\equiv\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_k\frac{1}{2}\beta_k\{\ \mu_k^T\lambda_k\mu_k+2\mu_k^T\lambda_k m_k\}\right]+c\\
&=\mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_k\frac{1}{2}(\mu_k-m_k)^T\beta_k\lambda_k(\mu_k-m_k)\quad\right]+c
\end{aligned}
$$

We find

$$
\ln\left(q^*\left(\theta_{\mu|\lambda}\right)\right)\propto\mathbb{E}_{q(\theta_{\setminus\mu})}\left[\sum_k\frac{1}{2}(\mu_k-m_k)^T\beta_k\lambda_k(\mu_k-m_k)\quad\right]
$$

With

$$
\begin{aligned}
m_k&=\left(\beta_0 m_0+\sum_n\rho_{n,k}\,x_n\right)\left(\beta_0+\sum_n\rho_{n,k}\right)^{-1}\\
\beta_k&=\beta_0+\sum_n\rho_{n,k}
\end{aligned}
$$

As expected since Normal is conjugate to Normal in the mean, which is again a Normal distribution. For the last term we first need to calculate

$$
\begin{aligned}
&\mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\big[\ln\big(p(x|z,\mu,\lambda)\big)+\ln\big(p(\mu|\lambda)\big)+\ln\big(p(\lambda)\big)\big]\\
&=\mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\left[\ln\left(\prod_k\prod_n p(x_n|\mu_k,\lambda_k)^{z_{nk}}\right)+\ln\left(\prod_k p(\mu_k\,|\mathrm{m}_0,(\beta_0\lambda_\mathrm{k})^{-1})\right)+\ln\left(\prod_k p(\lambda_k|W_0,v_0)\right)\right]
\end{aligned}
$$

$$= \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\Bigg[\sum_n \sum_k z_{n,k} \left\{ \frac{1}{2}\ln(|\lambda_k|) - \frac{D}{2}\ln 2\pi - \frac{1}{2}(x_n - \mu_k)^T \lambda_k (x_n - \mu_k) \right\} + \sum_k \left\{ \frac{D}{2}\ln(|\beta_0|) + \frac{1}{2}\ln(|\lambda_k|) - \frac{D}{2}\ln 2\pi - \frac{1}{2}(\mu_k - m_0)^T \beta_0 \lambda_k \ (\mu_k - m_0) \right\} + \sum_k \frac{v_0 - D - 1}{2}\ln(|\lambda_k|) - \frac{1}{2}Tr(W_0^{-1}\lambda_k) \Bigg]$$

The conjugate of a Normal is a Normal-Wishart and is again a Normal-Wishart. Therefore, combining

$$\ln\left(q^*(\theta_{\mu,\lambda})\right) - \ln\left(q^*(\theta_{\mu|\lambda})\right)$$

$$\propto \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\Bigg[\sum_n \sum_k z_{n,k} \left\{ \frac{1}{2}\ln(|\lambda_k|) - \frac{D}{2}\ln 2\pi - \frac{1}{2}(x_n - \mu_k)^T \lambda_k (x_n - \mu_k) \right\} + \sum_k \left\{ \frac{D}{2}\ln(|\beta_0|) + \frac{1}{2}\ln(|\lambda_k|) - \frac{D}{2}\ln 2\pi - \frac{1}{2}(\mu_k - m_0)^T \beta_0 \lambda_k \ (\mu_k - m_0) \right\} + \sum_k \frac{v_0 - D - 1}{2}\ln(|\lambda_k|) - \frac{1}{2}Tr(W_0^{-1}\lambda_k) \Bigg] - \mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\left[\sum_k -\frac{1}{2}(\mu_k - m_k)^T \beta_k \lambda_k (\mu_k - m_k) + \frac{1}{2}\ln(|\lambda_k|) \ -\frac{D}{2}\ln 2\pi \ \right]$$

$$= \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\Bigg[\sum_n \sum_k z_{n,k} \left\{ \frac{1}{2}\ln(|\lambda_k|) - \frac{1}{2}(x_n - \mu_k)^T \lambda_k (x_n - \mu_k) \right\} + \frac{D}{2}\ln(|\beta_0|) - \frac{1}{2}(\mu_k - m_0)^T \beta_0 \lambda_k \ (\mu_k - m_0) + \frac{v_0 - D - 1}{2}\ln(|\lambda_k|) - \frac{1}{2}Tr(W_0^{-1}\lambda_k) + \frac{1}{2}(\mu_k - m_k)^T \beta_k \lambda_k (\mu_k - m_k) \Bigg] + c$$

$$= \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\Bigg[\frac{1}{2}\left((v_0 - D - 1) + \sum_n z_{n,k} \right)\ln(|\lambda_k|) - \sum_n z_{n,k}\frac{1}{2}(x_n - \mu_k)^T \lambda_k (x_n - \mu_k) - \frac{1}{2}(\mu_k - m_0)^T \beta_0 \lambda_k \ (\mu_k - m_0) + \frac{1}{2}(\mu_k - m_k)^T \beta_k \lambda_k (\mu_k - m_k) - \frac{1}{2}Tr(W_0^{-1}\lambda_k) \Bigg] + c$$

$$= \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\Bigg[-\frac{1}{2}Tr\Bigg[\Big\{ W_0^{-1} + \beta_0(\mu_k - m_0)(\mu_k - m_0)^T + \sum_n z_{n,k}(x_n - \mu_k)(x_n - \mu_k)^T - \beta_k(\mu_k - m_k)(\mu_k - m_k)^T \Big\} \lambda_k \Bigg] + \frac{1}{2}\left((v_0 - D - 1) + \sum_n z_{n,k} \right)\ln(|\lambda_k|) \Bigg] + c$$

$$\equiv \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\left[-\frac{1}{2}Tr\left[W_k^{-1}\lambda_k \right] + \frac{1}{2}(v_k - D - 1)\ln(|\lambda_k|) \right] + c$$

We find

$$\ln\left(q^*(\theta_\lambda)\right) \propto \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\left[-\frac{1}{2}Tr\left[W_k^{-1}\lambda_k\right] + \frac{1}{2}(v_k - D - 1)\ln(|\lambda_k|)\right]$$

With

$$W_k^{-1} = W_0^{-1} + \beta_0(\mu_k - m_0)(\mu_k - m_0)^T + \sum_n \rho_{n,k}(x_n - \mu_k)(x_n - \mu_k)^T - \beta_k(\mu_k - m_k)(\mu_k - m_k)^T$$

$v_k = v_0 + \sum_n \rho_{n,k}$

The conjugate of a Normal is a Normal-Wishart and is again a Normal-Wishart. Therefore, without the Normal on the mean we only keep the Wishart.

Calculation of expectations

$$\mathbb{E}_{q(\theta_\lambda)}[\ln(|\lambda_k|)] = \sum_D \psi\left(\frac{v_k + 1 - i}{2}\right) + D\ln(2) + \ln(|W_k|)$$
$$\equiv \mu_{\theta_\lambda}$$

Where ψ is the digamma function.

$$\mathbb{E}_{q(\theta_{\lambda,\mu})}[(x_n - \mu_k)^T\lambda_k(x_n - \mu_k)] = v_k(x_n - m_k)^T W_k(x_n - m_k) + D\beta_k^{-1}$$
$$\equiv \mu_{\theta_{\lambda,\mu}}$$

$$\mathbb{E}_{q(\theta_{\pi_k})}[\ln(\pi_k)] = \psi(\alpha_k) - \psi\left(\sum_k \alpha_k\right)$$
$$\equiv \mu_{\theta_{\pi_k}}$$

$$\mathbb{E}_{q(\theta_z)}[z_{nk}] = \frac{\rho_{n,k}}{\sum_k \rho_{n,k}}$$
$$\equiv \mu_{\theta_z}$$

Hidden Markov Model with a Multivariate Gaussian

In this section the transition between each binding state is incorporated into the model, which is done according to the framework of Hidden Markov Models (HMM). In HMMs the transitions are modelled by discrete probabilities and therefore given rise to a transition matrix A. The transition from state l to state k is given by the $k, l-th$ matrix element $A_{k,l}$.

The transition probabilities are added to the probability distribution of the observations from the previous section, please recall that the probability for a mixture of multi-variate distributions from the previous section was given by

$$P(x, z, \pi, \lambda, \mu) = \; p(x|z, \mu, \lambda)p(z|\pi)p(\pi)p(\mu|\lambda)p(\lambda).$$

This change by adding all the transition probabilities

$$P(x, z, \pi, \lambda, \mu) = \; p(z|\pi, A)p(A)p(\pi)p(x|z, \mu, \lambda)p(\mu|\lambda)p(\lambda),$$

where the state probabilities are

$$p(z|\pi, A) = p(z_1|\pi)\prod_{t>1} p(z_t|z_{t-1}, A),$$

with

$$p(z_t|z_{t-1}, A) = \prod_k \prod_l p\left(z_{t,k}\middle|z_{t-1,l}, A\right)^{z_{t,k}z_{t-1,l}}.$$

The initial state and the transition between each state is given by discrete probabilities, respectively:

$$p\left(z_{t,k}\middle|z_{t-1,l}, A\right) = A_{k,l},$$
$$p(z_{nk} = 1|\pi_k) = \pi_k,$$

where to enable the calculation of the posterior probabilities we chose (again) the corresponding conjugate

$$p(A) = \prod_k \prod_l p\left(A_{k,l}|\alpha_0\right),$$
$$p(A_k|\alpha_0) = Dir(A_k|\alpha_0),$$
$$p(\pi|\rho_0) = Dir(\pi|\rho_0).$$

The multi-variate mixture distributions from the previous section stay unchanged. However, now indexed by time (t) instead of the observation number (n) and the $k-th$ state instead of the k-th mixture:

$$p(\pi) = p(\pi_1, \dots, \pi_K|\rho_0),$$
$$p(z_1|\pi) = \prod_k p\left(z_{1,k} = 1\middle|\pi_k\right)^{z_{1,k}},$$
$$p(x|z, \mu, \lambda) = \; \prod_k \prod_t p(x_t|\mu_k, \lambda_k)^{z_{t,k}},$$

$$p(\mu|\lambda) = \prod_k p(\mu_k\,|\mathrm{m}_0, (\beta_0\lambda_\mathrm{k})^{-1}),$$
$$p(\lambda) = \prod_k p(\lambda_k|W_0, v_0).$$

Here for the hidden Markov model we chose

$$p(x_t|\mu_k, \lambda_k) = N(\mu_k, \lambda_k),$$
$$p(\mu_\mathrm{k}|\mathrm{m}_0, (\beta_0\lambda_\mathrm{k})^{-1}) = N(\mu_\mathrm{k}|\mathrm{m}_0, (\beta_0\lambda_\mathrm{k})^{-1}),$$
$$p(\lambda_k|W_0, v_0) = W(\lambda_k|W_0, v_0),$$

Again, we can obtain the update rules for our VBEM algorithm

$$\ln\left(q^*(\theta_j)\right) = \ \mathbb{E}_{q(\theta_{\setminus j})}\left[\ln\left(p(x, z, \theta_j|\theta_{\setminus j})\right)\right] + \ln\left(Z_{\theta_j}\right).$$

For

$$p(x, z, \theta) = p(x, z|\theta)\prod_j p(\theta_j|\theta_{\setminus j}).$$

we find

$$\ln(p(x, z, \theta)) = \ln\big(p(z|\pi, A)\big) + \ln\big(p(A)\big) + \ln\big(p(\pi)\big) + \ln\big(p(x|z, \mu, \lambda)\big) + \ln\big(p(\mu|\lambda)\big) + \ln\big(p(\lambda)\big)$$

and therefore, we write the following factorization

$$q^*(z, \theta) = q^*(z)q^*(\theta_A)q^*(\theta_\pi)q^*\big(\theta_{\mu|\lambda}\big)q^*(\theta_\lambda)$$

For which we need to calculate

$$\ln\big(q^*(z)\big) \propto \mathbb{E}_{q(\theta_{\setminus z})}\big[\ln\big(p(z|\pi, A)\big) + \ln\big(p(x|z, \mu, \lambda)\big)\big]$$
$$\ln\big(q^*(\theta_\pi)\big) \propto \mathbb{E}_{q(\theta_{\ \setminus \pi})}\big[\ln\big(p(z|\pi, A)\big) + \ln\big(p(\pi)\big)\big]$$
$$\ln\big(q^*(\theta_A)\big) \propto \mathbb{E}_{q(\theta_{\ \setminus A})}\big[\ln\big(p(z|\pi, A)\big) + \ln\big(p(A)\big)\big]$$
$$\ln\left(q^*\big(\theta_{\mu|\lambda}\ \big)\right) \propto \mathbb{E}_{q(\theta_{\setminus\mu|\lambda})}\big[\ln\big(p(x|z, \mu, \lambda)\big) + \ln\big(p(\mu|\lambda)\big)\big]$$
$$\ln\big(q^*(\theta_\lambda)\big) = \ \ln\left(q^*\big(\theta_{\mu,\lambda}\big)\right) - \ln\left(q^*\big(\theta_{\mu|\lambda}\big)\right)$$

with

$$\ln\left(q^*\big(\theta_{\mu,\lambda}\big)\right) \propto \mathbb{E}_{q(\theta_{\setminus(\mu,\lambda)})}\big[\ln\big(p(x|z, \mu, \lambda)\big) + \ln\big(p(\mu|\lambda)\big) + \ln\big(p(\lambda)\big)\big]$$

The first term

$$\mathbb{E}_{q(\theta_{\setminus z})}\big[\ln\big(p(z|\pi, A)\big) + \ln\big(p(x|z, \mu, \lambda)\big)\big] =$$
$$= \left[\ln\left(\prod_k p\big(z_{1,k} = 1\big|\pi_k\big)^{z_{1,k}}\prod_{t>1}\prod_k\prod_l p\big(z_{t,k}\big|z_{t-1,l}, A\big)^{z_{t,k}z_{t-1,l}}\right)\right.$$
$$\left. + \ln\left(\prod_t\prod_k p(x_t|\mu_k, \lambda_k)^{z_{t,k}}\right)\right]$$

$$= \mathbb{E}_{q(\theta_{\setminus z})}\Bigg[\sum_k z_{1,k} \ln\Big(p\big(z_{1,k}=1|\pi_k\big)\Big) + \sum_{t>1}\sum_k\sum_l z_{t,k}z_{t-1,l} \ln\Big(p\big(z_{t,k}|z_{t-1,l},A\big)\Big) + \sum_t\sum_k z_{t,k}\ln\big(p(x_t|\mu_k,\lambda_k)\big)\Bigg]$$

$$= \mathbb{E}_{q(\theta_{\setminus z})}\Bigg[\sum_k z_{1,k}\ln(\pi_k) + \sum_{t>1}\sum_k\sum_l z_{t,k}z_{t-1,l}\ln\big(A_{k,l}\big) + \sum_{t>1}\sum_k z_{t,k}\left\{\ln(\sqrt{|2\pi\lambda_k|}) - \frac{1}{2}(x_t-\mu_k)^T\lambda_k^{-1}\ (x_t-\mu_k)\right\}\Bigg]$$

We don't know what kind of distribution this is in closed form, but we can calculate the parameters we need, such as the expected values $\mathbb{E}_{q(\theta_z)}[.]$ using the forward backward algorithm.

The second term

$$\mathbb{E}_{q(\theta\ \setminus A)}\big[\ln\big(p(z|\pi,A)\big)+\ln\big(p(A)\big)\big]$$

$$= \mathbb{E}_{q(\theta\ \setminus A)}\left[\ln\left(\prod_k p\big(z_{1,k}=1|\pi_k\big)^{z_{1,k}}\prod_{t>1}\prod_k\prod_l p\big(z_{t,k}|z_{t-1,l},A\big)^{z_{t,k}z_{t-1,l}}\right) + \ln\left(\prod_k\prod_l p\big(A_{k,l}|\alpha_A\big)\right)\right]$$

$$= \mathbb{E}_{q(\theta\ \setminus A)}\left[\sum_k z_{1k}\ln\big(p(z_{1k}=1|\pi_k)\big) + \sum_{t>1}\sum_k\sum_l z_{t,k}z_{t-1,l}\ln\Big(p\big(z_{t,k}|z_{t-1,l},A\big)\Big) + \sum_k\sum_l p\big(A_{k,l}|\alpha_A\big)\right]$$

$$= \mathbb{E}_{q(\theta\ \setminus A)}\left[\sum_{t>1}\sum_k\sum_l z_{t,k}z_{t-1,l}\ln\big(A_{k,l}\big) + C(\alpha_A)(\alpha_A-1)\sum_k\sum_l\ln\big(A_{k,l}\big)\right] + c$$

$$= \mathbb{E}_{q(\theta\ \setminus A)}\left[\sum_k\sum_l\left\{C(\alpha_A)(\alpha_A-1)+\sum_{t>1}z_{t,k}z_{t-1,l}\right\}\ln\big(A_{k,l}\big)\right]$$

We find

$$\ln\big(q^*(\theta_A)\big) \propto \mathbb{E}_{q(\theta\ \setminus A)}\left[\sum_k\sum_l\left\{C(\alpha_A)(\alpha_A-1)+\sum_{t>1}z_{t,k}z_{t-1,l}\right\}\ln\big(A_{k,l}\big)\right]$$

The third term

$$\mathbb{E}_{q(\theta\ \setminus \pi)}\big[\ln\big(p(z|\pi,A)\big)+\ln\big(p(\pi)\big)\big]$$

$$= \mathbb{E}_{q(\theta\ \setminus \pi)}\left[\ln\left(\prod_k p(z_{1k}=1|\pi_k)^{z_{1k}}\prod_{t>1}\prod_k\prod_l p\big(z_{t,k}|z_{t-1,l},A\big)^{z_{t,k}z_{t-1,l}}\right) + \ln\left(\prod_k p(\pi_k|\alpha_\pi)\right)\right]$$

$$= \mathbb{E}_{q(\theta_{\setminus\pi})}\left[\sum_k z_{1k}\ln\big(p(z_{1k}=1|\pi_k)\big) + \sum_{t>1}\sum_k\sum_l z_{t,k}z_{t-1,l}\ln\Big(p\big(z_{t,k}\big|z_{t-1,l},A\big)\Big) + \sum_k \ln\big(p(\pi_k|\alpha_\pi)\big)\right]$$

$$= \mathbb{E}_{q(\theta_{\setminus\pi})}\left[\sum_k z_{1k}\ln\big(p(z_{1k}=1|\pi_k)\big) + \sum_k \ln\big(p(\pi_k|\alpha_\pi)\big)\right] + \mathrm{c}$$

$$= \mathbb{E}_{q(\theta_{\setminus\pi})}\left[\sum_k z_{1k}\ln(\pi_k) + C(\alpha_\pi)(\alpha_\pi-1)\sum_k \ln(\pi_k)\right] + c$$

We find

$$\ln\big(q^*(\theta_\pi)\big) \propto \mathbb{E}_{q(\theta_{\setminus\pi})}\left[\sum_k \{C(\alpha_\pi)(\alpha_\pi-1)+z_{1k}\}\ln(\pi_k)\right]$$

The fourth term $\ln\Big(q^*\big(\theta_{\mu|\lambda}\big)\Big)$, $\ln\Big(q^*\big(\theta_{\mu,\lambda}\big)\Big)$ and $\ln\big(q^*(\theta_\lambda)\big)$ are independent of A and therefore the same as previously.

Calculation of expectations

The missing expectations are given by the forward-backward algorithm

$$\begin{aligned}\gamma_{t,k} &= \mathbb{E}_{q(z)}\big[z_{t,k}\big]\\ \xi_{t,k,l} &= \mathbb{E}_{q(z)}\big[z_{t,k}z_{t-1,l}\big]\end{aligned}$$

Corrected rate estimates

Estimating the corrected on-rate

The single molecule pipeline determines the corrected on-rate from the rastergrams of both the specific and non-specific traces. Here we assume that the specific traces are a sum of specific binding and non-specific binding and that the non-specific traces are only non-specific binding. Under this assumption the single molecule pipeline determines the non-specific on rate from the average time it takes to a binding event:

$$\hat{k}_{ns} = \mathbb{E}[t_{on}]^{-1}$$

And the fraction of all dark location that have an on event:

$$\hat{h} = \mathbb{E}\big[1_{t_{on}>0}\big]$$

Subsequently, the corrected-on rate is obtained by maximum likelihood estimation. The cumulative density function (CDF) is given by

$$CDF(t_{on}) = 1-(1-h)\,exp(-k_s t_{on}) - h\,exp(-(k_s+k_{ns})t_{on})$$

Therefore, the probability density function (PDF) is given by

$$P(t_{on}) = \frac{\partial}{\partial t_{on}}\;\; 1-(1-h)\,exp(-k_s t_{on}) - h\,exp(-(k_s+k_{ns})t_{on})$$

And the corresponding loglikelihood function is

$$L(k_s) = \prod_{\forall t_{on}} ln\big(P(t_{on})\big)$$

which is maximized to obtain the estimate $\hat{k}_s$.

Estimating the corrected off-rate

The pipeline also determines the corrected off-rate from the rastergrams of both the specific and non-specific traces. Here we assume that the specific traces are a sum of specific unbinding, non-specific unbinding and bleaching and that the non-specific traces are only non-specific unbinding and bleaching. The CDF for the non-specific traces is given by

$$CDF_{ns}\big(t_{off}\big) = 1 - (1 - h_1)\, exp\big(-k_{bl} t_{off}\big) - h_1\, exp\big(-k_{off_{ns}} t_{off}\big)$$

Subsequently, analogous to maximum likelihood estimation presented for the on-rate maximization of

$$L\left(h_1, k_{bl}, k_{off_{ns}}\right) = \prod_{\forall t_{on}} ln\left(P_{ns}\big(t_{off}\big)\right)$$

procedure results in the following estimates: $\hat{h}_1, \hat{k}_{bl}, \hat{k}_{off_{ns}}$.

$$CDF\big(t_{off}\big) = 1 - h_2(1 - h_1)\, exp\big(-k_{bl} t_{off}\big)$$
$$-h_2 h_1\, exp\big(-k_{off_{ns}} t_{off}\big) - h_2\, exp\big(-k_s t_{off}\big)$$

Finally, the maximization the likelihood function

$$L(h_2, k_s) = \prod_{\forall t_{on}} ln\left(P\big(t_{off}\big)\right)$$

results in the estimates $\hat{h}_2, \hat{k}_s$

APPENDIX 2: Scripts provided with the package

The installation package contains four scripts:

- pipeline_interface.m
  This script opens a graphic user interface. To execute the script, type pipeline_interface in the command line in MatLab.
- Example_dataset_analysis.m
  This script encodes the analysis of the Example dataset presented in this tutorial. Pre-processing steps (gain calibration, alignment of cameras, drift correction) and spot detection were performed as described in the manual. Target locations of poor quality were manually removed to illustrate this feature. Binding events were detected; association and dissociation rates were calculated and corrected for non-specific binding of the mobile component to the slide. Finally, number of binding states was estimated by the Bayesian HMM method. To view parameters used in this analysis, open the script in Editor in MatLab. To execute the script, type Example_dataset_analysis in the command line in MatLab.
- Example_dataset_analysis_with_thresholds.m
  This script encodes the analysis of the Example dataset presented in this tutorial. Pre-processing steps (gain calibration, alignment of cameras, drift correction) and spot detection were performed as described in the manual. Target locations of poor quality were automatically removed by modifying threshold parameters. Binding events were detected; association and dissociation rates were calculated and corrected for non-specific binding of the mobile component to the slide. To view parameters used in this analysis, open the script in Editor in MatLab. To execute the script, type Example_dataset_analysis_with_thresholds in the command line in MatLab.
- Example_dataset_pooling_processed_data.m
  This script merges two datasets, which correspond to two biological replicates and were processed independently. Processed data of target and dark locations are provided with the package. Association and dissociation rates were calculated and corrected for non-specific binding of the mobile component to the slide. To view parameters used in this analysis, open the script in Editor in MatLab. To execute the script, type Example_dataset_pooling_processed_data in the command line in MatLab.

pipeline_interface.m

```
clearvars
close all

gpuDevice;
affine2d;

addpath(genpath('./helperfunctions'))

%% LOAD GUI
h = createParentFigure;
cameraIndexDriftEst=1;

htabgroup = uitabgroup(h);
gdlObj = guiDataLoader(htabgroup);
set(0, 'CurrentFigure', h)
gaObj = guiAlignmentw(htabgroup,gdlObj);
gdcObj = guiDriftCorr(htabgroup,gdlObj,cameraIndexDriftEst);
gsdObj = guiSpotDetector(htabgroup,gdlObj,[],[],[],[],true);
gadObj = guiHMMAnalyse(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdrObj = guiDarkROIsw(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdcorObj = guiDarkCorrector(htabgroup,gadObj,gdrObj);
gdcObk = guiDataCollector(gdlObj,gaObj,gdcObj,gsdObj,gadObj,gdrObj);

clc
msgbox('The execution was successful','Operation Completed');
```

Example_dataset_analysis.m

```
%%
close all
clearvars

addpath(genpath('.\helperfunctions'))

path = '.\Example_dataset\';

fileStructure.Grid = [path 'Grid.tif'];
fileStructure.Dark = [path 'Dark.tif'];
fileStructure.BeadData = [path 'Beads.tif'];
fileStructure.Data = [path 'Experiment.tif'];
fileStructure.rg = -1;
fileStructure.NFramesGroup = 1;
fileStructure.lambdal = 550;
fileStructure.lambdar = 647;
fileStructure.intT = 5;
fileStructure.flipCams = 0;

numberOfSummedFrames = 1;

h = createParentFigure;
cameraIndexDriftEst=1;

htabgroup = uitabgroup(h);
gdlObj = guiDataLoader(htabgroup,fileStructure,[],numberOfSummedFrames,1);
set(0, 'CurrentFigure', h)
gaObj = guiAlignmentw(htabgroup,gdlObj);
gdcObj = guiDriftCorr(htabgroup,gdlObj,cameraIndexDriftEst);
gsdObj = guiSpotDetector(htabgroup,gdlObj,[],[],[],[],true);
gadObj = guiHMMAnalyse(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdrObj = guiDarkROIsw(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdcorObj = guiDarkCorrector(htabgroup,gadObj,gdrObj);
gocObj = guiDataCollector(gdlObj,gaObj,gdcObj,gsdObj,gadObj,gdrObj);

gaObj.alignmentEst;

%%

C = [...
  235   299;...
  165    90];

gdcObj.driftEst(C);

% set paramsPreFilterFits

gsdObj.paramsPreFilterFits.circularityMin=0.5;

gsdObj.paramsPreFilterFits.circularityMax=2;

gsdObj.paramsPreFilterFits.PH1Min=0;
```

```
gsdObj.paramsPreFilterFits.PH1Max=1;
gsdObj.paramsPreFilterFits.minPixelDist=7;
gsdObj.paramsPreFilterFits.clusterSizeMin=0;
gsdObj.paramsPreFilterFits.clusterSizeMax=100;

% set paramsFilterFits
gsdObj.paramsFilterFits.MinPhotons=50;
gsdObj.paramsFilterFits.MaxPhotons=Inf;
gsdObj.paramsFilterFits.MinBg=0;
gsdObj.paramsFilterFits.MaxBg=Inf;
gsdObj.paramsFilterFits.MinPValue=0;
gsdObj.paramsFilterFits.MaxPValue=1;
gsdObj.paramsFilterFits.MinPixelDist=2;
gsdObj.paramsFilterFits.MinCRLBSTD=0;
gsdObj.paramsFilterFits.MaxCRLBSTD=0.5;

gsdObj.detectSpots

gadObj.analyze

gadObj.spotsIncluded([2,5,6,10,12,15,16,19,32,35,44,45,55,64,69,74,77,...
    79,84,85,86,88,92,104,125,129,135,140,143,147,156,159,161,166,184,...
    187,194,201,203,214,238,242,265,267,269,279,280,284,288,297,298,301,...
    312,315,316,318,320,321,323,327,340,345,351,354,360,364,368,370,395,...
    396,398,414,418,423,426,429,432,442,443,445,446]) = 0;

gadObj.setThreshold([150 1 1 4.6]);
gabDurVec = gadObj.getGabDurSliders;
gadObj.setGabDurSliders([5 5]);

gadObj.sortType = 3;
gadObj.sortState = 1;
gadObj.align = 0;
gadObj.delete = 0;
gadObj.rastergramStartframe = 80;
gadObj.colorArray = [1 1 1;0 0 1;1 0 0;1 0 0;0 1 0;1 0.4 0.4;0 0 0];
gadObj.numberOfSpotsForRastergram = [];
gadObj.runRastergram

gdrObj.analyze

gdrObj.spotsIncluded([]) = 0;

gabDurVec = gdrObj.getGabDurSliders;
gdrObj.setGabDurSliders([5 5]);

gdrObj.sortType = 3;
gdrObj.sortState = 1;
gdrObj.align = 0;
gdrObj.delete = 0;
gdrObj.rastergramStartframe = 80;
gdrObj.colorArray = [1 1 1;0 0 1;1 0 0;1 0 0;0 1 0;1 0.4 0.4;0 0 0];
gdrObj.numberOfSpotsForRastergram = [];
gdrObj.runRastergram
```

```
gdrObj.savevars([mfilename date 'DarkROIs.mat']);
gadObj.savevars([mfilename date 'Target.mat']);

gadObj.prior.alpha = 1;
gadObj.prior.kappa = 100;
gadObj.prior.m = '[500 1000 1500 2000; 500 1000 1500 2000]';
gadObj.prior.v = 10;
gadObj.prior.W = 10;
gadObj.NTimes = 10;
gadObj.NOrder= 4;
gadObj.bayesSignalFitGMM

Nsignal=1000;
sortStateSignal=1;
gadObj.bootStrapRastergram(Nsignal,sortStateSignal);

Ndark=1000;
sortStateDark=1;
gdrObj.bootStrapRastergram(Ndark,sortStateDark);

gdcorObj.getCorrectedOnrate('cdf');
gdcorObj.getCorrectedOffrate('cdf');
```

Example_dataset_analysis_with_thresholds.m

```
%%
close all
clearvars

addpath(genpath('.\helperfunctions'))

path = '.\Example_dataset\';

fileStructure.Grid = [path 'Grid.tif'];
fileStructure.Dark = [path 'Dark.tif'];
fileStructure.BeadData = [path 'Beads.tif'];
fileStructure.Data = [path 'Experiment.tif'];
fileStructure.rg = -1;
fileStructure.NFramesGroup = 1;
fileStructure.lambdal = 550;
fileStructure.lambdar = 647;
fileStructure.intT = 5;
fileStructure.flipCams = 0;

numberOfSummedFrames = 1;

h = createParentFigure;
cameraIndexDriftEst=1;

htabgroup = uitabgroup(h);
gdlObj = guiDataLoader(htabgroup,fileStructure,[],numberOfSummedFrames,1);
set(0, 'CurrentFigure', h)
gaObj = guiAlignmentw(htabgroup,gdlObj);
gdcObj = guiDriftCorr(htabgroup,gdlObj,cameraIndexDriftEst);
gsdObj = guiSpotDetector(htabgroup,gdlObj,[],[],[],[],true);
gadObj = guiHMMAnalyse(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdrObj = guiDarkROIsw(htabgroup,gdlObj,gaObj,gdcObj,gsdObj);
gdcorObj = guiDarkCorrector(htabgroup,gadObj,gdrObj);
gocObj = guiDataCollector(gdlObj,gaObj,gdcObj,gsdObj,gadObj,gdrObj);

gaObj.alignmentEst;

%%

C = [...
  235   299;...
  165    90];

gdcObj.driftEst(C);

% set paramsPreFilterFits
gsdObj.paramsPreFilterFits.circularityMin=0.8;
gsdObj.paramsPreFilterFits.circularityMax=1.4;
gsdObj.paramsPreFilterFits.PH1Min=0;
gsdObj.paramsPreFilterFits.PH1Max=1;
gsdObj.paramsPreFilterFits.minPixelDist=7;
gsdObj.paramsPreFilterFits.clusterSizeMin=0;
gsdObj.paramsPreFilterFits.clusterSizeMax=50;
```

```
% set paramsFilterFits
gsdObj.paramsFilterFits.MinPhotons=50;
gsdObj.paramsFilterFits.MaxPhotons=Inf;
gsdObj.paramsFilterFits.MinBg=0;
gsdObj.paramsFilterFits.MaxBg=Inf;
gsdObj.paramsFilterFits.MinPValue=0;
gsdObj.paramsFilterFits.MaxPValue=1;
gsdObj.paramsFilterFits.MinPixelDist=2;
gsdObj.paramsFilterFits.MinCRLBSTD=0;
gsdObj.paramsFilterFits.MaxCRLBSTD=0.5;

gsdObj.detectSpots

gadObj.analyze

gadObj.spotsIncluded([]) = 0;

gadObj.setThreshold([150 1 1 4.6]);
gabDurVec = gadObj.getGabDurSliders;
gadObj.setGabDurSliders([5 5]);

gadObj.sortType = 3;
gadObj.sortState = 1;
gadObj.align = 0;
gadObj.delete = 0;
gadObj.rastergramStartframe = 80;
gadObj.colorArray = [1 1 1;0 0 1;1 0 0;1 0 0;0 1 0;1 0.4 0.4;0 0 0];
gadObj.numberOfSpotsForRastergram = [];
gadObj.runRastergram

Nsignal=1000;
sortStateSignal=1;
gadObj.bootStrapRastergram(Nsignal,sortStateSignal);

gdrObj.analyze

gdrObj.spotsIncluded([]) = 0;

gabDurVec = gdrObj.getGabDurSliders;
gdrObj.setGabDurSliders([5 5]);

gdrObj.sortType = 3;
gdrObj.sortState = 1;
gdrObj.align = 0;
gdrObj.delete = 0;
gdrObj.rastergramStartframe = 80;
gdrObj.colorArray = [1 1 1;0 0 1;1 0 0;1 0 0;0 1 0;1 0.4 0.4;0 0 0];
gdrObj.numberOfSpotsForRastergram = [];
gdrObj.runRastergram

Ndark=1000;
sortStateDark=1;
gdrObj.bootStrapRastergram(Ndark,sortStateDark);
```

```
gdcorObj.getCorrectedOnrate('cdf');
gdcorObj.getCorrectedOffrate('cdf');

gdrObj.savevars([mfilename date 'DarkROIs.mat']);
gadObj.savevars([mfilename date 'Target.mat']);
```

Example_dataset_pooling_processed_data.m

```
clearvars
close all

gpuDevice;
affine2d;

addpath(genpath('./helperfunctions'))
addpath(genpath('./helperfunctions/ext'))


%% Pooling example
clearvars
close all

path = './Example_processed_data/';

fileNameBase = [path 'Example_rep']

filesDark = dir([fileNameBase '*DarkROIs.mat']);
filesExp = dir([fileNameBase '*Target.mat']);
for i = 1:length(filesExp)
    experimentFiles{i} = [path filesExp(i).name];
    experimentFilesDarkROIs{i} = [path filesDark(i).name];
end

h2 = createParentFigure;
htabgroup2 = uitabgroup(h2);
gepObj2 = guiExperimentPooling(htabgroup2,experimentFiles);
gepdrObj2 = guiExperimentPoolingDarkROI(htabgroup2,experimentFilesDarkROIs);
gdcorObj = guiDarkCorrector(htabgroup2,gepObj2,gepdrObj2);
```

References

1. Smith, C. S., Joseph, N., Rieger, B. & Lidke, K. A. Fast, single-molecule localization that achieves theoretically minimum uncertainty. *Nat Methods* **7**, 373-375 (2010).

2. van Vliet, L. J., Sudar, D. & Young, I. T. Digital Fluorescence Imaging Using Cooled CCD Array Cameras invisible. *JE Celis (eds); Cell Biol. Second Eddition* **3**, 109-120 (1998).

3. Smith, C. S. et al. Nuclear accessibility of β-actin mRNA is measured by 3D single-molecule real-time tracking. *J Cell Biol* **209**, 609-619 (2015).

4. Tuan, Q. P., Marijn, B., Lucas, J. V. V., Klamer, S. & Cris, L. L. H. Performance of optimal registration estimators. **5817**, 5817-5812 (2005).

5. Smith, C. S., Stallinga, S., Lidke, K. A., Rieger, B. & Grunwald, D. Probability-based particle detection that enables threshold-free and robust in vivo single-molecule tracking. *Mol Biol Cell* **26**, 4057-4062 (2015).

6. Kay, S. M. *Fundamentals of statistical signal processing. Vol 1, Estimation theory* (Englewood Cliffs, NJ: Prentice-Hall PTR, 1993).

7. Sjoerd, S. & Bernd, R. Accuracy of the Gaussian Point Spread Function model in 2D localization microscopy. *Opt. Express* **18**, 24461-24476 (2010).

8. Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006).