

# Segmentation and quantification of subcellular structures in fluorescence microscopy images using Squassh

Aurélien Rizk<sup>1,2</sup>, Grégory Paul<sup>2,5</sup>, Pietro Incardona<sup>2</sup>, Milica Bugarski<sup>1</sup>, Maysam Mansouri<sup>1</sup>, Axel Niemann<sup>3</sup>, Urs Ziegler<sup>4</sup>, Philipp Berger<sup>1</sup> & Ivo F Sbalzarini<sup>2</sup>

<sup>1</sup>Paul Scherrer Institute, Biomolecular Research, Molecular Cell Biology, Villigen PSI, Switzerland. <sup>2</sup>MOSAIC Group, Center of Systems Biology Dresden, Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany. <sup>3</sup>ETH Zurich, Institute of Molecular Health Sciences, Zürich, Switzerland. <sup>4</sup>Center for Microscopy and Image Analysis, University of Zurich, Zürich, Switzerland. <sup>5</sup>Present address: ETH Zurich, Computer Vision Laboratory, Zürich, Switzerland. Correspondence should be addressed to P.B. ([philipp.berger@psi.ch](mailto:philipp.berger@psi.ch)) or I.F.S. ([ivos@mpi-cbg.de](mailto:ivos@mpi-cbg.de)).

Published online 13 February 2014; doi:10.1038/nprot.2014.037

**Detection and quantification of fluorescently labeled molecules in subcellular compartments is a key step in the analysis of many cell biological processes. Pixel-wise colocalization analyses, however, are not always suitable, because they do not provide object-specific information, and they are vulnerable to noise and background fluorescence. Here we present a versatile protocol for a method named ‘Squassh’ (segmentation and quantification of subcellular shapes), which is used for detecting, delineating and quantifying subcellular structures in fluorescence microscopy images. The workflow is implemented in freely available, user-friendly software. It works on both 2D and 3D images, accounts for the microscope optics and for uneven image background, computes cell masks and provides subpixel accuracy. The Squassh software enables both colocalization and shape analyses. The protocol can be applied in batch, on desktop computers or computer clusters, and it usually requires <1 min and <5 min for 2D and 3D images, respectively. Basic computer-user skills and some experience with fluorescence microscopy are recommended to successfully use the protocol.**

## INTRODUCTION

An increasing amount of biological and medical research relies on single-cell imaging to obtain information about the phenotypic response of cells to a variety of chemical, mechanical and genetic perturbations. Although it is possible to distinguish obvious phenotypes by eye, computational analyses enable the following: the processing of large data sets; the generation of quantitative and less biased results; and the detection of subtler changes in phenotype by statistical analysis. In addition, the ability to observe and quantify multiple fluorescent markers in the same cell under various conditions and over time opens doors for spatiotemporal modeling of biological processes<sup>1</sup>.

Quantification of the shapes and spatial distributions of subcellular objects is an important task, as the fluorescent probes are usually related to cellular markers. Colocalization of objects between different color channels can be quantified by various methods that are either pixel-based or object-based<sup>2</sup>. Pixel-based methods compute an overlap measure between the pixel intensities of the different color channels and include the following: Pearson’s correlation coefficient<sup>3</sup>; the overlap and Manders’ overlap coefficients<sup>4</sup>; intensity correlation<sup>5</sup>; cross-correlation<sup>6</sup>; and techniques correcting for unspecific (random) colocalization<sup>7</sup>. Object-based methods first detect and delineate the objects represented in the image and then quantify their overlap<sup>3,8</sup> or nearest-neighbor distances<sup>9</sup>. This approach also allows correction for the cellular context and for unspecific colocalization<sup>8</sup>. Methods based on intensity correlation are not suitable when fluorophores are not in ratiometric numbers<sup>10</sup>, whereas methods looking for co-occurrence are very sensitive to noise and background levels. A more robust method that indicates the fraction of colocalized molecules on the basis of cross-correlation and autocorrelation has been proposed, but it cannot be automatically applied to a set of images<sup>11</sup>.

Object-based analysis provides access to additional features of biological relevance, such as the spatial distribution of objects within the cell and the shapes and sizes of objects. This allows interactions between objects to be inferred and statistical hypotheses about their distribution (e.g., random versus nonrandom) to be tested<sup>9</sup>. However, object-based methods require that the objects in the image be first detected and delineated, which is a nontrivial task that involves image segmentation. Object detection and image segmentation are still frequently done by hand or by using *ad hoc* heuristics such as thresholding or hand-crafted pipelines of filters. However, recent progress in computer vision has provided well-founded theories that can give justification to the methodology<sup>12</sup>.

## An overview of Squassh and its advantages

Here we present a protocol for Squassh and the colocalization of subcellular shapes. Squassh makes use of a segmentation method that directly connects the image-segmentation task with biological reality through prior knowledge about the imaged objects, the image-formation process and the noise present in the image. This allows the same method to be applied to a wide spectrum of images by adjusting the prior knowledge (i.e., changing parameter values). In addition, the segmentation method used here provides theoretical performance and robustness guarantees, is independent of manual initialization and directly corrects for microscope blur and detector noise, yielding optimally deconvolved segmentations<sup>13</sup>. This last feature is achieved by accounting for the microscope’s point-spread function (PSF), improving the capacity to segment objects with sizes close to the resolution limit. The algorithm makes no assumptions about the expected shapes of the segmented objects, hence minimally biasing the results. The algorithm is not limited to spot-like or spherical objects, and it can



be applied to the segmentation of more complex shapes, as shown in ref. 13 for an endoplasmic reticulum and epithelial tissue.

Squash uses the segmentation algorithm from ref. 13 by implementing it in a user-friendly multithreaded software plug-in for the free open-source bioimage processing frameworks ImageJ<sup>14,15</sup> and Fiji<sup>16</sup>, as part of the MosaicSuite. We extend the original algorithm by accounting for the fact that different objects may have different fluorescence intensities<sup>17</sup>, and by allowing subpixel accurate segmentations in both 2D and 3D. Subpixel segmentation has previously only been available in two dimensions<sup>18,19</sup>, and it has been shown to be useful, e.g., for studying the live morphology of endosomes<sup>20</sup>. We further extend the method with cell masks that allow the analysis to be restricted to a subpopulation of cells in an image, e.g., to transfected cells. The workflow is completed with a script, automatically generated by the plug-in as part of its output, for the free open-source statistical software program R (ref. 21), thereby providing statistical significance tests for the quantitative data generated.

### Limitations of Squash

The quality and reliability of Squash analysis mainly depends on the success of image segmentation. We recommend visually checking the image-segmentation outcome in at least one case per condition in order to confirm its quality. The algorithm used in Squash accounts for the linear characteristics of the microscope, but imposes no further prior knowledge. Although this helps limit bias in the analysis, it also solely relies on the image data, and segmentation artifacts may occur. The most frequent problem is that objects in the same color channel that are separated by less than the half-width of the PSF will be fused and detected as a single object. In addition, very dim objects may be missed altogether. Although this could potentially be avoided by imposing a shape prior (e.g., that all objects should be roundish), we choose not to do so, as such priors always bias the result toward objects of the sought-for shape. We prefer segmenting all shapes with equal probability, and defer any shape-based filtering to the postprocessing step. There, objects below, e.g., certain sphericities can be filtered out by using the provided statistical analysis script.

The Squash software also does not correct for chromatic aberration and other nonlinear optical effects. Positional shift between color channels must be corrected for either before (by image warping) or after (by object coordinate transformation) Squash analysis.

Colocalization analysis is limited to detecting overlap between objects from two-color channels, and it cannot be used to infer patterns within a single channel or long-range order in the object distribution across channels. A separate method is available for that<sup>9</sup> and implemented in software<sup>22</sup>.

The Squash protocol is limited to fluorescence microscopy. Other imaging modalities are not currently supported.

### Image segmentation

Image segmentation is a well-researched topic in computer vision, and many technological advances have successfully been applied to bioimage analysis<sup>12</sup>. Many user-friendly software tools are available for analyzing and quantifying fluorescence microscopy images<sup>23</sup>. They are either based on applying various filters to the pixels of the image (e.g., CellProfiler<sup>24</sup>), on using machine-learning

techniques to classify pixels as belonging to an object or to the background (e.g., Ilastik<sup>25</sup>), or on including models of the imaged objects and the image-formation process (e.g., Region Competition<sup>19</sup>). Filter-based approaches require the user to construct an appropriate pipeline of filters and to select all filter parameters. Machine-learning approaches require the user to manually label or segment a part of the data for the algorithm to learn the task. Model-based approaches require the user to design or choose a model that appropriately describes the type of image to be segmented.

Model-based methods aim to find the segmentation that best explains the image. In other words, they compute the segmentation that has the highest probability of resulting in the actually observed image when imaged with the specific microscope used. We chose this framework because it is generic to a wide range of image types and segmentation tasks, and because it provides direct access to statistical quantities that can be used in downstream analyses. We based our work on a recent extension of a family of image-segmentation models that include a variety of denoising and deconvolution tasks<sup>13</sup>. The image-segmentation method used in Squash is independent of initialization and robustly finds the optimal solution, because the underlying optimization problem is convex and hence has only globally optimal solutions; see ref. 13 for theoretical and algorithmic details.

We consider an image model where the intensity distribution within each object is homogeneous and where the image locally around each object consists of two regions: a brighter foreground and a darker background. The noise in the image can be either Gaussian or Poisson. Three quantities are estimated for each object in a given image: the segmentation, which is the estimated outline of the object; the fluorescence intensity in the local background around the object; and the fluorescence intensity in the interior of the object. The method used here provides optimal segmentations using prior knowledge about the microscope optics (see **Supplementary Note** for details). This strategy hence combines image denoising, deconvolution and segmentation<sup>13</sup>; it is not necessary to separately denoise or deconvolve the images beforehand. The results provided by such a joint deconvolution-segmentation procedure are of higher quality and are more robust than those obtained by standard deconvolution, followed by segmentation<sup>13</sup>. This is because the two tasks naturally regularize each other when considered jointly. This helps cope with very noisy data and with small objects close to the diffraction limit.

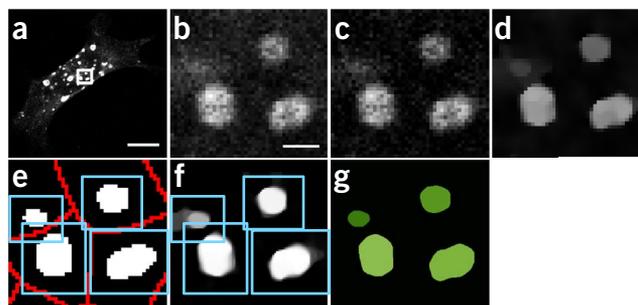
We account for spatial variations in the background intensity and for different objects having different foreground intensities by iteratively applying the procedure to local windows in the image around each object. Hence, each object may have a different estimated intensity, and the background around each object may locally be different (**Fig. 1**).

### Overview of the procedure

**Segmentation procedure.** We illustrate the segmentation procedure implemented in Squash by using an example image of Cherry-tagged RAB5 endosomes (**Fig. 1a,b**). First, object detection and separation is performed (segmentation stages (S)1–4), followed by estimation of the local background and object intensities and computation of the optimal segmentation (S5–7). The individual stages are described in detail below and referred to in the PROCEDURE.

## PROTOCOL

**Figure 1** | Workflow of the Squassh protocol illustrated on endosome segmentation. (a) Original image of a Cherry-RAB5–transfected HEK293 cell; scale bar, 10  $\mu\text{m}$ . (b) Close-up of the region highlighted by the white square in a; scale bar, 1  $\mu\text{m}$ . (c) The same close-up after background subtraction. (d) Object model image found by Squassh. Intuitively, this is a denoised and deconvolved version of c taking into account the microscope’s PSF. (e) Objects (white) obtained by thresholding image d. The image is decomposed into regions (blue boxes), and objects are separated by Voronoi decomposition (red lines). (f) Refined subpixel object model images obtained by applying the Squassh segmentation method inside each region with individual estimates for the local object and background intensities. (g) Final segmentation with the estimated object intensities displayed in shades of green.



- (S1) *Background subtraction* (Fig. 1c) is performed first, as the segmentation model assumes locally homogeneous intensities. Background variations are nonspecific signals that are not accounted for by this model. We correct uneven background intensity by using the rolling-ball algorithm<sup>26</sup>. This algorithm computes intensity histograms in a window moving across the image (i.e., a ‘rolling ball’). The edge length of the window is set by the user (Step 4 of the PROCEDURE). In each window, the most frequently occurring intensity value is taken as the local background estimate. This is based on the assumption that the objects of interest are smaller than the window size and cover less than half of the total window area.
- (S2) *Object detection* is performed over the entire image (Fig. 1d). This stage is not the final segmentation, but only serves to find regions in the image that contain objects of interest. This initial rough detection is done with the model-based algorithm from ref. 13 (see **Supplementary Note** for an intuitive description), but with fixed values for the mean background and object intensities (Steps 6 and 7 of the PROCEDURE). We find these fixed values by performing *k*-means clustering of all the pixels in the image.
- (S3) *Thresholding* of the initial detection reveals the initial objects (Fig. 1e). The threshold value is set to the user-defined minimum intensity of objects to be included in the analysis (Step 7 of the PROCEDURE). All objects with peak intensities lower than this threshold are discarded, and the others are retained. This parameter allows the user to control the sensitivity of the analysis, and it can be set to zero without compromising segmentation accuracy. Connected regions are identified as individual objects<sup>19</sup>. Connected regions share a common object and common background intensity. They can, however, be separated into individual objects during the segmentation performed in stages S6 and S7 below.
- (S4) *Decomposition* of the image into smaller parts is done in order to allow locally different background and foreground intensities for different objects (Fig. 1e). We decompose the image by (possibly overlapping) 2D or 3D boxes around each detected object from the previous stage (blue boxes in Fig. 1e). We ensure that objects in different boxes do not influence each other by computing the Voronoi diagram (on the basis of the actual shapes of the objects and not just their centroids) of the binary mask obtained in stage S3 (red lines in Fig. 1e). The image region considered for each object is then the intersection of the box around it with the Voronoi cell containing it (Fig. 1e).
- (S5) *Local background and object intensities* are estimated in each image region (Fig. 1e). This determines what will be considered background and foreground in the subsequent segmentation stage, and it provides local analysis of the image (Step 9 of the PROCEDURE). Intensities are estimated by solving the optimization problem described in the **Supplementary Note**.
- (S6) *Individual object segmentation* is obtained by running the algorithm from Paul *et al.*<sup>13</sup> separately for each image region (Fig. 1f; Steps 5–7 of the PROCEDURE). This can be done with a spatial resolution that is higher than the pixel resolution of the image, thus providing subpixel accuracy<sup>20</sup> (Step 8 of the PROCEDURE). As this oversampling is only done in local patches around the objects, rather than on the whole image, it has only moderate impact on the computational cost of the whole procedure (Table 1). As the segmentation algorithm accounts for the PSF of the microscope (Step 5 of the PROCEDURE), the influence of each pixel on its neighboring pixels is accounted for when segmenting apart individual objects. If the image is more likely to be the result of imaging two separate objects, they are split into two. If the image is better explained by assuming a single, connected object, the object is kept together. This always provides the segmentation that has the highest probability of explaining the observed image, given the knowledge of the PSF and the imaging noise model. Any further prior information, for example, about the expected shapes of objects, can be included in the postprocessing analysis.
- (S7) *The final segmentation* (Fig. 1g) is obtained by optimizing stage S6 to minimize the segmentation error, automatically done according to a rigorous optimality theory as previously described<sup>13</sup>.

**TABLE 1** | Computer time and memory requirements on a dual-core 2.3-GHz Intel Core i5 with 8 GB of RAM.

$x \times y \times z$ image size (pixels) and subpixel oversampling	Computer time (s)	Memory (MB)
512 × 512	17.2	275
512 × 512, 8× pixel oversampling	23.7	407
1024 × 1024	55.6	478
512 × 512 × 15	180.7	1,116
512 × 512 × 15, 4× pixel oversampling	201.3	1,536

**Colocalization analysis.** Object-based colocalization is computed after segmenting the objects by using information about the shapes and intensities of all objects in both channels. This allows straightforward calculation of the degree of overlap between objects from the different channels. We consider three different colocalization measures (see **Supplementary Note** for details). The first one,  $C_{\text{number}}$ , counts the number of objects that overlap between the two channels. Objects are considered overlapping if at least 50% of their volumes coincide. The second measure,  $C_{\text{size}}$ , quantifies the fraction of the total volume occupied by objects that overlap with objects from the other channel. The third measure,  $C_{\text{signal}}$ , quantifies colocalization in an intensity-dependent manner. It computes the sum of all pixel intensities in one channel in all regions where objects overlap with objects from the other channel. This signal-based definition has been described and used before<sup>8</sup>, but here we use the estimated object intensities from the segmentation rather than the raw pixel values. This improves robustness against noise and optical blur from the microscope

PSF, because the current segmentation method computes an optimally denoised and deconvolved estimate.

**Cell masks.** Often, analysis should be restricted to a single cell or a subset of the cells present in an image. Examples include working with transfected cells or with mixed cell populations. The Squassh software provides the option to compute cell masks by thresholding the original image and filling the holes of the resulting binary mask. Analysis is then restricted to the intersection of the image with the cell mask in order to ensure that only positive cells are considered in both channels.

**Statistical analysis.** Subcellular structures are often studied across a range of biological perturbations. A script for the R free open-source statistical software<sup>21</sup> is automatically generated by the plug-in, and it can be used to perform one-way ANOVA<sup>27</sup>, followed by a Tukey-Kramer test<sup>28</sup> for the statistical significance of differences observed between different sets of data (**Supplementary Note**).

## MATERIALS

### EQUIPMENT

- Confocal or wide-field fluorescence microscope. When you are using the protocol for colocalization analysis, the microscope must be capable of two-channel acquisition
- Image data files in any format supported by ImageJ (see <http://rsbweb.nih.gov/ij/features.html> for supported formats)
- A computer running Linux, MacOS X or Microsoft Windows with at least 2 GB of RAM (4 GB of RAM for 3D images)
- The free ImageJ (<http://rsbweb.nih.gov/ij/>) or Fiji (<http://www.fiji.sc>) software installed on the computer
- The free MosaicSuite plug-in, available from <http://mosaic.mpi-cbg.de/?q=downloads/imagej>. In addition to Squassh, MosaicSuite also provides software for single-particle tracking<sup>29</sup>, image segmentation<sup>19</sup>, spatial pattern analysis<sup>9,22</sup> and various utilities (for example, for estimating the microscope PSF from images)
- (Optional for statistical analysis) The free statistical analysis software R (<http://www.r-project.org/>) to use the plug-in-generated script
- (Optional for reading Leica .lif files) Leica .lif file extractor ImageJ/Fiji macro (provided in the **Supplementary Data**) to automatically extract individual images from .lif files. It uses BioFormats<sup>30</sup> and the channel rearrangement tools of ImageJ

### EQUIPMENT SETUP

**Installing Squassh in ImageJ** Download the MosaicSuite plug-in JAR file (<http://mosaic.mpi-cbg.de/?q=downloads/imagej>) and install the plug-in by moving the file into the plug-in subfolder of ImageJ, or by dragging and dropping the file into the ImageJ main window (**Supplementary Video 1**).

**Installing Squassh in Fiji** Register the update site <http://mosaic.mpi-cbg.de/Downloads/update/Fiji/MosaicToolsuite/> with your Fiji. In order to do so, open the Fiji Updater, then click on 'Advanced', and then on 'Manage update sites'. Click 'Add' and paste the URL to add the new update site. After adding the update site, run a Fiji update or select 'Mosaic ToolSuite.jar' from the list of files of the newly added update site, then restart Fiji. The Mosaic submenu should now appear under the 'Plug-ins' menu. The advantage of using the update site is that the Fiji installation will automatically upgrade MosaicSuite every time a new version of it becomes available in the future. If automatic updates are not required, use the manual installation procedure described for ImageJ (**Supplementary Video 1**).

**Installing the .lif file extractor macro** Follow the ImageJ/Fiji menu path 'Plug-ins' → 'Macros' → 'Install' to install the macro. Alternatively, paste the macro content into 'Plug-ins' → 'Macros' → 'Startup Macros' for a permanent installation.

## PROCEDURE

### Image data preparation

**1** | Export images from the microscope software. The plug-in works with any image format supported by ImageJ, but all stacks and channels of a single image have to be in the same .tif file. For Leica .lif files, use the .lif Extractor ImageJ macro to automatically extract dual-channel TIFF images. To do so, select 'Plug-ins' → 'Macros' → 'Lif Extractor' from the ImageJ/Fiji menu-bar and choose a folder containing one or more .lif files. A new folder containing the extracted TIFF images is then created for each .lif file.

**▲ CRITICAL STEP** Channel order needs to remain the same across all files. The Squassh software refers to 'Channel 1' and 'Channel 2' to indicate the location of segmented objects and for colocalization measures. It is possible to attribute other, clear-text names to the channels in the 'Visualization and output' settings (Step 15).

**▲ CRITICAL STEP** For colocalization analysis, care has to be taken to select spectrally separated fluorophores in order to avoid cross talk between the color channels, which would lead to spurious colocalization. Before computing colocalization between different color channels, it is important to correct for chromatic aberration as well as possible, as this biases the results<sup>9</sup>.

### ? TROUBLESHOOTING

## PROTOCOL

**Figure 2** | Screenshots of the graphical user interface of the Squassh software. (a) Main window with file selection, help and links to background subtraction, segmentation, cell masks and visualization options. (b) Background-subtraction options. (c) Segmentation parameter options. (d) Window for specifying the microscope's point-spread function by using a Gaussian model. (e) Cell mask window. (f) Cell mask preview. (g) Visualization and output options.

### Plug-in launch and input selection

2| Start the segmentation by choosing the ImageJ menu item 'Plug-ins' → 'Mosaic' → 'Segmentation' → 'Squassh'. This opens the main graphical user interface Squassh window shown in **Figure 2**. Protocol Steps 4–16 indicate how to navigate the four option windows that can be opened from this main window.

### ? TROUBLESHOOTING

3| Select the image file to be processed with the 'Select File/Folder' button (**Fig. 2a**). When selecting files containing two channels, both channels will be segmented and object-based colocalization analysis will be performed. For single-channel files, only segmentation is performed. For batch analysis, select a folder containing multiple images. It is recommended to first analyze a few individual images to determine suitable parameters that can then be applied across the whole set of images. Time-lapse images are not natively supported; they should be split into the different time points and analyzed separately. It is possible to batch-process a set of images by placing them in the same folder and selecting this folder as Squassh's input. The software is also compatible with ImageJ macros, enabling batch automation of larger analyses.

### Background subtraction

4| Reduce the background fluorescence by using the rolling ball algorithm. Click on 'Background subtraction' options, select 'Remove background' and enter the window edge length in units of pixels (**Fig. 2b**). This length should be large enough so that a square with that edge length cannot fit inside the objects to be detected, but smaller than the length scale of background variations. This step corresponds to segmentation stage S1 (**Fig. 1b,c**).

### ? TROUBLESHOOTING

### Segmentation parameters

5| Set the microscope PSF (**Fig. 2c**). To correct for diffraction blur, the software needs information about the PSF of the microscope. Either a theoretical PSF (option A) or a measured PSF (option B).

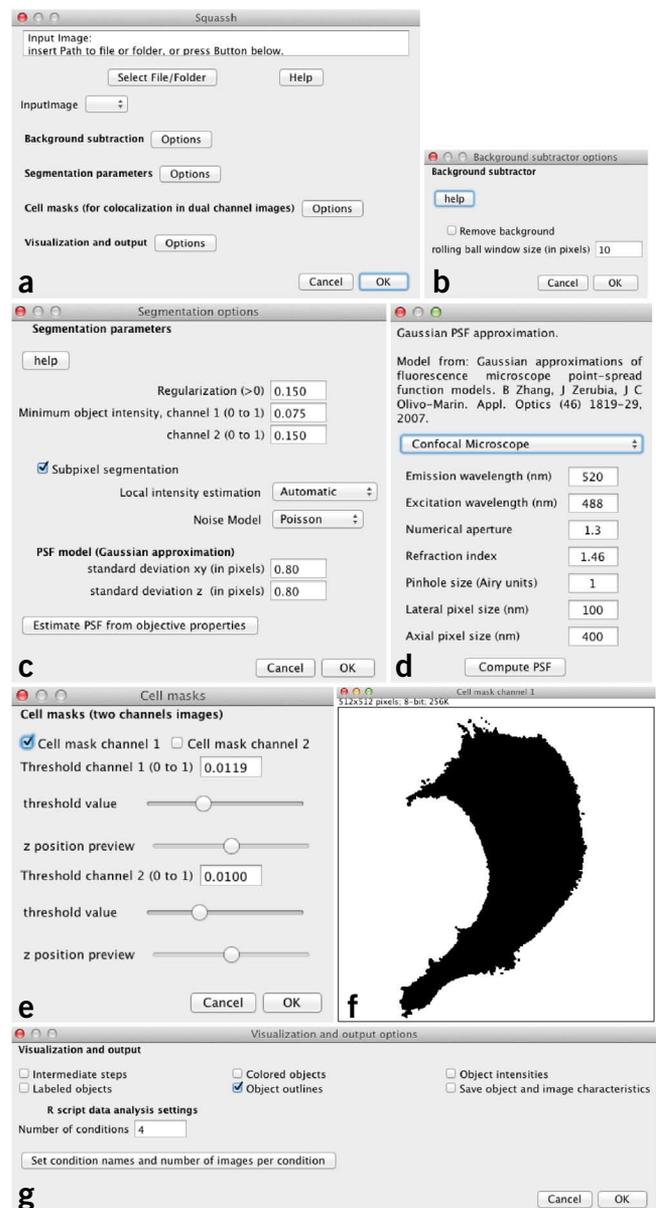
#### (A) Using a theoretical PSF

- Use a theoretical PSF model<sup>31</sup> by specifying the imaging condition parameters in the 'Segmentation parameters' options 'Estimate PSF from objective properties' subwindow (**Fig. 2d**). The software provides models for confocal and wide-field microscopes. Airy units only have to be entered for confocal microscopes.

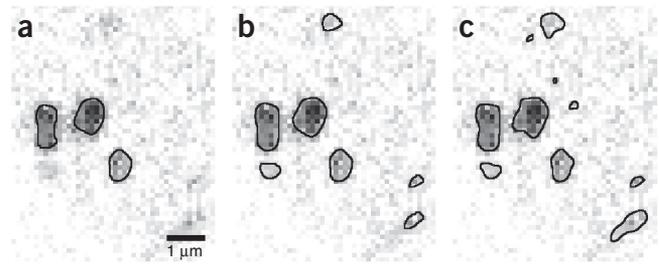
- Click on 'compute PSF'.

#### (B) Measuring the PSF

- Measure the microscope PSF from images of fluorescent subdiffraction beads. In this case, input the s.d. of the PSF separately for the lateral ( $x, y$ ) and axial ( $z$ ) directions (**Fig. 2c**). Use the menu item 'Plug-ins' → 'Mosaic' → 'PSF Tool' to measure these parameters from images of beads.



**Figure 3** | Illustration of how the parameters affect segmentation results by using endosomes in a close-up view of a Cherry-RAB5-transfected HEK293 cell as an example. **(a)** Segmentation with the minimum intensity threshold set to 0.100 and the regularization weight set to 0.250. **(b)** Segmentation with the minimum intensity threshold decreased to 0.050, resulting in a larger number of dimmer objects being detected. **(c)** Segmentation with a minimum intensity threshold of 0.050 and the regularization weight decreased to 0.075, resulting in a closer fit of the segmentation with the image data, but an increased sensitivity to noise (e.g., the small islands segmented, which may correspond to noise pixels).



**6|** Set a regularization parameter for the segmentation (**Fig. 2c**). Use higher values to avoid segmenting noise-induced small intensity peaks (**Supplementary Note and Fig. 3**). Typical values are between 0.05 and 0.25. This parameter controls segmentation stages S2 and S6 (**Fig. 1d,f**).

▲ **CRITICAL STEP** Regularization is a key segmentation parameter. The effect of varying the regularization parameter is illustrated in **Figure 3**. Once determined, the same parameter must be used across all studied images in order to allow comparison of the results.

? **TROUBLESHOOTING**

**7|** Set the threshold for the minimum object intensity to be considered (**Fig. 2c**). Intensity values are normalized between 0 for the smallest value occurring in the image and 1 for the largest value. This parameter controls segmentation stages S2 and S6 (**Fig. 1d,f**). Use low values for increased sensitivity and high values to force object separation.

▲ **CRITICAL STEP** Minimum intensity is a key segmentation parameter. The effect of varying the minimum intensity parameter is illustrated in **Figure 3**. Once determined, the same parameter must be used across all studied images to allow comparison of the results.

? **TROUBLESHOOTING**

**8|** (Optional) Select ‘subpixel segmentation’ to compute segmentations with subpixel resolution in stage S6 (**Fig. 2c**). The resolution of the segmentation is increased by an oversampling factor of 8 for 2D images and a factor of 4 for 3D images. Note that subpixel segmentation requires more computer time and memory. See **Table 1** for typical requirements.

**9|** (Optional) When images contain objects with inhomogeneous internal intensity distribution, the automatic local intensity estimation performed in segmentation stage S5 may not always be suitable. The ‘Local intensity estimation’ parameter thus provides the ability to estimate local object intensities by clustering (**Fig. 2c**). Select ‘Low’, ‘Medium’ or ‘High’ to set the local object intensity to the low-, medium- or high-intensity clusters obtained by *k*-means clustering of the pixels within the object. All results and benchmarks given in ANTICIPATED RESULTS section are obtained with the ‘Automatic’ setting, which is the recommended default setting.

**10|** Set the noise model corresponding to the microscope and detector used (**Fig. 2c**). We recommend using the Poisson model for confocal microscopes and the Gaussian model for wide-field microscopes. However, your equipment may differ. Detailed information should be found in the manufacturer’s data sheet.

**Cell masks**

**11|** A cell mask allows the analysis to be restricted to a certain region of an image, e.g., to a transfected cell. Select ‘cell mask channel 1’ and/or ‘cell mask channel 2’ to compute cell masks on the basis of the respective channel. Cell masks are computed by thresholding the respective channel and filling holes in the obtained binary mask. Open the ‘Cell masks’ option (**Fig. 2e**) and set a value between 0 and 1 for the threshold in ‘threshold channel 1’ and/or ‘threshold channel 2’. Adjust the threshold slider for a live preview (**Fig. 2f**) of the resulting mask. Adjust the z-slider to simultaneously adjust the z-position in the mask preview and the original image.

▲ **CRITICAL STEP** Once they have been determined, the same cell mask parameters must be used across all studied images in order to allow comparison of the results. When set of images is analyzed, cell masks are saved in files ending in ‘mask c1.zip’ and ‘mask c2.zip’ for the respective channel. Open them with ImageJ in order to check that the cell masks are correct for all images.

? **TROUBLESHOOTING**

**Visualization and output**

**12|** Select one or several of the output visualization options ‘Colored objects’, ‘Labeled objects’, ‘Object intensities’ and ‘Object outlines’ (**Fig. 2g**). With ‘Colored objects’, each object is visualized in a different, random color. ‘Labeled objects’



## PROTOCOL

assigns to all pixels of an object a value that is identical to the label (index) of that object in the result file. 'Object intensities' displays objects in their estimated fluorescence intensities. 'Object outlines' shows an overlay of the original image with the segmented outlines in red (**Supplementary Note**). For two-channel images, an additional visualization is displayed, which attributes a distinct color to each channel in order to reveal colocalization.

**13|** (Optional) Select 'Intermediate steps' to visualize also the background subtraction result (step S1; **Fig. 1c**) and the initial segmentation approximation (step S2; **Fig. 1d**).

**14|** (Optional) Select 'Save object and image characteristics' to store the resulting object quantifications in .csv files, which can be opened with any spreadsheet software or with MATLAB. This option is always selected by default when applying the analysis in batch mode to a set of multiple images (**Fig. 2g**).

**15|** (Optional) In order to generate an R script for statistical analysis of the results, set the number of experimental conditions considered under 'R script data analysis settings'. Click 'Set condition names and number of images per condition' to set the number of images per condition, the condition names (you can freely choose them) and the object names (can also be freely chosen). Names are used to generate graphical output and plots in human-readable form. The number of conditions and the number of images per condition are used to divide the set of images into the biological conditions studied. The set of images is divided according to the lexicographical ordering of the image file names.

### Run the analysis

**16|** Click 'OK' in the main window (**Fig. 2a**) to start the analysis. Depending on the speed of the computer, the analysis may take up to several minutes, and on older computers it may take even longer. See **Table 1** for typical runtimes on a typical modern computer.

### ? TROUBLESHOOTING

### Quantitative results

**17|** Open the ImageJ log window for a preview of the analysis results. The log displays the number of objects found and the signal colocalization  $C_{\text{signal}}$  (**Supplementary Note**).

**▲ CRITICAL STEP**  $C_{\text{signal}}$  should only be used if the same microscope gain and offset settings were used to acquire all images.

**18|** Use any software that can read .csv files in order to open the file ending in '\_ImagesData.csv'. This file contains the mean object features from each image. It stores the number of objects found in the image, the mean object size (in terms of area in 2D or volume in 3D), the mean object surface in 3D or perimeter in 2D, the mean length of objects (i.e., the maximum extension in the most extended direction) and the mean fluorescence intensity of the objects. For two-channel images, the file also stores the colocalization coefficients  $C_{\text{signal}}$ ,  $C_{\text{number}}$  and  $C_{\text{size}}$ , as well as the Pearson correlation across the whole image and inside the cell mask. All .csv data files are contained in the folder in which the original images are located.

**19|** Open files ending in '\_ObjectsData.csv' in order to obtain individual, per-object features. Each object is indexed by its segmentation label. For two-channel images, the file also stores the amount of overlap for each object with objects from the other channel, and the sizes and intensities of all colocalizing objects from the other channel. These features are used by the R script to statistically evaluate colocalization.

### Graphical output and statistical analysis (optional)

**20|** (Optional) Edit lines 23 to 27 of the automatically generated statistical analysis script 'R\_analysis.R' to set thresholds for minimum and maximum object sizes, minimum object intensity and minimum number of objects in an image. Objects and images violating the thresholds will be excluded from the analysis.

### ? TROUBLESHOOTING

**21|** (Optional) Edit line 32 of 'R\_analysis.R' to change the order in which the conditions are displayed in the resulting plots and graphs.

**22|** Start the R statistical software. Change to the working folder where 'R\_analysis.R' is located and type 'source('R\_analysis.R')' in the R console. This generates bar plots and performs statistical analysis of the data with one-way ANOVA<sup>27</sup> followed by a Tukey-Kramer test<sup>28</sup> for statistical significance of differences between different conditions. 'Colocalization.pdf' (**Supplementary Note**) displays the colocalization measures defined in the **Supplementary Note** and the Pearson correlation coefficients in the whole image or inside cell masks. It also provides the *P* value from a one-way ANOVA. 'ColocalizationCI.pdf'

(**Supplementary Note**) gives 95% confidence intervals for the difference of colocalization between pairs of conditions, together with their *P* values. The confidence interval and the *P* values are computed by a Tukey-Kramer test. The Tukey-Kramer test is suited for multiple comparisons (e.g., if there are more than two conditions), and it provides *P* values to estimate the statistical significance of differences between pairs of conditions. ANOVA *P* values test the hypothesis that all conditions have identical means. The two last pdf files generated contain the mean object properties across all conditions. See the **Supplementary Note** for examples of the output generated by this script.

▲ **CRITICAL STEP** If the number of objects detected for any condition is small (fewer than about ten), or the statistical distribution of features significantly differs from Gaussian, the results of the statistical test are unreliable and should not be trusted.

## ? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 2**.

**TABLE 2** | Troubleshooting table.

Step	Problem	Possible reason	Solution
1	The 'Lif Extractor' menu item is not present	The .lif file extractor macro has not been installed	Install the macro as described in the Equipment section
2	The Squassh menu item is not present	The MosaicSuite has not been installed	Install the MosaicSuite as described in the Equipment section
4	Background removal removes objects of interest (check by selecting the visualization option 'intermediate steps')	Inappropriate parameter settings	Reduce the edge-length parameter
6,7,16	Segmentation tends to fuse neighboring objects	Image is saturated	Avoid saturation by adjusting microscope gain and offset. There should be no more than a few, isolated saturated pixels
		Inappropriate parameter settings	Decrease the regularization parameter
		Segmentation is too sensitive. Background perturbations are interpreted as objects	Increase the regularization parameter to remove small low-intensity objects. Increase the minimum intensity parameter to remove low-intensity objects
11	Masks from two positive cells fuse together and include parts of a negative cell	Cell spacing is too small	Use images in which cells are farther apart or manually edit/curate the mask to exclude the space between two positive cells
16	The plug-in hangs during computation	There is not enough memory available or the image is too large	Analyze a down-sampled version of the image. Image size can be lowered from ImageJ by selecting 'Image' → 'Adjust' → 'Size'. If this solves the problem, it is a memory issue. Increase ImageJ maximum memory by selecting 'Edit' → 'Options' → 'Memory'. Memory requirements are given in <b>Table 1</b>
		Computer is slow or data set is large	Wait longer. See <b>Table 1</b> for typical runtimes on a modern computer. On older computers, the software can take markedly longer to complete the task
20	Some segmented objects are not biologically relevant and should be removed from the analysis	Fluorescence is not specific to the studied structures	Remove objects from the analysis by setting filters on object size or intensity in Step 20.

## ● TIMING

The only protocol steps requiring computational time in Squassh are image segmentation (Step 16) and statistical analysis (Step 22). All other steps only require setting parameter values and incur no computational cost. As a guideline, we provide here typical computation times measured on a dual-core 2.3 GHz Intel Core i5 with 8 GB of RAM. Statistical analysis (Step 22) required <1 min for a data set of 100 3D images of sizes between 512 × 512 × 10 and 512 × 512 × 25. **Table 1** provides



typical times for segmentation of 2D and 3D images with and without subpixel refinement. The computational cost only depends on image size for stages S1–S3 of the segmentation procedure. All other stages do not depend on image size, but rather on the sizes and numbers of objects detected in the images. The data in **Table 1** was obtained for images containing ~100 objects each. Image sizes are indicated in the table. For other numbers of objects, the times scale proportionally. The overhead incurred by subpixel oversampling mainly depends on the number and the sizes of the objects, and it ranged from 10 to 40% for the images used here. All pixel-intensity values are normalized between 0 and 1 and stored as 64-bit Java double variables. This renders the computational performance of the software independent of the bit depth of the original images.

**ANTICIPATED RESULTS**

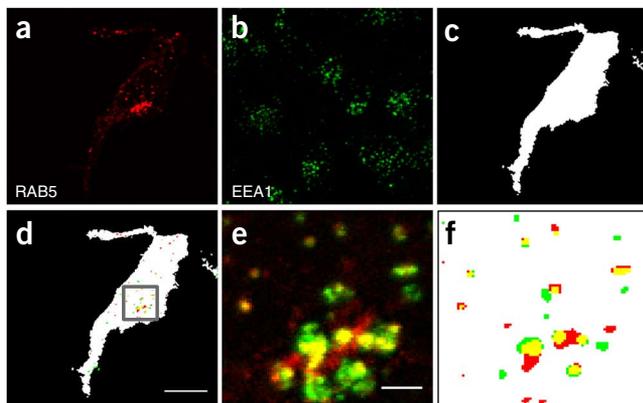
**Figure 4** shows typical images and colocalization results for segmentation of a single dual-channel image. Squassh allows the user to visually confirm that the cell mask appropriately delineates the transfected cell, and that the objects are satisfactorily segmented (close-ups in **Fig. 4e,f**). The segmented objects (**Fig. 4f**) appear smaller and more compact than their images (**Fig. 4e**), because the segmentation algorithm corrects for the diffraction blur from the PSF of the microscope, which causes objects to appear more blurry and fused in the image. Dim objects (especially visible in the red channel) are cut off owing to the ‘minimum intensity’ parameter.

**Figure 5** demonstrates a range of applications for Squassh. **Figure 5a** shows an analysis of subcellular localization of four RAB GTPases (RAB5, RAB4, RAB11 and RAB7) with known subcellular distribution and function, by using the following markers: early endosome antigen 1 (EEA1) as a marker for early endosomes; lysobisphosphatidic acid (LBPA) for late endosomes; lysosome-associated membrane protein 2 (LAMP-2) for lysosomes; and protein disulfide isomerase (PDI) for the endoplasmic reticulum. Two example images with close-ups of EEA1 (green) and RAB5 (red), and EEA1 (green) and RAB7 (red) are shown in **Figure 5b,c**, respectively. Example images for all other conditions are shown in **Supplementary Note**. Squassh analysis confirmed previously described localization data of RAB GTPases<sup>32–35</sup>. In addition, it allowed the maturation of endosomes to be monitored quantitatively, either by loss of colocalization with EEA1 or gain of colocalization with LBPA.

**Figure 5d** demonstrates that Squassh can be used in infection assays; Squassh successfully found a linear correlation between virus titer and the percentage of virus-infected cells in Sf21 cells infected with different amounts of YFP-expressing baculovirus<sup>36</sup>.

**Figure 5e** demonstrates that Squassh can be used to study changes in cell morphology. In cells overexpressing GDAP1, a fission factor, which leads to fragmented mitochondria<sup>37</sup>, Squassh was able to detect reduced mitochondria lengths and increased numbers of mitochondria (**Fig. 5e**). The total volume of mitochondria remained constant within the statistical noise (**Fig. 5e**). This is in line with previous manual analyses<sup>37</sup>. Example images from both conditions are shown in **Figure 5f**, with the outlines as segmented by Squassh overlaid in green. We observe the same effect that the segmented outlines seem to be smaller than the apparent objects in the image, because they correct for the diffraction blur from the PSF, hence deconvolving and segmenting at once.

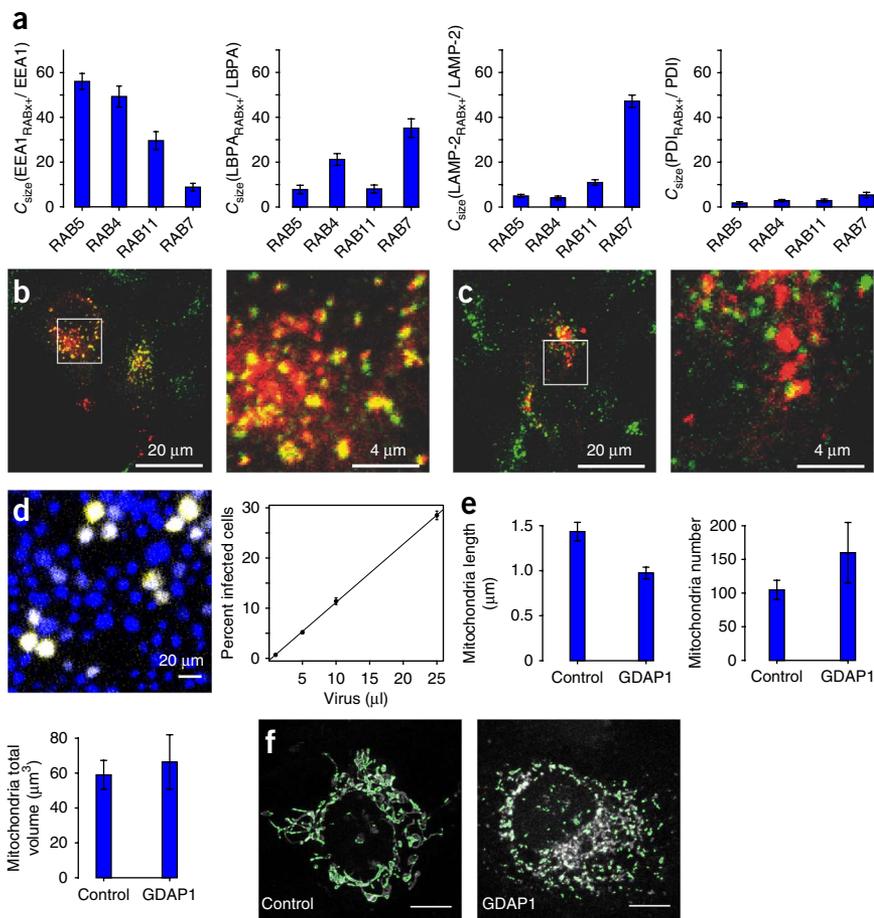
We quantitatively compared the segmentation method used in Squassh with eleven other methods on a test set of noisy synthetic images of cells with fluorescently labeled nuclei and vesicles. All images contained out-of-focus blur and uneven background illumination. The ground truth of all vesicle outlines was available and was used to compute the F-score quality measure. With an F-score of 0.62, the method used in Squassh performs as well as the previous winner on this benchmark<sup>38</sup>. This is remarkable, because the previous winner was a spot detector, tuned to detect round objects such as those present in these images, whereas Squassh makes no assumptions about the shapes of the objects (see ref. 13 for examples of complex-shaped structures, which would not be possible with a spot detector).



**Figure 4** | Segmentation and colocalization of EEA1 and RAB5. HEK293 cells were transiently transfected with a Cherry-RAB5 expression construct. EEA1, a marker for early endosomes, was detected with a rabbit monoclonal antibody (Cell Signaling, cat. no. 3288). All images show maximum intensity projections along the optical axis. (a) Raw image of the Cherry-RAB5 channel. (b) Raw image in the EEA1 channel. (c) Mask of a transfected cell determined by using the RAB5 channel. (d) Segmentation results of the vesicles from both channels overlaid with the cell mask. EEA1-expressing vesicles are shown in green, and RAB5-expressing vesicles are shown in red. The colocalization coefficient computed for this image is  $C_{\text{number}}(\text{EEA1}_{\text{RAB5}^+}/\text{EEA1}) = 0.36$ ; it is  $C_{\text{number}}(\text{EEA1}_{\text{RAB5}^+}/\text{EEA1}) = 0.02$  when not using a cell mask. (e) Close-up view of the raw image data in the area highlighted by the black square in d. (f) Object segmentation and overlaps in this area. Scale bars, 10  $\mu\text{m}$  (a–d) and 2  $\mu\text{m}$  (e,f).

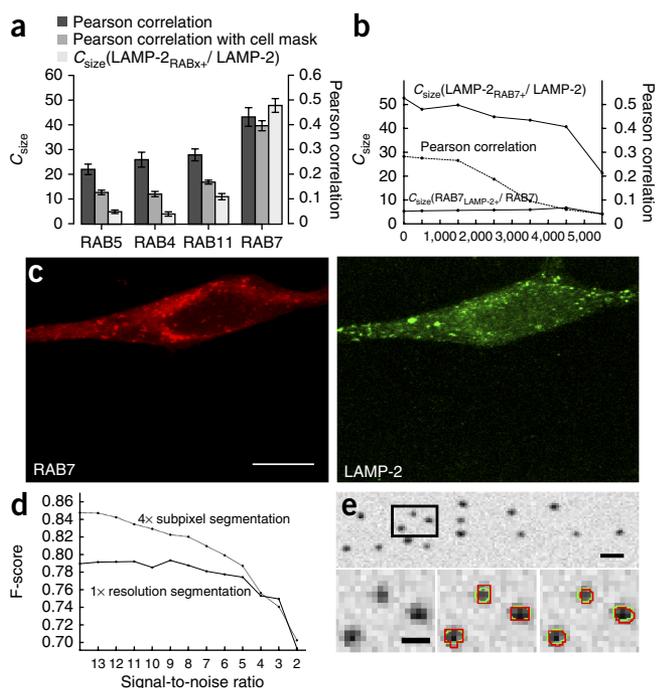


**Figure 5** | Anticipated results of the Squassh protocol. HEK293 cells were transiently transfected with Cherry-tagged RAB5, RAB4, RAB11 and RAB7. Subcellular compartments were immunostained with the following antibodies: EEA1 for early endosomes, LBPA for late endosomes, LAMP-2 for lysosomes and PDI for the endoplasmic reticulum. (a) Colocalization results for subcellular markers with different RAB GTPases. Error bars show means and s.e.m. over 20 images per condition of the fraction of subcellular marker colocalizing with the RAB channel, as determined by Squassh. 'RABx' represents a RAB in RAB5, RAB4, RAB11 or RAB7. (b) Example image with close-up of EEA1 (green) and RAB5 (red). (c) Example image with close-up of EEA1 (green) and RAB7 (red). Examples images for all conditions, as well as statistical analyses, are provided in the **Supplementary Note**. (d) Sf21 cells were infected with YFP-expressing baculovirus and stained with Hoechst 33258. A linear correlation between the amount of virus and the fraction of infected cells is observed. (e) Morphological analysis of mitochondria without and with overexpression of the fission factor ganglioside-induced differentiation-associated protein 1 (GDAP1). Error bars show means and s.e.m. over eight images per condition. (f) Example images from both conditions with segmentation outlines in green; scale bars, 10  $\mu\text{m}$ .



The Squassh object-based approach to colocalization analysis was compared with a pixel-based Pearson correlation analysis (**Fig. 6a**). The pixel-based method resulted in spurious colocalization of LAMP-2 with RAB4, RAB5 and RAB11. The object-based Squassh approach yields the best

result. Two phenomena affect the pixel-based results: First, many objects in the LAMP-2 channel overlap with objects in the RAB7 channel, but not vice versa (compare the two  $C_{\text{size}}$  graphs in **Fig. 6b**). In such asymmetric situations, pixel-based colocalization analysis is not appropriate, as it lumps both sides together. Second, pixel-based analysis is more sensitive to imaging noise than object-based analysis. This is shown in **Figure 6b**, where the Pearson correlation score tends to zero with increasing noise level<sup>3</sup> (**Fig. 6c**).



**Figure 6** | Benchmarks of the Squassh protocol. HEK293 cells were transiently transfected with Cherry-tagged RAB7. Lysosomes were identified by LAMP-2 immunostaining. (a) Pixel-based Pearson correlation (range -1 to +1) and object-based colocalization (range 0–100%) results for LAMP-2 with RABx (where 'x' stands for any RAB). For object-based analyses, images were segmented with the Squassh software. LAMP-2 is known to colocalize mainly with RAB7. Error bars show means and s.e.m. over 20 images per condition. (b) Pearson correlation and object-based colocalization coefficients for the image shown in c corrupted with increasing amounts of Gaussian noise. (c) Example image with both channels shown; scale bar, 10  $\mu\text{m}$ . (d) F-score of segmentation accuracy for fourfold subpixel-level and normal pixel-level segmentation. We show the mean values over five random images for each signal-to-noise ratio. (e) Example of a benchmark image with a signal-to-noise ratio of 12. The close-up views below show the ground truth outlines in green and those reconstructed by Squassh in red. The middle image uses normal pixel-level segmentation, whereas the right image uses fourfold subpixel oversampling; scale bars, 1  $\mu\text{m}$  for the top image and 0.5  $\mu\text{m}$  for the close-up images.

The accuracy of subpixel segmentation is illustrated on a synthetic image obtained by blurring a previously determined segmentation of RAB5 endosomes with the PSF of the microscope and adding modulatory Poisson noise of various magnitudes. We computed the F-score of the obtained segmentations with respect to ground truth. Subpixel segmentation improved the segmentation accuracy for signal-to-noise ratios (defined for Poisson noise according to ref. 29) above four (Fig. 6d). Figure 6e shows example results in comparison with standard pixel-level segmentation.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

**ACKNOWLEDGMENTS** We thank J. Cardinale (MOSAIC Group) for help with the software implementation of the ImageJ plug-in and B. Cheeseman (MOSAIC Group) for the voiceover in the video tutorial (Supplementary Video 1). This work was supported by SystemsX.ch, the Swiss initiative in systems biology under grant IPP-2011-113, evaluated by the Swiss National Science Foundation. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 290605 (PSI-FELLOW/COFUND) and from the German Federal Ministry of Research and Education under funding code 031A099.

**AUTHOR CONTRIBUTIONS** P.B., I.F.S. and U.Z. designed the project; G.P., I.F.S. and A.R. developed the image-processing algorithm; A.R., P.I. and I.F.S. were involved in software development and implementation; A.R., I.F.S., G.P. and P.B. were involved in benchmark design; experimental data were obtained by M.B., M.M., A.N. and P.B.; and A.R., I.F.S. and P.B. wrote the manuscript with input from G.P., A.N. and U.Z. Figures were prepared by A.R. and P.B., and the video tutorial was created by I.F.S.

**COMPETING FINANCIAL INTERESTS** The authors declare no competing financial interests.

Reprints and permissions information is available online at <http://www.nature.com/reprints/index.html>.

1. Sbalzarini, I.F. Modeling and simulation of biological systems from image data. *Bioessays* **35**, 482–490 (2013).
2. Adler, J. & Parmryd, I. Quantifying colocalization by correlation: the Pearson correlation coefficient is superior to the Mander's overlap coefficient. *Cytometry A* **77**, 733–742 (2010).
3. Bolte, S. & Cordelières, F.P. A guided tour into subcellular colocalization analysis in light microscopy. *J. Microsc.* **224**, 213–232 (2006).
4. Manders, E.M.M., Verbeek, F.J. & Aten, J.A. Measurement of co-localization of objects in dual-colour confocal images. *J. Microsc.* **169**, 375–382 (1993).
5. Li, Q. *et al.* A syntaxin 1,  $G_{\alpha_o}$ , and N-type calcium channel complex at a presynaptic nerve terminal: analysis by quantitative immunocolocalization. *J. Neurosci.* **24**, 4070–4081 (2004).
6. van Steensel, B. *et al.* Partial colocalization of glucocorticoid and mineralocorticoid receptors in discrete compartments in nuclei of rat hippocampus neurons. *J. Cell Sci.* **109** (Part 4): 787–792 (1996).
7. Costes, S.V. *et al.* Automatic and quantitative measurement of protein-protein colocalization in live cells. *J. Biophys.* **86**, 3993–4003 (2004).
8. Ramirez, O., Garcia, A., Rojas, R., Couve, A. & Hartel, S. Confined displacement algorithm determines true and random colocalization in fluorescence microscopy. *J. Microsc.* **239**, 173–183 (2010).
9. Helmuth, J.A., Paul, G. & Sbalzarini, I.F. Beyond co-localization: inferring spatial interactions between sub-cellular structures from microscopy images. *BMC Bioinformatics* **11**, 372 (2010).
10. Bolte, S. & Cordelières, F.P. Reply to letter to the editor. *J. Microsc.* **227**, 84–85 (2007).
11. Zinchuk, V., Wu, Y., Grossenbacher-Zinchuk, O. & Stefani, E. Quantifying spatial correlations of fluorescent markers using enhanced background reduction with protein proximity index and correlation coefficient estimations. *Nat. Protoc.* **6**, 1554–1567 (2011).
12. Danuser, G. Computer vision in cell biology. *Cell* **147**, 973–978 (2011).
13. Paul, G., Cardinale, J. & Sbalzarini, I.F. Coupling image restoration and segmentation: a generalized linear model/Bregman perspective. *Int. J. Comput. Vis.* **104**, 69–93 (2013).
14. Abramoff, M.D., Magalhães, P.J. & Ram, S.J. Image processing with ImageJ. *Biophoton. Int.* **11**, 36–42 (2004).

15. Schneider, C.A., Rasband, W.S. & Eliceiri, K.W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9**, 671–675 (2012).
16. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
17. Paul, G., Cardinale, J. & Sbalzarini, I.F. An alternating split Bregman algorithm for multi-region segmentation. in *Proc. 45th IEEE Asilomar Conf. Signals, Systems, and Computers* 426–430 (Asilomar, 2011).
18. Helmuth, J.A. & Sbalzarini, I.F. Deconvolving active contours for fluorescence microscopy images. in *Proc. Intl. Symp. Visual Computing (ISVC)*, vol. 5875 of *Lecture Notes in Computer Science*, 544–553 (Springer, 2009).
19. Cardinale, J., Paul, G. & Sbalzarini, I.F. Discrete region competition for unknown numbers of connected regions. *IEEE Trans. Image Process.* **21**, 3531–3545 (2012).
20. Helmuth, J.A., Burckhardt, C.J., Greber, U.F. & Sbalzarini, I.F. Shape reconstruction of subcellular structures from live cell fluorescence microscopy images. *J. Struct. Biol.* **167**, 1–10 (2009).
21. R Development Core Team. *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, 2012).
22. Shivanandan, A., Radenovic, A. & Sbalzarini, I.F. MosaicIA: An ImageJ/Fiji plug-in for spatial pattern and interaction analysis. *BMC Bioinformatics* **14**, 349 (2013).
23. Eliceiri, K.W. *et al.* Biological imaging software tools. *Nat. Methods* **9**, 697–710 (2012).
24. Carpenter, A.E. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006).
25. Sommer, C., Straehle, C., Koethe, U. & Hamprecht, F.A. Ilastik: Interactive learning and segmentation toolkit. *Genome Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on* doi:10.1109/ISBI.2011.5872394 (IEEE, 2012).
26. Sternberg, S.R. Biomedical image processing. *Computer* **16**, 22–34 (1983).
27. Chambers, J., Freeny, A. & Heiberger, R. Analysis of variance; designed experiments. in *Statistical Models in S* 145–193 (Wadsworth and Brooks/Cole Advanced Books and Software, 1992).
28. Miller, J.R. *Simultaneous Statistical Inference* (Springer-Verlag, 1981).
29. Sbalzarini, I.F. & Koumoutsakos, P. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.* **151**, 182–195 (2005).
30. Linkert, M. *et al.* Metadata matters: access to image data in the real world. *J. Cell Biol.* **189**, 777–782 (2010).
31. Zhang, B., Zerubia, J. & Olivo-Marin, J.C. Gaussian approximations of fluorescence microscope point-spread function models. *Appl. Opt.* **46**, 1819–1829 (2007).
32. Sönnichsen, B., De Renzis, S., Nielsen, E., Rietdorf, J. & Zerial, M. Distinct membrane domains on endosomes in the recycling pathway visualized by multicolor imaging of Rab4, Rab5, and Rab11. *J. Cell Biol.* **149**, 901–914 (2000).
33. Ballmer-Hofer, K., Andersson, A.E., Ratcliffe, L.E. & Berger, P. Neuropilin-1 promotes VEGFR-2 trafficking through Rab11 vesicles thereby specifying signal output. *Blood* **118**, 816–826 (2011).
34. Kriz, A. *et al.* A plasmid-based multigene expression system for mammalian cells. *Nat. Commun.* **1**, 120 (2010).
35. Matsuo, H. *et al.* Role of LBPA and Alix in multivesicular liposome formation and endosome organization. *Science* **303**, 531–534 (2004).
36. Fitzgerald, D.J. *et al.* Protein complex expression by using multigene baculoviral vectors. *Nat. Methods* **3**, 1021–1032 (2006).
37. Niemann, A., Ruegg, M., La Padula, V., Schenone, A. & Suter, U. Ganglioside-induced differentiation associated protein 1 is a regulator of the mitochondrial network: new implications for Charcot-Marie-Tooth disease. *J. Cell Biol.* **170**, 1067–1078 (2005).
38. Ruusuvoori, P. *et al.* Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images. *BMC Bioinformatics* **11**, 248 (2010).