

TECHNICAL ADVANCE

Dot Scanner: open-source software for quantitative live-cell imaging in planta

Holly Allen^{1,*†}, Brian Davis^{2,†}, Jenna Patel¹ and Ying Gu^{1,*} ¹Department of Biochemistry and Molecular Biology, Pennsylvania State University, University Park, Pennsylvania 16802, USA, and²Department of Astronomy and Astrophysics, Pennsylvania State University, University Park, Pennsylvania 16802, USA

Received 18 October 2023; revised 19 December 2023; accepted 22 January 2024; published online 4 February 2024.

*For correspondence (e-mail yug13@psu.edu, holly.allen@colorado.edu).

†These authors contributed equally to this work.

SUMMARY

Confocal microscopy has greatly aided our understanding of the major cellular processes and trafficking pathways responsible for plant growth and development. However, a drawback of these studies is that they often rely on the manual analysis of a vast number of images, which is time-consuming, error-prone, and subject to bias. To overcome these limitations, we developed Dot Scanner, a Python program for analyzing the densities, lifetimes, and displacements of fluorescently tagged particles in an unbiased, automated, and efficient manner. Dot Scanner was validated by performing side-by-side analysis in Fiji-ImageJ of particles involved in cellulose biosynthesis. We found that the particle densities and lifetimes were comparable in both Dot Scanner and Fiji-ImageJ, verifying the accuracy of Dot Scanner. Dot Scanner largely outperforms Fiji-ImageJ, since it suffers far less selection bias when calculating particle lifetimes and is much more efficient at distinguishing between weak signals and background signal caused by bleaching. Not only does Dot Scanner obtain much more robust results, but it is a highly efficient program, since it automates much of the analyses, shortening workflow durations from weeks to minutes. This free and accessible program will be a highly advantageous tool for analyzing live-cell imaging in plants.

Keywords: automated data analysis software, live-cell imaging, confocal microscopy, cellulose synthesis, technical advance.

INTRODUCTION

Live cell imaging of fluorescently tagged plant proteins is a vital tool for studying the dynamics, localization, function, and coordination of proteins within trafficking networks. It has revolutionized many areas of plant biology including the cytoskeleton (Boudaoud et al., 2014), organ morphogenesis (Grossmann et al., 2018), and cellulose biosynthesis (Allen et al., 2021). Assessing the various aspects of fluorescent particle dynamics often relies on analyzing a large number of high-quality images manually, which can quickly become time-consuming, and is prone to user error and unconscious bias.

Fiji-ImageJ is the most widely used program to analyze microscopy images across the field of biology (Schneider et al., 2012). While Fiji-ImageJ is both a highly useful and widely accessible tool for image analysis, it has limited options for automating analyses, and it relies heavily

on user input that can introduce errors. For instance, quantifying the velocity and lifetimes of particles requires the user to manually trace kymographs for each particle.

Also, Fiji-ImageJ cannot differentiate between more than one population of fluorescent particles in the same image, which is a common occurrence for plant proteins that are transported to the plasma membrane. Many fluorescent marker lines, for example, express a bright, aggregated cytosolic signal and a fainter punctate signal at the plasma membrane. This is seen in tagged cellulose synthase proteins (YFP-CESA6 and GFP-CESA7), and clathrin-mediated endocytosis marker lines such as mOrange-tagged clathrin light chain proteins (CLC-mOrange) and YFP-tagged adaptor protein complex 2 medium subunit proteins (AP2M-YFP) (Bashline et al., 2013; Konopka et al., 2008; Li et al., 2016; Paredes et al., 2006). To account for this when analyzing the density of particles with Fiji-ImageJ, the user must carefully

draw a region of interest (ROI), excluding cytosolic signal to analyze particles at the plasma membrane. Although some image analysis software exists to automate some of these analyses and reduce user error, such as Imaris, they are not free to use, and still require significant user input.

Due to the wide availability of open-source free software, programming languages, online resources, and active online communities, software development has become accessible to those without a computer science background. As a result, computer programming is often implemented in scientific research to automate data analysis, remove human error and biases, and perform tasks that are often not possible to do manually. For instance, the image-analysis tools available on Fiji-ImageJ have grown and diversified significantly since its creation, due to the implementation of different tools created by the scientific community (Schindelin et al., 2015; Schneider et al., 2012).

We designed a program called Dot Scanner, using the Python programming language, which quantifies the densities, lifetimes, and displacements of fluorescently labeled particles in plant tissues in a streamlined and unbiased manner. Specifically, Dot Scanner can detect punctate particles at the plasma membrane and distinguish these from strongly labeled particles in the cytosol of plant tissues, without requiring the user to exclude these regions manually. Furthermore, Dot Scanner can accurately detect punctate particles, even when the signal strength is weak and there is high interference from background signal, which naturally occurs during imaging due to the photobleaching of fluorescent particles. Dot Scanner provides a variety of inbuilt tools that give users control over the parameters most important to their analyses, particularly in the accurate detection of particles, and allows users to automate their analyses in an efficient and easy-to-use program.

RESULTS AND DISCUSSION

Features of Dot Scanner

Dot Scanner features a graphical user interface (GUI) with user-friendly controls to make the program widely accessible to those without prior programming knowledge (Figure 1). Users can easily change the parameters of the analysis using the configurations screen (Figure 1a), the threshold adjustment screen (Figure 1c), or by editing the configuration file by clicking the 'Edit defaults' button in the configurations screen. Full details on the program are included in the readme file prominently displayed on the GitHub project page (<https://github.com/bdavis222/dotscanner>).

Accurate particle detection

We developed an algorithm to accurately differentiate between bright aggregated signals, weaker punctate signals, and background signals, which we will describe here. Dot Scanner determines the size of punctate particles

(dots) and cytosolic signal (blobs) based on the total brightness of pixels (px) across a user-defined area. The Dot Scanner default sizes are set to two and five for dots and blobs, respectively (Figure 1a). A dot size of 2 means that the area of a dot is defined by a square region five px wide and five px tall (i.e., a central pixel extending an additional two px in each of the leftward, rightward, upward, and downward directions). Dot and blob sizes therefore closely approximate a radius of exclusion, in pixels.

For example, a dot size of two means that for every dot that is detected, all bright pixels that are detected within that dot's region (roughly a two px radius distance from the center of that original dot), will be excluded from the analysis. Due to the variable nature of particle fluorescence, these bright pixels could be the same particle and therefore should not be counted twice. Depending on the type of particle being examined, the user can adjust the dot size on the threshold adjustment screen to ensure particles are accurately captured (Figure 1c). The user has the ability to try a few different sizes and visually compare them to ensure which is appropriate, as shown in Figure 2a. Increasing the values of dot and blob sizes increases the size of the particle detected (Figure 2a). The default blob size is set to five because cytosolic signals typically require a wider radius of exclusion, since they are often saturated and overexposed.

In addition to the size of dots and blobs, there are three thresholds the user can adjust to alter the sensitivity of detecting dots and blobs, either on the configuration screen (Figure 1a), the threshold adjustment screen (Figure 2b), or in the configuration file. This is arguably the most crucial setting for accurately distinguishing blobs and dots, and for reducing the detection of background signal in the analysis. Figure 2b provides an example of how changing these thresholds can change the coverage of blobs and dots: the number of dots can be increased by lowering the lower dot threshold and the number of blobs can be increased by decreasing the upper dot threshold. To analyze different types of dots in the same images, one can use the 'Use previous analysis' feature, which is described in more detail in the 'additional features' section below.

Minimizing user bias

We have incorporated various features into Dot Scanner to help eliminate user bias that is often unavoidable in manual analyses. Firstly, to calculate particle densities, the Dot Scanner divides the total number of dots by the total area of the ROI minus the area occupied by the blob signal. This means that users do not have to carefully trace around cytosolic regions (Figure 1d), as these regions will automatically be removed from the total area. Secondly, Dot Scanner can detect short-lived particles of only a few seconds that are often missed in manual kymograph analyses (Figure S1). Smaller particle lifetimes are much harder to detect by eye, and as a result, there is tendency for

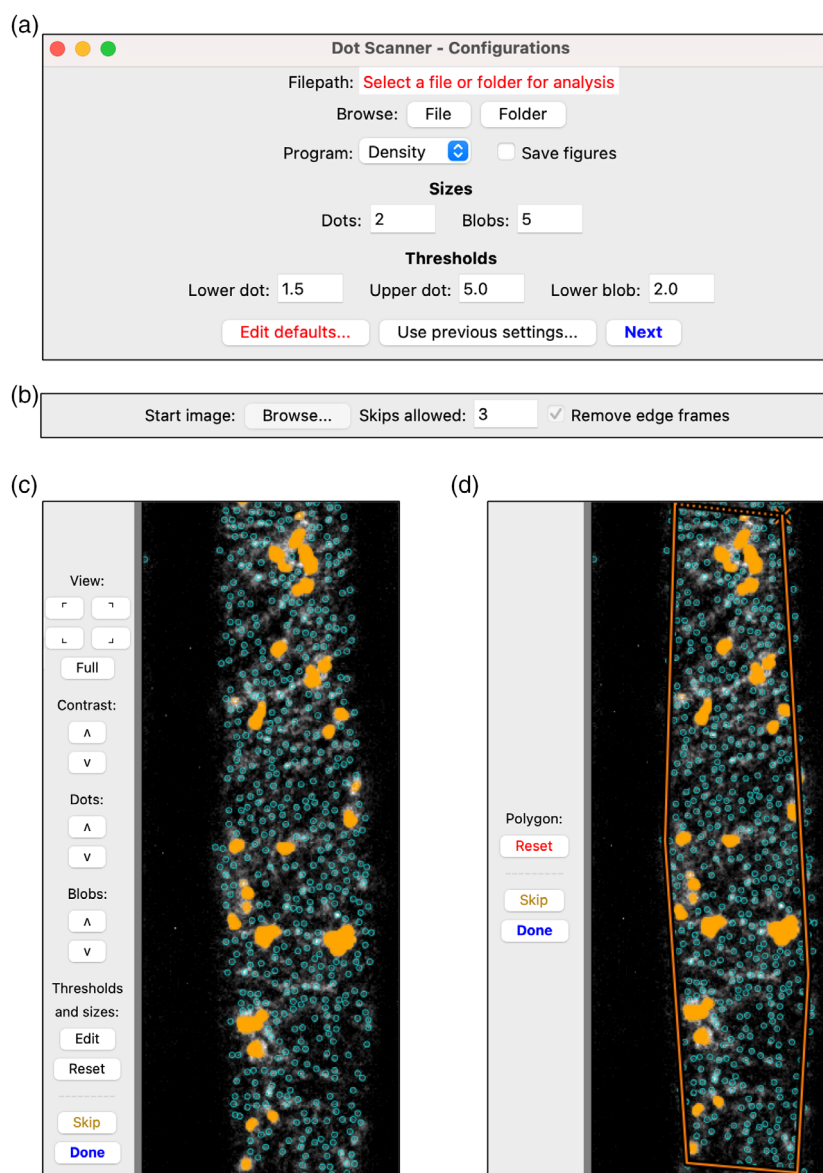


Figure 1. The Dot Scanner graphical user interface.

(a) Configuration screen with the 'Density' program selected. Users can select the 'File' or 'Folder' button next to browse to select the Filepath for where the data is stored to allow the program to access it. Users can also select whether to save figures with a check box and select the dot sizes and thresholds if known.

(b) When lifetime is selected some additional options appear.

(c) Threshold adjustment screen. Here the user can edit the dot and blob thresholds and sizes if they do not match what is shown on the screen by using the arrows by 'Dots' and 'Blobs' or by entering in values with the 'Edit' button. Users can also adjust the 'Contrast' and zoom in by selecting the arrows underneath 'View' to determine the dot and blob capture rate more accurately.

(d) Selecting the area of the image for analysis. The user clicks the image to start the polygon and then presses again to form a straight line. Once two lines are made the polygon will automatically fill in with a dotted line between the start and the end point.

selecting longer, brighter lines which can unintentionally inflate higher average lifetimes. Thirdly, Dot Scanner can automatically remove particles detected in the edge frames (the first and last image) from the analysis. When a particle is detected in the first frame of an image, it cannot be determined whether the particle existed before the first image was taken, so it may be better to exclude it from the lifetime analysis (and the same may also be true for

particles in the last frame). We often see this phenomenon in manual kymographs, where the start and end of a particle lifetime cannot be accurately assessed (Figure S1).

Dot Scanner allows users to 'skip' a set number of consecutive images in the lifetime analysis (Figure 1b), which can be useful for dimmer dots where an image or two in a series move out of focus, resulting in a false non-detection for those frames. Fluctuations in focus are often experienced

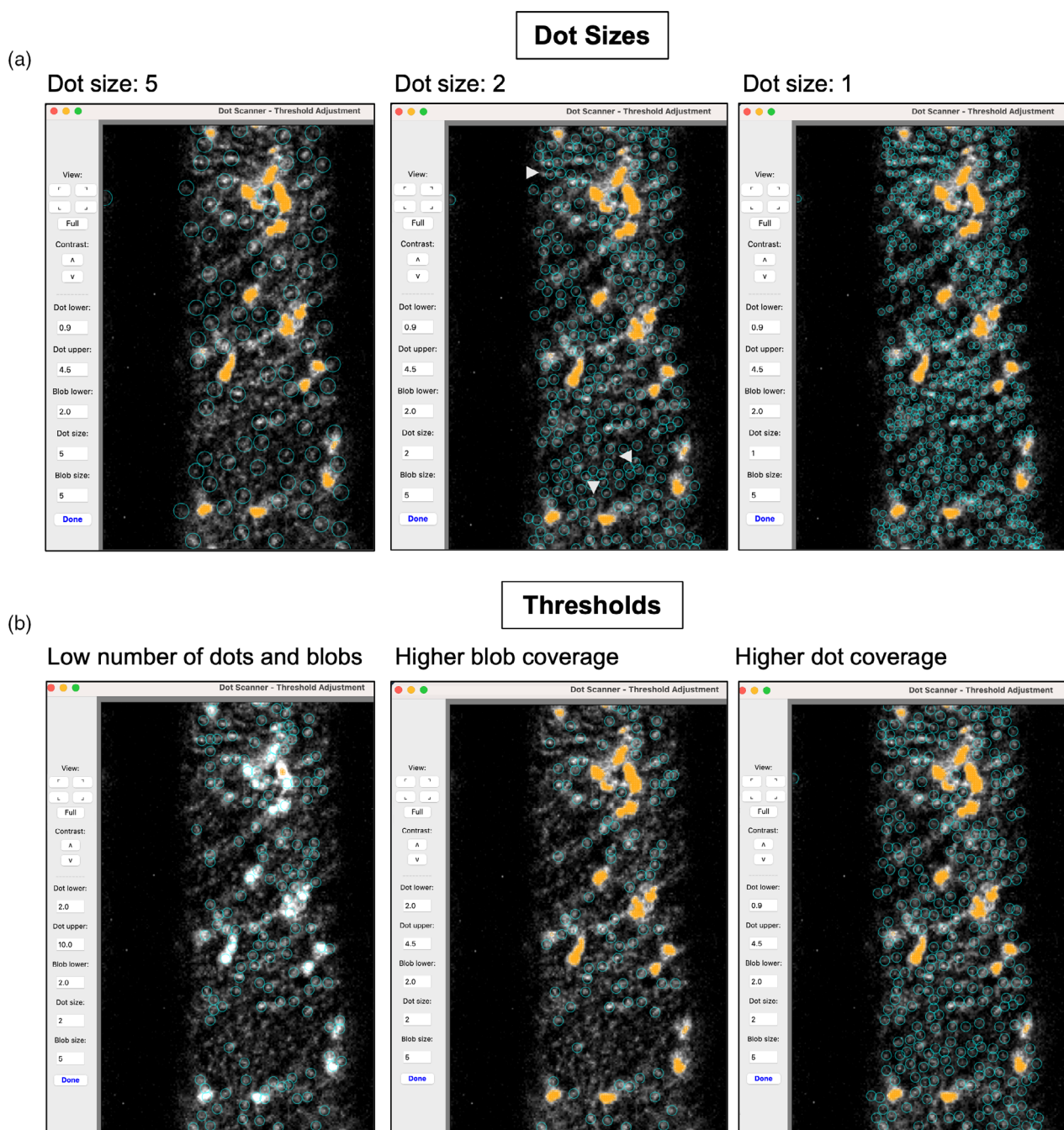


Figure 2. Adjusting the parameters to accurately capture punctate signals (dots) and aggregated signal (blobs).

Each panel shows an example of different dot and blob sizes (a) or thresholds (b), for capturing the YFP-CESA6 signal.

(a) Dot and blob sizes. Left panel: The dot size of five does not capture all the visible YFP-CESA6 punctate signal (cyan circles) as it is too wide. Middle panel: A smaller dot size of two captures most particles but is still missing some (white arrowheads). Right panel: A dot size of one captures more particles than a dot size of two but may capture more background.

(b) Thresholds. Left panel: The set 'Lower dot' and 'Upper dot' thresholds capture a low number of punctate dots (cyan circles) and aggregated blobs (orange circles). Middle panel: Decreasing the 'Upper dot' threshold, causes more aggregated blobs to be detected by Dot Scanner. Right panel: Decreasing the 'Lower dot' threshold, causes more punctate dots to be detected by Dot Scanner. The values for dot thresholds represent the number of standard deviations about the median brightness. The value for the blob threshold represents the number of times by which the upper dot threshold is multiplied.

when imaging live plant tissues at 100 $\times$ magnification, as very slight changes in movement can cause a displacement. To account for this temporary loss of focus, users must often

manually adjust the focus, which can take a few seconds. This can be seen in manual kymographs, where particles disappear and then reappear a few pixels later (Figure S1).

By increasing the number of skips allowed, these particles will be retained if they are back in focus and bright enough for detection in subsequent frames. In manual analysis, however, these may be counted as two separate particles. It is important that the appropriate number of skips selected for each experiment is carefully considered and based on the length of time intervals used. For instance, in a time series where images are taken every second, a skip of three frames (a total of 3 sec) is less consequential than if longer time points of 5 sec are used (a total of 15 sec). Since Dot Scanner's output files save the coordinates of every particle, users can manually check the lifetimes to see if the number of skips used is suitable.

Automation

Dot Scanner greatly decreases the time required to analyze images, particularly for particle lifetimes, as it can calculate particle densities and lifetimes automatically. The user does not have to calculate the size of the ROI, the number of particles per ROI, the length of the particle lifetimes, or particle displacements, significantly reducing the time taken for the analysis. When the user has selected the ROI on the region selection screen (Figure 1d), Dot Scanner calculates the particle densities, or lifetimes and displacements (depending on the program selected), and conveniently outputs it to a text file in the same directory in which the raw images are located.

Dot Scanner also reduces the time taken to analyze images by minimizing the amount of user input and automating the production of high-quality figures and films. Users can set specific analysis parameters and data output settings by interacting with the GUI (Figure 1) or by editing the configurations file. Once these parameters are set, Dot Scanner will perform the desired analyses on every image in the selected directory using the user-defined settings.

Additional features

In order to find the most optimal set of parameters for analyzing a set of images, users frequently rerun the analysis on the same set of images, changing the parameters each time. On the configuration screen, users can select the 'Use previous analysis' button (Figure 1a) to re-populate the parameters used from a previous analysis, including the ROI. This allows users to rerun their analysis using identical ROIs or any other parameters, saving time and enabling the user to directly compare the results of different analyses on the same image. In many cases, users may have images that have more than one channel, such as RGB (red, green, blue) images. The 'Use previous analysis' feature enables the same ROI to be used to analyze all channels from the same image set, allowing for direct comparisons (Figure S2).

On the configuration screen, users can select whether analyzed images should be saved. Various aspects of these

images can be tailored by editing the configuration file, accessed by clicking the 'Edit defaults' button (Figure 1a). For instance, the color of the identifiers for the dots and blobs, the thickness of the dot and blob markers, the presence of blobs or the ROI polygon, the output filetype of the image, and the image resolution can all be changed.

Dot Scanner can also produce images that track the detected particles throughout a time series (Figure S3), allowing the user to visualize the individual particle lifetimes (Movie S1)—something that is not currently possible with Fiji-ImageJ analysis. The output file provides the following information for every particle measured: lifetime (seconds), coordinates (x, y), and starting image file name (where the particle first appeared). This data output allows users to manually track any particle, if necessary (Figure S3). Furthermore, users can use Dot Scanner to plot the movement of cytosolic particles over the course of the time series (Movie S1), which can be used to make inferences about cytosolic trafficking.

Validation

To demonstrate the accuracy and usability of Dot Scanner, we analyzed a set of images using both Fiji-ImageJ and Dot Scanner and compared the results. We selected two markers for the quantification of live-cell imaging: YFP-CESA6, a common marker for CSCs, and CLC-mOrange, a common marker for clathrin-mediated endocytosis. Both of these markers produce a strong cytosolic or cortical Golgi signal, and a fainter punctate signal at the plasma membrane (Konopka et al., 2008; Paredes et al., 2006). We analyzed these tagged proteins in a wild-type/rescued line and a mutant line, *trs85-1*, so that we could compare the results to existing data (Allen et al., 2024). We performed these analyses double-blind by labeling the images with random numbers and having one author perform the manual analysis in Fiji-ImageJ while another performed the automated Dot Scanner analysis.

Assessing density of YFP-CESA6 particles at the plasma membrane

To directly compare the density analysis capabilities of Fiji-ImageJ and Dot Scanner, we generated images of YFP-CESA6 in *prc1-1* and *prc1-1 trs85-1* backgrounds and then calculated the density using both programs (Figure 3). We found that the densities of YFP-CESA6 particles reported by Dot Scanner ranged between 1 and 2 μm^2 (Figure 3c,d), similar to the Fiji-ImageJ analysis (Figure 3a,b) and previous studies (Polko et al., 2018; Zhu et al., 2018). Furthermore, YFP-CESA6 particles were significantly more abundant in the *prc1-1 trs85-1* background compared to *prc1-1* in both analyses (Figure 3), in accordance with published data (Allen et al., 2024). Therefore, in addition to being much faster and easier to use than Fiji-ImageJ, Dot Scanner also produces equally reliable results.

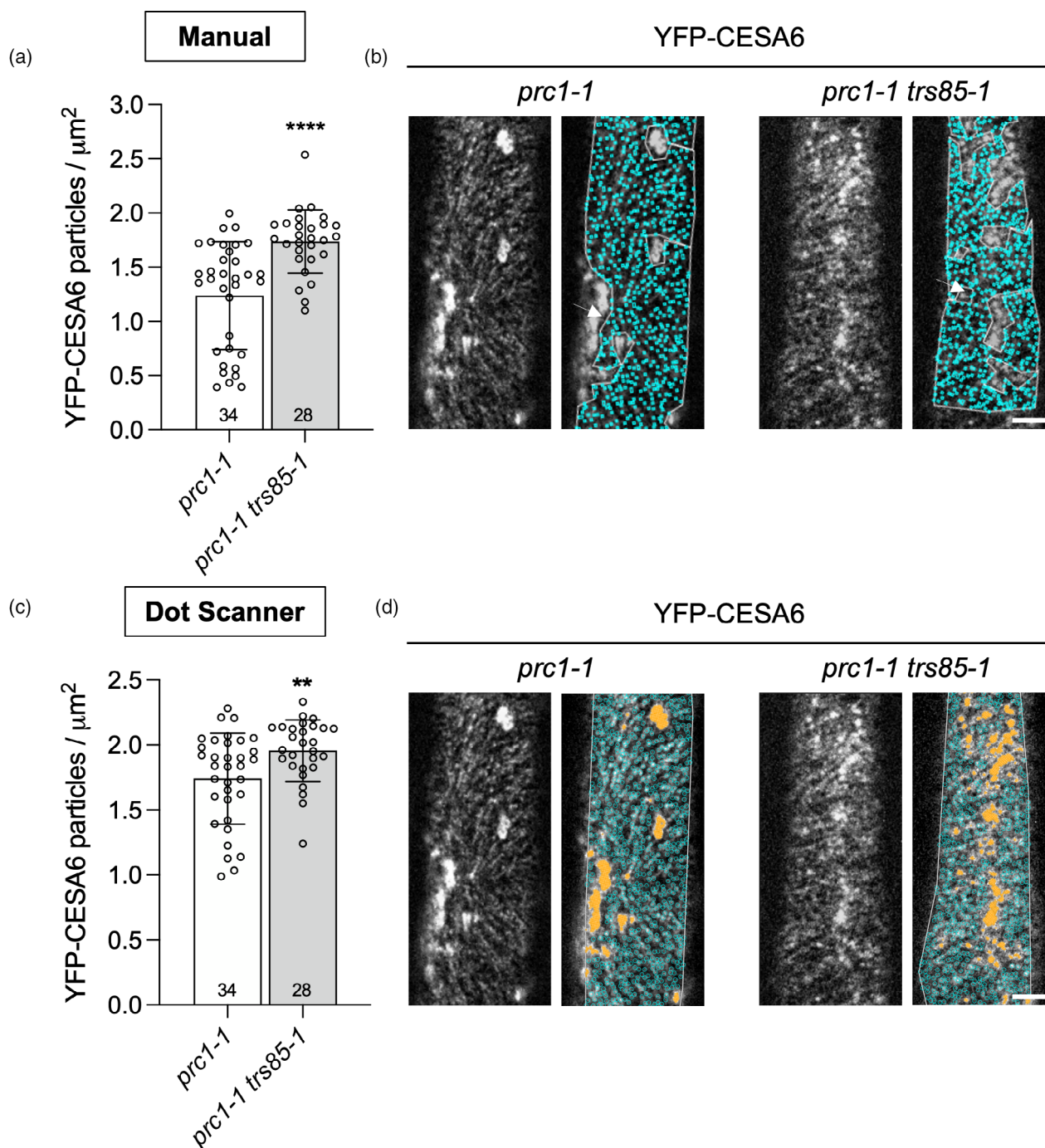


Figure 3. Measuring YFP-CESA6 density at the plasma membrane.

Density was measured (a) manually using Fiji-ImageJ and (c) using a Dot Scanner. Asterisks indicate significance based on a *T*-test (****, $P \leq 0.001$, **, $P \leq 0.01$). Numbers on the bars indicate number of images analyzed. Bars show mean $\pm$ standard deviation.

(b, d) Raw images of YFP-CESA6 particles used for analysis are shown on the left for each genotype, and annotated images for analysis are shown on the right. Cyan squares (b) and circles (d) represent punctate YFP-CESA6 labeled dots at the plasma membrane and orange circles (d) represent aggregated cytosolic YFP-CESA6 signal. White arrows indicate dark areas not included in the ROI (b) YFP-CESA6 particles were detected using the 'Find Maxima' function in Fiji (ImageJ) and the capture area (white) was drawn manually.

(d) YFP-CESA6 particles were detected using user-defined thresholds in Dot Scanner and the capture area (white) was drawn manually. Pixels marked with orange circles were omitted from the calculation of the total area. Scale bars indicate 5 μm .

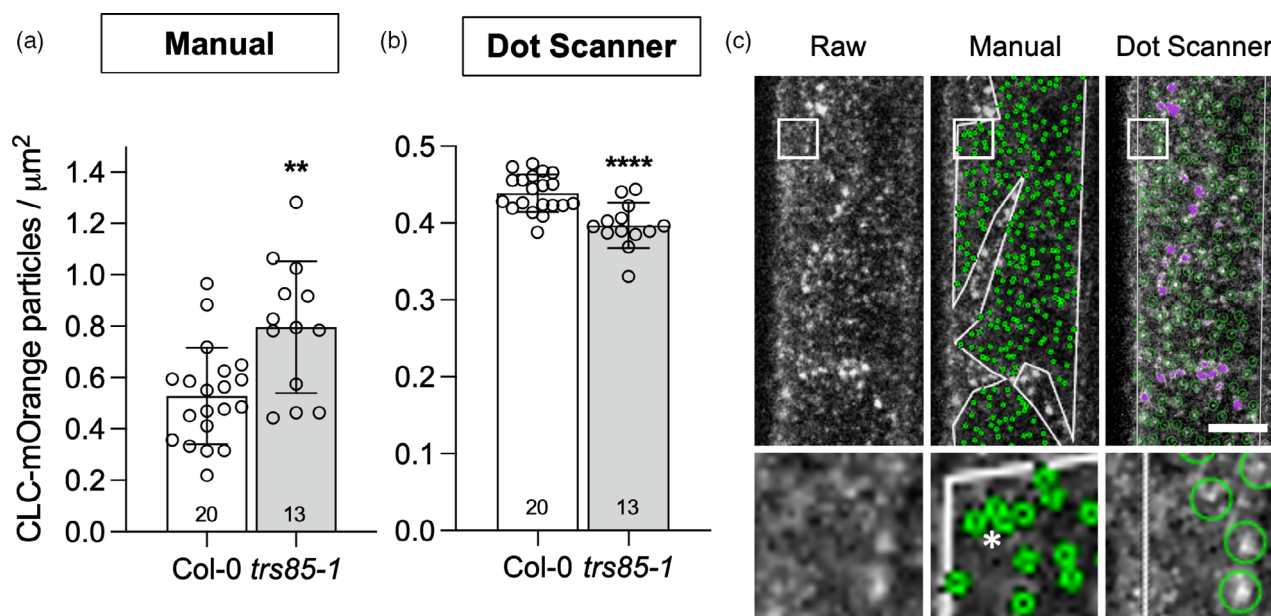


Figure 4. Measuring CLC-mOrange density at the plasma membrane.

Density was measured (a) manually using Fiji-ImageJ and previously with (b) Dot Scanner. Asterisks indicate significance based on a *T*-test (****, $P \leq 0.001$, **, $P \leq 0.01$). Numbers on the bars indicate sample size. Bars show mean $\pm$ standard deviation.

(b) This figure has been reproduced with permission from (Allen et al., 2024). (c) Detection of dots in a representative CLC-mOrange, *trs85-1* image with different programs. From left to right: raw image, image analyzed using Fiji-ImageJ, and image analyzed using Dot Scanner. Green circles and squares represent punctate CLC-mOrange labeled dots at the plasma membrane and purple circles represent cortical Golgi CLC-mOrange signal. Pixels marked with purple circles were omitted from the calculation of the total area in the previous analysis (Allen et al., 2024). Dots marked with an asterisk are likely background signal as they are not discernable as punctate dots in the raw image. Scale bars indicate 5 μm .

Assessing density of CLC-mOrange particles at the plasma membrane

We also compared the density of CLC-mOrange particles at the plasma membrane using Fiji-ImageJ and Dot Scanner, since these particles often present more background signal due to higher photobleaching during imaging. We compared our previous analysis of CLC-mOrange density in wild-type and *trs85-1* performed with Dot Scanner (Allen et al., 2024) by calculating particle density using Fiji-ImageJ on the same set of images. We found that the densities of CLC-mOrange particles reported by Fiji-ImageJ ranged between 0.3 and 1.0 μm^2 (Figure 4a), similar to previous studies (Figure 4b) (Allen et al., 2024; Bashline et al., 2015).

Interestingly, we found that the density of CLC-mOrange was 50% higher in *trs85-1* than the wild-type (Figure 4a), whereas a previous Dot Scanner analysis found that the density of CLC-mOrange was lower in *trs85-1* (Figure 4b) (Allen et al., 2024). This difference is mainly caused by the high amount of background signal we observed in CLC-mOrange *trs85-1* that is mistakenly detected as punctate particles in the Fiji-ImageJ analysis. By examining the analyzed images closely, it is clear that Fiji-ImageJ incorrectly detects background signal as particles (Figure 4c). This shows that Dot Scanner can more accurately detect weak signals in images with high background. Typically, density analysis in

Fiji-ImageJ involves altering one threshold to determine particle detection, whereas Dot Scanner provides three separate thresholds for brightness and two for particle size, providing greater sensitivity for accurate particle detection.

Assessing the lifetime of CLC-mOrange particles at the plasma membrane

We next assessed the accuracy of the lifetime program in Dot Scanner by measuring CLC-mOrange particle longevities at the plasma membrane. For the Dot Scanner analysis, we removed particles detected in edge frames and allowed for three skipped images taken at one-second intervals. Since CLC-mOrange particles do not tend to migrate in the membrane, we also removed particles that had a total displacement of over 10 pixels from the final analysis. Approximately 40% of the lifetimes recorded (Col-0 = 8721/13911 and *trs85-1* = 6027/9869) were 1 sec long. We thought that these were unlikely to represent true endocytosis events based on previous studies (Konopka et al., 2008; Narasimhan et al., 2020), and so these were removed from the analysis. We also found that there was a one-second increase in average CLC-mOrange lifetimes in *trs85-1* compared to the wild-type in Dot Scanner (Figure 5b), similar to our previous analysis (Allen et al., 2024). There was also a one-second increase in CLC-mOrange lifetimes in *trs85-1* in the Fiji-ImageJ analysis (Figure 5a).

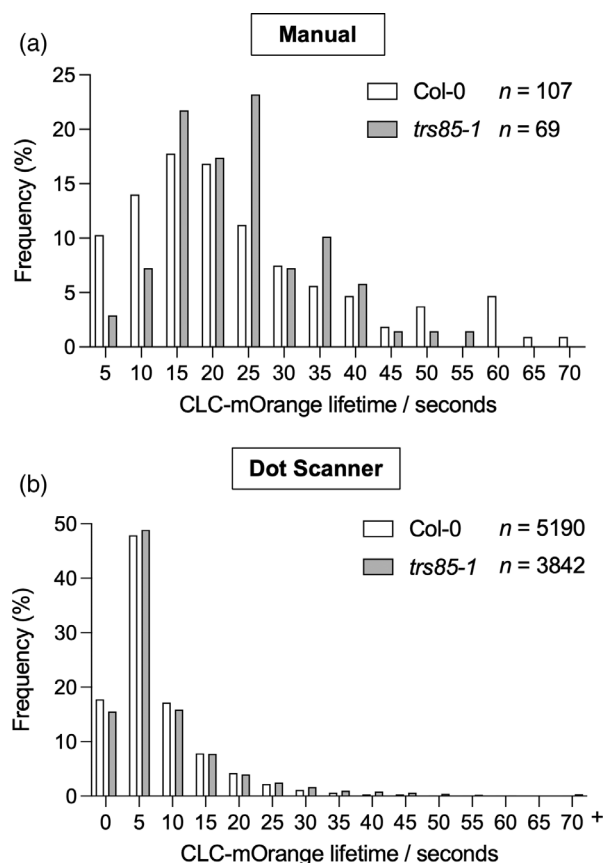


Figure 5. Measuring the lifetime of CLC-mOrange particles. Normalized histogram of CLC-mOrange particle lifetimes.

(a) Average CLC-mOrange lifetimes from Fiji-ImageJ analysis: CLC-mOrange, Col-0 = 23 ± 15 SD seconds ($n = 107$, 7 films) and, CLC-mOrange, *trs85-1* = 24 ± 10 SD seconds ($n = 69$, 6 films).

(b) Average CLC-mOrange lifetimes from Dot Scanner analysis: CLC-mOrange, Col-0 = 8 ± 3 sec ($n = 5190$, 6 films) and CLC-mOrange, *trs85-1* = 9 ± 3 sec ($n = 3842$, 5 films). The final bin represents bins 70–105 sec.

Overall, the average CLC-mOrange lifetimes reported by Dot Scanner are 8–9 sec, (Figure 5b), which is approximately 50% lower than the lifetimes reported by Fiji-ImageJ (Figure 5a). Previous studies have reported that CLC lifetimes are ~20 sec based on Fiji-ImageJ analysis (Konopka et al., 2008) and a MATLAB program (Narasimhan et al., 2020). We hypothesize that CLC-mOrange lifetimes are shorter when analyzed with Dot Scanner (compared to previous studies and our Fiji-ImageJ analysis) due to fundamental differences in the data analysis. For example, the Dot Scanner is much more sensitive in detecting short-lived particles of only a few seconds that are much harder to detect by the eye in kymographs (Figure S1). In manual kymograph analysis performed in Fiji-ImageJ, there is natural bias towards selecting longer, more prominent lines, which positively skews the average lifetimes.

Indeed, the shortest lifetime we selected in our Fiji-ImageJ analysis was 4 sec (Figure 5a), when almost half

the lifetimes detected by Dot Scanner were under 4 sec (Figure 5b). This supports the hypothesis that a significant proportion of short-lived particles are missed in manual analyses in Fiji-ImageJ because they are harder to detect by eye. Additionally, in the study by the Narasimhan group, all CLC lifetimes detected in five frames (5 sec) or fewer were excluded from the analysis (Narasimhan et al., 2020), which increased the average lifetimes.

Quantifying the velocity of YFP-CESA6 particles at the plasma membrane

In addition to calculating the lifetimes of individual particles, Dot Scanner can also calculate the total displacement of those particles over time. By combining these two sets of analyses, individual particle velocities can be calculated. We used a Dot Scanner to obtain the lifetime and displacements of YFP-CESA6 particles at the plasma membrane in the *prc1-1* background (Figure S4). We set a stringent lifetime threshold of 12 frames, which excluded any particle lifetimes that were less than a minute long and unlikely to represent true YFP-CESA6 particles. Overall, we found that the average velocity reported by Dot Scanner was 358.2 ± 265.4 SD nm min⁻¹ (Figure S4), similar to previous findings (Paredes et al., 2006; Xin et al., 2023).

Applications

Dot Scanner is an open-source program that can be used to quantify the densities, lifetimes, and displacements of fluorescently tagged particles in confocal microscopy images. It is highly accessible to a wide range of users of different disciplines, as it is free to use and does not require a knowledge of programming to operate. We have shown that Dot Scanner outperforms manual analysis achieved with Fiji-ImageJ in terms of reducing biases that come from manual input, detecting weaker signals more accurately, and decreasing the time taken for the analysis.

Dot Scanner provides an efficient analysis pipeline by automating the calculation of densities, lifetimes, and displacements, and the production of high-quality figures. The number of lifetimes calculated in the Dot Scanner analysis was over 50-fold higher than in Fiji-ImageJ (Figure 5) and was achieved with very little user input or bias. In fact, given the time required to perform the Fiji-ImageJ analysis, calculating the same number of lifetimes with Fiji-ImageJ would take weeks, as opposed to under an hour with Dot Scanner.

Dot Scanner also accurately detects many of the short-lived particles that are likely missed from manual analysis, which will have implications for our understanding of protein behaviors observed during live cell imaging. For instance, our lifetime analysis of CLC-mOrange indicates that shorter CLC lifetimes are more common than previously reported. As CLC particle lifetimes are used as a proxy for the formation and scission of clathrin-coated pits

(Konopka et al., 2008), our Dot Scanner analysis indicates this process occurs much more rapidly than previously thought.

We envision that Dot Scanner can be applied to other fluorescent particles, different tissues, and perhaps different organisms. This program will make a notable contribution to the analysis of confocal images, particularly for lifetimes that are time-consuming to generate and more subject to inherent biases. We expect that Dot Scanner will be a highly versatile program that will continue to be adapted for additional purposes, as users submit feature requests or submit their own code changes to the project. For any issues that may arise in the program's current state, we have included some troubleshooting solutions (Figure S5).

EXPERIMENTAL PROCEDURES

Plant growth conditions

All fluorescently tagged Arabidopsis lines used (YFP-CESA6 prc1-1, YFP-CESA6 prc1-1 trs85-1, CLC-mOrange Col-0, and CLC-mOrange trs85-1) have been previously described (Allen et al., 2024). Seeds were sterilized using 30% bleach (w/v) for 10 min, after rinsing with autoclaved water before placing on a ½ MS plate for germination in the dark. Dark-grown seedlings were grown for 2.5 days in 16-h light/8-h dark cycle at 22°C.

Live-cell imaging conditions

2.5-day-old etiolated seedlings were mounted with ddH₂O and positioned between two coverslips for imaging. All images and movies were taken of the epidermal cells approximately 0.5–2 mm below the apical hook using a Yokogawa CSUX1 spinning disk system as described previously (Xin et al., 2023). CLC-mOrange films were taken at 1 sec intervals over a total period of 2 min and YFP-CESA6 films were taken at 5 sec intervals over a total period of 5 min. A three-line laser merger with 445 nm, 488 nm, and 561 nm lasers with band-pass filters for emission filtering and images were captured using Metamorph (Molecular Devices, San Jose, CA, USA). Images were analyzed with either Fiji-ImageJ (version 2.3.0/1.53q) or Dot Scanner.

Image analysis in Fiji-ImageJ

All images were assigned to random numbers and the analysis was performed by an author that did not use Dot Scanner so that the analysis could be performed double-blind. For YFP-CESA6 and CLC-mOrange density analysis, images were converted to 8-bit and an ROI was manually drawn. The 'Find Maxima' function was used to detect particles and a threshold of 8 and 30 was used for YFP-CESA6 and CLC-mOrange, respectively. The dataset used for the CLC-mOrange was the same as described previously using Dot Scanner (Allen et al., 2024). For CLC-mOrange lifetime analysis, the contrast of images was enhanced by 0.3% and the background was subtracted using a rolling ball radius of 50 pixels. Kymographs were generated manually on stacks of images, and the length of different visible lines was measured using the line tool.

Image analysis in Dot Scanner

All images were assigned to random numbers and the analysis was performed by an author that did not perform the Fiji-ImageJ

analysis. For YFP-CESA6 density analysis, the following parameters were used: *dot size* = 1, *blob size* = 5, *dot lower* = 0.9, *dot upper* = 4.5, and *blob lower* = 2. For the CLC-mOrange density analysis the following parameters were used: *dot size* = 1, *blob size* = 5, *dot lower* = 0.8, *dot upper* = 4.5, and *blob lower* = 2. The dataset used for the CLC-mOrange was the same as described previously but different Dot Scanner parameters were used (Allen et al., 2024). For CLC-mOrange lifetime analysis, the following parameters were used: *dot size* = 3, *blob size* = 5, *dot lower* = 2, *dot upper* = 4.5, *blob lower* = 2, *skips* = 3, and *remove edge frames* = true. All lifetimes that were 1 sec long and/or had an average displacement of over 10 pixels were removed from the analysis. For YFP-CESA6 velocity analysis, the following parameters were used: *dot size* = 2, *blob size* = 5, *dot lower* = 0.9, *dot upper* = 4.5, and *blob lower* = 2, *skips* = 1, and *remove edge frames* = false. All lifetimes that were 60 sec long were removed from the analysis.

Dot Scanner

Dot Scanner was developed using the Python programming language. The software is available on GitHub, and the project homepage (<https://github.com/bdavis222/dotscanner>) contains all the documentation needed for its installation and use, including the README file (<https://github.com/bdavis222/dot-scanner/blob/main/README.md>). As mentioned in the README, Python 3 must be installed prior to Dot Scanner installation (<https://www.python.org/downloads/>).

ACKNOWLEDGMENTS

We thank S. Bednarek for providing the CLC-mOrange line. This work was supported by the National Science Foundation (NSF#1951007).

SUPPORTING INFORMATION

Additional Supporting Information may be found in the online version of this article.

Figure S1. Examples of errors frequently made in manual lifetime analysis that can be captured accurately with Dot Scanner. Representative kymograph images generated manually in Fiji-ImageJ. Left panel: Particles with small lifetimes are more likely to be missed by eye during manual quantification. Arrows indicate particle lifetimes of 3 sec. Middle panel: Particles may shift out of focus during film capture and may look like two separate particles. A particle that disappears from frame for 3 sec before reappearing is represented as a broken line (white box). Right panel: Particles that appear before the film starts (top arrow) and particles that disappear after the film ends should not be counted as the true length of the particle lifetime is unknown (bottom arrow).

Figure S2. Analyzing different channels from GFP-CSI3 RFP-CSI1 image sets, using the 'Use previous settings' feature. Images of GFP-CSI3 particles taken in the green channel are shown in the left panel and images of RFP-CSI1 particles taken in the red channel are shown in the right panel. In each panel, raw images used for analysis are shown on the left, and annotated images for analysis are shown on the right. Both raw images were taken simultaneously at the same position. By using the 'Use previous settings' feature, the ROI that was drawn for the GFP-CSI3 analysis (left) was automatically applied to the RFP-CSI1 analysis (right). Cyan and green circles represent punctate GFP-CSI3 and RFP-CSI1 dots at the plasma membrane respectively, and orange and purple circles represent aggregated cytosolic GFP-CSI3 and RFP-CSI1 signal, respectively. Scale bar indicates 5 µm.

Figure S3. Manually tracking particles from lifetime analyses. A representative example of the data (left panel) and figures produced (right panel) from lifetime analysis performed by Dot Scanner. Right panel: Three examples of dot coordinates, lifetimes, and starting image are highlighted and numbered. Left panel: Two sequential images from a time series are shown: t5 (top) and t6 (bottom). Green circles represent punctate CLC-mOrange labeled dots at the plasma membrane. Each axis is a total of 512 pixels long. Numbered dots from the right panel are indicated on images t5 and t6 with white arrows.

Figure S4. Measuring the velocity of YFP-CESA6 particles. Normalized histogram of YFP-CESA6 velocity. Velocities were measured by combining the lifetime and displacement data for each particle. Particles with a lifetime of 65 sec or higher that were retained in the final analysis. A total of five films of 5 min long were analyzed.

Figure S5. Potential issues and troubleshooting solutions for using Dot Scanner. A list of potential issues that may arise during the use and installation of Dot Scanner, and their corresponding solution(s). If the issue cannot be solved, users can submit the problem to (<https://github.com/bdavis222/dot-scanner/issues>).

Movie S1. The dynamics of CLC-mOrange particles from images processed with Dot Scanner. Small punctate signals circled in green represent CLC-mOrange particles at the plasma membrane and particles highlighted in purple represent cortical Golgi particles. Users can decide whether to exclude the labeling of cytosolic particles (top panel) or include them (lower panel). Images were taken of 2.5-day-old dark-grown hypocotyls at 1 sec intervals over a total of 2 min, with confocal microscopy. Frame rate = 7 fps. Scale bars indicate 10 μ m.

REFERENCES

- Allen, H., Wei, D., Gu, Y. & Li, S. (2021) A historical perspective on the regulation of cellulose biosynthesis. *Carbohydrate Polymers*, **252**, 117022.
- Allen, H., X., Zhu, X., Li, S. & Gu, Y. (2024). The TRAPP3 subunit, Trs85, has a dual role in the trafficking of cellulose synthase complexes in Arabidopsis. *Plant Journal*, *Accepted*.
- Bashline, L., Li, S., Anderson, C.T., Lei, L. & Gu, Y. (2013) The endocytosis of cellulose synthase in *Arabidopsis* is dependent on mu2, a clathrin-mediated endocytosis adaptin. *Plant Physiology*, **163**(1), 150–160.
- Bashline, L., Li, S., Zhu, X. & Gu, Y. (2015) The TWD40-2 protein and the AP2 complex cooperate in the clathrin-mediated endocytosis of cellulose synthase to regulate cellulose biosynthesis. *Proceedings of the National Academy of Sciences of the United States of America*, **112**(41), 12870–12875.
- Boudaoud, A., Burian, A., Borowska-Wykret, D., Uyttewaald, M., Wrzalik, R., Kwiatkowska, D. *et al.* (2014) FibrilTool, an ImageJ plug-in to quantify fibrillar structures in raw microscopy images. *Nature Protocols*, **9**(2), 457–463.
- Grossmann, G., Krebs, M., Maizel, A., Stahl, Y., Vermeer, J.E.M. & Ott, T. (2018) Green light for quantitative live-cell imaging in plants. *Journal of Cell Science*, **131**(2), 1–14.
- Konopka, C.A., Backues, S.K. & Bednarek, S.Y. (2008) Dynamics of *Arabidopsis* dynamin-related protein 1C and a clathrin light chain at the plasma membrane. *Plant Cell*, **20**(5), 1363–1380.
- Li, S., Bashline, L., Zheng, Y., Xin, X., Huang, S., Kong, Z. *et al.* (2016) Cellulose synthase complexes act in a concerted fashion to synthesize highly aggregated cellulose in secondary cell walls of plants. *Proceedings of the National Academy of Sciences of the United States of America*, **113**(40), 11348–11353.
- Narasimhan, M., Johnson, A., Prizak, R., Kaufmann, W.A., Tan, S., Casillas-Perez, B. *et al.* (2020) Evolutionarily unique mechanistic framework of clathrin-mediated endocytosis in plants. *eLife*, **9**, 1–30.
- Paredes, A.R., Somerville, C.R. & Ehrhardt, D.W. (2006) Visualization of cellulose synthase demonstrates functional association with microtubules. *Science*, **312**(5779), 1491–1495.
- Polko, J.K., Barnes, W.J., Voiniciuc, C., Doctor, S., Steinwand, B., Hill, J.L., Jr. *et al.* (2018) SHOU4 proteins regulate trafficking of cellulose synthase complexes to the plasma membrane. *Current Biology*, **28**(19), 3174–3182 e3176.
- Schindelin, J., Rueden, C.T., Hiner, M.C. & Eliceiri, K.W. (2015) The ImageJ ecosystem: an open platform for biomedical image analysis. *Molecular Reproduction and Development*, **82**(7–8), 518–529.
- Schneider, C.A., Rasband, W.S. & Eliceiri, K.W. (2012) NIH image to ImageJ: 25 years of image analysis. *Nature Methods*, **9**(7), 671–675.
- Xin, X., Wei, D., Lei, L., Zheng, H., Wallace, I.S., Li, S. *et al.* (2023) CALCIUM-DEPENDENT PROTEIN KINASE32 regulates cellulose biosynthesis through post-translational modification of cellulose synthase. *BioRxiv*, **239**, 2212–2224.
- Zhu, X., Li, S., Pan, S., Xin, X. & Gu, Y. (2018) CSI1, PATROL1, and exocyst complex cooperate in delivery of cellulose synthase complexes to the plasma membrane. *Proceedings of the National Academy of Sciences of the United States of America*, **115**(15), E3578–E3587.