


SOFTWARE

Open Access



# LLAMA: a robust and scalable machine learning pipeline for analysis of large scale 4D microscopy data: analysis of cell ruffles and filopodia

James G. Lefevre<sup>1</sup>, Yvette W. H. Koh<sup>1</sup>, Adam A. Wall<sup>1</sup>, Nicholas D. Condon<sup>1</sup>, Jennifer L. Stow<sup>1</sup> and Nicholas A. Hamilton<sup>1,2\*</sup> 

\*Correspondence:  
n.hamilton@imb.uq.edu.au  
<sup>1</sup> Institute for Molecular  
Bioscience, The University  
of Queensland, Brisbane,  
QLD, Australia  
Full list of author information  
is available at the end of the  
article

## Abstract

**Background:** With recent advances in microscopy, recordings of cell behaviour can result in terabyte-size datasets. The lattice light sheet microscope (LLSM) images cells at high speed and high 3D resolution, accumulating data at 100 frames/second over hours, presenting a major challenge for interrogating these datasets. The surfaces of vertebrate cells can rapidly deform to create projections that interact with the micro-environment. Such surface projections include spike-like filopodia and wave-like ruffles on the surface of macrophages as they engage in immune surveillance. LLSM imaging has provided new insights into the complex surface behaviours of immune cells, including revealing new types of ruffles. However, full use of these data requires systematic and quantitative analysis of thousands of projections over hundreds of time steps, and an effective system for analysis of individual structures at this scale requires efficient and robust methods with minimal user intervention.

**Results:** We present LLAMA, a platform to enable systematic analysis of terabyte-scale 4D microscopy datasets. We use a machine learning method for semantic segmentation, followed by a robust and configurable object separation and tracking algorithm, generating detailed object level statistics. Our system is designed to run on high-performance computing to achieve high throughput, with outputs suitable for visualisation and statistical analysis. Advanced visualisation is a key element of LLAMA: we provide a specialised tool which supports interactive quality control, optimisation, and output visualisation processes to complement the processing pipeline. LLAMA is demonstrated in an analysis of macrophage surface projections, in which it is used to i) discriminate ruffles induced by lipopolysaccharide (LPS) and macrophage colony stimulating factor (CSF-1) and ii) determine the autonomy of ruffle morphologies.

**Conclusions:** LLAMA provides an effective open source tool for running a cell microscopy analysis pipeline based on semantic segmentation, object analysis and tracking. Detailed numerical and visual outputs enable effective statistical analysis, identifying distinct patterns of increased activity under the two interventions considered in our example analysis. Our system provides the capacity to screen large datasets for specific



structural configurations. LLAMA identified distinct features of LPS and CSF-1 induced ruffles and it identified a continuity of behaviour between tent pole ruffling, wave-like ruffling and filopodia deployment.

**Keywords:** Machine learning, Semantic segmentation, High performance computing, Object detection and tracking, Macrophage, Ruffles, Filopodia

## Background

Recent developments in microscopy include the introduction of the lattice light sheet microscope (LLSM) [8] which utilises a 2D optical lattice of Bessel beams to achieve resolution in X, Y and Z plane near diffraction-limited level with high signal-to-noise ratio. Beside maintaining good optical resolution, another key advantage of the LLSM is the low phototoxicity and photobleaching of specimens, which enables cells to be surveyed for extended periods of time in physiologically relevant conditions. This imaging now enables live, 3D fluorescence imaging of sufficient speed, duration and temporal-spatial resolution to adequately capture and record exquisite details of dynamic cell surface projections.

In vertebrate cells the cell surface can be dynamically deformed to produce a variety of membrane projections that are used for interactions with other cells and the micro-environment. As innate immune cells, macrophages are well known for their ability to extend dramatic filopodia, ruffles and phagocytic cups which contribute to their roles in immune surveillance and defence [33]. Continuous, dynamic ruffling is a feature of the macrophage surface, occurring constitutively, it is further enhanced by cell activation [9]. Large membrane ruffles can close, entrapping and internalising fluid-phase cargo in vacuolar macropinosomes which are also hubs for receptor signalling and for endocytic and recycling membrane traffic [34, 35].

LLSM studies have recorded novel features of ruffling, macropinocytic cups, filopodia and other surface projections in amoeba and vertebrate immune cells [9, 11, 22, 36]. LLSM recordings in 3D extending over many hours can capture thousands of cell surface protrusions, routinely resulting in terabyte-scale datasets that cannot be interrogated manually or with traditional segmentation, nor with techniques such as thresholding and automatic spot detection, which require careful calibration or manual editing [13]. Machine learning provides a promising approach, allowing manually defined example data to be extrapolated via sophisticated models. These models can be applied at scale, robustly performing tasks such as classification and semantic segmentation.

We earlier used live imaging and LLSM to record cell surface ruffling on activated macrophages [9]. The macrophage surface has many spike-like projections or filopodia, in addition to constant, undulating wave-like membrane ruffles [35]. LLSM also revealed a new type of ruffle, so-called 'tent pole ruffles' characterised by filopodia (tent poles) embedded in the ruffles. The tent poles appear to raise up the intervening ruffle and then twist together to close the ruffle for the formation of fluid-filled macropinosomes. The distinctive tent pole ruffles were characterised on lipopolysaccharide (LPS) activated macrophages but are also detected on other cell types such as cancer cells [9]. The relationships between filopodia, ruffles and tent pole ruffles remain to be fully investigated to define their modes of formation and deployment by cells under different conditions. These and other cell projections perform key roles in cell migration, immune defence,

and the formation of macropinosomes for environmental sampling and nutrient acquisition. LLSM datasets are valuable troves of data for the quantitative analyses required to gain critical insights into cell surface behaviours, but machine learning is necessary to unlock this information.

Semantic segmentation is designed to distinguish between classes of structures, and it will not necessarily separate individual objects. An effective system for analysis of LLSM data must therefore supplement machine learning segmentation with an object separation and tracking system. Such algorithms are well studied, but care must be taken to ensure the selected approach also works robustly at scale, without the need for manual editing. Macrophage ruffles are complex and highly dynamic structures, representing perhaps the worst-case scenario for object delineation and tracking.

Deep learning models, particularly the U-net (Ronneberger, Fischer, & Brox, 2015), provide a powerful approach to semantic segmentation, but require extensive annotated training data. Sophisticated instance segmentation algorithms have also been developed that allow for the direct detection of individual objects, notably Mask R-CNN [12], which like the U-net is based on a convolutional neural net architecture. However, such instance segmentation models require even more extensive annotation of training data, including segmentations of numerous individual objects. Lacking the required training data, we instead designed a platform that would allow for the rapid development and modification of models with minimal data annotation. This approach is particularly suitable for cases where image properties are likely to change between datasets due to experimental requirements and markers, and different structures may need to be identified.

While non-interactive and distributed computation is a natural choice for computational analysis of terabyte scale data sets, an adaptable platform as outlined above will require integrated visualisation for interactive tasks such as training data selection, parameter selection and quality control for the algorithms applied, as well as display of final outputs such as segmentations or rendered surfaces. Popular bioimaging analysis platforms including ImageJ/Fiji [30–32], Vaa3D [19], and proprietary software such as Bitplane Imaris, are designed to combine analysis and visualisation in an interactive environment. Scripting systems then allow a process developed interactively on in-memory imaging to be applied to unseen data. Although these approaches are clearly effective, integration into a pipeline involving distributed computational resources adds an additional layer of complexity and is limited in most cases. Our platform includes a specialised visualisation tool to address this need.

It should also be noted that for a specific image analysis task, there may be specialised software available; for example, the quantification and tracking of filipodia [7]. To the best of the authors' knowledge there is no method which claims to directly detect and quantify more complex membrane ruffling structures in the same fashion. There is, however, an alternative approach to the analysis of cell surface structures, based on the division of a defined surface into convex objects. These structures can then be analysed and classified by shape-based features [10]. While potentially very powerful, this approach is predicated on high quality rendering of the cell surface including complex and fine structures. This 2-class segmentation problem is challenging since threshold based methods tend to remove fine structure, but in recent work the Curvature-Enhanced Random Walker [21] provides an effective if computationally intensive method. The algorithm

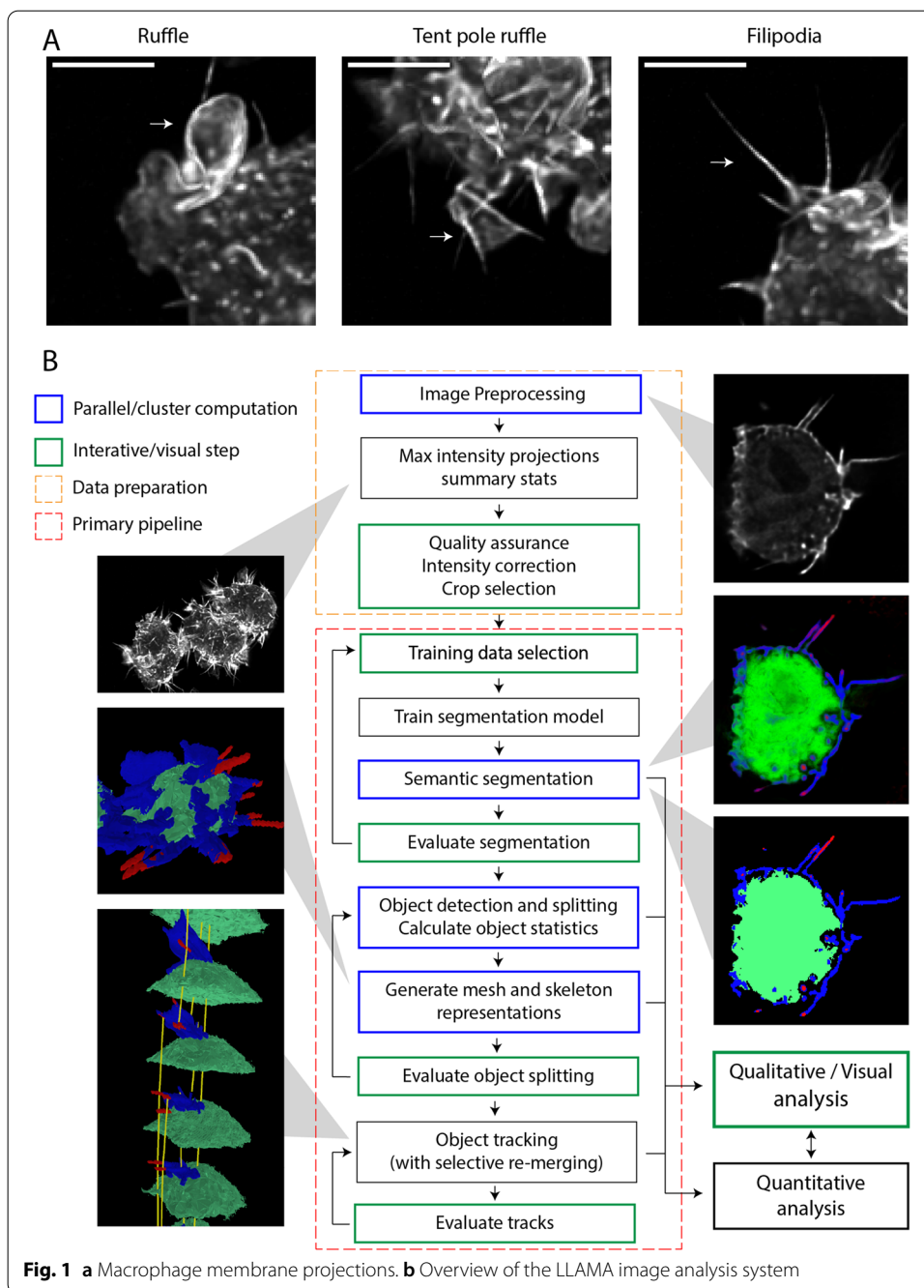
requires tuning of 2 parameters, but the need for annotated data is much lower than for deep learning models. This is clearly an important approach to cell membrane analysis. However, without some modification it does not seem to be able to address one key part of the task addressed here—the segmentation of ‘tent poles’ within ruffles.

### Implementation

We demonstrate a scalable, configurable and modular analysis platform suitable for large 4D microscopy datasets (Fig. 1b), LLAMA (large scale light microscopy analysis with machine learning). A demonstration dataset is provided at <https://doi.org/10.14264/3084db2>, including code, the visualiser app, and scripts configured to demonstrate the computational pipeline on a local Linux machine. Source code is also provided (github.com/jameslefevre/4D-microscopy-pipeline, github.com/jameslefevre/visualiser-4D-microscopy-analysis), as well as a detailed set of protocols (see Additional file 1: Supplementary material).

Our software was developed specifically for the analysis of actin-rich protrusions consisting of spike-like filopodia, ruffles and tent pole ruffles on the surface of Lifeact-labelled, activated macrophages in terabyte-scale LLSM datasets (Fig. 1a). Filopodia and ruffles are distinguished from cell bodies, and individual structures are identified and tracked over time. The pipeline is based on a machine learning approach to perform semantic segmentation, assigning each voxel to a defined class, followed by object separation and tracking algorithms designed to deal effectively with ambiguous structure delineation. The output datasets contain rich information on cell surface features over time, suitable for both statistical analysis and visualisation. This pipeline also provides a template for the application of ImageJ [32] based algorithms to large scale image datasets, including deployment to HPC (high performance computing) systems. We also provide a specialised 4D visualisation tool, the LLAMA visualiser, designed to support the pipeline (training data selection, parameter selection and validation) as well as to visualise the outputs. Effective visual validation and testing is a critical element in the LLAMA platform; the alternative is to provide extensive annotations that would allow quantitative scoring of semantic segmentation, and of the identification and tracking of structures. When confident in the segmentation model and the object detection and tracking parameters, the pipeline can be fully automated.

The platform is designed to be adaptable to other types of large 4D image datasets where the general analysis approach (identification and tracking of different classes of structures over time, with output of quantitative data) is suitable for addressing the research question. While we present results for one model and one biological and imaging system, it should be noted that the same segmentation, object analysis and tracking algorithms are employed on the dissimilar cell body, ruffle and filopodia classes, with all customisation via per-class training data and parameter selection. This use of generalised algorithms is designed to allow ready application of the platform to novel imaging and tasks, together with an approach that emphasises rapid development of models on new and un-annotated image data. The incorporation of 3D visualisation presumes that this is a useful tool for model validation and tuning, and assessment of final outputs. While this will be more difficult for structures that



are internal to cells or tissues, the visualiser includes the capacity to hide individual segmentation classes, and to switch seamlessly between 3 and 2D views.

Ruffle, tent pole ruffle and filopodia. Example images from LPS treated cell samples, maximum intensity z-projection. Scale bars are 5 μm.

Key steps in analysis pipeline; images are illustrative. Our system is designed for systematic analysis of large scale 4D microscopy datasets and provides high throughput by performing intensive computations (blue) in parallel for each time step, suitable for an HPC cluster. The LLAMA visualiser is designed to support all interactive

steps (green), including quality assurance, segmentation model development and parameter selection as well as analysis of results. Development of the segmentation model and the selection of parameters for object detection and tracking should be performed using a representative selection from the dataset. The pipeline can then be run non-interactively, using uniform settings to ensure comparable output across the dataset. See Methods and Materials and supplementary protocols for details.

Our computational pipeline is provided as a set of linked protocols (see Additional file 1: Supplementary material), with complete and commented code ([github.com/jameslefevre/4D-microscopy-pipeline](https://github.com/jameslefevre/4D-microscopy-pipeline)). The implementation primarily uses headless ImageJ [32] processes, invoked via parameterised Groovy scripts, which combine core ImageJ functionality with selected plugins and custom extension code. This provides a common interface to each computational step with flexible calling options, including interactively from the ImageJ interface, as a background process on a local machine, or on a remote server or cloud service. While this approach exposes some complexity, requiring manual editing of scripts and selecting paths and other parameters, it also provides extensive flexibility in deployment, including local or remote execution as well as modification, removal or substitution of steps. The protocols provide a detailed guide for deploying the pipeline using a remote cluster with a PBS batch job system to perform the main computations (semantic segmentation and object analysis, see Fig. 1b) in parallel for each time step, enabling a scalable, high-throughput system. The code provided includes example PBS job scripts. ImageJ/FIJI was selected since it is free and open source, provides access to a wide range of image processing algorithms (via plugins as well as core ImageJ), and also has the benefit of a fully cross-platform Java-based system and simple installation (admin privileges not required, optionally bundles Java to avoid system dependency).

The custom 3/4D LLAMA visualisation software we developed ([github.com/jameslefevre/visualiser-4D-microscopy-analysis](https://github.com/jameslefevre/visualiser-4D-microscopy-analysis)), is built on the Processing 3 environment and language [28], which provides a powerful framework for interactive visualisation; this is the only component of the system which is not based on ImageJ. It should be noted that the visualiser is not directly part of the analysis pipeline, and following our modular approach it may be ignored, or alternative tools used. For example, the Vaa3D system, while particularly focussed on neuronal analysis, provides powerful 3D image volume and surface display capacities. However, the custom tool provides tight coupling with the analysis pipeline and targeted features; we found this to be a vital support tool throughout the process, enabling easy and direct comparison of imaging with segmentations and object representations, and between alternative versions of these outputs. This tool is designed specifically to support training data selection in 3D and customisation and quality control across the pipeline, as well as visualising outputs. It features rapid switching between slice and 3D view, and between the original image and one or more, from a single selected perspective. The segmentations may similarly be compared to the object representations derived from them. Additional files 2, 3: Supplementary Videos 1 and 2 demonstrate the key features of the visualiser and its use as part of the system. Additionally, the supplementary protocols document includes a guide, and a manual is included with the code.



### Semantic segmentation method

Our semantic segmentation approach is based on the Trainable Weka 3D software [2], which is implemented as a plugin to the ImageJ image processing platform [32]. This machine learning tool produces segmentations using a two-step process. Using ImageJ, a range of sophisticated pre-defined 3D image features are computed at multiple scales, capturing rich spatial context for each pixel. Then the set of features for each pixel is used with a selected machine learning algorithm from the Weka toolkit [39] to produce a per-pixel classification. We employed a reengineered high-throughput pipeline suitable for use on a large scale 4D dataset with cluster-based computation ([github.com/jameslefevre/4D-microscopy-pipeline](https://github.com/jameslefevre/4D-microscopy-pipeline)).

We briefly describe the Trainable Weka system and our reengineered pipeline, designed to more effectively deal with large scale data. The Trainable Weka machine learning segmentation algorithm has the following steps:

#### Model training

- Define segmentation classes and select training data for each class from a training image.
- Compute selected image features for the training image.
- Extract image features and class for each training set voxel, generating a data table.
- Train classification model on this training data table using a selected Weka algorithm.

#### Model use

- For each image stack, compute the image features that are required for the trained segmentation model.
- Extract the image features for each voxel and apply the trained model to classify the voxel.
- Combine voxel classifications to produce a segmentation and (optionally) a probability map giving the estimated probability distribution over the classes for each voxel.

The role of the image features, using algorithms provided by the ImageJ platform and the ImageScience plugin [23], resembles that of the earlier convolutional layers in the deep learning models such as U-Net [29] sometimes used for semantic segmentation [41]. However, using predefined image features radically reduces the cost of training in computational time and in the requirement for manually segmented training data, although the modeller must ensure that the selected features and scales capture sufficient spatial context for pixel classification. Full manual segmentation to produce training data may represent weeks of effort for a biologist, and represents a major limitation in the use of deep learning; since the generalisability of the model to new data is uncertain, potentially limiting the useful life of the model, the required effort is often impractical. The Trainable Weka approach uses a much smaller set of manually selected training data, although high quality results typically require visual assessment on a representative set of full images, and model iteration.

The Trainable Weka plugin uses the ImageJ interactive environment for both training and application of segmentation models in both 2D and 3D, which is highly effective for smaller image sets. Trained models may also be saved from the interactive environment, and either reloaded or deployed via non-interactive scripted processes. We encountered several bottlenecks when attempting to apply this software at scale: image features in 3D are expensive in computational time leading to long delays during the interactive process; available RAM limits the number of image features that can be used; software instability was encountered under heavy load, compounding the previous issues; training data selection was often difficult due to a lack of 3D context; and extending or revising existing training data selections, or combining training data for multiple stacks, is only possible using an ad-hoc process outside the interactive environment. We created our LLAMA visualiser, in the first instance, to provide clear 3D context during training data selections: see Additional file 2: Supplementary Video 1 for a demonstration of the visualiser in training data selection and the evaluation and revision of segmentation models. The other issues identified motivated our reengineered pipeline, in which the Trainable Weka plugin is used for the interactive selection of training data, but all computation is done non-interactively. Other key modifications are listed below:

- Image features are generated in a stand-alone process and cached to disk to minimise computational cost.
- Selected image features may be approximated using a down-sampled image. Features are calculated on a range of selected scales (the parameter *sigma*), and the computational cost increases with scale. But larger scales may be required to capture the required spatial context. We can effectively approximate larger scale features by down-sampling the original image, calculating the feature with appropriately reduced sigma, then up-sampling with interpolation. This greatly reduces computational cost. Importantly, the processes and code provided ensures that the features are calculated in a consistent way during training and deployment.
- Training data selections are recorded in an ImageJ macro. This process allows for editing and documentation of the selections and an easy way to resume or extend selections. It also enables the extraction of training data features to be handled by a separate non-interactive process with access to the macro file.
- Full flexibility is allowed in feature selection. In the plugin, each selected feature is used at each selected scale; the modified process allows any combination of feature and scale if desired.

The protocols document provides detailed instructions for the segmentation model training process and for the deployment of trained models. It is often necessary to modify the model one or more times after evaluation of segmentation results on a larger set of image data, and an additional protocol is provided as a guide for this iterative process.

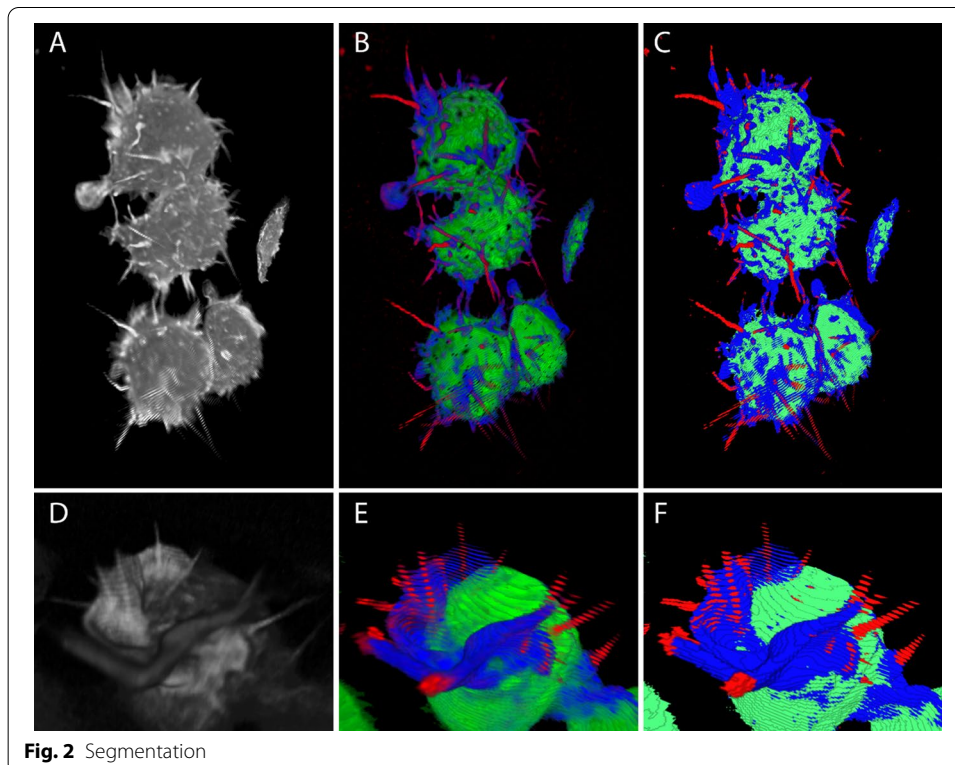
#### **Object detection and quantification method**

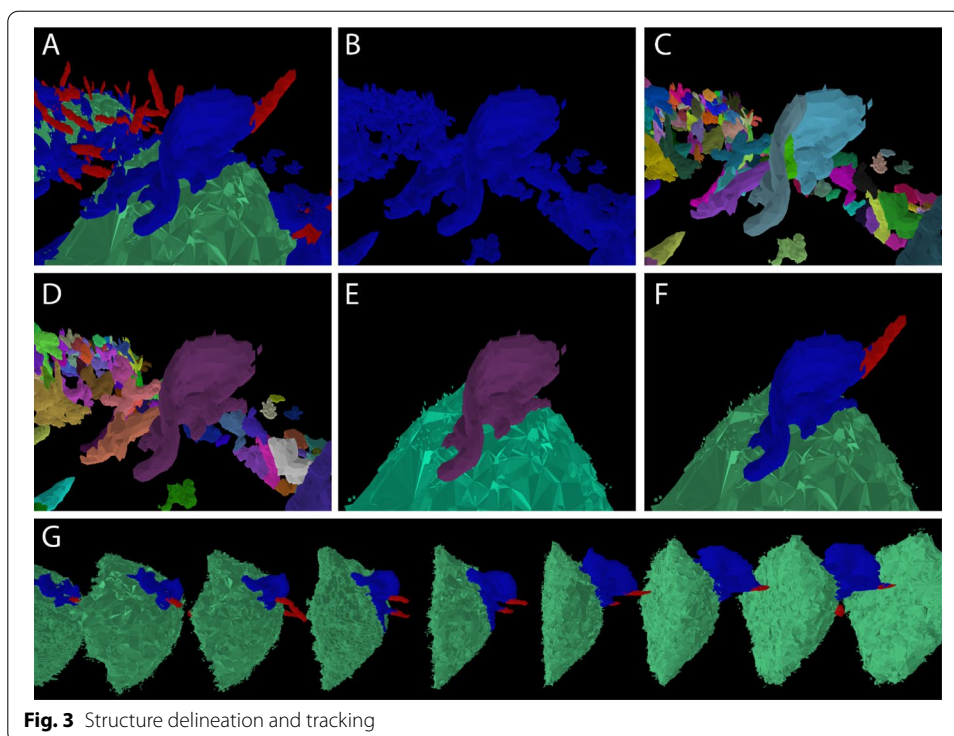
Semantic segmentation is designed to classify each voxel as belonging to a type of structure, such as filopodia or ruffle, and will not necessarily separate individual adjacent objects. However, the most useful quantitative analysis of cell imaging will typically be



based on individual cells or subcellular structures, tracked over time. For example, in Fig. 2c (see “Results”) we can clearly distinguish 4 cells (and the edge of a structure which is primarily outside the imaging frame), but these are not fully separated in the segmentation. These cells must be delineated, and each tracked over time. Macrophage ruffles are more complex and highly variable, representing perhaps the worst-case scenario for object delineation and tracking. It can occasionally be unclear even to the human analyst whether an object should be considered as one structure or two, or where the boundary is, or when an object should first be considered a ruffle that is distinct from the cell membrane. Biological variation and noise in the imaging process mean that an automated process attempting to decide these questions without putting the data in temporal context may give inconsistent results over a sequence of time steps, making coherent tracking of structures impossible. However, due to the scale of the data and the need for computational feasibility it is necessary to *segment* each image stack in isolation, without incorporating information from adjacent time steps, and the algorithm used for this task cannot rely on manual editing.

In response to these challenges, we developed a sophisticated and configurable approach in which touching structures are split using a watershed algorithm [5] on edge distance, then selectively recombined as part of the tracking process in order to pool information across time steps and produce coherent tracks of structures over time (see Fig. 3 in “Results”). The watershed parameters are chosen to ensure that all necessary object splits are made, at the cost of potential over-separation which is corrected in the tracking algorithm. This is illustrated in panel C and D of Fig. 3. Most computations, including the expensive watershed split step, are performed in parallel for each image





stack. This allows a scalable high-throughput analysis pipeline. The tracking algorithm is then applied using light-weight summary object information aggregated across time steps. Methods involving a re-merging step in combination with watershed split are well known, for example [37]. The key innovation in our algorithm is to integrate this re-merging with the tracking algorithm as a way of efficiently pooling information across time. The algorithm is described in detail below.

A potential challenge with the macrophage segmentation is a degree of unresolvable ambiguity between the ruffle and filopodia classes; some structures appear to be truly intermediate, giving complex and unstable segmentations. In order to allow analysis of these structures, such as filopodia in the process of forming or decaying, we also applied the object detection and tracking algorithm to the merged ruffle and filopodia classes, in addition to the separate analyses. The average class probabilities of each structure are provided as an additional feature (measuring the degree to which a composite structure most resembles a ruffle or filopodia). This capacity to analyse merged segmentation classes is included as a general feature of the platform, documented in the protocols provided.

The watershed algorithm selected was extended maxima watershed on the edge distance [27]. Edge distance is the distance from each voxel in an object to the nearest external voxel; the idea is to use this measure to identify and remove narrow connections between more compact regions that are likely to represent separate objects. Classical watershed uses local maxima (equivalently minima) as seeds of expanding regions, but irregular object shapes may result in multiple local maxima in an object, leading to excessive splitting. The extended maxima approach solves this problem by specifying a minimum difference in the underlying measure (edge distance in this case) between

adjacent regions. This minimum difference is the parameter “dynamic”. Given any two local maxima  $m_1$  and  $m_2$ , they will be merged into the same region if there is a path between them where the minimum value is greater than  $\min(m_1, m_2) - \text{dynamic}$ . The dynamic parameter is used to tune the algorithm, with smaller values leading to more splitting.

The watershed algorithm implementation (ExtendedMinimaWatershed) was provided by the MorphoLibJ plugin [17], working with 3D distance maps (edge distance transform) provided by the 3D ImageJ Suite (mcib3d-suite) plugin [25]; 3D distance maps were also used to provide auxiliary statistical information on proximity between structures of different classes. The 3D ImageJ Suite plugin was additionally used for 3D hole filling, and to produce object meshes using the marching cubes algorithm [20] followed by mesh pruning. Skeleton representations of linear structures are produced using the plugins Skeletonize3D and AnalyzeSkeleton [3]. Skeletons and meshes provide an efficient means of object visualisation and may also be used in statistical analysis.

A range of descriptive features are calculated for each object, including three that are used in the tracking algorithm: position, volume, and the amount of contact with each adjacent object. The measure of contact between objects A and B is the number of distinct pairs of adjacent voxels (a, b), where a and b are contained in A and B respectively. Two voxels are considered adjacent if they are equal in one coordinate and differ by at most 1 in other coordinates, so a non-edge voxel is adjacent to 18 neighbours. This summary data is saved in tabular form for each stack and forms the inputs for the tracking step described below. These features are also carried through to the computed tracks as part of the output, and combined appropriately when objects are merged.

### Tracking algorithm

In this section we provide a technical description of the tracking algorithm. Detailed instructions for applying the algorithm are included in the supplementary protocols, but the following text should also be consulted as a guide to selecting the key parameters used and understanding the output. We conclude with a brief guide to using the LLAMA visualiser to assess tracks and revise parameters.

The algorithm presented is designed to deal with data where adjacent structures often touch, the identity of individual structures is ambiguous, and methods used to delineate them can give inconsistent results between consecutive time steps. For example, it may be unclear at what point a dividing structure becomes two structures, or a disappearing structure completely recedes into the cell membrane. Biological variation between time steps or noise in the observational process means that a fixed algorithm applied to each time step may produce inconsistent segmentation results, leading to low quality tracking of structures over time, and the goal of the following is to correct these.

The approach is to pool information across time to give temporally stable and coherent representations of structures. We start with object data for individual time steps where an algorithm such as water-shedding has been used to separate touching structures, and the resulting objects have been reduced to a summary form; this has the advantage of allowing most computation to be done at the level of individual time steps, enabling parallel processing with relatively modest resource requirements. The splitting algorithm should be run under aggressive parameters to ensure that all true structures

are split apart, at the cost of initially incorrectly subdividing many structures (if it is possible to completely avoid both under and over splitting, this tracking approach is not required and a simple matching algorithm is sufficient). We then selectively *merge* adjacent objects and *match* between successive time steps to form coherent structure tracks, seeking a balance between merging and matching in a sensible way and maintaining tracks over time.

The algorithm is customised with the 6 parameters specified in Table 1. Tracking is performed separately for each class of structure, so these parameters are specified for each class individually. See header information in the script “get\_tracks.groovy” for details of how to set these and other parameters.

The general approach is to assign scores for merging adjacent objects at the same time step, based on the contact area, sizes and centre of mass distance between the objects, and combine with a distance measure for matching objects between consecutive time steps (with a threshold above which no match is made). We start by running a matching algorithm between consecutive time steps, forming a draft set of tracks (modified Hungarian algorithm [16] with distance threshold). Then we run an iterative process of track merging.

#### Quantification of object matches between adjacent time steps and merges at each time step

The distance measure used to match objects across time takes relative size into account as well as distance, with the relative importance controlled by the parameter  $W_l$ . Given objects  $a$  and  $b$  with positions  $\mathbf{p}_a$ ,  $\mathbf{p}_b$ , and volumes  $v_a$ ,  $v_b$ , then the distance is defined as

$$d(a, b) = \sqrt{|\mathbf{p}_a - \mathbf{p}_b|^2 + \left(W_l \log\left(\frac{v_a}{v_b}\right)\right)^2}$$

We require  $d(a, b) < d_{max}$  for a match to be allowed. In order to help evaluate a set of tracks produced by matching and merging operations, we allocate a matching score

$$S_{match} = d_{max} - d(a, b)$$

This score penalises poorer matches while rewarding longer tracks, since any match within the threshold gives a positive score.

**Table 1** Tracking parameters

Parameter	Name in script	Purpose
$W_l$	logSizeWeight	Weight to put on size difference when matching objects
$d_{max}$	matchThreshold	Maximum distance for matching
$Q_p$	relativeNodeDistance_referenceValue	Calibrates the object merging score based on centre of mass proximity
$Q_c$	relativeNodeContact_referenceValue	Calibrates the object merging score based on contact between objects
$W_c$	relativeNodeContact_weight	Relative weight for contact measure versus distance measure
$W_{match}$	matchScoreWeighting	Relative weight on matching score versus merging score

The object merging score (defined below) is designed to indicate whether two adjacent objects with the same class and time step are truly distinct, or if they should be merged and treated as a single structure. This score is a weighted average of two factors. The relative contact  $R_c$  is an estimate of the contact area as a proportion of the surface area of the smaller object, which is approximated by the surface area of a sphere of the given volume (the lower bound of true surface area); both are in pixel units. The relative proximity  $R_p$  is the inverse distance between the object centres with a correction factor based on the object volumes, since a given distance between centres of mass is indicative of a greater degree of separation for smaller objects. To get this correction factor, for each object we calculate the radius of the sphere with the same volume as the object, then we sum these 2 radii. This sum is then divided by the distance between the centre of masses to give  $R_p$ . Given objects  $a$  and  $b$  with volume and position as above and contact area  $C(a, b)$ , the formulae are

$$R_c(a, b) = C(a, b) / \left( 36\pi \min(v_a, v_b)^2 \right)^{1/3}$$

$$R_p(a, b) = \left( v_a^{1/3} + v_b^{1/3} \right) / \left( (4\pi/3)^{1/3} |\mathbf{p}_a - \mathbf{p}_b| \right)$$

These factors are divided by reference values  $Q_c$  and  $Q_p$ , indicating the neutral level at which the score is not thought to be evidence for or against aggregation. The final merging score is a weighted sum of these factors, with weights based on the estimated usefulness of each.

$$S_{merge}(a, b) = \frac{W_c R_c(a, b)}{Q_c} + \frac{(1 - W_c) R_p(a, b)}{Q_p} - 1$$

Final merge scores above 0 are taken as evidence for aggregation, but negative scores may still be consistent with aggregation when balanced by other score terms. The overall objective function is then a weighted sum of the scores of all merging and matching operations:

$$O = \sum_{(a,b) \in merges} S_{merge}(a, b) + W_{match} \sum_{(a,b) \in matches} S_{match}(a, b)$$

### Tracking and iterative merging process

We seek to create a set of tracks by selectively merging objects at each time step and matching objects between consecutive times, in order to maximise the objective function above. This optimisation problem is computationally infeasible to solve in general, so we first solve the problem for matching only, then iteratively modify the result by merging tracks, allowing for matches to be added or removed in the process (splitting or joining tracks across time).

The objects are assembled into an initial set of tracks by matching objects at each pair of consecutive time steps. Matching distance is typically subject to a reasonably strict threshold ( $d_{max}$ ), such that we expect an object to be matched to at most one object in each temporal direction. But we account for the possibility that an object may have

more than 1 possible match within this threshold, adapting the Hungarian algorithm to ensure that matches are unique (and hence tracks are simple/unbranched), and that total matching distance is minimised within these constraints. To do this, we first calculate the distance matrix between the 2 object sets. Then we replace all values greater than  $d_{max}$  in the matrix with  $d_{max}$ , and pad the smaller dimension of the distance matrix with dummy objects, with all associated distances set to  $d_{max}$ . We then apply the Hungarian algorithm to this square matrix, and finally discard all distance  $d_{max}$  matches and the dummy objects.

The initial set of tracks is then iteratively modified by track merging operations, in which two or sometimes more tracks, existing at the same or overlapping time periods, are merged together over some time interval  $[t_1, t_2]$ . Although there may be more than two tracks involved, exactly two must exist at each time between  $t_1$  and  $t_2$ , and between 0 and 2 tracks may extend beyond the merged interval in each direction. The merge operations may involve one or more cuts to the tracks at the ends of the merged interval; if 2 tracks continue before or after the merged interval, at least one must be cut to avoid track branching. A prospective merge operation is scored by adding the merge scores at each time step and the change in matching score (the scores of all new matches made minus the scores of all discarded matches) weighted by  $W_{match}$ .

We begin the iterative merging process by considering every pair of tracks which are adjacent for at least one time point (touching objects). We find and score the optimal merge between the two tracks, by considering all possible merge intervals in the period where both tracks exist. For each possible merge interval, we calculate whether continuing tracks should be included into the merged track or cut into separate tracks, in order to give the best match score adjustment, and this adjustment is included in the score for the interval. In the case where the merge score is positive, but the overall score is negative after match score adjustment, we consider extending to further tracks (since this situation may be an artefact of a single point tracking failure). If the optimal merge interval continues to the end of the common time period, but one track continues beyond this time, then we look for tracks that are adjacent to the continuing track and start immediately after the merge period. If the matching at the end of the merge interval is improved by this new track, it is added to the hypothetical merge operation, and the merge interval is extended. This extension is continued in both directions while possible, adding new tracks as indicated, unless an overall positive merge score is achieved.

We then identify and execute the potential merge operation with the highest score, provided the score is positive. If possible, we extend the merged track by matching to existing tracks. Merge scores are then recalculated for the merged track, and any tracks formed by splits, and the highest merge score in this adjusted list is found. The process continues until no merges with positive score are available.

### **Assessing tracks and revising parameters using visualiser**

The LLAMA visualiser may be used for assessing tracks, and in combination with the information above it may be used in setting or revising the object splitting and tracking parameters. Additional file 3: Supplementary Video 2 provides a demonstration of the visualiser in this role. Note that multiple track sets based on the same object data can be compared. In the data specification screen, select 2 or more object datasets with the



same object folder but different track files, and provide labels to allow easy identification when the visualiser is running.

Useful tools include the “Selected” filter with specified track id and the “multiple times” option. Using “Object Coloring” you may see the objects as split by the watershed algorithm (“Object” option) versus the possibly merged structures produced by the tracking algorithm (“Node” or “Track” option; these are provided as separate options to support branched tracks if required).

A special feature is activated with the ‘c’ key in the visualiser. When showing tracks, the values of  $R_p(a, b)$  and  $R_c(a, b)$  are shown for each pair of touching nodes. Both values are multiplied by 100 and rounded to the nearest integer, then displayed as a pair  $R_p(a, b) / R_c(a, b)$  between the two nodes. By comparing the displayed values to your judgement of whether objects are best regarded as one structure or two, the appropriate reference values  $Q_p$  and  $Q_c$  can be selected.

## Results and discussion

The software presented is intended to provide an integrated approach to a relatively new practical issue: the quantitative analysis of terabyte scale microscopy data; here we present an analysis that demonstrates the main features of the approach as well as its capacity to produce original biological findings.

We demonstrate our analysis pipeline with a quantitative analysis of two different interventions that stimulate macrophage ruffling: (i) bacterial lipopolysaccharide (LPS) LPS is a potent endotoxin that stimulates morphological changes associated with arming innate immune responses in macrophages through actin reorganisation and tyrosine phosphorylation of Pyk2 and focal adhesion, paxillin [38]; and (ii) macrophage colony stimulating factor (CSF-1), a cytokine involved in differentiation of macrophages that induces cells via WAVE2-Abi mediated pseudopod assembly for cell chemotaxis [15]. Both stimuli also induce macropinocytosis, an actin driven process that facilitates the bulk engulfment of extracellular fluid via ruffling [6, 40]. For these studies, samples containing 17 complete cells were each imaged at a resolution of  $1.04 \mu\text{m} \times 1.04 \mu\text{m} \times 2.68 \mu\text{m} \times 5.3 \text{ s}$  for 53 min total over two captures (before and after treatment). Cell tracks were matched between the two captures and excluded if cells moved partially or fully out of the field of view during imaging. 901,696 objects were found that were associated with the analysed cells, arranged into 76,386 tracks that met minimum size thresholds. Statistics produced included volume, maximum extension from the cell surface and the set of adjacent structures. We identified and analysed 1188 significant ruffling events, defined by peak ruffle volume exceeding  $14.5 \mu\text{m}^3$ .

### Macrophage segmentation

Segmentation classes used to dissect the LLSM macrophage imaging were background, cell body, ruffle and filopodia, where filopodia include the “tent pole” structures contained within some ruffles. Training data was obtained from four image stacks, selected from four different captures and two imaging sessions. Compared to the deep learning models sometimes used for semantic segmentation [41], the Trainable Weka approach requires only a very small selection of training data, although high quality results may require an iterative process involving assessment of representative

segmentations, followed by updating the model with additional training data. Our macrophage model required two revision steps, primarily to add robustness against heterogeneity in noise and contrast between image captures. Final training data was drawn from 4 image stacks selected from 4 samples over 2 separate experiments. It consisted of 2036 simple elliptical image samples: 1101, 220, 244, and 471 respectively for the background, cell body, filopodia and ruffle classes. These produced respectively 9113, 3339, 1057 and 2821 labelled voxels. The total of 16,330 labelled voxels is equivalent to less than 0.1% of a single image stack cropped to the size of a macrophage (approx.  $60000\mu\text{m}^3$ ). Fluorophore intensity was adjusted between and within captures using cytoplasm intensity as a benchmark and modelling an exponential decay curve in each capture; the same intensity adjustment process was also used for analysed data. In addition, the segmentation model was made more robust against base fluorophore intensity variation by fourfold replication of training data with intensity scaled by factors 0.5, 0.75, 1, 1.25. The final model used the Weka random forest algorithm with default parameters on class-balanced training data. The random forest was selected from the range of models available in Weka as it was judged to perform best in generalising beyond the training data. The full range of features was used, with kernel size (sigma of 1,2,4,8, and 16  $\mu\text{m}$  (16  $\mu\text{m}$  is approximately  $15 \times 15 \times 6$  voxels. For computational efficiency, a twofold down-sampling in x and y was used for all features at sigma = 16, and the ImageJ filters (mean, median, minimum, maximum, variance at sigma = 8. The maximum sigma value must reflect the amount of spatial context necessary to classify each voxel, and the down-sampling feature was added to our system to minimise the computational cost of computing features at larger scales. The wide range of features used was necessary to provide robust and clean segmentations in the presence of heterogeneous image quality and noise; otherwise a more parsimonious model could have been used, allowing faster computation.

The robustness of the model was tested on the training data using a tenfold cross validation, giving a 0.73% error rate. Training on similar image datasets consistently gave cross validated errors below one percent, and this continuing very low error rate shows a highly robust model. It should be reiterated that the limited and biased nature of the training data selection means that this metric is a necessary but not sufficient performance measure; testing is performed visually on larger scale data.

The segmentation model was successful in identifying the four classes, distinguishing ruffles and filopodia and identifying filopodial-like “tent poles” embedded in major ruffle structures (Fig. 2).

Sample segmentation output, showing macrophage cells imaged on the LLSM, segmented into background (black/transparent), cell body (green), ruffle (blue), and filopodia/tent pole (red). (ABC) Full image stack, view rotated so that plate is on left side. (DEF) Detail of ruffle containing tent pole like structures. (AD) Original image after pre-processing (de-skew and deconvolution). (BE) Probability map, showing the estimated probability that a voxel will occur in each class. (CF) Semantic segmentation; each voxel is assigned to the single class considered most likely. All images produced using our LLAMA visualisation software ([github.com/jameslefevre/visualiser-4D-microscopy-analysis](https://github.com/jameslefevre/visualiser-4D-microscopy-analysis)).

The segmentation results, displayed alongside the LLSM imaging using LLAMA visualiser, proved effective in helping to manually identify tent pole ruffling events (see Additional file 4, 5: Supplementary Videos 3,4).

### Detection and tracking of cells and surface structures

The object splitting and tracking parameters used are given in Table 2 (see Table 1 above for definitions, except for the watershed parameter *dynamic* which is discussed under object detection and quantification). These were decided by iterative refinement of initial estimates, with results assessed primarily in the visualiser.

In addition, objects that were removed by the watershed algorithm were replaced if above 75 voxels in size, but a minimum size threshold was applied to structures used for tracking: 9000 voxels for cell body, 200 voxels for ruffle, and 30 voxels for filopodia. Image resolution is  $0.104 \mu\text{m} \times 0.104 \mu\text{m} \times 0.268 \mu\text{m}$ , so voxel volume is  $0.0029 \mu\text{m}^3$ .

(A) Semantic segmentation into cell body (green), ruffle (blue), filopodia/tent pole (red), with a prominent ruffle in the foreground. (B) Isolation of ruffle class. (C) Watershed split algorithm separates ruffle from touching objects but incorrectly splits the foreground structure; except in the simplest cases, this initially excessive splitting is necessary to ensure that all required separations are performed. (D) Structure boundaries after re-merging step; this is integrated with the tracking algorithm to maximise consistency with other time steps. (E) Foreground ruffle correctly delineated and associated with the cell body. (F) Ruffle with associated call and filopodia structures (identified using adjacency data) using the original class colour scheme. (G) Structures in F tracked over time; the timestep shown in A-F is in the central position. All images shown were produced using our LLAMA visualiser using only the standard display options available in the graphical user interface, with no post-editing. This application provides a powerful tool for customisation and validation of the algorithm as well as visualising results.

### Statistical analysis

Our system provides a rich set of data for each structure, including size, position, shape, orientation and distance from a specified reference class (measuring, for example, the maximum extension of a structure from the cell surface). The contact between touching objects of all classes is quantified, allowing relationships between structures to be determined. These data are recorded in tabular form, for easy loading into any statistical software. In addition, skeleton and mesh representations of objects are (optionally) produced. These are intended as an aid for visualisation using the LLAMA visualiser or other software, and may also be used to provide additional statistical information such as the length of linear structures.

**Table 2** Watershed and tracking parameters used

Parameter	Dynamic	$W_l$	$d_{max}$	$Q_p$	$Q_c$	$W_c$	$W_{match}$
Cell body	0.2 $\mu\text{m}$	9 $\mu\text{m}$	12 $\mu\text{m}$	0.7	0.06	0.66	3.5 $\mu\text{m}$
Ruffle	0.01 $\mu\text{m}$	2.2 $\mu\text{m}$	2 $\mu\text{m}$	0.8	0.04	0.66	2.5 $\mu\text{m}$
Filopodia/tent pole	0.01 $\mu\text{m}$	2.2 $\mu\text{m}$	2 $\mu\text{m}$	0.5	0.02	0.66	2.5 $\mu\text{m}$

Each macrophage sample analysed was imaged for 300 timesteps before and after treatment, at intervals of approximately 5.3 s; we refer to these as the pre and post captures. Matching of cell tracks allowed direct pre vs post comparisons for each cell after stimulation. Observed changes were compared to untreated control cells, which were prepared in parallel and imaged in the same session. Analysis was performed in the R statistical programming environment using the track files produced at the final stage of the image analysis.

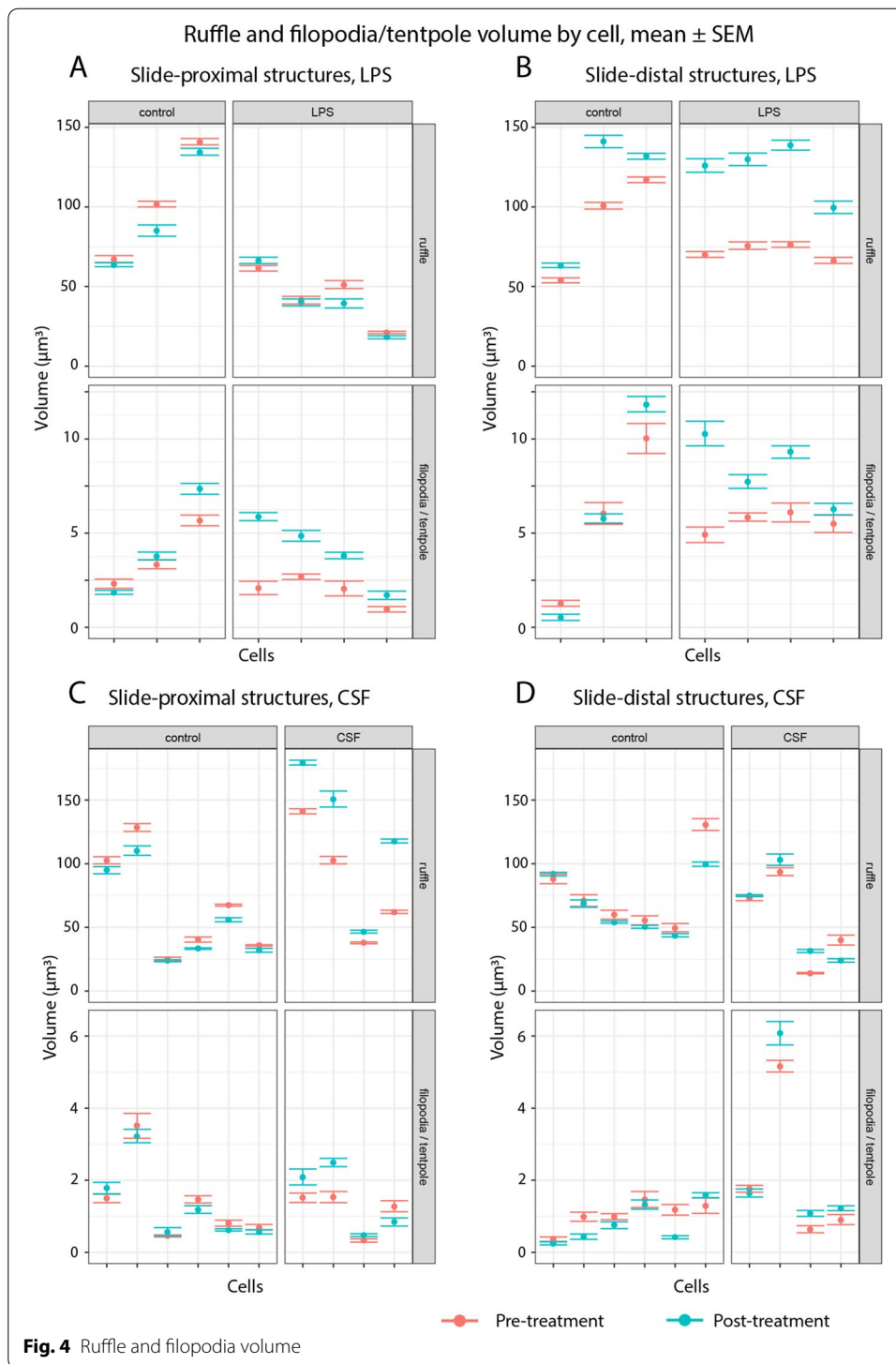
Many ruffles and filopodia were found to extend along the slide surface and appeared to be attached to the slide, resulting in a morphology distinct subset from the structures in the upper region of the cell. Differential behaviour was also considered likely between these regions, so for analysis we classified structures as slide-proximal or slide-distal, depending on whether the distance from the estimated slide position to the structure centre of mass is less or greater than 1.5  $\mu\text{m}$ .

In the following sections we demonstrate types of analysis and conclusions that can be drawn using these methodologies. Distinct patterns of macrophage ruffling from LPS and CSF stimulation (as discussed in the introduction) were seen, with increased activity in the distal and proximal regions of the cell surface respectively. Using data at the level of individual tracked structures, we could attribute this to increased frequency of ruffling events with CSF, while LPS stimulation lead to larger as well as more numerous ruffles. The duration of individual ruffling events did not change significantly in either case.

#### **Distinct patterns of increased ruffling and filopodia volume in LPS and CSF treated cells**

We performed an analysis based on the total volume of ruffle and filopodia structures in each cell before and after cell treatments (Fig. 4), identifying spatially distinct patterns of increased activity. A consistent increase in ruffle volume was seen for both LPS and CSF cells, however this increase occurred exclusively in the plate-distal regions of the LPS treated cells, and the plate-proximal regions of the CSF treated cells. In contrast, a consistent increase in filopodia/tent pole volume was seen in both proximal and distal regions of the LPS cells, while no significant changes were seen in the CSF treated cells.

Aggregate ruffle and filopodia (including tent pole) volume per cell, mean and SEM pre and post treatment. Structures are classified as plate-proximal (within 1.5  $\mu\text{m}$  of imputed plate position) or plate-distal (greater than 1.5  $\mu\text{m}$  from plate). (A) LPS experiment, plate-proximal; (B) LPS experiment, plate-distal; (C) CSF experiment, plate-proximal; (D) CSF experiment, plate-distal. Means are calculated across the time steps in each capture, with SEM adjusted for autocorrelation. This adjustment does not account for possible trends over time that are independent of treatment, so changes in control cells (left) are included for comparison with treated cells (right). In the LPS treated cells, consistent increases are seen in the slide-distal ruffles and the filopodia in both distal and proximal regions. In contrast, the CSF experiment exhibits consistent increase in the slide-proximal ruffles and no change to filopodia volume in either region. Plots were produced with the ggplot2 package in the R statistical programming language, using the tabular output from the object tracking code.

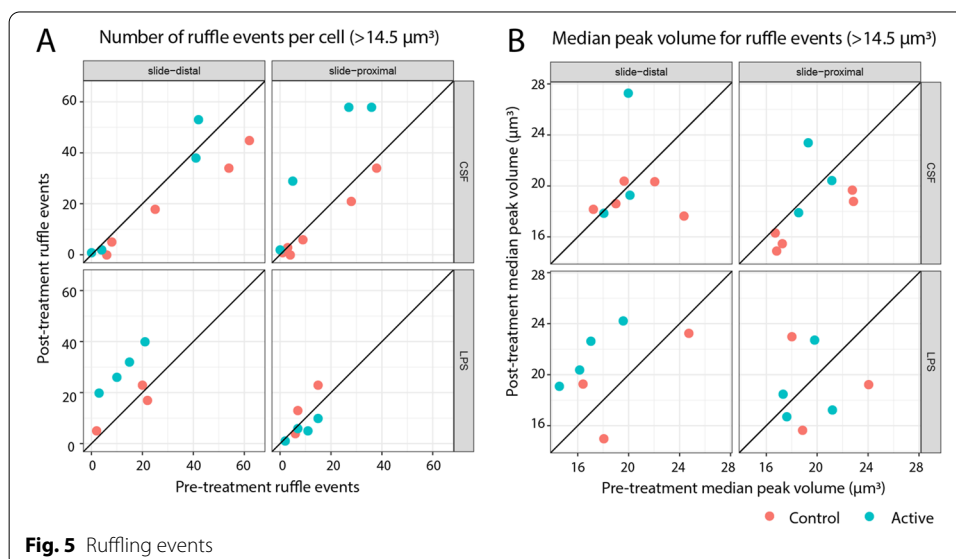


**Increased ruffling associated with higher frequency of events in CSF cells, increased frequency and size in LPS cells**

The aggregate volume analysis in Fig. 4 demonstrates spatially biased patterns of increased ruffling. An important question is whether this is reflected in any change

to the nature of the ruffling events that are associated with macropinocytosis or cell motility under different conditions of cell stimulation. As from our data and others [4, 6, 18, 24, 26], CSF regulates mesenchymal cell motility and macropinocytosis, whereas LPS facilitates primarily macropinocytosis in macrophages under these conditions. Therefore, is the increased aggregate volume associated with an increased number, size or duration of events during these different processes? The track dataset provides a rich set of information for addressing questions such as these. Individual ruffle tracks could not be reliably used as a proxy for ruffling events, as numerous tracks were identified which persisted well beyond a distinct ruffling event and even included two or more consecutive events; this is consistent with actin recycling between ruffling events, and with the spatial clustering behaviour observed in [9]. We isolated major ruffling events from the tracked ruffle structures by identifying track intervals in which the peak volume is above  $14.5 \mu\text{m}^3$  (5000 voxels; this threshold was established by inspection in the visualiser) and the volume at the end points is less than half the peak value. Hence, we are able to classify each LifeAct-labelled event from the initiation of actin polymerisation/extension (increased in ruffling volume) until its retraction.

More ruffling events were observed after treatment in both LPS and CSF cells, with the increases occurring in the slide-distal region of the LPS cells and the slide-proximal region of the CSF cells (Fig. 5a), consistent with the aggregate volume results. There was also an increase in the median peak volume of the slide-distal ruffling events in the LPS cells (Fig. 5b). We observed that LPS stimulate circular dorsal ruffles that protrude exclusively from the plate-distal region/or the peripheral surface of the cell. Whereas in CSF stimulated cells, lamellipodia-like structures along the plate proximal region predominantly emerge upon stimulation. Filopodia/tent-poles, as finger-like linear actin structures responsible for sensing chemical and mechanical cues, elongate and increase in quantity under both circumstances.





The number of ruffling events increases in the slide-distal region of LPS treated cells and the slide-proximal region of CSF treated cells, while peak volume increases for slide-distal LPS only. Ruffling events are defined as a ruffle structure tracked over a time period in which the peak volume is at least  $14.5 \mu\text{m}^3$  (5000 voxels) and the volume at the start and end of the time period is less than half this peak value (or the end of the capture period is reached). (A) Number of ruffling events per cell, before and after treatment. (B) Median peak ruffle size per cell, before and after treatment. Points above the diagonal line indicate an increase in number of events or median size. Plots were produced with the ggplot2 package in the R statistical programming language, using the tabular output from the object tracking code.

A range of additional object features are automatically generated by our system, and several were analysed but not included here, as no clear experimental effect was found. Duration of ruffling events was considered as a possible contributing factor to the overall increase in activity, but no change was detected for either CSF or LPS, suggesting that ruffle deployment but not structure is modulated in different physiological conditions. The maximum extension of each structure from the cell surface was measured, and the plate-proximal structures were found to be substantially more elongated on average, but this difference did not depend on treatment. Further null results were given by measurements of filopodia size and the number of filopodia associated with each ruffle.

#### **Tent pole ruffling motifs occur within a diversity of ruffling behaviour**

One of the key motivations for pursuing a systematic quantitative analysis of LLSM macrophage data was to develop a more complete understanding of tent-pole ruffling, including the frequency of these events and the variability between them. The plate-distal regions in the LPS treated macrophages showed a high level of ruffle and filopodia/tent pole activity (Fig. 4), and examples of filopodia ruffling were readily found (Additional file 4, 5: Supplementary Videos 3,4). However, an initial visual assessment did not find that the majority of ruffling events (Fig. 5) corresponded directly with tent pole ruffling, although apparent tent poles were often present, and previously described behaviours could be identified. This visual assessment was greatly complicated by the rapid and dynamic nature of actin turnover. To gain a more systematic understanding we undertook a complete analysis of the first 13 min (150 timesteps) in an LPS treated sample containing 4 macrophage cells.

Firstly, we manually identified four stereotypical examples in which (at a specific time) a pair of prominent tent poles were joined by a ruffle “veil”, and analysed these examples using the object statistics to develop a filter to detect all similar cases. The selected filter required: (1) a ruffle object of volume  $10.15 \mu\text{m}^3$  (3500 voxels) or greater, with at least  $2 \mu\text{m}$  mean and  $4 \mu\text{m}$  maximum distance from the cell surface, and at least  $1.5 \mu\text{m}$  mean distance from the plate; (2) two filopodia/tent pole objects adjacent to the ruffle, each with volume  $0.22 \mu\text{m}^3$  (75 voxels) or greater, and minimum contact score with the ruffle of 240 (this arbitrary score is based on pairs of adjacent voxels, one in each object).

Computationally applying these filter criteria to the 4 macrophage cells over 150 timesteps, we identified 133 examples belonging to 33 distinct tracked ruffles. We then sampled 10 of these tracked ruffles at random for detailed visual analysis. In 9 cases we observed rapid twisting of the tent pole pair during or after the collapse of the

connecting ruffles, which we consider the canonical feature of tent pole ruffling and a key closure mechanism for macropinosomes [9]. Variations in the origin and timing of the ruffles and tent poles were also recorded as follows. Ruffles developed from a) part of a pre-existing ruffle structure (4 cases, see Additional file 7: Supplementary Video 6); b) de novo on the dorsal region of the cell (3 cases, of which only 1 features tentpoles during the initial emergence, see Additional file 6: Supplementary Video 5); c) de novo on the proximal region of the cell, attached to the plate (2 cases). Tent poles appeared either during the terminal phase, after the ruffle formed a semi-circle and started to close (5 cases, see Additional file 6: Supplementary Video 5); tent poles exhibited a configuration with a “veil” in between forming prior to the final phase (3 cases, see Additional file 7: Supplementary Video 6) and a tent pole pair appeared to emerge from the hinge point of a large ruffle (1 case, Additional file 8: Supplementary Video 7).

Our approach of computational filtering combined with visual analysis allows us to detect different features within a range of complex and multi-faceted ruffles, including distinctive tent pole ruffling events. Importantly, the variety of behaviours observed here suggests that tent pole ruffling exists on a continuum with non-tent pole ruffling. The fact that tent pole ruffling and the involvement of filopodia in ruffle formation are increased by LPS activation of macrophages [9], suggests that this continuum of ruffling morphologies is ‘tuneable’. By revealing that ruffling constitutes a range of membrane formations, rather than discrete subtypes of ruffles, now frames future studies that will dissect the physiological demands and molecular functions that drive this variation.

## Conclusions

Realising the full potential of terabyte scale microscopy data requires new approaches to image analysis. Even storing and transferring image datasets on this scale can overwhelm the local IT resources available to a typical researcher; a high-end workstation running specialised software is certainly capable of visualising an individual image stack produced by an LLS microscope, as well as performing analytical processes such as thresholding and object counting, but running a systematic quantitative analysis over hundreds of time steps is not feasible. Processes that can be run on high performance computing facilities are required, and the algorithms employed must be robust enough to produce reliable results with minimal recourse to manual editing and adjustment. Machine learning methods provide a promising approach and have proved capable of producing robust semantic segmentations of microscopy imaging. Segmentation into defined tissue classes requires supervised machine learning, meaning that data must be labelled for training, and the models derived from this training data represent a sophisticated means to extrapolate to the larger dataset. But such a model can only be reliably applied on data that is well represented by the training data selection, and may be invalidated by changes in experimental conditions, markers and microscopy settings. In response to these challenges, we selected a machine learning approach and software platform featuring rapid model training with minimal data labelling requirements. We especially emphasised the development of a custom visualisation tool, to optimise training data selection and the assessment of draft segmentations.

The LLAMA image analysis system was designed to semi-automatically detect and decipher characteristics of cell membrane protrusions from large scale data acquired through the LLSM, with integration of statistical and visual analysis. Semantic segmentation provides the starting point for an object delineation and tracking system designed to convert the segmentation into a rich set of data at the level of individual structures over time. The interactive steps in the pipeline use a visual approach to customising the algorithms without the need for extensive training data; processing of additional data can then be fully automated. While designed as a general-purpose tool, initially our primary goal was to analyse macrophage ruffles, filopodia and the filopodia-like “tent poles” embedded within ruffles. The complex behaviour of macrophage ruffles proved a particular challenge, and the system features a novel approach designed for separating and tracking these structures, under the constraint that segmentation and other computationally difficult tasks are performed for each time step in isolation. The algorithm is parameterised to allow customisation to other structure types and was successfully adapted to the cell body and filopodia classes. Our system is designed for flexible deployment and is suitable for cluster or cloud computing, providing a general-purpose system for tracking and quantifying structures in large scale 4D microscopy data. The included code also provides a template for deploying an ImageJ based processing pipeline on high performance computing facilities.

Using this system, we were able to perform for the first time a systematic quantitative analysis of RAW264.7 macrophage cells under two experimental conditions, LPS and CSF treatment. We were able to tease out features that varied under different stimulation; these included the frequency of occurrence, location (proximal or distal to plate) as well as size of the actin structures on the cells. In addition, there are aspects of these actin structure where no differences were found, such as the mean lifetime and maximum distance from the cell surface. These physical variations in actin structures reflect the different physiological requirement of the cell under LPS (pathogen uptake) versus CSF (cell migration and invasion).

We were also able to conduct a systematic qualitative analysis of tent pole ruffling in LPS treated cells, using the capacity of the LLAMA visualiser to link numerical and visual data. This analysis highlighted the considerable challenge of completely characterising macrophage membrane protrusions, with the largest ruffles in particular exhibiting complex and multi-faceted behaviour. Most importantly, we were able to demonstrate continuity between tent pole ruffling and the wave-like ruffling behaviour as traditionally understood.

The LLAMA pipeline is designed to be broadly applicable, with an approach based on distinguishing, tracking, and quantifying structures that is not bound to the specific problem of macrophage surface projections. The system is modular, tuneable via a range of parameters, and based on fully open code. As such our system is suitable for use on a wide range of large-scale cellular and other microscopy data. The macrophage analysis presented demonstrates the value of the system, deriving biologically significant results which could not practically be replicated using standard methods.

### Abbreviations

LLSM: Lattice light sheet microscope; LPS: Bacterial lipopolysaccharide; CSF (CSF-1): Macrophage colony stimulating factor.

## Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12859-021-04324-z>.

**Additional file 1.** Detailed instructions for the image analysis pipeline, including use of the visualiser.

**Additional file 2.** Supplementary Video 1. Supplementary Videos 1 and 2 demonstrate the key features of the visualiser and its use as part of the platform. Video 1 covers features using raster (voxel based) imaging, including training data selection and viewing image segmentations.

**Additional file 3.** Supplementary Video 2. Demonstration of key object based features of the visualiser, including review of object splitting and tracking results.

**Additional file 4.** Supplementary Video 3. Supplementary videos 3 and 4 show two selected examples of tent pole ruffling events in LPS treated macrophage cells, showing LLS imaging and corresponding segmentation probability map. Segmentation colours are as described in Fig. 2. Videos show direct capture from the LLAMA visualiser with no additional image processing.

**Additional file 5.** Supplementary Video 4.

**Additional file 6.** Supplementary Video 5. Supplementary Videos 5–7 show representative examples selected from ten randomly sampled ruffling events that include a “tent pole ruffle” configuration in which a pair of filopodia / tent poles are connected by a prominent ruffle. Segmentation colours are as described in Fig. 2. Videos show direct capture from the LLAMA visualiser with no additional image processing.

**Additional file 7.** Supplementary Video 6.

**Additional file 8.** Supplementary Video 7.

### Acknowledgements

The authors wish to thank Tatiana Khromykh for technical assistance. Some of the LLSM datasets were acquired at AIC Janelia and we acknowledge the support of HHMI and the Gordon and Betty Moore Foundation. Most of the LLSM datasets were produced at IMB with additional technical support from 3i. We wish to acknowledge The University of Queensland's Research Computing Centre (RCC) for its support in this research in providing high performance computing facilities.

### Authors' contributions

JGL wrote all code, ran computations, performed data analysis, and wrote the majority of the manuscript. YWHK performed experimental design, sample preparation, staining, and LLSM imaging for the data analysed, and contributed significantly to the manuscript. AAW and NDC produced LLSM imaging used in preliminary work, developed protocols for use of the LLS microscope at the IMB, and provided crucial technical advice and help with subsequent LLSM microscopy. AAW, JLS and NAH conceived and planned the project. NAH provided overall direction on the mathematical and computational work, while JLS directed the biological investigation. Both JLS and NAH made substantial contributions and revisions to the manuscript. All authors reviewed and approved the manuscript.

### Funding

Research funding was from the Australian Research Council Discovery Grant DP180101910 (NAH, JLS, AAW) and from the National Health and Medical Research Council of Australia Investigator Grant APP1176209 (JLS). LLSM imaging was performed using the resources of the IMB Microscopy facility, with funding from the Australian Research Council Linkage Grant LE170100206, and the Australian Cancer Research Foundation funded Cancer Ultrastructure and Function Facility.

### Availability of data and materials

Sample data designed to demonstrate most features of the software is provided at <https://doi.org/10.14264/3084db2>.

## Declarations

### Ethics approval and consent to participate

Not applicable.

### Availability and requirements

Project name: LLAMA. Project home page: [github.com/jameslefevre/4D-microscopy-pipeline](https://github.com/jameslefevre/4D-microscopy-pipeline). Operating system(s): Platform independent, but computational pipeline configured for Linux. Programming language: Java, Groovy, Processing. Other requirements: Fiji installation with the following additional plugin libraries/update sites: 3D\_ImageJ\_Suite, ImageScience, IJPB\_plugins. Licence: MIT. Consent for publication: Not applicable.

### Competing interests

The authors declare that they have no competing interests.

**Author details**

<sup>1</sup>Institute for Molecular Bioscience, The University of Queensland, Brisbane, QLD, Australia. <sup>2</sup>Research Computing Centre, The University of Queensland, Brisbane, QLD, Australia.

Received: 18 February 2021 Accepted: 10 August 2021

Published online: 19 August 2021

**References**

1. Arbelle A, Raviv TR. Microscopy cell segmentation via convolutional LSTM networks. 2018
2. Arganda-Carreras I, Kaynig V, Rueden C, Eliceiri KW, Schindelin J, Cardona A, Sebastian Seung H. Trainable Weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*. 2017;33(15):2424–6. <https://doi.org/10.1093/bioinformatics/btx180>.
3. Arganda-Carreras I, Fernández-González R, Muñoz-Barrutia A, Ortiz-De-Solorzano C. 3D reconstruction of histological sections: application to mammary gland tissue. *Microsc Res Tech*. 2010;73(11):1019–29. <https://doi.org/10.1002/jemt.20829>.
4. Barthwal MK, Anzinger JJ, Xu Q, Bohnacker T, Wymann MP, Kruth HS. Fluid-phase pinocytosis of native low density lipoprotein promotes murine M-CSF differentiated macrophage foam cell formation. *PLoS ONE*. 2013;8(3): e58054. <https://doi.org/10.1371/journal.pone.0058054>.
5. Beucher S, Lantuéj C. Use of watersheds in contour detection. In: Paper presented at the international workshop on image processing: real-time edge and motion detection/estimation, Rennes, France. 1979. <http://cmm.enscm.fr/~beucher/publi/watershed.pdf>
6. Canton J, Schlam D, Breuer C, Gutschow M, Glogauer M, Grinstein S. Calcium-sensing receptors signal constitutive macropinocytosis and facilitate the uptake of NOD2 ligands in macrophages. *Nat Commun*. 2016;7:11284. <https://doi.org/10.1038/ncomms11284>.
7. Castilla C, Maska M, Sorokin DV, Meijering E, Ortiz-de-Solorzano C. 3-D quantification of filopodia in motile cancer cells. *IEEE Trans Med Imaging*. 2019;38(3):862–72. <https://doi.org/10.1109/TMI.2018.2873842>.
8. Chen B-C, Legant WR, Wang K, Shao L, Milkie DE, Davidson MW, et al. Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. *Science (New York, NY)*. 2014;346(6208):1257998–1257998. <https://doi.org/10.1126/science.1257998>.
9. Condon ND, Heddleston JM, Chew T-L, Luo L, McPherson PS, Ioannou MS, et al. Macropinosome formation by tent pole ruffling in macrophages. *J Cell Biol*. 2018;217(11):3873–85. <https://doi.org/10.1083/jcb.201804137>.
10. Driscoll MK, Welf ES, Jamieson AR, Dean KM, Isogai T, Fiolka R, Danuser G. Robust and automated detection of subcellular morphological motifs in 3D microscopy images. *Nat Methods*. 2019;16(10):1037–44. <https://doi.org/10.1038/s41592-019-0539-z>.
11. Fritz-Laylin LK, Riel-Mehan M, Chen BC, Lord SJ, Goddard TD, Ferrin TE, et al. Actin-based protrusions of migrating neutrophils are intrinsically lamellar and facilitate direction changes. *Elife*. 2017. <https://doi.org/10.7554/eLife.26990>.
12. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. *IEEE Trans Pattern Anal Mach Intell*. 2020;42(2):386–97. <https://doi.org/10.1109/TPAMI.2018.2844175>.
13. Heddleston JM, Chew T-L. Light sheet microscopes: novel imaging toolbox for visualizing life's processes. *Int J Biochem Cell Biol*. 2016;80:119–23. <https://doi.org/10.1016/j.biocel.2016.10.002>.
14. Isensee F, Jäger PF, Kohl SAA, Petersen J, Maier-Hein KH. Automated design of deep learning methods for biomedical image segmentation. 2019. <https://doi.org/10.1038/s41592-020-01008-z>.
15. Kheir WA, Gevrey J-C, Yamaguchi H, Isaac B, Cox D. A WAVE2-Abi1 complex mediates CSF-1-induced F-actin-rich membrane protrusions and migration in macrophages. *J Cell Sci*. 2005;118:5369–79.
16. Kuhn HW. The Hungarian method for the assignment problem. *Nav Res Logist*. 1955;2:83–97.
17. Legland D, Arganda-Carreras I, Andrey P. MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics*. 2016. <https://doi.org/10.1093/bioinformatics/btw413>.
18. Lin XP, Mintern JD, Gleeson PA. Macropinocytosis in different cell types: similarities and differences. *Membranes (Basel)*. 2020. <https://doi.org/10.3390/membranes10080177>.
19. Long F, Simpson JH, Myers EW, Peng H, Ruan Z. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat Biotechnol*. 2010;28(4):348–53. <https://doi.org/10.1038/nbt.1612>.
20. Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH Comput Graph*. 1987;21(4):163–9. <https://doi.org/10.1145/37402.37422>.
21. Lutton EJ, Collier S, Bretschneider T. A curvature-enhanced random walker segmentation method for detailed capture of 3D cell surface membranes. *IEEE Trans Med Imaging*. 2021;40(2):514–26. <https://doi.org/10.1109/TMI.2020.3031029>.
22. Manley HR, Potter DL, Heddleston JM, Chew TL, Keightley MC, Lieschke GJ. Frontline science: dynamic cellular and subcellular features of migrating leukocytes revealed by in vivo lattice lightsheet microscopy. *J Leukoc Biol*. 2020;108(2):455–68. <https://doi.org/10.1002/JLB.3HI1020-589R>.
23. Meijering E. FeatureJ: an ImageJ plugin suite for image feature extraction. 2015. <http://imagescience.org/meijering/software/featurej/>
24. Norbury CC, Hewlett LJ, Prescott AR, Shastri N, Watts C. Class I MHC presentation of exogenous soluble antigen via macropinocytosis in bone marrow macrophages. *Immunity*. 1995;3(6):783–91. [https://doi.org/10.1016/1074-7613\(95\)90067-5](https://doi.org/10.1016/1074-7613(95)90067-5).
25. Ollion J, Cochenne J, Loll F, Escudé C, Boudier T. TANGO: a generic tool for high-throughput 3D image analysis for studying nuclear organization. *Bioinformatics*. 2013;29(14):1840–1. <https://doi.org/10.1093/bioinformatics/btt276>.
26. Pixley FJ. Macrophage migration and its regulation by CSF-1. *Int J Cell Biol*. 2012;2012: 501962. <https://doi.org/10.1155/2012/501962>.

27. Qin Y, Wang W, Liu W, Yuan N. Extended-Maxima transform watershed segmentation algorithm for touching corn kernels. *Adv Mech Eng*. 2013. <https://doi.org/10.1155/2013/268046>.
28. Reas C, Fry B. Processing: programming for the media arts. *J Hum Centred Syst*. 2006;20(4):526–38. <https://doi.org/10.1007/s00146-006-0050-9>.
29. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention—MICCAI 2015*, vol. 9351. Cham: Springer International Publishing;2015. pp. 234–241
30. Rueden CT, Schindelin J, Hiner MC, DeZonia BE, Walter AE, Arena ET, Eliceiri KW. Image J2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics*. 2017;18(1):529–529. <https://doi.org/10.1186/s12859-017-1934-z>.
31. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, et al. Fiji: an open-source platform for biological-image analysis. *Nat Methods*. 2012;9(7):676. <https://doi.org/10.1038/NMETH.2019>.
32. Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nat Methods*. 2012;9(7):671–5.
33. Stow JL, Condon ND. The cell surface environment for pathogen recognition and entry. *Clin Transl Immunol*. 2016;5(4): e71. <https://doi.org/10.1038/cti.2016.15>.
34. Stow JL, Hung Y, Wall AA. Macropinocytosis: insights from immunology and cancer. *Curr Opin Cell Biol*. 2020;65:131–40. <https://doi.org/10.1016/j.ccb.2020.06.005>.
35. Swanson JA. Shaping cups into phagosomes and macropinosomes. *Nat Rev Mol Cell Biol*. 2008;9(8):639–49. <https://doi.org/10.1038/nrm2447>.
36. Veltman DM, Williams TD, Bloomfield G, Chen BC, Betzig E, Insall RH, Kay RR. A plasma membrane template for macropinocytotic cups. *Elife*. 2016. <https://doi.org/10.7554/eLife.20085>.
37. Wählby C, Sintorn IM, Erlandsson F, Borgefors G, Bengtsson E. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *J Microsc*. 2004;215(1):67–76. <https://doi.org/10.1111/j.0022-2720.2004.01338.x>.
38. Williams L, Ridley A. Lipopolysaccharide induces actin reorganization and tyrosine phosphorylation of Pyk2 and Paxillin in monocytes and macrophages. *J Immunol*. 2000;164(4):2028–36.
39. Witten IH, Frank E, Hall MA, Pal CJ. *Data mining: practical machine learning tools and techniques*. Amsterdam: Elsevier; 2016.
40. Zanon I, et al. CD14 controls the LPS-induced endocytosis of toll-like receptor 4. *Cell*. 2011;147(4):868–80. <https://doi.org/10.1016/j.cell.2011.09.051>.
41. Zinchuk V, Grossenbacher-Zinchuk O. Machine learning for analysis of microscopy images: a practical guide. *Curr Protoc Cell Biol*. 2020. <https://doi.org/10.1002/cpcb.101>.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

