

Three-dimensional microtubule behaviour in *Xenopus* egg extracts reveals four dynamic states and state-dependent elastic properties

Philipp J. Keller¹, Francesco Pampaloni¹, Gianluca Lattanzi² and Ernst H.K. Stelzer¹

¹Cell Biology and Biophysics Unit, European Molecular Biology Laboratory,
Meyerhofstrasse 1, D-69117 Heidelberg, Germany

²Department of Medical Biochemistry, Biology and Physics, TIRES-Center and INFN,
University of Bari, Piazza Giulio Cesare 11, I-70124 Bari, Italy

Supplementary Material

A) Supplementary Figures

Supplementary Figure 1: Flow diagram of 3D image processing and data analysis

Supplementary Figure 2: Microtubule speed steps and dependency of the persistence length l_p on the microtubule growth speed

B) Supplementary Tables

Supplementary Table 1: 2D and 3D microtubule dynamic instability in interphase
Xenopus laevis egg extracts

Supplementary Table 2: 3D microtubule persistence length measurements as a function of
the kinetic state and the microtubule filament length

C) Supplementary Methods

Three-dimensional data analysis of microtubule dynamics

Overview of the processing algorithms

Microtubule dynamic instability data analysis

3D microtubule contour processing

Setting-up the processing environment: a user's guide

Derivation of the statistical error in the measurement of microtubule lengths

Derivation of the statistical error in the measurement of time intervals

Derivation of statistical errors in the analysis of microtubule elasticity

A) Supplementary Figures

Supplementary Figure 1: Flow diagram of 3D image processing and data analysis

Boxes with solid lines indicate programs that were employed to derive the data presented in this study. Boxes with dashed lines indicate programs (or program subroutines) provided in our algorithm packages, but that have not been used in the data analysis of this study.

Supplementary Figure 2: Microtubule speed steps and dependency of the persistence length l_p on the microtubule growth speed

(a) Global histogram of single speed steps that were recorded at an average temporal sampling of 3.6 s. The arrow indicates a sharp incline in step counts close to zero speed steps, possibly arising from the pause state(s).

(b) Analysis of the persistence length's dependency on the microtubule growth speed. The total statistics was subdivided into the speed intervals 2-8.3, 8.3-9.7, 9.7-11.3, 11.3-18.5, 18.5-30 $\mu\text{m min}^{-1}$. The intervals were chosen such that each bin contributed approximately the same statistics ($n \approx 260$ per bin) and yielded microtubule length distributions with identical centers ($23.54 \pm 0.05 \mu\text{m}$, s.e.m.), to rule out a bias due to the microtubule length-dependency of the persistence length. The growth speed error is the standard deviation $a/\sqrt{3}$ of the respective bin size a . The error of the persistence length is calculated as described in **Figure 4c**. A linear fit is indicated in red.

B) Supplementary Tables

Supplementary Table 1: 2D and 3D microtubule dynamic instability in interphase *Xenopus laevis* egg extracts

reference & system	average v_g [$\mu\text{m min}^{-1}$]	SD of v_g [$\mu\text{m min}^{-1}$]	average v_s [$\mu\text{m min}^{-1}$]	SD of v_s [$\mu\text{m min}^{-1}$]	J [$\mu\text{m min}^{-1}$]	ρ_{growth}	$\rho_{\text{shrinkage}}$	ρ_{pause}
[S1], 2D	11.1 8.6 8.5	2.1 3.2 4.9	15.2 12.3 16.6	4.7 4.6 6.8	–	–	–	–
[26], 2D	6.5 ± 0.3 7.6 ± 0.4	– –	9.3 ± 0.5 9.4 ± 0.5	– –	(0.58) (0.67)	–	–	–
[29], 2D	11	4	11.5	4.9	–	–	–	–
[S2], 2D	22.7	7.9	15.2	9.5	–	0.655	0.163	0.182
this study, 3D	11.1 ± 0.5	6.5	11.5 ± 1.9	8.3	$9.1 \pm 1.2^*$	0.772	0.021	0.207

reference & system	f_{cat} [min^{-1}]	f_{res} [min^{-1}]	$f_{g \rightarrow g\text{-pause}}$ [min^{-1}]	$f_{g\text{-pause} \rightarrow g}$ [min^{-1}]	$f_{g\text{-pause} \rightarrow s}$ [min^{-1}]	$f_{s \rightarrow s\text{-pause}}$ [min^{-1}]	$f_{s\text{-pause} \rightarrow s}$ [min^{-1}]	$f_{s\text{-pause} \rightarrow g}$ [min^{-1}]
[S1], 2D	1.86 0.78 0.60	0.96 0.54 0.54	–	–	–	–	–	–
[26], 2D	0.72 ± 0.18 0.66 ± 0.18	1.2 ± 0.3 0.96 ± 0.30	–	–	–	–	–	–
[29], 2D	0.34 ± 0.05	–	–	–	–	–	–	–
[S2], 2D	0.30 ± 0.09	0.30 ± 0.19	–	–	–	–	–	–
this study, 3D	0.06 ± 0.03	1.79 ± 0.89	0.71 ± 0.09	2.58 ± 0.36	0.31 ± 0.13	3.58 ± 1.26	0.79 ± 0.56	2.76 ± 1.04

Comparison of the speeds and transition frequencies from this 3D study and from previous 2D studies by Belmont et al. 1990 [S1], Verde et al. 1992 [2], Arnal et al. 2000 [23] and Niethammer et al. 2007 [S2]. The values from the study of Arnal et al. represent the situation of no stathmin/Op18 addition (i.e. the control data). All listed data are based on comparable experimental conditions with respect to the egg extract system and the time points of data acquisition. The values for the microtubule population growth speed J shown in brackets were calculated using the four measured parameters of the conventional kinetic model of microtubule dynamic instability [2]. All indicated errors are errors of the average values.

Supplementary Table 2: 3D microtubule persistence length measurements as a function of the kinetic state and the microtubule filament length

microtubule persistence lengths		
sub-population	l_p [mm]	$\langle L \rangle$ [μm]
all microtubules	1.8 ± 0.4	24.5 ± 0.2
growing microtubules	1.7 ± 0.4	24.3 ± 0.2
pausing microtubules	2.1 ± 0.5	25.3 ± 0.5
length regime 10-15 μm	0.7 ± 0.3	13.0 ± 0.1
length regime 15-20 μm	1.1 ± 0.3	17.5 ± 0.1
length regime 20-22.5 μm	1.7 ± 0.4	21.2 ± 0.0
length regime 22.5-25 μm	1.9 ± 0.4	23.7 ± 0.1
length regime 25-30 μm	2.1 ± 0.3	27.5 ± 0.1
length regime 30-35 μm	2.8 ± 0.3	32.3 ± 0.1
length regime 35-40 μm	3.1 ± 0.5	37.1 ± 0.1
length regime 40-50 μm	3.3 ± 0.5	43.5 ± 0.2
persistence length limit	5.2 ± 1.2	∞
growth speed regime 2-8.3 $\mu\text{m min}^{-1}$	1.8 ± 0.5	23.6 ± 0.4
growth speed regime 8.3-9.7 $\mu\text{m min}^{-1}$	1.5 ± 0.4	23.4 ± 0.5
growth speed regime 9.7-11.3 $\mu\text{m min}^{-1}$	1.5 ± 0.4	23.5 ± 0.6
growth speed regime 11.3-18.5 $\mu\text{m min}^{-1}$	1.6 ± 0.3	23.7 ± 0.4
growth speed regime 18.5-30 $\mu\text{m min}^{-1}$	1.3 ± 0.3	23.6 ± 0.9

Numerical results of the measurements are visualized in **Figure 4d**. The sub-population's mean microtubule contour length is provided in the right column. The error of l_p is discussed in the **Supplementary Methods**. The error of $\langle L \rangle$ is provided as the standard error of the mean (s.e.m.). The persistence length limit was determined by a sigmoidal fit analysis.

C) Supplementary Methods

Three-dimensional data analysis of microtubule dynamics

Custom processing routines were developed in Matlab (The MathWorks, Natick, MA, USA, see **Supplementary Software**) to analyze the three-dimensional data sets of microtubule dynamics over time. Detailed descriptions of these programs and a spatio-temporal error analysis are provided below.

The position of the microtubule aster's centrosome is calculated by determining the intensity's centre-of-mass after applying a threshold at 95 % of the intensity interval defined by the data stack's minimal and maximal intensity values. Subsequently, an algorithm compensates for the aster's translational movement by re-centring the image stacks with respect to the calculated positions of the centrosome. An interface for the image processing routines allows a manual tracing of microtubules by defining points along the three-dimensional structure via a graphical user interface (GUI). In our analysis, we provided up to six tracing points per microtubule and time point. Long and strongly bent microtubules required more tracing points than short and straight microtubules.

Subsequently, second-order and third-order polynomial curve fits are computed for each set of 3D tracing points (corresponding to one microtubule filament trace). The coordinates in each set are transformed into a new Cartesian reference frame defined by the axis given by the end-to-end vector of the microtubule and two axes perpendicular to it. Using the calculated pairs of polynomials, the microtubule's length at each time point is derived by analytical curve integration. The statistical errors introduced by the measurements and analyses result in 75 ms for the temporal regime and 988 nm for the spatial regime. Using different methods for microtubule length derivation (third-order polynomial fits, second-order polynomial fits or summation of linear coordinate interconnections) resulted in average variations of 0.2 % for the parameters of dynamic instability.

Next, processing routines performed the automated phase detection in the length over time data sets. Phase detection is a three-stage process and explained in the supplemental section "Data analysis". The routines were designed to test four different kinetic models: the conventional two-state model, the four-state model and the combinations of these models with a consideration of dynamic speed switching. The processing kernel of the phase detection module is based on an algorithm introduced by Gildersleeve et al. [S3]. For each coordinate in the microtubule length-over-time diagram, a second-order polynomial fit was applied to a range of N data points to either side. The resulting slope of the fit at the respective coordinate was then used to classify the microtubule's state at that time point. In the two-state model, a positive slope identifies a growth phase, while a negative slope corresponds to microtubule shrinkage. In the four-state model, a slope with an absolute value smaller than $v = 2\sigma_{length} / (2Nt_{interval})$ is accounted for as a pause. σ_{length} is the spatial error of the measurements and $t_{interval}$ is the time interval between two microtubule length measurements. Thus, v denotes the maximum slope that can statistically result in a non-dynamic microtubule's length-over-time diagram due to the spatial

error of the measurements. Larger fitting speeds were attributed to growth or shrinkage, depending on the sign (see also [S3]).

The algorithm for the detection of dynamical speed switching subdivides all detected growth and shrinkage phases into a number of k sub-intervals (with $k = 1$ to 5) and computes the statistical probability of a linear model with k sub-phases for all possible configurations of temporal coordinates. The set of coordinates that describes the data with the highest statistical probability is used for the temporal segmentation of the corresponding phase.

A post-processing routine eliminates all phases that exhibit statistical irregularities that result from a failure of the initial second-order slope analysis. Specifically, the control algorithm excludes those phases from the analysis that last shorter than the temporal spacing in the recordings (so-called null phases) or dynamic phases that are characterized by a speed sign that contradicts the phase assignment (which can occur at the start and end points in the length over time data sets).

The output of the automated phase fitting routines are sets of linear fits with either positive, negative or zero slopes (indicating growth, shrinkage and pause) for all analyzed microtubules (see **Figure 2** for some examples). By intersecting subsequent linear fits, the time points of phase switching are derived with a precision better than the time unit of the recordings, thereby ensuring spatio-temporal consistency (i.e. avoiding jumps in the length over time diagrams). Each intersection point is both the end time point of the preceding dynamic phase and the start time point of the next dynamic phase. This final computational step provides the identities, the kinetic properties and the temporal coordinates of all dynamic phases. A data harvesting routine converts these data sets into transition frequencies, state probabilities, phase duration statistics and dynamics speed distributions.

The average transition frequency $f_{initial \rightarrow final}$ from an initial state $S_{initial}$ to a final state S_{final} is calculated by dividing the total number n of transition events from the state $S_{initial}$ to the state S_{final} by the total observation time T of the state $S_{initial}$ in all experiments. The denoted statistical error of the average transition frequency is given by $f_{initial \rightarrow final} / \sqrt{n}$.

Overview of the processing algorithms

Two sets of image and data processing programs were developed in the Matlab computer language. The first package facilitates the image processing of three-dimensional time-lapse data of dynamic microtubule asters. It consists of algorithms for image pre-processing, manual and automated microtubule tracing and the analysis of microtubule populations and population statistics. The second package is employed for the analysis of the microtubule tracing data obtained with the first package. These algorithms quantify the three-dimensional geometry of microtubules, obtain length-over-time data sets from dynamic microtubules, perform an automated dynamic phase detection, quantify the parameters of microtubule dynamic instability for different kinetic models, analyze the microtubule growth contrast, compute parameter correlations in microtubule pausing and evaluate the state-dependent elasticity of microtubules in 3D. Since the algorithms are only sparsely annotated, a brief description of all programs is

provided (see also the task flow diagram in **Supplemental Figure 1**). More detailed explanations of individual routines are given in the following sections. Note that programs (2)-(4) require pre-processing data that is generated by the program processTL and will therefore not operate unless these data are available.

Image processing programs (including graphical user interfaces):

(1) **processTL**

This tool provides routines for image pre-processing (ROI definition, axial interpolation, detection of the centrosome). The program generates 3D image stacks from the raw microscopy data that are re-centred with respect to the centrosome and cropped according to the user's settings. The program provides options for the mapping of microtubule transits through a sphere around the centrosome, for the automated tracing of microtubule contours over time and for a temporal microtubule registration, allowing to identify microtubules at different time points. The program includes a graphical user interface. processTL is the only program that operates directly on the raw microscopy data.

(2) **analyzeMap**

The program operates on the microtubule transit mapping data, which is provided by processTL and allows to quantify the thermal fluctuations of microtubule transits through a sphere around the centrosome by setting a user-specified radius. The program contains routines for the temporal identification of microtubule transits and an automated rotational drift correction that allows tracking relative transit coordinates independently of the microtubule aster's rotational and translational movements.

(3) **defineMT**

The program operates on the re-centred image stacks that are provided by processTL and allows for the manual tracing of microtubules over time. All geometrical tracing data presented in this paper were obtained with this routine. The user is able to move along the spatial and temporal axis, while viewing slices of the 3D microtubule aster. The definition of 3D structures is facilitated by clicking at the corresponding coordinates in the images. The program has to be located in the "interpolation" directory that is generated by processTL.

(4) **lengthDistribution**

Similarly to defineMT, this program displays the re-centred image stacks of the dynamic microtubule asters. The user interface has been optimized for the manual tracing of entire microtubule populations that are required for microtubule population statistical analyses.

Data analysis programs (no GUIs, operate on data from programs (1)-(4)):

(5) **populationStatistics(p1, p2)**

This routine analyses the data that is obtained with lengthDistributions. The average population growth speed J and the start time point of microtubule aster growth are

computed from these data sets. The program has to be located in the “interpolation” directory that is generated by processTL. Parameter p1 indicates the temporal spacing in the 2D/3D recordings (in seconds); parameter p2 corresponds to the pixel size in the processed images (in μm).

(6) **fitTraces(p)**

The program operates on the raw tracking data that are obtained manually or by defineMT, computes the microtubule lengths (arithmetically and by second/third order polynomial fitting) and determines microtubule bending and microtubule tip deflections. The parameter p indicates the pixel size in the processed images (in μm), i.e. corresponds to the length unit in the contour data. The algorithm searches all sub-directories for coordinate data files, which are labelled “xCoordinatesX.txt”, “yCoordinatesX.txt” and “zCoordinatesX.txt” (X represents a natural number). The number of microtubule data sets is read from a file “microtubules.txt” in the corresponding directory that contains the number of microtubules as plain text. The coordinate files are also plain ASCII text files and are formatted according to the following layout: each row contains a temporal coordinate in the first column and an arbitrary number of tracing points along the filament in the remaining columns. The first spatial coordinate identifies the microtubule minus-end, while the last corresponds to the plus-end. An arbitrary number of time points can be entered in such a data file, although they have to be provided in increasing order. Each row must contain the same number of coordinates. If the number of available tracing coordinates is not constant over time (e.g. due to significant changes in filament length, zeros should be entered to fill the row’s elements. 3D microtubule tip-deflections are computed as the distance of the plus-end to the tangent to the second order polynomial fit at the minus-end. The coefficients of the second- and third-order fits are stored. Note that six coefficients are stored for the second-order fits and eight for the third-order fits, since fitting occurs in 3D space.

(7) **phaseDetection(p1, p2, p3, p4)**

The algorithm performs an automated phase detection for the microtubule length-over-time data sets that are computed by fitTraces. A detailed explanation of its functions is provided in the next section, since this program is essential for the microtubule dynamic instability analysis. The parameter p1 indicates the spatial error in the microtubule length measurements. Four additional open parameters in this program allow the user to define the kinetic model that is applied in the analysis (three being accessible via the function’s interface). The parameter p2 defines the slope range of the local second-order fits as defined by Gildersleeve et al. [S3] (see below). Moreover, the user can choose between a two-state and a four-state model of microtubule dynamic instability (parameter p3, “0” for two-state, “1” for four-state). Speed switching within dynamic phases can be considered via parameter p4 (“0” for deactivation, “1” for activation). The other (internal) parameters allow to define the statistical criteria that are employed for the definition of microtubule pausing and the statistical complexity of the analysis of speed switching. Apart from the definition of pausing based on the second-order slope analysis (our standard approach), a module along the lines of Brittle et al. [5] has been implemented and can be activated by setting p3 to “2”.

- (8) **phaseProcessing**
The program evaluates the phase assignments that are computed by phaseDetection and reconstructs a continuous phase history for each microtubule length over time data set.
- (9) **phaseCleanup(p)**
The program reads the phase history that is generated by phaseProcessing and removes all phases that indicate a failure of the second-order fitting (performed by phaseDetection) at any of the contributing data points. These phases are not considered in the downstream processing. The algorithm provides an (internal) option for the visualization of the raw microtubule tracing data superimposed with the detected phases. The parameter p indicates the spatial error in the microtubule length measurements.
- (10) **analyzeDynamics**
The algorithm evaluates the post-processed phase history for all microtubules and presents the information in a data table that contains the parameters of dynamic instability. This table has always the same layout, independently on the kinetic model that was selected in the algorithm phaseDetection. Parameters that are not included in the chosen kinetic model are replaced by zeros. The table comprises state probabilities, the centres and widths of the two speed distributions (growth and shrinkage), the number of observed phases, state observation times, the number of observed transition events, transition frequencies, the statistical standard deviation of transition frequencies and the average phase durations prior to state transition.
- (11) **analyzeContrast**
The program analyses microtubule phase diagrams with respect to the occurrence of growth phase pairs. The algorithm distinguishes between un-separated growth phase pairs and growth phase pairs separated by g-pauses. The program generates growth contrast histograms and computes the correlation between microtubule pause durations and the observed growth contrasts.
- (12) **analyzeMechanics(p)**
This program evaluates the microtubules' 3D geometries and combines this information with the analysis of 3D microtubule dynamic instability, which is performed by programs 7-10. The distributions of microtubule tip deflections are fitted with three theoretical models of linear mechanics (including the deflection analysis as described below and the model developed by Wilhelm & Frey [S4]). The persistence length is obtained and plotted in dependency of the filament length. From these data the elastic modulus and the bending modulus of the microtubules is obtained. The same analysis is performed for each microtubule dynamic state. The parameter p indicates the spatial error in the microtubule plus-end measurements.
- (13) **analyzePauses**
The algorithm combines the statistics of those experiments, for which microtubule population statistical measurements were performed (i.e. the start time point of microtubule aster polymerization is available). Using this pool of data, stochastic aspects in

microtubule pausing are analyzed. The analysis includes a correlation of pause duration, microtubule length at pause and start time point of pausing.

(14) **visualizePopulations(p)**

The algorithm analyses the population statistical measurements and computes the global average population growth speed as well as the precision of the global time axis. All measurements are visualized in an overview plot (by individually shifting each series on the length axis according to the shifts vector p).

The data analysis programs (6)-(14) must be located in the same directory. By executing the job processor “completeAnalysis” the programs (6)-(12) are executed sequentially, facilitating a complete microtubule dynamic instability and microtubule elasticity analysis for the raw tracing coordinate data sets that the user (or defineMT) provides in the subdirectories. Programs (13) and (14) require pre-processing of microtubule population statistics by program (5) and are therefore not automatically started by “completeAnalysis”. “completeAnalysis” requires six input parameters: the image pixel size (see (6)), the microtubule length error (see (9)), the microtubule plus-end tip position error (see (12)), the second-order fit range (see (7)), the identifier of the kinetic model (see (7)) and the identifier for activation of dynamic speed switching (see (7)).

Microtubule dynamic instability data analysis

The processing algorithms (6)-(10) were developed in the Matlab computer language to facilitate the analysis of two- and three-dimensional microtubule tracking data with respect to the filaments’ statistical dynamic behavior. This package was designed with a particular focus on a complete automation of all processing steps. The algorithms allow the user to choose between the conventional two-state model and the four-state model of microtubule dynamic instability that was developed in this study. Additionally, the algorithms can be configured to consider multiple growth or shrinkage phases, i.e. dynamic phases that exhibit an intrinsic switching of speeds.

The five programs are executed sequentially. The first program, “fitTraces” (see below), operates directly on the raw microtubule contour tracing coordinates. It derives microtubule length-over-time diagrams for three different types of geometrical fits: a simple arithmetic adding-up of intersection distances between the tracking points on the microtubules as well as three-dimensional second- and third-order fits to the microtubule contours. Thus, three length-over-time diagrams result for each microtubule. The three fitting schemes were implemented for two reasons: First, to investigate the resulting variability in dynamic instability parameters and second, to analyze the microtubules’ three-dimensional geometry itself.

The second algorithm (“phaseDetection”) determines the dynamic phases on the basis of the microtubule length-over-time diagrams. The phase analyses are performed independently for the three fitting data sets. The processing kernel is based on an algorithm introduced by Gildersleeve et al. [S3]. For each coordinate in the microtubule length-over-time diagram, a second-order polynomial fit is applied to a range of N data points to either side. The resulting

slope of the fit at the respective coordinate was used to classify the microtubule's state at that time point. In the two-state analysis, a negative slope was attributed to shrinkage and a positive slope to a growth phase. In the four-state analysis, a slope with an absolute value smaller than

$$v = \frac{2\sigma_{length}}{2Nt_{interval}} \quad \text{Suppl. Eq. 1}$$

was considered as a pause. σ_{length} is the spatial error of the measurements as discussed in the next paragraph, and $t_{interval}$ is the time interval between two microtubule length measurements. Thus, v denotes the maximum speed that can statistically result in a non-dynamic microtubule's length-over-time diagram due to the spatial error of the measurement. Larger fitting speeds were attributed to growth or shrinkage, depending on the sign. In order to adopt criteria that are consistent with the existing literature, N was chosen as 5 in the dynamic instability reference analysis [S3].

Optionally, an analysis of multiple dynamic phases can be performed. In this case, each growth or shrinkage phase that was identified by the algorithm is analyzed by linear regression. In a reference analysis, the χ^2 -value of a linear fit to the entire detected dynamic phases is computed. The probability of the linear (i.e. two-parameter, $n = 2$) model to describe the data set is $p_2(\chi^2) = \exp\{-\chi^2/2\}$.

In the next step, the dynamic phase's time interval is split into k sub-intervals (each containing at least three data points) and k linear fits are computed to the data points within these k sub-intervals. All possible combinations of sub-divisions of the original phase are considered. Again, the probabilities for the k fits are computed and the resulting p -values are multiplied. If the product of the probabilities is smaller than the reference probability, the reference probability is replaced by the new value and the parameters characterizing the sub-division of the phase are stored. This multi-phase fit analysis is performed for $k = 2, 3, 4$ and 5 . The dynamic phase characterization with the best overall p -value is stored. We show in table T1 that the consideration of dynamic speed switching mainly affects the width of the growth speed distribution. Therefore, we analyzed the sensitivity of this parameter with respect to the maximum number of considered sub-divisions k_{max} . The standard deviation of the growth speed distribution (Gaussian fit) results as $1.97 \mu\text{m}/\text{min}$, if speed changes are not taken into account. Including growth speed switching and setting $k_{max} = 3$ yielded a standard deviation of $3.24 \mu\text{m}/\text{min}$. For $k_{max} = 5$, we obtained a standard deviation of $3.32 \mu\text{m}/\text{min}$. Thus, while the introduction of growth speed switching itself results in an increase of 65 % of the distribution's standard deviation, increasing k_{max} from 3 to 5 resulted merely in a 2 % increase of the fitted parameter. Therefore, we consider the computation of speed switching using $k_{max} = 5$ to be sufficiently accurate. It should be noted that computational time increases exponentially with k_{max} .

In the subsequent step, the algorithm "phaseAnalysis" evaluates the phases identified by the program "phaseDetection". Linear fits are applied to the dynamic phases and positional mean values are computed for pauses. Temporal intersection points are derived for each pair of subsequent phases by intersecting the corresponding linear fits. Phase types, phase durations, speeds and standard deviations of the fitting parameters are stored in a matrix. The matrix is

post-processed by the algorithm “phaseCleanup”. The purpose of this post-processing is the elimination of statistical inconsistencies in the automatic analysis of the data sets. First, so called “null-phases” are removed from the matrix. These are phases that last for less than one time unit (which includes negative phase durations). Typically, null-phases can occur at the start or end time points in the data sets or at transitions from growth to shrinkage with high (de-)polymerization speeds. Null-phases indicate a failure of the second-order polynomial fit. The post-processing algorithm also removes those phases from the matrix that exhibit inconsistently high growth or shrinkage speeds or which are characterized by a negative growth speed or positive shrinkage speed. The maximum absolute speed was set to 40 $\mu\text{m}/\text{min}$, resulting in the elimination of less than 1% of the detected phases on average. In total, less than 5% of all detected phases were removed during post-processing data control. The post-processing algorithm also provides an option for the generation of visualization plots that show the microtubule length-over-time diagrams and the corresponding detected phase fits.

Finally, dynamic instability parameters are computed from the post-processed phase matrices by the algorithm “analyzeDynamics”. The algorithm determines the number of dynamic phases, average speeds and their standard deviations (also time-weighted averages and parameters of Gaussian fits), phase probabilities, transition frequencies and their standard deviations and average phase durations. The most important outputs of this last algorithm are three overview tables that contain the dynamic instability parameters for the three different geometrical fitting types. The tables are labeled “microtubuleStatistics_X”, where “X” substitutes “arithmetic”, “2nd” or “3rd”, depending on the fitting type. The layout of these tables looks as follows:

n_g	T_g / T_{all}	n_s	T_s / T_{all}	n_p	T_p / T_{all}
v_g	std. dev. of v_g	$-v_s$	std. dev. of v_s	l_p	std. dev. of l_p
$v_{g, \text{weighted}}$	std. dev. of $v_{g, \text{weighted}}$	$-v_{s, \text{weighted}}$	std. dev. of $v_{s, \text{weighted}}$	$l_{p, \text{weighted}}$	std. dev. of $l_{p, \text{weighted}}$
$v_{g, \text{weighted, Gaussian}}$	std. dev. of $v_{g, \text{weighted, Gaussian}}$	$-v_{s, \text{weighted, Gaussian}}$	std. dev. of $v_{s, \text{weighted, Gaussian}}$	$l_{p, \text{weighted, Gaussian}}$	std. dev. of $l_{p, \text{weighted, Gaussian}}$
T_g	T_g / n_g	T_s	T_s / n_s	T_p	T_p / n_p
$n_{g \rightarrow s}$	$n_{g \rightarrow g\text{-pause}}$	$n_{s \rightarrow g}$	$n_{s \rightarrow s\text{-pause}}$	$n_{p \rightarrow g}$	$n_{p \rightarrow s}$
$T_{g \rightarrow s} / n_{g \rightarrow s}$	$T_{g \rightarrow g\text{-pause}} / n_{g \rightarrow g\text{-pause}}$	$T_{s \rightarrow g} / n_{s \rightarrow g}$	$T_{s \rightarrow s\text{-pause}} / n_{s \rightarrow s\text{-pause}}$	$T_{p \rightarrow g} / n_{p \rightarrow g}$	$T_{p \rightarrow s} / n_{p \rightarrow s}$
f_{cat}	$f_{g \rightarrow g\text{-pause}}$	f_{res}	$f_{s \rightarrow s\text{-pause}}$	$f_{p \rightarrow g}$	$f_{p \rightarrow s}$
std. dev. of f_{cat}	std. dev. of $f_{g \rightarrow g\text{-pause}}$	std. dev. of f_{res}	std. dev. of $f_{s \rightarrow s\text{-pause}}$	std. dev. of $f_{p \rightarrow g}$	std. dev. of $f_{p \rightarrow s}$
$T_{g\text{-pause}}$	$n_{g\text{-pause} \rightarrow g}$	$n_{g\text{-pause} \rightarrow s}$	$T_{s\text{-pause}}$	$n_{s\text{-pause} \rightarrow g}$	$n_{s\text{-pause} \rightarrow s}$
	$T_{g\text{-pause} \rightarrow g} / n_{g\text{-pause} \rightarrow g}$	$T_{g\text{-pause} \rightarrow s} / n_{g\text{-pause} \rightarrow s}$		$T_{s\text{-pause} \rightarrow g} / n_{s\text{-pause} \rightarrow g}$	$T_{s\text{-pause} \rightarrow s} / n_{s\text{-pause} \rightarrow s}$
	$f_{g\text{-pause} \rightarrow g}$	$f_{g\text{-pause} \rightarrow s}$		$f_{s\text{-pause} \rightarrow g}$	$f_{s\text{-pause} \rightarrow s}$
	std. dev. of $f_{g\text{-pause} \rightarrow g}$	std. dev. of $f_{g\text{-pause} \rightarrow s}$		std. dev. of $f_{s\text{-pause} \rightarrow g}$	std. dev. of $f_{s\text{-pause} \rightarrow s}$
	$f_{g\text{-pause} \rightarrow g}$	$f_{g\text{-pause} \rightarrow s}$		$f_{s\text{-pause} \rightarrow g}$	$f_{s\text{-pause} \rightarrow s}$

“g”, “s” and “p” are abbreviations for growth, shrinkage and pause. The letter “n” indicates phase counts, “T” observation times, “v” average speeds, “l” microtubule lengths and “f” transition frequencies. The average growth and shrinkage speeds are calculated in three different ways: v_g and v_s are the arithmetic means of all observed phases, $v_{g, \text{weighted}}$ and $v_{s, \text{weighted}}$

are the time-weighted averages and $v_{g,weighted,Gaussian}$ and $v_{s,weighted,Gaussian}$ are the centers of the Gaussian fits to the time-weighted distributions of speeds. A green background indicates parameters characterizing microtubule growth; orange indicates microtubule shrinkage and blue microtubule pause (independently of the microtubule's history). The refined analysis of microtubule pausing is indicated by grey and pink. A grey background specifies g-pause, while a pink background represents s-pause statistics.

3D microtubule contour processing

The microtubule contours are analysed by program 6. For the analytical description of the microtubules' 3D geometry, the tracing points along the filament are fitted with two second-order or alternatively with two third-order polynomials (describing the filament's geometry in 3D space via parameterization in two orthogonal planes). Since only the second-order polynomials are employed in the analysis of the filaments' elasticity, the processing shall be outlined for this situation.

First, the tracing points T_i are transformed into a new Cartesian coordinate system that is generated for each trace individually. The first dimension in this new coordinate system is given by the vector $\vec{m} = (m_x, m_y, m_z)$, which points from the microtubule's minus-end to its plus-end. While any of the three components of \vec{m} can theoretically become zero, the first two components will never become zero at the same time. This situation would correspond to a microtubule that is oriented precisely along the axial recording axis (i.e. the axis of worst resolution) and thus represents a case that is avoided in the manual microtubule tracing. The remaining two dimensions are therefore given by the vectors $\vec{m}_1 = (m_{1,x}, m_{1,y}, m_{1,z})$ and $\vec{m}_2 = (m_{2,x}, m_{2,y}, m_{2,z})$, which are perpendicular to each other and perpendicular to \vec{m} , as follows:

$$\vec{m}_1 = (m_{1,x}, m_{1,y}, m_{1,z}) := \frac{1}{\sqrt{m_x^2 + m_y^2}} (m_y, -m_x, 0) \quad \text{Suppl. Eq. 2}$$

$$\vec{m}_2 = (m_{2,x}, m_{2,y}, m_{2,z}) := \frac{1}{\kappa} \left(\frac{m_{1,y}}{\sqrt{m_{1,x}^2 + m_{1,y}^2}}, -\frac{m_{1,x}}{\sqrt{m_{1,x}^2 + m_{1,y}^2}}, -\frac{m_x m_{2,x} + m_y m_{2,y}}{m_z} \right) \quad \text{Suppl. Eq. 3}$$

$$\text{with } \kappa = \sqrt{1 + \frac{(m_x m_{1,y} + m_y m_{1,x})^2}{m_z^2 (m_{1,x}^2 + m_{1,y}^2)}}$$

Next, we define $g: \vec{x} = \vec{m}t + \vec{d}$ as the straight line through the microtubule minus-end \vec{d} with the slope \vec{m} and compute the vectors \vec{v}_i from the tracing points T_i to the respective closest points on g . These vectors \vec{v}_i can be now transformed into the new coordinate system

and expressed as linear combinations of \vec{m}_1 and \vec{m}_2 (referred to as \vec{w}_i). Thus, the 3D geometry of the trace can be projected on the two independent directions \vec{m}_1 and \vec{m}_2 .

In the final step, two polynomial functions are fitted to the projections of the set to vectors \vec{w}_i on the $\vec{m} - \vec{m}_1$ - and the $\vec{m} - \vec{m}_2$ -planes. In the case of the second-order polynomials the two fit functions $f_{m1}(t)$ and $f_{m2}(t)$ are specified by three parameters each,

$$f_{m1}(t) = a_1 t^2 + b_1 t + c_1 \quad \text{and} \quad f_{m2}(t) = a_2 t^2 + b_2 t + c_2. \quad \text{Suppl. Eq. 4}$$

t is the normalized distance used in the definition of g and comprises values between zero (indicating the position of the microtubule minus-end) and one (the microtubule plus-end). Since the new coordinate system is a Cartesian system, the length L of the fitted curve is given by:

$$L = \int_0^1 dt \sqrt{(f'_{m1}(t))^2 + (f'_{m2}(t))^2 + 1}. \quad \text{Suppl. Eq. 5}$$

We obtain an analytical formula for the microtubules' filament length:

$$L = \left| \frac{2\alpha t + \beta}{4\alpha} \sqrt{\alpha t^2 + \beta t + \gamma} + \frac{4\alpha\gamma - \beta^2}{8\alpha^{3/2}} \ln \left(\frac{2\alpha t + \beta}{\sqrt{\alpha}} + 2\sqrt{\alpha t^2 + \beta t + \gamma} \right) \right|_{t=0}^{t=1} \quad \text{Suppl. Eq. 6}$$

with $\alpha = 4(a_1^2 + a_2^2)$, $\beta = 4(a_1 b_1 + a_2 b_2)$, $\gamma = b_1^2 + b_2^2$

Setting-up the processing environment: a user's guide

This section shall serve as a brief technical protocol for the application of the processing routines to 2D or 3D microtubule contour data sets. In order to set up the processing environment, we recommend the creation of a new directory in the processing machine's file system. The Matlab algorithms of the data processing package should be copied into this directory. The routines will expect the tracing data sets to be located in separate subdirectories for independent experiments. The data sets in these subdirectories are automatically detected and processed by our algorithms. Essentially, an arbitrary number of experiments can be provided. While directory names can also be chosen arbitrarily, the user should avoid the identifiers "contrast", "data", "graphs", "mechanics", "pausing" and "traces", since these names correspond to output data directories that will be generated by the programs and will therefore not be scanned for valid data sets.

An ASCII text file "microtubules.txt" must be present in each data sub-directory, which contains the number of microtubule contour data sets that are provided in this directory as plain text. Furthermore, an ASCII text file "temporalSpacing.txt" must be provided in each sub-

directory, which contains the temporal spacing in the microscopy recordings (measured in seconds) for the corresponding experiment. Finally, the user needs to provide the actual contour tracing data as ASCII text files. These data files are labeled “xCoordinatesX.txt”, “yCoordinatesX.txt” and “zCoordinatesX.txt”, where X is the microtubule identifier and ranges from 1 to the number provided in “microtubules.txt”. The layout of the contour files is as follows: Each row contains the temporal coordinate of the measurement in the first column and an arbitrary number of tracing points along the filament in the remaining columns (entered as natural numbers in the unit of pixels in the recordings). The first spatial coordinate identifies the microtubule minus-end, while the last corresponds to the plus-end. The data from an arbitrary number of time points can be entered in such a data file. However, the spacing between each pair of subsequent temporal coordinates must be equal to the temporal spacing provided in “temporalSpacing.txt” (i.e. spatial coordinates must be provided for all time points within the considered time interval). While each row must have the same number of coordinate entries, different numbers of available tracing coordinates at the individual time points can be indicated by entering zeros in the remaining columns. 2D data can be considered by providing the corresponding contour data in the x- and y-coordinate files and defining a z-coordinate data file (“zCoordinatesX.txt”) with only zeros as spatial coordinates.

After starting Matlab, the user can run most analyses (microtubule dynamic instability, microtubule elasticity and microtubule growth contrast) by switching to the processing directory and starting “completeAnalysis”. The analysis of microtubule pausing requires additional pre-processing by the program “populationStatistics” and is therefore not automatically started by “completeAnalysis”.

“completeAnalysis” is a job processing function that requires six input parameters (in the following order): Parameter 1 is pixel size of the recordings (entered in μm). Parameter 2 is the spatial error in the contour data (provided as the error in filament length in μm). Parameter 3 is the spatial error in the microtubule plus-end position (in μm). Parameter 4 is the range of the local second-order fit in the phase detection (measured in data points, as defined in [S3]). Parameter 5 is the identifier for the choice of the kinetic model and set to “0” for the two-state model or “1” for the four-state model. Alternatively, by choosing “2” the four-state model is employed on the basis of the pausing definition by Brittle et al. [5]. Parameter 6 is the identifier for the module for dynamic speed switching (“0” disables the module, “1” enables the module).

Finally, we would like to provide a simple example of an input data set. We assume we wish to analyze one single experiment with two traced microtubules. The temporal resolution in this experiment was e.g. 3 seconds (i.e. every 3 seconds the entire 3D volume of the microtubule aster has been recorded). First, we generate a new directory “processing” and copy the data analysis algorithms in this directory. Next, we generate a sub-directory “contours” with two text files: a file “microtubules.txt” that contains the number 2 (2 microtubule contour data sets are provided) and a file “temporalSpacing.txt”, which contains the number 3 (corresponding to the temporal spacing in the recordings). Finally, we provide six contour files in the same directory: “xCoordinates1.txt”, “yCoordinates1.txt”, “zCoordinates1.txt”, “xCoordinates2.txt”, “yCoordinates2.txt” and “zCoordinates2.txt”. The layout of each set of three coordinate files should then be:

t_start	<minus_end_frame1>	<coordinate2_frame1>	<coordinate3_frame1>	...
t_start + 3	<minus_end_frame2>	<coordinate2_frame2>	<coordinate3_frame2>	...
t_start + 6	<minus_end_frame3>	<coordinate2_frame3>	<coordinate3_frame3>	...
...

Here, t_{start} indicates the first time point of the time interval to be analyzed for the microtubule. Of course the dynamics of this particular microtubule could also be evaluated for n non-contiguous time intervals (which might be desirable e.g. if the imaging quality is not sufficiently good for tracing at intermediate time intervals) by simply providing n contour data sets with different microtubule identifiers.

Derivation of the statistical error in the measurement of microtubule lengths

There are three sources of errors in the microtubule length measurements: 1) the lateral and axial positioning errors for microtubule start points arising from the precision limits of the processing algorithm, 2) the lateral and axial positioning errors for microtubule end points arising from the microscope's resolution as well as 3) an additional manual positioning error for microtubule end points due to the local signal-to-noise ratio in the data sets.

For a given detection wavelength λ , a refractive index n and a numerical aperture NA , the formula for the diffraction limited lateral resolution r reads [S5]:

$$r = \frac{\lambda}{n\sqrt{3 - 2\cos(\alpha) - \cos(2\alpha)}} \quad \text{with } \alpha = \arcsin\left(\frac{NA}{n}\right) \quad \text{Suppl. Eq. 7}$$

For a detection wavelength of 515 nm (maximum in the emission spectrum of Alexa-488), a refractive index of 1.33 and a numerical aperture of 1.0 the resolution is therefore 288 nm.

The lateral and axial positioning errors for the microtubule's tip (a rather diffuse part of the microtubule) are affected by the microscope's resolution limit, since the corresponding measurement is performed manually. However, in order to render the manual end point tracking more reliable, an isotropic bi-cubic interpolation of the data sets is performed, resulting in a lateral and axial pixel size of 128 nm. Prior to the rescaling, the axial pixel size is 700 nm. Thus, we obtain slightly more than two axial pixels per axial resolution unit (1500 nm) prior to rescaling. The z-slice, which contains the brighter microtubule intersection, can be easily identified by eye in the corresponding two z-planes (approximately 15 pixels per axial resolution unit are provided after axial bi-cubic rescaling). Thus, the original axial pixel size of 700 nm is a safe estimate of the axial contribution to the manual microtubule end point error. The lateral situation is similar, since also slightly more than two pixels are provided per resolution unit (288 nm). However, since no additional bi-cubic interpolation is performed in these two dimensions, we assume the lateral resolution limit of 288 nm as the lateral contribution to the manual microtubule end point tracking. Thus, the total error arising from the manual resolution-limited end point tracking is $\sigma_{\text{tip,resolution}} = \sqrt{(700 \text{ nm})^2 + (288 \text{ nm})^2} = 757 \text{ nm}$.

Additionally, the detection of microtubule end points is hampered by the signal-to-noise level in the recorded data sets, resulting in an additional estimated positioning error of maximal 8 pixels during end point tracking (corresponding to $8 \times 129 \text{ nm} = 1032 \text{ nm}$). The resulting standard deviation equals to $3^{-1/2} \times 1032 \text{ nm} = 596 \text{ nm}$, if we assume a rectangular error distribution. We obtain the total positioning error for the microtubule's tip

$$\sigma_{\text{tip}} = \sqrt{\sigma_{\text{tip,resolution}}^2 + \sigma_{\text{tip,noise}}^2} = \sqrt{(757 \text{ nm})^2 + (596 \text{ nm})^2} = 963 \text{ nm}.$$

Next, the lateral and axial positioning errors for microtubule start points are derived. These points are automatically computed by an algorithm, which approximates their positions by the centrosome's center of mass. The error in this calculation is determined by the precision limit in the recorded data sets. Since the centrosome's geometry is approximately rotationally symmetric, the precision in calculating its center of mass is considerably better than the actual resolution limit (e.g. [S6, S7]). We derived an estimate for the error in the centrosome's position by calculating the standard deviation of the detected position over the last 20 time points in an aster time-lapse data set recorded next to the polymer wall. In this last time interval, the aster has a relatively stable position since many microtubules are attached to the plastic polymer wall. Hence, the resulting fluctuation in computing the centrosome's centre is a safe estimate of the positional microtubule start point error. The positional standard deviations result as 86 nm laterally and 36 nm axially. The total error arising from the centre-of-mass calculation is

$$\sigma_{\text{start,centre-of-mass}} = \sqrt{(86 \text{ nm})^2 + (36 \text{ nm})^2} = 93 \text{ nm}.$$

An additional positioning error is introduced in the automated axial processing of microtubule start points due to a discretization step in the processing algorithm. As an approximation, the slice closest to the centrosome's computed center of mass is defined as the slice that actually contains its center of mass. Thus, the maximal error in the axial position is $700 \text{ nm} / 2 = 350 \text{ nm}$, since 700 nm is the axial pixel size prior to rescaling. A standard deviation of $3^{-1/2} \times 350 \text{ nm} \approx 202 \text{ nm}$ has to be taken into account for the position detection of the microtubules' starting points, assuming a rectangular distribution of the positioning error. Hence, the total error in the microtubule start point position is

$$\sigma_{\text{start}} = \sqrt{\sigma_{\text{start,centre-of-mass}}^2 + \sigma_{\text{start,discretization}}^2} = \sqrt{(93 \text{ nm})^2 + (202 \text{ nm})^2} = 222 \text{ nm}.$$

The total error σ_{length} in microtubule length measurements (accounting for the positioning errors of the microtubule's start point and the microtubule's tip) is:

$$\sigma_{\text{length}} = \sqrt{\sigma_{\text{start}}^2 + \sigma_{\text{tip}}^2} = \sqrt{(963 \text{ nm})^2 + (222 \text{ nm})^2} \approx 988 \text{ nm} \quad \text{Suppl. Eq. 8}$$

Derivation of the statistical error in the measurement of time intervals

If a microtubule structure in the three-dimensional recording volume moves between two time points of a time-lapse, resulting in a different z-localization in the corresponding z-stacks, the calculated time interval does not represent the actual time spacing for this specific structure.

Let $t_{interval}$ be the time interval that passes between the recording of two subsequent image stacks and $N_{z-stack}$ the number of frames in one image stack. Accounting for a maximal fluctuation of one z-plane for a microtubule between two timepoints (empirical observation), the maximal error $t_{fluctuation}$ in the time spacing results in $t_{fluctuation} = t_{interval} / (N_{z-stack} - 1)$.

For a time-lapse with a time unit of $t_{interval} = 3.5$ sec and $N_{z-stack} = 28$ images per z-stack, a maximal error of $t_{fluctuation} = 130$ ms and a standard deviation of $3^{-1/2} \times 130$ ms = 75 ms result (assuming a rectangular error distribution). Since the error in the time measurement for the process of data acquisition itself is smaller than 1 ms, it can be safely neglected and the total error σ_{time} in time interval measurements results in $\sigma_{time} = 75$ ms.

Derivation of statistical errors in the analysis of microtubule elasticity

Two sources of errors have to be taken into account in the determination of the persistence length. The major contribution arises from the uncertainty in computing the microtubule tip deflection d . In order to obtain a conservative estimate of this error, we considered the total positioning error of the tip $\sigma_{tip} = 963$ nm (see "Derivation of statistical errors in microtubule length measurements") rather than the statistical component along the normal axis with respect to the microtubule minus-end tangent. In all persistence length analyses, deflection histograms were computed for two situations: In the first case, the measured deflections were binned. In the second case, the tip positioning error was added to all measured deflections and the resulting data set was binned. We then computed the difference in persistence lengths that were obtained from these two data sets. This difference σ_{p1} represents the worst-case error in the persistence length that can arise from the uncertainty of deflection measurements. Additionally, we considered the error that is introduced by data binning. For each bin of magnitude m , a statistical error of m/\sqrt{n} (with n contributing measurements to this bin) was taken into account. The contribution of this error to the uncertainty in the fitting was computed and the total resulting error σ_{p2} was considered as the error introduced by the data analysis itself. The total error for any obtained persistence length is given by $\sqrt{\sigma_{p1}^2 + \sigma_{p2}^2}$.

In one analysis (see **Figure 4d**), we restricted the data set of deflection measurements to specific microtubule length intervals. For this analysis, a third error source has to be taken into account. Since statistics from microtubules of slightly different lengths are combined, a conservative estimate yields an additional error of $a/\sqrt{3}$ (twice the standard deviation of the uniform microtubule length interval of the size a) in the filament length that is characterized by the obtained persistence length.

Finally, we performed an additional test for statistical robustness in the analysis of the filament length dependency of the persistence length. In order to perform a reliable fit, the underlying statistics must ensure sufficiently populated histograms. In the calculation of the persistence length, one effect of too low statistics is the underestimation of the histogram's width, thus resulting in an overestimation of the persistence length. In order to assess this issue, we decreased the statistics for each data point to one-half and computed the corresponding persistence lengths. The average absolute deviation of the resulting values as compared to the

parameters that were obtained for the full statistics was 10 %. For two data points, the persistence length increased when reducing the statistics, while it decreased for four other data points and remained constant (considering three digits) for the remaining two data points, indicating undirected statistical noise.

SUPPLEMENTARY REFERENCES

- [S1] Belmont, L. D., Hyman, A. A., Sawin, K. E. & Mitchison, T. J.: **Real-time visualization of cell cycle-dependent changes in microtubule dynamics in cytoplasmic extracts.** *Cell*, **62**:579-589 (1990).
- [S2] Niethammer, P., Kronja, I., Kandels-Lewis, S. , Rybina, S., Bastiaens, P. and Karsenti E.: **Discrete States of a Protein Interaction Network govern Interphase and Mitotic Microtubule Dynamics.** *PLoS Biology*, **5**:e29 (2007).
- [S3] Gildersleeve, R.F., Cross, A.R., Cullen, K.E., Fagen, A.P., & Williams, R.C., Jr.: **Microtubules grow and shorten at intrinsically variable rates.** *J Biol Chem*, **267**:7995-8006 (1992).
- [S4] Wilhelm, J. & Frey, E.: **Radial distribution function of semiflexible polymers.** *Phys Rev Let*, **77**:2581-2584 (1996).
- [S5] Grill, S. & Stelzer, E. H. K.: **Method to calculate lateral and axial gain factors of optical setups with a large solid angle.** *J Opt Soc Am A*, **16**:2658-2665 (1999).
- [S6] Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., et al.: **Imaging intracellular fluorescent proteins at nanometer resolution.** *Science*, **313**:1642-1645 (2006).
- [S7] Thompson, R. E., Larson, D. R. & Webb W. W.: **Precise nanometer localization analysis for individual fluorescent probes.** *Biophys J*, **82**:2775-2783 (2002).



