

Jan 19, 2026

## Cytosoftware-RPQ: A Kinetics-Aware Computational Workflow for Red Particle Quantification

DOI

<https://dx.doi.org/10.17504/protocols.io.x54v9bp5ql3e/v1>

Kanokporn Sornsuwan<sup>1,2</sup>, Weeraya Thongkum<sup>1,2</sup>, Umpa Yasamut<sup>2,3,4</sup>, Kittichai Wantanajittikul<sup>5</sup>, Chatchai Tayapiwatana<sup>2,3,4</sup>

<sup>1</sup>Office of Research Administration, Chiang Mai University, Chiang Mai, Thailand;

<sup>2</sup>Center of Biomolecular Therapy and Diagnostic, Faculty of Associated Medical Sciences, Chiang Mai University, Chiang Mai, Thailand;

<sup>3</sup>Division of Clinical Immunology, Department of Medical Technology, Faculty of Associated Medical Sciences, Chiang Mai University, Chiang Mai, Thailand;

<sup>4</sup>Center of Innovative Immunodiagnostic Development, Department of Medical Technology, Faculty of Associated Medical Sciences, Chiang Mai University, Chiang Mai, Thailand;

<sup>5</sup>Department of Radiologic Technology, Faculty of Associated Medical Sciences, Chiang Mai University, Chiang Mai, Thailand

Kittichai Wantanajittikul: Corresponding author; kittichai.wan@cmu.ac.th;

Chatchai Tayapiwatana: Corresponding author; chatchai.t@cmu.ac.th;



kanokporn.sornsuwan

### Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



**DOI:** <https://dx.doi.org/10.17504/protocols.io.x54v9bp5ql3e/v1>

**Protocol Citation:** Kanokporn Sornsuwan, Weeraya Thongkum, Umpa Yasamut, Kittichai Wantanajittikul, Chatchai Tayapiwatana 2026. Cytosoftware-RPQ: A Kinetics-Aware Computational Workflow for Red Particle Quantification. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.x54v9bp5ql3e/v1>

**Manuscript citation:**

This protocol is a part of the manuscript entitled "**CytoSoftware-RPQ: single-particle quantification of phagolysosomal acidification dynamics in live-cell imaging**" (Manuscript ID: PONE-D-26-02644) submitted to PLoS One

**License:** This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

**Protocol status:** Working

**We use this protocol and it's working**

**Created:** January 19, 2026

**Last Modified:** January 19, 2026

**Protocol Integer ID:** 238870

**Keywords:** aware computational workflow for red particle quantification, red particle quantification, red fluorescent particle, cytosoftware, algorithm for blob detection, blob detection, quantitative analysis of fluorescent particle, microscopy data, fluorescent particle, series microscopy data, based python workflow, python workflow, kinetic analysis, detailed technical overview of rpq, aware computational workflow, rpq system, kinetic, detection setting, rpq, core of the rpq system

## Abstract

The quantitative analysis of fluorescent particles in time-series microscopy data presents a significant scientific challenge, requiring robust methods to distinguish genuine signals from background noise and artifacts across thousands of frames. To address this, the Red Particle Quantification (RPQ) system has been developed as a specialized, Jupyter-based Python workflow. Its primary function is to detect and count candidate red fluorescent particles within standard MP4 video files, providing researchers with a powerful tool for kinetic analysis.

The core of the RPQ system is built on a layered architecture that ensures maintainability and clarity. It features a robust, multi-stage image processing pipeline centered on a sophisticated Laplacian-of-Gaussian (LoG) algorithm for blob detection. This processing core is coupled with an interactive parameterization workflow, which empowers users to visually tune detection settings for their specific imaging conditions before committing to a full analysis run. The purpose of this white paper is to provide a detailed technical overview of RPQ's architecture, methodology, and capabilities for an audience of scientists and software engineers seeking to understand and leverage its functionality.

# Guidelines

## 3. Interactive Workflow and Parameterization

The effectiveness of the RPQ system relies on its interactive workflow, which empowers the user to precisely tune the detection pipeline for their specific imaging conditions and particle characteristics. Rather than requiring users to guess parameters, RPQ provides immediate visual feedback, ensuring that the chosen settings are appropriate before launching a time-consuming analysis of the full video.

### 3.1 User workflow

The primary user workflow follows four distinct steps:

- 1) **Load Video:** The user begins by loading an MP4 video file containing the time-series microscopy data.
- 2) **Configure Parameters:** Using the interactive *ipywidgets* interface, the user adjusts the key detection parameters that govern the image processing pipeline.
- 3) **Apply and Preview:** The user applies the current settings to a single, representative frame of the video. The system provides immediate visual feedback by displaying the frame with colored circles overlaid on each detected particle. This step is crucial for iteratively tuning the parameters until detection is accurate.
- 4) **Run Analysis:** Once the user is satisfied with the preview, they initiate the full analysis. This step processes the entire video using the validated parameters, generating the final output

### 3.2 Parameter tuning guidance

CytoSoftware-RPQ provides several adjustable parameters that control particle detection, tracking, and quantitative signal extraction. While default settings are optimized for typical live-cell phagolysosome imaging conditions, variations in signal intensity, particle size, and background noise may require parameter adjustment to achieve optimal performance.

This section offers practical guidance on how to tune key parameters, explains their technical purpose. Users are encouraged to iteratively refine parameters to balance sensitivity and specificity, ensuring robust and reproducible single-particle measurements across experiments. The table below details the key detection parameters available to the user.

**Table 1.** Adjustable parameters and their technical purpose for red particle quantification in CytoSoftware-RPQ

|  | A          | B                         | C                                                             |
|--|------------|---------------------------|---------------------------------------------------------------|
|  | Background | BG sigma / BG_BLUR_SI GMA | Sets the Gaussian sigma for background estimation used in red |



|  | A             | B                            | C                                                                                             |
|--|---------------|------------------------------|-----------------------------------------------------------------------------------------------|
|  |               |                              | channel enhancement.                                                                          |
|  | Red filtering | Red dom / RED_DOM_RATIO      | Defines the required ratio for red channel intensity over green and blue.                     |
|  | Red filtering | Red min abs / RED_MIN_ABS    | Sets the minimum absolute intensity (0–1 scale) for the red-enhanced signal to be considered. |
|  | LoG blobs     | $\sigma$ min / LOG_SIGMA_MIN | Defines the smallest particle size (scale) for the LoG detector to search for.                |
|  | LoG blobs     | $\sigma$ max / LOG_SIGMA_MAX | Defines the largest particle size (scale) for the LoG detector to search for.                 |
|  | LoG blobs     | # $\sigma$ / LOG_NUM_SIGMAS  | Controls the number of size steps to search between LOG_SIGMA_MIN and LOG_SIGMA_MAX.          |
|  | LoG blobs     | LoG thr / LOG_THRESHOLD      | The base sensitivity threshold for the LoG response; lower values are more sensitive.         |
|  | LoG blobs     | Robust k / ROBUST_K          | Multiplier for the Median                                                                     |

|  | A              | B                            | C                                                                                           |
|--|----------------|------------------------------|---------------------------------------------------------------------------------------------|
|  |                |                              | Absolute Deviation (MAD) to create a dynamic, frame-adaptive threshold.                     |
|  | Size filter    | Area min / MIN_PARTICLE_AREA | Rejects detected blobs with a pixel area smaller than this value.                           |
|  | Size filter    | Area max / MAX_PARTICLE_AREA | Rejects detected blobs with a pixel area larger than this value.                            |
|  | Quality filter | SNR min / SNR_MIN            | Sets the minimum signal-to-noise ratio for a detection to be considered valid (0 disables). |

Recommendation: Based on the interactive preview, users can systematically optimize detection accuracy:

**1) To Reduce False Positives:** If the system is detecting noise or artifacts, the recommended adjustments are to increase RED\_DOM\_RATIO, RED\_MIN\_ABS, LOG\_THRESH, or MIN\_PARTICLE\_AREA.

**2) To Increase Detection of Missed Particles:** If the system is failing to detect valid particles, the recommended adjustments are to decrease LOG\_THRESH, ROBUST\_K, or RED\_MIN\_ABS. For very small particles, reducing LOG\_SIGMA\_MIN may also be necessary.

## Materials

- 1) **Python: Version 3.9+ (3.9–3.11 recommended)**
- 2) **Jupyter: Jupyter Notebook or JupyterLab with *ipwidgets* enabled**
- 3) **Core Libraries:**
  - opencv-python-headless
  - scikit-image
  - numpy
  - pandas
  - matplotlib
  - tqdm
  - ipywidgets

## System Architecture and Technical Stack

- 1 The RPQ system is implemented as a three-part functional organization, with each layer leveraging industry-standard Python libraries to perform its specific function:
  - 1) Interactive UI Layer: This top layer is built using Jupyter ipywidgets. It provides a graphical user interface directly within the Jupyter Notebook environment, enabling users to load data and adjust analysis parameters with interactive sliders, text boxes, and buttons for an intuitive tuning experience.
  - 2) Image Processing Layer: This is the core analytical engine of the RPQ system. It leverages the high-performance computer vision capabilities of OpenCV and the advanced algorithmic functions of scikit-image to execute the multi-stage particle detection pipeline.
  - 3) Data and Plotting Layer: Responsible for all data handling and visualization, this layer uses NumPy for efficient numerical operations, pandas for structuring and exporting tabular data (CSVs), and matplotlib for generating plots and visual feedback.

### 1.1

## The RPQ Image Processing Pipeline

- 2 The methodological heart of the RPQ system is its image processing pipeline. This multi-stage process is meticulously designed to systematically isolate and identify target fluorescent particles while rejecting the noise and background artifacts that are common in fluorescence microscopy. Each frame of the input video is subjected to a sequence of filtering and detection steps, transforming the raw pixel data into a set of qualified particle detections.
- 3 **2.1 Background Estimation and Enhancement:** The first step in the pipeline is to estimate and account for non-uniform background fluorescence. A Gaussian blur, governed by the **BG\_BLUR\_SIGMA** parameter, is applied to the frame to create a smooth background model. This model represents low-frequency spatial variations in intensity and is subtracted from the red channel to enhance foreground particles and correct for uneven illumination.
- 4 **2.2 Red-Dominance Filtering:** To achieve high specificity for the target particles, RPQ employs a critical color-based filtering stage. This stage uses a two-criteria system to identify pixels that are truly "red dominant" and not just broadly fluorescent:
  - 1) Relative Dominance: The **RED\_DOM\_RATIO** parameter ensures that a pixel's intensity in the red channel is significantly greater than its intensity in the green and blue channels. This ratio-based check is crucial for rejecting white or yellow-tinged artifacts.

2) Absolute Intensity: The **RED\_MIN\_ABS** parameter establishes a minimum threshold for the red-enhanced intensity value. This prevents low-signal noise, which might coincidentally satisfy the dominance ratio, from being considered a valid part of a particle.

- 5 **2.3 Particle Detection via Laplacian-of-Gaussian (LoG):** With the image pre-processed, RPQ uses the Laplacian-of-Gaussian (LoG) method for particle detection. LoG is a robust blob detection algorithm ideal for identifying bright, circular objects of varying sizes on a dark background. The LoG method is particularly effective for this application due to its inherent scale-invariance—achieved by searching across a range of sigmas—and its function as a band-pass filter, making it excellent at identifying objects of a specific size range while being insensitive to the low-frequency background gradients common in microscopy. The algorithm is parameterized to search for particles across a range of scales, defined by:

**1) LOG\_SIGMA\_MIN:** The standard deviation of the smallest Gaussian kernel, corresponding to the smallest detectable particle size.

**2) LOG\_SIGMA\_MAX:** The standard deviation of the largest Gaussian kernel, corresponding to the largest detectable particle size.

**3) LOG\_NUM\_SIGMAS:** The number of intermediate scales to search between the minimum and maximum, controlling the granularity of the size search.

- 6 **2.4 Dynamic and Static Thresholding:** The sensitivity of the LoG blob detection is controlled by a sophisticated two-part thresholding mechanism that adapts to imaging conditions:

Static Threshold: The **LOG\_THRESH** parameter sets a base sensitivity level for the LoG algorithm's response. Lowering this value makes the detection more sensitive to faint particles.

Dynamic Threshold: The **ROBUST\_K** parameter enables an advanced adaptive threshold. This value is used as a multiplier for the frame's Median Absolute Deviation (MAD), a robust measure of statistical dispersion that is less sensitive to outliers than standard deviation. This dynamic adjustment makes the detection more robust to frame-to-frame variations in overall brightness or background level, helping maintain more consistent particle detection even in videos with photobleaching or focus drift.

- 7 **2.5 Post-Detection Filtering**

After initial blob detection, a final set of filters is applied to refine the results and improve accuracy:

**1) Area-Based Filtering:** The **MIN\_PARTICLE\_AREA** and **MAX\_PARTICLE\_AREA** parameters are used to reject detected objects that are physically too small or too large to be valid particles, effectively removing salt-and-pepper noise and large, non-specific fluorescent aggregates.

**2) Signal-to-Noise Ratio (SNR) Filtering:** An optional filter controlled by the **SNR\_MIN** parameter can be applied to remove detections that do not have a sufficiently strong signal relative to their local background. Setting this parameter to zero disables the filter.



## Interactive Workflow and Parameterization

8 The effectiveness of the RPQ system relies on its interactive workflow, which empowers the user to precisely tune the detection pipeline for their specific imaging conditions and particle characteristics. Rather than requiring users to guess parameters, RPQ provides immediate visual feedback, ensuring that the chosen settings are appropriate before launching a time-consuming analysis of the full video.

### 9 **3.1 User workflow**

The primary user workflow follows four distinct steps:

**1) Load Video:** The user begins by loading an MP4 video file containing the time-series microscopy data.

**2) Configure Parameters:** Using the interactive *ipywidgets* interface, the user adjusts the key detection parameters that govern the image processing pipeline.

**3) Apply and Preview:** The user applies the current settings to a single, representative frame of the video. The system provides immediate visual feedback by displaying the frame with colored circles overlaid on each detected particle. This step is crucial for iteratively tuning the parameters until detection is accurate.

**4) Run Analysis:** Once the user is satisfied with the preview, they initiate the full analysis. This step processes the entire video using the validated parameters, generating the final output files.

### 10 **3.2 Parameter tuning guidance**

CytoSoftware-RPQ provides several adjustable parameters that control particle detection, tracking, and quantitative signal extraction. While default settings are optimized for typical live-cell phagolysosome imaging conditions, variations in signal intensity, particle size, and background noise may require parameter adjustment to achieve optimal performance.

This section offers practical guidance on how to tune key parameters, explains their technical purpose. Users are encouraged to iteratively refine parameters to balance sensitivity and specificity, ensuring robust and reproducible single-particle measurements across experiments. The table below details the key detection parameters available to the user.

11 **Recommendation:** Based on the interactive preview, users can systematically optimize detection accuracy:

**1) To Reduce False Positives:** If the system is detecting noise or artifacts, the recommended adjustments are to increase **RED\_DOM\_RATIO\*\***, **RED\_MIN\_ABS\*\***, **LOG\_THRESH\*\***, or **MIN\_PARTICLE\_AREA\*\***.

**2) To Increase Detection of Missed Particles:** If the system is failing to detect valid particles, the recommended adjustments are to decrease **LOG\_THRESH\*\***, **ROBUST\_K\*\***, or **RED\_MIN\_ABS\*\***. For very small particles, reducing **LOG\_SIGMA\_MIN** may also be necessary.

- 11.1 This end-to-end pipeline transforms each raw video frame into a final, qualified set of particle detection, a process the user directs through an interactive workflow

## System Outputs

- 12 A critical function of any scientific analysis tool is the generation of clear, machine-readable, and comprehensive outputs. The RPQ system produces a bundle of three distinct files for each analysis run, designed to provide both high-level quantitative summaries and deep, per-particle feature data for subsequent analysis and visual verification.
- The three core output files are:
- 1) red\_particles\_by\_time.csv:** This file provides a per-frame summary of the analysis. It is the primary output for kinetic studies and contains columns for the frame index (frame\_index\_original, **frame\_index\_used**), **the corresponding time in seconds (time\_sec)**, and **the total count of detected red particles in that frame (n\_red\_particles)**.
  - 2) red\_blob\_features.csv:** This file offers a detailed, per-blob dataset containing rich feature information for every single particle detected across the entire video. For each particle, it records its coordinates (x, y), **size (sigma/radius, area)**, **intensity (mean RGB, peak/background stats)**, **color dominance ratios**, and **signal-to-noise ratio (SNR)**. **Frame and time identifiers are included to link each particle to its origin.**
  - 3) annotated\_red\_particles.mp4:** This is the primary visual output for verification. It is an MP4 video identical to the input but with detected particles marked by circles in each frame. This allows researchers to visually confirm the accuracy of the detection settings across the entire time series. Note that the output video's frame rate is adjusted to reflect any frame skipping settings used during the analysis.

## Interpreting the Results

- 13 The generated data can be used to answer a variety of scientific questions:
- 1) Analyzing Kinetics:** The **red\_particles\_by\_time.csv** file is ideal for plotting particle count over time. This allows for the direct study of accumulation rates, quantification of kinetic slopes, and measurement of key metrics like time-to-plateau, which are useful for comparing experimental conditions.
  - 2) Post-Hoc Data Filtering:** The extensive feature set in **red\_blob\_features.csv** empowers researchers to perform more advanced, post-hoc analysis. After a run is complete, results can be further refined by filtering particles based on their size, intensity, or SNR, allowing for the isolation of specific subpopulations of interest without re-running the entire video analysis.
- It is critical to note a key clarification on the system's scope: ***This build counts particles directly (not particle-positive cells). Counting positive cells requires a separate cell segmentation workflow.***

## Conclusion and Best Practices

- 14 The RPQ system represents a powerful, interactive, and transparent workflow for the challenging task of quantifying red fluorescent particles in microscopy videos. Its combination of a robust, LoG-based detection algorithm with an intuitive, interactive tuning process makes it an effective and accessible tool for biological research. By separating the complexities of image processing from the user-facing parameterization, RPQ enables scientists to generate accurate, reproducible quantitative data from time-series experiments.
- 15 ***Recommendations for Reproducibility:*** To ensure the highest standards of scientific rigor and reproducibility, researchers using the RPQ system should adhere to the following best practices
  - 1) Report Acquisition Metadata:** Fully document and report all relevant image acquisition parameters, including the microscope objective, camera exposure settings, channel mapping, frames per second (FPS), and pixel size, if known.
  - 2) Document Analysis Parameters:** Record and report all RPQ detection parameters used for the final analysis. This includes any frame skipping settings (***\*\*PROCESS\_EVERY\_N\_FRAMES\*\****) that may have been applied to speed up processing.
  - 3) Archive the analysis configuration file alongside the output files (CSVs, annotated MP4) to ensure full traceability of the results and parameters.